Artículo de revisión

Revisión de la plataforma robótica *LEGO Mindstorms* para aplicaciones educativas y de investigación

A review of *LEGO Mindstorms* robotics platform for educational and research applications

Edgar Tello-Leal
etello@uat.edu.mx

Tania Yukary Guerrero-Melendez
tyguerre@uat.edu.mx

Vicente Paul Saldivar-Alonso
vpsaldiv@uat.edu.mx

Universidad Autónoma de Tamaulipas
Ciudad Victoria - México

Fecha de recepción: Agosto 21 de 2013
Fecha de aceptación: Septiembre 5 de 2013

Palabras clave
LEGO, Robot, NXT, LeJOS, Sensor, Lenguaje de Programación.

Keywords
LEGO, Robot, NXT, LeJOS, Sensor, Programming Language.

Colciencias tipo 3

Resumen

La robótica es considerada como una de las áreas de gran interés de la ciencia y la tecnología de hoy en día. La robotización, es decir, los sistemas automatizados, se encuentran en casi todos los sectores de la sociedad. En la industria, los robots son utilizados para soldar, transportar, montar y pintar piezas. En medicina, robots sofisticados ayudan a llevar a cabo cirugías complejas. En el sector militar, los robots son usados para explorar áreas hostiles. Por tal motivo, la robótica móvil y los sistemas autónomos son tópicos de creciente interés en los programas educativos de Ingeniería y Ciencias Computacionales. Por lo anterior, en este trabajo se presenta una revisión de las principales características de los robots educativos LEGO Mindstorms, examinando los entornos de desarrollo disponibles para la compilación de programas de software que permitan definir el comportamiento del robot. Además, un análisis de casos de éxito en la utilización de los robots NXT en programas educativos universitarios, así como su uso en prototipos en trabajos de investigación.

Abstract

Nowadays, robotics is considered one of the main areas of interest in science and technology. In almost all sectors of the society we encounter the robotization, i.e., automated systems. In industry, robots are used for welding, transporting, assembling and painting pieces. In medicine, sophisticated robots help to conduct complex surgeries. In the military sector, robots are used to explore hostile areas. Therefore, mobile robotics and autonomous systems are topics of growing interest in Computer Science and Engineering careers. Therefore, in this paper we present a review of the main characteristics of the LEGO Mindstorms robots, examining the development environments available for compiling software programs that help to define the robot behavior. In addition, an analysis is carried out of success cases using NXT robots in university courses, as well as its use in research prototypes.

I. Introducción

En los últimos 100 años las personas han pensado, soñado y escrito sobre robots. Actualmente los robots son reales y pueden realizar una amplia variedad de trabajos importantes y desempeñar funciones específicas. Los robots se utilizan para realizar tareas de exploración en otros planetas, monitorear actividades volcánicas, estudiar comportamientos en aguas profundas, ensamblar automóviles y apoyar en cirugías de personas, entre otras muchas actividades.

Los robots móviles autónomos han sido ampliamente adoptados como instrumento de motivación para la introducción de la robótica y de la mecatrónica en todos los niveles educativos. Una gran cantidad de experiencias han sido reportadas sobre la construcción de robots y su utilización en la educación, tales como desarrollos de equipos de aficionados (Das, Yost, & Krishnan, 2010), vehículos comerciales de pequeña escala (Buiu, 2008) robots de juguete basados en Meccano (Parkin, 2002), y los robots LEGO Mindstorms NXT[1] (Erwin, Cyr, & Rogers, 2000; Hirst, Johnson, Petre, Price, & Richards, 2003). El equipo LEGO Mindstorms NXT (de ahora en más referido como NXT) es sin duda la plataforma educativa más utilizada, ya que tiene un costo relativamente accesible; es reutilizable, robusto y reconfigurable; y sus módulos son fáciles de utilizar por estudiantes de todos los niveles educativos (Kim & Jeon, 2009). Además, el equipo NXT contiene todos los elementos o partes necesarias para construir un robot funcional. Es ampliamente utilizado para propósitos educativos, así como para los experimentos de investigación en robótica (Griffin, 2010; Kelly & Smith, 2011).

Los orígenes del robot NXT se remontan a 1980 en el laboratorio del MIT Media (MIT del inglés, *Massachusetts Institute of Technology*), donde se inventó el concepto de *brick* (ladrillo) programable. Dos conceptos importantes a definir en el ambiente de la robótica y específicamente en el ambiente de los robots NXT son: *robot* y *programa*. En una primera definición de *robot*, se puede mencionar que es un dispositivo mecánico con forma humana que imita las acciones humanas (Kelly, 2010). También, se puede definir como una máquina electrónica que funciona de forma independiente, sin control humano (Kelly, 2010). Sin embargo, una definición más completa y que se utilizará como base en el presente artículo es la siguiente: un *robot* es un dispositivo que está diseñado para realizar acciones en forma independiente e interactuar con su entorno (Kelly, 2010). Entonces, un *robot* debe ser capaz de moverse y reaccionar en forma independiente, esto es, debe poder estudiar su entorno, responder a los obstáculos, tomar de decisiones (por ejemplo, elegir una bola roja dentro de un contenedor con una mezcla de bolas de diferentes colores), y otras actividades sin la ayuda de un humano.

1. LEGO, el logo de LEGO, MINDSTORMS, MINDSTORMS RCX, NXT, EV3 y el logo de MINDSTORMS son marcas registradas del Grupo LEGO; LEGO, the LEGO logo, MINDSTORMS, MINDSTORMS RCX, NXT, EV3, and the MINDSTORMS logo are trademarks of the LEGO Group.

Sin embargo, para que un robot pueda realizar estas acciones en forma independiente requiere de un programa de software. Un *programa* es un conjunto de instrucciones basadas en un lenguaje específico, que permiten al robot realizar una acción (Kelly, 2010). Entonces, cuando un programa se ejecuta, el robot colecta datos por medio de sus sensores y mueve los motores, de acuerdo con las instrucciones contenidas en el programa de software.

El presente trabajo tiene como objetivo, estudiar las principales características del robot NXT versión 2.0, así como analizar su funcionamiento y la operación de sus sensores y motores. También, se examinan los principales lenguajes de programación compatibles con el robot NXT, tanto lenguajes gráficos, como lenguajes basados en texto. Además, se presentan casos de éxito y experiencias de la utilización de robots LEGO NXT en diferentes niveles educativos.

