

Hardware libre en el aula: Una experiencia de capacitación en el uso de recursos educativos abiertos en escuelas técnicas en Tucumán, Argentina

Free hardware in the classroom: an experience of training in the use of open educational resources in technical schools in Tucumán, Argentina.

Valentín Basel

CONICET - Universidad Nacional de Córdoba, Argentina

E-mail: valentinbasel@gmail.com

Resumen

La robótica, como muchas de las áreas del conocimiento social, al ser utilizada en el contexto educativo tiene múltiples aplicaciones. De esa forma puede ser aprovechada en el aula como herramienta transversal para el aprendizaje de conceptos de matemática, física o lenguajes de programación. Pero, ¿qué es lo que hace de ella algo relevante en el contexto educativo contemporáneo? Para dar a cuenta de ello, en el presente escrito analizaremos algunas experiencias vinculadas al taller “Diseñando tu propio robot”, realizado entre 2014-2015 por el CIIDEPT (Centro de Innovación e Investigación para el Desarrollo Educativo, Productivo y Tecnológico) y destinado a capacitar a docentes y alumnos en el diseño y fabricación casera de robots para la enseñanza de programación, usando hardware de especificaciones abiertas y software libre como recurso educativo abierto desarrollados por el proyecto ICARO.

Palabras clave: robótica educativa; software libre; hardware libre; GNU/Linux; Conectar igualdad

Abstract

Robotics, like many areas of social knowledge, when used in the educational context has multiple applications. In this way, it can be used in the classroom as a transversal tool for learning mathematical concepts, physics or programming languages. But what makes it relevant in the contemporary educational context? To give an account of this, in this paper we will analyze some experiences related to the workshop “Designing your own robot”, held between 2014-2015 by CIIDEPT (Center for Innovation and Research for Educational Development, Productive and Technological) and designed to train teachers and students in the design and home manufacture of robots for teaching programming, using open specifications hardware and free software as an open educational resource developed by the project ICARO.

Keywords: educational robotics; free software; free hardware; GNU/Linux; Conectar igualdad

Fecha de recepción: Octubre 2019 • Aceptado: Diciembre 2019

BASEL, V. (2020). Hardware libre en el aula: Una experiencia de capacitación en el uso de recursos educativos abiertos en escuelas técnicas en Tucumán, Argentina *Virtualidad, Educación y Ciencia*, 20 (11), pp. 202-211.

Introducción

De acuerdo a su carácter multidisciplinario, la robótica permite abordar distintas etapas del conocimiento por parte de los alumnos y estudiantes; por su particularidad, además de poder ser aplicada en una gran variedad de temáticas, y su “espectacularidad” (entendida como la capacidad de generar algún tipo de asombro en la población estudiantil), ayuda a los docentes en la tarea de impartir la currícula planteada en el curso. Sin embargo, la gran dificultad técnica y la cantidad de conocimientos específicos (como electrónica, mecánica y ciencias computacionales), además de un costo generalmente elevado de los kits que se consiguen en el mercado, hace que la enseñanza de robótica en las escuelas este relegada generalmente a colegios que pueden financiar los costes de capacitación y adquisición de estos kits comerciales.

El proyecto ICARO busca desarrollar una solución técnica basada en hardware de especificaciones abiertas y software libre para facilitar una serie de demandas pedagógicas en la labor del docente a la hora de abordar contenidos técnicos complejos que implica el abordaje de una disciplina como la robótica y, además, abaratar costos al ser ICARO un proyecto pensado para poder implementarse en pequeña escala, sin equipamiento industrial y con la idea de que docentes y alumnos fabriquen (soldar los componentes) el hardware ICARO. Desde esta perspectiva, el CIIDEPT (Centro de Innovación e Investigación para el Desarrollo Educativo, Productivo y Tecnológico), viene implementando desde el año 2014 el taller para docentes y alumnos “Diseñando tu propio robot“, basado en el proyecto ICARO y aprovechando recursos educativos abiertos (REA/OER) pre instalados en el sistema operativo HUAYRA GNU/Linux del Programa Conectar Igualdad (PCI), destinado a alumnos de Nivel Secundario, Escuelas Técnicas con Especialidad en Electromecánica y Electrónica y Escuelas Secundarias con Orientación en Informática.

