"Experiencia de laboratorio en Robótica: De las Tortugas Simuladas a los Robots y Juguetes Autónomos una Oportunidad Para la Enseñanza de Programación"

J. Ierache [(1)] M. Bruno [(1)], N. Mazza [(1)]
[(1)]Instituto de Sistemas Inteligentes y Enseñanza experimental de la Robótica (ISIER)
Facultad de Informática Ciencia de la Comunicación y Técnicas Especiales
Universidad de Morón Cabildo 134, (B1708JPD) Morón, Buenos Aires, Argentina
54-11-56272000 interno 189/746
[(1)] {jierache, mabruno, nmazza} @unimoron.edu.ar

Abstract

The experiences are framed in the context of the Technologies in Education, in the matter of robotics, those appear specifically that contribute to the conformation of a laboratory in robotics, integrated basically by robots classic, biped and quadruped which they are in the category of independent toys, this last characteristic facilitated an ample access to different universities, institutes and individuals of different parts from the world motivated by his low cost and the documentation available, generating the development of sensors, specific environments of programming in C, JAVA, among other languages, like thus also specific developments for necessary firmware. Against this background, the education of programming applied through physical agents robots allows the interaction between agents and of these with the performance atmosphere. The tools presented with the application of robots physical (Lego RCX, NXT, Robosapien, Robopet) compared with the simulated turtles (logo), present a strategic opportunity for the formation of human resources.

Key words: Robotics, Technologies in Education, Programming, Intelligent Systems, Agents and Multiagents

Resumen

Las experiencias se enmarcan en el contexto de las Tecnologías en Educación, en materia de robótica, se presentan específicamente aquellas que contribuyen a la conformación de un laboratorio en robótica, integrado básicamente por robots clásicos, bípedos y cuadrúpedos que se encuentran en la categoría de juguetes autónomos, esta última característica facilitó un amplio acceso a distintas universidades, institutos y particulares de distintas partes del mundo motivados por su bajo costo y la documentación disponible, generando el desarrollo de sensores, ambientes específicos de programación en C, JAVA, entre otros lenguajes, como así también desarrollos específicos para el necesario firmware. En este contexto, la enseñanza de programación aplicada a través de agentes robots físicos permite la interacción entre agentes y de estos con el ambiente de actuación. Las herramientas presentadas con la aplicación de robots físicos (Lego RCX, NXT, Robosapien, Robopet) comparadas con las tortugas simuladas (logo), presenta una oportunidad estratégica para la formación de recursos humanos.

Palabras claves: Robótica, Tecnologías en Educación, Programación, Sistemas Inteligentes, Agentes y Multiagentes.

1. INTRODUCCION

La emoción de ver como una tortuga deambula en nuestro monitor esquivando obstáculos virtuales hasta llegar a su meta en la esquina de un monitor [16] no tiene el mismo impacto emocional que

observar como un agente robot implementado sobre la base de juguetes autónomos puede esquivar obstáculos hasta llegar a su meta en el rincón de una habitación e interaccionar con nosotros a través de nuestro celular. Existen proyectos relacionados entre los que caben destacar el que se lleva a cabo en la Universidad Estatal de Campinas (UNICAMP); Brasil, donde se utiliza como base al lenguaje Logo con fines educacionales [27]. En particular, trabajan con SuperLogo, una versión del lenguaje LOGO desarrollada por el NIED, que permite el control del ladrillo programable LEGO RCX, que reproduce los movimientos de la tortuga en pantalla. El control del RCX se puede hacer por medio de una PC, en forma local o remota, utilizando la Internet. En la Universidad de Medford/Somerville en Massachussets, utilizan Lego RCX, Robolab y LabView para la enseñaza con distintos niveles de complejidad [24]. Por ejemplo en las clases de Introducción a la Ingeniería Mecánica. También los utilizan en cursos de robótica para estudiantes de primer año y en cursos avanzados de robótica con desarrollo de algoritmos de alto nivel e inteligencia distribuida.
Muchos proyectos involucran un control centralizado clásico, la computadora le dice al motor uno que se encienda, gire en sentido de las agujas del reloj, a media potencia bajo una secuencia planificada de acciones, pero también se puede aplicar el mismo agente robot para explorar sistemas descentralizados y de conductas autoorganizadas [1], [2], [28], [29]. Por ejemplo si consideramos un agente que deambula por su hábitat el que cuenta con áreas iluminadas y áreas oscuras, nuestro agente cuenta con dos reglas, una que le indica al agente moverse hacia delante cuando detecta áreas iluminadas y moverse hacia atrás cuando detecta áreas oscuras, el agente deambula hasta que llega a una sombra, entonces retrocede hasta que sale de ella y luego hacia delante otra vez, sigue así oscilando en el borde de la sombra; en este caso podemos considerar a nuestro agente robot como un criatura que detecta bordes, en si esta capacidad no esta explícitamente declarada en sus dos reglas, en realidad es una conducta grupal que surge de la interacción de las dos reglas, algo similar a como la conducta de una bandada surge de la interacción entre los pájaros [22].
Los estudiantes tienden a considerar a sus criaturas según diferentes niveles en distintos momentos, a veces ven a sus criaturas a un nivel mecanicista, al analizar como una pieza mueve a otra, otras veces pasa a nivel de información y exploran como pasa la información entre la computadora y los motores y sensores. En otras ocasiones los estudiantes consideran a las criaturas en un nivel psicológico, atribuyendo a la criatura intencionalidad o personalidad. Una criatura quiere ir a la luz, a otra criatura le gusta la oscuridad, otra le teme a los ruidos, los estudiantes pasan rápidamente entre estos niveles y aprenden en función de la situación de contexto que nivel es el mejor, piensan en sistemas en termino de niveles [22]. La idea de aprender mediante el diseño es un aspecto de lo que Seymour Papert [19] denominó como enfoque construccionista del aprendizaje y la educación. Los seres humanos construyen su conocimiento con particular eficacia cuando participan en la construcción de productos que son afectivos.