II. Características de los robots NXT

El NXT es un equipo de robótica que parte de la idea de construir un robot mediante la unión de piezas y de la programación de acciones, de forma sencilla, a través del ensamblado de bloques con instrucciones concretas (LEGO Group, 2012). NXT es el segundo robot de la línea LEGO Mindstorms; sus principales características técnicas son: está conformado por una mini-computadora o *brick* (ladrillo) con una capacidad de almacenamiento en memoria RAM de 64 Kb e incluye cuatro puertos de entrada utilizados para conectar los diferentes sensores y tres puertos de salida usados para interconectar los motores (cada motor tiene embebido un sensor de rotación), los cuales habilitan la movilidad del robot. Estos motores se caracterizan por permitir movimientos precisos y controlados mediante el sensor de rotación, así como una sincronización en el tiempo de ejecución de una acción (rotación) con otros motores. Los sensores proporcionan información del mundo o contexto donde se ubica el robot, esto es, información de los objetos externos que rodean al robot NXT. Esta información es interpretada por una aplicación de software que permite detectar objetos, determinar su ruta de desplazamiento e identificar su ubicación o localización. Los sensores básicos en un robot NXT son: luz, sonido, tacto, color, ultrasónico y de rotación (integrado en los motores) (Griffin, 2010). En la Tabla 1 se presenta un resumen de las características técnicas de los diferentes modelos o versiones de los robots LEGO Mindstorms.

Los sensores se clasifican en dispositivos de entrada o de salida. Los dispositivos de entrada se sub-dividen en análogos y digitales. Los sensores de color, luz, sonido y tacto son dispositivos de entrada análogos, y el sensor ultrasónico se clasifica como dispositivo de entrada digital (NI, 2009). Además, existen sensores con mayor grado de especialización para el robot NXT, tales como posicionamiento (GPS), direccionamiento, temperatura, y aceleración, entre otros. Los sensores de rotación son considerados como dispositivos de salida. El motor NXT o servomotor contiene un sensor de rotación integrado para determinar cuánto gira el motor, que permite realizar movimientos muy precisos (Griffin, 2010; Kelly, 2010). Adicionalmente, mediante este

sensor se puede controlar el movimiento del robot NXT y la dirección que sigue en un trayecto, lo que permite controlar la velocidad y duración de la rotación.

El *sensor de luz* mide el nivel de brillo o luminosidad frente al sensor (Valk, 2010) y tiene la capacidad de distinguir entre los colores negro, blanco y tonos de gris. El sensor de luz es de gran utilidad para seguir una ruta mediante la detección de una línea de alguno de los colores soportados y también para medir el nivel de luminosidad de una fuente de luz. El sensor tiene un *led* en la parte frontal que emite una luz que permite medir tanto el reflejo de la luz, como la luz del ambiente (Brigandi, Field, & Wang, 2010). El *sensor de color* tiene las mismas funcionalidades del sensor de luz, con la capacidad adicional para distinguir con mayor precisión los diferentes colores de los objetos. El dato que genera el sensor al detectar un color será un valor numérico: 1 – negro, 2 – azul, 3 – verde, 4 – amarillo, 5 – rojo, y 6 – blanco (Griffin, 2010).

El *sensor de sonido* mide el nivel de ruido alrededor del robot NXT (Valk, 2010). Cuando se está trabajando con el sensor de sonido, es importante entender exactamente lo que el sensor está midiendo. Por ejemplo, los humanos cuando escuchamos un sonido podemos discernir sus características, incluyendo diferencias en intensidad y tono, incluso podemos identificar diferentes instrumentos musicales por los sonidos que emiten. Sin embargo, el sensor de sonido es menos sofisticado que nuestro sentido de la audición, únicamente tiene la capacidad de medir la intensidad de un sonido. Esta intensidad se mide con un número entre 0 y 100, siendo 0 el sonido más suave y 100 el sonido más alto (Bishop, 2008). El sensor de sonido tiene una capacidad de detección de ruido de hasta 90dB (Astolfo, Ferrari, & Ferrari, 2007).

El *sensor táctil* detecta cuándo se presiona el botón en la parte frontal del sensor (Bishop, 2008). El funcionamiento del sensor consiste en determinar el estado del sensor, y emitir una señal lógica de falso o verdadero. Entonces, si el estado que se busca en el sensor es activado se realizará una acción predeterminada en la aplicación de software. Los estados soportados por el sensor táctil son: presionado, liberado o pulsado (presionar-liberar) (Perdue & Valk, 2011). Este sensor se utiliza normalmente para detectar que el robot NXT se ha topado con un objeto o que alguna articulación del robot ha encontrado y tiene posesión de un objeto.

El *sensor ultrasónico* mide la distancia entre el sensor y un objeto (Perdue & Valk, 2011). El funcionamiento del sensor consiste en enviar ondas sonoras de alta frecuencia, para posteriormente medir el tiempo en que retorna la onda, lo que le permite al sensor determinar la distancia que existe entre el objeto y el sensor ultrasónico (Valk, 2010). La forma y textura de un objeto afectan en gran medida la precisión o exactitud del cálculo de la distancia realizado por el sensor ultrasónico; algunas superficies reflejan las ondas sonoras mejor que otras y, por lo tanto, son objetos más fáciles de detectar. Las superficies planas y sólidas son las más fáciles de detectar, ya que reflejan la mayor parte de las ondas sonoras de regreso hacia el sensor. Las superficies curvas reflejan algunas ondas sonoras en retorno hacia el sensor, pero algunas otras ondas van en diferentes direcciones. Además, los objetos suaves tienden a absorber las ondas sonoras

en lugar de reflejarlas. Por consiguiente, el sensor será capaz de detectar objetos planos y sólidos con mayor precisión y a una distancia mayor que objetos blandos y curvos. El sensor ultrasónico detecta objetos (frente al sensor) en un rango de 0 a 254 centímetros (100 pulgadas), con un margen de error de +/- 3 centímetros, y genera las ondas sonoras en una frecuencia de los 40kHz (Griffin, 2010).

La comunicación entre robots NXT se realiza vía inalámbrica a través de Bluetooth, en donde sólo pueden estar conectados un máximo de tres equipos, estableciéndose desde un inicio el estado de la conexión Bluetooth de cada robot, esto es, maestro o esclavo. Estas conexiones tienen que ser establecidas antes de ejecutar los programas de cada robot (Brigandi, Field, & Wang, 2010). La actualización de programas en el *brick* NXT se puede realizar mediante el puerto USB o por una conexión Bluetooth.

a) Breve historia de los robots LEGO Mindstorms

En 1998 LEGO introduce al mercado los robots Mindstorms como un juguete educativo orientado a niños y adolescentes. Este desarrollo fue derivado del trabajo colaborativo entre la empresa LEGO y el laboratorio de Ingeniería del MIT. El primer robot fue llamado RCX (*Robotic Command eXplorers*), compuesto por un *brick* con una capacidad de almacenamiento en la memoria RAM de 32Kb. Este modelo de *brick* incluye únicamente tres puertos de entrada para la conexión de los sensores y tres puertos de salida para la interconexión de los motores. En esta versión del robot se integra una bocina, la cual permite reproducir audio precargado al ejecutar alguna acción. La actualización de programas en el robot RCX, desde una computadora personal, se realiza mediante una conexión por un puerto infrarrojo, el cual además permite establecer una comunicación con otros robots RCX para ejecutar en conjunto alguna acción (Ver Figura 1A y Tabla 1 para mayor detalle).

