



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO



Felipe Andrés Delgado Rodríguez

Diseño, desarrollo y construcción de una estructura robótica con tecnología CNC y control mediante LabVIEW

Informe Proyecto de Título de Ingeniero Civil Electrónico



Escuela de Ingeniería Eléctrica



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Diseño, desarrollo y construcción de una estructura robótica con tecnología CNC y control mediante LabVIEW

Felipe Andrés Delgado Rodríguez

Informe de proyecto para el grado de Ingeniero Civil Electrónico,
aprobada por la comisión de la
Escuela de Ingeniería Eléctrica de la
Pontificia Universidad Católica de Valparaíso
conformada por

Sr. Héctor Renato Vargas Oyarzún

Profesor Guía

Sr. Cristian Andrés Castro Lagos

Segundo Revisor

Sr. Sebastián Carlos Fingerhuth Massmann

Secretario Académico

Valparaíso, 19 de Abril de 2018

A mi familia, los que me han guiado en todos los caminos.

Agradecimientos

Este trabajo fue realizado gracias a muchas personas que me apoyaron, creyeron en mí y me motivaron a seguir este largo camino, a continuación, nombraré a las personas que fueron fundamentales en este proceso.

Primero que todo, a mis padres, quienes se esforzaron durante toda su vida para ver los logros de sus hijos, desde aquí les puedo decir: ¡lo lograron! En lo que a mí respecta, sus enseñanzas y su forma de vivir lograron que me formara en lo que soy ahora, así como mis otros hermanos. Somos geniales por culpa suya.

Quiero agradecer a mis hermanos, Roberto y Edwin, quienes siempre me apoyaron en todos mis proyectos, sus consejos me han hecho llegar hasta aquí y se los agradezco infinitamente.

Obviamente agradecer a Vanessa Lara, mi amada, la más rica, gracias a ella he logrado muchas cosas y me ha ayudado a enderezarme y a volver a creer en mí incontables veces, se lo agradezco mucho y espero seguir siendo el mejor para ella en todo lo que se nos viene y seguir haciéndola feliz para siempre.

También agradecer a Javier Castro, mi socito, mi amigo, yo sé que me faltarían hojas para agradecerle todo lo que ha hecho por mí y en todo lo que me ha ayudado, sin duda este trabajo de tesis no sería ni la mitad de bueno si no fuera por él.

Y finalmente, no quiero dejar fuera a todos mis amigos y compañeros, especialmente a Erik, Oliver, Cinthya, Angela, Sebastián, Camila, Giovanni, Francisca, Esteban, Luis, y ojalá no se me haya olvidado uno que se sienta mal... todos me han acompañado durante este tiempo, todos aportaron en algo para que yo llegara a este lugar, y por eso ¡muchas gracias!

Valparaíso, 19 de abril de 2018

F. D

Resumen

En el presente proyecto se plantea la construcción de una estructura robótica que tiene movilidad en los ejes X, Y y Z mediante el control de motores paso a paso utilizando tecnología CNC, específicamente por medio de códigos G. Adicionalmente se incluye una cámara web que establece el área de trabajo y permite identificar objetos, formas o colores, esto con el fin de entregarle al actuador información relevante sobre lo que se encuentra en la superficie de trabajo, para que de esta forma la máquina desempeñe la tarea objetivo.

Previo a los resultados obtenidos y los pasos a realizar para la construcción de la estructura, se realiza un breve repaso de los tópicos más relevantes referidos a la máquina, su futuro uso en docencia, así como de las propuestas de aplicaciones y objetivos a realizar.

El actuador es una placa Arduino Mega 2560 y una shield RAMPS v1.4. El firmware utilizado es Marlin 1.0.7, el cual, es el intermediario entre los códigos G recibidos vía software y el movimiento de la máquina.

La máquina se controlará mediante una interfaz desarrollada en LabVIEW. Esta interfaz debe tener varias características, como el monitoreo vía webcam del área de trabajo, comunicación serial para el envío de los códigos G, así como de distintos módulos para las aplicaciones que pueda desarrollar la máquina, a la vez que siendo intuitiva y amigable para el usuario.

Finalmente, se hace entrega de un Manual de Usuario, documento que busca interiorizar al alumno en el uso de la máquina de forma correcta y segura. También se entrega una serie de Guías de Ejercicios, con miras de que la máquina sea utilizada en futuras sesiones del Laboratorio de Control Automático, cuyo objetivo final es la introducción del uso de códigos G y herramientas CNC dentro del programa del curso.

Palabras claves: Control Numérico Computarizado, Visión Artificial, Reconocimiento de Objetos, Programación, LabVIEW, Códigos G, Motores Paso a Paso.

Abstract

In this project is presented the building of a robotic structure which has mobility in the X, Y and Z axis through the control of step motors using CNC technology, specifically by the use of G-Codes. Additionally, a camera it's included that establishes the work area and allows to identify objects, shapes or colors, this in order to deliver the controller relevant information about what's in the workspace so that the machine can perform the objective of the task.

Prior to the obtained results and the steps to perform for the building of the structure, a brief review is made about the main topics of the project, its future use in teaching, as well as the proposals and objectives to be performed.

The controller is an Arduino MEGA board with a RAMPS shield. The used firmware is Marlin, which it's the translator between the G-Codes received via software and the movement of the machine.

The machine will be controlled through a LabVIEW interface. This interface must have several characteristics, as the monitoring of the workspace, serial communication for the sending of G-Codes, as well as different application modules and, at the same time, being intuitive and user-friendly.

Finally, a User Manual is delivered, document that seeks to guide the user in the use of the machine in a safe and correct way. Also, an Exercise Guide is delivered, with the goal that the machine will be used in future sessions of the Automatics Control Laboratory, which its main objective is the introduction of G-Codes and CNC Tools to the course program.

Key words: Computer Numerical Control, Computer Vision, Object Recognition, Programming, LabVIEW, G-Codes, Step Motors.

Índice general

Introducción	1
1 Descripción del Proyecto	4
1.1 Contextualización	4
1.1.1 La Educación en el Control Automático	4
1.1.2 Control Numérico Computarizado.....	5
1.1.3 Visión Computacional.....	6
1.2 Problemática	7
1.3 Solución propuesta	8
1.4 Propuestas de Aplicaciones.....	8
1.4.1 Búsqueda y Recolección	9
1.4.2 Automatización de Huerto	9
1.4.3 Escritura o Grabados	10
1.5 Objetivos	11
2 Diseño.....	12
2.1 Estructura de la Máquina	12
2.2 Materiales.....	13
2.3 Especificaciones.....	14
2.4 Piezas 3D.....	15
2.4.1 Soporte de la Estructura.....	15
2.4.2 Guías de Cables	16
2.4.3 Soporte para los sensores “Final de Carrera” del eje Y.....	17
2.4.4 Soporte Webcam.....	17
2.4.5 Caja para la Electrónica	18
2.4.6 Herramientas Intercambiables	20
2.5 Electrónica y Cableado	22
3 Movimiento de la Estructura.....	24
3.1 Área de Trabajo	24
3.2 Componentes Electrónicos.....	24
3.2.1 Arduino	25
3.2.2 Shield RAMPS	25
3.2.3 Motores Paso a Paso.....	26

3.2.4 Sensores “Final de Carrera”	30
3.3 Firmware Marlin	31
3.3.1 Pasos máximos por eje.....	31
3.3.2 Pasos por unidad	32
3.3.3 Velocidad Máxima de Movimiento.....	32
3.3.4 Aceleración Máxima	32
4 Códigos G.....	33
4.1 Introducción a los Códigos G	33
4.2 Lógica de Movimiento.....	34
4.2.1 Punto Home	34
4.2.2 Posicionamiento	34
4.2.3 Movimiento	35
5 Interfaz de LabVIEW.....	36
5.1 Información General.....	36
5.2 Procesos	37
5.2.1 Instrucciones	37
5.2.2 Calibración.....	38
5.2.3 Movimientos Básicos	38
5.2.4 Búsqueda y Obtención.....	39
5.2.5 Huerto Autónomo.....	42
5.3 Reconocimiento con Color Pattern Matching.....	43
Discusión y Conclusiones	45
Bibliografía.....	48
A Manual de usuario	50
A.1 Indicaciones de Seguridad.....	50
A.2 Interfaz.....	50
A.2.1 Información General	50
A.2.2 Calibración	52
A.2.3 Movimientos Básicos	53
A.2.4 Búsqueda y Obtención	54
A.2.5 Huerto Autónomo	55
A.3 Movimiento	56
A.4 Códigos G.....	57
A.4.1 Códigos G más utilizados	58
A.4.2 Lógica de Movimiento	58
B Guías de Ejercicios	60
B.1 Primeros Pasos	60
B.2 Figuras planificadas.....	61
B.3 Reconocimiento de Objetos	62
B.4 Búsqueda y Obtención.....	63

B.5 Modificaciones a Búsqueda y Obtención.....63

Índice de Figuras

Figura 1-1: Diagrama de funcionamiento de una máquina CNC	5
Figura 1-2: Diagrama de funcionamiento básico de la visión computacional.....	6
Figura 1-3: Ejemplo de detección de figuras geométricas en LabVIEW	7
Figura 1-4: Planta Hexapod (Quanser) que ofrece 6 grados de libertad.....	7
Figura 1-5: Robots RCX340+iVY2 de <i>Yamaha</i> ordenando dados en una bandeja	9
Figura 1-6: Plataforma de Farmbot.io	10
Figura 1-7: Detalle de grabación laser con una máquina CNC	11
Figura 2-1: Representación 3D de la base de la plataforma.....	12
Figura 2-2: Altura de la Cámara Web.....	13
Figura 2-3: Diseño de los soportes de la estructura.....	15
Figura 2-4: Soporte para la Webcam (A) y para el “Final de Carrera” (B)	16
Figura 2-5: Diseño (A) y resultado (B) del Acople de Motor con entrada para cables	16
Figura 2-6: Diseño (A) y resultado (B) de Anclaje para Cables de ejes Y y Z.....	17
Figura 2-7: Diseño (A) y resultado (B) del soporte para los sensores “Final de Carrera” del eje Y	17
Figura 2-8: Diseño (A) y resultado (B) de Soporte de Cámara	18
Figura 2-9: Diseño (A) y resultado (B) de unión perfiles de aluminio de cámara	18
Figura 2-10: Diseño (A) y resultado (B) de la Caja para la electrónica.....	19
Figura 2-11: Diseño (A) y resultado (B) del soporte del eje Z	20
Figura 2-12: Diseño (A) y resultado (B) del Módulo Electroimán	21
Figura 2-13: Diseño (A) y resultado (B) del Módulo LED	21
Figura 2-14: Diseño (A) y resultado (B) del Módulo Escritura.....	22
Figura 2-15: Conexión tarjeta RAMPS.....	23
Figura 3-1: Área efectiva de trabajo en los ejes	24
Figura 3-2: Placa de desarrollo Arduino Mega	25
Figura 3-3: Tarjeta RAMPS v1.4.	26
Figura 3-4: Motor Paso a Paso NEMA 17	26
Figura 3-5: Rotor y estator de un motor de reluctancia variable.....	27
Figura 3-6: Rotor y estator de un motor híbrido	27
Figura 3-7: Devanados de un motor unipolar y de un motor bipolar	28
Figura 3-8: Controlador DRV8825.....	29
Figura 3-9: Controlador A4988.....	29
Figura 3-10: Sensor “Final de Carrera”	30
Figura 3-11: Pines para finales de carrera en la tarjeta RAMPS.....	31
Figura 5-1: Panel frontal de la interfaz	36

Figura 5-2: Diagrama de Flujo general de la aplicación	37
Figura 5-3: Diagrama de Flujo de la sección Instrucciones	38
Figura 5-4: Diagrama de Flujo de la sección Calibración	38
Figura 5-5: Ejemplo de búsqueda de una imagen (template) dentro del área de trabajo	39
Figura 5-6: Diagrama de Flujo de sección Movimientos Básicos	40
Figura 5-7: Creación de nuevo template desde la imagen capturada por la Webcam.....	41
Figura 5-8: Diagrama de Flujo de sección Búsqueda y Obtención.....	41
Figura 5-9: Diagrama de Flujo de sección Huerto Autónomo.....	42
Figura 5-10: Muestra de un reconocimiento de una planta en un almacigo.....	43
Figura 5-11: Detalle del Color Location con pastillas de colores	44
Figura 5-12: Detalle del Grayscale Pattern Matching con fusibles	44

Índice de Tablas

Tabla 2-1: Lista de materiales para el proyecto.....	13
Tabla 2-2: Especificaciones estructurales	14
Tabla 2-3: Especificaciones de Software.....	14
Tabla 2-4: Especificaciones Eléctricas	14
Tabla 2-5: Especificaciones de Motores	14
Tabla 2-6: Especificaciones de Cámara Web [11]	14
Tabla 2-7: Especificaciones de Piezas 3D	14
Tabla 2-8: Características cable AWG 26	22
Tabla 2-9: Conexión tarjeta RAMPS	23
Tabla 3-1: Características principales de la placa Arduino Mega	25
Tabla 3-2: Características principales motor paso a paso 17HS4401	28
Tabla 3-3: Comparación entre controladores A4988 y DRV8825	29
Tabla 4-1: Tipo de códigos G	33
Tabla 4-2: Respuestas típicas en Marlin.....	34

Introducción

Las plantas de laboratorio son máquinas que representan un fenómeno físico o sistema a escala, lo que permite trabajar con un sistema complejo abaratando costos y espacio de trabajo, a la vez que los estudiantes puedan acceder de forma fácil a estas tecnologías [1]. Generalmente tienen una interfaz que permiten al usuario controlar las variables de la planta y también obtener datos de su funcionamiento en tiempo real para estudios posteriores.

El trabajo en laboratorio cumple un rol muy importante, que es el de permitir a los estudiantes llevar a cabo experiencias que van más allá de lo teórico. Los estudiantes de hoy en día tienen una preferencia por el aprendizaje a través de la práctica e instrucciones lúdicas, tema que queda demostrado en muchas investigaciones sobre la materia [2].

Diversos estudios a lo largo de la historia demuestran con distintas cifras que, en el campo de la ciencia, el trabajo práctico en laboratorios resulta muy positivo para el aprendizaje de los estudiantes. Los resultados dan cuenta que los estudiantes, en general, aprenden de mejor manera un tópico cuando estos tienen que buscar por sí mismos las respuestas o ver el cambio que ocurre cuando ellos manipulan una variable de forma distinta, en resumen, ser partícipes del fenómeno que se está estudiando [2]. Esto es precisamente lo que ofrece una planta de laboratorio.

Es importante entregarles a los estudiantes no solo herramientas que cumplan alguna función en particular, sino que además, estas sean interesantes y atractivas, para que así, la actividad no se vuelva aburrida y/o rutinaria. Para esto es importante que la naturaleza de las plantas que sean incorporadas en el laboratorio cuenten con funciones que hagan de las experiencias una herramienta útil en el aprendizaje y el desarrollo personal, así mismo como mantener el interés de parte del alumnado.

El Control Numérico Computarizado (CNC) es un sistema de control y automatización de la posición de un elemento físico, generalmente una herramienta montada en una máquina. Este sistema se basa en operar a la máquina mediante comandos programados, utilizando un sistema de coordenadas previamente establecido, el cual especifica el movimiento posible de la herramienta. Cabe destacar que el CNC no solo controla la posición de la herramienta, sino también la forma cómo ésta se desplaza, esto es, la velocidad, aceleración, fuerza de los motores, etc.

El origen del CNC se remonta a principios de los años 50 en el Massachusetts Institute of Technology (MIT) [3], en donde se automatizó por primera vez una máquina de tipo fresadora. Con el paso del tiempo esta tecnología se ha masificado y ha revolucionado a la industria moderna, principalmente por el bajo costo de los microprocesadores y las facilidades de programación actuales.

El éxito obtenido por las máquinas CNC radica en la precisión de sus movimientos, por lo cual, en la etapa de implementación, es de vital importancia la correcta calibración de sus elementos, como también el registro claro, minucioso y documentado de los procesos empleados para futuras mantenciones preventivas o reparaciones requeridas.

En cuanto a las máquinas de tipo CNC, es indispensable contar con métodos que permitan el control de estas mismas de una forma efectiva. Para esto se emplean los códigos G, ampliamente utilizados debido a sus características y alcances en el control de máquinas CNC. Los códigos G son instrucciones predeterminadas que buscan controlar el movimiento de la máquina junto con parámetros como velocidad, aceleración, tiempos de espera, entre otros. En orden de poder controlar de forma correcta la máquina, es fundamental comprender en profundidad los códigos G, esto para que la máquina no intente hacer movimientos que podrían dañar su estructura, así como también, para lograr el comportamiento deseado en las experiencias de laboratorio.

Para poder controlar el movimiento de la máquina mediante códigos G, es necesario que estos sean traducidos por el actuador, que en este caso es un Arduino Mega con una tarjeta RAMPS v1.4 adosada. Para esto se carga en el Arduino el firmware Marlin, un firmware utilizado para controlar máquinas CNC, especialmente impresoras 3D. Este ha sido creado, y es constantemente revisado, por la comunidad maker en internet, haciéndolo uno de los firmwares de código libre más utilizado y confiable para programar una máquina CNC.

Por otro lado, la Visión Computacional es un campo interdisciplinario que aborda la forma en que los computadores pueden comprender y procesar lo que observan a partir de fotos o vídeos obtenidos por una cámara. Uno de los campos de la visión computacional es el reconocimiento de imágenes, disciplina muy importante en el campo de la ingeniería, pues permite automatizar tareas en las que se precisa de visión de ciertos procesos o eventos. Esto se logra luego de procesar, analizar y entender imágenes digitales extraídas del mundo real, o sea, convertir estas imágenes en datos, símbolos y números, para su posterior trabajo dentro del campo de la programación.

Cuando la visión computacional, y más específicamente el reconocimiento de objetos, entrega información relevante para que el proceso pueda desarrollar una función a partir de estos datos, se abre una gama de posibilidades enorme. Debido a esto, es necesario enfocar el trabajo no solamente en que desarrolle las funciones objetivo, sino que también pueda ser modificada para que en el futuro pueda desarrollar otras funciones.

En este sentido, el toolkit VISION de LabVIEW, enfocado precisamente en el reconocimiento de imágenes en general, entrega funcionalidades relativamente fáciles de implementar y hace que el proceso de reconocimiento sea muy personalizable. La mejor alternativa para trabajar con reconocimiento con este toolkit es el Color Pattern Matching, un proceso que combina dos procesos de reconocimiento para mejorar la calidad de este, estos son Color Location y Grayscale Pattern Matching. Básicamente lo que hace al combinar estos dos procesos es primero, ubicar los colores de un template o plantilla dentro de la imagen donde se efectuará la búsqueda, luego, en las regiones cercanas al color ubicado, se realiza un proceso de convertir la imagen a escala de grises, con el fin de obtener información sobre los contornos de la imagen. Al realizar esto, se les asigna un puntaje a las posibles coincidencias, este puntaje es configurado y es, en palabras simples, "qué tanto se parece" la imagen obtenida con la plantilla a buscar.

Es importante que el puntaje de exigencia del reconocimiento sea personalizable por el usuario, en orden de que exista un umbral más o menos exigente dependiendo del objeto a reconocer. En este mismo sentido, se requiere que la imagen obtenida por la cámara sea de una buena calidad, esto para asegurar que el reconocimiento sea realizado de una buena forma para evitar cualquier problema de falso o bajo reconocimiento.

El origen de este proyecto, se basa en los esfuerzos de entregar una planta que fuera más allá de lo común y que, por ende, fuera más ambiciosa en su proceso de funcionamiento. Se incluye la necesidad de contar con varios grados de libertad, en este caso, los ejes de coordenadas controlados por CNC y también una propuesta novedosa, que sería el reconocimiento de imágenes para la obtención de las coordenadas de distintos objetos que cumplen un rol dentro de la tarea a realizar.

Combinar estas áreas de trabajo, la tecnología CNC y el reconocimiento de imágenes, en una sola máquina, entrega muchas posibilidades de aplicaciones distintas, por lo que la realización de esta máquina presenta una valiosa adquisición de una planta para el Laboratorio de Control Automático.

1 Descripción del Proyecto

Previo a definir la creación y características del proyecto, es importante investigar y documentar las tecnologías que se utilizarán y también cómo han sido abordados, por distintas entidades, proyectos similares al propuesto.

1.1 Contextualización

A continuación, se presentarán de forma breve los 3 temas sobre los que se trata este proyecto, la educación en el control automático en la actualidad, el control numérico computacional, un poco de su historia y sobre su funcionamiento y finalmente la visión computacional, explicando su origen y funcionamiento.

1.1.1 La Educación en el Control Automático

Actualmente, se suele impartir una asignatura básica de Control Automático, en la cual se estudian sistemas de control lineales analizados en el tiempo y en el dominio complejo s . Esta asignatura demanda un conocimiento medio de cálculo, ecuaciones diferenciales y transformadas de Laplace.

Dada la actual accesibilidad a la tecnología y al conocimiento en general, la educación hoy en día demanda una mayor participación de parte del estudiante [4]. Es por esto que además de la enseñanza tradicional en temas de control clásico, es muy importante que se impartan también experiencias de laboratorio enfocadas en apoyar el aprendizaje, haciendo una conexión entre lo teórico y lo real [5].

Aunque, a través de la historia, muchas veces se ha desacreditado y en ocasiones descalificado, el trabajo en los laboratorios tiene un impacto enorme en la educación en las ciencias en general. Los trabajos de laboratorio son un apoyo educativo muy importante, en especial en el campo de las ciencias. Esto siempre y cuando las experiencias prácticas en el laboratorio no sean “una receta de cocina”, sino que lleven a los alumnos a reflexionar sobre el trabajo relacionado, involucrarse en tareas grupales y ser un ente relevante en el desarrollo de las diversas tareas encomendadas [6].

