

Sistema didáctico para la enseñanza de la programación con metodologías de aprendizaje basado en problemas

Ricardo Coppo¹ Javier Iparraguirre¹ Germán Feres¹ Gustavo Ursua¹
Ana Cavallo²

¹Universidad Tecnológica Nacional - Facultad Regional Bahía Blanca

² Instituto Superior Juan XXIII - Bahía Blanca

Mail de contacto: rcoppo@frbb.utn.edu.ar

1. Resumen

La enseñanza de la programación de computadoras es una materia básica de las carreras relacionadas con las ciencias de la computación y de la ingeniería electrónica. Muchos estudios demuestran que la deserción inicial en este área es sumamente elevada. Para evitar esto se debería contar con elementos específicos y una metodología apropiada que motiven a los alumnos a visualizar en forma concreta los resultados de los programas escritos por ellos mismos. Se introdujo como elemento didáctico una placa electrónica basada en principios de hardware y software open-source de bajo costo para la realización de prácticas de programación cuyas entradas y salidas son sensores y actuadores reales. Al mismo tiempo se trabajó con una metodología basada en la resolución de problemas (PBL). Las primeras experiencias muestran un fuerte incremento en el interés del alumno en aprender a resolver correctamente la lógica del problema y percibir los resultados sobre el sistema.

Palabras clave: enseñanza de la programación, hardware didáctico, motivación, PBL.

2. Introducción

La mayoría de los educadores y docentes de programación coinciden en señalar que son pocos los estudiantes que afirman que aprender a programar una computadora es una tarea sencilla. El problema se agrava ya que la mayoría

de los cursos de programación se encuentran en los currículos de los primeros años de las carreras de ingeniería y de las ciencias de la computación, años que significan para la mayoría de los estudiantes transiciones importantes en sus hábitos de estudio, de vida, y comportamiento social. [4, 6]

La literatura especializada enfoca la enseñanza de la programación en casi su totalidad en tecnologías noveles (proyecciones, multimedios, y otros) que pueden ser aplicadas a la didáctica del docente y en la organización de sus clases. [7, 8] Menos atención ha sido dedicada a los factores cognitivos presentes en el proceso de aprendizaje. El estilo de aprendizaje y la motivación se destacan como factores cognitivos que afectan el desempeño del alumno.

La motivación, que por definición es “la dirección y magnitud del comportamiento humano”, tiene como factores indispensables la elección del curso de acción, la persistencia en el mismo y el esfuerzo que el alumno decidirá realizar. [1, 5]

Una vez decidido el curso de acción el factor emocional/afectivo tiene un rol preponderante. El alumno se relaciona no solo cognitivamente con su proyecto sino también en forma afectiva; característica que es fundamental para determinar el tiempo y esfuerzo que luego dedicará a la conclusión del mismo.

Tradicionalmente la motivación ha sido considerada bajo una perspectiva individualista. Sin embargo, como toda acción humana se rea-

liza en un contexto, este ejerce una influencia indiscutible en todo el proceso de aprendizaje.

Mediante la aplicación de las metodologías de aprendizaje basado en problemas PBL (*Problem Based Learning*) [1, 2] se intenta crear un contexto interactivo en el que al trabajar sobre la resolución de problemas individuales, se incrementa la autoestima y la cooperación entre los alumnos, integrando conocimientos adquiridos en otras áreas o materias.

En una primera evaluación, la mayoría de los docentes de informática, presienten que el aprendizaje de la programación en estos primeros años obedece más a una motivación extrínseca, como aprobar la materia con altas calificaciones o lucir como intelectual entre sus pares. Sin embargo, los alumnos que demuestran un verdadero interés personal en adquirir la capacidad de aprender a programar correctamente (motivación intrínseca) son los que luego logran los mejores resultados.

Visto esto, este trabajo intenta explorar el uso de un sistema de hardware y software que introducen fuertes factores de motivación intrínseca en los alumnos, generándoles verdaderos deseos de poder desarrollar programas de software.

3. Descripción de sistema

El sistema didáctico se divide en los siguientes subsistemas: un software de desarrollo que se ejecuta sobre una laptop estándar, una placa de hardware Arduino interconectada con la computadora y diversos sensores o actuadores que le permiten al hardware interactuar con el entorno o medio físico. (Figura 1).