El construccionismo y la robótica educativa

Los robots usados como herramientas didácticas para enseñar programación, tienen su origen en el trabajo elaborado por Seymour Papert y su equipo en el MIT (Massachusetts Institute of Technology) durante la década de los setenta y ochenta, principalmente con el desarrollo del entorno LOGO, un lenguaje formal que permitía a los estudiantes aprender a programar una “tortuga robótica”, la cual en un principio fue un sistema robot físico para luego ser una representación virtual en la pantalla de las micro computadoras de la época.

La creación del lenguaje LOGO vino a reforzar la concepción del aprendizaje (apoyada en el uso de entornos) postulada por Papert, bautizada como construccionismo (Papert, 1980). Esta perspectiva, considerada por el mismo Papert como una evolución del constructivismo piagetiano, considera que las actividades de confección o construcción de artefactos son facilitadores del aprendizaje. Los sujetos, al estar involucrados activamente mientras aprenden, “construyen” también sus propias estructuras de conocimiento de manera paralela a la construcción de un objeto (ya sea físico o conceptual). Esta teoría también afirma que los sujetos aprenden mejor cuando construyen objetos que sean interesantes para ellos, al tiempo que los elementos fabricados ofrecen la posibilidad de hacer más concretos conceptos abstractos o teóricos haciendo que sean más sencillos de comprender. Por lo tanto, se plantea la necesidad de la acción como parte integral del proceso de aprendizaje. Seymour Papert (1987) parte del supuesto que el conocimiento lo construye el propio sujeto que aprende a

través de la acción, entonces el proceso de diseñar y construir un producto (en este caso por ejemplo un robot) funcionaria como facilitador del aprendizaje. Desde esa perspectiva podemos considerar que los ambientes de estudio generados por la robótica educativa, permiten posicionar al estudiante en un rol activo y protagónico de su propio proceso de aprendizaje.

En cuanto al uso de robótica como recurso educativo, Sánchez y Guzmán (2012) dicen que por su carácter multidisciplinario, la robótica es una herramienta interesante para el uso como recurso facilitador del aprendizaje y el desarrollo de competencias generales, dado que permite trabajar transversalmente múltiples disciplinas, y sirve como motivador para que los estudiantes lleven a cabo proyectos donde puedan experimentar y desarrollar sus actitudes cognitivas, tal como dice Pittí (et al., 2010), la robótica es una herramienta constructorista.

Hardware y software libre

El software libre es un movimiento que comenzó en el año 1983 cuando Richard Stallman (2007) anunció el proyecto GNU en contra de la posición a la aparición de monopolios artificiales en el desarrollo de software (Busaniche, et. al, 2010). Se podría decir que la meta del movimiento fue dar libertad a los usuarios de programas de computadoras reemplazando el software con términos de licencias restrictivas (software privativo) por una alternativa libre.

La comunidad de desarrolladores de software libre plantea que el software, para ser considerado libre, debe poder ser copiado, estudiado, distribuido y modificado libremente por cualquier persona o comunidad. En este sentido, se vuelve de vital importancia contar con el código fuente (y no sólo el código máquina o binario) de los programas para poder estudiarlos y modificarlos; además de tener una licencia (la licencia GNU/GPL) que proteja el derecho de autor y permita que ese código pueda ser distribuido sin el peligro de que sea apropiado por alguien más.

Teniendo en cuenta los conceptos mencionados, podemos explicar algunas de las definiciones que se utilizan para que un software sea considerado como software libre. Richard Stallman define cuatro “libertades” que tiene que tener el software para considerarlo libre:

- libertad 0: La libertad de usar el programa, con cualquier propósito (Uso)
- libertad 1: La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (Estudio).
- libertad 2: La libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (Distribución).
- libertad 3: La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (Mejora).

