

## Mendieta, un Robot por Escuela: Robot de Uso Concurrente para la Educación Tecnológica

Gonzalo Zabala<sup>1</sup>, Ricardo Morán<sup>1-2</sup> y Matías Teragni<sup>1</sup>

<sup>1</sup> Universidad Abierta Interamericana, Centro de Altos Estudios en Tecnología Informática, CABA, Argentina

<sup>2</sup> Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Calle 526 e/ 10 y 11, La Plata, Buenos Aires, República Argentina

{gonzalo.zabala, ricardo.moran, matias.teragni}@uai.edu.ar

**Abstract.** En los últimos años ha crecido el uso material concreto tecnológico en las escuelas de Argentina. La inserción de la robótica en el aula permite abordar un conjunto de contenidos vinculados con la tecnología, como aspectos mecánicos, electrónicos y de programación. No es casual que el sistema educativo argentino haya incorporado en su diseño curricular contenidos vinculados a la disciplina, como el surgimiento de una nueva orientación técnica dirigida a la programación y la robótica. Sin embargo, los costos de equipamiento, la complejidad de mantenimiento, los conocimientos técnicos necesarios para su uso y un correcto aprovechamiento didáctico del recurso son limitantes para que la robótica llegue a todas las escuelas. Por estos motivos se ha diseñado un proyecto que supere estas dificultades: un robot autónomo de bajo costo controlado mediante un Arduino Nano, con un servidor web incorporado en una computadora Orange Pi, que permite la programación de todo un curso al mismo tiempo, encolando los pedidos de ejecución de manera tal que con un único dispositivo se pueda llevar adelante la clase en forma dinámica. Esto permite introducir la robótica con material concreto en una escuela con un valor menor a los USD 180. Por otra parte, todo el desarrollo es de hardware y código abierto, ofreciendo a la comunidad un framework para el surgimiento de otras propuestas con la misma arquitectura.

**Keywords:** Educación Tecnológica, Robótica, Servidor Web, Arduino, Orange Pi

### 1 Introducción

Desde el año 2010 ha comenzado, en forma cada vez más masiva, la incorporación de robots en las aulas argentinas como recurso didáctico y ha crecido el interés de incorporar la disciplina a nivel curricular junto con la programación (Ripani, 2017). Podemos mencionar entre otros al proyecto “Todos a la robótica” en la provincia de San Luis (Munizaga, 2013), que incorporó desde 2011 hasta 2014 material, capacitación y un conjunto de actividades en todas sus escuelas primarias. Otro proyecto digno de

mención es “Aprender conectados” (primariamente denominado Escuelas del Futuro, 2017) que alcanzó a unas 3000 escuelas de todo el país, tanto a nivel primario como secundario, con materiales de mecánica, electrónica y robótica, así como capacitaciones y actividades para el uso en el aula (Ministerio de Educación – Presidencia de la Nación, 2017). En la Provincia de Buenos Aires encontramos su programa de Robótica (2018) que también alcanza a sus escuelas de nivel primario y secundario con diferentes tipos de kits (Banchoff Tzancoff et al., 2019). Finalmente, en el año 2018 el Ministerio de Educación de la Nación definió por primera vez objetivos de aprendizaje para la educación obligatoria en programación y robótica, para iniciar el camino de su inclusión formal en las propuestas de enseñanza y aprendizaje en las escuelas de la República Argentina (Ripani, 2017).

Más allá de la tarea realizada y el crecimiento del interés en la disciplina, el sostenimiento en el tiempo de la actividad es una característica muy difícil de lograr. Los materiales físicos sufren un desgaste natural en su uso (lo que no ocurre, por ejemplo, con computadoras o con software), además de que tienen un nivel de obsolescencia acelerado. Por otra parte, las políticas públicas vinculadas con la inserción de tecnología en las escuelas se modifican en cada gestión cuando se produce un cambio de partido en el gobierno. En conclusión, no alcanza con una compra y capacitamiento inicial: es necesario tener recursos económicos para renovar periódicamente el material y las actividades, y un apoyo permanente de parte de la gestión responsable de los recursos económicos y humanos.

