



**UNIVERSIDAD POLITÉCNICA DE PUEBLA**

**PROGRAMA ACADÉMICO DE  
INGENIERÍA EN INFORMÁTICA**

# **Reconstrucción de Ambientes en Dos Dimensiones a través de un Robot Tipo LEGO**

*Erika Cruz González*

Reporte Técnico PII-20-08-09

## **COMITÉ EVALUADOR**

M.C. Javier Caldera Miguel (*Asesor*)  
C.Dr. Javier Velásquez Sandoval (*Sinodal*)  
Dr. Jorge de la Calleja Mora (*Sinodal*)

## ***PROFESORA DE PROYECTO DE INVESTIGACIÓN II***

Dra. María Auxilio Medina Nieto

Juan C. Bonilla, Puebla  
Agosto 2009

## ÍNDICE

### Capítulo 1. Reconstrucción de ambientes en dos dimensiones

1.1	Introducción.....	11
1.2	Objetivo general.....	12
1.3	Objetivos específicos.....	12
1.4	Justificación.....	12
1.5	Cronograma de actividades.....	14
1.6	Alcances y limitaciones.....	14
1.7	Recursos de hardware y software.....	14

### Capítulo 2. Marco teórico

2.1	Robótica móvil.....	16
2.1.1	Dispositivos de movimiento.....	18
2.1.2	Dispositivos de adquisición de datos.....	18
2.1.3	Tipos de locomoción en robots móviles.....	19
2.2	Características de robot LEGO.....	21
2.3	Dispositivos de comunicación.....	22
2.4	Características de ambientes.....	24
2.5	Reconstrucción virtual de ambientes.....	24
2.5.1	Representación de la realidad.....	24
2.5.2	Etapas de un sistema de reconstrucción artificial.....	25
2.5.3	Método de reconstrucción GPS .....	26
2.6	Planificación de trayectorias.....	27
2.7	Aplicaciones de la reconstrucción de ambientes.....	28

### Capítulo 3. Metodología

3.1	Análisis de requerimientos del sistema .....	28
3.2	Situación actual .....	29
3.3	Requisitos.....	29
3.4	Requerimientos funcionales .....	29
3.5	Requerimientos no funcionales.....	31
3.6	Desarrollo del algoritmo de desplazamiento.....	32

<b>Capítulo 4. Implementación</b>	
4.1 Algoritmo de desplazamiento.....	34
4.1.1 Descripción de software y herramientas de LEGO.....	36
4.1.2 Calibración de sensores.....	38
4.1.3 Implementación del algoritmo de desplazamiento.....	41
4.2 Comunicación bluetooth entre el robot con la computadora.....	44
4.3 Implementación de la interfaz para pintar el mapa.....	45
<b>Capítulo 5. Pruebas de ambientes explorados</b>	
5.1 Imágenes de ambientes explorados.....	49
<b>Capítulo 6. Conclusiones.....</b>	<b>54</b>
<b>Referencias.....</b>	<b>65</b>

## ÍNDICE DE FIGURAS

### Capítulo 1

Figura 1.1. Cronograma de actividades Mayo-Diciembre 08 .....	13
---	----

### Capítulo 2

Figura 2.1 Sistema Ackerman.....	18
Figura 2.2. Locomoción con direccionamiento diferencial.....	18
Figura 2.3. Receptor IS1U60.....	23
Figura 2.4 Imagen en dos dimensiones.....	25
Figura 2.5. Etapas de reconstrucción.....	26

### Capítulo 3

Figura 3.1. Sensor ultrasónico.....	30
Figura 3.2. Sensor óptico.....	30

### Capítulo 4

Figura 4.1. Diagrama de flujo del algoritmo de desplazamiento.....	34
Figura 4.2 Pantalla inicial de LEGO Mindstorms Education NTX.....	35
Figura 4.3 Valores máximo y mínimo de la configuración del sensor óptico.....	37
Figura 4.4 Valores máximo y mínimo de la configuración del sensor ultrasónico.....	38
Figura 4.4 Implementación del paso 1 del algoritmo de desplazamiento.....	39
Figura 4.5 Configuración del interruptor.....	39
Figura 4.6 Implementación del paso 2 del algoritmo de desplazamiento.....	40
Figura 4.7 Implementación del paso 3 del algoritmo de desplazamiento.....	41
Figura 4.8 Implementación del paso 4 del algoritmo de desplazamiento.....	42
Figura 4.9 Implementación del paso 5 del algoritmo de desplazamiento.....	43
Figura 4.10 Comunicación entre IVT BlueSoleil y LEGO NTX.....	44
Figura 4.11 Colocación de herramientas en el formulario .....	45
Figura 4.12 Ejemplo de la implementación de la función pintar.....	47

### Capítulo 5

Figura 5.1. Reconstrucción del ambiente 1.....	48
Figura 5.2. Reconstrucción del ambiente 2 con una salida.....	49
Figura 5.3. Reconstrucción del ambiente 3 con una salida.....	49
Figura 5.4. Reconstrucción de ambiente 4.....	50

## **RESUMEN**

La reconstrucción de ambientes presentada en este trabajo consiste de la obtención de imágenes o mapas a partir de la exploración de un robot tipo LEGO Mindstorms NXT. Para la reconstrucción se realizó un algoritmo que permite al robot desplazarse por el ambiente, una conexión bluetooth entre la computadora y el robot y una interfaz para pintar los ambientes explorados. Los datos se obtienen de un sensor ultrasónico, se guardan en un archivo de texto, posteriormente se envían a la computadora y finalmente se grafican en dos dimensiones.

# Capítulo 1. Planteamiento del problema de investigación

## 1.1 Introducción

Un robot es una máquina autónoma, capaz de ejecutar diversas tareas, que pueden ir de las más simples hasta las más complejas. Los robots móviles tienen grandes aplicaciones tanto para la industria como para el campo científico. En la industria, un robot móvil pueda trasladar artefactos pesados de un lugar a otro. Dentro del campo científico, puede realizar exploraciones de áreas geográficas extensas en lugares donde el ser humano correría algún peligro o simplemente donde no le sería imposible permanecer. Por ejemplo, el robot “spirit” [Rivera 2008] realiza exploraciones en el planeta Marte.

Dentro del mundo de la robótica móvil existen diferentes tipos de robots. Para el desarrollo de este trabajo se han tomado en cuenta las características de desplazamiento y movilidad de los robots considerando tipos como rover, bípedo, bípedo dinámico, entre otros.

Para que un robot móvil pueda desplazarse, es necesario tener conocimiento del área donde se va a mover. Algunos robots móviles de exploración, cuentan con la información del lugar de desplazamiento (desplazamiento de ambientes conocidos). Los robots de exploración adquieren el conocimiento del ambiente en el que se están desplazando. Estos robots emplean sensores ópticos o ultrasónicos, cámaras fotográficas, entre otros [Modesti 2006]. Estos dispositivos se encargan de obtener la información del ambiente que el robot está explorando, partiendo de esta información, se realiza la reconstrucción del ambiente.

En el presente trabajo se desarrollará la reconstrucción de ambientes reales no conocidos por medio de la comunicación entre un robot tipo LEGO y una computadora.

## 1.2 Objetivo general

Reconstruir ambientes en dos dimensiones a través de un robot tipo LEGO.

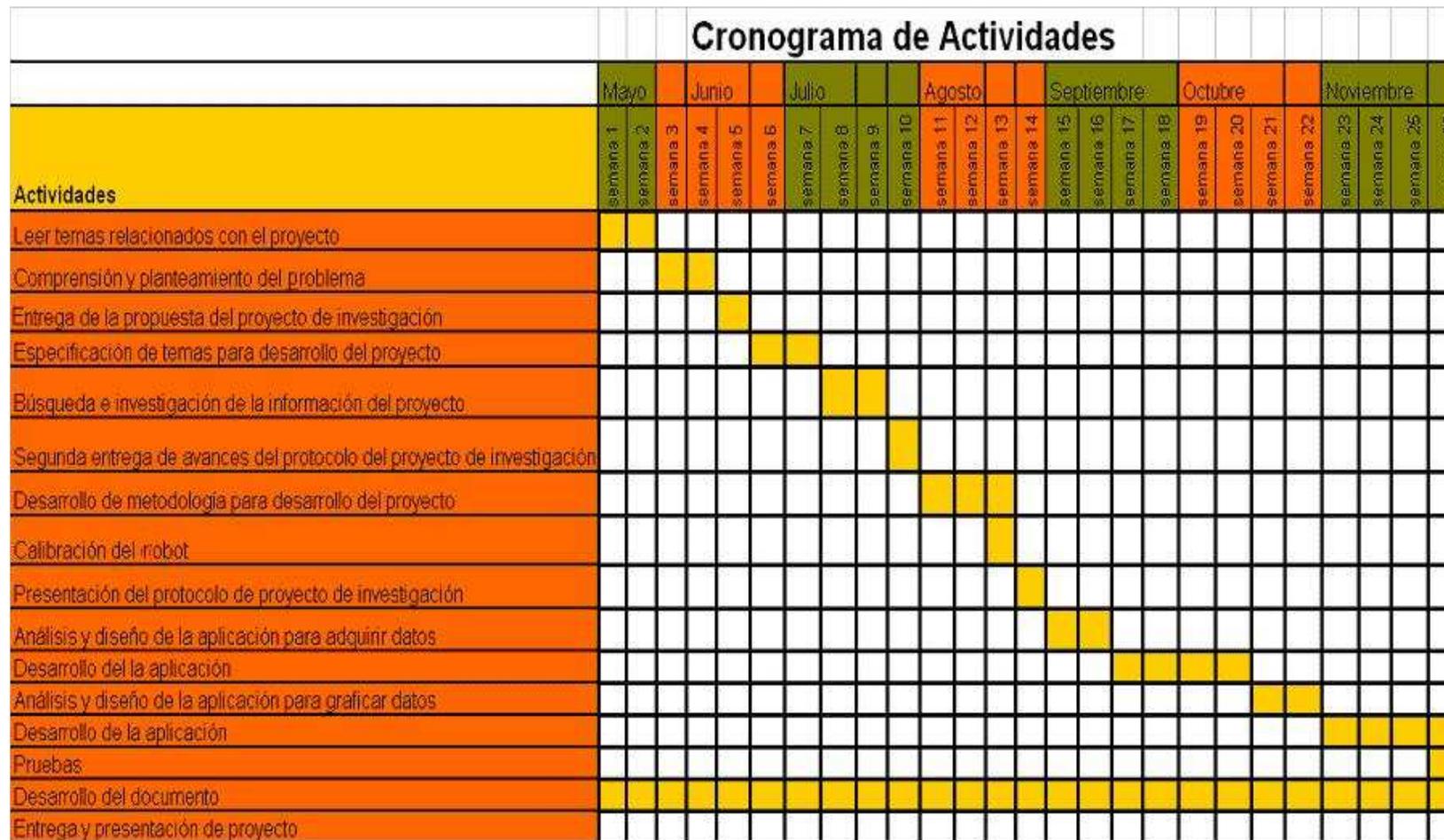
### **1.3 Objetivos específicos**

- Implementar un algoritmo que permita al robot desplazarse en un ambiente.
- Establecer una comunicación entre la computadora y el robot vía bluetooth
- Desarrollar una aplicación de software que grafique un mapa del ambiente explorado por el robot LEGO.

