



**UNIVERSIDAD CATÓLICA DE SANTIAGO DE
GUAYAQUIL**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO(A) EN TELECOMUNICACIONES CON MENCIÓN EN
GESTIÓN EMPRESARIAL**

TEMA:

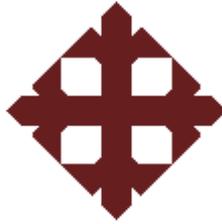
**ESTUDIO DE LA APLICACIÓN DIDÁCTICA DEL LEGO
MINDSTORMS NXT PARA LA ENSEÑANZA DE LA
ROBÓTICA EN LA CARRERA DE INGENIERÍA EN
TELECOMUNICACIONES Y ELECTRÓNICA EN
CONTROL Y AUTOMATISMO.**

ALUMNOS:

Kerry Emmanuel Sagal Espinoza
Fredy Antonio Mora Pallo

DIRECTOR

MSC. EDWIN PALACIOS MELÉNDEZ



TESIS DE GRADO

**ESTUDIO DE LA APLICACIÓN DIDÁCTICA DEL LEGO
MINDSTORMS NXT PARA LA ENSEÑANZA DE LA
ROBÓTICA EN LA CARRERA DE INGENIERÍA EN
TELECOMUNICACIONES Y ELECTRÓNICA EN
CONTROL Y AUTOMATISMO.**

**Presentada a la Facultad de Educación Técnica para el Desarrollo, Carrera
de Ingeniería en Telecomunicaciones de la Universidad Católica de
Santiago de Guayaquil**

REALIZADO POR:
Kerry Emmanuel Sagal Espinoza
Fredy Antonio Mora Pallo

**Para dar cumplimiento con uno de los requisitos para optar por el título de:
Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial**

MsC. Edwin Palacios Meléndez
Director de Tesis

Ing.....
Vocal

Ing.....
Vocal

MBA. Héctor Cedeño Abad
Decano de la Facultad

MsC. Manuel Romero Paz
Director de Carrera

CERTIFICACIÓN

Certifico que el proyecto de grado titulado “ESTUDIO DE LA APLICACIÓN DIDÁCTICA DEL LEGO MINDSTORMS NXT PARA LA ENSEÑANZA DE LA ROBÓTICA EN LA CARRERA DE INGENIERÍA EN TELECOMUNICACIONES Y ELECTRÓNICA EN CONTROL Y AUTOMATISMO” desarrollado por Kerry Emmanuel Sagal Espinoza y Fredy Antonio Mora Pallo fue realizado, corregido y terminado, razón por la cual está apto para su presentación y sustentación.

**Kerry Emmanuel Sagal Espinoza
Fredy Antonio Mora Pallo**

**MsC. Edwin Palacios Meléndez
DIRECTOR DE TESIS**

Índice General

Agradecimientos	9
Dedicatoria.....	10
Resumen	11
Abstract.....	12
CAPÍTULO 1: GENERALIDADES	13
1.1. INTRODUCCIÓN	13
1.2. ANTECEDENTES	13
1.3. DEFINICIÓN DEL PROBLEMA	15
1.4. JUSTIFICACIÓN DEL PROBLEMA.....	15
1.5. OBJETIVOS DE LA INVESTIGACIÓN	16
1.5.1. OBJETIVO GENERAL.....	16
1.5.2. OBJETIVOS ESPECÍFICOS.....	16
1.6. IDEA A DEFENDER.....	17
1.7. METODOLOGÍA	17
CAPÍTULO 2: ESTADO DEL ARTE DE LA ROBÓTICA Y LEGO MINDSTORMS NXT	18
2.1. ROBÓTICA	18
2.2. ROBÓTICA Y AUTOMATIZACIÓN EN LA SOCIEDAD.....	20
2.3. ROBOTS MÓVILES	25
2.4. ROBOT LEGO MINDSTORM NXT	26
2.4.1. INTRODUCCIÓN A LEGO MINDSTORMS.....	27
2.4.2. Por qué escoger el robot LEGO Mindstorms	28
2.5. OTRAS PLATAFORMAS (ROBOTS) SIMILARES A LEGO MINDSTORM NXT	29
2.5.1. PLATAFORMA FISCHERTECHNIK	29
2.5.2. PLATAFORMA ARDUINO	30
2.5.3. PLATAFORMA PARALLAX.....	31
2.5.4. SR1.....	32
2.5.5. PLATAFORMA ROBOTIS BIOLOID.....	33
2.5.6. PLATAFORMA ROBOBUILDER.....	34
2.6. CARACTERÍSTICAS Y COMPONENTES DE LA PLATAFORMA LEGO MINDSTORMS NXT.....	35
2.6.1 MICROCONTROLADOR NXT.....	36
2.6.2 MOTORES DEL NXT.....	37
2.6.3 SENSORES DEL NXT	38

2.6.4	COMUNICACIÓN USB Y BLUETOOTH	41
CAPÍTULO 3: ESTADO DEL ARTE DE LA PROGRAMACIÓN DEL LEGO MINDSTORMS NXT		
		43
3.1.	INTRODUCCIÓN A LAS HERRAMIENTAS DE PROGRAMACIÓN.....	43
3.2.	PROGRAMACIÓN NXT.....	44
3.3.	PLATAFORMAS COMERCIALES DE PROGRAMACIÓN DEL NXT.....	45
3.3.1.	SOFTWARE NXT-G EDUCACIÓN.....	45
3.3.2.	SOFTWARE NXT-G	47
3.3.3.	SOFTWARE ROBOLAB	48
3.3.4.	SOFTWARE ROBOTC	50
3.4.	PLATAFORMAS DE LIBRE USO PARA PROGRAMACIÓN NXT.....	52
3.4.1.	LABVIEW TOOLKIT	52
3.4.2.	MATLAB y SIMULINK.....	53
3.4.3.	BRICXCC	55
3.4.4.	ENCHANTING	56
3.4.5.	NBC.....	56
3.4.6.	NXC.....	57
3.4.7.	LEJOS NXJ.....	57
3.4.8.	MICROSOFT ROBOTICS DEVELOPER STUDIO.....	58
CAPÍTULO 4: DESARROLLO EXPERIMENTAL DEL ROBOT LEGO MINDSTORMS NXT.....		
		60
4.1.	PRÁCTICA 1: Sistema de Defensa Robótico – Francotirador.....	60
4.2.	PRÁCTICA 2: Robot Hexápodo - Strider.....	74
4.3.	PRÁCTICA 3: Robot escalador.....	85
4.4.	PRÁCTICA 4: Brazo robótico autónomo.....	92
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.....		
		99
5.1.	CONCLUSIONES.....	99
5.2.	RECOMENDACIONES.....	100
REFERENCIAS BIBLIOGRÁFICAS.....		101
Anexo A.....		103

Índice de Figuras

Capítulo 1

Figura 1. 1: Ladrillo de control (microprocesador) del Mindstorms RCX.....	14
--	----

Capítulo 2

Figura 2. 1: Los robots C3P0 y R2D2.	19
Figura 2. 2: Primeros robots industriales. (a) Devo- Engelberger fundan Unimation (1956). Primer robot industrial, (b) Primer robot con accionamiento eléctrico: IRb6 ASEA (1973); y (c) Robot SCARA del Prof. Makino (Univ. Yamanashi de Japón-1982)	21
Figura 2. 3: Robots inspirados en los humanos, llamados humanoides:	22
Figura 2. 4: Robots aplicados a la industria: (a) Fabricación y	24
Figura 2. 5: Robots aplicados en la medicina:.....	25
Figura 2. 6: Robots móviles (a) explorador terrestre, y.....	26
Figura 2. 7: Plataforma creada con Fischertechnik	30
Figura 2. 8: Plataforma tipo placa de Arduino	31
Figura 2. 9: Plataforma tipo móvil de PARALLAX	32
Figura 2. 10: Plataforma del Robot móvil SR1	33
Figura 2. 11: Plataforma del Robotis Bioloid	33
Figura 2. 12: Plataforma del Robot RoboBuilder	34
Figura 2. 13: Partes principales del LEGO Mindstorms NXT que son:	35
Figura 2. 14: Microcontrolador NXT	36
Figura 2. 15: Especificaciones técnicas del bloque Mindstorms NXT	37
Figura 2. 16: Motor del NXT.....	37
Figura 2. 17: Engranajes del servomotor NXT.	38
Figura 2. 18: Sensor de contacto.	38
Figura 2. 19: Sensor de sonido.	39
Figura 2. 20: Sensor de luz.	39
Figura 2. 21: Sensor ultrasónico.	40
Figura 2. 22: Cobertura del sensor ultrasónico.....	40
Figura 2. 23: Comunicación entre PC y NXT a través del cable USB.....	41
Figura 2. 24: Comunicación inalámbrico entre PC y NXT a través del bluetooth.....	41

Capítulo 3

Figura 3. 1: Variedad de textos publicados por profesionales y de la comunidad del LEGO.	45
---	----

Figura 3. 2: Software de programación NXT-G Education.	46
Figura 3. 3: Software de programación NXT-G.	47
Figura 3. 4: Software de programación Robolab.	49
Figura 3. 5: Entorno de la programación en Robolab.	49
Figura 3. 6: Entorno de programación RobotC.	51
Figura 3. 7: Ejemplo de un programa en el entorno de programación en C.	51
Figura 3. 8: Entorno de programación gráfica a través de LabVIEW.	52
Figura 3. 9: Entorno de programación textual en Matlab para el NXT.	53
Figura 3. 10: Entorno de programación por diagrama de bloques en Simulink de Matlab.	54
Figura 3. 11: Entorno de programación IDE de la plataforma BricxCC.	55
Figura 3. 12: Entorno de programación Scratch de la plataforma Enchanting.	56
Figura 3. 13: Plataforma de programación NBC.	57
Figura 3. 14: Plataforma de programación Lejos de Java.	58
Figura 3. 15: Plataforma de programación Lejos de Java.	59

Capítulo 4

Figura 4. 1: Sistema de Defensa Robótico.	60
Figura 4. 2: Piezas para la construcción del sistema de defensa robótico.	61
Figura 4. 3: Paleta completa de programación.	63
Figura 4. 4: Diseño de bloques para el funcionamiento de los sensores de color.	64
Figura 4. 5: Configuración del loop para el sensor de color.	64
Figura 4. 6: Configuración de cada uno de los sensores de color.	65
Figura 4. 7: Diseño de bloques para el funcionamiento de los motores.	65
Figura 4. 8: La configuración individual de los motores.	66
Figura 4. 9: Primera parte de la programación básica modo autónomo.	67
Figura 4. 10: Configuración interna del motor para determinada aplicación.	67
Figura 4. 11: Segunda parte de la programación básica modo autónomo.	68
Figura 4. 12: Configuración interna para otra manera de programación básica de modo autónomo.	68
Figura 4. 13: Tercera parte de la programación básica modo autónomo.	69
Figura 4. 14: Configuración de los bloques adicionales del paso 3.	69
Figura 4. 15: Cuarta parte de la programación básica modo autónomo.	70
Figura 4. 16: Configuración de los bloques adicionales del paso 4.	70
Figura 4. 17: Programa final básica del sistema de defensa robótico autónomo.	71
Figura 4. 18: Configuración del bloque espera para la programación intermedia.	72
Figura 4. 19: Programa final intermedia del sistema de defensa robótico autónomo.	72

Figura 4. 20: Configuración final para la programación intermedia.....	73
Figura 4. 21: Robot Hexápodo – Strider.....	74
Figura 4. 22: Piezas para la construcción del robot hexápodo – Strider.....	75
Figura 4. 23: Vista de la conexión entre los puertos del NXT y los sensores del robot hexápodo Strider.	76
Figura 4. 24: Programación básica para mover las patas del robot hexápodo.	77
Figura 4. 25: Configuración básica para ejecución correcta de los motores.....	77
Figura 4. 26: Programación de bloques para el movimiento del robot hexápodo.	78
Figura 4. 27: Configuración del tiempo que se moverá el robot.	78
Figura 4. 28: Programación del paso 1 para el robot hexápodo.	79
Figura 4. 29: Configuración de los bloques del paso 1.....	79
Figura 4. 30: Programación del paso 2 para el robot hexápodo.	80
Figura 4. 31: Configuración de los bloques del paso 2.....	80
Figura 4. 32: Programación del paso 3 para el robot hexápodo.	81
Figura 4. 33: Configuración de los bloques del paso 3.....	82
Figura 4. 34: Programación del paso 4 y 5 para el robot hexápodo.	83
Figura 4. 35: Configuración de los bloques del paso 4.....	83
Figura 4. 36: Configuración de los bloques del paso 5.....	84
Figura 4. 37: Programación intermedia del robot hexápodo.	84
Figura 4. 38: Configuración de los bloques de programación intermedia.	85
Figura 4. 39: Vista del robot escalador.....	86
Figura 4. 40: Piezas para la construcción del robot escalador.	86
Figura 4. 41: Programación del paso 1 para el robot escalador de chimenea.	88
Figura 4. 42: programación y configuración de los bloques del paso 2, encendiendo los motores de las ruedas y extendiendo los brazos.	89
Figura 4. 43: Programación y configuración de los bloques del paso 3.....	91
Figura 4. 44: Brazo robótico modo autónomo.	92
Figura 4. 45: Mecanismo de funcionamiento del brazo robótico modo autónomo.	93
Figura 4. 46: Piezas para construcción del brazo robótico modo autónomo.....	94
Figura 4. 47: Programación y configuración del brazo robótico para agarrar objetos ..	95
Figura 4. 48: Programación y configuración del brazo robótico para liberar objetos....	96
Figura 4. 49: Programación y configuración del brazo robótico para encontrar objetos.	97
Figura 4. 50: Programación y configuración del brazo robótico para liberar objetos....	97
Figura 4. 51: Programación y configuración del brazo robótico acercándose a objetos	98
Figura 4. 52: Programa final para la operación completa del brazo robótico.	98

Agradecimientos

Doy gracias a Dios por su bendita misericordia, porque este día me permite llegar a un año más de vida, por todas las muestras de cariño que he recibido, la fe el esfuerzo y el optimismo dedicado a lo largo de los años de estudio son el fruto de la gente que creyó en mi persona .

Señor gracias por esa tremenda bendición de darme unos padres únicos e irremplazables en el mundo ya que ellos me apartaron de lo malo me inculcaron buenos valores se preocuparon por mis problemas, por sus desvelos, porque nunca me fallaron, porque siempre estuvieron ahí, tendiéndome un cálido abrazo, siempre creyeron en mí, por eso son mi modelo a seguir para que mis futuros hijos se sientan orgullosos de mi así como yo estoy orgulloso de mis padres.

También te pido que bendigas a mis 4 hermanos y 2 hermanas porque a pesar de tantas adversidades que vivimos nunca dejamos de estar unidos luchando por un solo fin, “la salud de mi amada madre” y eso fue lo que me ayudo a seguir frenteando a la vida, porque no flaqueamos ni renegamos de ti Padre, gracias a la perseverancia obtuvimos tremenda recompensa. Te pido que los proveas Señor mío porque ellos han sido buenos conmigo cuando estuve medio lázaro jeje.

Te agradezco Padre por poner esa personita especial en mi vida que me enseñó muchas cosas y fue mi compañera en tiempos de angustias y alegrías, gracias por mis nuevos amigos que aportaron mucho en el cambio de vida al seguir tus caminos.

Señor te pido que bendigas a nuestro director de tesis, el Ing. Fernando Palacios porque nos propuso un excelente tema de tesis, que sé que ayudaran a las nuevas generaciones en la Facultad Técnica, Señor se tu quien provea a su familia tenlos con salud y lo más importante, es que sientan más tu presencia en sus corazones y que sepan que no hay nada imposible para ti mi Dios.

Todo esto te lo pido y te agradezco mi Dios en nombre de nuestro Señor Jesucristo Amen y Amen

No temas, porque yo estoy contigo; no desmayes,
porque yo soy tu Dios que te esfuerzo;
siempre te ayudaré,
siempre te sustentaré con la diestra de mi justicia.

Isaías 41:10

Dedicatoria

A nuestros padres por su entrega abnegada y sublime siendo pilares fundamentales para culminar nuestra vida universitaria, a nuestros hermanos y amigos por su apoyo incondicional ayudándonos con sus experiencias y consejos para así cumplir nuestras metas, convirtiéndonos en profesionales de éxito y mejores personas.

A nuestro amigo, que como Director de Tesis nos supo ayudar a realizar un trabajo digno y de mucho provecho para las siguientes promociones de estudiantes que se inclinen a la Ingeniería en Telecomunicaciones.

Resumen

La presente tesis analiza el potencial del kit LEGO MINDSTORMS NXT (<http://mindstorms.lego.com/>) para ser utilizado como plataforma de bajo coste que permita a los alumnos adquirir las competencias relativas a informática industrial dentro de los estudios de ofrece la Facultad Técnica para el Desarrollo. Para ello, se compara con otras plataformas similares y se analizan algunas de las diferentes herramientas de programación disponibles sobre LEGO MINDSTORMS.

Además, se propone la creación de Fundamentos de Robótica o simplemente Robótica con programa de estudios actualizado en las carreras de Telecomunicaciones y Electrónica en Control de la UCSG. Se utiliza la metodología Project Based Learning (PBL) para proponer a los alumnos de pregrado pequeños proyectos donde experimenten la problemática habitual del desarrollo de aplicaciones empotradas.

De hecho, como se verá la plataforma LEGO proporciona un entorno de trabajo muy rico con el que además de trabajar las competencias específicas relativas a la programación industrial resulta valioso para trabajar competencias de otras materias como Instrumentación Electrónica, Control Automático, Microcontroladores, Robótica o Mecatrónica.

Abstract

This thesis analyzes the potential of LEGO MINDSTORMS NXT kit (<http://mindstorms.lego.com/>) to be used as low-cost platform that allows students to acquire skills related to industrial computer within studies offers Technical Development Department. This is compared with other similar platforms and discusses some of the different programming tools available for LEGO MINDSTORMS.

It is also proposed creating Fundamentals of Robotics or simply Robotics with updated curriculum racing Telecommunications and Electronics in Control UCSG. Approach is used Project Based Learning (PBL) to propose a small undergraduate students in projects where experience the usual problems of developing embedded applications.

In fact, as will the LEGO platform provides a rich working environment in addition to specific skills to work on industrial programming work is valuable skills in other subjects such as Electronic Instrumentation, Automatic Control, Microcontrollers, Robotics and Mechatronics.

CAPÍTULO 1: GENERALIDADES

1.1. INTRODUCCIÓN

En la primera década del 2000 hasta la actualidad, la robótica, un área multidisciplinaria, se ha desplegado y evolucionado en universidades de todo el mundo, promoviéndose cada vez más su desarrollo mediante concursos de robótica (varias modalidades), como el concurso CER-2010 que ocurrió este año 2011 en la Universidad Católica de Santiago de Guayaquil. A pesar de que la implementación de un laboratorio de robótica está fuera del presupuesto de muchas universidades por los altos costos de robots manipuladores y móviles comerciales, es posible hacer prácticas de robótica con paquetes comerciales de bajo costo y construcciones propias.

El uso de los robots y de las tareas para las cuales fueron construido, se han ido diversificándose, como por ejemplo se pueden encontrar robots que realizan (ejecutar) tareas muy distintas para las cuales se pensaba que los robots solo eran utilizados para el ensamblaje de automóviles y exploración (navegación) de planetas. Las formas y los tamaños son diversos, se pueden encontrar robots grandes y pequeños, en forma de brazo utilizado para de pintura automotriz y en forma de insectos respectivamente.

1.2. ANTECEDENTES

LEGO Mindstorms NXT no se ha iniciado de un día para otro, sino que ha sido la fusión entre LEGO y el Instituto Tecnológico de Massachusetts (MIT), originalmente desde 1986, y plasmada en un acuerdo, que entre otros aspectos, LEGO apoyará económicamente investigaciones del grupo de epistemología y aprendizaje del MIT, para así obtener nuevas ideas para sus productos, sin tener que pagar regalías.

A partir del apoyo financiero se iniciaron las investigaciones de parte del MIT, nace o converge el **Programmable Brick**, dedicado para catedráticos e investigadores universitarios y adaptables al estudio del aprendizaje, para lo cual LEGO desarrollaría lo que sería la primera generación de Mindstorms, el

RCX (ver Figura 1.1) de las tres que acabaría teniendo. Salió al mercado en septiembre de 1998 a un precio de unos 200 dólares, que superaron toda expectativa, vendiéndose más de 80.000 unidades en tres meses, además de hacerse un hueco en la escena robótica internacional.



Figura 1. 1: Ladrillo de control (microprocesador) del Mindstorms RCX.

De las tres versiones del RCX contaba con un ladrillo inteligente de potencia muy inferior al del NXT (CPU Hitachi H8/3292 a 16MHz, 16 KB de memoria ROM y 32 KB de RAM), adicional manejaba tres entradas para sensores (de color, sonido y tacto) y tres salidas para actuadores (motores de paso, cámara fotográfica, además de un puerto infrarrojo). La presentación del RCX es muy parecida a los bloques tradicionales de LEGO; es decir que tanto los sensores como el microprocesador (bloque programable) tenían la apariencia de ladrillo característica del fabricante.

Debido a los malos resultados de LEGO durante el 2003, con pérdidas cercano a los 200 millones de dólares, se extendió el rumor a inicios del 2004 que abandonaría la línea Mindstorms y volvería a su mercado tradicional de juguetes legos. No obstante para enero del 2006, LEGO resurgió el NXT, que salió al mercado para su comercialización en junio 2006. No son pocos los proyectos de investigación y trabajos que hay ya realizados usando LEGO Mindstorms. Incluso existen diversos proyectos en **Ingeniería de Control Avanzada**. Se lo puede comprobar a través de una rápida búsqueda en Internet para darnos cuenta de las investigaciones y la suficiente cantidad de material disponible, tanto para expertos como para quienes no lo son tanto, sobre los Mindstorms de LEGO.

LEGO introdujo al mercado el NXT enfocado hacia el mercado recreativo (niños y adolescentes), pero su flexibilidad y capacidad de este lo han llevado a ser aceptado como una herramienta de ayuda académica para mejorar el aprendizaje de los estudiantes que se inclinen por estudios avanzados de robótica.

1.3. DEFINICIÓN DEL PROBLEMA

Necesidad de desarrollar aplicaciones con fines didácticos en el Laboratorio de Electrónica a través del LEGO Mindstorms NXT para la enseñanza de robótica en el ámbito de creatividad y programación de alto nivel a los estudiantes de la Facultad de Educación Técnica para el Desarrollo en las carreras de Ingeniería en Telecomunicaciones y Electrónica en Control y Automatismo.

1.4. JUSTIFICACIÓN DEL PROBLEMA

Actualmente, los laboratorios de Electrónica, Microcontroladores, Microprocesadores y de control de la Facultad de Educación Técnica para el Desarrollo, la enseñanza “práctica” se reduce a simulaciones por ordenador, por disponer en general de unas pocas maquetas de procesos, con sus limitaciones, debido al alto precio de éstas. Motores de corriente continua con conexión a PC, plantas simples y procesos industriales a pequeña escala son un material muy específico, que repercute s en el presupuesto de los laboratorios ya mencionados anteriormente.

Por el contrario, un NXT se puede adquirir desde unos \$500 dólares americanos (un precio razonablemente bajo comparado con las maquetas habituales), e incluye, además de una CPU de una potencia a tener en cuenta, una pequeña colección de sensores y actuadores de buena calidad, conjuntamente a una generosa cantidad de piezas de LEGO Technic, que, con un poco de habilidad, permiten construir en principio un amplio abanico de escenas apropiadas para una clase práctica de robótica en las carreras de Ingeniería en Telecomunicaciones y Electrónica en Control y Automatismo.

Así, desde este punto de partida, surge un interesante estudio de las posibilidades de este robot, pese a la desconfianza inicial que pudiera suscitar al estar hablando de un juguete. Pero, como se demuestra aquí, LEGO Mindstorms NXT es mucho más que un juguete. El presente estudio permitirá mejorar y estimular la creatividad de los alumnos de la Facultad Técnica en las carreras de Ingeniería en Telecomunicaciones e Ingeniería Electrónica en Control y Automatismo, así como también la de los docentes en el área de la Electrónica asignatura introductoria de la robótica.

No solamente sería la aplicabilidad en los estudiantes y docentes, sino también de los docentes investigadores de la Universidad Católica de Santiago de Guayaquil, específicamente en el Instituto de Transferencia Tecnológica (ITT). El impacto será incursionar a profundidad en el estudio de la robótica y aplicaciones que servirán de herramienta para futuras investigaciones en el ámbito de la robótica.

1.5. OBJETIVOS DE LA INVESTIGACIÓN

1.5.1. OBJETIVO GENERAL

Realizar el estudio de una aplicación didáctica para la enseñanza de Robótica en la Facultad Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, utilizando Lego Mindstorms NXT 2.0 aportando a los estudiantes en sus investigaciones avanzadas como la programación y control, a través de simuladores compatibles como Simulink de Matlab y Labview de National Instrument.

1.5.2. OBJETIVOS ESPECÍFICOS

- Diagnosticar la situación actual del estado del arte sobre Mindstorms NXT y la enseñanza de robótica.
- Determinar las diferentes aplicaciones que puede tomar el Mindstorms NXT donde muestre la idoneidad de este robot para uso docente para el aprendizaje.

- Realizar la programación en bloques a través del software de simulación desarrollado en Java.
- Realizar la pruebas experimentales de cada uno de los robots que puede adoptar para fines investigativos.
- Proponer talleres de robótica dirigido a bachilleres y estudiantes de las carreras en la Facultad de Educación Técnica para el Desarrollo como un aporte al futuro Club de Robótica.

1.6. IDEA A DEFENDER

La experiencia de utilizar el Lego Mindstorms NXT para diferentes proyectos de experimentación en el campo de la robótica, permitirá a los estudiantes de las carrera de la FETD formar una idea de cómo medir un proceso simple, o decidir sobre una estructura para soportar un peso específico sin tener que construir modelos a escala, también contribuirá a los docentes a explicar algoritmos sin necesidad de un laboratorio.

1.7. METODOLOGÍA

Para alcanzar los objetivos propuestos se desarrolla una metodología, que se presenta a continuación:

1. Revisar las investigaciones ya realizadas por expertos.
2. Construir con el Lego Mindstorms NXT varios robots que trate de simular al máximo los movimientos.
3. Representar los comportamientos básicos elegidos a nivel de software, en módulos programados en diagramas de bloques e implementarlos en el robot.
4. Desarrollar un plan de pruebas para el robot, ponerlo en práctica y registrar los resultados obtenidos. En base a eso y a los resultados esperados realizar las modificaciones necesarias en la implementación o en el robot.

CAPÍTULO 2: ESTADO DEL ARTE DE LA ROBÓTICA Y LEGO MINDSTORMS NXT

Para el presente capítulo se presentará una introducción a los conceptos básicos que hacen referencia el proyecto de investigación y del estado del arte del Lego Mindstorms.

2.1. ROBÓTICA

La robótica es la ciencia encaminada a diseñar y construir aparatos y sistemas capaces de realizar tareas propias de un ser humano.¹ Una característica distinguible de la robótica que es multidisciplinaria por su naturaleza, es decir que para diseñar un robot exitosamente, se deben poner en práctica conocimientos de ingeniería eléctrica, mecánica, industrial y telecomunicaciones.

En 1942 Isaac Asimov² publicó una historia de ciencia ficción denominada **“Runaround”** el cual se utilizó el término robótica, en la cual postula las siguientes leyes:

1. *Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que sufra daño.*
2. *Un robot ha de obedecer las órdenes recibidas por el ser humano, excepto cuando las órdenes vayan contra la primera ley.*
3. *Un robot debe proteger su propia existencia, siempre que tal protección no vaya contra las leyes primera o segunda.*

Asimov desconocía totalmente sobre la fabricación de los robots, ya que en su obra indicaba que los robots tenían *cerebros positrónicos*³. La aparición de las primeras computadoras ganó mayor influencia por las leyes ya mencionadas.

