

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**FACULTAD DE INGENIERÍA ELECTRÓNICA**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO EN INGENIERÍA ELECTRÓNICA**

**“PROTOTIPO DE IMPRESORA PARA MODELOS 3D EN  
CERA”**

**Juan Carlos Bonilla Arroyo  
Santiago Javier Dávalos Ávila**

**SANGOLQUÍ - ECUADOR**

**2006**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado titulado “Prototipo de Impresora para Modelos 3D en Cera” ha sido desarrollado en su totalidad por el Sr. Juan Carlos Bonilla Arroyo con C.I. 171420144-7, y el Sr. Santiago Javier Dávalos Ávila con C.I. 171271057-1 bajo nuestra dirección.

Ing. Paúl Ayala  
DIRECTOR

Ing. Hugo Ortiz  
CODIRECTOR

## **AGRADECIMIENTO**

“Tengo muchas personas a quien agradecer la realización de este proyecto y sobre todo de este título, en primer lugar quiero agradecer al Sr. Víctor Velasteguí que a más de ser un apoyo fundamental en la parte técnica llegó a ser un amigo. A mi papi que se convirtió en albañil, carpintero, electrónico, electricista, y diseñador de robots, y que gracias a él, la persona que más respeto, he aprendido un poco de todo. A mi mami que sin saber electrónica me dió el apoyo suficiente para seguir adelante. A los directores de tesis que fueron un apoyo de principio a fin de la realización de este proyecto.

Un agradecimiento especial a mi novia Karla que se esforzó tanto como yo para el cumplimiento de este objetivo.

Quiero agradecer también a Juan Carlos que a más de ser compañeros de tesis después de convivir seis meses se convirtió en un buen amigo que me aguantó todos mis malos humores y enojos.”

Santiago Dávalos.

## **AGRADECIMIENTO**

“Mi agradecimiento va dirigido especialmente a mis padres, que me han apoyado, guiado y ayudado incondicionalmente en toda mi vida y en esta etapa universitaria, a mi hermana que me brinda toda su ayuda en los momentos que más necesitaba, a mi familia que de una u otra manera siempre me han estado dando ánimos para seguir adelante.

A la familia Dávalos Ávila que me ayudaron en la realización de este proyecto. A mi amigo y compañero de tesis Santiago que me brinda su apoyo y comprensión para concluir nuestro objetivo. Al Sr. Víctor que ayudó en la construcción del prototipo.”

Juan Carlos Bonilla.

## **DEDICATORIA**

“No tengo palabras para agradecer a las personas que han estado estos seis años apoyándome en los buenos y malos momentos, en especial quiero agradecer a mis padres que son el ejemplo de mi vida que sin ellos no podría haber llegado a donde me encuentro, a mis hermanas que siempre han estado dispuestas a extenderme sus manos como unas amigas. A mi amiga y novia Karla que ha sido el apoyo fundamental en estos últimos años y me ha ayudado a conseguir, en ella, la clave que me faltaba para la felicidad en mi vida, y quiero dar un agradecimiento especial, aunque no se de cuenta todavía, a mi sobrina Andrea que le ha dado un nuevo sentido a la vida de mi familia.

A todos ellos les dedico mi esfuerzo.”

Santiago Dávalos.

## **DEDICATORIA**

“Dedico este trabajo a mis Padres, Roberto y Teresa, por el esfuerzo que han hecho durante estos veinticuatro años de mi vida para que llegue a ser un hombre correcto y de bien. A mi hermana Anita y a mi sobrino José David que llenan de alegría mi hogar y me dan fuerzas para seguir adelante.

A mi familia y amigos que son parte importante de mí, que estuvieron apoyándome y brindándome su ayuda en los momentos felices y difíciles de esta etapa de mi vida. “

Juan Carlos Bonilla.

## **PRÓLOGO**

Este proyecto de desarrollo tecnológico, nuevo en nuestro país, a más de abrir nuevos campos en cualquier tipo de industria, permite ir un paso delante de la simple visualización de la información habitualmente conocida, a través de la construcción de modelos tridimensionales que permiten a los usuarios comprobar el comportamiento y características especiales de ellos, obteniendo mucha más información.

Este prototipo de Impresora 3D utiliza una combinación de las técnicas de fabricación de inyección y fabricación por capas, mediante el funcionamiento de un robot cartesiano que permite a su herramienta final, dispensador de cera, moverse en cualquier coordenada del espacio definida por movimientos lineales de sus tres ejes, inyectando cera caliente que obedece las ordenes enviadas por el Software de control.

# INDICE

<b>AGRADECIMIENTO</b>	<b>3</b>
<b>DEDICATORIA</b>	<b>4</b>
<b>PRÓLOGO</b>	<b>5</b>
<b>INDICE</b>	<b>6</b>
<b>CAPÍTULO I</b>	<b>9</b>
<b>INTRODUCCION</b>	<b>9</b>
1.1 Impresoras para modelos 3D existentes	9
1.1.1 Estereolitografía (SLA)	10
1.1.2 Sinterización selectiva por láser (SLS)	13
1.1.3 Moldeado por deposición de hilo fundido (FDM)	16
1.1.4 Fabricación por corte y laminado (LOM)	18
1.1.5 Fabricación por inyección (Ink Jet Printing)	20
1.1.6 Estratoconception	22
1.2 Reconocimiento de los tipos de Robots y controladores existentes	23
1.2.1 Clasificación de la robótica desde el punto de vista industrial	24
1.2.2 Clasificación desde el punto de vista del control de sus movimientos	25
1.2.3 Clasificación desde el punto de vista de su geometría	26
1.3 Descripción de software de dibujo 3D, lenguajes de programación	28
1.3.1 AutoCAD	28
1.3.2 RhinoCeros	29
1.3.3 Lenguajes de programación para interfaces HMI	31
<b>CAPÍTULO II</b>	<b>34</b>
<b>DISEÑO DE LA ESTRUCTURA MECÁNICA DEL ROBOT CARTESIANO</b>	<b>34</b>
2.1 Diseño del robot cartesiano	34
2.2 Diseño parte mecánica del robot cartesiano	40
2.2.1 Fase 1	40
2.2.2 Fase 2	42
2.2.3 Fase 3	45
2.2.4 Fase 4	46
2.2.5 Fase 5	49
2.2.6 Fase 6	50
2.2.7 Fase 7	52
2.2.8 Fase 8	54

2.3	Diseño de la parte motriz del robot	56
2.3.1	Motores	56
2.3.2	Acoples y Soportes	59
2.4	Diseño del sistema dispensador de cera	63
<b>CAPÍTULO III</b>		<b>68</b>
<b>HARDWARE DE CONTROL</b>		<b>68</b>
3.1	Diseño del hardware control	68
3.1.1	Tipos de Sistemas de control	68
3.1.2	Computadora	71
3.1.3	Controlador	77
3.1.4	Etapas de Potencia	83
3.2	Programación del Controlador	93
3.2.1	Lógica de Programación	94
3.2.2	Hardware de Programación	97
<b>CAPÍTULO IV</b>		<b>98</b>
<b>DESARROLLO DEL SOFTWARE</b>		<b>98</b>
4.1	Software de conversión de coordenadas	98
4.1.1	Convertidor ACE	99
4.2	Interfaz HMI	100
4.2.1	HMI	101
4.2.2	Archivos CAD (*.dxf).	101
4.2.3	Conversión archivo *.dxf a Código G	102
4.2.4	Tipo de Dibujo (Ortopoligonal o Curvas)	105
4.2.5	Modelo Prediseñado	105
4.2.6	Impresión del Modelo	106
<b>CAPÍTULO V</b>		<b>108</b>
<b>IMPLEMENTACIÓN</b>		<b>108</b>
5.1	Implementación Robot Cartesiano	108
5.1.1	Fase 1	108
5.1.2	Fase 2	111
5.1.3	Fase 3	113
5.1.4	Fase 4	114
5.1.5	Fase 5	114
5.1.6	Fase 6	114
5.1.7	Fase 7	116
5.1.8	Fase 8	117
5.2	Implementación sistema dispensador de cera	119

5.3	Implementación del hardware de control y potencia	121
5.4	Prototipo de Impresora para modelos 3D en cera – final	126
<b>CAPÍTULO VI</b>		<b>128</b>
<b>PRUEBAS Y RESULTADOS</b>		<b>128</b>
6.1	Pruebas del sistema completo	128
6.2	Resultados	130
6.2.1	Resultados con Cera	136
<b>CAPÍTULO VII</b>		<b>142</b>
<b>CONCLUSIONES Y RECOMENDACIONES</b>		<b>142</b>
7.1	Conclusiones	142
7.2	Recomendaciones	144
<b>BIBLIOGRAFÍA</b>		<b>146</b>
<b>ANEXOS</b>		<b>149</b>
<b>MANUAL DE USUARIO</b>		<b>194</b>
<b>ÍNDICE DE TABLAS</b>		<b>214</b>

## CAPÍTULO I

### INTRODUCCIÓN

#### 1.1 Impresoras para modelos 3D existentes

Tocar la información puede ayudar a visualizarla. La idea básica consiste en la descomposición de un modelo en un conjunto de secciones o capas y fabricarlas una por una. Una vez acabada una sección o capa, se construye encima la siguiente, y así sucesivamente hasta completar el objeto.

“Hoy en día existen impresoras 3D capaces de crear modelos en resinas, metales, cerámica, elastómeros e incluso papel con precisiones de hasta 0.025 mm. La propia naturaleza de estos sistemas y procesos, capa a capa hace que la complejidad del objeto sea irrelevante ya que se pueden construir objetos con oquedades, entrantes y salientes prácticamente sin limitación. El tiempo de construcción se mide en unas pocas horas dependiendo del volumen del objeto.”<sup>1</sup>

Las impresoras 3D tienen que ver mucho con la visualización de información, ya que los ingenieros que desarrollan piezas u objetos complejos pueden obtener mucha más información de su modelo teniéndolo en la mano y

---

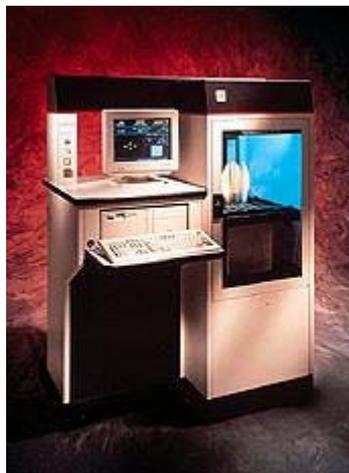
<sup>1</sup> DURSTELES Juan, InfoVis.net , 2006, Alemania

comprobando su comportamiento que simplemente viéndolo en una proyección sobre la pantalla de un computador; el llamado prototipado rápido (rapid prototyping) permite realizar un prototipo funcional en una fracción del tiempo de antaño.

Para crear los modelos se tiene una infinidad de sistemas que se basan en distintos procesos, como por ejemplo:

### 1.1.1 Estereolitografía (SLA)

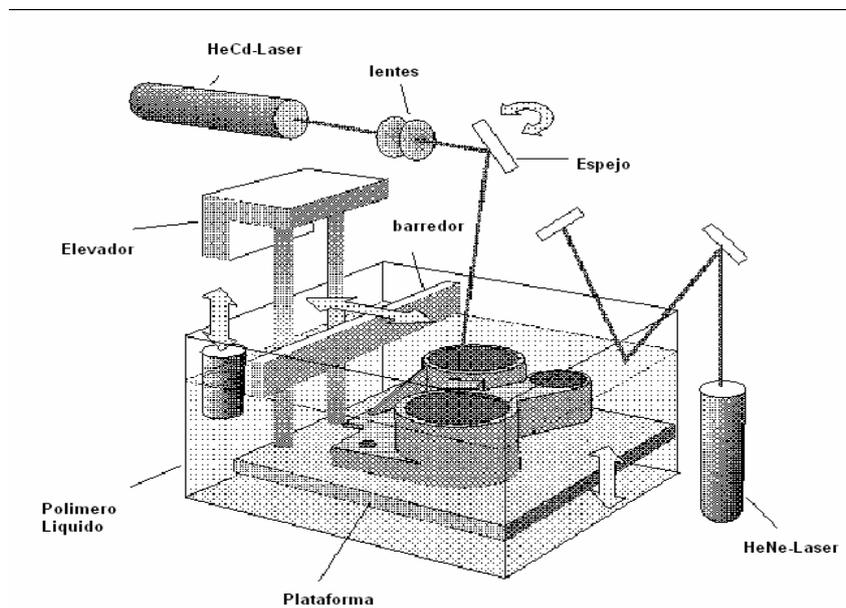
La estereolitografía es la tecnología pionera en lo que refiere a prototipado rápido y actualmente es la más utilizada, crea piezas tridimensionales de plástico directamente a partir de la información proveniente de un sistema asistido por computadora (CAD) como se observa en la figura 1.1



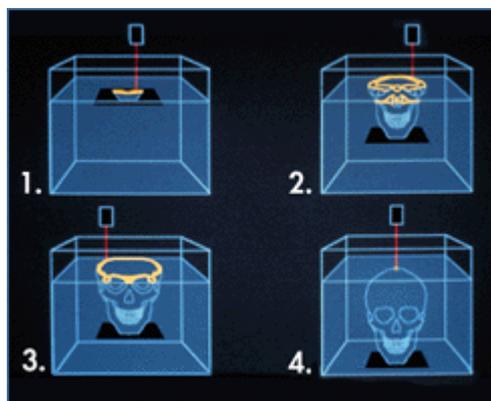
**Figura. 1.1. Impresora 3D de fabricación de modelos por estereolitografía**  
Fuente: [www.milparts.net/sla.html](http://www.milparts.net/sla.html)

El sistema SLA construye modelos físicos capa a capa. La información que describe la geometría de la pieza es cortada matemáticamente en finas capas transversales. Luego, un rayo Láser Ultra Violeta (UV) recorre

sucesivamente estas secciones dibujándolas sobre la superficie de una batea que contiene resina fotosensible (figura 1.2 y figura 1.3).<sup>2</sup>



**Figura. 1.2. Esquema Estereolitografía**  
Fuente: [www.me.psu.edu](http://www.me.psu.edu)

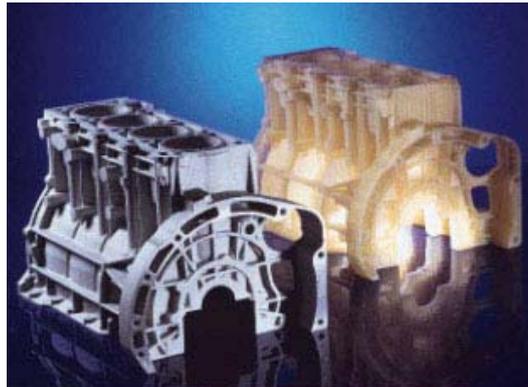


**Figura. 1.3. Proceso de fabricación por estereolitografía**  
Fuente: [www.biomodel.com](http://www.biomodel.com)

El plástico líquido se solidifica solamente en aquellas áreas que han sido iluminadas por el rayo láser. Luego una nueva capa de líquido cubre la última capa sólida y el rayo láser dibuja una nueva sección que solidifica y se adhiere a la anterior. Este proceso se repite hasta terminar la figura.

<sup>2</sup> DURSTELLES Juan, InfoVis.net , 2006, Alemania

Este no es un proceso particularmente rápido. Depende del tamaño y del número de objetos que serán creados; el láser puede tardar entre uno y dos minutos para cada capa. Un objeto (figura 1.4) típico toma entre seis y doce horas en ser terminado llegando a tomar días si se trata de un objeto muy grande. El tamaño del objeto depende de la resolución y capacidad de la impresora.



**Figura. 1.4. Muestra fabricada por estereolitografía**  
Fuente: [www.nvision3d.com](http://www.nvision3d.com)

Una vez que el objeto está terminado éste se debe llevar a un horno ultravioleta para curar el plástico (figura 1.5).<sup>3</sup>



**Figura. 1.5. Horno ultravioleta**  
Fuente: [www.rapidpro.com/files/whatis.htm](http://www.rapidpro.com/files/whatis.htm)

#### Tamaño máximo de las piezas:

- 19x19x25cm.
- 25,5 x 25,5 x 25,5 cm.

---

<sup>3</sup> [www.weissdesign.com/rp1](http://www.weissdesign.com/rp1)

- 51x51x60cm.

Materiales: fotopolímeros.

Ventajas: tecnología más madura; rápido; precisión del 2%.

Desventajas: los modelos son traslucidos; los modelos son quebradizos; algunas partes requieren soportes.<sup>4</sup>

### 1.1.2 Sinterización selectiva por láser (SLS)

En la tecnología de sinterización selectiva por láser (figura 1.6) se deposita una capa de polvo, de unas décimas de mm., en una cuba que se ha calentado a una temperatura ligeramente inferior al punto de fusión del polvo. Seguidamente un láser CO<sub>2</sub> sinteriza el polvo en los puntos seleccionados.



**Figura. 1.6. Maquina SLS**  
Fuente: [www.devicelink.com/emdm.html](http://www.devicelink.com/emdm.html)

Esta capa de polvo puede ser de múltiples materiales como poli carbonato, nylon, caucho ABS, metales y materiales cerámicos. Al utilizar polvo en lugar de líquido no es necesario crear estructuras de soporte.

Como se muestra en la figura 1.7 y figura 1.8 un haz del láser de CO<sub>2</sub> es reflejado mediante espejos sobre el polvo. Este láser realiza el aporte de

---

<sup>4</sup> Puesta en marcha y aplicaciones de una máquina de desarrollo rápido de prototipos con sistema de moldeado por deposición de hilo fundido, Patricia Eugenia Contreras Aparicio, Departamento de Mecánica, Universidad Técnica Federico Santa María, Noviembre 2001

energía necesaria para fundir las partículas de polvo logrando que éstas se unan las unas a las otras.<sup>4</sup>

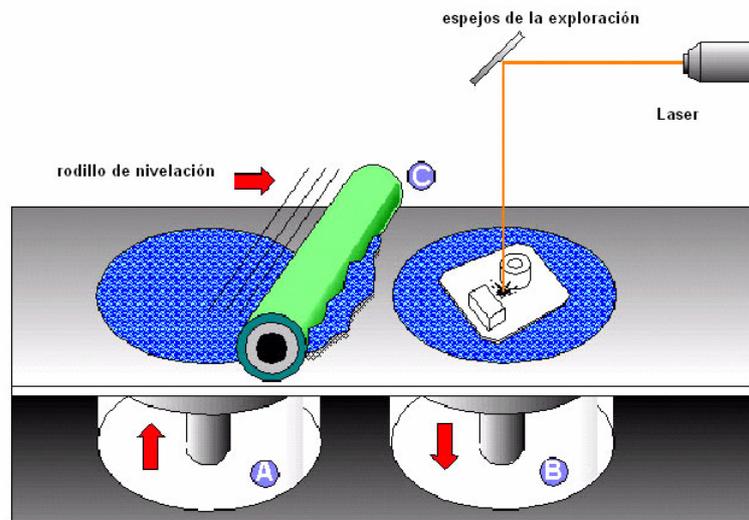


Figura. 1.7. Esquema SLS

Fuente: [www.me.psu.edu/lamancusa/rapidpro/primer/chapter2.htm](http://www.me.psu.edu/lamancusa/rapidpro/primer/chapter2.htm)

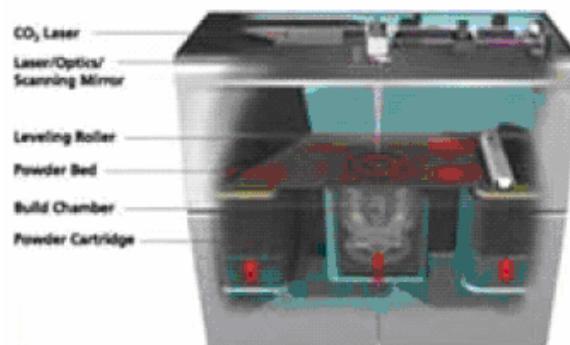
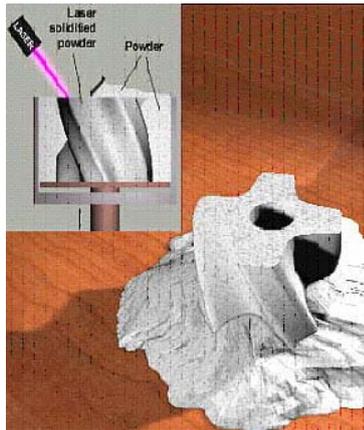


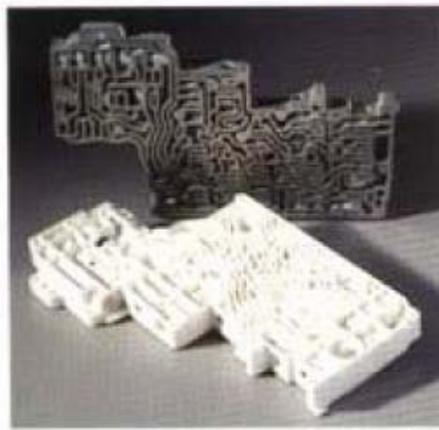
Figura. 1.8. Esquema SLS

El aporte de nuevas capas se efectúa mediante un proceso de alimentación que se realiza a través de un pistón que expulsa polvo a la superficie y un rodillo que se encarga de extender esta nueva capa de polvo de manera uniforme para ser sintetizado.

Este proceso se repite una y otra vez hasta estar la figura terminada (figura 1.9 y figura 1.10).



**Figura. 1.9. Muestra (1) SLS**  
Fuente: [www.nib.unicamp.br](http://www.nib.unicamp.br)



**Figura. 1.10. Muestra (2) SLS**

Tamaño máximo de las piezas: Diámetro 30 a 38 cm.

Materiales: PVC, termoplásticos, policarbonatos, metal, nylon, ceras y arena.

Ventajas:

- Material más barato. No huele.
- Más variedad de materiales que tienen mejores propiedades mecánicas que los fotopolímeros.
- Mayor precisión
- Tiene mejores propiedades mecánicas debido a la forma de polimerización.

Desventajas:

- El proceso es nuevo y por lo tanto menos extendido
- Salvo el PVC, las piezas tienen un aspecto laminado.
- Es el equipo más caro.
- Complejidad.
- Tamaño del equipo (pesa 5 toneladas), los modelos son traslucidos, los modelos son quebradizos.

### 1.1.3 Moldeado por deposición de hilo fundido (FDM)

En la tecnología de deposición de hilo fundido una boquilla que se mueve en el plano XY horizontal deposita un hilo de material a  $1^{\circ}\text{C}$  por debajo de su punto de fusión. Este hilo se solidifica inmediatamente sobre la capa precedente (figura 1.11).

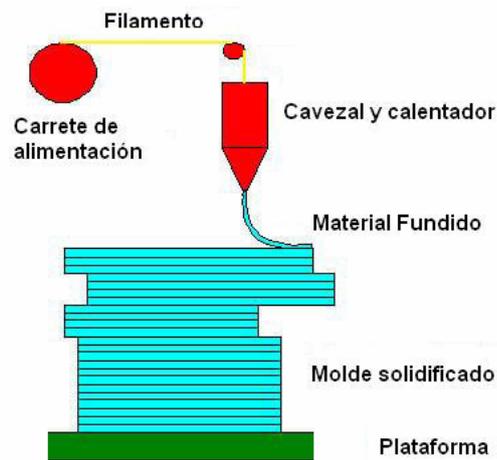


**Figura. 1.11. Maquina FDM**

Fuente: [http://mtc.engr.mun.ca/gallery\\_images/Fdm.jpg](http://mtc.engr.mun.ca/gallery_images/Fdm.jpg)

El filamento posee un diámetro de 1.587mm aproximadamente que se desenrolla de un carrete que alimenta un cabezal, éste último calienta el filamento hasta fundirlo y controla la salida de este filamento sobre la superficie en construcción.

El cabezal se encuentra sobre una pieza metálica que se mueve en el plano XY dando forma al modelo a fabricar, el filamento una vez depositado en la estructura se tarda menos de 0.1 segundos en enfriarse, permitiendo así que se pueda depositar rápidamente la siguiente capa (figura 1.12).



**Figura. 1.12. Esquema FDM**  
Fuente: [www.maxfac.com/stereosc.htm](http://www.maxfac.com/stereosc.htm)

En general los materiales utilizados pueden ser variados, incluso nylon y polímeros que pueden ser mecanizables. La reciente introducción de plásticos ha llevado a una mayor aceptación de este proceso ya que es menos frágil al manipular el objeto, en general, los materiales son de bajo costo (figura 1.13 y figura 1.14).<sup>5</sup>



**Figura. 1.13. Muestra (1) SLS**  
Fuente: [cybercut.berkeley.edu/mas2/html/processes/FDM](http://cybercut.berkeley.edu/mas2/html/processes/FDM)



**Figura. 1.14. Muestra (2) SLS**

Tamaño máximo de las piezas: 30,5 x 30,5 x 30,5 cm.

Materiales: Ceras, ABS y plástico tipo nylon.

Ventajas:

- Material más barato.
- No produce olores.
- No hacen falta soportes.

<sup>5</sup> [www.stratasys.com](http://www.stratasys.com)

- Más rápido que SLA.
- Obtener rápidamente un molde y posteriormente un prototipo.

#### Desventajas:

- Apariencia granulada

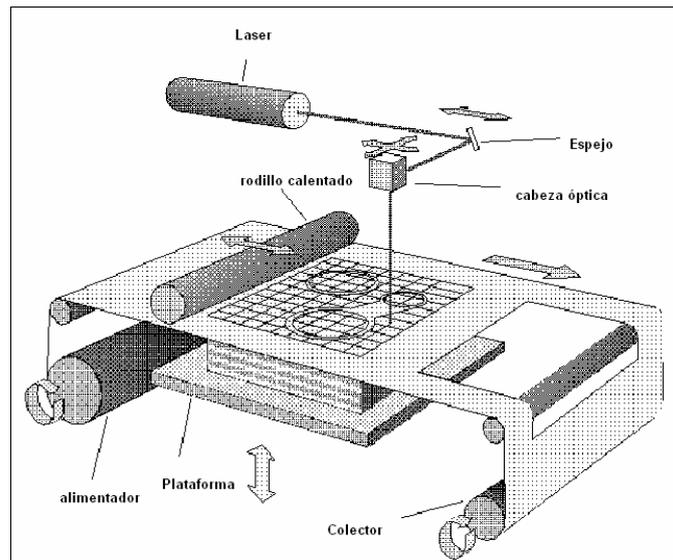
### 1.1.4 Fabricación por corte y laminado (LOM)

En la fabricación por corte y laminado una hoja de papel encolado se posiciona automáticamente sobre una plataforma y se prensa con un rodillo caliente que la adhiere a la hoja precedente (figura 1.15).



**Figura. 1.15. Maquina LOM**  
**Fuente: [mtc.engr.mun.ca/prod.html](http://mtc.engr.mun.ca/prod.html)**

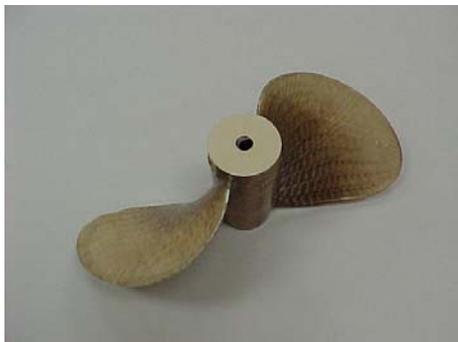
El material consta de papel laminado, nylon y celulosa, materiales que son relativamente baratos, la alimentación de esta tecnología consta de un rollo calentado a una temperatura predeterminada (figura 1.16).



**Figura. 1.16. Esquema LOM**

Fuente: [www.me.psu.edu/lamancusa/rapidpro/primer/lom.gif](http://www.me.psu.edu/lamancusa/rapidpro/primer/lom.gif)

Un láser, que se desplaza en el plano, recorta el papel dándole la forma de la capa correspondiente. Esta secuencia se repite hasta formar el objeto (figura 1.17 y figura 1.18).<sup>6</sup>



**Figura. 1.17. Muestra (1) SLS**  
Fuente: [www.tekniikka.oamk.fi](http://www.tekniikka.oamk.fi)



**Figura. 1.18. Muestra (2) SLS**  
Fuente: [www.helisys.com](http://www.helisys.com)

Tamaño máximo de las piezas: 25,4 x 33 x 38 cm.

Materiales: Papel, poliéster y nylon-celulosa.

Ventajas: Precisión hasta 0,1%

- Más rápido que otros métodos, porque el láser sólo tiene que trazar los contornos de la capa
- Barato.

<sup>6</sup> [www.helisys.com](http://www.helisys.com)

- No huele

**Desventajas:**

- Deformación de las láminas por los esfuerzos que puede producir el calor

**1.1.5 Fabricación por inyección (Ink Jet Printing)**

En la tecnología de inyección se deposita una capa de cera (material generalmente usado), de unos mm. o cm., sobre una plataforma plana; la temperatura a la que debe estar la cera almacenada y al momento de depositarla, es importante ya que esta debe estar en estado líquido pero sin llegar a punto de ebullición, con todo esto se va formando las capas siguiendo los puntos de la figura prediseñada del computador. Esta secuencia se repite hasta formar el objeto.



**Figura. 1.19. Maquina de inyección (1)**  
Fuente: [www.tekniikka.oamk.fi](http://www.tekniikka.oamk.fi)



**Figura. 1.20. Maquina de inyección (2)**  
Fuente: [www.3dsystems.com](http://www.3dsystems.com)

En la figura 1.21 se observa en detalle el esquema de la tecnología de inyección.<sup>7</sup>

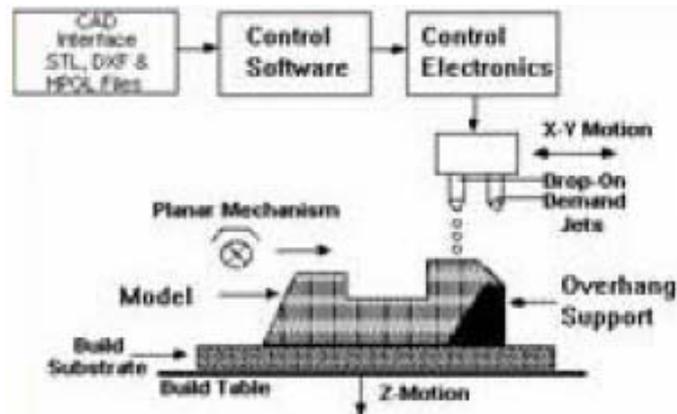


Figura. 1.21. Esquema de Ink Jet Printing

En la siguiente figura 1.22 se muestra la variedad de modelos que se pueden construir con este tipo de máquinas.



Figura. 1.22. Muestras de Ink Jet Printing con cera

Tiene algunas ventajas: la cera como material es barato, no produce olores ni ruido.

<sup>7</sup> [www.3dsystems.com](http://www.3dsystems.com)

### 1.1.6 Estratoconception

Es el proceso que permite construir un modelo por capas con la ayuda de fresadoras CNC. (Figura 1.23)

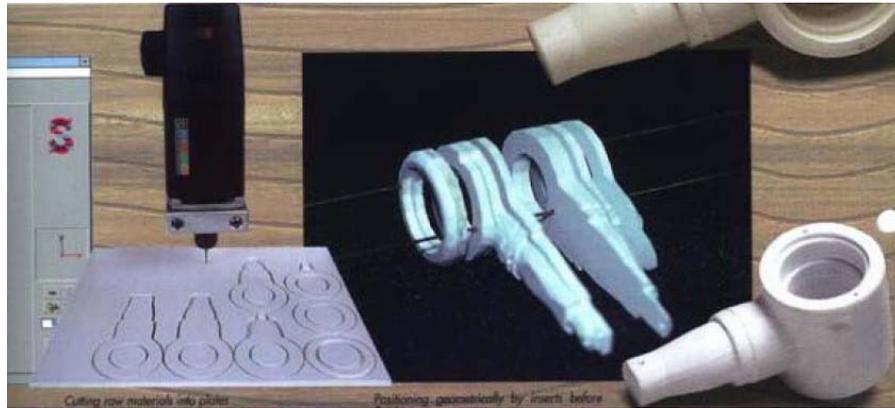


Figura. 1.23. Estratoconception

En la actualidad aplicaciones importantes de estos sistemas surgen en la **biomédica**, que son utilizados para la construcción de modelos de los órganos a operar, obteniendo información verdaderamente interactiva de un paciente concreto.

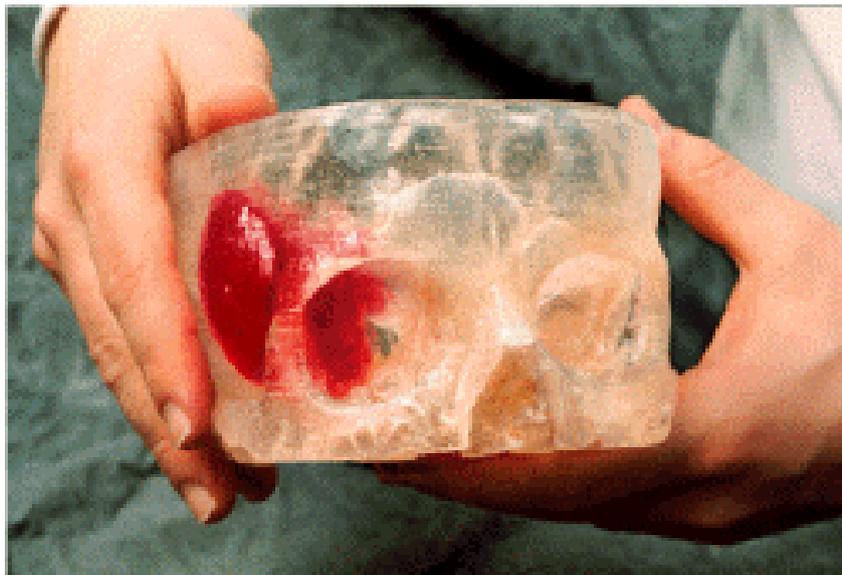


Figura. 1.24. Modelo de calavera humana afectada de una patología (en rojo)

Fuente: [www.biomodel.com](http://www.biomodel.com)

La visualización multisensorial (visual, táctil) del campo operatorio constituye una fuente de información y de comprensión de la realidad que proporciona una gran ayuda en la preparación de intervenciones.

Las impresoras 3D son una forma muy sofisticada de visualización, ya que permiten ver, tocar e incluso comprobar las propiedades de objetos virtuales.

## 1.2 Reconocimiento de los tipos de robots y controladores existentes

“Un robot es una máquina multifuncional, reprogramable, diseñada para manipular materiales, partes, herramientas, o dispositivos especiales, mediante movimientos variables, programados para la realización de una variedad de tareas”<sup>8</sup>

La Robótica es una tecnología que abarca:

- Automática.
- Mecánica.
- Electrónica.
- Informática
- Economía y Sociología.

Anteriormente, los robots eran demasiado grandes y costosos, además había que asociarles computadoras también muy caras. La aparición del Microprocesador redujo los precios.

---

<sup>8</sup> RIA - Robot Institute of America

### 1.2.1 Clasificación de la robótica desde el punto de vista Industrial

La diferencia fundamental de un robot y una máquina/herramienta autómatas estriba en que esta última está especializada en su trabajo, mientras que el robot es más versátil y puede utilizarse como parte fundamental de una línea de producción flexible.

Tipos de Robots:

- Robots Inteligentes: son manipuladores o sistemas mecánicos multifuncionales controlados por computadores capaces de relacionarse con su entorno a través de sensores y tomar decisiones en tiempo real. Concepto de “Inteligencia Artificial”.<sup>9</sup>
- Robots con control por Computador: similares a los anteriores pero carecen de la capacidad de relacionarse con el entorno que les rodea.
- Robots de Aprendizaje: se limitan a repetir una secuencia de movimientos realizada con la intervención de un operador y luego lo memorizan. También se denominan Robots de Macro.<sup>9</sup>
- Robots Manipuladores: son sistemas mecánicos multifuncionales cuyo sencillo sistema de control permite gobernar el movimiento de sus elementos de las formas siguientes:

---

<sup>9</sup>GONZALES Ana, ESPINA Mauricio, Robótica y Aplicaciones, 2006, México.

- Manual: el operador lo controla directamente.
  
- De Secuencia Variable: es posible alterar algunas de las características de los ciclos de trabajo.

### 1.2.2 Clasificación desde el punto de vista del control de sus movimientos

Tipos de Robots:

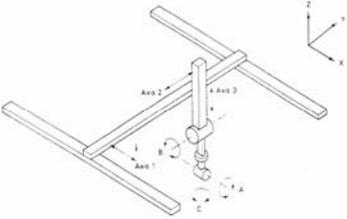
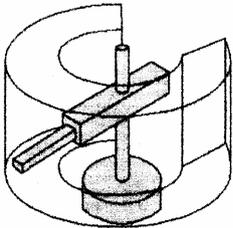
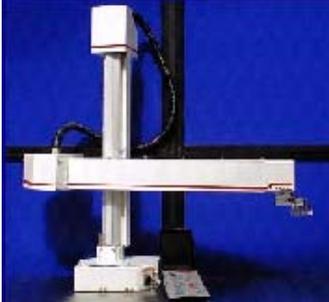
- Sin Servocontrol: el programa que controla el movimiento de los diferentes componentes del robot se realiza en un posicionamiento punto a punto en el espacio.<sup>9</sup>
  
- Con Servocontrol: este tipo de control permite dos formas de trabajo diferentes:
  - Gobierno de los movimientos de los elementos del robot en función de sus ejes. Los desplazamientos pueden realizarse punto a punto o con trayectoria continua.
  
  - Los movimientos se establecen en función de la posición respecto a los ejes de coordenadas (x, y, z) y de la orientación de la mano o herramienta del robot.

### 1.2.3 Clasificación desde el punto de vista de su geometría

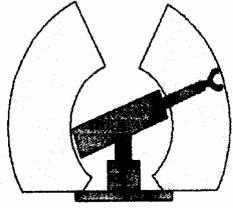
Se basan básicamente en la forma de su entorno de trabajo:

- Cartesianos Transversal
- Cilíndricos
- Esféricos
- Articulados verticales
- Articulados horizontales

Tabla 1.1: Tipos de robots por su geometría <sup>10</sup>

Esquema Básico	Ejemplo Real
 <p data-bbox="316 1473 703 1509"><b>Figura. 1.25. Robot cartesiano</b></p>	 <p data-bbox="879 1480 1310 1516"><b>Figura. 1.26. Robot cartesiano (R)</b></p>
 <p data-bbox="325 1839 697 1874"><b>Figura. 1.27. Robot cilíndrico</b></p>	 <p data-bbox="887 1854 1302 1890"><b>Figura. 1.28. Robot cilíndrico (R)</b></p>

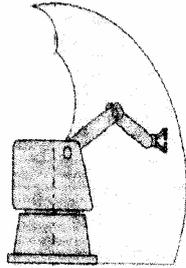
<sup>10</sup> ORTIZ Hugo, Presentación Robótica, 1998, Ecuador



**Figura. 1.29. Robot esférico**



**Figura. 1.30. Robot esférico (R)**



**Figura. 1.31. Robot art. vertical**



**Figura. 1.32. Robot art. Vertical (R)**

Los controladores más comunes y utilizados en la actualidad para robots son los siguientes:

- Por computadores
- Controlador Lógicos Programables (PLC's)
- Microprocesadores
- Microcontroladores
- Tarjeta de Adquisición de Datos

### 1.3 Descripción de software de dibujo 3D, lenguajes de programación

Software de dibujo 3D:

#### 1.3.1 AutoCAD

Es un programa de diseño asistido por ordenador (DAO o en inglés CAD - Computer Aided Design) para diseño 2D y 3D.

Programa, en su base, del tipo vectorial (aunque admite incorporar archivos bitmap), donde se dibuja figuras básicas (líneas, circunferencias, rectángulos, etc.), y junto con herramientas de edición, se crean gráficos más complejos. Puede organizar los objetos por medio de capas (layers) y bloques (blocks), a la vez permite plotear con distintos espesores de línea. La extensión del archivo de AutoCAD es .dwg, aunque permite exportar en otros formatos (el más conocido es el .dxf).<sup>11</sup>

El formato .dxf permitía compartir dibujos con otras plataformas de dibujo CAD, reservándose AutoCAD el formato .dwg para sí mismo. El formato .dxf puede editarse con un procesador de texto básico, por lo que se puede decir que es abierto. En cambio, el .dwg sólo podía ser editado con AutoCAD, si bien desde hace poco tiempo se ha liberado este formato, con lo que muchos programas CAD distintos del AutoCAD lo incorporan, y permiten abrir y guardar en esta extensión.

Las aplicaciones del programa son múltiples, desde proyectos y presentaciones de ingeniería, hasta diseño de planos para delineación y arquitectura.

---

<sup>11</sup> Manual de ayuda – Autocad 2006

### 1.3.2 RhinoCeros

Rhino es un software que proporciona herramientas para modelar proyectos con precisión y dejarlos listos para el renderizado, la animación, el dibujo, la ingeniería, el análisis y la fabricación. Este programa puede crear, editar, analizar y traducir curvas NURBS, superficies y sólidos en Windows. No hay límite de complejidad, grado o tamaño. Rhino también es compatible con mallas poligonales y nubes de puntos.<sup>12</sup>

#### Características:

- Precisión necesaria para el diseño, los prototipos, la ingeniería, el análisis y la fabricación de cualquier producto, desde aviación hasta joyería.
- Compatibilidad con otros diseños, dibujos, CAM, ingeniería, análisis, renderizado, animación y software de dibujo.
- Accesible. Fácil de aprender y de utilizar que se puede dedicar al diseño y la visualización sin tener que preocuparse por el software.
- Formatos de archivo que soporta: DWG/DXF (AutoCAD 2000, 14, 13, y 12 ), SAT (ACIS), X\_T (Parasolid), 3DS, LWO, STL, OBJ, AI, RIB, POV, UDO, VRML, BMP, TGA, JPG, CSV (propiedades de exportación e hidrostática), archivos TIFF sin comprimir, STEP, VDA, GHS, SLC, Deep Paint 3D.

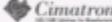
Existen muchísimos programas en el mercado de todo el mundo, aunque no son muy conocidos en el medio es importante conocer varias opciones.