### **1.4 Justificación**

Algunos robots realizan la exploración de ambientes utilizando cámaras fotográficas y de video. Esta exploración puede ser costosa en tiempo y dinero, ya que el procesamiento de imágenes debe realizarse antes de la reconstrucción.

Existen otros dispositivos de adquisición de datos como sensores que, disminuyen el problema anterior y facilitan la reconstrucción de mapas X, Y, donde la planificación de trayectorias es fácil de implementar. La reconstrucción de ambientes por medio de sensores sólo se ha implementado en ambientes simulados o conocidos por el robot.



**Figura 1.1.** Cronograma de actividades Mayo-Diciembre 08

## **1.5 Cronograma de actividades**

Para el desarrollo de este proyecto se tendrán las siguientes actividades principales. El resto se muestra en la Figura 1.1.

1. Armar el robot LEGO
2. Calibrar los sensores de acuerdo al ambiente a explorar
3. Configurar los movimientos de los motores del robot
4. Calibrar los movimientos del robot
5. Realizar la conexión entre el robot LEGO y la computadora
6. Elaborar una tabla de comparación de distancias
7. Implementar la interfaz para pintar el mapa
8. Elaborar pruebas

## **1.6 Alcances y limitaciones**

- La reconstrucción del ambiente será en dos dimensiones
- Sólo se reconstruirán áreas cerradas.
- En el ambiente no se consideran tipos de obstáculos
- No se podrán visualizar los objetos en el mapa

## **1.7 Recursos de hardware y de software**

Robot: LEGO Mindstorms NXT

- Microcontrolador 32-bit ARM7
- 256 Kbytes FLASH, 64 Kbytes RAM
- Microcontrolador 8-bit AVR
- 4 Kbytes FLASH, 512 Byte RAM
- Comunicación Bluetooth wireless
- 6 Puertos USB (12 Mbit/s)
- 3 puertos de salida, 6-wire display gráfico 100 x 64 pixel LCD
- Fuente de poder 6 baterías AA

Emisor y receptor para conexión inalámbrica:

- Belkin F8T003 ver. 2
- Adaptador AVM BT
- Módulo bluetooth Dell TrueMobile
- Tarjeta interna Dell Wireless 350
- Adaptador USB Bluetooth v2.0

Servomotor

- Sensor de rotación
- Medidas de precisión de + / - un grado en motor rotacional
- Rotación de 360°

Sensores de sonido:

- Niveles de presión sonora de hasta 90 dB

Sensores ultrasónicos:

- Medir 0 a 255 centímetros con una precisión de + / - 3 cm.

Computadora:

- Disco duro de 40G
- Pentium 4
- Memoria 512

Software:

Windows XP

LEGO Mindstorms Education NTX

Delphi v.7

Block de Notas

## **Capítulo 2. Marco teórico**

### **2.1 Robótica móvil**

Los robots móviles surgieron con la necesidad de extender el campo de aplicación de la robótica. Los robots móviles tienen como precedentes los dispositivos electromagnéticos creados desde los años treinta. En los años setentas, se comenzó a trabajar en el desarrollo de robots móviles dotados de autonomía, la mayoría de ellos eran desarrollados en plataformas que soportan sistemas de visión. Sin embargo, el desplazamiento autónomo no era eficiente aún. En los ochentas, la autonomía de los robots móviles aumentó debido al incremento de la capacidad computacional, el desarrollo de nuevos sensores, mecanismos y sistemas de control [Ollero 2001].

El sistema de navegación es la base de la autonomía de un robot, estos sistemas se encargan de la planificación, percepción y control del robot.

#### **2.1.1 Dispositivo de movimiento**

Un servomotor se define según [Martín 2004] como un dispositivo en forma de caja que se encarga del movimiento de un robot el cual está compuesto por:

- Un motor pequeño
- Una caja de engranajes
- Un potenciómetro
- Un circuito integrado pequeño

Un servomotor controla:

- Cuerpo del robot
- El giro y posicionamiento del eje del motor
- El movimiento de rotación angular del motor
- La posición del potenciómetro interno

Los giros y posiciones del motor se controlan a través de señales. Los servomotores tienen tres cables: el de masa (-), el de alimentación (+) y el de la señal de control. Los cables de los servos siguen casi siempre el mismo código de colores: el color rojo (V+), negro (masa) y blanco (señal de control).

La velocidad del motor, así como la dirección del movimiento de los servos se controlan mediante servopulsos [Palacios 2006]. El servomotor convierte los servopulsos en un movimiento mecánico. Este tipo de pulsos está formado por una señal digital que se genera aproximadamente cada 20 milisegundos. La anchura de los pulsos va de un mínimo de 1 ms a un máximo de 2 ms. La relación anchura de pulso y posición del eje es de 1,5 ms normalmente. Si la anchura del pulso es de 1 ms, el servomotor se posiciona en el extremo izquierdo, mientras que si el pulso tiene una anchura de 2 ms la posición del servomotor es el extremo puesto. Esta técnica se conoce como PWM (*Pulse Width Modulation*). El servomotor trabaja comparando la anchura del pulso de entrada con la anchura del pulso producido por un *timer* interno que es controlado por el potenciómetro. La diferencia entre la anchura del pulso de entrada y la anchura del pulso interno se utiliza como señal de error.

El servomotor se encarga de determinar la dirección en la que ha de girar el motor para minimizar dicho error, activando los *drivers* de salida apropiados. El motor girará modificando la posición del potenciómetro de retroalimentación. Cuando llega el siguiente pulso se vuelve a realizar la comparación, comprobando de forma continua la posición del eje y realizando también constantemente las correcciones necesarias en la posición del mismo.

Si la posición del potenciómetro no se iguala con la posición deseada del eje, el motor se moverá hacia adelante o hacia atrás, hasta que la posición del potenciómetro sea equivalente a la posición deseada del eje. En este momento la corriente del motor se apaga. La precisión al posicionarse depende tanto de la precisión del potenciómetro como de la precisión de la anchura de los pulsos que llegan al motor. La señal de error suele ser de unos 5 $\mu$ s, diferencia entre el ancho del pulso de la señal de entrada y el ancho del

pulso de la señal interna, esto corresponde con una fracción de grado del recorrido del servomotor. Al ser el cero crítico, cuando el error está en este rango (conocido como zona muerta), el servomotor apaga los drivers del motor. Si la señal de error no está por debajo de  $5\mu\text{s}$ , la electrónica interna continuará intentando cancelar el error haciendo girar el motor atrás o adelante en un movimiento conocido como “hunting” [Palacios 2006].

### **2.1.2 Dispositivos de adquisición de datos**

*Sensor.*- Un sensor en un robot trata de imitar los sentidos de un ser humano midiendo las características del ambiente en el que se encuentra y proporcionando señales eléctricas. Su función es captar y guardar información del medio ambiente para después transformarla en parámetros que puedan ser reconocidos por el robot.

*Sensores ópticos.*- Los sensores ópticos tienen la capacidad de detectar magnitudes físicas y químicas, toleran las condiciones de ambientes difíciles como campos electromagnéticos, temperaturas elevadas y medios corrosivos. Los sensores ópticos están compuestos por un LED (infrarrojo o láser), fibra óptica y un fotodetector [Pallás 2007].

*Sensores ultrasónicos.*- Los sensores ultrasónicos se basan en una radiación mecánica superior a las auditivas, al incidir estas radiaciones en un objeto pueden ser reflejadas, transmitidas y absorbidas. Un sensor ultrasónico está relacionado con la velocidad, el tiempo de propagación y en algunos casos con la atenuación o interrupción del haz propagado [Pallás 2007].

*Sensores táctiles.*- Los sensores de contacto o táctiles indican si hay algún contacto o no con algún objeto, pueden ser interruptores de límite o microinterruptores que cuando se contactan con un objeto cambian de estado. Los sensores de contacto pueden ser utilizados para proporcionar información palpable. La detección del objeto es análoga a

los sentidos del ser humano, responden a la presencia o ausencia de un objeto en la disposición más simple. Algunas de sus características son [Hurtado 2008]:

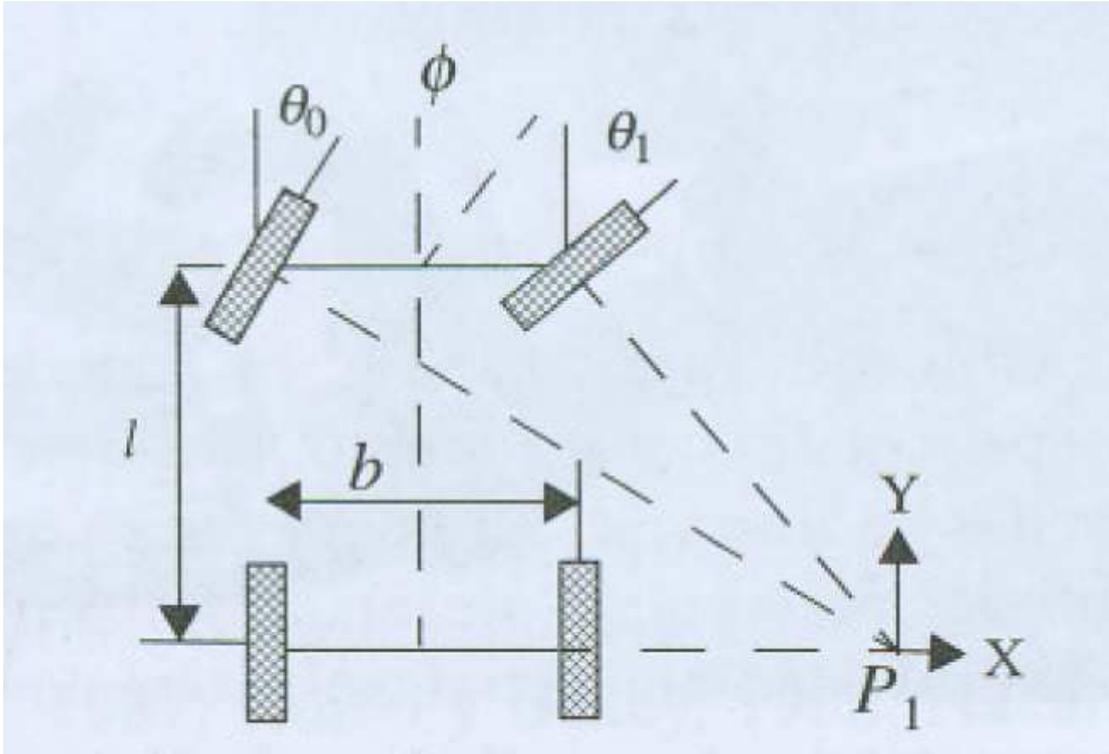
- Distancia de detección: 8mm
- Alcance real: 6.4mm
- Voltaje de alimentación: 12-24V DC
- Máxima corriente de salida: 200mA

### **2.1.3 Tipos de locomoción en robots móviles**

Una parte importante que caracteriza a los robots móviles es el sistema de locomoción con el que se desplazan. [Ollero 2001] menciona los tipos de locomoción siguientes:

#### a) *Ackerman*

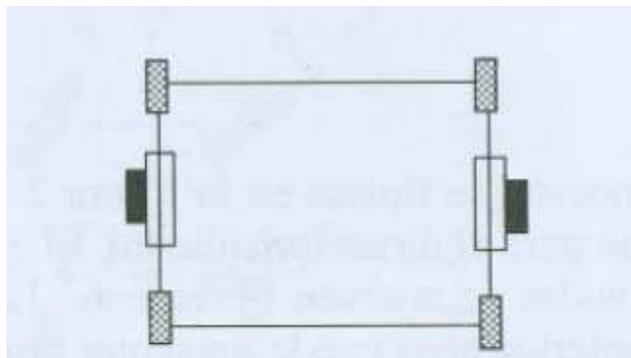
Este sistema de locomoción es utilizado en vehículos de cuatro ruedas, funciona de la siguiente manera: La rueda delantera interior gira en un ángulo ligeramente superior a la rueda delantera exterior, las prolongaciones de los ejes de las ruedas delanteras se intersecan en un punto sobre la prolongación del eje de las dos ruedas traseras. Los puntos trazados por los centros de las ruedas son circunferencias concéntricas con el eje de rotación  $p_1$  como se muestra en la Figura 2.1. Los vectores de velocidad instantánea son tangentes a las curvas.



