



CAPITULO 3

LEGO MINDSTORMS Y leJOS

Podemos decir que este trabajo está conformado por dos elementos principales: el robot construido con piezas del kit de Lego Mindstorms, y el programa desarrollado para dicho robot en la plataforma de leJOS (Java). Es por eso que en este capítulo nos enfocamos en presentar una visión general tanto de los Lego Mindstorms como de leJOS, para que teniendo una idea de estos dos conceptos tan importantes en el marco de esta tesis podamos abordar el capítulo de la implementación realizada.

3.1 LEGO MINDSTORMS

Aunque parecieran ser simples juguetes para que los adultos puedan tener entretenidos a los niños cuando parece que nada puede calmarlos, este kit de LEGO para los “inventores” e ingenieros puede ser una herramienta poderosa y muy práctica al momento de desarrollar y crear prototipos. Dependiendo del punto de vista en el que nos queramos ubicar, este kit contiene las herramientas suficientes para poder representar muchas cosas que vemos a nuestro alrededor desde una simple silla o un escritorio (obviamente a escala) hasta un reloj o una licuadora. Fuera de esta visión el kit incluso pudiera servir para representar modelos de carros, trenes, botes, etc. Si quisiéramos hablar de la naturaleza el kit es capaz de crear simulaciones de algunos animales, por decir algo una araña o el comportamiento de un perro o gato. Cómo podemos darnos cuenta nos

brinda las herramientas necesarias para probar y construir una gran variedad de proyectos que, en un futuro, se pueden desarrollar de forma más seria.

3.1.1 INTRODUCCIÓN A LEGO MINDSTORMS

El concepto de los MINDSTORMS fue inicialmente desarrollado en conjunto tanto por el LEGO Group como por el MIT (Massachusetts Institute of Technology). Dentro del MIT se estaba desarrollando un proyecto llamado Ladrillo Programable (“*Programmable Brick*”), este proyecto se inició alrededor del año de 1987 y originalmente el Grupo LEGO era un patrocinador del proyecto. El principal impulsor de este proyecto fue Fred G. Martin.^[Bagnall, 2002] El resultado obtenido de este proyecto es el MIT Programmable Brick, una computadora pequeña y portable que se encuentra dentro de un bloque (*brick*) de Lego y que es capaz de interactuar con su medio gracias a sensores y motores que se conectan al bloque.^[WEB 7]

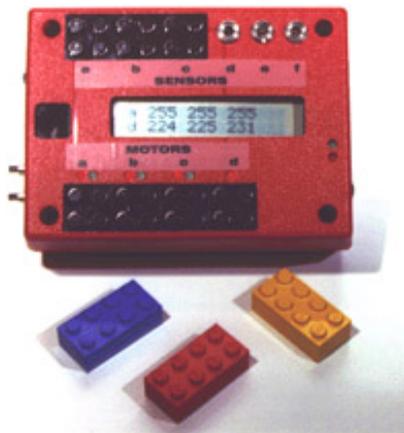


Imagen 3.1 Ladrillo Programable del MIT y el LEGO RCX desarrollado por el Grupo Lego.^[IMG 12]



Podemos considerar este proyecto como el antecesor del concepto actual de los LEGO MINDSTORMS y más específicamente como el concepto inicial del RCX, el cuál es el cerebro y control de cualquier creación que hagamos con nuestro kit. Cabe señalar que el Grupo LEGO desarrolló su propia tecnología diferente a la utilizada en el proyecto del MIT. ^[WEB 8] Hoy en día todos los productos que la compañía diseñó para crear robots controlados por computadora se engloban dentro del concepto de MINDSTORMS y como parte de esta familia de productos se encuentra el RIS (Robotics Invention System) que es el kit con el cual se trabajó en este proyecto.

3.1.2 PARTES DEL KIT.

BLOQUE RCX.

