



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**DISEÑO DE UNA TARJETA DE ADQUISICIÓN DE DATOS
PARA CONTROL SUPERVISORIO Y DIAGNÓSTICO EN
PROCESOS DE MÚLTIPLES LAZOS USANDO JAVA COMO
INTERFAZ GRÁFICA**

Autor: Miguel Freytes

Tutor: MSc. José Pérez Colón

Urb. Yuma II, calle N.^a 3. Municipio San Diego
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO DE UNA TARJETA DE ADQUISICIÓN DE DATOS
PARA CONTROL SUPERVISORIO Y DIAGNÓSTICO EN
PROCESOS DE MÚLTIPLES LAZOS USANDO JAVA COMO
INTERFAZ GRÁFICA**

Proyecto del Trabajo de Grado presentado para optar al título de

INGENIERO ELECTRÓNICO

Autor:

Miguel Freytes

C.I. 27.919.861

Tutor: MSc. José Pérez Colón

San Diego, julio de 2021



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERIA
ESCUELA DE ELECTRÓNICA**

**CONSTANCIA DE APROBACIÓN PARA LA PRESENTACIÓN PÚBLICA
DEL TRABAJO DE GRADO**

Quien suscribe, MSc. José Pérez Colón, portador(a) de la cédula de identidad N° 8.829.908, en mi carácter de tutor (a) del trabajo de grado presentado por el(la) los ciudadano(a) Miguel Freytes, portador(es) de la cédula de identidad N° 27.919.861, titulado **DISEÑO DE UNA TARJETA DE ADQUISICIÓN DE DATOS PARA CONTROL SUPERVISORIO Y DIAGNÓSTICO EN PROCESOS DE MÚLTIPLES LAZOS USANDO JAVA COMO INTERFAZ GRÁFICA** presentado como requisito parcial para optar al título de Ing. Electrónico, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 6 días del mes de mayo del año dos mil veintidos.

MSc. José Pérez Colón
8.829.908



UNIVERSIDAD JOSÉ ANTONIO PÁEZ
COORDINACIÓN DE PASANTÍA Y TRABAJO DE GRADO

ACTA DE APROBACIÓN

INFORME FINAL DE PASANTÍA

TRABAJO DE GRADO

El jurado designado por la Facultad de Ingeniería para la evaluación del Informe Final de Pasantía o Trabajo de Grado titulado:

Diseño de un Proyecto de Abstracción de Datos para Control Supervisado y Diagnóstico en Proceso de Múltiples Pasos usando JAVA como interfaz gráfica.

Realizado por el (la) Br. Miguel Freytes.

C.I. N° 27.919.861 cursante de la carrera de Ing. Electrónica

hace constar después de analizar su contenido y oír la exposición oral, considera que el Informe Final o Trabajo de Grado ha obtenido la calificación de:

APROBADO

NO APROBADO

El Jurado

[Signature]
Tutor Académico (Coordinador)
Nombre: 8.829.908
C.I.: 8.829.908

[Signature]
Jurado
Nombre: GERSON Sánchez
C.I.: V-7173 386

[Signature]
Jurado
Nombre: Wilmer Saiz
C.I.: 7130496

Fecha: 01/06/2022



UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
DECANATO DE INGENIERÍA



FI E 004 2022-1CR TG

Valencia, 27 de abril de 2022

Ciudadano:
FREYTES MORENO, MIGUEL ANDRES
27.919.861

Presente -

Cumplo con informarle que la comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 3-2022 de fecha 16/02/2022 aprobó el proyecto de grado titulado:

Diseño de una tarjeta de adquisición de datos para control supervisorio y diagnóstico en procesos de múltiples lazos usando java como interfaz gráfica

Presentado por usted como requisito para optar al título de Ingeniero Electrónico

Se ratifica la designación del Tutor Académico que lo asesorará en el desarrollo de este proyecto a:
Ing. José Ramon Pérez Colon, titular de la cédula de identidad V- 8.829.908



Atentamente

Dr. Francisco Gelanzé Sevilla.
Decano de Ingeniería

c.c. Coordinación de Pasantías y Trabajo de Grado

DEDICATORIAS

A mis padres, por brindarme todo el apoyo en cada una de las situaciones que se han presentado no solo en el desarrollo de este trabajo de grado si no también en el desarrollo de mi carrera universitaria y en la vida.

A mi tutor y profesor el MSc. José Pérez Colón, sin el cual no hubiera sido posible materializar todas las ideas propuestas al inicio del desarrollo de esta investigación y el cumplimiento de todos los objetivos propuestos desde un comienzo.

A mi familia y amigos, en especial a mis compañeros de promoción José Ramos y César Romero quienes me han acompañado a recorrer este camino desde el momento en que inicio mi carrera universitaria.

Finalmente, a todos los profesores que han sido parte de mi formación y han puesto un granito de arena para convertirme en el profesional y en la persona que soy hoy en día.

INDICE

CONTENIDO	PP.
ÍNDICE DE CUADROS.....	vii
INDICE DE FIGURAS.....	vii
RESUMEN.....	X
INTRODUCCIÓN	1
CAPÍTULO	
I EL PROBLEMA	
1.1 Planteamiento del Problema.....	3
1.2 Formulación del Problema.....	5
1.3 Objetivos de la Investigación.....	5
1.3.1 Objetivo General.....	5
1.3.2 Objetivos Específicos.....	5
1.4 Justificación del Problema	6
1.5 Alcance	6
1.6 Limitaciones.....	6
II MARCO TEÓRICO	
2.1 Antecedentes	7
2.2 Bases Teóricas	10
2.2.1 Sistemas de Adquisición de Datos.....	10
2.2.2 Presión.....	11
2.2.3 Temperatura	12
2.2.4 Nivel.....	13
2.2.5 Caudal	14
2.3 Bases legales	21
2.4 Definición de términos básicos	22
III MARCO METODOLÓGICO	
3.1 Tipo de Investigación.....	23
3.2 Diseño de la investigación	23
3.3 Nivel de la Investigación.....	24
3.4 Población y muestra.....	24
3.4.1 Población.....	24
3.4.2 Muestra	24
3.5 Técnica de recolección de datos	25
3.5.1 Observación directa	25
3.5.2 Observación documental.....	25
3.6 Instrumento de recolección de datos	26
3.6.1 Lista de Chequeo.....	26
3.6.2 Ficha de Trabajo	26

3.7 Fase metodológica	26
Fase I. Diagnóstico.....	27
Fase II. Diseño de la Tarjeta de Adquisición de Datos.....	27
Fase III. Desarrollo de la Interfaz Gráfica.....	27
Fase IV. Factibilidad.....	28
IV RESULTADOS	
4.1 Fase I. Diagnóstico.....	29
4.2 Fase II. Diseño de la Propuesta.....	42
4.2.1 Diseño de la Tarjeta de Adquisición de Datos.....	42
4.2.2 Programación del Microcontrolador	69
4.3 Fase III. Desarrollo de la Interfaz Gráfica	86
4.3.1 Programación de la HMI.....	86
4.3.2 Simulación de la Propuesta.....	95
4.4 Fase IV. Factibilidad.....	97
4.4.1 Estudio de Costos.....	98
4.4.2 Factibilidad Técnica.....	103
4.4.3 Factibilidad Operativa.....	103
V CONCLUSIONES Y RECOMENDACIONES	
5.1 Conclusiones	105
5.2 Recomendaciones	107
REFERENCIAS BIBLIOGRÁFICAS.....	108
APÉNDICES	
APÉNDICE A	111

ÍNDICE DE CUADROS

CUADRO	Pp
1 Lista de chequeo empleada para el análisis de los diagramas	35
2 Tabla Comparativa Modelos de Arduino	38
3 Códigos de acción del protocolo de comunicación diseñado.....	72
4 Recursos necesarios para la implementación de la propuesta	98
5 Cuadro Comparativo entre la UIAD y alternativas comerciales	102

INDICE DE FIGURAS

FIGURA	Pp
1 Características de Sistemas de Adquisición de Datos comerciales.....	4
2 Características de los Transductores de Presión	12
3 Características de los Transductores de Temperatura.....	13
4 Características de los Transductores de Nivel	14
5 Características de los Transductores de Flujo.....	15
6 Esquema de un sistema de medición.....	15

7 Diagrama convencional del Lazo de corriente de 4 – 20 mA	17
8 Diagrama de acondicionamiento de una señal al rango del ADC.....	19
9 Código de identificación de instrumentos. ISA-S5.1-84 (R-1992).....	30
10 Símbolos de actuadores y tipos de señales. ISA-S5.1-84 (R-1992).....	31
11 Diagrama de Control de Columna de Destilación.....	32
12 Diagrama de Control de Reactor Químico.....	32
13 Diagrama de Control de Evaporador Triple.....	33
14 Diagrama de Control de Desacople para Evaporador	33
15 Diagrama de Control de Caldera con Control Fino de Oxígeno.....	34
16 Diagrama de Control de Sistema de Regeneración.....	35
17 Diagrama de funcionamiento de los canales de entrada	358
18 Diagrama funcional del demultiplexor 74HC238.....	40
19 Esquema del Integrado LM324N.....	41
20 Hoja de datos del LM7805.....	41
21 Borneras de 2 y 3 Pines.....	42
22 Diagrama del Integrado 4066.....	43
23 Esquema del circuito empleando el demultiplexor 74HC238.....	44
24 Tabla de la verdad del demultiplexor 74HC238	44
25 Pines de control del acondicionamiento de las señales de la UIAD	45
26 Acondicionamiento de la señal de 4 mA a 20 mA.....	46
27 Pines Analógicos del ADC reservados	47
28 Acondicionamiento de la señal de 10 mA a 50 mA.....	48
29 Acondicionamiento de la señal de -5V a 5V; Atenuación de la señal	49
30 Acondicionamiento de la señal de -5V a 5V; Adición del offset de DC	50
31 Acondicionamiento de la señal de -5V a 5V.....	50
32 Acondicionamiento de la señal de 0 V a 10 V	52
33 Acondicionamiento de la señal de 1 V a 5 V	52
34 Alimentación de 5V de la UIAD empleando el LM7805	53
35 Pines PWM de salida de la UIAD.....	54
36 Salida a transistor de los canales de la UIAD	54
37 Diagrama para PCB de la UIAD	56
38 Modelo 3D de la UIAD.....	57
39 Carcasa de almacenamiento de la UIAD	58
40 Diagrama Esquemático de la UIAD; Microcontrolador y Alimentación.....	59
41 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 0	60
42 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 1	61
43 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 2	62
44 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 3	63
45 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 4	64

46 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 5	65
47 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 6	66
48 Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 7	67
49 Diagrama Esquemático de la UIAD; Canales de Salida	68
50 Diagrama de flujo del algoritmo de comunicación	69
51 Código del microcontrolador del algoritmo de comunicación.....	71
52 Diagrama del protocolo de comunicación entre la UIAD y la HMI	71
53 Separación del mensaje recibido según el delimitador	73
54 Respuesta del mensaje en caso de que el código sea 00	74
55 Respuesta del mensaje en caso de que el código sea 01 y una entrada	77
56 Función que define los pines a emplear dependiendo del parámetro.....	79
57 Respuesta del mensaje en caso de que el código sea 01 y el canal sea una salida.	80
58 Función que define la frecuencia de salida del PWM.....	81
59 Respuesta del mensaje en caso de que el código sea 02	83
60 Diagrama de flujo del algoritmo para leer los canales especificados	84
61 Respuesta del mensaje en caso de que el código sea 03	85
62 Función que define el pin a controlar dependiendo del parámetro pasado	86
63 Estructura física de la HMI	88
64 Archivo de Configuración de la HMI	89
65 Tabla de Canales de la HMI.....	90
66 Pestaña “Archivo” de la HMI	90
67 Ventana de Configuración de la UIAD.....	91
68 Ventana de Configuración del Tipo de Canal	91
69 Ventana de Configuración de los datos del Canal	92
70 Ventana de Confirmación de la Verificación de un Sensor	92
71 Ventana de Mediciones con diagrama de Planta.....	93
72 Ventana de Mediciones con Gráfica	94
73 Manual de la UIAD y HMI integrado en la interfaz	95
74 Conexión exitosa de la HMI con el puerto virtual de Proteus	96
75 Verificación exitosa de un sensor con una señal de 0V a la entrada.....	97
76 Curva obtenida al aplicar una señal de voltaje senoidal al Canal 1	97
77 Características y Costos de los equipos de la empresa Measurement Computing.	99
78 Características y Costos de los equipos de la empresa DATAQ Instruments.....	100
79 Características y Costos de los equipos DAQ USB de la LabJack	101



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERIA
ESCUELA DE ELECTRÓNICA

Autor: Miguel Freytes

Tutor: MSc. José Pérez Colón

Fecha: Julio, 2021

RESUMEN

Los sistemas de adquisición de datos son uno de los elementos más importantes de la industria de la actualidad, ya que a través de ellos se realiza la toma de mediciones de las magnitudes involucradas en los procesos para su registro y posterior control. Sin embargo, a pesar de su remarcada importancia, dichos sistemas de adquisición de datos, por sus siglas SAD, poseen un costo elevado, limitando su uso a procesos complejos e industrias con un gran poder adquisitivo. Razón por la cual este trabajo de investigación tiene como objetivo el diseño de una tarjeta SAD para control supervisorio y diagnóstico en procesos de múltiples lazos, empleando Java como interfaz gráfica, teniendo presente el coste de sus componentes para ofrecer a la industria en general un equipo comercialmente competitivo, de bajo coste y que cumpla con los estándares en tecnología de sistemas de medición, registro y control usado en la industria en la actualidad. La línea de investigación del presente estudio es Ciencias Cognitivas y Aplicadas, mientras que metodológicamente es una investigación tipo proyecto factible, sustentada en una investigación de campo de nivel descriptivo con un enfoque cuantitativo, la cual tuvo como población los sistemas de adquisición de datos que cumplen labores de control supervisorio y diagnóstico en procesos de múltiples lazos, tomando como muestra significativa a los dispositivos electrónicos de adquisición de datos de bajo costo alternativos a los equipos comerciales. Como resultado se espera que el diseño de la tarjeta de adquisición de datos sea factible y tenga un costo reducido confirmando su viabilidad.

Descriptor: sistemas de adquisición de datos, automatización industrial, control de procesos.

INTRODUCCIÓN

El desarrollo industrial en la actualidad se ha visto acelerado durante los últimos años, motivado principalmente por las diferentes innovaciones tecnológicas que han sido implementadas en todos los niveles y áreas del sector, siendo una de las más favorecidas por este fenómeno el área de la automatización industrial, donde los sistemas de adquisición de datos han representado una gran revolución debido a su incremento en funciones y posibilidades, apoyado por la implementación de microcontroladores con mayor capacidad de cómputo y algoritmos de control mejorados con inteligencia artificial. Los sistemas de adquisición de datos son aquellos sistemas electrónicos que se encargan de procesar magnitudes físicas captadas mediante sensores para diversos fines, donde el control y diagnóstico de procesos son las funciones de mayor demanda.

No obstante, la mejora en los sistemas de adquisición de datos e incremento de sus funciones, también aumenta proporcionalmente la complejidad de los mismos y su costo, dejando de ser herramientas asequibles para convertirse en instrumentos destinados a un público más selecto, por esta razón surge la necesidad de elaborar esta investigación cuyo objetivo es el de proponer el diseño de una tarjeta de adquisición de datos de bajo costo que cumpla con las necesidades básicas de control y diagnóstico de un determinado proceso sin dejar de lado la versatilidad y funcionalidad del equipo, siendo capaz de registrar y controlar las variables de proceso de mayor importancia, mediante una interfaz gráfica o HMI desarrollada en el lenguaje de programación Java, permitiendo su compatibilidad con una amplia variedad de dispositivos y sistemas operativos , además de poseer un diseño interactivo y amigable incluso para usuarios inexpertos.

El trabajo está estructurado en cuatro capítulos:

Capítulo I, El Problema, en el cual se describe el planteamiento, formulación, objetivos, justificación, alcance y limitaciones.

Capítulo II, Marco Teórico, donde se encuentran los antecedentes, bases teóricas, bases legales y definición de términos básicos.

Capítulo III, Marco Metodológico, tipo, diseño, nivel, población y muestra, técnicas e instrumentos de recolección de datos y las fases metodológicas.

Capítulo IV, Resultados, en el cual se describe el desarrollo de las fases metodológicas y sus respectivos productos.

Capítulo V, Conclusiones y Recomendaciones, en donde se detallan las consecuencias de los resultados obtenidos y posibles mejoras a la investigación.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del Problema

Los procesos industriales son cada vez más complejos. Antiguamente los sistemas consistían de una o dos variables a controlar como por ejemplo la presión y la temperatura en la máquina de vapor de los primeros ferrocarriles del siglo XVIII. Hoy día todos los procesos son de múltiples entradas y salidas, es decir, requieren diversos tipos de energías para su sustentación y diversos tipos para su control. Basta recordar que para que una variable pueda ser controlada en su valor adecuado, la misma debe ser medible, así que cada variable se puede decir que representa un lazo o punto de control de un proceso más grande. Según lo anterior los procesos están constituidos por múltiples lazos a controlar y que por ende requieren ser medidos y esas mediciones se transforman en datos numéricos, tendencias y gráficos que nos hablan del comportamiento del proceso en general y permiten tomar decisiones.

El avance tecnológico industrial se ha convertido en un proceso que evoluciona en la implementación de sistemas de adquisición de datos con la finalidad de realizar acciones de control y diagnóstico sobre las diversas magnitudes físicas que son de interés en la industria. Dichas magnitudes reciben el nombre de señales y son provenientes de las variables involucradas en un proceso y que se adquieren través de sensores o transductores, que son capaces de convertir diversos tipos de energía en señales eléctricas. Estas señales eléctricas son acondicionadas y digitalizadas por una tarjeta de adquisición de datos, mediante Microcontroladores o DSP, la cual transfiere los resultados a un computador, en el cual el usuario puede interactuar con estos datos mediante una Interfaz Hombre-Máquina o HMI desarrollada con un lenguaje de programación.

En la actualidad, existen múltiples

opciones de sistemas de adquisición de datos en el mercado que cumplen funciones complejas, sin embargo, dichos sistemas suelen requerir ser operados por técnicos muy especializados y con costo elevado sobre todo si el sistema a controlar es de múltiples lazos. A. Suarez (2019) en su trabajo de grado recopila una serie de sistemas de adquisición de datos y sus características (Ver figura 1), entre las que destaca la columna correspondiente al costo de los mismos, donde el más básico de ellos poseía un costo de 99\$ al momento de la publicación del trabajo, mientras que el más avanzado tenía un costo aproximado de 630\$, evidenciando el alto costo de esta clase de dispositivos.

Serie	Entradas analógicas			Salidas analógicas		Precio
	Canales	Resolución	Muestreo	Canales	Resolución	
USB-200	8se	12 bits	500kS/s	2	12 bits	99
USB-230	8se/4diff	16 bits	100kS/s	2	16 bits	260
MCC118	8se	12 bits	100kS/s	-	-	99
MCC152	-	-	-	2	12 bits	99
1208FS	8se/4diff	12-14 bits	50kS/s	2	10-12bits	129
1608FS	8se	16 bits	100kS/s	-	-	420
1608G	16se/8diff	16 bits	500kS/s	2	16 bits	420
E-1608	8se/4diff	16 bits	250kS/s	2	16 bits	505
1208LS	8se/4diff	12 bits	47kS/s	2	12 bits	189
Temp-tc	8diff	24 bits	2S/s/ch	-	-	375
E-tc	8diff	24 bits	4S/s/ch	-	-	505
2408	16se/8diff	24 bits	1kS/s	2	16 bits	630
1208HS	8se/4diff	13 bits	1MS/s	4	12 bits	525
DT9812	8se	12 bits	100kS/s	2	12 bits	415
DT9813	16se	12 bits	100kS/s	2	12 bits	495
DT9814	24se	12 bits	50kS/s	2	12 bits	570
DT9816	6se	16 bits	750kS/s	-	-	510
DAS08	8se	12 bits	40kS/s	-	-	280
DAS16	16se/8diff	16 bits	1kS/s	2	12 bits	525

Figura 1: Características de Sistemas de Adquisición de Datos comerciales

Fuente: Suarez (2019)

Por otra parte, en ocasiones es requerido el uso de dispositivos adicionales para el funcionamiento de los sistemas de adquisición de datos, esto se debe a múltiples factores, uno de ellos es la necesidad de amplificación y linealización de las señales por parte de algunos sensores, funciones que no suelen estar presentes en los sistemas de adquisición de datos más sencillos. Adicionalmente, algunos sistemas de adquisición de datos presentan problemas de compatibilidad, tanto de controladores, como de puertos, limitando su uso a unas determinadas plataformas o sistemas

operativos, excluyendo el resto de ellos. Como consecuencia se tienen, sistemas de adquisición de datos con interfaces HMI muy complejas, problemas de compatibilidad con determinados sistemas y plataformas, poco interactivos y no amigables con usuarios inexpertos, implicando que el usuario tenga que invertir con alto costo en sistemas que no se adaptan a sus requerimientos específicos.

1.2 Formulación del Problema

Teniendo en cuenta lo planteado anteriormente, surge la siguiente interrogante: ¿Es posible el control supervisorio y diagnóstico en procesos de múltiples lazos sin el uso de equipos comerciales de alto costo obteniendo las mismas prestaciones y resultados?

1.3 Objetivos de la Investigación

1.3.1 Objetivo General

Diseñar una tarjeta de adquisición de datos para control supervisorio y diagnóstico en procesos de múltiples lazos usando Java como interfaz gráfica.

1.3.2 Objetivos Específicos

- Determinar las necesidades mínimas de un proceso de múltiples lazos en cuanto a número de entradas y salidas, características de las señales a acondicionar, así como los componentes electrónicos necesarios para dicho acondicionamiento.
- Diseñar el sistema electrónico y el algoritmo de adquisición de datos empleando un lenguaje informático apropiado para la programación del microcontrolador.
- Desarrollar una interfaz gráfica interactiva haciendo uso del lenguaje de programación Java.
- Realizar un estudio de la factibilidad técnica, operativa y un análisis de costos para la implementación del proyecto.

1.4 Justificación del Problema

La finalidad de esta investigación es la de brindar a la industria, una herramienta capaz de realizar diversas tareas de control y supervisión, mediante el empleo de una tarjeta de adquisición de datos de fácil uso e instalación, compatible con una multitud de dispositivos y sistemas, aunado a su interfaz de control interactiva con múltiples opciones y un costo reducido, permitiendo su implementación por parte tanto de usuarios experimentados en el área como usuarios inexpertos. Finalmente, se espera que el presente trabajo sirva como antecedentes o base para la realización de futuras investigaciones en el campo de la ingeniería de control u otras áreas relacionadas con el tema.

1.5 Alcance

La presente investigación pretende abarcar sólo el diseño de una tarjeta de adquisición de datos con: las diferentes magnitudes que será capaz de registrar, la conversión de las señales analógicas y digitales, y por último el microcontrolador a utilizar. Adicionalmente, se pretende simular el proceso de control de una planta de múltiples lazos, utilizando una interfaz gráfica diseñada con el lenguaje de programación Java que permita el diagnóstico y control supervisorio de la misma, comunicándose con el diseño de la tarjeta de adquisición de datos elaborado en el simulador de circuitos electrónicos Proteus.

1.6 Limitaciones

El diseño de la tarjeta de adquisición de datos presentado en esta investigación estará limitado a un número reducido de magnitudes y sensores compatibles debido a cuestiones de recursos y temporales. Por otra parte, la interfaz gráfica a diseñar será compatible solo con computadores que cumplan con ciertos requisitos mínimos, tanto de espacio en disco disponible, suficiente memoria RAM, así como un puerto USB libre para la conexión de la tarjeta de adquisición de datos.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Una vez ya planteado los objetivos de la investigación, se iniciaron las búsquedas previas referentes a investigaciones anteriores con similitudes al problema ya planteado, de las cuales destacan las siguientes:

Primeramente, Pertuz (2018) de la facultad de Ingeniería de la Universidad Nacional Abierta y a Distancia en Barranquilla, Colombia, en su trabajo para optar al título de Ingeniero en Sistemas, titulado **“Análisis y Diseño de Aplicación para la Gestión Económica y de Servicios de la empresa Vital Plus”** tuvo como objetivo general, analizar y diseñar una aplicación de escritorio que permita llevar un control de los tratamientos odontológicos y las historias clínicas de los pacientes de la empresa Vital Plus, con el fin de optimizar su prestación de servicios en salud oral, siendo de relevancia para la presente investigación, ya que en esta obra se describe el desarrollo de una aplicación de escritorio usando el lenguaje de programación Java, que, según menciona el autor, sus principales ventajas son su versatilidad, su habilidad de permanencia y su fácil legibilidad, llevando a que el autor concluyera que el desarrollo de una aplicación contribuye a que la prestación de servicios por parte de la empresa sea más óptima y satisfactoria, estableciendo un precedente para el uso de aplicaciones con la finalidad de aumentar la productividad de una determinada empresa, que si adicionalmente son desarrolladas en Java, les brindará mayor soporte y compatibilidad a los usuarios.

Por otra parte, González, Pérez y Dunia (2018) en su artículo científico titulado como **“Diseño de un sistema de adquisición y procesamiento de datos con el uso de multisensores para laboratorios de física universitarios”** para la revista Ingeniería UC de la facultad de Ingeniería de la Universidad de Carabobo, cuyo objetivo era presentar el diseño y construcción de un sistema de adquisición y procesamiento de datos empleando tecnología PIC y multisensores analógicos-

digitales tuvo como conclusión que la tarjeta de adquisición de datos cumplió con eficacia las tareas de procesamiento y ejecución de ensayos en las prácticas de laboratorio con la interfaz gráfica de usuario debido al empleo del protocolo de comunicación USB en lugar de los puertos serial-paralelo del computador. Esta investigación servirá como soporte al presente trabajo debido a las consideraciones de diseño presentadas a lo largo de todo el artículo, tanto el diseño circuital enfocado a la tarjeta de adquisición de datos.

Igualmente, Calderón (2017) de la facultad de Ciencias de la Universidad Central de Venezuela, en su trabajo para optar al título de Licenciada en Física, titulado **“Sistema de Adquisición de Datos Ambientales en los páramos venezolanos para el estudio del cambio climático”** obtuvo como resultado una tarjeta de adquisición de datos basada en dos sensores de temperatura analógicos, el LM35 y un termistor NTC de $10k\Omega$ capaz de realizar mediciones de temperatura entre los 0 y 70 °C, a pesar de que se estima que la tarjeta de adquisición de datos a diseñar sea capaz de medir un rango superior de temperatura, además de otras variables de proceso, la obra de Calderón ofrece una propuesta de diseño de gran potencial y de bajo costo, la cual puede ser empleada como referencia para el diseño final del presente trabajo, adicionalmente, la autora ofrece recomendaciones en cuanto a la elección del microprocesador y condiciones ambientales de la tarjeta de adquisición de datos, factores fundamentales al tener en cuenta durante la etapa de diseño de la tarjeta.

Así mismo, Sánchez (2017) de la facultad de Ingeniería de la Universidad Central de Venezuela en su trabajo para optar al título de Ingeniero Electricista, titulado como **“Diseño de un Sistema de Adquisición de Datos y Control de un banco de pruebas para Reguladores de Gas”** tuvo como objetivo general, diseñar e implementar un sistema de adquisición y control para un banco de pruebas de reguladores de gas según la norma ANSI B109.4-1998, en él se describe el proceso de diseño de un sistema de adquisición de datos para el banco de pruebas, introduciendo los diferentes tipos de reguladores de gas, los medidores y

transductores de presión a utilizar y el desarrollo de la interfaz gráfica de usuario usando LabView. Lo más destacable de la obra de Sánchez y de interés para la presente investigación se encuentra en el proceso de desarrollo de la interfaz gráfica de usuario, ya que, según indica, se centró principalmente en la realización de una interfaz gráfica intuitiva y de fácil manejo para el control del proceso y de la tarjeta de adquisición de datos, de tal forma que elaboró una serie de recursos gráficos, diagramas de flujo y esquemas que servirán como referencia para la elaboración de la interfaz de gráfica de usuario interactiva propuesta durante el planteamiento de los objetivos específicos de la investigación.

Finalmente, Barreto y Márquez (2016) de la facultad de Ingeniería de la Universidad de Carabobo, en su trabajo para optar por el título de Ingeniero Electricista, titulado **“Desarrollo de un Sistema de Adquisición de datos Portátil para el Banco de turbina Kaplan instalada en el Laboratorio de Hidráulica “Elías Sánchez Díaz”, en la Escuela De Ingeniería Civil de la facultad de Ingeniería de la Universidad De Carabobo”** tenían como objetivo general diseñar un sistema de adquisición de datos para el banco de la turbina Kaplan instalada en el Laboratorio de Hidráulica “Elías Sánchez Díaz” de la Escuela de Ingeniería Civil de la Facultad de Ingeniería de la Universidad de Carabobo, y concluyeron que la selección adecuada de sensores, el correcto acondicionamiento de las señales y el uso de un software especializado para el control de la tarjeta de adquisición de datos son piezas fundamentales para la implementación de un sistema de adquisición de datos.

El trabajo descrito es de relevancia para la investigación ya que el mismo podrá proveer sustento teórico al desarrollo de la tarjeta de adquisición de datos, debido a que en él se describen técnicas de acondicionamiento de señales, características de los microcontroladores e información detallada de diversos transductores y señales de campo, así como recomendaciones de diseño para la interfaz de usuario y posibles actualizaciones del equipo para mantener el equipamiento al día con las diversas tecnologías que surjan.

2.2 Bases Teóricas

Como apoyo bibliográfico a la presente investigación, se requiere la descripción de diversos conceptos que servirán como sustento teórico para el desarrollo efectivo de la obra y la consecución de los objetivos planteados, de esta manera, a continuación, se detallan los conceptos más relevantes para el efectivo desenvolvimiento de este trabajo de grado.

2.2.1 Sistemas de Adquisición de Datos

Osorio, Pérez y Rodríguez (2010) plantean que un Sistema de Adquisición de Datos (SAD) es un sistema que permite integrar diversos recursos para controlar y monitorear diferentes variables de un proceso, los autores describen el funcionamiento de un SAD en cuatro etapas, la primera de ellas captura las señales del proceso que contienen la información requerida para el análisis mediante el uso de transductores adecuados, la segunda etapa se encarga del acondicionamiento de las variables con el propósito de convertirlas a señales eléctricas compatibles con las especificaciones del SAD para posteriormente ser digitalizadas en la tercera etapa y ser enviadas a un computador a través de un protocolo. Finalmente, la cuarta etapa consiste en el procesamiento de las variables capturadas para su posterior manipulación a través de un ordenador empleando un programa especializado.