El objetivo del proyecto “Mendieta, un robot por escuela” es facilitar la incorporación de robótica en las escuelas secundarias del país y su sostenimiento en el tiempo. A la luz de los problemas descriptos en el párrafo anterior, este proyecto debe cubrir los siguientes aspectos:

- El desarrollo del robot debe ser bajísimo costo y altamente reusable y actualizable.
- La construcción del robot debe ser sencilla e implementable directamente en la escuela.
- El hardware y software utilizados debe ser abierto para construir una comunidad de mantenimiento y creación de nuevos modelos.
- Es necesaria la construcción de una comunidad docente que aporte secuencias didácticas centradas en Mendieta y sus derivados.

Con respecto a la solución de bajo costo para la robótica educativa, se pueden encontrar diversas propuestas en la literatura (Aroca et al., 2012; Darrah et al., 2018; Junior et al., 2013; Lopes Filho et al., 2011; López-Rodríguez & Cuesta, 2016; Rubenstein et al., 2012; Saleiro et al., 2013). En todos los casos implica la necesidad de un robot por grupo de estudiantes y en general, la incorporación de dispositivos externos como celulares. Por otro lado, la electrónica que proponen es demasiado compleja para ser resuelta por docentes sin conocimientos de electrónica. Para reducir aún más el costo, y cumplir con los objetivos mencionados, Mendieta ha sido diseñado como un robot con

un servidor web montado en él, utilizando componentes estandarizados y de fácil conexión. De esta manera, como describiremos más adelante, con un único robot (o con una pequeña cantidad de ellos) todo el curso puede estar trabajando en forma simultánea.

En particular, para la electrónica del robot, se utilizan componentes que se consiguen fácilmente en el mercado nacional y de bajo costo. Con respecto a su estructura, se proporcionan modelos en formato STL para ser impresos en cualquiera de las impresoras 3D más económicas, dado que no superan su tamaño límite. Tanto la electrónica como el diseño 3D tiene licencias Creative Commons amplias.

En relación con la construcción de una comunidad de práctica, se desarrollará una plataforma donde los docentes pueden consultar actividades clasificadas por nivel de complejidad y disciplina transversal asociada. También podrán proponer sus secuencias a partir de un formulario plantilla, que luego será moderado por nuestro grupo para su publicación final.

A continuación, se presentarán las propuestas de solución para las cuestiones de orden técnico. En la primera parte haremos un análisis de diversas arquitecturas de hardware de robótica educativa, finalizando con una descripción de la elegida para Mendieta. Y en la segunda parte haremos foco en la arquitectura del software, tanto en el firmware del robot como en la plataforma de programación montada en el servidor web.

## 2 Arquitectura del Hardware

Para reducir el costo de materiales podemos considerar diversas opciones. Por un lado, el material de robótica no está en uso permanente en las aulas. Por lo tanto, el set debe poder circular entre curso y curso. Esto deja de lado kits que impliquen procesos constructivos que superen la hora de clase en la semana, dado que de esta manera el material quedaría reservado para un grupo en particular.

Al mismo tiempo debe tener la versatilidad de poder presentar problemas de diversa complejidad. Con este fin, existen diversas propuestas de robots de bajo costo que cumplen con estas premisas. Una de ellas es Cellbots (Aroca et al., 2012) que utiliza un celular como unidad de procesamiento. Propone el uso de la entrada de micrófono para el control de los motores y lectura de sensores externos, y el uso de los sensores del propio celular para enriquecer la captación del ambiente. Otra propuesta en el mismo sentido es Andruino - AI (López-Rodríguez & Cuesta, 2016), donde se incorpora una placa Arduino Uno como dispositivo para conexión de sensores y vinculación con el smartphone. En ambos casos, el modelo sigue siendo de un robot por equipo, aunque la forma de comunicación propuesta en este último proyecto permitiría un uso similar al ofrecido por Mendieta. De todas formas, en el costo no está considerado el celular, porque lo pondrían los estudiantes, con el consabido riesgo de accidentes en robots móviles.

Otras opciones como las que encontramos en (Junior et al., 2013; Lopes Filho et al., 2011; Rubenstein et al., 2012) tienen una complejidad constructiva alta, y mantienen el modelo de un robot por grupo de estudiantes.

