

TESIS DE GRADO EN INGENIERÍA ELECTROMECAÁNICA

JULIO, 2015

“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

Autores:	CORONEL, Martín	Matrícula:	12488
	IRIANI, Matías		12519
Carrera:	Ingeniería Electromecánica		
Tutor:	Ing. UICICH Gustavo		
Evaluadores:	Ing. SUAREZ, Juan Antonio		
	Dr. Ing. MACRI, Mario		
	Ing. MURCIA, Guillermo		
	Ing. GARIN, Eduardo		



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

TESIS DE GRADO EN INGENIERÍA ELECTROMECAÁNICA

JULIO, 2015

“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

Autores:	CORONEL, Martín	Matrícula:	12488
	IRIANI, Matías		12519
Carrera:	Ingeniería Electromecánica		
Tutor:	Ing. UICICH Gustavo		
Evaluadores:	Ing. SUAREZ, Juan Antonio		
	Dr. Ing. MACRI, Mario		
	Ing. MURCIA, Guillermo		
	Ing. GARIN, Eduardo		

Agradecemos a nuestros familiares por el apoyo incondicional brindado en esta etapa que termina y en la vida, a nuestros amigos y compañeros que supieron acompañarnos en el estudio y el ocio en nuestra estadía lejos de la familia, a los docentes de la facultad de los cuales hemos aprendido importantes valores en lo profesional y en lo personal.

También agradecemos en particular a las personas que nos brindaron su ayuda para poder realizar este trabajo final, como son el Ingeniero Gustavo Uicich, quien depositó su confianza en nosotros para poder formar parte de su proyecto, al Doctor Ingeniero Amadeo Daniel Sosa, profesor del área de Tecnología Mecánica quien nos brindó su apoyo incondicional en el diseño y construcción de la máquina, además de facilitarnos la utilización de las instalaciones del taller del área, en donde se construyó el prototipo, al Ingeniero Matías Valdés, quien desde la división de corrosión del INTEMA financió el desarrollo del equipo y al Sr. Juan Blanco, quien es pañolero en el taller de Tecnología Mecánica y supo ayudarnos con la construcción de la máquina.

ÍNDICE

CAPITULO I INTRODUCCIÓN Y OBJETIVOS	6
1.1 – OBJETIVO GENERAL.....	6
1.1.1 – Deposición por spray.....	6
1.1.2 – Recubrimiento por inmersión (dipping)	9
1.2 – OBJETIVOS ESPECÍFICOS.....	10
CAPITULO II DISEÑO	12
2.1 – INTRODUCCIÓN.....	12
2.1.1 – Mecanismos de transmisión del movimiento	16
2.1.2 – Estructuras y mecanismos básicos	20
2.2 – DISEÑO DE DEPOSITADORA POR SPRAY	22
2.2.1 – Diseño de estructura principal	22
2.2.2 – Diseño del puente móvil	27
2.2.3 – Diseño de la torre.....	32
CAPITULO III PROCESO DE FABRICACIÓN Y ENSAMBLAJE.....	38
3.1 – INTRODUCCIÓN.....	38
3.2 – ESTRUCTURA PRINCIPAL	39
3.3 – PUENTE MÓVIL.....	45
3.4 – TORRE VERTICAL.....	51
3.5 – CABLEADO	57
CAPITULO IV HARDWARE PARA MONITOREO Y CONTROL	60
4.1 – INTRODUCCIÓN AL SISTEMA SCADA.....	60
4.2 – SENSORES Y ACTUADORES.....	62
4.2.1 – Sensores.....	62
4.2.1.1 – Sensor diferencial de presión.....	63
4.2.1.2 – Finales de carrera.....	64

4.2.2 – Actuadores	65
4.2.2.1 – Motores Paso a Paso (PAP)	66
4.2.2.2 – Electroválvula	70
4.2.2.3 – Placa calefactora	73
4.3 – UNIDAD TERMINAL REMOTA (RTU)	74
4.3.1 – Controlador: Arduino	75
4.3.1.1 – Selección de la placa	76
4.3.1.2 – Limitaciones.....	77
4.3.2 – Hardware de soporte.....	78
4.3.2.1 – Shield CNC.....	78
4.3.2.2 – Driver Polulu a4988 para motores PAP	83
4.3.2.3 – Placa de distribución de conexiones y aislación de controlador	87
4.3.2.4 – Placa para controlar temperatura.....	92
3.3.2.5 – Conexionado de todas las partes.....	99
CAPITULO V SOFTWARE PARA MONITOREO Y CONTROL	103
5.1 – COMUNICACIÓN	103
5.2 – ÓRDENES DE INTERCAMBIO ENTRE EL RTU Y MTU:.....	104
5.2.1 – Control de depresión en la campana de seguridad	104
5.2.2 – Control de placa de temperatura	105
5.2.3 – Configuración de velocidad y aceleración	105
5.2.4 – Orden de movimiento	106
5.2.5 – Orden de ir a posición de reposo.....	107
5.3 – SOFTWARE DEL CONTROLADOR (RTU)	108
5.3.1 – Declaración de variables	109
5.3.2 – Diagrama de flujo	110
5.4 – UNIDAD MAESTRA E INTERFAZ HOMBRE-MÁQUINA (MTU/HMI)	113
5.4.1 – Software de MTU como máquina de estados.....	115

5.4.2 – Software para depositadora por Spray	116
4.4.3 – Software para proceso de Dipping	124
CAPITULO VI EVALUACIÓN DE IMPACTO AMBIENTAL	131
CAPITULO VII ANÁLISIS ECONÓMICO.....	133
7.1 – ANÁLISIS DE COSTOS.....	133
7.2 – COMPARACIÓN CON PRODUCTO SIMILAR:	136
CAPITULO VIII CONCLUSIÓN	137
BIBLIOGRAFÍA	140
Anexo A - ARDUINO.....	141
Anexo B - LABVIEW	148
Anexo C - ELECTRÓNICA.....	181
Anexo D - ELECTROVÁLVULA	186
Anexo E - SENSOR DIFERENCIAL DE PRESIÓN	190
Anexo F - MOTORES PASO A PASO	191
Anexo G - PLACA CALEFACTORA	199

CAPITULO I INTRODUCCIÓN Y OBJETIVOS

1.1 – OBJETIVO GENERAL

El presente proyecto documenta el diseño, construcción y prueba del prototipo de un sistema electromecánico con control automático para la aplicación de películas delgadas. Comprende un posicionador de tres ejes con los mecanismos, accionamientos, subsistema de control, electrónica y software asociados, orientados a la producción de recubrimientos por inmersión (dipping) o por rociado (spray). Estos recubrimientos se realizan con el fin de modificar las propiedades de superficie del sustrato sobre el cual se aplican, tales como aspecto, adhesión, tensión superficial, resistencia a la corrosión o al desgaste, etc. La calidad del mismo en términos de homogeneidad y espesor dependerán intrínsecamente de la precisión en el control de las variables intervinientes.

1.1.1 – Deposición por spray

La deposición por spray produce láminas o films delgados de material depositado sobre el sustrato. La técnica conocida como SPT (Spray Pyrolysis Technique) es una variante que consolida la deposición a elevadas temperaturas. Entre los parámetros que definen el resultado del proceso se pueden mencionar temperatura del sustrato, concentración de la solución aplicada, velocidad de deposición, cantidad de aplicaciones, etc.

En el caso del presente trabajo el mecanismo consiste en el rociado de la solución en estudio mediante un atomizador que la eyecta debido a la diferencia de presión que se genera por el flujo de un gas que puede ser aire o nitrógeno a alta velocidad tal como se muestra en la figura 1.1.

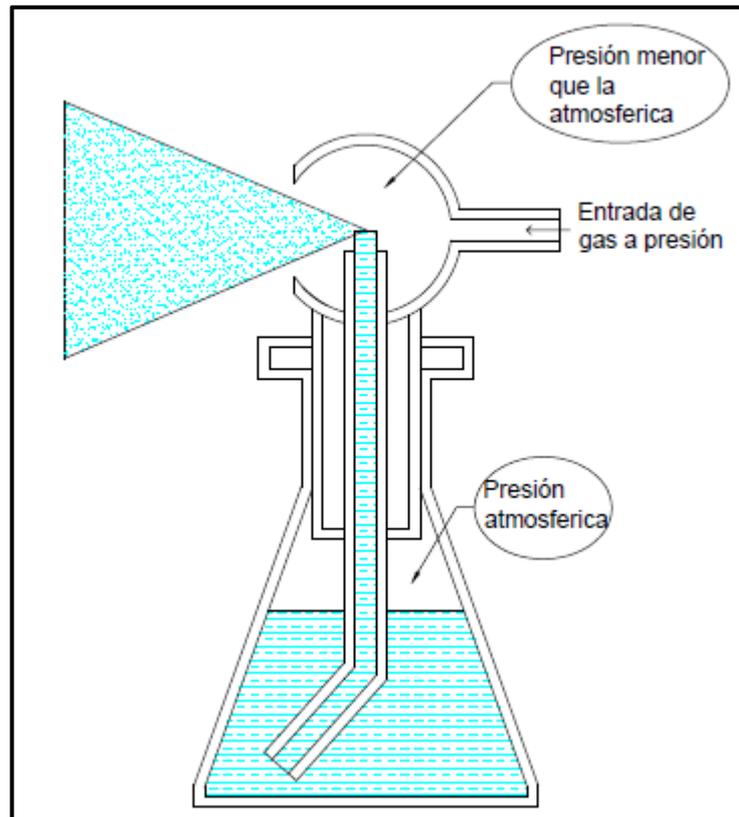


Figura 1.1 – Esquema del atomizador utilizado en depósito por spray.

Este rociado se genera sobre sustratos de vidrio que se encuentran sobre una placa calefactora que varía su temperatura entre 450 y 550 grados centígrados. Este proceso se realiza dentro de una campana, en donde un extractor de gases asegura que los vapores de la solución sean expulsados hacia la atmosfera y así evitar posibles intoxicaciones del usuario.

En este trabajo, se busca automatizar el proceso de deposición de la solución (aquí se incluye el posicionador electromecánico y el accionamiento de la correspondiente válvula que habilita la inyección de gas) como también el control de la temperatura de la placa calefactora y el monitoreo de la depresión en el interior de la campana con respecto a la presión atmosférica.

Básicamente el conjunto es como el que se muestra en la figura 1.2.

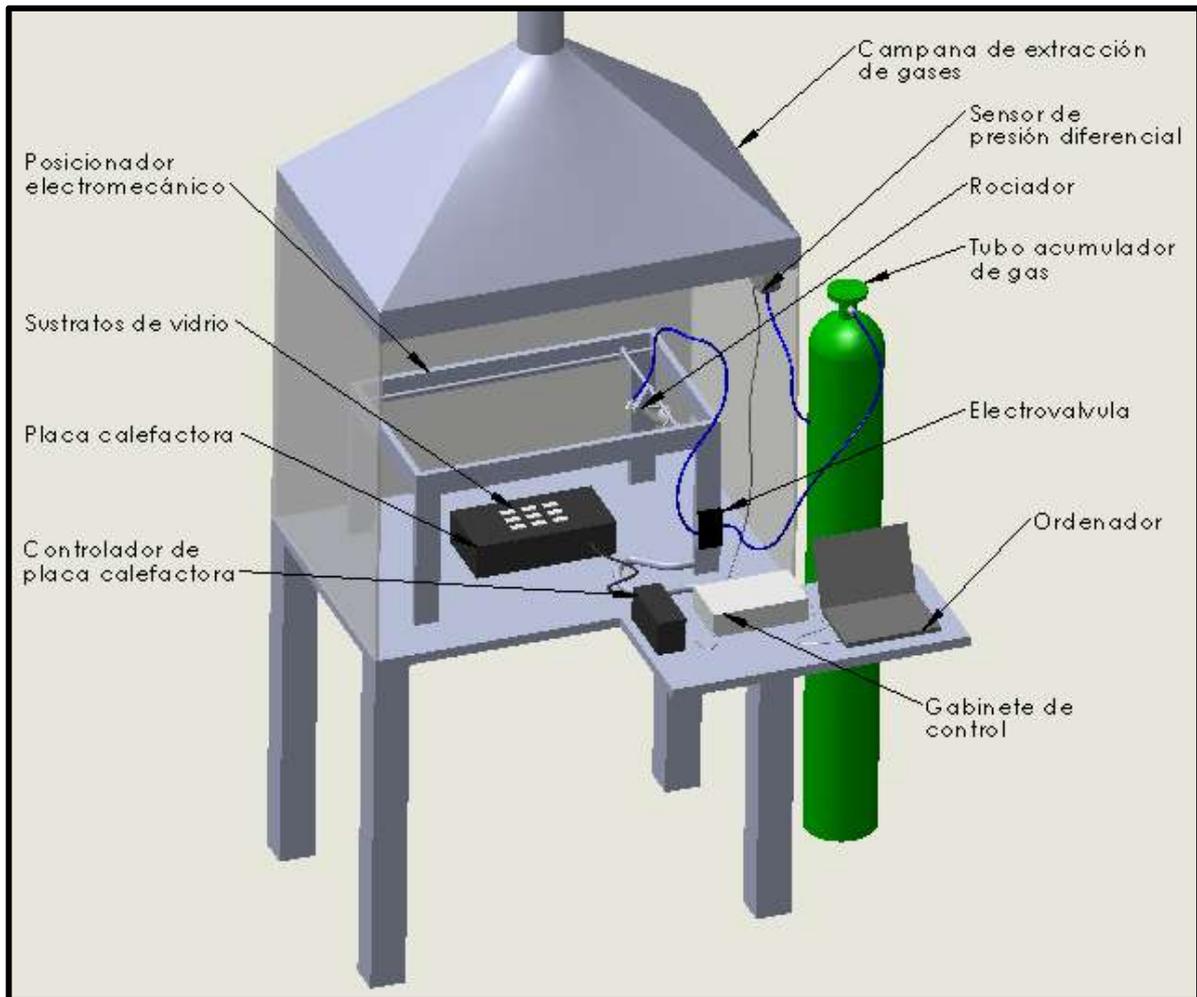


Figura 1.2 – Esquema del sistema de depósito por spray.

En la figura 1.3 se intenta ilustrar el funcionamiento básico del equipo a construir y la función de cada parte del mismo.

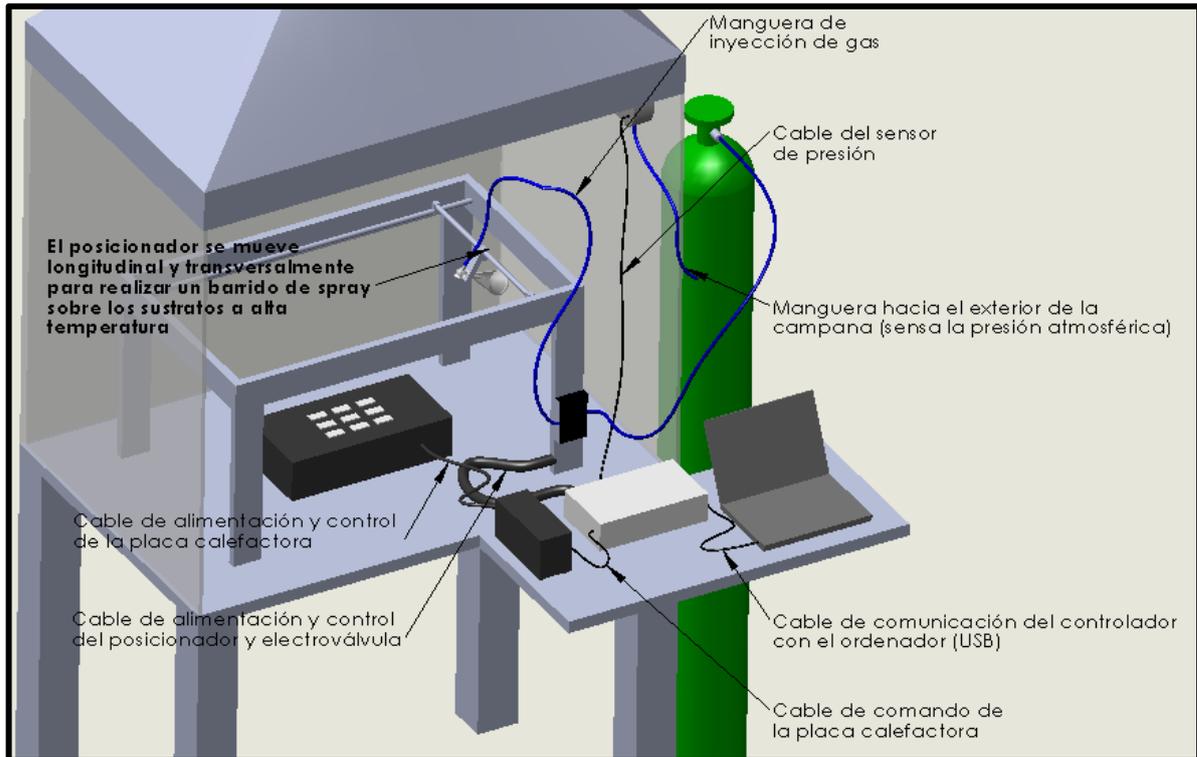


Figura 1.3 - Esquema explicativo de las partes del sistema de depósito por spray

1.1.2 – Recubrimiento por inmersión (dipping)

El recubrimiento por inmersión es una manera de crear laminas delgadas en donde un sustrato es introducido dentro de la solución deseada, luego se lo deja sumergido por un tiempo determinado hasta que comienza a ser elevado a una velocidad constante, la velocidad de extracción es la variable principal de este proceso ya que a una velocidad de extracción mayor se produce una capa de recubrimiento más gruesa. Luego de retirado el sustrato, el solvente de la solución se evapora formando la lámina delgada. En la figura 1.4 se ilustra el sistema explicado anteriormente, el cual sería implementado con el mismo sistema de posicionamiento que para el caso de depósito por spray ilustrado en las figuras 1.2 y 1.3.

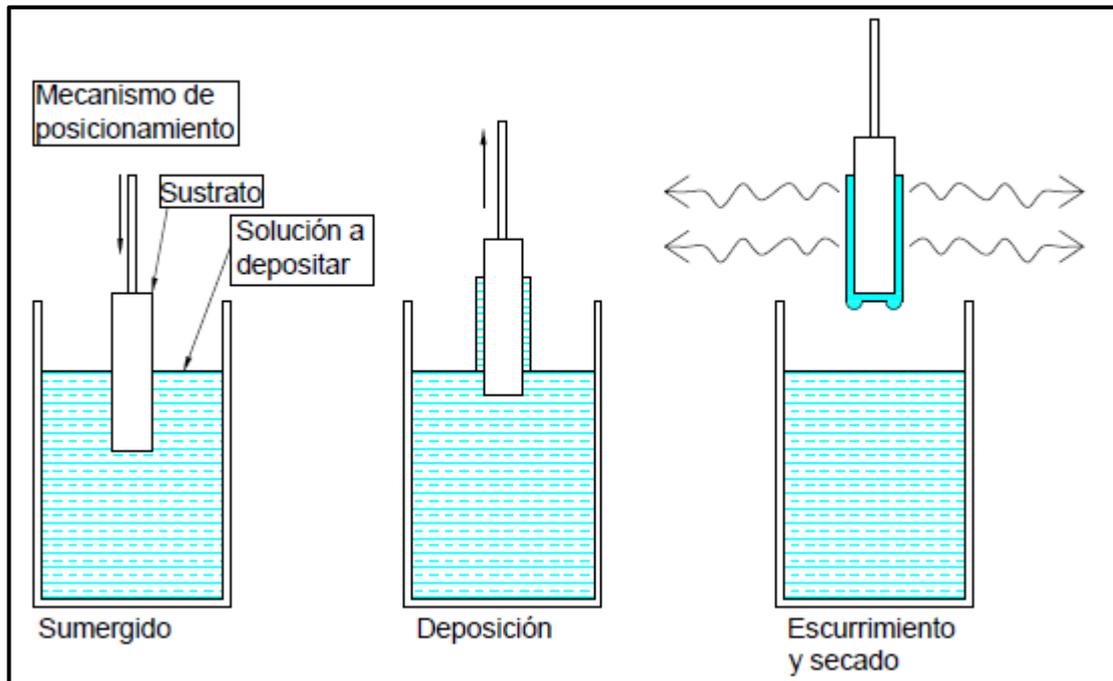


Figura 1.4 – Esquema del mecanismo de depósito por dipping.

Como punto de partida para el diseño de la máquina nos basamos en los requerimientos con los que debía cumplir el mecanismo como ser, longitudes de desplazamiento, velocidades y las medidas máximas de la máquina para que entren dentro de la campana de extracción de gases.

Consultando con quien sería el usuario de la máquina se determinaron los siguientes valores de velocidades máximas: 5mm/s en el eje longitudinal (dirección de deposición), 2 mm/s en el eje transversal y 2 mm/s en el eje vertical. Los desplazamientos deberían ser de 600 mm en el eje longitudinal, 300 mm en el eje transversal y 25 mm en el eje vertical.

Las medidas de la campana son de 1100mm de largo, 700mm de ancho y 700mm de alto.

1.2 – OBJETIVOS ESPECÍFICOS

Como objetivos específicos se pueden enumerar los siguientes:

1. Investigación de los diseños y las características de las máquinas de control numérico disponibles en el mercado;
2. Diseño de un posicionador apto para realizar depositados por spray y dipping cumpliendo con las restricciones antes mencionadas;

3. Análisis de los materiales disponibles en el mercado local;
4. Construcción de la maquina;
5. Desarrollo de la electrónica necesaria para el control de los motores y de la placa calefactora;
6. Diseño de un software amigable para la supervisión y control del posicionador;
7. Implementación de un sistema de monitoreo de la presión diferencial entre el interior de la campana de seguridad y el ambiente para asegurar una correcta extracción de gases.

CAPITULO II DISEÑO

Para poder desarrollar las tareas necesarias, nuestra maquina debía ser capaz de realizar movimientos controlados a lo largo de tres ejes ortogonales. Hemos hecho una investigación para poder determinar cuál era el diseño más conveniente y para saber cuáles son las máquinas de similares características que se encuentran en el mercado.

Sin duda alguna, una máquina que desarrolla una tarea muy similar a la que debe realizar la nuestra es la fresadora de Control Numérico por Computadora o fresadora CNC. Afortunadamente, hay mucha información y muchos diseños en la web, gracias a la filosofía “hágalo usted mismo” o DIY (“do it yourself”) en donde distintas personas alrededor del mundo suben sus diseños y recomendaciones para permitir que cualquier persona acceda al diseño y así poder utilizarlo en su proyecto.

Actualmente las máquinas de mecanizado por CNC al igual que las impresoras 3D son herramientas que a casi cualquier persona le gustaría tener en la casa y gracias al DIY, hoy es posible lograrlo a un costo no muy elevado. Es por esto que los diseños son muy variados y comprenden muchas variantes.

Para proceder explicando como hicimos nuestro diseño, primero vamos a dar una introducción acerca de las máquinas de movimiento controladas por computadora.

2.1 – INTRODUCCIÓN

Si para la mecanización total de un número de piezas fuera necesario realizar las operaciones de fresado, mandrilado y perforado, es lógico que se alcanzaría la mayor eficacia si este grupo de máquinas herramientas estuvieran agrupadas, pero se lograría una mayor eficacia aún si todas estas operaciones se realizaran en una misma máquina. Esta necesidad, sumada a numerosos y nuevos requerimientos que día a día aparecieron, ha forzado la utilización de nuevas técnicas que reemplazaran al operador humano. De esta forma se introdujo el control numérico en los procesos de fabricación, impuesto por varias razones:

- Necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes sin recurrir a la automatización del proceso de fabricación.

- Necesidad de obtener productos hasta entonces imposibles o muy difíciles de fabricar, por ser excesivamente complejos para ser controlados por un operador humano.
- Necesidad de fabricar productos a precios suficientemente bajos.

La primera máquina CNC, fue creada en el Instituto Tecnológico de Massachusetts (MIT) en 1952. Era una fresadora Cincinnati modificada y tenía la capacidad de coordinar los movimientos de sus ejes de coordenadas para mecanizar una superficie compleja, la figura 2.1 la ilustra junto con los controladores electrónicos de la época.



Figura 2.1 – Fresadora marca Cincinnati modificada en el MIT (1952).

Hoy en día el desarrollo tecnológico permitió implementar el ordenador tanto en la etapa del diseño con las herramientas CAD (“Computer Aided Design” o diseño asistido por computadora) como en la etapa de fabricación de las mismas gracias a los sistemas CAM (“Computer Aided Manufacturing” o fabricación asistida por computadora), la figura 2.2 muestra dos máquinas CNC modernas.



Figura 2.2 – Torno CNC Cancelarich (a la izquierda) y fresadora CNC de la misma marca (a la derecha).

La Fabricación Asistida por Ordenador es muy ventajosa respecto de los procedimientos de fabricación mediante operadores humanos. Por lo general, los equipos CAM conllevan la eliminación de los errores del operador y la reducción de los costes de mano de obra. Sin embargo, la precisión constante y el uso óptimo previsto del equipo representan ventajas aún mayores.

Los equipos CAM se basan en una serie de códigos numéricos, almacenados en archivos informáticos, para controlar las tareas de fabricación. Este Control Numérico por Computadora (CNC) se obtiene describiendo las operaciones de la máquina en términos de los códigos especiales y de la geometría de formas de los componentes, creando archivos informáticos especializados o programas de piezas. La creación de estos programas de piezas es una tarea que, en gran medida, se realiza hoy día por software informático especial que crea el vínculo entre los sistemas CAD y CAM.

Básicamente hasta ahora hemos hablado de las máquinas de control numérico, las cuales consisten básicamente en un mecanismo de posicionamiento automatizado el cual sigue una trayectoria específica la cual se genera con las herramientas CAD/CAM y en una herramienta de corte que se ocupa de retirar el material que se encuentra en su camino.

En los últimos años se han puesto de moda las impresoras 3D las cuales están teniendo una repercusión muy importante en la manufactura de piezas. Básicamente consisten en un mecanismo de posicionamiento semejante al de las maquinas herramientas de control numérico pero con la gran diferencia de que en vez de

utilizar una herramienta para arranque de viruta se utiliza un inyector que adiciona material a medida que el posicionador se mueve.

Existen una amplia variedad de impresoras 3D en el mercado. Sobre todo en estos últimos tiempos en donde se han globalizado las practicas DIY. De esta manera se pueden encontrar en la web maquinas comerciales de gran complejidad y otras más sencillas hechas en casa. En la figura 2.3 se pueden observar una variedad de impresoras 3D de distintos diseños, tamaños y niveles de sofisticación, las cuales en su mayoría se adquieren comprando un kit de componentes fundamentales para que la maquina sea funcional, como ser los motores, la electrónica, las guías lineales, etc., y la construcción finaliza imprimiendo las piezas secundarias.

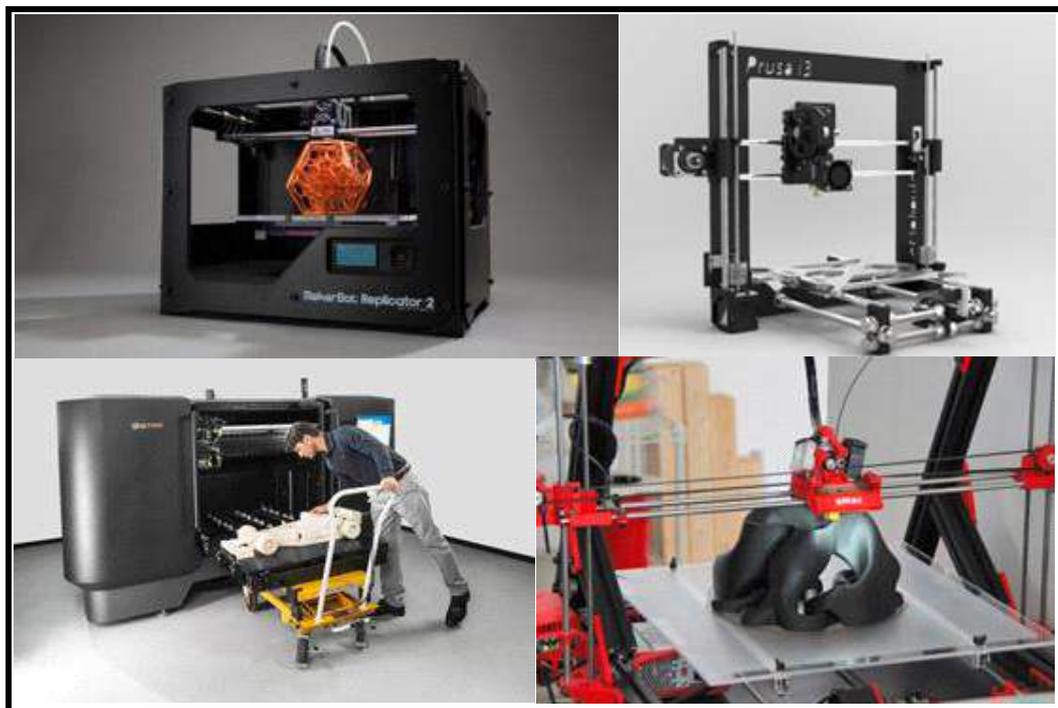


Figura 2.3 - Variedad de impresoras 3D comerciales

Hoy en día las máquinas de control numérico abarcan un campo importante en la automatización industrial. Así es como existen máquinas para distintas aplicaciones como puede ser mecanizar una pieza de aluminio con una fresadora, cortar madera con láser, y hasta cortar planchas de acero de hasta seis pulgadas de espesor. En la figura 2.4 se muestran una cortadora laser y una por plasma. Esto hace que en el mercado se puedan encontrar una variedad de máquinas de distintos tamaños,

potencias y hasta incluso distinta cantidad de ejes de movimiento o “grados de libertad”.

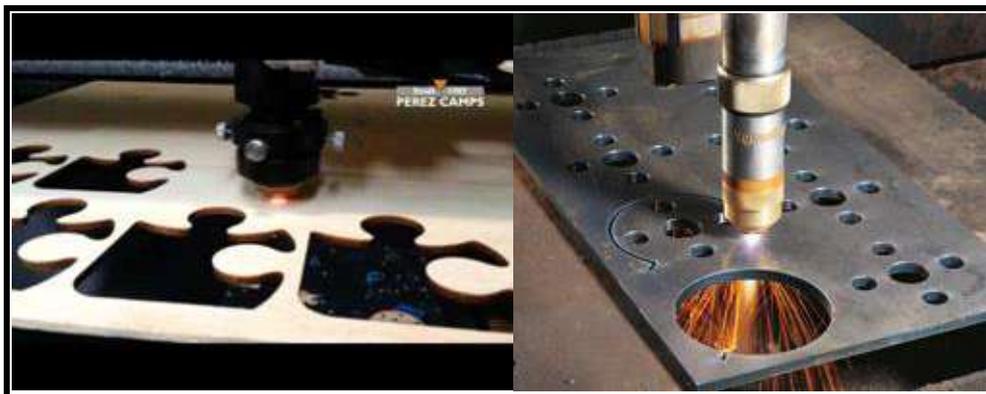


Figura 2.4 - Cortadora laser (izquierda) y cortadora por plasma (derecha)

Una característica que es de interés para este trabajo son los distintos mecanismos para la transmisión del movimiento. A continuación se dan más detalles de ellos.

2.1.1 – Mecanismos de transmisión del movimiento

Existen gran variedad de mecanismos para transmitir potencia mecánica. Fundamentalmente lo que se busca en ellos es transformar un movimiento rotativo en un movimiento traslacional. A la hora de elegir de qué manera utilizar la potencia de un motor para desarrollar un movimiento de traslación, es necesario analizar las ventajas y desventajas de cada sistema en función de la aplicación. A continuación se describen brevemente cada uno.

Transmisión por correa dentada y polea

A primera vista resulta muy tentador el precio de compra de un router a transmisión por correa. La economía y el tiempo necesario para su montaje final, termina siendo un ahorro fundamental de costos, comparado con el valor y la cantidad de horas hombre que requiere instalar un tornillo de precisión o una cremallera. En la figura 2.5 se puede observar un montaje de este sistema en un router CNC.

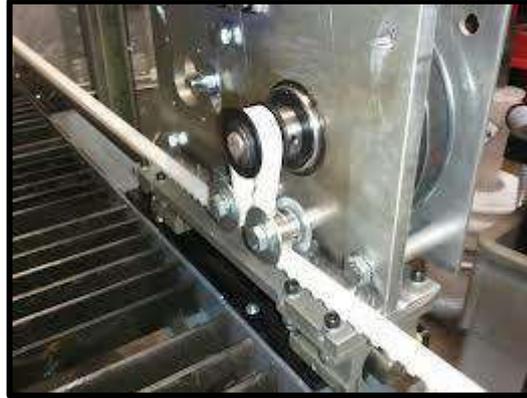


Figura 2.5 - Sistema de transmisión por correa dentada y polea.

Entre las virtudes se puede destacar que la correa funciona como un fusible mecánico entre las partes rígidas de la máquina, absorbe vibraciones, no requiere lubricación y es de rápido montaje. Por otro lado entre las desventajas, es necesario realizar un cambio de correa periódicamente debido a que estas sufren un rápido desgaste y luego de un determinado tiempo de uso se cortan, por otra parte el eje del motor se encuentra solicitado permanentemente para tensar la correa lo cual provoca la fatiga del mismo y posteriormente la rotura como se ilustra en la figura 2.6.



Figura 2.6 - Motor paso a paso con el eje partido.

Transmisión por cremallera de precisión y piñón

Entre sus virtudes ofrecen: solidez, precisión en largos desplazamientos y velocidad aceptable. Alternativa ideal para instalar en routers de robusta estructura, con puentes sobredimensionados, motores de fresado de gran torque y peso. Este sistema garantiza décadas de vida útil, ya que la aleación metálica de la cremallera es de mayor dureza que la del piñón, elemento de menor costo en caso de necesitar recambio por desgaste. Entre las desventajas: requieren lubricación, limpieza y mantenimiento general además de su alto costo inicial y dificultad en el montaje ya que requiere una gran exactitud en la alineación. En la figura 2.7 se muestra un ejemplo del montaje del sistema.

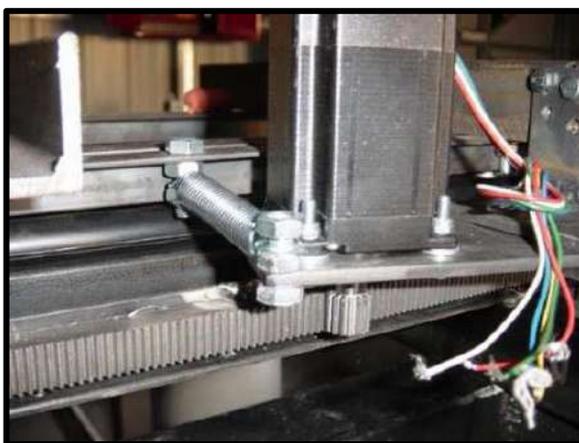


Figura 2.7 - Sistema de transmisión por cremallera y piñón.

Transmisión por tornillo sin fin

El tornillo sin fin es un mecanismo en el cual se transforma un movimiento rotacional de un tornillo, en uno longitudinal gracias a la acción de la geometría helicoidal de sus filetes. Generalmente se trata de un tornillo de gran diámetro para que el paso garantice un desplazamiento significativo en cada vuelta. Entre las ventajas se puede destacar que proporciona una buena precisión, sirve para grandes desplazamientos y no requiere de mucho mantenimiento más que garantizar la lubricación y la limpieza. Entre las desventajas, son de difícil alineamiento en el montaje y la alta fricción exige que los motores posean un mayor torque. Existe una amplia variedad de perfiles de roscas, como las que se muestran en la figura 2.8.

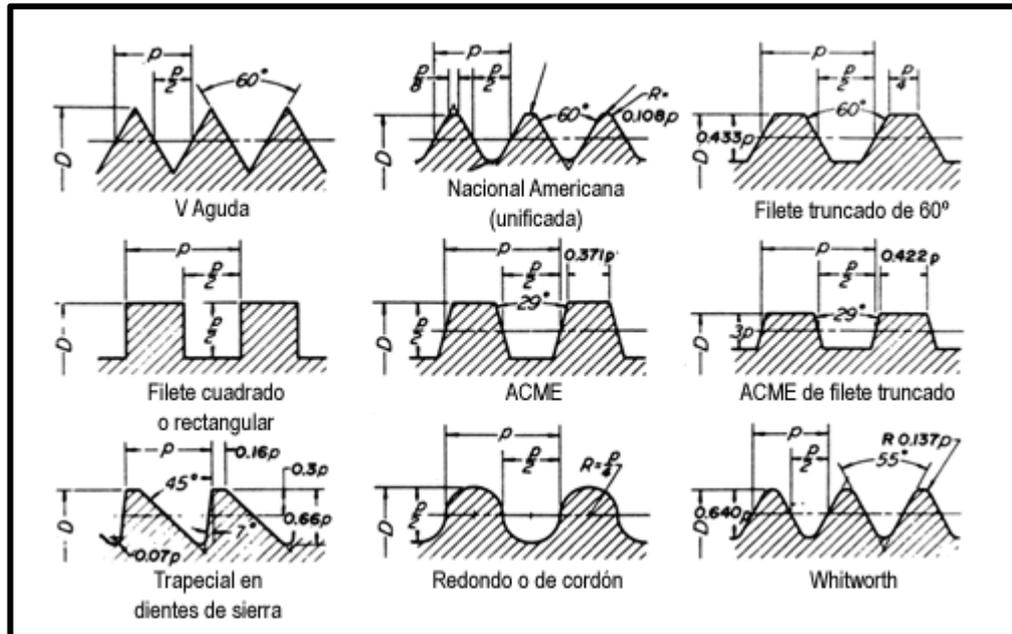


Figura 2.8 - Perfiles de roscas para tornillos.

Cada uno de estos perfiles posee distintos rendimientos resultando el perfil ACME el más conveniente para la aplicación en transmisión de movimiento por su baja fricción.

Además existe otro tipo de tornillo llamado “tornillo de bolas recirculantes” como el mostrado en la figura 2.9, en donde se garantiza una marcha suave, “muy liviana y cero juego” y mínimo rozamiento, gracias a que entre la tuerca y el husillo existe unas bolillas en rodadura con ambos cuerpos al igual que entre las pistas de un rodamiento rígido de bolas.



Figura 2.9 - Tornillo y tuerca de bolas recirculantes.

Cuando una bolilla llega al fin de su recorrido, es conducida otra vez hacia el extremo opuesto de la tuerca para volver a alinearse en la hilera y recircular nuevamente con toda independencia y sin ningún tipo de impedimentos con

respecto al tornillo, porque este sistema de circulación está especialmente diseñado para que el rozamiento de cada bolilla con la superficie del tornillo sea despreciable, logrando de esta manera una insignificante pérdida de potencia por fricción.

2.1.2 – Estructuras y mecanismos básicos

En general los routers CNC se componen de tres ejes ortogonales los cuales tienen la posibilidad de moverse independientemente. Para esto se podrían adoptar múltiples configuraciones como las que adoptan las máquinas de medición por coordenadas:

- “Cantiléver” o de brazo en voladizo:
Este tipo presenta inconvenientes en la rigidez motivo por el cual no es utilizado en routers CNC debido a que en piezas pequeñas se obtendrían grandes errores por la falta de precisión (figura 2.10).

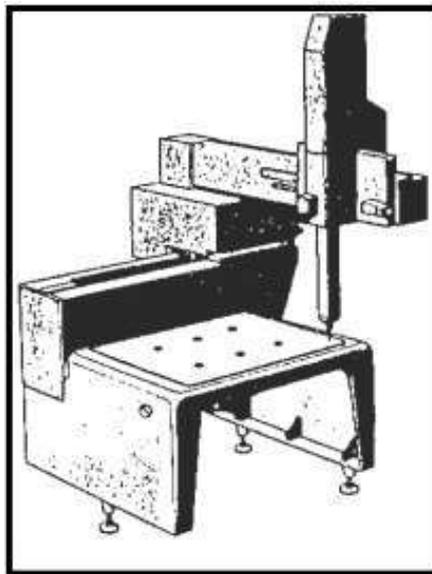


Figura 2.10 - Máquina de medición por coordenadas de configuración "Cantiléver".

- De tipo puente:
Este tipo es el más utilizado en Routers CNC ya que garantiza una mejor rigidez, lo cual hace que sea factible realizar una estructura más grande y por ende mecanizar piezas de mayor longitud (figura 2.11).

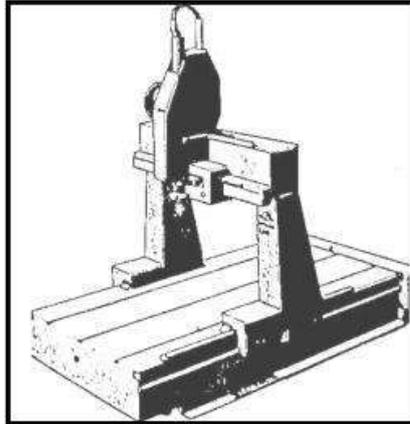


Figura 2.11 - Fresadora CNC de configuración Puente.

- Tipo de “pilares”:

Cuando la sección transversal es muy grande, crece la inercia de las partes móviles, reduciéndose la aceleración obtenible en los movimientos del eje longitudinal, lo cual hace que los tiempos de mecanizado o de medición se vuelvan excesivamente prolongados.

Como en la estructura de puente tenemos, además de la viga horizontal, columnas verticales que se mueven longitudinalmente, cuando se sobrepasa un cierto tamaño de sección transversal, es preferible pasar a la estructura de pilares, que permite ahorrar dichas columnas, el sistema es mostrado en la figura 2.12. Un sistema similar es el que se utiliza en los puentes grúa, en donde el puente transversal se desliza sobre vigas empotradas longitudinalmente a una determinada altura con respecto al piso.

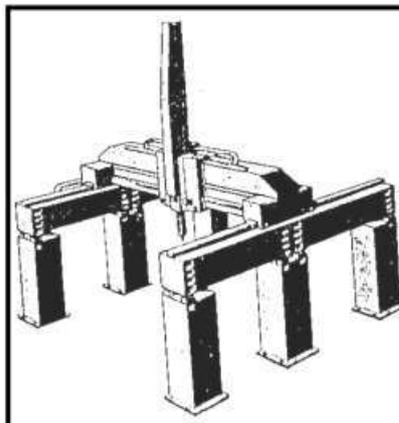


Figura 2.12 - Máquina de medición por coordenadas de tipo "de pilares".

2.2 – DISEÑO DE DEPOSITADORA POR SPRAY

Basándonos en los requerimientos del usuario diseñamos una máquina de dimensiones adecuadas para poder ser introducida dentro de la campana de extracción de gases y que fuera capaz de realizar movimientos a velocidad controlada y dentro de los rangos requeridos. Para ello utilizamos la configuración tipo puente, la cual fue mencionada anteriormente, ya que nos permitía tener una mejor rigidez del sistema en comparación con la “cantiléver” y al ser una máquina que no es sometida a esfuerzos muy grandes su robustez no requiere la utilización de una configuración del tipo de pilares.

En cuanto a los sistemas de transmisión de movimiento, inicialmente se había optado por utilizar tornillos de movimiento, los cuales nos iban a brindar una buena precisión. La gran desventaja que poseía este sistema es que en el movimiento longitudinal se requería una velocidad relativamente elevada, con lo cual necesitábamos conseguir un tornillo cuyo paso (milímetros por revolución) sea lo suficientemente grande para poder operar dentro de los límites de velocidad de los motores (de los cuales se hablará más adelante). Finalmente hemos optado por otro mecanismo para la transmisión del movimiento en el eje longitudinal, este mecanismo será explicado con más detalle más adelante en este capítulo.

A continuación se explicará cómo es el diseño de la maquina dividiendo este en tres módulos como son la estructura principal, el puente móvil y la torre, dejando los detalles constructivos para el capítulo siguiente.

2.2.1 – Diseño de estructura principal

Para diseñar la estructura principal nos basamos en las medidas de la campana de extracción de los gases y las longitudes de los movimientos a realizar, así pensamos en armar un chasis compuesto por cuatro patas, dos vigas longitudinales y dos vigas transversales. Dichas piezas se construirían con extrusión de aluminio estructural. Para realizar los encuentros pusimos chapas plegadas de manera de poder atornillarlas a las piezas a unir y así darle rigidez al ensamblaje. La figura 2.13 ilustra el ensamblaje explicado.

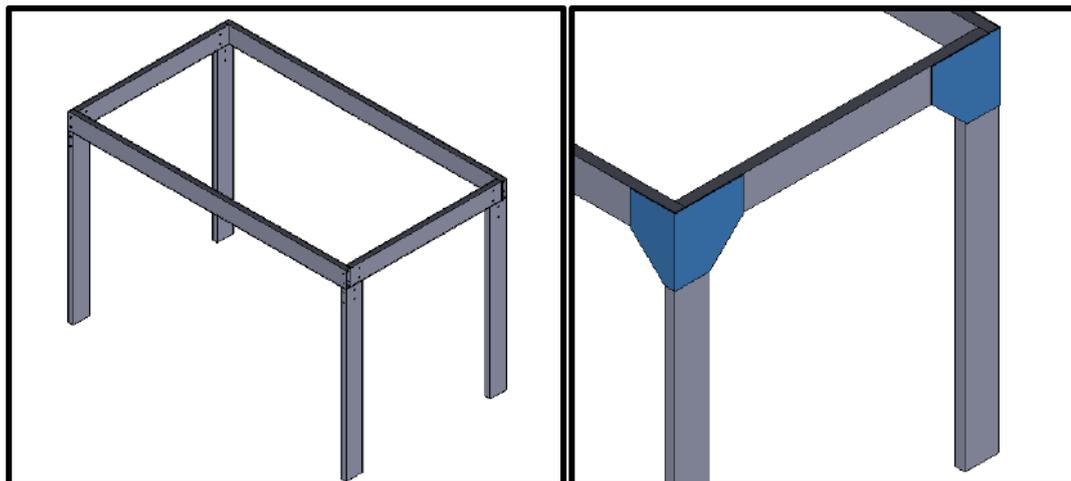


Figura 2.13 - Ensamblaje de patas y vigas (izquierda) y encuentros (derecha).

Para promover el desplazamiento del puente móvil sobre esta estructura se colocaron varillas cilíndricas amuradas las vigas transversales mediante soportes de sujeción. En la figura 2.14 se pueden apreciar este tipo de soportes sujetando la varilla contra la viga transversal.

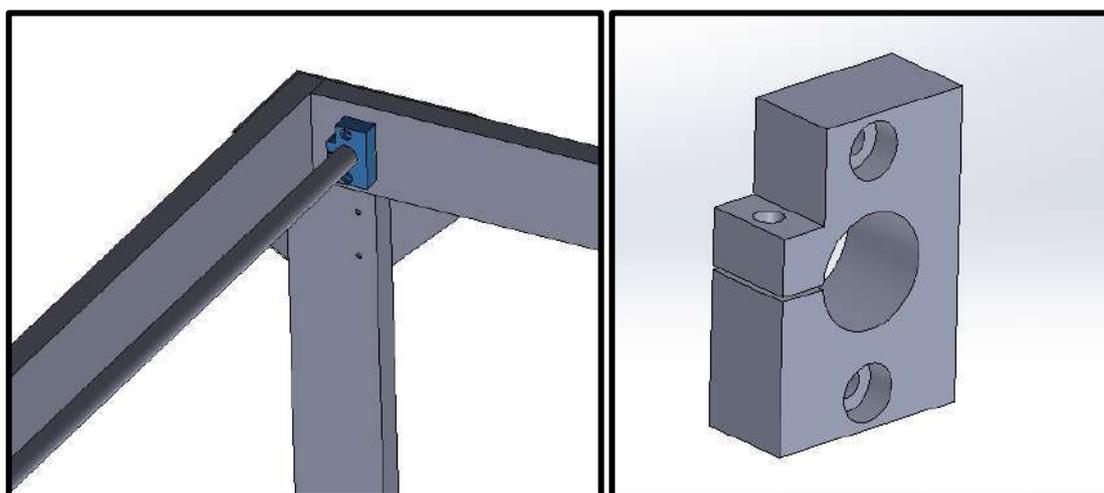


Figura 2.14 - Varilla montada en la estructura (izquierda) y soporte utilizado para el montaje (derecha).

Como se mencionó anteriormente, al iniciar se pretendió utilizar un tornillo de un paso amplio para poder darle el movimiento al puente móvil. Ese tornillo debía estar amurado a las patas de nuestro chasis implementando algún sistema de montaje que utilice rodamientos de bolas y así facilitar el movimiento de este. En la figura 2.15 se muestra un esquema aproximado de cómo iba a ser el montaje del tornillo.

Como ya se explicó, una limitante de este sistema era la velocidad máxima a la que se podía operar el motor. Otro problema importante que posee este sistema, es que requiere un montaje muy preciso en cuanto al alineamiento del tornillo con las dos varillas por sobre las cuales deslizaría el puente móvil, lo cual se torna una tarea compleja ya que para lograr esto las tolerancias de fabricación deberían ser demasiado pequeñas, tarea que nos resultaría prácticamente imposible en estas longitudes.

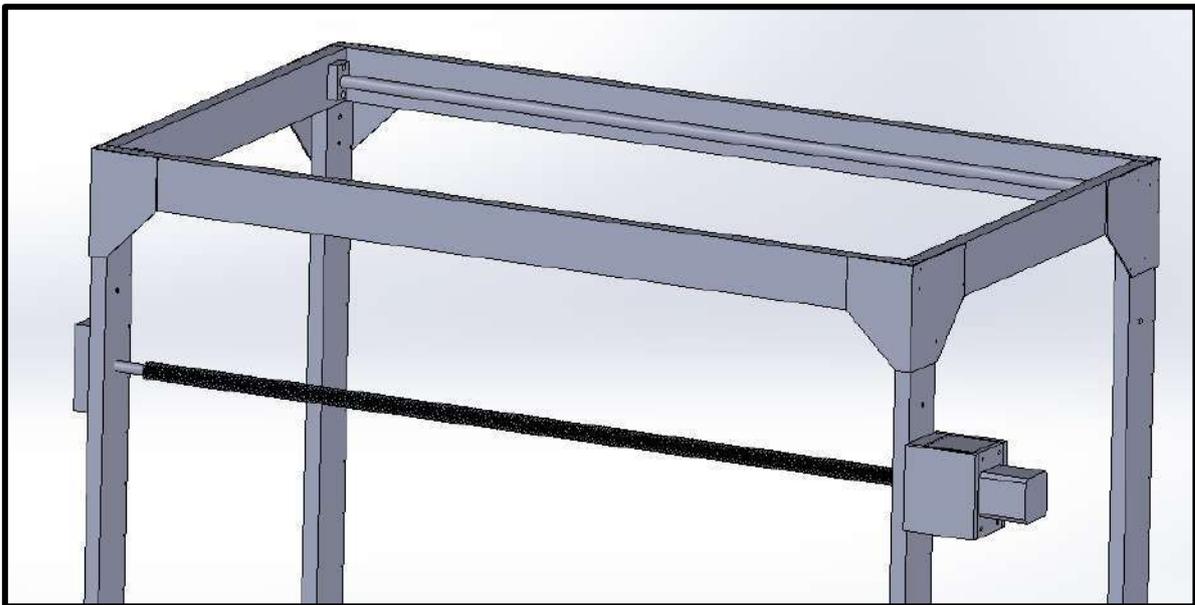


Figura 2.15 - Montaje del tornillo sin fin para la estructura principal.

Finalmente consultando el asunto con una persona experimentada nos recomendó utilizar un sistema en donde en el eje del motor se coloque un rodillo en el cual se enrolla una cuerda de acero cuyas puntas se encuentran fijas a las patas del chasis, así, cuando el rodillo gira, un extremo de la cuerda se tensiona más que el otro generando una fuerza resultante que provoca el movimiento del puente móvil. La figura 2.16 ilustra cómo se enrolla la cuerda según el sentido de giro del motor y la dirección del movimiento del puente.

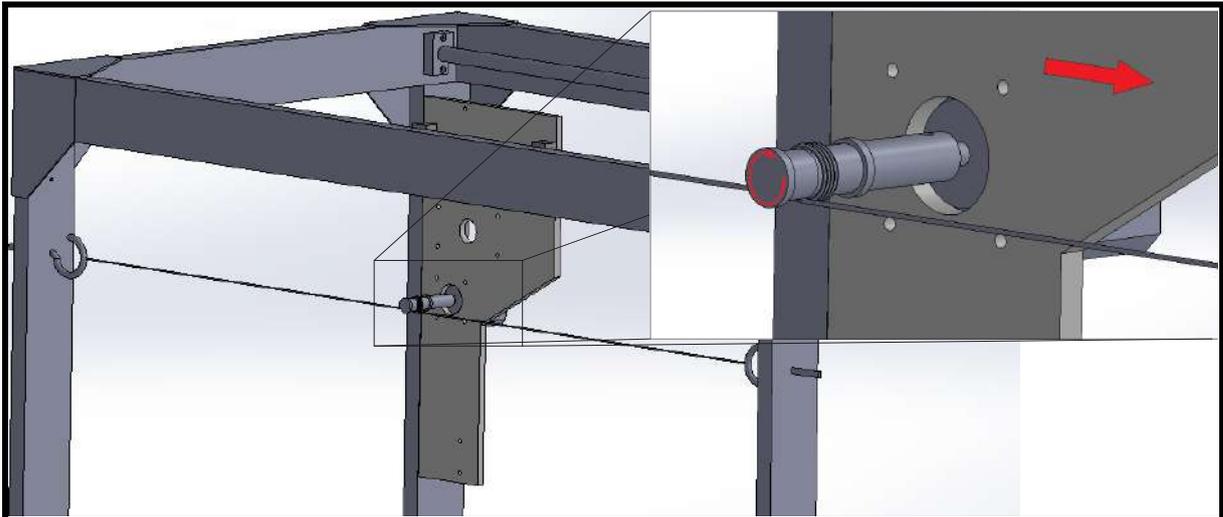


Figura 2.16 - Sistema de transmisión de movimiento a través de cuerda fija y rodillo.

En la figura se ve cómo se realiza el montaje del motor sobre el puente móvil, cuyo diseño será explicado en mayor detalle más adelante en este capítulo.

Este mecanismo de transmisión de movimiento consiste básicamente en provocar un movimiento sobre el puente móvil gracias a la acción de la fuerza de rozamiento en el rodillo, la cual genera una diferencia de tensiones entre un tramo y el otro de la cuerda. Es de esperarse que en este sistema exista un resbalamiento de la cuerda con respecto al rodillo, lo cual es una desventaja importante ya que hace que el sistema pierda exactitud considerablemente. Pero sin ir más lejos, en nuestra aplicación se requiere un movimiento a una velocidad controlable, en tal caso, el error absoluto que se pueda llegar a obtener en la posición del puente no es algo del cual debemos preocuparnos.

Este sistema presenta una serie de ventajas frente al sistema del tornillo sin fin. Una de ellas es las tolerancias de fabricación para el montaje de las piezas, ya que no se requiere una alineación tan precisa entre la cuerda y las guías lineales como si ocurría con el tornillo. La otra ventaja fundamental es que las velocidades alcanzables con este sistema son muy superiores ya que colocando un rodillo de 10mm de diámetro se produce un movimiento de aproximadamente 31mm por revolución, cuando en el caso del tornillo (utilizando un tornillo de rosca ACME de 5/8”) se iban a poder obtener solo 5mm por revolución, lo que hubiera implicado que el motor tuviese que girar 6 veces más rápido.

Como se verá más adelante, el puente móvil desliza sobre las varillas cilíndricas montadas en la estructura principal. Dicho deslizamiento conlleva a la existencia de una fuerza de rozamiento en cada uno de los apoyos, por lo cual si se coloca un motor sobre uno de los lados del puente, este tendería a cruzarse por la acción de las fuerzas de fricción en el otro lado. Para promover un movimiento uniforme se colocó un motor de cada lado y de esta manera, si se mantiene el sincronismo entre ellos, el puente no se cruzaría.

A medida que se fue avanzando con la construcción del prototipo, hemos notado cierta falta de rigidez en la estructura, por lo que nos vimos obligados a tener que reforzarla. Para ello decidimos agregarle a nuestro diseño varillas de sección cuadrada colocadas longitudinal y transversalmente a unos 150mm del límite inferior de las patas. Para fijarlas a la estructura utilizamos un método similar al ya utilizado para fijar las vigas superiores. En la figura 2.17 se muestran los refuerzos montados a la estructura con las correspondientes chapas plegadas para realizar los encuentros, dichas chapas se resaltan en color azul.

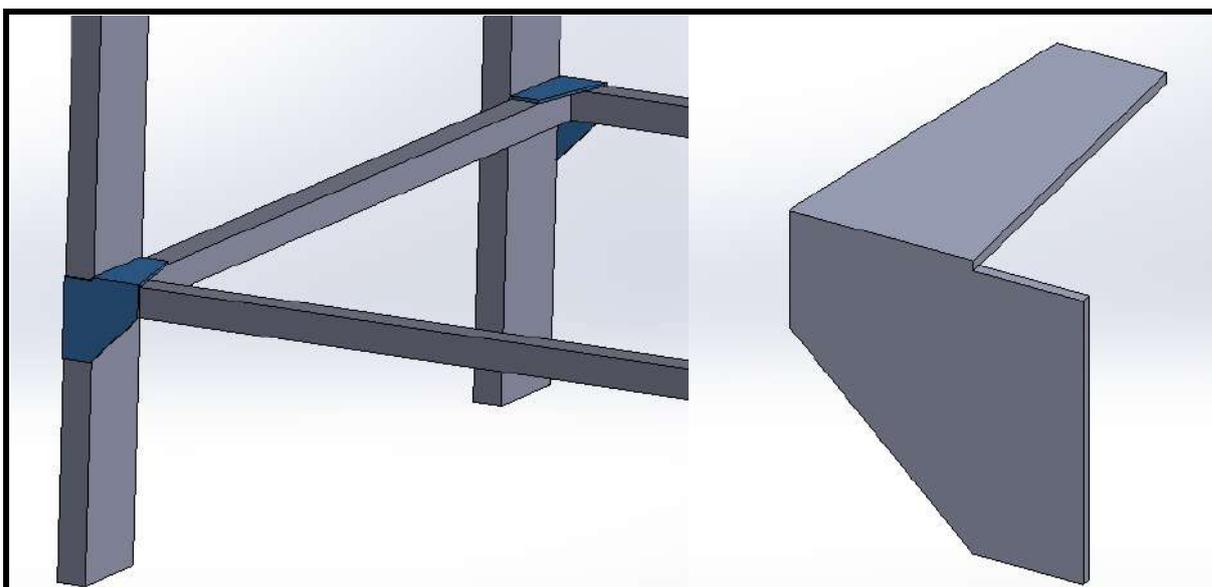


Figura 2.17 - Ensamblaje de los refuerzos (izquierda) y chapa utilizada para los encuentros (derecha).

Para reforzar aún más la estructura se colocaron chapas en la parte superior de los encuentros de las vigas superiores, las cuales son mostradas en la figura 2.18.

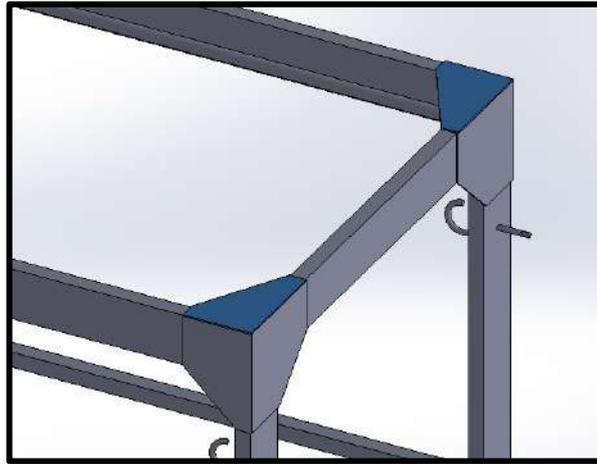


Figura 2.18 - Chapas triangulares para reforzar la estructura.

2.2.2 – Diseño del puente móvil

El puente móvil es la pieza que desliza longitudinalmente y contiene los elementos necesarios para promover el movimiento transversal de la torre. El diseño de dicho puente consiste en dos placas unidas entre sí mediante varillas cilíndricas similares a las utilizadas en la estructura principal, que funcionan a su vez como guías lineales por sobre las cuales desliza la torre. Estas varillas son sujetas a las placas mediante los mismos soportes que fueron utilizados en el caso anterior para sujetar las varillas al chasis de la máquina. En la figura 2.19 se muestra el ensamblaje de dichas placas con las varillas, los soportes se muestran en color azul para su distinción.

