



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL**

**CARRERA:  
INGENIERÍA EN ELECTRÓNICA**

**Tesis previa a la obtención del título de:  
INGENIERO ELECTRÓNICO**

**TEMA:**

**“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE  
ORIENTADOS A LA INSTRUMENTACIÓN FÍSICA Y  
VIRTUAL EN EL SECTOR INDUSTRIAL, MEDIANTE PLC’S  
ARDUINO”**

**AUTORES:**

**MENDEZ GOZALEZ VICTOR LUIS  
ZAMBRANO FLORES KIARA MADELAINE**

**TUTOR:**

**Ph.D. David Barrios Puerto**

**GUAYAQUIL, 2020**

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.**

Nosotros, MENDEZ GONZALEZ VICTOR LUIS y ZAMBRANO FLORES KIARA MADELAINE autorizamos a la **Universidad Politécnica Salesiana** la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro. Además, declaramos que los conceptos y los análisis desarrollados con sus conclusiones en el presente trabajo son de exclusiva responsabilidad de los autores.

Guayaquil, Diciembre del 2019



---

Méndez González Victor Luis  
C.I: 0927505685



---

Zambrano Flores Kiara Madelaine  
C.I: 0928297860

## **CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UPS**

Nosotros, **MENDEZ GONZALEZ VICTOR LUIS**, con documento de identificación N° **0927505685** y **ZAMBRANO FLORES KIARA MADELAINE** con documento de identificación N° **0928297860**, Por medio de la presente, declaramos que el proyecto técnico de titulación: **DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO PARA EL CONTROL DE PROCESOS MEDIANTE PLC'S ARDUINO**, el mismo que ha sido desarrollado para obtener el título de: **INGENIERO ELECTRÓNICO**, que se presentó ante el Consejo de la Carrera de Ingeniería Electrónica, es cedida en su totalidad los derechos de propiedad intelectual que nos corresponden, a favor de la **UNIVERSIDAD POLITÉCNICA SALESIANA**.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana



Mendez Gonzalez Victor Luis  
C.I:0927505685



Zambrano Flores Kiara Madelaine  
C.I: 0928297860

## **CERTIFICACIÓN DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.**

En calidad de Tutor del trabajo de titulación proyecto técnico “**IMPLEMENTACION DE MODULOS DE APRENDIZAJE ORIENTADOS A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR INDUSTRIAL, MEDIANTE PLC’S ARDUINO**”, presentado por el señor: **MENDEZ GONZALEZ VICTOR** y por la Señorita **ZAMBRANO FLORES KIARA MADELAINE** para optar por el título de Ingeniero Electrónico, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Guayaquil,



---

**PhD. David Barrios Puerto**



## **DEDICATORIA**

Dedico este trabajo a Dios y a toda mi familia en especial a mi madre, esposa e hijo pilares fundamentales de mi vida

Victor Mendez Gonzalez.

## **DEDICATORIA**

Este proyecto de graduación se lo dedico a mis padres Wilson Zambrano. y Luz Maria Flores, quienes supieron guiarme por el buen camino y por darme todas fuerzas para poder seguir adelante, para nunca desmayar en los problemas que se me presentaban.

Además, dedicar este proyecto a mis amigos y docentes que de alguna u otra forma estuvieron en los momentos difíciles que atravesé a lo largo de este camino.

Kiara Zambrano Flores

## **AGRADECIMIENTO**

Agradezco infinitamente a Dios, a mi madre, a mi esposa e hijo , pues han sido un pilar fundamental a lo largo de mi vida y con sus sabios consejos, esfuerzos y enseñanzas han logrado formar en mí, el criterio necesario para alcanzar cualquier objetivo que me propusiera.

Quiero agradecer también a toda mi familia, amigos y docentes que han servido de apoyo y motor para conseguir nuevas metas y propósitos.

Víctor Méndez González.

## **AGRADECIMIENTO**

Agradezco a Dios por protegerme, darme fuerzas a lo largo de mi carrera, por saberme guiar por el camino correcto y darme fuerzas para enfrentar todas las dificultades que a lo largo de mi vida se me han presentado

A mis padres por ser el ejemplo a seguir, que han sido toda la vida mis pilares fundamentales para mi realización como persona.

Y como no agradecer a mis buenos amigos, docentes y compañeros por confiar y creer en mí, los que siempre me han motivado y han hecho de mi etapa universitaria una de las mejores en mi vida.

Kiara Zambrano.

## RESUMEN

AÑO	ALUMNOS	DIRECTOR DE TESIS	TEMA TESIS
2018	<b>MENDEZ GONZALEZ VICTOR LUIS</b>  <b>ZAMBRANO FLORES KIARA MADELAINE</b>	ING.GIANFRANCO DI MATTIA CASTRO	<b>“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR INDUSTRIAL, MEDIANTE PLC’S ARDUINO”</b>

El presente proyecto de titulación tiene por objetivo el diseño, construcción e implementación de un módulo de aprendizaje orientados a la instrumentación física y virtual en el sector industrial mediante PLC’S arduino, aplicado en un proceso de dosificación y mezclado de sustancia a una determinada temperatura, con la finalidad de reforzar los conocimientos impartidos a los estudiantes de la materia de microcontroladores en un proceso real en el cual podrán despejar todas sus inquietudes del aula a la parte practica el sistema también incluye el monitoreo de todo el proceso por medio de una HMI. Otorgando la facilidad a los estudiantes de aprender diseños de uso más industrial con arduino.

El sistema está compuesto de dos módulos didácticos utilizando tecnologías open source, el primer módulo está conformado por un PLC Mduino 38R con sus respectivas entradas digitales a través de pulsadores y salidas digitales a través de leds, así como la simulación de señales analógicas por medio de potenciómetros. El segundo módulo está conformado de un PLC Mduino 19R, este cumplirá con la función de esclavo, ambos módulos contarán con elementos básicos de control como, relés, pulsadores, luces indicadoras, potenciómetros.

La implementación de este sistema se basa en un modelo didáctico a escala de un proceso de dosificación y mezclado de sustancias, mediante la programación de arduino, refleja la facilidad de automatizar diversidad de procesos industriales con una programación

entendible y sin licencias, además de la comprensión, aplicación de conceptos teóricos de diversas materias como:

- Automatización industrial
- Microcontroladores
- Electrónica analógica

Enfocando así al estudiante al uso de nuevas tecnologías al sector industrial.

Palabras clave: Arduino IDE, Conexiones I2C/USB/RS232, Sistema a escala, Open Source.

## ABSTRACT

ANY	STUDENTS	SUPERVISOR	THESIS TOPIC
2017	<b>MENDEZ GONZALEZ VICTOR LUIS</b>  <b>ZAMBRANO FLORES KIARA MADELAINE</b>	ING.GIANFRANC O DI MATTIA CASTRO	<b>IMPLEMENTATI ON OF LEARNING MODULES ORIENTED TO PHYSICAL AND VIRTUAL INSTRUMENTAT ION IN THE INDUSTRIAL SECTOR BY PLC ARDUINO</b>

The purpose of this degree project is to design, build and implement a learning module aimed at physical and virtual instrumentation in the industrial sector through Arduino PLC'S, applied in a process of dosing and mixing of substance at a certain temperature with the purpose of reinforcing the knowledge imparted to students of the subject of microcontrollers in a real process in which they can clear all their concerns from the classroom to the practical part of the system also includes the monitoring of the entire process through an HMI. Granting the facility to students to learn more industrial dsesigns with Arduino.

The system is composed of two teaching modules using open source technologies; the first module is made up of a Mduino 38R PLC with its respective digital inputs through pushbuttons and digital outputs through LEDs, as well as the simulation of analog signals through potentiometers. The second module is made up of a 19R Mduino plc, this will fulfill the function of a slave, both modules will have basic control elements such as relays, pushbuttons, indicator lights, potentiometers.

The implementation of this system in a didactic scale model of a process of dosing and mixing of substances, through arduino programming, reflects the ease of automating diversity of industrial processes with an understandable and unlicensed programming, in addition to

compression, application of theoretical concepts of various subjects such as:

- Industrial automation
- Microcontrollers
- Analogue electronics

Thus, focusing students on the use of new technologies in the industrial sector.

Keywords: Arduino IDE, I2C / USB / RS232 connections, Scale system, Open Source.

## ÍNDICE

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN. ...II	
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UPS .....	III
CERTIFICACIÓN DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN .....	IV
DEDICATORIA .....	V
DEDICATORIA .....	VI
AGRADECIMIENTO .....	VII
AGRADECIMIENTO .....	VIII
RESUMEN .....	IX
ABSTRACT.....	XI
INTRODUCCIÓN .....	18
1. El Problema.....	20
1.1 Descripción del problema .....	20
1.2 Antecedentes .....	20
1.3 Importancia y Alcance .....	20
1.4 Delimitación.....	21
1.4.1 Temporal .....	21
1.4.2 Espacial.....	21
1.4.3 Académico .....	21
1.5 Explicación del problema.....	21
1.6 Objetivos .....	22
1.6.1 Objetivo General.....	22
1.6.2 Objetivos específicos .....	22
1.7 Marco Metodológico.....	23
1.7.1 Método Inductivo.....	23
1.7.2 Método Deductivo.....	23
1.7.3 Método Experimental.....	24
2. Fundamentos Teóricos. ....	24
2.1. Definición de PLC .....	24
2.1.1 Ventaja de los PLC'S.....	25
2.1.2 Tipos de PLC .....	26
2.2 PLC Open Source .....	27
2.3 PLC M-DUINO Ethernet 38 R + .....	28
2.3.1 Especificaciones técnicas M-DUINO Ethernet 38R.....	29
2.4 PLC M-DUINO 19R.....	30
2.4.1 Especificaciones Técnicas M-DUINO 19R .....	31

2.5	Banana TouCH PI.....	32
2.5.1	Especificaciones Técnicas Banana Touch PI.....	33
2.6	Arduino Mega 2560 .....	33
2.6.1	Especificaciones técnicas Arduino Mega 2560.....	37
2.7	Tarjeta Latte Panda Enterprise.....	37
2.7.1	Especificación técnica LattePanda.....	39
2.8	Bus I2C .....	39
2.8.1	Señales del Bus I2C .....	40
2.9	Protocolo de comunicación I2C.....	41
2.10	Definición de términos Protocolo I2C .....	42
2.11	Comunicación Serial.....	43
2.12	Labview.....	45
3.	Marco metodológico .....	46
3.1.	Introducción del proyecto. ....	46
3.2.	Diseño del tablero. ....	48
3.3.	Cableado del tablero .....	50
3.4.	Diseño de la Planta.....	55
3.4.1.	Instalación del sensor ultrasónico .....	58
3.5.	Interfaz Panel Touch LatePanda. ....	59
4.-	Resultados.....	60
4.1.-	Prácticas.....	60
5.-	Conclusiones:.....	71
6.-	Recomendaciones: .....	71
7.-	Anexos .....	73
8.-	Bibliografía.....	148

## ÍNDICE DE FIGURAS

Figura 2. 1: Proceso de llenado de botellas con agua .....	24
Figura 2. 2: Controlador lógico programable.....	25
Figura 2. 3: Ventajas de PLC'S .....	25
Figura 2. 4: PLC Compacto .....	26
Figura 2. 5: Estructura de un PLC Modular.....	26
Figura 2. 6: PLC OpenSource .....	27
Figura 2. 7: PLC M-DUINO .....	28
Figura 2. 8: PLC M-DUINO Ethernet 38 R.....	28
Figura 2. 9: Especificaciones técnicas M-DUINO Ethernet 38R .....	29
Figura 2. 10: Parte Frontal M-DUINO 38R.....	30
Figura 2. 11: PLC M-DUINO 19R Frontal.....	31
Figura 2. 12: Especificaciones técnicas M-DUINO 19R.....	31
Figura 2. 13: Banana Touch PI .....	33
Figura 2. 14: Arduino Mega 2560.....	34
Figura 2. 15: Entorno de programación IDE.....	35
Figura 2. 16: Diagrama tarjeta Arduino Mega 2560.....	36
Figura 2. 17: Tarjeta Latte Panda Enterprise .....	37
Figura 2. 18: Pantalla con Latte Panda Integrada .....	38
Figura 2. 19: Especificaciones técnicas .....	39
Figura 2. 20: Comunicación de datos I2C.....	40
Figura 2. 21: Esquema de un Bus I2C.....	40
Figura 2. 22: Maestro-Esclavo I2C .....	41
Figura 2. 23: Señales Bus I2C.....	41
Figura 2. 24: Comunicación de dos líneas .....	42
Figura 2. 25: Protocolo I2C.....	43
Figura 2. 26: Configuración puerto serial DB9.....	44
Figura 2. 27: Configuración puerto DB25 .....	44
Figura 2. 28: Equipos de National Instruments.....	45
Figura 2. 29: Interconexiones con diferentes equipos NI .....	46
Figura 3.1: Diagrama de bloques del proyecto .....	47
Figura 3.2: Diagrama de control .....	47
Figura 3.3: Estructura galvanizada al frio módulos maestro esclavo.....	48
Figura 3.4: Estructura galvanizada terminada al frio módulos maestro esclavo.....	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Figura 3.5: Estructura galvanizada al frio módulos maestro esclavo con acabado terminado .....	49
Figura 3.6: Instalación de canaletas y módulos programables.....	52
Figura 3.7: Avance del cableado en el tablero .....	52
Figura 3.8: Vista externa del tablero ya integrado luces piloto.....	53
Figura 3.9: Ubicación del panel en la parte externa del tablero.....	53
Figura 3.10: Cableado interno de las luces piloto y panel .....	54
Figura 3.11: Vista externa del tablero con sus elementos instalados.....	55
Figura 3.12: Parte posterior de la planta .....	55
Figura 3.13: Ubicación de resistencia térmica .....	56
Figura 3.14: Vista frontal superior de la planta .....	56
Figura 3.15: Cableado de la parte interna del tablero de control para la planta.....	57
Figura 3.16: Diseño de tarjetas para regular el voltaje de entrada.....	57
Figura 3.17: Parte posterior del diseño de la tarjeta.....	58

Figura 3.18: Vista real de la tarjeta .....	58
Figura 3.19: Sensor ultrasónico.....	58
Figura 3.20: Ubicación del sensor en la planta .....	59
Figura 3.21: Vista del panel operativo .....	59
Figura 3.22: Líneas de programación de Arduino en el panel .....	60

## ÍNDICE DE TABLAS

Tabla 2. 1: Especificaciones del Arduino Mega 2560.....	20
---	----

## INTRODUCCIÓN

Observar que en el laboratorio de micro controladores no hay suficientes equipos para el desarrollo de prácticas con aplicación Industrial, que permitan a los estudiantes disipar dudas que tengan de la teoría dada en el aula por las materias de especialización, ha creado una necesidad de innovación y desarrollo para la puesta en marcha de módulos prácticos que simulen procesos del sector industrial para lo cual refuercen sus conocimientos. Por ello se lleva a cabo el siguiente proyecto:

- La implementación de módulos didácticos de aprendizaje que simulen plantas industriales, la cual utilice tecnología OPEN SOURCE, aportaría con el desarrollo de profesionales más capacitados y con propuestas de soluciones poco usadas como lo son el uso de los microcontroladores, en el sector industrial, y con un software libre de licencia, como lo es Arduino, fácil de programar y amigable a los sistemas operativos de Windows. El enfoque de este proyecto está en el interés e interpretación de nuevas tecnologías para implementar en el ámbito industrial. Esto beneficiará al alumnado. Con ello mejorará sus conocimientos adquiridos en el aula y le ayudará al desarrollo de mejores propuestas a futuro.

El sistema lo conforman dos módulos didácticos utilizando tecnologías open Source. El primer módulo tiene un PLC Mduino 38R+ con sus respectivas entradas digitales a través de pulsadores y salidas digitales mediante LEDs, así como la simulación de señales analógicas por medio de potenciómetros. El segundo módulo está conformado de un PLC Mduino 19R, este cumplirá con la función de esclavo, ambos módulos contarán con elementos básicos de control como, relés, pulsadores, luces indicadoras, potenciómetros.

También incorpora la conexión de puertos I2C, USB, RS232 y una pantalla HMI de 10.1". La programación se realizará mediante Arduino IDE. La pantalla HMI cuenta con una tarjeta Latte Panda.

En conclusión, se diseña un modelo didáctico a escala de un proceso de dosificación y mezclado de sustancias, estructurado bajo la plataforma de Arduino. Con este sistema se refleja la facilidad de automatizar una diversidad de procesos industriales con una

programación entendible y libre de licencias, además de la comprensión y aplicación de conceptos teóricos de diversas materias de la especialidad como:

- Automatización industrial
- Microcontroladores
- Electrónica analógica

El cual aporta al desarrollo y aprendizaje para los estudiantes en el uso de nuevas tecnologías para el sector industrial.

## **1. El Problema**

### **1.1 Descripción del problema**

En la actualidad existe la necesidad de reforzar el laboratorio de Microcontroladores con equipos que permitan a sus estudiantes poner en práctica soluciones a problemáticas del sector técnico – industrial, con módulos en los cuales ayuden a desarrollar aplicaciones orientadas a problemáticas más comunes en la automatización, como el control y monitoreo de sistemas térmicos, mecánicos, eléctricos y neumáticos.

La idea de implementar módulos prácticos surgió para generar una visión más amplia de las diferentes aplicaciones que se les puede dar a los microcontroladores en el sector industrial, de manera que sirva para orientar y motivar a los estudiantes al uso de nuevas tecnologías como lo es **open Source**. Este tipo de soluciones Open Source son más flexibles y accesibles que las soluciones industriales estándar.

### **1.2 Antecedentes**

En la búsqueda de ampliar las opciones de equipos disponibles para el aprendizaje de la materia de microcontroladores y a fin de que los estudiantes puedan utilizarlo como soluciones en el ámbito técnico-industrial, se plantea como necesario la implementación de un módulo compuesto por un Controlador Lógico Programable (PLC) cuya estructura principal constituya un Arduino, dando una nueva alternativa para la solución a problemas de tipo industrial en el área de automatización con equipos de bajo coste y de software flexible.

La aplicación de este módulo en la materia de microcontroladores aumentará el interés de la misma en los estudiantes, dando apertura a procesos o soluciones que sean objeto de investigación, así como generando profesionales listos para dar ideas innovadoras y soluciones con tecnologías actualizadas en la optimización de procesos e incluso en la implementación de nuevos proyectos en el sector industrial.

Los dispositivos utilizados en el desarrollo de esta implementación ampliarán de manera considerable la visión del uso de microcontroladores en ambientes industriales

### **1.3 Importancia y Alcance**

El proyecto busca fortalecer de forma práctica la teoría explicada en las materias de microcontroladores I y II mediante el diseño, construcción y programación de módulos didácticos, en los cuales puedan simular y desarrollar situaciones o mejoras que se presentan en el sector industrial, incentivando así a los estudiantes al uso de otras tecnologías en la implementación de soluciones a procesos industriales.

Este proyecto busca fomentar la investigación de nuevas tecnologías, así como familiarizar a los estudiantes con las diversas aplicaciones y beneficios que ofrece las soluciones open source para incrementar la visualización en la interacción con otras materias impartidas en la universidad, como redes de computadoras, instrumentación, comunicaciones, etc.

Los módulos didácticos presentados permiten orientar más a los lectores en el uso de los mismos para que de esta forma sea una referencia en proyectos posteriores de electrónica industrial o telecomunicaciones.

## **1.4 Delimitación**

### **1.4.1 Temporal**

El proyecto de titulación tuvo una duración de 18 meses contados a partir de la fecha de aprobación del tema.

### **1.4.2 Espacial**

El proyecto está asignado para el laboratorio de microcontroladores donde se imparten materias como electrónica digital e instrumentación de la Universidad Politécnica Salesiana sede Guayaquil y será desarrollado en el año 2019.

### **1.4.3 Académico**

El proyecto consta de dos módulos didácticos y una planta que simula un proceso industrial. El primer módulo, que hace de maestro, contiene elementos de control como el PLC Arduino (que es un hardware de código abierto, open source hardware). Está especialmente diseñado para su uso en entorno profesional. Este PLC dispone de 38 E/S, también dispone de diferentes sistemas de comunicación lo que ofrece mayor flexibilidad y control.

El PLC Mduino permite expandirse hasta con 127 módulos mediante el sistema I2C, lo que le permite gobernar hasta 7100 E/S en modo maestro-esclavo. Además, posee módulos adicionales de sensores. Se programa mediante la plataforma Arduino IDE, que puede ser descargada desde la web de Arduino sin ningún costo, y puede ser programado mediante el puerto USB.

La pantalla táctil cuenta con varias E/S para evitar emplear un PLC en aplicaciones simples. La Touchberry permite conectarse con todos los equipos que distribuye la compañía industrial Shield y de otras marcas, para procesos complejos se pueden interconectar con diferentes TOUCHBERRY PI, esto permite automatizar toda una planta de producción y disponer de todos los datos necesarios en cualquier momento, es flexible y dinámico de poner en marcha, solo se configura la SD.

## **1.5 Explicación del problema.**

Se propone implementar dos módulos para prácticas de laboratorio esto permitirá al estudiante un mejor aprendizaje y desarrollo en el laboratorio de microcontroladores utilizando tecnologías tipo **Open Source**, los módulos constan de:

El primer módulo estará conformado por un PLC Mduino 38R, HMI TOUCHBERRY PI y una planta donde se podrán simular todo tipo de señales de entradas digitales a través de

pulsadores y señales digitales de salida a través de Leds así como también se podrán simular señales analógicas de entrada mediante potenciómetros [ 0 a 10 Vdc] y una entrada analógica real de temperatura a través de una termocupla. Con estas entradas y salidas se podrá desarrollar variedad de prácticas con un entorno industrial.

El segundo módulo estará conformado por un PLC Mduino 19R y en una simulación cumplirá las funciones de esclavo. En este módulo también tendrá la posibilidad de conectarse a través de sus puertos de comunicación con otros dispositivos **open Source** I2C, USB,

## **1.6 Objetivos**

### **1.6.1 Objetivo General**

Implementar módulos de aprendizaje orientados al uso de microcontroladores simulando diferentes problemáticas en los procesos del sector industrial, mediante PLC's open Source constituido por programas libres de licencias, aplicado a un proceso de dos tanques en donde se controla temperatura, nivel y se dosifica líquidos para luego ser mezclados.

### **1.6.2 Objetivos específicos**

- Implementar aplicaciones industriales con hardware, software que no requiera licencia y con código abierto que pueda ser adaptable a diferentes ambientes.
- Diseñar dos módulos maestros/esclavo open Source utilizando equipos Mduino 38R y Mduino 19R
- Configurar y programar los módulos utilizando programación arduino IDE.
- Desarrollar el proceso de una dosificadora /mezcladora de líquidos que contenga sensores compatibles con Arduino para controlar y monitorear temperatura / nivel,
- Configurar y programar la pantalla Touch PI, mediante la cual permita monitorear continuamente el estado de las variables de entrada y salida de la dosificadora/mezcladora, configurando los distintos parámetros de temperatura y nivel.
- Realizar la respectiva configuración de las librerías para el uso del sistema basado en Arduino.
- Parametrizar la variable de temperatura, para poder establecer valores específicos de dosificación de líquido de un recipiente a otro e implementar un control PID de temperatura.
- Realizar pruebas de funcionamiento para verificar la comunicación del proceso, con los módulos y la pantalla HMI.

- Elaborar una guía completa de 10 prácticas didácticas en las cuales se desarrolle de manera gradual el manejo y aplicación de los módulos del laboratorio, las cuales son las siguientes:
  - Práctica #1: Reconocer las funciones que tienen los módulos PLC MDuino38R (comprobación de salidas Digitales).
  - Práctica #2: Reconocer salidas y entradas digitales del PLC maestro.
  - Práctica #3: Lectura de señales analógicas PLC maestro.
  - Práctica #4: Prueba de señales de entrada y salidas modulo esclavo 19R.
  - Práctica #5: Comprobación de las entradas analógicas modulo esclavo.
  - Práctica #6: Comunicación serial entre modulo maestro - esclavo
  - Práctica #7: Comunicación serial modulo esclavo – maestro.
  - Práctica #8: Reconocimiento de la señal de una pt-100 modulo maestro-Planta.
  - Práctica #9: Proceso de recirculación desde el tanque de muestra hacia los tanques 1 y 2 de la planta contralados por el módulo maestro.
  - Práctica #10: Control de planta dosificadora/mezcladora desde el maestro con la librería R-LINK LabVIEW.

## **1.7 Marco Metodológico.**

El desarrollo de este Proyecto se emplea las siguientes técnicas de investigación:

### **1.7.1 Método Inductivo.**

Los módulos diseñados cuentan con elementos de control que son fáciles de adquirir y manipular en la industria actualmente. Esto significa que los módulos serán funcionales en cualquier ámbito que involucre procesos industriales.

### **1.7.2 Método Deductivo**

Este método fue utilizado durante la recolección y tabulación de datos, para concluir los patrones de configuración para cada práctica propuesta.

Se observaron diferentes sistemas a escala de dosificadores y mezcladoras implementados en la industria, recolectando información para poder definir el diseño y programación de nuestros módulos didácticos, determinando las variables a controlar, así como los elementos

de recepción y accionamiento de las señales.

### 1.7.3 Método Experimental.

Este método fue utilizado al desarrollar las pruebas preliminares para la configuración de cada una de las prácticas establecidas en los objetivos, de esta forma se determinaron parámetros de programación.

## 2. Fundamentos Teóricos.

### 2.1. Definición de PLC

Se denomina PLC a un controlador lógico programable el cual trabaja similar a una computadora que es frecuentemente utilizada para la automatización de procesos. Una de las características más importantes del equipo es que permite controlar las secuencias en tiempo real haciendo un sistema más ágil y eficiente, como ejemplo: un control de llenado de botellas con agua. Estas necesitan disminuir tiempos de llenado para optimizar el sistema y obtener más producción en menos tiempo, para ello se utilizan PLC's. (Electric, 2020)

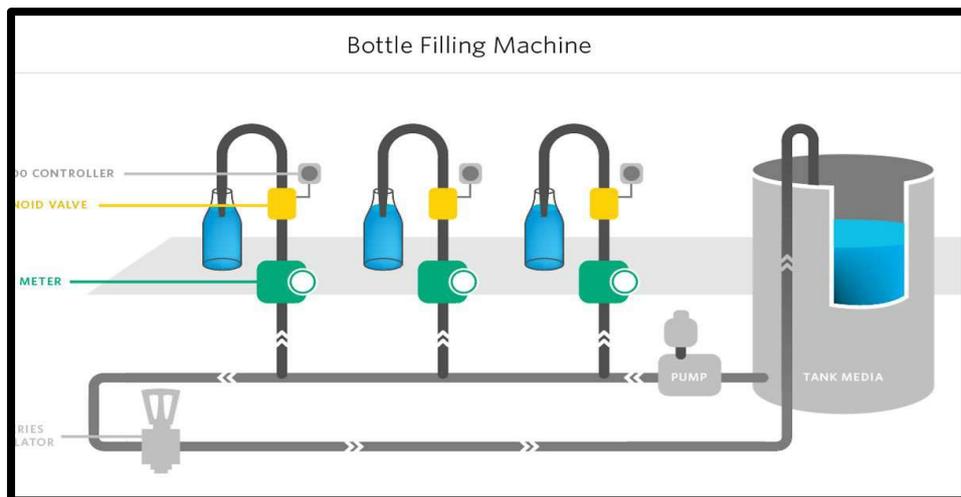


Figura 2. 1: Proceso de llenado de botellas con agua (Electric, 2020)

La definición más precisa de estos elementos viene dada por la NEMA (Asociación Nacional de Fabricantes Eléctricos) en la que establece al PLC como un elemento electrónico que utiliza entre sus principales elementos una memoria programable donde se guardan instrucciones como operaciones lógicas, secuencias de acciones, especificaciones temporales, contadores y cálculos para el control mediante módulos E/S analógicos o digitales sobre diferentes tipos de máquinas / procesos. (Mecafenix, 2018)

El campo de aplicación de los PLCs es muy diverso, abarcando distintos tipos de industrias (construcción, aeroespacial, automoción, alimentos, etc.).

Una gran diferencia con las computadoras de uso general es que los PLCs fueron diseñados para trabajar con múltiples señales de entradas y salidas, amplios rangos de temperatura, inmunidad al ruido eléctrico, resistencia al impacto y vibraciones. En términos generales fue diseñado para aplicaciones industriales, manejando rangos de variables no aptas para una PC. (Mecafenix, 2018)



Figura 2. 2: Controlador lógico programable (Mecafenix, 2018)

### 2.1.1 Ventaja de los PLC'S

- Introducir modificaciones sin cambiar el cableado.
- Mínimo espacio de ocupación.
- Economía de mantenimiento.
- Posibilidad de gobernar varias máquinas con un mismo autómata.
- Menor tiempo para la puesta en funcionamiento del proceso.
- Menor coste de mano de obra para la instalación.
- Sistema de producción más eficiente.
- Optimizar el sistema en tiempo real.
- Existe un mejor monitoreo del proceso lo que le permite al operador detectar fallas en menor tiempo. (Recursostic, 2015)



Figura 2. 3: Ventajas de PLC'S (Recursostic, 2015)

### 2.1.2 Tipos de PLC

**PLC NANO:** Es un PLC de tipo compacto donde se integran fuente de alimentación, Unidad Central de Proceso (CPU), entradas y salidas. Tienen la capacidad de operar un grupo reducido de entradas y salidas, por lo general es un número menor a 100.

**PLC COMPACTO:** Estos PLC tienen la misma estructura del PLC nano, pero tienen a disposición el control de varias entradas y salidas (alrededor de 500) y pueden expandirse con varios módulos especiales como se detallan a continuación: (Recursostic, 2018)

- Contadores.
- Módulos de comunicaciones.
- Interfaces.
- Módulos de expansiones de entrada y salida.



Figura 2. 4: PLC Compacto (Recursostic, 2018)

**PLC MODULAR:** Están compuestos por un grupo de elementos (módulos de expansión) hasta llegar al controlador final entre sus partes tenemos:

- Rack
- La fuente de alimentación
- La CPU
- Los módulos de entrada y salida.

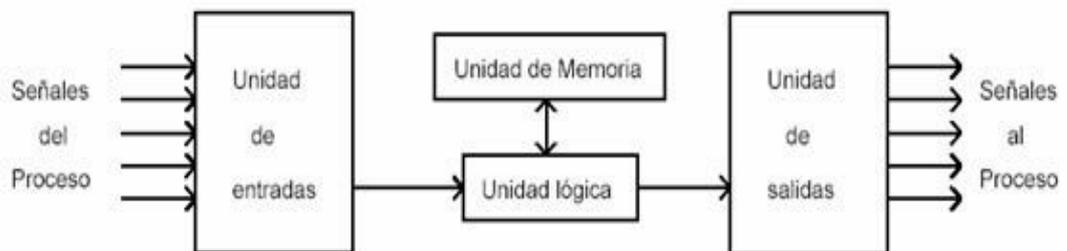


Figura 2. 5: Estructura de un PLC Modular (Recursostic, 2018)

## 2.2 PLC Open Source

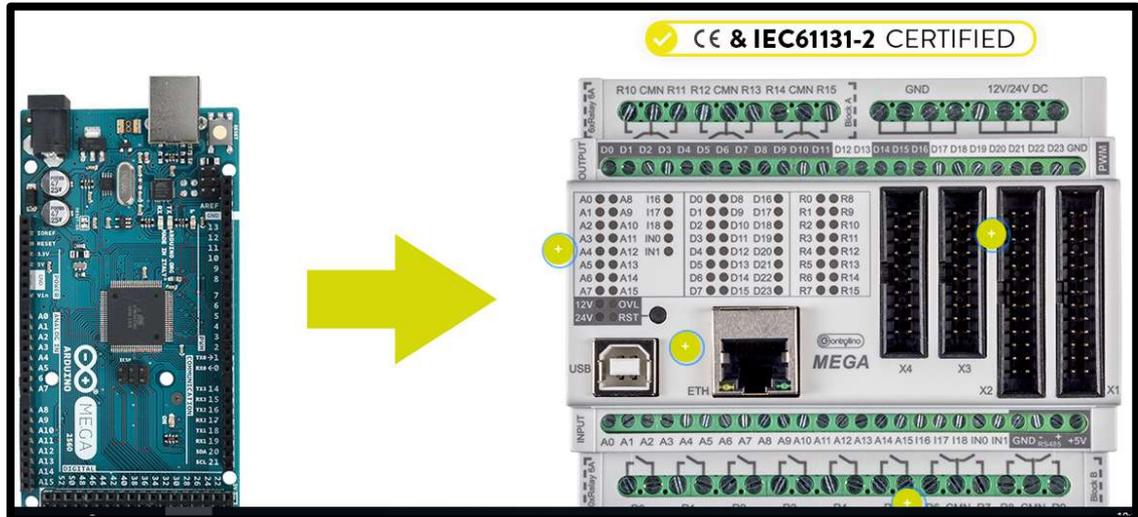


Figura 2. 6: PLC Open Source (Controllino, 2020)

Un PLC open Source combina la flexibilidad que tiene el sistema Arduino con la seguridad que tienen los PLC en el sector industrial. Es un elemento que maneja una estructura hardware de código abierto preparado para ser utilizado en entornos industriales, lo que da la facilidad de adquisición para la automatización en el sector. Como ventajas, los costes en programas, monitoreo y plataformas de datos son más bajos para un PLC open source en comparación con PLCs de otras marcas que necesitan tener una licencia y eso determina más gastos para implementar un sistema. (Controllino, 2020)

Teniendo en cuenta también que no necesita un software especial para el diseño de un sistema de supervisión (SCADA) el equipo es completamente amigable con softwares como Matlab, Labview, etc.