En 2006 LEGO lanza al mercado el segundo robot de la línea Mindstorms, llamado NXT, compuesto por un *brick* dotado de mayor capacidad de procesamiento que su antecesor. Esta versión del NXT es reemplazada en 2009, iniciando la comercialización de la versión NXT 2.0 (descrita al inicio de esta sección). Las versiones NXT 1.0 y NXT 2.0 presentan similares características técnicas en el *brick*. La versión 2.0 del NXT presenta mejoras en el software de desarrollo e incluye nuevos sensores, como el sensor de color RGB, que reemplaza al sensor de luz (Calvo & Perianez, 2010). En la Figura 1B se muestra el brick NXT y en la Tabla1 se presenta un resumen de las características del robot NXT.

En julio del año 2013 se inició la comercialización del tercer modelo del robot LEGO Mindstorms, denominado EV3 (su nombre viene de la palabra EVolution) (Mindstorms, 2013). El *brick* de este modelo cuenta con una capacidad de 16Mb de memoria flash y 64Mb de memoria RAM, así como con una ranura lectora de tarjetas mini SD que soporta hasta una capacidad de 32Gb, la cual permite tener una mayor capacidad de almacenamiento. En la Figura 1C se ilustra el *brick* EV3. Con respecto a dispositivos para establecer la comunicación entre el *brick* y la computadora personal, el robot EV3

cuenta con un puerto USB 2.0, Bluetooth, y con una tarjeta de red inalámbrica (WI-FI) conectada mediante el puerto USB, que permiten programar al robot EV3 de forma inalámbrica. El sistema operativo del robot EV3 está basado en Linux y tiene capacidad de interconexión con dispositivos móviles inteligentes mediante la compatibilidad del sistema operativo con IOS y Android. En el modelo EV3 se incluyen nuevos sensores, tales como un sensor infrarrojo digital (IR), un sensor giroscópico, un sensor de color digital con capacidad de detectar ausencia de color y hasta siete diferentes colores y un generador de señal infrarroja, que puede ser usado también como control remoto del robot EV3.

Tabla 1. Resumen de las principales características técnicas de los robots LEGO Mindstorms

	RCX	**NXT**	**EV3**
Procesador	Hitachi H8/3292 10 – 16 MHz, 16 KB–ROM, 32 KB-RAM	Atmel 32-Bit - ARM7 48 MHz, 256KB flash, 64 KB RAM.	ARM 9 300 MHz, 16 MB – flash, 64 MB RAM
Sistema operativo	Propietario	Propietario	Abierto (basado en Linux)
Puertos	3 puertos para motores 3 puertos para sensores	3 puertos para motores 4 puertos para sensores	4 puertos para motores 4 puertos para sensores
Comunicación	Puerto IR en el frente del *brick* utilizado para comunicación con el equipo de cómputo y con otro RCX	USB 12 Mbps Bluetooth	Bluetooth v2.1 Wi-Fi mediante el puerto USB
Almacenamiento extra	N/A	N/A	Ranura Micro SD
Comunicación con dispositivos móviles inteligentes	N/A	Android	Android / iOS
Pantalla	LCD	LCD monocromática, 100 x 64 pixeles	LCD monocromática, 178 x 128 pixeles
Otras características.	-	Co-procesador Atmel 8-Bit AVR 8MHz, 4KB flash, 512 Byte RAM.	Auto-detección de dispositivos conectados al robot EV3. Control remoto (IR). Hilos de procesos. Compatible con los motores y sensores del robot NXT 2.0

Figura 1. Diferentes brick del robot LEGO Mindstorms (LEGO Group, 2012)

III. Aplicación de robots LEGO Mindstorms en la educación

Los robots son ampliamente utilizados en la educación en diversos niveles educativos (Cruz-Martín et al., 2012; Huang & Huang, 2011; Yu, 2012; Lofaro, Giang, & Oh, 2009; Shih, Chang, Chen, Chen, & Liang, 2012) y con diferentes objetivos pedagógicos. La utilización de robots NXT puede presentarse en áreas de ciencias, tecnología, ingeniería y matemáticas debido a la versatilidad que proporciona la característica de modularidad, que posibilita modificar físicamente el armado del robot mediante piezas ensamblables y la definición de su comportamiento en base a algoritmos especificados mediante programación visual o basada en texto.

El uso de robots NXT es una estrategia que permite observar los resultados en forma inmediata, en la cual los estudiantes aprenden, comprenden y aplican las bases de programación utilizándolo y observando la reacción del robot al ejecutar un programa, lo que le permite al alumno adquirir conocimientos en forma activa, con lo cual, se le proporciona al alumno un medio creativo e intuitivo de aprendizaje de materias del área de ciencias y tecnología (Fiorini, 2005; Tec et al., 2010).

En varios países se han fortalecido los programas educativos utilizando robots educativos, orientados a niños entre 7 y 14 años, con el objetivo de que el alumno se relacione con la robótica y las ciencias computacionales en una edad temprana (NASA, 2013; Worcester Polytechnic Institute, 2013; The Robotics Institute of Yucatán, 2013; Robotics Learning, 2013; LEGO Education UK, 2013). El aprendizaje apoyado con robots debe presentarse en un contexto en el cual los estudiantes trabajen en la búsqueda de soluciones de problemas reales (por ejemplo, construir un robot que detecta las luces de un semáforo y dependiendo de color de luz realice una acción determinada), en cuya búsqueda se incluya el diseño, la construcción, la programación, la prueba y la corrección del comportamiento del robot.

En la medida en que los estudiantes se involucren con la utilización de robots educativos desde edades tempranas hasta niveles más avanzados de educación (bachillerato o pre-universitario), será mayor su interés en la elección de programas educativos universitarios

relacionados con la ciencia computacional (Slangen, van Keulen, & Gravemeijer, 2011; Barak, & Zadok, 2009).

En un nivel universitario, en programas educativos de mecatrónica, el robot NXT es utilizado en cursos introductorios, –es el primer contacto del alumno con el contexto de la robótica–, permitiendo incrementar las habilidades de razonamiento, colaboración y trabajo en equipo en los alumnos (Stier, Zechel, & Beitelschmidt, 2011). En programas educativos de ciencias de la computación se ha implementado el uso de robot NXT para involucrar al alumno en el desarrollo de algoritmos y aplicaciones de software mediante el lenguaje de programación NXT-G. También, se han implementado cursos de programación orientada a objetos con la utilización del robot, en los cuales el alumno aplica conocimiento del paradigma orientado a objetos en el comportamiento del robot (Szweda, Wilusz, & Flotynski, 2012; Yu, 2012). Adicionalmente, se han desarrollado cursos de control de sistemas e inteligencia artificial, usando robots NXT, para fortalecer el aprendizaje de estos cursos mediante experimentos prácticos (Sugumaran, Nanal, Jain, & Wadoo, 2013; Yoonsoo, 2011). Además, en otros programas educativos como Ingeniería en Electrónica se han implementado cursos de métodos matemáticos, en los cuales los conceptos de matemáticas básicas son convertidos en algoritmos en MATLAB con el objetivo de controlar y definir el comportamiento de robots NXT basados en conceptos matemáticos (Behrens, Atorf, Schneider, & Aach, 2011). Por otro lado, en algunos programas educativos se han desarrollado manuales de prácticas utilizando robots NXT para cursos de adquisición de datos, ingeniería de sistemas de control, y sistemas en tiempo real, los cuales han facilitado la adquisición de conocimientos de los alumnos mediante la realización de prácticas (Cruz-Martín et al., 2012; Ierache, Garcia-Martinez, & De Giusti, 2009).