Calderón y Montero (2004) definen una medición como la determinación de una medida mediante un instrumento o sistema de medición, donde dicha medida usualmente es determinada indirectamente, mediante el efecto que produce en una variable secundaria conocida. En el ámbito industrial dichas mediciones son realizadas mediante elementos primarios o sensores, estos dispositivos se caracterizan por ser transductores que responden cuantitativamente a la medida de una magnitud física y proporcionan una salida útil para ser usada como variable de entrada a un sistema de procesamiento de información.

En cuanto a las variables de proceso, son aquellas magnitudes físicas involucradas en un proceso industrial determinado, las cuales son de interés medir y/o controlar para diversos fines, como puede ser el incremento de la eficiencia de un sistema dado.

Para la medición de cada una de estas variables es importante escoger el sensor adecuado, el cual, debe adaptarse al ambiente al que se le va a exponer, tiene que poseer un rango de medición ajustado al rango al que va a operar el sistema y finalmente debe tener una salida acorde al sistema de control del proceso. A continuación, se describen las variables de proceso y las características de sus transductores más importantes.

2.2.2 Presión

La presión se define como la razón que existe entre una fuerza aplicada sobre un área y su magnitud, puede ser expresada en bar, kg/cm², pascal, atmósferas, etc. Los transductores mayormente empleados para la medición de presión se describen en la figura 2.

Transductor de presión	Rango de medición (psi)	Rango de Temperatura (°C)	Nivel de Salida	Excitación	Exactitud (%)
Tubo de Bourdon	0,5 a 100.000	90 (máx)	-	-	0,5 - 1
Diafragma	0,071 a 28,466	90 (máx)	-	-	0,5 - 1
Fuelle	0,142 a 28,466	90 (máx)	-	-	0,5 - 1
Capacitivo	14,696 μ a 14,696	0 a 816	> 100 mV	AC	$\pm 0,05$ - $\pm 0,5$
LVDT	30 a 10.000	-55 a 150	> 100 mV	AC	$\pm 0,05$
Piezoeléctrico	0,1 a 10.000	-240 a 260	< 100 mV	AC (Autog	± 1

				enerada)	
Resistivo	5 a 10.000	-54 a 149	> 100 mV	AC y DC	± 1
Extensiométrico	0,5 a 10.000	-270 a 371	< 100 mV	AC y DC	$\pm 0,25$

Figura 2: Características de los Transductores de Presión

Fuente: Cárdenas (2009)

2.2.3 Temperatura

Cárdenas (2009) describe la temperatura como el grado de agitación térmica de las moléculas de un cuerpo, es evidente que esta definición involucra la energía interna de un determinado cuerpo, dicha energía interna está definida por la ley cero de la termodinámica, la cual establece que “cuando dos cuerpos se encuentran en equilibrio térmico con un tercero, estos están a su vez en equilibrio térmico entre sí”. Este principio es fundamental para la medición de la temperatura, ya que se emplean patrones conocidos de temperatura para graduar los instrumentos de medición o termómetros y posteriormente determinar la temperatura de un determinado proceso. Los transductores de temperatura más utilizados se muestran en la figura 3:

Transductor de Temperatura	Rango de Medición (°C)	Exactitud (% fe)	Aplicaciones
RTD de Platino	-200 a 950	0,01	Temperaturas de fluidos y superficie
RTD de Níquel	-150 a 300	0,5	Temperaturas de fluidos y superficie
RTD de	-200 a	0,1	Temperaturas de fluidos y superficie

Cobre	120		
Termopar Tipo T	-200 a 260	0,4 – 2	Biofísica, Máquinarias, Combustión, Laboratorios, etc
Termopar Tipo J	550 a 750	0,3 – 0,5	Biofísica, Máquinarias, Combustión, Laboratorios, etc
Termopar Tipo K	Hasta 1100	0,8	Biofísica, Máquinarias, Combustión, Laboratorios, etc
Termopar Tipo R y S	Hasta 1500	0,5 – 0,8	Biofísica, Máquinarias, Combustión, Laboratorios, etc
Pirómetro	-50 a 4600	$\pm 0,5$	Muy altas temperaturas
Termómetro Bimetálico	-73 a 538	$\pm 0,5$ - ± 1	Control, Interruptores por sobrecarga
Termistores	-100 a 450	$\pm 0,005$ - $\pm 0,1$	Compensación de temperatura Industria en general
Termómetros de CI	-55 a 150	$\pm 0,5$	Compensación de unión de referencia en Termopares

Figura 3: Características de los Transductores de Temperatura

Fuente: Cárdenas (2009)

2.2.4 Nivel

Dulhoste (2014) define en su obra al nivel como la altura a la que se encuentra la superficie libre del fluido a partir de una línea de referencia, siendo la medición de nivel la determinación de dicha altura, su unidad normalizada en el sistema internacional es el metro (m), sin embargo, industrialmente también son empleadas los pies (ft), las pulgadas (in) y los centímetros (cm). Los instrumentos de medición de nivel más utilizados se muestran en la figura 4:

Instrumento	Campo de	Precisión de %	Presión máxima	Temperatura máxima del	Aplicaciones
-------------	----------	----------------	----------------	------------------------	--------------

	Medida (m)	escala	(bar)	fluido (°C)	
Conductivo	Ilimitado	-	80	200	Líquidos
Capacitivo	0.6	± 1	80 - 250	200 - 400	Tanques presurizados o no
Ultrasónico	0.3	± 1	400	200	Tanques y líquidos
Radiación	0 – 2.5	$\pm 0.5 - \pm 2$	-	150	Sin contacto
Laser	0 – 2	$\pm 0.5 - \pm 2$	-	1500	Sin contacto

Figura 4: Características de los Transductores de Nivel

Fuente: A. Creus (2010)

2.2.5 Caudal

El caudal o flujo de fluidos puede ser representado de tres formas distintas, la primera es el caudal o flujo volumétrico, el cual es el volumen de un fluido que pasa por un punto en un tiempo determinado, la segunda es el flujo másico que es representado en unidades de masa por unidad de tiempo y por último se encuentra la velocidad de flujo que representa la velocidad de un material y se expresa en metros por segundo. Los transductores de flujo o caudal más importantes se muestran en la figura 5:

Transductor de Flujo	Costo Relativo	Temperatura Máxima del Fluido (°C)	Presión Máxima del Fluido (psi)	Exactitud
Placa Orificio	Bajo	Ninguna	Limitada por el transductor	$\pm 2 - \pm 4$ (%FE)
Área Variable	Bajo	350	300	$\pm 1 - \pm 10$ (%FE)
Desplazamiento	Medio	-25 – 82	1500	$\pm 0,5$ (% de

Positivo				la Velocidad)
Turbina	Alto	-240 – 538	7500	$\pm 0,25$ (% de la Velocidad)
Vortex	Alto	-40 – 400	-	± 1 (% de la Velocidad)
Electromagnético	Alto	-	-	$\pm 0,5$ (% de la Velocidad)

Figura 5: Características de los Transductores de Flujo

Fuente: Cárdenas (2009)

En cuanto a la segunda y tercera etapa que describen Osorio, Pérez y Rodríguez (2010), pueden ser agrupadas en una misma etapa de procesamiento de la señal, conformada por el acondicionamiento de la señal y la digitalización de la misma para luego ser enviada al computador mediante un protocolo de comunicación. El procesamiento de la señal consiste en adaptar una determinada señal para que se adecúe a los requerimientos necesarios de un sistema en específico, ya sea de amplitud, nivel de ruido o de frecuencia. Gálvez (2002) plantea que el acondicionamiento de una señal puede ser dividido en una fase de conversión de la señal, otra de modificación del nivel de la misma, otra de linealización de la respuesta y la última de filtrado, pudiéndose omitir cualquiera de estas de ser necesario.

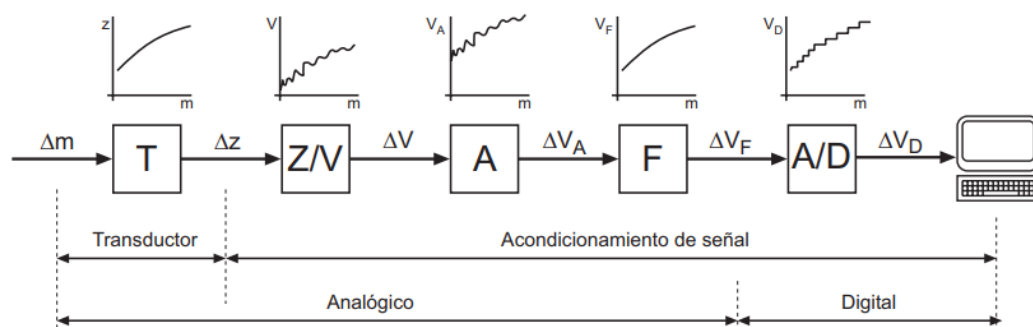


Figura 6: Esquema de un sistema de medición

Fuente: Gálvez (2002)

La conversión de la señal se produce cuando se requiere que las variaciones de la variable de proceso sean expresadas en función de otra variable, debido a que la mayoría de los transductores aprovechan variaciones de capacitancia, inductancia o resistencia, es usual que se requiera convertir estas variables a variaciones de voltaje o de corriente, el método de conversión de resistencia a tensión más común, es el circuito potenciométrico, el cual emplea un divisor resistivo simple que proporciona un voltaje de salida proporcional a las variaciones de resistencia.

La etapa de linealización de la señal se produce cuando la señal no tiene una relación lineal entre la magnitud que se está midiendo y la variable secundaria asociada, como es el caso de la relación entre la temperatura y el voltaje de salida de un termopar, para estas situaciones se requiere un circuito que linealice la relación entre ambas variables, para ello se suelen utilizar arreglos de operacionales o la aplicación de regresiones matemáticas luego de digitalizar la señal. Finalmente, se tiene el proceso de filtrado, donde son excluidas frecuencias no deseadas de la señal a transmitir mediante la utilización de filtros activos y pasivos. Durante la etapa de procesamiento de la señal, también se produce la digitalización de la misma, el cual es el proceso donde se convierte un voltaje analógico en una señal digital que pueda ser leída por un microcontrolador, este procedimiento es realizado por los convertidores análogos – digitales o ADC, dichos convertidores se clasifican en convertidores por aproximaciones sucesivas, convertidores de rampa, convertidores de doble pendiente y convertidores de tipo flash. Muchos microcontroladores poseen ADC integrados internamente, sin embargo, de no tener un ADC propio, es posible emplear un ADC externo para la conversión análogo digital.

En la mayoría de los procesos industriales, los transductores empleados poseen un transmisor compatible que realiza la etapa de acondicionamiento y envía la información mediante una señal estandarizada al PLC o a la tarjeta de adquisición de datos quien se encarga de enviar y recibir la información obtenida del proceso hacia el ordenador para así realizar las correspondientes acciones de monitoreo y control. Dentro de los estándares industriales que se utilizan para la transmisión de

información en los procesos, A. Creus (2010), destaca 6 tipos de señales, entre las que se encuentran las señales neumáticas, hidráulicas, electrónicas, digitales, ópticas y por radio, ya que la tarjeta de adquisición de datos a diseñar debe ser compatible con multitud de sensores que operan tanto con señales electrónicas como digitales, serán ellas las descritas a continuación. En cuanto a las señales electrónicas o analógicas se emplean lazos de corriente o fuentes de tensión proporcionales a la variable de proceso a medir, existen procesos que emplean señales de 0 a 10 V, 1 a 5 V, -5 a 5 V y -10 a 10V, sin embargo, las señalizaciones que emplean tensión suelen ser susceptibles a perturbaciones electromagnéticas por lo que el estándar habitual es el lazo de corriente de 4 a 20 mA, donde el transmisor es alimentado con una fuente de 24 V y el receptor recibe una conexión de 2 hilos a una resistencia de 250Ω , de tal forma que el sistema es inmune a perturbaciones electromagnéticas y posee valores determinados en caso de desconexión o corto circuito. Otro tipo de señales analógicas que igualmente emplean el lazo de corriente, son las señales de 0 a 20 mA y -20 a 20 mA, aunque las mismas no son tan comunes, se presentan en algunos sensores ocasionalmente, sin embargo, tienen el inconveniente que no poseen valores determinados para desconexión o corto circuito.

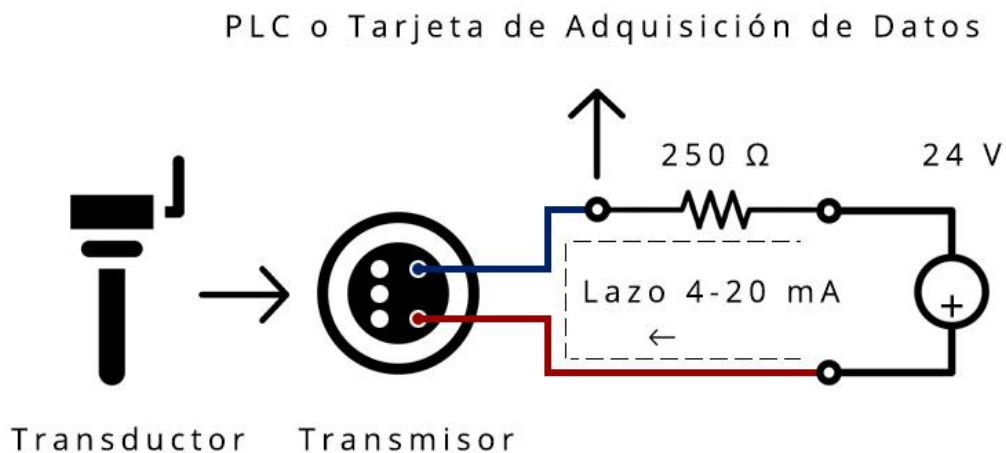


Figura 7: Diagrama convencional del Lazo de corriente de 4 – 20 mA

Fuente: Freytes (2022)

Tanto si se van a usar señales de tensión como señales de corriente, el proceso de acondicionamiento va a resultar en un valor de tensión que será digitalizado por la tarjeta de adquisición de datos o por un PLC mediante el ADC incorporado. Existen múltiples tipos de ADC, los más comunes son los de aproximaciones sucesivas y los convertidores Flash por su relación entre velocidad de conversión y su costo, sin embargo, independientemente del tipo de ADC integrado en el equipo que hará las acciones de control, se deben tener en cuenta las limitaciones del ADC al momento de ajustar los parámetros de acondicionamiento de la señal, dos de estos factores que se deben considerar son la resolución y el rango de medición. La resolución del ADC consiste en el número de bits resultantes de la conversión de la señal analógica, a mayor cantidad de bits, es mayor la precisión de las mediciones realizadas. Por otra parte, el rango de medición se define como los límites admisibles de una señal para ser medidos por el ADC, también suele ser denominado como Fondo de Escala o FSR, González (2020) en su obra define el mismo como los extremos de la señal analógica a la entrada, de tal forma que las señales analógicas a digitalizar no pueden superar dichos extremos.

Los microcontroladores empleados en las tarjetas de adquisición de datos suelen poseer ADCs incorporados, con resoluciones de 8 a 10 bits y un FSR cercano a los 5V, por lo tanto, los acondicionamientos a diseñar deben brindar una señal de salida dentro del rango del ADC y no puede poseer variaciones superiores a la resolución del mismo. Este límite establecido por el rango máximo del ADC implica que el acondicionamiento aplicado a las señales de corriente tanto positivas como negativas debe resultar en una señal de tensión positiva dentro del rango, al igual que las señales de tensión estandarizadas que se extienden dentro de valores tanto positivos como negativos deben ser acondicionadas para no superar el rango del ADC. Si la señal tiene un valor que supera el rango máximo establecido por el ADC, es posible escalarla mediante el uso de circuitos atenuadores como el divisor resistivo, luego, agregando un nivel de tensión de continua a la señal es posible

eliminar los valores negativos que tenga la misma para que se adecúe al rango de tensión admitido por el ADC.

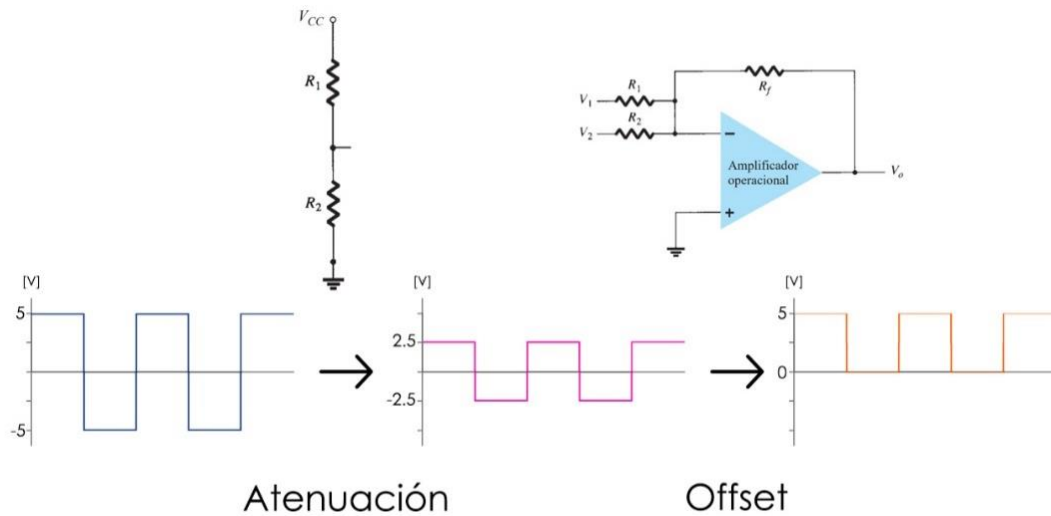


Figura 8: Diagrama de acondicionamiento de una señal al rango del ADC

Fuente: Freytes (2022)

Es importante destacar que para cualquiera que sea la señal a acondicionar o digitalizar, se debe considerar el teorema de muestreo de Shannon-Nyquist, el cual establece que para poder reconstruir una señal de forma correcta se deben tomar muestras a una frecuencia al menos al doble de la frecuencia de la señal que se desea reconstruir, de tal forma que el sistema de muestreo a utilizar para digitalizar las señales debe cumplir con este parámetro y tener una alta tasa de muestreo.

Ya que la tarjeta de adquisición de datos a diseñar debe contar con ciertas características de entradas y salidas, así como un ADC con suficiente resolución, se debe escoger un microcontrolador con algunos parámetros específicos, para ello es relevante mencionar a los microcontroladores y sus propiedades. Peña (2008) relata que un microcontrolador es un circuito integrado que en su interior está conformado por una Unidad Central de Procesamiento (CPU), unidades de memoria (RAM y ROM), entradas, salidas y periféricos. Todas estas partes se encuentran interconectadas entre sí y son controladas mediante un programa diseñado para realizar una tarea específica. Dicho programa es elaborado con un lenguaje de bajo

nivel como ensamblador, Basic o C y es cargado al microcontrolador a través de un programador externo que se conecta al computador, haciendo de este proceso una tarea poco amigable para el usuario. Sin embargo, desde hace unos años surgieron algunas placas de prototipado electrónico que poseen tanto un lenguaje de programación más amigable para el usuario como un sistema de carga de programas más eficiente, ya que hace uso de un cargador de arranque o bootloader que comunica al microcontrolador con el entorno de desarrollo mediante protocolos convencionales, pudiendo cargar el programa a la placa con un cable USB sin necesidad de programadores adicionales.

La placa de prototipado de mayor uso en la actualidad es el Arduino, que además de poseer un bootloader para la carga de programas al microcontrolador mediante USB, cuenta con un entorno de desarrollo independiente al del microcontrolador incorporado, de tal forma que su programación es muy sencilla e incluye múltiples librerías para un desarrollo más eficiente, la más relevante para el diseño de esta propuesta es la librería del protocolo Serial, que con unas pocas instrucciones permite el intercambio de información entre el microcontrolador y el computador, el cual corresponde a la etapa final realizada por la tarjeta de adquisición de datos luego del acondicionamiento de la señal. Existen múltiples protocolos de comunicación que pueden ser utilizados para la comunicación entre la tarjeta de adquisición de datos y el computador, sin embargo, el protocolo de comunicación de mayor auge en la actualidad es el protocolo USB, el cual define un conjunto de paquetes y procedimientos para controlar y administrar el acceso al bus del mismo nombre por parte de los periféricos y permitir el entendimiento entre las dos partes en la comunicación. El protocolo de comunicación transmite la información al computador para la última etapa, la cual mediante un software especializado se encarga de manipular las variables capturadas por la tarjeta de adquisición de datos, ya sea para la visualización de las variables en pantalla mediante gráficos o el establecimiento de un punto de ajuste para controlar las variables de proceso, que a través de la aplicación de algoritmos de control enviarán de vuelta a la tarjeta de

adquisición de datos una serie de parámetros para regular las magnitudes involucradas en un determinado proceso.

El programa o software especializado de la etapa final es denominado como Interfaz Hombre Máquina o HMI por sus siglas en inglés, la cual es básicamente una interfaz gráfica de usuario destinada al control de un determinado proceso, la HMI es desarrollada mediante un lenguaje de programación de alto nivel como Python, C#, C++ o Java, siendo este último una de las opciones más empleadas por parte de los desarrolladores debido a su versatilidad y alta compatibilidad, ya que funciona mediante una máquina virtual denominada como JVM o Máquina Virtual de Java, la cual se encarga de convertir el código Java compilado o Bytecode en código nativo, de tal forma que el programa es desarrollado una única vez y puede ser utilizado en cualquier plataforma capaz de ejecutar la Máquina Virtual de Java.

2.3 Bases legales

Son las regulaciones que imponen metas y acciones correspondientes al tipo de organización que elabora la planificación, fijando sanciones cuando estas no son cumplidas. Las bases legales dan sustento a la investigación por medio de leyes, reglamentos y decretos. A continuación, se citarán los fundamentos legales de la investigación:

- Constitución de la República Bolivariana de Venezuela (1999)
 - Artículo 98. La creación cultural es libre. Esta libertad comprende el derecho a la inversión, producción y divulgación de la obra creativa, científica, tecnológica y humanística, incluyendo la protección legal de los derechos del autor o de la autora sobre sus obras. El Estado reconocerá y protegerá la propiedad intelectual sobre las obras científicas, literarias y artísticas, invenciones, innovaciones, denominaciones, patentes, marcas y lemas de acuerdo con las condiciones y excepciones que establezcan la ley y los tratados internacionales suscritos y ratificados por la República en esta materia.

- Ley Orgánica de Ciencia, Tecnología e Innovación (2010)
 - Artículo 2. Las actividades científicas, tecnológicas y de innovación son de interés público y de interés general.
 - Artículo 9. El Ministerio de Ciencia y Tecnología apoyará a los organismos competentes por la materia, en la definición de políticas tendentes a proteger y garantizar la propiedad intelectual colectiva de los conocimientos, tecnologías e innovaciones de los pueblos indígenas y los conocimientos tradicionales.
 - Gaceta oficial N.º 38.095 del decreto N.º 3.390 Artículo N.º 1 (2004)

2.4 Definición de términos básicos

A continuación, se definen los conceptos que dan inicio y hacen referencia a esta investigación.

Lenguaje de Programación, es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes y algoritmos a modo de controlar el comportamiento físico y lógico de una computadora con el objetivo de que produzca diversas clases de datos.

Microcontrolador, es un circuito integrado conformado por diversos componentes interconectados en su interior, que según el programa que se encuentre cargado en el mismo puede ser empleado para realizar múltiples tareas.

Señal, se le define de esta manera a toda magnitud física cuyo valor varía en el tiempo. Esta puede ser analógica o digital.

Software, es el conjunto de los programas de cómputo que forman parte de las operaciones de un sistema informático.

Protocolo de Comunicación, se le denomina al conjunto de pautas que posibilitan que distintos elementos que forman parte de un sistema establezcan comunicaciones entre sí, intercambiando información.

Transductor, es un dispositivo capaz de convertir o transformar una determinada manifestación de energía de entrada en otra diferente de salida

CAPÍTULO III

MARCO METODOLÓGICO

Hernández, Fernández y Baptista (2006) relatan que “La investigación científica se concibe como un conjunto de procesos sistemáticos y empíricos que se aplican al estudio de un fenómeno; es dinámica, cambiante y evolutiva. Se puede manifestar de tres formas: cuantitativa, cualitativa y mixta”. El estudio a realizar tiene un enfoque cuantitativo, debido a que se planea recolectar datos para responder las preguntas de investigación y probar las hipótesis establecidas previamente. Los autores citados definen que “el enfoque cuantitativo usa la recolección de datos, para probar hipótesis, con base en la medición numérica y análisis estadísticos, para establecer patrones de comportamiento y probar teorías” (p.05).

3.1 Tipo de Investigación

La investigación a realizar se relaciona con la modalidad de investigación de tipo proyecto factible ya que se busca resolver los objetivos propuestos a través del planteamiento de una propuesta viable. Según el Manual de Trabajos de Grado de Especialización y Maestría de la Universidad Pedagógica Experimental Libertador (UPEL, 2011), define esta modalidad investigativa como:

“...la elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades (p.21).”

3.2 Diseño de la investigación

Arias (2006) establece que “una investigación de campo consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna”

(p.31). Por lo tanto, el presente estudio es una investigación de campo ya que la información se obtendrá por medio de fuentes vivas y directas en su contexto natural.

De igual manera, el presente estudio es una investigación documental, ya que la información adicional que respalda la presente investigación será extraída de autores pasados. De esta forma el autor ya mencionado, define la investigación documental como “un proceso basado, en la búsqueda, recolección, análisis, crítica investigación de los datos secundarios, es decir los datos obtenidos y registrados por otros investigadores en fuentes documentales” (p.27).

3.3 Nivel de la Investigación

El nivel de investigación considerado para efecto de este estudio es de tipo descriptivo ya que se desea describir, en todos sus componentes principales, una realidad; Arias (2006) plantea que “la investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento” (p.26).

3.4 Población y muestra

3.4.1 Población

Es necesario establecer una población a los cuales se les aplicará los instrumentos de recolección de datos, los cuales son un grupo de individuos con características en común; dicha población es definida por Fideas Arias (2006) como “...un conjunto finito o infinito con características comunes para las cuales serán extensivas las conclusiones de la investigación. Esta queda delimitada por el problema y por los objetivos del estudio” (p. 81). La población de la presente investigación estará conformada por los sistemas de adquisición de datos empleados en plantas de procesos de múltiples lazos, cuyas funciones abarquen el control supervisorio y diagnóstico de dichos procesos.

3.4.2 Muestra

La muestra es la que permite evaluar la problemática, ya que esta genera datos por medio de los cuales se puede hacer inferencias o generalizar resultados de las fallas detectadas. Arias (2006) define la muestra como “un subconjunto

representativo y finito que se extrae de la población accesible” (p.81). Para fines de esta investigación, la muestra será representada por los dispositivos electrónicos para la adquisición de datos alternativos a los equipos comerciales que posean las características de costo reducido, universalidad y facilidad operativa.

3.5 Técnica de recolección de datos

Con respecto a las técnicas e instrumentos de recolección de datos, Arias (2006) expresa que:” se entenderá por técnica, el procedimiento o forma particular de obtener datos o información, y los instrumentos como medios materiales que se emplean para recoger y almacenar información” (p.67). Para el desarrollo de la investigación se emplearán las siguientes técnicas e instrumentos de recolección de datos: observación directa y encuesta escrita.

3.5.1 Observación directa

Según Tamayo y Tamayo (2004) la observación, “es aquella en el cual el investigador puede observar y recoger los datos mediante su propia observación”. (p.122). En base a este concepto, la técnica a aplicar para este proyecto será la observación de carácter directo por estar en contacto el fenómeno a investigar con el instrumento en sí, fue de observación simple o no participativa la cual Reyes Triana (2012) define cómo: “Una observación con propósitos definidos. El investigador se vale de ella para obtener información y datos sin participar en los acontecimientos de la vida del grupo que estudia, permaneciendo ajeno al mismo”.

3.5.2 Observación documental

Por otra parte, se procederá a realizar una observación documental, la cual Bavaresco, A (2006) reseña como “Tanto los libros, folletos, documentos, revistas, periódicos... vienen a brindar al lector-investigador, todo el soporte del marco teórico, lo que significa que se percata de todo lo escrito o que esté relacionado con el tema que escogió como investigación” (p.99). En el presente trabajo de investigación dicha técnica tiene como objetivo recopilar información mediante análisis de fuentes secundarias del tópico estudiado, como soporte teórico para posteriormente realizar el diseño de la propuesta.

3.6 Instrumento de recolección de datos

Según Tamayo y Tamayo (2004) definen que: “Un instrumento de recolección de datos es cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registrar o almacenar información” (p. 69). Los instrumentos son fundamentales para llevar a cabo las técnicas que los investigadores han de seleccionar para su investigación; en concreto, van a ser éstos el físico que contuvo toda la información recabada. Es por esto que para la presente investigación se escogerá como instrumento para realizar la recolección de datos, la lista de chequeo y las fichas de trabajo.

3.6.1 Lista de Chequeo

Oliva, P (2009) describe este instrumento como “...implica una lista de factores, propiedades, aspectos, componentes, criterios, dimensiones, la presencia o cantidad de determinados procedimientos, con el fin de realizar una tarea determinada” (p. 8). Por consiguiente, se usó la lista de chequeo por su facilidad en la organización de tareas y para llevar un control detallado de la propuesta antes mencionada, lo cual permite la recolección de información relevante con respecto a los objetivos antes planteados.

3.6.2 Ficha de Trabajo

Según Tamayo y Tamayo (2004) establecen que: “La ficha de trabajo es el instrumento que nos permite ordenar y clasificar los datos consultados, incluyendo nuestras observaciones y críticas, facilitando así la redacción del escrito. (p. 182)”. De tal forma que este instrumento será empleado para recopilar información con respecto al fenómeno estudiado, realizando anotaciones que permitirán posteriormente sintetizar de forma detallada el escrito con la documentación recabada.

3.7 Fase metodológica

Tomando en cuenta diferentes concepciones, un proyecto factible contempla las siguientes etapas: el diagnóstico de las necesidades, que puede basarse en una investigación de campo o documental, planteamiento y fundamentación teórica de la

propuesta, el procedimiento metodológico, las actividades y recursos necesarios para su ejecución, el análisis de viabilidad o factibilidad del proyecto (económica, política, social, entre otros) y la posibilidad de ejecución.

Fase I. Diagnóstico.

Durante esta fase se determinarán las necesidades que debe cubrir la tarjeta de adquisición de datos a diseñar, el número de entradas y salidas mínimas presentes en procesos de múltiples lazos, los tipos de señales más comunes brindadas por los sensores en los diferentes procesos y los componentes electrónicos a emplear para el acondicionamiento de dichas señales. Para la determinación de estas necesidades se recurrirá a bibliografía especializada donde se detallen modelos de procesos con múltiples lazos, así como consultas a profesionales del área de la automatización industrial.

Fase II. Diseño de la Tarjeta de Adquisición de Datos.