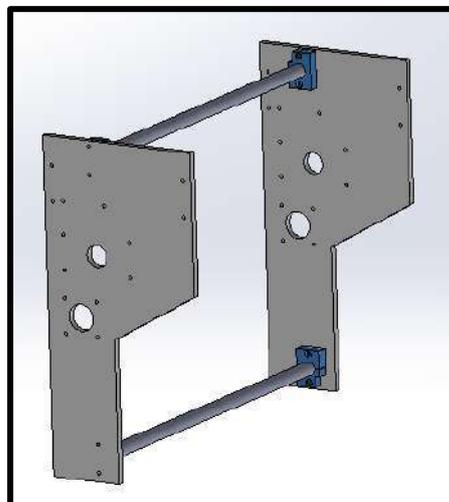


Figura 2.19 - Ensamblaje de las placas y las varillas cilíndricas.

Luego sobre estas placas se montaron las piezas que se encargarían de sostener el puente sobre las varillas que se encuentran sujetas al chasis de la máquina. En la figura 2.20 se puede observar cómo queda montado el puente móvil sobre la estructura principal con los respectivos apoyos resaltados en color azul.

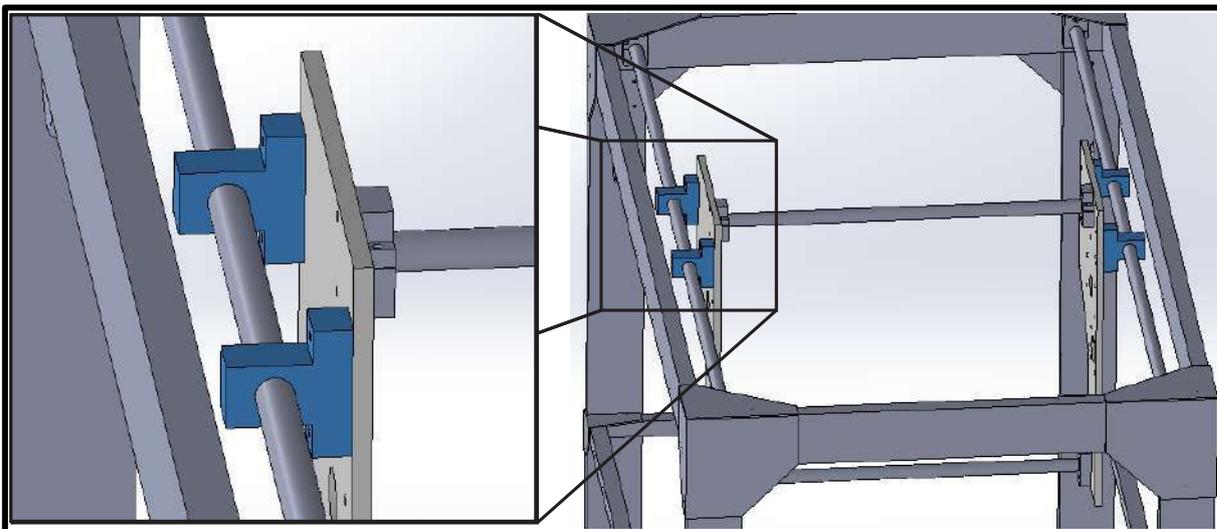


Figura 2.20 - Montaje del puente móvil a la estructura principal.

Luego, como ya se mostró anteriormente, en las placas van montados los motores con los respectivos rodillos encargados de generar el movimiento en el eje longitudinal de la máquina. En la figura 2.21 se aprecian dichos motores montados.

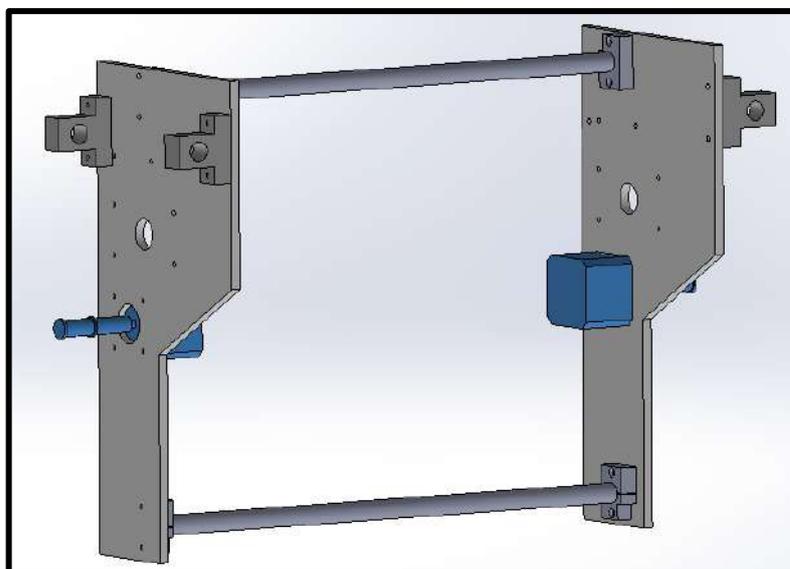


Figura 2.21 - Motores montados en las placas del puente.

Para generar el movimiento de la torre se optó por el sistema de tornillo sin fin. Fue necesario diseñar un sistema que permitiera sujetar dicho tornillo a las placas del puente manteniendo la alineación y un bajo rozamiento. Se utilizaron rodamientos rígidos de bolas para disminuir la fricción y se diseñaron aloja-rodamientos para poder fijarlos a las placas.

El proceso de diseño de este sistema requirió que realicemos una planificación adecuada de montaje para el tornillo, ya que este debía estar alineado con suficiente precisión y con el menor juego axial posible y a su vez debía ser fácil de extraer. Para ello, se implementó la configuración de las figuras 2.22 y 2.23 en donde se muestra que el tornillo ingresa al puente por uno de los laterales y hasta donde el conjunto hace tope.

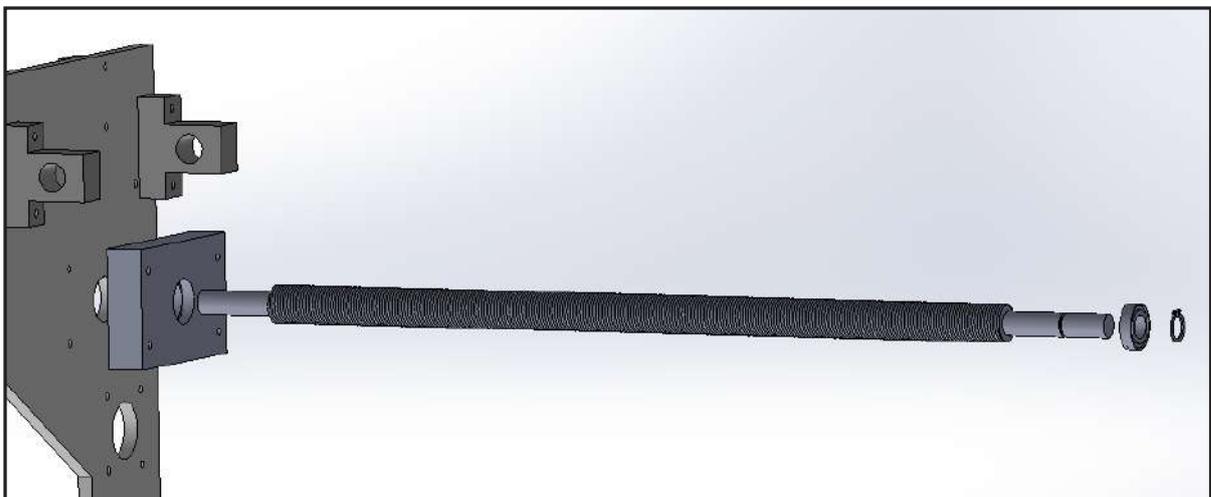


Figura 2.22 - Ensamblaje explotado del tornillo sin fin en el lado izquierdo del puente.

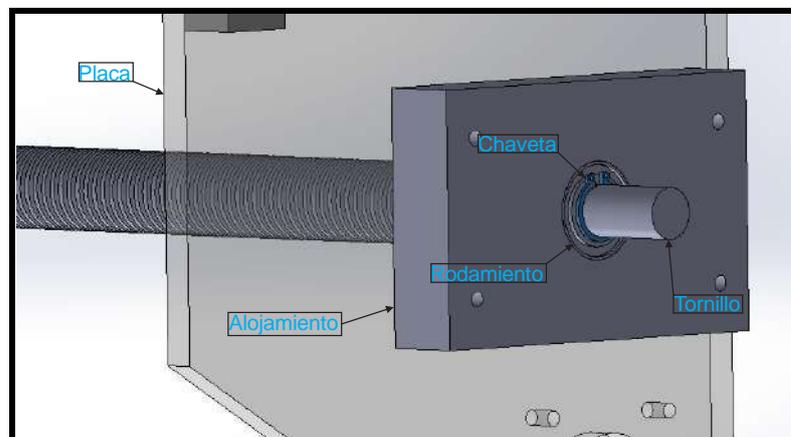


Figura 2.23 - Ensamblaje del tornillo en el lado izquierdo del puente.

Luego, sobre la placa opuesta del puente, es colocado el aloja-rodamiento y se finaliza el montaje con la colocación de un bulón que tracciona el tornillo hasta quitarle el juego. Este sistema es ilustrado en las figuras 2.24 y 2.25.

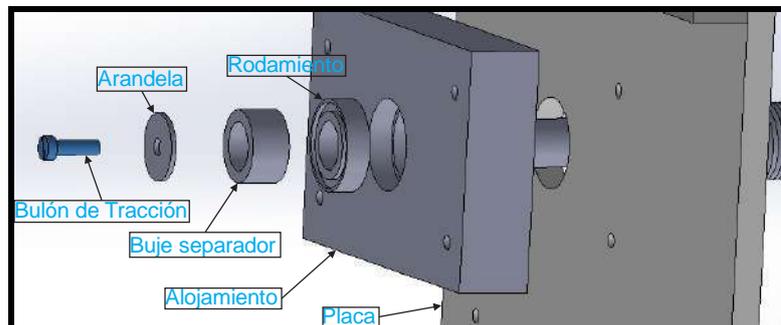


Figura 2.24 - Ensamblaje explotado del tornillo en el lado derecho del puente.

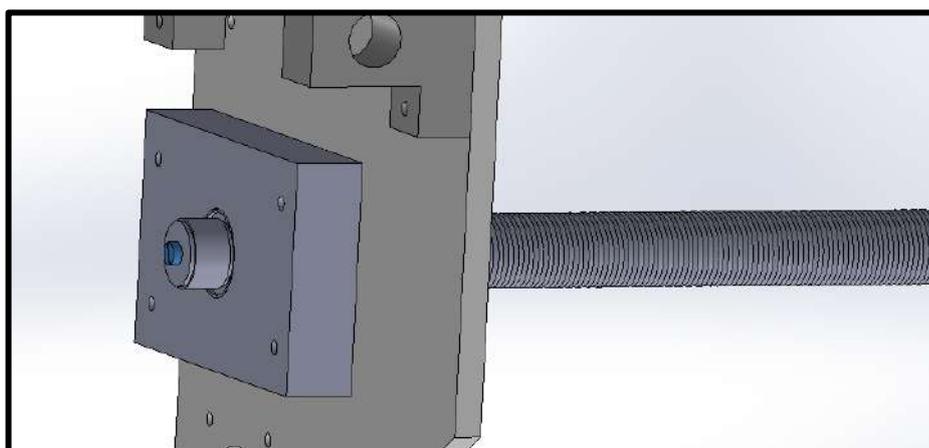


Figura 2.25 - Ensamblaje del tornillo en el lado derecho del puente.

Para el acople del motor con el tornillo se requería un cierto espaciado entre el eje del motor y el tornillo propiamente para poder colocar un acople elástico que absorba las diferencias de alineación de ambos ejes. Para ello se utilizaron separadores que se encontrarían fijos al alojamiento y en el otro extremo tendrían una placa en la cual estaría sujeto el motor. De esta manera el sistema queda como el ilustrado en la figura 2.26.

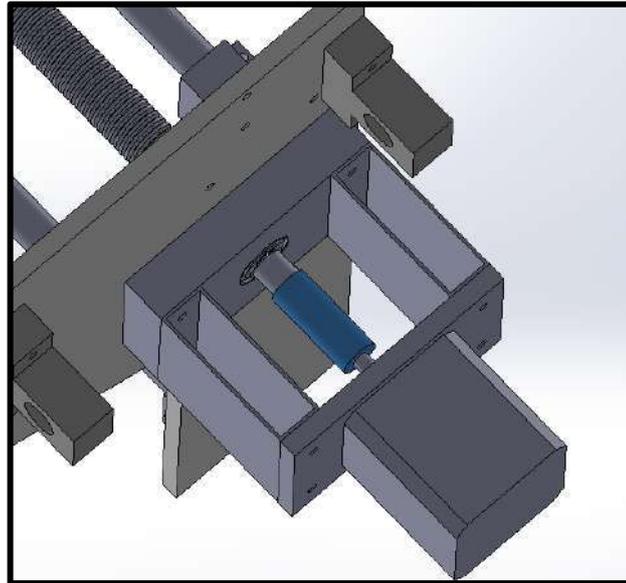


Figura 2.26 - Montaje del motor en el puente mediante separadores y el acople elástico.

Finalmente se colocaron unas tapas para que el acople elástico no quede a la vista y así evitar el contacto con esta pieza que se encontraría en movimiento.

Por último se colocó un refuerzo entre las placas para mejorar la rigidez del sistema. Este refuerzo consistió en una pieza de extrusión de aluminio estructural sujeta a ambas placas del puente, la figura 2.27 muestra el diseño final del mismo.

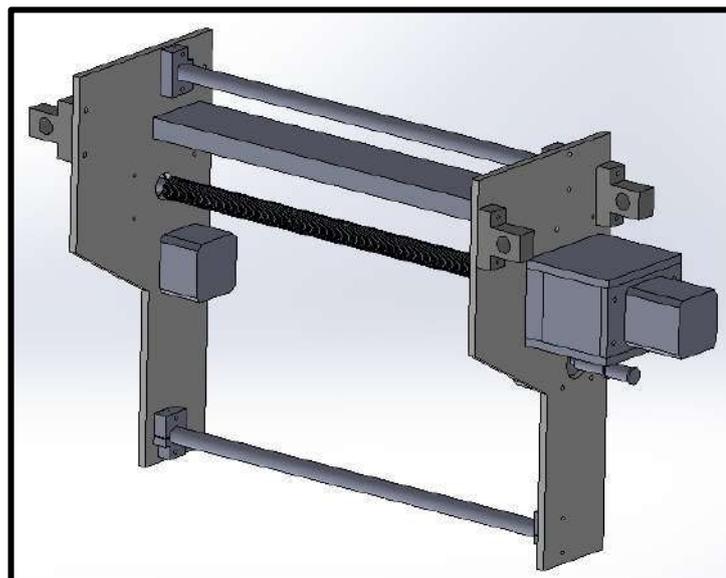


Figura 2.27 - Ensamblaje final del puente móvil.

2.2.3 – Diseño de la torre

La torre es la pieza que se posiciona transversalmente a lo ancho del puente y es la encargada de realizar el movimiento vertical en nuestra máquina. El diseño consiste básicamente en una placa rectangular a la que llamamos “espalda” la cual tiene sujetos en ella cuatro soportes que, al igual que en el caso del puente el cual desliza sobre las varillas de la estructura principal, deslizan sobre las varillas cilíndricas del mencionado puente.

Para poder llevar a cabo el movimiento transversal mediante el tornillo montado en el puente móvil, es necesario colocar la tuerca correspondiente. Dado que el tornillo no es perfectamente recto, se optó por utilizar una “media tuerca” de manera que se admita un pequeño desprendimiento del tornillo respecto de la misma y así reducir la fricción del sistema. Finalmente el conjunto espalda, soportes y tuerca se muestra en la figura 2.28.

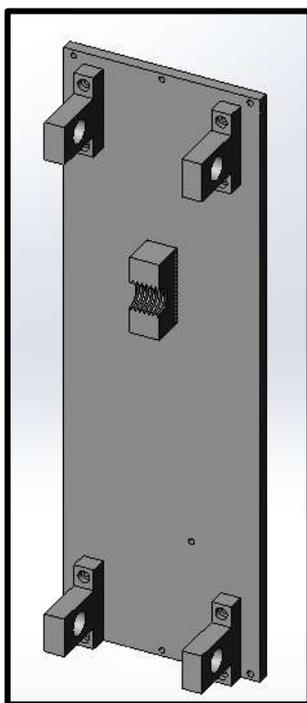


Figura 2.28 - Ensamblaje de la "espalda" con los soportes y la tuerca.

Luego se colocaron dos placas en los extremos superior e inferior de la espalda, sujetas a estas placas se colocaron dos varillas verticales que funcionarían como guías para el movimiento vertical. Dicha sujeción se realizó mediante soportes similares a los utilizados para amarrar las varillas tanto en la estructura principal

como en el puente móvil. En la figura 2.29 se ilustran las piezas mencionadas ensambladas a la espalda.

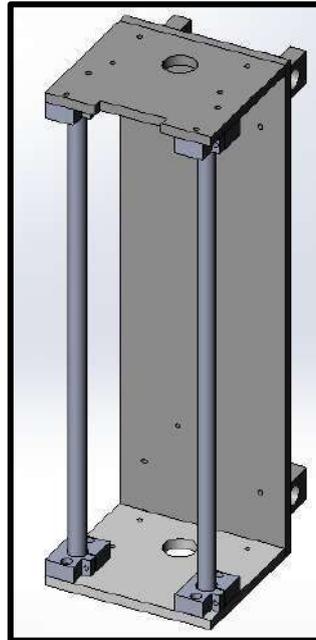


Figura 2.29 - Ensamblaje de las placas y las guías lineales a la espalda.

Luego se diseñó el sistema de transmisión del movimiento el cual consiste en el mismo principio que el caso del puente móvil, en donde el tornillo queda cargado a la tracción para limitar el juego axial. En la figura 2.30 se muestra la torre con el tornillo montado.

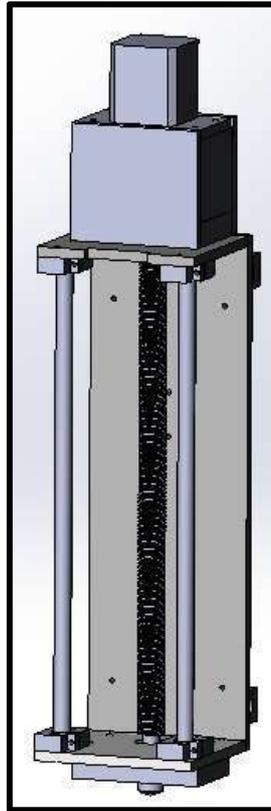


Figura 2.30 - Ensamblaje del tornillo sin fin y el motor a la torre.

La pieza que se mueve verticalmente consiste en otra placa, la cual posee dos orificios por donde las guías lineales la atraviesan y además tiene adosada la media tuerca por medio de la cual se produce el movimiento gracias a la rotación del tornillo. Se adicionó una prolongación vertical donde sería colocada la pieza de sujeción del Erlenmeyer o la de la pinza para dipping de las cuales se hablará más adelante. En la figura 2.31 se muestra en detalle la mencionada placa.

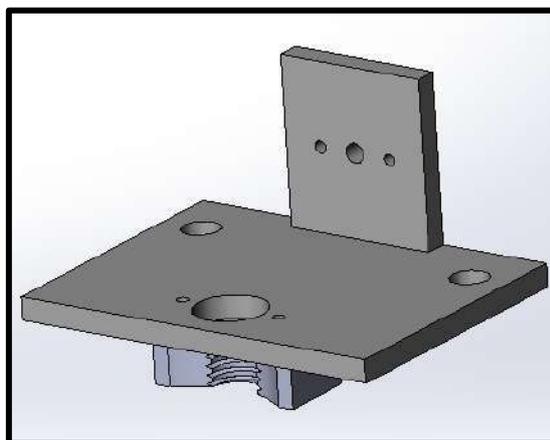


Figura 2.31 - Ensamblaje de placa de movimiento vertical con tuerca y prolongación vertical.

Esta configuración presentó el problema de que al ser las guías lineales no coplanares con el tornillo de movimiento y los orificios para las varillas levemente de mayor diámetro que estas (lo que conlleva a un juego amplio), se producía un cruce de la placa, lo que provocaba que las guías produjeran una fricción inadmisibles en los orificios. Para remediar esto se colocó una placa adicional, idéntica a la anterior, para hacer que la imprecisión de los agujeros no repercuta en un cruce tan considerable de la placa. Finalmente el diseño de esta pieza queda construido como se muestra en la figura 2.32. En la imagen 1.33 se muestra esta pieza ya montada en la torre.

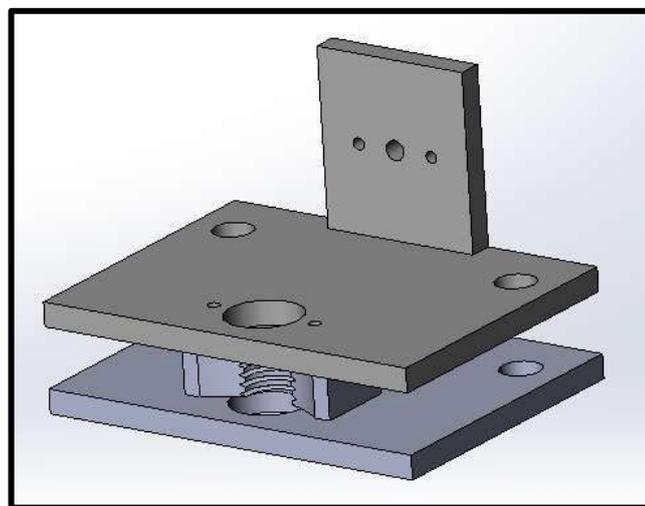


Figura 2.32 - Ensamblaje de placa de movimiento vertical.

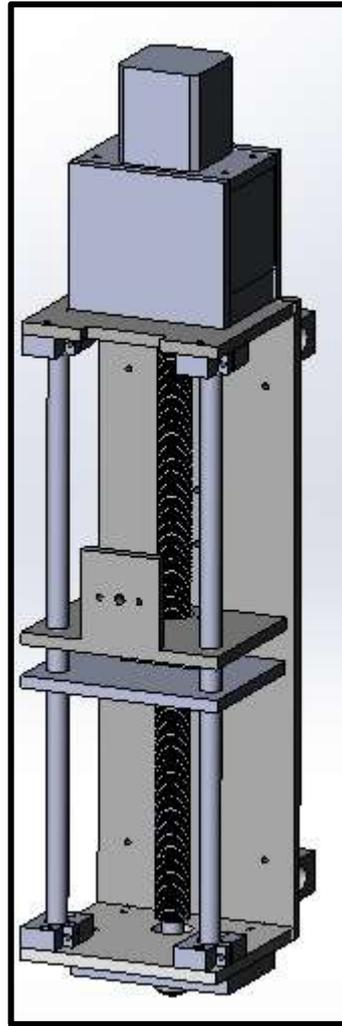


Figura 2.33 - Ensamblaje de la torre con la placa de movimiento vertical.

La pieza que va adosada a esta placa depende del proceso que se quiera realizar con la máquina, es decir, si se pretende utilizar para hacer depositado por spray, se utilizará la pieza de la figura 2.34, la cual permite sujetar el Erlenmeyer desde el cuello del mismo, y si se desea hacer depositado por dipping se utilizará la de la figura 2.35, la cual permite agarrar el sustrato con un pinza metálica para papeles.

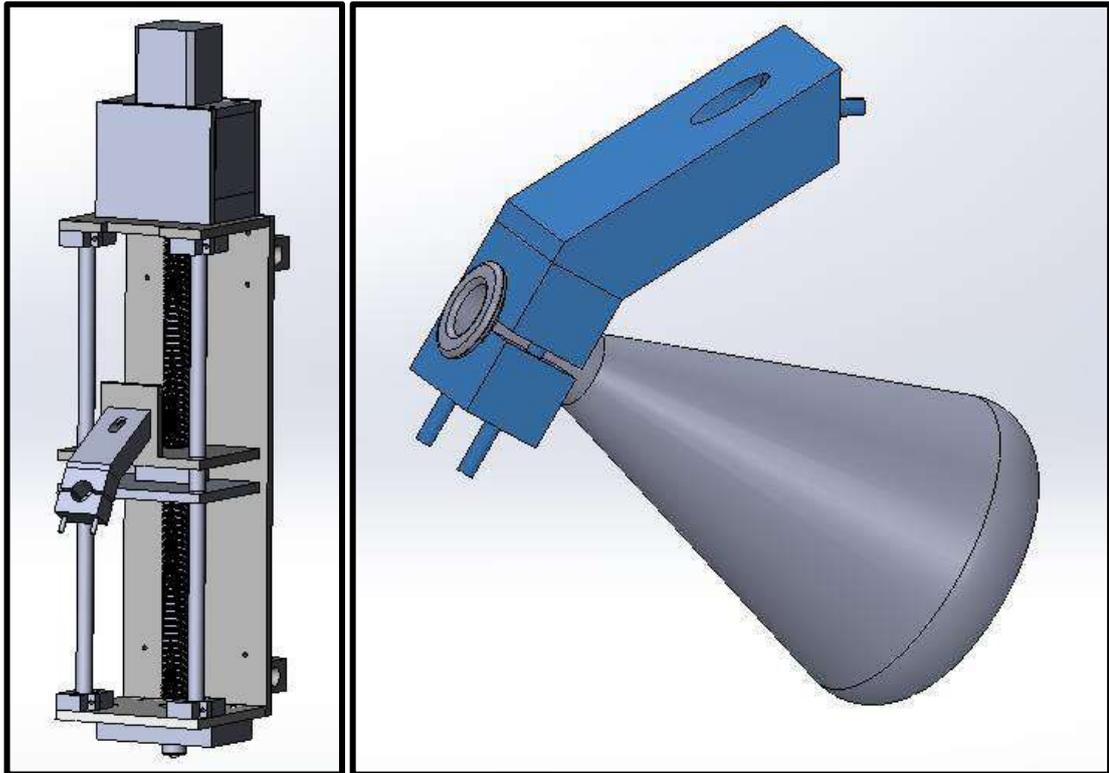


Figura 2.34 – Soporte para el Erlenmeyer.

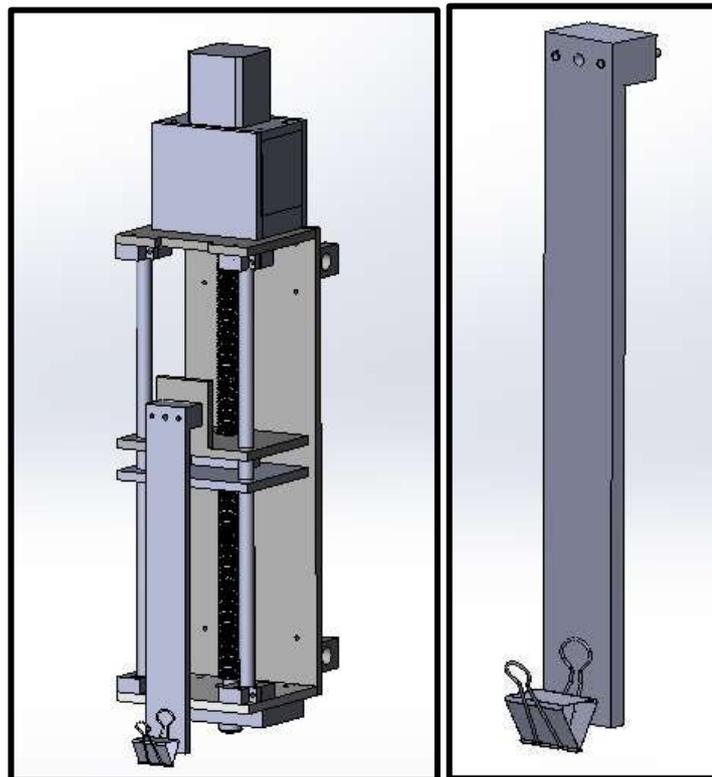


Figura 2.35 – Soporte para pinza de dipping.

CAPITULO III PROCESO DE FABRICACIÓN Y ENSAMBLAJE

3.1 – INTRODUCCIÓN

El proceso de fabricación de la máquina ha comenzado una vez finalizado el diseño de la estructura principal y el mecanismo de movimiento. Cabe aclarar que el diseño ha ido cambiando a lo largo de la construcción debido a que la fabricación ha sido un proceso de aprendizaje para nosotros y nos permitió ir mejorando el diseño a medida que avanzábamos.

En este capítulo explicaremos como fue construido el posicionador incluyendo la descripción de los materiales y herramientas utilizadas.

La construcción al igual que el diseño, se dividió en tres etapas, construcción de la estructura principal, construcción del puente móvil y construcción de la torre vertical, finalmente se realizó el montaje y cableado de los motores y sensores.

Cabe mencionar en esta introducción que debido a la atmosfera ligeramente corrosiva a la que se expondrá la máquina, fue necesario analizar que materiales a utilizar para la construcción de la misma.

Para la estructura de la maquina hemos utilizado perfilería de aluminio y para las piezas móviles, polietileno de alto peso molecular (APM), elección que se llevó a cabo luego de realizar el siguiente análisis:

Por un lado el Grilón® (poliamida 6) posee un coeficiente de rozamiento con el acero muy bajo y una muy buena resistencia al desgaste pero, en contrapartida, es altamente higroscópico, lo que lo hace no apto para atmosferas húmedas. El Delrin® (resina acetálica) es un plástico con buena resistencia al desgaste y a la corrosión y con una rigidez elevada. El APM posee menos rigidez que el Delrin® pero similares características anti-desgaste y anti-corrosivas a un costo considerablemente menor, por este motivo es que lo consideramos más conveniente para nuestra aplicación.

A continuación describiremos como fue el proceso de construcción de cada etapa.

3.2 – ESTRUCTURA PRINCIPAL

La estructura principal tal como se explicó en el capítulo dos, consta de cuatro vigas (dos longitudinales y dos transversales), cuatro patas y cuatro refuerzos (al igual que en el caso de las vigas, dos longitudinales y dos transversales) estos fueron contruidos a base de perfiles de aluminio estructural y cuyas medidas se muestran en la figura 3.1.

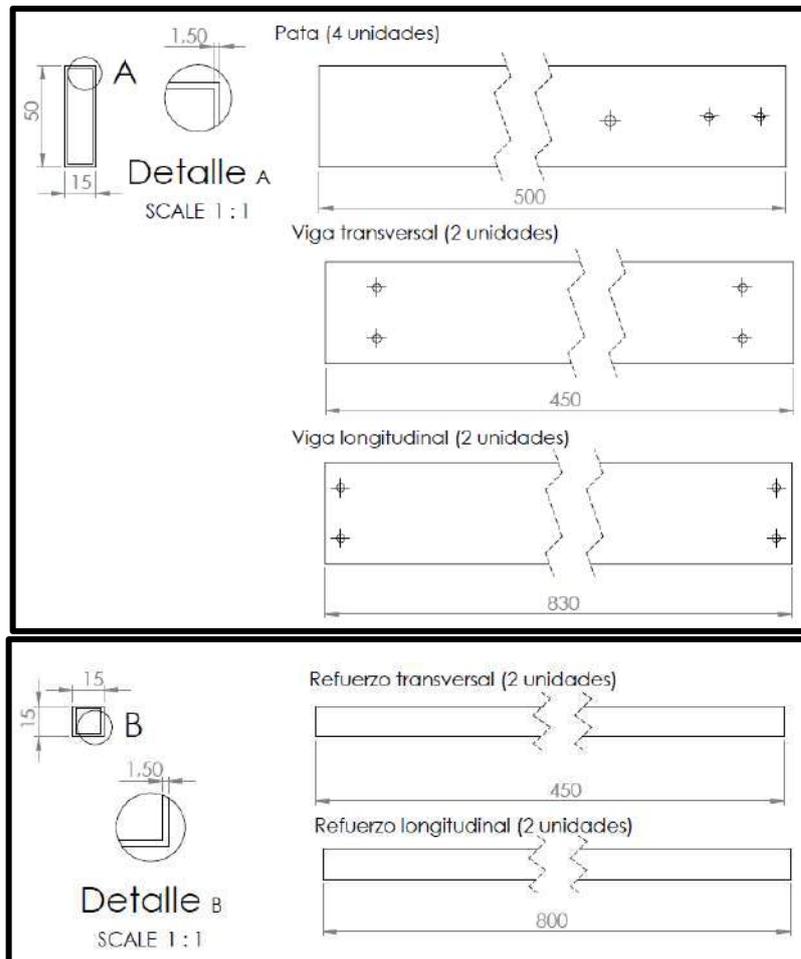


Figura 3.1 – Perfiles de aluminio con medidas de patas, vigas y refuerzos.

En el mercado local conseguimos estos materiales en tiras de seis metros de longitud y luego fueron cortadas a medida por medio de una sierra de descenso hidráulico la cual se encuentra en el taller de tecnología mecánica de facultad, dicha maquina se muestra en la figura 3.2.



Figura 3.2 - Sierra de descenso hidráulico.

Luego como se explicó en el capítulo dos, los encuentros de dichas piezas fueron realizados por medio de chapas plegadas las cuales se muestran en la figura 3.3, estas piezas fueron cortadas con la guillotina de la figura 3.4 y plegadas con la plegadora de la figura 3.5.

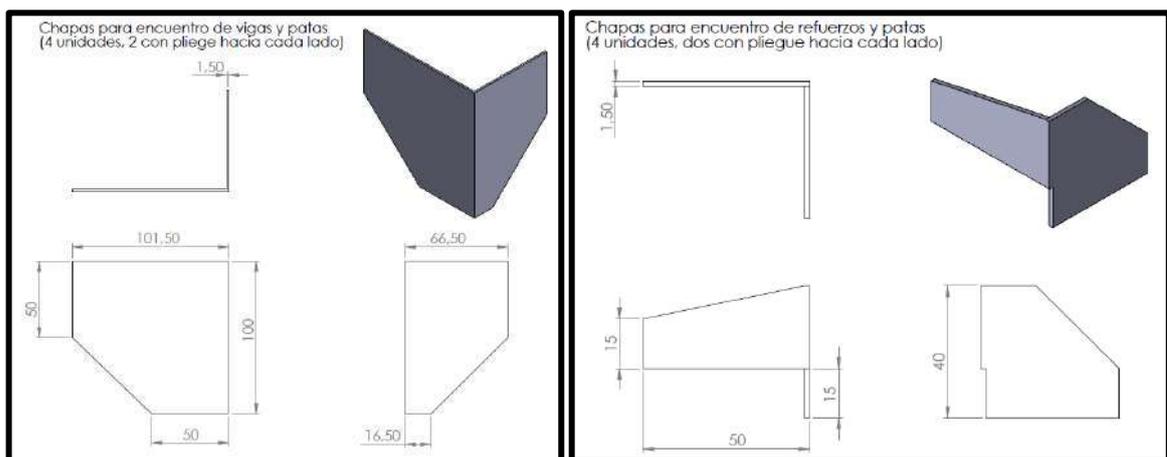


Figura 3.3 - Chapas plegadas para encuentros de vigas, patas y refuerzos-



Figura 3.4 - Guillotina para chapa.



Figura 3.5 - Plegadora manual.

Las piezas fueron perforadas con el taladro de banco de la figura 3.6 y atornilladas con tornillos auto-perforantes de 4mm de diámetro. Estos refuerzos han sido agregados una vez avanzada la construcción de la máquina, en la figura 3.7 se muestra la estructura ensamblada sin los refuerzos.



Figura 3.6 - Taladro de banco.



Figura 3.7 - Estructura principal ensamblada (sin los refuerzos).

Sobre la estructura principal van montadas las varillas por sobre las cuales desliza el puente móvil del cual se hablará más adelante. Para montar dichas varillas se emplearon piezas como la de la figura 3.8, las cuales están hechas de Polietileno de alta densidad o APM (alto peso molecular) como se lo conoce en la industria. Compramos una plancha rectangular del mencionado material, cuyas dimensiones eran 500x300x15mm.

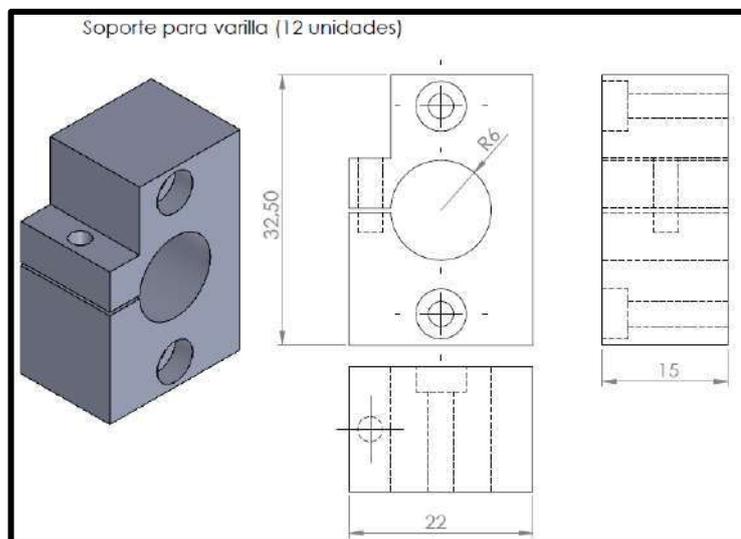


Figura 3.8 - Soporte para varillas.

Para poder cortarlas y llevarlas a la medida adecuada se empleó la fresadora de la figura 3.9. En dicha figura se muestra también la fresa utilizada. En la figura 3.10 se muestra el parte del proceso de mecanizado de las piezas y el resultado final del mismo.



Figura 3.9 – Fresadora marca Venier y herramienta de corte utilizada.



Figura 3.10 - Proceso de fresado y resultado en las piezas.

Luego las guías lineales fueron implementadas con varilla de 12 mm de acero SAE 1045 rectificadas. Fueron cortadas a medida por medio de una sierra sensible, como se muestra en la figura 3.11.



Figura 3.11 - Corte de varillas con sierra sensitiva.

Debido a que las varillas son de acero rectificado, es decir, no poseen ningún tratamiento anticorrosivo, fue necesario realizarles un niquelado para mejorar la resistencia a la corrosión a las mismas.

Luego las vigas transversales de la estructura principal fueron perforadas con un taladro de mano para poder montar dichos soportes.

3.3 – PUENTE MÓVIL

Como se explicó en el capítulo dos, el puente móvil consta de dos placas unidas entre sí mediante las varillas que funcionan como guías lineales para el movimiento de la torre. Para construir dichas placas optamos por utilizar el mismo material que en el caso de los soportes (polietileno de alta densidad o APM) pero en este caso en un espesor de 6 mm. Las medidas de dichas placas se muestran en la figura 3.12.

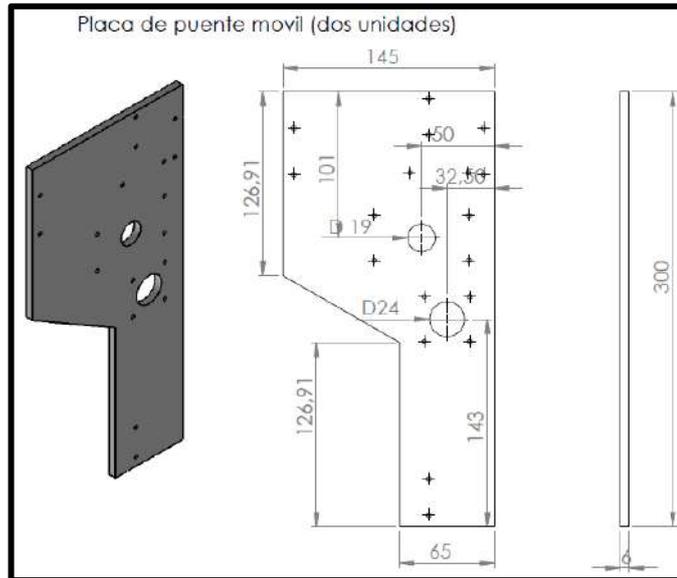


Figura 3.12 - Placa de puente móvil.

Luego sobre estas placas se montaron las piezas de la figura 3.13, las cuales, al igual que los soportes para las varillas de la estructura principal se hicieron de APM de 15 mm y se fresaron para darles las medidas adecuadas como se muestra en la misma figura. También sobre las placas se montaron los soportes para sujetar las varillas por donde deslizaría la torre.

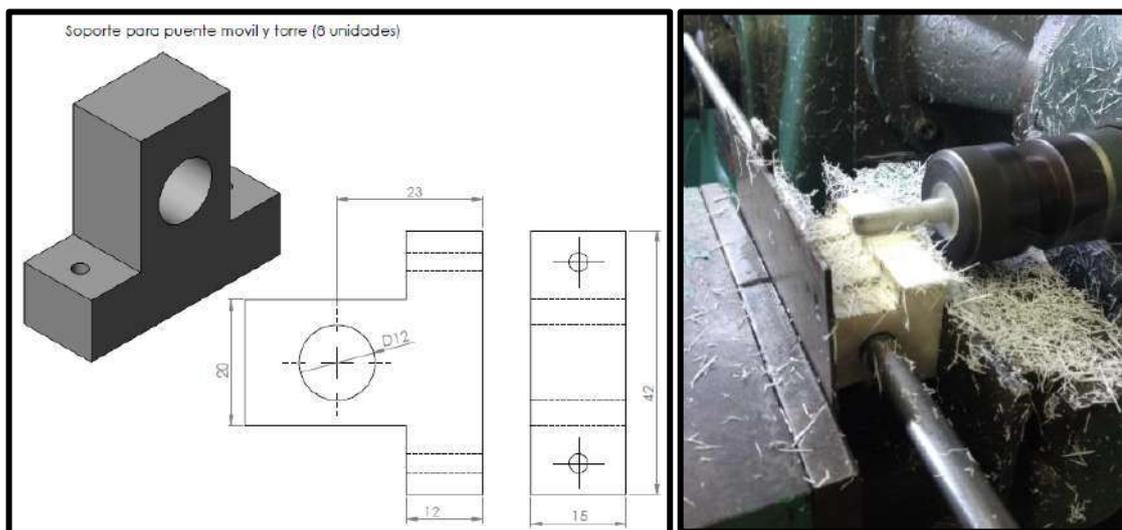


Figura 3.13 – Plano y proceso de mecanizado de los soportes deslizantes para el puente móvil.

Para ensamblar las piezas mencionadas utilizamos tornillos de 3mm de diámetro con rosca whitworth con sus respectivas tuercas y arandelas, en la figura 3.14 se muestra el proceso de armado.

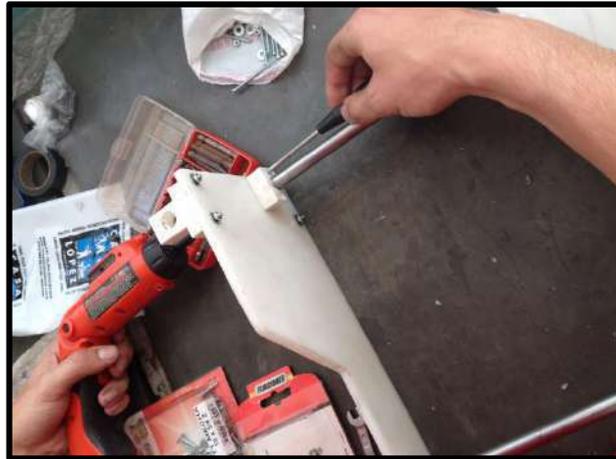


Figura 3.14 - Ensamblaje del puente móvil.

Luego se fabricaron los rodillos que van montados en los motores que generan el movimiento longitudinal del posicionador, las medidas de los mismos son las mostradas en la figura 3.15. Para construirlos compramos una varilla cilíndrica de bronce, la cual fue cortada y posteriormente torneada como se muestra en la figura 3.16. Además fue necesario taladrar un agujero para el eje del motor y también se hizo otro agujero para poder colocar un prisionero de ajuste para evitar que el rotor se salga del eje del motor. Para ello se hizo un agujero de 2,5 mm y luego se pasó un macho de rosca 3x0.5.

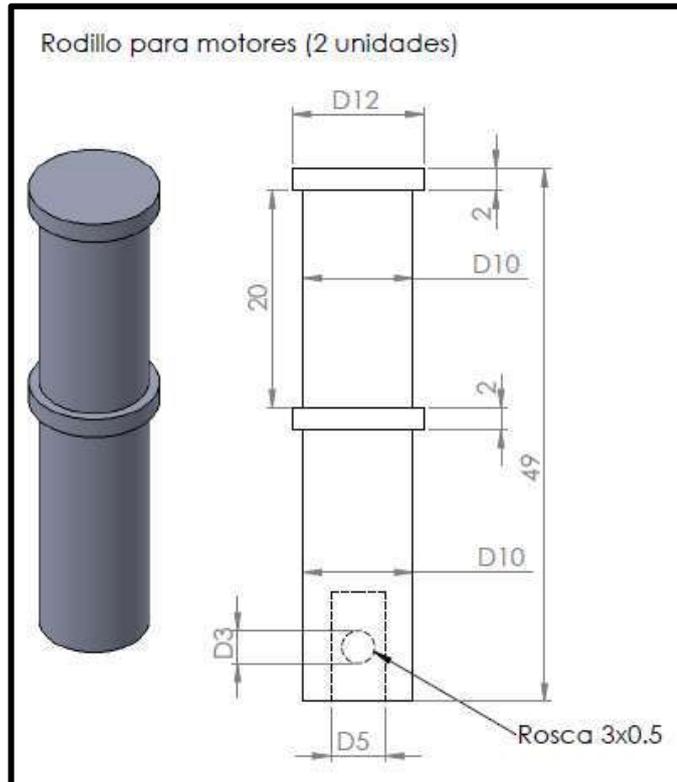


Figura 3.15 - Rodillos para movimiento longitudinal.



Figura 3.16 - Torno paralelo y proceso de mecanizado de rodillos.

Luego se construyeron los alojamientos para los rodamientos, los cuales se construyeron en APM de 15mm, después se hizo la placa en la cual se montaría el motor, la cual se implementó con APM de 6mm. Los separadores se hicieron del mismo perfil de aluminio que el utilizado para las patas y vigas de la estructura principal. Las medidas de dichas piezas se muestran en la figura 3.17.

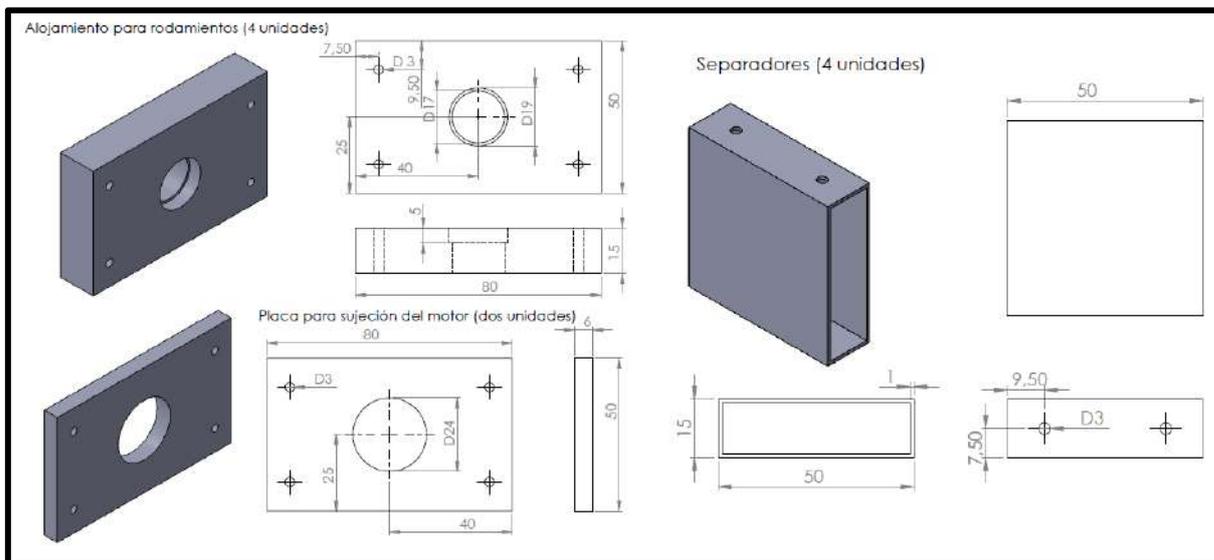


Figura 3.17 - Medidas de piezas necesarias para montaje de motor.

Para realizar el asiento del rodamiento en el alojamiento, debimos montar la pieza en el torno y mecanizarla con la herramienta de interiores que se muestra en la figura 3.18.



Figura 3.18 - Herramienta de interiores para torno paralelo.

El tornillo de movimiento fue implementado con una varilla roscada de 5/8” de diámetro y con un paso de 11 hilos por pulgada. Esta varilla fue cortada con la sierra sensitiva mostrada anteriormente y luego fue mecanizada con el torno paralelo para obtener las medidas que se indican en la figura 3.19.

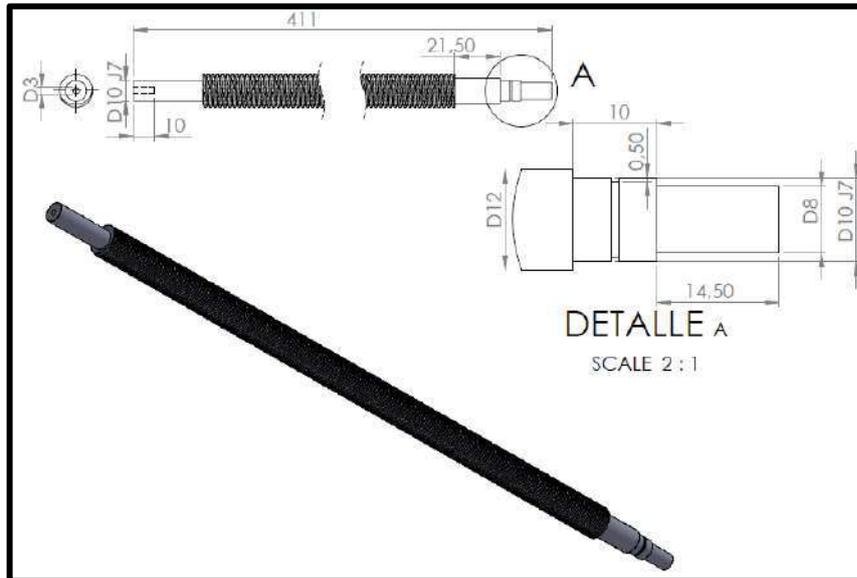


Figura 3.19 - Plano de tornillo para el puente móvil.

Debido a que la tolerancia necesaria para poder realizar un montaje correcto del rodamiento era muy fina, fue necesario mecanizar el tornillo en principio con una herramienta de corte y luego se hizo la terminación empleando tela de lija. En la figura 3.20 se muestra el resultado del proceso mencionado.

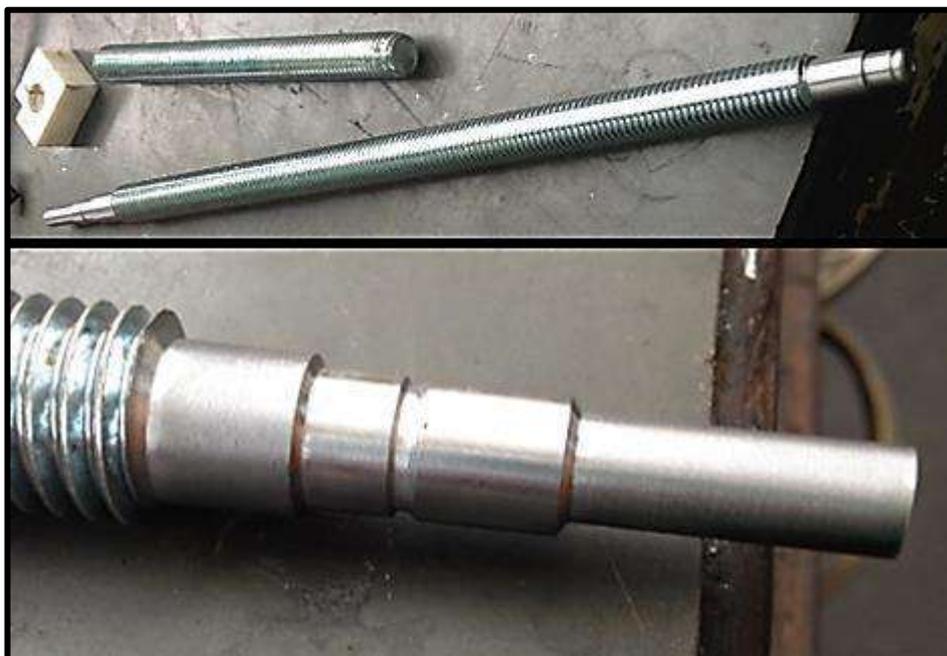


Figura 3.20 - Tornillo del puente móvil terminado.

Para continuar con el montaje del tornillo fue necesario avanzar con la construcción de la torre, por lo tanto continuamos explicando la construcción de la misma.

3.4 – TORRE VERTICAL

La torre vertical consta de una “espalda” la cual fue construida con APM de 6mm, las medidas de dicha pieza se muestran en la figura 3.21, en dicha espalda se montaron los soportes deslizantes para las guías lineales los cuales son iguales a los utilizados anteriormente en el puente móvil.

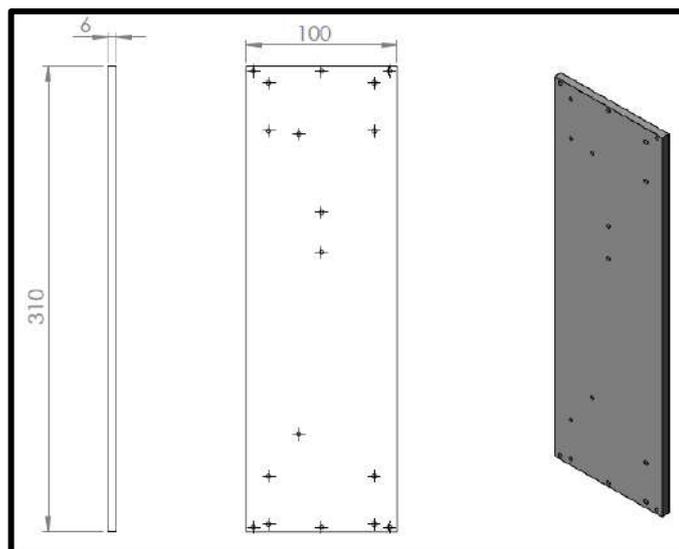


Figura 3.21 - Medidas generales de la "espalda".

En la figura 3.22 se puede apreciar la mencionada espalda y los soportes montados ensamblados en el puente móvil y también el conjunto montado en la estructura principal.



Figura 3.22 - Pre-ensamble de la torre en el puente móvil y en la estructura principal.

Para realizar el movimiento del conjunto es necesario anclar el sistema a una tuerca, la cual fue implementada con una pieza de APM de 15 mm, cuyo diseño y medidas son las de la figura 3.23, como se mencionó en el capítulo dos, hemos optado por utilizar solo media tuerca de manera de admitir un “despegue” del tornillo con respecto a la tuerca y así no tener mayores inconvenientes con la desalineación del mismo. Luego el montaje del tornillo se realizó como se explicó en el capítulo dos y el acople elástico fue implementado con un trozo de manguera de gas para alta presión, mediante la cual se acoplaron los ejes del motor y del tornillo. En la figura 3.24 se puede apreciar una manguera similar.

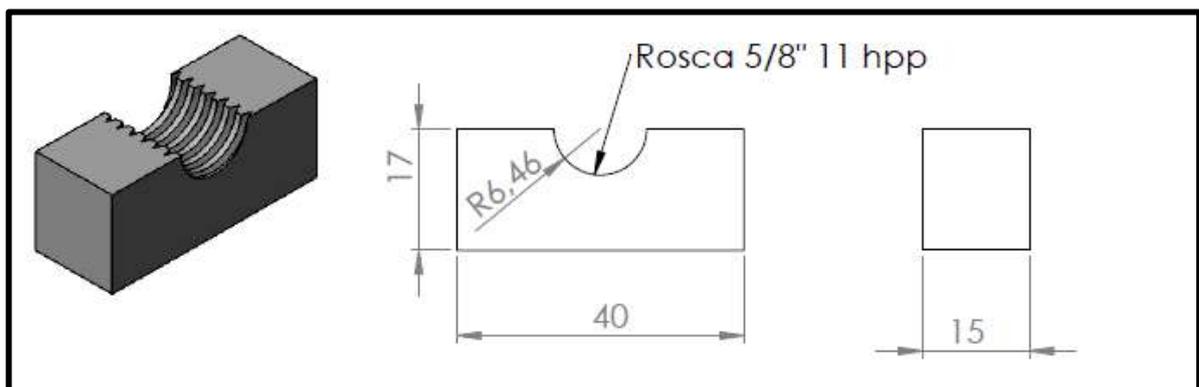


Figura 3.23 - Tuerca de movimiento transversal.



Figura 3.24 - Manguera para alta presión, acople elástico.

Una vez que se finalizó con el mecanizado de la espalda, se hicieron las placas superior e inferior de la torre, para ello se utilizó APM de 6 mm de espesor. También, en el mismo material, se hicieron las placas del movimiento vertical. El plano de las piezas presentadas en este párrafo se muestra en la figura 3.25.

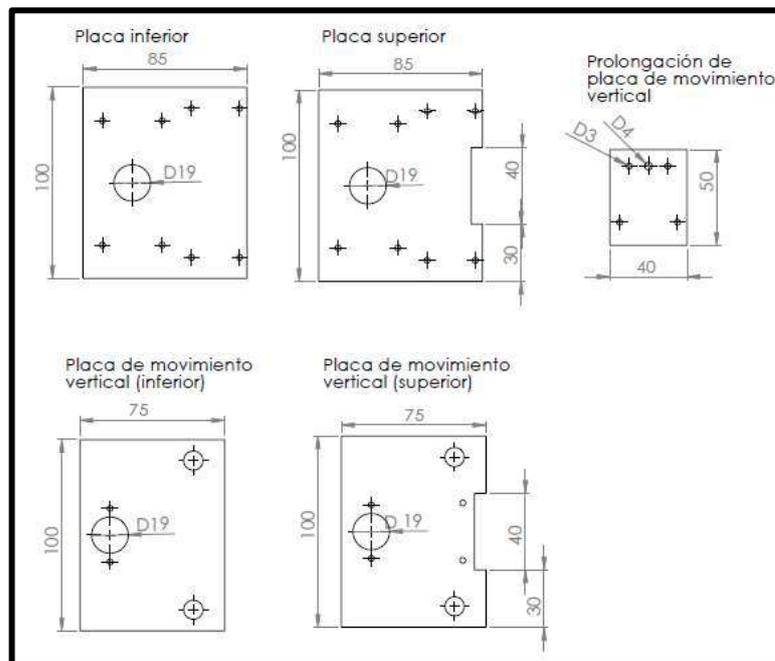


Figura 3.25 - Medidas de placas inferior y superior, y placas de movimiento vertical.

Luego se montaron los alojamientos para los rodamientos, el tornillo sin fin, los separadores y la placa sujetadora del motor de la misma manera que como se hizo para el puente móvil. Cabe destacar que en este caso el tornillo de movimiento debía ser de 367 mm a diferencia del tornillo del puente móvil el cual era de 411 mm. Finalmente el mecanismo mencionado quedó ensamblado como se muestra en la figura 3.26.

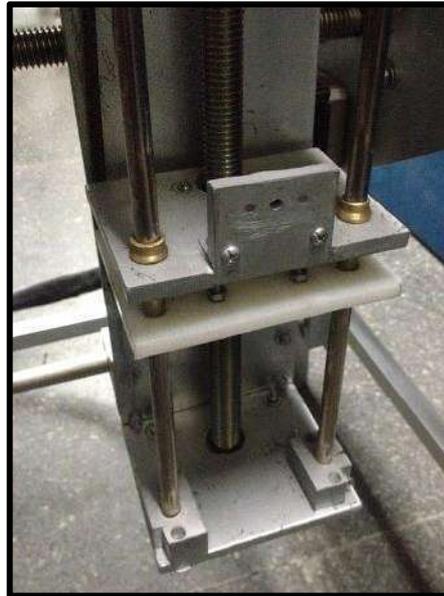


Figura 3.26 - Torre vertical y puente móvil ensamblados en la estructura principal.

En APM de 15 mm se mecanizó el soporte para el Erlenmeyer que se utilizaría para hacer depósito por spray cuyas medidas debían ser como las mostradas en la figura 3.27, el mismo fue mecanizado en fresadora y perforado con el taladro de banco.