Las experiencias de laboratorio se realizan basadas en guías de práctica, las cuales buscan principalmente hacer partícipe al alumno en su aprendizaje. Estas guías de práctica, están enfocadas a trabajos con distintas plantas disponibles en el laboratorio, las cuales realizan distintas operaciones, y por medio de interfaces de monitoreo y control, los alumnos pueden editar los ajustes necesarios. Esto permite a los estudiantes obtener el modelo matemático del sistema (función de transferencia) y a partir de este, desarrollar controladores para requerimientos específicos.

Algunos de los beneficios obtenidos de lo mencionado anteriormente son [7]:

- **Aprender y comprender**, a la vez que **hacer y aprender a hacer**.
- Facilitar a los estudiantes el acceso a la tecnología para aprendizaje.
- Hacer la conexión entre teoría impartida en la universidad y la vida real.
- Aportar en la preparación del estudiante al mundo laboral.
- Aumentar no solo las habilidades de resolución de problemas, sino que también las habilidades sociales y comunicativas.
- Potenciar las fortalezas individuales a la vez que estimular el trabajo cooperativo.
- Explorar áreas de interés dentro del contexto del control automático.

1.1.2 Control Numérico Computarizado

El Control Numérico Computarizado (CNC) es el control y monitoreo de los movimientos de una máquina/herramienta mediante el uso de una computadora. Estas máquinas/herramientas tienen diversos usos, como, por ejemplo: máquinas fresadoras, tornos, corte laser, impresoras 3D, estampadoras, brazos robóticos, entre otros [8].

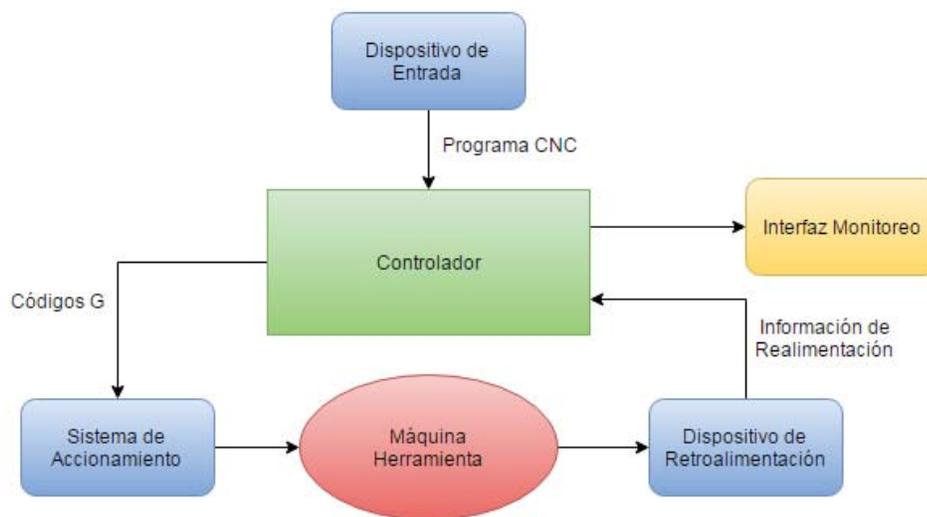


Figura 1-1: Diagrama de funcionamiento de una máquina CNC

El funcionamiento de una máquina CNC es básicamente un controlador que recibe órdenes de una computadora, estas órdenes vienen en forma de códigos previamente establecidos, los cuales son interpretados en acciones que un actuador debe realizar, como, por ejemplo, mover los motores de los ejes de la máquina, cambiar su velocidad, detener/iniciar el proceso, etc. La máquina realimenta al controlador con datos como su posición o velocidad para su posterior ajuste. Todo este proceso es mostrado por medio de una interfaz de monitoreo. De forma básica se puede ver el funcionamiento de una máquina CNC en la Figura 1-1.

Todas las máquinas CNC tienen 2 o más direcciones de movimiento programables, estos son los ejes de movimiento. En general, entre más ejes, más compleja es la máquina. Los ejes lineales más comunes son un eje X e Y, que permiten posicionarse sobre un plano y un eje Z que permite acercarse o alejarse verticalmente de dicho plano.

En orden de poder controlar a la máquina, se utilizan los mencionados códigos de programación. Estos códigos son básicamente un listado secuencial de instrucciones que el actuador traduce en distintos movimientos y operaciones. Dentro de los códigos más utilizados se encuentra el código N, el cual trabaja mediante bloques de operación, el código XYZ, que solo trabaja en coordenadas cartesianas y el código G, basado en funciones preparatorias. Este último tiene la particularidad de que no solamente le comunica a la máquina *qué* hacer, sino que también *cómo* hacerlo.

Finalmente el actuador o sistema de accionamiento, de la máquina CNC es el que se encarga de recibir las órdenes y llevarlas a cabo mediante el control de movimiento de los ejes.

1.1.3 Visión Computacional

David Marr (importante neurofisiólogo que contribuyó de manera significativa a procesamiento visual computacional) define la visión como: “un **proceso** que produce, a partir de las imágenes del mundo exterior, una **descripción** que es útil para el **observador** y que no tiene **información irrelevante**” [9]. En base a esta definición, se pueden desprender los siguientes aspectos:

- La visión puede ser un proceso computacional.
- La descripción obtenida depende de las características del observador.
- Es necesario que exista un proceso de filtraje para eliminar la información irrelevante.

En la Figura 1-2 se puede ver un diagrama básico de cómo funciona la visión computacional.



Figura 1-2: Diagrama de funcionamiento básico de la visión computacional

Uno de los campos más importantes de la visión computacional es el reconocimiento de objetos, cuyo objetivo es, tal como su nombre lo indica, la capacidad de un sistema computacional para encontrar e identificar objetos en una imagen o una secuencia de vídeo. Esto permite realizar las siguientes acciones [10]:

- Determinar, entre otros, la ubicación, forma y colores de un objeto en una imagen.
- Construir una representación tridimensional de un objeto.
- Analizar un objeto para determinar su calidad.

La obtención de la descripción de una imagen ocurre cuando un sensor (cámara) registra la radiación (luz) que ha interactuado con objetos. Esta imagen pasa luego a un proceso de interpretación, en donde la imagen entrega finalmente la información requerida, esto por medio de métodos, como aquellos basados en apariencia (emparejamiento de bordes, escala de grises, etc.) o características (árboles de interpretación, conjetura y pruebas, etc.).

Hoy en día, el reconocimiento de objetos cumple un rol importante en las áreas de control y automatización, y para tales fines, un sistema de reconocimiento de imágenes de buena calidad es indispensable.

El reconocimiento de imágenes puede ser muy difícil de implementar, por lo que programas como LabVIEW presentan herramientas prefabricadas que facilitan mucho la implementación de procesos de reconocimiento de imágenes [11]. Uno de los ejemplos incluidos en el toolkit VISION de LabVIEW puede verse en la Figura 1-3.

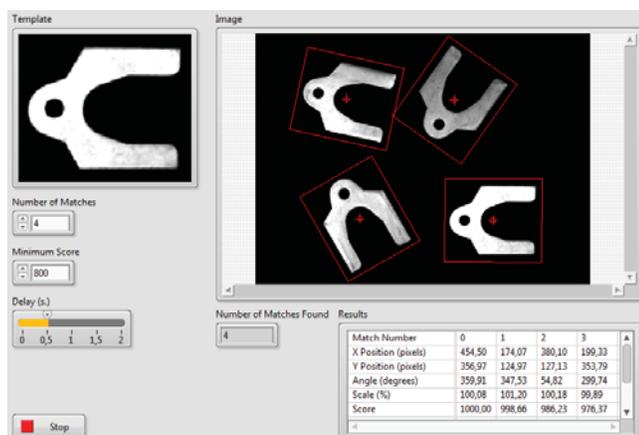


Figura 1-3: Ejemplo de detección de figuras geométricas en LabVIEW

1.2 Problemática

El uso de una máquina CNC como planta de laboratorio es una alternativa muy atractiva por los grados de libertad que ofrece, y lo es aún más si se tiene en cuenta la implementación de reconocimiento de objetos para controlar la posición. Pero, además de desarrollar una planta que se diferencie de las demás en cuanto a sus características, un punto importante es enfocar el uso de esta planta para temas educativos, es decir, no solo diseñar y programar la planta para que funcione de manera correcta, sino que también proveer al laboratorio con una interfaz de control y monitoreo que permita que los alumnos puedan realizar todas las actividades para controlar la planta, con los requerimientos propuestos en experiencias de laboratorio guiadas por un profesor o un ayudante.



Figura 1-4: Planta Hexapod (Quanser) que ofrece 6 grados de libertad

Actualmente existen muchas empresas que desarrollan plantas con propósitos educativos, es decir, que además de la planta incluyen una interfaz de control, manuales de uso y guías de desarrollo. Algunas de estas empresas son: National Instruments, Quanser, Feedback, Bytronic, TecQuipment y ECP Systems.

Aunque existen algunas plantas que tienen mucha complejidad, como el Hexapod de Quanser (Figura 1-4) que ofrece 6 grados de libertad, brazo robótico de Bytronic (5 grados de libertad) o la planta de torsión (más de 9 configuraciones distintas) de ECPSystems. La mayoría de estas soluciones incluyen tipos de plantas típicos, tales como control de motor DC, control de péndulo, control de temperatura, control de levitación magnética, etc. Aplicaciones más complejas son vendidas como herramientas para procesos automáticos a gran escala y no con intenciones de docencia.

Aun con los elevados precios de compra de estas herramientas, las entidades como las universidades, buscan adquirir estas plantas, debido al valor educativo que entregan a sus alumnos.

1.3 Solución propuesta

Para lograr que el proyecto se implemente como una solución a las problemáticas mencionadas, es necesario, además del diseño, implementación y programación de la planta, incluir una interfaz intuitiva, fácil de utilizar, que permita no solo monitorear a la planta, sino que también controlarla. Además, debería incluir módulos de uso, por ejemplo, controlar los diversos ejes de coordenadas, reconocimiento de imágenes, etc.

Al incluir una interfaz que permita hacer lo antes mencionado, la planta no solo es una máquina que realiza cierta tarea, sino que, se convierte en una herramienta que permite que los estudiantes puedan aprender haciendo.

Para que la planta logre posicionarse como una solución, debe tener las siguientes características:

- Funcionamiento óptimo de la planta, fácil de calibrar y también de reparar, en caso de alguna avería.
- Interfaz *user-friendly*, que permita controlar la planta para iniciar, detener, editar parámetros, etc.
- Incluir guía de prácticas con ejercicios propuestos.
- Incluir un “Manual de Usuario” con instrucciones de uso y precauciones de la máquina.

Adicional a esto, se busca que la planta pueda funcionar, por medio de la interfaz en las siguientes aplicaciones:

- Movimiento de los ejes de forma individual o en conjunto.
- Desarrollar diferentes modos de uso.
- Aprendizaje del sistema de reconocimiento de imágenes (obtención de nuevas plantillas).

1.4 Propuestas de Aplicaciones

Una planta con estas características tiene una amplia gama de posibles aplicaciones, por lo que a continuación se detallan algunas.

1.4.1 Búsqueda y Recolección

Al ubicar piezas de distinta forma y color en la plataforma, se busca poder reconocer una o varias piezas en particular y lograr que la máquina la coja por medio del electroimán y la devuelva a un punto predeterminado, o que incluso la mueva de posición.

Haciendo uso de la cámara y mediante reconocimiento de objetos, la máquina podría buscar la pieza requerida, enviar al actuador las coordenadas en el plano donde se encuentra para que este, utilizando códigos G, se mueva utilizando la mejor ruta para tomarla con el electroimán y dejarla en donde se requiera. Esto debería poder hacerse para varias piezas de características semejantes.

La mejor ruta no necesariamente puede ser una línea recta, puesto que también, una de las características de esta aplicación podría ser esquivar objetos o seguir trayectorias predeterminadas (por ej. Seguimiento de líneas).

Esto permitiría no solamente recolectar las piezas necesarias en un depósito ubicado en un lugar predeterminado, sino que, al aumentar el aprendizaje del reconocimiento de piezas, podría servir para clasificar, agrupar y ordenar distintos objetos.

Algunos ejemplos de esta aplicación son:

- Robots tipo SCARA de **Yamaha RCX340+iVY2** [12], un brazo robótico que junto con una cámara clasifican distintos objetos que pasan por una correa transportadora (Figura 1-5).

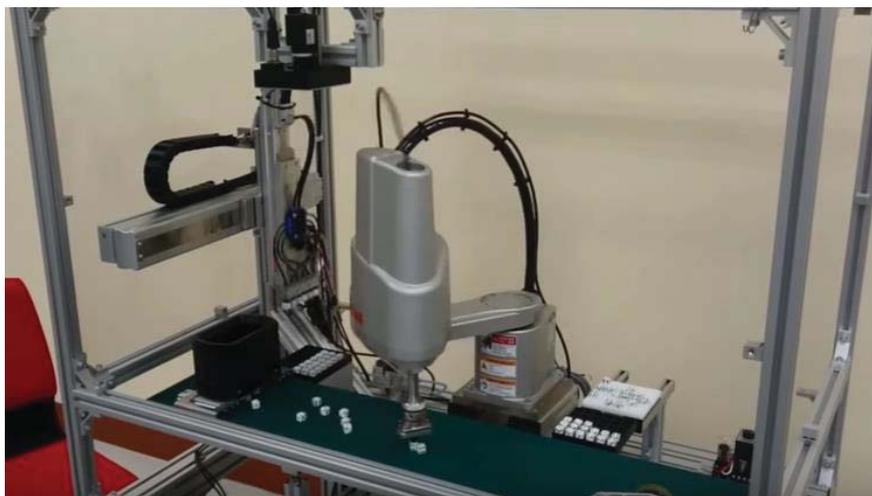


Figura 1-5: Robots RCX340+iVY2 de *Yamaha* ordenando dados en una bandeja

- Muchos modelos de robots de **FANUC**, destacándose el robot **M-1iA/0.5A** [13] que con rápidos movimientos y alta precisión es capaz de ordenar objetos de distintas características mediante una fotografía tomada en una superficie.

1.4.2 Automatización de Huerto

La máquina tendría el objetivo de cuidar y mantener un huerto pequeño, por ejemplo, almácigos en sus primeras etapas de germinación. Haciendo uso de la cámara web, la máquina obtiene las coordenadas de las plantas para intervenir de distintas formas durante el proceso.

Haciendo uso de la cámara, se pueden identificar no solo la ubicación de las plantas, sino que también cosas como: el color de la tierra (que al oscurecerse representa humedad) o maleza que aparezca (puntos verdes fuera del rango de plantación). De esta forma la máquina podría, realizar un goteo preventivo, o eliminar malezas mediante alguna pieza especial para este fin.

Mediante funciones predeterminadas de códigos G, la máquina puede mantener rutinas de riego para distintas plantas en un mismo almácigo. Al saber dónde se ubica cada tipo de planta, la máquina es capaz de moverse a través del almácigo regando mediante goteo solamente a las plantas que necesitan hidratación.

La máquina, además, podría ser capaz, si se le ajusta un sensor adecuado, de medir la humedad del suelo, manteniendo así estándares necesarios para el buen cuidado de ciertas plantas.

Un ejemplo de esta aplicación es:

- **Farmbot.io** [14], una máquina CNC que está diseñada para mantener un jardín en una cama de tierra de grandes dimensiones, permitiendo plantar semillas, regar plantas y monitorear todo desde una interfaz para computador y móviles (Figura 1-6).

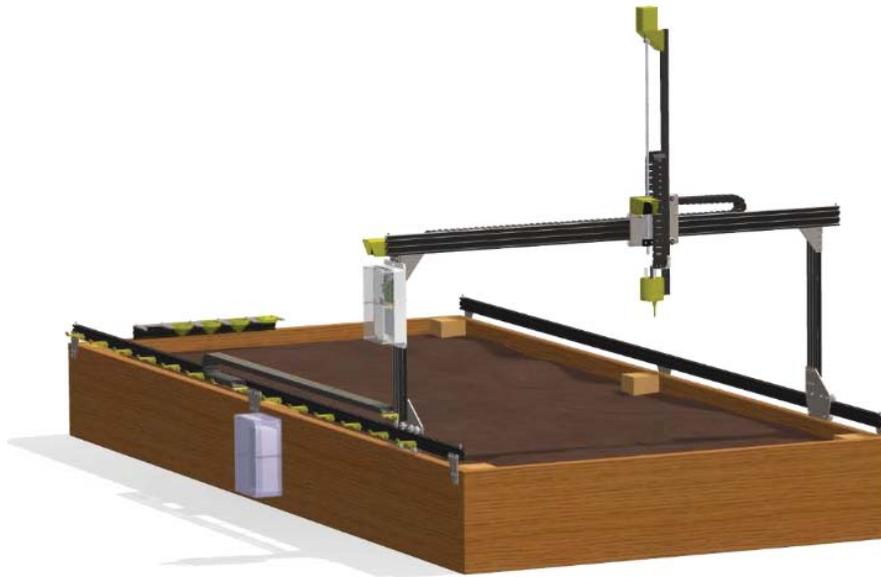


Figura 1-6: Plataforma de Farmbot.io

1.4.3 Escritura o Grabados

Dada sus características, y al integrar un cabezal adecuado, la máquina es capaz de escribir o grabar con láser, no solo palabras, sino que también formas predeterminadas. Dependiendo de los alcances del software, también podría ser capaz de tomar imágenes, transformar estas a códigos G y dibujarlas sobre una superficie (Figura 1-7).

Esta es una de las aplicaciones más comunes en máquinas CNC, se pueden hacer máquinas de escritura, enrutadores, grabadoras laser, grabado de madera y metales, entre otras.

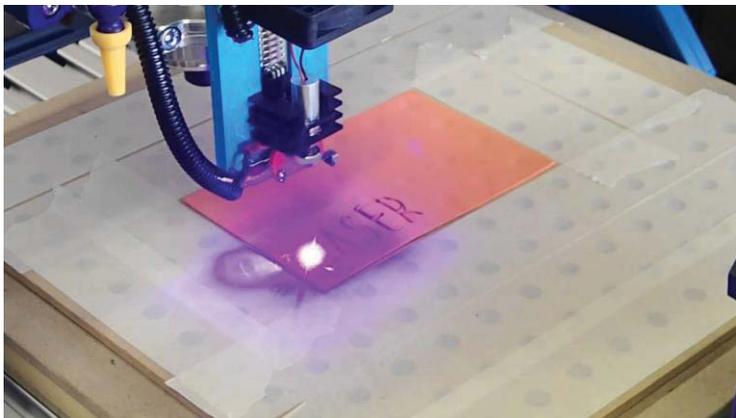


Figura 1-7: Detalle de grabación laser con una máquina CNC

1.5 Objetivos

Teniendo en cuenta lo anteriormente mencionado, el objetivo de alto nivel del proyecto es el diseño, desarrollo y construcción de una estructura robótica con tecnología CNC cuyo control esté dado por una interfaz de LabVIEW. En orden de cumplir con este objetivo, es preciso definir los objetivos que llevarán a concluir con esta propuesta de forma óptima.

- Fundamentar la propuesta en base a su aplicación en docencia universitaria.
 - Encontrar formas de adaptar el uso de la planta en materias educativas.
 - Desarrollar guías de práctica para futuras sesiones del laboratorio.
- Diseñar las piezas en 3D en software de diseño 123D Design.
 - Dejar documentadas las piezas de la estructura en formato de impresión 3D.
- Implementar la estructura de hardware propuesta en el punto anterior.
 - Implementar la estructura de soporte de la cámara.
 - Desarrollar una estructura en el eje Z para el control de las herramientas.
 - Desarrollar una estructura base de la planta.
- Desarrollar aplicación de monitoreo y control de la plataforma en software LabVIEW.
 - Interfaz *user-friendly* con gráficos en tiempo real.
 - Distintos modos de uso.
- Realizar pruebas de funcionamiento del sistema desarrollado.
 - Pruebas de funcionamiento de la mecánica, electrónica y programación.
 - Pruebas de conexión interfaz-planta.
- Presentación y análisis de resultados finales.
 - Entrega de documentos elaborados.
 - Entrega de planta terminada.

2 Diseño

Para la realización de una plataforma dinámica, multifuncional y con alta adaptabilidad para la ejecución de diferentes tareas es fundamental contar con un proceso de diseño meticuloso. Este será transversal a lo largo del proyecto, teniendo que adaptarlo continuamente, durante el proceso de calibración o al comprobar ciertos problemas que se van observando sobre el desarrollo del proyecto. A continuación, se detallará las modificaciones en el diseño de la estructura y mencionando las consideraciones a tomar en cuenta en el desarrollo de los mismos.

2.1 Estructura de la Máquina

A partir de una plataforma de perfiles metálicos (Figura 2-1) enfocada como una máquina CNC obtenida por el laboratorio, se realiza la construcción de la máquina. A esta plataforma se le deben ir haciendo diversas modificaciones para lograr las características deseadas. Estas modificaciones son: añadir la Webcam, conectar los motores a la tarjeta RAMPS, instalar sensores “Final de Carrera”, realizar piezas 3D para modificar algunas piezas, entre otras.

