

# **UNIVERSIDAD MAYOR DE SAN ANDRÉS**

## **FACULTAD DE INGENIERÍA CARRERA DE INGENIERÍA ELECTRÓNICA**



### **PROYECTO DE GRADO**

## **DISEÑO DE UN SISTEMA DE ASISTENCIA QUE AUTOMATIZA LA TOMA DE MEDICIONES EN EL PROCESO DE CALIBRACIÓN DE MEDIDORES DE FLUJO LÍQUIDO.**

POSTULANTE: BRYAN JHAMIL BASCOPE APAZA

TUTOR: ING. FÉLIX JAVIER SANABRIA GARCÍA

La Paz – Bolivia  
2022



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE INGENIERIA**



**LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.**

***Dedicatoria.***

*Dedicado a la memoria de mi madre, Juana Apaza,  
por el apoyo, guía y amor incondicional,  
jamás te olvidare.*

### **Agradecimientos.**

*A mi tutor, Javier Sanabria García, por su paciencia y guía en el desarrollo y culminación de este proyecto.*

*Al Ingeniero Erwin Choque de la empresa METROLAB S.R.L., por su apoyo para la elaboración de este trabajo.*

*La Universidad Mayor de San Andrés, a la Facultad de Ingeniería y a la Carrera de Ingeniería Electrónica por mi formación académica.*

*A mis tíos, Luciano Cuentas y Teresa Apaza, por brindarme apoyo y haberme acogido en los momentos difíciles de mi vida.*

*A mi familia, por su colaboración y confianza.*

## **RESUMEN**

En la industria y laboratorios se utilizan medidores de flujo líquido en diversas aplicaciones; por lo cual, estos equipos deben ser calibrados de forma periódica. Actualmente, la empresa METROLAB SRL brinda el servicio de calibración de medidores de flujo líquido. Para ello, con el procedimiento actual, se operan válvulas, bombas, transmisores, y se realizan toma de mediciones de forma manual. Esto implica la participación mínima de dos operadores realizando tareas de forma simultánea; lo cual deriva en inconvenientes para obtener medidas confiables. Entonces, el procedimiento de toma de mediciones es ineficiente.

Por lo mencionado, en el presente proyecto se diseña un sistema de asistencia que automatiza la toma de mediciones en el proceso de calibración de medidores de flujo líquido. Para ello, se considera el método de calibración por comparación usando señales de pulsos y se toma como modelo el banco de calibración del Laboratorio de Flujo de METROLAB.

Basado en el diseño, se implementa un prototipo que integra Hardware y una Interfaz gráfica de usuario. El hardware consiste de tarjetas de circuito impreso conectadas con un microcontrolador que realiza las lecturas de señales analógicas provenientes de los transmisores de temperatura, presión y caudal, y el conteo de pulsos de los medidores de flujo líquido. La interfaz gráfica se programa en LabVIEW y permite la configuración, visualización y almacenamiento de las mediciones.

Finalmente, para validar el diseño, se realizan pruebas de toma de medidas con el prototipo para demostrar que el sistema es funcional; también, se analiza la consistencia y repetibilidad de los datos adquiridos para verificar que se obtienen mediciones confiables

# Índice de Contenido

<b>1. Capítulo 1- Introducción.</b>	<b>1</b>
1.1. ANTECEDENTES.....	2
1.2. SITUACIÓN ACTUAL.....	5
1.2.1. FACTORES QUE INFLUYEN EN UNA MEDICIÓN.....	5
1.2.2. NORMATIVA BOLIVIANA.....	5
1.2.3. CONFIABILIDAD DE LAS MEDICIONES.....	6
1.2.4. PROCEDIMIENTOS DE CALIBRACIÓN.....	6
1.2.5. TRABAJOS RELACIONADOS.....	7
1.3. PLANTEAMIENTO DEL PROBLEMA.....	7
1.4. OBJETIVOS.....	8
1.5. JUSTIFICACION.....	8
1.6. ALCANCES Y LIMITACIONES.....	9
<b>2. Capítulo 2 - Marco Teórico.</b>	<b>11</b>
2.1. SISTEMAS DE MEDIDA ELECTRÓNICOS.....	11
2.1.1. ADQUISICION DE DATOS.....	11
2.1.2. SISTEMAS DE MEDIDA CON UN ADC POR CANAL.....	12
2.2. SISTEMAS DE ADQUISICIÓN DE DATOS.....	13
2.2.1. SISTEMAS BASADOS EN COMPUTADOR.....	14
2.3. TRANSMISORES.....	14
2.3.1. CONEXIÓN.....	15
2.3.2. REGION DE OPERACIÓN.....	16
2.3.3. TIEMPO DE RESPUESTA.....	17
2.3.4. ESCALAMIENTO.....	18
2.4. SEÑALES DE PULSOS EN MEDIDORES.....	19
2.5. AISLAMIENTO.....	20
2.6. LabVIEW.....	21
2.6.1. INSTRUMENTOS VIRTUALES (VI).....	22
2.6.2. MÁQUINAS DE ESTADOS.....	24
2.6.3. ARQUITECTURA PRODUCTOR/CONSUMIDOR.....	26
2.7. MICROCONTROLADOR dsPIC30F4013.....	27
<b>3. Capítulo 3 - Marco de Referencia.</b>	<b>30</b>

3.1.	MÉTODO DE CALIBRACION POR COMPARACIÓN.....	30
3.1.1.	CALIBRACIÓN USANDO SEÑALES DE PULSOS. ....	31
3.2.	BANCO DE CALIBRACION DE METROLAB .....	33
3.2.1.	EQUIPOS USADOS POR METROLAB. ....	34
3.3.	FACTORES DE INFLUENCIA EN LAS MEDICIONES.....	37
3.4.	ANALISIS DE CONFIABILIDAD DE LAS MEDICIONES.....	38
3.4.1.	ANALISIS DE CONSISTENCIA.....	38
3.4.2.	ANALISIS DE REPETIBILIDAD. ....	39
<b>4.</b>	<b>Capítulo 4 - Desarrollo del Sistema. ....</b>	<b>41</b>
4.1.	ANÁLISIS DE REQUERIMIENTOS. ....	41
4.1.1.	VISIÓN DEL SISTEMA. ....	41
4.1.1.1.	ESQUEMA GENERAL.....	41
4.1.2.	FUNCIONALIDADES.....	42
4.1.3.	ATRIBUTOS DEL SISTEMA.....	43
4.2.	ARQUITECTURA Y DISEÑO. ....	44
4.2.1.	HARDWARE.....	46
4.2.2.	PROGRAMA PARA EL DISPOSITIVO PROGRAMABLE. ....	53
4.2.3.	INTERFAZ GRÁFICA DE USUARIO. ....	64
<b>5.</b>	<b>Capítulo 5 - Prototipo.....</b>	<b>73</b>
5.1.	HARDWARE.....	73
5.1.1.	ACONDICIONAMIENTO DE SEÑALES DE TRANSMISORES M1. ....	73
5.1.2.	ACONDICIONAMIENTO Y SINCRONIZACIÓN DE SEÑALES DE PULSOS M2. 83	
5.2.	PROGRAMACION DEL DP.....	90
5.2.1.	DEFINICION DE VARIABLES PARA EL PROGRAMA. ....	94
5.2.2.	FUNCIONES DE LOS MÓDULOS.....	96
5.2.2.1.	MODULO ADC.....	96
5.2.2.2.	MODULO DE CONTEO.....	98
5.2.2.3.	MODULO I/O. ....	100
5.2.2.4.	MODULO COMM.....	101
5.2.3.	ORDENAMIENTO DEL CÓDIGO. ....	104
5.3.	PROGRAMACIÓN DE LA GUI.....	104
5.3.1.	ELEMENTOS VISUALES.....	105

5.3.2.	PROGRAMACIÓN DEL DIAGRAMA DE BLOQUES.....	105
5.3.2.1.	BLUCLE PRODUCTOR.....	106
5.3.2.2.	BUCLE CONSUMIDOR.....	112
5.3.2.3.	PROGRAMA.....	112
5.4.	INTEGRACIÓN.....	114
<b>6.</b>	<b>Capítulo 6 – Validación del diseño.....</b>	<b>117</b>
6.1.	FUNCIONAMIENTO DEL SISTEMA.....	117
6.2.	ANÁLISIS DE CONFIABILIDAD DE LAS MEDICIONES.....	122
6.2.1.	CONSISTENCIA DE LAS MEDICIONES.....	124
6.2.1.	REPETIBILIDAD DE LAS MEDICIONES.....	128
<b>7.</b>	<b>Capítulo 7 – Conclusiones y Recomendaciones.....</b>	<b>130</b>
7.1.	CONCLUSIONES.....	130
7.2.	RECOMENDACIONES.....	131
<b>8.</b>	<b>Bibliografía.....</b>	<b>132</b>
	ANEXO A – Periféricos y registros del microcontrolador DSPIC30F4013.....	135
	ANEXO C – Valores de C crítico para análisis de datos atípicos.....	142
	ANEXO D – Valores de <b>t<sub>95</sub></b> para cálculo de repetibilidad.....	143
	ANEXO E – Relación Voltaje-Corriente de los canales del módulo M1 del Hardware.....	144
	ANEXO F – Código completo del microcontrolador.....	149
	ANEXO G – Funciones del programa en LabVIEW.....	157
	ANEXO H – Tarjetas del Hardware.....	166
	ANEXO H.1. TARJETA 1.....	166
	ANEXO H.2. TARJETA 2.....	167
	ANEXO H.3. TARJETA 3.....	169
	ANEXO I – Cartas.....	171



# Índice de Figuras.

Figura 1. Banco de calibración.....	2
Figura 2. Línea de calibración en el laboratorio de METROLAB.....	3
Figura 3. Funciones de un sistema de medida.....	11
Figura 4. Etapas en la adquisición de datos. ....	12
Figura 5. Sistema de medida multicanal con un único ADC. ....	12
Figura 6. Sistema de medida multicanal con un ADC por canal. ....	13
Figura 7. Esquema sistema de adquisición.....	14
Figura 8. Sistema de adquisición basado en computador.....	14
Figura 9. Conexión de un transmisor a 4 hilos. ....	15
Figura 10. Conexión de transmisor a 2 hilos.....	16
Figura 11. Circuito para obtener señal de voltaje de un transmisor a 2 hilos.....	16
Figura 12. Región de operación típica de un transmisor .....	17
Figura 13. Tiempo de respuesta en un transmisor. ....	18
Figura 14. Escalamiento en un transmisor.....	19
Figura 15. a) conexión de salida de pulsos activa. b) conexión de salida de pulsos pasiva. .....	20
Figura 16. Lazo de tierra. ....	21
Figura 17. Panel frontal de un VI en LabVIEW.....	23
Figura 18. Diagrama de bloques de un VI en LabVIEW .....	23
Figura 19. Máquina de estados típica para sistemas de instrumentación .....	25
Figura 20. Programación de una máquina de estados en LabVIEW .....	26
Figura 21. Esquema de programación productor consumidor.....	27
Figura 22. Diagrama de pines del dsPIC30F4013.....	28
Figura 23. Diagrama esquemático de calibración por comparación.....	30
Figura 24. Calibración por comparación utilizando las señales de pulsos.....	31
Figura 25. Sincronización de pulsos. ....	32
, Figura 26. Elementos de la planta de calibración.....	34
Figura 27. Medidor de flujo patrón KHRONE IFC110. ....	35
Figura 28. Conexión de pulsos activos del medidor patrón a) de tipo activo b) de tipo pasiva.....	35
Figura 29. a) Transmisor de temperatura b) Trasmisor de presión c) Transmisor de caudal. .....	36
Figura 30. Perfil ideal de flujo en circulación. ....	37
Figura 31. Esquema general del sistema.....	41
Figura 32. Distribución de unidades en los componentes del sistema.....	44
Figura 33. Elementos del Hardware.....	45
Figura 34. Arquitectura del sistema.....	45
Figura 35. Entradas y salidas internas del hardware.....	46
Figura 36. Módulos para el hardware. ....	47
Figura 37. Módulos de primer y segundo nivel para el hardware.....	48
Figura 38. Submódulos para entrada de transmisor de 2 o 4 hilos.....	49
Figura 39. Acondicionamiento y medida de transmisores.....	50

Figura 40. Conteo de pulsos en medidor bajo prueba y patrón.....	51
Figura 41. Diagrama de tiempos para la sincronización de los pulsos.....	52
Figura 42. Elementos del módulo de M2.....	52
Figura 43. Organización de módulos y submódulos de Hardware.....	53
Figura 44. Entradas y salidas al DP.....	54
Figura 45. Trama generada.....	55
Figura 46. Módulos e interconexión en el DP.....	56
Figura 47. Funciones módulo M1.....	56
Figura 48. Funciones módulo M2.....	57
Figura 49. Funciones módulo M3.....	57
Figura 50. Funciones módulo M4.....	57
Figura 51. Procesos para el programa del DPC.....	59
Figura 52. Diagrama de flujo en PROCESO 1.....	60
Figura 53. Diagrama de flujo para el PROCESO 2 en a) DUT b) medidor patrón.....	61
Figura 54. Diagrama de PROCESO 0 y 3.....	63
Figura 55. Interfaz hombre máquina.....	64
Figura 56. Elementos de la interfaz gráfica de usuario.....	65
Figura 57. Conexión de elementos de la GUI.....	66
Figura 58. Áreas de la pantalla.....	67
Figura 59. Arquitectura del programa de la GUI.....	69
Figura 60. Máquina de estados para procesamiento y almacenamiento.....	70
Figura 61. Diagrama de flujo para el consumidor.....	72
Figura 62. Submódulos para entrada de transmisor de 2 o 4 hilos.....	73
Figura 63. Selección de tipo de transmisor con conmutador bipolar.....	74
Figura 64. Circuito para submódulo M11.....	76
Figura 65. Operación lineal del SFH600.....	76
Figura 66. Circuito conversor de voltaje a corriente.....	77
Figura 67. Circuito de aislamiento con SFH600.....	78
Figura 68. Circuito del submódulo M12.....	78
Figura 69. Circuito amplificador no inversor.....	79
Figura 70. Circuito del submódulo M13.....	80
Figura 71. Filtro RC.....	80
Figura 72. Circuito para submódulo M14.....	81
Figura 73. Circuito para módulo M1.....	82
Figura 74. Relación voltaje corriente.....	83
Figura 75. Elementos del módulo de M2.....	83
Figura 76. Circuito submódulo M21.....	84
Figura 77. Circuito comparador con histéresis no inversor.....	85
Figura 78. Circuito del submódulo M22.....	86
Figura 79. Circuito del módulo M23.....	87
Figura 80. Diagrama de tiempos para la sincronización de los pulsos.....	87
Figura 81. Circuito para sincronización de los pulsos del medidor bajo prueba y patrón...	88
Figura 82. Circuito de módulo M2.....	89
Figura 83. Esquema de interacción de microcontrolador y elementos del sistema.....	91

Figura 84. Designación de pines en el microcontrolador.....	93
Figura 85. Interfaz de usuario.....	105
Figura 86. Esquema de bloques en el programa de LabVIEW.....	106
Figura 87. Diagrama de bloques del estado Crear fichero.....	107
Figura 88. Diagrama de bloques del estado "programación".....	108
Figura 89. Diagrama de bloques del estado "grabación en uC".....	109
Figura 90. Diagrama de bloques del estado "grabación en uC".....	110
Figura 91. Diagrama de bloques del estado "guarda dato".....	111
Figura 92. Diagrama de bloques del estado "detener proceso".....	111
Figura 93. Diagrama de bloques de bucle consumidor.....	112
Figura 94. diagrama de bloques del programa en LabVIEW.....	113
Figura 95. Esquema de tarjetas y su conexión.....	115
Figura 96. Cable PL2303 para conexión de Hardware con la GUI.....	116
Figura 97. Inicio del programa.....	117
Figura 98. Visualización y conteo en pruebas fuera de laboratorio.....	118
Figura 99. Datos adquiridos en pruebas fuera de laboratorio.....	118
Figura 100. Sistema funcionando en laboratorio.....	119
Figura 101. Pulsos del DUT y patrón.....	121
Figura 102. Datos obtenidos en las pruebas de laboratorio.....	122
Figura 103. Tiempos de conversión Analógico a digital.....	136
Figura 104. Diagrama de bloques simplificado de módulo UART.....	139
Figura 105. Relación voltaje corriente canal 1.....	144
Figura 106. Relación voltaje corriente canal 2.....	145
Figura 107. Relación voltaje corriente canal 3.....	146
Figura 108. Relación voltaje corriente canal 4.....	147
Figura 109. Relación voltaje corriente canal 5.....	148
Figura 110. SubVI encavezado_reportes.....	157
Figura 111. SubVI separa datos.....	158
Figura 112. SubVI datainport2.....	159
Figura 113. SubVI ADC_valores.....	160
Figura 114. SubVI escalamiento_general.....	161
Figura 115. SubVI valores_medio_sensores.....	162
Figura 116. SubVI Pulse_vol.....	163
Figura 117. SubVI. imprime_resultados.....	164
Figura 118. SubVI escalamiento 4_20_mA.....	165
Figura 119. Esquemático Tarjeta 1.....	166
Figura 120. Simulación Tarjeta 1.....	166
Figura 121. Tarjeta 1.....	167
Figura 122. Esquemático Tarjeta 2.....	167
Figura 123. Simulación Tarjeta 2.....	168
Figura 124. Tarjeta 2.....	168
Figura 125. Esquemático Tarjeta .3.....	169
Figura 126. Simulación Tarjeta 3.....	169
Figura 127. Tarjeta 3.....	170

