

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO  
CENTRO UNIVERSITARIO UAEM TEXCOCO



**TESIS**

MOVIMIENTO AUTOMATIZADO EN TRES DIMENSIONES PARA  
APLICACIÓN DE TRATAMIENTOS EN CULTIVOS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

HERNÁNDEZ RIVAS FRANCISCO JAVIER

DIRECTOR:

DR. EN C. LUGO ESPINOSA OZIEL

# Contenido

Índice Figuras .....	iv
Índice Tablas.....	vi
Resumen.....	1
Introducción .....	2
Planteamiento del Problema .....	3
Objetivos .....	4
Objetivo General .....	4
Objetivos Específicos.....	4
Justificación .....	5
Metodología.....	6
Antecedentes .....	8
Capitulo I.    Marco Teórico. ....	10
Automatización .....	11
Funciones Trigonométrica.....	11
Voltaje .....	11
Corriente .....	12
Motores pasó a paso.....	12
Servomotor .....	13
Sensor.....	14
Fuentes Reguladas Conmutadas .....	15
Capitulo II.    Arduino.....	17
Concepto Arduino. ....	19
Hardware Arduino.....	20
Entradas/Salidas de Arduino .....	24
Shield de Arduino .....	24
El Software de Arduino IDE (Integrated Development Environment) .....	29
Instalación del software de Arduino en una computadora .....	30
Entorno de desarrollo de Arduino.....	31
Lenguaje de Programación de Arduino.....	32
Prueba del Circuito con Arduino .....	36
Capitulo III.    Sensores. ....	39
Concepto de Sensores.....	40

Clasificación de Sensores .....	40
Sensores analógicos y digitales .....	41
Sensores de Distancia.....	42
HC-SR04 Sensor Ultrasónico.....	43
Sensor de Temperatura.....	44
Sensor LM35.....	44
Sensor Final de Carrera .....	45
Capitulo IV. Prototipo. ....	47
Fase de Investigación preliminar.....	48
Pruebas de Factibilidad .....	48
Costo de desarrollo .....	49
Fase de definición de los requerimientos del sistema .....	49
Fase de diseño técnico .....	50
Movimiento .....	50
Conexiones Eléctricas.....	60
Rutinas de Software .....	65
Fase Pruebas. ....	74
Pruebas de movimiento .....	74
Conclusiones .....	76
Bibliografía .....	77

# Índice Figuras

Figura 1 Motor pasó a paso.....	13
Figura 2 Servomotor.....	14
Figura 3 Tipos de sensores .....	15
Figura 4 Fuente conmutada .....	16
Figura 5 Prototipo del primer Arduino.....	21
Figura 6 Componentes de Arduino Mega .....	23
Figura 7 Dual Step Motor Driver Shield.....	25
Figura 8 RAMPS 1.4 .....	26
Figura 9 Driver Pololu A4988.....	26
Figura 10 Sensor Shield .....	27
Figura 11 1Sheeld.....	28
Figura 12 Modulo L298N.....	29
Figura 13 Entorno Arduino.....	31
Figura 14 Barra de Herramientas .....	32
Figura 15 Puerto COM.....	36
Figura 16 Señal Digital.....	42
Figura 17 Señal Analógica .....	42
Figura 18 HC-SR04 sensor ultrasónico .....	43
Figura 19 LM35.....	45
Figura 20 Final de Carrera .....	46
Figura 21 Primer Prototipo.....	50
Figura 22 Movimiento Horizontal .....	51
Figura 23 Movimiento Vertical.....	51
Figura 24 Polea y Banda Dentada .....	52
Figura 25 Makeblock XY-Plotter Robot (Makeblock, 2015) .....	53
Figura 26 Robot que dibuja (Makeblock, 2015) .....	53
Figura 27 V-Slot .....	53
Figura 28 Ejemplo 1 V-Slot (OPENBUILS, 2015) .....	54
Figura 29 Ejemplo 2 V-Slot (OPENBUILS, 2015) .....	54
Figura 30 Primer Prototipo con movimiento X Y .....	54
Figura 31 Segundo Prototipo con movimiento X Y .....	55
Figura 32 Diseño básico de bandas.....	55
Figura 33 Adaptación polea (OPENBUILS, 2015).....	56
Figura 34 Primer prototipo 2 motores integrados.....	57
Figura 35 Propuesta para eje X .....	57
Figura 36 Prototipo Mecánico final.....	58
Figura 37 Brazo robótico .....	59
Figura 38 Modelo matemático.....	59
Figura 39 Lectura de distancia .....	60
Figura 40 Conexión Motor paso a paso.....	61
Figura 41 Conexión final de carrera .....	61

Figura 42 Conexión LM35.....	62
Figura 43 Conexión Servomotor.....	63
Figura 44 Conexión Ultrasónico .....	64
Figura 45 Conexión del Prototipo Completo.....	65
Figura 46 Diagrama de flujo general .....	66
Figura 47 Función temperatura.....	67
Figura 48 Función Origen .....	68
Figura 49 Función Calibrar .....	69
Figura 50 Función servo .....	71
Figura 51 Función movY .....	72
Figura 52 Función movX.....	73

## Índice Tablas

Tabla 1 Versiones de Arduino (Arduino, 2015) .....	22
Tabla 2 Características técnicas Arduino Mega (Arduino, 2015) .....	23
Tabla 3 Tipos de datos en Arduino .....	35
Tabla 4 Costo de desarrollo.....	49
Tabla 5 Pruebas de movimiento.....	74
Tabla 6 Pruebas Sistema de detección de objetos.....	75
Tabla 7 Pruebas en Rutinas .....	75

## Resumen

Los sistemas apoyan todas las actividades en las organizaciones, y últimamente se han insertado más en el área agrícola, para la eficiencia de los procesos y aumentar la calidad de los productos, por lo que se presenta el desarrollo de un prototipo para automatizar procesos de monitoreo y aplicación de sustancias en campos experimentales para obtener resultados con mayor exactitud, al tener un control preciso al momento del monitoreo o inyección de cualquier tipo de sustancias necesarias en diversos proyectos de investigación.

El prototipo tiene la característica de que es ajustable a cambios del entorno en el cual se ejecuta como lo es la modificación de las dimensiones del área de trabajo, al igual que el número y posición de elementos a tratar.

Se utiliza la plataforma Arduino para el control de la posición de los actuadores con los cuales trabajará el prototipo, así como la lectura de parámetros que serán obtenidos del entorno con ayuda de sensores interconectados al sistema.

Las principales aplicaciones del prototipo son: fertirrigación, riego por goteo, monitoreo de cultivos experimentales.

## Introducción

Según (SANCHEZ V, 2000), Fertirrigación o fertigación, son los términos para describir el proceso por el cual los fertilizantes son aplicados junto con el agua de riego. Este método es un componente de los modernos sistemas de riego actuales. Con esta técnica, se puede controlar fácilmente la parcialización, la dosis, la concentración y la relación de fertilizantes.

Otro sistema que es posible implementar en el prototipo es el riego por goteo en donde (Groppa, 2014) lo define como: El riego por goteo es uno de los sistemas de riego más eficientes, efectivos, prácticos y económicos, no sólo por su menor uso del agua sino por las economías logradas en el uso de mano de obra y fertilizantes además de incrementar drásticamente la producción y la calidad.

La aplicación de tratamientos en cultivos sirve para mejorar la calidad de los productos, aumentar la producción y disminuir los costos de producción.

El desarrollo de la investigación que involucra el aplicar sustancias o productos de manera repetitiva para el desarrollo o prueba de nuevos fertilizantes, así como pruebas en la inyección de distintas cantidades de agua .Requiere de tiempo y dedicación para aplicar los tratamientos de manera adecuada y obtener los resultados esperados.

Se propone la construcción de un prototipo movable en 3 dimensiones para automatizar el proceso de aplicación de las sustancias requeridas en el proceso de investigación de manera automática sin la intervención humana, para así obtener resultados más concretos.

## Planteamiento del Problema

La automatización y el uso de robots en la investigación tienen un costo alto, debido a que dicha tecnología debe ser importada y en ocasiones se desarrolla sobre pedido a una empresa especializada.

La aplicación de tratamientos (aplicación de sustancias, mediciones de características, entre otras) en la investigación debe realizarse de la manera más exacta posible para no influir sobre los resultados finales y poder determinar resultados válidos y eficientes en las ciencias.

No existe una tecnología de manufactura nacional que ayude a el monitoreo y aplicación de tratamientos que faciliten y optimicen el trabajo de investigadores de campo, ya que al ser México un país en el cual la agricultura es una de sus actividades primarias que carece de importancia en gran parte del consumo, en este aspecto provienen de origen extranjero, considerando que pueden ser generados en México.

# Objetivos

## Objetivo General

Generar un prototipo para la optimización de tratamientos a cultivos, la cual se pretende automatizar el monitoreo e inyección de los éstos tratamientos en diversos tiempos establecidos.

## Objetivos Específicos

- Desarrollo del movimiento automatizado en tres dimensiones para la aplicación de tratamientos
- Implementar los sistemas de detección de objetos mediante sensores ultrasónicos
- Desarrollo de las rutinas de software para la calibración del área de trabajo

## Justificación

Se ayuda a los investigadores en la realización de proyectos y obtención de resultados validos con el fin de optimizar de manera eficiente las cantidades exactas de los tratamientos y así poder maximizar sus resultados, se disminuyen los gastos y utiliza materiales de fácil adquisición.

Este prototipo genera un gran impacto dentro de los campos de cultivos para tener un mejor control de las cantidades de tratamientos, cabe mencionar que al manejar una aplaca Arduino se puede controlar, tiempos, cantidades y movimientos del prototipo para ser más eficientes, lo cual presenta una respuesta innovadora a la problemática presentada.

Al no existir empresas nacionales que genere este tipo de tecnología, es necesario importar de otros países, esto genera costos elevados, por esta razón se pretende la implementación del prototipo presentado para poder así reducir considerablemente los costos, ya que al ser construido con materiales de fácil acceso y con la implantación de Software y Hardware.

## Metodología

Para la construcción del software se utilizara la metodología de proceso evolutivo que se basa en la creación de una implementación parcial de un sistema, para el propósito explícito de aprender sobre los requerimientos del sistema. Un prototipo es construido de una manera rápida tal como sea posible. Esto es dado a los usuarios, clientes o representantes de ellos, posibilitando que ellos experimenten con el prototipo. Estos individuos luego proveen la retroalimentación sobre lo que a ellos les gustó y no les gustó acerca del prototipo proporcionado, quienes capturan en la documentación actual de la especificación de requerimientos la información entregada por los usuarios para el desarrollo del sistema real.

Las fases que comprende el método de prototipos evolutivos son:

- **Fase de Investigación preliminar.** Las metas principales de esta fase son: determinar el problema y su ámbito, la importancia y los efectos sobre la organización por una parte y, por otro lado, identificar una idea general de la solución para realizar un estudio de factibilidad que determine la viabilidad de una solución software.
- **Fase de definición de los requerimientos del sistema.** El objetivo de esta etapa es registrar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto bajo desarrollo. Esta etapa es la más importante de todo el modelo, es aquí donde el desarrollador determina los requisitos mediante la construcción, demostración y retroalimentaciones del prototipo. El objetivo del análisis de sistemas es comprender situaciones, no resolver problemas. Por tanto, se debe conocer el modo de operación del sistema e identificar los requerimientos que tienen los usuarios para modificarlo o proponer un nuevo análisis.
- **Fase de diseño técnico.** Durante la construcción del prototipo, el desarrollador no ha realizado el diseño detallado. El sistema debe ser entonces rediseñado y

documentado según los estándares de la organización y para ayudar al mantenimiento del sistema. Esta fase de diseño técnico tiene dos etapas: por un lado, la producción de una documentación de diseño que especifica y describe la estructura, el control de flujo, las interfaces de usuario y las funciones, y como segunda etapa, la producción de todo lo requerido para realizar cualquier mantenimiento futuro al software.

- **Fase de implementación y pruebas.** Los cambios son identificados en el diseño técnico, son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.

## Antecedentes

### **Automatización de un sistema de fertirrigación por goteo.**

Se desarrolló un programa de cómputo para el manejo y control de la fertirrigación, en el lenguaje de programación Edlog, y se empleó un Datalogger CR10X<sup>1</sup>, como dispositivo de control. Esto, con la finalidad de contar con una herramienta para calcular de manera automática la cantidad diaria de solución nutritiva para un cultivo, en función de sus etapas fenológicas y aplicarla en el momento oportuno. (Cruz Bautista, 2007)

### **Programador ACTIVA**

Indicado para explotaciones de pequeña y mediana escala, espacios verdes: parques, jardines, campos de fútbol, etc. Se pueden activar los distintos elementos que intervienen en la gestión del riego: bombas, válvulas, inyectoros, agitadores, limpieza de filtros, etc. Cabe mencionar que puede ser suministrado para empotrar, en caja individual o montado en cuadro de comando automático o manual. Cuenta con la opción de llevar un plano de la finca en serigrafía, para una mejor visualización de los sectores de riego y cultivos, con iluminación de aquellos que estén activos. (Hermisan, 2015)

### **Hidrocomputador SUPRA**

Este controlador es una completa herramienta de gestión que incluye un exclusivo diseño de hardware y software. Está compuesto por un sistema electrónico gestionado por un PC industrial, y por un potente software PLC<sup>2</sup> basado en Windows y los distintos programas de aplicación desarrollados bajo dicha plataforma informática. Puede controlar, simultánea e independientemente, hasta 10 instalaciones completas, tanto en campo abierto como en invernadero y permite

---

<sup>1</sup> El Datalogger CR10X es un standard para meteorología.

<sup>2</sup> Un controlador lógico programable o PLC, es un dispositivo digital que se utiliza para automatizar los procesos electromecánicos.

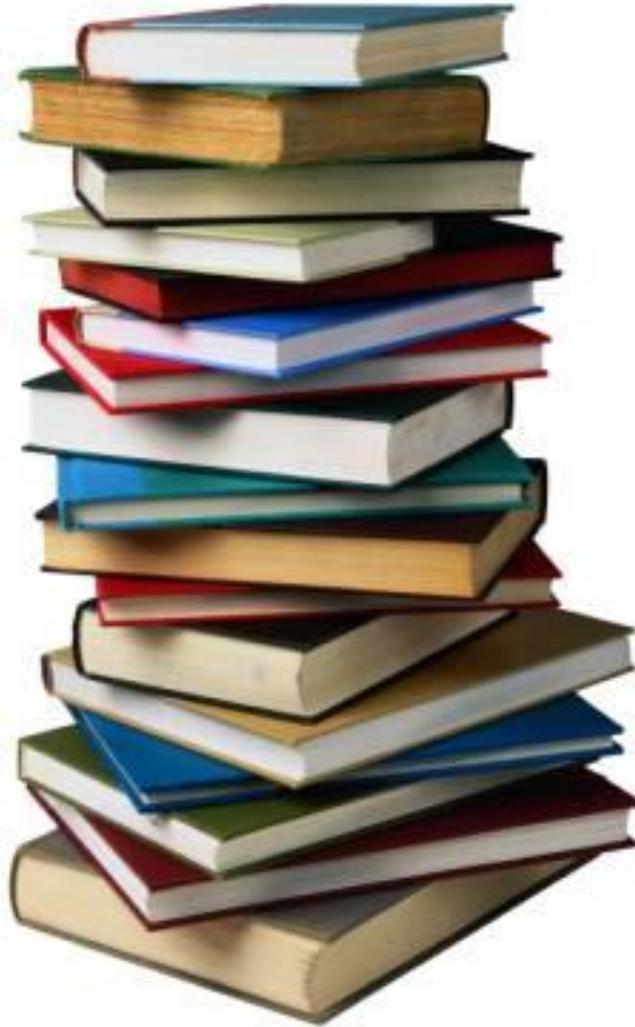
compartir elementos entre ellas como bombas, tanques de fertilizantes, filtros, contadores de caudal, etc. Tiene una capacidad práctica que permite gestionar programas, sectores, abonados, zonas climáticas, etc., lo que permite al usuario contar con una poderosa herramienta de gestión para el control de su riego, fertirrigación y clima. Cuenta con características como: su carácter modular, flexible y actualizable. (Hermisan, 2015)

## **TRENES DE RIEGO**

Los Trenes de Riego permiten una perfecta distribución del riego y la fertirrigación en la producción de plantas en semilleros. Su automatización se realiza por medio de un programador MERIDIAN, pudiéndose conectar múltiples trenes en red para su gestión remota desde un PC, lo que permite controlar la totalidad del proceso productivo en estos cultivos de una forma eficaz, cómoda y segura.

El desplazamiento del tren se realiza por medio de 2 raíles de aluminio bajo los cuales se halla suspendido lo que ofrece un producto de mayor robustez y durabilidad al evitar vibraciones y daños en la estructura, los soportes y herrajes han sido tratados previamente para resistir las condiciones climáticas de los semilleros, permite la programación de hasta 8 velocidades para el desplazamiento del tren, con arranques y paros suaves mediante moto reductores con variadores de velocidad, puede seleccionar diferentes zonas de riego o abonar dentro de cada línea de semilleros, excluyéndose las calles de paso. (Hermisan, 2015)

## Capítulo I. Marco Teórico.



---

En este primer capítulo se dota al lector de conocimientos acerca del tema de este trabajo, iniciando principalmente sobre el concepto de automatización, la cual requiere de diferentes instrumentos de medición, toda esta descripción es con el fin de tener un mayor entendimiento de cada proceso en el del prototipo con movimiento.

## Automatización

Como se menciona en (Automatización Industrial, s.f.), La automatización es un proceso mediante el cual se simplifica y mejora la realización de un trabajo teniendo como base la conjunción de tecnologías.

Para la automatización se utilizara Arduino que es una plataforma de Hardware Open Source para la creación de prototipos basada en software y hardware flexibles y fáciles de usar, Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando; luces, motores y otros actuadores., en la parte de la investigación se utilizara para controlar los motores que colocaran el prototipo en el lugar indicado para así poder realizar las acciones necesarias.

## Funciones Trigonométrica

Existen diversas funciones trigonométricas pero para este caso solo se hará referencia a la función seno la cual es definida por (Campos O. , 2000) como, *Seno*: Es la razón entre el cateto opuesto y la hipotenusa. Si llamamos al cateto opuesto C.O. y a la hipotenusa Hip, esta función se pueden escribir así:

$$\text{sen} = \frac{\text{C.O.}}{\text{Hip.}}$$

Para convertir radianes a grados (Campos O. , 2000) menciona que es necesario multiplicar el número de grados por el factor  $\pi/ (180^\circ)$  el resultado se simplifica de der posible.

## Voltaje

Está definido por (Allier, 2006) como la magnitud física que, en un circuito eléctrico, impulsa a los electrones a lo largo de un conductor. Es decir, conduce la energía eléctrica con mayor o menor intensidad.

## Corriente

(Allier, 2006) Lo señala como el flujo de carga eléctrica que circula a través de un material por unidad de tiempo.

## Motores pasó a paso

Como se menciona en (TodoTobot, s.f.) Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ), para completar un giro completo de  $360^\circ$ . Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizada, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.



*Figura 1 Motor pasó a paso*

## Servomotor

(Lozano, 2015) Tiene la idea que un Servomotor en su definición más básica, es un motor que puede ser controlado en su velocidad de funcionamiento y en la posición dentro de un rango de operación para ejecutar una actividad requerida. Este control es realizado mediante un dispositivo llamado encoder<sup>3</sup>, que mediante una señal electrónicamente codificada, indica las acciones de velocidad y movimiento a ejecutar. El servomotor es instalado en un equipo o máquina, para permitir que esta tenga control de la posición, dirección y velocidad de una carga o herramienta, mediante su utilización.

