

# MICROSOFT ROBOTICS DEVELOPER STUDIO UNA NUEVA TECNOLOGÍA DE SOFTWARE PARA EL DISEÑO, SIMULACIÓN E IMPLEMENTACIÓN DE PLATAFORMAS ROBÓTICAS.

## A New Software Technology of the Design, Simulation and Implementation of Robotic Platform Microsoft Robotics Developer Studio

### RESUMEN

Este artículo contiene una introducción a Microsoft Robotics Developer Studio, el cual es una herramienta útil a la hora de realizar diseños robóticos. En este se muestran los resultados obtenidos con base a la construcción de una aplicación robótica manejada mediante un control Genérico USB, enviando las señales de control en tiempo real mediante una computadora comunicada con el Kit Lego Mindstorms en configuración Tribot por medio de un dispositivo Bluetooth.

**PALABRAS CLAVES:** Plataformas Robóticas, Servicios, Simulación e Implementación.

### ABSTRACT

*This article contains Microsoft Robotics Developer Studio an overview; this is a useful tool for making designs of all kinds of robotic Platform. This shows the results obtained based handled by a generic USB driver on the construction of a robotic application, issuing instructions by computer through a Bluetooth device and received by the Lego Mindstorms Kit Tribot Settings.*

**KEYWORDS:** Robotics Platform, Service, simulation and Implementation.

### 1. INTRODUCCIÓN

Como en muchos diseños de la ingeniería, es de suma importancia en la rama de la robótica la realización de simulaciones en ambientes ideales que permitan verificar el funcionamiento real de los prototipos que serán construidos posteriormente, con las cuales se puedan determinar fallos o inconsistencias en el diseño y de esta manera realizar los ajustes pertinentes en nuestro prototipo, permitiendo un aumento en la eficiencia de la tarea específica realizada por el robot para la cual está construido disminuyendo las pérdidas económicas.

En la actualidad existen una gran variedad de robots, clasificados según su finalidad, habilidades y estructura básica, normalmente se dividen en categorías como: móviles o fijos, autónomos o atados, con ruedas o patas los cuales se ajustan a las necesidades del usuario según la aplicación a realizar. Por esta razón el software usado en el diseño de este tipo de plataformas es costoso y robusto, además se deben tener conocimientos previos que estén a la altura tanto en programación como en diseño de hardware para cumplir con todos los requerimientos que exige el diseño de un robot. Todas estas tareas se hacen mucho más sencillas con la ayuda de Microsoft Robotics Developer Studio (MRDS), ya que es un software gratuito usado para simulación e

### JOHNNY MAURICIO ARIAS H.

Candidato a Ingeniero Electrónico.  
Universidad Tecnológica de Pereira  
Semillero Investigación Colossus  
johmauarias@utp.edu.co

### MARCELO BERNAL GÓMEZ.

Candidato a Ingeniero Electrónico.  
Universidad Tecnológica de Pereira  
Semillero Investigación Colossus  
mabernal@utp.edu.co

### ESTEBAN M. CORREA

#### AGUDELO

Estudiante de Maestría en Ingeniería  
Universidad de los Andes  
em.correa20@uniandes.edu.c

implementación de robots [1], este cuenta con varios servicios para la creación de aplicaciones al control de robots de forma tal que cualquier persona esté al alcance de utilizarlo manejando un mismo lenguaje de maquina en todo momento, ya sea mediante código o programación visual, facilitando el control de los robots físicos y simulados mediante el uso del mismo código fuente, llegando a ser un software completo y competente utilizado hoy en día en muchas de las aplicaciones de investigación, ofreciendo muchas oportunidades para los programadores, ya sea usted un aficionado, un estudiante de investigación, o un empleado de una gran corporación.

Este documento se encuentra dividido en tres etapas: La primera es la introducción a la tecnología usada por MRDS identificando cada una de los cinco paquetes que conforman a Microsoft Robotics, la segunda contiene la descripción del hardware utilizado en la implementación real con MRDS y por último se desglosan los resultados obtenidos de la realización de una implementación robótica mediante una interfaz Control Genérico USB- Computadora- Bluetooth y Kit Lego Mindstorms en configuración Tribot.

