



Universidad Tecnológica de la Mixteca

**Diseño y Construcción de un Brazo Robótico
Pedagógico Jugador de Gato, dotado de un
Sistema Básico de Visión Artificial.**

T E S I S

Que para obtener el título de:

Ingeniero en Computación

Presenta

José Alberto Chávez Aragón

Trabajo dirigido por:

M.I.A. Héctor Gabriel Acosta Mesa

A Bibiana y Pepe
mi esposa e hijo quienes
siempre me han brindado
todo su amor

AGRADECIMIENTOS

Doy Gracias a Dios por la bendición de la vida.

Gracias a mis Padres: José Chávez Platas y Bertha Aragón de Chávez por su apoyo incondicional y gran amor, a Aracely y Laura por el cariño que me tienen.

Gracias a los profesores que me asesoraron y orientaron durante el desarrollo de este proyecto, gracias a los maestros: Enrique Guzmán Ramírez, Gerardo Cruz González., Mario Alberto Moreno Rocha, Sonia Mestizo Gutiérrez.

Gracias al Maestro en I.A. Héctor Gabriel Acosta Mesa y al Dr. José Baez Rojas quienes me guiaron y apoyaron en el desarrollo de este trabajo.

Gracias a todos mis compañeros y amigos que me ayudaron en la culminación de esta tesis
Gracias a Margarita Silva, Yanet López, Hernán González, Gilberto Alfaro, David Cruz, Fernando Iturbide.

U. T. M. 9950

Índice

INTRODUCCIÓN	1
CAPÍTULO 1 Inteligencia Artificial y Robótica	
1.1 Introducción	6
1.2 IA.....	8
1.3 Los problemas que aborda la IA.....	9
1.4 ¿Qué se considera una técnica de IA?.....	10
1.5 Robótica	12
1.6 Robots.....	14
1.7 Tipos de robots no industriales	18
1.8 Robots en la educación	19
1.9 Robótica Pedagógica	21
1.10 Brazo Robótico Pedagógico dotado de un Sistema Básico de Visión Artificial	24
CAPÍTULO 2 Diseño del Brazo Robótico y su Ambiente	
2.1 Introducción	30
2.2 Elementos y Articulaciones	30
2.3 Amplificadores Mecánicos y Motores DC	37
2.4 Interfaz con la PC	44
2.5 Software de Control del Robot Pedagógico	54
2.6 Tablero de Juego y Soporte del Robot Pedagógico	64
CAPÍTULO 3 Sistema básico de Visión Artificial	
3.1 Introducción	67
3.2 Visión Artificial	68
3.3 Adquisición de la Imagen	71
3.4 Segmentación de la Imagen	75
3.5 Reconocimiento de Patrones y Toma de Decisiones	83

U. N. 9950

CONCLUSIONES	87
--------------------	----

BIBLIOGRAFÍA	91
--------------------	----

APÉNDICE A Manual de operación del Brazo Robot

Componentes del Brazo Robot	94
Descripción	95
Instalación del Robot	95
Configuración de la cámara de vídeo	98
Software de operación del robot	98
Puesta en marcha del robot	103

APÉNDICE B Código Documentado de los Procedimientos y
Funciones más importantes del Sistema

Programa de Control del Brazo Mecánico	105
Algoritmos de Visión Artificial	115
Algoritmo del Juego del Gato	120

INTRODUCCIÓN

La **computación** esta inmersa en prácticamente todos los aspectos de nuestra vida, hoy en día las computadoras no solo son herramientas para resolver problemas, también las utilizamos como un medio de control, aprendizaje, comunicación y entretenimiento. A este gran cambio la sociedad le ha llamado la nueva revolución industrial que da paso a la sociedad de la Información.¹

Por otro lado apoyándose de la ciencia de la computación encontramos a la **Robótica**, que se encarga de la concepción y creación de **Robots**. Los robots son un conjunto de mecanismos automatizados capaces de realizar el tipo de tarea para la cual fueron creados, por tanto una de las disciplinas que ha tenido mayor crecimiento en los años recientes es la **Robótica Industrial**, que se encarga de diseñar y construir robots que auxilien o realicen parte del proceso de producción en la industria moderna, esto con el fin de eficientar los procesos productivos al mismo tiempo de que se eleva la calidad del producto y se reducen los costos de producción.

Si bien a los robots se les puede encomendar diversas tareas que pueden ser muy complicadas o peligrosas para el ser humano o aquellas que requieran de gran velocidad y precisión, a un robot todavía no se le puede encomendar la tarea de enseñanza, pero podemos auxiliarnos de ellos para transmitir conocimiento en

¹ [RUIZ-VELAZCO.1991]

áreas muy particulares como son: la electricidad, electrónica, mecánica, computación, programación, entre muchas otras más. De aquí nace una nueva disciplina la **Robótica Pedagógica**, que ayuda a los educandos en los procesos cognitivos¹, esto se logra despertando su interés por interactuar con un robot que generalmente es una reproducción a menor escala de aquellos que se utilizan en la industria.

Los investigadores, profesores y estudiantes aprenden mucho en la construcción de mecanismos robóticos y si además el robot se construye para ayudar a entender los conceptos de algunas otras disciplinas, imaginémonos pues el potencial de la robótica pedagógica.

Es por esto que, a manera de trabajo de Tesis para obtener el título de **Ingeniero en Computación** se decidió construir un robot pedagógico dotado de un sistema básico de visión artificial y cuyo objetivo es jugar correctamente el juego de tres en línea o como comúnmente se le conoce "**El juego del gato**" contra una persona.

La idea de la construcción de este robot surgió precisamente porque se tenía inquietud en desarrollar una aplicación en la que se pusieran en manifiesto las técnicas de visión artificial o visión robótica, además de obtener experiencia en el campo de la robótica y más específicamente en brazos robóticos. Una motivación más para desarrollar este proyecto es la dejar un antecedente en la construcción de una herramienta pedagógica robusta que puede ser útil tanto para profesores que se apoyen en él para reforzar el proceso de enseñanza, como para estudiantes de generaciones subsecuentes, ya que el brazo robótico diseñado podrá ser adaptado para realizar diversas tareas así como para simular algún proceso de producción.

Los objetivos generales de este trabajo son los de construir una herramienta pedagógica robusta, útil para apoyar el proceso de enseñanza-aprendizaje en

¹ Procesos de aprendizaje

conceptos relacionados con la computación, la electrónica y la robótica; como control de dispositivos periféricos, interfaces con la computadora a través del puerto paralelo, algoritmos básicos de visión artificial, entre otros más.

Una consideración importante que se tuvo para diseñar el robot es que, se diseñó y construyó casi en su totalidad con materiales muy comunes que pueden ser obtenidos fácilmente y a muy bajo costo en ferreterías, tiendas de electrónica, refaccionarias, etc.

Lo que hace que nuestro robot sea de muy fácil y económico mantenimiento

Tomando las consideraciones anteriores, se obtuvo como resultado el diseño de un robot cuyas características son las siguientes:

El centro de procesamiento y control del robot es una computadora personal, el actuador es un brazo mecánico de dos grados de libertad construido en aluminio y cuyos mecanismos están compuestos de amplificadores mecánicos formados por engranes, poleas y bandas, el poder de movimiento se lo proporcionan tres motores DC.

Su sensor principal es una cámara de video acoplada al centro de control del robot y dispuesta de tal manera que obtenga una imagen panorámica sobre el estado del juego en un momento determinado, el brazo robótico y la cámara están montados sobre un bastidor de fierro tubular que le da soporte a todos los componentes del robot y al tablero de juego.

El siguiente trabajo de reporte de tesis, consta de un total de 3 capítulos, y dos apéndices, estructurados de la siguiente manera:

En el primer capítulo titulado **Inteligencia Artificial y Robótica** se estudian los conceptos básicos de estas dos disciplinas enfocándose principalmente en

aquellos temas que por la naturaleza del problema son de mayor utilidad en su resolución.

En este primer capítulo se revisa a la Robótica Pedagógica, analizando esta nueva disciplina, mostrando sus capacidades y en menor medida sus limitantes, es también en este capítulo donde se propone la construcción de nuestro robot pedagógico justificando su construcción con base en su potencial educativo y flexibilidad como campo de experimentación y prueba.

El Segundo capítulo titulado **Diseño del Brazo Robótico y su Ambiente** es el espacio en donde se encuentra registrada la investigación que se hizo con respecto a las diversas arquitecturas de brazos mecánicos y con base en ésta, se diseñó un brazo tal que se adaptará a la resolución de nuestro problema y que además fuera lo suficientemente flexible para adaptarlo a muchas otras situaciones.

También aquí se acotó el problema del robot que juega al gato, delimitando perfectamente su tarea y por tanto, su funcionamiento y forma de interactuar con el jugador contrario. Para ello se diseñó el tablero de juego de tal manera que el brazo tuviera máxima movilidad sin perturbar la obtención de las imágenes y la interacción con su contrincante.

También en este capítulo en la sección **software de Control del Robot Pedagógico**, se da una explicación detallada del funcionamiento del programa de computadora que controla el brazo mecánico y la manera en como dicho programa debe compensar el desajuste natural del brazo debido a las características de sus motores y mecanismos de movimiento.

En el tercer capítulo titulado **Visión Artificial** se analiza esta rama de la Inteligencia Artificial, y nos adentramos a estudiar específicamente aquellas técnicas que se utilizaron en la resolución de nuestro problema, es aquí donde se

explican los algoritmos que se implementaron, gracias a los cuales nuestro robot puede “ver”.

Después del tercer y último capítulo se presentan las **Conclusiones** y algunas sugerencias para experimentar con el robot.

Al final de este documento se agregaron los Apéndices A y B, los cuales contienen el manual del usuario y el código documentado respectivamente.

En el Apéndice A, titulado **Manual del Usuario**, se da una explicación detallada de la manera en la que el robot debe ser operado correctamente, también aquí se discuten algunas consideraciones importantes en cuanto al acoplamiento de los diferentes subsistemas que conforman el robot, esto con la finalidad de dar al lector la información suficiente para que el robot pueda ser programado y adaptado para realizar tareas diversas, por último se dan algunos consejos sobre su mantenimiento.

En el Apéndice B, se lista el código documentado de las funciones y procedimientos más importantes de los diferentes programas.

Se espera le sea de utilidad el material que aquí se presenta, fruto de un intenso trabajo de investigación y experiencia propia adquirida con la experimentación en este fascinante mundo de la **Robótica**.

INTELIGENCIA ARTIFICIAL Y ROBÓTICA

1.1 Introducción

Desde siempre el hombre ha querido crear e imitar la majestuosidad de la naturaleza, desde antes de existir la computadora e inclusive antes de poder imaginar algún mecanismo complejo capaz siquiera de dar la mínima posibilidad de creatividad, el hombre ha deseado poder compartir la inteligencia con un subordinado encargado de hacer las tareas que al él le desagradan, por ser monótonas, complicadas o imposibles para el ser humano. Algunos ejemplos de autómatas a lo largo de la historia son: "Mi hija Francine" creada por René Descartes en 1649; En el siglo XVIII, Jacques de Vaucanson, fue el creador de una "oca mecánica" que imitaba a las orgánica casi a la perfección, aunque no tenía alguna aplicación útil sin duda era el comienzo de un viaje infinito por imitar el encanto de la vida orgánica.¹

Aunque todo parece indicar que la Inteligencia Artificial y la Robótica cambiarán la vida de la humanidad en su totalidad, hay que ser realistas ya que seguramente todavía no es el momento, si bien es cierto que la Inteligencia Artificial y la Robótica son disciplinas científicas que prometen mucho también son muy

¹ {R- PEDAGÓGICA. 1996}

recientes. En las escasas décadas de vida que tiene la Inteligencia Artificial ha conseguido pocos logros, sin embargo va cobrando personalidad y nos muestra un camino que espera ser explorado y explotado, solo nos resta trabajar y esperar el espectáculo.

Los conceptos modernos de control de los robots, así como los diversos sensores de visión artificial y táctiles tienen sus raíces en la investigación en Inteligencia Artificial (IA).

La IA trata de desarrollar sistemas que parezcan que se comportan de manera inteligente, estos trabajos se realizan como una parte de las ciencias de la computación aunque también tiene elementos de psicología, lingüística y matemáticas, para comprender mejor lo que es la IA y por la dificultad para definirla, es conveniente en este trabajo iniciar este primer capítulo con una introducción a la Inteligencia Artificial, revisar los problemas que aborda, ¿Qué se considera una técnica de IA?, para luego ver una de sus ramas : la Robótica.

Dentro de la disciplina de la Robótica en general, se revisarán sus orígenes, la evolución que han sufrido los robots a través de la historia y actualmente como se clasifican, enfocándose principalmente en los robots no industriales, y específicamente en los robots pedagógicos. Se estudiará como ayudan los robots en el proceso de enseñanza campo de una nueva disciplina: la Robótica Pedagógica, por último se propondrá la construcción del brazo robótico pedagógico dotado de un sistema básico de visión artificial, y se analizarán algunas de sus características.

1.2 Inteligencia Artificial

La Inteligencia Artificial es un término que hoy en día es comúnmente utilizado. Definirlo es una tarea muy complicada, algunas definiciones de la *Inteligencia Artificial (IA)* se presentan a continuación:

- 1) *"La IA, Estudia cómo lograr que las máquinas realicen tareas, que por el momento, son realizadas mejor por los seres humanos "*²
- 2) *"La IA es el arte de hacer máquinas capaces de hacer cosas que requerirían inteligencia en caso de que fuesen hechas por seres humanos"*³
- 3) *"La IA es el estudio de los cálculos que permiten percibir, razonar y actuar".*⁴

Si bien a las definiciones anteriores les hace falta mencionar que la IA también se encarga de problemas que no pueden ser resueltos adecuadamente ni por máquinas ni por hombres, como aquellos en donde la explosión combinatoria de los posibles caminos para obtener las soluciones es muy elevada y por tanto sobrepasa las capacidades de los hombres y de velocidad de las computadoras, nos dan una idea en que consiste la Inteligencia Artificial.

La Inteligencia Artificial tiene dos enfoques, el primero es el enfoque ingenieril y el segundo es el científico. El primer enfoque es útil para los ingenieros de la computación, que requieren conocer la IA para aplicar sus técnicas, de tal manera que les permitan hacer que las computadoras sean más eficientes en la resolución de problemas reales, comúnmente lo logran utilizando ideas acerca de cómo

² [RICH,1994], pp 3.

³ {AIR,1998}

⁴ [WINSTON,1994], pp 15.

representar y utilizar el conocimiento. Y el enfoque científico es útil para los psicólogos, biólogos, filósofos y lingüistas que utilizan la IA para tratar de entender los principios de la inteligencia, ya que las teorías de IA derivadas pensando principalmente en las computadoras en gran medida sugieren una guía útil para entender el pensamiento humano y otros tipos más de inteligencia.

La IA trata de abarcar y estudiar muchas de las capacidades del hombre para poder ofrecérselas a la máquina y también al propio hombre en el entendimiento de los principios de su inteligencia, de aquí se puede entender el por que la IA tiene diversas ramas de estudio como son: Agentes, Lógica, Redes Neuronales, Sistemas expertos, Vida artificial, Tecnología del habla y síntesis, Visión artificial Y Robótica, entre otras.

Son estas dos últimas ramas de la IA las que se estudiarán más detalladamente en este trabajo, ya que diversos conceptos de ellas se aplicaron en la construcción del brazo robótico pedagógico dotado de un sistema básico de visión artificial.

1.3 Los problemas que aborda la IA

Las primeras incursiones dentro de la Inteligencia Artificial se enfocaron principalmente a las tareas formales como la demostración de teoremas y juegos tales como el juego de las damas y el ajedrez, ya que los juegos y la demostración de teoremas son tareas en las que es necesaria la inteligencia para llevarlas a cabo, se pensó al principio que ésto era sencillo, bastaba con programar a la computadora para que buscara todos los posibles caminos para solucionar estos problemas y escoger el mejor, sin embargo ésto es falso, ya que no existen computadoras lo suficientemente rápidas para explorar la explosión combinatoria que generan la mayoría de estos problemas.

En la clase de problemas en los que se requiere el sentido común para resolverlos también la IA incursionó en un principio y dio como resultado los resolvers de problemas como el GPS (General Problem Solver), que resolvía problemas tomando como base la forma en como los humanos lo hacemos, construido por Newell, Shaw y Simon. El GPS finalmente solo podía resolver tareas muy simples.

La IA ha encontrado nuevas áreas de investigación de las cuales destacan: las tareas de percepción como la vista y el habla, la comprensión de lenguaje natural y los sistemas expertos que son capaces de resolver problemas en campos especializados de la actividad humana.

1.4 ¿Qué se considera una técnica de IA?

Hasta el momento no hay una definición precisa de lo que es una Técnica de IA, según Elaine Rich, *una técnica de IA es un método que utiliza conocimiento representado de tal forma que:*

- *El conocimiento represente las generalizaciones, es decir agrupar las situaciones que comparten propiedades importantes en lugar de representar situaciones individuales.*
- *Debe ser comprendido por las personas que los proporcionan.*
- *Puede modificarse fácilmente para corregir errores y reflejar los cambios del mundo.*
- *Puede usarse en diferentes situaciones aún cuando no sea totalmente preciso o completo.*

- *Puede usarse para ayudar a superar su propio volumen, ayudando a acotar el rango de posibilidades que normalmente deben ser consideradas*⁵

Para entender mejor lo anterior veamos un ejemplo.

Una computadora puede ser programada para que juegue correctamente el juego del gato o tres en línea, dotando a la computadora con todas las posibles combinaciones del juego es decir 3^9 tableros diferentes con todas las jugadas, o mejor aún realizar un algoritmo con diversas bifurcaciones en las que el programa tendrá que probar muchas condiciones sugeridas por el programador con base en su experiencia en el juego pero, estas dos técnicas para resolver el problema como es evidente no cumplen con las condiciones para ser consideradas técnicas de IA, ya que las soluciones están estrechamente ligadas con el problema.

¿Qué sucedería si se cambiara el problema a 3 dimensiones?. Es obvio que no funcionarían estas técnicas, tendrían que ser rediseñadas en su totalidad.

Una técnica de IA que resuelve este problema es el algoritmo **minimax**. Este algoritmo verifica diversas secuencias de movimientos para encontrar la que lleve a la victoria, es decir, intenta maximizar la posibilidad de victoria suponiendo que el oponente intentará minimizar dicha posibilidad, esto lo hace con una estructura de datos que representa al tablero, y por medio de la cual construye un árbol de búsqueda, y al ser evaluadas cada una de las ramas de este árbol el algoritmo determina cual es la mejor tirada. Mediante este procedimiento se puede construir un jugador experto del gato y además en el caso de que se aumenten las dimensiones del tablero solo se le harían pequeñas modificaciones para que pueda seguir funcionando apropiadamente.

⁵ [RICH.1994], pp 8.

1.5 Robótica

Los orígenes de la robótica se encuentran en la ciencia-ficción. Ya que ha contribuido en gran medida al desarrollo de la robótica, implantando ideas innovadoras en mentes creativas, creando el deseo de conocimiento de esta nueva tecnología, esto lo podemos ver en novelas como Frankenstein de Mary Shelley, publicada en Inglaterra en 1817, Rossum's Universal Robots publicada en 1917 por Karel Capek, que es precisamente esta obra la que dio lugar al término robot, también Isaac Asimov contribuyó con varias narraciones sobre robots en 1939. Asimov en alguna de sus obras hace aparecer una máquina bien diseñada y con seguridad garantizada, que actúa según las tres leyes de la robótica que él mismo escribió y son:

- 1. Un robot no puede actuar contra un ser humano o, mediante la inacción, permitir que un ser humano sufra daños*
- 2. Un robot debe obedecer las órdenes dadas por los seres humanos, salvo que estén en conflicto con la primera ley.*
- 3. Un robot debe proteger su propia existencia, a no ser que este en conflicto con las dos primeras leyes.*⁶

La robótica se ha fundamentado en algunos desarrollos tecnológicos implementados a través de los años. En los siglos XVII y XVIII se crearon algunos dispositivos mecánicos que tenían ya algunas características de los robots actuales, en el siglo XVII fueron construidos músicos de tamaño humano por Jacques de Vaucanson. Henri Maillardet construyó una muñeca mecánica, cuya tarea era la de realizar dibujos, durante la revolución industrial también

⁶ [GROOVER, 1990]

fueron creados ingeniosos artefactos principalmente en el sector textil, podemos citar la hiladora giratoria de Hargreaves(1770), la hiladora mecánica de Crompton (1779), los telares mecánicos de Jacquard (1801) y Cartwright (1785).

Ya en años recientes la robótica se ha desarrollado gracias a dos tecnologías importantes, el control numérico y la telequérica.

El control numérico, que como su nombre lo indica es el control de las acciones de una máquina por medio de números, su antecedente es el uso de tarjetas perforadas para el control de una herramienta desarrollado originalmente por John Parsons a finales de los años 40 y principios de los 50, aunque el telar mecánico de Jacquard y el ejecutor de obras de piano de 1876 ya utilizaban una cinta de papel perforada a manera de programa.

La Telequérica, es la utilización de un manipulador remoto controlado por un ser humano, para transmitir los movimientos del operador humano en los correspondientes movimientos en la posición remota, esto es comúnmente utilizado en el manejo de sustancias peligrosas y radioactivas.

La conjunción de estas dos tecnologías es lo que constituye la base para el robot moderno.

Actualmente la Robótica es una ciencia multidisciplinaria, ya que ésta hace uso de los recursos que le proporcionan otras ciencias afines para diseñar e implementar mecanismos robóticos, debido a que en el proceso de diseño y construcción de un robot intervienen muchos campos pertenecientes a la mecánica, la electrónica, la computación, las matemáticas, entre muchas otras.

La robótica es una tecnología que surgió como tal aproximadamente en el año 1960, desde entonces han pasado pocos años, sin embargo el interés que ha despertado es muy superior a cualquier previsión.⁷

La robótica más o menos ha seguido un proceso paralelo al desarrollo de las computadoras, si bien es cierto que todavía los robots no han penetrado en los hogares de las personas, también es cierto que se han vuelto indispensables en un número cada vez más grande de industrias.

La robótica como ciencia ofrece un gran campo de desarrollo a investigadores y aficionados, que motivados por la innovación proponen robots cada vez más evolucionados. Son precisamente los aficionados los que partiendo de sus propuestas e ideas firmes e innovadoras los que han dado gran auge y desarrollo a la robótica.

Por otro lado la ingeniería mecánica también ha aportado una serie de avances importantes en la construcción de motores, de sistemas de transmisión y de sistemas hidráulicos que aunados al desarrollo de la tecnología de las computadoras han podido contribuir a la construcción de los robots que son utilizados actualmente.

1.6 Robots

La palabra robot fue inventada por el checoslovaco Karel Capek en los años veinte, la cual significa trabajar. Los robots son máquinas que pueden trabajar sin que una persona las opere todo el tiempo.

⁷ {AIR,1998}

Sin embargo esta definición puede ser confusa, ya que entonces una simple máquina lava ropa podría ser un robot.⁸

A continuación se presentan algunas definiciones de los tipos más comunes de robots que existen.

Robot: *“Es un manipulador reprogramable de uso general con sensores externos que pueden efectuar diferentes tareas de montaje”.*⁹

Robot: *“Es un dispositivo mecánico el cual puede ser programado para realizar algunas tareas de manipulación o locomoción mediante un control automático”*

Robot Industrial: *“Es un manipulador reprogramable y multifuncional diseñado para mover materiales, partes, herramientas o dispositivos especializados, a través de movimientos programados variables para la realización de una diversidad de tareas”*

Robot toma y deposita: *“Es un robot simple generalmente con dos o tres grados de libertad el cual transfiere diferentes objetos de un lugar a otro, por medio de movimientos de punto a punto, puede o no contener un pequeño control de trayectorias”*

Manipulador: *“Es un mecanismo que generalmente consiste en una serie de segmentos articulados o deslizantes uno con respecto a otro, con el propósito de tomar y mover objetos a menudo en varios grados de libertad. Puede ser controlado remotamente por una computadora o un humano. En el caso de que sea controlado por un humano, nos indica que no es automático ”*

⁸ [FU,1990]

⁹ [FU,1990], pp1

Robot Inteligente: *“es un robot el cual puede ser programado para tomar decisiones eventuales con base en sus entradas sensitivas”*

Robot de parado estático: *“Es un robot cuyo punto de control es fijo pero no así su control de trayectorias, esto es, cada uno de los ejes, tiene determinado unos límites finales en los cuales debe detenerse, así que no puede detenerse excepto es los límites establecidos. De tal manera que un robot de n grados de libertad puede detenerse en no más de $2n$ localidades (cuando la localidad incluye posición y orientación). Con frecuencia se pueden obtener muy buenas secuencias repetitivas con un robot de parado estático”*

Androide: *“Es un robot cuya apariencia es semejante a la de un ser humano”*

Robot controlado por sensores: *“Es un robot en el cual la secuencia de su programa puede ser modificada en función de la información obtenida del ambiente, por medio de sus sensores, puede ser servomotorizado o no”*

Robot de ciclo abierto: *“Es un robot el cual no tiene incorporada la retroalimentación, es decir no tiene un mecanismo de comparación entre la salida y el comando de entrada de posición actuales”*

Robot móvil: *“Es un robot el cual esta montado en una plataforma móvil”*

Robot de grados de libertad limitados: *“Es un robot capaz de posicionar y orientar el efector final en menos de seis grados de libertad”¹⁰*

Las anteriores definiciones dan una idea más clara a cerca de lo que es un robot y de los diversos tipos que existen de ellos, además, sugiere que un robot debe

¹⁰ [SANDLER, 1991]

poseer inteligencia, que se debe principalmente a los algoritmos de una computadora asociados con su sistema de control y sensorial.

Una manera práctica entonces de entender lo que es un robot es: Los robots pueden ser vistos como un conjunto de mecanismos automatizados que son capaces de realizar el tipo de tarea para la cual han sido creados.

Generalmente los robots son fabricados para hacer aquellas tareas que resultan complicadas o peligrosas para el ser humano y en aquellas áreas en las que la velocidad y la precisión de los mecanismos automatizados superen a las habilidades del hombre.

Una de las disciplinas que ha tenido mucho éxito en los últimos años es la robótica industrial, la cual se encarga de diseñar y construir robots cuya misión es acelerar diversos procesos de producción, al mismo tiempo que aumentan la calidad en el producto final.

En la construcción de robots se deben considerar dos aspectos importantes:

- El robot debe mantener un movimiento preciso en condiciones que varían es decir en un ambiente dinámico.
- El robot debe ejecutar una secuencia de operaciones previamente determinadas es decir, un programa o un plan.

1.7 Tipos de robots no industriales

En esta sección se revisarán a grandes rasgos algunos tipos de robots que no entran dentro de la familia de robots industriales, solo se mencionarán sus principales características.

