

## Laboratorio de Robots Autónomos en el Contexto de las Carreras de Informática Experiencias para su Implementación

J. Ierache <sup>(1,2)</sup> M Dittler<sup>(1,2)</sup>, H Padovani <sup>(2)</sup>

<sup>(1)</sup>Instituto de Sistemas Inteligentes y Enseñanza experimental de la Robótica (ISIER)

<sup>(2)</sup>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

<sup>(1,2)</sup> {jierache, pdittler, hpadovani } @unimoron.edu.ar

### Resumen

*Este trabajo presenta nuestras experiencias para la implementación de un laboratorio de sistemas autónomo de robots aplicado a la formación de estudiantes iniciales y avanzados en carreras de informática, considerando las tecnologías específicas, robots, juguetes autónomos, lenguajes y ambientes de programación, como así también la construcción de un robot bípedo de programación abierta a partir de un juguete. Estos robots por su bajo costo y la documentación disponible facilitan un amplio acceso a distintas Universidades e Institutos. En este contexto, la enseñanza de programación aplicada a través de sistemas autónomos de robots facilita la interacción entre sistemas autónomos, el ambiente de actuación, dispositivos de comunicación celulares, PDA, entre otros dispositivos. Presentan una oportunidad estratégica para la formación de recursos humanos cubriendo diversos aspectos desde la elaboración de la especificación, modelado, desarrollo del software e implementación y pruebas de un sistema autónomo de robots*

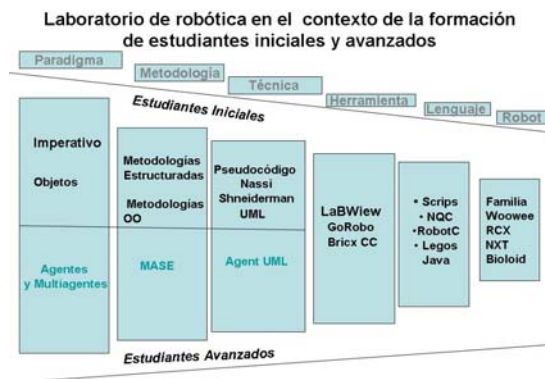
**Palabras claves:** *Robótica, Tecnologías en Educación, Programación, Sistemas autónomos, laboratorio robótica*

### 1. Introducción

El desarrollo de las tecnologías aplicadas a la educación contribuye en el proceso de aprendizaje, en particular la aplicación de un Laboratorio de Sistemas Autónomos de Robots (LSAR) colabora con distintas áreas en los procesos de formación de alumnos de la disciplina informática; desde la interpretación de un requerimiento, hasta la implementación del sistema autónomo, facilitando la creatividad del alumno en materia de la construcción física hasta la optimización de su software y la integración de sensores, como así también el desarrollo de ambientes cooperativos y competitivos entre robots. Para un alumno observar cómo una tortuga se desplaza 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, cómo un Sistema Autónomo Robot (SAR), puede esquivar obstáculos hasta llegar a su meta en el rincón de una habitación, e interactuar con nosotros a través de nuestro celular. Consideramos que las TICs, en particular las asociadas a los laboratorios de robots autónomos brindan un apoyo en los procesos de aprendizaje de estudiantes iniciales y avanzados, en el marco del contexto propuesto en figura 1, se contempla: [a] **Paradigma** bajo el cual el alumno desarrolla su trabajo, [b] **Metodología** que se aplica bajo el paradigma seleccionado, [c] **Técnicas** que facilitan el desarrollo de fases y etapas de la

metodología aplicada, [d] **Herramientas** sobre las cuales se aplican las técnicas, [e] **Lenguajes de programación**, [f] **Robots**. En este orden podemos considerar para estudiantes iniciales el paradigma imperativo, el de objetos, sus metodologías, técnicas como diagramas de Nassi Seiderman [x], UML [5], lenguajes de programación como C en particular NQC [24], y JAVA, en particular LeJOS Java [21], para el desarrollo del software que correrá sobre robots RCX [3], NXT [2]. Para estudiantes avanzados consideramos el paradigma multiagente, con metodologías como MaSE [4], técnicas como Agent-UML [1].