Por estas razones, entendemos que la menor cantidad posible de materiales en un curso es el de un único robot. Con este objetivo, el robot propuesto no solo tiene la responsabilidad de coordinar sus sensores y actuadores en función de los programas desarrollados por los estudiantes, sino que también debe poder recibir, interpretar y coordinar el trabajo concurrente de dichos estudiantes. Para permitir el acceso concurrente para programar, compilar y ejecutar programas, los estudiantes acceden mediante sus propias computadoras o celulares a una página web montada sobre el robot, que oficia como servidor local, y a través de la cual se puede programar, depurar, solicitar la ejecución, y monitorear el estado del dispositivo.

Dado que las capacidades limitadas de los microcontroladores y el hardware utilizado para acceder a los sensores y actuadores resultan incompatibles con los requisitos establecidos para proveer los servicios necesarios, se puede utilizar hardware especializado y de mayor costo, o complementar al microcontrolador con un microprocesador que tenga el poder de cómputo necesario para funcionar como servidor web. **Cuando nos referimos tanto a microcontroladores como microprocesadores nos referimos a placas integrales que complementan los micros correspondientes con la electrónica y firmwares necesarios para una aplicación sin necesidad de conocimientos técnicos complejos.**

En tal sentido, existen microprocesadores conocidos y de relativo bajo costo que tienen acceso a sensores y actuadores, pero que no cuentan con un conversor analógico - digital. Por lo tanto, si se desea leer sensores analógicos, como por ejemplo de luz, humedad, distancia, etc., se debería agregar hardware adicional y perder la mitad de los puertos de acceso que se dispone para sensores. Además, esta solución implica un desarrollo electrónico de mayor complejidad

En base a lo expuesto, el diseño del robot cuenta con dos placas de procesamiento. Por un lado, un microprocesador donde se encuentra el servidor web que les permite a los docentes monitorear y configurar al robot, y a los estudiantes programar, depurar y ejecutar código. Por otra parte, cuenta con un microcontrolador encargado de realmente ejecutar el programa de los estudiantes, accediendo a los actuadores y sensores que tenga conectado.

Al respecto, existen múltiples opciones de hardware que pueden cumplir con las tareas delineadas, pero dada la naturaleza de este proyecto, que busca ser open-software y open-hardware, la facilidad de acceso a los componentes requeridos para el ensamble es un factor de peso. Para la selección particular se priorizó el uso de componentes estándar en la comunidad educativa y hobbista a nivel mundial. Por lo tanto, se eligió un Arduino como microcontrolador, siendo una plataforma abierta, de bajo costo y amplia aceptación de la comunidad mundial, y una Orange Pi como microprocesador, que cuenta con una amplia disponibilidad y documentación.

En particular, con respecto al microcontrolador se eligió Arduino Nano (Arduino Nano | Arduino Official Store, s/f) dado que resulta una de las versiones de la placa más utilizada en el país actualmente, y que puede ser reemplazada por otras versiones con un mínimo trabajo en caso de ser necesario. Es de menor costo y consumo que Arduino Uno, con prestaciones de orden similar, suficientes para el presente proyecto.

Al mismo tiempo, como microprocesador se eligió el modelo Orange Pi Zero (orange pi zero - OrangePi, s/f), por su menor consumo. A pesar de ser menos popular

que la Raspberry Pi, tiene mayor capacidad de procesamiento por un costo similar, lo que permite una mejor performance.

Hasta aquí se ha definido la base mínima de la arquitectura del hardware. El objetivo es que a partir de este framework se desarrollen diferentes propuestas de robots. A continuación, describiremos un primer prototipo con el cual haremos las primeras experiencias en aula.