¹ <http://www.roboticspot.com/robotica.php>, consultado el 04/Abril/2011

² Isaac Asimov es un escritor famoso de ciencia ficción y de variedad de libros de ciencia: http://www.asimovonline.com/asimov_home_page.html

³ Artefacto tecnológico ficticio que opera con una unidad central de procesamiento para los robots, y les dota de cierta forma de conciencia.

Asimismo después de varios años apareció la popular trilogía *La Guerra de las Galaxias* (1977), donde la representación de los robots fue amistosa y simpática. En la figura 2.1 se muestra los robots participantes de la trilogía como el C3P0 y R2D2.



Figura 2. 1: Los robots C3P0 y R2D2.

Durante el siglo XXI, y gracias al considerable avance tecnológico, han ido progresivamente apareciendo diversos tipos de sistemas artificiales de apariencia antropomórfica, conocidos con el nombre de robots. Existen muchas clases de robots, pudiendo ser diferenciados de acuerdo a su arquitectura interna, tamaño, materiales con los que están hechos, la forma en que estos materiales se han unido, los actuadores que utilizan (motores y transmisores), los tipos de sistemas sensoriales que poseen, sus sistemas de locomoción, los microprocesadores y Microcontroladores que tengan a bordo [J. Santos, 2004].

Si bien hasta la fecha los robots han permitido una automatización elevada de tareas simples y repetitivas en procesos industriales y otras áreas, la construcción de robots que exhiban un cierto grado de inteligencia humana es todavía un problema abierto.”Un robot considerado inteligente deberá ser una máquina autónoma capaz de extraer selectivamente información.

Según el diccionario de la Real Academia Española⁴, define al robot como una *“máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas”*

Los robots son controlados específicamente para realizar tareas y este control tiene que ver con su estructura mecánica e incluye 3 fases distinguibles entre sí: percepción, procesamiento y acción. Haciendo uso de la Teoría de Control (Señales y Procesamiento de Señales Digitales) y de los sensores que obtienen información de velocidad, posición en latitud, altitud y longitud, etc. Dicha información permite calcular las señales que recibirán los servomotores mismos que permiten el movimiento de la estructura mecánica para realizar rutas, reconocer patrones, evitar obstáculos, entre otros.

2.2. ROBÓTICA Y AUTOMATIZACIÓN EN LA SOCIEDAD

En la actualidad la sociedad se interesa en aspectos tanto teóricos como prácticos de la automatización y la robótica, a través de máquinas inteligentes (inteligencia artificial) y de sistemas que permite ofrecer servicios a los seres humano, a la industria e inclusive para exploración espacial. Los dispositivos inteligentes realizan trabajos peligrosos y tediosos que los seres humanos no lo pueden hacer, obviamente dependiendo de los límites de inteligencia y capacidad de algunas máquinas que pueden ser autónomas, semiautónomas o teleoperadas.

Las máquinas inteligentes tienen aplicaciones en la medicina, defensa militar, exploración espacial, marina (incluye submarinos), procesos industriales, ensamblaje y entretenimiento [IEEE Robotics, 2007]. Los robots han revolucionado activamente en las fábricas de manera global, fácilmente reconocibles, es común observar un brazo mecánico móvil con distintos grados de libertad, mismos que poseen muchas articulaciones impulsadas por servomotores (mecanismos) y controlados por una computadora, para que el

⁴ Ver <http://www.rae.es/>

robot se ubique con precisión en un determinado punto dentro de su radio de acción.

Los robots inicialmente intentaban simular la estructura humana (constituido por cerebro, músculos unidos al sistema nervioso), es decir que cuando deseamos ejecutar una acción, el cerebro programa la secuencia de movimientos a realizar, transmite una señal que ordena a los músculos a través del sistema nervioso. En la figura 2.2 se puede observar los primeros robots industriales y en la figura 2.3 observamos los robots actualizados que toman la forma de un humano, también llamados humanoides.

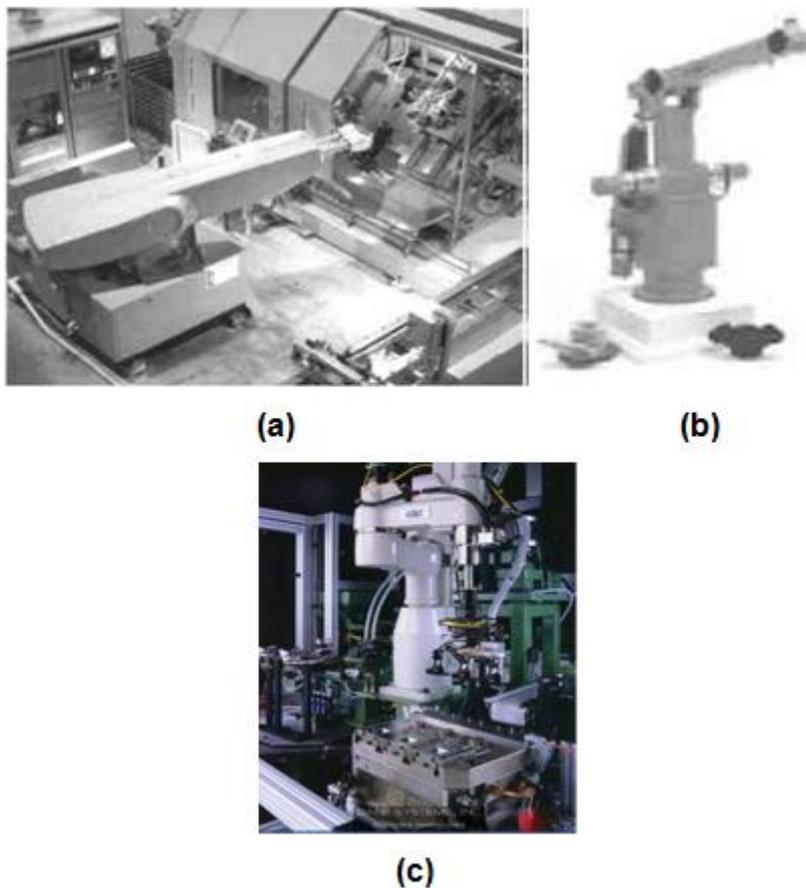


Figura 2. 2: Primeros robots industriales. (a) Devol- Engelberger fundan Unimation (1956). Primer robot industrial, (b) Primer robot con accionamiento eléctrico: IRb6 ASEA (1973); y (c) Robot SCARA del Prof. Makino (Univ. Yamanashi de Japón-1982)
Fuente: http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.1.htm

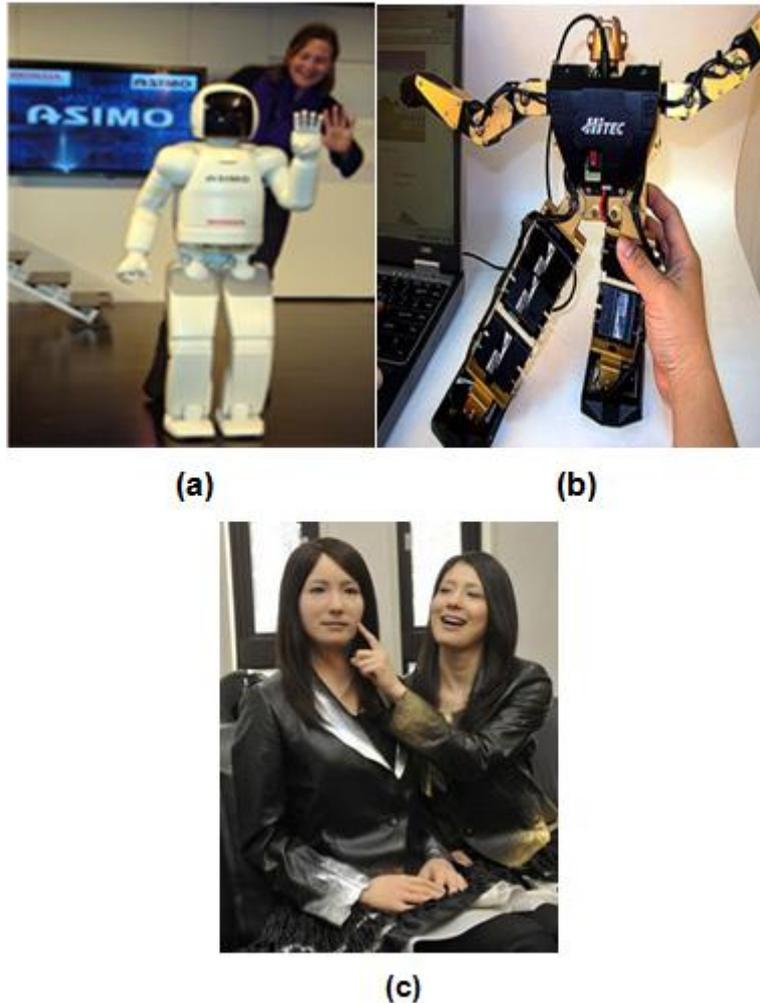


Figura 2. 3: Robots inspirados en los humanos, llamados humanoides:
 (a) ASIMO de Honda, (b) ROBONOVA de Hitec,
 (c) Geminoid-F del Profesor Hiroshi Ishiguro (Univ. Osaka de Japón)

Fuente: <http://asimo.honda.com/gallery/>
http://www.robonova.de/store/download/download/files/ROBONOVA-I-A4_Anleitung.pdf
<http://spectrum.ieee.org/automaton/robotics/humanoids/geminoid-f-takes-the-stage>
<http://www.irc.atr.jp/~ishiguro/>

Asimismo existe diversidad de aplicaciones de la robótica en el ámbito industrial considerados de mayor utilidad en la actualidad. Aunque existen dificultades en la definición formal de lo que es un robot industrial, debido a la diferencia conceptual entre el mercado japonés y el euro-americano en lo que es un robot y un manipulador. Existen diferentes actualizaciones en la definición, la más común posiblemente sea de la RIA⁵ indica que:

⁵ RIA (*Robotic Industry Association*), Asociación de Industrias de Robótica

"Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas"

La cual ha sido ligeramente modificada y adoptada por la ISO⁶ que define al robot industrial como:

"Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas"

Ya la ISO incluye grados de libertad, existe una definición más completa establecida por la AFNOR⁷, que define primero el manipulador y, basándose en dicha definición, el robot:

Manipulador: mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

Robot: manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno.

Por último, la IFR⁸ (*International Federation of Robotics*) distingue entre robot industrial de manipulación y otros robots:

"Por robot industrial de manipulación se entiende una maquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento"

⁶ ISO, Organización Internacional de Estándares

⁷ AFNOR, Asociación Francesa de Normalización

⁸ Federación Internacional de Robótica

Con lo expuesto anteriormente se podría resumir que las aplicaciones robóticas han sido para la fabricación, ensamblaje, soldadura, paletizado entre otras que son manipulados, automáticamente controlados y programables en tres o más ejes, como se muestra en la figura 2.4

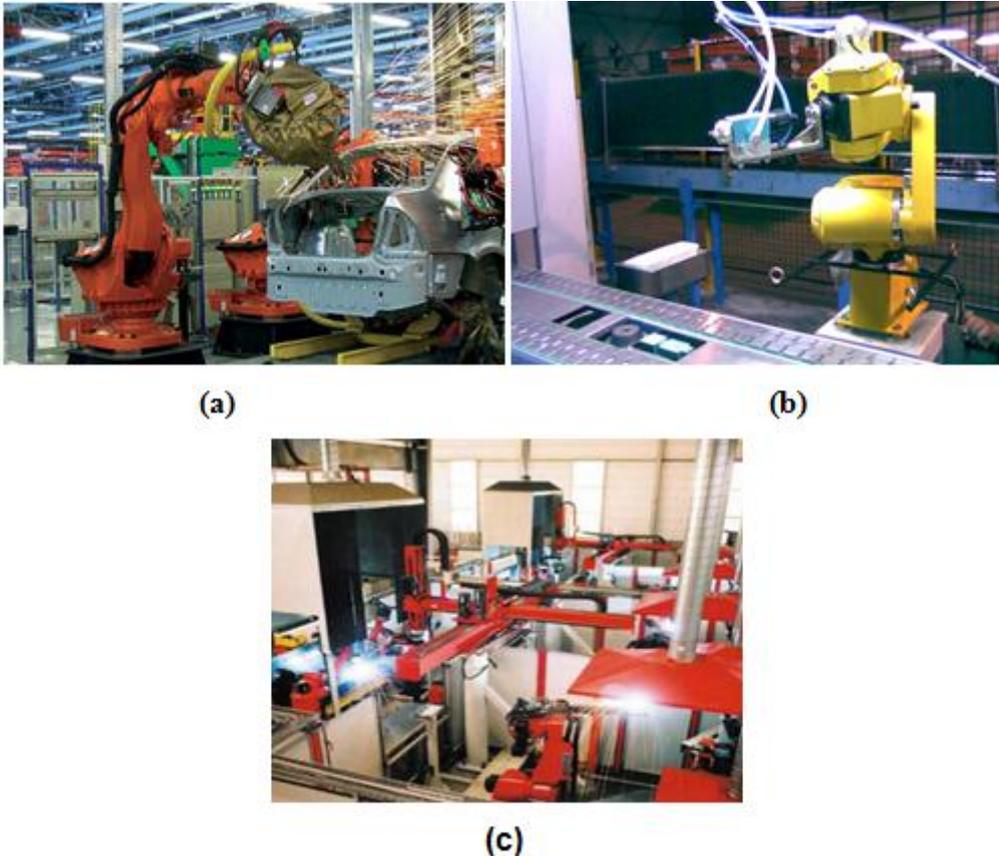


Figura 2. 4: Robots aplicados a la industria: (a) Fabricación y ensamblaje de vehículos (b) Paletizado de discos y (c) Soldadura industrial

Fuente: <http://roboticaysusaplicaciones.blogspot.com/p/generaciones-de-robots.html>
<http://www.ochoamaquinaria.com/productos-robots.htm>
http://www.reisrobotics.com.mx/aplicaciones/welding_systemsolutions.html

Una de las potencias mundiales es Japón quienes cuentan con experiencia en la fabricación y despliegue de la robótica inteligente, que la clasifican en *robots industriales*, aplicados a la industria en trabajo destinados a la manipulación de piezas y herramientas; *robots médicos*, aplicados para operaciones complicadas e inclusive prótesis de funcionamiento autónomo (ver Figura 2.5); y los *robots móviles* capaces de desplazarse en terrenos no propicios para el hombre.

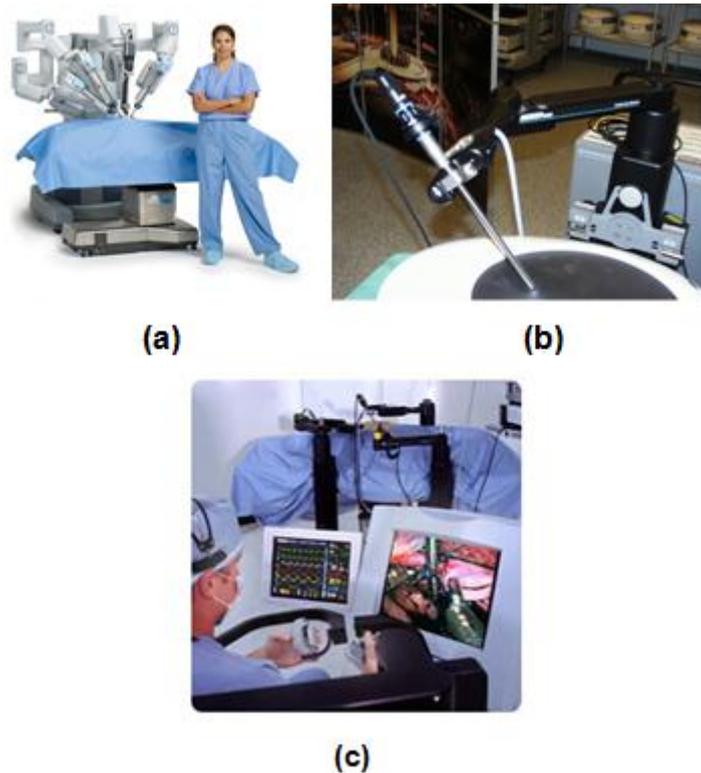


Figura 2. 5: Robots aplicados en la medicina:
 (a) Da Vinci (b) AESOP y (c) Zeus

Fuente: http://www.intuitivesurgical.com/products/davinci_surgical_system/

2.3. ROBOTS MÓVILES

Desde los inicios de la robótica móvil los seres humanos han desarrollado distintos tipos de robots como es el caso de los robots móviles, con capacidad de movimiento autónomo que se desplazan en un área y sin localización física, a diferencia de los robots industriales que localizan en puntos fijos.

Estos robots (ver figura 2.6) se pueden mover por tierra, aire, bajo el agua o incluso en el espacio exterior, utilizando su propia energía. Gracias a sus sensores y su programación, son capaces de guiarse por el entorno autónomamente, pero sin embargo, no es de extrañar que su funcionamiento esté monitorizado por seres humanos o incluso teledirigido. En este último caso, el robot mantiene una comunicación constante con el operador, ya sea por cable o inalámbricamente. Su morfología es variable dependiendo del entorno y de la tarea a realizar.

Es importante destacar que los robots móviles en la actualidad son casos de estudio e investigaciones en casi todas las universidades alrededor del planeta tierra, debido a los factores que intervienen en el diseño de un robot móvil.



Figura 2. 6: Robots móviles (a) explorador terrestre, y (b) explorador marino

Fuente: <http://www.blogger.com/feeds/8278729939162470944/posts/default>
<http://www.techmez.com/2009/05/07/disenan-robot-submarino-para-explorar-el-fondo-del-oceano/>

El instituto francés Ifremer⁹ ha desarrollado diferentes robots para tareas submarinas, como por ejemplo el robot SAR (Système Acoustique Remorqué - Towed Acoustic System), el mismo que tiene un conjunto de sensores geofísicos de alta resolución para el estudio de la estructura de los fondos marinos, y es capaz de sumergirse a 6.000 metros de profundidad.

2.4. ROBOT LEGO MINDSTORM NXT

En el presente proyecto de graduación de Ingeniero en Telecomunicaciones se analiza el potencial de los robots LEGO MINDSTORMS NXT proponiendo casos de estudio que permitan utilizar las metodologías anteriores en la docencia de los estudios de Ingeniería en Telecomunicaciones y de Electrónica en Control y Automatismo, en la Facultad Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil.

⁹ <http://www.ifremer.fr>, Institut français de recherche pour l'exploitation de la mer. Consultada 7 abril 2011.

2.4.1. INTRODUCCIÓN A LEGO MINDSTORMS.

Con la implantación de la nueva LOES¹⁰ las IES¹¹ debe adoptar nuevas metodologías que consigan una mayor implicación de los alumnos en el proceso de aprendizaje. En este escenario, ciertas metodologías conocidas bajo el nombre de *Problem Based Learning* y *Project Based Learning* están ganando mayor aceptación. Básicamente, estas metodologías requieren la formulación de un problema, cercano a los alumnos de forma que al resolverlo éstos adquieran las competencias propias de las materias a estudiar.

Sin embargo, en entornos científicos y tecnológicos normalmente estas metodologías requieren el uso de equipamientos que normalmente resultan caros. LEGO ha lanzado al mercado su última generación de robots en bloques para la construcción de los robots llamados LEGO Mindstorms NXT¹². El LEGO Mindstorms NXT 2.0 permite construir diversidad de modelos de robots que puedan interactuar con los seres humano de manera autónoma, el mismo que es programado a través de varios software (en el capítulo 3 se describirán los tipos de software y el apropiado para la presente tesis) de computadora.

A diferencia de otros robots el LEGO le permite utilizar su imaginación para la construcción de diferentes robots para determinadas aplicaciones, no como otros que vienen definidas sus funciones. LEGO Mindstorms es tanto un juguete como una excelente herramienta de ingeniería y existen muchos ejemplos de ingenieros profesionales alrededor del mundo usando esta plataforma para desarrollar prototipos de robots. Reúne todos los elementos necesarios para el entretenimiento de un niño, así como para la exploración por parte de expertos que quieren poner en práctica conceptos matemáticos, físicos, de mecánica y computación [Bagnall, 2007].

¹⁰ LOES, Ley Orgánica de Educación Superior, según Oficio No. T. 4454-SNJ-10-1512 Quito, 6 de octubre de 2010.

¹¹ IES, Instituciones de Educación Superior

¹² LEGO Mindstorms NXT: <http://mindstorms.lego.com/en-US/>

2.4.2. Por qué escoger el robot LEGO Mindstorms

Porque en la actualidad los alumnos de Ingeniería en Telecomunicaciones y Electrónica en Control y Automatismo de la FETD en la UCSG requieren la adquisición de sólidas competencias en robótica, que incluyen ciertas materias como son la programación en lenguajes de alto nivel, el diseño de sistemas empotrados (Microprocesadores y Microcontroladores) que interaccionan con el medio o las comunicaciones normalmente se adquieren en el laboratorio a base de programar ejercicios sencillos sobre PCs en algún lenguaje de programación de alto nivel (C, Java, Simulink, Matlab, ...) de forma que o bien se simulan los sistemas finales (sensores, actuadores, tarjetas de adquisición, etc.) o bien se utilizan unos equipos normalmente caros y frecuentemente complejos de utilizar.

A pesar de que inicialmente se propuso como un juego de robótica, la plataforma LEGO MINDSTORMS ha ido adquiriendo un interés creciente en entornos educativos a diferentes niveles. Por ejemplo, Butler y Martin **[Butler, 2001]** analizan esta plataforma para impulsar el desarrollo de habilidades tecnológicas entre los alumnos y profesores de educación primaria de Irlanda usando técnicas de aprendizaje basado en proyectos.

Por su parte, Atmatzidou y otros **[Atmatzidou, 2008]** investigan la efectividad de usar LEGO Mindstorms como herramienta para introducir a los alumnos de primaria y secundaria los conceptos básicos de programación a través de actividades lúdicas. También en enseñanza secundaria, Moundridou y Kalinoglou **[Moundridou, 2008]**, describen un estudio empírico relativo al uso de la plataforma Lego Mindstorms en enseñanza secundaria en el campo de ingeniería mecánica.

Sin embargo, es en entornos universitarios donde esta plataforma está ganando más aceptación debido a que proporciona una plataforma potente y de coste reducido para ilustrar conocimientos de diferentes áreas de conocimiento.

Sólo por mencionar algunos autores, Gómez de Gabriel y otros **[Gómez, 2010]**, describen cómo utilizan la plataforma LEGO para desarrollar competencias en mecatrónica en la enseñanza universitaria de tercer nivel (pregrado). Otros autores la usan para desarrollar competencias en programación en diferentes lenguajes de programación, tanto a nivel básico **[Gandy, 2010]** como avanzado **[Lew, 2010]**.

2.5. OTRAS PLATAFORMAS (ROBOTS) SIMILARES A LEGO MINDSTORM NXT

LEGO no es el único fabricante de robots que comercializa plataformas de desarrollo de robots. Existen diversas compañías que ofertan plataformas similares a LEGO MINDSTORMS NXT. A continuación se analizan algunas de estas plataformas:

2.5.1. PLATAFORMA FISCHERTECHNIK¹³

Con esta plataforma (ver figura 2.7) es posible construir modelos robustos y realistas de máquinas, robots, etc. gracias a sus excelentes piezas y al microcontrolador que tiene incorporado. Son dirigido a usuarios que deseen construir modelos a escalas para verificar el funcionamiento de maquinas, aunque también las personas pueden utilizarlo para realizar robots. La programación de la plataforma Fischertechnik se realiza con el software ROBO Pro-Software que es proporcionado por el fabricante al comprar el producto.

La mencionada plataforma cuenta con un microcontrolador de 16 bits, con entradas y salidas USB y puerto serie y conexiones para sensores y actuadores. Se trata de una excelente plataforma que sin embargo tiene un coste elevado y presenta cierta complejidad de uso.

¹³ FISCHERTECHNIK, <http://www.fischertechnik.de/en/>

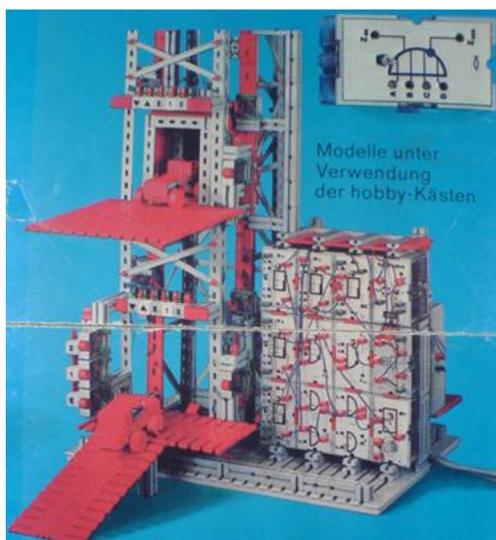


Figura 2. 7: Plataforma creada con Fischertechnik

Fuente: <http://www.fischertechnik.de/en/>

2.5.2. PLATAFORMA ARDUINO¹⁴

Esta plataforma (hardware, ver figura 2.8) basada en una placa muy sencilla que tienen entradas y salidas. Su entorno de desarrollo es el lenguaje de programación *Processing/Wiring*. La plataforma Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado al software del PC. Es posible adquirir la placa y los componentes por separado o también ensamblados, para lo cual LEGO Mindstorms tiene muchos beneficios a lado de Arduino.

Dicha placa es posible añadir sensores y actuadores gracias a las entradas y salidas que dispone (del tipo RS232, USB, I2C, Bluetooth, etc.). El microcontrolador integrado a la plataforma es de 8 bits. El entorno de desarrollo integrado (para la respectiva programación) se puede descargar gratuitamente desde la página web de Arduino.

Está orientado para persona que recién se inician en el mundo de la electrónica y la programación, pero el costo es muy elevado en función de los componentes adquiridos.

¹⁴ ARDUINO, <http://www.arduino.cc/>

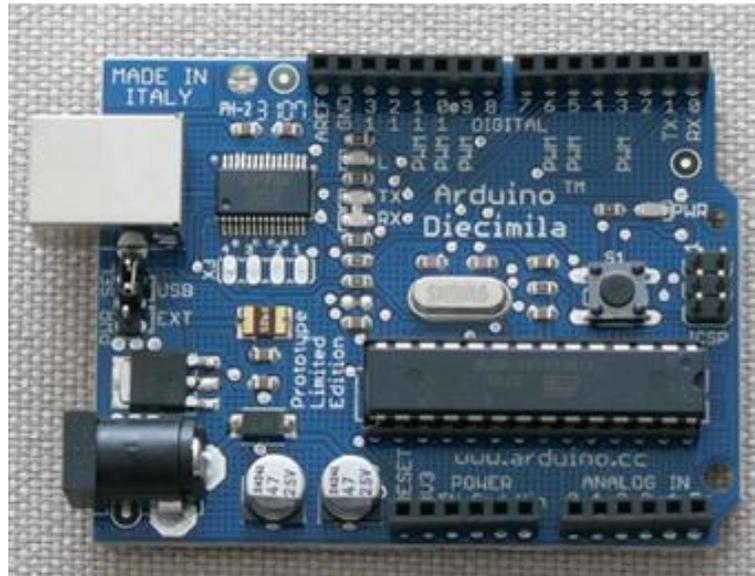


Figura 2. 8: Plataforma tipo placa de Arduino
Fuente: <http://www.arduino.cc/>

2.5.3. PLATAFORMA PARALLAX¹⁵

Se puede destacar que la plataforma PARALLAX cuenta con un amplio abanico de productos relacionados con la robótica. Uno de similares características funcionales LEGO Mindstorms es el “Home BOE-Bot”, proporcionan todos los componentes necesarios para construir un completo microbot móvil sin necesidad de contar con grandes conocimientos técnicos sobre Microcontroladores.