## 2. DESCRIPCIÓN DEL SOFTWARE

Microsoft Robotics Developer Studio es una plataforma integrada .NET, se basa en un entorno de desarrollo altamente escalable para distribuida Gamma de aplicaciones y concurrencia de procesos, diseñada para la creación, programación, simulación e implementación de diferentes plataformas Robóticas [2]. Consta de cinco componentes: Concurrencia en Tiempo de ejecución y Coordinación (CCR), Descentralización de Servicios de Software (DSS), Lenguaje de Programación Visual (VPL) y Entorno de simulación visual (VSE) [3] (Ver figura 1).



Figura 1. Componentes del MRDS.

### 2.1. CONCURRENCIA EN TIEMPO DE EJECUCIÓN Y COORDINACIÓN (CCR).

El CCR es una biblioteca de gestión que proporciona clases y métodos para ayudar a la concurrencia, coordinación y la verificación de errores en el sistema. Lo que hace menos complejo el manejo de entradas y salidas asincrónicas, eliminando las dificultades en el manejo de activación y desactivación de cada uno de los subsistemas. Está orientado al Descentralizado Servicios de Software (DSS). Este marco le permite crear módulos de programas que pueden inter-operar en un robot y en los ordenadores personales conectados mediante un protocolo simple y abierto permitiendo escribir los segmentos de código que operan de forma independiente.

Se comunica cuando sea necesario mediante el envío de mensajes. Cuando se recibe un mensaje, se coloca en la cola, llamado puerto, hasta que pueda ser procesado por el receptor. Cada segmento de código se pueden ejecutar simultáneamente y de forma asincrónica, y muchas veces no hay necesidad de sincronizarlos a causa de las colas de mensajes. Cuando es necesario esperar, la biblioteca CCR proporciona las construcciones necesarias de procesamiento de datos. La CCR proporciona uno o varios despachadores, que determinan qué segmento de código está actualmente en ejecución, un elemento de programación asincrónica que a menudo se pasa por alto es el control de errores. Es posible el uso de excepciones para aislar las fallas dentro de un segmento de código único, pero los datos que causaron el error se aprobaron a través de un mensaje de otro segmento en la ejecución y la excepción del código. El controlador no tiene manera

de saber dónde se originó la operación, o cuando el error se debe informar. Para resolver este problema, el CCR proporciona causalidades. Una relación de causalidad contiene una referencia a un puerto que se utiliza para informe de fallos. La causalidad se asocia con un mensaje y se sigue con el mensaje que pasa a otro segmento de código y luego de nuevo a otro segmento. En cualquier momento, si se produce un error, un mensaje se envía al puerto de error asociado con la causalidad. De esta manera, los errores se informan a la serie de sesiones de código que se inició.

### 2.2. SERVICIOS DESENTRALIZADOS DE SOFTWARE (PROTOCOLO DSS).

El DSS es un módulo usado para la construcción de aplicaciones basadas en un modelo de servicios de acoplamiento flexible y su función principal es administrar el flujo de mensajes a través de los puertos, este está dividido en dos secciones:

#### 2.2.1 DSS MANIFIEST EDITOR (DSSME).

El DSSME proporciona la creación de escenarios de configuración de la aplicación. La ventana DSSME incluye menús para crear y cargar servicios permitiendo el ahorro de memoria, así como la edición de las entradas que se cargaron en el Manifest. En general, el editor crea un proyecto que consiste en un archivo, así como cualquier configuración de los archivos asociados al programador de aplicaciones. También incluye una barra de herramientas de acceso rápido a comandos comunes, y tres ventanas de cuadro de herramientas. Las tres ventanas son:

- **Ventana de Servicios.**
- **Propiedades y Caja de herramientas:** muestra los servicios disponibles para incluir en el Manifest y los parámetros de configuración para algunos servicios.
- **Herramientas del proyecto:** muestra el archivo de manifiesto y los archivos de configuración asociados (Ver figura 2).

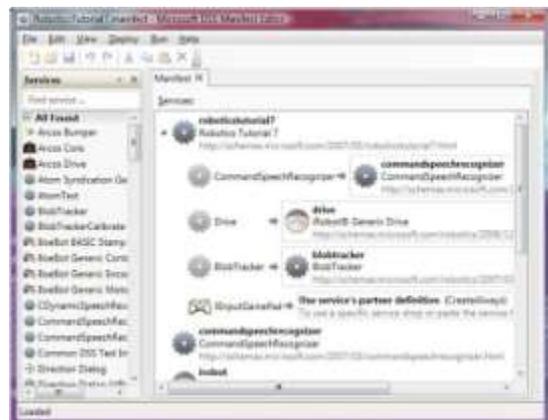


Figura 2. Microsoft DSS Manifest Editor [4]

### 2.2.2. DSS LOG ANALYZER (DSSLA).

El DSSLA permite ver los flujos de mensajes a través de servicios múltiples DSS, permitiendo inspeccionar los detalles del mensaje. Los flujos de filtro de mensajes pueden ser modificados a través de servicios múltiples DSS mediante mensajes (Ver figura 3).

