

Cnosos: una biblioteca para enrutamiento en Arduino Car Kits

Ana I. Gómez

Esc. Tec. Sup. de Ingeniería Informática
Universidad Rey Juan Carlos
ana.gomez.perez@urjc.es

Domingo Gómez, Lara González de Cos

Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria
{domingo.gomez, lara.gonzalez}@unican.es

Resumen

La teoría de grafos es un contenido troncal dentro del «Libro Blanco de la Ingeniería Informática» publicado por la ANECA, siendo fundamental en el diseño de algoritmos y la inteligencia artificial. Debido a que las herramientas educativas para la teoría de grafos están despertando un gran interés por la dificultad de la materia y que una posible aplicación de la teoría de grafos es el control de sistemas robóticos, es interesante explorar su aplicación conjunta a la docencia. Arduino es una plataforma hardware de bajo coste adecuada para prototipado, siendo popular su uso en diferentes niveles del sistema educativo español. En este trabajo presentamos un recurso software educativo que permite una implementación sencilla de diversos algoritmos de enrutamiento en coches robóticos. Esta biblioteca, junto con un simulador visual sencillo, ha sido utilizada para la impartición de varias sesiones de laboratorio de la asignatura de «Algorítmica y Complejidad» del grado de ingeniería informática de la Universidad de Cantabria. Los resultados preliminares son prometedores. Los beneficios de utilizar esta biblioteca incluyen la promoción del trabajo en equipo, el aumento de la tasa de participación y la autonomía de aprendizaje. Además, ofrece una introducción a los dispositivos con recursos limitados.

Abstract

Graph theory is a main subject within the “Libro Blanco de la Ingeniería Informática” published by ANECA, being fundamental in the design of algorithms and artificial intelligence. Given that educational tools for graph theory are attracting great interest due to the difficulty of the subject and the applications of graph theory to the control of robotic systems, it is interesting to explore a joint approach in teaching. Arduino is a low-cost hardware platform suitable for prototyping, being popular in different levels of the Spanish educational system. In this paper we present an educational software resource that allows a simple imple-

mentation of various routing algorithms in robotic cars. This library, together with a intuitive visual simulator, has been used in several laboratory sessions for “Algorithms and Complexity”, a subject of the Computer Science Degree at the University of Cantabria. Preliminary results are promising. The benefits of using this library include the promotion of teamwork, increased participation rate and learning autonomy. In addition, it provides an introduction to resource-constrained devices.

Palabras clave

Algorítmica, tecnologías educativas, robótica, Arduino, enseñanzas universitarias.

1. Introducción

El uso de las nuevas tecnologías de la información permite facilitar el aprendizaje, así como abordar problemáticas actuales con nuevas herramientas que motiven y refuercen el conocimiento del alumnado. Por ello, la sociedad demanda que los docentes adapten continuamente sus dinámicas de enseñanza aprovechando el potencial de las TIC en el aula. Desafortunadamente, su uso conlleva un trabajo extra para el docente, que incluye la búsqueda de recursos educativos y una adaptación continua que limita su adopción de forma generalizada. Esta problemática ha sido detectada recientemente [6], mediante un cuestionario realizado a los profesores del Máster de Formación de Profesorado de la Universidad Autónoma de Madrid. Los resultados muestran que las TIC siguen siendo un gran reto en las aulas tanto para docentes como para estudiantes.

Por tanto, es clave diseminar nuevas herramientas tecnológicas con el potencial de abordar las cuatro áreas de la educación STEM (Science, Technology, Engineering and Maths) como complemento al enfoque tradicional. Dentro de esta filosofía, se puede destacar R2P2 [4], un simulador de robots en espacios bi-

dimensionales. Esta cita está sacada del artículo: «el hardware robótico real presenta costes y riesgos adicionales que, habitualmente, son difícilmente aceptables en el contexto de un aula». Esto explica el porqué de la escasez de propuestas que usen plataformas de cómputo alternativas al PC en la educación universitaria.

Después de una revisión de la literatura, hemos encontrado propuestas de uso de hardware específico como microcontroladores [3] y otras herramientas relacionadas con el denominado Internet de las cosas [9]. Aliagas et al.[1] describen el diseño de un sistema de varios procesadores que permite a los alumnos analizar la escalabilidad de los entornos de desarrollo MPI a un precio de alrededor de cinco mil euros. Queremos hacer notar que este presupuesto no es siempre asumible.