2. CARACTERÍSTICAS DE LOS AGENTES ROBOTS Y ENTORNOS DE PROGRAMACIÓN

El objetivo de nuestro trabajo es brindar una orientación general de los Kits de Robótica económicos actuales de Lego RCX y el reciente NXT, sus herramientas de programación en el ambiente de LabView, Robot C, NQC, entre otras. En materia de Juguetes autónomos se detallaran las características más relevantes de “Robosapien”. (robot bípedo) y de “Robopet, Robotail, Roboraptor”(robots cuadrúpedos), se considera además las interfases de comunicación y herramientas de programación, en particular GoRobo.

2.1- NXT y RCX, los robots de Legomidstorms

NXT, la nueva generación de robots de Legomindstorms [14] (Figura 1) ofrece características mejoradas con respecto a su anterior versión el RCX. En una comparación de las principales características de ambos kits se pueden mencionar:

Tabla Características de RCX Vs NXT

	LegoMidstorms RCX	Legomidstorms NXT
Ladrillo inteligente	Ladrillo RCX, con 5 slots para guardar programas del usuario.	Ladrillo NTX, que tiene una mayor capacidad de cómputo que el ladrillo RCX. Incluye funciones para testear los sensores (función Try me). Permite la personalización de los sonidos que el NXT puede reproducir, etc.
Puertos para Motores/ Sensores	Tres puertos para motores	Tres puertos para motores.
	Tres puertos para sensores	Cuatro puertos para sensores.
Sensores/ Motores	1 Sensor de luz, que permite distinguir diferentes niveles de luminosidad y oscuridad. 2 Sensores de tacto, que pueden detectar tres estados: o Presionado. o Libre. o Rebote.	1 Sensor de Luz 1 Sensor de Sonido, que tiene dos modos de configuración: o Para detectar sonidos perceptibles por el oído humano. o Para detectar además los sonidos no perceptibles por el oído humano (como ultrasonidos). 1 Sensor de Ultrasonido que actúa como un radar permitiendo la detección de objetos. Se lo puede configurar para detectar objetos lejanos o cercanos. Detecta objetos entre 0 y 255 cm con una precisión de +- 3 cm. 2 Sensores de tacto, que pueden detectar tres estados: o Presionado. o Libre. o Rebote.
	3 Motores.	3 Servomotores, mejorados con respecto a la versión RCX, ya que tienen integrados sensores de rotación. De este modo se pueden lograr movimientos mucho más precisos y controlados, además de que los dos motores se puedan sincronizar perfectamente.
Comunicación/ Descarga de programas	La descarga de programas se hace a través de la torre de infrarrojos que incluye el kit. La comunicación con otros RCX es través de infrarrojo también	El NXT posee un puerto USB, para la descarga de programas. Este kit incluye además Comunicación Bluetooth inalámbrica, permitiendo descargar programas de este modo, así como también la interacción con celulares y también con PC y laptops, etc. Permitiendo utilizar la capacidad de cómputo de otros equipos. La comunicación con otros NXT también es vía bluetooth
Otras características	1 parlante.	1 parlante de alta fidelidad, mejorado con respecto al del RCX.
	conexiones cableadas	Las conexiones para sensores y servomotores son RJ11.