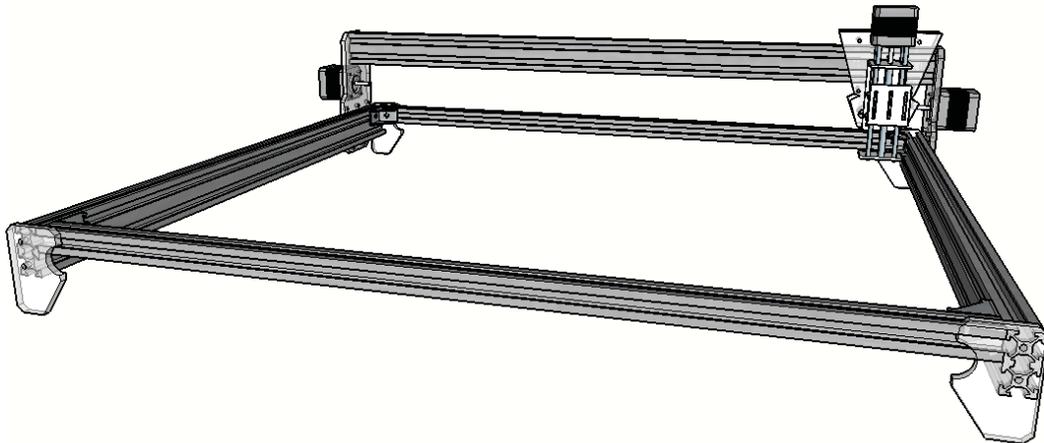


Figura 2-1: Representación 3D de la base de la plataforma

Para determinar cuál era la mejor posición para instalar la Webcam, se establecieron diversos factores a tener en consideración y se decidió por fijarla en la parte superior de la estructura, o sea que apuntara desde arriba hacia abajo. Lo más importante en cuanto a la ubicación de la cámara es que esta tome

toda el área de trabajo de la máquina, para esto es necesario ubicarla a aproximadamente 140[cm] desde la base, como se puede ver en la Figura 2-2.



Figura 2-2: Altura de la Cámara Web

2.2 Materiales

En la Tabla 2-1 se puede ver una lista de los principales materiales y componentes involucrados durante todo el proceso de construcción.

Tabla 2-1: Lista de materiales para el proyecto

Material	Cantidad	Explicación
Estructura metálica	1	Base metálica
Piezas de Acrílico	4	Soportes para los perfiles metálicos
Motores paso a paso	3	Movimiento de los ejes X, Y
Motor paso a paso con estructura	1	Movimiento del eje Z
Controlador Motor DRV8825	3	Controlador de movimiento de los motores
Sensores "Final de Carrera"	5	Calibración de los ejes
Tarjeta RAMPS v1.4	1	Interfaz actuadora
Arduino MEGA	1	Programación
Electroimán	1	Obtención de las piezas
Ventiladores 12V	1	Enfriamiento de la tarjeta RAMPS
Fuente de 12V	1	Alimentación para la máquina
Webcam Logitech C270 HD	1	Cámara para reconocimiento de objetos
Conectores Banana	8	Conectores para herramientas intercambiables

2.3 Especificaciones

En esta sección se comentan los detalles técnicos más relevantes de los componentes de la máquina, estos son detalles tanto de hardware como de software.

Tabla 2-2: Especificaciones estructurales

Material Perfiles	Aluminio
Dimensiones (externas)	72x63[cm]

Tabla 2-3: Especificaciones de Software

Versión de LabVIEW	LabVIEW 2013 32 bits
Versión Módulo NI VISA	16.0
Versión Módulo NI Visión	13.0
Versión Módulo NI-IMAQ	4.8
Versión de Windows	7 Ultimate 64 bits
Marlin	1.0.7

Tabla 2-4: Especificaciones Eléctricas

Voltaje de Entrada	12[V]
Corriente Máxima permitida	5[A]
Comunicación	Serial (USB)
Placa Electrónica	Arduino MEGA 2560
Shield CNC	RAMPS V1.4
Cables utilizados	AWG26

Tabla 2-5: Especificaciones de Motores

Modelo de Motores	NEMA17 17HS4401
Controladores de los Motores	DRV8825

Tabla 2-6: Especificaciones de Cámara Web

Cámara Web	Logitech C270 HD [15]
Captura de vídeo	Hasta 1280x720[píxeles]
Fotos	Hasta 3.0[megapíxeles]
Certificación USB	2.0 de alta velocidad

Tabla 2-7: Especificaciones de Piezas 3D

Material del Filamento	PLA 3D850
Color	Gris
Grosor	1.75[mm]

2.4 Piezas 3D

Gran parte del diseño de la máquina está dada por la estructura de perfiles metálicos adquirida por el laboratorio, pero también es necesario hacer adaptaciones a la estructura y realizar piezas que se alineen a los objetivos requeridos por el proyecto. En este sentido la impresión 3D aparece como una atractiva alternativa debido principalmente a la facilidad para diseñar piezas a medida y crearlas en un corto periodo de tiempo, permitiendo a la vez, realizar ajustes en caso de que la pieza no se haya realizado de forma satisfactoria.

Estas piezas deben ser diseñadas basándose en las medidas reales de los elementos de la máquina, así como del movimiento alcanzado por la máquina, esto último para no entorpecer el movimiento ni reduciendo de forma considerable la zona de trabajo.

Para realizar los diseños de las piezas se utiliza el software de diseño 3D, 123D Design de Autodesk, software seleccionado debido a que es gratuito y entrega poderosas herramientas de diseño muy fáciles de utilizar.

La impresora 3D MakerBot Replicator 2 que se encuentra en el Laboratorio de Robótica de la universidad proporcionará el servicio de impresión 3D para este proyecto, junto al software MakerBot Print, utilizado para traspasar las figuras 3D diseñadas a códigos G que la impresora puede interpretar para imprimir. Además, es importante recalcar que algunas piezas de mayor complejidad y tamaño han sido desarrolladas en la impresora 3D creada por el tesista Javier Castro y con colaboración directa del ejecutor de este proyecto.

2.4.1 Soporte de la Estructura

El rediseño de los soportes nace de la necesidad de entregar mayor utilidad al eje Z, esto aumentando la altura de la estructura. Además, se incluyen 2 soportes que tienen una estructura para los sensores "Final de Carrera" del eje X. Uno de los soportes además tiene la pieza que aguanta el pilar del soporte de la cámara. En la Figura 2-3 se puede ver el diseño de los 3 tipos de soporte (de izq. a der.: Soporte estructura de Webcam, Soporte de sensores "Final de Carrera" y Soporte normal).

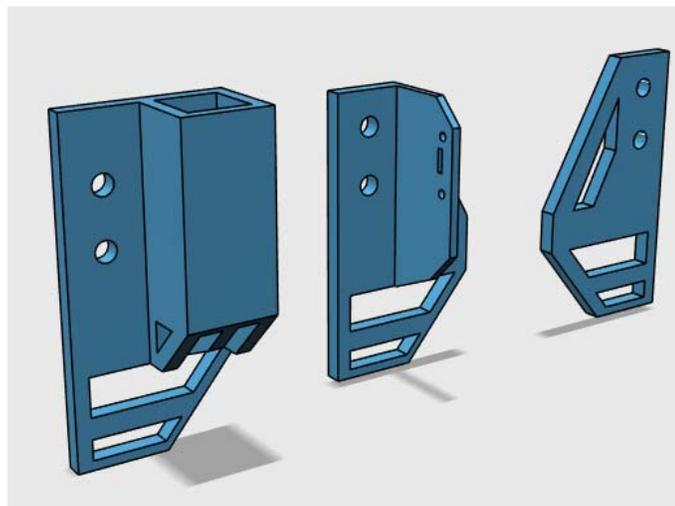
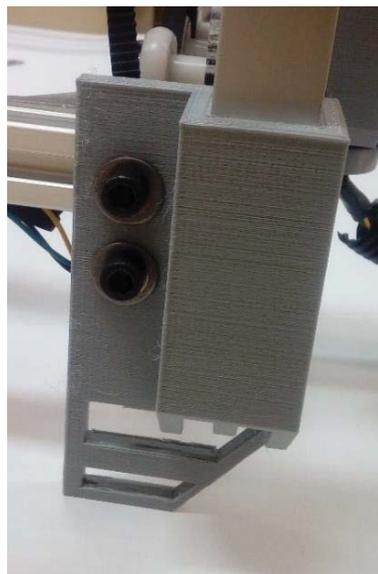


Figura 2-3: Diseño de los soportes de la estructura.

A continuación se pueden ver los resultados de los soportes, tanto del soporte para la Webcam (Figura 2-4 A) y de los sensores “Final de Carrera” (Figura 2-4 B).



(A)

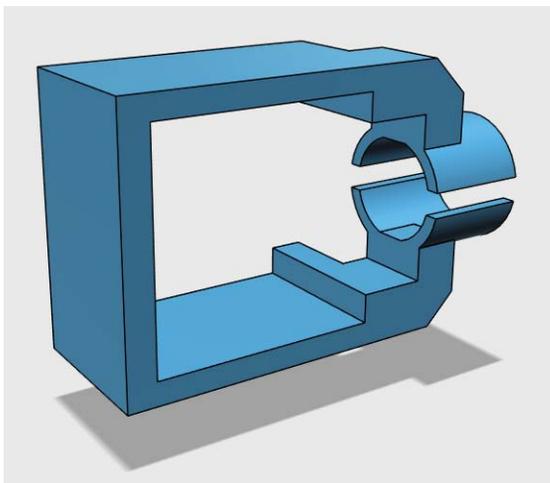


(B)

Figura 2-4: Soporte para la Webcam (A) y para el “Final de Carrera” (B)

2.4.2 Guías de Cables

Primero, se diseñaron e imprimieron en 3D 2 piezas idénticas, estas son utilizadas para sujetar y dar orden a los cables de los motores del eje X. El diseño de esta pieza puede ser visto en la Figura 2-5 A y el resultado en la Figura 2-5 B.



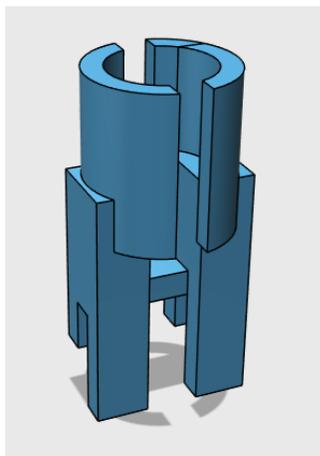
(A)



(B)

Figura 2-5: Diseño (A) y resultado (B) del Acople de Motor con entrada para cables

Luego, se diseñó e imprimió un anclaje para los cables del eje Y y del eje Z, El diseño y el resultado de este se puede ver en la Figura 2-6.



(A)

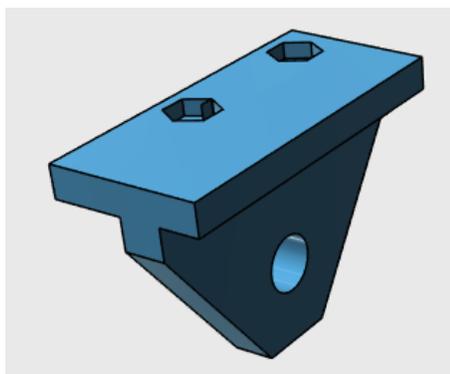


(B)

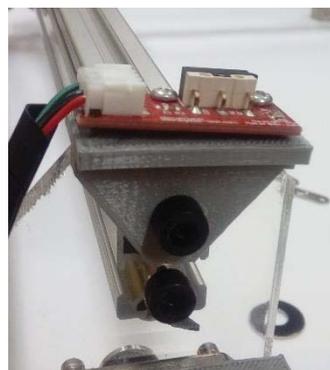
Figura 2-6: Diseño (A) y resultado (B) de Anclaje para Cables de ejes Y y Z

2.4.3 Soporte para los sensores “Final de Carrera” del eje Y

Un soporte adecuado para los sensores finales de carrera resulta indispensable, tanto por un tema estético como para evitar que estos dispositivos se desconecten o se muevan y queden fuera del alcance del movimiento de la máquina. El diseño y resultado de estos puede ser visto en la Figura 2-7.



(A)

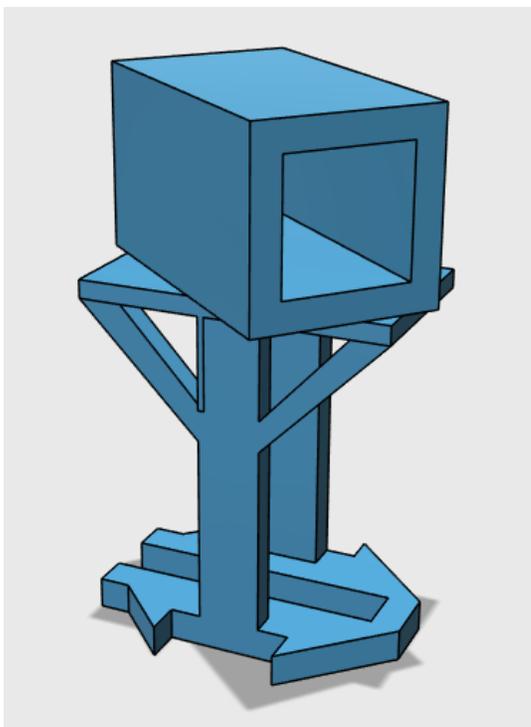


(B)

Figura 2-7: Diseño (A) y resultado (B) del soporte para los sensores “Final de Carrera” del eje Y

2.4.4 Soporte Webcam

Para el concepto del soporte de la cámara se desarrollan 3 piezas, la pieza del soporte antes vista (Figura 2-4 A) que sujeta el perfil de aluminio vertical de la cámara, el soporte en el que la cámara va sujeta (Figura 2-8 A) y una pieza que une ambos perfiles de aluminio del soporte de la cámara (Figura 2-9 A). Se pueden ver los resultados de estas piezas (Figura 2-8 B y Figura 2-9 B), demostrando que cumplen su propósito a la perfección.

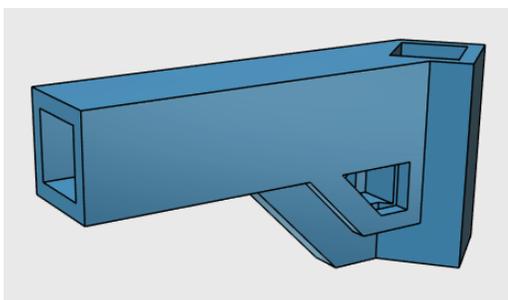


(A)

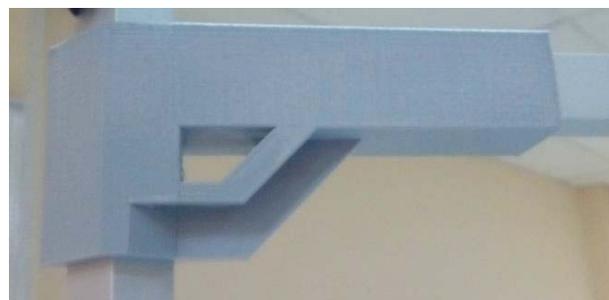


(B)

Figura 2-8: Diseño (A) y resultado (B) de Soporte de Cámara



(A)



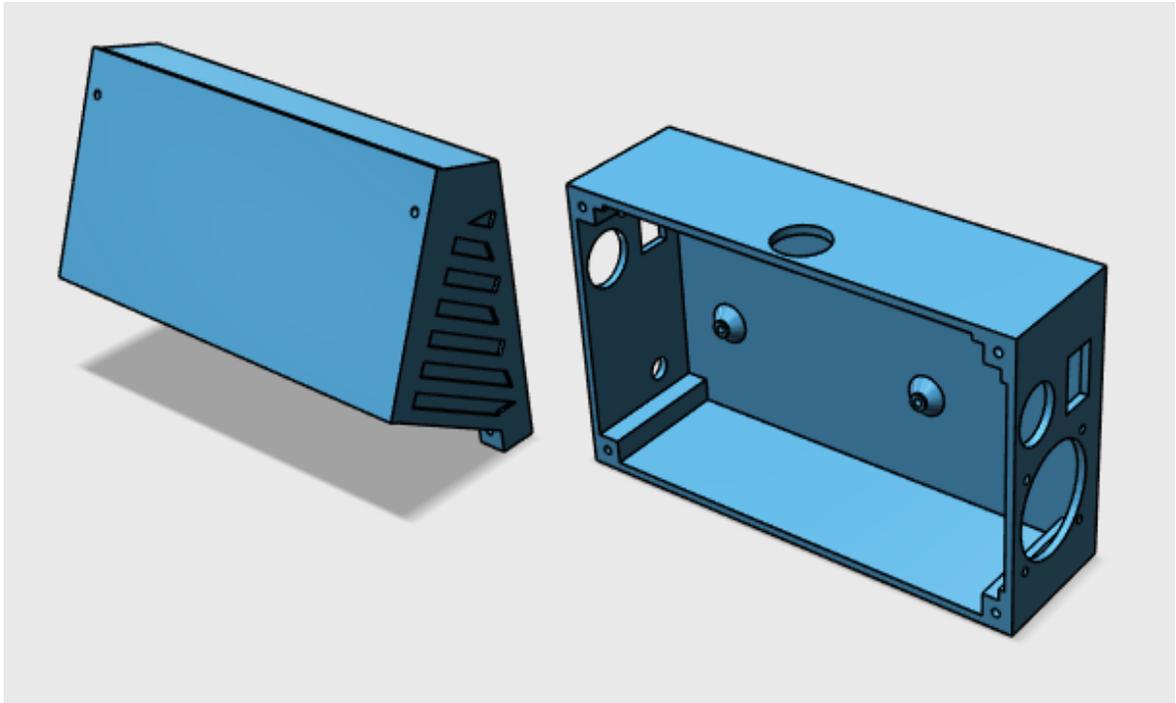
(B)

Figura 2-9: Diseño (A) y resultado (B) de unión perfiles de aluminio de cámara

2.4.5 Caja para la Electrónica

Diseñar una caja en 3D para la placa Arduino y la tarjeta RAMPS se vuelve necesario ya que, aparte de entregar orden y un aspecto más profesional, es necesario colocar un ventilador para evitar aumentos de temperatura, especialmente en los controladores de los motores, así como de un interruptor que desconecte la máquina de la alimentación. Así mismo se adquirió una fuente de alimentación de 12[V] con 5[A], lo que entrega un excelente desempeño.

El diseño de la caja se puede ver en la Figura 2-10 A y su resultado en la Figura 2-10 B.



(A)



(B)

Figura 2-10: Diseño (A) y resultado (B) de la Caja para la electrónica

2.4.6 Herramientas Intercambiables

Una de las partes fundamentales de la máquina y que entrega la versatilidad de sus aplicaciones, es una pieza que se ubique en el eje Z y que tenga la capacidad de que se le puedan integrar distintas piezas intercambiables para ejercer las variadas funcionalidades futuras de la máquina.

Es necesario también que esta pieza tenga terminales alimentados para las distintas herramientas que puedan ser montadas en los pines auxiliares de la tarjeta RAMPS. Desde la tarjeta RAMPS se pueden obtener 3 tipos de salidas: GND, +12[V] y +5[V], permitiendo estas muchas combinaciones de herramientas, tales como el electroimán, servomotores, leds, sensores, entre otras, siempre y cuando no se exija más corriente de la que la alimentación entrega.

Se diseña entonces el módulo para las herramientas intercambiables, permitiendo que la incorporación y extracción de las diferentes herramientas se realice de forma fácil y fluida gracias a los conectores banana. Adicionalmente se le agregan distintos conectores de voltaje (de +5V y +12V) para permitir que las herramientas se puedan energizar de forma fácil mediante los mismos conectores banana macho y hembra.

El diseño del soporte para las herramientas intercambiables y su resultado se puede ver en la Figura 2-11. Este soporte permite la conexión de una herramienta de +12V, una herramienta de +5V y sus correspondientes GND. Todas estas herramientas son configurables mediante códigos G, permitiendo el control absoluto de estas mediante la interfaz.