Lo que sí es importante recalcar es que se necesita personal calificado para el manejo y programación del equipo. (Controllino, 2020)

Este tipo de PLC beneficia a las pequeñas industrias ya que pueden tener grandes recursos y beneficios desde el principio de la adquisición del equipo. (Shields, 2020)

Tiene algunas características entre las más importantes tenemos:

- Entradas y salidas análogas, el límite mínimo es 61
- 21 entradas digitales con protección ESD
- 24 salidas digitales
- Proporciona interfaces vía serial, RS485, I2C o SPI
- Salidas Relays 230 V
- Integra RTC (Real time clock)
- Es compatible con varios softwares como ejemplo Matlab, Labview, etc. (Shields, 2020)



### 2.3.1 Especificaciones técnicas M-DUINO Ethernet 38R

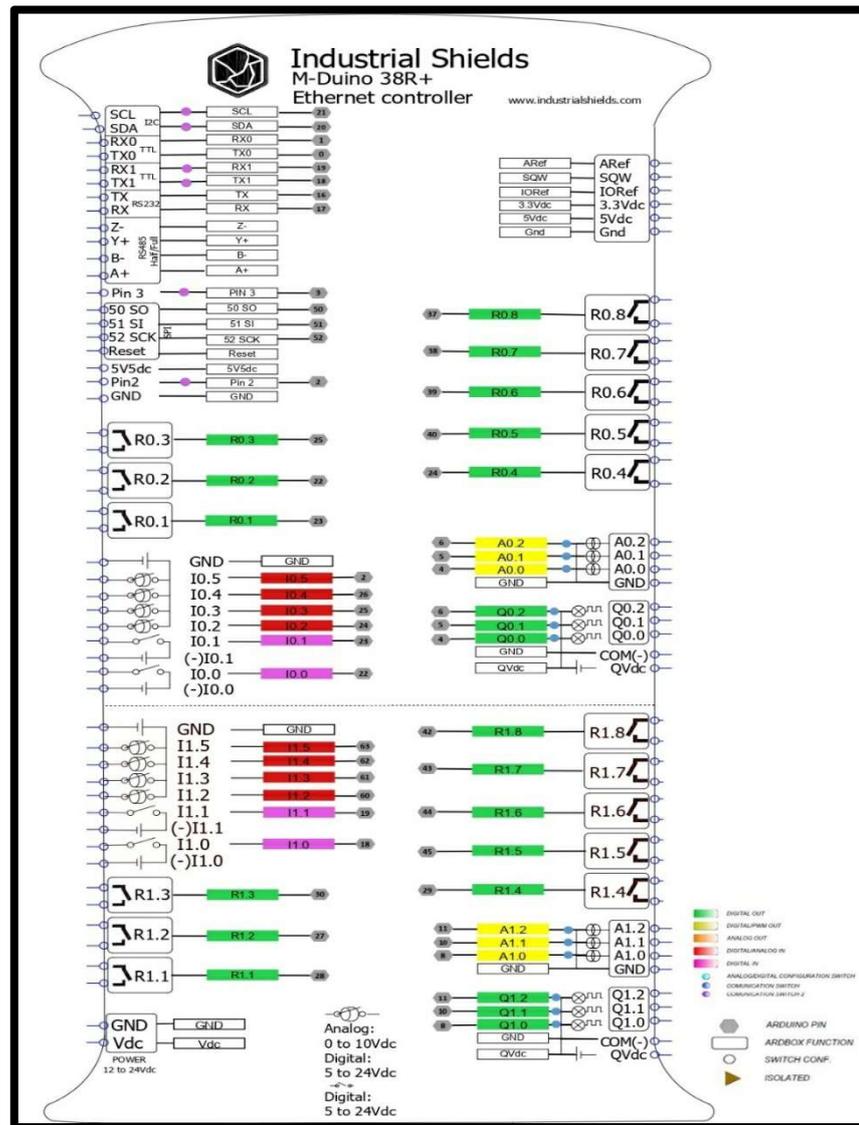


Figura 2. 9: Especificaciones técnicas M-DUINO Ethernet 38R (Shields, 2020)

#### Entradas:

- 10 entradas configurables (analógicas de 0 a 10 vdc) o digitales (5 a 24 vdc)
- 4 entradas digitales aisladas de (5 a 24 vdc)

#### Salidas:

- 8 salidas de relé (220 Vac – 5A)
- 6 salidas configurables analógicas (0 a 10 Vdc, 8 bit) / Digitales (5 a 24 Vdc)
- 5 digitales aisladas (5 a 24 Vdc)

#### Comunicación:

- 1 Puerto Ethernet
- 1 Puerto USB (type B)
- 1 Puerto I2C
- 3 Puertos TTL

- 1 Puerto RS-232
- 1 Puerto HALF/FULL Duplex RS-485
- 1 Puerto SPI externo (Usa MOSI, MISO, SS pines Arduino)
- TCP IP/Modbus TCP / Modbus RTU
- Capacidad de Web Server
- Protocolos Industriales.

Otras especificaciones técnicas:

- RTC (Real Time Clock)
- Montaje en riel DIN
- Máximo consume de corriente (1.2 A)
- Memoria flash 256 KB (8 KB para bootloader)
- SRAM 8 KB
- EEPROM 4 KB
- Clock Speed 16 MHZ



Figura 2. 10: Parte Frontal M-DUINO 38R (Shields, Industrial Shields, 2018)

## 2.4 PLC M-DUINO 19R

Similar que el PLC 38R este equipo contiene como componente principal en la estructura de Hardware una plataforma Arduino. (Shields, Industrial Shields, 2018)

Entre las características que resaltan al equipo:

- Dispone de módulos de E/S
- Tiene a disposición módulos de comunicación, lo que permite tener un excelente grado de confiabilidad y flexibilidad a una menor escala que el 38R.
- Se puede programar a través del puerto USB o de forma remota a través del puerto Ethernet con la plataforma de Arduino IDE. (Shields, Industrial Shields, 2018)



Figura 2. 11: PLC M-DUINO 19R Frontal (Shields, Industrial Shields, 2018)

## 2.4.1 Especificaciones Técnicas M-DUINO 19R

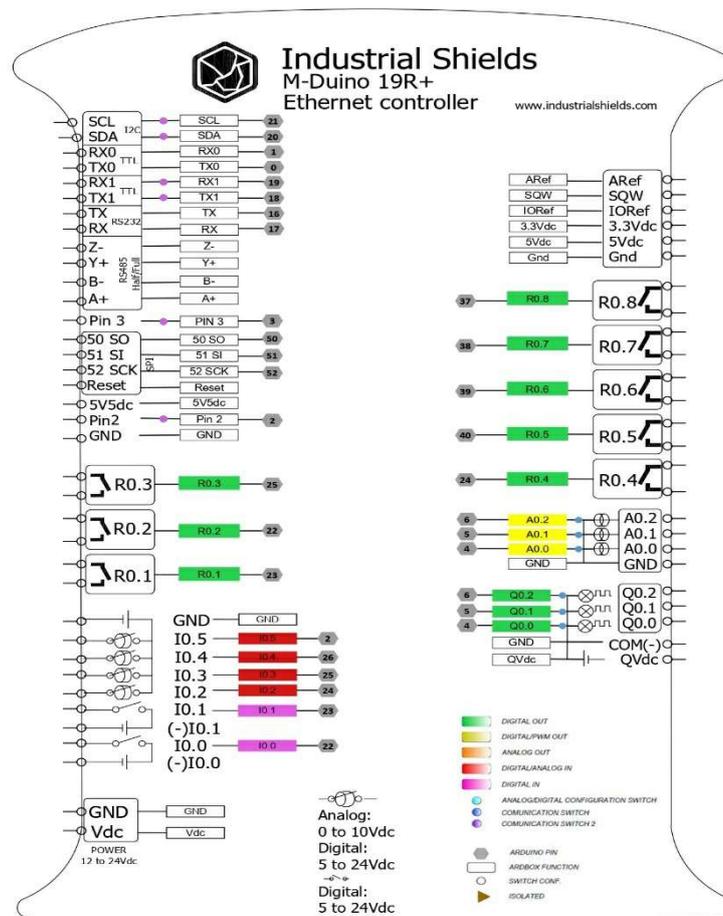


Figura 2. 12: Especificaciones técnicas M-DUINO 19R (Shields, Industrial Shields, 2018)

### Entradas:

- 6 entradas Analógicas (0 a 10 Vdc, 10 bit) / digitales (5 a 24 Vdc) todas configurables mediante software.
- 2 entradas de interrupción (5 a 24 Vdc), también pueden trabajar como entradas digitales de (5 a 24 Vdc).

### Salidas:

- 8 salidas de relé (220 VAC – 5 A)
- 3 salidas Analógicas (0 a 10 VDC, 8 bit) / Digitales (5 a 24 Vdc)

Comunicaciones:

- 1 Puerto Ethernet
- 1 Puerto USB (Tipo B)
- 1 Puerto I2C
- 3 Puertos TTL
- 1 Puerto RS-232
- 1 Puerto HALF/FULL Duplex RS-485
- 1 Puerto SPI externo
- TCP IP / Modbus RTU
- Capacidad Web Server
- Protocolos Industriales

Otras especificaciones:

- Montaje en riel DIN
- Máximo consume de corriente de 1.2 A
- Memoria Flash de 256 KB
- SRAM de 8KB
- EEPROM 4 KB
- Velocidad de Reloj 16 MHz

## **2.5 Banana TouCH PI**

El dispositivo panel Pc está basado en el sistema operativo GNU/Linux Debían OS, Android Os o Windows por defecto instalados en tarjeta SD, el cual contiene una variedad de interfaces integradas. Entre estas tenemos:

- USB
- ETHERNET
- Wifi

Se tiene a disposición la opción de poder controlar de forma remota todos los parámetros, entradas y salidas del sistema de control.

Los protocolos de hardware abierto de estos equipos no solo permiten la comunicación con otros PLC's de la marca industrial Shields por el contrario, existe la opción de realizar la comunicación con otros dispositivos y máquinas de otras marcas.



Figura 2. 13: Banana Touch PI (Schneider Electric, 2011)

Para otros sistemas de un rango de complejidad más elevado se puede establecer redes entre varios paneles Pc, el cual desarrolla y da una solución integral para supervisar y controlar de manera efectiva las plantas de producción completas. Con una lectura de datos en tiempo real, permite diagnosticar fallos en menor tiempo y esto implica ahorro de costo en el proceso, dando mucha confianza en la industria.

### 2.5.1 Especificaciones Técnicas Banana Touch PI

- CPU 1.2 GHz Quad-core ARM C rtex A53 con procesador de 64 Bit
- Puertos USB (2.0), OTG (Micro USB)
- Almacenamiento en memoria microSD 64 GB
- Sistema Operativo Linux Deb an
- Comunicaciones con dispositivos de bajo nivel tales como UART – I2C- SPI
- Sistema multit ctil LVDS resistiva
- Alta Resoluci n 1280x720
- Resistencia de temperatura de trabajo de 0 grados a 40 grados cent grados.

### 2.6 Arduino Mega 2560

Arduino es mundialmente conocido por estar basado en hardware Open Source, que actualmente tiene a disposici n una amplia gama de dispositivos, permitiendo al personal interesado desarrollar proyectos innovadores y de gran alcance con un bajo coste. Es una plataforma de desarrollo que integra un microcontrolador re-programable y una serie de pines hembra, lo cual da apertura para conectar diferentes sensores o actuadores pudiendo llevar a cabo la realizaci n de varios proyectos.

La tarjeta electr nica Arduino Mega 2560 se dise n  para proyectos con mayor complejidad. Para esto es necesario tener m s entradas y salidas, m s memoria de almacenamiento y RAM.

Las caracter sticas m s relevantes para adquirir Arduino:

- **Arduino es libre y extensible:** cualquiera que desee ampliar y mejorar el dise o hardware de las placas, as  como el entorno de desarrollo, puede hacerlo sin problemas. Esto permite que exista un rico ecosistema de placas electr nicas no oficiales para distintos prop sitos y de librer as de software de tercero que pueden

adaptarse mejor a nuestras necesidades (Arduino.cl, 2020)

- **Su entorno de programación es multiplataforma.** Tiene la facilidad de poder instalar y ejecutarse en sistemas operativos Windows, Mac OS y Linux (Arduino.cl, 2020)
- **Lenguaje de programación de fácil comprensión.** Su lenguaje de programación basado en C++ es de fácil comprensión. Da la facilidad a una entrada sencilla para nuevos programadores (Arduino.cl, 2020)
- **Re-usabilidad y versatilidad.** Tiene la facilidad de ser utilizado para varios proyectos, porque una vez terminado el proyecto es muy fácil su posterior desmontaje de los componentes externos a la placa y dar inicio con un nuevo proyecto. Todos los pines del microcontrolador están con una accesibilidad mediante conectores hembra lo cual da oportunidad de sacar su mayor rendimiento a todas las características del microcontrolador con un bajo riesgo de fallar en una conexión (Arduino.cl, 2020)
- **Bajo costo:** Su costo es relativamente bajo, comparado con otros equipos del medio y tomando en cuenta la gran apertura que tienen los diseños de estar tarjetas electrónicas (Arduino.cl, 2020)

Esta tarjeta se conforma por un microcontrolador de la familia Amaga 2560 de Atmel, llegando a la conclusión que no es más que la actualización de la placa Arduino mega anterior.

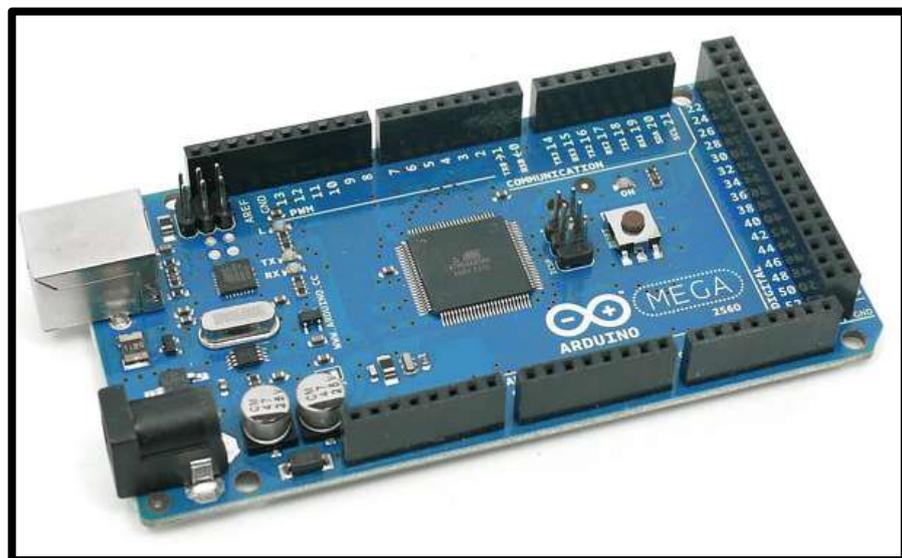


Figura 2. 14: Arduino Mega 2560 (TOOLS, 2018)

El Arduino mega 2560 consta de:

- 54 pines de entrada /salida (E/S) digital (de los cuales 15 se pueden utilizar como modulación de anchura de impulso (PWM)).
- 16 entradas analógicas.

- Puertos UARTs (puerto serie de hardware).
- Oscilador de 16 MHz.
- Conexión USB.
- Botón de Reset.
- Tiene 256KB de memoria flash de las cuales se utilizan 8KB para el arranque.
- 8 KB de SRAM.
- 4 KB de EEPROM que puede ser leída y escrita.
- Tiene un poli fusible reajutable el cual protege a los puertos USB. En caso de circular una cantidad mayor de 500mA por uno de los puertos el fusible interrumpirá la conexión hasta que se repare el cortocircuito (TOOLS, 2018).

La placa Arduino Mega es muy flexible ya que tiene facilidad para comunicarse con una PC, otra placa o con otros microcontroladores. Incluye en su estructura cuatro puertos serie UART para comunicación serial usando niveles TTL (5V). (TOOLS, 2018)

La librería da la opción para la comunicación serial en cualquiera de los pines digitales. También es compatible con TWI y SPI, que son librerías para realizar comunicaciones según se requiera en el proyecto a diseñar. A continuación se muestra el entorno de programación IDE con el cual se programa la placa Arduino: (Arduino, 2020)



Figura 2. 15: Entorno de programación IDE (Arduino, 2020)

A continuación se muestra la configuración interna de la tarjeta Arduino Mega 2560:



### 2.6.1 Especificaciones técnicas Arduino Mega 2560

MICROCONTROLADOR	ATMEGA 2560
Tensión de trabajo	5v
Tensión de entrada (recomendada )	7 a 12 V
Tensión de Entrada Limite	6 a 20 V
Pines Digitales I/O	54
Pines de entradas analógicas	16
Dc Corriente por Pin I/O	20 m A
Dc Corriente por Pin 3.3 V	50 m A
Memoria Flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Velocidad de Reloj	16 MHz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Tabla 2. 1: Especificaciones del Arduino Mega 2560 (Autores, 2019)

### 2.7 Tarjeta LattePanda Enterprise



○

Figura 2. 17: Tarjeta Latte Panda Enterprise (Electrónica, 2020)

La placa Latte panda es el primer diseño de placa para desarrollo que tiene la disposición de ejecutar una versión completa de Windows 10, la cual tiene similares características a las de un computador; es compatible con casi todos los dispositivos como impresoras, joysticks, etc. En conclusión, cualquier dispositivo que funcione con una computadora tendrá el mismo desempeño con la tarjeta LattePanda.

Dicha placa consta de un procesador Intel Quad Core y consta de múltiples opciones de conectividad que son necesarias para ambientes de desarrollo de automatización. Entre ellas tenemos:

- 3 puertos USB
- Wifi
- Bluetooth 4.0 integrado.

Una ventaja muy relevante de esta placa es que consta de un co-procesador Arduino que permite como prioridad controlar y sensar variables a través de la adquisición de estos datos con dispositivos compatibles.

La tarjeta Latte panda por sus características y dominio se encuentra en un escalón por encima de la Raspberry Pi y de otras placas para desarrollo ya que existe la posibilidad de almacenar en ella un sistema completo de Windows 10 dando la fiabilidad a cada aplicación con un entorno más accesible y robusto.

Latte panda incorpora varias placas únicas a un nivel completamente nuevo de potencia y rendimiento, teniendo a consideración las siguientes características más relevantes:

- Procesador Intel Quad Core ya mencionado de 1.8 GHz.
- 2-4 GB de RAM.
- 32 a 64 GB de memoria flash puede llegar a ejecutar fácilmente reconocimiento de imágenes en tiempo real, control CNC en tiempo real y mas.



Figura 2. 18: Pantalla con Latte Panda Integrada (LATTEPANDA, 2020)

### 2.7.1 Especificación técnica LattePanda

- Procesador Intel Cherry Trail Z8350 Quad Core 1.8 GHz.
- Sistema Operativo: edición completa preinstalada de Windows 10.
- RAM 2GB.
- Capacidad de almacenamiento 32 GB.
- GPU: Intel HD Graphics, 12 EU @200-500 MHz, memoria de un solo canal.
- Un puerto USB 3.0 y 2 puertos USB 2.0.
- Wifi y bluetooth 4.0.
- Coprocesador Arduino Atmega 32u4.
- Salida de video: HDMI y MIPI-DSI.
- Conector de superposición de panel táctil integrado.
- Admite Ethernet de 100 Mbps.
- Alimentación 5V/2 A (LATTEPANDA, 2020).

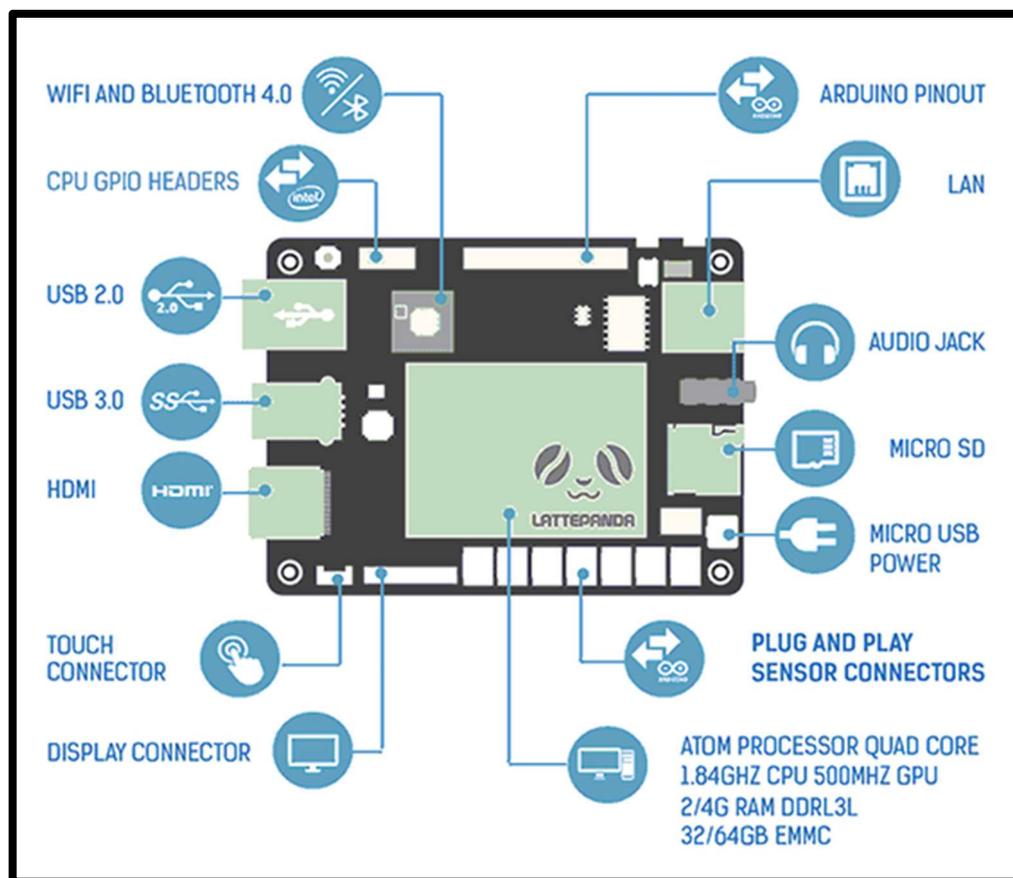


Figura 2. 19: Especificaciones técnicas (LATTEPANDA, 2020).

## 2.8 Bus I2C

El bus I2C es un estándar que se utiliza para conectar circuitos integrados (ICS). Da la factibilidad de simplificar la comunicación entre microcontroladores, dispositivos y memorias con un nivel muy elevado de inteligencia, el cual solo necesita dos líneas para realizar una comunicación efectiva donde envía la información y otra conocida como común o masa.

Este bus de comunicación fue diseñado por la compañía Philips y permite el intercambio de datos entre 2 o más dispositivos a una velocidad de 100 Kbits por segundo y en determinadas ocasiones el reloj llega hasta los 3.4 MHz (Didácticos, 2020).

La comunicación de datos del bus I2C puede ser de 2 tipos:

- Serie.
- Síncrona.

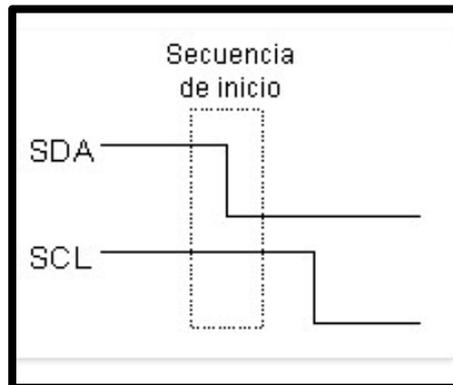


Figura 2. 20: Comunicación de datos I2C (Didácticos, 2020).

### 2.8.1 Señales del Bus I2C

A continuación, se realiza la descripción de las líneas de conexión en un bus I2C:

- SCL (System clock) esta línea genera pulsos de reloj que necesita el bus para realizar la sincronización del sistema.
- SDA (System Data) Esta línea permite realizar el transporte de los datos entre los dispositivos Esclavo/Maestro.
- GND (Masa) es el común de interconexión entre todos los dispositivos (Llamas, 2018).

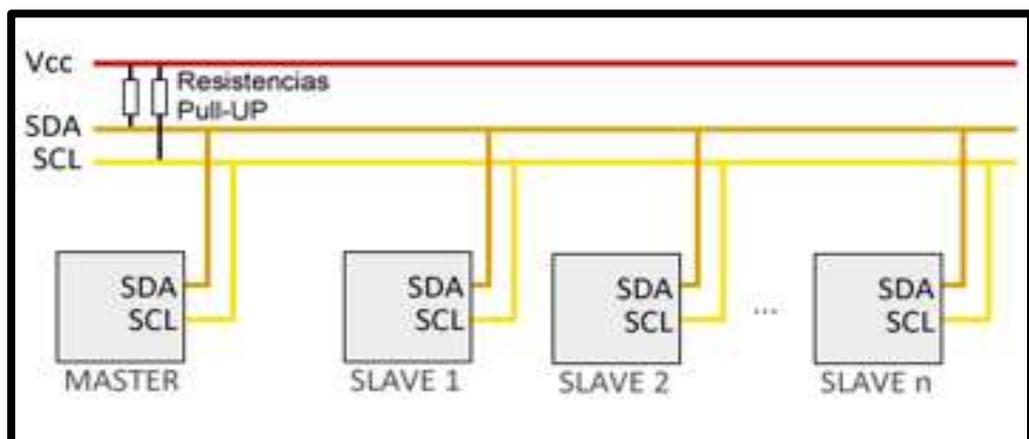


Figura 2. 21: Esquema de un Bus I2C (Llamas, 2018).

## 2.9 Protocolo de comunicación I2C

Hay que establecer los dispositivos conectados sobre el bus I2C para poder fijar una comunicación entre cada uno de ellos. Es indispensable respetar un protocolo de comunicación.

Considerando el paso más importante, se establece cuáles son los dispositivos a ser maestros y esclavos. Esto nos facilita dar prioridad de ser maestro ya que solo estos dispositivos pueden dar inicio a la comunicación (T.BEM, 2020).

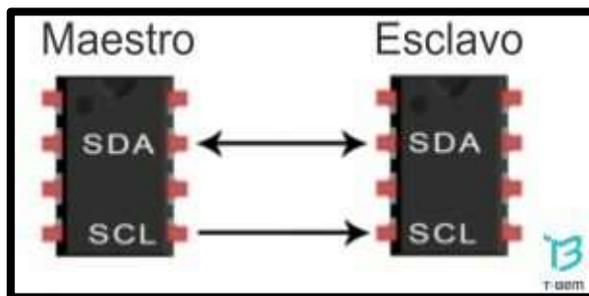


Figura 2. 22: Maestro-Esclavo I2C (T.BEM, 2020)

La condición inicial de bus liberado es cuando ambas señales están en estado lógico alto. En dicho estado cualquier dispositivo maestro que se encuentre conectado al bus puede ocuparlo dando como punto de partida una condición inicial (T.BEM, 2020).

Esta condición es presentada cuando un elemento maestro pone en estado bajo la línea de datos SDA pero deja en alto la línea de CLOCK (SCL).

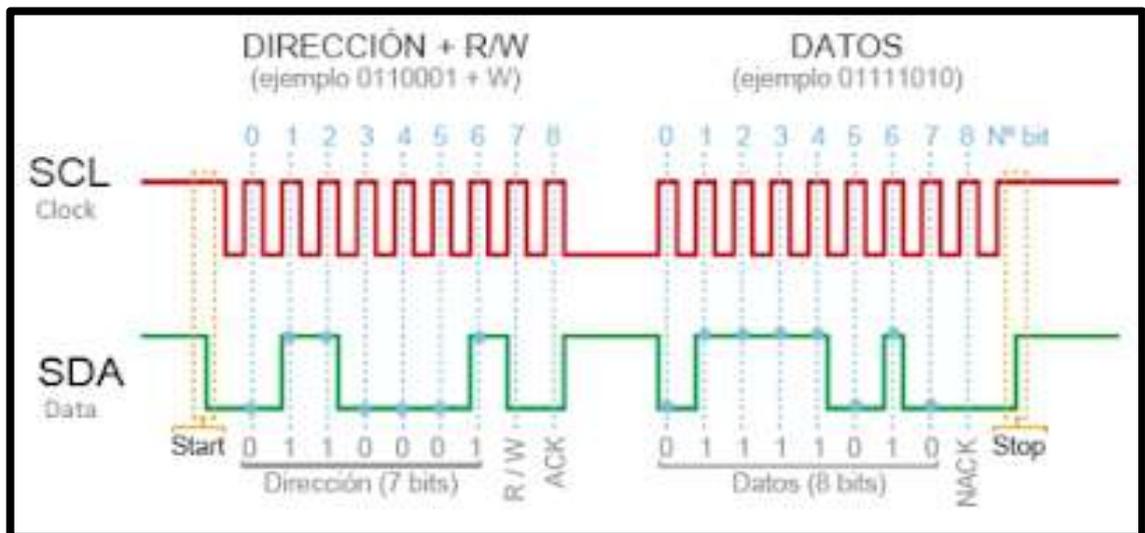


Figura 2. 23: Señales Bus I2C (electrónicos, 2020).

El primer byte es transmitido luego de la condición de inicio, la cual tiene 7 bits que integran la dirección del dispositivo que es necesario seleccionar y un octavo bit que pertenece a la operación que se requiere realizar, la cual puede ser de escritura o lectura (*electrónicos, 2020*).

Si el elemento con dirección corresponde a la dirección que se indica en los siete bits, esta debe estar presente en el bus (A0-A6) y el mismo responde con un bit en estado bajo, que se encuentra ubicado justo al lado del octavo bit que envió el dispositivo maestro.

Al bit de reconocimiento lo vamos a denominar (ACK) en estado bajo, el cual indica al maestro que el esclavo reconoce lo que se requiere y está disponible para establecer una comunicación (Didácticos, 2020).

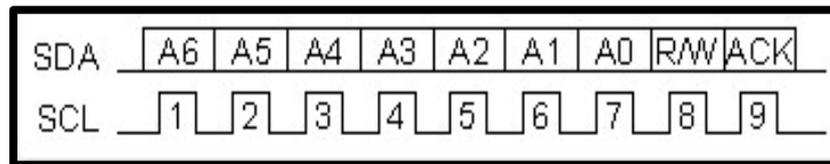


Figura 2. 24: Comunicación de dos líneas (Didácticos, 2020).

### 2.10 Definición de términos Protocolo I2C

- **Maestro (Master):** Equipo que realiza la determinación de los tiempos y dirección del tráfico en el bus. Es el único que tiene permitido la habilitación para aplicar los pulsos de reloj en la línea SCL.
- **Esclavo (Slave):** Son los equipos que se encuentren conectados al bus y no tienen permitido generar pulsos de reloj.
- **Bus libre (Bus Free):** Es el instante en el cual un dispositivo maestro puede comenzar a utilizar del bus, ya que las líneas se encuentran desactivadas.
- **Comienzo (Start):** Dicho estado se genera cuando un dispositivo maestro ocupa el bus, y comienza a generar condiciones.
- **Parada (Stop):** Un dispositivo maestro puede generar esta condición, permitiendo dejar libre al bus.
- **Dato válido:** Se determina que un estado es válido cuando un dato presente en la línea SDA no ha variado al tiempo que la línea SCL está a nivel lógico alto.
- **Reconocimiento:** Es el pulso de reconocimiento para habilitar la comunicación, conocido como ACK, se realiza ubicando la línea de datos a un nivel lógico bajo.
- **Dirección:** Todo dispositivo diseñado para funcionar en el bus tiene su propia y única dirección de acceso, que es ingresada por el fabricante (Didácticos, 2020).

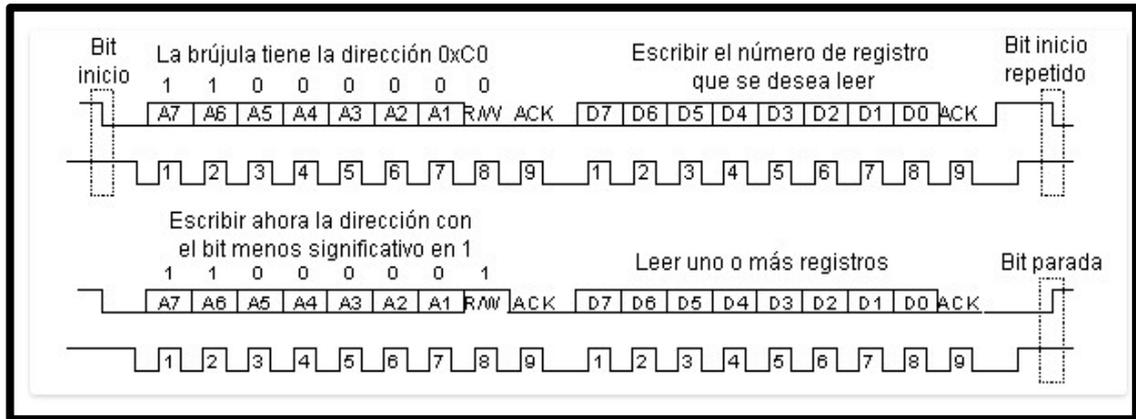


Figura 2. 25: Protocolo I2C (Didácticos, 2020).

## 2.11 Comunicación Serial

El puerto serial es un conector que en la actualidad todos los equipos PC lo integran. Esto es gracias a que el estándar del puerto serial se ha mantenido (catarina, 2020).

La norma RS-232-C escrita por la EIA (institución de normalización americana), detalla la regulación del protocolo de transmisión de los datos y toda la configuración que conlleva realizar esta comunicación entre dos o más equipos (catarina, 2020).

La comunicación que se realiza con el puerto serial es denominada asíncrona, en una comunicación síncrona la cual se precisa un bit adicional mediante el cual el emisor y receptor cambian de señal. Pero en una comunicación serial esto no es posible ya que las dos líneas están ocupadas por los datos y la tierra, por ello intercalan los datos de los estados en base al protocolo RS-232, la cual se detalla a continuación:

- **Bit de inicio:** Se requiere de un bit inicial, para que el receptor determine que la transmisión ha comenzado y dar inicio a la secuencia de datos en la línea en un tiempo y velocidad determinado.
- **Bit de parada:** Es necesario un bit de parada, el cual me indica el termino de datos transmitidos que pueden ser de 1, 1.5 y 2 bits.
- **Bit de paridad:** Para interpretar errores en la transmisión se utiliza el bit de paridad, el mismo puede ser par o impar.

### Protocolo RS-232

Existen dos tipos de conectores que funcionan mediante este protocolo el denominado DB9 (tiene 9 pines) y el DB25 (tiene 25 pines). Las señales con las que trabaja para el ingreso y salida de datos son digitales de +12V (0 lógico) y -12V (1 lógico) (catarina, 2020).

En las siguientes imágenes se muestra la configuración de los pines de cada conector, es importante hacer el reconocimiento de sus partes antes de iniciar cualquier comunicación serial (catarina, 2020).

#	Pin	E/S	Función	Conector DB 9
1			Tierra de Chasis	
2	RXD	E	Recibir Datos	
3	TXD	S	Transmitir Datos	
4	DTR	S	Terminal de Datos Listo	
5	SG		Tierra de señal	
6	DSR	E	Equipo de Datos Listo	
7	RTS	S	Solicitud de Envío	
8	CTS	E	Libre para Envío	
9	RI	S	Timbre Telefónico	

Figura 2. 26: Configuración puerto serial DB9 (catarina, 2020).

#	Pin	E/S	Función	Conector DB 25
1			Tierra de Chasis	
2	TXD	S	Transmitir Datos	
3	RXD	E	Recibir Datos	
4	RTS	S	Solicitud de Envío	
5	CTS	E	Libre para Envío	
6	DSR	E	Equipo de Datos Listo	
7	SG		Tierra de señal	
8	CD/DCD	E	Detector de Portadora	
15	TxC	S	Transmitir Reloj	
17	RxC	E	Recibir reloj	
20	DTR	S	Terminal de Datos Listo	
22	RI	S	Timbre Telefónico	
24	RTxC	S/E	Transmitir/Recibir Reloj	

Figura 2. 27: Configuración puerto DB25 (catarina, 2020).

## 2.12 Labview



Figura 2. 28: Equipos de National Instruments (instruments, 2020).

Es un software de desarrollo de ingeniería mediante programación de un lenguaje gráfico de fácil interpretación para proyectos de todo tipo, como lo son:

- Proyectos educativos a nivel de ingeniería.
- Proyectos para control y adquisición de datos en procesos industriales.
- Proyectos de robótica.
- Diseñar máquinas inteligentes o equipo industrial.
- Comprobar la eficiencia de un diseño electrónico.

El software tiene gran acogida debido a que maneja una variedad de herramientas virtuales permitiéndoles a los usuarios desarrollar proyectos ambiciosos. Otra de sus características se debe a su gran versatilidad de conexión con diferentes dispositivos llegando a ser un laboratorio virtual completo (instruments, 2020).