Robots móviles: *Estos dispositivos tienen un gran rango de aplicaciones, que con frecuencia son de naturaleza no industrial, A los robots móviles podemos clasificarlos en términos de la manera en la cual obtienen su propulsión, que pueden ser por ruedas, orugas u otros. Un robot móvil puede ser controlado de diversas maneras:*

Remotamente: por medio de alambres, cables o radio.

Automáticamente: (autónomo), o por un programa..

Guiados: por rieles, por mecanismos ópticos o magnéticos.

Los robots móviles se utilizan en situaciones peligrosas o ambientes hostiles como debajo del agua, en zonas contaminadas radioactivamente, biológicamente o en el espacio.

Exoesqueletos: *Estos dispositivos ayudan y en su caso aíslan protegiendo al usuario en situaciones peligrosas como pueden ser en el espacio o debajo del agua, también pueden servir para ayudarlo al multiplicar su fuerza muscular, ya que el usuario se enfunda dentro de estos dispositivos y ellos actúan como amplificadores de fuerza mediante la utilización de servomotores o pistones hidráulicos.*

Máquinas caminantes: *Los vehículos que utilizan ruedas para desplazarse, tienen el inconveniente de que requieren que existan caminos previamente establecidos por los que ellos deben transitar, este inconveniente es superado por los robots caminantes, que son máquinas que tratan de simular la técnica de propulsión que utilizan los animales principalmente los insectos, y de esta manera la máquina pueda moverse en todo terreno con la utilización de patas, este tipo de máquinas son mucho más difíciles de construir que aquellas que utilizan ruedas para desplazarse, se han construido vehículos de 1, 2, 3, 4, 6 y 8 patas como intentos para solucionar el problema del caminado automático artificial, existen prototipos de robots con patas para la exploración lunar entre otras aplicaciones más en donde se requieren robots todo terreno.*

Prótesis: *Son dispositivos de control automático que intentan reemplazar a algunos miembros humanos amputados, se trata de construir dispositivos, que sean cada vez más compactos y ligeros, además de que el tiempo que transcurre desde que el usuario envía un comando y se ejecute la acción asociada a él, en el dispositivo sea muy pequeño, el camino más eficiente para lograr esta meta es la de tomar la bioelectricidad de los músculos del usuario asociados al movimiento como comandos de control, es decir estos dispositivos toman estas señales, las amplifican, las analizan y las transforman en la acción deseada en el dispositivo mecánico.¹¹*

1.8 Robots en la Educación

En la actualidad existe un problema complejo en el ámbito educativo, el gran rezago que existe en la educación en áreas como matemáticas, ciencias y tecnología, además de la disociación entre los conocimientos de ciencia y

¹¹ [SANDLER, 1991]

tecnología, incrementado por las relaciones entre el estudiante y el profesor, que a menudo son manejadas casi en su totalidad por el maestro, lo cual se ve reflejado en la poca o nula retroalimentación por parte del alumno, no permitiéndole al estudiante aflorar su creatividad, mismo que se tiene que conformar con demostraciones experimentales guiadas por el profesor o realizar prácticas siguiendo la metodología de una receta de cocina siendo esta la única relación entre el alumno y el objeto científico o tecnológico.

Los robots pueden ser de gran utilidad en diversos procesos de la enseñanza, permitiendo la retroalimentación por parte del alumno hacia el maestro, además de poner al alumno en contacto directo con el objeto tecnológico.

En función del objetivo para el cual fue creado el robot, puede auxiliar en diversas ramas del saber o en alguna en particular, y considerando que para su construcción se requirieron de conceptos de muchas disciplinas como son: electrónica, mecánica, computación, matemáticas, entre otras, también podemos auxiliarnos de tales mecanismos para transmitir conocimiento y conceptos propios de cada uno de las disciplinas involucradas en su desarrollo.

El primer robot que se construyó con una finalidad exclusivamente educacional, fue un robot móvil que poseía una cámara de televisión, detectores de choque y enlace radial con una computadora. Este robot fue construido por el Stanford Research Institute en 1968.

TOPO es el nombre de un robot controlado por computadora por medio de un enlace infrarrojo, construido por Nolar Bushnell, y cuya finalidad también fue puramente educativa.

Existen diversos tipos de robots educacionales entre los que destacan: Los brazos robots, los robots de suelo, y los robots virtuales.

Cada tipo de robot apoya en la tarea educacional de una forma específica, por ejemplo, con un brazo robot educacional los estudiantes pueden experimentar simulando procesos de producción, programándolos para que realicen una serie de movimientos predeterminados, además si el robot posee sensores que le ayuden a interactuar con el mundo, podría inclusive tomar ciertas decisiones en un mundo cambiante.

En la clasificación de robots de suelo se encuentran todos aquellos robots que tienen la capacidad de desplazarse por una superficie ayudados por ruedas, orugas o patas, son dispositivos muy divertidos y didácticos, generalmente estos robots son controlados por computadora, y soportan un lenguaje de programación por medio del cual se programan las trayectorias que deben seguir, ejemplos de estos robots son: la tortuga LOGO, El Piper mouse, y la Tortuga Valiant.

Los robots virtuales son programas de computadora que apoyan en la enseñanza de conceptos de la programación estructurada y orientada a objetos, ya que cuentan con un conjunto de instrucciones, condiciones, así como estructuras de control, también tienen la capacidad de aprender tareas hechas con base en su lenguaje, un ejemplo de esto es el sistema KAREL.

1.9 Robótica Pedagógica

La robótica pedagógica es una disciplina que surgió recientemente con la finalidad de favorecer los procesos de aprendizaje (cognitivos) de los educandos al permitirles interactuar con robots en su educación.

Una definición de robótica pedagógica es la siguiente:

*“Es la actividad de concepción, creación y puesta en funcionamiento, con fines pedagógicos, de objetos tecnológicos que son reproducciones reducidas muy fieles y significativas de los procesos y herramientas robóticas que son usados cotidianamente, sobre todo, en el medio industrial”.*¹²

Como se puede ver en la definición anterior, la robótica pedagógica se encarga primeramente del diseño, creación y puesta en funcionamiento de un robot, la segunda actividad de la que se encarga la robótica pedagógica, es la de verificar que realmente el artefacto construido cumpla con los fines pedagógicos para los que fue creado. En estas dos actividades se fundamenta principalmente esta nueva disciplina.

La robótica pedagógica puede ayudar en diversos niveles de educación, desde el jardín de niños, hasta la formación de profesionistas, e inclusive puede ayudar a los discapacitados en su rehabilitación ayudándoles a tener una vida más productiva valiéndose o adiestrándolos con robots educativos o terapéuticos.

La robótica pedagógica en niveles de primaria y secundaria, ayuda a los niños a entender algunos conceptos de matemáticas y programación utilizando instrumentos que despierten su interés y curiosidad por aprender, no obligándolos a memorizar conceptos abstractos sin que ellos puedan ver una aplicación práctica, aquí la idea entonces es construir robots que les permitan aprender jugando, los robots que utilizan a esta edad generalmente son de suelo, los niños aprenden a programarlos cada vez con mayor complejidad, también son

¹² [VIVET, 1989]

muy usados los robots virtuales que llevan al niño de la mano para aprender nuevos conceptos valiéndose de novedosas técnicas multimedia.

En la formación profesional también son muy útiles los robots pedagógicos, es evidente que son muy importantes para los estudiantes de ingeniería, y por supuesto, para todas aquellas carreras que tienen que ver con la computación, ya que los estudiantes aparte de aprender conceptos nuevos se familiarizan con la tecnología que encontrarán en la industria una vez que salgan al campo laboral, pero no solo estudiantes de estos tipos de carreras se ven beneficiados, si no en general casi en cualquier área del saber se pueden utilizar los robots pedagógicos. En el trabajo Robótica y Medicina¹³ de Luis Guillermo Pedraza Moctezuma y Francisco Javier Fernández, se describe un Robot-Workstation, este robot fue desarrollado para enseñar a los estudiantes de medicina las funciones cerebrales, robots similares ayudan a entender también la mecánica de la contracción muscular entre muchos otros conceptos y procedimientos médicos.

Otro enfoque que le han dado los investigadores a esta nueva disciplina de la robótica pedagógica, es la de ayudar a las personas discapacitadas construyendo robots que les faciliten la realización de sus actividades diarias, ayudándoles a tener una actividad remunerada o ser educacionales y/o terapéuticos, tal es el caso de Michel Gilbert y Richard Howell que han trabajado en el diseño de robots manipuladores como apoyos cognitivos y físicos para estudiantes ortopédicamente discapacitados.

Pero no todo es positivo en la robótica pedagógica, también existen ciertas desventajas, entre ellas debemos considerar que el hecho de solo interactuar con esta tecnología no apoya tanto en los procesos de aprendizaje, como si se involucrara desde el diseño y construcción del robot, otra desventaja es que:

¹³ [RUIZ-VELAZCO, 1991]

Aún para construir un robot muy sencillo se requiere de conocimientos profundos en muchas ramas de la ciencia, de la misma manera que se necesita de usuarios especializados o con cierto tipo de conocimiento en el área de aplicación del robot. Sin embargo como ya hemos visto antes, el potencial de la robótica pedagógica para apoyar en el proceso de enseñanza es muy grande y en gran medida suple estas desventajas.

Recapitulando, la robótica pedagógica ayuda y estimula la adquisición de conocimientos aligerando la tarea de enseñanza-aprendizaje en diferentes áreas y a diferentes niveles, la robótica pedagógica ha probado su efectividad en todas ellas, desde los niños que aprenden estimulados por el deseo de jugar, hasta los obreros que se capacitan con robots reflejando los beneficios en un mejor nivel de vida. Es una excelente cama de pruebas para simular el comportamiento de diversos sistemas, procesos de producción, entre muchos otros propósitos didácticos.

1.10 Brazo Robótico Pedagógico Dotado de un Sistema Básico de Visión Artificial

Con fundamento en las ventajas estudiadas en las secciones anteriores que proporciona la robótica pedagógica, se propuso el desarrollo y puesta en funcionamiento de un brazo robótico dotado de un sistema básico de visión artificial, como apoyo a estudiantes de ingeniería en electrónica y computación en la enseñanza de conceptos tales como: control de dispositivos periféricos, interfaces con la computadora a través del puerto paralelo, algoritmos básicos de visión robótica, etc. Estos temas son tratados en las materias de: arquitectura de microcomputadoras, arquitectura de computadoras, inteligencia atificial, sistemas expertos y robótica, materias que forman parte del plan de estudios de las

carreras de ingeniería electrónica y computación de la UTM. Una vez que el estudiante interactúe con el robot de manera práctica, podrá programarlo para que ejecute diferentes secuencias de movimientos, esto fortalecerá al alumno en la comprensión de conceptos tales como planificación de tareas, y planificación de trayectorias. Una vez que es enfrentado el estudiante con problemas reales que es imposible encontrarse en la teoría, lo obligará a proponer soluciones prácticas.

El robot propuesto de inicio está programado para jugar el juego del Gato e interactuar con un jugador contrario, esta aplicación se eligió porque en ella se utilizaron muchas de las características de nuestro robot, ya que con esta tarea el manipulador mueve piezas en un área determinada, y su sistema de visión es útil para identificar las piezas del jugador contrario. Sus algoritmos controladores están acoplados a las salidas del algoritmo del juego del gato implementado con técnicas tradicionales de programación.

Se iniciará revisando las diversas anatomías para los brazos robóticos, a continuación se presentan las más comunes:

14

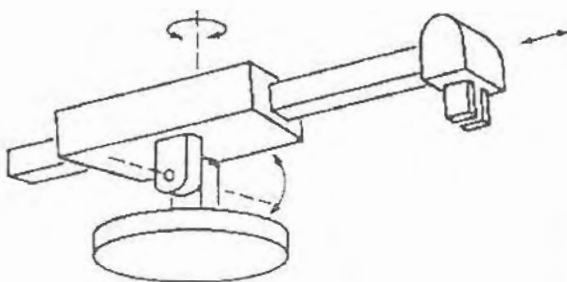


Figura 1.1 Configuración Polar

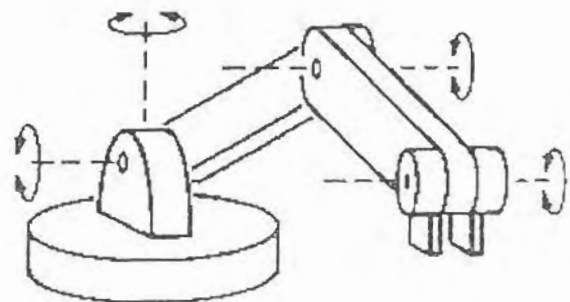


Figura 1.2 Configuración de Brazo Articulado

¹⁴ [GROOVER, 1990]

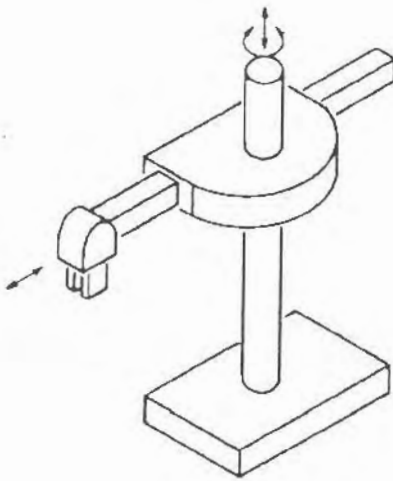


Figura 1.3 Configuración Cilíndrica

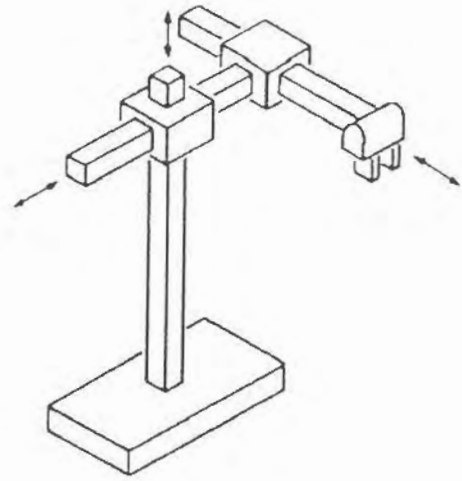


Figura 1.4 Configuración de Coordenadas Cartesianas

Por los objetivos que se persiguen con el robot en cuestión, se eligió para su construcción, la configuración de Brazo articulado, esta configuración es similar a la del brazo humano, está constituida por dos componentes rectos que corresponden al antebrazo y brazo humanos montados sobre un pedestal vertical. Estos componentes están conectados por dos articulaciones giratorias que corresponden al hombro y al codo. Una muñeca esta unida al extremo del antebrazo, con lo que se proporcionan varias articulaciones suplementarias.

A los movimientos de las articulaciones individuales se les denomina grados de libertad, y los movimientos del robot se realizan por medio de articulaciones accionadas.

El robot propuesto cuenta con 5 grados de libertad, sin embargo solo 2 articulaciones son accionadas con motores DC, las correspondientes al hombro y al codo, lo que le permite al robot moverse con 2 grados de libertad, el resto de las articulaciones pueden moverse manualmente, motorizarlas puede ser un gran ejercicio para aquellos que trabajen con este robot pedagógico.

Acoplado a la muñeca está el efector final, que en este robot es una pinza que le permite realizar la tarea de tomar las fichas para moverlas a través del tablero de juego, el efector tiene acoplado un tercer motor para sus funciones de abrir y cerrar la pinza.

A manera de efectores finales, existen diversas articulaciones de pinzas entre ellas se destacan las siguientes.

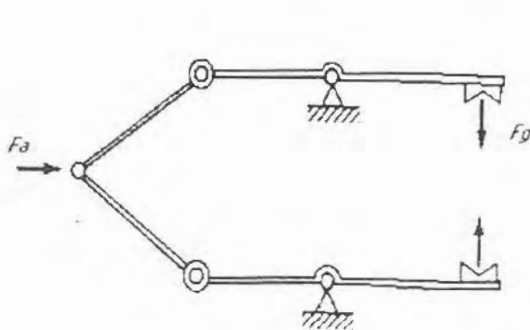


Figura 1.5

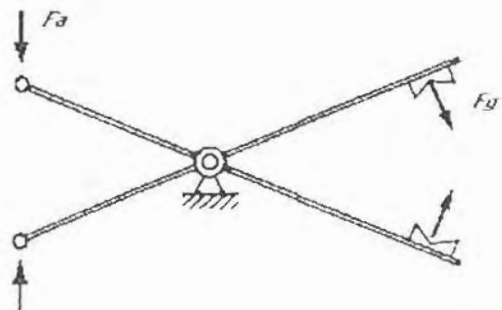


Figura 1.6

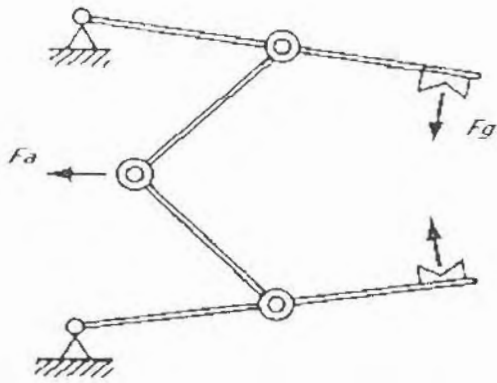


Figura 1.7

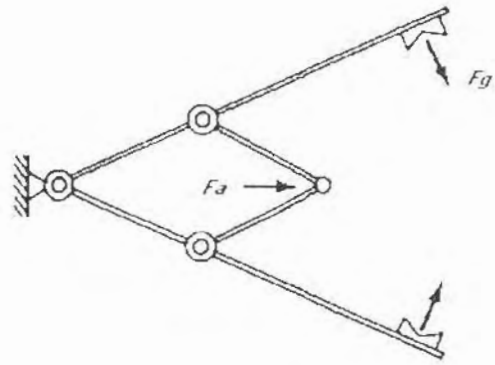


Figura 1.8

Para el robot pedagógico se utilizó una configuración de efector final similar al mostrado en la figura 1.6. Con una pequeña modificación en el pivote que produce un movimiento inverso al mostrado en la figura, obtuvimos entonces un efector como el que se muestra en la figura 1.9.

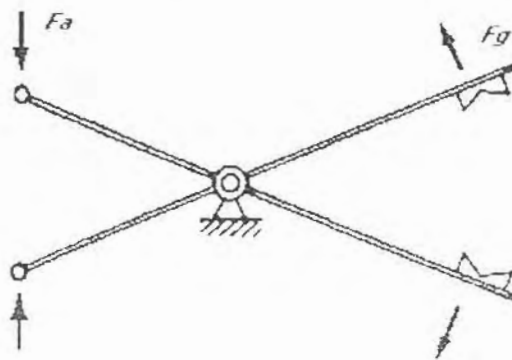


Figura 1.9

El brazo robótico está montado sobre una base translúcida de acrílico parte de la cual es el tablero de juego, un soporte tubular mantiene fijos el tablero, el brazo robot y una pequeña cámara de video, la cámara capta la imagen del tablero desde una perspectiva inferior, es decir toma imágenes de las jugadas, a través del tablero translucido, por tanto la cámara esta montada por debajo del tablero, las imágenes posteriormente son analizadas por el sistema básico de visión del robot, y con base en los resultados toma la decisión de realizar el mejor movimiento de las piezas de juego. El robot mueve piezas en forma de cruz y el usuario piezas circulares.

Hasta aquí se ha visto a grosso modo algunas características físicas del robot pedagógico que se ha propuesto, a manera de introducción a los siguientes capítulos, en los cuales se tratará más a detalle estas características y se seguirá paso a paso la manera en como se llevó a cabo la construcción del manipulador pedagógico dotado de un sistema básico de visión artificial.

DISEÑO DEL BRAZO ROBÓTICO
Y SU AMBIENTE

2.1 Introducción

En este capítulo se tratará el diseño del brazo robótico pedagógico dotado de un sistema básico de visión artificial. Se revisará la anatomía física del robot, sus elementos, articulaciones y amplificadores mecánicos, todo esto apoyándose en esquemas para una fácil interpretación. Se revisará la manera de como se acopló con una PC por medio del puerto paralelo para su control por software, y la manera en como programarle una secuencia de movimientos, por último se revisará la estructura que sujeta al robot y sus diferentes componentes, además del tablero del juego por medio del cual se acotó perfectamente el problema del robot jugador de gato.

2.2 Elementos y Articulaciones del Robot

A la construcción física del cuerpo, brazo y muñecas de un robot, se le denomina: **Anatomía del Robot**. Generalmente los robots utilizados en las fábricas están montados sobre una base que esta sujeta al suelo, el cuerpo esta unido a la base y el conjunto del brazo esta unido al cuerpo.

Al final del brazo está la muñeca que está constituida por varios componentes que le permiten orientarse en una diversidad de posiciones.

Los movimientos relativos entre los diversos componentes del cuerpo, brazo y muñeca son proporcionados por una serie de articulaciones. Estos movimientos de las articulaciones implican deslizamientos o giros.

El robot pedagógico que se propone se pretende que sea una reproducción a pequeña escala de un robot industrial.

En el capítulo anterior se revisaron las diferentes configuraciones de los brazos robóticos así como los diferentes tipos de pinzas a manera de efectores finales del brazo, lo que correspondería a la mano en un brazo humano, y debido a los objetivos del robot pedagógico que juega al gato, se eligió la configuración de brazo articulado y un tipo especial de pinza para tomar las fichas de juego, estos elementos se consideraron debido al **Volumen de Trabajo** que proporciona esta configuración.

El término **Volumen de Trabajo** se refiere al espacio dentro del cual el robot puede manipular el extremo de su muñeca, con la configuración de brazo articulado el volumen de trabajo puede ser hasta una esfera, claro está, esto depende del tamaño de los componentes del cuerpo, brazo y muñeca. Así como de los límites de movimiento de las articulaciones del robot.

Debido a que en principio el robot se mueve automáticamente con dos grados de libertad, el volumen de trabajo es cilíndrico como el que se muestra en la figura 2.1. Ésto es cierto, si se considera que a diferentes aperturas de la pinza el brazo puede tomar objetos a diversas alturas. Sin embargo tendrá la capacidad de lograr un volumen de trabajo semiesférico como en la figura 2.2, una vez que el

resto de sus articulaciones sean motorizadas, ésto puede hacerse en un futuro en cumplimiento del objetivo pedagógico del robot.

Volúmenes de Trabajo

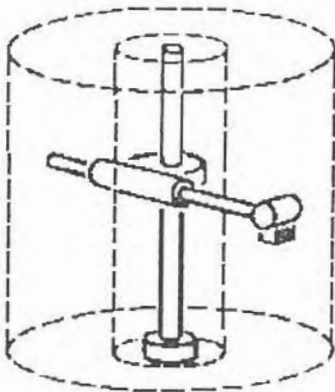


Figura 2.1 *Cilíndrico*

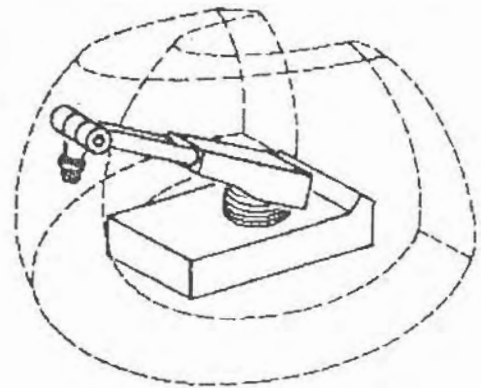


Figura 2.2 *Esfera parcial*

El movimiento de los robots se hace por medio de articulaciones accionadas, generalmente tres articulaciones suelen estar asociadas con la acción del brazo y del cuerpo, y dos o tres articulaciones se suelen emplear para el movimiento de la muñeca.

Las **Uniones** son elementos rígidos que se emplean para la conexión de las diversas articulaciones del robot. En una cadena de unión-articulación-unión, se le denomina unión de entrada al eslabón que está más cerca de la base del robot, a la otra unión por consiguiente se le llama unión de salida. La unión de salida es la que se desplaza con respecto a la unión de entrada.

Las articulaciones del brazo y del cuerpo del robot, le permiten desplazar su efector final a la posición deseada, siempre que se encuentre entre los límites permitidos para el robot. Para los robots con configuración polar, cilíndrica o de brazo articulado, los movimientos del brazo y del cuerpo son:

1. **Transversal vertical:** Es la capacidad para desplazar la muñeca hacia arriba o hacia abajo para proporcionar la postura vertical deseada.
2. **Transversal radial:** Implica la extensión o retracción (movimiento hacia adentro o afuera) del brazo desde el centro vertical del robot.
3. **Transversal rotacional:** Es la rotación del brazo alrededor del eje vertical¹

Con respecto al movimiento de la muñeca, esta suele tener tres grados de libertad, éste es lo que le permite al efector final orientarse adecuadamente para tomar o manipular la pieza de trabajo, los movimientos típicos de la muñeca son:

1. **Giro de la Muñeca:** También denominado oscilación de la muñeca, implica la rotación del mecanismo de la muñeca alrededor del eje del brazo.
2. **Elevación de la Muñeca:** Si el giro de la muñeca está en su posición central, la elevación implicaría la rotación arriba o debajo de la misma, a este movimiento también se le denomina flexión de la muñeca.
3. **Desviación de la muñeca:** Si el giro de la muñeca está en su posición central, la rotación implicaría, la rotación a derecha o izquierda de la muñeca.²

Hasta el momento se ha visto que se pueden describir las características físicas de un manipulador robótico, con base en su configuración que a su vez tiene asociado un volumen de trabajo.

¹ [GROOVER, 1990]

² [GROOVER, 1990]

Esquema de la anatomía del robot propuesto.

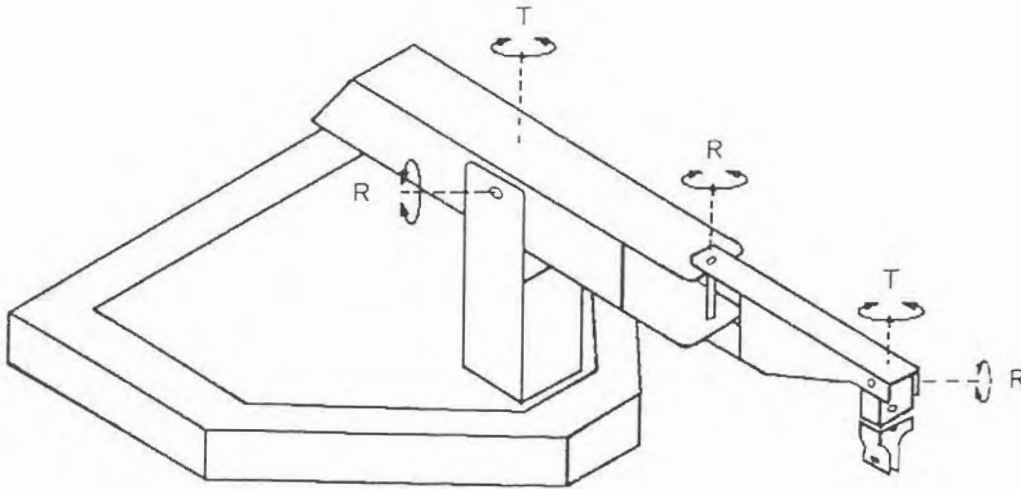


Figura 2.7 Anatomía del brazo robótico

Uno de los objetivos que se persiguió en el diseño del robot pedagógico, es que fuera de fácil construcción, con materiales comunes, además de que debía ser ligero y resistente, es por esto, que los elementos del manipulador robótico que se propuso fueron construidos completamente en aluminio, material que es resistente y ligero, el brazo se construyó de aluminio tubular cuadrado de 1 ½ pulgadas, el antebrazo fue hecho en aluminio tubular de ½ x 1 ½ pulgadas, la base es de ángulo de aluminio de 1 ½ pulgadas, la muñeca, los soportes para motores y cajas de engranajes son de lámina de aluminio, los engranes son de plástico, y los ejes de fierro soldados con bronce. Para unir los diferentes componentes de los elementos se usaron remaches pop de aluminio de 1/8 de pulgada.