---

<sup>12</sup> [www.rhinoceros.com](http://www.rhinoceros.com)

A continuación se enlista algunos de ellos en la tabla 1.2:

**Tabla 1.2: Software para CAD**  
Fuente: [www.directindustry.es](http://www.directindustry.es)

	programa CAD 2D - 3D TopSolid'Design	
	programa CAD 3D para vaciadores, moldeadores y operarios especializados TopSolid'Mold	
	programa CAD 2D - 3D	
	programa CAD, modelador 3D volúmico	
	programa CAD 2D - 3D MEDUSA4	
	programa CAD, modelador 3D volúmico	
	programa CAD 2D - 3D	
	programa CAD, modelador 3D volúmico	
	programa CAD 3D para vaciadores, moldeadores y operarios especializados WorkNC-CAD	
	programa CAD 3D para vaciadores, moldeadores y operarios especializados Cimatron E	
	programa CAD 3D para vaciadores, moldeadores y operarios especializados	
	programa CAD, modelador 3D volúmico	
	programa CAD 3D para vaciadores, moldeadores y operarios especializados	
	programa CAD 2D - 3D	

### 1.3.3 Lenguajes de programación para interfaces HMI

#### Visual C++

Es una herramienta de desarrollo de aplicaciones creada por Microsoft para programadores de C++. El Visual C++ soporta la programación orientada a objetos de aplicaciones Windows de 32-bit con el ambiente de desarrollo integrado (Integrated Development Environment: IDE), un compilador C/C++, y una biblioteca de clases llamada Microsoft Foundation Classes (MFC). El IDE incluye un AppWizard, ClassWizard, y funciones de testeo para facilitar la programación.

Visual C++ es un entorno integrado de desarrollo que permite la programación orientada a objetos (POO) conjuntamente con el sistema de desarrollo SDK (también denominado API) de Windows. Al ser un entorno integrado Visual C++ incluye, entre otras, las siguientes herramientas de desarrollo:

- Editor de texto
- Compilador/Enlazador
- Depurador
- Visor de datos y dependencias (Browser)

Pero si desde el punto de vista del usuario Windows es un sistema amigable, desde el punto de vista del desarrollador observaremos todo lo contrario.

#### Visual Basic

Es un lenguaje de programación desarrollado por Microsoft. Está basado en el lenguaje BASIC, y es uno de los primeros productos que proveyó un ambiente de programación gráfico para desarrollar interfaces de usuario. En lugar de preocuparse por los detalles de sintaxis, el programador de Visual

Basic puede agregar una cantidad sustancial de código simplemente arrastrando y soltando controles, como ser botones y cajas de diálogo, para luego definir su apariencia y comportamiento.

Aunque no es un auténtico lenguaje de programación orientado a objetos en su sentido estricto, el Visual Basic tiene una innegable filosofía orientada a objetos.

Desde su lanzamiento en 1990, el enfoque del Visual Basic se convirtió en una norma para los lenguajes de programación. El Visual Basic suele conocerse como sistema RAD (Rapid Application Development) porque permite que los programadores construyan rápidamente aplicaciones prototipo.<sup>13</sup>

### LabView

Es una herramienta gráfica de test, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G.

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y Linux y va por la versión 7. Los programas hechos con LabVIEW se llaman VI (Virtual Instrument), lo que da una idea de uno de sus principales usos: el control de instrumentos. El lema de LabVIEW es: "La potencia está en el Software". Esto no significa que la empresa haga únicamente software, sino que busca combinar este software con todo tipo de hardware, tanto propio -tarjetas de adquisición de datos, PAC, Visión, y otro Hardware- como de terceras empresas.<sup>14</sup>

---

<sup>13</sup> Visual Basic – [www.monografias.com](http://www.monografias.com)

<sup>14</sup> [www.ni.com](http://www.ni.com)

Es usado principalmente por ingenieros y científicos para tareas como:

- Adquisición de datos
- Control de instrumentos
- Automatización industrial o PAC (Controlador de Automatización Programable)
- Diseño de control: prototipaje rápido y hardware-en-el-bucle (HIL)

Su principal característica es la facilidad de uso, personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales.

También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. Para los amantes de lo complejo, con LabVIEW pueden crearse programas de miles de VIs (páginas de código) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación.<sup>15</sup>

Hoy en día existen una infinidad de lenguajes de programación y software que permiten la creación y diseño de interfaces HMI. A continuación se indica otras opciones que se conocen en el medio:

- Lookout
- InTouch
- X-factory

---

<sup>15</sup> [www.ni.com](http://www.ni.com)

## CAPÍTULO II

### DISEÑO DE LA ESTRUCTURA MECÁNICA DEL ROBOT CARTESIANO

#### 2.1 Diseño del robot cartesiano

Para cumplir con el objetivo de diseño e implementación de un prototipo de impresora 3D de modelos en cera, el tipo de robot que se adapta a los requerimientos es el CARTESIANO, puesto que se trabajará con modelos prediseñados en software, cuyo método de dibujo es por medio de coordenadas sea en dos o tres dimensiones con lo que se utilizarán los ejes (x, y, z).

El robot cartesiano se caracteriza básicamente por la forma de su entorno de trabajo cuya geometría genera una figura cúbica, donde se puede observar claramente sus movimientos lineales. (Figura 2.1)

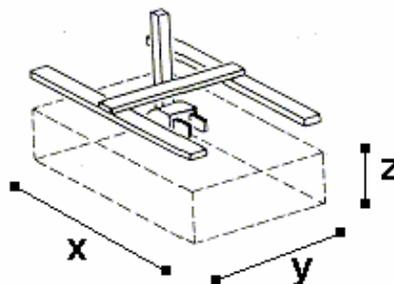
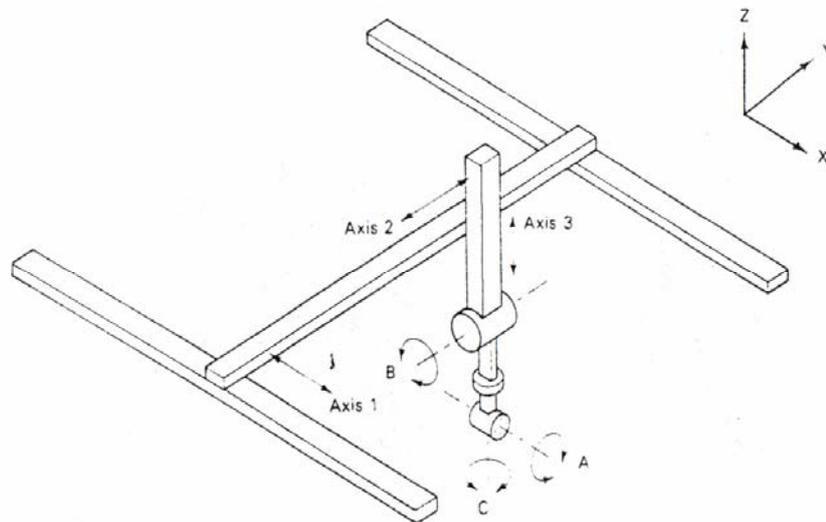


Figura. 2.1. Entorno de trabajo – Robot Cartesiano  
Fuente: [www.monografias.com](http://www.monografias.com)

El robot cartesiano posee tres movimientos lineales, es decir, tiene tres grados de libertad, los cuales corresponden a los movimientos localizados en los ejes X, Y y Z. (Figura 2.2).



**Figura. 2.2. Grados de Libertad – Robot Cartesiano**

Los movimientos que realiza este robot entre un punto y otro son con base en interpolaciones lineales. Interpolación, en este caso, significa el tipo de trayectoria que realiza el manipulador cuando se desplaza entre un punto y otro. A la trayectoria realizada en línea recta se le conoce como interpolación lineal y a la trayectoria hecha de acuerdo con el tipo de movimientos que tienen sus articulaciones se le llama interpolación por articulación.<sup>16</sup>

Los robots CARTESIANOS tienen una estructura morfológica muy sencilla. Sobresalen por sus altas prestaciones en cuanto a velocidad, para realizar tareas en entornos muy estructurados.

<sup>16</sup> GONZALES Ana, ESPINA Mauricio, Robótica y Aplicaciones, 2006, México

Sin embargo, sus aplicaciones prácticas, requieren que los entornos de trabajo no sean excesivamente complejos. Tanto la sencillez en su construcción como en el control son sus características más destacadas, siendo idóneos en determinadas aplicaciones como la alimentación de máquinas herramientas.<sup>17</sup>

Tomando en cuenta las partes básicas de la estructura y funcionamiento de un robot cartesiano el diseño se detalla a continuación:

- Entorno de trabajo

Por el tipo de robot cartesiano su entorno de trabajo es una figura regular, específicamente un cubo, que está descrito por los movimientos lineales en los tres ejes X, Y y Z. (Figura 2.1)

- Número de ejes y grados de libertad

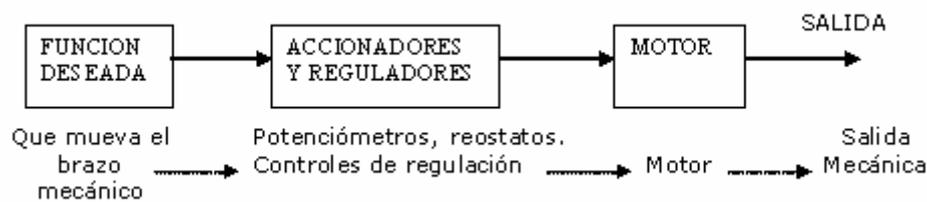
El robot cartesiano tiene tres movimientos lineales, es decir, posee tres grados de libertad, los cuales corresponden a los movimientos localizados en los ejes X, Y y Z. (Figura 2.2).

- Tipo de control

El tipo de control se basa en el sistema de lazo abierto. Conociendo que en un sistema de control de lazo abierto la salida no influye en la entrada, es decir, la salida no tiene efecto sobre la acción de control.

---

<sup>17</sup> GONZALES Ana, ESPINA Mauricio, Robótica y Aplicaciones, 2006, México.



**Figura. 2.3. Sistema de Lazo abierto**

Para el control del robot cartesiano, serán enviadas las ordenes desde el computador, según el modelo a imprimirse, luego llegarán a los microcontroladores (PIC 16F877) y éstos deberán ejecutar su programa para realizar el control de los motores, es decir, manejar la velocidad y sentido de giro.

- Lenguaje de programación

El lenguaje de programación que se utilizará en el controlador PIC será el Lenguaje C, ya que se posee conocimientos en éste y además existen compiladores y programadores de microcontroladores que permiten el uso de este lenguaje.

- Sistema de impulsión del robot

Se utilizará impulsión eléctrica, esto es cuando se usa la energía eléctrica para que el robot ejecute sus movimientos.

La impulsión eléctrica se utiliza para robots de tamaño mediano, pues éstos no requieren de tanta velocidad ni potencia como los robots diseñados para funcionar con impulsión hidráulica.

Los robots que usan la energía eléctrica se caracterizan por una mayor exactitud y repetibilidad.

- Modelo físico

El robot cartesiano tendrá una estructura similar a la de una mesa hueca, con dos bases para los soportes y el tornillo del eje X, una tercera base en la cual irá el segundo tornillo y guías para el eje Y; además este último soporte será el que transporte al dispensador de cera.

El sistema mecánico que se utilizará para los movimientos es el del tornillo sin fin, con acoplamiento entre los ejes de los motores y las barras roscadas (tornillos sin fin).

En cuanto al eje Z, se construirá una plataforma que subirá y bajará con el mismo sistema del tornillo sin fin.

A continuación en la figura 2.4 se presenta el esquema del modelo físico del robot cartesiano.

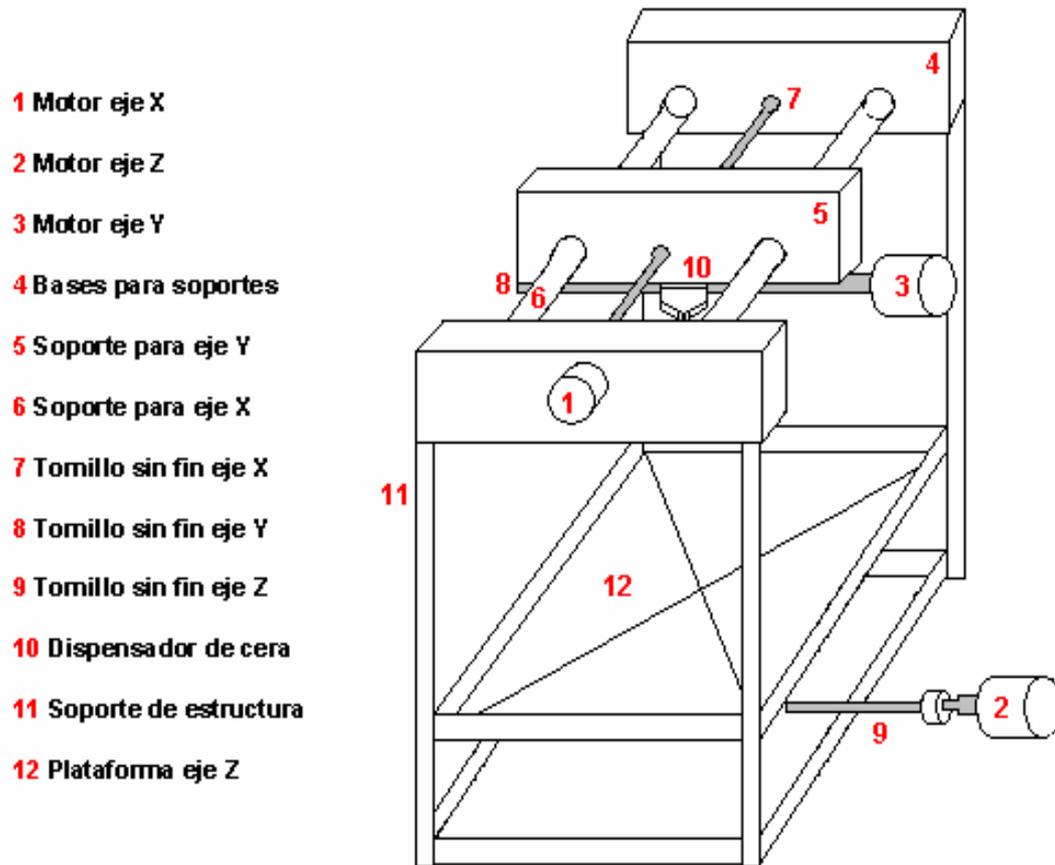


Figura. 2.4. Esquema Diseño Robot Cartesiano

## 2.2 Diseño parte mecánica del robot cartesiano

### 2.2.1 Fase 1

En esta fase se colocará: las bases (canaletas tipo C), soportes para el eje X, una base (tubo T) para el eje Y, y la primera barra roscada (tornillo) para el desplazamiento en el eje X. (Figuras: 2.5, 2.6 y 2.7 (\*))

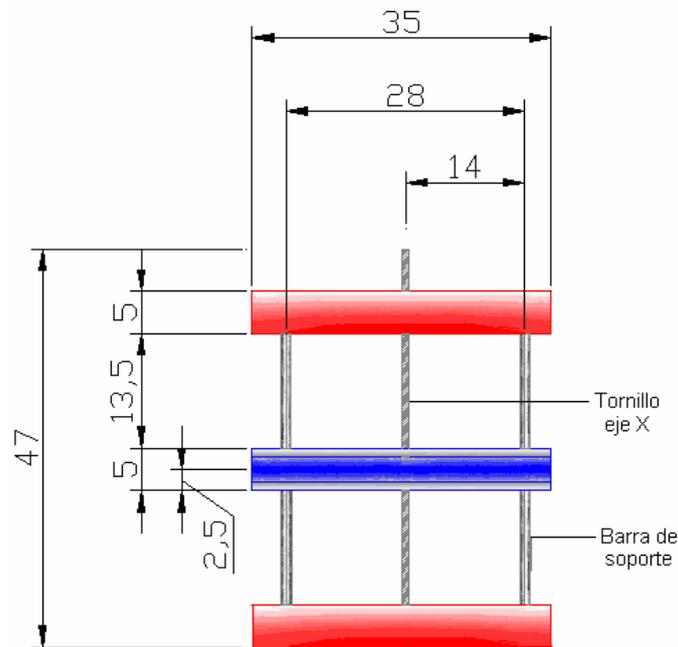


Figura. 2.5. Vista Superior- Fase 1

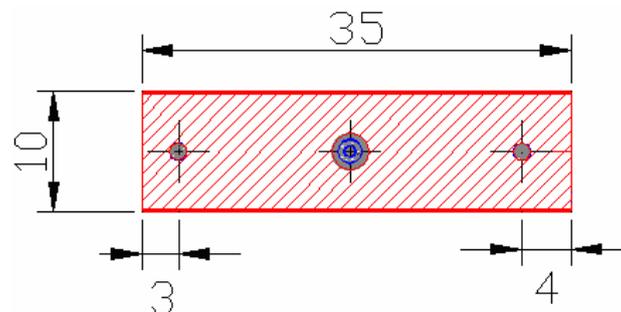


Figura. 2.6. Vista Frontal - Fase 1

\* La unidad de los planos es: centímetros [cm.]

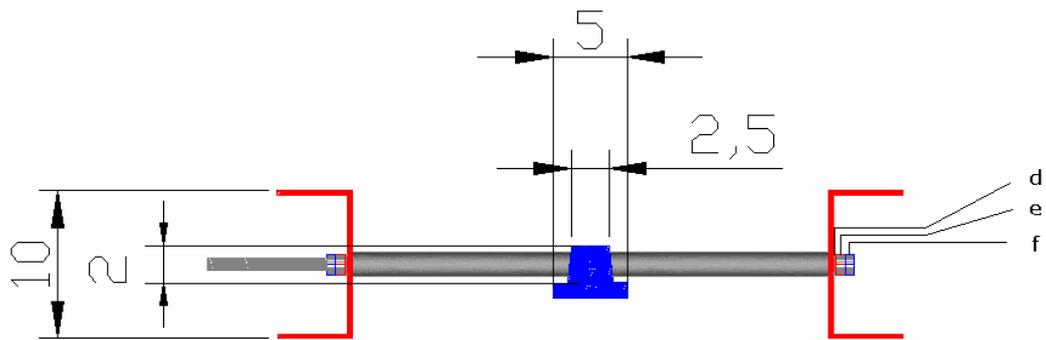


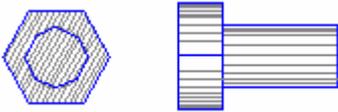
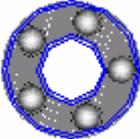
Figura. 2.7. Vista Lateral - Fase 1

Materiales:

Tabla. 2.1. Materiales - Fase 1

Material	Esquema (*)
Barra de transmisión (2)	<p>Figura. 2.8. Barra de transmisión</p>
Barra roscada (tornillo)	<p>Figura. 2.9. Barra roscada</p>
Canaleta Tipo C (2)	<p>Figura. 2.10. Canaleta C</p>
Tubo Tipo T	<p>Figura. 2.11. Tubo T</p>

\* La unidad de los planos es: centímetros [cm.]

Material	∅ Interno	∅ Externo	Largo	Grueso	Esquema (*)
Rodela plana (d)	0.8 cm.	1.4 cm.		0.2 cm.	 <p>Figura. 2.12. Rodela plana</p>
Tuerca (e)	0.8 cm.	1.4 cm.		0.6 cm.	 <p>Figura. 2.13. Tuerca</p>
Perno (f)	0.8 cm.	1.4 cm.	2.1 cm.		 <p>Figura. 2.14. Perno</p>
Rodamiento	0.9 cm.	2.0 cm.		0.4 cm.	 <p>Figura. 2.15. Rodamiento</p>

## 2.2.2 Fase 2

En esta fase se colocará: la segunda barra roscada para el movimiento en el eje Y, un bocín roscado para el primer tornillo y dos rodamientos lineales tipo bocín para el desplazamiento de la base (tubo T) sobre los soportes.

\* La unidad de los planos es: centímetros [cm.]

Figuras: 2.16, 2.17 y 2.18 (\*)

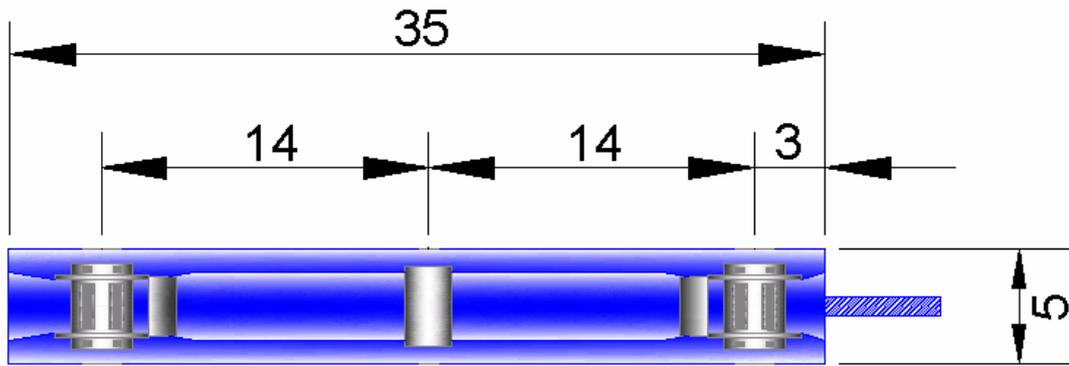


Figura. 2.16. Vista Superior -Tubo T con Tornillo eje Y

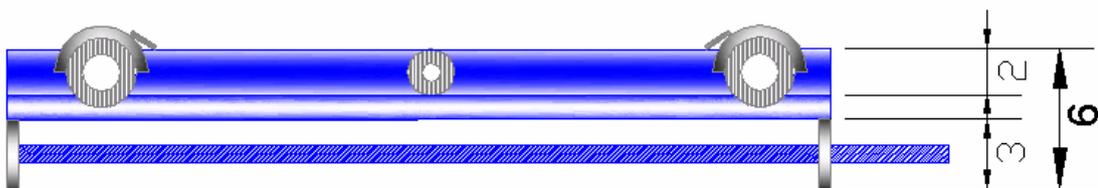


Figura. 2.17. Vista Frontal -Tubo T con Tornillo eje Y

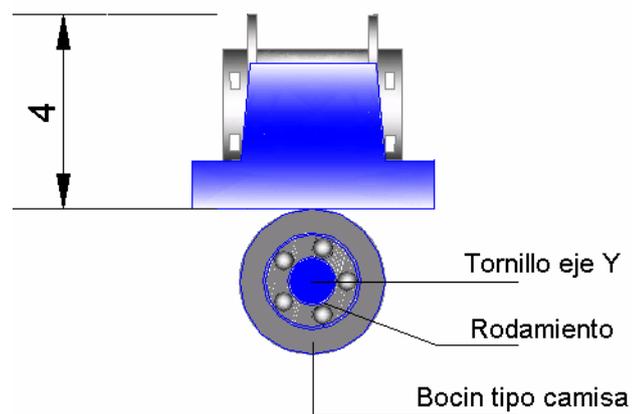
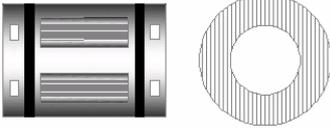
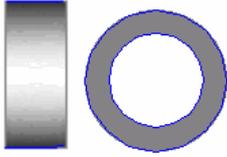
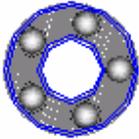
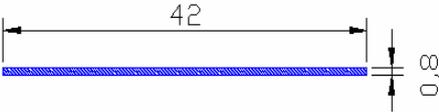


Figura. 2.18. Vista Lateral -Tubo T con Tornillo eje Y

\* La unidad de los planos es: centímetros [cm.]

Materiales:

**Tabla. 2.2. Materiales - Fase 2**

Material	∅ Interno	∅ Externo	Largo	Grueso	Esquema
<b>Bocin con roscado interno</b>	0.8 cm.	2.0 cm.	3.5 cm.		 <p><b>Figura. 2.19. Bocín</b></p>
<b>Rodamiento Lineal</b>	1.6 cm.	3.0 cm.	3.7 cm.		 <p><b>Figura. 2.20. Rodamiento lineal</b></p>
<b>Bocin tipo camisa</b>	2.0 cm.	3.0 cm.		0.5 cm.	 <p><b>Figura. 2.21. Bocín tipo camisa</b></p>
<b>Rodamiento</b>	0.9 cm.	2.0 cm.		0.4 cm.	 <p><b>Figura. 2.22. Rodamiento</b></p>
<b>Barra roscada (tornillo)</b>		0.8 cm.	35 cm.		 <p><b>Figura. 2.23. Barra roscada</b></p>

### 2.2.3 Fase 3

En esta fase se acoplará la fase 1 y la fase 2 con los respectivos cambios y ajustes mecánicos que necesite el robot, por ejemplo lubricante, limpieza, pulir, limar, etc. (Figuras 2.24, 2.25 y 2.26(\*))

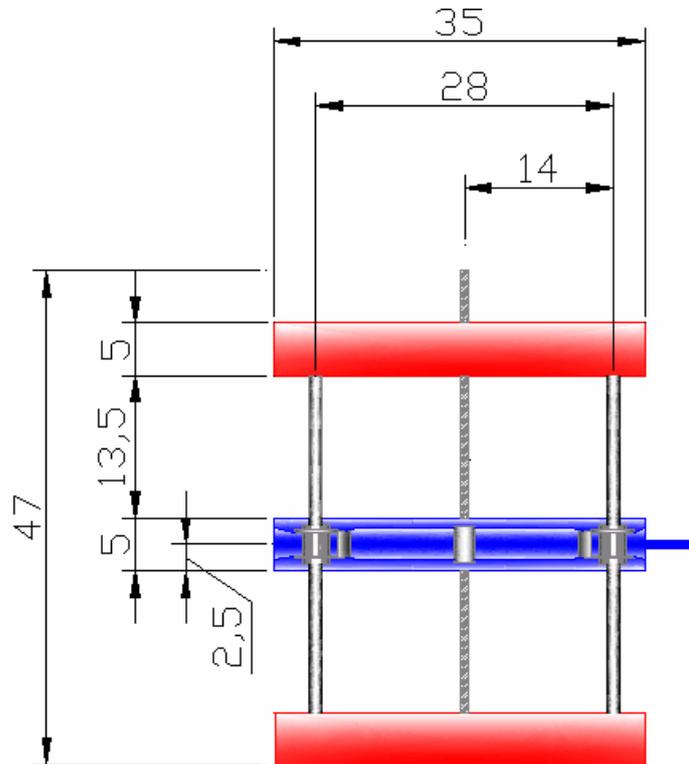


Figura. 2.24. Vista Superior- Fase 3

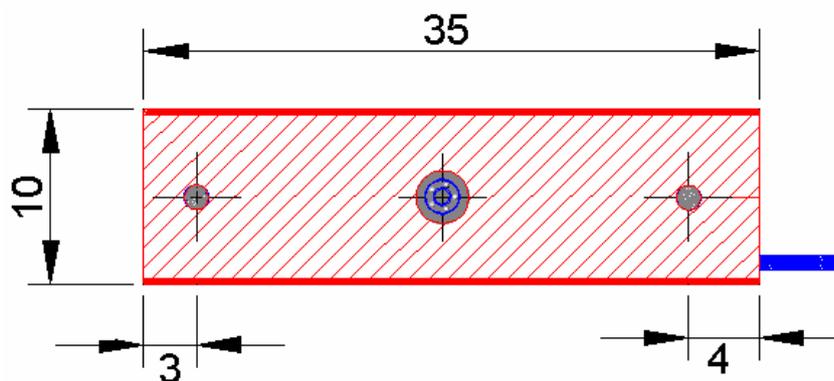


Figura. 2.25. Vista Frontal- Fase 3

\* La unidad de los planos es: centímetros [cm.]

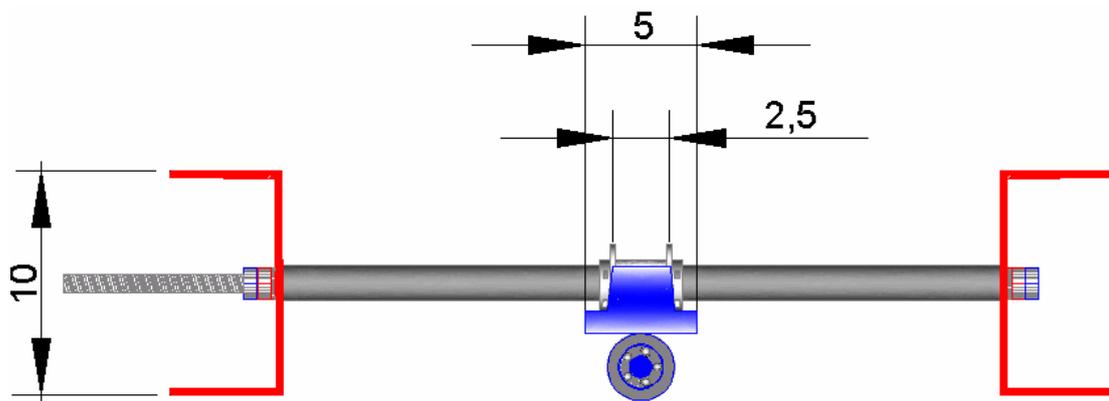


Figura. 2.26. Vista Lateral- Fase 3

#### 2.2.4 Fase 4

En esta fase se construirá la estructura que soportará al robot a una altura específica del suelo, cuyo espacio estará destinado para la plataforma y mecanismo para el movimiento en el eje Z. (Figura 2.27\*)

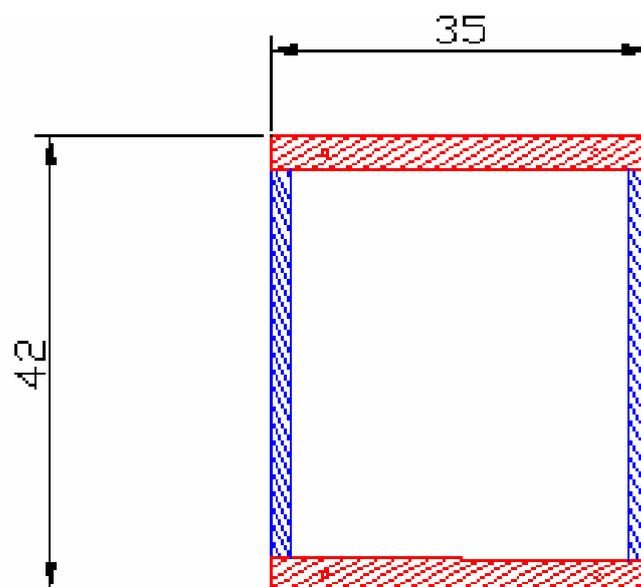


Figura. 2.27. Vista Superior – Estructura

\* La unidad de los planos es: centímetros [cm.]

Figuras 2.28 y 2.29 (\*)

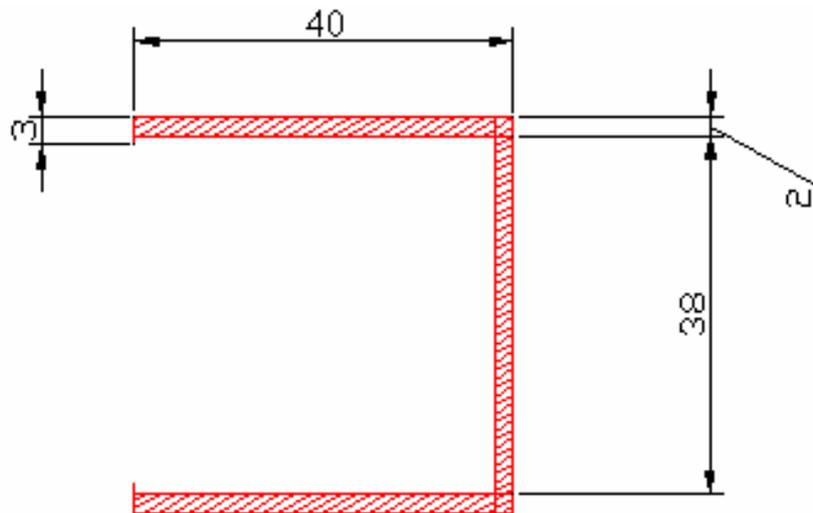


Figura. 2.28. Vista Lateral – Estructura

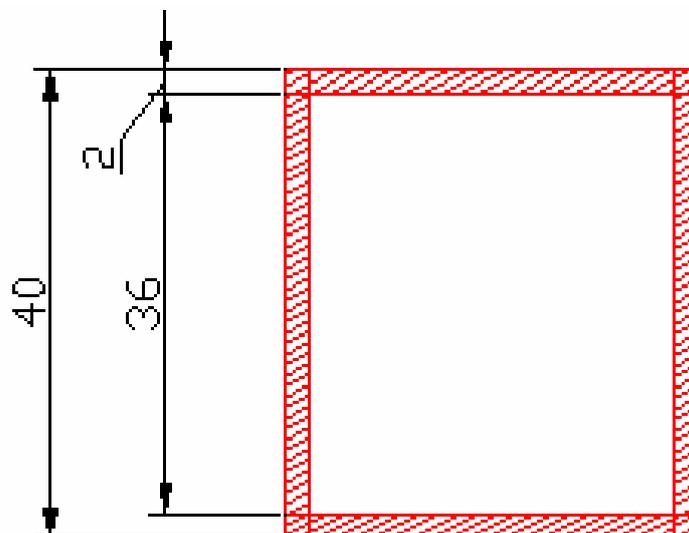
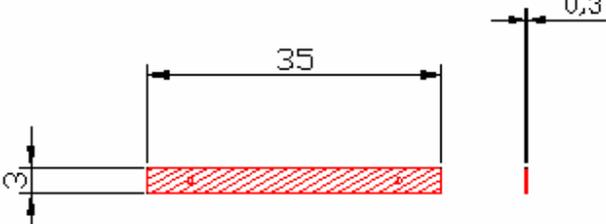
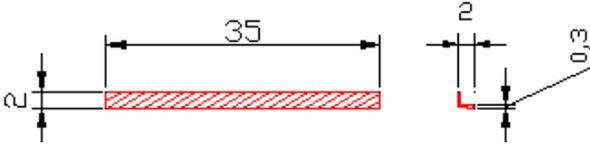
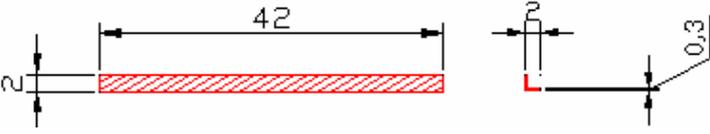
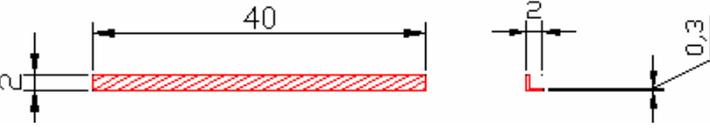


Figura. 2.29. Vista Frontal - Estructura

\* La unidad de los planos es: centímetros [cm.]

Materiales:

**Tabla. 2.3. Materiales Fase 4**

Material	Cantidad	Dimensiones (*)
Platinas de Soporte	2	 <p data-bbox="837 750 1093 784">Figura. 2.30. Platina</p>
Angulo	2	 <p data-bbox="813 1052 1117 1086">Figura. 2.31. Escuadra 1</p>
Angulo	2	 <p data-bbox="813 1344 1125 1377">Figura. 2.32. Escuadra 2</p>
Angulo	4	 <p data-bbox="813 1635 1125 1668">Figura. 2.33. Escuadra 3</p>

\* La unidad de los planos es: centímetros [cm.]

## 2.2.5 Fase 5

Acople de la fase 3 y fase 4. Figuras 2.34, 2.35 y 2.36 (\*)

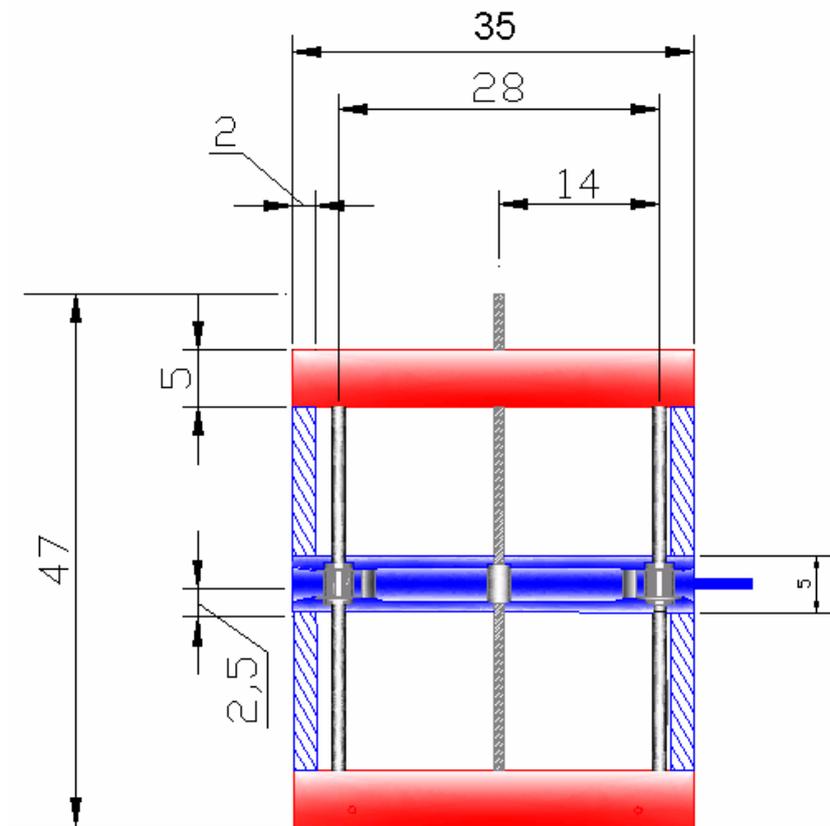


Figura. 2.34. Vista Superior – Estructura

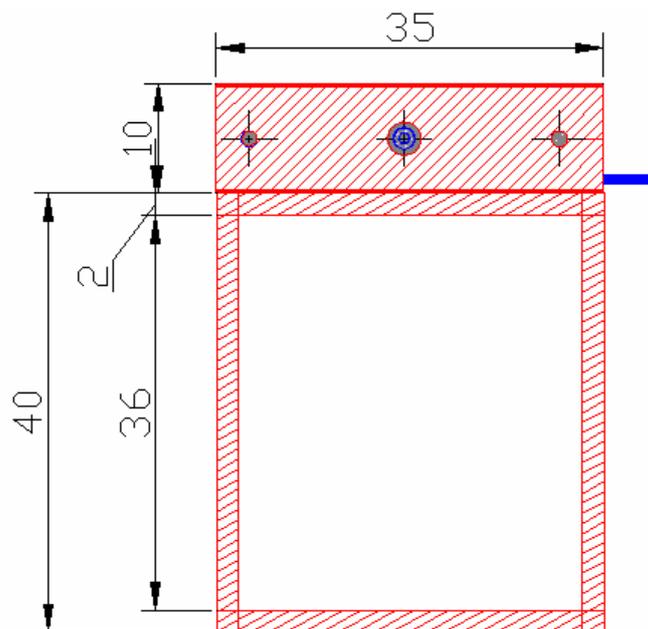


Figura. 2.35. Vista Frontal - Estructura

\* La unidad de los planos es: centímetros [cm.]

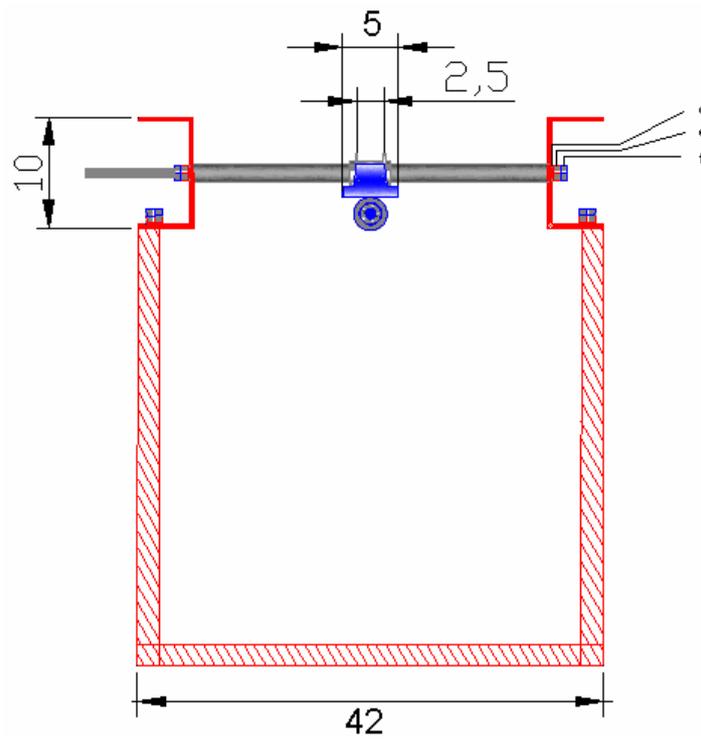


Figura. 2.36. Vista Lateral - Estructura

## 2.2.6 Fase 6

En esta fase se colocarán las guías y estructura que se moverá por el eje Y que a la vez transportará el dispensador de cera.

Figuras 2.37, 2.38 y 2.39 (\*)

\* La unidad de los planos es: centímetros [cm.]