Los robots NXT se utilizan constantemente en diversas áreas de la investigación, en las cuales participan mayoritariamente estudiantes de licenciatura y posgrado como becarios. En Brigandi, Field y Wang (2010) se presenta un sistema multi-robot basado en NXT que permite mostrar ventajas de estos sistemas, tales como la redundancia, sus funcionalidades de manejo de espacio y la flexibilidad de reconfiguración, las cuales permiten una navegación satisfactoria de los robots que participan en un sistema de este tipo. Un sistema de asistencia al conductor basado en el contexto es presentado en Alghamdi, Shakshuki y Sheltami (2012). El sistema permite vincular a los conductores con el medio físico que los rodea. Esto se consigue mediante un sistema de alertas vehicular que ayuda a los conductores a evitar colisiones, mejorando los tiempos de respuesta. Para demostrar la viabilidad del sistema se implementa en el entorno de robots NXT. Una de las principales funcionalidades de los robots es la detección de objetos, para lo cual Paturca, Novischi, y Llas (2010) presentan un análisis y comparación de detección de objetos mediante sensores de visión y la capacidad de procesamiento de los robots. Se utilizan las plataformas de robots NXT y SRV-1 Blackfin para llevar a cabo la comparación, mediante la implementación de un algoritmo, evaluando la precisión del reconocimiento de los objetos, detección de varios objetos y el tiempo de

procesamiento requerido para tomar una decisión para evitar un obstáculo. También en el área de sistemas embebidos se han utilizado robots NXT para realizar prototipos o simulación de los sistemas de información. Santos, Tarrataca, y Cardoso (2010) presentan un enfoque de un algoritmo de navegación con control autónomo y proponen un sistema embebido de navegación en teléfonos móviles inteligentes, probando mapeo, localización, mapeo y localización simultáneo y planeación de ruta; utilizando robots NXT para ejecutar la navegación o ruta determina por el algoritmo que se ejecuta en el teléfono inteligente.

IV. Entornos de desarrollo de aplicaciones para Robot NXT

En esta sección se presentan los lenguajes de programación y entornos de desarrollo soportados por el robot NXT. Entre estos lenguajes de programación se pueden mencionar NXT-G (Griffin, 2010; Kelly, 2010), Robotics Developer Studio (Microsoft, 2012), Not eXactly C (Next Byte Codes, 2011), LeJOS (LeJOS NXJ, 2012), RobotC (Carnegie Mellon Robotics Academy- Robomatter, 2012), PbLUA (PbLua, 2012), y PyNXC (PyNXC, 2010). Sin embargo, se describen únicamente los lenguajes más utilizados en los laboratorios de las universidades para desarrollar aplicaciones de software en robots NXT.

A. NXT-G

El software oficial de programación de los robots NXT es el lenguaje de programación gráfico NXT-G (Kelly, 2010). Este lenguaje, desarrollado por las compañías LEGO y National Instruments, opera sobre la plataforma de software LabVIEW (NI, 2009). Todo el concepto de NXT-G se basa en bloques de programación, en donde un bloque puede ser personalizado para realizar una acción específica. Existen bloques para el control de los motores, para cada uno de los sensores soportados y para el control del flujo de datos u operación del robot. Entonces, un conjunto de bloques relacionados entre sí puede llevar a cabo una serie de acciones; la forma en que el programa se comporta depende de qué bloques se utiliza y cómo están unidos los bloques. El lenguaje de programación NXT-G está básicamente dirigido a niños y adolescentes con poca experiencia en la programación, sin embargo, también soporta programación avanzada y permite configuraciones y construcciones de robots con acciones complejas, posibilitando funcionalidades de registro de datos de los sensores y su análisis para la toma de decisiones, así como la integración y uso de constantes físicas, unidades de medida, sistemas de coordenadas, mínimos, máximos, medias y fórmulas lineales. En la Figura 2 se muestra la interfaz del lenguaje de programación NXT-G, desplegando en el centro de la imagen, diferentes bloques de programación y sus relaciones mediante conexiones de datos.

Con el fin de proporcionar un entorno de programación más flexible son incorporados dos bloques relacionados con los conceptos de programación en el software NXT-G: el bloque de tipo *switch* y el bloque de tipo *loop*. El bloque de tipo *switch* que se ilustra

