

Sensores Tortuga 2.0: Cómo el hardware y software abiertos pueden empoderar a las comunidades de aprendizaje Turtle

Sensors 2.0: How open hardware and software empower learning communities

Guzmán Trinidad

Liceo Solymar N°1 (ANEP-CES). Uruguay.
guzman.trinidad@gmail.com

Andrés Aguirre

Universidad de la República. Uruguay.
aaguirre@fing.edu.uy

Alan Aguiar,

Universidad de la República. Uruguay.
aaguiar@fing.edu.uy

Tony Forster

Sugar Labs
forster@ozonline.com.au

Walter Bender

MIT, E.E.U.U.
walter@mit.edu

Facundo Benavides

Universidad de la República. Uruguay.
fbenavid@fing.edu.uy

Federico Andrade

Universidad de la República. Uruguay.
fandrade@fing.edu.uy

Resumen

En este trabajo describimos un enfoque abierto en el uso de la computación con el objetivo de explorar la ciencia y la ingeniería que no está predeterminado o prescrito; por el contrario, su uso está destinado a ser emergente, donde el cambio es una consecuencia de que los maestros, los niños y sus padres aprenden juntos, descubriendo nuevas posibilidades, y comparten esos descubrimientos. Estos diseños abiertos, tanto en hardware como en software, empoderan a los estudiantes y sus comunidades. Ellos son libres de dar forma a las herramientas que utilizan para experimentar y para participar en proyectos que son auténticos en su diseño. Postulamos que cuando a los estudiantes y a sus comunidades se les da el control son más eficaces en sus aprendizajes.

Palabras Clave

Construccionismo, Programación, Logo, Robótica, Sensores, Hardware Abierto, Software Libre

Abstract

We described an open approach to the use of the computing for exploring science and engineering that is not predetermined or prescribed. Rather, its use is intended to be emergent, where change is a byproduct of teachers, children, and their parents learning together, discovering new possibilities, and sharing those discoveries. These open designs—both hardware and software—empower students and the communities in which they live. They are free to shape the tools that they use to experiment and to engage in

projects that are authentic in their design. When students and communities are given control over their learning we postulate that they are more effective learners.

Keywords

Constructionism, Programming, Logo, Robotics, Sensors, Open Hardware, Software Libre

1. Introducción

Mientras estamos escribiendo este artículo, la sonda espacial New Horizons está pasando por la órbita de Plutón un viaje de alto riesgo de nueve años que tiene el potencial de aumentar nuestro conocimiento sobre nuestro sistema planetario y dar lugar a nuevos conocimientos sobre la naturaleza de nuestro universo. El camino a Plutón era cualquier cosa menos una línea recta. La NASA utiliza múltiples impulsos gravitacionales para acelerar la nave espacial y corregir su camino. Por ejemplo, New Horizons no se acercó a Júpiter para visitarlo, sino para utilizar la velocidad del planeta alrededor del sol para aumentar la velocidad de la sonda (relativa al sol). En este artículo describimos un conjunto de plataformas de hardware y software abierto que intentamos actúen como aceleradores intelectuales. Nuestro objetivo no es simplemente sumergir a los estudiantes en el aprendizaje de un lenguaje particular o herramienta o conseguir un resultado experimental particular, sino que lo usamos como una forma de estimularlo para perseguir sus ambiciones y objetivos propios. No sabemos el grado de éxito que hemos logrado en conseguir este objetivo. Citamos casos individuales de estudio pero no tenemos información longitudinal que nos permita alcanzar una conclusión definitiva. Más allá del éxito de la misión New Horizons como misión científica, ya ha logrado aprovechar un rasgo muy humano: el deseo de explorar lo desconocido y llegar a nuevas alturas; es de esperar que el entusiasmo de la próxima generación de niños los empuje a llegar aún más lejos. El trabajo descrito en este documento, en su humilde forma, se nutre de la misma característica humana: entusiasmar a la próxima generación de niños para sobrepasar sus límites.

1.1 Herramientas abiertas

La computadoras de bajo costo como la computadora portátil XO de OLPC (XO, 2008), Raspberry Pi (Raspberry Pi, s.f.), y BeagleBone (BEAGLEBONE, s.f.) se encuentran entre un número cada vez mayor de sistemas de hardware diseñados con énfasis en la apertura y la apropiación final del usuario. Por ejemplo, la computadora portátil XO fue diseñada para facilitar la elaboración de extensiones sencillas por el usuario, tales como sensores de bajo costo que se conectan al conector de entrada del micrófono externo o las placas de entrada y salida (E/S) externas de bajo costo que se conectan al puerto USB. La computadora Raspberry PI expone puertos de entrada/salida de propósito general (GPIO) al igual que la BeagleBone, la cual provee un conjunto amplio de puertos que pueden ser utilizados con varios sensores y actuadores. Las placas de entrada salida de bajo costo como Arduino (Arduino, s.f.), ExpEyes (ExpEyes, s.f.), USB4Butiá (USB4Butiá, s.f.), entre otras, permiten que las computadoras convencionales tengan la capacidad adicional de controlar actuadores y leer sensores. Estos dispositivos permiten realizar una serie de experimentos científicos a un costo insignificante, más allá del computador en sí: por ejemplo, una computadora portátil puede funcionar como un registrador de datos a través de una serie de sensores que se puede construir con tecnología obtenida de desechos (de disponibilidad universal) a un costo insignificante.

La plataforma de software para el aprendizaje Sugar (Sugar, s.f.), originalmente diseñada para la computadora portátil XO, también pone énfasis en ser abierta. Es Software Libre licenciado bajo la Licencia Pública General (GPL) (GNU Licences, s.f.) la cual permite al usuario final realizar modificaciones sin restricciones. Sugar da un paso más hacia adelante respecto a la mayoría del Software Libre, en el hecho de que provee propiedades para el usuario final que le permiten realizar duplicación y modificación. Los usuarios no solo están amparados en una licencia de software para hacer cambios, se les dan los medios y se les anima a hacer cambios.

Postulamos que la combinación de hardware y software de diseño abierto permite la apropiación tecnológica tanto a los estudiantes como a las comunidades donde viven.

Los estudiantes y sus maestros, objeto de este estudio, han desarrollado herramientas y explorado problemas siendo independientes de la tecnología comercial o preestablecida para sus experimentos científicos o proyectos de diseño. Nuestra pregunta de investigación en curso es: Cuando se les da a los estudiantes y sus comunidades el control sobre su aprendizaje, ¿son estudiantes más eficaces?

Existe una larga tradición que apoya este postulado. En resumen, existe una amplia evidencia de que la motivación intrínseca es más potente y más duradera que la motivación extrínseca (Lepper et al., 1997). Ejemplos concretos son la clase 6.270 MIT, desarrollada por Randy Sargent y Fred Martin, donde los estudiantes construyen y programan robots (Martin, 2000), un formato que ha sido ampliamente replicado como por ejemplo en el evento First Robotics (First, s.f.). Otros ejemplos incluyen el MIT 8.01X, un curso de física de primer año. 8.01X introduce a los estudiantes a la mecánica clásica mediante "una práctica en el enfoque, y se acerca a la mecánica a través de experimentos para llevar a casa" (Física 1, 2002). También en Uruguay, desde el año 2004 la Facultad de Ingeniería de la Universidad de la República está trabajando con este enfoque con el evento abierto sumo.uy (sumo, s.f.) con el objetivo de motivar a los estudiantes de diferentes edades al estudio de las ciencias en un contexto cooperativo. Los estudiantes a menudo faltan a clases, pero logran construir sus propios aparatos y realizar los experimentos. En el mismo sentido, los estudiantes con peores calificaciones han sido más motivados para trabajar y cooperar con este tipo de propuestas (Enlaces Fing, 2012).

Un beneficio adicional de los proyectos abiertos es que pueden ser generadores de comunidades descentralizadas y multinacionales que trascienden las fronteras geográficas. Este artículo ha sido escrito por autores en los Estados Unidos, Uruguay y Australia. Describe sus proyectos educativos independientes pero cooperativos.

2. Aprender haciendo

En el aula convencional, el profesor posee "objetos" de conocimiento (por ejemplo, datos sobre números y adición, los sustantivos y verbos de las matemáticas) que se entregan a los estudiantes. Ser educado es acumular estos objetos, es decir, la educación es algo que se da al estudiante. Hay alternativas a recibir pasivamente un cuerpo de conocimiento: nos centramos en ayudar a los niños a apropiarse del conocimiento al ponerlo a trabajar apoyado en problemas que son significativos para ellos. Como consecuencia, el aprendizaje es algo hecho por el estudiante. En este tipo de aprendizaje, un niño no sabrá simplemente el nombre de una cosa y cómo podría ser utilizado en un contexto en sentido estricto; ellos entenderán su utilidad y sus limitaciones. Puesto de forma más concisa: "Aprendes cosas haciendo, así que si quieres

aprender más, quieres hacer más." Esto no es una idea nueva: la mayor parte de las bases fueron establecidas por John Dewey hace más de un siglo en su obra sobre la educación experimental y aprendizaje práctico que hacían hincapié en la participación activa del alumno. (Dewey, 1938).

A comienzos del siglo XX en Uruguay, el Dr. Pedro Figari (como Director de la Escuela de Artes y Oficios entre 1915 y 1917) aplicó los mismos conceptos profundamente innovadores en la reformulación de la educación de dicha institución. Las estrechas similitudes entre las visiones de ambos pensadores fueron claramente indicadas por el Dr. Arturo Ardao en el Prólogo del libro *Educación y Arte* (Figari, 1965), escrito originalmente en 1919: "Merece, sin duda, un estudio detenido este paralelismo de las ideas pedagógicas de Figari con las de Dewey, cuya difusión en el mundo hispánico comenzó recién en la década del 20." (p. XVIII).

Lamentablemente el país no estaba preparado para asimilar estas ideas de vanguardia y el proyecto no prosperó: "A Figari le duele que...Batlle," (José Batlle y Ordóñez, Presidente del Uruguay 1903-1907 y 1911-1915) no advierte que su idea es revolucionaria... El 14 de Abril de 1917... pone punto final a su experiencia. Está molesto, amargado por las resistencias que encuentra y por una actitud que considera equívoca del gobierno." (Sanguinetti, 2002)

Una visión más contemporánea del potencial del aprender haciendo, se puede encontrar en los artículos que Seymour Papert escribió en la década de 1970 sobre la enseñanza de la matemática en los niños. Papert argumentó que no era efectivo enseñar matemáticas a los niños pequeños en abstracto, divorciado de otras actividades que sean significativas para ellos. Papert no estaba buscando mejorar los métodos tradicionales de enseñanza de álgebra y geometría. En cambio, él escribió:

Me pregunto si se puede identificar y enseñar (o fomentar el crecimiento de) algo que no sea el álgebra o la geometría, que, una vez aprendido, hará que sea fácil de aprender álgebra o geometría. . . En nuestro ideal de una escuela laboratorio de matemática el ordenador se utiliza como un medio para controlar procesos físicos con el fin de alcanzar los objetivos definidos. . . por ejemplo, como parte del sistema de piloto automático para volar aviones a escala, o como el 'sistema nervioso' de un modelo animal con reflejos de balanceo, la capacidad de caminar, la capacidad visual simple y así sucesivamente. Para alcanzar estos objetivos se necesitan principios matemáticos; por el contrario, en este contexto los principios matemáticos se convierten en fuentes de poder, adquiriendo así significado para la gran cantidad de estudiantes que no logran ver ningún sentido o placer en la lectura de los libros de matemáticas. (Papert, 1971)

La visión de Papert fue que el sentimiento de poder, que también podría describirse como la autonomía o independencia, experimentado por los estudiantes que usan las matemáticas para lograr objetivos interesantes o importantes, ofrece una vía rápida para eliminar la resistencia al aprendizaje de las matemáticas abstractas. El poder es una sensación de satisfacción y motivación que hace que el aprendizaje sea fácil. Cómo

ayudar a los niños a hacer una conexión emocional con el aprendizaje, es un objetivo pedagógico fundamental, que las herramientas abiertas intentaron facilitar.

2.1 La programación como un medio para hacer (y rehacer)

En nuestro trabajo, ponemos énfasis en la programación. El objetivo no es el de lograr aumentar los ingresos a las escuelas de ingeniería. Pocos niños elegirán carreras de ciencias de la computación. Pero la mayoría de los niños van a vivir en un mundo que estará en gran parte impreso por la computación. La experiencia y la comprensión del pensamiento computacional ayudarán a los niños a navegar ese mundo.

Años de trabajo con niños y la informática, han demostrado que la programación presenta una oportunidad, para participar en dar respuesta a problemas que presentan más de una forma correcta de ser resueltos, generando poder en un ambiente seguro: no hay nada que romper aparte de su programa, y cuando se rompe se genera la oportunidad para encontrar y solucionar el problema. La naturaleza misericordiosa de la programación ofrece un lugar seguro para tomar riesgos intelectuales: no hay una sola manera correcta de hacer las cosas, por lo cual proporciona autonomía al usuario.

El proceso de reparación o "depuración" de un programa - que Cynthia Solomon describió como la gran oportunidad educativa del siglo 21 (Solomon et al., 1985) proporciona una base para el aprendizaje activo a través del ensayo y error. El éxito en el proceso de reparación de un programa también ofrece a los estudiantes la confianza de que pueden aplicar las mismas habilidades identificando problemas, desarrollando hipótesis, creando pruebas y ejecutando soluciones a otros problemas que pueden encontrar. Dado que nuestra intención es la de consolidar una generación de innovadores, abogamos por el uso de la programación como un lugar seguro donde los niños puedan ganar la comprensión y la tolerancia para poder valorar aprender de los errores.