Ventajas:

- Interfaz gráfica flexible y simple de usar.
- El software está integrado con una plataforma universal para diseñar numerosas aplicaciones.
- Tiene una interfaz fácil para manejar variedad de iconos para adquisición de datos.

Desventaja:

- El costo es elevado en comparación con otros equipos en la industria.

- Labview es de fuente única y algunas compañías talvez no les genere tanta confianza debido a que no está estandarizada por la industria.

Labview tiene sus desventajas, pero tiene ventajas que son más efectivas y de mayor beneficio para el desarrollo y aprendizaje del usuario.

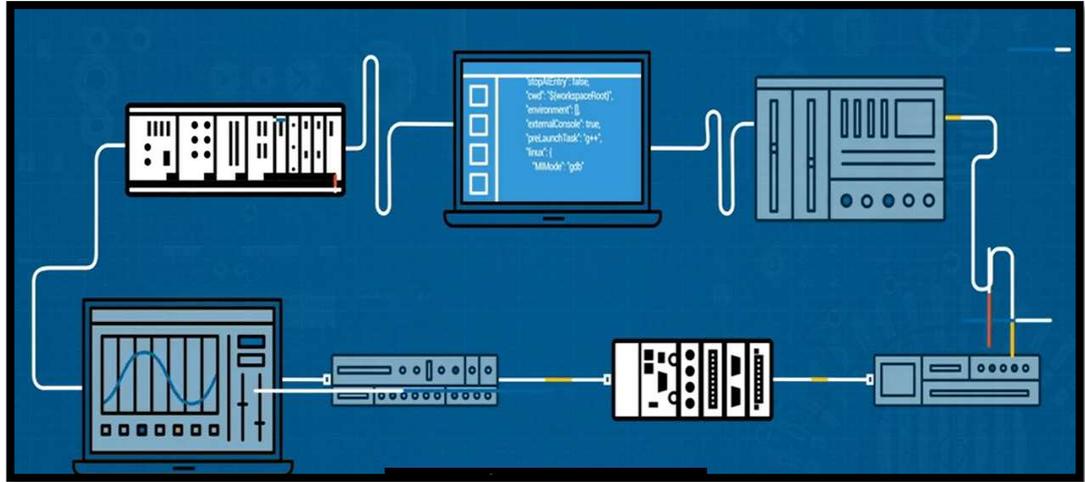


Figura 2. 29: Interconexiones con diferentes equipos NI (instruments, 2020).

### 3. Marco metodológico

#### 3.1. Introducción del proyecto.

Realizando observaciones de que no existen suficientes equipos de aplicación industrial en los laboratorios para el desarrollo de prácticas que permitan a los estudiantes disipar dudas que tengan de la teoría dada en el aula por las materias de especialización, se llegó a la conclusión de crear un proyecto de innovación y desarrollo el cual trata de la puesta en marcha de módulos prácticos que simulen procesos del sector industrial para que refuercen sus conocimientos. A continuación se detalla el proyecto.

El sistema lo conforman dos módulos didácticos utilizando tecnologías open source. El primer módulo tiene un PLC Mduino 38R con sus respectivas entradas digitales a través de pulsadores y salidas digitales mediante Leds, así como la simulación de señales analógicas por medio de potenciómetros. El segundo módulo está conformado de un PLC Mduino 19R, este cumplirá con la función de esclavo, ambos módulos contarán con elementos básicos de control como, relés, pulsadores, luces indicadoras, potenciómetros.

También incorpora la conexión de puertos I2C, USB, RS232, una pantalla HMI de 10.1". Para la programación se la realizará mediante Arduino IDE, la pantalla HMI cuenta con una tarjeta Raspberry, la cual cuenta con E/S para evitar emplear un PLC en aplicaciones simples.

En la siguiente figura se muestra un diagrama de bloques de los instrumentos que van instalados en los tableros con sus entradas y salidas que forman parte del proyecto:

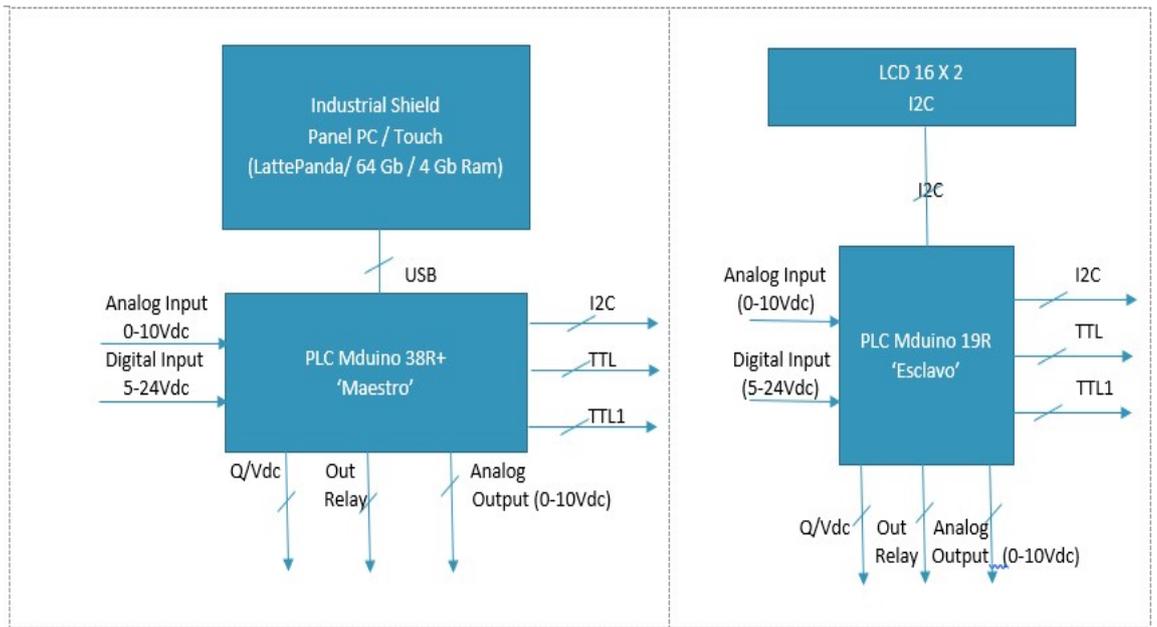


Figura 3.1: Diagrama de bloques del proyecto.

También se procedió a realizar el diseño de la planta, para una mejor interpretación. En la siguiente figura se muestra un diagrama PID de la planta, con sus respectivas siglas:

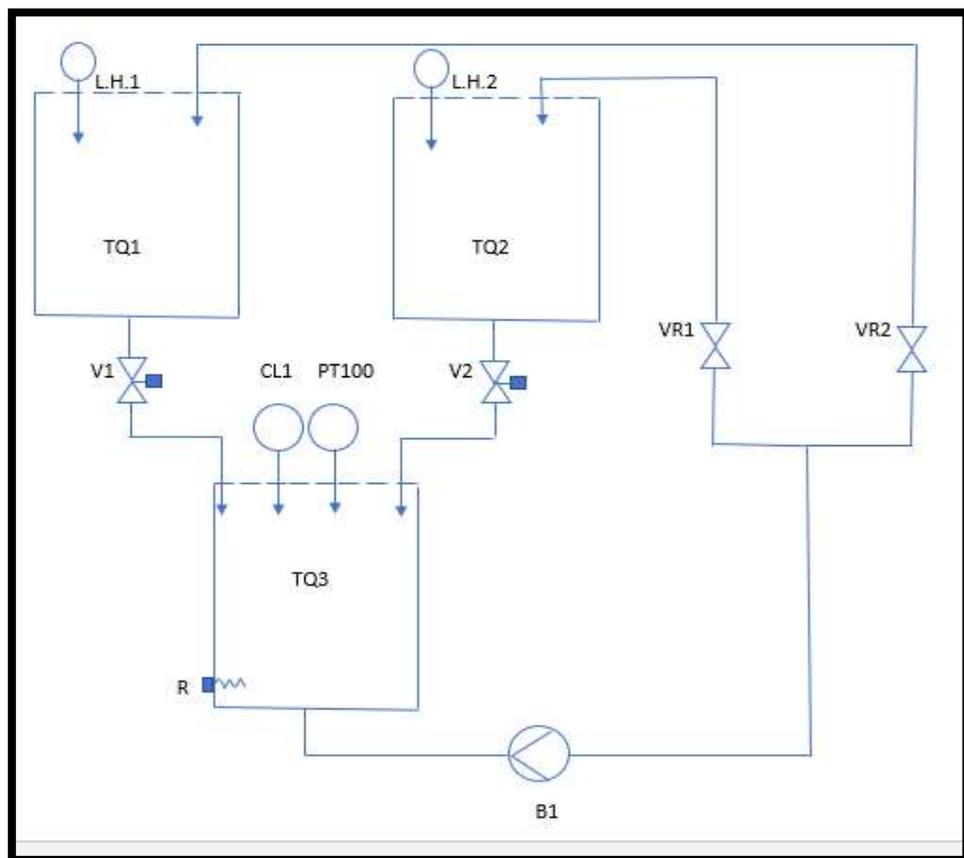


Figura 3.2: Diagrama de control

Significado de las siglas:

TQ1/TQ2: Tanques en el cual se encontrará el líquido o sustancia a mezclar.

TQ3: Tanque en el que se efectúa la mezcla.

V1/V2: Válvulas.

VR1/VR2: Válvulas de recirculación.

L.H1/L.H2: Sensor de nivel alto.

PT100: sensor de medición de temperatura.

B1: Bomba.

R: Resistencia de recalentamiento.

### 3.2. Diseño del tablero.

El primer paso a realizar es el diseño del tablero donde serán ubicados los PLC (Maestro/Esclavo). Se llegó a la conclusión de que sea una estructura galvanizada al frío. A continuación se muestra en la figura los resultados de los tableros al desnudo:



Figura 3.3: Estructura galvanizada al frío módulos maestro esclavo.

Posteriormente se realizó el diseño de las patas para mayor comodidad que parezca una mesa y así sea más provechosa para el estudiante en el laboratorio. En la siguiente figura se muestra los resultados:

Figura 3.4:  
terminada al  
esclavo.



Estructura galvanizada  
frio módulos maestro

A continuación, se procedió a realizar el pintado de los tableros para el maestro y el esclavo, quedando de la siguiente manera:



Figura 3.4: Estructura galvanizada al frio módulos maestro esclavo con acabado terminado.

### 3.3. Cableado del tablero

Para proceder a realizar el cableado, lo más importante es tener establecido la cantidad de materiales y equipos a utilizar. Por ello en la siguiente tabla los detallamos:

CANT	DESCRIPCION
16	JACKS BANANAS PARA MODULO MAESTRO (6 AQ - 2DO = 8 COLOR ROJAS Y 8 NEGRAS)
6	JACKS BANANAS PARA MODULO ESCLAVO 3 ROJAS Y 3 NEGRAS
2	SENSORES DE NIVEL TIPO BOYA CON CONTACTO N.C (PLANTA DIDACTICA)
1	PEDAZO DE PLANCHA DE 60X50 CM ACERO INOX. 2MM ESPESOR (PLANTA DIDACTICA)
3	ESTRUCTURAS CILINDRICAS DE ESTANTES CON SUS RESPECTIVOS DOBLES, Y SOLDADURA TIG. (PLANTA DIDACTICA)
2	METROS DE ANGULOS 3/4 x 1/8 pulgada para las bases de reservorios (PLANTA DIDACTICA)
4	VALVULA SOLENOIDE 24 VDC. (PLANTA DIDACTICA)
1	RELE DE ESTADO SOLIDO MARCA: Q&W , IN: 3-32 VDC / OUT : 24-480 VAC PARA CONTROL DE RESISTENCIA (PLANTA DIDACTICA)
1	RESISTENCIA PARA ESTANTE SUPERIOR EN ACERO INOXIDABLE, DE ROSCA 1 PULGADA HILO GRUESO DE 1500 W A 110 VAC (PLANTA DIDACTICA)
1	BREAKER 2P-20 A PARA FUERZA DE RESISTENCIA DE CALENTAMIENTO DEL RESERVORIO SUPERIOR DE LA PLANTA. (PLANTA DIDACTICA)
1	BREAKER 1P-2A, PARA CONTROL DE TABLERO DE PLANTA DIDACTICA
20	BORNERAS DE CONEXIÓN PARA CABLE Nº 18 Y Nº 14 (PLANTA DIDACTICA)
1	ROLLO DE CABLE FLEXIBLE Nº 18 PARA CABLEADO DE TABLERO DE PLANTA
1	VENTILADOR 12 VDC PARA HMI INDUSTRIAL SHIELD (MODULO MAESTRO)
2	METROS DE TUBERIAS DE 1/2 PULG PVC
1	ACCESORIOS PARA TUBERIAS PVC, CODO, UNIONES UNIVERSALES
4	MANIBELAS PARA LEVANTAR LA TAPA DE LOS MODULOS DIDACTICOS
1	ESTRUCTURA PARA MODULO MAESTRO EN PLANCHA GALVANIZADA AL FRIO, Y PINTURA ELECTROSTATICA.
1	ESTRUCTURA PARA MODULO MAESTRO EN PLANCHA GALVANIZADA AL FRIO, Y PINTURA ELECTROSTATICA.
1	ADQUISICION DE TARJETA LATTEPANDA 4GB/64GB ENTERPRISE
1	FUENTE DE VOLTAJE DE 24 VDC MARCA EATON, PARA RIEL DIN
1	FUENTE DE VOLTAJE DE 12 VDC MARCA EATON, PARA RIEL DIN PARA HMI INDUSTRIAL SHIELD
10	PULSANTES DE 16 mm, COLOR VERDE CAMSCO LAB 16S (MODULO MESTRO)
6	LUCES PILOTOS COLOR VERDE DE 16 mm, COLOR VERDE 12-24 VDC (MODULO MAESTRO)

12	LUCES PILOTOS COLOR VERDE DE 16 mm, COLOR VERDE 12-24 VDC (MODULO MAESTRO)
2	SELECTORES DE 16 mm, CAMSCO LAS16-2, CONTACTOS: 1 N.C/ 1 N.O
6	RELES 3P CGC MK3P-I, 24 VDC, CON SU RESPECTIVA BASE SOCKET (MODULO MAESTRO)
1	RELES 4P CGC MY4, 24 VDC, CON SU RESPECTIVA BASE SOCKET
2	PORTAFUSIBLE SASSIN DE 32A, PARA RIEL DIN (MODULO MAESTRO)
1	BREAKER STECK 2P-4A P/RIEL DIN PARA CONTROL (MODULO MAESTRO)
1	CONECTOR EXTENSOR HUB, DE 4 PUERTOS USB.
1	CABLE USB DE 2 M, PARA CARGAR EL PROGRAMA RESPECTIVO EN MODULO MAESTRO
1	CONECTOR HEMBRA DE 110 VAC PARA ALIMENTACION DE MODULO MAESTRO, PARA CHASIS
2	CAJAS DE BORNERAS PARA RIEL DIN P/CABLE Nº 18 Y Nº 14 (MODULO MAESTRO)
1	INTERRUPTOR ON/OFF DE 110 VAC, PARA ENERGIZAR MODULO MAESTRO
2	TIRAS DE 3m DE CANALETA RANURADA DE 30X30 mm para modulo maestro
5	PULSANTES DE 16 mm, COLOR VERDE CAMSCO LAB 16S (MODULO ESCLAVO)
3	LUCES PILOTOS COLOR VERDE DE 16 mm, COLOR VERDE 12-24 VDC (MODULO ESCLAVO)
1	SELECTOR DE 16 mm, CAMSCO LAS16-2, CONTACTOS: 1 N.C/ 1 N.O (MODULO ESCLAVO)
4	RELES 3P CGC MK3P-I, 24 VDC, CON SU RESPECTIVA BASE SOCKET (MODULO ESCLAVO)
1	PORTAFUSIBLE SASSIN DE 32A, PARA RIEL DIN (MODULO ESCLAVO)
1	CONECTOR HEMBRA DE 110 VAC PARA ALIMENTACION DE MODULO MAESTRO, PARA CHASIS.
1	BREAKER STECK 2P-4A P/RIEL DIN PARA CONTROL (MODULO ESCLAVO)
1	CABLE USB DE 2 M, PARA CARGAR EL PROGRAMA RESPECTIVO EN MODULO ESCLAVO
	<b>TOTAL DE MATERIALES QUE CONTIENEN AMBOS MODULOS</b>
1	M-DUINO PLC ARDUINO ETHERNET 19R I/O RELAY/ANALOG/DIGITAL PLUS 6 INPUTS 11 OUTPUTS
1	PANEL PC INDUSTRIAL EMC DE ALUMINIO, IS.TBENCL 10.1AL_RPIB3_16LSD
1	M-DUINO PLC ARDUINO ETHERNET 38R+ I/O RELAY/ANALOG/DIGITAL PLUS 12 INPUTS 22 OUTPUTS
1	SENSOR ULTRASONICO SICK UM30-213113 ,DC: 9-30 V / OUT: 4-20 ma , 0-10 vdc
1	FUENTE LOGO POWER SIEMENS DE 24 VDC 2,5A 6EP 1332 1SH43 (MODULO ESCLAVO)
2	POTENCIOMETROS LINEALES MULTIVUELTAS DE PRECISION 10 KOHM (MODULO MAESTRO)
1	POTENCIOMETROS LINEALES MULTIVUELTAS DE PRECISION 10 KOHM (MODULO ESCLAVO)

- Procedemos a colocar las canaletas internas de cada tablero (módulo maestro/esclavo)



Figura 3.5: Instalación de canaletas y módulos programables.

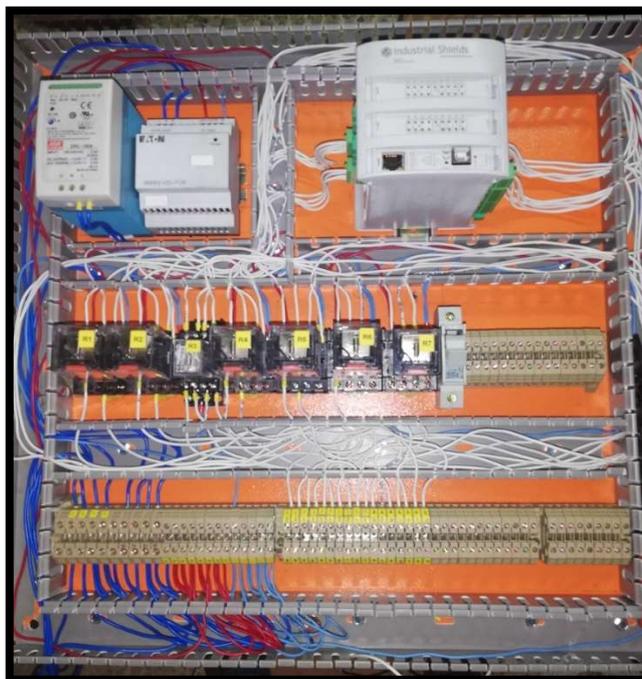


Figura 3.6: Avance del cableado en el tablero

- El paso a seguir es ubicar las luces piloto en la parte frontal del tablero para el PLC que tiene la etiqueta de esclavo, el cual podemos observar en la siguiente figura:



Figura 3.7: Vista externa del tablero ya integrado luces piloto.

Luego procedemos a ubicar las luces piloto, pulsantes, panel Touch (lattepanda) en la parte frontal del tablero del PLC que es etiquetado como MAESTRO, quedando de la siguiente manera:



Figura 3.8: Ubicación del panel en la parte externa del tablero.

Internamente fue tomando mas forma a medida que se avanzaba con el cableado de cada instrumento y equipo que lo integra, para visualizarlo mejor podemos observar la siguiente figura en la cual se encuentra

- panel Touch.
- los pulsantes.
- luces piloto.
- Potenciómetros.

Los mismos que nos permitirán realizar las distintas prácticas para el laboratorio:

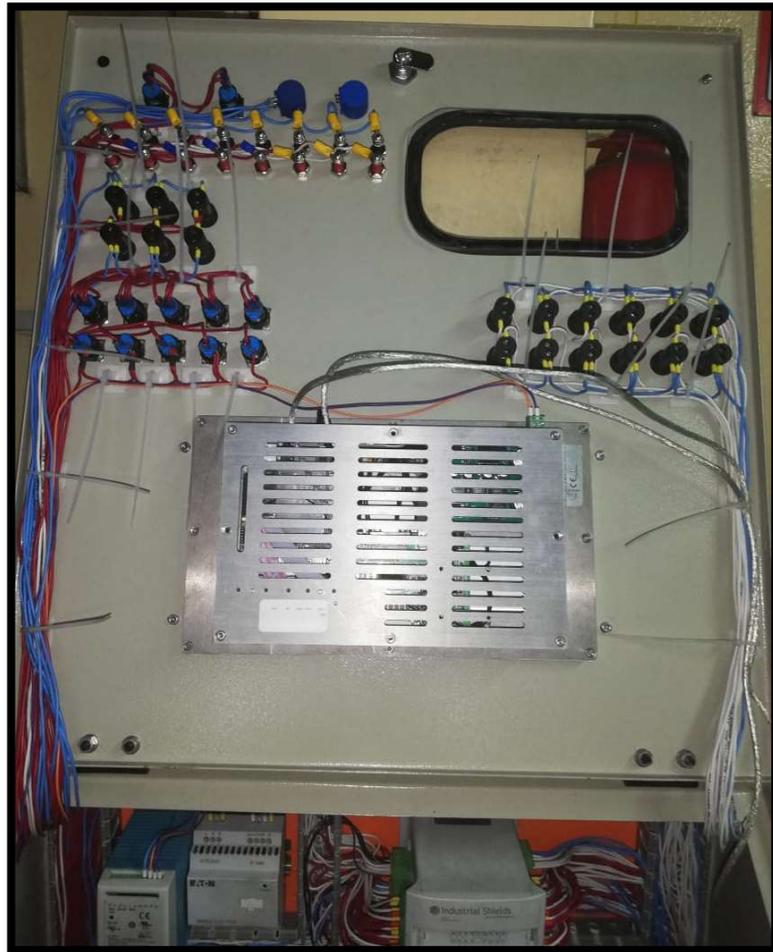


Figura 3.9: Cableado interno de las luces piloto y panel.

Al finalizar todo el cableado y ubicación de los equipos, se muestra en la figura el resultado de todo el proceso realizado:



Figura 3.10: Vista externa del tablero con sus elementos instalados.

### 3.4. Diseño de la Planta

Se realizó la conexión de tuberías para que pueda existir una recirculación del líquido que se va a utilizar en los tanques y así no exista desperdicios. Como podemos observar en la figura se capturo la parte posterior de la planta.

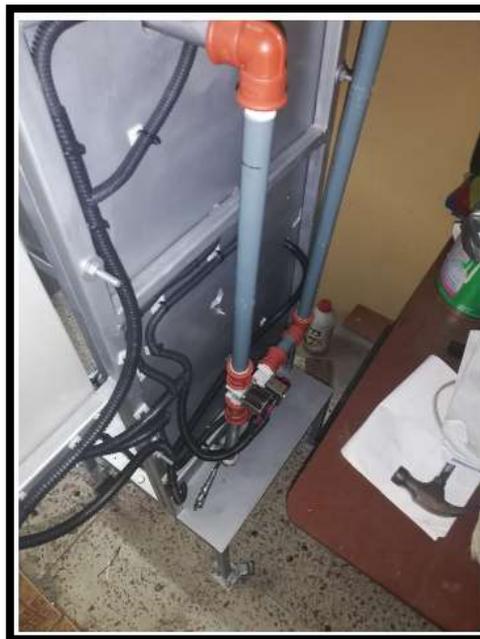


Figura 3.11: Parte posterior de la planta.

Posteriormente, se ubica las tuberías en la parte frontal con los respectivos tanques y se coloca la resistencia como observamos en la figura, el cual su funcionamiento será de calentar el líquido que contenga.



Figura 3.12: Ubicación de resistencia térmica

Se procedió a ubicar los sensores de nivel con su respectiva funda de protección para los cables de señal. Como observamos en la siguiente figura, se encuentra ubicado en la parte superior un sensor de nivel para cada tanque.



Figura 3.13: Vista frontal superior de la planta.

El cableado del tablero eléctrico para la planta, como podemos observar en la siguiente figura, contiene breakers de protección, borneras y se procedió al diseño de unas pequeñas tarjetas para regular el voltaje de entrada (0 – 10 voltios) que van a ingresar a las entradas analógicas del PLC, las cuales se encuentran ubicadas en la parte inferior del tablero.

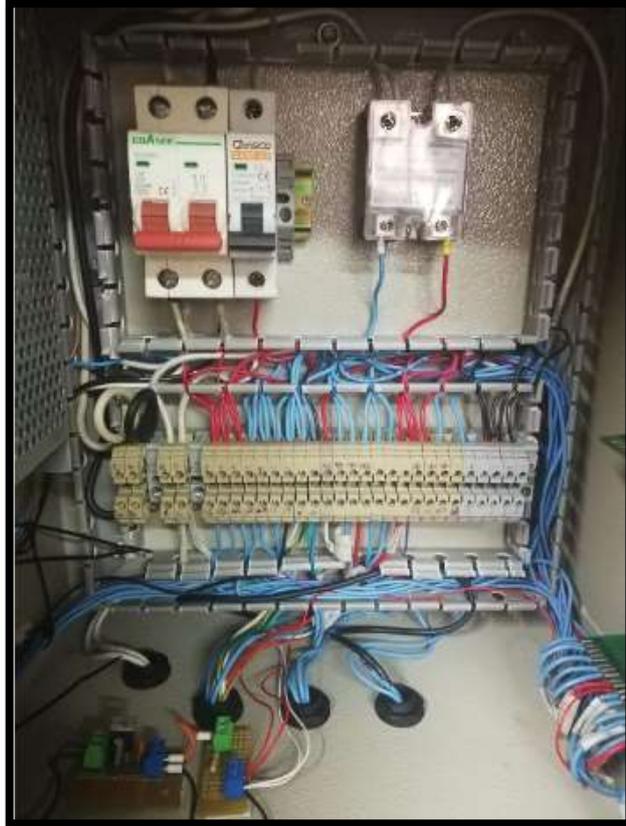


Figura 3.14: Cableado de la parte interna del tablero de control para la planta.

Como podemos observar en la siguiente figura, se muestra el layout de la tarjeta diseñada. La figura muestra la vista superior e inferior y sus respectivas mediciones.

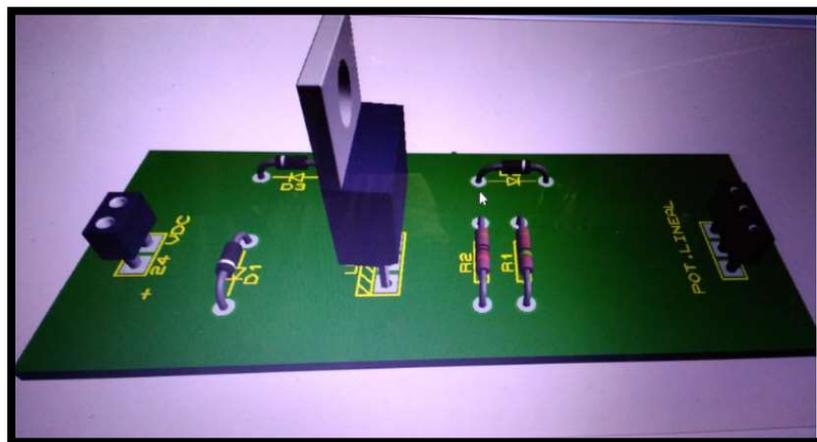


Figura 3.15: Diseño de tarjetas para regular el voltaje de entrada

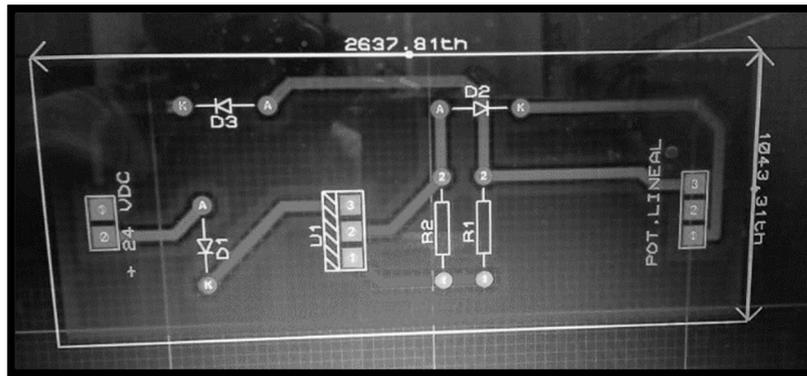


Figura 3.16: Parte posterior del diseño de la tarjeta

En la siguiente figura se puede observar ya la tarjeta impresa, soldada con los elementos respectivos y en estado óptimo para trabajar.

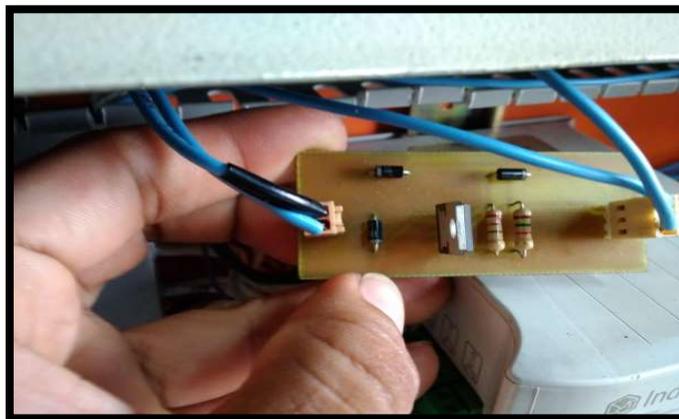


Figura 3.17: Vista real de la tarjeta

### 3.4.1. Instalación del sensor ultrasónico

El elemento más relevante es el sensor ultrasónico, el cual se compró:

PEPPERL+FUCHS

Modelo: UB800-18GM40-I-V1

Salida: 0 - 10voltios

Rango: 3.5 cm – 25 cm



Figura 3.18: Sensor ultrasónico

El sensor ultrasónico me permite realizar la medición del nivel del líquido enviando una señal al PLC el cual será interpretada por el mismo.  
Quedando ubicado como se muestra en la figura a continuación:



Figura 3.19: Ubicación del sensor en la planta

### 3.5. Interfaz Panel Touch LattePanda.

Al realizar la instalación del panel Touch, para una mayor comodidad se determinó instalarle Windows 10 ya que es un sistema operativo mayormente conocido, con ello facilitamos al usuario una navegación más confortable. A continuación, se muestra en la siguiente figura:

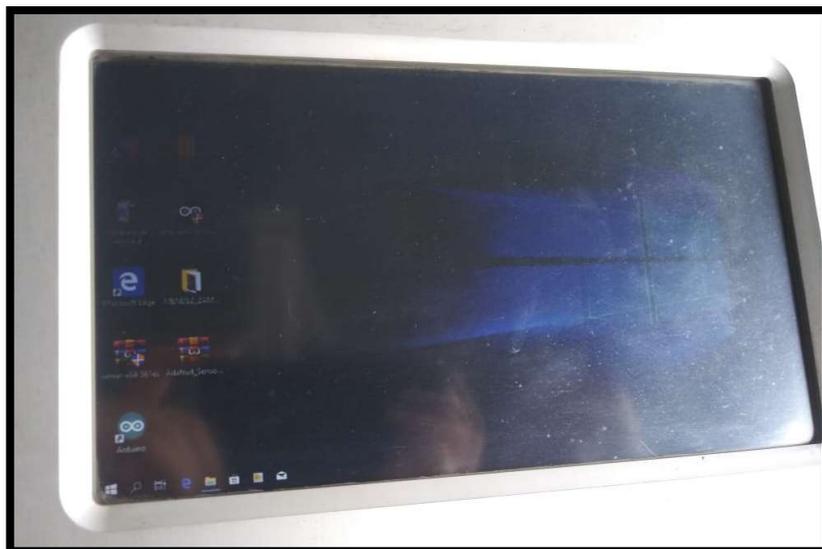


Figura 3.20: Vista del panel operativo

Por medio de la lattepanda se programó el PLC maestro y esclavo, los mismos utilizan el lenguaje de programación de arduino, permitiendo a todo estudiante de ingeniería entenderlo facilmente. En la siguiente figura, podemos observar las líneas de programación realizadas:

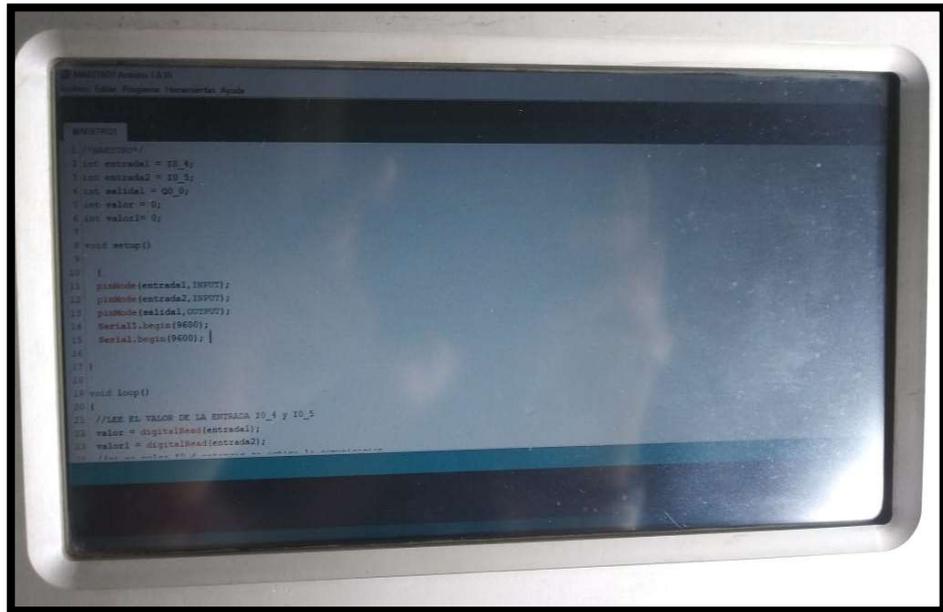


Figura 3.21: Líneas de programación de Arduino en el panel

## 4.- Resultados

### 4.1.- Prácticas

En este ítem se procede a detallar las diez prácticas realizadas con el proyecto de tesis; en formato tipo informe.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	1	○ TÍTULO PRÁCTICA: Semáforo de dos Vías	
<b>OBJETIVO:</b> 1) - - <b>OBJETIVO GENERAL.</b>  Verificar las salidas digitales del PLC Mduino38R  <b>OBJETIVOS ESPECÍFICOS:</b> - Demostrar que todas las entradas y salidas digitales están totalmente operativas - Programar el módulo de PLC arduino			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2. Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa en los módulos Mduino 38R y 19R para la ejecución de un semáforo de dos vías			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de entradas y salidas de los módulos del PLC Mduino. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