Un programa es software libre si otorga a los usuarios todas estas libertades de manera adecuada. Por lo tanto, el hecho de contar con el código fuente es una condición necesaria para poder ejercer las cuatro libertades que plantea la FSF (Free Software Foundation). Lo dicho hasta aquí supone que todos los programas desarrollados y distribuidos bajo licencias libres (por ej. -la licencia GNU/GPL V3) tienen que ser distribuidos con los archivos de código fuente además de los archivos ejecutables (en el caso de programas compilados).

El movimiento de hardware libre (o hardware de especificaciones abiertas) busca llevar el concepto del software libre (la libertad de usar, estudiar, distribuir o mejorar el software) al diseño de componentes físicos, especificando una licencia que permite distribuir planos y código fuente de desarrollos de PCBs (Printed Circuit Board por sus siglas en inglés) y hardware electrónico. Asimismo, se considera que un diseño de circuito (esquemático, diseño de PCB y archivos GERBER) debe ser desarrollado con software libre y usando formatos abiertos.

De acuerdo a la declaración de principios de la Open Source Hardware Association, podemos decir que:

Hardware de Fuentes Abiertas (OSHW en inglés) es aquel hardware cuyo diseño se hace disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender, tanto el original como otros objetos basados en ese diseño. Las fuentes del hardware (entendidas como los ficheros fuente) habrán de estar disponibles en un formato apropiado para poder realizar modificaciones sobre ellas (OSHW, s. f.).

Es importante notar como la OSHW plantea que para que un hardware sea considerado de estándares abiertos, no solo deben distribuirse sus planos y esquemáticos con un formato libre. Además, el diseño debería contemplar la posibilidad de fabricar ese mismo hardware por parte de los usuarios. El hardware de fuentes abiertas da la oportunidad de controlar la tecnología y al mismo tiempo permite compartir conocimientos.

Robótica educativa de hardware libre

Uno de los mayores problemas de la implementación de la robótica en el aula es la gran carga de contenido técnico que debe afrontar el docente para poder trabajar con su currícula. En ese sentido, varios fabricantes diseñaron kits para el uso de robótica con fines educativos; sin embargo, estos kits de robótica suelen ser extremadamente caros y por lo tanto restrictivos para su utilización masiva por parte de los docentes. El coste operativo de implementar un kit de robótica comercial puede ser prohibitivo para colegios de pocos ingresos, pero también el costo de mantenimiento (reparación y reemplazo de piezas defectuosas, actualizaciones, etc.) termina siendo un factor clave a la hora de usar los robots como herramientas pedagógicas, a causa del “peligro” de que los alumnos los rompan y reponerlos salga tan caro como comprar un kit nuevo.

Desde esa perspectiva, la robótica educativa con software y hardware libre resulta una opción viable para la aplicación en el proceso de aprendizaje, porque permite a las escuelas adaptar la tecnología a las necesidades específicas de la institución, posibilitando el reciclaje de componentes que se encuentren en el colegio (aportados por la comunidad escolar), y así ahorrar costos. A su vez, por la gran expectativa que genera en los alumnos, y al ser de código fuente libre, ayuda a romper barreras culturales y políticas que pueden ir en detrimento de la calidad educativa, barreras como el alto costo de adquisición de los elementos, licencias privativas para el software de control o la falta de documentación específica para comunidades minoritarias (traducciones a idiomas que no son viables comercialmente, por ejemplo). La robótica educativa con software y hardware libre permite aprovechar las ventajas que ofrece la filosofía de desarrollo que existe en las comunidades de software libre, donde “mil ojos ven más que uno” (Raymond, 1998), involucrando a la comunidad escolar en su conjunto y donde las soluciones aportadas por el grupo podrán ser utilizadas en otros colegios

y viceversa, generando situaciones donde todos los involucrados salen favorecidos (Olivencia & Martínez, 2015).