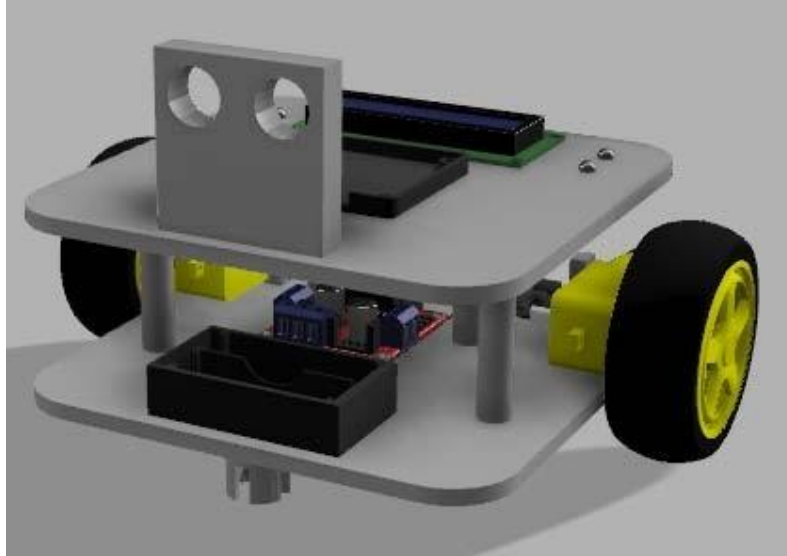
En relación con los sensores, se seleccionaron dos tipos, de bajo costo, alta presencia en el mercado y múltiples usos posibles. Por un lado, el ultrasónico HC-SR04 (Freaks, 2016) que permite medir distancias con un rango de 2 a 400 cm con una precisión en el orden de los 3 mm. Por otra parte, se suman dos módulos TCRT5000 que ya poseen la electrónica complementaria incorporada para conectar con el Arduino. El TCRT5000 (Semiconductors, 2009) es un sensor óptico reflexivo muy utilizado para la detección de blanco, negro y vacío en el piso donde circula el robot.

Con respecto al movimiento, se seleccionaron dos motores de corriente continua con una caja de reducción incorporada (Motor Yellow 3-12VDC 2 Flats Shaft, s/f), también de uso popular en el mundo de la electrónica. Para poder realizar sincronización entre las ruedas, se le agrega un disco de encoder a cada una de ellas. Para la lectura del encoder, se utiliza un optointerruptor fc-03 (MOCH22A Interrupter Datasheet pdf - OPTO Interrupter. Equivalent, Catalog, s/f) basado en un comparador LM393. Los motores se controlan mediante el módulo basado en el driver L298N (Handson Technology, s/f).

En relación con la alimentación, la misma se realiza a través de dos baterías recargables de ion litio 14500 de 3.7V cada una, que tienen el mismo tamaño de las AA tradicionales, pudiendo utilizar un portapilas convencional. Para su recarga, incorporamos el módulo TP4056 (NanJing Top Power ASIC Corp., s/f) que permite hacerlo con un voltaje de entrada de 4.5 a 5.V y una ficha miniUSB como la de los cargadores de celulares tradicionales.

En lo que respecta a la salida de datos hacia los estudiantes, además de las herramientas que posee la interfaz de desarrollo, que se verán más adelante, se agregó un display de cristal líquido de 16 caracteres por dos líneas compatible con el controlador Hitachi HD44780 (Hitachi, s/f).

Para finalizar, la estructura mecánica de este primer prototipo está en proceso, usando herramientas de diseño 3D. El objetivo es realizar una cantidad mínima de piezas que puedan ser obtenidas de las impresoras 3D más económicas del mercado. Es por esta razón que ninguna de ellas superará los 14 cm en alguna de sus dimensiones, dado que las camas de impresión tienen un límite de 15 cm en los modelos económicos. Para que la impresión sea más sencilla y que los componentes sean más fáciles de ubicar, el diseño se realizó en dos capas. En la imagen que se presenta a continuación los elementos en gris son los que se imprimen. Los demás componentes están ubicados para estudiar su organización.



**Fig. 1.** Diseño 3D del primer prototipo basado en la arquitectura propuesta

### 3 Arquitectura del Software

#### Arquitectura de la Solución

La constitución de un servidor web en el propio robot es una propuesta que ya estaba presente en el mundo de la robótica educativa. En (Saleiro et al., 2013) el objetivo es habilitar la programación del robot desde cualquier dispositivo Wi-Fi mediante un navegador con javascript, y sin instalar ningún software adicional. También se propone el uso del framework Blockly como herramienta de programación. En otros casos (Erdemir et al., 2015), el servidor tiene la función de permitir el uso del robot en forma remota. Pero en ninguno de los ejemplos encontrados, el objetivo es de realizar un uso concurrente del robot por parte de los estudiantes.

A continuación, presentamos un diagrama tentativo de la arquitectura del sistema a desarrollar. El diseño aquí expuesto servirá como referencia para la implementación, pero no debe considerarse como última versión.

