

# Butiá: Plataforma robótica genérica para la enseñanza de la informática

Gonzalo Tejera, Andrés Aguirre, Federico Andrade, Pablo Gindel, Santiago Margni y Jorge Visca  
{gtejeralaaguirrelfandradelpablodlsmargni|jvisca}@fing.edu.uy

Instituto de Computación, Facultad de Ingeniería, Universidad de la República J. Herrera y Reissig 565,  
Montevideo, Uruguay

**Resumen**—El aprendizaje de la robótica en los niveles iniciales de la educación es una herramienta poderosa para transmitir a los profesores, estudiantes y sus familias conocimientos básicos sobre las nuevas tecnología y sus aplicaciones. Existen muchos mitos sobre las computadoras y los robots, desconocimientos básicos tanto sobre lo que pueden como lo que no pueden hacer, en ambos sentidos, y que generan por un lado miedos infundados y por otro expectativas desmedidas. La incorporación de los robots y de la inteligencia computacional se está dando de manera progresiva en nuestra sociedad, y es importante entonces contribuir a mejorar el conocimiento sobre estas tecnologías. Este artículo presenta una plataforma simple y económica que permite a los alumnos de los liceos públicos del Uruguay interiorizarse con la programación de robots.

**Index Terms**—Educación inicial, enseñanza de informática y robótica.

## I. INTRODUCCIÓN

LOS robots pueden ser una herramienta pedagógica poderosa y flexible. Permite a los estudiantes realizar elaboraciones mentales de orden superior, reflexionar sobre el por qué de las cosas, experimentar e identificar las repercusiones de las decisiones que se toman y comprenderlas.

El proyecto Butiá[1] desarrollado con fondos de la Agencia Nacional de Investigación e Innovación[2] crea una plataforma que permite a alumnos de liceos públicos, en coordinación con docentes e inspectores del Consejo de Enseñanza Secundaria (CES)[3], interiorizarse con la programación del comportamiento de robots.

El proyecto provee a los liceos seleccionados por el CES una plataforma simple y económica que permite a los alumnos definir el comportamiento de un robot, que proporciona ambiente lúdico e idóneo para que los estudiantes se interioricen con la programación y la robótica.

Es importante señalar que, en cuanto a la robótica, existe actualmente una profunda asimetría entre liceos públicos y privados, extendiéndose de manera creciente la robótica en la currícula de muchas instituciones privadas. Esta propuesta pretende contribuir a disminuir esta brecha acercando este sistema robótico a instituciones públicas de nivel secundario en las que hasta el momento, este tipo de sistemas no ha sido ampliamente incorporado.

## II. LA ROBÓTICA Y LA EDUCACIÓN

Dada la creciente importancia que tiene la tecnología hoy en día y el amplio terreno que viene ganando la robótica

y la mecatrónica en la vida de las personas en el mundo desarrollado, resulta conveniente comenzar a incorporar estos conceptos en las primeras etapas de la formación educativa de nuestros jóvenes. En este sentido, es interesante ofrecer a alumnos de educación secundaria la posibilidad de acercarse a nuevas tecnologías a través del manejo de robots y lenguajes de programación simples que les permitan controlarlos. Se espera con esta experiencia que los alumnos dispongan de un elemento más para definir su vocación hacia orientaciones Científico – Tecnológicas.

Programar los comportamientos de un robot móvil genera mucho interés para los adolescentes. Además, les permite alcanzar resultados visuales inmediatos de sus programas, estimula su creatividad, así como al mismo tiempo se les enseñan conceptos básicos de programación. Se entiende que el trabajo con robots, desde una perspectiva de la robótica pedagógica potencia el desarrollo del aprendizaje inductivo y por descubrimiento guiado; posibilita el diseño de situaciones didácticas que permiten a los estudiantes construir su propio conocimiento. [4]

Esta propuesta busca generar entornos de aprendizaje centrados en la actividad de los propios estudiantes. Uno de los factores a destacar es la posibilidad de integración de las diferentes áreas, como matemáticas, ciencias experimentales, comunicación, filosofía, entre otras, que amplían la propuesta de trabajo a nivel de los centros educativos, posibilitando que se desarrollen proyectos de fin de año integrando varias asignaturas al trabajo con los robots, como se propone en el plan de estudios vigente de bachillerato.