*Docente:* \_\_\_\_\_

*Firma:* \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Activación de salidas y entradas digitales	
<b>OBJETIVO:</b> <b>OBJETIVO GENERAL.</b>  Utilizar entradas y salidas digitales del módulo maestro.			
<b>OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>• Verificar el funcionamiento de todas las entradas y salidas digitales.</li> <li>• Realizar la programación para la activación en secuencia de las salidas digitales en el PLC Mduino 38R (Modulo maestro)</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1.- Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2.-Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1.- Desarrollar un programa en el módulo Mduino 38R para el reconocimiento de entradas y salidas digitales			
2.-Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de entradas y salidas digitales de los módulos del PLC Mduino. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
<b>CARRERA:</b> Ingeniería Electrónica		<b>ASIGNATURA:</b> Laboratorio de microcontroladores	
<b>NRO. PRÁCTICA:</b>	3	<b>TÍTULO PRÁCTICA:</b> Activación de entradas digitales y analógicas Modulo maestro	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b>  Verificar la correcta lectura de señales analógicas y digitales del PLC maestro  <b>- OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>• Realizar la práctica utilizando las entradas digitales y analógicas del PLC Mduino 38R+.</li> <li>• Verificar el correcto funcionamiento de todas sus variables</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1.-Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2.- Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1.- Desarrollar un programa en el módulo Mduino 38R para el reconocimiento del funcionamiento correcto en entradas digitales y analógicas que contiene el modulo maestro.			
2.- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de entradas digitales y analógicas del módulo del PLC Mduino maestro. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Activación de entradas y salidas digitales en el módulo esclavo PLC 19R	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b> Realizar la práctica para la activación de señales digitales de entrada y salida en el módulo esclavo 19R  <b>- OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>• Realizar pruebas de las entradas y salidas digitales del módulo esclavo PLC 19R</li> <li>• Verificar el funcionamiento del PLC.</li> </ul>			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):		1.- Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado. 2.- Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa en el módulo Mduino 19R para el reconocimiento del funcionamiento correcto en entradas y salidas digitales que contiene el módulo esclavo.			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de entradas y salidas digitales del módulo del PLC Mduino esclavo. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	5	TÍTULO PRÁCTICA: Comunicación serial con modulo LCD 16 x 2	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b> <ul style="list-style-type: none"> <li>- Utilizar la entrada lógica del PLC Mduino 19R y establecer comunicación i2c con un módulo LCD 16x2.</li> </ul> <b>- OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>- Establecer comunicación i2c con un módulo LCD 16 x 2</li> <li>- Utilizar la entrada lógica del PLC Mduino 19R.</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		<ol style="list-style-type: none"> <li>1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.</li> <li>2. Verificar que el cable DB25 este bien conectado para que exista la comunicación</li> </ol>	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
<ol style="list-style-type: none"> <li>1. Desarrollar un programa en el módulo MDuino 19R para el reconocimiento del funcionamiento correcto de las entradas analógicas que contiene el módulo esclavo.</li> <li>2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.</li> </ol>			
<b>RESULTADO(S) OBTENIDO(S):</b> <ul style="list-style-type: none"> <li>- Verificación del funcionamiento correcto de entradas analógicas del módulo del PLC MDuino esclavo.</li> <li>- Ampliar el conocimiento sobre programación en el entorno IDE.</li> </ul>			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaran cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Comunicación serial maestro - esclavo	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b> Utilizar los puertos de comunicación serial disponibles en los PLC 38R+ modulo maestro y 19R modulo esclavo.			
<b>OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>• Habilitar y utilizar los puertos de comunicación serial disponibles en los PLC</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2. Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa entre el módulo Mduino 19R y el 38 R para habilitar la comunicación serial de ambos, no olvidando que la versión 38R es el maestro y 19R es el esclavo.			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de la comunicación serial entre ambos PLC versión Mduino. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	7	TÍTULO PRÁCTICA: Comunicación serial esclavo - maestro	
<b>OBJETIVO:</b> - <b>OBJETIVO GENERAL.</b> Utilizar los puertos de comunicación serial disponibles en los 19R modulo maestro y PLC 38R+ modulo esclavo  <b>OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>• Habilitar y utilizar los puertos de comunicación serial disponibles en los PLC</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2. Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa entre el módulo Mduino 19R y el 38 R para habilitar la comunicación serial de ambos, no olvidando que la versión 38R es el esclavo y 19R es el maestro.			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Verificación del funcionamiento correcto de la comunicación serial entre ambos PLC versión Mduino. - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	8	TÍTULO PRÁCTICA: Utilización de señal analógica entre el modulo maestro y Planta	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b>  Obtener la señal analógica de la planta hacia el modulo maestro  <b>- OBJETIVOS ESPECÍFICOS:</b> - Reconocer la señal de la PT-100 en el módulo maestro - Verificar la activación de todas las señales que se encuentren en la planta			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2. Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa para el módulo el 38R (maestro) para reconocer la variable análoga de una PT-100			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Obtener una lectura analógica dentro de los rangos establecidos - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
CARRERA: Ingeniería Electrónica		ASIGNATURA: Laboratorio de microcontroladores	
NRO. PRÁCTICA:	9	TÍTULO PRÁCTICA: Proceso de recirculación de fluidos controlado por modulo maestro	
<b>OBJETIVO:</b> - <b>OBJETIVO GENERAL.</b>  - Ejecutar el proceso de recirculación de la planta controlado por el modulo maestro 38R  - <b>OBJETIVOS ESPECÍFICOS:</b> - Verificar la activación de la recirculación del líquido en los tanques ubicados en la planta. - Realizar el control en el módulo maestro para la recirculación en los tanques			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la práctica revisar que todos los cables y conexiones estén en buen estado.	
		2. Verificar que el cable DB25 este bien conectado para que exista la comunicación	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa para el módulo el 38R (maestro) en el cual se realice el proceso de recirculación de fluidos en la planta diseñada.			
2. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.			
<b>RESULTADO(S) OBTENIDO(S):</b> - Visualizar el funcionamiento de la planta con el control ejecutado por el PLC Mduino 38R - Ampliar el conocimiento sobre programación en el entorno IDE.			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaron cualquier duda.			
<b>RECOMENDACIONES:</b> No olvidar que tienen que activar las librerías en el entorno IDE.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
<b>CARRERA:</b> Ingeniería Electrónica		<b>ASIGNATURA:</b> Laboratorio de microcontroladores	
<b>NRO. PRÁCTICA:</b>	10	<b>TÍTULO PRÁCTICA:</b> Control de planta dosificadora/mezcladora desde el modulo maestro.	
<b>OBJETIVO:</b> <b>- OBJETIVO GENERAL.</b> <ul style="list-style-type: none"> <li>- Realizar el control de planta dosificadora/mezcladora desde el modulo maestro</li> </ul> <b>OBJETIVOS ESPECÍFICOS:</b> <ul style="list-style-type: none"> <li>- Utilizar librería R-LINK de Labview</li> <li>- Realizar el control en Labview de la planta dosificadora/mezcladora</li> <li>- Verificar la comunicación del módulo maestro</li> <li>- Realizar un control Pid de temperatura</li> </ul>			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Antes de iniciar la practica revisar las conexiones del labview con el plc maestro y esclavo	
		2. Cargar correctamente los drivers hacia cada uno de los plc a través de la librería Rlink de labview.	
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Desarrollar un programa en labview			
2. Realizar las respectivas pruebas a fin de conseguir a través de la comunicación del Rlink de labview el correcto funcionamiento de la planta .			
<b>RESULTADO(S) OBTENIDO(S):</b> <ul style="list-style-type: none"> <li>- Visualizar el control desarrollado para la planta y ejecutado por el PLC Mduino 38R</li> <li>- Ampliar el conocimiento sobre programación en el software labview.</li> </ul>			
<b>CONCLUSIONES:</b> Los estudiantes reforzaran los conocimientos aprendidos en el aula mediante las prácticas a desarrollar y despejaran cualquier duda.			
<b>RECOMENDACIONES:</b> Realizar el correcto proceso de carga en cada uno de los plc antes de dar start al proceso.			

**Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

## 5.- Conclusiones:

- Por medio del desarrollo de este proyecto , se ha obtenido conocimientos por medio de las distintas aplicaciones que se pueden dar a los dispositivos de control existentes en el mercado actual, la relación que existe entre la parte mecánica y eléctrica es fundamental porque van de la mano y están muy ligadas, para el correcto funcionamiento del proceso aplicado a los módulos didácticos que se implementó.
- Se obtuvo el conocimiento en el manejo de los equipos PLC de la compañía industrial Shields, que permite llevar al campo industrial todos los beneficios del software open Source (arduino IDE).
- Estos módulos tendrán un impacto positivo en el área de los microcontroladores, permitirá elevar el alcance de dichas materias a aplicaciones industriales con automatismos de bajo coste, robustos y flexibles.
- Un punto relevante es que el HMI instalado en el módulo maestro posee internamente una tarjeta LattePanda que hace funcionar al equipo como una PC, es decir la podemos utilizar para visualizar las variables de control a través de Labview, así como programar los módulos maestros y esclavos a través del arduino IDE.
- El amplio campo de acción en el área de automatización, le permitirá a estos módulos la capacidad de expandirse hacia futuras tesis con el objetivo de cumplir a cabalidad con el gran campo de funcionalidad que estos equipos poseen.
- La versatilidad que da Labview y Arduino con la librería R-Link en Labview permite a los estudiantes, ingenieros expandir el campo de estudio aún más.
- Al terminar el proyecto se llega a la conclusión que todos los conocimientos adquiridos durante nuestra preparación como ingenieros juegan un papel fundamental al momento de ponerlos en práctica, materias como microcontroladores, automatización industrial e instrumentación se unieron para darle vida a este humilde proyecto.

## 6.- Recomendaciones:

Al utilizar los módulos del proyecto, se recomiendan lo siguiente:

- Al momento de encender el módulo maestro y se desee trabajar con el HMI se deberá accionar el segundo pulsante que posee internamente en su estructura, esta característica vino de fábrica.
- Si se desea trabajar con módulos externos usar los plugs adecuados y cuidar que las salidas de control no sobrepasen los 0.60 mA .de corriente, de ser así usar módulos de relays.

- Cada 3 meses se recomienda limpiar los filtros de las válvulas de descarga 1 y 2 del tanque principal de la planta, con el fin de eliminar residuos que impidan la circulación de agua adecuada.
- El nivel de agua inicial del tanque principal debe estar en 2 cm por arriba de la resistencia a fin de evitar falsas lecturas.
- Los valores de calibración del sensor de temperatura y nivel se encuentran en la cajonera del PLC maestro en sus respectivos manuales.
- Todos los software se encuentran instalados en el HMI, si se desea trabajar desde un equipo externo los instaladores de los programas están en el escritorio en una carpeta de nombre (programas open Source).
- Conectar los módulos DB25 adecuadamente en cada práctica, revisar que los tornillos de sujeción estén correctamente conectados.
- Revisar los manuales de los equipos que se encuentran en la respectiva cajonera de cada módulo, así como las especificaciones técnicas de los equipos de instrumentación.
- Si se desea programar los PLCs desde el HMI, conectar el teclado y mouse en el conector ubicados en la parte posterior derecha del módulo, el teclado y mouse se encuentran en la cajonera del módulo maestro.
- Como mejoras al proyecto, se podría implementar una válvula de descarga motorizada a fin de evacuar el líquido de los tanques mediante un control PID.

## **7.- Anexos**

## **ANEXOS**

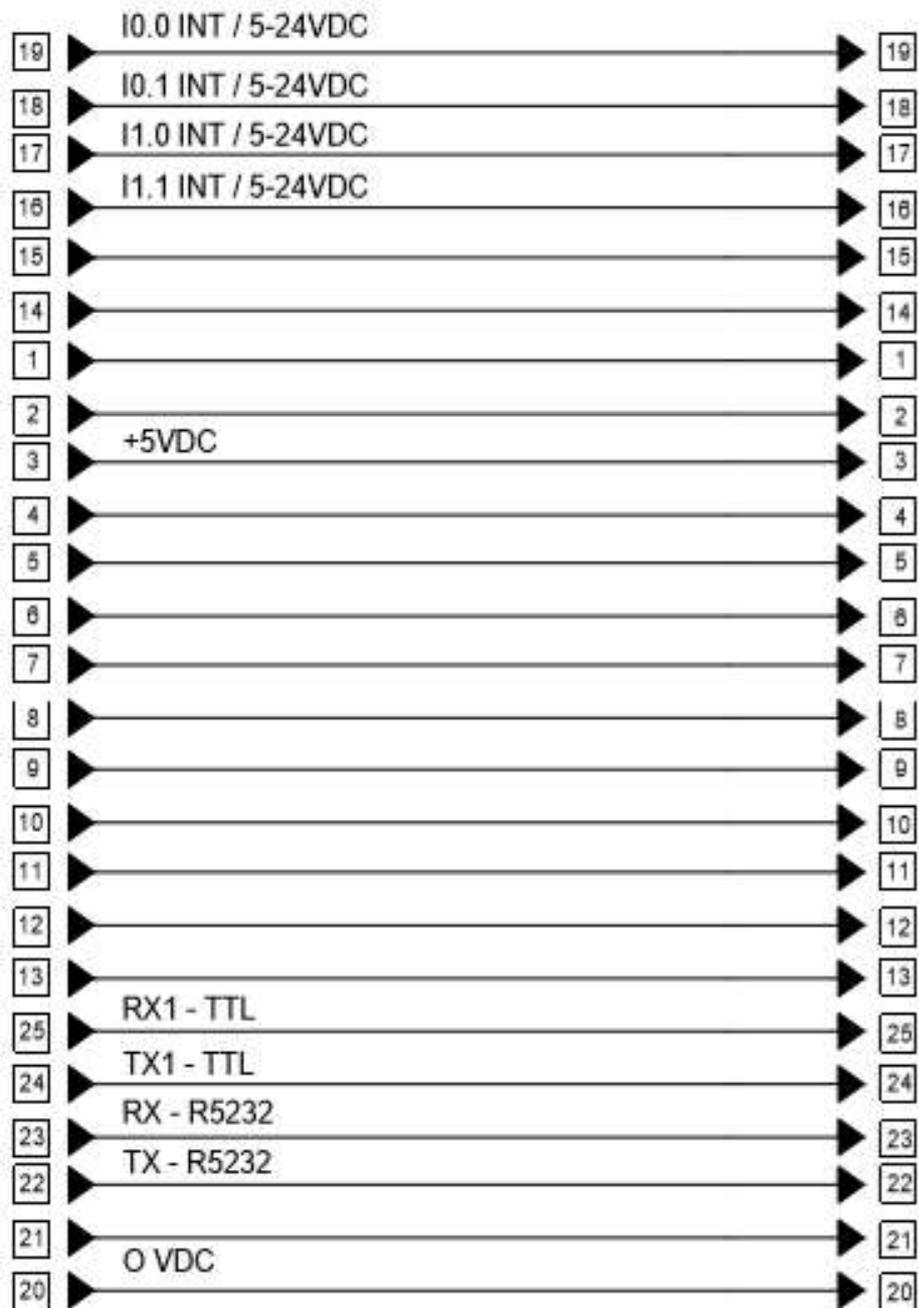
**ANEXO 1**  
**GASTOS DEL PROYECTO**

item	Descripcion	unidades	valor	total
1	PLC Mduino 38 R+	1	750	750
2	PLC Mduino 19 R	1	450	450
3	HMI Touchberry Pi	1	650	650
4	Tablero Mestro	1	300	300
5	Tablero esclavo	1	250	250
6	elementos de control relays, borneras, luces piloto , boto	1	300	300
7	fuentes de voltaje de 12 vdc para riel din	3	80	240
8	fuentes de voltaje de 24 vdc para riel din	1	55	55
9	fuentes conmutadas de 12 vdc a 5 amp	2	20	40
10	elementos de control planta	1	200	200
11	control de temperatura : pt100 , transductor y conversor	1	280	280
12	elementos de control de nivel ultrasonico y conversor	1	330	330
13	tanques de acero inoxidable y estructura de planta	1	280	280
14	solenoides a 12 vdc	2	15	30
15	valvulas solenoides a 120 vdc	2	148	296
16	varios	1	100	100
			<b>total</b>	<b>4551</b>

**ANEXO 2**

**PLANOS ELÉCTRICOS Y DE CONTROL**

## MODULO MAESTRO 38R CONECTOR DB25 MAESTRO - ESCLAVO



UNIVERSIDAD POLITECNICA SALESIANA

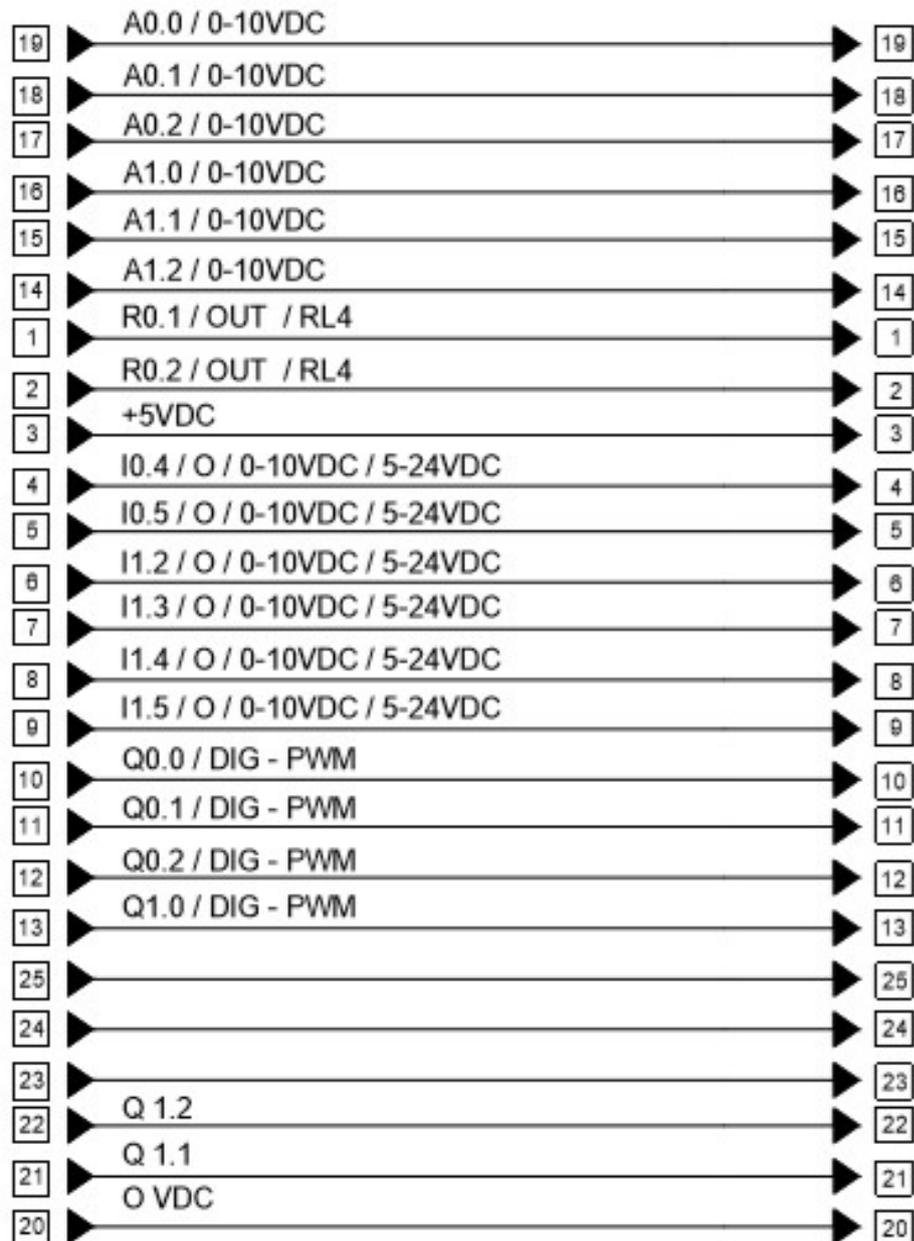
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

## MODULO MAESTRO 38R CONECTOR DB25 MAESTRO - PLANTA



UNIVERSIDAD POLITECNICA SALESIANA

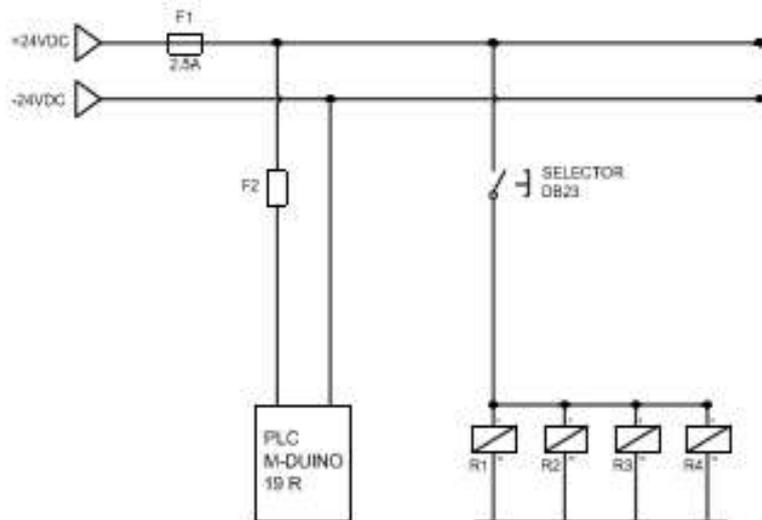
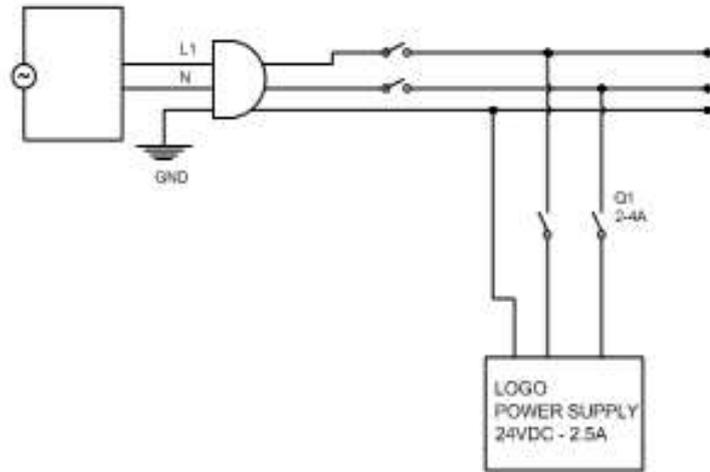
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

# CONEXIONES PLC ESCLAVO CONTROL ELECTRICO



UNIVERSIDAD POLITECNICA SALESIANA

CARRERA: INGENIERÍA ELECTRÓNICA

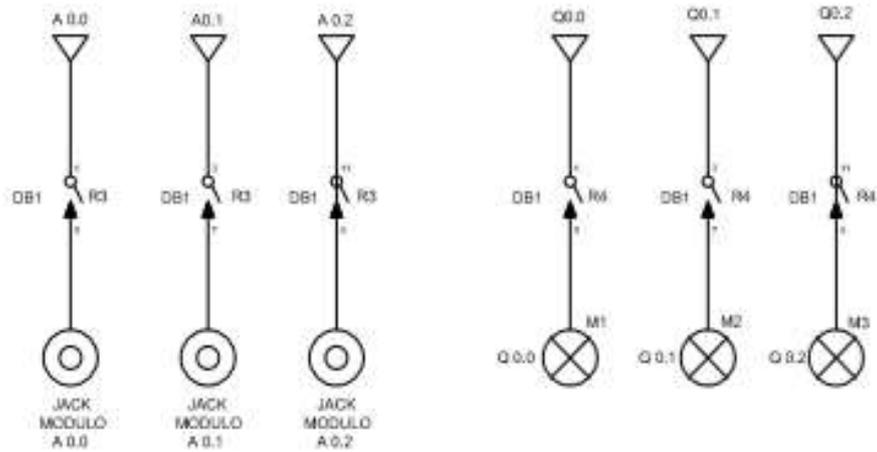
PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

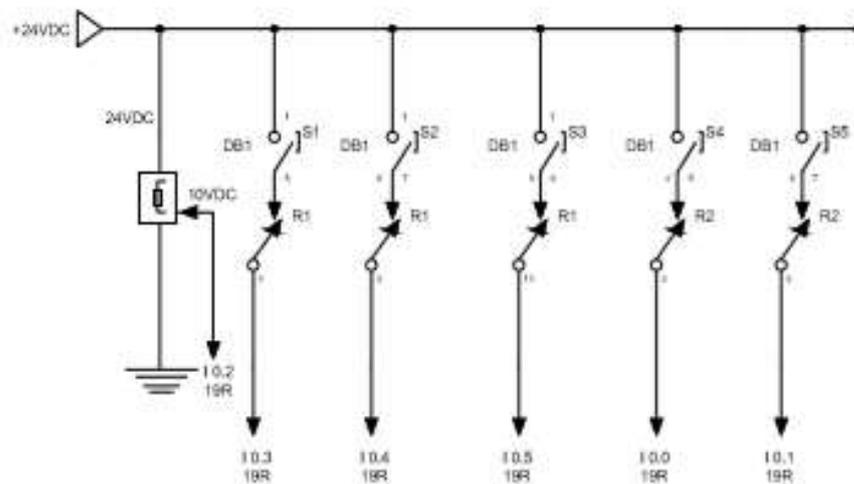
KIARA ZAMBRANO

# ENTRADAS Y SALIDAS DIGITALES MODULO ESCLAVO/ PLC M-DUINO 19R

## SALIDAS ANALÓGICAS/ DIGITALES



## ENTRADAS DIGITALES/ANALÓGICAS

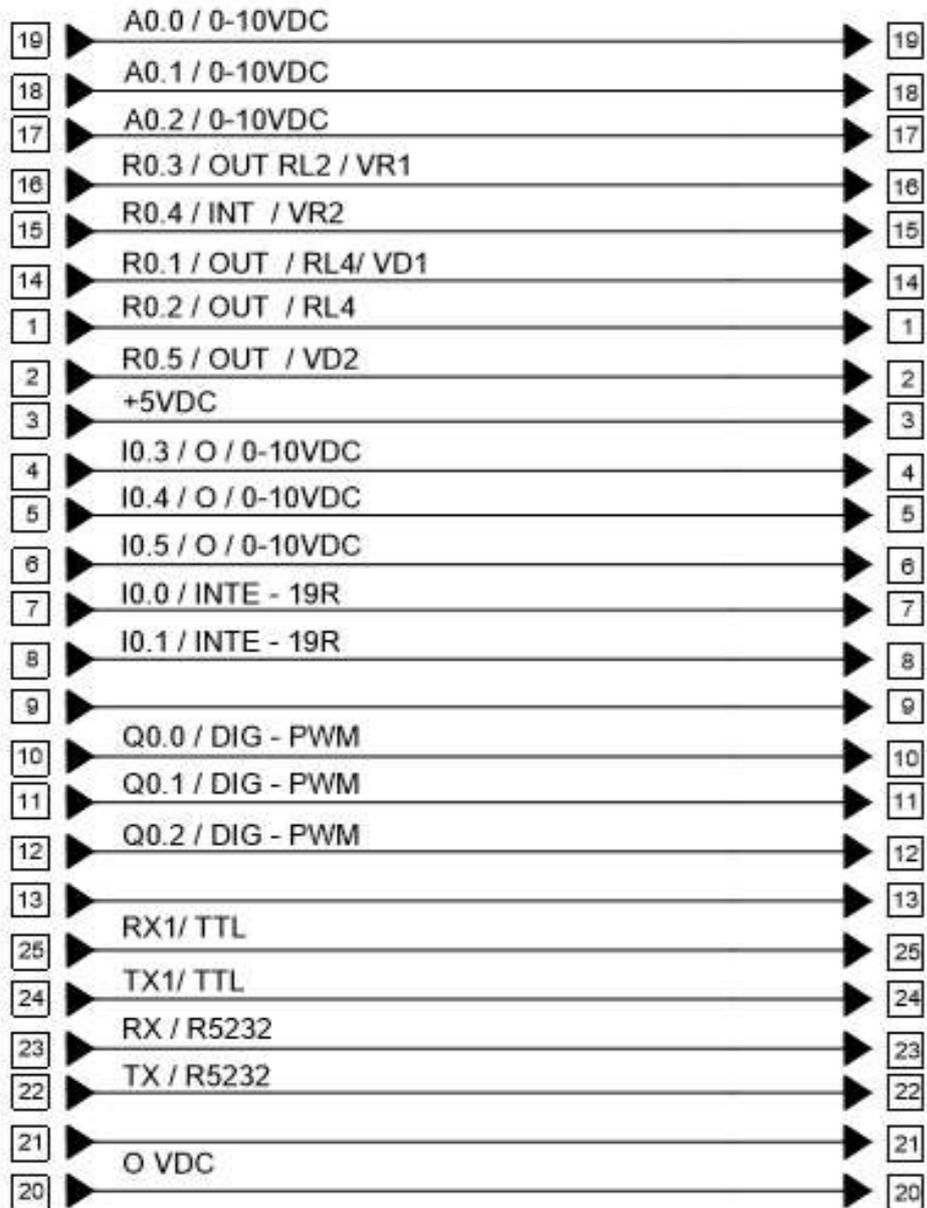


UNIVERSIDAD POLITECNICA SALESIANA

CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:  
VÍCTOR LUIS MÉNDEZ  
KIARA ZAMBRANO

## MODULO ESCLAVO 19R CONECTOR DB25



UNIVERSIDAD POLITECNICA SALESIANA

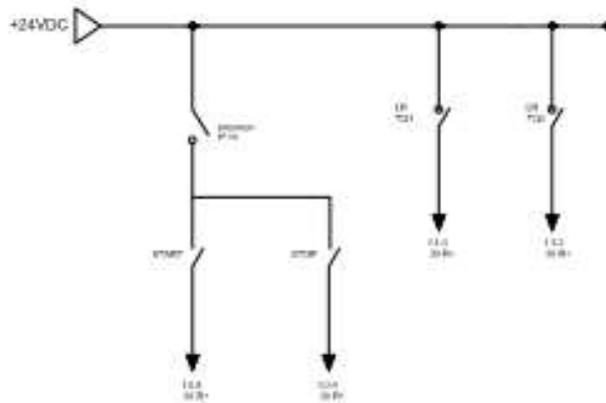
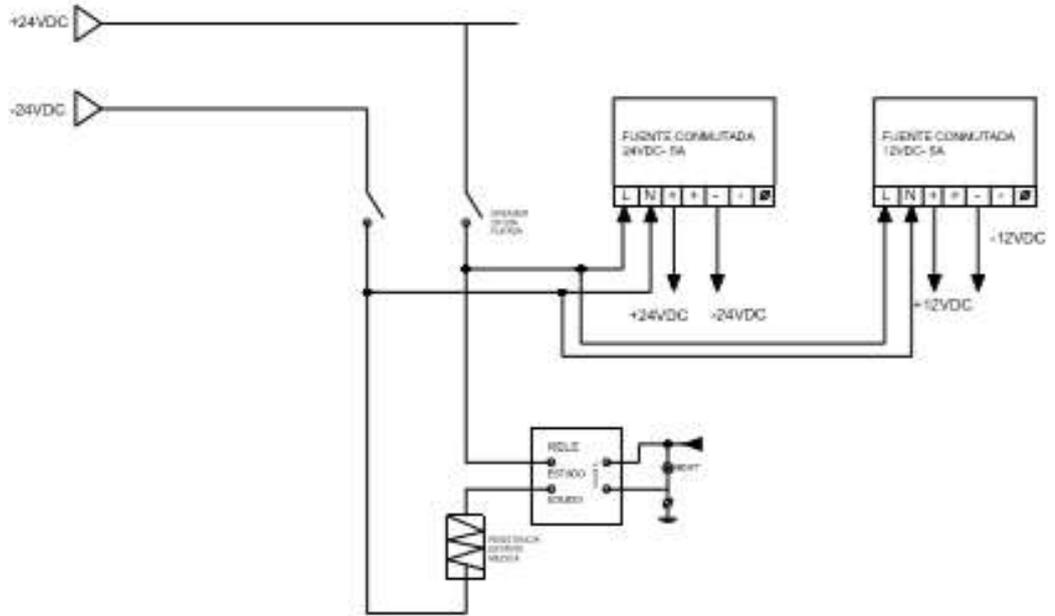
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

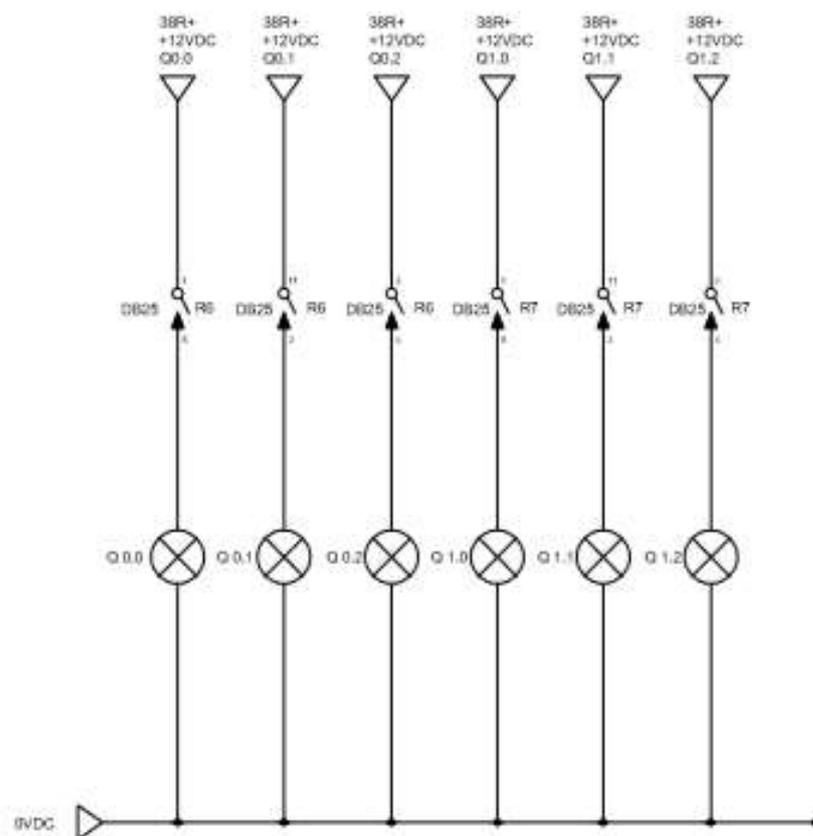
# PLANO CONTROL INTERNO/ PLANTA DOSIFICACION



UNIVERSIDAD POLITECNICA SALESIANA  
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:  
VÍCTOR LUIS MÉNDEZ  
KIARA ZAMBRANO

## DIAGRAMA CONEXIONES LUCES PILOTO



### LUCES PILOTO

Q 0.0= LUZ ROJA 1  
 Q 0.1= LUZ AMARILLA 1  
 Q 0.2= LUZ VERDE 1  
 Q 1.0= LUZ ROJA 2  
 Q 1.1= LUZ AMARILLA 2

UNIVERSIDAD POLITECNICA SALESIANA

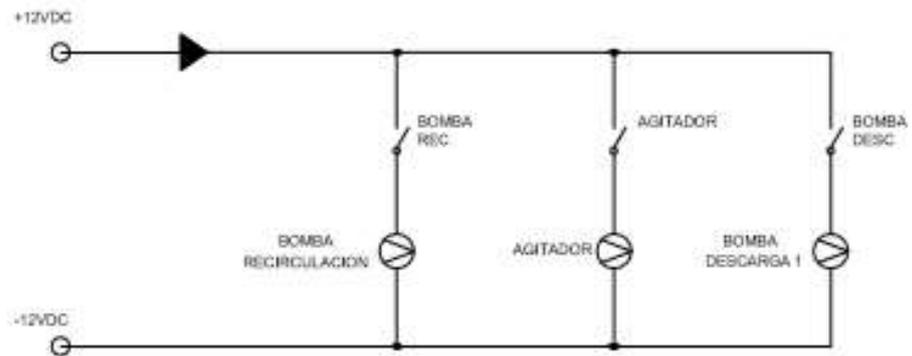
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