Figura 128. Carta de consentimiento METROLAB..... 171

# Índice de tablas

Tabla 1. Periféricos del dsPIC30F4013 .....	28
Tabla 2. Toma de datos en el proceso de calibración por comparación.....	33
Tabla 3. Resumen de características de los transmisores.....	36
Tabla 4. Comandos para el dispositivo de procesamiento y comunicación. ....	55
Tabla 5. Pseudocódigo del proceso 1.....	60
Tabla 6. Pseudocódigo del proceso 2.....	61
Tabla 7. Pseudocódigo para proceso 0 y 3.....	62
Tabla 8. Elementos visuales por área de pantalla. ....	69
Tabla 9. Resumen de características de los transmisores.....	75
Tabla 10. Periféricos del dsPIC30F4013.....	90
Tabla 11. Designación de módulos y pines del microcontrolador. ....	93
Tabla 12. Estructura para programación del microcontrolador DSPIC30F4013 .....	94
Tabla 13. Definición de variables para el programa. ....	96
Tabla 14. Código de función init_adc.....	97
Tabla 15. Código de función leer_adc. ....	98
Tabla 16. Código de función init_timer23.....	99
Tabla 17. Código de función init_timer23.....	99
Tabla 18. Código para reinicio de contadores. ....	100
Tabla 19. Código para reinicio de contadores.....	100
Tabla 20. Código para lectura de valores de los contadores.....	100
Tabla 21. Código para función init_io.....	101
Tabla 22. Código para lectura y establecimiento de pines I/O.....	101
Tabla 23. Código de función init_uart2. ....	102
Tabla 24. Código para función de recepción de datos.....	103
Tabla 25. Código para función enviar_string. ....	104
Tabla 26. Código para función enviar_char. ....	104
Tabla 27. Valores indicados en la interfaz e instrumentos en la línea.....	120
Tabla 28. Pulsos y Volumen obtenidos en el patrón para medición de 300 L.....	123
Tabla 29. Pulsos y Volumen obtenidos en el patrón para medición de 500 L.....	123
Tabla 30. Pulsos y Volumen obtenidos en el patrón para medición de 800 L.....	124
Tabla 31. Análisis de consistencia en mediciones a 300 L. ....	125
Tabla 32. Análisis de consistencia en mediciones a 500 L. ....	126
Tabla 33. Análisis de consistencia en mediciones a 800 L. ....	127
Tabla 34. Análisis de repetibilidad en las mediciones a 300 L.....	128
Tabla 35. Análisis de repetibilidad en las mediciones a 500 L.....	129
Tabla 36. Análisis de repetibilidad en las mediciones a 800 L.....	129
Tabla 37. Valores de k crítico. ....	141
Tabla 38. Valores de C crítico. ....	142
Tabla 39. Valores t95 para cálculo de repetibilidad. ....	143
Tabla 40. Datos de voltaje-corriente en canal 1.....	144
Tabla 41. Datos de voltaje-corriente en canal. ....	145
Tabla 42. Datos de voltaje-corriente en canal 3.....	146

Tabla 43. Datos de voltaje-corriente en canal 4.....	147
Tabla 44. Datos de voltaje-corriente en canal 5.....	148

# Índice de ecuaciones

Ecuación 1. Escalamiento de un transmisor.....	19
Ecuación 2. Relación volumen constante de pulsos.....	20
Ecuación 3. Desviación estándar.....	38
Ecuación 4. k de Mandel.....	39
Ecuación 5. C de Cochran.....	39
Ecuación 6. Repetibilidad.....	40
Ecuación 7. Submódulo M11.....	48
Ecuación 8. Submódulo M12.....	48
Ecuación 9, Submódulo M13.....	49
Ecuación 10. Submódulo M14.....	49
Ecuación 11. Relación submódulo M11.....	74
Ecuación 12. Región de operación transmisor de temperatura.....	75
Ecuación 13. Región de operación transmisor de presión.....	75
Ecuación 14. Región de operación transmisor de caudal.....	75
Ecuación 15. Conversor voltaje a corriente.....	77
Ecuación 16. Amplificador no inversor.....	79
Ecuación 17. Filtro pasa bajo RC.....	80
Ecuación 18. Valor de referencia en comparador con histéresis.....	85
Ecuación 19. Cantidad de histéresis para comparador.....	85

# Glosario de Términos.

**Medición:** Proceso que consiste en obtener experimentalmente uno o varios valores que pueden atribuirse a una magnitud física.

**Metrología:** Ciencia de las mediciones y sus aplicaciones. La misma incluye todos los aspectos teóricos y prácticos de las mediciones.

**Sistema:** Conjunto de elementos que interactúan entre sí para lograr un objetivo determinado.

**Hardware:** placa de circuito impreso integrada por componentes electrónicos, eléctricos, mecánicos y/o electromecánicos.

**Periférico:** Aparato auxiliar e independiente conectado a la unidad central de un dispositivo electrónico.

**Buffer:** ubicación de la memoria de un dispositivo electrónico que está reservada para el almacenamiento temporal de información. Mientras los datos están en el buffer, aguardan para ser procesados.

**Función:** En el aspecto informático, refiere a un bloque o líneas de código que realiza una o varias operaciones.

**Planta:** Conjunto de elementos que funcionan de forma conjunta para efectuar una operación particular.

**Monitoreo:** Visualización u observación del progreso de las variables físicas o de algún otro tipo en un determinado proceso mediante un monitor o pantalla.

**Procesamiento:** Aplicación sistemática de una serie de operaciones sobre un conjunto de datos.

**Canal:** Medio físico (Como cables) por donde se transmite una señal analógica o digital.

**Pseudocódigo:** Descripción de alto nivel, compacta e informal que define el principio de operación de un programa o algoritmo.

**Eficiencia:** Capacidad de lograr los resultados deseados con el mínimo posible de recursos.



## **Acrónimos.**

**GUI.** *Graphical User Interface.* Interfaz gráfica de usuario.

**ADC.** *Analog to Digital Converter.* Converso analógico a digital.

**DP.** Dispositivo Programable.

**DUT.** *Device Under Test.* Dispositivo bajo prueba.

**HMI.** Human Machine Interface – Interfaz Hombre Maquina

# **1. Capítulo 1- Introducción.**

METROLAB SRL es una empresa que provee servicios de calibración de diversos instrumentos de medición como ser: manómetros, balanzas, termómetros, luxómetros, material volumétrico, sonómetros y medidores de flujo líquido. Sus laboratorios se encuentran en las ciudades de La Paz y Santa Cruz del Estado Plurinacional de Bolivia, y los mismos están acreditados bajo la norma ISO 17025:2018 [1]. El presente proyecto trata, específicamente, sobre el servicio de calibración de medidores de flujo líquido que brinda la empresa.

Respecto a la calibración, la misma se define como: “el conjunto de operaciones que establecen, bajo condiciones específicas, la relación entre las cantidades indicadas por un instrumento o sistema de medida y los valores que corresponden a un estándar o patrón de referencia” [2]. Básicamente, consiste en comparar la medida realizada por el equipo sujeto a calibración, denominado dispositivo bajo prueba (*Device Under Test* - DUT), con el valor o medida del patrón.

En cuanto a medidores de flujo líquido, la magnitud que se compara es la cantidad de volumen o masa; y se distinguen tres (3) métodos para su calibración [3], los cuales son:

- Método Gravimétrico: Cuando el patrón referencia es una medida de masa.
- Método Volumétrico: Cuando el patrón referencia es un recipiente volumétrico.
- Método por comparación: Cuando el patrón referencia es otro medidor de flujo ya calibrado y con mayores prestaciones metrológicas (exactitud y precisión) que el DUT.

El procedimiento actual en METROLAB para la calibración de medidores de flujo líquido se basa en el método por comparación y la magnitud que se compara es la cantidad de volumen.

El D.S 24498 de 17 de febrero de 1997, organiza y establece las actividades de metrología<sup>1</sup>, y en su artículo 37 indica: “Los Laboratorios de calibración tendrán por objeto procurar la uniformidad y confiabilidad de las mediciones que se realizan en el país, tanto en lo concerniente a las transacciones comerciales y de servicios, como a los procesos industriales y a los trabajos de investigación científica y desarrollo tecnológico” [4]. Para lograr este fin, los procedimientos deben seguir las pautas y recomendaciones establecidas en normas y estándares nacionales o internacionales; y declaraciones de los fabricantes; asegurando de esta forma buenas prácticas y condiciones en la toma de medidas.

---

<sup>1</sup> La metrología es la ciencia que estudia las mediciones y sus aplicaciones.

Bajo este escenario, el actual proyecto presenta el diseño de un sistema que automatiza la toma de mediciones en el proceso de calibración de medidores de flujo líquido. A partir del diseño, se construye un prototipo que cuenta con una interfaz gráfica desarrollada en LABVIEW en la que se configura parámetros necesarios para realizar mediciones; y hardware basado en un microcontrolador que realiza la lectura de datos de los instrumentos de medición (transmisores y medidores de flujo líquido). Para validar el diseño, se realiza pruebas con el prototipo y se demuestra que, la toma de medidas con el sistema es eficiente en cuanto los recursos que demanda y la confiabilidad de las mediciones.

### 1.1. ANTECEDENTES.

Para realizar el servicio de calibración, METROLAB, cuenta con un banco de calibración, el cual se representa en la Figura 1.

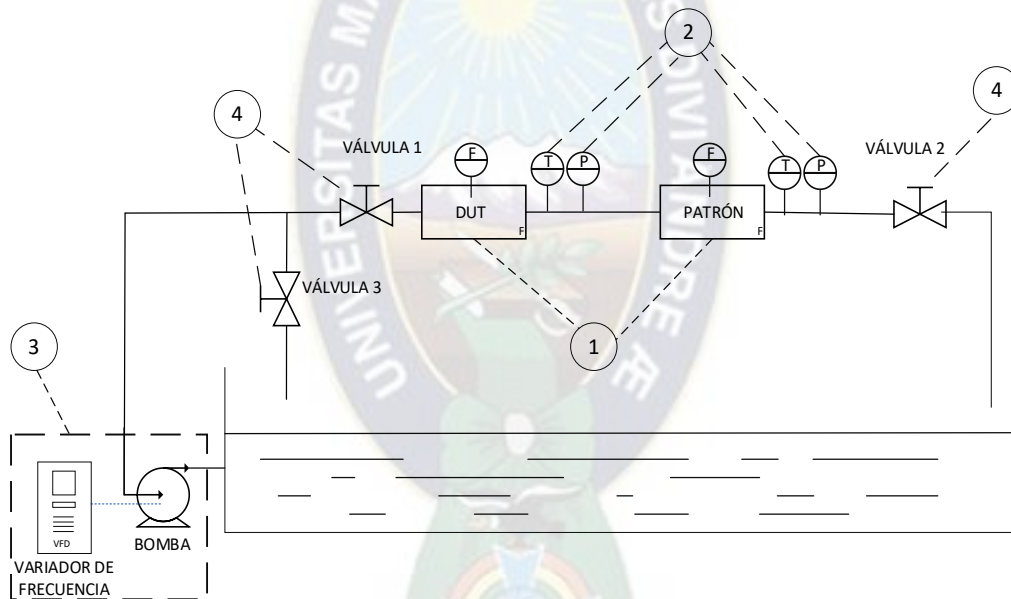


Figura 1. Banco de calibración.

Fuente: Elaboración propia.

de la Figura 1 se distinguen cuatro (4) conjuntos de elementos en el banco de calibración instalado METROLAB, los cuales son:

1. Primarios. Medidores de flujo líquido, DUT y patrón, que miden la cantidad de volumen.
2. Secundarios. Transmisores para la medición de temperatura y presión del líquido circulante.
3. De impulsión. Constituido por una bomba conectada un variador de frecuencia para su funcionamiento. Este elemento permite iniciar o detener la circulación de líquido y establecer el caudal.
4. Circuito de circulación. Compuesto por las tuberías y válvulas que conectan los elementos anteriores.

De acuerdo a los trabajos que realiza el personal durante la calibración, se distinguen tres (3) etapas: Preparación, Mediciones y Reporte de resultados. Cada una de estas etapas implica la ejecución de diversas tareas.

La **Etapa de Preparación** es la fase en la cual se realizan los preparativos para poder tomar mediciones, las actividades que se efectúan son:

- Registro de datos relevantes del DUT, como ser: marca, modelo, serie y rango de trabajo.
- Instalación del DUT en la línea de tuberías.
- Elección de los puntos de calibración. Un punto de calibración se define como el par (volumen, caudal). Entonces se elige que cantidades de volumen se medirán y a que caudales de circulación de líquido.

Dependiendo de las características de dimensión y operación del DUT, su instalación puede requerir de 2 o más operarios.

La **Etapa de Mediciones** consiste en la toma y registro de medidas en cada punto de calibración elegido en la etapa anterior. Con el procedimiento actual, esta etapa requiere el trabajo de al menos dos operarios para manejar los elementos necesarios y anotar los valores de medición. En la Figura 2 se representa la distribución del personal en el circuito de calibración.

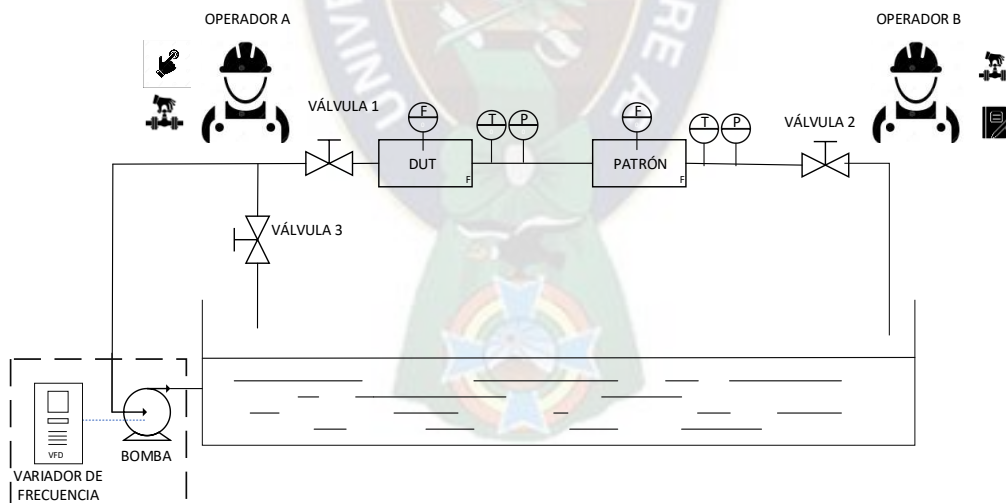


Figura 2. Línea de calibración en el laboratorio de METROLAB.

Fuente: Elaboración propia.

A continuación, se detalla las tareas que se realiza para poder efectuar y registrar una medición en un punto de calibración.

- Con las válvulas 1 y 2 abiertas y la válvula 3 cerrada, un operario A establece el caudal variando la frecuencia del motor de la bomba. Entonces, empieza a circular líquido por los medidores.

- El operario A abre la válvula 3 y cierra la válvula 1. A su vez, un operario B cierra la válvula 2. De esta manera, el líquido queda confinado en la tubería entre los medidores.
- El operario B anota los valores indicados en los visualizadores de los medidores instalados en la bancada, el medidor bajo prueba y el patrón.
- El operario B anota los valores de los transmisores instalados en la línea y las condiciones (temperatura, humedad relativa y caudal).
- El operario A abre la válvula 1 y cierra la válvula 3. Al mismo tiempo, el operario B abre la válvula 2. Entonces, nuevamente circula el líquido por los medidores.
- Una vez que se llega a un valor aproximado de volumen deseado, el operario A cierra la válvula 1 y abre la válvula 2. De manera simultánea, el operario B cierra la válvula 2, de esta forma se detiene el flujo de líquido.
- Finalmente, el operario B anota los valores de volumen indicados por el DUT y el patrón; también, anota los valores de transmisores instalados en la línea y condiciones ambientales.

La **Etapa de Reporte de resultados** es la fase final donde se exportan los valores de mediciones tomados en la etapa anterior y se prepara el certificado de calibración. Las tareas de esta etapa se pueden realizar por un operador, y las mismas son:

- Transcripción de los datos, registrados en la etapa anterior, a la planilla de cálculo de MS Excel. Los resultados se obtienen a través de las fórmulas programadas en tablas.
- Con los valores obtenidos en el punto anterior, se analiza si los resultados son confiables. En caso de que las mediciones presenten altos errores o variación, se repite la etapa de mediciones; en otro caso, se prepara el reporte de calibración.

Como se puede observar las tareas que realizan los operadores se hacen de forma manual. Algunos de estos aspectos pueden resultar en:

- la incorporación de errores,
- incumplimiento los procedimientos y buenas prácticas de medición; basados normas y estándares nacionales y/o internacionales,
- omitir factores que influyen en una medición, que deberían ser tomados en cuenta según declaración de los fabricantes.