Podemos considerarlo como el corazón de los MINDSTORMS y su nombre proviene de las iniciales de Robotic Command eXplorer (RCX). ^[Bagnall, 2002] Al decir que es el corazón es porque este bloque es el que le da al robot la capacidad de poderse mover sin necesidad de estar conectado a la computadora o a una corriente eléctrica. Las dimensiones del RCX son 6cm X 10cm X 4cm, utiliza 6 pilas AA y su peso, con pilas incluidas es de aproximadamente 280grs. Dentro del bloque encontramos el procesador, que es un HITACHI H8/3292, capaz de correr entre 10MHz y 16MHz. ^[Bagnall, 2002] El RCX tiene una capacidad de 32KB que principalmente almacenan dos cosas: el *firmware* y los programas que nosotros creamos. El *firmware* se refiere básicamente al “sistema operativo” que vayamos a utilizar. Dave Baum ^[Baum, 2000] define el *firmware* como un término genérico para describir el software construido dentro de un dispositivo; el término de *firmware* pretende definir algo entre el hardware y la noción común de software. Cuando adquirimos el kit éste trae un CD con el software de LEGO, el RCX 2.0 Software Development Kit



(SDK). [Bagnall, 2002] Dentro de este software se encuentra el *firmware* original de LEGO, el cual nos permite programar al robot en el ambiente incluido en el software, este lenguaje llamado Mindscript se compone de sencillas instrucciones previamente programadas y lo que el usuario se limita a hacer es a arrastrar bloques para crear una secuencia de movimientos. Actualmente se han desarrollado varios *firmwares* para programar el RCX dentro de los cuales está incluido leJOS (Lego Java Operating Systems), Not Quite C (NQC) desarrollado por Dave Baum [Baum, 2000] basado en lenguaje C, legOS que es un sistema operativo independiente y de libre uso para LEGO Minsdtorms y, aunque menos conocidas también se han hecho implementaciones en Ada.

En el RCX se encuentran concentradas todas las funcionalidades del robot, ya que aquí es donde encontramos tanto los puertos de entrada como los puertos de salida de datos, por llamarles de alguna manera, así como también alberga el puerto de comunicación infrarrojo, la pantalla de información LCD, un altavoz y los botones de control. Los puertos de entrada se refieren básicamente a los tres puertos en los cuales se pueden conectar una gran variedad de sensores LEGO, como lo son el sensor de tacto o de luz, incluidos en el kit básico. Gracias a estos puertos el robot obtiene datos del exterior que puede usar para modificar su comportamiento. Los puertos de salida son usados para controlar los llamados “actuadores” o dispositivos controlables. El kit incluye dos motores que pueden conectarse a cualquier de estos tres puertos, y ya uno es libre de decidir el diseño para que el robot se mueva, ya sean llantas, orugas o incluso piernas.



Imagen 3.2 RCX. ^[IMG 13]

El puerto infrarrojo permite, como tarea principal, que se puedan descargar los programas al RCX, además de esto permite la transmisión de datos entre la computadora y el robot, o incluso entre dos robots. La pantalla de información ó LCD como también se conoce es solo una pequeña parte del RCX pero resulta muy útil al momento de probar nuestros programas. Aquí podemos consultar por ejemplo los valores que esté leyendo un sensor que esté conectado. Además de que la pantalla incluye algunos símbolos que nos indican si tiene almacenado, o si está ejecutando algún programa actualmente, además de que posee unos indicadores para cuando un programa está siendo descargado. El pequeño altavoz no es visible desde el exterior, pero se localiza detrás del puerto infrarrojo y permite al robot reproducir una variedad de sonidos de diferentes frecuencias.

Por último tenemos los botones que componen al RCX que son cuatro. El primero es el rojo de “On/Off”, que prende y apaga el RCX. Después tenemos el verde de “Run” que empieza a correr el programa actual. El botón negro de “View” permite ver las lecturas de los sensores conectados al RCX, siempre y cuando tengamos instalado el *firmware* original de LEGO. Por último el botón gris de “Program” permite seleccionar cual programa queremos correr, con el *firmware* original se pueden almacenar hasta 5 programas al mismo tiempo.



TORRE INFRAROJA (IR TOWER).