En esta fase se procederá a realizar el diseño de la tarjeta de adquisición de datos con los parámetros requeridos establecidos durante el diagnóstico, rangos de operación y magnitudes que procesará la tarjeta de adquisición de datos, así como también se elaborará el diseño de la estructura que almacenará el hardware y el programa de control del microcontrolador. El diseño de la tarjeta de adquisición de datos será realizado en el software de simulación de circuitos electrónicos Proteus, en él se realizarán las pruebas correspondientes y la generación de los esquemáticos, en cuanto a la estructura para almacenar la tarjeta de adquisición de datos será diseñada en un software de modelado 3D para finalizar exportando las imágenes renderizadas de la misma.

Fase III. Desarrollo de la Interfaz Gráfica.

Durante esta fase se realizará el diseño y desarrollo de la interfaz gráfica o HMI que se comunicará con la tarjeta de adquisición de datos para la graficación de las señales digitalizadas y el control de los procesos, dicha interfaz gráfica será desarrollada mediante el lenguaje de programación Java, específicamente mediante la librería Swing y se emplearán los lineamientos del “UX Design” como metodología

para la elaboración de interfaces de usuario basadas en la usabilidad y el principio de utilidad. Una vez finalizado el desarrollo de la interfaz gráfica, en esta fase también se realizará una simulación de la tarjeta de adquisición de datos en funcionamiento mediante el software de simulación de circuitos electrónicos Proteus, comunicándose con la HMI mediante un puerto serial virtual que proporciona el mismo simulador.

Fase IV. Factibilidad.

Finalmente, durante esta fase se determinará la validez de la solución propuesta para así asegurar la implementación de la misma y garantizar el mejor aprovechamiento de los recursos siguiendo el siguiente orden lógico, cálculo de los costos relacionados con el modelo propuesto en función de los materiales, recursos humanos y tecnológicos.

CAPÍTULO IV

RESULTADOS

A lo largo de este capítulo se describen el desarrollo de las fases planteadas en el capítulo anterior para la consecución de los objetivos propuestos, para ello, se empleó la información recabada a través de la aplicación de las diversas técnicas de recolección de datos, entre ellas la observación directa y la observación documental, para posteriormente ser plasmada mediante el uso de los instrumentos de recolección de datos como las listas de chequeo y las fichas de trabajo; con la finalidad última de realizar el diseño electrónico y la simulación de una tarjeta de adquisición de datos de costo reducido que permita el diagnóstico y control supervisorio de procesos de múltiples lazos empleando una interfaz gráfica realizada con el lenguaje de programación Java. De esta forma, a continuación, se describen los resultados obtenidos en cada una de las fases.

4.1 Fase I. Diagnóstico

Durante esta fase se realizó la determinación de los parámetros mínimos necesarios que debía cumplir la tarjeta de adquisición de datos a diseñar, para ello se recurrió a bibliografía especializada en el control de procesos y la instrumentación industrial, a partir de ellos, se extrajeron múltiples diagramas de procesos estandarizados con la norma ISA-S5.1-84, la cual establece la nomenclatura que se debe seguir a la hora de realizar diagramas de procesos, la misma define un conjunto de reglas y símbolos para la identificación tanto de los instrumentos presentes en un proceso, como las señales involucradas y los lazos de control a los que pertenecen cada uno de ellos. Para el análisis de los diagramas extraídos se empleó la obra **Instrumentación Industrial** de A. Creus (2010), en ella se detalla la norma ISA-S5.1-84 y sus especificaciones, donde se definen la identificación de las funciones realizadas por cada uno de los instrumentos empleados en los procesos, a

continuación, se muestra la figura resumen con el significado de los identificadores literales de la norma.

PRIMERA LETRA (4)			LETRAS SUCESIVAS (3)		
	Variable medida o inicial	Letra de modificación	Lectura o función de lectura pasiva	Función de salida	Letra de modificación
A	Análisis(5, 19)		Alarma		
B	Quemador, combustión		Libre(1)	Libre(1)	Libre(1)
C	Libre (1)			Control (13)	
D	Libre (1)	Diferencial (4)			
E	Tensión (f.e.m.)		Sensor (Elemento primario)		
F	Caudal	Relación (4)			
G	Libre (1)		Vidrio, Dispositivo visión (9)		
H	Manual				Alto (7,15,16)
I	Corriente (eléctrica)		Indicar (10)		
J	Potencia	Exploración (7)			
K	Tiempo, programación tiempo	Variación de tiempo (4,21)		Estación de control (22)	
L	Nivel		Luz (11)		Bajo (7,15,16)
M	Libre (1)	Momentáneo (4)			Medio, Intermedio (7,15)
N	Libre (1)		Libre (1)	Libre (1)	Libre (1)
O	Libre (1)		Orificio, Restricción		
P	Presión, Vacío		Punto (Ensayo) Conexión		
Q	Cantidad	Integrar, Totalizar (4)			
R	Radiación		Registro (17)		
S	Velocidad, Frecuencia	Seguridad (8)		Interruptor (13)	
T	Temperatura			Transmisión (18)	
U	Multivariable (6)		Multifunción (12)	Multifunción (12)	Multifunción (12)
V	Vibración, Análisis mecánico (19)			Válvula, Regulador tiro, Persiana (13)	
W	Peso, fuerza		Vaina, Sonda		
X	Sin clasificar (2)	Eje X	Sin clasificar (2)	Sin clasificar (2)	Sin clasificar (2)
Y	Evento, Estado o Presencia (20)	Eje Y		Relé, Cálculo, Conversión (13,14,18)	
Z	Posición, Dimensión	Eje Z		Motor, Actuador, Elemento final de control sin clasificar	

Figura 9: Código de identificación de instrumentos. ISA-S5.1-84 (R-1992)

Fuente: A. Creus (2010)

Otra de las características que define la norma ISA es la nomenclatura para la identificación de los actuadores y los tipos de señales presentes en los diagramas de procesos, ambas nomenclaturas se presentan a continuación.

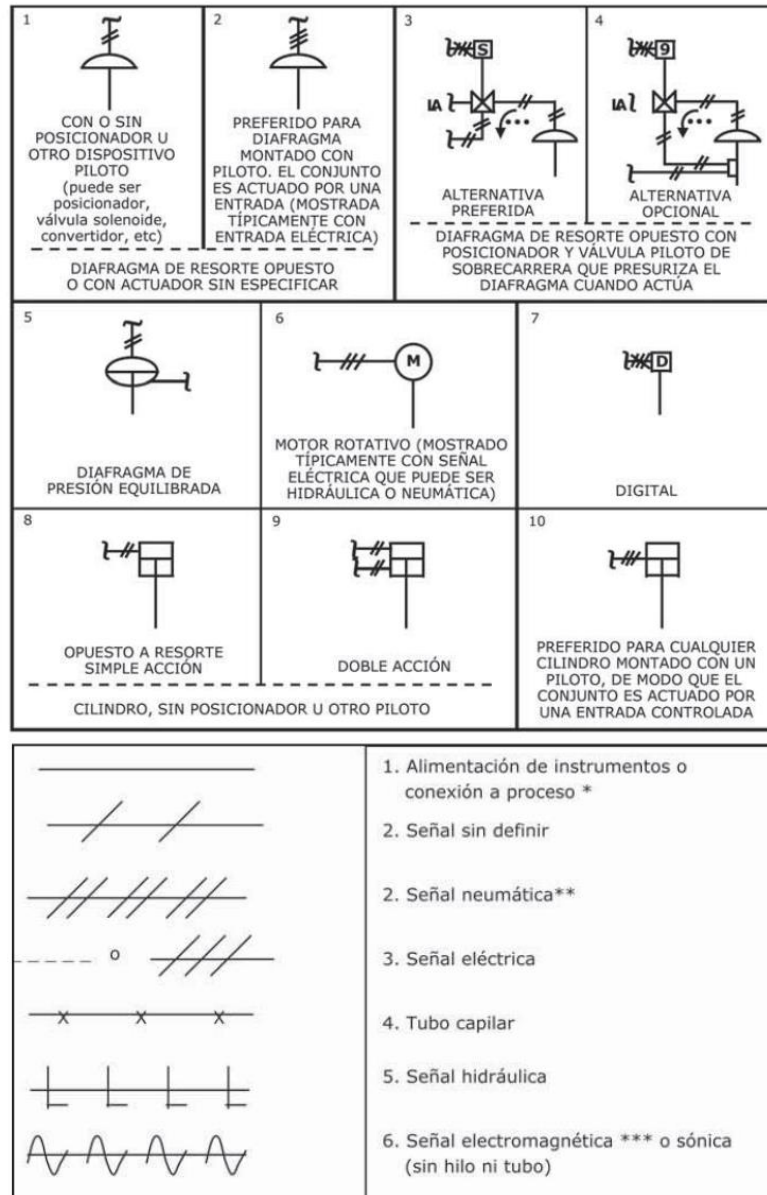


Figura 10: Símbolos de actuadores y tipos de señales. ISA-S5.1-84 (R-1992)

Fuente: A. Creus (2010)

Habiendo analizado los diagramas de procesos mediante la nomenclatura descrita por la norma ISA-S5.1-84, se procedió a establecer una línea general con respecto al número de entradas y salidas presentes en la mayoría de estos procesos industriales, con el objetivo de determinar un número de entradas y salidas acorde a los requerimientos mínimos de la industria, las siguientes figuras ejemplifican algunos de los diagramas analizados.

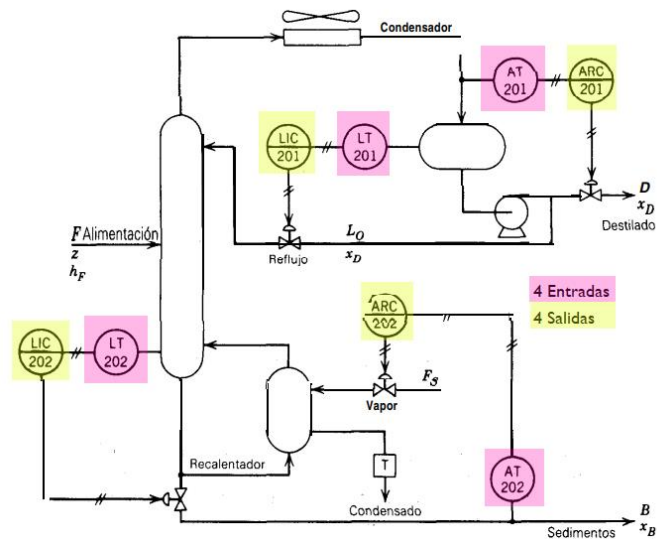


Figura 11: Diagrama de Control de Columna de Destilación

Fuente: C. Smith (1991)

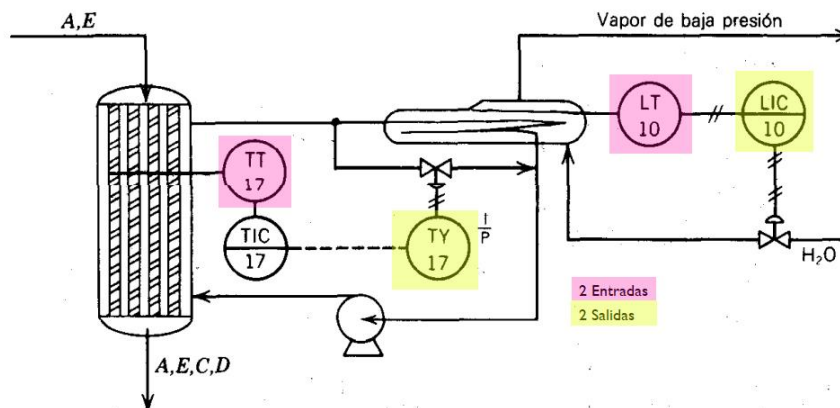


Figura 12: Diagrama de Control de Reactor Químico

Fuente: C. Smith (1991)

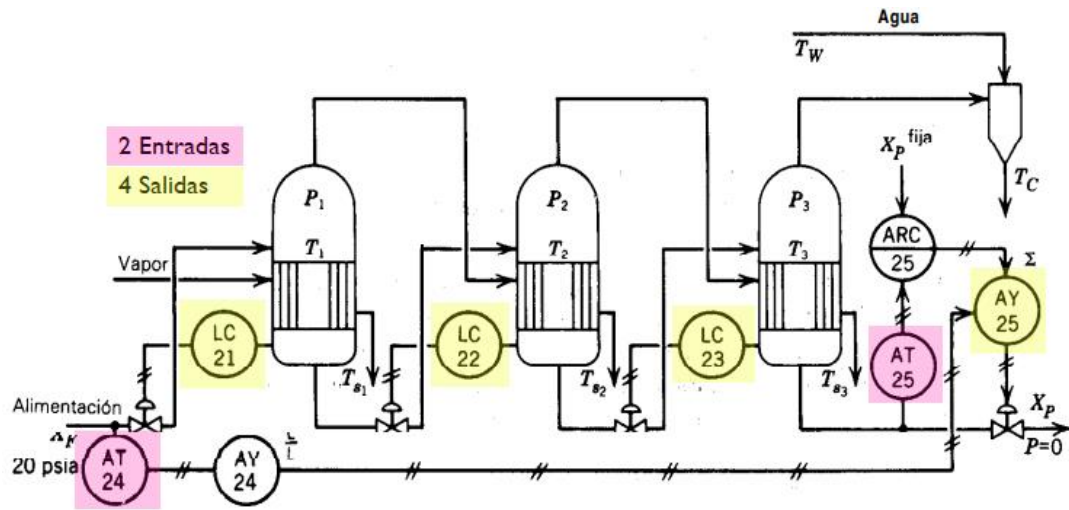


Figura 13: Diagrama de Control de Evaporador Triple

Fuente: C. Smith (1991)

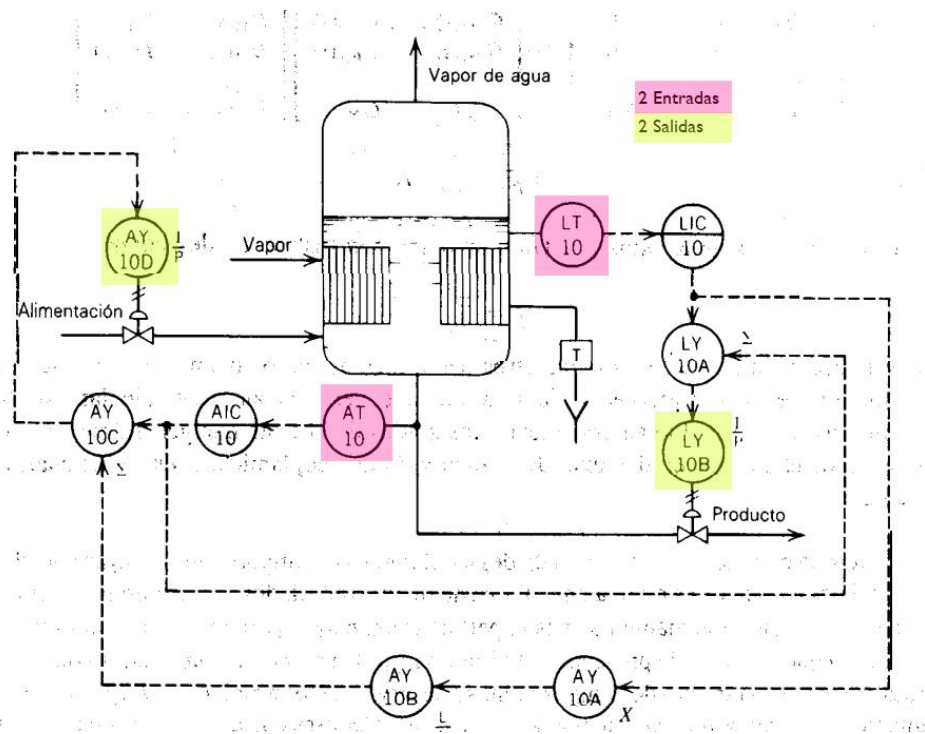


Figura 14: Diagrama de Control de Desacople para Evaporador

Fuente: C. Smith (1991)

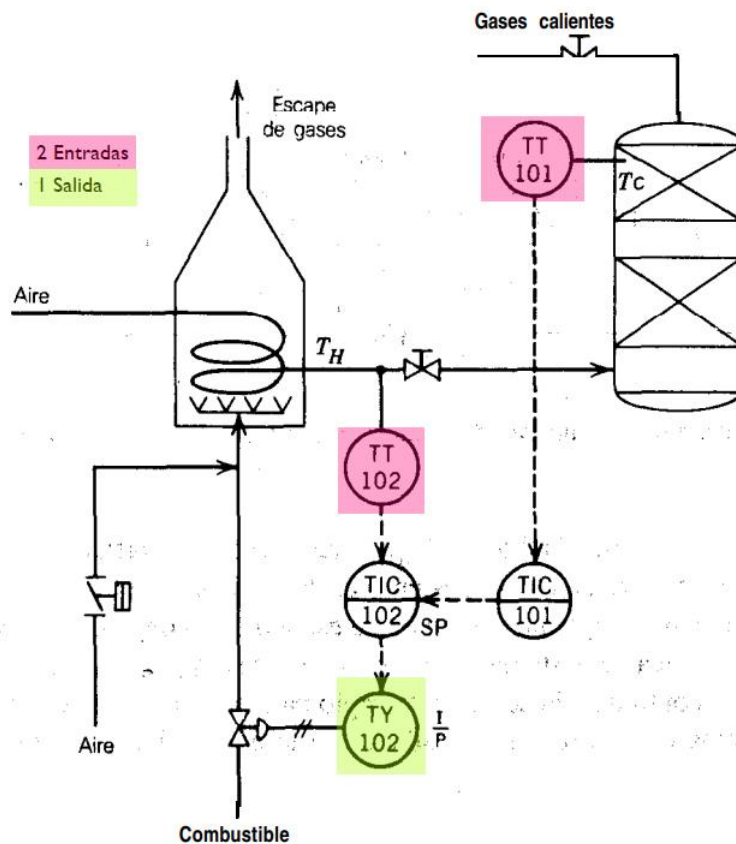


Figura 16: Diagrama de Control de Sistema de Regeneración

Fuente: C. Smith (1991)

Habiendo recopilado un número de veinte diagramas de procesos distintos con la norma ISA-S5.1-84, se procedió a realizar un análisis mediante una lista de chequeo con el objetivo de determinar el número de entradas y salidas que debía poseer la tarjeta de adquisición de datos, los resultados obtenidos luego de analizar la lista de chequeo fueron los siguientes.

Cuadro 1: Lista de chequeo empleada para el análisis de los diagramas

Nro	Descripción	Respuestas afirmativas	Respuestas negativas	Observación
1	¿El proceso tiene más de 4	18	2	Gran parte de los procesos analizados tenían al menos 4

	entradas?			entradas y 4 salidas, de tal forma que un número de canales menor a este, estaría muy limitado
2	¿El proceso tiene más de 4 salidas?	10	10	
3	¿El proceso tiene más de 8 entradas?	4	16	
4	¿El proceso tiene más de 8 salidas?	2	18	Una cantidad muy pequeña de los procesos estudiados tenían más de 8 salidas a controlar
5	¿El proceso tiene múltiples lazos?	17	3	Se hizo énfasis en esta clase de diagramas ya que los mismos son los procesos objetivo de la presente investigación

Fuente: Freytes (2022)

Ya que todos los diagramas analizados son diagramas de procesos presentes en la industria, los mismos fueron utilizados como referencia para la elección del número de entradas y salidas del diseño de la tarjeta de adquisición de datos. En cuanto a las entradas, se decidió que el diseño tuviese un mínimo de 8 entradas para que fuera capaz de realizar el control de cualquiera de los procesos mostrados anteriormente y adicionalmente tener capacidad extra para agregar más lazos de control, por otra parte, los sistemas de adquisición de datos comerciales más sencillos incluso llegan a tener tan solo 4 canales, de tal forma que la elección de 8 canales brinda características extras con respecto a los productos más sencillos del mercado. Para las salidas, se optó por que el sistema tuviese igualmente 8 canales, permitiéndole al usuario escoger entre salidas de tipo NPN o salidas a relé, esto con la finalidad de realizar labores de control avanzadas y brindarle más posibilidades al usuario final.

En cuanto a la elección de las señales que iba a ser capaz de medir la tarjeta de adquisición de datos, se consultó la obra de Calderón y Montero (2004) denominada

como **Mediciones e Instrumentación Industrial**, en ella se clasifican los tipos de señales mayormente empleadas por los transmisores industriales, destacando las señales de corriente de 4 mA a 20 mA, 0 a 20 mA y 10 mA a 50 mA, mientras que las señales estandarizadas de voltaje son de 1 V a 5 V, 0 a 10 V y -5 V a 5 V. De esta manera el resultado obtenido durante esta fase determinó que la tarjeta de adquisición a datos a diseñar durante la siguiente fase debía tener 8 canales de entrada y 8 canales de salida, así como ser capaz de medir señales estandarizadas de corriente y de voltaje distribuidas en los rangos mencionados.

Ya que la tarjeta de adquisición de datos tendría que ser capaz de medir múltiples tipos de señales, fue necesario elaborar una lista de componentes para el diseño tanto de la tarjeta como de los acondicionadores para las señales, primeramente, la tarjeta de adquisición de datos a diseñar debía ser operada mediante un microcontrolador, siendo el componente más importante de esta lista se inició por su elección, para ello se tomaron en cuenta dos criterios de selección, el primero de ellos fue la compatibilidad con el protocolo USB del mismo, ya que la tarjeta de adquisición de datos debía ser compatible con múltiples dispositivos y sistemas operativos, fue imperativo escoger un microcontrolador el cual tuviese amplia compatibilidad con el protocolo y documentación al respecto de su implementación, mientras que el segundo criterio que se empleó para seleccionarlo fue la cantidad de entradas y salidas disponibles, tomando en cuenta que la propuesta debía tener 8 canales de entradas y 8 canales de salida, el microcontrolador escogido debía poseer al menos 16 pines de I/O aunado a una serie de pines analógicos y digitales que realizarían diversas funciones internas dentro de la tarjeta. Siguiendo el primero de los criterios se decidió emplear una placa de prototipado de la empresa Arduino que, a pesar de utilizar internamente microcontroladores de Microchip, poseen un entorno de desarrollo independiente y una amplia gama de librerías especializadas, siendo la librería de comunicación Serial ideal para la propuesta a diseñar debido a su simplicidad y eficiencia.

Habiendo delimitado la selección del microcontrolador a las placas de prototipado Arduino, se aplicó el segundo de los criterios para determinar el microcontrolador a utilizar, para ello se recurrió a la siguiente tabla comparativa con los diferentes modelos de Arduino.

Cuadro 2: Tabla Comparativa Modelos de Arduino

	Uno	Mega 2560	Leonardo	Duo	ADK	Nano	Pro Mini
Microcontrolador	ATMega328	ATMega2560	ATMega328U4	AT91SAM3X8E	ATMEga2560	ATMega168	ATMega168
Pines Digitales	14	54	20	54	54	14	14
Pines PWM	6	15	7	12	15	6	6
Pines Analógicos	6	16	12	12	16	8	8
Memoria ROM	32K	256K	32K	512K	256K	16K	16K
Reloj	16 MHz	16 MHz	16 MHz	84 MHz	16 MHz	16 MHz	8 MHz
Conexión	USB	USB	Micro USB	Micro USB	USB Mini-B	USB Mini-B	Modulo USB Ext

Fuente: Freytes (2022)

Luego de observar las distintas características de cada uno de los modelos disponibles, se determinó que la opción más adecuada para los parámetros requeridos era el Arduino Mega, el cual cuenta con 54 pines digitales, 15 pines con funcionalidad PWM y 16 pines analógicos, todo esto adicionado a un conversor analógico – digital de 10 bits, lo hicieron el microcontrolador que mejor se adaptó a los requerimientos estipulados. Para la determinación del resto de componentes electrónicos a utilizar, se recurrió al diseño de un diagrama de funcionamiento de los canales de entrada de la tarjeta de adquisición de datos, obteniendo la siguiente figura.

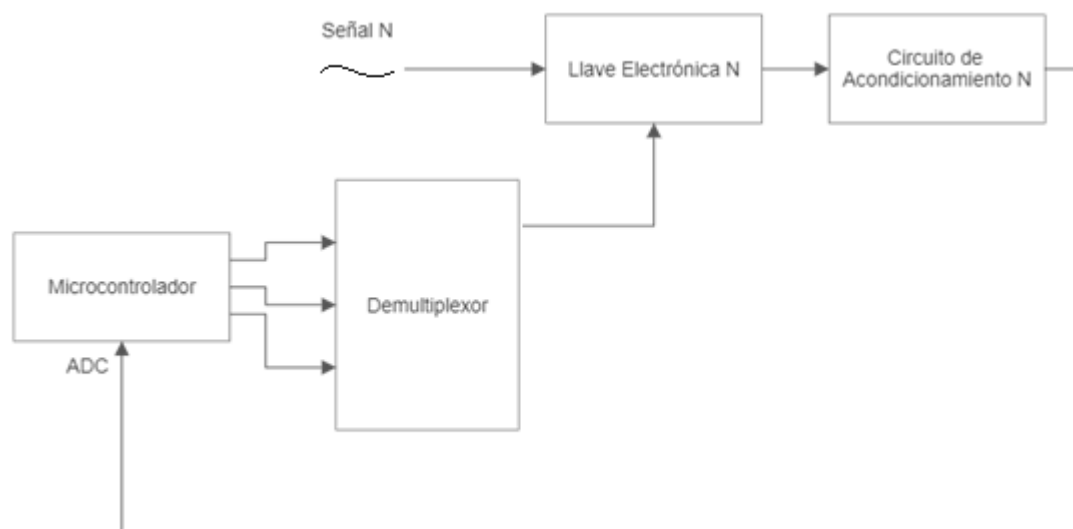


Figura 17: Diagrama de funcionamiento de los canales de entrada

Fuente: Freytes (2022)

Como se puede observar en la figura, el funcionamiento para cada canal de entrada sería el descrito a continuación, el microcontrolador mediante diversos pines de selección envía una señal binaria a un demultiplexor, cuya función sería la de determinar el acondicionamiento a aplicar a una determina señal según la orden dada por el microcontrolador, de tal forma que el demultiplexor activa una llave electrónica que permite el paso de la señal desde el canal de entrada hacia el circuito de acondicionamiento escogido para posteriormente ser leído por el ADC y procesado por el microcontrolador, por lo tanto, los siguientes componentes electrónicos que fueron determinados, fueron los que intervenían en el diagrama mostrado, el demultiplexor, la llave electrónica y los componentes para los diversos circuitos de acondicionamiento.

Debido a que fueron seis las señales escogidas para ser compatibles con la tarjeta de adquisición de datos, el demultiplexor a escoger debía ser de al menos seis canales de salida, por lo tanto, se determinó que la mejor opción para este componente sería un demultiplexor de 3 a 8, ya que cumple con el mínimo de canales de salida disponibles y posee al menos dos canales extras para utilizar en caso de ser necesario, de tal manera que el demultiplexor escogido sería el 74HC238, el mismo

fue seleccionado por su bajo consumo de potencia, costo reducido, capacidad para encontrarse en ambientes hostiles y su rango de alimentación varía entre los 2V y 6V, pudiendo ser alimentado con la misma fuente usada en el microcontrolador.

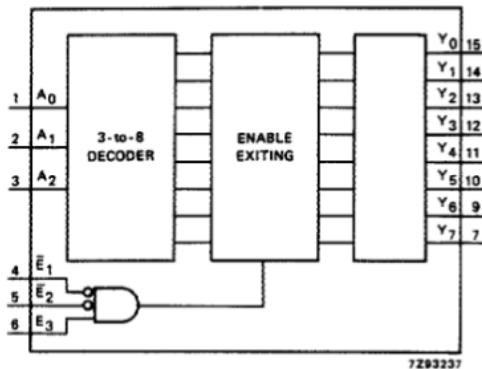


Figura 18: Diagrama funcional del demultiplexor 74HC238

Fuente: Philips (1990)

El siguiente componente a determinar sería la llave electrónica, ya que se iban usar múltiples de ellas, el criterio de selección fue basado en encapsulados que poseían más de una llave electrónica en su interior, tomando en cuenta este criterio, la opción escogida fue el CD4066B, el cual es un integrado cuádruple bilateral de tipo CMOS, que permite activar un interruptor electrónico mediante una señal de control aplicada a uno de sus pines, pudiendo ser usado según especifica el diagrama diseñado. En cuanto a los circuitos de acondicionamiento, los mismos serían diseñados mediante el uso de amplificadores operacionales, por esta razón, el siguiente componente a determinar debían ser los amplificadores operacionales, los cuales serían determinados mediante el mismo criterio que el caso anterior, múltiples operacionales en el interior de un mismo encapsulado a un costo reducido, de tal forma que el integrado seleccionado fue el LM324. Dicho integrado posee en su interior cuatro amplificadores operacionales de propósito general a un muy bajo costo, siendo la opción adecuada para la propuesta a diseñar.

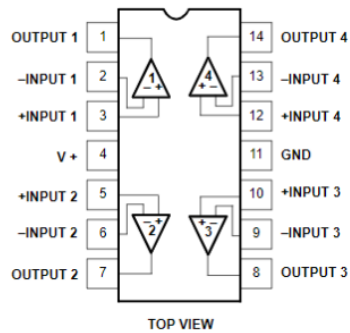


Figura 19: Esquema del Integrado LM324N

Fuente: Philips (1995)

Para la alimentación tanto de los integrados de la tarjeta de adquisición de datos como el microcontrolador fue escogida una tensión de 5V, permitiendo utilizar la misma tensión para alimentar cada uno de los elementos circuitales a excepción de los amplificadores operacionales, para ello se escogió el integrado LM7805, el cual es un regulador lineal de 5V que brinda una tensión constante a la salida si es alimentado con una tensión superior a los 7V, adicionalmente, puede surtir una corriente máxima de hasta 1A, cumpliendo perfectamente con los requerimientos del diseño.

ABSOLUTE MAXIMUM RATINGS (Ta = 25 °C)

Parameter	Symbol	Typ	Unit
Input Voltage	V_i	35	V
Output Voltage	V_o	5.0	V
Peak Current	I_{PK}	2.2	A
Operating Temperature Range	T_{OPR}	0-125	°C
Storage Temperature Range	T_{STG}	-65-150	°C

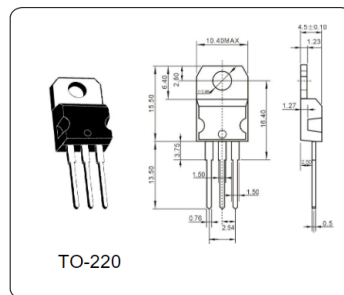


Figura 20: Hoja de datos del LM7805

Fuente: Fairchild Semiconductor (2006)

Finalmente, como elementos circuitales adicionales fue necesario escoger transistores para colocar en los canales de salida, los cuales se escogieron los transistores NPN 2N2222 por su bajo costo y prestaciones, que, aunado a sus parámetros de frecuencia los hacen una buena elección para situarse en las salidas

PWM del microcontrolador. Por otra parte, también fue necesario seleccionar las borneras para los canales de entrada, salida y la alimentación, para los dos primeros se seleccionaron borneras de dos pines, mientras que para la fuente de poder se seleccionó una bornera de tres pines ya que dicha fuente tendría que ser simétrica para alimentar los amplificadores operacionales LM324, en la siguiente figura se observan las borneras seleccionadas.



