

Robobo: la siguiente generación de robot educativo

Bellas, Francisco; Prieto, Abraham; Duro, Richard J.

*Grupo Integrado de Ingeniería, Escola Politécnica Superior, Campus de Esteiro,
Universidade da Coruña*

RESUMEN

Las carreras universitarias de informática e ingeniería han estado utilizando robots móviles en diferentes asignaturas desde hace mucho tiempo. Hasta el momento, debido principalmente a limitaciones económicas, estos robots educativos han sido bastante simples en términos tecnológicos. Esto no era un gran problema porque la robótica no se consideraba un mercado real en la industria, por lo que los robots se usaban en las clases como prototipos, sin una expectativa de aplicación a la realidad. Pero como todos sabemos, la situación actual, y el futuro cercano, hacen que la robótica sea un mercado clave para los ingenieros y los informáticos que se forman en las universidades. Como consecuencia, los robots utilizados en las aulas universitarias deben ser actualizados para adecuarlos a la realidad tecnológica que se maneja en el mundo industrial.

Este artículo presenta Robobo, un robot móvil educativo de bajo costo desarrollado en la Universidade da Coruña. Robobo combina una base con ruedas simple con un teléfono inteligente, que proporciona la última tecnología al robot. Con Robobo, los estudiantes pueden desarrollar sus propios proyectos usando cámaras, micrófonos o pantallas de alta resolución, acercando la enseñanza universitaria al mercado real que encontrarán cuando finalicen sus estudios.

PALABRAS CLAVE: Robótica educativa, Herramientas tecnológicas para el aula, Dispositivos móviles.

CITA RECOMENDADA:

Bellas, F., Prieto, A., Duro, R. J. (2018). Robobo: la siguiente generación de robot educativo. En E. de la Torre Fernández (ed.) (2018). *Contextos universitarios transformadores: retos e ideas innovadoras. II Jornadas de Innovación Docente*. Cufie. Universidade da Coruña (pág. 13-30).

DOI capítulo: <https://doi.org/10.17979/spudc.9788497496780.013>

DOI libro: <https://doi.org/10.17979/spudc.9788497496780>

ABSTRACT

Computer science and engineering majors have been using mobile robots in different subjects for a long time. So far, due primarily to economic constraints, these educational robots have been quite simple in technological terms. This was not a big problem because robotics was not considered a real market in the industry, so robots were used in classes as prototypes, without an expectation of application to reality. But as we all know, the current situation, and the near future, make robotics a key market for engineers and IT graduates in universities. As a consequence, the robots used in the university classrooms must be updated to adapt them to the technological reality that is handled in the industrial world.

This article presents Robobo, a low cost educational mobile robot developed at the University of Coruña. Robobo combines a simple wheeled base with a smartphone, which provides the latest technology to the robot. With Robobo, students can develop their own projects using cameras, microphones or high-resolution displays, bringing university education closer to the real market they will find when they finish their studies.

KEY WORDS: Educational robotics, Technological tools for the classroom, Mobile devices.

1. INTRODUCCIÓN

Los títulos universitarios en ciencias de la computación e ingeniería han estado usando robots móviles en diferentes temas desde hace mucho tiempo [1][2][3]. Son una herramienta básica para enseñar robótica autónoma, pero, además, han demostrado ser muy útiles en otros campos como la visión artificial [4] o los sistemas de control. Hasta ahora, principalmente debido a limitaciones económicas, estos robots educativos han sido bastante simples en términos tecnológicos. El robot educativo más popular en la educación superior es el Lego Mindstorms [5], un robot modular equipado con un procesador Arm926ej-S Core @ 300 Mhz en su última versión (EV3). Este robot está equipado con un sensor de sonda o infrarrojo, un giroscopio y un sensor de color, y tiene un precio promedio de 350-400 €. Otro robot móvil muy popular en las universidades europeas es el Thymio II [6], adecuado para la enseñanza de sistemas multi-robot. Solo cuesta alrededor de 120 € y está equipado con sensores de infrarrojos, un sensor de temperatura y un acelerómetro. Tiene un microcontrolador PIC24 a 32 MHz de bajo rendimiento. Como consecuencia de esta simplicidad, el uso práctico de robots reales en la educación superior se ha limitado a simples mecanismos de control y aplicaciones de laboratorio.