2.2 Comunidades de aprendizaje

Rogoff (Rogoff, 1995) describe la progresión del aprendizaje de la comunidad a través de sus planes de aprendizaje, participación guiada y apropiación participativa. Consideramos como: (1) los estudiantes son guiados (apprenticed) en la práctica por compañeros más experimentados; (2) las interacciones individuales contribuyen a la evolución de la comunidad; y (3) ambas de estas iniciativas conducen al crecimiento individual y desarrollo. En esta sección, consideramos nuestras comunidades de diseño a la luz de estos tres planos para ilustrar su compromiso en el proceso de resolución de problemas y examinamos cómo las potencialidades tecnológicas han estado construyendo conexiones creativas adoptivas para aprender y que ayudan a los miembros de la comunidad a progresar a través de los planos de Rogoff.

Sugar va un paso más allá que la mayoría de los entornos de computación, ya que pone el código fuente de todo programa inmediatamente disponible para el usuario (Bender, et al., 2012). La premisa es que tomando una parte y volviéndola a ensamblar de diferentes maneras, es una clave para comprenderla. En Sugar no hay cajas negras: el usuario ve cómo se hace y cómo lo hace. Con solo pulsar una tecla, la característica "ver fuente" de Sugar, permite al usuario ver en cualquier programa que esté ejecutando (el

procesador de texto, el grabador de video, el navegador web, la calculadora, etc.) y también modificarlo. Vamos más allá “Más allá de las cajas negras” (Resnick, et al., 2000) en que no solo hacemos Sugar visible (por ejemplo, de código abierto), sino que proporcionamos una licencia para hacer cambios y múltiples potencialidades al usuario para realizarlos .

La capacidad no solo de aprender con la máquina y el software, sino también manipular y modificar el software y el hardware, abre por sí mismo la puerta para aprender lecciones mucho más importantes que aquellas necesarias para pasar una prueba. Se lleva a los niños al descubrimiento de que ellos son auténticos creadores de soluciones de problemas en el mundo real. Cuando el presidente de Uruguay entre 2010 y 2015, José Mujica, se enteró que un niño de doce años de un pequeño pueblo en el este de Montevideo había escrito una media docena de actividades para las XO, sonrió y dijo triunfalmente: “Ahora tenemos hackers”. A sus ojos, la capacidad de este niño de contribuir con el desarrollo de una comunidad más amplia era el indicador de cambio y desarrollo en el país (Bender, et al., 2012).

El hacker de doce años y los demás jóvenes con esas habilidades no emergieron espontáneamente. En Uruguay, hay un grupo activo de jóvenes que trabajan como aprendices bajo la guía de mentores dentro de su comunidad, por ejemplo, Flavio Danesse, mentor de Python Joven (Python Joven, s.f.) y apoyo de desarrolladores de la comunidad Sugar en el internet-relay chat (IRC). La tutoría de los jóvenes ocurre de forma natural en el proceso de revisión del código fuente, el cual es central en el ciclo de desarrollo de software. Esto es muy común en los proyectos de software libre, ya que el equipo de desarrollo está disperso geográficamente, la tutoría se produce en línea. La tutoría es una parte indispensable de la cultura libre / Software Libre.

El mismo espíritu abierto promovido en la comunidad de Sugar en el proceso de desarrollo de software está presente en proyectos relacionados con el hardware como ser el Proyecto Butiá y Física con XO. Muchos estudiantes y maestros nos envían correos electrónicos preguntándonos cómo pueden hacer su propio robot, incluso en un país como Uruguay, donde todas las escuelas públicas ya tienen varios kits de robótica comerciales. Si bien puede parecer poco probable que los estudiantes quieran construir sus propios robots, encuentran la experiencia de la construcción de un robot motivador. Es una experiencia de aprendizaje de gran alcance que no sólo ofrece la oportunidad de desarrollar habilidades tecnológicas, sino que también ayuda a los estudiantes a ser independientes y más seguros de sí mismos en sus habilidades para alcanzar nuevas metas. Un ejemplo de esto es una estudiante de 12 años de edad, que tuvo en su propia iniciativa como proyecto de verano, la construcción de un robot desde cero, que se lleva a cabo con una mínima orientación de nuestra comunidad. Federica demuestra la capacidad y el esfuerzo que pueden hacer los niños cuando están automotivados.

De una manera análoga a los proyectos de periodismo llevados a cabo por la comunidad en la década de 1990 (Smith , et al. , 2000), donde los practicantes enseñaron redacción y edición en lugar de conocimientos de informática, la comunidad de Sugar, que consta de desarrolladores y educadores, profesionales y voluntarios, motivan a los estudiantes en el desarrollo de software: ¿cómo crear una aplicación?, ¿cómo crear un parche de software de forma tal que sea aceptado por el equipo principal de desarrolladores?, la

forma de escribir el documento, y así sucesivamente. La atención se centró siempre en el desarrollo y no en la forma de programar. Es decir, los miembros de la comunidad estudiantil no recibieron directamente la formación en Python o Javascript. Ellos finalmente aprendieron habilidades de programación que sin duda mejoraron sus habilidades de desarrollo, pero primero aprendieron a dominar los procesos de desarrollo.

El aprendizaje se produce en la práctica, ya que los estudiantes participan en la creación de sus aplicaciones y en la corrección de errores en el código de base existente. En lugar de separar la instrucción de la práctica real, se anima a los estudiantes a crear, y reciben retroalimentación cuando notamos que están luchando con problemas. Por lo general, esta información viene durante las reuniones de grupo (en el caso de Python Joven) o en línea (en listas de IRC o correo electrónico) o como comentarios sobre los parches enviados a nuestro sistema de seguimiento de errores.

La evidencia de participación guiada se encuentra al examinar la historia de parches del proyecto. Cada corrección de errores o una nueva función para el software de base y las actividades de Sugar se registra en el repositorio git del proyecto (git Sugar Labs, sf). El 30 % de los parches en la versión 0.104 fueron presentados por los jóvenes en la comunidad de Sugar (Notas Sugar Labs Release 0.104, s.f.). Ese número se incrementó a 50 % para la liberación de la versión 0,106 (Notas de Sugar Labs Release 0.106, s.f.). Los aprendices Sugar son participantes activos en su proyecto. Algo tal vez aún más importante, es que los alumnos de Sugar después de algún tiempo se mueven a otros proyectos. Ellos han llegado a un nivel de dominio suficiente, de manera que puedan aplicar sus habilidades en otro lugar, completando la progresión de Rogoff.

Como Smith et al. han señalado (Smith et al., 2000), es raro que los niños obtengan crítica sin diluir de los adultos. Los mentores del proyecto permiten a los estudiantes escribir aplicaciones, proponer características y enviar parches como compañeros y estar expuestos a los mismos criterios de crítica como los demás miembros de la comunidad: no se hacen distinciones en cuanto a la edad o experiencia. Vemos esto como una valiosa oportunidad para que los estudiantes aprendan; como desarrolladores, deben estar preparados para defender sus posiciones con sus pares y la comunidad de usuarios.

3. Una plataforma abierta

La mayor parte del trabajo que describimos en este artículo fue hecho utilizando la computadora portátil XO (ver Figura 1) desarrollada por One Laptop Per Child (OLPC). El software que se ejecuta en ella fue producido por Sugar Labs, un proyecto miembro del Software Freedom Conservancy (SFC, 2015).



Figura 1: La portátil XO

Todas las versiones de la portátil XO incorporan numerosos sensores. La versión 4.0 por ejemplo tiene los siguientes sensores incorporados: cámara, micrófono, pantalla multi-touch, touchpad, teclado, botones y game pad, sensor de campo magnético, termómetro, receptor de radio WiFi, sensor de luz y acelerómetro (XO, 2008)

La XO puede conectarse a cualquier placa de entrada/salida (E/S) de bajo costo a través de los conectores de los tres puertos serie universal USB disponibles para ampliar sus capacidades de interacción con sensores y actuadores. Los puertos USB se han utilizado para conectar lectores RFID, Arduino, Lego Mindstorms NXT (Mindstorms, s.f.), Lego WeDo (WeDo, s.f.), placa GoGo (GoGo, s.f.), USB4Butiá (USB4Butiá, s.f.) y otros dispositivos de entrada/salida (incluido ultrasonido portable).

La característica más significativa de la XO para nuestro trabajo, es el conector para micrófono externo, el cual además puede utilizarse como entrada para la adquisición de datos acoplado para corriente continua (DC). La tarjeta de sonido de la XO fue diseñada para permitir seleccionar por software el acople a señales alternas (AC) o continuas (DC). Esta decisión, aparentemente simple, (inspirada por Papert e implementada por Mark Foster) supuso una apertura significativa de las posibilidades de la XO para conectarse al mundo físico mediante medidas de voltaje continuo (DC) y resistencia. En Uruguay promovemos el diseño, construcción, calibración y programación de estos sensores, para fomentar la creatividad y el empoderamiento de niños y jóvenes con sus XO (ver Figura 2). Animamos a todos los fabricantes a incluir esta característica en el hardware que desarrollan para el mercado de la educación.

La figura 2 corresponde a un Taller que hicimos en el Museo Pedagógico “José Pedro Varela” (Montevideo, Uruguay) para un grupo de escolares que no tenía conocimiento previo alguno de trabajo con circuitos y programación: en una hora y media aprendieron a montar sensores y crear programaciones divertidas con ellos. Al terminar una niña nos abrazó repentinamente y exclamó: “¡muchas gracias por enseñarnos todo esto!”. Se trata de una de las mejores repercusiones que tuvo nuestro trabajo.

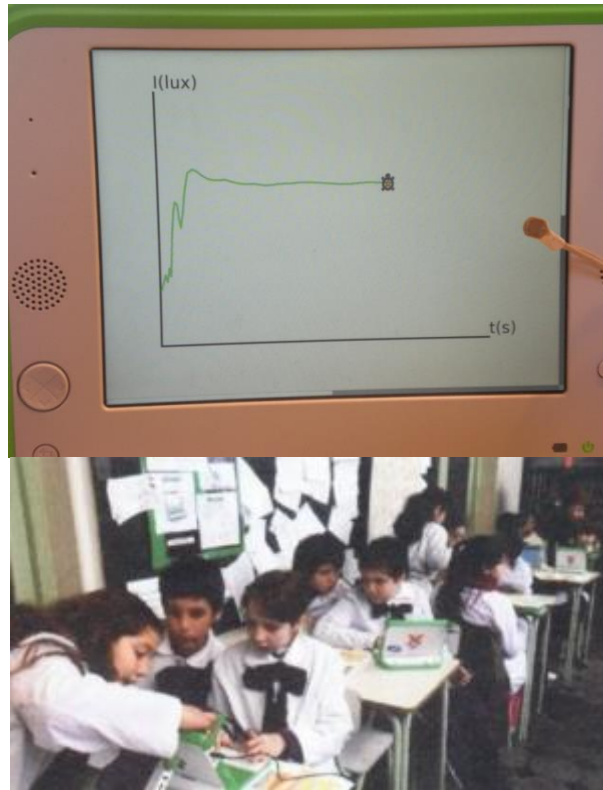


Figura 2: Un sensor de luz y sus medidas mostradas en forma gráfica en la pantalla de una XO (arriba); escolares de Uruguay explorando la construcción y programación de los sensores (abajo).

3.1 Magnitudes físicas medidas a través del conector de entrada de micrófono externo

La XO (al igual que cualquier otra computadora), además de registrar pequeños voltajes de corriente alterna -AC- a través de su conector de entrada de micrófono externo, es capaz de medir voltajes de corriente continua DC (en el rango de 0.40 a 1.85 V aproximadamente) y resistencia (en el rango de 700 a 14.000 ohm) (ver Figura 3). Dichos rangos varían según la unidad y el modelo particular utilizados. Se pueden hacer lecturas con una frecuencia máxima de muestreo de unos 20 Hz. Esto permite la medida directa de magnitudes físicas. La capacidad de medida de la computadora puede ampliarse aún más mediante el uso de transductores: componentes electrónicos que traducen las magnitudes físicas como cantidad de luz incidente, temperatura, campo magnético, etc., en voltaje o resistencia. La elección de los mismos dependerá de la disponibilidad local y de su precio. Otra consideración es la seguridad. A diferencia de los transductores resistivos, los cuales no suponen ningún riesgo, los de voltaje pueden causar daños irreparables a la XO si se la somete a valores de tensión inadecuados. En nuestra experiencia de aula, no hemos puesto restricción alguna al uso de los sensores resistivos, pero supervisamos cuidadosamente el uso de aquellos basados en voltaje. Utilizamos estos últimos únicamente en el trabajo con adolescentes.

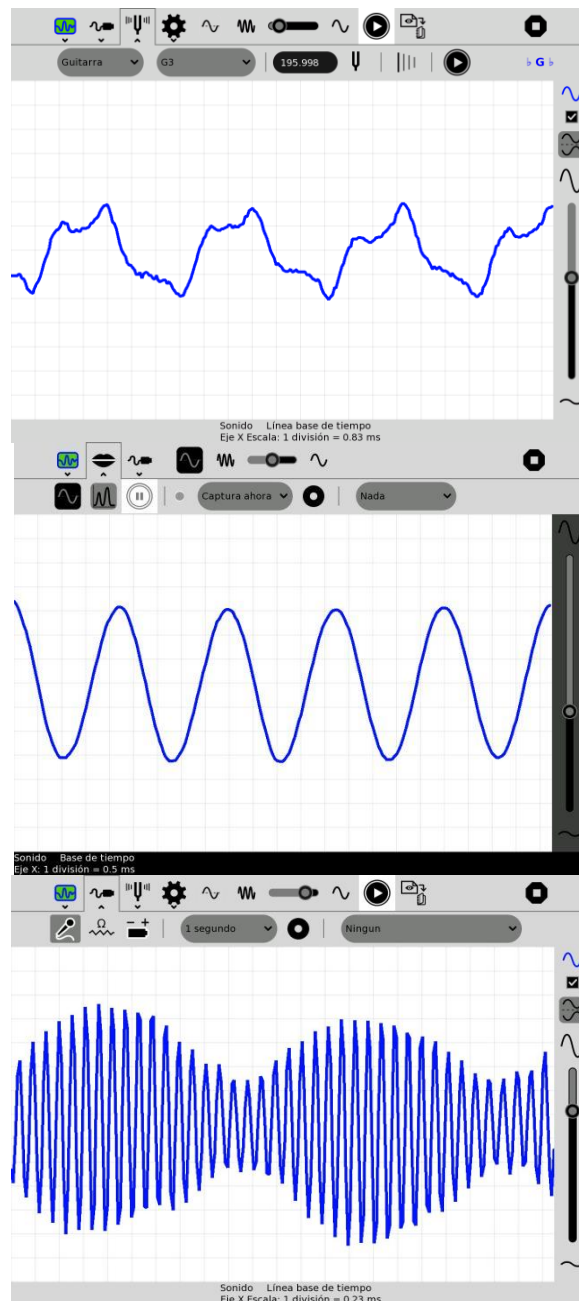


Figura 3: La Actividad MEDIR es un osciloscopio que puede utilizarse para visualizar entradas analógicas de audio y de corriente continua.