**Figura 2.1** Sistema Ackerman [Ollero 2001].

b) *Direccionamiento diferencial*

El sistema *direccionamiento diferencial* emplea cuatro ruedas de castor no actuadas para dar soporte de plataforma. El desplazamiento es dado por la diferencia de velocidades de las ruedas laterales del sistema, (ver la Figura 2.2.). Esta configuración es frecuente en robots para interiores.



**Figura 2.2.** Locomoción con direccionamiento diferencial [Ollero 2001]

c) *Skid Steer*

El sistema *Skid Steer* dispone de varias ruedas en cada lado del robot que actúan de forma simultánea. La combinación de las velocidades de las ruedas de la izquierda con las de la derecha dan como resultado el movimiento.

d) *Pista de deslizamiento*

La impulsión y direccionamiento en los sistemas se consigue mediante pistas de desplazamiento. Su función es similar al del sistema *Skid Steer*. Este sistema de locomoción es ideal para el desplazamiento en terrenos irregulares. La resistencia al desplazamiento y al desgaste es mayor que en otros sistemas de locomoción.

e) *Síncrona*

En sistema con locomoción síncrona todas las ruedas actúan simultáneamente (de forma síncrona). El movimiento se consigue mediante coronas de engranaje o coronas concéntricas.

f) *Locomoción mediante patas*

La locomoción mediante patas aísla el cuerpo del terreno, emplea únicamente puntos de soporte. Para mantener estabilidad se adapta un polígono de soporte. Este sistema es capaz de atravesar terrenos difíciles y llenos de obstáculos. La complejidad del mecanismo y el consumo de energía son mayor en los sistemas de locomoción con patas, las configuraciones más comunes son seis u ocho patas, cuadrúpedos, bípedos, los dos últimos mantiene el equilibrio de forma dinámica

## **2.1 Características de robot LEGO**

El robot LEGO es fácil de manejar, tiene múltiples aplicaciones y puede ser programado por múltiples lenguajes [Lego, 2008]. Cuenta con los siguientes sensores:

- Sensor de tacto.- Permite al robot “sentir” y reaccionar al entorno.
- Sensor de sonido.- Permite al robot reaccionar al sonido.

- Sensor de luz.-Se emplea para detectar luz y color.
- Sensor de ultrasonidos.- Sirve para medir distancias y reaccionar al movimiento.
- Servomotores.- Aseguran los movimientos del robot con precisión controlada.

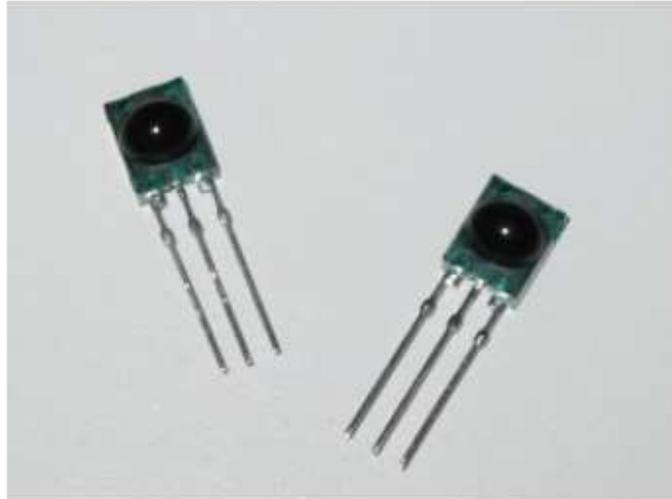
Las características técnicas del robot LEGO son:

- Servomotores
- Microprocesador de 32-bits más memoria FLASH.
- Interfaz compatible con PC y Mac.
- 3 puertos de entrada y salida.
- Display matricial.
- Altavoz piezoeléctrico.
- Soporte USB 2.0 y Bluetooth.
- Entorno de programación intuitivo basado en iconos.
- Requiere 6 pilas 1,5 V AA

## **2.3 Dispositivos de comunicación**

### **A) Infrarrojo**

La señal infrarroja es una de las tecnologías que establecen la comunicación inalámbrica, permite enviar señales a un robot. La comunicación por medio de señales infrarrojas está compuesta por dos partes, la emisión y recepción de la señal [de Lope 2001]. La recepción utiliza un dispositivo que unifica el receptor de luz infrarroja, un lente y la distribución de señales moduladas a una determinada frecuencia. Utiliza receptores IS1U60 que se activan cuando reciben una luz infrarroja modulada a una frecuencia de 38 kHz (ver la Figura 2.3).



**Figura 2.3.** Receptor IS1U60

La emisión de señales infrarrojas utiliza una ligera modificación para conseguir un 50% de ciclo de trabajo, que es lo que requiere el receptor. Sólo durante la mitad del tiempo envía la señal. El tiempo que está encendido el LED es el mismo que el que está apagado. Al tratarse de una frecuencia de 38 kHz, este tiempo es de 13.158  $\mu$ s. Los condensadores de filtro y transistores activan el receptor con menos elementos, esto aumenta la corriente que circula por el LED infrarrojo [de Lope 2001]

#### b) Bluetooth

Bluetooth es el nombre de la especificación industrial IEEE 802.15.1 que define un estándar global de comunicación inalámbrica, posibilita la transmisión de datos por medio de un enlace de radiofrecuencia. El canal de comunicación es de máximo 720 kb/s con rango óptimo de 10 metros (opcionalmente 100 metros con repetidores). La frecuencia de tráfico con la que trabaja se encuentra en el rango 2,4 a 2,48 GHz. El dispositivo bluetooth está compuesto por un dispositivo de radio y un controlador digital [Pérez 2007].

## **2.4 Características de ambientes**

Existe dos tipos de ambientes por los cuales un robot se puede desplazar: cerrados y abiertos. Para cada uno existen diferentes tipos de restricciones como se describe a continuación:

*Ambientes cerrados.*- Los modelos de exploración en ambientes cerrados se construyen a partir de un mapa que depende tanto de señales externas del entorno como de información generada internamente en el robot. Con frecuencia, la entrada depende sólo de la imagen captada por el robot.

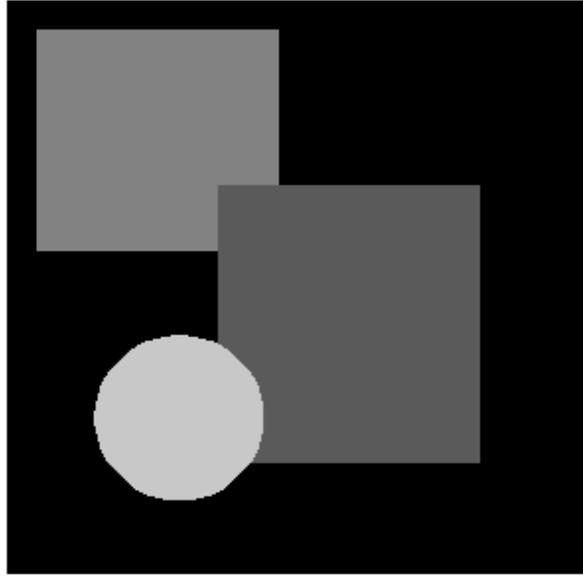
*Ambientes abiertos.*- La exploración en campos abiertos por medio de sensores es más compleja debido al corto alcance de los dispositivos de adquisición del robot (sensores). Dispositivos como cámaras fotográficas proyectan información del ambiente por donde el robot se desplaza, esta información debe de ser procesada de diferente manera a la información que proviene de sensores, por ello se adoptan diferentes métodos de reconstrucción.

## **2.5 Reconstrucción virtual de ambientes**

### **2.5.1 Representación de la realidad**

La representación de objetos existentes en la realidad en la computadora es un ejemplo de visión artificial. Estas representaciones pueden ser imágenes estáticas, escenas tridimensionales o imágenes en movimiento [Vélez 2003].

Una imagen bidimensional es una función donde a cada par de coordenadas  $(x, y)$  se le asocia un valor (ver la Figura 2.4). Una imagen sin información de color se representa como una superficie en la cual la altura de cada punto indica su nivel de brillo. Las escenas en 3D son otra forma de representar la realidad, consiste en asignar a cada punto del espacio  $(x, y, z)$  una propiedad del objeto [Vélez 2003].

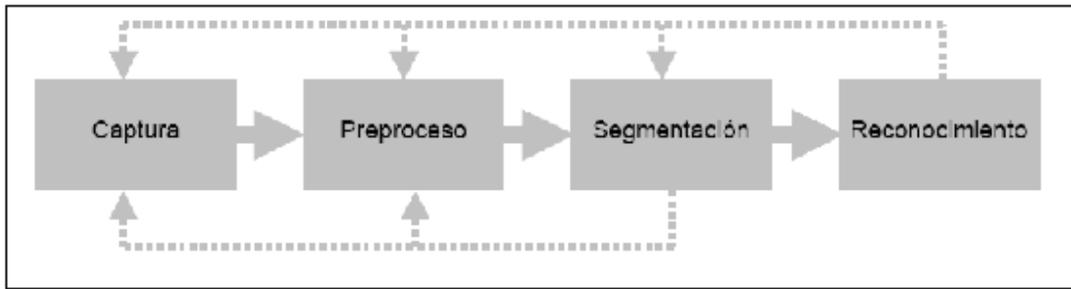


**Figura 2.4** Imagen en dos dimensiones [Vélez 2003]

### **2.5.2 Etapas de un sistema de visión artificial**

La visión artificial define las cuatro etapas de la Figura 2.5, las cuales se describen a continuación:

- *Captura o adquisición* de las imágenes digitales mediante algún tipo de sensor.
- *Pre proceso* o tratamiento digital para eliminar o realzar partes de la imagen mediante filtros y transformaciones geométricas.
- *Segmentación* que aísla los elementos que interesan de una escena.
- *Reconocimiento o clasificación* para distinguir los objetos segmentados a través del análisis de ciertas características que se establecen para diferenciarlos.



**Figura 2.5.** Etapas de reconstrucción [Vélez 2003]

Las etapas no se siguen siempre de manera secuencial, puede haber un retroceso si alguna falla [Vélez 2003].