Esta limitación no ha sido un problema real porque la robótica no era un mercado relevante en la industria, por lo que los robots se usaban en clases como prototipos simples, sin un interés claro sobre su aplicación en la realidad. Pero, como es bien sabido, la situación actual y futura es que la robótica será un mercado clave para titulados en ingeniería en informática en un plazo muy corto de tiempo [7]. Como consecuencia, los robots utilizados en las aulas de la Universidad deben actualizarse para incluir todas las características de los robots reales que estarán presentes en nuestra realidad, tanto en el ámbito industrial como en el doméstico. Esto implica que las clases robóticas deben cambiar plataformas simples o robots simulados por otras realistas.

Si uno se mueve a robots con sensores más potentes y mayor capacidad de cómputo, el precio aumenta. Por ejemplo, el robot e-puck [8] es un robot móvil que fue muy popular en la

última década en varias universidades europeas. También tiene sensores infrarrojos y un controlador dsPIC30F6014A @ 64MHz, e incluye una cámara de baja resolución de 640x480 píxeles. El precio promedio de una unidad es de 600 €. Un robot más potente ampliamente utilizado en la educación superior es el Kephhera IV [9], que tiene una cámara de resolución 752×480 , un procesador ARM Cortex 800MHz, sensores infrarrojos y de ultrasonidos, pero con un precio de alrededor de 2500 €. Si nos pasamos a robots móviles de gama alta como el NAO [10], con una CPU Atom Z530 de 1.6 Ghz, dos cámaras de 1280x960, ultrasonidos y sensores táctiles, el precio es aún mayor (alrededor de 6000 €), algo inalcanzable para la mayoría de las instituciones de educación superior en España, considerando además la gran cantidad de robots que se necesitan en un aula (no más de dos o tres estudiantes por robot y, preferiblemente, solo uno). La plataforma más destacable en términos de equilibrio de características y coste es TurtleBot [11]. Es un robot móvil de bajo coste (520 €) y de código abierto, muy popular para los usuarios de ROS (Robot Operating System) como plataforma de prueba, principalmente para la investigación en SLAM (localización y mapeado simultáneos), ya que contiene un Sistema LIDAR de 360° en la última versión. Su aplicación a una gama más amplia de dominios robóticos, como la interacción humano-robot, el uso de reconocimiento de voz, cámaras o interfaces táctiles, es limitada.

Ante esta situación, hace unos años, los autores de este artículo decidimos comenzar el desarrollo de un robot móvil educativo de bajo coste que pudiese incorporar tecnología de última generación para ser utilizado en las titulaciones de ingeniería e informática. Este robot se ha llamado Robobo [12], y combina una base con ruedas simple con un teléfono inteligente (Smartphone). Con Robobo, los estudiantes pueden desarrollar proyectos de ingeniería usando cámaras, micrófonos o pantallas de alta resolución, acercando la enseñanza al mercado real que encontrarán cuando terminen sus estudios. Este artículo se centra en la presentación de la versión actual del hardware y software de Robobo, que se explica en el apartado 2. Además, su potencialidad en la educación superior se ilustrará a través de un ejemplo de proyecto de estudiante, en el apartado 3. Este proyecto explota las características de interacción humano-

robot (HRI) que proporciona el smartphone en función de sus capacidades visuales, auditivas y táctiles.