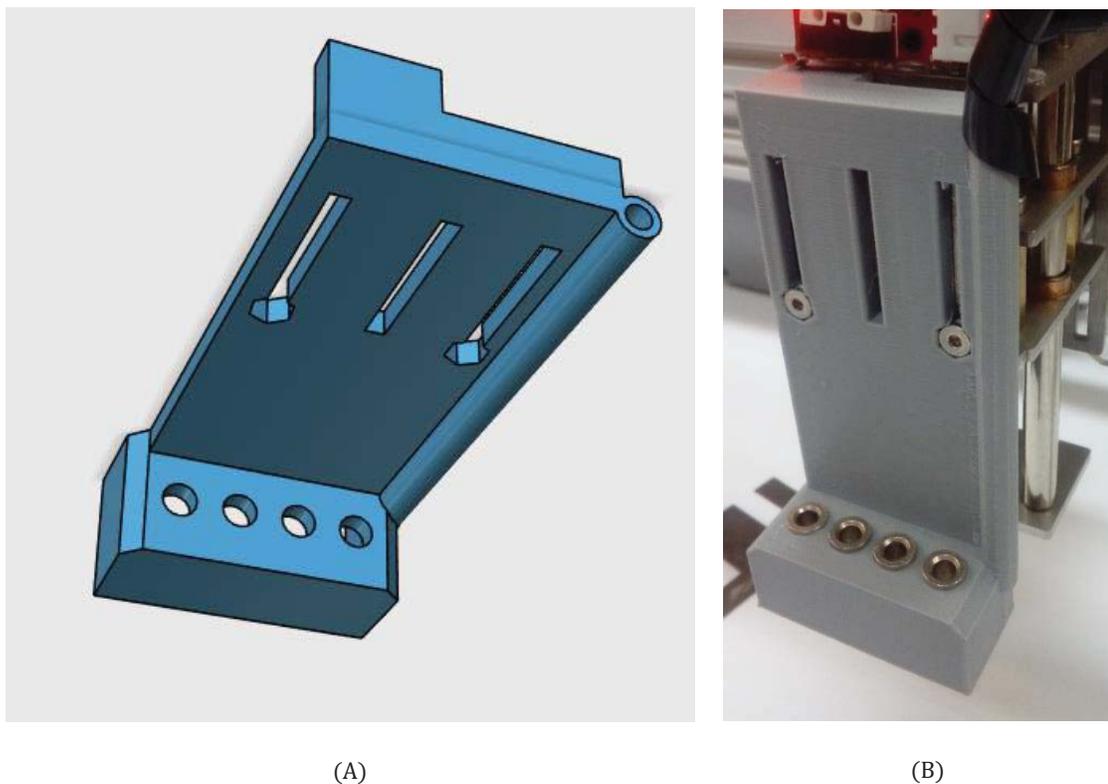
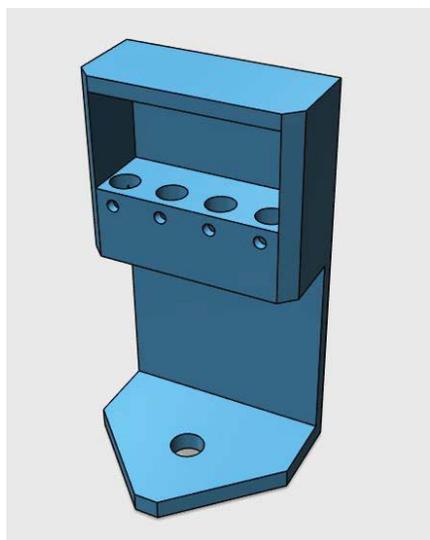


Figura 2-11: Diseño (A) y resultado (B) del soporte del eje Z

El diseño y resultado del Módulo Electroimán se pueden ver en la Figura 2-12. Este está alimentado por +12[V] y su GND. Este módulo permite moverse verticalmente en 33[mm], siendo esta distancia la óptima para poder obtener piezas metálicas planas desde la plataforma.



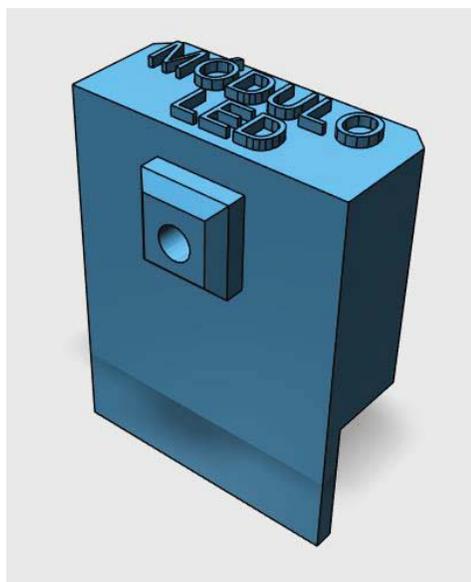
(A)



(B)

Figura 2-12: Diseño (A) y resultado (B) del Módulo Electroimán

El diseño y resultado del Módulo LED se pueden ver en la Figura 2-13. Este módulo se alimenta con +5[V] y su GND y su función es básicamente demostrar que se puede realizar una acción con una herramienta de +5[V] en el futuro. Este módulo permite bajar los 52[mm] del eje sin tocar la base.



(A)



(B)

Figura 2-13: Diseño (A) y resultado (B) del Módulo LED

El diseño y resultado del Módulo Escritura se pueden ver en la Figura 2-14. La función de este módulo es poder colocar un lápiz en el espacio habilitado para así, poder dibujar con la máquina. Las dimensiones permitidas por esta herramienta son 48[mm].

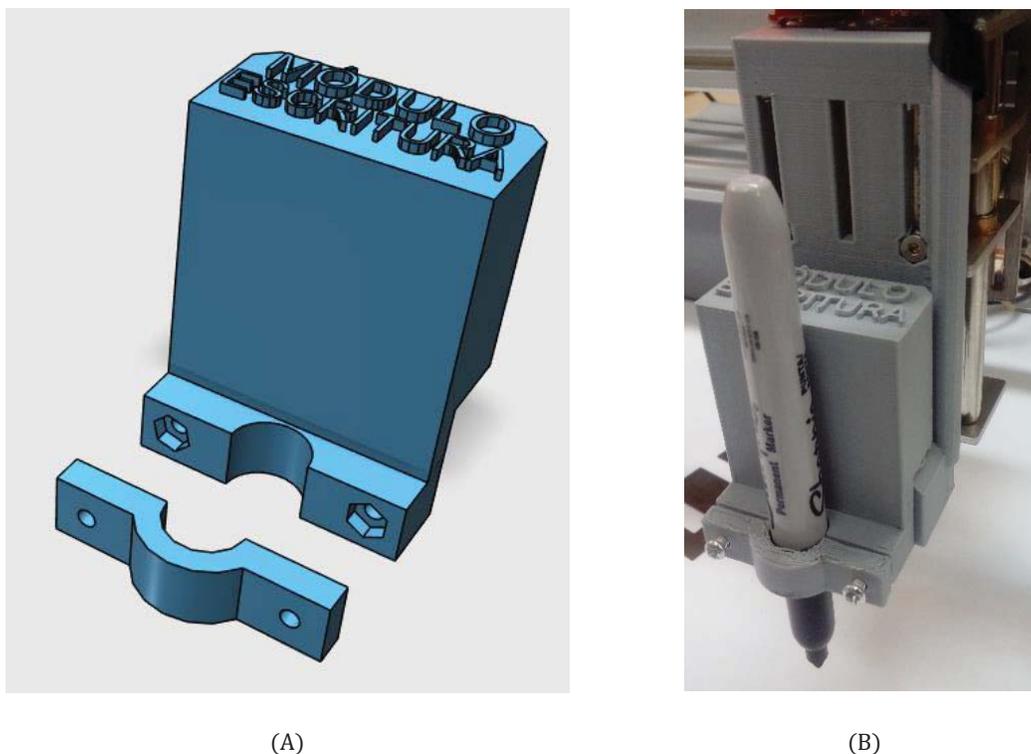


Figura 2-14: Diseño (A) y resultado (B) del Módulo Escritura

2.5 Electrónica y Cableado

Respecto a los cables de la estructura (de los motores y de los sensores “Final de Carrera”), primero se realiza la medición de longitud que cada cable requiere para que estos no queden tirantes cuando la estructura alcance los límites máximos de movimiento.

Es recomendable que los cables sean del tipo AWG 24 o 26 (*American Wire Gauge*), cables bastante comunes en el mercado. Estos ofrecen excelentes características para el proyecto, tales como baja resistencia interna, menos susceptibilidad a la interferencia y manejo de corrientes adecuadas para este proyecto. En este caso se escogió cable del tipo AWG 26, sus características más relevantes se ven en la Tabla 2-8.

Tabla 2-8: Características cable AWG 26

Característica	Valor
Diámetro [mm]	0.4049
Área [mm ²]	0.129
Resistencia eléctrica en cobre [Ω /1 km]	133.891

En la Figura 2-15 se muestra un diagrama de la tarjeta RAMPS con las conexiones para los motores,

mostrando no solamente donde va cada dispositivo conectado, sino que también el código de colores de los cables para evitar confusiones.

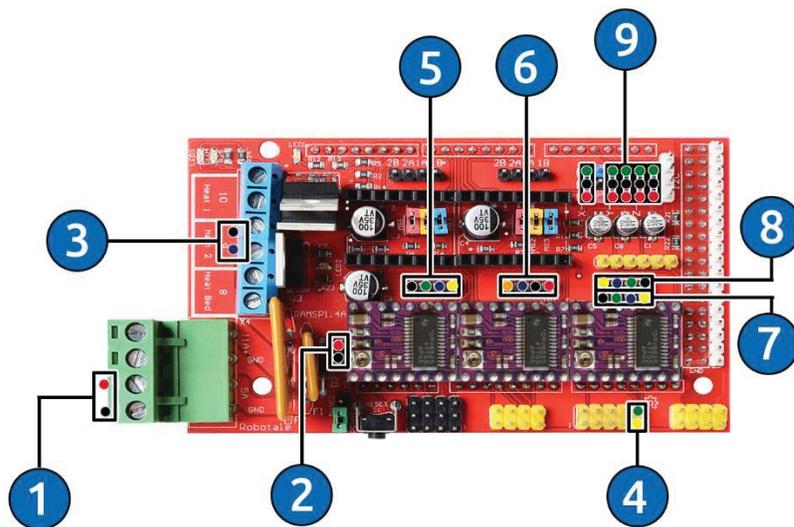


Figura 2-15: Conexión tarjeta RAMPS

En la Tabla 2-9 se puede ver el detalle de cada uno de los componentes de este diagrama de conexión de la tarjeta RAMPS.

Tabla 2-9: Conexión tarjeta RAMPS

#	Componente
1	Alimentación +12[V]
2	Ventilador +12[V]
3	Pin de +12[V] para herramienta
4	Pin de +5[V] para herramienta
5	Motor eje Z
6	Motor eje Y
7	Motor eje X (izq)
8	Motor eje X (der)
9	Pines sensores "Final de Carrera"

3 Movimiento de la Estructura

En orden de lograr un movimiento correcto de la estructura, una tarea fundamental es calibrar cada uno de los componentes para que la máquina responda de forma correcta a los valores medidos.

3.1 Área de Trabajo

A pesar de que el área interna de la estructura es de 680x590[mm], el área de trabajo que se puede utilizar es de 616x405[mm] y 52[mm] en Z, esto debido al espacio que utiliza el carro del eje X y su consecuente influencia en la visión de la Webcam.

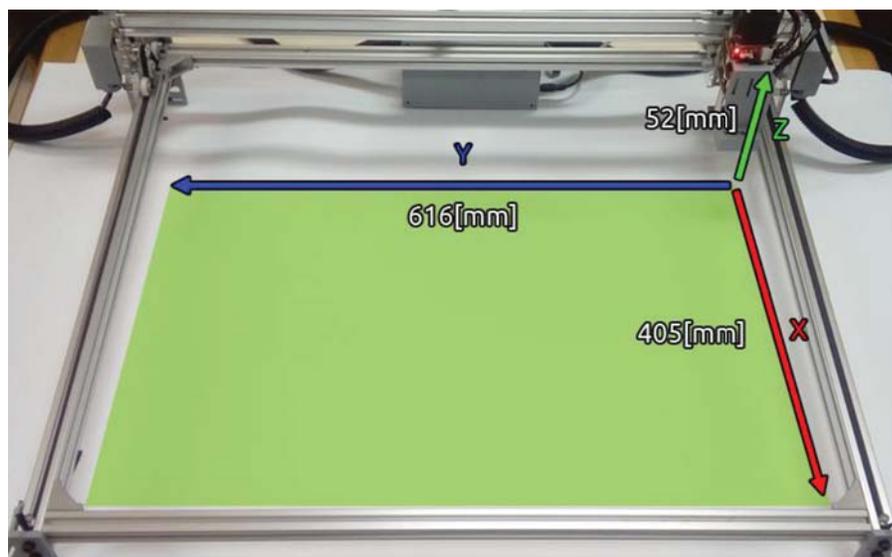


Figura 3-1: Área efectiva de trabajo en los ejes

Las dimensiones mostradas en la Figura 3-1 limitarán el movimiento y deberán ser incluidas en la configuración de Marlin, como se verá más adelante.

3.2 Componentes Electrónicos

El entendimiento de los componentes electrónicos, así como de su calibración, son pasos fundamentales para comprender la lógica de movimiento de la máquina.

3.2.1 Arduino

La placa de desarrollo Arduino MEGA (Figura 3-2) es una plataforma física computacional *open-hardware*, que cuenta entre muchas otras cosas, con un microcontrolador ATmega2560, puertos digitales y analógicos de entrada/salida (que permiten la conexión de distintos shields que aumentan las capacidades de uso del Arduino) y puerto de conexión USB que no solo sirve para alimentar a la placa, sino que también para programarla (en código C) desde una interfaz de usuario muy fácil de usar.



Figura 3-2: Placa de desarrollo Arduino Mega

En la Tabla 3-1: Características principales de la placa Arduino Mega se muestran sus principales características.

Tabla 3-1: Características principales de la placa Arduino Mega

Característica	Valor
Voltaje de Operación (V)	5
Voltaje de Entrada (V)	7-12
Pines digitales E/S (-)	54
Pines analógicos E/S (-)	16
Corriente DC por pin E/S (mA)	20
Velocidad de Reloj (MHz)	16

El rol de la placa Arduino es el de ejecutar el software, en este caso el software es Marlin (un código que combina programación C y C++ y que será detallado más adelante) y se encuentra almacenado en el microcontrolador, el cual por medio de la tarjeta RAMPS, es el que realizará el control de la planta.

3.2.2 Shield RAMPS

Para este proyecto se utiliza la tarjeta RAMPS (RepRap Arduino Mega Pololu Shield) en su versión 1.4 (Figura 3-3). Esta tarjeta está diseñada para controlar toda la electrónica necesaria para crear impresoras 3D. La facilidad de implementar la tarjeta RAMPS es el motivo por el cual se utilizará para este proyecto. Es necesario que posteriormente se editen sus parámetros por medio de software para que cumpla los requerimientos que se necesitan en el proyecto y no los de una impresora 3D.

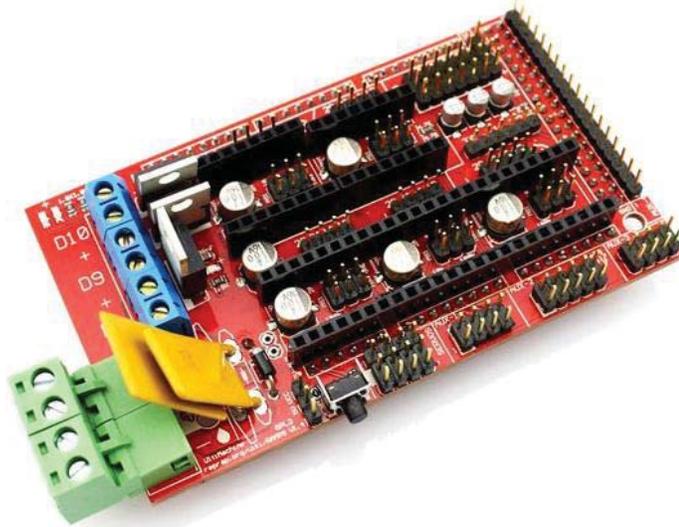


Figura 3-3: Tarjeta RAMPS v1.4.

Esta tarjeta es un shield para Arduino Mega, la cual tiene sus pines listos para conectar los drivers de los motores y demás necesidades.

Dentro de las características relevantes para el proyecto, se encuentran:

- Expandible para controlar otros accesorios.
- Pueden conectarse hasta 5 drivers de motores.
- Permite la conexión de 2 motores para un solo eje (el eje X).
- Fusible a 5[A] para mayor seguridad y protección de componentes.

3.2.3 Motores Paso a Paso

Los Motores Paso a Paso (Figura 3-4) son dispositivos electromecánicos que convierten, de manera similar a un convertor digital-analógico, impulsos eléctricos en desplazamientos angulares discretos (pasos), en otras palabras, es capaz de girar a ciertos ángulos dependiendo de sus valores de entrada.

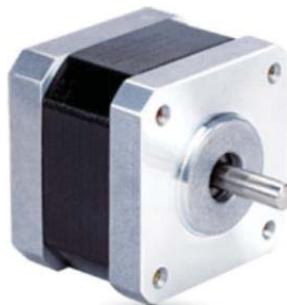


Figura 3-4: Motor Paso a Paso NEMA 17

Estos motores son utilizados en diversas aplicaciones, tales como robots, drones, automatización, entre otros [16]. Lo que lo hace perfecto para estas aplicaciones son sus características, comunes entre los motores paso a paso:

- No tienen escobillas: una de las características más propensas a errores, crean arcos eléctricos indeseados, y a veces, peligrosos en ciertas aplicaciones.
- Independiente de la carga: los motores paso a paso funcionarán a la velocidad deseada sin importar la carga, siempre y cuando esta no supere los valores recomendados de torque del motor.
- Posicionamiento en lazo abierto: Al moverse en pasos cuantificados, la posición de los motores siempre es conocida, sin necesidad de un elemento de realimentación.
- Par de retención: Son capaces de mantener el eje estacionario.
- Excelente respuesta: para iniciar, parar y revertir el movimiento.

Existen 3 tipos básicos de motores paso a paso: de reluctancia variable, imán permanente e híbridos, estos dos últimos pueden ser unipolares o bipolares.

1. Los motores de reluctancia variable (Figura 3-5) tienen de 3 a 5 devanados conectados a un terminal común en el estator y un rotor multipolar de hierro. Estos motores rotan cuando los dientes más cercanos del rotor son atraídos a los devanados energizados del estator, obteniéndose así la ruta con menor reluctancia. Cuando los devanados no están energizados, el par estático de estos motores es nulo [16].

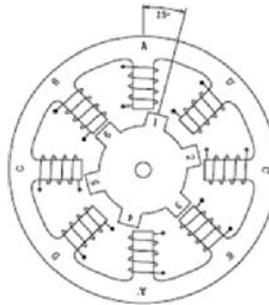


Figura 3-5: Rotor y estator de un motor de reluctancia variable

2. Los motores de imán permanente permiten mantener un par estático diferente a cero cuando los devanados del estator no están energizados. Generalmente el rotor tiene forma de disco y se encuentra magnetizado.
3. Los motores híbridos (Figura 3-6) combinan características de ambos motores mencionados anteriormente: tiene un estator dentado y el rotor contiene dos piezas de polos separados por un imán permanente con los dientes opuestos desplazados.

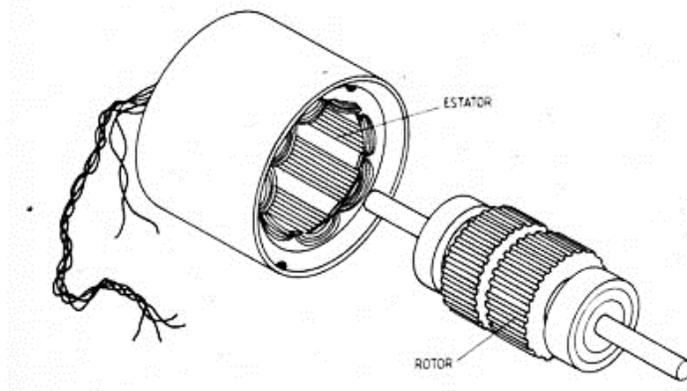


Figura 3-6: Rotor y estator de un motor híbrido

Los motores unipolares, si bien son más fáciles de implementar (no requiere de un controlador que incluya un puente H para utilizarlo), el problema que tienen es que el torque generado por ellos es bastante menor que el ofrecido por los bipolares, por lo que son utilizados en aplicaciones de bajo torque [16]. Es por esto que, en este proyecto se utilizarán motores paso a paso bipolares.

La diferencia entre los devanados de un motor unipolar y un motor bipolar se pueden ver en la Figura 3-7.

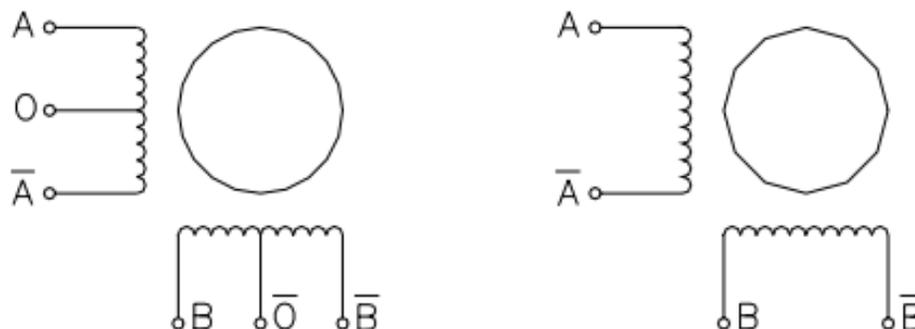


Figura 3-7: Devanados de un motor unipolar y de un motor bipolar

Para poder operar los motores paso a paso bipolares, es necesario un controlador IC (Integrated Circuit), que incluya un puente H, esto principalmente debido a que, en orden de revertir la polaridad del estator, la corriente necesita revertirse. Existen también otras razones para las cuales se requiere de un puente H:

1. El consumo de corriente es relativamente alto.
2. Las bobinas del estator son básicamente inductancias, por lo que, cuando las corrientes en el inductor cambian de dirección, se genera un peak, este peak es controlado por el puente H.

Para efectos de este trabajo se cuenta con motores paso a paso híbridos bipolares de modelo 17HS4401 (NEMA 17), cuyas características se ven en la Tabla 3-2 [17].

Tabla 3-2: Características principales motor paso a paso 17HS4401

Característica	Valor
Angulo por paso (°)	1.8
Pasos (-)	200
Longitud (mm)	40
Corriente nominal (A)	1.7
Resistencia de fase (Ω)	1.5
Inductancia de fase (mH)	2.8
Torque mínimo (N×cm)	2.2
Torque máximo (N×cm)	40
Peso del motor (g)	280

La NEMA (National Electric Manufacturers Association – USA) crea estándares para que se sigan ciertas normas de diseño en la manufactura de distintos dispositivos, incluyendo los motores paso a paso. En este caso la clasificación NEMA 17 de los motores 17HS4401 corresponde al tamaño de la placa

principal en pulgadas, en este caso 1.7"x1.7". Esto se utiliza bastante, dado que cualquier motor de la misma clasificación NEMA, puede reemplazar a otro motor de la misma clasificación en una máquina [18].