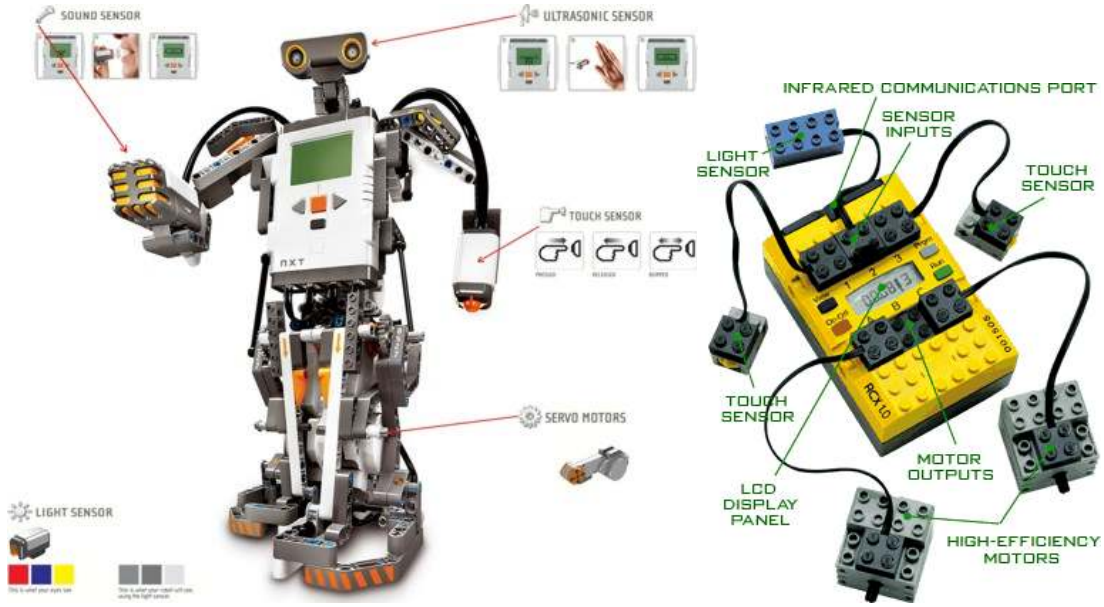

Figura 1: Legomindstorms NXT (izq) y RCX (der)

Estos agentes robots son programables en entornos gráficos nativos, para RCX [14] (figura 2, izq) y LabView [17] para NXT (figura 2, der). En cuanto a LabView cabe destacar que fue desarrollado por National Instruments y utilizado por la NASA, para monitorear y controlar al robot Sojourner Rover, en la misión de exploración de la superficie de Marte [17].

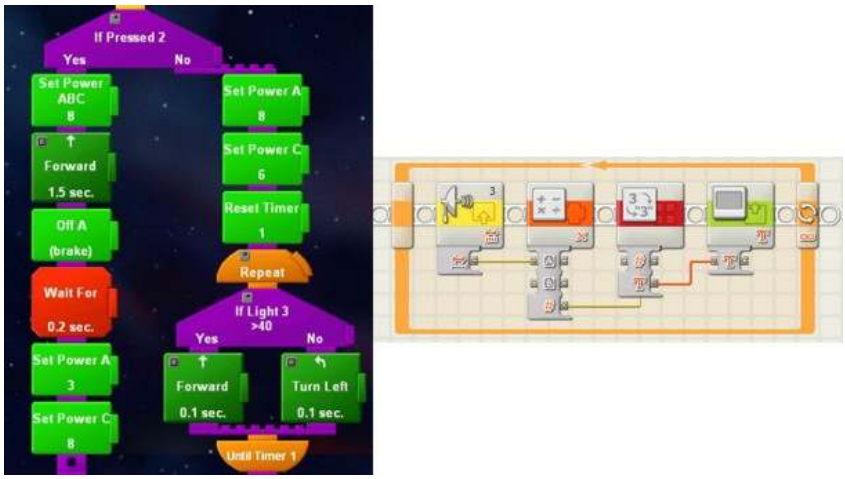

Figura 2: Ambiente de programación de RCX (izq) y LabView (der)

Estos entornos utilizan bloques que se ensamblan para formar un programa completo. Estos bloques incluyen control de motores (avance, retroceso, encendido y apagado), ciclos repetitivos (while, repeat), estructuras de control (if else), adquisición de datos de los sensores, utilización de variables, constantes y timers. Además de estos entonos gráficos existen una serie de programas que permiten su programación en códigos más tradicionales, como por ejemplo en Java. Es el caso del API Lejos para RCX y iCommand para NXT [15]. Uno de los programas más utilizados y que aumenta enormemente las posibilidades de programación es el NQC [1], [2] de Dave Baum, utilizado para programar al RCX en un lenguaje similar al C. Para el NXT, existe un programa llamado RoboC [20], que es mucho más completo que el NQC del RCX, y que además incluye un firmware propio,

que lo hace muy potente. A continuación se enumeran las características más importantes de las principales herramientas de programación.