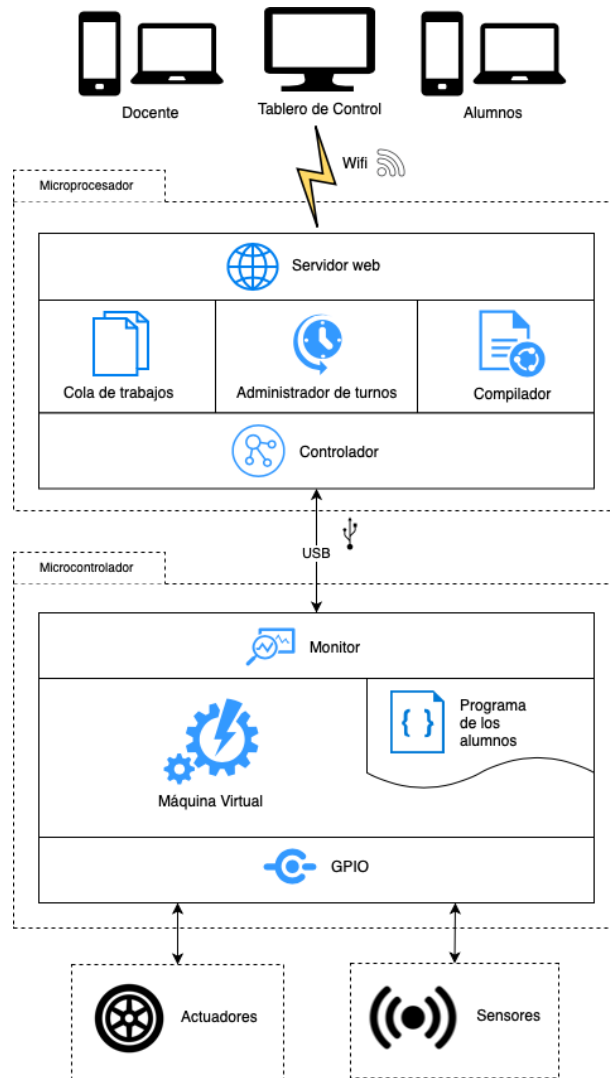


Fig. 2. Diagrama de la arquitectura de software tentativa de Mendieta

Tanto estudiantes como docentes podrán acceder al dispositivo mediante el navegador de internet (ya presente en los dispositivos de los usuarios). Para ello, el microprocesador del robot expondrá un servidor web que permitirá acceder tanto a la interfaz de configuración y administración de los turnos, como al entorno de desarrollo desde donde los estudiantes podrán programar el robot.

Este diseño permitirá programar al robot de forma inalámbrica (a través de la red Wifi) y usando cualquier tipo de dispositivo (notebook, computadora de escritorio, celular, tablet, etc.). Opcionalmente, se puede conectar también un tablero de control que permita a toda el aula visualizar la cola de trabajos pendientes.

Por su parte, el módulo de administración de turnos será responsable de decidir cuándo se ejecutará cada trabajo. Para ello hará uso de la cola de trabajos, que irá almacenando en orden cada uno de los trabajos enviados por los estudiantes. Por defecto, se respetará el orden de llegada, pero si el docente lo deseara, se podrá modificar manualmente el orden de ejecución de los trabajos, así como eliminar trabajos de la cola.

Al mismo tiempo será el compilador el responsable de asegurar, por un lado, que los trabajos que se agregan a la cola sean sintácticamente válidos y, por otro, de generar las instrucciones que se enviarán al microcontrolador para su posterior ejecución.

En relación con la comunicación con el microcontrolador, el módulo controlador será responsable de su ejecución, permitiendo tanto el control del robot mediante el envío de programas a correr, así como la recepción de información del estado del robot que luego podrá poner a disposición del servidor web para informar al usuario.

Asimismo, en el microcontrolador estará instalado un firmware que permitirá la programación interactiva del robot desde el microprocesador usando los pines de comunicación serial. Este firmware estará compuesto por un programa monitor que será responsable de la comunicación por puerto serie, y una máquina virtual que será la que ejecute los programas escritos por el usuario.

Por su parte, el monitor implementará el protocolo de comunicación que permitirá, por un lado, aceptar los mensajes que envíe el microprocesador y, por otro lado, enviará las notificaciones que correspondan. Los mensajes que el microprocesador reciba permitirán interactuar con el robot, ejecutar un nuevo programa, modificar el valor de una variable, o controlar un actuador. Las notificaciones que el monitor envíe permitirán al microprocesador monitorear el estado del robot, de sus sensores, de sus variables, de las tareas del programa, etc.