2. EL PROYECTO ROBOTO

El hardware de Roboto se compone de dos elementos: una base móvil con ruedas y un Smartphone que está conectado a la base, como se muestra en la Figura 1. Debe destacarse que el robot Roboto no es solo la base móvil, sino la combinación de base y Smartphone. Esta configuración proporciona las siguientes características:

- *Bajo coste:* la institución educativa debe adquirir solo la base móvil, ya que el Smartphone será el del propio estudiante. Nuestra experiencia en este aspecto es que los estudiantes perciben esta aplicación extra de su propio teléfono inteligente como positiva. Roboto se comercializará en paquetes de clase con un coste de referencia por unidad que será menor que los robots educativos más populares, TurtleBot 3 o LEGO EV3.
- *Actualización continua:* la tecnología del Smartphone, tanto en hardware como en software, está en actualización continua, por lo que Roboto se puede mejorar fácilmente simplemente reemplazando el Smartphone por otro más potente. Es bien sabido que los estudiantes suelen tener los últimos modelos, por lo que las nuevas características de hardware se pueden incorporar fácilmente en las clases. La base de Roboto admitirá nuevas funciones del Smartphone simplemente actualizando su firmware y software.
- *Duración a largo plazo:* en relación con las dos características anteriores, se puede concluir que la validez de Roboto es mayor que otros robots educativos, y la inversión en la base móvil dura más tiempo.
- *Tecnología de vanguardia:* además de los sensores y actuadores que se encuentran en la base móvil y que se explicarán más adelante en detalle, los estudiantes pueden usar todos los elementos de hardware que se incluyen en los Smartphone actuales:

- 2 cámaras de alta resolución
- Sensores de proximidad, luz y temperatura
- Giroscopio, acelerómetro y magnetómetro
- GPS
- Micrófono
- Pantalla LCD de alta resolución
- Conectividad: 4G, WI-FI, USB, ...
- Altavoz



Figura. 1. Robobo con el Smartphone conectado (izquierda). La base móvil con un empujador impreso en 3D (derecha)

Además del hardware de Robobo, se ha desarrollado una arquitectura de software que permite programar Robobo mediante ROS (Robot Operating System) [13]. En las siguientes secciones, se describirá el diseño y la implementación del hardware y software de Robobo.

2.1 HARDWARE DE ROBOBO

El hardware de Robobo está compuesto por dos elementos: el Smartphone y la base móvil. Se puede utilizar cualquier Smartphone comercial que cumpla los siguientes requisitos:

- *Sistema operativo:* Android 5.1 o superior (la versión de iOS está en desarrollo)
- *Tamaño de pantalla:* mínimo 4.7 pulgadas y máximo 5.5 pulgadas
- *Espesor:* 95 mm máximo

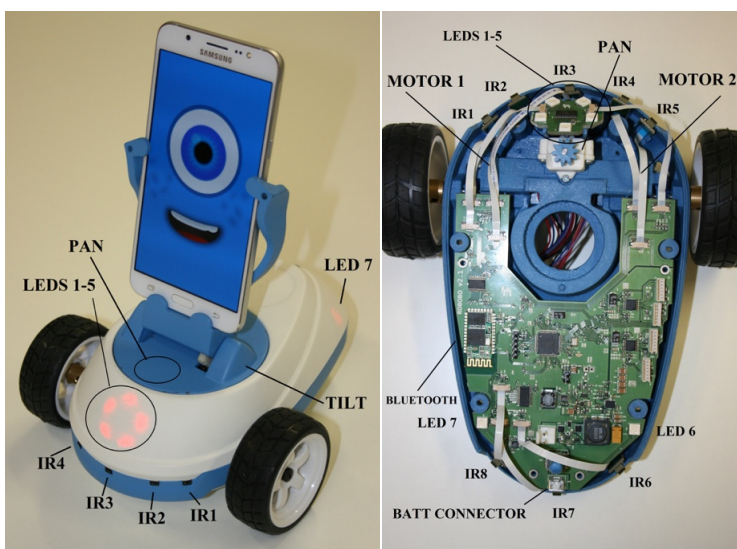


Figura 2. Elementos principales de hardware de la base de Robobo

Los principales elementos del hardware de la base se muestran en la Figura 2. Tiene dos motores en las ruedas (MT1, MT2) que se encargan de realizar el desplazamiento de la plataforma. Son motores de reducción 150:1 con encoder, junto con un diseño de piñón y cable. Esta relación adicional aumenta el par máximo disponible y la aceleración del robot. En la parte posterior de la base, hay una bola giratoria que permiten un movimiento omnidireccional. Además, la base está equipada con dos motores adicionales que permiten un movimiento independiente de la base y el Smartphone:

