

# Programando y Pensando Robots en la Universidad y en Escuelas Secundarias.

Osella Massa, Germán<sup>1</sup>; Álvarez, Eduardo<sup>1</sup>; Useglio, Gustavo<sup>1</sup>; Luengo, Pablo<sup>1</sup>; Llanos, Emmanuel<sup>1</sup>; Sarobe, Mónica<sup>1</sup>; Russo, Claudia<sup>1</sup>

<sup>1</sup>ITT, Universidad Nacional del Noroeste de la Provincia de Buenos Aires

## Abstract

*El presente trabajo está dirigido a la realización de un robot de tipo móvil y de desplazamiento basado en rodamientos. Mediante un diseño incremental, se busca desarrollar modelos de complejidad creciente que podrán utilizarse como fuente de requerimientos para obtener, desde mecanismos y circuitos electrónicos cada vez más refinados, hasta algoritmos de programación cada vez más complejos a medida que se le exija al autómata más capacidad de resolución de problemas. Los modelos de baja complejidad podrán usarse, sin dudas, como una excelente motivación para la comprensión y aplicación de lenguajes de programación para estudiantes en su fase inicial, utilizando interfaces apropiadas. Por último, se intenta construir el robot con la máxima integración de partes nacionales.*

**Palabras Clave:** Robot, Enseñanza, Programación.

## Introducción

La construcción y el estudio de robots autónomos junto con la programación necesaria para conseguir que éstos se comporten de la manera buscada resulta ser una excusa más que atractivas para buscar motivar en niños y jóvenes la creación de estrategias para la resolución de problemas, cultivar el pensamiento estructurado y perseguir el desarrollo de su inteligencia lógica-matemática, al mismo tiempo que puede ser incentivo para el trabajo colaborativo, compartiendo experiencias a partir de desarrollos en grupo. Estos robots ofrecen un gran estímulo a la hora de enseñar matemáticas, mecánica, electrónica y programación. Desde la comprensión y construcción de la estructura física del robot, el entendimiento de la electrónica que lo controla, la utilización de distintos tipos de sensores analógicos y digitales

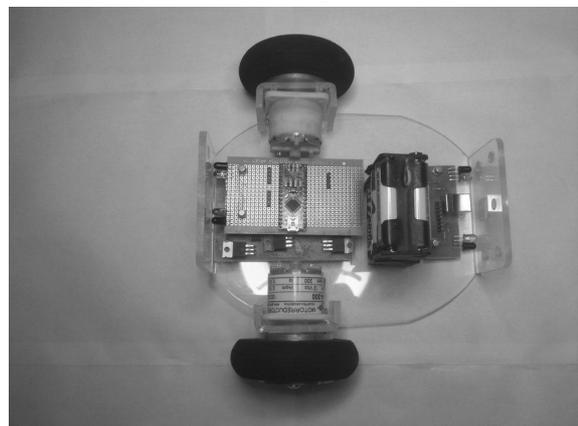
junto con los diferentes actuadores que pueden emplearse, pasando por cuestiones de comunicación, coordinación y control de los robots a través de diversos medios de transmisión de datos y sus correspondientes protocolos de comunicación, llegando a los aspectos relacionados con la programación de los mismos, tanto para comportamientos triviales como extremadamente complejos, puede observarse que el abanico de temas a tratar a partir del uso de los robots es sumamente amplio. Finalmente, no debe menospreciarse la sensación tras ver “cobrar vida” a un robot luego de invertir muchas horas en su armado, conexión y programación, la cuál entremezcla la satisfacción de alcanzar el objetivo buscado (indirectamente favoreciendo el desarrollo de las habilidades mencionadas) con el aspecto lúdico de interactuar con el robot una vez que éste comienza a funcionar. Por todo esto, resulta de interés contar con la posibilidad de construir y programar un robot que esté disponible a nivel nacional, de desarrollo abierto y libre, para ser utilizado en las escuelas por alumnos en los diferentes niveles. Nuestro objetivo entonces es poder construir ese robot destinado para uso didáctico, con la máxima integración de partes nacionales que combine, por un lado, la posibilidad concreta de poder ser armado y puesto en marcha por alumnos avanzados de carreras técnicas; y por otro lado, una vez que se cuente con el robot en servicio, brindar a alumnos de diversos niveles escolares distintas interfaces de programación acorde a las habilidades que se quieran desarrollar. La enseñanza comienza con el reconocimiento del robot, promoviendo el

acercamiento de los estudiantes al uso de lenguajes de programación en donde los docentes a cargo podrán descubrir en sus alumnos habilidades tempranas. El uso de diseños propios y flexibles otorga a los docentes la posibilidad de dotar a sus robots de características particulares acorde a sus necesidades.