### 2.5.3 Método de reconstrucción GPS

El Sistema de Posicionamiento Global (GPS) es un sistema de localización que proporciona estimaciones de posición, velocidad y tiempo para determinar la altitud, longitud y latitud de cualquier objeto en la Tierra. El objetivo es calcular la posición de un punto cualquiera en un espacio de coordenadas (x, y, z) utilizando una red de computadores y un conjunto de 24 satélites. El sistema GPS se descompone en tres segmentos básicos [Pozo 2007]:

- *Segmento-espacio* formado por 24 satélites GPS con una órbita de 26560 Km. de radio y un periodo de 12 h.
- *Segmento-Control*, consta de cinco estaciones monitoras que mantienen en órbita los satélites y supervisan su funcionamiento, tres antenas terrestres que envían señales de transmisión a los satélites y una estación experta de supervisión de operaciones.
- *Segmento-usuario* formado de antenas y receptores pasivos situados en Tierra. Los receptores calculan distancias y proporcionan una estimación de posición y tiempo a partir de los mensajes que provienen de cada satélite visible.

La distancia entre el usuario y un satélite se mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación. Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados. La desviación en los relojes de los receptores añade una incógnita más que hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones.

## **2.6 Planificación de trayectorias**

La planificación de trayectorias consiste en determinar en qué parte del ambiente se va a desplazar el robot, estableciendo un camino desde la posición inicial a una final definiendo los puntos intermedios de paso. Existen diferentes métodos para la planificación de trayectorias, [Molina 2006] menciona los siguientes:

- Planificación basada en grafos de visibilidad
- Planificación basada en modelado del espacio libre
- Planificación basada en la descomposición en celdas
- Planificación basada en campos potenciales
- Planificación de ruta on-line

## **2.7 Aplicaciones de la reconstrucción de ambientes**

La reconstrucción de ambientes en dos dimensiones puede tener diversas aplicaciones como: la limitación del ambiente, reconstrucción de pequeños ambientes, objetos dentro de los bordes de un ambiente, planificación en ambientes desconocidos, la planificación de trayectorias, la planificación en medios ambientes reales desconocidos generando planes de acción del ambiente a explorar.

## **Capítulo 3. Metodología**

La reconstrucción de ambientes en dos dimensiones se realiza a través de un robot LEGO Mindstorms NTX que se desplaza en un ambiente no conocido. El robot se encuentra en una posición inicial a partir de la cual emite señales de luz a través de los sensores integrados en él. Con los datos obtenidos de los sensores, se busca una ruta sin obstáculos para que pueda desplazarse libremente. Una vez encontrada la ruta de desplazamiento, el robot comienza a moverse y toma las señales de los sensores ultrasónicos para delimitar el contorno del ambiente.

### **3.1 Análisis de requerimientos del sistema**

El proyecto requiere que el robot LEGO Mindstorms NTX se desplace en un ambiente no conocido y que adquiera datos del ambiente para realizar un mapa de éste en dos dimensiones.

### **3.2 Situación actual**

En la actualidad, la existencia de robots exploradores ya no es una novedad, estos robots realizan la reconstrucción de ambientes simulados por medio de fotos, videos o sensores. Este proyecto se realiza la reconstrucción de ambientes desconocidos de forma diferente a las que existen en la actualidad.

### **3.3 Requisitos**

Para poder realizar la reconstrucción de un ambiente se requiere de:

- Un ambiente a explorar
- Un algoritmo de desplazamiento
- Conexión de robot a la PC
- Una aplicación para pintar los datos obtenidos por el robot

### **3.4 Requerimientos funcionales**

Los requerimientos funcionales para este proyecto son: la especificación de movimientos del robot, la adquisición de datos para pintar la interfase y búsqueda de la ruta de desplazamiento que se describen a continuación:

#### **a) Especificación de movimientos**

Para que el robot pueda realizar cualquier tipo de movimiento hace uso de dos motores determinados por MI (motor izquierdo) y MD (motor derecho). Se requieren las funciones siguientes de desplazamiento:

Desplazamiento hacia delante: se ejecuta cuando los dos motores MD y MI avanzan en sentido de las manecillas de reloj.

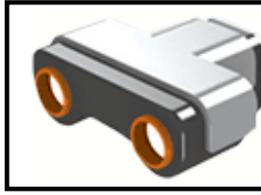
Desplazamiento hacia atrás: se efectúa cuando los dos motores MD y MI avanzan en sentido contrario a las manecillas de reloj.

Movimiento derecha: se ejecuta cuando MD se detiene y MI gira en sentido de las manecillas del reloj

Movimiento izquierdo: se realiza cuando MI se detiene y MD gira en sentido contrario a las manecillas del reloj.

#### **b) Adquisición de datos para trazar el mapa**

El robot LEGO Mindstorms NTX está equipado con 5 sensores, dos de ellos ultrasónicos. La Figura 3.1 (a) muestra uno, estos sensores mandan señales de sonido que son rebotadas cada vez que chocan con un objeto, la distancia que hay entre el sensor y un objeto es obtenida, por el tiempo que se toma la señal en salir del sensor y regresar dividida entre dos, esta distancia se registra en un archivo de texto para pintar el contorno del ambiente por donde se desplaza el robot. Los sensores ultrasónicos están colocados en los costados derecho e izquierdo del robot.



**Figura 3.1.** Sensor ultrasónico.

### c) **Búsqueda de ruta de desplazamiento**

El desplazamiento del robot se realiza con la ayuda de tres sensores ópticos Fig. 3.2., colocados en el robot SOD(Sensor Óptico Derecho) colocado al costado derecho del robot, SOI(Sensor Óptico Izquierdo) colocado al costado izquierdo del robot, y SOF(Sensor Óptico Frente), este último sensor colocado en la parte de enfrente del robot. Los sensores son configurados para verificar dos tipos de colores, los claros o blancos y los oscuros o negros. Si el sensor óptico detecta un color negro u oscuro esto indica que hay un obstáculo por lo tanto, el robot se detiene y busca otra ruta para desplazarse, (ver algoritmo de desplazamiento), si el sensor esta posicionado frente a un objeto claro o blanco el robot avanza. Las lecturas del sensor cuando el robot se esta desplazando se realiza cada segundo y en este segundo el robot recorre 24 cm.



**Figura 3.2.** Sensor óptico

## **3.5 Descripción del algoritmo de desplazamiento**

El desplazamiento del robot emplea un algoritmo formado de los siguientes 6 pasos que se describe a continuación.

Paso 1: El robot se coloca en una posición inicial con los sensores ópticos y ultrasónicos encendidos.

Paso 2: Los sensores ópticos emiten una señal de luz.

Paso 3: Se verifica el SOF. Si esté sensor detecta un color blanco o claro, el robot se desplaza hacia delante. En caso contrario, realiza el paso 4.

Paso 4: Se verifica el SOD. Si esté sensor detecta un color claro o blanco, el robot realiza un movimiento a la derecha y se desplaza hacia delante. En caso contrario, realiza el paso 5.

Paso 5: Se verifica el SOI. Si esté sensor detecta un color claro o blanco, el robot realiza un movimiento a la izquierda y se desplaza. En caso contrario, realiza el paso 6.

Paso 6: El robot realiza un movimiento a la derecha, se verifica el SOD. Si esté sensor detecta un color claro o blanco, gira a la derecha y se desplaza hacia delante. En caso contrario, no se ejecuta ninguna función.

### **3.6 Descripción de la interfaz para pintar**

Para obtener un grafico el robot debe haber realizado una exploración, de esta exploración se deben de almacenar los datos obtenidos del sensor ultrasónico, estos datos son enviados en un archivo de texto a la computadora por medio de bluetooth, en la interfase se abre este archivo y los valores se verifican antes de ser enviados al vector, si su valor numérico es mayor a 20cm. no se almacenan en el vector, si su valor es menor entonces se almacena, de esta manera se llena el vector con los valores de Y, el vector que almacena las coordenadas en X se obtiene aplicando la formula  $Distancia = tiempo \times velocidad \text{ del sonido} / 2$  [Bustillos 2004].

Una vez obtenidos los valores de X y Y se mandan a pintar con la propiedad `canvas.lineto()` de delphy, finalmente se guarda el mapa dibujado por la interfase con `tsavepicture`.

### **3.7 Requerimientos no funcionales**

La reconstrucción de ambientes se realiza en cualquier lugar cerrado, no se considerarán restricciones en cuanto a claves y contraseñas de usuario. La aplicación trabaja bajo la plataforma Windows XP, está desarrollada en el lenguaje de programación Delphi versión 7. No existen requerimientos en relación a la velocidad de desplazamiento del robot.

## Capítulo 4. Implementación

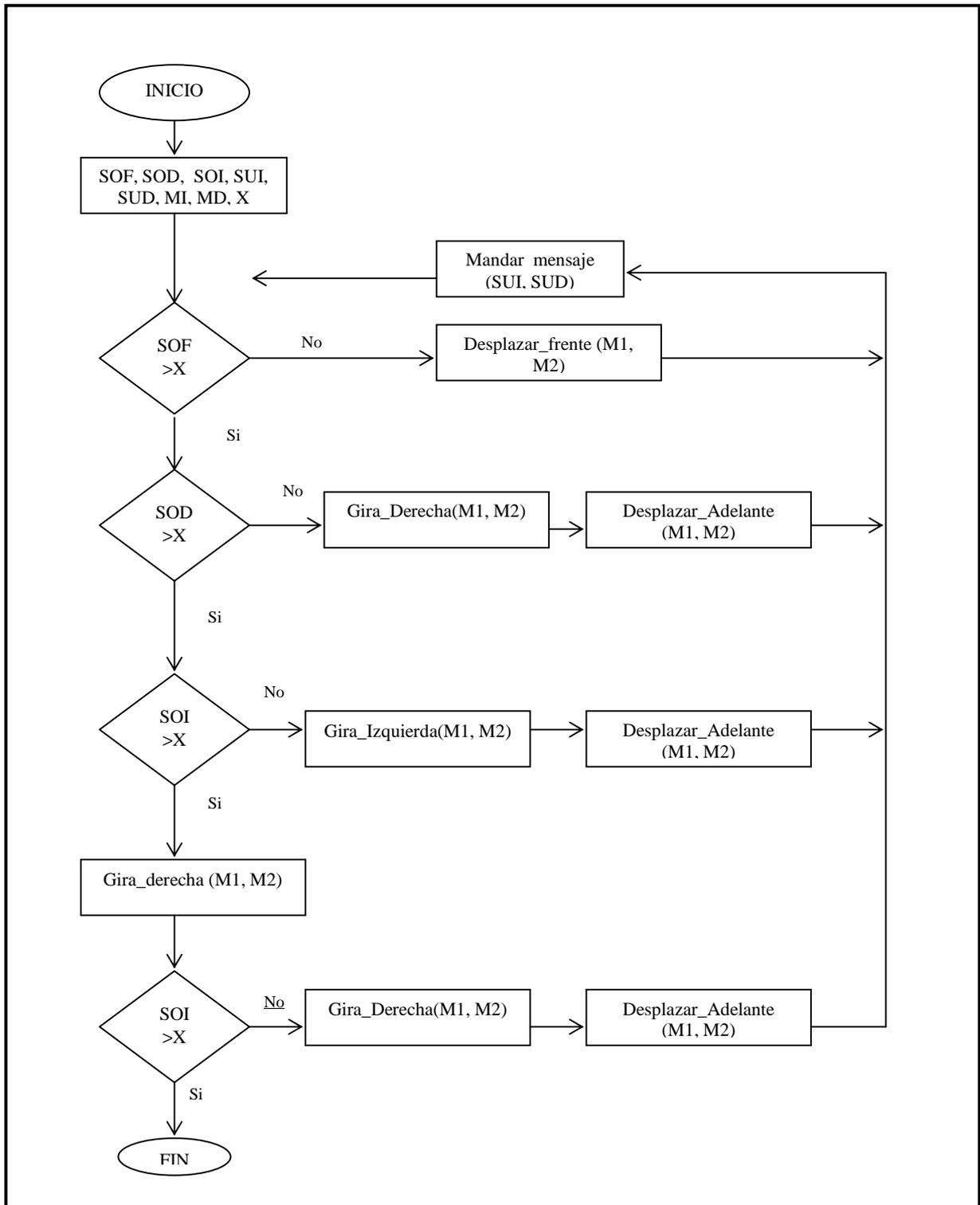
La implementación de la reconstrucción de ambientes en dos dimensiones se compone por 3 fases: 1) implementación del algoritmo de desplazamiento, 2) conexión entre el robot y la computadora y 3) aplicación encargada de pintar el ambiente explorado. Estas fases se describen en las secciones siguientes.