La plataforma electrónica propuesta para uso en el proyecto ICARO, se basa en el micro controlador de la firma MICROCHIP, concretamente el modelo 18F4550. La elección de este microcontrolador se da principalmente por la necesidad de contar con un integrado en formato THT (Through-Hole Technology) que permita soldar el encapsulado (en formato DIP) con equipos hogareños (soldador de estaño tipo lápiz), además es un microcontrolador que cuenta con integración USB 2.0 a nivel de hardware. Los microcontroladores 18f4550 cuentan con las siguientes características:

- Módulo USB 2.0. Soporta Low speed 1.5Mb/s y full speed 12Mb/s.
- 35 pines I/O disponibles
- Memoria de programa flash de 32 kB
- RAM de 2048 Bytes
- EEPROM de datos de 256 Bytes
- Oscilador externo de dos modos hasta 48 MHz
- ADC de 10 bits y 13 canales
- Voltaje de operación 4.2V a 5.5V
- 4 Timer (desde Timer0 a Timer3). Uno de 8 bits y 3 de 16 bits
- 2 módulos de captura/comparación/PWM EUSART, SPP, SPI, I²C.
- 20 fuentes de interrupciones (3 externas)
- Soporta 100,000 ciclos de borrado/escritura en memoria flash
- Soporta 1,000,000 ciclos de borrado/escritura en memoria EEPROM
- Encapsulado DIP de 40 pines

El hardware del proyecto ICARO está diseñado para ser construido en el transcurso de una jornada de capacitación (en una clase intensiva o varias, dependiendo del enfoque de trabajo) donde la idea principal es que alumnos y docentes que participen, vayan soldando cada componente en el PCB (Printed Circuit Board) mientras se les explica el funcionamiento de la placa. De este modo, lo que se busca es que los alumnos puedan entender el funcionamiento del hardware a nivel electrónico.

Cuando se diseñó el PCB ICARO (actualmente en la versión NP07) se tomó en cuenta que la placa tuviera las siguientes características:

- Simple Faz (una sola capa de cobre).
- Pistas relativamente gruesas (20 Mils).
- Componentes de tipo THT.
- Facilidad para producir el PCB en pequeña escala, usando técnicas caseras de fabricación.

- Componentes simples de conseguir en el mercado local (o reciclados).
- Puente H para manejo de motores de corriente continua.
- ULN2803 para manejos de salidas digitales (hasta 12V x 1A).

Para evitar usar tecnología In-Circuit Serial Programming (ICSP) y necesitar de un programador de Pics, se optó por aprovechar el Bootloader diseñado para el proyecto Pinguino-Boards, que permite cargar firmwares escritos con el compilador libre Small Device C Compiler (SDCC), de esa forma no es necesario un hardware especial para cargar un firmware en el microcontrolador, asimismo al ser SDCC un proyecto de software libre, todo el software para generación y carga de firmware funciona en sistemas operativos GNU/Linux, permitiendo que el software del proyecto ICARO pueda formar parte del sistema operativo HUAYRA GNU/Linux.

Por otra parte, para generar el código fuente del microcontrolador, se usa el software ICARO-BLOQUES, una interface gráfica (GUI) escrita en python 2.7 usando las librerías GTK+2 y CAIRO, que permite trabajar mediante bloques gráficos que representan instrucciones de código C, con la idea de facilitar una primera aproximación al desarrollo de código fuente en lenguaje C usando el compilador SDCC.