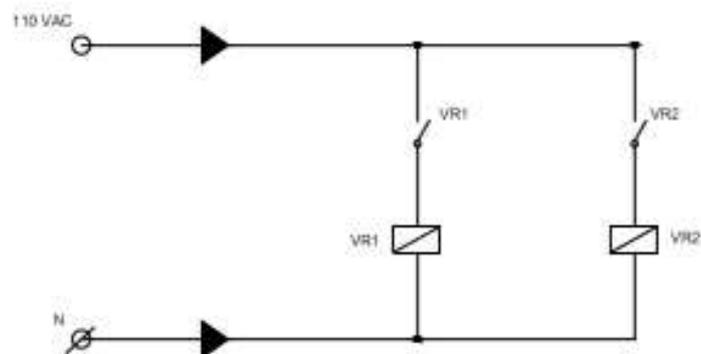
VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

## FUERZA / ELEMENTOS PLANTA 12 VDC



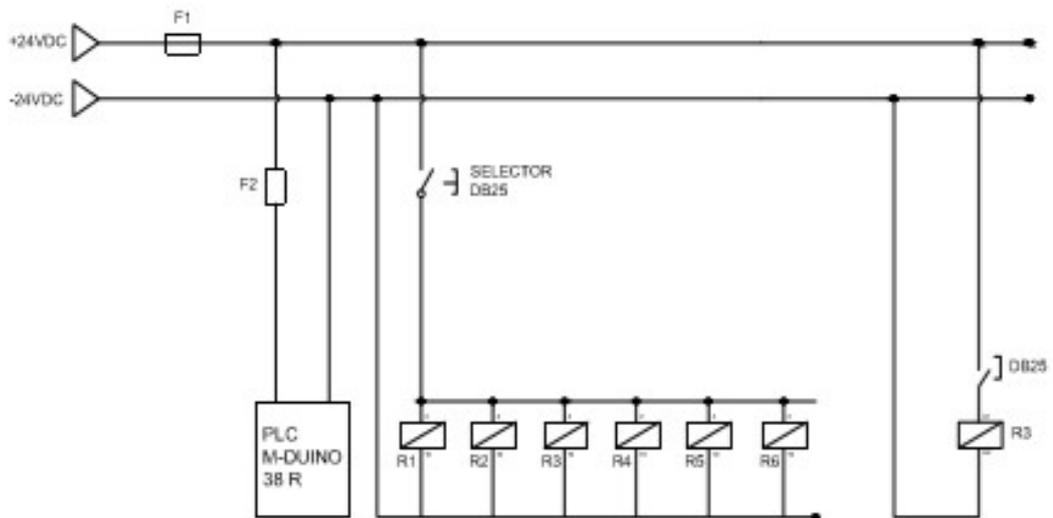
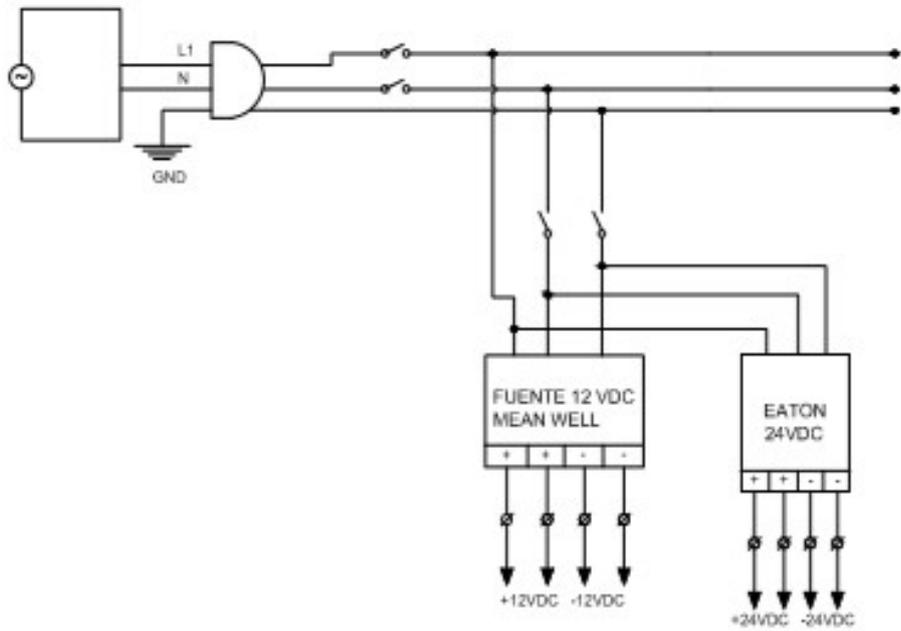
## FUERZA / ELEMENTOS SELENOIDES VR1 Y VR2



UNIVERSIDAD POLITECNICA SALESIANA  
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:  
VÍCTOR LUIS MÉNDEZ  
KIARA ZAMBRANO

# CONTROL INTERNO MODULO MAESTRO 38R



UNIVERSIDAD POLITECNICA SALESIANA

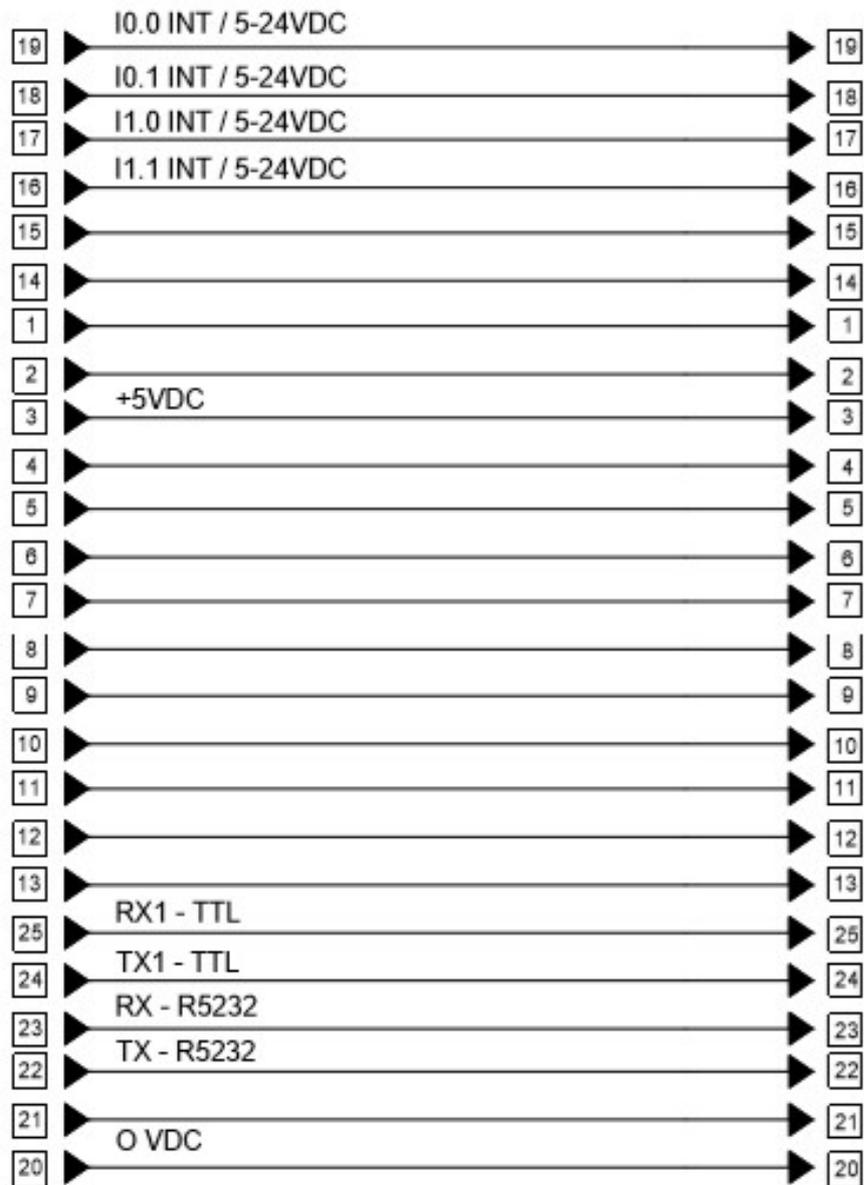
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

## MODULO MAESTRO 38R CONECTOR DB25 MAESTRO - ESCLAVO



UNIVERSIDAD POLITECNICA SALESIANA

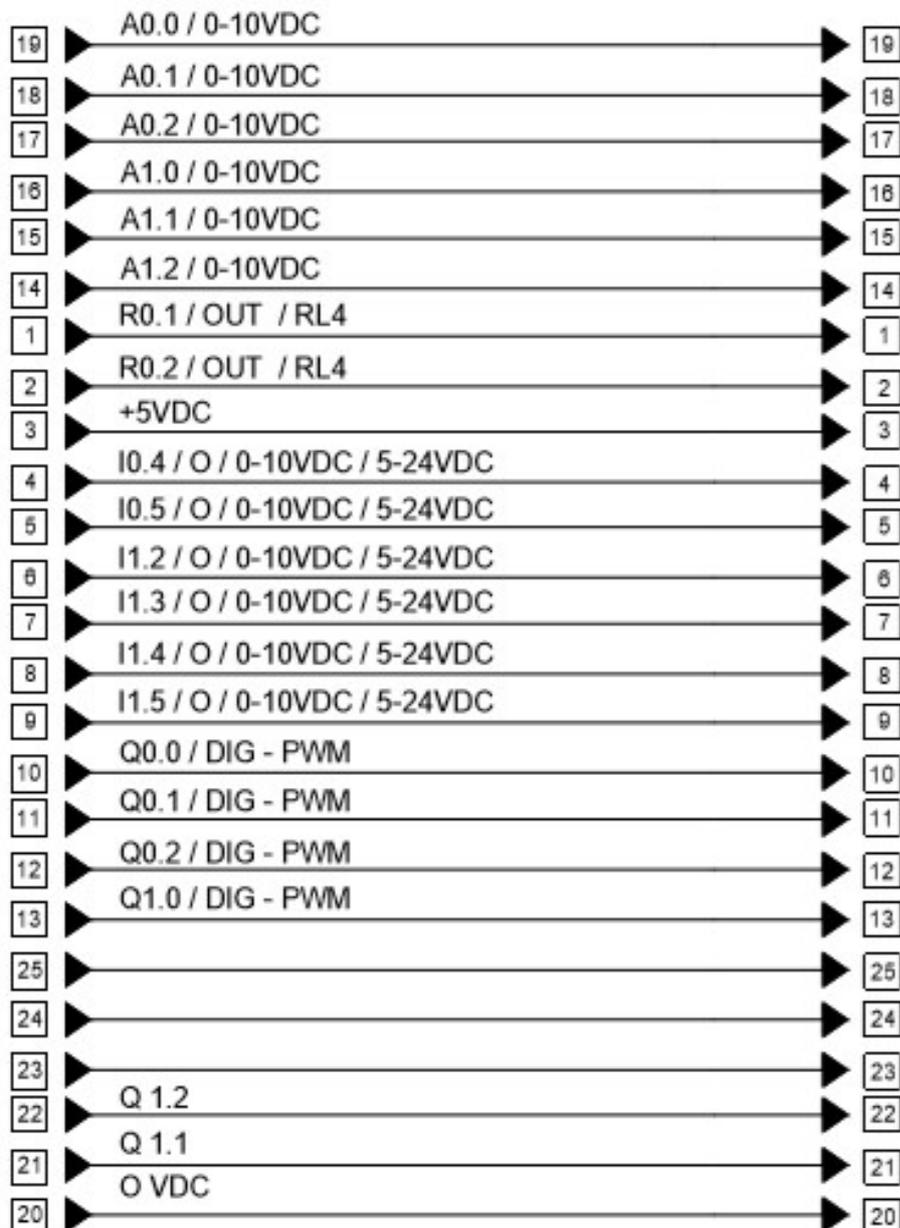
CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

## MODULO MAESTRO 38R CONECTOR DB25 MAESTRO - PLANTA



UNIVERSIDAD POLITECNICA SALESIANA

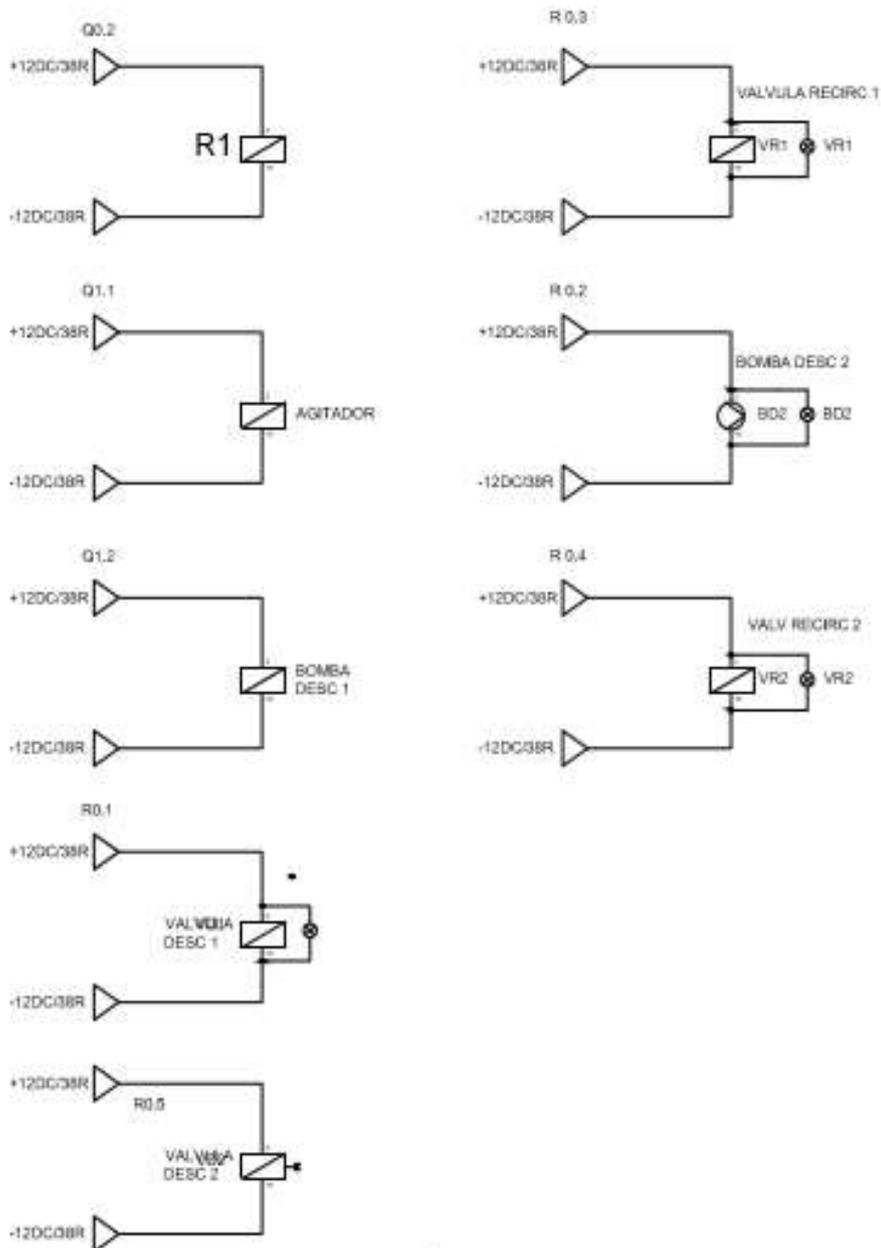
CARRERA: INGENIERIA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

## CONTROL INTERNO MODULO MAESTRO 38R



UNIVERSIDAD POLITECNICA SALESIANA

CARRERA: INGENIERÍA ELECTRÓNICA

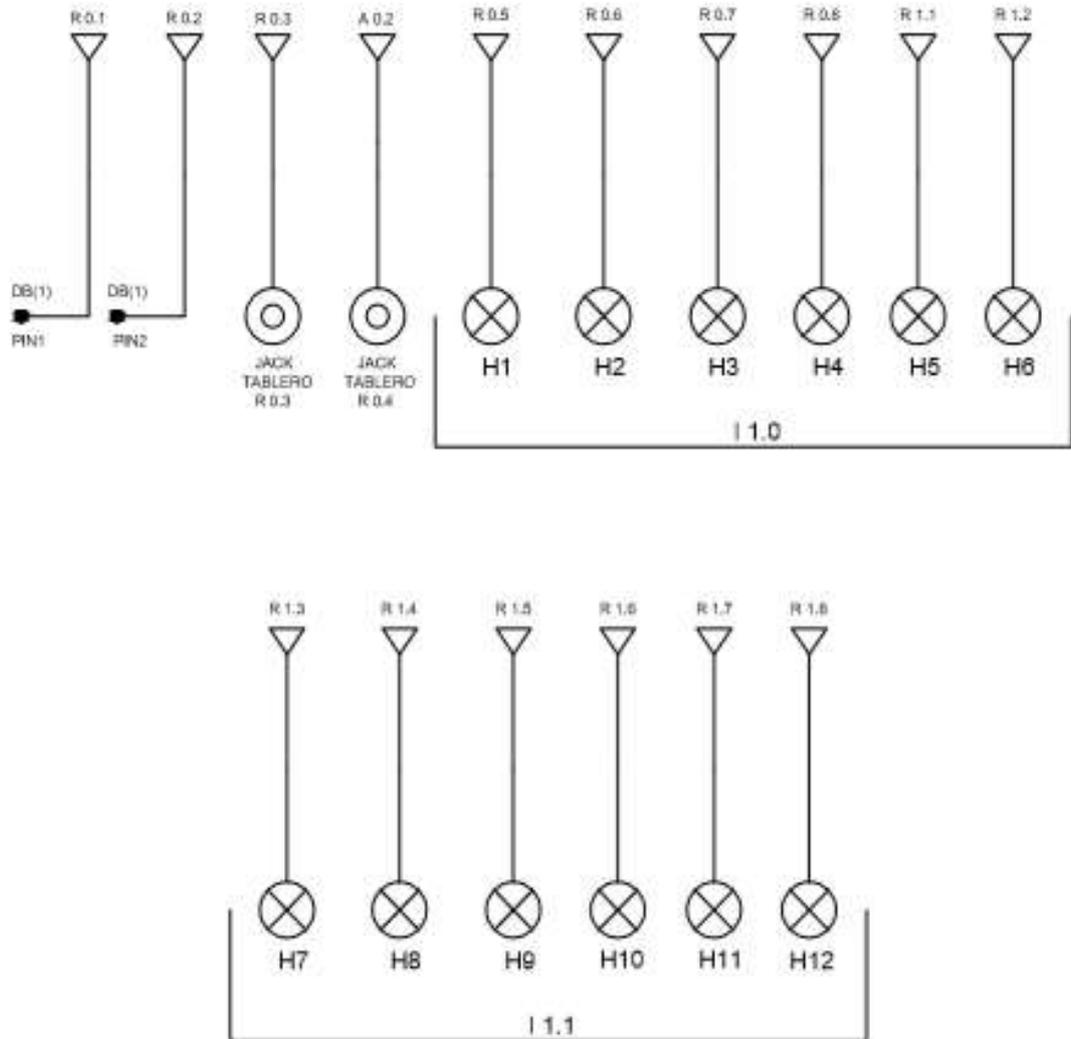
PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

# MODULO MAESTRO/ PLC M-DUINO 38R+

## SALIDAS RELE 220VAC/ 110VAC



UNIVERSIDAD POLITECNICA SALESIANA

CARRERA: INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN:

VÍCTOR LUIS MÉNDEZ

KIARA ZAMBRANO

**ANEXO 3**  
**INFORME DE PRÁCTICAS**

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 1

TEMA:

SEMAFORO DE DOS VÍAS

- **OBJETIVO GENERAL:**

Verificar las salidas digitales del PLC Mduino38R

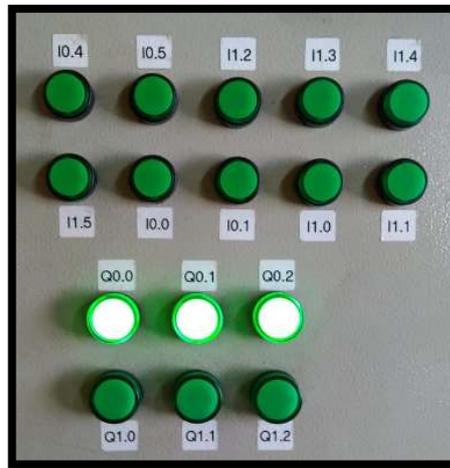
- **OBJETIVOS ESPECIFICOS:**

Demostrar que todas las salidas digitales están totalmente operativas

- **ACTIVIDAD.-**

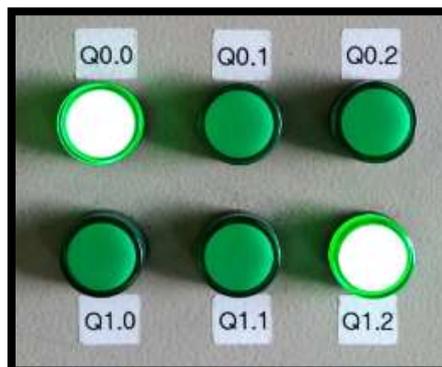
- Desarrollar un programa en los módulos Mduino 38R y 19R para la ejecución de un semáforo de dos vías
- Verificación del funcionamiento correcto de entradas y salidas de los módulos del PLC Mduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**



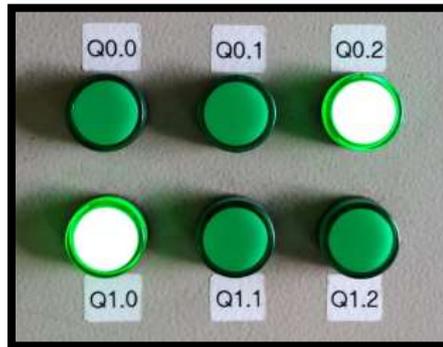
La secuencia del programa está determinada de la siguiente manera:

- Inicialmente se activa la luz roja1 (la cual es interpretada por el PLC como Q0.0) y la luz verde2 procede a encenderse (es interpretada por el PLC como Q1.2).

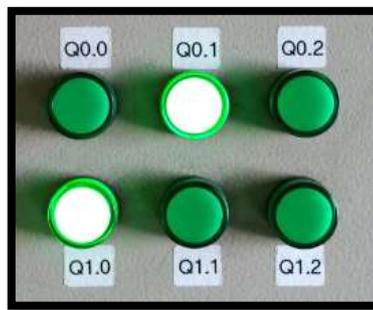


- Luego de transcurrir 4 segundos, se apaga la luz verde2 (Q1.2), y se activa la luz amarilla2 (Q1.1) por un lapso determinado de 4 segundos.
- Posteriormente transcurrido estos 4 segundos, se apaga la luz roja1 (Q0.0), y se activa (modo ON) la luz verde1 (Q0.2).

- La siguiente secuencia debe apagarse la luz amarilla2 (Q1.1) y se activa la luz roja2 (Q1.0).



- Seguido de un tiempo posterior de 4 segundos, se apaga la luz verde1 (Q0.2), y se activa la luz roja2 (Q1.0) consecuentemente se enciende la luz amarilla1 (Q0.1). Transcurrido el mismo intervalo de tiempo se apaga la luz roja2 (Q1.0) y la luz amarilla1 (Q0.1).



- **MARCO PROCEDIMENTAL**

Diseñar un programa que me permita ejecutar la secuencia de un semáforo por medio del PLC Mduino.

**PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO**

Se procede a declarar salidas digitales:

- Q0.0 indica (LUZ ROJA1)
- Q0.1 indica (LUZ AMARILLA1)
- Q0.2 indica (LUZ VERDE1)
- Q1.0 indica (LUZ ROJA2)
- Q1.1 indica (LUZ AMARILLA2)
- Q1.2 indica (LUZ VERDE2)

Se procede a declarar las variables de salidas con las siguientes denominaciones:

- salida1 = Q0.0
- salida2 = Q0.1
- salida3 = Q0.2
- salida4 = Q1.0
- salida5 = Q1.1
- salida6 = Q1.2

```
practica_1
void loop() {
21 // put your main code here, to run repeatedly:
22 digitalWrite(salida1, HIGH);
23 digitalWrite(salida6, HIGH);
24 delay(4000);
25 digitalWrite(salida1, HIGH);
26 digitalWrite(salida6, LOW);
27 digitalWrite(salida5, HIGH);
28 delay(4000);
29 digitalWrite(salida1, LOW);
30 digitalWrite(salida3, HIGH);
31 digitalWrite(salida5, LOW);
32   digitalWrite(salida4, HIGH);
33 delay(4000);
34 digitalWrite(salida3, LOW);
35   digitalWrite(salida4, HIGH);
36     digitalWrite(salida2, HIGH);
37     delay(4000);
38     digitalWrite(salida4, LOW);
39     digitalWrite(salida2, LOW);
40 }
```

```
Prueba1 $
1 int entrada1 = I0_4; // I0.4 ENTRADA DIGITAL - ANALOGICA
2 int salida1 = Q0_2;
3 int salida2 = Q1_0;
4 int salida3= Q1_1;
5 int salida4=R0_1;
6 int salida5=R0_2;
7 int valuel = 0;
8
9 void setup() {
10 // put your setup code here, to run once:
11 pinMode(salida1, OUTPUT);
12 pinMode(salida2, OUTPUT);
13 pinMode(entrada1, INPUT);
14 }
15 void loop() {
```

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión ethernet.

## CONCLUSIONES:

El haber realizado esta práctica permite al programador interactuar con un lenguaje fácil de interpretar como lo es Arduino, además de reconocer e interactuar con todas las salidas digitales de los PLC de

la versión Mduino.

**RECOMENDACIONES:**

Tener en cuenta para que el ciclo se vuelva a repetir se utiliza el lazo VOID LOOP.

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 2

TEMA:

ACTIVACIÓN DE SALIDAS Y ENTRADAS DIGITALES

- **OBJETIVO GENERAL:**

- Utilizar entradas y salidas digitales del módulo maestro.

- **OBJETIVOS ESPECIFICOS:**

- Verificar el funcionamiento de todas las entradas y salidas digitales.
- Realizar la programación para la activación en secuencia de las salidas digitales en el PLC Mduino 38R (Modulo maestro)

- **ACTIVIDAD.-**

- Desarrollar un programa en el módulo Mduino 38R para el reconocimiento de entradas y salidas digitales
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

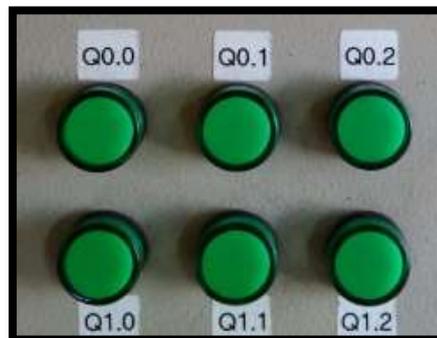
- **DESARROLLO.-**

La secuencia del programa es la siguiente:

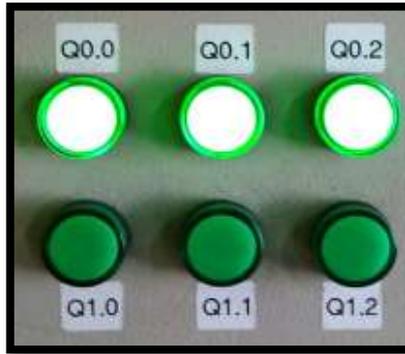
- Si se pulsa la entrada digital I0.4, se activarán las salidas Q0.0, Q0.1 Y Q0.2



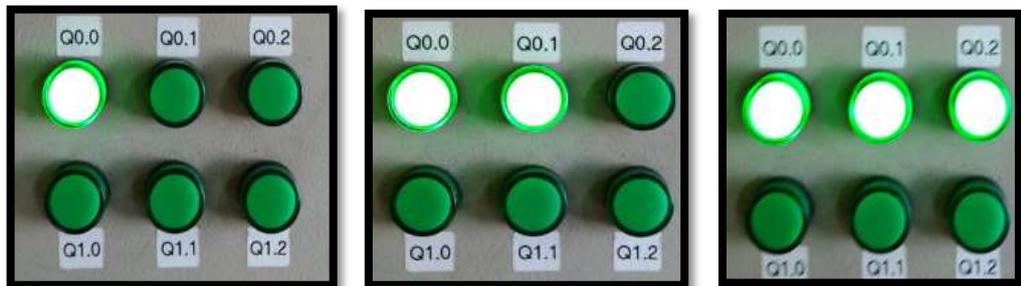
- Transcurrido un tiempo de tres segundos se desactivan



- Después del mismo intervalo tiempo se vuelven activar las mismas salidas



- por último se apagan indefinidamente.
- Al pulsar I0.5 se activa una secuencia, es decir primero Q0.0 durante tres segundos, posteriormente se activa Q0.1 durante tres segundos y por último Q0.2 durante otros tres segundos, culminando con el estado de apagado en todas las salidas.



### • MARCO PROCEDIMENTAL

Diseñar un programa que me permita ejecutar una secuencia de encendido del PLC MDuino.

### PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO

- Lo primero que se procede a realizar es la declaración de las entradas digitales:

I0.4

I0.5

I1.2

- Posteriormente se procede a declarar las salidas digitales a:

Q0.0

Q0.1

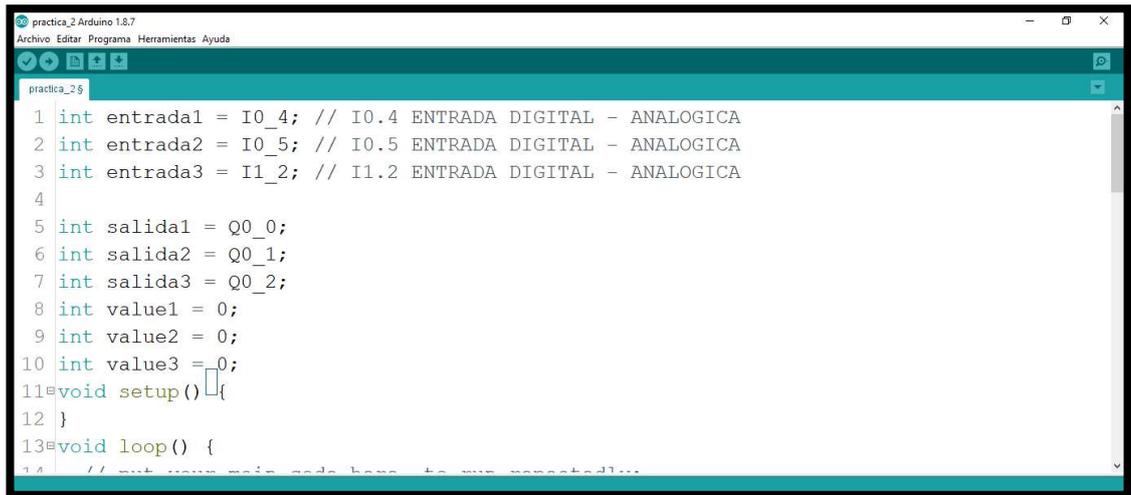
Q0.2

- Se definen las variables para guardar los estados de las entradas digitales, las cuales son:

VALUE1

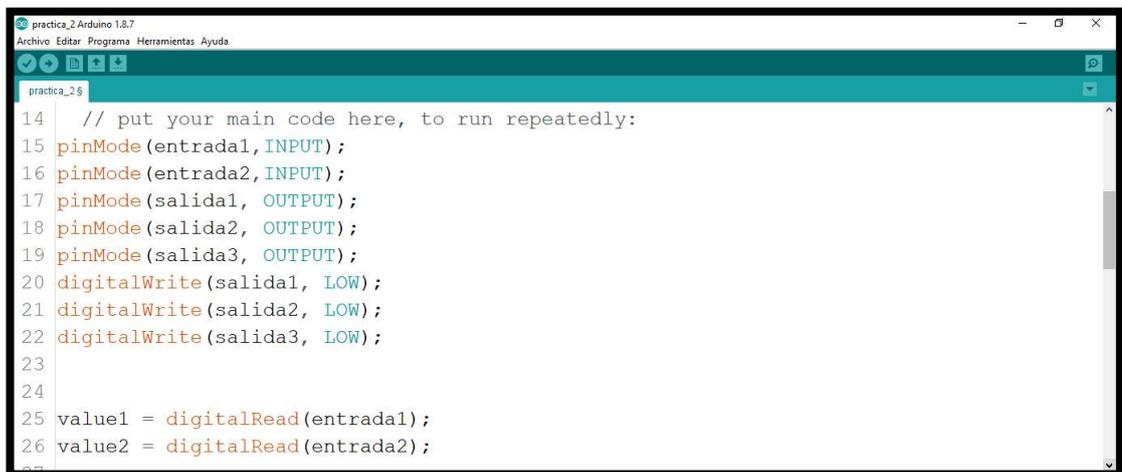
VALUE2

VALUE3

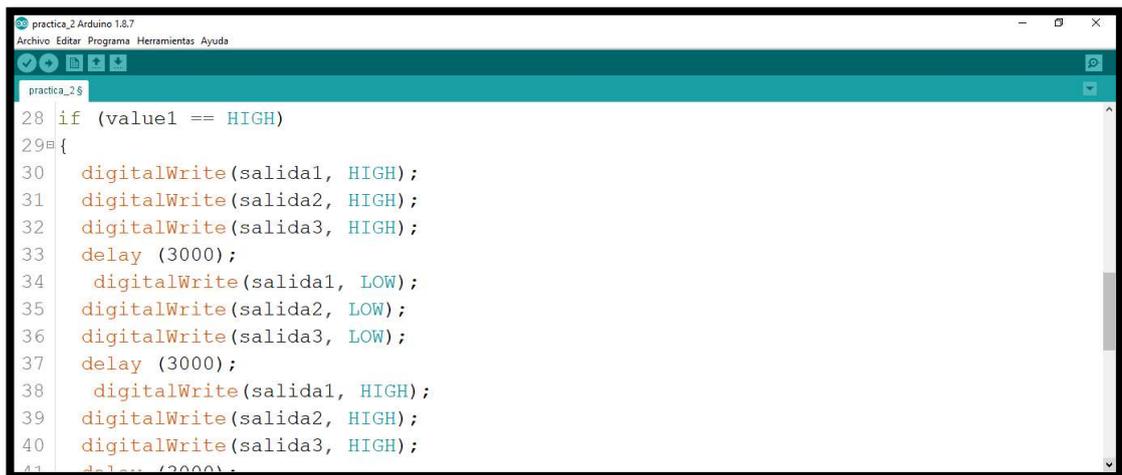


```
practica_2 $
1 int entrada1 = I0_4; // I0.4 ENTRADA DIGITAL - ANALOGICA
2 int entrada2 = I0_5; // I0.5 ENTRADA DIGITAL - ANALOGICA
3 int entrada3 = I1_2; // I1.2 ENTRADA DIGITAL - ANALOGICA
4
5 int salida1 = Q0_0;
6 int salida2 = Q0_1;
7 int salida3 = Q0_2;
8 int value1 = 0;
9 int value2 = 0;
10 int value3 = 0;
11 void setup() {
12 }
13 void loop() {
14 // put your main code here, to run repeatedly:
```

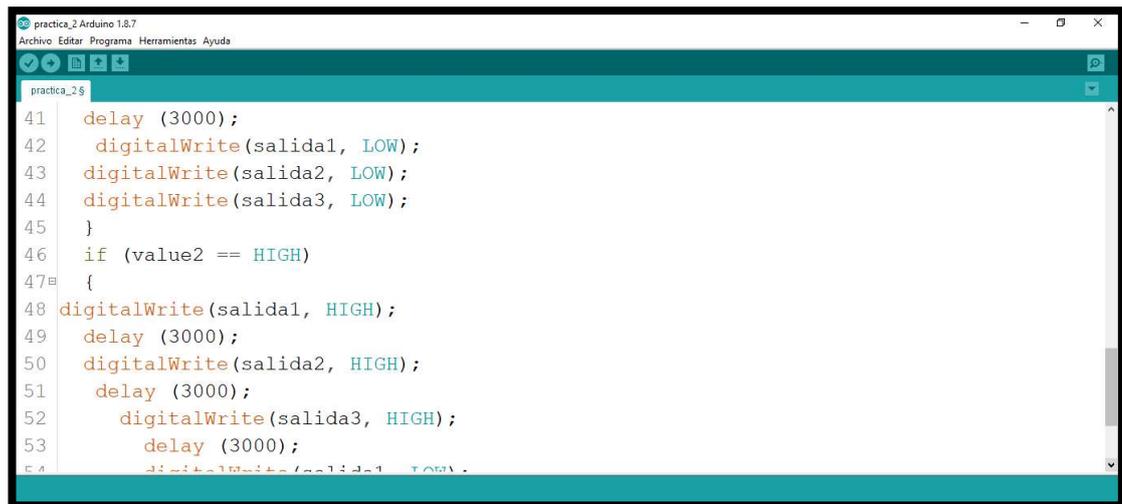
En la siguiente imagen se procede a declarar los puertos de entrada y salida del PLC. Y activación de las salidas digitales en secuencia.



```
practica_2 $
14 // put your main code here, to run repeatedly:
15 pinMode(entrada1, INPUT);
16 pinMode(entrada2, INPUT);
17 pinMode(salida1, OUTPUT);
18 pinMode(salida2, OUTPUT);
19 pinMode(salida3, OUTPUT);
20 digitalWrite(salida1, LOW);
21 digitalWrite(salida2, LOW);
22 digitalWrite(salida3, LOW);
23
24
25 value1 = digitalRead(entrada1);
26 value2 = digitalRead(entrada2);
```



```
practica_2 $
28 if (value1 == HIGH)
29 {
30 digitalWrite(salida1, HIGH);
31 digitalWrite(salida2, HIGH);
32 digitalWrite(salida3, HIGH);
33 delay (3000);
34 digitalWrite(salida1, LOW);
35 digitalWrite(salida2, LOW);
36 digitalWrite(salida3, LOW);
37 delay (3000);
38 digitalWrite(salida1, HIGH);
39 digitalWrite(salida2, HIGH);
40 digitalWrite(salida3, HIGH);
41 delay (3000);
```



```
practica_2_Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_2$
41 delay (3000);
42 digitalWrite(salida1, LOW);
43 digitalWrite(salida2, LOW);
44 digitalWrite(salida3, LOW);
45 }
46 if (value2 == HIGH)
47 {
48 digitalWrite(salida1, HIGH);
49 delay (3000);
50 digitalWrite(salida2, HIGH);
51 delay (3000);
52 digitalWrite(salida3, HIGH);
53 delay (3000);
54 digitalWrite(salida1, LOW);
55 }
```

## **RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)**

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión ethernet.