Esta torre permite la comunicación entre la computadora y el robot, es la encargada de mandar todos los datos que deseamos transferir al RCX. Pero no solo cumple esta importante función sino que si lo necesitamos, se pueden diseñar programas en los cuales el robot mande datos a la computadora para que ésta los procese y de ser necesario, los regrese al robot. Existen dos tipos de torre, una de conexión por puerto serial y actualmente la que incluye el kit que es por conexión USB.



Imagen 3.3 Torre Infrarroja. [IMG 13]

MOTORES.

El kit incluye dos motores de 9 volts. Estos se conectan al RCX y dependiendo en la forma en que conectamos el cable se define si el motor gira en el sentido o en contra de las manecillas del reloj. Cuando se programa el robot es posible definir, es una escala del 0 al 9, la potencia que esperamos de los motores. Esto es importante porque es muy común que los motores giren con potencias diferentes, lo cual se ve reflejado al momento que el robot avanza, si por ejemplo trazamos una guía en el suelo y esperamos que el robot avance

en línea recta pero un motor gira más que otro, entonces el robot empezará a avanzar pero comenzará a desviarse hacia el lado derecho o izquierdo de la guía, dependiendo del motor que gire menos.

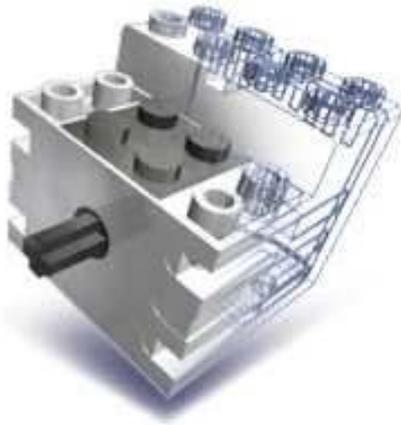


Imagen 3.4 Diseño del motor. ^[IMG 13]

SENSORES.

Dentro del kit vienen incluidos tres sensores: dos de tacto y uno de luz. Los sensores son muy útiles y son una fuente de información extraordinaria para el robot, porque le permiten obtener datos del mundo exterior en base a los cuales podrá definir cuál es el próximo movimiento a hacer. Los movimientos de tacto son los más sencillos de utilizar pues solo tienen dos estados posibles: presionado y liberado. Por otro lado el sensor de luz es útil pero es más difícil de usar y de medir sus datos, además de que requiere que se active para que el RCX le mande señal y empiece a trabajar.

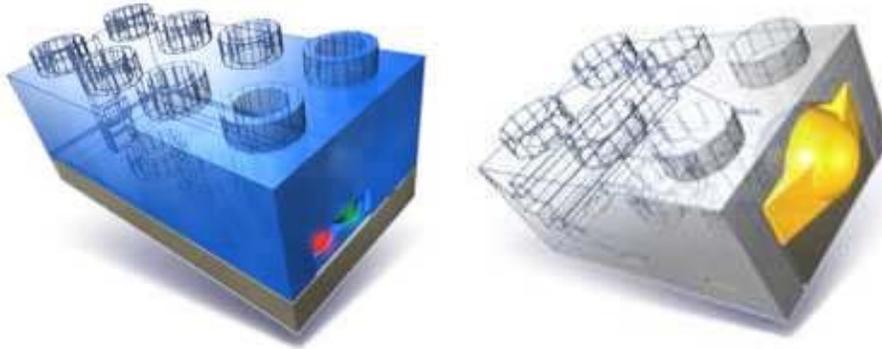


Imagen 3.5 Sensores incluidos en el kit: luz y tacto (2) ^[IMG 13]

No incluidos en este kit pero que se pueden adquirir en otros paquetes se encuentra la LEGOCam, y el sensor de rotación, capaz de contar el número de giros de la pieza que tenga conectada. De igual manera se pueden adquirir más sensores de tacto y luz.

PIEZAS LEGO.