En este trabajo proponemos un recurso software educativo, que integra varias áreas STEM, aplicado a la enseñanza de diversos algoritmos de enrutamiento utilizando la plataforma Arduino. Esta primera aproximación muestra que la comprensión de las bases teóricas y conceptos abstractos es más precisa, más clara y más atractiva con la experimentación visual y complementa la metodología tradicional de clases magistrales y prácticas de programación con PCs. La herramienta se compone de una biblioteca Arduino de desarrollo propio denominada Cnosos y un simulador software de comportamiento. Esto minimiza tanto costes, como riesgos asociados al uso de robots en el aula [4]. Se presentan en primer lugar los costes asociados al proyecto y una propuesta de prácticas y, finalmente, las primeras conclusiones de la experiencia docente analizando principalmente el impacto en las calificaciones obtenido en la asignatura «Algorítmica y Complejidad» durante varios cursos.

2. Materiales y métodos

La biblioteca Cnosos, desarrollada en el lenguaje C/C++, está pensada para ser utilizada en cualquier coche robótico basado en el microcontrolador Arduino, mediante la configuración adecuada de pines de conexión. La mayoría de los modelos de coche robótico implementan una funcionalidad común mediante sensores siguelíneas basados en infrarrojos, así como motores y una comunicación básica basada en bluetooth o un mando infrarrojo.

En la experiencia docente se han utilizado dos modelos, el modelo Elegoo UNO R3 Project Smart Robot Car Kit mostrado en la Figura 1. El precio de un kit ronda los 70€ (sin gastos de envío). Por otro lado, se construyeron modelos clónicos haciendo la estructura con una impresora 3D y comprando los componentes electrónicos por un precio aproximado de 50€.

En total se prepararon cuatro coches robóticos pa-

ra impartir sesiones de laboratorio a grupos de treinta estudiantes, trabajando en varios grupos compartiendo el material necesario (cargadores de batería, conectores,...).

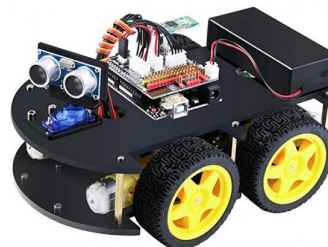


Figura 1: Elegoo UNO R3 Project Robot Car Kit.

Como entorno de programación de los coches robóticos se empleó Arduino Studio, disponible gratuitamente en la página del proyecto Arduino <https://www.arduino.cc/>, con la biblioteca Cnosos. Este desarrollo «software» se basó en el principio «Keep It Simple for Students»[8], intentando abstraer los detalles físicos (sensores, ruedas, etc).

La biblioteca Cnosos está pensada con el objetivo de que los coches robóticos recorran rutas trazadas mediante cinta negra en el suelo que emula la organización en nodos y aristas de un grafo. Las aristas se numeran según el orden de visita en sentido horario, empezando después de la etiqueta del nodo (ver Figura 2). La tarea de navegación se divide en partes, estando resumido el movimiento del coche robótico a las siguientes órdenes:

- `lee_nodo`: Recorre el perímetro de un vértice almacenando la etiqueta del nodo, el grado del nodo y el número de arista por la que se ha accedido.
- `sal_izq`: Toma la primera salida a la izquierda.
- `sal`: Sale del vértice por la arista indicada.

Además se proporciona una función `espera_mando` para la comunicación con un mando de infrarrojos o mediante un módulo Bluetooth.

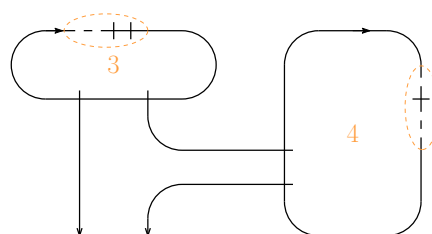


Figura 2: Diseño de grafo reproducido en el suelo.

2.1. Diseño de los materiales

Comentaremos brevemente el diseño de las prácticas, así como los objetivos de aprendizaje. Los materiales están diseñados para debate y la búsqueda de soluciones en equipo. Cada actividad está motivada por un problema subyacente que el estudiante debe resolver mediante la aplicación de un algoritmo visto en las clases de teoría. Previamente, se deben fomentar debates sobre la resolución del problema usando el modelo de caja negra [2]. Para ello, los estudiantes disponen de una serie de programas que muestran la respuesta del coche robótico a las órdenes implementadas. Por otro lado, debido a las restricciones de cómputo, los estudiantes deben reflexionar sobre el concepto de «online algorithms», donde la información se va recogiendo a medida que se realizan otras operaciones [7].