---

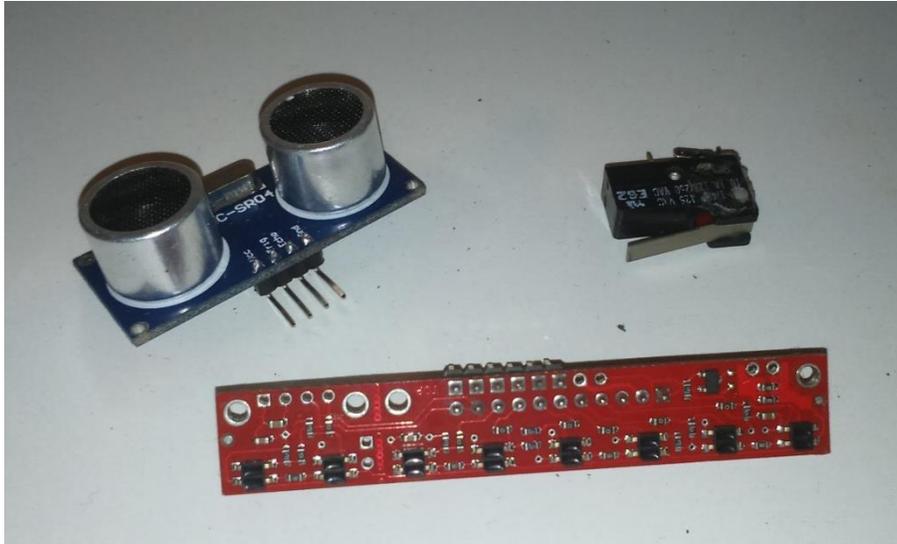
<sup>3</sup> Un Encoder suele ser un dispositivo electromecánico usado para convertir la posición angular de un eje a un código digital.



*Figura 2 Servomotor*

## Sensor

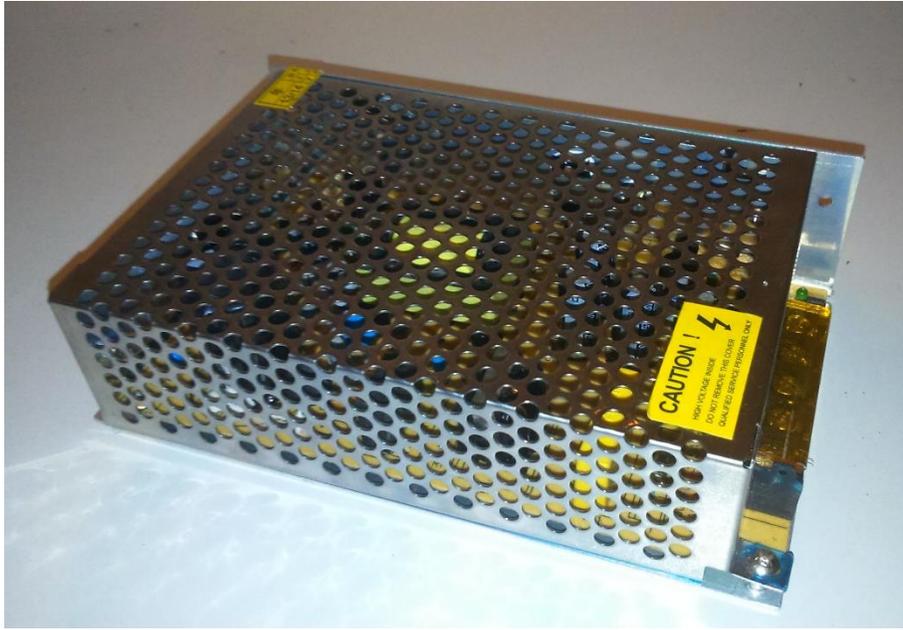
Como se puede encontrar en (ISA, 2014), un sensor es un dispositivo eléctrico o mecánico que convierte magnitudes físicas (luz, magnetismo, presión, etc.) en valores medibles de dicha magnitud. Esto se realiza en tres fases: Un fenómeno físico a ser medido es captado por un sensor, y muestra en su salida una señal eléctrica dependiente del valor de la variable física. A continuación la señal eléctrica es modificada por un sistema de acondicionamiento de señal, cuya salida es un voltaje. Posteriormente se transforma o amplifica la tensión de salida, la cual es recibida por el dispositivo controlador.



*Figura 3 Tipos de sensores*

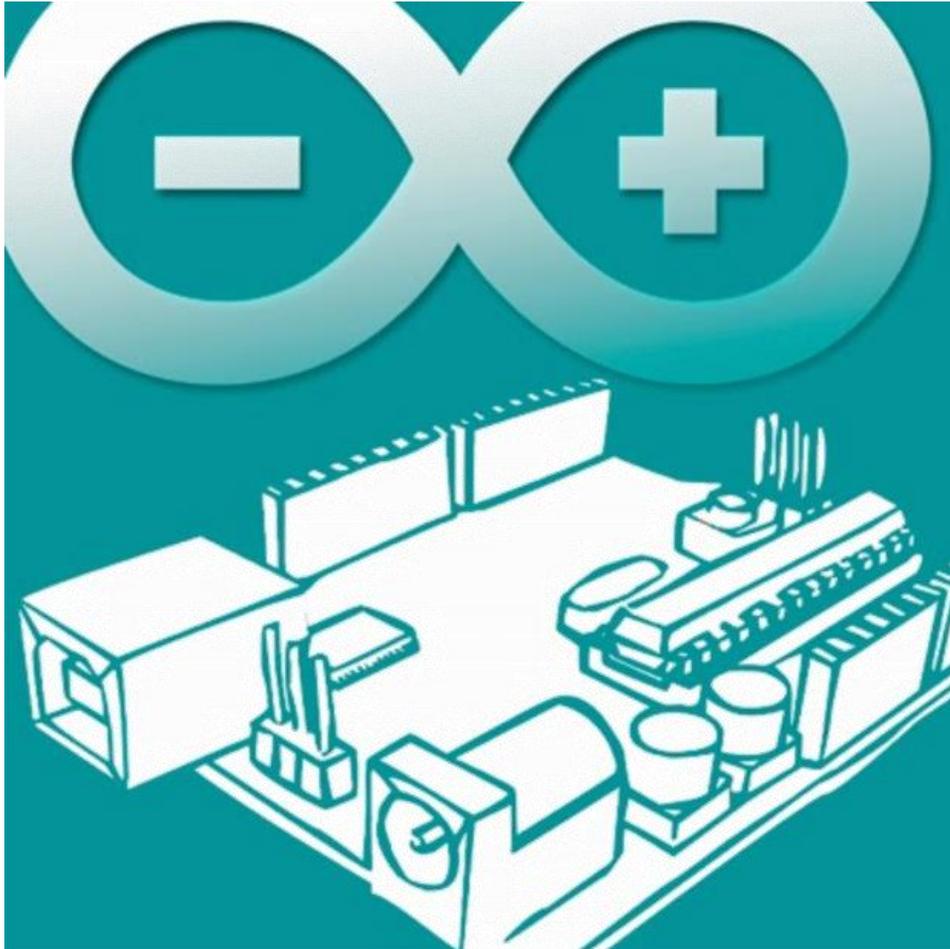
## Fuentes Reguladas Conmutadas

(Mohan, Undeland, & Robbins, 2003) Mencionan que la topología básica de una fuente regulada conmutada está compuesta por una etapa de potencia, compuesta por un convertidor AC/DC y un convertidor conmutado (convertidor DC/DC), así como una etapa de control compuesta por una red de realimentación y su respectivo controlador. La utilización de estos dispositivos se da en innumerables aplicaciones dentro de todos los campos de estudio de la electrónica.



*Figura 4 Fuente conmutada*

## Capítulo II. Arduino.



---

En el primer capítulo se explicó los conceptos y aspectos necesarios para el desarrollo del prototipo, donde se mencionaron los elementos que serán utilizados, ahora en este capítulo se hará mención principalmente de las características y ventajas de este dispositivo para la creación de prototipos que son capaces de interactuar con el medio que los rodea, haciendo un énfasis en la versión de placa que se utiliza para el desarrollo del prototipo de movimiento.

Hay muchas otras plataformas disponibles para desarrollo de prototipos. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, y muchas otras con funcionalidad similar. Todas estas herramientas toman los desordenados detalles de la programación del microcontrolador y la encierran en un paquete fácil de usar. Arduino simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas, como son:

- Precio: Las placas Arduino son relativamente baratas comparadas con otras plataformas. La versión más económica puede encontrarse por unos \$382 si es oficial, fabricada por el equipo de Arduino, o por \$200 en caso de ser fabricada por otros.
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux.
- Entorno de programación simple y clara: El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores ATMEGA de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos

pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

Concepto Arduino.

Arduino es una plataforma de hardware libre (Open Hardware), basado en un micro controlador Atmel.

Existen muchas definiciones referentes a la plataforma Arduino, como menciona (Banzi, 2011), Arduino es una plataforma de computación física de código abierto basado en entradas/salida (E/S) y un entorno de desarrollo que implementa el Lenguaje Processing

En el mundo de hoy, el diseño de interacción se refiere a la creación de experiencias significativas entre nosotros y los objetos, esta plataforma trata de una buena manera de explorar la tecnología. El diseño que propone Arduino de Interacción alienta a un proceso basado en prototipos. Esto realmente implica crear y diseñar objetos interactivos que puedan comunicarse con los seres humanos por medio de sensores y actuadores controlados por un comportamiento implementado como el software se ejecuta dentro de un micro controlador.

Anteriormente para poder elaborar algún dispositivo que interactuara directamente con una o varias personas, requería de conocimientos en distintas áreas. Arduino y otras plataformas han logrado introducir a muchas personas en este tipo de desarrollo ya que no requiere de grandes conocimientos. Con Arduino, (Banzi, 2011) señala que un diseñador o artista puede llegar a conocer los conceptos básicos de la electrónica y sensores muy rápidamente, esto permite comenzar a construir prototipos con muy poca inversión y pocos conocimientos.

Su potencialidad, aseguran sus creadores es enorme, ya que desde el inicio de este proyecto de hardware libre la colaboración entre los usuarios es un punto clave en materia de Arduino, ya que existe un foro que se puede encontrar en <http://www.arduino.cc> donde las personas de diferentes partes del mundo se ayudan mutuamente a aprender acerca de la plataforma. Así como una gran cantidad de sitios no oficiales que manejan el mismo concepto de colaboración,

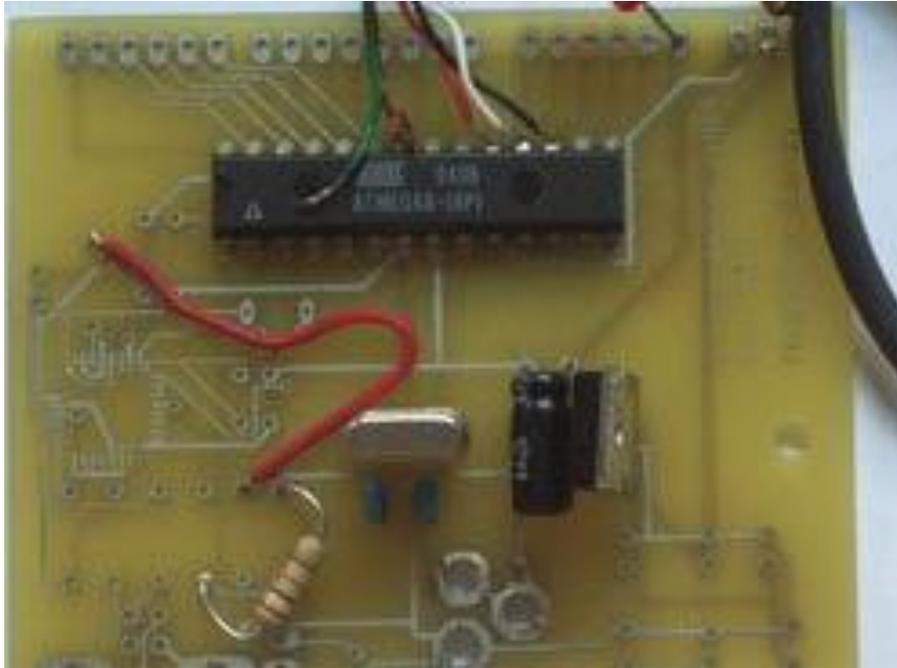
como lo son ElectroniLab (<http://electronilab.co/tutoriales>), Intruncables (<http://www.instructables.com>) y Hacedores (<http://hacedores.com>) por mencionar algunos.

Por experiencia propia en el desarrollo de este proyecto de tesis, la placa Arduino es un incentivo para colaborar y adentrarse innumerables proyectos de todo tipo, esto ayuda a conocer usuarios y diferentes espacios para colaborar en proyectos, este tipo de actividades de compartir y ayudar es un aspecto más que ayuda al rápido aprendizaje de todo el entorno Arduino.

Por lo tanto puedo definir a Arduino como una plataforma para crear proyectos interactivos que consta básicamente de dos partes, la placa Arduino, que es la pieza de hardware y el IDE de Arduino, la pieza de software que se ejecuta en un equipo.

#### Hardware Arduino

En el mercado existen muchas versiones del hardware, ya que Arduino nació como un proyecto educativo por el año 2005 sin pensar que algunos años más tarde este se convertiría en líder del mundo DIY (Do It Yourself o en español hágalo usted mismo). El primer prototipo fue desarrollado en el instituto IVRAE y tenía un aspecto como se muestra en la Figura 5.



*Figura 5 Prototipo del primer Arduino*

El hardware de Arduino se distingue por la potencia de trabajo que tiene cada versión y algunas de las características principales que destacan uno de otro es el micro controlador con el que trabaja cada placa, también una característica que hace la diferencia es la cantidad de entradas/salidas analógicas y digitales .

En la actualidad existe una gran variedad de placas, cada una con características diferentes a las de los demás, con ello se obtiene una amplia gama de opciones para poder elegir la que se adapte más al propósito deseado. En la Tabla 1 se enlistan las placas disponibles actualmente entre ellas la placa Arduino Mega2560, que fue la seleccionada ya que sus características se adaptan a las necesidades del prototipo.

Modelo	Microcontrolador	Voltaje (V)	I/O digitales	Entradas Analógicas	Memoria Flash	Velocidad de reloj
UNO	ATmega328	5	14	6	32 KB	16 MHz
Leonardo	ATmega32u4	5	20	12	32 KB	16 MHz
Due	AT91SAM3X8E	3.3	54	12	512 KB	84 MHz
Yún	ATmega32u4	5	20	12	32 KB	16 MHz
Robot	ATmega32u4	5	5	4	32 KB	16 MHz
Esplora	ATmega32u4	5			32 KB	16 MHz
Mega ADK	ATmega2560	5	54	16	256 KB	16 MHz
Ethernet	ATmega328	5	14	6	32 KB	16 MHz
<b>Mega 2560</b>	<b>ATmega2560</b>	<b>5</b>	<b>54</b>	<b>16</b>	<b>256 KB</b>	<b>16 MHz</b>
Mini	ATmega328	5	14	8	32 KB	16 MHz
Nano	ATmega168	5	14	8	16 KB	16 MHz
Pro Mini	ATmega168	3.3	14	8	17 KB	8 MHz
Pro	ATmega168	3.3	14	6	16 KB	8 MHz
Micro	ATmega32u4	5	20	12	32 KB	16 MHz
Fio	ATmega328P	3.3	14	8	32 KB	8 MHz

Tabla 1 Versiones de Arduino (Arduino, 2015)

A continuación se muestra en la Figura 6 se muestra la forma física del Arduino Mega si como los elementos que lo componen, esta placa es la que cumple con los elementos necesarios para el desarrollo del prototipo.

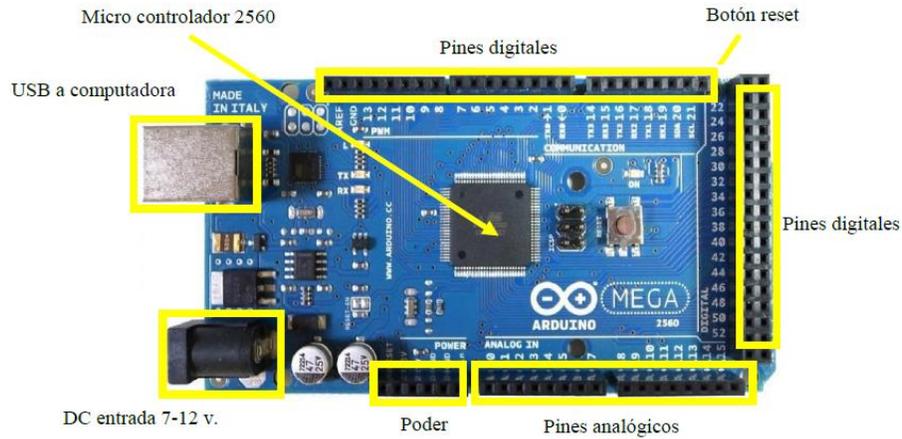


Figura 6 Componentes de Arduino Mega

En la Tabla 2 se muestran específicamente las características técnicas del Arduino Mega 2560.

Micro controlador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada recomendado	7-12V
Voltaje de entrada limite	6-20V
Digital entrada/salida Pines	54 (15 pines de salida PWM )
Pines de entrada analógicos	16
DC corriente Pin	40 mA
DC corriente 3.3V Pin	50 mA
Memoria flash	256 KB - 8 KB
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Tabla 2 Características técnicas Arduino Mega (Arduino, 2015)

## Entradas/Salidas de Arduino

Los pines digitales pueden ser entradas o salidas, es decir tienen la capacidad de obtener valores externos a través de dispositivos (sensores o captadores) que funcionen con valores digitales (valores que se reducen a 0 o a 1).

Los pines analógicos están dedicados específicamente como entradas, estos tienen la capacidad de obtener valores de dispositivos externos que van desde 0 hasta 5v. Otros pines digitales pueden ser reprogramados para salida analógica mediante el IDE.

## Shield de Arduino

Casi todas las placas de Arduino cuentan con Shield de expansión. Estos Shield cuentan con diferentes módulos para ampliar las posibilidades del Arduino sin tener que soldar nada. Por lo que son una solución fiable a las necesidades de expansión de características como es una pantalla LCD, o bluetooth, WiFi, Ethernet, 3G, GPS, Cámara, botones, leds, sensores, etc.

Cada Shield se conecta al Arduino a través de pines, además a su vez cada uno puede conectarse encima de otro aumentando así su funcionalidad. Sin embargo tenemos que tener en cuenta que cada placa para funcionar necesitara unos pines determinados del Arduino y que el resto de placas, esto implica que no es posible acoplar todos los Shield disponibles si estos ocupan los mismos pines para realizar su función. Por ello es recomendable consultar el datasheet<sup>4</sup> de cada Shield donde indican que pines usa cada uno, para poder realizar el acoplamiento adecuando de más de un Shield.