2.3 Amplificadores Mecánicos y Motores DC

Para proporcionar la fuerza motriz real de las articulaciones del robot se utilizaron motores de corriente directa, debido a que requieren de un mínimo mantenimiento y son fácilmente controlables, sin embargo como es muy difícil encontrar un motor que cumpla con las características de velocidad y fuerza exactas para mover cada una de las articulaciones, fue necesario utilizar algún tipo de transmisión de potencia cuya función como su nombre lo indica, es transmitir la potencia a una distancia determinada y actuar como un transformador de potencia. Existen varias formas de efectuar la transmisión de potencia mecánica, estas formas incluyen las poleas y bandas, cadenas y ruedas dentadas, los engranes, los ejes de transmisión y los tornillos.

Para el sistema de transmisión del robot se utilizaron poleas, bandas, y engranes, debido a que estos elementos son fáciles de conseguir en el mercado y económicos.

Las poleas con bandas y los engranes funcionan de manera similar, ambos se utilizan para transmitir un movimiento giratorio desde una eje a otro. Esta transferencia puede realizarse entre ejes paralelos, ejes de intersección o ejes sesgados, el tipo de engranajes más sencillos son los de transmisión entre ejes paralelos y estos son precisamente los que se utilizaron mayormente en este proyecto.

Las poleas y bandas a diferencia de los engranes se utilizan casi exclusivamente en la transmisión de movimiento entre ejes paralelos, y su uso se hace frecuente cuando se requiere transmitir movimiento entre dos ejes cuya distancia de separación es grande. El tiempo de respuesta en la transmisión del movimiento de un eje al otro dependen de factores tales como: la tensión de la banda y las

características físicas tanto de las bandas como de las poleas, también las bandas y poleas son útiles como mecanismos de seguridad, ya que estos dispositivos son tolerantes a fallas, como en el caso en que uno de los ejes se atorara no necesariamente el otro también lo hace, en este caso la banda patina en la polea y protege al resto del sistema de una tensión excesiva.

Un tren de engranajes típico de ejes paralelos es mostrado en la figura siguiente.

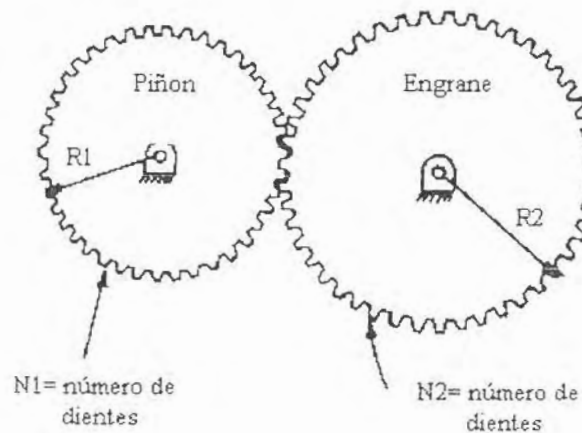


Figura 2.8 *Tren de engranajes rectos*

En el caso del tren de engranajes mostrado en la figura anterior, el engrane impulsor es el más pequeño, y se conoce como piñón y al otro se le llama engrane conducido, por ejemplo si el piñón es $1/3$ de tamaño del engrane conducido, por cada revolución del piñón el engrane conducido da $1/3$ de revolución, por ello a este tipo de tren de engranaje se le llama reductor de velocidad. Puesto que la velocidad se reduce un tercio, el par motor o la fuerza con la que se mueve el eje del engrane conducido es multiplicada por tres, la potencia de salida del tren de engranajes sigue siendo la misma.

El número de dientes de un engrane es proporcional a su diámetro, suponiendo que N_1 es el número de dientes del piñón y N_2 el del engrane conducido, la relación de engranajes esta dada por:

$$n = N_1/N_2$$

La velocidad de salida con respecto a la entrada es:

$$\omega_0 = n\omega_m$$

En donde ω_0 la velocidad de salida y ω_m es la velocidad de entrada. El par motor de salida es:³

$$T_0 = - \frac{T_m}{n}$$

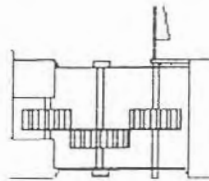


Figura 2.9 Vista del tren de engranajes de la articulación codo

En la figura 2.9 se puede observar parte de la caja de engranes que da el movimiento al codo de nuestro robot, esta caja esta compuesta por 6 engranes, el piñón es movido por el motor usando un par de poleas y una banda.

³ [GROOVER, 1990]

Lo que hacen en conjunto los engranes, es reducir la velocidad, de tal manera que el movimiento del antebrazo sea controlable.

Los engranes que se utilizaron para el tren de engranajes, se consiguieron de mecanismos de juguetes de desecho, así como de grabadoras y estéreos.

En el robot pedagógico las articulaciones correspondientes al hombro y al codo obtienen su fuerza motriz de 2 pequeños motores DC a los que llamaremos motor tipo "A", uno en cada articulación acoplados a un sistema de tracción compuesto de engranes, poleas y bandas, las características de estos motores se muestran a continuación:

Corriente (mA)	Torque (g/cm)	RPM (max)	Longitud del eje (pulg)	Tamaño (Diam" x L")	Peso (lbs)
70	24	5200	0.329	0.778 x 1.078	.046

Sentido de rotación	Rango de voltaje	Diámetro del eje
Reversible	1.5-3 VDC	0.78"

Tabla 2.1

Por otra parte, el efector final que es una pinza, obtiene su fuerza motriz de un tren de engranajes movido por un motor DC que llamaremos motor tipo "B", con las características siguientes:

Corriente (mA)	Torque (g/cm)	RPM (max)	Longitud del eje (pulg)	Tamaño (Diam" x L")	Peso (lbs)
70	17	4800	0.336	0.819 x 1.121	.046

Sentido de rotación	Rango de voltaje	Diámetro del eje
Reversible	1.5-3 VDC	0.78" ^{mm}

Tabla 2.2

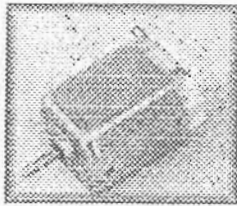


Figura 2.10 Motor tipo "A"



Figura 2.11 Motor tipo "B"

Enseguida se presenta una vista lateral del robot, en la cual se puede observar el detalle del movimiento de la pinza, la ubicación de los motores y los trenes de engranes, así como su apariencia definitiva.

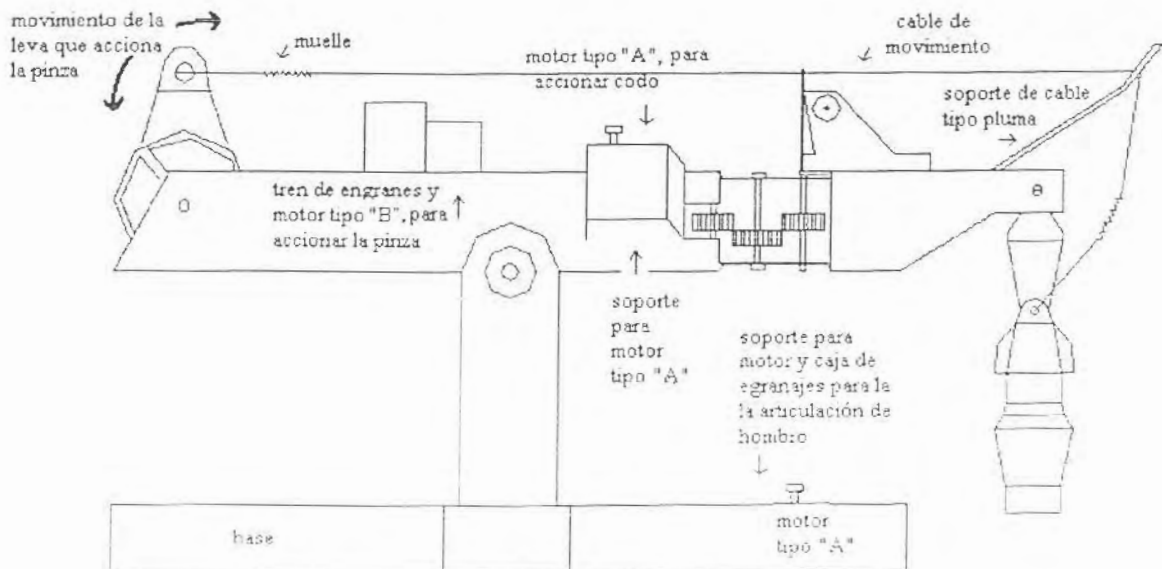


Figura 2.12 Vista lateral del brazo

La caja de engranes que mueve la articulación del hombro tiene una configuración similar a la del codo, la única diferencia es que el eje de salida tiene acoplado una polea más, que a su vez transmite el movimiento a la articulación en cuestión, esto hace que el movimiento del hombro sea más lento, pero con mucha más fuerza que el codo, ya que debemos tener en cuenta que esta articulación permite orientar todo el brazo.

Por último en esta sección se muestran tres vistas diferentes del brazo mecánico con sus respectivas cotas expresadas en metros, para apreciar las dimensiones reales que tiene, se aprecia también, parte de su base y el tablero de juego.

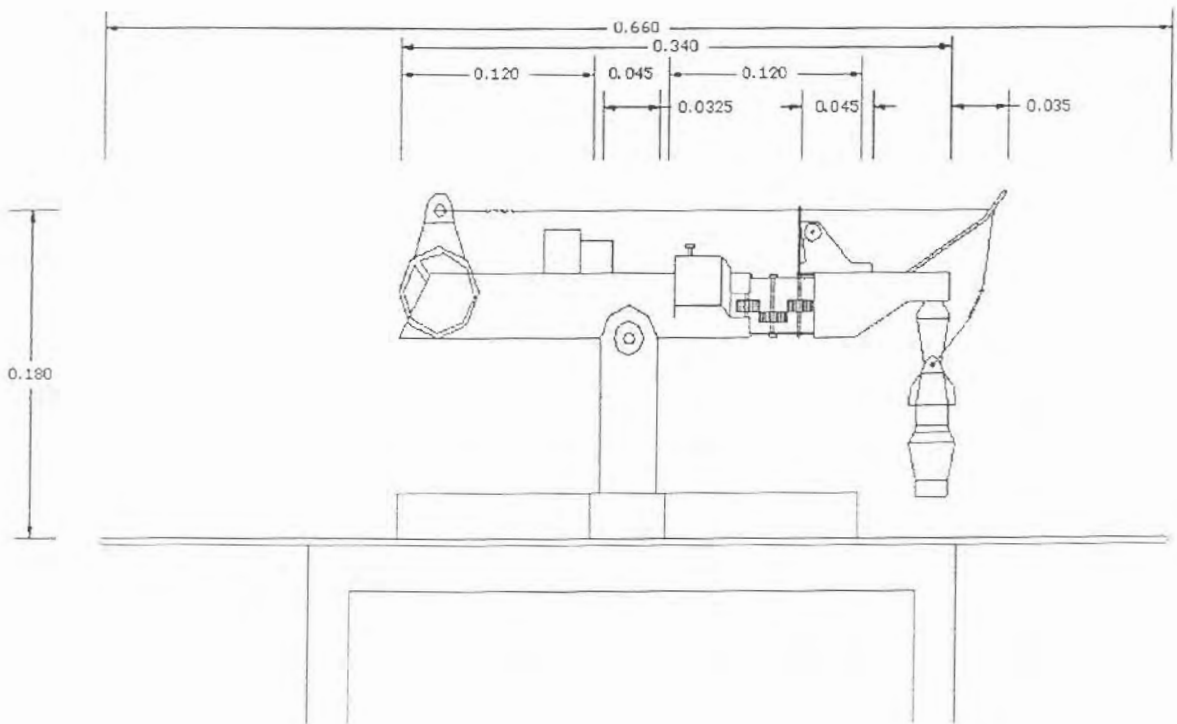


Figura 2.13 Vista lateral del brazo con cotas

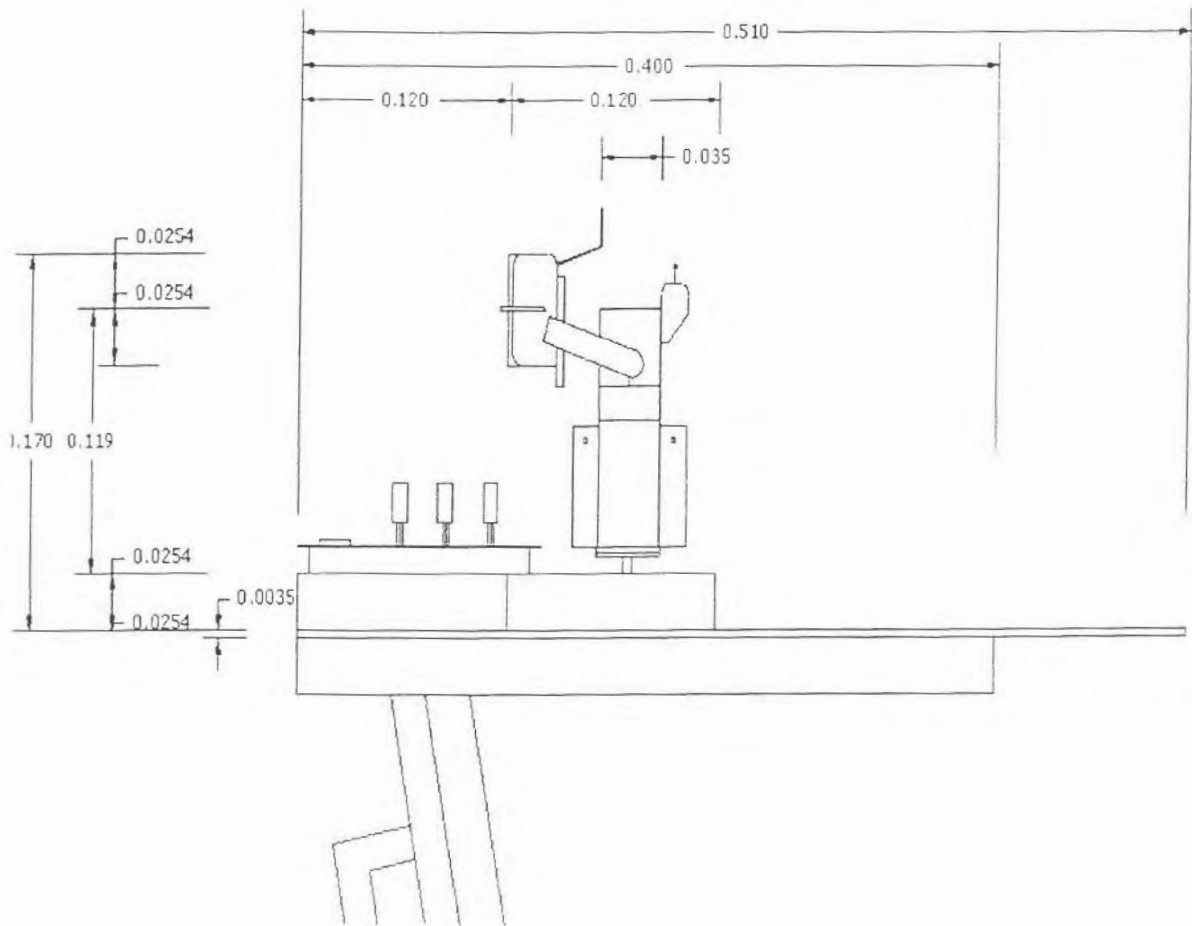


Figura 2.14 Vista posterior del brazo con cotas

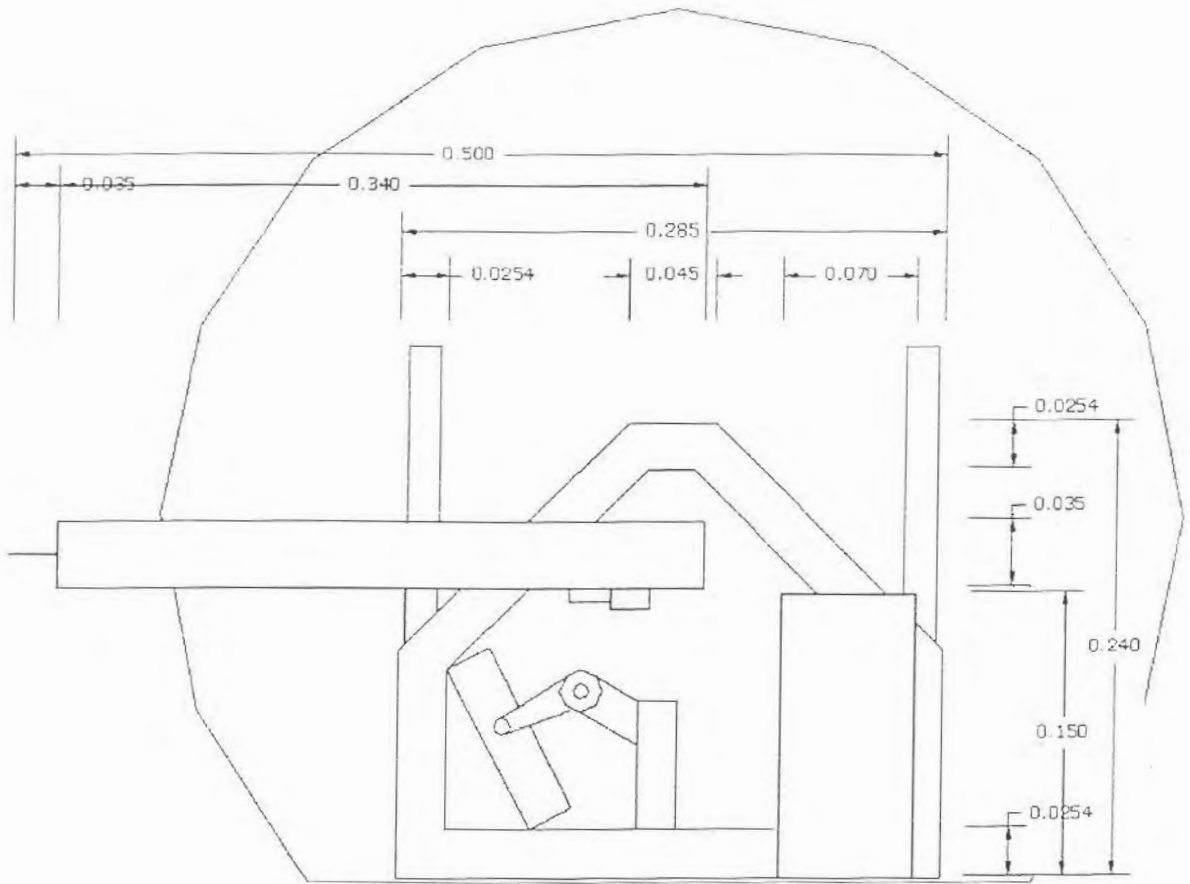


Figura 2.15 Vista superior del brazo con cotas

2.4 Interfaz con la PC

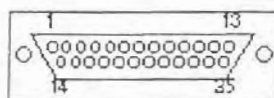
Se eligió al puerto paralelo como el dispositivo de entrada/salida por medio del cual se controla el robot pedagógico, debido a que el puerto paralelo es una poderosa puerta de salida, útil para implementar proyectos de control real de periféricos externos de la computadora.

El puerto de impresión tiene 8 salida de nivel TTL, 5 entradas y 4 pines bi-direccionales de control.

De los 25 pines que tiene el DB-25 macho, las salidas de datos del puerto se encuentran desde el pin 2 (data 0) hasta el pin 9 (data 7). Las salidas de control se encuentran en los pines 17, 16, 14 y 1. Los pines 17, 14, y 1 están invertidos, es decir trabajan con lógica negativa.

Las salidas del puerto son de 5 Volts valor compatible con TTL, lo que hace aún más fácil la aplicación de este puerto en proyectos de control de periféricos de la PC. La salida de datos es un número entero desde 0 a 255 ya que la puerta paralela es de 8 bits. Es importante conectar la tierra de la PC, a la tierra del circuito que se desee implementar

Las entradas que se pueden leer desde el conector DB25 son 5, los pines 11, 10, 12, 13 y 15; desde el más significativo al menos significativo. Estos bits corresponden desde el bit7 al bit3 respectivamente, ya que de los primeros tres bits, 2 están reservados y el otro restante es del IRQ. El bit7 es el único que trabaja con lógica negativa.



DB-25 Conector macho

Figura 2.16

El paquete completo de funciones del puerto se reparte en tres direcciones: la primera de datos, la segunda de estado, que es de entrada y, el puerto de control. Estas direcciones están en orden secuencial. Algunas posiciones típicas del puerto paralelo se muestran en la siguiente tabla.

Impresora	Puerto de Datos	Puerto de Estado	Puerto de Control
LPT1	0X03bc	0X03bd	0X03be
LPT2	0X0378	0X0379	0X037A
LPT3	0X0278	0X0279	0X027A

Tabla 2.3

En el proyecto, lo que se necesita para controlar el robot pedagógico es escribir al puerto de la impresora un byte de datos durante un cierto tiempo, cada uno de los movimientos del brazo se asoció a un byte diferente como se muestra en la siguiente tabla.

Movimiento del brazo	Binario	ASCII
Hombro izquierda	01100000	'
Hombro derecha	01010000	P
Codo izquierda	01000010	B
Codo derecha	01000001	A
Abre pinza	01001000	H
Cierra pinza	01000100	D
Parada	01000000	@

Tabla 2.4

La palabra que se debe escribir para realizar cada uno de los movimientos del brazo está íntimamente relacionado con la construcción del circuito de control de los motores, que está acoplado al puerto paralelo, dependiendo de la palabra que se escriba, se debe activar o desactivar cierto motor, así como también la palabra indica el sentido en el cual debe rotar el mismo.

Para escribir al puerto paralelo se utilizó el lenguaje de programación C, sobre plataforma DOS, la instrucción para escribir por ejemplo la palabra 01000100 en la puerta de datos 0X0378 es:

```
int port = 0x378;
int value = 'D';
outport(port,value);
```

Cuando se escribe una palabra en el puerto paralelo, dicha palabra permanece constante en las salidas del puerto, hasta que se escriba una nueva palabra, por ejemplo, si se ejecutara en un programa las líneas en C escritas anteriormente, y si se asumiera que el circuito controlador de los motores está acoplado al paralelo, el motor asociado a la pinza de robot giraría en el sentido tal que le permitiera a la pinza cerrarse durante un tiempo indefinido, este problema se soluciona con el código siguiente:

```
int port = 0x378;
int value = 'D';
int tiempo =500;
outport(port,value);
delay(tiempo);
value = '@';
outport(port,value);
```

En el código anterior la palabra asociada al movimiento de cerrar pinza estará en la salida del puerto escrita solamente 500 milisegundos, al término de los cuales se escribe una nueva palabra, que indica al circuito que se deben apagar todos los motores, esta es la manera fundamental de cómo se controlan todos los movimientos del robot pedagógico.

Como podemos darnos cuenta, para controlar el robot pedagógico solo es necesario escribir la palabra asociada al movimiento que se desea que el robot realice e indicarle al programa el tiempo que debe estar prendido el motor correspondiente, dependiendo del alcance y de la duración del movimiento.

Siguiendo esta lógica se escribió un programa en lenguaje C, para DOS llamado **puerto.exe**, que recibe como parámetros de la línea de comandos, el movimiento que se desea hacer, la duración del movimiento, y el puerto al cual se desea escribir, la sintaxis de este programa es:

Puerto *movimiento tiempo puerto_de_datos archivo* ←

En donde cada uno de los parámetros puede tomar los valores que se muestran en las siguientes tablas.

Movimiento	Movimiento del robot
0	Parada
1	Hombro izquierda
2	Hombro derecha
3	Codo izquierda
4	Codo derecha
5	Abre pinza
6	Cierra pinza
7	Ejecuta secuencia de un archivo

Tabla 2.5

El parámetro **tiempo** debe estar expresado en milisegundos.

Puerto de datos	Puerto de datos
1	0x378
2	0x278

Tabla 2.6

El parámetro **archivo**, es el nombre del archivo que contiene una secuencia de movimientos preestablecida, este parámetro es tomado en cuenta por el programa cuando el primer parámetro es 7, en otro caso no se toma en cuenta, pero debe ponerse de cualquier forma.

Por ejemplo, si se quiere que el robot mueva el elemento asociado a la articulación codo hacia la derecha por medio segundo y si la puerta de datos es 0x378, se debe escribir en la línea de comandos:

Puerto 4 500 1 X ←

Si el primer parámetro es 7 entonces los otros dos parámetros carecen de importancia, ya que el parámetro 7, indica que el programa ejecutará una secuencia de movimientos preestablecidos en un archivo por el usuario, el nombre de este archivo debe expresarse en el último parámetro y éste debe estar ubicado en el mismo directorio del programa puerto.exe, el hecho que los parámetros tiempo y puerto_de_datos, no tengan importancia en este caso, no significa que no se deban de escribir.

El archivo de secuencia de movimientos contiene una serie de movimientos que el robot ejecutará, la sintaxis de las instrucciones de este archivo no se revisarán por el momento, ya que la interfaz final del robot se encarga de escribir dicho archivo presentando al usuario una interfaz amigable, en donde se pueda programar una secuencia de movimientos del robot, este programa final está programado sobre

ambiente Windows y DOS, y utiliza para escribir al puerto el programa `puerto.exe`, sin embargo la interfaz del usuario se revisará en la sección de “*software de control del robot pedagógico*”, de este mismo capítulo, por el momento se revisará el circuito controlador de los motores del robot.