**Figura 1 Contexto de la formación de estudiantes**

En otro orden el LSAR facilita la explicitación de requerimientos formulados a partir de estándares IEEE 830 [6], la validación de éstos y la realización de pruebas con la aplicación de técnicas como complejidad ciclomática [7]. Contribuye con el mejoramiento de procesos constructivos en un ambiente de trabajo en equipos de alumnos altamente motivados por el desarrollo de sus robots o mascotas. Podemos destacar que existe una amplia diversidad de proyectos, desde la utilización del clásico logo en la Universidad Estatal de Campinas (UNICAMP); Brasil, con fines educacionales [33], donde 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 enseñanza con distintos niveles de complejidad [30]. 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 [14], [15], [28]. 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 adelante 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 adelante otra vez, sigue así oscilando en el borde de la sombra; en este caso podemos considerar a nuestro agente robot como una criatura que detecta bordes, en sí esta capacidad no está 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 las ven 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 qué nivel es el mejor, piensan en sistemas en término de niveles [22]. La idea de aprender mediante el diseño es un aspecto de lo que Seymour Papert [25] denominó como enfoque constructor del aprendizaje y la educación. Los seres humanos construyen su conocimiento con particular eficacia cuando participan en la construcción de productos que les son afectivos.

## 2. Robots de Lego

El objetivo de esta sección es brindar una orientación general de los Kits de Robótica económicos actuales de Lego RCX [3] y el reciente NXT [2] sus herramientas de programación en el ambiente de LabView [30], RobotC [26], NQC [24], LeJOS [21], entre otras.

### 2.1. Legomindstorms RCX

El RCX (Figura 2.a) se caracteriza por contar con : tres puertos para motores, cinco slots para guardar programas, un sensor de luz, que permite distinguir diferentes niveles de luminosidad y oscuridad, dos sensores de tacto, que pueden detectar tres estados (presionado, libre, rebote), cuenta además con un parlante para la emisión de sonidos. 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 su port frontal infrarrojo.

### 2.2 Legomindstorms NXT

El NXT (Figura 2.b) es la nueva generación de robots de Legomindstorms [3], se caracteriza por una mayor capacidad de cómputo que el RCX. El NXT incluye

funciones para testear los sensores, la personalización de los sonidos que puede reproducir, tiene tres puertos para motores, cuatro puertos para sensores.

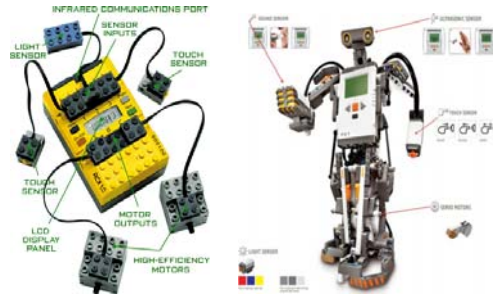


Figura 2.a RCX

Figura 2.b NXT

Se equipa con un sensor de luz, un sensor de sonido para detectar sonidos, dos sensores de tacto, que pueden detectar tres estados (presionado, libre, rebote), un sensor de ultrasonido que actúa como un radar permitiendo la detección de objetos, al que se lo puede configurar para detectar objetos lejanos o cercanos, detecta objetos entre 0 y 255 cm con una precisión de  $\pm 3$  cm, además cuenta con un parlante de alta fidelidad, mejorado y tres servomotores, mejorados con respecto a la versión RCX. Los servomotores tienen integrados sensores de rotación que permiten movimientos precisos, controlados y una perfecta sincronización de los motores. El NXT posee un puerto USB, para la descarga de programas. Soporta comunicación bluetooth inalámbrica, permitiendo descargar programas de este modo, así como también la interacción con celulares, con PCs y laptops, etc. La comunicación con otros NXT también es vía bluetooth. Estos robots pueden ser programables en entornos gráficos nativos, para RCX [3] y LabView [23] para NXT. 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 [54]. 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 entornos 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 [21], [47] y para NXT el LeJOS NXJ [46], iCommand un paquete java para el control de comunicaciones sobre conexiones Bluetooth [45]. Uno de los programas más utilizados y que aumenta enormemente las posibilidades de programación es el NQC [8], [9] de Dave Baum, utilizado para programar al RCX en un lenguaje similar al C. Para el NXT, existe un programa llamado RobotC [26], 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 de los legos mindstorms.