### **Elementos del Trabajo y metodología**

Para el diseño del robot, se tomó como base a diversos modelos comerciales, siendo [1] un ejemplo concreto de ellos, adaptándolos a nuestras necesidades y buscando además cubrir los requerimientos extraídos a partir de las experiencias llevadas a cabo por un grupo de trabajo del LINTI de la Facultad de Informática de la UNLP, que desde hace unos años está trabajando con robots en escuelas de nivel medio [2]. Nuestro diseño se encuentra en constante evolución, comenzando con el descrito en [3]. El prototipo actual posee un chasis de policarbonato, cuyo espesor fue llevado progresivamente hasta los 4mm, suficientes para evitar deformaciones y permitir la incorporación por incrustación de todo tipo de anclajes de manera estanca y segura. Sobre el chasis se encuentran montados dos motores de corriente continua con sus respectivos anclajes y rodamientos de espuma de goma. Se utiliza una tercera rueda que es sólo de apoyo. El procesador utilizado es un Arduino Nano [4], basada en un ATmega328 de 8 bits funcionando a una frecuencia de reloj de 16 MHz, contando con 14 pines de E/S digital, 8 de E/S analógica, 30 KB de memoria flash, 2 KB de memoria SRAM y 1 KB de memoria EEPROM. Se agregan al diseño cuatro sensores IR de proximidad, dos sensores IR orientados para funcionar como seguidor de línea y dos sensores lumínicos junto a una cámara digital con capacidad de adquirir imágenes RGB de 640x480 pixeles. Todo lo anterior se completa con un módulo bluetooth para conexión inalámbrica, que permite la comunicación con cualquier otro dispositivo que implemente ese estándar.

En cuanto a las baterías que alimentan al robot, se seleccionaron elementos de formato AA y del tipo NiMH, pudiendo extraerse el pack completo directamente para su reemplazo ó recargarlo dentro del mismo robot mediante un cargador incorporado. En cuanto a los motores, dentro de la industria nacional encontramos impulsores de corriente continua con reducción de marcha incorporado que resultaron de calidad aceptable pero con peso y dimensiones poco apropiados para un diseño miniaturizado. Las ruedas son de diseño y construcción propios ya que no se encuentran en el mercado ruedas apropiadas para éste uso. El diseño permite cambiar diámetros y modelos de forma sencilla, pudiendo incluso ‘personalizarlas’ según los requerimientos del usuario. Se ha dotado al robot de dos niveles de acceso, uno para su uso vía bluetooth y otro para su programación y puesta en marcha vía medio físico cableado con puerto de conexión mini-USB. La tecnología electrónica, el diseño industrial y la selección de materiales es la que mejor se adapta a las posibilidades de nuestro país. Dado el fin último del proyecto, hemos decidido abrir a la comunidad educativa el diseño del aspecto exterior del robot, realizando un concurso de diseño para tal fin, el cual se encuentra en ejecución y aún no se dispone de un diseño finalizado. La apariencia actual del prototipo es la que muestra en la figura 1. La elección de las baterías y los motores fueron puntos críticos que se tuvieron en cuenta ya que existe una relación directa entre el peso final del robot,