Figura 21: Borneras de 2 y 3 Pines

Fuente: Freytes (2022)

4.2 Fase II. Diseño de la Propuesta

Durante esta fase se realizó el diseño de la tarjeta de adquisición de datos siguiendo los parámetros definidos durante la fase anterior, de igual forma durante esta fase se generaron los esquemáticos del diseño electrónico finalizado, así como el modelado 3D del circuito y su estructura de almacenamiento, para finalizar con el diseño de la interfaz gráfica o HMI y la simulación de la propuesta de la tarjeta de adquisición de datos comunicándose con la misma.

4.2.1 Diseño de la Tarjeta de Adquisición de Datos

Habiendo determinado todos los parámetros y componentes a utilizar en la propuesta durante la fase anterior, se inició con el diseño del circuito de la UIAD o Unidad Integrada de Adquisición de Datos, para ello se empleó el simulador de circuitos electrónicos Proteus, que, además de permitir realizar el diseño del circuito, permitió simular y probar el mismo, una vez estuvo finalizado su diseño. La primera parte del circuito de la UIAD que se elaboró fue el sistema de acondicionamiento de las señales, ya que cada una de las entradas sería capaz de leer todos los tipos de señales establecidas en la fase previa, entonces fue necesario diseñar un sistema que

basado en una selección del usuario desde la HMI permitiera determinar el acondicionamiento indicado para la señal de la entrada, para ello se empleó un sistema de llaves electrónicas utilizando el integrado 4066, del cual se muestra su diagrama a continuación.

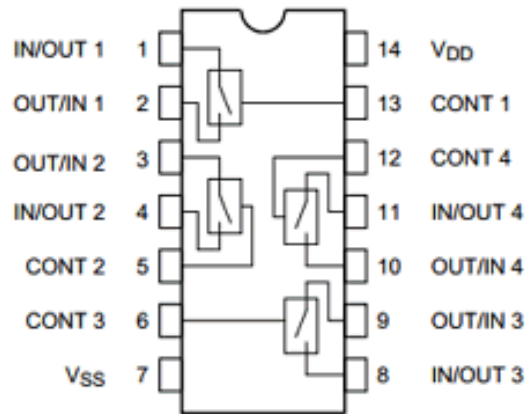


Figura 22: Diagrama del Integrado 4066

Fuente: Unisonic Technologies (2020)

La función de esta llave electrónica es la de cerrar el circuito entre un par de pines del integrado mediante una señal alta aplicada en el pin de control, su integración en el diseño de la UIAD permite determinar el acondicionamiento a aplicar a la señal según el interruptor que sea activado a través de la señal de control. Para el control de los interruptores del 4066 se empleó un demultiplexor 74HC238, que mediante tres pines de entrada permite activar una de las 8 salidas del mismo, de esta forma, al enviar la selección del acondicionamiento mediante la combinación binaria de tres pines del microcontrolador, este, automáticamente activará el interruptor determinado para el acondicionamiento específico, de tal manera que la señal solo es acondicionada una única vez por el circuito adecuado para ello. En la figura 19 se puede observar el demultiplexor 74HC238 con las conexiones realizadas.

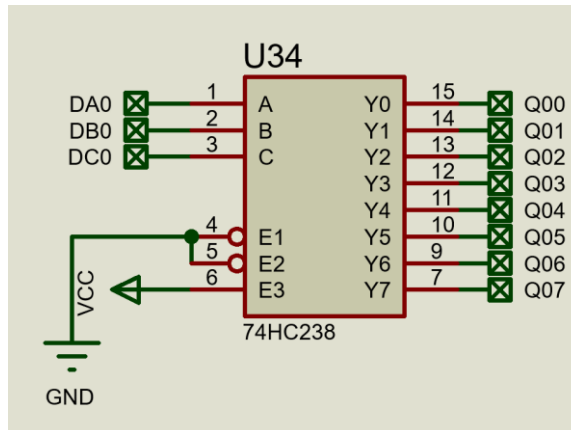


Figura 23: Esquema del circuito empleando el demultiplexor 74HC238

Fuente: Freytes (2022)

En la parte izquierda del integrado se observan las conexiones de las entradas binarias del mismo, las cuales permiten seleccionar la salida que será activada, se denotan con las letras A, B y C, donde C es el bit más significativo del número binario, a continuación, se muestra la tabla de la verdad del integrado, donde se incluyen las diferentes combinaciones de los bits de entrada y sus respectivas salidas.

INPUTS						OUTPUTS							
\bar{E}_1	\bar{E}_2	E_3	A_0	A_1	A_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
H	X	X	X	X	X	L	L	L	L	L	L	L	L
X	H	X	X	X	X	L	L	L	L	L	L	L	L
X	X	L	X	X	X	L	L	L	L	L	L	L	L
L	L	H	L	L	L	H	L	L	L	L	L	L	L
L	L	H	H	L	L	L	H	L	L	L	L	L	L
L	L	H	L	H	L	L	L	H	L	L	L	L	L
L	L	H	H	H	L	L	L	L	L	H	L	L	L
L	L	H	L	L	H	L	L	L	L	L	H	L	L
L	L	H	H	H	H	L	L	L	L	L	L	H	L
L	L	H	H	H	H	L	L	L	L	L	L	L	H

Figura 24: Tabla de la verdad del demultiplexor 74HC238

Fuente: Philips (1990)

Como se observa en la figura 20, para cada una de las combinaciones de los dígitos binarios a la entrada del demultiplexor, se obtiene una única salida activa, de tal forma, que empleando tres salidas del microcontrolador por cada uno de los canales, se puede determinar el acondicionamiento que se realizará en cada uno de ellos, los pines empleados para esta función en el microcontrolador abarcan desde el

pin 22 del Arduino Mega hasta el pin 45 y se denotaron con la letra “D” indicando que son pines digitales, seguido de la letra “A”, “B” o “C” dependiendo del bit que controle en el arreglo binario y por último un dígito numérico referente al canal al que pertenece, los mismos van desde el número 0 al 7.

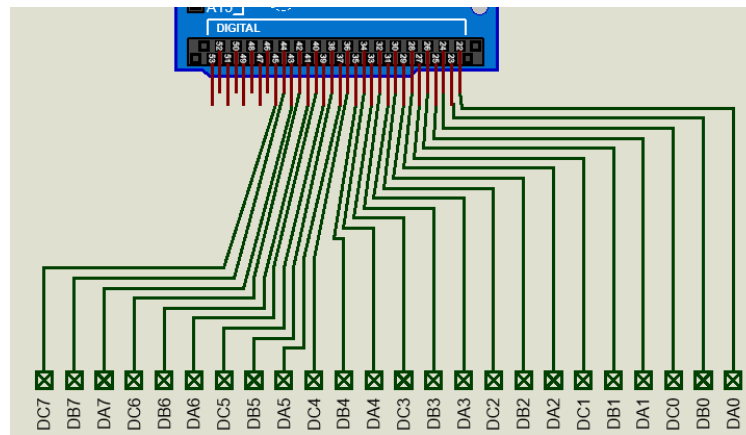


Figura 25: Pines de control del acondicionamiento de las señales de la UIAD

Fuente: Freytes (2022)

Una vez diseñado el sistema que se iba a emplear para escoger el acondicionamiento de la señal adecuado para cada uno de los canales de entrada de la UIAD, se procedió a diseñar los acondicionamientos para los tipos de señales estandarizadas determinadas en la fase anterior. Ya que los pines de control del acondicionamiento conectados a los demultiplexores inicialmente se encuentran en un estado lógico bajo, se decidió por no emplear la primera de las salidas brindadas por el 74HC238 o “Y₀”, ya que se activaría involuntariamente al momento de iniciar la UIAD, de tal forma que el primero de los acondicionamientos, es decir, el acondicionamiento para la señal de corriente de 4 mA a 20 mA, estaría controlado por la segunda salida ofrecida por el demultiplexor es decir la salida “Y₁”. En la siguiente figura, se muestra el circuito de acondicionamiento del primer tipo de señal.

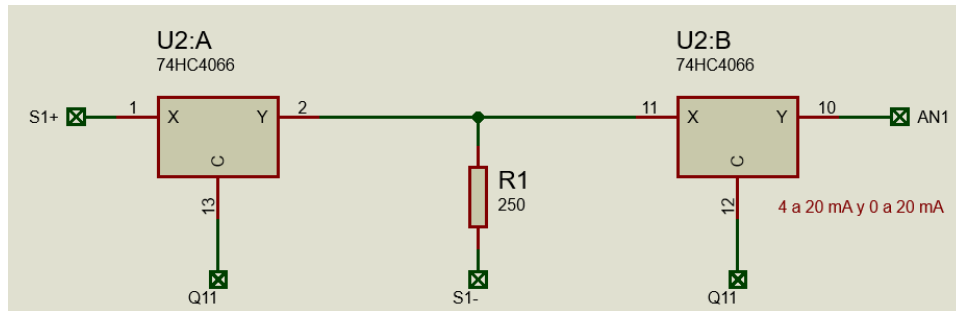


Figura 26: Acondicionamiento de la señal de 4 mA a 20 mA

Fuente: Freytes (2022)

Debido a que la primera de las señales empleadas es una señal de corriente estandarizada, se optó por realizar un acondicionamiento sencillo mediante una resistencia de medición de 250Ω , en este circuito, la señal ingresa al primer interruptor de la llave electrónica 4066 a través del terminal “SX+”, el cual solo es activado si los pines de control del acondicionamiento de la UIAD determinan que la salida del 74HC238 debe ser el pin “Y₁” el cual está conectado a su vez a los terminales “QX1”, de tal forma, que si la salida “Y₁” del demultiplexor es activada, simultáneamente se activarán el par de llaves del acondicionamiento, la primera de ellas separa la señal estandarizada de los acondicionamientos para que no sean empleados si no es necesario, mientras que la segunda es empleada como un filtro que evita el retorno de la señal hacia otros acondicionamientos, ya que todos los acondicionamientos finalizan en el mismo terminal “ANX” se debe emplear un sistema de protección que evite retornos. Luego que ambas llaves se activan, comienza a fluir la corriente desde “SX+” hasta “SX-” pasando a través de la resistencia de 250Ω , la cual producirá una caída de tensión proporcional al valor de la señal de corriente que será medida empleando el ADC del Arduino Mega, para ello se reservaron 8 pines del ADC del microcontrolador, uno para cada canal de entrada, los cuales se denotaron con las letras “AN” seguido del número del canal al cual debía sensar. Los pines reservados abarcan desde el pin 54 hasta el pin 61 del Arduino Mega y los mismos se muestran a continuación.

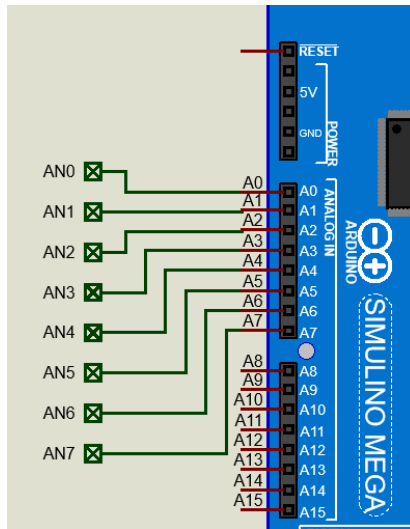


Figura 27: Pines Analógicos del ADC reservados para la medición de los canales de entrada

Fuente: Freytes (2022)

En cuanto al acondicionamiento de la segunda señal estandarizada, es decir la señal de corriente de 0 mA a 20 mA, ya que cubre el mismo rango de operación de la señal anterior se optó por emplear el mismo acondicionamiento para ambas señales, solo que en para el caso de la señal de 0 mA a 20 mA, no tendría ningún tipo de verificación de errores en la conexión como si lo tiene la señal de 4 mA a 20 mA al descender por debajo del rango mínimo.

Por parte de la tercera señal estandarizada se decidió emplear el mismo sistema de acondicionamiento de la señal de 4 mA a 20 mA, ya que la misma abarca desde los 10 mA hasta los 50 mA, se incluyó una resistencia de medición de 100 Ω , que brinda una caída de tensión de entre 1V y 5V dependiendo del valor de la señal, por lo tanto, si los pines de control del acondicionamiento de la UIAD definen como salida del demultiplexor a "Y₂", entonces se activa el respectivo acondicionamiento mediante las llaves controladas por "QX2", fluyendo la corriente a través de la resistencia de 100 Ω , generando una caída de tensión medida por el pin "ANX". En la figura 24 se demuestra el acondicionamiento de las señales de 10 mA a 50 mA.

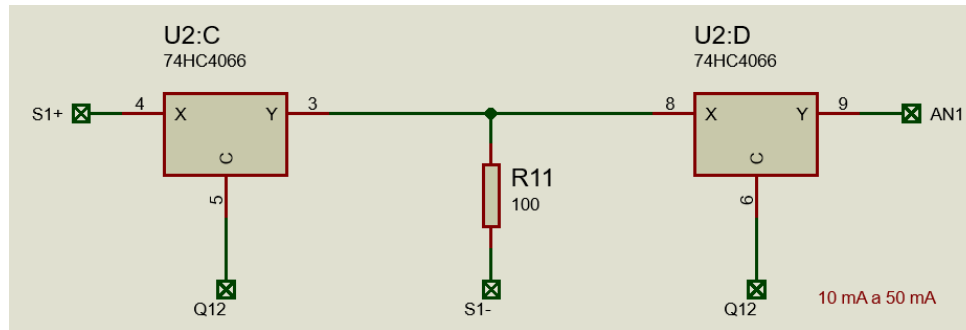


Figura 28: Acondicionamiento de la señal de 10 mA a 50 mA

Fuente: Freytes (2022)

Avanzando con el acondicionamiento de la siguiente señal, se trata de la señal de -5 V a 5 V, siendo la primera de las señales estandarizadas de voltaje se diseñó un acondicionamiento fundamentado en el concepto relatado en el capítulo 2, el cual consiste en la adaptación de una señal cuyos límites máximos superan el fondo de escala del ADC mediante la atenuación de la misma para posteriormente agregar un nivel de offset de DC, de tal forma que la señal no supere los límites establecidos por el fabricante del microcontrolador. Ya que la señal de voltaje se encontraba entre los 5 V y -5 V, primeramente, se procedió a realizar el proceso de atenuación de la señal, para ello se empleó un arreglo de resistencias en la configuración de divisor resistivo con la finalidad de atenuar la señal a la mitad de su valor, por consiguiente, se utilizaron dos resistencias de $10\text{K}\Omega$, produciendo una tensión de salida atenuada entre 2.5 V y -2.5 V. Tomando en cuenta que la señal debía ser adicionada con un offset de DC para eliminar los valores negativos, se optó igualmente por agregar al divisor resistivo un amplificador operacional LM324 en configuración de seguidor unitario, de tal manera que el acoplamiento a otros arreglos de operacionales no afecte la tensión atenuada.

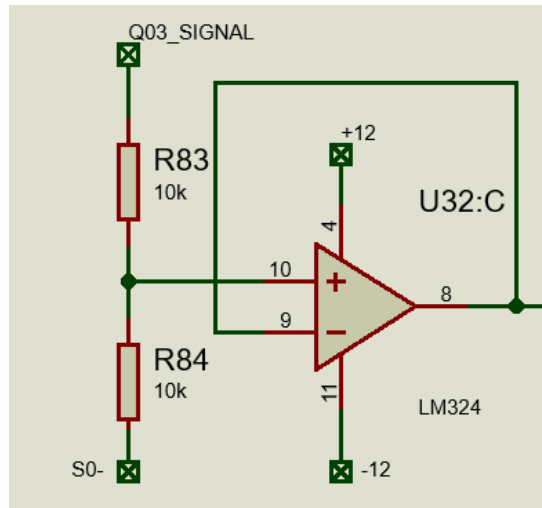


Figura 29: Acondicionamiento de la señal de -5V a 5V; Atenuación de la señal

Fuente: Freytes (2022)

Una vez la señal ya se encontraba atenuada, se procedió a realizar la adición del offset de DC a la señal original, para ello se empleó un arreglo de operacionales en configuración sumador inversor con ganancia unitaria, donde la primera de las señales del sumador sería la señal atenuada, mientras que la segunda señal del sumador sería el offset de DC. Para obtener una tensión constante que sirviera como offset se empleó el mismo método que para atenuar la señal, ya que se requería un offset de 2.5 V para que la señal atenuada quedara limitada a un rango de 0 a 5 V, se empleó un divisor resistivo en la alimentación de 12 V con dos resistencias de 38K Ω y 10K Ω respectivamente, resultando una tensión de 2.5 V constantes, que, adherido a un operacional como seguidor unitario, permitieron la adición de ambas señales mediante el sumador inversor.

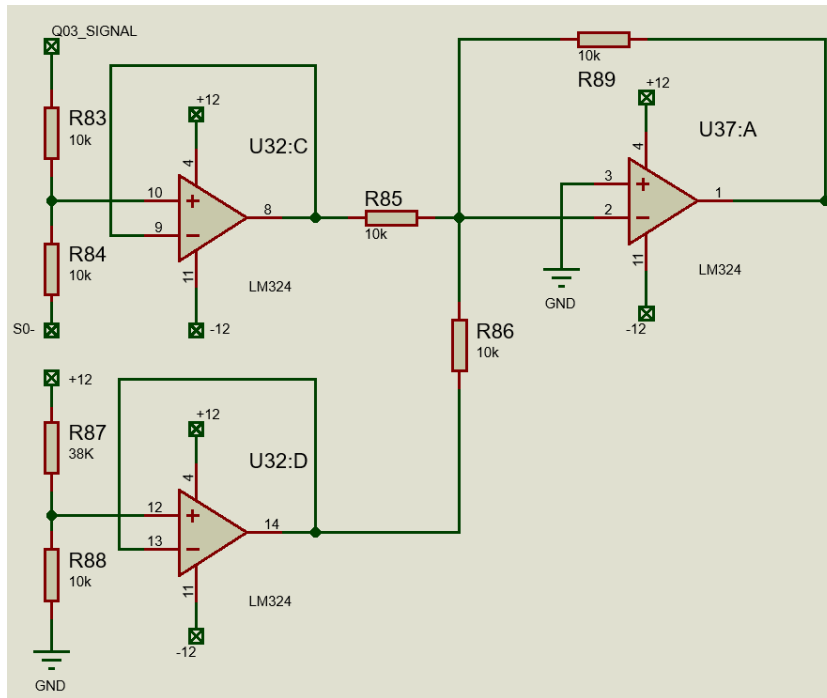


Figura 30: Acondicionamiento de la señal de -5V a 5V; Adición del offset de DC

Fuente: Freytes (2022)

Finalmente, ya que la señal es invertida por el sumador inversor, se empleó un amplificador inversor de ganancia unitaria para revertir la inversión de la etapa anterior, culminando así con el acondicionamiento de la señal de -5 V a 5 V. Siguiendo con la línea de los acondicionamientos anteriores, este proceso también es controlado por los pines de control de acondicionamiento de la UIAD, los cuales si determinan que la salida del 74HC238 debe ser la salida “Y₃”, entonces activa ambas llaves electrónicas pudiendo ser medida la señal por el ADC a través del pin “ANX”. A continuación, se muestra el acondicionamiento de la señal culminado.

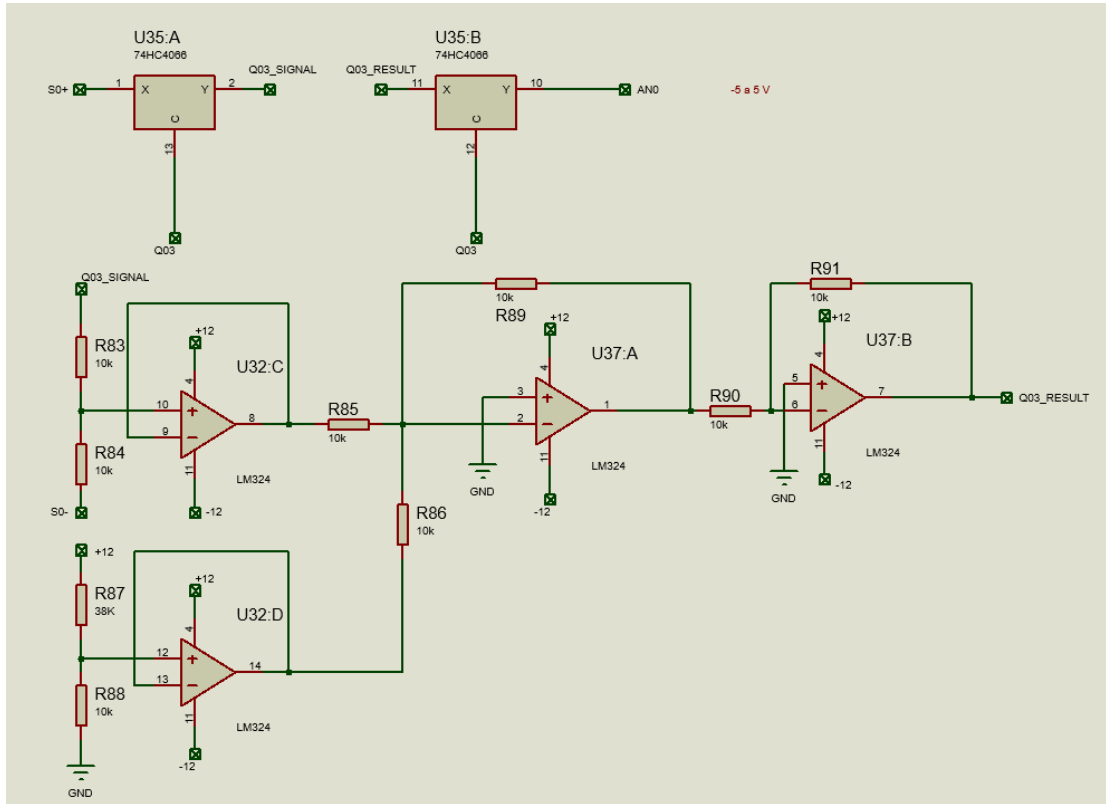


Figura 31: Acondicionamiento de la señal de -5V a 5V

Fuente: Freytes (2022)

Para los dos últimos acondicionamientos se emplearon métodos más sencillos ya que las señales solo requerían ser atenuadas para encontrarse dentro del rango permitido por el ADC. Para la señal de 0 V a 10 V, se empleó un divisor resistivo aunado a un seguidor unitario, ya que el ADC admite señales de 0 V a 5 V, solo fue necesario atenuar la señal un 50%, resultando en una señal de 0 V a 5 V compatible con el ADC, para ello se emplearon dos resistencias de 10KΩ. Por otra parte, el seguidor unitario aquí se utiliza para que la señal no se vea afectada ni por las impedancias de los elementos circuitales aledaños ni por las impedancias de los integrados.

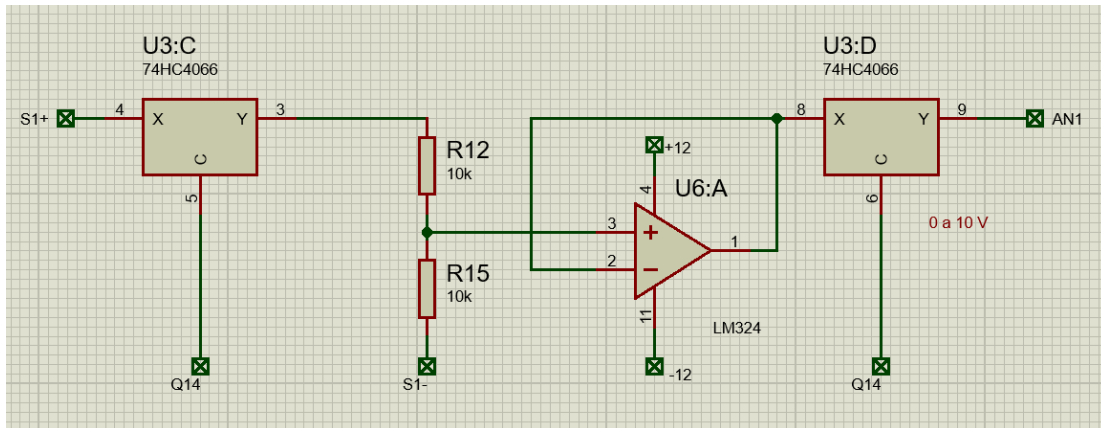


Figura 32: Acondicionamiento de la señal de 0 V a 10 V

Fuente: Freytes (2022)

En cuanto a la última señal por acondicionar, ya que la misma ya se encuentra dentro del rango permitido por el ADC, debido a que son señales de entre 1 V y 5 V, tan solo se empleó un operacional en configuración de seguidor de tensión con la misma finalidad del acondicionamiento anterior, de igual forma en ambos acondicionamientos, si los pines de control del acondicionamiento de la UIAD determinan que la salida del demultiplexor debe ser “Y4” o “Y5”, entonces se activarán las llaves electrónicas respectivas, realizando la medición con el uso del ADC y un canal determinado.

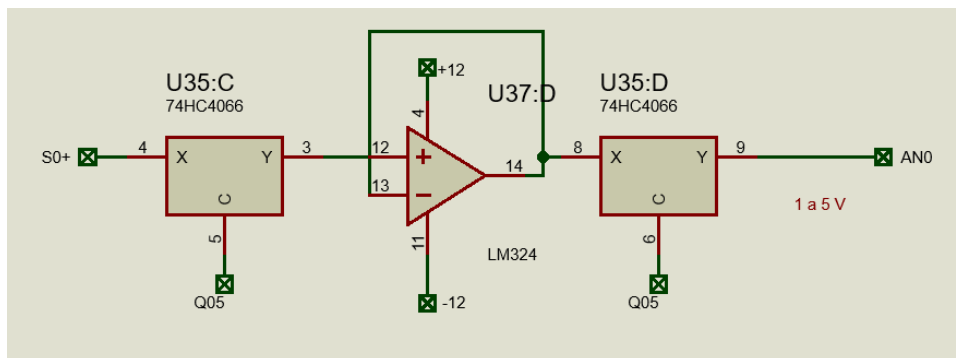


Figura 33: Acondicionamiento de la señal de 1 V a 5 V

Fuente: Freytes (2022)

Es importante destacar que todos los operacionales empleados para los acondicionamientos emplean fuentes de alimentación simétricas, es decir, requieren tanto la aplicación de una tensión positiva como una tensión negativa, es por esto que la alimentación de la UIAD, está conformada por una fuente de tensión simétrica de +12V y -12V, el diseño de dicha fuente de tensión no fue cubierto por la presente obra, sin embargo, en el apéndice A, se presenta un posible diseño para una fuente de tensión simétrica de 12 V. Para la alimentación del resto de integrados, incluido el Arduino Mega, se requirió de una alimentación de 5V, para lo cual se empleó el integrado LM7805 conectado a la alimentación de +12 V que suministra 5V hasta 1.5 A de corriente, suficiente para cubrir la alimentación del resto de elementos del circuito.

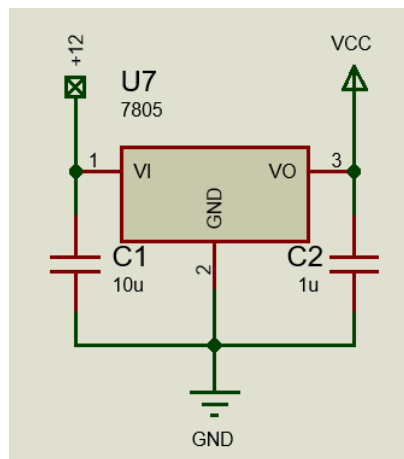


Figura 34: Alimentación de 5V de la UIAD empleando el LM7805

Fuente: Freytes (2022)

Una vez se realizaron todos los acondicionamientos para cada uno de los canales de entrada, se procedió a realizar los canales de salida de la UIAD, ya que la misma sería capaz de realizar diversos tipos de control, se emplearon salidas a transistor conectadas a ocho pines PWM del Arduino Mega, por consiguiente, la UIAD sería capaz tanto de realizar controles de tipo ON/OFF empleando relés a la salida de los transistores o incluso controlar electroválvulas. Los transistores empleados para los canales de salida fueron los 2N2222 que brindan buenas

prestaciones por un costo reducido, a la base de los mismos fueron conectados los pines PWM del microcontrolador que a través de una resistencia de $10K\Omega$, limitan la corriente de salida del Arduino a 0.5 mA , pudiendo controlar cargas de hasta 50mA con el 2N2222, en caso de que se desee controlar cargas mayores es posible conectar a la salida de transistor de la UIAD otros transistores en configuración Darlington para incrementar la corriente de salida.