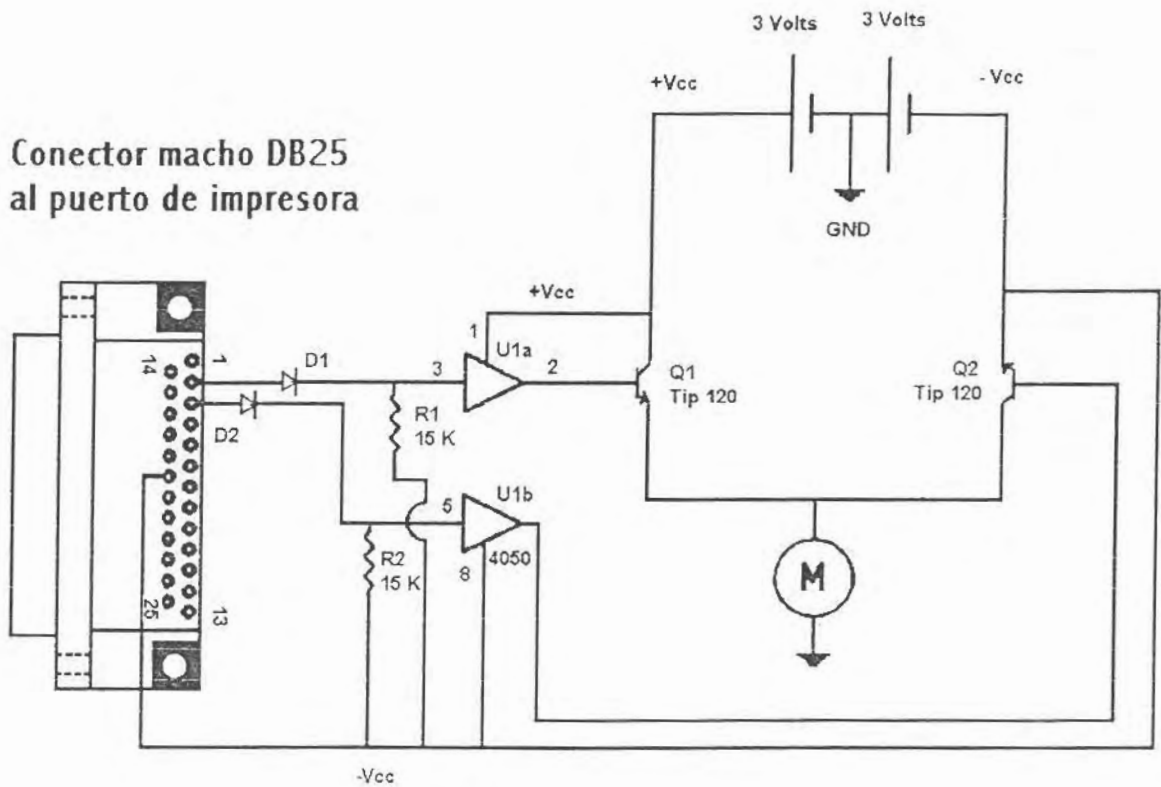


Figura 2.17 Circuito de interfaz para controlar un motor DC

El circuito electrónico del esquema anterior permite controlar un motor DC, al enviar una palabra al puerto paralelo, en este caso el circuito se activa una vez que exista una señal en cualquiera de los bits menos significativos, del byte

de datos, ya que las entradas del circuito están conectadas al pin2, y pin3, Data 0 y Data 1 respectivamente.

Por ejemplo si se escribe al puerto la palabra 01000010, que equivale en código ASCII a 'B', el circuito activa el motor en una dirección, ya que en el pin2 existen 0 volts y en el pin3, 5 volts, si se deseara cambiar el sentido de la rotación del motor se debería escribir 01000001 ó por su código ASCII 'A'.

Como mecanismo de protección para el puerto de la impresora se utiliza un par de diodos switch y un buffer MC14050, también hay que tener en cuenta que es necesario una fuente bipolar de ± 3 volts.

El circuito anterior, se puede ver como una caja negra o un bloque, que tiene como entradas dos señales provenientes del DB25, la tierra que se conecta al pin19, dos salidas que se conectan al motor DC, y se alimenta con una fuente bipolar de ± 3 volts y 1Amp, como se muestra en la figura siguiente.

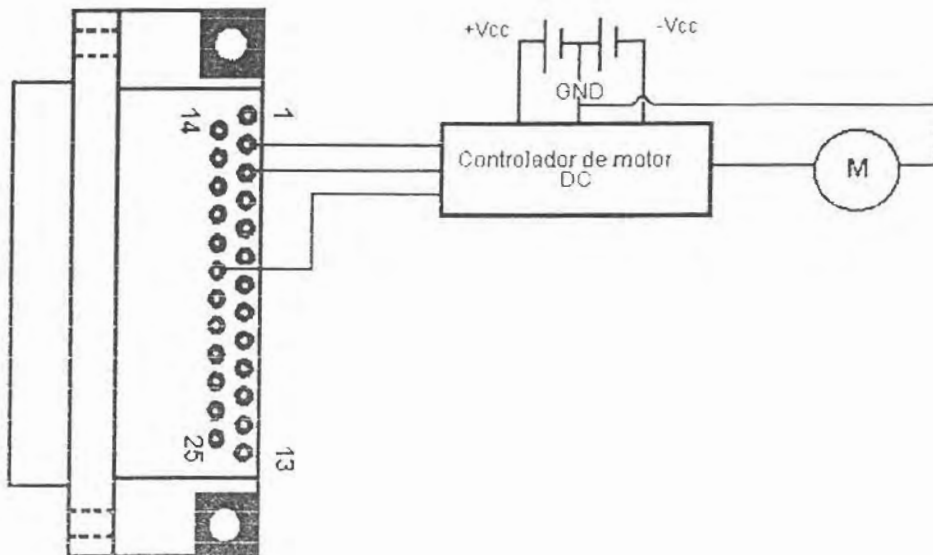


Figura 2.18 Circuito de interfaz para controlar un motor DC

Como ya se ha mencionado el motor se activa cuando se escribe al puerto de la impresora una 'A' o una 'B', entonces este motor es el que proporciona la fuerza motriz al codo del robot, si se agregaran dos bloques más, se obtendría una interfaz para controlar tres motores DC como se ve en el esquema siguiente.

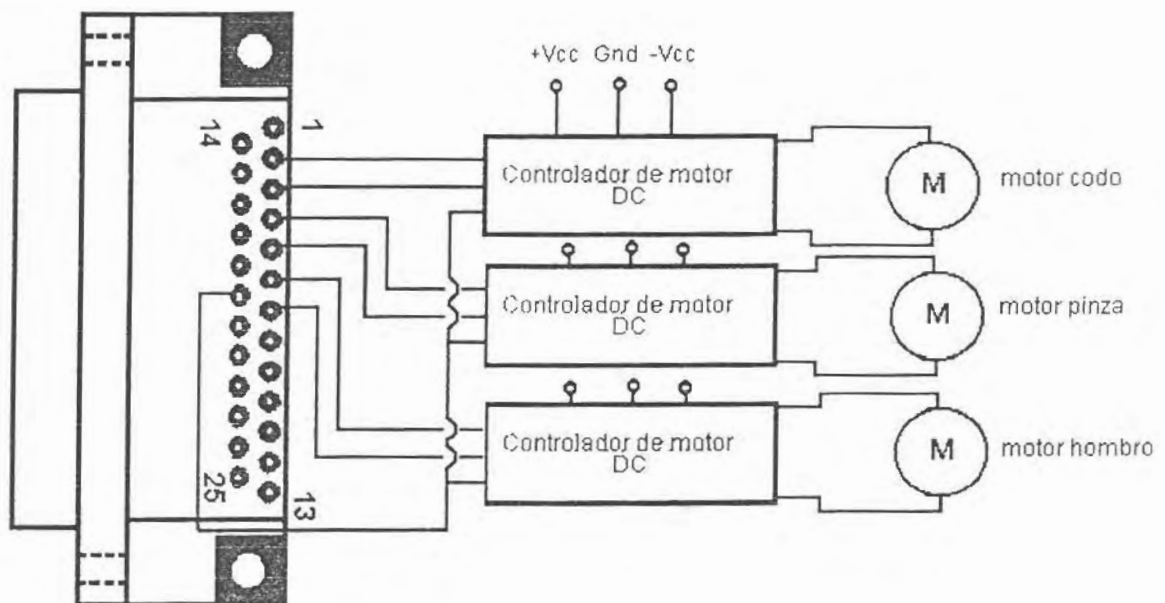


Figura 2.19 Circuito de interfaz del robot pedagógico

El esquema anterior muestra al circuito de interfaz que se utilizó para controlar el robot pedagógico, a continuación se encuentra la lista de los componentes electrónicos que fueron necesarios en la construcción de dicho circuito.

- 6 Transistores NPN, Tip 120
- 6 Resistencias 15K-ohms, ¼ Watt

- 6 Diodos 1N914 ó equivalente
- C.I. MC14050, hex noninverting buffer
- Conector DB25 macho
- Una tarjeta perforada
- Una fuente bipolar variable

Para alimentar el circuito fue necesario construir una fuente de alimentación bipolar, con un regulador de voltaje ajustable a ± 3.5 volts, voltaje al cual trabaja adecuadamente la interfaz. La figura 2.20 muestra el circuito de la fuente que se implementó.

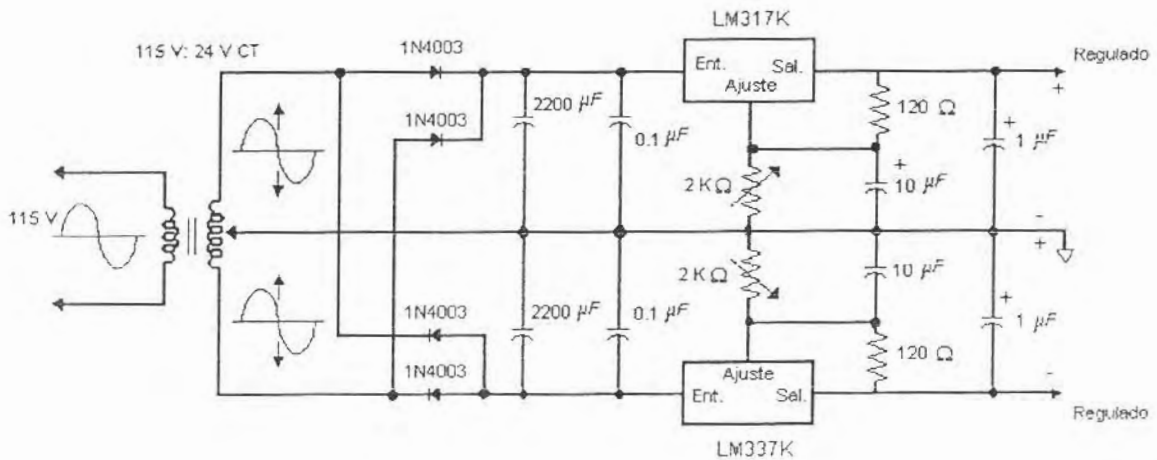


Figura 2.20 fuente de alimentación bipolar ajustable

Es importante mencionar, que antes de iniciar a operar el robot pedagógico, es recomendable verificar los niveles de voltaje de la fuente, que se debe encontrar operando en ± 3 volts.

2.5 Software de Control del Robot Pedagógico

Para controlar de manera eficiente el brazo robótico, y poder encomendarle tareas repetitivas, como es el caso de poner piezas de juego en nueve diferentes casillas de un tablero de gato, fue necesario desarrollar un software en el cual se pudieran programar al robot una secuencia de movimientos, y que éste la repitiera cada vez que fuese necesario. Para esto lo primero que se hizo fue definir una posición inicial o un punto de salida a partir del cual el robot deberá iniciar cualquier secuencia de movimiento, y al término de dicha secuencia retornar a esa posición inicial.

Antes de iniciar cualquier tarea se debe llevar al robot a su posición inicial, para asegurarse que se llevó al robot a la posición correcta, se instaló en el lado izquierdo de la base del robot, dos topes (*para el hombro y codo*) y un sensor de tacto, a los cuales el robot debe llegar, una vez que se conduzca hasta chocar con los topes, se asegura que se encuentra en la posición inicial, el sensor que está acoplado al segundo tope indica cuando el antebrazo está alineado con el resto del brazo, es decir el brazo está completamente estirado.

Si el robot está en cualquier otra posición diferente a la inicial, la secuencia que se debe seguir para conducir el brazo a la posición inicial debe ser la siguiente.

1. *Abrir la pinza para evitar cualquier choque con objetos bajos, o en nuestro caso las piezas de juego.*
2. *Mover el elemento asociado al codo, es decir el que corresponde al antebrazo hacia la derecha, de tal manera, que el antebrazo se encuentre doblado en un ángulo de unos 45 grados con respecto al resto del brazo.*

3. Mover la articulación hombro hacia la izquierda hasta llegar al tope designado para el hombro.
4. Mover el antebrazo lentamente hacia la izquierda, hasta llegar al tope designado para él, una vez que ésto pase, el sensor acoplado a este tope cortará la corriente del motor del codo y por tanto hay más posibilidades de que el brazo esté en su posición inicial con el antebrazo perfectamente bien alineado al resto del brazo.

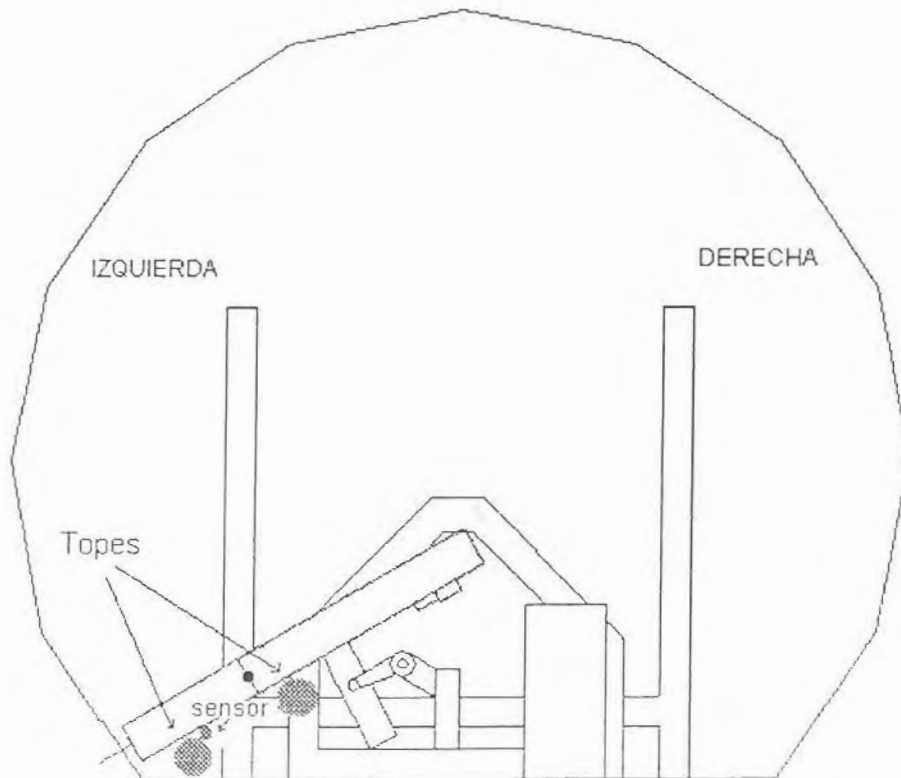


Figura 2.21 Vista superior del brazo en su posición inicial

El mecanismo del sensor no permite que una vez que el hombro está en su posición inicial, el antebrazo vaya más allá de lo necesario para alinearse con el resto del brazo, inclusive si el programa continuara enviando la señal para moverlo.

El programa que controla el robot pedagógico se llama *control_robot.exe* está programado sobre plataforma Windows, y se puede crear un acceso directo a él en el escritorio de Windows. La ruta es: *c:\brazo_robot\programa\control_robot.exe*.

Para guiar al robot a su posición inicial, en el menú principal del programa de control del robot, existe el submenú *brazo robot*, se debe llamar a la opción *secuencia de ajuste*, de dicho submenú, la cual presentará la ventana siguiente.

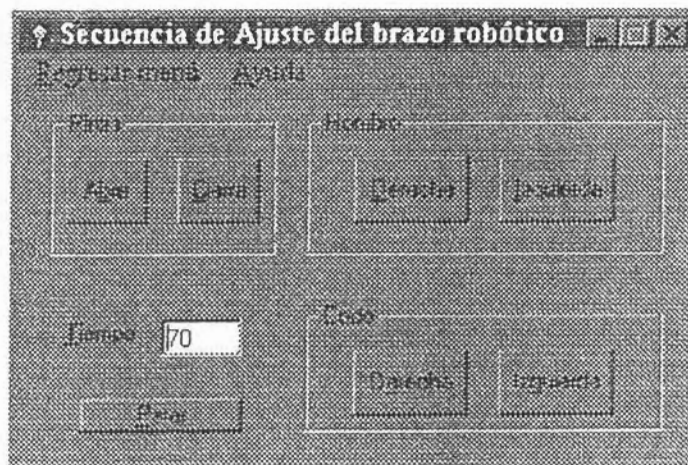


Figura 2.22 Ventana de Secuencia de Ajuste del robot

Es recomendable antes de conectar el robot al puerto paralelo, primero llamar este programa y presionar el botón parar, de esta manera se asegura que una vez que se conecte el robot, no hará cosas inesperadas.

El siguiente paso es conducir al robot a su estado inicial, para esto, basta con presionar el botón que corresponda al movimiento que se quiere realizar.

En la caja de texto tiempo, se le debe indicar al programa la duración del movimiento que se desea hacer, por omisión aparecen 70 milisegundos, es recomendable no excederse en tiempo a más de 500 milisegundos.

Se debe considerar que una vez que se ponga al robot en su posición inicial, el antebrazo ya no responderá al programa, debido a que su sensor de posición inicial está activado, esta consideración es muy importante, ya que se debe tener en cuenta, que para iniciar cualquier secuencia de movimiento a partir de la posición inicial, se debe mover primero la articulación del hombro del robot hacia la derecha, y una vez que el antebrazo deje de oprimir el sensor de tacto todas las posibilidades de movimiento del robot nuevamente estarán activadas.

El programa controlador del robot permite guardar quince diferentes secuencias de movimientos para el robot pedagógico, nueve de estas secuencias son utilizadas para que el robot conteste cada una de las jugadas de su contrincante en las nueve diferentes casillas del tablero, cinco secuencias pueden ser utilizadas para indicarle al robot en donde debe tomar cada una de las fichas de juego, con las que debe contestar, en el caso de que las fichas estén puestas en cinco diferentes lugares, y una más, que puede ser usada para guardar una secuencia de demostración. Cada una de estas secuencias de movimientos, están guardadas en un archivo diferente con extensión *.rob, en el mismo directorio del programa y tienen un nombre preestablecido.

Para grabar una nueva secuencia es necesario seleccionar la opción **programar secuencia** del submenú **brazo robot**.

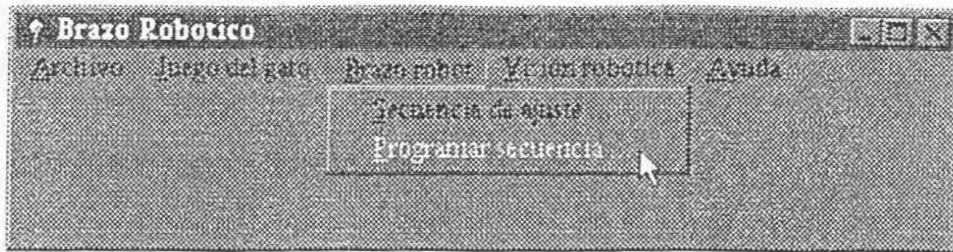


Figura 2.23 Menú principal

Una vez que se seleccione esta opción aparece la siguiente pantalla.

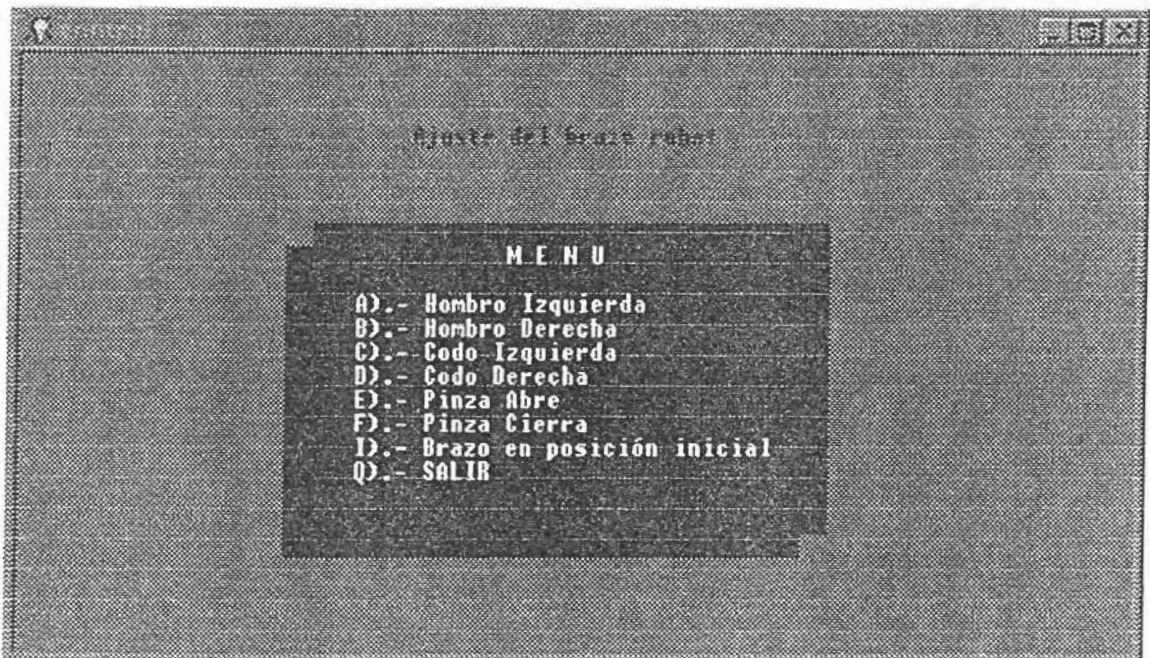


Figura 2.24 Ventana de ajuste del brazo robot

En este menú como se puede observar, también se pueden hacer algunos ajustes al brazo para llevarlo a su posición inicial, el hecho de que en este menú también permita hacer el mismo trabajo de la ventana *secuencia de ajuste del brazo*, que ya se ha revisado anteriormente, es con el fin que el usuario antes

de programar las secuencias de movimiento del robot, pueda también si es necesario hacer algunos ajustes rápidos en la posición inicial del robot.

Una vez que el robot está en su posición inicial, se selecciona la opción *I)*, misma que muestra la siguiente ventana.

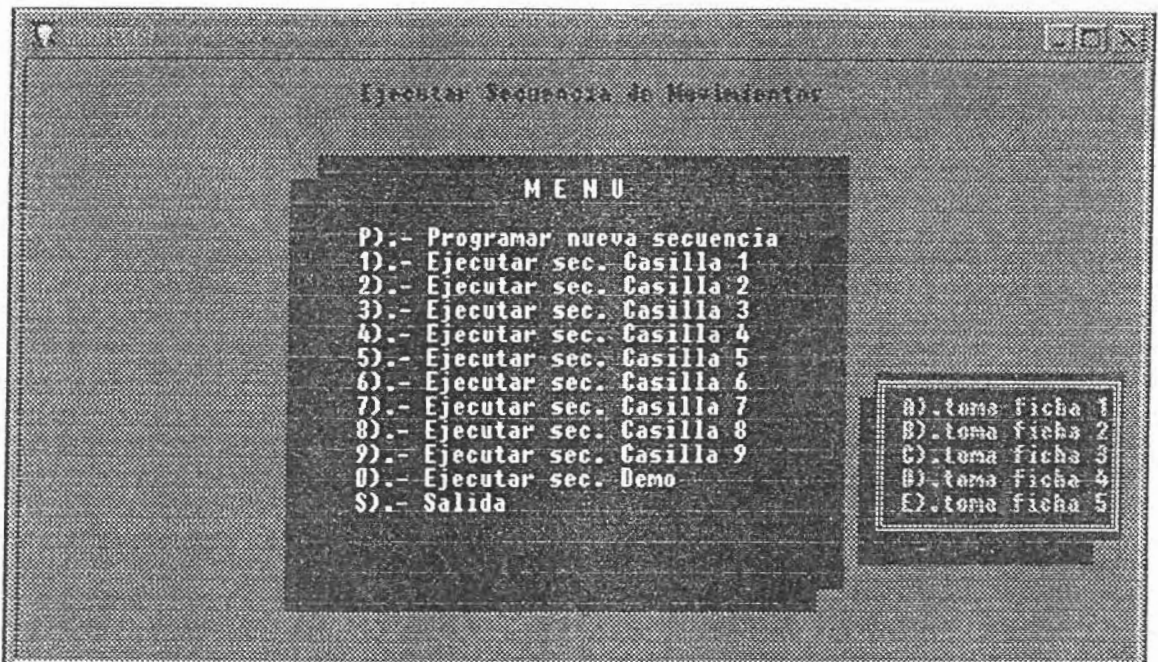


Figura 2.25 Ventana para ejecutar una secuencia de movimientos

Esta ventana permite ejecutar las quince secuencias diferentes para el robot pedagógico, la opción **Programar una secuencia nueva**, lleva a la siguiente ventana: **Programar secuencia de movimientos**, en la cual se puede cambiar alguna secuencia ya establecida o realizar una nueva.

Se puede guardar algún movimiento asociado a poner una ficha en alguna de las nuevas casillas del juego, o algún movimiento para tomar una de las cinco fichas de juego que utilizará el robot. Éstas pueden estar distribuidas sobre la

plataforma del robot, para ésto basta con seleccionar algunas de las opciones presentadas en el menú de la ventana siguiente.