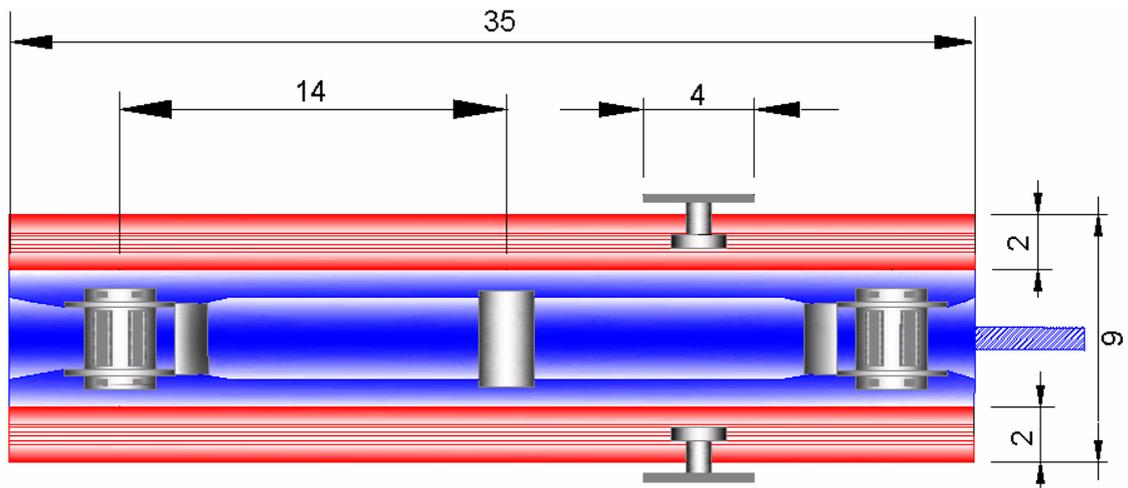


Figura. 2.37. Vista Superior – Fase 6

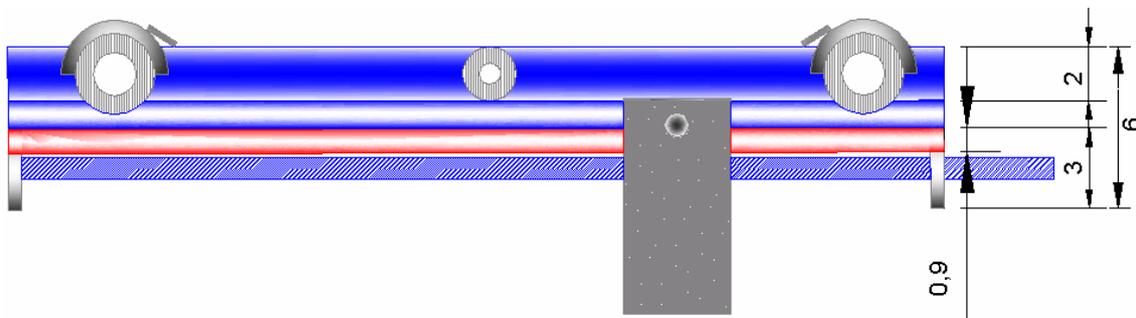


Figura. 2.38. Vista Frontal – Fase 6

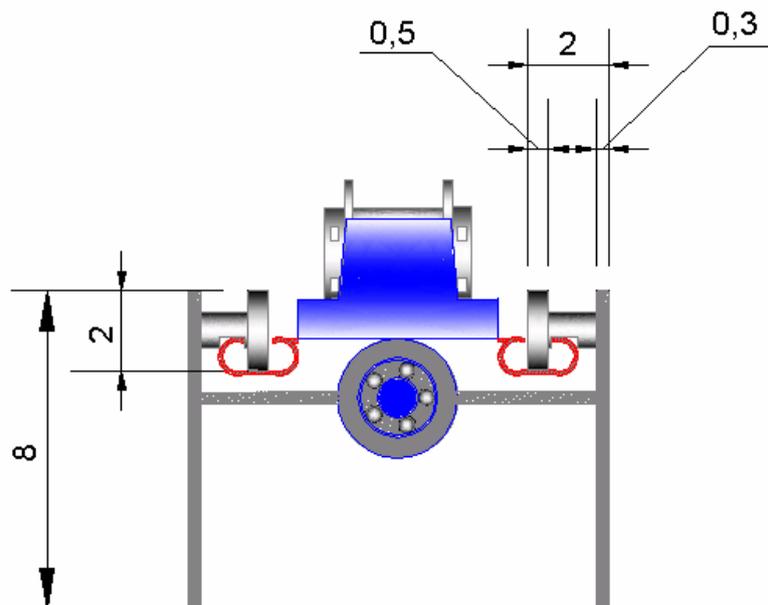


Figura. 2.39. Vista Lateral – Fase 6



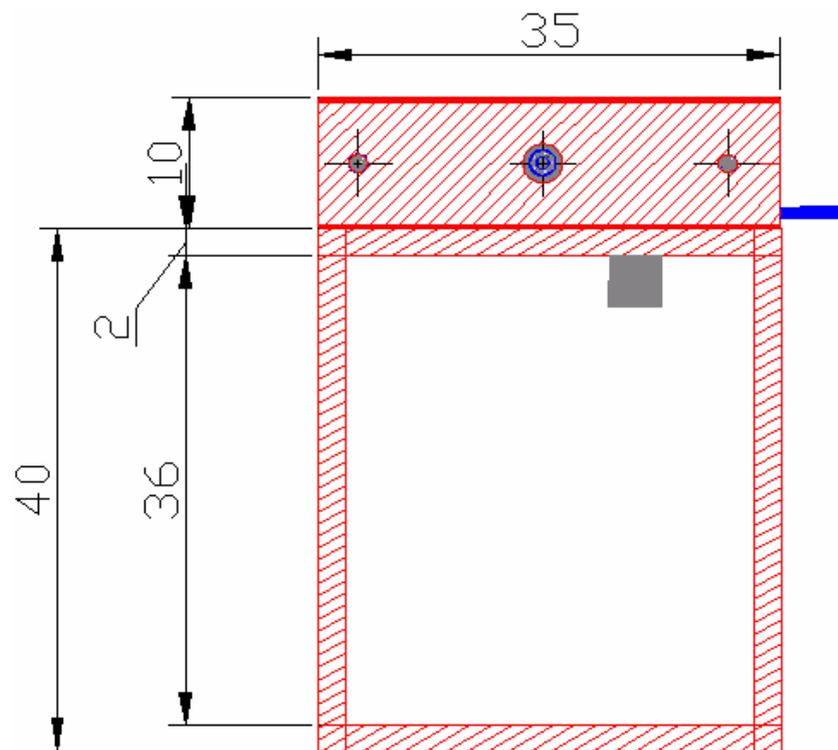


Figura. 2.41. Vista Frontal – Fase 7

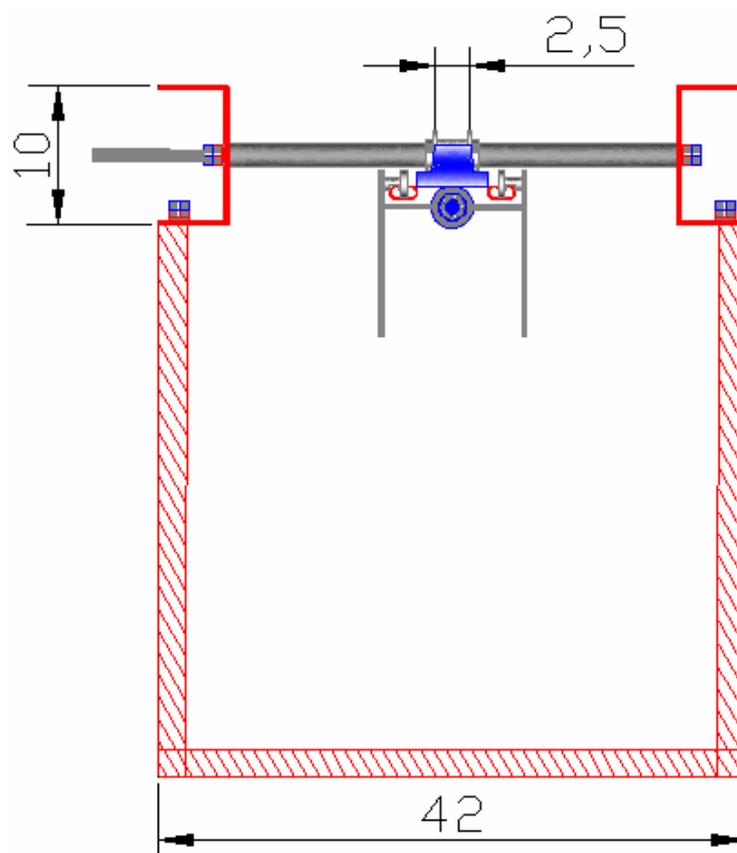


Figura. 2.42. Vista Lateral – Fase 7

### 2.2.8 Fase 8

En ésta fase se colocará: los tres motores X, Y y Z uno para cada eje respectivamente y se completará la construcción del eje Z con el mecanismo de los brazos de accionamiento de una gata mecánica, con los respectivos ajustes, así como la plataforma.

(Figuras 2.43, 2.44 y 2.45)(\*)

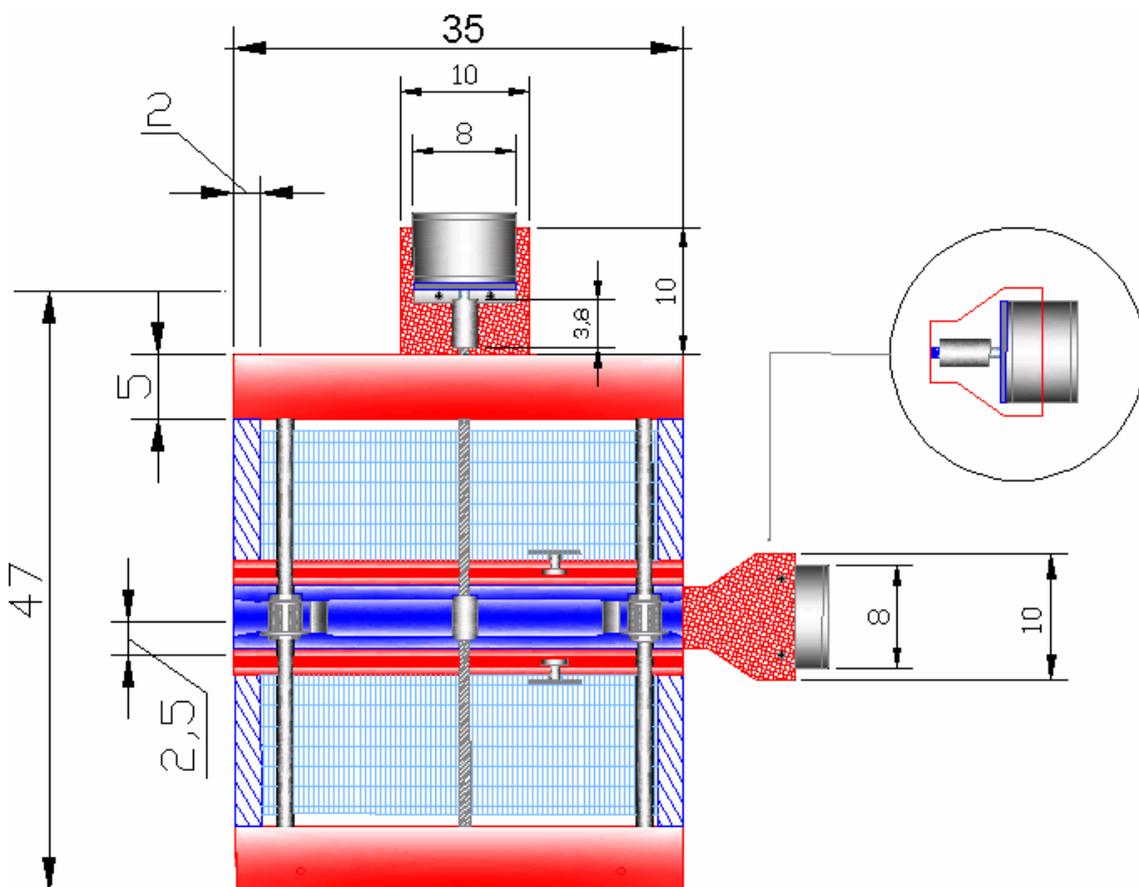


Figura. 2.43. Vista Superior – Fase 8

\* La unidad de los planos es: centímetros [cm.]

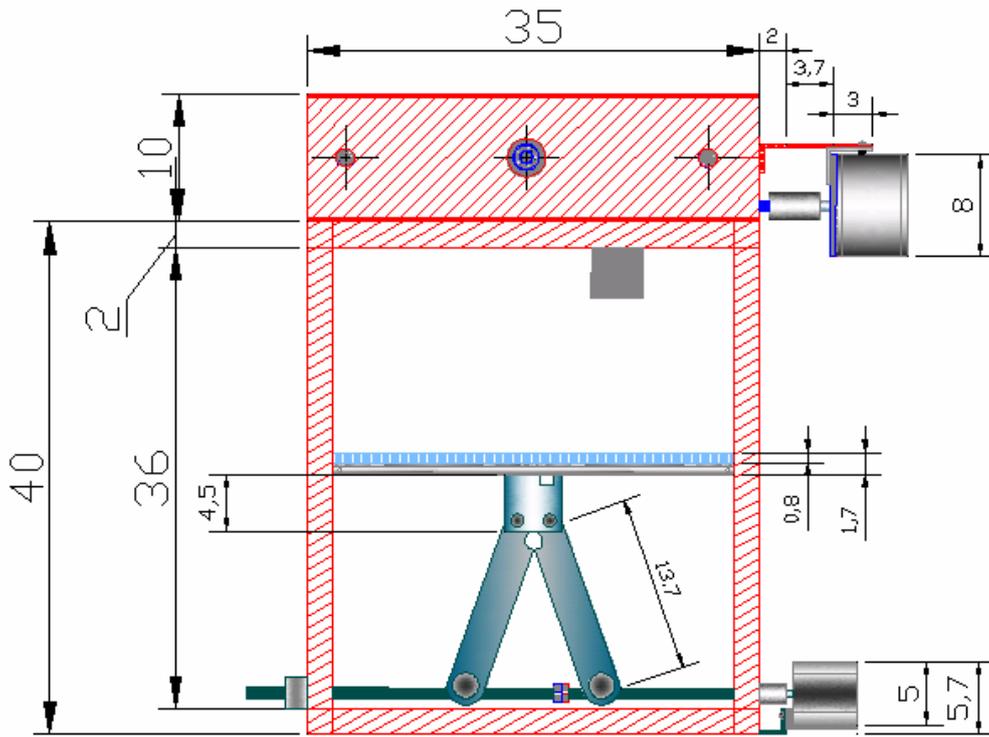


Figura. 2.44. Vista Frontal – Fase 8 (\*)

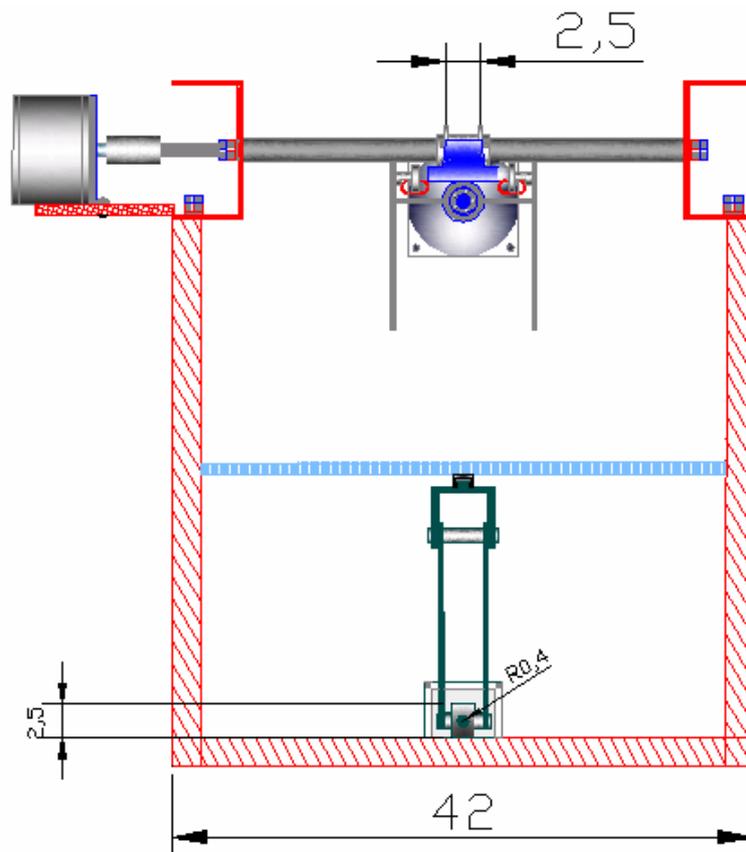


Figura. 2.45. Vista Lateral – Fase 8 (\*)

\* La unidad de los planos es: centímetros [cm.]

## 2.3 Diseño de la parte motriz del robot

### 2.3.1 Motores

“Un motor es un sistema basado en las leyes del electromagnetismo básicas, convierte la energía eléctrica en energía mecánica. Además es un sistema reversible.”<sup>18</sup>

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos; la característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Los motores P-P del tipo de imán permanente son los más usados en robótica.

En cuanto a motores se utilizará los de paso a paso, ya que estos nos permitirán controlar tanto el sentido de giro como su velocidad, y además se adaptarían a nuestras necesidades con respecto a la precisión. El sistema para el movimiento que se manejará, junto con los motores de paso, es el del tornillo sin fin, esto se aplicará para los tres ejes del robot cartesiano X, Y y Z.

En la tabla 2.4 se detallan las especificaciones de los motores a pasos que se utilizarán.

**Tabla. 2.4. Características de Motores a paso (Diseño)**

<b>Motor</b>	Motor a Pasos
<b>Tipo</b>	Unipolar
<b>Número de Bobinas</b>	4
<b>Grados por paso</b>	2.5°/paso
<b>Voltaje de alimentación</b>	4.5 – 5 [V] DC
<b>Corriente</b>	1 [A]

<sup>18</sup> <http://alek.pucp.edu.pe/cursos/pregrado/iee215/pasos>

En las figuras 2.46, 2.47 y 2.48 se observa las dimensiones físicas de los motores a utilizarse para los ejes X y Y. (\*)

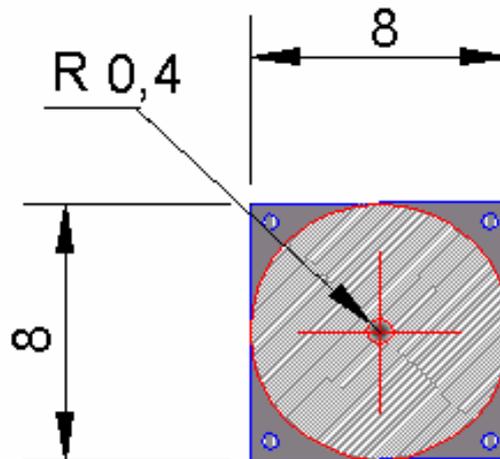


Figura. 2.46. Vista Frontal

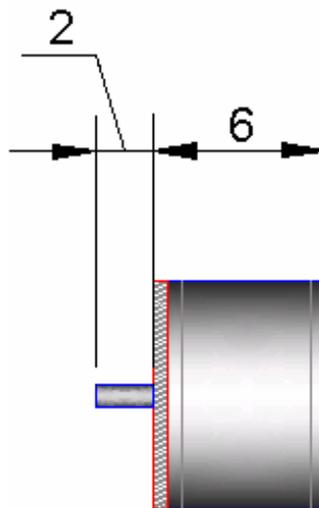


Figura. 2.47. Vista Lateral

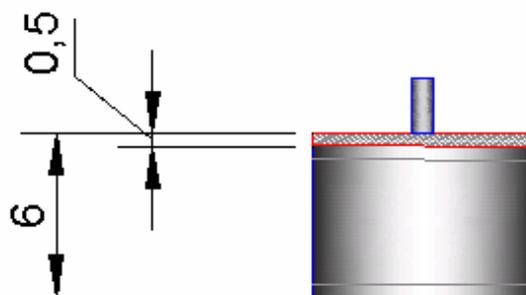


Figura. 2.48. Vista Superior

\* La unidad de los planos es: centímetros [cm.]

En las figuras 2.49, 2.50 y 2.51 se observa las dimensiones físicas del motor a utilizarse para el eje Z (\*)

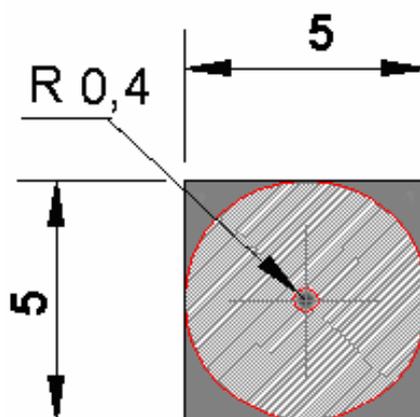


Figura. 2.49. Vista Frontal

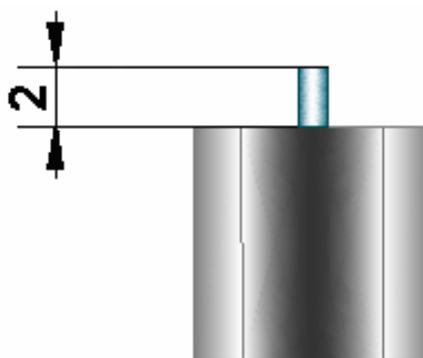


Figura. 2.50. Vista Frontal

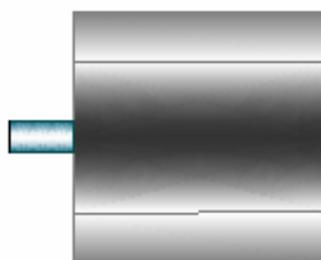


Figura. 2.51. Vista Frontal

\* La unidad de los planos es: centímetros [cm.]

### 2.3.2 Acoples y Soportes

El acople entre el eje del motor y la barra roscada (tornillo sin fin), será mediante bocines de acero roscados, en un extremo existirá un agujero con el diámetro del eje del motor y al otro un agujero con el diámetro del tornillo; para la sujeción se deberá utilizar prisioneros (tornillos sin cabeza). Figura 2.52 (\*).

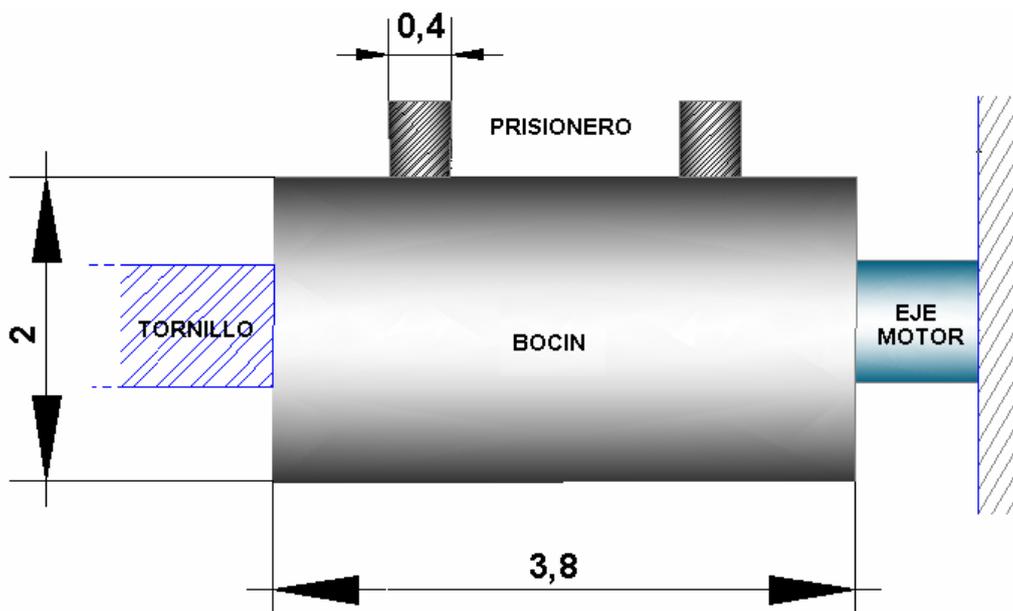


Figura. 2.52. Acople entre eje de motor y tornillo

Los soportes de los motores serán platinas de acero hechas a medida y sujetadas con pernos y tuercas. Se deberá tener un soporte para cada motor teniendo en cuenta las dimensiones, el peso y conexiones futuras.

\* La unidad de los planos es: centímetros [cm.]

Soportes: (Figuras 2.53, 2.54 y 2.55) (\*)

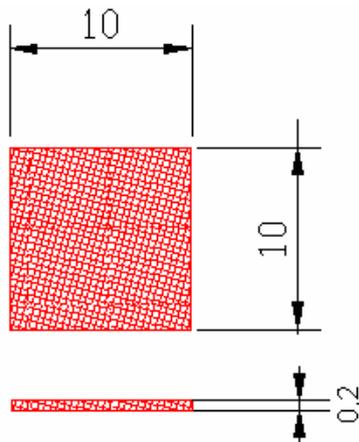


Figura. 2.53. Soporte de motor del eje X

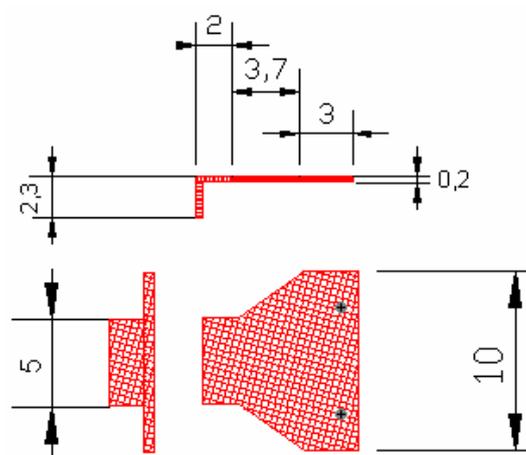


Figura. 2.54. Soporte de motor del eje Y

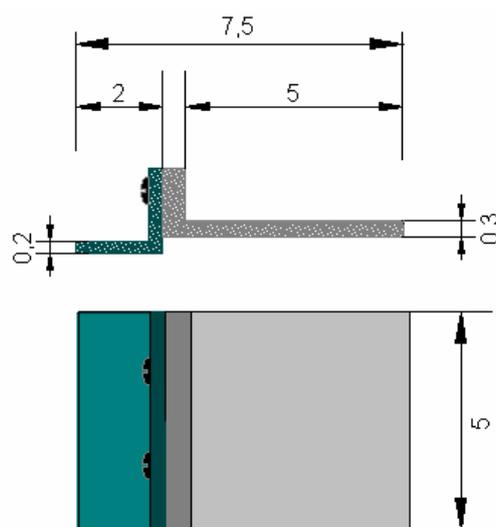
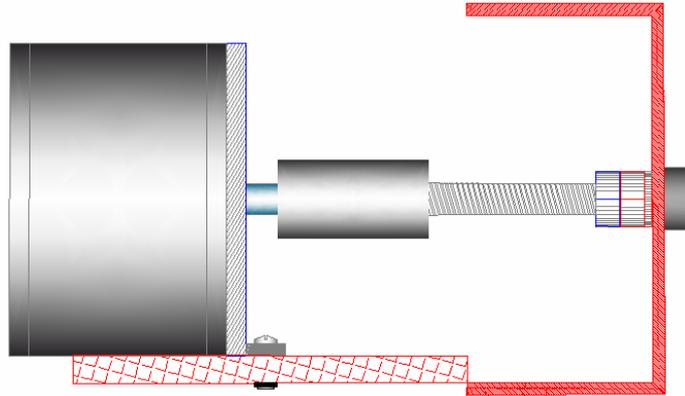


Figura. 2.55. Soporte de motor del eje Z

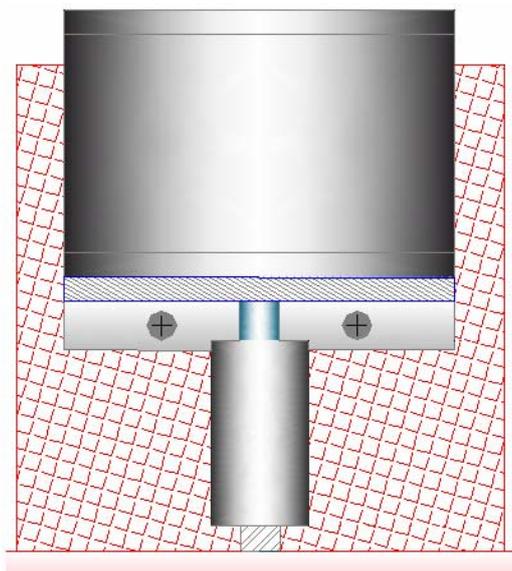
\* La unidad de los planos es: centímetros [cm.]

En las figuras siguientes se presenta como deberán estar montados los motores y acoples de cada eje.

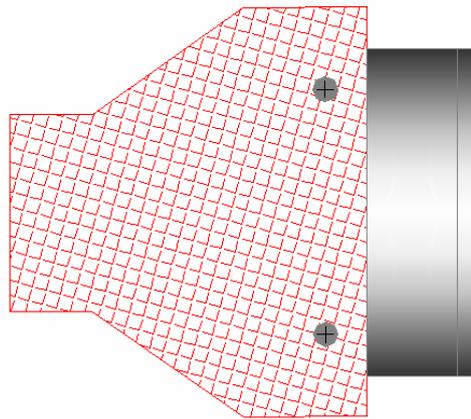
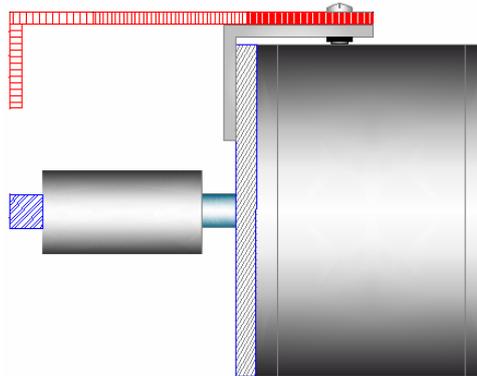
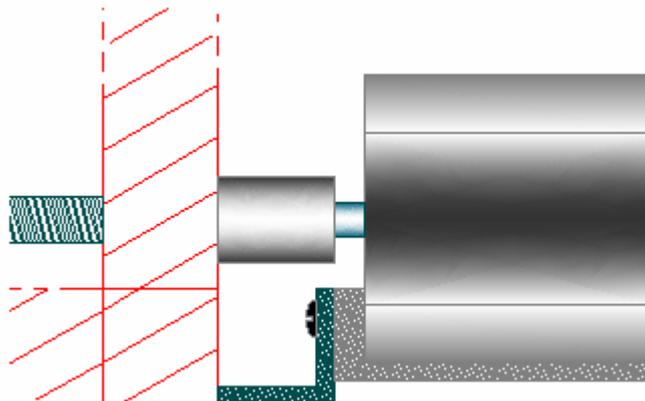
Motor Eje X:



**Figura. 2.56. Acoplamiento motor del eje X – Vista Lateral**



**Figura. 2.57. Acoplamiento motor del eje X – Vista Superior**

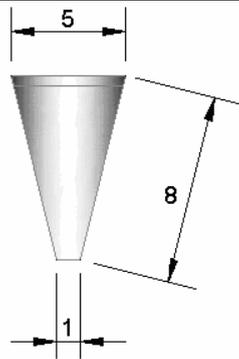
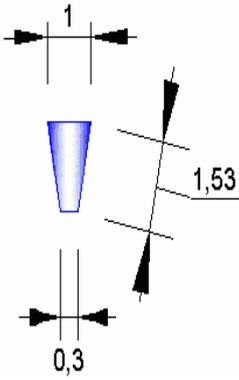
Motor Eje Y:**Figura. 2.58. Acoplamiento motor del eje Y – Vista Superior****Figura. 2.59. Acoplamiento motor del eje Y – Vista Lateral**Motor Eje Z:**Figura. 2.60. Acoplamiento motor del eje Z – Vista Lateral**

## 2.4 Diseño del sistema dispensador de cera

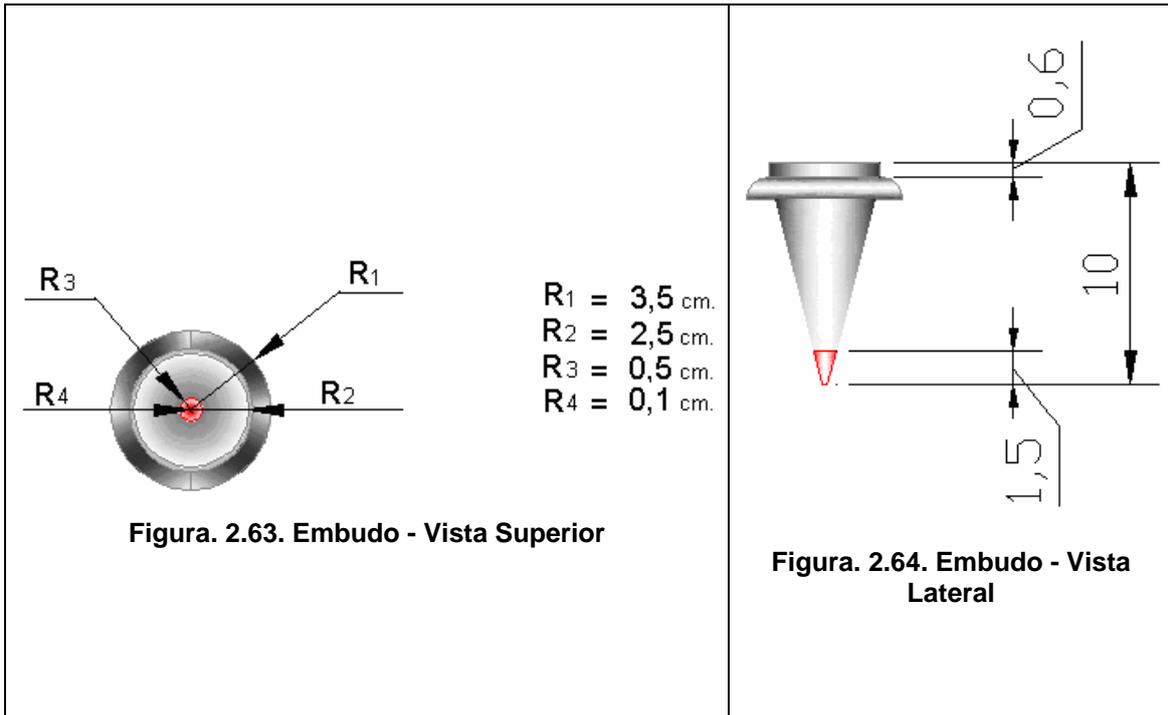
El dispensador será una de las partes principales y críticas de la impresora, es por esto, que de muchas opciones existentes para su construcción se ha diseñado de la siguiente manera:

Se utilizara un embudo de Aluminio para conseguir un calentamiento uniforme en su interior, su diámetro menor será de 3mm para tener un acabado fino.

Tabla. 2.5. Embudo para dispensador

Pieza	Material	Descripción	Esquema (*)
<b>Embudo</b>	Aluminio	Embudo: Altura: 8 cm. Diámetro superior: 5 cm. Diámetro Inferior: 1 cm.	 <p>Figura. 2.61. Embudo</p>
<b>Boquilla para embudo</b>	Aluminio	Diámetro superior: 1 cm. Diámetro Inferior: 0.3 cm.	 <p>Figura. 2.62. Boquilla</p>

\* La unidad de los planos es: centímetros [cm.]



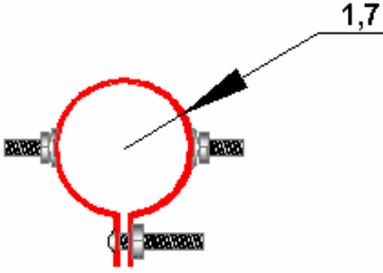
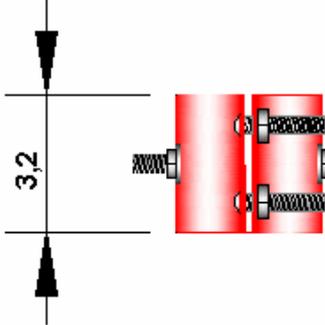
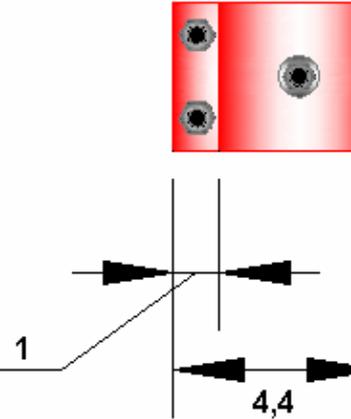
Se utilizara un recubrimiento de PVC para el soporte del dispensador.

**Tabla. 2.6. Recubrimiento del dispensador**

Pieza	Material	Descripción
Recubrimiento	PVC	Recubrimiento y soporte del dispensador.
<b>Esquema (*)</b>		
<p><b>Figura. 2.65. Vista Superior</b></p>	<p><b>Figura. 2.66. Vista Frontal</b></p>	<p><b>Figura. 2.67. Vista Lateral</b></p>

Se utilizara una resistencia de 150w para el calentamiento interno del embudo.

**Tabla. 2.7. Resistencia para calentamiento de Parafina**

Pieza	Material	Descripción
Resistencia	Aluminio	Resistencia para el calentamiento de la parafina (150 W)
<b>Esquema (*)</b>		
 <p data-bbox="247 1171 598 1205"><b>Figura. 2.68. Vista Superior</b></p>	 <p data-bbox="651 1223 976 1256"><b>Figura. 2.69. Vista Frontal</b></p>	 <p data-bbox="1029 1256 1348 1290"><b>Figura. 2.70. Vista Lateral</b></p>

El diseño final del dispensador se presenta en las figuras 2.71 a la 2.74.

\* La unidad de los planos es: centímetros [cm.]

Figuras 2.71, 2.72, 2.73 y 2.74 (\*)

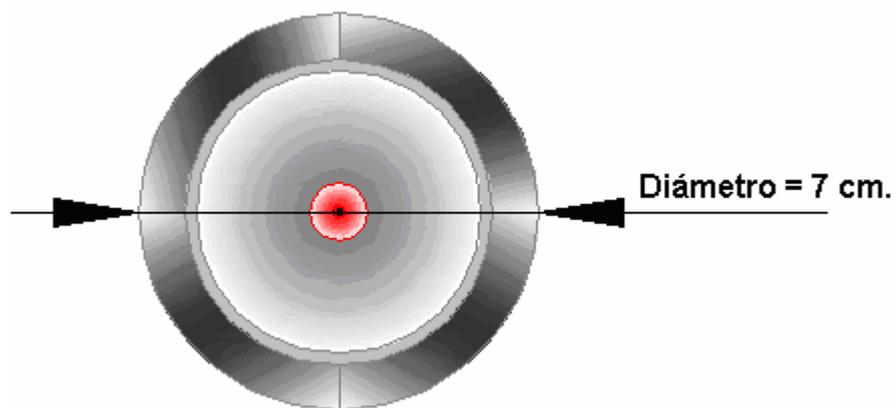


Figura. 2.71. Dispensador de cera – Vista Superior

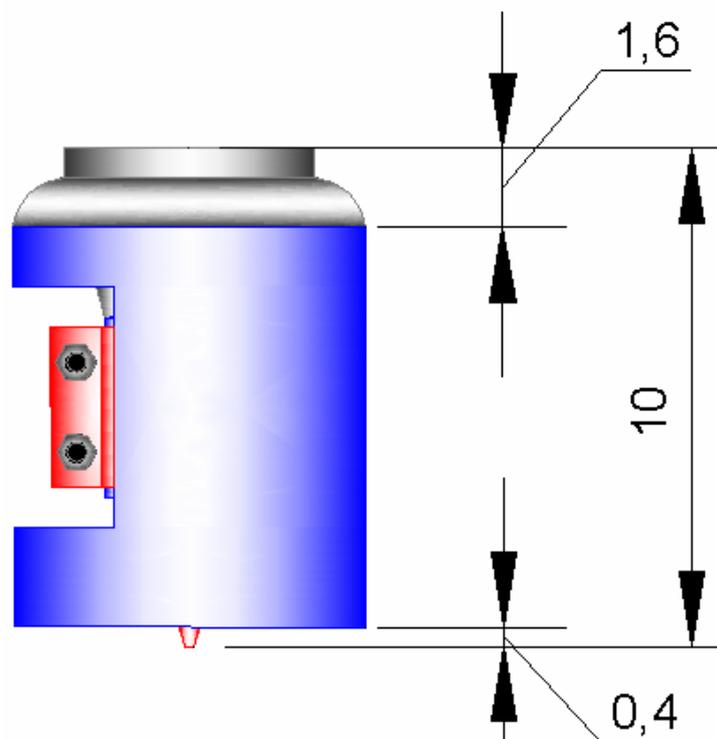


Figura. 2.72. Dispensador de cera – Vista Lateral

\* La unidad de los planos es: centímetros [cm.]

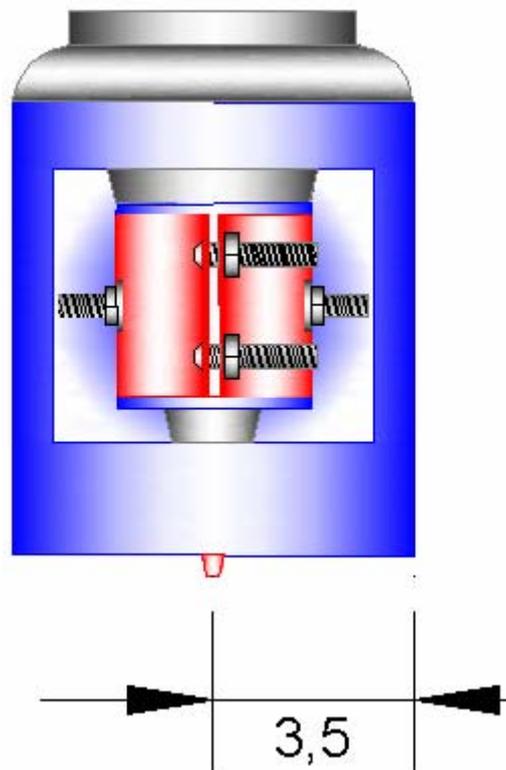


Figura. 2.73. Dispensador de cera – Vista Frontal

La tapa será controlada por un sistema piñón cremallera

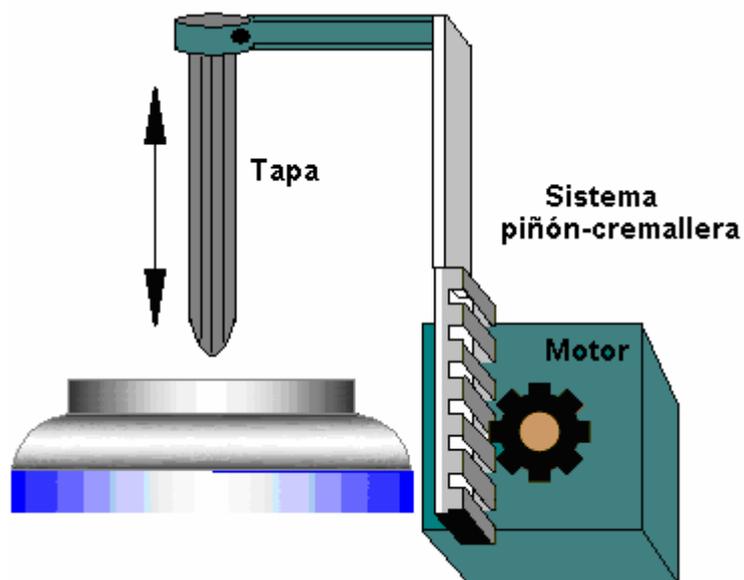


Figura. 2.74. Sistema de Apertura y Cierre del Dispensador

## CAPÍTULO III

### HARDWARE DE CONTROL

#### 3.1 Diseño del hardware control

Para realizar el diseño del hardware de control se debe tener en cuenta el tipo de sistema de control a implementarse, es por esto que a continuación se describe los tipos de sistemas de control que existen y por consiguiente se podrá establecer el sistema que será útil para el proyecto.

##### 3.1.1 Tipos de Sistemas de control

###### Sistema de Lazo Abierto

Los sistemas de control en los que la salida no tiene efecto sobre la acción de control, se denominan sistemas de control de lazo abierto. En otras

palabras, en un sistema de control de lazo abierto la salida ni se mide ni se retroalimenta para compararla con la entrada.<sup>19</sup>

En cualquier sistema de control de lazo abierto, no se compara la salida con la entrada de referencia. Por tanto, para cada entrada de referencia corresponde una condición de operación fija, cabe anotar que la precisión del sistema depende de la calibración.