- *Motor de giro horizontal (PAN):* es un motor de reducción 150:1 con encoder con reducción de corona-piñón. Esta reducción brinda la posibilidad de realizar movimientos suaves y precisos, con un desplazamiento angular mínimo de 4.86° .
- *Motor de giro vertical (TILT):* es un motor de reducción 1000:1 con encoder.

Estos dos motores le proporcionan a Robobo muchos movimientos diferentes y un tipo más rico de posibilidades de interacción humano-robot.

Se diseñó, además, una placa de circuito impreso (PCB) específica para Robobo con el fin de maximizar el rendimiento y las capacidades de la base, y para adaptar los componentes electrónicos al diseño externo. Estos son sus principales componentes electrónicos:

- *Microcontrolador*: la estructura electrónica del Robobo está construida alrededor de un microcontrolador PIC32. La CPU funciona a 80 MHz, lo que es suficiente para controlar los sensores, actuadores y llevar a cabo tareas diversas que debe realizar el PIC.
- *Sensores*: Robobo utiliza 8 sensores infrarrojos Vishay I2C VCNL4040 (IR1 a IR8) con tres propósitos, que dependen de su orientación:
 - Detección de objetos (IR1, IR3, IR5, IR7): se colocan en la parte delantera y trasera de la base orientada paralela al suelo. Se usan para la detección y evitación de obstáculos en entornos desordenados.
 - Evitación de caídas (IR2, IR4, IR6, IR8): colocadas en la parte delantera y trasera de la base, se colocan en una orientación de 45° hacia el suelo, con el objetivo de detectar agujeros y evitar caídas en mesas o escaleras.
 - Detección de luz ambiental: el VCNL4040 también se puede usar como un sensor de luz ambiental.
- *Actuadores*: como se comentó anteriormente, se utilizan 4 motores DC-DC para mover y controlar la base. Los 4 motores tienen un eje de codificador magnético de 6 polos para medir con precisión las revoluciones del motor.
- *LED RGB*: hay 7 LEDs RGB (L1 a L7 en la Figura 2). Cinco de ellos se colocan en la parte delantera de la base y dos en la parte trasera. El controlador LED tiene una resolución de 4096 bits por color, lo que proporciona una alta resolución.

- *Módulo Bluetooth*: se utiliza un módulo Bluetooth Microchip BM78 para establecer la comunicación entre la base y el Smartphone. Este módulo fue seleccionado por ser simple, de bajo coste, con certificación CE y FCC.
- *Sistema de alimentación*: en la base se monta una batería LiPo de 2500 mAh y 3.7V. Fue seleccionada por su alta densidad de carga, con el fin de proporcionar un alto nivel de autonomía en el mínimo espacio posible. El rango de voltaje de estas baterías es 4.2V-3.0V. Además, hay un administrador de batería en la placa de circuito para recargar la batería a través de un conector USB BATT (ver Figura 2).