El microbot “Home BOE-Bot” consiste fundamentalmente en una estructura motriz que se gobierna mediante la tarjeta “Home Work” de Parallax, la cual incluye un potente microcontrolador “BasicStamp 2”, como se puede observar en la figura 2.9.

¹⁵ PARALLAX, <http://www.parallax.com/>

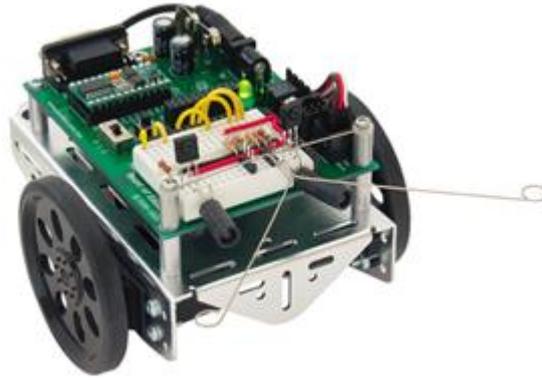


Figura 2. 9: Plataforma tipo móvil de PARALLAX
Fuente: <http://www.parallax.com/>

Una de las ventajas de este microbot radica en su reducido precio. También posee un excelente tutorial y bibliografía que facilitan su programación. Su programación se realiza con “PBASIC”, un lenguaje de alto nivel basado en el lenguaje "Basic".

La desventaja es que se debe adquirir piezas o materiales para construir otro microbot a diferencia de LEGO Mindstorms que dependiendo de la imaginación pueden construir diversidad de aplicaciones y su programación también es de alto nivel como ya se menciono anteriormente.

2.5.4. SR1¹⁶

Es un robot multifuncional dirigido a aquellos entusiastas y aficionados a la robótica que quieran aprender y profundizar en la construcción de robots móviles. Se trata de un kit de sencillo montaje que permite realizar complejos de diferente complejidad. El kit incluye un chasis sobre el que se montan la placa de control, los sensores y el sistema motriz (consistente en dos ruedas con tracción y otra libre).

¹⁶ SR1, <http://www.superrobotica.com/S300010.htm>



Figura 2. 10: Plataforma del Robot móvil SR1
Fuente: <http://www.superrobotica.com/S300010.htm>

2.5.5. PLATAFORMA ROBOTIS BIOLOID

La plataforma robótica BIOLOID (ver figura 2.11) tiene ciertas características similares a LEGO Mindstorms NXT, la cual es construida con tecnología inteligente servo-controlada, compuesto por un potente microcontrolador, servomotores, sensores de proximidad y luminosidad frontal y lateral, un micrófono y un pequeño altavoz.



Figura 2. 11: Plataforma del Robotis Bioloid
Fuente: http://www.robotis.com/xe/bioloid_en

El potente microcontrolador contiene un microprocesador de 16 MHz con 128 KBytes de memoria flash, con puerto serie RS-232, comunicación inalámbrica y botones para su control. El microcontrolador es programable utilizando lenguaje C mediante el compilador GNU GCC WinAVR. Los actuadores tienen un microcontrolador capaz de procesar 50 instrucciones (comandos), los mismos permiten fijar o leer parámetros de su comportamiento (puede medir la posición actual, la corriente consumida, o la variación de la temperatura aplicada en el mismo), lo que permite controlar la retroalimentación de cada articulación del robot.

Esto tiene aplicaciones por ejemplo en los robots bípedos, ya que sin necesidad de inclinómetros o acelerómetros, se pueden conseguir efectos de equilibrio.

2.5.6. PLATAFORMA ROBOBUILDER

Basado en bloques reutilizables, que se caracterizan por ser fáciles de conectar y ensamblar mecánicamente entre sí con múltiples configuraciones y orientaciones, como se puede observar en la figura 2.12. Esta plataforma viene con funcionalidades avanzadas que no contienen otras (excepto la Plataforma Lego Mindstorms) como son: controles PID, detección de objetos, detección de sonido, control remoto por infrarrojos, y comunicaciones Bluetooth y vía serie.



Figura 2. 12: Plataforma del Robot RoboBuilder
Fuente: <http://www.robobuilder.net/eng/>

Los actuadores de la plataforma proporcionan realimentación de velocidad, posición y par. Cada actuador se puede programar con los límites de estas variables, además, los PID utilizados son totalmente parametrizables. Con un simple comando, los actuadores se pueden conmutar entre control angular (servo) al modo de rotación continua (motor DC), permitiendo ensamblar articulaciones y ruedas con los mismos módulos actuadores.

La programación se puede realizar mediante el software proporcionado por RoboBuilder y en programación en C. Este último es el más indicado para los usuarios más avanzados.

2.6. CARACTERISTICAS Y COMPONENTES DE LA PLATAFORMA LEGO MINDSTORMS NXT.

En la figura 2.13 se ilustra el LEGO MINDSTORMS NXT que está compuesto por un controlador, cuatro sensores (ultrasonidos, contacto, micrófono y luz), tres servomotores y un conjunto de piezas para crear estructuras mecánicas compatibles con las de otros productos LEGO.



Figura 2. 13: Partes principales del LEGO Mindstorms NXT que son: [1] Microcontrolador NXT, [2] Sensor de contacto, [3] Sensor de sonido, [4] Sensor de luz, [5] Sensor de ultrasonido y [6] Servomotores.

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

Las partes principales detalladas en la figura 2.13 se describen brevemente.

2.6.1 MICROCONTROLADOR NXT.

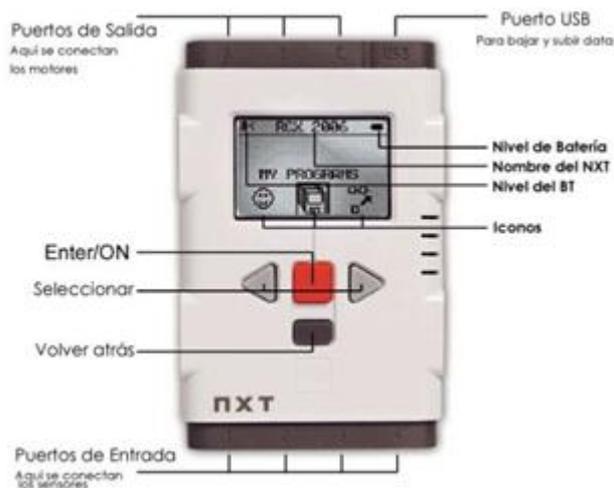
El microcontrolador NXT (ver la figura 2.14) es el controlador inteligente de los bloques LEGO y el cerebro del robot Mindstorms, en él se guardará todos los programas que permitirán construir modelos de robots que ejecutarán tareas predeterminadas por los estudiantes de ingeniería.



Figura 2. 14: Microcontrolador NXT

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

Este bloque inteligente del Sistema de Invención de Robótica que se detalla gráficamente en la figura 2.15, mucho más refinado que el anterior Mindstorms RCX, ya que contiene un procesador ARM de 32 bits de Atmel de 48 MHz, con acceso directo a memoria RAM de 64 KB y memoria flash de 256 KB. Asimismo viene adicionalmente incorporado un procesador AVR de 8 bits a 8 MHz para el funcionamiento de los servomotores y de los sensores de rotación con acceso a memoria flash de 4 KB y memoria RAM de 512 bytes.



Especificaciones Técnicas

- Microcontrolador de 32-bit ARM7
- 256 Kbytes FLASH, 64 Kbytes RAM
- Microcontrolador 8-bit AVR
- 4 Kbytes FLASH, 512 Byte RAM
- Comunicación inalámbrica con Bluetooth (Bluetooth Class II V2.0 compliant)
- Puerto USB (12 Mbit/s)
- 4 puertos de entrada
- 3 puertos de salida
- Pantalla LCD de 100 x 64 pixel LCD
- Cornetas - 8 kHz sound quality.
- Fuente de Poder: 6 Baterías AA

Figura 2. 15: Especificaciones técnicas del bloque Mindstorms NXT

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

2.6.2 MOTORES DEL NXT.

A diferencia de otros motores los del NXT tienen incorporados un tacómetro que transmite información de los ejes de la dirección para que el robot tenga la habilidad de propiocepción¹⁷, con lo que obtiene y salva la información de la posición de sus partes y permite controlar los movimientos de manera precisa, permitiendo el diseño de modelos para poder navegar y guiar brazos robóticos.

El sistema contiene tres servomotores que a diferencia del modelo original RIS son poco lentos, pero al parecer esta desventaja se compensa con la velocidad producida por la cantidad de energía que generan. El sensor de rotación sirve para medir las rotaciones del motor en grados o completar una rotación (equivalente a 360°).



Figura 2. 16: Motor del NXT

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

¹⁷ Propiocepción, es el sentido que informa al organismo de la posición de los músculos.

La velocidad máxima que puede alcanzar este servomotor es de 170 rpm (comparado con los 360 rpm del modelo anterior), y generan hasta 60 mA (3,5 mA en el modelo RIS) de corriente proporcionando un mejor y mayor torque. La disminución de la velocidad en el presente modelo se debe a la cantidad de engranes (ver figura 2.17) dentro del servomotor que genera el torque extra.

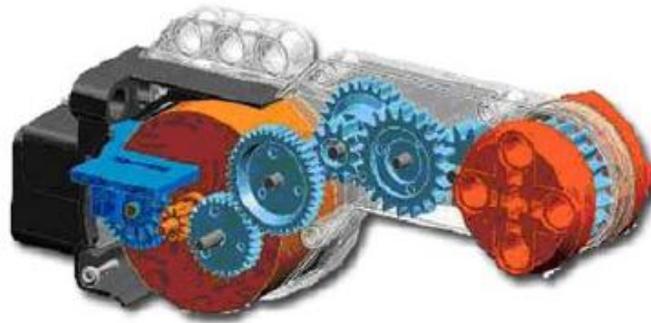


Figura 2. 17: Engranajes del servomotor NXT.
Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

2.6.3 SENSORES DEL NXT

Cuatro tipos de sensores son con los que cuenta el LEGO Mindstorms NXT para poder sensor de acuerdo a condiciones del ambiente y de ahí determinar futuras acciones que debe ejecutar el robot. A continuación se detallarán los 4 sensores:

- **Sensor de contacto:** es uno de los sensores más básico el cual permite al robot tener la capacidad del tacto. Como se ilustra en la figura 2.18, el sensor detecta cuando es presionado y liberado instantáneamente.



Figura 2. 18: Sensor de contacto.
Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

- **Sensor de sonido:** es uno de los sensores que permite detectar el sonido (presión) en decibeles¹⁸ (dB y dBA) como se puede observar en la figura 2.19.



Figura 2. 19: Sensor de sonido.

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

dBA: decibelios ajustados que durante la detección el sensor es adaptada a la sensibilidad del oído humano.

dB: decibeles estándares que durante la detección todos los sonidos se miden con la misma sensibilidad, es decir aquellos sonidos muy altos o muy bajos por el oído humano.

- **Sensor de luz:** es uno de los sensores que permite simular la visión del robot para distinguir entre claro y oscuro, midiendo la intensidad de la luz a través de un lente localizado frente del sensor (ver la figura 2.20). Este sensor cuenta con un LED (color rojo) que es capaz de detectar luz invisible al ojo humano como la luz infrarroja emitida por un control remoto de un equipo de sonido.



Figura 2. 20: Sensor de luz.

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

¹⁸ Un decibel es la medida de la presión del sonido.

- **Sensor ultrasónico:** es uno de los sensores que permite al robot ver y detectar objetos. En la figura 2.21 se muestra el sensor que es capaz de evitar obstáculos, medir distancias y detectar movimiento.



Figura 2. 21: Sensor ultrasónico.

Fuente: <http://mindstorms.lego.com/en-us/Default.aspx>

El funcionamiento de este sensor es muy similar al que se utilizan en otros muchos sistemas de medida de distancia por ultrasonidos: mide el tiempo que tarda una señal ultrasónica en ir y volver. Si el tiempo es pequeño, el objeto está cerca; si el tiempo es grande, el objeto está lejos y si el tiempo es mayor que un determinado valor, el objeto está fuera de alcance.

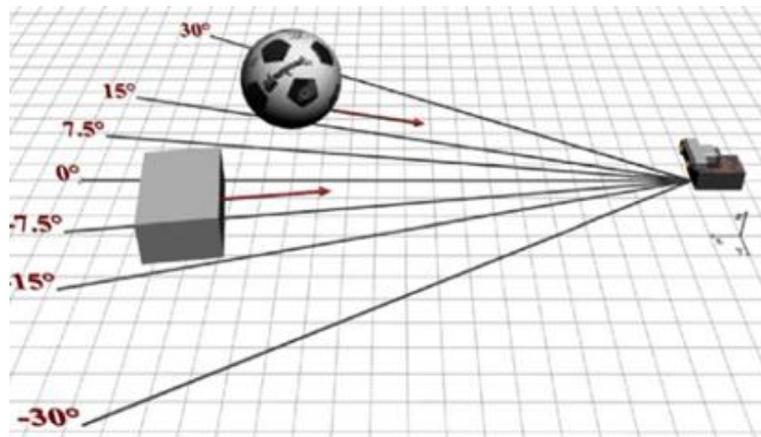


Figura 2. 22: Cobertura del sensor ultrasónico.

Fuente: Manual Lego Mindstorms

En la figura 2.22 se puede observar como el sensor ultrasónico produce un cono sónico para detectar objetos enfrente de él con un ángulo de aproximadamente 30°, por ejemplo si el objeto se encuentra a una distancia de 180 cm, él cono tendría un diámetro de 90 cm.

Adicional a esto el dispositivo es único en el NXT con tecnología I²C¹⁹ el cual permite una exactitud en distancia que van desde los 6 cm hasta 180 cm con un error de ± 3 cm.

2.6.4 COMUNICACIÓN USB Y BLUETOOTH

El NXT se puede comunicar con el computador (fijo o portátil) de dos maneras, a través del cable USB (ver figura 2.23) y por intermedio del bluetooth (ver figura 2.24). La velocidad de transmisión del puerto USB es de 12 Mbps, mientras que la velocidad de la conexión inalámbrica es de 461 Kbps mucho menor que del USB, pero permite trabajar a distancias mayores a las que el cable USB estaría limitado.



Figura 2. 23: Comunicación entre PC y NXT a través del cable USB.
Fuente: Los Autores



Figura 2. 24: Comunicación inalámbrica entre PC y NXT a través del bluetooth.
Fuente: Los Autores

¹⁹ **I²C** es un bus de comunicaciones en serie. Su nombre viene de *Inter-Integrated Circuit* (Circuitos Inter-Integrados). La velocidad es de 100Kbps en el modo estándar, aunque también permite velocidades de 3.4Mbps. Es un bus muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados (*Embedded Systems*) y generalizando más para comunicar circuitos integrados entre sí que normalmente residen en un mismo circuito impreso.

El uso del Bluetooth brinda la posibilidad de crear programas que utilizan más de 256 kb de memoria, que pueden incluir video, mapeo, reconocimiento de voz, ya que al programar se reflejan las acciones entre el NXT y la memoria del computador que accede vía bluetooth.

CAPÍTULO 3: ESTADO DEL ARTE DE LA PROGRAMACIÓN DEL LEGO MINDSTORMS NXT

En el presente capítulo se describirá el estado del arte de las aplicaciones y herramientas de programación del LEGO Mindstorms NXT que han sido desarrolladas por la comunidad de robótica mundial utilizando esta plataforma.

3.1. INTRODUCCIÓN A LAS HERRAMIENTAS DE PROGRAMACIÓN.

En la actualidad existe una gran diversidad de entornos de programación que permiten elaborar, diseñar o construir programas para el funcionamiento de los robots LEGO MINDSTORMS en diferentes lenguajes de programación. Un buen punto de partida para seleccionar el lenguaje y el entorno de una extensiva y actualizada lista para programar (software compatible con Lego Mindstorms) al microcontrolador en cualquier lenguaje de alto nivel, incluyendo algunos de los más extendidos: C, C++, Java, Ada, .NET (C#, Visual Basic), LabVIEW, Python, Ruby, MATLAB, etc.

Adicional a esto, existen lenguajes de programación basados en lenguajes de programación estándar y cuyas características específicas facilitan la programación del microcontrolador. Estos lenguajes de programación predominan los lenguajes NQC, NXC y RobotC, basados en C y NXT-G (lenguaje de programación gráfico basado en LabVIEW).

Asimismo la existencia de la diversidad de lenguajes de programación disponibles, y cuya característica importante es la programación de los robots LEGO. En la mayoría de los casos, se programan los robots LEGO desde entornos que se utilizan a nivel superior (profesional), como Visual Studio .NET, Eclipse, LabVIEW o muchos otros entornos que soporten el compilador y herramientas GCC. La programación con estos entornos de programación se realiza desde un PC (Windows / Linux) donde se utilizan herramientas para crear los programas, cargarlos en la CPU.

Otra característica importante es la portabilidad de plataformas industriales, como OSEK, diseñado para proporcionar una arquitectura estándar de software para las diferentes ECUs del automóvil, a las CPUs LEGO Mindstorms. Esta plataforma, que se conoce con el nombre de nxtOSEK incluye el sistema operativo, una pila de comunicaciones y un protocolo para la administración de redes, proporciona un sistema operativo de tiempo real que gestiona el funcionamiento del dispositivo en el que está implantado. Para ello dispone de estas herramientas: tareas, eventos, alarmas, etc.

3.2. PROGRAMACIÓN NXT.

Profesionales y entusiastas de todas las edades y de todo el mundo, involucrados en compartir ideas, diseños y técnicas de programación asociados a la plataforma LEGO Mindstorms, organizados en grupos afines, de acuerdo al área de trabajo o por diversas aplicaciones que las desarrollan en conjunto, donde sacan al máximo provecho esta tecnología.

Entre los diferentes entornos de programación para el NXT existen los comerciales y los desarrollados por la comunidad (ver figura 3.1). Los de carácter comercial son destinados para educación, y algunos utilizan el mismo firmware²⁰, algunos utilizan el software oficial de LEGO y otros un firmware propio específicamente desarrollado para un determinado lenguaje de programación. Para visualizar la información en la pantalla, el NXT utiliza diferentes caminos, uno de ellos es por medio de los ficheros en formato RIC, que a primera vista tienen formato en mapa de bits, pero que abren muchas posibilidades en programación.

²⁰ El **firmware** es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil (ROM, EEPROM, flash, etc), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación.

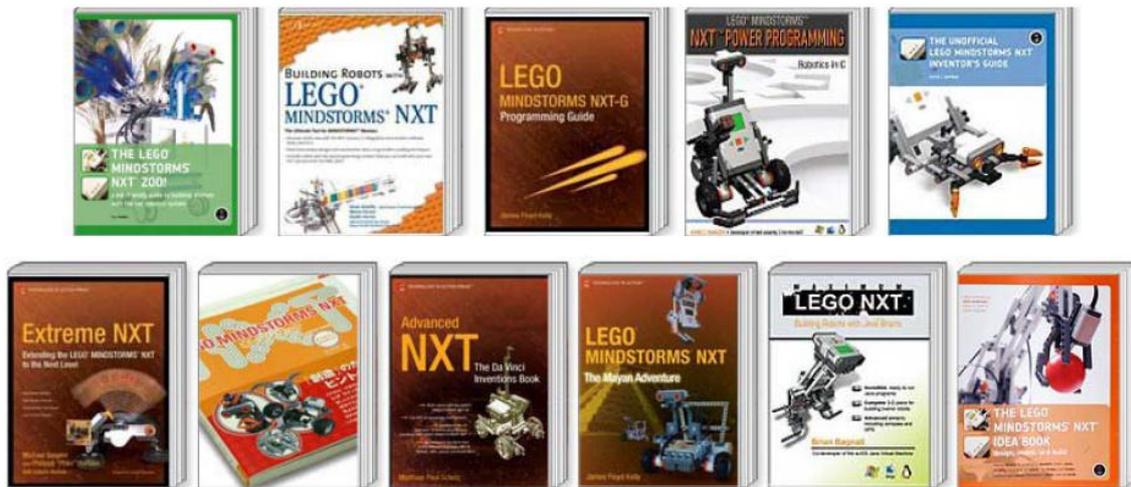


Figura 3. 1: Variedad de textos publicados por profesionales y de la comunidad del LEGO.

Fuentes: Los Autores

3.3. PLATAFORMAS COMERCIALES DE PROGRAMACIÓN DEL NXT.

A continuación se analizan las principales características de algunas plataformas comerciales de programación especialmente relevantes.

3.3.1. SOFTWARE NXT-G EDUCACIÓN

Este software sustituye a Robolab como software propio de LEGO dirigido al sector educativo. Al igual que Robolab está basado en LabVIEW y ha sido desarrollado a partir de la colaboración entre National Instruments y LEGO. Es un software gráfico (al modo de LabVIEW) en el que se programa por medio de secuencias de bloques por medio de arrastrar y soltar. En la web de National Instruments puede verse una demo²¹.

En la web de LEGO Engineer²² se puede ver cuál es la evolución prevista de este software así como una comparativa entre este software, Robolab y la versión para consumo general. La parte de análisis de datos que ofrecía Robolab en su sección de Investigador ha sido eliminada aquí y se ofrecerá como software aparte.

²¹ <http://www.ni.com/academic/mindstorms/>

²² <http://www.legoengineering.com/content/view/25/36/>

Los requerimientos de software de pueden encontrar en el blog de LEGO Educación. Este software viene preparado para utilizar los sensores del RCX. La versión 2.0 del software NXT-G Educación está disponible en castellano, el cual aporta importantes mejoras respecto a la versión 1.0/1.1. Las principales diferencias respecto de la versión comercial son:

- Guía de aprendizaje con proyectos que permiten aprender a programar el robot en NXT-G y el funcionamiento de las diferentes herramientas de las que dispone la versión.
- Interfaz específica para la colección de datos desde los sensores que viene con el kit y su posterior análisis.
- La última versión es la 2.1 la figura 3.2 muestra la pantalla del NXT-G Educación, la cual es posible actualizar a esta versión desde la 2.0 en la web de LEGO Educación.

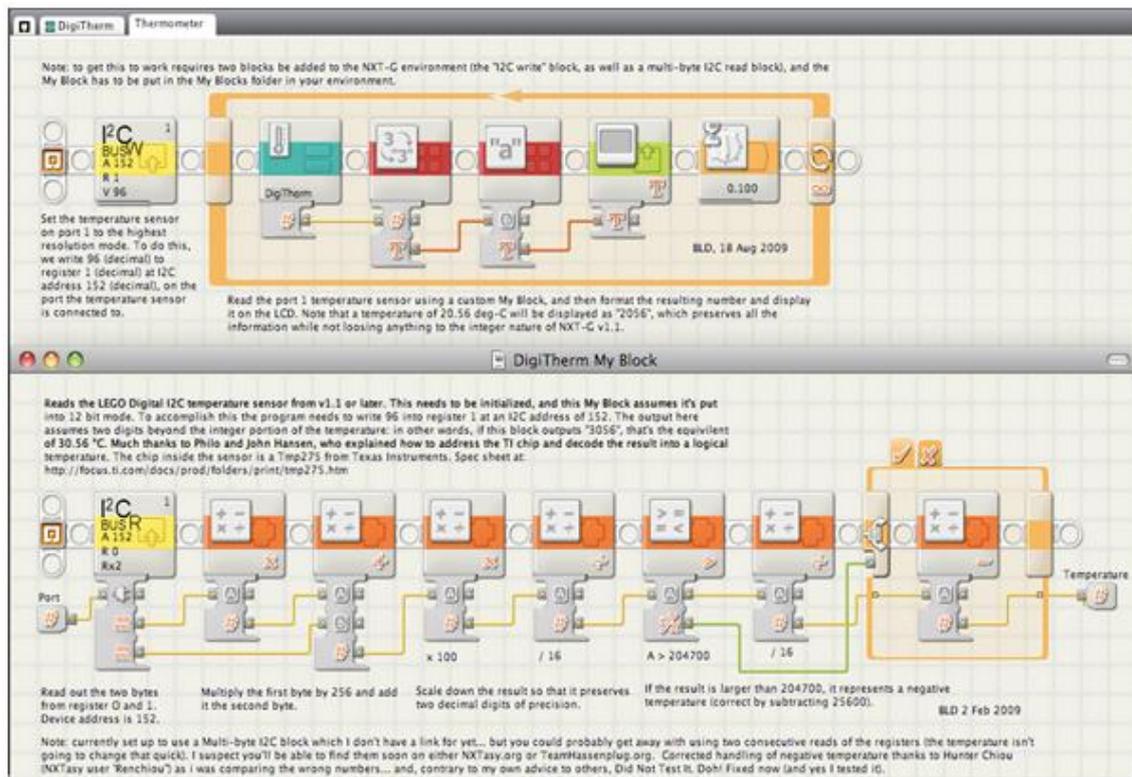


Figura 3. 2: Software de programación NXT-G Education.

Fuente: <http://blog.electricbricks.com/en/2010/08/nxt-g-education-software-2-1/>

3.3.2. SOFTWARE NXT-G

NXT-G versión 1.0/1.1, esta es la versión que lleva el NXT. Si nos fijamos en su funcionalidad es muy similar al NXT-G educación. No ofrece en origen los bloques de programación correspondientes a los sensores y motores del RCX, pero se pueden encontrar en la página de actualizaciones de LEGO²³. En el caso de NXT-G 1.0 para incorporar nuevos bloques de programación es necesario instalar la actualización "*Dynamic Block Update*". La versión 1.1 ya no la requiere.

Este es el software comercial de LEGO MINDSTORMS NXT. Está basado en el motor de LabVIEW de National Instruments, en la página de actualizaciones de LEGO se puede encontrar la última versión del firmware oficial. En la figura 3.3 se puede observar el NXT-G versión 2.0, es la versión que lleva el segundo set comercial que salió a la venta en 2009. Entre otras mejoras cuenta con un editor de imágenes y otro de sonido para poder crear material personalizado que se puede usar en los programas.