### 2.3 Lenguajes y herramientas

Entre los lenguajes se caracterizan: NQC aplicado para RCX y RobotC similar al NQC para los RCX, pero mucho más poderoso, permite programar a los robots en un C reducido. Incluye un firmware propio. Tiene soporte para la comunicación con bluetooth. Este es uno de los nuevos programas que existen actualmente para desarrollar con LegoMindstorms NXT. Para la programación en JAVA de NXT se encuentra el Lejos Java. Su firmware alternativo para el NXT, se caracteriza por: [a] permitir realizar programas en JAVA para controlar robots NXT, [b]funcionar en entorno Windows y Linux, [c] comunicarse con el NXT vía USB, [d] soportar 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. 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

## 3 Juguetes Autónomos

En materia de Juguetes autónomos se detallarán las características más relevantes de “Robosapien”. (robot bípedo) y de “Robopet, Robotail, Roboraptor” (robots cuadrúpedos), Roboquad (robots exopodos) de la familia Woowee [37] . Se consideran además las interfases de comunicación y herramientas de programación, en particular GoRobo. Aunque se venden como juguetes, ofrecen unas prestaciones tan avanzadas que hacen de éstos un medio excelente para experimentar con la robótica

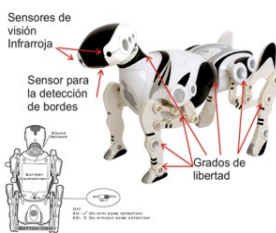
**3.1 Bípedos:** estos robots, poseen sensores de sonido estéreo, visión infrarroja, y también sensores de tacto para detectar obstáculos y un importante número de grados de libertad. Dentro de esta categoría encontramos: [a] Robosapien V1 es una versión con menores prestaciones a nivel de sensores que el V2, no incorpora visión, las capacidades de desplazamiento son similares en sus funcionalidades, si bien el RSV1 al ser más pequeño tiene un mejor desplazamiento, [b] Robosapien V2, además de las características antes mencionadas incluye también sensores táctiles en los guantes, y en las palmas de las manos, (Figura 3) lo que le permite tomar objetos con ellas. Además posee una cámara que le permite reconocer colores [29]. [c] Robosapien Multimedia, [36], 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

**3.2 Cuadrúpedos.** Estos robots, también están equipados con visión infrarroja, sensores de sonido estéreo y motores. Entre éstos se destacan: [a] Robopet, (Figura 4.) además de lo dicho anteriormente es capaz de interactuar con Robosapien, y también puede detectar los bordes de por ejemplo una mesa [51]. [b] Robotail, (Figura 5.a.) 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 sólo se calma “al encontrar comida” [52]. [c] Roboraptor, (Figura 5.b.) es el roboreptil [53] que también es capaz de interactuar con Robosapien



**Figura 4.** Robopet



**Figura 5.a** Robotail



**Figura 5.b.** Roboraptor



**Figura 5.c** Roboquad

**3.3 Artrópodos:** En este grupo presentamos a Roboquad [41], Figura 5.c, tiene cuatro patas con un chasis que está diseñado para moverse en cualquier dirección con 3 velocidades diferentes se caracteriza por identificar movimientos a una distancia de alrededor de tres metros, una vez identificado puede seguir el movimiento del objeto. Tiene sensores de bordes capaces de detectar marcos de puertas, bordes de mesas y sillas

### 3.5 Lenguajes y herramientas de programación familia WowWee

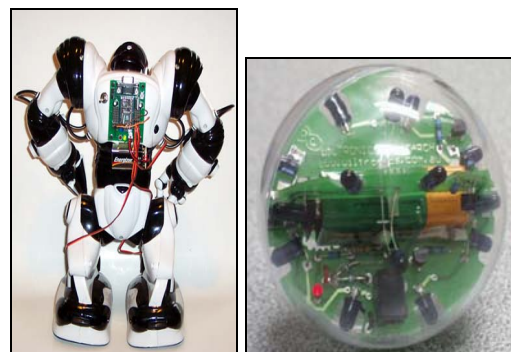
Es un entorno de programación “GoRobo” [18], permite controlar a la mayoría de los robots mencionados de la familia WowWee [37]. (Roboraptor, Robopet, Roboreptile, RSV2 y RS Multimedia). El lenguaje de programación que utiliza se denomina GRIDscript (Go-Robo ID script) [18]. Utiliza una sintaxis de programación simple y consistente basado en las prácticas modernas de productos de programación comerciales (Visual Basic, C++, etc). GRIDscript [18] utiliza sintaxis básica de programación (While/EndWhile, For, If/Else/Endif, Repeat/EndRepeat), para la creación de procedimientos y el uso de variables. El estudiante inicial 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 a través de un torre infrarroja, como USB-UIRT [32]. y RedRat3 [27]. 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. Incorpora además un editor de escenas al que se le puede sumar sonido. Otras opciones de Programación, para el 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 [17], 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 es el transplante de cerebro al Robosapien, hay quienes han optado por reemplazar la cabeza del Robosapien V1 por una Palm [57], 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.