2.1.1- Lejos Java for Legomindstorms

- Es un nuevo firmware alternativo para el NXT.
- Permite realizar programas en JAVA para controlar robots.
- Funciona en entorno windows y Linux y permite comunicarse con el NXT vía USB.
- En la primera versión todavía no hay soporte para comunicación por Bluetooth, manejo de sensores que se comuniquen por I2C (como el sensor de ultrasonidos) o el soporte de sonido.
- La versión Alpha 0.2 contiene soporte preliminar para bluetooth, soporte para sensor de ultrasonido (I2C) y soporte para el sonido.
- Lejos es el API de JAVA para programar el RCX.
- Su última actualización fue Lejos 3.0
- El nuevo API de JAVA para NXT, se llama iCommand, del cual se acaba de publicar la versión 0.5 Que incluye entre otras cosas Soporte para Webcam y Soporte para brújula electrónica

2.1.2- LabView (entrono grafico de programación)

- Entorno nativo para programar al NXT. Viene junto con el kit de NXT (figura 2, der).
- Utiliza un lenguaje icónico de programación, muy intuitivo.
- Se baja el programa de la PC al NXT por medio del USB o bluetooth.
- Se pueden crear bloques propios
- Comunicación con Bluetooth: el firmware del NXT permite una configuración de tipo Amo-Esclavo para la comunicación con bluetooth.Con bluetooth se pueden comunicar hasta 3 NXT.

2.1.3- RobotC

- Es similar al NQC para los RCX, pero mucho más poderoso.
- Permite programar a los robots en un C reducido
- Incluye un firmware propio.
- Por el momento está disponible la versión beta que es gratuita por 30 días.
- Tiene soporte para la comunicación con bluetooth
- Este es uno de los mejores programas que existen actualmente para desarrollar con LegoMindstorms NXT.

2.2 Juguetes autónomos

Aunque se venden como juguetes, ofrecen unas prestaciones tan avanzadas que hacen de estos un medio excelente para experimentar con la robótica. Así el Robosapien [23], [30] cuenta con siete motores que le proporcionan un número sorprendente de grados de libertad, además de sensores de contacto en los dedos y los pies, y sensor de sonido. Niveles múltiples de interacción ambiental, con humanos y objetos, vista, sonido y sensores de tacto. Movimientos humanoides, que incluyen agacharse, sentarse y pararse, acostarse, levantarse, artes marciales, trucos, etc.

2.2.1- Bípedos

Estos robots, poseen sensores de sonido estéreo, visión infrarroja, y también sensores de tacto para detectar obstáculos y un importante número de grados de libertad. Dentro de esta categoría encontramos:

- Robosapien V1 es una versión con menores prestaciones a nivel de sensores que el V2, no incorpora visión, las capacidades de desplazamiento son similares en sus funcionalidades, si bien la V1 al ser mas pequeña tiene un mejor desplazamiento.
- Robosapien V2, además de las características antes mencionadas incluye también sensores táctiles en los guantes, y en las palmas de las manos, (Figura 3, izq) lo que le permite tomar objetos con ellas. Además posee una cámara que le permite reconocer colores [23].
- Robosapien Multimedia, [23], [30] aumenta aún más las capacidades de RSV2, incluyendo como característica más importante una memoria mini SD, en donde se lo puede programar directamente, a través de un editor de código de tipo gráfico, que es propio de esta versión. Por default tiene cuatro personalidades que el usuario puede modificar. Además puede grabar videos, sonidos en formato mp3, tomar fotos, y luego reproducirlos en su display LCD

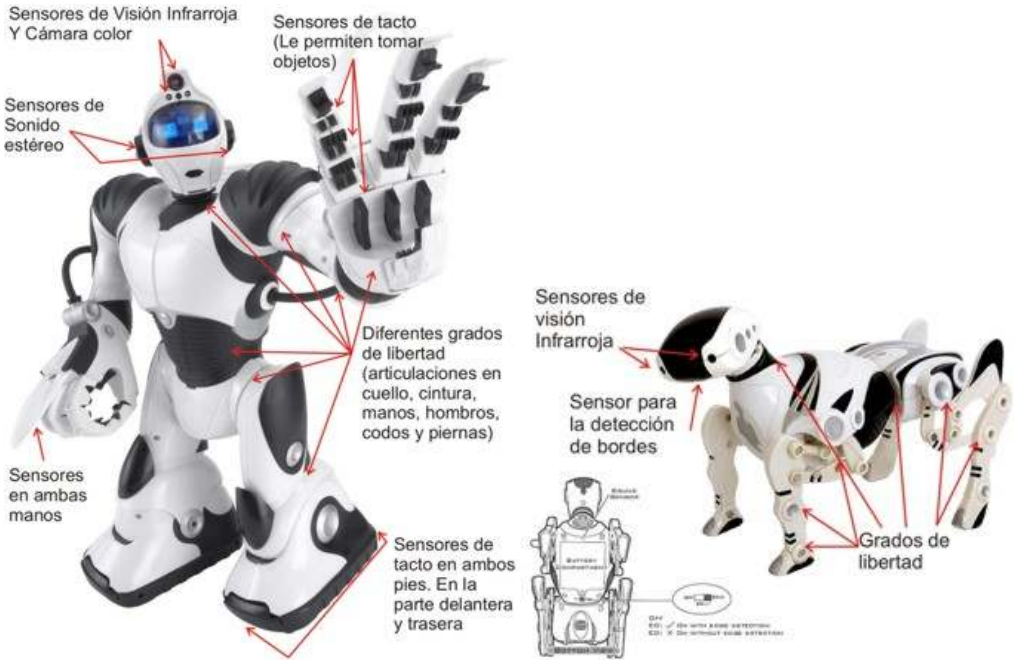

Figura 3: Robosapien V2 (izq), Robopet (der)