### 4.1 Algoritmo de desplazamiento

El algoritmo de desplazamiento emplea el diagrama de flujo de la Figura 4.1. Este algoritmo fue descrito en el capítulo 3. La Tabla 4.1 describe las variables utilizadas.

**Tabla 4.1** Descripción de variables para el diagrama de flujo de la Figura 4.1

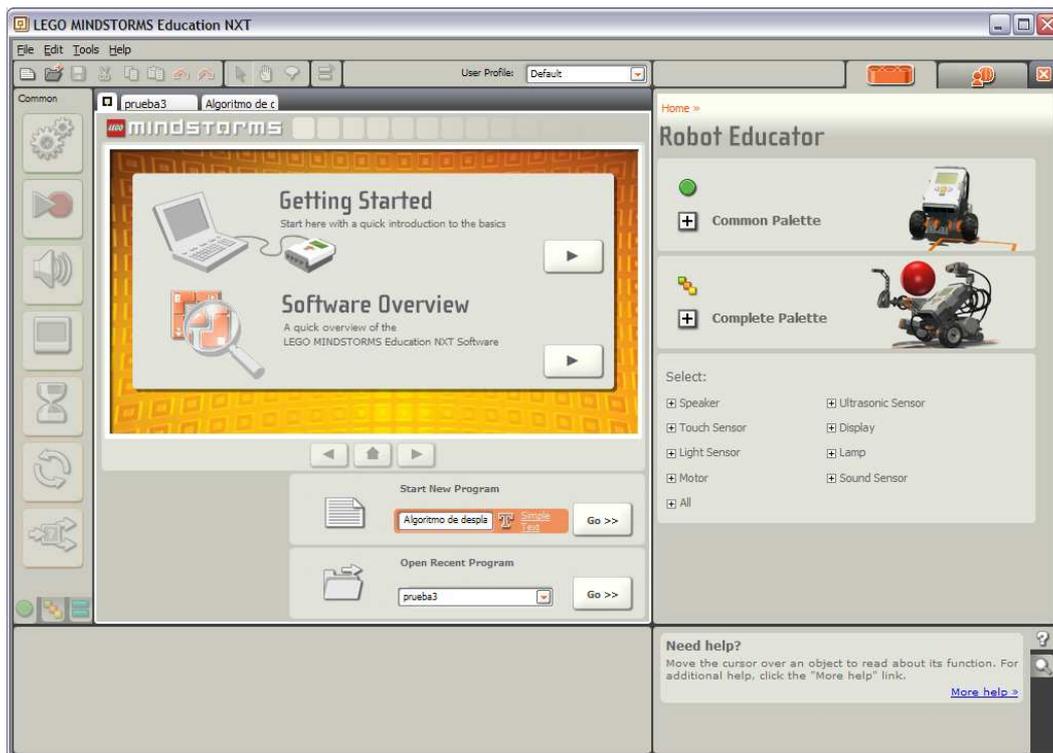
<b>Siglas</b>	<b>Descripción</b>
SOF	Sensor óptico frente
SOD	Sensor óptico derecho
SOI	Sensor óptico izquierdo
SUI	Sensor ultrasónico derecho
SUD	Sensor ultrasónico izquierdo
MD	Motor derecho
MI	Motor izquierdo
X	Rango de valores para los sensores ópticos (valor máximo y mínimo)



**Figura 4.1.** Diagrama de flujo del algoritmo de desplazamiento

### 4.1.1 Descripción de software y herramientas de LEGO

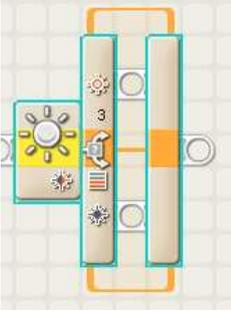
El robot LEGO trae consigo su propio software de programación llamado LEGO Mindstorms Education NTX, el cual cuenta con un ambiente gráfico y funciones predefinidas que el usuario puede utilizar, (ver la Figura 4.2).



**Figura 4.2** Pantalla inicial de LEGO Mindstorms Education NTX.

De la barra de herramientas de LEGO Mindstorms Education NTX, se utilizan las descritas en la Tabla 4.2.

**Tabla 4.2.** Herramientas de LEGO

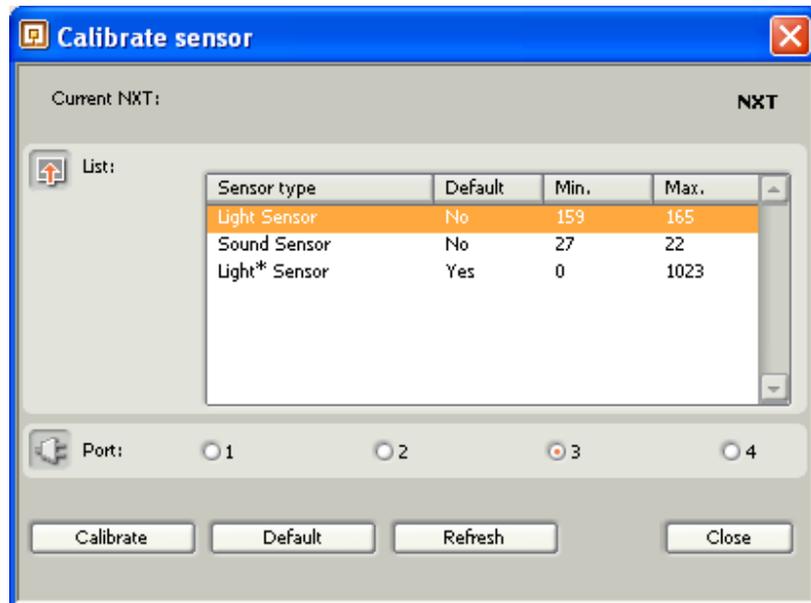
Herramienta	Función
	<p>MOTORES: Manipula los movimientos del robot</p>
	<p>SENSOR ULTRASÓNICO: Calcula la distancia entre sensor ultrasónico y un objeto</p>
	<p>SWITCH o INTERRUPTOR: Determina la dirección que debe seguir el robot según los valores que verifique el sensor óptico.</p>
	<p>ENVIAR MENSAJE: Envía un archivo de texto a la computadora.</p>
	<p>CICLO: Implementa un ciclo.</p>

### 4.1.2 Calibración de sensores

La calibración de sensores se realiza de acuerdo al tipo de sensor (óptico o ultrasónico) como se describe a continuación:

a) *Sensores ópticos.*- Para calibrar los sensores ópticos se toman dos valores: 1) el valor máximo que se obtiene cuando el sensor está colocado frente a un objeto de color oscuro

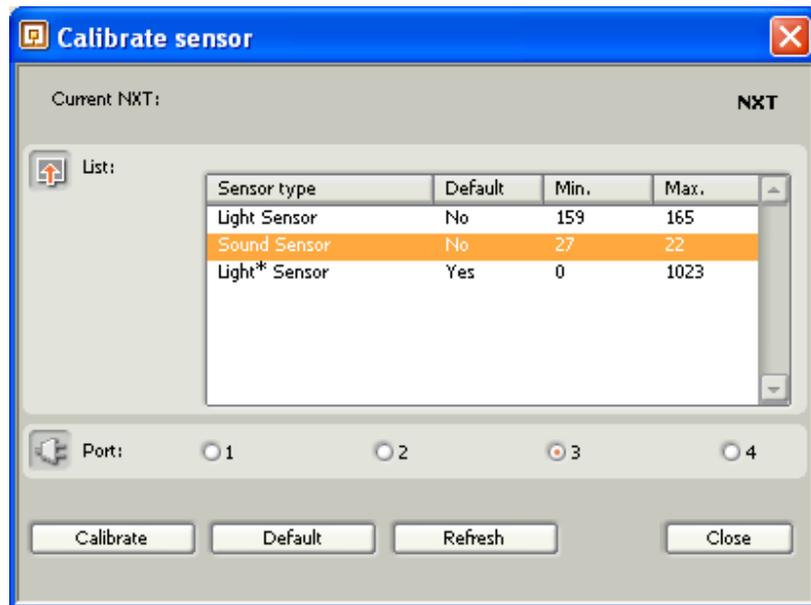
o negro y 2) el valor mínimo se obtiene cuando el sensor está colocado frente un objeto de color claro o blanco. La calibración de los sensores ópticos se ve afectada cada vez que el ambiente cambia. La Figura 4.3 muestra un ejemplo de estos valores.



**Figura 4.3** Valores máximo y mínimo de la configuración del sensor óptico

Los valores para los tres sensores de luz (SOF, SOD y SOI) son los mismos.

b) *Sensores ultrasónicos.*- Para calibrar los sensores ultrasónicos (al igual que en los sensores ópticos) se establecen valores máximo y mínimo. La calibración es la misma para los dos sensores ultrasónicos que el robot utiliza. La Figura 4.4 muestra los valores máximo y mínimo definidos que el sensor ultrasónico toma para su calibración. Estos valores se determinan experimentalmente.