**Figura 1:** Apariencia del actual prototipo.

la potencia de los motores y la capacidad de las baterías, donde a mayor peso se requiere de motores más potentes y por ende, de baterías de mayor capacidad, aumentando así el peso del robot. Se intentó encontrar un equilibrio adecuado entre autonomía y capacidad de movimiento de acuerdo al fin que se le quiere dar a este robot. En cuanto al software, se cuenta con dos sistemas diferentes pero comunicados entre sí. Por un lado, se programó para el propio robot un sistema de control que se ejecuta sobre el Arduino Nano, que permite comandar los actuadores e inspeccionar el estado de los sensores del robot a través de un protocolo de comunicaciones empleado sobre el perfil SPP (Serial Port Profile) de bluetooth [5]. Además de un control de bajo nivel de los actuadores, el sistema provee un control de más alto nivel, obedeciendo órdenes como “girar a la derecha” o “rotar”, para mencionar algunos ejemplos. Estas operaciones de control permiten dos modos de ejecución: indefinidamente o durante un cierto lapso de tiempo. Así, puede indicársele al robot que avance durante dos segundos y se detenga o simplemente que avance en forma continua. Este sistema de control está pensado para ser controlado por otro sistema corriendo sobre una computadora operada por los alumnos. Se desarrolló una biblioteca que provee el acceso a los robots, permitiendo detectarlos (siempre que se encuentren dentro del alcance de la transmisión por bluetooth), conectarse y ejecutar las primitivas de movimiento que cada robot entiende. Además, permite obtener el estado de los sensores que el robot posee, incluyendo la adquisición de imágenes en RGB. Dicha biblioteca está escrita para ser usada desde el lenguaje de programación Python, particularmente, desde cualquier versión de CPython de la rama 3.x [6]. La elección de Python como lenguaje de programación no es caprichosa: Es un lenguaje simple y poderoso sobre el cual los alumnos podrán escribir sus programas para controlar a uno o más robots. El lenguaje fomenta la escritura de código claro, incorpora

construcciones de muy alto nivel y permite realizar un descubrimiento exploratorio de las distintas herramientas que provee por la razón de que es un lenguaje interpretado y cuenta con un REPL (Read-Eval-Print Loop o intérprete de línea de comandos) en donde el alumno puede escribir una sentencia y observar inmediatamente las consecuencias de ejecutarla, sin necesidad de realizar el típico ciclo de edición/compilación/prueba/depuración. Existen varias experiencias positivas en donde se utiliza a Python como el lenguaje con el que los alumnos tienen su primer contacto con la programación. En la UNNOBA ya se lo está utilizando para la enseñanza de programación imperativa en el primer año de las carreras de informática. Para poder controlar al robot y observar el estado de los sensores mediante una interfaz visual, que resulta de gran utilidad al momento de depurar un programa en ejecución o poder hacer una parada de emergencia luego de observar un comportamiento anómalo, se ideó una herramienta con una GUI (Graphical User Interface) para tal fin, la que se implementó usando la biblioteca PyQt [7]. Esta herramienta puede ser utilizada en paralelo al programa escrito por el alumno, actuando como monitor del robot y con la capacidad de intervenir sobre los comandos que el robot está ejecutando en el caso que se lo requiera. Para esto, se ideó una arquitectura cliente-servidor, donde existe un servidor que se encarga del diálogo con el robot a través de la interfaz bluetooth y tanto la biblioteca de Python como el programa de control y monitoreo actúan como clientes de ese servidor. Por un lado, este nivel de indirección resulta necesario ya que por cuestiones técnicas inherentes al perfil SPP de bluetooth, sólo es posible que un dispositivo a la vez se conecte con el robot. Esto imposibilita que tanto el programa monitor como el que el alumno escribe puedan interactuar a la vez con el mismo robot. Con el esquema planteado, el servidor permite multiplexar el canal de datos, de forma que un número arbitrario de programas puedan interactuar al mismo con

el robot. Además, el servidor se encarga de coordinar la ejecución de los comandos enviados al robot, de forma que no se envíen comandos contradictorios simultáneamente. También permite reducir la cantidad de comunicación entre la computadora y el robot al momento de conocer el estado de los sensores, puesto que como no tiene sentido muestrear un sensor a más de una determinada frecuencia, cuando dos o más programas intenten conocer el estado de dicho sensor, el servidor puede contar con un caché que almacene el último valor medido y si aún no transcurrió el tiempo suficiente como para realizar otra medición, simplemente informa el último valor sin incurrir en tiempos y costos asociados a la comunicación con el robot. También se explota el hecho de que en una misma trama de bluetooth puede enviarse el estado de todos los sensores a bordo del robot, por lo que si un programa solicita el valor de un sensor e inmediatamente solicita el valor de un segundo sensor diferente, no implicará dos encuestas al robot sino que con un único intercambio de información se podrá informar ambos datos. En cuanto a la comunicación entre los clientes y el servidor que corren sobre la computadora, se utilizó la biblioteca ØMQ [8], que permite simplificar notablemente la interacción entre las distintas partes del sistema, brindando la capacidad de comunicación punto a punto en un ciclo de pedido/respuesta así como la de distribuir un mismo dato a un número arbitrario de clientes usando un esquema PUB/SUB (Publish/Subscribe). Esto permite que en un futuro se implementen nuevas interfaces visuales para comandar al robot utilizando el actual servidor para comunicarse con el robot, sin necesidad de tener que implementar nuevamente toda esa funcionalidad en el nuevo software. La elección de las bibliotecas y herramientas usadas no es fortuita; se buscó que sean multiplataformas y todas con licencias permisivas. El software producido en este desarrollo junto con los esquemas

necesarios para la construcción del robot será publicado bajo una licencia permisiva.