2.2.2- Cuadrúpedos

Estos robots, también están equipados con visión infrarroja, sensores de sonido estéreo y motores Entre estos se destacan:

- Robopet, (Figura 3, der) además de lo dicho anteriormente es capaz de interactuar con Robosapien, y también puede detectar los bordes de por ejemplo una mesa [23].
- Robotail, (Figura 4, en verde) posee un sensor de tacto en su lomo, que cuando es presionado, hace que tenga comportamientos diferentes. Además cuando está "hambriento", se vuelve muy "agresivo", y solo se calma "al encontrar comida" [23].
- Roboraptor, (Figura 4) es el roboreptil [23] que también es capaz de interactuar con Robosapien.

Figura 4: Roboraptor y Robotail

2.2.3- Entornos de Programación " GoRobo".

Es un entorno de programación [11], que permite controlar a todos los robots mencionados de la familia WowWee [31]. (Roboraptor, Robopet, Roboreptile, RSV2 y RS Multimedia).

El lenguaje de programación que utiliza se denomina GRIDscript (Go-Robo ID script) [12]. Utiliza una sintaxis de programación simple y consistente basado en las prácticas modernas de productos de programación comerciales (Visual Basic, C++, etc). GRIDscript [12] utiliza sintaxis básica de programación (While/EndWhile, For, If/Else/Endif, Repeat/EndRepeat), la creación de procedimien -tos y el uso de variables.

El alumno puede utilizar este lenguaje para definir procedimientos simples que, posteriormente, pueden combinarse para crear otros más complejos. Además, los robots pueden programarse para interactuar entre sí, ya que el software permite el control de seis de ellos de forma simultánea. Las órdenes se transmiten a través de una torre infrarroja que a modo de intérprete envía los comandos a cada robot, distinguido por su tipo. La versión actual de GoRobo [12], no soporta la torre infrarroja de LegoMindstorms. Solo soporta USB-UIRT [26]. y RedRat3 [21]. A lo largo de este año, sus desarrolladores planean extender el rango de los dispositivos soportados, para incluir a los IR. Dado que por el momento RSV2, no da ningún feedback a GRIDscript, hay una limitación en cuanto a lo que se puede hace con los sensores. Algunas de las cosas que se pueden hacer con GoRobo son: utilizar bloques condicionales y de repetición de instrucciones, utilizar eventos condicionados por timers, se puede introducir ejecución aleatoria de código.Este lenguaje, se ha diseñado para adaptarse a cualquier edad y utilizarse tanto en un contexto educativo como en uno más profesional, donde se genera un ambiente de interacción de los lenguajes clásicos y formales de programación con los comandos nativos de los robots utilizados.

2.2.4- Otras opciones de Programación

Tanto Robosapien como Robopet, reciben los comandos vía IR por control remoto, de este modo hay quienes han realizado un mapeo de dichos comandos a hexadecimal [11], esto hace posible la programación de Robosapien descargando el código con la torre de IR de Lego Mindstorms. El problema es que hay una limitante con la cantidad de instrucciones que el RS puede recibir, que son veinte como máximo. Otra opción más radical: El transplante de cerebro al Robosapien, hay quienes han optado por reemplazar la cabeza del Robosapien V1 por una Palm [23], de este modo, se elimina el problema de la cantidad de instrucciones que se pueden enviar al RS, ampliando enormemente la capacidad de cómputo para Robosapien V1.

3. DESARROLLO AVANZADO PARA AGENTES ROBOTS

La programación de robots presenta interesantes desafíos especialmente en el desarrollo de aplicaciones que actúan en ambientes dinámicos, cooperativos, la elaboración de estrategias para alcanzar sus metas, con la oposición de contrincantes, como así también la capacidad de reaccionar ante situaciones no consideradas en la programación de su control, sin actuación de supervisión