3.1. Características del hardware

El elemento principal del hardware del sistema consta de una plataforma open-source denominada Arduino UNO [9], basada en un microcontrolador Atmega328 de la firma ATMEL. La plataforma dispone de 14 entradas/salidas digitales, 6 entradas analógicas, una interfase USB, y 32 KBytes de memoria FLASH. La interfase USB es capaz de suminis-

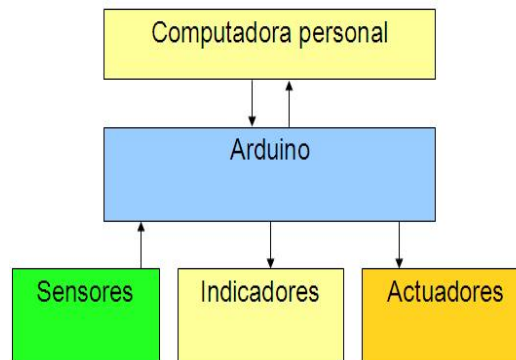


Figura 1: Diagrama en Bloques

trar la energía suficiente para la placa y algunos sensores básicos. Para mayores consumos se requiere el uso de fuentes externas adicionales.

La gran cantidad de entradas y salidas permite construir pequeños experimentos con sensores básicos de temperatura, LEDs, y displays de bajo consumo sin tener que recurrir a conexiones complejas. La placa se monta en un soporte de acrílico para incentivar al alumno a preguntar sobre sus componentes y el funcionamiento del mismo. Las conexiones externas aceleran la configuración del sistema para diferentes ejercicios. (Figura 2).

El hardware open-source es aquel que su propietario publica, por ejemplo a través de la Internet, los datos de diseño y los esquemas circuitales para que cualquier persona o empresa pueda fabricarlo o reproducirlo sin pagar licencias. (Requiere sí un expreso reconocimiento de los derechos de autor y de marcas registradas). Este modelo comercial se encuentra sustentado por el concepto del software open-source propulsado por los programadores del sistema operativo GNU/Linux.

La libre disponibilidad de la información de diseño permite a diversos proveedores de hardware la creación y venta de kits de componentes para su armado, y posibilitan al alumno entusiasta construir su propia plataforma de experimentación a bajo costo.

Además, en sincronía con la teoría open-source, se ha creado una comunidad de personas entusiastas que aportan ideas y planes para proyectos de mayor envergadura, muchos

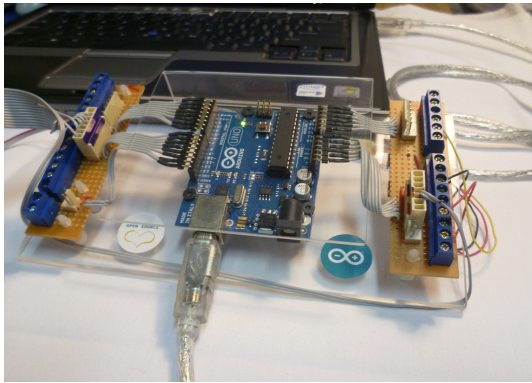


Figura 2: Placa y conexión a la PC

de los cuales poseen componentes lúdicos que refuerza la motivación inicial recibida en clase.

3.2. Características del sistema de desarrollo

La programación de la placa se realiza por medio de un entorno de desarrollo integrado (IDE) de distribución gratuita denominado “Arduino” el cual fue escrito en Java. Mediante el mismo, el usuario puede escribir aplicaciones en C o C++, las cuales se denominan “sketches”. Una vez compiladas estas últimas, son transmitidas a la placa por medio del puerto USB a través del IDE.

El sistema operativo embebido en la placa presenta una API de programación a través de funciones de “callback” de sencilla interpretación por parte del alumno que recién se inicia. En una aplicación didáctica inicial el programador solo debe definir las funciones de inicialización y la función de repetición, base del ciclo de control del hardware.

4. Aplicación educativa

En las primeras clases de la materia el alumno aprende sobre el sistema binario, los conceptos de algoritmo y las primeras instrucciones de bifurcación y ciclo en el lenguaje C. Estos temas resultan abstractos y difíciles de entender para la mayoría de los estudiantes no motivados.

Un ejercicio clásico, realizado en la mayoría de las cátedras de programación para fijar es-

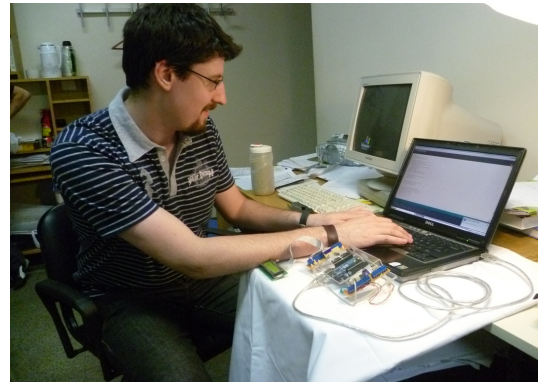


Figura 3: Alumno interactuando con el sistema

tos conceptos, es la realización de un contador en el que se hace conversión de números de base decimal a la base dos con salida a pantalla de línea de comando en modo terminal. El ejercicio, de nivel conceptual interesante para aprender a programar, no resulta motivador para el estudiante porque el alumno percibe su salida como algo antiguo o de poca utilidad.