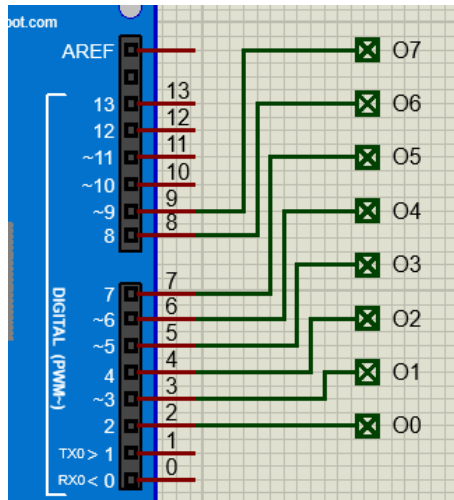


Figura 35: Pines PWM de salida de la UIAD

Fuente: Freytes (2022)

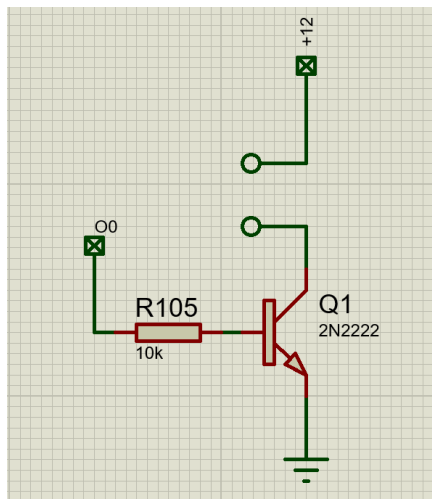
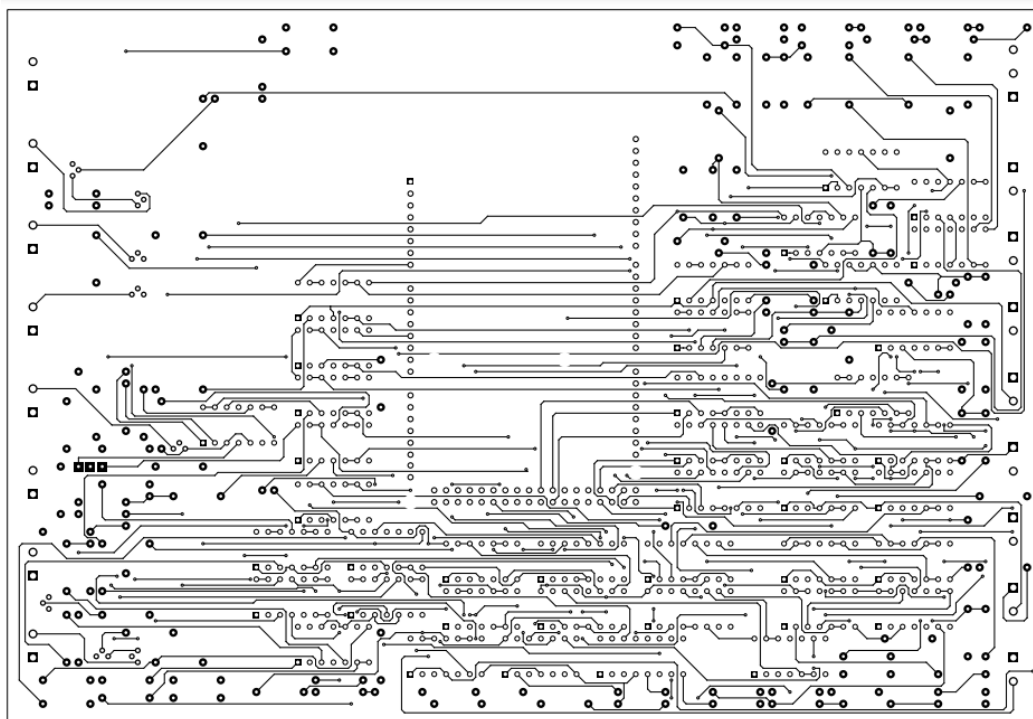


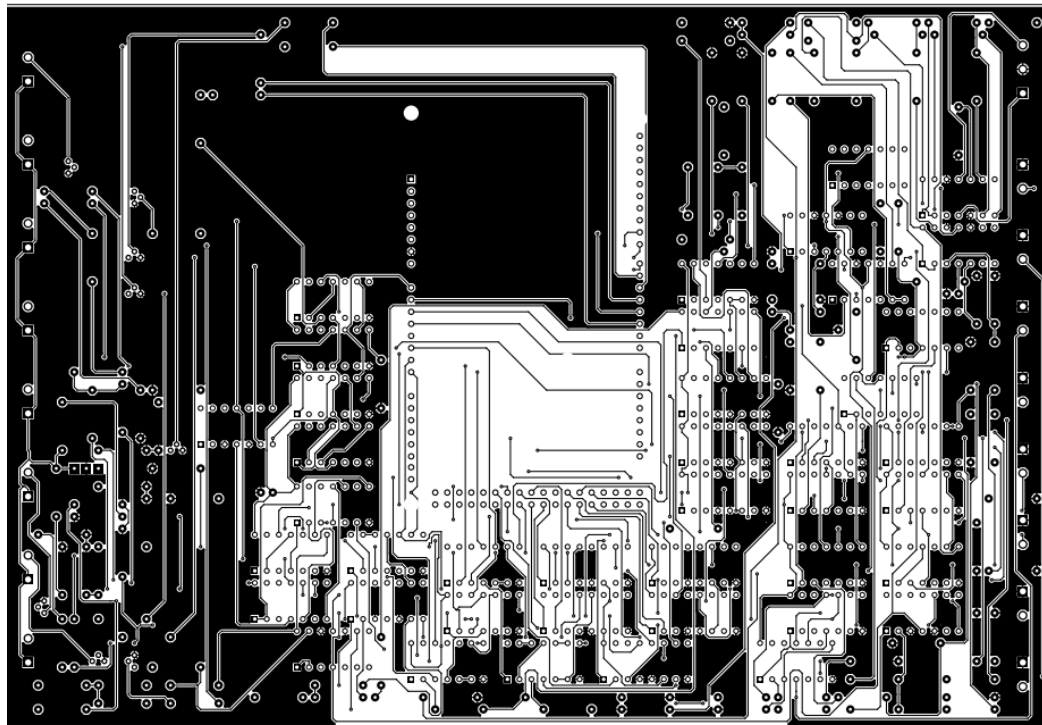
Figura 36: Salida a transistor de los canales de la UIAD

Fuente: Freytes (2022)

Luego de culminar tanto con los canales de entrada como los canales de salida de la UIAD, se procedió a generar todos los diagramas y esquemáticos del circuito, para ello fue necesario realizar algunas correcciones a los diagramas circuitales, ya que la simulación no permite incluir las borneras de forma previa y el modelo simulado del Arduino Mega no posee plantilla para PCB. Una vez agregadas las borneras y sustituido el Arduino Mega por su plantilla para PCB, se generaron los diagramas finales, el diagrama de PCB y el modelo 3D de la UIAD, tanto para el diagrama de PCB como el modelo 3D fue necesario definir las dimensiones de la tarjeta electrónica, las cuales, tomando en cuenta las dimensiones del Arduino Mega y el número de borneras empleadas, se establecieron en 20cm de largo por 15.3 cm de ancho, que, luego de incluidas en Proteus, resultaron en la generación de los siguientes diagramas.



a)



b)

Figura 37: Diagrama para PCB de la UIAD; a) Capa Superior b) Capa Inferior

Fuente: Freytes (2022)

Debido a la gran cantidad de componentes empleados en el diseño de la UIAD, se optó por utilizar un PCB de doble capa, es decir con pistas impresas por ambas caras del mismo, esto con la finalidad de reducir el tamaño final de la tarjeta electrónica y evitar problemas a la hora de generar las conexiones en el programa de diseño. Por otra parte, Proteus también ofrece la posibilidad de visualizar el modelo 3D del diseño electrónico realizado, pudiendo exportar el modelo a otros programas de diseño 3D para la realización de mejoras en el mismo, en este caso, se exportó el modelo 3D con la finalidad de diseñar una estructura de almacenamiento para la UIAD, a continuación, se muestra el modelo 3D de la UIAD generado mediante el programa de diseño de circuitos electrónicos Proteus.

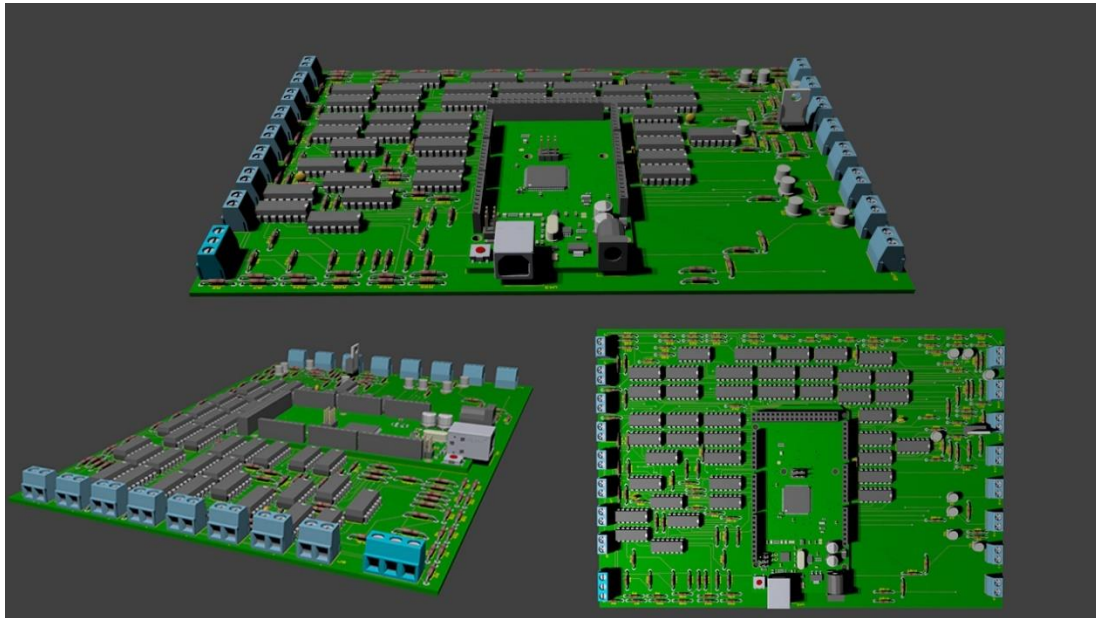


Figura 38: Modelo 3D de la UIAD

Fuente: Freytes (2022)

La estructura de almacenamiento de la UIAD fue diseñada en un programa de modelado 3D denominado Autodesk 3DS Max, la misma está conformada por dos piezas de plástico ensambladas a presión, que brindan soporte y protección al circuito, la pieza inferior posee 4 cilindros sobresalientes sobre los cuales descansa la tarjeta y permite fijarla mediante tornillos en la parte central de la placa. Una vez estuvo finalizado su diseño se generaron imágenes renderizadas de la misma, obteniendo los siguientes resultados.

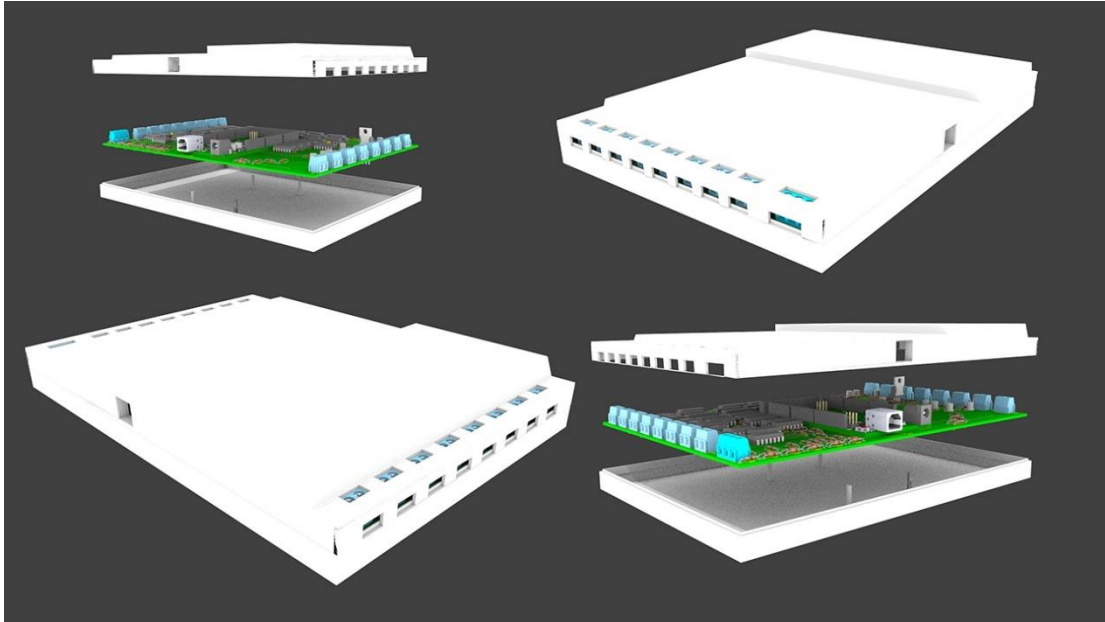


Figura 39: Carcasa de almacenamiento de la UIAD

Fuente: Freytes (2022)

Por último, se muestran los esquemáticos finalizados de la UIAD, los mismos se encuentran divididos por secciones, la primera de ellas agrupa al Arduino Mega con sus conexiones, las borneras para la fuente de alimentación simétrica y el circuito del LM7805, de la sección número dos a la sección número nueve, se encuentran los circuitos de acondicionamiento para cada uno de los ocho canales, así como su respectivo demultiplexor 74HC238 y las borneras de entrada de la señal. Finalmente, la sección número 10 corresponde a los canales de salida, en ella están diagramados los 8 canales con sus respectivas salidas a transistor y las borneras de 2 pines.

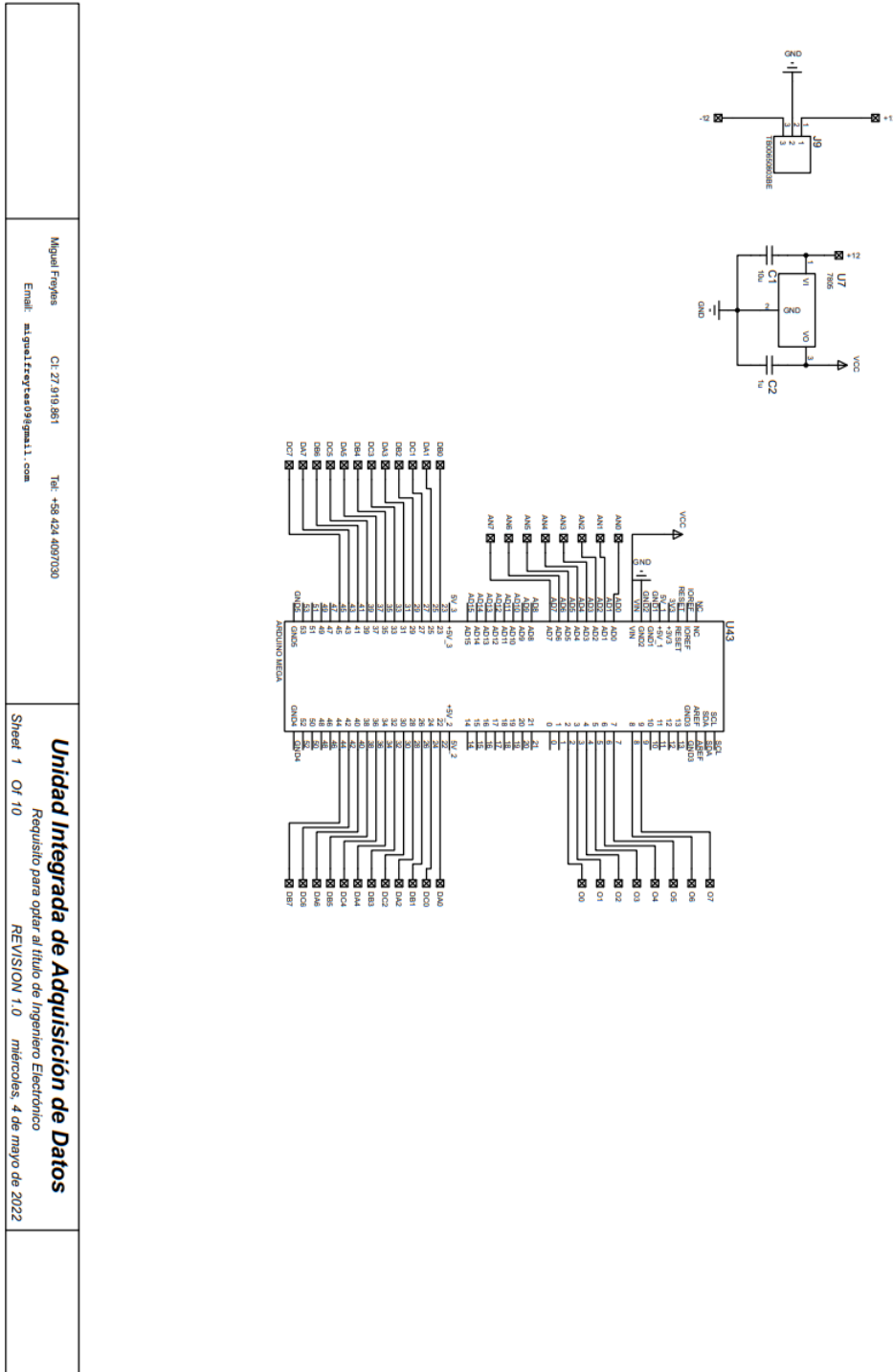


Figura 40: Diagrama Esquemático de la UIAD; Microcontrolador y Alimentación
Fuente: Freytes (2022)

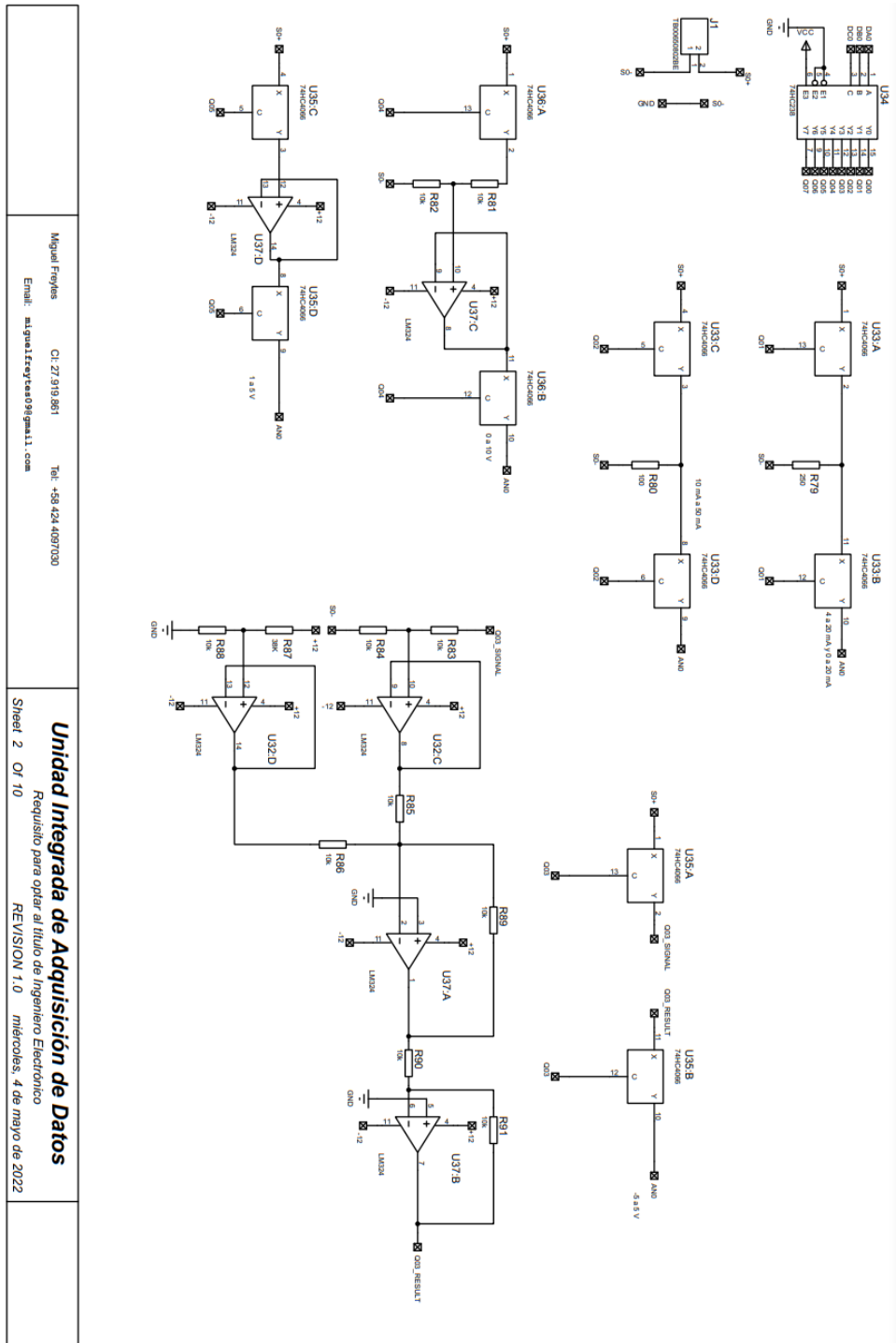


Figura 41: Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 0

Fuente: Freytes (2022)

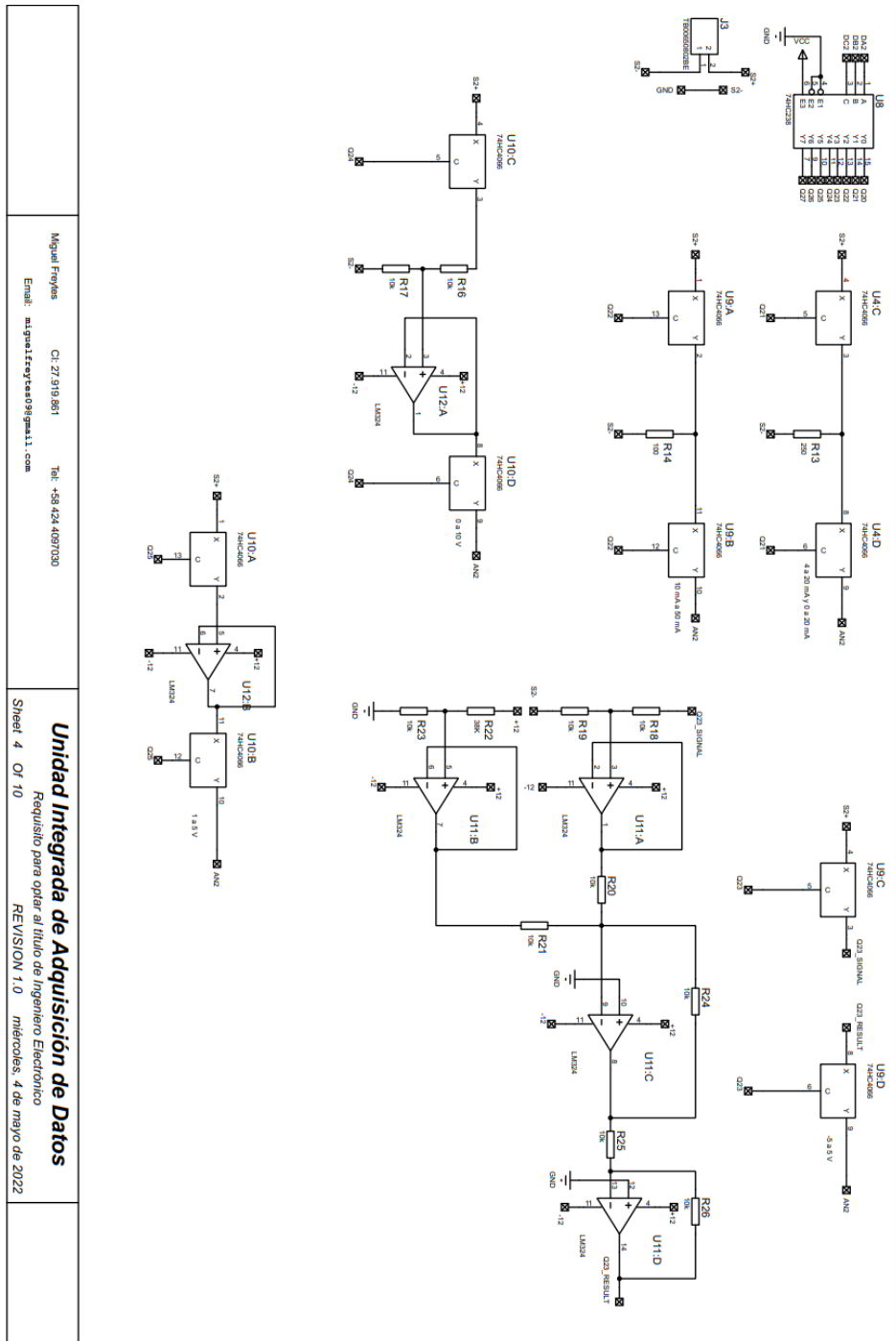


Figura 43: Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 2

Fuente: Freytes (2022)

<p>Miguel Freytes C.I: 27.919.861 Tel: +58 424 4097030</p> <p>Email: miguel.freytes@gmail.com</p>	<p>Unidad Integrada de Adquisición de Datos</p> <p>Requisito para optar al título de Ingeniero Electrónico</p> <p>REVISION 1.0 miércoles, 4 de mayo de 2022</p>
---	---

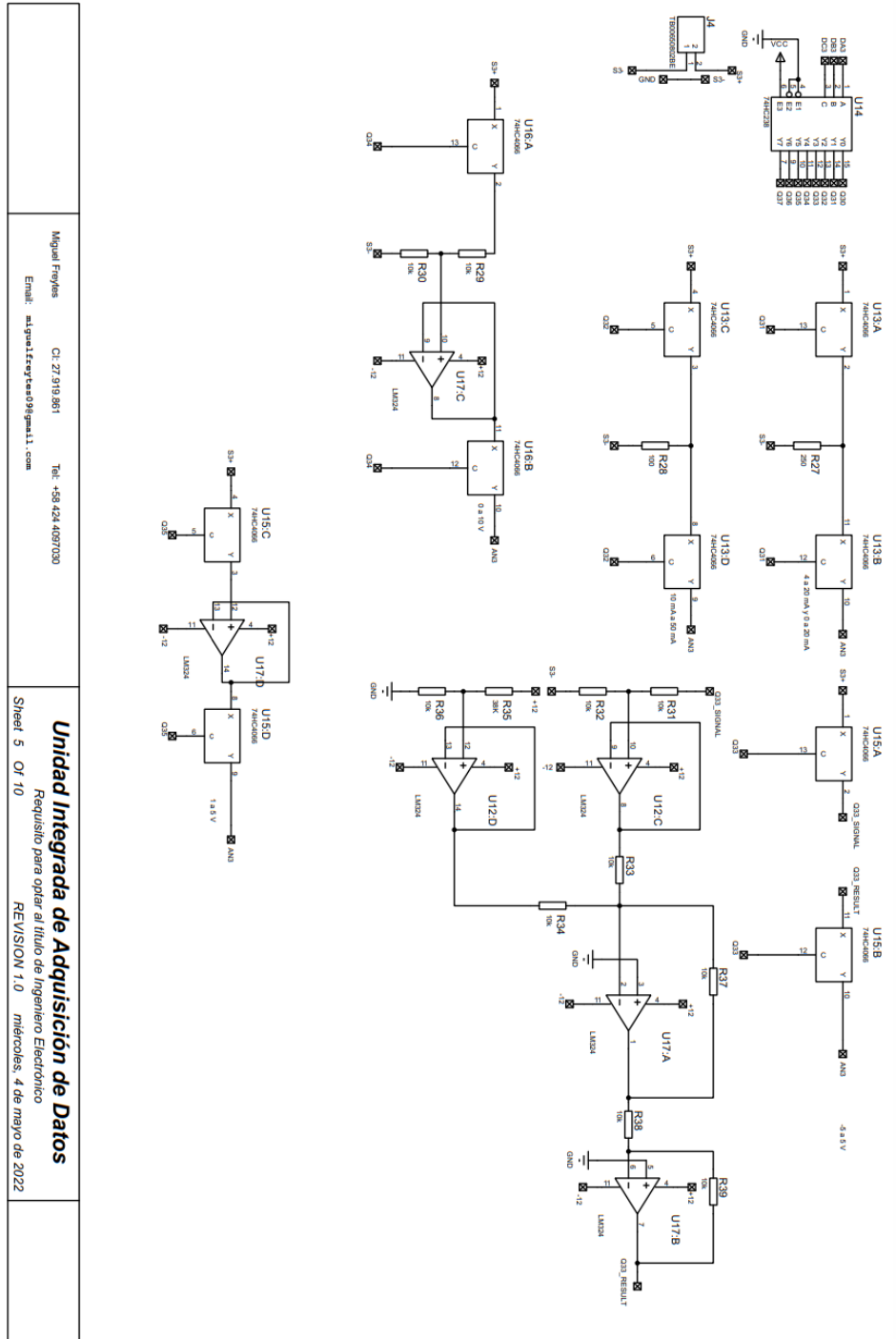


Figura 44: Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 3

Fuente: Freytes (2022)