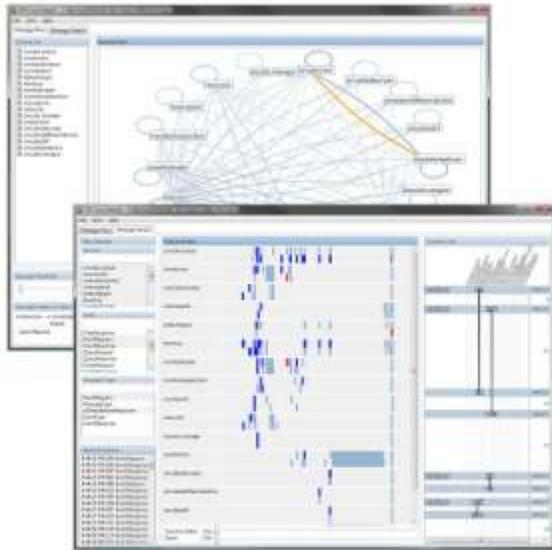


Figura 3. Microsoft DSS Log Analyzer Tool [4]

Al integrar las dos herramientas (CCR y DSS) en un proyecto de MRDS, se obtiene como resultado la siguiente estructura de operación (Ver figura 4):

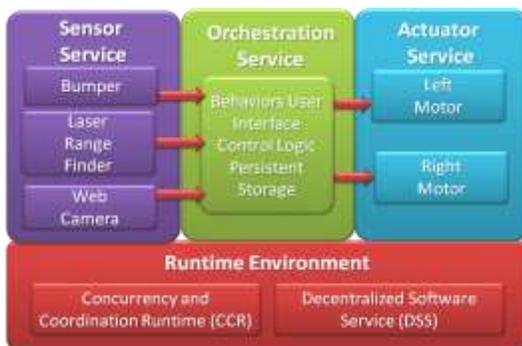


Figura 4. Sistema Integrado en Microsoft Robotics [2].

### 2.3. LENGUAJE DE PROGRAMACIÓN VISUAL (VPL).

VPL proporciona un lenguaje simple de diagrama de flujo en lugar de una típica secuencia de comandos (Ver figura 5), facilitando la creación de aplicaciones robóticas. VPL también proporciona la capacidad de asumir un conjunto de bloques conectados y reutilizarlos como único bloque en otra parte de su programa, siendo capaz de generar código legible mediante C #.

Cada bloque representa un servicio, un cálculo, una condición, o un diagrama anidado, y estos se llaman las actividades. Las líneas entre los bloques representan los mensajes que fluyen de un servicio a otro. Los datos de flujo funciona bien para representar los mensajes que se pasan en una aplicación típica de DSS. Los diagramas VPL pueden ser arbitrariamente complejos, ya que cada bloque puede representar otro diagrama anidado, que a su vez puede tener otros diagramas anidados.

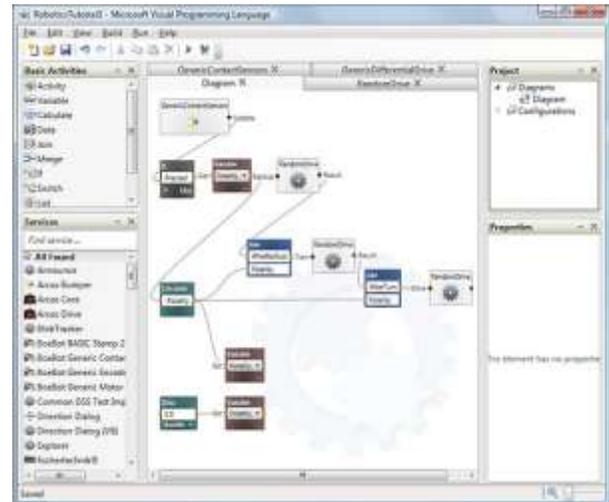


Figura 5. Plataforma del VPL [4].