Las empresas que realizan servicios de calibración operan en un ambiente competitivo, entonces para que sus mediciones sean precisas y se lleven a cabo de la manera más consistente y eficiente posible, automatizan sus procesos de calibración [5]. Bajo este escenario, algunas empresas o fabricantes utilizan u ofrecen sistemas integrados por módulos de hardware y software para automatizar los procesos de calibración. Fluke, por ejemplo, cuenta con el software “COMPAS ®” combinado con los componentes “molblox/molbloc” para automatizar sus procesos de calibración y aplicar las mejores prácticas [6].

## 1.2. SITUACIÓN ACTUAL.

Antes de abordar aspectos propios del proyecto, se expone, el estado de situación que se observa actualmente en cuanto a factores que influyen en una medición, normativa Boliviana, confiabilidad de las mediciones, procedimientos de calibración y trabajos relacionados.

### 1.2.1.FACTORES QUE INFLUYEN EN UNA MEDICIÓN.

- El documento técnico escrito por el Dr. Nicolaus Mathies “Calibration technical information” presentado en 2013 y publicado por el fabricante KROHNE [7], aborda temas sobre principios y definiciones, procesos de calibración de medidores de flujo líquido, factores de influencia práctica en las mediciones y patrones de referencia y medida. *De este escrito, se toma en cuenta los factores de influencia práctica en las mediciones en la calibración de medidores de flujo líquido.*
- En el blog “Flowmeter calibration: Five best practices you need to know” publicado por Larry Renda el 13 de febrero de 2013 para FLUKE, se brinda pautas para realizar las mejores prácticas en la calibración de medidores de flujo [6]. *De esta publicación se considera la recomendación sobre mediciones de flujo en estado estacionario.*
- En la guía “Pressure transmitter response time” publicado por el fabricante Yokogawa en octubre de 2019, se define el tiempo de respuesta de transmisores y su forma de aplicación en la medición con transmisores [8]. *Se toma en cuenta dichas pautas para el desarrollo del proyecto.*
- En el curso dictado por David A. Snyder “Understanding 4-20mA loops” de 2008, se brinda conceptos sobre uso de transmisores de 2 y 4 hilos [9]. *De esta guía se usa el análisis sobre regiones de operación en transmisores.*

### 1.2.2.NORMATIVA BOLIVIANA.

El Instituto Boliviano de Metrología - IBMETRO, señala que los laboratorios de calibración acreditados deben seguir los lineamientos de la Norma NB/ISO/IEC 17025:2018 “Requisitos Generales para la competencia de laboratorios de ensayo y calibración”. Esta última establece que los laboratorios de ensayo y calibración, deben demostrar que son técnicamente competentes, operan un sistema de gestión y son capaces de generar **resultados de ensayo y calibración técnicamente válidos** [10].

El D.S 24486 de 17 de febrero de 1997, organiza y establece las actividades de metrología, y en su artículo 37 indica: “**Los Laboratorios de calibración tendrán por objeto procurar la uniformidad y confiabilidad de las mediciones que se realizan** en el país, tanto en lo concerniente a las



transacciones comerciales y de servicios, como a los procesos industriales y a los trabajos de investigación científica y desarrollo tecnológico” [4].

*Tomando en cuenta lo anterior, para fines de este proyecto, se define como parámetro de eficiencia la confiabilidad de las mediciones.*

### **1.2.3. CONFIABILIDAD DE LAS MEDICIONES.**

La confiabilidad se define como: “Grado en que un experimento, ensayo o procedimiento de medición arroja resultados exactos en ensayos repetidos” [11]. De esta forma, se asocia la confiabilidad con la exactitud en mediciones repetidas. Para cuantificar este parámetro se considera:

- La Norma Ecuatoriana NTE INEN-ISO 5725-1 “Exactitud (veracidad y precisión de los resultados y sus métodos de medición)” [12] que asocia la exactitud con la consistencia y repetibilidad de los resultados. *De esta se considera las definiciones relacionadas con exactitud y formas de evaluación.*
- El curso dictado por Hugo Guerrero Postigo sobre “Validación y Verificación de Métodos” organizado por la Dirección Técnica de Acreditación DTA de IBMETRO, donde se abordan temas conceptuales y de requisitos para validación y verificación basado en la norma ISO 5725 [13]. *De este curso se toma la metodología y expresiones matemáticas para realizar el estudio de consistencia en las mediciones.*
- El estándar Británico BS ISO 11631 “Measurement of fluid Flow – Methods of specifying flowmeter performance” [14] muestra la aplicación de especificaciones técnicas y descripciones que brindan los fabricantes de medidores de flujo líquido. *De este documento se usa la metodología para el cálculo de repetibilidad en las mediciones.*

### **1.2.4. PROCEDIMIENTOS DE CALIBRACIÓN.**

- En el artículo escrito por Sejong Chun et. al. Titulado “Water Flow Meter Calibration with a Master Meter Method” de 2017, se describe en forma general los pasos para realizar la calibración con el método por comparación y se destaca los beneficios de este método [15]. *De este documento se destaca los esquemas y descripción sobre el método por comparación.*
- En el curso dictado por Dario Loza y Emmanuel Rios en el Centro Nacional de Metrología CENAM, sobre “Metrología de Flujo” en 2008, se brindan aspectos teóricos y prácticos en los procesos de calibración y sistemas de medición [16]. *De esta guía se identifica los datos que se deben tomar cuando se realizan mediciones con contadores de pulsos.*
- En el manual técnico del medidor de energía estándar SC10V fabricado por la empresa Columbus, se especifica las características y forma de calibración del equipo [17]. *De esta guía se considera la forma de sincronizar las señales de pulsos para una calibración.*

### 1.2.5. TRABAJOS RELACIONADOS.

- En la publicación realizada por Xueyuan Hangzhou “The Calibration System of Gas Flow Meter” de 2012 en la universidad China Jilang, se desarrolla un sistema de calibración de medidores de flujo gaseoso, basado en Software LabVIEW y tarjeta de adquisición PCI-6601; teniendo como resultado menor tiempo de trabajo y mejor exactitud en la rutina de calibración [18]. *Este artículo se toman las recomendaciones para implementar un circuito de sincronización de pulsos.*

### 1.3. PLANTEAMIENTO DEL PROBLEMA.

Por todos los aspectos mencionados, se puede observar que la calibración es una actividad que se realiza por empresas e instituciones nacionales o internacionales, y que además está en constante mejora de sus procesos para cumplir con las recomendaciones de estándares y fabricantes nacionales o internacionales. Para el caso de las empresas nacionales dedicadas a este rubro se establece como objeto garantizar la confiabilidad de sus mediciones.

En el caso específico de la empresa METROLAB, con base en su procedimiento actual, tomar mediciones implica anotar los datos desplegados en los visualizadores de los instrumentos de medida. Dicha tarea, no se puede realizar cuando el líquido circula de forma continua; puesto que, los valores de volumen, indicados en la pantalla de los medidores, cambian a cada instante. Entonces, los operadores deben segmentar el volumen de líquido circulante, mediante la apertura y cierre de válvulas. Esta última tarea consiste, básicamente, en abrir válvulas permitiendo la circulación de líquido, y, cuando se llegue a una determinada cantidad de volumen cerrar las mismas. Llevar a cabo las tareas de esta forma puede generar efectos, tales como:

- Toma de datos errónea debida a factores como inexperiencia o tareas presurosas por parte del personal,
- Fallo en la apertura y/o cierre de válvulas por mala operación de las mismas o falta coordinación,
- Las tareas de registro de mediciones en la calibración de un medidor se realicen en un tiempo excesivo,
- Necesidad de una alta cantidad de repeticiones en las mediciones. Dado que, los valores obtenidos pueden presentar variación; es decir, ser imprecisos,

Como consecuencia de las condiciones descritas, el sistema requiere de al menos dos operarios que deben ejecutar diversas tareas de forma simultánea y manual; además, que las mediciones son imprecisas. Por lo tanto, El procedimiento actual para la etapa mediciones es ineficiente.

En vista de lo anterior se plantea el diseño de un sistema que, a partir de la obtención de datos a través de las señales disponibles en los instrumentos de medida y un adecuado procesamiento de las mismas,



automatice la toma de mediciones. De esta forma se busca reducir el número de operadores, las tareas necesarias para realizar una medición y la imprecisión de los valores obtenidos.

Para lograr lo mencionado, es necesario comprender aspectos relacionados a: sistemas de instrumentación; adquisición, acondicionamiento, filtrado, aislamiento y escalamiento de las señales de los instrumentos de medida; la metodología de calibración de medidores de flujo líquido por comparación usando señales de pulsos y el análisis de confiabilidad de las mediciones.

#### **1.4. OBJETIVOS.**

Diseñar un sistema para automatizar la etapa toma de mediciones, en el proceso de calibración de medidores de flujo líquido, que tiene como actividad la empresa METROLAB, utilizando el método por comparación con señales de pulsos.

Para alcanzar este objetivo será necesario:

- Investigar el procedimiento y datos requeridos para la calibración de medidores de flujo líquido, utilizando señales de pulsos.
- Identificar los requerimientos para una toma de mediciones de forma automática, para un diseño adecuado del sistema.
- Diseñar un sistema para la adquisición de datos y generación de reportes. Para ello, se realiza un análisis general, de requerimientos, arquitectura y/o interconexión de elementos necesarios.
- Validar el diseño, mediante la construcción de un prototipo que permita realizar la adquisición de datos en la calibración de un medidor de flujo líquido.

#### **1.5. JUSTIFICACION.**

- **Técnica.** - La finalidad del sistema es que se constituya en una herramienta disponible para automatizar la toma de mediciones, en el proceso de calibración de medidores de flujo líquido. De esta forma, se optimiza los recursos para lograr mediciones precisas. Inicialmente, se busca su uso en el laboratorio de flujo de la empresa METROLAB.
- **Académica.** – El presente proyecto permite, a través del diseño, incursionar y sentar un precedente en temas relacionados al desarrollo de sistemas electrónicos. También, permite conocer temáticas referidas a la adquisición de datos a partir de las señales disponibles en instrumentos de medida; procesamiento de las señales de medidores de flujo líquido y sistemas de instrumentación.

## 1.6. ALCANCES Y LIMITACIONES.

Según los objetivos planteados, los alcances considerados son:

- Se aborda concretamente aspectos referidos a las mediciones y variables que se deben tomar, para la calibración de medidores de flujo líquido.
- Se toma como modelo el banco de calibración instalado en la empresa METROLAB. Se realiza una descripción de los elementos y su distribución.
- Se identifica las características técnicas de los transmisores y medidor patrón, instalados en banco de calibración.
- El diseño propuesto considera que los medidores deben tener salidas de pulsos disponibles para conectar al sistema.
- El sistema permite el monitoreo de datos, configuración y conteo de pulsos, presentación de resultados y generación de reporte.
- Los reportes que contienen los datos adquiridos por el sistema, se presentan en formato de archivos de texto.
- Se considera que el prototipo es para validación del diseño, demostrando que se reduce los recursos para la etapa de Adquisición de datos y también la imprecisión en los datos obtenidos. Por tanto, no se contempla su producción.
- Para validar el diseño se realizan mediciones con un medidor de flujo líquido y se calculan parámetros relacionados con la confiabilidad como ser consistencia y repetibilidad. Así mismo, se verifica si es eficiente en cuanto a la demanda de recursos como ser cantidad de operadores y tareas manuales.
- No se realiza ninguna evaluación de costos por tratarse de un caso de factibilidad técnica.

En cuanto a las limitaciones se tiene:

- Los procedimientos de calibración de METROLAB no son de carácter público, por lo cual no se menciona aspectos específicos del mismo. Para entender el método de calibración por comparación con señales de pulsos se usa publicaciones de institutos o laboratorios que realicen actividades de calibración.
- El sistema no realiza el ajuste de caudal a través del variador de frecuencia. Esta operación continúa siendo manual.
- El sistema no realiza la apertura y cierre de válvulas. Sin embargo, esta operación ya no es necesario; dado que, se procesan las señales de pulsos para calcular el volumen de circulación.
- En la implementación del prototipo se utiliza componentes disponibles en el territorio del Estado Plurinacional de Bolivia.

- El sistema permite al usuario ingresar datos de condiciones ambientales desde teclado. Esto es debido a que los fabricantes de equipos de medición para estas variables no presentan señales estándar para sus salidas.



## 2. Capítulo 2 - Marco Teórico.

### 2.1. SISTEMAS DE MEDIDA ELECTRÓNICOS.

Un sistema de medida electrónico es un equipo cuya finalidad es obtener los datos de un proceso físico. Los datos obtenidos se pueden visualizar, procesar y/o almacenar. Muchas variables pueden ser medidas en los procesos, por ejemplo, temperatura, presión, caudal, nivel y otros

Como se muestra en la Figura 3, se distingue tres funciones en un sistema de medida [19]:

- **Adquisición de datos.** La magnitud física a ser medida se convierte en una señal eléctrica, posteriormente se realiza un acondicionamiento para que se pueda procesar. Generalmente, se transforma a una señal de tensión adecuada para un conversor analógico - digital (ADC).
- **Procesamiento de datos.** Consiste en realizar un conjunto de operaciones matemáticas con los datos medidos y así obtener datos más significativos para el proceso.
- **Visualización o almacenamiento.** Los datos medidos y procesados, se almacenan en una memoria o registro al que un operador puede acceder en cualquier momento.

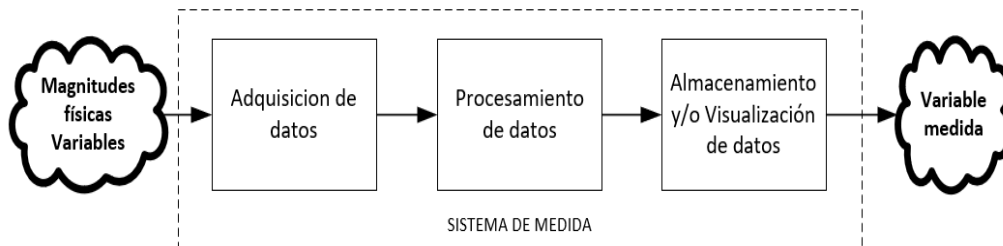


Figura 3. Funciones de un sistema de medida

Fuente: Instrumentación Electrónica [19].

#### 2.1.1. ADQUISICION DE DATOS.

La adquisición de datos de procesos consiste en la medición de variables físicas para generar datos digitales que se puedan visualizar, almacenar y analizar. Para ello, se leen señales analógicas, que pueden provenir de sensores o transmisores, y se convierten en una secuencia de datos o códigos digitales, lo cuales representa a la variable en un instante de tiempo dado. Una vez que la señal análoga es digitalizada, se torna más flexible para su almacenamiento, análisis y procesamiento de diversas formas [20].

Los elementos necesarios para la adquisición de datos de procesos son: Sensor o transmisor; acondicionamiento, conversor analógico a digital (*Analog to digital converter* – ADC). La Figura 4 muestra las etapas en la adquisición de datos de una variable física [19].

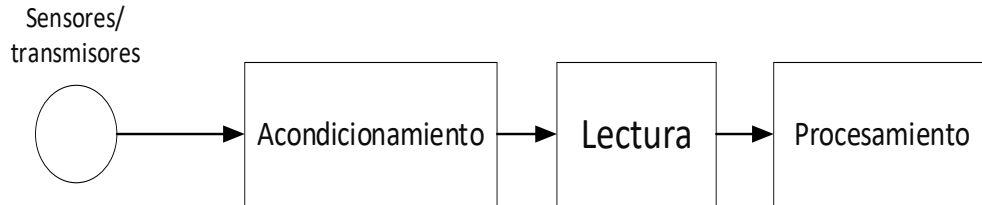


Figura 4. Etapas en la adquisición de datos.

Fuente: Instrumentación Electrónica [19].

El acondicionamiento consiste en realizar operaciones básicas como filtrado, aislamiento, amplificación sobre una señal para que sea adecuada para un procesamiento.

Después del acondicionamiento, la señal es convertida del mundo analógico a la digital mediante un elemento de lectura, el cual asigna una palabra de bits al valor de la señal acondicionada.

Finalmente, cuando se tiene el valor digital de la señal, la misma pasa a una etapa de procesamiento donde se realizan una serie de operaciones para que se puede interpretar la información y se pueda visualizar y almacenar.

### 2.1.2. SISTEMAS DE MEDIDA CON UN ADC POR CANAL.

En la mayoría de los procesos es necesario adquirir y procesar más de una variable o magnitud física. Para ello, considerando señales de tipo analógico, se tienen 2 arquitecturas básicas [19]:

- **Sistema de medida multicanal con un único ADC.** Los canales para la medición son seleccionados con un multiplexor analógico. La medición de las señales no se puede realizar de forma simultánea. La ventaja de esta arquitectura es que requiere menor cantidad de componentes para su implementación. En la Figura 5 se muestra esta arquitectura.