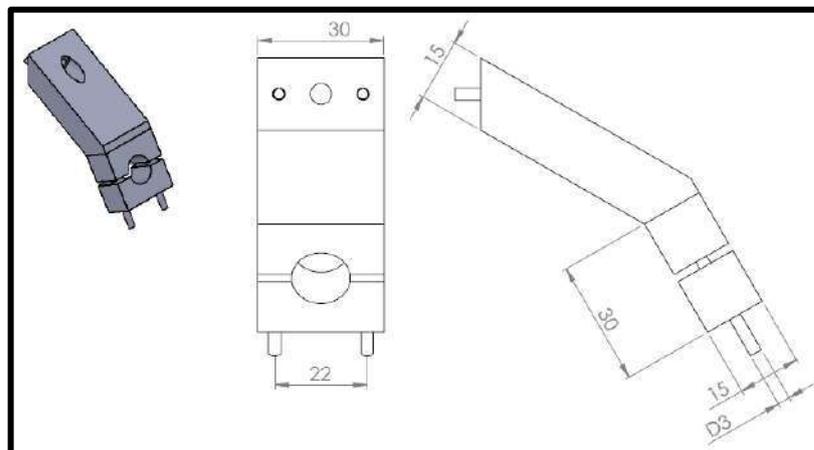


Figura 3.27 - Plano de instrumento para depósito por spray.

Para sujetar esta pieza fue necesario construir una pieza plástica (en el capítulo de diseño se habló de esta pieza como una “prolongación” de la placa de movimiento vertical), la cual fue amarrada a la pieza móvil de la torre mediante una chapa en forma de L. En la figura 3.28 se muestra el ensamblaje de las placas de movimiento vertical junto con la chapa L y la prolongación mencionada.

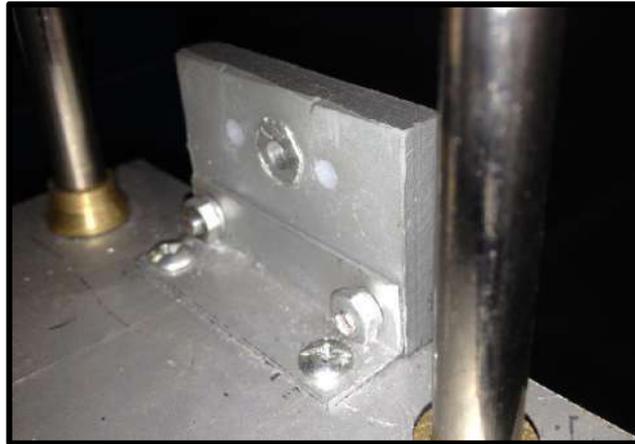


Figura 3.28 – Piezas para sujetar el soporte del Erlenmeyer.

Luego se fabricaron las piezas de la figura 3.29, las cuales se usaron como soporte para el gancho metálico que se utilizaría para hacer depósito por dipping. Finalmente la pieza terminada quedó como se muestra en la figura 3.30.

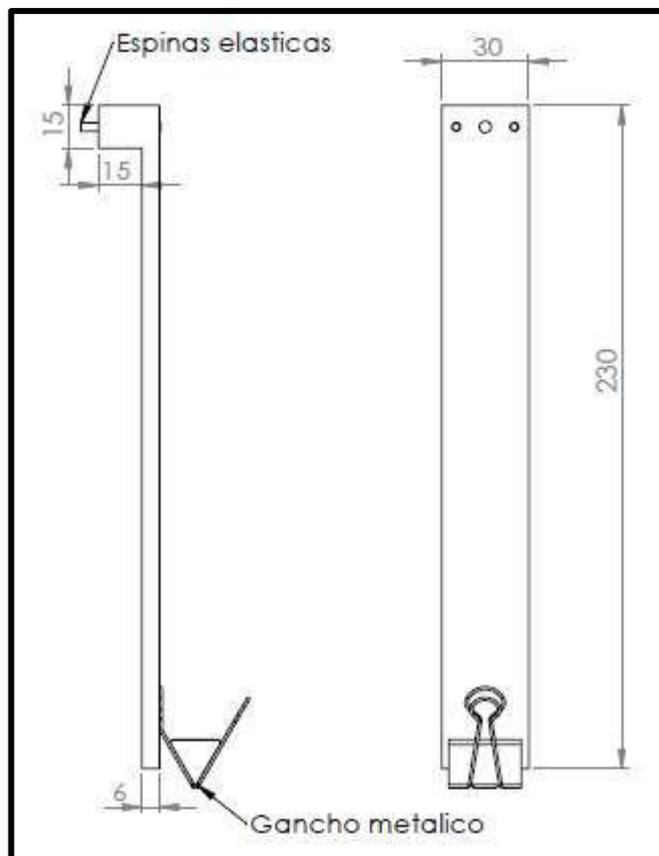


Figura 3.29 - Plano de instrumento para depósito por dipping.



Figura 3.30 - Instrumento para depósito por dipping terminado.

Para rigidizar el puente móvil se optó por colocar una pieza de perfil estructural de aluminio (ver figura 3.31), al cual se le colocaron tapones en los extremos de APM de 15 mm, sujetados con espigas elásticas. Estos tapones permitieron atornillar los extremos del refuerzo a las placas del puente móvil.



Figura 3.31 - Refuerzo del puente móvil ensamblado.

Se continuó desarmando todo el ensamblaje, limpiando pieza por pieza y pintándolas con pintura en aerosol color aluminio como se muestra en la figura 3.32.



Figura 3.32 - Mecanismo desarmado y pintado.

3.5 – CABLEADO

Por último se realizó el cableado de los motores y los sensores, también se hizo el montaje de la manguera que va desde la electroválvula hasta la tobera del Erlenmeyer.

Para sujetar los cables y la manguera a la estructura de la máquina y a las partes móviles se utilizaron las grampas que se muestran en la figura 3.33. En la Figura 3.34 se muestra el ensamblaje final de la máquina con la disposición de los cables y de la manguera.

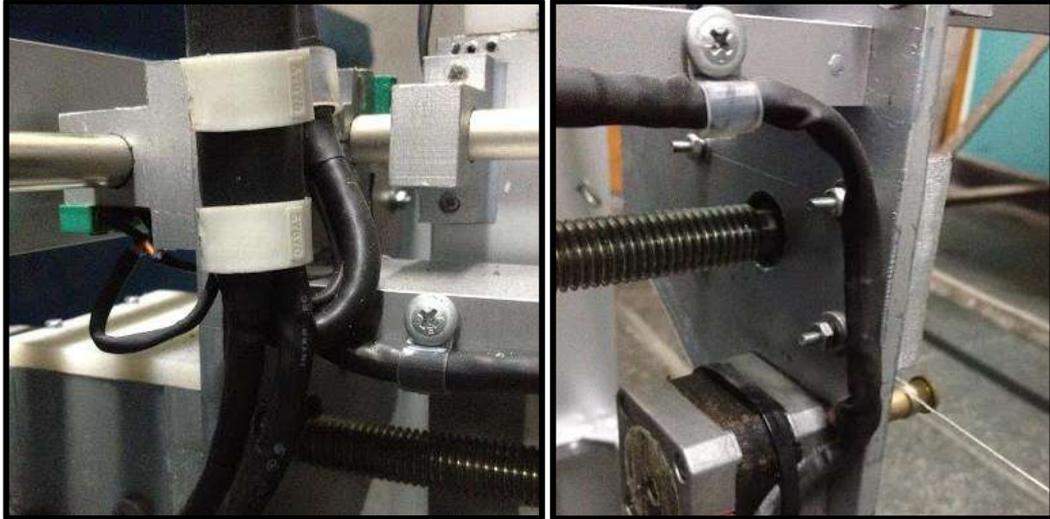


Figura 3.33 - Grampas para sujeción de cables.

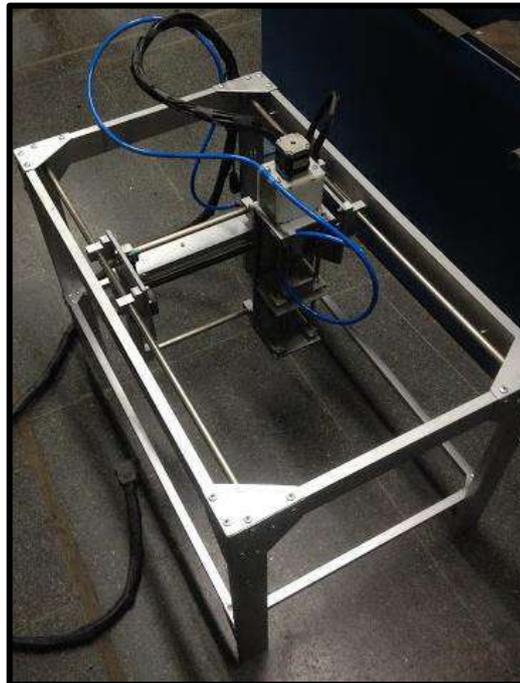


Figura 3.34 - Ensamblaje final.

Para facilitar el conexionado de la maquina se optó por utilizar conectores DB25 (tradicionalmente utilizado como puerto paralelo en impresoras antiguas), la maquina tiene sobre una de sus patas, uno de estos conectores atornillados como se muestra en la figura 3.35, luego mediante un alargue, la maquina se conecta al gabinete (ver figura 3.36) donde se encuentra la fuente de 24V y el controlador. En el mismo gabinete van conectados el sensor de presión diferencial y el controlador de la placa calefactora, el cual se realizó mediante conectores de micrófono (ver figura 3.36).



Figura 3.35 - Conector DB25.



Figura 3.36 - Gabinete controlador con conectores.

CAPITULO IV HARDWARE PARA MONITOREO Y CONTROL

El presente capítulo pretende explicar el principio de control de la máquina mediante la creación de un sistema de supervisión, control y adquisición de datos (SCADA). En este capítulo se explicarán las partes que lo integran, y luego la manera en que está compuesto el hardware de cada una de ellas y de qué manera lo hemos resuelto en nuestro proyecto.

4.1 – INTRODUCCIÓN AL SISTEMA SCADA

La expresión SCADA, tal como mencionamos anteriormente, es el acrónimo de “Supervisory Control and Data Acquisition”.

Se trata de un sistema capaz de obtener y procesar información de procesos industriales dispersos y de actuar en forma remota sobre los mismos. Esto significa que permite supervisar simultáneamente procesos e instalaciones industriales distribuidas en grandes áreas, tales como las redes de distribución eléctrica, oleoductos, gasoductos, etc.

El SCADA describe un número de unidades terminales remotas (RTU’s, Remote Terminal Units) instaladas en las cercanías del proceso, las cuales se comunican con una estación maestra (MTU, Master Terminal Station) ubicada en una sala de control central, a la cual el usuario u operador tiene acceso a la información del sistema mediante una interfaz hombre máquina (HMI).

Una RTU es un sistema que cuenta con un microprocesador e interfaces de entrada y salida tanto analógicas como digitales que permiten tomar la información del proceso, provista por los dispositivos de instrumentación y control en una localidad remota y, utilizando técnicas de transmisión de datos, enviarla al sistema centralizado maestro. La MTU, bajo un software de control, permite la adquisición de la data a través de todas las RTU’s ubicadas remotamente y brinda la capacidad de ejecutar comandos de control remoto cuando es requerido por el operador. Los datos adquiridos por el MTU se presentan a través de una interfaz gráfica en forma comprensible y utilizable (HMI) para ser monitoreada por el usuario.

En nuestro proyecto, el sistema posee un MTU, que será la PC por la cual el usuario enviará los parámetros deseados; y un único RTU que se encargará de realizar la

supervisión de los sensores sobre la máquina y campana de seguridad y de controlar el movimiento de los motores, electroválvula, etc.

Las funciones básicas de un sistema SCADA son las que se describen a continuación:

- Supervisión remota de instalaciones
- Control remoto de instalaciones
- Procesamiento de información
- Presentación de gráficos dinámicos
- Generación de reportes
- Presentación de alarmas
- Almacenamiento de información histórica
- Presentación de gráficos y tendencias
- Programación de eventos

De las cuales destacamos entonces tres tareas importantes a realizar por el sistema:

- Recolección periódica, procesamiento y monitoreo de información del sistema a controlar.
- Control remoto de dispositivos y reemplazo de valores en la base de datos del sistema.
- Presentación de despliegues y alarmas a los operadores del sistema.

Y los elementos básicos que lo componen y que se han mencionado anteriormente son:

- Interfaz Humano Máquina (HMI)
- Unidad Maestra (MTU)
- Canales o medios de comunicación
- Unidades Remotas (RTU)
- Sensores o transductores y actuadores

En nuestro proyecto tanto la MTU y la HMI funcionan en la PC del usuario.

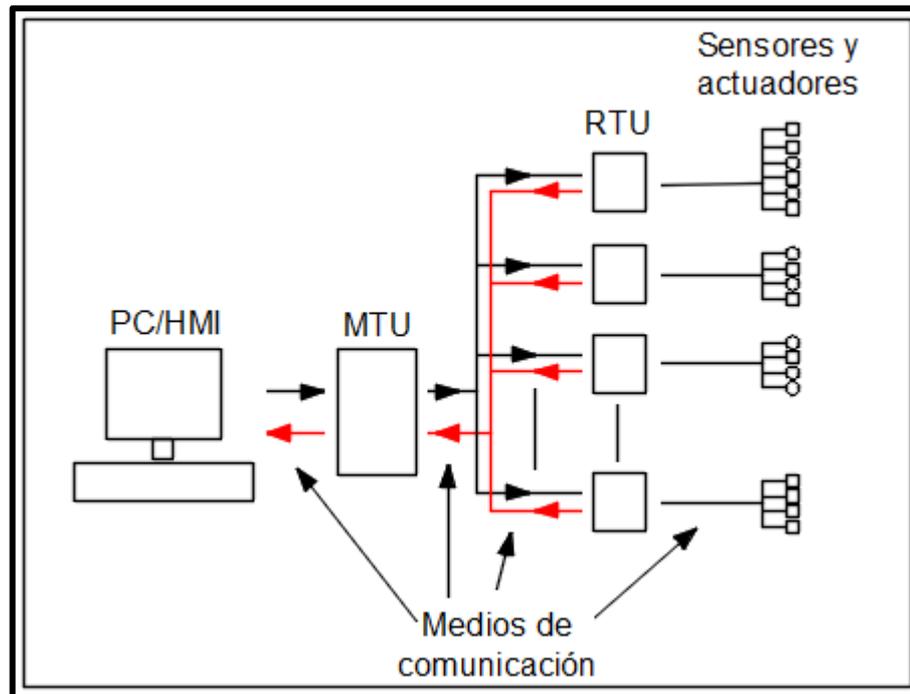


Figura 4. 1 - Esquema de elementos y conexionado de un sistema SCADA

En ésta sección se irá describiendo como hemos realizado cada una de éstas partes dentro de nuestro proyecto, comenzando por los sensores y actuadores en su recorrido hacia el HMI.

4.2 – SENSORES Y ACTUADORES

4.2.1 – Sensores

Los sensores son dispositivos capaces de detectar magnitudes físicas o químicas, y transformarlas en señales eléctricas. Estas señales brindan información sobre el estado del proceso o máquina y permiten realizar una retroalimentación.

Son ampliamente utilizados por ejemplo, para detectar posiciones críticas, coordinar etapas de procesos, detectar estados de riesgo en ambientes, sensor temperatura, voltajes, corrientes, luminiscencia, velocidad, etc. En base a los datos recibidos, se pueden identificar variables del proceso y actuar en función a ello como corresponda.

En nuestro proyecto hemos utilizado una serie de sensores, que pueden clasificarse según sean para proteger al usuario y el entorno de la máquina, y para proteger a la máquina en sí, de posibles daños.

4.2.1.1 – Sensor diferencial de presión

La sustancia que se emplea para realizar el recubrimiento superficial mediante spray puede ser dañina para cualquier ser humano que se encuentre expuesto a la misma y por lo tanto se debe evitar cualquier concentración elevada de ésta en el espacio físico donde se encuentre la máquina.

El proceso debe garantizar mínimamente el flujo de solventes, suspensiones y otros volátiles de modo que no alcancen al operador por razones de seguridad. Es por ello que la máquina será instalada dentro de una campana de seguridad (construida previo al comienzo de este proyecto), y el proceso se debe realizar con la misma en un nivel de hermeticidad.

Lo ideal para detectar este tipo de riesgos es utilizar sensores capaces de detectar una concentración de sustancias peligrosa para el ser humano, pero dado que ésta puede variar y que por lo tanto se debería utilizar un sensor diferente para cada sustancia peligrosa (diferentes concentraciones, métodos de detección, etc.), hemos optado entonces, por controlar la presión dentro de la campana, asegurándose una depresión con respecto a la atmosférica donde se encuentra el usuario. De este modo nos aseguramos que el extractor de aire de la campana está en funcionamiento y los vapores están siendo extraídos al exterior de manera correcta. En caso de que no haya depresión (campana no suficientemente hermética, extractor apagado, etc.), el MTU no ordenará el comienzo del proceso o detendrá el mismo, enviando una alarma escrita al usuario.

Para nuestro proyecto hemos utilizado un sensor diferencial de presión que funciona como un switch o llave eléctrica (Ver Anexo E) cuando se detecta un nivel superior al calibrado. Hemos obtenido uno cuyo rango de regulación varía entre 0,2 [mbar] y 2 [mbar], el cual es mostrado en la figura 4.2.



Figura 4. 2 - Sensor diferencial de presión.

Para determinar si el mismo sería suficiente o los valores necesarios según nuestro proceso entrarían en dicho rango, hemos experimentado en la campana, asegurando una hermeticidad elevada y ubicando al sensor en la zona más desfavorable para probar su correcto funcionamiento. En cuanto se daban las condiciones de hermeticidad (campana cerrada), el sensor diferencial de presión era capaz de detectar la depresión, mientras que al abrir la misma en un pequeño área, la presión dentro de la campana ya se aproximaba a la de ambiente en un valor menor al de la sensibilidad mínima del sensor. Por lo tanto concluimos en que el mismo es adecuado para nuestra aplicación ya que sería capaz de detectar si el extractor se encuentra encendido y los vapores o gases son liberados al exterior del edificio, evitando así que entre en contacto con cualquier ser humano.

4.2.1.2 – Finales de carrera

Éste tipo de sensor es utilizado para proteger a la máquina y a los motores de cualquier colisión que dañe al equipo. Por un lado el movimiento y choque puede dañar sus partes, y por otro lado, los motores pueden sufrir desperfectos si se los sobrecarga en el caso de quedar bloqueada alguna parte móvil. Además, como hemos mencionado, nos permiten colocar un punto de referencia para nuestra máquina.

Se han seleccionado para ésta aplicación los micro-pulsadores comerciales que se muestran en la figura 4.3:

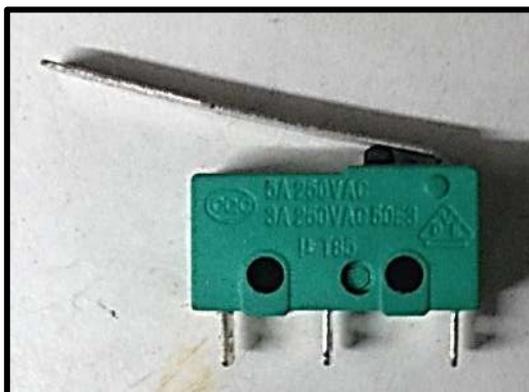


Figura 4. 3 - Micropulsador utilizado como final de carrera

Para cada eje se requieren dos de estos dispositivos para indicar la posición extrema en cada uno de sus sentidos. Ésta posición extrema es determinada de manera tal que ninguna de las partes en movimiento (incluyendo el Erlenmeyer) pueda colisionar.

Al accionarse uno de estos pulsadores, se cierra un circuito conectado al RTU, el cuál reconocerá que se ha alcanzado la posición crítica y en ese instante se detienen los motores. Una vez alcanzada una de estas posiciones, se dará aviso al usuario y este podrá identificar la causa de la falla y actuar según sea necesario.

4.2.2 – Actuadores

Los actuadores son los elementos que integran el sistema o proceso, encargados de transformar la energía eléctrica, hidráulica, neumática, etc., para generar un efecto sobre un proceso automatizado.

En nuestro proceso utilizamos diversos tipos de actuadores, como lo son los motores capaces de generar el desplazamiento de las partes, electroválvula para permitir el paso de fluido gaseoso cuando es necesario, y por otro lado la placa calefactora para calentar las probetas a recubrir.

Se explicará a continuación brevemente cada uno de ellos y bajo qué criterios han sido seleccionados.

4.2.2.1 – Motores Paso a Paso (PAP)

Para este proyecto, uno de los cálculos más difíciles de resolver, es el de la potencia necesaria de los motores para poder realizar los movimientos de la máquina. Nuestro sistema no realiza grandes esfuerzos dado que no hay fuerzas opositoras importantes más allá del peso de los componentes y el rozamiento, siendo este último el de mayor influencia y de muy difícil estimación.

Por otro lado, nuestra máquina debe ser capaz de localizarse en cualquiera de los tres ejes, de modo tal de poder generar un movimiento asistido por computadora. Para esto existen diferentes opciones en el mercado, entre las cuales se enumeran las más viables:

- **Motorreductor de corriente continua:** Consiste en un motor de corriente continua, el cual viene con un reductor incorporado que sirve de multiplicador de fuerza. Estos tienen la ventaja de tener un gran torque, aunque para ello el número de vueltas no puede ser tan elevado. Por otro lado, el sistema de reducción es muy ruidoso, y además se requiere de un encoder u otro sistema de posicionamiento externo para realizar una realimentación.
- **Motor Paso a Paso (PAP):** Son otro tipo de motores de corriente continua, los cuales permiten girar su rotor en pequeños ángulos controlados por pulsos. Este tipo de funcionamiento ha sido ampliado en el anexo F, y lo hacen muy simple a la hora de controlar, teniendo la ventaja de posicionar o enviar el número de pulsos necesarios para alcanzar una posición. Es por ello que los hemos seleccionado como la mejor opción para utilizar en nuestro proyecto.

Inicialmente disponíamos de tres motores que fueron extraídos de una máquina de revelar fotografías que se encontraba obsoleta, dos VEXTA PX244-11A-C19 como el que se muestra en la figura 4.4 y un VEXTA PH268-E2.3B-C1 mostrado en la figura 4.5 el cual es descartado de nuestra aplicación por ser el valor de corriente nominal excesivamente grande para el driver seleccionado cuya elección se discutirá más adelante en este informe. En ambas imágenes se pueden apreciar los valores de tensión y de corrientes nominales.

Procedimos buscando más información sobre los mismos y desafortunadamente encontramos que ya no son fabricados y es por esto no hay información disponible. Con los valores de tensión y corriente especificados en las carcasas hemos podido buscar en el mercado actual motores con aproximadamente las mismas características y así poder tener una información adicional.

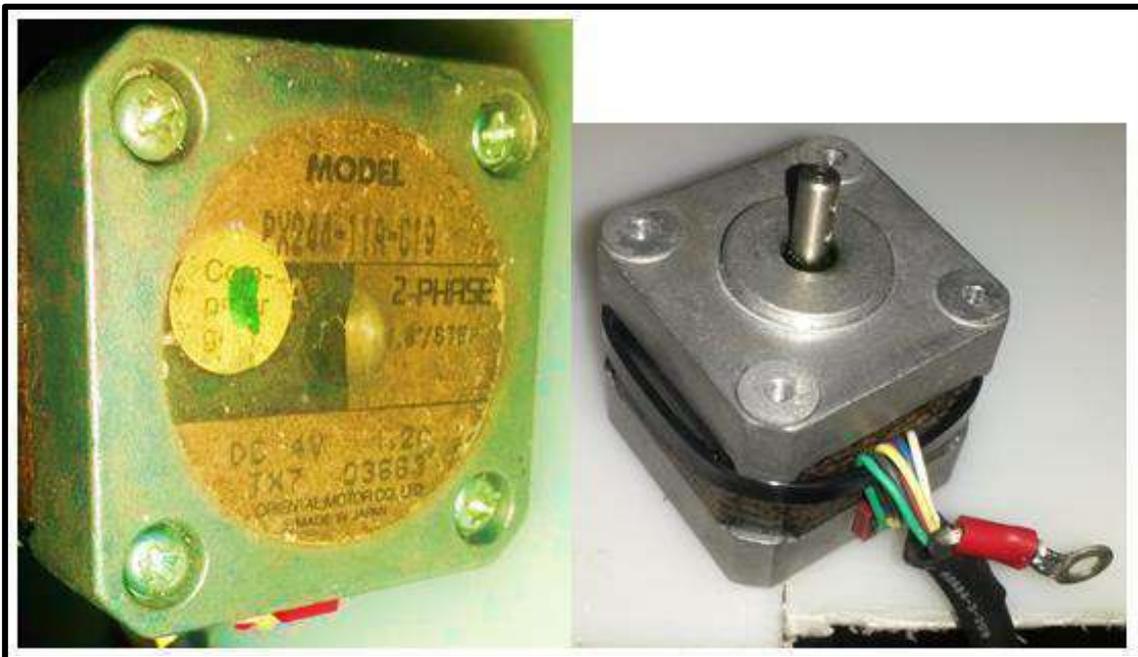


Figura 4. 4 - Motor VEXTA PX244-11A-C19



Figura 4. 5 - Motor PAP Vexta PH268-E2.3B-C1

En la figura 4.6 se muestra un motor que posee las mismas dimensiones y aproximadamente los mismos valores nominales de tensión y corriente.



Figura 4. 6 - Motor de características similares

De esta manera sabemos que el holding torque de estos motores es aproximadamente 31 [oz.in].

Luego, una vez finalizado el montaje de la maquina se realizaron pruebas con los motores pequeños montándolos en el eje X de nuestra máquina y se determinó que funcionaban correctamente para nuestra aplicación ya que no se presentaban pérdidas de paso aún a grandes velocidades.

Por otro lado necesitábamos dos motores más para colocar en los ejes Y, y Z, en donde previamente habíamos probado los motores pequeños y estos perdían pasos. Esto era de esperarse ya que como se vio en el capítulo referido a diseño, en los ejes Y, y Z el movimiento se realiza a través de tornillos de fijación, por lo tanto las fuerzas de rozamiento adquieren cierto protagonismo. Además, dichos tornillos no son perfectamente rectos, y como se puede apreciar en la figura 4.7-a el conjunto tornillo-guías lineales (resaltados en color azul) es un sistema rígido, el cual no admite una deflexión del tornillo, esto provoca que cualquier desalineamiento del mismo deba ser absorbido por las demás piezas mediante deformación y por lo tanto demanda un par motor mayor, lo mismo se muestra en la figura 4.7b en donde se aprecia el mismo efecto en el eje Z.

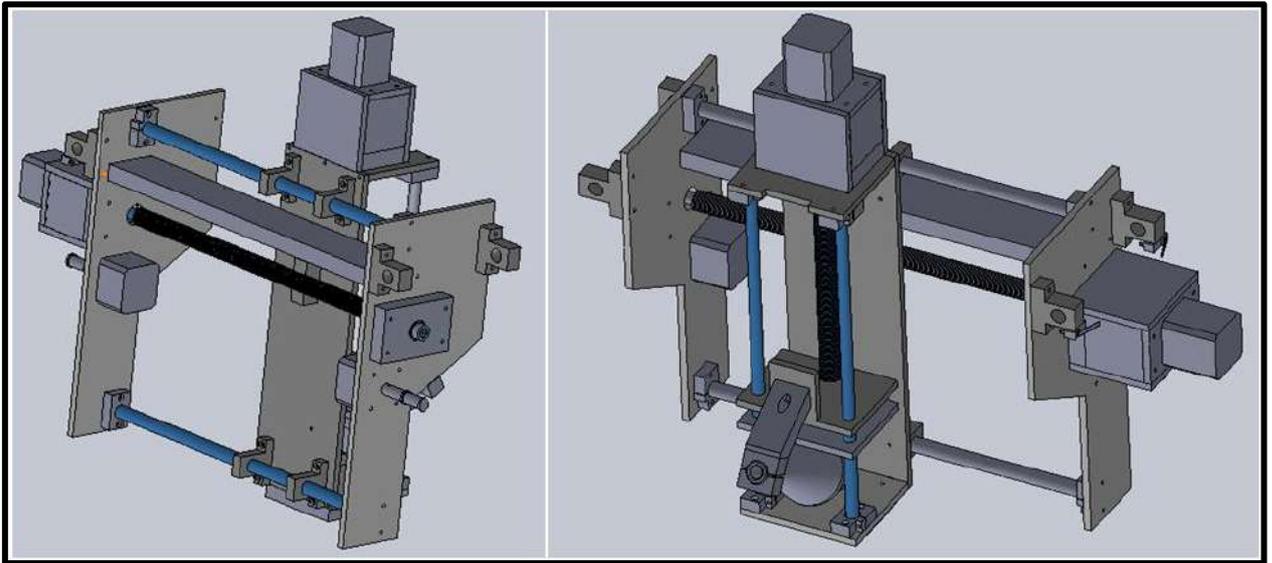


Figura 4. 7 (a y b, izquierda y derecha respectivamente) - Sistema ejes-tornillo de movimiento

Para determinar un motor que nos sea útil para nuestra aplicación, realizamos una investigación en las páginas de los distintos fabricantes de máquinas CNC e impresoras 3D, y buscamos cuales podían ser los motores más apropiados para nuestra aplicación. Encontramos que los más recomendados son los que se muestran en la figura 4.8



Figura 4. 8 - Motor PAP y especificaciones de los modelos recomendados por otros fabricantes

Estos motores poseen el doble de torque que los utilizados para el eje X por lo tanto consideramos que iban a ser adecuados.

Luego se compraron los motores y realizamos el cableado pertinente. Finalmente las pruebas con los mismos resultaron satisfactorias y por lo tanto fueron montados finalmente en la máquina.

4.2.2.2 – Electroválvula

Una electroválvula es una válvula electromecánica, diseñada para controlar el paso de un fluido por un conducto. Es decir, que consiste en una válvula que cambia de estado en función del accionamiento de un solenoide. Este tipo de válvulas generalmente se utilizan para alternar entre un estado de cierre y apertura (todo o nada). Se han diseñado múltiples electroválvulas dependiendo del número de vías o canales por los cuales debe pasar el fluido, si en estado de reposo deben permitir el paso del fluido o interrumpirlo, modo de accionamiento, etc.

En nuestra aplicación debemos utilizar una electroválvula que permita el ingreso de fluido por un orificio, presente comportamiento normalmente cerrado y que al accionar el solenoide a través de nuestro controlador, la misma permita el paso del fluido desde el ingreso a la máquina hacia el Erlenmeyer. Es decir, que requerimos de la utilización de una electroválvula de dos vías y dos posiciones (2/2). Para mayor información con respecto a válvulas y modos de accionamientos ver el anexo D.

Electroválvula seleccionada:

Para la selección de la misma hemos hecho foco en los requerimientos o exigencias a las cuales se encontraría sometida en funcionamiento normal, es decir:

- Tipo de válvula: 2 vías
- Presiones de trabajo: 1,2 [atm] a 1,5 [atm] (absolutas)
- Fluido a controlar: CO₂, aire comprimido
- Tensión de solenoide: 24 [V]

Basándonos en estas características hemos buscado modelos de electroválvulas en catálogos Web de diferentes fabricantes (Festo, Jefferson, etc.) y en comercios de Mar del Plata para facilitar la compra y evitar la logística de entrega. Luego de un análisis técnico-económico de la oferta encontrada, hemos optado por comprar una válvula genérica, de cinco vías en lugar de las dos que nosotros requeríamos (las cuales no se encontraron en el mercado de nuestra localidad), pero cuyas entradas y salidas pueden ser utilizadas como una de dos vías, dejando libres los otros orificios. De este modo se cumple con la función requerida.

Como hemos comprado otra, el esquema de circuito interno de la misma es el de la figura 4.9 que se muestra a continuación:

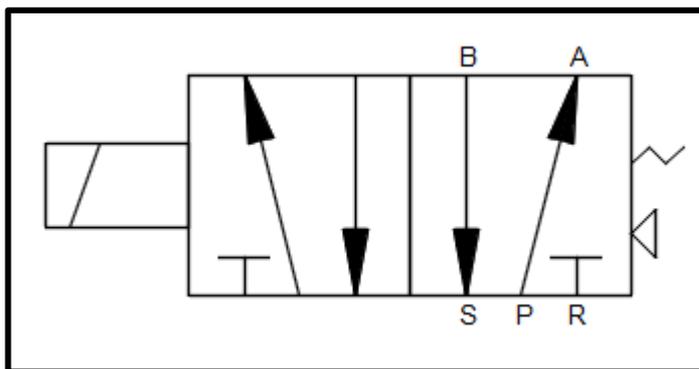


Figura 4. 9 - Esquema de electroválvula de cinco vías y dos posiciones

En la Figura se pueden observar las 5 vías. En ésta válvula, el fluido a presión ingresa por el orificio “P” y la salida por la “B”. De este modo se logra que al accionar la electroválvula se pase de la posición de reposo (derecha), al de la izquierda, donde se le permitirá al fluido salir por el orificio B.

Para evitar que en la posición de reposo el fluido salga por A mientras que está en reposo, hemos cerrado la misma con un tapón. Las vías S y R son de escape, que en nuestro caso no serán utilizados y por lo tanto se le colocarán silenciadores o filtros para evitar que ingrese suciedad dentro de la misma.

Para conectar los conductos a la electroválvula fue necesario utilizar acoples, en nuestro caso utilizamos un simple acople a presión para el conducto de entrada de fluido, debido a que el mismo conducto es el que proviene del tanque acumulador y ya dispone de una manguera sin acople rápido. Para la salida, hemos optado por un acople rápido, que permite un acople/desacople rápido y sin necesidad de realizar tanta fuerza al conectar el conducto. Se ha decidido el acople rápido del lado de la máquina para poder conectar fácilmente el conducto de gas a la tobera, la cual es frágil y en su manipulación se corre riesgo de rotura.

En la figura 4.10 se muestra el montaje y la conexión de nuestra electroválvula sobre una de las patas de la estructura:

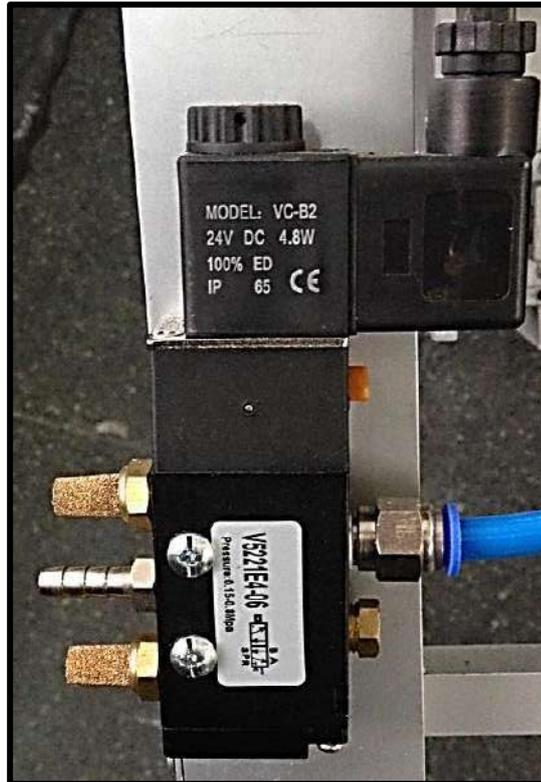


Figura 4. 10 – Electroválvula montada en la máquina

La entrada de aire entonces se realiza por la boca central de la izquierda del cuerpo de la válvula. Los extremos poseen los silenciadores. Del otro lado se observa el acople con el conducto propio de la máquina, y debajo de este el tapón que bloquea la salida de aire por el orificio A. Por encima del cuerpo de la válvula se encuentra el vástago, en este caso accionado por un solenoide cuyos contactos se encuentran en el bloque de la derecha, los cuales han sido cableados hasta el RTU.

El esquema del circuito neumático queda compuesto entonces según la figura 4.11.

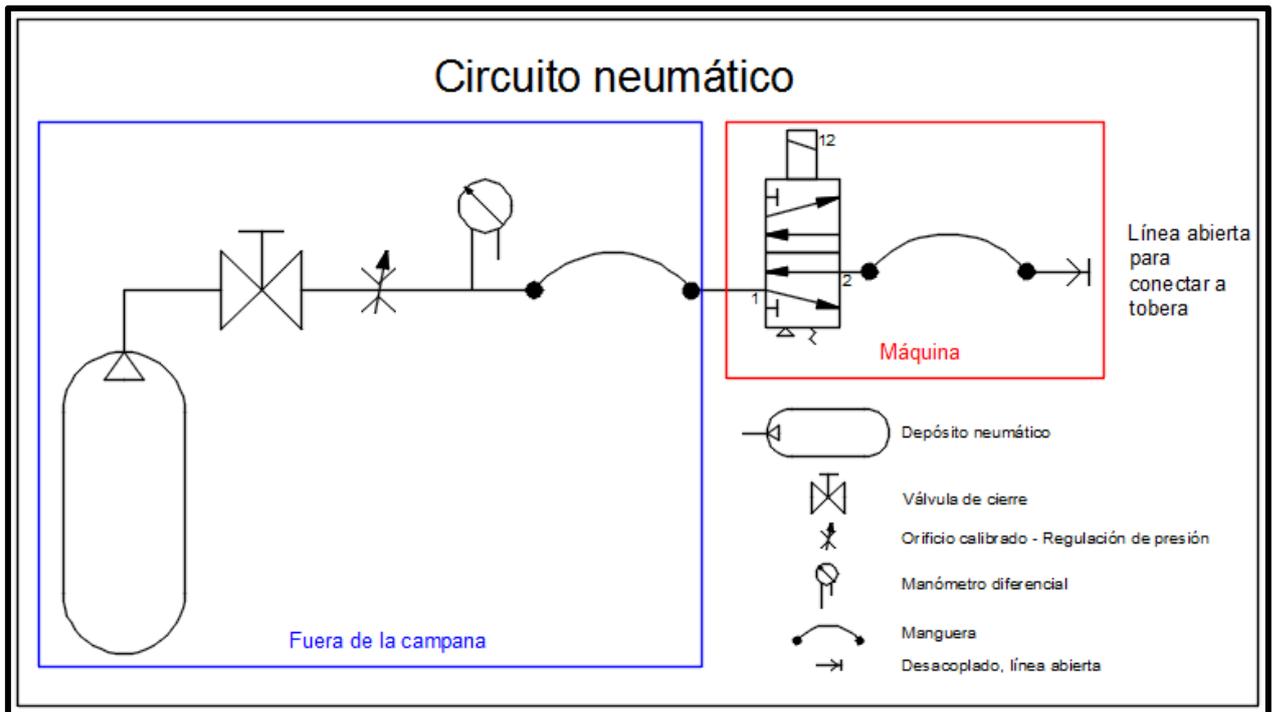


Figura 4. 11 – Circuito neumático

4.2.2.3 – Placa calefactora

La placa calefactora es la encargada de elevar y mantener la temperatura de las muestras o probetas a un valor de referencia para que el recubrimiento por spray sea correcto. La misma ya se encontraba en la campana de seguridad previa a comenzar con el proyecto.

La placa corresponde al modelo CERAN® S22 de Harry Gestigkeit GMBH, cuya hoja de datos se muestra en el anexo G junto a la información correspondiente a la etapa de control. La misma posee un rango de variación de temperatura de 50 a 500 [°C], y posee un controlador externo con un dial (potenciómetro) para variar la temperatura deseada. La figura 4.12 muestra la imagen comercial de la misma:

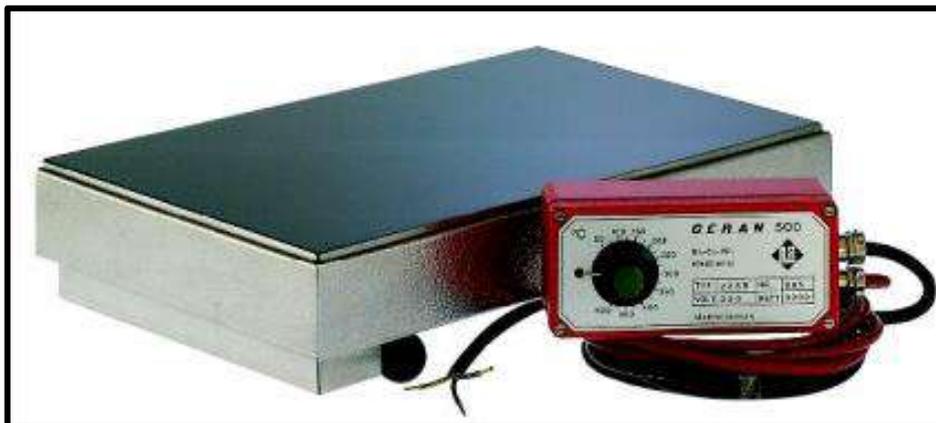


Figura 4. 12 - Placa calefactora CERAN SR22

Dentro de los objetivos de nuestro proyecto incluimos el control de la temperatura desde la PC, para que el usuario pueda configurar todos los parámetros desde la misma (excepto la presión que es manual mediante válvula reguladora de presión y manómetro), es por ello que hemos diseñado un circuito electrónico capaz de reemplazar el control manual que posee la placa por defecto, el cual será explicado con detalle más adelante en hardware de soporte.

4.3 – UNIDAD TERMINAL REMOTA (RTU)

En nuestro sistema poseemos un único RTU. Éste es el encargado de procesar las señales de los sensores, procesarlos o enviarlos al MTU para que el usuario decida y actuar según corresponda.

La función del RTU es realizada por el microcontrolador, el cual utiliza para su funcionamiento hardware adicional, como lo es una etapa de potencia de sus salidas, aislación para protegerlo tanto en entradas como salidas, y placas de distribución de cableado para facilitar el montaje.

Es necesario un controlador para procesar las señales eléctricas tal como se mencionó antes, y de esta manera generar señales digitales o en valores lógicos de variables para poder tomar las decisiones. Por otro lado, la potencia de las salidas del controlador no es suficiente para accionar los actuadores que se utilizan en esta aplicación tales como las bobinas de los motores PAP, el solenoide de la electroválvula, etc.

A lo largo de esta sección se irán explicando las diferentes partes de la RTU.

4.3.1 – Controlador: Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring¹) y el entorno de desarrollo Arduino (basado en Processing²). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a una PC, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP, etc.).

En nuestro caso hemos optado por utilizar Arduino debido a las siguientes importantes ventajas frente a otros fabricantes de microcontroladores y plataformas de desarrollo:

- *Economía* – Tanto la placa Arduino como los accesorios que ésta requiere como etapas de potencia, sensores, etc. son de bajo costo.
- *Cross-platform* – El software de Arduino puede ser utilizado en los sistemas operativos más utilizados (Windows, Macintosh OSX, Linux).
- *Entorno de programación simple* – El entorno es fácil de usar para principiantes y a su vez es lo suficientemente flexible para que los usuarios avanzados desarrollen sus propios productos.
- *Software open source y extensible* – El software de Arduino es open source y así también sus herramientas. Puede ser fácilmente expandido a través de librerías C++ o el lenguaje de programación AVR C propio del microcontrolador.

¹ Entorno de desarrollo (IDE). Incluye editor de texto y compilador.

² Lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java.

- *Hardware open source y extensible* – El Arduino está basado en microcontroladores Atmel ATmega, dentro de los cuales pueden ser modificados según cada usuario para crear una propia versión, extenderlo e incluso mejorarlo a criterio para cada aplicación.

Hemos presentado mayor información acerca de la familia de Arduino en el anexo A en los cuales se tratan las características constructivas y el software necesario.

4.3.1.1 – Selección de la placa

De todas las familias que hay disponibles, hemos optado por seleccionar el Arduino UNO, la cual presenta las siguientes características:

- 14 entradas/salidas digitales (6 salidas PWM).
- Procesador ATmega328.
- Velocidad CPU de 16 [Mhz].
- Tensión de operación de 5 a 12 [V].
- 6 entradas analógicas.
- EEPROM 1 [KB].
- SRAM 2 [KB].
- Flash 32 [KB].
- 1 conector USB.
- 1 conector UART.

Para la selección de la misma, nos hemos basado en los requerimientos de nuestro proyecto:

- 6 salidas digitales para control de motores en tres ejes (dirección y pulsos/pasos).
- 1 salida digital para habilitación de los drivers de motores.
- 1 salida PWM para control de temperatura de la placa calefactora.
- 1 salida digital para relé de encendido de placa calefactora.
- 1 salida digital para accionar electroválvula.
- 3 entradas digitales para finales de carrera.
- 1 entrada digital para sensor diferencial de presión.
- 1 puerto USB para comunicación con PC.

- Tensión de funcionamiento de 5V (para alimentarlo a través de USB desde la PC).
- Mejor criterio costo-beneficio.

De las 14 entradas/salidas digitales que tiene disponibles, una debe ser utilizada para la salida PWM, y los pines 0 y 1 son utilizados internamente para la comunicación del microcontrolador con otros dispositivos y con la PC, denominados “TX” y “RX”. Por lo tanto quedan disponibles sólo 11. Se requieren 9 salidas digitales y 4 entradas también digitales. Como no es suficiente, es posible adoptar dos entradas analógicas de las seis disponibles como entradas digitales.

Bajo esta configuración entonces es posible usar el Arduino UNO como controlador de nuestra máquina.

4.3.1.2 – Limitaciones

Como todo dispositivo electrónico, la placa tiene ciertas limitaciones en cuanto a potencia, corriente máxima y velocidad de procesamiento. A continuación enumeramos una serie de limitaciones que presenta para nuestra la aplicación la placa Arduino UNO y de qué manera tratamos de evitarlas.

- Los terminales de Arduino configurados como salidas (OUTPUT), quedan en un estado de baja impedancia y pueden proporcionar 40 [mA] de corriente a otros dispositivos y circuitos (esto no ocurre cuando son configurados como entradas (INPUT, por defecto), ya que poseen una resistencia interna de 20 [K Ω] y se dice que quedan en estado de alta impedancia).

Esta corriente es suficiente para alimentar un diodo LED, pero no es suficiente para alimentar otras cargas de mayor consumo como relés, solenoides, o motores.

En nuestro proyecto, la mayoría de las cargas requiere más de 40[mA] para su funcionamiento y por lo tanto es indispensable una etapa de potencia intermedia para poder cumplir con los objetivos de control. Estas etapas serán explicadas en la sección siguiente.

- En todo proyecto electrónico existen transitorios al producirse un cambio de estado de alguno de sus elementos o dispositivos. Estos transitorios

generalmente producen sobretensiones o corrientes elevadas que pueden dañar partes del circuito electrónico y es por ello que conviene proteger de alguna manera al circuito lógico o de control, del circuito de potencia. Debido a éste inconveniente hemos decidido aislar al circuito de control utilizando optoacopladores, los cuales fueron ubicados en una placa adicional que también se explicará en la sección siguiente.

4.3.2 – Hardware de soporte

Se ha mencionado dentro de las limitaciones del controlador Arduino que es necesario utilizar dispositivos o elementos de potencia, protección y control adicionales a la placa estándar para lograr adquirir datos de los sensores y controlar los diferentes periféricos de nuestra máquina.

En esta sección se van a explicar cada una de esas etapas y como se conectarán con el Arduino.

4.3.2.1 – Shield CNC

Para explicar cuál es la shield que nosotros utilizaremos para nuestro Arduino, comenzaremos por definir que es precisamente una shield.

Una shield es una placa impresa que se puede conectar directamente encima de la placa Arduino y de este modo ampliar sus capacidades, pudiendo ser apiladas una encima de la otra. Es posible apilar varias de estas placas permitiendo que de manera compacta se logren muchas funciones extras a las propias de la placa.

En el momento de diseñar cual sería el montaje de las partes necesarias para controlar la máquina entre el Arduino y los sensores y actuadores, nos encontramos en la necesidad de utilizar una placa o más placas para facilitar la conexión entre ésta. Investigando en el mercado electrónico actual, hemos encontrado una shield que permite el montaje de los drivers para los motores PAP en una única placa, que permite además filtrar y proteger a los mismos de manera sencilla, teniendo conectores a los cuales directamente se pueden cablear los motores y demás dispositivos.

En la figura 4.13 puede observarse una imagen de la shield CNC de la cual estamos hablando, conectada sobre la placa Arduino UNO y los drivers de motores en los espacios destinados a su montaje.

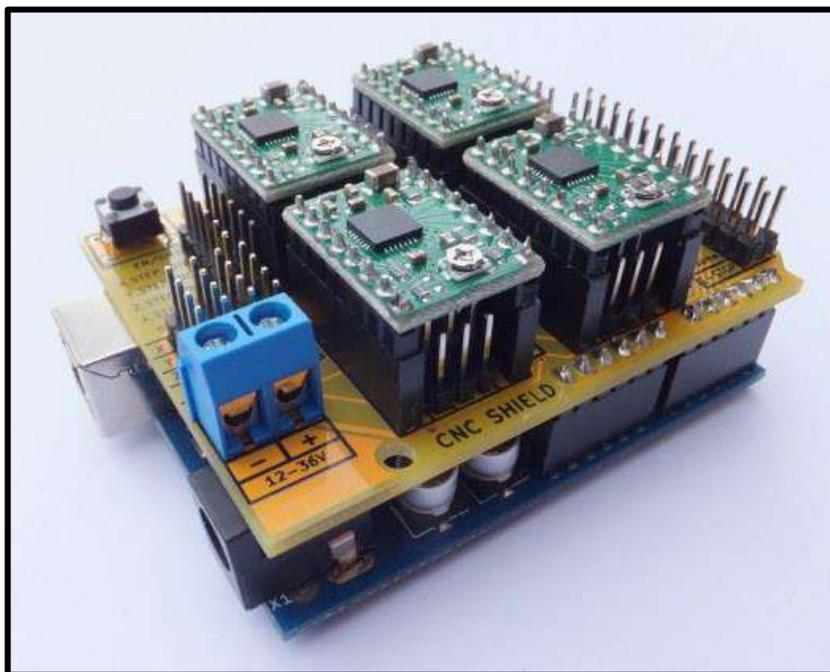


Figura 4. 13 - Shield CNC montada sobre Arduino UNO

Esta shield ha sido diseñada para ser compatible con GBRL (firmware creado para Arduino que permite al mismo interpretar el código utilizado mundialmente para máquinas CNC, conocido como G-Code). Viene con una configuración propia para este tipo de aplicaciones que mostraremos a continuación, y nosotros analizamos si es posible modificarla para utilizarla en nuestra propia aplicación ya que es una herramienta para simplificar espacio y conexionado de circuitos.

En la figura 4.14 se observa la configuración de pines del Arduino para ser utilizado este con un software ya diseñado para interpretar el código G (GBRL), y de esa manera seguir cualquier trayectoria a lo largo de la máquina.

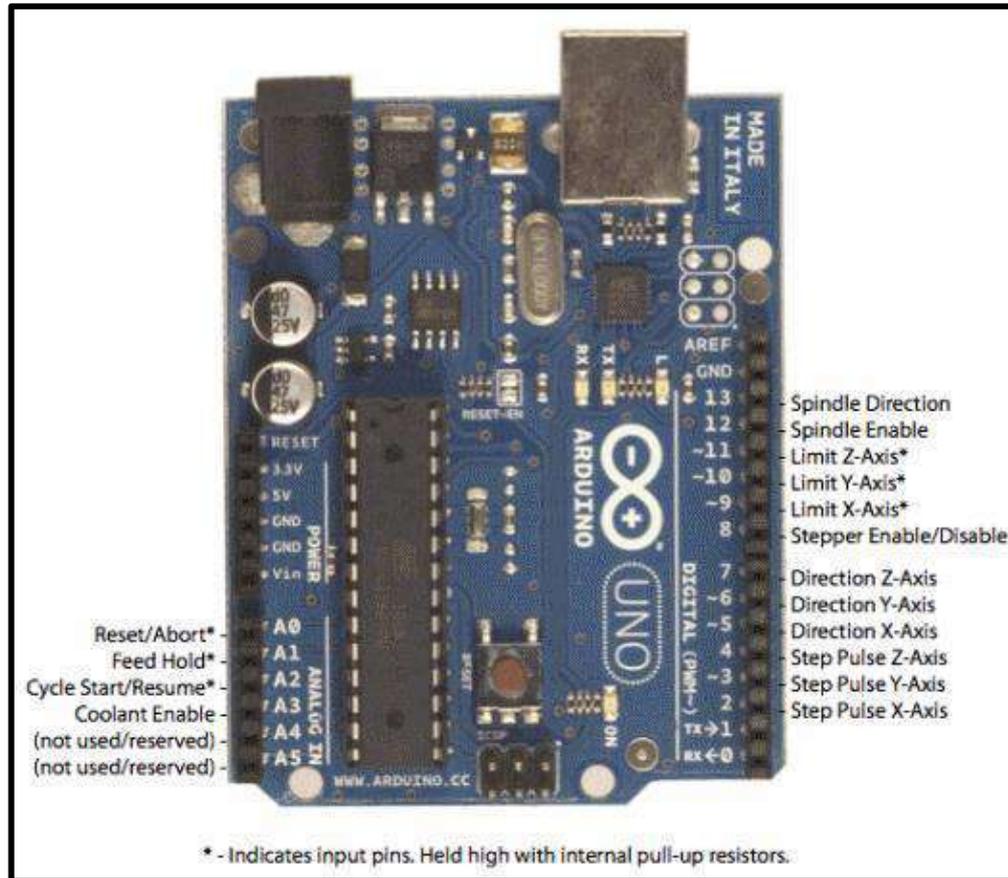


Figura 4. 14 - Configuración de terminales de Arduino para ser compatible con GBRL

Los terminales tienen el mismo lugar dentro de la Shield CNC y en el Arduino. Todos los pines que hacen referencia a motores (2 a 8) se dirige a los headers propios donde se colocan los drivers, pasando donde corresponde por capacitores y resistencias, por lo que no es recomendable alterarlos y que es fácilmente programable. Los terminales 9, 10 y 11 correspondientes a finales de carrera en ambas placas están configurados como entradas digitales y conviene utilizarlos como tal para no alterar el funcionamiento, ya que de todos modos debemos utilizarlos con el mismo fin. Los terminales 12 y 13 de Spindle (en caso de utilizar un motor para fresar o taladrar), se encuentran configurados como salidas digitales y por lo tanto es posible darles esa finalidad también en nuestra aplicación. Por último, sabemos que requerimos de una entrada digital más y una salida digital más, para lo cual utilizaremos los pines A4 y A5 correspondientes a entradas analógicas y que están sin uso en la configuración predeterminedada de GBRL.

Considerando lo dicho en el párrafo anterior, asumimos que es posible la utilización de ésta shield para nuestra aplicación y por lo tanto decidimos adquirirla ya que facilita el diseño y la fabricación, obteniendo un producto de calidad y que puede ser sencillamente montado sobre la placa Arduino UNO.

A continuación se enumeran entonces las funciones básicas o principales de la shield y aparte las funciones adicionales que facilita dentro de la misma placa electrónica:

Funciones principales:

- Permitir el montaje de los drivers para cuatro motores PAP (tres ejes y uno puede ser clonado).
- Permitir alimentación en 24[V] para los motores.
- Permitir conexión directa de los finales de carrera.
- Permitir conexión directa de los demás terminales de temperatura, presión, etc.

Funciones secundarias:

- Permite clonar uno de los tres ejes en el caso de querer controlar dos motores para un mismo eje, conectado al driver ubicado en la posición A de la shield.
- Poder configurar los motores para que trabajen en micropasos (1/2, 1/4, 1/8, 1/16) a partir de los jumpers M0, M1 y M2 ubicado debajo de cada driver tal como se puede ver en el layout e imagen real de la shield mostrados en la figura 4.15.

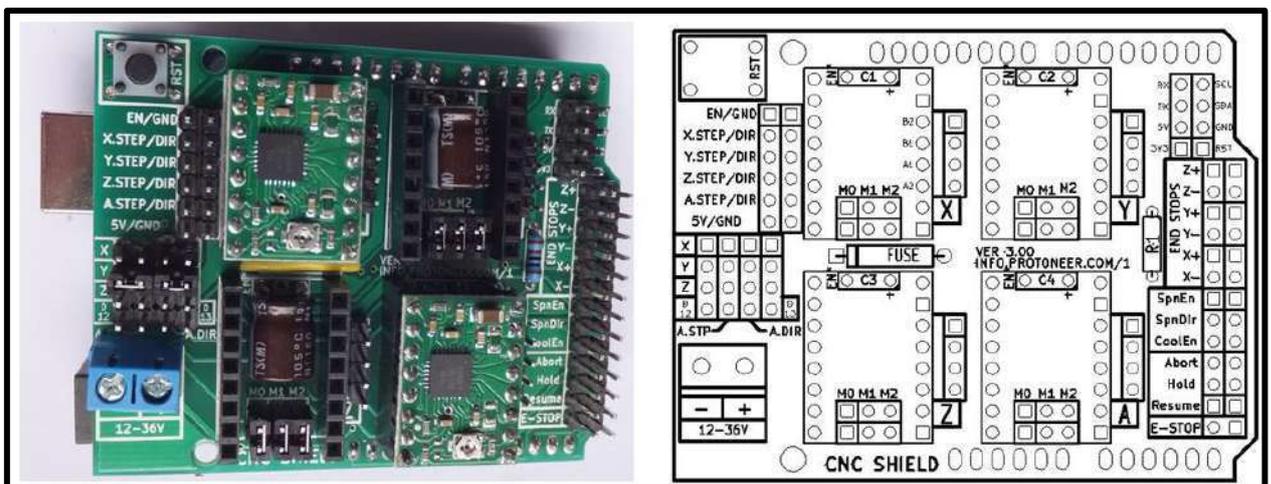


Figura 4. 15 - Fotografía y layout de la shield CNC

En nuestro caso hemos clonado el eje longitudinal al que hemos adoptado como eje X y por lo tanto debemos configurar dos jumpers para que el mismo sea repetido al driver A. Para ello el conexionado debe ser el que se muestra en la figura 4.16:

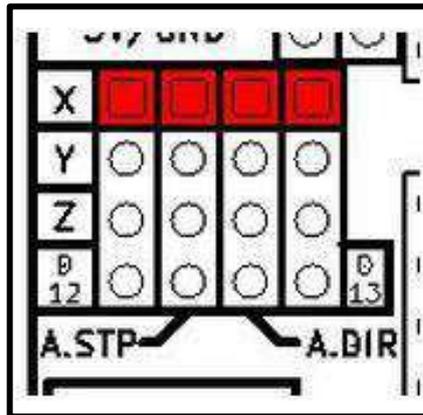


Figura 4. 16 - Conexión de jumpers para clonar eje X

También hemos adoptado una configuración de 1/8 de pasos para el eje longitudinal cuyo sistema de transmisión permite avances a grandes velocidades y utilizando una resolución de pasos menor se logra una mayor suavidad en el desplazamiento del carro sobre los ejes. Para realizar esto, debimos colocar tanto para el eje X como el clon A la configuración de jumpers que se muestra en la figura 4.17 en los pines para tal fin que se encuentran debajo del header del controlador.

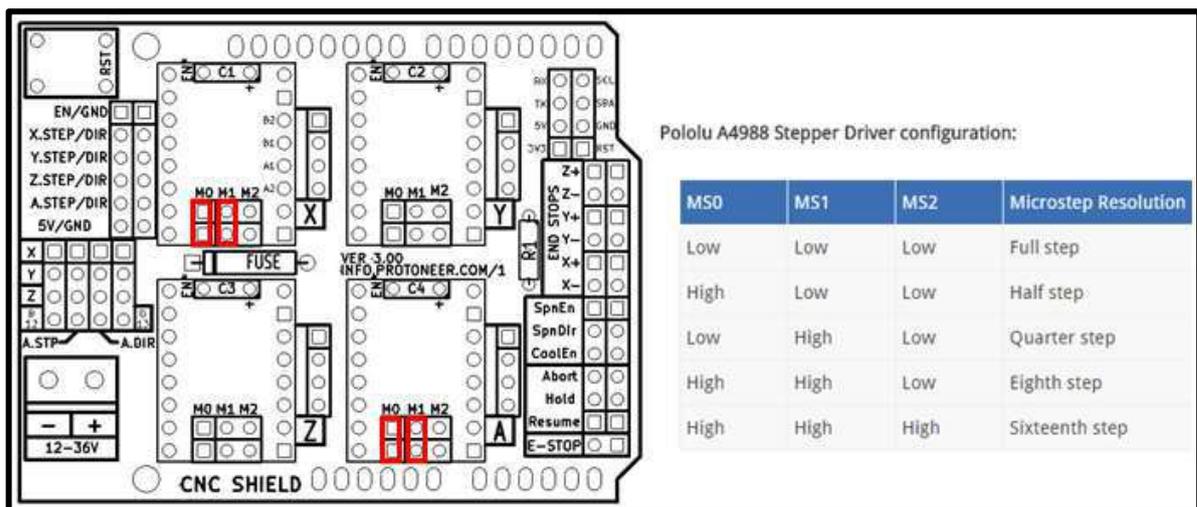


Figura 4. 17 - Configuración de jumpers para lograr micro pasos 1/8 en eje longitudinal

Con estos pasos quedaría configurada la shield CNC para que nuestro controlador tenga acceso a todos los elementos terminales de la máquina.