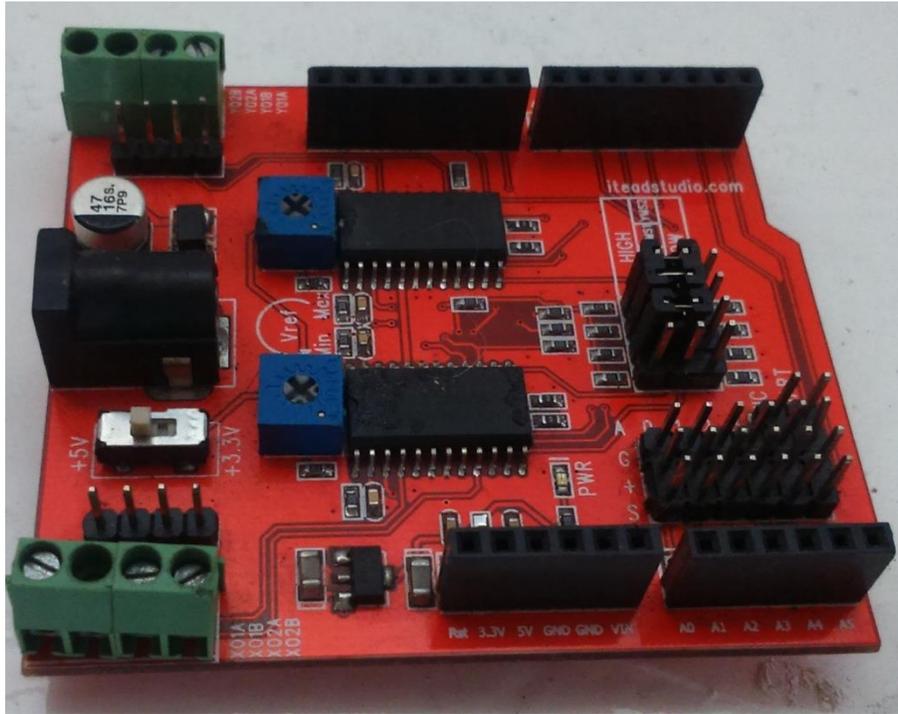
A continuación se hará mención de los Shield que fueron considerados para el desarrollo de este prototipo y las razones por que fueron o no utilizados:

---

<sup>4</sup> Un datasheet es un documento que resume el funcionamiento y otras características de un componente o subsistema con el suficiente detalle para ser utilizado.

### *ITEAD Dual Step Motor Driver Shield*

En el sitio web (ITEAD, 2015) se menciona que este Shield es capaz de controlar 2 motores paso a paso al mismo tiempo, trabaja con una fuente de alimentación de 4.75V a 30V. Ofrece una alta precisión en el control y es principalmente usado en máquinas CNC<sup>5</sup>.



*Figura 7 Dual Step Motor Driver Shield*

Este componente funciono con el primer prototipo peor al incrementar las dimensiones de este, no conseguir alimentar correctamente a los motores paso a paso con el amperaje necesario para poder realizar los movimientos deseados, además de que limitaría el crecimiento del proyecto debido a que solo de es posible manejar 2 motores.

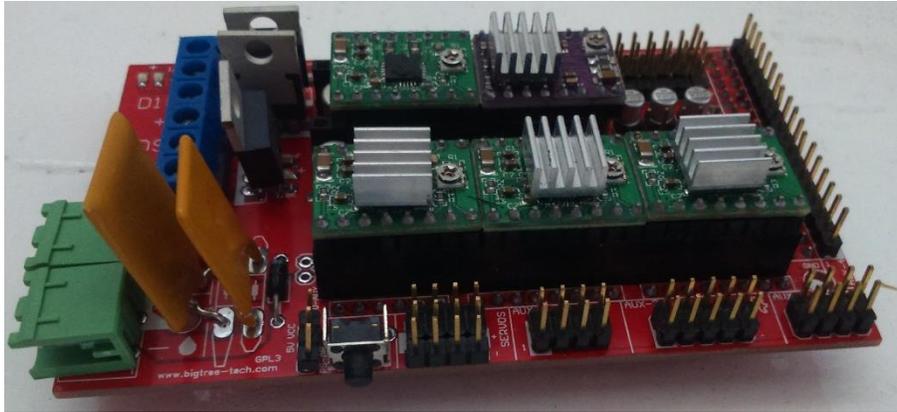
### **RAMPS 1.4**

(Caparrós, 2014) Menciona que el Shield RAMPS<sup>6</sup> está diseñado para albergar toda la electrónica de una impresora 3D en un pequeño pack de bajo precio. Al mismo

<sup>5</sup> control numérico computarizado

<sup>6</sup> RepRap Arduino Mega Pololu Shield

tiempo que se basa en la potente plataforma Arduino MEGA y tiene espacio suficiente para su expansión. El diseño modular del Shield incluye conectores para los drivers de los motores paso a paso y la electrónica del extrusor, para facilitar el mantenimiento, sustitución, actualización de piezas y capacidad de expansión. Además, siempre y cuando se mantenga la RAMPS en la parte superior de la pila, se pueden añadir otros escudos o tarjetas de expansión a Arduino.



*Figura 8 RAMPS 1.4*

Este Shield al ser diseñado principalmente para ser implementado en la construcción de una impresora 3D, pero no por ello no puede ser utilizado en otro tipo de prototipos, debido a que cuenta con potencial de crecimiento ya que es capaz de controlar hasta 5 motores paso a paso con su correspondiente driver Pololu A4988 como el que se muestra en la Figura 9, el problema que apareció fue que en diversas ocasiones los Driver se quemaban dejando el prototipo inmóvil, por este motivo es que fue descartado el uso del mismo.



*Figura 9 Driver Pololu A4988*

## SENSOR SHIELD

El propósito que se menciona en (ArduinoInfo, 2015) del Sensor Shield, es que sea fácil de conectar los cables y los dispositivos a los pines de Arduino correctos. No es un dispositivo activo. Simplemente conecta los pines de Arduino a muchos conectores están listos para usar para conectarse a diversos dispositivos como servos y sensores con cables simples.

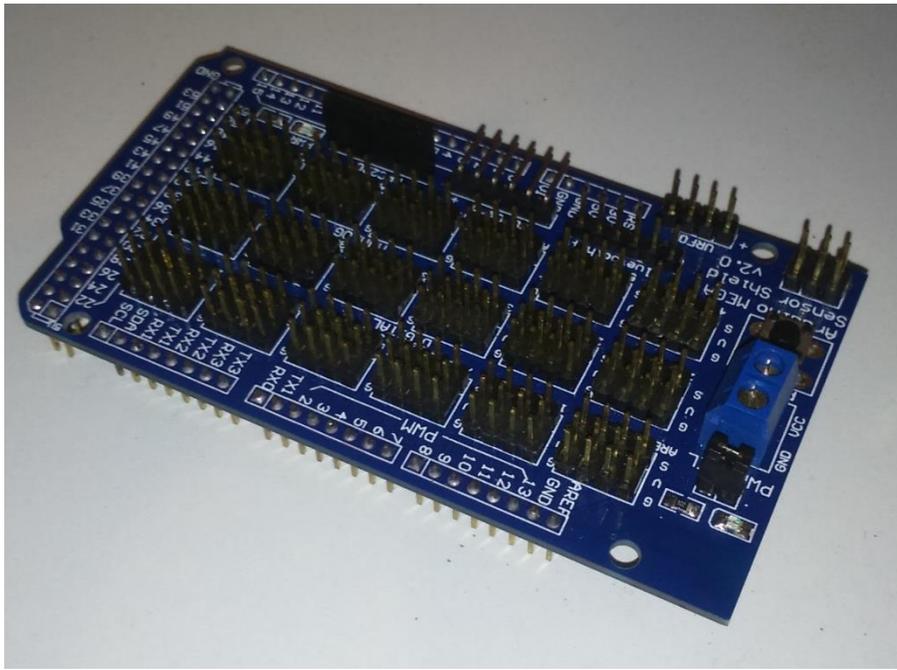


Figura 10 Sensor Shield

Como se menciona anteriormente este Shield no es un es un dispositivo activo, su única función es ayudar al montaje del prototipo reduciendo puentes y posibles fallas al conectar. Es una solución viable para prototipos cambiantes como lo es el que se presenta.

## 1Shield

En (1Sheeld, 2015) se describe a el 1Sheeld como dos partes. La primera parte es un Shield que está conectado físicamente a Arduino, la comunicación entre Arduino y cualquier Smartphone Android se realiza mediante una conexión Bluetooth. La

segunda parte es una plataforma de software para teléfonos inteligentes Android que gestiona la comunicación entre 1Sheeld y el teléfono inteligente y deja elegir entre diferentes Shield disponibles.

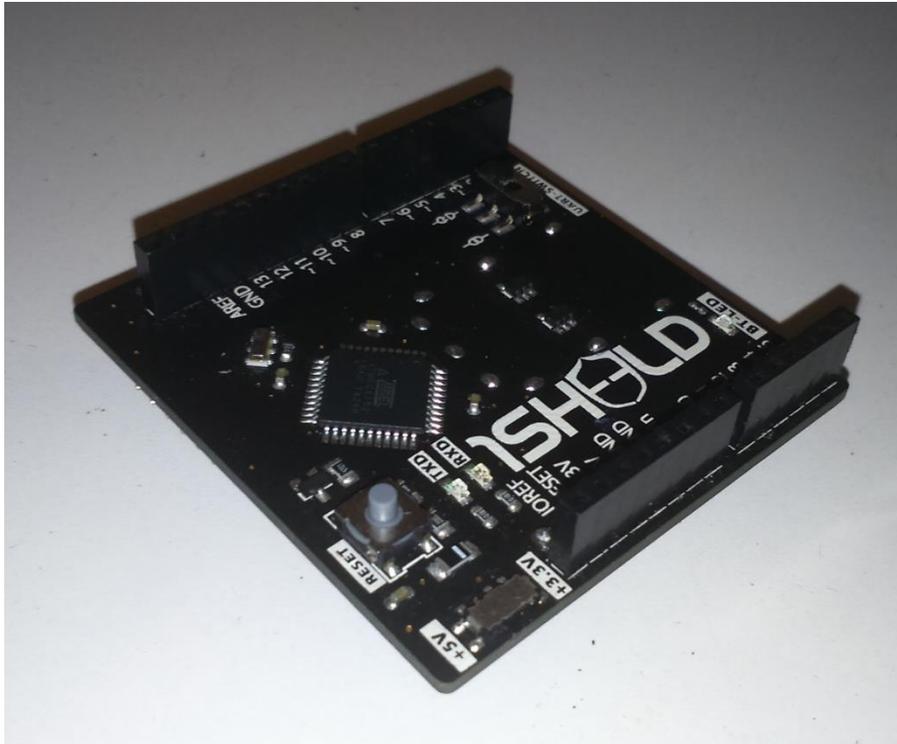


Figura 11 1Sheeld

Debido a que el principal objetivo es desarrollar un prototipo con movimiento es posible auxiliarse de herramientas como lo es este Shield para mantenerse enfocado en el objetivo que son las rutinas y el movimiento, ya que con este Shield se obtiene una gran cantidad de rutinas ya definidos que solo serán llamadas.

#### Módulo L298N

Es un módulo que cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, estos están bien descritos en (ELECTRONILAB Ingeniería y Diseño Electrónico, 2014), entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B), la salida A está conformada por OUT1 y OUT2 y la salida B por

OUT3 y OUT4, los pines de habilitación son ENA y ENB respectivamente. En la Figura 12 se muestra la composición y ubicación de los elementos antes mencionados.

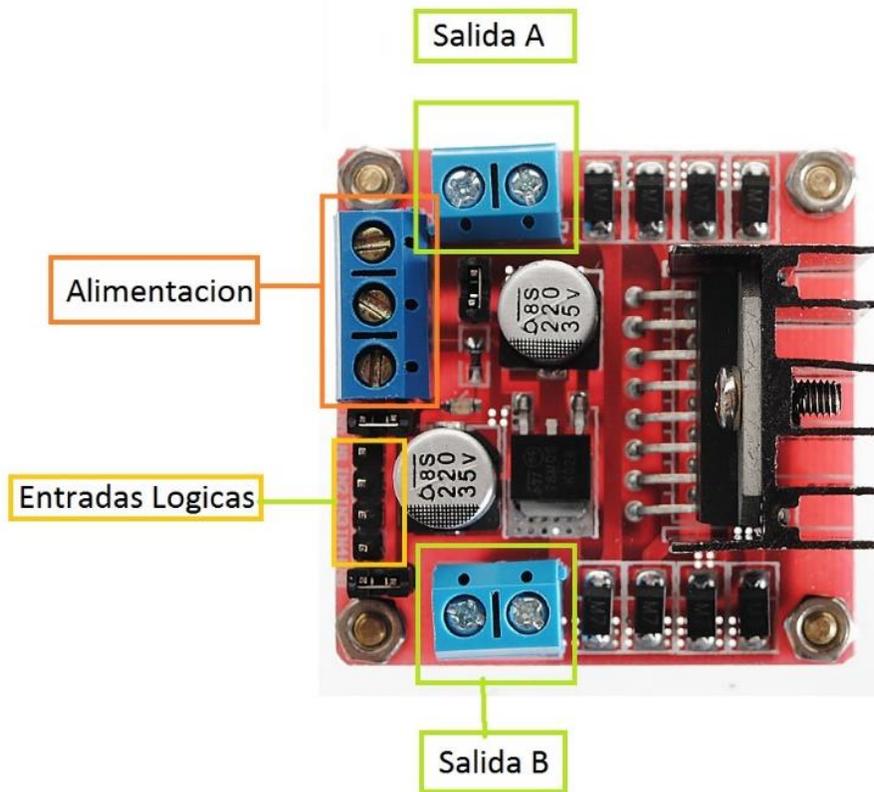


Figura 12 Modulo L298N

Hasta el momento este módulo ha sido la mejor opción para implementar el movimiento ya que es capaz de suministrar el amperaje necesario para que los motores sin quemarse. Con esto se obtiene el movimiento necesario.

El Software de Arduino IDE (Integrated Development Environment)

El IDE de Arduino es un programa especial que se ejecuta en cualquier equipo que permite escribir código para la placa Arduino en un lenguaje relativamente sencillo. El proceso que lleva a cabo este hardware puede resultar muy fácil de verse, (Banzi, 2011) señala este proceso, primero el código que se tiene escrito en el entorno o interfaz de programación se transfiere a la placa y se traduce en lenguaje C,

posteriormente este código se pasa a un compilador gcc avr, que hace la traducción final en el lenguaje entendido por el micro controlador ATmega.

El ciclo para programar en Arduino está compuesto por 4 pasos, según (Banzi, 2011) que se mencionan a continuación:

- Conectar tarjeta en un puerto USB de una computadora.
- Escribir un sketch (programa que contendrá el micro controlador).
- Cargar este sketch a la placa a través de la conexión USB y esperar un par de segundos.
- El Arduino ejecuta el sketch.

Instalación del software de Arduino en una computadora

El entorno de desarrollo que ofrece Arduino está disponible para distintos Sistemas operativos, como Linux, Windows y otros. El utilizado en este prototipo es el sistema operativo Windows 8.1

Para instalar el software es necesario primero descargarlo de la página oficial de Arduino <https://www.arduino.cc/en/Main/Software>, dentro de este sitio se muestran las diferentes opciones disponibles para ser descargadas, se debe de seleccionar el adecuado para el sistema operativo con el que se va a trabajar. El peso aproximado de la última versión disponible que es la Arduino 1.6.5 es de 77.6MB, el cual contiene el software primario de Arduino el paquete de descarga no solo incluye el software sino que también algunas de las librerías necesarias para utilizar funciones específicas (ej. Stepper contiene las funciones necesarias para controlar motores paso a paso), el paquete también contiene ejemplos muy claros de código que se pueden implementar en el hardware.

Una vez que se tiene el paquete se puede iniciar con la instalación de los controladores correspondientes para cada tipo o versión de Arduino, estrictamente pudiendo variar un poco la instalación, se señala una serie de pasos para poder iniciar el software con el hardware.

- Conectar la placa Arduino mediante los puertos USB

- Se iniciara una ventana de asistente de instalación de hardware el cual automáticamente intentara encontrar el controlador.
- Al no encontrar el controlador en el sitio de Windows update, se tendrá que elegir la opción de instalar un controlador manualmente.
- En el paquete de descarga también se puede encontrar los controladores por lo cual solo se tiene que elegir la ruta correspondiente, esta es /Arduino 1.0.4/drivers.

#### Entorno de desarrollo de Arduino

La interfaz de Arduino ha evolucionado dramáticamente a partir de su origen. La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. El entorno que presenta Arduino es un conjunto de menús a través de los cuales el usuario se comunica con el programa, se ha intentado desde su origen que este entorno sea amigable, es decir fácil de usar. La Figura 13 muestra la interfaz de programación.

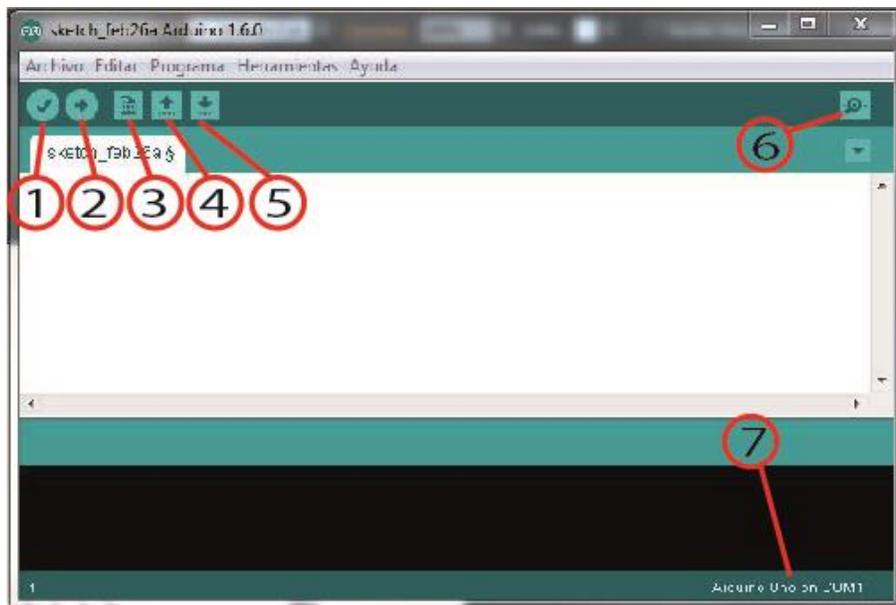


Figura 13 Entorno Arduino

1. Verificar: Comprueba si el programa tiene errores o no.
2. Subir: Grava el código en el microcontrolador.

3. Nuevo.
4. Abrir.
5. Guardar.
6. Abrir monitor serial.
7. Muestra el modelo de placa y el COM<sup>7</sup> del Arduino que esté conectado a la computadora.

Dentro de la interfaz se encuentra una barra que contiene opciones las cuales tienen funciones específicas además del acceso a las mismas opciones disponibles en la interfaz. La barra se muestra en la Figura 14.



*Figura 14 Barra de Herramientas*

Las principales opciones disponibles en esta barra es la de poder seleccionar el tipo de Arduino con la que se está trabajando y así mismo elegir el puerto COM en el que se encuentra ubicado el hardware. Es claro que existen distintas opciones que ayudan durante el trabajo o la construcción de un sketch.

#### Lenguaje de Programación de Arduino

En cuanto al lenguaje de programación que utiliza la plataforma Arduino está basado en programación Processing (lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java), dentro de los atributos que ofrece esta plataforma es el desarrollado un entorno de programación amigable para cualquier persona, que tenga los conocimientos necesarios para programar.

#### *Estructura de un sketch*

En cuanto a la estructura básica del lenguaje de programación de Arduino se puede decir que es bastante simple y se compone de al menos dos partes. Estas dos

---

<sup>7</sup> El COM es un puerto serie o puerto serial, es una interfaz de comunicaciones de datos digitales.

partes son necesarias en cualquier sketch (Código o parte lógica de programación de Arduino), cada una encierra bloques que contienen declaraciones o instrucciones.

(Banzi, 2011) Señala la estructura básica de un programa en Arduino como:

```
void setup () {  
    Contenido  
}  
  
void loop () {  
    Contenido  
}
```

En donde setup () es la parte encargada de recoger la configuración y loop () es la que contienen el programa que se ejecutará de manera cíclica (la parte del loop realizará la mayor parte del trabajo del programa).