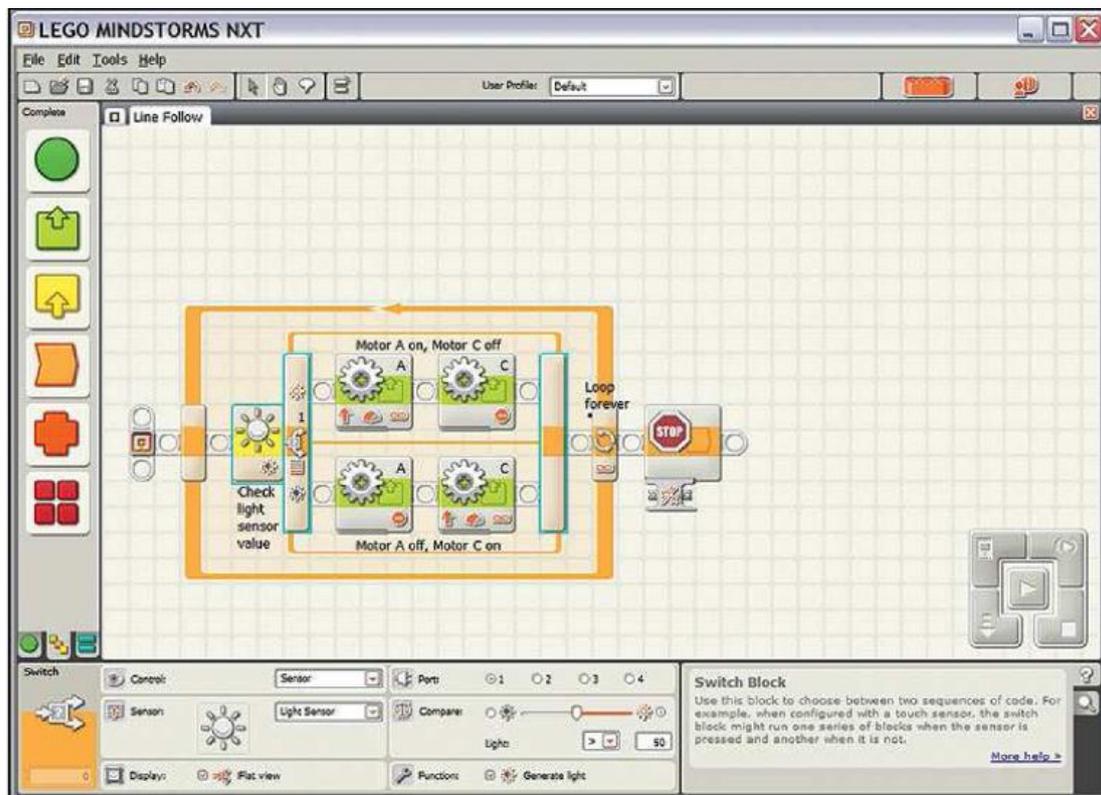


Figura 3. 3: Software de programación NXT-G.
Fuentes: Los Autores

²³ <http://mindstorms.lego.com/support/updates/>

Este lenguaje de alto nivel funciona por medio de bloques con funcionalidades específicas, que se conectan generando rutinas que luego son transferidas al procesador del robot para su posterior ejecución. En la figura 3.3 se muestra el interfaz gráfico del NXT. Esta versión también introduce la coma flotante, por lo que se actualizan todos los bloques que pueden hacer uso de esta función.

Además de los bloques de programación que se encuentran en la interfaz de NXT-G existe un toolkit de NI LabView que permite crear bloques adicionales. Existe un buen número de estos bloques que amplían las funciones de la interfaz y dan acceso a más opciones que puedes descargar en la sección bloques adicionales para NXT-G.

3.3.3. SOFTWARE ROBO LAB

Esta es la versión de Robolab (software de programación gráfico) adaptada al NXT desarrollada como herramienta de transición del software antiguo al nuevo. Permite programar tanto el RCX como el NXT (sólo comunicaciones por USB entre el ordenador y el NXT, no bluetooth). Para el RCX ofrece un nuevo firmware que permite utilizar nuevos tipos de variables (coma flotante), una mayor velocidad, más velocidades para los motores.

En LEGO Engineering se puede encontrar una actualización para Robolab. Aparte de corregir algunos problemas incorpora los bloques de programación para varios sensores de Hitechnic y Mindsensors. CEEO²⁴ ofrece los Robobooks²⁵, una iniciativa dirigida al ámbito educativo, el mismo que requiere de Robolab 2.9 (ver figura 3.4) para su funcionamiento.

²⁴ <http://www.ceeo.tufts.edu/> , página web del Centro de Enseñanza de Ingeniería

²⁵ <http://web.mac.com/chrisbuerginrogers/RoboBooks/Welcome.html> , página web del centro de difusión de ingeniería educacional.



Figura 3. 4: Software de programación Robolab.
Fuente: <http://www.ceco.tufts.edu/>

En realidad Robolab fue creada originalmente por *Tufts University*²⁶ (USA), en la figura 3.5 se muestra el entorno de programación, que posee una interfaz gráfica desarrollada en el sistema LabVIEW de *National Instruments*, considerado como un programa intuitivo que permite hacer operaciones con números enteros y flotante.

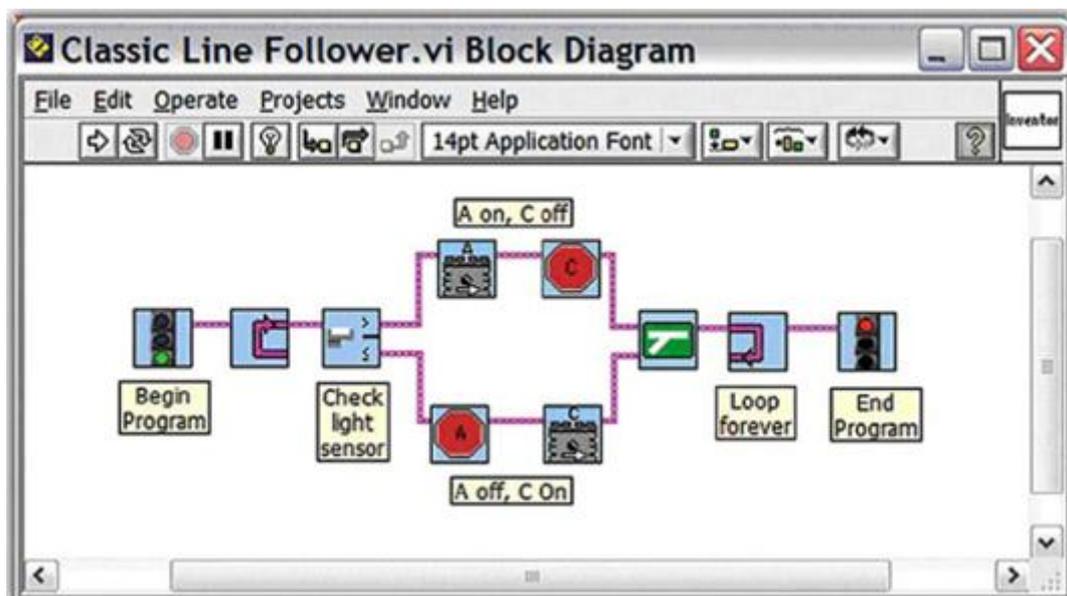


Figura 3. 5: Entorno de la programación en Robolab.
Fuente: <http://www.ceco.tufts.edu/>

²⁶ <http://www.tufts.edu/>

3.3.4. SOFTWARE ROBOTC

RobotC es un software destinado para la educación media y superior, desarrollado por *Robotics Academy*²⁷ (*Carnegie Mellon University*) que permite programar en C y bajo el ambiente Windows para escribir y depurar programas diseñados para robots de la plataforma LEGO. RobotC²⁸ da soporte al bus I²C del NXT, y cuenta con un compilador propio y un depurador en tiempo real. Ha sido en *Carnegie Mellon University* donde se desarrollo el nuevo currículo para LEGO Mindstorms NXT Educación.

RobotC es un software comercial que se vende por \$ 30 en la versión descargable por Internet. RobotC utiliza un firmware propio no compatible con NXT-G, y con mayor rapidez que el original de LEGO. En las figura 3.6 y 3.7 se muestra el entorno de programación RobotC que se presenta como una solución que permite a los estudiantes aprender un lenguaje basado en C, utilizado en aplicaciones complejas tanto educativas como profesionales. Su desventaja principal es que debe reemplazarse el firmware en el bloque de NXT de forma que puedan ser ejecutados los programas.

Hay disponible documentación en formato web²⁹ y en formato pdf, en este último caso, "*Programing with robots*³⁰", se trata de un trabajo de *Albert Schueller Xander Soldaat*, mismo que ha desarrollado drivers para diversos sensores de terceras partes para RobotC con diversos ejemplos³¹. Asimismo existe un libro on-line sobre programación en C denominado "*C-Book*³²".

²⁷ <http://www.education.rec.ri.cmu.edu/>

²⁸ <http://www.robotc.net/>

²⁹ <http://www.robotc.net/education/curriculum/nxt/>

³⁰ <http://carrot.whitman.edu/Robots/notes.pdf>

³¹ <http://rdpartyrobotcdr.sourceforge.net/documentation/index.html>

³² http://publications.gbdirect.co.uk/c_book/

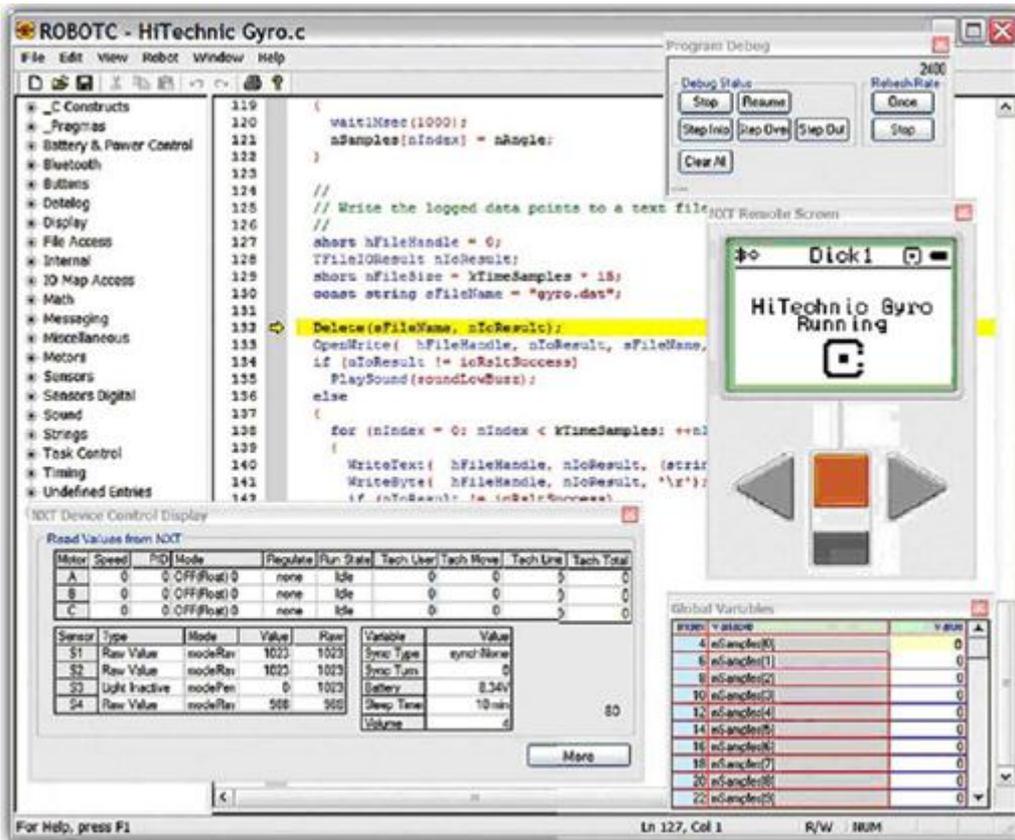


Figura 3. 6: Entorno de programación RobotC.
Fuente: <http://www.robotc.net/education/curriculum/nxt/>

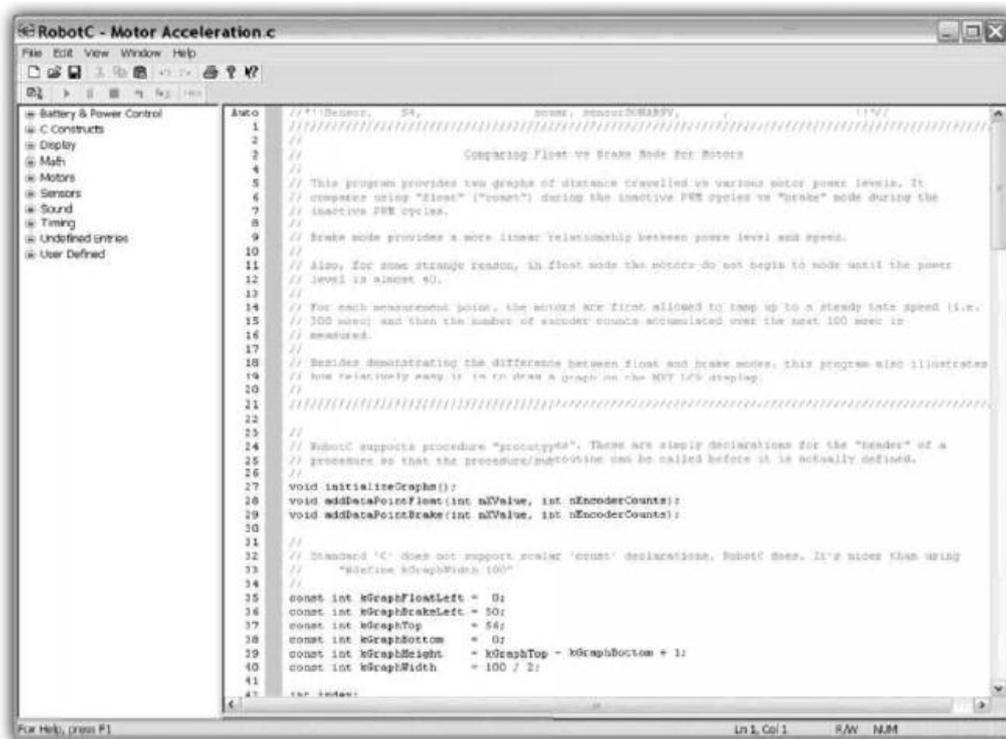


Figura 3. 7: Ejemplo de un programa en el entorno de programación en C.
Fuente: <http://www.robotc.net/education/curriculum/nxt/>

3.4. PLATAFORMAS DE LIBRE USO PARA PROGRAMACIÓN NXT.

A continuación se analizan las principales características de algunas plataformas gratuitas de programación especialmente relevantes.

3.4.1. LABVIEW TOOLKIT

LabVIEW Toolkit permite programar gráficamente, el cual permite programar el NXT directamente desde LabVIEW con el *LabVIEW Toolkit for LEGO Mindstorms NXT*, disponible en la página web: <http://www.ni.com/academic/mindstorms/>. Una característica importante de este software que el uso del mismo es cada vez más extendida en entornos científicos y de ingeniería para realizar tareas de adquisición de datos y control.

A diferencia del resto de software de programación esta permite una programación mucho más avanzada que con NXT-G, con la posibilidad de desarrollar nuevos bloques de programación para NXT-G (ver el entorno de programación en la figura 3.8). En caso de no disponer de Labview, la empresa propietaria, National Instruments (<http://www.ni.com/>), permite la descarga gratuita del software LabVIEW 7.1 versión estudiantil. Existen diversas guías y tutoriales para aprender a programar los Legos [Gasperi³³, 2009].

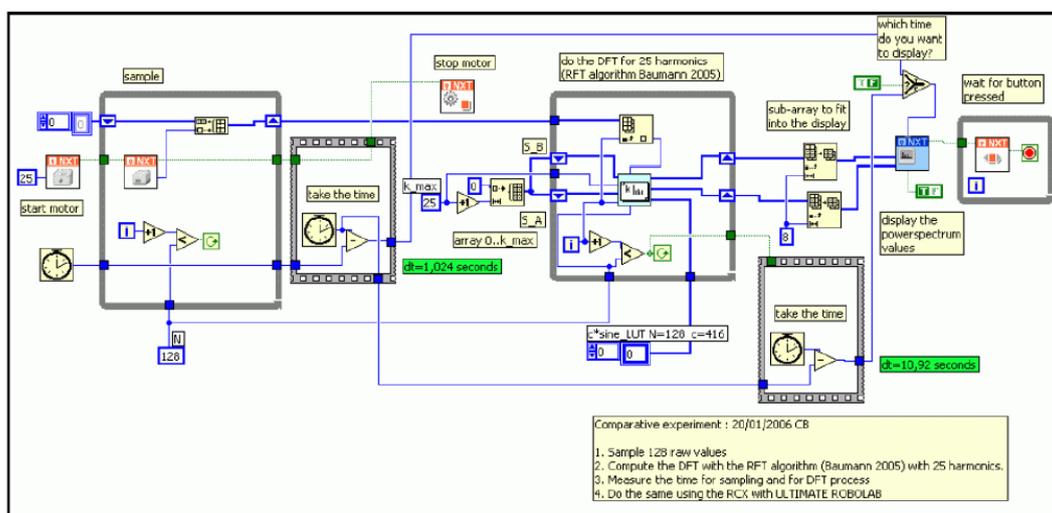


Figura 3. 8: Entorno de programación gráfica a través de LabVIEW.

<http://www.ntspress.com/publications/labview-for-lego-mindstorms-nxt/sample-chapter/>

³³ <http://www.ntspress.com/publications/labview-for-lego-mindstorms-nxt/sample-chapter/>, muestra gratuita de un capítulo del libro escrito por Gasperi.

3.4.2. MATLAB y SIMULINK

El NXT Toolbox para MATLAB³⁴ fue desarrollado para controlar al robot LEGO Mindstorms NXT vía conexión bluetooth. Se trata de una programación en código abierto. En la página web de MathWorks: <http://www.mathworks.com/academia/lego-mindstorms-nxt-software/> puede encontrarse el código necesario para la programación respectiva como por ejemplo realizar un control remoto y otras aplicaciones utilizando MATLAB y Simulink.

Es interesante la entrada en el blog de *MATLAB "LEGO Mindstorms NXT in Teaching"*³⁵ por Loren on the Art of MATLAB". En la figura 3.9 se puede observar el entorno de programación textual de Matlab:

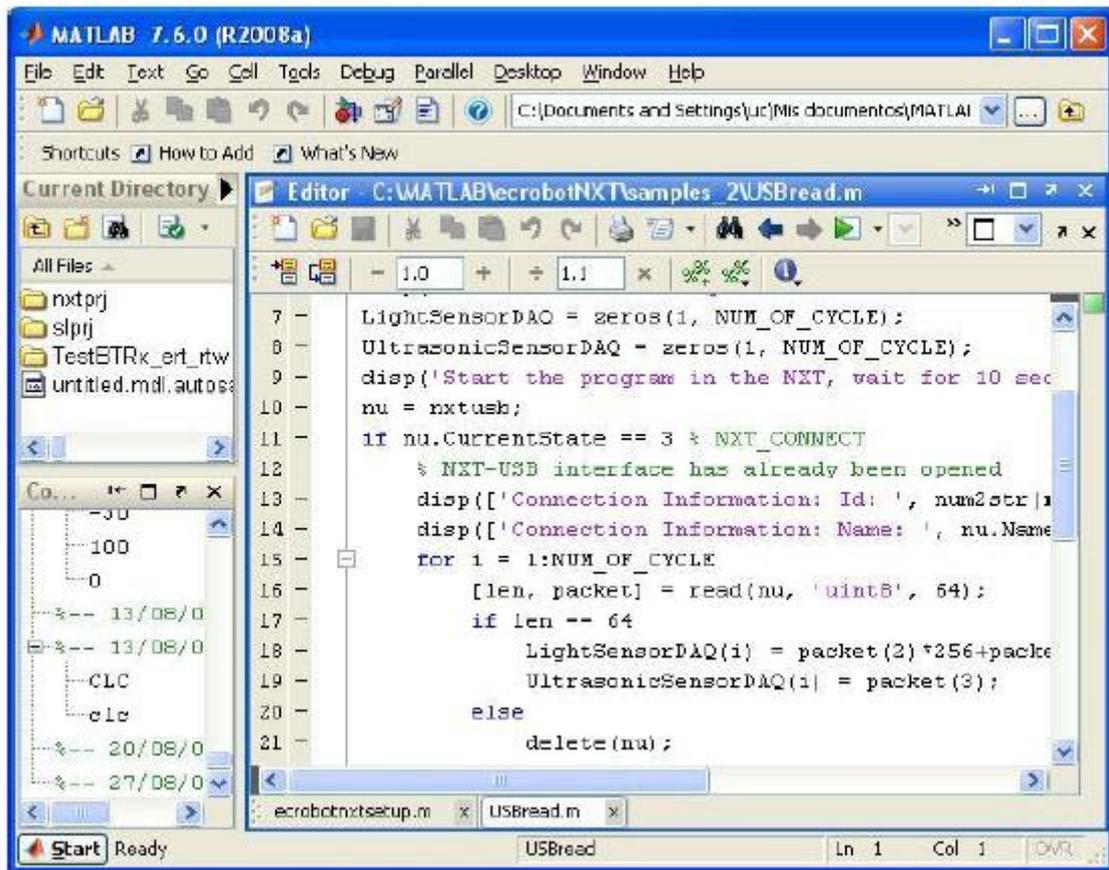


Figura 3. 9: Entorno de programación textual en Matlab para el NXT.

Fuente: <http://mindstorms.lfb.rwth-aachen.de/index.php/en>

³⁴ <http://mindstorms.lfb.rwth-aachen.de/index.php/en>, página del proyecto de ingeniería eléctrica e Información tecnológica.

³⁵ <http://blogs.mathworks.com/loren/2008/06/30/lego-mindstorms-nxt-in-teaching/>

Otra página web interesante para programar en Matlab es *RWTH MINDSTORMS NXT Toolbox*³⁶. Mientras que existen proyectos que utiliza Simulink compatible con el NXT³⁷ en la misma página de MATLAB. También ofrecen de modo libre el código necesario para desarrollar un sistema embebido para control incrustado³⁸.

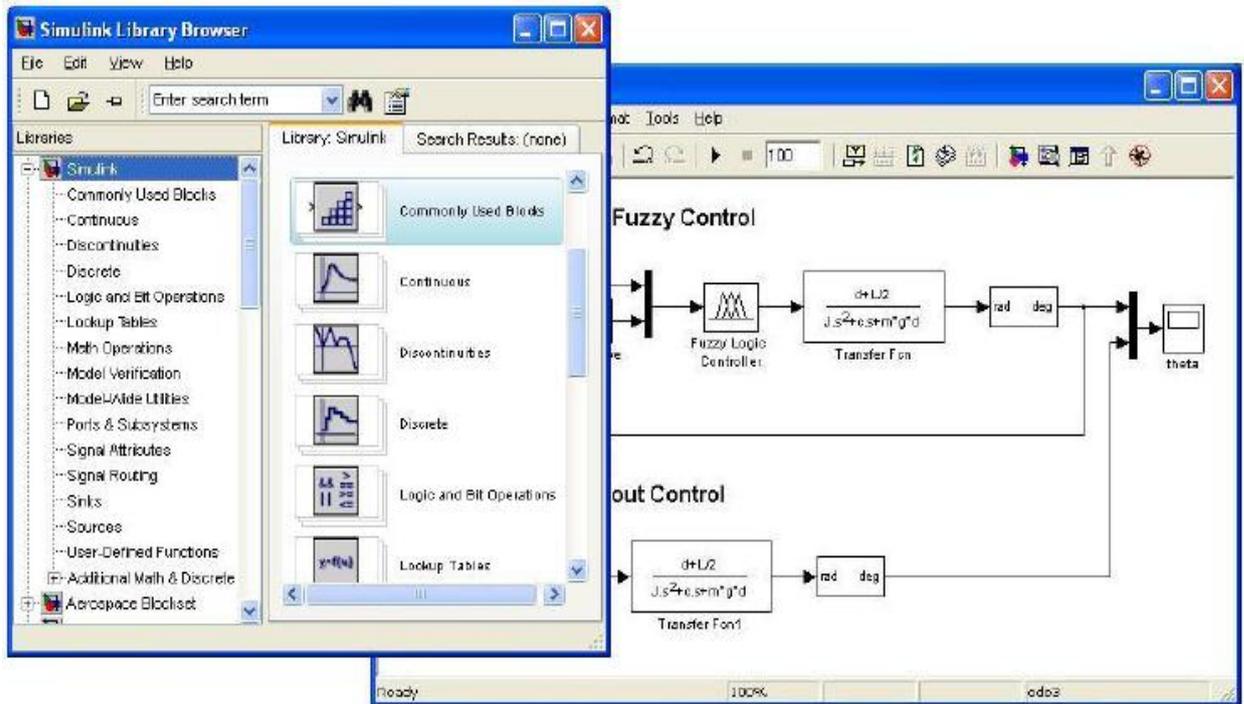


Figura 3. 10: Entorno de programación por diagrama de bloques en Simulink de Matlab.

Fuente: Manual de Matlab-Simulink

Simulink es la plataforma que se utilizó para la realización de la programación del sistema del robot explorador, haciendo uso de su Toolbox específico así como el resto de paquetes de bloques para realizar operaciones conjuntas entre sensores. Para la utilización de Simulink es necesario familiarizarse con la manipulación de bloques y la construcción de modelos. Para esto es necesario también conocer la variedad de bloques existentes. Y por último manejar las herramientas de análisis provistas por Simulink.

³⁶ <http://www.mindstorms.rwth-aachen.de/>

³⁷ <http://www.mathworks.com/matlabcentral/fileexchange/13399>

³⁸ <http://www.mathworks.com/academia/lego-mindstorms-nxt-software/>

3.4.3. BRICXCC

Bricx Command Center³⁹, es un conocido IDE⁴⁰ que soporta programación del RCX con NQC, C, C++, Pascal, Forth, y Java utilizando brickOS, pbForth y leJOS. Con BricxCC se pueden desarrollar programas en NBC y NXC. Tanto NBC como NXC utilizan el firmware estándar del NXT. Este software está disponible en código abierto. John Hansen ofrece diversas utilidades, que poco a poco va incluyendo en BricxCC.

John Hansen ha publicado las herramientas⁴¹ necesarias para utilizarlo en MacOX. Esta plataforma sirve para desarrollar habilidades de programación, que a partir de III Ciclo se enseña a programar en C (Informática III) y en VI Ciclo se programa en PicBasic y PicC en las asignaturas de Microcontroladores y Diseño Electrónico como lenguajes de programación de alto nivel.

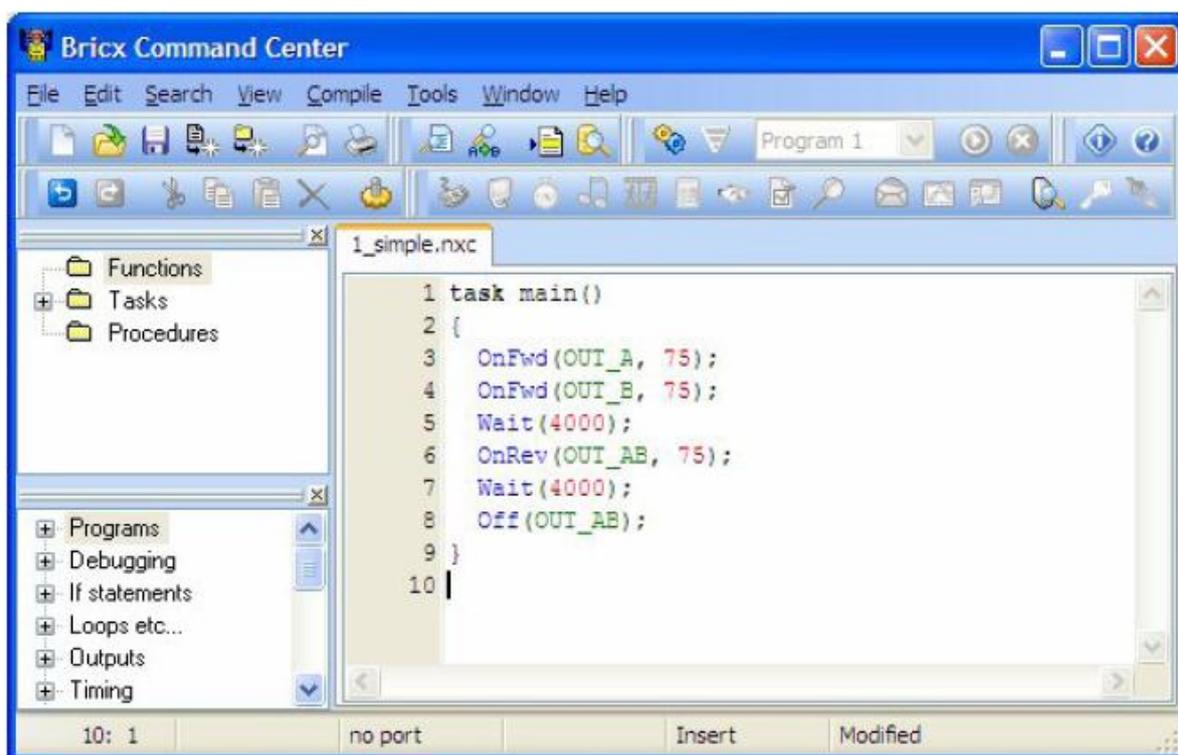


Figura 3. 11: Entorno de programación IDE de la plataforma BricxCC

Fuente: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf

³⁹ <http://bricxcc.sourceforge.net/>

⁴⁰ IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de [interfaz gráfica](#) (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje [Visual Basic](#), por ejemplo, puede ser usado dentro de las aplicaciones de [Microsoft Office](#), lo que hace posible escribir sentencias [Visual Basic](#) en forma de [macros](#) para [Microsoft Word](#).