exterior. Su programa de control no debe definir de modo explícito todas las posibles acciones ante todas las posibles situaciones que se pueden presentar en su entorno.
El robot no debe ser totalmente preprogramado [5], [6], [7], [8], [9] debe poseer una arquitectura cognitiva que permita una relación entre sus entradas sensoriales y sus acciones sobre el ambiente, capacidad de generar su mapa autónomo de sensorizaciones - acciones para sobrevivir y alcanzar sus objetivos. El robot debería en su tiempo de vida realizar cada vez mejor sus tareas, aprendiendo por ejemplo de sus errores. El ambiente dinámico, facilita la evolución en función de las distintas sensorizaciones de cada robot, sus metas y la interacción con agentes humanos. Se presenta un escenario favorable para el desarrollo de aplicaciones centradas en contexto donde la participación de actores robots y humanos resulte de interés en un entorno cooperativo
El trabajo de programación del robot además debe desarrollar la interacción con humanos y sistemas de información apoyando la participación de mascotas robots integradas con las tecnologías de información para facilitar la comunicación con humanos en ambientes dinámicos y distribuidos.
La estrategia inicial de comunicaciones para apoyar la interacción entre los agentes autónomos y humanos se basa en el empleo de las facilidades de comunicación inalámbrica Bluetooth entre los agentes robots NXT y dispositivos manuales (celulares, palm, etc). En relación a la familia Robosapien esta se integrará a través de IR en la primera etapa con comunicaciones desde el procesador local de ambiente, el que actuará de gateway entre el mundo de comunicaciones IR y el de Comunicación Bluetooth para el escenario local, este gateway en principio facilitará la interacción de agentes en distintos escenarios distribuidos.
Se busca permitir la interacción de los agentes: NXT, como agente clásico, Robosapien V2 como bípedo y robopet como cuadrúpedo, en un escenario conformado por un Ambiente Dinámico, que en este caso será fútbol de robots, donde los Agentes Robots Autónomos interactúen entre sí a fin de facilitar el desarrollo e investigación en robótica autónoma.
Estos agentes se han seleccionado porque además de tener un costo relativamente bajo con respecto a otros robots, poseen cualidades que permiten explotar sus capacidades al máximo, fomentando el estímulo del aprendizaje en los estudiantes. Los jóvenes se ven atraídos de inmediato por estos robots y las cosas que son capaces de hacer. En su deseo de que se comporten de determinadas maneras, tal vez como hasta el momento pensaban que era posible solo en las películas de ciencia ficción, los hace interesar en temas como la programación, los conceptos de agentes y multiagentes, visión artificial, adquisición de datos, etc. De este modo, se puede aprovechar el interés de los jóvenes en estos robots, para utilizarlo como el motor impulsor, que genera la necesidad de aprender nuevos conceptos que los ayuden a programar sus robots. Por ejemplo, en el caso de los robots de la familia WowWee [23], [30], se tiene un entorno de programación sencillo, que permite introducir al estudiante en los conceptos básicos de la programación, como estructuras de control, bloques condicionales, eventos temporales, etc. Con GoRobo [12], el estudiante puede programar una secuencia de acciones, para el Robosapien V2, para el Robopet o Robotail, y controlarlos en forma simultánea a través de una torre USB-UIRT, en este caso. En este nivel de programación, no se actúa directamente sobre los sensores y motores de los robots, sino que se envían instrucciones que engloban a todas estas acciones de forma transparente para el usuario. Sin embargo, los estudiantes más avanzados, pueden programar al robot NXT de Legomindstoms [14]. En este caso, para su programación, se actúa directamente sobre los sensores y los motores. Además con el NXT, el estudiante puede dar vuelo a su imaginación y modificar el diseño de su robot cuantas veces quiera. De este modo puede construir desde un bípedo, hasta una mano robótica, un robot motorizado, etc.
En cuanto a los entornos de programación [5], estos pueden ser gráficos del tipo de LABView, este ultimo específicamente fue aplicado por la NASA, para la adquisición de datos y control de robots como Sojourner Rover en la misión de exploración de la superficie marciana [17]. o de código

como RobotC [20], iCommand [15], etc. En ambos casos y como se dijo anteriormente, la programación debe actuar directamente sobre los motores, en base a los valores medidos por los sensores, etc. En este nivel, el estudiante adquiere conocimientos de programación de más bajo nivel que en el anterior, y los códigos resultantes con mucho más complejos que en el caso de los robots de la familia WowWee, pero al mismo tiempo, las posibilidades de lo que puede hacer crecen enormemente. Por otra parte, se pretende brindar información global del ambiente a través de la integración de un sistema de visión, que permita la detección y localización de objetos y agentes en el escenario, en este caso el nivel de complejidad se aumenta aún más, posibilitando, que el robot, además de procesar la información de sus propios sensores, tenga información de todo lo que pasa a su alrededor. De este modo el objetivo final del proyecto es el de facilitar a los agentes autónomos (robots y humanos), interactuar entre sí en un escenario bajo un ambiente dinámico, basándose en la información global del ambiente y la propia información del agente obtenida de sus sensores. Es en este nivel, donde se logra la mayor complejidad, y donde se abarca la mayor cantidad de conceptos. A esta altura, el estudiante debe estar familiarizado con los conceptos de agentes y multiagentes, que tienen sus propias metas, objetivos y creencias, que son capaces de interactuar entre sí, e inclusive cooperar para lograr un objetivo común, (en el caso del fútbol, anotar la mayor cantidad de goles, y evitar que el equipo contrario haga goles). También se familiarizará con los conceptos de inteligencia artificial, que posibiliten que los robots, tengan un comportamiento autónomo que varíe en función de la situación, y que no esté preprogramado, sino que se adapte dinámicamente en función de lo que es más conveniente. Y por último, deberá familiarizarse con los conceptos de visión artificial y procesamiento distribuido, que son necesarios para que cada robot, tenga un apoyo de información global, para mejorar la toma de decisiones.