## **CONCLUSIONES:**

Al realizar dicha práctica se determinó que el PLC Mduino funciona exactamente similar a un PLC convencional. Los temporizadores funcionan correctamente y sus entradas con salidas digitales de la misma manera.

## **RECOMENDACIONES:**

Tener en cuenta para que el ciclo se vuelva a repetir se utiliza el lazo VOID LOOP.

		REVISION 1/1	Página 1 de -
 <b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> ECUADOR		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 3

TEMA:

ACTIVACIÓN DE ENTRADAS DIGITALES Y ANALÓGICAS MODULO  
MAESTRO

- **OBJETIVO GENERAL:**
  - Verificar la correcta lectura de señales analógicas PLC maestro
- **OBJETIVOS ESPECIFICOS:**
  - Realizar la práctica utilizando las entradas digitales y analógicas del PLC Mduino 38R+.
  - Verificar el correcto funcionamiento de todas sus variables
- **ACTIVIDAD.-**
  - i. Desarrollar un programa en el módulo Mduino 38R para el reconocimiento del funcionamiento correcto en entradas digitales y analógicas que contiene el modulo maestro.
  - ii. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
  - iii. Ampliar el conocimiento sobre programación en el entorno IDE.
- **DESARROLLO.-**

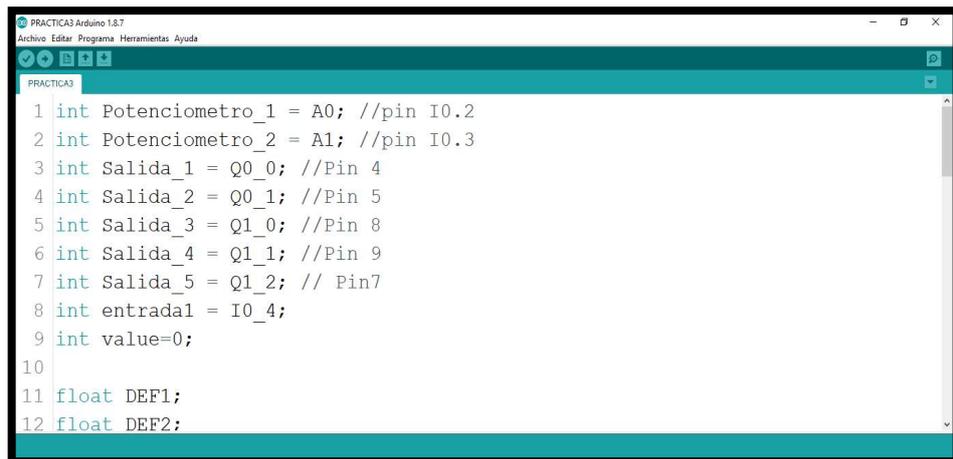
La secuencia de la práctica se detalla en lo siguiente:

- Consiste en pulsar la entrada I0.4, para dar inicio y proceda a leer el valor de la variable en la entrada analógica 1 (A0/I0.2) con la condición de que sea menor a un 80 % del valor en la entrada analógica que se esté leyendo.
- Se activarán las salidas Q0.0 y Q0.1, dicha variación del valor analógico la podemos observar vía monitor serie de Arduino IDE.
- Al llegar al 80 % del valor analógico, las salidas digitales Q0.0 Y Q0.1 se apagarán de manera continua y se lee el valor de la entrada analógica 2 (A1/ I0.3).
- Posteriormente se activarán las salidas Q1.0, Q1.1, Q1.2 mientras se lee el valor de esta última entrada analógica 2, debe ser menor o igual a un 60% del valor analógico leído, cuando se llegue a este 60%, se apagarán estas salidas Q1.0, Q1.1, Q1.2.

- **MARCO PROCEDIMENTAL**

Diseñar un programa que me permita ejecutar una secuencia de encendido del PLC Mduino.

PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO



```
1 int Potenciometro_1 = A0; //pin I0.2
2 int Potenciometro_2 = A1; //pin I0.3
3 int Salida_1 = Q0_0; //Pin 4
4 int Salida_2 = Q0_1; //Pin 5
5 int Salida_3 = Q1_0; //Pin 8
6 int Salida_4 = Q1_1; //Pin 9
7 int Salida_5 = Q1_2; // Pin7
8 int entrada1 = I0_4;
9 int value=0;
10
11 float DEF1;
12 float DEF2;
```

A continuación, se detalla el procedimiento que se llevó a cabo para la programación de la práctica:

- Se realiza la declaración de las entradas I0.2 - I0.3 como analógicas

I0.4 como entrada digital.

- Se detalla cómo se declararon las salidas digitales en la programación:

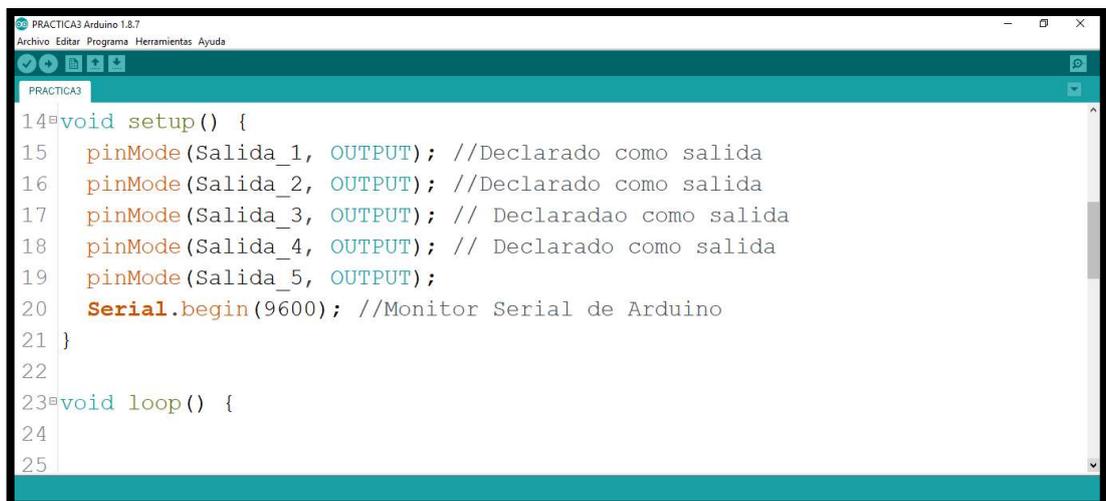
Salida1 = Q0.0

Salida2 = Q0.1

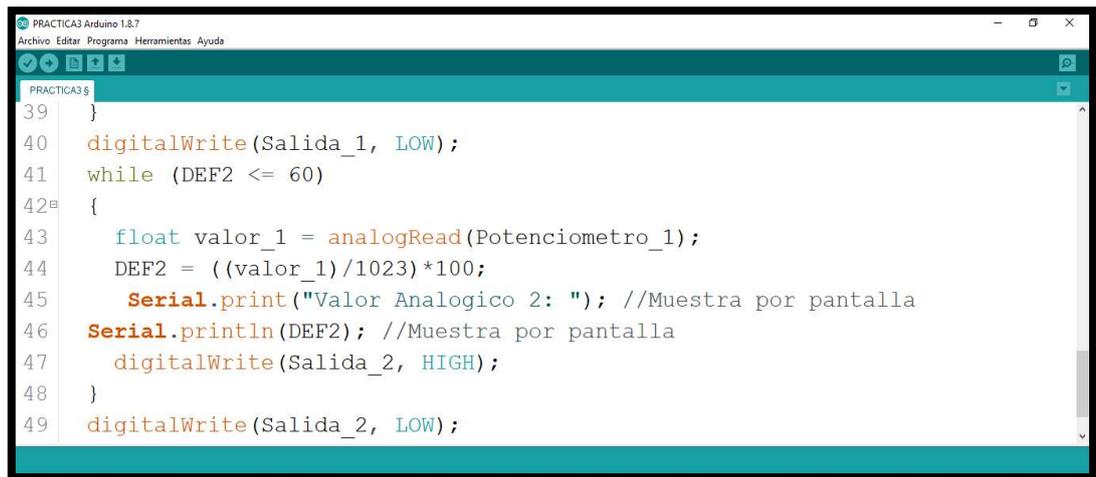
Salida3 = Q1.0

Salida4 = Q1.1

Salida5 = Q1.2



```
14 void setup() {
15   pinMode(Salida_1, OUTPUT); //Declarado como salida
16   pinMode(Salida_2, OUTPUT); //Declarado como salida
17   pinMode(Salida_3, OUTPUT); // Declaradao como salida
18   pinMode(Salida_4, OUTPUT); // Declarado como salida
19   pinMode(Salida_5, OUTPUT);
20   Serial.begin(9600); //Monitor Serial de Arduino
21 }
22
23 void loop() {
24
25
```



```
PRACTICA3 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
PRACTICA3 $
39 }
40 digitalWrite(Salida_1, LOW);
41 while (DEF2 <= 60)
42 {
43     float valor_1 = analogRead(Potenciometro_1);
44     DEF2 = ((valor_1)/1023)*100;
45     Serial.print("Valor Analogico 2: "); //Muestra por pantalla
46     Serial.println(DEF2); //Muestra por pantalla
47     digitalWrite(Salida_2, HIGH);
48 }
49 digitalWrite(Salida_2, LOW);
```

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión ethernet.

## CONCLUSIONES:

Al realizar esta práctica concluimos que las entradas analógicas funcionan de una manera óptima para realizar diversas prácticas con mayor complejidad, dándole un eficaz uso académico.

## RECOMENDACIONES:

Ninguna

		REVISION 1/1	Página 1 de -
 <b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> ECUADOR		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 4

TEMA:

ACTIVACIÓN DE ENTRADAS Y SALIDAS DIGITALES EN EL  
MÓDULO ESCLAVO PLC 19R

- **OBJETIVO GENERAL:**

Realizar la práctica para la activación de señales digitales de entrada y salida en el módulo esclavo 19R

- **OBJETIVOS ESPECIFICOS:**

- Realizar pruebas de las entradas y salidas digitales del módulo esclavo PLC 19R
- Verificar el funcionamiento del PLC.

- **ACTIVIDAD.-**

- i. Desarrollar un programa en el módulo Mduino 19R para el reconocimiento del funcionamiento correcto en entradas y salidas digitales que contiene el módulo esclavo.
- ii. Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- iii. Ampliar el conocimiento sobre programación en el entorno IDE.

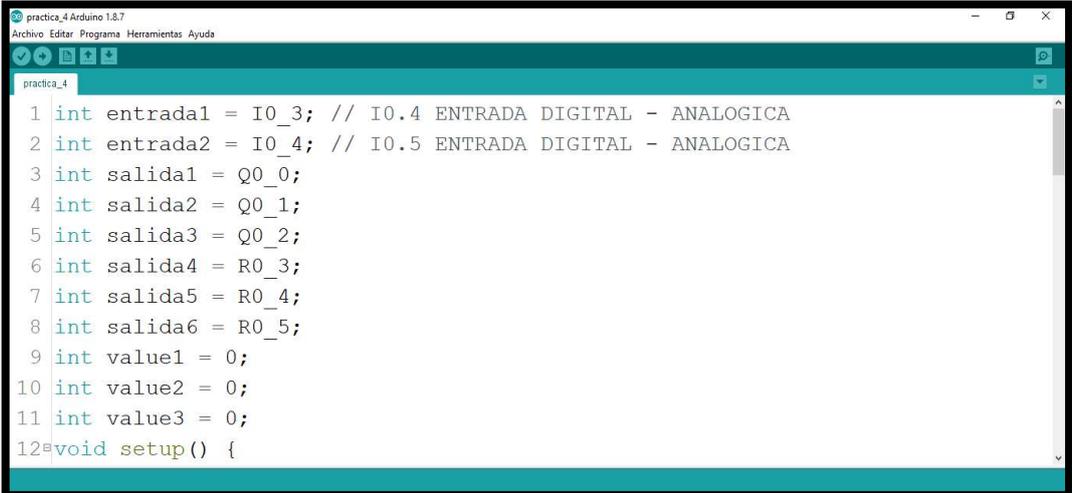
- **DESARROLLO.-**

El funcionamiento de la práctica consiste, que al activar la entrada digital I0.3 da paso a la activación de las salidas Q0.0, Q0.1 Y Q0.2 a la vez comienza a transcurrir un intervalo de tiempo de tres segundos, pasado estos tres segundos, se apagan. Luego de tres segundos se vuelven activar durante el mismo intervalo de tiempo, y por último se apagan.

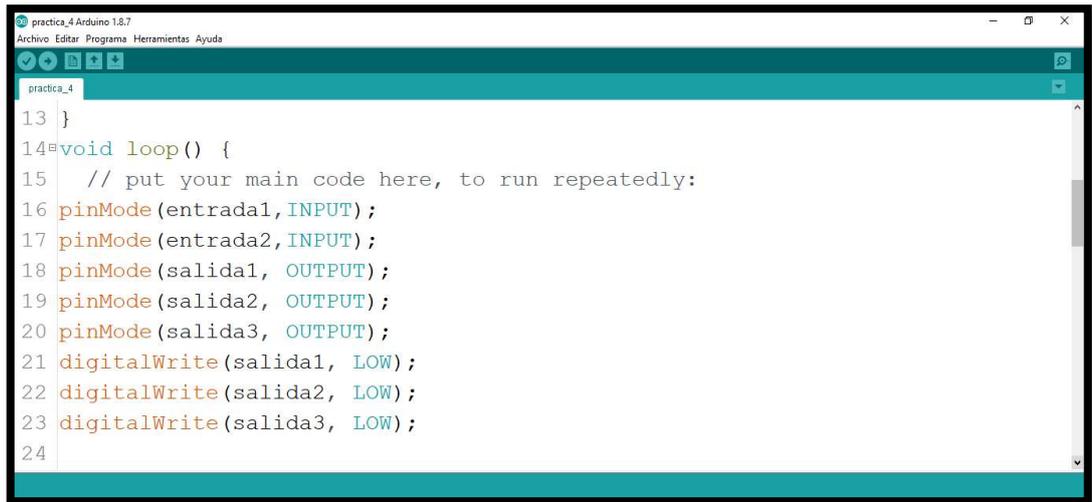
- **MARCO PROCEDIMENTAL**

Diseñar un programa para la activación de señales digitales de entrada y salida en el módulo esclavo 19R

**PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO**



```
practica_4 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_4
1 int entrada1 = I0_3; // I0.4 ENTRADA DIGITAL - ANALOGICA
2 int entrada2 = I0_4; // I0.5 ENTRADA DIGITAL - ANALOGICA
3 int salida1 = Q0_0;
4 int salida2 = Q0_1;
5 int salida3 = Q0_2;
6 int salida4 = R0_3;
7 int salida5 = R0_4;
8 int salida6 = R0_5;
9 int value1 = 0;
10 int value2 = 0;
11 int value3 = 0;
12 void setup() {
```



```
practica_4 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_4
13 }
14 void loop() {
15   // put your main code here, to run repeatedly:
16   pinMode(entrada1, INPUT);
17   pinMode(entrada2, INPUT);
18   pinMode(salida1, OUTPUT);
19   pinMode(salida2, OUTPUT);
20   pinMode(salida3, OUTPUT);
21   digitalWrite(salida1, LOW);
22   digitalWrite(salida2, LOW);
23   digitalWrite(salida3, LOW);
24
```

El primer paso a realizar es la definición y declaración de las entradas:

entrada1 = I0.3

entrada2 = I0.4

Se definen y se declaran las siguientes salidas del PLC 19R

salida1 = Q0.0

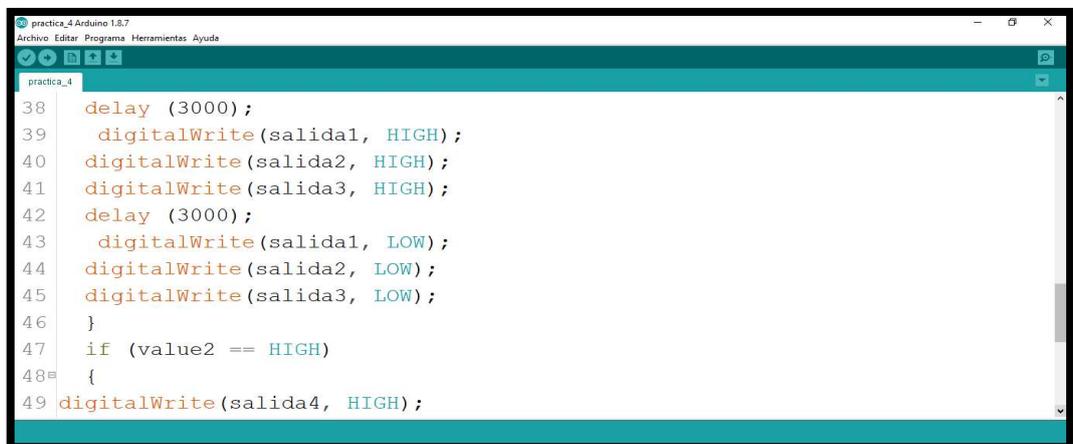
salida2 = Q0.1

salida3 = Q0.2

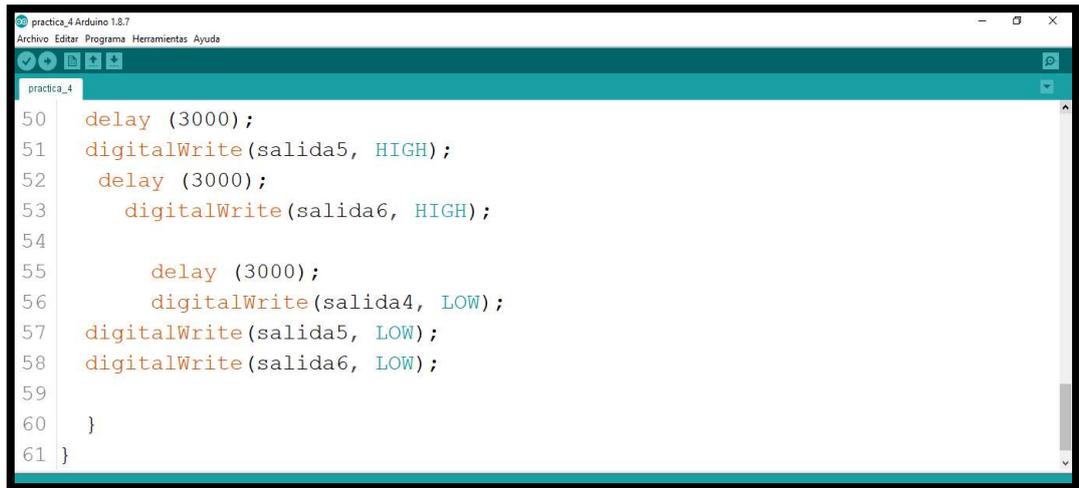
salida4 = R0.3

salida5 = R0.4

salida6 = R0.5



```
practica_4 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_4
38   delay (3000);
39   digitalWrite(salida1, HIGH);
40   digitalWrite(salida2, HIGH);
41   digitalWrite(salida3, HIGH);
42   delay (3000);
43   digitalWrite(salida1, LOW);
44   digitalWrite(salida2, LOW);
45   digitalWrite(salida3, LOW);
46   }
47   if (value2 == HIGH)
48   {
49   digitalWrite(salida4, HIGH);
```

A screenshot of the Arduino IDE interface. The window title is "practica\_4 Arduino 1.8.7". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar shows icons for file operations and execution. The main editor area displays the following code:

```
50 delay (3000);
51 digitalWrite(salida5, HIGH);
52   delay (3000);
53     digitalWrite(salida6, HIGH);
54
55         delay (3000);
56           digitalWrite(salida4, LOW);
57     digitalWrite(salida5, LOW);
58     digitalWrite(salida6, LOW);
59
60 }
61 }
```

### **RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)**

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.

### **CONCLUSIONES:**

Se verificó en la realización de la práctica que todas las entradas y salidas digitales del módulo esclavo 19R funcionan muy bien. Dándole el visto bueno a la parte digital para que puedan proceder a realizar otras prácticas de mayor complejidad.

### **RECOMENDACIONES:**

Hay que recalcar que las salidas R0.3, R0.4 Y R0.5, son salidas de relé las cuales están cableadas a 24 VDC.

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 5

TEMA:

COMUNICACIÓN SERIAL MODULO LCD LECTURA DE ENTRADA  
ANALÓGICA.

- **OBJETIVO GENERAL:**

Utilizar la entrada lógica del PLC Mduino 19R y establecer comunicación I2C con un módulo LCD 16x2.

- **OBJETIVOS ESPECIFICOS:**

- Establecer comunicación i2c con un módulo LCD 16 x 2
- Utilizar la entrada lógica del PLC Mduino 19R.

- **ACTIVIDAD.-**

- Desarrollar un programa en el módulo Mduino 19R para el reconocimiento del funcionamiento correcto de las entradas analógicas que contiene el módulo esclavo.
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**

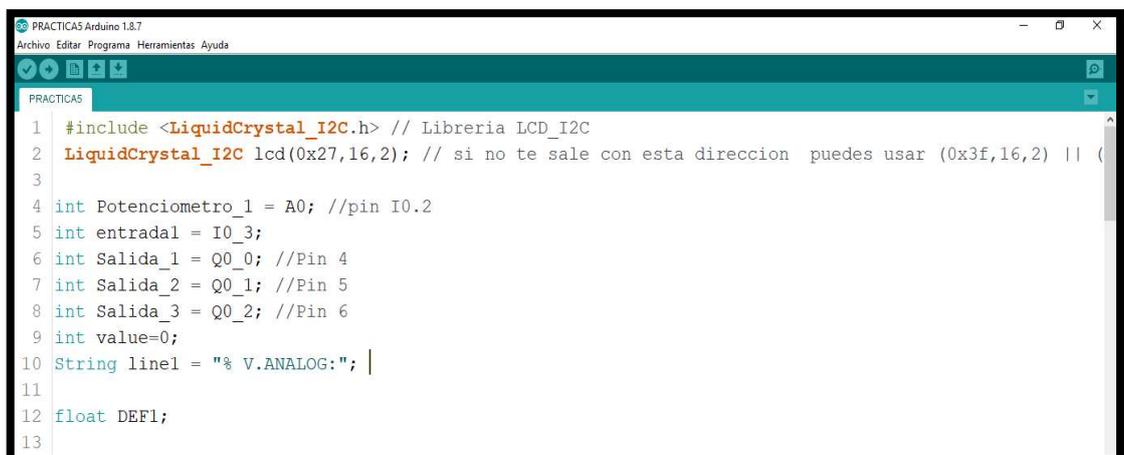
Al momento de pulsar la Entrada digital I0.3 en el módulo, de manera automática se presenta en el LCD 16x2, el porcentaje de valor analógico que se está leyendo en la entrada analógica I0.2, a través del potenciómetro lineal de 10 K instalado en el módulo, el cual está conectado a una tarjeta reguladora de voltaje de 0-10 Vdc, variando así este nivel de voltaje en la entrada I0.2

Diseñar un programa que me permita interpretar la entrada lógica del PLC Mduino 19R y establecer comunicación i2c con un módulo LCD 16x2

### PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO

Para dar inicio a la programación, se agrega la librería LiquidCrystal\_I2C del LCD, al código se declara la entrada A0 a la entrada analógica I0.2 del PLC 19R se establecen las salidas digitales

entrada1 = I0.3  
Salida\_1 = Q0.0  
Salida\_2 = Q0.1  
Salida\_3 = Q0.2



```
1 #include <LiquidCrystal_I2C.h> // Libreria LCD_I2C
2 LiquidCrystal_I2C lcd(0x27,16,2); // si no te sale con esta direccion puedes usar (0x3f,16,2) || (
3
4 int Potenciometro_1 = A0; //pin I0.2
5 int entrada1 = I0_3;
6 int Salida_1 = Q0_0; //Pin 4
7 int Salida_2 = Q0_1; //Pin 5
8 int Salida_3 = Q0_2; //Pin 6
9 int value=0;
10 String line1 = "% V.ANALOG:"; |
11
12 float DEF1;
13
```

```
PRACTICAS Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
PRACTICAS
14 void setup() {
15   lcd.init();
16   lcd.backlight();
17   lcd.clear();
18   pinMode(Salida_1, OUTPUT); //Declarado como salida
19   pinMode(Salida_2, OUTPUT); //Declarado como salida
20   pinMode(Salida_3, OUTPUT); // Declaradao como salida
21   Serial.begin(9600); //Monitor Serial de Arduino
22 }
23
24 void loop() {
25
```

Se ha configurado un valor del 80%, del valor analógico leído, el cual cuando se supera el mismo se activarán las salidas digitales:

Q0.0

Q0.1

Q0.2, del módulo esclavo.

```
PRACTICAS Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
PRACTICAS
25
26 value= digitalRead(entradal);
27 if (value == HIGH)
28 {
29   //Lee el valor analogo y lo guarda en la variable valor
30   DEF1=0;
31
32   while (DEF1 <= 80)
33   {
34     float valor_1 = analogRead(Potenciometro_1);
35     DEF1 = ((valor_1)/1023)*100;
36     Serial.print("Valor Analogico 1:"); //Muestra por pantalla
37     Serial.println(DEF1);
38
```

```
PRACTICAS Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
PRACTICAS
39   lcd.setCursor(0,0);
40   lcd.print("=>PRACTICA 5<=");
41   lcd.setCursor(0,1);
42   lcd.print(line1);
43   lcd.setCursor(11,1);
44   lcd.print(DEF1);
45   digitalWrite(Salida_1, LOW);
46   digitalWrite(Salida_2, LOW);
47   digitalWrite(Salida_3, LOW);
48   delay(800);
49 }
50 digitalWrite(Salida_1, HIGH);
51 digitalWrite(Salida_2, HIGH);
52 digitalWrite(Salida_3, HIGH);
53
```

Para continuar con la lectura, tendremos que activar nuevamente IO\_3, cuando se varié el potenciómetro y si el valor es inferior al 80% establecido, de manera automática se desactivarán las 3 salidas digitales en mención.

## **RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)**

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.

## **CONCLUSIONES:**

Se realizó la comunicación serial y la lectura de entrada analógica con resultados positivos, sin dar mayor inconveniente.

## **RECOMENDACIONES:**

Hay que tener en cuenta que para continuar con la lectura de la entrada analógica, tendremos que activar nuevamente I0.3.

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 6

TEMA:

COMUNICACIÓN SERIAL MAESTRO - ESCLAVO

- **OBJETIVO GENERAL:**

Utilizar los puertos de comunicación serial disponibles en los PLC 38R+ modulo maestro y 19R modulo esclavo.

- **OBJETIVOS ESPECIFICOS:**

- Habilitar y utilizar los puertos de comunicación serial disponibles en los PLC

- **ACTIVIDAD.-**

- Desarrollar un programa entre el módulo Mduino 19R y el 38 R para habilitar la comunicación serial de ambos, no olvidando que la versión 38R es el maestro y 19R es el esclavo.
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**



Es importante el uso de la función Serial1.begin (9600), porque con la misma se habilita el envío de datos a través de los puertos RX1 y TX1 de este PLC. Además, se habilita la función Serial. Begin (9600), para poder observar los datos a comunicarse a través del monitor serial.

Entonces al pulsar I0.4 o I0.5, se activará la comunicación serial hacia el módulo esclavo, enviando como dato la letra "r" y luego de 4seg la letra "l" además, estas letras serán presentadas en el monitor serie y se activa la Salida Q0.0, indicando que la comunicación serie ha iniciado.

Diseñar un programa que me permita utilizar los puertos de comunicación serial disponibles en los PLC 38R+ modulo maestro y 19R modulo esclavo.

### **PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO**

SE DECLARAN ENTRADAS DIGITALES:

entrada1 = I0.4

entrada2 = I0.5

salida1 = Q0.0

## MAESTRO 38R

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO1
1 /*MAESTRO*/
2 int entrada1 = IO_4;
3 int entrada2 = IO_5;
4 int salida1 = Q0_0;
5 int valor = 0;
6 int valor1= 0;
7 |
8 void setup()
9
```

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO1
10 {
11   pinMode(entrada1, INPUT);
12   pinMode(entrada2, INPUT);
13   pinMode(salida1, OUTPUT);
14   Serial1.begin(9600);
15   Serial.begin(9600);
16
17 }
18
19 void loop()
20 {
```

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO1
21 //LEE EL VALOR DE LA ENTRADA IO_4 y IO_5
22 valor = digitalRead(entrada1);
23 valor1 = digitalRead(entrada2);
24 //si se pulsa IO_4 entonces se activa la comunicacion
25
26
27 if (valor == 1)
28 {
29   digitalWrite(salida1, HIGH);
30   Serial1.write("r");
31   Serial.write("r");
```

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO1
32   delay(4000);
33   Serial1.write("l");
34   Serial.write("l");
35   delay(4000);
36 }
37 else
38 {
39   digitalWrite(salida1, LOW);
40 }
41 if (valor1 == 1)
42 {
```

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
MAESTRO1
43 digitalWrite(salida1,HIGH);
44 Serial1.write("r");
45 Serial.write("r");
46 delay(4000);
47 Serial1.write("l");
48 Serial.write("l");
49 delay(4000);
50 }
51 else
52 {
53 digitalWrite(salida1,LOW);
```

```
MAESTRO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO1
46 delay(4000);
47 Serial1.write("l");
48 Serial.write("l");
49 delay(4000);
50 }
51 else
52 {
53 digitalWrite(salida1,LOW);
54 }
55
56 delay(25);
57 }
```

FUNCIONAMIENTO DEL PROGRAMA DEL MODULO ESCLAVO PLC MDUINO 19R

### ESCLAVO 19R

```
ESCLAVO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
ESCLAVO1
1 #include <LiquidCrystal_I2C.h>
2 #include <LiquidCrystal_I2C.h> // Libreria LCD_I2C
3 LiquidCrystal_I2C lcd(0x27,16,2);
4
5 int salida1 = Q0_0;
6 int salida2 = Q0_1;
7 int salida3 = Q0_2;
8
9 void setup()
10 {
11 lcd.init();
12 lcd.backlight();
13 lcd.clear();
```

```
ESCLAVO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
ESCLAVO1
14 lcd.setCursor(0,0);
15 lcd.print(">PRACTICA 6<=");
16 lcd.setCursor(0,1);
17 lcd.print("SERIAL:MAST-SLAV");
18
19 pinMode(salida1,OUTPUT);
20 pinMode(salida2,OUTPUT);
21 pinMode(salida3,OUTPUT);
22 Serial1.begin(9600);
23
24 }
25 void loop()
26 {
```

Se declaran las salidas:

salida1 = Q0.0

salida2 = Q0.1

salida3 = Q0.2

```
ESCLAVO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
ESCLAVO1
27 lcd.display();
28 delay(500);
29 lcd.noDisplay();
30 delay(500);
31
32 char dato = Serial1.read();//Guardamos en la variable dato el valor leido
33
34 switch(dato)
35 { //Comprobamos el dato
36     case 'r': //Si recibimos una 'r' ...
37         for(int i=0; i<20; i++){
38             digitalWrite(salida1,HIGH);
39             digitalWrite(salida2,HIGH);
```

```
ESCLAVO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
ESCLAVO1
40         delay(100);
41         digitalWrite(salida1,LOW);
42         digitalWrite(salida2,LOW);
43         delay(100);
44     }
45     break;
46     case 'l': //si recibimos una 'l' ...
47         for(int i=0; i<10; i++)
48         {
49             digitalWrite(salida3,HIGH);
50             delay(300);
51             digitalWrite(salida3,LOW);
```

Es necesario iniciar la comunicación serial mediante la función Serial1.begin (9600) para poder recibir datos que provienen del Maestro.

En el módulo LCD con I2C, se muestra el tema de la práctica al momento de que el módulo maestro envía el dato "r" y "l" el módulo esclavo lo recibe y de ser así para el dato r cumple la siguiente función:

- Activa la salida Q0.0 y Q0.1, durante 100ms alternadamente 20 veces.

Para el dato I cumple la siguiente función:

- Activa la salida Q0.2 durante 300 ms y la desactiva durante el mismo tiempo 10 veces.