Los alumnos de estos liceos podrán presentar los trabajos realizados sobre el sistema robótico en el marco del evento de robótica organizado por la Facultad de Ingeniería de la Universidad de la República[5], simultáneamente con trabajos realizados por alumnos y docentes universitarios, generando un rico ambiente de intercambio de experiencias y conocimientos.

El proyecto Butiá se apoya, en todo sentido, sobre las computadoras OLPC (One Laptop per Child)[6] proporcionadas al sistema de educación público del Uruguay a través del Plan Ceibal[7], llevado adelante por el gobierno nacional del Uruguay a partir del año 2007. En la Figura 1 podemos apreciar una de las computadoras del mencionado plan sobre la plataforma Butiá.



Figura 1. La computadora OLPC sobre la plataforma Butiá.

### III. ARQUITECTURA DEL LA PLATAFORMA BUTIÁ

Al momento de diseñar la arquitectura del sistema se tuvo como principal objetivo la portabilidad del mismo, ponderando plataformas con pocas prestaciones de hardware. Otro punto en el que se tuvo especial cuidado es en brindar una interfaz clara y simple que permita integrar la arquitectura del robot con diferentes lenguajes de programación.

Una vista simplificada de los principales componentes de la solución puede visualizarse en la Figura 2. La arquitectura Butiá consta de tres capas. Tomando en cuenta un enfoque botton-up, podemos identificar el nivel más cercano al hardware, la primer capa (capa 1), encargada de interactuar con los sensores y actuadores. En esta capa se especifican los servicios que el firmware brinda, además estos servicios son agrupados de forma lógica en componentes de software llamados *módulos de usuario*, los cuales son una abstracción del sensor o actuador que se desea controlar en el robot. Estos componentes de software residen en el firmware de la placa de entrada/salida (E/S). La separación entre esta capa y la siguiente está bien definida por un stack de protocolos, lo cual permite obtener un gran nivel de independencia del hardware subyacente. El prototipo construido es funcional con la placa de E/S Arduino y actualmente se presenta un gran nivel de avance con la placa USB4all [8]; además se espera portar a otro tipo de plataformas como Handy Cricket, PicoBoard o GoGoBoard. Existe una segunda capa (DSI - descubrimiento e invocación de servicios) que se encarga de descubrir de forma dinámica los módulos presentes en la placa de E/S junto con los servicios o funcionalidades que estos brindan y una vez descubiertos son expuestos para poder ser consumidos por una aplicación. Para permitir un uso más versátil al momento de integrarse con diferentes lenguajes de programación se realizó una tercer capa (capa 3), que ofrece estos servicios a un nivel mayor de abstracción lo que posibilita que sean invocados en la red. Esta última capa permite interactuar con el robot desde cualquier lenguaje de programación que tenga soporte de sockets.

La arquitectura construida es genérica, permitiendo una

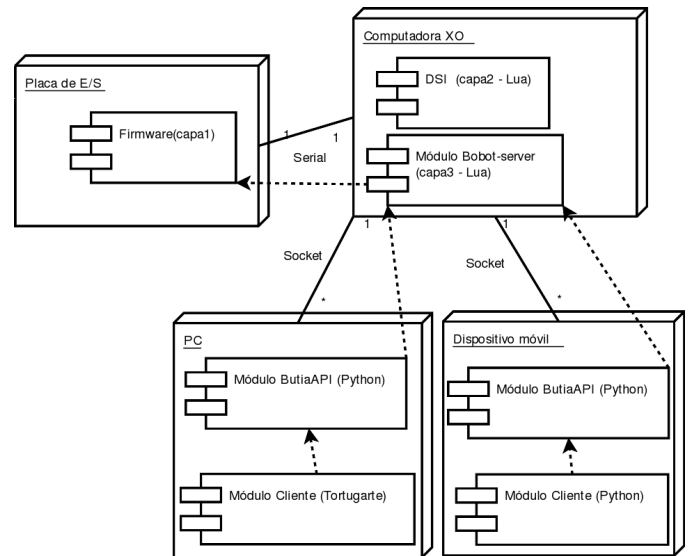


Figura 2. Componentes de la arquitectura Butiá.

fácil extensión de sus funcionalidades independientemente del hardware de bajo nivel subyacente, así como portable a diferentes plataformas de bajos recursos de hardware.