⁴¹ <http://bricxcc.sourceforge.net/utilities.html>

3.4.4. ENCHANTING

Enchanting⁴² es una herramienta gráfica de programación del NXT basado en Scratch⁴³. Funciona con leJOS NXJ[30] (Java para el NXT). Enchanting gratuito y de código abierto.

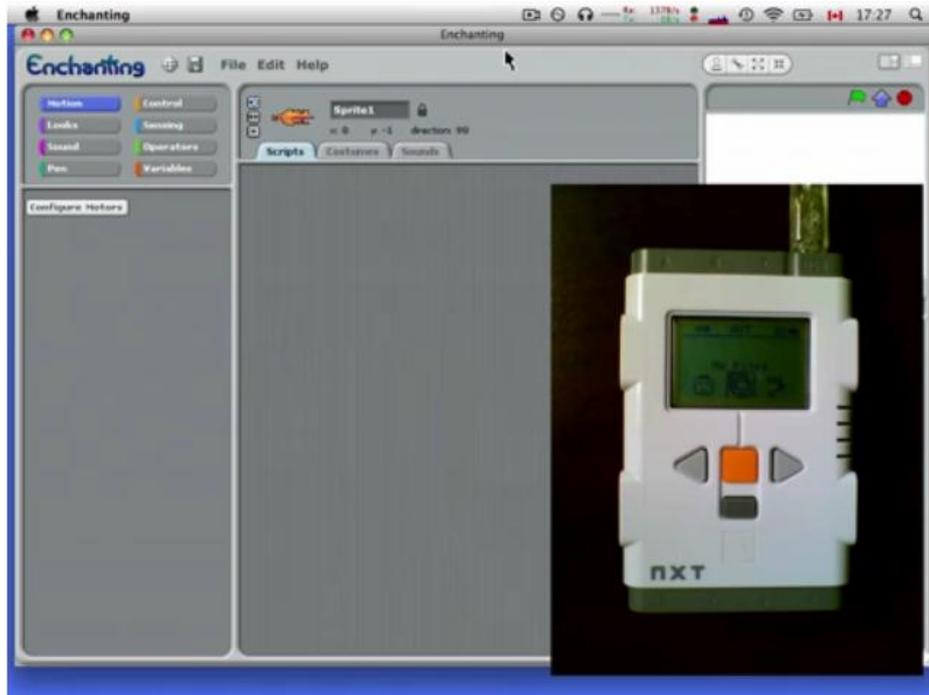


Figura 3. 12: Entorno de programación Scratch de la plataforma Enchanting
Fuente: <http://enchanting.robotclub.ab.ca/tiki-index.php>

3.4.5. NBC

NBC es un lenguaje de programación con una sintaxis muy parecida al ensamblador que nos permite programar el NXT a bajo nivel usando el firmware Standards. Se puede utilizar como editor BricxCC visto anteriormente. Sorosy⁴⁴ ofrece un depurador para la programación con NBC.

En la figura 3.13 se muestra la plataforma NBC, la misma no es fácil de usar pero nos permite escribir programas muy rápidos y eficaces y que además ocupan poco espacio. NBC es totalmente gratuito. Se puede encontrar más información en la página del NBC.

⁴² <http://enchanting.robotclub.ab.ca/tiki-index.php>, consultado 20 abril 2011

⁴³ <http://irobotikas.net/wiki/index.php?title=Scratch>, consultado 20 abril 2011

⁴⁴ <http://www.sorosy.com/lego/nxtdbg/> consultado 20 abril 2011

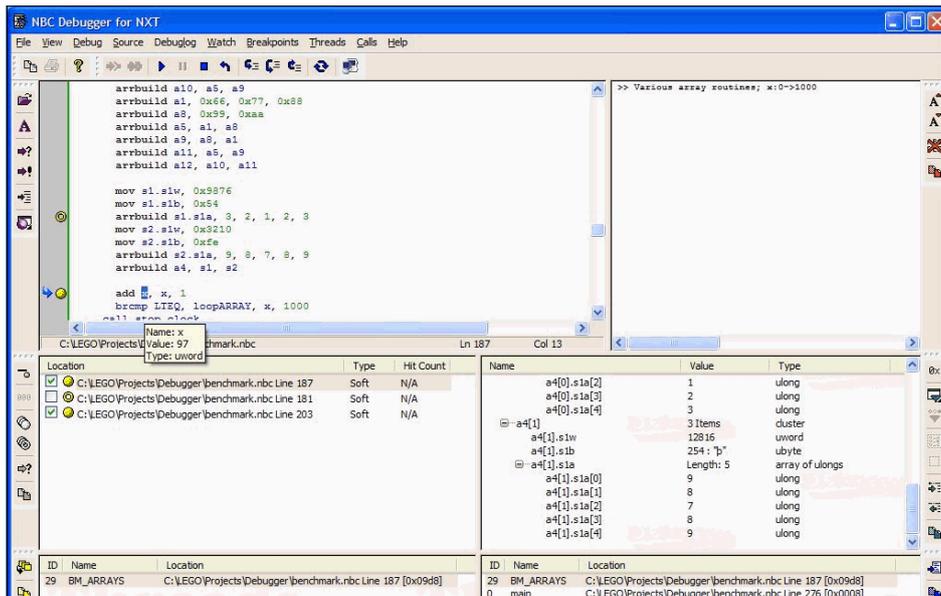


Figura 3. 13: Plataforma de programación NBC.

Fuentes: <http://bricxcc.sourceforge.net/nbc/>

3.4.6. NXC

NXC es un lenguaje de alto nivel similar a C. Utiliza el firmware original de LEGO y está disponible para Win32, Mac OSX y Linux. Ha sido desarrollado por John Hansen. Hay disponible una guía del programador y un tutorial en inglés. Se puede utilizar como editor BricxCC. John Hansen ha publicado un libro sobre programación con NBC y NXC: *NXT Power Programming Robotics in C*⁴⁵[37]

3.4.7. LEJOS NXJ

LEJOS NXJ⁴⁶ facilita la programación del NXT con Java, en la figura 3.14 se muestra la apariencia de la plataforma de programación. Se trata de un completo firmware que sustituye el oficial de LEGO que funciona tanto en Windows como en Linux. Poco a poco se va completando e incorporando nuevos servicios. Un buen libro sobre leJOS NXJ es "*Maximum LEGO NXT Building Robots with Java Brains*" de Brian Bagnall⁴⁷. Este software está disponible en código abierto.

⁴⁵ <http://www.variantpress.com/books/nxt-power-programming>, consultado 20 abril 2011.

⁴⁶ <http://lejos.sourceforge.net/>, consultado 20 abril 2011.

⁴⁷ <http://www.variantpress.com/books/maximum-lego-nxt>, consultado 20 abril 2011.

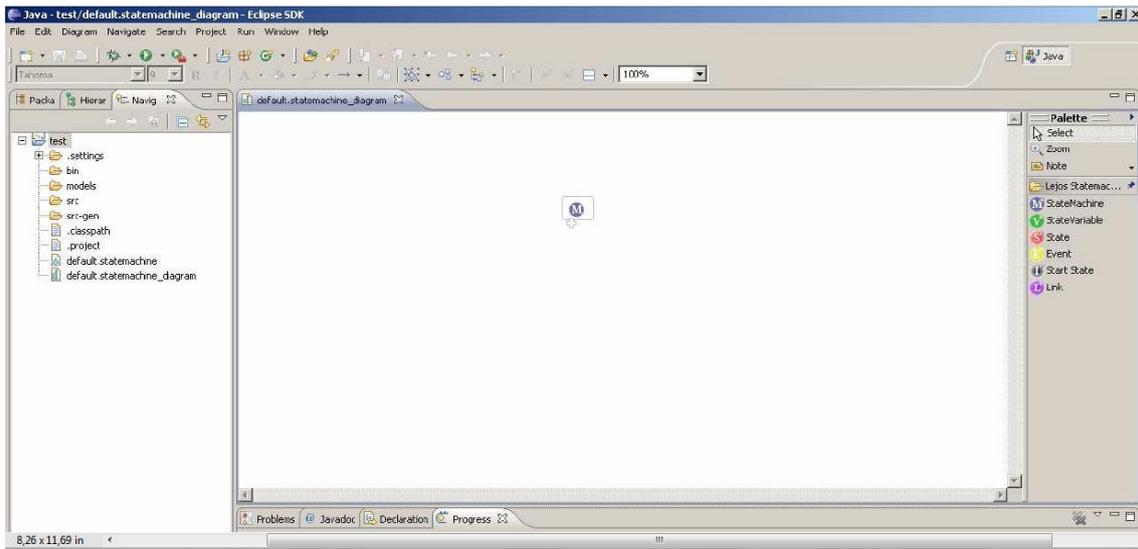


Figura 3. 14: Plataforma de programación Lejos de Java.

Fuente: <http://www.juanantonio.info/lejos-ebook/>

Hay un tutorial de LEJOS y eclipse en la web de Christoph Bartneck⁴⁸, asimismo existe un ebook de LEJOS de Juan Antonio Breña Moral⁴⁹

3.4.8. MICROSOFT ROBOTICS DEVELOPER STUDIO

En la figura 3.15 se observa la apariencia del software Microsoft Visual que incluye un entorno de programación que permite hacer simulaciones en 3D. El software es dirigido tanto a educación, aficionados como a desarrolladores comerciales, el mismo puede soportar diferentes plataformas, como LEGO RCX y NXT, KUKA, Fischertechnik... La última versión del software se publicó en el 2008 R3⁵⁰ totalmente gratuita para todo tipo de usuarios.

Microsoft Robotics Developer Studio facilita programar el NXT con Visual Studio. Los sistemas operativos que lo soportan son Windows CE, Windows Server 2003 Service Pack 2, Windows Server 2008, Windows Vista y Windows XP. En el caso de Windows CE, soporta los servicios desarrollados para dicha plataforma, no así las herramientas de desarrollo.

⁴⁸ <http://www.bartneck.de/2008/03/04/java-lego-nxt-eclipse-tutorial/>, consultado 20 abril 2011

⁴⁹ <http://www.juanantonio.info/lejos-ebook/>, consultado 20 abril 2011

⁵⁰ <http://www.microsoft.com/robotics/>, consultado 20 abril 2011

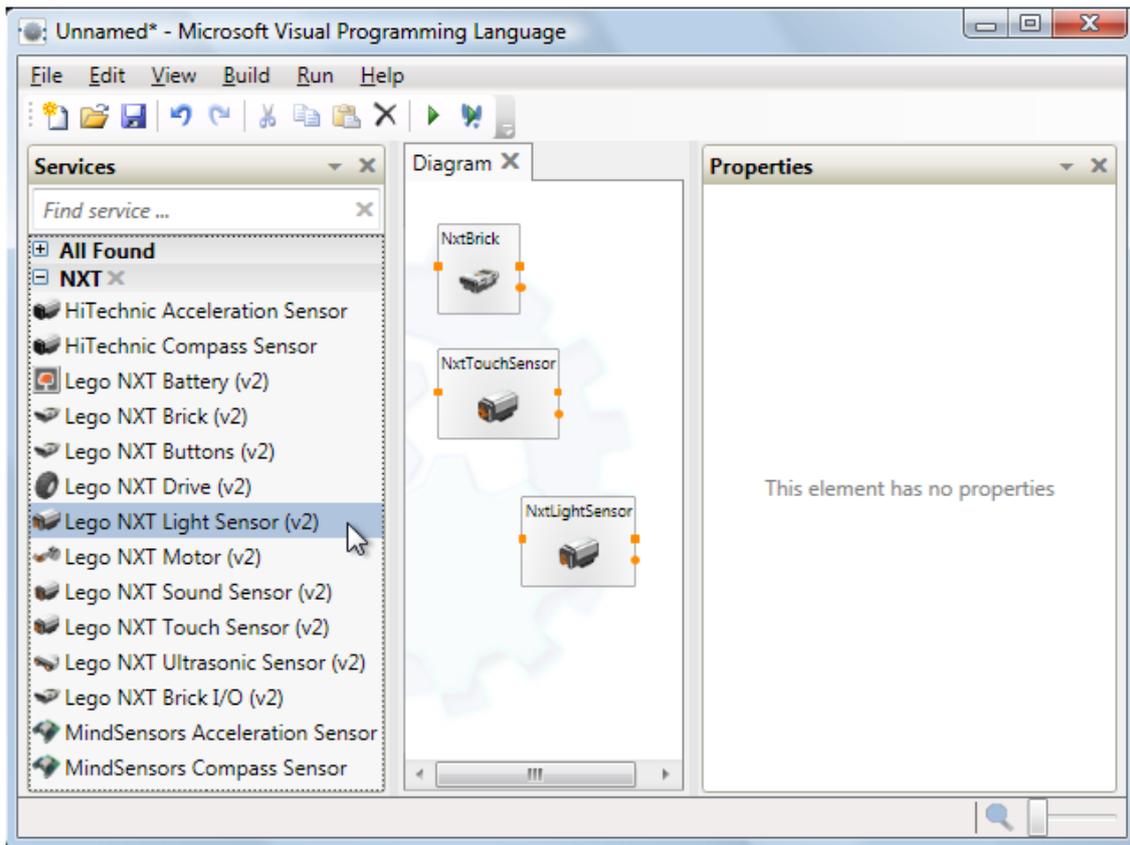


Figura 3. 15: Plataforma de programación Lejos de Java.

Fuente: <http://www.microsoft.com/robotics/>

Asimismo se pueden utilizar los siguientes programas: Visual Studio 2008 Express Editions, Visual Studio 2008 Standard Edition, Visual Studio 2008 Professional Edition o Visual Studio 2008 Team System Edition, Visual Studio 2005 Express Editions, Visual Studio 2005 Standard Edition, Visual Studio 2005 Professional Edition o Visual Studio 2005 Team System Edition.

La empresa francesa SimplySim ofrece un entorno virtual⁵¹ para hacer simulaciones en 3D de los programas que se desarrollen para el NXT.

⁵¹ <http://www.simplysim.net/index.php?scr=scrViewNews&t=2&idnews=18>, consultado 20 abril 2011

CAPÍTULO 4: DESARROLLO EXPERIMENTAL DEL ROBOT LEGO MINDSTORMS NXT.

En el presente capítulo se describirán cada una de las prácticas para la construcción (armado) y programación (simulación de los movimientos a través de sensores) para así comprobar experimentalmente el correcto funcionamiento de los robots LEGO Mindstorms NXT, las mismas que desarrollamos durante el desarrollo del presente trabajo de investigación.

4.1. PRÁCTICA 1: Sistema de Defensa Robótico – Francotirador.

Una vez revisado la parte fundamental de cómo controlar los motores, sensores, y el software de programación NXT, podemos empezar con el diseño máquinas más complejas. En la presente práctica se mostrará cómo construir y programar una aplicación de sistema de defensa robótico (denominado francotirador), como se muestra en la Figura 4.1.



Figura 4. 1: Sistema de Defensa Robótico
Fuente: Los autores

Esta aplicación experimental puede funcionar en modo de mando a distancia o autónomo. Para el modo autónomo, realiza todas las funciones programadas por sí mismo. Se puede programar para mirar alrededor y tirar a objetivos descubiertos por el sensor ultrasónico o responder a señales luminosas detectadas por el Sensor de color.

En el modo de mando a distancia, se deben utilizar dos sensores de tacto (sensibles al ser tocados) para controlar completamente las acciones del robot. Dichos sensores se comportan como botones (o pulsadores) para activar los movimientos del tirador. A diferencia de aplicaciones de robots del tipo explorador y de rescate, el francotirador requiere de tres motores.

Las funciones que ejecuta cada uno de los motores son: hacer girar alrededor para buscar objetivos, mover la torre hacia arriba y abajo; y el último motor (que sirve como tirador) puede realizar un tiro. La combinación de estos tres motores permite al robot apuntar en cualquier dirección.

Materiales para la construcción del sistema de defensa robótico.

Para la construcción del sistema de defensa robótico, primero debe seleccionar las piezas correctamente, como se muestra en la figura 4.1 para evitar inconvenientes.



Figura 4. 2: Piezas para la construcción del sistema de defensa robótico
Fuente: Los autores

Conexión de sensores y motores

Conectar correctamente los sensores y motores al microprocesador (ladrillo) NXT según la Tabla 4.1, asegurando que no interfieran con las ruedas. Un modo de lograrlo, es enrollar los cables alrededor de diferentes piezas del robot. Asimismo, debe asegurarse que el robot pueda girar libremente, mover la torrecilla arriba y abajo con facilidad, sin que los cables bloqueen estos movimientos. Antes de proceder a las pruebas experimentales, comprobar moviendo la torrecilla y las ruedas manualmente.

Tabla 4.1: Conexión entre sensores y motores

Del motor/ sensor	Al puerto NXT	Longitud del Cable
Turret motor	Output port A	Mediano (35 cm/15 in)
Turn motor	Output port B	Mediano
Firing motor	Output port C	Mediano
Color Sensor	Input port 3	Mediano
Ultrasonic Sensor	Input port 4	Corto (20 cm/8 in)

Fuente: Los autores

Programación del sistema de defensa robótico

Para programar el robot, necesitará crear nuevos bloques de programación necesaria para la comprobación experimental:

➤ **Aprendiendo a programar sensores y motores.**

Explorar la paleta completa de programación.

El software NXT cuenta con tres paletas de programación diferentes. La paleta común y la paleta de encargo siendo las más básicas al momento de programar, por esto es necesario utilizar la paleta completa que se muestra en la Figura 4.3.

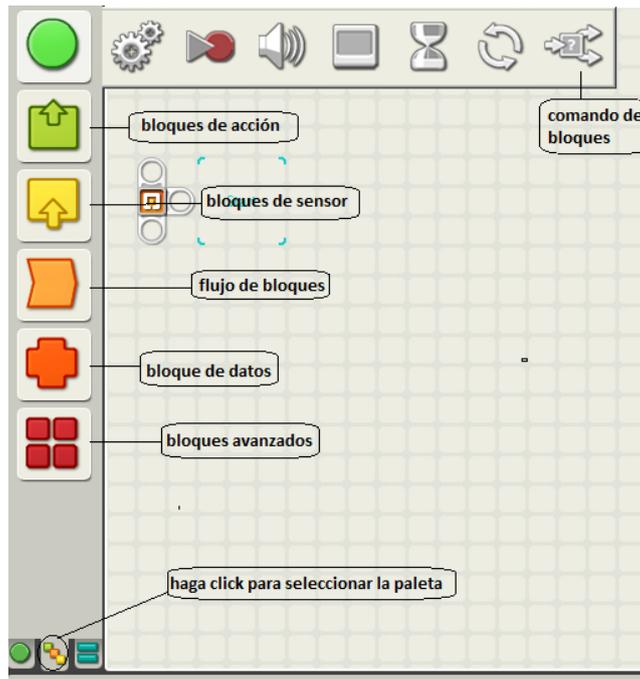


Figura 4. 3: Paleta completa de programación.
Fuente: Los autores

La paleta completa de la figura 4.3 contiene todos los bloques que pueden ser usados en el software de programación NXT, excepto la opción **Mis Bloques** que permite al programador crear sus propios bloques para determinada tarea. Cada icono (coloreado) sobre la paleta completa representa un cierto tipo de bloque en las categorías: bloques de acción, de sensor, de flujo de bloques, de datos, y avanzados.

Bloques de acción, son bloques para hacer que el robot realice una acción, como la rotación de los motores, juego de un sonido, o la demostración de una línea de texto sobre la pantalla de NXT.

Bloques de sensor, son bloques para leer valores de los sensores para el empleo en sus programas.

Bloques de flujo, típicamente son usados para cambiar el flujo de un programa. Por ejemplo, algunos bloques pueden ser repetitivos (a través de un loop o lazo).

➤ **Configuración de sensor de color.**

En la figura 4.4 se muestra como configurar el sensor de color, el mismo que sirve para determinar el color de una superficie, pero también sirve como una lámpara de colores roja, verde, o azul. Para usar el bloque de lámpara de color, abrir el panel de configuración, seleccione el puerto al que está unido. Escoger cuatro bloques para programar el sensor de color (ver figura 4.4 los bloque a, b, c y d), en la figura 4.5 se muestra la configuración del *loop*⁵² (bucle) de manera indefinida.

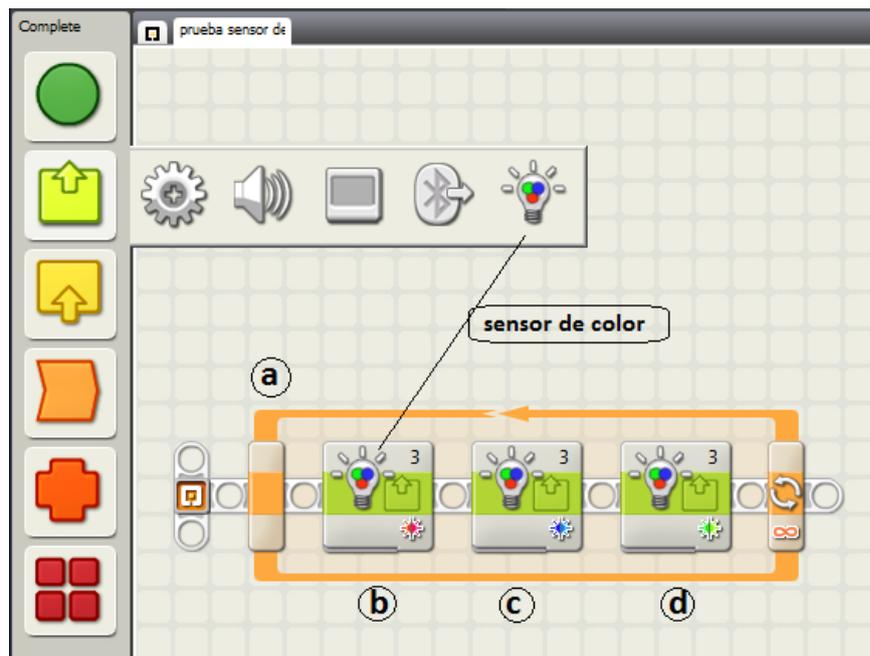


Figura 4. 4: Diseño de bloques para el funcionamiento de los sensores de color
Fuente: Los autores

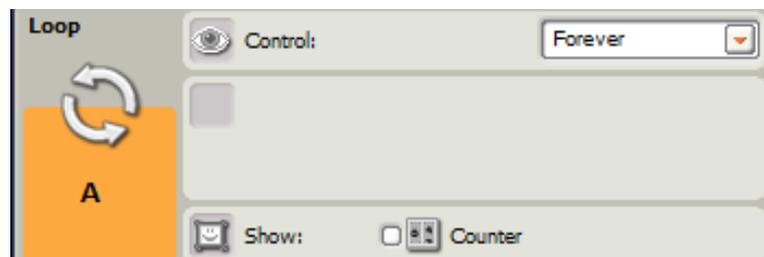


Figura 4. 5: Configuración del loop para el sensor de color
Fuente: Los autores

⁵² **Loop**, es una sentencia que se realiza repetidas veces a un trozo aislado de código, hasta que la condición asignada a dicho bucle deje de cumplirse. Generalmente, un bucle es utilizado para hacer una acción repetida sin tener que escribir varias veces el mismo código, lo que ahorra tiempo, deja el código más claro y facilita su modificación en el futuro.

Y finalmente en la figura 4.6 muestra la configuración interna de cada uno de los 3 sensores.

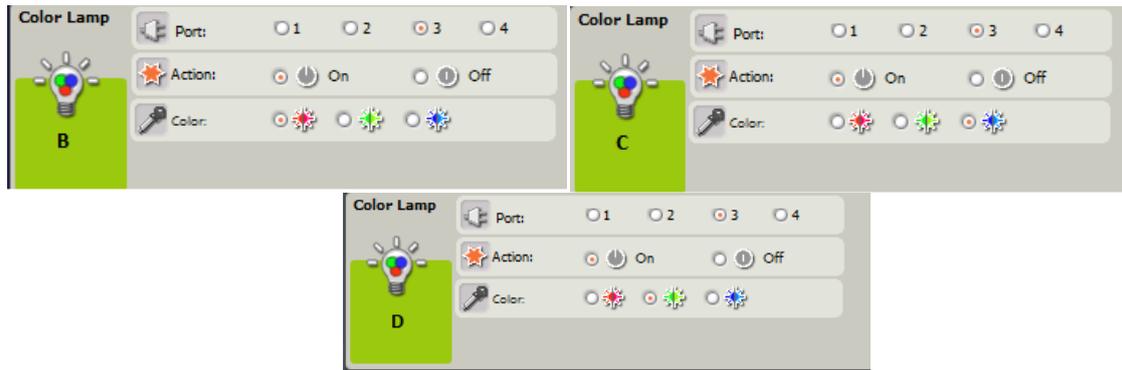


Figura 4. 6: Configuración de cada uno de los sensores de color.
Fuente: Los autores

➤ **Configuración de los motores.**

En la opción bloques de acción podemos encontrar el bloque motor (ver figura 4.7) que permite controlar los motores. Una importante consideración entre los dos es que el bloque motor tiene rasgos extras para controlar motores individuales como se muestra en la figura 4.8, perfecto para vehículos con dos ruedas.

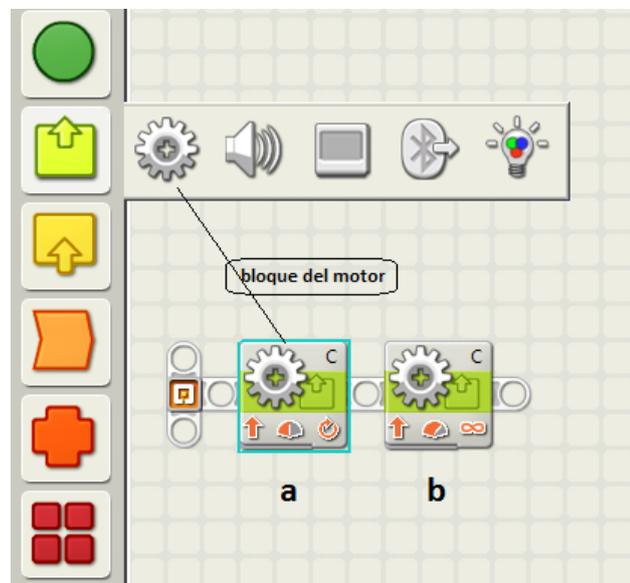


Figura 4. 7: Diseño de bloques para el funcionamiento de los motores.
Fuente: Los autores

Normalmente, cuando se pone un motor en movimiento a un cierto nivel de poder (como 50, ver figura 4.8), las vueltas del motor son más lentas, sin embargo, al usar la opción “control motor power” el NXT automáticamente aplicará más poder con el motor siempre que exista una carga sobre ella, de modo que el motor siga dando vueltas a una velocidad constante.