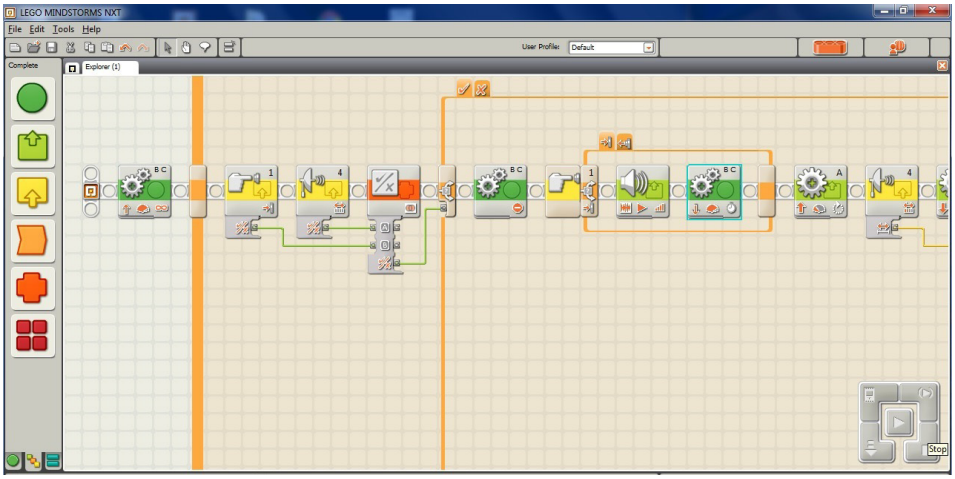

Figura 2. Interfaz del IDE NXT-G

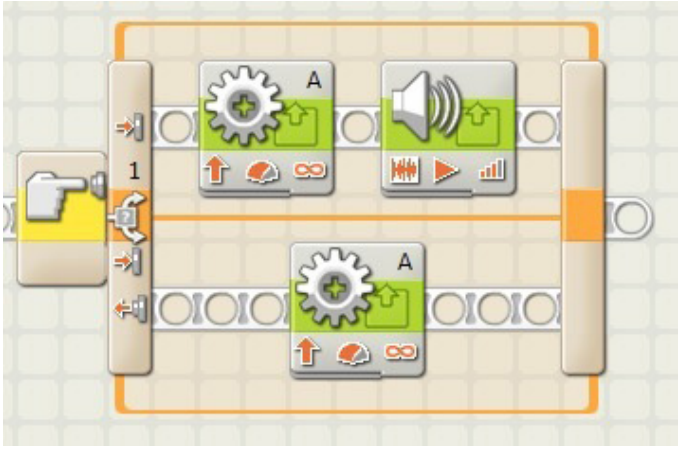

Figura 3. Bloque tipo *switch* del lenguaje NXT-G

en la Figura 3 se comporta en forma similar a la instrucción IF de otros lenguajes de programación, tales como Java o C. Por ejemplo, en la Figura 3, al configurarse un sensor de tacto seguido de un bloque *switch*, en donde si la condición es verdadera (si se presiona el sensor de tacto) se inician todos los bloques de la parta superior del bloque *switch*. De lo contrario, se inician todos los bloques de la parte inferior del bloque *switch*.

El bloque de tipo *loop* es de los más importantes que se han incorporado en el lenguaje de programación NXT-G; se comporta exactamente igual que la sentencia DO-WHILE. En la Figura 4 se muestra el bloque *loop*, en donde se garantiza que cualquier bloque colocado dentro de los corchetes cuadrados del *loop* se ejecutará una vez y se repetirá hasta que la condición sea falsa. En este tipo de bloque se debe establecer la condición que finalizará el ciclo, por ejemplo, el tiempo transcurrido, un número de repeticiones, una señal lógica o un dato colectado por un sensor. También, puede establecerse un bucle sin un fin definido.

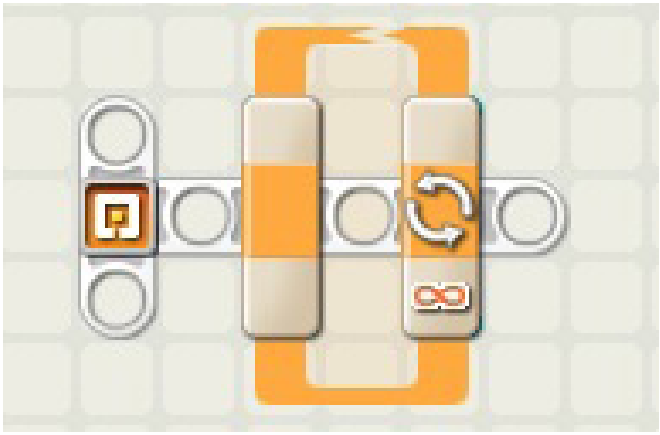

Figura 4. Bloque tipo *loop* del lenguaje NXT-G

B. Robotics Developer Studio

Microsoft (2012) ha desarrollado el software *Robotics Developer Studio* [RDS] que consiste en un entorno de desarrollo que permite construir aplicaciones para controlar diversos tipos de robots. Es un entorno de desarrollo integrado .NET para el diseño, ejecución y depuración, altamente escalable y concurrente de aplicaciones de robótica distribuida, que permite hacer frente a los retos actuales del desarrollo de aplicaciones para robots, como son la coordinación, la *observabilidad*, la configuración, la implementación y la reutilización. RDS proporciona un entorno de programación visual similar al entorno de desarrollo de LEGO.

Las principales funcionalidades de RDS son: una arquitectura de ejecución extensible, la cual soporta desarrollo de aplicaciones para robots que utilizan procesadores de 8-bits, 16-bits, o 32-bits, así como robots que utilizan múltiples procesadores; un conjunto de herramientas útiles para la programación y depuración (*debugging*) en un entorno de simulación, que permiten al depurarse los programas, incrementar su eficiencia al ser ejecutados por el robot; y un conjunto de servicios de librerías que facilitan el desarrollo de aplicaciones para el robot. Además, incluye un entorno de programación visual con una interfaz de arrastrar y soltar (*drag-and-drop*) que permite que el usuario pueda programar robots sin tener conocimientos avanzados de programación.

En la Figura 5 se muestra un ejemplo del entorno de desarrollo RDS que ilustra cómo el sensor ultrasónico envía notificaciones con las actualizaciones cada vez que realiza una colecta de datos. La acción a ejecutar por el robot NXT estará condicionada a la distancia en que se encuentre el objeto detectado por el sensor ultrasónico, lo que se representa mediante un bloque de decisión IF.

C. Not eXactly C

Not eXactly C (NXC) es un lenguaje de alto nivel similar al lenguaje C, construido sobre el compilador NBC (del inglés *Next Byte Codes*) (Next Byte Codes, 2011). Para ejecutar un programa de NXC en el *brick* LEGO, el compilador de NXC traduce el

programa fuente en código de bytes de NXT (Benedettelli, 2008). Las estructuras de control de NXC son similares a C, sin embargo, NXC no es un lenguaje de programación de propósito general debido a las restricciones que se derivan del intérprete de código de bytes de NXT. La lógica interna del NXC se definió en dos componentes: el lenguaje NXC, que describe la sintaxis utilizada para escribir los programas, y la interfaz de programación de aplicaciones (API, del inglés *Application Programming Interface*) que describe las funciones del sistema, las constates y los macros que pueden ser utilizados por los programas.

NXC es el lenguaje no oficial más utilizado para programar robots NXT, en combinación con el IDE (del inglés *Integrated Development Environment*) *Bricx Command Center* (BricxCC) (BricxCC, 2012). El BricxCC se utiliza para escribir, compilar y probar los programas, además de incluir importantes herramientas, tales como (BricxCC, 2012; Benedettelli, 2008):

- Utilería de control directo. Permite controlar los motores y sensores del robot, posibilitando establecer el tipo y modo de operación de los sensores. Esta utilería fue desarrollada para ser utilizada con fines de depuración de los programas.
- Diagnóstico. Esta herramienta despliega toda la información disponible sobre el robot NXT conectado. Muestra la versión del firmware, la carga de batería disponible, el método de conexión (USB o Bluetooth), la dirección del Bluetooth, el nombre del NXT y la memoria disponible.
- Monitor del *brick*. Esta utilería permite obtener la información que los sensores están capturado, los parámetros de los motores y los mensajes generados, en tiempo de ejecución del programa.
- Monitor configurable. Esta herramienta tiene una función similar a la anterior más la característica de poder seleccionar los sensores o motores a monitorear.
- Explorador de archivos NXT. Esta herramienta es un explorador de archivos de la memoria flash del robot NXT. Permite administrar los archivos, esto es, cargar, descargar, borrar y desfragmentar la memoria.
- Pantalla NXT. Esta utilería permite visualizar y capturar el contenido de la pantalla del robot NXT en la PC, así como presionar los botones del NXT en forma remota. Es de gran utilidad cuando el robot se encuentra en movimiento.
- Convertidor de sonido. Permite convertir archivos MIDI o WAV en código o en un archivo RSO, para su utilización en el robot.