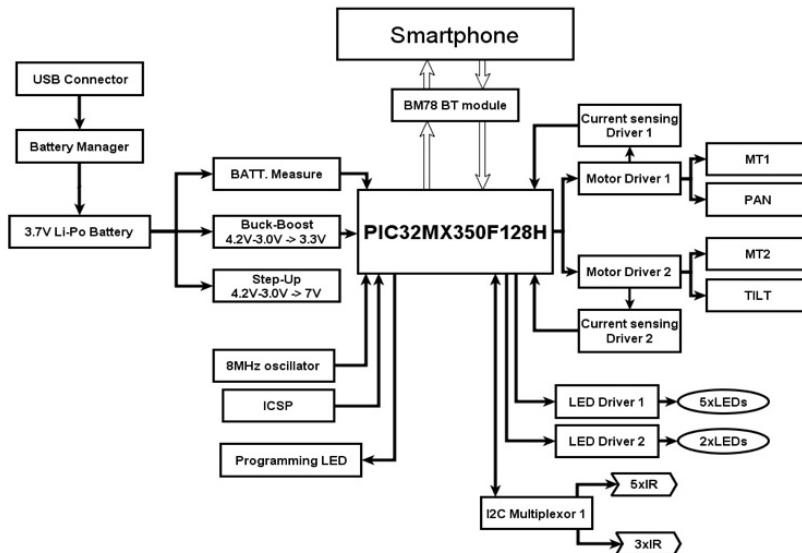


Figura 3. Arquitectura electrónica de Robobo

2.2 SOFTWARE DE ROBOBO

El desarrollo del software de Robobo se puede organizar en bloques de bajo nivel (firmware) y de alto nivel (arquitectura de software):

- *Bajo nivel*: la Figura 3 muestra la arquitectura de hardware general de la base móvil, donde solo se representan las rutas de conexión principales para aclarar el esquema. El núcleo del sistema es el microcontrolador PIC32, que se encarga de controlar los

diferentes circuitos integrados, sensores y actuadores en la base. Además, el PIC32 gestiona las comunicaciones entre el microcontrolador y el módulo Bluetooth, que envía información al Smartphone y recibe los comandos del motor.

- *Alto nivel:* el principal requisito de la arquitectura software de Robobo era permitir la programación del robot utilizando diferentes paradigmas como ROS, Java o Scratch. Para ello diseñó una arquitectura de software completamente modular, que se muestra en la Figura 4. Se basa en el concepto de módulo Robobo y una biblioteca muy ligera, llamada Robobo Framework, que proporciona los mecanismos esenciales para administrar y ejecutar esos módulos en un teléfono Android. Además de esta biblioteca, se pueden cargar dos tipos diferentes de módulos:
 - Módulos de funcionalidad (en naranja en la Fig. 4): implementados en Java utilizando la API estándar de Android, proporcionan diferentes funcionalidades para el robot, como reconocimiento de voz, detección de rostros, detección de ambiente o movimiento físico.
 - Una serie de módulos proxy (módulos ros-proxy y remote-proxy en la Figura 4), también implementados en Java, que proporcionan interfaces particulares a otros entornos de programación.

Robobo viene por defecto con un conjunto particular de estos módulos, pero, como están completamente desacoplados entre ellos y con respecto al entorno de desarrollo, los usuarios avanzados pueden personalizar el conjunto de módulos e incluso implementar nuevos módulos para admitir nuevas funcionalidades de robot, o incluso nuevos paradigmas de programación a través de módulos de proxy.