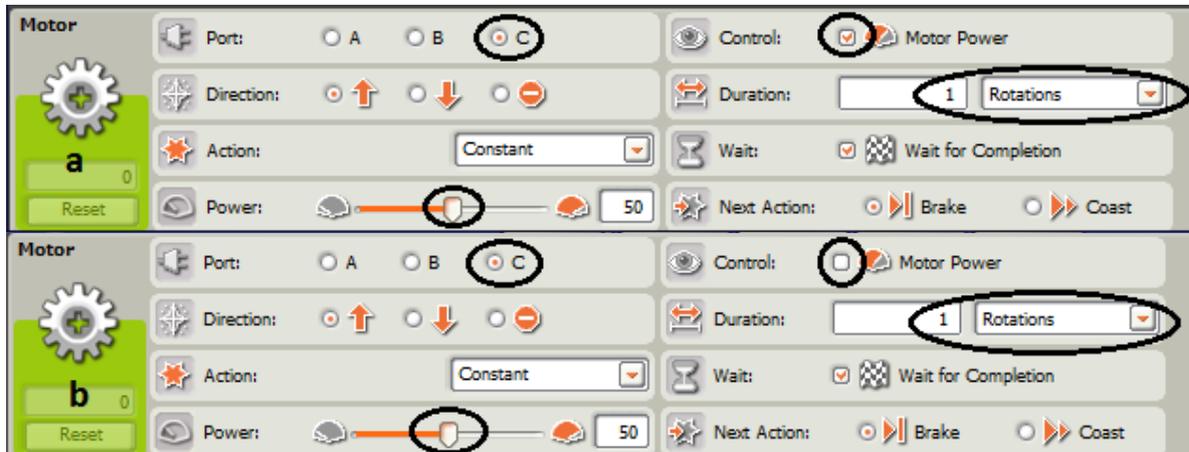


Figura 4. 8: La configuración individual de los motores.

Fuente: Los autores

➤ **Programación básica del sistema robótico de defensa modo autónomo.**

Antes de la creación de programas avanzados para el robot de defensa, empezaremos con programar las funciones mecánicas del robot para darle una mejor comprensión de cómo trabajan e inclusive para localizar problemas. Por ejemplo, si el motor no permite girar al robot, esto ocurre porque probablemente en la construcción del robot están desalineados. Crearemos un programa que hace que cada motor sobre el robot avance y luego retroceda un momento, con velocidades diferentes del motor.

En este modo autónomo, el NXT controla los motores y las acciones del robot basadas en la programación de los sensores. Para programar, usaremos el sensor de color, para cual usaremos el sensor ultrasónico. Su programación permitirá girar para continuar con la búsqueda del objetivo de un lanzamiento. Si el robot ve un objetivo, esto levantará la torrecilla.

Paso 1: En la figura 4.9 se muestra la programación (inicial) del motor para que pueda girar activando el motor B de duración ilimitada. En el caso del rodillo de tiro que gira hacia la derecha, con la opción esperar, permite que sensor ultrasónico busque el objetivo. Una vez que el robot cumple su objetivo, cercano a 45 cm (ver la configuración interna en la figura 4.10) el motor B se apaga, provocando que la torrecilla se levante haciendo girar el motor hacia atrás.

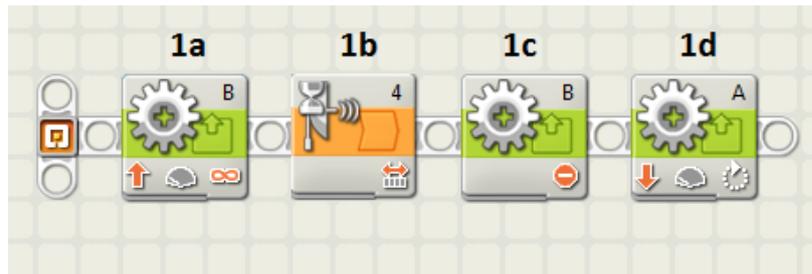


Figura 4. 9: Primera parte de la programación básica modo autónomo.
Fuente: Los autores

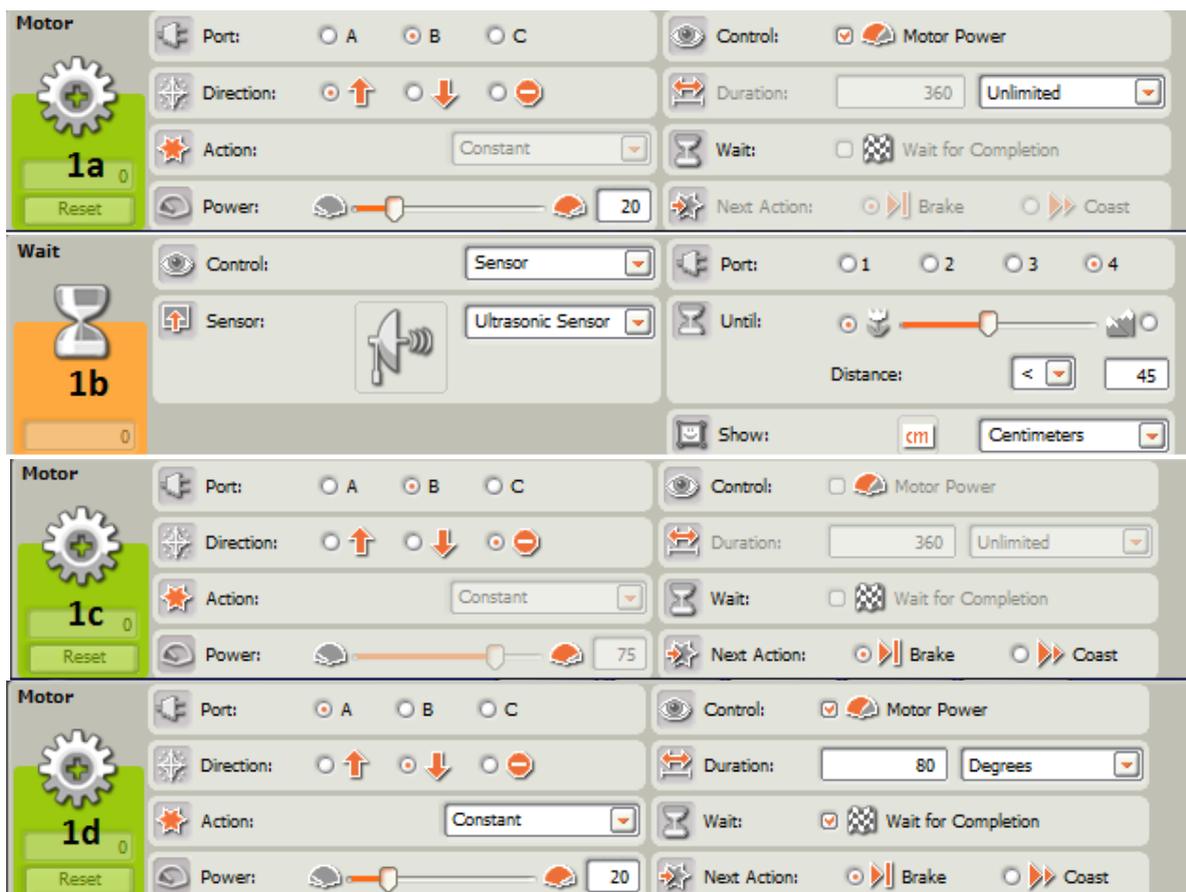


Figura 4. 10: Configuración interna del motor para determinada aplicación.
Fuente: Los autores

Paso 2: Los bloques creados en el paso 1, sirve para detectar objetivos a través del sensor algo más cerca, pero esto no podía determinar exactamente como un objetivo cercano. Para averiguar, usaremos ahora un bloque tipo interruptor mostrado en la figura 4.11. Si el objeto es descubierto a una distancia de 25 cm (10 pulgadas), inmediatamente pega un tiro a dos balas.

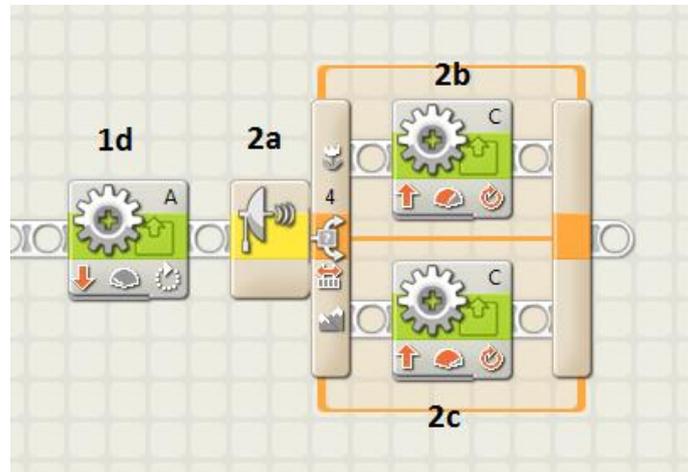


Figura 4. 11: Segunda parte de la programación básica modo autónomo.
Fuente: Los autores

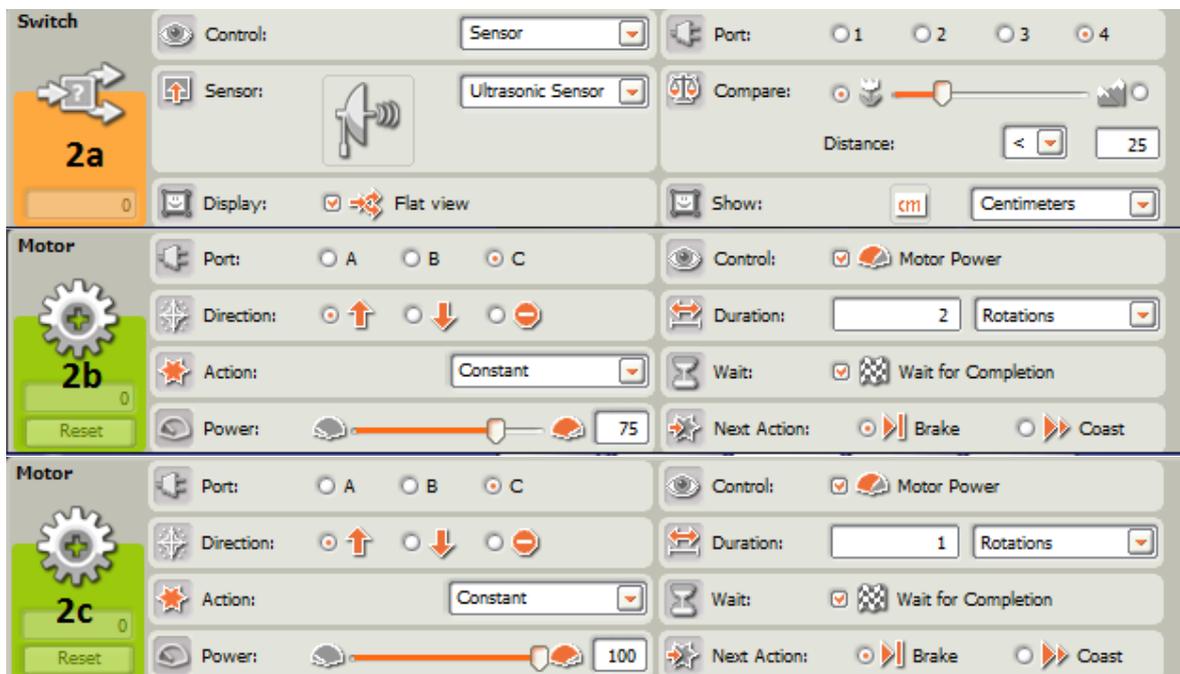


Figura 4. 12: Configuración interna para otra manera de programación básica de modo autónomo.
Fuente: Los autores

Paso 3: En el paso 2 se programa al robot para que tire las pelotas, para el siguiente nuevo bloque (ver figura 4.13) que permite programar el tiempo de espera durante 0,5 s donde baja la torrecilla, da vueltas ligeramente, para luego comenzar ejecutar el programa entero (compuesto de 5 pasos) para tirar el mismo objetivo repetidamente.

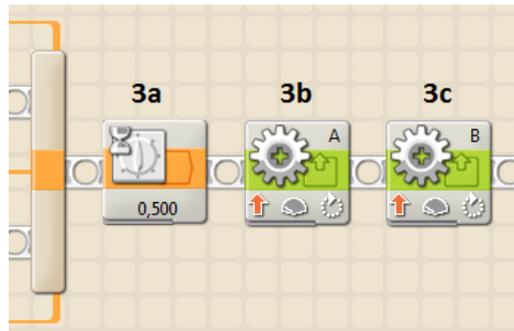


Figura 4. 13: Tercera parte de la programación básica modo autónomo.
Fuente: Los autores

En la figura 4.14 se muestra la configuración del tiempo de espera y la activación de los motores, que se mostró en la figura 4.13.



Figura 4. 14: Configuración de los bloques adicionales del paso 3.
Fuente: Los autores

Paso 4: Ahora incluiremos bloque de loop (bucle) para repetir la secuencia de bloques pero colocando un loop al inicio de su programa y luego el seleccionar el resto de los bloques. El resultado es el programa mostrado en figura 4.15 y en la figura 4.16 se muestra la configuración del bucle de manera indeterminada.

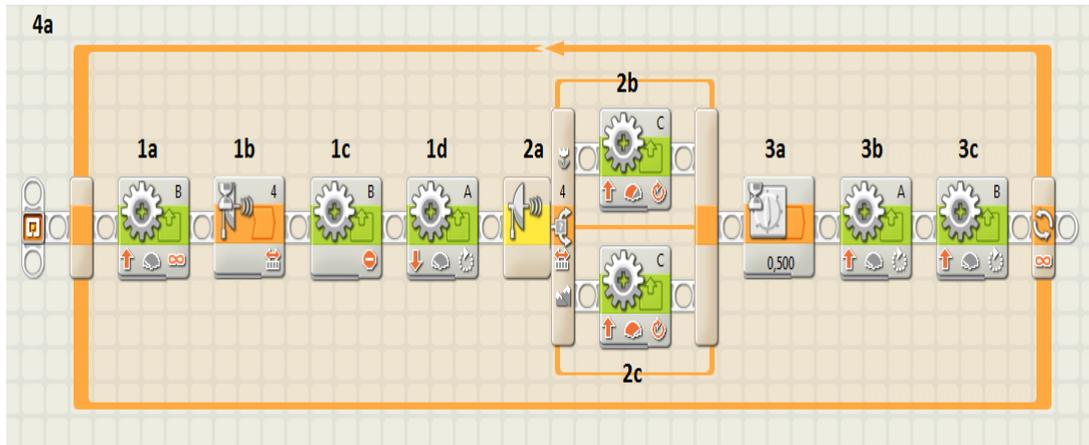


Figura 4. 15: Cuarta parte de la programación básica modo autónomo.
Fuente: Los autores



Figura 4. 16: Configuración de los bloques adicionales del paso 4.
Fuente: Los autores

Paso 5: Para finalizar, insertaremos bloques adicionales del sensor de color para indicar el estado de lanzamiento, configuraremos los colores de cada bloque insertado (sensor de color). En la figura 4.17 se muestra el programa final.

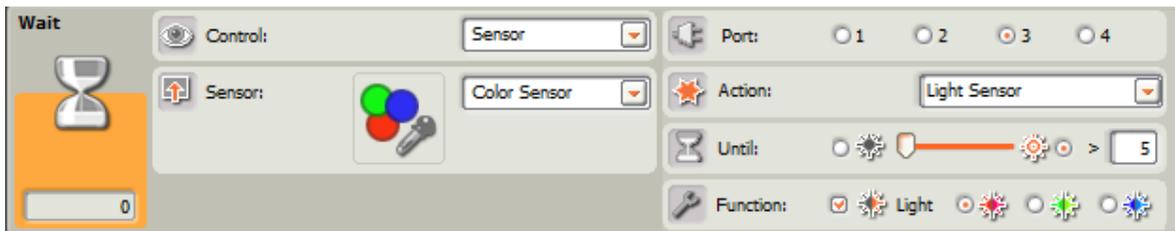


Figura 4. 18: Configuración del bloque espera para la programación intermedia.
Fuente: Los autores

Para la siguiente programación (ver Figura 4.19) el robot necesita de un espacio oscuro, por ejemplo, como un cuarto de baño sin ventanas con las luces apagadas. Una vez realizada la programación para el lanzamiento, colocar el robot en este espacio, abandonar el mismo y esperar 30 segundos teniendo cuidado al cerrar la puerta.

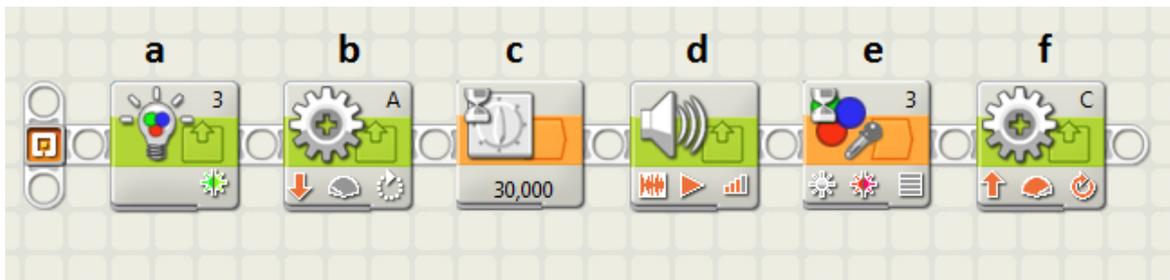


Figura 4. 19: Programa final intermedia del sistema de defensa robótico autónomo
Fuente: Los autores

Solo en este espacio oscurecido, la medida del sensor de color de movimiento debería estar cerca una vez que la puerta se ha abierto, y una vez que el valor moderado excede 5 (ver configuración en la figura 4.20), el robot del sistema de defensa realiza el lanzamiento, al detectar que alguien ha abierto la puerta, defendiendo su territorio.

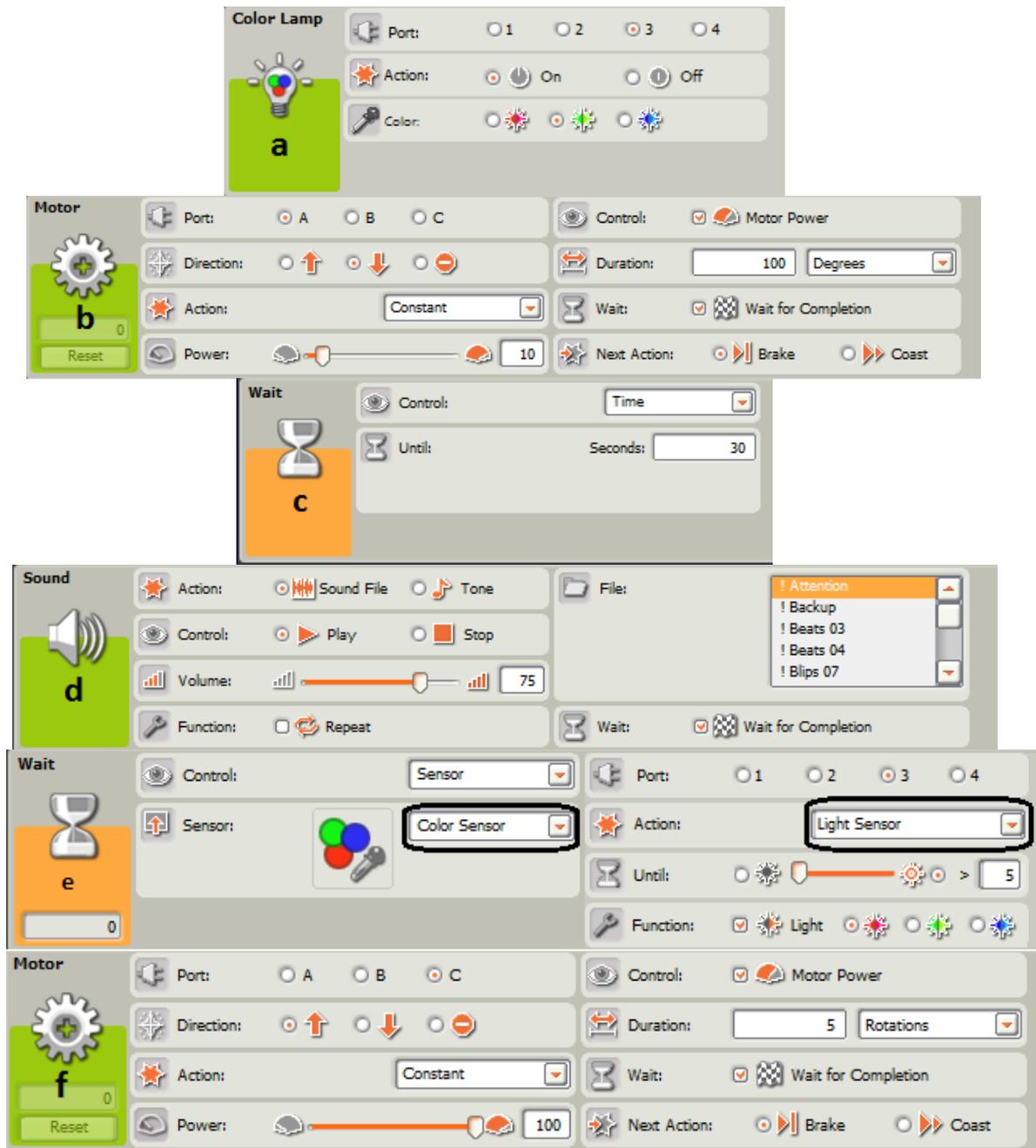


Figura 4. 20: Configuración final para la programación intermedia.

Fuente: Los autores

Si el robot no controla la programación esperada, el valor para el lanzamiento es 5, puede ser demasiado baja por el espacio que no está completamente oscuro. Para resolver este inconveniente daremos un valor más alto, de 10.

4.2. PRÁCTICA 2: Robot Hexápodo - Strider.

Una vez familiarizado con modelos interesantes que se mueven sobre ruedas, para la siguiente prueba experimental construiremos un robot que pueda caminar. Tal creación es un poco más difícil de hacer, pero en esta práctica encontrará instrucciones para la construcción y programación del robot hexápodo – Strider, caminando en seis patas, como se muestra en la figura 4.21.



Figura 4. 21: Robot Hexápodo – Strider.
Fuente: Los autores

Ahora vamos a pasar a la construcción del robot siguiendo las instrucciones del **Anexo 2**. Antes de empezar a construir, seleccionaremos las piezas que necesita para completar el robot, como se muestra en la figura 4.22. Una vez terminada la construcción del robot hexápodo, procederemos a la programación del mismo para que pueda caminar y responder a la interacción humana.



Figura 4. 22: Piezas para la construcción del robot hexápodo – Strider.
Fuente: Los autores

Los robots hexápodos usan tres conjuntos de motores idénticos para caminar, donde cada uno de los motores puede controlar un par de patas. Los módulos están interconectados con la pierna de un marco en forma de triángulo que también lleva el NXT con varios sensores conectados entre sí. Dos sensores de contacto se incluirán al robot Strider como antenas, la detección por el tacto de objetos o personas en el entorno (no se utilizan como topes para evitar que el robot tropiece).

Adicional usaremos el sensor de ultrasonido que permitirá medir la distancia de objetos cercanos, mientras que el sensor de color en el modo de sensor de luz puede detectar si está oscuro o claro.

En la Tabla 4.2 y Figura 4.23 muestra la manera de conexión de los sensores. Al conectar los cables, les guiará a través del espacio bajo el NXT para que no sobresalgan.

Tabla 4.2: Conexión del cableado para el robot hexápodo – Strider.

Del motor/ sensor	Al puerto NXT	Longitud del Cable
Sensor Touch Derecho	Input Port 1	Mediano (50 cm/20 in)
Sensor Touch Izquierdo	Input Port 2	Largo
Sensor de color	Input Port 3	Mediano (35 cm/15 in)
Sensor de ultrasonido	Input Port 4	Mediano

Fuente: Los autores



Figura 4. 23: Vista de la conexión entre los puertos del NXT y los sensores del robot hexápodo Strider.

Fuente: Los autores

➤ **Pruebas básicas para que pueda caminar el robot hexápodo.**

En la figura 4.24 se muestra la programación en bloque de los tres motores para ejecutar los movimientos de los seis patas del robot hexápodo.

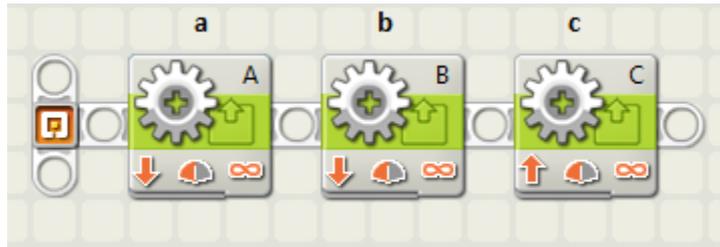


Figura 4. 24: Programación básica para mover las patas del robot hexápodo.
Fuente: Los autores

En la figura 4.25 se puede apreciar la configuración de cada uno de los tres motores (A, B y C), con diferentes direcciones, con potencia baja y de duración indefinida.



Figura 4. 25: Configuración básica para ejecución correcta de los motores.
Fuente: Los autores

Otro programa interesante para pruebas rápidas del robot hexápodo Strider para caminar hacia adelante, donde en la figura 4.26 se ha creado un programa donde hay bloque llamado **adelante** (internamente se encuentra la programación vista en la figura 4.24) y otro que permite que se mueva durante un intervalo de 5 s, en la figura 4.27 se muestra el intervalo de tiempo para ejecutar dicho movimiento.

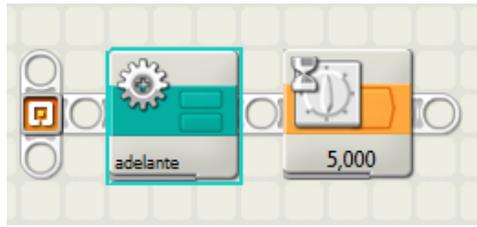


Figura 4. 26: Programación de bloques para el movimiento del robot hexápodo.
Fuente: Los autores

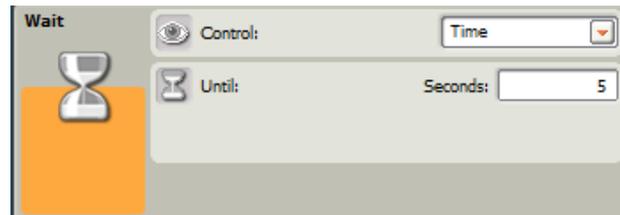


Figura 4. 27: Configuración del tiempo que se moverá el robot.
Fuente: Los autores

➤ **Programación básica del robot hexápodo – Strider.**

Ahora programaremos dos nuevos bloques para que el robot hexápodo pueda desplazarse tanto por la izquierda como por la derecha. Los bloques son esencialmente similares a las vistas en las figuras 4.24 y 4.26, salvo que para la siguiente programación configuraremos las direcciones en las que tendrá que moverse el robot, como se muestran en la figura 4.28. Antes de iniciar con la ya mencionada programación veremos la tabla 4.3 para conocer la dirección y potencia de cada uno de los tres motores.

Tabla 4.3: Configuración y ajustes de potencia para los 3 motores.

Nombre del Bloque	Motor A	Motor B	Motor C
Adelante	Retroceder / 50	Avanzar / 60	Retroceder / 50
Izquierda	Avanzar / 60	Retroceder / 50	Retroceder / 50
Derecha	Retroceder / 50	Retroceder / 50	Avanzar / 60

Fuente: Los autores

A continuación seguir las instrucciones (pasos) para la programación en bloque del robot hexápodo – Strider.

Paso 1: La programación de los bloques de la figura 4.28 forman la estructura principal del programa, que permite el funcionamiento hasta que manualmente pueda abortar la programación, por lo que utilizaremos un bucle, dentro del mismo, se colocan dos bloques de contacto para saber si el sensor está presionado.