La función setup () se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

```
void setup () {  
    pinMode (pin, OUTPUT); // configura el 'pin' como salida  
    digitalWrite (pin, HIGH); // pone el 'pin' en estado HIGH  
}
```

Después de llamar a setup (), la función loop () hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa.

```
void loop () {  
    digitalWrite (pin, HIGH); // pone en uno (on, 5v) el 'pin'  
    delay (1000); // espera un segundo (1000 ms)  
    digitalWrite (pin, LOW); // pone en cero (off, 0v.) el 'pin'
```

```
        delay (1000);
    }
```

Algunos otros conceptos que facilitan el entendimiento del sketch se mencionan a continuación:

### **{ } Entre llaves**

Las llaves sirven para definir el principio y el final de un bloque de instrucciones. Se utilizan para los bloques de programación `setup ()`, `loop ()`, `if`, etc.

```
type función () {
    Instrucciones;
}
```

Una llave de apertura “{” siempre debe ir seguida de una llave de cierre “}”, si no es así el programa dará errores.

### **/\* ... \*/ bloque de comentarios**

Los bloques de comentarios, o comentarios multi-línea son áreas de texto ignorados por el programa que se utilizan para las descripciones del código o comentarios que ayudan a comprender el programa. Comienzan con `/ *` y terminan con `* /` y pueden abarcar varias líneas.

### **// Línea de comentarios**

Una línea de comentario empieza con `//` y terminan con la siguiente línea de código. Al igual que los comentarios de bloque, los de línea son ignoradas por el programa y no ocupan espacio en la memoria.

### *Funciones en Arduino*

Lo que es conocido como función en otros lenguajes de programación resulta ser muy similar en esta plataforma ya que la sintaxis es muy parecida a otros, por lo tanto se puede decir que son bloques de código escritas por usuarios para realizar tareas repetitivas, las cuales ayudan a reducir el tamaño del programa.

Todas las funciones se declaran con un tipo de valor que está asociado a lo que devolverá la función, como resultado de la ejecución del bloque. Pueden existir funciones las cuales no requieran devolver algún valor por lo cual se tiene que definir con la palabra void que indica que el bloque no retornara nada, simplemente ejecutara un proceso. Es necesario dar un nombre al bloque ya que será con este con el que se llame dentro de la función loop (), posteriormente entre paréntesis se indicara si recibirá algún parámetro para que se pueda ejecutar.

```
void espera () {
    delay (1000);
}
```

#### *Tipos de Datos en Arduino*

Como todo lenguaje de programación es necesario utilizar variables, que son áreas de memoria de la placa Arduino, donde se podrá almacenar datos para poder manipularlos, por lo cual es necesario especificar el tipo de dato que contendrá. En la Tabla 3 se señala once tipos distintos de datos.

<b>Tipo de dato</b>	<b>Característica</b>
boolean	valores de verdadero o falso
char	Posee un carácter único, como A
byte	Números entre 0-255
int	Números entre -32,768 y 32,767
unsigned int	Números entre 0 y 65,535
long	Números -2,147,483,648 y 2,147,483,647
unsigned long	Números 0 y 4,294,967,295
float	Valores de punto flotante
double	Doble precisión de punto flotante
string	Cadena de caracteres
array	Variables con acceso por un índice

*Tabla 3 Tipos de datos en Arduino*

## Comunicación Serial con Arduino

Se tiene la posibilidad de comunicarse con dispositivos a través de USB, este puerto utiliza un protocolo de comunicación serial. Arduino puede conectarse a un puerto COM para comenzar a enviar y recibir datos en serie, generalmente se utilizan 9600 bits por segundo (bps) con el IDE Arduino serial, pero se puede usar otras velocidades que están disponibles, hasta 115.200 bps.

Por lo general en una computadora se pueden tener múltiples dispositivos conectados a distintos puertos COM, por lo que en la interfaz se tiene que elegir el puerto correspondiente, como se muestra en la Figura 15.

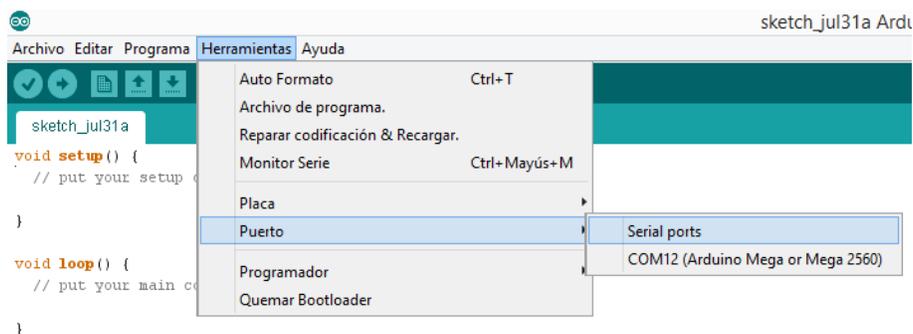


Figura 15 Puerto COM

Para poder utilizar esta opción dentro de un sketch es necesario indicarlo en la función `setup()` que como se menciono es la primera en ejecutarse. La sintaxis para iniciarlo es `Serial.begin(9600);`

Esta funcionalidad envía datos al puerto serie y la codificación en el sketch tiene distintas variaciones para poder utilizar la transferencia de datos al monitor, se debe decir que tales datos se tratan como texto plano.

## Prueba del Circuito con Arduino

Cuando todo el proceso de instalación de la placa con el software se ha realizado, el programador puede comenzar diseñando un sketch, en el cual está plasmando una idea de un objeto interactivo, este sketch realiza la lectura de valores a través

de ciertos sensores, todo esto con el fin de tener una respuesta reflejada en distintos actuadores.

El ensamble de un prototipo viene acompañado tanto de un sketch como de un circuito, ambos requieren de un cierto cuidado al momento de su elaboración ya que al no construir de forma correcta la lógica del sketch no se tendrán los resultados que se imaginan, de igual manera el circuito se debe de construir de forma que se acople correctamente a la lógica funcional.

Teniendo lista ambas partes, es necesario alimentar de corriente eléctrica al circuito es decir a cada uno de los componentes que lo requieran, en este punto donde se realiza una primera prueba pueden ocurrir inconvenientes, más que nada algún corto circuito. Arduino tiene sistemas de seguridad para evitar que alguno de sus componentes falle, al realizar las conexiones debidas si existiera algún corto circuito, un led verde dentro de Arduino que permanece encendido desde que tiene una fuente de alimentación, se apagara al detectar una acumulación de energía, lo que indica inmediatamente quitar o desconectar de la fuente de alimentación.

Para la solución de problemas en un circuito (Banzi, 2011) indica que solo existe un método, trabajar paso a paso depurar cada elemento y descartar una posible falla en él.

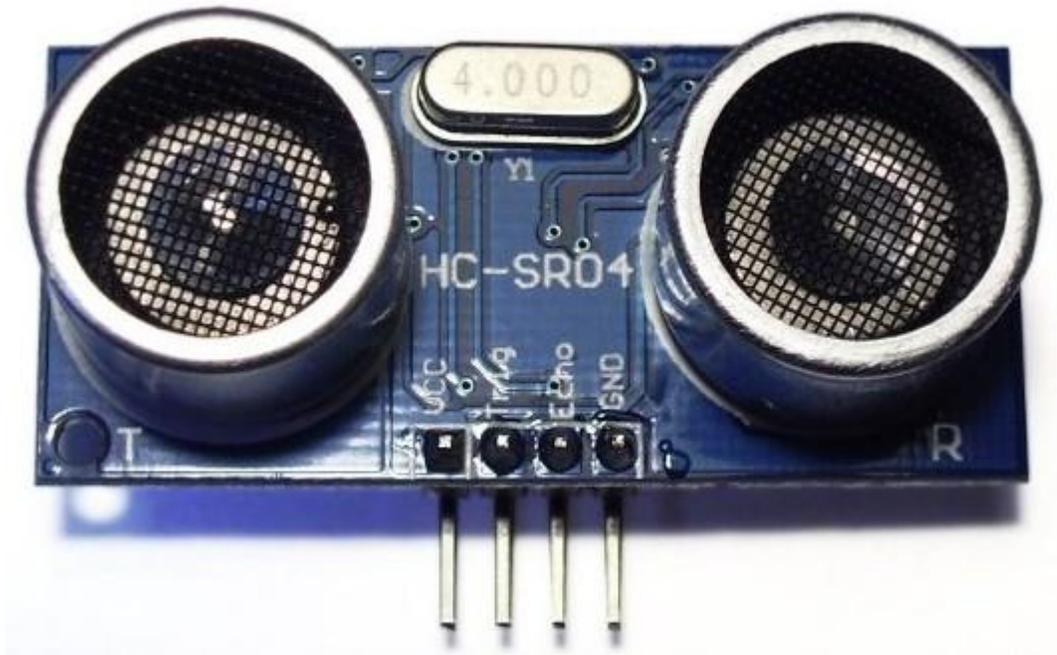
Así que para evitar cualquier inconveniente, antes de empezar a culpar al prototipo, se debe de asegurar que todo esté en orden, desde el software y la conexión con el hardware hasta el sketch y el circuito. Por tanto podemos describir en cuatro pasos las pruebas requeridas antes de comenzar a trabajar con un prototipo:

- Asegurarse de que la fuente de alimentación este encendida, esto realmente puede suceder. Si el led verde marcado con PWR (power) se enciende, significa que todo está en orden ahora bien si el led parece muy débil, algo está mal con la fuente: se puede probar con un cable diferente u otra fuente de alimentación.
- Asegurarse de que el puerto COM asignado a la placa, sea el mismo que esta seleccionado en la interfaz del software.

- Verificar la lógica del programa y compilar el sketch que se cargara a la placa Arduino.
- Comprobar las conexiones de cada elemento perteneciente al circuito e ir depurando paso a paso un posible corto circuito.

Por ultimo para descartar una falla general en la plataforma, se puede cargar a la placa un programa básico y así se comprobara el funcionamiento.

### Capítulo III. Sensores.



---

---

El objetivo del capítulo de sensores, es que el lector tenga un conocimiento más a fondo de este tipo de elementos, cuál es su funcionamiento y el por qué es son implementados en este prototipo.

## Concepto de Sensores

Hoy en día los sensores o captadores son una herramienta esencial para la industria, ya que con ellos se puede lograr medir, limitar o hasta automatizar procesos, se pueden ver en las casas, trabajos, hospitales y en lugares donde esté presente la tecnología, aunque muchas veces no podemos percatarnos de la existencia de estos elementos, ya que en la actualidad se han hecho común utilizarlos en dispositivos que diariamente se usan.

Resulta necesario tener un entendimiento más claro debido a que es el tema de este trabajo de tesis, (Kalpakjian, 2002) indica que un sensor es un dispositivo que produce una señal en respuesta a una detección o medida de una propiedad, como por ejemplo fuerza, torque, presión, temperatura, humedad, aceleración, vibración, velocidad y posición, precisamente estas dos últimas son parte del objetivo del desarrollo de la tesis.

Algunas personas al observar que estos dispositivos arrojan datos que necesitan ser transformados a valores entendibles para nosotros los llaman transductores. Ya que estos dispositivos arrojan valores distintos dependiendo de la lectura que realizan del mundo exterior es necesario tener una clasificación, para poder distinguirlos unos de otros.

## Clasificación de Sensores

En el trabajo presentado por (Campos P. A., 2008) Los sensores pueden ser clasificados por diferentes criterios:

- Por funcionamiento. Estos son clasificados según la manera en que obtienen parámetros del entorno.
- Según su aporte de energía. Básicamente se clasifican por la manera en que suministran la salida, ya sea con su alimentación propia o con ayuda externa.
- Según el tipo de salida (analógicos o digitales). Simplemente se dividen en la forma en que presentan su salida, ya sea analógica o digital.

Ahora bien los sensores enfocados a la manufactura se pueden clasificar de forma general como a continuación se muestra:

- Sensores mecánicos. Estos miden cantidades tales como posición, forma, velocidad, fuerza, presión y otros.
- Sensores eléctricos. Básicamente sirve para medir voltajes, corrientes y conductividad.
- Sensores magnéticos. Miden un campo o flujo magnético (son los sensores utilizados en esta tesis).
- Sensores térmicos. Para medir temperaturas y calor específico, además de otras variables.
- Otros. Dentro de esta categoría están aquellos para medir los sonidos, la luz radiación etc.

De acuerdo con la aplicación, un sensor puede estar formado por distintos materiales como, metal, no metálicos, orgánicos y por fluidos, gases o semiconductores. Algunos de los sensores obtienen valores a través de los materiales con los que están hechos y esos valores logran convertirlos en señales digitales o analógicas.

Sensores analógicos y digitales

Para el caso de este prototipo la clasificación importante a considerar es por el tipo de salida, según sea la forma de la señal de esta, un sensor se puede determinar si es analógico o digital.

Para poder entender la diferencia entre estos dos tipos de sensores a continuación se presenta la siguiente información.

La señal de salida de un sensor digital resulta ser mucho más simple que la de un analógico, pero se debe decir que hay una dificultad, ya que son pocos los sensores que logran dar directamente una señal digital en respuesta a una magnitud del exterior, los más comunes son los codificadores de posición, estos consisten en una regla o disco dotados de un sistema de lectura sensible a alguna propiedad. La Figura 16 muestra una señal digital que puede ser proporcionada por un sensor de este tipo.

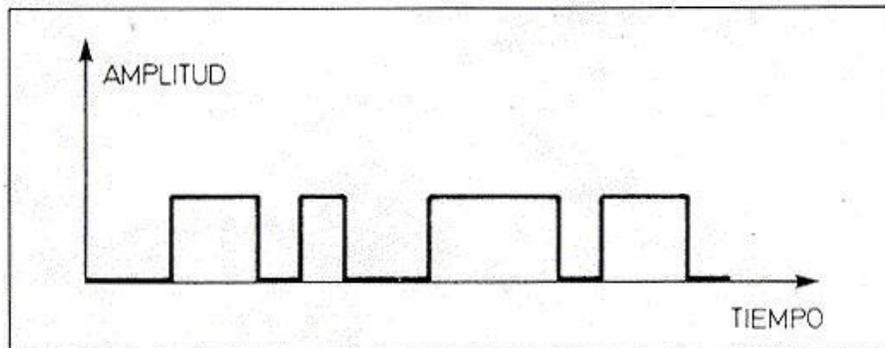


Figura 16 Señal Digital

Los captadores analógicos proporcionan una señal analógica continua, por ejemplo voltaje o corriente eléctrica. Esta señal puede ser tomada como el valor de la variable física que se mide. Este tipo de sensor debe poseer ciertas propiedades indispensables como:

Calibración, rango de funcionamiento, confiabilidad, velocidad de respuesta, exactitud precisión, sensibilidad, linealidad entre otros. Esto con el fin de que el control de la variable que se mida, se lleve a cabo de la mejor manera y en el menor tiempo posible.

Ahora en la Figura 17 se muestra una señal analógica que puede dar un sensor de este tipo.

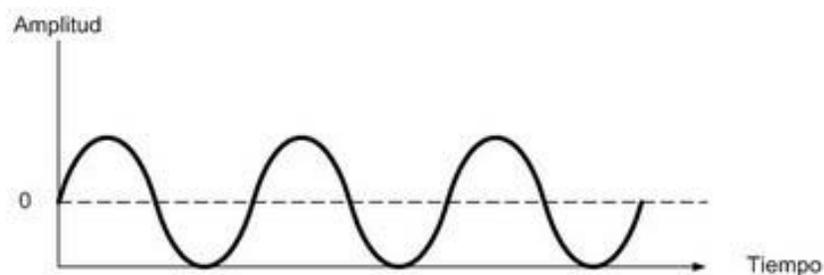


Figura 17 Señal Analógica

### Sensores de Distancia

En (Sen Sing, 2015) dice que los sensores de distancia, están pensados para realizar la medida de distancia lineal o desplazamiento lineal de una forma

automatizada, ya que proporcionan una señal eléctrica según la variación física, en este caso la variación física es la distancia.

Los rangos de medida disponibles son muy diversos, según el tipo de sensor de distancia empleado. Así pues hay modelos que tienen rangos de unas pocas micras y otros modelos que pueden llegar a medir cientos de metros.

#### HC-SR04 Sensor Ultrasónico

Según (Inteligencia artificial, 2015) El HC-SR04 es un sensor ultrasónico de bajo costo que no sólo puede detectar si un objeto se presenta, sino que también puede sentir y transmitir la distancia al objeto. Cuenta con dos transductores<sup>8</sup>, básicamente, un altavoz y un micrófono. Ofrece una excelente detección sin contacto con elevada precisión y lecturas estables en un formato fácil de usar. El funcionamiento no se ve afectado por la luz solar.



*Figura 18 HC-SR04 sensor ultrasónico*

---

<sup>8</sup> Dispositivo que tiene la misión de recibir energía de una naturaleza eléctrica, mecánica, acústica, etc., y suministrar otra energía de diferente naturaleza, pero de características dependientes de la que recibió.

Como se mencionó anteriormente este sensor permite obtener del ambiente la distancia que hay entre él y un objeto deseado, para posteriormente procesar este dato para poder colocar el dispositivo a una distancia considerable del objeto a analizar.

### Sensor de Temperatura

(Medirtemperatura, 2015) Define a los sensores de temperatura como dispositivos que transforman los cambios de temperatura en cambios en señales eléctricas que son procesados por equipo eléctrico o electrónico. Típicamente suele estar formado por el elemento sensor, la vaina que lo envuelve y que está rellena de un material muy conductor de la temperatura, para que los cambios se transmitan rápidamente al elemento sensor y del cable al que se conectarán el equipo electrónico.

### Sensor LM35

El LM35 es un sensor de temperatura con una precisión calibrada de  $1^{\circ}\text{C}$ . Su rango de medición abarca desde  $-55^{\circ}\text{C}$  hasta  $150^{\circ}\text{C}$ . La salida es lineal y cada grado centígrado equivale a  $10\text{mV}$ . No requiere de circuitos adicionales para calibrarlo externamente. La baja impedancia<sup>9</sup> de salida, su salida lineal y su precisa calibración hace posible que esté integrado sea instalado fácilmente en un circuito de control. Debido a su baja corriente de alimentación se produce un efecto de auto calentamiento muy reducido. (X-robotics, s.f.)

---

<sup>9</sup> Resistencia aparente de un circuito dotado de capacidad y autoinducción al flujo de una corriente eléctrica alterna, equivalente a la resistencia efectiva cuando la corriente es continua.