### Sistema de Lazo Cerrado

En un sistema de lazo de control cerrado, la variable ha ser controlada es continuamente medida y así comparada con un valor predeterminado. Si existe una diferencia entre estas dos variables (error o desviación del sistema), los ajustes son realizados hasta que la diferencia cuantificada es eliminada y la variable controlada iguala la variable de referencia. Con frecuencia se llama así a los sistemas de control retroalimentado. La señal de error actuante, que es la diferencia entre la señal de entrada y la de retroalimentación (que puede ser la señal de salida o una función de la señal de salida y sus derivadas), entra al controlador para reducir el error y llevar la salida del sistema a un valor deseado. El término lazo cerrado implica siempre el uso de la acción de control retroalimentado para reducir el error del sistema.<sup>19</sup>

Al conocer las características de tipo de sistema de control se elige el Sistema de control de Lazo Abierto, puesto que éste se adapta a la forma de controlar la impresora, en donde la salida no tiene efecto sobre la acción de control, es decir, la salida no se mide ni se retroalimenta para compararla con la entrada. En la impresora, se envía la orden desde el computador al controlador (entrada) y los motores del robot cartesiano la ejecutan (salida),

---

<sup>19</sup> [http://www.steamcontrol.com/index\\_docum\\_valvcontrol.html](http://www.steamcontrol.com/index_docum_valvcontrol.html)

entonces, al no comparar la entrada con la salida para efectuar la acción por consiguiente el sistema de control ha utilizarse es el de lazo abierto.

La visión general del control a realizarse se presenta en la figura 3.1.

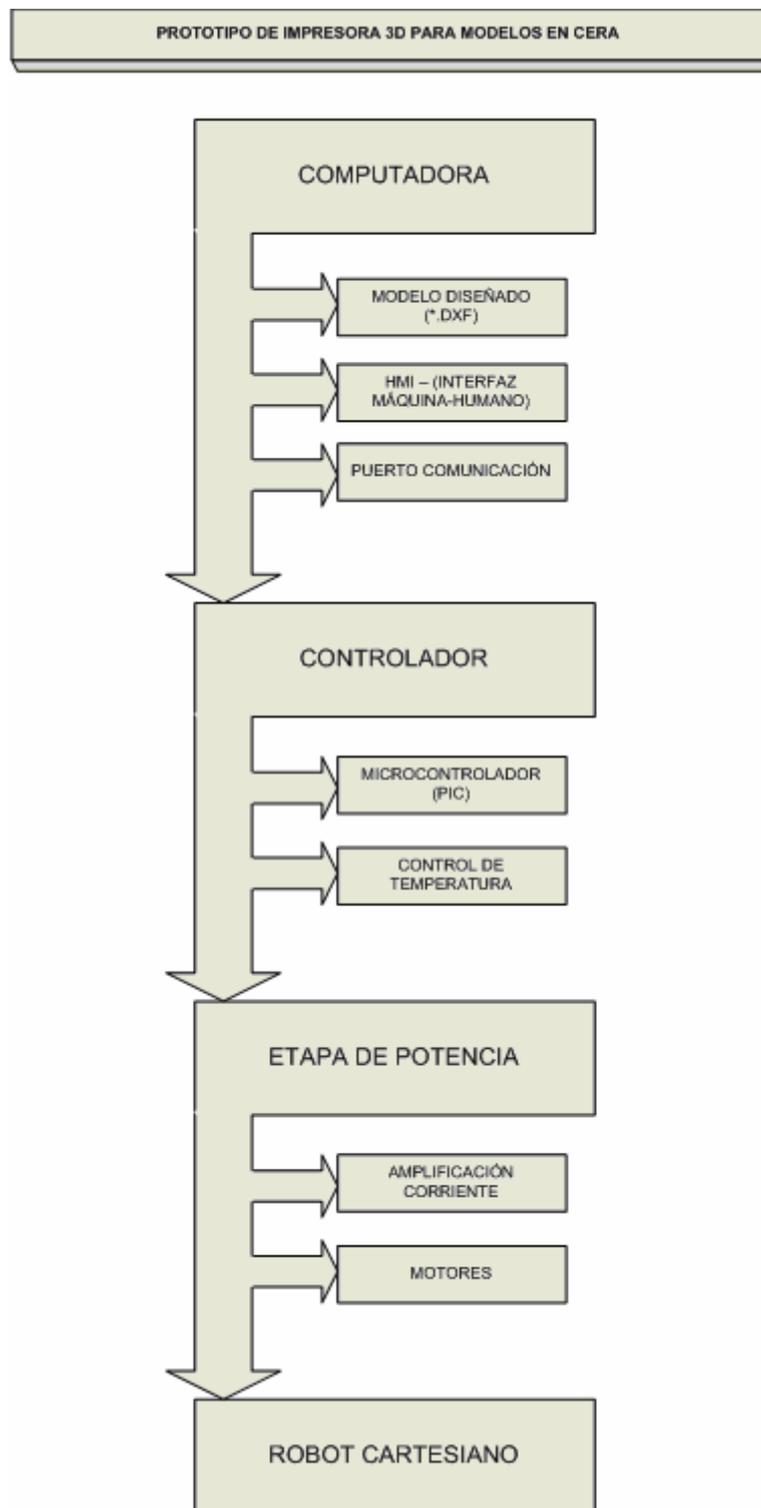


Figura. 3.1. Diagrama de Bloques del Hardware de Control

### 3.1.2 Computadora

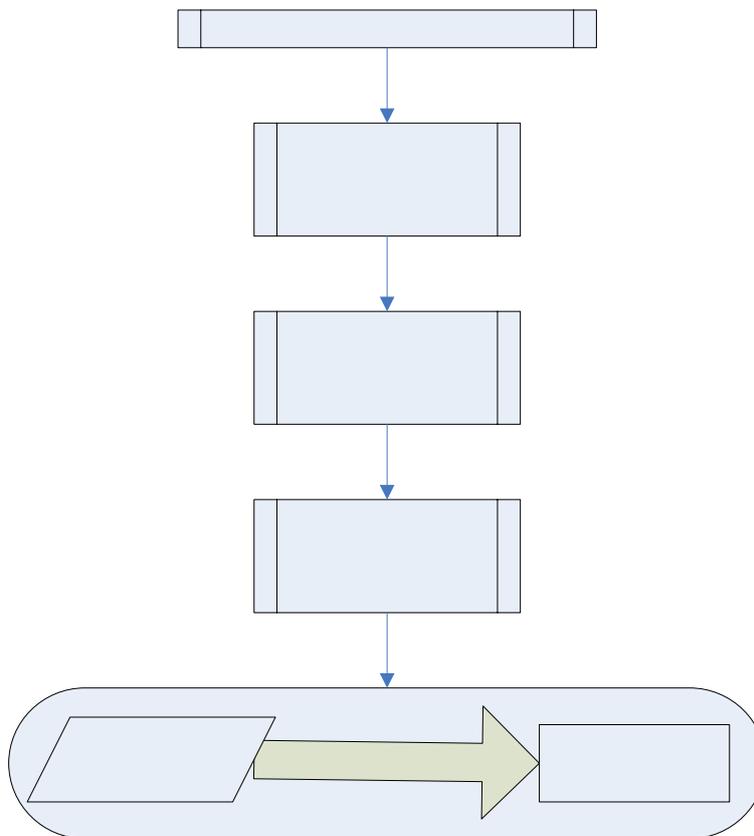


Figura. 3.2. Diagrama de Flujo del Proceso del Computador

En el ordenador se deberá tener el modelo a imprimirse ya diseñado, en un archivo de extensión dxf, en un software que permita dibujar en 3 dimensiones.

Al obtener éste archivo ya se lo puede utilizar en el HMI<sup>20</sup> que está diseñado para la impresora 3D; el software se ejecutará y enviará datos al controlador mediante el puerto paralelo del computador.<sup>21</sup>

<sup>20</sup> Human Machine Interface (Interfaz maquina-humano).

<sup>21</sup> Información ampliada sobre el HMI en el Capítulo V.

COMPUT

MOD  
DISEÑ  
(\*D

Ejecución  
(Interfaz  
Hum

Envío d

### 3.1.2.1 Puerto Paralelo

El puerto paralelo está diseñado para que permita la conexión de impresoras paralelas, además puede ser usado como un puerto de entrada y/o salida genérico, se ha diseñado para manejar impresoras desde la gama de microcomputadores PC/XT/AT.<sup>22</sup>

Desde el punto de vista del hardware, el puerto consta de un conector hembra DB25 con doce salidas latch y cinco entradas, con ocho líneas de tierra. En cuanto al software, el puerto paralelo consta de tres registros (datos, estado y control) de 8 bits cada uno, en la figura 3.3 se representa al conector DB25 y sus registros.

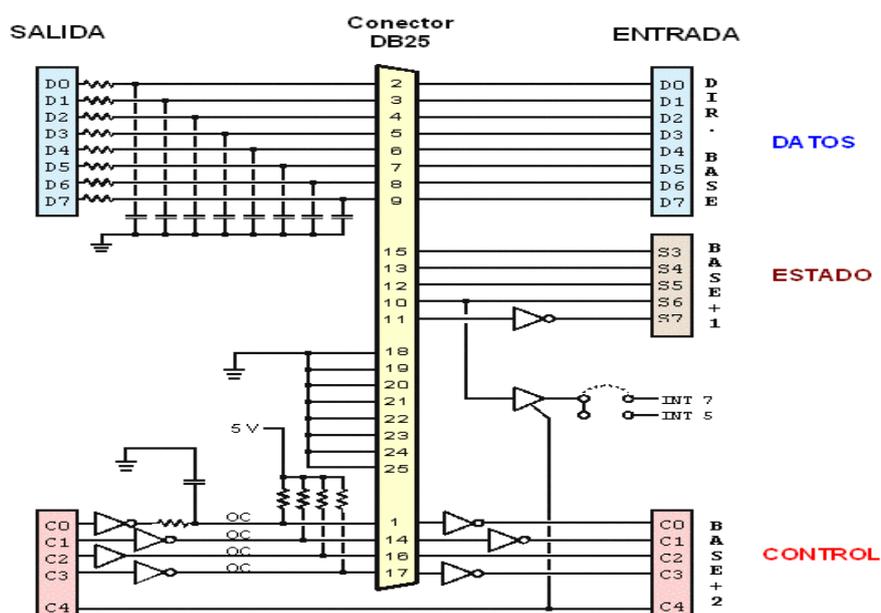


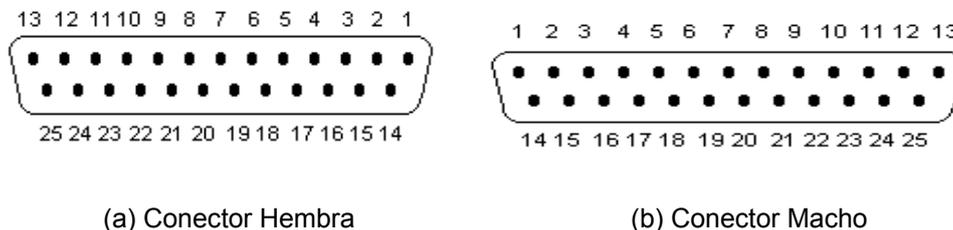
Figura. 3.3. Esquema conector DB25

<sup>22</sup> Lenguaje ensamblador y programación para PC IBM® y compatibles. Peter Abel. Ed. Prentice Hall Hispanoamericana, S.A.1996

La función normal del puerto consiste en transferir datos mediante 8 líneas de salida, usando las señales restantes como control de flujo. Sin embargo, puede ser usado como un puerto E/S de propósito general por cualquier dispositivo o aplicación que se ajuste a sus posibilidades de entrada/salida.

### 3.1.2.2 Conector Físico y Acceso al Puerto

La conexión del puerto paralelo al mundo exterior se realiza mediante un conector hembra DB25. Observando el conector de frente y con la parte que tiene mayor número de pines hacia arriba, se enumera de derecha a izquierda y de arriba a abajo, del 1 al 13 (arriba) y del 14 al 25 (abajo). En las figuras 3.4(a) y 3.4 (b) se presentan sus esquemas.<sup>23</sup>



**Figura. 3.4. Conectores DB25**

Al arrancar el computador, la BIOS<sup>24</sup> chequea direcciones específicas de entradas y salidas en busca de puertos paralelos y construye una tabla de las direcciones halladas en la posición de memoria 40h: 8h (o 0h: 0408h). Esta tabla contiene hasta tres palabras de 16 bits, cada palabra con el byte bajo primero seguido por el byte alto. Cada palabra es la dirección de Entrada/Salida base del puerto paralelo. La primera corresponde a LPT1, la segunda a LPT2 y la tercera a LPT3.<sup>22</sup>

<sup>23</sup> Información sobre los pines del Puerto paralelo en el Anexo 1

<sup>24</sup> BIOS: (Basic Input Output System)

Las direcciones base estándar para los puertos paralelos son: 03BCh, 0378h y 0278h.

### 3.1.2.3 Protocolo de comunicación entre PC y controlador

“Protocolo de comunicación es aquel que establece una descripción formal de dos formatos que deberán presentar los mensajes para poder ser interpretados por equipos de cómputo; además éste debe definir las reglas y parámetros que los mensajes deben seguir para lograrlo.”<sup>25</sup>

En general se ha utilizado varios parámetros del protocolo del puerto paralelo, con respecto a la comunicación entre la conexión paralela del computador y los puertos de los microcontroladores.

La interconexión de un computador con un microcontrolador a través del puerto paralelo posibilita un intercambio de información de una forma rápida, esto es debido a que se produce una transferencia de datos de byte en byte en lugar de ser de bit en bit (1 byte = 8 bits). Cabe anotar que el sistema de comunicación que se maneja en el prototipo de impresora 3D es de tipo unidireccional.

Por medio del puerto de salida de 8 bits, y luego de un proceso de ordenamiento de coordenadas, variables, tiempos, etc., en el software e interfaz HMI se envían datos hacia los microcontroladores (PIC 16F877) que controlan las velocidades de los 4 motores de la impresora, para esto se ha utilizado los valores que componen las 256 combinaciones posibles en este puerto ( $2^8$  combinaciones (0 - 255)).

Con el uso del puerto de salida de 4 bits, con los valores que se generan de las combinaciones de éstas líneas ( $2^4 = 16$  combinaciones) y después del

---

<sup>25</sup> RIOS Ramiro, Fundamentos de Redes - Apuntes, 2005, Ecuador

procedimiento necesario en el programa del HMI se ha logrado controlar el sentido de giro de los motores, claro está, a través de los microcontroladores cubriendo así los 4 cuadrantes del plano cartesiano con respecto al eje X y eje Y con sus respectivas validaciones, al igual que el eje Z y la apertura y cierre de la válvula.

#### **3.1.2.4 Medios de Comunicación**

Los medios de comunicación o conocidos como medios de transmisión son parte importante en todos los circuitos, instalaciones, redes, etc., que se deben tomarlos en cuenta para que no se presenten fallas como pérdida de información o desperdicio de energía.

Existen muchos tipos de medios de transmisión pero en éste caso y en general para circuitos y proyectos electrónicos se utilizan los medios de transmisión por cable.

Características de la conexión vía cable del PC al controlador:

- **Funcionamiento**

Es importante citar que el prototipo de impresora 3D trabajará con sistema Online, es decir, que únicamente funcionará cuando esté conectada al computador vía el cable al puerto paralelo.

- **Distancia**

Al utilizar el puerto paralelo no puede transmitir datos a distancias mayores de 4 o 5 metros (para mayores distancias se usan amplificadores de línea), en el diseño es suficiente ésta distancia

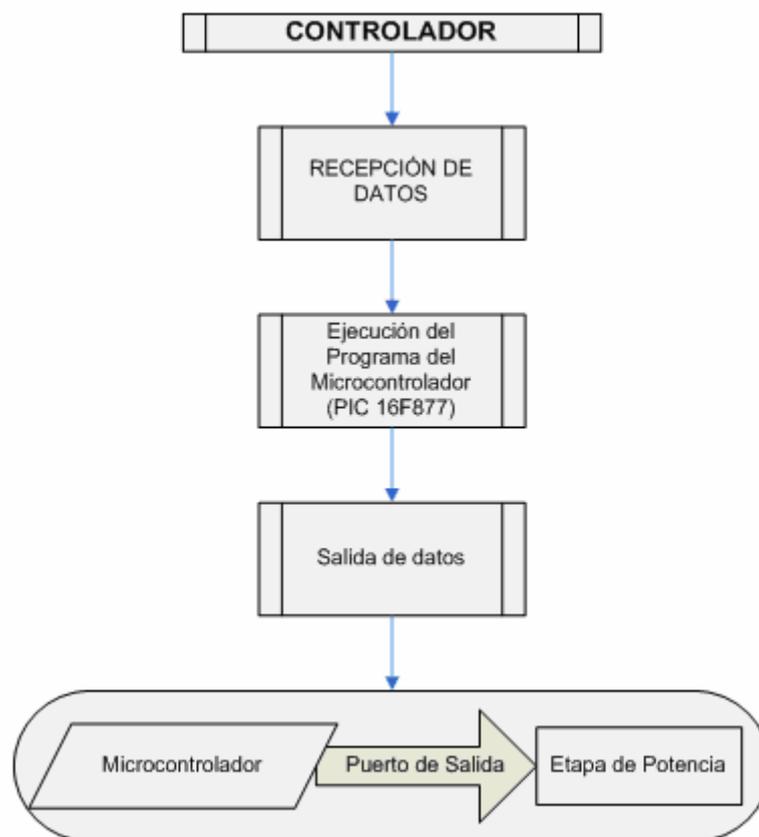
puesto que para el funcionamiento del prototipo de impresora 3D se necesita de una computadora, razón por la cual los datos no viajarán más de 1 o 2 metros como máximo, en la figura 3.5 se presenta un gráfico del cable conector a emplearse.



**Figura. 3.5. Cable conector para puerto paralelo**

- **Conectores**  
Como parte de los elementos están los conectores DB25 macho y hembra para el computador como para el circuito receptor, éstos permiten el flujo de los datos de manera eficiente sin pérdidas significativas de energía. (Figuras 3.4a-b).
- **Circuito impreso**  
En cuanto al circuito de control se implementará en una placa debidamente tratada con elementos nuevos, un buen rutiado lo cual evitará interferencia, ruido y pérdida de energía en todo el proceso.

### 3.1.3 Controlador



**Figura. 3.6. Diagrama de Flujo del Proceso del Controlador**

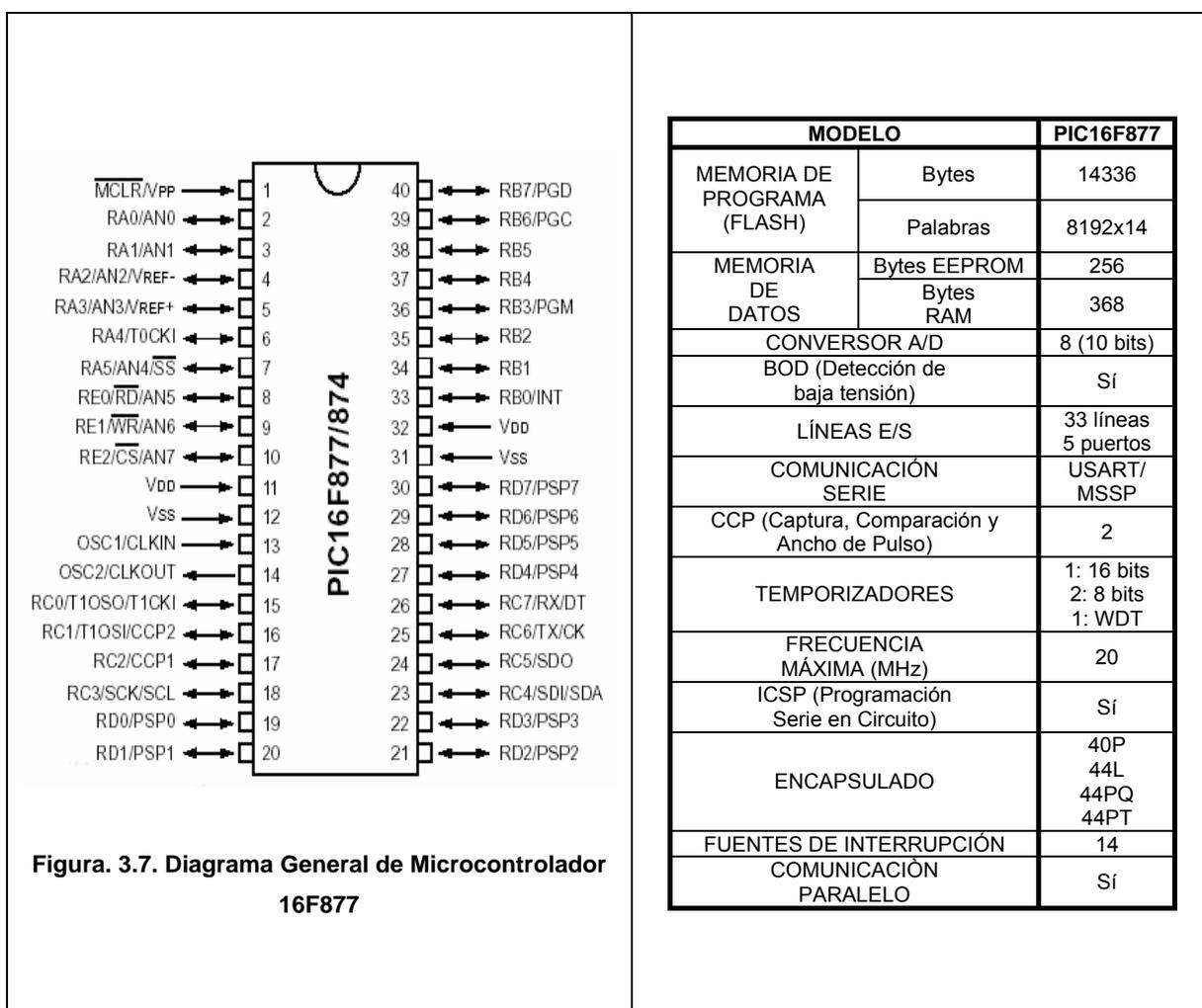
El controlador a utilizarse es el Microcontrolador (PIC) 16F877, éste posee los puertos necesarios de entrada y salida, la velocidad de procesamiento es suficiente y además es ajustable, además la programación es sencilla, tanto en software como en hardware.

Se manejará el puerto A que posee 6 líneas y el puerto C de 8 líneas, en configuración de entradas digitales, mediante los cuales se recibirán los datos enviados desde el computador.

El puerto B, 8 líneas, estarán configuradas como salidas digitales, por éste se enviará la secuencia para el movimiento de los motores, tanto su velocidad como sentido de giro.

A continuación en la tabla 3.1 se presenta las características principales y datos técnicos del PIC 16F877.

**Tabla. 3.1. Características Microcontrolador 16F877**



Es necesario conocer que se utilizará dos microcontroladores los cuales manejarán dos motores de pasos cada uno, así:

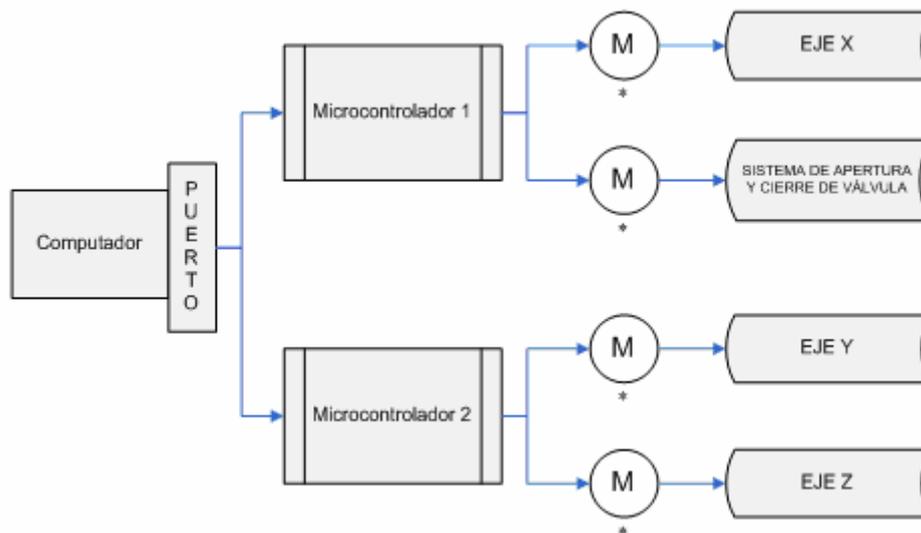
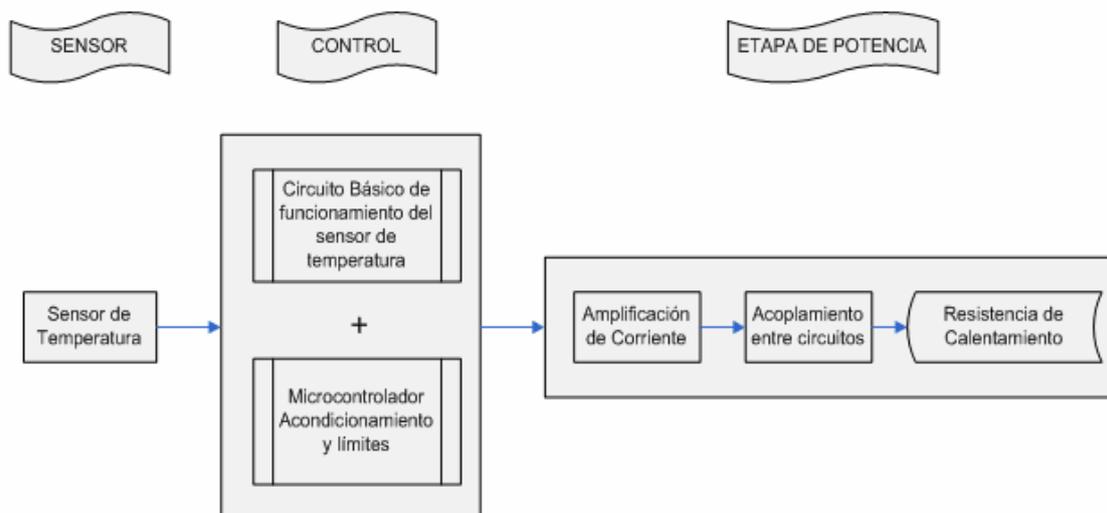


Figura. 3.8. Diagrama de Flujo del Controlador

### 3.1.3.1 Control de Temperatura del Dispensador de Cera

Para la inyección de la parafina es necesario controlar la temperatura de ésta, ya que si llega a una temperatura muy elevada se tornará en una solución líquida y se dispersará por la plataforma, lo que dificultará la formación óptima del modelo.

Por lo que se ha optado diseñar e implementar un control de temperatura, éste estará basado en un sensor y su acondicionamiento y una etapa de potencia, en la figura 3.9 se observa el diagrama de bloques del diseño de este control.



**Figura. 3.9. Diagrama de Bloques del Control de Temperatura**

El sensor de temperatura ha utilizarse será el circuito integrado LM335<sup>26</sup> con su respectivo circuito de funcionamiento, para el acondicionamiento del sensor de 0 a 100 °C se usará un microcontrolador PIC16F877, y por último, para la etapa de potencia se empleará un circuito de amplificación de corriente basado en un transistor y el acoplamiento entre circuitos de control (5 VDC) y potencia (110 VAC) será mediante un relé.

En la figura 3.10 se indica el esquema de conexiones del circuito a implementarse para el control de temperatura del dispensador de cera.

<sup>26</sup> Información sobre características y funcionamiento del sensor LM335 en el Anexo 7

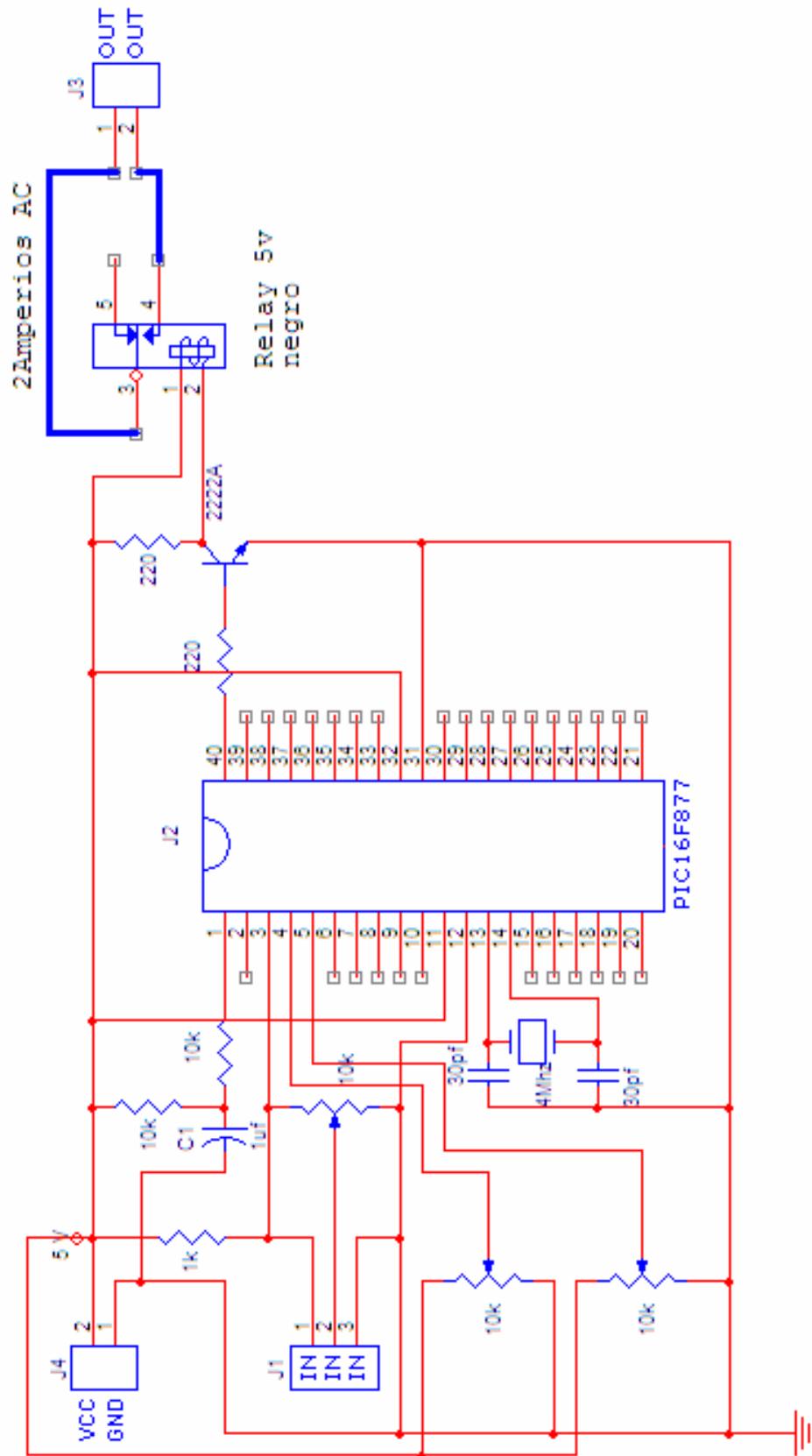


Figura 3.10: Diagrama de Conexiones del Control de Temperatura

Los elementos que se utilizarán para implementar el circuito de control se detalla en la tabla 3.2.

**Tabla. 3.2. Elementos del Circuito de Control de Temperatura del Dispensador de Cera**

<b>ELEMENTO</b>	<b>NUMERACIÓN</b>	<b>CARACTERÍSTICAS</b>	<b>CANTIDAD</b>
<b>J1 (entradas)</b>	LM335	Sensor de temperatura	1
<b>J3 (salidas)</b>	Resistencia (150W)	Para el calentamiento de la parafina	1
<b>J4 (Energización)</b>	VCC: 5V GND	Energización	1
<b>Microcontrolador</b>	16F877	Microcontrolador o conocido como PIC.	1
<b>Cristal</b>	4 MHz.	Reloj del microcontrolador	1
<b>Relé</b>	5VDC 120VAC 10A	Acoplamiento mecánico entre dos circuitos diferentes.	1
<b>Resistencia</b>	1 K $\Omega$	Resistencias, elementos pasivos, protecciones.	3
<b>Resistencia</b>	10 K $\Omega$	Resistencias, elementos pasivos, protecciones.	2
<b>Potenciómetro</b>	10 K $\Omega$	Resistencias variable de 0 a 10 K $\Omega$	3
<b>Capacitor</b>	30 pF	Almacenador de energía, para evitar ruido	2
<b>Capacitor</b>	1 uF	Almacenador de energía, para evitar ruido	1

### 3.1.4 Etapa potencia

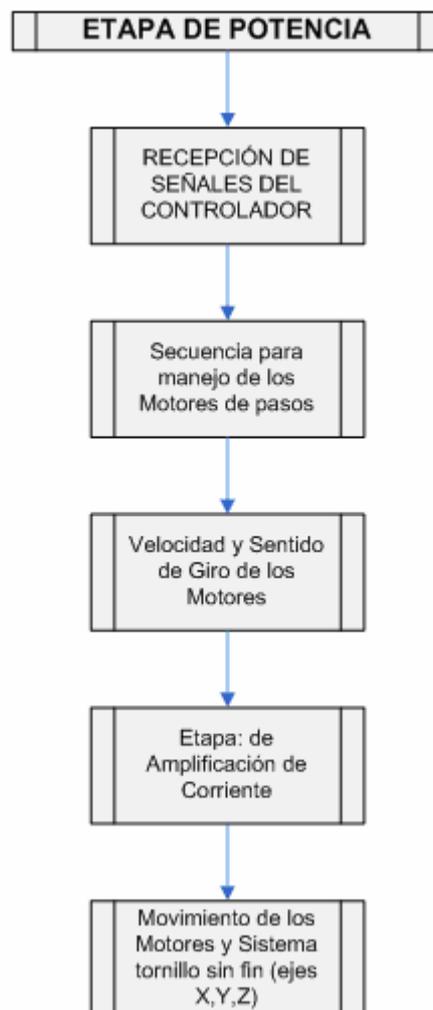


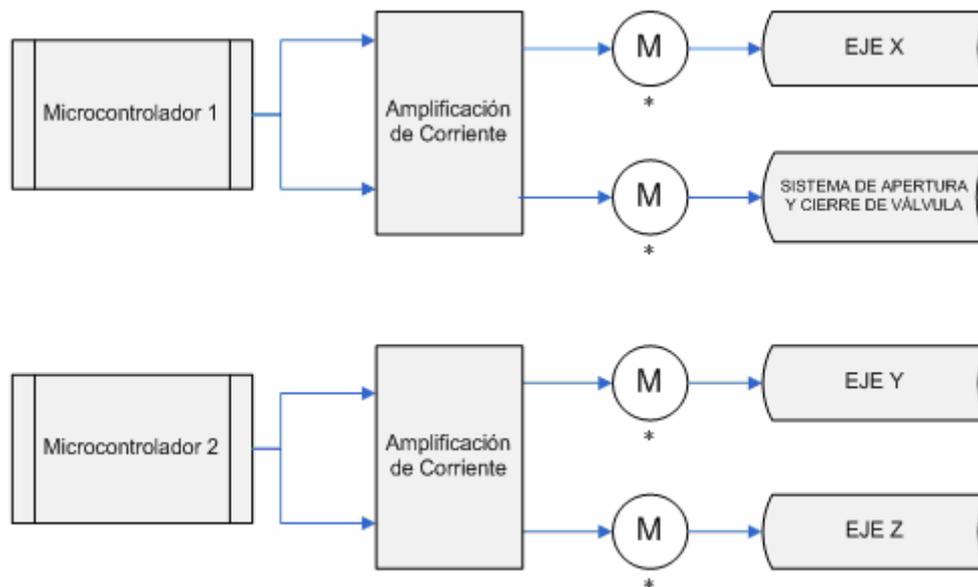
Figura. 3.11. Diagrama de Flujo de la Etapa de Potencia

Del puerto de salida del controlador se obtendrá las señales necesarias para poner en movimiento a los motores controlando el sentido de giro y velocidad de éstos.

Para el control del movimiento de los motores de pasos se utilizará una secuencia adecuada y una etapa de amplificación de corriente que es necesaria por el consumo de los motores.

Para el acoplamiento entre la etapa de control y la etapa de potencia se utilizará los circuitos integrados PC817 que son opto-acopladores<sup>27</sup>, luego se amplificará la corriente mediante transistores de potencia ( D1889) que permitirán suministrar a los bobinados de los motores la corriente suficiente para su óptimo funcionamiento.

El diagrama de bloques de la etapa de potencia se la puede observar en la figura 3.12 y el circuito de conexiones en la figura 3.13.



**Figura. 3.12. Diagrama de la Etapa de Potencia**

<sup>27</sup> Información sobre las características y funcionamiento de los opto-acopladores en el Anexo 5.

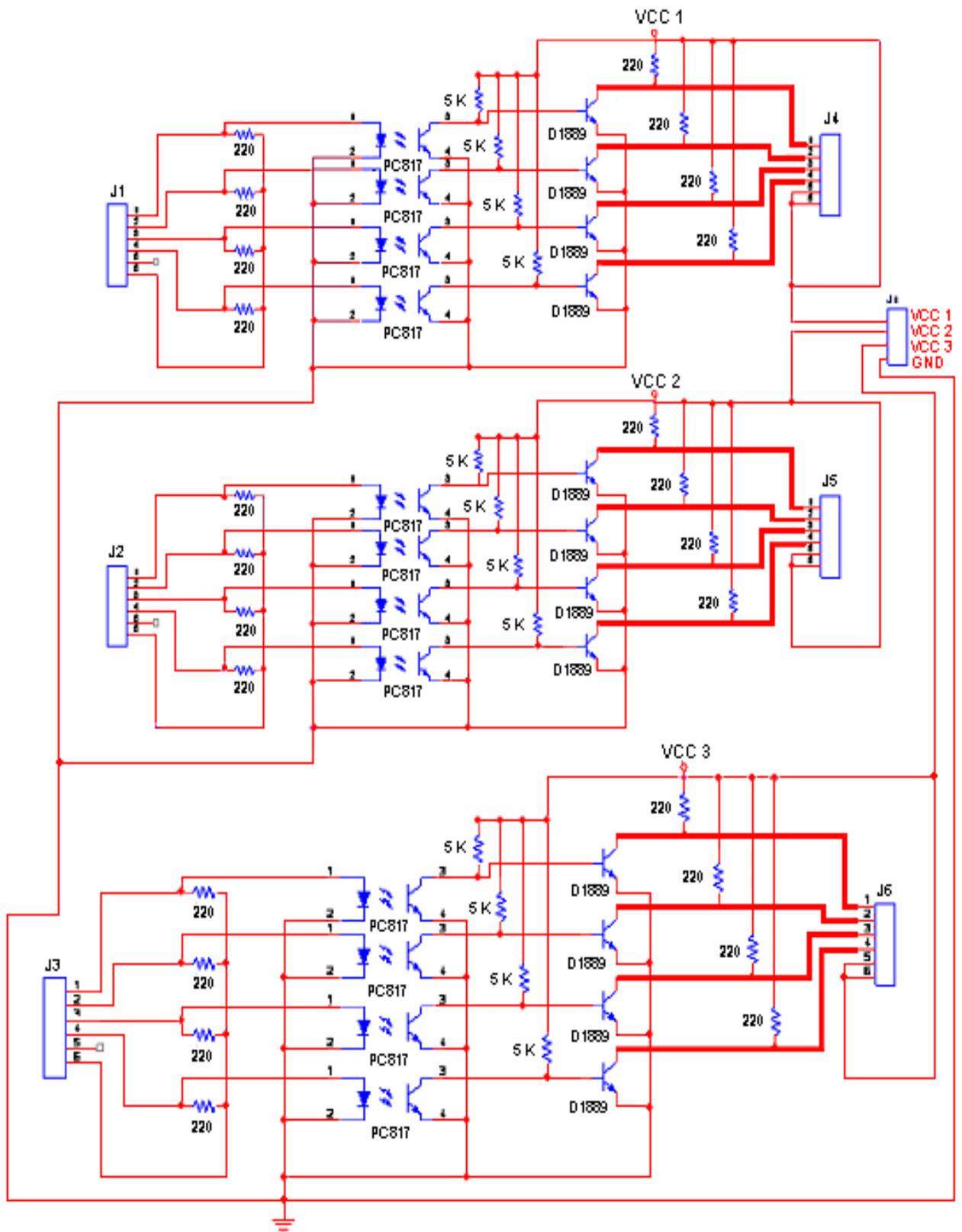


Figura. 3.13. Esquema de Conexiones de la Etapa de Potencia

A continuación se profundiza en el funcionamiento, secuencia de giro y control de los motores de pasos:

### 3.1.4.1 Motores paso a paso

Se conoce que un motor es un sistema basado en las leyes del electromagnetismo básicas, convierte la energía eléctrica en energía mecánica. Además es un sistema reversible.

La parte motriz del robot está dada por tres barras roscadas (tornillos), uno para cada eje del robot cartesiano (X, Y, Z), que actuarán según el sistema de tornillo sin fin, acoplados por medio de bocines a los ejes de 3 motores paso a paso respectivamente.



**Figura. 3.14. Motor de Pasos**

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ),

para completar un giro completo de 360°. Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres.

Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas. Los motores paso a paso del tipo de imán permanente son los más usados en robótica.

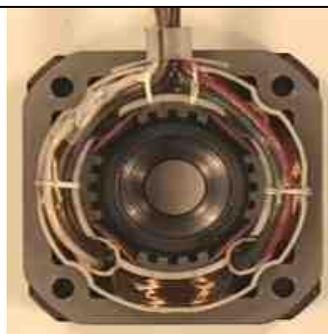
#### 3.1.4.2 Principio de funcionamiento

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator. (Figuras 3.15 y 3.16)

Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.



**Figura. 3.15. Rotor de un motor a pasos**

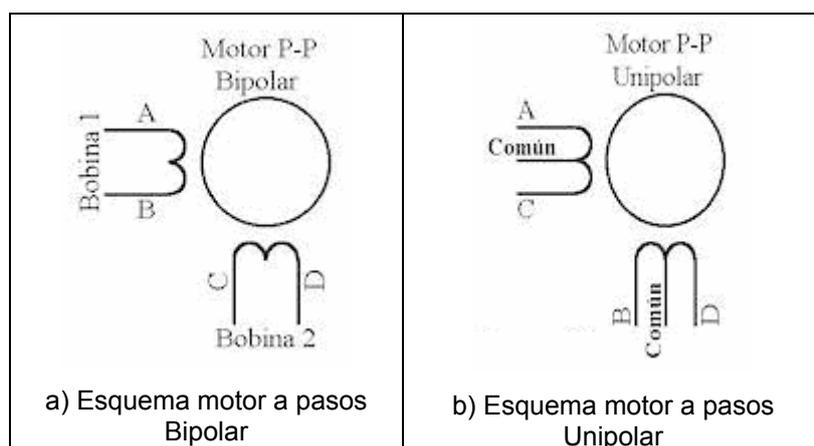


**Figura. 3.16. Estator de 4 bobinas de un motor a pasos**

Existen dos tipos de motores paso a paso de imán permanente:

Bipolar: Estos tienen generalmente cuatro cables de salida (figura 3.17-a). Necesitan ciertas consideraciones para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

Unipolar: Estos motores suelen tener 6 o 5 cables de salida, dependiendo de su conexión interna (figura 3.17-b).