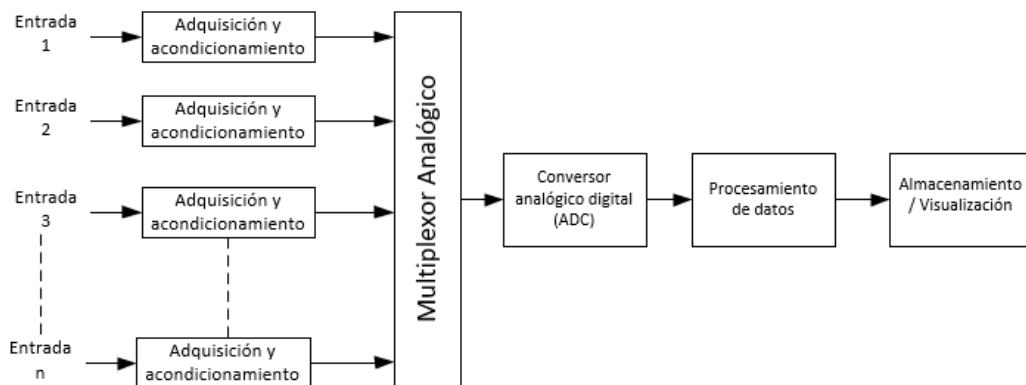
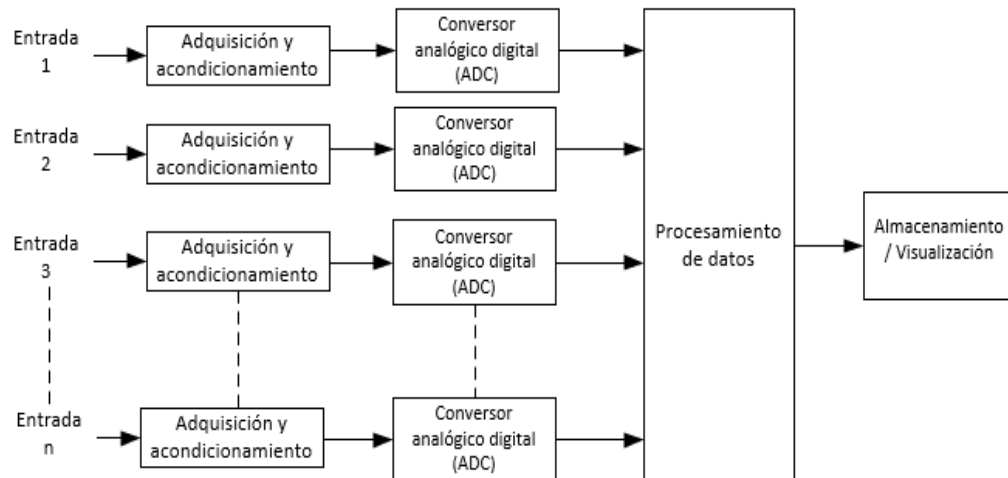


Figura 5. Sistema de medida multicanal con un único ADC.

Fuente: Instrumentación Electrónica [19].

- **Sistema de medida multicanal con un ADC por canal.** Esta arquitectura tiene la ventaja de que se registra los datos de forma simultánea y más rápida que con un multiplexor. Una desventaja es que requiere mayor cantidad de componentes para su implementación, por esta razón, se recomienda su implementación cuando no se tiene una alta cantidad de canales en el sistema. Esta arquitectura se muestra en la Figura 6.



*Figura 6. Sistema de medida multicanal con un ADC por canal.*

*Fuente: Instrumentación Electrónica [19].*

## 2.2. SISTEMAS DE ADQUISICIÓN DE DATOS

Son instrumentos o dispositivos, que sirven para obtener información de un proceso o realizar mediciones de las variables físicas y registrarlos para su análisis [21]. Un sistema de adquisición de datos contiene los elementos de adquisición y adiciona un elemento de visualización o configuración como se muestra en la Figura 7. Entonces, se tienen 5 componentes básicos:

1. Los sensores o transmisores de los cuales se obtiene una señal eléctrica a partir de un fenómeno físico.
2. Un bloque de acondicionamiento.
3. Un sistema de lectura que convierte la señal analógica a digital. En este punto, dependiendo del proceso, se puede utilizar un sistema de medida multicanal.
4. El procesamiento que realiza operaciones sobre los valores digitales de las lecturas para obtener información útil.
5. La Visualización y/o Configuración que comunica los resultados al usuario u operador y brinda la posibilidad de realizar configuración para la adquisición.



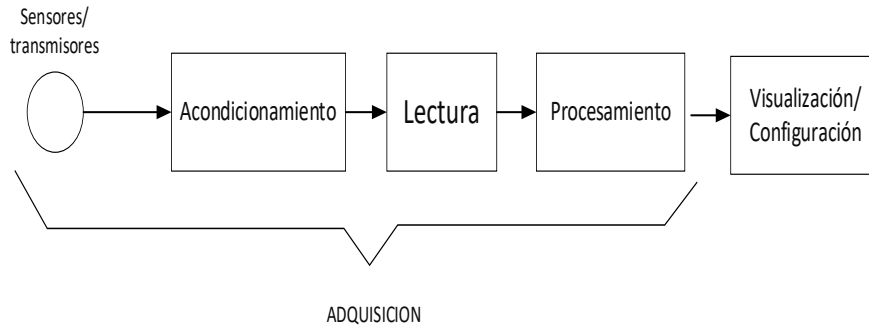


Figura 7. Esquema sistema de adquisición.

Fuente: Instrumentación Electrónica [19].

### 2.2.1. SISTEMAS BASADOS EN COMPUTADOR.

Para este tipo de sistemas se en el elemento de Visualización/Configuración se tiene un equipo computador. La adquisición se realiza mediante componentes y dispositivos electrónicos integrados en Hardware, que, en la mayoría de los casos se denomina tarjeta de adquisición. El equipo computador y el Hardware para adquisición se conectan a través de comunicación [19]. En la Figura 8 se muestra el esquema para un sistema de adquisición basado en computador.

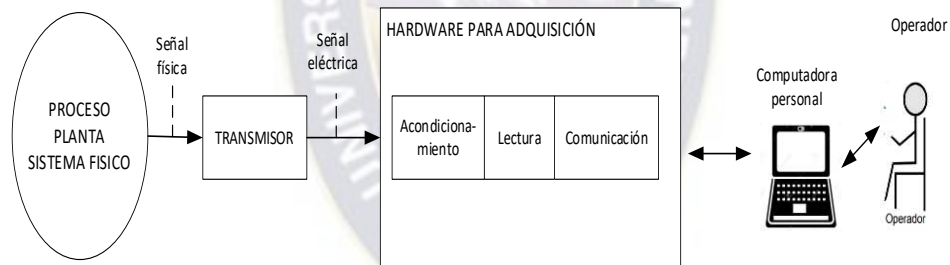


Figura 8. Sistema de adquisición basado en computador.

Fuente: Instrumentación Electrónica [19].

### 2.3. TRANSMISORES.

Un transmisor es un dispositivo que transforma una señal de magnitud pequeña proveniente de un sensor, a una señal con niveles estándares en la industria. Los niveles estándar utilizados en la industria son: 0 mA a 20 mA, 4 mA a 20 mA, 1 V a 5 V, 0 V a 10 V. La señal obtenida del transmisor representa la variable física del proceso.

Como se mencionó, la salida de un transmisor es un rango de voltaje (1 V a 5 V) o corriente (4 mA a 20 mA). El rango de salida del transmisor representa el 0% al 100 % de la variable física medida. Por ejemplo, un transmisor puede producir una salida de 4 mA a 20 mA que representa la temperatura

medida por un sensor de 0 °C a 100 °C, donde 4 mA → 0 °C (0%) y 20 mA → 100 °C (100%) [9].

### 2.3.1. CONEXIÓN.

Para el funcionamiento de un transmisor se requiere una fuente y un dispositivo indicador o controlador, que será el que procesa la señal de salida. Como se menciona en [22], Las formas más comunes para conectar un transmisor son a 4 hilos o 2 hilos.

Los transmisores de 4 hilos presentan 2 conectores para la alimentación y 2 conectores para la señal de salida sea de corriente o voltaje. La alimentación se realiza con una fuente DC o AC según especifica el fabricante. La señal de salida puede ser en tensión o corriente y se conecta directamente al indicador o controlador. Este esquema es el más sencillo, en la Figura 9 se muestra como conectar un transmisor a 4 hilos [22].

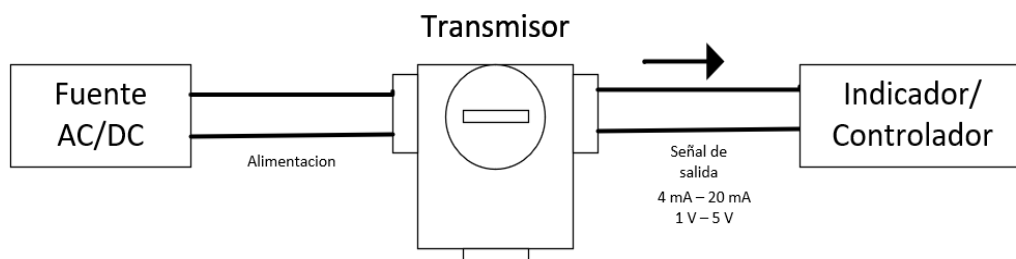


Figura 9. Conexión de un transmisor a 4 hilos.

Fuente: *What are 2-wire and 4-wire Transmitter Output Loops?* [22]

Los transmisores a 2 hilos presentan solo 2 conectores. En este caso, la fuente de alimentación y el controlador o indicador se conectan en serie. Comúnmente, la fuente de alimentación es de corriente continua DC. La señal de salida será la corriente del lazo, generalmente, de 4 mA a 20 mA. El esquema de conexión para transmisores de 2 hilos se muestra en la Figura 10 [22].



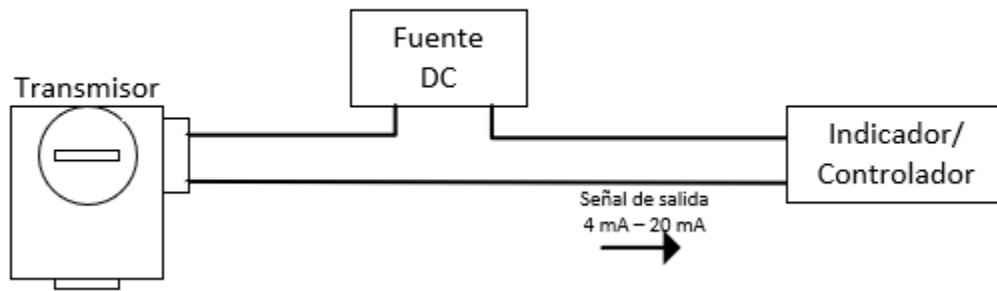


Figura 10. Conexión de transmisor a 2 hilos

Fuente: *What are 2-wire and 4-wire Transmitter Output Loops?* [22]

La señal de corriente es la preferida cuando se usa transmisores. Una de las razones principales es que presenta menos pérdidas y atenuación debida a la longitud del cable. La señal de corriente de un transmisor se puede adaptar o transformar a voltaje a través de una resistencia de precisión en el lazo. Esto para aplicaciones de visualización y/o control. En la Figura 11 se muestra el circuito para obtener una señal de voltaje a partir de la señal de corriente de un transmisor de 2 hilos.

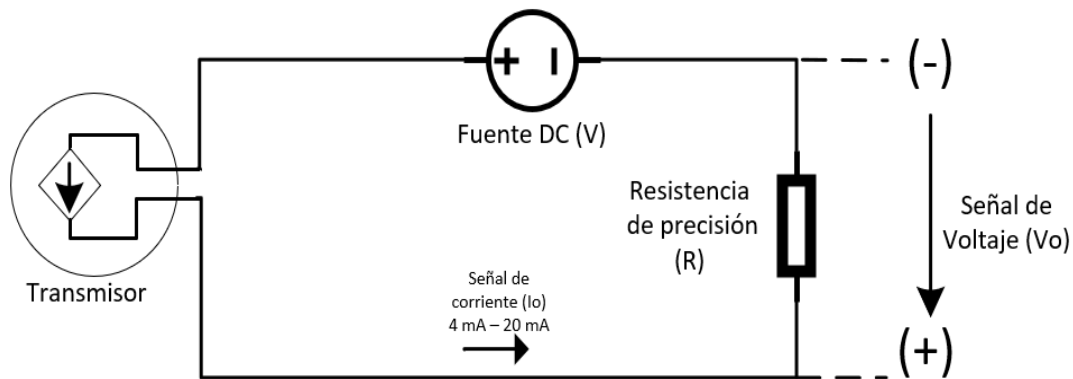


Figura 11. Circuito para obtener señal de voltaje de un transmisor a 2 hilos

Fuente: *What are 2-wire and 4-wire Transmitter Output Loops?* [22]

De la Figura 11, la señal de voltaje de salida  $V_o$  varía en función de la señal de corriente del lazo  $I_o$ . La relación entre  $V_o$  e  $I_o$  esta dada por la ley de Ohm mediante la resistencia de precisión  $R$ , es decir,  $V_o = R * I_o$ .

### 2.3.2. REGION DE OPERACIÓN.

Para el correcto funcionamiento del transmisor se debe elegir los valores adecuados de Voltaje en la fuente de alimentación y resistencia de lazo. Cada transmisor tiene una región de operación dentro de la cual el funcionamiento es efectivo y eficiente [23]. En la práctica, se debe elegir el valor de la

resistencia de lazo y voltaje de fuente de alimentación según la región de operación del transmisor. En la Figura 12 se muestra la representación de la región de operación típica de un transmisor.

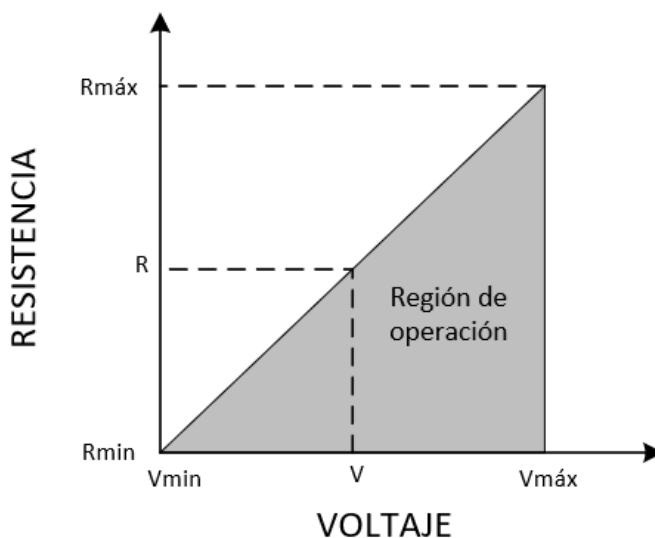


Figura 12. Región de operación típica de un transmisor  
Fuente: Understanding 4 – 20 mA [9]

### 2.3.3. TIEMPO DE RESPUESTA.

El tiempo de respuesta en un transmisor es la cantidad de tiempo en segundos que la señal de salida analógica demora en alcanzar el 63,2 % del valor un cambio de la magnitud medida. Generalmente los transmisores esta respuesta es parecida a la de un sistema de primer orden.

El tiempo de respuesta  $T_R$  tiene dos componentes que son el tiempo muerto  $T_m$  y la constante de tiempo  $T_c$ . El tiempo muerto es el tiempo previo a la transición de la señal de salida del instrumento, es decir antes del régimen transitorio. La constante de tiempo es el tiempo que el instrumento demora en llegar al 63,2 % del valor de la magnitud medida en el régimen transitorio. En la Figura 13 se muestra, de forma gráfica, el tiempo de respuesta de un transmisor.

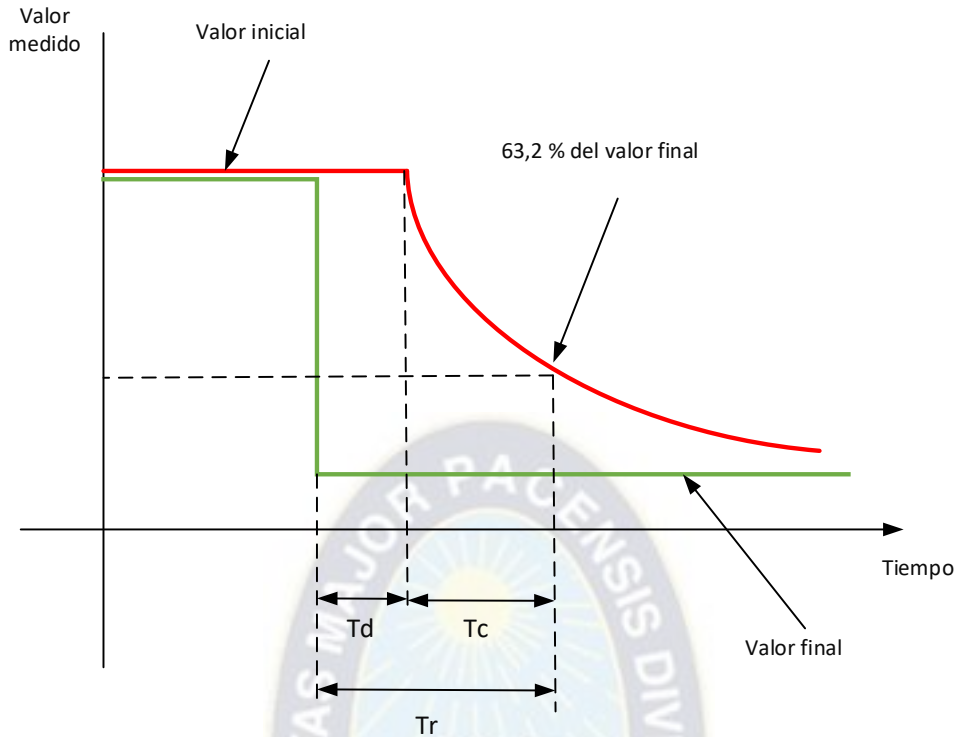


Figura 13. Tiempo de respuesta en un transmisor.