Por otro lado, PyNXC es un proyecto *open source* basado en NXC que tiene como funcionalidad convertir el código generado en el lenguaje de programación *Python* a código en el lenguaje NXC (Blais, 2010), de manera que el programa sea compilado con el NXC y descargado en el robot NXT para su ejecución, sin necesidad de reemplazar el *firmware*. Este proyecto presenta importantes limitaciones, ya que se encuentra en etapas tempranas de desarrollo.

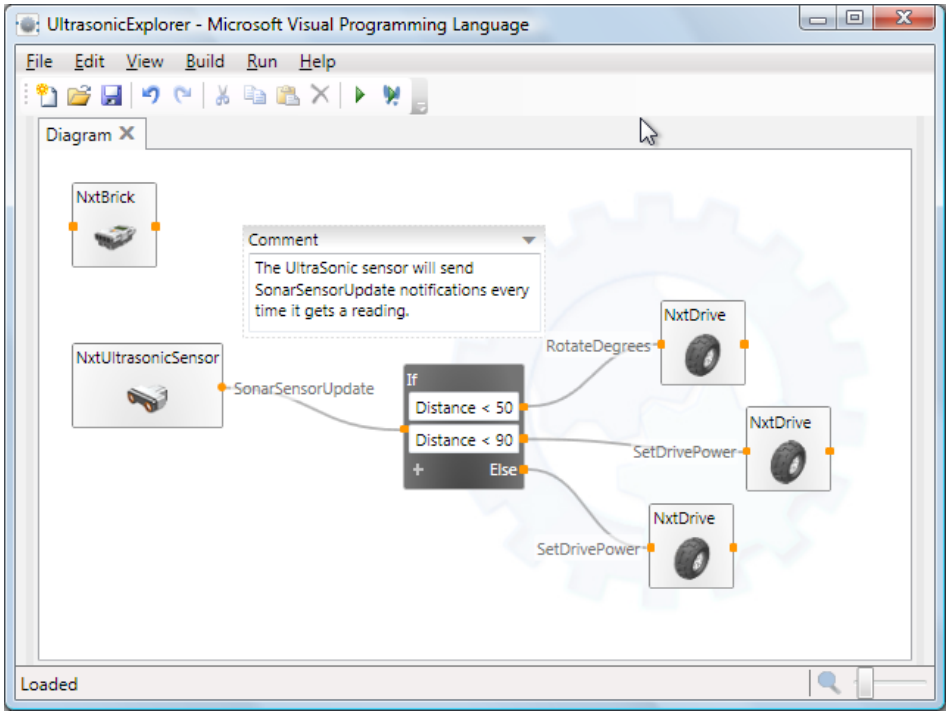

Figura 5. Interfaz del IDE del lenguaje RDS

D. LeJOS

LeJOS es un proyecto *open source* creado para desarrollar aplicaciones de software para productos de LEGO Mindstorms utilizando la tecnología Java (LeJOS NXJ, 2012). El ambiente de desarrollo LeJOS NXJ está orientado hacia los robots de tecnología NXT, RCX, y EV3. Mediante el proyecto de LeJOS NXJ se desarrolló una máquina virtual de Java (JVM, del inglés *Java Virtual Machine*) para los robots LEGO Mindstorms reemplazando el firmware original del *brick* NXT, permitiendo así que los robots NXT puedan ejecutar aplicaciones de software basadas en el lenguaje de programación Java (Lew, Horton, & Sherriff, 2010), lo que permite disponer de extensas librerías de clases que soportan funciones de alto nivel, tales como la navegación y robótica basada en el comportamiento.

LeJOS NXJ es un lenguaje orientado a objetos, que soporta importantes funcionalidades, tales como: recursividad, arreglos y matrices multidimensionales, sincronización, manejo de hilos y excepciones (Breña Moral, 2009; LeJOS NXJ, 2012). También, posibilita el uso de tipos de datos en Java como son punto flotante, *string* y entero.

Por otro lado, LeJOS cuenta con un API que permite la comunicación entre las aplicaciones (desarrolladas con LeJOS) en una computadora personal o un dispositivo móvil con los programas que se estén ejecutando en el robot NXT utilizando Java

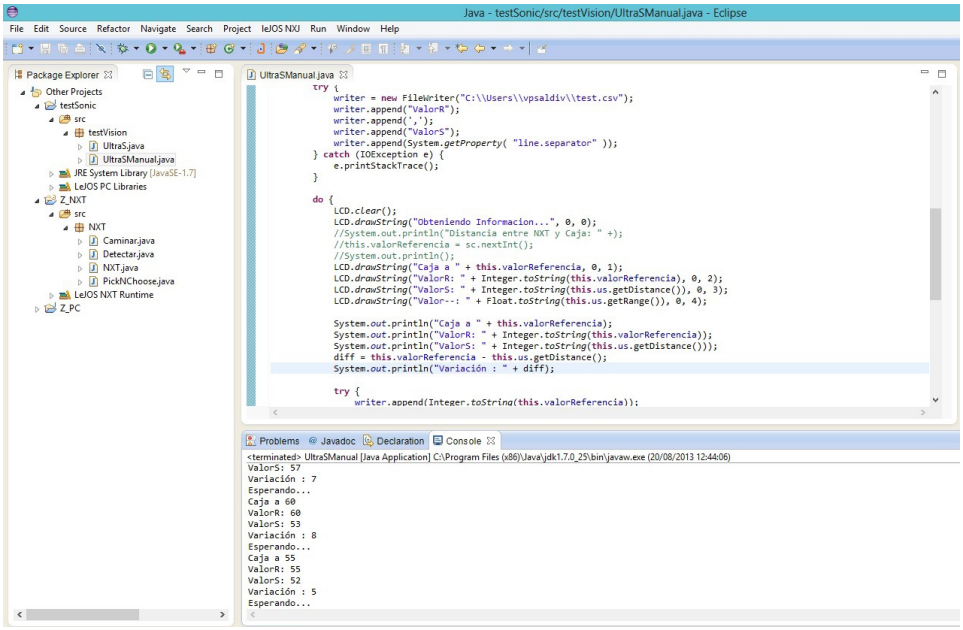

Figura 6. Interfaz del IDE Eclipse con plug-in de LeJOS

streams sobre Bluetooth, Wi-Fi o el puerto USB (Oliveira, Silva, Lira, & Reis, 2009; LeJOS NXJ, 2012). Lo anterior, permite que los datos colectados por el robot NXT se almacenen y procesen en la computadora cliente, generando mapas, gráficas o estadísticas. Adicionalmente, se han desarrollado *plug-in* que permiten la utilización de los IDE de Eclipse y de NetBeans, mejorando el entorno de desarrollo (ver Figura 6).