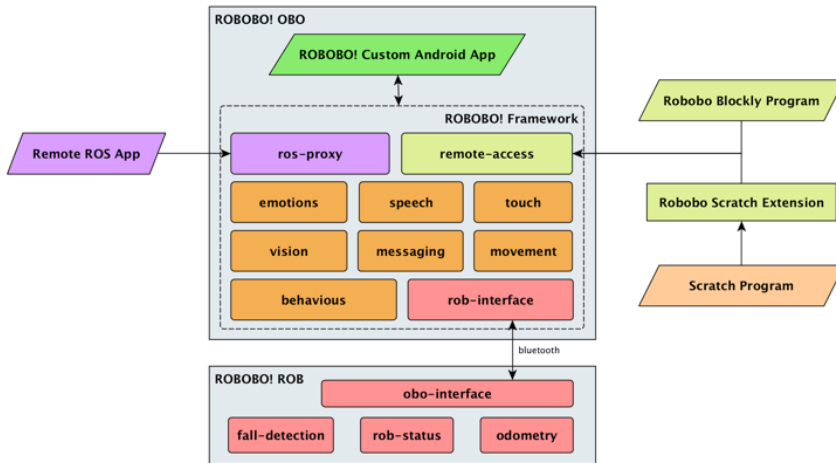


Figura 4. Diagrama de bloques de la arquitectura del software Robobo

Finalmente, es importante tener en cuenta que existe un módulo para conectar el Robobo Framework y sus módulos a la base robótica Robobo. El módulo *rob-interface*, que se muestra en rosa en la Figura 4, implementa el protocolo de comunicaciones Bluetooth de la base y proporciona una API de control para que otros módulos lo utilicen.

La programación Java se soporta directamente por la API nativa proporcionada por los diferentes módulos. Al utilizar el Robobo Framework, los usuarios pueden crear aplicaciones personalizadas de Android que controlan el comportamiento del robot. Para la programación con bloques, actualmente se admiten dos enfoques diferentes: Scratch y nuestro propio IDE basado en bloques que usa Blockly. Como se puede ver en la Figura 4, ambos enfoques están conectados al marco utilizando el mismo proxy, el *protocolo de acceso remoto de Robobo* (RRAP). Finalmente, Robobo también se puede programar utilizando el Robot Operating System (ROS), que es el lenguaje de programación principal para enseñar robótica en niveles de educación superior.

3. EJEMPLO DE PROYECTO PARA LOS ESTUDIANTES

Para ilustrar el tipo de proyecto que se puede llevar a cabo utilizando Robobo en titulaciones de informática o ingeniería, esta sección describe un ejemplo específico que se propuso a los

estudiantes en la asignatura de "Robótica" del grado de Informática en la Universidad de Coruña, durante el año académico 2016/17. El proyecto se centró en interacción humano-robot, específicamente, en la programación de Robobo para actuar como mascota. Es decir, el robot debe pedirle al usuario alimento, atención y "afecto". La forma en que se realiza debe ser seleccionada por los estudiantes, dándoles la oportunidad de crear soluciones altamente realistas. Los módulos de funcionalidad necesarios para resolver este proyecto usan todas las modalidades interactivas de Robobo:

- *Habla*: la capacidad de voz del robot se implementó utilizando las bibliotecas de Android y se implementó un módulo *speechProducer*.
- *Reconocimiento de sonido*: *speechRecognition* y *soundRecognition* se implementaron utilizando las bibliotecas Pocketsphinx [14] y TarsosDSP [15].
- *Visión*: se implementó un módulo *faceRecognition* utilizando las bibliotecas de Android, y un módulo de detección de color usando la biblioteca OpenCV [16].
- *Sensor táctil*: el módulo *touchDetection* realiza el reconocimiento de los gestos táctiles sobre la pantalla del Smartphone. Permite un toque rápido y simple en la pantalla (*tap*), un toque sostenido sobre la pantalla (*touch*) o un deslizamiento sobre la pantalla más o menos rápido (*caress* y *fling*). Se usaron las bibliotecas de Android para implementar este módulo.
- *Emoción visual*: el módulo *emotionProducer* permite mostrar diferentes imágenes o animaciones en la pantalla del Smartphone, proporcionando una alta capacidad de interacción a los usuarios para mostrar las emociones del robot.
- *Movimientos*: el módulo *motorCommand* permite al usuario mover los cuatro motores de la base Robobo.