3.2 Empoderando al usuario con Software Libre.

La interfaz Sugar que usamos en nuestros desarrollos es software libre: cualquiera es libre de copiarlo o modificarlo. Los programas de Sugar, llamados “Actividades”, vienen con una invitación explícita a la experimentación. Las Actividades están escritas en Python (Python, 2015) o Javascript (W3C 2015). Como en ambos casos se trata de lenguajes interpretados, el código fuente puede ser fácilmente visto y editado por los usuarios. El mecanismo antes descrito permite ver el código fuente mediante un simple clic del mouse. Un segundo clic desde la ventana de visualización del código, crea un duplicado del mismo, disponible para modificaciones del usuario final (ver Figura 4). Como Sugar emplea un enfoque de modificaciones del tipo “copia mientras se escribe”

no hay peligro de que el usuario final dañe el programa instalado; por lo tanto la sanción por “tomar el riesgo” de modificar el código es esencialmente nula.

4. La programación como una actividad

En los últimos años se ha visto una explosión de proyectos de "aprender a codificar". Es de destacar que los lenguajes basados en bloques, eliminan la posibilidad de errores sintácticos que a menudo hacen tropezar a los programadores novatos. Normalmente, esto tiene un precio: al mantener las cosas simples se imposibilita que los usuarios lleguen a la complejidad y las ideas poderosas subsiguientes que son inherentes a Logo y otros lenguajes basados en texto. En nuestro trabajo, no rehusamos en dar a los usuarios el acceso a los conceptos de programación de gran alcance, tales como variables y procedimientos sin diluir; también se encuentran disponibles múltiples mecanismos de extensiones al lenguaje y pasarelas explícitas desde los bloques a la programación basada en texto.

Turtle Blocks (Turtle Blocks, s.f., Bender, 2015) es una actividad de Sugar inspirada en Logo (Abelson y diSessa, 1980), (Harvey, 1997) "tortuga" gráfica que dibuja el arte colorido basado en elementos de programación visual que se encajan. Los programas son creados uniendo entre sí bloques que representan elementos de programación.

El "piso bajo" de Turtle Blocks proporciona un punto de entrada fácil para los principiantes. También cuenta con funciones de programación de "techo alto" que desafían a los estudiantes más aventureros. Al igual que en la mayoría de los ambientes Logo, en Turtle Blocks, la tortuga existe en tres formas: (1) como un robot que comparten el mismo espacio físico como el niño; (2) como un objeto computacional que se mueve en la pantalla; y (3) como una entidad matemática abstracta.

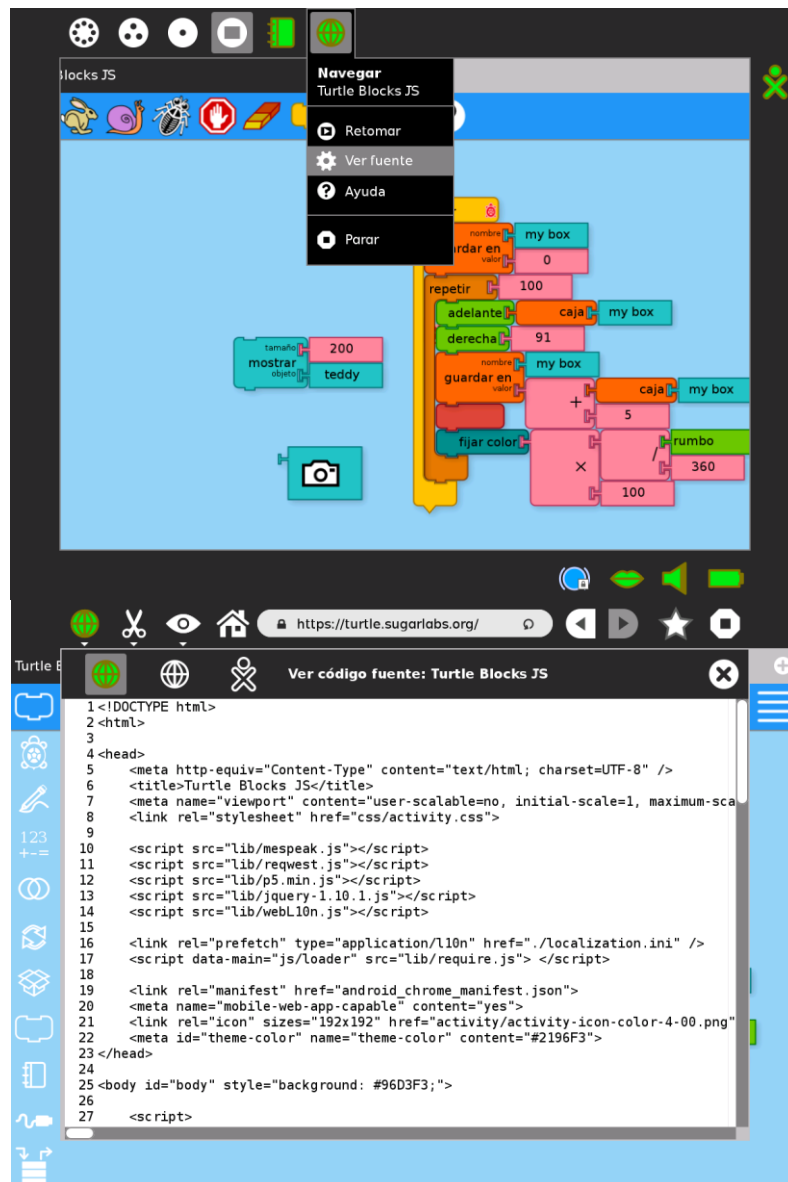


Figura 4: El ítem de menú Ver fuente (arriba) y la visualización del código fuente del navegador (abajo).

Turtle Blocks surge del programa Turtle Art de Brian Silverman (Turtle Art, s.f.), diseñado para explorar más a fondo la riqueza del entorno de Sugar. A diferencia de Turtle Art, Turtle Blocks incluye bloques para la interfaz de sensores, el acceso a multimedia y funciones de programación adicionales, como una pila, variables y procedimientos nombrables, y el acceso en línea a Python. (El concepto de extensiones en línea ya ha sido explorada en GameMaker (Vigilantes, s.f.), que proporciona un editor de código interno.) Turtle Blocks está disponible como una actividad Sugar escrita en Python y como aplicaciones web Android escritas en Javascript (Turtle Blocks 2015b).

Muchas de las funciones de Turtle Blocks se han añadido a petición y / o con la ayuda de los profesores. Por ejemplo, un profesor en Uruguay, pidió la incorporación de un bloque de raíz cuadrada para usar en la enseñanza del teorema de Pitágoras. En ese momento, la creación de un nuevo bloque era tedioso y estaba más allá de las

capacidades de la mayoría de los profesores, por lo que fue añadido por el mantenedor de la aplicación. Como se describe en la siguiente sección, la versión actual de Turtle Blocks hace que esas peticiones puedan cumplirse en forma trivial. Se han hecho esfuerzos para extender una invitación a la comunidad de usuarios a modificar, ampliar y experimentar con Turtle Blocks, y así se convierta en un agente activo de construcción y mantenimiento de las herramientas utilizadas en el aprendizaje.

4.1 Bloques de sensores

Turtle Blocks tiene una paleta integrada para diferentes tipos de sensores: brillo, estado del mouse, teclado, hora, audio, voltaje, resistencia, entre otros. (Ver Figura 5). Los plugins, descritos más abajo, se utilizan para agregar bloques de sensores adicionales al lenguaje. Los bloques de sensor, que son intercambiables con los bloques numéricos, hacen que sea fácil programar con la entrada del sensor.

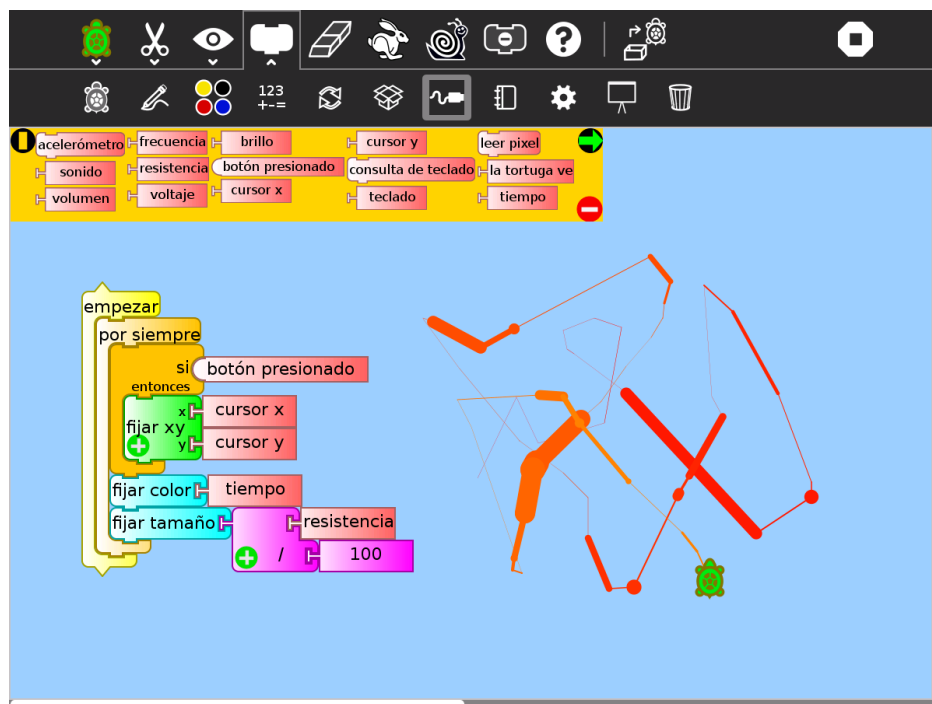


Figura 5: Accediendo a sensores con Turtle Blocks (en este ejemplo, un simple programa de dibujo, un fotoresistor es usado para cambiar el tamaño de la pluma y un sensor de tiempo es usado para cambiar el color).

Por ejemplo, el programa mostrado en la Figura 6 mide e imprime la frecuencia dominante de una señal leída desde el micrófono interno o la entrada externa de micrófono. (La frecuencia de la componente más fuerte es reportada en Hertz. La resolución es ± 8 Hz). Esta simple pila de bloques podría ser usada, de forma introductoria, en un programa afinador de guitarra.

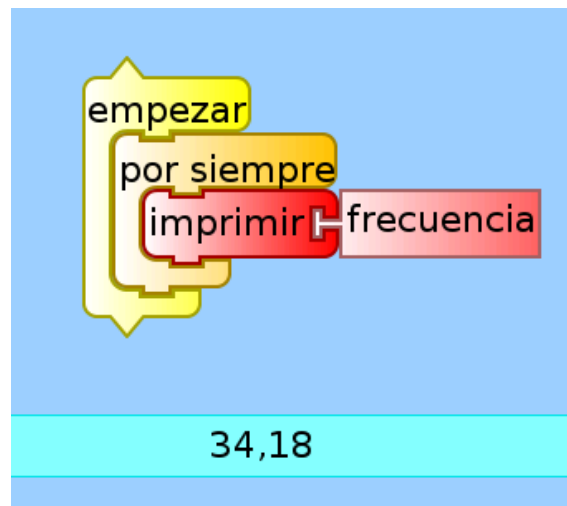


Figura 6: Midiendo la frecuencia dominante

Uno de los desafíos típicos del uso de sensores es la calibración. Esto también es cierto en Turtle Blocks. El ciclo de vida típico de un proyecto incluye: (1) caracterizar al sensor -a menudo usando la actividad Medir como se muestra en la Figura 3-, (2) leer valores como se muestra en la Figura 6, (3) graficar los valores a medida que cambian en el tiempo, (4) encontrar los valores mínimo y máximo y, finalmente, (5) incorporar el bloque de sensor en un programa. Por ejemplo, el valor del umbral de 3000 usado en el ejemplo que se muestra en la Figura 9, puede ser determinado por un proceso de este tipo.

4.2 Modificaciones y extensiones

Turtle Blocks viene con bloques integrados para la conexión de sensores y dos mecanismos adicionales para la conexión de dispositivos: (1) Bloques de código Python y Javascript y (2) plugins. Nuestro énfasis en la modificación y extensión de Turtle Blocks es pragmática: los sensores y la tecnología robótica a disposición de maestros y estudiantes varían de una escuela a otra y de un despliegue de Sugar a otro. Para dar apoyo a estas variaciones desde un punto central requeriría más tiempo del que los mantenedores pudieran comprometer. Un modelo descentralizado ha demostrado, una y otra vez, cómo grupos locales extienden el soporte para satisfacer sus necesidades. También tenemos una razón pedagógica para este énfasis: queremos facilitar la progresión de la comunidad de aprendizaje a la etapa de apropiación participativa de Rogoff.