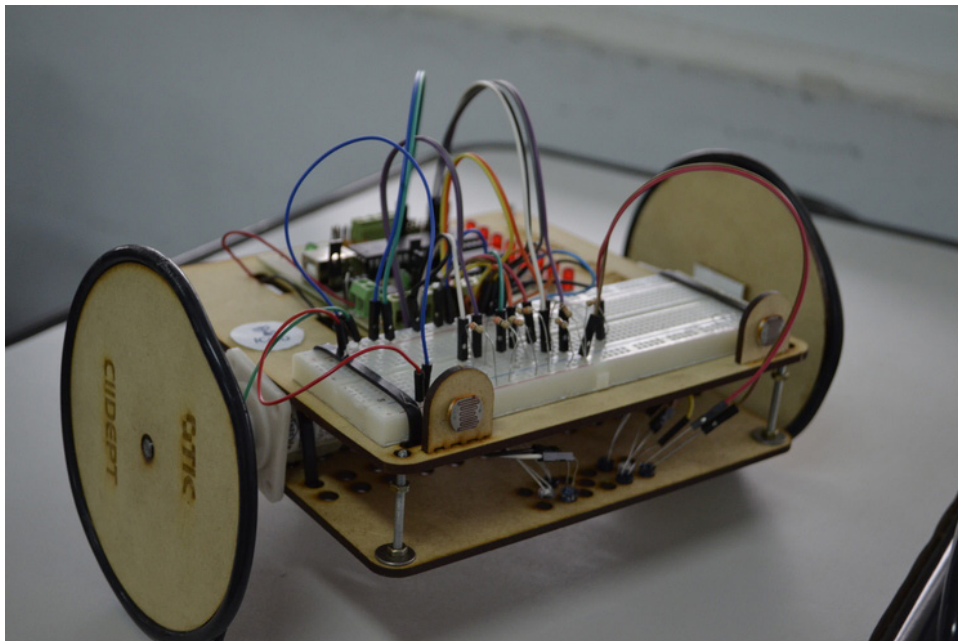
Conectar igualdad y el sistema operativo Huayra GNU/Linux

La historia de las políticas de inclusión digital educativa en la región cuenta con tres décadas y fue siguiendo los avances tecnológicos. El laboratorio o sala de informática, en donde las actividades con el computador adquirían un carácter disciplinar, de uso compartido y supervisado para la capacitación en herramientas, constituye una primer etapa. Más allá de sus limitaciones, es todavía recordada por algunos docentes en tanto espacio de delegación y control. Se da paso luego a una nueva relación, en donde las netbooks salen del gabinete para incorporarse a las actividades cotidianas del aula, asumiendo un carácter transversal. El modelo 1 a 1 viene así a tensionar ciertas rutinas y dinámicas tradicionales, donde el docente se ubica como fuente principal de conocimiento. Junto con ello, los jóvenes adquieren un mayor protagonismo, haciéndose más responsables de sus procesos de aprendizaje (Morales Capilla, Trujillo Torres, & Raso Sánchez, 2015).

En América Latina, Uruguay fue el país pionero en implementar el modelo 1 a 1, mediante el Plan Ceibal lanzado en 2007. Los equipos fueron desarrollados en el marco del programa One Laptop per child (OLPC). En años subsiguientes se fueron sumando iniciativas particulares en cada país, las cuales difieren en su población objetivo -inicial, primaria o secundaria/ estudiantil o docente- así como en su alcance -focalizadas o universales- (Uruguay, Una computadora para cada niño, 2009).

El caso argentino se definió por una política universal hacia la escuela secundaria. El PCI implicó un gran despliegue inicial de recursos económicos, técnicos, de infraestructura y capacitación, además de la creación de contenidos y propuestas educativas. Se trató de una iniciativa conjunta entre la Jefatura de Gabinete de Ministros, el Ministerio de Educación, el Ministerio de Planificación Federal y la Administración Nacional de la Seguridad Social (Consejo de Educación, 2010).

Imagen 1: Robot diseñado por el CIIDEPT para cursos iniciales de programación



Fuente: Elaboración propia

Los objetivos del programa vinculan la política educativa con la social, apuntando a garantizar el acceso a las tecnologías, superar brechas y reducir desigualdades. Tienen además la pretensión de repercutir sobre la totalidad de los actores que conforman la comunidad escolar: directivos, docentes, alumnos, familias y referentes tecnológicos.