#### **4. Construcción de Sistemas autónomos de programación abierta a partir de juguetes.**

Robosapien es un robot humanoide con funciones de comportamiento pre-programadas que pueden ser ejecutadas desde un control remoto inalámbrico infrarrojo. El Robosapien [29], [37] cuenta con siete motores que le proporcionan diferentes grados de libertad, en las piernas, brazos, manos, cadera y cabeza, además de sensores de contacto en los dedos y los pies, y sensor de sonido. El humanoide incluye 67 funciones simples preprogramadas en la memoria interna. Una de las ventajas más importante es el diseño de la parte motriz del robot, lo que brinda un desplazamiento muy veloz y sincronizado. Por otro lado, no

deja de ser un juguete, por lo que no cuenta con una interfaz de comunicación ni almacenamiento interno. Debido a eso se nos ha planteado la necesidad de agregarle un “cerebro extra” a través de un microcontrolador acoplado a las funciones motrices del robot. (figura 6 ).Estas modificaciones llevan a poder programar el microcontrolador a través de una interfaz de desarrollo muy sencillas y con funciones compatibles con MS Visual Basic. El microcontrolador empleado es el BX-24p . [55] a un microcontrolador programable en Basic. El Basic X24 reúne en un circuito del tamaño de un chip de 24 patas a un microcontrolador Atmel, una memoria EEPROM de 32Kbytes, un regulador de tensión y un puerto RS232, de forma que basta con escribir los programas en Basic en el PC y luego volcarlos al módulo para que éste los ejecute de forma totalmente autónoma. El lenguaje de programación, BasicX [39] cuenta con todas las estructuras de programación conocidas (While Do, If them, Case, For) y con todos los tipos de datos que son necesarios para manipular los datos de entrada desde los sensores y los datos de salida para los servo-motores. La combinación de la versatilidad del Robosapien V1 [29], [36], [37] y la potencia del microcontrolador BX-24 [39] pueden llevar al Estudiante a programar el robot y hacerlo trabajar de forma completamente autónoma en ambientes emergentes.



**Figura 6. V1BX-24 Figura 7 Roboball IR**

El estudiante trabaja en un editor básico para el entorno de programación BX-24

(BasicX). Sumándole complejidad se pueden agregar sensores de lectura Infrarrojo (IS) , por lo que el robot puede detectar una bola que emite señales de IR [40] (figura 7) y perseguirla tratando de patearla y conduciéndola hacia delante (principio de funcionamiento para que el robot juegue al fútbol. Esto lo logra leyendo los datos de los sensores de IR, la bola (IS) y los sensores de luz (LS) que escanean continuamente el campo de juego (Rojo: campo a defender , Blanco: campo hacia dónde atacar), con eso se le provee al robot de sentido de ubicación respecto al campo de juego. (Figura 8). Así como se implementan sensores de IR , de luz , también se pueden plantear proyectos de desarrollo de análisis de señales de otras características de entradas, por ejemplo sensores de temperatura (donde el robot tratará de mitigar un posible nicho de fuego), análisis de señales de proximidad (sensores de ultrasonido), donde se puede hacer interactuar al robot con otros autómatas para ejecutar en ambientes cooperativos (juego de equipo) con estrategias determinadas para llegar a realizar un determinado objetivo

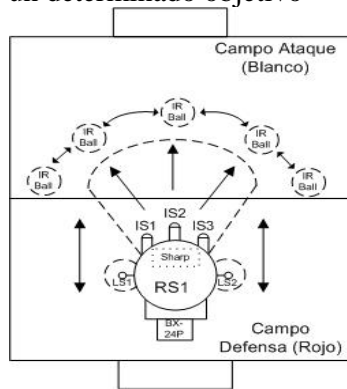


Figura 8. Robosapien con Sensores

## 5 Robots de Vigilancia

Spykee Robot [42], figura 9.a, se caracteriza por ser un espía remoto: se puede tomar el control del robot de cualquier parte del mundo a través de Internet usando Skype Video Call service, el monitor de seguridad hogareña puede