**Figura. 3.17. Esquema motores a pasos**

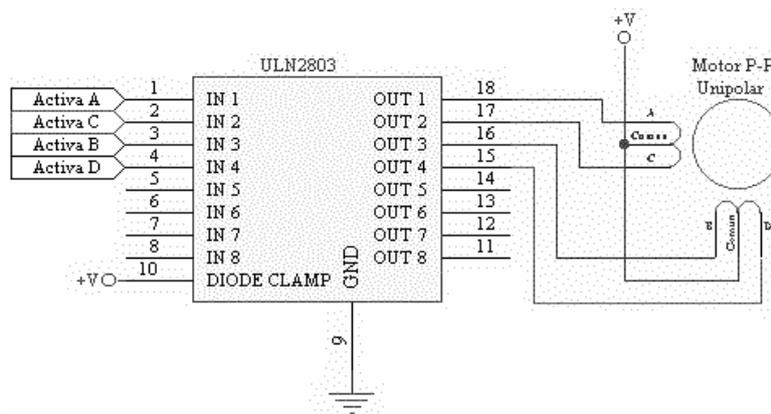
Los motores ha utilizarse en el diseño del prototipo de impresora 3D son de imán permanente **unipolares**.

Este tipo de motor se caracteriza por ser más simple de controlar.

En la figura 3.18 se puede observar un diagrama de conexiones para controlar un motor paso a paso unipolar, mediante el uso del circuito integrado

ULN2803, el cual es un conjunto de 8 transistores tipo Darlington capaces de manejar cargas de hasta 500mA.

Las entradas de activación (Activa A, B, C y D) pueden ser directamente activadas por un microcontrolador.



**Figura. 3.18. Control con CI ULN2803 de motor a pasos Unipolar**

En el diseño planteado se utilizará motores paso a paso unipolares, y para esto se analiza a continuación la secuencia de manejo de éstos.

### 3.1.4.3 Secuencia para manejar motores paso a paso Unipolares

Existen algunas secuencias posibles para el manejo de este tipo de motores, pero la que se empleará se detalla a continuación.

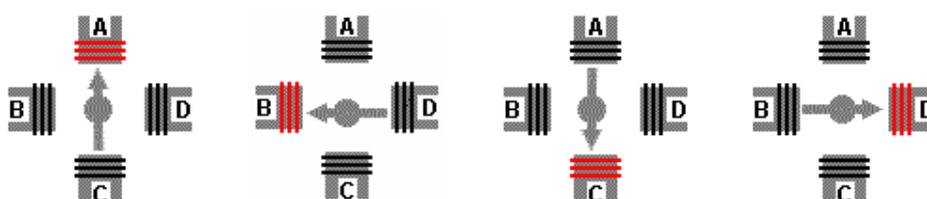
Secuencia del tipo wave drive:

En esta secuencia se activa solo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La contrapartida es que al estar solo una bobina activada, el torque de paso y retención es menor. (Tabla 3.3 y Figura 3.19)

Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay una bobina activada. Para revertir el sentido de giro, simplemente se deben ejecutar las secuencias en modo inverso.

**Tabla. 3.3. Estator de 4 bobinas de un motor a pasos**

PASO	Bobina A	Bobina B	Bobina C	Bobina D
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON



**Figura. 3.19. Secuencia animada en modo wave drive**

Los motores paso a paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos aplicados es un punto muy importante a tener en cuenta. En tal sentido el

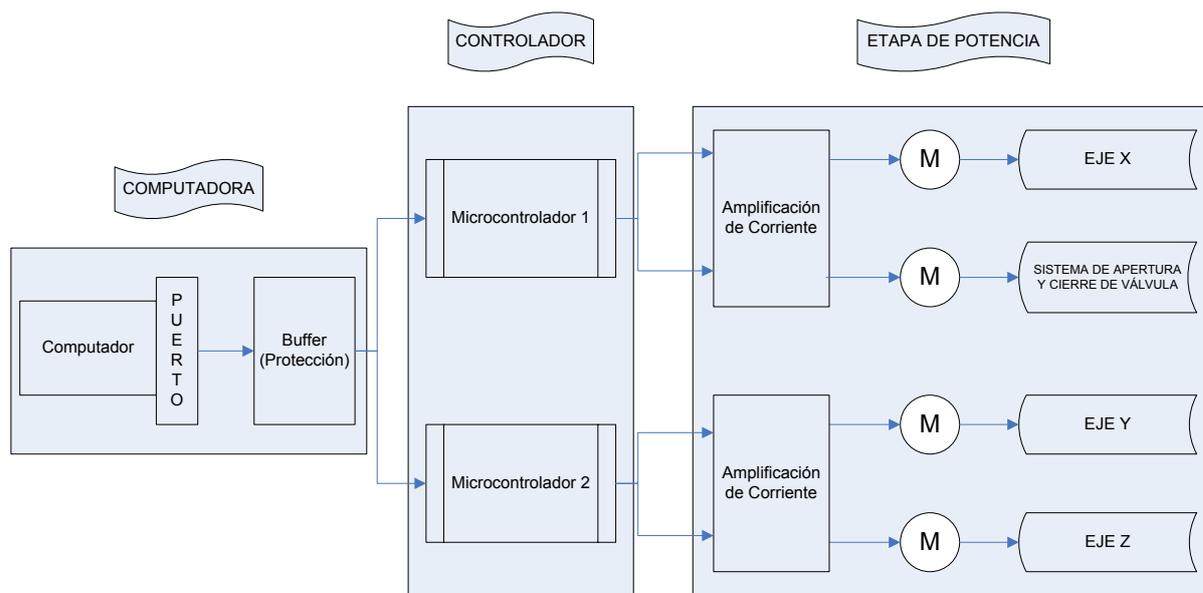
motor debe alcanzar el paso antes que la próxima secuencia de pulsos comience. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar en alguna de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar pero sin llegar a girar.
- Puede girar erráticamente.
- Puede llegar a girar en sentido opuesto.

Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y gradualmente ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada. El giro en reversa debería también ser realizado previamente bajando la velocidad de giro y luego cambiar el sentido de rotación.

En caso de no conocer la distribución de los cables a los bobinados y el cable común de un motor de pasos se presenta información en el Anexo 2.

El hardware total se resume en la figura 3.20.



**Figura. 3.20. Diagrama de Flujo del Hardware Total**

La información procesada por el HMI en el computador genera los datos e instrucciones que se enviarán mediante el puerto paralelo, éstos pasan por los buffers de protección, luego llegan a los puertos de los microcontroladores (Puertos A y C), se ejecutan los programas en los PICs y estos envían sus señales de salida por el puerto B a la etapa de amplificación de corriente, a su vez estos últimos están alimentando y controlando las bobinas de cada motor y en consecuencia la velocidad y sentido de giro.

Cada uno de los microcontroladores está encargado de manejar dos motores y están divididos así:

PIC1 → Motores del eje X y sistema de apertura o cierre de la válvula.

PIC2 → Motores del eje Y y eje Z.

### 3.2 Programación del Controlador

Para la programación de los microcontroladores (PIC 16F877), son necesarios cuatro elementos indispensables, que se detalla a continuación:

- Lenguaje de Programación.
- Compiladores, Ensambladores.
- Software para programación de microcontroladores (PICs).
- Hardware de programación.

En cuanto a los elementos antes citados, existen un sin número de opciones disponibles al alcance del programador, pero, para el propósito del proyecto se debe tomar en cuenta que el lenguaje que se escoja tenga un compilador que permita trabajar de manera rápida y precisa, además es aconsejable utilizar un simulador para depurar errores antes de proceder a la programación ya en el PIC, para éste último paso también se cuenta con algunos programas especializados. En la tabla 3.4 se señalan algunas opciones.

**Tabla. 3.4. Lenguajes, compiladores y programador disponibles**

Lenguajes	Compiladores	Programador para PICs
Assembler-Sides	MPLAB	IC-PROG
Lenguaje C	MPASM	
	HT-PIC	

Para el proyecto se optará por emplear: Lenguaje C, el compilador HT-PIC y el software IC-PROG, para creación del programa, compilación y programación del PIC respectivamente.<sup>28</sup>

<sup>28</sup> El proceso de programación del microcontrolador y explicación del software se puede observar en el ANEXO 3.

### 3.2.1 Lógica de Programación

La lógica de programación se representa en el diagrama de flujo de la figura 3.23, pero a continuación se explica de forma detallada su contenido:

- Luego del procesamiento de la información del modelo a imprimirse, tanto en el software de dibujo en 3D, como en el de conversión de coordenadas y en el HMI, los datos están listos para su envío.
- El envío de datos se lo hace por medio del puerto paralelo del computador hacia los microcontroladores.
- Los datos llegan al de forma simultánea a los dos microcontroladores, éstos son receptados por los puertos A (4bits) y C (8bits) de cada PIC.
- Según el valor recibido por el puerto A se escoge el motor y la dirección que se debe ejecutar.
- De la misma manera para el puerto C según el valor receptado se determina la velocidad de cada motor.
- Los cuadrantes y condiciones referidos en el diagrama de flujo se presentan en la figura 3.22 y en la tabla 3.5.\*

---

\* Los programas grabados en los microcontroladores se los puede observar en el Anexo 1.

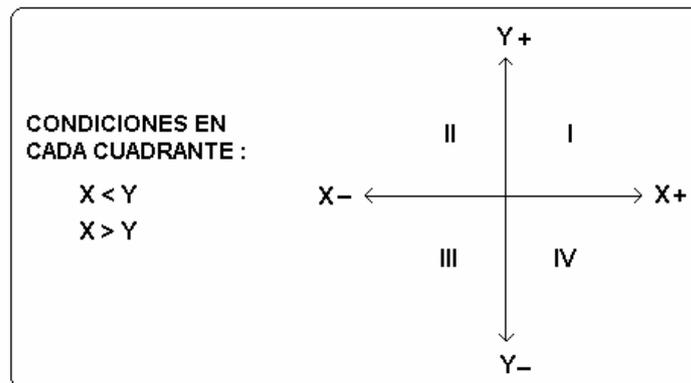


Figura. 3.22. Cuadrantes del Plano cartesiano

Tabla. 3.5. Combinaciones y funciones

# COMBINACION	FUNCION Y CONDICION
0	EJE X (POSITIVO)
1	EJE X (NEGATIVO)
2	EJE Y (POSITIVO)
3	EJE Y (NEGATIVO)
4	CUADRANTE I ( $X > Y$ )
5	CUADRANTE I ( $X < Y$ )
6	CUADRANTE IV ( $X > Y$ )
7	CUADRANTE IV ( $X < Y$ )
8	CUADRANTE III ( $X > Y$ )
9	CUADRANTE III ( $X < Y$ )
10	CUADRANTE II ( $X > Y$ )
11	CUADRANTE II ( $X < Y$ )
12	APERTURA VALVULA
13	CIERRE VALVULA
14	EJE Z
15	NO TRABAJA NINGUN MOTOR

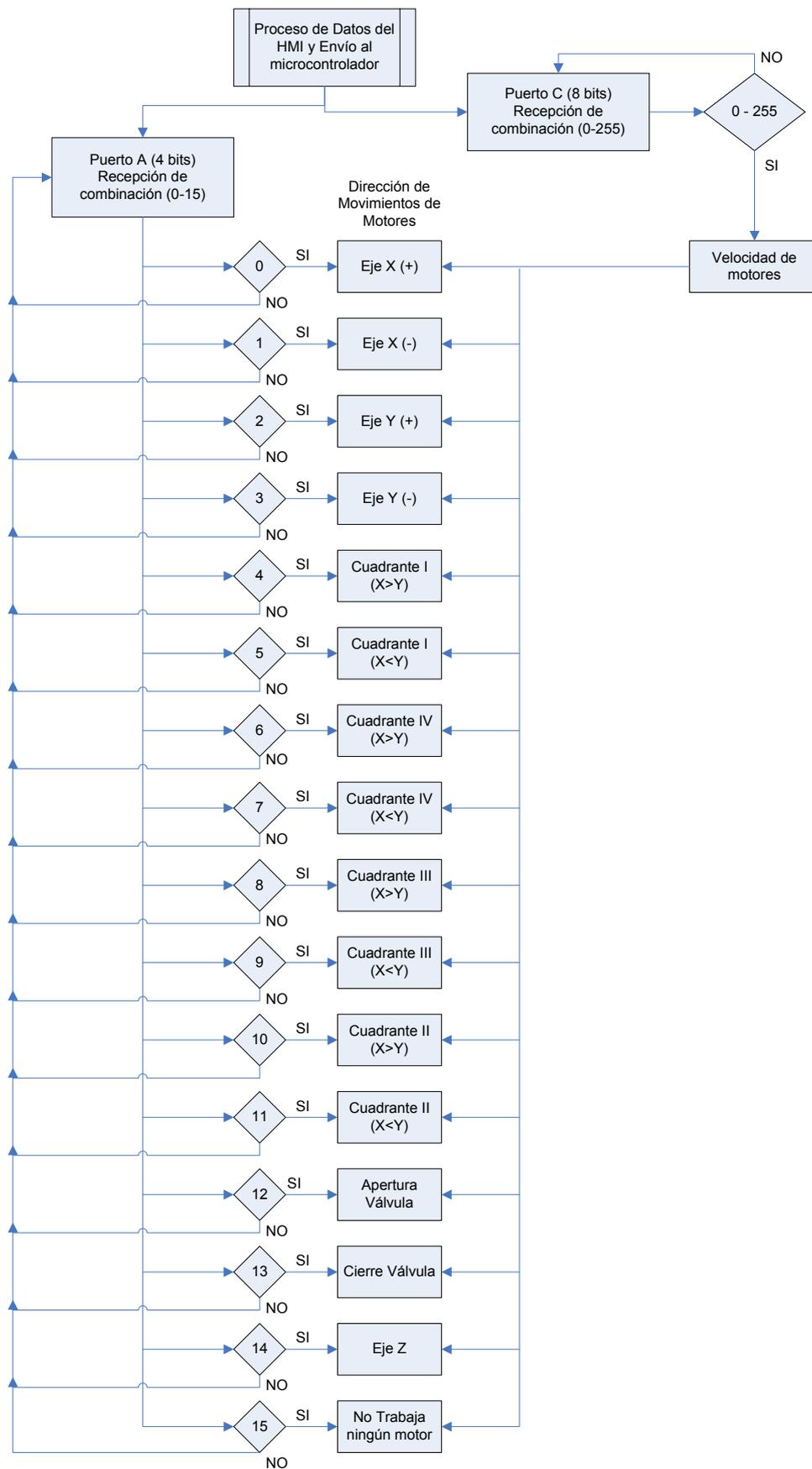
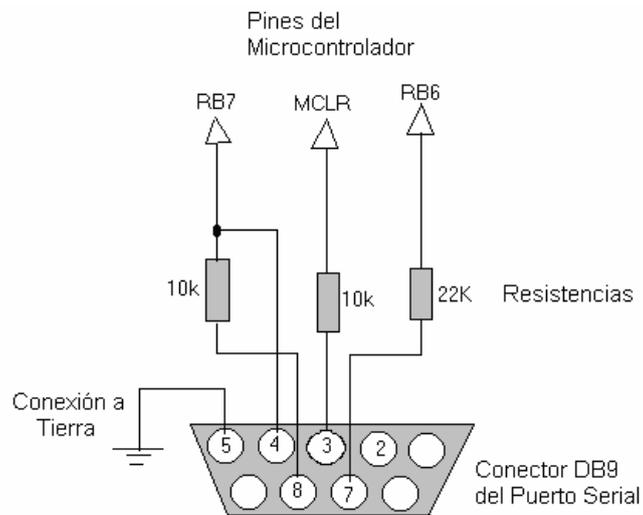


Figura. 3.23. Diagrama de Flujo de la Lógica de Programación del Controlador

### 3.2.2 Hardware de Programación

Se ha implementado un circuito que permite la programación del PIC Microcontrolador 16F877, comunicado con la computadora vía puerto serial y que es compatible con el software de programación IC-PROG. (Figura 3.24)



**Figura. 3.24. Diagrama Conexión DB9-PIC**

## CAPÍTULO IV

### DESARROLLO DEL SOFTWARE

#### 4.1 Software de conversión de coordenadas

Luego de tener el modelo dibujado en el software de diseño gráfico en 3D se necesita conocer las coordenadas exactas de los puntos principales que conforman el dibujo.

Todo software de dibujo tiene un archivo de intercambio de datos con otro software en el cual presenta la información de todo el dibujo, color, coordenadas, tipo, especificaciones, etc. La extensión de este archivo es \*.dxf.<sup>29</sup>

Para el control de el Robot Cartesiano se necesita solamente las coordenadas específicas hacia donde se debe mover cada eje, siendo por esta razón que se necesita de un convertidor de archivos dxf a Código G, este tipo de código se utiliza tanto en robots cartesianos como en máquinas CNC.<sup>30</sup>

---

<sup>29</sup> Dxf: Drawing Exchange File

<sup>30</sup> CNC: Control numérico computarizado

Para la conversión de los archivos dxf a código G se utiliza el software “ACE converter”; este programa es libre. Este software entrega códigos para máquinas CNC sin un orden específico, para el caso del prototipo de impresora 3D es necesario que los puntos entregados por el convertidor sigan un orden específico, por lo que se genera este orden a través de una macro en Microsoft Excel.

#### **4.1.1 Convertidor ACE**

##### Introducción

El convertidor ACE traduce archivos del CAD<sup>31</sup> (diseño automatizado) a archivos manipulables por maquinas CNC y Robots.

La mayoría de los usos del CAD permiten hoy que el usuario exporte archivos de dibujo en varios diversos formatos. Uno de esos formatos es DXF (formato de dibujo del intercambio). Se hace esto para poder leer el mismo dibujo en cualquier software de dibujo. El convertidor ACE lee archivos DXF y convierte los dibujos a código G.

##### Requisitos del Sistema

- Funciona en cualquier versión de Windows, de Win95 hacia adelante.
- Requiere disco libre aproximadamente 500kM (3M para el código de fuente y las herramientas de desarrollo).<sup>32</sup>

---

<sup>31</sup> CAD: Computer assist Design

<sup>32</sup> Información sobre el Software Conversor ACE en el Anexo 4.

### 4.2 Interfaz HMI<sup>33</sup>

La interfaz gráfica ha sido diseñada para que el usuario este apto para manejar el software sin ningún problema, se diseñó siguiendo una secuencia lógica de programación la cual se detalla en el diagrama de bloques de la figura 4.1.

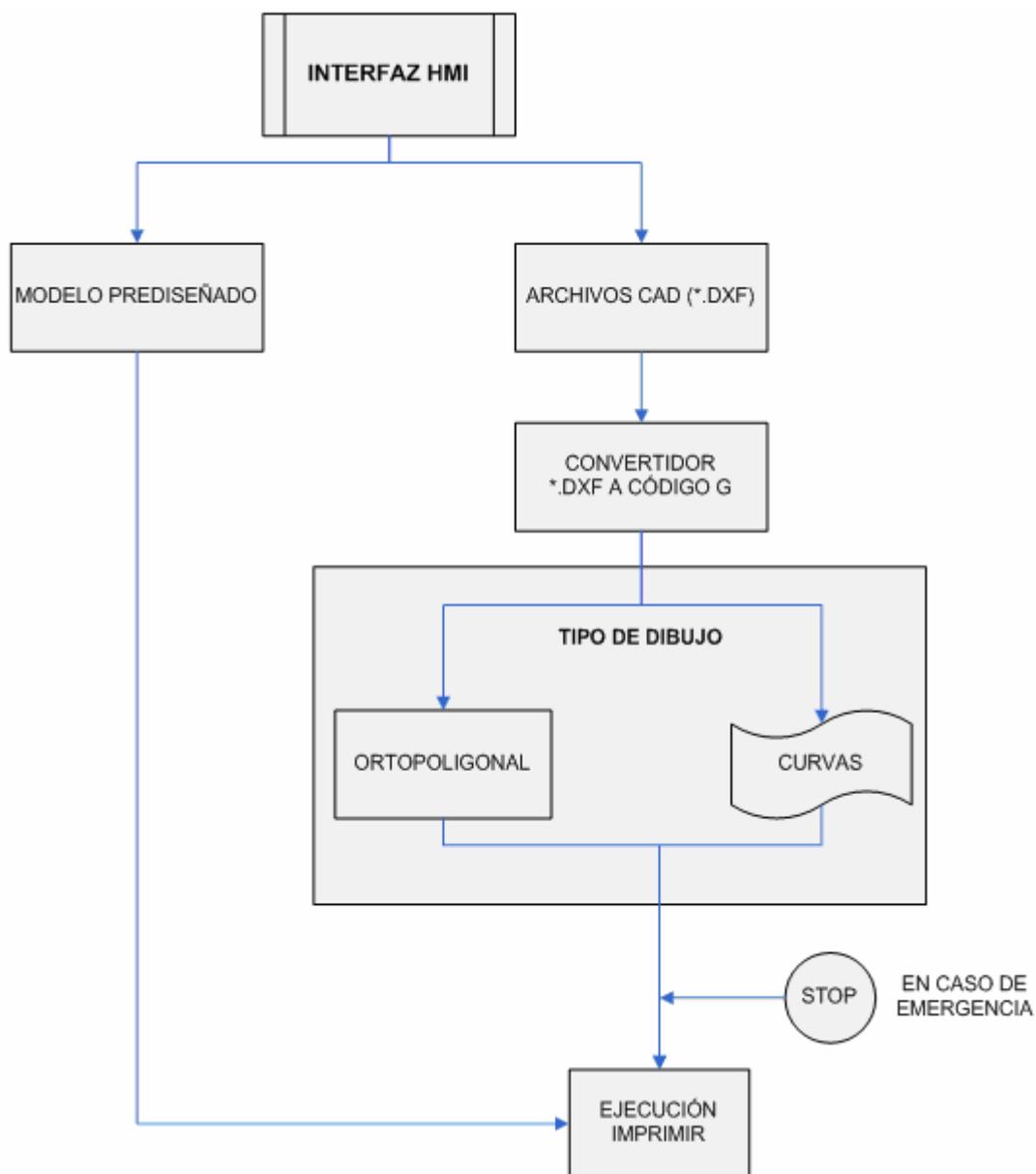


Figura. 4.1. Diagrama de Bloques del HMI

<sup>33</sup> HMI: Interfaz Maquina-Humano

### 4.2.1 HMI

En esta pantalla se presentan todas las opciones que se requieren para el uso adecuado del HMI.

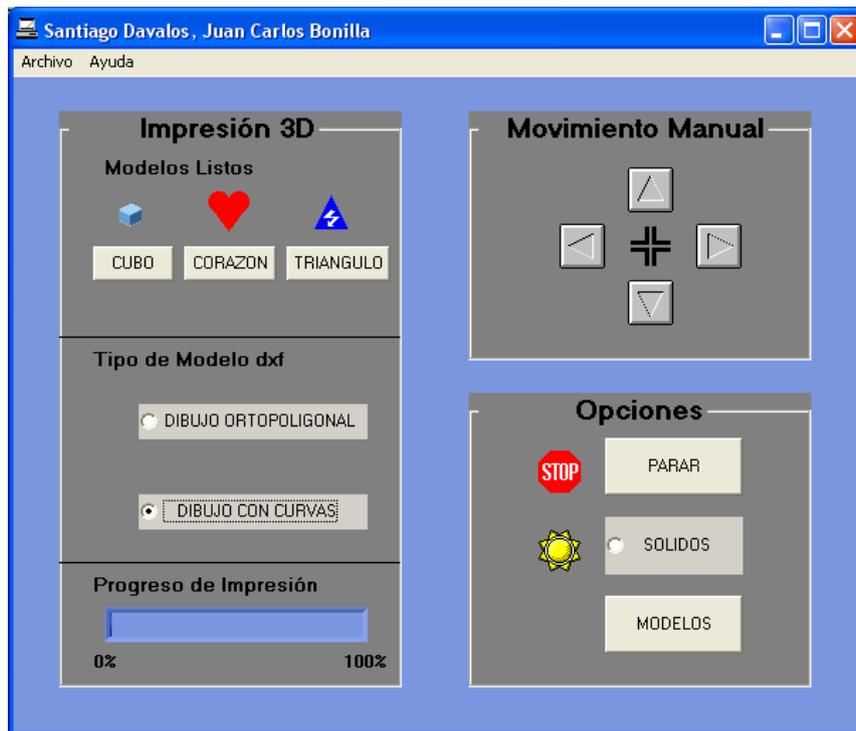


Figura. 4.2. HMI

### 4.2.2 Archivos CAD (\*.dxf).

La mayoría de software de dibujo permite exportar archivos en varios formatos. Uno de esos formatos es DXF (formato de dibujo del intercambio). El convertidor ACE lee archivos DXF y convierte los dibujos a código G. En esta pantalla se permite importar hacia el HMI cualquier archivo con extensión dxf, el cual luego será convertido a Código G.

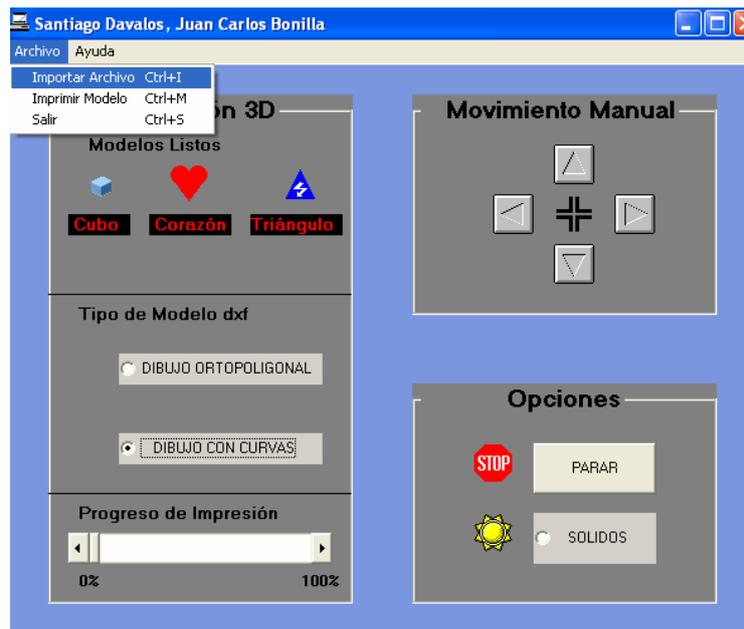


Figura. 4.3. Bloque Archivos CAD

#### 4.2.3 Conversión archivo \*.dxf a Código G

Luego de seleccionar la opción “Importar Archivo” el Convertidor ACE de archivos dxf a Código G se desplegará en la pantalla como lo muestra la figura 4.4.

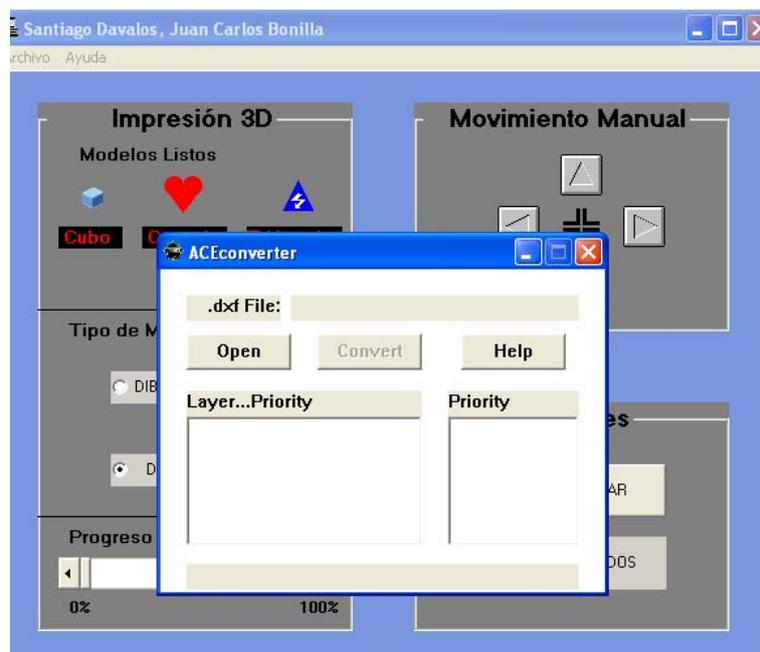
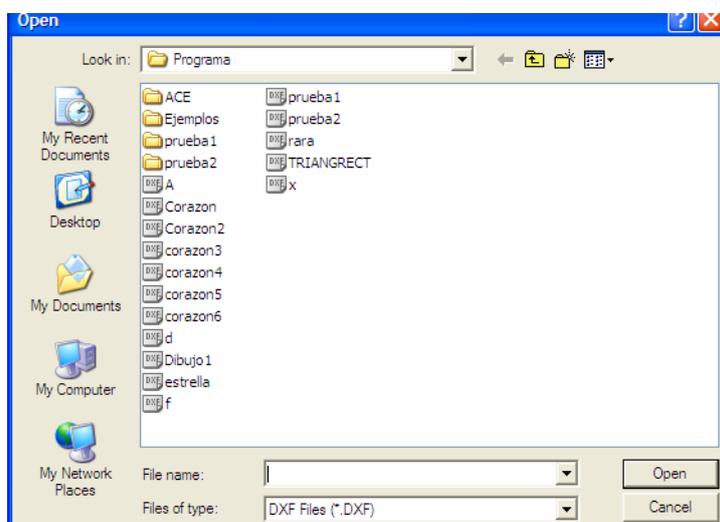


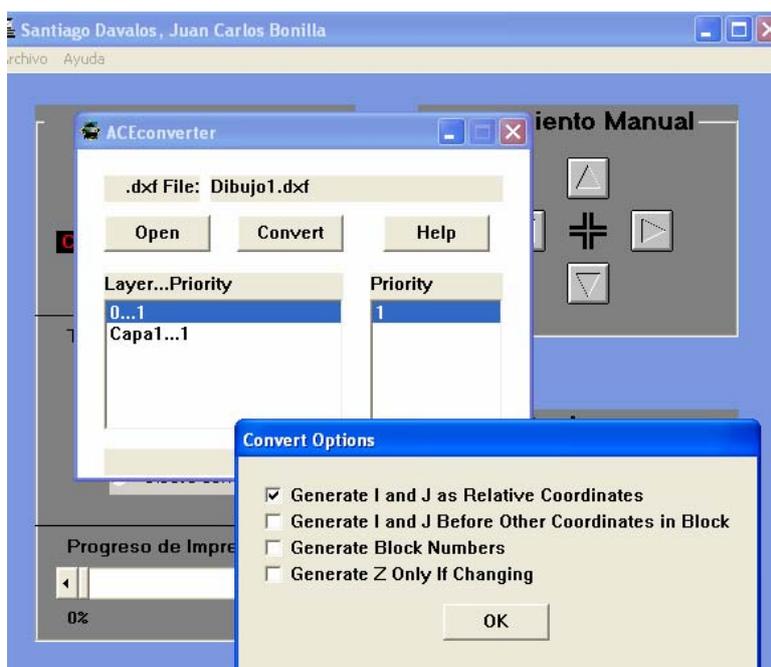
Figura. 4.4. Convertidor ACE

Elegimos la opción OPEN para importar al HMI el archivo Dxf de nuestro diseño la cual se muestra en la siguiente pantalla:



**Figura. 4.5. Importar Archivo \*.dxf**

Se selecciona el archivo y se convierte a con la opción “Convert”, al desplegarse un cuadro de opciones se elige la primera ya que es utilizada para máquinas CNC y robots.



**Figura. 4.6. Conversión de Archivo**

Luego el convertidor genera un archivo de extensión \*.txt, donde se encuentran todas las coordenadas necesarias para la impresión, por último se despliega un mensaje de conversión completa, como se observa en las figuras 4.7 y 4.8.

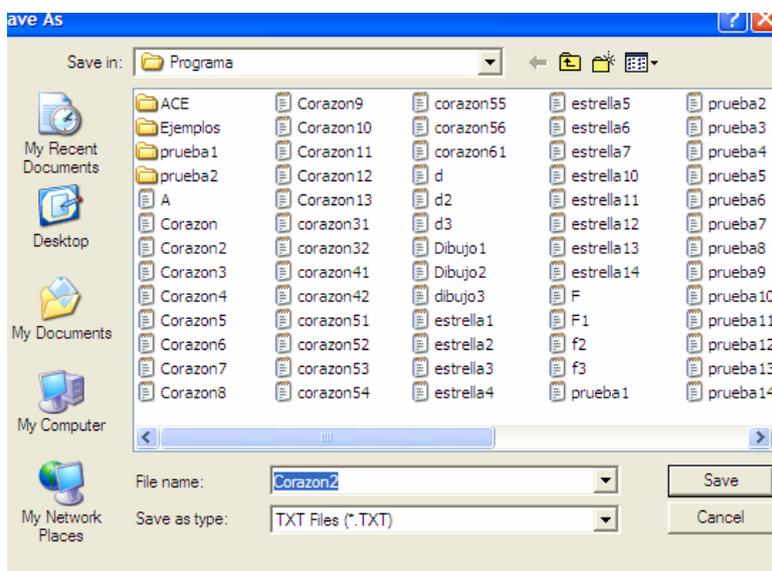


Figura. 4.7. Conversión de Archivo

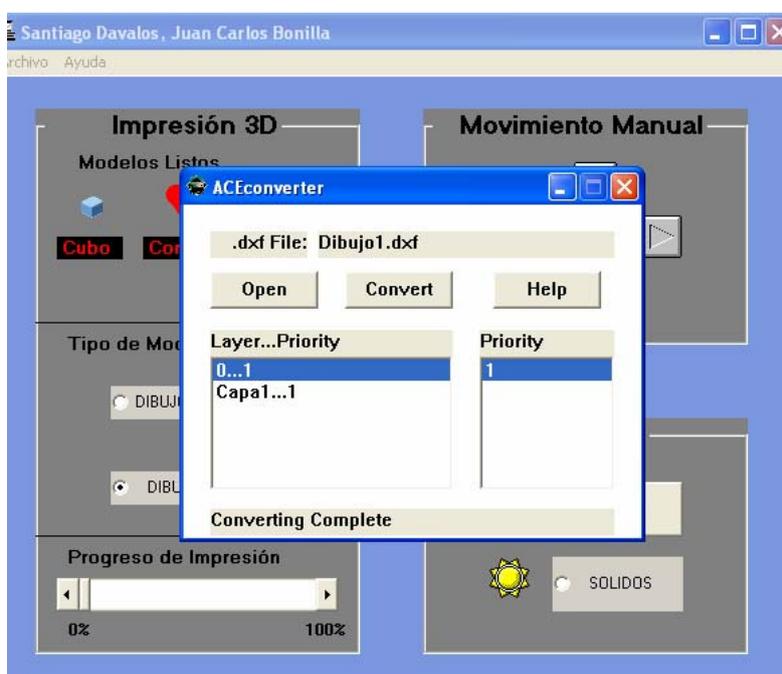


Figura. 4.8. Conversión completa

#### 4.2.4 Tipo de Dibujo (Ortopoligonal o Curvas)

Para mayor rapidez en la impresión de modelos ortopoligonales el HMI cuenta con la opción “Dibujo Ortopoligonal”, a su vez para mayor definición en modelos que contengan arcos o círculos el HMI también posee la opción “Dibujo con curvas” la cual divide a estas entidades en segmentos de rectas. Se debe escoger una de las opciones antes de ejecutar la impresión.

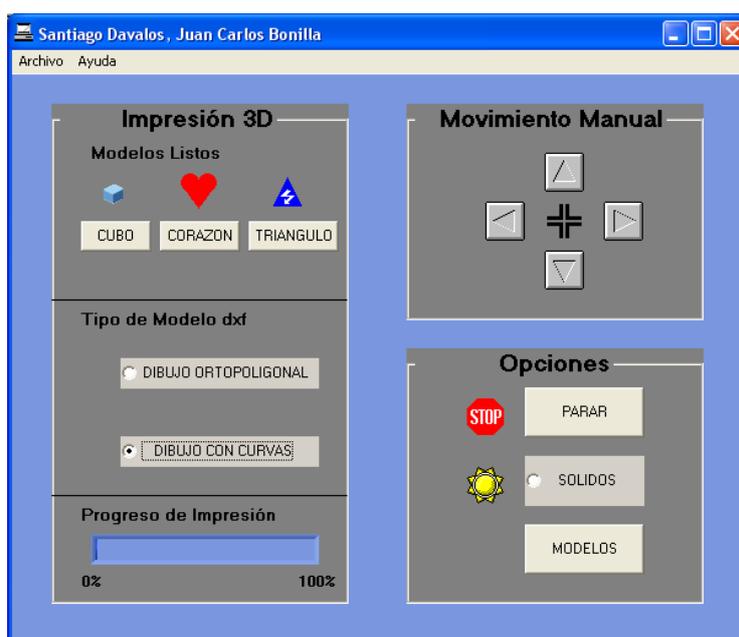


Figura. 4.9. Selección tipo de dibujo

#### 4.2.5 Modelo Prediseñado

El HMI consta con modelos prediseñados que reducen el tiempo de impresión.

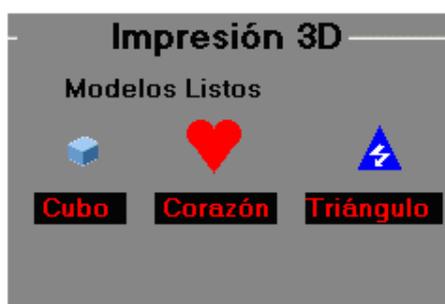


Figura. 4.10. Modelo prediseñado

### 4.2.6 Impresión del Modelo

Luego de configurar la impresión siguiendo los pasos anteriores el modelo esta listo para ser impreso.

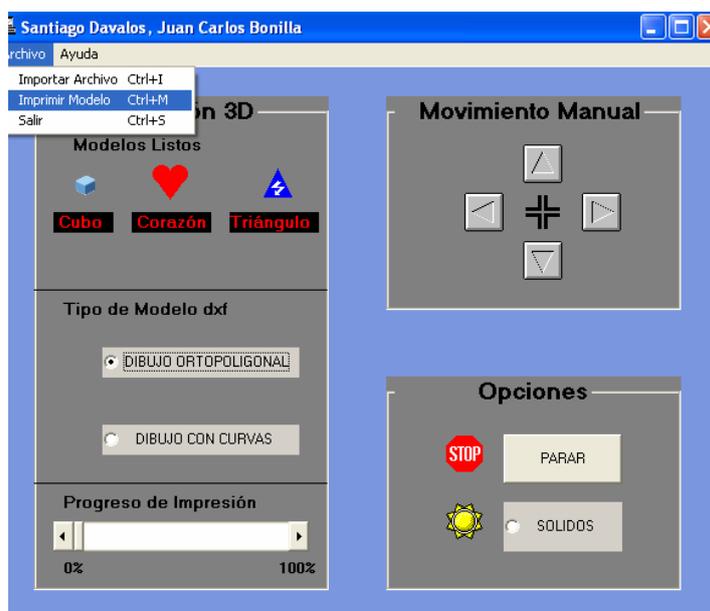


Figura. 4.11. Impresión

Se elige el archivo \*.txt previamente generado por el convertidor.

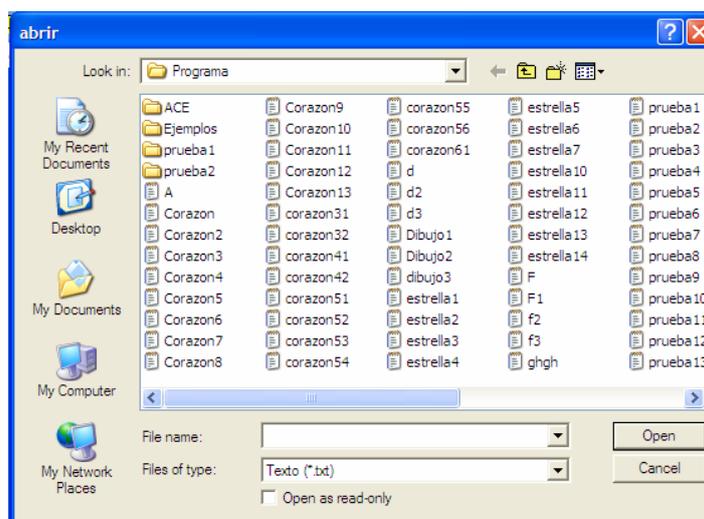


Figura. 4.12. Selección de archivo txt

El HMI despliega un mensaje indicando que se va iniciar la impresión.

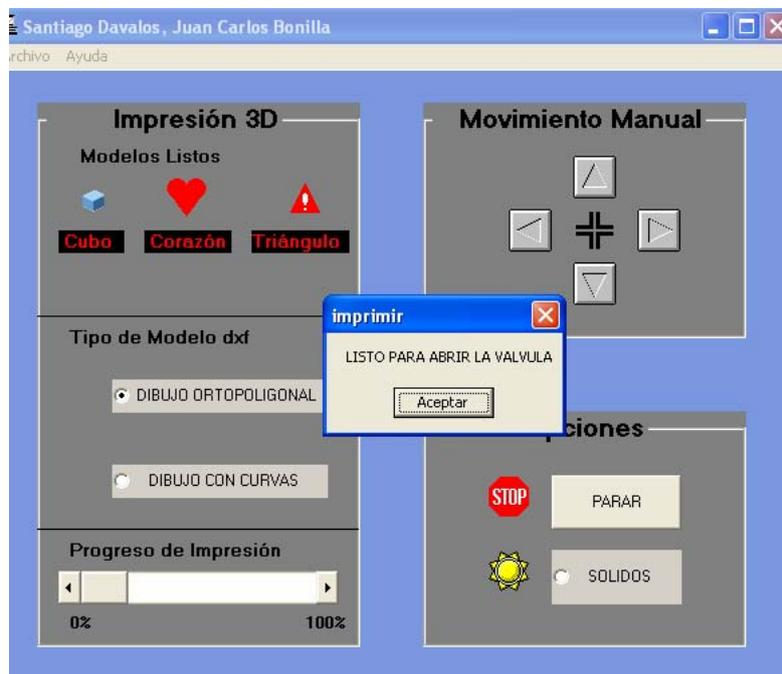


Figura. 4.13. Impresión

Una opción importante que se presenta en el software de la impresora, es la de SÓLIDOS, ésta permite rellenar el modelo, es decir, al seleccionar esta opción las partes que se imprimirán serán sólidas.