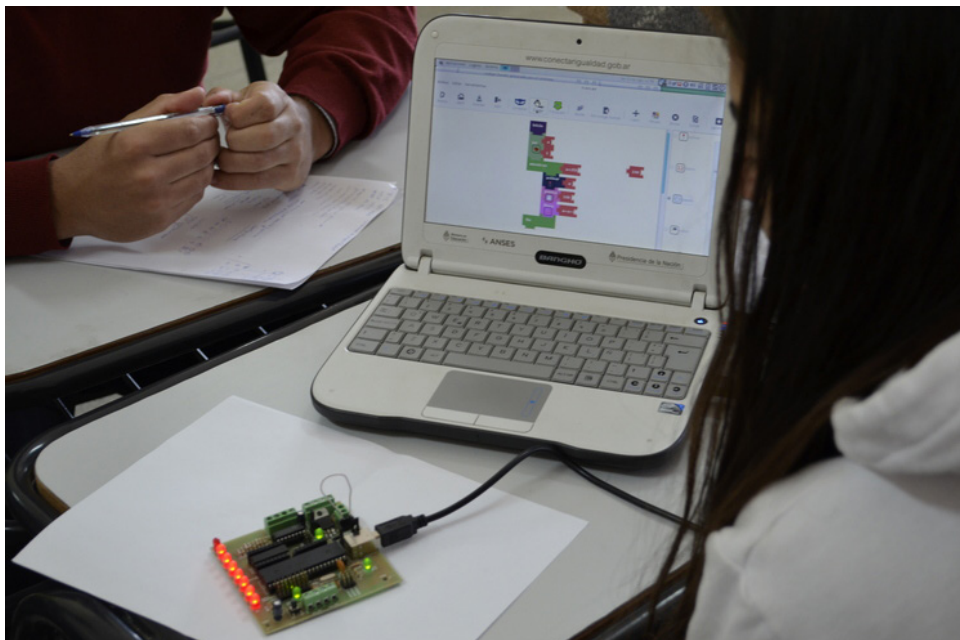
Programa educativo “Diseñando tu propio robot”

En el año 2014, el Ministerio de Educación a través de su Coordinación Provincial TIC en conjunto con el CIIDEPT, empezaron a impartir cursos de capacitación para docentes y alumnos de colegios técnicos y agro-técnicos pertenecientes a la provincia de Tucumán. Desde esa perspectiva, el CIIDEPT se encargó de capacitar en jornadas de 5 días a grupos compuestos por un docente y 5 alumnos, pertenecientes a cada escuela técnica de las distintas ciudades de la provincia de Tucumán. El objetivo de los talleres es realizar proyectos en los que, mediante el diseño, la construcción y la programación de robots, los estudiantes puedan visualizar, explorar y comprobar conceptos de razonamiento de las áreas de conocimiento involucradas (como la mecánica, la electrónica, la inteligencia artificial y la ingeniería de control, entre otros campos) y además, formular y experimentar alternativas para solucionar problemas o realizar tareas de automatización (Benotti, Gómez, & Martínez, 2017).

El formato del curso de capacitación consta de cinco clases presenciales, donde un “facilitador pedagógico” del CIIDEPT trabaja con cinco grupos de cuatro alumnos y un docente (en total veinticinco personas por cohorte). Las cinco clases presenciales están preparadas para que los alumnos y docentes puedan fabricar la placa ICARO; en el primer encuentro, se enseñan los conceptos básicos para manejar un microcontrolador (sistema binario, definición de sensores digitales y analógicos) y, a continuación, en la siguiente clase, ya poder armar la placa. Cada colegio participante arma una placa

entregada por el CIIDEPT y crea firmware específico para el proyecto de final de curso usando el software de control ICARO-BLOQUES que viene instalado en la distribución HUAYRA GNU/Linux, el sistema operativo de las netbooks del PCI. Al final de la jornada de capacitación, los alumnos y docentes, presentan un trabajo final que consiste en un sistema de automatización, domótica o un robot sencillo, que fabrican con componentes reciclados.