Conclusiones

El presente artículo revisa las características técnicas de los robots LEGO Mindstorms, detallando los principales sensores usados por los robots NXT y comparando las versiones existentes de los robots LEGO Mindstorms. Estos robots son ampliamente utilizados en las instituciones de educación con el objetivo de motivar a los alumnos para establecer una relación con la ciencia y tecnología a través de la robótica, estimulando la creatividad, colaboración y trabajo en equipo.

En las universidades, los robots móviles y los sistemas autónomos son un tópico de gran importancia que en los últimos años se ha incorporado en los programas educativos de Ingeniería y Ciencias Computacionales. Además, en los centros de investigación son utilizados los robots NXT para desarrollar simulaciones y prototipos de robótica, definición y prueba de algoritmos de comportamiento y algoritmos de toma de decisiones basado en el ambiente o contexto.

En el análisis que se realizó de los lenguajes de programación para robots NXT, se

destacan NXT-G y LeJOS por sus funcionalidades y paradigmas de programación. NXT-G, es un lenguaje gráfico basado en bloques de programación que puede ser utilizado por usuarios con bajo conocimiento de programación, únicamente siguiendo la intuición y lógica para que el robot realice una acción específica o para definir la solución a un problema existente en el contexto. Este lenguaje también permite una programación compleja utilizando conexiones de datos entre los bloques, lo cual habilita la toma de decisiones basada en los datos colectados mediante los sensores del robot. Por otra parte, mediante el lenguaje basado en texto (LeJOS) se posibilita especificar el comportamiento del robot mediante programación orientada a objetos, además de incorporar importantes funcionalidades en el lenguaje basadas en Java, tales como recursividad, arreglos y matrices dimensionales, así como disponer de librerías de clases que permite funciones de alto nivel como robótica basada en comportamiento.

El trabajo futuro se enfocará en la definición de un algoritmo y su programación mediante el lenguaje LeJOS, para los robots NXT y EV3, que disponga de funcionalidades que permitan identificar objetos, determinar sus dimensiones y calcular el espacio disponible entre dichos objetos, con la meta de determinar la mejor ruta para alcanzar un punto objetivo sin obstáculos con el menor costo posible.

Referencias bibliográficas

Alghamdi, W., Shakshuki, E., & Sheltami, T. R. (2012). Context-Aware DriverAssistance System. *Procedia Computer Science, 10*, 785-794

Astolfo, D., Ferrari, M., & Ferrari, G. (2007). *Building Robots with LEGO Mindstorms NXT*. Burlington, MA: Syngress - Elsevier

Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, tecnology and problem solving. *International Journal of Technology and Design Education, 19*(3), 289-307

Behrens, A., Atorf, L., Schneider , D., & Aach, T. (2011). Key Factors for Freshmen Education using MATLAB and LEGO Mindstorms. *Intelligent Robotics and Applications*, 553-562.

Benedettelli, D. (2008). *Creating Cool MINDSTORMS NXT Robots*. New York, NY: Apress-Springer.

Bishop, O. (2008). *Programming LEGO Mindstorms NXT*. Burlington, MA: Syngress Publishing - Elsevier

Blais, B. S. (2010). Using Python to Program LEGO Mindstorms Robots: The PyNXC Project. *The Python Papers, 5* (2), 1-7

Breña Moral, J. A. (2009). *Develop LeJOS Programs Step by Step*. Madrid, España: JAB

BricxCC. (2012). *Bricx Command Center 3.3*. Recuperado de http://bricxcc.sourceforge.net/

Brigandi, S., Field, J., & Wang, Y. (2010).

A LEGO Mindstorms NXT Based Multirobot System. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 135-139). Montreal, Canadá: IEEE

Buiu, C. (2008). Hybrid Educational Strategy for a Laboratory Course on Cognitive Robotics. *IEEE Transactions on Education*, *51*(51),100-107

Calvo, I., & Perianez, G. (2010). Uso conjunto de la plataforma LEGO Mindstorms NXT y metodologias PBL en informatica industrial. *Ikastorratza e-Revista de didáctica* (6), 2-18

Carnegie Mellon Robotics Academy-Robomatter (2012). *RobotC a C Programming Language for Robotics.* Recuperado de http://www.robotc.net/

Cruz-Martín, A., Fernandez-Madrigal, J., Galindo, C., Gonzalez-Jiménez, J., Stockmans-Daou, C., & Blanco-Claraco, J. (2012). A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduated courses. *Computers & Education*, *59*(3), 974-988.

Das, S., Yost, S. A., & Krishnan, M. (2010). A 10-Year Mechatronics Curriculum Development Initiative: Relevance, Content, and Results - Part I. *IEEE Transactions on Education*, *53*(2), 194-201

Erwin, B., Cyr, M., & Rogers, C. (2000). LEGO Engineer and Robolab: Teaching Engineering with LabVIEW from Kindergarten to Graduate School. *International Journal of Engineering Education*, *16*(3),181-192

Fiorini, P. (2005). LEGO Kits in the Lab. *IEEE Robotics & Automotion Magazine*, *12*(4), 5.

Griffin, T. (2010). *The Art of LEGO Minstroms NXT-G Programming.* San Francisco, CA: No Starch

Hirst, A. J., Johnson, J., Petre, M., Price, B. A., & Richards, M. (2003). What is the best programming environment/ language for teaching robotics using Lego Mindstorms. *Artificial LIfe and Robotics*, *7*(3), 124-131.

Huang, K. H., & Huang, P.-L. (2011). Lego Robotics and Group Learning: exploring the effects of gender, age and family cackground. *In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on* (pp. 219-222). Piscataway, NJ: IEEE

Ierache, J., Garcia-Martinez, R., & De Giusti, A. (2009). A Proposal of Autonomus Robotic System Educative Environment. *Springer Berlin Heidelberg*, 224-231

Kelly, J. F. (2010). *LEGO Mindstorms NXT-G Programming Guide.* New York, NY: Apress-Springer

Kelly, J. F., & Smith, C. (2011). *The LEGO Mindstorms NXT: Mars Base Command.* New York, NY: Apress-Springer

Kim, S. H., & Jeon, J. W. (2009). Introduction for Freshmen to Embedded Systems using LEGO Mindstorms. *IEEE Transactions on Education, Vol. 52, No. 1*, 99-108.

LEGO Education UK. (2013). *Brick by brick* [portal web]. Recuperado de http://legoeducationuk.wordpress.com

LEGO Group. (2012). *LEGO Mindstorms*

[portal web]. Recuperado de http://mindstorms.lego.com/en-us/default.aspx

LeJOS NXJ. (2012). *LeJOS Java for LEGO Mindstorms* [portal web]. Recuperado de http://lejos.sourceforge.net/index.php

Lew, M. W., Horton, T. B., & Sherriff, M. S. (2010). Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course. *23rd IEEE Conference on Software Engineering Education and Training (CSEE&T 2010)* (pp. 121-128). Pittsburgh, PA: IEEE Computer Society.