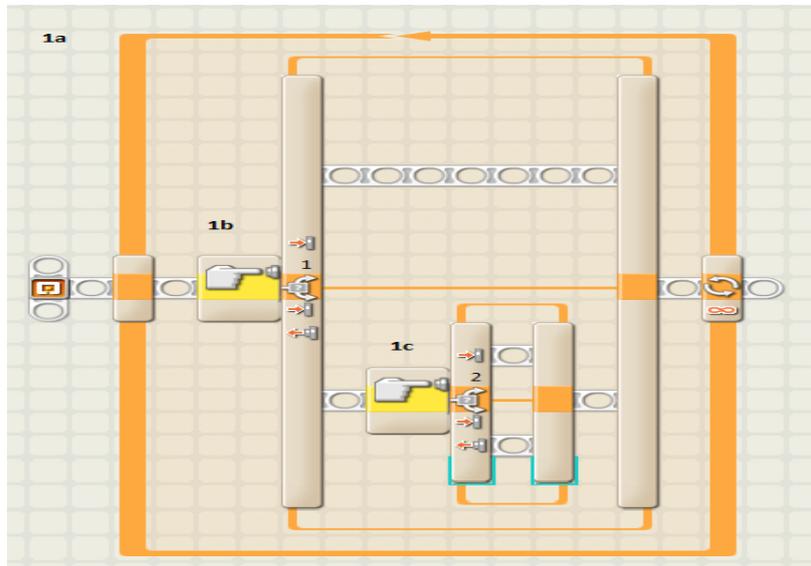


Figura 4. 28: Programación del paso 1 para el robot hexápodo.

Fuente: Los autores

En la figura 4.29 se muestra la configuración de cada uno de los bloques de la programación del paso 1.

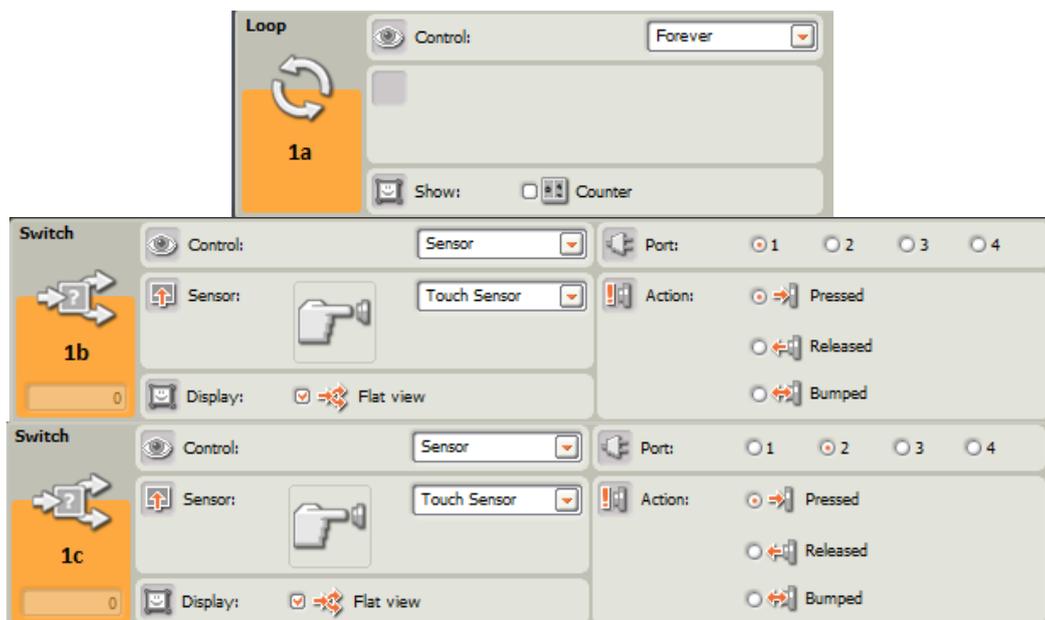


Figura 4. 29: Configuración de los bloques del paso 1.

Fuente: Los autores

Paso 2: Ahora configuraremos los movimientos del robot mediante la creación de nuevos bloques (adelante, izquierda, derecha) mostrados en la figura 4.30. Al pulsar el sensor de contacto conectado al puerto 1 del robot hexápodo, camina a la derecha y espera un tiempo para seguir adelante en esta dirección durante cinco segundos (ver configuración en la figura 4.31).

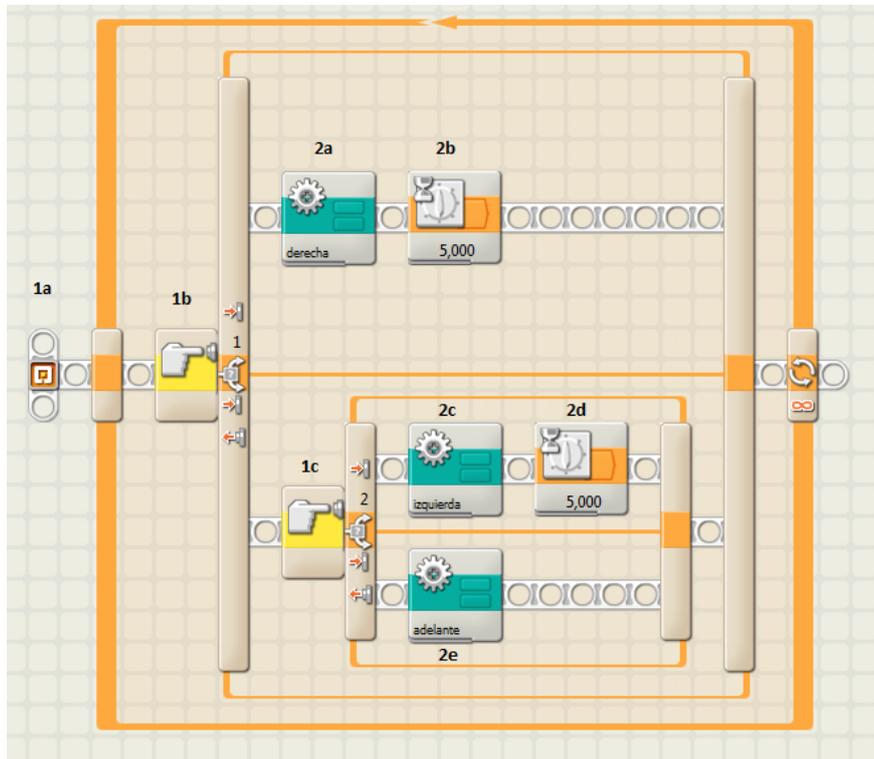


Figura 4. 30: Programación del paso 2 para el robot hexápodo.
Fuente: Los autores

Asimismo se debe volver al inicio del programa para ver si el sensor está presionado. Si no hay un sensor que se presiona en ese momento, el robot debe seguir caminando hacia adelante.

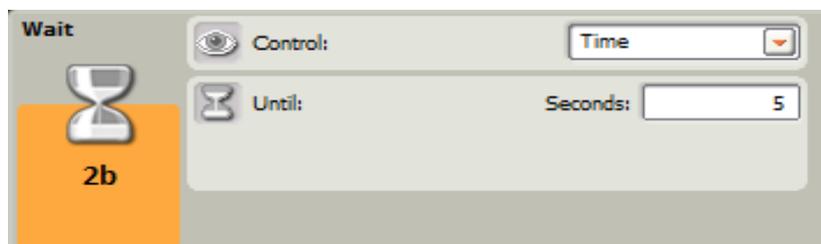


Figura 4. 31: Configuración de los bloques del paso 2.
Fuente: Los autores

La programación en bloque de los pasos 1 y 2 constituyen la base fundamental de esta aplicación, para lo cual cargaremos la programación al robot hexápodo para comprobar su correcto funcionamiento. Para hacerlo más interactiva su programación, añadiremos una pantalla para sonido, y bloques de sensores de color (tipo lámpara) en los pasos 3, 4 y 5.

Paso 3: Aquí agregaremos otro bloque (ver figura 4.32) de programación que debe ejecutar, cuando el sensor de contacto derecho es presionado, una luz azul se enciende, para lo cual el robot debe detenerse, y a través de la pantalla (display) del robot mostrar el siguiente mensaje "Se ha pulsado el sensor de la derecha" en la pantalla del NXT.

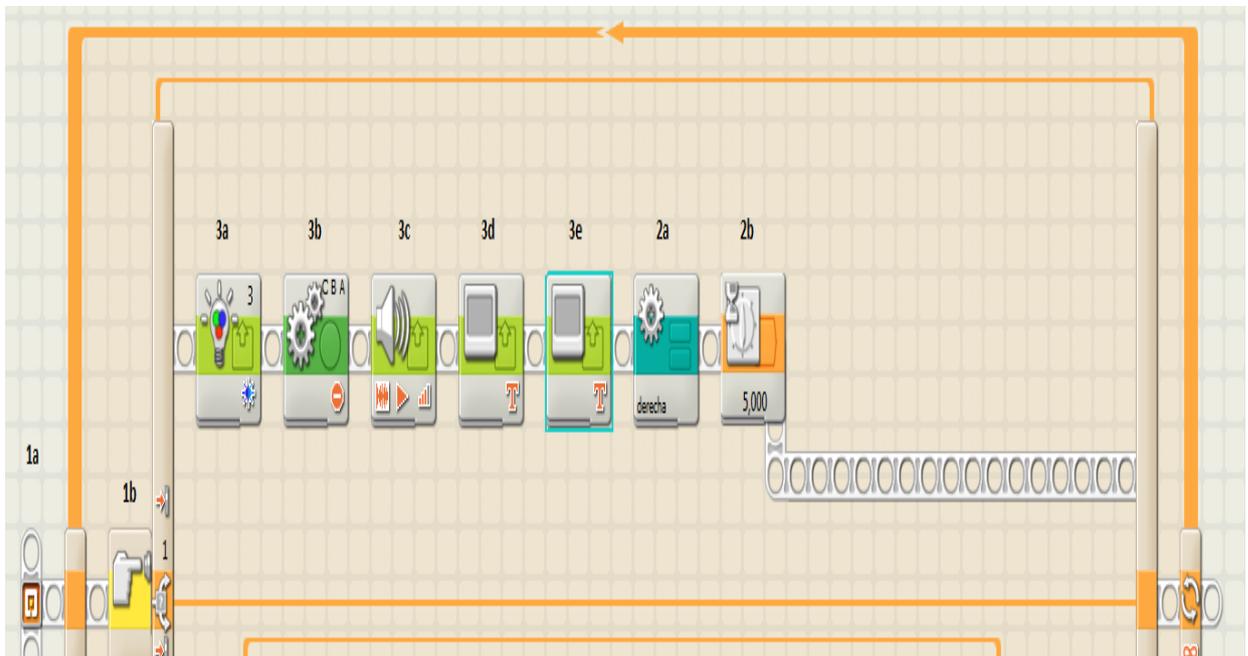


Figura 4. 32: Programación del paso 3 para el robot hexápodo.

Fuente: Los autores

En la figura 4.33 se muestra la configuración de cada uno de los bloques programados en el paso 3.



Figura 4. 33: Configuración de los bloques del paso 3.

Fuente: Los autores

Paso 4: Los bloques de programación aquí son similares a los del paso 3. Excepto que se ejecutan cuando el sensor de contacto es presionado (ver figura 4.34). En consecuencia, para el bloque de sonido y de visualización deben estar configurados para que el robot camine hacia la izquierda.

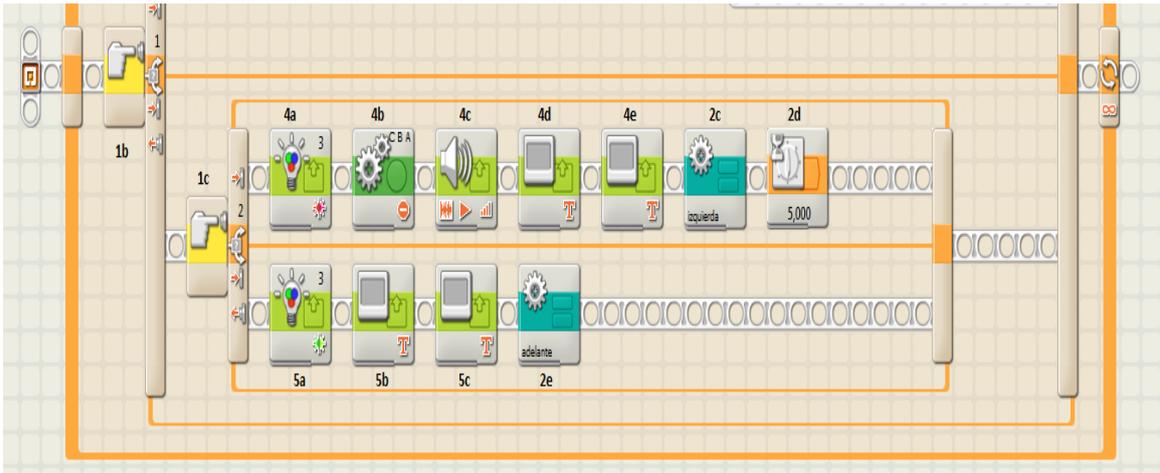


Figura 4. 34: Programación del paso 4 y 5 para el robot hexápodo.
Fuente: Los autores

La figura 4.35 muestra la configuración de los bloques de programación del paso 4.

Figura 4. 35: Configuración de los bloques del paso 4.
Fuente: Los Autores

Paso 5: Los bloques de aquí se configuran para cuando el sensor de tacto no toque con nada, por lo que la pantalla del NXT mostrará "No ha sido presionado ningún sensor". Después de configurar para que el robot camine hacia adelante (*Walk-Forward*), el programa se remonta para ver si los sensores son presionados, por lo que no necesita de un bloque de espera.

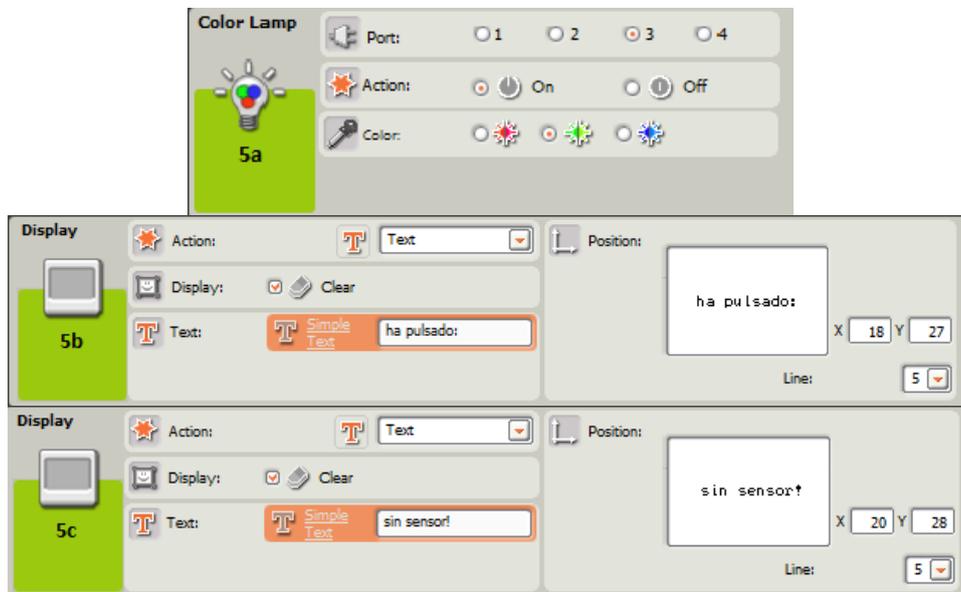


Figura 4. 36: Configuración de los bloques del paso 5.

Fuente: Los autores

➤ **Programación intermedia del robot hexápodo – Strider.**

Para la presente programación intermedia el robot hexápodo tendrá que caminar hacia adelante hasta que alguien enciende una luz, o detecte un sonido (como un grito) para lo cual inmediatamente quedarse quieto hasta que la luz se apague. Antes de crear esta programación, necesita aprender dos nuevos trucos de programación (cajas de comentarios y umbrales) para lo cual debe seguir las instrucciones que se muestran en la Figura 4.37.

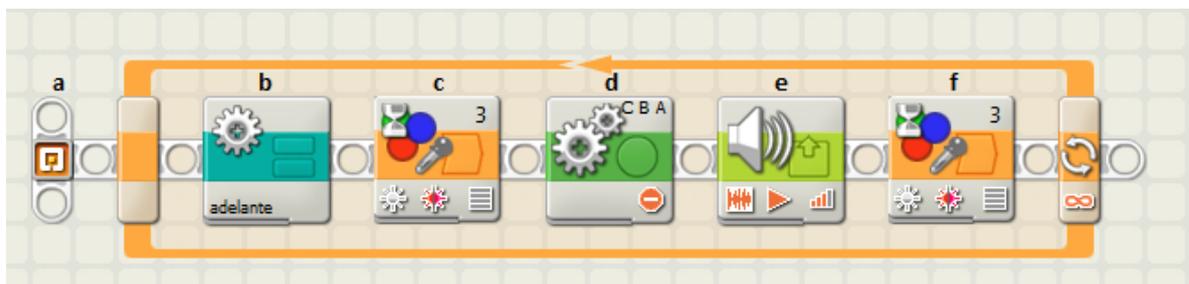


Figura 4. 37: Programación intermedia del robot hexápodo.

Fuente: Los autores

Después del movimiento hacia adelante del robot hexápodo, ahora su programación es para que pueda esperar un nivel de luz que supera el umbral. Una vez superado el umbral, se detienen los motores y el robot emite un sonido fuerte. Para terminar el sensor de luz al detectar que es de noche, se reinicia el ciclo y comienza a caminar de nuevo robot Strider.

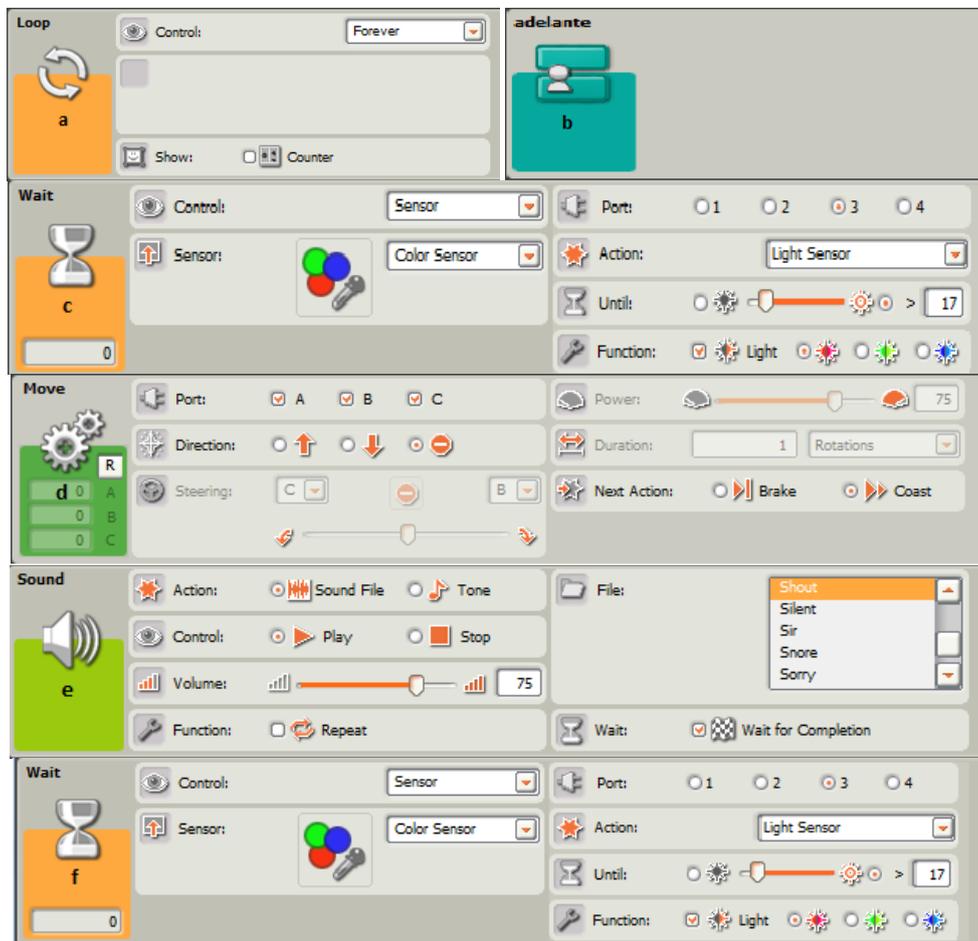


Figura 4. 38: Configuración de los bloques de programación intermedia.
Fuente: Los autores

4.3. PRÁCTICA 3: Robot escalador

Muchos robots NXT pueden moverse sobre ruedas, algún movimiento en las piernas, y otros sin dejar de hacer algo en su lugar. Sin embargo, el robot escalador que se desea construir en este acápite es diferente a los anteriores, moviéndose verticalmente. Este robot escalador puede cumplir un objetivo como subir entre dos paredes, por una chimenea como se muestra en la figura 4.39.

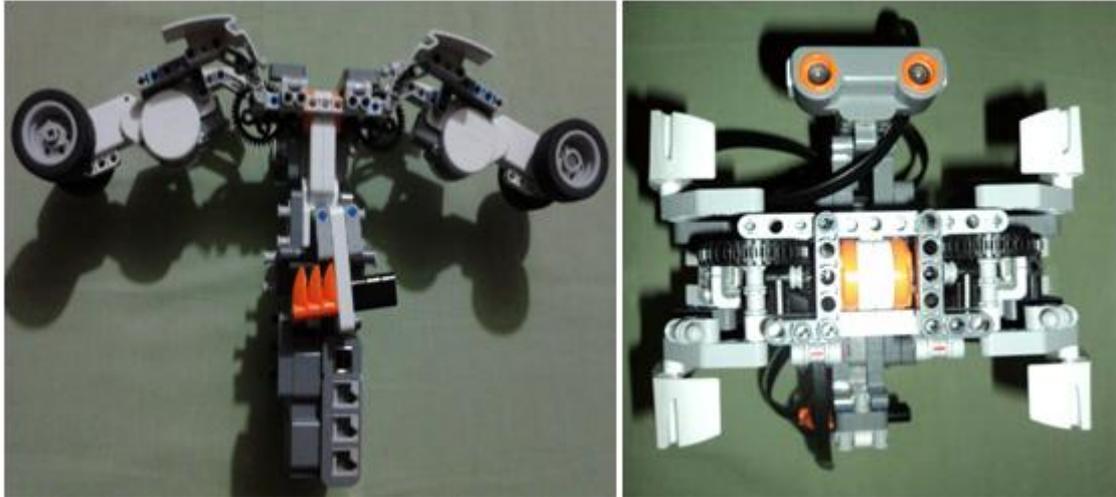


Figura 4. 39: Vista del robot escalador.
Fuente: Los autores

Para la construcción del robot debemos seguir instrucciones (pasos) en las siguientes páginas. Primeramente seleccionamos los materiales necesarios para la construcción del robot mostrados en la figura 4.40.



Figura 4. 40: Piezas para la construcción del robot escalador.
Fuente: Los autores

Antes de programar este robot, tendríamos que utilizar alguna aplicación como por ejemplo "una chimenea" que es apropiada para que el robot pueda subir, donde las paredes de dicha chimenea deberían ser así:

- ❖ Sólido e incapaz de moverse mientras asciende el robot.
- ❖ Perfectamente paralela el uno al otro.
- ❖ La separación entre las paredes será de 32 cm (12,5 in).

Incluso si las paredes de la chimenea cumplen con estas exigencias, hay que tener mucho cuidado con el robot porque puede caerse. Primero probaremos que el robot suba para eso debemos dirigirlo con nuestras manos de modo que pueda agarrarlo si esto comienza a hacerse inestable. Para evitar daños en el robot, pondremos una almohada en el fondo de la supuesta chimenea y después que el robot se levante del piso.

El programa que crearemos para este robot hará que se mueva hasta llegar al tope el techo. Si una o ambas paredes no alcanzan el techo, el robot no sería capaz de verlo. En aquel caso, detenerlo con la mano colocando la misma delante del sensor para que el robot pueda bajar.

➤ **Programación del robot escalador de chimenea.**

Una vez construido el robot escalador con las piezas de LEGO y encontrado la chimenea apropiada para que pueda subirla, entonces ya podemos empezar con la programación en solo 3 pasos. Recordemos que la programación del robot permitirá que suban por la chimenea hacia el techo y regresar a su posición inicial de partida.

Paso 1: Ampliación de los brazos.

Primero, el programa debe permitir encender los motores de las ruedas B y C, después, los brazos son ampliados para tocar las paredes de la chimenea haciendo girar el motor. Como la subida de robot, hace que el NXT siga aplicando una potencia necesaria al motor de modo que las ruedas del robot sean presionadas contra las paredes, dándolos mayor presión. Configuraremos los bloques que realizan estas acciones como se muestra la figura 4.41.

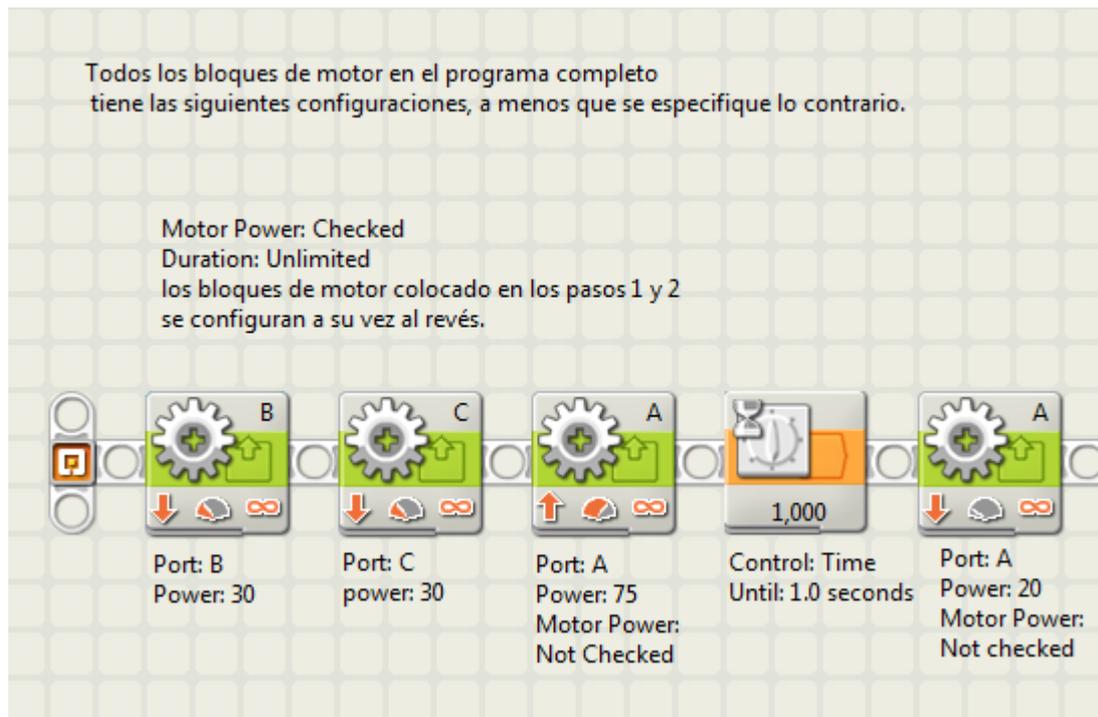


Figura 4. 41: Programación del paso 1 para el robot escalador de chimenea.

Fuente: Los autores

Paso 2: Montañismo y permanencia de equilibrio.

El robot escalador de chimeneas también puede actuar con el sensor de color para hallar errores de equilibrio sobre el eje de ordenadas. Al subir el robot sondea este sensor repetidamente y controla los motores de las ruedas para mantener el equilibrio hasta que el sensor ultrasónico descubra la cima de la chimenea, la programación de la misma se puede ilustrar en figura 4.42.