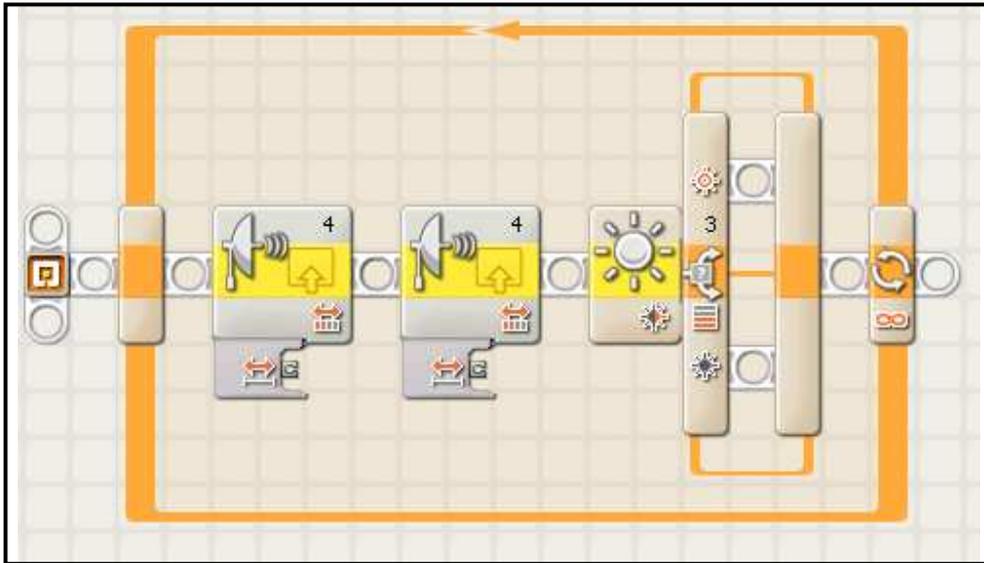


**Figura 4.4** Valores máximo y mínimo de la configuración del sensor ultrasónico

### 4.1.3 Implementación del algoritmo de desplazamiento

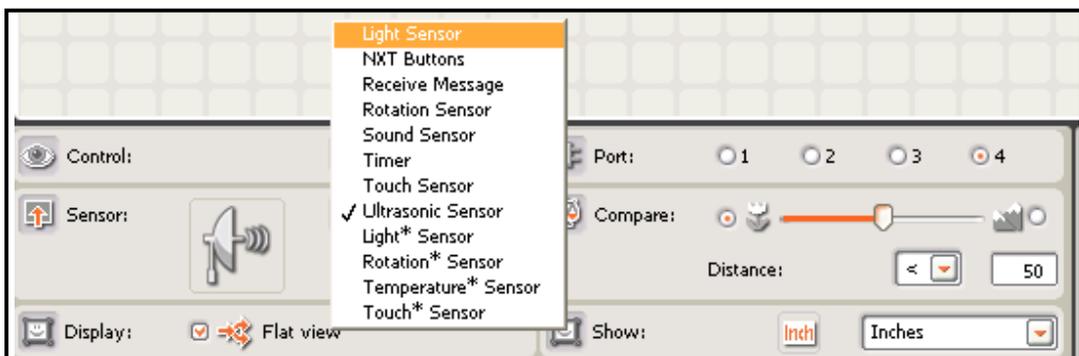
La implementación del algoritmo de desplazamiento se realiza en los pasos siguientes:

Paso 1: Se colocó la herramienta de ciclo y dentro de ésta, el SUD y el SUI, que obtienen y registran los valores de las distancias, en un archivo de texto necesarias para pintar el ambiente, seguido de un interruptor que verifica las señales del SOF, el cual determina que acción se realizará, tal como se muestra en la Figura 4.4.



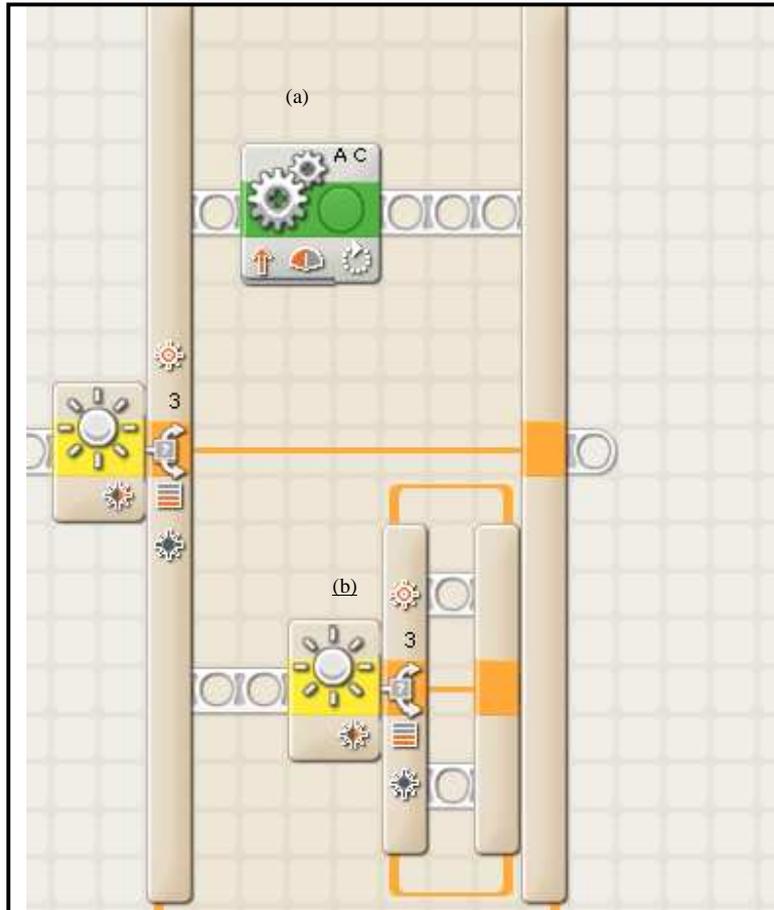
**Figura 4.4** Implementación del paso 1 del algoritmo de desplazamiento.

La Figura 4.5 muestra la configuración del interruptor. Esta configuración se realizó en todos los interruptores que se utilizaron en el desarrollo del algoritmo.



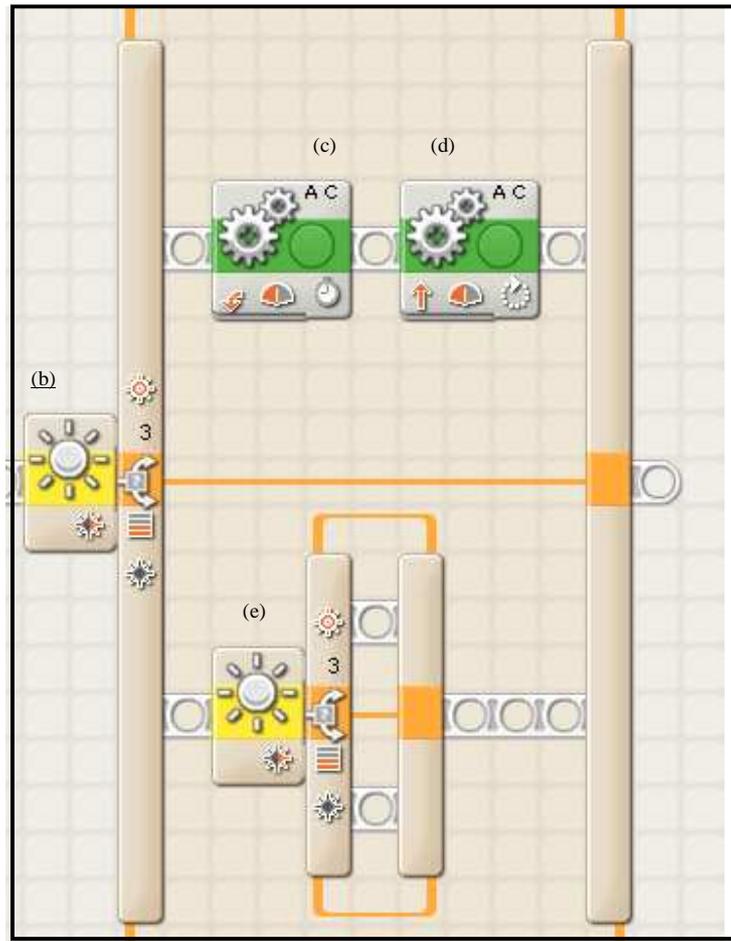
**Figura 4.5.** Configuración del interruptor

Paso 2: La representación de este paso se ilustra en la Figura 4.6. De acuerdo a las condiciones del primer interruptor descrito en el paso 1, se colocó un motor (a) en la parte superior que cumple la función de rotar hacia el frente. Del lado inferior se colocó un segundo interruptor (b) que valida los valores del SOD.



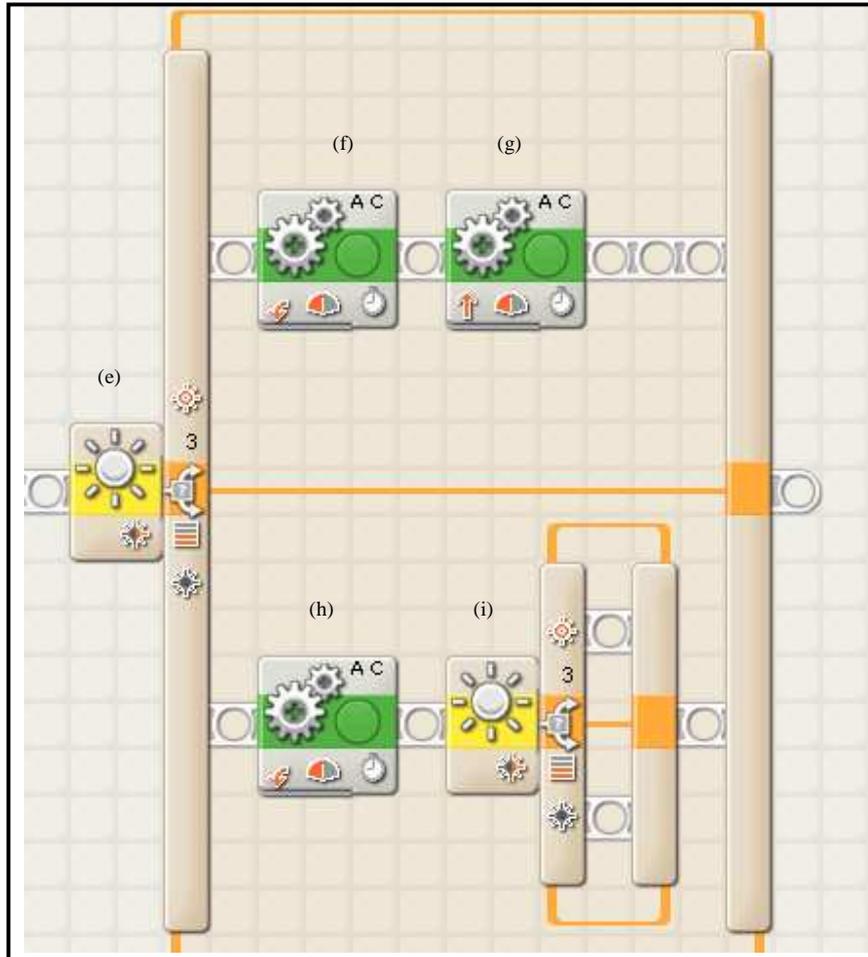
**Figura 4.6** Implementación del paso 2 del algoritmo de desplazamiento.

Paso 3: La representación de este paso se ilustra en la Figura 4.7. Después de haber colocado el interruptor (b) y establecido las condiciones de acciones, en la parte inferior se colocó el motor (c) que desempeña la función de girar a la derecha, seguido del motor (d) el cual rota hacia delante. En la parte inferior se colocó un interruptor (e) que valida el SOI, según sea el caso.