## **Resultados**

Del resultado de las pruebas hechas al robot han surgido las modificaciones al mismo. Se ha ido cambiando su diseño buscando robustez, mínimo peso, máxima autonomía, bajo costo, materiales constructivos acordes con las disponibilidades de plaza local y proveedores de partes confiables. Elementos tales como el chasis, la ruedas tractoras y de apoyo y el pack de pilas son mejorados permanentemente en función de obtener los objetivos recién mencionados. De la interacción con el grupo de trabajo del LINTI se acordaron tres etapas de construcción individualizadas según la cantidad de sensores disponibles, siempre montados sobre la misma plataforma. En una primera etapa se prevé la incorporación de los sensores considerados como indispensables: un sensor de obstáculos al frente, un sensor de obstáculos en la parte trasera, dos sensores detectores de línea y dos sensores de luz. Para una segunda etapa, se aumentará la cantidad de sensores para detectar obstáculos a los costados, contando con tres sensores al frente y tres sensores atrás. También se agregará un cuenta-vueltas para las ruedas, de forma que se pueda refinar el control sobre los motores para lograr movimientos más precisos, junto con una cámara digital para capturar imágenes que luego puedan ser procesadas en la computadora. Para una tercera etapa, se reemplazaría uno de los sensores de obstáculos (el frontal del medio) por un sensor ultrasónico o similar para poder tomar medidas de distancia con precisión. También se agregaría un micrófono para adquirir muestras de audio. Cabe resaltar que si bien cada etapa propuesta aumenta las capacidades del robot, también se traduce tanto en un costo económico asociado a los componentes que se agregan así como también la complejidad del ensamblado de las partes y el consumo de

energía por el aumento del número de componentes involucrados.

### **Discusión**

Se plantea permanentemente la disyuntiva entre la miniaturización y la visibilidad de cada una de las partes componentes. Por razones pedagógicas se trata siempre de individualizar claramente cada sección del robot, muchas veces sacrificando costos y miniaturización. Un ejemplo es el del ‘puente H’, que puede incorporarse en forma integrada ó puede ser construido con todas sus partes en forma discreta para una mejor comprensión de su funcionamiento. Queda pendiente explorar maneras alternativas de controlar al robot, alejándose de la forma tradicional de escribir un programa con instrucciones en un lenguaje de programación para quizás integrar ideas de lenguajes visuales como lo hace el kit de LEGO Mindstorms [9]. También pueden implementarse nuevas bibliotecas de comunicación con el robot para otros lenguajes de programación, siempre que existiera ese requerimiento.

### **Conclusión**

Quizás el mayor logro en el diseño, construcción y puesta en marcha del robot sea la fácil adaptación a los cambios de materiales y tecnologías constructivas manteniendo inalterables a las interfaces dado lo imprevisible del suministro de materiales. En cuanto al desempeño, se han obtenido resultados similares al modelo comercial de referencia. El LINTI ha planteado incluso una serie de limitaciones del modelo original que han sido corregidas.

### **Referencias**

- [1] <http://www.parallax.com/tabid/455/Default.aspx>
- [2] “Programando con robots”, LINTI, UNLP, <http://robots.linti.unlp.edu.ar/>
- [3] “Programando Robots en Escuelas Secundarias”; Osella Massa, Alvarez, Useglio, Sarobe, Russo; V Congreso Nacional de Extensión Universitaria; Córdoba, Argentina, 2012.
- [4] <http://arduino.cc/es/Guide/ArduinoNano>
- [5] Serial Port Profile: [https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc\\_id=8700](https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=8700)
- [6] <http://www.python.org/>
- [7] <http://www.riverbankcomputing.co.uk/software/pyqt/intro>
- [8] <http://zeromq.org/>
- [9] <http://mindstorms.lego.com/>

### **Datos de Contacto:**

*Claudia Russo. Universidad Nacional del Noroeste de la Provincia de Buenos Aires, [crusso@unnoba.edu.ar](mailto:crusso@unnoba.edu.ar)*