En la figura 4.18 se muestra la configuración de pines que hemos adoptado nosotros para programar nuestro controlador Arduino y que por supuesto coinciden con los de la shield CNC.

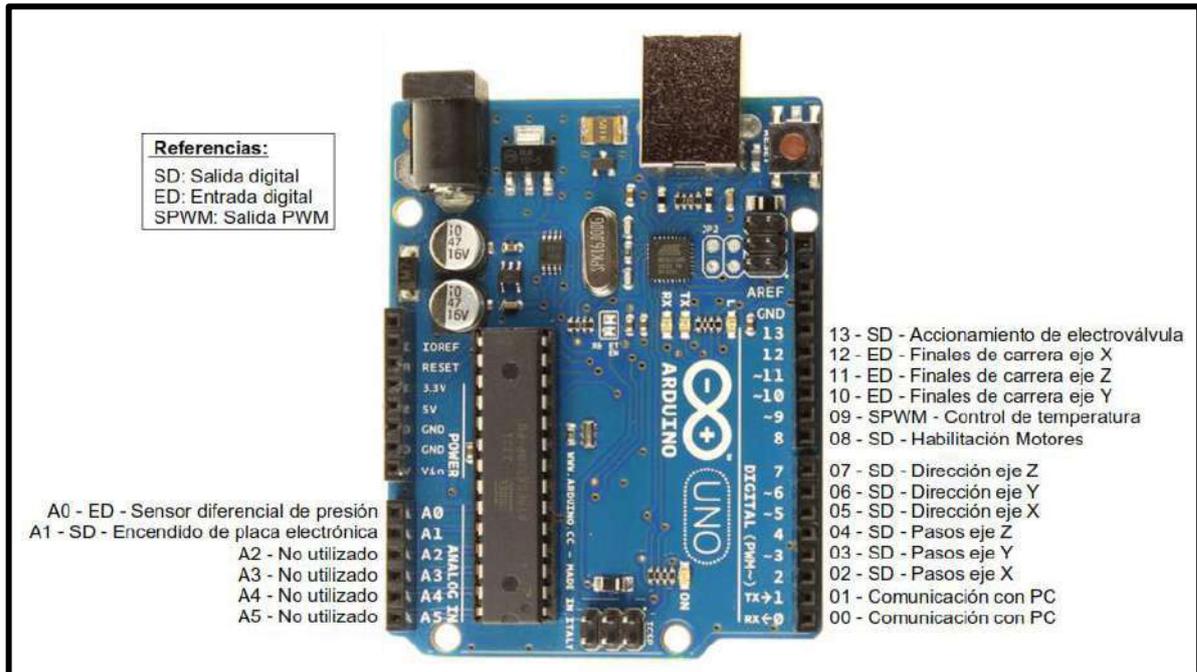


Figura 4. 18 - Configuración de terminales de Arduino para nuestro proyecto

4.3.2.2 – Driver Polulu a4988 para motores PAP

Para controlar los motores desde un Arduino, tal como se ha explicado antes, se requiere de una etapa de potencia. Este controlador, además de proveer la misma, cumple otras funciones importantes que facilitan el control de los mismos, tales como opción de configurar los micropasos, permitir una regulación de corriente justa para el motor sin necesidad de una tensión estable, y por otro lado, reducir el número de pines necesarios para controlar un motor. Esta información se seguirá ampliando a medida que se detalle el driver utilizado.

Indagando en la web y consultando con fabricantes de impresoras 3D y maquinas CNC, encontramos que uno de los controladores más utilizados para este tipo de aplicaciones es el Polulu A4988. En la figura 4.19 se muestra el controlador.

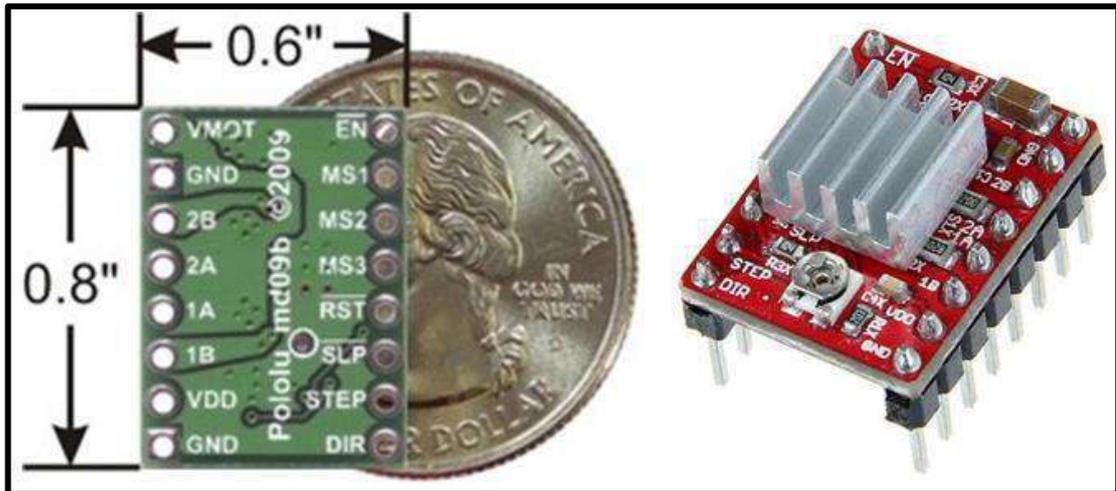


Figura 4. 19 - Controlador de motores PAP, Polulu a4988.

Este driver permite entre otras cosas controlar el motor utilizando solo dos pines de nuestro Arduino, uno para indicar la dirección de giro y otro por donde se envía el tren de pulsos que hará que el motor se mueva un paso por pulso.

Por otro lado este controlador hace un control de corriente a lazo cerrado midiendo la caída de tensión que se produce en unas pequeñas resistencias colocadas en serie con los devanados del motor, luego compara esta caída de tensión con un valor de referencia el cual se puede ajustar mediante un potenciómetro (indicado en la figura 4.20 por la flecha roja) y finalmente varía la tensión de salida del controlador mediante PWM (modulación de ancho de pulso).

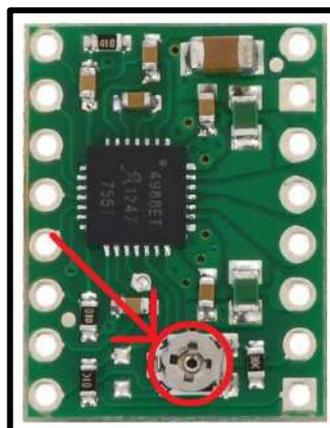


Figura 4. 20 - Potenciómetro para ajustar valor de referencia de tensión

Este controlador está basado en el chip A4988 de Allegro, es para manejar exclusivamente motores paso a paso bipolares de hasta un máximo de 1A por bobina sin disipador o 2A por bobina con un disipador y flujo de aire forzado

adecuados. Por otro lado admite 5 resoluciones de micropasos, “full-step”, “half-step”, “quarter-step”, “eighth-step” y “sixteenth-step” (ver anexo F).

En la imagen 4.21 se muestra un diagrama de conexionado elemental para el controlador:

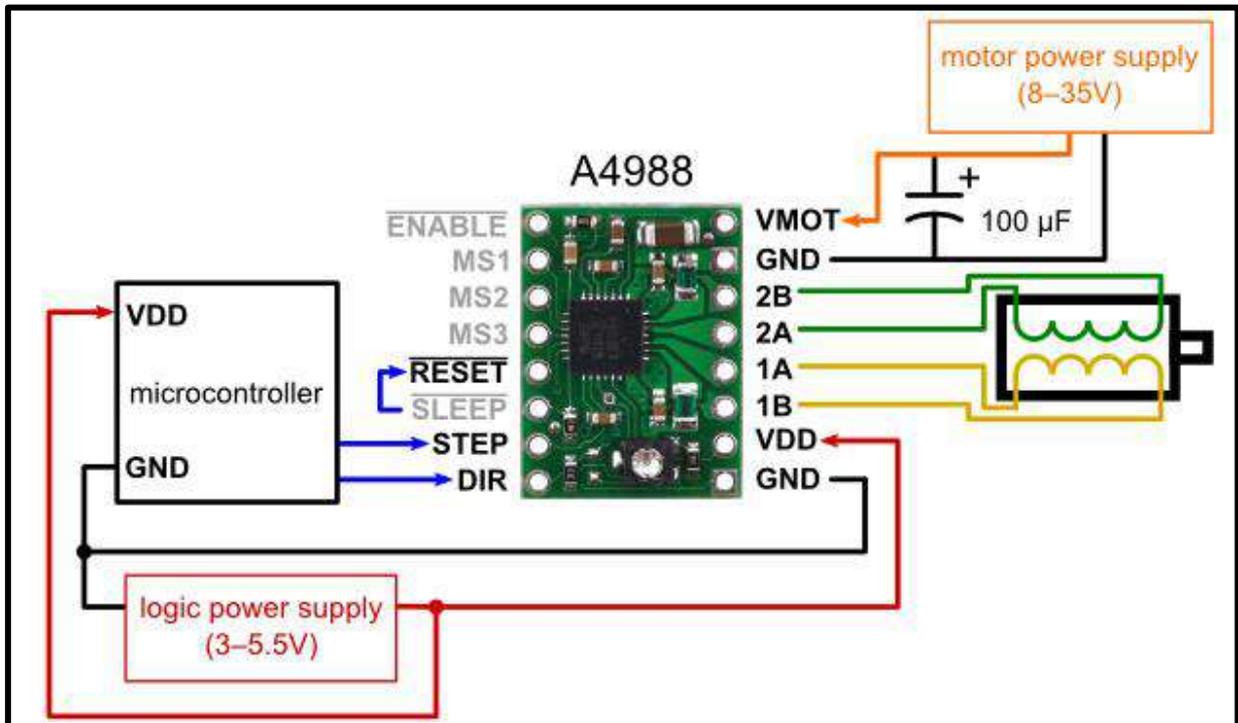


Figura 4. 21 - Conexionado adecuado para el driver

La falta de experiencia con estos controladores ha provocado que a lo largo del desarrollo hayan surgido ciertos percances en donde hemos quemado algunos controladores y hemos ido aprendiendo a manipularlos. El total de controladores comprados fue de ocho, de los cuales cuatro se encuentran funcionando y los otros cuatro se encuentran obsoletos por las siguientes razones:

- Se recibió uno ya sin funcionar,
- Otro se quemó por conectar la fuente de tensión con la polaridad invertida,
- Otro se quemó por colocar el disipador incorrectamente y se hizo un cortocircuito entre varios de sus pines,
- Otro se quemó mientras era probado en una protoboard sin el capacitor a la entrada del circuito lo que provocó que al variar la corriente tan rápidamente en el motor (carga inductiva prácticamente pura) se generen sobretensiones suficientemente elevadas para dañar los mosfet del driver.

Una vez que pudimos reponer los controladores optamos por realizar un pequeño ensayo colocando el capacitor necesario y midiendo con un osciloscopio las tensiones en bornes de una de las bobinas del motor e ir aumentando lentamente la tensión de alimentación.

El circuito de conexión fue similar al mostrado en la figura 4.21, con el agregado de que las puntas del osciloscopio fueron conectadas una en cada extremo de uno de los bobinados del motor y las masas fueron conectadas a la masa de la fuente de alimentación. De esta manera se pudo visualizar la tensión de cada extremo del devanado medida desde la masa de la fuente. En el ensayo, nuestro microcontrolador no se encontraba enviando pulsos, sino que la única señal observable es la correspondiente a la modulación que realiza el driver para hacer el control de corriente.

En la figura 4.22 se muestra la señal de la tensión que se pudo visualizar en el osciloscopio, en el caso de la imagen, se estaba alimentando el sistema con 12V y vemos que con el capacitor conectado existe un sobre pico de tensión en las bobinas de aproximadamente un 60% de la tensión de alimentación. Aseguramos que sin el capacitor, la tensión en bornes de las bobinas hubiera sido mucho mayor, lo suficiente para dañar los MOSFETs del controlador.

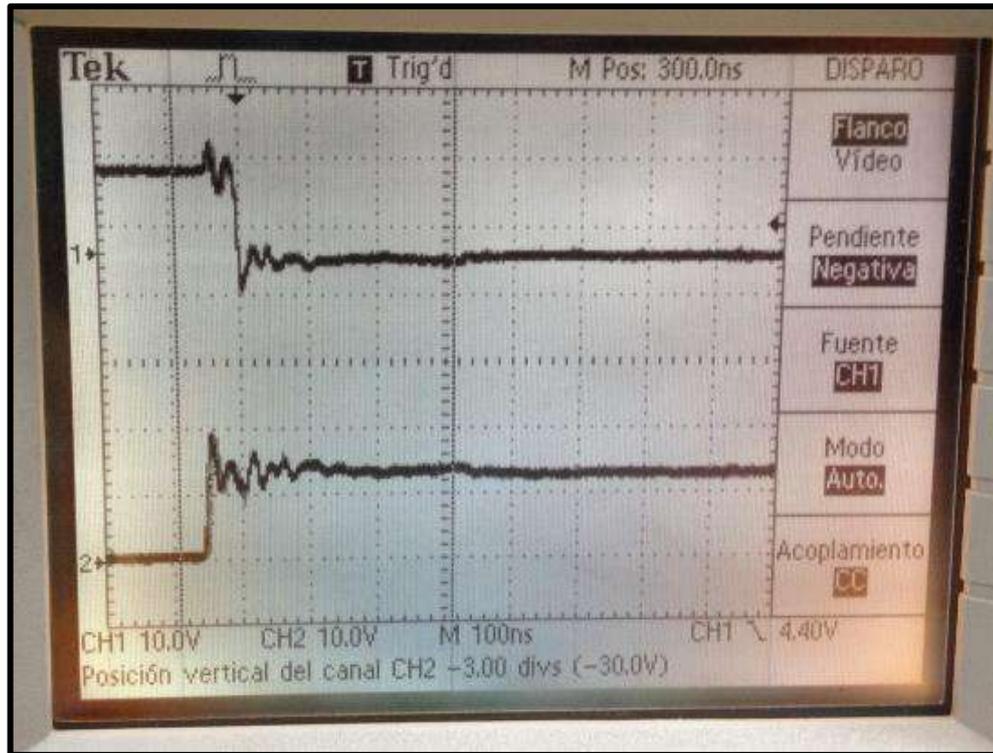


Figura 4. 22 - Vista de pantalla de osciloscopio durante ensayos al driver

4.3.2.3 – Placa de distribución de conexiones y aislación de controlador

Esta es una placa que ha sido diseñada por nosotros para ser montada dentro del gabinete que oficia de RTU o controlador remoto de la máquina. Cumple con dos funciones principales:

- Aislar los terminales del controlador de sobretensiones que se puedan presenciar en la etapa de alta tensión.
- Facilitar el cableado de los diferentes actuadores y sensores para luego ingresar a la shield CNC y de ésta manera al Arduino.

Para cumplir con la primera de las funciones mencionadas, se ha optado por utilizar optoacopladores, los cuales permiten aislar el circuito de baja tensión y corriente de un microcontrolador, de la etapa de alta tensión. Como mencionamos anteriormente en la introducción, esto es necesario dado que si se presenta alguna sobretensión ya sea por la red de alimentación o por algún componente presente en la instalación, puede dañar permanente el microcontrolador, siendo este un componente costoso dentro de la instalación.

El optoacoplador es un dispositivo de emisión y recepción óptica que funciona como un interruptor activado mediante luz, la cual es emitida mediante un LED que satura un componente receptor de luz o fototransistor. Como el acople es óptico, las sobretensiones o sobre corrientes, no dañarán el controlador, quedando este altamente aislado. El símbolo electrónico de este dispositivo es que se puede ver en la figura 4.23, donde se ve el LED emisor y el fototransistor.

El LED emisor es encendido desde el lado de control o de baja tensión, activando el fototransistor de la etapa de potencia, es decir, cerrando el interruptor de potencia y por lo tanto permitiendo el funcionamiento de la etapa de alta tensión.

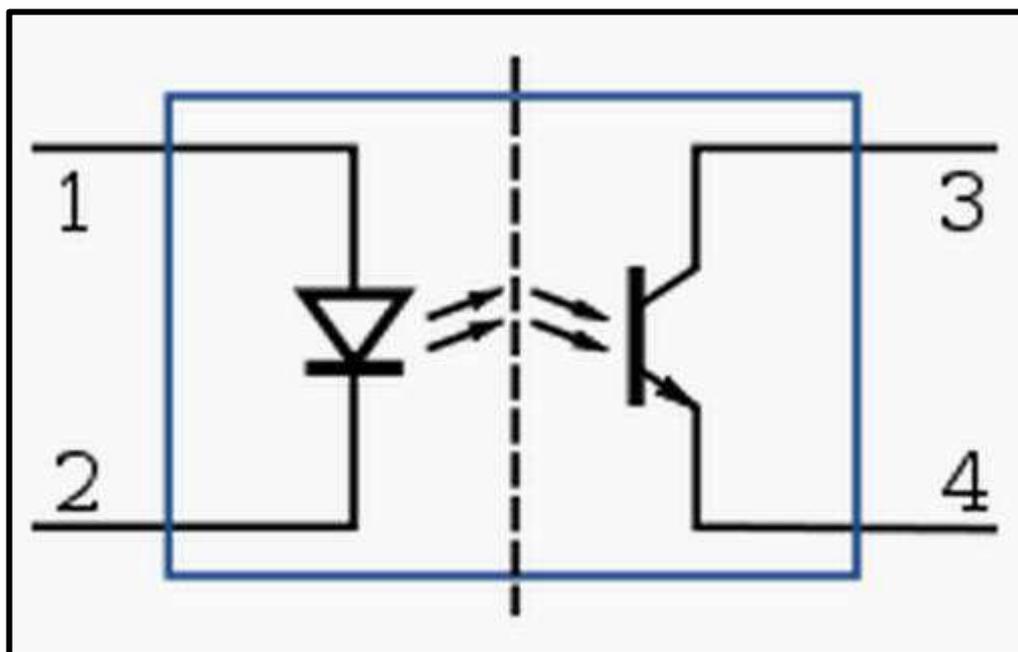


Figura 3. 23 - Símbolo electrónico de un optoacoplador

Para cumplir con la segunda función, de facilitar el conexionado de los cables entre la máquina y el microcontrolador, se han colocado múltiples pines dentro de esta placa. Por un lado los que se dirigen a la máquina, y luego un conjunto de pines a los cuales se les conecta un cable de bus de datos que se envía a la shield.

En la figura 4.24 se muestra el diseño realizado en el software Proteus, para luego realizarla y colocarla en nuestro gabinete.

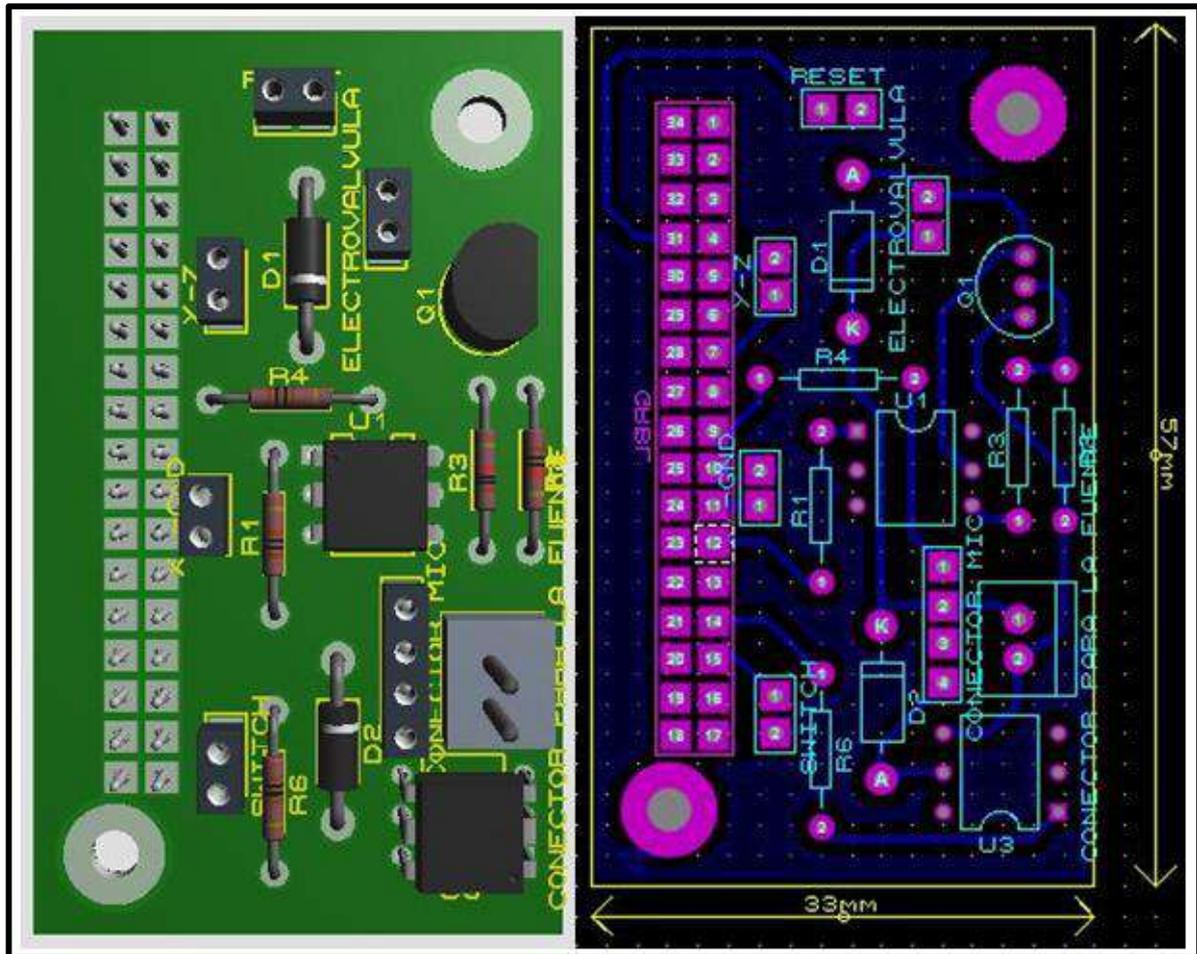


Figura 4. 24 - Vista superior (izq.); Layout de pistas y puntos de conexión (der.)

En un principio intentamos fabricar nosotros mismos dichas placas, mediante la utilización de un papel termo sensible. En este se imprimió el circuito y luego de presionarlo y apoyarlo a una superficie caliente a este sobre el cobre de una placa, se esperaba que todo el circuito se refleje en el cobre. El proceso no resultó de la manera que esperábamos. En la figura 4.25 se observa el layout impreso sobre el papel y luego el conjunto del papel pegado al cobre dentro de un recipiente con agua para poder despegar al mismo.

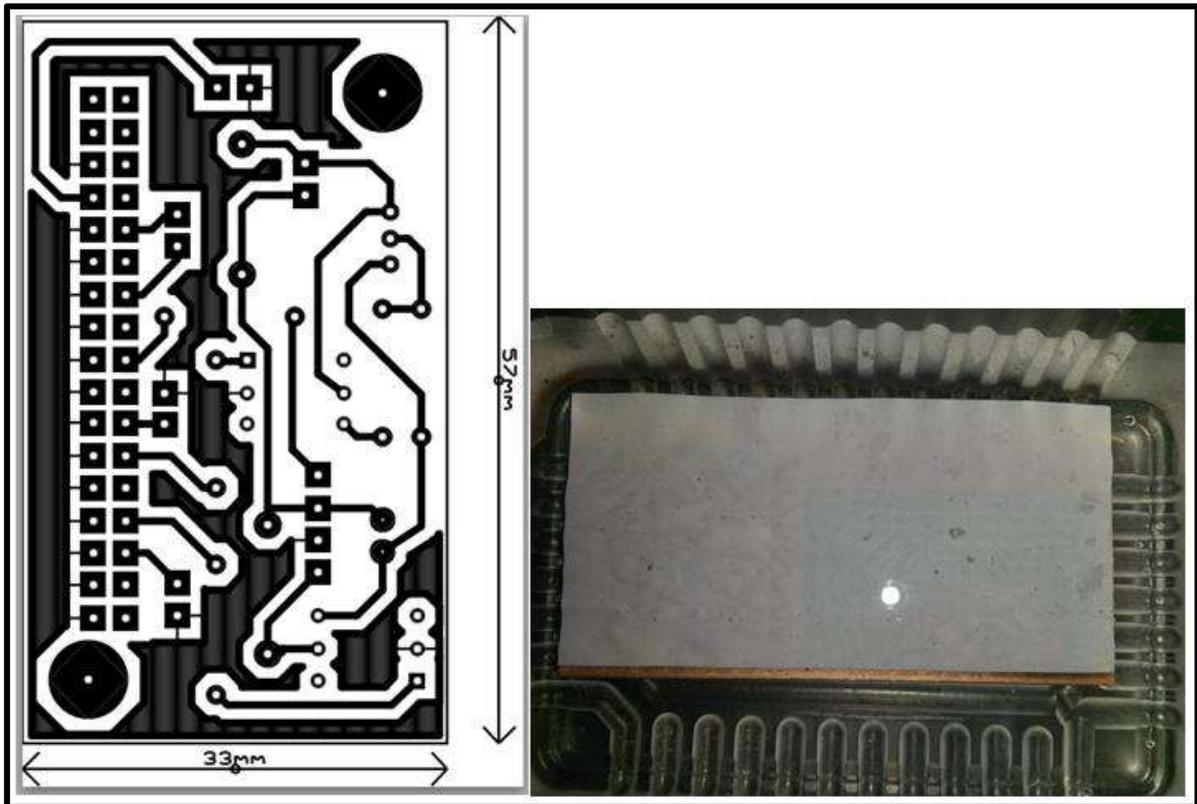


Figura 4. 25 - Layout impreso sobre papel (izq.); Papel ya presionado con una plancha caliente y colocado en agua para aflojar y despegar el papel (der.)

Al quitar el papel de la placa, observamos que la tinta no estaba en su totalidad adherida al cobre, por lo que se la reforzó con una fibra indeleble, quedando como el de la figura 4.26, para luego sumergirla en el percloruro de hierro, encargado de corroer el cobre no protegido por la tinta, y así crear las pistas que necesitamos para nuestro circuito.

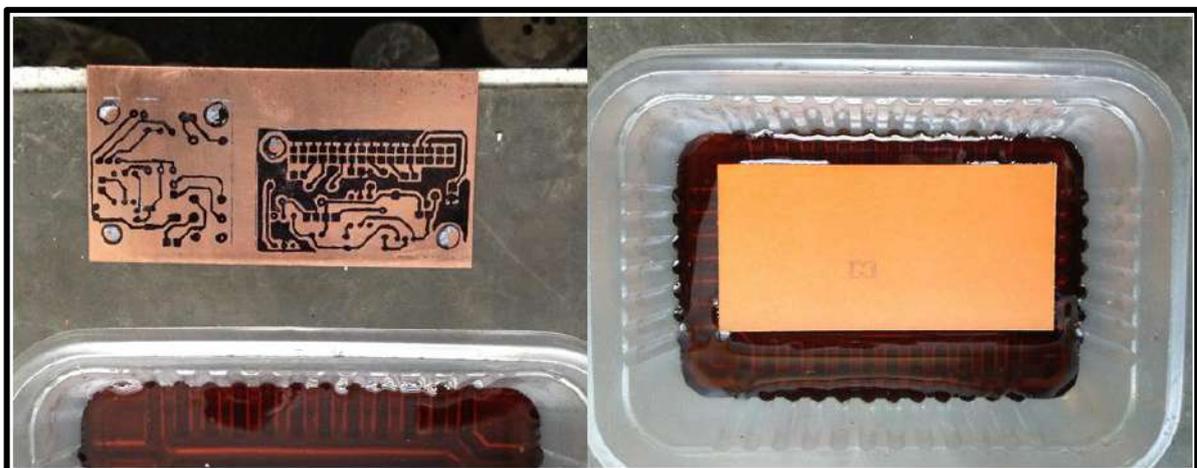


Figura 4. 26 - Layout traspuesto sobre el cobre, reforzado con un fibra indeleble (izq.); Placa colocada dentro de percloruro de hierro (der.)

El resultado final de la placa no fue el que esperábamos, por otro lado el circuito nos quedó en espejo con respecto al que necesitábamos, lo que imposibilitó la fijación del lado frontal de algunos componentes. Es por ello que en lugar de rehacerla, se decidió enviarla a hacer más allá de la demora que esto requería, aunque en cuanto a costos no había diferencia significativa. En la figura 4.27 se observan los resultados obtenidos al imprimirla nosotros, y la que enviamos a hacer.

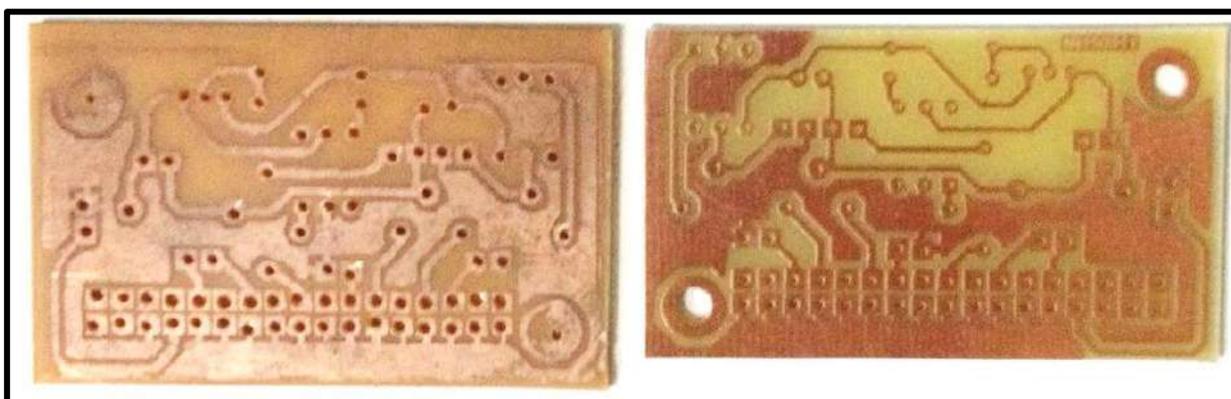


Figura 4. 27 - Resultado de nuestro intento (izq.); Resultado de un especialista (der.)

Finalmente se le soldaron los componentes. El resultado puede verse en la figura 4.28.

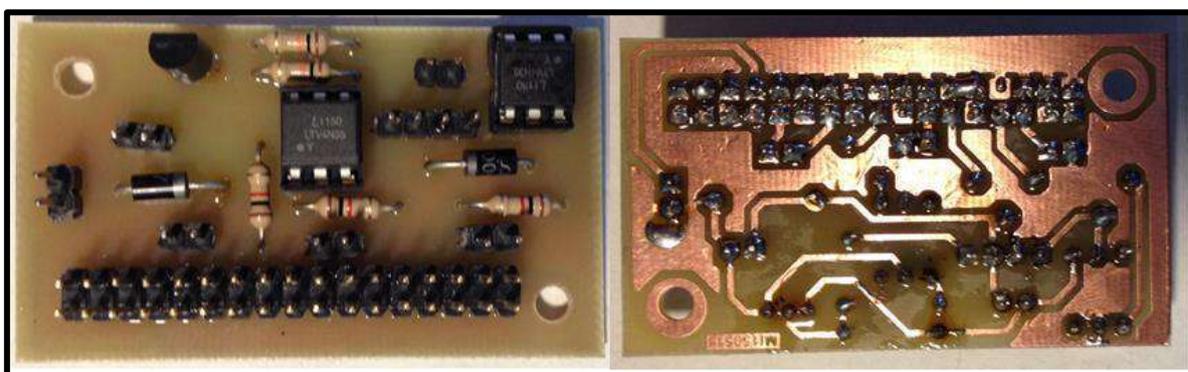


Figura 4. 28 - Vista superior placa terminada (izq.); Vista inferior placa terminada (der.)

4.3.2.4 – Placa para controlar temperatura

Otra placa necesaria para controlar los actuadores de nuestro proyecto, es la que nos permite controlar la placa calefactora o “hot plate” utilizado para calentar las muestras. Para controlar a la misma debíamos cumplir dos funciones:

- Activar la alimentación de la red.
- Enviar un valor de temperatura al cual debía estar la placa.

Para realizar la primera de estas funciones, se optó por un utilizar un relé que se maneje con la tensión de 24 [V] del controlador de nuestro gabinete de control o RTU, el cual activamos desde el microcontrolador a través de un optoacoplador. Este relé se encuentra en paralelo con el interruptor propio de la controladora permitiendo que se alimente con al menos uno de los dos activado (condición lógica OR).

Para la segunda función de enviar el valor de referencia, hemos tenido que investigar y relevar el circuito propio de la placa controladora. Esta misma funciona con un potenciómetro capaz de regular la temperatura entre 50 y 500 [°C], y la realimentación a partir de una termocupla tipo “K”, la cual se pudo identificar por los colores normalizados de sus conectores y los valores de tensión de respuesta.

En un principio creíamos que el potenciómetro regulaba una tensión de referencia entre 0 y 24 [V] de la cual se alimenta todo el control de la placa calefactora. De este modo, enviar una tensión de referencia para reemplazar el control desde el potenciómetro nos resultaba muy sencillo y se podía realizar con un optoacoplador y un transistor simplemente. Pero al realizar las mediciones de tensión de salida del potenciómetro, nos encontramos con que éste regulaba entre 1 [mV] y 33 [mV], lo que nos permite un rango de regulación de 32 [mV].

Dicho rango de regulación no es casualidad, sino más bien similar al rango de diferencia de tensión que genera la termocupla al someterse la punta caliente a esas temperaturas (ver figura 4.29).

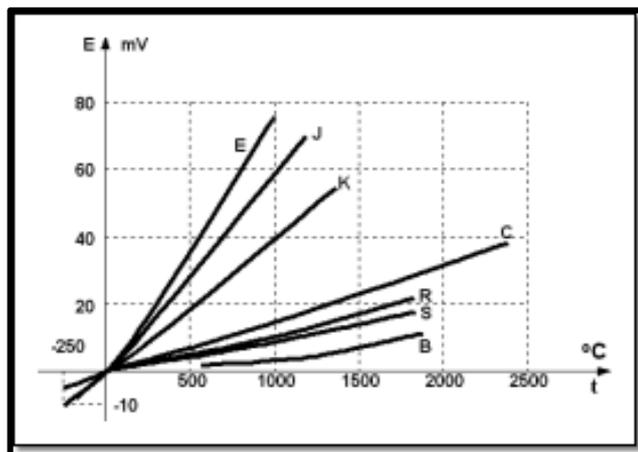


Figura 4. 29 - Curvas de respuesta teórica de termocuplas en función de la temperatura

La curva de la termocupla tipo “K” (Cromel – Alumel) es aproximadamente lineal para el rango de temperatura capaz de alcanzar la placa calefactora, es por ello que el principio de funcionamiento del control de la temperatura está basado en la comparación entre el valor de tensión del potenciómetro y el de la termocupla, y por ello los valores de tensión tan bajos para el potenciómetro. La resolución de esta termocupla es de $36 [\mu\text{V}/^{\circ}\text{C}]$.

Como se mencionó antes, se levantó el circuito de la placa original que permite configurar y controlar la temperatura de la placa, cuyo esquemático puede verse en la figura 4.30. Y sobre este esquema hemos planteado las diferentes opciones para alterar la misma, siempre con el objetivo de permitir una configuración manual o automático de la placa calefactora. Esto fue pensado así para que en casos en que se prefiera utilizar la placa al modo original, ya sea para el mismo proceso para el cual es diseñada o en otro uso.

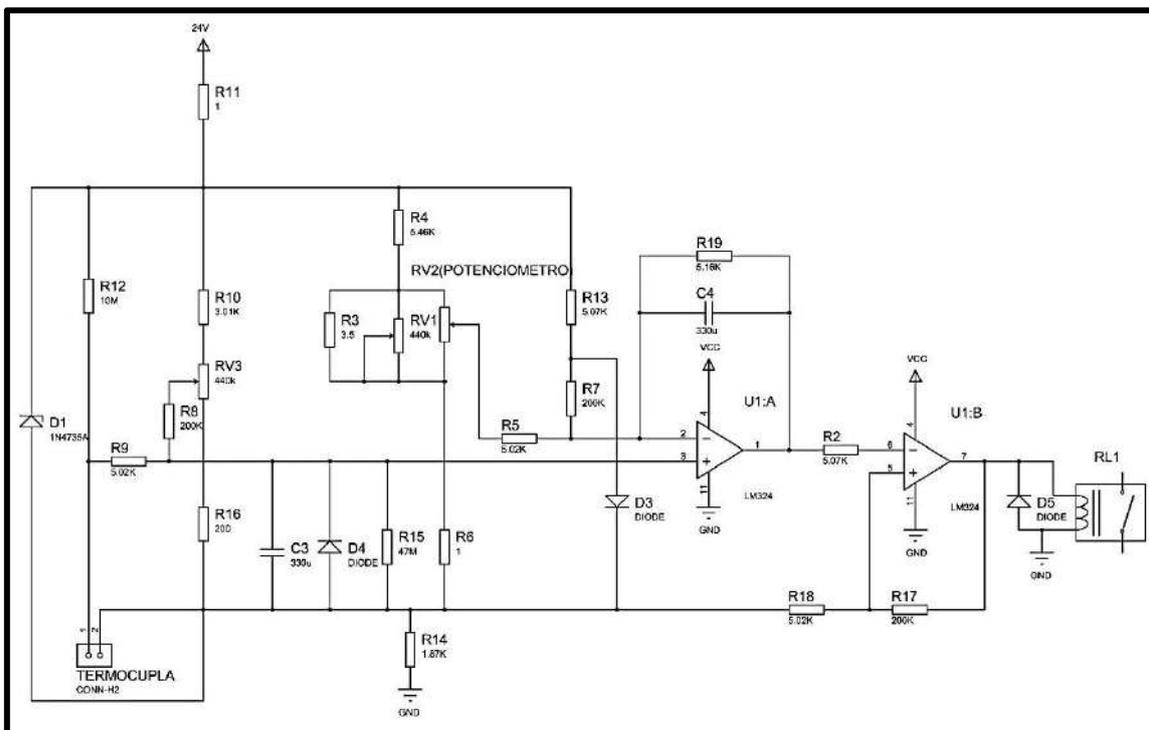


Figura 4. 30 – Esquemático del circuito electrónico para control manual de temperatura

De las opciones de intervenir al circuito con valores tan bajos de tensión, la que consideramos más indicada y por lo tanto fue utilizada es la que se muestra en la figura 4.31, junto con las referencias de sus conectores.

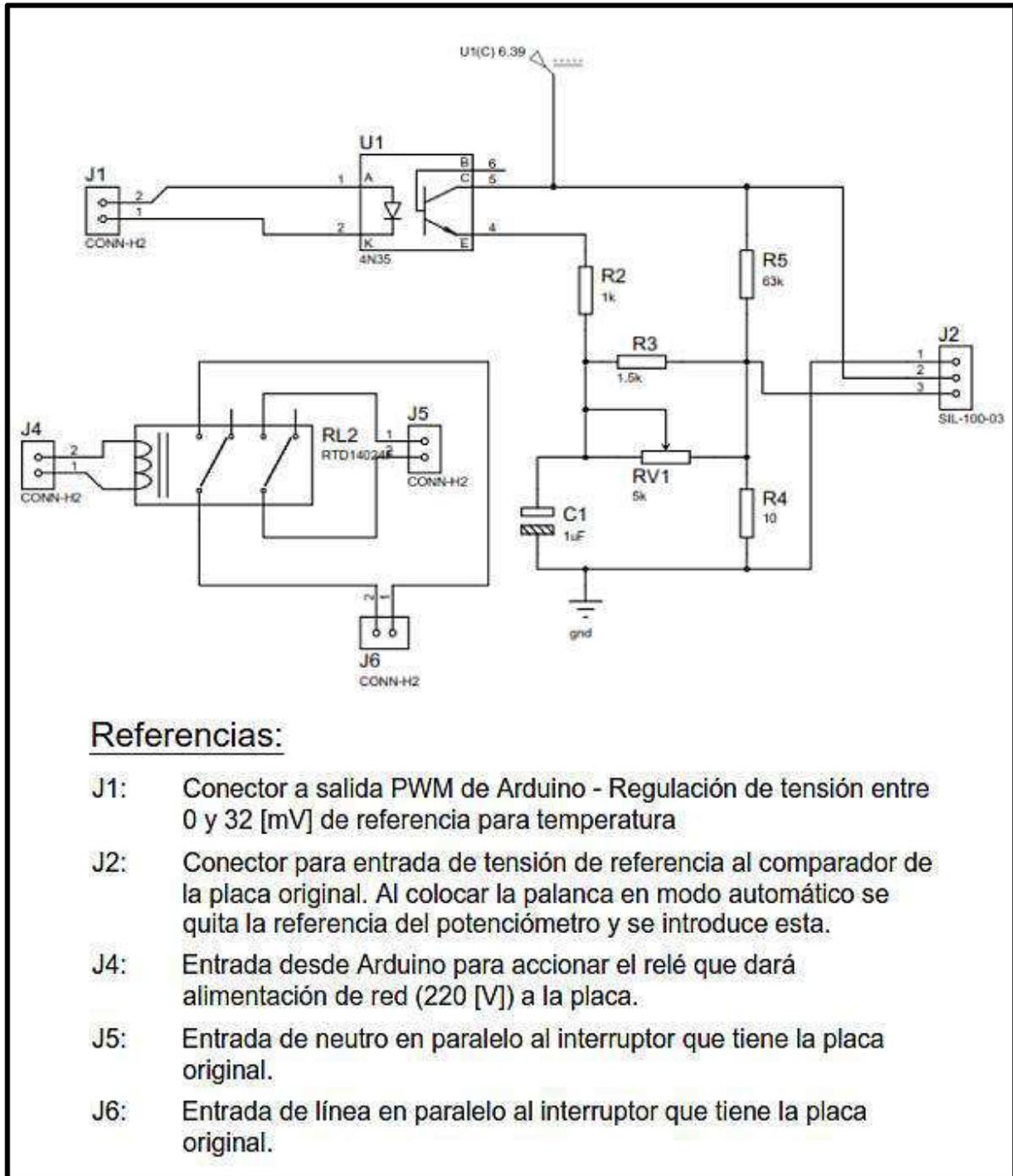


Figura 4. 31 - Esquemático de circuito para controlar la placa calefactora desde la PC

El control automático de la temperatura se realiza abriendo la tensión de referencia proveniente del potenciómetro, y reemplazándola por un circuito donde podemos intervenir con el valor de referencia generado por nuestro circuito.

Para ello debemos regular la tensión mediante PWM (“Pulse width modulation” o modulación de ancho de pulso) entre 1 y 33 [mV]. Para lograr dicha regulación hemos creado un divisor resistivo a partir de una tensión de referencia de 6,4 [V] que

tomamos desde la misma placa y se le colocaron filtros. De este modo, al estar nuestro transistor en corte, la tensión del punto bajo debería ser de 1 [mV], mientras que cuando está saturado, la tensión alcanzada sería de 33 [mV] para mantener la misma relación. Para valores intermedios, al modular mediante PWM se varía el ciclo de alto y bajo o en saturación y corte del transistor, generando el valor medio de tensión esperado entre los mencionados, para mayor detalle sobre PWM ver las referencias presentadas.

Finalmente se diseñó el circuito que debemos colocar dentro del gabinete controlador de la placa calefactora en Proteus y los resultados pueden verse en la figura 4.32. Y el esquemático de la misma en la figura 4.33

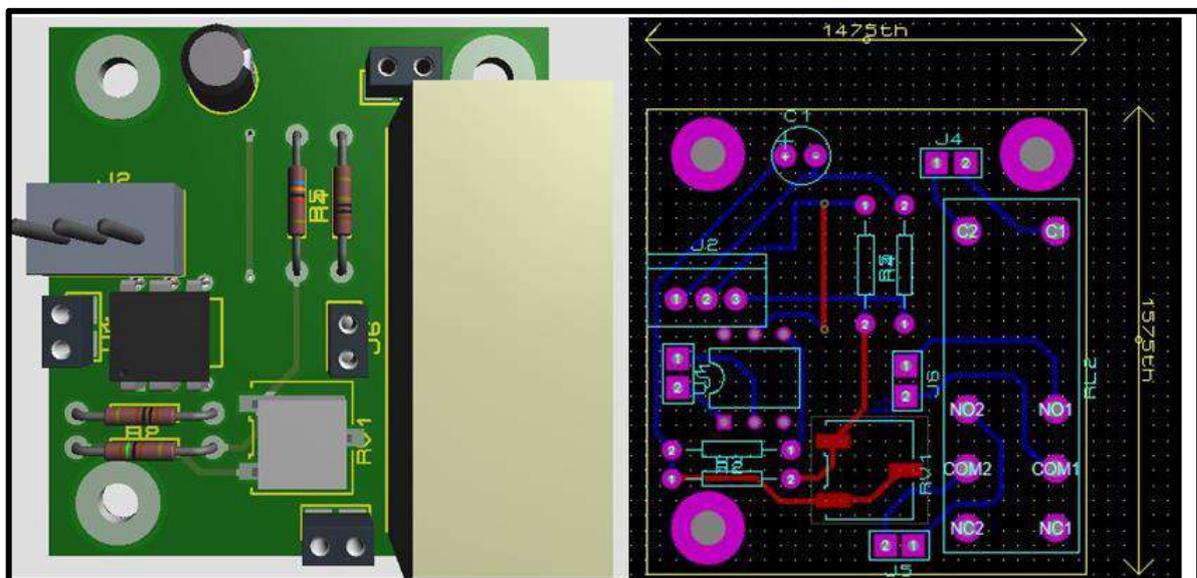


Figura 4. 32 - Vista superior de la placa para control de temperatura (izq.); Layout de pistas y puntos de conexión de placa para control de temperatura (der.).

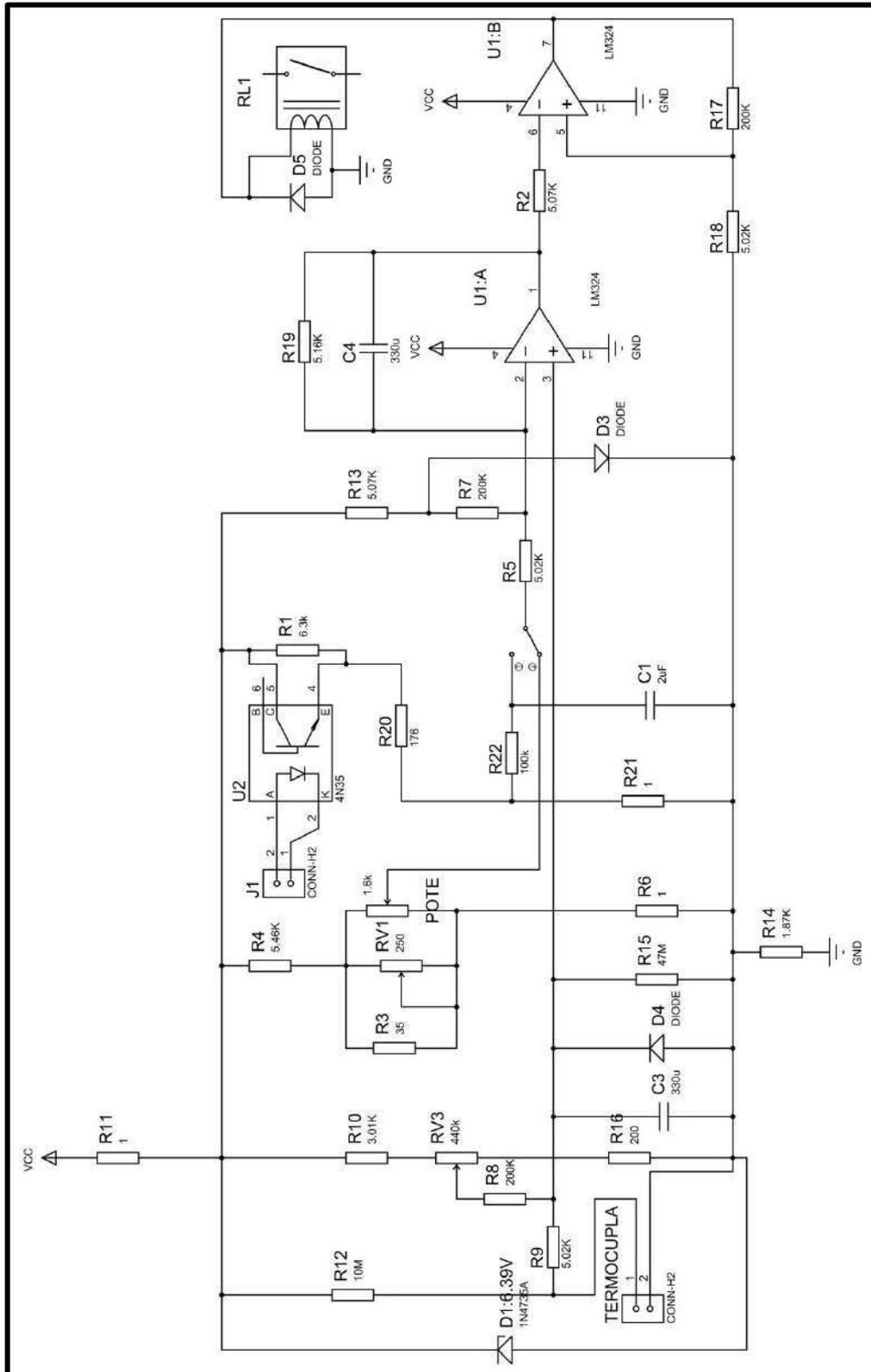


Figura 4. 33 - Esquemático de circuito para control de temperatura automática.

Se procedió de la misma manera que con la placa de aislación y distribución, y finalmente quedó de la siguiente manera (ver figura 3.34):



Figura 4. 34 - Placa para control de temperatura automático, montada sobre gabinete de placa calefactora

Una vez que teníamos la placa lista, debíamos realizar un ensayo de la misma y relevar una curva de tensión de referencia o nivel digital de PWM versus temperatura de la placa. De esta manera utilizar una ecuación que relacione ambos parámetros y de este modo permitir que cuando el usuario ingrese el valor de temperatura deseado en grados centígrados, obtenga los mismos.

La salida PWM o a veces mencionada como analógica del microcontrolador Arduino es regulable enviando valores enteros entre 0 y 255. Por lo tanto el rango de regulación de los 32 [mV] será entre esos 256 valores. De esta manera, por cada valor digital que podemos configurar desde nuestro Arduino estaremos manejando una resolución de tensión de 0,125 [mV], equivalente a 125 [μV].

$$\xi_{tensión} = \frac{32[mV]}{256} = 0,125 [mV] = 125 [\mu V]$$

$$\xi_{temperatura} = \frac{125[\mu V]}{36 [\frac{\mu V}{^{\circ}C}]} = 3,47 [^{\circ}C]$$

Siendo: $\xi_{tensión} = \text{resolución de regulación de tensión.}$

$\xi_{temperatura} = \text{resolución de regulación de temperatura}$

Con lo dicho anteriormente, logramos una resolución teórica en la regulación de 3,47 [°C]. Consideramos que es teórica dado a varios factores, entre los cuales se presentan, la inercia térmica del conjunto, la precisión de la termocupla, las

variaciones de tensión o interferencia que se puede presentar en el circuito debido a que es difícil filtrar tensiones tan bajas.

3.3.2.5 – Conexión de todas las partes

Para que todo el sistema funcione, es necesario que sus partes estén interconectadas.

En nuestro caso tendremos un cable USB para conectar la PC al RTU, el cual preferentemente debe ser mallado para evitar posibles interferencias y de este modo afectar a la comunicación.

A partir del controlador, ya disponemos de un cable de datos por cada terminal o pin del Arduino que será utilizado por nuestro sistema, adicionando uno de +5 [V] y el GND para utilizar en la placa distribuidora como ya se ha explicado anteriormente.

Los cables que se utilizan para conectar los motores son mallados debido a que estos son una fuente de ruido para el resto del sistema, y con ello afectarían la lectura de los diferentes sensores. Además, para evitar falsas lecturas de los sensores, hemos conectado capacitores de 10 [nF] entre terminales de cada final de carrera y sensor diferencial de presión. Este capacitor junto con la resistencia de pull-up interna del Arduino, forman un filtro pasa-bajos de modo que se filtrarían los rebotes y cualquier interferencia que pueda ingresar en los cables por efecto de algún campo electromagnético externo.

En la figura 3.35 se muestra un esquema simple de cómo ha sido realizado el cableado entre las diferentes placas y fichas de conexión dentro del gabinete de control, y en la figura 3.36 un esquema de borneras para identificar el origen del dato que se transmite por cada cable. Luego, en la figura 3.37 se muestra una fotografía del resultado obtenido.

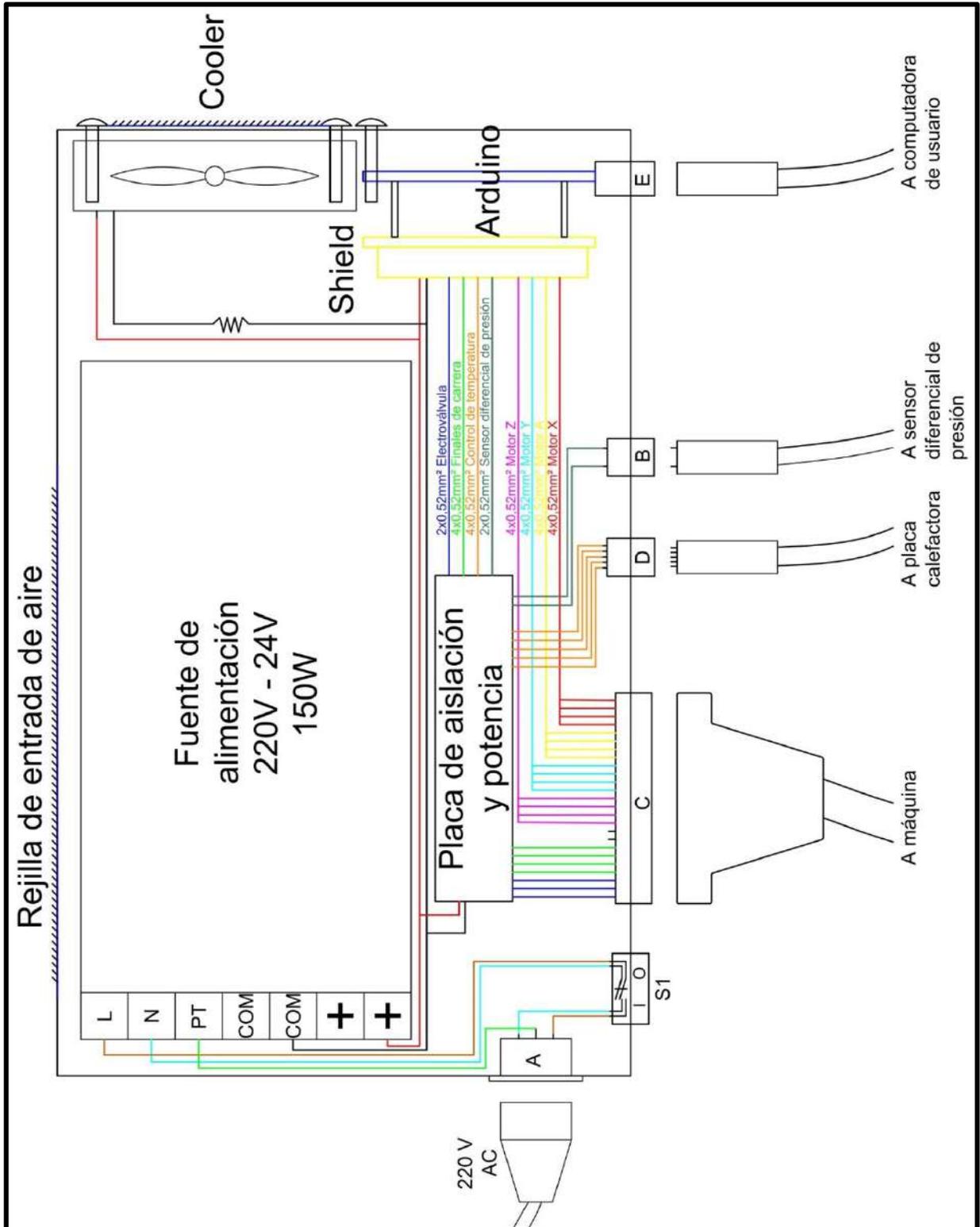


Figura 4. 33 - Esquema de posición y conexionado simplificado

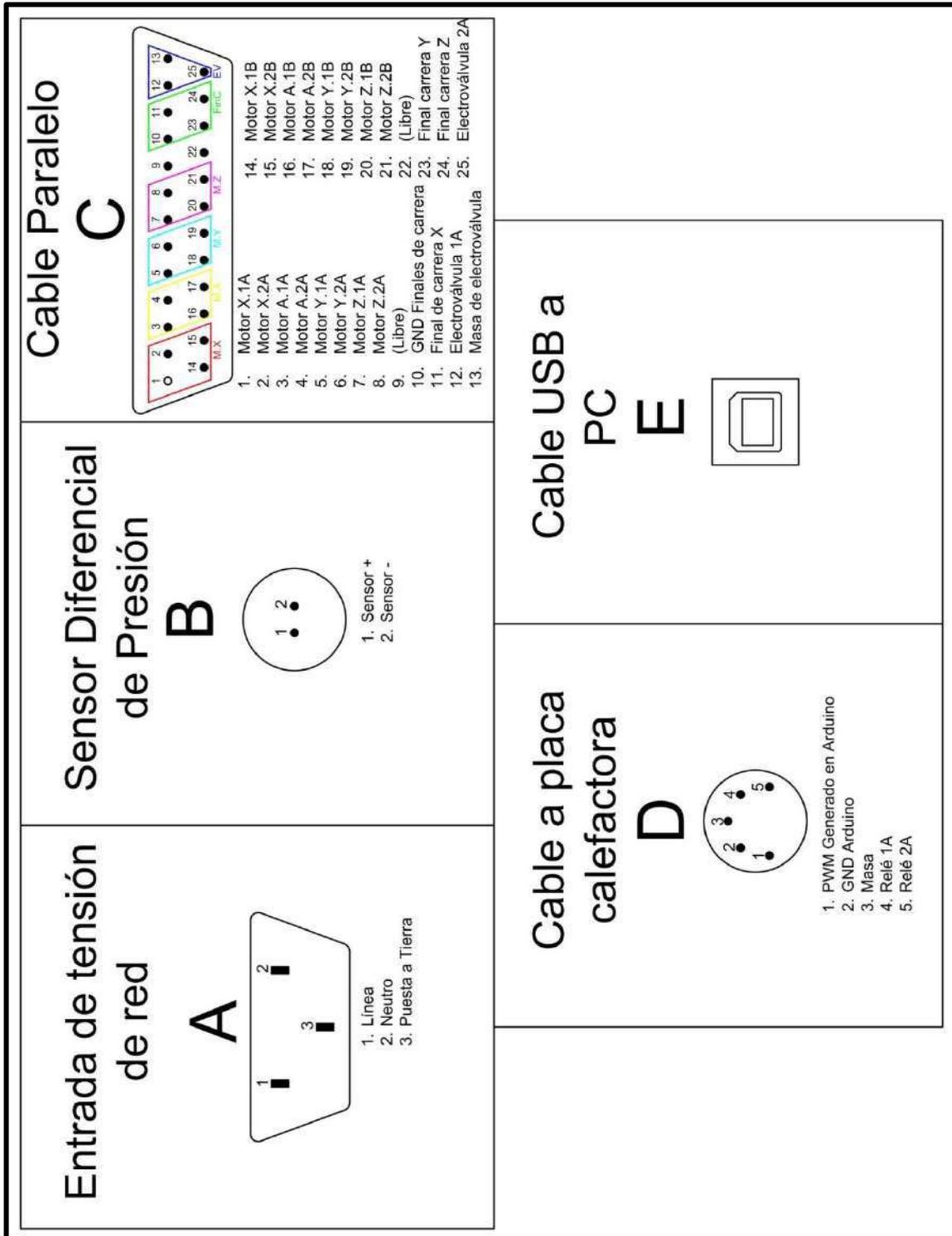


Figura 4. 346 - Esquema de borneras

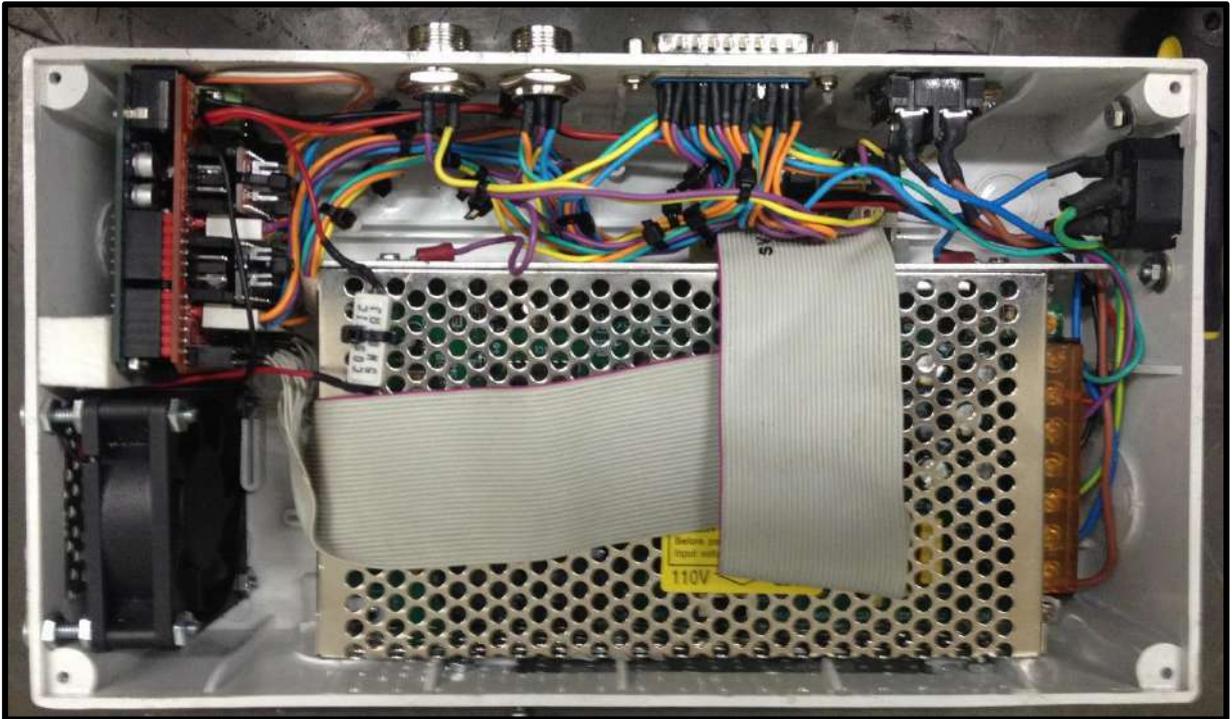


Figura 4. 37 – Fotografía de vista superior del interior del gabinete RTU

CAPITULO V SOFTWARE PARA MONITOREO Y CONTROL

En el capítulo anterior se explicaron cada una de las partes que componen al hardware de nuestro sistema SCADA. Durante el presente capítulo, se explicará el funcionamiento y diseño de los programas de nuestro proyecto, por un lado el que posee el RTU o unidad remota, y por otro lado el de nuestro maestro o MTU coincidente con el HMI.