Fuente: Pressure Transmitter Response Time [8].

El tiempo de respuesta es una especificación esencial para el comportamiento de un transmisor. El hecho de que sea más rápido no siempre quiere decir que sea mejor, las buenas prácticas indican que se debe considerar esta especificación al realizar mediciones [8].

#### 2.3.4. ESCALAMIENTO.

Para conocer e interpretar la señal de salida de un transmisor se usan escalamientos. La mayoría de los transmisores presentan una respuesta lineal. Entonces, el escalamiento consiste en hallar una relación lineal entre la señal de entrada y la salida. Para hacer esta operación se requiere el rango de salida y rango de medición de un transmisor que son indicador en las especificaciones del fabricante [9].

En términos generales un transmisor tendrá un rango de salida en corriente o voltaje. Suponiendo que la salida es de corriente, la misma puede variar entre  $I_1$  e  $I_2$ . El rango de medición puede estar entre  $y_1$  e  $y_2$ . Entonces el escalamiento está dado por la Ecuación 1. En la figura 14 se muestra de forma gráfica el escalamiento.

$$y - y_1 = \frac{y_2 - y_1}{I_2 - I_1} (I - I_1) \dots \dots (1)$$

Ecuación 1. Escalamiento de un transmisor.

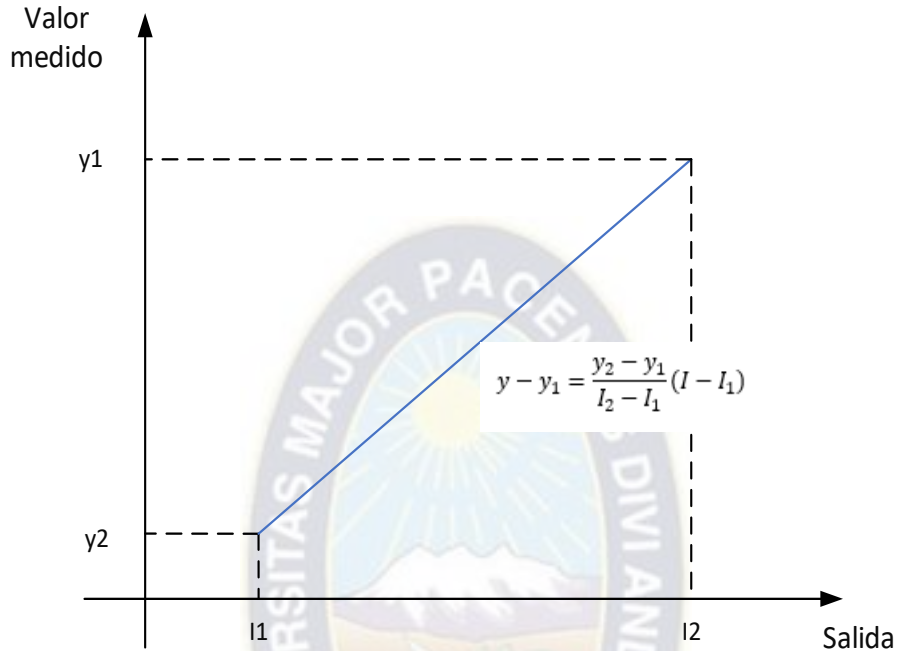


Figura 14. Escalamiento en un transmisor.

Fuente: Understanding 4-20 mA loops [9].

Usualmente la salida de un transmisor suele ser fija por ejemplo 4 mA – 20 mA o 0 mA – 20 mA; sin embargo, la salida  $y_1 - y_2$  suele ser configurable. Por ello es necesario hacer el escalamiento en los transmisores según las especificaciones o configuración del equipo.

#### 2.4. SEÑALES DE PULSOS EN MEDIDORES.

En la actualidad, los medidores de flujo líquido presentan una señal de salida de pulsos que indica la cantidad de volumen de fluido que pasa a través del medidor. La relación entre la cantidad de volumen de fluido y la cantidad de pulsos define la constante del medidor de flujo líquido "k". La expresión para la constante del medidor es:  $k = \frac{Npulsos}{Volumen}$ , en general las unidades de la constante del medidor son  $\left[\frac{pulso}{L}\right]$  o  $\left[\frac{pulso}{m^3}\right]$ . El valor de la constante de pulsos del medidor de flujo está indicado en su manual o en la placa de características del medidor. Para hallar el volumen de fluido que circula a través de un medidor se usa la Ecuación 2.

$$V = \frac{N_{pulsos}}{k} \quad (2)$$

*Ecuación 2. Relación volumen constante de pulsos.*

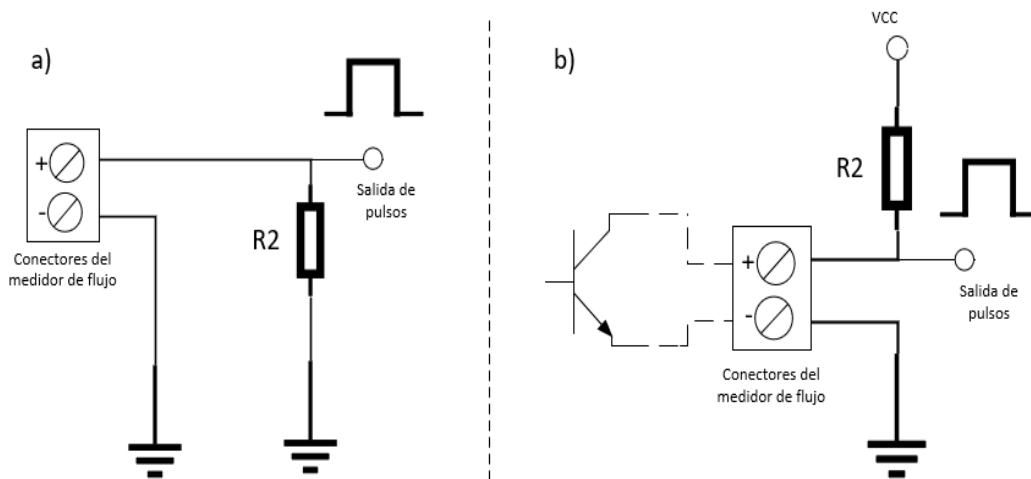
Dónde:

$V$ : Volumen que circula a través del medidor.

$N_{pulsos}$ : Número de pulsos del medidor.

$k$ : Constante de pulsos/volumen, propio del medidor.

Las salidas de pulsos de los medidores de flujo líquido pueden ser activas o pasivas. Una salida activa no requiere fuente de alimentación externa y se conecta directamente a una resistencia de carga. Una salida de pulsos pasiva, generalmente, es de colector abierto y requiere una fuente de alimentación y una resistencia para obtener la señal [17]. En la Figura 15 se muestra la conexión de una salida de pulsos a) activa y b) pasiva.



*Figura 15. a) conexión de salida de pulsos activa. b) conexión de salida de pulsos pasiva. Fuente: Manual del estándar de energía SC 10V [17].*

Para el caso de señales de tipo activa, los niveles de tensión están entre 0 V – 24 V. En el caso de señales de tipo pasiva los niveles de tensión estarán entre 0 V – VCC y la resistencia se selecciona de tal forma que se sature el transistor.

## 2.5. AISLAMIENTO.

Al trabajar con equipos industriales como transmisores, medidores de flujo y variadores de frecuencia a menudo se presentan interferencias electromagnéticas, ruido y corrientes de fugas. Estos fenómenos pueden ser producidos por la longitud de los cables, equipos no conectados a un sistema de

aterramiento y altas corrientes en la línea AC, derivando en: Voltajes de modo común altos y lazos de tierra.

Los voltajes de modo común son señales pequeñas de altas frecuencias que agregan componentes indeseables a la información (ruido). Estos voltajes pueden ser amplificados por el sistema de medida, ocasionando malas mediciones [24].

Los lazos de tierra ocurren cuando dos puntos conectados a masa presentan diferentes potenciales. Estos lazos pueden ocurrir en la conexión de equipos como transmisores no aislados en sistemas de medida. En la Figura 16 se muestra de forma esquemática este problema.

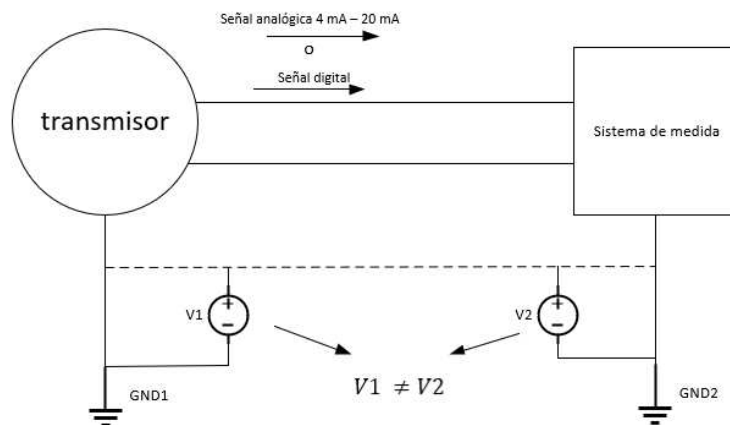


Figura 16. Lazo de tierra.

Fuente: *Back to Basic: The fundamentals of 4 – 20 mA Current loops* [24]

Para evitar dichos inconvenientes se usan circuitos de aislamiento galvánicos<sup>2</sup> de tipo magnético, capacitivo u óptico. Comúnmente, se usa el aislamiento óptico por disponibilidad de componentes y por su alta inmunidad al ruido eléctrico y magnético. Los componentes para aislamientos óptico generalmente están diseñados con fotodiodos y foto transistores, y se denominan optoacopladores. Estos dispositivos pueden ser lineales para señales analógicas o de saturación para señales digitales.

## 2.6. LabVIEW.

LabVIEW es el acrónimo de *Laboratory Virtual Instrument Engineering Workbench*, es un entorno de programación gráfico desarrollado por *National Instruments* (NI). La programación en LabVIEW se realiza mediante figuras, bloques y cables que conectan los bloques, esto ayuda a que se puedan crear

<sup>2</sup> El aislamiento galvánico consiste en separar dos circuitos y sus fuentes de alimentación, de esta forma se puede transferir señales sin que haya contacto entre los componentes.

aplicaciones de forma rápida y estructurada. En general, LabVIEW se utiliza para desarrollar programas de:

- sistemas de instrumentación.
- adquisición de datos.
- procesamiento de datos.
- procesamiento de señales.
- automatización de procesos de investigación y producción.

LabVIEW presenta facilidades en el uso de:

- Interfaces de comunicación. Por ejemplo, puerto serial RS232, esto hace posible la conexión con microcontroladores u otro tipo de hardware.
- Visualización y manejo de gráficos. Esto hace posible visualizar en tiempo real el estado de elementos de medición y/o actuadores en un proceso.
- Herramientas de interfaz para operadores. Se destacan botones, controles numéricos, indicadores numéricos o de texto entre otros, que se pueden utilizar para una operación amigable.
- Almacenamiento y procesado de datos.

Las características mencionadas hacen de LABVIEW una herramienta ideal en el desarrollo de sistemas de instrumentación y automatización de procesos. Así mismo, su naturaleza grafica hace que la programación sea más rápida y sencilla, brindando la posibilidad de crear interfaces amigables para un operador [25], [26] y [27].

### **2.6.1. INSTRUMENTOS VIRTUALES (VI).**

Originalmente LabVIEW estaba orientado a aplicaciones de control de instrumentos electrónicos en sistemas de instrumentación de manera virtual. Por esta razón, los programas de LabVIEW se llaman instrumentos virtuales (VI por sus siglas en ingles) y se guardan con la misma extensión (.vi).

Un VI es un elemento de programación de LabVIEW. El objetivo de un VI es emular un instrumento electrónico real con todo su funcionamiento interno a través de software [28]. De este concepto se asignan los componentes de un VI que son:

- Panel Frontal. Es la parte que ve el usuario u operador. Se asemeja a la pantalla de un instrumento real. En este componente hay botones, gráficos, entradas de control numérico y de texto. En la Figura 17 se puede ver el panel frontal de un VI en LabVIEW.

- Diagrama de bloques. Se asemeja al circuito interno de un instrumento real. Es donde se realiza la programación para funcionamiento y las conexiones entre bloques. En la Figura 18 se puede ver el diagrama de bloques de un VI en LabVIEW.

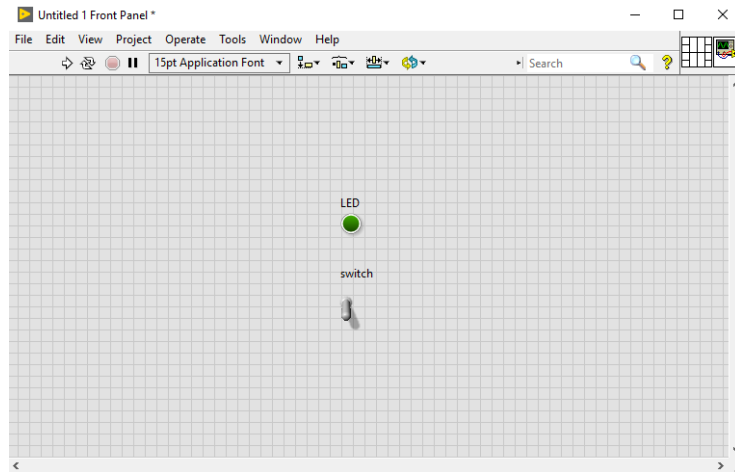


Figura 17. Panel frontal de un VI en LabVIEW

Fuente: LabVIEW Graphical Programming [27]

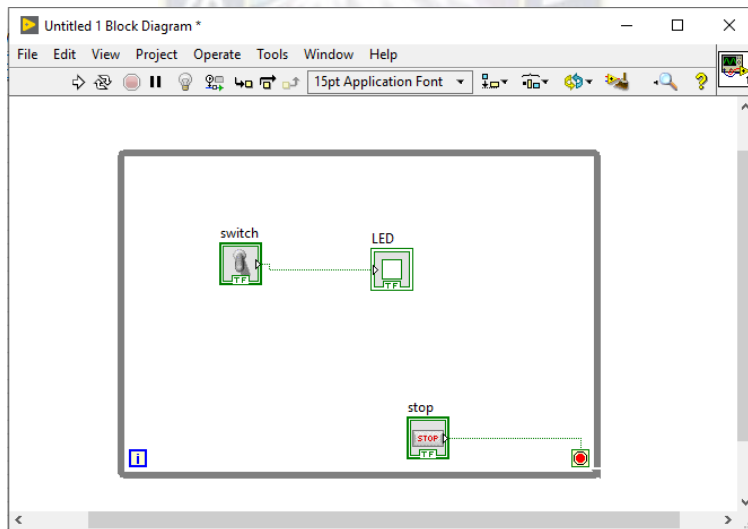


Figura 18. Diagrama de bloques de un VI en LabVIEW

Fuente: LabVIEW Graphical Programming [27]

Un VI puede ser usado dentro de otro VI entonces se denomina subVI. Un subVI sería similar a una función o subrutina en otros lenguajes de programación como por ejemplo C++ o BASIC. En LabVIEW se tiene una paleta de funciones que sería una colección de VI's que se pueden usar en el programa.



Además de las funcionalidades, en LABVIEW se puede usar diversas arquitecturas o estrategias de programación para el desarrollo de aplicaciones. Específicamente nos interesan dos arquitecturas: Máquinas de estado y Productor/Consumidor. Las arquitecturas mencionadas son utilizadas mayormente en el desarrollo de interfaces para sistemas de instrumentación.

### 2.6.2. MÁQUINAS DE ESTADOS.

Una máquina de estado es un modelo del comportamiento de algún sistema. Se compone por 3 conceptos que son: estado<sup>3</sup>, evento<sup>4</sup> y acción<sup>5</sup>. Las máquinas de estado suelen utilizarse para modelar la construcción de hardware y software.

El funcionamiento de una máquina de estado se puede describir de la siguiente forma: en algún instante de tiempo se está en un estado conocido “q”, se recibe una señal de entrada “a”, entonces, al siguiente instante de tiempo se pasa a otro estado conocido “r”. Lo descrito se puede definir en términos matemáticos como  $estado_{siguiente} = f(estado_{actual}, entrada_a)$ , siendo  $f$  la función de transición, que se representa mediante una tabla o diagrama de estado y eventos.

En LabVIEW una máquina de estado es una estrategia de programación utilizado para desarrollo e implementación de algoritmos. Las máquinas de estado son bastante utilizadas para aplicaciones de monitoreo, test, adquisición de datos y control. Esta estrategia de programación se puede utilizar cuando se deban responder a determinadas condiciones y controles accionados por un operador. Como ejemplo, en la Figura 19 se muestra una máquina de estados típica para sistemas de instrumentación [29], en este ejemplo los eventos que definen las transiciones son botones accionados por algún operador.

---

<sup>3</sup> En programación “Estado” es un término abstracto, que se refiere a una situación temporal en la que se está a la espera de alguna ocurrencia externa. Estado se refiere a una fracción de programa que este sujeto a cambios en el tiempo.

<sup>4</sup> “Eventos” son ocurrencias en el tiempo que tienen significancia en una fracción de código. Estas ocurrencias definen los cambios de estado en una máquina de estados.