4.2.1 Python/Javascript en línea

La forma más directa de modificar Turtle Blocks es usar el bloque Python. (Un mecanismo similar existe en la versión Javascript de Turtle Blocks. El bloque Eval permite insertar arbitrariamente código Javascript en un bloque. Como se ve en la Figura 7, el bloque Python toma una cadena de texto como entrada, así como también hasta tres argumentos que pueden ser sustituidos en una cadena de texto antes de ser pasados al intérprete Python. En el ejemplo, la cadena de texto de entrada es usada para calcular el seno de un ángulo expresado en grados. Por razones de seguridad, solo un

conjunto limitado de módulos están disponibles para el intérprete Python cuando se ejecutan en este modo.

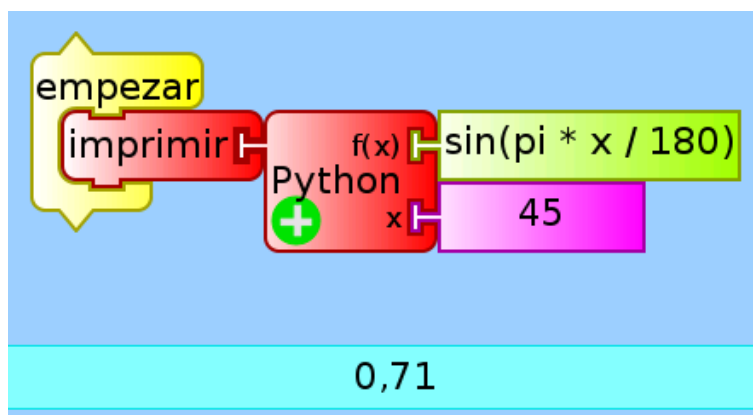


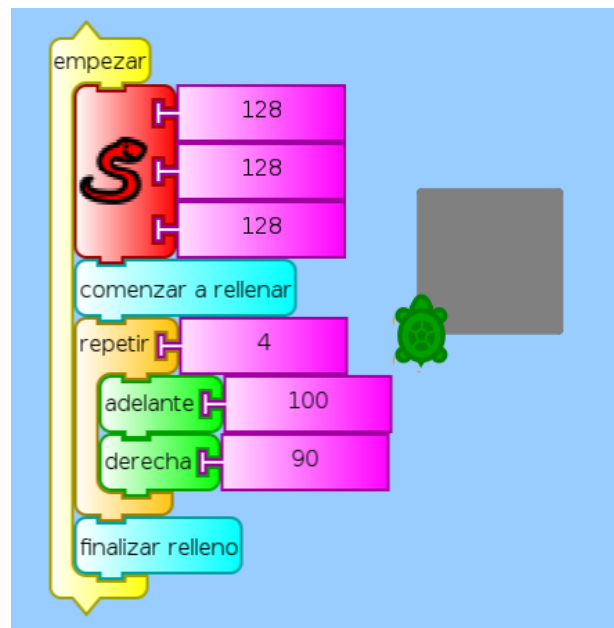
Figura 7: El uso de código Python en línea para ampliar Turtle Block

4.2.2 Bloque de código Python incrustado (Embedded Python-code block)

Ya que Turtle Blocks está escrito en Python, es posible incrustar código Python dentro de un bloque especial destinado a expandir el lenguaje. Brian Harvey y Jens Mönig hicieron algo similar cuando construyeron sus propios proyectos Build Your Own Blocks (BYOB) y Snap! (Snap, s.f.), que tienen como objetivo estudiantes de secundaria. En el ejemplo de la Figura 8, el código incrustado es usado para establecer el valor RGB de la pluma de la tortuga.

Otro ejemplo sencillo se muestra en la Figura 9. El conector de entrada de micrófono externo (acoplado DC) es usado para hacer un timbre de puerta/alarma antirrobo con un interruptor que puede ser construido a partir de dos cables. Este es un proyecto con un costo de materiales insignificante y adecuado para hacerse con materiales reciclados. La computadora mide la resistencia conectada a la entrada de micrófono y hace sonar una alarma cuando su valor cae por debajo de un nivel predeterminado.

En el momento en que el programa de la Figura 9 fue escrito, Turtle Blocks no tenía un bloque para producir un tono de audio, pero por el diseño abierto, uno podía crearlo. Debido a que el bloque de sonido codificado en Python resultó ser tan útil, fue finalmente añadido a la paleta estándar de bloques de Medios, un ejemplo de cómo las modificaciones de los usuarios finales pueden trabajar “aguas arriba” de la “rama principal” del proyecto.

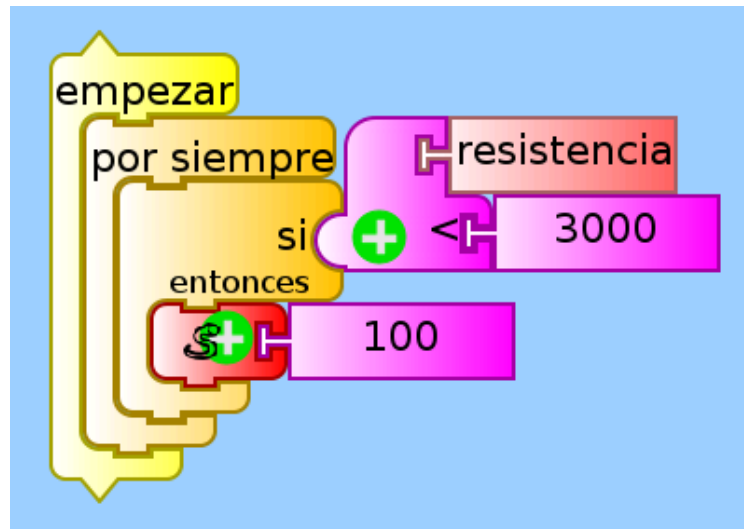


```
def myblock(tortuga, rgb):
    """ Establece el color rgb desde los valores """
    tortuga.canvas._fgrgb = rgb[:]
```

Figura 8: Importando este código en Python para agregar un bloque “Establecer RGB”

Turtle Blocks proporciona soporte para plugins, un conjunto de componentes de software que agrega habilidades específicas a una aplicación de software más grande. Los plugins de Turtle Block permiten a otros desarrolladores agregar paletas y bloques para soportar funcionalidades adicionales sin tener que hacer cambios en el núcleo del código principal de Turtle Blocks. Si un plugin está presente, es cargado cuando Turtle Blocks se inicia y cualquier paleta o bloque definido en el plugin está disponible para el usuario.

El mecanismo de plugins es actualmente usado para proporcionar soporte para los sensores, la cámara, RFID, y las paletas Medios, Extras, y Portafolio. Un plugin ha sido desarrollado para Butiá, Arduino, LEGO NXT, LEGO WEDO, FischerTechnik y GoGo. Otro plugin para un motor de simulación de física 2D Box-2D (Box2D. s.f.). Para añadir un plugin, simplemente se debe poner el archivo dentro de la carpeta de plugins de Turtle Blocks. La próxima vez que ejecute Turtle Blocks, los bloques definidos en el plugin estarán presentes. Por ejemplo, un bloque “vuelta en U” agregado por el mecanismo de plugins mostrado en la Figura 10).



```
def myblock(tw, x):
    import os
    os.system('speaker-test -t sine -l 1 -f %d' % *int(x))
```

Figura 9: Un simple timbre de puerta implementado en Turtle Blocks usando la extensión Python



Figura 10: Adición de un bloque de cambio de sentido (giro en U) a la paleta de tortuga

Un mecanismo similar está disponible para la versión Javascript del programa. Un script codificado en JSON puede ser cargado en tiempo de ejecución para extender las paletas. En el caso de Javascript, un número servicios de web han sido escritos (por jóvenes) para extender el lenguaje, incluyendo plugins para acceder a Google Maps (Ver Figura 11), el clima, servicios de traducción, etc.

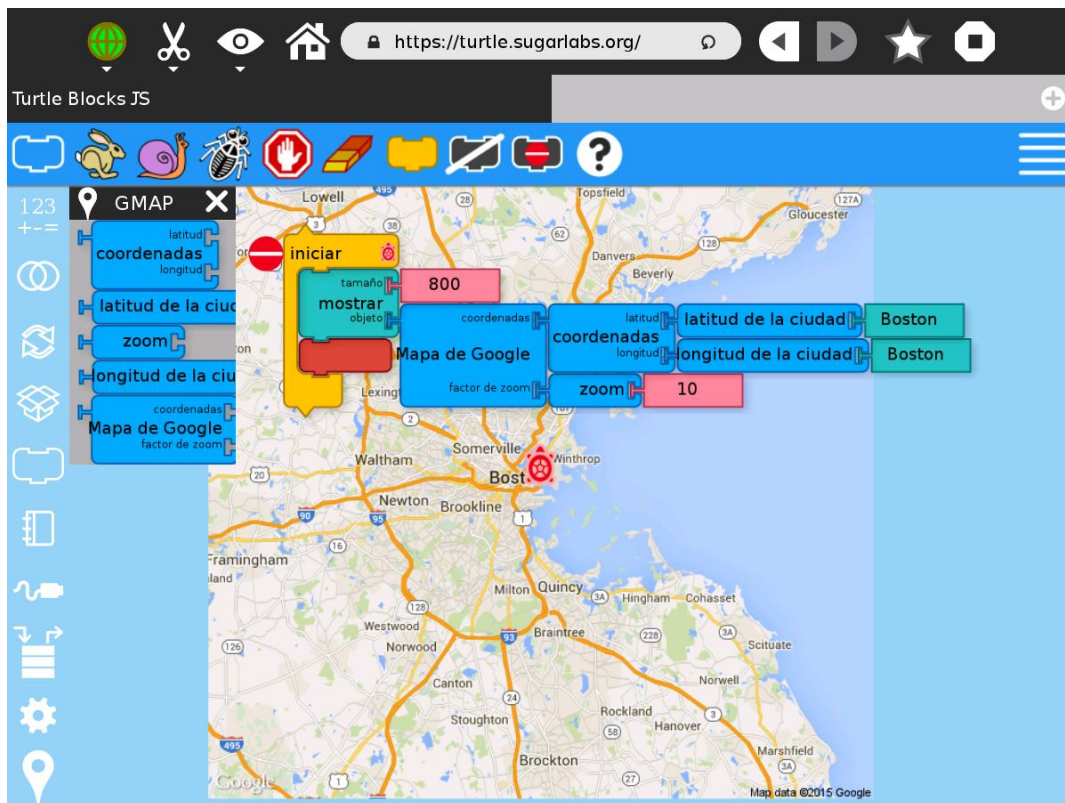
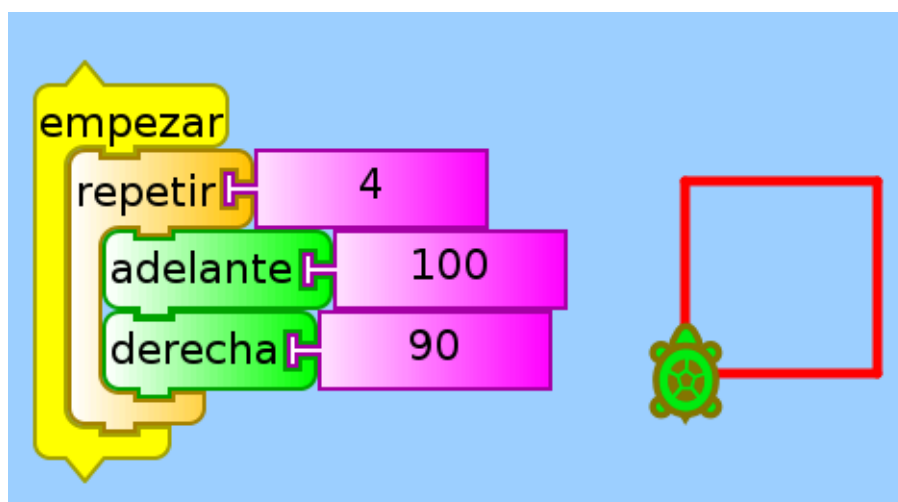


Figura 11: El plugin de Google Maps

4.2.4 Exportación de programas como Python

Turtle Blocks tiene otra característica añadida por Marion Zepf (Zepf, 2013) que fomenta otro tipo de modificación para el usuario final: cualquier programa escrito en lenguaje de bloques puede ser exportado como Python. Este código puede ser editado en un editor de textos y puede ser empaquetado (bundled) y ejecutarse como una actividad Sugar independiente o ejecutarse directamente desde la consola/intérprete (shell) de Linux. La Figura 12 muestra el código de Turtle Blocks para dibujar un rectángulo y su correspondiente código en Python que es generado con la opción de “exportar a Python”.



```
#!/usr/bin/env python
```

```

from pyexported.window_setup import *
tw = get_tw()
ACTION = {}
def start():
    turtles = tw.turtles
    turtle = turtles.get_active_turtle()
    canvas = tw.canvas
    for i in range(int(4.0)):
        turtle.forward(100.0)
        turtle.right(90.0)
    yield True
yield True
ACTION["start"] = start

```

Figura 12: Exportación de código Python para dibujar un cuadrado.

El mecanismo de exportación sirve como puente entre la programación basada en bloques y la programación basada en texto. Nuestro objetivo no es capturar a los estudiantes en Turtle Blocks para siempre, sino darles incentivos y oportunidades de avanzar hacia la esfera más expresiva de lenguajes basados en texto.

5. Sensores creados con tecnología de reciclaje

A partir de que la XO es capaz de tomar medidas a través de su conector de micrófono externo (acoplado DC) se pueden construir sensores de bajo costo (ver Figuras 13 y 14) utilizando piezas disponibles aún en los lugares más remotos.