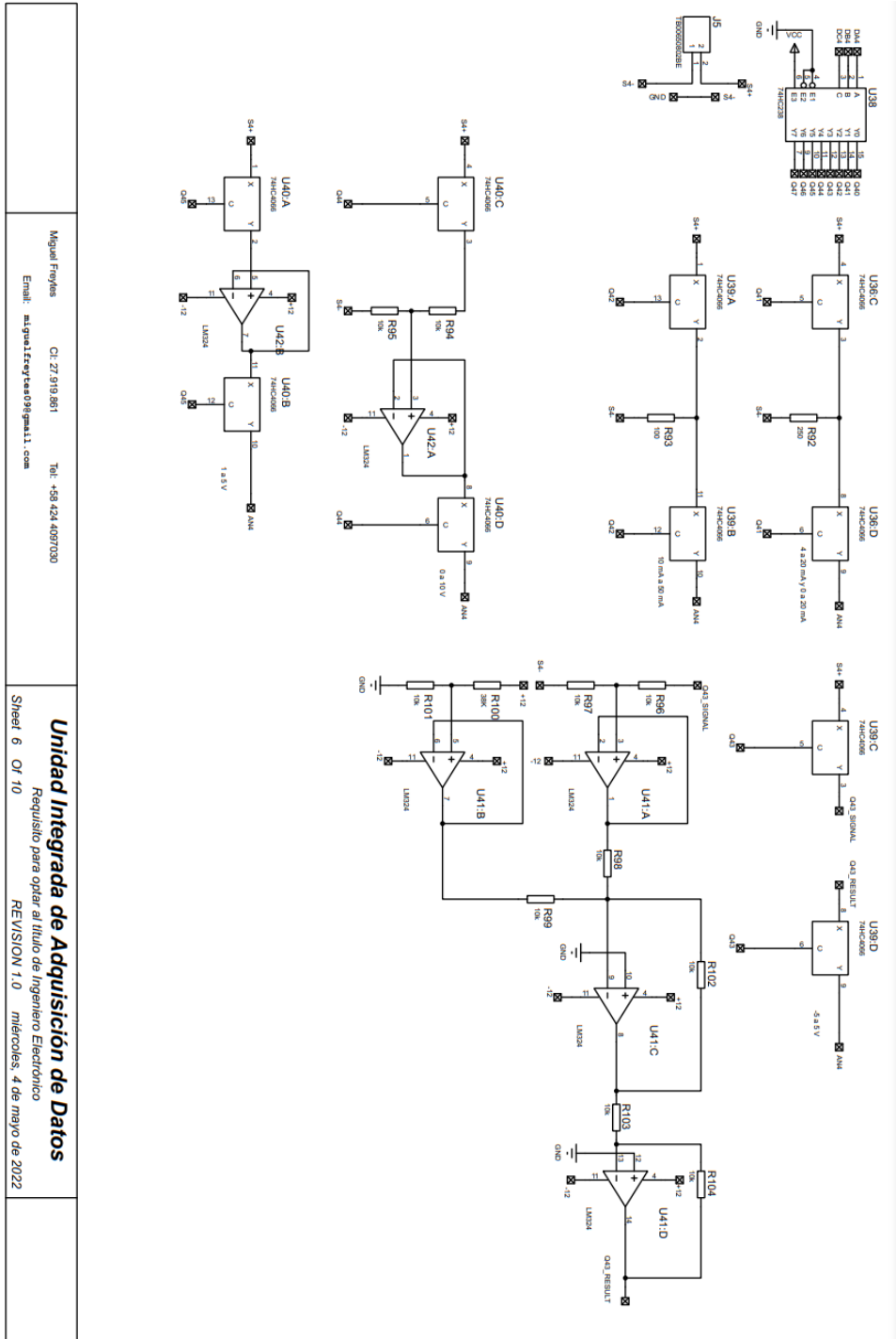
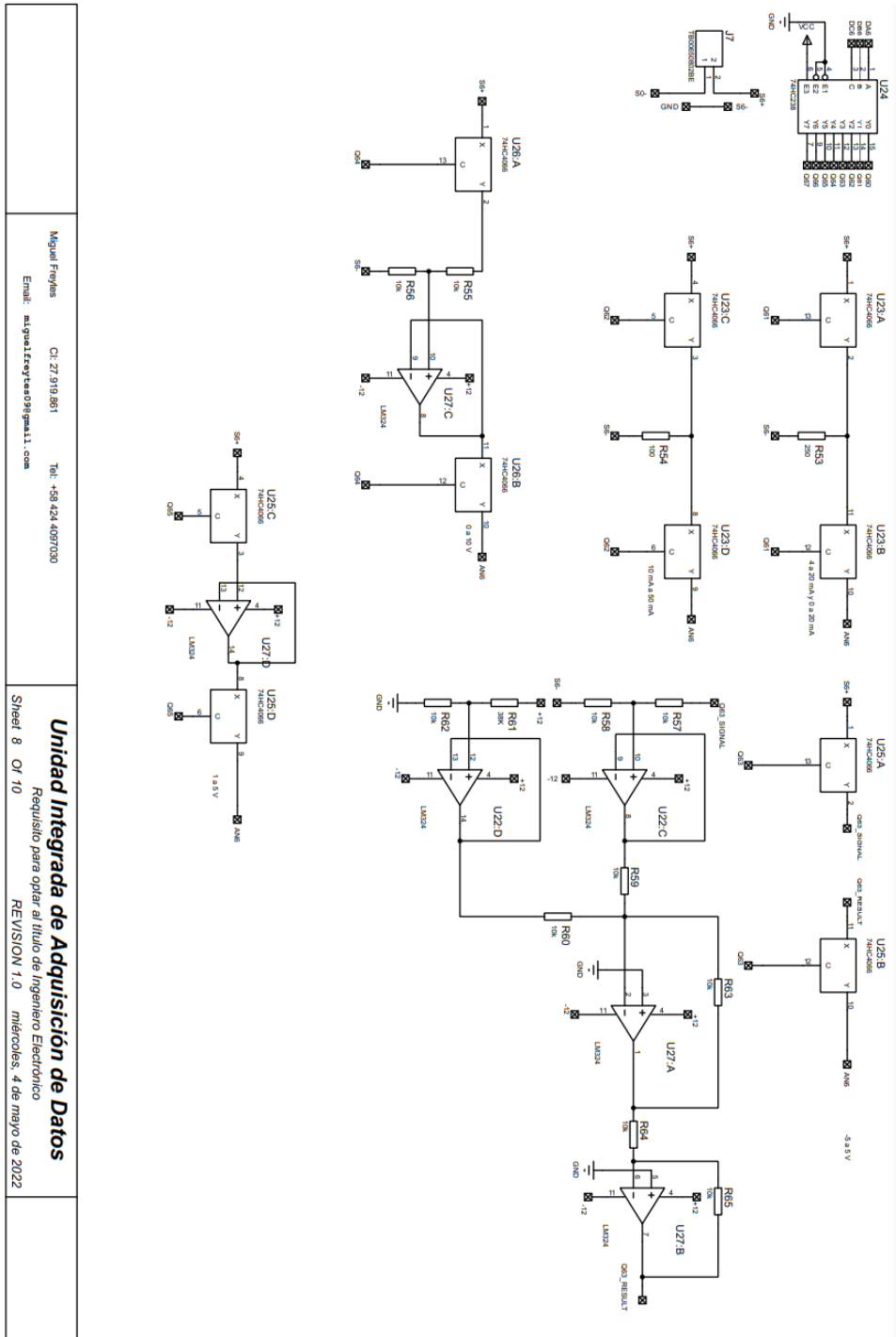


Figura 45: Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 4

Fuente: Freytes (2022)



Miguel Freytes Email: miguel.freytes@99@gmail.com	Cí: 27.919.881 Tel: +58.424.4097030	Unidad Integrada de Adquisición de Datos Requisito para optar al título de Ingeniero Electrónico REVISION 1.0 microles, 4 de mayo de 2022	
--	--	---	--

Figura 47: Diagrama Esquemático de la UIAD; Acondicionamientos de la Entrada 6

Fuente: Freytes (2022)

	<p>Miguel Freytes CI: 27.919.881 Tel: +58 424 4097030 Email: miguel.freytes09@gmail.com</p>	<p>Unidad Integrada de Adquisición de Datos Requisito para optar al título de Ingeniero Electrónico REVISIÓN 1.0 miércoles, 4 de mayo de 2022</p>	
--	--	---	--

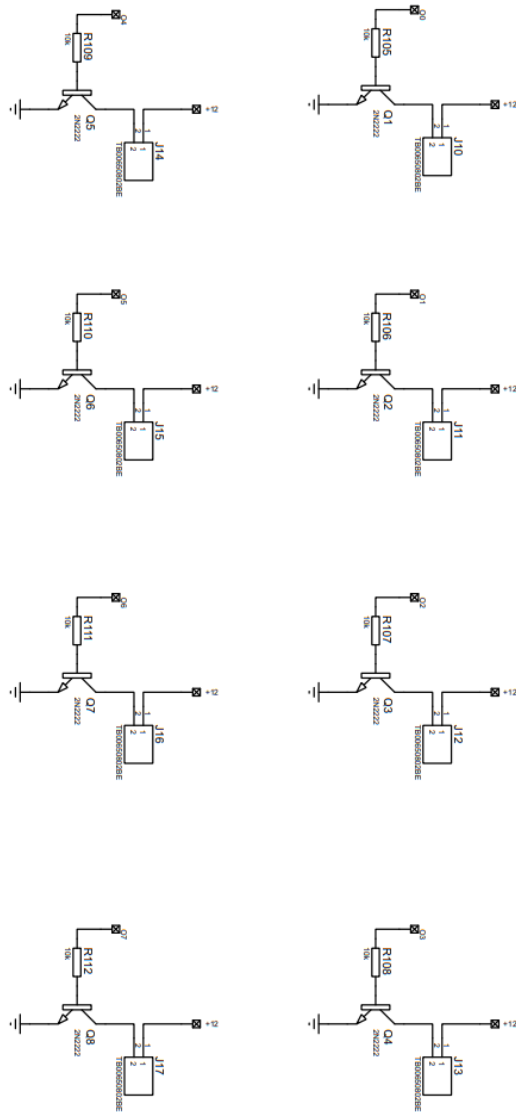


Figura 49: Diagrama Esquemático de la UIAD; Canales de Salida

Fuente: Freytes (2022)

4.2.2 Programación del Microcontrolador

La programación del microcontrolador en la UIAD, tiene la función de realizar todas las acciones de control dentro del circuito, tanto determinar el acondicionamiento específico para cada canal de entrada y leer las señales acondicionadas mediante el ADC como controlar los canales de salida a través de los pines PWM. Sin embargo, todas las acciones de control de la UIAD no son posibles sin la influencia de la comunicación con la HMI, en ella se realizan los accionamientos de los diferentes comandos que manipulan el circuito a través de configuraciones establecidas por el usuario, para realizar dicha comunicación primeramente fue necesario configurar los parámetros básicos del Arduino, inicialización de las variables primarias, creación de las constantes y configuración de los pines empleados como salidas o como entradas. Una vez realizado esto, se procedió a programar el algoritmo de comunicación Serial entre la HMI y la UIAD, para ello se recurrió a la documentación de la librería Serial del Arduino, donde se especificaba el método de lectura de datos a través de la comunicación Serial, de tal forma que se desarrolló el siguiente algoritmo mostrado en el diagrama de flujo.

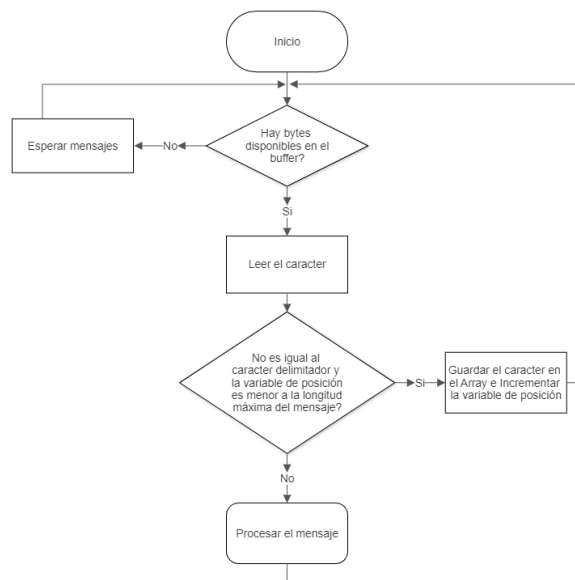


Figura 50: Diagrama de flujo del algoritmo de comunicación

Fuente: Freytes (2022)

Como se observa en el diagrama, la comunicación Serial mediante el Arduino posee un buffer de entrada que se va llenando a medida que se envían datos hacia el Arduino, al momento de ejecutar la función `Serial.read()`, el buffer de entrada devuelve el primer byte disponible en el buffer y lo elimina del buffer, de tal forma que cuando se leen todos los bytes del buffer el mismo se vacía, es por esto que la primera parte del algoritmo de comunicación verifica si existen bytes disponibles en el buffer serial con la función `Serial.available()`, ya que la misma devuelve el número de bytes presentes en el buffer, si este número es mayor que 0 comienza a leerse el mensaje de lo contrario se continúa verificando hasta que se envíe un mensaje al Arduino. En caso de que exista un mensaje en el buffer serial, se procede a inicializar dos variables, la primera de ellas crea un array de caracteres con el número máximo de bytes que se definió como límite de cada mensaje, el cual se estableció en 30 bytes, mientras que la segunda variable, inicializa a 0 la posición del array de caracteres, de tal forma que se guarden los bytes desde el comienzo del array. Luego de inicializar ambas variables se procede a leer el primer byte del buffer por medio de la función `Serial.read()` y se procede a realizar una segunda comprobación, ya que los mensajes se envían de forma continua es necesario establecer un carácter delimitante entre mensajes, para ello se estableció el carácter de salto de línea o “/n”, de tal forma que se comprueba si el carácter leído no es igual al carácter delimitador y si la posición del array no supera el límite de bytes por mensaje, si ambas condiciones se cumplen, se procede a almacenar el carácter en el array e incrementar la posición de la variable guía.

Una vez leído el mensaje completo, condición que se conoce cuando el último carácter leído es el carácter delimitador, se agrega un carácter nulo al array y entonces se continúa con el procesamiento del mensaje, que, luego de haber finalizado el mismo, se reinicia la variable de posición del array y comienza nuevamente todo el ciclo.

```

void loop()
{
  while (Serial.available() > 0){

    // Inicializa las variables para el mensaje
    static char message[MAX_MESSAGE_LENGTH];
    static unsigned int message_pos = 0;
    Serial.println(message_pos);

    // Lee el siguiente byte disponible del buffer serial
    char inByte = Serial.read();

    // Si el byte no es igual al caracter de final de mensajes y la posición del caracter es menor que el tamaño maximo del mensaje entonces:
    if (inByte != '\n' && (message_pos < MAX_MESSAGE_LENGTH - 1)){
      // Se agrega el nuevo byte al mensaje
      message[message_pos] = inByte;
      message_pos++;
    }
    // Una vez se recibe el mensaje completo
    else{
      // Se agrega un caracter nulo al final del array
      message[message_pos] = '\0';

      String messageConverted = message;

      // Procesamiento del mensaje con el protocolo

      // Reiniciar la posición para el siguiente mensaje
      message_pos = 0;
    }
  }
}

```

Figura 51: Código del microcontrolador del algoritmo de comunicación

Fuente: Freytes (2022)

El procesamiento del mensaje consiste en las acciones que realiza la UIAD dependiendo del contenido del mensaje recibido, con el fin de estandarizar el intercambio de información entre la HMI y la UIAD, se desarrolló un protocolo para dicho intercambio, el cual consiste en el siguiente proceso.

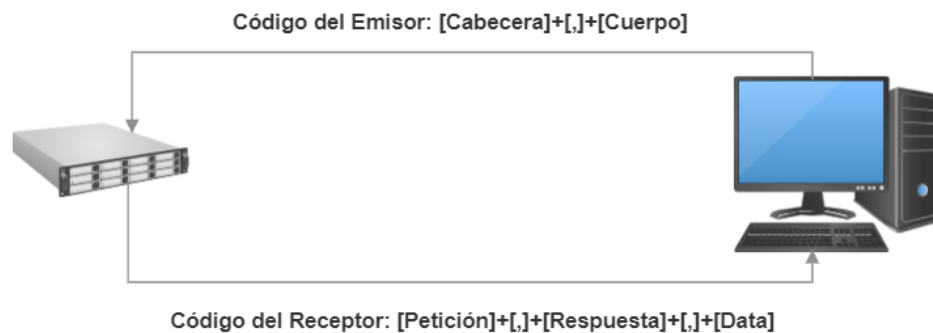


Figura 52: Diagrama del protocolo de comunicación entre la UIAD y la HMI

Fuente: Freytes (2022)

El protocolo de comunicación diseñado para la propuesta es bastante sencillo y consta de dos tipos mensajes, el primero de ellos es enviado por la HMI hacia la UIAD y está conformado por dos partes separadas por una coma, la cabecera es la que contiene el código de acción y dependiendo de él la UIAD realiza una labor u otra, mientras que el cuerpo almacena la información necesaria para llevar a cabo esa acción, de tal forma que el procesamiento del mensaje consiste en separar ambas partes del mensaje, leer el código de acción enviado y a partir de él emplear la información contenida en el cuerpo para realizar una determinada labor. Luego de realizada la acción determinada, la UIAD se encarga de codificar el segundo tipo de mensaje de vuelta hacia la HMI, el cual está conformado por tres partes separadas por comas, la petición, la respuesta y la data, el primer elemento devuelve el código de acción enviado en primer lugar, esto con la finalidad de que la HMI identifique el mensaje al que se le está dando respuesta, el segundo elemento consiste en un código que determina si se pudo completar la labor o no, mientras que el último elemento contiene la información recabada luego de realizar la acción, si la acción es una lectura del canal o una verificación de un sensor, es en este elemento que se almacena la lectura. La siguiente figura muestra los diferentes códigos del protocolo, su función y posibles respuestas.

Cuadro 3: Códigos de acción del protocolo de comunicación diseñado

Código	Función	Posibles Respuestas	Información Recibida
00	Verificación de Conexión	Exitosa, Puerto Ocupado o sin respuesta	-
01	Verificación de un Canal de Entrada o Salida	Exitosa o Problemas con el Canal	Lectura actual del puerto o Salida establecida o ninguna respuesta
02	Lectura de Canales	Exitosa o Errónea	Valores de las lecturas o

	Especificados		ninguna respuesta
03	Activación de una salida	Exitosa o Errónea	-

Fuente: Freytes (2022)

Una vez recibido el mensaje, el microcontrolador debe dividir en dos partes el mensaje según la coma, para ello se emplea la librería `StringSplitter` que dota a `c++` de esa funcionalidad, dicha librería separa en dos elementos el mensaje y posteriormente se crean dos variables de tipo `String` para almacenar tanto la cabecera como el cuerpo del mismo, la siguiente figura muestra el mencionado proceso.

```
String messageConverted = message;

StringSplitter *splitter = new StringSplitter(messageConverted, ',', 2);
String codigo = splitter->getItemAtIndex(0);
String data = splitter->getItemAtIndex(1);
```

Figura 53: Separación del mensaje recibido según el delimitador

Fuente: Freytes (2022)

Luego de separados ambas partes el microcontrolador procede a verificar el código recibido y a realizar una acción dependiendo del mismo, por ejemplo, el código 00 del protocolo verifica la conexión entre la UIAD y la HMI, ya que el mismo solo puede generar una respuesta exitosa si la conexión es correcta, entonces si el mensaje recibido por el microcontrolador es este, directamente codifica el mensaje de respuesta con el código 200, indicando que la conexión fue realizada con éxito. Si el puerto está ocupado se verifica directamente desde el HMI, al igual que si no hay respuesta indica que la conexión no fue exitosa lanzando el error directamente en la interfaz gráfica.

```

if (codigo == "00")
{
    Serial.print(message);
    Serial.print(DELIMITER);
    Serial.print(SUCCESS);
    Serial.print(DELIMITER);
    Serial.println("00");
}

```

Figura 54: Respuesta del mensaje en caso de que el código sea 00

Fuente: Freytes (2022)

Si el código recibido es 01, entonces el microcontrolador procede a decodificar el cuerpo del mensaje, en el cual se envía el canal y un conjunto de datos dependiendo si el tipo de canal es una entrada o una salida. Si el canal a verificar es una entrada, se envía dicho canal y el tipo de señal que se va a verificar, ambos datos separados por el carácter “&”, de tal forma que el Arduino separa los datos y realiza una acción dependiendo del tipo de señal del canal verificado, si la señal estandarizada posee un rango para la detección de errores, como es el caso de la señal de 4 mA a 20 mA, se verifica que la lectura del ADC esté por encima de dicho rango, si la afirmación es correcta se envía el código 200 en la respuesta de la UIAD y en la data del mensaje se adiciona el valor de la lectura, caso contrario se envía un código de error 400 y la data del mensaje de respuesta queda vacía.

```

}else if (codigo == "01"){

    StringSplitter *dataSplitter = new StringSplitter(data, '&', 3);
    String io = dataSplitter->getItemAtIndex(0);

    // Verificar si es una entrada o una salida
    if (String(io[0]) == "I"){
        // Entrada
        channel(String(io[1]));
        String signal = dataSplitter->getItemAtIndex(1);
    }
}

```

```

//Señal de 4 a 20 mA
if (signal == "0"){

    digitalWrite(digitalOutput0, HIGH);
    digitalWrite(digitalOutput1, LOW);
    digitalWrite(digitalOutput2, LOW);
    delay(10);
    if(analogRead(analogInput)>195){
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+"signal+"&");
        Serial.println(analogRead(analogInput));
    }else{
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(400);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+"signal+"&");
        Serial.println("00");
    }
}

//Señal de 0 a 20 mA
}else if(signal == "1"){
    digitalWrite(digitalOutput0, HIGH);
    digitalWrite(digitalOutput1, LOW);
    digitalWrite(digitalOutput2, LOW);
    delay(10);
    Serial.print(codigo);
    Serial.print(DELIMITER);
    Serial.print(SUCCESS);
    Serial.print(DELIMITER);
    Serial.print(io+"&"+"signal+"&");
    Serial.println(analogRead(analogInput));
}

```

```

//Señal de 10 a 50 mA
}else if(signal == "2"){
    digitalWrite(digitalOutput0, LOW);
    digitalWrite(digitalOutput1, HIGH);
    digitalWrite(digitalOutput2, LOW);
    delay(10);
    if(analogRead(analogInput)>195){
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+"signal+"&");
        Serial.println(analogRead(analogInput));
    }else{
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(400);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+"signal+"&");
        Serial.println("00");
    }
}

//Señal de -5V a 5V
}else if (signal == "3"){
    digitalWrite(digitalOutput0, HIGH);
    digitalWrite(digitalOutput1, HIGH);
    digitalWrite(digitalOutput2, LOW);
    delay(10);
    Serial.print(codigo);
    Serial.print(DELIMITER);
    Serial.print(200);
    Serial.print(DELIMITER);
    Serial.print(io+"&"+"signal+"&");
    Serial.println(analogRead(analogInput));
}

```

```

// Señal de 0 a 10V
}else if(signal == "4"){
    digitalWrite(digitalOutput0, LOW);
    digitalWrite(digitalOutput1, LOW);
    digitalWrite(digitalOutput2, HIGH);
    delay(10);
    Serial.print(codigo);
    Serial.print(DELIMITER);
    Serial.print(200);
    Serial.print(DELIMITER);
    Serial.print(io+"&"+signal+"&");
    Serial.println(analogRead(analogInput));

// Señal de 1V a 5V
}else if(signal == "5"){
    digitalWrite(digitalOutput0, HIGH);
    digitalWrite(digitalOutput1, LOW);
    digitalWrite(digitalOutput2, HIGH);
    delay(10);
    if(analogRead(analogInput)>195){
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+signal+"&");
        Serial.println(analogRead(analogInput));
    }else{
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(400);
        Serial.print(DELIMITER);
        Serial.print(io+"&"+signal+"&");
        Serial.println("00");
    }
}
}
}

```

Figura 55: Respuesta del mensaje en caso de que el código sea 01 y el canal sea una entrada

Fuente: Freytes (2022)

Al inicio de esta sección del programa, se puede observar una función denominada como “channels”, la cual recibe como parámetro a un String y realiza la labor de establecer los pines a modificar, de tal forma que, si se requiere emplear los pines de control de un determinado canal, esta función guarda dichos pines dentro de diversas variables dependiendo del canal que se pase como parámetro, a continuación, se muestra dicha función.

```
void channel(String in)
{
    if(in=="0"){
        digitalOutput0 = 22;
        digitalOutput1 = 23;
        digitalOutput2 = 24;
        analogInput = 54;
    }else if(in=="1"){
        digitalOutput0 = 25;
        digitalOutput1 = 26;
        digitalOutput2 = 27;
        analogInput = 55;
    }else if(in=="2"){
        digitalOutput0 = 28;
        digitalOutput1 = 29;
        digitalOutput2 = 30;
        analogInput = 56;
    }else if(in=="3"){
        digitalOutput0 = 31;
        digitalOutput1 = 32;
        digitalOutput2 = 33;
        analogInput = 57;
    }else if(in=="4"){
        digitalOutput0 = 34;
        digitalOutput1 = 35;
        digitalOutput2 = 36;
        analogInput = 58;
    }else if(in=="5"){
        digitalOutput0 = 37;
        digitalOutput1 = 38;
        digitalOutput2 = 39;
        analogInput = 59;
    }
}
```

```

    }else if(in=="6"){
        digitalOutput0 = 40;
        digitalOutput1 = 41;
        digitalOutput2 = 42;
        analogInput = 60;
    }else if(in=="7"){
        digitalOutput0 = 43;
        digitalOutput1 = 44;
        digitalOutput2 = 45;
        analogInput = 61;
    }
}
}

```

Figura 56: Función que define los pines a emplear dependiendo del canal pasado como parámetro

Fuente: Freytes (2022)

Por otra parte, si el canal a verificar es una salida, se generan dos posibles situaciones, la primera de ellas considera si la salida es de tipo todo/nada, de tal forma que solo se envía dicho canal y un “0” representando este tipo de salida, una vez es recibido el código por el Arduino, se coloca la salida del canal a verificar en un “1” lógico y se envía el mensaje de respuesta, el cual contiene el código “01”, una respuesta positiva y un “1” lógico informando que la salida se activó correctamente. En cuanto a la segunda situación, ella considera el caso de que la salida sea de tipo PWM, para esta condición se requiere un parámetro adicional en el cuerpo del mensaje, de tal forma que la estructura del mismo estará conformado por tres parámetros separados por el carácter “&”, el primero es el canal a verificar, el segundo el tipo de salida del mismo, el cual va a ser igual a “1” y el último parámetro contiene un dato numérico que permitirá escoger la frecuencia a la que operará el PWM, pudiendo escoger un valor entre el “0” y el “5”. Aquí el código simplemente realiza la separación de los datos contenidos en la estructura del cuerpo y ejecuta la función “frequency” que le indica al Arduino que frecuencia debe tener el pin PWM, posteriormente escribe en el pin analógico una señal PWM con un ciclo de trabajo del

50% con la finalidad de verificar que todo está funcionando correctamente, para posteriormente enviar el mismo mensaje de respuesta que en la situación anterior con la diferencia que aquí envía el valor escrito en el pin PWM, en este caso el número 127, correspondiente a la mitad de 255, valor máximo que se puede escribir en el pin analógico para un ciclo de trabajo del 100%.

```

    }else{
        //Salida
        String tipo = dataSplitter->getItemAtIndex(1);
        control(String(io[1]));

        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);

        if(tipo=="0"){
            digitalWrite(controlOutput, HIGH);
            Serial.println(io+"&"+tipo+"&1");
        }else{
            String freq = dataSplitter->getItemAtIndex(2);
            frequency(String(io[1]), freq);
            analogWrite(controlOutput, 127);
            Serial.println(io+"&"+tipo+"&127");
        }
    }
}

```

Figura 57: Respuesta del mensaje en caso de que el código sea 01 y el canal sea una salida

Fuente: Freytes (2022)

```

void frequency(String channel, String freq){
    if(channel=="0" || channel == "1" || channel == "3"){
        if(freq=="0"){
            TCCR3B = TCCR3B & B11111000 | B00000010; // Establecer un divisor al timer 3 de 8 para una frecuencia de 3921.16 Hz
        }else if(freq == "1"){
            TCCR3B = TCCR3B & B11111000 | B00000011; // Establecer un divisor al timer 3 de 32 para una frecuencia de 980.39 Hz
        }else if(freq == "2"){
            TCCR3B = TCCR3B & B11111000 | B00000100; // Establecer un divisor al timer 3 de 64 para una frecuencia de 490.20 Hz
        }else if(freq == "3"){
            TCCR3B = TCCR3B & B11111000 | B00000101; // Establecer un divisor al timer 3 de 128 para una frecuencia de 245.10 Hz
        }else if(freq == "4"){
            TCCR3B = TCCR3B & B11111000 | B00000110; // Establecer un divisor al timer 3 de 256 para una frecuencia de 122.55 Hz
        }else if(freq == "5"){
            TCCR3B = TCCR3B & B11111000 | B00000111; // Establecer un divisor al timer 3 de 1024 para una frecuencia de 30.64 Hz
        }
    }

    if(channel == "2"){
        if(freq=="0"){
            TCCR0B = TCCR0B & B11111000 | B00000010; // Establecer un divisor al timer 0 de 8 para una frecuencia de 7812.50 Hz
        }else if(freq == "1"){
            TCCR0B = TCCR0B & B11111000 | B00000011; // Establecer un divisor al timer 0 de 64 para una frecuencia de 976.56 Hz
        }else if(freq == "2"){
            TCCR0B = TCCR0B & B11111000 | B00000100; // Establecer un divisor al timer 0 de 256 para una frecuencia de 244.14 Hz
        }else if(freq == "3"){
            TCCR0B = TCCR0B & B11111000 | B00000101; // Establecer un divisor al timer 0 de 1024 para una frecuencia de 61.04 Hz
        }
    }
}

```

```

if(channel=="4" || channel == "5" || channel == "6"){
  if(freq=="0"){
    TCCR4B = TCCR4B & B11111000 | B00000010; // Establecer un divisor al timer 4 de 8 para una frecuencia de 3921.16 Hz
  }else if(freq == "1"){
    TCCR4B = TCCR4B & B11111000 | B00000011; // Establecer un divisor al timer 4 de 32 para una frecuencia de 980.39 Hz
  }else if(freq == "2"){
    TCCR4B = TCCR4B & B11111000 | B00000100; // Establecer un divisor al timer 4 de 64 para una frecuencia de 490.20 Hz
  }else if(freq == "3"){
    TCCR4B = TCCR4B & B11111000 | B00000101; // Establecer un divisor al timer 4 de 128 para una frecuencia de 245.10 Hz
  }else if(freq == "4"){
    TCCR4B = TCCR4B & B11111000 | B00000110; // Establecer un divisor al timer 4 de 256 para una frecuencia de 122.55 Hz
  }else if(freq == "5"){
    TCCR4B = TCCR4B & B11111000 | B00000111; // Establecer un divisor al timer 4 de 1024 para una frecuencia de 30.64 Hz
  }
}

if(channel == "7"){
  if(freq=="0"){
    TCCR2B = TCCR2B & B11111000 | B00000010; // Establecer un divisor al timer 2 de 8 para una frecuencia de 3921.16 Hz
  }else if(freq == "1"){
    TCCR2B = TCCR2B & B11111000 | B00000011; // Establecer un divisor al timer 2 de 32 para una frecuencia de 980.39 Hz
  }else if(freq == "2"){
    TCCR2B = TCCR2B & B11111000 | B00000100; // Establecer un divisor al timer 2 de 64 para una frecuencia de 490.20 Hz
  }else if(freq == "3"){
    TCCR2B = TCCR2B & B11111000 | B00000101; // Establecer un divisor al timer 2 de 128 para una frecuencia de 245.10 Hz
  }else if(freq == "4"){
    TCCR2B = TCCR2B & B11111000 | B00000110; // Establecer un divisor al timer 2 de 256 para una frecuencia de 122.55 Hz
  }else if(freq == "5"){
    TCCR2B = TCCR2B & B11111000 | B00000111; // Establecer un divisor al timer 2 de 1024 para una frecuencia de 30.64 Hz
  }
}
}
}

```

Figura 58: Función que define la frecuencia de salida del PWM según los parámetros establecidos

Fuente: Freytes (2022)

La función “frequency” de la figura 55 recibe como parámetro el canal y un valor numérico que representa la frecuencia deseada, la finalidad de esta función es la de cambiar la frecuencia de los temporizadores o timer que posee el Arduino, para ello se modifican los registros TCCRnB, para los cuales, si son modificados los últimos tres bits de dicho registro, es posible establecer un preescalador que divide la frecuencia del reloj del Arduino, ajustando la frecuencia de dichos timer. Los timer tienen diversas funciones dentro del microcontrolador, sin embargo, la más importante de ellas es la de establecer interrupciones en un determinado momento, el cual se determina cuando el temporizador alcanza un valor dado, las señales PWM de salida del Arduino son simplemente pulsos enviados durante las interrupciones de cada uno de los timer, variando la frecuencia de los temporizadores también varía la frecuencia a la que se realizan las interrupciones, modificando de igual forma la frecuencia a la que se envían los pulsos.