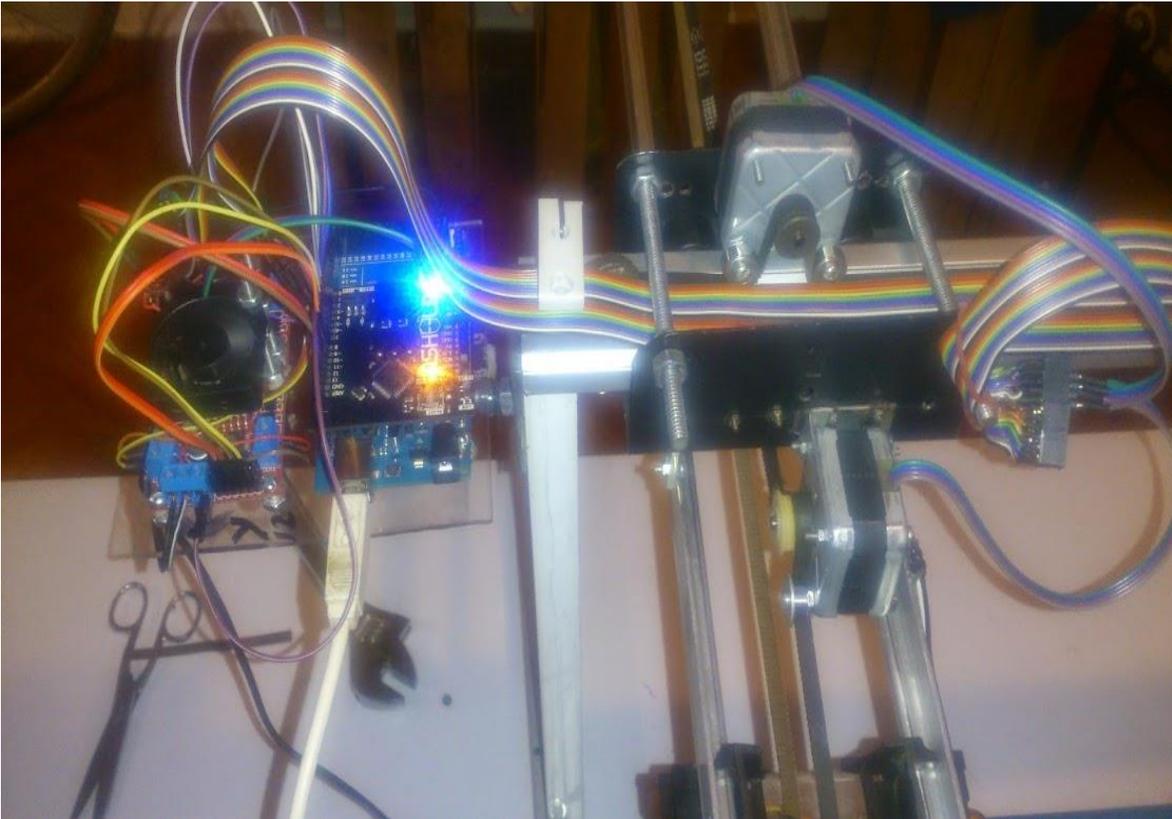




*Figura 20 Final de Carrera*

Al ser este un prototipo de movimiento siempre es necesario conocer las dimensiones con las cuales se cuanta. Por esta razón se implementan dispositivos como el que se muestra en la Figura 20 para poder limitar completamente el área de trabajo.

## Capitulo IV. Prototipo.



---

En este último capítulo se describe el proceso para obtener el prototipo final, así como los prototipos que le antecedieron y las razones del por qué fueron descartadas, se muestran as conexiones que se realizaron y se describirán las de software necesarias para el funcionamiento del mismo.

Fase de Investigación preliminar.

La automatización y el uso de robots en la investigación tienen un costo alto, debido a que dicha tecnología debe ser importada y en ocasiones se desarrolla sobre pedido a una empresa especializada. En este caso la automatización será implementada para la aplicación de tratamientos (aplicación de sustancias, mediciones de características, entre otras) en investigaciones en donde deben realizarse de la manera más exacta posible para no influir sobre los resultados finales y poder determinar resultados válidos y eficientes en las ciencias. Actualmente no existe una tecnología de manufactura nacional que ayude a el monitoreo y aplicación de tratamientos que faciliten y optimicen el trabajo de investigadores de campo, ya que al ser México un país en el cual la agricultura es una de sus actividades primarias, que carece de importancia en gran parte del consumo, en este aspecto provienen de origen extranjero, considerando que pueden ser generados en México.

La funcionalidad del prototipo se delimitara a lograr el movimiento en 3 ejes para poder aplicar los tratamientos necesarios, buscando que el prototipo se a adaptable a diversas dimensiones sin necesidad de modificar la programación del mismo, haciendo que este obtenga sus parámetros de funcionalidad de su entorno y sea capaz de adaptarse por sí solo.

A todo lo antes mencionado se le dará solución con la implementación de dispositivos como lo es Arduino, motores paso a paso, servomotores y sensores, todos interactuando para poder realizar el movimiento deseado.

Pruebas de Factibilidad

*La factibilidad operativa*

Dado que la interfaz del sistema, que fue elaborado con la clase java swing, es fácil de usar por parte de los usuarios y su instalación no requiere conocimientos detallados, ya que es totalmente intuitivo.

*La factibilidad técnica*

A la problemática presentada se le dará solución con la implementación de dispositivos como lo es Arduino, motores paso a paso, servomotores y sensores, todos interactuando para poder realizar el movimiento deseado. Todos estos dispositivos son de fácil adquisición y a bajo costo, por lo que representan una solución adecuada. Además de que este tipo de tecnología es Open Source y existen grandes comunidades que colaboran entre sí, no es necesario tener un gran

conocimiento acerca del tema, ya que es posible aprender fácilmente lo que sea necesario en el momento.

*La factibilidad económica*

Debido a que no existe manufactura nacional de este tipo de proyectos, si se desea implementar algo parecido es necesario exportarlos de otros países, lo que conlleva a un alto costo, al realizar este prototipo se pueden obtener las mismas funcionalidades con un bajo costo y los mismos resultados, además de que este contara con la opción de incremento de características aumentando así su valor.

Costo de desarrollo

Nombre	Precio	Cantidad	Total
Arduino Mega 2560	\$995.34	1	\$995.34
Servo-Micro	\$109.48	2	\$218.96
Motor paso a paso	\$60.00	2	\$120.00
Sensor de Ultrasonido	\$105.27	1	\$105.27
Final de carrera	\$20.00	4	\$80.00
Shield Sensores	\$180.00	1	\$180.00
Sensor de temperatura LM35	\$22.46	1	\$22.46
Módulo L298N	\$75.00	2	\$150.00
Fuente conmutada	\$600.00	1	\$600.00
Banda dentada	\$40.00	3	\$120.00
Poleas	\$20.00	2	\$40.00
Valeros	\$50.00	4	\$200.00
Otros	\$200.00	1	\$200.00
			Costo Total
			\$3,032.03

*Tabla 4 Costo de desarrollo*

Fase de definición de los requerimientos del sistema

Se requiere principalmente automatizar la inyección a tratamientos de cultivos, de acuerdo a los tiempos previamente establecidos.

Para este prototipo los requerimientos principales son conseguir en movimiento en horizontal y vertical, el movimiento horizontal para colocarse sobre el objeto a analizar y el movimiento vertical para poder colocarse a la altura adecuada para poder realizar los procedimientos necesarios.

Posteriormente se requiere implementar la posibilidad de modificar automáticamente los parámetros con los que trabaja el prototipo para así poder cambiar fácilmente las dimensiones de trabajo.

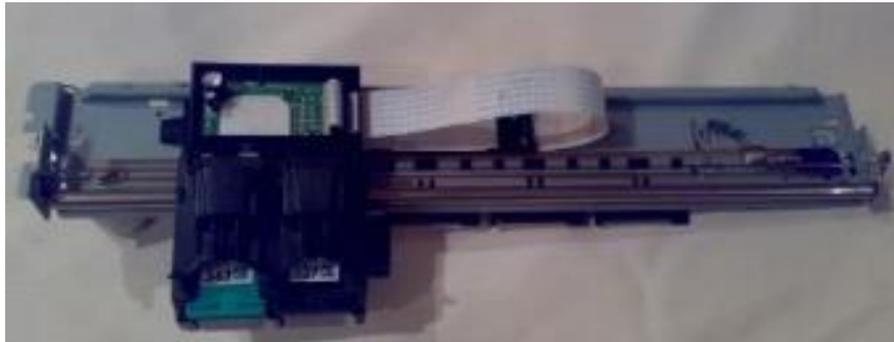
Finalmente hacer a el prototipo portable, debido a que se busca cubrir una área de trabajo considerablemente grande el prototipo tendrá dimensiones difíciles de transportar, para solucionar este problema este debe ser fácil de armar y desarmar para poder trasladarlo fácilmente.

Fase de diseño técnico

En esta fase se presenta la manera en la cual se fue desarrollando el prototipo paso a paso

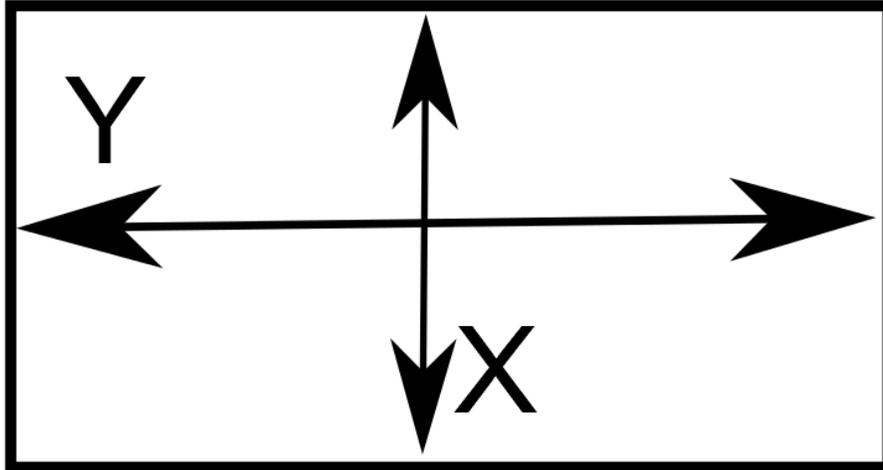
Movimiento

El primer prototipo que fue implementado, consta simplemente una adaptación de un carro de impresora como el que se muestra en la Figura 21, para obtener un sencillo movimiento en una sola dirección.



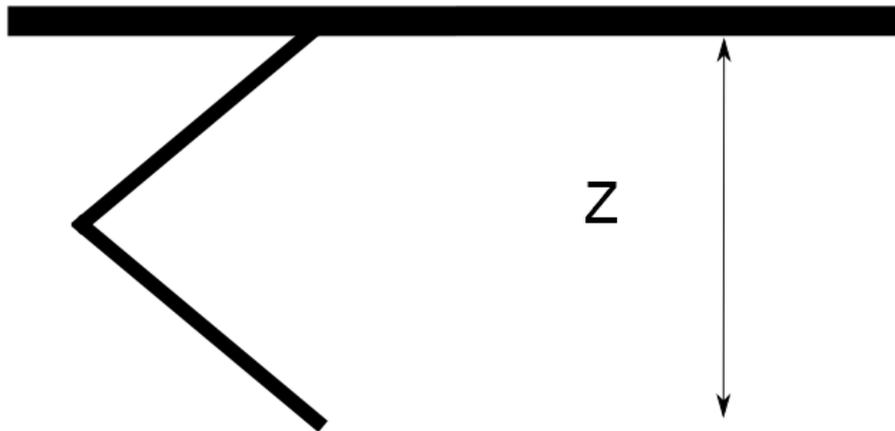
*Figura 21 Primer Prototipo*

Con investigaciones posteriores se llega al concepto de dividir el movimiento básicamente en dos partes, el primero puede ser denominado como Movimiento Horizontal, como el que se muestra en la Figura 22 visto desde un ángulo superior.



*Figura 22 Movimiento Horizontal*

El segundo movimiento será definido como Movimiento Vertical, como se muestra en la Figura 23 desde una vista lateral.



*Figura 23 Movimiento Vertical*

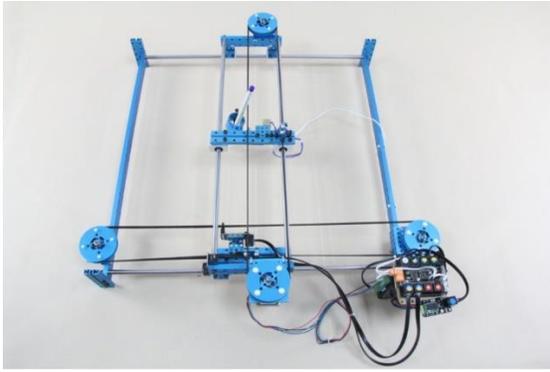
*Movimiento Horizontal*

Al igual que el primer prototipo, los posteriores conservaron el mismo mecanismo de movimiento basado en polea y banda dentada como las que se muestran en la Figura 24.

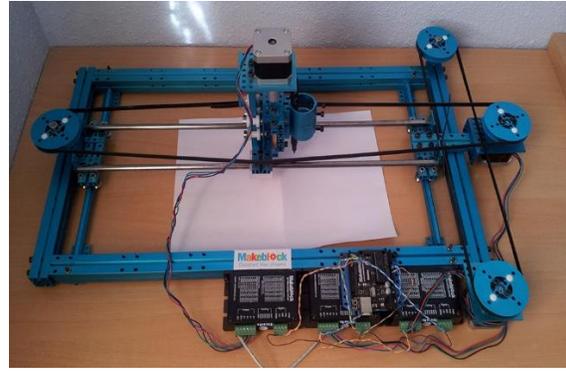


*Figura 24 Polea y Banda Dentada*

Con el fin de obtener un movimiento en los ejes “X” y “Y”, se comenzó una búsqueda de prototipos que ofrecieran este tipo de movimientos, para ser tomados como referencia para la creación del presente. Los principales que fueron tomados en cuenta están basados en un kit de la marca MAKEBLOCK, ambos son prototipos diseñados para hacer dibujos, cada uno diseñado por personas diferentes y se muestran a continuación.



*Figura 25 Makeblock XY-Plotter Robot (Makeblock, 2015)*



*Figura 26 Robot que dibuja (Makeblock, 2015)*

Basándose en estos 2 modelos se comenzó a desarrollar mis primeros prototipos, para el desarrollo de estos fue necesario buscar algún tipo de materia de fácil manipulación, tras una exhaustiva búsqueda se encontró con un perfil lineal que cumplía con estas características, este se muestra a continuación en la Figura 27. Este material es distribuido por una empresa extranjera llamada OpenBuilds.



*Figura 27 V-Slot*

Este tipo de perfil es de fácil manipulación y ayuda a crear estructuras firmes y fáciles de armar sin necesidad de taladrar o soldar, simplemente con ayuda de ángulos y tornillos. Con este material se pueden hacer estructuras como las que se muestran enseguida.

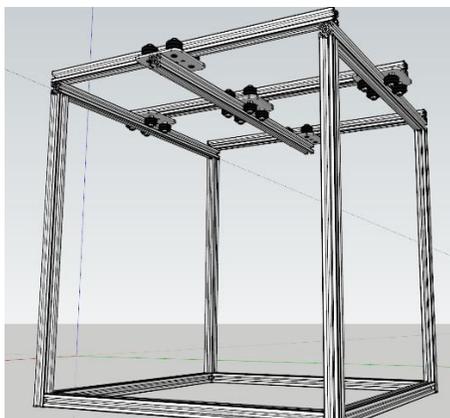


Figura 28 Ejemplo 1 V-Slot (OPENBUILS, 2015)



Figura 29 Ejemplo 2 V-Slot (OPENBUILS, 2015)

Una vez que se obtuvo el perfil V-Slot, este fue utilizado para crear el eje de movimiento más largo que en este caso es el eje “Y”, y para conseguir el movimiento en el eje “X” se ajustó el carro de la impresora de tal manera que se pudiera conseguir el movimiento, este diseño se muestra en la siguiente figura.

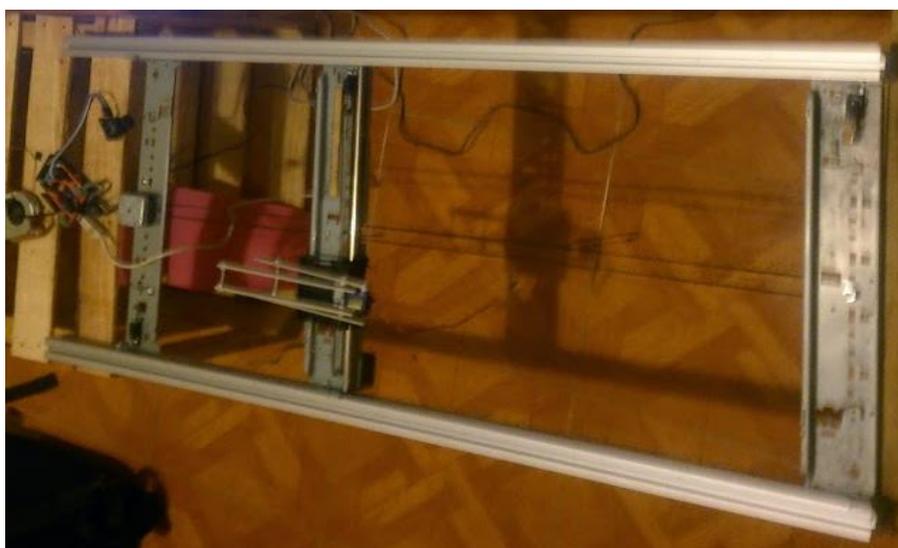


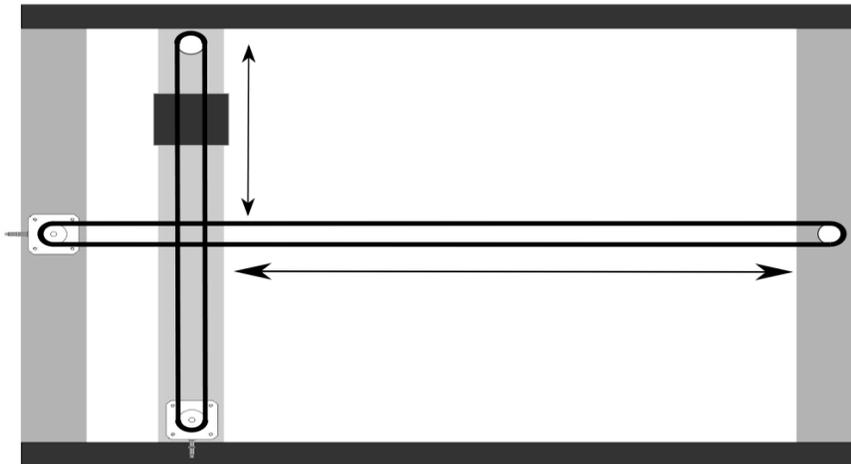
Figura 30 Primer Prototipo con movimiento X Y

Con este prototipo se logró un movimiento forzado en el eje “Y” y un movimiento suave en el eje “X” lo cual causaba inconformidad ya que se forzaba mucho tanto el Driver de control como el motor del eje “Y”, por esta razón se buscó reducir el peso del eje y cambiando la base de acero por 2 varillas más ligeras y con esto obtener un movimiento más suave en el eje “Y” objetivo que fue alcanzado.



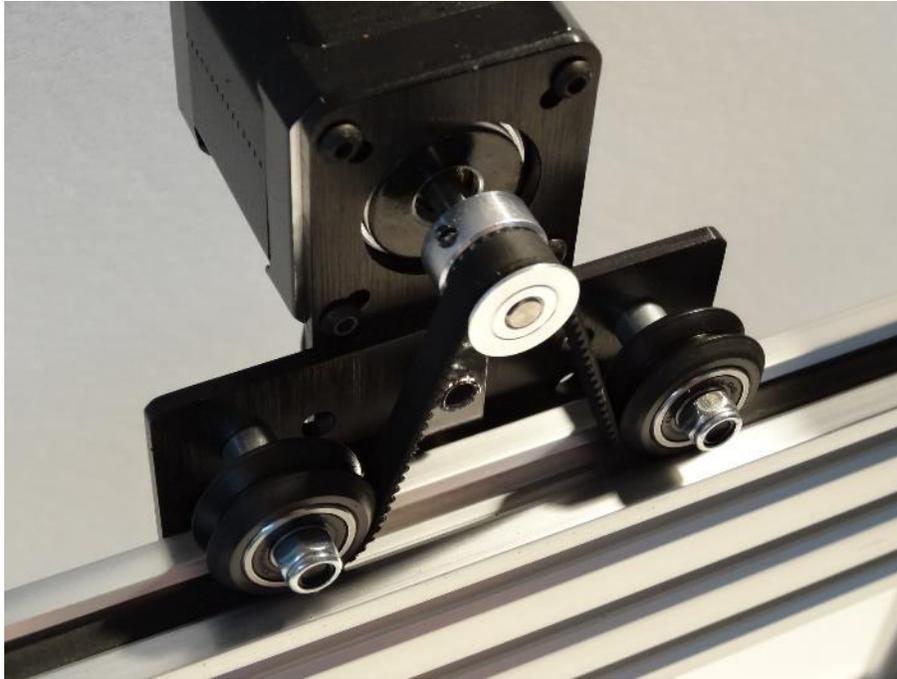
*Figura 31 Segundo Prototipo con movimiento X Y*

Tanto en el primer como el segundo prototipo el sistema de bandas es el mismo, este sistema se muestra en la Figura 32, el inconveniente encontrado en esta ocasión es que la dimensión de cada banda debe de ser el doble de la distancia recorrida, el problema era que no se encontró una banda de ese tamaño a un precio considerable.