Por último, el módulo GPIO permitirá abstraer los detalles del hardware y acceder tanto a los sensores como a los actuadores.

### **Interfaz Gráfica de Usuario**

Al igual que en el caso de la arquitectura, las interfaces que se expondrán a continuación deben ser consideradas meros prototipos para ayudar en la toma de decisiones de diseño y guiar la implementación. Las mismas probablemente terminen cambiando drásticamente en su versión final.

En lo que respecta a las herramientas de programación, se desarrollará una solución basada en un lenguaje visual basado en bloques. A continuación, presentaremos un prototipo de la interfaz de usuario que se presentará a los estudiantes para la programación del robot. Este prototipo es una adaptación de la herramienta Physical Bits, desarrollada con anterioridad con el objetivo de proveer un entorno de programación interactivo para robótica educativa.



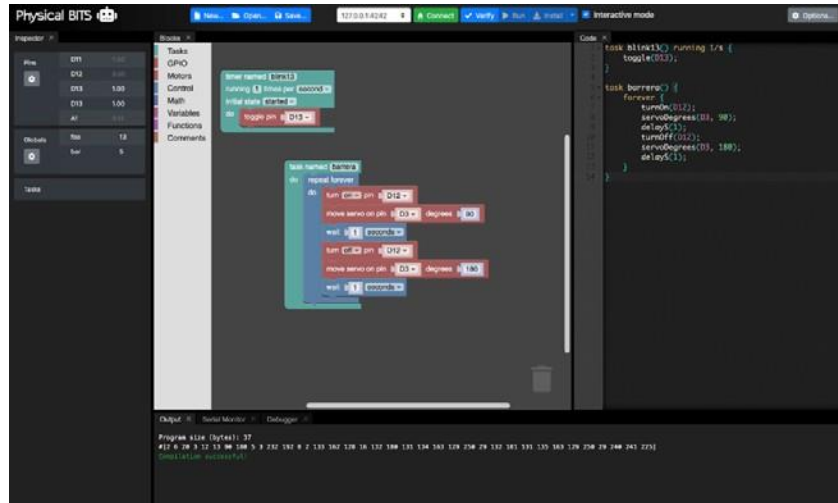


Fig. 3. Interfaz de usuario para la programación del robot

Para poder visualizar tanto los sensores disponibles en el robot como las variables definidas en el programa se desarrollará una herramienta de monitoreo que permitirá al estudiante elegir qué valor desea observar y luego visualizarlo en el tiempo de distintas maneras. Un prototipo de la interfaz de monitoreo se puede ver a continuación:

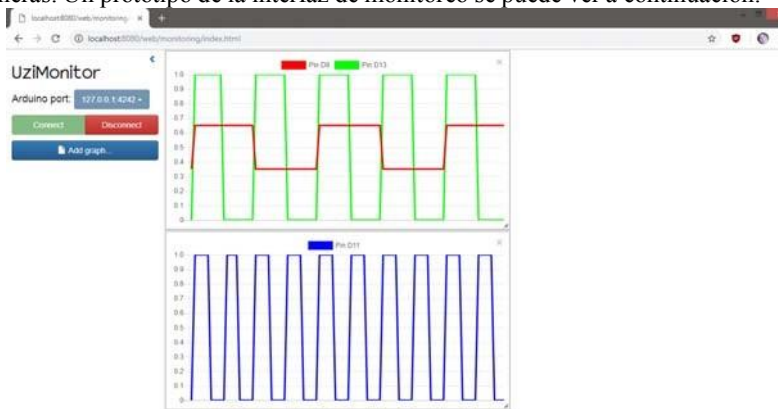


Fig. 4. Interfaz de monitoreo

En el caso de programas complejos resulta indispensable algún tipo de herramienta que permita frenar la ejecución del programa, ejecutar cada instrucción paso a paso, y observar el estado del programa en detalle para poder entender y resolver errores en su lógica. Para ello se desarrollará una interfaz similar a la que se observa a continuación.