4. NUESTROS AGENTES ROBOTS

Hasta el momento en el Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robotica (ISIER), se cuenta con los siguientes robots: tres RCX, un NXT, un Robosapien V2, un Robosapien V1,un Robopet y in Robotail También los siguientes softwares: GoRobo con licencia Home Edition, RobotC, en su versión beta, iCommand en su versión beta. Hasta el momento se ha conseguido controlar simultáneamente a los tres robots de la familia WowWee (al RSV2, Robopet y Robotail), con el programa GoRobo, utilizando para ello, una torre USB-UIRT de infrarrojos, y también se ha hecho interactuar al RSV2 con el Robopet (Figura 6, izq).

Figura 6: Interacción de RSV2, Robopet y Robotail (izq), NXT siguiendo la línea negra (der)

En cuanto al NXT, se lo programó con LabView, el entorno nativo, con programas simples como por ejemplo el de seguir la línea negra (Figura 6, der). También se lo programó con RobotC, previa bajada del firmware de dicho software. En este caso se simuló el comportamiento de un animal

herbívoro, deambulando por su hábitat, que en este caso fue una alfombra delimitada por una estructura de madera, con papeles verdes distribuidos aleatoriamente, que representaban comida. Así este herbívoro, deambulaba tranquilamente, hasta que encontraba un sector de comida. En este punto se detenía a "comer", además era capaz de detectar los bordes del hábitat, gracias a su sensor de tacto y esquivarlos. En el momento que detectaba el sonido de un predador (con el sensor de sonido), que en este caso fue el Robotail, o que algo se posicionaba por detrás (cosa que detectaba con el sensor de ultrasonido), se "asustaba" dejando de comer y comenzando a escapar a velocidad. En este caso, se pueden observar claramente dos tipos de comportamiento, uno de búsqueda de comida, y otro de escape, que dependen de la interacción del NXT, con su entorno y con robotail (Figura 7).

Figura 7: NXT interactuando con el ambiente y Robotail

5. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Se presentaron las características mas significativas de distintos robots y juguetes autónomos de bajo costo, acompañados de sus lenguajes de programación que facilitan el desarrollo de experiencia en el área de la robótica y contribuyen a la enseñanza de programación de alumnos principiantes, como así también facilitan por parte de alumnos avanzados el desarrollo de agentes autónomos, orientando al desarrollo en este caso a una amplia variedad de modelos computacionales descentralizados, como por ejemplo, las redes neuronales, arquitecturas de subsunción y autómatas celulares, entre otros. Finalmente se presenta la base inicial para una arquitectura que facilite la interacción de agentes en ambientes dinámicos, en este contexto las líneas futuras de investigación se orientan al desarrollo de framework de integración de los distintos robots, el desarrollo de capacidades de simulación interoperable entre mundos virtuales y reales a fin de enriquecer el escenario de aprendizaje de los robots, el procesamiento distribuido del ambiente entre agentes autónomos y la integración de agentes robots autónomos en hábitat inteligentes.

Agradecimientos: Este trabajo ha sido soportado en parte por el proyecto A01-007/06 Secretaria de Ciencia y Tecnología de UM y la FICCTE.

6. BIBLIOGRAFÍA

[1]. BAUM, Dave. *NQC Manual.* [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://bricxcc.sourceforge.net/nqc/doc/NQC_Manual.pdf>

[2]. BAUM, Dave; HANSEN, John. *NQC Programmer's Guide*. [en línea]. Versión 3.1 r5. [ref. 2 de Agosto de 2006]. Disponible en Web: < http://bricxcc.sourceforge.net/nqc/doc/NQC_Guide.pdf>

[3]. BEARD, R. *"Ball prediction for Robot Soccer*". Department of Electrical & Computer Engineering, Brigham Young University, Provo. 2003

[4]. BILL, N; SCHILIT, Norman Adams; et. al. "Context-Aware Computing Applications, *IEEE Workshop on Mobile Computing Systems and Application*, December 1994.