Finalmente en el momento de la impresión existe la opción "Parar Impresión" la cual la detiene.

## CAPÍTULO V

### IMPLEMENTACIÓN

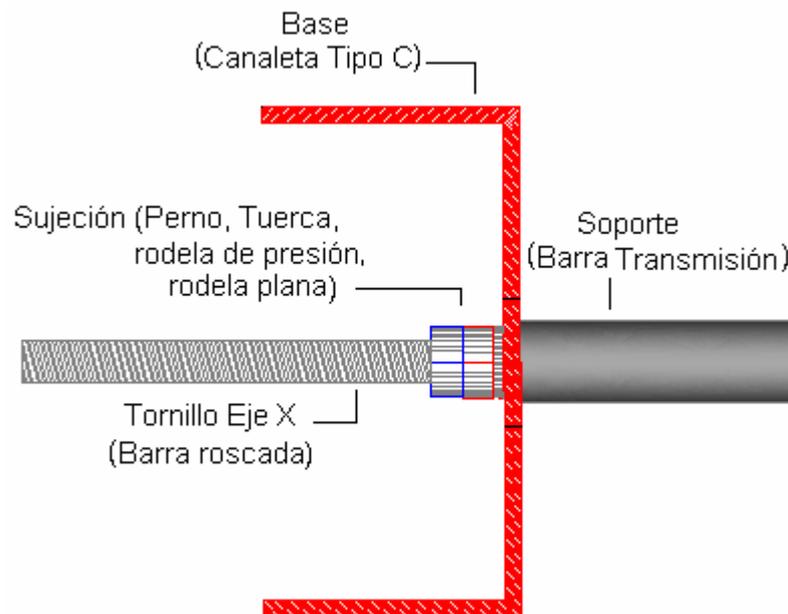
#### 5.1 Implementación robot cartesiano

##### 5.1.1 Fase 1

A continuación se explica paso a paso la primera fase:

- Se colocó las bases, que son dos canaletas tipo C.
- Luego, dos cilindros soportes, conocidos también como ejes de transmisión, estos son de acero, sólidos y lisos en su exterior.
- Para sujetar los soportes a la base, se hizo dos agujeros, uno en cada extremo del eje de transmisión, se ajustó con una rodela plana, una rodela de presión, una tuerca y un perno, en ese orden, de afuera hacia adentro, como se observa en la figura 5.1 y en el plano #1 de Anexo 6.

- Además se instaló el primer tornillo (barra roscada) en el medio de los dos soportes, éste es para el movimiento del eje X.



**Figura. 5.1. Esquema de Implementación – Fase 1**

- Para su sujeción del tornillo a la base, se hizo un agujero en ésta última del mismo diámetro del tornillo y se soldó un rodamiento, colocando luego una rodela plana y una bincha metálica, lo cual permite que el tornillo gire en cualquier sentido sin salirse.
- También se acopló la base, tubo tipo T, que se deslizará sobre el tornillo del eje X y los dos soportes, al mismo tiempo ésta llevará el tornillo del eje Y y transportará el dispensador de cera.
- A esta tercera base se la colocó con bocines hechos a medida tanto para los ejes de transmisión como para el tornillo. (Plano #1 de Anexo 6)



**Figura. 5.2. Canaleta tipo C**



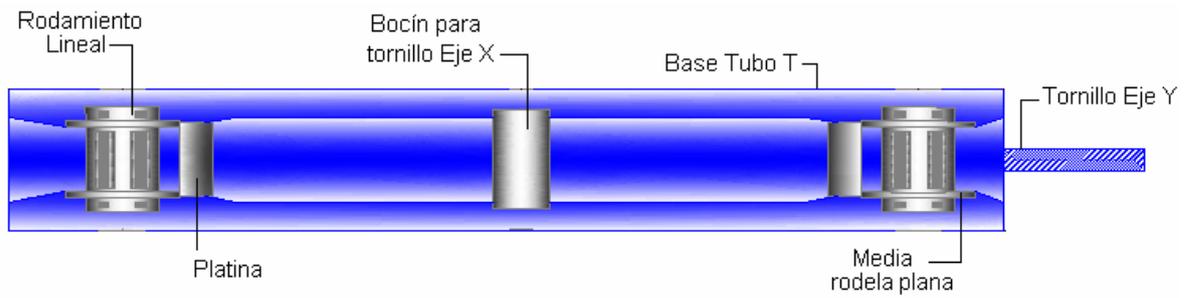
**Figura. 5.3. Ejes de Transmisión y Barra Roscada eje X**



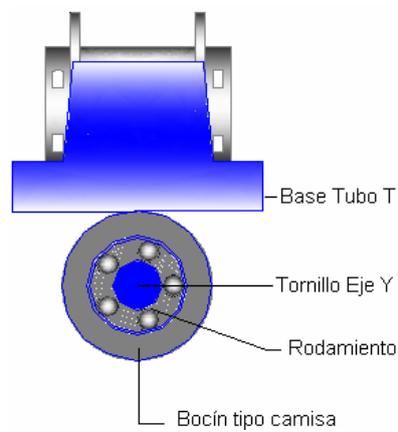
**Figura. 5.4. Ejes de Transmisión y Canaleta tipo C**

### 5.1.2 Fase 2

- Se instaló el segundo tornillo para el eje Y en el tubo tipo T. Para acoplar el tornillo se introdujo en dos rodamientos en cada extremo y estos a su vez encajados en dos bocines tipo camisa soldados al tubo T.
- Se cambió los bocines de la tercera base de las barras de transmisión por problemas de rozamiento y se optó por rodamientos lineales, que fueron sujetos con media rodela plana y una platina cada uno, además reforzados con suelda. El bocín por donde pasa el tornillo se hizo con una forma roscada en su interior, para que la base del eje Y se mueva junto con el giro del tornillo. (Figuras 5.4, 5.5 y plano # 2 de Anexo 6).



**Figura. 5.5. Esquema de Implementación – Fase 2**



**Figura. 5.6. Esquema de Implementación – Fase 2**



**Figura. 5.7. Vista Superior -Tubo tipo T**



Figura. 5.8. Rodamiento lineal tipo bocín

### 5.1.3 Fase 3

- En esta fase se acopló la fase 1 y la fase 2 con los respectivos cambios y correcciones mecánicas que necesitaba el robot, por ejemplo lubricante para el deslizamiento y limpieza. (Plano # 3 de Anexo 6)

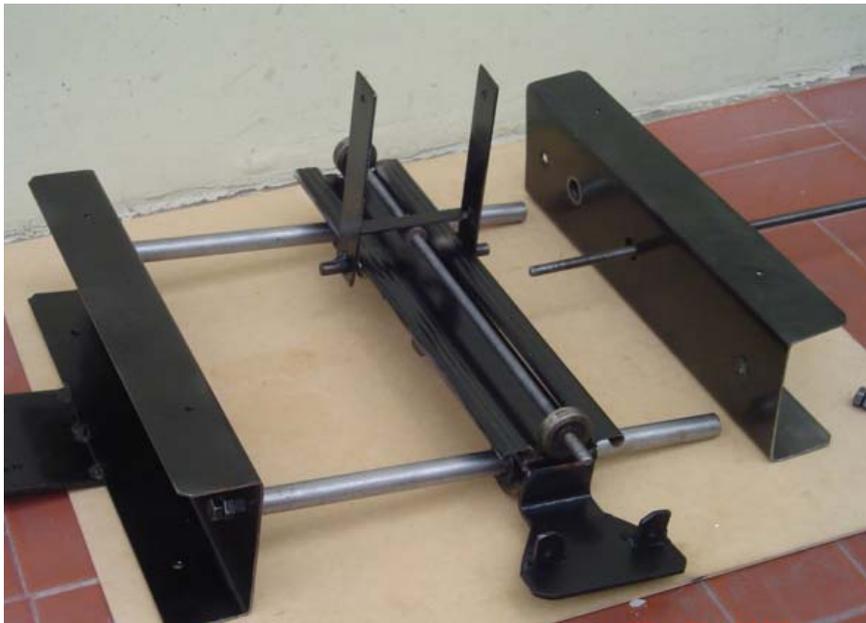


Figura. 5.9. Acoplamiento Fase1 y Fase2

### 5.1.4 Fase 4

- Aquí se construyó la estructura que sostendrá el robot a una altura específica del suelo en cuyo espacio estará la plataforma del eje Z donde se construirá el modelo de cera. (Plano # 4 de Anexo 6)



Figura. 5.10. Estructura Metálica de Soporte

### 5.1.5 Fase 5

- Se acopló la fase 3 y fase 4. (Plano # 5 de Anexo 6)

### 5.1.6 Fase 6

- Se construyó e instaló las guías y la estructura que constituyen el eje Y, ésta última es la que transportará el dispensador de cera.

- Para el movimiento sobre las guías se utilizó rodamientos sujetos con dos soportes acoplados a un bocín roscado colocado en el tornillo del eje Y. (Plano # 6 de Anexo 6)

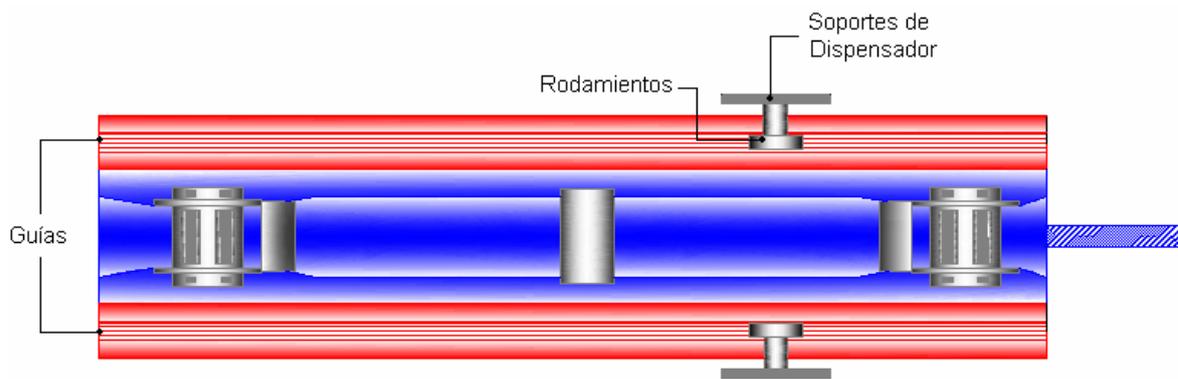


Figura. 5.11. Esquema de Implementación – Fase 6



Figura. 5.12. Estructura del Eje Y



Figura. 5.13. Estructura del Eje Y

### 5.1.7 Fase 7

- Se acopló la fase 6 con toda la estructura y se hacen los ajustes correspondientes. (Plano # 7 de Anexo 6)



Figura. 5.14. Fase 7

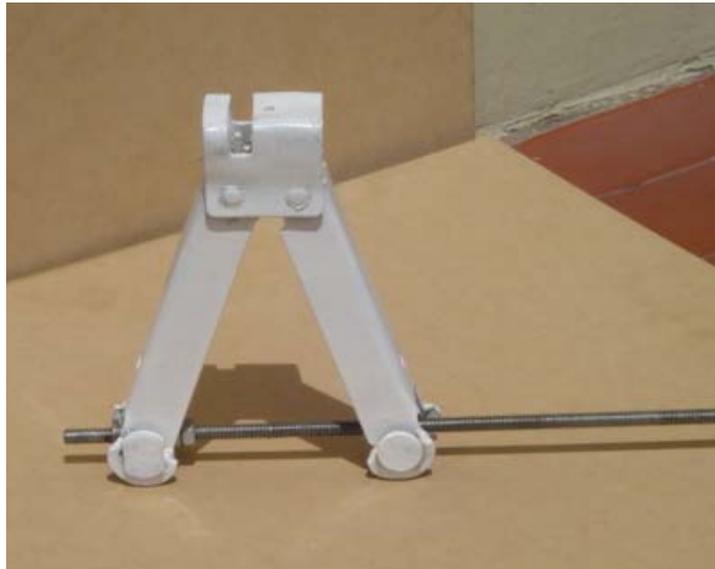
### 5.1.8 Fase 8

- En ésta fase se colocó los tres motores X, Y y Z uno para cada eje respectivamente, su acople fue mediante bocines roscados los cuales están sujetos tanto al eje como al tornillo con prisioneros (tornillos sin cabeza), para su sujeción se utilizaron platinas a la medida para cada motor.



**Figura. 5.15. Motores de los ejes X,Y,Z**

- También se instaló el eje Z con el mecanismo de los brazos de accionamiento de una gata mecánica, con los respectivos cambios, igualmente se utilizó una barra roscada (tornillo sin fin) para el movimiento de este eje.



**Figura. 5.16. Brazos de Accionamiento gata mecánica y barra rosada del eje Z**

- Para el movimiento de la plataforma de arriba hacia abajo o viceversa se colocaron guías en las cuatros esquinas de la estructura sujetando a la placa con ángulos de 90 grados manteniendo el paralelismo que se necesita. (Plano # 8 de Anexo 6)



**Figura. 5.17. Rieles para el eje Z**

## 5.2 Implementación sistema dispensador de cera

- Para el sistema dispensador de cera se utilizó un embudo de aluminio ya que este material es buen conductor de calor, el diámetro menor del embudo fue de 3mm para que se obtenga un acabado fino el momento de inyectar parafina. El embudo sirvió como deposito de parafina. Se utilizó un tubo de PVC para el soporte del embudo



Figura. 5.18. Partes del Sistema Dispensador de cera



Figura. 5.19. Sistema dispensador de cera

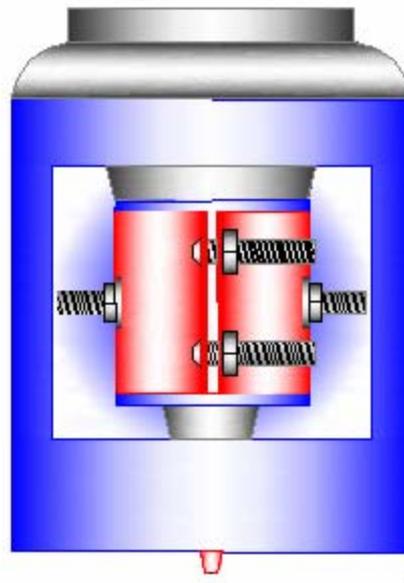


Figura. 5.20. Sistema Dispensador de cera

- Para el cierre y apertura del Dispensador se utilizó el sistema piñón cremallera, que consiste en un engrane conectado al eje del motor el cual está unido a una cremallera de aluminio, este sistema proporcionó un movimiento lineal a través de un movimiento circular que es generado por el motor. En la parte superior de la cremallera se colocó una tapa metálica del mismo grosor del diámetro menor del embudo de aluminio.

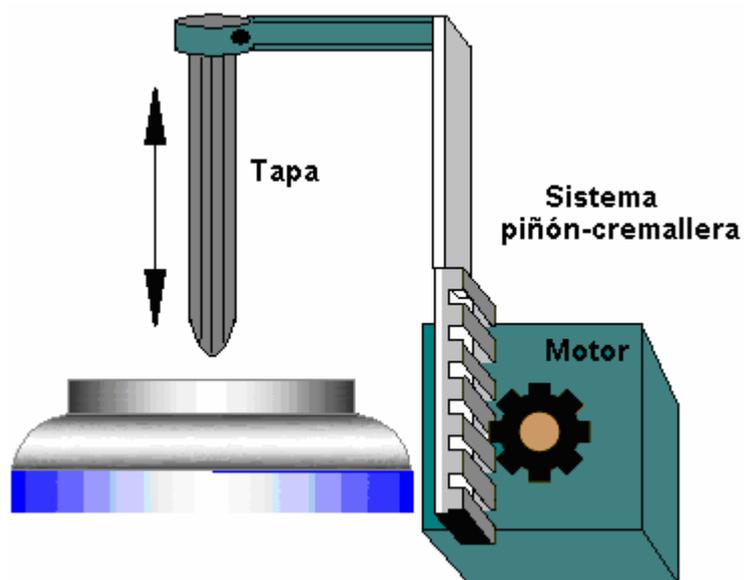


Figura. 5.21. Sistema de Apertura y Cierre del Dispensador

### 5.3 Implementación del hardware de control y potencia

Luego del diseño y pruebas del circuito electrónico en protoboard y tras las verificaciones respectivas se procedió a la realización del circuito impreso con el cual se reduce ruido y pérdida de energía, además facilita la manipulación. En las figuras 5.22 y 5.23 se puede observar los circuitos tanto de control como de potencia respectivamente, que se implementarán en placas.

Las características de los elementos ha emplearse en los circuitos de control y de potencia se detallan en las tabla 5.1 y tabla 5.2.

**Tabla. 5.1. Elementos del Circuito de Control**

<b>ELEMENTO</b>	<b>NUMERACION</b>	<b>CARACTERISTICAS</b>	<b>CANTIDAD</b>
Conector	DB25 - Macho	Para conexión con el puerto paralelo del PC.	1
Buffer	LM74244	Circuito integrado de protección para el puerto paralelo del PC.	2
Microcontrolador	16F877	Microcontrolador o conocido como PIC.	2
Cristal	4 MHz.	Reloj de los microcontroladores.	1
Drive	ULN2803	Circuito integrado amplificador de corriente. (Hasta 500mA.)	2

Tabla. 5.2. Elementos del Circuito de Potencia

ELEMENTO	NUMERACION	CARACTERISTICAS	CANTIDAD
Opto-acoplador	PC817	Acoplamiento electrónico entre dos circuitos diferentes.	12
Amplificador	D1889	Transistor amplificador de corriente. Amplificación máxima 12 A.	12
Resistencias	220 $\Omega$	Resistencias, elementos pasivos, protecciones.	24
Resistencias	5 K $\Omega$	Resistencias, elementos pasivos, protecciones.	12

En la tabla 5.3 se detalla las características de los motores a pasos a utilizarse en el robot cartesiano.<sup>34</sup>

Tabla. 5.3. Características de Motores a paso

<b>Motor</b>	Motor a Pasos
<b>Tipo</b>	Unipolar
<b>Número de Bobinas</b>	4
<b>Grados por paso</b>	2.5°/paso
<b>Voltaje de alimentación</b>	4.5 – 5 [V] DC
<b>Corriente</b>	1 [A]

El voltaje utilizado para la alimentación del circuito de control es de 5 voltios DC., ya que los circuitos integrados y motores de pasos empleados se polarizan con éste valor de voltaje.

<sup>34</sup> Figuras y dimensiones de los motores se detalla en el capítulo II.

El esquema de conexiones del circuito de control implementado se presenta en la figura 5.22.

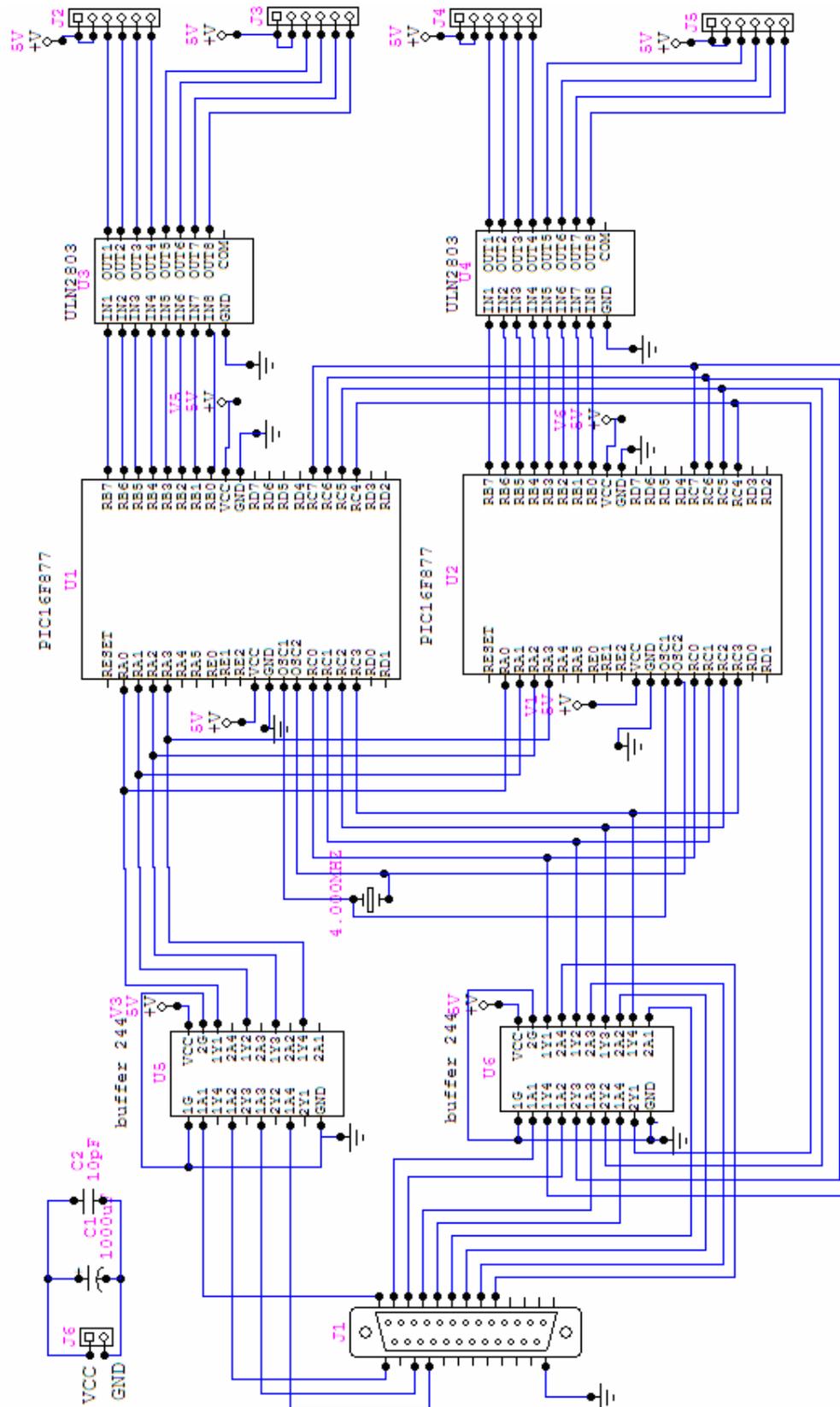


Figura. 5.22: E esquema de conexión del Hardware de Control

El esquema de conexiones del circuito de potencia implementado se presenta en la figura 5.23.

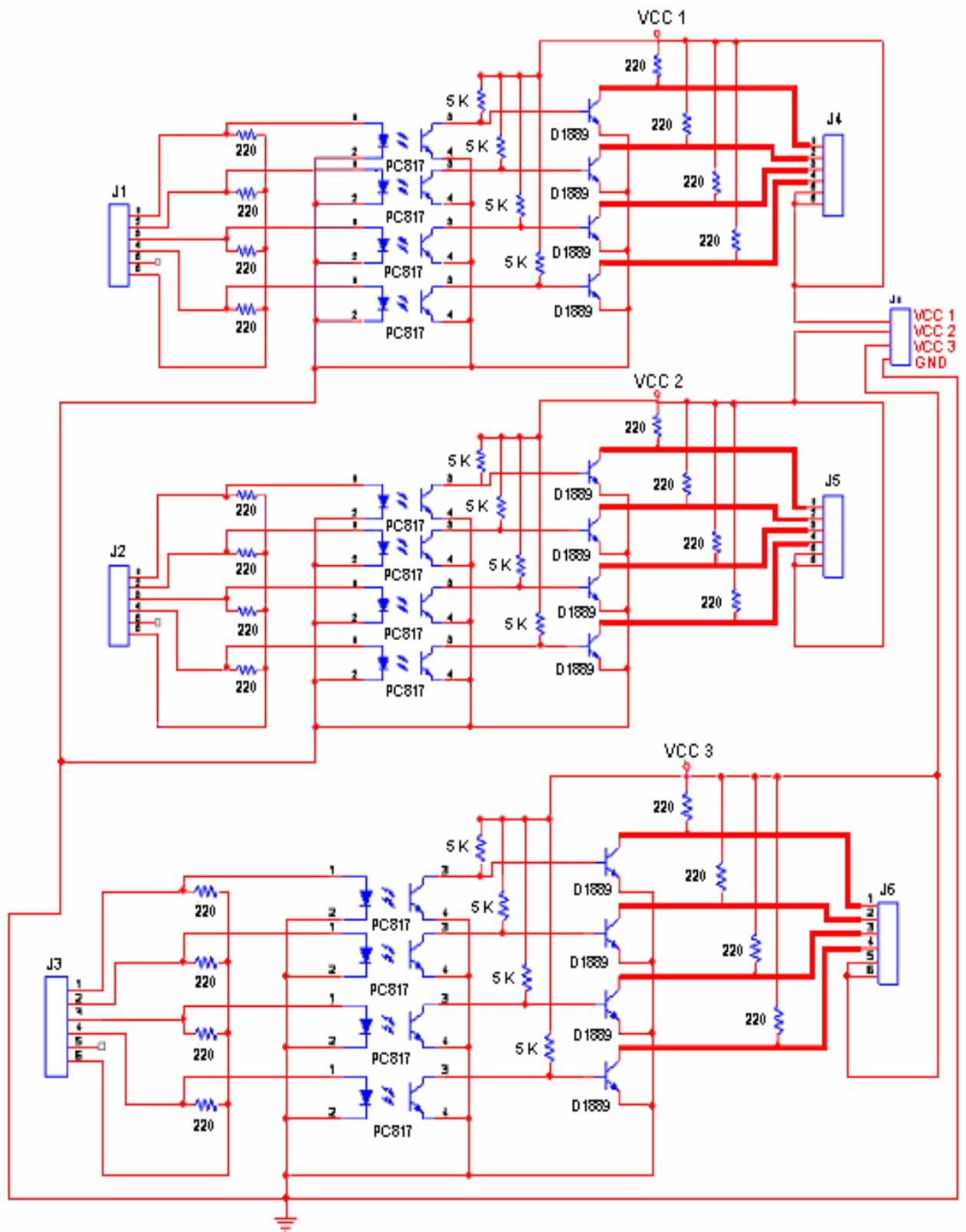


Figura. 5.23. Esquema de conexión del Hardware de Potencia

- Para la implementación del circuito impreso se lo realizó en el software de diseño de circuitos impresos PROTEL, el cual permite un ruteado adecuado y aprovechamiento máximo del espacio.



**Figura. 5.24. Circuitos Impresos en PROTEL**

- Para la alimentación del circuito electrónico se utilizó una fuente de computadora, que entrega 5v y un alto amperaje que es requerido para el funcionamiento del circuito y la parte de potencia.



**Figura. 5.25. Fuente**

#### 5.4 Prototipo de Impresora para Modelos 3D en cera - Final



Figura. 5.26. Impresora para modelos 3D en cera (1)



Figura. 5.27. Impresora para modelos 3D en cera (2)



Figura. 5.28. Impresora para modelos 3D en cera (3)



Figura. 5.29. Impresora para modelos 3D en cera (4)

## CAPÍTULO VI

### PRUEBAS Y RESULTADOS

#### 6.1 Pruebas del sistema completo

Luego de finalizada la construcción de la parte mecánica, etapa de control y de potencia de la impresora 3D, se procederá a realizar las pruebas necesarias para comprobar el funcionamiento correcto de ésta.

Para las pruebas se siguieron estos pasos:

Software:

- Comprobación de conversión de coordenadas
- Envío correcto de datos de control desde el PC hacia la etapa de control electrónico.

Hardware:

Etapa de Control de motores

- Comprobación de secuencias para el movimiento de motores
- Comprobación de velocidades de los motores

Etapa de Control de temperatura

- Verificación de los límites de temperatura del controlador
- Comprobación del funcionamiento del lazo de histéresis del control

Etapa de Potencia

- Medición de la corriente necesaria para el movimiento óptimo de los motores

Pruebas

- Primeros dibujos en 2D utilizando un marcador como herramienta final
- Primeros dibujos en 2D sólidos utilizando un marcador como herramienta final
- Modelos en 2D utilizando el dispensador de cera sin control de temperatura.
- Modelos en 2D utilizando el dispensador de cera con el control de temperatura.
- Modelos en 2D sólidos utilizando el dispensador de cera con el control de temperatura.
- Primeros modelos en 3D con la impresora completa
- Modelos finales en 3D con el sistema completo

## 6.2 Resultados

Los resultados obtenidos en las pruebas realizadas se detallan a continuación:

Software:

- Comprobación de conversión de coordenadas
  - Realización del modelo en software CAD (\*.dxf)

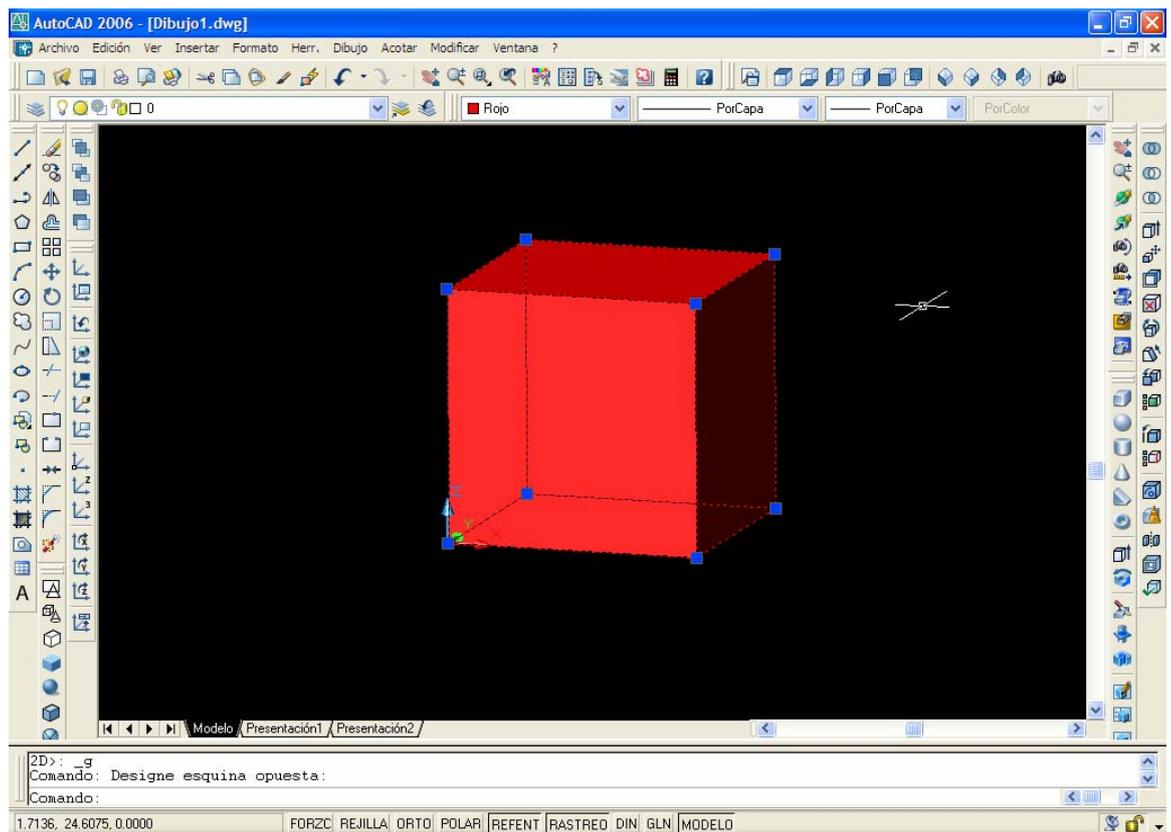


Figura. 6.1. Archivo dxf

- o Conversión mediante software ACE

```
Dibujo1 - Notepad
File Edit Format View Help
G01 X5.0000 Y5.0000 Z2.0000
G01 X5.0000 Y5.0000 Z1.0000
G01 X5.0000 Y0.0000 Z1.0000
G01 X5.0000 Y0.0000 Z2.0000
G01 X5.0000 Y0.0000 Z1.0000
G01 X0.0000 Y0.0000 Z1.0000
G01 X5.0000 Y0.0000 Z1.0000
G01 X5.0000 Y5.0000 Z1.0000
G01 X0.0000 Y5.0000 Z1.0000
G01 X0.0000 Y0.0000 Z1.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X5.0000 Y0.0000 Z2.0000
G01 X5.0000 Y5.0000 Z2.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X5.0000 Y5.0000 Z2.0000
G01 X5.0000 Y5.0000 Z3.0000
G01 X5.0000 Y5.0000 Z2.0000
G01 X5.0000 Y0.0000 Z2.0000
G01 X5.0000 Y0.0000 Z3.0000
G01 X5.0000 Y0.0000 Z2.0000
G01 X5.0000 Y5.0000 Z2.0000
G01 X5.0000 Y5.0000 Z3.0000
G01 X5.0000 Y5.0000 Z2.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X0.0000 Y5.0000 Z3.0000
G01 X0.0000 Y5.0000 Z2.0000
G01 X0.0000 Y0.0000 Z3.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y0.0000 Z3.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X5.0000 Y0.0000 Z2.0000
G01 X5.0000 Y0.0000 Z3.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y0.0000 Z3.0000
G01 X0.0000 Y0.0000 Z2.0000
G01 X0.0000 Y0.0000 Z3.0000
G01 X0.0000 Y0.0000 Z4.0000
G01 X0.0000 Y0.0000 Z3.0000
G01 X0.0000 Y5.0000 Z3.0000
G01 X0.0000 Y5.0000 Z4.0000
G01 X0.0000 Y5.0000 Z3.0000
G01 X5.0000 Y5.0000 Z3.0000
G01 X5.0000 Y5.0000 Z4.0000
```

Figura. 6.2. Código G

- Envío correcto de datos de control desde el PC hacia la etapa de control electrónico.

Mediante el puerto paralelo y leds en la salida del cable DB25 se comprobó que el puerto de control y el puerto de datos, realizaba correctamente el envío de datos a través del software previamente desarrollado.

Hardware:

Etapa de Control de motores

- Comprobación de secuencias y velocidades para el movimiento de motores
- Verificación del funcionamiento del circuito de control



Figura. 6.3. Circuito de control de motores

Etapa de Control de temperatura

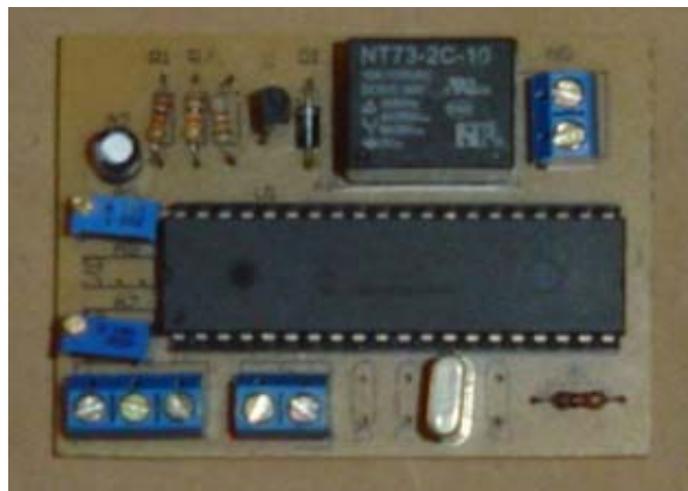


Figura. 6.4. Circuito de control de temperatura

## Etapa de Potencia

- Medición de la corriente necesaria para el movimiento óptimo de los motores



Figura. 6.5. Circuito de Potencia y motores

- Primeros dibujos en 2D utilizando un marcador como herramienta final

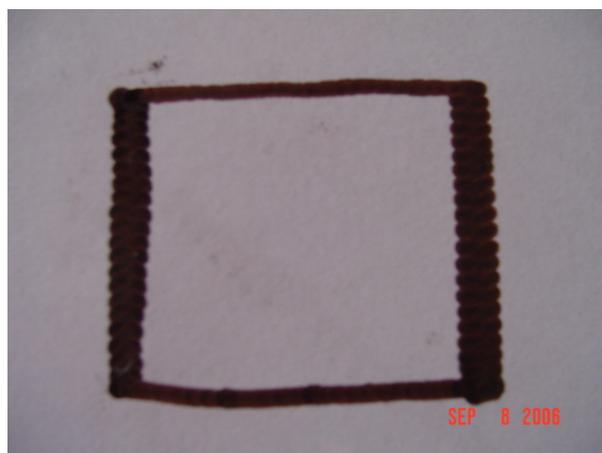


Figura. 6.6. Cuadrado de 5cm. por lado



Figura. 6.7. Corazón

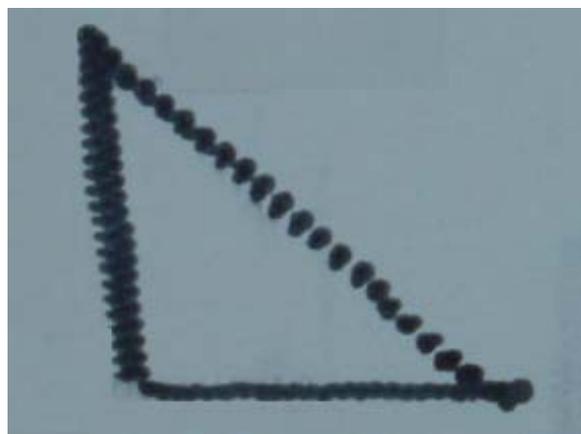


Figura. 6.8. Triángulo rectángulo

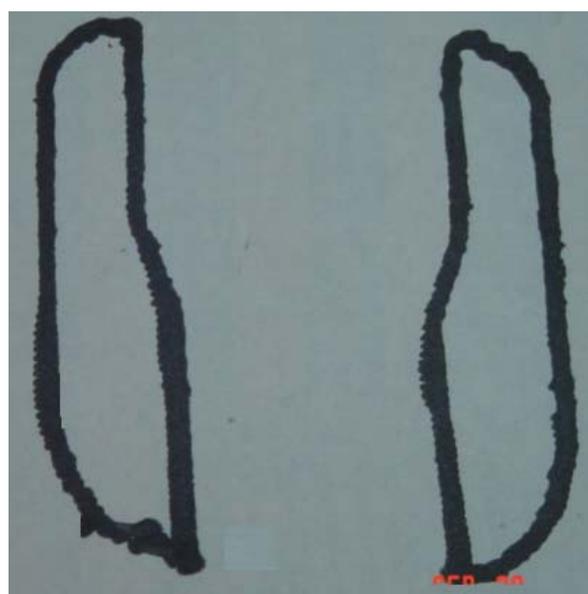


Figura. 6.9. Partes de un Mouse para PC

- Primeros modelos en 2D sólidos utilizando un marcador como herramienta final

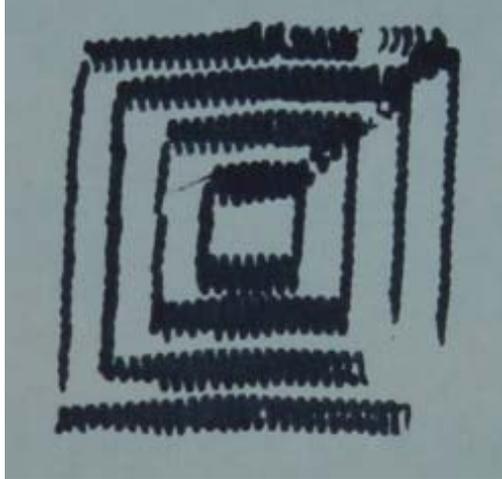


Figura. 6.10. Cuadrado de 5 cm. - Sólido

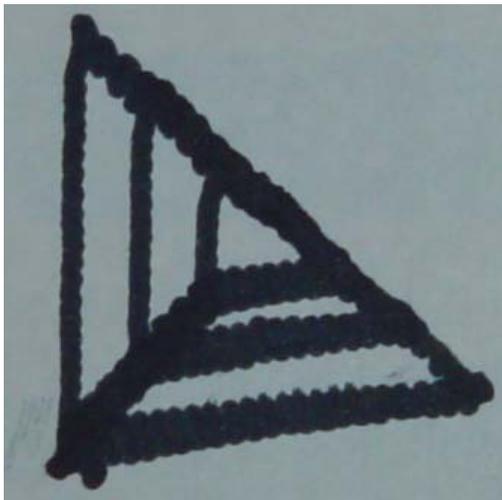


Figura. 6.11. Triángulo Rectángulo - Sólido

### 6.2.1 Resultados con Cera

En la figura 6.12 se puede observar el proceso de impresión.

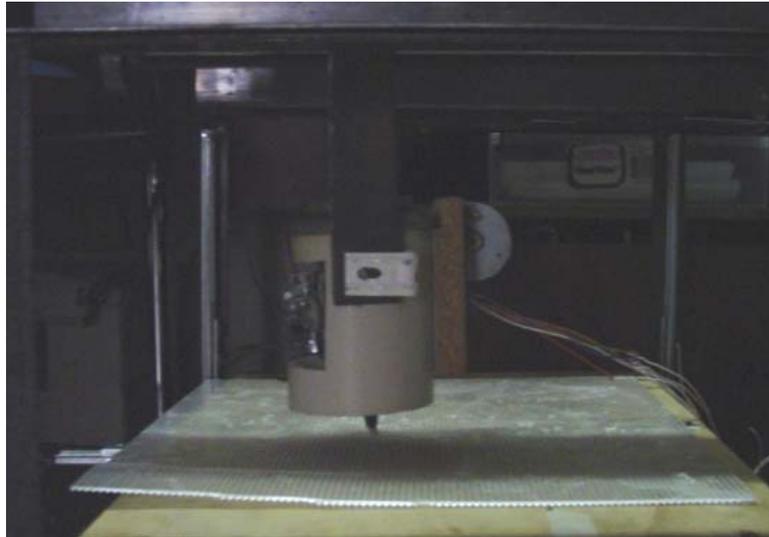


Figura. 6.12. Impresión con cera

- Modelos en 2D utilizando el dispensador de cera sin control de temperatura.

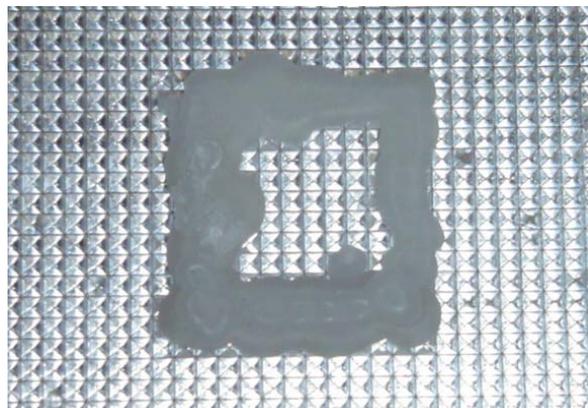


Figura. 6.13. Cuadrado de 5 cm. por lado



**Figura. 6.14. Letra P**

- Modelos en 2D utilizando el dispensador de cera con el control de temperatura.