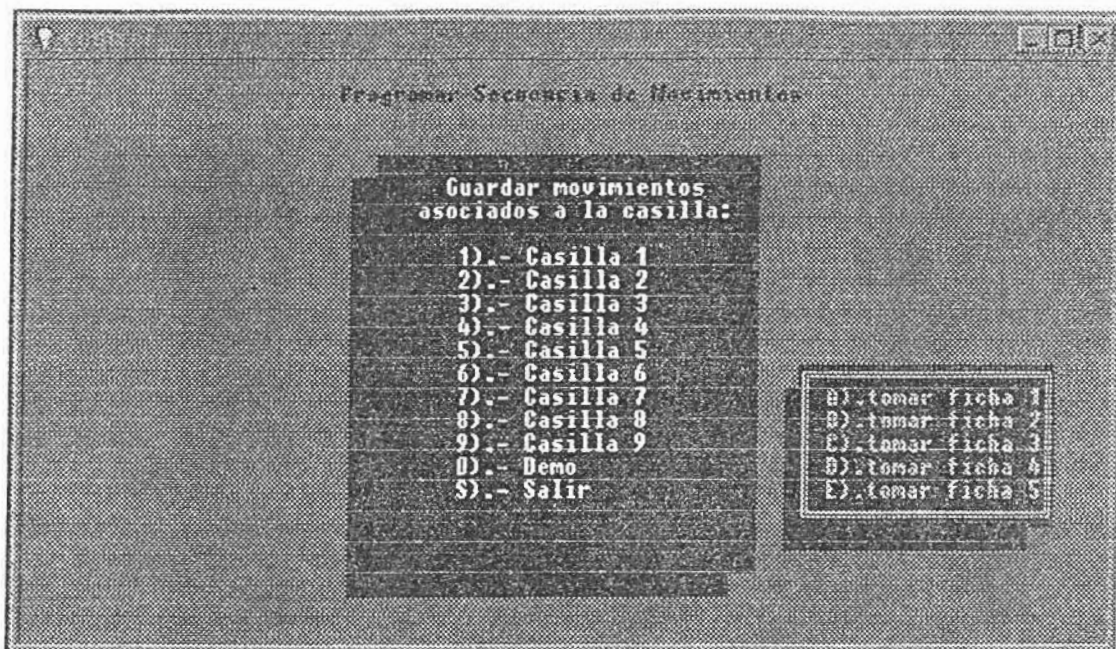


Figura 2.26 Ventana para programar una secuencia de movimientos

Una vez que se selecciona alguna opción aparece una ventana que avisa que la secuencia anterior asociada a esa tarea será borrada, y en su lugar será escrita una nueva.

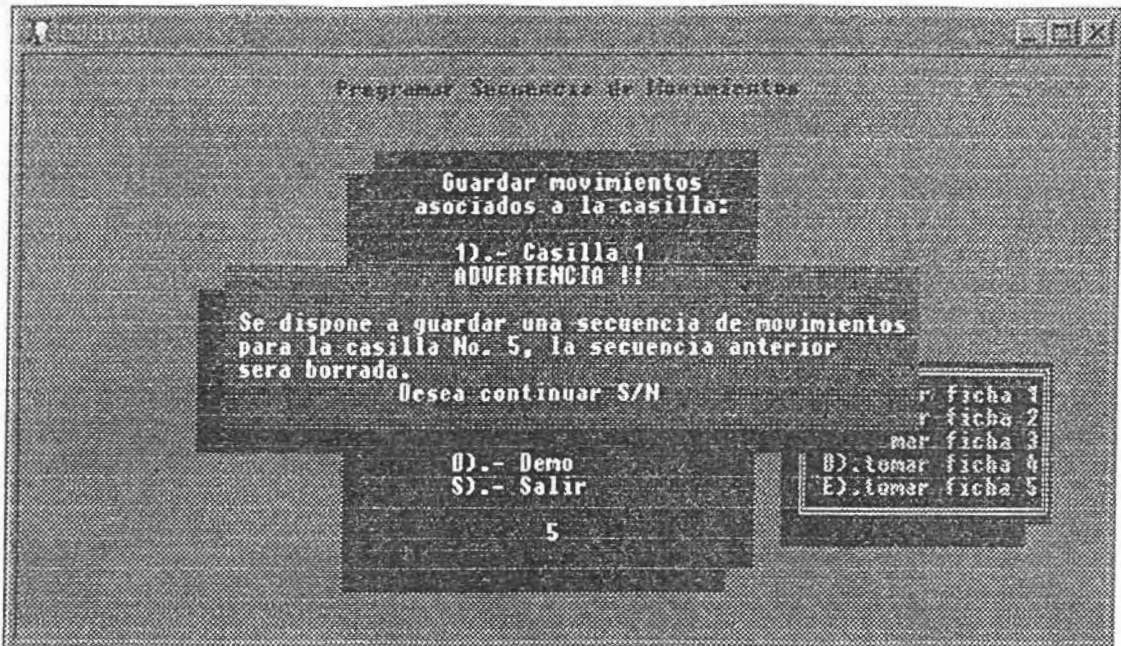


Figura 2.27 Ventana para programar una nueva secuencia de movimientos

Si realmente se quiere borrar la secuencia anterior, se contesta al programa con una "S", a continuación, se presenta ya la ventana para programar un nuevo movimiento.

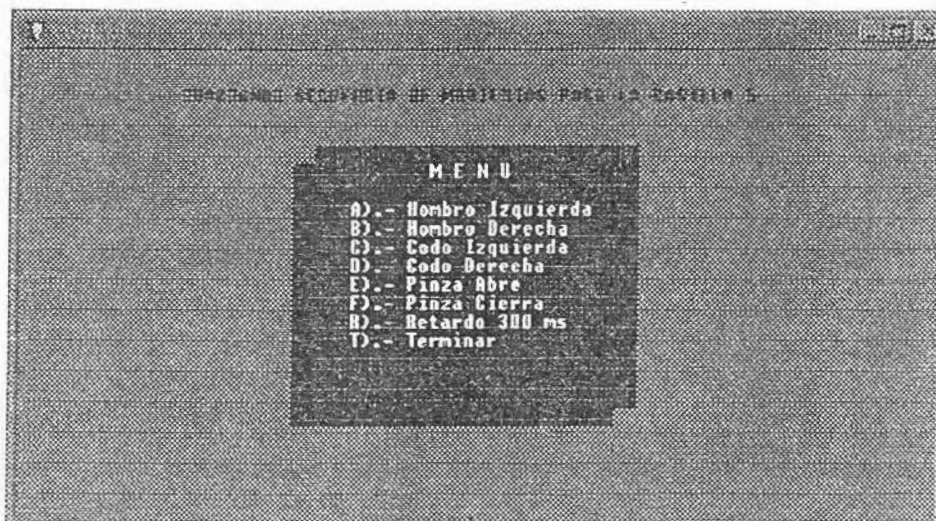


Figura 2.28 Ventana de comandos para programar el robot

Como se puede observar en la figura 2.28, esta ventana es similar a la de ajuste del brazo robot, con la diferencia de que en ésta cada movimiento que se manda a hacer al robot, será almacenado en memoria permanente (archivo), otra diferencia radica en que, en esta última ventana existe una opción retardo, esto obedece a que también se deben programar los retardos que se necesitan para que el robot se detenga por un momento para tomar o depositar una ficha, entre muchos usos más que le podamos dar en un momento determinado.

Para programar una secuencia se deben tener ciertas consideraciones, como por ejemplo, cuando se guían los movimientos del brazo por medio de este menú, se hacen a una cierta velocidad, generalmente menor a la que el robot reproducirá los movimientos, esto se debe tener muy en cuenta, ya que por lo general, una vez que se ha programado una secuencia y se le indica al robot que la reproduzca, nos damos cuenta que el movimiento tiene más alcance que el que se deseaba en un principio, esto es debido a que el robot al realizar todos los movimientos seguidos alcanza mayor velocidad, y por lo tanto la inercia del cuerpo es mayor y tarda más en detenerse. Así que para evitar este inconveniente se puede programar una secuencia en la cual entre paso y paso se introduzca un retardo, y al reproducirla se parecerá más a la secuencia que se desea, u otra manera es, una vez que se tenga la secuencia final ajustar el archivo con un editor ASCII, a prueba y error hasta tener la secuencia exacta de los comandos que deba ejecutar el robot, este último es el método que se utilizó para programar al robot, sin embargo por lo general el archivo que se obtiene con el programa es una buena aproximación del archivo final de movimientos al que se debe llegar.

Como ya se ha mencionado los archivos en donde se guardan los movimientos del brazo robótico tienen extensión *.rob*, se encuentran en el mismo directorio del programa *control_robot.exe*, y sus nombres son: *uno.rob*, *dos.rob*, hasta *nueve.rob*, para los movimientos necesarios para que el robot ponga una ficha en

cada una de las casillas, y *fich1.rob*, *fich2.rob*, hasta *fich5.rob*, que son los archivos que guardan los movimientos que le indican al brazo como tomar cada una de las 5 fichas con las que puede jugar, y un archivo más llamado *demo.rob*, en cual se puede guardar una secuencia de demostración.

Enseguida se mostrará el contenido de un archivo de movimientos del brazo, es fácil identificar los comandos que el robot entiende, cada comando es antecedido por un dígito, realmente este número es lo único que le importa al programa, el resto del comando es útil para el programador del robot, para conocer a que movimiento hace referencia.

```
2hde
2hde
7ret
3ciz
3ciz
3ciz
3ciz
7ret
7ret
6pci
1hiz
1hiz
1hiz
1hiz
7ret
7ret
3ciz
3ciz
7ret
5pab
7ret
1hiz
1hiz
1hiz
8ter
```

Es importante tener en cuenta cuando se edita algún archivo de movimientos como el anterior, no olvidarse de poner la última línea *8ter*, también no se deben dejar líneas en blanco entre comandos.

En el apéndice A de este documento se encuentra el código fuente del programa de control del robot pedagógico que se ha estado revisando.

Antes de finalizar este capítulo se revisará el soporte y tablero de juego del robot pedagógico.

2.6 Tablero de Juego y Soporte del Robot Pedagógico

La figura siguiente muestra una vista superior del tablero de juego y sus dimensiones, se debe recordar que la cámara de vídeo esta instalada en la parte de abajo del tablero y toma las imágenes de la partida a través de la base translúcida.

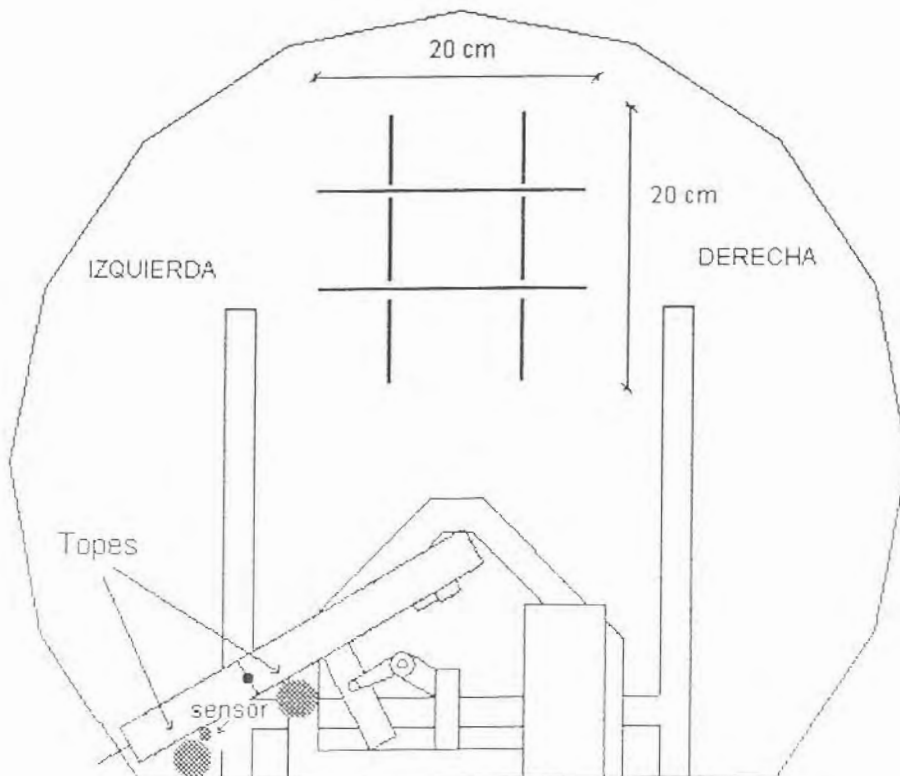


Figura 2.29 Vista superior del brazo y el tablero de juego

Por último en este capítulo se mostrará el esquema de la estructura de fierro que sostiene al brazo robot, el tablero y la cámara de vídeo, esta estructura está construida en fierro tubular cuadrado de 1 pulgada.

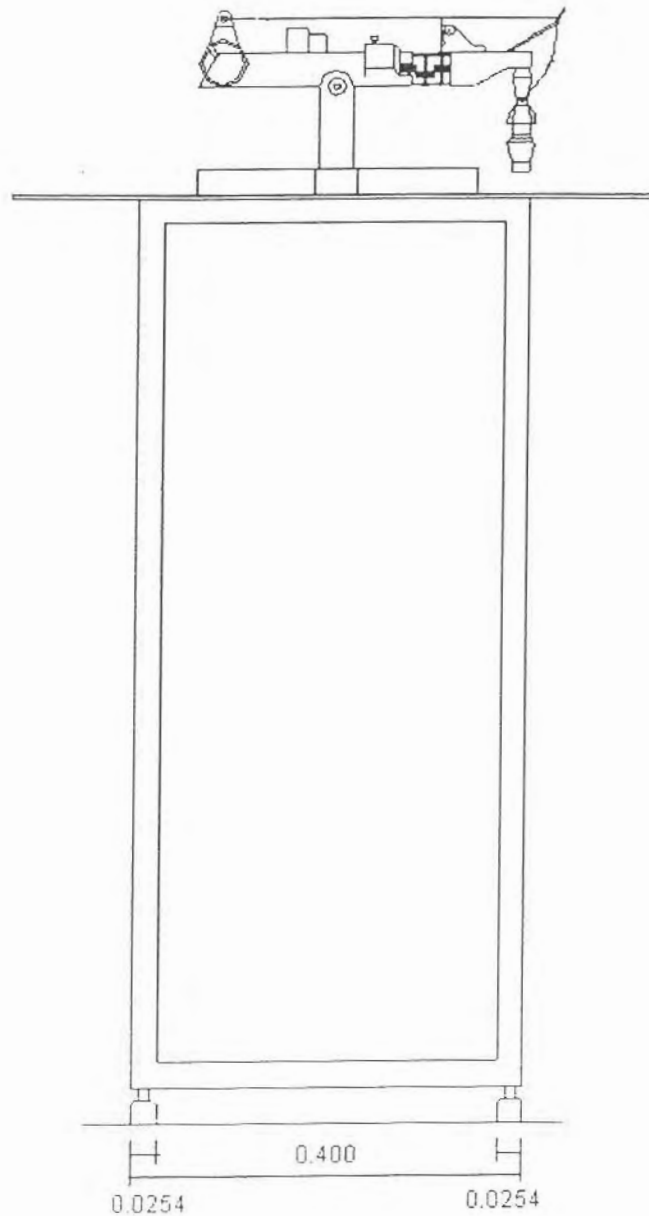


Figura 2.30 Vista frontal del soporte del brazo robot

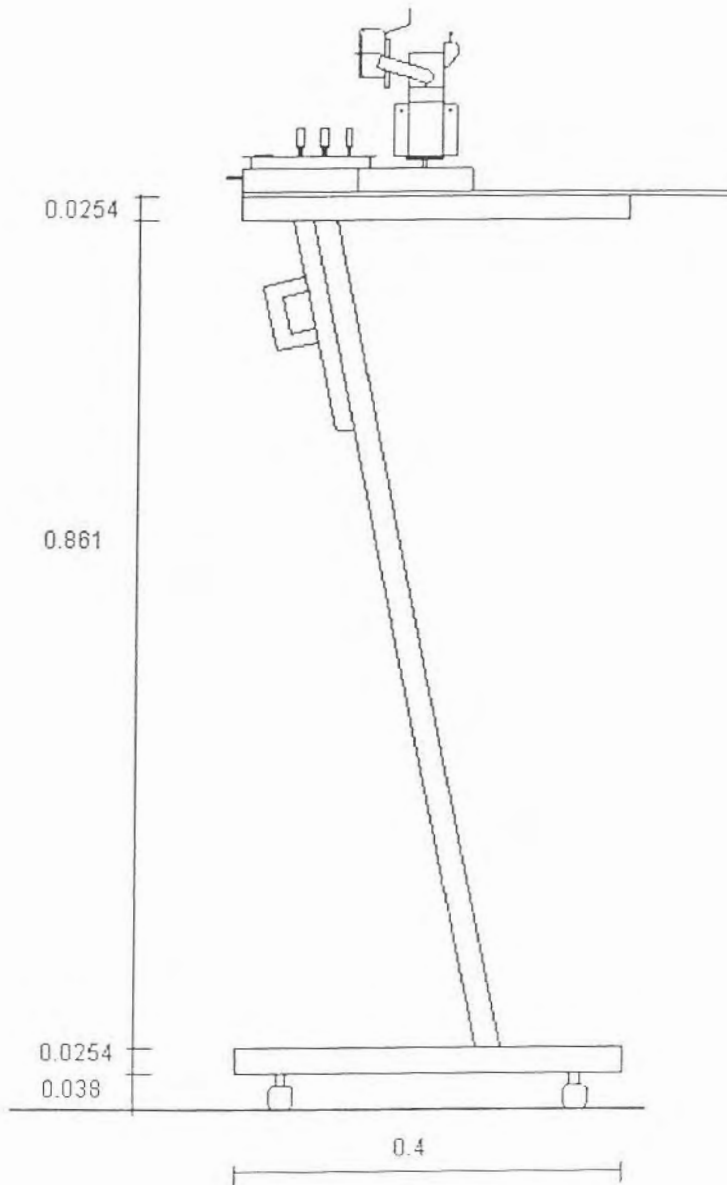


Figura 2.31 Vista lateral izquierda del soporte del brazo robot

SISTEMA BÁSICO DE VISIÓN
ARTIFICIAL

3.1 Introducción

En este capítulo se tratará la Construcción del Sistema Básico de Visión Artificial, que una vez acoplado al brazo robótico pedagógico le permita responder a su entorno de una manera inteligente y flexible, Se revisarán algunos conceptos básicos del procesamiento digital de imágenes que utilizaron para desarrollar el sistema de visión del brazo robótico. De la misma forma, se revisarán las características de la cámara que se utilizó para la adquisición de las imágenes, así como el proceso de calibración de ésta.

Se verá también, algunos de los procesos que están involucrados, dentro del concepto más amplio de visión computacional o visión artificial que se utilizaron en el sistema de visión por computadora, dentro de los cuales se encuentran: la captación o adquisición de las imágenes, la segmentación, el reconocimiento de patrones y la toma de decisiones.

Todo lo anterior con la finalidad de construir un sistema de visión que guíe al robot pedagógico propuesto, y le permita interactuar de manera inteligente con su entorno, que para nuestro caso específico, es un tablero del juego del gato y sus correspondientes fichas de juego.

3.2 Visión Artificial

Al igual que sucede en el ser humano, la capacidad de visión dota al robot de un sofisticado mecanismo de percepción, que le permite responder a su entorno de manera más flexible e inteligente, en comparación con otros mecanismos de detección.¹

La visión se considera el más importante y sofisticado sistema de percepción de un robot, evidentemente los sensores, los conceptos y los procesos asociados a la visión artificial, son considerablemente más complejos que los asociados a otras capacidades de percepción.

Es importante iniciar este apartado, con una definición de Visión Artificial.

*La Visión Artificial, es el proceso de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional.*²

La Visión Artificial, también llamada Visión Computacional, puede ser dividida en seis áreas principales:

Captación: *Es el proceso a través del cual se obtiene una imagen visual.*

Preprocesamiento: *Incluye técnicas tales como la reducción de ruido y realce de detalles.*

¹ [FU, 1990], pp. 306.

² [FU, 1990]

Segmentación: Es el proceso que divide a una imagen en objetos que sean de nuestro interés.

Descripción: Es el proceso mediante el cual se obtienen características convenientes para diferenciar un tipo de objeto de otro, por ejemplo tamaño y forma.

Reconocimiento: Es el proceso que identifica los objetos de una escena, por ejemplo diferentes tipos de piezas en un tablero de juego.

Interpretación: Es el proceso que asocia un significado a un conjunto de objetos reconocidos.

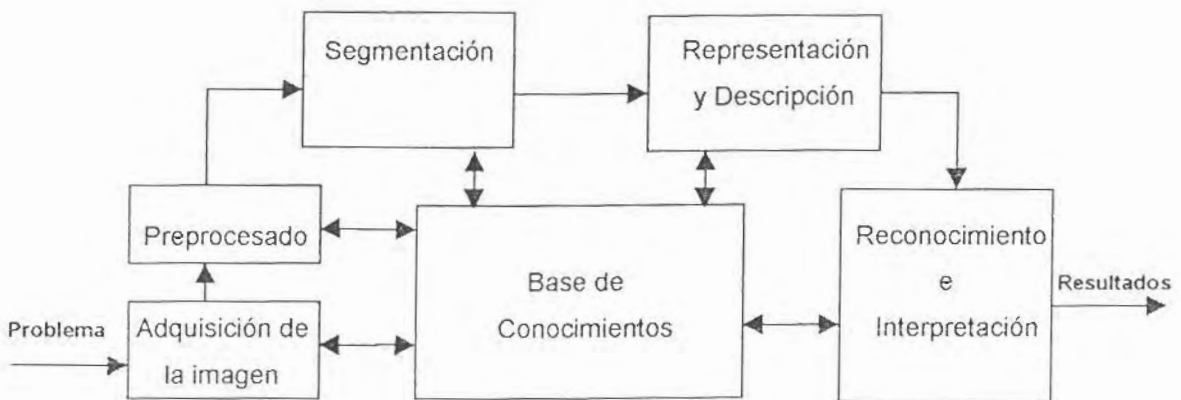


Figura 3.1 Etapas fundamentales de un Sistema de Visión Artificial

El objetivo de un Sistema de Visión Artificial, es crear un modelo del mundo real a partir de imágenes.

Un sistema de Visión Artificial, toma información útil a cerca de una escena a partir de su proyección en dos dimensiones.

Los Sistemas de Visión por Computadora, tienen diversas aplicaciones para solucionar problemas tales como:

- El reconocimiento de caracteres.
- Análisis de imágenes médicas.
- Automatización Industrial.
- Robótica.
- Cartografía.
- Medicina forense.
- Imágenes de radar.
- Percepción remota.

El Sistema Básico de Visión Artificial, que esta acoplado al brazo robótico pedagógico, esta dividido en varias etapas, dentro de las que destacan: **La adquisición de la Imagen, la segmentación, el reconocimiento de patrones, y la toma de decisiones**, cada una de dichas etapas se revisarán a continuación.

Antes de iniciar se debe aclarar que no fue necesario implementar, una etapa de **preprocesamiento**, previo al procedimiento de segmentación, debido a que las imágenes que se obtuvieron en la etapa anterior, no tienen demasiado ruido y el contraste es bueno, además de que ésto permite tener un ahorro considerable en tiempo de computo.

3.3 Adquisición de la imagen

Los principales dispositivos usados en la adquisición de las imágenes en la visión artificial, son las cámaras de televisión, o cámaras de vídeo, que están basadas en dispositivos de estado sólido o válvulas electrónicas.

Los primeros ofrecen grandes ventajas con respecto a las cámaras de tubos, como son: menor peso, menor tamaño, más larga vida y menor consumo de potencia, sin embargo en algunas ocasiones la resolución de este tipo de cámaras, no es tan bueno con en las cámaras de tubos.

En este trabajo específico, se utilizó para la captación de las imágenes una cámara digital de vídeo de la firma **Connectix**, modelo **Quickcam Vc**, para plataforma Windows PC. Conectada por puerto paralelo, como la mostrada en la figura 3.2.



Figura 3.2 Cámara digital de vídeo

La cámara de vídeo fue montada sobre un soporte ajustable, por debajo del tablero de juego, que como ya se ha mencionado es translucido, la cámara se ajustó en una posición perpendicular al tablero, de tal forma que permite tomar fotografías en un ángulo normal a la superficie del tablero.

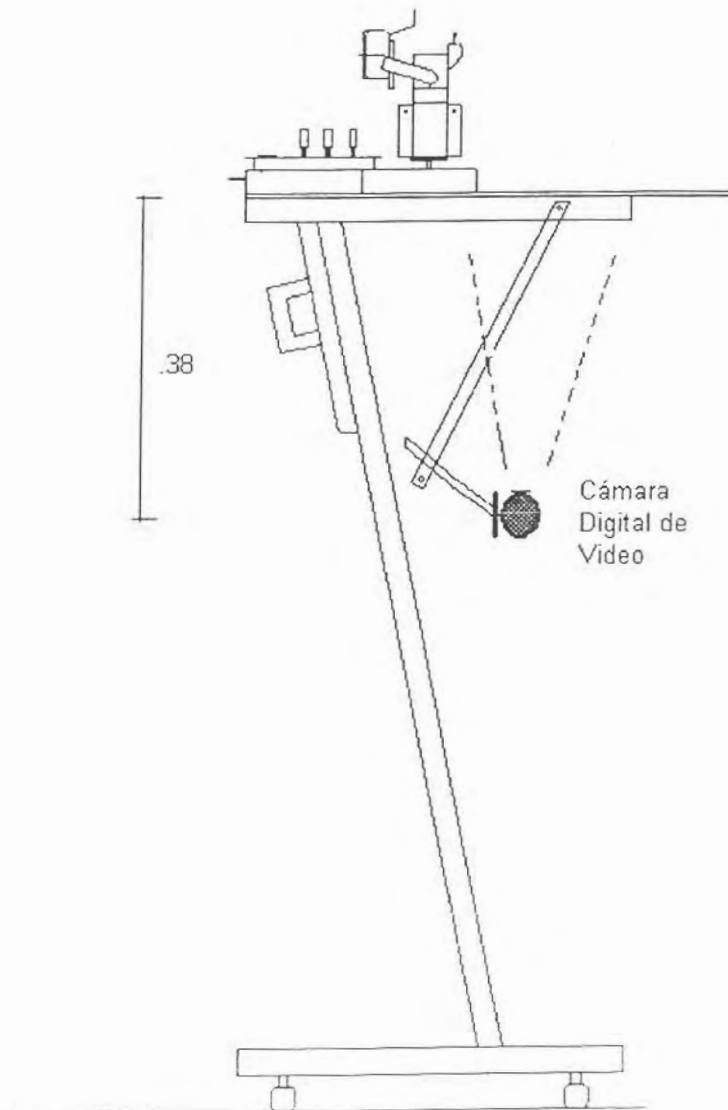


Figura 3.3 Vista lateral del soporte del brazo robot y cámara

La iluminación de la imagen juega un papel importante, ya que éste es un problema clásico de los algoritmos de visión.

En este caso se utilizó solo la iluminación del entorno y se obtuvieron resultados aceptables, sin embargo se tiene presente que con una buena fuente de iluminación que permita disminuir las sombras y aumentar el contraste, los resultados de los algoritmos de visión tendrán mayor probabilidad de éxito.

Una vez que la cámara estuvo montada en su base, el proceso siguiente fue su calibración, cabe destacar que la cámara que se utilizó conjuntamente con su software original, regresa imágenes en color, en formato BMP con diferentes resoluciones. Se configuró la cámara para que se obtuvieran imágenes con resolución de 160 x 120 píxeles. Se ajustó el soporte de la cámara y su lente, de tal manera que la imagen que se obtuviera fuera de la cuadrícula del tablero, y que las dimensiones de uno de sus lados correspondiera con el alto de la imagen. Entonces se obtuvieron imágenes como la siguiente.