La clasificación NEMA solamente habla de sus dimensiones y no implica que un motor con mayor dimensión sea más poderoso en términos de torque. Es perfectamente posible que un motor NEMA 14 "tire" de un motor NEMA 17 [19].

En cuanto al controlador utilizado para controlar los motores, se utilizará en este proyecto el DRV8825 de Texas Instruments (Figura 3-8), este está especialmente diseñado para controlar motores paso a paso bipolares, para esto cuenta con 2 puentes H y un indexador de micropasos.

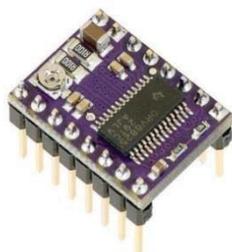


Figura 3-8: Controlador DRV8825



Figura 3-9: Controlador A4988

Se ha escogido este controlador por sobre el más popular Pololu A4988 (Figura 3-9). En la Tabla 3-3 se pueden ver sus características más relevantes comparadas [20] [21].

Tabla 3-3: Comparación entre controladores A4988 y DRV8825

Característica	A4988	DRV8825
Precio aproximado	6.80 USD las 5 unidades	10 USD las 5 unidades
Máxima corriente	2[A]	2.5[A]
Máximos micropasos	16	32
Color PCB	Verde/Roja	Morada
Protección contra temperatura	Si	Si

Si bien las diferencias son sutiles, la resolución de 32 micropasos máximos que ofrece DRV8825 y, además, la máxima corriente que soporta lo hace una alternativa muy atractiva para este proyecto.

Las características más destacables de estos controladores son:

- Entregan una interfaz simple de control de pasos y de dirección.
- Un potenciómetro que permite ajustar fácilmente la salida de corriente del motor.
- Ofrecen la posibilidad de seis resoluciones de paso: paso completo, medio paso, 1/4 de paso, 1/8 de paso, 1/16 de paso y 1/32 de paso.
- Seguridad: protección de corriente, voltaje y apagado por sobrecalentamiento, etc.

Uno de los pasos más importantes en el uso de los motores paso a paso, es definir bien la corriente que circula por ellos y por su controlador.

Para calibrar la corriente por los controladores se sigue el siguiente procedimiento (siempre desconectando la alimentación de 12[V] entre cada paso):

1. Se desconectan todos los dispositivos de la tarjeta RAMPS y se mide la corriente en la entrada con un multímetro, este valor de corriente debería ser de 90[mA].
2. Se conecta el controlador del eje Y y el respectivo motor, se ajusta el potenciómetro del controlador cuidadosamente hasta obtener un valor de 340[mA] (90[mA] + 250[mA]). 250[mA] es la corriente recomendada para que el motor se mueva de manera correcta. Este paso se repite para el eje Z.
3. Para el eje X se deben conectar ambos motores y ajustar el potenciómetro hasta obtener una corriente de 490[mA] (90[mA] + 400[mA]). Intuitivamente se podría decir que la corriente circulante por este eje debería ser de 500[mA], pero al moverse ambos motores de forma sincronizada, esto no es necesario.
4. Con todos los elementos conectados, la lectura de corriente debería ser de 990[mA] hasta 1.1[A].

3.2.4 Sensores “Final de Carrera”

Los sensores “Final de Carrera” son dispositivos de naturaleza mecánica, electrónica o incluso neumática, que permiten saber si una pieza móvil ha alcanzado cierto punto de su recorrido, esto al cambiar su estado natural. Para este proyecto se utiliza un “Final de Carrera” tipo mecánico, modelo Mech Endstop v1.2 de MakerBot (Figura 3-10), este es uno de los modelos más básicos de “Final de Carrera” y es capaz de cambiar sus contactos internos (NA: normalmente abierto o NC: normalmente cerrado) al recibir una presión ejercida en su brazo metálico.



Figura 3-10: Sensor “Final de Carrera”

Para utilizar los “Finales de Carrera”, es necesario situarlos en los extremos de los ejes de la máquina. El objetivo principal de utilizar finales de carrera es evitar que la máquina pudiera estropearse debido a que los motores se intentarían mover más allá de los límites del área de trabajo. Otro objetivo es que permite calibrar de forma correcta el dispositivo, en caso de que, se pierdan pasos en las correas con el uso de la máquina, esto permitiría eventualmente configurar los pasos máximos y mínimos de cada eje.

La tarjeta RAMPS tiene una sección dedicada para los ejes de carrera mostrada en la Figura 3-11. Estos pines representan a la parte inicial y la final de cada uno de los 3 ejes.

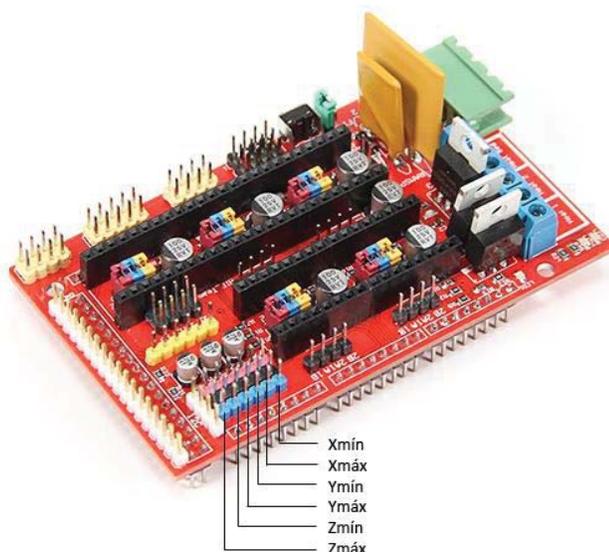


Figura 3-11: Pines para finales de carrera en la tarjeta RAMPS

En el firmware Marlin, estos fines de carrera se pueden activar o desactivar y programar en función a si se estos se encuentran en modo NA o NC.

3.3 Firmware Marlin

Marlin es un firmware para electrónica tipo RepRap (máquinas auto replicables). Contiene todo lo necesario para configurar de forma rápida y fácil una impresora 3D, tales como control de funcionamiento de los motores, controlador PID para la temperatura, interfaz por pantalla LCD, capacidad de memoria SD, etc.

Este software debe ser estudiado con detención, para su posterior modificación, esto debido a que está pensado y programado para controlar una impresora 3D. Para este proyecto se usará exclusivamente la parte de control de motores y requerirá de cambios sustanciales para adaptar otros elementos a utilizar, como, por ejemplo, el uso de las herramientas intercambiables.

3.3.1 Pasos máximos por eje

Para calcular los pasos del motor y por ende, para configurar de forma correcta, en el firmware, las características de cada eje, es primordial obtener las medidas utilizables de la máquina, en este caso el espacio efectivo de la máquina se puede ver en la Figura 3-1.

En Marlin, específicamente en el archivo **configuration.h**, se encuentran las dimensiones máximas de los ejes (Listado 3-1), estas deben ser las mismas que las dimensiones de trabajo efectiva de la plataforma.

Listado 3-1: Dimensiones máximas de cada eje

```
736 #define X_MIN_POS 0
737 #define Y_MIN_POS 0
738 #define Z_MIN_POS 0
739 #define X_MAX_POS 405
740 #define Y_MAX_POS 616
741 #define Z_MAX_POS 52
```

3.3.2 Pasos por unidad

Estos son los pasos por unidad, de cada eje. Esto quiere decir, la cantidad de pasos que los motores avanzan por cada milímetro (Listado 3-2).

Listado 3-2: Pasos por unidad de cada eje

```
492 #define DEFAULT_AXIS_STEPS_PER_UNIT { 161.14, 160.31, 1562.5, 0 }
```

Para obtener este valor, se debe mover la estructura una cantidad de pasos conocida en longitud, por ejemplo, mover la estructura en x[cm] y haciendo uso de un lápiz, marcar una línea en una hoja, se mide la línea y luego se utiliza una simple regla de tres:

$$\text{Pasos por unidad reales} = \frac{\text{Pasos por unidad actuales} * \text{Distancia ideal}}{\text{Distancia real}}$$

Se reemplaza este valor obtenido en Marlin y se repite esta medición hasta que la longitud de la línea sea la misma que la longitud deseada.

3.3.3 Velocidad Máxima de Movimiento

Estos son los valores de velocidad de cada eje en [mm/s] (Listado 3-3).

Listado 3-3: Velocidad de cada eje

```
502 #define DEFAULT_MAX_FEEDRATE { 500, 500, 10, 0 }
```

3.3.4 Aceleración Máxima

Estos son los valores de aceleración de cada eje en [mm/s²] (Listado 3-4).

Listado 3-4: Aceleración de cada eje

```
511 #define DEFAULT_MAX_ACCELERATION { 1000, 1000, 15, 0 }
```

4 Códigos G

El movimiento de la máquina se ejecuta mediante una serie de instrucciones llamadas “códigos G”, a continuación, se muestran sus detalles más relevantes.

4.1 Introducción a los Códigos G

Los Gcodes o códigos G, también conocidos por RS-274, es el lenguaje de programación para control numérico más utilizado en la industria. Estos códigos permiten controlar la máquina herramienta con instrucciones predeterminadas que no solo indican “qué hacer” sino que también “cómo hacerlo”.

Los códigos G no son totalmente compatibles con todos los sistemas, afortunadamente para este trabajo, el firmware Marlin tiene soportado la mayoría de los códigos G.

Estos códigos se componen de una letra y luego de números, estos pueden ser enteros o racionales. A pesar de llamarse códigos G, cualquier letra del alfabeto puede ser un tipo de comando en código G [22]. El formato de alguno de ellos es explicado en la Tabla 4-1 [19].

Tabla 4-1: Tipo de códigos G

Letra	Significado
Gnn	Comando Gcode estándar, indica cómo moverse hasta un punto
Mnn	Comando definido por la comunidad RepRap
Tnn	Seleccionar la herramienta nn
Snn	Parámetro de comando, como la tensión enviada a un motor
Pnn	Parámetro de comando, como el tiempo en milisegundos
Xnn	Una coordenada en el eje X
Ynn	Una coordenada en el eje Y
Znn	Una coordenada en el eje Z
Inn	Parámetro
Jnn	Parámetro
Fnn	Feedrate o velocidad de movimiento en mm por minuto

Una vez enviado los códigos necesarios, la máquina responde de diversas formas, las respuestas más comunes se pueden ver en la Tabla 4-2.

Tabla 4-2: Respuestas típicas en Marlin

Respuesta	Explicación
ok	El código se recibió de manera correcta
wait	En espera
busy: processing	El comando está siendo ejecutado

4.2 Lógica de Movimiento

Los códigos G se basan en instrucciones a las que el actuador responde “fielmente”, esto es, sigue la instrucción designada pase lo que pase (por ej. si se cruza algo en el camino de la herramienta a manejar, esta intenta si o si llegar a su destino, pudiendo resultar en averías de la máquina o la herramienta utilizada).

Generalmente, las máquinas CNC no tienen un control en lazo cerrado [23], por lo que solo dependen de la configuración y calibración que tienen actualmente, entonces, es necesario tomar ciertas precauciones al momento de utilizarlas.

4.2.1 Punto Home

Una de las principales medidas que se debe tomar en una máquina CNC es la de llevar al punto Home (0, 0, 0) a la máquina al momento de iniciarla [22]. Esto debido a que, al encenderse, la máquina “piensa” que se encuentra en el punto Home, llevando a posibles problemas de movimiento (por ej. si un eje tiene un límite de 600[mm] y al encenderse la máquina se encuentra en 300[mm], se le puede enviar el comando de moverse 600[mm] y la máquina intentará avanzar a 900[mm]).

Para los problemas de avanzar más allá de los límites permitidos, los sensores “Final de Carrera” son indispensables, pero, de todas formas, no se puede manipular una máquina sin un punto inicial real.

El comando para enviar a la máquina al punto Home es *G28*, si este se acompaña adicionalmente de uno o más ejes, solo llevará al punto inicial a los ejes comentados (por ej. *G28 X Y* llevará al punto inicial solamente al eje X e Y).

4.2.2 Posicionamiento

Los códigos G ofrecen dos formas de posicionamiento, el posicionamiento absoluto (*G90*) o posicionamiento relativo (*G91*). El primero se basa en que todos los movimientos realizados se hagan desde el punto inicial. El relativo se basa en que todos los movimientos realizados se hagan desde el punto actual.

Por ejemplo:

```
G90
G1 X50
G1 X75
```

Este código terminaría en el punto X: 75

```
G91
G1 X50
G1 X75
```

Este código terminaría en el punto X: 125

4.2.3 Movimiento

Como se comentó anteriormente, la máquina sigue el comando recibido sin importar lo que pase, y no tiene un método de corregir el movimiento una vez que este se ha enviado.

Los códigos G son enviados al actuador y estas son leídas línea por línea, empezando por la letra del código hasta el salto de línea que indica que el comando terminó de enviarse [24]. Esto hace que la máquina lea sin parar cada línea y ejecutándola al terminar el movimiento de la anterior, sin permitir que se pueda detener o modificar en el medio (a menos de que se apague la máquina).

A pesar de que generalmente los códigos G son generados por software que siguen ciertas reglas para hacer lo requerido de forma óptima, cuando una persona crea los códigos G es indispensable tomar en cuenta todas las implicancias que un movimiento puede ejecutar (por ej. es muy importante pensar que, en caso de estar dibujando con un lápiz, el eje Z se tiene que levantar entre los trazos que se quieren dibujar).

Es importante recordar que cuando se genera un código grande, se pueden comentar, tanto en una línea nueva, como al final de un comando utilizando el punto y coma (;).

5 Interfaz de LabVIEW

En este capítulo se comentarán los aspectos técnicos del desarrollo de cada una de las secciones de la interfaz de LabVIEW, haciendo un paralelismo entre el aspecto visual, así como de su funcionamiento mediante los respectivos diagramas de flujo.

5.1 Información General

La interfaz de LabVIEW es la pieza clave para controlar de manera adecuada la máquina, esta busca entregar una interfaz amigable, intuitiva y fácil de utilizar por parte del usuario, permitiendo tanto que la máquina cumpla sus objetivos, como también protegiendo a la estructura de cualquier movimiento mal introducido por el usuario. La comunicación se realiza mediante serial [25] y para ello se utiliza el toolkit NI-VISA.

La interfaz (Figura 5-1) contiene cada función de la máquina en distintas secciones, requiriendo por parte del usuario que prepare la estructura de forma adecuada para realizar la acción. En todo momento la aplicación muestra tres módulos básicos: la configuración inicial de la tarjeta, su consola de códigos G y la visión de la cámara web.

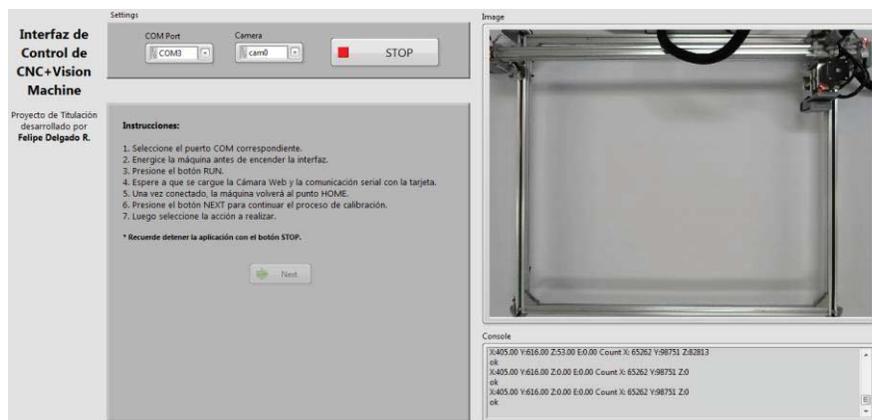


Figura 5-1: Panel frontal de la interfaz

La aplicación tiene un funcionamiento cíclico, donde constantemente recibe imágenes por parte de la Webcam y la respuesta de la tarjeta ante los códigos enviados. Al finalizar el programa pulsando el botón STOP, la aplicación se desconecta de la Webcam y cierra la sesión con el puerto COM. En la Figura 5-2 se puede ver un diagrama general de funcionamiento de la interfaz.

El bloque color rojo indica el proceso que se esté ejecutando en la interfaz. En el siguiente apartado estos procesos son detallados de forma individual.

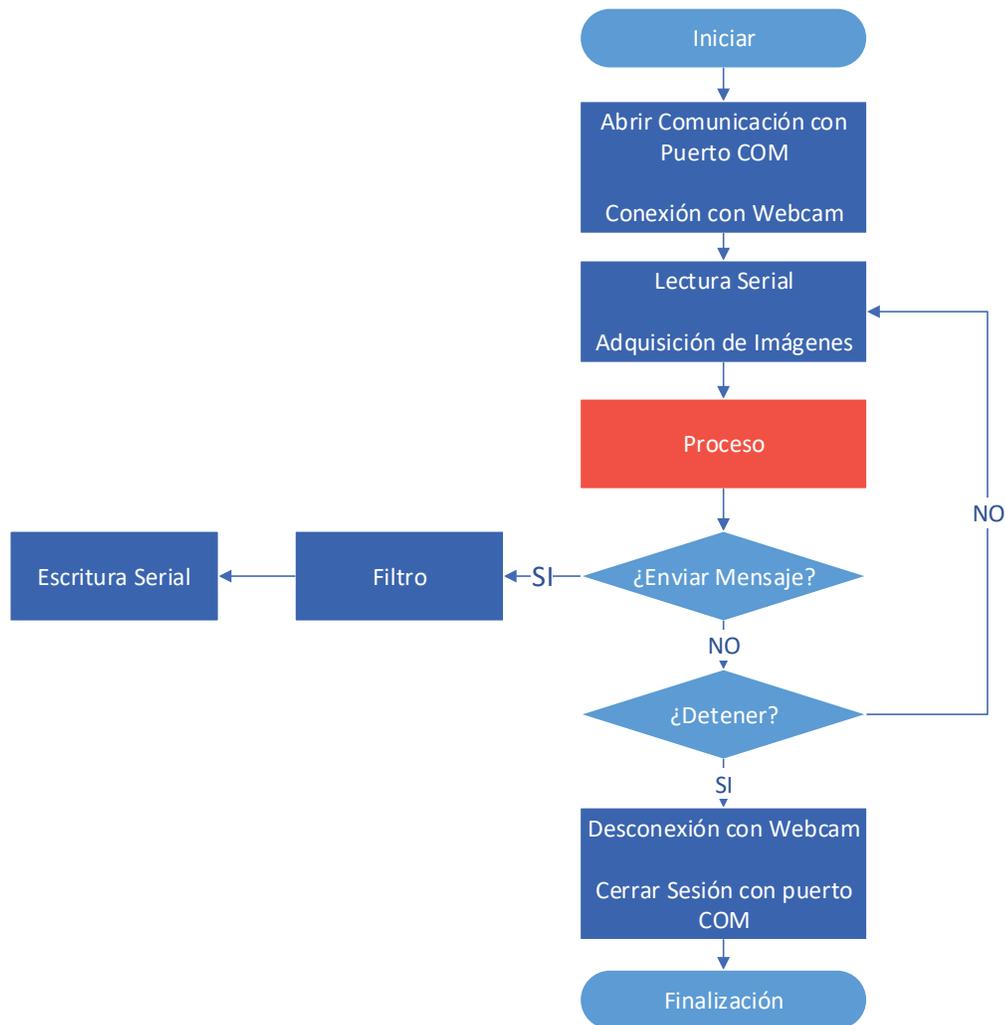


Figura 5-2: Diagrama de Flujo general de la aplicación

5.2 Procesos

Los “procesos” (cuadro rojo en la Figura 5-2) varían dependiendo de la sección en que se encuentre el usuario, a continuación se explican detalladamente cada uno de estos.

5.2.1 Instrucciones

La sección Instrucciones es básicamente una pantalla con instrucciones de uso de la interfaz, pero también cumple con una importante tarea: llevar a la máquina al punto inicial. Solo cuando la máquina alcanza el punto inicial permite continuar utilizando la interfaz. Esto es sumamente importante teniendo en cuenta que la máquina “piensa” que su punto inicial es la coordenada actual al encenderla, lo que podría llevar a problemas de movimiento posteriormente.

En la Figura 5-3, se puede ver el Diagrama de Flujo de esta sección.

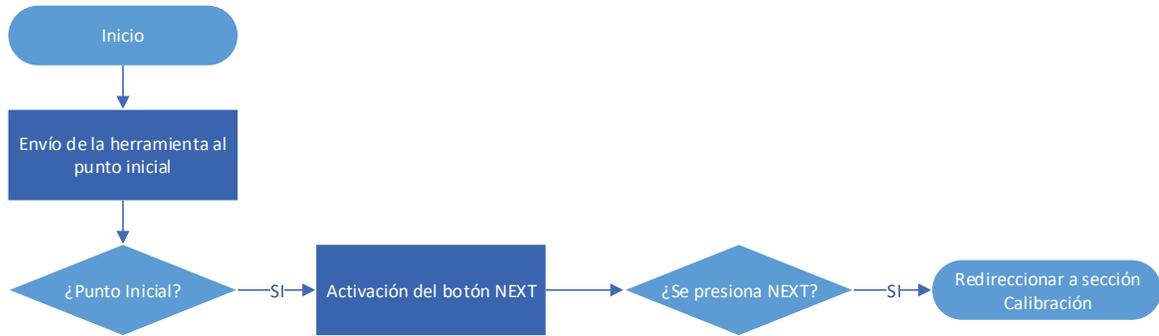


Figura 5-3: Diagrama de Flujo de la sección Instrucciones

5.2.2 Calibración

La sección Calibración es una antesala a la función a realizar con la máquina. Para esto, permite seleccionar la herramienta a utilizar, lo que redefine los límites máximos de movimiento con esa herramienta. Una vez que el usuario escoge la herramienta a utilizar, puede seleccionar la sección que utilizará.