monitorear un hogar con una cámara interna y su sistema de orugas que puede adaptarse a diferentes superficies. Tiene capacidades de VOIP: actúa como un teléfono VOIP por lo que se puede comunicar a través de él con cualquiera que vea a través de su cámara. Entre otras características tiene Auto Carga de baterías: cuando el Spykee Robot se detecta con bajas baterías, éste trata de ubicar su cargador y automáticamente se conecta para recargarse. En otro orden puede operar en Modo guardia: detecta movimientos a través de sus sensores y puede tomar fotos con su cámara interna y la manda a través de e-mail con sonido de alarma si es necesario. Rovio [48] figura 9.b, es un robot que se caracteriza por: [a] tener un ágil desplazamiento, gracias a sus tres ruedas omni-direccionales, [b] incorpora WiFi 802.11b/g , webcam, micrófono y parlante con la cual puede actuar como una plataforma móvil para una conferencia remota o vigilar un área, [c] Incorpora un GPS NorthStar que le permite navegar y localiza objetos con exactitud, [d] cuenta con una herramienta que permite planificar sus actividades en el ambiente de actuación, [e] cuenta con led que le permiten utilizarlo en su navegación durante la noche, [f] tiene la capacidad de regresar a su bahía de alimentación cuando sus baterías están bajas para su recarga, [g] tiene una conexión USB



Figura 9.a Spykee



Figura 9.b Rovio

## 6 Robot Voladores

FlyTech Dragonfly [49], figura 10.a, es una libélula, su vuelo basado en el comportamiento de insectos, es radio controlada a través de la velocidad y

sentido de giro de su rotor de cola y la velocidad de batido de sus alas muy livianas y flexibles impulsada por un micromotor, cuenta con un sistema de recarga rápida de baterías incorporada en el cuerpo de Dragonfly, con unos leds en sus ojos que se iluminan, blanquean, y cambian de color indica el estado de carga de su batería interna. Por otro lado FlyTech Bladestar [50], figura 10.b es un novedoso robot volador de alas rotativas para vuelo interno, cuenta con modo auto piloto con sensores para guiarlo en su vuelo, evitando el techo y volando a través de los obstáculos, otro modo de dirección es con la interacción de la presencia de las manos como guía de vuelo, para un control completo cuenta con un radio de tres canales digitales IR que permiten el control de altura, potencia, adelante y atrás, izquierda y derecha. Su tiempo de vuelo es de aproximadamente cinco minutos. Ambos robots voladores no cuentan a la fecha con interfaces de programación, en las futuras líneas de investigación para el desarrollo de un Framework del LASR se contempla la exploración para su integración



10. a Dragonfly

10.b Bladestar

## 7 Robots Multimodulares

El Kit de Robot Bioloid [43], Figura 11, se caracteriza por disponer de un total de 18 servomotores, sensores de infrarrojos en la cabeza para comunicarse con otros robots y sensores de proximidad hacia delante y hacia los lados, micrófono y altavoz. Bioloid Comprehensive es el kit de la plataforma robótica modular adecuado para construir robots avanzados de hasta 18 grados de libertad como los humanoides. Es adecuado para aprendizaje, hobby, investigación y competición. El kit es como

una versión superior de Meccano y está hecho de muchos bloques constructivos que el usuario puede unir y encajar con tornillos. Su lenguaje de programación es C.



Figura 11 Robot Bioloid

## 8. Laboratorio de Desarrollo de Sistemas Autónomos de Robots

La conformación de un LSAR presenta una oportunidad para el aprendizaje de los alumnos especialmente en el contexto de la programación de robots que actúan en ambientes dinámicos, cooperativos y requieran la elaboración de estrategias para alcanzar sus metas frente a la oposición de contrincantes, sin actuación de supervisión exterior. Los programa de control del SAR no pueden 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 [10], [11], [12], [13], [14], [15] debe poseer una arquitectura cognitiva que permita una relación entre sus entradas sensoriales y sus acciones sobre el ambiente. Debe tener capacidad de generar su mapa autónomo de sensorizaciones-acciones para sobrevivir y alcanzar sus objetivos. Un LSAR presenta también 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 interactivo a través de celulares, Internet, etc. 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. Los



estudiantes avanzados se interesan también en las metodologías en el contexto multiagente, herramientas, Sistemas autónomos Inteligentes, conceptos de Inteligencia Artificial, visión y procesamiento distribuido [57], [19], [34], [35] [31]. Por otra parte, un LSAR puede incorporar 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 y se adapte dinámicamente