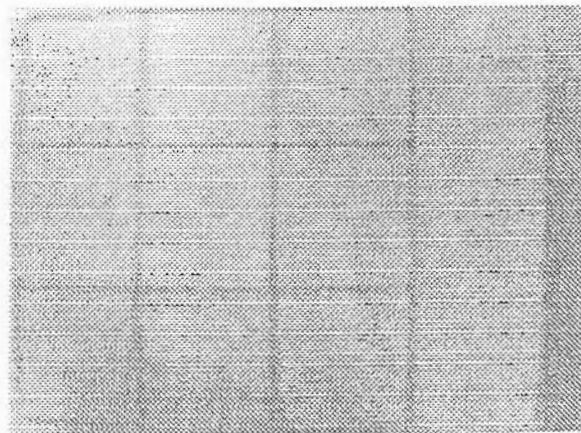


Figura 3.4 Imagen del tablero de juego, para efectos de calibración de la cámara

En la figura 3.4 aparte del tablero de juego, se puede observar en el lado derecho de la imagen parte de la plataforma y soporte del brazo robot.

La porción de la imagen de mayor interés es el lado izquierdo, es decir una porción de 120 x 120 pixeles, que en la realidad representa a un tablero de 20 x 20 cm, es decir, que cada centímetro del tablero esta representado por 6 pixeles aproximadamente, por consiguiente un pixel representa 0.0016 m.

El software que se utilizó para capturar las imágenes, es el mismo que viene incluido en la distribución de la cámara de vídeo.

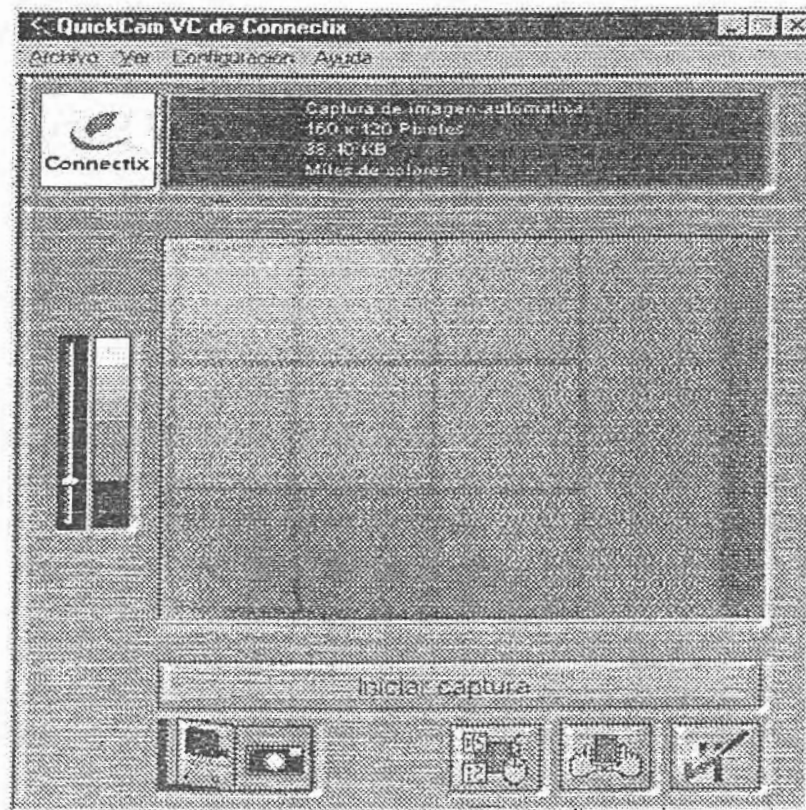


Figura 3.5 Interfaz de la cámara de vídeo

El software de la cámara de vídeo, en su menú configuración, entre otras opciones más permite configurar el programa para que tome una fotografía cada cierto período de tiempo y que la guarde en algún directorio seleccionado por el usuario.

Se configuró el programa de la cámara de vídeo en captura automática para que tome una fotografía, que es guardada en la ruta: *c:\brazo_robot\vision\captura.bmp*, y que ésta sea actualizada cada 2 segundos.

De esta manera cada 2 segundos se cuenta con una imagen actualizada de la situación del tablero de juego.

Como última observación en este apartado. Antes de iniciar el programa principal del robot, en donde están incluidas las rutinas de visión y control del brazo, es necesario cargar el software de la cámara e iniciar la captura automática.

3.4 Segmentación de la imagen

Una vez que se tiene la imagen capturada, es necesario segmentarla, en áreas y objetos que sean de interés, la imagen como ya se ha mencionado está almacenada en formato BMP, de tal forma que para hacer el procesamiento digital de la imagen, fue necesario conocer su estructura para acceder a la información.

El formato BMP, se detalla a continuación.

Cabecera de un Archivo BMP

Desplazamiento	Tamaño	Nombre	Descripción
0	2	BfType	ASCII "BM"
2	4	BfSize	Tamaño en longwords (4 bytes) del archivo
6	2	BfReserved1	Cero
8	2	BfReserved2	Cero
10	4	BfOffBits	Distancia en bytes, desde el final de la cabecera al inicio de la imagen

Tabla 3.1

Cabecera de la imagen

Desplazamiento	Tamaño	Nombre	Descripción
14	4	BiSize	Tamaño de esta cabecera, 40 bytes
18	4	BiWidth	Ancho de la imagen en pixeles
22	4	BiHeight	Alto de la imagen en pixeles
26	2	BiPlanes	Número de planos de la imagen, debe ser 1
28	2	BiBitCount	Bits por pixel, 1,4,8, o 24
30	4	BiCompression	Tipo de compresión
34	4	BiSizeImage	Tamaño en bytes de la imagen comprimida
38	4	BiXPelsPerMeter	Resolución horizontal en pixeles/metro
42	4	BiYPelsPerMeter	Resolución vertical en pixeles/metro

46	4	BiClrUsed	Número de colores usados (N)
50	4	BiClrImportant	Numero de colores importantes
54	4*N	BmiColors	Mapa de colores

Tabla 3.2

Seguido de la cabecera del archivo se encuentra, la cabecera de la imagen, y opcionalmente el mapa de colores.

Las imágenes que usan 1, 4, u 8 bits por pixel, contienen un mapa de colores, el tamaño del mapa de colores, es de 2, 16, o 256 entradas respectivamente. Si BiClrUsed, es diferente de cero, contiene el número de colores usados. El cual es también el número de entradas del mapa de colores.

Para imágenes de 24 bits por pixel, que son las que se utilizaron, no existe un mapa de colores, y la imagen esta directamente en color RGB, es decir cada grupo de 3 bytes representa 1 pixel.

Para el procesamiento digital de la imagen, se trabajó sobre una copia de la imagen guardada por la cámara, esto debido a que el procesamiento puede tardar algunos instantes, y causar problemas de concurrencia al acceder simultáneamente el mismo archivo dos aplicaciones diferentes.

De la copia de la imagen que realizan automáticamente los algoritmos de visión artificial, se utilizó solo una de las bandas de color, ya que al seleccionar una de las tres bandas se ahorró tiempo de cómputo necesario para convertir la imagen a tonos de grises, se eligió la banda verde después de realizar pruebas con cada una de las tres bandas, y toda vez que se notó ligeramente más información útil en el canal verde.

Todo el procesamiento digital de las imágenes a partir de aquí, solo se realizó con la banda verde de las imágenes originales.

En la etapa de segmentación, lo que se quiere encontrar son objetos en ciertas regiones de la imagen, debiendo recordar, que una de las acotaciones del problema es que, el robot siempre juega con las piezas en forma de cruz y transparentes, mientras que el usuario, juega con piezas circulares y oscuras.

Dado que el sistema del juego del gato recuerda en donde ha tirado el robot, entonces, lo que interesa es identificar las piezas puestas por el usuario, de forma que el trabajo se limitó a encontrar objetos oscuros, e identificar en que región del tablero se encuentran.

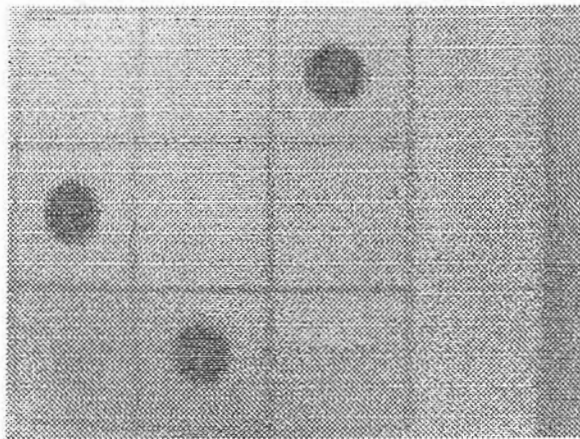


Figura 3.6 *Imagen del tablero de juego*

La figura 3.6 muestra el tablero de juego con la distribución de las piezas de la forma siguiente:

X O X
 O
 O



Recordemos que las imágenes son tomadas desde una perspectiva inferior y a través del tablero.

Para segmentar la información útil se utilizó una técnica de **umbral**, por medio de la cual la imagen anterior fue convertida en una imagen binaria.

Es decir se fijó un valor de umbral, y se comparó el valor de todos los píxeles de la imagen, aquellos que fueron menores o iguales al valor del umbral, se sobrescribieron con valor cero, que corresponde al color negro, y los píxeles que fueron mayores al valor del umbral se enviaron a blanco, por su valor 255, de esta manera se discriminó toda la información que no fue útil, ya que las piezas circulares, tienen valores muy cercanos a cero.

Se determinó un valor de umbral tal que permitiera minimizar las pérdidas de información valiosa calculando el **histograma** de la imagen, considerando únicamente el canal verde. El histograma de una imagen está definido por:

El **histograma** de una imagen, indica la frecuencia de ocurrencia de un valor de píxel dentro de la imagen, una definición más formal es:

Histograma: *Es una función que muestra para cada valor de píxel D , el número de píxeles $H(D)$ en la imagen que tiene ese valor. $H(D)$ también se le llama la frecuencia. Al graficar, D es el eje horizontal (abscisas), y $H(D)$ es el eje vertical (ordenadas).*

El histograma también se puede interpretar como la función de densidad de probabilidad. Por ejemplo $H(3)/16$, es la probabilidad de que un pixel que pertenece a una imagen de 4×4 elementos, tenga valor de 3.

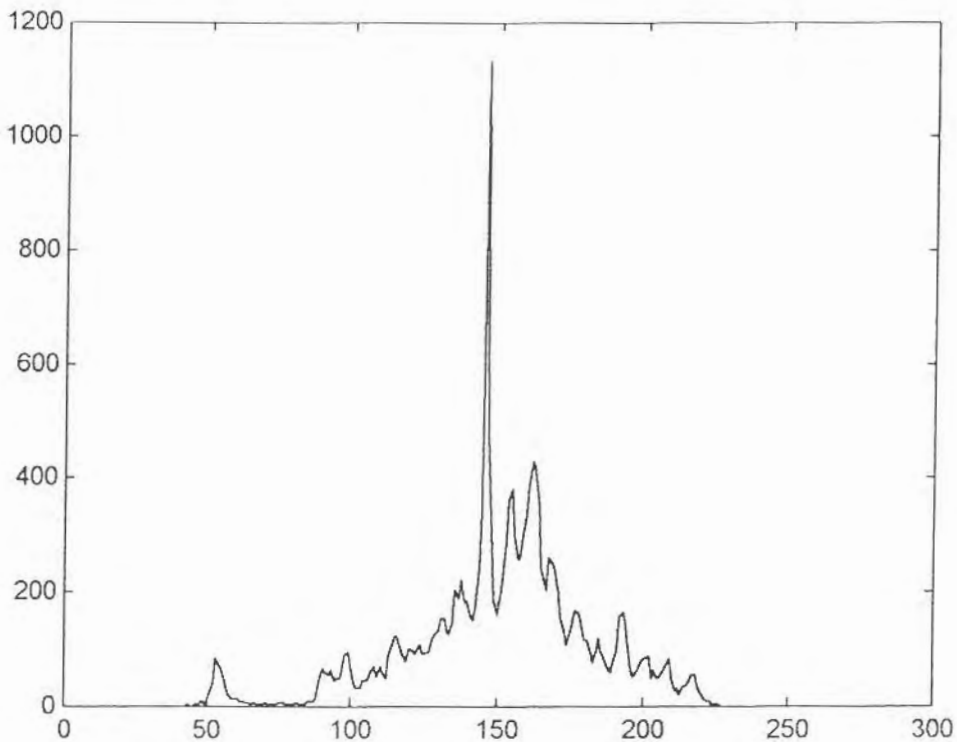


Figura 3.7 Histograma de la escena de juego, usando la banda verde

Del histograma de la figura 3.7, se obtiene información relevante sobre la fotografía, la interpretación es la siguiente: el pico mayor, que se encuentra centrado aproximadamente en 145 representa en la imagen a la mayor parte del tablero, a partir del pico mayor, los picos que se encuentra hacia la derecha, representan las partes más claras del tablero, los valores de los píxeles que se encuentran entre 105 y 145, representan, las partes de la imagen más oscuras como son, las líneas del tablero, y algunas pequeñas porciones del soporte

de la plataforma, así como las fichas en forma de cruz, y algunas sombras, se distingue un pequeño pico en 100, lo que se interpreta como la mayor parte del soporte del robot que se alcanza a apreciar en la fotografía, y la información que se encuentra en el pico centrado en 55, corresponde a las fichas del juego circulares y negras, así como la parte del soporte del tablero del robot, mas cercana a la base del brazo.

Gracias al histograma, se pudo determinar el umbral que se aplicó para segmentar la información de interés. Se eligió un umbral de **60**, ya que la información relevante está entre 0 y 60, y cuyo pico, se encuentra centrado en 55, entonces aplicando dicho umbral, se obtuvo:

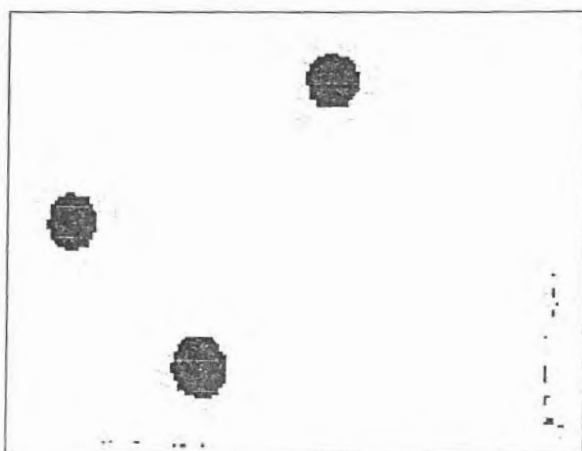


Figura 3.8 *Imagen segmentada mediante el procedimiento de umbral*

Hasta este punto, se observa una imagen con la información de interés, pero recordemos que en la fotografía, también aparece, parte de la base del robot y el soporte, así que se debió discriminar toda esta información para la siguiente etapa de los algoritmos de visión artificial.

Se pudo observar la presencia de ruido en la parte inferior y en el extremo derecho de la imagen, ocasionado por la sombra que proyecta el brazo mecánico sobre el tablero y una parte del soporte del robot respectivamente, sin embargo fue despreciado, ya que no afecta en proporciones considerables el resultado del sistema de visión, debido a que la región derecha de la imagen en donde aparece parte de este ruido no es procesada por los algoritmos de visión, y el ruido de la parte inferior, solo aumenta muy poco el área oscura de dos de las nueve regiones de la imagen que se procesan por separado, sin llegar a afectar el resultado de las etapas posteriores de reconocimiento de patrones y toma de decisiones como se verá en la siguiente sección.

Se necesitaba conocer la ubicación de los objetos en el tablero, así que se delimitó la imagen a fin de segmentarla en diez regiones, nueve de ellas corresponden a las nueve casillas del tablero, los valores de estos límites y la imagen obtenida hasta este momento, es la entrada al algoritmo de reconocimiento de patrones.

Los cinco valores que segmentan a la imagen como se puede ver en la figura 3.9, así como el valor del umbral son los parámetros que se envían a la siguiente etapa del sistema de visión artificial, estos parámetros pueden ajustarse antes de iniciar una partida del juego del gato, dependiendo de las condiciones del ambiente en las que se vaya a operar el robot.

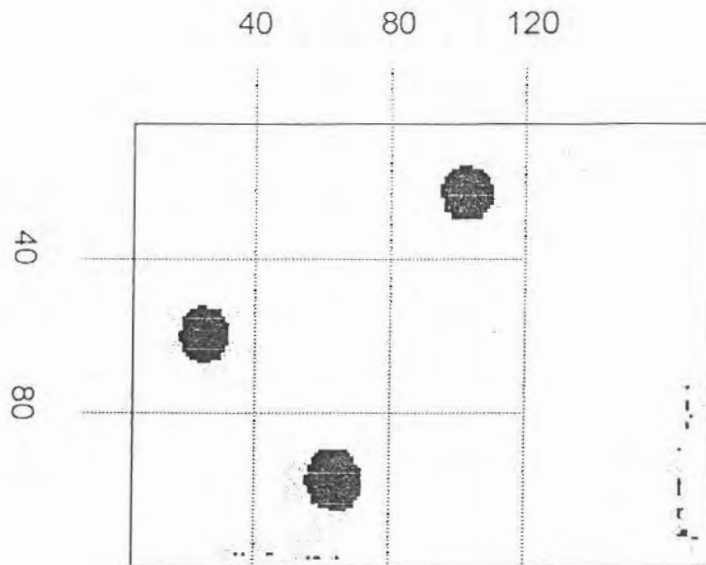


Figura 3.9 Imagen con regiones acotadas y segmentadas

3.5 Reconocimiento de Patrones y Toma de Decisiones

La función de los algoritmos de reconocimiento, es identificar cada objeto segmentado de una escena y asignarle una etiqueta.

En la mayoría de los sistemas de visión, en la etapa de reconocimiento, se trabaja haciendo ciertas suposiciones, una de estas es: se supone que los objetos han sido segmentados como unidades individuales.

Otra limitación común es que las imágenes deben ser tomadas con una geometría de visión conocida, generalmente la cámara esta dispuesta perpendicular al espacio de trabajo, esto facilita mucho el trabajo de la

segmentación y del reconocimiento, ya que se disminuyen la diversidad de características de forma.

Los métodos que se usan, para reconocer patrones, se pueden dividir en dos categorías importantes: métodos de **decisión teórica**, que se basan en descripciones cuantitativas, como es, estadísticos de textura, y los métodos **estructurales** que utilizan descripciones simbólicas y sus relaciones, esto es secuencias de direcciones de fronteras codificadas en cadenas.³

Estos tipos de procedimientos por lo general, se utilizan para reconocer representaciones bidimensionales de los objetos.

El algoritmo que realiza proceso de etiquetado en el sistema de visión del robot pedagógico, se puede considerar dentro de los métodos de decisión teórica. Ya que se basa principalmente en el cálculo del área de los objetos de la escena y de su posición dentro de las regiones segmentadas.

El funcionamiento del algoritmo es sencillo, consiste en calcular el área de las regiones oscuras para cada uno de los nueve segmentos de la imagen, si existe un objeto oscuro y cuya área es mayor o igual a un parámetro predeterminado, el algoritmo decide etiquetar esa sección con presencia de pieza de juego negra o ausencia de ella.

Para determinar el parámetro que determina si el objeto es o no una pieza negra, se hicieron algunas consideraciones.

³ [FU, 1990], pp 440

Las piezas negras circulares, tienen un diámetro de 2.9 cm, por lo tanto su área es igual a:

$$\text{Área de la pieza: } (3.1416) \times (1.45^2) = 6.605 \text{ cm}^2$$

Como ya se ha mencionado, en la imagen cada centímetro lineal, está representado por 6 píxeles, por tanto, un centímetro cuadrado está representado en la imagen con 36 píxeles, siguiendo esa lógica una pieza negra, utiliza:

$$(36) \times (6.605) = \mathbf{237} \text{ píxeles aproximadamente.}$$

En este orden de ideas, un criterio para decidir si en una casilla existe o no una pieza negra, es comparar el área de las regiones oscuras con el valor que se calculó anteriormente, sin embargo, esto no se puede hacer directamente, ya que se debe considerar que al segmentar la imagen se pierde alguna información y aparece un efecto de diente de sierra en el contorno de las piezas, por tanto el área de las piezas que aparecen en la imagen una vez que se aplica el umbral disminuye.

Para determinar entonces un rango apropiado de valores de área, que permita diferenciar una pieza de juego de un objeto oscuro, que puede ser ruido o alguna sombra, se hicieron varias pruebas, y en cada uno de ellas se calculó el área de las piezas una vez segmentadas y se obtuvo un valor promedio, con base en estos cálculos se definió un rango en el cual deben de caer el valor del área de las piezas circulares, el rango es.

$$237 > \text{Área} > 110$$

Para cada una de las nueve regiones segmentadas de la imagen se calculó el área de los objetos oscuros y se aplicó el criterio anterior para decidir si era o no una pieza negra circular.

El algoritmo de reconocimiento de patrones da como resultado una matriz de 3x3, en donde las casillas etiquetadas con valor 1, corresponden a casillas con una pieza oscura, y las casillas con valor 0, corresponden, a casillas en donde no se pudo determinar si están vacías o tienen una pieza en forma de cruz, pero esto último no es ningún inconveniente ya que tenemos una matriz más, en donde se almacenan todos los movimientos del robot, y del usuario.

Hasta aquí, se cuenta ya con todos los subsistemas que forman el brazo robótico pedagógico, dotado de un sistema básico de visión artificial programado para jugar al gato, más adelante en el apéndice A de este trabajo, se describirá la manera de operar el robot, y el funcionamiento del programa principal, el cual hace uso de todos estos subsistemas.

CONCLUSIONES

Un número importante de investigadores en todo el mundo ven en la robótica pedagógica un gran campo de investigación. Esta nueva disciplina requiere de una buena colaboración de especialistas de diferentes áreas, para poder construir robots que puedan apoyar el proceso de enseñanza-aprendizaje de manera eficiente.

Al inicio de este trabajo algunos de los objetivos que se plantearon fueron: construir una herramienta pedagógica robusta, que pudiera ser útil para apoyar y reforzar el proceso de enseñanza, desarrollar una aplicación en la que se pusieran de manifiesto algunas técnicas básicas de visión robótica, además de experimentar en la construcción de un brazo robótico.

Con la construcción del **Brazo Robótico Pedagógico dotado de un Sistema Básico de Visión Artificial**, programado para jugar el Juego del Gato, se considera que los objetivos planteados al inicio de este trabajo de Tesis se alcanzaron satisfactoriamente.

La herramienta pedagógica resultante es útil para auxiliar en el proceso de enseñanza de conceptos relacionados con la computación, la electrónica y la robótica, conceptos tales como: control de dispositivos periféricos, interfaces con la computadora a través del puerto paralelo, algoritmos básicos de visión robótica, etc.

El robot como herramienta pedagógica también tiene algunas limitantes, el hecho de solo interactuar con el brazo robot, no favorece el aprendizaje del educando en la misma medida que si se involucrara en el diseño y construcción del artefacto tecnológico.

Sin embargo se considera que es posible involucrarse en el diseño y construcción de un robot similar, tomando como modelo el robot que se propuso en este trabajo, siguiendo los lineamientos aquí tratados en cuanto al uso de componentes de bajo costo y fácil adquisición.

Si bien es cierto que el robot de inicio está programado para jugar al gato con un contrincante humano, también es cierto que puede ser adaptado fácilmente para desempeñar otro tipo de tarea, que involucre el mover objetos a través de un plano, esto debido a la arquitectura del brazo y los grados de libertad que actualmente tiene el actuador final.

Con la arquitectura del brazo que se construyó, el actuador final tiene la posibilidad de desplazarse con cinco grados de libertad, pero para la solución del problema inicial, que fue el construir un robot que jugara al gato, únicamente se requirió que el actuador se desplazara con dos grados de libertad, esto implicó adaptar tres motores que proporcionan la fuerza motriz al actuador y a dos articulaciones más del robot, con la posibilidad futura de adaptar motores a las tres articulaciones restantes.

Con respecto a los algoritmos de Visión Artificial con los que fue dotado el sistema, tienen ciertas limitantes, debido a que no se utilizó una fuente auxiliar de iluminación, solo se trabajó con base en la iluminación del entorno, esto limitó el tiempo de respuesta del robot para realizar sus tiros una vez que el usuario le indica que puede hacerlo, esto es debido a que se necesita un tiempo de ajuste automático de la cámara una vez que la escena ha cambiado, si se

procesaran las imágenes adquiridas un instante después de que el usuario ha realizado su movimiento, las posibilidades de éxito del sistema de visión bajan hasta un 50%, cifra que se obtuvo después de realizar repetidas pruebas al sistema, para evitar esto, se procesan las imágenes obtenidas de 2 a 4 Segundos después de que se ha realizado un cambio en la escena, lo que nos da un porcentaje de éxito de hasta el 90%.

El control del robot, es atendido por una computadora personal a través del puerto paralelo, se tuvieron algunos problemas al tratar de reproducir de manera exacta los movimientos del brazo, esto fue debido a que se utilizaron motores de corriente continua, y por tanto el control de un movimiento solamente dependía del tiempo que estuviera encendido el motor, esto implicó que los movimientos del brazo mecánico tuvieran un grado de desajuste estimado en un 2%.

El alcance correcto de un movimiento dependía de variables tales como: la velocidad del procesador de la computadora, el voltaje y la corriente de alimentación de los motores, así como el desajuste natural de los engranes y la lubricación de los mismos.

El problema del desajuste de los movimientos se contrarrestó programando secuencias en donde los movimientos se dividieron en pequeños pasos, es decir se definió una unidad de movimiento para cada uno de los motores en sus dos sentidos de rotación, para el caso del motor que mueve la articulación del codo, un paso corresponde, a que el motor permanezca encendido 70 milisegundos seguido de un retardo de hasta 300 milisegundos, de la misma manera se definió un paso para el motor de la articulación del hombro en 100 milisegundos, y el del motor del actuador en 500 milisegundos.

El funcionamiento del Robot que juega al gato, no es completamente automático, ya que se le debe indicar el momento en el cual debe de realizar su siguiente tirada, esto se hace por medio de un click en el botón establecido para este fin, así también, las piezas con las cuales juega el robot deben de encontrarse a su alcance cada vez que requiera utilizarlas, es decir requiere de un sistema manual para alimentarse de las fichas de juego. De aquí surgen algunas propuestas que pueden ser consideradas como trabajos futuros en este Sistema Robótico:

- Automatizar el sistema de alimentación de las fichas de juego.
- Proporcionarle un mecanismo basado en los algoritmos del sistema de visión artificial, para permitirle al robot determinar el momento en el cual puede realizar automáticamente sus movimientos dentro de la dinámica del juego.
- Proporcionarle un sistema de control de la posición del actuador final en un momento determinado, esto puede realizarse sustituyendo los motores actuales por motores de pasos con sus respectivos ajustes en el sistema.
- Proporcionar mayor movilidad al actuador final, adaptándole motores a las tres articulaciones del brazo que carecen de ellos.