En la Figura 5-4 se puede ver el Diagrama de Flujo de esta sección.

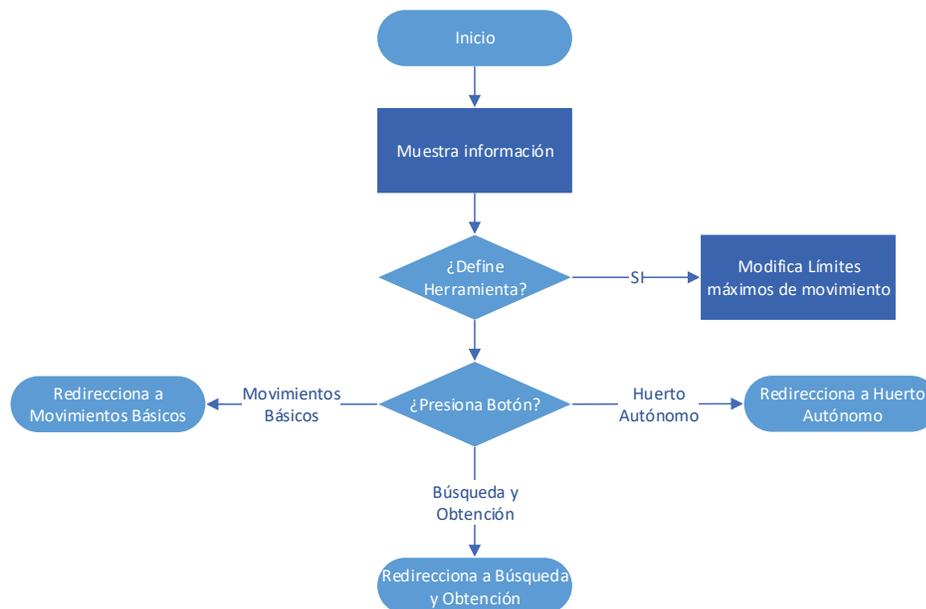


Figura 5-4: Diagrama de Flujo de la sección Calibración

5.2.3 Movimientos Básicos

La sección Movimientos Básicos permite al usuario enviar Códigos G simples, tales como, comandos prefabricados, líneas rectas y circunferencias, así como también, códigos que podrían ser más complejos en la opción de Códigos G Manuales.

Los comandos prefabricados son cuatro: enviar al punto inicial, obtener la posición actual, encender la herramienta de +5[V] y encender la herramienta de +12[V].

En el apartado de líneas rectas, la interfaz permite poner las coordenadas en los tres ejes para realizar un movimiento rectilíneo. Además, permite ajustar la velocidad del movimiento y la naturaleza de este, esto es, posicionamiento absoluto (desde el punto inicial) o relativo (desde el punto actual).

En el apartado de movimientos circulares, la interfaz permite ajustar las coordenadas X e Y del centro del círculo, la coordenada Z de la herramienta y el radio del círculo. Internamente la interfaz realiza un ajuste de estos parámetros para realizar el círculo deseado.

En el apartado de códigos manuales, la interfaz permite escribir los códigos G a ejecutar, revisando internamente cada línea para asegurarse de que, en caso de haber algún movimiento, chequear si supera los límites permitidos de movimiento.

En la Figura 5-6 se puede ver el Diagrama de Flujo de la sección Movimientos Básicos.

Para realizar la comprobación de movimiento dentro de los límites de movimiento se crearon 4 SubVI:

- Sub_Write: Archivo principal en el envío de códigos por serial, dentro de este se ejecutan los otros 3 SubVI.
- Sub_LineCount: Cuenta las líneas de código que fueron enviadas.
- Sub_Position: Guarda el tipo de posicionamiento (absoluto o relativo) de los códigos enviados.
- Sub_AxisCounter: Busca en cada línea de código el movimiento solicitado por el usuario comparándolo con los límites máximos que tiene asociado el eje, en caso de superar este límite, la línea de código no es enviada al actuador.

5.2.4 Búsqueda y Obtención

La sección Búsqueda y Obtención permite encontrar una figura predeterminada (template) dentro de la imagen de la Webcam, obtener su posición en la plataforma, ir a buscarla haciendo uso del Módulo Electroimán y finalmente devolverla en el punto inicial, un ejemplo de esta búsqueda se puede ver en la Figura 5-5.



Figura 5-5: Ejemplo de búsqueda de una imagen (template) dentro del área de trabajo

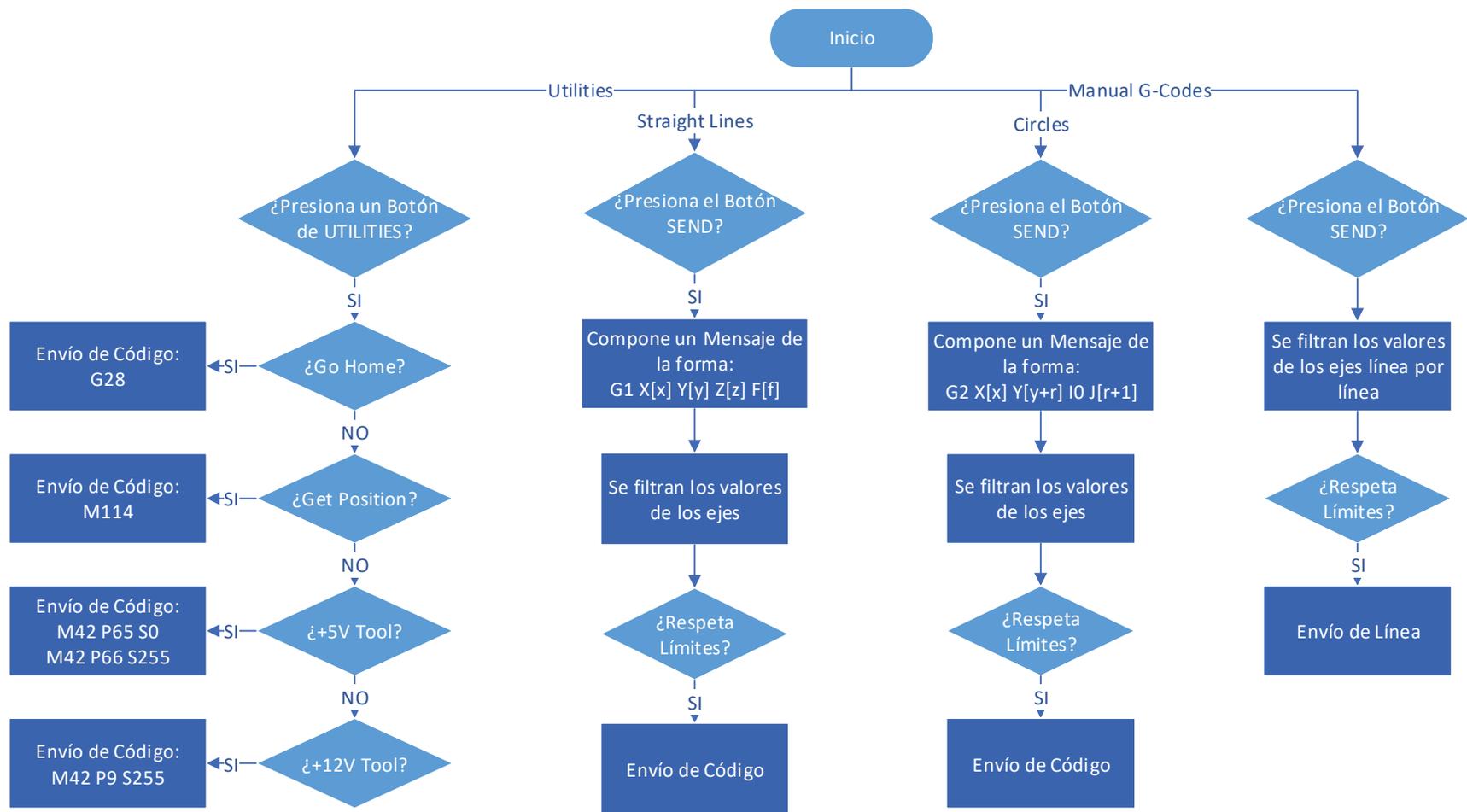


Figura 5-6: Diagrama de Flujo de sección Movimientos Básicos

Además, permite crear nuevos templates de forma sencilla, seleccionando el template desde la imagen vía Webcam, como se puede apreciar en la Figura 5-7.

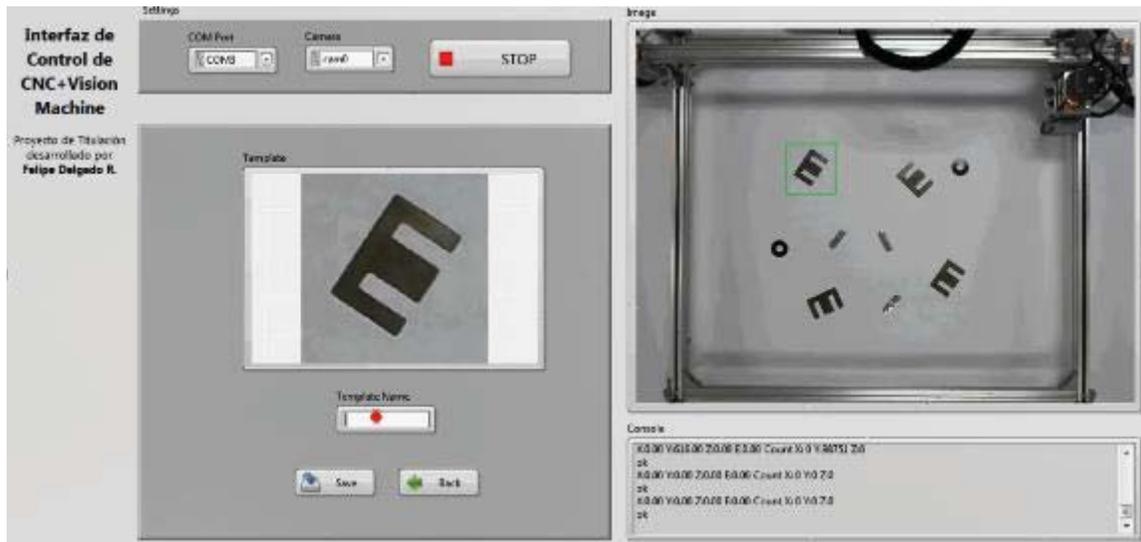


Figura 5-7: Creación de nuevo template desde la imagen capturada por la Webcam

Una vez realizada la búsqueda, la interfaz entrega las coordenadas de hasta 4 resultados y genera un Código G con el movimiento asociado.

En la Figura 5-8 se puede ver el Diagrama de Flujo de la sección Búsqueda y Obtención.

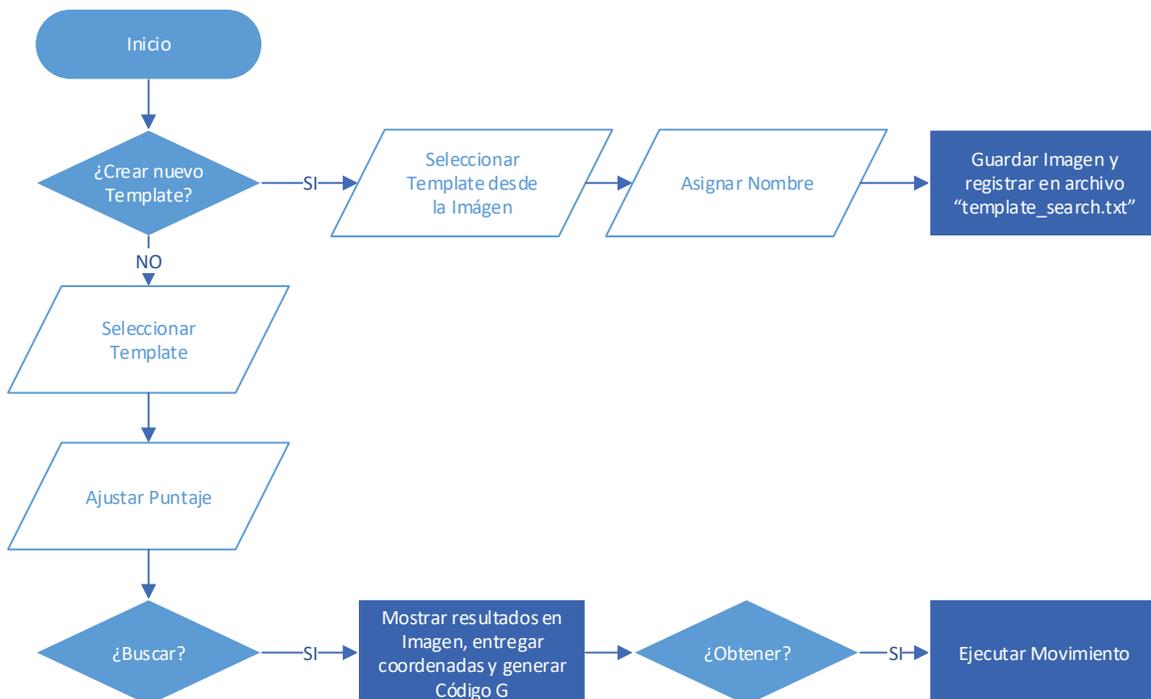


Figura 5-8: Diagrama de Flujo de sección Búsqueda y Obtención

5.2.5 Huerto Autónomo

La sección Huerto Autónomo permite realizar rutinas compuestas de códigos G predefinidos para actuar en un tipo de planta (identificada mediante reconocimiento de imágenes) de forma programada, el objetivo final es que la máquina se encargue de supervisar y controlar el estado de ciertos tipos de plantas de forma automática.

Cabe destacar que para esta aplicación se ha realizado una base inicial para que el desarrollo completo de esta nueva funcionalidad pueda ser realizada por otro estudiante y/o profesor en futuros trabajos.

Actualmente la interfaz permite monitorear un solo tipo de planta, realizando una actividad y congelando toda acción de la interfaz mientras se desarrolla la rutina.

En la Figura 5-9 se puede ver el Diagrama de Flujo de la sección Huerto Autónomo.

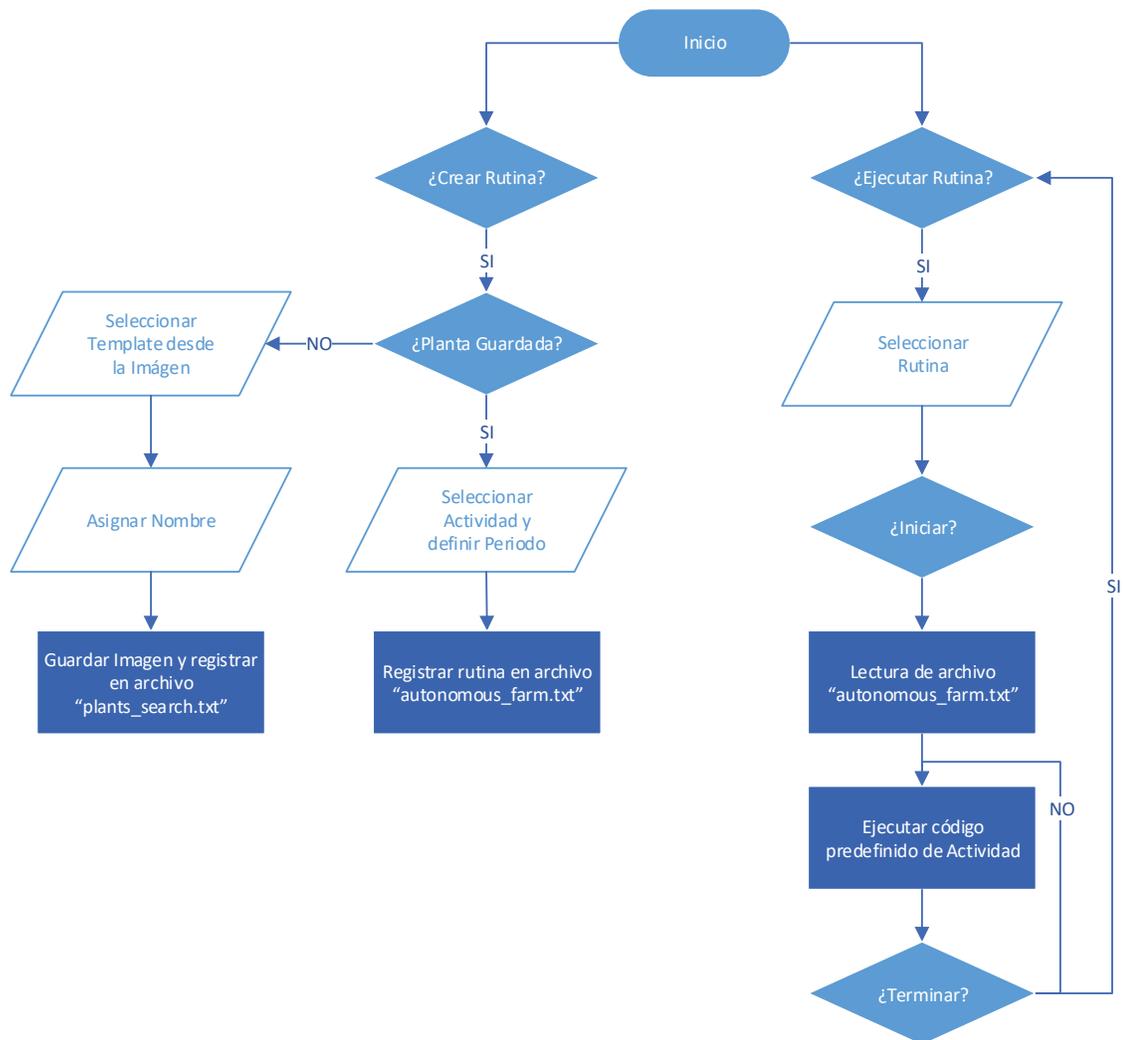


Figura 5-9: Diagrama de Flujo de sección Huerto Autónomo

Para crear una rutina nueva, el usuario escoge el “template” de la planta a controlar, selecciona la actividad a realizar y el periodo de repetición de la actividad.

Para iniciar una rutina existente, el usuario selecciona una rutina y presiona el botón de Inicio, la interfaz congela todas sus funciones durante el desarrollo de la rutina.

La secuencia de trabajo, básicamente, es identificar el “template” en la plataforma (Figura 5-10) y cuando transcurre el tiempo indicado, realizar cierta actividad prefabricada. Estas actividades buscan ser utilidades en un huerto, tales como riego, fumigación, etc.

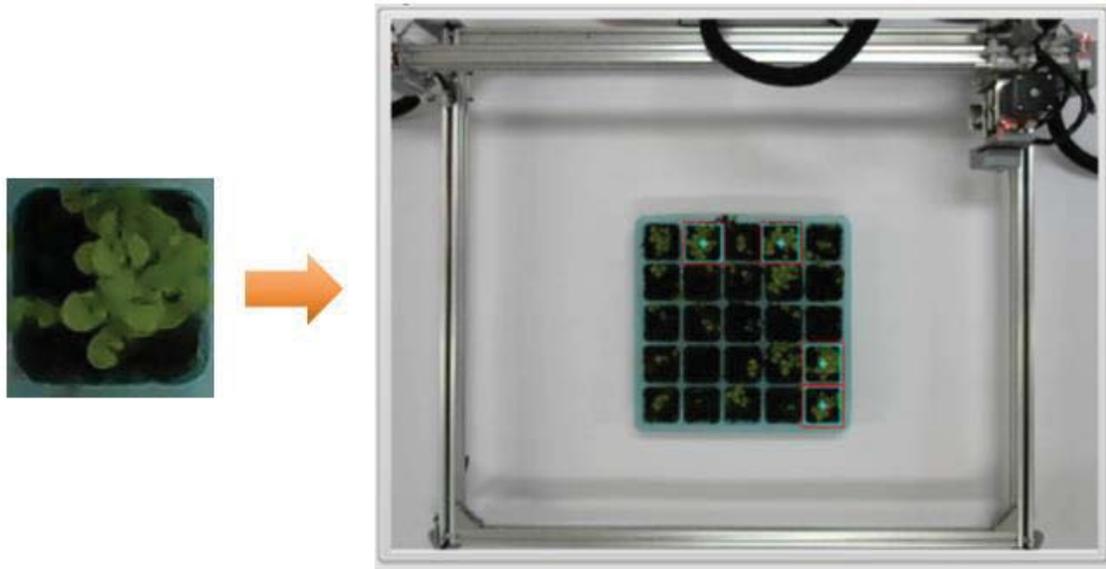


Figura 5-10: Muestra de un reconocimiento de una planta en un almacigo

Para el funcionamiento tanto de las secciones “Búsqueda y Obtención” y “Huerto Autónomo” se desarrollaron 3 SubVI:

- Sub_ColorPatternMatching: Se encarga de hacer el reconocimiento de objetos mediante el “Color Pattern Matching”, éste entrega las “matches” o coincidencias de los objetos encontrados.
- Sub_SearchAndObtain: Se encarga de tomar las “matches” del proceso de “Color Pattern Matching” y procesarlas para mostrarlas en pantalla, esto es, realizar una interpolación de coordenadas de visión a las coordenadas del plano de la máquina y mostrándolas marcadas en la imagen de la cámara.
- Sub_Matches: Se encarga de tomar la coordenada de la coincidencia y hacerle un cuadro indicador para ser mostrado en pantalla.