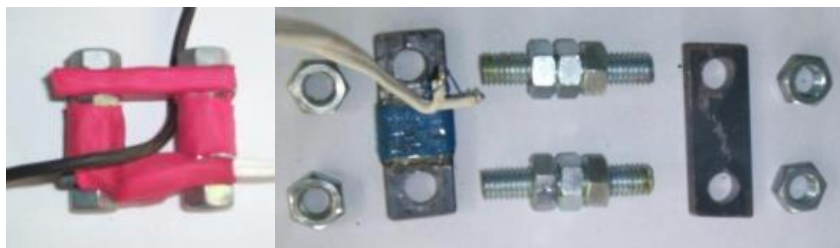
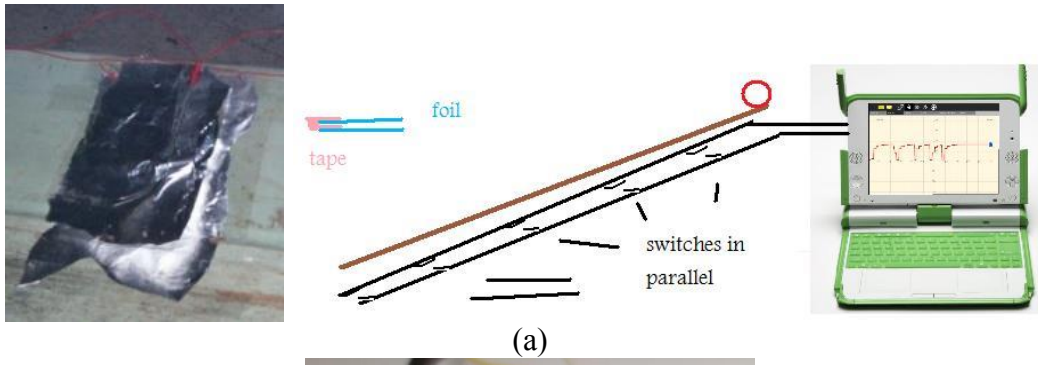


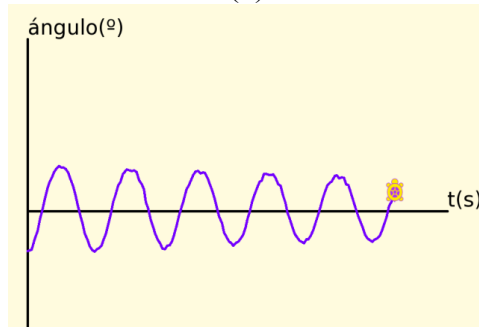
Figura 13: Este transformador de corriente de bajo costo utiliza tecnología fácilmente disponible. Cuando se lo conecta a la entrada de micrófono externo, puede ser utilizado para medir intensidad de corriente alterna (y por inferencia Potencia de alimentación de corriente alterna).



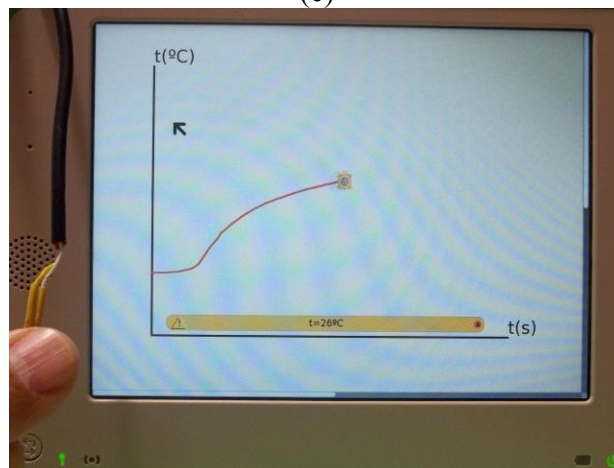
(a)



(b)



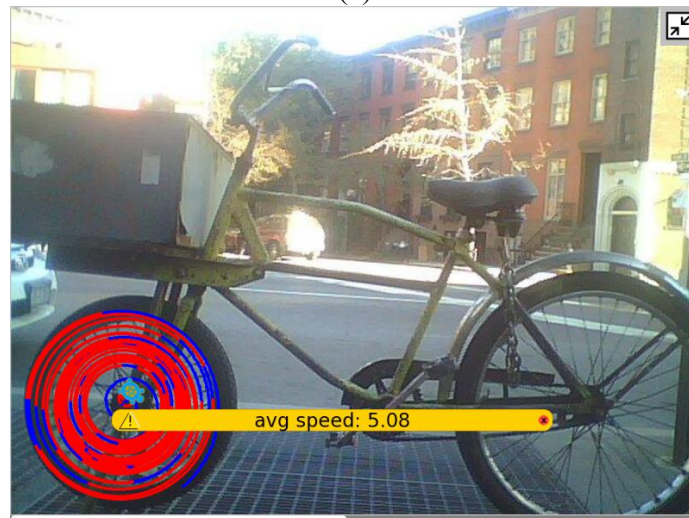
(c)



(d)



(e)



(f)

Figura 14: (a) Interruptores simples (izquierda) - fabricados con hojas de aluminio- utilizados para medir la aceleración en un plano inclinado. A medida que una esfera desciende por el plano los interruptores se activan en sucesión. El intervalo de tiempo entre los eventos permite calcular la aceleración gravitatoria; (b) Un LDR (transductor resistivo de luz incidente) como sensor de luz; (c) Un resistor variable (usualmente llamado Potenciómetro) como un sensor de posición angular. La figura muestra las oscilaciones de un péndulo; (d) Un termistor NTC (transductor resistivo de temperatura) como sensor graficando la temperatura de la mano; (e) Ídem graficando la temperatura del *mate*; (f) Utilizando un sensor de efecto Hall (campo magnético) como un odómetro/velocímetro de bicicleta. El sensor está montado en el cuadro para detectar la presencia de un imán unido a uno de los rayos de la rueda delantera.

5.1 Física con sensores y programación simple

Trabajamos con niños de tan sólo seis años de edad, y con este grupo nos esforzamos por desdibujar las líneas en la mente del niño entre el juego y el aprendizaje y el

refuerzo de las recompensas de lo que Papert llama "diversión dura". Los estudiantes han demostrado tener un vínculo emocional con su hardware. Esto se puede ver en las "personalizaciones" que han hecho con sus computadoras portátiles XO (por ejemplo los niños añaden imágenes, leyendas, adhesivos, etc. Algunos incluso llegan tan lejos hasta reprogramar sus computadoras de escritorio). Aprovechamos esta tendencia de personalización en el proyecto Física con XO (Trinidad, 2013): los estudiantes, la mayoría de los cuales nunca han tenido acceso o conocimiento acerca de la programación de computadoras, crean divertidos proyectos que utilizan instrumentos hechos a medida para el registro de magnitudes físicas. Un principio fundamental es utilizar un mínimo conocimiento de la programación y la electrónica y el diseño de casi todos sus proyectos directamente

Un proyecto introductorio utiliza un programa Turtle Blocks y el micrófono incorporado a la XO para detectar si la frecuencia de sonido o el volumen superan cierto valor umbral programado con el fin de activar la tortuga para que dibuje o emita un mensaje hablado. A partir de este ejemplo motivador inicial los alumnos construyen sus propios proyectos interactivos utilizando sensores de contacto (interruptores simples on /off), sensores de luz (basados en LDR), o de temperatura (basados en termistores). Los programas rara vez requieren más de seis a diez bloques y el circuito es simplemente un dispositivo conectado a un cable de audio, ya sea por pinzas de cocodrilo o soldadura. Un ejemplo del uso de un LDR para registrar el pulso se muestra en la Figura 15.

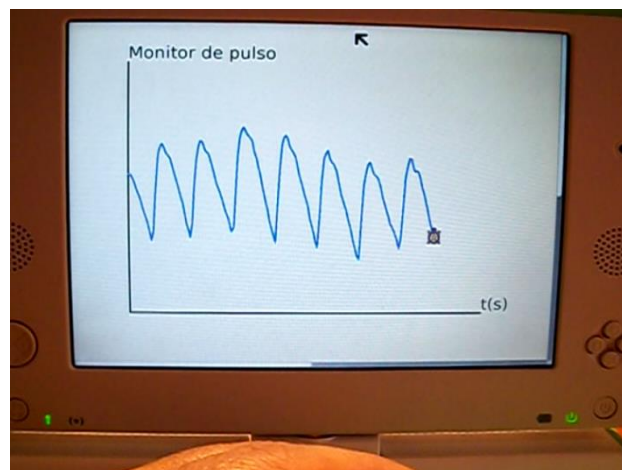


Figura 15: Colocando el extremo de un dedo de su mano entre un LDR y una luz intensa se puede registrar el pulso de la persona.

Para los proyectos más complejos pueden utilizarse sensores integrados (como los de efecto Hall para medir campos magnéticos), alimentándolos con los 5 V DC disponibles en los conectores USB. Tomando las precauciones del caso puede estudiarse el decaimiento exponencial de la función $V = f(t)$ correspondiente a la descarga de un capacitor a través de un resistor (ver Figura 16).

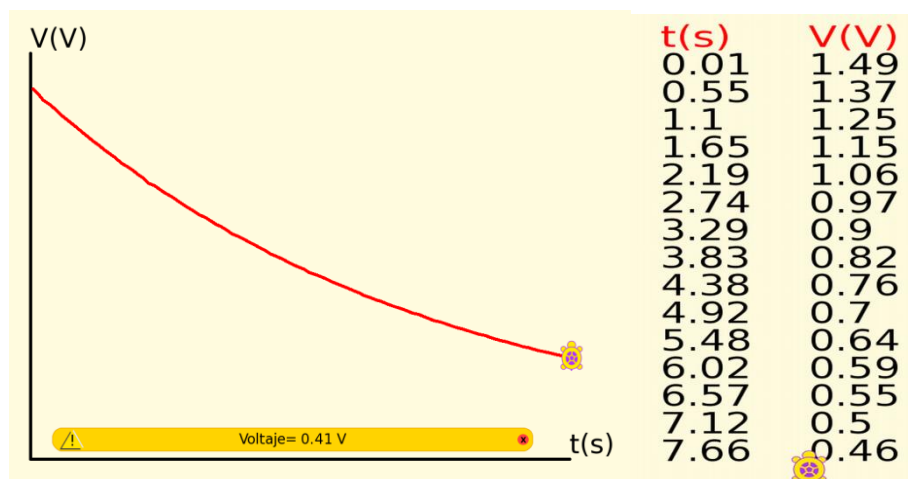


Figura 16: Uso del bloque *voltaje* para medir el decaimiento exponencial.

Un proyecto que ha generado el interés de los estudiantes es la programación de una alarma antirrobo. Se puede hacer con sólo cerrar un contacto (como se describió anteriormente) utilizando únicamente dos cables. Pero lo hicimos con un sensor piroeléctrico que cierra un contacto entre dos terminales cuando un cuerpo caliente (por ejemplo, una persona o animal) cambia de posición. El sensor es alimentado por una batería de 9 voltios; los terminales están unidos al cable de audio conectado a la entrada de micrófono externo de la XO (Ver Figura 17). Esta variación mínima en el diseño despertó tal interés en los estudiantes que corrieron a jugar con el aparato. Esto revela (de nuevo) que es más importante el cómo diseñar una actividad de aprendizaje atractivo y cómo la motivación puede llevar a la apropiación del conocimiento por los estudiantes.

Todos los detalles técnicos relacionados con la construcción de sensores, programación y más de 30 aplicaciones, como juegos y experimentos de física, están publicados en Física con XO (Trinidad, 2013).

Aunque el proyecto original se basa en el trabajo con el hardware de la XO, todas estas ideas se pueden desarrollar en computadoras que utilicen otras interfaces con el mundo físico (Ver USB4Butiá descrita más adelante).

Considerando solamente desde la fecha de creación del sitio web Física con XO (15/09/2009) hasta fines de 2012, son decenas las exposiciones, conferencias, ponencias, Talleres y entrevistas en prensa escrita, radio y televisión en las cuales el autor ha participado para la difusión y enseñanza de las aplicaciones. Se estima en más de 10,000 las personas que tomaron contacto directo con el Proyecto (incluyendo solamente las visitas desde Uruguay al canal Youtube), lo cual es un impacto destacable teniendo en cuenta que la población del país apenas supera los 3 millones y que se trata de un esfuerzo individual. Se decidió difundir básicamente entre docentes, estudiantes de magisterio y profesorado para que el efecto multiplicador fuese mayor.

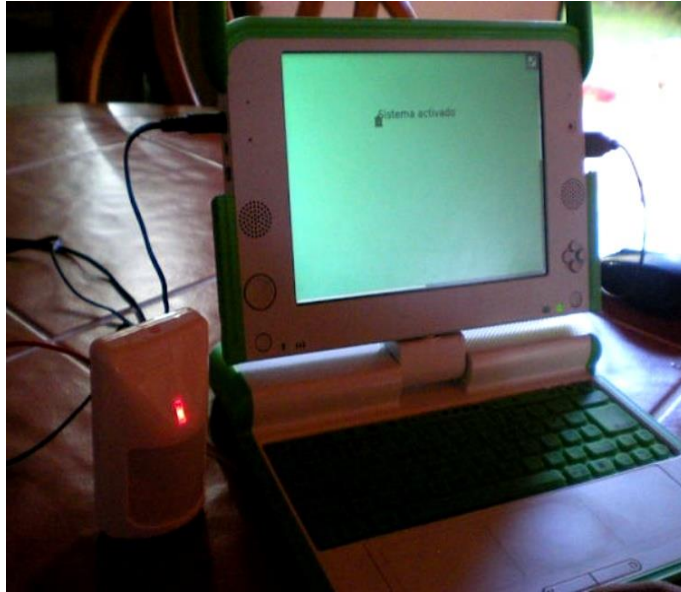


Figura 17: Un sensor piroeléctrico usado como sensor de movimiento.