A diferencia de otras plataformas donde la lógica de control se desarrolla completamente en una placa de E/S microcontrolada, la arquitectura Butiá fomenta un diseño híbrido, donde solo se implementa en la placa de E/S la interacción directa con el hardware encapsulándolo en los módulos de usuario, luego la lógica de control es desarrollada en un lenguaje de alto nivel en base a los servicios que estos módulos exponen. Esto permite utilizar herramientas de mayor nivel de abstracción, generando código con un alto nivel de mantenibilidad y entendimiento, facilitando la comprensión por alumnos que se inician en las ciencias de la computación. En la Figura 3 un niño de 9 años trabaja en el comportamiento del robot Butiá.



Figura 3. Niño trabajando con la plataforma durante el sumo.uy.

La distribución del sistema operativo instalado en las computadoras OLPC incluye diversos entornos y lenguajes de programación. Entre ellos se encuentra el entorno de programación gráfico Tortugarte[9] basado en el lenguaje LOGO[10].

En el marco del proyecto se extendió el lenguaje agregando una paleta para interactuar con la plataforma Butiá. En la parte superior de la Figura 4 se muestra la paleta Butiá integrada al entorno Tortugarte que permite controlar los motores y acceder a la lectura de los sensores. En la parte inferior de la misma se presenta un comportamiento para evitar obstáculos utilizando el sensor Botón.

Además de esta extensión se desarrollaron APIs para los lenguajes de programación Python, C y Lua.

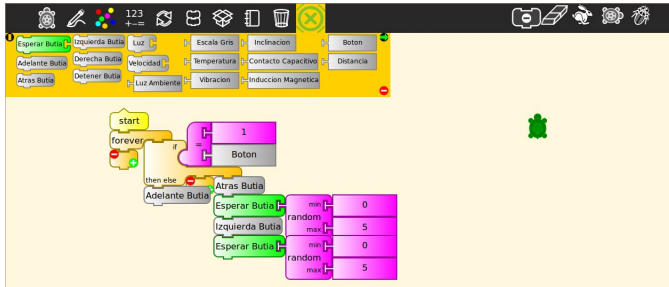


Figura 4. Evitando obstáculos con Tortugarte.

#### IV. IMPLEMENTACIÓN DEL PROTOTIPO

Con el objetivo de cumplir con los requerimientos de portabilidad, los componentes de software que tienen interacción directa con el hardware fueron implementados utilizando el lenguaje de programación ANSI C, esto permite lograr tener mayor control sobre el hardware y minimizar las latencias vinculadas con los tiempos de comunicación. La lógica de mayor nivel de abstracción y más propensa a cambios, se realizó utilizando el lenguaje interpretado Lua[11]. Lua es un lenguaje de gran nivel de abstracción, su máquina virtual es muy pequeña y esta desarrollada en ANSI C lo que permite portarlo fácilmente a diferentes arquitecturas, particularmente en aquellas con bajas prestaciones de hardware. Dado que es un lenguaje interpretado el software desarrollado utilizando su máquina virtual es sumamente versátil, evitando generar binarios para las distintas arquitecturas como es usual en otros lenguajes, lo cual es un gran beneficio al momento del desarrollo, simplificando el testing y puesta en producción sobre la plataforma objetivo. También existe una versión de Lua, llamada eLua pensada para sistemas con aún menos recursos como microcontroladores.

Se espera que Butiá soporte al menos las siguientes plataformas:

1. OLPC. Esta es la plataforma del Plan Ceibal. El hardware contiene un procesador *AMD Geode*, con arquitectura x86. Los primeros modelos poseían 256MB RAM actualmente ya disponen de 512MB. El disco duro es de estado sólido de 1GB. El software consiste en un kernel *Linux* con Sugar como interfaz de usuario.
2. Intel Classmate. Es la plataforma para enseñanza de Intel, y es un nombre genérico para una línea de productos de distintos fabricantes. El hardware es típico para un netbook de bajo costo: procesador *Intel Atom* (x86), a partir de 512MB de RAM, disco duro magnético.