<sup>5</sup> “Acciones” son la repuestas a los eventos, la máquina de estados define qué tipo de acción realizar de acuerdo al evento ocurrido.

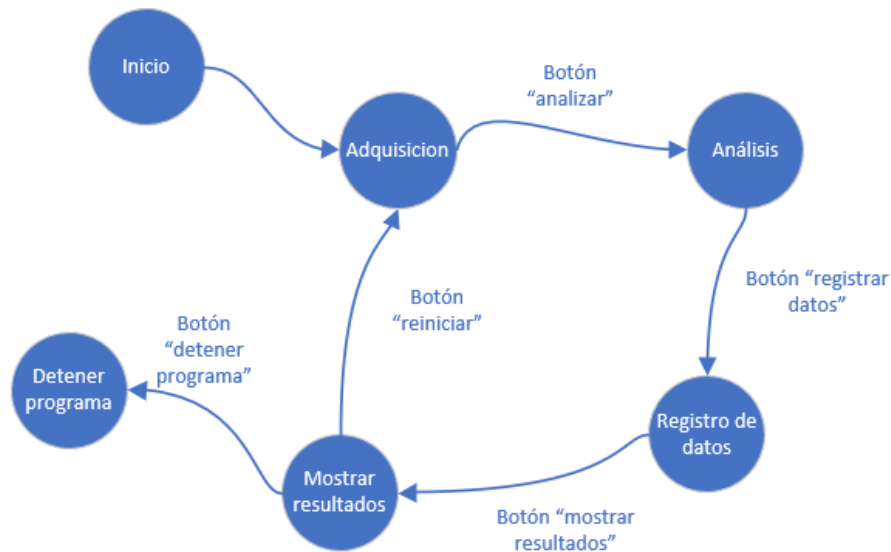


Figura 19. Máquina de estados típica para sistemas de instrumentación

Fuente: LabVIEW Core 1 y Core 2 [29]

Para la implementación de una máquina de estados se hace uso de dos estructuras propias del software: *While*, *Case*. La estructura *While* se encarga de que el programa se ejecute de forma constante y periódica. La estructura *case* es donde se encuentran los estados y las tareas asociadas a esos estados (acciones). Así mismo, para la transición entre estados se usa dos herramientas propias de LabVIEW: El *Shift register* (registro de desplazamiento) almacena el estado siguiente al cual debe dirigirse la máquina. La constante numérica "*Enum constant*" donde se almacenan todos los estados de la máquina.

Para la programación de una máquina de estados en LabVIEW, dentro un bucle *while* se pone la estructura *case*, se cablea con el registro de desplazamiento y la constante numérica. Cabe mencionar que esta programación se realiza en el diagrama de bloques del VI. Una máquina de estados básica y las estructuras y elementos que intervienen en esta programación se muestra en la Figura 20.

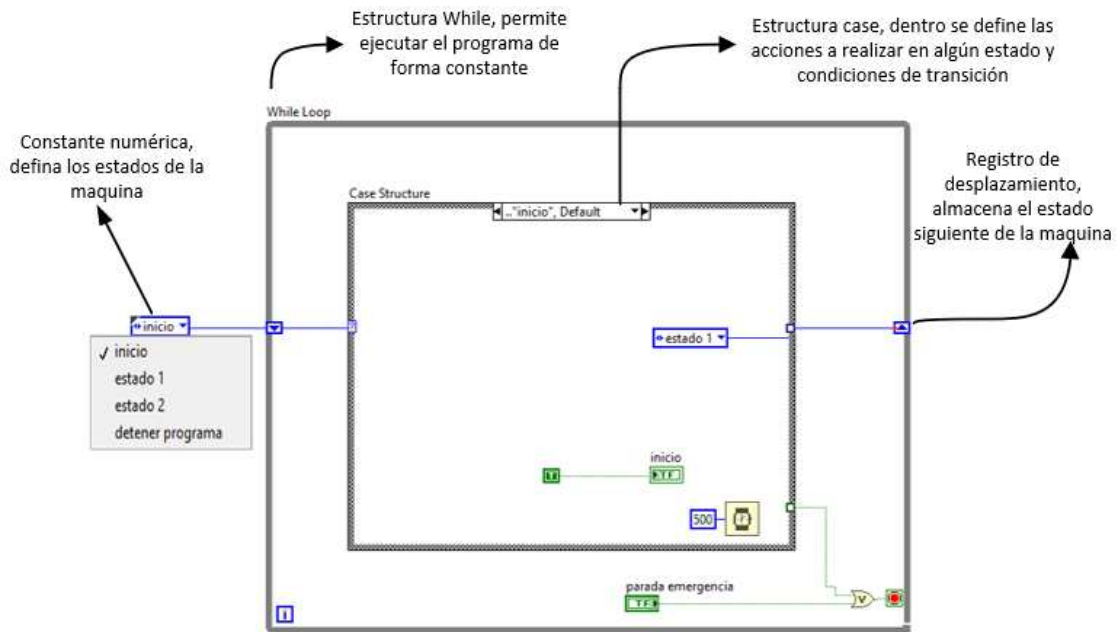


Figura 20. Programación de una máquina de estados en LabVIEW

Fuente: LABVIEW Core 1 y Core 2 [29]

Las máquinas de estado en LabVIEW pueden responder a ocurrencias o eventos externos. El accionamiento de botones o entradas de datos pueden representar eventos para la transición de estados en una máquina. La programación como máquinas de estados hace que el programa sea escalable y sostenible; por esta razón, se usa para sistemas de adquisición, monitoreo y control, y sistemas de instrumentación.

### 2.6.3. ARQUITECTURA PRODUCTOR/CONSUMIDOR.

Es una estrategia de programación en la cual varios bucles se ejecutan de forma paralela y simultánea. Generalmente, esta arquitectura se usa cuando se debe responder a varios controles de interfaz y adquirir o procesar datos al mismo tiempo. En esta arquitectura se tienen dos tipos de bucles: los que producen datos y los que consumen los datos producidos.

En la programación se utilizan dos estructuras While, una como bucle productor y la otra como bucle consumidor; y una cola de datos (queue). El bucle de producción obtiene datos desde algún hardware de adquisición y los pone en la cola (enqueue). El bucle consumidor procesa, visualiza o almacena los de la cola (dequeue). Puede utilizarse este tipo de programación si se requiere adquirir datos y visualizarlos mientras se responde a los comandos de un operador [30]. La Figura 21 muestra el esquema de programación productor consumidor.

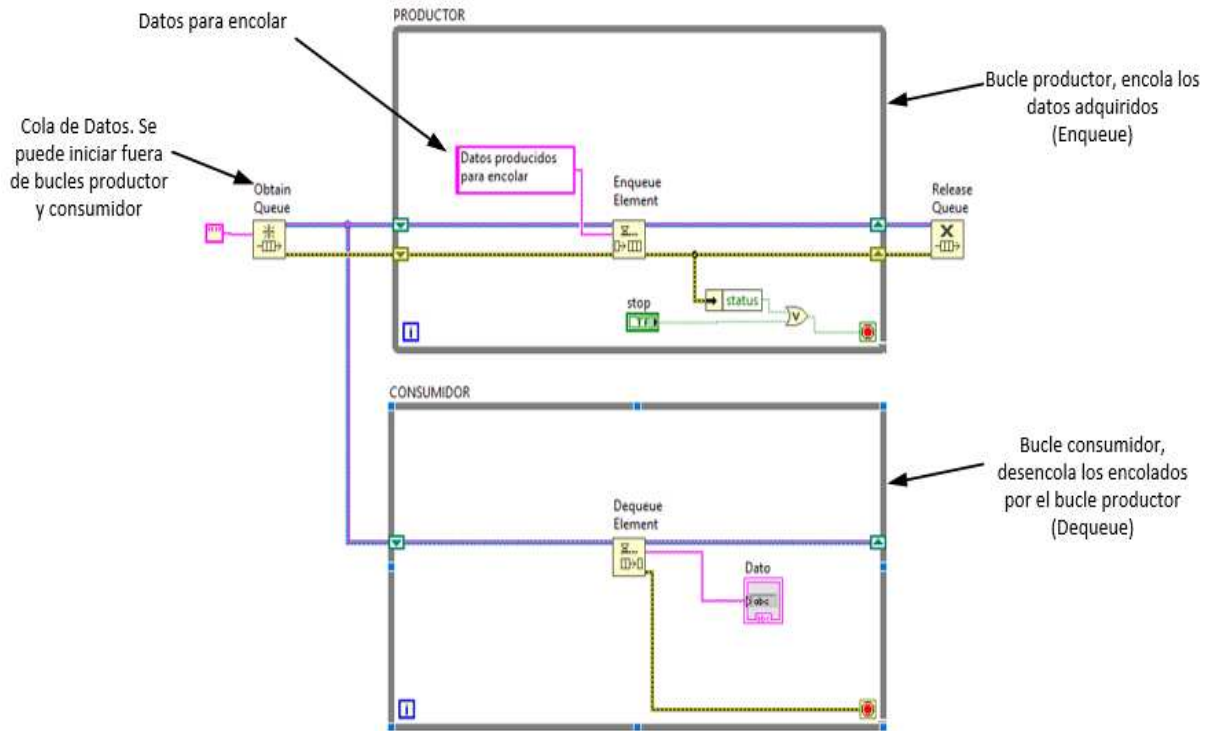


Figura 21. Esquema de programación productor consumidor.

Fuente: Arquitectura productor consumidor [30]

Dentro de cada bucle productor y/o consumidor puede estar una máquina de estados. Dependiendo de las especificaciones del sistema se elegirá como combinar ambas arquitecturas. Por las características de esta arquitectura, en combinación con máquinas de estado se puede desarrollar interfaces de usuario amigables, escalables y sostenibles.

## 2.7. MICROCONTROLADOR dsPIC30F4013.

Los dsPIC4013 son microcontroladores<sup>6</sup> de 16 bits de la familia 30F (dsPIC30F) fabricados por la empresa Microchip<sup>7</sup>. Estos dispositivos combinan la arquitectura y facilidad de programación de los microcontroladores de 8 bits y las ventajas de velocidad de procesamiento de un DSP<sup>8</sup>.

Las características destacables de los dsPIC30F4013 son [31]:

- Voltaje de alimentación en el rango de 2.5 VDC a 5.5 VDC.

<sup>6</sup> Un microcontrolador es un circuito integrado programable, que cuenta con puertos de Entrada/salida y periféricos útiles en el desarrollo de hardware.

<sup>7</sup> Microchip es una empresa de Estados Unidos que fabrica componentes electrónicos como microcontroladores, memorias y semiconductores.

<sup>8</sup> DSP (*Digital Signal Processing*, procesador digital de señales) es un procesador digital cuyo hardware y arquitectura esta optimizado para aplicaciones de procesamiento de señales, es decir, para realizar operaciones matemáticas a altas velocidades.

- Arquitectura Harvard.
- Operación de hasta 30 MIPS (Millones de instrucciones por segundo).
- Hasta 33 fuentes de interrupción: 8 niveles de prioridad seleccionables por el usuario y 3 fuentes de interrupción externa.
- Longitud de palabra de control de 24 bits. Registros de datos y control con longitud de 16 bits
- Se encuentran gran variedad de periféricos en la familia 30F de dsPIC's, como ser temporizadores, conversor analógico digital, comunicación serial, comunicación I2C o SPI, entre otros.

El dsPIC30F4013 en su forma de encapsulamiento DIP (Dual in-line package) tiene con 40 pines. En la Figura 22 se muestra el diagrama de pines del dsPIC30F4013. Así mismo, en la

Tabla 1 se muestra un resumen de los periféricos disponibles para este microcontrolador.

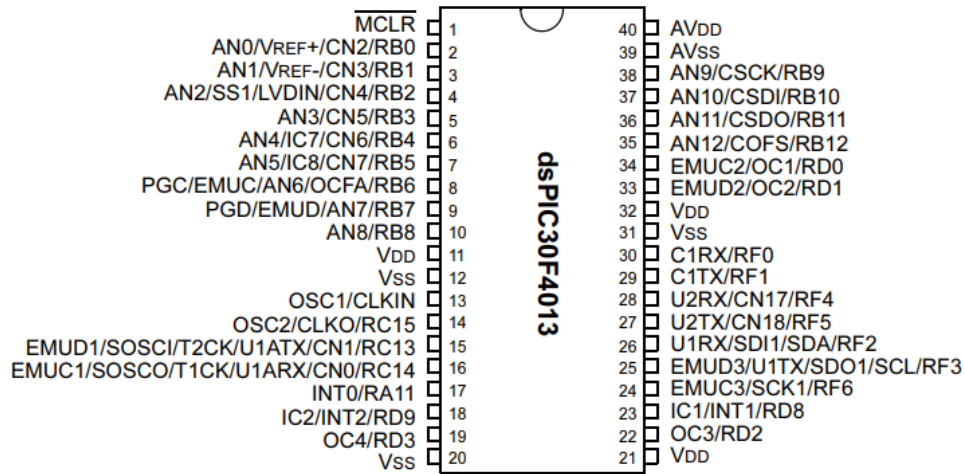


Figura 22. Diagrama de pines del dsPIC30F4013.

Fuente: Hoja de datos dsPIC30F4013 [32].

Pines	Temporizadores (Timer) 16-bit	Salida PWM/Comparación	Conversor A/D 12 bit	UART	SPI	I2C	CAN
40/44	5	4	13	2	1	1	1

Tabla 1. Periféricos del dsPIC30F4013

Fuente: Hoja de datos dsPIC30F4013 [32].

Para la programación de estos microcontroladores de Microchip, la empresa brinda un IDE (Integrated Development Environment - Entorno integrado de desarrollo), que es MPLAB. La

programación en MPLAB se realiza en lenguaje C. Al igual que otros entornos de desarrollo MPLAB cuenta con librerías, sin embargo, su mayor beneficio es que se puede trabajar directamente con los registros del microcontrolador. En el anexo A se brinda una descripción sobre los periféricos y registros de control del dsPIC30F4013.





# 3. Capítulo 3 - Marco de Referencia.

## 3.1. MÉTODO DE CALIBRACION POR COMPARACIÓN.

La calibración de medidores de flujo líquido se realiza comparando la cantidad de fluido indicada por un DUT con la cantidad de fluido indicada de un medidor patrón. La comparación se realiza en términos de cantidad de volumen de fluido o líquido. El DUT y el medidor patrón se colocan en la misma línea de tuberías y se hace circular la misma cantidad de líquido en ambos. El objetivo del proceso de calibración es presentar un reporte que indique el grado de error e incertidumbre de medición del DUT.

En términos generales, el procedimiento para la calibración de medidores de flujo líquido por comparación con medidor patrón se puede resumir de la siguiente manera [15]:

- 1) Instalar el DUT y el medidor patrón en la misma línea, un diagrama esquemático se muestra en la Figura 23.
- 2) Verificar el rango y resolución de señales de salida de los instrumentos asociados: transmisores, indicadores de condiciones ambientales y pulsos de los medidores de flujo.
- 3) Identificar los puntos de calibración para el proceso, es decir, valores de caudal y volumen.
- 4) Encender los elementos de la línea como ser bomba, transmisores, válvulas y medidores de flujo.
- 5) Ajustar el caudal y hacer circular la cantidad de volumen según los puntos de calibración identificados en el punto 3)
- 6) Registrar los valores indicados por el IBP y el medidor patrón.
- 7) Registrar los valores indicados por los sensores de presión y temperatura instalados en la línea y las condiciones ambientales.
- 8) Repetir los pasos 5) al 7) para todos los puntos identificados en el paso 3)
- 9) Realizar el reporte de calibración con los datos obtenidos.

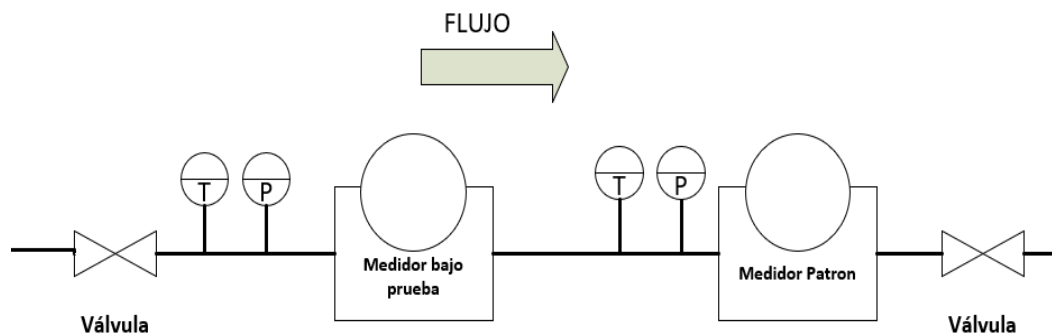


Figura 23. Diagrama esquemático de calibración por comparación.

Fuente: *Water Flow Meter Calibration with a Master Meter Method* [15].

La comparación de volumen entre el DUT y el medidor patrón se puede realizar de dos formas: observando la pantalla de los medidores; o a través de sus salidas de pulsos. En el caso de trabajar

con la pantalla de los medidores, se debe tomar una lectura inicial y una lectura final de volumen indicado, la diferencia entre estas lecturas indica la cantidad de líquido que circuló por el medidor en términos de volumen. Si se utiliza la salida de pulsos, se calcula el volumen mediante la constante de pulsos del DUT y el medidor patrón.