Es importante destacar que no es posible establecer una frecuencia independiente para cada uno de los pines PWM del Arduino, ya que el microcontrolador solo posee seis temporizadores para los quince pines PWM del mismo, de tal forma que es necesario compartir un temporizador por cada dos o tres pines PWM. En la propuesta diseñada la frecuencia de salida es compartida entre los canales “O0”, “O1” y “O3” ya que comparten el temporizador tres, al igual que los canales “O4”, “O5” y “O6” que comparten el temporizador cuatro, tanto los canales “O2” como “O7” comparten igualmente un temporizador con otros pines del Arduino, sin embargo, ya que esos pines no están en uso, ambos canales poseen una frecuencia independiente de salida. El programa referente a la función “frequency” modifica los registros del timer según la frecuencia deseada y el canal empleado, ya que el timer del canal “O2” es de 16 bits, el mismo posee una mayor frecuencia de salida, pudiendo ser variada desde los 7 KHz hasta los 60 Hz, mientras que la frecuencia del resto de canales varía entre 3.9 KHz y 30 Hz.

En cuanto al código “02” del protocolo, el mismo define el proceso de lectura de los canales especificados en el cuerpo del mensaje, de modo que a dichos canales se les realiza la medición mediante el ADC y posteriormente se envía la respuesta con todos los valores de las lecturas. Es importante destacar que dicho código solo es posible activarlo desde la HMI una vez se hayan realizado todas las comprobaciones de los sensores de tal forma que el ADC no realiza mediciones hasta no haber verificado los mismos. La estructura del cuerpo de este mensaje desde la HMI está conformada por el número del canal seguido por un “1” o un “0”, indicando si el canal debe ser leído o no respectivamente, repitiéndose esta nomenclatura para todos los canales, de modo que el cuerpo del mensaje será una cadena de ocho pares de dígitos separados por el símbolo “&”. La sección del programa que procesa este código del protocolo se muestra en la siguiente figura.

```

else if(codigo == "02"){
    Serial.print(codigo);
    Serial.print(DELIMITER);
    Serial.print(SUCCESS);
    Serial.print(DELIMITER);

    StringSplitter *dataSplitter = new StringSplitter(data, '&', 8);

    for (int i = 0; i < 8; i++){
        String io = dataSplitter->getItemAtIndex(i);
        if(String(io[1])=="1"){
            if(i!=7){
                channel(String(io[0]));
                Serial.print(analogRead(analogInput));
                Serial.print("&");
            }else{
                channel(String(io[0]));
                Serial.println(analogRead(analogInput));
            }
        }else{
            if(i!=7){
                Serial.print("X&");
            }else{
                Serial.println("X");
            }
        }
    }
}
}

```

Figura 59: Respuesta del mensaje en caso de que el código sea 02

Fuente: Freytes (2022)

Como se observa en la figura 56, primeramente, se procede a separar el cuerpo del mensaje según el delimitador “&”, de tal forma que se obtienen 8 cadenas de caracteres distintas, una para cada uno de los canales de entrada, a las cuales se les aplicará el proceso de lectura siguiendo el algoritmo descrito en el diagrama de flujo de la figura 60, para cada uno de las cadenas separadas se debe obtener la bandera de lectura para determinar si el canal debe ser leído o no, de tal forma que se verifica el segundo carácter de la cadena determinado para ello, si la bandera es igual a “1” se realiza la lectura del canal y se adiciona la lectura al mensaje de respuesta, de lo contrario se van a adicionar caracteres nulos a la misma, representados por el carácter

“X”, ya en la HMI cuando se reciba el mensaje, solo serán mostrados los valores con una lectura válida dejando de lado los valores con caracteres nulos.

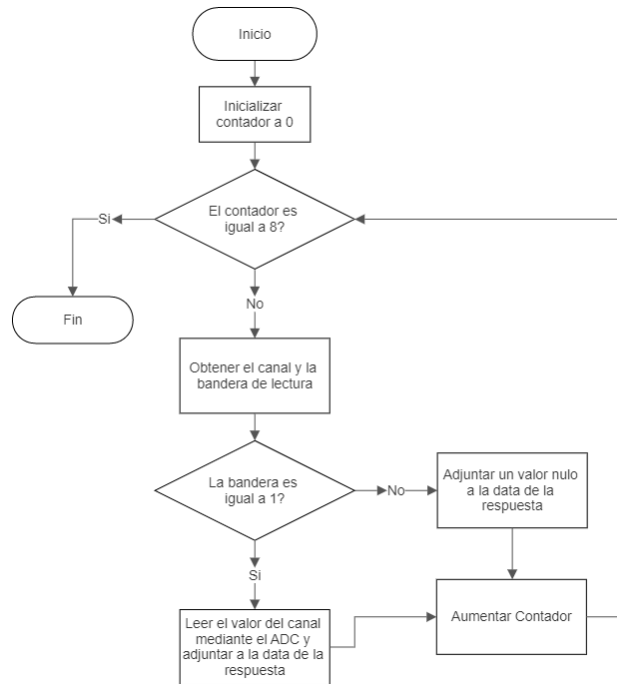


Figura 60: Diagrama de flujo del algoritmo para leer los canales especificados

Fuente: Freytes (2022)

Finalmente, se realizó la sección del programa para procesar el código “03” del protocolo de comunicación, cuya función es la de activar o desactivar los canales de salida según los parámetros establecidos en el cuerpo del mensaje. La estructura de dichos parámetros está conformada por tres valores, el primero de ellos establece el tipo de salida deseada, ya sea todo/nada o de tipo PWM, en cuanto al segundo valor, el mismo contiene el canal de salida al que se le será aplicado el control, mientras que el ultimo posee el valor a establecer en la salida, si la salida es de tipo todo/nada, este valor contendrá un “0” o un “1”, en cambio, si la salida es de tipo PWM, este valor puede variar entre 0 y 255. A continuación, se muestra la figura que contiene esta sección del código, la misma realiza el proceso de separación del cuerpo del mensaje

en los tres valores mencionados anteriormente y según el tipo de señal de salida que se desee, realiza una escritura analógica o digital en el canal determinado.

```

}else if(codigo=="03"){
    StringSplitter *dataSplitter = new StringSplitter(data, '&', 3);
    String tipo = dataSplitter->getItemAtIndex(0); //Determina si la señal es de tipo ON/OFF o PWM
    String canal = dataSplitter->getItemAtIndex(1); //Determina el canal a controlar
    String valor = dataSplitter->getItemAtIndex(2); //Determina el valor a establecer en la salida

    control(String(canal[1]));

    if(tipo=="0"){
        digitalWrite(controlOutput,valor.toInt());
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);
        Serial.println(valor);
    }else{
        analogWrite(controlOutput,valor.toInt());
        Serial.print(codigo);
        Serial.print(DELIMITER);
        Serial.print(SUCCESS);
        Serial.print(DELIMITER);
        Serial.println(valor);
    }
}
}

```

Figura 61: Respuesta del mensaje en caso de que el código sea 03

Fuente: Freytes (2022)

Tanto esta sección como la del código “01” de salida, contienen una función definida para el establecimiento del pin a controlar dependiendo del canal escogido, dicha función se denomina como “control” y se muestra en la figura 57.

```

void control(String out){
    if(out=="0"){
        controlOutput = 2;
    }else if(out == "1"){
        controlOutput = 3;
    }else if(out == "2"){
        controlOutput = 4;
    }else if(out == "3"){
        controlOutput = 5;
    }else if(out == "4"){
        controlOutput = 6;
    }else if(out == "5"){
        controlOutput = 7;
    }else if(out == "6"){
        controlOutput = 8;
    }else if(out == "7"){
        controlOutput = 9;
    }
}

```

Figura 62: Función que define el pin a controlar dependiendo del parámetro pasado

Fuente: Freytes (2022)

4.3 Fase III. Desarrollo de la Interfaz Gráfica

A lo largo de esta fase se procedió a desarrollar la interfaz gráfica que iba a realizar las acciones de control comunicándose con la UIAD, igualmente, esta fase sirvió como prueba piloto de la tarjeta de adquisición de datos en funcionamiento, ya que en la misma se realizaron las simulaciones del circuito electrónico mediante Proteus, el cual, a través de un puerto serial virtual brindado por el mismo simulador, permitía la representación de la comunicación entre la HMI y la UIAD, probando así la efectividad de los acondicionamientos y el sistema diseñado, así como el protocolo de comunicación realizado.

4.3.1 Programación de la HMI

Inicialmente, se realizó el diseño y la programación de la HMI, para ello se emplearon los lineamientos del “UX Design” el cual es una metodología para el diseño de proyectos digitales eficientes basado en la experiencia del usuario, Aguirre. Et al (2020) en su publicación para la revista de divulgación “Espacios”, mencionan que existen dos factores que intervienen en la experiencia del usuario, el primero de ellos es el principio de usabilidad, el cual se trata de un atributo cualitativo que determina que tan fácil es utilizar un producto para determinados usuarios y por otra parte se halla el segundo factor, el cual es el principio de utilidad, el mismo es el principio por el cual se aprueba o desaprueba una determinada acción según la tendencia de aumentar o disminuir la felicidad y el interés del usuario, de tal forma que una experiencia de uso positiva está directamente arraigada al hecho de que los proyectos digitales sean sencillos de utilizar y se eliminen todos los factores que puedan generar emociones negativas en el usuario.

Para ello los autores, recomiendan un desarrollo de productos digitales basado en cuatro etapas, el aterrizaje del problema, la investigación del contexto, el diseño y el testeo, la primera de ellas se trata de la organización y la definición del problema, análisis de las tareas que debe realizar el usuario y como va a estar organizada la

información, la segunda etapa se trata de entender el contexto en el que se va a emplear dicho producto digital, las necesidades del usuario y su conocimiento del manejo de la tecnología, luego, se pasa a una tercera etapa que consiste en la realización de bosquejos y prototipos del producto culminado para finalizar con la fase de testeo donde se prueba directamente el producto con usuarios de forma que puedan brindar sus aportes para la mejora del resultado.

Tomando en cuenta estas recomendaciones, se procedió con el desarrollo de la interfaz gráfica, siempre teniendo presente al usuario final como referencia para la toma de decisiones del diseño de la misma. El lenguaje de programación empleado para el desarrollo de dicha interfaz es el lenguaje orientado a objetos Java, el cual por su filosofía de universalidad, permite ser ejecutado en múltiples plataformas sin la necesidad de realizar modificaciones en el código, de tal manera que una vez estuvo finalizado el desarrollo, la HMI podía ser ejecutada en múltiples sistemas operativos sin hacer adaptaciones importantes, que, adicionado al hecho de que el lenguaje posee librerías integradas para el desarrollo de interfaces, lo hicieron la opción predilecta para el desenvolvimiento de esta investigación. La librería empleada para el desarrollo fue la librería Swing de Java, que ofrece un conjunto de herramientas que facilitan el diseño, además de brindar un aspecto moderno y minimalista.

Los primeros elementos diseñados de la interfaz, fue todo lo referente al aspecto físico, para posteriormente iniciar con la funcionalidad de cada uno de los botones, para ello se escogió una estructura basada en cuatro paneles, un panel superior en el que se situarían los botones del menú de acción, un panel central en el que se situaría una tabla con los ocho canales de entradas y salidas que interactuarían con el resto de paneles, un panel lateral donde habría información más detallada al respecto de cada canal y un panel inferior o pie en donde se situaría el estado actual de conexión con la UIAD, en la siguiente figura se muestra el resultado final del aspecto físico de la HMI.

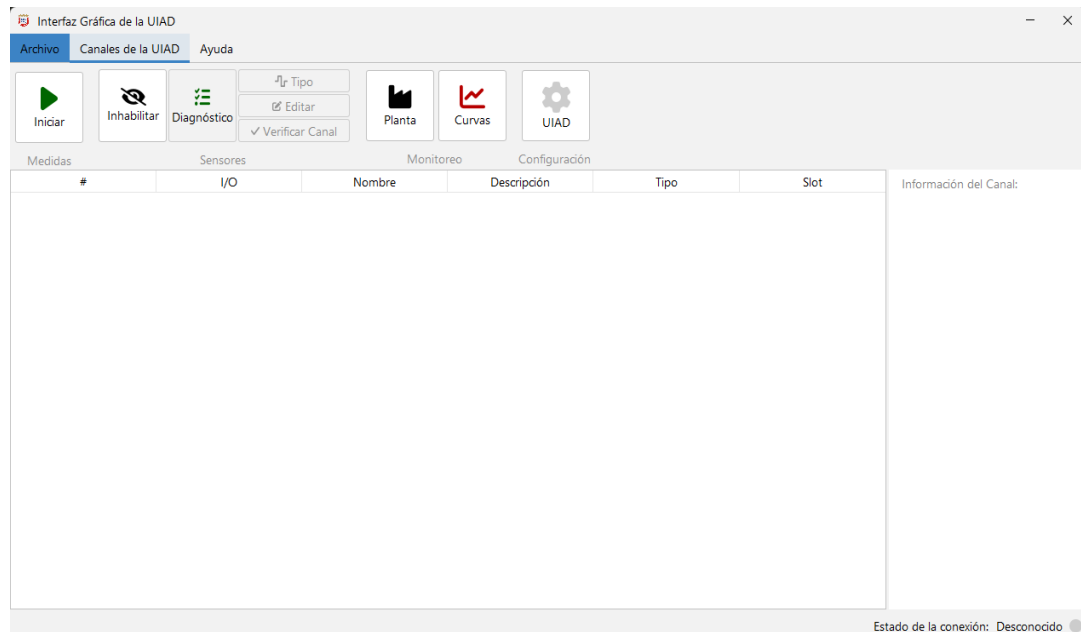


Figura 63: Estructura física de la HMI

Fuente: Freytes (2022)

Como se observa en la figura, el panel superior cuenta con un sistema de pestañas para separar los diferentes elementos de la interfaz, cuenta con una pestaña de “Archivo” en la que se agruparon todos los botones referentes al archivo de configuración de la interfaz, mientras que en una segunda pestaña denominada como “Canales de la UIAD” se situaron todos los botones de configuración y puesta en marcha de las mediciones. Finalmente se agregó una última pestaña de “Ayuda” en la que se colocaron un par de botones, uno referente al manual tanto de la interfaz gráfica como de la UIAD, así como un botón de información de la autoría del proyecto.

Luego de haber desarrollado la estructura física se procedió a programar la funcionalidad de la interfaz, comenzando con el archivo de configuración de la misma, dicho archivo tendría la labor de almacenar los datos de los canales configurados para posteriormente ser mostrados en la tabla, para ello el archivo denominado como “channels.csv” contaría con 16 registros separados por saltos de línea y cada uno de ellos tendría un conjunto de datos separados por el delimitador

“&”, el mismo estaría ubicado en la raíz de la carpeta de la HMI y podría ser modificado frecuentemente para acceder y modificar esa información. Los datos almacenados en cada registro son el id del canal, si es una entrada o una salida, el nombre y la descripción, el tipo del canal, donde se almacena la señal en el caso de las entradas y en las salidas si es ON/OFF o PWM, también se almacena el slot correspondiente en la UIAD, los valores mínimos y máximos en las entradas o la frecuencia y el set point en las salidas, la unidad del proceso físico y una bandera que indica si el canal está habilitado o no. A continuación, se muestra un archivo de configuración en blanco con todos los campos mencionados anteriormente.

1&	0&	NC&	NC&	404&	I0&	0&	0&	17&	FALSE
2&	0&	NC&	NC&	404&	I1&	0&	0&	17&	FALSE
3&	0&	NC&	NC&	404&	I2&	0&	0&	17&	FALSE
4&	0&	NC&	NC&	404&	I3&	0&	0&	17&	FALSE
5&	0&	NC&	NC&	404&	I4&	0&	0&	17&	FALSE
6&	0&	NC&	NC&	404&	I5&	0&	0&	17&	FALSE
7&	0&	NC&	NC&	404&	I6&	0&	0&	17&	FALSE
8&	0&	NC&	NC&	404&	I7&	0&	0&	17&	FALSE
9&	1&	NC&	NC&	404&	O0&	0&	0&	17&	FALSE
10&	1&	NC&	NC&	404&	O1&	0&	0&	17&	FALSE
11&	1&	NC&	NC&	404&	O2&	0&	0&	17&	FALSE
12&	1&	NC&	NC&	404&	O3&	0&	0&	17&	FALSE
13&	1&	NC&	NC&	404&	O4&	0&	0&	17&	FALSE
14&	1&	NC&	NC&	404&	O5&	0&	0&	17&	FALSE
15&	1&	NC&	NC&	404&	O6&	0&	0&	17&	FALSE
16&	1&	NC&	NC&	404&	O7&	0&	0&	17&	FALSE

Figura 64: Archivo de Configuración de la HMI

Fuente: Freytes (2022)

Este archivo de configuración sería leído al comienzo de la ejecución de la HMI, de tal forma que se cargan los datos automáticamente en la tabla una vez es abierta la interfaz, en el caso de que sea la primera vez que es abierta la misma o el archivo de configuración fue eliminado, la interfaz automáticamente crearía un nuevo archivo con los datos mostrados en la figura, las celdas con valores “NC” representan datos “No Configurados”, mientras que las celdas con datos genéricos como “404” o “17” son aquellas en las que no se ha seleccionado una opción para esos datos dentro de la HMI. Una vez diseñado el archivo de configuración se implementó la

funcionalidad de lectura del mismo para cargar los datos dentro de la tabla, la cual dependiendo de los datos leídos muestra la información deseada.

Medidas	Sensores	Monitoreo	Configuración		
#	I/O	Nombre	Descripción	Tipo	Slot
1	Entrada	NC	NC	NC	10
2	Entrada	NC	NC	NC	11
3	Entrada	NC	NC	NC	12
4	Entrada	NC	NC	NC	13
5	Entrada	NC	NC	NC	14
6	Entrada	NC	NC	NC	15
7	Entrada	NC	NC	NC	16
8	Entrada	NC	NC	NC	17
9	Salida	NC	NC	NC	O0
10	Salida	NC	NC	NC	O1
11	Salida	NC	NC	NC	O2
12	Salida	NC	NC	NC	O3
13	Salida	NC	NC	NC	O4
14	Salida	NC	NC	NC	O5
15	Salida	NC	NC	NC	O6
16	Salida	NC	NC	NC	O7

Figura 65: Tabla de Canales de la HMI

Fuente: Freytes (2022)

En la pestaña “Archivo” también se agregó funcionalidad relacionada con el archivo “channels.csv”, donde cada uno de los cuatro botones permiten realizar acciones con el mismo, ya sea cargar un archivo de configuración, crear uno nuevo, exportar el archivo actual e incluso obtener información acerca de él.



Figura 66: Pestaña “Archivo” de la HMI

Fuente: Freytes (2022)

Luego de finalizar toda la funcionalidad adyacente al archivo de configuración se avanzó con la funcionalidad de los botones de la segunda pestaña del menú, donde el primer botón programado fue el botón de “UIAD” en el que se establece el puerto serial a conectarse y una vez escogido el mismo se envía el código “00” del protocolo para comprobar la conexión, en caso de que la conexión sea exitosa se muestra un

símbolo verde en pantalla y en el panel inferior, mientras que si la conexión es errónea o el puerto se encuentra ocupado, se muestra un símbolo rojo.



Figura 67: Ventana de Configuración de la UIAD

Fuente: Freytes (2022)

Una vez finalizada la programación del botón de configuración de la UIAD se implementó funcionalidad en los botones de configuración de los canales, el botón “Habilitar”, activa o desactiva el canal seleccionado, de tal forma que el mismo no es tomado en cuenta para los diagnósticos al sistema, el botón “Tipo” despliega una ventana que según se seleccione una entrada o una salida permite configurar la señal del sensor o el tipo de salida respectivamente.



Figura 68: Ventana de Configuración del Tipo de Canal

Fuente: Freytes (2022)

Por otra parte, el botón “Editar” despliega una ventana que permite cambiar el nombre y la descripción de los canales, estableciendo una guía visual de los canales que se están configurando.

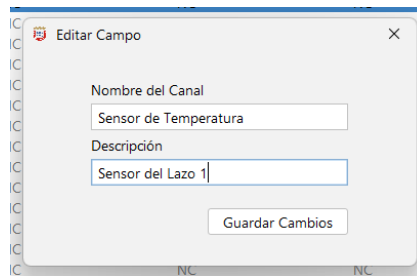


Figura 69: Ventana de Configuración de los datos del Canal

Fuente: Freytes (2022)

En cuanto al botón de “Verificar Canal”, el mismo solo puede ser presionado una vez se haya establecido conexión con la UIAD, al presionarlo muestra una ventana de confirmación con los datos del sensor y posteriormente muestra un diagrama de conexión para el tipo de canal escogido, esto con la finalidad de evitar conexiones erróneas, agregando una capa de protección al proceso. Si el usuario verifica que los datos son correctos y las conexiones fueron realizadas correctamente, la HMI envía el código “01” del protocolo a la UIAD y muestra la respuesta recibida.



Figura 70: Ventana de Confirmación de la Verificación de un Sensor

Fuente: Freytes (2022)

El botón “Diagnóstico” igualmente realiza una serie de comprobaciones, pero ya no solo al canal seleccionado, sino a todo el conjunto de canales que se encuentren habilitados al momento de iniciar el diagnóstico, igualmente este botón solo puede

ser presionado si la conexión con la UIAD se realizó de forma correcta y si todos los canales habilitados fueron configurados previamente. Una vez realizado el diagnóstico, el botón de “Iniciar” es habilitado de tal forma que ya es posible comenzar a realizar mediciones, al momento de presionar el botón, se envía desde la HMI el código del protocolo “02” y se abre la ventana de la “Planta”, pudiendo seleccionarse el tiempo de muestreo deseado, donde a cada respuesta se almacena en un array de flotantes la lectura de los canales, para posteriormente graficar los datos obtenidos en la sección de “Curvas”.

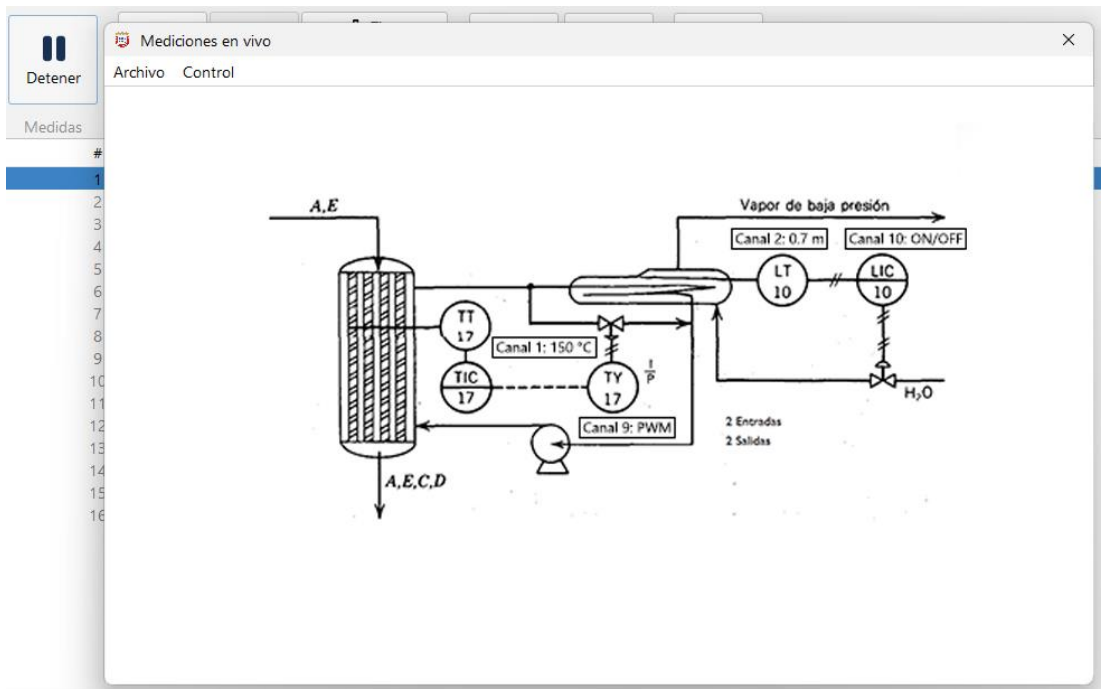


Figura 71: Ventana de Mediciones con diagrama de Planta

Fuente: Freytes (2022)

En la figura se puede observar la ventana de mediciones con el diagrama de la planta de procesos, dicho diagrama puede ser cargado mediante el menú superior y se pueden agregar los diferentes canales al diagrama para visualizar sus mediciones en vivo, adicionalmente en el menú superior también es posible configurar el control de los procesos ya sea mediante controles todo o nada o controles PID, para los cuales es

necesario establecer los valores de las constantes requeridas en el proceso y el valor del set point.

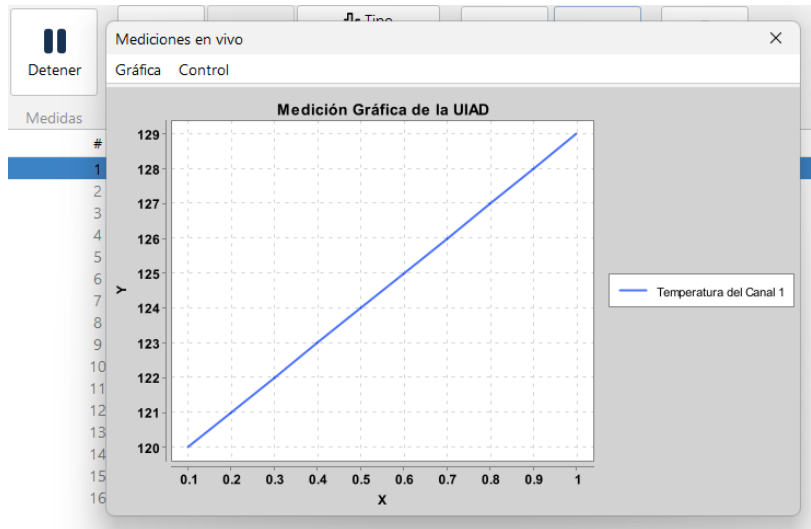


Figura 72: Ventana de Mediciones con Gráfica

Fuente: Freytes (2022)

Mientras que la ventana de curvas muestra las curvas generadas en tiempo real pudiendo ser configurada a través del menú superior al igual que en el anterior tipo de ventana de medición. Finalmente, la última sección programada fue la referente a la ventana de ayuda en la que se incluyeron los manuales de la UIAD y la HMI, los cuales a través de un sistema de archivos permiten agregar ficheros HTML, donde se redacta la información y se adicionan imágenes.



Figura 73: Manual de la UIAD y HMI integrado en la interfaz

Fuente: Freytes (2022)

4.3.2 Simulación de la Propuesta

Habiendo programado toda la interfaz gráfica, se procedió a realizar la simulación del circuito y la comunicación con la HMI, para ello se emplearon los drivers del puerto virtual disponibles en el simulador Proteus y un software de simulación de puertos virtuales denominado como “Virtual Serial Port” de la empresa Eltima Software, dicho software se encarga de crear los puertos virtuales a los que se conectará la UIAD para realizar las simulaciones. Una vez configurado Proteus para conectarse con el puerto virtual, se realizaron las primeras pruebas empleando el monitor Serial del IDE del Arduino, el cual permite enviar y recibir datos al puerto serial configurado, verificando que todo estaba funcionando correctamente.

Ya para la comunicación con la HMI, fue necesario implementar una librería en el proyecto de Java denominada como JSerialComm, la cual brinda un conjunto de atributos y parámetros para operar de forma sencilla con el puerto Serial, la misma ofrece la posibilidad de trabajar con el protocolo de forma sincrónica y asincrónica, para lo cual se decidió operar de forma asíncrona ya que procesa los mensajes apenas son recibidos sin necesidad de pausar el flujo del programa hasta que los mensajes

son recibidos. Para la operación de forma asíncrona se emplea un método denominado como “listener” u oyente, el cual está constantemente “escuchando” y esperando a recibir un mensaje a través de la comunicación Serial, cuando un mensaje es recibido el procesamiento se envía a otro hilo del procesador de tal forma que en ningún momento se detiene la ejecución del programa principal, pudiendo recibir múltiples mensajes en segundo plano.

Las pruebas realizadas fueron mayormente de los diferentes acondicionamientos de las señales en la tarjeta electrónica y su lectura a través de la HMI, al igual que la activación y desactivación de los canales de salida a través de la interfaz sin manipulación en el simulador, dando resultados positivos para cada una de las pruebas efectuadas.

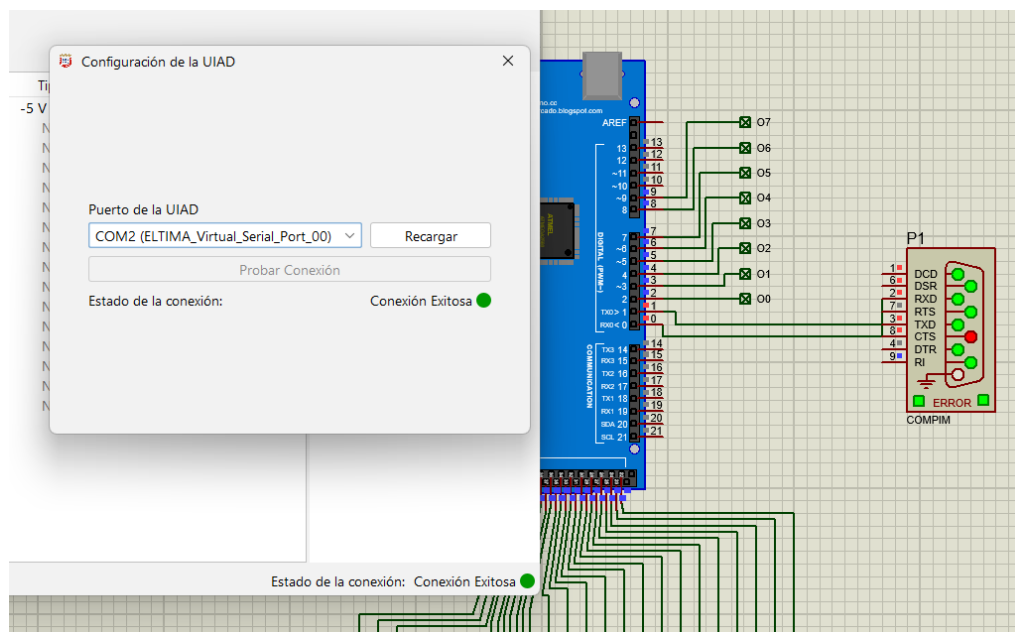


Figura 74: Conexión exitosa de la HMI con el puerto virtual de Proteus

Fuente: Freytes (2022)

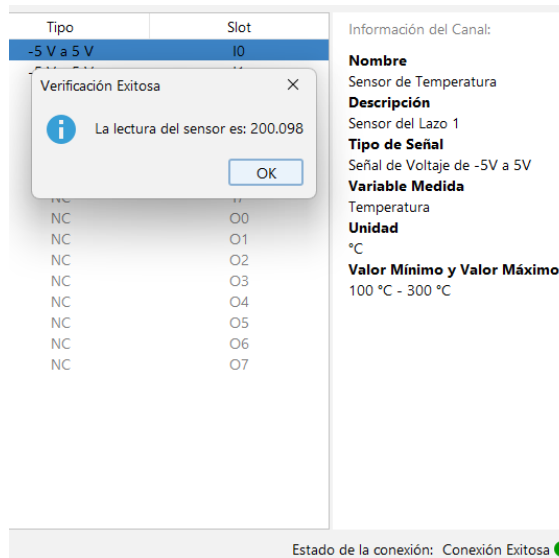


Figura 75: Verificación exitosa de un sensor con una señal de 0V a la entrada

Fuente: Freytes (2022)

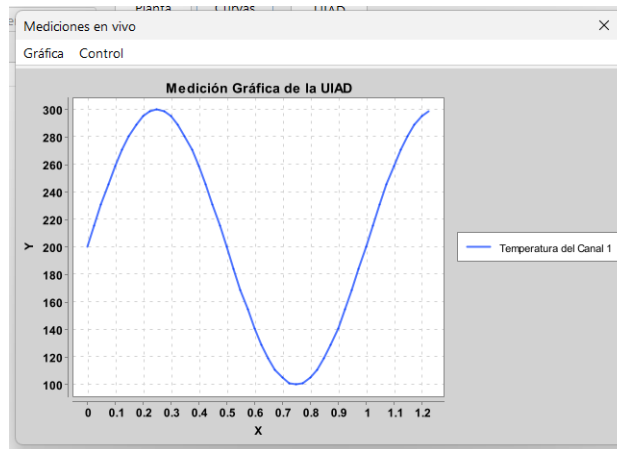


Figura 76: Curva obtenida al aplicar una señal de voltaje senoidal al Canal 1

Fuente: Freytes (2022)

4.4 Fase IV. Factibilidad

Durante esta fase se realizaron los diferentes estudios de factibilidad con el objetivo de establecer o no la viabilidad de la propuesta diseñada, para ello se consideraron tres estudios, un primer estudio de costos en el que se determinaría si el costo final del producto sería competitivo en comparación con el costo de productos comerciales, un segundo estudio de factibilidad técnica con el objetivo de determinar

si se disponían los recursos técnicos necesarios para implementar la propuesta, para finalizar con un estudio de factibilidad operativa en el que se determinaría si el recurso humano disponible sería capaz de llevar a cabo la operación de la propuesta una vez estuviese implementada.