3. Router Inalámbrico. Plataforma de costo y poder de cómputo mínimo prevista. Un ejemplo típico es un router *Asus 520GU*. Consiste en un sistema embebido con un procesador *Broadcomm*, con 16MB de RAM y 8MB de almacenamiento Flash. En el marco del proyecto se realizaron pruebas con *OpenWRT*, una distribución de *Linux* para sistemas embebidos.
4. Smartphones. Hay una gran cantidad de fabricantes y de plataformas de software. Usualmente contienen un procesador *ARM*, más de 64MB de RAM y almacenamiento flash. El sistema operativo puede estar basado en *Linux*, *Windows CE*, u otros sistemas dedicados.
5. Sistemas basados en *Single Board Computers* (SBC), se espera que la arquitectura funcione en la placa *Beagle-Board* y *Foxboard*, para esta última tenemos actualmente un prototipo funcional.

Se plantea la necesidad de disponer de un componente que pueda ser desplegado con mínimas modificaciones en todas las plataformas de interés, y que implemente las siguientes funcionalidades:

- Acceda a la placa de E/S implementando el protocolo USB4all sobre el tipo de conexión asociado (USB o Serial).
- Ofrezca una API que permita acceder las funcionalidades provistas por Butiá desde las aplicaciones de usuario.

Este componente se implementó en Lua como dos componentes: una biblioteca que implementa la comunicación con la placa (bobot), y una aplicación que usa esta biblioteca y exporta su funcionalidad mediante un socket (bobot-server). Esta arquitectura es la solución de referencia y de máxima portabilidad.

El bobot accede a la placa microcontroladora mediante USB o Serial. Para la primer opción, se enlaza con libusb, una biblioteca portable de espacio de usuario para manipular dispositivos USB. Este enlace se realiza mediante un binding desarrollado, llamado *luausb*[12]. Para acceder mediante serial se utiliza una pequeña librería en C que implementa una comunicación orientada a mensajes, llamada *serialcomm*. Bobot es fácilmente extensible para agregar soporte a nuevos módulos de la placa controladora (USB4all/Arduino). Esto se logra mediante drivers cargados dinámicamente de acuerdo a los módulos declarados por la placa controladora.

En las plataformas robóticas comercialmente más difundidas, el usuario debe llevar nota de qué dispositivo conectó en cada puerto de comunicación y declararlo en algún punto de la estructura del software. Esto difiere radicalmente respecto a la capacidad de auto-configuración o *plug&play* (PnP) de la plataforma Butiá. La idea es exponer al usuario las capacidades sensoriales y de actuación del robot Butiá, de una forma ordenada que facilite su aprovechamiento evitando etapas de configuración. Siguiendo estos principios se diseñó un conector PnP flexible, en el cual se puedan conectar tanto sensores como actuadores, analógicos como digitales, PWM e incluso I2C. Dicho conector dispone entre otros de pines de identificación los cuales se cablean a vcc o gnd directamente

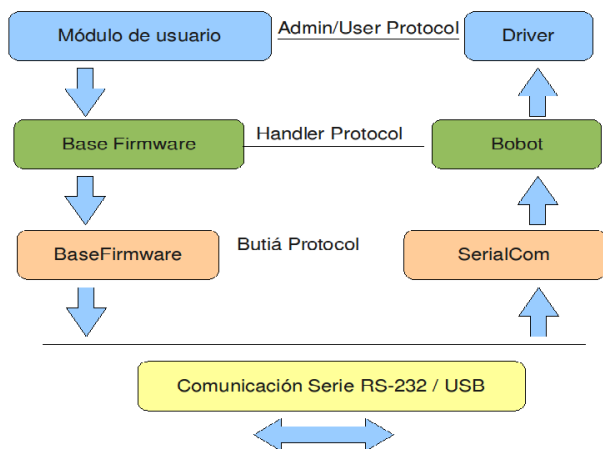


Figura 5. Interacción entre el sistema Butiá y la placa de E/S.

o a través de resistencias, logrando diferenciar el tipo de dispositivo conectado.

Ventajas del uso de los conectores PnP Butiá:

- Constituye una potente herramienta de diagnóstico.
- Simplifica el acceso al hardware.
- Permite que el firmware seleccione automáticamente el módulo adecuado para manejar cada tipo de dispositivo.

Se diseñó una placa adaptadora a la controladora de E/S que dispone de ocho conectores PnP Butiá, y en forma separada el conector para el bus Dynamixel (que también cuenta con auto-detección) y un conector de potencia para dos motores de continua o un motor de paso. Un ejemplo concreto de implementación es el que podemos ver en el Cuadro I

Cuadro I  
 ESPECIFICACIONES DEL ROBOT BUTIÁ.