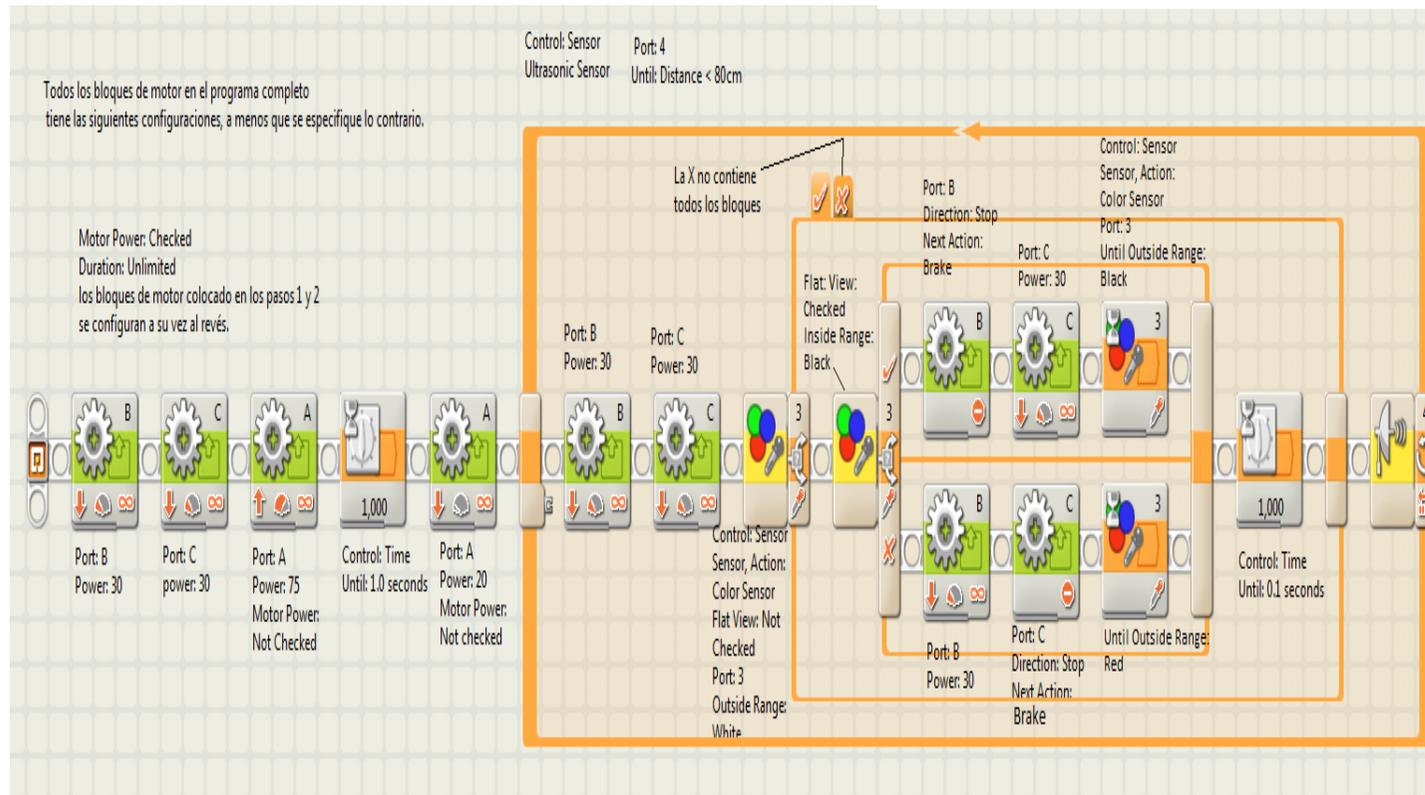


Figura 4. 42: programación y configuración de los bloques del paso 2, encendiendo los motores de las ruedas y extendiendo los brazos.
Fuente: Los autores

Paso 3: Bajada, permanencia en equilibrio, y parada.

Una vez en lo alto, el robot deja de moverse, los bloques que controlan el robot para bajar son similares a los que se programo para subir la chimenea, donde los bloques de los motores ahora hacen el movimiento de robot en la dirección contraria cuando ascendía. Por lo tanto, simplemente copiaremos el bloque de programación anterior en contenido y después modificaremos su programación mostrada en la figura 4.43, realizando los ajustes necesarios para cada bloque como se ha indicado anteriormente.

Para este robot se requiere un poco más de atención que otros programas ya realizados hasta ahora. Antes de controlar al robot, asegúrese de que el robot es colocado en medio de la chimenea

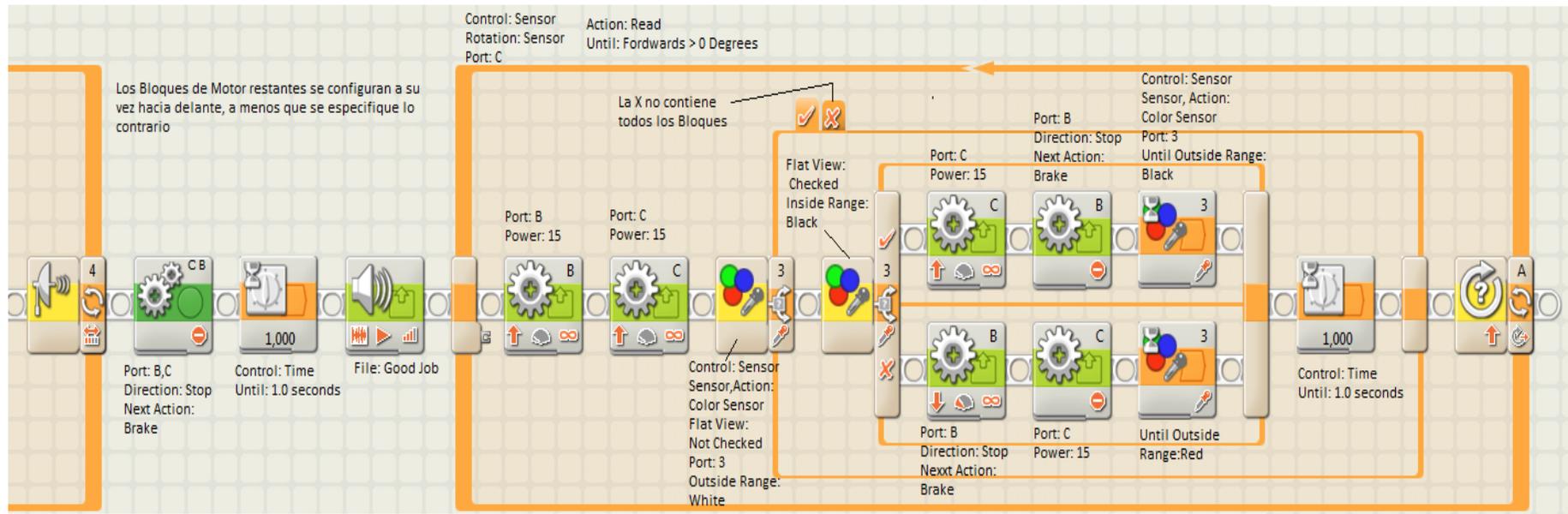


Figura 4. 43: Programación y configuración de los bloques del paso 3.

Fuente: Los autores

4.4. PRÁCTICA 4: Brazo robótico autónomo

En las prácticas anteriores se enseñaron acerca de la programación de robots NXT. Ahora que hemos alcanzado un nivel avanzado de la programación por bloques, estaremos listos para construir algunos robots más complicados como el de la práctica 4. Esta práctica aprenderá a construir un brazo robótico autónomo que puede encontrar y recoger objetos, como se muestra en la Figura 4.44.



Figura 4. 44: Brazo robótico modo autónomo.
Fuente: Los autores

El robot utiliza dos motores NXT para controlar un conjunto de bandas de rodadura, permitiendo que el robot se mueva en cualquier dirección y controlando su movimiento, mediante el ajuste de la potencia. Un brazo robótico normal puede agarrar y levantar objetos a través de dos motores: uno para agarrar el objeto y otro para levantarlo. Pero este brazo robótico requiere de un motor (motor de mano) para llevar a cabo ambas tareas, a causa de una construcción única de vigas, ejes y engranajes del LEGO Mindstorms NXT.

La figura 4.45 muestra el mecanismo de agarrar objetos. A medida que el motor gira hacia adelante, una marcha pequeña (se indica con un número 1) tiene un engranaje más grande (2) gira en el sentido indicado por la flecha.

Esta rotación inicia una reacción en cadena de las vigas en movimiento, lo que finalmente hace que el brazo robótico puede captar objetos situados entre los dedos (6). Cuando el motor gira al revés, ocurre lo contrario, y se abre la pinza. Las vigas de marcado 3, 4, 5 y simplemente transfieren el movimiento de rotación del motor de la pinza para que pueda cerrar sus garras.

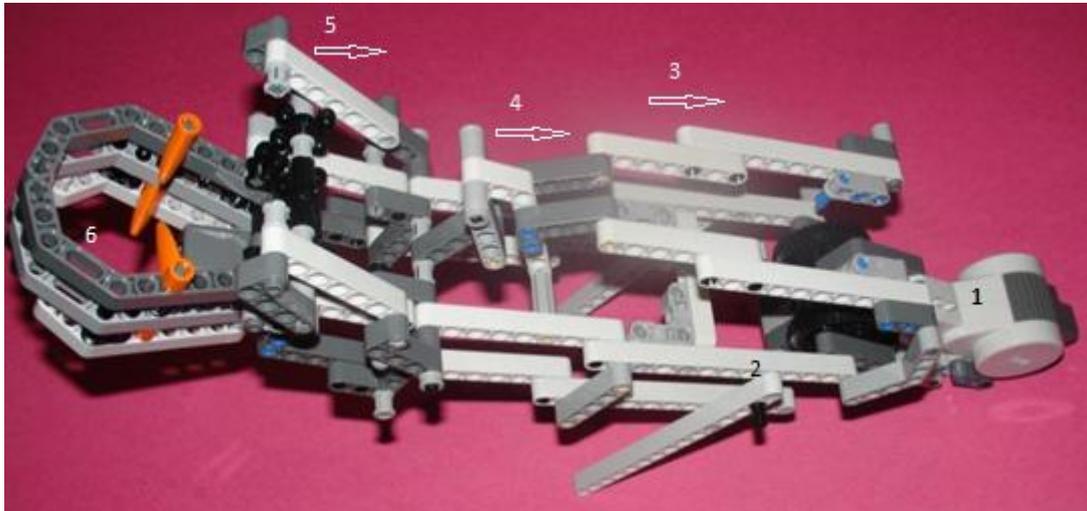


Figura 4. 45: Mecanismo de funcionamiento del brazo robótico modo autónomo.
Fuente: Los autores

Para el mecanismo de elevación, una vez que el robot agarra un objeto, lo puede inmediatamente levantar. Pero antes de ver cómo el robot lo hace, podríamos tener una versión simplificada de la situación. No importa que el movimiento de estas partes sea paralelo al suelo. Para entender realmente cómo funciona esto, construiremos la estructura mecánica con las piezas (ver figura 4.46) del LEGO Mindstorms NXT.



Figura 4. 46: Piezas para construcción del brazo robótico modo autónomo.
Fuente: Los autores

Podríamos modificar el brazo robótico para coger casi cualquier cosa, como también que el objeto no sea demasiado pesada, pero la versión que se construyó está diseñado para recoger anillos de papel en diferentes colores (amarillo, azul, rojo y verde) antes de programar el robot.

Después de haber construido el ladrón y creados los objetos (anillos), procederemos a la programación del brazo robótico que permite al mismo encontrar, agarrar, levantar y mover un objeto, así como identificar el color del objeto. Cada tarea debe ejecutarse de forma autónoma, lo que significa que todas deben ser realizadas sin intervención humana.

Para cumplir con lo planteado anteriormente usaremos el sensor de ultrasonido para localizar objetos, con la ayuda de los motores que se accionan una vez que se acercaron al objeto y agarrarlo entre los dedos del robot. Asimismo es necesario la creación de varios bloques debido su complejidad, obteniendo un programa avanzado para el brazo robótico de 5 bloques para que sea fácil de entender.

Bloque 1: Agarrar objetos.

Este bloque permite al robot agarrar y levantar un objeto colocado entre los dedos del robot, si no hay nada que agarrar se cierran los dedos del robot. Configurar los motores para continuar hacia delante, hasta que el sensor de contacto detecta que se ha levanta todo.

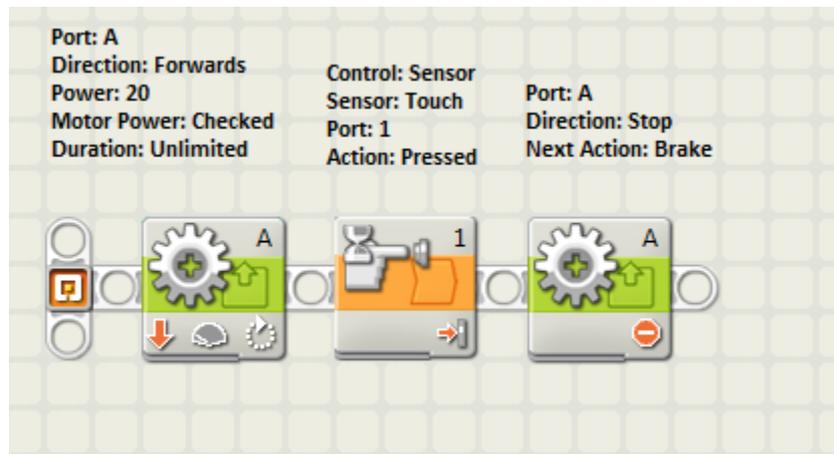


Figura 4. 47: Programación y configuración del brazo robótico para agarrar objetos
Fuente: Los autores

Bloque 2: Liberar objetos

Para este bloque de programación hay que disminuir la fuerza empleada en la mano del brazo robótico para que pueda abrir sus garras y liberar el objeto, para esta acción el programa principal se configura como se muestra en la figura 4.48, y luego lo convierte en un bloque llamado "**Mi versión**".

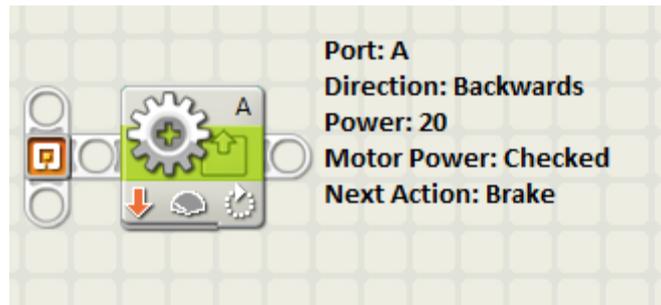


Figura 4. 48: Programación y configuración del brazo robótico para liberar objetos
Fuente: Los autores

Bloque 3: Búsqueda del objeto

A medida que el robot gira en 180°, el sensor ultrasónico mide constantemente las distancias. El valor más bajo que registra es la distancia al objeto, y lo almacena en la memoria del NXT. Se establece un valor máximo de 256 cm, ya que el sensor no puede medir nada más lejos. Cada vez que el sensor mide una distancia menor, el valor antiguo es borrado de la memoria y la nueva medida del sensor se guarda.

Los valores de los sensores de rotación en los motores cambian a medida que el robot gira. Debido a que el robot tiene que recordar donde vio el objeto más cercano, almacena el valor del sensor de rotación del motor C en una dirección siempre que el valor más cercano se actualiza con un nuevo valor. Al final, la variable de dirección debe contener el valor del sensor de rotación, cuando el robot vio el objeto más cercano.

Después de dar vuelta de 180°, el brazo robótico de gira a la izquierda y se detiene cuando el valor del sensor de rotación es igual a la variable de dirección. Se detiene el robot cuando apunta hacia el objeto y está listo para acercarse y cogerlo. Para aquello definiremos dos variables llamadas “más cercano” y “dirección” (variables numéricas). Para programar configuraremos los bloques para encontrar objetos como se muestran en las figuras 4.49 y 4.50, y luego convertirlo en un solo bloque.

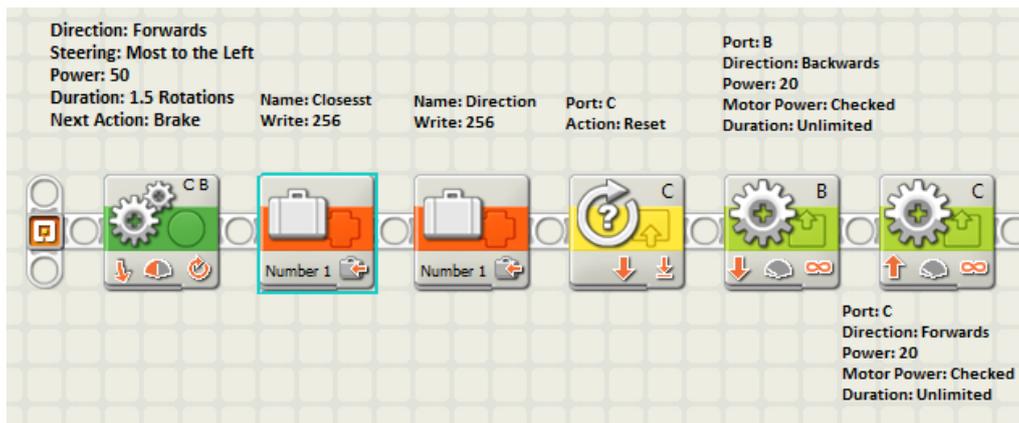


Figura 4. 49: Programación y configuración del brazo robótico para encontrar objetos.
Fuente: Los autores

La figura 4.49 permite al brazo robótico girar a la izquierda, inicializa las variables, restablece el sensor de rotación en el motor C. y comienza a girar a la derecha.

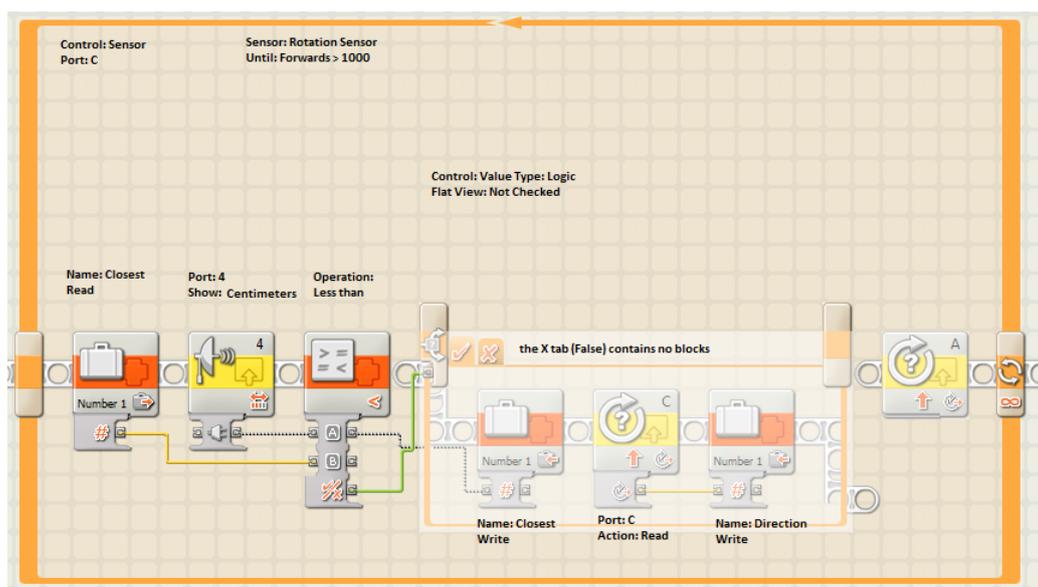


Figura 4. 50: Programación y configuración del brazo robótico para liberar objetos
Fuente: Los autores

Bloque 4: Acercándose

Este bloque hace que el brazo robótico se acerque al objeto encontrado, basado en la distancia medida previamente (ver figura 4.51). Si el objeto está lejos, el brazo se mueve hasta la distancia permitida buscando al objeto más cercano. Para lograr esto, creamos un bloque llamado *multiplicar* que permita

multiplicar el valor de las variables más cercano por 45° y transfiere el resultado a la configuración de duración del bloque de movimiento. Por ejemplo, una distancia mínima registrada de 10 cm enfocada por el robot, dando como resultado 450° de rotación del motor, mientras que una medida de 5 cm haría 225°.

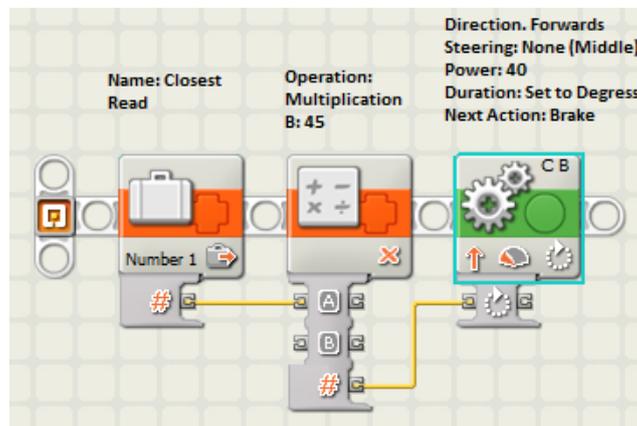


Figura 4. 51: Programación y configuración del brazo robótico acercándose a objetos
Fuente: Los autores

Bloque 5: Indicar color del objeto

Este bloque simplemente reproduce un archivo de sonido (como "Red") basado en el color del objeto detectado. El diseño es fácil siguiendo las instrucciones de la página web de LEGO.

Programa final

Una vez terminado los bloques (5) denominado *My Blocks*, continuamos con la creación del programa final (ver figura 4.52) para hacer que el brazo robótico encuentre de forma autónoma agarrando, levantando y moviendo un objeto.

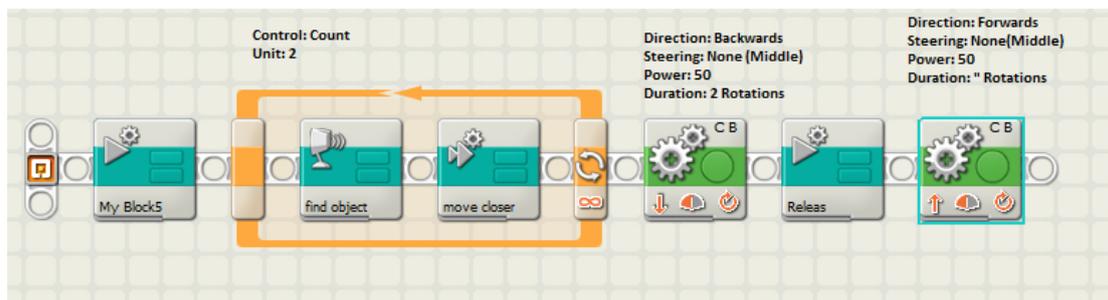


Figura 4. 52: Programa final para la operación completa del brazo robótico.
Fuente: Los autores

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES

- A mediados del 2006, LEGO lanzó al mercado el Mindstorms NXT, que reemplazo a su predecesor el Mindstorms RCX que en 1998 se mantuvo por 8 años en el mercado. El Mindstorms RCX fue el resultado de años de investigación (aproximadamente 5 décadas) en técnicas de construcción modular y electrónica, basados en la experiencia de métodos de educación por computador del Instituto de Tecnología de Massachusetts.
- A pesar de que la plataforma Mindstorms NXT fue diseñada para niños, fueron los adultos, estudiantes y profesionales de Ingeniería Electrónica, Control y Ciencias de la Computación, los que hicieron de esta plataforma un éxito total, gracias a la colaboración e investigación de un sinnúmero de investigadores alrededor del mundo.
- El desarrollo del documento del proyecto de grado fue obtenido gracias a la documentación existente en internet, publicaciones de artículos científicos y de proyectos de graduación de universidades del exterior y otras del Ecuador y por la misma comunidad de robótica de LEGO que no se detiene en ningún momento.
- En cuanto al software que utiliza el Mindstorms NXT tuvimos contactos por correo electrónico y de foros de la comunidad de robótica LEGO, donde se lograron aclarar dudas respecto a los alcances de determinadas interfaces de programación con el bloque programable.
- La programación que se desarrollo en el presente proyecto fue diseñada por nosotros, ya que la plataforma no viene incluido los programas ni mucho menos están armados o contruidos los robots, ya que la idea principal es incentivar la creatividad de los alumnos de la Facultad de Educación Técnica para el Desarrollo, para futuras investigaciones.

- Existieron pequeños problemas que se encontraron a medida en que se avanzaba en el desarrollo del proyecto de investigación en cuanto a la compatibilidad del hardware y software, resultando en ocasiones engorrosas, debido a que teníamos que conseguir drivers especiales para las computadoras portátiles y la información para solucionar dichos problemas fue escasa.

5.2. RECOMENDACIONES

- Promover talleres o cursos de capacitación en robótica orientado a los alumnos que recién inicien sus estudios en la FETD para promover en ellos la investigación.
- Promover concursos de robótica internos en la FETD y del Club de Robótica en la UCSG para asociarse a la Federación Ecuatoriana de Robótica (FER).
- Motivar a los estudiantes y docentes de la FETD el desarrollo y la investigación de aplicaciones del Lego NXT para futuros proyectos integrando nuevas tendencias tecnológicas para encontrar soluciones a las mismas, además de ser una de las líneas de investigación que fomenta el actual Gobierno Ecuatoriano, en cuanto a las TIC's.
- Sugerir a la FETD a través de sus autoridades académicas la creación de la materia de Fundamentos de Robótica en la carrera de Ingeniería en Telecomunicaciones y lo mismo para Ingeniería en Electrónica en Control y Automatismo pero adicional una materia llamada Robótica avanzada.

REFERENCIAS BIBLIOGRÁFICAS

[Atmatzidou, 2008] Atmatzidou, S., Markelis, I., Demetriadis, S., "The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning", Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, Venice (Italy) 2008 November,3-4 pp. 22-30.

[Bagnall, 2007] Bagnall Brian, "Maxium LEGO NXT, Building Robots with Java Brains. Variant Press, 2007.

[Butler, 2001] Butler, D., Martin, F., "Learning with LEGO Mindstorms in Irish Primary Education", The Spring Symposium of the American Association for Artificial Intelligence (AAAI), March 2001, Stanford, California USA

[Gandy, 2010] Gandy, E. A. "The Use of LEGO MINDSTORMS NXT Robots in the Teaching of Introductory Java Programming to Undergraduate Students", Disponible online en:
<http://www.ics.heacademy.ac.uk/italics/vol9iss1/pdf/paper01.pdf>

[Gómez, 2010] Gómez-de-Gabriel, J. M., Mandow, A., Fernández-Lozano, J., García-Cerezo, A., "Using LEGO NXT Mobile Robots With LabVIEW for Undergraduate Courses on Mechatronics", IEEE Transactions on Education, Disponible on-line:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5430870>

[IEEE Robotics, 2007] IEEE Robotics and Automation Society, extraído en abril 2011 de la página web: <http://www.ieee-ras.org/>

[Lew, 2010] Lew, M., Horton, T., and Sherriff, M., "Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course." *The 23rd Annual IEEE-CS Conference on Software Engineering Education and Training*, Pittsburg, PA, Mar 9-12, 2010.

[Moundridou, 2008] Maria Moundridou, Alexander Kalinoglou, “Using LEGO Mindstorms as an Instructional Aid in Technical and Vocational Secondary Education: Experiences from an Empirical Case Study”, Conference: European Conference on Technology Enhanced Learning - ECTEL , pp. 312-321, 2008.

[Santos, 2004] J. Santos and R. J. Duro, Evolución artificial y robótica autónoma: Editorial RAMA, 2004.

Anexo A