*Figura 32 Diseño básico de bandas*

Por el motivo antes mencionado, fue necesario buscar una alternativa diferente, y se recurrió a las galería de imágenes con las que cuenta OpenBuilds, entre uno de los proyectos que presenta se encontró una adaptación de la banda y polea que compila con los nuevos requerimientos, esta nueva forma de acomodar los componentes se muestra en la Figura 33.



*Figura 33 Adaptación polea (OPENBUILS, 2015)*

Con base al nuevo modelo a seguir se llega a la conclusión de que es posible implementar los 2 motores en una sola placa para así reducir el cableado suelto que se tenía en los prototipos anteriores, igualmente con este sería posible reducir la longitud de la banda del eje “Y” y así reducir el costo, para esto se logró la creación de una base como la que se muestra a continuación.



*Figura 34 Primer prototipo 2 motores integrados*

Ya que con la implementación de esta nueva base se lograría cargar todo el peso sobre un solo perfil, se llega a la conclusión que el soporte del eje “Y” se reduciría a solo un perfil y así reducir el peso de todo el prototipo en general. En este punto ya está resuelto lo forma en que se obtendrá el desplazamiento en el eje “Y”, ahora solo faltaría definir como sería la estructura para el desplazamiento en el eje “X”, para ello se realiza un primer intento con un nuevo prototipo como el que se muestra en seguida.



*Figura 35 Propuesta para eje X*

Nuevamente se logra solo un movimiento suave en un eje, pero en el otro por el desequilibrio del peso se obtiene un movimiento lento y forzado, por esta razón este

prototipo es descartado y nuevamente se comienza con la búsqueda de un nuevo método dando paso al siguiente y último prototipo.



*Figura 36 Prototipo Mecánico final*

Una vez que se ha llegado a este punto se ha obtenido un prototipo con las características deseadas, que son movimiento suave y facilidad de armar y desarmar.

#### *Movimiento Vertical*

El movimiento horizontal a pesar de ser en 2 direcciones es básicamente logrado con el mismo mecanismo como ya fue demostrado, ahora es turno del Movimiento Vertical

Este será conseguido con la ayuda del HC-SR04 Sensor Ultrasónico y 2 Servomotores, estos tres elementos fueron integrados con el propósito de obtener un pequeño brazo como el que se muestra en la Figura 37 , al igual que se mostrara la forma en que se obtuvo los movimientos necesarios para obtener la posición deseada.

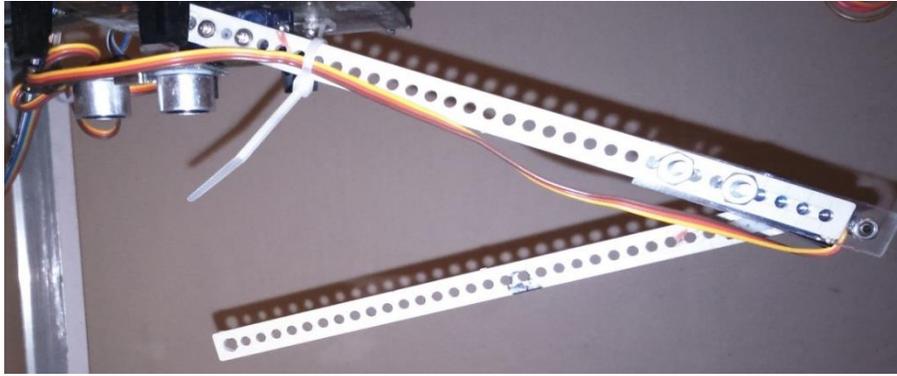


Figura 37 Brazo robótico

El brazo robótico consta de 2 tiras de plástico con una medida de 16 cm de largo, esta medida será considerada como la hipotenusa en el modelo matemático para obtener el ángulo en el cual se deben de mover ambos servomotores, este modelo es descrito en la siguiente imagen.

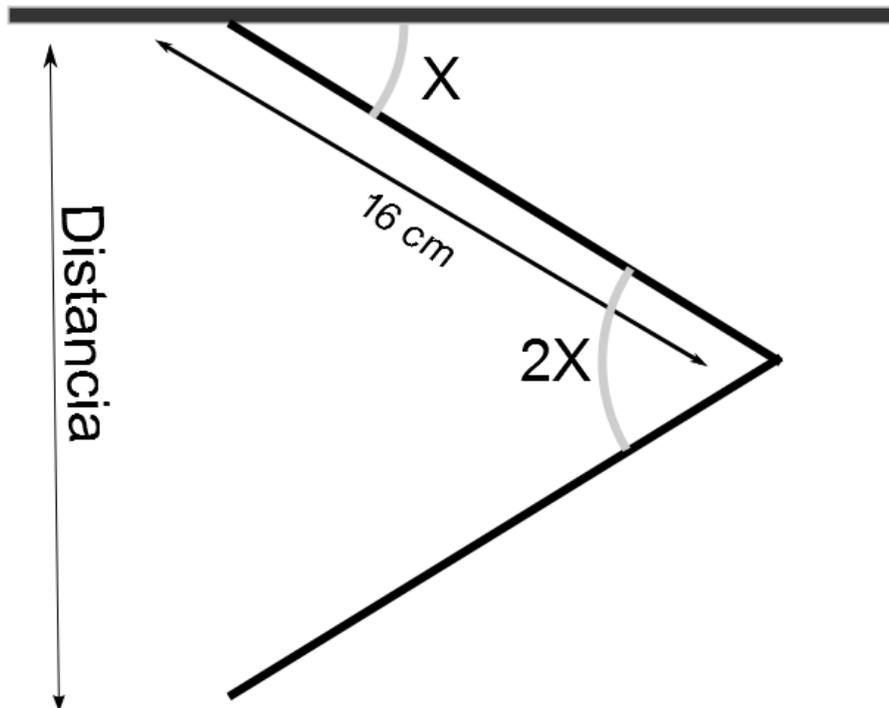


Figura 38 Modelo matemático

En primera instancia el primer paso es obtener la distancia de que se tiene del objeto al prototipo con ayuda del sensor ultrasónico, este proceso es ilustrado en la imagen siguiente.

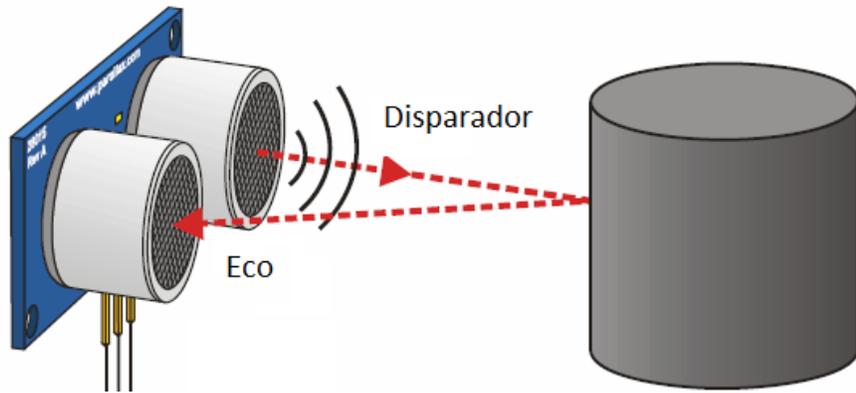


Figura 39 Lectura de distancia

Posteriormente se pasa a tratar este dato con la función  $\text{sen } \theta = \frac{C.O.}{Hip.}$  sustituyendo los valores y despejando quedando una ecuación como  $\theta = \text{sen}^{-1} \frac{C.O.}{Hip.}$

Ya que Arduino opera con Radianes y los servomotores necesitan como parámetro de entrada un valor en grados es necesario realizar la conversión del valor obtenido para que pueda ser interpretado correctamente por los servomotores.

$$\text{angulo} = \theta \left( \frac{\pi}{180} \right)$$

El valor obtenido es el que se manda como parámetro a los servomotores y por este medio es como se obtiene el movimiento exacto para acercarse al objeto sin tocarlo.

#### Conexiones Eléctricas

Posteriormente del armado se debe de realizar las conexiones necesarias para el funcionamiento del prototipo, a continuación se muestran la forma de realizar las conexiones de la Figura 40 a la Figura 44.

Comenzando con las conexiones de los motores paso a paso, estos requieren de 5 salidas digitales, 4 que controlan el giro y velocidad, y el ultimo es utilizado para activar o desactivar el motor en cuestión.

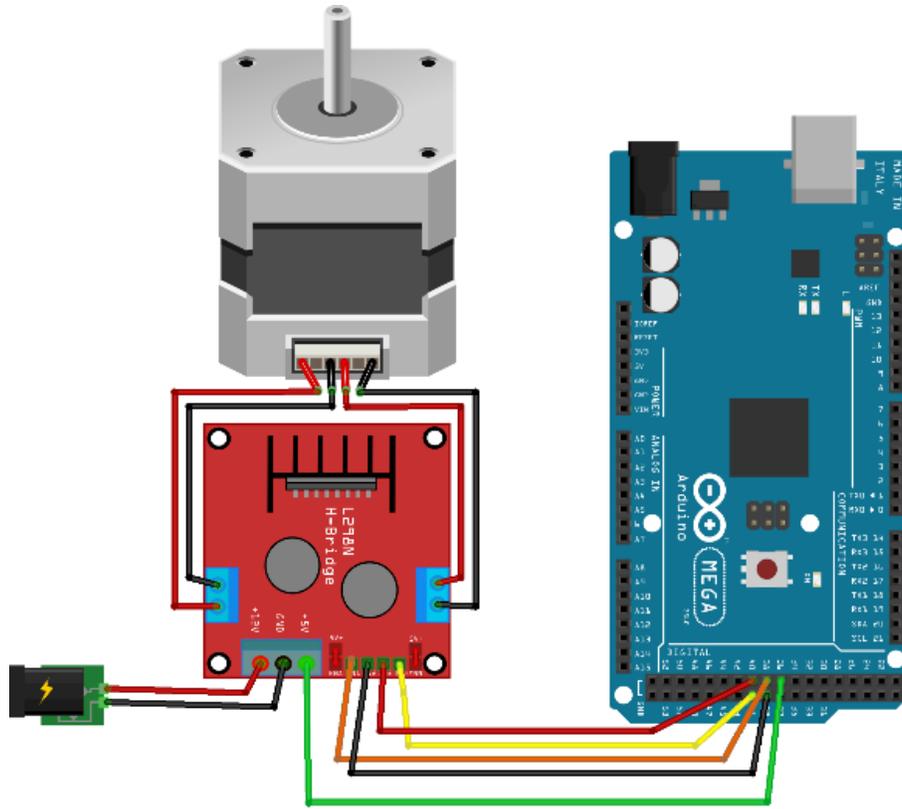


Figura 40 Conexión Motor paso a paso

Una vez realizada esta conexión es necesario delimitar el área de trajo, por esta razón es que se implementa la conexión de los finales de carrera, estos requieren de una conexión tierra y corriente con su salida digital para leer si el sensor está activo o no.

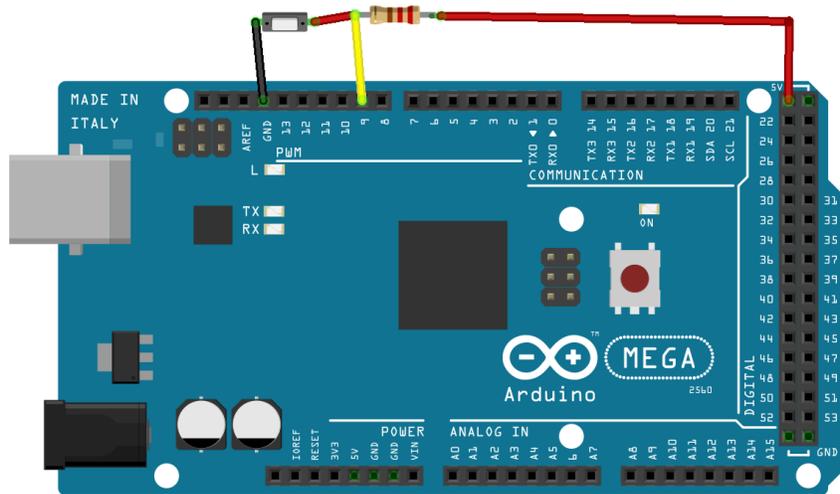


Figura 41 Conexión final de carrera

Para poder monitorear la temperatura de los puentes H se integra al prototipo un sensor LM35, este sensor tiene una salida analógica y su correspondiente alimentación.

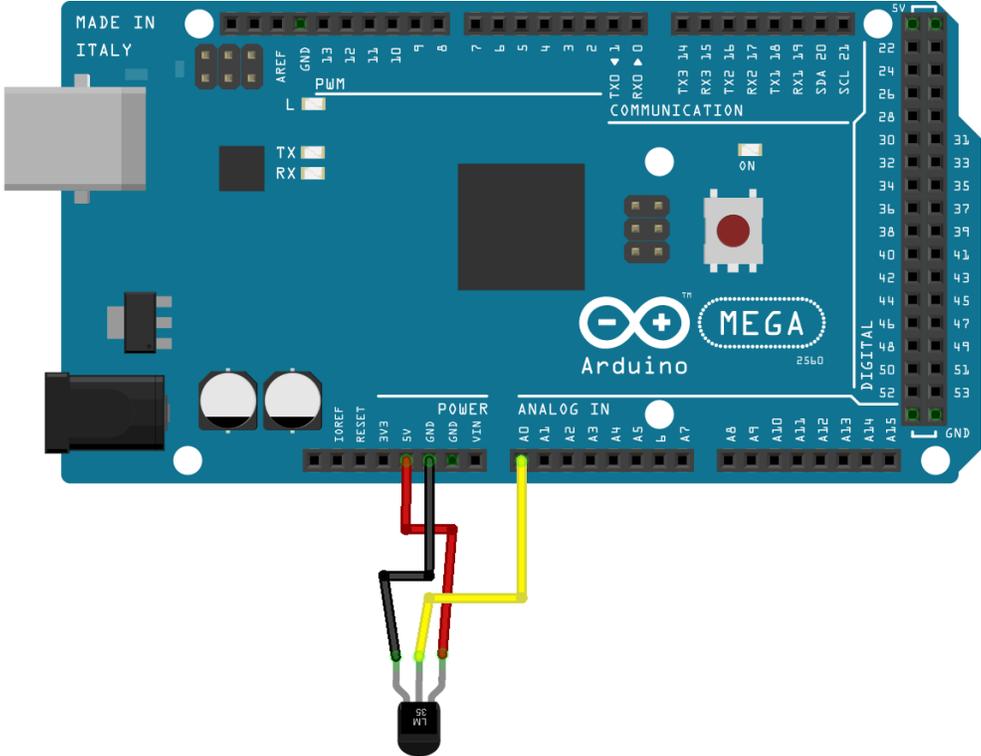


Figura 42 Conexión LM35

Los servomotores tienen una conexión relativamente fácil ya que solo cuentan con 3 cables, entrada de voltaje, tierra y señal, esta última manejada como una señal analógica.

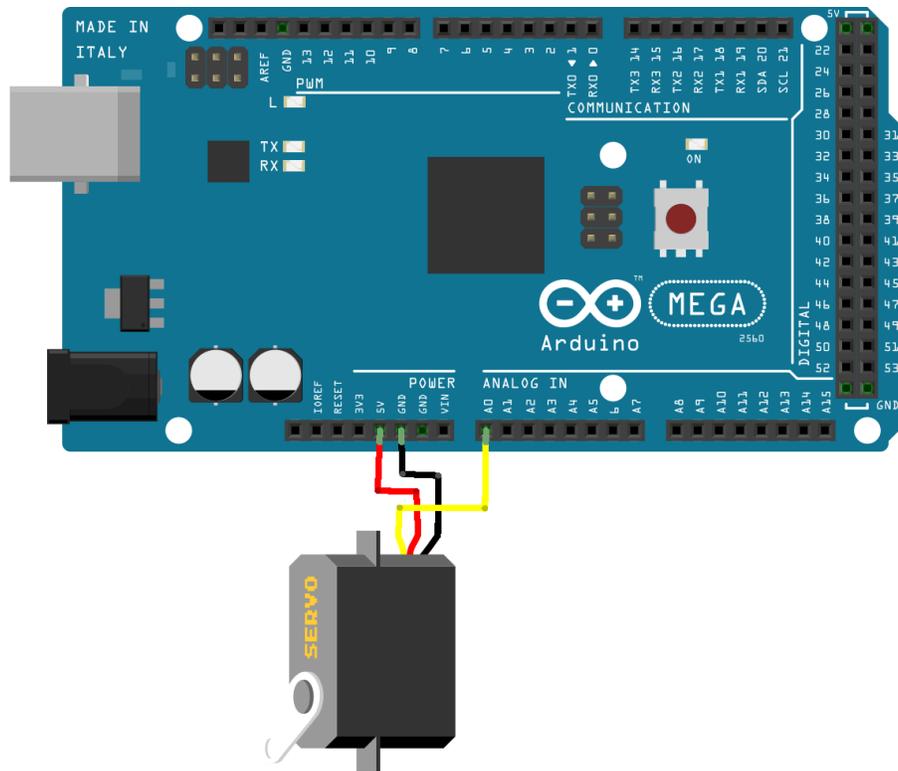


Figura 43 Conexión Servomotor

Por último es necesario conectar el sensor HC-SR04 que ayudara a obtener la distancia a la que se encuentra el objeto, este sensor tiene 1 salida digital (trig) y una entrada igualmente digital (echo).

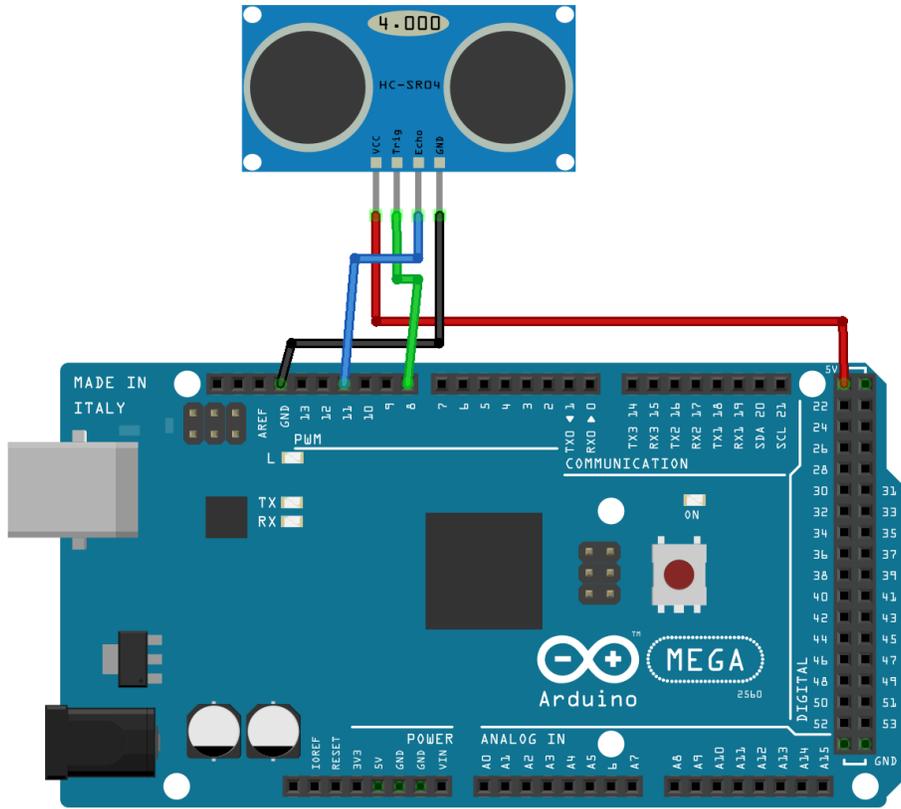


Figura 44 Conexión Ultrasónico

Una vez que se ha mostrado las conexiones independientes de cada dispositivo, a continuación se mostrara la conexión en su totalidad de todos los componentes.