### 2.4. ENTORNO DE SIMULACIÓN VISUAL (VSE).

El VSE proporciona la capacidad de simular y probar aplicaciones robóticas utilizando una herramienta de simulación basado en la física 3D (Ver figura 6). Esto permite a los desarrolladores crear aplicaciones de robótica sin el hardware. Ya que es mucho más barato destruir un robot virtual por un error de programación que uno real. Con ayuda de las herramientas genéricas de MRDS podemos simular la aplicación sin tener los dispositivos reales de los que está compuesto el Robot, dando las opciones apropiadas para que su simulación se adapte a los requerimientos de diseño deseados.

MRDS facilita hacer retos difíciles de software presentes en la robótica como la coordinación, observabilidad, configuración, implementación y reutilización. Además interactúa con un simulador sofisticado 3D con un motor de física y un modo de edición que permite crear nuevos escenarios con las entidades existentes de simulación para garantizar fidelidad y suplir todos los requisitos de la escala de los escenarios de simulación avanzada. En este es posible interactuar con gran cantidad de dispositivos mecánicos y electrónicos como motores, sensores, engranajes, webcams entre otros los cuales permiten al desarrollador crear servicios para una amplia variedad de hardware y de esta forma darle ese toque real a nuestro robot.

El entorno de simulación contiene modelos de varios robots, incluyendo un NXT de LEGO Mindstorms, iRobot Creator, Mobile Robots de Pioneer 3DX y un brazo robótico LBR3 KUKA entre otros.



Figura 6. Entorno de visualización VSE [5]

Además de todos estos componentes, el equipo ha puesto en marcha MRDS con numerosas muestras y ayudas completas para proporcionar ejemplos de programación y módulos para aplicaciones de usuario, MRDS cuenta con un amplio paquete de herramientas y accesorios que permiten al usuario simular y diseñar sistemas robustos ya que contiene las librerías educativas e industriales existentes en el mercado.

### 3. DESCRIPCIÓN DEL HARDWARE

#### 3.1 Robot: Mindstorms NXT Fabricate: LEGO

Mindstorms NXT es un kit completo de la compañía LEGO que no requiere herramientas y habilidades especiales para su montaje, sirve para construir cualquier tipo de robot. La inteligencia básica es proporcionada por un NXT, siendo este un micro-ordenador inteligente "ladrillo" integrado por una pantalla LCD, botones, un procesador ARM (32 bits), S.O propio (Firmware), funciona con 6 pilas AA (9V) y conexión Bluetooth para interconexión inalámbrica con su PC (Ver figura 7). Este es programable mediante puerto USB, Bluetooth y software NXT-G. Puede interactúa con 4 sensores y 3 actuadores (motores) a la vez mediante puertos de entrada y de salida [6].

El kit consta de:

- 619 elementos para crear sus propios robots.
- LEGO TECHNIC: Elementos de construcción como engranajes, ruedas, cadenas y neumáticos.
- 1 NXT micro-ordenador que actúa como cerebro.
- 2 sensores de contacto (Tacto).
- 1 sensor ultrasónico (Movimiento).
- 1 sensor de color o lámpara (RGB).

- 7 cables de conexión para conectar motores y sensores.
- Guía de usuario (Instrucciones) y demostraciones de 4 tipos de robots.
- CD con software de programación NXT-G16.
- 3 Servo motores (Precisión de +/- 1 grado) y ruedas.
- Engranajes y ejes.
- Firmware propio (Versión 2 o 1.05).



Figura 7. Tecnología NXT [6].

Otras Características y Recomendaciones:

- Al actualizar el firmware de este NXT debe tener en cuenta la selección correcta del firmware ya que las dos versiones son incompatibles entre sí.
- NXT es compatible Microsoft Windows (XP, Vista y Seven) y MacOS X (v10.4 ó v10.5).

#### 3.4. OTRAS PLATAFORMAS:

En la actualidad MRDS cuenta con una cantidad de plataformas para la realización de sus implementaciones, las cuales pueden variar con respecto al grado de complejidad y las acciones que necesite realizar el robot.