6. Butiá: Expandiendo capacidades con una placa E/S

En setiembre de 2009 el grupo de investigación MINA de la Facultad de Ingeniería de la Universidad de la República de Uruguay comenzó a trabajar en Proyecto Butiá (Proyecto Butiá página inicial, s.f.). El objetivo del proyecto era crear una plataforma robótica de código abierto, de bajo costo y versátil, para enseñar robótica e informática en las escuelas públicas. Los objetivos fueron: (1) ampliar la capacidad de una computadora portátil para controlar los actuadores y leer los sensores; y (2) reducir las asimetrías entre los sistemas educativos públicos y privados. Se basa en el software Sugar y en el hardware de la portátil XO, con el fin de convertir esta computadora en un robot móvil educativo de bajo costo con alta capacidad y potencia de cálculo. Recientemente se ha brindado compatibilidad con otras plataformas de hardware, como ser netbooks o tablets.

La naturaleza física del robot "permite a los niños dibujar en sus habilidades e intuiciones sofisticadas para detectar y manipular los entornos en los que viven, mientras que la capacidad de programación digital les permite convertir estas intuiciones en conocimiento formal" (Picard et al., 2004).

Detrás de estas ideas están presentes los siguientes conceptos: (1) la resolución de problemas se puede tomar como el proceso mediante el cual se aplica el conocimiento previo a las nuevas situaciones, muy lejos de aquel desde el que se originó (Villela, 2006), y (2) que cooperan para trabajar juntos con el fin de alcanzar objetivos comunes. En una situación de cooperación, los individuos buscan resultados que son beneficiosos para ellos y todos los demás miembros del grupo (Johnson et al., 1999).

Como se ve en la Figura 18 (a la derecha), el kit robótico Butiá consiste en una plataforma con ruedas sobre la cual se asienta la XO y un conjunto de sensores y actuadores a los que se conecta. Está diseñado para promover la experimentación, permitiendo que el estudiante elija dónde poner los sensores en la plataforma utilizando las piezas de plástico perforadas proporcionadas con el kit.

Nuestro objetivo es permitir a los estudiantes explorar y aprender la plataforma. Con el fin de ofrecer a los estudiantes las herramientas necesarias para programar el comportamiento del robot necesitábamos representar los detalles técnicos de una manera sencilla y comprensible. Elegimos apoyar el uso de las actividades de Sugar con las cuales los estudiantes en Uruguay ya estarían familiarizados, para reducir la brecha de aprendizaje. En consecuencia, el uso de Turtle Blocks era la elección natural para programar el robot en los grados más jóvenes (y para los estudiantes que no habían tenido experiencia previa en ciencias de la computación). Un beneficio adicional del uso de Turtle Blocks era su soporte incorporado para la interacción física, lo que permite un fácil acceso al micrófono incorporado y a la cámara de vídeo.

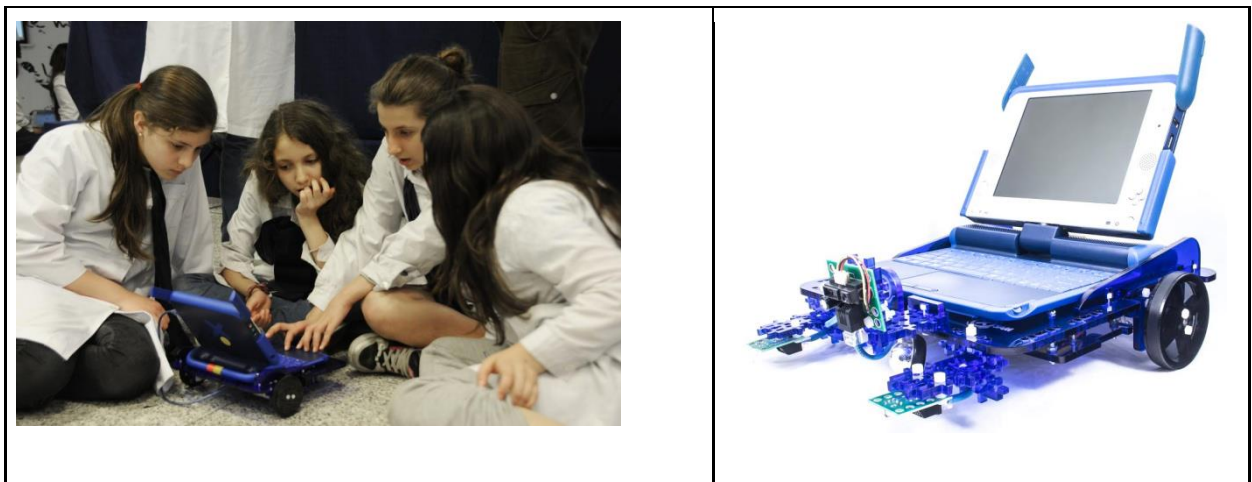


Figura 18: A la izquierda, estudiantes programando el robot Butiá (fotografía cortesía de ANTEL); a la derecha, un robot Butiá (fotografía de Pablo Brera)

6.1 El plugin Butiá de Turtle Blocks

Para controlar el robot Butiá se ha desarrollado un plugin para Turtle Blocks. Este software (TurtleBots página inicial, s.f.) permite que los programas de Turtle Blocks interactúen con los dispositivos conectados al robot. Los sensores y actuadores son un conjunto de servicios que tienen una correspondencia directa con los bloques de Turtle Blocks. Estos servicios pueden estar disponibles o no en todo momento. Decidimos cambiar el color de los bloques en función de la disponibilidad del servicio y la conexión física del dispositivo evitando la necesidad de configurar un puerto físico en la placa E/S. Este mecanismo es fácil de usar y también es útil para las pruebas: si el bloque es de color verde se puede concluir que la comunicación con el robot está funcionando correctamente.

La utilización de muchos sensores del mismo tipo es una práctica común en la robótica. Para implementar esto tenemos la paleta Butiá la cual genera automáticamente nuevos bloques para cada dispositivo conectado. Por ejemplo, si dos sensores de grises están conectados ambos se van a diferenciar por el número de puerto en que están conectados. Como se muestra en la Figura 19, el robot tiene conectado dos sensores de grises, uno en el puerto 1 y otro en el 4. Además uno de distancia en el puerto 5.

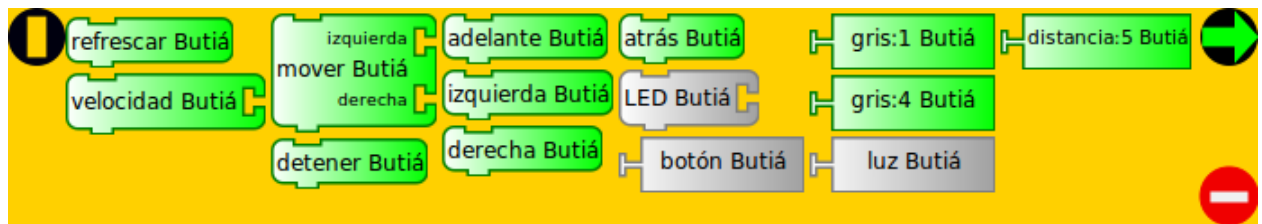


Figura 19: Una instancia de la paleta Butiá en Turtle Blocks

La visión por computadora es un tema importante de investigación en el campo de la robótica. Incluye el reconocimiento de colores, formas y el seguimiento de objetos. Los algoritmos relacionados con estos conceptos requieren un alto nivel de conocimientos en matemáticas y programación, por lo que tienden a estar más allá del alcance de los estudiantes principiantes de informática. Por esa razón, se optó por dar apoyo a las primitivas de orden superior en el entorno de programación que permita a los estudiantes crear comportamientos robóticos básicos de una manera fácil. Hemos desarrollado un componente implementado como un plugin Turtle Blocks llamado FollowMe, que permite a los usuarios elegir un color de un objeto y luego obtener las coordenadas de dicho objeto en la vista de la cámara. Uno de los problemas clásicos de la robótica es el reconocimiento y seguimiento de objetos. Este plugin permite a los estudiantes trabajar hacia las soluciones a esta clase de problemas, mientras que encapsula la complejidad de los algoritmos de visión por computador en el plugin FollowMe (Ver Figura 20).



Figura 20: El plugin FollowMe

Un plugin de detección de patrones se desarrolló con el mismo espíritu que el plugin FollowMe (Reconocimiento de Marcas, s.f.); el mismo permite a los usuarios crear bloques para detectar distintos patrones de una actividad externa. Un conjunto de bloques de señales de tránsito se incluyen con el plugin para usos didácticos (Ver Figura 21).

6.2 Convirtiendo una computadora portátil en un robot móvil

Las capacidades de actuación y sensoriales de una computadora portátil se pueden aumentar usando placas de entrada/salida de bajo costo. Mantener estas placas lo más abiertas posible es importante ya que uno de nuestros objetivos es facilitar la comprensión del estudiante de cómo funcionan los sensores y brindarles herramientas para hacer sus propios sensores con materiales reciclados y con piezas de desecho de computadoras. El acceso abierto se logra utilizando una arquitectura de capas para manejar la complejidad, la combinación del firmware en el robot y del software que se ejecuta en la computadora. Los estudiantes pueden construir sensores compatibles con la plataforma sin necesidad de tener una comprensión completa de los detalles de bajo nivel, mientras que los estudiantes que desean profundizar más pueden aprovechar la plataforma abierta.

Cuando se inició el Proyecto Butiá, se utilizó una placa de E/S Arduino. Pero Arduino es difícil de ensamblar a mano, por lo tanto, la elección de la plataforma estaba en desacuerdo con nuestro objetivo de que los niños participen en todo el proceso de la construcción de un robot como parte de su papel en convertirse en un desarrollador de plataforma. También se encontró que los costos unitarios totales se reducen significativamente mediante el uso de un diseño de microcontrolador que era específico para el robot Butiá (en lugar de utilizar un controlador de propósito general). En el USB4Butiá que siguió al proyecto original Butiá (Aguirre, 2012) migramos al microcontrolador PIC18F4550.

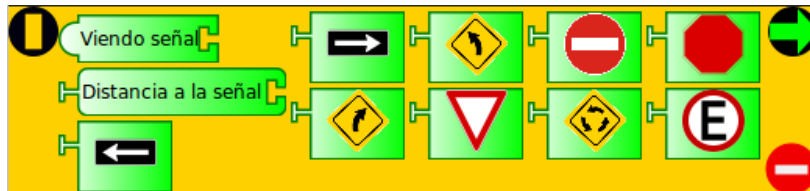


Figura 21: Plugin de señales Turtle Blocks (arriba) y su uso en un reto de enrutamiento (abajo) (foto cortesía de la Fundación Julio Ricaldoni)

USB4Butiá es una placa de E/S abierta y libre con interfaz USB desarrollada en el contexto de Butiá. Fue diseñada con el objetivo de que la reproducción de la placa fuera bastante simple y que pudiera hacerse sin la necesidad de equipos de montaje y fabricación caros, usando componentes disponibles en el mercado electrónico local del Uruguay (en muchos países en desarrollo es muy difícil acceder a la electrónica con tecnología de última generación). El circuito impreso de la placa USB4Butiá sólo tiene una capa, que hace que sea fácil de imprimir y transferir utilizando elementos simples que están disponibles, tales como una plancha para ropa que se encuentra en la mayoría de los hogares. A pesar de su simplicidad, USB4Butiá tiene características sofisticadas como sensores Hot-Plug y capacidades Plug-and-Play que liberan a los niños de la preocupación de muchos de los detalles que hacen la programación difícil para los

principiantes (el firmware que hemos desarrollado para USB4Butiá es fácilmente portado a otras arquitecturas, como Arduino, placas GoGo, entre otras.).

El diseño flexible del Proyecto Butiá permite a una variedad de usuarios (estudiantes o alumnos) participar en un nivel determinado por sus conocimientos e intereses, como ser: (1) estar motivados por los conceptos de informática y programación pura del comportamiento del robot; (2) mostrar interés en la programación de los dispositivos incorporados, disfrutando de la programación de microcontroladores y el acceso a los detalles de bajo nivel; (3) estar interesados en hacer sus propios sensores y actuadores; y (4) sólo tener acceso a la programación del robot utilizando herramientas de alto nivel, tales como Turtle Blocks.

6.3 Exploración de ideas con los robots

Turtle Blocks es compatible con diferentes plataformas de robótica; al momento de escribir estas líneas: Fischer LT, Lego WeDo, Lego NXT, Arduino, Butiá, Rodi y ExpEyes, lo que permite a los niños crear robots con opciones creativas y explorar ideas diferentes en función de las plataformas a las que tienen acceso y el problema que quieren resolver. También les permite explorar enfoques híbridos como la combinación de múltiples kits de sensores de diferentes proveedores (Ver Figura 22).



Figura 22: A la izquierda, compatibilidad mecánica Butiá con el kit de Lego NXT (fotografía cortesía de ANTEL); a la derecha, un robot Butiá hecho por los estudiantes, aumentado con un brazo hecho con un kit de Lego (fotografía de Luca Culeddu)

En el diseño de la plataforma Butiá, en cada punto de decisión nos preguntamos, ¿cómo logrará impactar la capacidad del alumno para utilizar las herramientas y componentes que serán fácilmente disponibles para ellos? Una consecuencia de las características abierta y libre de la plataforma Butiá, es que los estudiantes son capaces de crear robots Butiá utilizando la tecnología en desuso; en la wiki Butiá (Butiá Wiki, s.f.) muchas opciones de componentes reciclados han sido sugeridas por la comunidad Butiá. Algunos usuarios han ido tan lejos como para compartir sus propias versiones modificadas de la plataforma (Ver Figura 23)

6.4 Trabajo de campo Butiá

En setiembre de 2010, durante los talleres de robótica que organiza cada año la Facultad de Ingeniería de la Universidad de la República de Uruguay (Sumo.uy s.f.), los 28 primeros robots Butiá, fueron entregados a las escuelas secundarias de todas las

regiones del país . Desde ese contacto inicial hemos seguido trabajando con las escuelas secundarias a través de un curso de la Universidad para los estudiantes de ciencias de la computación e ingeniería eléctrica. Cada estudiante en la clase tiene que: (1) proponer y desarrollar nuevas características para el robot Butiá; (2) dar apoyo a una de las escuelas que ya tiene un robot Butiá, animándoles a participar en los nuevos desafíos; y (3) participar en talleres de robótica con los estudiantes de las escuelas públicas o escuelas secundarias en diferentes regiones del país.