```

ESCLAVO1 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
ESCLAVO1
50     digitalWrite(salida3,HIGH);
51     delay(300);
52     digitalWrite(salida3,LOW);
53     delay(300);
54   }
55   break;
56   default:
57     digitalWrite(salida1,LOW);
58     digitalWrite(salida2,LOW);
59     digitalWrite(salida3,LOW);
60     break;
61 }

```

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.

## CONCLUSIONES:

En esta práctica se realizó la comunicación maestro (38R) y esclavo (19R), tomando en cuenta que la comunicación serial se ejecuta mediante la función Serial1.begin (9600) para poder recibir datos que provienen del Maestro (en este caso 38R).

## RECOMENDACIONES:

Es importante el uso de la función Serial1.begin (9600), porque con la misma se habilita el envío de datos a través de los puertos RX1 y TX1 de este PLC. Además, se habilita la función Serial. Begin (9600), para poder observar los datos a comunicarse a través del monitor serial.

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 7

TEMA:

COMUNICACIÓN SERIAL ESCLAVO - MAESTRO

- **OBJETIVO GENERAL:**

Utilizar los puertos de comunicación serial disponibles en los 19R modulo maestro y PLC 38R+ modulo esclavo

- **OBJETIVOS ESPECIFICOS:**

- Habilitar y utilizar los puertos de comunicación serial disponibles en los PLC

- **ACTIVIDAD.-**

- Desarrollar un programa entre el módulo MDuino 19R y el 38 R para habilitar la comunicación serial de ambos, no olvidando que la versión 38R es el esclavo y 19R es el maestro
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**

Es importante el uso de la función Serial1.begin (9600), porque con la misma se habilita el envío de datos a través de los puertos RX1 y TX1 de este PLC.

Además, se habilita la función Serial. Begin (9600), para poder observar los datos a comunicarse a través del monitor serial.

Entonces al pulsar I0.4 o I0.5, se activará la comunicación serial hacia el módulo esclavo, enviando como dato la letra "r" y luego de 4seg la letra "l" además estas letras serán presentadas en el monitor serie y se activa la Salida Q0.0, indicando que la comunicación serie ha iniciado.

Diseñar un programa que me permita utilizar los puertos de comunicación serial disponibles en los 19R modulo maestro y PLC 38R+ modulo esclavo

## PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO

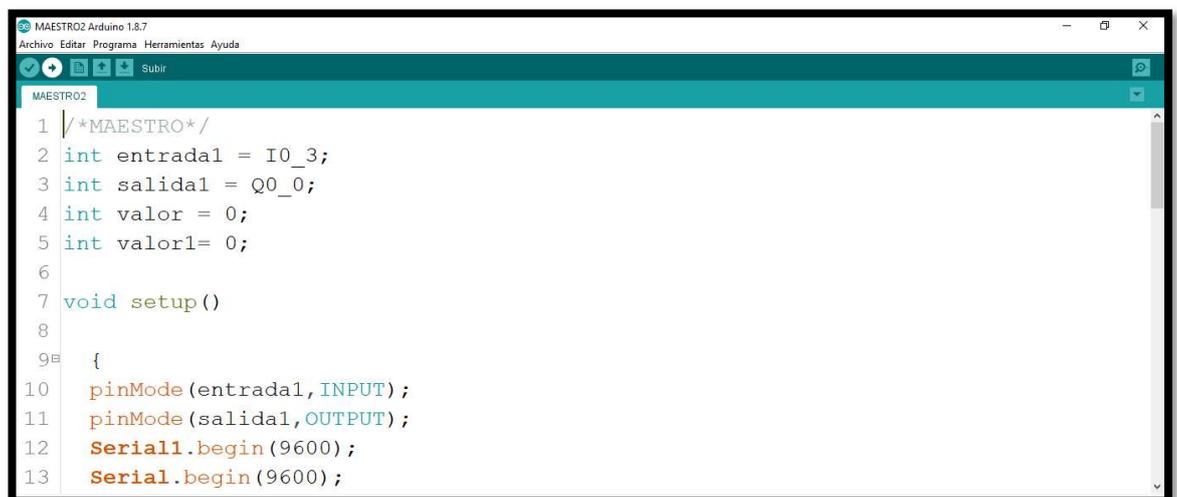
SE DECLARAN ENTRADAS DIGITALES:

entrada1 = I0.4

entrada2 = I0.5

salida1 = Q0.0

### MAESTRO



```
1 /*MAESTRO*/
2 int entrada1 = I0_3;
3 int salida1 = Q0_0;
4 int valor = 0;
5 int valor1= 0;
6
7 void setup()
8
9 {
10  pinMode(entrada1, INPUT);
11  pinMode(salida1, OUTPUT);
12  Serial1.begin(9600);
13  Serial.begin(9600);
```

```
MAESTRO2 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
MAESTRO2
15 }
16
17 void loop()
18 {
19 //LEE EL VALOR DE LA ENTRADA IO_4 y IO_5
20 valor = digitalRead(entradal);
21 //si se pulsa IO_4 entonces se activa la comunicacion
22
23
24 if (valor == 1)
25 {
26 digitalWrite(salidal,HIGH);
27 Serial1.write("r");
```

```
MAESTRO2 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Nuevo
MAESTRO2
28 Serial1.write("r");
29 delay(4000);
30 Serial1.write("l");
31 Serial.write("l");
32 delay(4000);
33 }
34 else
35 {
36 digitalWrite(salidal,LOW);
37 }
38 if (valor1 == 1)
39 {
40 digitalWrite(salidal,HIGH);
```

```
MAESTRO2 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
MAESTRO2$
41 Serial1.write("r");
42 Serial.write("r");
43 delay(4000);
44 Serial1.write("l");
45 Serial.write("l");
46 delay(4000);
47 }
48 else
49 {
50 digitalWrite(salidal,LOW);
51 }
52 delay(25);
53 }
```

Es importante el uso de la función Serial1.begin (9600), porque con la misma se habilita el envío de datos a través de los puertos RX1 y TX1 de este PLC.

Además, se habilita la función Serial. Begin (9600), para poder observar los datos a comunicarse a través del monitor serial.

Entonces al pulsar IO.4 o IO.5, se activará la comunicación serial hacia el módulo esclavo, enviando como dato la letra "r" y luego de 4seg la letra "l" además estas letras serán presentadas en el monitor

serie y se activa la Salida Q0.0, indicando que la comunicación serie ha iniciado.

## FUNCIONAMIENTO DEL PROGRAMA DEL MODULO ESCLAVO PLC MDUINO 19R

Se declaran las salidas:

salida1 = Q0.0

salida2 = Q0.1

salida3 = Q0.2



```
ESCLAVO2
3 int salida1 = Q0_0;
4 int salida2 = Q0_1;
5 int salida3 = Q0_2;
6
7 void setup()
8 {
9
10
11   pinMode(salida1,OUTPUT);
12   pinMode(salida2,OUTPUT);
13   pinMode(salida3,OUTPUT);
14   Serial1.begin(9600);
15
16 }
```

Líneas de programación para activar salidas digitales y programando tiempo de espera.

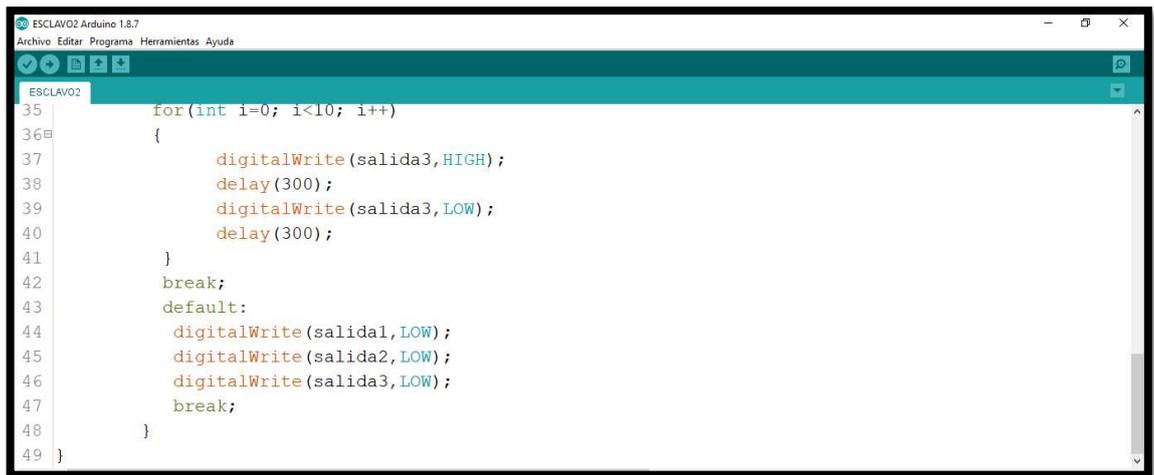


```
ESCLAVO2
17 void loop()
18 {
19
20   char dato = Serial1.read();//Guardamos en la variable dato el valor leído
21
22   switch(dato)
23   {
24     case 'r': //Comprobamos el dato //Si recibimos una 'r' ...
25       for(int i=0; i<20; i++){
26         digitalWrite(salida1,HIGH);
27         digitalWrite(salida2,HIGH);
28         delay(100);
29         digitalWrite(salida1,LOW);
30         digitalWrite(salida2,LOW);
```

Programación de casos, para encendido secuencial.



```
31     delay(100);
32   }
33   break;
34   case 'l': //si recibimos una 'l' ...
35     for(int i=0; i<10; i++)
36     {
37       digitalWrite(salida3,HIGH);
38       delay(300);
39       digitalWrite(salida3,LOW);
40       delay(300);
41     }
42     break;
43     default:
44       digitalWrite(salida1,LOW);
```



```
ESCLAVO2 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
ESCLAVO2
35     for(int i=0; i<10; i++)
36     {
37         digitalWrite(salida3,HIGH);
38         delay(300);
39         digitalWrite(salida3,LOW);
40         delay(300);
41     }
42     break;
43     default:
44         digitalWrite(salida1,LOW);
45         digitalWrite(salida2,LOW);
46         digitalWrite(salida3,LOW);
47         break;
48     }
49 }
```

Es necesario iniciar la comunicación serial mediante la función `Serial1.begin(9600)` para poder recibir datos que provienen del Maestro.

En el módulo LCD con I2C, se muestra el tema de la práctica al momento de que el módulo maestro envía el dato "r" y "l" el módulo esclavo lo recibe y de ser así

Para el dato r cumple la siguiente función:

- activa la salida Q0.0 y Q0.1, durante 100ms alternadamente 20 veces.

Para el dato l cumple la siguiente función:

- activa la salida Q0.2 durante 300 ms y la desactiva durante el mismo tiempo.

10 veces.

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.

## CONCLUSIONES:

En esta práctica se realizó la comunicación Maestro/Esclavo, cambiando de orden a los PLC maestro (19R) y esclavo (38R), hay que tener en cuenta que la comunicación serial se ejecuta mediante la función `Serial1.begin(9600)` para poder recibir datos que provienen del Maestro (en este caso 19R).

## RECOMENDACIONES:

Es importante el uso de la función `Serial1.begin(9600)`, porque con la misma se habilita el envío de datos a través de los puertos RX1 y TX1 de este PLC. Además, se habilita la función `Serial.begin(9600)`, para poder observar los datos a comunicarse a través del monitor serial.

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 8

TEMA:

UTILIZACIÓN DE SEÑAL ANALÓGICA ENTRE EL MODULO  
MAESTRO Y PLANTA

- **OBJETIVO GENERAL:**

Obtener la señal analógica de la planta hacia el modulo maestro

- **OBJETIVOS ESPECIFICOS:**

- Reconocer la señal de la PT-100 en el módulo maestro
- Verificar la activación de todas las señales que se encuentren en la planta

- **ACTIVIDAD.-**

- Desarrollar un programa para el módulo el 38R (maestro) para reconocer la variable análoga de una PT-100
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**

Se da inicio al proceso descargando el tanque 1 sobre el principal activándose la bomba de descarga 1 y la válvula de descarga 1.

Cuando el nivel supera el 30% del llenado del tanque se apaga la válvula de descarga 1 y la bomba de descarga 1. Dando paso a la descarga del tanque 2 luego se activa la válvula de descarga 2 y la bomba de descarga 2.

En el momento que el tanque principal llega hasta el 80% se detiene la descarga del tanque 2, se procede a apagar la válvula de descarga 2 y la bomba de descarga 2.

El siguiente paso es dar inicio al proceso de calentamiento de las mezclas, encendiendo la resistencia hasta que el líquido alcance los 45 grados posteriormente a los 30 grados enciendo el mezclador y va a estar encendido hasta que se llegue a la temperatura seteada (45 grados).

Una vez realizado el proceso de mezclado y calentamiento, enciendo el proceso de recirculación (activando la bomba de recirculación), la válvula de recirculación 1 y la válvula de recirculación 2 hasta q ambos tanques se llenen y se activen ambos sensores de nivel alto que se encuentran ubicados en tanque 1 y tanque 2.

Diseñar un programa que me permita obtener la señal analógica de la planta hacia el modulo maestro.

## **PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO**

A continuación se muestra en las siguientes figuras la declaración de variables a utilizar en el programa, y las líneas de programación para la secuencia respectiva:

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
1 int inicio = I0_5; // I0.4 ENTRADA DIGITAL - ANALOGICA
2 int bomba = Q0_2; // bomba recirculacion
3 int vr1 = R0_3; // R0.3 RECIRCULACION 1 Y R0.4 RECIRCULACION 2
4 int vr2 = R0_4;
5 int agitador = Q1_1; // agitador
6 int vd1 = R0_1; // valvula de descarga 1
7 int bd2 = R0_2; // bomba de descarga 2
8 int bd1 = Q1_2; // bomba de descarga 1
9 int value1 = 0;
10 int value2 = 0;
11 int value3 = 0;
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
12 int sensor = I1_5; // pt100
13 int nivel1 = 0;
14 int nivel2 = 0;
15 int lih2 = I1_3; // nivel alto TQ2
16 int lih1 = I1_4; // nivel alto TQ1
17 int vd2 = R0_5; // valvula de descarga 2
18 int cont = 0;
19 int sensor_ultrasonico = I1_2;
20 float DEF1 = 0;
21 int value;
22 float temp = 0;
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
23 float nn = 0;
24 // primero empieza el proceso de llenado automatico tanque principal maximo
25 // llena del tanque uno hasta el 40% luego llena el tanque 2
26 // una vez lleno enciendo la resistencia con un nivel de seteo a 40 grados
27 // cuando la temperatura llegue a 30 grados enciendo el agitador
28 // una vez que llegue a los 40 grados apago el agitador y empieza el proces
29
30 void setup()
31 {
32 // put your setup code here, to run once:
33 pinMode(vr1, OUTPUT);
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
practica_recirculacion
34 pinMode(vr2, OUTPUT);
35 pinMode(agitador, OUTPUT);
36 pinMode(inicio, INPUT);
37 pinMode(vd1, OUTPUT);
38 pinMode(vd2, OUTPUT);
39 pinMode(bd2, OUTPUT);
40 pinMode(bomba, OUTPUT);
41 pinMode(bd1, OUTPUT);
42 pinMode(agitador, OUTPUT);
43 Serial.begin(9600); //Monitor Serial de Arduino
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
45 }
46 void loop()
47 {
48   value1 = digitalRead(inicio);
49   DEF1=0;
50   // put your main code here, to run repeatedly:
51   while(DEF1<8)
52   { // proceso de llenado tanque principal una vez linealizado la señal del se
53     float valor_1 = analogRead(sensor_ultrasonico);
54     DEF1 = ((valor_1)/1023)*10;
55     Serial.print("nivel taque principal es :"); //Muestra por pantalla
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
56     Serial.println(DEF1);
57     delay(1000);
58     if(DEF1<=4);
59     {
60       digitalWrite(vd1, HIGH);
61       digitalWrite(bd1, HIGH);
62     }
63     if(DEF1>4)
64     {
65       digitalWrite(bd1, LOW);
66       digitalWrite(vd1, LOW);
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
67     digitalWrite(vd2, HIGH);
68     digitalWrite(bd2, HIGH);
69     }
70     }
71     digitalWrite(bd2, LOW);
72     delay(500);
73     digitalWrite(vd2, LOW);
74     DEF1=0;
75
76
77     while (temp<=35)
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
78 {
79     float valor_1 = analogRead(sensor); // sensor equivale a la variable que r
80     temp= ((valor_1)+260.22)/20.68;
81     Serial.print("nivel de temperatura tanque principal :"); //Muestra por
82     Serial.println(temp);
83     delay(2000);
84     analogWrite(Q0_0,1023); // enciendo la resistencia pwm
85
86     if (temp>30)
87     {
88         digitalWrite(agitador , HIGH); // empieza a funcionar agitador hasta que
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
90     }
91     }
92     temp=0;
93     digitalWrite(agitador , LOW);
94     analogWrite(Q0_0,0);
95
96     nn=0;
97     while (nn<1)
98     {
99
100     value2 = digitalRead(lih1);
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
90 }
91 }
92 temp=0;
93 digitalWrite(agitador , LOW);
94 analogWrite(Q0_0,0);
95
96 nn=0;
97 while (nn<1)
98 {
99
100 value2 = digitalRead(lih1);
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
112 if(value3== LOW)
113 {
114     digitalWrite(vr2, LOW);
115 }
116 else
117 {
118     digitalWrite(vr2, HIGH);
119 }
120 if(value3== LOW && value2== LOW)
121 {
122     digitalWrite(bomba, LOW);
```

```
practica_recirculacion Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
practica_recirculacion
123     nn=1;
124 }
125 }
126 digitalWrite(bomba, LOW);
127 }
128
```

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC MDuino y accesorios de accionamiento
- Cable de conexión Ethernet.

## CONCLUSIONES:

Al poner en marcha el proceso de la práctica # 8, se verifico el funcionamiento correcto de los equipos de la planta y sus señales fueron receptadas por el modulo maestro de forma óptima.

## RECOMENDACIONES:

Ninguna

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 9

TEMA:

PROCESO DE RECIRCULACIÓN DE FLUIDOS CONTROLADO POR  
MODULO MAESTRO

- **OBJETIVO GENERAL:**

- Ejecutar el proceso de recirculación de la planta controlado por el modulo maestro 38R

- **OBJETIVOS ESPECIFICOS:**

- Verificar la activación de la recirculación del líquido en los tanques ubicados en la planta.
- Realizar el control en el módulo maestro para la recirculación en los tanques

- **ACTIVIDAD.-**

- Desarrollar un programa para el módulo el 38R (maestro) en el cual se realice el proceso de recirculación de fluidos en la planta diseñada.
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**

Al dar inicio, el proceso empieza con el llenado automático del tanque principal máximo llenado del tanque uno hasta el 40%, luego llena el tanque 2. Una vez lleno, enciendo la resistencia con un nivel de seteo a 40 grados.

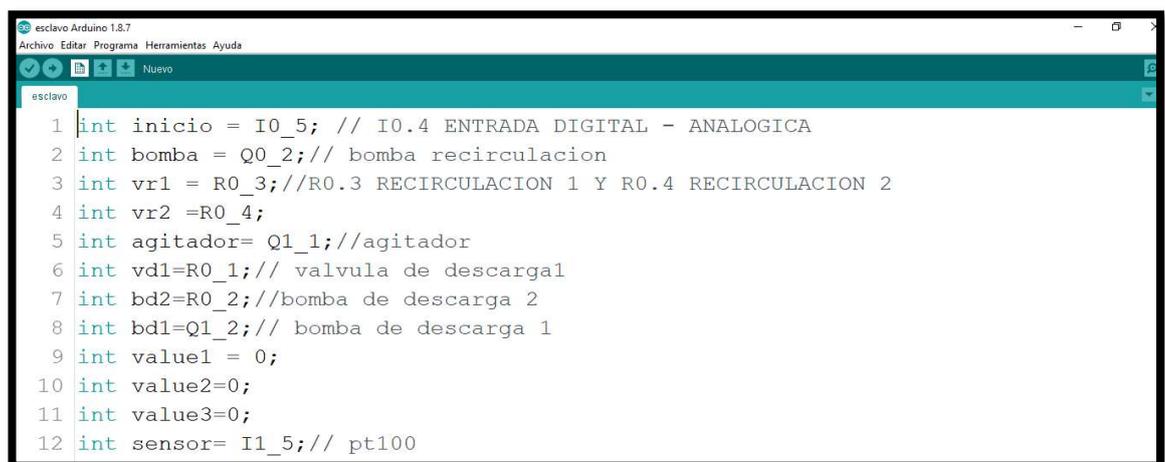
Cuando la temperatura llegue a 30 grados, enciendo el agitador, una vez que llegue a los 40 grados apago el agitador. Este es el proceso que debe de controlar el PLC y para ello se debe diseñar un programa que me permita Ejecutar el proceso de recirculación de la planta controlado por el modulo maestro 38R

## PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO

A continuación se muestran las imágenes de las líneas de programación de la ventana IDE, para el modulo esclavo:

- Se deben declarar las variables a utilizar

### ESCLAVO



```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Nuevo
esclavo
1 int inicio = I0_5; // I0.4 ENTRADA DIGITAL - ANALOGICA
2 int bomba = Q0_2; // bomba recirculacion
3 int vr1 = R0_3; //R0.3 RECIRCULACION 1 Y R0.4 RECIRCULACION 2
4 int vr2 =R0_4;
5 int agitador= Q1_1; //agitador
6 int vd1=R0_1; // valvula de descarga1
7 int bd2=R0_2; //bomba de descarga 2
8 int bd1=Q1_2; // bomba de descarga 1
9 int value1 = 0;
10 int value2=0;
11 int value3=0;
12 int sensor= I1_5; // pt100
```

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
13 int nivel1=0;
14 int nivel2=0;
15 int lih2 = I1_3; // nivel alto TQ2
16 int lih1 = I1_4; // nivel alto TQ1
17 int vd2= R0_5; // valvula de descarga 2
18 int cont=0;
19 int sensor_ultrasonico=I1_2;
20 float DEF1=0;
21 int value;
22 float temp=0;
23 float nn=0;
24 //primero empieza el proceso de llenado automatico tanque principal maximo llena
```

Primero empieza el proceso de llenado automático del tanque principal máximo llenado del tanque uno hasta el 40%, luego llena el tanque 2. Una vez lleno, enciendo la resistencia con un nivel de seteo a 40 grados.

Cuando la temperatura llegue a 30 grados, enciendo el agitador, una vez que llegue a los 40 grados apago el agitador.

En las siguientes líneas se realiza la declaración de las variables como entradas o salidas:

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
25 // llena del tanque uno hasta el 40% luego llena el tanque 2
26 // una vez lleno enciendo la resistencia con un nivel de seteo a 40 grados
27 // cuando la temperatura llegue a 30 grados enciendo el agitador
28 // una vez que llegue a los 40 grados apago el agitador y empieza el proceso de
29
30 void setup()
31 {
32 // put your setup code here, to run once:
33 pinMode(vr1, OUTPUT);
34 pinMode(vr2, OUTPUT);
35 pinMode(agitador,OUTPUT);
36 pinMode(inicio, INPUT);
```

No olvidar escribir SERIAL.BEGIN (9600) para activar el monitor serial de arduino

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
37 pinMode(vd1, OUTPUT);
38 pinMode(vd2, OUTPUT);
39 pinMode(bd2,OUTPUT);
40 pinMode(bomba,OUTPUT);
41 pinMode(bd1, OUTPUT);
42 pinMode(agitador,OUTPUT);
43 Serial.begin(9600); //Monitor Serial de Arduino
44
45 }
46 void loop()
47 {
48 value1 = digitalRead(inicio);
```

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
esclavo
49 DEF1=0;
50 // put your main code here, to run repeatedly:
51 while(DEF1<8)
52 { // proceso de llenado tanque principal una vez linealizado la señal del sensor
53   float valor_1 = analogRead(sensor_ultrasonico);
54   DEF1 = ((valor_1)/1023)*10;
55   Serial.print("nivel tanque principal es :"); //Muestra por pantalla
56   Serial.println(DEF1);
57   delay(1000);
58   if(DEF1<=4);
59   {
60     digitalWrite(vd1, HIGH);
```

Cuando se cumpla la condición, se activa y desactivan determinados puertos digitales

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
esclavo
61   digitalWrite(bd1, HIGH);
62   }
63   if(DEF1>4)
64   {
65     digitalWrite(bd1, LOW);
66     digitalWrite(vd1, LOW);
67     digitalWrite(vd2, HIGH);
68     digitalWrite(bd2, HIGH);
69   }
70   }
71   digitalWrite(bd2, LOW);
72   delay(500);
```

Si cumple la condición, lee el valor de la entrada analógica que conecta al sensor y presenta el nivel de temperatura del tanque principal.

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
esclavo
73   digitalWrite(vd2, LOW);
74   DEF1=0;
75
76
77   while(temp<=35)
78   {
79     float valor_1 = analogRead(sensor); // sensor equivale a la variable que repres
80     temp= ((valor_1)+260.22)/20.68;
81     Serial.print("nivel de temperatura tanque principal :"); //Muestra por pant
82     Serial.println(temp);
83     delay(1000);
84     analogWrite(Q0_0,1023); // enciendo la resistencia pwm
```

Si la temperatura es mayor a 40 grados, se activa el agitador. Caso contrario se apaga.

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
86   if (temp>40)
87   {
88       digitalWrite(agitador , HIGH); // empieza a funcionar agitador hasta que la t
89   }
90   }
91   }
92   temp=0;
93   digitalWrite(agitador , LOW);
94   analogWrite(Q0_0,0);
95
96   nn=0;
97   while (nn<1)
```

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Nuevo
esclavo
98   {
99
100   value2 = digitalRead(lih1);
101   value3 = digitalRead(lih2);
102   digitalWrite(bomba,HIGH);
103   if(value2== LOW)
104   {
105       digitalWrite(vr1, LOW);
106   }
107   else
108   {
109       digitalWrite(vr1, HIGH);
```

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
110   }
111
112   if(value3== LOW)
113   {
114       digitalWrite(vr2, LOW);
115   }
116   else
117   {
118       digitalWrite(vr2, HIGH);
119   }
120   if(value3== LOW && value2== LOW)
121   {
```

```
esclavo Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
esclavo
115 }
116 else
117 {
118   digitalWrite(vr2, HIGH);
119 }
120 if(value3== LOW && value2== LOW)
121 {
122   digitalWrite(bomba, LOW);
123   nn=1;
124 }
125 }
126 digitalWrite(bomba, LOW);
127 }
```

## MAESTRO

Se declaran las variables que se utilizaran para el modulo maestro y los pines que servirán de entrada o salida:

```
PRACTICA_9 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
PRACTICA_9
1 /*MAESTRO*/
2 int start = I0_3;
3 int paro= I0_4;
4 int salidal = Q0_0;
5 int valor = 0;
6 int valor1= 0;
7
8 void setup()
9
10 {
11   pinMode(start, INPUT);
12   pinMode(paro, INPUT);
13   pinMode(salidal, OUTPUT);
```

```
PRACTICA_9 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
PRACTICA_9
14   Serial1.begin(9600);
15   Serial.begin(9600);
16
17 }
18
19 void loop()
20 {
21   //LEE EL VALOR DE LA ENTRADA I0_4 y I0_5
22   valor = digitalRead(start);
23   //si se pulsa I0_4 entonces se activa la comunicacion
24
25
26   if (valor == 1)
```

```
PRACTICA_9 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
PRACTICA_9
27 {
28   digitalWrite(salida1,HIGH);
29   Serial1.write("r");
30   Serial.write("r");
31   delay (4000);
32   Serial1.write ("1");
33   Serial.write("1");
34   delay (4000);
35 }
36 else
37 {
38   digitalWrite(salida1,LOW);
39 }
```

```
PRACTICA_9 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
PRACTICA_9
40 if (valor1 == 1)
41 {
42   digitalWrite(salida1,HIGH);
43   Serial1.write("r");
44   Serial.write("r");
45   delay (4000);
46   Serial1.write ("1");
47   Serial.write("1");
48   delay (4000);
49 }
50 else
51 {
52   digitalWrite(salida1,LOW);
```