El Brazo Robótico dotado de un Sistema Básico de Visión Artificial, está listo para ser utilizado como un artefacto tecnológico que apoye la adquisición de conocimiento en muy diversas ramas de la ciencia, solo hace falta aplicarlo en la enseñanza con imaginación, ya que su potencial para este fin es muy grande, y ha quedado demostrado en la experiencia que ahora poseemos en el fascinante mundo de la **Robótica**.

BIBLIOGRAFÍA

- [ALFARO, 1997] Alfaro Pérez Hector; *Diseño de un robot controlado por computadora, orientado a la enseñanza pedagógica*, Tesis de Licenciatura, UTM, Febrero, (1997).
- [CASTLEMAN, 1996] Castleman Kenneth R., *Digital Image Processing*, Prentice Hall (1996).
- [COUGHLIN, 1997] Coughlin Robert F., Driscoll Frederick F., *Amplificadores Operacionales y Circuitos Integrados Lineales*, Prentice Hall (1997).
- [FU, 1990] Fu K. S., Gonzalez C. R., Lee G. S. C.; *Robotica Control, Detección, Visión e Inteligencia*, Mac Graw Hill, México (1990).
- [GONZALEZ, 1993] Gonzalez C. Rafael, Woods E. Richard; *Digital Image Processing*, Addison Wesley (1993).
- [GROOVER, 1990] Groover Mikell P., *Robótica Industrial*, Mac Graw Hill, México (1990).
- [IOVINE, 1998] Iovine John; *Robots, Androids and Animatrons*, Mac Graw Hill, España (1998).
- [JAIN, 1995] Jain Ramesh, Kasturi R., Schunck B.; *Machine Vision*, Mac Graw Hill (1995).
- [KAY, 1993] Kay David C., Levire John R. *Graphics File Formats*, Windcrest, Mac Graw Hill (1993).
- [RICH, 1994] Rich Elaine , Knight Kevin; *Inteligencia Artificial* , Mac Graw Hill, España (1994).
- [RUIZ-VELAZCO, 1991] Ruíz Velasco, E., *Robotica Pedagógica*, Memoria de la 3a Conferencia Internacional, México (1991)
- [RUSSELL, 1996] Russell Stuart, Norvig Peter, *Inteligencia Artificial Un Enfoque Moderno*, Prentice Hall (1996)
- [SANDLER, 1991] Sandler Ben-Zion, *Robotics*, Prentice Hall (1991)

- [VIVET, 1989] Vivet Martial, Nonnon Pierre, *Actes du Premier Congrès Francophone de Robotique Pédagogique*, Le Mans, Francia (1989)
- [WINSTON, 1994] Winston Henry Patrick; *Inteligencia Artificial*, Addison Wesley Iberoamericana (1994).

REFERENCIAS ELECTRÓNICAS

- { AIR, 1998} **Artificial Intelligence Resources Center**
<http://www.aircenter.net/index1.html>
- {CIA, 1999} **Centro de Inteligencia Artificial**
<http://www-cia.mty.itesm.mx/ligas.html>
- {FTIVC, 1999} **Fundamentos del Tratamiento de Imágenes. Visión por Computador**
<http://bucanero.ugr.es/wwwcurso/ccordoba/ccordoba.html>
- {Q-ROBOTICA, 1999} **¿Qué es la Robótica?**
<http://www.eupmt.es/cra/robotica.htm>
- {M-TECNOLOGIA, 1999} **Marcadores de Tecnología**
<http://www.geocities.com/Athens/Olympus/4788/tecnologia/mark.htm>
- {R-ARGENTINA, 1999} **Robotica Argentina**
<http://www.geocities.com/CapeCanaveral/Lab/2767/>
- {R- PEDAGOGICA, 1999} **Robótica Pedagógica**
<http://www.fcencias.unam.mx/revista/soluciones/SA40/rob-ped.html>
- {U-PARALELO, 1999} **Uso del Puerto Paralelo del Pc.**
<http://ing.umayor.cl/misc2/ppar/index.htm>
- {VAVH, 1999} **Visión Artificial y Visión Humana, Aplicación a la Percepción Visual en Robótica.**
<http://sirio.psi.ucm.es/PROYECTOS/VISIONROBOT/vavh.html>
- {V-COMPUTACIONAL, 1999} **Vision Computacional**
<http://w3.mor.itesm.mx/~esucar/Vision/indice.html>
- {WGFFS, 1999} **Where to Get File Format Specifications**
<http://www.geocities.com/~vmushinskiy/ffformats/files/whtogtqf.htm>

**MANUAL DE OPERACIÓN DEL
BRAZO ROBOT**

Componentes del Brazo Robot

Las partes que componen al robot que juega al gato son las siguientes.

- Brazo mecánico
- Soporte y tablero de Juego
- Fuente bipolar de ± 3 volts
- Interfaz del puerto paralelo y los motores
- Cámara digital de vídeo, QuickCam VC
- Computadora personal

Requerimientos mínimos de operación

- *Procesador Pentium o superior*
- *Windows 95 o versiones más recientes*
- *32 MB en RAM*
- *Disco duro 1 GB.*
- *Teclado PS/2*
- *2 Puertos paralelos*
- *Tarjeta de sonido y bocinas*
- Software controlador del robot

Descripción

El brazo robot se encuentra montado a un costado del tablero de juego sobre una plataforma de acrílico transparente, parte de la plataforma corresponde al tablero de juego, este segmento está cubierto por papel encerado sobre el que se encuentra rotulado el tablero del gato.

La plataforma de acrílico está sujeta al soporte tubular el cual mantiene fijos a casi todos los elementos del robot que son: el brazo mecánico, el circuito de interfaz de los motores del robot, la fuente bipolar, el tablero de juego y la cámara de vídeo, el único elemento que no está contenido por el soporte es la computadora personal.

El soporte tubular tiene en su base cuatro tornillos niveladores que permiten llevar al robot y a la plataforma a una posición completamente horizontal, para su correcto funcionamiento.

Instalación del robot

Podemos distinguir dos partes fundamentales que componen al brazo robot dotado de un sistema básico de visión artificial programado para jugar al gato.

- 1.- *El brazo mecánico, con su soporte tubular que contiene todos los componentes mecánicos, algunos circuitos electrónicos, y sensores*
- 2.- *El centro de control del robot, que en este caso es una computadora personal, con las características antes mencionadas.*

La instalación es sencilla, el soporte del brazo tiene 2 cables para puerto paralelo, un cable de alimentación que se debe conectar a un toma corriente convencional de 120 Volts. y un cable especial de alimentación de la cámara.

Cada uno de los cables para los puertos paralelos tienen diferente color, el cable negro, que es el de la cámara de vídeo, se debe conectar al puerto LPT2 de la PC, y el cable blanco que es el que permitirá la comunicación con el brazo mecánico en el puerto LPT1.

El cable de alimentación de la cámara, tiene dos conectores, el conector macho se conectará en el puerto del teclado, y el teclado se debe conectar con la entrada hembra del adaptador del cable de alimentación de la cámara.

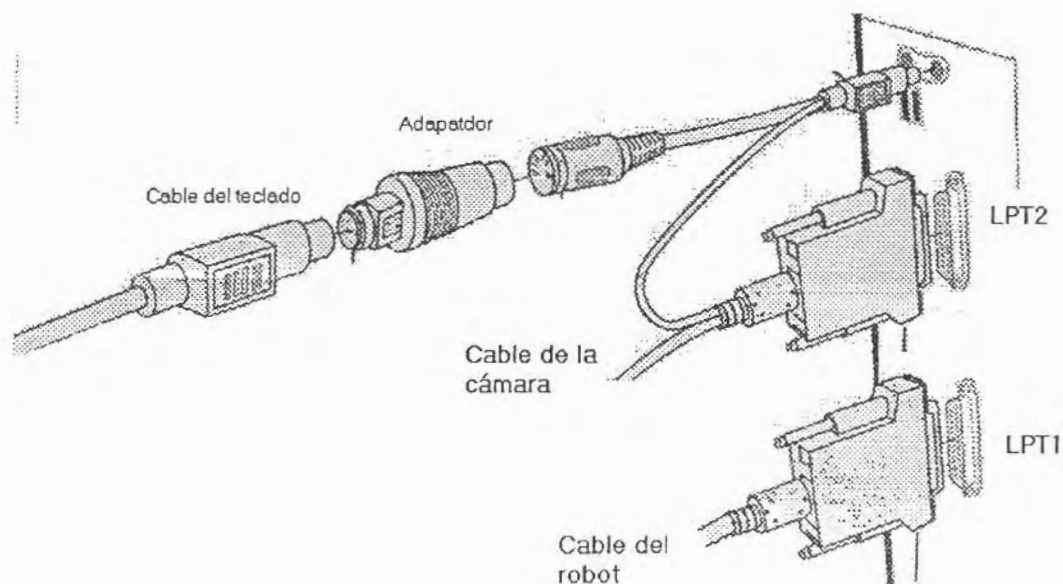


Figura 4.1 Esquema de conexiones del robot y la Pc.

La computadora personal además de cumplir con las características mencionadas al inicio de este Apéndice, debe instalársele el software de la cámara de vídeo, la instalación de dicho software se detalla en los manuales y el CD que acompañan a la cámara, además de los programas propios del control del brazo robot.

Enseguida se listan los archivos y programas necesarios para el funcionamiento del robot así como la ubicación de estos.

C:\brazo_robot\programa\control_robot.exe
C:\brazo_robot\programa\puerto.exe
C:\brazo_robot\programa\control.exe
C:\brazo_robot\gato\gat_rob.exe
C:\brazo_robot\programa\contro.bat
C:\brazo_robot\programa\ejecuta1.bat hasta ejecuta9.bat
C:\brazo_robot\programa\demo.bat
C:\brazo_robot\programa\uno.rob
C:\brazo_robot\programa\dos.rob
C:\brazo_robot\programa\tres.rob
C:\brazo_robot\programa\cuatro.rob
C:\brazo_robot\programa\cinco.rob
C:\brazo_robot\programa\seis.rob
C:\brazo_robot\programa\siete.rob
C:\brazo_robot\programa\ocho.rob
C:\brazo_robot\programa\nueve.rob
C:\brazo_robot\programa\circulo.bmp
C:\brazo_robot\programa\cruz.bmp
C:\brazo_robot\vision\configuracion

Todos estos archivos se encuentran contenidos en el disco flexible que acompaña a este documento, en el archivo llamado robot.zip.

Configuración de la cámara de vídeo

La configuración de la cámara debe ser tal que permita capturar una fotografía cada 2 segundos en el archivo: `C:\brazo_robot\vision\captura.bmp`. La manera de realizar esta configuración se discute ampliamente en la ayuda del software de la cámara y en la sección **Adquisición de la imagen** del **capítulo 3**, de este documento, en esa misma sección se encuentra la imagen muestra, que servirá para ajustar la cámara de tal manera que se obtengan imágenes de utilidad para los algoritmos de visión.

Software de operación del robot

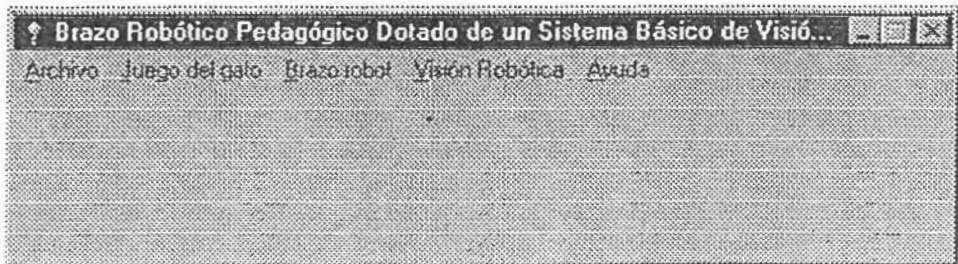


Figura 4.2 Menú principal del programa control_robot.exe

El programa principal es: `C:\brazo_robot\programa\control_robot.exe`, se puede ejecutar desde la línea de comandos o desde su ícono en el escritorio de Windows.

En esta sección sólo se revisarán las opciones: **Juego del gato** y **Visión robótica**, ya que la opción **Brazo robot**, ya fue analizada en el **Capítulo 2** de este mismo documento.

Se iniciará con la opción **Visión Robótica**, una vez seleccionada esta opción aparecerá una ventana, la cual permitirá verificar el buen funcionamiento de los algoritmos de visión, además de calibrar y cambiar diferentes parámetros que afectarán a los algoritmos que procesarán digitalmente la imagen, parámetros tales como: el umbral, los límites que segmentarán la imagen en nueve secciones, así como la sección de la imagen a la cual se le aplicará el procesamiento digital.

La configuración se almacenará en un archivo llamado **C:\brazo_robot\vision\configuracion**

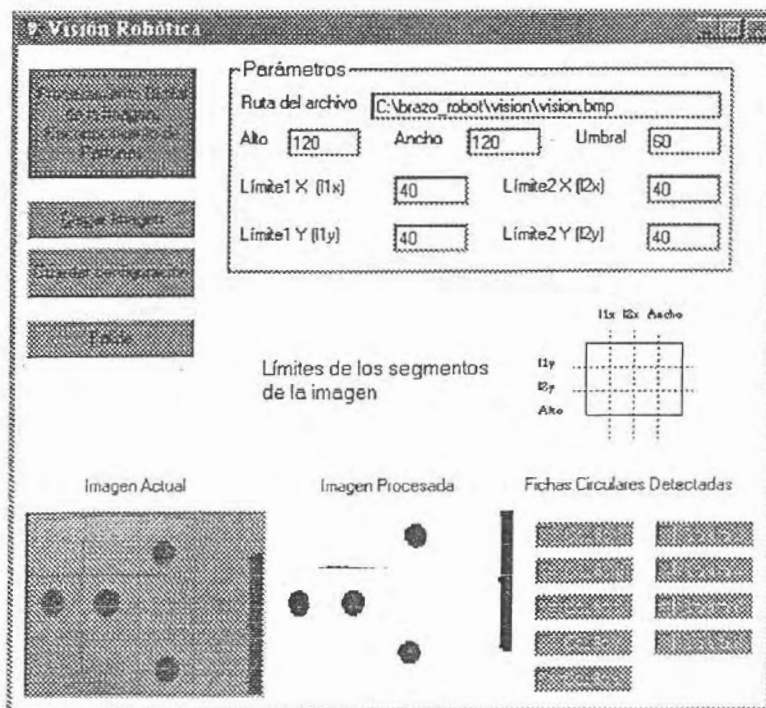


Figura 4.3 Interfaz de los algoritmos de visión artificial

La ventana anterior contiene toda la información necesaria para verificar el buen funcionamiento de los algoritmos de visión artificial, además permite calibrar, algunos parámetros, para garantizar el buen desempeño del programa, en diversas condiciones de luminosidad.

Los botones de la parte superior derecha pueden realizar tareas, tales como: cargar la imagen más reciente que ha tomado la cámara, realizar el procesamiento digital de la imagen y el reconocimiento de patrones.

Una vez que se haya aplicado el algoritmo de reconocimiento de patrones, los resultados se mostrarán en las cajas de verificación de la esquina inferior derecha que representan las casillas del juego.

En caso de que el resultado de los algoritmos no sea el esperado, se deben ajustar algunos parámetros, principalmente el umbral, dependiendo de la cantidad de luz que haya en el ambiente, hasta que se obtenga un buen resultado, una vez que se tengan resultados satisfactorios, la configuración deberá ser guardada, ya que será utilizada en tiempo de ejecución del juego. Para regresar al menú principal, se utiliza el botón salida



Figura 4.4 Ventana del juego del gato

La figura 4.4 muestra la interfaz para jugar al gato con el robot, para llegar a esta ventana es necesario seleccionar la opción **Juego del gato** del menú principal del programa *control_robot.exe*.

La ventana anterior también contiene un menú, y al elegir la opción **Juego Nuevo**, se iniciará una partida, no sin antes elegir por medio de su submenú quien la iniciará, el robot o el usuario.

La opción **Salida** se utiliza para regresar al menú principal.

La opción llamada **Modificar estado del juego**, permite cambiar el estado general del juego de manera que será posible ver como se comporta el robot ante situaciones que regularmente no suceden en un juego, esta opción se diseñó y fue de utilidad en la fase de pruebas del robot pedagógico, sin embargo, no forma parte de la dinámica del juego del gato, al seleccionar esta opción aparecerán en la esquina superior derecha de la ventana nueve casillas y un botón **Guardar**, en estas casillas se deben escribir los caracteres X, O, -, que señalarán, el nuevo estado del juego, una vez que es salvado el nuevo estado, la memoria que tiene el robot sobre la situación del juego de ese momento, será alterada por la nueva situación.

Al seleccionar la opción **Juego Nuevo**, del menú y la modalidad deseada, se iniciará el juego del gato.

En caso que el usuario sea el que inicie, debe empezar tirando en alguna de las nueve casillas del tablero con las piezas negras, y cada vez que haya terminado de tirar, es necesario indicarle al robot que ya puede él tirar, esto se hace presionando el botón **turno del robot**.

Al centro de la ventana aparecerá de manera gráfica, el estado del juego, representado por un tablero y los símbolos tradicionales del gato, esto será útil para corroborar el buen desempeño del sistema del brazo robótico.

En caso de que el robot inicie el juego, se le debe indicar, el momento en el que debe hacer su primer movimiento y cada una de las subsecuentes tiradas, de la misma manera que el caso anterior, esto se hará con el botón *turno del robot*, la mecánica del juego será la misma en cualquiera de los dos casos, hasta que exista un ganador, suceda un empate o un movimiento ilegal en el juego, en este último caso, aparecerá un mensaje que indica que se ha realizado un movimiento, no válido, y será necesario abortar el juego, e iniciar uno nuevo.

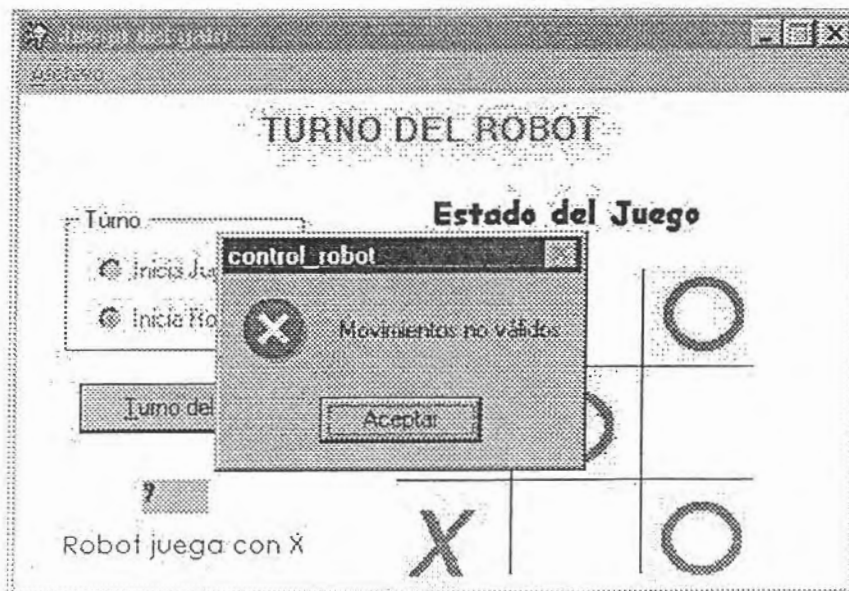


Figura 4.5 Ventana del juego del gato

En la esquina inferior izquierda, estará situada una etiqueta, la cual mostrará un número, que indicará la casilla en la cual se detectó el más reciente movimiento de una pieza negra. Esta etiqueta es un mecanismo más que servirá para verificar el buen desempeño de los algoritmos de visión artificial.

Puesta en marcha del robot

Una vez que se ha instalado el software controlador del robot y el software de la cámara de vídeo en la computadora que controlará al robot, y antes de conectar el brazo mecánico a la computadora, será necesario ejecutar el programa *control_robot.exe*, y seleccionar la opción *secuencia de ajuste*, que se encuentra contenida en el submenú *brazo robot*.

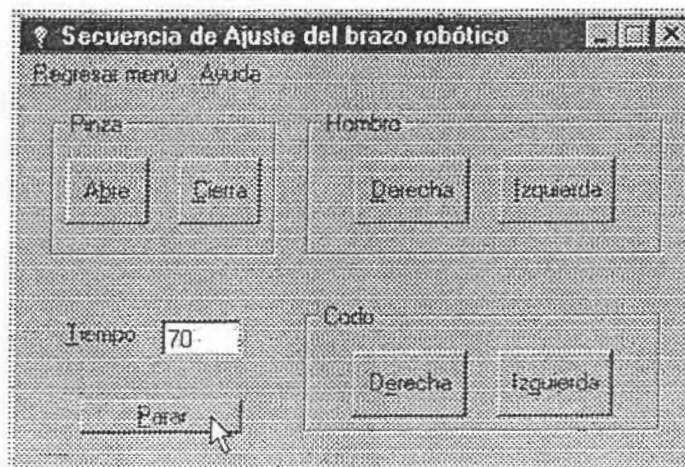


Figura 4.6 Comando para detener el brazo robot

Aparecerá la ventana de la figura 4.6, por medio de la cual se puede controlar el robot, como se vio en el *Capítulo 2* en la sección *Software de Control del Robot Pedagógico*, pero en esta ocasión, se utilizará esta ventana para asegurar que el

puerto paralelo de la PC en el cual será conectado el brazo mecánico, no contenga alguna información, que provoque que el brazo realice movimientos inesperados al momento de ser encendido, es por ello que se deberá seleccionar esta opción y se deberá dar al menos un click en el botón parar, ya que con esto se asegurará que cuando se accione el interruptor del brazo en la posición de encendido, el robot permanecerá estático.

Por último se especifica, que para finalizar una sesión con el Robot pedagógico, será necesario antes de detener completamente el brazo mecánico, llevarlo a su posición inicial, de la manera en como se menciona en la sección **Software de Control del Robot Pedagógico** del **Capítulo 2**, y posteriormente accionar el interruptor del brazo en la posición apagado.

Para salir del programa **control_robot.exe**, se debe elegir la opción **Archivo** y posteriormente **salir**.

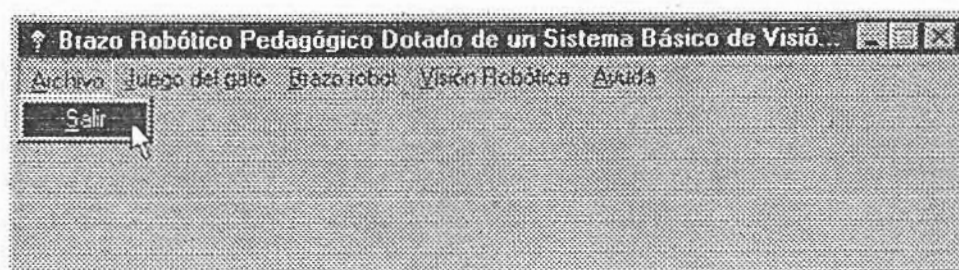


Figura 4.7 Comando para terminar el programa principal

**CÓDIGO DOCUMENTADO DE LOS
PROCEDIMIENTOS Y FUNCIONES
MÁS IMPORTANTES DEL SISTEMA**

Programa de Control del Brazo Mecánico

Este programa permite operar el brazo mecánico, proporcionando la interfaz necesaria para programar una serie de movimientos que posteriormente pueden ser reproducidos por el robot, el programa fue codificado en lenguaje C, para DOS, su código se lista a continuación.