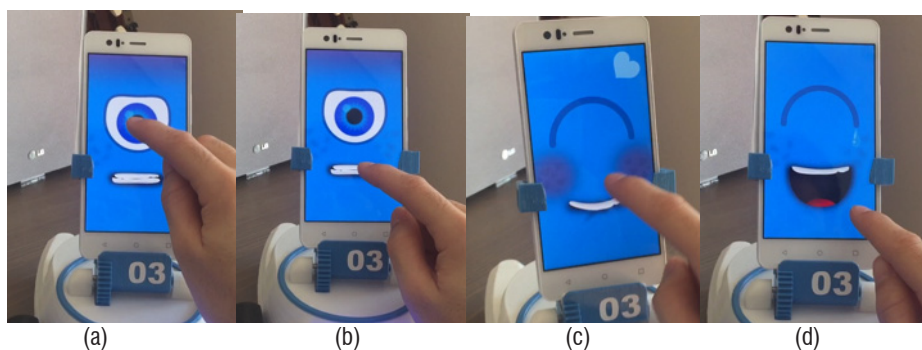


Figura 5. Expresiones de interacción táctil de Robobo en el ejemplo como mascota

Con estos módulos de funcionalidad y el lenguaje de programación seleccionado por los estudiantes (y con el apoyo de ROS), las especificaciones del comportamiento de las mascotas fueron las siguientes:

- *Instintos básicos*: Robobo debe almacenar el nivel de hambre y sed y, cuando están por debajo de un umbral, debe decir "Tengo hambre" o "Tengo sed" hasta que se vuelvan a llenar.
- *Movimiento*: Robobo debe moverse para capturar la atención del usuario si no se ha ejecutado ningún evento durante un lapso de tiempo predefinido. Por ejemplo, puede girar o emitir un sonido.
- *Interacción táctil*: se deben implementar dos comportamientos diferentes según el tipo de modalidad de pantalla táctil:
 - *Tap*: si el usuario toca la pantalla con un solo toque, Robobo debe reaccionar de manera diferente dependiendo de la parte de la "cara" que toque. Si está en el "ojo" o en la "boca", moverá el motor de inclinación hacia atrás y mostrará una expresión de enojo (las figuras 5a y 5b muestran ejemplos). Si está en cualquier otro punto, mostrará una carcajada y emitirá un sonido que dice "cosquillas" (véanse las figuras 5c y 5d).
 - *Fling*: si el usuario desliza un dedo sobre la pantalla, la unidad de pan-tilt se mueve en consecuencia. Las posibles direcciones de deslizamiento se deben

discretizar en cuatro movimientos de pan-tilt: inclinar hacia atrás o hacia adelante e inclinar hacia la derecha o hacia la izquierda.

- **Voz:** Robobo debe reaccionar a las siguientes frases predefinidas:
 - "*Aquí viene la comida*": el robot se prepara para recibir comida.
 - "*Aquí viene la bebida*": el robot se prepara para recibir la bebida.
 - "*Hola*": el robot responde diciendo "Hola".
 - "*¿Cómo estás?*": el robot responderá diciendo si tiene sed o hambre.
- **Visión:**
 - **Color:** Robobo debe detectar 2 colores diferentes, verde para comida y azul para bebida, pero solo después de que se haya detectado el comando de voz correspondiente. En ambos casos, debe decir si el color es correcto o no. Por ejemplo, si el usuario dice "Aquí viene la bebida", el robot debe buscar un área de color azul de un tamaño predefinido en su campo de visión (la imagen de la izquierda de la Figura 6 muestra un ejemplo), y después de un tiempo transcurrido, debe decir "Gracias por la bebida" o "No veo la bebida".
 - **Cara:** Robobo debe detectar la cara del usuario. Si está por debajo de un umbral, el robot se moverá hacia atrás (Figura 6 derecha).

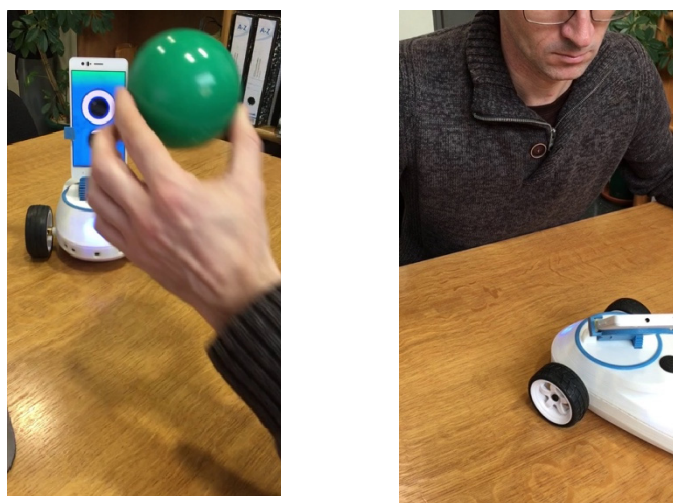


Figura 6. Bola verde que representa “comida” (izquierda) y reconocimiento de caras (derecha)