Las siguientes se destacan por ser utilizadas en diversas aplicaciones industriales y militares, además son las sugeridas por los fabricantes de MRDS desde la página oficial [1]:

- iRobot [7]
- Beo-Bot (Parallax)
- Pioneer 3Dx [8]
- Lynx 6 Robotics Arm (Lynxmotion)
- Stinger, Tranxter & Serializer (Robotics Connections)
- CoroBot (CoroWare)

### 4. DESARROLLO DE LA IMPLEMENTACIÓN

Se realizó una implementación utilizando servicios genéricos existentes en Microsoft Robotics Developer Studio R2, haciendo uso del entorno Visual Programming

(VPL) explicado anteriormente. Posteriormente se realizará una simulación en el entorno visual VSE 3D con las características reales del LEGO Mindstorms.

#### 4.1. Creación de la interfaz VPL

Al crear un nuevo proyecto, se incluyen los dos Servicios necesarios para la creación de la interfaz, el GameController es el servicio encargado de reconocer el control USB y el GenericDiferencial Drive (Ver figura 8), este último se encargara de generar el giro diferencial enviando la activación a cada uno de los motores.

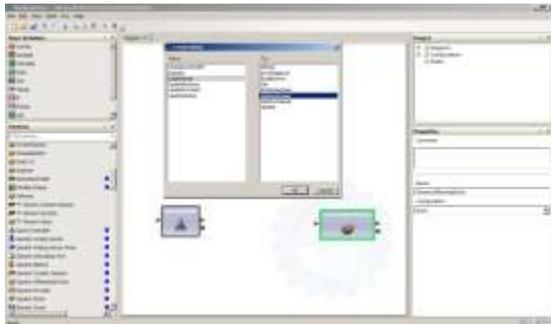


Figura 8. Configuración del la interfaz (VPL).

A continuación se unen los dos servicios con una conexión de envío de mensajes, estableciendo la configuración de entrada en UpdateAxes y la salida en SetDrivePower (Ver figura 9).

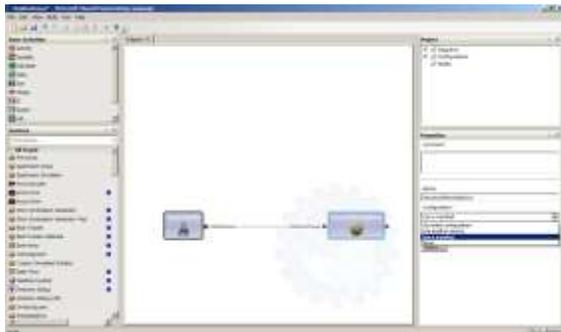


Figura 9. Configuración del la interfaz (VPL).

Por último se carga el Manifest Simulation (Ver figura 10) encargado de establecer la conexión para el envío de mensajes al hardware simulado.

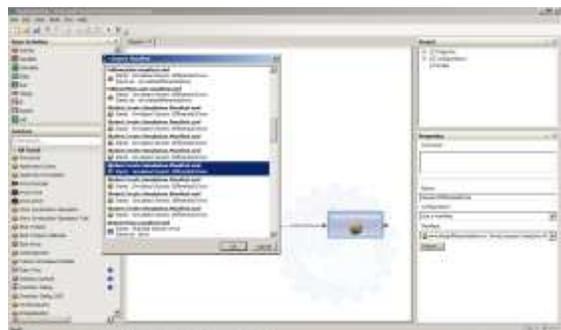


Figura 10. Configuración del Manifest LEGO (VPL).

En primer lugar se simula el sistema para comprobar fallos y corregirlos antes de ser implementados físicamente. Por esta razón se compila el código fuente y estableciendo la interfaz de forma rápida, controlando el Kit Lego Mindstorms simulación (Ver figura 11) en configuración Tribot por medio del Joystick de un control genérico USB conectado a la computadora portátil.

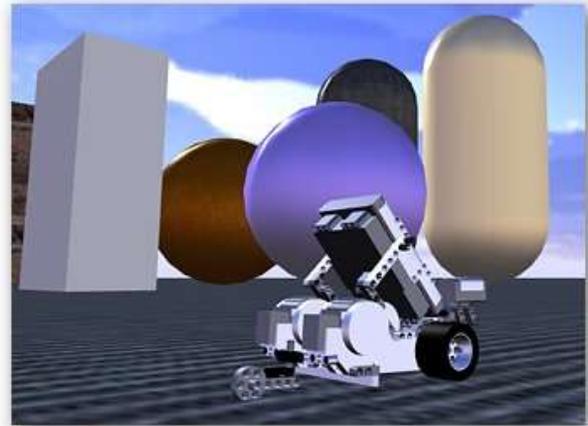


Figura 11. Simulación LEGO Mindstorms (VSE).