/ Declaración de funciones */*

```
int hizq(void);
int hder(void);
int cizq(void);
int cder(void);
int pabr(void);
int pcerr(void);
int ejecuta(int);
int guarda(void);
int inicial(void);
int menu1(void);
```

```
int port = 0x378; /* Dirección del puerto al cual esta conectado el robot */
```

```

int main(void) /* Función principal */
{
    char car, cap; /* Declaración de variables */

    int value = '@';
    int i;

    value='@';
    outport(port, value); /* Se escribe al puerto para detener e inicializar el robot */

    menu1(); /* Llamada a la función menú principal */

do{ /* Bucle principal */
do{
    cap=toupper(getch());
    }while(cap!='A'&&cap!='B'&&cap!='C'&&cap!='D'&&cap!='E'&&cap!='F'&&cap!='I'&&cap!='Q');

    gotoxy(18,12); printf("%c",cap);
    gotoxy(18,12);

    switch(cap) /* Estructura de control para elegir diferentes movimientos del brazo */

        { case 'A': hizq(); break;
          case 'B': hder(); break;
          case 'C': cizq(); break;
          case 'D': cder(); break;
          case 'E': pabr(); break;
          case 'F': pcerr(); break;
          case 'I': inicial(); menu1(); break;
          case 'Q': break;
        }

        value='@';
        outport(port, value);

    }while(cap!='Q');

    textcolor(7);
    window(1,1,80,25); textbackground(0); clrscr(); /* Ventana final de la salida del programa */

    return 1;
}

```

```

int inicial() /* Función menú que permite elegir entre programar o ejecutar una secuencia */
             /* de movimientos */
{
    int port = 0x378;
    char car,cap;
    int value = '@';

    do{

        textcolor(0);
        window(1,1,80,25); textbackground(3); clrscr();
        gotoxy(24,2); printf(" Ejecutar Secuencia de Movimientos");

        textcolor(7);

        window(61,15,77,21); textbackground(0); clrscr();
        window(62,14,79,20); textbackground(1); clrscr();
        gotoxy(1,1); printf("_____");
        gotoxy(1,2); printf("\ A).toma ficha 1 |");
        gotoxy(1,3); printf("\ B).toma ficha 2 |");
        gotoxy(1,4); printf("\ C).toma ficha 3 |");
        gotoxy(1,5); printf("\ D).toma ficha 4 |");
        gotoxy(1,6); printf("\ E).toma ficha 5 |");
        gotoxy(1,7); printf("\ _____|");

        window(20,6,57,23); textbackground(0); clrscr()
        textcolor(15);
        window(22,5,59,22); textbackground(1); clrscr();

        gotoxy(16,2); printf("M E N U ");
        gotoxy(4,4); printf("P).- Programar nueva secuencia");
        gotoxy(4,5); printf("1).- Ejecutar sec. Casilla 1 ");
        gotoxy(4,6); printf("2).- Ejecutar sec. Casilla 2 ");
        gotoxy(4,7); printf("3).- Ejecutar sec. Casilla 3 ");
        gotoxy(4,8); printf("4).- Ejecutar sec. Casilla 4 ");
        gotoxy(4,9); printf("5).- Ejecutar sec. Casilla 5 ");
        gotoxy(4,10); printf("6).- Ejecutar sec. Casilla 6 ");
        gotoxy(4,11); printf("7).- Ejecutar sec. Casilla 7 ");
        gotoxy(4,12); printf("8).- Ejecutar sec. Casilla 8 ");
        gotoxy(4,13); printf("9).- Ejecutar sec. Casilla 9 ");
        gotoxy(4,14); printf("0).- Ejecutar sec. Demo ");
        gotoxy(4,15); printf("S).- Salida ");
        gotoxy(19,17);

    }do{

        cap=toupper(getch());

        jwhile(cap!='1'&&cap!='2'&&cap!='3'&&cap!='4'&&cap!='5'&&cap!='6'
        &&cap!='7'&&cap!='8'&&cap!='9'&&cap!='S'&&cap!='P'&&cap!='0'
        &&cap!='A'&&cap!='B'&&cap!='C'&&cap!='D'&&cap!='E');

        gotoxy(19,17); printf("%c",cap);
        gotoxy(19,17);
    }
}

```

```
switch(cap) /*Permite seleccionar la ejecución de alguna secuencia de movimientos */
```

```
{ case '1': ejecuta(1); break;
  case '2': ejecuta(2); break;
  case '3': ejecuta(3); break;
  case '4': ejecuta(4); break;
  case '5': ejecuta(5); break;
  case '6': ejecuta(6); break;
  case '7': ejecuta(7); break;
  case '8': ejecuta(8); break;
  case '9': ejecuta(9); break;
  case '0': ejecuta(0); break;
  case 'A': ejecuta(10); break;
  case 'B': ejecuta(11); break;
  case 'C': ejecuta(12); break;
  case 'D': ejecuta(13); break;
  case 'E': ejecuta(14); break;
  case 'P': guarda(); break;
  case 'S': break;
}
```

```
value='@';
outport(port, value); /* Detiene el robot */
```

```
}while(cap!='S');
```

```
return 1;
```

```
}
```

```
int cder() /* Función que mueve el codo del robot, hacia la derecha por 70 ms */
```

```
{
  int port = 0x378;
  int value='A';
  outport(port, value);
  delay(70);
  value='@';
  outport(port, value);
  return 1;
}
```

```
int cizq() /* Función que mueve el codo del robot, hacia la izquierda por 70 ms */
```

```
{
  int port = 0x378;
  int value='B';
  outport(port, value);
  delay(70);
  value='@';
  outport(port, value);
  return 1;
}
```

```
int pcerr() /* Función que cierra la pinza */
```

```
{  
  int port = 0x378;  
  int value='D';  
  outport(port, value);  
  delay(500);  
  value='@';  
  outport(port, value);  
  return 1;  
}
```

```
int pabr() /* Función que abre la pinza */
```

```
{  
  int port = 0x378;  
  int value='H';  
  outport(port, value);  
  delay(500);  
  value='@';  
  outport(port, value);  
  return 1;  
}
```

```
int hder() /* Función que mueve el hombro del robot, hacia la derecha por 100 ms */
```

```
{  
  int port = 0x378;  
  int value='P';  
  outport(port, value);  
  delay(100);  
  value='@';  
  outport(port, value);  
  return 1;  
}
```

```
int hizq() /* Función que mueve el hombro del robot, hacia la izquierda por 100 ms */
```

```
{  
  int port = 0x378;  
  int value="";  
  outport(port, value);  
  delay(100);  
  value='@';  
  outport(port, value);  
  return 1;  
}
```

```

int ejecuta(int par) /* Función que ejecuta una secuencia de movimientos de un archivo */
                    /* especificado */
{
    FILE *pt;
    char archivo[20];
    char var[10];

    switch(par)
    { case 1: strcpy(archivo,"uno.rob");    break;
      case 2: strcpy(archivo,"dos.rob");   break;
      case 3: strcpy(archivo,"tres.rob");  break;
      case 4: strcpy(archivo,"cuatro.rob"); break;
      case 5: strcpy(archivo,"cinco.rob"); break;
      case 6: strcpy(archivo,"seis.rob");  break;
      case 7: strcpy(archivo,"siete.rob"); break;
      case 8: strcpy(archivo,"ocho.rob");  break;
      case 9: strcpy(archivo,"nueve.rob"); break;
      case 0: strcpy(archivo,"demo.rob");  break;
      case 10: strcpy(archivo,"fich1.rob"); break;
      case 11: strcpy(archivo,"fich2.rob"); break;
      case 12: strcpy(archivo,"fich3.rob"); break;
      case 13: strcpy(archivo,"fich4.rob"); break;
      case 14: strcpy(archivo,"fich5.rob"); break;
    }

    if ((pt = fopen(archivo, "r"))
        == NULL)
    {
        fprintf(stderr, "no se puede abrir el archivo de movimientos %d ",par);
        getch();
        return 1;
    }

    while(!feof(pt))
    {
        fgets(var,10,pt);

        switch(var[0])
        { case '1': hizq();    break;
          case '2': hder();   break;
          case '3': cizq();   break;
          case '4': cder();   break;
          case '5': pabr();   break;
          case '6': pcerr();  break;
          case '7': delay(300); break;
          case '8':          break;
        }
    }

    fclose(pt);
    return 1;
}

```

```

int guarda() /* Función que guarda una secuencia de movimientos en un archivo */
{
int x,cas,cas1;
FILE *pt;
char cap;
char archivo[20];

do{
    archivo[15]='\0';
    textcolor(0);
    window(1,1,80,25); textbackground(3); clrscr();
    gotoxy(24,2); printf("Programar Secuencia de Movimientos");
    textcolor(7);
    window(57,15,74,21); textbackground(0); clrscr();
    window(58,14,76,20); textbackground(1); clrscr();
    gotoxy(1,1); printf("_____");
    gotoxy(1,2); printf("A).tomar ficha 1 |");
    gotoxy(1,3); printf("B).tomar ficha 2 |");
    gotoxy(1,4); printf("C).tomar ficha 3 |");
    gotoxy(1,5); printf("D).tomar ficha 4 |");
    gotoxy(1,6); printf("E).tomar ficha 5 |");
    gotoxy(1,7); printf("_____|");

    window(25,6,52,23); textbackground(0); clrscr();
    textcolor(15);
    window(27,5,54,22); textbackground(1); clrscr();

    gotoxy(4,2); printf(" Guardar movimientos ");
    gotoxy(4,3); printf("asociados a la casilla:");
    gotoxy(7,5); printf("1).- Casilla 1 ");
    gotoxy(7,6); printf("2).- Casilla 2 ");
    gotoxy(7,7); printf("3).- Casilla 3 ");
    gotoxy(7,8); printf("4).- Casilla 4 ");
    gotoxy(7,9); printf("5).- Casilla 5 ");
    gotoxy(7,10); printf("6).- Casilla 6 ");
    gotoxy(7,11); printf("7).- Casilla 7 ");
    gotoxy(7,12); printf("8).- Casilla 8 ");
    gotoxy(7,13); printf("9).- Casilla 9 ");
    gotoxy(7,14); printf("0).- Demo ");
    gotoxy(7,15); printf("S).- Salir ");
    gotoxy(14,17);

do{
cap=toupper(getch());
}while(cap!='1'&&cap!='2'&&cap!='3'&&cap!='4'&&cap!='5'&&cap!='6'
&&cap!='7'&&cap!='8'&&cap!='9'&&cap!='S'&&cap!='0'
&&cap!='A'&&cap!='B'&&cap!='C'&&cap!='D'&&cap!='E');

gotoxy(14,17); printf("%c",cap);
gotoxy(14,17);

```



```

switch(cap)
{ case '1': strcpy(archivo,"uno.rob");   cas=1; break;
  case '2': strcpy(archivo,"dos.rob");   cas=2; break;
  case '3': strcpy(archivo,"tres.rob");   cas=3; break;
  case '4': strcpy(archivo,"cuatro.rob"); cas=4; break;
  case '5': strcpy(archivo,"cinco.rob");  cas=5; break;
  case '6': strcpy(archivo,"seis.rob");   cas=6; break;
  case '7': strcpy(archivo,"siete.rob");  cas=7; break;
  case '8': strcpy(archivo,"ocho.rob");   cas=8; break;
  case '9': strcpy(archivo,"nueve.rob");  cas=9; break;
  case '0': strcpy(archivo,"demo.rob");   cas=0; break;
  case 'A': strcpy(archivo,"fich1.rob");  cas=-1; cas1=1; break;
  case 'B': strcpy(archivo,"fich2.rob");  cas=-1; cas1=2; break;
  case 'C': strcpy(archivo,"fich3.rob");  cas=-1; cas1=3; break;
  case 'D': strcpy(archivo,"fich4.rob");  cas=-1; cas1=4; break;
  case 'E': strcpy(archivo,"fich5.rob");  cas=-1; cas1=5; break;
  case 'S': break;
}

if(cap !='S')
{
    window(14,11,64,17); textbackground(0); clrscr();

    textcolor(15);
    window(16,10,66,16); textbackground(4); clrscr();
    gotoxy(18,1); printf("ADVERTENCIA !!");

if(cas>=0)
{ gotoxy(2,3); printf("Se dispone a guardar una secuencia de movimientos");
  gotoxy(2,4); printf("para la casilla No. %c, la secuencia anterior",cap);
  gotoxy(2,5); printf("sera borrada.");
  gotoxy(14,6); printf("Desea continuar S/N ");
}
else
{gotoxy(2,3); printf("Se dispone a guardar una secuencia de movimientos");
  gotoxy(2,4); printf("para tomar la ficha No. %d, la secuencia anterior",cas1);
  gotoxy(2,5); printf("sera borrada.");
  gotoxy(14,6); printf("Desea continuar S/N ");
}

do{
    cap=toupper(getch());
    }while(cap!='S'&&cap!='N');

textcolor(7);
window(1,1,80,25); textbackground(0); clrscr();

```

```

if(cap=='S')
{
    // guardar en archivo

    if ((pt = fopen(archivo, "w+"))
        == NULL)
    {
        fprintf(stderr, "no se puede abrir el archivo\n");
        return 1;
    }

    textcolor(0);
    window(1,1,80,25); textbackground(3); clrscr();

if(cas>=0) {
    gotoxy(15,3); printf("GUARDANDO SECUENCIA DE MOVIENTOS PARA LA CASILLA
%d",cas);
}

else {
    gotoxy(15,3); printf(" GUARDANDO SECUENCIA DE MOVIENTOS PARA LA FICHA %d
",cas1);
}

    window(25,7,52,20); textbackground(0); clrscr();
    textcolor(15);
    window(27,6,54,19); textbackground(1); clrscr();

    gotoxy(11,2); printf("M E N U ");
    gotoxy(4,4); printf("A).- Hombro Izquierda ");
    gotoxy(4,5); printf("B).- Hombro Derecha ");
    gotoxy(4,6); printf("C).- Codo Izquierda ");
    gotoxy(4,7); printf("D).- Codo Derecha ");
    gotoxy(4,8); printf("E).- Pinza Abre ");
    gotoxy(4,9); printf("F).- Pinza Cierra ");
    gotoxy(4,10); printf("R).- Retardo 300 ms ");
    gotoxy(4,11); printf("T).- Terminar ");
    gotoxy(14,13);

do{
do{

    cap=toupper(getch());

}while(cap!='A'&&cap!='B'&&cap!='C'&&cap!='D'&&cap!='E'&&cap!='F'&&cap!='T'&&cap!='R');

    gotoxy(14,13);
    gotoxy(14,13);
    printf("%c",cap);

    switch(cap)
    { case 'A': hizq(); fputs("1hiz\n",pt); break;
      case 'B': hder(); fputs("2hde\n",pt); break;
      case 'C': cizq(); fputs("3ciz\n",pt); break;
      case 'D': cder(); fputs("4cde\n",pt); break;

```

```

        case 'E': pabr(); fputs("5pab\n",pt); break;
        case 'F': pcerr(); fputs("6pci\n",pt); break;
        case 'R':      fputs("7ret\n",pt); break;
        case 'T':      fputs("8ter\n",pt); break;
    }

}while(cap!='T');

    fclose(pt);
    gotoxy(6,13); printf("Proceso terminado");
    delay(500);

} //igual a s if

} // dire de s if

else{ }

}while(cap!='S');

    return 1;
}

int menu1(){ /* Función que imprime el menú principal en la pantalla */

    textcolor(0);
    window(1,1,80,25); textbackground(3); clrscr();
    clrscr();
    gotoxy(29,4); printf("Ajuste del brazo robot\n\n");
    window(20,9,56,21); textbackground(0); clrscr();
    textcolor(15);
    window(22,8,58,20); textbackground(1); clrscr();

    gotoxy(15,2); printf("M E N U");
    gotoxy(4,4); printf("A).- Hombro Izquierda ");
    gotoxy(4,5); printf("B).- Hombro Derecha ");
    gotoxy(4,6); printf("C).- Codo Izquierda ");
    gotoxy(4,7); printf("D).- Codo Derecha ");
    gotoxy(4,8); printf("E).- Pinza Abre ");
    gotoxy(4,9); printf("F).- Pinza Cierra ");
    gotoxy(4,10); printf("I).- Brazo en posición inicial ");
    gotoxy(4,11); printf("Q).- SALIR ");
    gotoxy(18,12);

return 1;

}

```

Algoritmos de Visión Artificial

Los algoritmos de Visión Artificial, fueron codificados en lenguaje Visual Basic 5, para Windows, se revisará el procedimiento de lectura y segmentación de la imagen, así como el algoritmo de Reconocimiento de Patrones y toma de decisiones, a continuación se lista el código.

```

Sub leer_imagen() ' Procedimiento que lee y segmenta la imagen utilizando solo el canal
' verde
On Error GoTo et15

Dim i, j, col, ren, ar1, va As Integer, char$ ' Definición de variables
Dim cadena As String
Dim var1
Dim p1, p2, p3 As Integer
'Dim imagen(1 To 120, 1 To 480) As Integer
'Dim reconoce(1 To 120, 1 To 120) As Integer

Screen.MousePointer = vbHourglass ' Se cambia el puntero del mouse al modo de espera

Call lee_configuracion ' Se lee la configuración actual

'Se hace una copia de la imagen
'y se trabaja con ella para evitar problemas de concurrencia

FileCopy "C:\brazo_robot\vision\captura.bmp", ruta
ar1 = FreeFile ' archivo entrada
Open ruta For Binary As #ar1

' Se lee la cabecera del archivo BMP

cadena = Input(54, #ar1) 'leo la inf

For ren = 1 To 120
i = 1
For col = 1 To 160

char$ = Input(1, #ar1) ' Se lee el canal rojo
char$ = Input(1, #ar1) ' Se lee el canal verde y se guarda

p2 = Asc(char$)

```

' Se aplica el umbral para segmentar la imagen

```
If p2 > umbral Then
    p2 = 255
Else
    p2 = 0
End If
```

```
imagen_p(ren, i) = p2
char$ = Input(1, #ar1) ' Se lee el canal azul
```

```
    j = i + 1
Next col
Next ren
```

Screen.MousePointer = Deafult ' Se cambia el puntero del mouse en el modo por omisión

Close #ar1

Exit Sub

```
et15:
Screen.MousePointer = Deafult
numero = Err
Select Case numero
Case 1
Case Else
MsgBox "Error en la imagen"
Exit Sub
End Select
```

End Sub

Sub rec_patrones() ' Procedimiento de reconocimiento de patrones y toma de decisiones

Dim i, j, px As Integer

' Se lee la imagen y se obtiene una matriz binaria

Call leer_imagen

```
For i = 1 To 9
c(i) = 0
Next i
```

* Se Calcula el área de las regiones negras, para cada una de las 9 secciones de la imagen

```
For j = 1 To l1y
  For i = 1 To l1x
    px = imagen_p(j, i)
```

```
    If px = 0 Then
      c(1) = c(1) + 1
    End If
```

```
  Next i
Next j
```

```
For j = 1 To l1y
  For i = l1x To (l1x + l2x)
    px = imagen_p(j, i)
```

```
    If px = 0 Then
      c(2) = c(2) + 1
    End If
```

```
  Next i
Next j
```

```
For j = 1 To l1y
  For i = (l2x + l1x) To ancho
    px = imagen_p(j, i)
```

```
    If px = 0 Then
      c(3) = c(3) + 1
    End If
```

```
  Next i
Next j
```

```
For j = l1y To (l2y + l1y)
  For i = 1 To l1x
    px = imagen_p(j, i)
```

```
    If px = 0 Then
      c(4) = c(4) + 1
    End If
```

```
  Next i
Next j
```

```
For j = l1y To (l1y + l2y)
  For i = l1x To (l2x + l1x)
    px = imagen_p(j, i)
```

```
If px = 0 Then  
c(5) = c(5) + 1  
End If
```

```
Next i  
Next j
```

```
For j = l1y To (l2y + l1y)  
For i = (l1x + l2x) To ancho  
px = imagen_p(j, i)
```

```
If px = 0 Then  
c(6) = c(6) + 1  
End If
```

```
Next i  
Next j
```

```
For j = (l1y + l2y) To alto  
For i = 1 To l1x  
px = imagen_p(j, i)
```

```
If px = 0 Then  
c(7) = c(7) + 1  
End If
```

```
Next i  
Next j
```

```
For j = (l1y + l2y) To alto  
For i = l1x To (l2x + l1x)  
px = imagen_p(j, i)
```

```
If px = 0 Then  
c(8) = c(8) + 1  
End If
```

```
Next i  
Next j
```

```
For j = (l1y + l2y) To alto  
For i = (l1x + l2x) To ancho  
px = imagen_p(j, i)
```

```
If px = 0 Then  
c(9) = c(9) + 1  
End If
```

```
Next i  
Next j
```

' Toma de decisiones, presencia o ausencia de fichas oscuras en una casilla determinada

```
If c(1) > 110 and c(1) < 237 Then  
c(1) = 1  
Else  
c(1) = 0  
End If
```

```
If c(2) > 110 and c(2) < 237 Then  
c(2) = 1  
Else  
c(2) = 0  
End If
```

```
If c(3) > 110 and c(3) < 237 Then  
c(3) = 1  
Else  
c(3) = 0  
End If
```

```
If c(4) > 110 and c(4) < 237 Then  
c(4) = 1  
Else  
c(4) = 0  
End If
```

```
If c(5) > 110 and c(5) < 237 Then  
c(5) = 1  
Else  
c(5) = 0  
End If
```

```
If c(6) > 110 and c(6) < 237 Then  
c(6) = 1  
Else  
c(6) = 0  
End If
```

```
If c(7) > 110 and c(7) < 237 Then  
c(7) = 1  
Else  
c(7) = 0  
End If
```

```
If c(8) > 110 and c(8) < 237 Then  
c(8) = 1  
Else  
c(8) = 0  
End If
```

```
If c(9) > 110 and c(9) < 237 Then  
c(9) = 1  
Else
```



```
c(9) = 0  
End If  
End Sub
```

Algoritmo del Juego del Gato

El algoritmo del Juego del gato se implementó con técnicas tradicionales de programación, en lenguaje C para DOS, éste se comunica con el resto de los programas que conforman el sistema del Brazo Robótico Pedagógico dotado de un Sistema Básico de Visión Artificial, por medio de un archivo llamado *Tablero*, en el cual se describe la situación más reciente del juego, a continuación se lista las partes del código más relevantes.

```
struct nodetype {          /* Estructura de datos, para representar el tablero de juego */  
    char board[4][4];  
    int turn;  
    struct nodetype *hijo;  
    struct nodetype *proximo;  
};  
  
typedef struct nodetype *NODEPTR;      /* Definiciones de variables globales */  
char op1='';  
struct nodetype **cabeza;  
char posic[10];  
int xx=0,yy=0;  
char archivo[100];  
NODEPTR buildtree(char[4][4],int);  
  
main(int argc, char *argv[]) /* Función principal */  
{  
    int i,j,t=0,turno=0,pp;  
    int primer_tiro=0;  
    char tab[4][4],tab1[4][4],car;  
    FILE *ap;  
    strcpy(archivo,argv[1]);  
    clrscr();
```

```

if (argc != 2)
{
printf("Error, falta nombre de archivo\n");
printf("Gat_rob Archivo\n");
exit(0);
}

else
{
if ((ap = fopen(argv[1], "r")) == NULL)
{printf("Error al tratar de abrir el archivo\n %s", argv[1]);
exit(1);
}

for(i=0; i<10; i++) posic[i]=' '; /* inicialización de variables */
for(i=0; i<3; i++) {
for(j=0; j<3; j++){
tab[i][j]=' ';
tab1[i][j]=' ';
}}
tab[0][3]='\0';
tab[1][3]='\0';
tab[2][3]='\0';

pp=0;

while (t<13)
{
car=fgetc(ap);
//printf("%c", car);
if(car=='X' || car=='0'){primer_tiro=1;}
if(car=='0' || car=='X' || car=='-')
{pp++;}

if(car=='0')
{
switch(pp)
{
case 1: tab[0][0]='0'; posic[0]='*'; break;
case 2: tab[0][1]='0'; posic[1]='*'; break;
case 3: tab[0][2]='0'; posic[2]='*'; break;
case 4: tab[1][0]='0'; posic[3]='*'; break;
case 5: tab[1][1]='0'; posic[4]='*'; break;
case 6: tab[1][2]='0'; posic[5]='*'; break;
case 7: tab[2][0]='0'; posic[6]='*'; break;
case 8: tab[2][1]='0'; posic[7]='*'; break;
case 9: tab[2][2]='0'; posic[8]='*'; break;

};
}

if(car=='X')
{
switch(pp)

```

```

    {
    case 1: tab[0][0]='X'; posic[0]='*'; break;
    case 2: tab[0][1]='X'; posic[1]='*'; break;
    case 3: tab[0][2]='X'; posic[2]='*'; break;
    case 4: tab[1][0]='X'; posic[3]='*'; break;
    case 5: tab[1][1]='X'; posic[4]='*'; break;
    case 6: tab[1][2]='X'; posic[5]='*'; break;
    case 7: tab[2][0]='X'; posic[6]='*'; break;
    case 8: tab[2][1]='X'; posic[7]='*'; break;
    case 9: tab[2][2]='X'; posic[8]='*'; break;

    };
}

    if(t==12)
    {
        if (car=='1'){ op1='A'; } /* Maquina inicia */
        else{op1='B';} /* Jugador inicia */
    }
    t++;
}

}

fclose(ap);

```

/* Inicia máquina , primer tiro*/

```

if(op1=='A'&&primer_tiro==0)
{ tab[0][0]='X'; posic[0]='*';
  imprimir(tab);
}
if(op1=='A'&&primer_tiro!=0)
{
  int pos;
  xx=0; yy=0;
  posib_triunfo(tab,'X');
  if(xx!=0 || yy!=0){
    tab[xx][yy]='X';
  }
else{
  xx=0; yy=0;
  posib_triunfo(tab,'0');
  if(xx!=0 || yy!=0){
    tab[xx][yy]='X';
  }
  else
  {
    if(tab[1][1]==' '){tab[1][1]=tab[1][1]='X'; }
    else{
      nextmove(tab,9,'X',tab1);
      for(i=0; i<3; i++) {
        for(j=0; j<3; j++){
          tab[i][j]=tab1[i][j].;}}
    }
  }
}

```

```

    }
  }
}

```

```
imprimir(tab);
```

```

if(tab[0][0]!=' '){posic[0]='*'; }
if(tab[0][1]!=' '){posic[1]='*'; }
if(tab[0][2]!=' '){posic[2]='*'; }
if(tab[1][0]!=' '){posic[3]='*'; }
if(tab[1][1]!=' '){posic[4]='*'; }
if(tab[1][2]!=' '){posic[5]='*'; }
if(tab[2][0]!=' '){posic[6]='*'; }
if(tab[2][1]!=' '){posic[7]='*'; }
if(tab[2][2]!=' '){posic[8]='*'; }
sí_gano(tab);

```

```
} /* Fin de, si la maquina es primero */
```

```
/* La máquina inicia el juego */
```

```

if(op1=='B'){
  int pos;
  xx=0; yy=0;
  posib_triunfo(tab, '0');

  if(xx!=0 || yy!=0){
    tab[xx][yy]='0';
  }
  else{
    xx=0; yy=0;
    posib_triunfo(tab, 'X');
    if(xx!=0 || yy!=0){
      tab[xx][yy]='0';
    }
    else
    {
      if(tab[1][1]==' ')
        {tab[1][1]=tab1[1][1]='0';}
      else{

        if(tab[0][2]=='X'&&tab[1][1]=='0'&&tab[2][0]=='X'&&tab[0][0]==' '
        && tab[0][1]==' ' &&tab[1][0]==' ' &&tab[1][2]==' ' &&tab[2][1]==' '
        &&tab[2][2]==' '){tab[2][1]=tab1[2][1]='0';}

        else{

          if(tab[0][2]==' ' &&tab[1][1]=='X' &&tab[2][0]==' ' &&tab[0][0]=='0'
          && tab[0][1]==' ' &&tab[1][0]==' ' &&tab[1][2]==' ' &&tab[2][1]==' '
          &&tab[2][2]=='X'){tab[0][2]=tab1[0][2]='0';}
          else{

```



```
ptree->hijo =NULL;
ptree->proximo =NULL;
```

```
/* Creación del resto del árbol de juegos */
```

```
expand(ptree, 0, looklevel);
return(ptree);
}
```

```
/* Función de evaluación de las posibles jugadas */
```

```
evaluacion(char ev[4][4],char sib)
{ int i,j;
  int x,y,posibx=0,posiby=0;
  for(j=0; j<3; j++){
    for(i=0; i<3; i++){
      ev[j][i]=ev[j][i];
    }
  }

  for(j=0; j<3; j++){
    x=y=0;
    for(i=0; i<3; i++){
      if(ev[j][i]=='X'){x++;}
      if(ev[j][i]=='O'){y++;}
    }
    if(x>=1 && y==0){posibx++;}
    if(y>=1 && x==0){posiby++;}
  }

  for(j=0; j<3; j++){
    x=y=0;
    for(i=0; i<3; i++){
      if(ev[j][i]=='X'){x++;}
      if(ev[j][i]=='O'){y++;}
    }
    if(x>=1 && y==0){posibx++;}
    if(y>=1 && x==0){posiby++;}
  }

  x=y=0;
  for(i=0; i<3; i++)
  {
    if(ev[i][i]=='X'){x++;}
    if(ev[i][i]=='O'){y++;}
  }
  if(x>=1 && y==0){posibx++;}
  if(y>=1 && x==0){posiby++;}
  x=y=0;
  if(ev[2][0]=='X'){x++;}
```

```

if(ev[1][1]=='X'){x++;}
if(ev[0][2]=='X'){x++;}

    if(ev[2][0]=='0'){y++;}
    if(ev[1][1]=='0'){y++;}
    if(ev[0][2]=='0'){y++;}
    if(x>=1 && y==0){posibx++;}
    if(y>=1 && x==0){posiby++;}

    if(sib=='X'){return(posibx-posiby); }
    else{return(posiby-posibx);}

return 1;
}

```

/* Contenido del Archivo tablero */

```

--X
XX-
0-X
2

```

El parámetro 1 indica que la maquina tira primero con X
 2 El usuario tira primero con X
 NOTA: los caracteres que se utilizan son X, 0, -