El kit incluye más de 500 piezas de LEGO que consiste, entre otras cosas de los bloques clásicos y algunas piezas más avanzadas de LEGO Technic. El kit también incluye bloques especiales como vigas, esquinas, bloques inclinados, bloques planos de varios tamaños, desde 1x1 hasta 6x10 (tres bloques planos dan la altura de un bloque básico), se incluyen varias tipos de conectores (*pins*) que permiten unir vigas y, dependiendo del tipo de conector usado, permitir que se muevan o simplemente las asegura. También se incluye una variedad de engranes de diferentes tamaños y cantidad de dientes que pueden usarse para controlar la potencia del robot. El kit también nos ofrece una variedad de 8 diferentes tipos de llantas, desde ruedas anchas para construir un carrito típico hasta las orugas para construir automóviles “todo terreno”. Se incluyen varias piezas con formas de brazos que



usualmente se utilizan para construir las defensas (*bumpers*). Este es solo un ejemplo de algunas de las principales piezas que se incluyen en el kit.

Como mencionamos con anterioridad también es posible adquirir kits para aumentar la cantidad y tipo de piezas y así construir proyectos más grandes si es necesario.

CD CON EL SOFTWARE OFICIAL DE LEGO.

El kit viene con un CD donde se encuentra todo el software desarrollado por LEGO para echar a andar el kit y desarrollar una gran variedad de proyectos. En el disco se encuentra el ambiente de trabajo ó programación desarrollado por LEGO, el cual está lleno de colores y resulta muy fácil de usar, incluso para aquellas personas que no sepan sobre programación. También se incluye el *firmware* de LEGO y los controladores necesarios para que la torre responda adecuadamente.

3.1.3 VENTAJAS Y LIMITACIONES USANDO LEGO MINDSTOMRS

Hemos hablado sobre todos los componentes que se incluyen en el kit y también de las posibilidades de adquirir piezas extras en caso de ser necesario. Una ventaja que nos da el usar el kit de LEGO es que no hay que adquirir forzosamente los kits de expansión oficiales si lo único que necesitamos son piezas para construir estructuras más grandes o resistentes.

Podemos decir que no hay una regla o reglas exactas a seguir al momento de construir y de programar nuestros robots. Como dice un proverbio chino: “Dime y olvidaré, muéstrame y tal vez recuerde, involúcrame y entenderé”. Esto encaja perfecto con la concepción que LEGO tiene, cada uno es libre de armar el modelo como quiera y de



programarlo en la firma que mejor le venga en gana, lo único cierto es que, al involucrarnos en cada parte del diseño seguro aprenderemos y sobretodo entenderemos bien qué es lo que esperamos de nuestro robot. Esto es una gran ventaja porque podemos decir que prácticamente todos los proyectos realizados con el kit serán diferentes unos de otros y la posibilidad de crear cosas nuevas siempre está presente y el único límite es la propia imaginación de quien está construyendo.

Otro punto a favor trabajando con RIS es que por lo general los proyectos realizados no son muy voluminosos físicamente hablando, por lo cual es posible transportarlos de un lado a otro en caso de ser necesario, y por lo cual también los hace un producto con el cual sea agradable y fácil trabajar puesto que no es difícil llevarlo de un lado a otro, por ejemplo para trabajar con él en un salón de clases.

Las desventajas de trabajar con LEGO Mindstorms son varias y principalmente mencionaré las referentes al RCX y sus puertos para conectar los sensores. Para los propósitos de este trabajo hubo varios problemas a los que no se les pudo dar solución en gran parte porque no se contaba con la cantidad suficiente de puertos a los cuales conectar más sensores. En este caso no se necesitaban más de tres motores, de hecho solo dos, pero en el caso de querer construir o simular un brazo mecánico con varios grados de libertad serían necesarios más motores para una buena implementación, pero nos encontramos ante la limitación de solo tres puertos a los cuales conectar motores.

Una situación igual es a la que nos enfrentamos en el caso de querer dotar a un robot con más de tres sensores. Si bien es cierto que podemos conectar dos sensores sobrepuestos en un mismo puerto, en ocasiones es necesario identificar qué sensor está recibiendo los datos para así poder actuar de una cierta manera. De esta manera es muy probable que algunos proyectos se vean limitados en el hecho de ya no poder agregar más sensores al



robot y por lo mismo su funcionalidad es menor y es muy probable que el desempeño final del robot sea de menor calidad que lo esperado originalmente.