### 3.1.1.CALIBRACIÓN USANDO SEÑALES DE PULSOS.

Las señales de pulsos disponibles en los medidores de flujo líquido se pueden usar para realizar las mediciones de cantidad de volumen. Para comparar el volumen del DUT y el medidor patrón, se debe contar los pulsos emitidos por ambos equipos y utilizar la Ecuación 2 mostrada en el capítulo anterior. Para que se pueda realizar una correcta comparación se requiere utilizar un circuito de sincronización y conteo de pulsos [18]. En la Figura 24 se muestra un esquema para el método de calibración de medidores de flujo líquido utilizando las señales de pulsos.

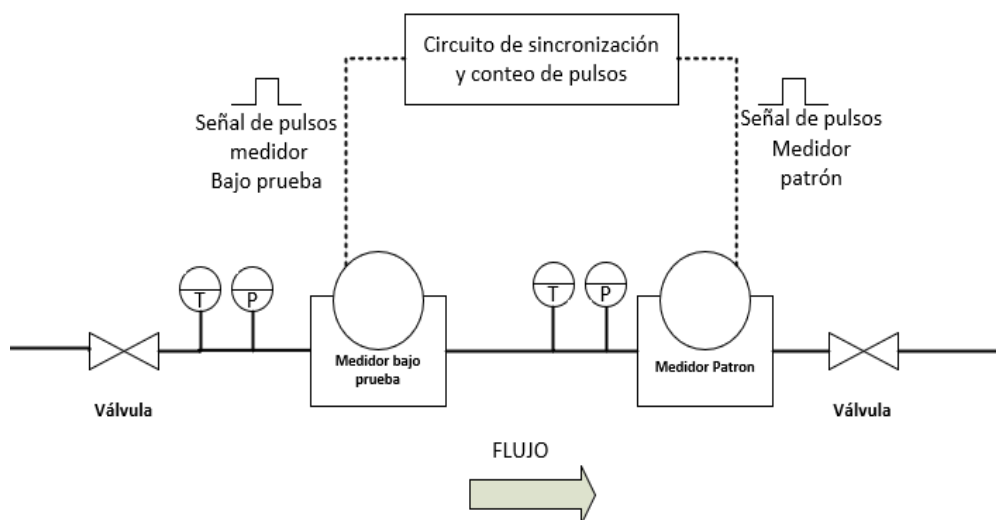


Figura 24. Calibración por comparación utilizando las señales de pulsos.  
Fuente: Curso teórico práctico – Metrología de Flujo [16].

La sincronización de pulsos se puede realizar de diversas formas, en el manual del medidor de energía SC10V que también tiene salidas de pulsos, se brinda una forma de conteo de forma tal que no se pierdan pulsos en el DUT. Para ello, mientras se cuentan N pulsos en el DUT se cuenta los pulsos del patrón como se muestra en la Figura 25.



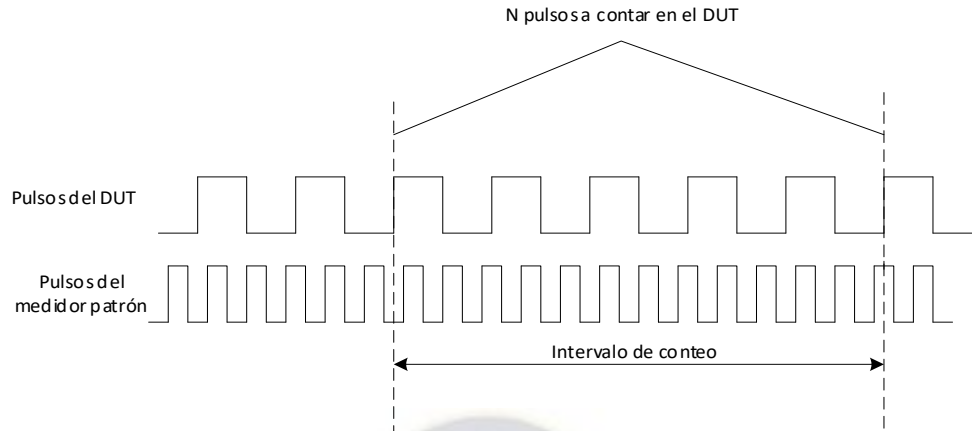


Figura 25. Sincronización de pulsos.

Fuente: Manual del estándar de energía SC 10V [17].

Entonces, el proceso de calibración se realiza de la siguiente manera:

- 1) Identificar los puntos de calibración, caudal y volumen deseados.
- 2) Ajustar el caudal de circulación en la línea
- 3) Programar una cantidad de pulsos  $N_{DUT}$  a contar, dicha cantidad de pulsos representa una cantidad de volumen mediante la constante  $k_{IBP}$ .
- 4) Habilitar el conteo de pulsos del DUT ( $N_{DUT}$ ) y del patrón ( $N_p$ ).
- 5) Realizar la cuenta en ambos medidores hasta llegar a los  $N_{IBP}$  pulsos programados en el paso 2)
- 6) Con los valores de pulsos contados en el DUT y el medidor patrón, obtener la cantidad de volumen de fluido indicado por cada medidor a través de sus constantes.
- 7) Registrar los valores indicados por los transmisores de presión y temperatura instalados en la línea. También, registrar la temperatura, humedad relativa y presión atmosférica del laboratorio.
- 8) Repetir los pasos 2) al 5) para cada valor de volumen y caudal definidos en el paso 1)
- 9) Reportar el error y la incertidumbre de medición.

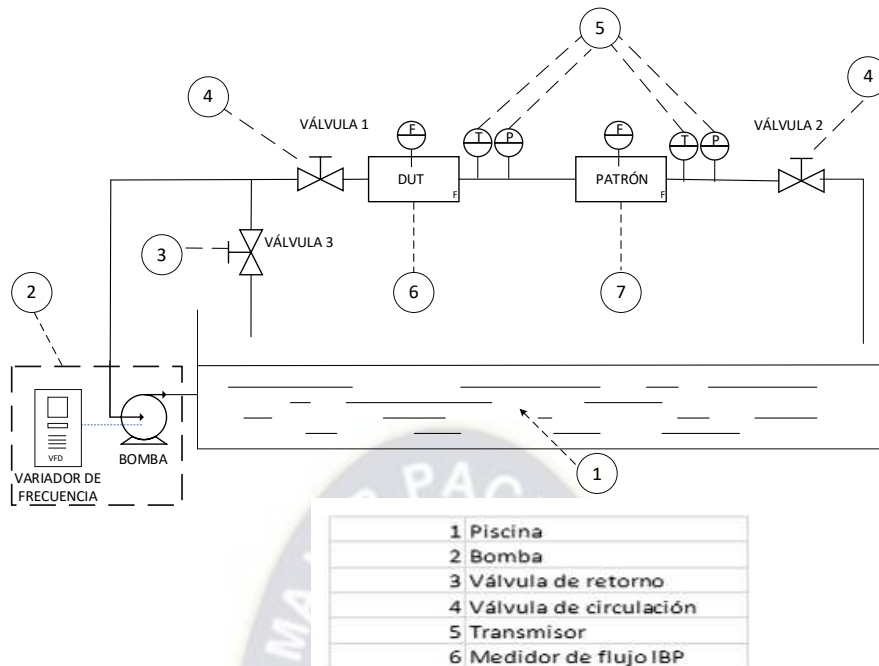
Como se mencionó en los pasos anteriores, en el proceso de calibración es necesario registrar los valores indicados por los medidores de flujo líquido (DUT y patrón), transmisores de temperatura y presión instalados en la línea y las condiciones ambientales de laboratorio. Los datos que se deben registrar se muestran en la Tabla 2.

N	Flujo	CONDICIONES AMBIENTALES			MEDIDOR PATRON			MEDIDOR BAJO PRUEBA		
		<i>Caudal</i> [m3/hora]	<i>T_amb</i> [°C]	<i>HR</i> [%HR]	<i>P_amb</i> [kPa]	<i>N_p</i>	<i>P_p</i> [kPa]	<i>T_p</i> [°C]	<i>N_DUT</i>	<i>P_DUT</i> [kPa]
1										
2										
3										
4										

*Tabla 2. Toma de datos en el proceso de calibración por comparación.  
Fuente: Curso teórico práctico – Metrología de Flujo [16]*

### **3.2. BANCO DE CALIBRACION DE METROLAB**

En la , Figura 26 se muestra los elementos de la planta para calibración de medidores de flujo líquido. Tal como se ve, se tienen 7 elementos conectados por tuberías: La piscina, que almacena el líquido para la circulación en la línea. La bomba, que se encarga de hacer circular el líquido por las tuberías a un caudal determinado. La válvula de retorno y las válvulas de circulación, que determinan el sentido de circulación. Los transmisores, instalados para medir la temperatura y presión en la línea. El medidor bajo prueba y el medidor patrón, realizan las mediciones de volumen de circulación para la comparación n la calibración.



, Figura 26. Elementos de la planta de calibración.

Fuente: Elaboración propia.

Con las válvulas de circulación (4) abiertas y la válvula de retorno (3) cerrada, al activar la bomba, el líquido contenido en la piscina circula por los medidores (IBP y patrón) y vuelve a la piscina de forma cíclica; entonces, se adquieren los datos los medidores y transmisores instalados.

Como se puede ver en la , Figura 26 para la medición de temperatura y presión en la línea se tienen cuatro (4) transmisores instalados: dos (2) al lado del DUT (presión y temperatura) y dos (2) al lado del medidor patrón (presión y temperatura). Para la medición de caudal se puede utilizar el medidor patrón.

### 3.2.1.EQUIPOS USADOS POR METROLAB.

A continuación, se detallan los equipos de medición instalados en la planta de calibración de medidores de flujo líquido de METROLAB.

Como patrón se usa un medidor de tipo electromagnético KHRONE IFC 110. Este equipo que dispone de salidas de pulsos para el volumen y corriente para el caudal medido. En la Figura 27 se muestra este medidor y en la Figura 28 se muestra la conexión para obtener esta señal de pulsos de tipo activa o pasiva [33].



Figura 27. Medidor de flujo patrón KHRONE IFC110.

Fuente: Instrucciones de instalación y funcionamiento ALTOFLUX IFC 110 [33]

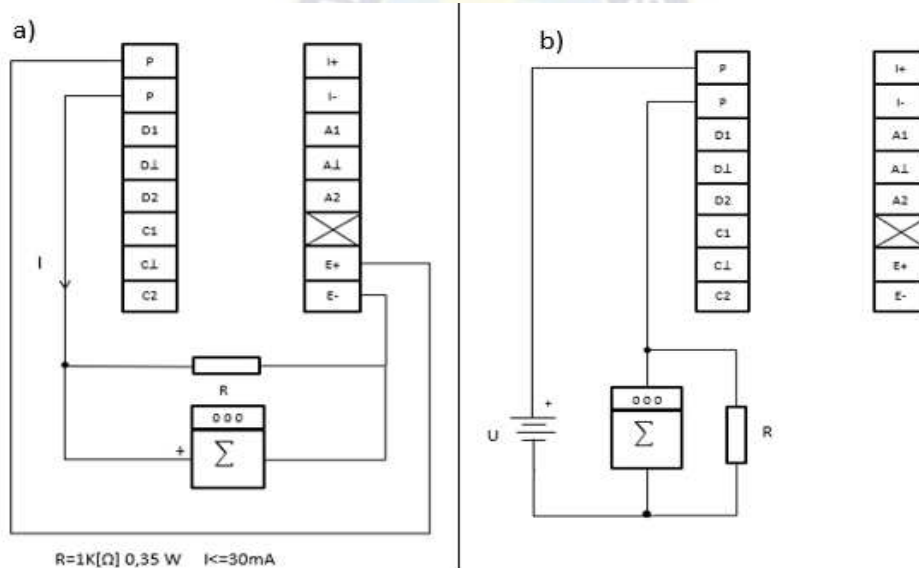


Figura 28. Conexión de pulsos activos del medidor patrón a) de tipo activo b) de tipo pasiva.

Fuente: Instrucciones de instalación y funcionamiento ALTOFLUX IFC 110 [33].

Para la medición de temperatura, presión y caudal en banco de calibración se dispone de los siguientes transmisores:

- **Temperatura:** transmisor ENDRESS HAUSER modelo TMT180 con sensor Pt100, este se puede ver en la Figura 29 a).
- **Presión:** transmisor YOKOGAWA modelo EJA530E con sensor piezoeléctrico, este se puede ver en la Figura 29 b).
- **Caudal:** Medidor de flujo patrón ENDRESS HAUSER IFC 110 con sensor electromagnético, este se puede ver en la Figura 29 c).

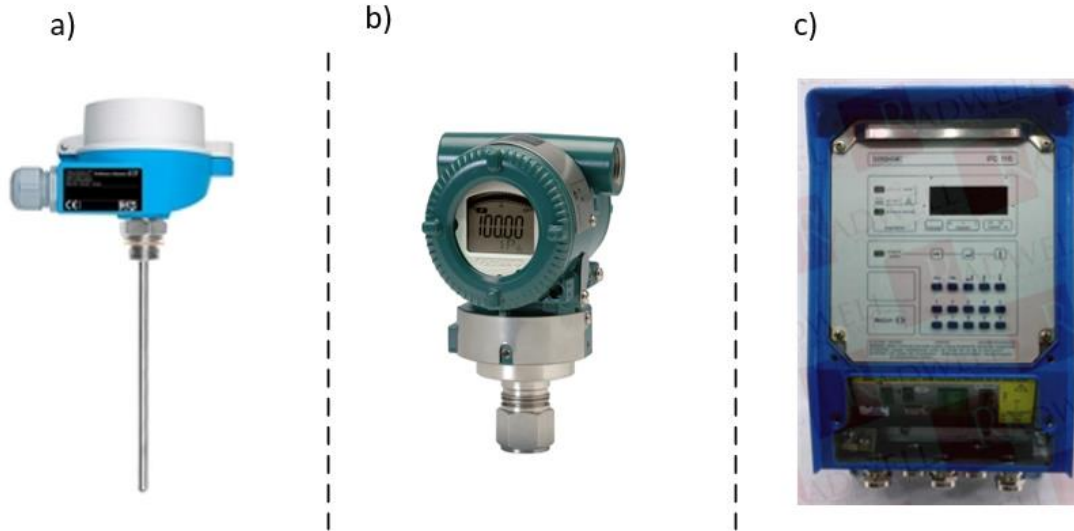


Figura 29. a) Transmisor de temperatura b) Trasmisor de presión c) Transmisor de caudal.  
Fuente: [33], [34], [35].

En la Tabla 3 se resume la marca, modelo, tipo de conexión, señal de salida, rango de medición, región de operación y tiempo de respuesta de los transmisores instalados en el bando de calibración de METROLAB [34], [35].

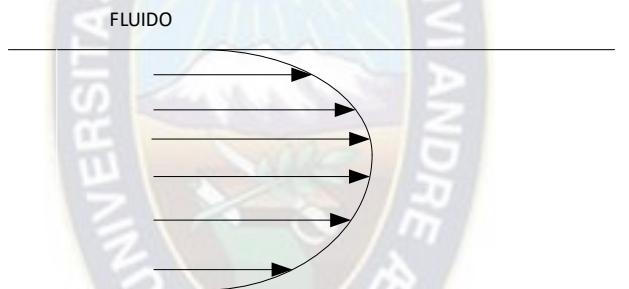
Magnitud Medida	Marca/ Modelo	Conexión	Señal de salida	Rango de medición	Región de operación	Tiempo de respuesta
Temperatura	Endress Hausser/TMT180	2 hilos	4 mA - 20 mA	5 °C - 30 °C	$0 \leq R \leq \frac{U - 10}{0.022} : [\Omega]$ $10 \leq U \leq 35 : [VDC]$	1 s
Presión	Yokogawa/EJA530E	2 hilos	4 mA - 20 mA	0 kPa - 400 kPa	$250 \leq R \leq \frac{U - 10,5}{0.0244} : [\Omega]$ $16,6 \leq U \leq 42 : [VDC]$	45 ms
Caudal	KHRONE/ IFC110	4 hilos	4 mA - 20 mA	$0 \text{ m}^3/\text{h}$ $- 100 \text{ m}^3/\text{h}$	$15 \leq R \leq 500 : [\Omega]$ $U = 220 : [VAC]$	0,2 s

Tabla 3. Resumen de características de los transmisores.  
Fuente: [33], [34], [35].

### 3.3. FACTORES DE INFLUENCIA EN LAS MEDICIONES.

Cuando se realizan mediciones con un medidor de flujo los valores obtenidos pueden presentar desviaciones; es decir, no presentar una medición confiable. Existen variables influyentes en las mediciones con un medidor de flujo. Dichas variables de influencia pueden ser de la instalación y de principio de medida. En cuanto a la instalación se puede distinguir que influye el diámetro y largo de tuberías, posición del medidor, entre otros. En cuanto al principio de medición las variables que influyen son [7]:

- Perfil de flujo en el líquido circulante. Idealmente distribución de velocidad del fluido debería ser más rápida en el centro y más lenta en los costados de la tubería como se muestra en la figura. Esta distribución de velocidades puede verse afectada por el largo de tuberías o apertura y cierre de válvulas que generan regímenes transitorios en la circulación. Una recomendación de buena práctica es que las mediciones se realicen en estado estacionario, es decir cuando las velocidades del fluido están con una distribución adecuada.