«Arduino en la isla de Minos»: El enunciado resume la leyenda de Teseo en el laberinto de Creta. Los estudiantes deben emular a Ariadna y programar un algoritmo de recorrido de grafos para recorrer completamente el grafo dibujado en el suelo. La idea es una extensión de la práctica Salida del laberinto [8] y también de la actividad «kidbot» donde un estudiante da ciertas órdenes a un robot para dirigirse de un punto de inicio a una meta. Se puede ver como un primer paso para llevar el planificador de caminos descrito por Cobos *et al.* [4] a un modelo real.

«Arduino a la búsqueda de atajos»: Se reta al estudiante a implementar el algoritmo de búsqueda en anchura. El objetivo de la práctica es, tomando un nodo como base, calcular la distancia en aristas que lo separa de cada uno de los demás, además de poder alcanzar cada uno de esos vértices por un camino de longitud mínima. Esta tarea plantea el reto de implementar estructuras de datos auxiliares como colas y árboles.

«Arduino en Königsberg»: La tarea propuesta consiste en explorar un grafo para determinar si contiene un camino euleriano y, en caso afirmativo, recorrerlo. Un trabajo clásico [5] proporciona un algoritmo parecido al conocido algoritmo de Hielhozer planteado con la siguiente filosofía:

«if one must move from node to node along an edge and cannot obtain node information until the node is reached».

2.2. Simulador Gráfico

Para depurar las soluciones de los estudiantes y disminuir las dificultades técnicas de comportamiento del coche robótico, los estudiantes disponían inicialmente de un pequeño simulador en modo texto escrito en C que simulaba la lectura de los sensores siguelíneas. Posteriormente se trabajó en el desarrollo de un simulador gráfico programado en Python, incluyendo un

traductor de C a Python. Es de código libre, y permite la depuración de errores paso a paso con una interfaz intuitiva (ver Figura 3). El simulador visual soluciona parte de los problemas observados en las sesiones prácticas y ha sido desarrollado como Trabajo Fin de Grado en Ingeniería Informática por Sergio Palomera. Para mostrar el comportamiento esperado, el usuario puede especificar tanto el circuito a recorrer como el fichero fuente de código ¹.

3. Resultados

Las sesiones prácticas se realizaron dentro de varios cursos sucesivos de la asignatura «Algorítmica y Complejidad» de segundo curso del grado en informática de la Universidad de Cantabria. Durante el curso se realizan cinco prácticas grupales de laboratorio. La experiencia docente con Arduino se planificó como la última práctica del curso, tradicionalmente con mayor absentismo.

En el curso previo a la realización la experiencia docente (2016-17), el número de matriculados fue de ochenta y dos estudiantes, posteriormente este número fue de sesenta y cinco (2017-18) y sesenta y un estudiantes (2018-19). Finalmente, la experiencia se interrumpió por las circunstancias sanitarias durante el curso 2019-20 con setenta y seis matrículas. Las calificaciones medias finales durante cada curso fueron respectivamente 65, 75, 58 y 77 sobre 100. Una comparativa gráfica de las calificaciones de las prácticas para los tres primeros cursos (Figura 4) muestra que la mediana de la calificación es la máxima del curso en la última práctica durante los cursos que se realiza la experiencia (2017-18 y 2018-19). Respecto a las calificaciones finales para cada curso, se puede observar que existe una variación entre los cursos independiente del número de estudiantes matriculados y que sigue lo observado en las calificaciones de las prácticas.

Finalmente los profesores han constatado externalidades positivas, con un aumento en la asistencia a las últimas sesiones de laboratorio y una mejora del ambiente de trabajo y de colaboración entre grupos, así como el interés posterior de los estudiantes por este tipo de plataformas hardware para sus proyectos. Las encuestas de opinión de la asignatura se mantuvieron favorables, descendiendo en el número de comentarios realizados con una participación similar. Por motivos de espacio, en la tabla 1 solo se reflejan varios comentarios críticos recibidos a las prácticas y posibles mejoras a realizar en el futuro.