Imagen 2: Hardware ICARO fabricado por docentes y estudiantes.



Fuente: Elaboración propia

Aprovechando las capacidades del microcontrolador 18f4550, los grupos participante armaron robots y sistemas de automatización, como equipos de riego automáticos, control de luces, un ascensor, o hasta un brazo robot. Debido a que desde el proyecto ICARO se busca “eliminar cajas negras”, desde el principio se motivó a los participantes de los cursos impartidos por el CIIDEPT a que explorarán el código fuente y los planos del hardware propuesto, por ejemplo, usando fuentes de alimentación personalizadas o etapas de potencias específicas de mayor capacidad de manejo de corriente. El objetivo principal fue capacitar a los docentes en el uso de las netbooks entregadas por el PCI y actualizar las currículas de los colegios técnicos para aprovechar las ventajas de HUAYRA GNU/Linux, sobre todo en aquellos con orientación en electrónica y electromecánica participantes del programa de capacitación.

Consideraciones finales

Implementar un proyecto de capacitación docente a nivel provincial en el uso de Tecnologías REA/OER (González & Hernández II, 2013) para la enseñanza de robótica implicó una serie de compromisos técnicos y logísticos para el equipo del CIIDEPT, como conseguir un fabricante de PCBs local que pudiera abastecer la demanda de placas que se necesitaron durante los dos años del curso de formación (aproximadamente se fabricaron 200 placas de entrenamiento NP06 y luego otras

200 de la versión NP07) más un proveedor para los componentes electrónicos (microcontrolador y electrónica discreta). También, se aprovechó la ventaja de tener unificado el software presente en las netbooks entregadas por el PCI, que al tener instalada (de fábrica) la distribución HUAYRA GNU/Linux que incluye el software ICARO-BLOQUES listo para trabajar, permitió ahorrar tiempo y esfuerzo a los facilitadores pedagógicos que impartían los cursos, proporcionando un marco estándar de enseñanza que permitió que el tiempo de clase se dedicara al trabajo de los conceptos teóricos implícitos en el desarrollo de un robots, y no tanto a tareas mecánicas de instalación o actualización de software (Martínez, Marotias, & Amado, 2013). Asimismo, se aprovechó el hecho de que gran parte de la comunidad de desarrollo de ICARO vive en Argentina lo cual posibilita a los docentes interactuar y hacer recomendaciones de diseño para las modificaciones en el software y el hardware.

Como resultado de estos cursos de capacitación, participaron 23 escuelas en el año 2014, 35 escuelas en el año 2015 y en total recibieron capacitación 299 alumnos y 75 docentes de las escuelas técnicas y agrotécnicas de la provincia de Tucumán. Gracias al programa de capacitación, los 75 docentes de los colegios técnicos pudieron actualizar sus espacios curriculares al uso de software y hardware libre; además, dos docentes (ingenieros electrónicos) aportaron recomendaciones para modificar el hardware propuesto (placas NP06 en ese momento) y del resultado se creó la versión 07 del hardware.

Las experiencias formadas por docentes y alumnos en el marco del curso de capacitación permitieron demostrar que es posible implementar un plan sistemático para el uso de recursos educativos abiertos (Hepp K et al., 2013), posibilitando que los docentes no queden “atados” a software privativo o kits comerciales que muchas veces los colegios no pueden financiar.

Sin embargo, cabe destacar la complejidad que conlleva una migración de un esquema de trabajo basado en software propietario a otro basado en software libre. La escasa documentación específica sobre educación, la falta de ofertas académicas que permitan a los docentes actualizar sus conocimientos en el área, y una resistencia al cambio por parte de los mismos, son algunos de los factores que hay que tener en cuenta para lograr que una experiencia como la aplicada por el CIIDEPT pueda ser efectiva en su cometido.