4.4.1 Estudio de Costos

Primeramente, fue realizado el estudio de costos englobando todos los recursos que serían necesarios para la implementación, ya sean recursos humanos, materiales, tecnológicos o logísticos, de tal forma que el costo total de la propuesta determinaría si la misma representaba una alternativa viable a las tarjetas de adquisición de datos comerciales. En la siguiente tabla se agrupan todos los recursos requeridos para dicha implementación.

Cuadro 4: Recursos necesarios para la implementación de la propuesta

Recursos	Descripción	Precio Unitario	Cantidad	Total
Humano	Técnico Electrónico	3\$/h	5h	15\$
Material	Arduino Mega 2560 Rev3	42.4\$	1u	42.4\$
	Integrado LM7805N	1.19\$	1u	1.19\$
	Condensador 10 μ F	0.6\$	1u	0.6\$
	Condensador 1 μ F	0.6\$	1u	0.6\$
	Integrado 74HC4066	0.49\$	20u	9.8\$
	Integrado 74HC238	0.52\$	8u	4.16\$
	Integrado LM324N	0.49\$	12u	5.88\$
	Transistor 2N2222	1.2\$	8u	9.6\$
	Bornera de 3 Pines	0.5\$	1u	0.5\$
	Bornera de 2 Pines	0.5\$	16u	8\$
	Resistencia 250 Ω 0.5W	0.2\$	8u	1.6\$
	Resistencia 100 Ω 0.25W	0.2\$	8u	1.6\$
	Resistencia 10K Ω 0.25W	0.1\$	88u	8.8\$
	Resistencia 38K Ω 0.25W	0.2\$	8u	1.6\$

Tecnológico	Impresión de la PCB mediante una empresa especializada en el extranjero	18\$	1u	18\$
	Impresión 3D de la carcasa de almacenamiento	20\$	1u	20\$
Logístico	Envío del PCB	13\$	1u	13\$
Total				162.33\$

Fuente: Freytes (2022)

Para aportar sustento a este estudio de costos se recurrió a diversos catálogos de múltiples empresas que comercializan esta clase de productos con la finalidad de comparar sus características y su costo con la propuesta diseñada, primeramente se estudiaron los costos de los equipos ofrecidos por la empresa **Measurement Computing**, la cual posee una línea de dispositivos de adquisición de datos de bajo costo, en ella los precios oscilan entre los 99\$ y los 579\$ al momento de la redacción de esta investigación, sin embargo, los dispositivos del margen inferior poseen características por debajo de las de la propuesta diseñada, siendo el caso más notable el dispositivo de la serie USB-200 denominado como USB-201, el cual tiene un costo de 105\$ sin embargo, solo posee 8 canales de I/O y ninguna salida analógica, aunado a que el software de adquisición de datos propietario de la empresa, “DAQami”, no está incluido en la compra del dispositivo, de tal forma que se incrementa hasta 154\$ el costo del equipo y todavía se poseen prestaciones muy pobres.

DEVICE	Bus Type	Analog Inputs	Sample Rate	Resolution	Digital I/O	Counters	Analog Outputs	PRICE
LOW COST								
USB-200 Series	USB	8 SE	500 kS/s	12-bit	8	1	Up to 2	From \$105
USB-230 Series	USB	8 SE/4 DIFF	100 kS/s	16-bit	8	1	2	From \$289
MCC 118	RPi	8 SE	100 kS/s	12-bit	—	—	—	\$99
MCC 128	RPi	8 SE/4 DIFF	100 kS/s	16-bit	—	—	—	\$199
BTH-1208LS	Bluetooth	8 SE/4 DIFF	1 kS/s	12-bit	8	1	2	\$209
USB-1208LS	USB	8 SE/4 DIFF	1.2 kS/s	12-bit	16	1	2	\$140
USB-1208FS-Plus	USB	8 SE/4 DIFF	50 kS/s	12-bit	16	1	2	\$199
E-1608	Ethernet	8 SE/4 DIFF	250 kS/s	16-bit	8	1	2	\$579
USB-1608G Series	USB	16 SE/8 DIFF	500 kS/s	16-bit	8	2	Up to 2	From \$459

Figura 77: Características y Costos de los equipos DAQ de la línea “low cost” de la empresa Measurement Computing

Fuente: Measurement Computing (2022)

De la lista de la figura anterior los equipos que más se asemejan a la propuesta diseñada, son los equipos de la serie USB-1208, ya que los mismos ofrecen hasta 8 canales analógicos simples o cuatro canales diferenciales, un selector de rango de entrada configurable mediante la interfaz y solo 2 canales analógicos de salida, sin embargo, ya los equipos de esta serie inician en un costo mínimo de 140\$ y aún no incluyen el software de adquisición de datos, pudiendo alcanzar costos de hasta 189\$. El siguiente catálogo estudiado fue el de la empresa **DATAQ Instruments**, del cual se filtraron solo los dispositivos con conexión USB y con 8 canales analógicos de entrada, obteniendo los equipos mostrados en la siguiente figura. En la figura se puede observar que el dispositivo con características similares a la UIAD es el modelo DI-2108, el cual posee 8 canales analógicos de entrada, 7 canales de salida digitales, un rango de medición de $\pm 10V$, no posee salidas analógicas y el software de medición incluido es una versión de prueba.













<p>DI-2008 V, mV, Thermocouple, Pulse DAQ \$399</p>  <ul style="list-style-type: none"> 4 or 8 differential and isolated channels Voltage or TC type per channel Expandable using ChannelStretch™ 16-bit measurement resolution $\pm 10mV$ to $\pm 50V$ measurement range Rate and count inputs Includes WinDaq and .NET class <p>Go to Product Page</p>	<p>Model DI-2108 High Speed USB DAQ \$199</p>  <ul style="list-style-type: none"> 4 or 8 differential analog inputs $\pm 10V$ measurement range Up to 220 kHz throughput 16-bit measurement resolution Anti-aliasing Filter per Channel 7 Digital Ports with rate and count Records to PC or USB flash memory Includes WinDaq and .NET class <p>Go to Product Page</p>	<p>DI-4108-UJ High Speed USB DAQ \$449</p>  <ul style="list-style-type: none"> 8 differential analog inputs Expandable using ChannelStretch™ 16-bit measurement resolution $\pm 200mV$ to $\pm 10V$ measurement range Anti-aliasing Filter per Channel 160 kHz analog throughput rate Records to PC or USB flash memory Includes WinDaq and .NET class <p>Go to Product Page</p>	<p>DI-4208-UJ High Speed USB DAQ \$449</p>  <ul style="list-style-type: none"> 8 differential analog inputs Expandable using ChannelStretch™ 16-bit measurement resolution ± 2 to $\pm 100 V$ measurement range Anti-aliasing Filter per Channel 160 kHz analog throughput rate Records to PC or USB flash memory Includes WinDaq and .NET class <p>Go to Product Page</p>
<p>DI-4108-E High Speed USB/Ethernet DAQ \$499</p>  <ul style="list-style-type: none"> 8 differential analog inputs Expandable using ChannelStretch™ 16-bit measurement resolution $\pm 200mV$ to $\pm 10V$ measurement range Anti-aliasing Filter per Channel 160 kHz analog throughput rate Records to PC or USB flash memory Includes WinDaq software <p>Go to Product Page</p>	<p>DI-4208-E High Speed USB/Ethernet DAQ \$499</p>  <ul style="list-style-type: none"> 8 differential analog inputs Expandable using ChannelStretch™ 16-bit measurement resolution ± 2 to $\pm 100 V$ measurement range Anti-aliasing Filter per Channel 160 kHz analog throughput rate Records to PC or USB flash memory Includes WinDaq software <p>Go to Product Page</p>	<p>DI-2108-P High Speed USB DAQ \$449</p>  <ul style="list-style-type: none"> 4 or 8 differential analog inputs Expandable using ChannelStretch™ True 16-bit measurement resolution Programmable measurement ranges $\pm 2.5, \pm 5, \pm 10, 0-5, 0-10$ Volts 160 kHz throughput 7 Digital Ports with rate and count High Common-mode range -$\pm 200V$ Records to PC or USB thumb drive Includes WinDaqLife software <p>Go to Product Page</p>	<p>DI-710-UH USB Data Acquisition \$499</p>  <ul style="list-style-type: none"> 16 analog inputs 4800 Hz max sample rate $\pm 1.25 V$ to $\pm 10 V$ programmable ranges Measure environmental temperature & humidity with HMP60 probe Includes WinDaq software <p>Go to Product Page</p>
<p>DI-710-UL USB Data Acquisition \$499</p>  <ul style="list-style-type: none"> 16 analog inputs 4800 Hz max sample rate $\pm 10 mV$ to $\pm 10 V$ programmable ranges Better than 2 μV resolution on the 10 mV range Includes WinDaq software <p>Go to Product Page</p>	<p>DI-710-UHS Stand-alone USB Data Logger \$599</p>  <ul style="list-style-type: none"> 16 analog inputs 14.4 kHz max sample rate $\pm 1.25 V$ to $\pm 10 V$ programmable ranges Records to PC or SD memory Measure environmental temperature & humidity with HMP60 probe <p>Go to Product Page</p>	<p>DI-710-UJS Stand-alone USB Data Logger \$599</p>  <ul style="list-style-type: none"> 16 analog inputs 14.4 kHz max sample rate $\pm 10 mV$ to $\pm 10 V$ programmable ranges Records to PC or SD memory Better than 2 μV resolution on the 10 mV range <p>Go to Product Page</p>	<p>DI-7188-UJ USB Data Acquisition \$595</p>  <ul style="list-style-type: none"> 8 analog inputs 4800 Hz max sample rate Uses isolated 8B amplifier modules Makes any industrial measurement Small form factor fits into tight spaces Wide power supply range (9-36 Vdc) Includes WinDaq software <p>Go to Product Page</p>

Figura 78: Características y Costos de los equipos DAQ USB de la empresa DATAQ Instruments

Fuente: DATAQ Instruments (2022)

Finalmente, el último catálogo estudiado fue el de la empresa **LabJack**, el cual posee el dispositivo U3, que posee tanto características similares a la UIAD como un precio del mismo rango, dicho dispositivo posee conexión USB, 16 Canales I/O flexibles que pueden ser usados como entradas analógicas de 0 a 3.6V y dos salidas analógicas, el mismo tiene un costo de 155\$ con su carcasa de almacenamiento y el software de adquisición incluido.







	<p>T7 \$520/\$460 OEM High performance, USB, Ethernet, WiFi</p> <ul style="list-style-type: none"> • USB, Ethernet, and WiFi connectivity • Fully compatible with Modbus TCP • 14 analog inputs • 16+ bits of resolution • T7-Pro: Up to 22-bit resolution • $\pm 10V$, $\pm 1V$, $\pm 0.1V$, and $\pm 0.01V$ ranges • 300k samples/s max stream rate • Expand to 84 channels with Mux80 • Optional NIST traceable Calibration Certificate. • AIN-EF: Hardware scaling & processing • 2 analog outputs • 23 digital I/O • Up to 12 timer and counter features • Fun Lua scripts in hardware <p>★ ★ ★ ★ ★ See our reviews ▾</p>		<p>U6 \$400/\$365 OEM Exceptional price and performance</p> <ul style="list-style-type: none"> • USB Connectivity • 14 high quality analog inputs • 16+ bits of resolution • U6-Pro: Up to 22-bit resolution • $\pm 10V$, $\pm 1V$, $\pm 0.1V$, and $\pm 0.01V$ ranges • 50 ksamples/s max stream rate • Expand to 84 channels with Mux80 • Optional NIST traceable Calibration Certificate. • 2 analog outputs • 20 digital I/O • 4 timers and 2 counters
	<p>T4 \$245/\$200 OEM Low cost USB and Ethernet</p> <ul style="list-style-type: none"> • USB and Ethernet connectivity • Fully compatible with Modbus TCP • Up to 12 analog inputs • 12-bit resolution • 4 dedicated $\pm 10V$ analog inputs • Up to 8 0-2.5V analog inputs • 50 ksamples/s max speed • AIN-EF: Hardware scaling & processing • 2 analog outputs • Up to 16 digital I/O • Up to 10 timer and counter features • Fun Lua scripts in hardware 		<p>UE9 The original USB and Ethernet DAQ</p> <ul style="list-style-type: none"> • USB and Ethernet connectivity • 14 analog inputs • 12-16 bit resolution • UE9-Pro: Up to 20-bit resolution • $\pm 5V$ and 0-5V ranges • 50 ksamples/s max stream rate • Expand to 84 channels with Mux80 • 2 analog outputs • 23 digital I/O • 6 timers and 2 counters
	<p>U3 \$155/\$120 OEM Our most affordable DAQ</p> <ul style="list-style-type: none"> • USB Connectivity • Up to 16 analog inputs • 12-bit resolution • Some 0-2.4V range and some $\pm 10V$ range • 50 ksamples/s max stream rate • 2 analog outputs • Up to 20 digital I/O • 2 timers and 2 counters 		<p>U12 \$210/\$170 OEM The original LabJack</p> <ul style="list-style-type: none"> • Our original groundbreaking USB DAQ device • 8 analog inputs, 12 bit resolution, 1.2 ksamples/s • 2 analog outputs • 20 digital I/O • 1 counter

Figura 79: Características y Costos de los equipos DAQ USB de la LabJack

Fuente: LabJack (2022)

Cuadro 5: Cuadro Comparativo entre la UIAD y alternativas comerciales

Características	UIAD	USB-201	USB-1208LS	DI-2108	U3
Canales de Entrada	8	8 SE/4 Diff	8 SE/4 Diff	8 Diff	16 I/O
Canales de Salida	8	8 I/O	16 I/O	7 I/O	
Salidas Analógicas	8 salidas PWM	Máximo 2	Máximo 2	-	2
Resolución del ADC	10 bits	12 bits	14 bits	12 bits a 16 bits	12 bits
¿Incluye Software de Adquisición de Datos?	Si	No	No	Versión de Prueba	Si
Máximo número de muestras por segundo	100 KS/s (600 KS/s Teóricas)	500 KS/s	50 KS/s	220 KS/s	50 KS/s
Costo del Equipo	162.33\$	Desde 105\$ sin la HMI	Desde 140\$ sin la HMI	Desde 199\$	155\$

Fuente: Freytes (2022)

Como se observa en la tabla comparativa, los dispositivos estudiados se encuentran en la misma gama de precios, sin embargo, las características que ofrecen los equipos de este rango se encuentran muy limitados en diversos aspectos, el

primero de ellos es el número de salidas analógicas, ya que el número de ellas suele ser muy bajo o inexistente en algunos casos, de tal forma que solo es posible realizar controles de tipo todo o nada, impidiendo controlar con precisión ciertos procesos, el otro factor a tener en cuenta es el software de adquisición de datos, la mayoría de los dispositivos de este rango de precio no incluyen el software propietario de la empresa que los fabrica, de tal forma que se tiene que recurrir a la compra adicional del mismo o recurrir a software de terceros que no van a brindar la misma compatibilidad y características del software propietario diseñado para controlar ese equipo. De esta manera, se determinó que el costo final de la propuesta diseñada es competitivo en comparación de los dispositivos comerciales más económicos, ya que a pesar que en algunos casos el mismo supera los costos de los dispositivos de esta gama, incluye características adicionales que compensan esa diferencia.

4.4.2 Factibilidad Técnica

En cuanto al estudio de factibilidad técnica, se procedió a analizar los diferentes recursos técnicos que serían necesarios al momento de implementar la propuesta, para ello se consideraron los diferentes contextos en los que se podría implementar este diseño y que recursos adicionales serían necesarios. Ya que la propuesta está pensada para ser implementada en un entorno industrial en el que se desee mejorar el control supervisorio de los procesos existentes, se determinó que la propuesta es factible técnicamente ya que los recursos técnicos necesarios para su implementación son los mismos existentes de los procesos actuales, como recurso técnico adicional que puede ser necesario es un computador para instalar la HMI, sin embargo, cualquier equipo disponible dentro de la planta puede ser empleado para esta labor ya que dicha interfaz no consume recursos importantes al momento de ser ejecutada.

4.4.3 Factibilidad Operativa

Y por parte del estudio de factibilidad operativa, también se determinó que la propuesta era factible operativamente ya que tanto la HMI, como la UIAD, fueron diseñados teniendo en cuenta este factor de operatividad, de tal forma que no son

necesarios conocimientos avanzados en ninguna disciplina relacionada con el control, la instrumentación o la programación, haciendo la manipulación de la UIAD y su interfaz accesible para cualquier operador dentro de la planta. Adicionalmente, fueron añadidos dentro de la HMI una serie de manuales que explican de forma detallada el funcionamiento de la tarjeta y la interfaz, de esa forma, si se llegaran a presentar dudas sobre la manipulación de alguna de ellas, se podría solucionar rápidamente sin la necesidad de realizar algún proceso de formación adicional.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

En la fase IV de la presente investigación referente a la factibilidad de la propuesta, se estudiaron diversas opciones comerciales de tarjetas de adquisición de datos que ofrecían prestaciones de control y supervisión de procesos industriales a través de la medición de variables de proceso adquiridas a través de los canales de entrada de dichos dispositivos, sin embargo, las opciones de más bajo costo poseían prestaciones limitadas a la hora de realizar labores de control moderno y la mayoría requerían la compra de recursos adicionales como canales de expansión o el mismo software de adquisición de datos, de tal forma que su adquisición no era factible, teniendo que recurrir a alternativas de gamas superiores, elevando de gran manera el costo de los equipos, evidenciando y confirmado lo expuesto en el planteamiento del problema, los equipos de adquisición de datos suelen tener un costo elevado y el usuario no obtiene prestaciones que se adapten a sus necesidades.

Partiendo de esta problemática se procedió a realizar la propuesta del presente trabajo de grado, para ello se estudió primeramente los requisitos mínimos que debían tener los procesos de múltiples lazos, de tal forma que la tarjeta de adquisición de datos por diseñar pudiera ser adaptada a un gran número de procesos industriales ofreciendo prestaciones de control moderno y supervisión, escogiendo el número de entradas y salidas viable para el control de los mismos, las señales analógicas estandarizadas de mayor uso en la actualidad y los componentes electrónicos con las mejores prestaciones a un bajo costo para realizar los respectivos acondicionamientos requeridos.

Una vez realizado el estudio previo para la realización del diseño de la tarjeta de adquisición de datos y su respectiva interfaz, se procedió con el desarrollo de la

segunda y tercera fase de esta investigación, en las que se abarcarían ambos diseños, obteniendo como resultado principal a la UIAD, una tarjeta de adquisición de datos que cumple con los requerimientos mínimos establecidos en la fase previa, un dispositivo de 8 canales de entrada analógicos que permite el acondicionamiento y digitalización de hasta seis tipos diferentes de señales estandarizadas, igualmente, la misma posee 8 canales de salida híbridos, es decir, pueden ser empleados tanto como salidas analógicas como salidas digitales, proporcionándole al usuario la capacidad de escoger el tipo de control que desea en su proceso, ya sea un control tradicional como es el caso del control todo o nada, o incluso metodologías del control moderno como lo es el control proporcional integral derivativo, posibilidad que está totalmente descartada en los equipos comerciales de las gamas más bajas. Adicionalmente, la UIAD solventa otro de los problemas expuestos en el planteamiento del problema de esta investigación, el cual es el tema de la compatibilidad y la usabilidad, el primero de ellos está relacionado a la capacidad de emplear la UIAD en cualquier sistema operativo y en cualquier ordenador, ya que se emplearon elementos de diseño universales para solucionar cualquier problema de compatibilidad, mientras que el segundo problema está asociado a la falta de simplicidad en la manipulación de las interfaces de usuario de alternativas comerciales, problemática que fue igualmente solventada con la HMI de la UIAD ya que fue desarrollada apoyándose en los fundamentos del “UX Design” aportando esa capa de humanización a la interfaz orientado a la buena experiencia del usuario final.

Por otra parte, la propuesta diseñada no solo solventa las problemas orientados a la experiencia del usuario, también incursiona en el mercado de las tarjetas de adquisición de datos de bajo costo, haciendo frente a sus precios, con un costo total de fabricación competitivo, el cual fue obtenido a partir de un profundo estudio de costos presentado en la fase IV de esta investigación, de tal forma que la misma ofrece prestaciones incluso superiores que las alternativas comerciales por un costo accesible, proporcionándole al usuario un producto con características interesantes por un precio adecuado. No obstante, la propuesta aquí planteada aún posee margen

de mejora y se invita a futuros investigadores a que continúen con el estudio de este tópico inexplorado y alcanzar objetivos en la materia cada vez más ambiciosos.

5.2 Recomendaciones

Primeramente, se recomienda a empresas con un presupuesto reducido que deseen mejorar el control y la supervisión de sus procesos industriales, que implementen en su totalidad la propuesta aquí planteada, ya que por un presupuesto similar es poco probable conseguir las características demostradas en la presente investigación.

En segundo lugar, para futuros investigadores, se recomienda la implementación de componentes de montaje superficial o SMD, de tal forma que las dimensiones del diseño electrónico se vean reducidas, aprovechando de una mejor manera el espacio y disminuyendo los costos tanto de la estructura de almacenamiento como del PCB.

Adicionalmente, es recomendado emplear convertidores analógicos-digitales externos con una mayor tasa de muestras por segundo, al igual que emplear un microcontrolador con una velocidad de reloj mayor, esto con la finalidad de poder muestrear señales con variaciones mucho más rápidas.

Por otra parte, para aplicaciones mucho más complejas se recomienda la extensión de la investigación hacia la implementación de protocolos digitales, de tal forma que la propuesta diseñada sea capaz de admitir cualquier tipo de señal estandarizada presente en el mercado.

Finalmente, es recomendado realizar la compra de los componentes electrónicos al mayor con el objetivo de reducir los costos por unidad, reduciendo enormemente el costo final de la propuesta.

REFERENCIAS BIBLIOGRÁFICAS

- Arduino (2022). **Hardware de Arduino**. Disponible en: <https://www.arduino.cc/en/hardware#boards-1> [Consulta: abril, 20, 2022]
- Arias, F. (2006). **Proyecto de Investigación: Introducción a la Metodología**. Edición N°5. Caracas: Editorial Episteme.
- Barreto & Márquez (2016) **Desarrollo de un Sistema de Adquisición de Datos Portátil para el Banco de Turbina Kaplan Instalada en el Laboratorio de Hidráulica “Elías Sánchez Díaz”, en la Escuela De Ingeniería Civil de la Facultad de Ingeniería de la Universidad de Carabobo** (Tesis de pregrado). Universidad de Carabobo, Valencia, Venezuela.
- Bavaresco, A. (2006). **Proceso metodológico en la investigación**. Disponible en: <https://gsosa61.files.wordpress.com/2015/11/proceso-metodologico-en-la-investigacion-bavaresco-reduc.pdf> [Consulta: junio, 25, 2021]
- Calderón (2017) **Sistema de Adquisición de Datos Ambientales en los Páramos Venezolanos para el Estudio del Cambio Climático** (Tesis de pregrado). Universidad Central de Venezuela, Caracas, Venezuela.
- Calderón & Montero (2004) **Mediciones e Instrumentación Industrial**. Edición N°5.
- Cárdenas (2009) **Transductores para Procesos Industriales**. Edición N° 1. Mérida, Universidad de los Andes.
- Constitución de la República Bolivariana de Venezuela** (1999). Disponible en: <http://pdba.georgetown.edu/Parties/Venezuela/Leyes/constitucion.pdf>
- Creus, A. (2010) **Instrumentación Industrial**. Edición N° 8 México, Editorial Alfaomega Grupo Editor
- DATAQ Instruments (2022). **Catálogo de Sistemas de Adquisición de datos USB**. Disponible en: <https://www.dataq.com/data-acquisition/usb/> [Consulta: abril, 28, 2022]

- Dulhoste (2014) **Medición de Nivel**. Edición N° 1, Mérida, Universidad de los Andes.
- Fairchild Semiconductor (2006) **Datasheet LM7805**. Disponible en: <https://www.jameco.com/Jameco/Products/ProdDS/786138.pdf> [Consulta: abril, 20, 2022]
- Gálvez, F (2002) **Acondicionamiento de Señal**. Edición N° 1, Valencia, Universidad de Valencia
- González, J. (2020) **Conversión Analógico-Digital de señales de electroencefalografía** (Tesis de pregrado). Universidad de Sevilla, Sevilla, España.
- González, Pérez & Dunia (2018) **Diseño de un sistema de adquisición y procesamiento de datos con el uso de multisensores para laboratorios de física universitarios**. Revista Ingeniería UC, Valencia, Venezuela.
- Hernández, Fernández y Baptista (2006) **Metodología de la investigación**. Disponible en: https://www.esup.edu.pe/descargas/dep_investigacion/Metodologia%20de%20Ia%20investigaci%C3%B3n%205ta%20Edici%C3%B3n.pdf [Consulta: junio, 25, 2021]
- LabJack (2022). **Catálogo de Sistemas de Adquisición de datos**. Disponible en: <https://labjack.com/products> [Consulta: abril, 28, 2022]
- Ley Orgánica de Ciencia, Tecnología e Innovación** (2010) (Gaceta Oficial N.º39.575). (2010, diciembre 16). Disponible en: http://www.superior.consejos.usb.ve/sites/default/files/GO_39575_16DIC10.pdf
- Measurement Computing (2022). **Catálogo de Sistemas de Adquisición de datos**. Disponible en: <https://mccdaq.com/pdfs/Data-Acquisition-Catalog-EN.pdf> [Consulta: abril, 28, 2022]

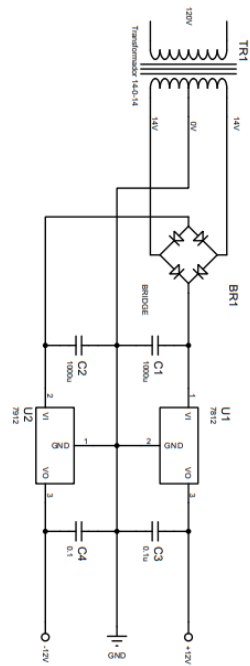
- Oliva, P (2009) **Construcción de Lista de Chequeo en Salud. La metodología para su construcción.** Edición N°1, Chile, Ministerio de Salud del Gobierno de Chile.
- Osorio, Pérez & Rodríguez (2010) **Implementación de un sistema de adquisición de datos para monitorear una máquina de corriente directa.** Revista Tecnura, Bogotá, Colombia. Universidad Pedagógica Libertador. (2011).
- Pertuz (2018) **Análisis y Diseño de Aplicación para la gestión económica y de servicios de la empresa Vital Plus** (Tesis de pregrado). Universidad Nacional Abierta y a Distancia, Barranquilla, Colombia
- Philips (1995) **Datasheet LM324N.** Disponible en: <http://www.datasheet.es/PDF/49376/LM324-pdf.html> [Consulta: abril, 20, 2022]
- Philips (1990) **Datasheet 74HC238.** Disponible en: <https://www.alldatasheet.es/datasheet-pdf/pdf/15566/PHILIPS/74HC238.html> [Consulta: abril, 20, 2022]
- Sabino, C. (2004) **El proceso de la investigación.** Disponible en: http://paginas.ufm.edu/sabino/ingles/book/proceso_investigacion.pdf [Consulta: junio, 25, 2021]
- Sánchez (2017) **Diseño de un Sistema de Adquisición de Datos y Control de un Banco de Pruebas para Reguladores de Gas** (Tesis de pregrado). Universidad Central de Venezuela, Caracas, Venezuela.
- Smith, C (1991) **Control Automático de Procesos. Teoría y Práctica.** Edición N°1, México, Editorial Limusa.
- Suarez, A. (2019) **Diseño de Hardware y Firmware para un Sistema Inalámbrico de Adquisición de Datos (DAQ) de bajo costo** (Tesis de pregrado). Universidad Nacional de Colombia, Manizales, Colombia
- Tamayo y Tamayo. (2004) **El Proceso de Investigación Científica.** Edición N°4 México, Editorial Limusa

Unisonic Technologies (2020) **Datasheet 4066**. Disponible en:

<http://www.unisonic.com.tw/datasheet/4066.pdf> [Consulta: abril, 20, 2022]

Universidad Pedagógica Experimental Libertador (2011) **Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales**. Edición N° 4, Editorial FEDEUPEL, Caracas, Venezuela.

APÉNDICE A



	<p>Miguel Freytes Email: mlgualfreytes09@gmail.com C1:27 919 861 Tel: +58 424 4097030</p>	<p>Fuente de Alimentación Simétrica Sheet 1 Of 1 REVISION 1.0 miércoles, 4 de mayo de 2022</p>	
--	--	---	--