Desde setiembre de 2011 cientos de estudiantes de escuelas primarias y secundarias en todo Uruguay han participado en desafíos de la robótica con un reto diferente ofrecido cada año (Sumo.uy s.f.). Los estudiantes participan con sus robots (ver Figura 24), algunos de ellos con Butiá y otros con kits comerciales. Los desafíos están diseñados para no dar ventajas basadas en las diferencias tecnológicas de los kits de robots. A pesar de la naturaleza competitiva de los problemas propuestos, los estudiantes colaboran, mientras solucionan los desafíos, compartiendo herramientas, sensores, ideas y enfoques para ayudar a resolver los problemas. Para promover aún más la colaboración entre los equipos, se ha establecido una nueva categoría en 2014 llamada Super Equipos; se lleva a cabo un sorteo para elegir los pares de equipos, donde los miembros de los mismos no se conocen entre sí antes de la competencia y el desafío es aquel en el que los robots deben cooperar a fin de llegar a una solución.

En noviembre de 2011 se realizó una encuesta entre los profesores de secundaria que habían estado trabajando con el robot Butiá durante al menos un año. Los encuestados informaron que el trabajo con el robot Butiá, había mejorado las habilidades de los estudiantes con sus computadoras portátiles y con sus conocimientos de programación. También se informó que los estudiantes consideran el trabajo con el robot Butiá divertido y motivador; que los ayuda a pensar desde diferentes puntos de vista y los estimula a hacer investigaciones. Los profesores también informaron que es muy fácil trabajar con el robot.

Se informó que la programación de los robots es atractiva y divertida para los estudiantes, motivándolos a aprender y a encontrar maneras de aplicar lo que ya habían aprendido en sus cursos tradicionales. Algunos maestros dijeron:

“Motiva mucho a los alumnos el trabajar directamente sobre el hardware y ver cómo el robot actúa de acuerdo a lo programado. También pienso que desarrolla el espíritu de investigación del alumno, el trabajo cooperativo, el aprendizaje en base a ensayo error.”

“...favorecer entornos de debate, intercambio de ideas, desarrollar en los alumnos la capacidad de solucionar problemas y resolver los desafíos a los que se enfrenta”

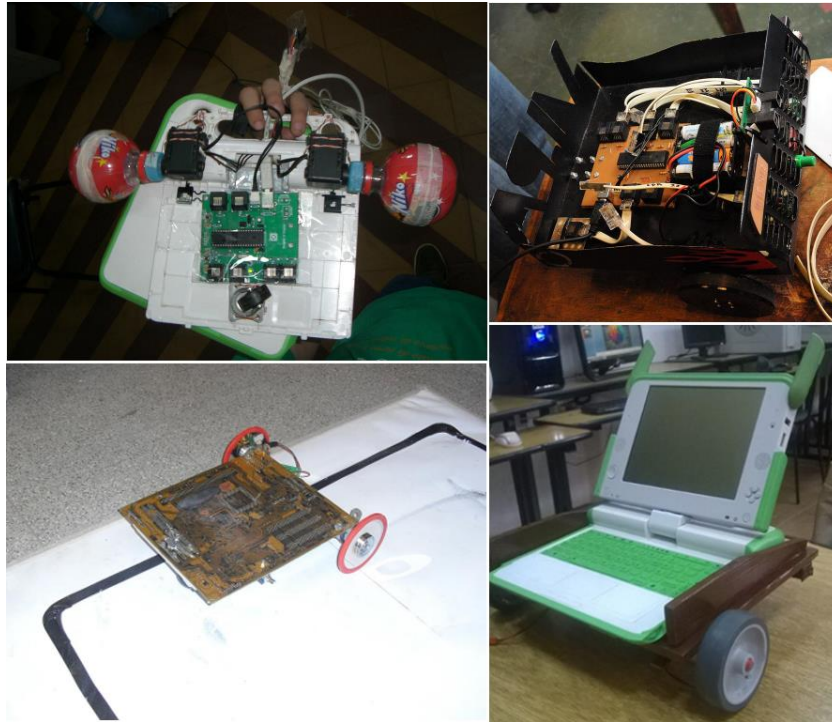


Figura 23 Diferentes robots Butiá caseros creados por los usuarios

La encuesta mostró algunos aspectos que deben mejorarse: los docentes informaron que un solo robot no es suficiente para un gran grupo de estudiantes. También informaron acerca del deseo de construir su propio Butiá; esta encuesta se utilizó como insumo para el desarrollo de una nueva versión del robot que se inició en el 2012; Butiá 2.0 (SAM) fue desarrollado para ser construido desde cero y puede ser lo suficientemente barato para apoyar un modelo de uno-a-uno. Todas las piezas de electrónica necesarias pueden ser construidas a mano, permitiendo la experiencia de construir para aquellos que estén interesados en el hardware; refuerza la idea de ser los desarrolladores plenos de la plataforma y no solamente usuarios.

Los estudiantes se apropiaron del robot, proponiendo varios proyectos por su cuenta. Uno de los más creativos era un robot Butiá utilizado para entregar los certificados de fin de año (Butiá proyectos de los estudiantes, s.f.). En la ceremonia de fin de curso, el director de la escuela puso el certificado en una mano de plástico adaptada al robot programado por los estudiantes, el que luego siguió un camino a lo largo de una línea negra pintada en el suelo para llegar al lugar donde los estudiantes estaban esperando.



Figura 24: desafío básico Sumo propuesto en 2013

Otro ejemplo de la apropiación es el trabajo de los estudiantes en la Escuela Tala (Tala, s.f.). Tomando la inspiración de los rovers de exploración de Marte, construyeron una versión modificada del robot Butiá (Ver Figura 22 derecha) diseñado para la recogida de diferentes muestras de terreno; su diseño incluyó la ampliación de su Butiá con piezas de otros kits comerciales. También diseñaron piezas personalizadas para la adquisición de dichas muestras, que se fabrican en una impresora 3D.

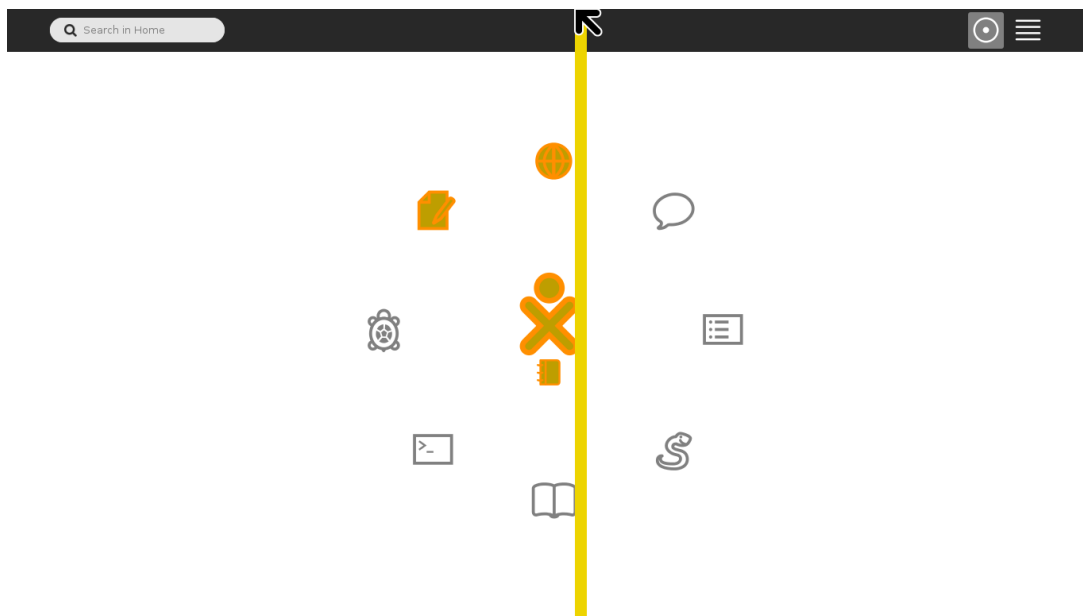


Figura 25: Una interfaz de barra de desplazamiento hecha con XEvents. Estudiantes con limitaciones motrices utilizan la interfaz para indicar las posiciones horizontal y vertical de la pantalla utilizando únicamente sus movimientos limitados.

En los últimos tres años nos hemos centrado en instruir maestros en el uso de los robots como una herramienta educativa; trabajamos con las instituciones, tanto en entornos educativos formales como informales. Mientras luchamos con la motivación de algunos docentes, otros están entusiasmados con el potencial que el robot lleva a sus salones de clase. A modo de ejemplo, en el año 2014 se formaron 200 educadores, los cuales construyeron su robot Butiá para trabajar como herramienta en sus clases, aprendiendo a programarlo y utilizarlo como elemento didáctico. Como consecuencia del trabajo de los docentes formados y otras actividades realizadas por la comunidad durante dicho año se alcanzó sensibilizar a aproximadamente 5500 estudiantes.

Recientemente, hemos estado trabajando con una docente que se especializa en niños con una variedad de discapacidades físicas; ella se enfrenta al problema de que sus alumnos no pueden utilizar el robot debido a la falta de una interfaz que se pueda adaptar a las habilidades de cada estudiante. Hemos desarrollado un nuevo plugin para Turtle Blocks llamado XEvents (ver Figura 25) que permite a los profesores diseñar prototipos de dispositivos de accesibilidad que pueden ser adaptados y diseñados para explotar los mecanismos motrices específicos de cada alumno mediante el uso de la variedad de sensores disponibles en los kits de robótica soportados en Turtle Blocks. El plugin agrega bloques para interactuar con el sistema operativo, con el fin de mover el puntero del ratón y generar eventos "clic" mediante programación con los programas de Turtle Blocks. Un miembro de 17 años de edad, de Python Joven es el desarrollador principal de este plugin y está trabajando con el equipo de la Universidad para poner este trabajo en producción.

7. Conclusiones

La programación con sensores y robots permite al estudiante pasar de lo abstracto a lo concreto: una idea abstracta, un diseño y, finalmente, un dispositivo funcional concreto. Los estudiantes son los que eligen sus propios proyectos. Cada uno de éstos se inicia desde sus conocimientos previos, la base de los nuevos conocimientos. Además, se desarrolla como un equipo, que promueve la construcción activa y colaborativa del conocimiento.

Mihaly Csikszentmihalyi descubrió que las personas se vuelven más profundamente comprometidos en actividades que son un reto, pero no insoportable (Csikszentmihalyi, 1997). Papert, al describir el concepto de "la diversión dura", observó que los alumnos se benefician más de las actividades que son "duras", siempre y cuando se conecten profundamente con sus intereses y pasiones (Papert, 1993). Es en su espíritu de compromiso auténtico que hemos presentado nuestro trabajo.

David Cavallo, en su artículo, "Modelos de crecimiento", argumentó que la "razón principal de la falta de cambio en la educación no se debe a la falta de ideas sobre el aprendizaje en un nivel micro o individual, sino más bien a una falta de modelos de crecimiento y cambio a un nivel macro o sistémica" (Cavallo, 2004). Cavallo describe las trampas del modelo "replicar y escalar", donde se impone un cambio totalmente concebido de arriba hacia abajo de forma jerárquica sólo para ser rechazado por los que están siendo alimentados a la fuerza, y el modelo piloto pequeño, en el cual rara vez se logra escalar. Hemos descrito un enfoque para el uso de las computadoras portátiles que no está predeterminado o prescrito. Más bien, su uso está destinado a ser emergente, donde el cambio es un subproducto del que los maestros, los estudiantes y sus padres,

aprenden juntos, descubriendo nuevas posibilidades y compartiendo esos descubrimientos.

Muchas preguntas continúan sin respuesta: ¿Qué habilidades adquieren los estudiantes? ¿Pueden estas habilidades transferirse a otros dominios de resolución de problemas? ¿Cómo podríamos determinar esto? Si bien somos ávidos partidarios de las plataformas abiertas, ¿son éstas demostrablemente mejores que las plataformas cerradas? ¿Hay alguna ventaja de trabajar con sensores y robots sobre la programación sin periféricos? ¿Es replicable nuestro trabajo? ¿Qué condiciones previas son necesarias para que lo sea? Suponiendo que las respuestas a estas preguntas lo ameriten, ¿cómo podemos hacer crecer la comunidad de aprendizaje basada en plataformas abiertas?

Nuestro objetivo en el desarrollo de la portátil XO, Butiá, Sugar y Turtle Blocks es intervenir en el enfoque adoptado para la educación. Si bien somos agnósticos sobre los planes de estudio, con la excepción notable de una insistencia en que cada niño debe aprender a programar computadoras, no somos agnósticos sobre el aprendizaje. Tratamos de organizar nuestro trabajo de tal manera que "los estudiantes estén aprendiendo, porque tienen una necesidad real de saber algo, [y] para que puedan utilizar este conocimiento para resolver un problema o responder a una pregunta que les importa a ellos" (Gold Standard PBL, 2015). Animamos a los profesores y alumnos a nutrir ambientes donde los estudiantes estén intrínsecamente motivados y donde el objetivo de la tecnología sea para "aprender a aprender".