Aunque estas son algunas de las limitaciones principales es cierto que cada proyecto es diferente y al momento de su desarrollo se van encontrando más detalles que corregir o problemas que en ocasiones por limitaciones del kit no podrán ser resueltos pero, hablando de manera general podríamos decir que el kit nos proporciona herramientas valiosas y suficientes para poner en marcha una gran cantidad de ideas y crear nuevos diseños y proyectos; cuando es posible adquirir los paquetes de expansión obviamente se abren muchas más nuevas opciones de desarrollo. Para muchos desarrolladores e incluso aficionados a trabajar con LEGO Mindstorms esa es su gran ventaja: el no estar atados a reglas y estructuras que desarrollar, sino la libertad de crear lo que se quiera o necesite y claro está, cómo uno lo quiera hacer, en realidad la única limitante está en la capacidad de imaginar nuevas ideas que desarrollar.

3.2 leJOS

Cómo se puede observar el RIS es una maravillosa herramienta para desarrollar nuestros robots y proyectos, pero para muchos programadores, especialmente aquellos quienes gustan de programar orientado a objetos puede ser muy probable que el ambiente de programación incluido en el kit no les resulte de todo muy agradable para desarrollar sus aplicaciones. En el resto de este capítulo hablaremos sobre otro elemento de gran importancia para el desarrollo de este proyecto: leJOS, un proyecto accesible para todos y que permite hacer programas modulares y bien organizados en Java.



3.2.1 INTRODUCCIÓN A leJOS

leJOS es un *firmware* que reemplaza el original desarrollado por Lego para el bloque Lego Mindstorms RCX y que permite programar los robots en Java. leJOS es un proyecto de *libre uso* (Open Source) comenzado a desarrollar en un principio por José Solórzano y Brian Bagnall. El antecesor a este proyecto que ahora es ampliamente usado fue TinyVM ^[WEB 9], un *firmware* que de igual manera reemplaza al original y que también permite programar en Java pero con más limitaciones que leJOS, por ejemplo no soporta el uso de números flotantes y no se puede usar la estructura ‘switch’.

leJOS, al igual que el J2SDK de Sun Microsystems está disponible en la red para ser descargado de manera gratuita y de igual manera utiliza un compilador para transformar el código fuente en instrucciones byte-code, de hecho cuando ejecutamos leJOS éste manda llamar el comando *javac* para producir el byte-code. leJOS incluye su JVM (Java Virtual Machine) para leer el byte-code generado y de esta manera ejecutar las instrucciones en el RCX; también cuenta con su API (Application Programming Interface) que está formado de varias clases que pueden ser implementadas por los programadores para darle mayor funcionalidad a sus programas. Aunque suene un poco impresionante cabe mencionar que todas estas funciones de leJOS tuvieron que ser implementadas para correr en una computadora con tan solo 32kB de memoria RAM, que es el espacio con el que cuenta el RCX.

3.2.2 ELEMENTOS RELEVANTES DE leJOS

JVM (Java Virtual Machine)

Podemos decir que la JVM es en efecto, el *firmware*, ya que es lo primero que debemos descargar al bloque RCX para posteriormente poder introducir nuestros



programas en Java. La JVM es la que sabe cómo interpretar el byte-code que descargamos al RCX para después comunicarle al RCX qué es lo que debe de hacer. ^[Bagnall, 2002] La JVM tiene algunos detalles técnicos que se especifican a continuación:

MEMORIA.

Cómo habíamos mencionado anteriormente el RCX tiene una memoria RAM total de 32kB, de los cuales 4kB están reservados para que el ROM ejecute sus rutinas. Esto deja libre 28kB de los cuales leJOS JVM ocupa 16kB dejando 12kB libres para los programas del usuario, como podemos observar en la figura 3.1. Aunque suene poca memoria por lo general esta suele ser más que suficiente para desarrollar buenas aplicaciones, a menos claro está que se estén recolectando grandes cantidades de datos (mapas de datos).