Elemento	Instancia
Motores	Dynamixel AX12
Control de bajo nivel	Arduino Mega
Control de alto nivel	Computadora OLPC
Chasis	Acrílico 5mm
Amortiguación	Placa acero alto carbono 0.5mm
Sensores	Kit DFRobot para Arduino y sensores Sharp

Utilizando la información de PnP es posible desde la interfaz de Tortugarte colorear los elementos de la plataforma Butiá de diferentes colores dependiendo si se encuentra o no conectados al robot. Esta característica permite que los usuarios puedan detectar rápidamente errores de hardware, reduciendo las frustraciones generadas por este tipo de situaciones.

## V. TRABAJO A FUTURO

Actualmente se están logrando grandes avances con la integración de la cámara web de la XO como un sensor más, se han realizado pruebas de factibilidad que permiten identificar colores y retornar sus coordenadas. También se está trabajando en agregar el micrófono de la XO al universo de sensores. En este sentido se están realizando pruebas para poder procesar el lenguaje natural y permitir programar el robot mediante el uso de la voz. Otro aspecto interesante en el cual se está

investigando es en permitir compartir un robot entre varias máquinas brindando los mecanismos necesarios para arbitrar el acceso al mismo. Por último se están consiguiendo grandes avances en portar la arquitectura a plataformas embebidas tipo ARM como son *FoxBoard* y *BeagleBoard*.

## VI. CONCLUSIONES

Se diseñó una arquitectura que brinda un enfoque genérico e independiente del hardware de control. Esta arquitectura permite desarrollar el comportamiento del robot Butiá mediante lenguajes de programación de fácil acceso y comprensión para estudiantes liceales como es Tortugarte. De todas formas al estar modularizado y bien definido el alcance de cada capa, así como el API para acceder a cada una de ellas, existen diferentes niveles en los cuales el alumno puede desarrollar el comportamiento, dependiendo de su nivel de conocimientos, llevando esto a la programación de nuevos módulos de usuario en el firmware, así como programación en otros lenguajes como Lua, Python, Java. Actualmente se ha logrado un nivel suficiente de madurez en el sistema, se dispone de 30 robots en producción distribuidos en todo el país.

Desde el punto de vista social, a través de este proyecto se logra acortar la brecha para los que por razones económicas no pueden alcanzar dichas tecnologías. En este sentido se logra un proyecto con un fin democratizador.

## REFERENCIAS

- [1] Butiá, "Proyecto butiá," Agosto 2010, <http://www.fing.edu.uy/inco/proyectos/butiá/>.
- [2] ANII, "Agencia nacional de investigación e innovación," Febrero 2011, <http://www.anii.org.uy/web/>.
- [3] CES, "Centro de educación secundaria," Febrero 2010, <http://www.ces.edu.uy/ces/>.
- [4] S. Colorado, M. María, and A. Gauthier, "Ambientes de aprendizaje con robótica pedagógica," Ingeniería Eléctrica y Electrónica - Universidad de los Andes, Memo de Investigación IEL 2003-001, 2005.
- [5] sumo.uy, "Campeonato de sumo robótico," Agosto 2010, <http://www.fing.edu.uy/inco/eventos/sumo.uy/>.
- [6] OLPC, "One laptop per child," Agosto 2010, <http://laptop.org/>.
- [7] Ceibal, "Portal del plan ceibal," Agosto 2010, <http://www.ceibal.edu.uy/>.
- [8] A. Aguirre, P. Fernández, and C. Grossy, "Interfaz usb genérica para comunicación con dispositivos electrónicos," Diciembre 2007.
- [9] S. Labs, "Turtleart activity," Agosto 2010, <http://wiki.sugarlabs.org/go/Activities/TurtleArt>.
- [10] (2011, Febrero) Web site of logo foundation. [Online]. Available: <http://el.media.mit.edu/logo-foundation/index.html>
- [11] L. H. d. F. Roberto Ierusalimschy, Waldemar Celes, "The programming language lua," Diciembre 1993, <http://www.lua.org/>.
- [12] J. Visca and A. Aguirre, "Libusb binding for lua," Agosto 2010, <http://luaforge.net/projects/lualibusb>.