Se comienza explicando cómo se realiza la comunicación entre ambos dispositivos, las órdenes o comandos que serán intercambiados, y finalmente un diagrama de flujo y descripción de los programas que ejecuta cada uno de ellos.

5.1 – COMUNICACIÓN

La comunicación entre el RTU y el MTU se realiza a través de un cable USB desde la PC al controlador. El RTU es configurado como un dispositivo serial, por lo cual utiliza un protocolo de comunicación que es estándar para cualquier PC. El Arduino del RTU será reconocido como tal y la PC le asignará un puerto Serial cuyo nombre dependerá del sistema operativo, para Windows será COM1, COM2, etc.

En la comunicación serial se envían y reciben bytes de información, generalmente en código ASCII, de a un bit a la vez (serie). Este tipo de comunicación es más lento que la comunicación paralela en la cual se pueden enviar de un byte (8bits) por vez, pero es más sencillo y permite grandes distancias de transmisión. Las características que configuran una comunicación se enumeran a continuación junto con los valores asignados en nuestro caso:

- Tasa de baudios (bits por segundo): 9600.
- Bits de datos (cantidad de bits): variable, hasta que se interrumpa la llegada de bits. Esto es fácil de configurar para cada dispositivo y es indispensable ya que las cadenas de texto son variables y por lo tanto los bits necesarios a transmitir.
- Bits de terminación (al finalizar la comunicación): Sí se utiliza.
- Bits de paridad (para detectar errores en la comunicación): No se utiliza.

Estos parámetros deben ser igual en todos los dispositivos que intenten comunicarse para que la transmisión de datos sea exitosa.

5.2 – ÓRDENES DE INTERCAMBIO ENTRE EL RTU Y MTU:

El Arduino ha sido programado para leer el puerto serial hasta recibir ciertas cadenas de texto a través del protocolo serial. En el programa esto se ha resuelto utilizando una estructura de casos y las variables para cada uno de ellos.

A continuación se enumera cada uno de estos casos que son los que envía el Labview (el cual será explicado más avanzado el presente capítulo), cuando el mismo determina que la máquina debe realizar cierta acción y el Arduino debe controlar la ejecución de la misma.

5.2.1 – Control de depresión en la campana de seguridad

Como hemos mencionado anteriormente, nuestro equipo será utilizado para realizar procesos químicos con sustancias peligrosas, es por ello que el mismo funciona dentro de una campana de seguridad instalada en unos de los laboratorios de nuestra facultad.

Nuestro sistema ha sido creado para realizar recubrimientos superficiales mediante dos procesos diferentes. Por un lado, depósito por spray, en el cual ingresa un importante volumen de gases y vapores peligrosos. Y por otro lado el denominado “Dipping”, que surge de la inmersión de la muestra en la sustancia a velocidades controladas. En este segundo proceso, la sustancia que puede ser peligrosa se encuentra en estado líquido, y más allá de que pueda generar vapores, su concentración estimamos que puede no ser peligrosa debido a que son utilizadas pequeños volúmenes de líquido.

Por lo mencionado en el párrafo anterior, hemos decidido apartar la necesidad de tener el extractor encendido entre un proceso y otro. De este modo es obligatorio para el proceso de spray (no modificable por el usuario), y desinteresado en el caso de dipping. De esta manera, se envía al comienzo del proceso un comando al RTU para indicar que sí debe ser tenido en cuenta el valor del SDP o no, dependiendo de la aplicación. Esto ha sido realizado enviando una cadena de dos caracteres “P0” o “P1”, siendo “P” el indicador de configuración de presión, y luego el número “0” para

indicar que debe ser tenido en cuenta el valor de SDP (Spray), y un “1” en caso de que no debe ser tenido en cuenta (dipping).

5.2.2 – Control de placa de temperatura

Cuando el Arduino recibe una cadena de texto de la siguiente manera: “S[temperatura]”, donde la letra S es un carácter enviado, y la palabra entre corchetes es el nombre de la variable que se transmite. En este caso, al recibir la letra S controlador ejecuta el código correspondiente a éste carácter.

Comienza por encender la placa calefactora a partir de la salida digital que activará el relé ubicado en el controlador de temperatura. Y luego, con el valor de la variable de temperatura el mismo lo transfiere a una salida PWM para que sea adoptado en la placa calefactora y el mismo alcance la temperatura configurada.

Ambas salidas permanecen en ejecución hasta que se le indique que detengan la placa. Una vez activadas, el controlador vuelve a leer el puerto serial esperando una orden.

5.2.3 – Configuración de velocidad y aceleración

Este parámetro no hace que el Arduino ejecute alguna acción en específico, pero sirve para que el mismo lea cual es la configuración de velocidad y aceleración que el usuario ingreso al HMI y de ésta manera cambiar esos parámetros para cuando se ejecute la orden de accionar a los motores.

En este caso el Arduino puede recibir cualquiera de estas cadenas o todas ellas dependiendo de la acción a ejecutar:

- Eje Longitudinal: “U[Velocidad X],[Aceleración X]”
- Eje transversal: “V[Velocidad Y],[Aceleración Y]”
- Eje vertical: “W[Velocidad Z],[Aceleración Z]”.

Cabe destacar que el usuario del HMI no tiene acceso a variar la aceleración, pero que de todos modos el intercambio se realiza ya que es una constante oculta dentro del software del Labview para poder realizar una puesta a punto y configurar cual es la correcta.

El valor de dicha constante es de 6000 [pasos/seg²] para el eje longitudinal y 4000 [pasos/seg²] para los ejes transversal y vertical con transmisión a tornillo y por lo tanto velocidades menores, basándonos en la bibliografía³ al pie de esta página.

De esta manera nosotros partimos de la aceleración que se requiere para la máxima velocidad en cada eje y luego se verificó que no afectaba a velocidades menores. La aceleración y desaceleración es ejecutada por la librería AccelStepper que utiliza nuestro programa.

Para los valores de velocidad no hay límite inferior ya que el RTU es capaz de controlar velocidades muy bajas para los motores. Pero si ha sido necesario configurar una velocidad máxima, por un lado partiendo de que para cualquier controlador de 16 [Mhz] el límite de pasos por segundo que podría enviar es de 4000, y por otro lado, el torque de los motores paso a paso tiene una tendencia a caer a grandes velocidades como puede verse en el anexo F.

5.2.4 – Orden de movimiento

El MTU es el encargado de coordinar los movimientos que debe ejecutar la máquina, pero el RTU el encargado de controlarlo y dar los impulsos eléctricos para que se produzca el movimiento.

Para ello hemos creado el comando M, el cual es un subprograma debido al código extenso que requiere. La cadena de caracteres que recibe el controlador en este caso es de la siguiente forma: “M[Posición X], [Posición Y], [Posición Z],[PosiciónON],[PosiciónOFF]”. Recordemos que dichas posiciones son absolutas, y es la librería AccelStepper la que lleva internamente los contadores de pasos de manera tal que siempre reconoce la posición absoluta en la que se encuentra y envía solo los pasos que hagan falta para llegar a la nueva absoluta que le enviamos desde el MTU. El movimiento se realiza a las velocidades que ha configurado el usuario en el HMI, manejando rampas de aceleración como se ha mencionado en la unidad anterior a esta.

Las posiciones de ON y OFF de la electroválvula deben estar entre un mínimo de cero, es decir, en el instante previo a comenzar el movimiento ya energiza el

³ Generate stepper-motor speed profiles in real time – David Austin
http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf

solenoides de la electroválvula, y como máximo la coordenada límite de movimiento en el eje longitudinal, la cual es de 45 [cm]. En el eje transversal la misma es de 25 [cm], mientras que para el eje vertical la distancia máxima que puede ser recorrida es de 14 [cm].

Tras enviar el comando M, el MTU esperará una respuesta del RTU al finalizar con el movimiento de la máquina, es decir, se finalizó de enviar pasos a los motores. El controlador se encuentra constantemente relevando el valor de los sensores que tiene conectados, es decir, de los finales de carrera, y en el comienzo a su vez el de presión diferencial en la campana. En caso de que alguno de ellos este activo o se active durante el movimiento (sea porque la extracción de vapores en la campana no es la adecuada o porque se ha llegado a un final de carrera y la máquina se detuvo), el Arduino enviará una respuesta al MTU que será:

- “R”: Si el movimiento finalizó correctamente, es decir, sin que se hayan activado los finales de carrera o el sensor diferencial de presión.
- “A”: Si algún final de carrera ha sido activado por alcanzarse los límites aceptables de movimiento.
- “P”: Si el sensor diferencial de presión se activó por falta de depresión en la campana.
- “E”: Si detecta que se ha presionado el botón de parada de emergencia y la máquina se reinicia.

El software del MTU es el encargado de procesar dicha información y actuar según corresponda, la manera en que resolvimos esta temática se explicará en la sección del software del MTU.

5.2.5 – Orden de ir a posición de reposo

Ante cualquier eventualidad es posible que la máquina quede en una posición errónea (como puede ser un corte en la alimentación eléctrica, pérdida de pasos por sobrecarga de los ejes, etc.). Como medida para solucionar este inconveniente, hemos creado este comando denominado H (éste es el carácter enviado al controlador y también subprograma que controla la acción), que se ejecuta automáticamente al conectarse el controlador y alimentarse la máquina, así como también por orden del usuario desde el HMI.

Esto es debido a que no tenemos realimentación de la posición en la que se encuentra la máquina para cada eje. Mediante el comando H, el controlador envía a la máquina hasta los finales de carrera más cercanos a la posición de reposo o “Home” como la hemos denominado nosotros. Una vez alcanzado los finales de carrera, el controlador se asegura de un punto de referencia real y luego realiza un pequeño offset en cada eje. En ese punto es donde marca la posición de cero para cada eje, desde el cual tomará las coordenadas absolutas que le serán enviadas.

Para evitar que la máquina alcance el límite de los finales de carrera, hemos optado por tomar un offset de 500 pasos como medida de seguridad, y que la máquina no sobrepase a los mismos cuando se lo envía a la posición de 0 absoluto en cualquiera de los ejes. Una vez alcanzada esta posición de offset y configurado como cero absoluto para los ejes, el controlador espera a recibir las órdenes del MTU.

En la figura 5.1 se observa la posición de Home que nosotros hemos definido, sobre un esquema simplificado de la máquina.

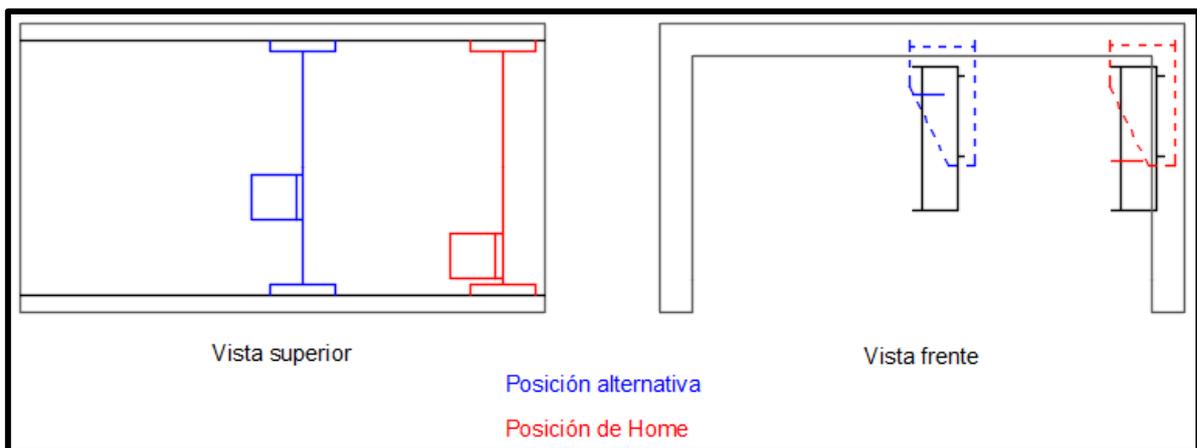


Figura 5. 1 - Vistas de posición alternativa y posición de "Home"

5.3 – SOFTWARE DEL CONTROLADOR (RTU)

El controlador Arduino debe ser programado para poder adquirir datos de los sensores y para controlar los actuadores. Para ello hemos utilizado el entorno de desarrollo del mismo fabricante, el Arduino IDE, cuyas funciones básicas pueden encontrarse entre la bibliografía utilizada. En esta sección nos limitaremos a explicar

de qué manera se han realizado las instrucciones y algoritmos de comunicación, adquisición y control.

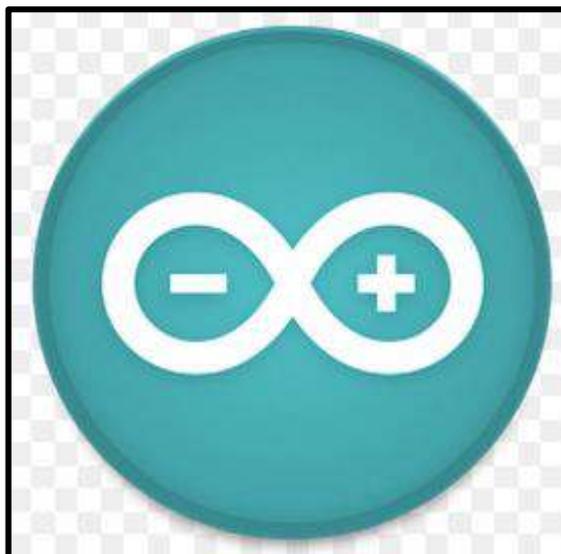


Figura 5. 2 - Logo de Arduino IDE

Nuestro programa comienza por la etapa de configuración de pines, la cual ha sido ya presentada en la sección del controlador Arduino. Para ello se han nombrado los pines respectivos a cada uno de los motores, estos se configuran según corresponda con la librería AccelStepper utilizada para facilitar el control de los motores, tanto en dirección como en pasos o pulsos para lograrlos, pines para controlar la temperatura y encender la placa, etc.

5.3.1 – Declaración de variables

El programa utiliza variables internas para el procesamiento de datos, contadores, cálculos auxiliares, etc., los cuales no serán mostrados en esta sección pero se podrán ver en el código y su respectiva descripción.

Por otro lado, ha sido necesario crear variables para controlar los actuadores o salidas y a su vez para poder intercambiar dichos valores con la PC y así permitir al usuario cambiar los parámetros de la máquina tales como velocidades, posiciones, etc.

Estas variables son las que se enumeran y describen en la tabla 4.1:

Tabla 5.1 – Variables de intercambio entre Arduino y Labview

Variables	Tipo	Unidad	Descripción
Posición de destino X	Word	Pasos	Se envía posición absoluta en pasos a la cual debe dirigirse cada motor según el eje que corresponda.
Posición de destino Y	Word	Pasos	
Posición de destino Z	Word	Pasos	
Velocidad X	Word	Pasos/seg	Se envía un valor numérico de pasos por segundo que debe ejecutar el Arduino a los motores para regular la velocidad.
Velocidad Y	Word	Pasos/seg	
Velocidad Z	Word	Pasos/seg	
Aceleración X	Word	pasos/seg ²	Se envía un valor numérico de aceleración y desaceleración.
Aceleración Y	Word	pasos/seg ²	
Aceleración Z	Word	pasos/seg ²	
PosiciónON	Word	pasos	Se envía un valor numérico de pasos en posición absoluta en los cuales debe activarse la electroválvula.
PosiciónOFF	Word	pasos	Se envía un valor numérico de pasos en posición absoluta en los cuales debe desactivarse la electroválvula.
Estado de placa calefactora	Booleana	-	Se envía cuando debe encenderse la placa calefactora.
Temperatura	Word	[°C]	Se envía un valor numérico de la temperatura en grados centígrados a la cual se debe elevar la placa.
Presión	Booleana	-	Determina si es necesario tener en cuenta el valor del SDP.

5.3.2 – Diagrama de flujo

Nuestro controlador es el encargado de activar y desactivar el valor de las salidas, o de manejar la tensión de referencia en el caso de la salida PWM, etc. Pero las

decisiones no las toma el mismo sino que provienen del MTU, es decir, de la PC manejada por el usuario y el software de Labview. Es por ello que debemos mantener una comunicación entre ambos dispositivos.

A diferencia del software del MTU que se explicará más adelante, nuestro RTU posee un único código a ejecutar. En la figura 5.3 se muestra el diagrama de flujo de este programa, en el cual cada recuadro es una secuencia de instrucciones.

INICIAR: Declaración de variables, configuración de terminales y estado inicial de los mismos. Luego entra en un ciclo “While” que espera acciones mientras el puerto serial se encuentre disponible.

PRESIÓN: Se recibe el carácter “P” seguido de otro que puede ser 0 o 1 tal como se mencionó en la sección 5.2.1. Consiste en instrucciones que analizan el estado del SDP, lo envían al MTU y lo tienen en cuenta en el resto del proceso dependiendo si es requerido o no.

TEMPERATURA: Se recibe la cadena “S[temperatura]”. Inicia una secuencia de instrucciones para activar o desactivar el relé de alimentación de la placa calefactora y además, configura la salida PWM para la regulación de temperatura.

HOMING: Se ejecuta el subprograma de Homing, en el cual se envía a la máquina a dicha posición ya antes explicada, y luego configura a la misma como el cero absoluto de los tres ejes.

COMANDO “M”: Se recibe la cadena de caracteres ya mencionada: “M[Posición X], [Posición Y], [Posición Z],[PosiciónON],[PosiciónOFF]”. Y en base a la librería para Arduino ejecuta las instrucciones para realizar el movimiento en todos los ejes dependiendo de los pasos (en posición absoluta) enviados en dicho comando. Durante el mismo se analiza el valor de los sensores de finales de carrera, presión y parada de emergencia para detener el proceso y dar aviso al MTU.

VALVULA: Desde el MTU se reciben las coordenadas límites entre las cuales se debe encender o no la electroválvula para permitir el pasaje de aire o fluido de transporte para realizar el spray. Pero de todos modos hemos creado una secuencia de instrucciones tal que al recibir el carácter “T” desde el puerto serial, conmutar el estado de la electroválvula. Esta sección fue añadida para verificar, depurar y en caso de ampliación del MTU.

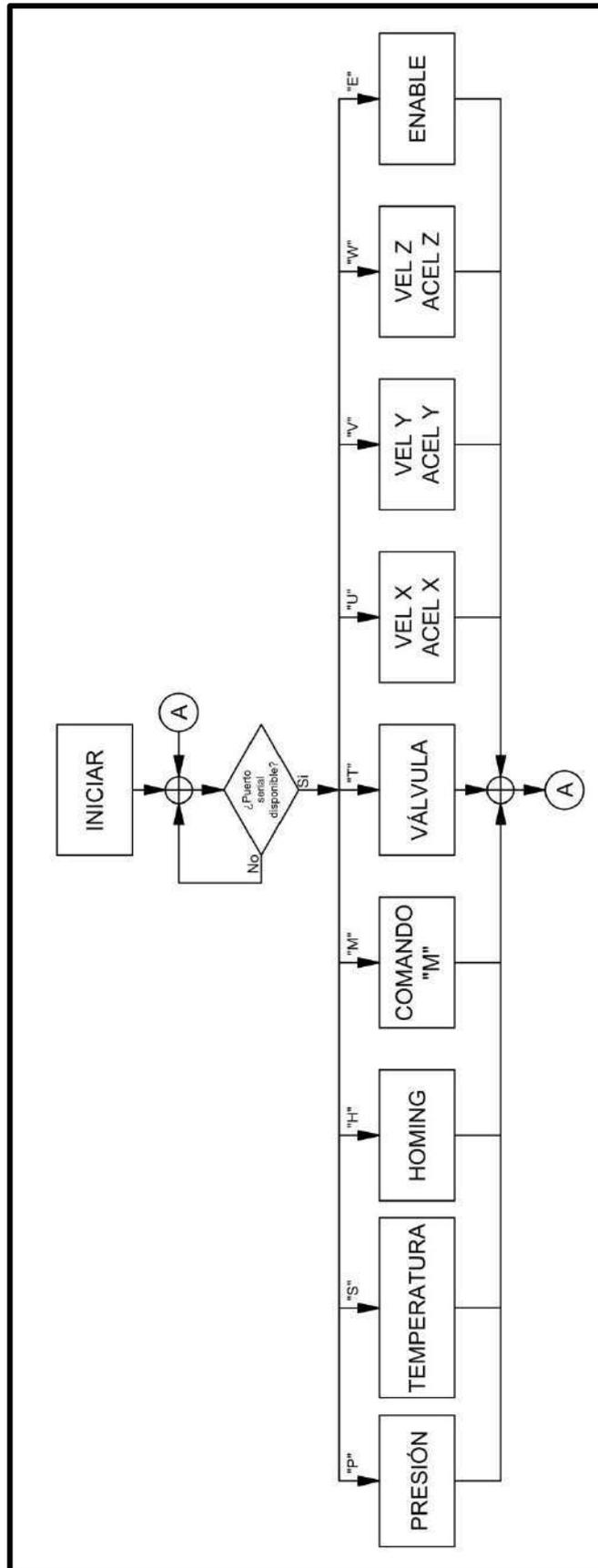


Figura 5. 3 - Diagrama de flujo de software de RTU

VEL X, ACEL X: El RTU recibe la configuración de velocidad y aceleración para el eje X a través de la siguiente cadena de caracteres “U[Velocidad X], [Aceleración X]” tal como se mencionó antes. Lo mismo con los otros ejes para las secuencias “V[.], [.]” y “W[.], [.].”

ENABLE: Al recibir el carácter “E” por puerto serial, el programa conmuta el estado del pin digital configurado como “enable” o salida de habilitación. La misma permite habilitar o deshabilitar los drivers de los motores PAP y de este modo interrumpir el paso de corriente a los mismos.

5.4 – UNIDAD MAESTRA E INTERFAZ HOMBRE-MÁQUINA (MTU/HMI)

Hemos ya mencionado en este informe que tanto la unidad maestra como la interfaz humano-máquina de nuestro sistema de control funcionan en la PC del usuario. Es por esto que han sido incluidos en el mismo software desarrollado en el Labview, cuyas funciones se enumeran a continuación:

- Presentar un panel de control amigable para el usuario que funcione como HMI.
- Coordinar las acciones del RTU y procesar información:
 - Manejar orden de secuencia.
 - Temporizaciones.
 - Manejo de errores.
 - Análisis de datos y valores numéricos.
 - Configuración de la comunicación.

El software Labview creado por National Instruments, es ideal para sistemas de medidas y control al integrar todas las herramientas que los ingenieros y científicos necesitan para construir una amplia variedad de aplicaciones en mucho menos tiempo. Es un entorno de desarrollo para resolver problemas, productividad acelerada y constante innovación.

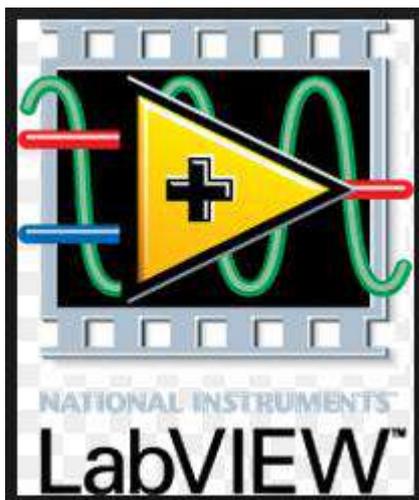


Figura 5. 4 - Logo del entorno Labview

Es una plataforma de programación gráfica que ayuda a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Es posible crear programas en cortos períodos de tiempo con infinidad de herramientas para abordar cualquier tema de la ingeniería, y permite ingresar códigos escritos en C para ganar flexibilidad a aquellos que dudan de la programación gráfica y prefieren el comando por líneas de instrucciones.

Existen diversas arquitecturas de programa, ya sea para Labview como cualquier otro entorno de programación. Analizando algunas de ellas, debimos optar por una que nos permita ejecutar las órdenes en una secuencia ordenada, clara, y que a la vez permita interrupciones en caso de que el usuario quiera detener el proceso.

Investigando sobre estas arquitecturas, determinamos que dos de ellas serían las más adecuadas para llevar a cabo nuestro programa. Las mismas se enumeran a continuación:

- “State Machine” o máquina de estados:⁴

Esta arquitectura facilita la definición de la secuencia de ejecución para las secciones de código. Esta implementación particular generalmente que se conoce como máquina Moore, la cual determina el siguiente estado en base a decisiones tomadas en el estado actual. El diseño de esta plantilla hace más fácil insertar nuevas secciones de código, eliminar secciones de código o cambiar el orden en el

⁴ Fuente: <http://www.ni.com/tutorial/14120/es/>

cual las secciones se ejecutan, todo sin hacer grandes modificaciones a la estructura de la aplicación.

- “Queued Message Handler” (QMH) o Controlador de mensajes en fila:⁵

Esta arquitectura facilita que múltiples secciones de código se ejecuten en paralelo y se envíen datos entre ellos. Cada sección de código representa un proceso, como adquirir datos y está diseñada de manera similar a una máquina de estado. La separación de estos componentes permite una interfaz de usuario eficiente y la habilidad de registrar datos continuamente mientras que otros mensajes se están ejecutando.

Tras analizar las ventajas y desventajas de cada una de ellas y comparándolas con nuestras necesidades, hemos determinado que la QMH es una arquitectura muy compleja y que en nuestra aplicación, creando estados necesarios para sincronizar la lectura y escritura por serial, podíamos cubrir correctamente nuestras necesidades evitando la programación tan compleja y que consume más recursos de nuestro procesador.

En cuanto a la creación de un panel frontal, el mismo no se ve afectado por el tipo de arquitectura o el manejo del código, ya que solo permite indicar cambios de variables y manejar los botones y controladores numéricos por parte del usuario.

5.4.1 – Software de MTU como máquina de estados

Una máquina de estados se define como un conjunto de estados en los cuales se ejecuta un código, se analizan variables, y para cada instante la salida depende únicamente del estado en el que se encuentre y de las entradas actuales.

Las transiciones de un estado a otro se generan en respuesta a eventos de entrada externos e internos; a su vez estas transiciones pueden generar otros eventos de salida. Esta dependencia de acciones del sistema a los eventos de entrada, hace que las máquinas de estado sean una herramienta adecuada para el diseño de sistemas reactivos y la programación conducida por eventos, tal como la que requiere nuestra máquina.

⁵ Fuente: <http://www.ni.com/tutorial/14119/es/>

5.4.2 – Software para depositadora por Spray

La aplicación principal de la máquina es la de realizar barridos y de esta manera depositar recubrimientos superficiales sobre muestras o probetas de vidrio y otros materiales. Es por ello que el panel frontal de nuestra aplicación, mostrada en la figura 5.5, es para llevar a cabo tal fin. Por otro lado, se nos pidió que la máquina sea capaz de realizar el proceso de “Dipping” que será explicado en la sección 5.4.3. Para ello realizamos una segunda aplicación o panel frontal, la cual puede ser abierta desde un botón ubicado en la principal, el cual se ve en la imagen del panel frontal.

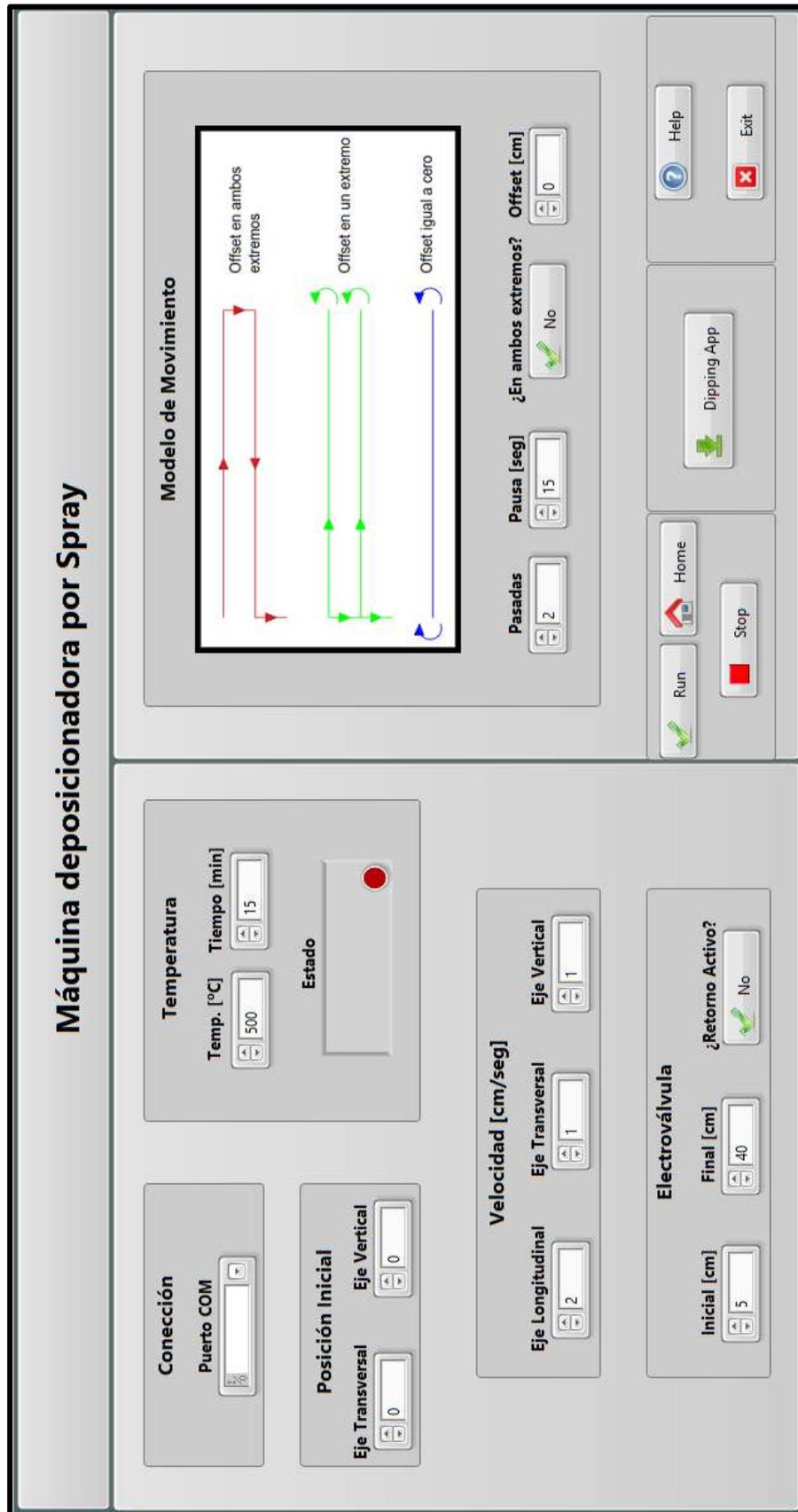


Figura 5. 5 - Panel frontal de la aplicación para depositar por spray

En la figura 5.7 se muestra el diagrama de flujo de nuestro programa para aplicar recubrimientos superficiales por spray, es decir, como se va dando la transición de un estado al siguiente dependiendo de la decisión que se deba tomar en caso de que corresponda, y a continuación se hará una breve descripción de cada estado. En algunos casos simplemente se cumple con una secuencia o grupo de instrucciones y luego se pasa siempre al mismo estado siguiente, lo cual es diseñado de este modo para asegurarse un cierto orden en la secuencia.

Debemos aclarar que la cantidad de estados que puede tener un programa puede ser muy diferente y sin embargo alcanzar los mismos resultados. En nuestro caso comenzamos con varios programas en los cuales la cantidad de estados era menor, pero cada vez que debíamos depurar un error había que modificar una gran cantidad de código dentro de cada estado. Al crear un estado por cada tarea o por pequeñas tareas, se logra mayor independencia en cada uno de ellos, lo que brinda flexibilidad a la hora de modificar el programa y depurar los errores que se presentan.

En la siguiente enumeración se pretende explicar que es lo que ocurre en cada uno de los estados a modo de comprender como se resolvió el control del proceso, para mayor detalle sobre las referencias a funciones del programa, ver anexo B.

INICIAR: En este estado se configura la comunicación entre el MTU y el RTU, y además se configuran algunos valores iniciales de algunos datos al “Shift register” o registro de desplazamiento. La configuración debe ser configurada según la figura 5.6 previo a presionar el botón de que comience el programa.

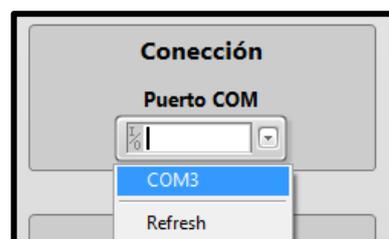


Figura 5. 6 - Configuración de puerto de comunicación entre MTU y RTU

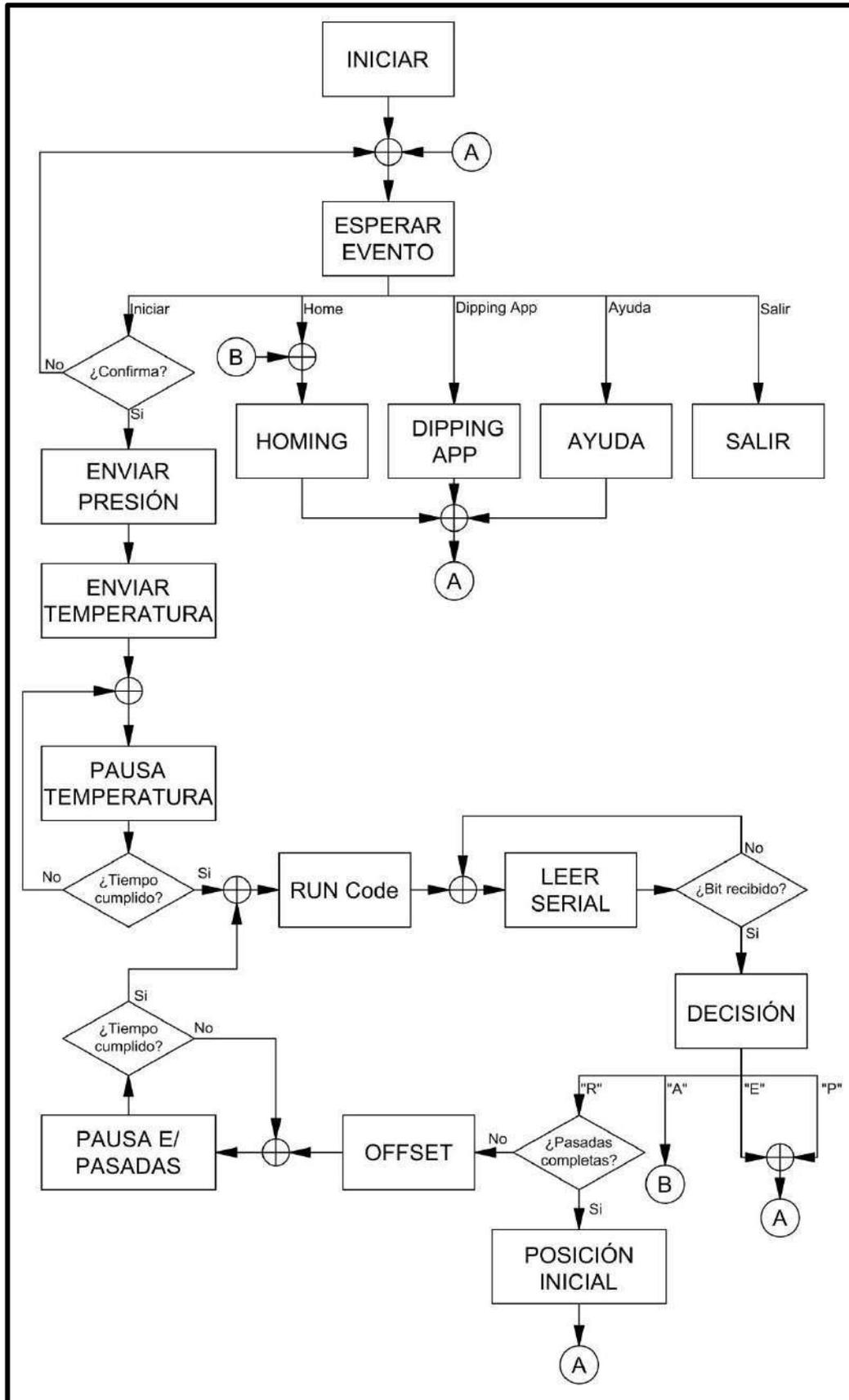
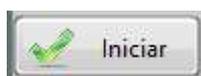


Figura 5. 7 - Diagrama de flujo de software para depositar por spray

ESPERAR EVENTO: este estado consiste en la espera de que algún botón de acción de los que puede encontrarse en el panel frontal sea oprimido (Iniciar, Home, Dipping App, Salir). Durante el mismo se pueden ir cargando los valores numéricos de las variables y de ahí a los registros de desplazamiento que corresponde. Para actualizar cada vez que el usuario modifique estos, se ha creado un timeout de 0,5 [seg] en el cual se reingresa al mismo estado. Si se presiona cada uno de los botones de función ocurrirá lo que se menciona a continuación:



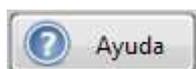
Primero carga valores de control numérico al registro de desplazamiento y luego pide confirmación para continuar. En caso de ser aceptada, el siguiente estado será el de comenzar con la ejecución del proceso, enviando la configuración de presión y la información de temperatura a la placa. En caso de que se rechace la confirmación, se vuelve al estado de esperar evento.



Este botón dirige directamente al estado denominado “Homing⁶”.

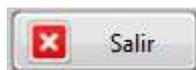


Al apretar este botón, dentro del mismo estado se ejecuta otro panel frontal conformado con otro programa (VI). Esperará en el mismo a que el otro VI finalice, el estado siguiente será esperar evento nuevamente.

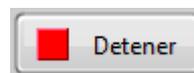


Se abrirá un archivo ayuda que consiste en un manual de uso de la aplicación, el cual se encuentra en el anexo C.4. En el mismo se pretenden explicar los pasos básicos para implementar un proceso de depósito por spray y Dipping.

⁶ Terminología en inglés para determinar el proceso de buscar la localización de un objeto o auto localizarse.



Este botón envía a un estado de Exit donde se pide confirmación para salir y en caso de ser aceptada cerrar la aplicación. En caso de rechazar la confirmación se vuelve al estado de esperar evento.



NOTA: Se observa en el panel frontal un botón  que puede ser oprimido y será reconocido en cualquier momento de la ejecución del programa, no como sucede con los explicados anteriormente solo serán reconocidos si han sido presionados o no cuando el programa esté inactivo esperando por una orden por parte del programa. Al presionarse dicho botón se detendrá el proceso del MTU, y por lo tanto no enviará más ordenes al RTU, pero el programa de este último no finalizará, ya que el mismo no admite un reseteo mediante serial o una interrupción de sus ciclos. Por lo tanto, si se desea detener la máquina durante su movimiento, se debe recurrir al botón manual de parada de emergencia que se encuentra en la campana de seguridad y que interrumpe la alimentación de la máquina.

ENVIAR PRESIÓN: Este estado es el encargado de enviar la configuración de presión para este proceso.

ENVIAR TEMPERATURA: A este estado se llega una vez confirmado el inicio del proceso. Durante el mismo se envía la cadena de caracteres “S[temperatura]” que configura la temperatura, tal como fue explicado anteriormente. Luego de enviar dicha cadena a través de serial hacia el RTU, el programa se dirige al siguiente estado denominado Pausa temperatura.

PAUSA TEMPERATURA: Este estado permite configurar una pausa entre el envío de la configuración de temperatura, y el arranque del movimiento o resto de proceso. Según ensayos que hemos realizado el tiempo que tarda la placa calefactora en llegar a su temperatura máxima de 400 [°C] es de 15 [min].

A estado se llega automáticamente luego de enviar la cadena de texto de temperatura. Este estado funciona como un contador de periodos de tiempo, configurado en este caso cada 0,5 [seg] de modo que cada ese período de tiempo, se reingresa al estado y se continua con la suma, hasta que se haya cumplido el

tiempo de pausa establecido. Luego el programa se dirige al estado denominado Run Code.

RUN CODE: Aquí se envía la cadena de texto “M[PosiciónX],[PosiciónY],[PosiciónZ],[PosiciónON],[PosiciónOFF]”. Durante esta etapa se incrementa el valor del contador de pasadas, el cual es necesario para detectar el fin del proceso. Para la primera vez que se ejecuta este estado, para la posición del eje transversal y longitudinal, se envía las coordenadas iniciales configuradas por el usuario. Luego de enviar, como hemos mencionado, se va al estado de leer serial para obtener una respuesta por parte del RTU.

LEER SERIAL: En este estado, se pretende detectar la presencia de un bit en el buffer del puerto serial del RTU. En caso de que no se detecte el mismo, se reingresa al estado con un cierto retardo de tiempo, de este modo, en el paso de un estado (aunque sea al mismo) a otro se permite actualizar el valor del botón Detener y así detener el proceso en caso de ser necesario.

Una vez que se detecta el bit en el puerto se lo almacena y la decisión según corresponda para cada carácter se hace en otro estado.

DECISIÓN: Es el estado encargado de procesar la información recibida desde el RTU. Si por serial se lee una “R”, es que se finalizó correctamente con el comando de movimiento. Si el número de iteración actual es igual al número de pasadas configurado por el usuario, el proceso estará finalizado y por lo tanto el estado siguiente es el de “Ir a posición inicial”, en caso contrario, el proceso debe continuar y por lo tanto el estado siguiente es el “Offset”; si el carácter recibido por serial es una “A”, indica que por algún error en el movimiento se han excedido los límites de control y tras enviar un mensaje de alarma al usuario, procede a la posición de Home en el eje que se ha producido la falla; si es una “P”, el sensor diferencial de presión ha detectado que la depresión en la campana no es suficiente y por lo tanto detiene el proceso y envía el mensaje al usuario, para que después de corregir la falla vuelva a iniciar; finalmente si es una “E”, causado por la detección del botón de parada de emergencia, avisará al usuario tras otra alarma, detendrá el proceso y esperará a que el usuario envíe órdenes nuevamente.

OFFSET:

Si el comando de movimiento finalizó correctamente, el proceso continúa con un manejo del offset⁷, según haya programado el modelo de movimiento el usuario. Para configurar una ruta de barrido y en qué momentos depositar o no spray sobre las muestras, nos hemos basado en una barrida lineal del carro sobre el eje longitudinal, movimiento durante el cual se realizará el depósito de spray sobre las muestras. En esta versión de software, la altura del eje vertical estará fija durante todo el proceso, dirigiéndose a la misma al principio de la operación. En cuanto al eje transversal, se ofrece una opción de posición inicial donde comenzar, y luego una serie de opciones para determinar en qué momento o no desplazarse sobre el mismo, lo cual ocurrirá en alguno de los extremos. El menú que permite configurar a la misma se muestra en la figura 5.8.

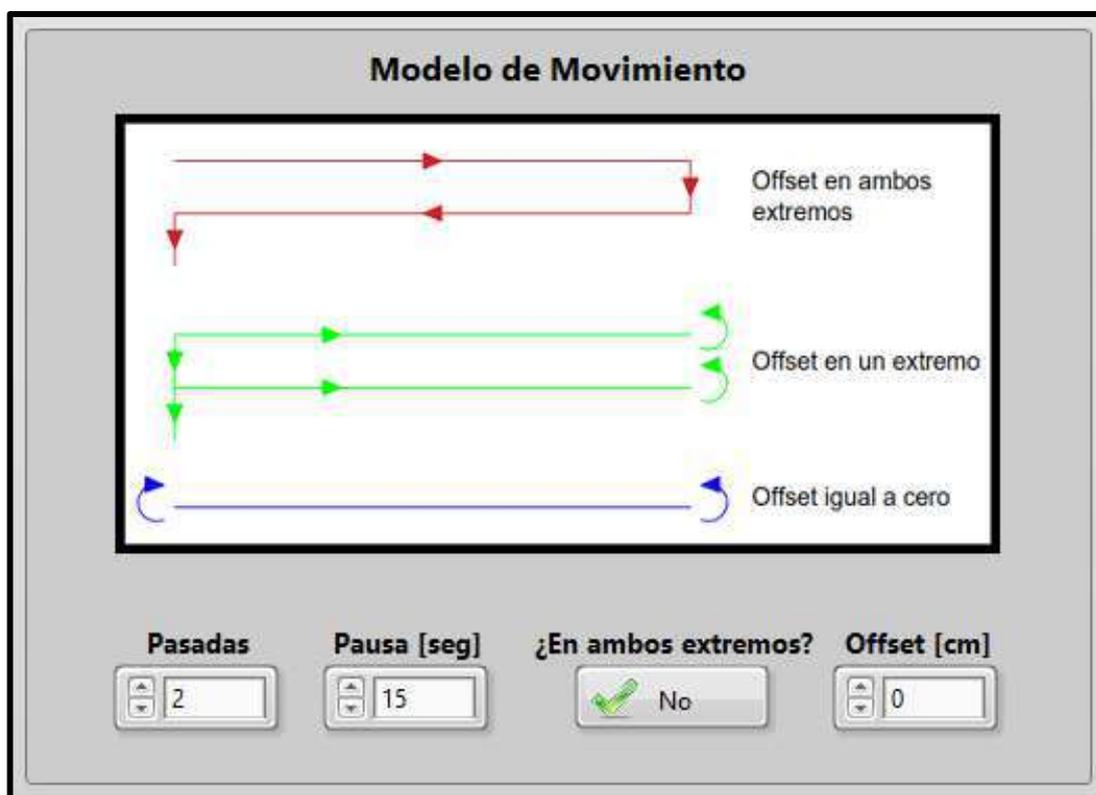


Figura 5. 8 - Menú para configurar el movimiento del pulverizador a lo largo del proceso

En dicha imagen entonces se permite que la máquina realice un espaciado transversal en ambos extremos, o solo en el extremo inicial. En caso de que la distancia de offset sea igual a cero, las distancias serán lineales en una misma posición transversal. Por otro lado se permite una pausa entre el final de una pasada y el comienzo de la otra, este parámetro fue creado para que el usuario considere un

⁷ Offset: denominación de la distancia transversal entre pasadas a lo largo del eje longitudinal.

tiempo intermedio donde la muestra pueda recuperar temperatura luego de que se le haya aplicado un recubrimiento a temperatura ambiente.

En otro menú aparte (ver figura 5.9), se le permite al usuario configurar si la electroválvula actuará en un único sentido de recorrido o en ambos (activando o desactivando el botón de Retorno Activo). Esto dependerá de que si el spray puede llegar a tener alguna orientación diferente dependiendo del ángulo en que es aplicado ya que no puede ser totalmente perpendicular (se derramaría el líquido del Erlenmeyer), y por lo tanto con esta opción se podrá determinar si existe o no alguna diferencia en el resultado final del recubrimiento.

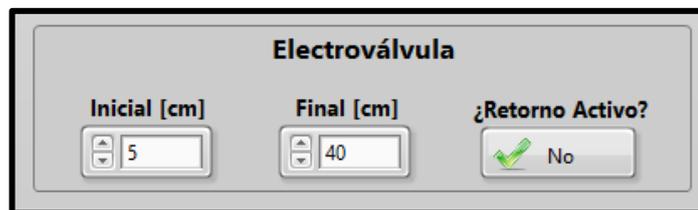


Figura 5. 9 - Menú para configurar el uso de la electroválvula

Una vez que realiza el offset según la configuración explicada, se procede a Pausa de barrido.

PAUSA BARRIDO: En esta etapa, ocurre lo mismo que en la pausa de temperatura, pero con otra variable de temporización. También tiene un reintegro al estado cada 0,5[seg] para permitir actualizar el valor del botón Detener. Al cumplir el tiempo de la pausa, el estado siguiente es el de Run Code.

HOMING: Este estado es el encargado de enviar al RTU el carácter “H”, que hará que el mismo vaya a la posición de Home. Esto está previsto en caso de que la máquina se detenga en cualquier posición arbitraria de la máquina, y mediante este proceso se recupera la referencia. En caso de que se presione el botón de emergencia externo o se accione algún final de carrera, el RTU enviará un mensaje al MTU y este tras dar aviso al usuario y esperar confirmación, enviará a este estado y por lo tanto la máquina se dirigirá a posición de Home.

4.4.3 – Software para proceso de Dipping

El proceso de Dipping⁸ consiste en sumergir las muestras en una sustancia a estudiar. De este modo se logra un recubrimiento superficial por secado de la sustancia sobre la muestra, las variables controlables que pueden modificar dicho recubrimiento superficial y que son realizables por nuestra máquina son: la velocidad de ingreso, tiempo de permanencia en el interior, velocidad de egreso y tiempo de permanencia en el exterior previo a ingresar nuevamente, un número determinado de veces. Para realizar el mismo hemos creado otra aplicación en el panel frontal a la cual se puede acceder desde el botón ya explicado. El panel frontal de ésta aplicación se puede observar en la figura 4.10.

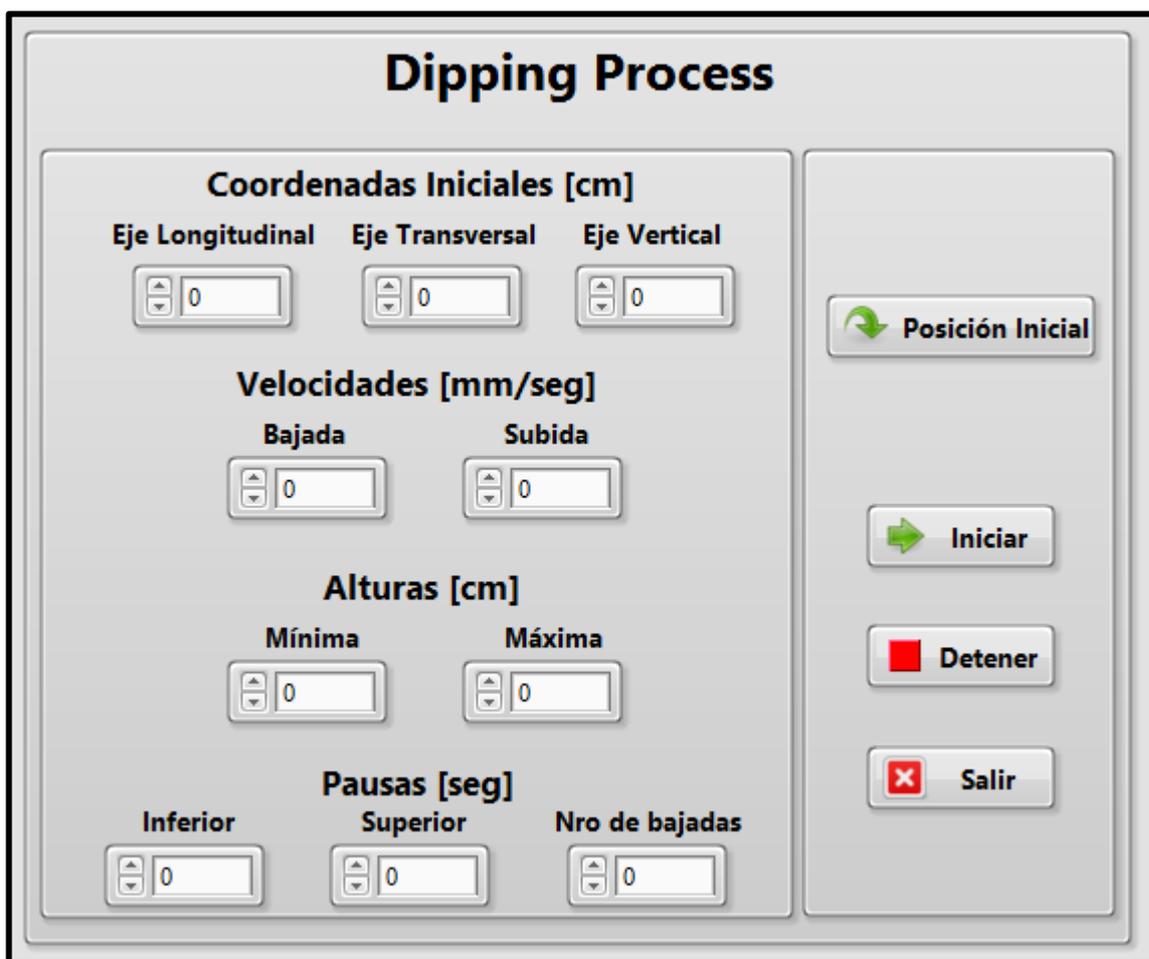


Figura 5. 10 - Panel frontal de aplicación para proceso de dipping

Esta aplicación está optimizada para realizar este proceso, permitiendo la configuración de variables específicas, como por ejemplo una velocidad de bajada diferente al de subida, pausas configurables de manera diferente, etc. De este modo

⁸ Dipping = Inmersión (traducción del inglés al español).

no es un entorno confuso, teniendo además una gran flexibilidad para diseñar el proceso de inmersión.

Para explicar el funcionamiento de este software se partirá del diagrama de flujos de este programa, el cual se muestra en la figura 5.11, y luego se describirán cada uno de sus estados. En este caso se tiene una cantidad grande de estados, para de esta manera simplificar aún más el código y la depuración de fallas, además de que en este proceso el número de decisiones a tomar por el programa es menor.

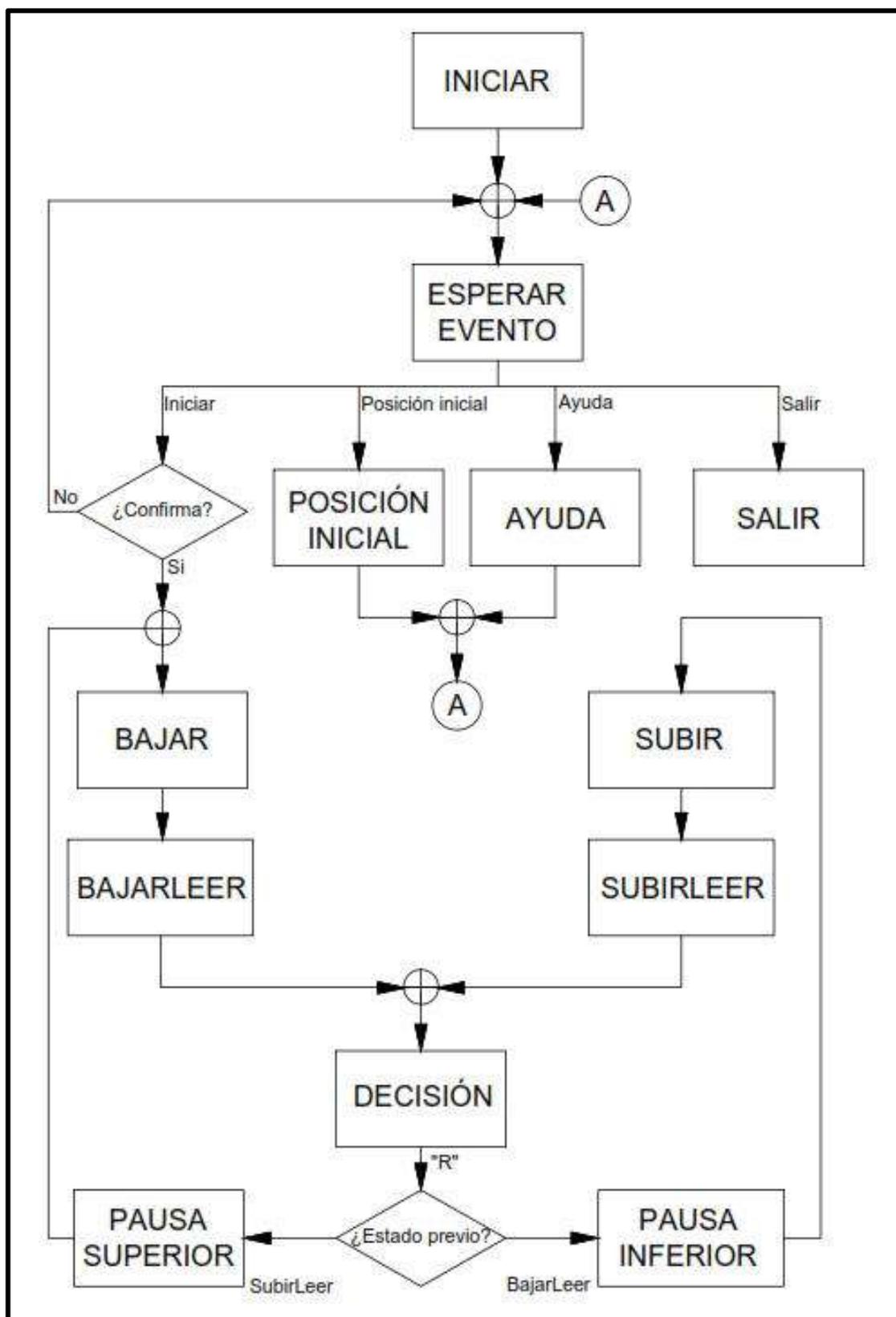
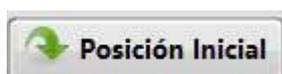


Figura 5. 11 - Diagrama de flujo de aplicación para dipping

Los estados presentes se describen a continuación:

INICIAR: El proceso comienza por este estado, donde se configura el puerto serial al igual que en la aplicación de spray, se borran los buffers del mismo, se configuran valores de variables y luego se envía la configuración de presión para esta aplicación (“P1”).

ESPERAR EVENTO: Durante este estado se espera a que el usuario presione algún botón de función y se actúa dependiendo de ello. A continuación se enumeran los mismos y la función principal para la cual fue diseñado.



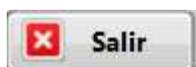
Con este botón se pretende enviar el cabezal o torre a una posición donde se va a realizar el proceso de Dipping, es decir, donde se colocará debajo el recipiente con la sustancia a utilizar y a partir de la cual se moverá luego verticalmente.