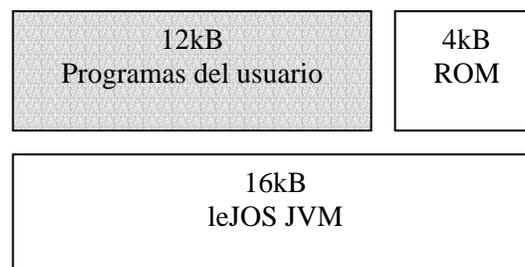


Figura 3.1 Distribución de la memoria en el RCX. ^[Elaboración Propia]

NÚMEROS FLOTANTES.

Actualmente leJOS es el único lenguaje para el RCX que permite usar números flotantes (decimales). Esto le da a leJOS la habilidad de representar números fraccionarios y calcular funciones trigonométricas como el seno, coseno y la tangente, las cuáles son importantes para que las funciones de navegación del robot sean más exactas.



HILOS (THREADS).

La programación multi-hilos permite que diferentes partes de un programa (los hilos) se ejecuten prácticamente al mismo tiempo compartiendo los recursos del CPU. En leJOS se ha implementado esta funcionalidad de una manera bastante completa, permitiendo la sincronización e interrupción de hilos. En leJOS los hilos se ejecutan por turnos, permitiéndole a cada hilo ejecutar aproximadamente 128 instrucciones antes de ceder el turno al siguiente hilo. Todas estas acciones son controladas por un sencillo planificador (*scheduler*) que es quien hace de manera automática el cambio de un hilo a otro.

ARREGLOS.

Los arreglos son estructuras de datos muy útiles para almacenar largas secuencias de números o de objetos. Los arreglos fueron implementados en leJOS y no solo los arreglos unidimensionales sino también los multidimensionales, lo cual puede ser usado para los sistemas de navegación que necesitan ir guardando las coordenadas que han ido recorriendo. No es muy recomendable el utilizar arreglos de más de 2 dimensiones puesto que el espacio de memoria libre como ya vimos son solo 12kB y si nos excedemos guardando datos se puede provocar el que sistema se quede sin recursos y se colapse.

RECURSIÓN.

La recursión es una técnica de programación que le permite a un método llamarse a sí mismo. La mayoría de los lenguajes de programación orientados a objetos permiten la recursión y no tienen un número limitado de iteraciones posibles. leJOS ha implementado la recursión pero con la limitante de que se pueden hacer máximo 10 recursiones,



dependiendo del número de variables que utilice el método que se está mandando a llamar, pues si éste utiliza demasiada variables el número de recursos libres se ve reducido y por consecuencia el número de repeticiones también.

RECOLECTOR DE BASURA.

Cómo ya sabemos Java tiene implementado un recolector de basura (Garbage Collection) el cual va liberando recursos al sistema cada vez que un objeto ya no va a ser usado más. Sin embargo esta funcionalidad aún no ha sido implementada en leJOS, pero esto no significa que el funcionamiento del programa se vaya a ver afectado, siempre y cuando cómo programador se tenga en mente el reutilizar objetos en lugar de crear nuevos objetos.

3.2.3 POR QUÉ SE USÓ leJOS

Cómo hemos explicado en los puntos anteriores leJOS se ha convertido en una alternativa de programación para los LEGO Mindstorms bastante viable y completa, y cada día son más los que la utilizan para desarrollar sus proyectos. El éxito de esto se puede deber a que utiliza uno de los lenguajes más poderosos de hoy en día: Java. Para la mayoría de los programadores experimentados el trabajar con el ambiente de programación incluido en el kit de RIS podrá resultar hasta cierto punto limitante, y la posibilidad de desarrollar código para el robot basado en Java es mucho más atractiva.

En este caso se decidió desarrollar el software en leJOS porque está basado en la tecnología de Java, la cual fue aprendida y usada durante gran parte del plan de estudios. Además el programar usando leJOS nos dio como resultado un código más ordenado y



estructurado ya que fue modelado por *Comportamientos*, y de esta manera resulta más fácil agregar, quitar o modificar las reacciones del robot.

El API de leJOS fue un recurso también utilizado pues sirve como una buena referencia para usar los métodos que ya han sido implementados, leJOS posee varias implementaciones muy útiles, por ejemplo de Navegación y Comportamiento, lo que facilita el trabajo y da como resultado final un código modular.