**Figura. 6.15. Cuadrado de 5 cm. por lado**



**Figura. 6.16. Triángulo Rectángulo**



**Figura. 6.17. Parte izquierda de un Mouse para PC**

- Modelos en 2D sólidos utilizando el dispensador de cera con el control de temperatura.



**Figura. 6.18. Partes por capas de un Mouse para PC**



Figura. 6.19. Partes complementarias de un Mouse para PC

- Modelos finales en 3D con el sistema completo

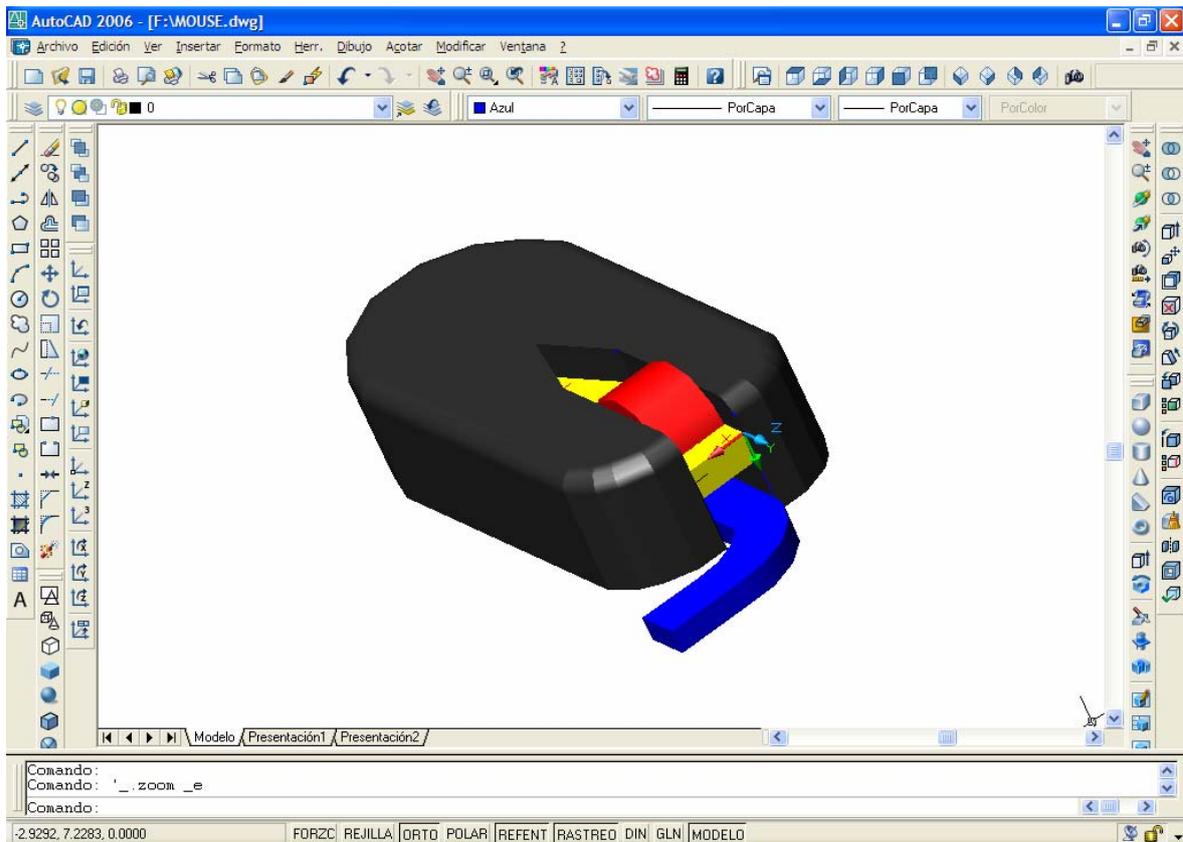


Figura. 6.20. Modelo a imprimirse - Mouse para PC



Figura. 6.21. Modelo 3D impreso



Figura. 6.22. Modelo 3D impreso



**Figura. 6.23. Modelo 3D impreso**



**Figura. 6.24. Modelo 3D impreso**

## CAPÍTULO VII

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1 Conclusiones

- Este sistema de impresión en cera de modelos 3D nos entrega una forma sencilla y sofisticada de visualización de las características de los modelos y requerimientos del cliente que pueden ser de forma o tamaño.
- El diseño de la impresora se basa en el mecanismo de un robot cartesiano, que se implementó tomando en cuenta las características principales de este tipo de robot, compuesto por los 3 ejes X, Y, Z.
- Para el sistema dispensador de cera fue necesario tener un control de temperatura seguro, con un rango de variación de  $\pm 3$  °C, con lo se consiguió una inyección regular de parafina en cuanto al flujo.

- 
- En el hardware de control de motores se utilizó microcontroladores, con los requerimientos necesarios para el cumplimiento del diseño planteado de la impresora, como son: el número de puertos disponibles, velocidad de procesamiento de la información y perspectiva de programación.
  - El software convertidor de archivos dxf a código G es utilizado para maquinas CNC, en el caso de la impresora 3D solo se utiliza como convertidor de coordenadas las cuales son procesadas y luego utilizadas para los movimientos del robot cartesiano.
  - Para el software de control del robot se optó por dividir al modelo Pre-dibujado en diferentes entidades, siendo cada línea o curva una entidad que posee coordenadas de origen y final, aquellas que permiten calcular la velocidad y dirección de los motores.
  - Al modelo dibujado en 3D se dividió en varias capas de acuerdo a la altura de la parafina inyectada, lo cual permitió trabajar al modelo 3D en capas planares.

## 7.2 Recomendaciones

- Para el diseño del hardware de control de los motores se debe tomar en cuenta la corriente máxima que estos consumen; siendo necesario la implementación de una etapa de amplificación.
- En la parte de ensamblaje del robot cartesiano es necesario tomar en cuenta el factor de rozamiento que se produce entre elementos mecánicos, con la utilización de rodamientos y lubricación se reduce al mínimo este factor.
- Para el calentamiento de la parafina es recomendable utilizar materiales como el aluminio que permitan mantener la temperatura.
- En el dispensador de cera se sugiere que la parte final de la herramienta sea de un radio pequeño para conseguir un acabado de los modelos fino y estilizado.
- En cuanto a la plataforma donde se irá construyendo el modelo, es necesario que ésta tenga una superficie antideslizante, con esto la parafina no se esparcirá y el modelo será más preciso, se recomienda utilizar acrílico con las características anteriores.

- Para la construcción del robot cartesiano es aconsejable utilizar motores a pasos, por la precisión que se puede obtener de estos, además es necesario que posean las mismas características eléctricas en cuanto grado por paso y corriente.
  
- Si se busca mayor precisión en los modelos se sugiere el manejo de una secuencia de medio paso para los motores sacrificando torque y aumentando el tiempo de impresión.

## BIBLIOGRAFÍA

### TEXTOS

- REHG, James, *Introduction to Robotics in CIM Systems*, Tercera Edición, Prentice-Hall Inc., Upper Saddle River-New Jersey -1997.
- BOYLESTAD, Robert L., NASHELSKY, Louis, *Electrónica: Teoría de Circuitos*, Sexta Edición, Prentice-Hall Inc., México -1997.
- HALVORSON, Michael, *Aprenda Visual Basic 6.0*, Quinta Edición, 2002.
- DORF, Richard, *Circuitos Eléctricos – Introducción al Análisis y Diseño*, Tercera Edición, Alfaomega, México – 2000.

### FOLLETOS Y PUBLICACIONES MENORES

- GALINDO, Magí, *Qué es el Prototipado Rápido o "Rapid Prototyping?",* Responsable del Área de prototipo rápido del centro CIM.
- WOHLERS, Terry, *Rapid Prototyping & Tooling State of the Industry*, Worldwide Progress Reports -1998,1999, 2000, 2001, 2002.
- GRIFFITH, Michelle, *Integration of rapid prototyping into design and manufacturing*, Sandia National Laboratories - Penn State.

- S.O Onih University of Wales Collage, New port, UK, K.K.B Hon The University of Liverpool,Liverpool, UK, *Integration of rapid prototyping technology into FMS for agile manufacturing.*
- NARANJO, Gerardo, *Técnicas de manufactura y el camino hacia la manufactura de clase mundial.*
- ZAIMOVIC-UZUNOVIC, Nermina, *What type of rp model will be chosen for different kind of products,* University of Sarajevo, Bosnia.
- EGÜEZ, Mariano, *Ecoeficiencia: Funciones de apoyo en el “LEMA”* ,Laboratorio de energía y medio ambiente - Facultad de Ingeniería, Universidad Diego Portales, Diciembre 2000.
- CONTRERAS, Patricia Eugenia, *“Puesta en marcha y aplicaciones de una máquina de desarrollo rápido de prototipos con sistema de moldeado por deposición de hilo fundido”,* Departamento de Mecánica, Universidad Técnica Federico Santa María, Noviembre 2001.
- SCHONBERGER, Richard J., *Manufactura de Categoría Mundial,* Norma, Biblioteca Facultad de Ingeniería Universidad Diego Portales.

#### PÁGINAS WEB - INTERNET

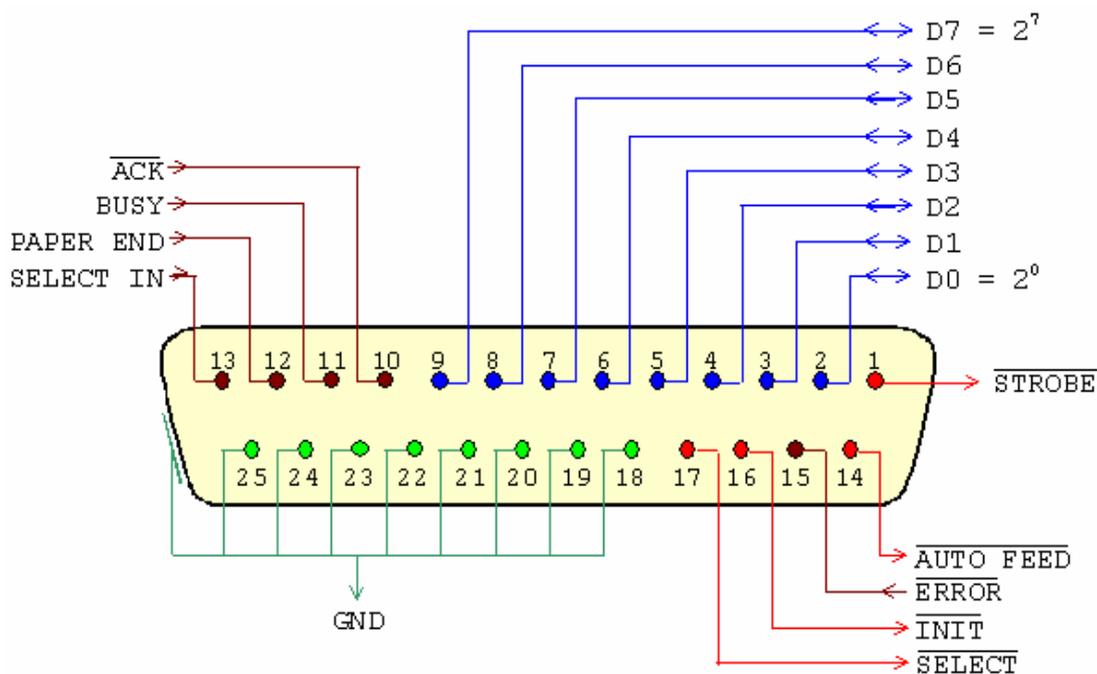
- <http://www.weissdesign.com/rp1>.
- <http://www.rapidpro.com/files/whatis.htm>, *Prototipado rápido- 3D.*
- <http://www.pitt.edu/~roztock/rapidpro/rapidpro.htm>, *Prototipado rápido*
- <http://www-rpl.stanford.edu>
- [http:// www.eos.info](http://www.eos.info)

- 
- [http:// www.phenix-systems.com](http://www.phenix-systems.com)
  - <http:// www.cubicttechnologies.com>
  - <http://www.fondef.cl/bases/fondef/PROYECTO/96/F/D96F1029.HTML>
  - <http://www.solid-scape.com>, *Impresoras 3D*.
  - <http://www.udp.cl>
  - <http://www.finam.cl/>
  - <http://www.microchip.com>, *Hoja técnica del Microcontrolador PIC16f877*.
  - <http://www.micropic.arrakis.es/marcos.htm>
  - <http://www.unicrom.com/Tutoriales.asp>
  - <http://www.infovis.net>, *Visualización de la información*.
  - <http://www.3dsystems.com/>, *Impresoras 3D*.
  - <http://www.stratasys.com/>, *Impresoras 3D*.
  - <http://web.mit.edu/afs/athena.mit.edu/org/t/tdp/www/index.html>
  - <http://www.biomodel.com/>, *Impresoras 3D*.
  - [http://www.msoe.edu/reu/Biomed\\_abstracts.shtml](http://www.msoe.edu/reu/Biomed_abstracts.shtml)

## **ANEXOS**

## **ANEXO 1**

### **Puerto Paralelo – Descripción de Pines de Conector**



Anexo 1-Figura. 1. Distribución de pines conector DB25

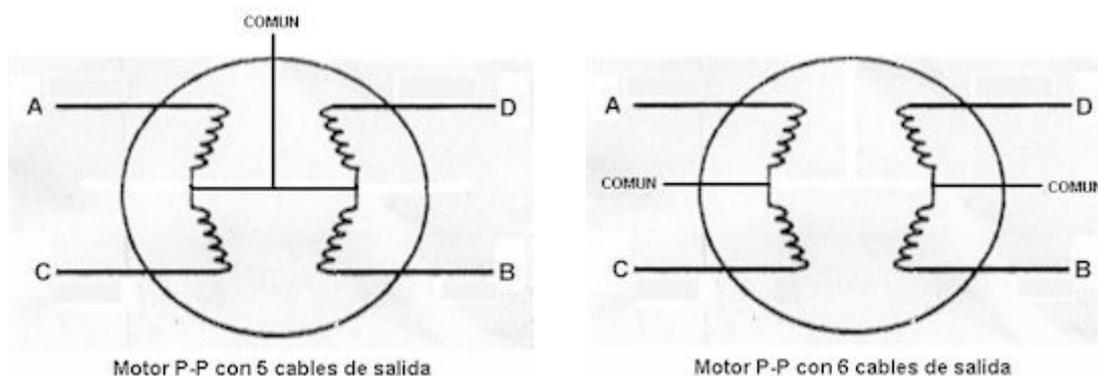
Descripción de los pines (Anexo 1- Figura 1):

- 8 líneas (pines) son para salida de datos (bits de DATOS). Sus valores son únicamente modificables a través de software, y van del pin 2 (dato 0, D0) al pin 9 (dato 7, D7).
- 5 líneas son de entrada de datos (bits de ESTADO), únicamente modificables a través del hardware externo. Estos pines son: 11, 10, 12, 13 y 15, del más al menos significativo.
- 4 líneas son de control (bits de CONTROL), numerados del más significativo al menos: 17, 16, 14 y 1. Habitualmente son salidas, aunque se pueden utilizar también como entradas y, por tanto, se pueden modificar tanto por software como por hardware.
- Las líneas de la 18 a la 25 son la tierra.

## **ANEXO 2**

### **Distribución de Bobinas de un motor de pasos**

Es posible averiguar la distribución de los cables a los bobinados y el cable común en un motor de paso unipolar de 5 o 6 cables siguiendo las instrucciones que se detallan a continuación:



**Anexo 2 – Figura. 1. Estator de 4 bobinas de un motor a pasos**

1. Aislando el cable(s) común que va a la fuente de alimentación:

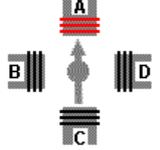
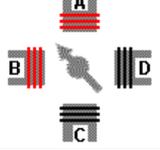
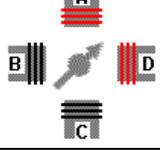
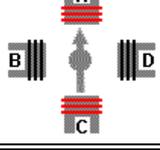
- En el caso de motores con 6 cables, estos poseen dos cables comunes, pero generalmente poseen el mismo color, por lo que lo mejor es unirlos antes de comenzar las pruebas.
- Usando un tester para chequear la resistencia entre pares de cables, el cable común será el único que tenga la mitad del valor de la resistencia entre ella y el resto de los cables.

Esto es debido a que el cable común tiene una bobina entre ella y cualquier otro cable, mientras que cada uno de los otros cables tiene dos bobinas entre ellos. De ahí la mitad de la resistencia medida en el cable común.

2. Identificando los cables de las bobinas (A, B, C y D):

- Aplicar un voltaje al cable común (generalmente 12 volts, pero puede ser más o menos) y manteniendo uno de los otros cables a tierra (GND) mientras vamos poniendo a tierra cada uno de los demás cables de forma alternada y observando los resultados.

El proceso se puede apreciar en el siguiente cuadro:

<p>Seleccionar un cable y conectarlo a tierra. Ese será llamado cable A.</p>	
<p>Manteniendo el cable A conectado a tierra, probar cuál de los tres cables restantes provoca un paso en sentido antihorario al ser conectado también a tierra. Ese será el cable B.</p>	
<p>Manteniendo el cable A conectado a tierra, probar cuál de los dos cables restantes provoca un paso en sentido horario al ser conectado a tierra. Ese será el cable D.</p>	
<p>El último cable debería ser el cable C. Para comprobarlo, basta con conectarlo a tierra, lo que no debería generar movimiento alguno debido a que es la bobina opuesta a la A.</p>	

**Anexo 2 – Tabla. 1. Identificación bobinas**

Nota: La nomenclatura de los cables (A, B, C, D) es totalmente arbitraria.

Recomendaciones:

- Un motor de paso con 5 cables es casi seguro de 4 fases y unipolar.
- Un motor de paso con 6 cables también puede ser de 4 fases y unipolar, pero con 2 cables comunes para alimentación, pueden ser del mismo color.
- Un motor de pasos con solo 4 cables es comúnmente bipolar.

## **ANEXO 3**

**Lenguaje de Programación – Compilador HT- PIC  
Programador IC-Prog**

## **Lenguaje C**

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado.

Este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina concreta. Se le suele llamar lenguaje de programación de sistemas debido a su utilidad para escribir compiladores y sistemas operativos, aunque de igual forma se puede desarrollar cualquier tipo de aplicación.

C, trabaja con tipos de datos que son directamente tratables por el hardware de la mayoría de computadoras actuales, como son los caracteres, números y direcciones. Estos tipos de datos pueden ser manipulados por las operaciones aritméticas que proporcionan las computadoras. No proporciona mecanismos para tratar tipos de datos que no sean los básicos, debiendo ser el programador quien los desarrolle. Esto permite que el código generado sea muy eficiente y de ahí el éxito que ha tenido como lenguaje de desarrollo de sistemas.

No proporciona otros mecanismos de almacenamiento de datos que no sea el estático y no provee mecanismos de entrada/salida. Ello permite que el lenguaje sea reducido y los compiladores de fácil implementación en distintos sistemas. Estas carencias se compensan mediante la inclusión de funciones de librería para realizar todas estas tareas, que normalmente dependen del sistema operativo.

Un programa (aplicación – PIC), en lenguaje C puede ser escrito en un PC utilizando cualquier procesador de palabras o editor en grado de generar archivos de tipo ASCII o archivo de texto; claro está que se deben incluir las librerías necesarias y la inicialización de los puertos a usarse en el PIC.

La primera operación a efectuar es escribir el programa fuente y su memorización en un archivo (file) de texto con extensión **.C**. Para hacer esto, se ha mencionado que es necesario utilizar un editor ASCII es decir, un programa de escritura como por ejemplo el **NOTEPAD.EXE** o **BLOC DE NOTAS** de Windows.

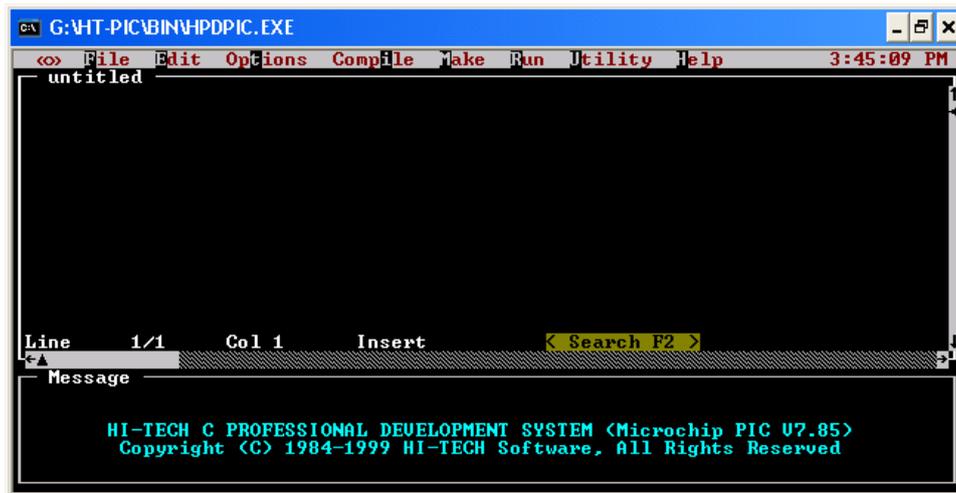
### **Compilador HT-PIC**

Es un compilador que facilita la creación del programa para los diferentes tipos y series de microcontroladores y que además, la programación es muy similar a la del lenguaje C con mínimos cambios.

Luego de tener el programa grabado como archivo con extensión **.c**, es necesario verificar errores y compilarlo para obtener ya el archivo con extensión **.hex** que será el que se grabará directamente en el microcontrolador.

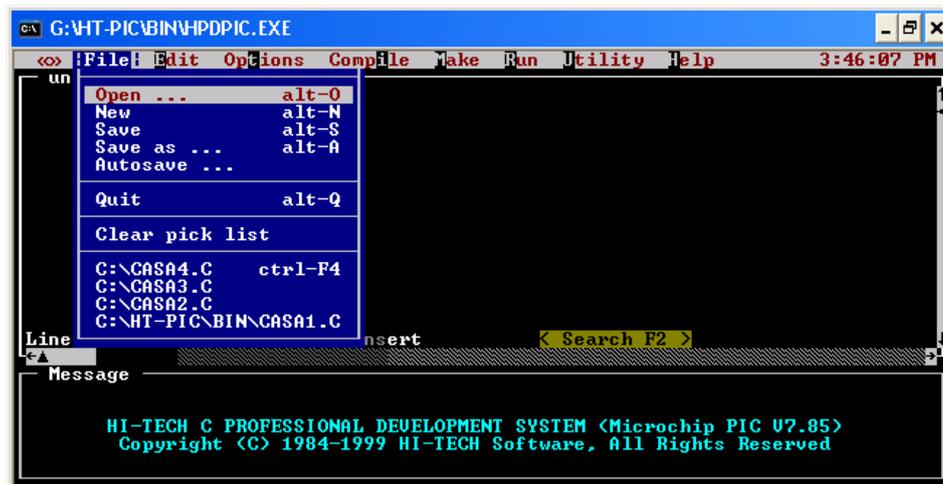
Para realizar este proceso, se realizan los siguientes pasos:

- 1) Ejecutar el software HT-PIC, que se encuentra en el directorio donde se instaló. C:\HT-PIC\BIN\HPDPIC.EXE. (Anexo 3 - Figura 1)

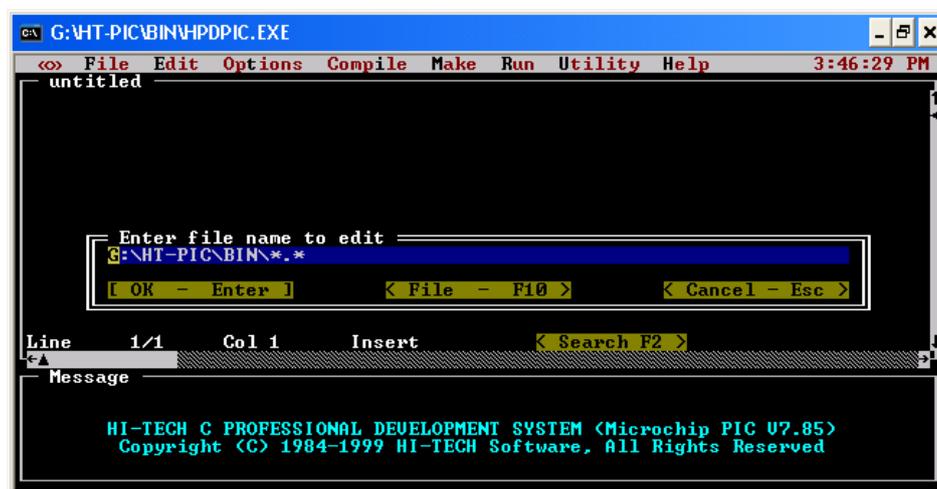


Anexo 3 – Figura. 1. Pantalla Principal de HT-PIC

2) Abrir el archivo con extensión .c. (Anexo 3 - Figuras 2, 3 y 4)



Anexo 3 – Figura. 2. Abrir archivo



Anexo 3 – Figura. 3. Buscar archivo

```

G:\VHT-PIC\BIN\HPPIC.EXE
File Edit Options Compile Make Run Utility Help 3:46:58 PM
MOTORX.C
Programa para movimientos de motor de eje x
Puerto A: entradas, puerto B: Salidas

#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main(void)
<
Line 1/155 Col 1 Insert C mode < Search F2 >
Message
HI-TECH C PROFESSIONAL DEVELOPMENT SYSTEM (Microchip PIC U7.85)
Copyright (C) 1984-1999 HI-TECH Software, All Rights Reserved

```

Anexo 3 – Figura. 4. Cargar archivo

## 3) Compilar archivo:

- Menú Compile, opción Compile and link (Anexo 3 - Figura 5)
- Se selecciona el número de microcontrolador que se va a utilizar. (Anexo 3 - Figura 6)
- Se elige el tipo de datos en cuanto al número de bits. (Anexo 3 - Figura 7)

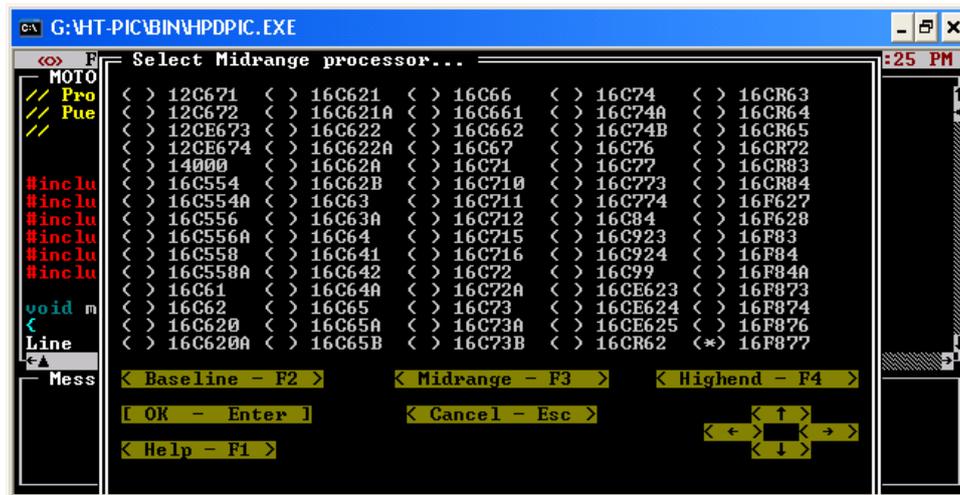
```

G:\VHT-PIC\BIN\HPPIC.EXE
File Edit Options Compile Make Run Utility Help 3:47:09 PM
MOTORX.C
Programa para movinien
Puerto A: entradas, pu
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

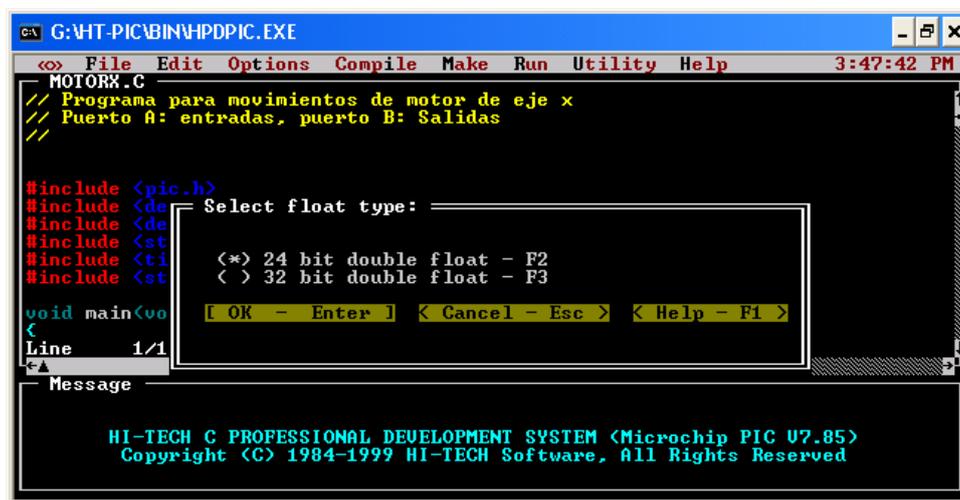
void main(void)
<
Line 1/155 Col 1
Message
HI-TECH C PROFESSIONAL DEVELOPMENT SYSTEM (Microchip PIC U7.85)
Copyright (C) 1984-1999 HI-TECH Software, All Rights Reserved

```

Anexo 3 – Figura. 5. Compilar



Anexo 3 – Figura. 6. Seleccionar Microcontrolador

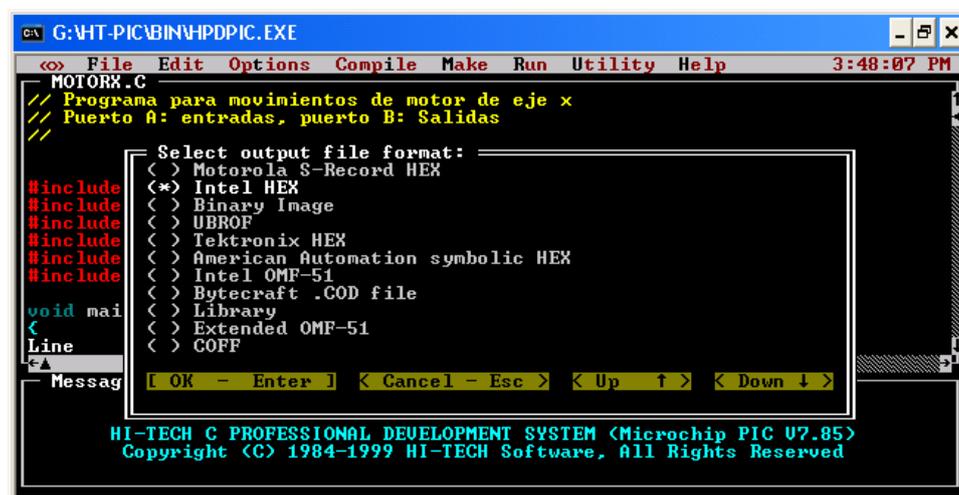


Anexo 3 – Figura. 7. Selección del tipo de datos

- Seleccionar algunas opciones para optimizar la compilación. (Anexo 3 – Figura 8 )
- También es importante escoger el formato de salida del archivo para la compatibilidad del microcontrolador. (Anexo 3 – Figura 9)
- Existen opciones de seteo en cuanto a símbolos y mapa del archivo. (Anexo 3 – Figura 10)



Anexo 3 – Figura. 8. Selección de Optimización para la compilación



Anexo 3 – Figura. 9. Selección del formato de salida del archivo



Anexo 3 – Figura. 10. Opciones de seteo del mapa y símbolos del archivo

- Ya seleccionados todos los parámetros para la compilación se procede a correr este proceso, en el menú **Run** (Ejecutar) se elige la opción Run y empezará la compilación. (Anexo 3 - Figura 11)
- Terminado éste proceso, si se ha encontrado ningún error la pantalla que saldrá es la figura 12 del Anexo 3 caso contrario saldrá indicado la línea del error.
- Si en caso, de un error, se debe corregir y realizar el mismo procedimiento anterior para la compilación.

```

G:\HT-PIC\BIN\HPDPIC.EXE
File Edit Options Compile Make Run Utility Help 3:49:08 PM
MOTORX.C
// Programa para movimientos de motor de eje x
// Puerto A: entradas, puerto B: Salidas
//
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main(void)
<
Line 1/155 Col 1 Insert C mode Search F2
Compiling "MOTORX.C"
66% completed
Compiling MOTORX.C ...
CGPIC.EXE -Zg1 -DMOTORX.SDB -CG:\HT-PIC\LIB\picinfo.ini -P16F877.2.8192.4.0.2,
0.112-127.1 C:\WINDOWS\TEMP\$$020466.001 C:\WINDOWS\TEMP\$$020466.000
ASPIC.EXE -16F877 -R2000 -X -CG:\HT-PIC\LIB\picinfo.ini -E2 -o -oMOTORX.OBJ C:
\WINDOWS\TEMP\$$020466.000
  
```

Anexo 3 – Figura.11. Compilación

```

G:\HT-PIC\BIN\HPDPIC.EXE
File Edit Options Compile Make Run Utility Help 3:48:39 PM
MOTORX.C
// Programa para movimientos de motor de eje x
// Puerto A: entradas, puerto B: Salidas
//
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main(void)
<
Line 1/155 Col 1 Insert C mode Search F2
Microchip PIC 07.85
Bank 0 RAM $0020 - $0031 $0012 (18) bytes
Bank 0 RAM $0070 - $007B $000C (12) bytes
$001E (30) bytes total Bank 0 RAM
  
```

Anexo 3 – Figura. 12. Compilación completa sin errores

- Con el proceso de compilación completa y sin errores, se genera el archivo **.hex**.

## **Programador IC-PROG:**

En unión del hardware adecuado, el software IC-PROG permite programar gran cantidad de dispositivos electrónicos.

### **Instalación del software IC-PROG**

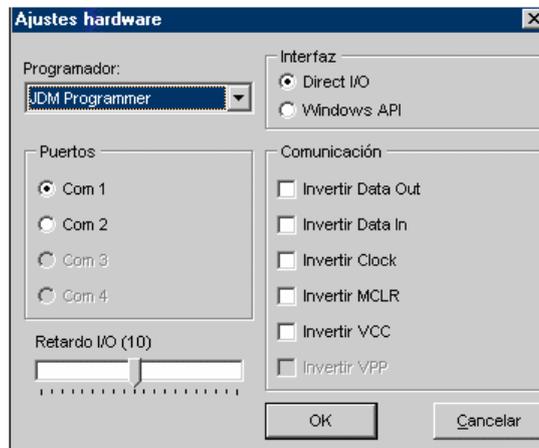
La instalación de este software es muy sencilla, y basta con descomprimir el fichero **ICProg.zip**, este archivo consta del fichero **lcprog.exe** que contiene todo el código necesario para su funcionamiento, con versiones de Windows 95, 98 ó Windows Milleniun.

En el caso de utilizar este software con Windows 2000, Windows NT ó Windows XP, también es necesario tener en el mismo directorio que el ejecutable, el fichero **icprog.sys** para Windows XP, sin el cual no se puede tener acceso a los puertos del ordenador.

### **Configurar el Hardware**

El paso siguiente es configurar el hardware necesario para programar los microcontroladores PIC. En la placa de programación/utilización que se ha realizado, la parte del circuito que se encarga de la programación, está basada en el programador tipo JDM. En pantalla principal de IC-PROG, se puede acceder a la configuración del hardware pulsando el icono  o bien mediante la pestaña **Ajustes** y después el comando **Tipo Hardware**, o bien

pulsando la tecla de función **F3**, de cualquiera de las maneras accederemos a la pantalla siguiente:



**Anexo 3 – Figura. 13. Compilación completa sin errores**

Esta es la configuración que se deberá elegir para el correcto funcionamiento del programador que forma parte de la placa de aplicación.

Una vez elegido el tipo de programador, como JDM y seleccionado el puerto serie adecuado, es importante no olvidarse de elegir el tipo de Interfase como Direct I/O, y en cuanto al tiempo de retardo, si hubiese problemas se pueden probar tiempos más largos.

## Menús

Los menús principales de IC-PROG tienen un icono gráfico en la pantalla principal, de tal forma que con éstos iconos prácticamente se pueden leer, borrar, programar y verificar cualquier dispositivo.

ICONO	FUNCION	ICONO	FUNCION
	Abrir un nuevo archivo		Guardar el buffer a un archivo
	Configurar el hardware		Opciones
	Leer un chip		Grabar un chip
	Borrar un chip		Verificar un chip
	Mostrar en ensamblador		Mostrar en hexadecimal

**Anexo 3 – Tabla. 1. Menús IC-PROG**

En la figura 14-a (Anexo 3) se muestra el tipo de dispositivo que se ha seleccionado. Pulsando en la flecha de la ventana se puede elegir cualquiera de los dispositivos soportados por el software de programación IC-PROG.

También se pueden ver las opciones de configuración del dispositivo seleccionado. Se muestra información del tipo de oscilador usado por el microcontrolador en cuestión (RC, XT, HS). También se observan los diferentes bits de configuración que tiene el microcontrolador. Estos bits serán diferentes para cada uno de los microcontroladores utilizados. Los de la imagen son los bits de configuración del PIC 16F84, y que muestran el estado del Wachtog (WDT), el Power on Reset (PWRT) y la protección de código (CP).

También se dispone de información del checksum (chequeo que se genera de todo el fichero de datos), y la posible identificación del microcontrolador, si este lo tuviera. (Anexo 3 - Figura 14-b)





Anexo 3 – Figura. 14. Configuración de parámetros

## Ajustes

Cuando se selecciona el idioma de empleo, se recorre a la pestaña **ajustes** y el comando **opciones**.



Anexo 3 – Figura. 15. Menú Ajustes

Los comandos posibles de esta pestaña son los siguientes:

Dispositivo Sirve para elegir el dispositivo con el que se va a trabajar (16F877).

Dispositivos recientes. Muestra el tipo de los últimos dispositivos utilizados.

Tipo Hardware F3. Aquí se elige el tipo de hardware que se va a usar para programar el dispositivo elegido anteriormente. (JDM Programmer).

Prueba hardware. Se puede realizar una prueba de que el hardware elegido está conectado al ordenador en el puerto elegido, y por tanto está listo para usarse.

Opciones. Desde este comando se pueden realizar las configuraciones que van a hacer que el uso de IC-PROG sea mucho más sencillo, práctico y además fiable.



**Anexo 3 – Figura. 16. Menú Opciones**

### Programación de un PIC 16F877

Los pasos que se exponen a continuación son prácticamente idénticos en cualquier tipo de dispositivo a programar, con las únicas variaciones de las características particulares de cada uno de los dispositivos.

El software IC-PROG debe estar configurado, como se explicó anteriormente, y el hardware de programación se encuentra conectado correctamente.

## 1. Abrir el archivo que contiene los datos a programar en el PIC 16F77.

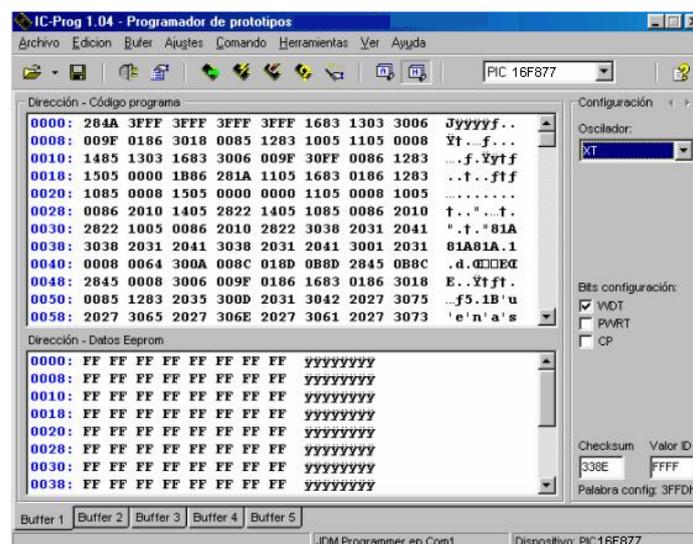
Mediante el icono  o bien con la pestaña **Archivo** y el comando **Abrir datos**, se accede al directorio dónde se guardan los archivos, como se observa en la figura 3.20. Los ficheros de datos que se usan para programar un PIC normalmente tendrán la extensión **.hex** (formato Intel hexadecimal-8 bits).



Anexo 3 – Figura. 17. Abrir Archivo

## 2. Ajustar la palabra de configuración y el tipo de oscilador.

Una vez cargados los datos del archivo correspondiente, la pantalla del IC-PROG presentará un aspecto como el mostrado en la figura 3.21. Se observa que los datos se han cargado en el área de código de programa



Anexo 3 – Figura. 18. Datos de Archivo

En el caso de que se haya programado la palabra de configuración, los bits correspondientes aparecerán marcados. Si se quiere cambiar alguno de ellos se puede hacer un clic directamente sobre él. También se puede cambiar el tipo de oscilador con que trabajará la aplicación una vez cargado en el PIC. Si en la aplicación no se programó ninguna palabra de configuración, se mostrarán los datos por defecto del programa (oscilador tipo RC y Watch Dog WDT activado).

Cualquiera de las áreas de datos (código de programa o datos EEPROM), se pueden modificar en este punto, ya que este software permite la edición de los datos que están presentes en el buffer del ordenador antes de ser enviados al PIC.

### 3. Programar el PIC 16F877.

Para proceder a la programación física del chip del PIC, basta con un clic sobre el icono , o bien activar la pestaña **Comando** y después **Programar todo**, o bien pulsar la tecla de función **F5**, y después de ser solicitada la confirmación de la orden dada, el chip comenzará a ser programado con los datos cargados en el buffer activo. El tiempo empleado en la grabación del PIC 16F877 dependerá de la rapidez del ordenador en que se esté trabajando.

Después de programar el código se programarán los datos de la memoria EEPROM y posteriormente la palabra de configuración del programa, todo ello sin requerir para nada la atención del usuario.

Una vez terminada la programación se procederá automáticamente a la verificación de los datos escritos en el chip. Del proceso de verificación también irá informando el PC.

En el caso de que la verificación haya sido correcta se informará de tal hecho y el proceso de grabación habrá finalizado.

Una buena práctica para evitar errores en la programación es la de, una vez terminado el proceso de grabación, ir a la sección de leer dispositivo y previo cambio del buffer activo leer de nuevo el PIC programado y proceder a la comparación entre los contenidos de los dos buffer, es decir comparar el archivo con el contenido del PIC.