Este botón permite comenzar con el movimiento vertical que se producirá entre los valores máximos y mínimos, a la velocidad configurada y con las pausas configuradas (ver figura XXXX).



Al igual que en la aplicación diseñada para spray, no se puede interrumpir el movimiento de los motores mediante serial (no se puede resetear el RTU), pero sí se puede interrumpir el envío de futuros comandos de manera automática hasta finalizar el proceso. Por lo tanto, este botón detiene el proceso y permite al usuario modificar las variables y luego volver a comenzar o detener para cerrar la aplicación. Si se desea detener la máquina de inmediato se debe presionar el botón de emergencia que se encuentra en el exterior de la campana.



Este botón permite cerrar la aplicación de dipping y volver a la de spray.



Este botón abrirá el archivo PDF que contiene el manual de uso tanto de aplicación de spray como Dipping. Facilita al usuario una guía de cómo utilizar y configurar ambos software.

A continuación se explican cada uno de los estados y de esta manera entender como es la secuencia del programa.

POSICIÓN INICIAL: Tal como se explicó anteriormente, con el botón de posición inicial se envía a la máquina a las coordenadas donde se realizara el proceso. El envío del comando de movimiento para alcanzar dicha posición se realiza durante este estado. Luego vuelve a Esperar evento para que el usuario configure, coloque las probetas, etc. hasta que haga clic en Iniciar. Las posiciones serán las configuradas por el usuario según ejes transversal y longitudinal, y en la posición de altura máxima según eje vertical.

BAJAR: Una vez que el usuario hace clic en el botón iniciar, se ingresa en este estado, el cual hará enviara un comando de movimiento (“M[Posic...”), que moverá la posición de la probeta entre una altura máxima y mínima configurada por el usuario. Además se incrementa el número de iteración o contador de bajadas, y luego se dirige al estado de “BajarLeer”, similar al de “Leerserial” de la aplicación de spray, solo que para ganar simplicidad se utiliza uno para la bajada y uno para la subida.

BAJARLEER: Este estado lee el puerto serial de modo que al detectar un carácter en el buffer, lo guarda en un registro de desplazamientos y continúa al estado de decisión. Además, guarda en una variable que el carácter recibido corresponde a la respuesta del movimiento de bajada cuyo uso se analiza más adelante.

DECISION: En este estado, al igual que en decisión del programa de spray, se analiza el carácter recibido y se ejecuta una acción al respecto. Lo único que diferencia a este estado del correspondiente a la otra aplicación, es que en caso de que el carácter recibido corresponda a un movimiento de bajada, el siguiente

estado será el de pausa inferior, caso contrario será pausa superior. Del diagrama de flujo se han evitado las acciones correspondientes a caracteres diferentes a “R” dado que las respuestas son idénticas a la aplicación de spray.

PAUSA INFERIOR/SUPERIOR: Al igual que otros estados de pausa, se lleva adelante un contador de tiempo, y cuando el mismo es alcanzado se dirige a un estado siguiente para que se ejecute el siguiente movimiento. En este caso, si el estado es el de pausa inferior, se procede al estado de “Subir”, si el estado es de la pausa superior, el siguiente será “Bajar”.

SUBIR: En este estado, se envía el comando de movimiento de subida, análogo al de bajada pero en este caso partiendo desde la posición inferior o mínima configurada para el eje vertical, hasta la posición máxima del mismo. En este caso no se debe incrementar el número de un contador de subidas dado que ya es realizado al momento de la bajada y con ese es suficiente. El siguiente estado es “SubirLeer”.

SUBIRLEER: Estado análogo al de “BajarLeer”, con la salvedad de que el carácter recibido corresponde al proceso de subida y por lo tanto se debe guardar dicha información como se mencionó anteriormente. El estado siguiente será nuevamente decisión. Y este determinará en este caso que debe dirigirse a pausa superior si es que el movimiento finalizó correctamente.

EXIT: Este estado solicita confirmación para salir de la aplicación y una vez obtenida, cierra los puertos de comunicación y cierra el VI. De este modo se retorna a la aplicación para depositar spray.

CAPITULO VI EVALUACIÓN DE IMPACTO AMBIENTAL

En este capítulo se pretende realizar un análisis del impacto ambiental (EIA) que puede generar el proceso de fabricación y posterior régimen de funcionamiento al realizar los procesos.

Debemos destacar que el análisis que realizamos en esta sección es simplificado, sin tener en cuenta la posible producción en serie o de grandes cantidades de estas máquinas, como tampoco será considerado su funcionamiento durante períodos importantes o con grandes volúmenes de sustancias peligrosas debido a que el objetivo de nuestro trabajo consiste en diseñar y construir la máquina, y no de prever una fabricación masiva de las mismas.

Con lo expresado en el párrafo anterior, podemos asegurar que los procesos de fabricación mediante los cuales hemos obtenido las piezas no consumen grandes cantidades de energía, agua, químicos, y además generan pocos desechos sólidos o de otro tipo.

Cabe destacar que los materiales utilizados para la construcción de la maquina (aluminio, polietileno, acero, bronce, etc.) son considerados no peligrosos para su utilización.

Existe una directiva de restricción para el uso de ciertas sustancias peligrosas en equipamiento eléctrico y electrónico adoptada por la unión europea (2002/95/EC) en el año 2006 que restringe el uso de seis materiales peligrosos en la fabricación de una variedad de tipos de equipamiento electrónico y eléctrico. Estos materiales son:

- Plomo
- Mercurio
- Cadmio
- Cromo Hexavalente
- Bifenilo Polibrominado
- Ether Difenilo Polibrominado

Esta directiva se aplica en la fabricación de electrodomésticos, equipamiento de telecomunicaciones, equipamiento de iluminación, herramientas eléctricas y electrónicas, juguetes, dispositivos médicos, instrumentos de monitoreo y control, expendedores automáticos y dispositivos semiconductores.

Esta directiva afectaría a nuestro proyecto debido a que se utilizó aleación de estaño y plomo para la soldadura de circuitos electrónicos, pero consideramos que nuestra maquina no pertenece a ninguna de las categorías mencionadas anteriormente, ya que se trata de un equipo de laboratorio de uso específico. Por otra parte, esta directiva solo se aplica en los países asociados a la unión europea, por lo que no es exigido su cumplimiento en nuestro país. De todas maneras, si se quisiera evitar el uso de este material sería necesario recurrir a soldaduras con aleación de estaño-plata, la cual no es posible obtener en el mercado local.

CAPITULO VII ANÁLISIS ECONÓMICO

7.1 – ANÁLISIS DE COSTOS

En el presente capítulo se pretende realizar un análisis simple de los costos de fabricación que ha involucrado nuestra máquina, y luego realizar una comparación con máquinas comerciales de similares características. De este modo pretendemos abarcar el punto de vista económico de nuestro proyecto.

Por otro lado, justificamos realizar un análisis simplificado debido a que el foco de nuestro proyecto no es la producción masiva de la máquina creada y por lo tanto evitaremos cualquier análisis de rentabilidad. Nos limitaremos con ello a una comparación en base a los datos numéricos que manejamos al realizar nuestras compras.

En la tabla 7.1 se puede observar un detalle de los gastos que hemos realizado en las compras de materiales, componentes, servicios de transporte, etc., clasificados bajo los siguientes aspectos a describir:

- Electricidad: Comprende compra de cables, termo contraíble, cinta porta cables helicoidal, grampas, etc., utilizados para la alimentación de motores y sensores.
- Electrónica: Comprende la compra de controladores, drivers, fuente, componentes electrónicos para las placas, terminales, estaño, etc.
- Materiales mecánicos: Comprende materiales utilizados para la construcción de las piezas de la máquina, utilizadas en la estructura, transmisión, acoples, estética, etc.
- Ferretería: Incluye gastos de herramientas, precintos, bulonería, etc.
- Terminación: Comprende gastos en pintura y gastos en materiales utilizados para darle un acabado correcto a la máquina.



PROVEEDOR	DESCRIPCIÓN	FECHA	IMPORTE [\$]	RUBRO
HDC Electro	HDC electro - Materiales electricos	30/03/2015	49,01	Electricidad
HDC Electro	Materiales eléctricos	16/04/2015	76,99	Electricidad
SynerGeek	Arduino UnoR3+ 3xDriver Pololu a4988	10/09/2014	768,70	Electrónica
CyberOffice	Shield CNC v3 + 3xDriver Polulu a4988	04/03/2015	532,00	Electrónica
CyberOffice	2xDriver Polulu a4988	05/03/2015	269,99	Electrónica
Marpla	Componentes electrónicos	06/03/2015	288,01	Electrónica
TEK MDQ	Componentes electrónicos	11/03/2015	574,00	Electrónica
TEK MDQ	Componentes electrónicos	12/03/2015	90,00	Electrónica
Marpla	Componentes electrónicos	26/03/2015	43,05	Electrónica
Marpla	Componentes electrónicos	30/03/2015	165,61	Electrónica
Electroventas	Fuente de 24V/5A	06/04/2015	385,00	Electrónica
Marpla	Componentes electrónicos	10/04/2015	60,05	Electrónica
Free Electron	Componentes Electrónicos	31/03/2015	113,40	Electrónica
Zenner	Componentes electrónicos	27/04/2015	116,00	Electrónica
Free Electron	Componentes electrónicos	20/05/2015	49,00	Electrónica
NY CNC	2 x Motores PAP	03/03/2015	400,00	Electrónica
Fescap	Plancha APM 6x400x1000 Bco	10/09/2014	299,50	Materiales Mec
Fescap	Perfil de aluminio 15x50x6000 mm	10/09/2014	301,75	Materiales Mec
Fornis	Barra torneada (Rosca ACME)	15/09/2014	306,00	Materiales Mec
Metal Store	Ejes lineales SAE 1020 rectific.	15/09/2014	310,00	Materiales Mec
Tratamientos MdP	Niquelado de ejes	20/09/2014	200,00	Materiales Mec
Fescap	Chapa Al 1,5x300x200+APM 15x200x300 Bco	23/09/2014	271,80	Materiales Mec
Fescap	Perfil Al. 15x15x6000	27/03/2015	136,41	Materiales Mec
Jose Alberto Lago	Accesorios para conexión	17/04/2015	140,00	Materiales Mec
Jose Alberto Lago	Electroválcula y accesorios	01/04/2015	575,00	Materiales Mec
Bulonera Comahue	Bulonería/Ferretería	27/03/2015	37,41	Ferretería
Bulonera Comahue	Bulonería/Ferretería	30/03/2015	42,25	Ferretería

Como se muestra en la tabla anterior, realizamos un análisis de cómo se distribuyeron los costos dentro de nuestro proyecto. Para facilitar el impacto visual debido a la gran diferencia de porcentajes presentes en el proyecto, incluimos el diagrama circular de la figura 7.1.

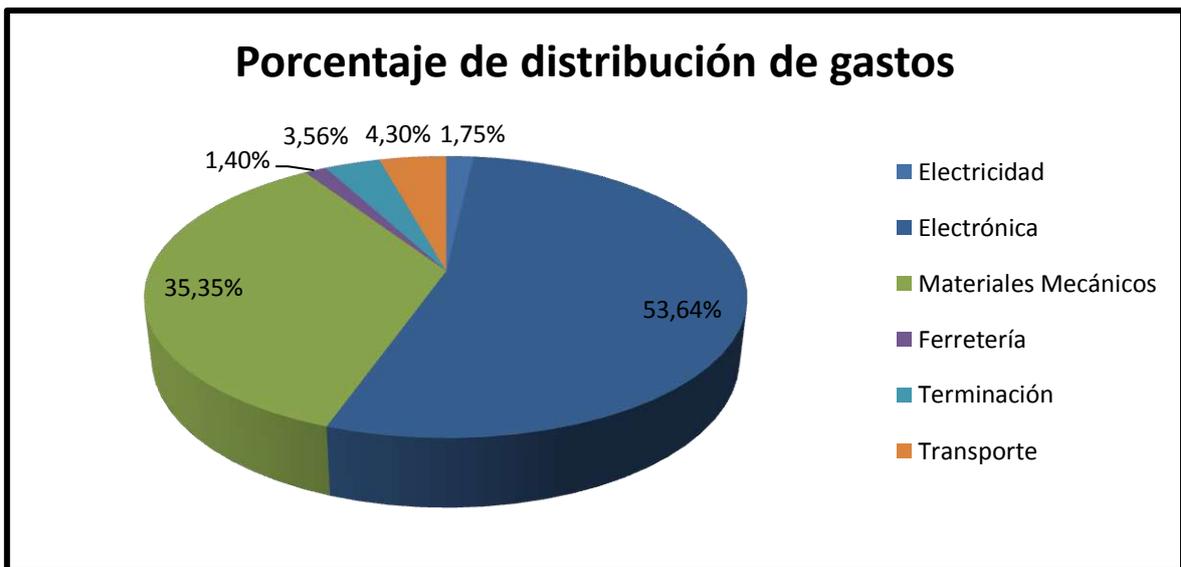


Figura 7.1 – Diagrama circular de distribución

Del diagrama previamente mostrado, debemos aclarar que varios de los gastos incluidos dentro de cada rubro han sido por piezas o elementos que finalmente no han quedado en la máquina debido a cambios en el diseño o que han quedado fuera de funcionamiento durante la construcción y pruebas de funcionamiento como es el caso de 3 drivers de motores PAP.

Se puede observar que la mitad del costo de la máquina está localizado en la electrónica, lo cual consideramos que no es una casualidad debido a que de las áreas incluidas, es la más costosa, vulnerable y de la cual poseemos menos experiencia. Lo cual nos ha demostrado la cautela con la que hay que tratar estos temas y dispositivos y realizar un análisis preciso previo a seleccionar componentes y ponerlos en funcionamiento ya que influyen grandemente en el costo total.

Por otro lado, un fuerte porcentaje se ubica en la compra de materiales mecánicos, lo cual es razonable debido a que son costosos y si bien hemos optimizado los recursos, siempre se obtiene un desperdicio o resto en la fabricación de cada pieza.

Es por ello que optamos por mejorar los procesos de fabricación tal como se discute en el capítulo de conclusiones.

7.2 – COMPARACIÓN CON PRODUCTO SIMILAR:

Esta sección pretende realizar una comparación de costos de nuestra máquina frente al precio de venta de una máquina comercial similar. Pretendemos calificarla con máquinas de sencilla elaboración, sin acabados o materiales de gran calidad. Debemos aclarar que tanto como los precios de nuestra máquina y sin considerar mano de obra, el costo de fabricación resulta groseramente aproximado. Además, consideramos que en caso de realizar más máquinas de este tipo, se resolverá la optimización y la compra de materiales en cantidades suficientes (descuentos por compra mayorista), mejor aprovechamiento por cambios en proceso de fabricación, etc.

De la investigación realizada en páginas de fabricantes, las opciones más económicas de estas máquinas rondan los 60.000\$.

En base a dicho valor, nosotros hemos tenido un costo de fabricación (sin mano de obra) de un 11%. En este caso, no podemos considerar el costo de mano de obra debido a que ha sido un prototipo, y el costo de diseño, ingeniería, y aprendizaje es invaluable. Sin embargo, sabemos que obteniendo un diseño óptimo, el tiempo de construcción de una máquina no debe superar una semana.

Con lo analizado anteriormente, consideramos que hemos cumplido con la obtención de una máquina de bajo presupuesto, y correcto desempeño.

CAPITULO VIII CONCLUSIÓN

Como conclusión podemos decir que el desarrollo de la maquina pudo llevarse a cabo como fue planeado, cumpliendo con todos los requerimientos del usuario.

Cabe destacar que para lograr los objetivos hemos tenido que atravesar una etapa de aprendizaje en la que se adquirieron conocimientos de diseño, tecnologías de fabricación, maquinas eléctricas, circuitos electrónicos y programación. Es por esto que el diseño de la maquina ha ido cambiando a lo largo del desarrollo de la misma, como también lo hicieron los sistemas de movimiento.

Consideramos que es importante mencionar que el diseño del equipo ha ido mutando a lo largo del desarrollo del proyecto y que ha sido necesario resolver ciertos inconvenientes en la medida que han ido surgiendo.

Un claro ejemplo fue la implementación del sistema de movimiento mediante dos motores con el sistema de cuerdas y rodillos para el eje longitudinal, en donde inicialmente se iba a utilizar un único motor con un tornillo de movimiento. En tal caso, el tornillo fue comprado y torneado para nuestra aplicación, pero a la hora del montaje nos dimos cuenta de que la falta de precisión en la alineación del tornillo y en el mecanizado de las piezas iban a resultar en un trabajo exhaustivo para lograr alinear el sistema y que igualmente iba a ser necesario rever la motorización ya que el par necesario iba a aumentar considerablemente (debido al rozamiento excesivo por la desalineación) y la velocidad de rotación iba a tener que ser también muy elevada ya que el tornillo posee un paso de 5 mm por revolución cuando con el sistema de rodillos y cuerda se logra aproximadamente 30mm por revolución, finalmente se decidió desestimar el uso del tornillo y recurrir al sistema de cuerdas el cual, cabe mencionar, resultó ser considerablemente más económico que el sistema del tornillo.

Si decidiéramos producir esta máquina para su comercialización evaluaríamos la posibilidad de hacer los siguientes cambios:

1. Utilización de APM de 15mm:

No utilizaríamos APM de 6mm y construiríamos las placas del puente móvil y las piezas de la torre directamente con placas de plástico de 15mm, lo cual le

- daría rigidez a las partes mencionadas y permitiría evitar la utilización de alojamientos para los rodamientos.
2. Varillas cromadas:
Compráramos varillas cromadas para evitar el proceso de niquelado y así obtener mejor resistencia al desgaste, corrosión y un menor rozamiento.
 3. Rodamientos lineales de bolillas recirculantes:
Los rodamientos lineales de bolillas recirculantes permitirían lograr un movimiento más relajado de la maquina garantizando una mayor vida útil de las guías, soportes y motores.
 4. Mecanizado del plástico por CNC:
Mejoraríamos la precisión del mecanizado y así evitaríamos inconvenientes de alineación, lo que permitiría aumentar la vida útil del equipo.
 5. Tornillo de movimiento de bolillas recirculantes:
Utilizando un tornillo de este tipo, el rozamiento en el mismo disminuiría considerablemente y los motores trabajarían más relajados.
 6. Ensamble de la estructura mediante soldadura:
Utilizando soldaduras para ensamblar las patas, vigas y refuerzos mejoraríamos la rigidez de la estructura y no sería necesario implementar las chapas plegadas para los encuentros.
 7. Desarrollo de una placa controladora integral:
Consideramos que si desarrollamos una placa en la que se pueda montar solo el Arduino y los drivers de los motores, y de ahí poder conectar los motores, la placa calefactora, y los sensores, ahorraríamos trabajo de cableado y armado del gabinete de control.
 8. Separación de cables de control de cables de motorización:
Utilizando cable multipar de 16 hilos mallado podríamos energizar los cuatro motores mejorando la susceptibilidad al ruido por inducción en los finales de carrera.
 9. Finales de carrera más robustos:
Consideramos que utilizando microswitchs más robustos, la vida útil de los mismos aumentaría considerablemente.
 10. Control de temperatura a lazo cerrado:

Pudiendo sensor la temperatura de la placa calefactora con nuestro controlador, podríamos automatizar el proceso de calentamiento de la placa y hacer que una vez que la misma alcance la temperatura deseada, la maquina realice los barridos correspondientes. También creemos conveniente poder mostrarle al usuario cual es la temperatura de la placa en cada instante.

11. Utilizar una electroválvula del tipo 2/2:

Creemos que la utilización de este tipo de válvulas es más acertada para nuestra aplicación ya que no tienen límite de presión mínima para su funcionamiento dado que son comandadas eléctricamente y no son pilotadas como en el caso de la utilizada.

Los cambios mencionados no fueron implementados debido a que no eran de vital importancia para el funcionamiento de la máquina, encarecerían el proyecto, y que además, fueron ideados una vez que la construcción era avanzada, por lo tanto hubiera sido necesario rediseñar el equipo e invertir tiempo innecesariamente considerando que se trata de un prototipo. De todas maneras consideramos que algunos de estos cambios pueden ser realizados en algún proyecto futuro de mejoramiento del prototipo, teniendo en cuenta el comportamiento de la maquina en el funcionamiento cotidiano.

Finalmente podemos decir que estamos muy conformes con el trabajo realizado ya que se permitió desarrollar una maquina la cual facilitará trabajos de investigación para desarrollo de recubrimientos con distintas propiedades, y fue construida a un costo considerablemente menor al que se hubiera requerido con una maquina comercial de igual aplicación.

BIBLIOGRAFÍA

1. VERSATILITY OF CHEMICAL SPRAY PYROLYSIS TECHNIQUE - Pramod S. Patil 1998
2. THIN FILM DEPOSITION USING SPRAY PYROLYSIS - Dainius Perednis & Ludwig J. Gauckler 2003
3. MÁQUINAS DE MEDIR POR COORDENADAS - Grupo de Tecnología Mecánica UNMdP
4. http://www.oocities.org/gabrielordonez_ve/SCADA.htm
5. <http://control-accesos.es/scada/%C2%BFque-es-un-sistema-scada>
6. http://www.portaleso.com/usuarios/Toni/web_simbologia_neuma/simbolos_neumatica_indice.html
7. <http://www.arduino.cc/en/>
8. <http://blog.protoneer.co.nz/arduino-cnc-shield/>
9. <http://blog.protoneer.co.nz/arduino-cnc-shield-v3-00-assembly-guide/>
10. <http://www.ni.com/labview/requirements/esa/>
11. <https://decibel.ni.com/content/blogs/labview-news-english/2013/04/19/complex-application-use-the-queued-message-handler>
12. <http://www.ni.com/academic/students/learn-rio/state-machine/esa/>
13. <https://luisarturo262.wordpress.com/2014/05/14/maquina-de-estados-mydaq-labview/>
14. <http://sine.ni.com/np/app/main/p/ap/ictrl/lang/es/pg/1/sn/n17:ictrl,n21:26/fmid/2757/>
15. <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>
16. https://en.wikipedia.org/wiki/Restriction_of_Hazardous_Substances_Directive

ANEXOS

Anexo A - ARDUINO

A.1 - Programa principal.ino

A continuación se muestra el código del programa principal que posee cargado el Arduino de nuestro RTU.

```
/*
Este programa fue desarrollado para controlar un posicionador XYZ para aplicación de
recubrimientos delgados por los procesos de Spray y Dipping. Para esto se requirió
utilizar comunicación serial, librería de control de motores paso a paso y utilización de
salidas analógicas (PWM) entre otras cosas;

Proyecto final de la carrera de Ingeniería Electromecánica;
Autores: Matías Iriani y Martín Coronel;
Facultad de Ingeniería de la Universidad Nacional de Mar del Plata;
*/

#include <AccelStepper.h> // se incluye librería para control de motores paso a paso;

//Motores a controlar, (tipo de coneccionado, pin de pulsos, pin de dirección);
AccelStepper stepper_X(1, 2, 5);
AccelStepper stepper_Y(1, 3, 6);
AccelStepper stepper_Z(1, 4, 7);

// Definición de entradas y salidas;
int enable_pin      = 8;
int temp_pin        = 9;
int Y_endstop       = 10;
int Z_endstop       = 11;
int X_endstop       = 12;
int gas_pin         = 13;
int heaterSwitch_pin = 14;
int pSwitch_pin     = 15;

// Definición de variables de posición;
long posX;
long posY;
long posZ;
long X0;
long X1;
long homtarget = -99999; //Posición de homing que hace que la maquina retroceda
                        // (indefinidamente) hasta tocar un final de carrera;

int X_offset = 350; //Offset es la posición a la cual se aleja el posicionador
int Y_offset = 500; //una vez alcanzado un final de carrera;
int Z_offset = 1500;

// Definición de variables de velocidad;
int homing_speed = 600;
int X_Speed      = 1000;
int Y_Speed      = 500;
int Z_Speed      = 1000;

// Definición de variables de aceleración;
int homing_acceleration = 8000;
int X_Acceleration      = 7500;
int Y_Acceleration      = 1875;
int Z_Acceleration      = 7500;

// Definición de variables de dirección;
boolean dirX;
boolean dirX_stop;
boolean dirY;
boolean dirY_stop;
boolean dirZ;
```

```
boolean dirZ_stop;

// Definición de variables auxiliares;
int X_endstop_lastState = 1;
int Y_endstop_lastState = 1;
int Z_endstop_lastState = 1;
int importaPresion = 0;
int pSwitch_value = 0;
boolean valorX = 1;
boolean valorY = 1;
boolean valorZ = 1;
boolean boolean_gas = false;
boolean boolean_enable = true;

void setup() // Este ciclo se realiza por única vez cuando el controlador es encendido;
{
  //Definición de salidas;
  pinMode(gas_pin, OUTPUT);
  pinMode(temp_pin, OUTPUT);
  pinMode(enable_pin, OUTPUT);
  pinMode(heaterSwitch_pin, OUTPUT);

  //Definición de entradas activando PullUps internos;
  pinMode(pSwitch_pin, INPUT_PULLUP);
  pinMode(X_endstop, INPUT_PULLUP);
  pinMode(Y_endstop, INPUT_PULLUP);
  pinMode(Z_endstop, INPUT_PULLUP);

  //Seteo de velocidades default;
  stepper_X.setMaxSpeed(X_Speed);
  stepper_Y.setMaxSpeed(Y_Speed);
  stepper_Z.setMaxSpeed(Z_Speed);

  //Seteo de aceleraciones default;
  stepper_X.setAcceleration(X_Acceleration);
  stepper_Y.setAcceleration(Y_Acceleration);
  stepper_Z.setAcceleration(Z_Acceleration);

  //Inicio de comunicación Serial a 9600 bites por segundo;
  Serial.begin(9600);

  //Al iniciar el controlador, la maquina se dirige a su posición de home;
  Homing_X();
  Homing_Y();
  Homing_Z();
}

void loop() //Este ciclo se ejecuta una y otra vez hasta que el controlador sea apagado;
{
  while (Serial.available()) // si hay información para leer proveniente del puerto serial,
  //entonces se entra en el siguiente ciclo;
  {
    char funcion = (char)Serial.read(); //Lee el caracter proveniente del puerto serial y lo
    //guarda en la variable "funcion";

    char caracter;
    switch (funcion) //Dependiendo del caracter que sea "funcion" se ejecuta el ciclo
    //correspondiente;
    {
      case 'M':
        //Desde el ordenador se envía una sucesión de caracteres mediante el puerto serial
        //con la siguiente sintaxis: M###,###,###,###,### donde cada conjunto de números son
        //leídos a continuación;
        {
          posX = Serial.parseInt();
          posY = Serial.parseInt();
          posZ = Serial.parseInt();
          X0 = Serial.parseInt();
          X1 = Serial.parseInt();

          pSwitch_value = digitalRead(pSwitch_pin); //Se lee la entrada digital del sensor de
          //presión;

          if (pSwitch_value == 0 || importaPresion) //Si el sensor de presión indica que la
          //depresión en la campana es la correcta o
```

```

//la presión no importa entonces se
//ejecuta el siguiente ciclo;
{
  if (M(posX, posY, posZ, X0, X1))//Se llama al comando M, el cual se define
    //posteriormente, el cual devuelve el valor True
    //si la posición indicada se alcanza
    //correctamente o False si no se pudo alcanzar la
    //posición porque se alcanzó algún final de
    //carrera;
    {
      Serial.println("R");//Se envía el caracter R desde el controlador al ordenador,
        //indicando que la posición fue alcanzada correctamente;
    }
  else//Si no se pudo alcanzar la posición, entonces se ejecuta el siguiente ciclo;
  {
    Serial.println("A");//Se envía el caracter A desde el controlador al ordenador,
      //indicando que la posición no fue alcanzada correctamente;

    caracter = (char)Serial.read();//Se espera una respuesta desde el ordenador;

    while (caracter != 'H')//Si esa respuesta no es el caracter H, entonces se
      //mantiene leyendo hasta que desde el ordenador se envíe
      //la letra H para llevar el posicionador hasta la
      //posición de home en el eje en el que se tocó el final
      //de carrera;
      {
        caracter = (char)Serial.read();
      }

    if (!valorX)
    {
      Homing_X();
    }
    else if (!valorY)
    {
      Homing_Y();
    }
    else
    {
      Homing_Z();
    }
  }
}
else//Si el sensor de presión determina que la presión no es suficiente y la
//presión importa entonces se envía al ordenador el caracter P;
{
  Serial.println("P");
}
}
break;//finaliza la función "M";

case 'S'://Desde el ordenador se envía una sucesión de caracteres mediante el puerto
//serial con la siguiente sintaxis: S###, donde los números indican la
//temperatura a la cual se debe setear la placa calefactora;
{
  int temp_val = Serial.parseInt();
  if (temp_val > 0)
  {
    digitalWrite(heaterSwitch_pin, HIGH);//Se acciona el switch de la placa
      //calefactora si la temperatura seteada
      //es mayor que 0;
  }
  else
  {
    digitalWrite(heaterSwitch_pin, LOW);//Se apaga el switch de la placa calefactora
      //si la temperatura seteada es igual a 0;
  }
  analogWrite(temp_pin, temp_val);//Se setea la temperatura de la placa calefactora
    //mediante la salida analógica del controlador;
}
break;

case 'P'://Desde el ordenador se envía una sucesión de caracteres mediante el puerto
//serial con la siguiente sintaxis: P# (P0 o P1) para determinar si importa
```

```
//la presión de la campana, el controlador devuelve 0 o 1 si la depresión es
//suficiente o no respectivamente;
{
  importaPresion = Serial.parseInt();
  pSwitch_value = digitalRead(pSwitch_pin);
  Serial.println(pSwitch_value);
}
break;

case 'U': //Desde el ordenador se envía una sucesión de caracteres mediante el puerto
//serial con la siguiente sintaxis: U###,### para modificar los valores de
//velocidad y aceleración del eje X
{
  X_Speed = Serial.parseInt();
  X_Acceleration = Serial.parseInt();
  stepper_X.setMaxSpeed(X_Speed);
  stepper_X.setAcceleration(X_Acceleration);
}
break;

case 'V': //Desde el ordenador se envía una sucesión de caracteres mediante el puerto
//serial con la siguiente sintaxis: V###,### para modificar los valores de
//velocidad y aceleración del eje Y;
{
  Y_Speed = Serial.parseInt();
  Y_Acceleration = Serial.parseInt();
  stepper_Y.setMaxSpeed(Y_Speed);
  stepper_Y.setAcceleration(Y_Acceleration);
}
break;

case 'W': //Desde el ordenador se envía una sucesión de caracteres mediante el puerto
//serial con la siguiente sintaxis: W###,### para modificar los valores de
//velocidad y aceleración del eje Z;
{
  Z_Speed = Serial.parseInt();
  Z_Acceleration = Serial.parseInt();
  stepper_Z.setMaxSpeed(Z_Speed);
  stepper_Z.setAcceleration(Z_Acceleration);
}
break;

case 'H': //Si se recibe desde el ordenador el caracter H, entonces la maquina se envía
//a la posición de home;
{
  Homing_X();
  Homing_Y();
  Homing_Z();
}
break;

case 'T': //Si desde el ordenador se recibe el carácter T, se conmuta la solenoide de la
//electroválvula (solo para pruebas);
{
  if (boolean_gas)
  {
    digitalWrite(gas_pin, LOW);
    boolean_gas = false;
  }
  else
  {
    digitalWrite(gas_pin, HIGH);
    boolean_gas = true;
  }
}
break;

case 'E': //Si desde el ordenador se recibe el carácter E, se cambia el estado del pin
//de Enable para deshabilitar los drivers de los motores;
{
  if (boolean_enable)
  {
    digitalWrite(enable_pin, LOW);
    boolean_enable = false;
    Serial.println("enable off");
  }
}
```

```
    }  
    else  
    {  
        digitalWrite(enable_pin, HIGH);  
        boolean_enable = true;  
        Serial.println("enable on");  
    }  
}   
break;  
}  
}  
}
```

A.2 - Comando M.ino

Código correspondiente al subprograma del comando “M” del Arduino.

```
//Esta función ejecuta el movimiento en las tres direcciones y acciona la electroválvula  
//entre las posiciones X0 y X1, si se toca algún final de carrera devuelve el valor False  
//y detiene el movimiento, si el movimiento es finalizado correctamente devuelve el valor  
//True;  
  
boolean M(int X, int Y, int Z, int Xa, int Xb)  
{  
    boolean respuesta = false;  
    //Se setean los objetivos de posición;  
    stepper_X.moveTo(X);  
    stepper_Y.moveTo(Y);  
    stepper_Z.moveTo(Z);  
  
    while (stepper_Z.distanceToGo() != 0 && Z_limit_reached())  
    //Mientras haya distancia por recorrer y no se haya alcanzado el final de carrera se  
    //realiza el movimiento en el eje Z (ver función Z_limit_reached());  
    {  
        stepper_Z.run();  
    }  
  
    while (stepper_Y.distanceToGo() != 0 && Y_limit_reached() && valorZ)  
    //Mientras haya distancia por recorrer, no se haya alcanzado el final de carrera y el  
    //movimiento en Z se haya realizado correctamente, se realiza el movimiento en el eje Y  
    //(ver función Y_limit_reached());  
    {  
        stepper_Y.run();  
    }  
  
    while (stepper_X.distanceToGo() != 0 && X_limit_reached() && valorZ && valorY)  
    //Mientras haya distancia por recorrer, no se haya alcanzado el final de carrera y los  
    //movimientos en Y y en Z se haya realizado correctamente, se realiza el movimiento en el  
    //eje X accionando la electroválvula entre las posiciones X0 y X1 (ver función  
    //X_limit_reached());  
    {  
        stepper_X.run();  
        if (stepper_X.currentPosition() >= Xa && stepper_X.currentPosition() <= Xb)  
        {  
            digitalWrite(gas_pin, HIGH);  
        }  
        else  
        {  
            digitalWrite(gas_pin, LOW);  
        }  
    }  
    digitalWrite(gas_pin, LOW);  
    int valor = valorX * valorY * valorZ;  
    return valor; //Esta función devuelve el valor True solo si todos los movimientos fueron  
                //realizados correctamente;  
}
```

```
boolean X_limit_reached()//Devuelve el valor False cuando el final de carrera es accionado,  
//setea como dirección de detención a la dirección que traía;  
{  
  
    if (stepper_X.targetPosition() > stepper_X.currentPosition())//Se define la dirección de  
                                                                    //movimiento;  
    {  
        dirX = true;  
    }  
    else  
    {  
        dirX = false;  
    }  
  
    valorX = true;  
    int microSwitch = digitalRead(X_endstop);//Se lee el estado del final de carrera del  
                                                //eje X y se guarda en la variable microSwitch;  
    if (!microSwitch)//Si el final de carrera esta accionado, entonces ValorX se setea falso;  
    {  
        valorX = false;  
    }  
    if (microSwitch < X_endstop_lastState)//Si el valor de microSwitch es menor que el estado  
                                                //anterior del final de carrera (es decir, el  
                                                //microSwitch pasa de 1 a 0) se define la dirección  
                                                //de detención como la dirección de movimiento  
                                                //actual;  
    {  
        dirX_stop = dirX;  
    }  
    X_endstop_lastState = microSwitch;//Se guarda el estado de microSwitch en la variable que  
                                        //define el último estado del final de carrera;  
    return valorX;  
}  
  
boolean Y_limit_reached()//Devuelve el valor False cuando el final de carrera es accionado,  
//setea como dirección de detención a la dirección que traía. Idem  
//X_limit_reached();  
{  
    if (stepper_Y.targetPosition() > stepper_Y.currentPosition())  
    {  
        dirY = true;  
    }  
    else  
    {  
        dirY = false;  
    }  
  
    valorY = true;  
    int microSwitch = digitalRead(Y_endstop);  
  
    if (!microSwitch)  
    {  
        valorY = false;  
    }  
    if (microSwitch < Y_endstop_lastState)  
    {  
        dirY_stop = dirY;  
    }  
    Y_endstop_lastState = microSwitch;  
    return valorY;  
}  
  
boolean Z_limit_reached()//Devuelve el valor False cuando el final de carrera es accionado,  
//setea como dirección de detención a la dirección que traía. Idem  
//X_limit_reached();  
{  
  
    if (stepper_Z.targetPosition() > stepper_Z.currentPosition())  
    {  
        dirZ = true;  
    }  
    else  
    {  
        dirZ = false;  
    }  
}
```

```
valorZ = true;
int microSwitch = digitalRead(Z_endstop);

if (!microSwitch)
{
  valorZ = false;
}
if (microSwitch < Z_endstop_lastState)
{
  dirZ_stop = dirZ;
}

Z_endstop_lastState = microSwitch;
return valorZ;
}
```

A.3 - Homing.ino

En este apartado se mostrará el código correspondiente al subprograma de “Homing” del software del Arduino.

```
void Homing_X()//Realiza el movimiento de retroceso a velocidad constante hasta alcanzar
               //los finales de carrera;
{
  delay (100);
  stepper_X.move(hometarget);//Setea la posición a alcanzar a una posición que se
                             //encontraría fuera de los límites de la máquina, de manera
                             //que se alcancen los finales de carrera;

  stepper_X.setMaxSpeed(homing_speed);//Se setean las velocidades de homing;
  stepper_X.setAcceleration(homing_acceleration);

  while (X_limit_reached() || dirX_stop)//Mientras que el final de carrera no fue accionado
                                         //o la dirección de detención es ascendente,
                                         //entonces el posicionador retrocede (hasta que el
                                         //final de carrera se accione y la dirección de
                                         //detención sea descendente);
  {
    stepper_X.run();
  }
  stepper_X.setMaxSpeed(homing_speed);
  stepper_X.setAcceleration(homing_acceleration);
  stepper_X.move(X_offset);//Se setea un offset como posición objetivo;
  delay (500);
  while (stepper_X.distanceToGo()>0)//Mientras haya distancia por recorrer, realiza el
                                     //movimiento;
  {
    stepper_X.run();
  }
  stepper_X.setCurrentPosition(0);//Setea la posición actual como 0;
  X_limit_reached();//Realiza un ciclo de la función X_limit_reached() para actualizar las
                    //variables;
  stepper_X.setMaxSpeed(X_Speed);//Se restablecen los valores para las velocidades y
                                //aceleraciones
  stepper_X.setAcceleration(X_Acceleration);
}

void Homing_Y()//Idem Homing_X()
{
  delay (100);
  stepper_Y.move(hometarget);

  stepper_Y.setMaxSpeed(homing_speed);
  stepper_Y.setAcceleration(homing_acceleration);

  while (Y_limit_reached() || dirY_stop)
  {
    stepper_Y.run();
  }
  stepper_Y.setMaxSpeed(homing_speed);
}
```

```
stepper_Y.setAcceleration(homing_acceleration);
stepper_Y.move(Y_offset);
delay (500);

while (stepper_Y.distanceToGo()>0)
{
  stepper_Y.run();
}
stepper_Y.setCurrentPosition(0);
Y_limit_reached();
stepper_Y.setMaxSpeed(Y_Speed);
stepper_Y.setAcceleration(Y_Acceleration);
}

void Homing_Z()//Idem Homing_X()
{
  delay (100);
  stepper_Z.move(hometarget);
  stepper_Z.setMaxSpeed(homing_speed);
  stepper_Z.setAcceleration(homing_acceleration);

  while (Z_limit_reached() || dirZ_stop)
  {
    stepper_Z.run();
  }
  stepper_Z.setMaxSpeed(homing_speed);
  stepper_Z.setAcceleration(homing_acceleration);
  stepper_Z.move(Z_offset);
  delay (500);

  while (stepper_Z.distanceToGo()>0)
  {
    stepper_Z.run();
  }
  stepper_Z.setCurrentPosition(0);
  Z_limit_reached();
  stepper_Z.setMaxSpeed(X_Speed);
  stepper_Z.setAcceleration(X_Acceleration);
}
```

Anexo B - LABVIEW

El presente anexo pretende explicar por un lado el funcionamiento de la arquitectura de programación utilizada en nuestra aplicación, y por otro lado documentar el código del programa.

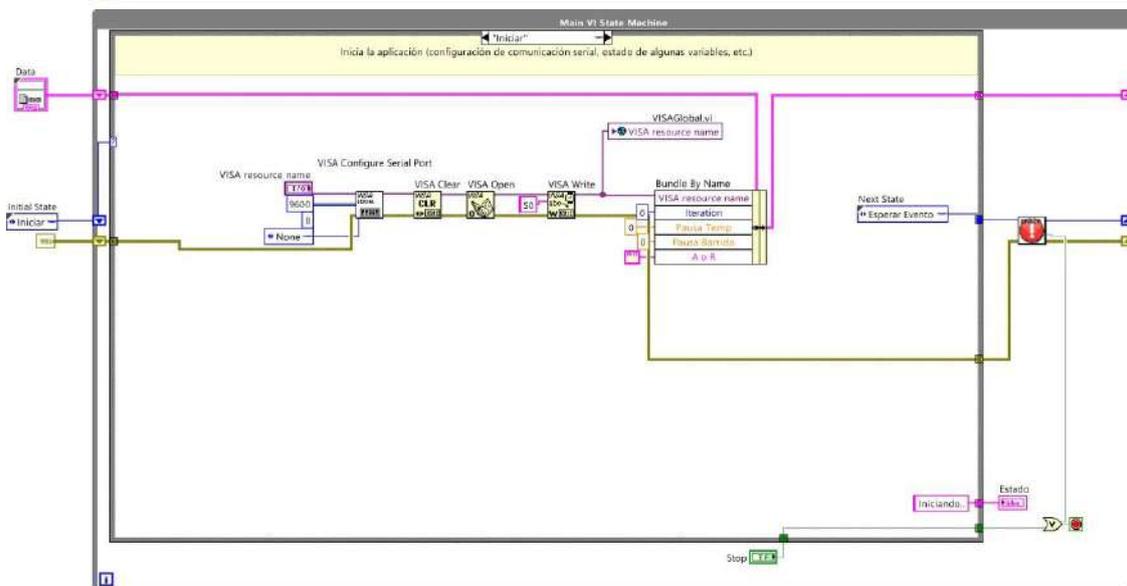
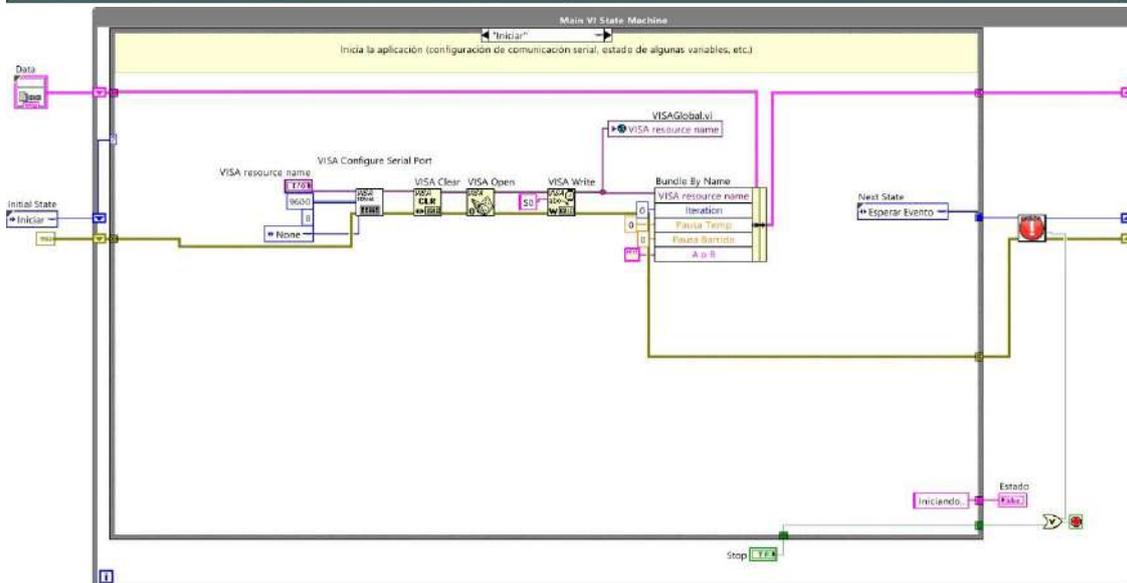
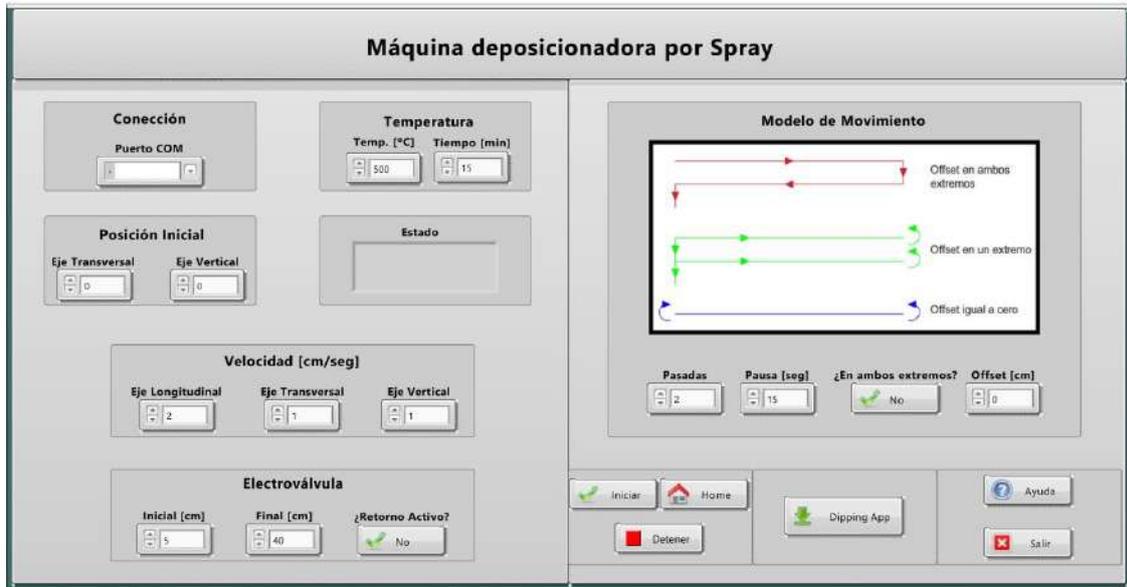
B.1 - Máquina de estados:

En cuanto a la explicación de la arquitectura de máquina de estados, nos hemos basado en el tutorial⁹ que provee el fabricante de LabView (National Instruments).

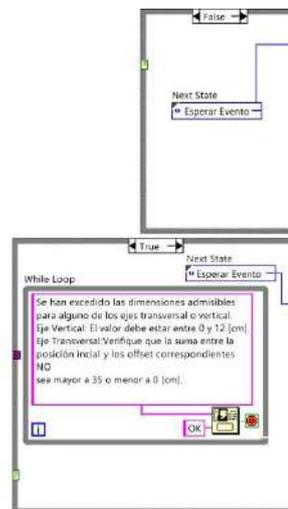
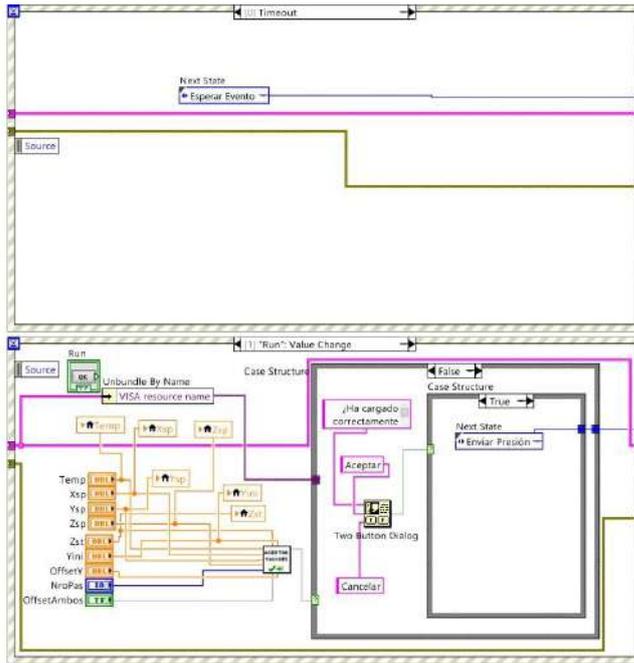
B.2 - Depósito por spray

B.2.1 - Programa principal

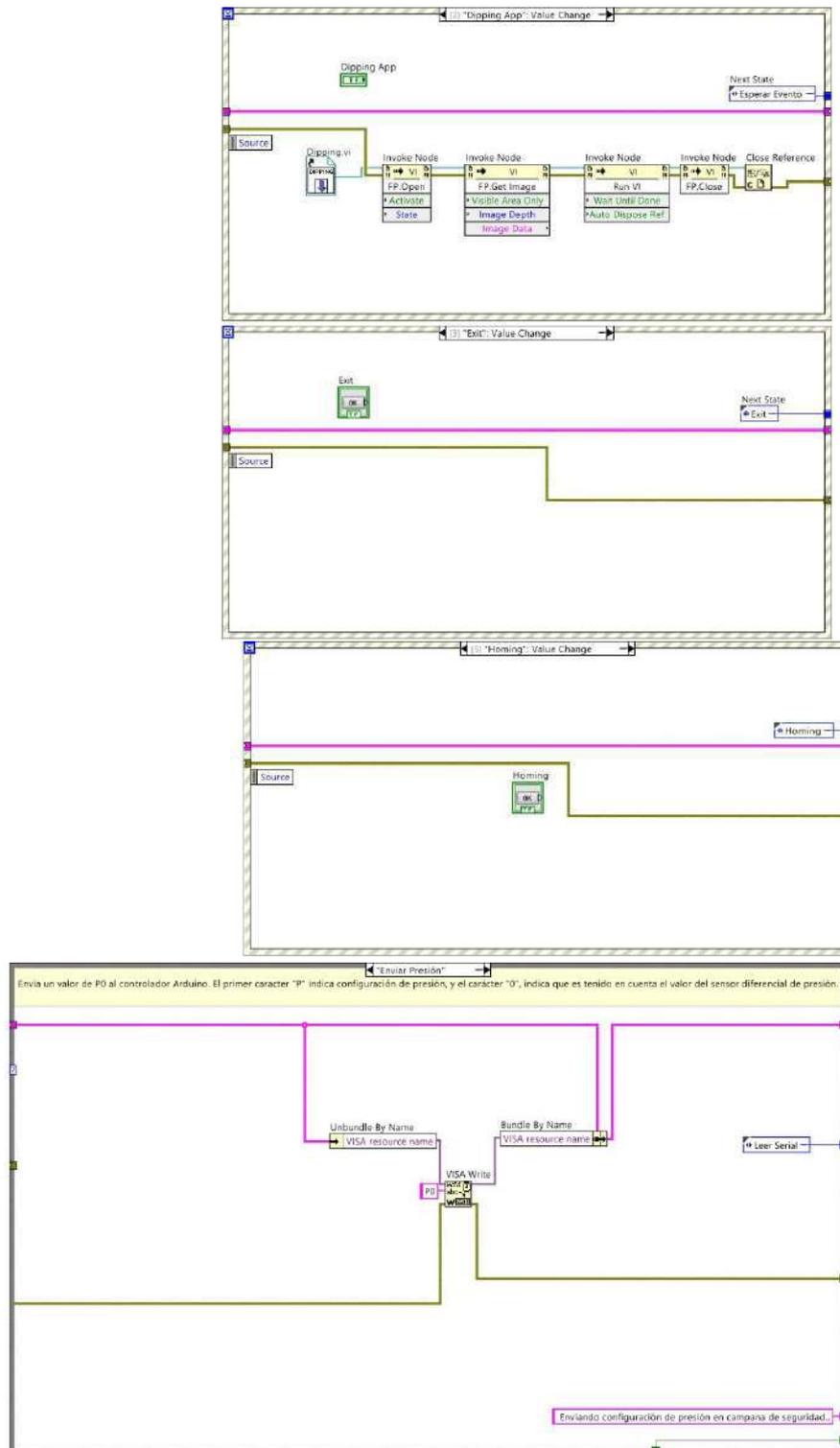
⁹ <http://www.ni.com/tutorial/14120/es/>



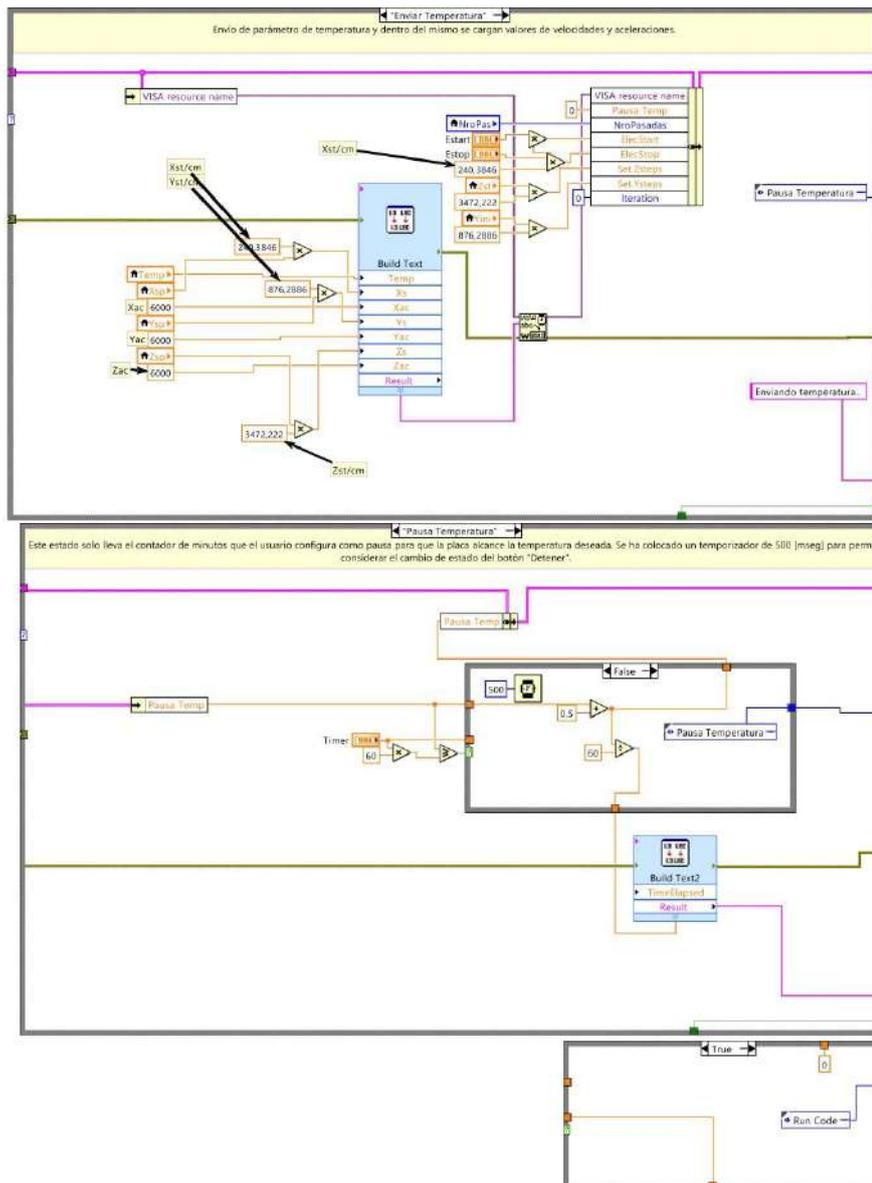
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”



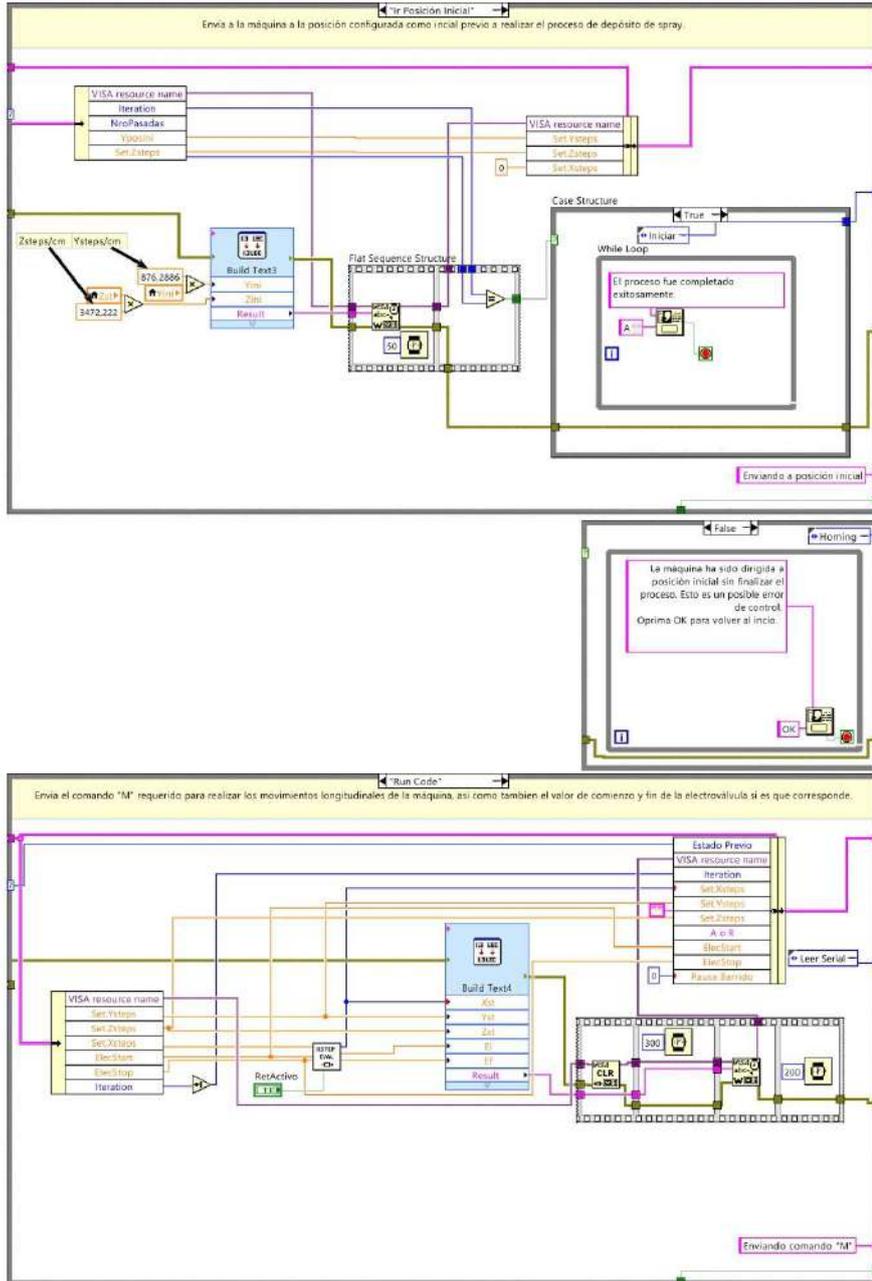
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”



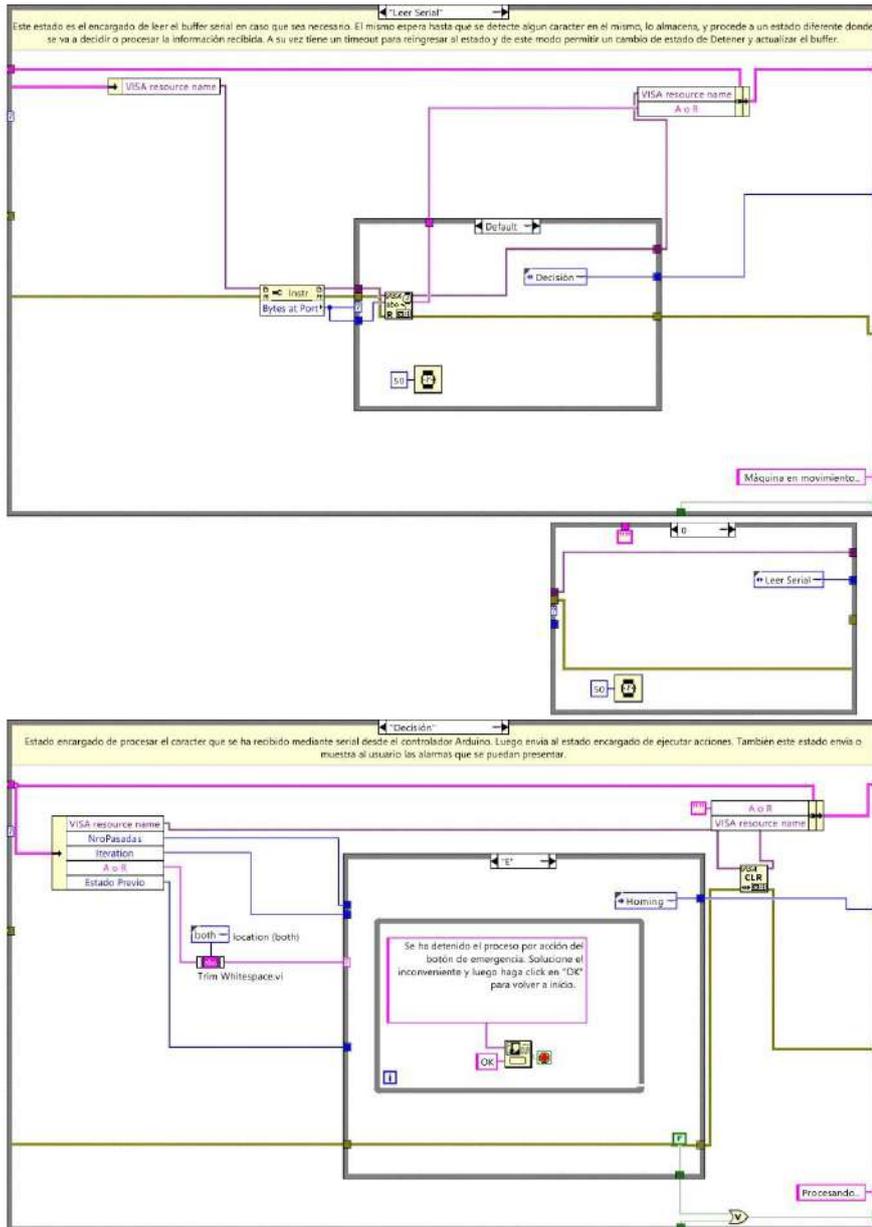
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