```
PRACTICA_9 Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
Subir
PRACTICA_9
52   digitalWrite(salida1,LOW);
53 }
54
55 delay (25);
56 }
57
58
59
```

## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.

**CONCLUSIONES:**

Al realizar la puesta en marcha de la práctica se validó la activación de la recirculación del líquido en los tanques ubicados en la planta y a la vez se realizó el control en el módulo maestro mediante el entorno de programación IDE para la recirculación en los tanques.

**RECOMENDACIONES:**

Ninguna

		REVISION 1/1	Página 1 de -
		INFORME DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electronica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

TESIS:

“IMPLEMENTACIÓN DE MÓDULOS DE APRENDIZAJE ORIENTADOS  
A LA INSTRUMENTACIÓN FÍSICA Y VIRTUAL EN EL SECTOR  
INDUSTRIAL, MEDIANTE PLC’S ARDUINO”

INFORME

DE

PRÁCTICA # 10

TEMA:

CONTROL DE PLANTA DOSIFICADORA/MEZCLADORA DESDE  
EL MODULO MAESTRO.

- **OBJETIVO GENERAL:**

- Realizar el control de planta dosificadora/mezcladora desde el modulo maestro

- **OBJETIVOS ESPECIFICOS:**

- Utilizar librería R-LINK de Labview
- Realizar el control en Labview de la planta dosificadora/mezcladora
- Verificar la comunicación del módulo maestro

- **ACTIVIDAD.-**

- Desarrollar un programa para el módulo 38R (maestro), en el cual se realice el control de planta dosificadora/mezcladora.
- Implementar el programa en el lenguaje de programación mediante la ventana IDE de arduino.
- Ampliar el conocimiento sobre programación en el entorno IDE.

- **DESARROLLO.-**

Ejecución del programa:

Al ejecutar el programa, para el funcionamiento de la planta comienza presionando Start (I0.4) desde el módulo esclavo.

Una vez presionado Start se activa la bomba de descarga 1 (BD1), y Solenoide 1 (S\_1), donde empieza el vaciado del tanque 1 al tanque principal.

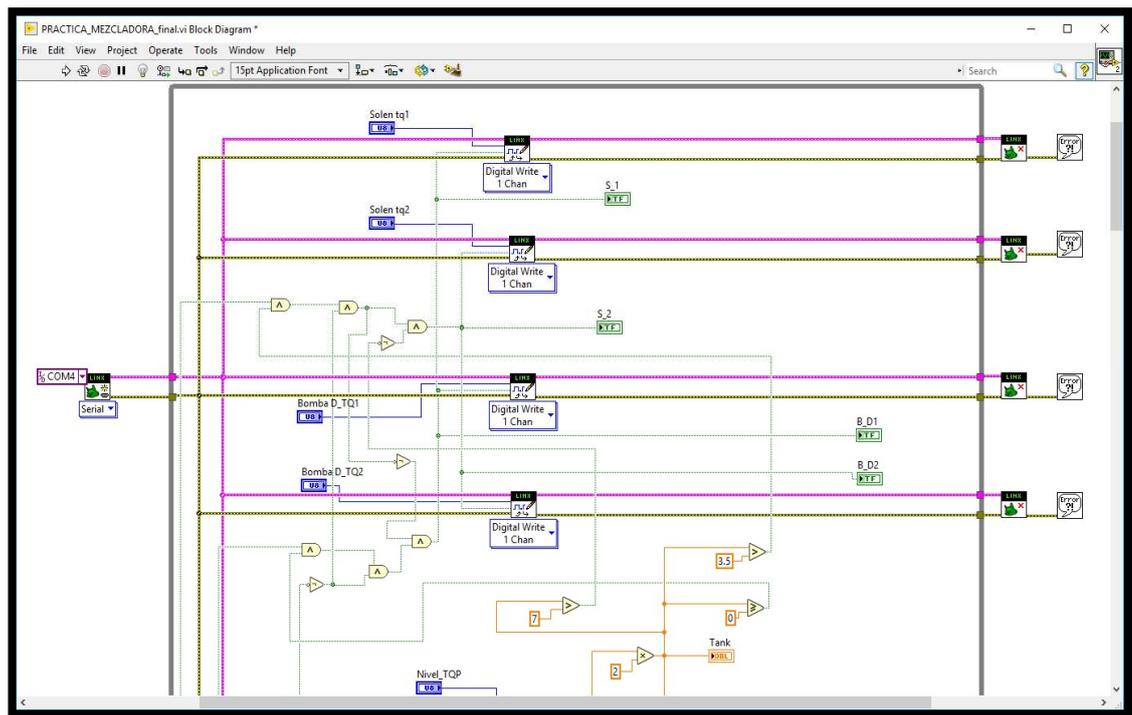


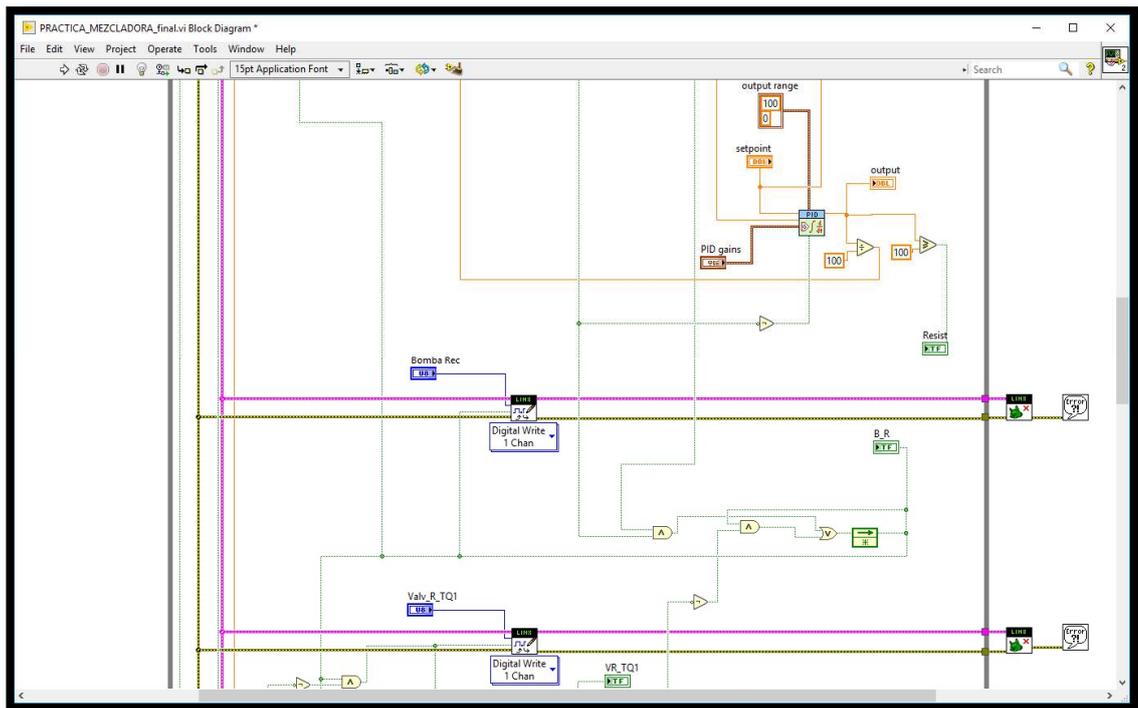
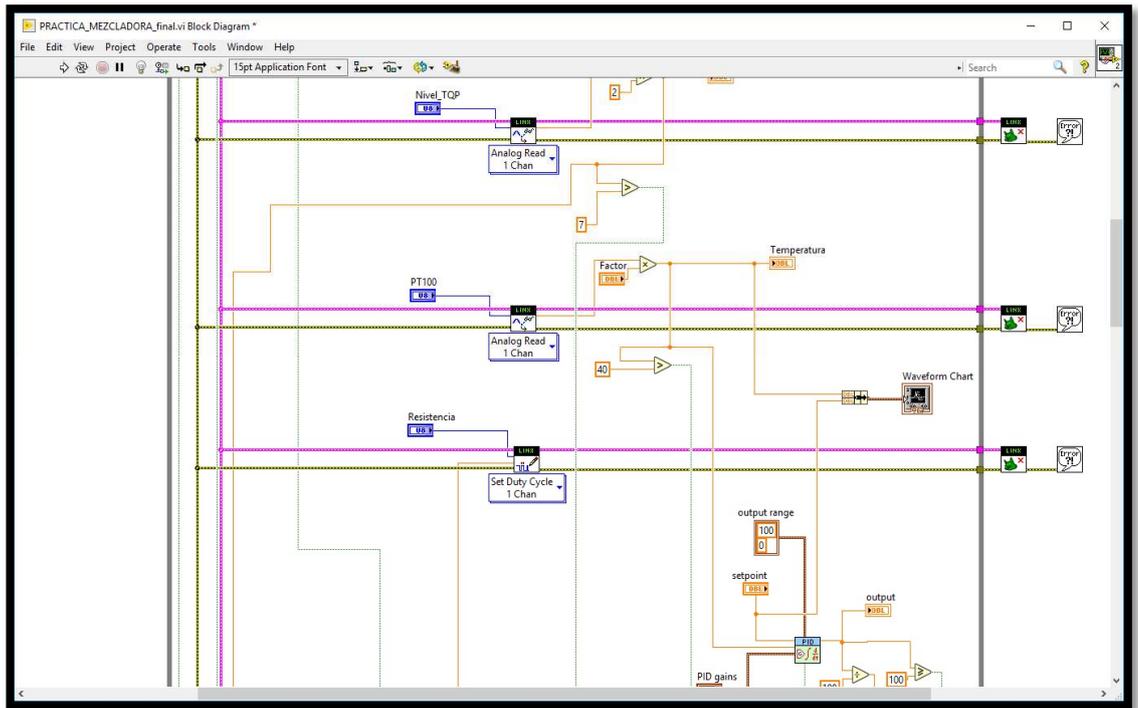
La siguiente figura se puede observar la pantalla en Labview de la práctica # 10 en modo OFF.

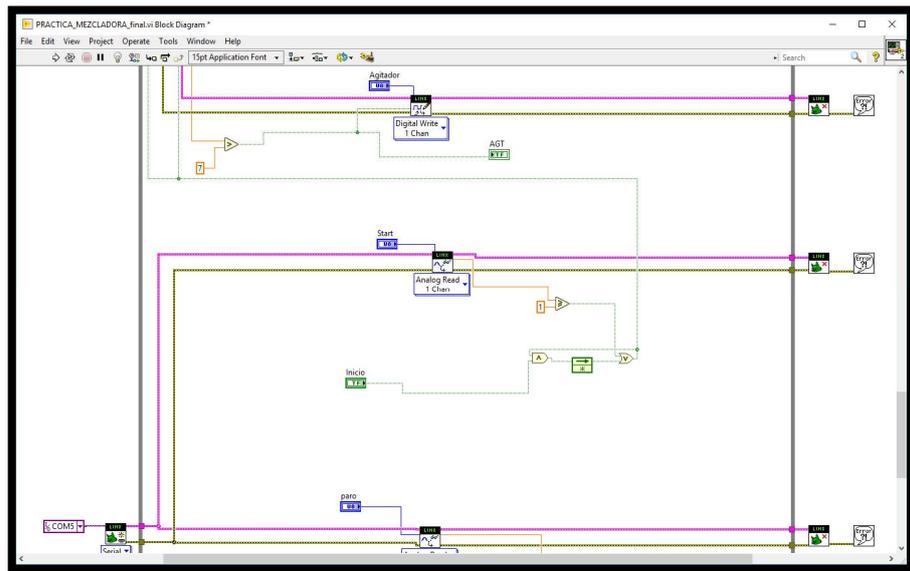
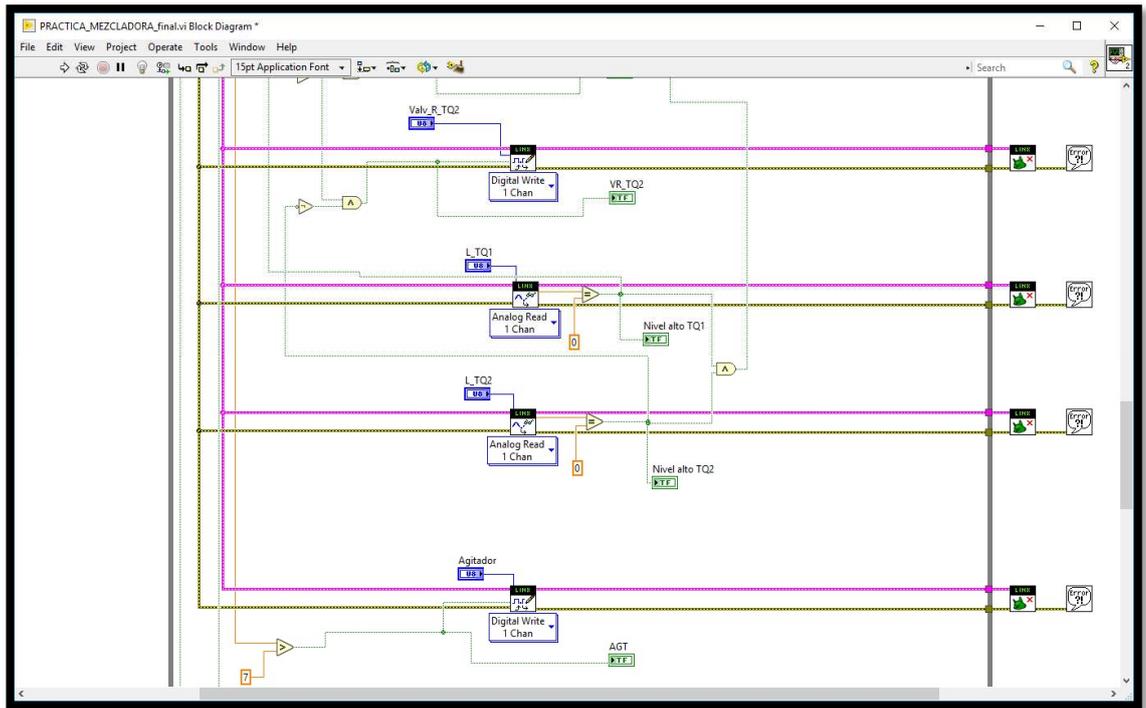


**PROGRAMA REALIZADO EN ENTORNO DE PROGRAMACIÓN IDE DE ARDUINO**

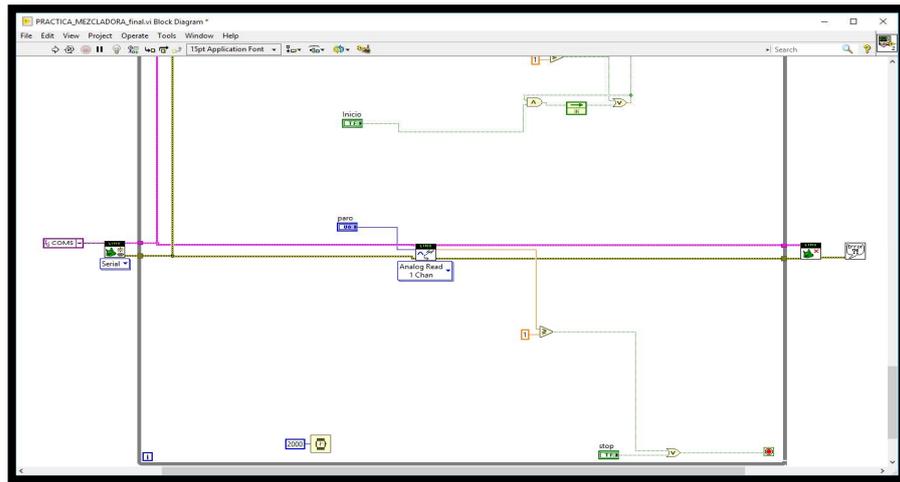
En las siguientes imágenes se muestra la programación en diagrama de bloques en Labview:







En la siguiente figura se realiza la programación en diagrama de bloques para la lectura de la entrada analógica del PLC:



Ejecución del programa:

Al ejecutar el programa, para el funcionamiento de la planta comienza presionando Start (I0.4) desde el módulo esclavo.

Una vez presionado Start se activa la bomba de descarga 1 (BD1), y Solenoide 1 (S\_1), donde empieza el vaciado del tanque 1 al tanque principal.



Cuando el agua en el tanque principal (Tank) llega a un nivel de 3,5 se desactiva (BD1) y (S\_1) y se activa la bomba de descarga 2 (B\_D2) y Solenoide 2 (S\_2) y empieza el vaciado del tanque 2.



Cuando el tanque principal llega a un nivel de 7:

- Se desactiva la bomba de descarga 2 (B\_D2) y Solenoide (S\_2).
- Se enciende el agitador mientras el nivel del tanque principal se mantenga mayor a 7.
- Funciona el PID y la resistencia empieza a calentar el agua en el tanque principal (Tank), y la temperatura comienza a elevarse.



Cuando la temperatura llega a 40°C:

- Empieza el proceso de recirculación.
- Se activa la bomba de recirculación (B\_R), válvula recirculación 1 (VR\_TQ1) y válvula recirculación 2 (VR\_TQ2).



Cuando ambos niveles del tanque 1 (Nivel alto TQ1) y tanque2 (Nivel alto TQ2) se activan:

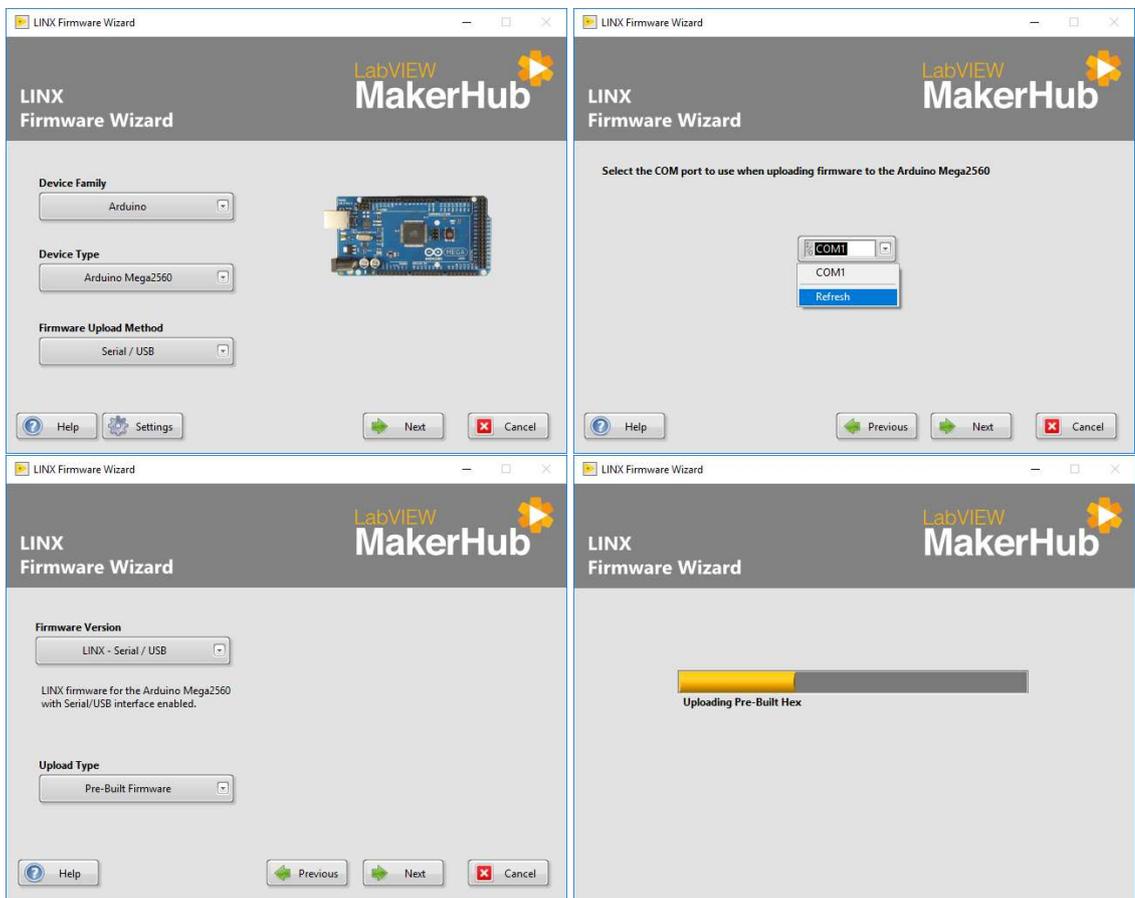
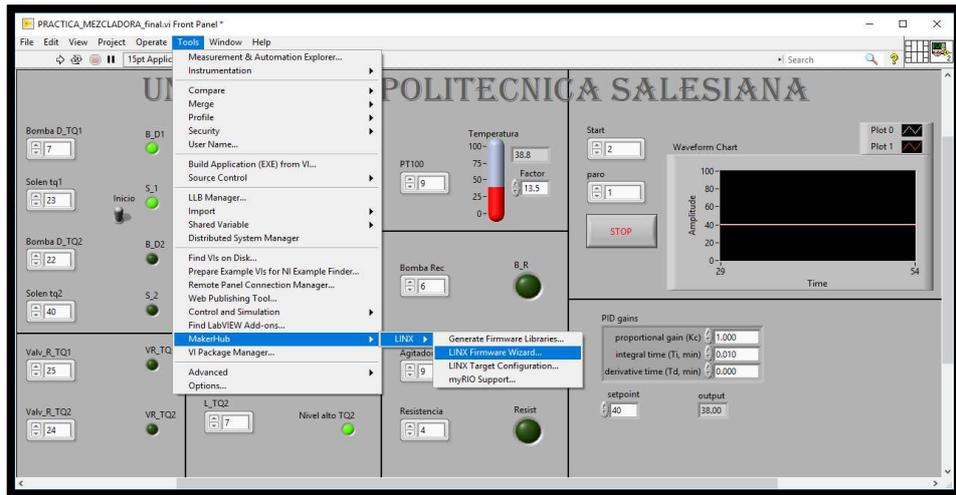
- Se desactiva válvula recirculación 1 (VR\_TQ1), válvula recirculación 2 (VR\_TQ2).
- Se desactiva bomba de recirculación (B\_R).
- Llenos tanque 1 y tanque 2 empieza nuevamente el proceso.

El proceso se detiene con el botón de Stop, desde el programa, o a través del módulo esclavo (IO.4).



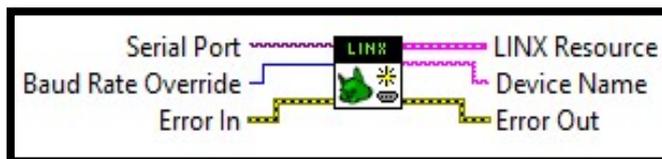
Pasos para ejecución del programa

- Para la comunicación entre LabView y la Planta se utilizó herramienta LINX, en las siguientes figuras se muestra como realizar el enlace entre Labview y dicho programa:

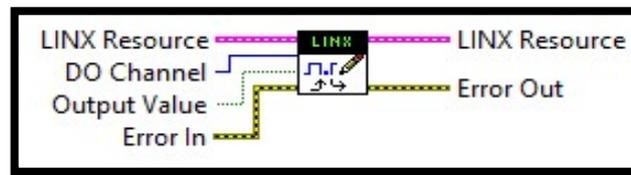


Es importante también identificar los bloques que se utilizaron en la programación de la Labview, entre ellas podemos mencionar las siguientes:

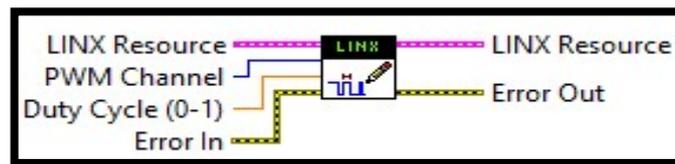
**Linx Open:** Abre una conexión en serie a un dispositivo LINX remoto



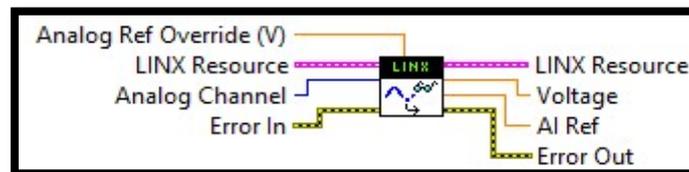
**Linx Digital Write:** Escribe valores en los canales de salida digital especificados.



**Set Duty Cycle:** Establece el ciclo de trabajo de los canales PWM especificados.  
Un ciclo de trabajo de 0 corresponde al 0% activado y un ciclo de trabajo de 1 corresponde al 100% activado.



**Analog Real:** Lee el valor del canal de entrada analógico especificado.



## RECURSOS UTILIZADOS (EQUIPOS, ACCESORIOS Y MATERIAL CONSUMIBLE)

- Computadores con el software Arduino IDE
- Tableros con PLC Mduino y accesorios de accionamiento
- Cable de conexión Ethernet.
- Cable Db25

## CONCLUSIONES:

Para realizar la práctica se utilizó la librería R-LINK de Labview, mediante la cual se procedió a establecer la lectura de las variables en las entradas analógicas, que se encuentran en la planta. Posteriormente se realizó el control en Labview de la planta dosificadora/mezcladora y se verificó la comunicación del módulo maestro hacia la planta, también al software labview. Con ello se demuestra que tanto los módulos como la planta están en un 100% habilitado para trabajar.

## RECOMENDACIONES:

Ninguna

## 8.- Bibliografía

- Arduino, P. (2020). *Proyecto Arduino*. Obtenido de <https://proyectoarduino.com/arduino-mega-2560/>
- catarina. (2020). *udlap\_interface grafica para medidor de nivel*. Obtenido de [udlap\\_interface grafica para medidor de nivel: http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lem/morales\\_h\\_oe/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/morales_h_oe/capitulo3.pdf)
- Controllino. (2020). *Controllino*. Obtenido de Controllino: <https://www.controllino.biz/>
- Didácticos, R. (2020). *Robots Didácticos*. Obtenido de <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>
- Electric, E. (19 de 02 de 2020). *Emerson*. Obtenido de Emerson: <https://www.emerson.com/es-es/industries/automation/packaging/primary-packaging/bottle-filling-sealing-testing>
- Electrónica, I. +. (2020). *I + D Electrónica*. Obtenido de I + D Electrónica: <https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/otros-sistemas/tarjetas-desarrollo/mini-pc-lattepanda-4gb-64gb-con-licencia-enterprise-no-activada-minicomputador-minipc-dfr0444-sistema-de-desarrollo-tarjeta-de-desarrollo-mini-pc-pa>
- electrónicos, E. (2020). *Equipos electrónicos*. Obtenido de <http://ebd90.blogspot.com/2017/06/el-bus-i2c.html?m=0>. Obtenido de <http://ebd90.blogspot.com/2017/06/el-bus-i2c.html?m=0>
- instruments, N. (2020). *National instruments*. Obtenido de National instruments: <https://www.ni.com/es-cr/innovations/platform.html>
- LATTEPANDA. (2020). *LATTEPANDA*. Obtenido de LATTEPANDA: <http://www.lattepanda.com/products/3.html>
- Llamas, L. (2018). *Ingenieria informatica y diseño*. Obtenido de Ingenieria informatica y diseño: <https://www.luisllamas.es/arduino-i2c/>
- Mecafenix. (2018). *ingmecafenix*. Obtenido de ingmecafenix: <https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>
- Rekursostic. (2015). *Rekursostic*. Obtenido de Recursostic: <http://recursostic.educacion.es>
- Rekursostic. (2018). Obtenido de <http://edu-hvcom.blogspot.com/p/curso-hvcom.html>
- Shields, I. (2018). *Industrial Shields*. Obtenido de Industrial Shields: [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1](https://www.industrialshields.com/es_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1)
- Shields, I. (2018). *Industrial Shields*. Obtenido de Industrial Shields: [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-19r-m-duino-plc-arduino-ethernet-19r-i-os-rele-analog-digital-plus-8?category=1](https://www.industrialshields.com/es_ES/shop/product/is-mduino-19r-m-duino-plc-arduino-ethernet-19r-i-os-rele-analog-digital-plus-8?category=1)
- Shields, I. (2020). *Industrial Shields*. Obtenido de Industrial Shields: [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-38r-m-duino-plc-arduino-ethernet-38r-i-os-rele-analogico-digital-plus-10](https://www.industrialshields.com/es_ES/shop/product/is-mduino-38r-m-duino-plc-arduino-ethernet-38r-i-os-rele-analogico-digital-plus-10)
- T.BEM. (2020). *T.BEM*. Obtenido de T.BEM: <https://teslabem.com/nivel-intermedio/fundamentos/>
- TOOLS, E. (19 de Junio de 2018). *E. TOOLS*. Obtenido de E. TOOLS: <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>
- AG, S. (2015). *Topologias de Red*. Obtenido de SIEMENS: [https://cache.industry.siemens.com/dl/files/313/36087313/att\\_74878/v1/BA\\_S7-1200-CSM1277\\_78.pdf](https://cache.industry.siemens.com/dl/files/313/36087313/att_74878/v1/BA_S7-1200-CSM1277_78.pdf)

- Allbiz. (2017). *Maquinas utomaticas para embotellar y taponar liquidos alimenticios*. Obtenido de Maquina alimentad: <http://buenos-aires.all.biz/maquina-alimentadora-de-envases-g127840#.WY2XWjPyjIU>
- arduino, E. (s.f.). *El arduino*. Obtenido de El arduino: <https://www.elarduino.com/mega/>
- Arduino, P. (s.f.). *Proyecto Arduino*. Obtenido de Proyecto Arduino: <https://proyectoarduino.com/arduino-mega-2560/>
- AutomationDirect. (Diciembre de 2012). USER MANUAL. *Alarm Codes*.
- AUTOMATIONDIRECT. (2017). *STP-DRV-6575*. Obtenido de AUTOMATION DIRECT.com: [https://www.automationdirect.com/adc/Shopping/Catalog/Motion\\_Control/Stepper\\_Systems/Stepper\\_Drives\\_-z-\\_Power/STP-DRV-6575](https://www.automationdirect.com/adc/Shopping/Catalog/Motion_Control/Stepper_Systems/Stepper_Drives_-z-_Power/STP-DRV-6575)
- Autores. (2019). Especificaciones Tecnicas Arduino Mega 2560. Guayaquil.
- Badinez, R. (junio de 2007). *Motores paso a paso*. Obtenido de Diseño y simulacion de un instrumento para la estimacion de torque de un motor paso a paso: [http://www.tesis.uchile.cl/tesis/uchile/2007/badinez\\_rl/sources/badinez\\_rl.pdf](http://www.tesis.uchile.cl/tesis/uchile/2007/badinez_rl/sources/badinez_rl.pdf)
- Barahona, B. (Febrero de 2011). *SELECCION DE MAQUINA ETIQUETADORA*. Obtenido de ESCUELA POLITECNICA NACIONAL.
- BrainChild. (2018). *Ethernet Wireless*. Obtenido de <https://www.brainchild.com.tw/en/product-226268/SW5502.html>
- Bustamante, E. (2017). <http://ebd90.blogspot.com/2017/06/el-bus-i2c.html?m=0>. Obtenido de <http://ebd90.blogspot.com/2017/06/el-bus-i2c.html?m=0>
- Cobena, R., & Salazar, G. (febrero de 2015). *Diseño e implemencacion de un sistema escala para el accionamiento y recepcion de señales discretas de campo mediante la red industrial AS-I para el Laboratorio de Fabricacion Flexible*. Obtenido de CPU Siemens 1212C: <http://dSPACE.ups.edu.ec/handle/123456789/10382>
- Collet, G. (02 de mayo de 2013). *Nuevas tendencias en automatizacion industrial*. Obtenido de Interempresas: <http://www.interempresas.net/Robotica/Articulos/108830-Nuevas-tendencias-en-automatizacion-industrial.html>
- controllino. (2020). *controllino*. Obtenido de controllino: <https://www.controllino.biz/>
- Crespo, W. (09 de febrero de 2011). *Automatizacion Industrial*. Obtenido de automatizacionindustrial: <https://automatizacionindustrial.wordpress.com/2011/02/09/queeslaautomatizacionindustrial/>
- Didácticos, R. (s.f.). *Robots Didácticos*. Obtenido de Robots Didácticos: <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>
- Direct Industry. (2017). *Transportador de cadena tipo table top*. Obtenido de Transportador de banda: <http://www.directindustry.es/prod/ngb/product-19754-391150.html>
- EATON. (2013). *EATON POWERING BUSINESS WORLDWIDE*. Obtenido de <http://www.eaton.cl/conosur/ProductosSoluciones/Energia/ProductosyServicios/AutomatizacionyControl/ConversionyManejodeEnergia/Fuentes/index.htm>
- Electric, E. (03 de 01 de 2020). *Emerson*. Obtenido de Emerson: <https://www.emerson.com/es-es/industries/automation/packaging/primary-packaging/bottle-filling-sealing-testing>
- Garcia, E. (1999). *Introduccion a las redes de comunicacion industriales*. Obtenido de Automatizacion de procesos industriales: <http://isa.umh.es/asignaturas/ci/Tema%201.pdf>
- HANYOUNG NUX. (septiembre de 2013). *PRODUCT SENSOR ENCODER*. Obtenido de [http://eng.hynux.com/sub01\\_01\\_01\\_view.php?lcode=05&mcode=0507&pcode=1305100091&scode=0507002](http://eng.hynux.com/sub01_01_01_view.php?lcode=05&mcode=0507&pcode=1305100091&scode=0507002)
- IFM. (2017). *O5H200 Sensores Fotoelectricos*. Obtenido de

- <http://www.ifm.com/products/mx/ds/O5H200.htm>  
 infoPLC. (18 de mayo de 2013). *Informe Ethernet Industrial: Profinet*. Obtenido de Automatizacion Industrial, Robotica e industria 4.0: <http://www.infopl.net/documentacion/7-comunicaciones-industriales/74-informe-ethernet-industrial>
- LattePanda. (2018). *LattePanda.com*. Obtenido de <http://www.lattepanda.com/products/3.html>
- LBA. (2017). *Tipos de Encoder*. Obtenido de LBA: <http://www.abm-industrial.com/2013/02/07/que-es-un-encoder/>
- LLamas, L. (Mayo de 2016). <https://www.luisllamas.es/arduino-i2c/>. Obtenido de <https://www.luisllamas.es/arduino-i2c/>
- MAQUINARIA ELECTRICA BILBAO. (2017). *Catálogo general motores trifásicos MEB S.A.* Obtenido de MAQUINARIA ELECTRICA BILBAO: <http://www.mebsa.com/pdf/Catalogo-General-Motores-Trifasicos-MEB.pdf>
- Mazzone, V. (Marzo de 2002). *CONTROLADORES PID*. Obtenido de <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
- Mecafenix, F. (2018). *ingmecafenix*. Obtenido de [ingmecafenix: https://www.ingmecafenix.com/automatizacion/que-es-un-plc/](https://www.ingmecafenix.com/automatizacion/que-es-un-plc/)
- Morales, M. (2017). *Teslabem*. Obtenido de <https://teslabem.com/nivel-intermedio/fundamentos-del-protocolo-i2c-aprende/>
- Nieto, E. (2017). *Sensores Fotoelectricos*. Obtenido de Fidestec : <http://fidestec.com/blog/sensores-fotoelectricos-industriales-fotocelulas/>
- PANASONIC. (2018). *LX-101-P*.
- plastimedia.Studio. (s.f.). *I+D electronica*. Obtenido de I+D electronica: <https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/otros-sistemas/tarjetas-desarrollo/mini-pc-lattepanda-4gb-64gb-con-licencia-enterprise-no-activada-minicomputador-minipc-dfr0444-sistema-de-desarrollo-tarjeta-de-desarrollo-mini-pc-pa>
- Recurstostic. (2015). *Recurstostic*. Obtenido de <http://recurstostic.educacion.es>
- recurstostic. (03 de 2018). *recurstostic*. Obtenido de HVCOM: <http://edu-hvcom.blogspot.com/p/curso-hvcom.html>
- Rios, J. (21 de Febrero de 2016). *Sistemas de control*. Obtenido de Diferencias entre lazo de control abierto y cerrado: <http://sistemadelazoabiertoycerrado.blogspot.com/>
- Rivadeneira, A. (Marzo de 2015). *Diseño e implementacion de un modulo didactico para un proceso de corte transversal y transporte de papel controlado y supervisado desde un sistema Scada diseñado en labview*. Obtenido de Accionamiento para motor paso a paso: <http://dSPACE.ups.edu.ec/bitstream/123456789/10390/1/UPS-GT001426.pdf>
- Schneider Electric. (20 de febrero de 2011). *Solucion integral para la automatizacion y el control de ensamblaje*. Obtenido de Info PLC: <http://www.infopl.net/descargas/178-schneider-electric/aplicaciones/391-aplicacion-solucion-integral-para-la-automatizacion-y-el-control-de-ensamblaje>
- SCHNEIDER ELECTRIC ARGENTINA. (2017). *Presentacion Variador Altivar 312*. Obtenido de Altivar 312 Variadores de Velocidad: <http://www.schneider-electric.com.ar/documents/local/altivar312.pdf>
- Shields, I. (2018). *Industrial Shields*. Obtenido de [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1](https://www.industrialshields.com/es_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1)
- Shields, I. (2018). *Industrial Shield*. Obtenido de [https://www.industrialshields.com/es\\_ES/industrial-plc-based-on-arduino-original-boards-automation-solutions](https://www.industrialshields.com/es_ES/industrial-plc-based-on-arduino-original-boards-automation-solutions)

- Shields, I. (2018). *Industrial Shields*. Obtenido de [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1](https://www.industrialshields.com/es_ES/shop/product/is-mduino-38ar-m-duino-plc-arduino-ethernet-38ar-i-os-analogico-digital-rele-plus-12?category=1)
- Shields, I. (2018). *Industrial Shields*. Obtenido de [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-19r-m-duino-plc-arduino-ethernet-19r-i-os-rele-analog-digital-plus-8?category=1](https://www.industrialshields.com/es_ES/shop/product/is-mduino-19r-m-duino-plc-arduino-ethernet-19r-i-os-rele-analog-digital-plus-8?category=1)
- Shields, I. (2020). *Industrial Shields*. Obtenido de Industrial Shields: [https://www.industrialshields.com/es\\_ES/shop/product/is-mduino-38r-m-duino-plc-arduino-ethernet-38r-i-os-rele-analogico-digital-plus-10](https://www.industrialshields.com/es_ES/shop/product/is-mduino-38r-m-duino-plc-arduino-ethernet-38r-i-os-rele-analogico-digital-plus-10)
- SICK. (2017). *Sensores de Horquilla*. Obtenido de Sensores de Registro: <https://www.sick.com/es/es/gama-de-productos/sensores-de-registro/sensores-de-horquilla/uf/c/g127489>
- SIEMENS. (2007). *Simatic HMI KTP600 Basic Color PN*. Obtenido de SIEMENS: <https://support.industry.siemens.com/cs/pd/297145?pdtd=td&lc=es-WW>
- SIEMENS. (noviembre de 2009). *Diagramas de conexion*. Obtenido de Tecnologia Electronica y Automatizacion.: <http://tecnologiaelectronicayautomatizacion.blogspot.com/2016/06/instalacion-de-un-plc-simatic-s7-1200.html>
- SIEMENS. (abril de 2011). *S7-1200 Easy Book-Manual de Producto*.
- SIEMENS. (31 de Enero de 2012). *Modulos Digitales*. Obtenido de ISSUU: <https://issuu.com/softwhisper/docs/named2b8a4>
- SIEMENS. (03 de mayo de 2013).
- SIEMENS. (01 de marzo de 2016). *Switches Scalance X. Comunicacion Industrial SIMATIC NET*. Guayaquil, Ecuador.
- SIEMENS. (01 de marzo de 2016). *Switches Calonche X. Comunicacion industrial SIMATIC NET*. Guayaquil, Ecuador.
- SIEMENS AG. (2005). *Aspectos generales SINAMICS G110*. Obtenido de SINAMICS G110: [https://www.technical.cat/PDF/Siemens/MECATRONICA/manual\\_sinamics\\_G110.pdf](https://www.technical.cat/PDF/Siemens/MECATRONICA/manual_sinamics_G110.pdf)
- SIEMENS ECUADOR. (01 de Mayo de 2017). *SINAMICS V20*. Guayaquil, Ecuador.
- Sobrevilla Gonzalez, M. (21 de agosto de 2012). *Arquitectura tipica de automata programable*. Obtenido de Sistema Automatizado PLCs: <https://es.slideshare.net/alexoluthor/plcs-14035728>
- SOCETTI. (27 de AGOSTO de 2017). *SINAMICS G110*. Obtenido de SIEMENS: <http://www.socetti.cl/variadores%20de%20frecuencia%20siemens%20G110%20220V.html>
- soloelectronicos. (s.f.). *soloelectronicos*. Obtenido de soloelectronicos: <https://soloelectronicos.com/tag/visual-studio/>
- Universidad del Cauca. (2017). Obtenido de CONTROLADORES PID: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Sistemas%20de%20Ctrl%20Electronica/pid.pdf>
- Veloso, C. (19 de Junio de 2018). *Electrontools.com*. Obtenido de <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>
- WAGO. (2017). *DISPOSITIVOS DE ACTIVACION*.
- WAGO. (2018). Obtenido de Industrial-ECO-Switch: <https://www.wago.com/global/industrial-switches/industrial-eco-switch/p/852-1111>
- WAGO. (2018). Obtenido de <https://www.wago.com/global/industrial-switches/industrial-eco-switch/p/852-1111>

WAGO. (2018). *Industrial-ECO-Switch*. Obtenido de <https://www.wago.com/global/industrial-switches/industrial-eco-switch/p/852-1111>