## 9. Conclusiones y futuras líneas de investigación

El uso de la tecnología robótica propuesta en el proceso de aprendizaje de los alumnos, facilita el desarrollo de distintas experiencias (fútbol de robots, rescate, navegación, entre otras) que impactan directamente en el proceso aprendizaje, mejorando la motivación de los alumnos, contribuyendo a la enseñanza de programación tanto de alumnos principiantes, como de alumnos avanzados en sus trabajos de finales, los que dirigen sus esfuerzos al desarrollo de Sistemas autónomos de Robots, orientando el desarrollo en este caso a una amplia variedad de modelos computacionales descentralizados, como por ejemplo, las redes neuronales, arquitecturas subsumidas y autómatas celulares, entre otros.

El LSAR presenta un espacio de oportunidades para el desarrollo de software de robots integradas con las tecnologías de información, en ambientes dinámicos y distribuidos. 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 los escenarios de aprendizaje de robots, el procesamiento distribuido del ambiente entre sistemas autónomos y la integración de sistemas robots autónomos de distintos tipos

## 10. Bibliografía

- [1] B. Bauer, J. P. Muller, J. Odell. "Agent UML: A Formalism for Specifying Multiagent Software Systems". Proc. ICSE 2000 Workshop on Agent-Oriented Software Engineering AOSE 2000, Limerick, 2000, Agent-UML
- [2] NXT <http://mindstorms.lego.com/>
- [3] RCX [www.lego.com/eng/education/mindstorms/](http://www.lego.com/eng/education/mindstorms/)
- [4] DeLoach, S.: Analysis and Design using MaSE and agent Tool. Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS). 2001
- [5] UML <http://www.uml.org/>
- [6] Requirements Specification ANSI/IEEE 830
- [7] Mc Cabe, T. "A Software Complexity Measure" IEEE Transactions on Software Engineering, vol 2, n° 4, pp 309-320, 1976
- [8] BAUM, Dave. NQC Manual. [http://bricxcc.sourceforge.net/nqc/doc/NQC\\_Manual](http://bricxcc.sourceforge.net/nqc/doc/NQC_Manual).
- [9] BAUM, Dave; HANSEN, John. NQC [http://bricxcc.sourceforge.net/nqc/doc/NQC\\_Guide](http://bricxcc.sourceforge.net/nqc/doc/NQC_Guide)
- [10] BEARD, R. "Ball prediction for Robot Soccer". Department of Electrical & Computer Engineering, Brigham Young University, Provo.
- [11] BILL, N; SCHILIT, Norman Adams; et. al. "Context-Aware Computing Applications, IEEE Workshop on Mobile Computing Systems and Application, December 1994.
- [11] 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 Amsterdam. 2002
- [12] BORENSTEIN, J; EVERETT, H. R; FENG, L. Navigating Mobile Robots: Systems and Techniques. 1996
- [13] BRATMAN, Michael E. Intention, Plans, and Practical Reason. Harvard University Press, Cambridge, Mass., 1987.
- [14] 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