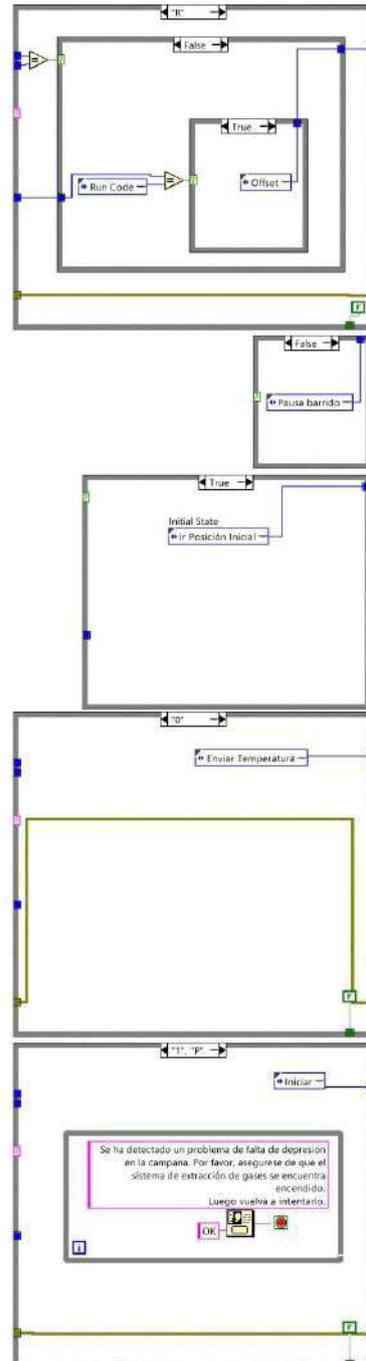


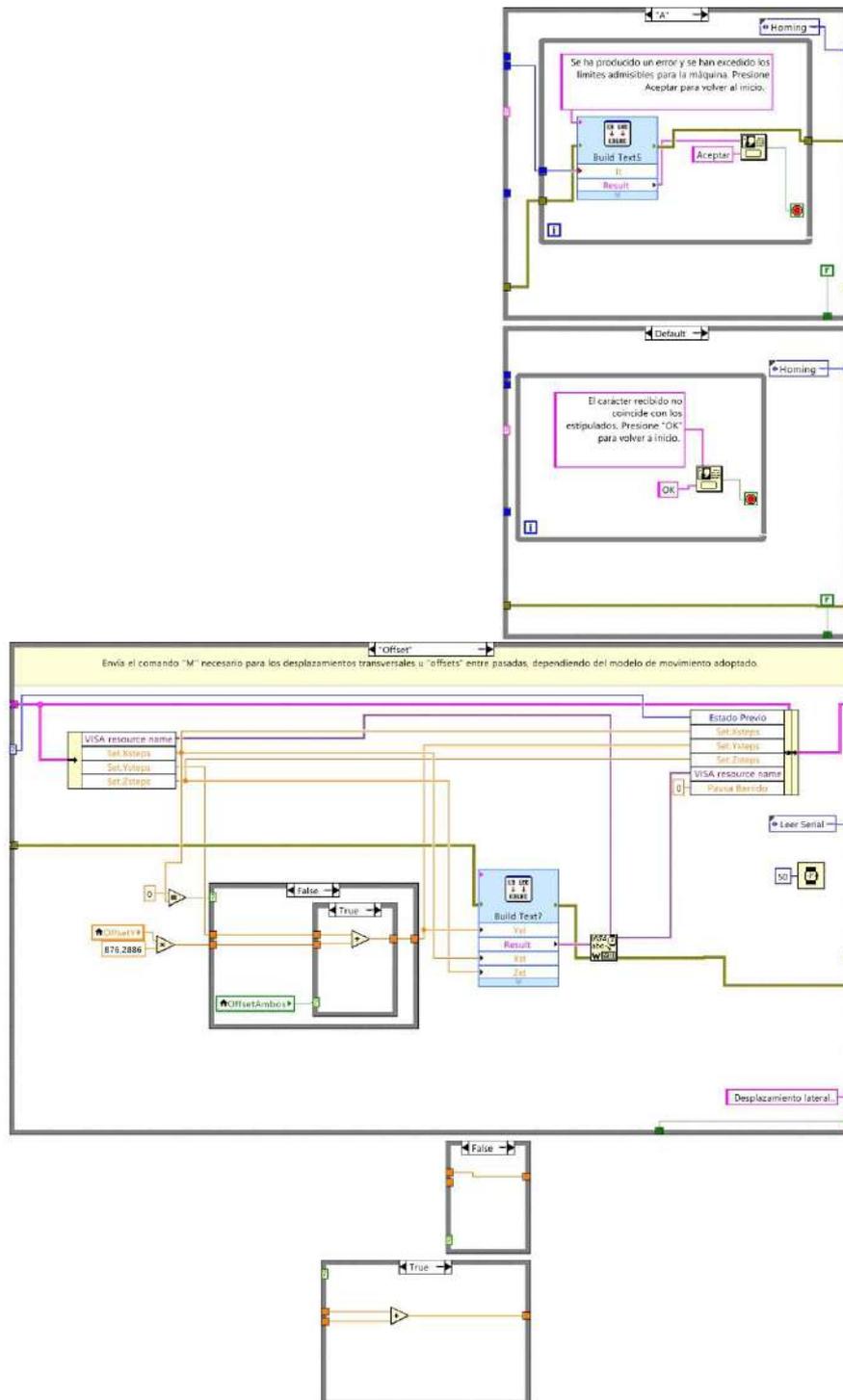
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”



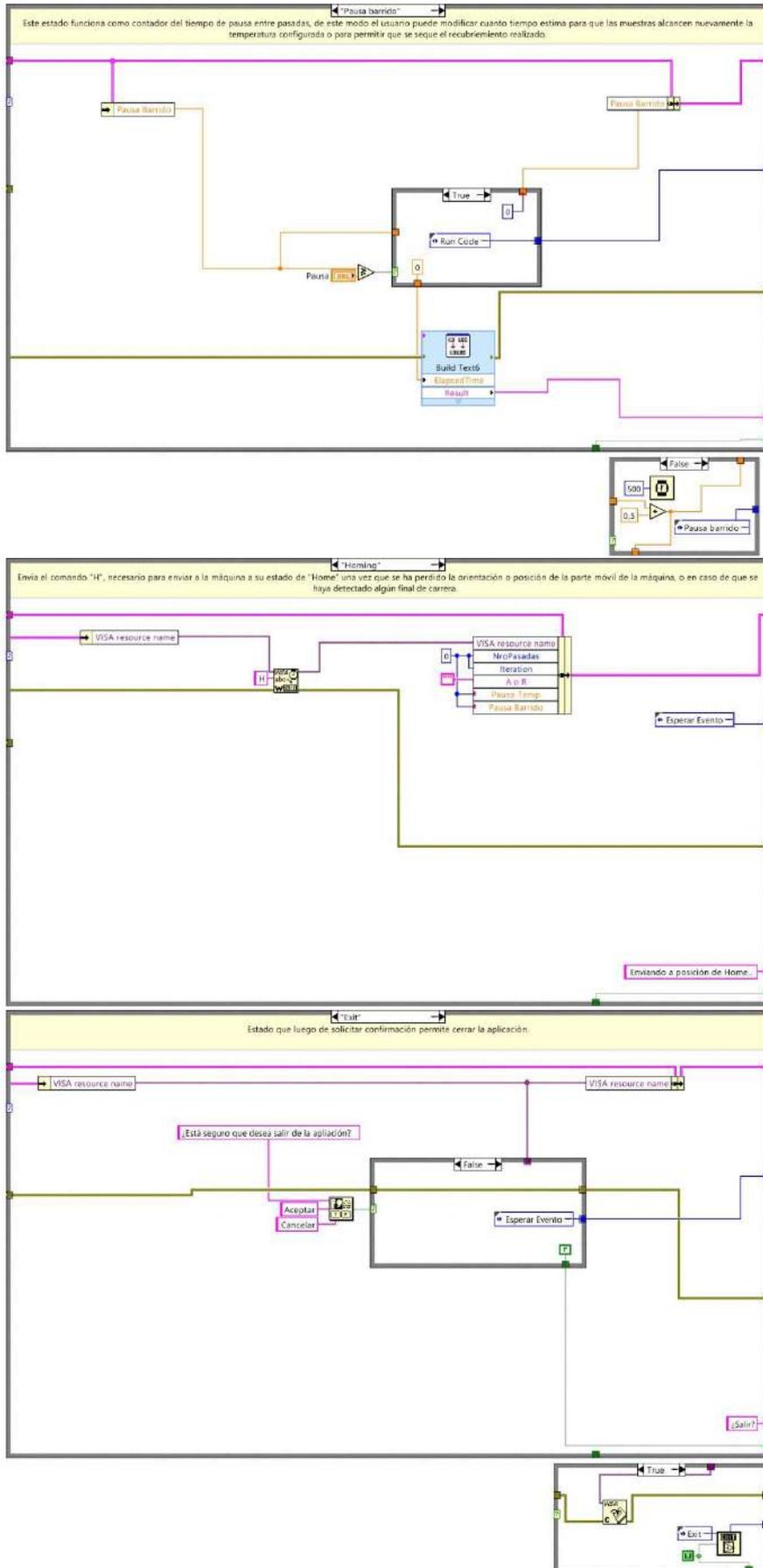
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”







“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

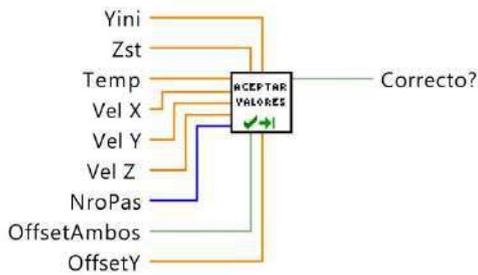


“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

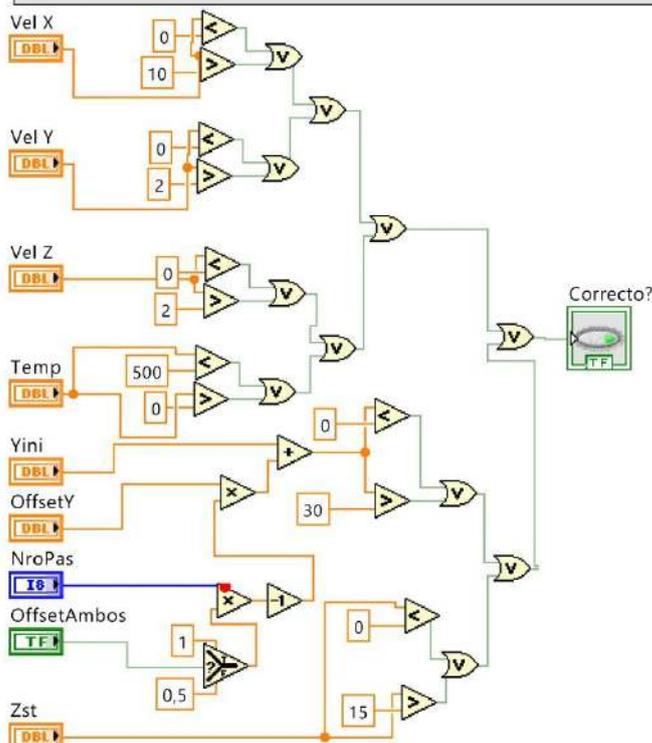
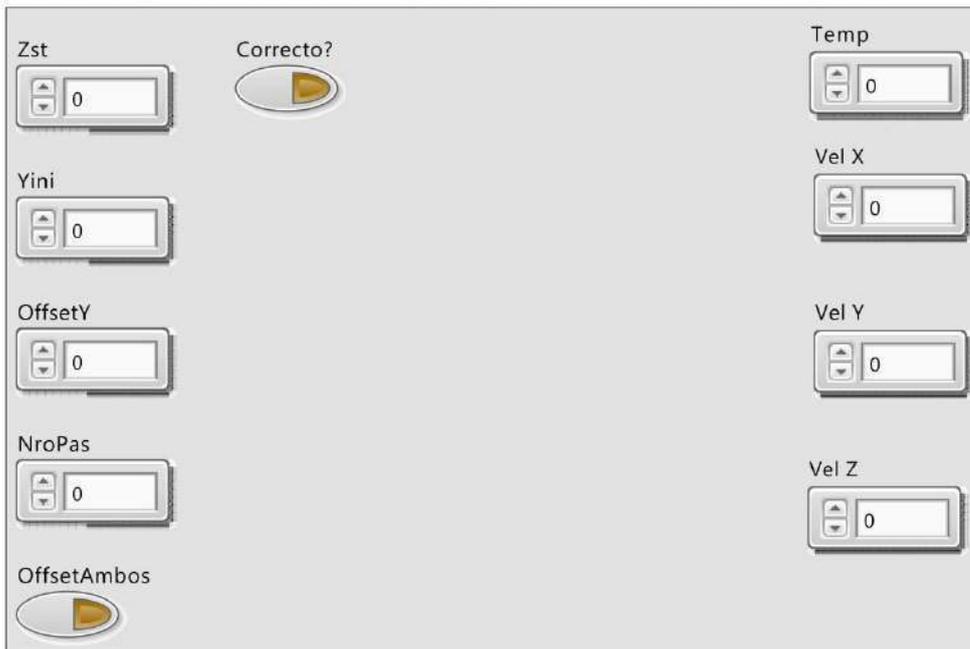


B.2.2 - SubVI para manejo de datos numéricos ingresados

ValoresInRange.vi

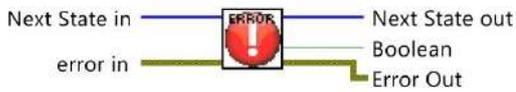


SubVI que controla que los valores de velocidad, distancias y temperaturas que ingrese el operador estén dentro del rango admisible de la máquina, para pedir que se corrijan los mismos previo a iniciar el proceso.

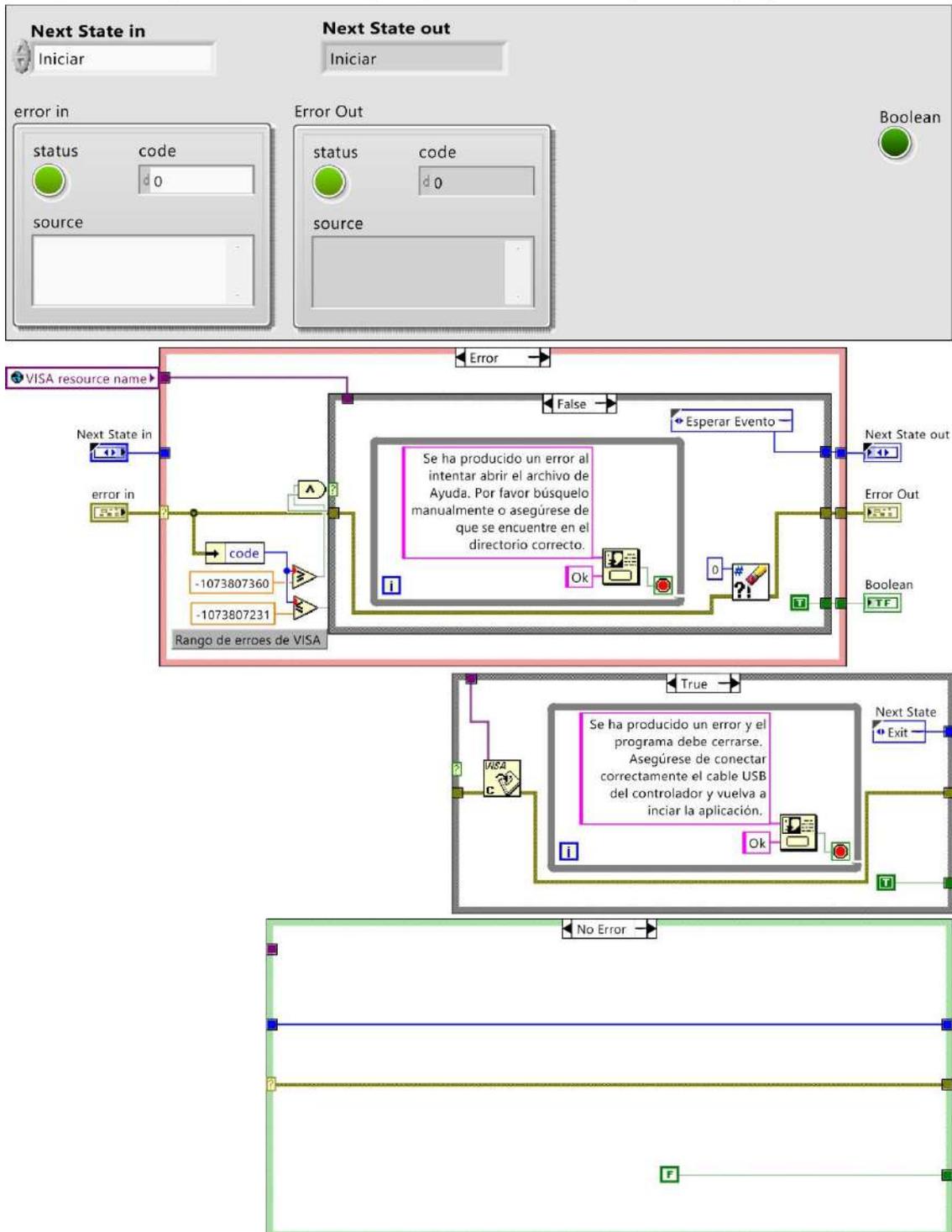


B.2.3 - SubVI de manejo de errores

Error Handler SubVI.vi

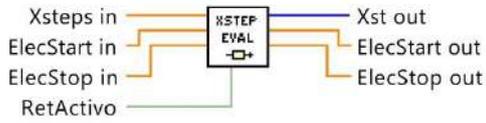


SubVI diseñado para actuar en caso de que se produzca un error en la ejecución del programa.

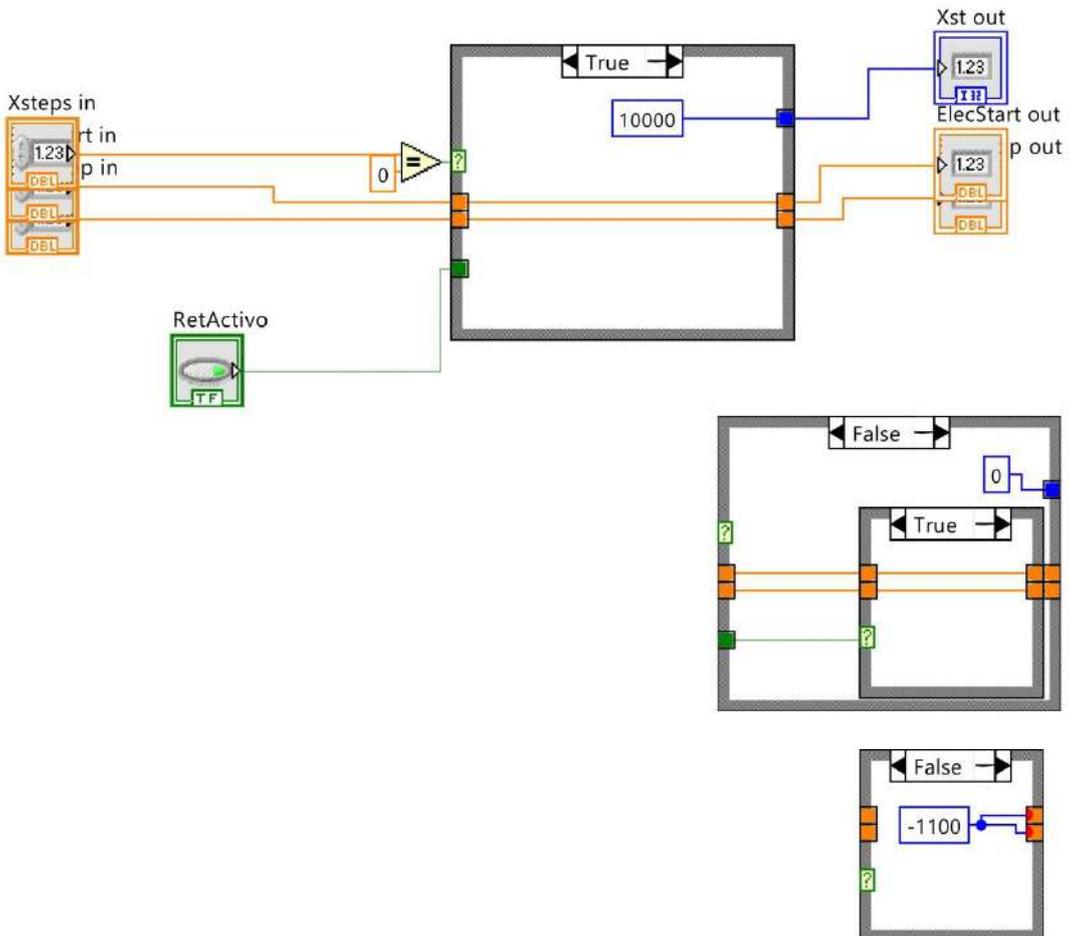
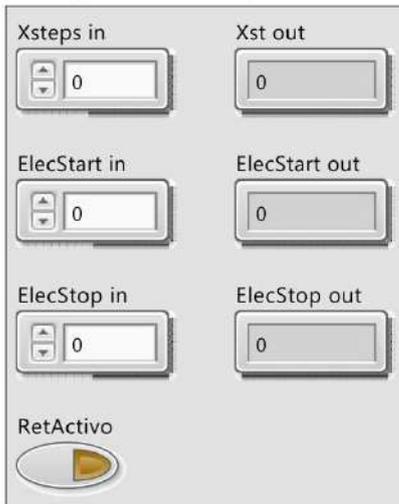


B.2.4 - SubVI de control de coordenadas de apertura y cierre de electroválvula

XstepEvalv (SubVI).vi



SubVI que administra la posición de inicio y corte del funcionamiento de la electroválvula.



VI de aplicación para depósito por inmersión o “dipping”.

Dipping.vi



Esta aplicación fue diseñada para realizar procesos de Dipping, es decir, de realizar movimientos verticales a velocidades controladas, con el objetivo de sumergir probetas en diferentes sustancias formando una película superficial.

Dipping Process

Coordenadas iniciales [cm]

Eje Longitudinal: 10 Eje Transversal: 10

Velocidades [mm/seg]

Bajada: 10 Subida: 10

Alturas [cm]

Mínima: 0 Máxima: 5

Pausas [seg]

Inferior: 0 Superior: 0 Nro de bajadas: 1

Posición Inicial

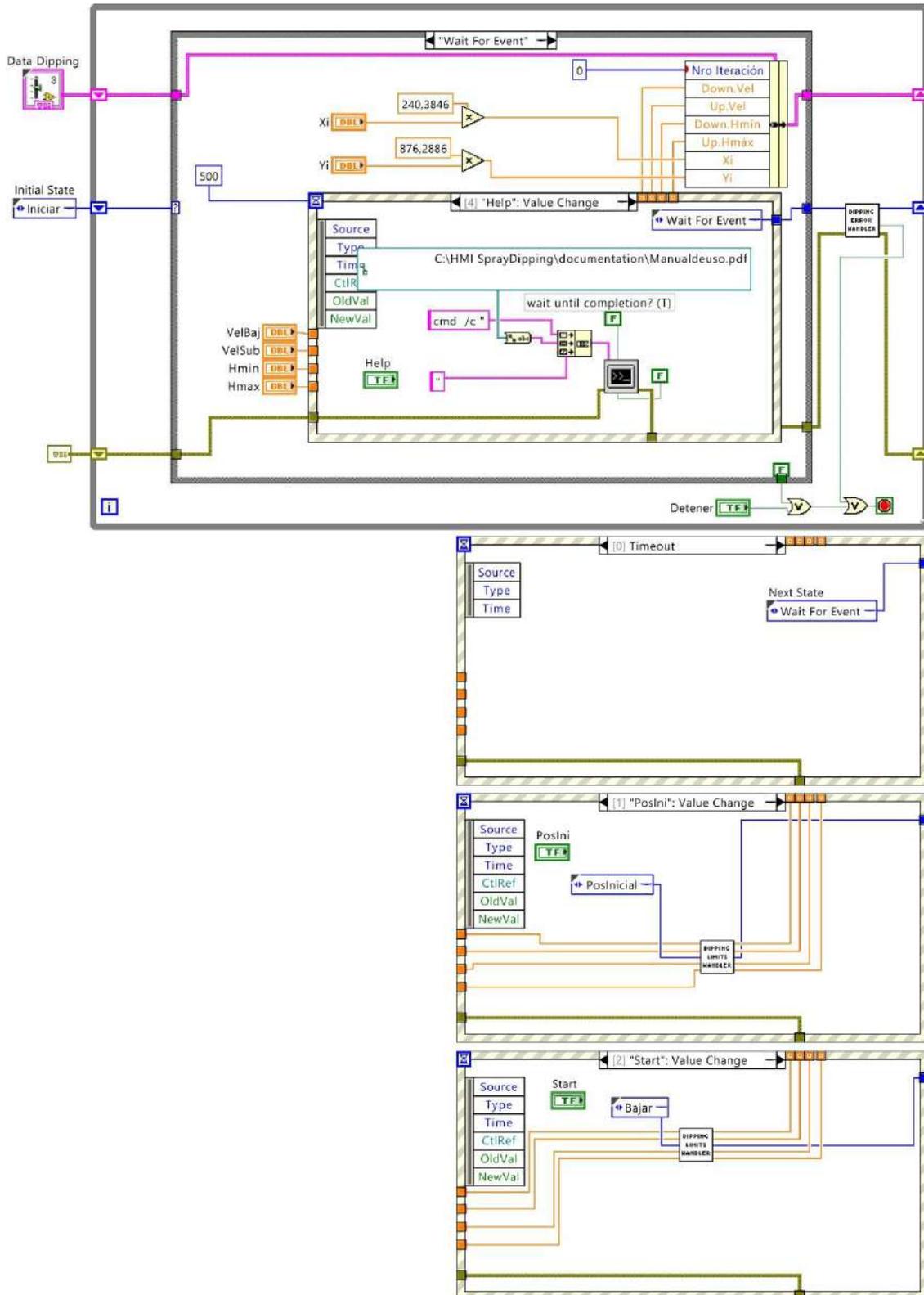
Iniciar

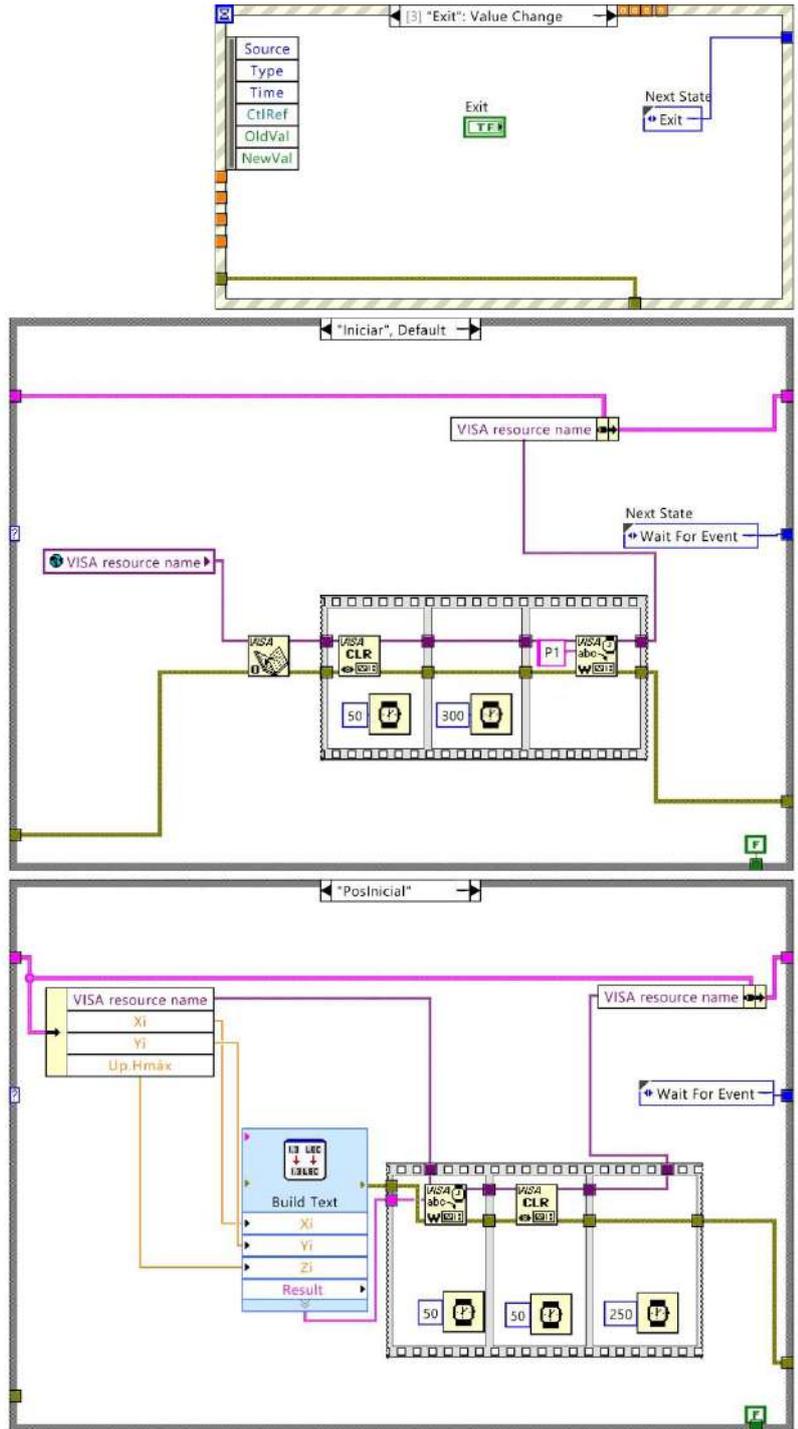
Detener

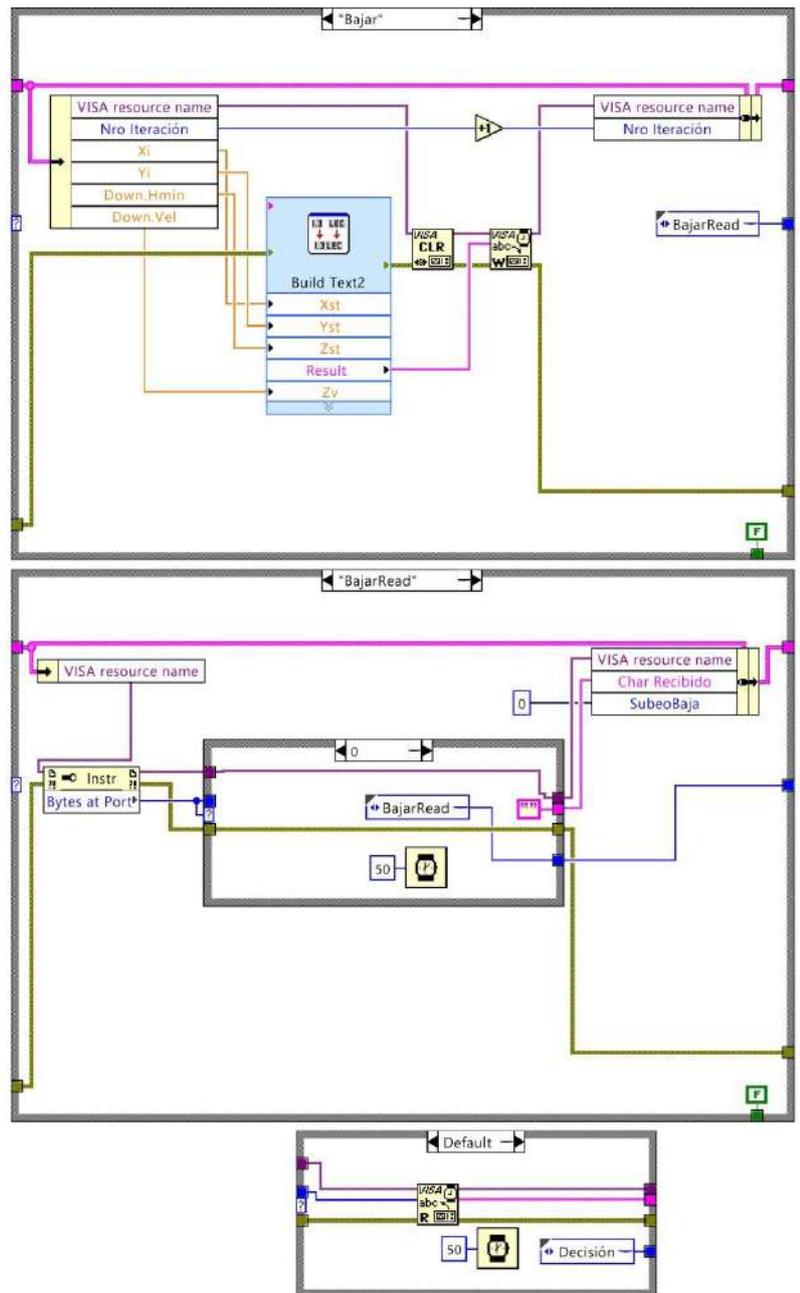
Salir

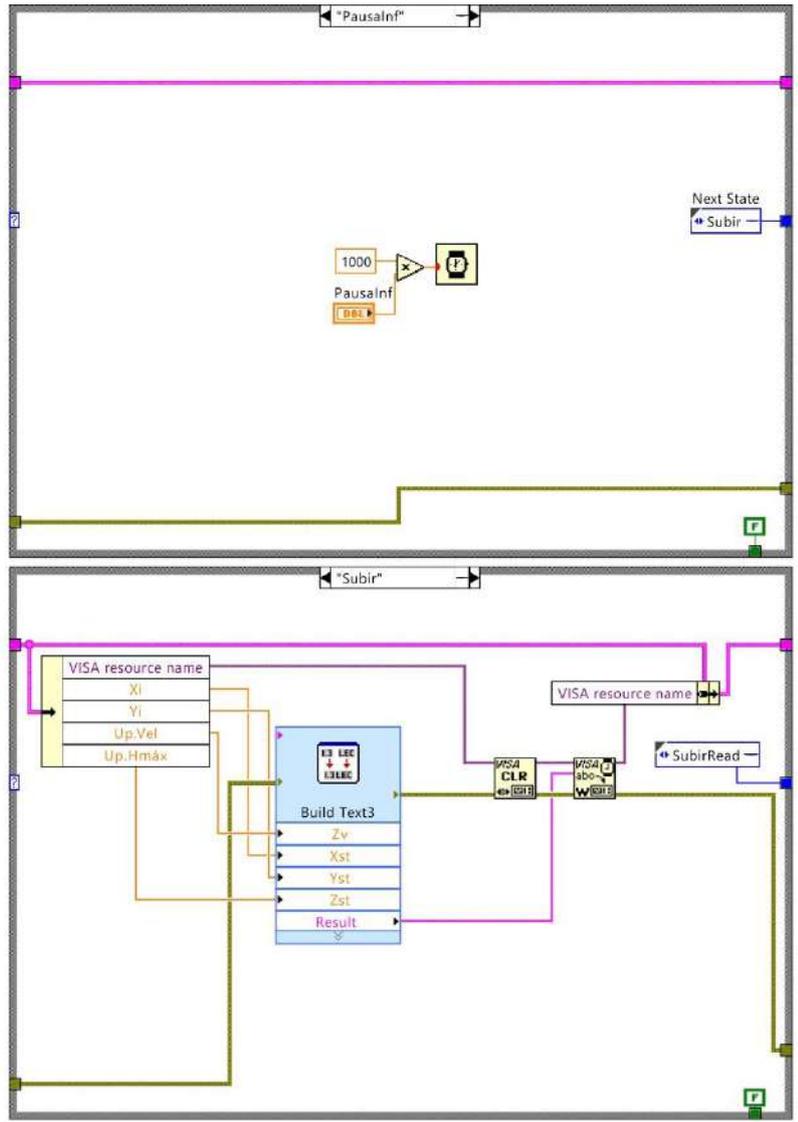
Help

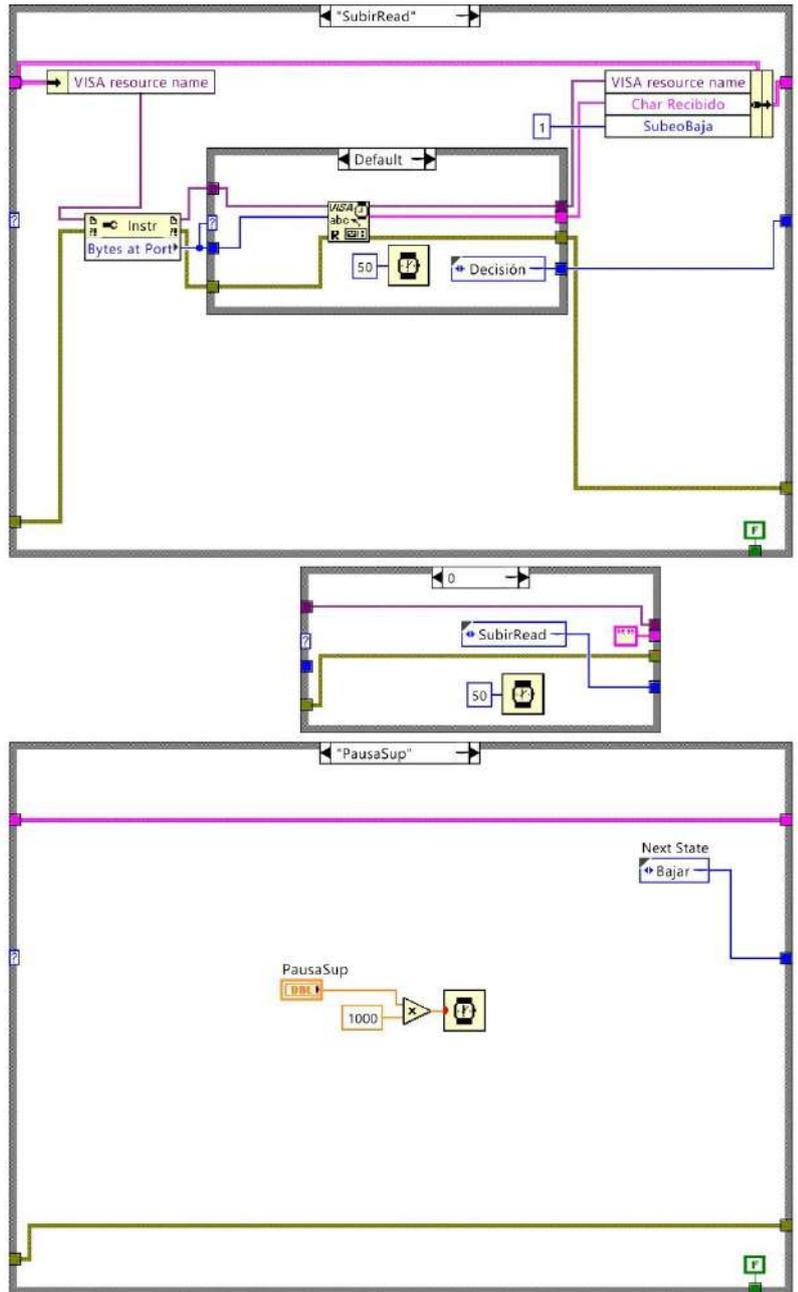
“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”

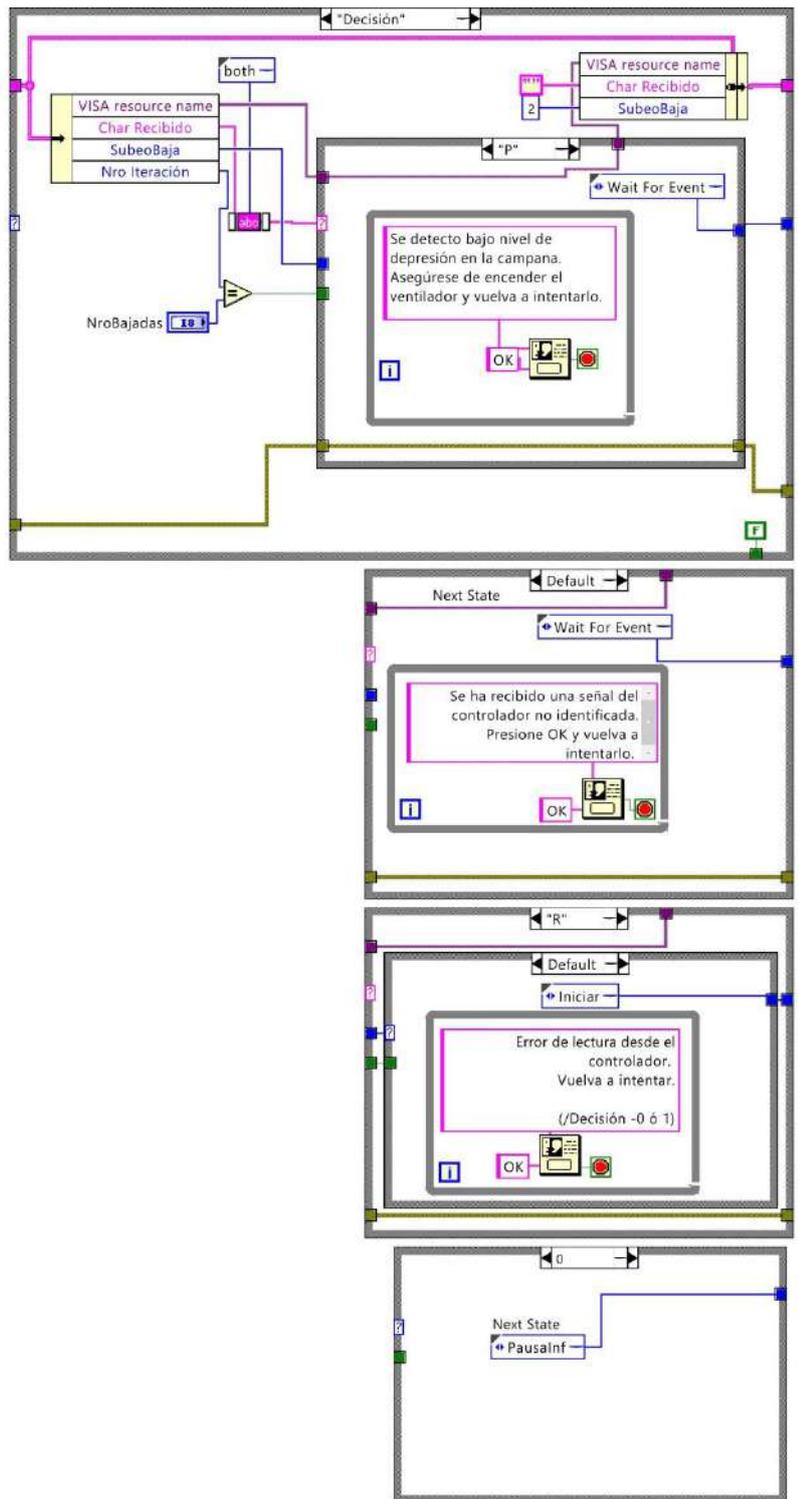


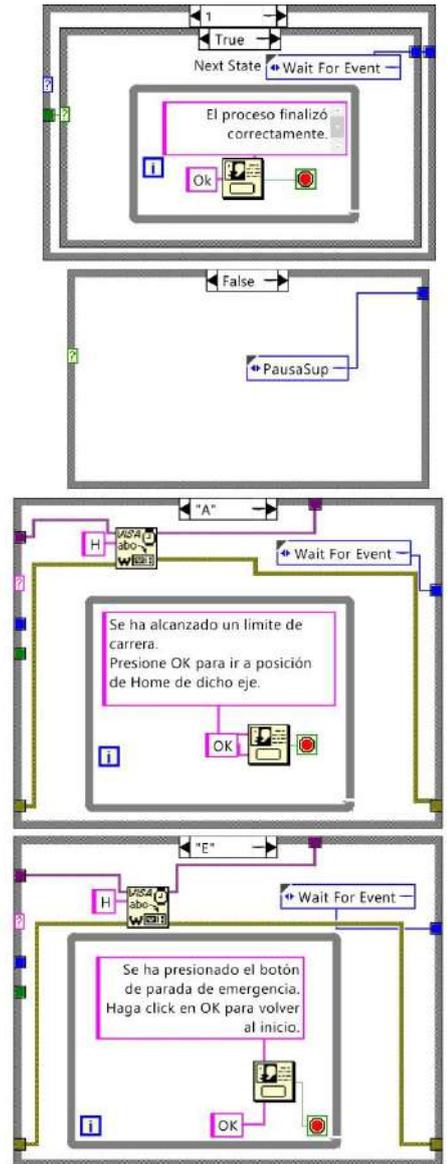


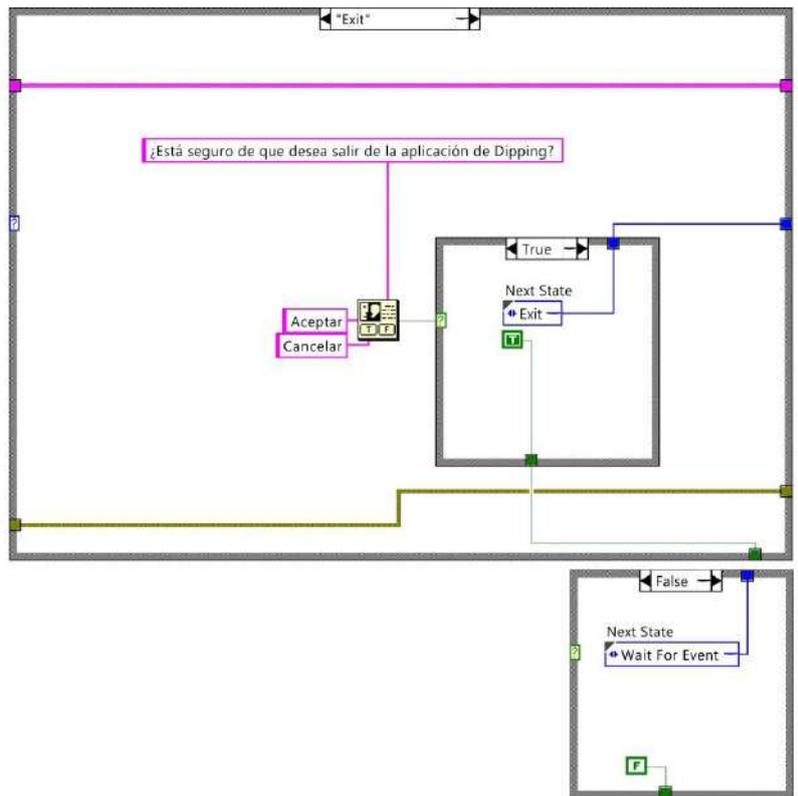






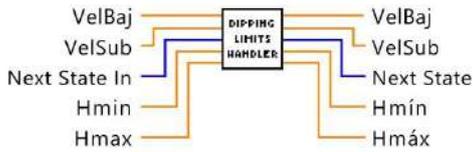




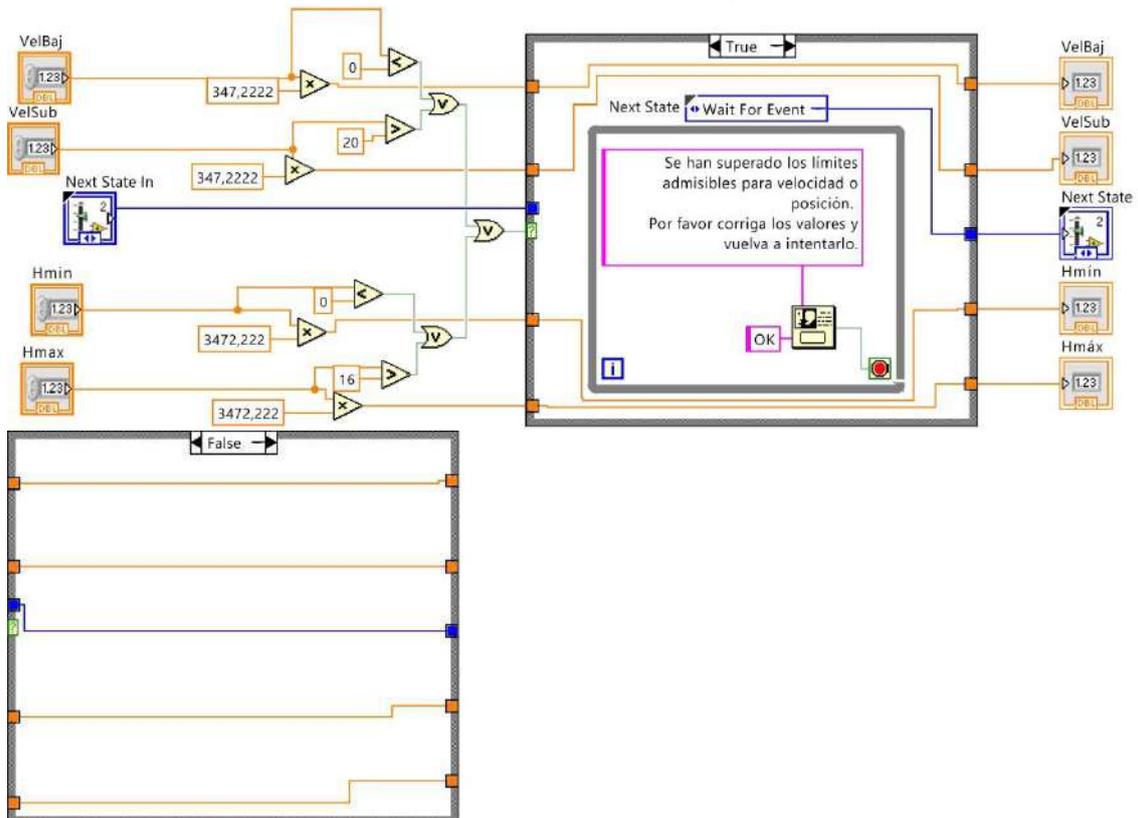
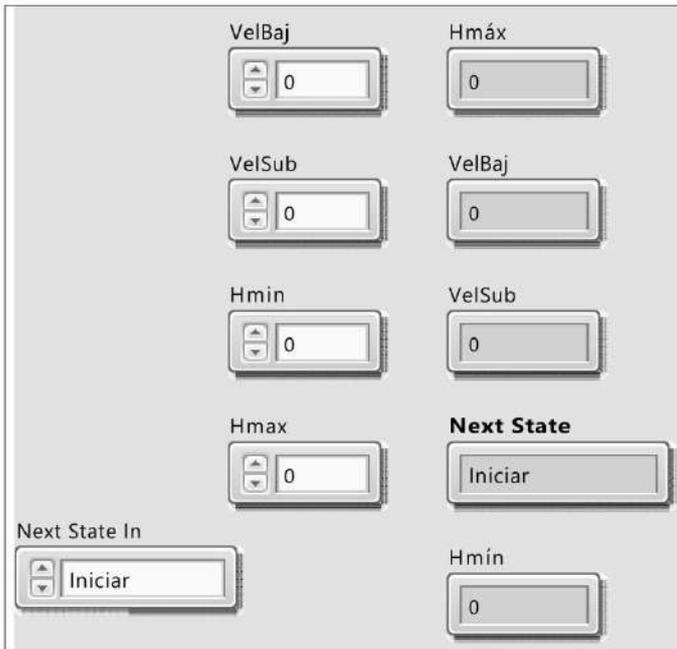


B.2.5 - SubVI para manejo de datos numéricos ingresados

Dipping Limits Handler.vi



SubVI encargado de verificar si los valores ingresados por el usuario están de acuerdo a los admisibles por la máquina.

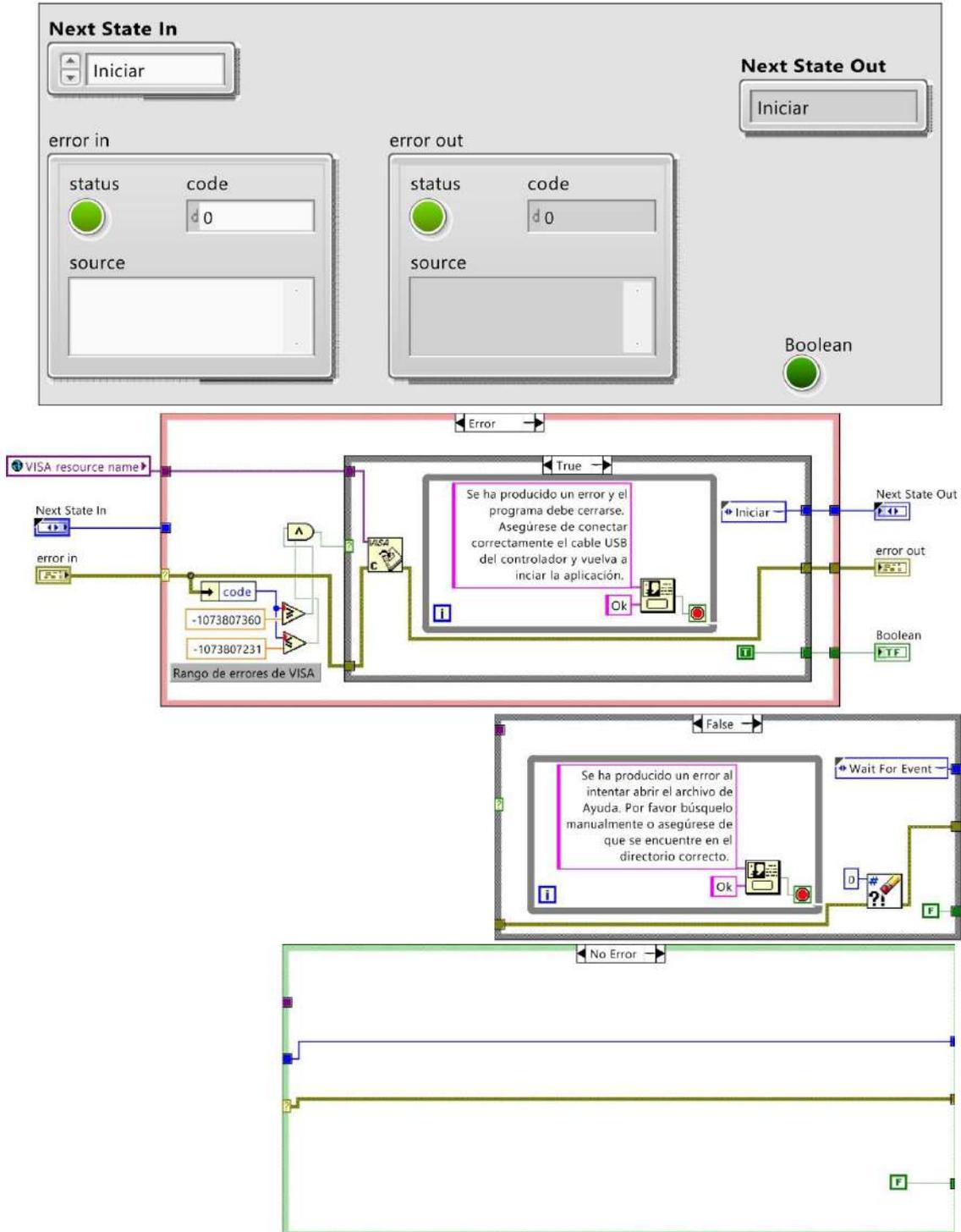


B.2.6 - SubVI para manejo de errores

Error Handler Dipping (SubVI).vi



SubVI para el manejo de errores en la ejecución de la aplicación para depósito por dipping.



B.3 - Manual de uso de software

MANUAL DE USO DE APLICACIONES DE LA MÁQUINA DEPOSICIONADORA

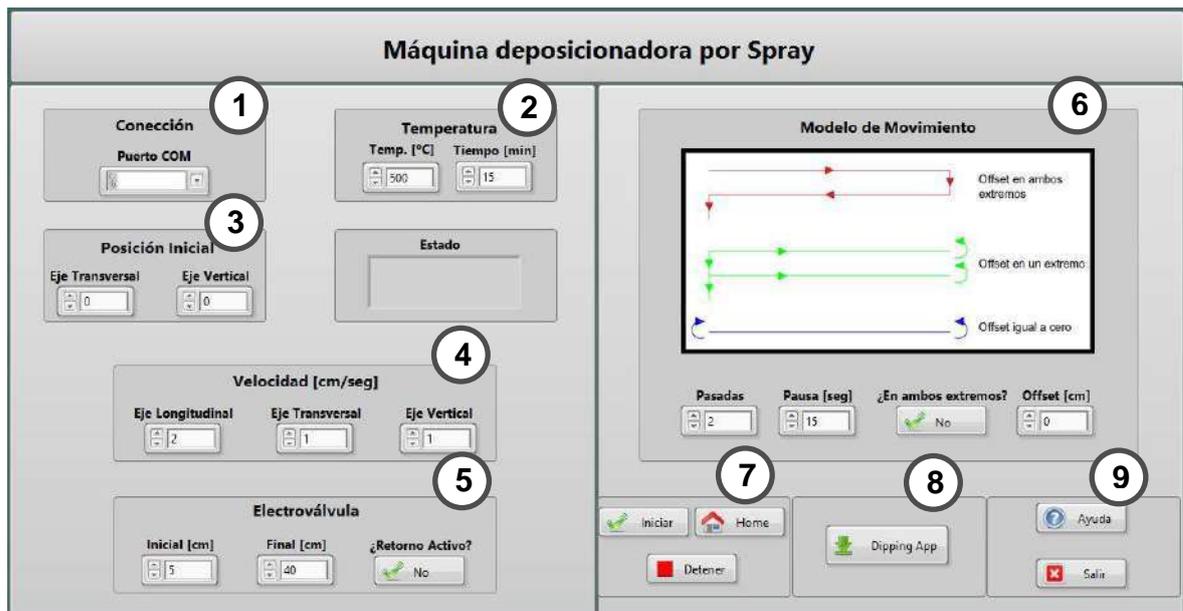
Contenido

1. Manual de Uso de aplicación para deposición por Spray.....	2
Botones y controles del Panel Frontal.....	2
1.1 Configurar comunicación con la controladora.....	2
1.2 Configurar parámetros de la placa calefactora.....	2
1.3 Configuración de la posición inicial.....	3
1.4 Configurar parámetros de velocidad.....	3
1.5 Configurar parámetros de electroválvula.....	3
1.6 Configuración del modelo de movimiento.....	3
1.7 Botones de función.....	4
1.8 Botón para iniciar aplicación de “Dipping”.....	5
1.9 Botones de Ayuda y de Cierre de aplicación.....	5
2. Manual de uso del entorno gráfico de Dipping.....	6
2.1 Configuración de coordenadas iniciales.....	6
2.2 Configuración de velocidades.....	6
2.3 Configuración de alturas máxima y mínima.....	7
2.4 Configuración de la secuencia del proceso.....	7
2.5 Botón de posición inicial.....	7
2.6 Botones de control de la máquina.....	7

1. Manual de Uso de aplicación para deposición por Spray

A continuación encontrará la información necesaria para hacer uso de la máquina con la finalidad de realizar Deposición por Spray.