Para realizar este mismo ejercicio sobre la placa Arduino solo bastan conectar algunos diodos emisores de luz (LED) a las salidas correspondientes. (Figura 4).

En la función de configuración se establecen constantes simbólicas y las habilitaciones correspondientes para las diferentes salidas. Esta función puede ser escrita por la cátedra para limitar al alumno al desarrollo del problema a resolver.

El alumno escribe e implementa la función de repetición que traduce un contador tradicional de un ciclo en su valor binario y lo convierte en las habilitaciones de los LEDs correspondientes para que se iluminen correctamente ejercitándose así los principios de programación señalados anteriormente.

Frecuentemente el primer intento del programa no funciona correctamente (luces fuera de secuencia, lógica errónea, entre otras fallas), pero a diferencia del ejercicio equivalente en pantalla, el alumno se motiva más, quiere ver su código funcionando, muestra mayor tenacidad en la búsqueda de la solución, y se retira del laboratorio con mayor satisfacción personal por haber cumplido el objetivo.

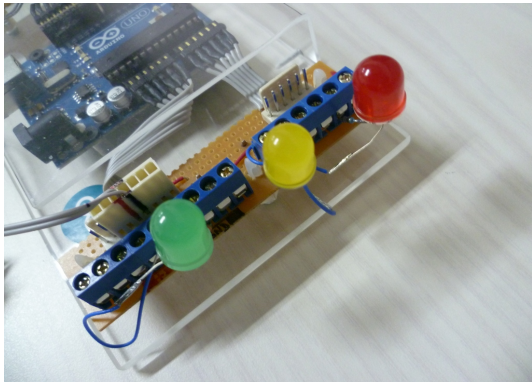


Figura 4: Arreglo de LEDs

Cabe señalar que la configuración presentada se adapta a ejercicios más complicados y a metodologías didácticas diferentes. Por ejemplo, fácilmente la configuración presentada se puede adaptar a la enseñanza de autómatas regulares simulando sistemas de alarmas, control de semáforos, electrodomésticos inteligentes, por mencionar solo algunos ejemplos.

5. Conclusiones y próximos pasos

El sistema integrado ha sido utilizado en cursos de Informática I de la Universidad Tecnológica, Facultad Regional Bahía Blanca en los que se enseña programación a los alumnos de primer año de Ingeniería Electrónica. Los primeros resultados han sido considerados altamente satisfactorios.

El hardware, de bajo costo, ha incentivado a varios estudiantes a realizar su adquisición para experimentación personal, perseverando así, en su interés por aprender a programar.

En nuestro diseño prevaleció la idea de experimentar con diversos sensores y medios de salida, en un esfuerzo para determinar experimentos motivadores y significativos para la enseñanza de la programación. En una versión final, más apta para cursos numerosos, se definirá una configuración básica de sensores y se construirá todo el sistema físicamente en una sola unidad. Eventualmente se podría extender el sistema básico a través de módulos independientes para periféricos más complejos o de mayor potencia.

Referencias

- [1] DÖRNYEI, Z., *Teaching and Researching Motivation*, Pearson Education, Malaysia, 2001.
- [2] O'KELLY J., GIBSON J.P., *Robocode & Problem Based Learning: A non-prescriptive approach to teaching programming*, In ITICSE '06: Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, 2006.
- [3] LIN H.T., KUO T.H., *Teaching programming technique with edutainment robot construction*, IEEE 2nd International Conference on Education Technology and Computer (ICETC), Shanghai, June 2010.
- [4] HUET I., PACHECO O.R., TAVARES J., WEIR G., *New Challenges in Teaching Introductory Programming Courses: a Case Study*, 34th ASEE/IEEE Frontiers in Education Conference, Savannah, Georgia, October, 2004.
- [5] JENKINS T., *Teaching Programming - A Journey from Teacher to Motivator*, 2nd Annual LTSN-ICS Conference, London, 2001.
- [6] JENKINS T., *On the Difficulty of Learning to Program*, 3rd Annual LTSN-ICS Conference, Loughborough University, 2002.
- [7] MATSUMURA K., DAISUKE S., AIGUO HE, *A C Language Programming Education Support System base on Software Visualization*, IEEE Joint Conferences on Pervasive Computing (JCPC), Taipei, December 2009.
- [8] SHYU Y.H., CHEN P.W., *PLL: A Programming Languages Lab System*, 21st International Conference on Distributed Computing Systems Workshops (ICDCSW01), Mesa, Arizona, USA, 2001.
- [9] ARDUINO TEAM, *Arduino Home Page*, <http://www.arduino.cc>, visitado: Marzo 2011.