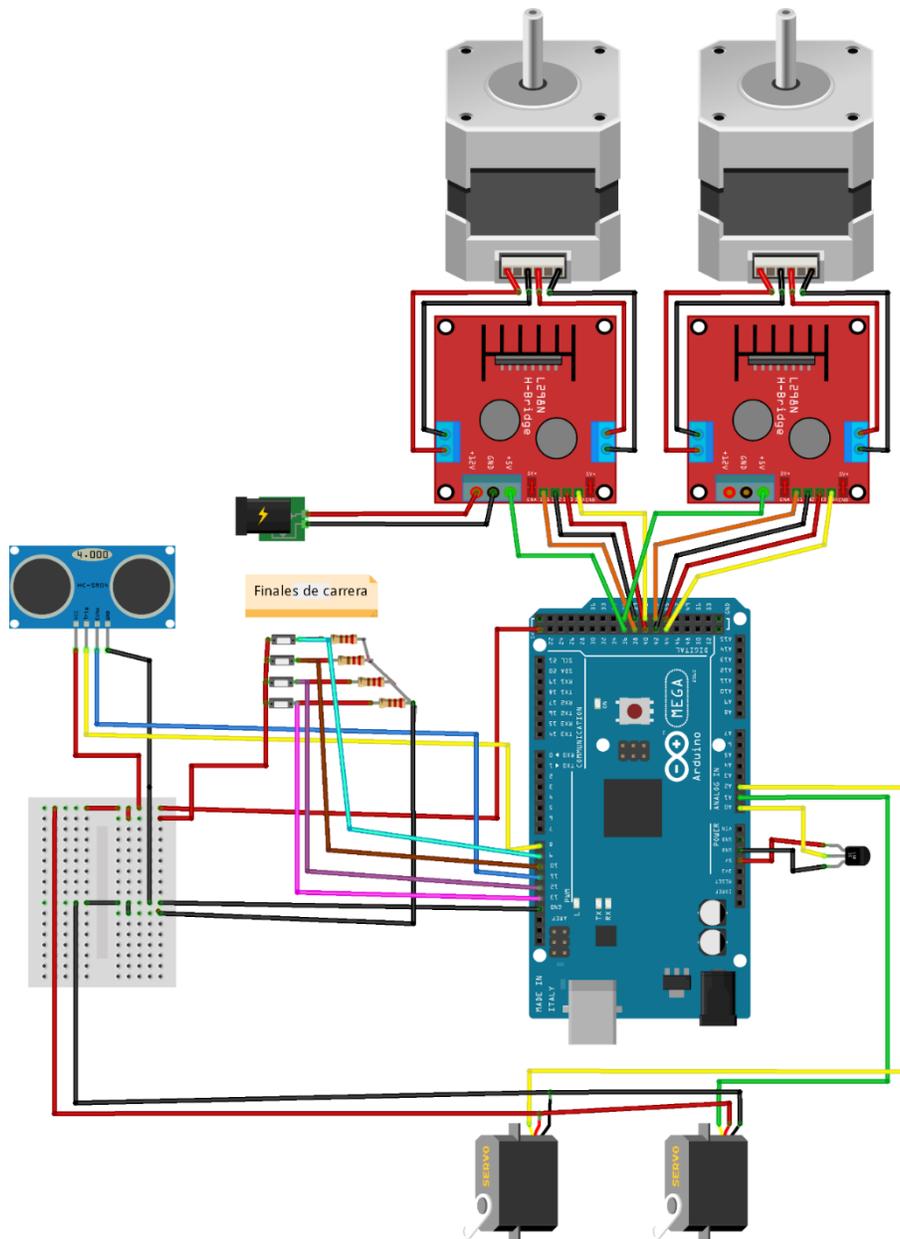


Figura 45 Conexión del Prototipo Completo

## Rutinas de Software

De la misma forma en que es importante la construcción del prototipo físico, las rutinas de ejecución también juegan un papel importante en el desarrollo de cualquier tipo de prototipo.

A continuación se muestra el desarrollo de las rutinas que son necesarias para el funcionamiento del prototipo, comenzando con el diagrama de flujo general, que se muestra en la Figura 46.

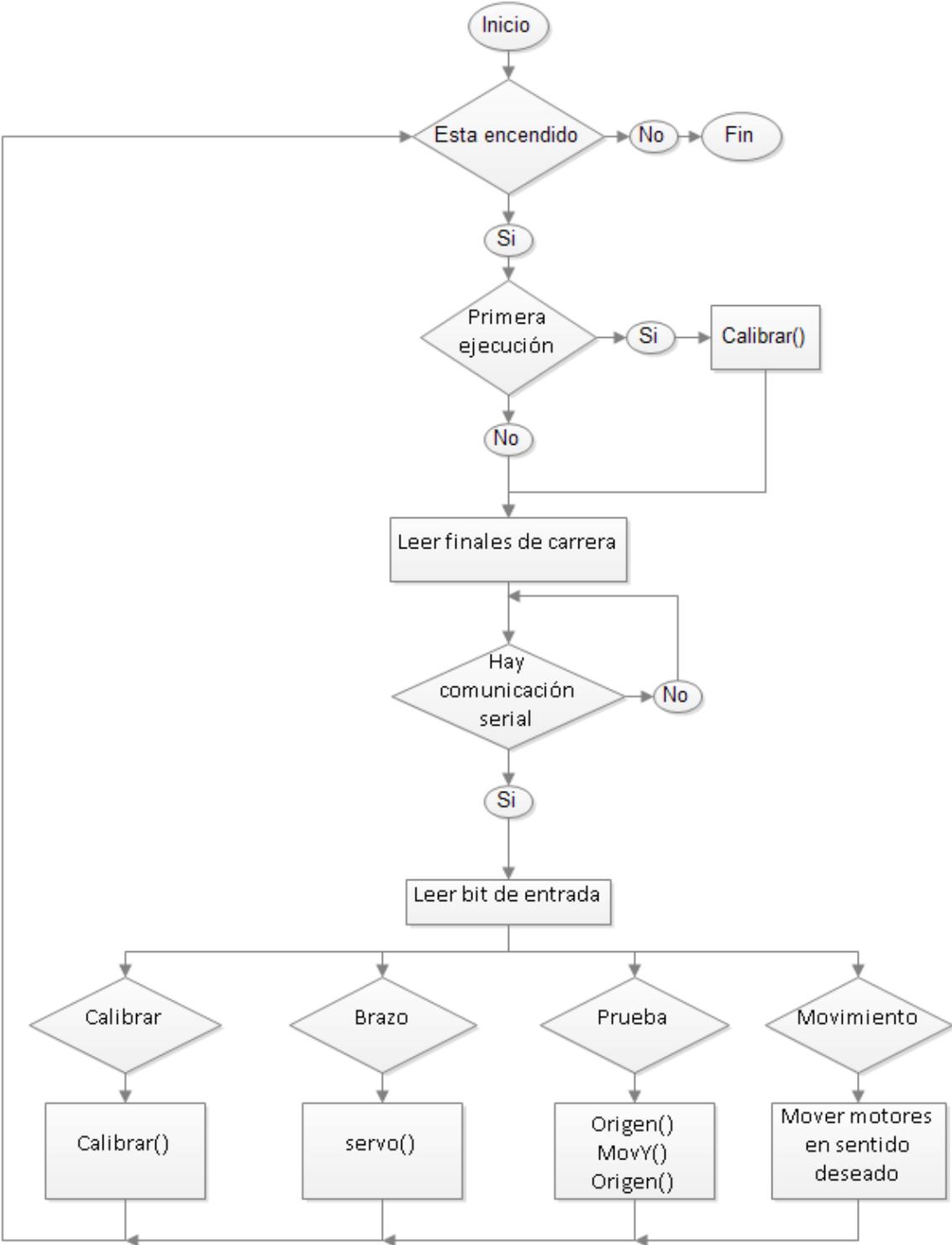
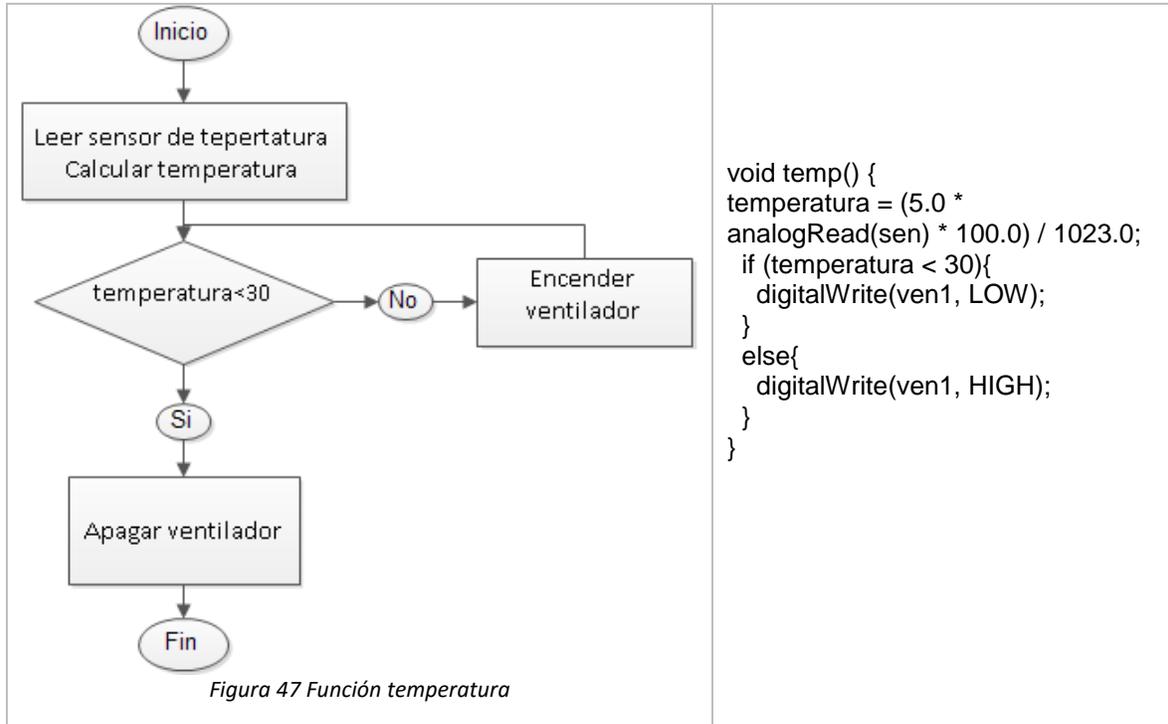
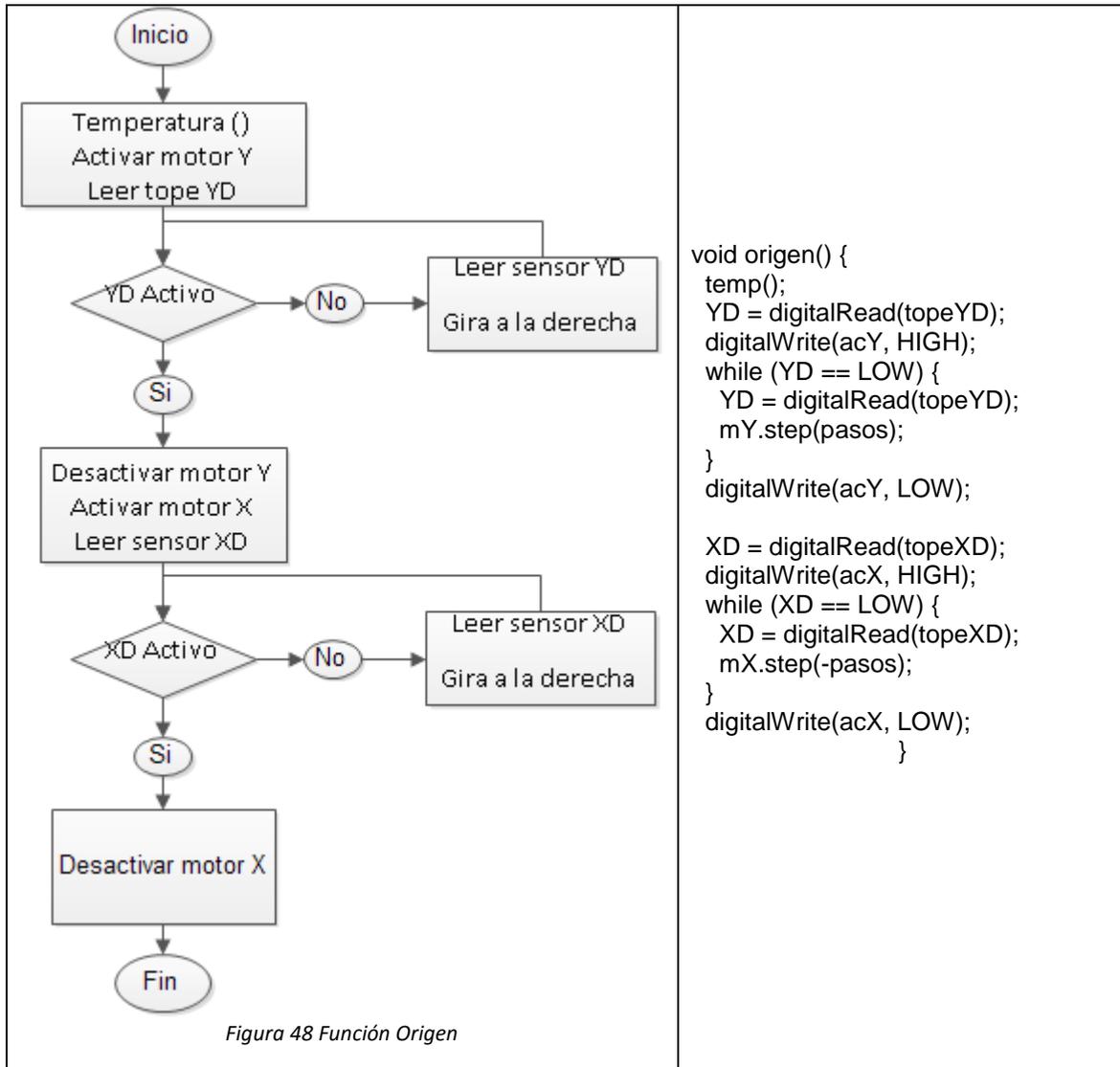


Figura 46 Diagrama de flujo general

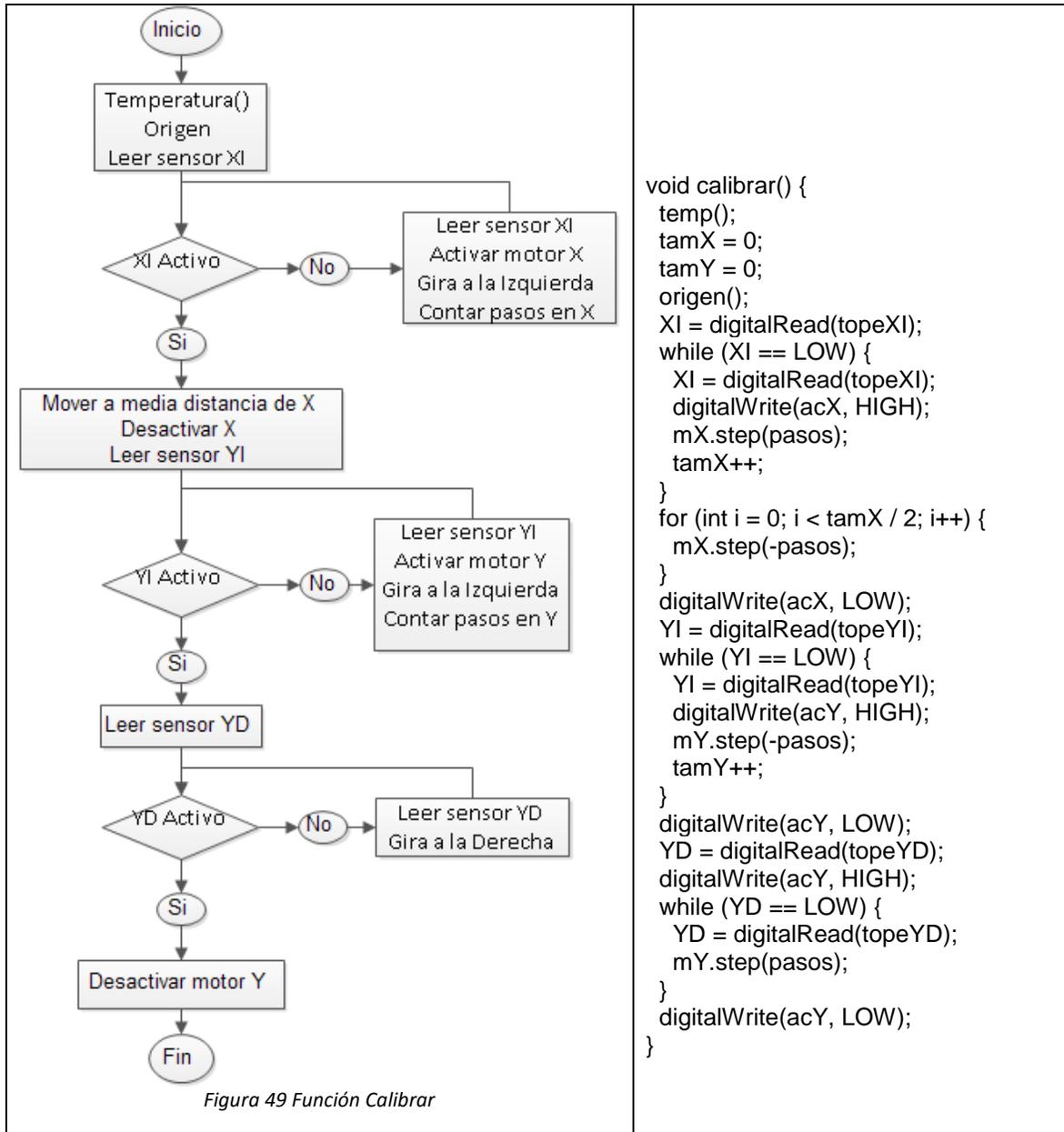
Uno de los elementos importantes a cuidar siempre ha sido el calentamiento por ello es necesario monitorear constantemente este parámetro, para esto se implementa la siguiente función.