Lofaro, D., Giang, T., & Oh, P. (2009). Mechatronics education: from paper design to product prototype Using LEGO NXT Parts. En *Progress in Robotics*, (pp.232-239).

Microsoft. (2012). *Microsoft Robotics Developer Studio 4* [portal web]. Recuperado de http://www.microsoft.com/robotics/

Mindstorms, E.R. (2013). *Mindstorms EV3 Robots* [portal web]. Recuperado de http://www.mindstormsev3robots.com/

National Aeronautics and Space Administration [NASA]. (2013). *The Robotics Alliance Project*. Recuperado de http://robotics.nasa.gov/students/summer_camps.php

National Instruments [NI]. (2009). *LabVIEW. Lego Mindstorms NXT Module Programming Guide*. Austin, TX: NI

Next Byte Codes. (2011). Welcome to Next Byte Codes, Not eXactly C, and SuperPro C. Recuperado de http://bricxcc.sourceforge.net/nbc/

Oliveira, G., Silva, R., Lira, T., & Reis, L. P. (2009). Environment Mapping using the Lego Mindstorms NXT and LeJOS NXJ. *14th Portuguese Conference on Artificial Intelligende, EPIA 2009* (pp. 267-278). Aveiro, Portugal: Universidade de Aveiro

Parkin, R. M. (2002). The mechatronics workbench. *Engineering Science and Education Journal, 13*(1), 36-40

Paturca, S., Novischi, D., & Llas, C. (2010). Performance Comparison of Vision Sensors and Processing Power of Two Robotic Platforms for Obstacle Avoidance. *Research and Education in Robotics - EUROBOT 2010* (págs. 108-117). Rapperswil-Jona, Suiza: Springer

PbLua. (2012). *PbLua - Lego Is Just A Hobby - Right?* Recuperado de http://hempeldesigngroup.com/lego/pblua/

Perdue, D. J., & Valk, L. (2011). *The unofficial LEGO Mindstorms NXT 2.0 Inventor's Guide*. San Francisco, CA: No Starch

PyNXC. (2010). *PyNXC - A Python to NXC Converter for programming LEGO MINDSTORMS Robots*. Recuperado de https://code.google.com/p/pynxc/

Robotics Learning. (2013). *Robotics learning*: LEGO Robotics activities and classes. Recuperado de http://www.roboticslearning.com

Santos, A. C., Tarrataca, L., & Cardoso, J. M. (2010). The Feasibility of Navigation Algorithms on Smartphones using J2ME. *Mobile Networks and Applications, Vol. 15, No. 6*, 1572-8153.

Shih, B.-Y., Chang, C.-J., Chen, Y.-H., Chen, C.-Y., & Liang, Y.-D. (2012). Lego NXT Information on Test Dimensionality using Kolb´s Innovative Learning Cycle. *Natural Hazards*, *64*(2), 1527-1548

Slangen, L., van Keulen, H., & Gravemeijer, K. (2011). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education*, *21*(4), 449-469

Stier, J., Zechel, G., & Beitelschmidt, M. (2011). A robot competition to encourage first-year students in mechatronic sciences. *Research and Education in Robotics - EUROBOT 2011* (pp. 288-299). Praga, República Checa: Springer-Verlag Berlin Heidelberg

Sugumaran, R., Nanal, H., Jain, R., & Wadoo, S. (2013). Establishing a cost effective embedded control and robotics engineering program: Observer based state feedback control using LEGOs. *2013 IEEE Integrated STEM Education Conference (ISEC)* (pp. 1-6). Princeton, NJ: IEEE Computer Society

Szweda, L., Wilusz, D., & Flotynski, J. (2012). Application of NXT based robots for teaching java-based concurrency. En *e-Learning and Games for Training, Education, Healt and Sports* [Lecture Notes in Computer Science, V.7516] (pp.54-64), Berlín, Alemania: Springer-Verlag

Tec, B., Uc, J., Gonzalez, C., Garcia, M., Escalante, M., & Mantañez, T. (2010). Analisis comparativo de dos formas de enseñar matemáticas básicas: robots lego nxt y animación con scratch. *Memorias de la Conferencia Conjunta Ibero-Americana sobre Tecnologias para el Aprendizaje*, (pp. 103-109). Disponible en http://www.tizimin.uady.mx/filesWeb/AnalisisComparativoScratchVsLego.pdf

The Robotics Institute of Yucatán. (2013). *The Robotics Institute of Yucatán* [portal web]. Recuperado de http://www.triy.org/

Valk, L. (2010). *The LEGO Mindstorms NXT 2.0 Discovery Book*. San Francisco, CA: No Starch

Worcester Polytechnic Institute. (2013). *Frontiers*. Recuperado de http://www.wpi.edu/academics/k12/frontiers.html

Yoonsoo, K. (2011). Control Systems Lab Using a LEGO Mindstorms NXT Motor System. *IEEE Transactions on Education*, *54*(3), 452-461

Yu, X. (2012). Using LEGO Mindstorms in the undergraduate curriculum of IT. *2012 International Symposium on Information Technology in Medicine and Education* (pp. 270-273). Piscataway, NJ: IEEE

Currículum vitae

Edgar Tello Leal

Es Licenciado en Computación Administrativa (TI) por la Universidad Autónoma de Tamaulipas, México y recibió el grado de Doctor en Ingeniería en Sistemas de Información por la Universidad Tecnológica Nacional de la República Argentina en Noviembre de 2012. Actualmente trabaja como Profesor de Tiempo Completo e Investigador en la Universidad Autónoma de Tamaulipas, México. Las líneas de investigación actuales son gestión de procesos de negocio, desarrollo dirigido por modelos, sistemas de información basados en agentes de software, y sistemas basados en conocimiento.

Tania Y. Guerrero Meléndez

Es Ingeniera en Telemática por la Universidad Autónoma de Tamaulipas y Máster en Ingeniería con opción en Telecomunicaciones por la Universidad Politécnica de Cataluña, en 2009. Actualmente se desempeña como profesor de Tiempo Completo en la Universidad Autónoma de Tamaulipas, México. Sus intereses actuales de investigación incluyen comunicaciones móviles, robótica y tecnologías aplicadas a la educación.

Vicente Saldivar Alonso

Es profesor de la carrera de Ingeniería en Telemática en la Universidad Autónoma de Tamaulipas. Obtuvo el grado de Ingeniero en Telemática por la Universidad Autónoma de Tamaulipas y la Maestría en Ingeniería por la Universidad Politécnica de Cataluña. Sus áreas de interés incluyen: redes inalámbricas, redes de sensores e inteligencia artificial para su aplicación en ambientes inteligentes y robótica.