5.3 Reconocimiento con Color Pattern Matching

El reconocimiento de imágenes en la interfaz se realiza mediante el uso del Color Pattern Matching (emparejamiento por patrón de color), un método de reconocimiento que combina dos tipos de

reconocimiento básicos: Color Location (locación de color) y Grayscale Pattern Matching (emparejamiento por patrón en escala de grises) [26].

Color Location básicamente, compara pixel a pixel el template de color con la imagen utilizando algoritmos especiales para acelerar el proceso, una muestra de este proceso se puede ver en la Figura 5-11.

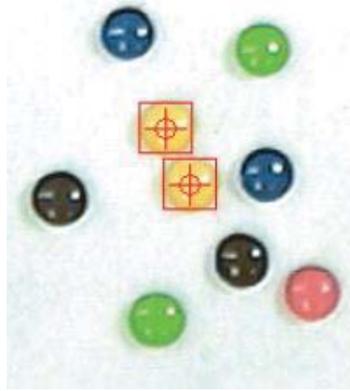


Figura 5-11: Detalle del Color Location con pastillas de colores

Grayscale Pattern Matching básicamente, convierte la imagen a escala de grises y luego, utilizando la información de los bordes del template a buscar, busca coincidencias en la imagen haciendo uso de algoritmos de búsqueda, una muestra de este proceso se puede ver en la Figura 5-12.

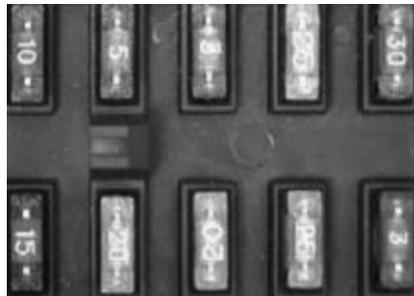


Figura 5-12: Detalle del Grayscale Pattern Matching con fusibles

Luego, se asigna un puntaje de acuerdo a la información obtenida de los dos procesos anteriores. Este puntaje se traduce en qué tanto se parece la supuesta coincidencia del template en la imagen. La interfaz permite ajustar el puntaje límite del reconocimiento, permitiendo bajar o subir la exigencia de la búsqueda.

Discusión y Conclusiones

En este informe se muestran los métodos y resultados obtenidos de la construcción de una planta de laboratorio que combina el uso de dos tecnologías muy sofisticadas, el Control Numérico Computarizado y la Visión Computacional. Según lo expuesto, se puede decir que los objetivos propuestos fueron cumplidos con el trabajo realizado y que el objetivo general fue alcanzado de manera correcta. La planta funciona tal como se esperaba, así como la interfaz y la documentación elaborada.

Las plantas con propósitos educativos en el laboratorio son de gran importancia, puesto que entregan un apoyo fundamental en el aprendizaje de los alumnos, esto al impulsarlos a reflexionar sobre las problemáticas asignadas en la experiencia, llevándolos a tomar decisiones a partir de la aplicación de sus conocimientos.

Por otra parte, el proceso de diseño es transversal a todo el proyecto debido a que constantemente se tienen que hacer arreglos que se ajusten a las necesidades de movimiento, calibración, entre otros. En el caso de este proyecto, se tuvieron que hacer muchos ajustes a las piezas 3D que se diseñaron para que la pieza se ajustara de forma óptima a las piezas metálicas, motores, electrónica, cables y más específicamente, al comportamiento de la máquina en movimiento.

Se puede decir con seguridad que el proceso de calibración es fundamental en la construcción y puesta en marcha de una máquina, en especial una con un grado de precisión superior, como la presentada en este proyecto. También es muy importante que estas calibraciones queden bien documentadas para futuros trabajos con la máquina, tanto en el escenario de agregar nuevas funcionalidades a la misma, como también para su uso cotidiano en experiencias de laboratorio. Se debe incentivar el mantenimiento preventivo de la máquina, con el fin de no incurrir en problemas tales como, tiempos de espera para reparaciones o gastos que podrían involucrar los mismos.

La impresión 3D entrega la facilidad de obtener piezas a medida, que ofrecen una buena calidad, así como un bajo coste. Si bien los tiempos de impresión pueden ser extensos, no se compara a lo que significaría emitir una orden de construcción de una pieza a una empresa o proveedor externo. En este sentido, otro de los aspectos que soluciona mucho el proceso de diseño son los softwares de diseño 3D, tales como SketchUp y 123D Design, que fueron utilizados en el proceso de diseño de este proyecto.

Es relevante mencionar la existencia de gran cantidad de software de diseño asistido por computadora (CAD), los cuales cuentan con grandes herramientas que permiten simplificar el proceso de diseño. Además, permiten, a través de sus funcionalidades, un fácil proceso entre la creación de la pieza 3D, exportación al formato STL y luego al proceso de impresión. Pero, estos softwares como Autodesk

Inventor, Fusion 360, entre otros, son pagados, es por ello que sería de gran apoyo al trabajo obtener una licencia de utilización para el desarrollo de futuros trabajos, permitiendo que el trabajo realizado sea cada vez más profesional y adaptado a las tecnologías usadas en grandes empresas del sector.

En cuanto al control del movimiento de la máquina, ha quedado demostrado que los códigos G hacen honor a su popularidad, puesto que son una herramienta muy robusta y al mismo tiempo, muy fácil de implementar. Esta facilidad para implementar es gracias al firmware utilizado, Marlin, que ha demostrado ser un firmware muy completo para controlar máquinas CNC y que aún se le debe mucha investigación para entender a cabalidad su funcionamiento.

La aplicación LabVIEW es sin duda una herramienta muy poderosa, que entrega de forma relativamente fácil, la función de diseñar interfaces funcionales con muchas opciones. Así mismo, el programa entrega muchos ejemplos para conocer cómo funciona su amplio inventario de herramientas. El lenguaje gráfico que LabVIEW utiliza es muy intuitivo, y si bien es fácil de comprender y desarrollar, hace falta mucha práctica para poder optimizar los códigos en funciones más empaquetadas y compactas que hagan del diagrama de programación mucho más fácil de comprender. El proceso de convertir los bloques Express a bloques de bajo nivel hizo posible comprender de mejor manera tanto la conexión con la webcam y la consiguiente adquisición de imágenes, como del reconocimiento visual utilizado en el Color Pattern Matching. Adicionalmente a esto, el traspaso de este tipo de bloques hizo posible una mayor libertad en la configuración de los procesos.

Se logró el objetivo de crear una interfaz gráfica que es capaz de realizar las tareas designadas. En el aspecto de diseño es amigable y fácil de utilizar, permitiendo que el usuario ajuste la acción a realizar a medida que también protege a la máquina de posibles movimientos erróneos. Adicionalmente, se realizó la programación de ésta teniendo siempre en mente la escalabilidad de la misma, permitiendo, sin dificultad, la capacidad de incorporar nuevas funcionalidades para la máquina.

La creación de un “Manual de Usuario” es un documento indispensable para que la máquina se utilice de manera correcta y no se cometan errores, ni mucho menos accidentes. Este documento debe ser realizado de forma resumida y simple, enfocado a que el usuario entienda de forma fácil el uso de la máquina.

Teniendo en cuenta las características de la máquina, las posibilidades de aplicaciones son infinitas, solo viéndose limitada por los límites de capacidad de la estructura, del manejo de los diversos softwares de diseño asistido por computadora y del ingenio en la programación de la aplicación de control. Esto también se debe, en parte, gracias al arduo estudio realizado en cada uno de los componentes utilizados, en donde se sobreestimó el tamaño de los motores, electrónica, fuente de alimentación, cableado y controladores, con el fin de que cuando exista un proceso de ampliación de las capacidades esta solo signifiquen un cambio de pequeños componentes y programación, y no en un nuevo proceso de compra de materiales, elección de electrónica y programaciones del sistema desde cero.

La implementación de las herramientas intercambiables con alimentación de 5[V] y 12[V] controlables desde la interfaz, son un paso crucial para la escalabilidad del proyecto. Esto permitirá, sumado con la adecuada documentación de la programación en LabVIEW, que en el futuro se puedan implementar nuevas acciones a la planta.

El proceso de construcción de esta máquina ha sido desarrollado con la idea constante de que en el futuro se puedan desarrollar más aplicaciones que las actuales y que sus capacidades crezcan gracias al trabajo de otro estudiante y/o profesor. Es por esto que sus características de armado, piezas 3D y la aplicación de LabVIEW han sido documentadas de buena forma para lograr una escalabilidad fácil de implementar.

Es muy importante que la maquina pueda crecer, en cuanto a sus funcionalidades y características, debido a las grandes capacidades que tiene y las diversas posibilidades de trabajo que pueden ofrecer. Durante el desarrollo de este informe se ha documentado detalladamente los pasos para que la máquina funcione de manera correcta, y también, se ha dejado en evidencia la potencialidad de la misma. Esto es esencial para determinar los nuevos requisitos de diseño y programación que puedan ser implementados en una futura versión del proyecto.

Bibliografía

- [1] E. L. A. R. L. A. B. J. P. G. A. N. Riveros, Prácticas integrales en planta piloto - Experiencia innovadora, Salta, Argentina, 2013.
- [2] A. Leva, A hands-on experimental laboratory for undergraduate courses in automatic control, Milano, Italy: IEEE, 2003.
- [3] J. J. Peters, «Introducción al control numérico computarizado (CNC),» Diciembre 2013. [En línea]. Available: [http://wiki.ead.pucv.cl/Introducci%C3%B3n_al_control_num%C3%A9rico_computarizado_\(CNC\)](http://wiki.ead.pucv.cl/Introducci%C3%B3n_al_control_num%C3%A9rico_computarizado_(CNC)).
- [4] M. G. K. A. M. Johansson, Interactive tools for education in automatic control, Suecia: IEEE, 1998.
- [5] I. S. Ching, «Herramientas para la Docencia en Automática orientadas hacia la metodología ECTS,» 2012.
- [6] D. Hodson, «Hacia un enfoque más crítico del trabajo de laboratorio,» The Ontario Institute for Studies in Education, Toronto, Canada, 1994.
- [7] J. A. O. Christian G. Quintero, Control Automático Aplicado, Barranquilla, Colombia: Universidad del Norte, 2011.
- [8] De Maquinas y Herramientas, «Introducción a la Tecnología CNC,» 28 Diciembre 2015. [En línea]. Available: <http://www.demaquinasyherramientas.com/mecanizado/introduccion-a-la-tecnologia-cnc>.
- [9] D. Marr, La Visión, San Francisco, 1982.
- [10] L. E. Sucar, Visión Computacional, Puebla, México, 2008.
- [11] J. R. Rábago, «Generación de fuentes virtuales de sonido en audífonos,» Universidad de las Américas, Puebla, México, 2005.

-
- [12] Yamaha, «Robot vision iVY2 (RCX340),» [En línea]. Available: <https://global.yamaha-motor.com/business/robot/lineup/ivy/ivy2/index.html>.
- [13] FANUC, «M-1iA/0.5A,» [En línea]. Available: <http://www.fanuc.eu/pl/en/robots/robot-filter-page/m1-series/m-1ia-05a>.
- [14] Farmbot, «Farmbot,» [En línea]. Available: <http://www.farmbot.io>.
- [15] Logitech, «C270 HD Product Overview,» [En línea]. Available: <http://www.logitech.com/es-es/product/hd-webcam-c270>.
- [16] Reston Condit, Microchip Technology Inc., «Stepping Motor Fundamentals,» Iowa.
- [17] MotionKing Motor Industry Co. Ltd., *17HS4401 Datasheet*.
- [18] National Electrical Manufacturers Association, «NEMA Standards Publication ICS 16,» Virginia, 2001.
- [19] RepRap Community, «RepRap,» [En línea]. Available: <http://reprap.org/>.
- [20] Allegro, *A4988 Datasheet*.
- [21] Texas Instruments, *DRV8825 Datasheet*.
- [22] J. Thornton, «LinuxCNC G code Tutorial,» 2012. [En línea]. Available: <http://gnipsel.com/linuxcnc/g-code/gen01.html>.
- [23] S. T. N. X. W. Xu, «Making CNC machine tools more open, interoperable and intelligent - a review of technologies,» Auckland, New Zealand, 2005.
- [24] Q. H. X.W Xu, «Striving for a total integration of CAD, CAPP, CAM and CNC,» Auckland, New Zealand, 2003.
- [25] Embedded Laboratory, «Embedded Laboratory,» 13 08 2016. [En línea]. Available: <http://embeddedlaboratory.blogspot.cl/2016/08/serial-communication-using-labview.html>.
- [26] National Instruments, «Color Pattern Matching Concepts Help,» 2013.

A Manual de usuario

El presente Manual de Usuario tiene como finalidad, mediante la explicación paso a paso de los procedimientos a realizar, permitir que el usuario pueda utilizar de forma correcta tanto la máquina **CNC+Vision Machine** como de su interfaz de LabVIEW.

A.1 Indicaciones de Seguridad

En esta sección se destacan algunas indicaciones de seguridad para no arriesgarse a sufrir accidentes o estropear la máquina.

- ✓ Dado que la máquina trabaja con 12[V] y un máximo de 5[A], se debe tener extrema precaución de la electrónica cuando está energizada.
- ✓ Cualquier reversión de polaridad o cortocircuito puede destruir los componentes e incluso causar un incendio.
- ✓ Es importante que los ventiladores se encuentren funcionando constantemente y no se encuentren obstruidos, pues la temperatura, especialmente en los controladores de los motores, puede ser muy alta.
- ✓ Mientras se manipula la electrónica se debe desconectar tanto la fuente de 12[V] como el cable USB de 5[V].
- ✓ Mover los motores de forma manual no es recomendado.

A.2 Interfaz

La aplicación reúne cada función de la máquina en una sola interfaz, requiriendo por parte del usuario que prepare la estructura de forma adecuada para realizar la acción. En todo momento la aplicación muestra tres módulos básicos: la configuración inicial de la tarjeta, su consola de códigos G y la visión de la cámara web.

A.2.1 Información General

La interfaz (Figura A-1) cuenta con 3 secciones en común: una sección de configuración, de visión de Webcam y del registro de la consola. Y adicionalmente un espacio donde se visualizan las opciones de la acción a realizar.

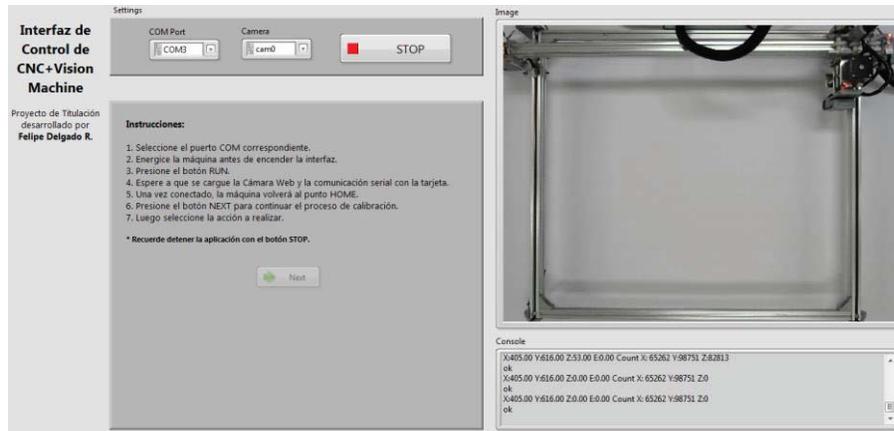


Figura A-1: Panel frontal de la interfaz

La sección de configuración (Figura A-2) entrega la configuración del puerto COM de la máquina, la cámara a utilizar para la adquisición de imágenes y un botón de STOP que detiene toda acción de la aplicación.

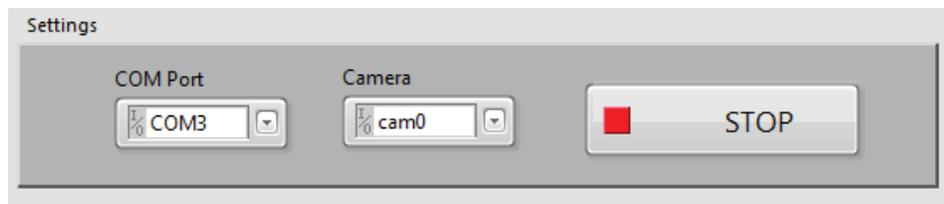


Figura A-2: Sección de Configuración

La sección de visión de Webcam (Figura A-3) muestra la visión de la base de la máquina mediante la Webcam, está tiene una configuración de 640x480 pixeles y toma muestras cada 50[ms].



Figura A-3: Visión de la Webcam

La sección del registro de la consola (Figura A-4), muestra la respuesta de la máquina ante los códigos G enviados.

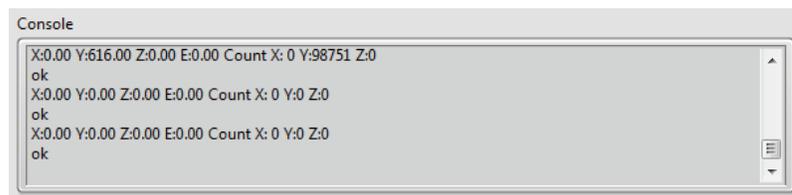


Figura A-4: Registro de consola

Al iniciar la aplicación se muestra una pantalla de inicio (Figura A-5) donde aparecen brevemente explicadas las instrucciones e información de seguridad. En este punto, la posición de la máquina es llevada al punto inicial, al hacer esto se permite utilizar el botón "Next", el cual dirige a la sección de calibración.

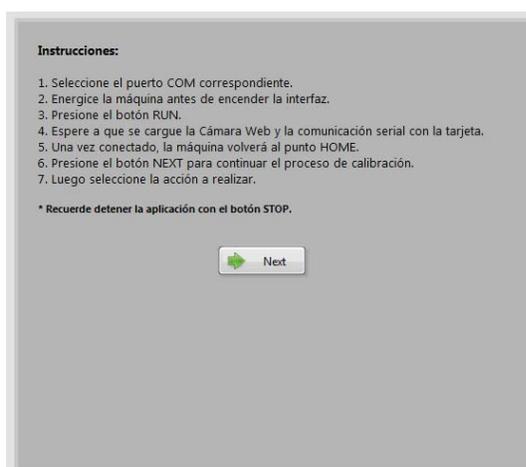


Figura A-5: Pantalla de Instrucciones

A.2.2 Calibración



Figura A-6: Pantalla de Calibración

La sección de calibración (Figura A-6) es una antesala que permite que el usuario tome todas las medidas de seguridad correspondiente para no llevar a la máquina a límites superiores de lo establecido, llevando a posibles errores o incluso el deterioro de la misma.

Esta sección permite elegir la herramienta a utilizar, esto para configurar los márgenes de movimiento, y también la acción que se realizará durante la sesión. Al elegir la herramienta adecuada, los límites se envían automáticamente mediante códigos G a la máquina, permitiendo operar la máquina en la sección correspondiente.

A.2.3 Movimientos Básicos

La sección de movimientos básicos (Figura A-7) permite realizar movimientos simples con la máquina, como líneas rectas, círculos o el envío de códigos G de forma manual.

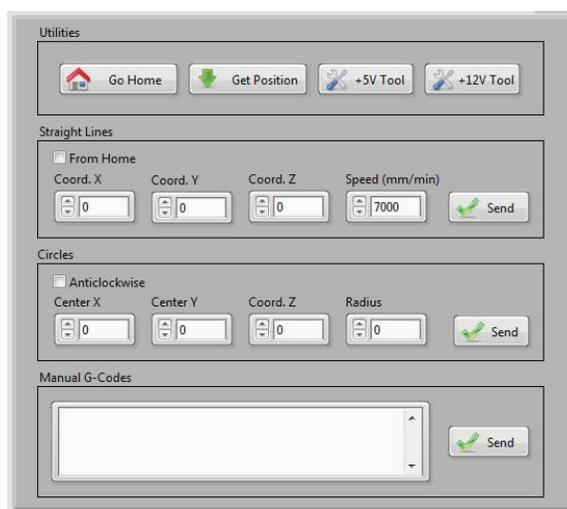


Figura A-7: Pantalla de Movimientos Básicos

Adicionalmente tiene opciones prefabricadas como volver al punto Home, entregar la posición actual de la herramienta y encender/apagar las herramientas de +5[V] y +12[V].

Su objetivo es la familiarización de códigos G por parte del usuario, y es una buena forma de comprender los alcances de movimiento de la estructura.

En la Figura A-8 se puede ver un diagrama de funcionamiento de la sección Movimientos Básicos.



Figura A-8: Diagrama de funcionamiento de Movimientos Básicos

A.2.4 Búsqueda y Obtención

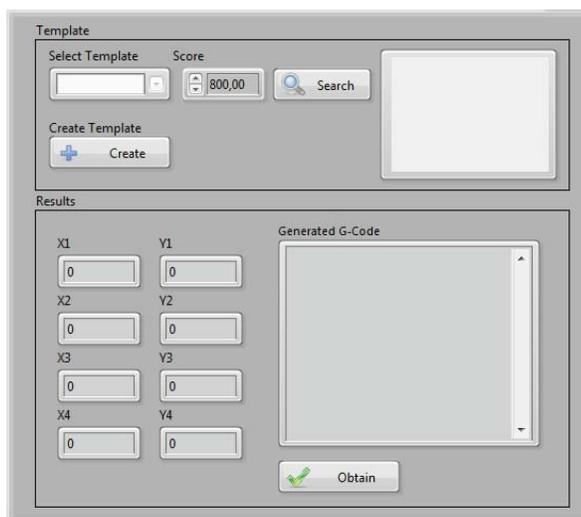


Figura A-9: Pantalla de Búsqueda y Obtención

En la sección de búsqueda y obtención (Figura A-9), el usuario debe elegir cuál es la pieza elegida para obtener y realizar una búsqueda de la misma en la base. Si la cámara identifica de forma correcta la pieza elegida, se bloquearán las coordenadas en la pantalla con un cuadro color celeste claro. Además, las coordenadas de los objetos encontrados se mostrarán en la parte de resultados. Junto con un código G que se muestra al usuario. Este código G será enviado a la máquina una vez se presione el botón de "Obtain", esto resultará en el movimiento de la máquina para recolectar las piezas pedidas por el usuario.