Si bien las netbooks entregadas por el PCI traen por defecto instalado la distribución HUAYRA GNU/Linux y todos los paquetes listos para trabajar, es necesario tener instancias de formación como las planteados desde el CIIDEPT, que permitan generar una mejor articulación entre el trabajo de los docentes y los desarrollos creados por las comunidades de software libre, en pos de lograr una mejor adopción de los recursos educativos abiertos.

Referencias bibliográficas

- BENOTTI, L., GÓMEZ, M. J., y MARTÍNEZ, C. (2017). UNC++ Duino: A kit for learning to program robots in Python and C++ starting from blocks. En *Robotics in Education* (pp. 181–192). Springer.
- BUSANICHE, B., y otros. (2010). *Monopolios artificiales sobre bienes intangibles*, 2007.
- de Educación, C. F. (2010). *Las políticas de inclusión digital educativa el Programa Conectar Igualdad*. Consejo Federal de Educación Argentino, Resolución CFE, (123710). Recuperado de <http://skat.ihmc.us/rid=1M4M20FDJ-VQ84YQ-3RP3/Inclusion%20digital.pdf>

- GONZÁLEZ, Z., y HERNÁNDEZ II, G. M. (2013). Recursos educativos abiertos. Recuperado de <http://www.medigraphic.com/pdfs/educacion/cem-2013/cem133p.pdf>
- HEPP K, P.; MERINO D, M. E.; BARRIGA J, M. V. y HUIRCAPÁN A, A. (2013). Tecnología robótica en contextos escolares vulnerables con estudiantes de la etnia Mapuche. *Estudios pedagógicos (Valdivia)*, 39(ESPECIAL), 75-84. <https://doi.org/10.4067/S0718-07052013000300006>
- MARTÍNEZ, S. L., MAROTIAS, A., y AMADO, S. (2013). Inclusión digital en la educación pública argentina. El Programa Conectar Igualdad. *Revista Educación y Pedagogía*, 24(62), 205–218.
- MORALES CAPILLA, M., TRUJILLO TORRES, J. M., y RASO SÁNCHEZ, F. (2015). Percepciones acerca de la integración de las TIC en el proceso de enseñanza-aprendizaje de la universidad. *Pixel-Bit. Revista de Medios y Educación*, (46). Recuperado de <http://www.redalyc.org/html/368/36832959012/>
- OLIVENCIA, J. J. L., y MARTÍNEZ, N. M. M. (2015). Recursos y estrategias educativas basadas en el uso de hardware de bajo coste y software libre: Una perspectiva pedagógica intercultural. *Revista científica electrónica de Educación y Comunicación en la Sociedad del Conocimiento*, 1(15). Recuperado de <http://eticanet.org/revista/index.php/eticanet/article/view/63>
- OPEN SOURCE HARDWARE ASSOCIATION, (s. f.). Declaración de Principios 1.0. Recuperado de <https://www.oshwa.org/definition/spanish/> [02/10/2019]
- PAPERT, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- PITTÍ PATIÑO, K.; CURTO DIEGO, B. y MORENO RODILLA, V. (2010). Experiencias constructoras con robótica educativa en el Centro Internacional de Tecnologías Avanzadas. *Teoría de la Educación*, 11(1), 26.
- RAYMOND, E. S. (1998). La catedral y el bazar. *The Linux Logic Home Page*, 12. Recuperado de <http://www.athanazio.com/downloads/livros/catedral-ou-bazar.pdf>
- BRAVO SÁNCHEZ, F. Á. y FORERO GUZMÁN, A. (2012). La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales. *Education in the knowledge society (EKS)*, 13(2), 120-136.
- SEYMOUR PAPERT. (1987). *Desafío de la mente*. Buenos Aires, Argentina. Editorial Galápagos
- STALLMAN, R. M. (2007). Software libre para una sociedad libre. *Traficantes de Sueños*. Recuperado de https://www.gnu.org/philosophy/fsfs/free_software.es.pdf [02/10/2019]
- Uruguay, una computadora para cada niño: Plan Ceibal: los ojos del mundo en el primer modelo OLPC a escala nacional. (2009). Prentice Hall.