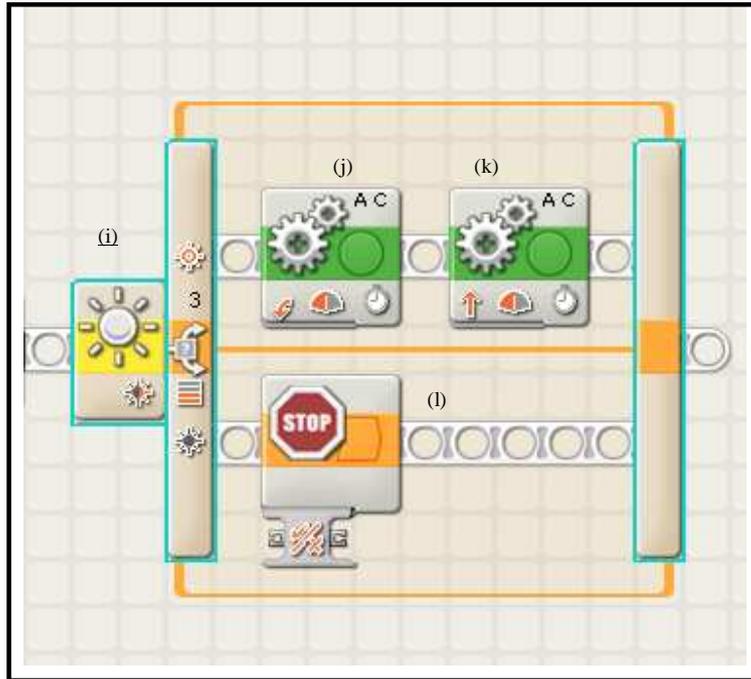
**Figura 4.7** Implementación del paso 3 del algoritmo de desplazamiento.

Paso 4: La representación de este paso se ilustra en la Figura 4.8. Para el interruptor (e) se establecieron nuevamente las condiciones de acción. En la parte superior del mismo interruptor se colocó el motor (f) que rota a la derecha, seguido del motor (g), en la parte inferior se colocó el motor (h) que da un giro a la izquierda para evaluar por última vez el SOD con el interruptor (i), según sea el caso.



**Figura 4.8** Implementación del paso 4 del algoritmo de desplazamiento.

Paso 5: La representación de este paso se ilustra en la Figura 4.9. El interruptor (i) verifica la última condición del algoritmo. Para ello se colocó en la parte superior el motor (j) que da un giro a la derecha y seguido del motor (k) que rota hacia el frente. En la parte inferior se colocó una función de paro, esto por si no se ejecuta la primera condición del interruptor (l).



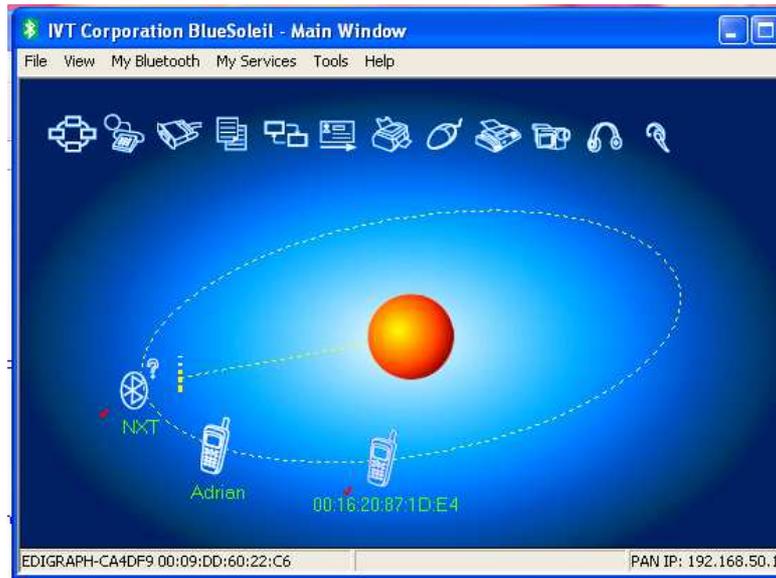
**Figura 4.9** Implementación del paso 5 del algoritmo de desplazamiento.

Una vez finalizada la implementación del algoritmo de desplazamiento, se agregó la herramienta para enviar mensajes con los datos de los sensores a la computadora.

## **4.2 Comunicación bluetooth entre el robot con la computadora**

El bluetooth fue el dispositivo que se eligió para establecer la comunicación entre el robot y la computadora. En el robot LEGO este dispositivo ya viene integrado y en la computadora se instaló el software IVT BlueSoleil para Windows, descrito en la sección de requerimientos del Capítulo 3. A través de esta comunicación el robot envía un archivo de texto con los datos obtenidos por el SUI y SUD para pintar el ambiente.

Se estableció al robot como maestro y a la computadora como esclavo, esto debido a que el robot es quien busca la conexión para enviar el archivo de texto a la computadora. La Figura 4.10 muestra la conexión bluetooth con el robot LEGO NTX y otros dispositivos.



**Figura 4.10** Comunicación entre IVT BlueSoleil y LEGO NTX.

Una vez establecida la comunicación entre la computadora y el robot, los archivos enviados se guardan en la carpeta Bluetooth\inbox:

(C:\Documents and Settings\Administrador\Mis documentos\Bluetooth\inbox).

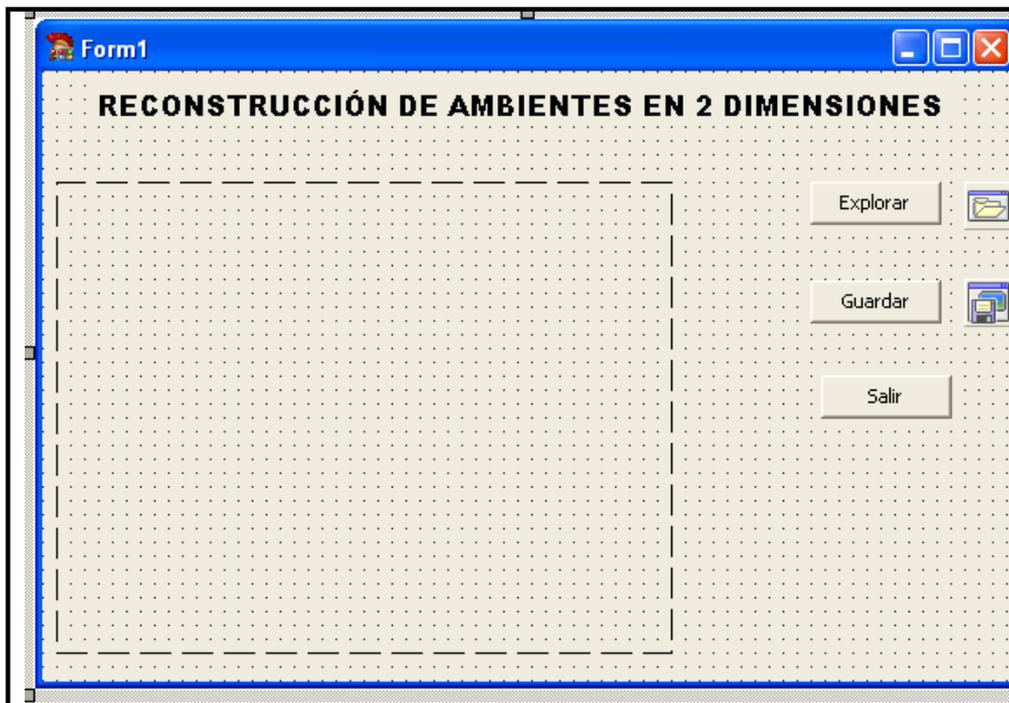
### 4.3 Implementación de la interfaz para pintar el mapa

La implementación de la interfaz que dibuja el mapa de desplazamiento del robot LEGO se implementó en Delphi 7 (como se menciona en la sección de requerimientos del Capítulo 3). Para el desarrollo de esta interfaz se utilizó la propiedad Canvas de Delphi. La Tabla 4.3 muestra las herramientas que se utilizaron para construir la interfaz de la aplicación.

**Tabla 4.3** Herramientas utilizadas de Delphi

Herramienta	Descripción
 TButton	Ejecuta las instrucciones elegidas
 TSavePictureDialog	Guarda la el mapa ya dibujado como una imagen
 TOpenDialog	Abre un archivo de texto para pintar
 TImage	Área donde se dibuja el mapa

Después de haber colocado cada herramienta sobre el formulario, la interfaz resultante se muestra en la Figura 4.11.



**Figura 4.11** Colocación de herramientas en el formulario

Las funciones de cada herramienta son las siguientes:

- *Botón explorar*: abre el archivo de texto enviado por el robot.
- *Botón guardar*: Guarda la imagen obtenida del ambiente explorado.
- *Cuadro de imagen*: muestra la imagen obtenida de la exploración (línea punteada).

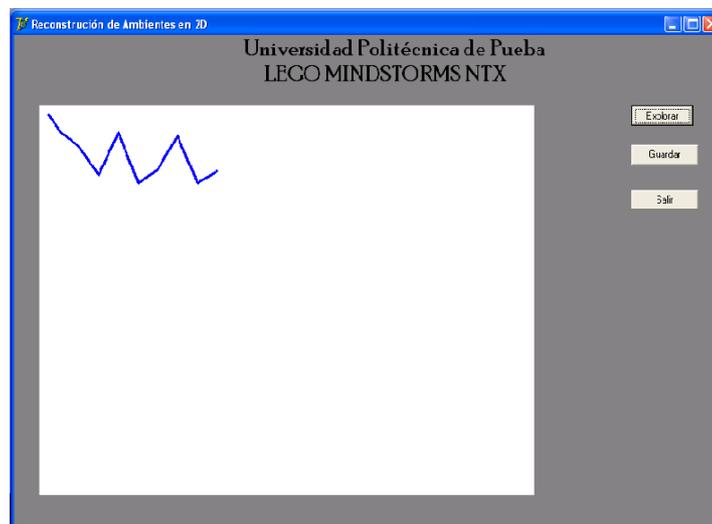
*Desarrollo de la función para pintar:*

Se abre el archivo de texto enviado por el robot con *TopenDilog*. Los datos obtenidos de este archivo de texto son verificados, si el valor de estos datos es mayor a cero y menor a veinte cm. se guardan en un vector que almacena los valores en X, en caso contrario no se toman en cuenta. Posteriormente se declara otro vector que almacena los valores en Y. Los valores en Y se obtienen de la fórmula siguiente:

$$\text{Distancia} = \text{tiempo} \times \text{velocidad del sonido} / 2 [\text{Bustillos 2004}]$$

Después de haber obtenido los valores de ambos vectores, se coloca un ciclo que recorre ambos vectores al mismo tiempo. Dentro de este ciclo se ejecuta la siguiente función para pintar: *Image1.Picture.Bitmap.canvas.lineto(coor[i1],coory[jj])*. El resultado se muestra en la Figura 4.12. Finalmente, los gráficos obtenidos se guardan como imágenes con la herramienta *TsavePictureDialog*. Las imágenes se almacenan en la carpeta: *aplicacion/imagenes*

(*C:\Documents and Settings\Administrador\Escritorio\Aplicacion\imagenes*)



**Figura 4.12.** Ejemplo de la implementación de la función pintar

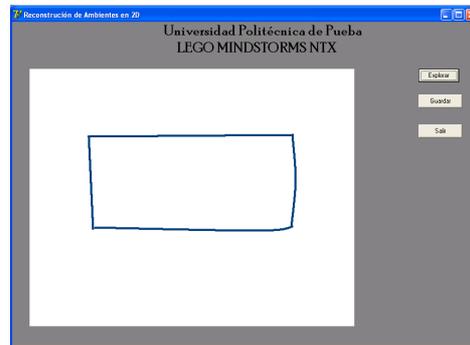
Después de haber concluido con las tres fases de la implementación, el capítulo siguiente describe las pruebas para validar el sistema.