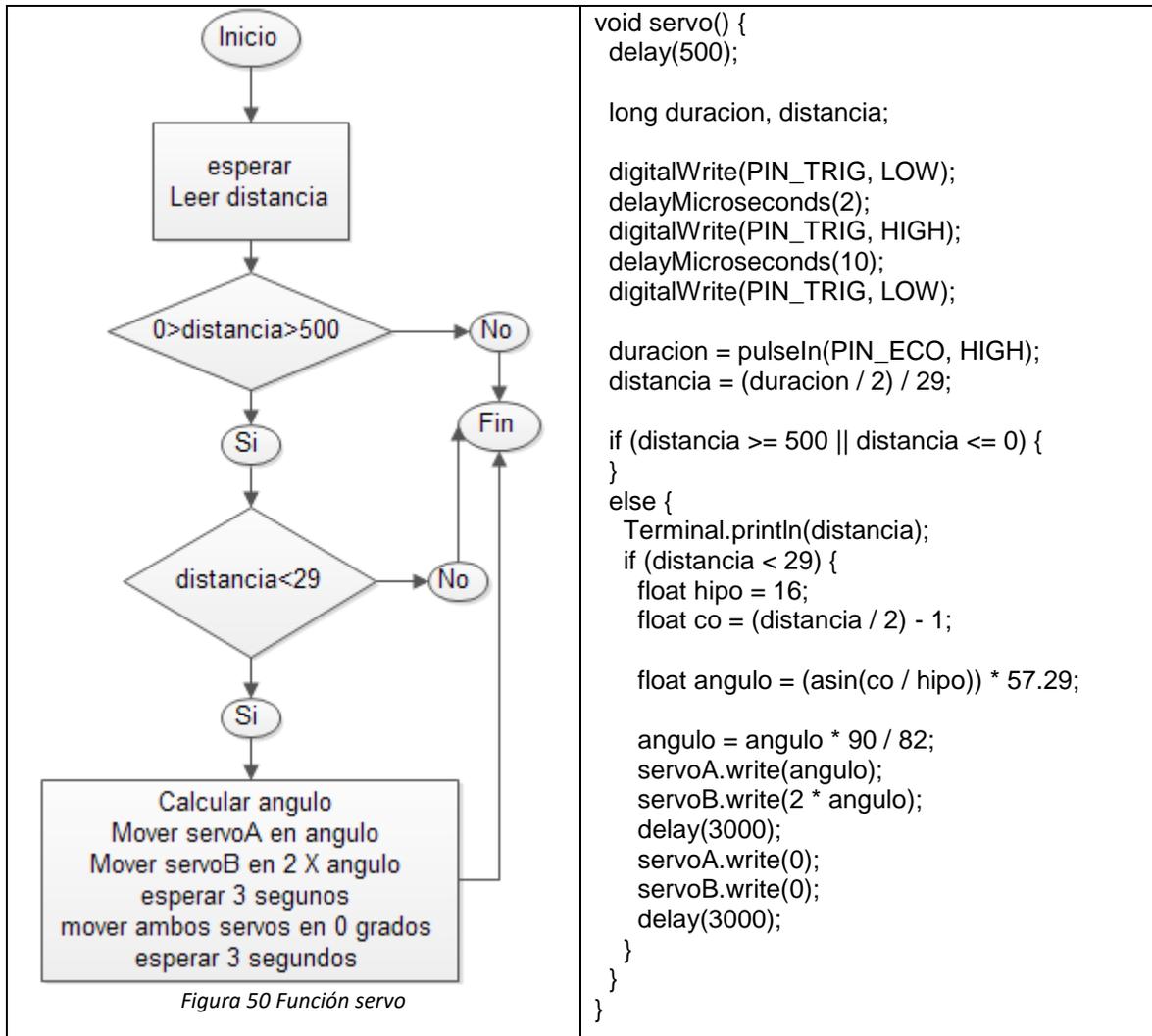
Posteriormente siempre es importante tener un punto de referencia para comenzar cualquier acción, para esto se creó la función origen.



El siguiente parámetro importante es el área de trabajo, este prototipo es capaz de adaptarse a cualquier dimensión, para esto se genera una función que hace un barrido completo del área de trabajo y la almacena para trabajar con estos parámetros posteriormente.



Al igual que es importante el movimiento horizontal es importante el movimiento vertical y para realizar este último, para esto se ocupa la siguiente función.



Por ultimo las funciones para realizar el movimiento horizontal son representadas por las siguientes figuras.

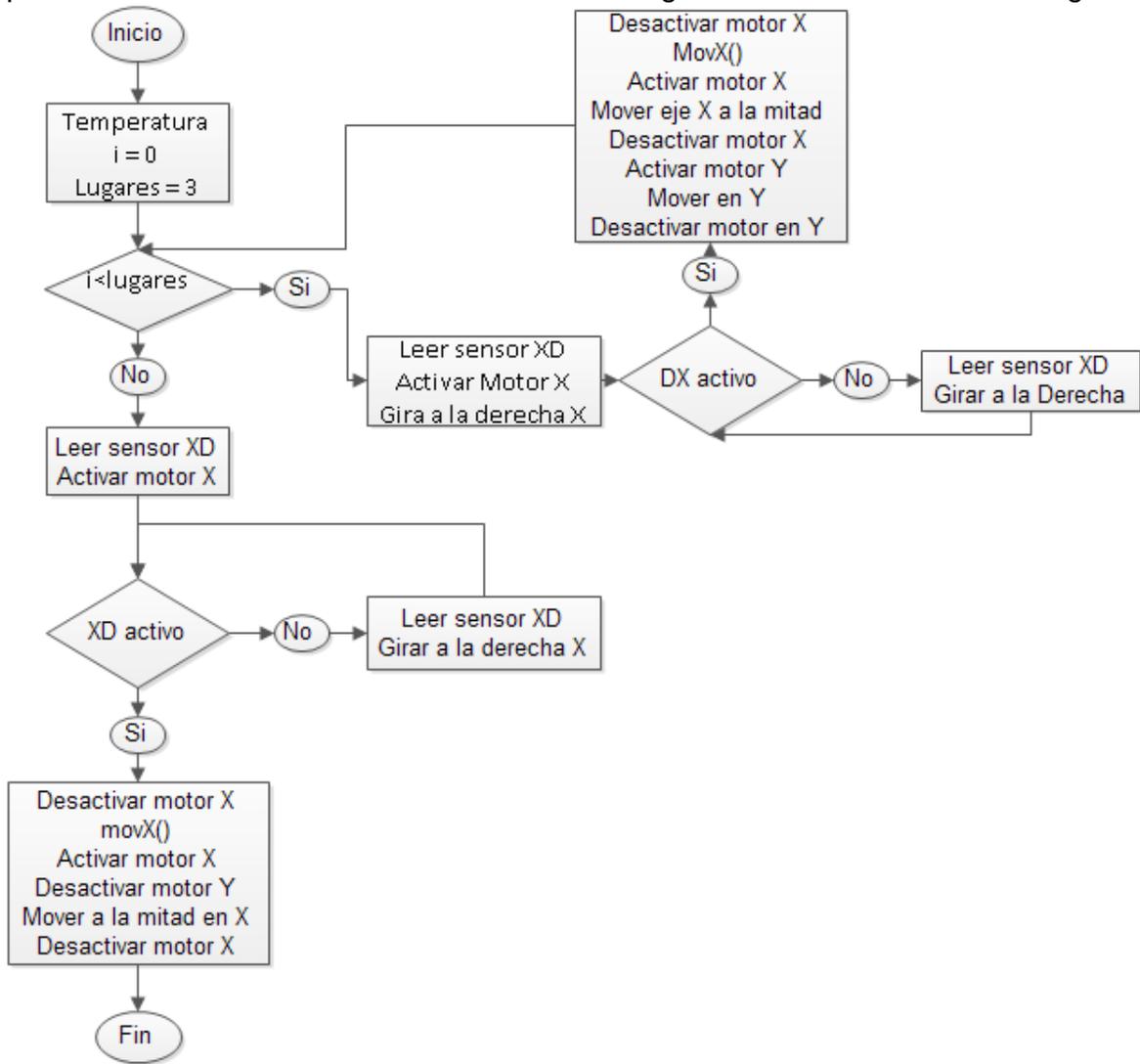


Figura 51 Función movY

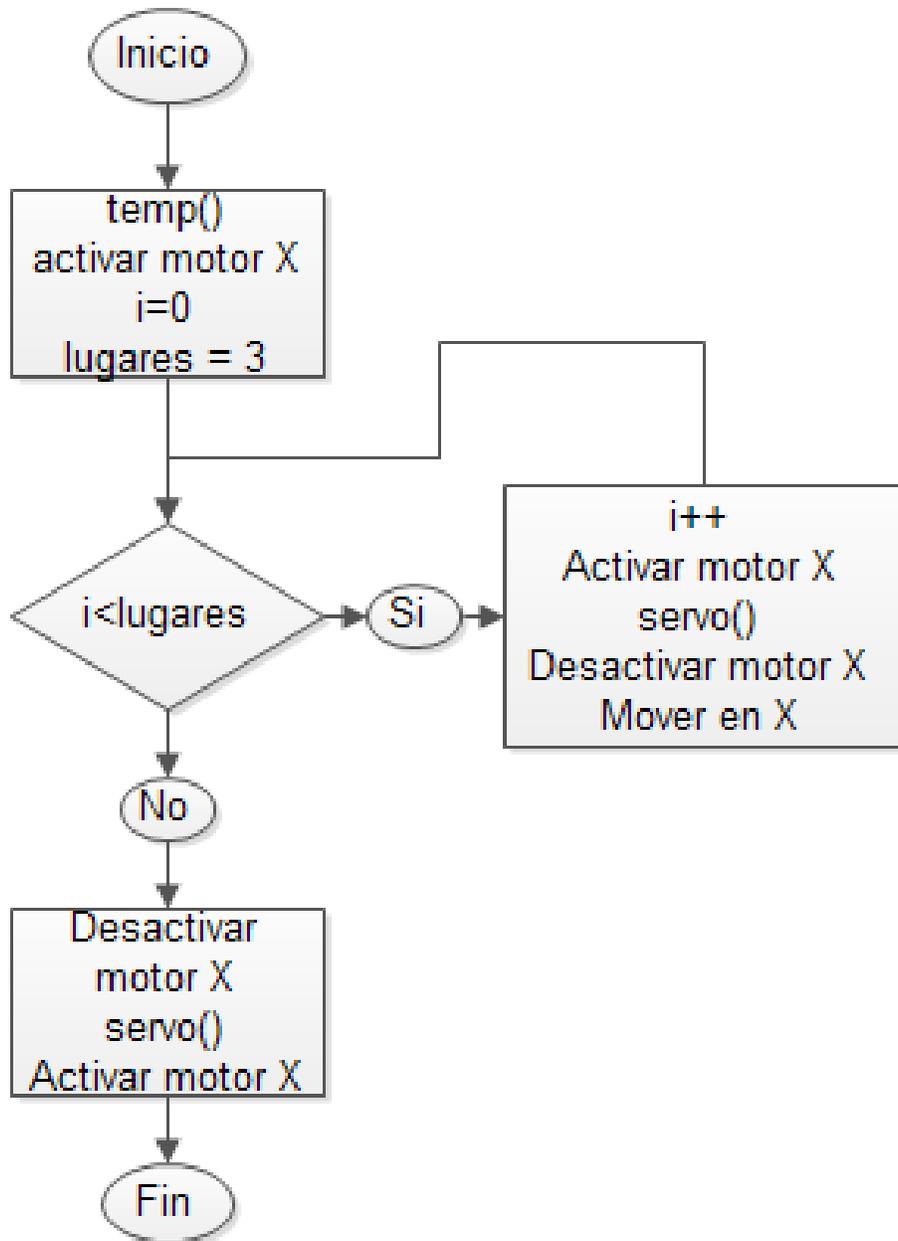


Figura 52 Función movX

Fase Pruebas.

Para concluir es necesario realizar las pruebas prudentes para verificar si los objetivos han sido alcanzados o no, para esto se divide el procedimiento en pequeñas pruebas que se definen en las siguientes pruebas.

Pruebas de movimiento

Debido a que el primer objetivo específico es obtener movimiento en 3 ejes es necesario verificar si este ha sido cumplido en su totalidad, para esto se realizaron las siguientes pruebas.

Objetivo	Resultado	Solución
Verificar movimiento horizontal. (Eje "Y").	El eje "Y" no cuenta con la fuerza suficiente para realizar el movimiento.	Disminuir la velocidad de movimiento al motor del eje "Y".
Verificar movimiento Horizontal. (Eje "X").	Es posible aumentar la velocidad de movimiento para reducir tiempos de ejecución.	Aumentar velocidad de movimiento al motor del eje "X".
Verificar movimiento en dirección correcta.	El eje "Y" realiza los movimientos en dirección opuesta, mientras que el eje "X" realiza el movimiento correcto.	Se invirtieron la forma de conexión de los cables de motor del eje "Y".
Prueba con finales de carrera.	Los finales de carrera fueron colocados en el lugar exacto y estos detectan a la perfección las dimensiones con las que se deberá de trabajar.	
Forzar movimiento una vez que se ha encontrado un final de carrera	El carro de movimiento intenta seguir moviéndose a pesar de que ya no cuenta con más espacio.	Implementación de una condicional, en la cual es imposible mover el carro una vez que ha llegado a un final de carrera.
Calentamiento de los motores.	Los motores sufren de calentamiento excesivo.	El calentamiento excesivo se debe a que estos se encuentran activos en todo momento, por tanto se opta por desactivarlos cuando no se requiere de movimiento.
Verificar movimiento con servomotores.	El movimiento es el adecuado.	

Tabla 5 Pruebas de movimiento

El siguiente objetivo a evaluar es implementar los sistemas de detección de objetos mediante un sensor ultrasónico.

Objetivo	Resultado	Solución
Definir si las dimensiones obtenidas son correctas.	Las medidas obtenidas varían por diferencia de 1 centímetro.	Modificación de la rutina de lectura de dimensiones, disminuyendo el resultado en un centímetro para que exista coherencia de resultados.
Verificar si los ángulos obtenidos son correctos.	Los resultados son los deseados.	
Detección de objetos de variada altura	El prototipo es capaz de detectar si existe un objeto o no, siempre y cuando los objetos tengan un rango de altura de entre 1 y 28 centímetros.	

*Tabla 6 Pruebas Sistema de detección de objetos*

Por último verificar si las rutinas creadas son capaces de adaptarse a diferentes áreas de trabajo.

Objetivo	Resultado	Solución
Prueba sin haber hecho una calibración previa.	El prototipo no realiza un funcionamiento correcto.	Modificar la rutina de arranque para que en toda ocasión que sea conectado lo primero a realizar en la detección de área a trabajar.
Modificación de dimensiones	El prototipo fue capaz de modificar sus propiedades para trabajar con las nuevas dimensiones asignadas.	

*Tabla 7 Pruebas en Rutinas*

## Conclusiones

- La construcción del prototipo tiene un costo aproximado de \$3000, ya que al utilizar software libre y hardware de plataforma abierta, se reducen considerablemente los costos.
- El prototipo que se desarrolló aun no es capaz de aplicar tratamientos debido a que el proyecto de investigación se limitó al desarrollo de la plataforma necesaria para generar el movimiento, y sólo es necesario hacer pequeñas adaptaciones para adecuarse a la aplicación de cualquier tratamiento que se requiera.
- Una de las mejoras que pueden ser implementadas al prototipo es la conexión a una base de datos para así poder procesar la información recolectada por los sensores.
- El control se llevó a cabo mediante la integración sensores y rutinas se Software que en conjunción logran la ejecución de los movimientos deseados y así el investigador no requiere estar presente para la aplicación de los tratamientos y sólo se dedicará a analizar los resultados obtenidos.
- El funcionamiento de los motores pasó a paso, requieren de un amperaje alto para su buen funcionamiento por lo que es necesario tener una fuente regulada de alimentación eléctrica; el no contar con este elemento, provocará la sobrecarga de los motores e inclusive puede provocar que queden inservibles.
- Es posible adaptar el prototipo a otras dimensiones según el espacio de trabajo con el que se cuente, ya que por sí solo al ser encendido realizará un barrido de su área de trabajo que se le asigne y guarda estos valores para posteriormente trabajar con base a ellos.

## Bibliografía

- 1Sheeld*. (2015). Obtenido de <http://1sheeld.com/>
- Allier, A. M. (2006). *Física III*. México: Universidad Nacional Autónoma de México.
- Arduino*. (2015). Obtenido de <https://www.arduino.cc/en/Main/Products>
- ArduinoInfo*. (2015). Obtenido de SENSOR SHIELD: <https://arduino-info.wikispaces.com/SensorShield>
- Automatizacion Industrial*. (s.f.). Obtenido de Introducción a la automatización 1: [http://www.uhu.es/diego.lopez/AI/auto\\_trans-tema1.pdf](http://www.uhu.es/diego.lopez/AI/auto_trans-tema1.pdf)
- Banzi, M. (2011). *Getting Started with Arduino Second Edition*. O'RAILLY.
- Campos, O. (2000). *Matemáticas 3 Geometría y trigonometría*. México: Publicaciones Cultural.
- Campos, P. A. (2008). *Sensores analógicos utilizados en la Automatizacion Industrila*.
- Caparrós, J. A. (14 de julio de 2014). *RepRap*. Obtenido de ramps 1.4: [http://www.reprap.org/wiki/Clone\\_wars:\\_Ronqu%C3%ADO/es](http://www.reprap.org/wiki/Clone_wars:_Ronqu%C3%ADO/es)
- Cruz Bautista, F. (2007). *Colegio de Postgraduados*. Obtenido de Automatización de un sistema de fertirrigación por goteo: <http://hdl.handle.net/10521/1249>
- DFRobot*. (2015). Obtenido de Makeblock XY-Plotter Robot Kit (With Electronics): [http://www.dfrobot.com/index.php?route=product/product&product\\_id=1051#.VcAgjfn09no](http://www.dfrobot.com/index.php?route=product/product&product_id=1051#.VcAgjfn09no)
- ELECTRONILAB Ingeniería y Diseño Electrónico*. (2014). Obtenido de Tutorial: Uso de DRIver L298N para motores DC y paso a paso con arduino: <http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>
- Groppa, M. (2014). Riego por goteo. *Revista Palmas*, 35.
- Hermisan. (2015). *Hermisan*. Obtenido de Programador ACTIVA: <http://www.hermisan.com/w/58/programador-activa>
- Inteligencia artificial*. (2015). Obtenido de <http://www.inteligenciaartificialyrobotica.com/esp/item/127/sensor-ultrasonico>
- ISA*. (2014). Obtenido de sensores: <http://www.isa.cie.uva.es/~maria/sensores.pdf>
- ITEAD*. (2015). Obtenido de ITEAD Dual Step Motor Driver Shield: <http://imall.itead.cc/im120417015.html>
- Kalpakjian. (2002). *Manufactura, ingeniería y tecnología*. Prentice hall.
- Lozano, C. E. (2015). Servomotores, la misma potencia con ahorro para la industria. *Metal Actual*, 35.
- Makeblock*. (2015). Obtenido de <http://www.makeblock.cc/>

*Medirtemperatura*. (2015). Obtenido de <http://medirtemperatura.com/sensor-temperatura.php>

Mohan, N., Undeland, T., & Robbins, W. (2003). *Power Electronics. Tercera Edición*. John Wiley & Sons, Inc.

*OPENBUILS*. (2015). Obtenido de <https://openbuilds.com.mx/>

Robot, T. (2015). *Todo Robot*. Obtenido de <http://www.todorobot.com.ar/tutorial-sobre-motores-paso-a-paso-stepper-motors/>

SANCHEZ V, J. (2000). *Fertirrigacion, principios, factores, aplicaciones*. Perú: FERTITEC S.A.

*Sen Sing*. (2015). Obtenido de [http://www.sensores-de-medida.es/sensing\\_sl/SENSORES-Y-TRANSDUCTORES\\_35/Sensores-de-distancia\\_36/](http://www.sensores-de-medida.es/sensing_sl/SENSORES-Y-TRANSDUCTORES_35/Sensores-de-distancia_36/)

*Tecnoroboticapantanista*. (2015). Obtenido de <http://tecnoroboticapantanista.blogspot.mx/p/sensores-final-de-carrera.html>

*TodoTobot*. (s.f.). Obtenido de Tutorial sobre Motores Paso a Paso:  
<http://www.todorobot.com.ar/tutorial-sobre-motores-paso-a-paso-stepper-motors/>

*X-robotics*. (s.f.). Obtenido de <http://www.x-robotics.com/sensores.htm>