- [15] CLARO, M.; GAZOLLI, A; POTENZA, A; VISCUSO, Germán; IERACHE, Jorge. “*El equipo Morasot*”. Facultad de Informática, Universidad de Morón. 2003
- [16] CompuBlog, El blog del aula de robótica <<http://complubot.educa.madrid.org/blog/>>
- [17] Controlling RoboSapien using LEGO IR-Tower.Trondheim-Bratislava, <<http://www.robotika.sk/maine.php>>
- [18]GOROBO: <http://www.q4technologies.com/>
- [19] GUESTRIN, C; CENKATARANAN, S; KOLLER, D. *Context Specific Multiagent Coordination and Planning with Factored MDPs*. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, Canada, 2002.
- [20] LEGO.com. MINDSTORMS NXT: <<http://mindstorms.lego.com/default.aspx>>
- [21]LeJOS, Java for Legomindstorms. SourceForge: <<http://lejos.sourceforge.net/>>
- [22] MORROLLON, M; SEGOVIANO, A. 1, 2, 3... *Logo (Ideas e Imaginación)*. Centro de Orientación de Sociología y Psicología Aplicada. Madrid, España: Cospa, 1985.
- [23] National Instruments. *LabVIEW*. [en línea]. [ref. Abril de 2007]. Disponible en Web: <<http://www.ni.com/academic/mindstorms/>>
- [24] NQC – Not Quite C.: <<http://bricxcc.sourceforge.net/nqc/index.html>>
- [25] PAPERT, S. Situating constructionism, en I. Harel y S. Papert (comps.), *Constructionism*. Norwood, NJ, Abel Publishing. 1991.
- [26] QUICK START GUIDE, Robotics Academy, CarnegieMellonUniversity, <<http://www.robotc.net/>>
- [27] RedRat3 – USB Universal Remote Control for <<http://www.redrat.co.uk/RedRat3/index.html>>
- [28] RESNICK, Mitchel. *Tortugas, Termitas y Atascos de Tráfico*, Gedisa, 2001
- [29] ROBOSAPIEN.tk, the first unofficial robosapien hacks and mods site. <<http://home.planet.nl/~pruim006/main.htm>>
- [30] 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. <<http://www.wcer.wisc.edu/archive/c11/ilt/extra/download/solution/rogerscw97.doc>>
- [31] 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.
- [32] USB-UIRT: <<http://www.usbuirt.com/>>
- [33] VIEGAS D’ABREU, João Vilhete; CHELLA, Marco. Superlogo-RCX <[http://www.nied.unicamp.br/~siros/doc/pedagogia\\_2005\\_cuba.pdf](http://www.nied.unicamp.br/~siros/doc/pedagogia_2005_cuba.pdf)>
- [34] WOOLDRIGE, Michael; JENNINGS, Nick. R. *Agent Theories, Architectures and Languages: a Survey in Eds. Intelligence Agents*. Berlin: Springer-Verlag, 1995. Vol 1, Nro 22
- [35] WOOLDRIGE, Michael. *An introduction to Multiagent Systems*. John Wiley & Sons. 2002.
- [36] WowWee <<http://www.rsmmediaonline.com/>>
- [37] <<http://www.woowee.com>>
- [38] Robodyssey <<http://www.robodyssey.com>>
- [39] BasicX Programming language <<http://www.basicx.com>>
- [40] Acroname IR Ball<<http://www.acroname.com/>>
- [41] WowWee Roboquad <<http://www.roboquadonline.com/>>
- [42] Spykee Spy Robot <<http://www.spykeeworld.com/>>
- [43] Tribotix Bioloid Constructive Kid <[http://www.tribotix.com/Products/Robotis/Bioloid/Bioloid\\_info1.htm](http://www.tribotix.com/Products/Robotis/Bioloid/Bioloid_info1.htm)>
- [44] Nassi, I.; Shneiderman, B.: Flowchart techniques for structured programming, SIGPLAN Notices XII, August 1973
- [45] Icommand NXT <[http://lejos.sourceforge.net/p\\_technologies/nxt/icommand/icommand.php](http://lejos.sourceforge.net/p_technologies/nxt/icommand/icommand.php)>
- [46] Lejos NXJ <[http://lejos.sourceforge.net/p\\_technologies/nxt/nxj/nxj.php](http://lejos.sourceforge.net/p_technologies/nxt/nxj/nxj.php)>
- [47] LeJOS RCX <[http://lejos.sourceforge.net/p\\_technologies/rcx/lejos.php](http://lejos.sourceforge.net/p_technologies/rcx/lejos.php)>
- [48] Rovio <<http://www.gizmag.com/wowwee-rovio-wifi-robot-webcam/8606/>>
- [49] <<http://flytechonline.com/about.html>>
- [50] <<http://www.bladestaronline.com/>>
- [51] Robopet: [www.robopetonline.com](http://www.robopetonline.com)
- [52] Robotail: [www.roboreptileonline.com](http://www.roboreptileonline.com)
- [53] Roboraptor : [www.roboraptoronline.com](http://www.roboraptoronline.com)
- [54]NI : <http://www.ni.com/academic/mindstorms/>
- [55] BX-24 <http://www.superrobotica.com/S3In>
- [56] Ierache, J et al. Elaboración de una Aproximación Metodológica para el desarrollo de Software Orientado a Sistemas Multiagentes ,2003 . <<http://www.fi.uba.ar/materias/7570/index.htm>>
- [57] Sven., Behnke et al., Using Handheld Computers to Control Humanoid Robots Proceedings Dextrous Autonomous Robots and Humanoids, 2005.