## Capítulo 5. Pruebas de ambientes explorados

Este capítulo describe las pruebas de los ambientes explorados por el robot LEGO. Para la serie de pruebas se eligieron tres tipos de ambientes: ambientes abiertos, ambientes cerrados y ambientes parcialmente abiertos. Se realizaron 5 pruebas con escala de cada 24 cm por cada píxel de las cuales se describen a continuación:

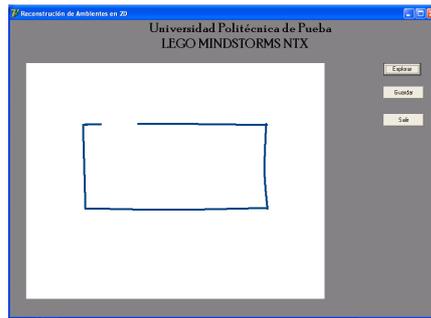
### 5.1 Ambientes explorados

Ambiente 1: Es un ambiente cerrado que mide 2m X 1.50m. Está elaborado de madera y forrado con cartulina negra, el tiempo utilizado para obtener este gráfico fue de 20 minutos. De este ambiente se obtuvo la imagen que se muestra en la Figura 5.1.



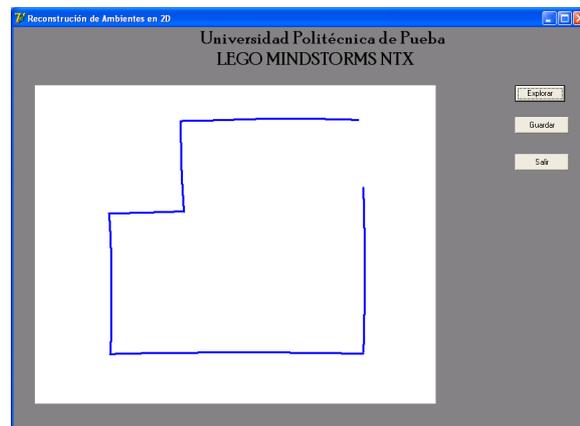
**Figura 5.1.** Reconstrucción del ambiente 1.

Ambiente 2: Es un ambiente con una salida. Para la segunda prueba se tomó el ambiente 1 y se simuló una salida, las características del ambiente son las mismas al igual que la escala, el tiempo que se tardó en graficar fue de 15 minutos. La imagen obtenida de esta exploración se muestra en la Figura 5.2.



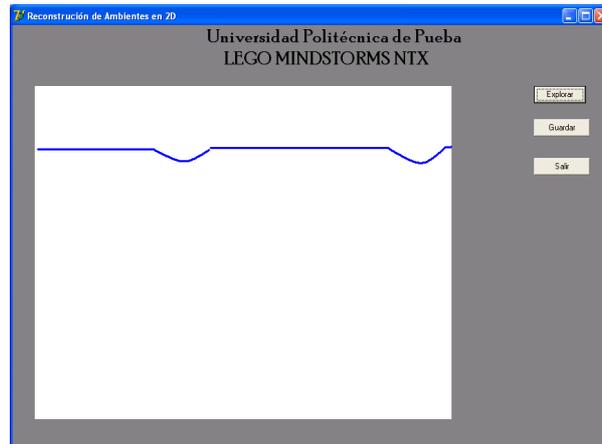
**Figura 5.2.** Reconstrucción del ambiente 2 con una salida.

Ambiente 3: Es un ambiente con una salida con dimensiones 4m X 3m el cual se forró con papel crepe negro. Fue un ambiente real que contenía un obstáculo en la esquina superior izquierda, lo que impidió graficar el contorno real del ambiente y el tiempo transcurrido en obtener el grafico fue de 28 minutos, La imagen obtenida de esta exploración se muestra en la Figura 5.3.



**Figura 5.3.** Reconstrucción del ambiente 3 con una salida.

Ambiente 4: Es un ambiente parcialmente abierto (solo se graficara un lado del ambiente debido a que es un ambiente grande). El tiempo utilizado para obtener el grafico fue de 20 minutos y la dimensión fue de 5.5 metros de largo. La imagen obtenida de esta exploración se muestra en la Figura 5.4.



**Figura 5.4.** Reconstrucción de ambiente 4.

Ambiente 5: Este ambiente es abierto. El robot se desplazó por el ambiente en un tiempo de 1 minuto las dimensiones son de 6 metros de ancho por 12 de ancho los valores obtenidos, fueron restringidos por la interfaz debido a esto la interfase no mostró ninguna imagen..

## Capítulo 6. Conclusiones

Para la elaboración de la reconstrucción en dos dimensiones se establecieron tres objetivos. El primero fue realizar un algoritmo que permitiera al robot desplazarse en un ambiente, el cual fue alcanzado con el desarrollo de un algoritmo de seis pasos que fue implementado en el software de la herramienta LEGO. El segundo objetivo fue establecer la comunicación bluetooth entre el robot y la máquina, este objetivo no se llegó a cubrir por completo debido a la incompatibilidad entre el bluetooth instalado en la computadora y el que viene integrado al robot LEGO. Se realizaron pruebas entre teléfonos móviles que contaban con bluetooth y la computadora, donde sí se logró el envío de información. Aunque la computadora si reconocía por medio de bluetooth al robot y viceversa, al tratar de enviar información siempre se presentaron fallos. El tercer objetivo se refiere a la implementación de una interfaz que grafica los datos enviados por el bluetooth en un archivo de texto. Los datos se validan mediante una serie de restricciones y finalmente se grafican.

Las pruebas realizadas para este proyecto fueron cinco: 1) un ambiente cerrado con dimensiones de 2m X 1.50m , 2 y 3) dos ambientes con una salida con dimensiones de 2m X 1.50m y 4m X 3m, el último contaba con un obstáculo, 4) un ambiente parcialmente abierto con dimensiones de 5.5 m del lado que se graficó y finalmente 5) un ambiente abierto con dimensiones de 6 X 12 m. Se obtuvieron imágenes de los 4 primeros ambientes, del último no debido a las restricciones implementadas en la interfaz.

Tanto la reconstrucción de ambientes en tres dimensiones como la reconstrucción en tiempo real y la detección de obstáculos son trabajos a futuro para este proyecto. Una vez obtenido el mapa del ambiente en 2 o 3 dimensiones, la detección de obstáculos podría implementarse para detectar posibles objetos y proporcionar información como la altura y posición de los objetos. La reconstrucción en tiempo real implicaría un ahorro de tiempo en la graficación del ambiente, debido a que el robot no tendría que realizar primero el

recorrido y después el envío de datos, sino que ambas tareas se realizarían al mismo tiempo.

## Referencias

[Alcala 2008] Lego Minsdstorms. Universidad de Alcalá. Notas de curso. Fecha de consulta: 30 de Mayo del 2008. Disponible en:

<http://www.Slideshare.net/naciendo/lego-nxt/>

[Bustillos 2004] Bustillos D. M. M., Campeche R. A. 2004. Control de un robot caminante aplicando algoritmos de búsqueda. Benemérita Universidad Autónoma de Puebla. No 1. 2004, Pág.140-151. Fecha de consulta: 7 de Mayo del 2009. Disponible en:

en: <http://www.revista-nanocinecia.ece.buap.mx>

[de Lope 2001] de Lope A. J. 2001. Emisión y recepción de infrarrojos. Universidad Politécnica de Madrid, Ingeniería de Sistemas Autónomos y Automática, documento de clase. Fecha de consulta: 24 de Julio del 2008. Disponible en:

<http://isa.umh.es/temas/micros/doc/infra.pdf>

[Hurtado 2008] Hurtado J. C. G. 2008. Sensores. Universidad del Valle. Notas de Curso. Fecha de consulta: 23 de Julio del 2008. Disponible en:

<http://190.6.172.20/Moodledata/40/Sensores.pdf>

[Martín 2006] Martín C. E., Asensio M. L. J. 2006. Servomotores y su control. Instituto Militar de Ingeniería. Notas de curso. Fecha de consulta: 10 de Agosto del 2008. Disponible en: [http://www.ime.eb.br/~pinho/micro/trabalhos/Robot\\_Bioins\\_I.pdf](http://www.ime.eb.br/~pinho/micro/trabalhos/Robot_Bioins_I.pdf)

[Modesti 2006] Modesti M. R. 2006. Sistemas Inteligentes. Universidad Tecnológica de Córdoba. Notas de curso de Robótica. Fecha de consulta: 23 de Mayo del 2008. Disponible en:

[http://www.proferores.frc.utn.edu.ar/Industrial/sistemasinteligentes/FFlexible/Robotica\\_2.pdf](http://www.proferores.frc.utn.edu.ar/Industrial/sistemasinteligentes/FFlexible/Robotica_2.pdf)

[Molina 2006] Molina S. Lastra O. R., C. 2006. Planificación de Rutas. Universidad de Chile. Notas de clase de Robótica Móvil. Fecha de consulta: 4 de Julio del 2008. Disponible en: <http://robotica.li2.uchile.cl/el710/Clase13.ppt>

[Ollero 2001] Ollero B. A. 2001. Robótica manipuladores y robots móviles. Alfaomega Marcombo.

[Palacios 2006] E. Palacios, F. Remiro. L.J. López. 2006. Microcontroladores PCI16f84. Alfaomega Ra-MA. Segunda edición.

[Pallás 2007] Pallás A.R. 2007. Sensores y acondicionadores de señal. Alfaomega.

[Pérez 2007] Pérez C. 2007. ¿Qué es el bluetooth?. Reporte técnico. Fecha de consulta: 24 de Julio del 2008. Disponible en: <http://tecnico.com/%c2%bfque-es-bluetooth/>

[Pozo 2007] Pozo R. A, Ribeiro A, García A., García L., Guinea D., Sandoval F.2007. Sistema de posicionamiento global (GPS). Instituto de Automática Industrial Consejo Superior de Investigaciones Científicas, Arganda Madrid. Notas de curso. Fecha de consulta: 28 de Julio del 2008. Disponible en: <http://www.iai.csic.es/users/gpa/postscript/Pozo-Ruz00a.pdf>

[Rivera 2008] A. Rivera, Diario el País, 2008. El robot 'Spirit' halla vestigios de fumarolas en Marte. Noticia Periódico. Fecha de consulta: 25 de Mayo del 2008. Disponible en: [http://www.elpais.com/articulo/sociedad /robot/Spirit/ halla/ vestigios/ fumarolas /Marte/elpepisoc/20080523elpepisoc\\_5/Tes](http://www.elpais.com/articulo/sociedad /robot/Spirit/ halla/ vestigios/ fumarolas /Marte/elpepisoc/20080523elpepisoc_5/Tes)

[Vélez 2003] Vélez S. J. F., Moreno D. A. B., Sánchez C. A. Sánchez M. J. L. E. 2003. Visión por computador. Dykinson. Segunda edición.