Por último para pasar de la simulación a la implementación se cambia el Manifest Simulation por el Manifest LEGO Mindstorms Implementation, se compila cargando cada una de las librerías necesarias para la comunicación, obteniendo como resultado una interfaz confiable que cumple todas las normas de comunicación en el área de la robótica (Ver Figura 12). El envío de los datos se realiza por medio de un dispositivo USB-Bluetooth conectado a la computadora portátil.



Figura 12. Implementación en Tiempo Real.

## 5. CONCLUSIONES Y RECOMENDACIONES

- MRDS es una herramienta valiosa para el diseño de robots ya que cuenta con diversas librerías que permiten al usuario suplir los requerimientos necesarios en sus implementaciones.
- Esta herramienta permite a los aficionados y expertos de la robótica manejar una interfaz competente en el mercado de forma gratuita y menos compleja, manejando un entorno gráfico interactivo y accesible de alto nivel.
- Los cinco paquetes principales de MRDS conforman un simulador de alta gama donde se integra los procesos fundamentales en el desarrollo de una aplicación robótica, desde el flujo de datos, ciclos de control hasta construcción física simulada del prototipo, eliminando códigos de error en los procesos para que a la hora de ser implementados se cuente con un mínimo margen de error y no perder dinero y tiempo.
- Las plataformas hardware utilizadas a la hora de implementar los diseños desarrollados en MRDS permite al programador manejar un mismo lenguaje en todo momento es por esto que estas se hacen mucho más rápidas, eficaces y menos complejas.
- Microsoft Robotics también permite la creación de proyectos con hardware no existente en sus librerías, permitiendo al usuario manejar los dispositivos electrónicos y estructura robótica deseada, con la condición que todo será creado desde cero mediante otras herramientas extras (Ver figura 14) como: AGEIA PhysX engine (Simulación de interfaces físicas), Microsoft DirectX (Api para acceso a gráficos), Framework Microsoft XNA (Interfaz Gráfica):



Figura 14. Software para la creación de interfaces extras.

Estas implementaciones y estudios en Microsoft Robotics Developer Studio se realizan actualmente en el semillero de investigación Colossus.NET perteneciente a la Universidad Tecnológica de Pereira, a medida que se ha avanzado en el manejo del software se han dictado conferencias en las facultades de ingenierías dando a

conocer el software como uno de los simuladores robustos en el área de la robótica, lo mejor de este es que es una herramienta totalmente gratuita.

Haciendo énfasis en el área de la investigación, actualmente en el semillero se realizan investigaciones para lograr otras implementaciones como el reconocimiento de formas y colores haciendo uso de la visión artificial que serán expuestos en el transcurso de este año 2011.

## 6. BIBLIOGRAFÍA

- [1] Microsoft Robotics Developer Studio 2008 R3, "Product Information", ©2011 Microsoft. En línea: <http://www.microsoft.com/robotics/#Product>.
- [2] Kyle Johns and Trevor Taylor, Professional Microsoft® Robotics Developer Studio, Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana.
- [3] Microsoft Robotics Developer Studio 2008 R3, "A Platform for Developing Robotics Applications," Data Sheet ©2011 Microsoft, Mayo 2002.
- [4] msdn Library, "User Guide and Class Reference of Microsoft Robotics," ©2010 Microsoft Corporation. All Rights Reserved. En línea: <http://msdn.microsoft.com/en-us/library/bb881626.aspx>
- [5] Ayuda Electronica, El blog elegido por los estudiantes de Ingeniería Electrónica, "Microsoft libera Robotics Studio Gratis, ", 27 May 2010. En línea: <http://ayudaelectronica.com/microsoft-robotics-studio-gratis/>.
- [6] Alvites Juárez J. Carlos and Quiroga Martin, "Manual de LEGO MINDSTORMS NXT, " Robótica, Instituto Superior Leonardo Da Vinci, Edited by Foxit PDF Editor, Copyright © by Foxit Software Company, 2004-2007 For Evaluation Only.
- [7] iRobot Create, Programmable Robot. En línea: <http://store.irobot.com/shop/index.jsp?categoryId=3311368>
- [8] MobileRobots Pioneer 3 (P3 DX), Conscious Robots. En línea: <http://www.consciousrobots.com/en/reviews/robots/mobilerobots-pioneer-3-p3-dx-8.html>.