Botones y controles del Panel Frontal



1.1 Configurar comunicación con el controlador

- Asegúrese de conectar el cable USB entre el Gabinete y la PC.
- Seleccione el Puerto COM que debería de indicarle el programa, en caso contrario, asegúrese de que se ha instalado correctamente el driver del controlador de la máquina.

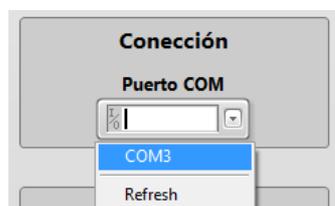


Figura 2 – Correcta selección del puerto COM

1.2 Configurar parámetros de la placa calefactora

- En el primer control debe marcar la temperatura deseada (entre 0 y 500°C).

- El control de tiempo es una pausa programable para no comenzar hasta que se haya transcurrido ese tiempo. El sistema de control no es realimentado, por lo tanto **se recomienda 15 minutos** (Valor predeterminado) para alcanzar la temperatura máxima.

1.3 Configuración de la posición inicial

- Estos controles están disponibles para modificar las coordenadas del punto inicial desde el cual se iniciarán los barridos de spray. Las unidades son en centímetros (cm). Comenzando por el punto inferior en el eje vertical, y el posterior o frente en el transversal.
- La coordenada vertical se mantendrá constante a lo largo de todo el proceso. La coordenada transversal será constante a menos que se elija un modelo de movimiento diferente tal como se mencionará más adelante.

1.4 Configurar parámetros de velocidad

- Las velocidades serán constantes a lo largo de todo el recorrido planteado.
- Los valores mínimos y máximos se enumeran en la siguiente tabla:

Eje	Velocidad Mínima [cm/seg]	Velocidad Máxima [cm/seg]
Longitudinal	0,5	10
Transversal	0,25	2
Vertical	0,25	2

Tabla 1 – Velocidades máximas y mínimas para cada eje

1.5 Configurar parámetros de electroválvula

- Se presentan tres controles: posición de encendido, posición de corte, y selección de retorno activo.

En los primeros dos se deben indicar las coordenadas del eje longitudinal que serán extremos del funcionamiento de la electroválvula (spray sobre las muestras). Por otro lado, si selecciona retorno activo (SI), el spray se producirá tanto en el barrido de ida como de vuelta. Si

el mismo se encuentra apagado (NO), entonces solo se hará fluir spray en el barrido de ida.

1.6 Configuración del modelo de movimiento

- En esta sección se permitirá determinar tanto el recorrido que debe realizar la máquina, como las pausas intermedias

Para ello se dispone de un control numérico de pausas (en segundos), que se cumplirá en cada extremo de la máquina.

- Por otro lado, con el manejo de los otros controladores se puede determinar si se realizará un desfasaje (Offset) con respecto a la posición inicial del recorrido, y si el mismo será en ambos extremos o sólo en el inicial o de “Home”, tal como se puede visualizar en la imagen a continuación:

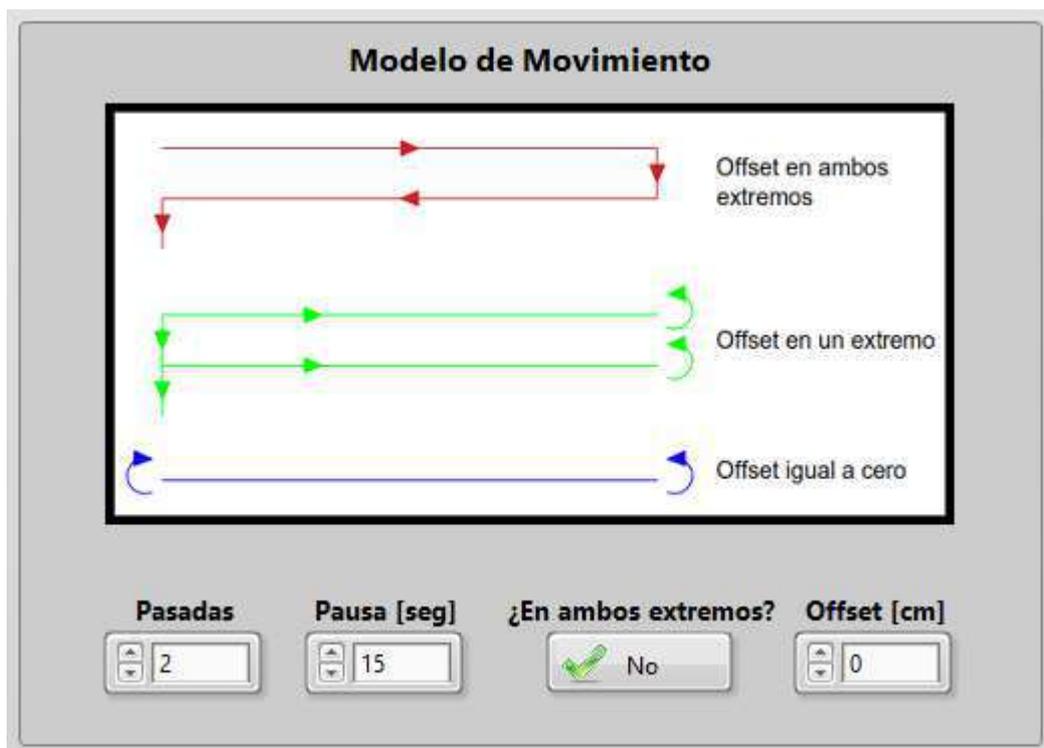


Figura 2 – Panel de configuración del modelo de movimiento a realizar.

Es de considerar que los valores de la imagen son los seleccionados como predeterminados. Es decir, que se realizará una ida y vuelta tal como lo muestra la imagen azul, con una pausa de 15 segundos entre barridos.

Se debe tener en cuenta que el software no permitirá configurar un modelo de movimiento en el cual se excedan los límites de recorrido para cada uno de ellos. Es por ello que en caso de exceder alguno de los límites el programa informará para revisar los mismos y cambiarlos. En la siguiente tabla se muestran los valores máximos a recorrer en los ejes transversal y vertical (en ésta aplicación la máquina siempre recorrerá de un extremo al otro en la dirección longitudinal).

Eje	Distancia máxima admisible [cm]
Transversal	20
Vertical	16

Tabla 2 – Distancias máximas permitidas en aplicación de Spray.

1.7 Botones de función

- Estos tres botones permiten al usuario iniciar o detener las diferentes secuencias.
 - El botón de iniciar (Run) hará que el controlador comience con la secuencia de movimiento configurada previamente por el usuario.
 - El botón de detener (Stop) detendrá la secuencia de envíos del controlador y por lo tanto la máquina se detendrá. Se debe considerar que para que se interrumpa la comunicación se deberá cerrar el programa y por lo tanto luego debe volver a ingresar al mismo.
 - El botón de Home, es el encargado de llevar al cabezal de la máquina a una posición reconocida como Home (Hogar – Casa). Éste es necesario en aquellos casos en que se interrumpa el movimiento accidentalmente y la máquina pierda referencias de dónde está localizada.

1.8 Botón para iniciar aplicación de “Dipping”

- El botón “Dipping App” abrirá una nueva ventana con un programa similar al ya presente, pero específicamente para realizar el proceso de “Dipping” (Introducir y quitar muestras de sustancias a velocidades

“Diseño y construcción de una máquina de control numérico para la realización de recubrimientos superficiales por depósito de spray e inmersión.”



controladas). Dicho panel frontal será explicado al final de este documento.

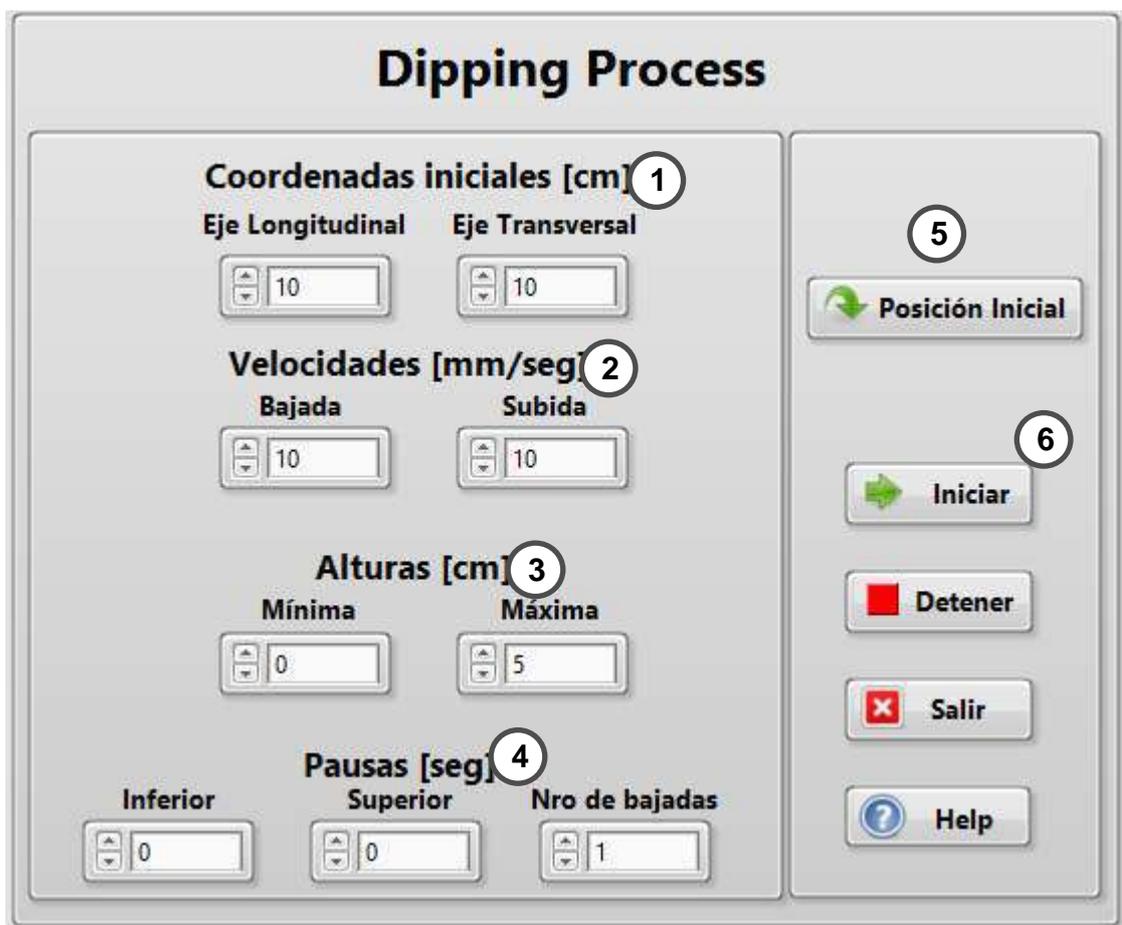
1.9 Botones de Ayuda y de Cierre de aplicación

- El botón de ayuda (Help) será el que habrá el Manual de uso del software.
- El botón de cierra (Exit) le permitirá al usuario cerrar la aplicación luego de ser confirmado por el usuario.

2. Manual de uso del entorno gráfico de Dipping

En ésta sección del archivo se le explicará cómo utilizar el entorno gráfico desarrollado exclusivamente para realizar el proceso de “Dipping”.

A continuación se muestra una imagen con el panel frontal que verá el usuario:



2.1 Configuración de coordenadas iniciales

- Este panel de control permite al usuario realizar el “Dipping” en cualquier punto específico dentro del rango de movimiento del cabezal. Permitiendo mayor comodidad para controlar el proceso y/o tener más de una muestra e ir alternando el proceso entre ellas (no automáticamente).

2.2 Configuración de velocidades

- A diferencia del proceso de Spray, en ésta aplicación solo se permite al usuario configurar la velocidad de interés, es decir, la velocidad de ingreso y egreso de la muestra a la sustancia. La misma se debe configurar en **milímetros por segundo** de avance, y se permite que la de bajada sea diferente a la de subida.

2.3 Configuración de alturas máxima y mínima

- Este módulo de control ha sido incluido para poder determinar cuál es la altura mínima hasta la que se debe bajar la muestra y cuál será la máxima para realizar el movimiento alternativo. La carrera puede tener un máximo de 120 mm, la mínima (inferior) será 0 [mm] y la máxima por lo tanto 120 [mm].

2.4 Configuración de la secuencia del proceso

- Aquí se podrá determinar el valor de las pausas (en segundos) superior e inferior, permitiendo que las mismas sean diferentes. Por otro lado, se podrá ingresar el valor del número de bajadas que realizará la muestra dentro del a sustancia.

2.5 Botón de posición inicial

- Éste permite al usuario enviar al cabezal a la posición inicial para poder ubicar el recipiente debajo y asegurarse de que las coordenadas iniciales ingresadas son las correctas.

2.6 Botones de control de la máquina

- La serie de botones que se encuentran en ésta sección permiten al usuario comenzar, detener o cerrar la aplicación de Dipping.

Para mayor información acerca del programa, consulte en el informe del trabajo de grado cuyo objetivo fue la creación de ésta máquina y su correspondiente entorno gráfico.

Autores: Coronel Martín martin.coronel91@gmail.com

Iriani Matías matiriani@gmail.com

Carrera: Ingeniería Electromecánica.

Director: Gustavo Uicich.

Fecha: Junio de 2015.

Anexo C - ELECTRÓNICA

En este apartado se enumerarán los datos técnicos correspondientes a los principales elementos de electrónica utilizados en nuestro proyecto, los cuales son, hoja de datos y esquemático de driver Polulu de los motores PAP, de la shield CNC v3 y de la placa distribuidora diseñada para simplificar conexiones.

C.1 - Driver Polulu a4988

Este controlador está basado en el chip A4988 de Allegro cuya hoja de datos puede ser encontrada en la página web ¹⁰del fabricante, la cual debido a su elevada cantidad de contenido no ha sido incluida en este apartado. En la figura C.1 se muestra el esquemático del circuito del controlador y en la figura C.2 se muestra el diagrama de conexionado del mismo con la fuente de alimentación, el procesador y el motor.

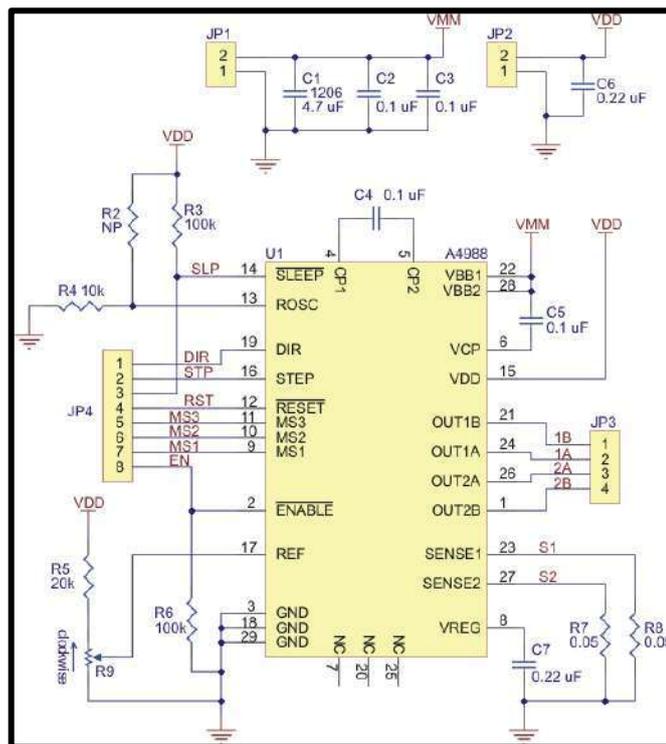


Figura C.1 – Esquemático del circuito del controlador.

¹⁰ <https://www.pololu.com/product/1182/specs>

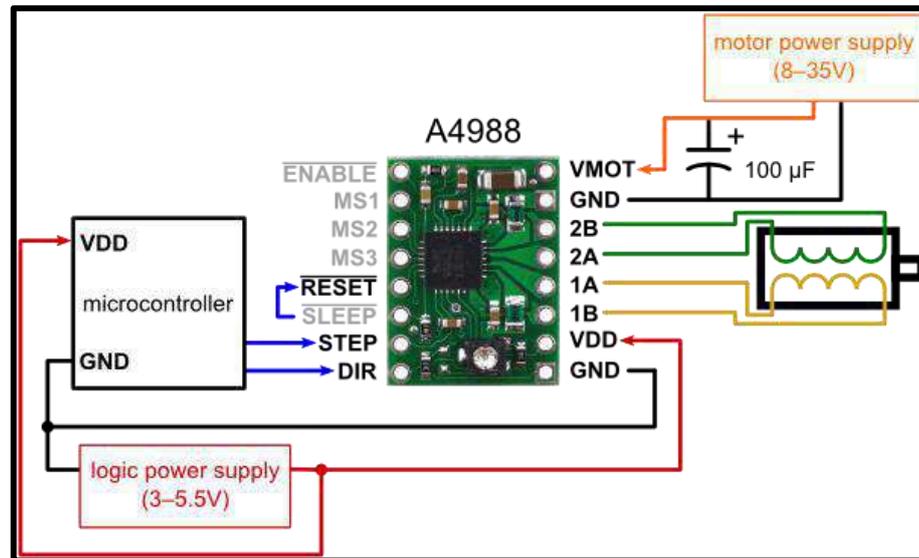


Figura C.2 - Diagrama de conexionado del controlador con la fuente de corriente, el procesador y el motor.

C.2 - Shield CNC v3.0

En la figura C.3 se muestra el esquemático del circuito que posee la Shield CNC v3.0 utilizada para montar los controladores de los motores.

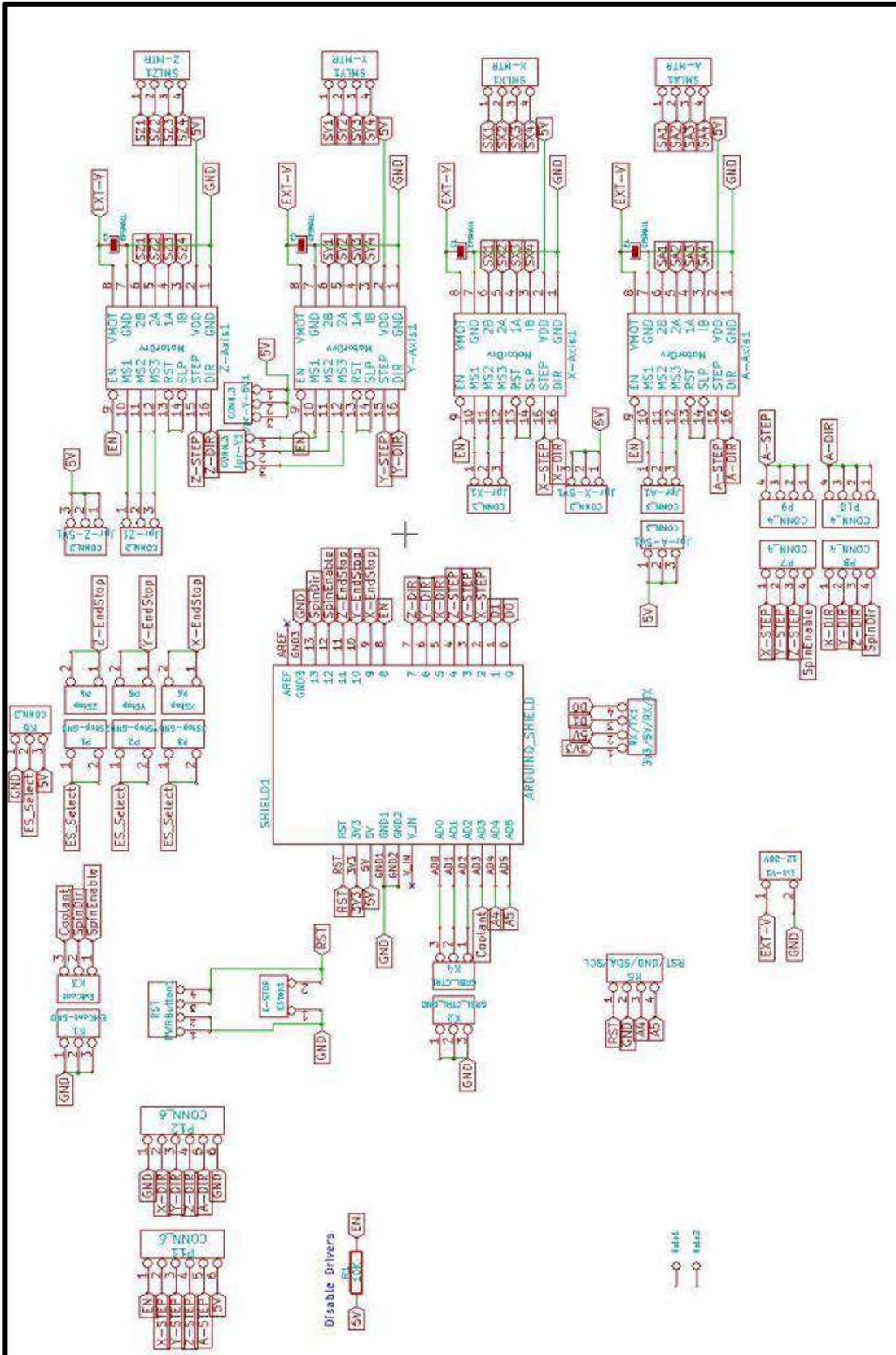


Figura C.3 - Esquemático del circuito de la Shield CNC v3.0

C.3 - Placa distribuidora

Como se explica en el capítulo 4 de hardware de control, se diseñó una placa para que cumpla la función de aislar el circuito del controlador respecto del circuito de potencia que maneja la electroválvula y la placa calefactora, y también para que sirva de placa de “distribución” facilitando el conexionado de los distintos circuitos que controlan la máquina.

Esta placa consta de un conector principal de 34 pines, el cual permite la conexión del cable proveniente de la Shield CNC desde donde se accede a las entradas y salidas del controlador Arduino. En la figura C.4 se muestra el esquemático del circuito de la placa distribuidora, en donde se puede apreciar el vínculo entre los conectores de los sensores tales como finales de carrera (X-GND Y Y-Z), Reset y Switch (sensor de presión diferencial) y el conector del Arduino.

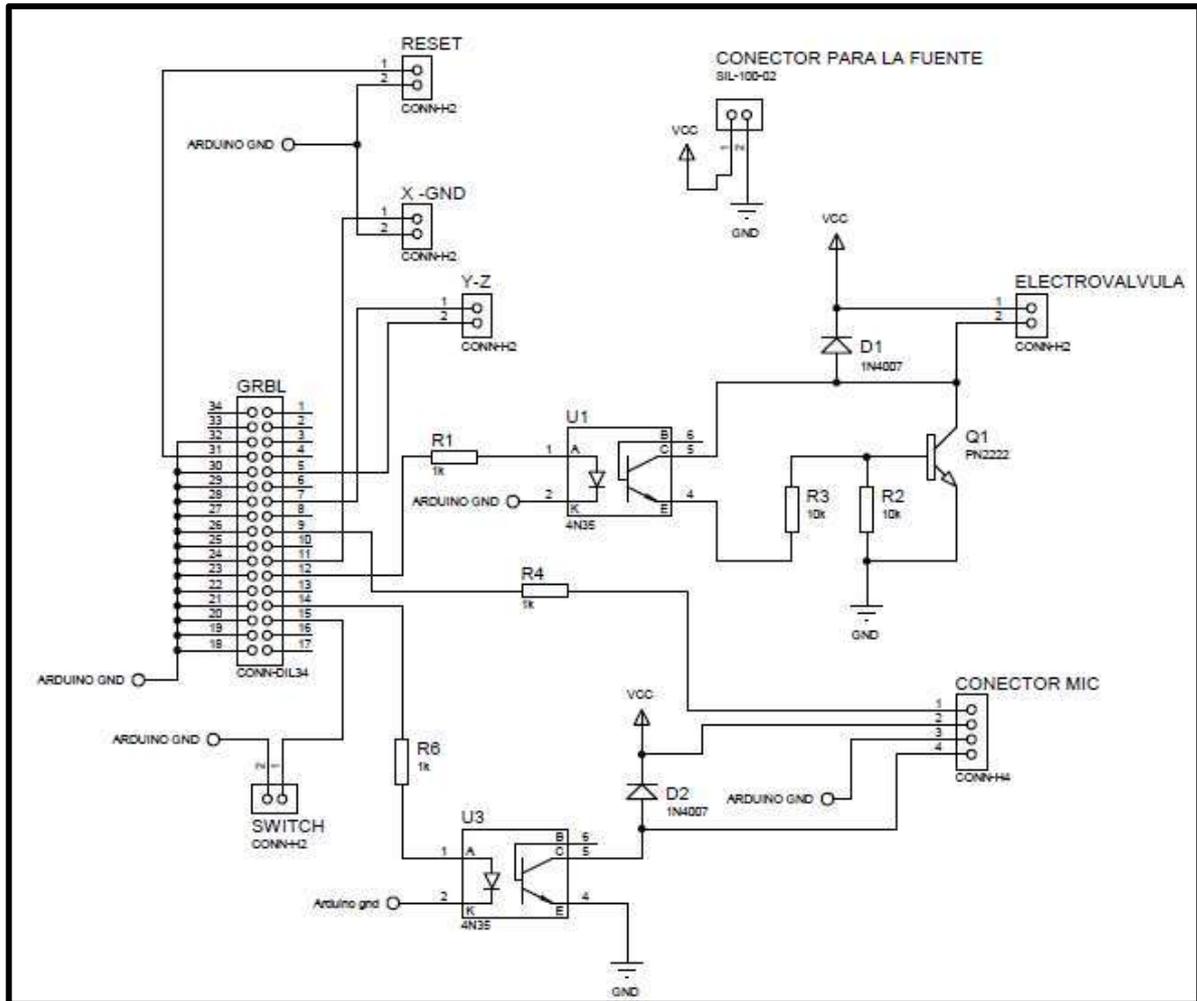


Figura C.4 - Esquemático del circuito de la placa distribuidora.

Por otra parte el circuito posee también dos subcircuitos de aislación, en donde mediante optoacopladores 4N35 se aísla el Arduino de la electroválvula y del relay que alimenta la placa calefactora, ambos artefactos alimentados con 24 V. En el caso de la electroválvula también se debió utilizar un transistor PN2222 para poder aumentar la corriente de salida del circuito sin quemar el optoacoplador, ya que la corriente de salida máxima del mismo es de 40 mA y la electroválvula consume 200 mA. Debido a la carga inductiva de las bobinas de la electroválvula y del relay fue necesario colocar un diodo 1N4007 en paralelo con cada una de manera que al desenergizarlas, la corriente pudiera continuar fluyendo hasta disipar la energía magnética almacenada.

Anexo D - ELECTROVÁLVULA

En este apartado se pretende mostrar la hoja de datos y especificaciones de la electroválvula utilizada, luego de una breve descripción del funcionamiento de la misma.

D.1 - Marco teórico

Posee dos partes fundamentales, el solenoide y la válvula.

El solenoide convierte la energía eléctrica en energía mecánica mediante la creación de un campo electromagnético, que por magnetismo desplazará un vástago ferromagnético que hará actuar a la válvula.

Las válvulas son elementos que permiten el paso de un fluido cerrando u abriendo un canal de paso para el mismo entre el conducto de entrada y el de salida. Existen diferentes tipos de válvulas, desde las simples válvulas de paso, con una entrada y una salida, hasta de mayor complejidad donde se permite el retorno luego del accionamiento del actuador, paso a dos salidas diferentes, o incluso alternar el paso de una entrada entre dos o más salidas.

En nuestro proyecto simplemente controlamos el paso o no de fluido a través de un conducto, por lo que evitamos controlar caudal, presión, etc. Por lo tanto utilizamos una electroválvula de paso, que alterna de la imagen de la derecha e izquierda de la figura D.1 al accionar el solenoide:

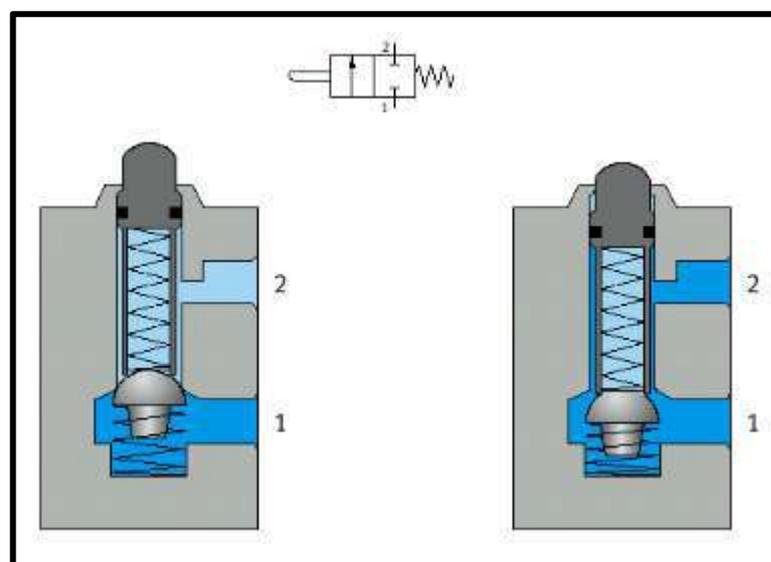


Figura D.1 - Esquema de válvula de dos vías y dos posiciones (2/2).

En la figura de la izquierda puede observarse su posición de normal cerrado, determinada por el resorte, y que al accionar, el vástago es empujado hacia abajo, permitiendo el paso de fluido desde el punto 1 al 2.

Para representar las válvulas se emplean esquemas y símbolos que indican el número de vías de entrada y salida, posiciones que pueden tener y como se realiza su accionamiento.

El esquema puede corresponder a dos normas diferentes:

- CETOP: Comité europeo de Transmisión Oleo hidráulica y Pneumática.
- ISO 1219: Organización Internacional de Normalización.
- DIN 24300: Instituto Alemán de Normalización.

Las válvulas direccionales son siempre representadas por un rectángulo, y éste dividido en cuadrados. El número de cuadrados es igual al número de posiciones que puede adoptar la válvula, representando la cantidad de movimientos que efectúa a través de los accionamientos. (Ver figura D.2)



Figura D.2 - Representación de posiciones de la válvula.

El funcionamiento y paso de fluido para cada posición se representa mediante un esquema dentro de cada cuadrado. Las flechas indican el sentido de circulación y las bocas cerradas se representan con líneas transversales.

En la figura D.3 se encuentra el esquema que debería tener la electroválvula de dos vías y de cierre o apertura según nuestra utilización:

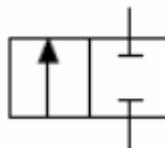


Figura D.3 - Esquema de válvula de dos vías y dos posiciones.

En esa representación, cuando se encuentra en la posición de la izquierda, se permite el paso de fluido de un orificio al otro. En cambio, cuando se encuentra en la posición de la derecha, la válvula no permitirá el paso de fluido ya que se encuentra cerrada.

Los orificios de las válvulas se representan mediante números o letras dependiendo de la norma adoptada. En la tabla D.1 se muestra el significado de cada letra que representa a las vías, y en la tabla D.2 puede verse la equivalencia entre la norma DIN 24300 y la norma ISO 1219.

Tabla D.1 - Significado de letras asignadas a vías según norma DIN 24300.

Vías	Significado
A, B, C	Tuberías o conductos de trabajo
P	Entrada de presión
R, S, T	Salidas de escape
X, Y, Z	Tuberías o conductos de pilotaje

Tabla D.2 - Equivalencia entre normas DIN 24300 e ISO 1219.

	Norma DIN 24300			Norma ISO 1219		
Presión	P			1		
Utilización	A	B	C	2	4	6
Escape	R	S	T	3	5	7
Pilotaje	X	Y	Z	10	12	14

Otro tipo de clasificación es el tipo de mando eléctrico, el cual está representado a la izquierda del rectángulo donde se indican las posiciones, mediante los símbolos que se observan en la figura D.4.

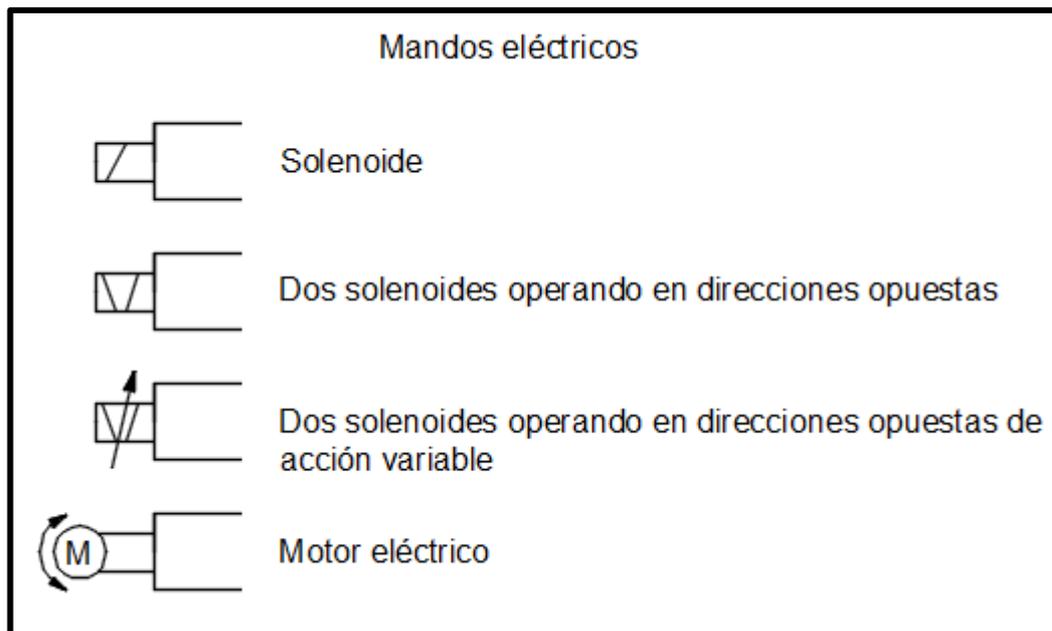


Figura D.4 - Representación de mando eléctrico.

D.2 - Especificaciones

En nuestro caso, la electroválvula utilizada es como la que se muestra en la figura 4.10 del informe. Se trata de una válvula de 5 vías y dos posiciones. Este tipo de válvulas es comúnmente utilizado para comandar cilindros de doble efecto en automatización neumática, el proveedor no pudo facilitarnos más información de esta electroválvula que la que figura en el cuerpo de la misma. De allí se pueden obtener las siguientes características:

Tabla D.3- Características nominales de la electroválvula.

Tensión nominal	24V
Tipo de corriente	DC
Potencia	4.8W
Grado IP	65
Rango de presiones de trabajo	0.15 – 0.8 Mpa
Tipo de válvula	5/2

Anexo E - SENSOR DIFERENCIAL DE PRESIÓN

El sensor utilizado es similar al Teb930 y tiene como principal aplicación el monitoreo de la caída de presión producida en un filtro de aire (comúnmente utilizados en ductos de ventilación). La diferencia de presión medida es aplicada en ambos lados de un diafragma que se encuentra dentro del sensor conectado al borne C (común). Este diafragma es empujado permanentemente por un resorte cargado contra una pared metálica conectada al borne NC (normal cerrado) y cuando la diferencia de presión es suficiente, el diafragma se mueve y hace contacto con otra pared metálica conectada al borne NA (normal abierto) del sensor. En la figura E.1 se muestra un interruptor similar al utilizado.

Entre sus características posee un rango de diferencia de presión que va de 30 a 300 Pa, máximo voltaje 250VAC y máxima corriente 1.5 A.



Figura E.1 - Sensor de presión diferencial TEB930.

Anexo F - MOTORES PASO A PASO

A la hora de seleccionar los motores para nuestra aplicación nos vimos obligados a motores Paso a paso por su facilidad de control y la precisión que manejan. Para ello tuvimos que investigar acerca de los mismos y así poder hacer una correcta selección de la cual se hablara más adelante. A continuación se muestra una introducción en donde se presentan algunos conceptos importantes que hay que entender para operar con estas máquinas.

F.1 - Introducción

Los motores paso a paso llenan un nicho único en lo que respecta al control de motores en el mundo. Algunos ejemplos de aplicación son en impresoras de chorro de tinta, maquinas CNC y bombas volumétricas. Algunas propiedades de este tipo de maquina la hacen ideal para estos tipos de aplicaciones. Estas propiedades son las siguientes:

- Brushless (sin escobillas) – Los motores paso a paso son Brushless, es decir no requieren de un mecanismo de conmutación como poseen los motores de corriente continua convencionales en donde un sistema de delgas rotantes y escobillas conmutan el devanado rotórico de manera de que en todo momento exista par rotante, sino que en este caso la conmutación se realiza electrónicamente como se explicará posteriormente.
- Independiente de la carga – El motor paso a paso gira a la velocidad configurada independientemente de la carga que posea conectada al eje, siempre y cuando el valor de la carga se encuentre dentro del rango de aplicación del motor.
- Posicionamiento a lazo abierto – El motor paso a paso se mueve en incrementos cuantificados o “pasos”. Mientras el motor funcione dentro de su especificación de torque, la posición del motor es conocida en todo tiempo sin necesidad de un mecanismo de realimentación.
- Torque de mantenimiento o “Holding torque” – Estos motores son capaces de mantener el eje estacionariamente.
- Excelente respuesta de arranque, detención y marcha en reversa.

F.2 - Tipos de motores paso a paso

Existen tres tipos básicos de motores paso a paso en función del tipo de rotor que posean: Imán permanente, de reluctancia variable e híbridos. Los motores de imán

permanente poseen un rotor magnetizado, mientras que los de reluctancia variable tienen rotores de hierro blando de geometría no cilíndrica. Los motores híbridos combinan aspectos de los dos anteriores.

El estator o la parte estacionaria del motor paso a paso contienen múltiples bobinados. El arreglo de estos bobinados es el factor principal que distingue los distintos tipos de motores desde un punto de vista eléctrico. Estos arreglos pueden ser: unipolares, bipolares o bifilares los cuales se explicaran a continuación.

F.2.1 - Motores paso a paso de reluctancia variable

Los motores de reluctancia variables tienen tres o 5 bobinados conectados a un terminal común. La figura F.1 muestra la sección transversal de un motor de reluctancia variable de cuatro bobinados y 15° por paso.

En todo momento el rotor “buscará” alinearse de forma tal que se minimice la reluctancia rotor-estator, circunstancia que se da cuando el espacio entre polos del estator queda lo mas ocupado posible por el material del rotor, es decir, orientando los dientes hacia los polos energizados del estator. Este motor se puede diseñar para tener pasos mas pequeños que los de iman permanente, tambien poseen una gran ventaja que es el bajo momento de inercia del rotor, lo cual varía considerablemente la respuesta dinamica del motor. Entre sus desventajas poseen un menor par motor en comparación con los de iman permanente.

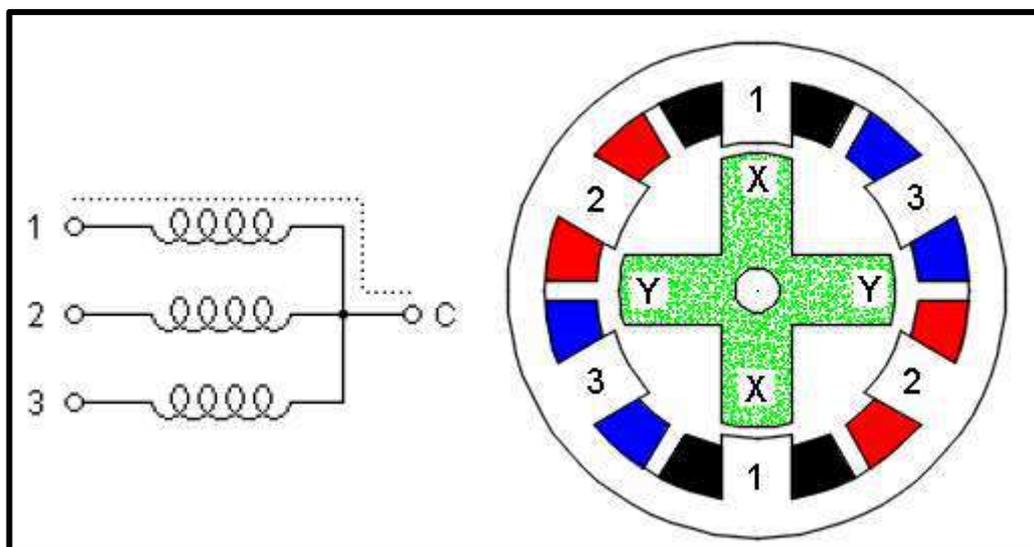


Figura F.1 – Esquema de configuración del rotor y de los bobinados del motor de reluctancia variable.

F.2.2 - Motor paso a paso unipolar

Este tipo de motores está compuesto de dos bobinados, cada uno con un punto medio. Dichos puntos medios son extraídos fuera del motor como dos cables separados o bien son conectados entre sí y extraídos como un único cable. Como resultado, motores unipolares poseen 5 o 6 cables. Luego estos puntos medios son conectados a la fuente de potencia y los conductores que corresponden a las puntas de los bobinados son conectados a masa alternadamente.

En vez de operar minimizando el entrehierro, donde la dirección del flujo de corriente es irrelevante, estos motores operan atrayendo el polo sur o norte del rotor magnetizado permanentemente hacia el polo del estator. Así, en estos motores, la dirección de la corriente a través de los bobinados del estator determinan que polo del rotor será atraído hacia que polo del estator. La dirección de la corriente en motores unipolares depende de qué mitad del bobinado es energizada. Físicamente, las mitades de los bobinados son enrolladas paralelamente entre sí. La figura F.2 muestra la sección transversal de un motor paso a paso unipolar de 30° por paso.

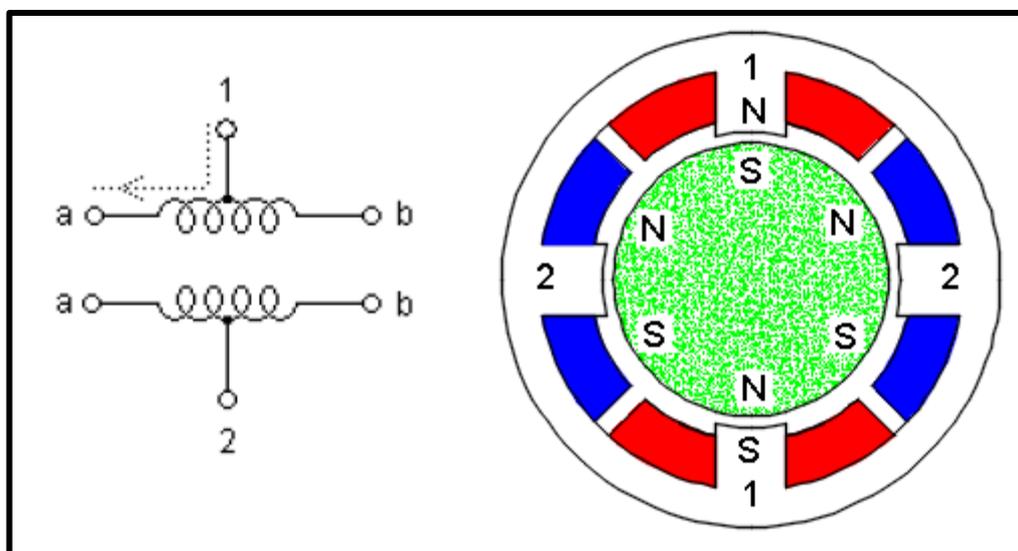


Figura F.2 - Esquema de configuración del bobinado y del rotor del motor unipolar.

F.2.3 - Motor paso a paso bipolar

Este tipo de motores es el que se utilizó en nuestro proyecto y está compuesto de dos bobinados. A diferencia de motores unipolares, los bipolares no tienen puntos medios. Los motores bipolares producen más torque que los unipolares en el mismo tamaño. La contrapartida es que el circuito de control es un poco más complejo.

El flujo de corriente en el bobinado de un motor bipolar es bidireccional. Esto requiere cambiar la polaridad de cada bobinado. Como se muestra en la figura F.3, la corriente fluirá desde izquierda a derecha en el bobinado 1 cuando 1^a es positivo y 1b es negativo. La corriente fluirá en la dirección opuesta cuando la polaridad en bornes del bobinado sea conmutada. Un circuito de control conocido como puente H, es usado para cambiar la polaridad en bornes de cada bobinado. Todos los motores bipolares tienen dos bobinados por lo tanto dos circuitos de control del tipo puente H debe son necesarios para controlar el motor.

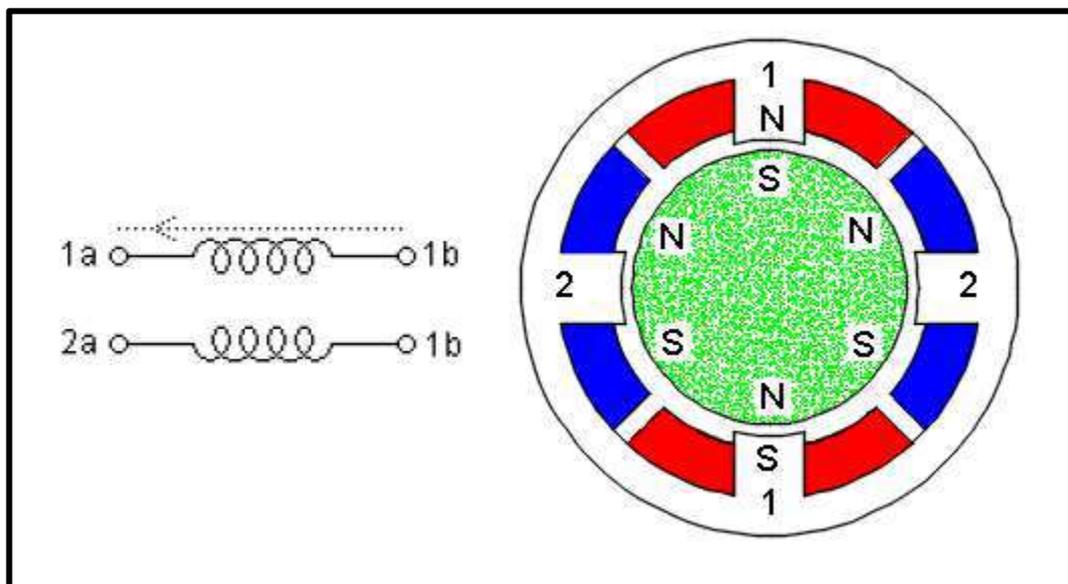


Figura F.3 - Esquema de configuración del bobinado y del rotor del motor bipolar.

F.2.4 - Motores paso a paso bifilares

El termino bifilar significa literalmente “doble bobinado”. Motores con bobinados bifilares son idénticos en rotor y estator a los motores bipolares con la excepción de que cada bobinado está constituido de dos cables enrollados paralelamente. Como

resultado, los motores bifilares poseen ocho cables en vez de cuatro como los bipolares.

Los motores bifilares son comandados de igual manera que los bipolares o los unipolares. Para usarlo como motor unipolar, los dos cables de cada bobinado son conectados en serie y este punto de conexión es utilizado como punto medio. Para usar un motor bifilar como bipolar, los dos cables de cada bobinado son conectados en serie o paralelo.

F.2.5 - Motores paso a paso Híbridos

Motores híbridos comparten los principios operativos de los de imán permanente y los de reluctancia variable. El rotor para un motor híbrido es dentado, como el motor de reluctancia variable y contiene un imán concéntrico alrededor del eje magnetizado axialmente. El diente en el rotor provee un camino que ayuda a guiar el flujo magnético hacia ubicaciones preferidas en el entrehierro. El imán concéntrico aumenta el torque de detención, el de mantenimiento y las características de torque dinámico del motor comparado con los otros motores vistos anteriormente.

F.3 - Características funcionales

Aun cuando el tipo de motor ya es determinado, todavía quedan varias decisiones por tomar para seleccionar un motor en particular. Torque, ambiente de operación, longevidad, tamaño físico, RPM máximas, son algunos de los factores que influyen en la decisión.

Sin lugar a dudas la característica más crítica a la hora de elegir un motor es el torque. Los motores paso a paso poseen diferentes tipos de torque nominales. Estos son:

- Holding torque (torque de mantenimiento) – el torque requerido para mover el eje del motor mientras los bobinados son energizados.
- Pull-in torque – es el torque contra el cual el motor puede acelerar sin perder pasos, siendo controlado a una velocidad constante.
- Pull-out torque – es la carga que el motor puede mover a la velocidad de operación.

- Detent torque – es el torque requerido para mover el eje cuando los bobinados no son energizados.

Los fabricantes pueden especificar algunos o todos estas características en sus hojas de datos.

El torque dinámico, pull-in y pull-out, son una función de la velocidad. Estas torques son importantes para determinar si el motor va a “resbalar” o no para alguna aplicación en particular. Un “resbalamiento” significa que el motor no se mueve cuando debería estar moviendo. En tal caso, el resultado es que el controlador no tendrá conocimiento de la posición del motor y el posicionamiento a lazo abierto falla en este caso. El motor debe estar adecuadamente seleccionado para prevenir esto o bien se deberá caer a un sistema de lazo cerrado en donde, mediante un sensor de posición, se realimente el sistema y así poder saber dónde se encuentra posicionado el motor.

El pull-in torque ofrecido por un motor paso a paso depende fuertemente del momento de inercia de la carga que esta aplicada rígidamente al rotor. Esto hace la tarea de selección un tanto problemática ya que el momento de inercia utilizado por los fabricantes para medir el pull-in torque es raramente brindado en las datasheets y es raramente igual al momento de inercia de la carga conducida en la aplicación.

La mayoría de los fabricantes brindan curvas de torque en sus datasheets. La figura F.4 muestra un ejemplo de una curva de torque para un motor paso a paso:

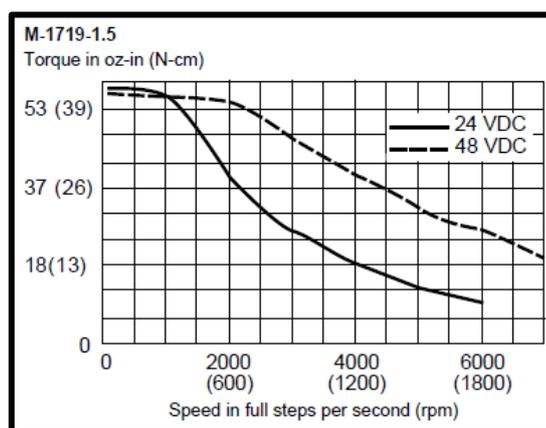


Figura F.4 - Característica Par vs. Velocidad de los motores PAP.

F.4 - Micropasos

Al alimentar el motor en la configuración de simple paso, los movimientos pueden ser un tanto bruscos, sobre todo a bajas velocidades. Micropasos es usado para aumentar la resolución y suavizar las transiciones entre pasos. En muchas aplicaciones, micropasos mejoran la performance de los motores a velocidades próximas a la de resonancia.

El micropasos consiste en el principio de transferir gradualmente la corriente desde un bobinado hacia el otro. Esto es logrado gracias a la modulación del ancho del pulso del voltaje a través de los bobinados del motor. El ciclo de trabajo de la señal que carga un bobinado va decreciendo a la vez que el ciclo de trabajo de la señal que carga el otro bobinado va aumentando.

Hay varios factores que afectan la linealidad del movimiento de un motor real al realizar micropasos, entre ellos, la fricción estática del sistema, la conversión digital analógica de la salida del controlador y la conversión analógica digital del sensor de corriente del sistema.

F.5 - Controlador

El circuito básico para manejar los bobinados de un motor paso a paso bipolar es el puente H, mostrado en la figura X.X.X.6. Un puente H puede ser configurado para permitir el flujo de corriente en ambas direcciones del bobinado. Refiriéndonos a la figura X.X.X.2.6 la corriente fluye a desde izquierda a derecha en el bobinado 1 cuando el MOSFET Q1 y Q4 son encendidos mientras que Q2 y Q3 están apagados. La corriente fluirá desde derecha a izquierda cuando Q2 y Q3 estén encendidos y Q1 y Q4 estén apagados.

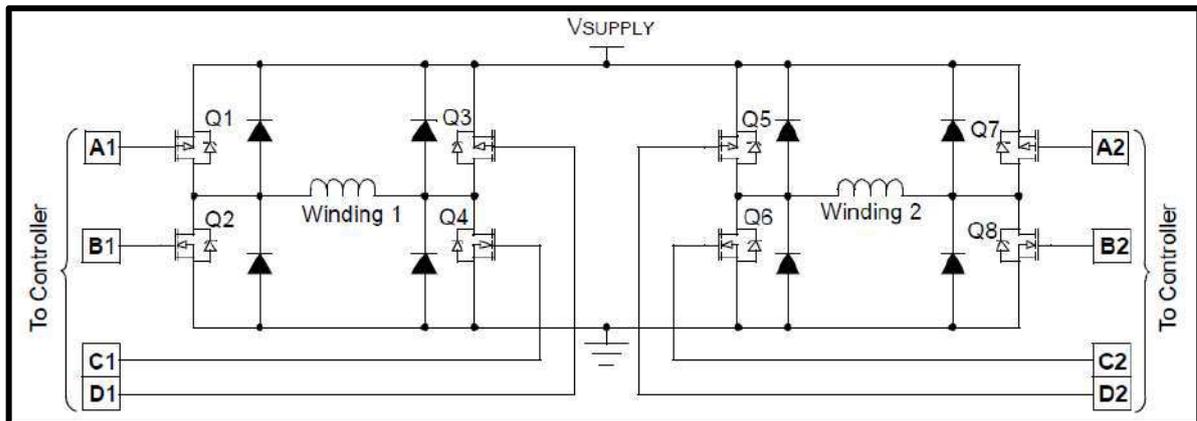


Figura F.5 - Circuito esquemático de los puentes H en un controlador de motores PAP.

Anexo G - PLACA CALEFACTORA

Heizplatten aus CERAN®
mit thermostatischer
Temperaturregelung 50..500 °C

CERAN® Hot Plates
with thermostatic
temperature control 50..500 °C

C 09



HARRY GESTIGKEIT GMBH

Fabrik für Labor-Apparate

Tel. +49 (0)203 74 63 46

Fax +49 (0)203 74 66 37

Postfach 34 01 30

D-40440 Düsseldorf

Angermunderstr. 12

D-40489 Düsseldorf

<http://www.gestigkeit.de>

info@gestigkeit.de



Heizplatten aus CERAN®

CERAN® - eine außergewöhnliche Glaskeramik - ist hoch bruchfest, temperaturwechselbeständig, verzugsfrei, UV-durchlässig sowie hochsäurefest.

Aufbau: Die massearme Platte aus CERAN® ganzflächig elektrisch beheizt, ist in einem Edelstahlrahmen eingebettet und auf einem innenisolierten Edelstahlgehäuse montiert.

Die Temperaturregelung erfolgt über einen elektronischen NiCr-Ni Temperaturregler (Regelbereich 50 ... 500 °C), welcher in einem Druckgußgehäuse aus Aluminium eingebaut ist. Anheizzeit auf 500 °C in ca. 8 Minuten.

Vier, um ca. 8 mm höhenverstellbare Füße, erlauben wackelfreie und waagerechte Aufstellung. Typ 4 A hat 4 Stabstahlfüße (Abstand 100 mm). Bei der Baureihe SR und EB sind die elektrischen Verbindungen zwischen Regelgehäuse und Heizplatte durch einen flexiblen Metallschlauch geschützt.

CERAN® - hot plates

CERAN® an unusual glass ceramic material which is highly resistant to breakage and temperature changes, free from distortion, permeable to ultra-violet light and highly acid-resistant.

Construction: The low mass CERAN® plate is electrically heated over its full surface and is bedded into a stainless steel frame which is mounted on an internally insulated stainless steel housing.

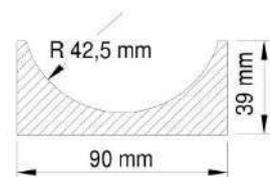
The temperature control system consists of an electronic regulator, activated by a NiCr-Ni temperature sensor (adjustable 50 ... 500°C) installed in a Aluminium pressurecast housing. Warm-up time, e. g. 500°C in 8 minutes. Four feet, which can be individually adjusted to an approx. height of 8 mm to ensure horizontal stability. Type 4A has 4 supporting rod holders (distance 100 mm) In Types SR and EB, the electrical connection between the control housing and the hot plate is protected by a flexible metal hose.

Baureihe A - mit angebautelem Regelgehäuse			Series A - with built-in control housing		
Typ Types	Watt	Volt	B x H x T (mm) W x H x D (mm)	CERAN® mm	Gewicht weight
4 A	1500	230 V, 50-60 Hz	440 x 100 x 300	430 x 140	5,0 kg
11 A	2000	230 V, 50-60 Hz	290 x 100 x 410	280 x 280	5,5 kg
22 A	3000	230 V, 50-60 Hz	290 x 100 x 560	280 x 430	6,5 kg
33 A	4400	3x400 V, N+PE, 50-60 Hz	440 x 100 x 560	430 x 430	9,0 kg
44 A	5700	3x400 V, N+PE, 50-60 Hz	590 x 100 x 560	580 x 430	11,5 kg

Best.-Nr. Ord.-No.	Zubehör für Typ Accessories for
ST 12	Stativstab support rod 12 mmØ, 650 mm
AM 25	Auflagemulde für Rundkolben 250 ml aus Aluminium interchangeable top mulde for round bottom flask 250 ml

Baureihe SR - mit separatem Regelgehäuse			Series SR - with separate control housing		
Typ Types	Watt	Volt	B x H x T (mm) W x H x D (mm)	CERAN® mm	Gewicht weight
4 SR	1500	230 V, 50-60 Hz	440 x 100 x 180	430 x 140	5,0 kg
11 SR	2000	230 V, 50-60 Hz	290 x 100 x 290	280 x 280	5,5 kg
22 SR	3000	230 V, 50-60 Hz	290 x 100 x 440	280 x 430	6,5 kg
33 SR	4400	3x400 V, N+PE, 50-60 Hz	440 x 100 x 440	430 x 430	9,0 kg
44 SR	5700	3x400 V, N+PE, 50-60 Hz	590 x 100 x 440	580 x 430	11,5 kg

Die Leitungslänge zwischen Regler und Heizgerät: 1,3 m. Längere Leitungen gegen Mehrpreis
The conducting wire between control housing and hot plate measures 1,3 m.



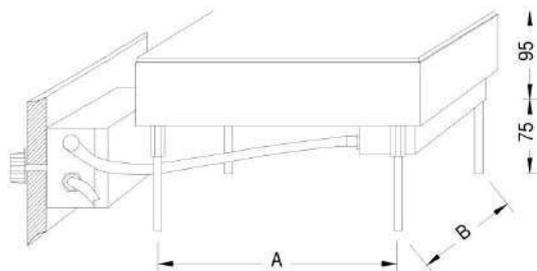
AM 25

Baureihe EB - als Einbaugerät für Tischeinbau				Series EB - as built-in units for bench installation		
Typ Types	Watt	Volt	B x H x T (mm) W x H x D (mm)	CERAN® mm	A x B mm	Gewicht weight
11 EB	2000	230 V, 50-60 Hz	290 x 100 x 290	280 x 280	240 x 240	5,5 kg
22 EB	3000	230 V, 50-60 Hz	290 x 100 x 440	280 x 430	390 x 240	6,5 kg
33 EB	4400	3x400 V, N+PE, 50-60 Hz	440 x 100 x 440	430 x 430	390 x 390	9,0 kg
44 EB	5700	3x400 V, N+PE, 50-60 Hz	590 x 100 x 440	580 x 430	390 x 540	11,5 kg

Die Leitungslänge zwischen Regelgehäuse und Heizgerät: 0,65 m. Längere Leitungen gegen Mehrpreis
The conducting wire between control housing and hot plate measures 0,65 m.

Typ = Best.-Nr.
andere Spannungen auf Anfrage

type = Order-No.
other voltage on request



Herstellungs-Programm

Hochtemperatur-Heizplatte aus TITAN bis 600°C - Präzisions-Heizplatten - Stativ-Heizplatten - Schnellverascher - Mehrzweck-Heizbad mit großer Tauchtiefe - Sandbäder - Serien-Heizgeräte - Mehrzweck-Heizgeräte Original MULTIHITZ - Wasser - und Ölbäder - Entwicklung - Sonderfertigung.

Änderungen, die dem technischen Fortschritt dienen, vorbehalten.

Further Fabrications

High temperature TITANIUM hot plate up to 600°C - Precision hot plates - Stand hot plates with supporting rod - rapid incinerator - Multi-purpose, extremely deep, heating bath - sand bath - Series heating units - Multipurpose heating equipment - Water- and Oil baths - Development - Special orders.

The right to make modification serving further technical development, is reserved