Este ejemplo resultó muy motivante para los estudiantes. Como se observa, aborda un campo de gran actualidad en la robótica real, como es la interacción hombre-robot, que se estaba tratando hasta ahora con unidades más caras, como el robot NAO. Aunque es un ejemplo simple, es fácil ver que hay un gran potencial en el uso de este tipo de robots simples con funcionalidades de gama alta (estructuras de detección y actuación proporcionadas por la combinación de la base) y el Smartphone. Nuestro objetivo es introducir progresivamente este enfoque en más cursos durante el próximo año.

4. CONCLUSIONES

Los robots tradicionales utilizados en la educación superior deben actualizarse para preparar a los estudiantes de ingeniería e informática para el mercado real que enfrentarán en el futuro cercano. En esta línea, hemos diseñado, desarrollado y probado un robot educativo de bajo coste llamado Robobo, que se compone de una base móvil simple que lleva un Smartphone acoplado. Con esta configuración, los estudiantes pueden utilizar tecnología de vanguardia en las clases ahora y en el futuro, porque Robobo se puede actualizar fácilmente solo mediante la

actualización del modelo de Smartphone. El robot estará disponible desde finales de 2017 a través de la página web del Proyecto Robobo: <http://www.theroboboproject.com>.

5. REFERENCIAS

- Schilling, K., Rösch, O. (2002) Mobile Mini-Robots for Engineering Education, *Global Journal of Engineering Education* 6(1), IGI
- Fagin, B., Merkle, L. (2002) Quantitative analysis of the effects of robots on introductory Computer Science education, *Journal on Educational Resources in Computing*, 4, Article 2
- Williams, A. (2003) The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering, in *IEEE Transactions on Education*, vol. 46, no. 1
- Sankowski, D., Nowakowski, J. (2014) *Computer Vision In Robotics And Industrial Applications*, World Scientific
- Danahy, E., Wang, E., Brockman, J., Carberry, A., Shapiro, B., Rogers, C. B. (2014) Lego-based robotics in higher education: 15 years of student creativity. *International Journal of Advanced Robotic Systems*, 11(2), 27
- Mondada, F. et al: Bringing Robotics to Formal Education: The Thymio Open-Source Hardware Robot, in *IEEE Robotics & Automation Magazine*, vol. 24, no. 1, pp. 77-85,
- Frey, C., Osborne, M. (2017) The future of employment: How susceptible are jobs to computerisation?, *Tech Forecasting and Social Change*, Vol 114, pp 254-280 (2017)
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J., Floreano, D., Martinoli, A. (2009) The e-puck, a robot designed for education in engineering, *Proceedings of the 9th conference on autonomous robot systems and competitions*
- Soares, J., Navarro, I., Martinoli, A. (2016) The Khepera IV mobile robot: performance evaluation, sensory data and software toolbox, *Robot 2015: Second Iberian Robotics Conference* 767-781

- Shamsuddin, S. *et al.* (2012) Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO, 2012 IEEE 8th International Colloquium on Signal Processing and its Applications, pp. 188-193
- TurtleBot web page: <http://www.turtlebot.com>
- The Robobo Project web page: <http://en.theroboboproject.com>
- ROS web page: <http://www.ros.org>
- Pocketsphinx web page: <https://github.com/cmusphinx/pocketsphinx>.
- TarsosDSP web page: <https://github.com/JorenSix/TarsosDSP>.
- OpenCV web page: <http://opencv.org>