La secuencia a seguir es, dirigirse a la coordenada de la pieza, acercar el electroimán mediante la bajada del eje Z al punto específico (menos de 1[cm] de distancia de la base para que el electroimán funcione de forma correcta), encender el electroimán, subir al punto 0 en Z, volver al punto Home, soltar la pieza, y repetir hasta para 4 piezas encontradas.

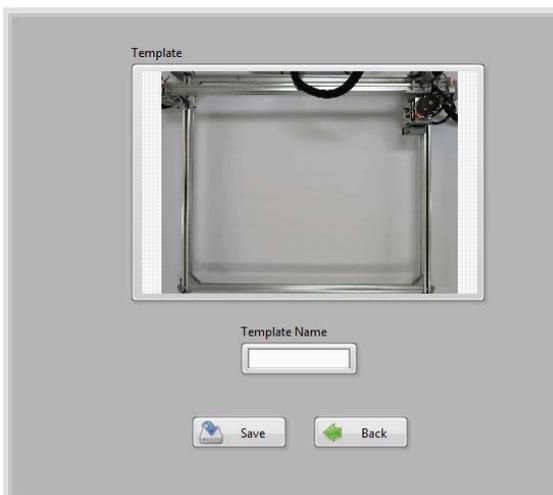


Figura A-10: Pantalla de crear plantilla para Búsqueda y Obtención

Adicionalmente, esta sección permite crear nuevas piezas, almacenando el archivo de imagen en la carpeta “src” y también ingresando el nombre de la pieza con el nombre de la imagen en una base de datos en el archivo “templates_search.txt”. La creación de las piezas resulta en la actualización automática de las piezas disponibles para búsqueda en la sección de Búsqueda y Obtención. La pantalla de creación de plantillas se puede ver en la Figura A-10.

El diagrama de funcionamiento de la sección Búsqueda y Obtención se puede ver en la Figura A-11.

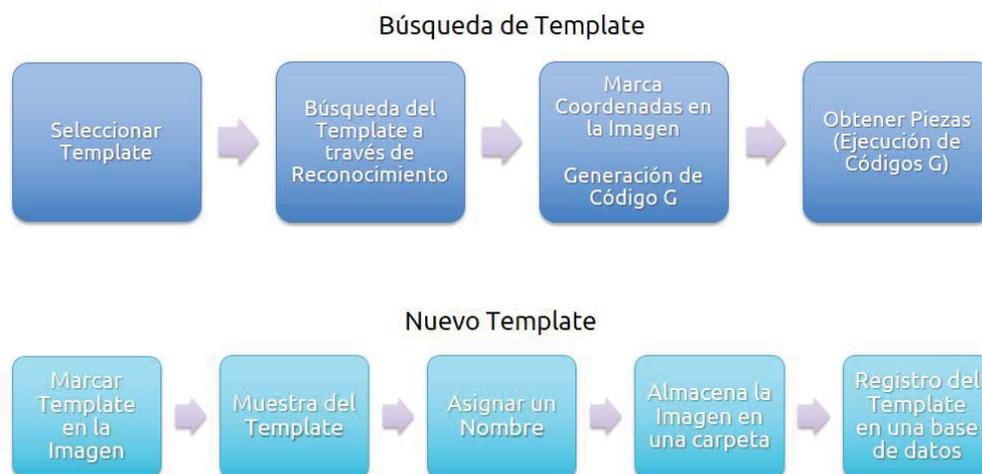


Figura A-11: Diagrama de funcionamiento de Búsqueda y Obtención

A.2.5 Huerto Autónomo

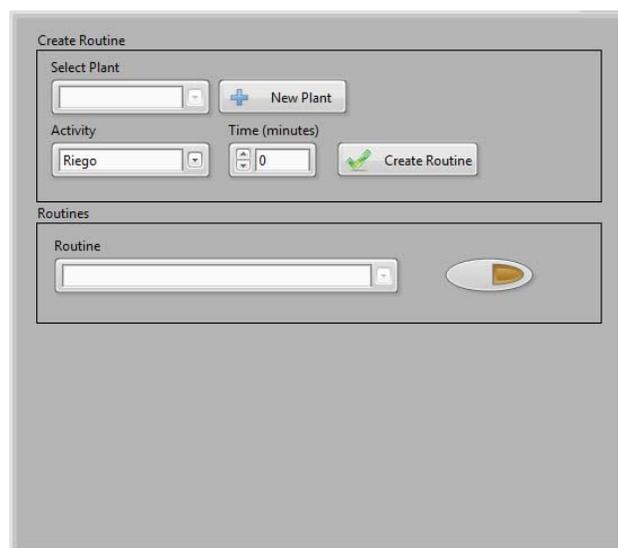


Figura A-12: Pantalla de Huerto Autónomo

La sección Huerto Autónomo (Figura A-12) permite crear rutinas prefabricadas para, en el futuro, poder controlar de forma automática una o varias plantas. Para hacerlo se debe crear una rutina nueva, seleccionando la plantilla de la planta que se desea supervisar, una actividad (que tiene un código G

preestablecido) y el tiempo de repetición de la rutina a realizar. Al crear una rutina, esta se almacena en el archivo “autonomous_farm.txt” en la carpeta “src”. Cuando se crea una rutina esta se vuelve disponible en la lista de Rutinas.

Al momento de querer iniciar una rutina, esta se debe escoger de la lista y presionar el botón de inicio de rutina. Inmediatamente la cámara identificará la planta correspondiente y generará un código G que se iniciará con la frecuencia de tiempo almacenada.

Adicionalmente esta sección permite la creación de plantillas para plantas, las cuales se guardan en el archivo “plants_search.txt”. La pantalla de creación de plantillas se puede ver en la Figura A-13.



Figura A-13: Pantalla de creación de plantillas para Huerto Autónomo

En la Figura A-14 se puede ver el diagrama de funcionamiento de la sección Huerto Autónomo.



Figura A-14: Diagrama de funcionamiento de Huerto Autónomo

A.3 Movimiento

El movimiento de la máquina está dado por la tarjeta RAMPS mediante la traducción de los códigos G enviados, estos códigos G son traducidos por Marlin, firmware que está contenido en el Arduino Mega. Los códigos G recibidos por la máquina se traducen en movimiento de los 3 ejes de la máquina.

Cabe destacar que la máquina se mueve con motores paso a paso, estos motores tienen mucha fuerza de movimiento, si la máquina se manipula de forma incorrecta, estos motores pueden chocar con la estructura, o lo que haya a su paso, y no se detendrán fácilmente. Eventualmente estos movimientos forzados deterioran de forma considerable la vida útil del motor.

El eje X se compone de dos motores conectados en paralelo, lo que permite que se muevan los dos como si fueran uno solo. Este eje necesita de la fuerza de dos motores para impulsarse debido a que es el que soporta más peso (eje Y, eje Z y la herramienta activa).

El eje Y se compone de un motor que mueve el carro que contiene el eje Z.

El eje Z se compone de un motor, que, mediante un husillo o hilo sin fin, mueve un carro que es el que soporta las herramientas intercambiables de la máquina. Este eje tiene un límite de 52[mm] de forma vertical, cabe destacar que esta distancia puede verse limitada aún más mediante software, dependiendo de la herramienta que se esté utilizando.

Al iniciarse la interfaz de LabVIEW envía a los 3 ejes al punto HOME, esto se realiza debido a que cuando la máquina se energiza, inmediatamente piensa que está en el punto de origen, por lo que cualquier movimiento será basado en esa lógica. Debido a esto, es importante tomar especial atención a dos cosas: que al iniciar la interfaz la máquina se encuentre energizada por +12[V] y que los sensores “Final de Carrera” estén bien conectados.

A.4 Códigos G

En esta sección se comentan los códigos G disponibles para utilizar, su función y sus características.

Los códigos G se componen de una letra y luego de números, estos pueden ser enteros o racionales. A pesar de llamarse códigos G, cualquier letra del alfabeto es un tipo de comando en código G. El formato de alguno de ellos es explicado en la Tabla A-1.

Tabla A-1: Tipos de códigos G

Letra	Significado
Gn	Comando Gcode estándar, indica cómo moverse hasta un punto
Mn	Comando definido para Impresoras 3D
Sn	Parámetro de comando, como la tensión enviada a un motor
Pn	Parámetro de comando, como el tiempo en milisegundos
Xn, Yn, Zn	Una coordenada en el eje X, Y o Z
In	Parámetro
Jn	Parámetro
Fn	Feedrate o velocidad de movimiento en mm por minuto

Una vez enviado los códigos necesarios, la máquina responde de diversas formas, las respuestas más comunes se pueden ver en la Tabla A-2.

Tabla A-2: Respuestas típicas en Marlin

Respuesta	Explicación
ok	El código se recibió de manera correcta
wait	En espera
busy: processing	El comando está siendo ejecutado
resend	Se solicita volver a enviar el código

A.4.1 Códigos G más utilizados

En la Tabla A-3 se muestran los códigos G más utilizados junto a un ejemplo y una explicación de este.

Tabla A-3: Códigos G más utilizados

Código	Significado	Ejemplo	Explicación
G0 – G1	Mover en línea recta	G1 X5.0 Y10.0 Z20.0 F3000	Moverse en línea recta desde el punto actual a (5,10,20) a 3000[mm/min]
G2 – G3	Movimiento en círculo (sentido reloj y anti reloj)	G2 X20.0 Y10.0 I0 J-5	Moverse en un círculo con sentido reloj con punto centro (20,15) con radio 5.
G4	Mantener	G4 P200	Detenerse durante 200 milisegundos
G28	Volver a Origen	G28	Hace que todos los ejes vuelvan al punto de origen (0,0,0)
G90	Fijar posicionamiento absoluto	G90	Todas las coordenadas desde ahora se basan desde el punto de origen
G91	Fijar posicionamiento relativo	G91	Todas las coordenadas desde ahora se basan desde el punto actual
G92	Fijar posición	G92 X10.0	Desde ahora, el punto de origen del eje X se encuentra en 10.0
M114	Obtener posición actual	M114	La máquina entrega los valores actuales de posición de los ejes
M119	Obtener estado de los finales de carrera	M119	Entrega los estados actuales de los sensores “Final de Carrera” conectados a la placa

A.4.2 Lógica de Movimiento

Una de las principales medidas que se debe tomar en una máquina CNC es la de llevar al punto Home (0, 0, 0) a la máquina al momento de iniciarla. Esto debido a que, al encenderse, la máquina “piensa” que se encuentra en el punto Home, llevando a posibles problemas de movimiento (por ej. si un eje tiene un límite de 600[mm] y al encenderse la máquina se encuentra en 300[mm], se le puede enviar el comando de moverse 600[mm] y la máquina intentará avanzar a 900[mm]).

Los códigos G ofrecen dos formas de posicionamiento, el posicionamiento absoluto (*G90*) o posicionamiento relativo (*G91*). El primero se basa en que todos los movimientos realizados se hagan

desde el punto inicial. El relativo se basa en que todos los movimientos realizados se hagan desde el punto actual. Por ejemplo:

```
G90
G1 X50
G1 X75
```

Este código terminaría en el punto X: 75

```
G91
G1 X50
G1 X75
```

Este código terminaría en el punto X: 125

Los códigos para líneas rectas G0 y G1. G0 representa a todos los movimientos en línea recta que no son parte del uso de la herramienta y G1 cuando se está utilizando la herramienta (ej. un grabador laser, cuando se mueva con el láser apagado usaría un G0 y con el láser encendido con G1). En la práctica, muchos softwares solamente utilizan G1 porque “no hace mucha diferencia” uno u otro. El formato de estos es:

G1 X[x] Y[y] Z[z] F[f]

Siendo (x, y, z) el movimiento en cada eje en mm y f la velocidad en [mm/min] del movimiento.

Los códigos para circunferencias G2 y G3 (en sentido reloj y sentido antireloj respectivamente) tienen unas consideraciones especiales. El formato de estos es:

G2 X[x] Y[y+r] I0 J[-r]

Esto para realizar una circunferencia con centro en (x, y) de radio r .

B Guías de Ejercicios

A continuación, se presentan 5 guías de ejercicios prácticos para realizar con la planta.

B.1 Primeros Pasos

En orden de aprender a ejecutar la interfaz y de interiorizarse con el uso de la máquina es que es necesario realizar esta práctica y desarrollar las actividades propuestas.

En esta actividad se detallarán los primeros pasos a seguir para ejecutar la interfaz y poner en movimiento la máquina, junto a un breve repaso de sus aplicaciones.

! NOTA: Antes de realizar esta actividad es obligatoria la lectura del Manual de Usuario de la máquina.

Inicialización

1. Iniciar la aplicación de LabVIEW "CNC+Vision Machine.vi".
2. Conectar el cable USB de la cámara y de la máquina al computador.
3. En "Settings" seleccionar el puerto adecuado de la máquina en "COM Port" y la Webcam en "Camera".
4. Encienda la máquina con el interruptor ubicado en la parte derecha de la caja.
5. Presione el botón RUN en la parte superior izquierda de la ventana.
6. Espere a que la Webcam cargue y que la máquina llegue al punto home, al alcanzar este punto se activará el botón "Next".
7. Al presionar el botón "Next" la interfaz se ubicará en la sección de Calibración.

Movimientos Básicos

1. Conecte el Módulo LED en el soporte de Herramientas y seleccione el Módulo LED en "Tool".
2. A continuación, presione el botón "Basic Movements".
3. En esta sección, presione el botón "+5V Tool" y asegúrese de que el indicador LED se encienda. Si lo presiona una vez más, el indicador LED se apagará.
4. En "Straight Lines" escriba 3 valores distintos para los ejes X, Y y Z (límites máximos: 504, 616 y 52 respectivamente). Presione el botón "Send". La máquina se moverá a la ubicación solicitada.

5. Presione el botón "Get Position" y podrá leer en "Console" la ubicación de los 3 ejes, esta debería ser la misma que la ingresada anteriormente.
6. Presione el botón "Go Home" y la máquina se dirigirá al punto inicial.
7. En "Circles", escriba valores para el centro en X, Y, una coordenada para bajar en Z y el radio del círculo, presione "Send". La máquina hará una circunferencia en las coordenadas pedidas.
8. En la sección "Manual G-Codes" ingrese comandos obtenidos del Manual de Usuario (apartado A.4.1).
9. Una vez termine la sesión, presione el botón "STOP" y apague la máquina.

B.2 Figuras planificadas

En esta práctica, se realizarán, utilizando el Módulo Escritura, figuras en la plataforma de trabajo.

Es recomendable poner un papel en la base de la plataforma para realizar distintos dibujos y fijarlo bien a la plataforma para que no se mueva durante el dibujo.

- ! NOTA: Es necesario realizar siempre la Inicialización descrita en la actividad B.1 antes de comenzar.**

Calibración del Lápiz

1. En la pantalla Calibración, seleccione el Módulo Escritura (este Módulo limita el movimiento del eje Z a 48[mm]).
2. Antes de conectar el Módulo Escritura en el soporte de herramientas, coloque un lápiz apuntando hacia abajo, asegúrese de que quede firme al soporte.
3. Conecte el Módulo Escritura.
4. En la sección "Basic Movements", utilice "Straight Lines" para mandar movimientos de 1[mm] en el eje Z. Esto hará que el lápiz baje lentamente, deténgase cuando el lápiz toque el papel.
5. Presione el botón "Get Position" y anote la posición del eje Z, esta posición se utilizará a continuación en los trazos a realizar.
6. Vuelva al punto inicial.

Dibujar Figuras

1. Realice 3 circunferencias concéntricas con radios crecientes de forma constante (ej. radio 20[mm], 40[mm] y [60[mm]) cuyo centro sea el punto centro del área de trabajo.
2. Realice sobre las circunferencias, una cruz que se encuentre dentro del área de la circunferencia más grande.
3. Realice 4 cuadros, uno en cada punto donde la cruz toca a la circunferencia, tocando un extremo de estos, la circunferencia más grande y el otro extremo, la circunferencia más pequeña.
4. El resultado final debería ser como se ve en la Figura B-1: Figura Final.
5. Utilizando los códigos mostrados en el Manual de Usuario, apartado A.4.1, escriba los códigos, primero en papel y luego en "Manual G-Codes".
6. Antes de enviar los códigos, es recomendable "ensayar" el movimiento antes, esto, sin bajar el lápiz hasta la distancia obtenida en la etapa de Calibración.
7. Recuerde que, entre trazos dibujados, debe subir el lápiz unos cuantos mm.

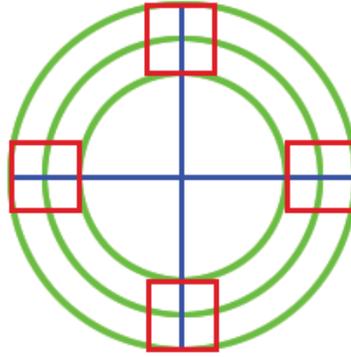


Figura B-1: Figura Final

B.3 Reconocimiento de Objetos

En esta actividad, se interiorizará en el reconocimiento de objetos.

! NOTA: Es necesario realizar siempre la Inicialización descrita en la actividad B.1 antes de comenzar.

Aprendizaje del reconocimiento

1. Coloque el Módulo LED y selecciónelo en la pantalla de Calibración.
2. Ingrese a la sección "Search and Obtain".
3. Presione el botón "Create Template".
4. Posicione una pieza en la plataforma.
5. Utilizando el mouse, seleccione en "Image", la pieza en la plataforma.
6. En "Template" se verá la pieza seleccionada, agréguele un nombre en "Template Name" y presione "Save".
7. Presione "Back" para volver a la sección "Search and Obtain".

Ejecutar el reconocimiento

1. En "Select Template" escoja el nombre ingresado.
2. La pieza se marcará con un cuadro rojo en la imagen.
3. Dependiendo de si el reconocimiento se logra o no, puede ir modificando el "Score". Entre más alto sea el puntaje, más exigente será el reconocimiento. Será necesario subirlo en caso de que se logre un "falso reconocimiento" (o sea, que se reconozca la pieza en lugares donde no está) y bajarlo en caso de que no logre reconocerlo.
4. Una vez que el reconocimiento se mantenga estable, presione el botón "Search", se marcará la figura con un cuadro celeste.
5. En "Results" se mostrarán las coordenadas donde se encuentra la figura y se generará un código G.
6. Comente este código G generado para entender lo que hace cada línea.
7. Presione "Obtain" y verifique que el movimiento ejecutado corresponda a las instrucciones dadas.

8. Repita estos pasos para varias figuras, anote para cada una, el puntaje adecuado para que el reconocimiento sea óptimo.

B.4 Búsqueda y Obtención

En esta actividad se pondrá en práctica el reconocimiento utilizando la sección “Search and Obtain” para obtener piezas de metal con el electroimán.

- ! NOTA: Es necesario realizar siempre la Inicialización descrita en la actividad B.1 antes de comenzar.**

Búsqueda y Obtención

1. Conecte el Módulo Electroimán y seleccione este módulo en la pantalla de Calibración.
2. Posicione las piezas de metal en la plataforma de forma aleatoria.
3. Seleccione en “Select Template” la figura a obtener.
4. Ajuste el “Score” para que exista un reconocimiento óptimo y presione “Search” hasta obtener todas las coincidencias de las figuras.
5. Presione “Obtain” y verifique que la obtención se logre de forma correcta.
6. Si la obtención se logró bien, todas las piezas requeridas serán llevadas al punto inicial.
7. Repita estos pasos para varias piezas de metal, agregando, si se desea, más piezas utilizando la sección de “Create Template”.

B.5 Modificaciones a Búsqueda y Obtención

En esta actividad se le harán algunas modificaciones a la acción de Búsqueda y Obtención utilizando el conocimiento aprendido en las prácticas anteriores.

El objetivo es realizar el movimiento de búsqueda y obtención, pero llevando a las piezas a otro punto de entrega en vez de al punto inicial.

Este punto de entrega será diseñado por el mismo alumno.

- ! NOTA: Es necesario realizar siempre la Inicialización descrita en la actividad B.1 antes de comenzar.**

Crear un Punto de Entrega

1. Utilizando el Módulo Escritura, realice los pasos necesarios para calibrar el lápiz del módulo.
2. En “Basic Movements”, e ingresando códigos G de forma manual, realice un dibujo en una hoja de papel pequeña (10[cm]x10[cm]), asegurándose de fijarla bien a la plataforma para que no se mueva durante el dibujo. Este dibujo será el punto de entrega de las piezas.
3. En “Search and Obtain”, cree un nuevo Template del dibujo realizado.
4. Mueva el papel a un punto aleatorio en la plataforma y haga la búsqueda. Anote las coordenadas obtenidas.

Búsqueda modificada

1. Utilice el Módulo Electroimán y selecciónelo en la pantalla de Calibración.
2. Posicione las piezas de metal en la plataforma y haga la búsqueda.
3. Copie el código G generado y péguelo en un editor de texto.
4. Modifique el código G para que el punto de entrega cambie a las coordenadas donde se encuentra el papel.
5. Vaya a la opción “Basic Movements” y copie el código en “Manual G-Codes”.
6. Ejecute el movimiento y asegúrese que la obtención fue realizada de forma correcta.