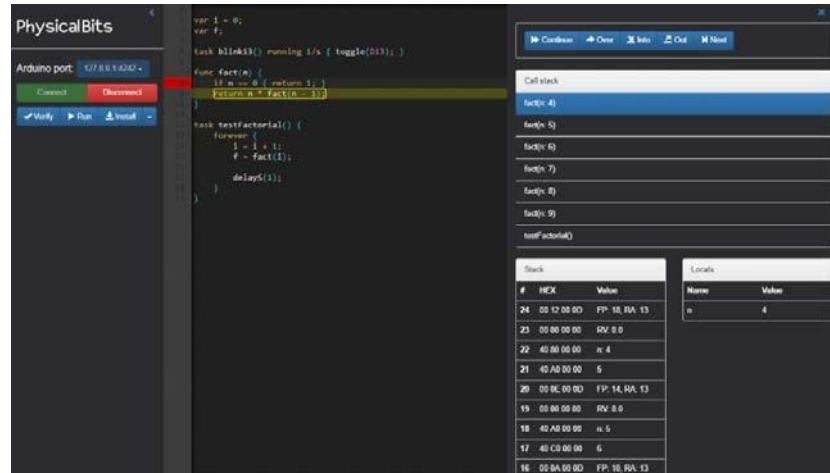


Fig. 5. Interfaz de depuración

El docente podrá acceder a una interfaz de administración que le permitirá visualizar los trabajos pendientes, los trabajos ya ejecutados, las opciones de configuración, y estadísticas acerca de la actividad. A continuación, se puede observar un prototipo de la interfaz de administración.

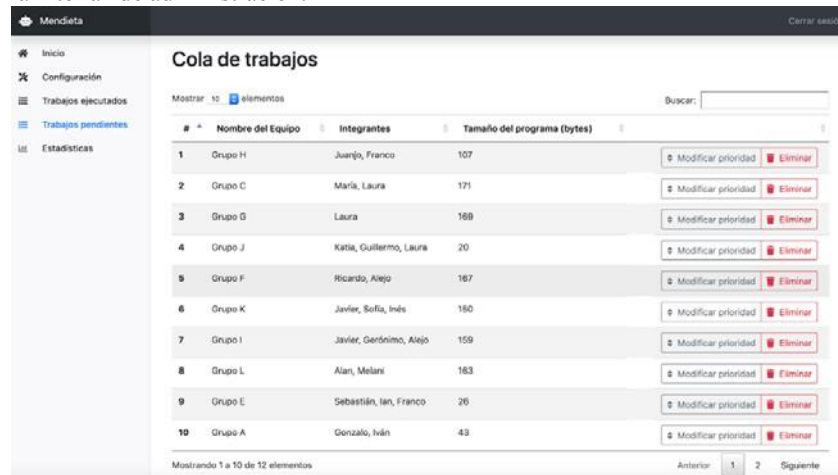


Fig. 6. Interfaz de administración

## 4 Conclusiones

Mendieta está en proceso de prototipado y es muy probable que se produzcan cambios tanto en su arquitectura de hardware como de software. A pesar de que en este primer prototipo están definidos los elementos que componen su estructura física, es preciso resaltar que es sólo un modelo de prueba, y que lo que se define es un framework que permitirá construir diferentes robots bajo el mismo esquema de uso (un robot por clase). Cabe destacar que es posible extender su uso a la virtualidad, desarrollando un sitio donde los estudiantes puedan programar en forma remota el robot que visualizan a través de una o más cámaras. En este caso, debería ser una propuesta donde el dispositivo y su ambiente puedan volver a punto cero en forma automática.

Más allá de que la primer interfaz que se planeó para su programación está basada en el modelo Blockly, también se espera que otros grupos puedan tomar este punto de partida para crear otro tipo de interfaces.

Cabe destacar que esta propuesta fue presentada en el año 2019 para el premio Clarín Zurich de Educación, obteniendo una mención que ha permitido financiar gran parte del proyecto.