Los programas utilizados en los microcontroladores son:

```
// MOTOR X y MOTOR VALVULA
```

```
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
```

```
int inter;
```

```
void motorx2()
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
```

```
DelayMs(10);
```

```
RB0=0;  
RB1=1;  
RB2=0;  
RB3=0;  
DelayMs(10);
```

```
RB0=1;  
RB1=0;  
RB2=0;  
RB3=0;  
DelayMs(10);  
}
```

```
void motorx1()
```

```
{  
RB0=1;  
RB1=0;  
RB2=0;  
RB3=0;  
DelayMs(10);  
RB0=0;  
RB1=1;  
RB2=0;  
RB3=0;  
DelayMs(10);
```

```
RB0=0;  
RB1=0;  
RB2=1;  
RB3=0;  
DelayMs(10);
```

```
RB0=0;  
RB1=0;  
RB2=0;  
RB3=1;  
DelayMs(10);  
}
```

```
void motorx11Y11()
```

```
{  
RB0=1;  
RB1=0;  
RB2=0;  
RB3=0;  
DelayMs(10);  
RB0=0;  
RB1=1;  
RB2=0;  
RB3=0;  
DelayMs(10);
```

```
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(10);
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
}
```

```
void motorx12Y12()
```

```
{
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(inter);
}
```

```
void motorx13Y23()
```

```
{
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(10);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(10);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(10);
RB0=0;
RB1=0;
```

```
RB2=0;
RB3=1;
DelayMs(10);
}
```

```
void motorx14Y24()
{
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(inter);
}
```

```
void motorx25Y25()
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(10);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(10);
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(10);
}
```

```
void motorx26Y26()
```

```
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(10);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(10);
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(10);
}
```

```
void motorx27Y17()
```

```
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(10);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(10);
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(10);
}
```

```
void motorx28Y18()
```

```
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(10);
}
```

```
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(inter);
RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(inter);
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(inter);
}
```

```
void valvula()
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(250);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(250);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(250);
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(250);
}
```

```
void valz()
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(250);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(250);
RB4=0;
RB5=0;
}
```

```
RB6=1;
RB7=0;
DelayMs(250);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(250);
}
```

```
void motorfin()
{
RB0=0;
RB1=0;
RB2=0;
RB3=0;
RB4=0;
RB5=0;
RB6=0;
RB7=0;
}
```

```
void main(void)
{
    ADCON1= 0X06;
    TRISA = 0b111111; // puerto A todas son entradas
    TRISB = 0b00000000; // puerto B todas son salidas
    TRISC = 0b111111; // puerto A todas son entradas
    PORTA=0;
    PORTB=0;
    PORTC=0;

    while(1)
    {

        while (PORTA==0)
        {
            inter=10;
            motorx1();
        }
        while (PORTA==1)
        {
            inter=10;
            motorx2();
        }

        while (PORTA==4)
        {
            inter=PORTC;
            motorx11Y11();
        }
        while (PORTA==5)
        {
            inter=PORTC;
        }
    }
}
```

```
motorx12Y12();
}

while (PORTA==6)
{
inter=PORTC;
motorx13Y23();
}
while (PORTA==7)
{
inter=PORTC;
motorx14Y24();
}
while (PORTA==8)
{
inter=PORTC;
motorx25Y25();
}
while (PORTA==9)
{
inter=PORTC;
motorx26Y26();
}
while (PORTA==10)
{
inter=PORTC;
motorx27Y17();
}
while (PORTA==11)
{
inter=PORTC;
motorx28Y18();
}
while (PORTA==12)
{
valvula();
}
while (PORTA==13)
{
valz();
}

while (PORTA==15)
{
motorfin();
}

}
```

---

```
// MOTOR Y y MOTOR Z
```

```
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
```

```
int inter;
```

```
void motorY2()
```

```
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
}
```

```
void motorY1()
```

```
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=1;
```

```
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
}
```

```
void motorx11Y11()
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(inter);
}
```

```
void motorx12Y12()
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
}
```

```
void motorx13Y23()
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
}
```

```
void motorx14Y24()
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
}
```

```
void motorx25Y25()
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
```

```
DelayMs(inter);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(inter);
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(inter);
}
```

```
void motorx26Y26()
{
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
}
```

```
void motorx27Y17()
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(inter);
}
```

```
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(inter);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(inter);
}
```

```
void motorx28Y18()
{
RB4=1;
RB5=0;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=1;
RB6=0;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=1;
RB7=0;
DelayMs(10);
RB4=0;
RB5=0;
RB6=0;
RB7=1;
DelayMs(10);
}
```

```
void motorzabajo()
{
RB0=0;
RB1=0;
RB2=0;
RB3=1;
DelayMs(25);
RB0=0;
RB1=0;
RB2=1;
RB3=0;
DelayMs(25);

RB0=0;
RB1=1;
RB2=0;
RB3=0;
DelayMs(25);
```

```
RB0=1;
RB1=0;
RB2=0;
RB3=0;
DelayMs(25);
}

void motorfin()
{
RB0=0;
RB1=0;
RB2=0;
RB3=0;
RB4=0;
RB5=0;
RB6=0;
RB7=0;

}

void main(void)
{
    ADCON1= 0X06;
    TRISA = 0b111111;    // puerto A todas son entradas
    TRISB = 0b00000000; // puerto B todas son salidas
    TRISC = 0b111111;    // puerto A todas son entradas
    PORTA=0;
    PORTB=0;
    PORTC=0;

    while(1)
    {

        while (PORTA==2)
        {
            inter=10;
            motorY1();
        }
        while (PORTA==3)
        {
            inter=10;
            motorY2();
        }
        while (PORTA==4)
        {
            inter=PORTC;
            motorx11Y11();
        }
        while (PORTA==5)
        {
            inter=PORTC;
            motorx12Y12();
        }
    }
}
```

```
while (PORTA==6)
{
inter=PORTC;
motorx13Y23();
}
while (PORTA==7)
{
inter=PORTC;
motorx14Y24();
}
while (PORTA==8)
{
inter=PORTC;
motorx25Y25();
}
while (PORTA==9)
{
inter=PORTC;
motorx26Y26();
}
while (PORTA==10)
{
inter=PORTC;
motorx27Y17();
}
while (PORTA==11)
{
inter=PORTC;
motorx28Y18();
}
while (PORTA==14)
{
inter=25;
motorzabajo();
}
while (PORTA==15)
{
motorfin();
}
}
}
```

## // CONTROL DE TEMPERATURA

```
#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main(void)
{
    ADCON0= 0X89;
    ADCON1= 0X80;           //entradas analogas
    TRISA = 0b1111111;     // puerto A todas son entradas
    TRISB = 0b00000000;   // puerto B todas son salidas
    TRISC = 0b00000000;   // puerto A todas son entradas
//    PORTA=0;
//    PORTB=0x80;
    PORTC=0;

    while(1)
    {
        unsigned int value,puerto; //creacion de variables enteras
        ADCON0=0X8D;
        DelayMs(250); //retardo necesario para la conversion

        while(ADCON0==5)
        {
        }
        value=(ADRESH<<8);
        value=value+ADRESL; //lee toma una muestra de voltaje en RA0
//    PORTB=value;
//    PORTC=ADRESL;
        if(value>=610)
        {
            puerto=0b00000000;
            PORTB=puerto;
        }
        if(value<=605)
        {
            puerto=0b10000000;
            PORTB=puerto;
        }
        if(value>605 && value<610)
        {
            PORTB=puerto;
        }
        DelayMs(250);
        DelayMs(250);
        DelayMs(250);
        DelayMs(250);
    }
}
```

## **ANEXO 4**

**Software Convertidor de Coordenadas a Código G “ACE”**

## Inicio del convertidor ACE

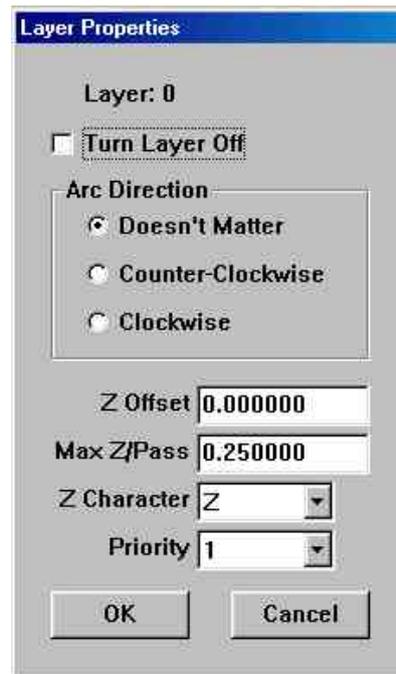
Cuando inicia el convertidor ACE presenta la siguiente pantalla:



**Anexo 4 – Figura. 1. Pantalla Principal del Convertidor a Código G**

El convertidor ACE cortará una capa a la vez según la prioridad. La prioridad 1 es primera. Una lista de las capas en el archivo actual del dxf y la prioridad para cada una se enumeran en la izquierda.

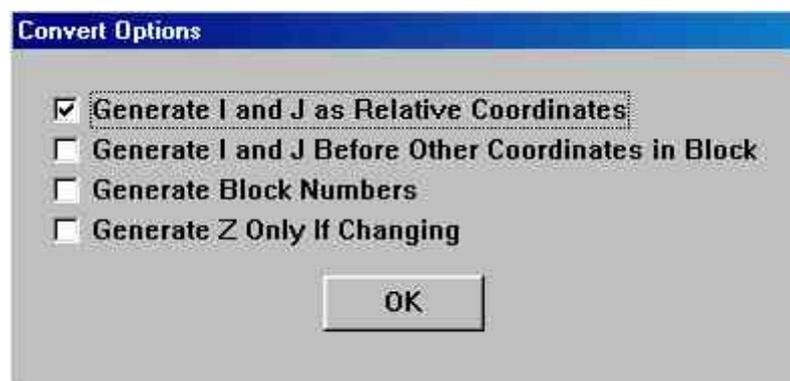
Se puede configurar las capas que se quieran convertir en la siguiente pantalla:



Anexo 4 – Figura. 2. Pantalla de Capas del Convertidor a Código G

El convertidor ACE cortará cualquier arco que esté presente en una dirección que usted elija con los botones de la dirección del arco. La compensación de Z fija la profundidad de cada línea para las máquinas.

Cuando se establecen la prioridades de las capas el Convertidor ACE genera un cuadro de dialogo con opciones para la conversión final del archivo dxf a código G.



Anexo 4 – Figura. 3. Pantalla Final del Convertidor a Código G

La opción de Generar Relativos Coordinates se utiliza en la mayoría de maquinas CNC y Robots Cartesianos. La opción de Coordinates in Block es parecida a la anterior, pero genera una línea al comienzo del código que sigue un orden lineal.

La última opción de conversión permite que usted genere coordenadas Z solamente cuando el eje Z requiera moverse. Esto conserva memoria y permite la mano fácil que corrige para las profundidades.

#### Localización de averías

- El Convertidor ACE no reconoce archivos que tengan solo arcos.
- El ACE no reconoce poli líneas.
- El as no procesa correctamente archivos procedentes de Visio

## **ANEXO 5**

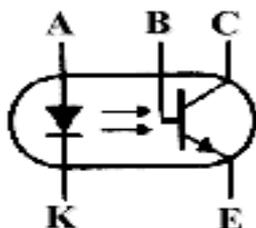
### **Opto-acopladores: Características y funcionamiento**

El optoacoplador es un dispositivo que se compone simplemente de un paquete que contiene un LED infrarrojo y un fotodetector, tal como un diodo de silicio, un par de transistores Darlington, o un SCR. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

El funcionamiento de los optoacopladores es muy simple, se basa en que al led infrarrojo emisor, le llega la señal a transmitir, la luz emitida por el led, varía en intensidad de acuerdo a la magnitud de la señal de entrada, esta luz puede activar ya sea un fotodiodo, un fototriac, un transistor.

En el caso del fotodiodo, o del fototriac, estos se activarán cuando la intensidad de la luz del led infrarrojo llegue a cierto nivel. Pero en el caso del transistor, la señal se puede transmitir tal como es a la entrada en el diodo, debido a que la luz del diodo actúa sobre la base de fototransistor, esto hace que la corriente que deja pasar el fototransistor, varíe de acuerdo a la cantidad de luz que recibe por parte del led.

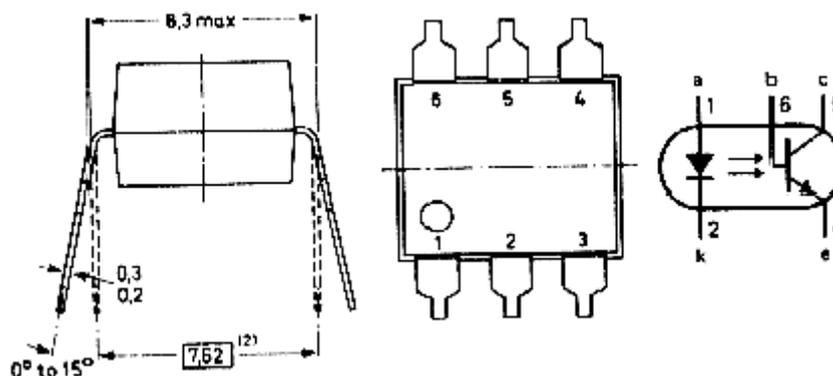
La respuesta de longitud de onda de cada dispositivo se ajusta para que sea lo más idéntica posible. Se diseñan con tiempos de respuesta tan pequeños que pueden emplearse para transmitir datos en el rango de MHz.



Anexo 5 – Figura. 1. Esquema interno de un optoacoplador

Para diseño el optoacoplador se lo representa como se muestra en la figura 1 del Anexo 5. El tiempo de conmutación de un optoacoplador disminuye con el aumento de corriente, en tanto que para muchos dispositivos ocurre exactamente lo contrario.

Una de las mayores aplicaciones de los optoacopladores, es como interfaz de potencia de un sistema digital a un motor, un switch, una máquina, cualquier sistema que no trabaje al nivel de voltaje o corriente de los circuitos digitales (por lo general de 3 a 10 V), logrando manejar niveles altos de voltaje (120VDC, o cualquier otro voltaje de alterna o directa, tan solo de debe de cuidar que el optoacoplador sea el adecuado). Físicamente el optoacoplador tiene la siguiente forma y las siguientes medidas como se muestra en la figura 2 del Anexo 5.



Anexo 5 – Figura. 2. Dimensiones de un optoacoplador

## **ANEXO 6**

### **Planos de Implementación del Prototipo de Impresora para Modelos 3D en Cera**

## **MANUAL DE USUARIO**

**SDJC**

**IMPRESORA PARA MODELOS 3D EN CERA**



**IMPORTANTE**

Para evitar cualquier malfuncionamiento y evitar daños, por favor lea detenidamente este manual de instrucciones, antes de conectar y operar este equipo.

**Manual de Usuario**

*Instrucciones para instalación del software*

*Impresión*

*Resolución de problemas*

## **Aviso de derechos reservados**

Quedan reservados todos los derechos. Ninguna parte de esta publicación podrá ser reproducida, almacenada en un sistema de recuperación ni transmitida en forma alguna por ningún medio electrónico, mecánico, de fotocopiado, de grabación o cualquier otro, sin previo consentimiento por escrito de: Santiago Dávalos y Juan Carlos Bonilla. Este manual contiene información específica para este equipo SDJC – 001. SDJC no se hace responsable si esta información es utilizada en otros productos.

SDJC no asumirá responsabilidad ante el comprador de este producto o ante terceros por daños, pérdidas, costos o gastos en que incurrieren éstos como consecuencia de: accidente, uso inadecuado o abuso de este producto o modificaciones, alteraciones no autorizadas al mismo, o no seguir rigurosamente las instrucciones de operación y mantenimiento de este manual.

## CONTENIDO

<b>CONTENIDO</b> .....	197
<b>Introducción</b> .....	198
<b>Instrucciones Importantes de Seguridad</b> .....	199
<b>Instalación de Software</b> .....	200
<b>Identificación de piezas</b> .....	201
<b>Preparativos</b> .....	203
<b>Funcionamiento</b> .....	205
Encendido y Apagado de la Impresora.....	205
Archivo dxf.....	205
Importación del archivo dxf.....	206
Conversión del archivo *.dxf a Código G.....	207
Impresión del Modelo.....	209
<b>Solución de Problemas</b> .....	211
<b>Información Legal</b> .....	212
Requerimientos del Sistema para Windows.....	212
Especificaciones Técnicas.....	213

## Introducción

La impresora 3D permite la visualización de la información, este dispositivo mecatrónico a más de su seguridad, eficiencia y confiabilidad es un equipo de fácil manejo, el cual simplifica el diseño de nuevos modelos en cualquier tipo de empresa.

La Impresora 3D permite la obtención de un modelo a escala real de cualquier dispositivo o herramienta que se requiera.

**Nombre del producto:** Impresora para Modelos 3D en Cera

**Modelo:** SDJC-001

## Instrucciones Importantes de Seguridad

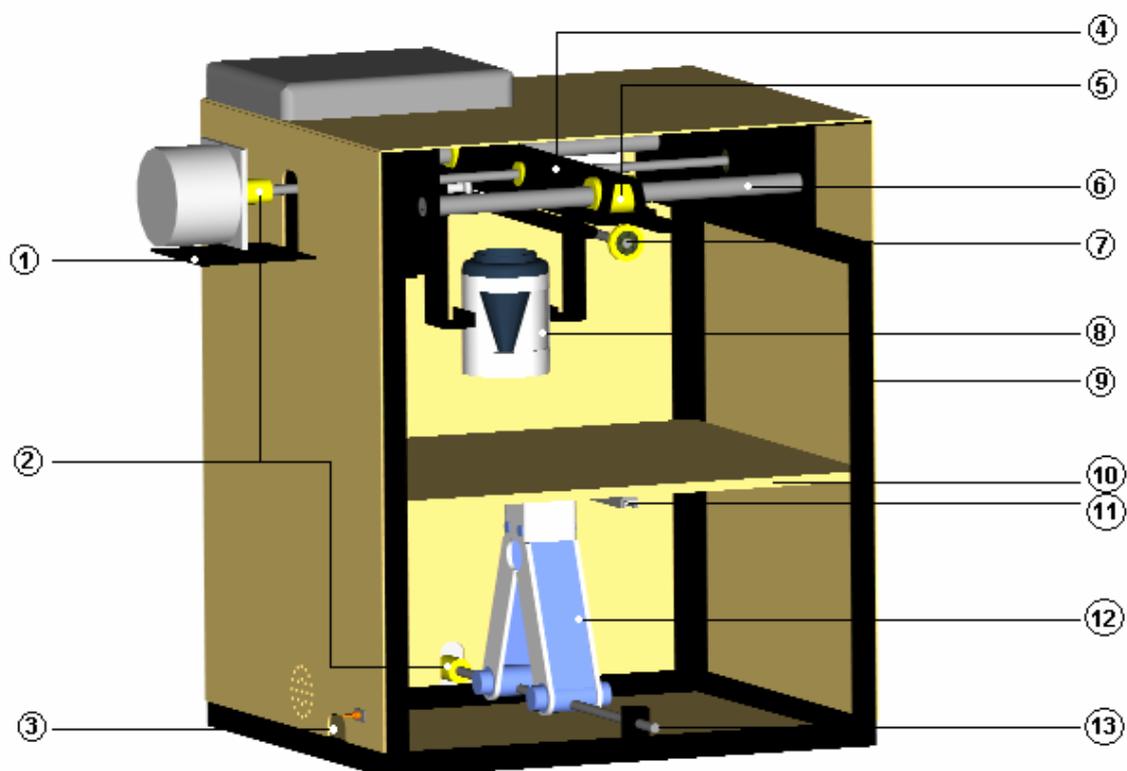
1. Para evitar el Riesgo de incendios y electrocución, no exponga la unidad a la lluvia ni a humedad.
2. Para evitar quemaduras o descargas eléctricas, no introduzca ningún elemento metálico o partes de su cuerpo en la unidad mientras se desarrolla el proceso de impresión. Solicite asistencia técnica solo a personal especializado.
3. Este producto ha sido probado y ha demostrado cumplir con las especificaciones establecidas.
4. No instale el equipo cerca de fuentes de calor como radiadores, registros, estufas u otros aparatos que produzcan calor.
5. Proteja el cable de corriente para que no caminen sobre él o quede atorado, particularmente en contactos, en receptáculos y en el punto por donde sale del aparato.
6. Desenchufe el equipo durante tormentas de rayos o cuando no vaya utilizarlo por periodos prolongados.

## Instalación de Software

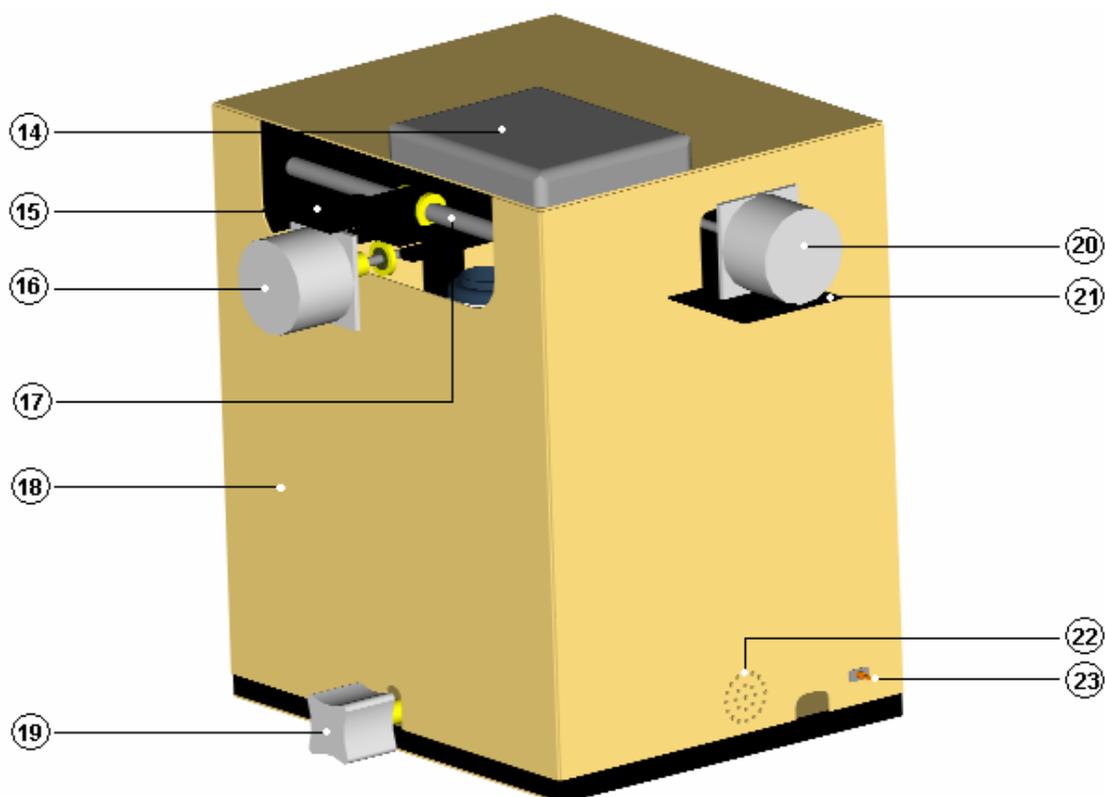
Para la instalación debe seguir los siguientes pasos:

1. Ejecutar el archivo setup.exe.
2. Seguir los pasos que se indica en el proceso de instalación.
3. Una vez instalado correctamente el Software se debe comprobar que el cable DB25 este conectado al computador
4. Verificar que la fuente de alimentación esté energizada.

## Identificación de piezas



1. Soporte Motor Eje X
2. Bocines para acople entre tornillo y eje motor
3. Conexión fuente de poder - energización
4. Eje X y soporte para Eje Y
5. Rodamientos lineales – movimiento Eje X
6. Eje de transmisión – soporte para movimiento de Eje X
7. Tornillo Eje Y
8. Dispensador de Cera
9. Estructura metálica
10. Plataforma para la fabricación del modelo
11. Riel para movimiento del Eje Z
12. Brazos mecánicos de Gata que produce el movimiento en el Eje Z
13. Tornillo y soporte para el Eje Z



- 14. Caja de control – circuitos de control y potencia – conexión con PC
- 15. Soporte para el motor del Eje Y
- 16. Motor para el movimiento del Eje Y
- 17. Eje de transmisión – soporte para movimiento de Eje X
- 18. Armazón de protección
- 19. Motor para el movimiento del Eje Z
- 20. Motor para el movimiento del Eje X
- 21. Soporte para el motor del Eje X
- 22. Ventilación para la fuente de poder
- 23. Interruptor ON/OFF – encendido y apagado de la impresora

## Preparativos

### Notas sobre la compatibilidad de datos de dibujo

- Esta impresora cumple las normas de diseño del estándar universal de sistemas de archivos de dibujos que son archivos de tipo dxf.

### Conexión de los motores a la tarjeta

- En la tarjeta principal se especifica el orden de los motores y su posición, se debe colocar con cuidado y asegurarse que estén bien conectados. Cada motor tiene un conector de 6 pines.

### Fuente de energía

- Utilizar la fuente recomendada:  
Switching Power Supply  
Input: 100 – 130 VAC, 50/60 Hz  
Output: 5V – 30 A
- Conecte la Fuente de Energía a una toma mural con el cable de alimentación

## Conexión PC - Impresora

- Esta Impresora se conecta al equipo a través de un cable DB25.



### Elimine la suciedad de la superficie del dispensador de cera

- Si la suciedad cambia el color de la cera o se pega a la superficie del dispensador debido al calor de la resistencia, puede que la impresión tenga problemas.

### Evite que la impresora se moje

- Si transporta o utiliza la unidad en lugares al aire libre, procure que la impresora no se moje. Ya que se puede producir condensación de humedad en los circuitos de la impresora y producir fallos en la misma.

### No exponga la impresora a sustancias como arena o polvo.

- El uso de la impresora en lugares con arena o polvo puede producir un funcionamiento incorrecto.

## Funcionamiento

### Encendido y Apagado de la Impresora

- Pulse el Interruptor ON/OFF, al estado ON para encender la impresora, se enciende en verde y la alimentación esta conectada en ese momento esta listo para utilizar la impresora.
- Pulse el Interruptor ON/OFF, al estado OFF y se apagara la luz verde y la impresora se desconectara.
- Para una correcta impresión es necesario esperar un tiempo de 15 a 20 minutos aproximadamente para el calentamiento previo de la parafina en el dispensador de cera.

**Nota:** NO RETIRE LA ALIMENTACION CUANDO LA UNIDAD ESTE EN FUNCIONAMIENTO, PODRIA PROVOCAR ANOMALIAS TANTO EN EL EQUIPO COMO EN EL MODELO QUE SE ESTA FABRICANDO.

### Archivo dxf

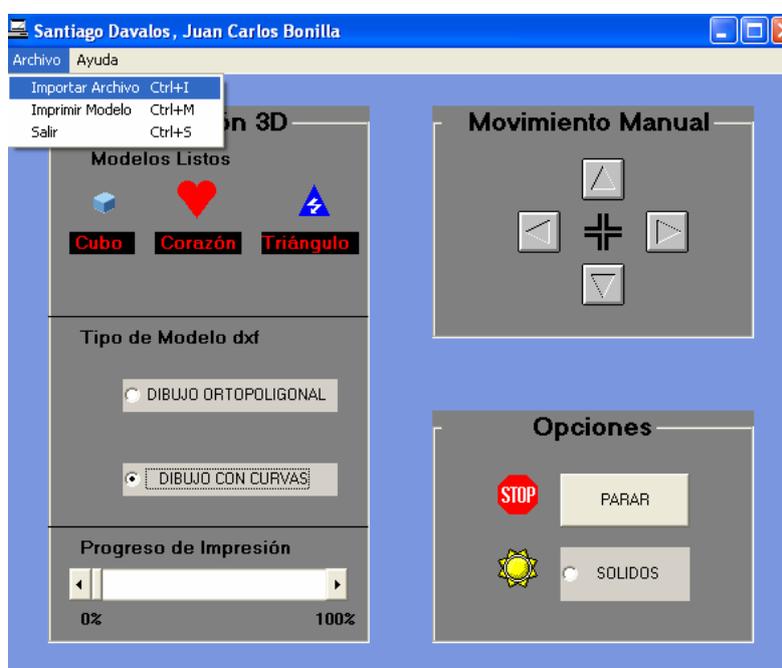
Para imprimir un modelo dibujado previamente en cualquier programa de diseño grafico, lo que se debe hacer es:

1. Guardar el diseño como un archivo de extensión dxf, la cual es reconocida por el Software de impresión.
2. Verificar que el punto de inicio del dispensador de cera esté en el origen de los ejes, el cual se detalla en la tapa de la impresora.

Una vez que se ha verificado todo lo concerniente al encendido del equipo y el formato del archivo que contiene el modelo a imprimirse, ejecute el software de la Impresora 3D y continúe con los pasos siguientes:

### Importación del archivo dxf

Ya en el Software de Impresión, para obtener un diseño hecho en cualquier programa de dibujo se debe ir a la opción importar Archivo del menú Archivo en la barra de herramientas.



Una vez que se escoge la opción, el software ejecuta directamente el programa convertidor ACE.

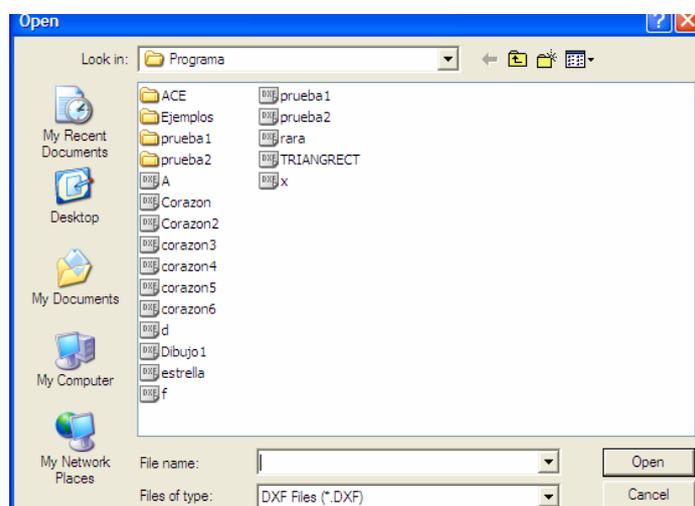
## Conversión archivo \*.dxf a Código G

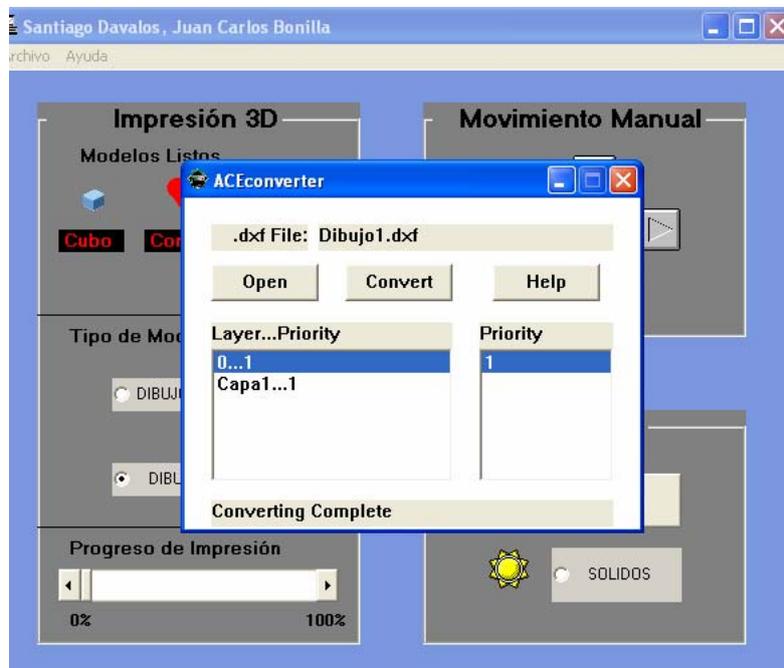
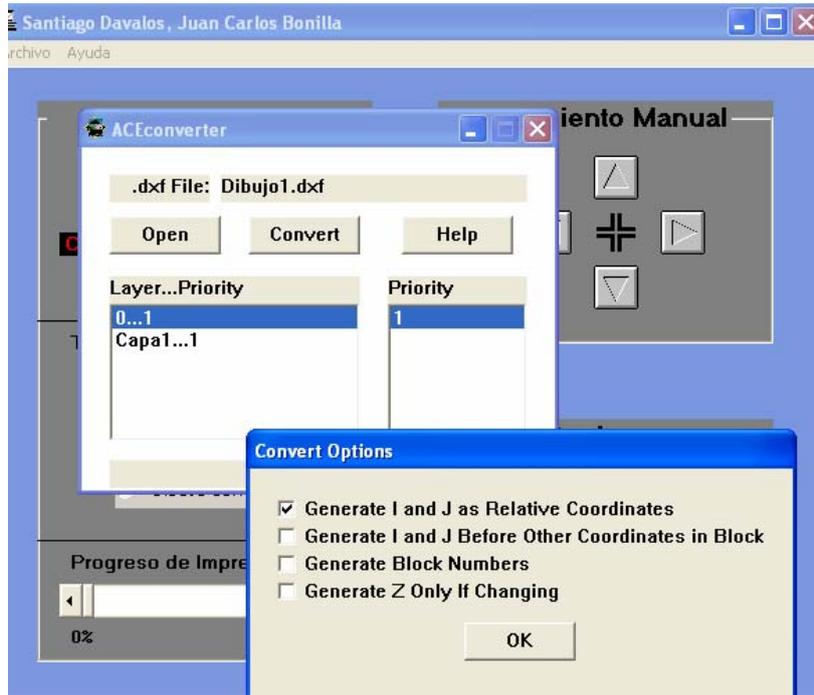
Cuando se esta ejecutando el Convertidor ACE se debe elegir la opción OPEN la cual abre un cuadro de diálogo donde seleccionamos el archivo dxf a convertir.



Luego, se selecciona el archivo \*.dxf del modelo a imprimirse.

Para concluir con la conversión del archivo el programa pide escoger algunas opciones de conversión, las cuales se deben mantener las previamente seteadas.

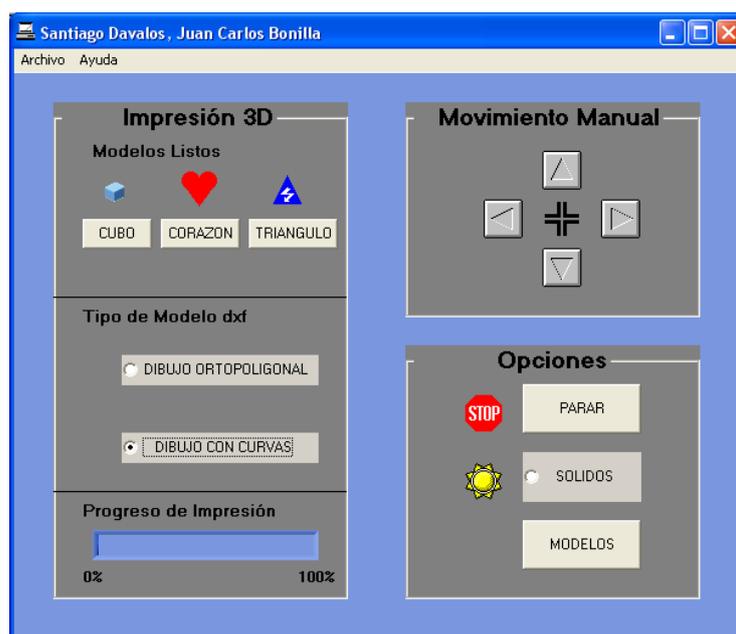




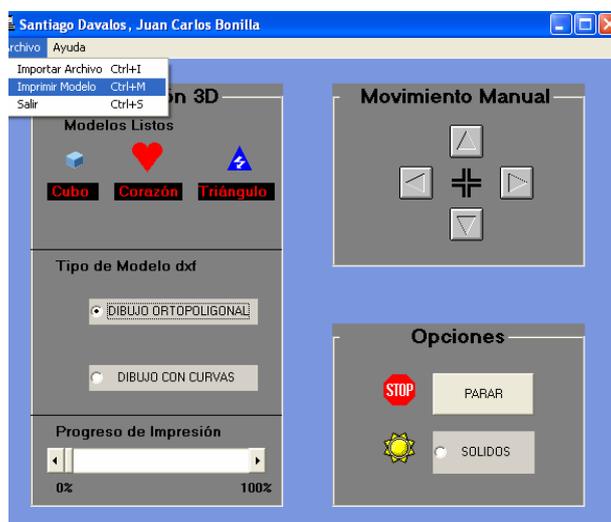
## Impresión del Modelo

Completados todos los pasos anteriores el software esta listo para comenzar con la impresión:

- Primero se selecciona la opción: si el modelo es ortopoligonal (solo rectas) o si es con curvas.

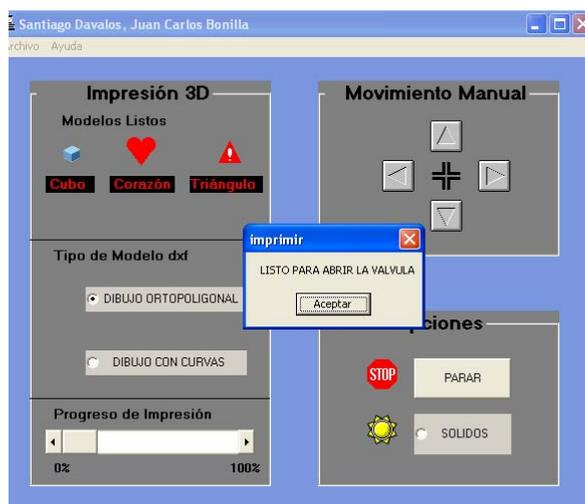


- Luego se escoge la opción de la barra de herramientas, Imprimir Modelo.



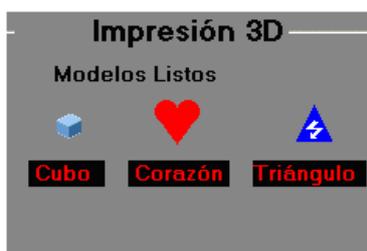
- Se puede o no escoger la opción de SÓLIDOS, ésta permite rellenar el modelo, es decir, que si se selecciona esta opción el modelo que se imprimirá será sólido.

Una vez que se realiza este procedimiento la Impresora esta lista para comenzar con el procedimiento.



**Nota:** Para mayor rapidez en la impresión de modelos que no tengan curvas se debe escoger la opción Dibujo Ortopoligonal.

El programa consta con una opción extra que es la impresión de modelos prediseñados que no necesita de los pasos previamente explicados sino basta con hacer clic en la pestaña con el dibujo a imprimir y comenzara con el procedimiento.



## **Solución de problemas**

### **¿Mi Impresora no se prende?**

- Mire si la fuente de alimentación está conectada correctamente, o si el cable de poder de la fuente esta dentro de la misma.

### **¿El Software funciona correctamente pero no imprime?**

Este problema se puede presentar por dos razones diferentes:

- Mire si el cable DB25 entre la computadora y la impresora esta conectada correctamente.
- Mire si los motores de la Impresora están en el orden y posición correcta (anteriormente descrito).

### **¿Se Produce un error al momento de la Conversión de mi archivo?**

- Mire si el archivo que quiere convertir es un archivo con extensión Dxf, o si no cambió las opciones preestablecidas en el convertidor.

### **¿Se produce un error al momento final de la impresión?**

- Esto se debe a que su dibujo no tenia curvas y usted eligió la opción Dibujo /Ortopoligonal.

## Información Legal

### Requerimientos del Sistema para Windows

Para el funcionamiento correcto de la Impresora SDJC de Modelos 3D y su software, el sistema debe disponer de:

- PC compatible con Microsoft Windows, con procesador equivalente a Pentium (se recomienda que sea mínimo Pentium III o superior).
- Microsoft Windows 98, Windows 2000, Windows ME o Windows XP.
- Memoria RAM de 128 MB como mínimo.
- 50 MB de espacio libre en el disco duro.
- Disponer de puerto paralelo (indispensable).

## Especificaciones Técnicas

<b>Impresión</b>	
Método de impresión	Inyección de parafina
Dirección de impresión	Tridireccional movimientos en ejes X , Y y Z
Resolución de impresión	± 0.5 cm.
Velocidad de impresión	1 cm. por cada 14 segundos
Interfaz	Compatible con bus paralelo Universal DB25 para transmisión de datos.
<b>Cartuchos de Parafina</b>	
Color	Cualquier color disponible de parafina
Duración del cartucho	6 meses a 25°C después de abrir el envoltorio
Temperatura	Almacenamiento: -10 a 25°C
<b>Mecánicas</b>	
Recorrido plataforma	Alimentador de bandeja, entrada frontal
Capacidad de alimentador	1 bandeja por modelo
Dimensiones	350mm (Ancho) x 470mm (Profundidad) x 520mm (Alto)
Peso	20 Kg. (44 lbs)
<b>Eléctricas</b>	
Voltaje de Entrada	110 -127 V a.C.
Frecuencia nominal	60 Hz.
<b>Ambientales</b>	
Temperatura:	Funcionamiento: 10 a 35 °C Almacenamiento: -40 a 60 °C
Humedad	15 al 80 por ciento de HR sin condensar

## ÍNDICE DE TABLAS

Tabla. 1.1.	Tipos de robots por su geometría	26
Tabla. 1.2.	Software para CAD	30
Tabla. 2.1.	Materiales - Fase 1	41
Tabla. 2.2.	Materiales - Fase 2	44
Tabla. 2.3.	Materiales Fase 4	48
Tabla. 2.4.	Características de Motores a paso (Diseño)	56
Tabla. 2.5.	Embudo para dispensador	63
Tabla. 2.6.	Recubrimiento del dispensador	64
Tabla. 2.7.	Resistencia para calentamiento de Parafina	65
Tabla. 3.1.	Características Microcontrolador 16F877	78
Tabla. 3.2.	Elementos del Circuito de Control de Temperatura	82
Tabla. 3.3.	Estator de 4 bobinas de un motor a pasos	90
Tabla. 3.4.	Lenguajes, compiladores y programador disponibles	93
Tabla. 3.5.	Combinaciones y funciones	95
Tabla. 5.1.	Elementos del circuito de Control	121
Tabla. 5.2.	Elementos del circuito de Potencia	122
Tabla. 5.3.	Características de Motores a paso	122

**Sangolquí, 17 de Octubre del 2006**

**ELABORADO POR:**

-----  
Sr. Juan Carlos Bonilla Arroyo

-----  
Sr. Santiago Javier Dávalos Ávila

**AUTORIDADES:**

-----  
Sr. Ing. Xavier F. Martínez  
TCRN. DE E.M.  
Decano de la Facultad de Ingeniería Electrónica

-----  
Sr. Dr. Jorge Carvajal  
Secretario Académico de la Facultad de Ingeniería Electrónica