¹Se puede descargar de <https://gitlab.com/serpa98/simulador-arduino.git>

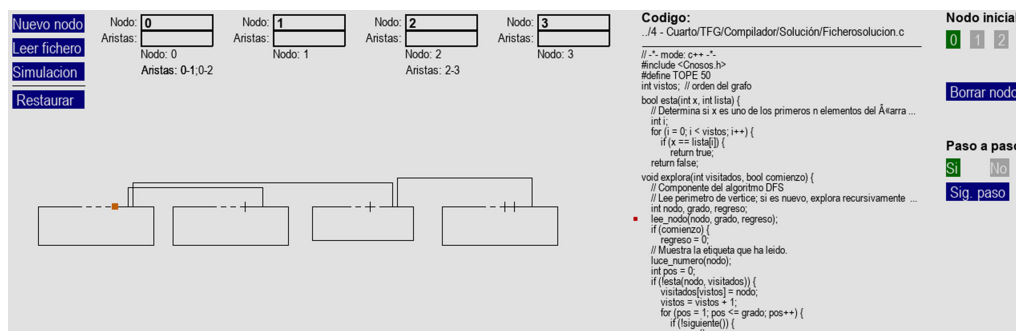


Figura 3: Simulador visual. La posición en el grafo (cuadrado rojo) depende de la línea de código (puntero rojo).

Curso	Comentarios
2016-17	1) Prácticas deberían ser más acordes con la asignatura 2) La materia que estamos dando apenas tiene utilidad (algoritmos desconocidos hasta para internet) y otros temarios son más propios de una carrera de matemáticas que de informática.
2017-18	1) Para próximos cursos deberían cambiarse las prácticas, ya que muchas de ellas tenían gran dificultad. 2) Las prácticas tendrían que ser más asequibles de realizar.
2018-19	1) Las prácticas tendrían que ser más guiadas.

Cuadro 1: Comentarios verbatim tomados de diversas encuestas anónimas a los estudiantes.

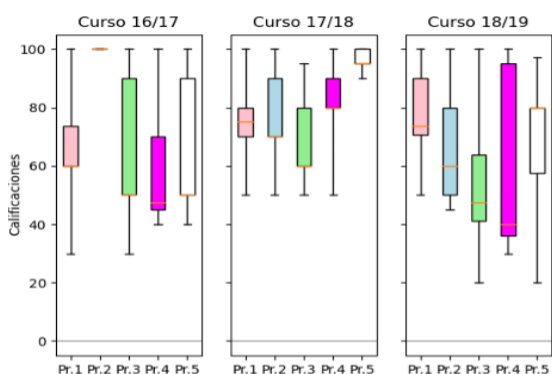


Figura 4: Calificaciones numéricas para las prácticas (Pr.1 a Pr.5). Cada caja representa el rango del 90 % de los discentes y su mediana (línea naranja). La Pr.5 utiliza la librería Cnosos (17/18 y 18/19) y una práctica en PC (16/17) respectivamente.

Referencias

- [1] Carles Aliagas, Pere Millán Marco, Carlos Maria Molina Clemente, y Roc Meseguer Pallarès. Minimulticomputador de bajo coste. En *Actas de las XXV Jornadas sobre Enseñanza Universitaria de la Informática*, pp. 31–38, 2019.
- [2] Martin Cápay y Martin Magdin. Alternative methods of teaching algorithms. *Procedia - Social and Behavioral Sciences*, pp. 431–436, 2013.
- [3] Carlos Catalán Cantero y Alfonso Blesa Gascón. Enseñanza de sistemas empujados: de Arduino a Raspberry Pi. En *Actas de las XXII Jornadas sobre Enseñanza Universitaria de la Informática*, pp. 351–354, 2016.
- [4] Mario Cobos, María D. R-Moreno, y David F. Barrero. R2p2: Un simulador robótico para la enseñanza de inteligencia artificial. En *Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática*, pp. 285–292, 2020.
- [5] Jack Edmonds y Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical programming*, pp. 88–124, 1973.
- [6] Maria Flores Tena, M^a Carmen Ortega Navas, y M^a Carmen Sánchez Fuster. Las nuevas tecnologías como estrategias innovadoras de enseñanza-aprendizaje en la era digital. *Revista Electrónica Interuniversitaria de Formación del Profesorado*, 24(1), 2021.
- [7] Dennis Komm. Teaching the concept of online algorithms. *Olympiads in Informatics*, pp. 58–70, 2011.
- [8] Lluís Ribas Xirgo y A. Josep Velasco González. Aplicación del principio KISS a la enseñanza de los fundamentos de programación. En *Actas de las XXV Jornadas sobre Enseñanza Universitaria de la Informática*, volumen 4, pp. 55–62, 2019.
- [9] Jaume Segura García, Santiago Felici Castell, Juan José Pérez Solano, Miguel García Pineda, y Antonio Soriano Asensi. Incorporación de Internet de las Cosas (IoT) en la docencia universitaria en dos etapas: hardware y software. En *Congreso In-Red 2018*, pp. 1–10, 2018.