## Referencias

1. *Arduino Nano* | *Arduino Official Store*. (s/f). Recuperado el 15 de junio de 2020, de <https://store.arduino.cc/usa/arduino-nano>
2. Aroca, R. V., Gomes, R. B., Tavares, D. M., Souza, A. A. S., Burlamaqui, A. M., Caurin, G. A., & Goncalves, L. M. (2012). Increasing students' interest with low-cost CellBots. *IEEE Transactions on Education*, 56(1), 3–8.
3. Banchoff Tzancoff, C. M., Martin, S., Gómez, S., & López, F. E. M. (2019). Experiencias en robótica educativa: Diez años trabajando con escuelas argentinas. *XIV Congreso Nacional de Tecnología en Educación y Educación en Tecnología (TE&ET 2019)*, (Universidad Nacional de San Luis, 1 y 2 de julio de 2019).
4. Darrah, T., Hutchins, N., & Biswas, G. (2018). Design and development of a low-cost open-source robotics education platform. *ISR 2018; 50th International Symposium on Robotics*, 1–4.
5. Erdemir, G., Kuzucuoglu, A. E., Kaplanoglu, E., & El-Kahlout, Y. (2015). *Design and Implementation of Web Based Mobile Robot Control Platform for Robotics Education*. Applied Mechanics and Materials; Trans Tech Publications Ltd. <https://doi.org/10.4028/www.scientific.net/AMM.704.283>
6. Finocchiaro, A. (s/f). *Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica-Resolución CFE 343/2018*.
7. Freaks, E. (2016). Ultrasonic ranging module hc-sr04. *HC-SR04 datasheet*.
8. Handson Technology. (s/f). *L298N Motor Driver*. Recuperado el 15 de junio de 2020, de <http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>
9. Hitachi. (s/f). *HD44780U (LCD-II), (Dot Matrix Liquid Crystal Display Controller/Driver)*. 60.
10. Hostler, A., Benson, B., & Warner, J. (2017). BudgetROV: An ultra low cost robotics platform for education and research. *OCEANS 2017-Aberdeen*, 1–4.

11. Junior, L. A., Neto, O. T., Hernandez, M. F., Martins, P. S., Roger, L. L., & Guerra, F. A. (2013). A low-cost and simple arduino-based educational robotics kit. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Robotics and Control (JSRC), December edition*, 3(12), 1–7.
12. Lopes Filho, J. A. B., Almeida, W. R. M., & Martins, S. G. (2011). Development of a multitasking mobile robot for the construction of educational robotics kits. *2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, 213–216.
13. López-Rodríguez, F. M., & Cuesta, F. (2016). Andruino-A1: Low-Cost Educational Mobile Robot Based on Android and Arduino. *Journal of Intelligent & Robotic Systems*, 81(1), 63–76. <https://doi.org/10.1007/s10846-015-0227-x>
14. *MOCH22A Interrupter Datasheet pdf—OPTO Interrupter. Equivalent, Catalog.* (s/f). Recuperado el 15 de junio de 2020, de <https://datasheets-pdf.com/pdf/685501/Junye/MOCH22A/4>
15. *Motor Yellow 3-12VDC 2 Flats Shaft.* (s/f). Wiltronics. Recuperado el 15 de junio de 2020, de <https://www.wiltronics.com.au/product/10137/yellow-motor-3-12vdc-2-flats-shaft/>
16. Ministerio de Educación - Presidencia de la Nación. (2017). *Escuelas del Futuro*. Recuperado el 12 de junio de 2020, de <http://www.bnm.me.gov.ar/giga1/documentos/EL005852.pdf>
17. Munizaga, R. A., Moleker, C., Yonzo, G., & Zabala, G., 2013. Everyone to Robotics: an educational robotics project in Argentina. Robocup 2013.
18. NanJing Top Power ASIC Corp. (s/f). *TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8*. Recuperado el 15 de junio de 2020, de [https://www.semtronics.net/shop/resources/pdf\\_document/18/c1/9.pdf](https://www.semtronics.net/shop/resources/pdf_document/18/c1/9.pdf)
19. *orange pi zero—Orangepi.* (s/f). Recuperado el 15 de junio de 2020, de <http://www.orange-pi.org/orangepizero/>
20. Ripani, María Florencia. (2017). *Programación y robótica: Objetivos de aprendizaje para la educación obligatoria*. Ministerio de Educación - Presidencia de la Nación. [https://www.argentina.gob.ar/sites/default/files/programacion\\_y\\_robotica\\_0.pdf](https://www.argentina.gob.ar/sites/default/files/programacion_y_robotica_0.pdf)
21. Rubenstein, M., Ahler, C., & Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. *2012 IEEE International Conference on Robotics and Automation*, 3293–3298.
22. Saleiro, M., Carmo, B., Rodrigues, J. M., & du Buf, J. H. (2013). A low-cost classroom-oriented educational robotics system. *International Conference on social robotics*, 74–83.
23. Semiconductors, V. (2009). *TCRT5000, TCRT5000L*. Aug.