[5]. BOER, R; KOK, J. "*The Incremental Development of a Synthetic Multi-Agent System:The UvA Trilearn 2001 Robotic Soccer Simulation Team*". Faculty of Science, University of Ámsterdam. 2002

[6]. BORENSTEIN, J; EVERETT, H. R; FENG, L. *Navigating Movile Robots: Systems and Techniques*. 1996

[7]. BRATMAN, Michael E. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, Mass., 1987.

[8]. CASTELO, C; FARSI, H; SCARPETTINI, F. Tesis de Licenciatura "*Fútbol de Robots: Revisión del Estado del Arte y Desarrollo del Equipo UBASot de Simulación*". Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. 2002

[9]. CLARO, M.; GAZOLLI, A; POTENZA, A; VISCUSO, Germán; IERACHE, Jorge. "*El equipo Morasot*". Facultad de Informática, Universidad de Morón. 2003

[10]. CompuBlog, El blog del aula de robótica Compubot. EducaMadrid [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web:<http://complubot.educa.madrid.org/blog/>

[11] Controlling RoboSapien using LEGO IR-Tower. Trondheim-Bratislava, July 2005 - January 2006. Diponible en Web: <http://www.robotika.sk/maine.php>

[12]. GOROBO, [en línea]. , [ref. 20 de Abril de 2007]. Disponible en Web: <http://www.q4tecnologies.com/>

[13]. GUESTRIN, C; VENKATARAMAN, S; KOLLER, D. *Context Specifffic Multiagent Coordination and Planning with Factored MDPs*. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, Canada, July 2002.

[14]. LEGO.com. MINDSTORMS NXT Home. [en línea]. [ref. 1 de Abril de 2007]. Disponible en Web: <http://mindstorms.lego.com/default.aspx>

[15]. LeJOS, Java for Legomindstorms. SourceForge [en línea]. [ref. 10 de Abril de 2007]. Disponible en Web: <http://lejos.sourceforge.net/>

[16] MOROLLÓN SÁNCHEZ, María; SEGOVIANO, Alonso. *1, 2, 3... Logo (Ideas e Imaginación)*. Centro de Orientación de Sociología y Psicología Aplicada. Madrid, España: Cospa, 1985.

[17]. Nacional Instruments. *LabVIEW*. [en línea]. [ref. 1 de Abril de 2007]. Disponible en Web: <http://www.ni.com/academic/mindstorms/>

[16]. NQC – Not Quite C. [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://bricxcc.sourceforge.net/nqc/index.html>

[19]. Papert, S. Situating constructionism, en I. Harel y S. Papert (comps.),Constructionism.Norwood,NJ,Abel Publishing. 1991.

[20]. QUICK START GUIDE, [en línea]. Robotics Academy, Carnegie Mellon University, [ref. 5 de Abril de 2007]. Disponible en Web: < http://www.robotc.net/>

[21]. RedRat3 – USB Universal Remote Control for PC. Disponible en Web: <http://www.redrat.co.uk/RedRat3/index.html>

[22]. Resnick Mitchel Tortugas, Termitas y Atascos de Tráfico, Gedisa, 2001

[23]. ROBOSAPIEN.tk, the first unofficial robosapien hacks and mods site. [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://home.planet.nl/~pruim006/main.htm>

[24]. ROGER, Chris. *LEGOS, ROBOLAB, and LabVIEW: Designing, Programming, and Collecting Data*. University of Wisconsin-Madison.The Institute on Learning Technology. National Institute of Science Education. EEUU. 2001. Disponible en Web: <http://www.wcer.wisc.edu/archive/cl1/ilt/extra/download/solution/rogerscw97.doc>

[25]. RUSSELL, Stuart; NORVIG, Peter. *Inteligencia artificial. Un enfoque moderno*. GUTIÉRREZ, Raúl Bautista (trad.); RANGEL, Raymundo Hugo (rev. Tec.). 1a ed. México: Prentice Hall Hispanoamericana, 1996. 979p. ISBN: 968-880-682-X

[26]. USB-UIRT Home Page. Disponible en Web: <http://www.usbuirt.com/>

[27]. VIEGAS D'ABREU, João Vilhete; CHELLA, Marco Túlio. Superlogo-RCX Ambiente para Robótica Educacional. Disponible en Web: <http://www.nied.unicamp.br/~siros/doc/pedagogia_2005_cuba.pdf>

[28]. WOOLDRIGE, Michael; JENNINGS, Nick. R. *Agent Theories, Architectures and Languages: a Survey in Eds. Intelligence Agents*. Berlin: Springer-Verlag, 1995. Vol 1, Nro 22

[29]. WOOLDRIGE, Michael. *An introduction to Multiagent Systems*. John Wiley & Sons. 2002.

[30]. WowWee Robotics. RS Multimedia. [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web:< http://www.rsmediaonline.com/>

[31]. WowWee Robotics.. [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web: <http://www.woowee.com>