Presentación del manuscrito: 10 de septiembre de 2015

Fecha de aprobación: 10 de septiembre de 2015

Fecha de publicación: 15 de septiembre de 2015

| |
|---|
| Guzmán, T. et al. (2015). Robótica Sensores Tortuga 2.0: Cómo el hardware y software abiertos pueden empoderar a las comunidades de aprendizaje Turtle. <i>RED. Revista de Educación a Distancia</i> , 46(5). 15 de Septiembre de 2015. Consultado el (dd/mm/aa) en http://www.um.es/ead/red/46 |
|---|

Referencias

Abelson, H. y diSessa, A. (1980). *Turtle Geometry*, The MIT Press.

Aguirre, A. (2012). USB4butia - A Truly Free (as in Freedom) Input/Output Board, from OLPC news website:
http://www.olpcnews.com/use_cases/technology/usb4butia_a_truly_free_as_in_freedom_input_output_board.html#sthash.79r0VaGn.dpuf

Arduino (s.f.). . [Página web]. Recuperado de <http://www.arduino.cc/> el 28 de Julio de 2015.

BeagleBone. (s.f.). . [Página web]. Recuperado de <http://beagleboard.org/bone> el 28 de Julio de 2015.

- Bender, W. (2015). *Guía de Programación con Turtle Art*
<https://github.com/walterbender/turtleblocksjs/blob/master/guide-es/README.md> 28 de Julio de 2015.
- Bender, W. et al. (2012). *Learning to Change the World: The Social Impact of One Laptop per Child*, Palgrave.
- Box-2D. (s.f.). [Página web]. Recuperado de <http://box2d.org> el 28 de Julio de 2015.
- Butiá Reconocimiento de Marcas. (s.f.) Recuperado de FING
https://www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/Butia_reconocimiento_marcas el 28 de Julio de 2015.
- Butiá Proyecto de estudiantes (s.f.) [Vídeo]. Recuperado de Youtube
<http://www.youtube.com/watch?v=Qvsp4dTU244> el 28 de Julio de 2015.
- Butiá Wiki. (s.f.). [Wiki]. Recuperado de FING
https://www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/Wiki_Buti%C3%A1 el 28 de Julio de 2015.
- Cavallo, D. (2004) 'Models of Growth' *BT Technology Journal* 22, no. 4.
- Csikszentmihályi, M. (1997) *Finding Flow: the psychology of engagement with everyday life*, HarperCollins.
- Dewey, J. (1938) *Experience and education*, Touchstone, (New York, NY.)
- ExpEyes (s.f.). [Página web]. Recuperado de <http://expeyes.in/> el 28 de Julio de 2015.
- Figari, P., (1965), *Educación y Arte*, (Prólogo de A. Ardao), Colección Clásicos Uruguayos, N° 81, Biblioteca Artigas. Recuperado de
http://figuras.liccom.edu.uy/_media/figari:obra:literatura:figari_pedro_-_educacion_y_arte_clasicos_uruguayos_n_81_1965_.pdf el 22 de Julio de 2015.
- FIRST Robotics Competition. (s.f.). [Página Web]. Recuperado de
<http://www.usfirst.org/roboticsprograms/frc> el 28 de Julio de 2015.
- GameMaker. (s.f.). [Página web]. Recuperado de <http://www.yoyogames.com/gamemaker> el 28 de Julio de 2015.
- GNU Licenses. (s.f.). [Página web]. Recuperado de Free Software Foundation (FSF)
<http://www.gnu.org/licenses/licenses.html#GPL> el 28 de Julio de 2015.
- GoGoBoard. (s.f.). [Página web]. Recuperado de <http://www.gogoboard.org/> el 28 de Julio de 2015.
- Gold Standard PBL: Essential Project Design Elements (s.f.) Recuperado de BIE.org
http://bie.org/object/document/gold_standard_pbl_essential_project_design_elements el 28 de Julio de 2015.
- Harvey, B. K. (1997), *Computer Science Logo Style*, 2 ed., Vol. 1-3, Cambridge, MA: MIT Press.

- Johnson, D., Johnson, R. y Holubec, E. (1999) *El aprendizaje cooperativo en el aula*, Buenos Aires, Paidós.
- Lepper, M., et al. (1997) "Intrinsic and Extrinsic Motivation: A Developmental Perspective," *Developmental Psychopathology: Perspectives on Adjustment, Risk, and Disorder*, Cambridge University Press.
- Martin, F. (2000) *Robotic Explorations: A Hands-On Introduction to Engineering*, Prentice Hall
- Mindstorms LEGO NXT . (s.f.) . [Página web]. Recuperado de <http://mindstorms.lego.com> el 28 de Julio de 2015.
- One Laptop per Child. (s.f.) . [Página web]. Recuperado de <http://laptop.org> el 28 de Julio de 2015.
- Papert, S. (1971) "Teaching Children to be Mathematicians vs. Teaching about Mathematics," *Artificial Intelligence Memo 249*, 3–4.
- Papert, S. (1993) *The Children's Machine*, Basic Books.
- Papert, S. and Solomon, C.(1971) "Twenty Things To Do With A Computer" *MIT AI Laboratory Memo 248*.
- Physics I: Classical Mechanics with an Experimental Focus* (2102) Recuperado de MIT Open Courseware website <http://ocw.mit.edu/courses/physics/8-01x-physics-i-classical-mechanics-with-an-experimental-focus-fall-2002/> el 28 de Julio de 2015.
- Picard, R., et al. (2004) 'Affective learning — a manifesto', *BT Technology Journal 22*, no 4.
- Plug-in (s.f.) [Wiki]. Recuperado de Wikipedia [https://en.wikipedia.org/wiki/Plug-in_\(computing\)](https://en.wikipedia.org/wiki/Plug-in_(computing)) el 28 de Julio de 2015.
- Proyecto Butiá Aprender jugando (2012) ROBÓTICA EDUCATIVA, *Revista Enlaces FING 8 6-10*. <https://www.fing.edu.uy/inco/proyectos/butia/files/Nota%20Buti%C3%A1%20-%20revista%20enlaces%208%20-%20fundaci%C3%B3n%20ricaldoni.pdf>
- Proyecto Butiá (s.f.) . [Página web]. Recuperado de FING <http://www.fing.edu.uy/inco/proyectos/butia> el 28 de Julio de 2015.
- Python. (s.f.) . [Página web]. Recuperado de Wikipedia <http://python.org> el 28 de Julio de 2015.
- Python Joven (s.f.) Recuperado de Google <https://sites.google.com/site/pythonjoven/> el 28 de Julio de 2015.
- Raspberry Pi. (s.f.) . [Página web]. Recuperado de [Raspberrypi.org](http://www.raspberrypi.org/) <http://www.raspberrypi.org/> el 28 de Julio de 2015.
- Resnick M, Berg R and Eisenberg M. (2000) 'Beyond black boxes: bringing transparency and aesthetics back to scientific investigation', *Journal of the Learning Sciences 9*, no. 1, 7–30.

- Rogoff, B. (1995) 'Observing sociocultural activity on three planes: participatory appropriation, guided participation and apprenticeship', in Wertsch J V, Rio P D and Alvarez A (Eds): *Socio-cultural Studies of the Mind*, Cambridge University Press, Cambridge, 139–164.
- Sanguinetti, J., (2002) *El Doctor Figari*. Ed. Aguilar. Montevideo.
- Smith, B., et al. (2000) "Silver Stringers and Junior Journalists: Active information producers" *IBM Systems Journal* 39, Nos 3&4.
- Snap! (s.f.) [Wiki]. Recuperado de Wikipedia [https://en.wikipedia.org/wiki/Snap!_\(programming_language\)](https://en.wikipedia.org/wiki/Snap!_(programming_language)) el 28 de Julio de 2015.
- Software Freedom Conservancy. (s.f.). . [Página web]. Recuperado de Wikipedia <http://sfconservancy.org> el 28 de Julio de 2015.
- Solomon, C., Minsky, M., Harvey, B. (1985) *LogoWorks: Challenging Programs in Logo*, Computing Mcgraw-Hill.
- Sugar Labs 0.102 Release Notes (2014) [Wiki]. Recuperado de Sugar Labs wiki <http://wiki.sugarlabs.org/go/0.102/Notes> el 28 de Julio de 2015.
- Sugar Labs 0.104 Release Notes (2015) [Wiki]. Recuperado de Sugar Labs wiki <http://wiki.sugarlabs.org/go/0.104/Notes> el 28 de Julio de 2015.
- Sugar Labs git repository (2015) . [Página web]. Recuperado de <http://github.com/sugarlabs> el 28 de Julio de 2015.
- Sugar Labs. (s.f.) . [Página web]. Recuperado de <http://sugarlabs.org> el 28 de Julio de 2015.
- Sumo.uy.(s.f.) . [Página web]. Recuperado de FING <http://www.fing.edu.uy/inco/eventos/sumo.uy/> el 28 de Julio de 2015.
- Tala school work (s.f.) . [Página web]. Recuperado de la Agencia EFE https://www.youtube.com/watch?v=cM_9-Bt2J0w&feature=youtu.be el 28 de Julio de 2015.
- Trinidad, G., Física con XO, (2013). [Página web]. Recuperado de <https://sites.google.com/site/solymar1fisica/fisica-con-xo-investigacion-/fisica-con-xo-el-libro> el 20 de Julio de 2015.
- Trinidad, G. (2013) Física con XO, Recuperado de http://sites.google.com/site/solymar1fisica/fisica-con-xo-investigacion_ el 28 de Julio de 2015.
- Turtle Art. (s.f.). [Página web]. Recuperado de <http://turtleart.org> el 28 de Julio de 2015.
- Turtle Blocks. (s.f.). [Página web]. Recuperado de Sugar Labs wiki <http://wiki.sugarlabs.org/go/Activities/TurtleArt> el 28 de Julio de 2015.
- Turtle Blocks (version de Javascript) (s.f.) <http://turtle.sugarlabs.org> el 28 de Julio de 2015b.
- TurtleBots. (s.f.). [Página web]. Recuperado de Sugar Labs wiki Sugar Labs Activity Portal <http://activities.sugarlabs.org/es-ES/sugar/addon/4434> el 28 de Julio de 2015.

USB4Butiá (s.f.). . [Página web]. Recuperado de FING

<https://www.fing.edu.uy/inco/proyectos/buti/mediawiki/index.php/USB4buti%C3%A1> el 28 de Julio de 2015.

Villela, J. (2006) *Ideas para enseñar a través de problemas*, Ediciones Espartaco, Uruguay.

WeDo LEGO Education. (s.f.). . [Página web]. Recuperado de <http://education.lego.com> el 28 de Julio de 2015.

XO Hardware Specification Page (2008) Recuperado de OLPC wiki

<http://wiki.laptop.org/go/Hardware> el 28 de Julio de 2015.

Zepf, M (2013) *Turtle Blocks Python Export Project* Recuperado de Sugar Labs wiki

http://wiki.sugarlabs.org/go/Summer_of_Code/2013/Turtle_Blocks_Python_export_project el 28 de Julio de 2015

Apéndice: 30 ideas para proyectos

Siguiendo la tradición iniciada por Papert y Solomon hace 30 años (Papert y Solomon, 1971), se describen a continuación varias ideas para proyectos con una breve descripción: (ver página siguiente)

| | |
|----|---|
| 1 | Dibujando Sonido: Uso del bloque <i>sonido</i> para cambiar el ancho del trazo del lápiz que dibuja. |
| 2 | Representación gráfica de la salida: Usando Turtle Blocks para graficar las lecturas de los sensores (cartesiana, polar, torta, barras, etc.) |
| 3 | Medida de temperatura: uso de un termoresistor para medir temperaturas (calibrándolo como un termómetro convencional) |
| 4 | Temperatura las 24 horas: Midiendo repetidamente la temperatura a lo largo de las 24 horas y graficando el resultado. |
| 5 | Medida de la humedad del suelo utilizando sondas de alambre conductor. |
| 6 | Medida de la salinidad del agua usando cables de cobre. |
| 7 | Generando electricidad mediante variaciones de campo magnético. Arrolle alambre aislado en torno a un clavo de hierro y frótelo frente a un imán de heladera (medida de voltaje). |
| 8 | Micrófono de carbón: construcción de un micrófono con una tapa de soda, carbón triturado, hojas de papel metálico y una banda elástica. |
| 9 | Pila de limón: con un clavo galvanizado y alambre de cobre. Medida de voltaje. (Conecte varios limones en serie) |
| 10 | Campanilla de timbre/alarma antirrobo: cuando dos alambres se tocan (cerrando un circuito) se emite un sonido. |
| 11 | Medida de corriente alterna (AC): construcción de un transformador de corriente con alambre de cobre. |
| 12 | Medida de Potencia. |
| 13 | Medida de corriente continua (DC). |
| 14 | Importar registro de datos en otras actividades: Puede registrar datos para analizarlos en otras actividades como hojas de cálculo. |
| 15 | Registro a intervalos regulares: registro de datos basados en un temporizador. |
| 16 | Aceleración en un plano inclinado: medida de la aceleración gravitatoria. |
| 17 | Uso de un resistor dependiente de la luz (LDR) para registrar el pulso colocando el extremo de su dedo entre el sensor y una luz brillante. |
| 18 | Medida de la salida de un panel fotovoltaico. |
| 19 | Medida de la constante de tiempo RC (resistor-capacitor) de un circuito. |
| 20 | Uso de XO como amplificador de audio. |

| | |
|----|---|
| 21 | Uso de XO como generador de señal de audio. |
| 22 | Construcción de un teletipo FSK para comunicación a distancia. |
| 23 | Captura de fotos a intervalos regulares de tiempo con la cámara. |
| 24 | Construcción de una alarma piroeléctrica que captura fotos de los intrusos. |
| 25 | Construcción de un juego de pasar la argolla por un alambre sin tocarlo. |
| 26 | Explorando campos magnéticos con sensores de efecto Hall. |
| 27 | Construcción de un ordenador de a bordo para bicicletas. |
| 28 | Construcción de un circuito cerrado de TV. |
| 29 | Construcción de una campanilla de timbre remoto. |
| 30 | Sus ideas van aquí... |