*Figura 30. Perfil ideal de flujo en circulación.*

*Fuente: Calibration Technical Information [7]*

- Temperatura y presión del líquido. Los fabricantes de medidores indican las condiciones de presión y temperatura del líquido circulante para una buena medición. Es decir, establecen intervalos donde se puede garantizar la fiabilidad de los valores obtenidos. Cuando el líquido circulante no se encuentra en los intervalos declarados por los fabricantes se puede aplicar un factor de corrección en la medición. Entonces una buena práctica para la medición es monitorear la presión y temperatura del líquido y en caso de ser necesario aplicar la corrección [3].
- Condiciones ambientales. Al igual que en el punto anterior, cambios en las condiciones ambientales de temperatura, Humedad relativa y presión ambiente pueden generar variaciones en la medición. Es una buena práctica registrar las condiciones ambientales en el momento de una medición [3].



### 3.4. ANALISIS DE CONFIABILIDAD DE LAS MEDICIONES.

Como se mencionó en el punto 1.2 del capítulo 1 el objeto de los laboratorios de calibración es garantizar la confiabilidad en las mediciones. Al respecto, La confiabilidad de las mediciones se define como el “Grado en que un experimento, ensayo o procedimiento de medición arroja resultados exactos en ensayos repetidos” [11].

El termino exactitud está relacionado a la consistencia y repetibilidad de las mediciones. La consistencia representa la coherencia entre las mediciones obtenidas. Por ejemplo, entre un conjunto de mediciones {12,1; 12,2; 12,4; 28,0; 12,6} el valor 28,0 es atípico y no presenta coherencia con el resto de valores. Ahora bien, valores inconsistentes pueden generarse debido a que el equipo de medición no está en condiciones adecuadas o que el sistema de medición no es adecuado [13].

La repetibilidad indica el grado de cercanía de los valores de mediciones realizadas bajo las mismas condiciones. Es claro que cuando se realizan mediciones bajo las mismas circunstancias se espera que los valores obtenidos sean los mismos o al menos no tengan una variación considerable. La razón de falta de repetibilidad puede ser debida a malas condiciones del equipo o sistema de medición [12].

La consistencia y repetibilidad de las mediciones se pueden evaluar de forma cuantitativa. A continuación, se detalla cómo realizar dichos cálculos.

#### 3.4.1. ANALISIS DE CONSISTENCIA.

El análisis de consistencia para un sistema de medición se realiza teniendo en cuenta ciertas condiciones que son:

- Se toman  $n$  mediciones por una cantidad de operadores  $N$  donde  $N > 1$ .
- Los operadores deben seguir el mismo procedimiento para las mediciones.
- El equipo con el que se realizan las mediciones es el mismo y se encuentra en buen estado.

Bajo esta circunstancia, los pasos para verificar la consistencia son:

1. Analizar la coherencia de las mediciones con la prueba de  $k$  de Mandel. Para ello se realizan las siguientes tareas [13]:

- a) Calcular la desviación estándar en las mediciones de cada operador con la Ecuación 3.

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \dots \dots (3)$$

*Ecuación 3. Desviación estándar.*

- b) Calcular el valor de  $k$ , con la

- c) Ecuación 4, para cada las mediciones de cada operador.

$$k = \frac{S_i \sqrt{N}}{\sqrt{\sum_{i=1}^N S_i^2}} \dots \dots (4)$$

*Ecuación 4. k de Mandel.*

- d) Calcular el valor  $k$  con un valor critico tabulado  $k_{cri}$  (Ver Anexo D). Si se cumple  $k < k_{cri}$  para cada una de las desviaciones  $S_i$  (con  $i = 1, 2, \dots, N$ ) entonces los datos son consistentes.

2. Analizar si existen valores atípicos mediante la prueba de C de Cochran's. Para lo cual se realizan las siguientes tareas:

- a) Calcular la desviación estándar en las mediciones de cada operador con la Ecuación 3.  
b) Calcular  $C$  para cada resultado con la Ecuación 5.

$$C = \frac{S_i^2}{\sqrt{\sum_{i=1}^N S_i^2}} \dots \dots \dots (5)$$

*Ecuación 5. C de Cochran.*

- c) Comparar el valor de  $C$  con un valor critico tabulado  $C_{cri}$  (Ver Anexo C). Si se cumple que  $C < C_{cri}$  no existen valores atípicos debido al sistema de medición y los datos se aceptan.

En caso de que los valores sean coherentes y no sean atípicos, se puede afirmar que las mediciones obtenidas con el sistema de medición brindan valores consistentes.

### 3.4.2. ANALISIS DE REPETIBILIDAD.

El análisis de repetibilidad para un sistema de medición se realiza teniendo en cuenta ciertas consideraciones que son:

- Se toman  $n$  mediciones por una cantidad de operadores  $N$  donde  $N > 1$ .
- Los operadores deben seguir el mismo procedimiento para las mediciones.
- El equipo con el que se realizan las mediciones es el mismo y se encuentra en buen estado.

Bajo estas condiciones se debe realizar las siguientes tareas [14]:

- a) Calcular la desviación estándar experimental en las mediciones de cada operador con la Ecuación 3.



- b) Cuando el valor de mediciones realizadas por cada operador ( $n$ ) es menor a 30, se calcula la repetibilidad ( $r$ ) con la siguiente formula:

$$r = t_{95} * \sqrt{2} * S \dots \dots (6)$$

*Ecuación 6. Repetibilidad.*

*Donde  $t_{95}$  es el valor de  $t$  de Student para una cobertura del 95 % (Ver anexo D)*

- c) Comparar el valor de repetibilidad obtenido ( $r$ ) con el límite de repetibilidad  $L_r$ ; este último valor se encuentra en las especificaciones técnicas del equipo con el que se realizan las mediciones. En caso de que la repetibilidad calculada  $r$  sea menor al límite de repetibilidad  $L_r$  se acepta las mediciones.

$$r < L_r$$



## 4. Capítulo 4 - Desarrollo del Sistema.

### 4.1. ANÁLISIS DE REQUERIMIENTOS.

Se describe el alcance del sistema, en cuanto a su funcionalidad y restricciones. Inicialmente se expone la visión proyectada del sistema, prosiguiendo con una exposición de las funcionalidades que contempla.

#### 4.1.1. VISIÓN DEL SISTEMA.

El sistema de asistencia tiene como finalidad constituirse una herramienta disponible y eficiente en las tareas de toma de medidas durante el proceso de calibración de medidores de flujo líquido, que realiza la empresa METROLAB. Para ello, Conforme los objetivos expuestos en el capítulo 1, y los marcos de referencia establecidos en el capítulo 2, el sistema automatiza la toma de mediciones de los medidores y transmisores instalados en el banco de calibración, permitiendo aplicar buenas prácticas como ser mediciones en régimen permanente y verificación de variables de influencia.

##### 4.1.1.1. ESQUEMA GENERAL.

Inicialmente, se brinda una idea general del sistema mediante el esquema que se muestra en la Figura 31, se puede observar que el sistema se compone por 4 unidades: Adquisición, Procesamiento y comunicación, Almacenamiento y la Interfaz Hombre Maquina (*Human Machine Interface – HMI*).

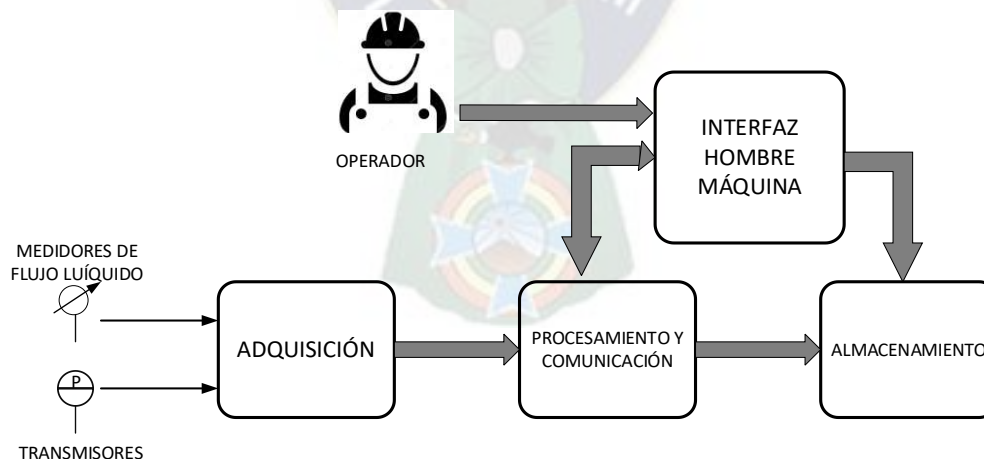


Figura 31. Esquema general del sistema.

Fuente: Elaboración propia.

La **Unidad de Adquisición** realiza el acondicionamiento y lectura de las señales de los medidores de flujo líquido y transmisores instalados en el banco de calibración. La **Unidad de Procesamiento**

realiza las operaciones necesarias sobre las señales adquiridas para su interpretación, visualización y almacenamiento. La **Unidad de Almacenamiento** genera reportes con los datos adquiridos, que son los necesarios para la calibración de medidores de flujo líquido. La **HMI** es el nexo entre el operador y el sistema como tal. Las unidades mencionadas deben contar con características funcionales propias para cumplir con la automatización en la toma de medidas.

#### **4.1.2. FUNCIONALIDADES.**

El Sistema debe contar con las siguientes funciones.

- a) Adquisición. con la finalidad de automatizar la toma de medidas, se concibió, inicialmente, la etapa que permita obtener señales con niveles y características adecuadas para su procesamiento. Las señales que se deben adquirir son temperatura, presión y pulsos de los medidores como se detalla en el punto 3.1.1 del marco referencial.

Se considera que el personal que realiza las tareas para la toma de mediciones no necesariamente tiene conocimientos en electrónica o instrumentación, por ello se considera que el sistema:

- i. Debe disponer entradas para conexiones de las señales de los transmisores sin necesidad de circuitos externos adicionales.
  - ii. Debe disponer entradas para conexión de las señales de los medidores de flujo sin necesidad de circuitos externos adicionales.
  - iii. Debe realizar el acondicionamiento de cada una de las señales analógicas y de pulsos de los transmisores y medidores respectivamente.
- b) Procesamiento.

Teniendo en cuenta las recomendaciones del punto 3.3 del Marco de Referencia, se considera que el sistema:

- i. Debe realizar mediciones en estado estacionario de la circulación de líquido, sin necesidad de segmentar o cortar la circulación del líquido.
- ii. Debe obtener mediciones de los instrumentos instalados en el banco de calibración siguiendo las recomendaciones de los fabricantes para buenas mediciones.

- iii. Debe realizar un escalamiento adecuado de las señales obtenidas. Esto con el fin de una adecuada interpretación.
- c) Interfaz Hombre Maquina. La interfaz hombre máquina permite realizar el monitoreo y configuración del proceso de toma de datos, por parte del operador. Debe contar con las siguientes características:
- i. Debe mostrar los valores obtenidos por los transmisores con indicadores numéricos y/o de texto.
  - ii. Debe presentar un esquema de distribución de los elementos en la planta de calibración para interpretación de los datos del proceso.
  - iii. Debe mostrar gráficos de las variables de influencia en las mediciones. Se considera las variables mencionadas en el punto 3.3 del marco de referencia.
  - iv. Debe permitir la configuración (inicio y final) de las mediciones. Para ello debería contener entradas numéricas, de texto y botones.
  - v. Debe contar con entradas numéricas para ingresar los datos de condiciones ambientales en el proceso de calibración. Dado que también son variables de influencia en la medición como se vio en el punto 3.3.
  - vi. Debe realizar el despliegue de resultados parciales durante las mediciones, de forma que se realice ajuste en el DUT en caso de ser necesario.
- d) Almacenamiento. Este componente permite:
- i. Generar un archivo donde se presentan los datos adquiridos y que son necesarios para la calibración, con la información capturada en las unidades anteriores.
  - ii. Registrar información complementaria del DUT como ser marca, modelo, número de serie e identificación interna.

#### **4.1.3. ATRIBUTOS DEL SISTEMA.**

Se observan algunas necesidades en cuanto al sistema para que pueda cumplir con sus funcionalidades, las cuales se detallan a continuación:

- El banco de calibración cuenta con elementos que pueden generar problemas de ruido, lazos de tierra e interferencias, por lo cual en el acondicionamiento de las señales el sistema:
  - i. Debe contar con una etapa de aislamiento.
- Para poder realizar mediciones en circulación del líquido, el sistema:

- ii. Debe realizar las mediciones contando los pulsos de los medidores
  - iv. Debe obtener los datos de las señales de los transmisores y medidores de flujo. Se deben seguir las recomendaciones de los fabricantes. Siguiendo las recomendaciones del punto 3.1.1 del marco de referencia.
- Para asegurar buenas mediciones se considera que el sistema:
    - iv. Debe seguir las recomendaciones de los fabricantes en cuanto a conexión, región de operación y tiempo de respuesta de transmisores y medidores.

#### 4.2. ARQUITECTURA Y DISEÑO.

Con base en los requerimientos definidos anteriormente y en la descripción de equipos en el punto 3.2.1, se identifica como entradas al sistema: las señales analógicas de los transmisores, las señales de pulsos de los medidores, los valores de condiciones ambientales. La salida del sistema es un reporte con los datos adquiridos.

Dado el esquema general (Figura 31) y considerando las entradas y salidas del sistema; inicialmente, cada una de las unidades se distribuye entre dos (2) componentes: Hardware y una Interfaz Gráfica de Usuario (*Graphical User Interface - GUI*). El **Hardware** contiene la unidad de adquisición y comunicación. La **GUI** contiene las unidades de procesamiento, HMI y almacenamiento. En la Figura 32, se muestra la distribución de las unidades en los componentes mencionados.

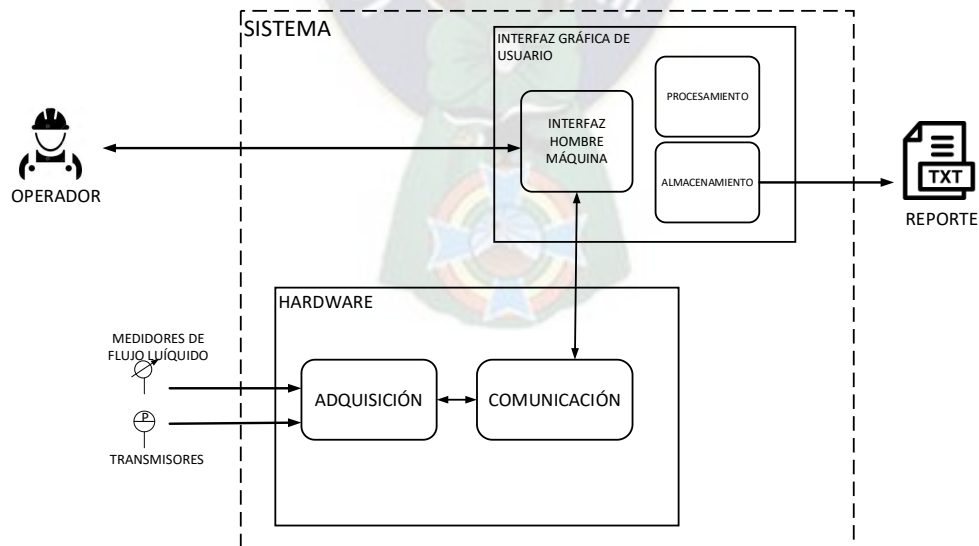


Figura 32. Distribución de unidades en los componentes del sistema.  
Fuente: Elaboración propia.

La unidad de adquisición realiza el acondicionamiento y la lectura de las señales que provienen de los transmisores y medidores. Entonces el hardware queda compuesto por elementos de acondicionamiento, lectura y comunicación como se muestra en la Figura 33. los dos últimos elementos (lectura y comunicación) quedan integrados en un tercer componente que es un dispositivo programable (DP).

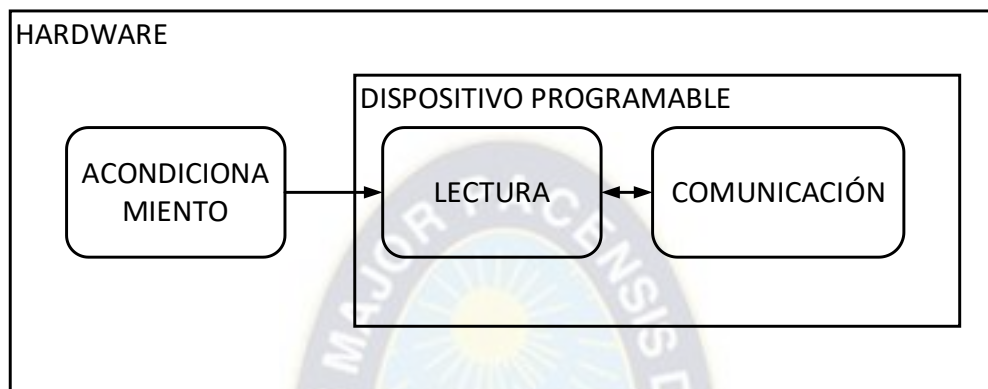


Figura 33. Elementos del Hardware.

Fuente: Elaboración propia.

Los componentes del sistema (Hardware, DP y la GUI) y sus elementos, se organizan para constituir un sistema de adquisición basado en computador, donde la GUI se ejecuta en la computadora. La arquitectura final para el sistema se muestra en la Figura 34.

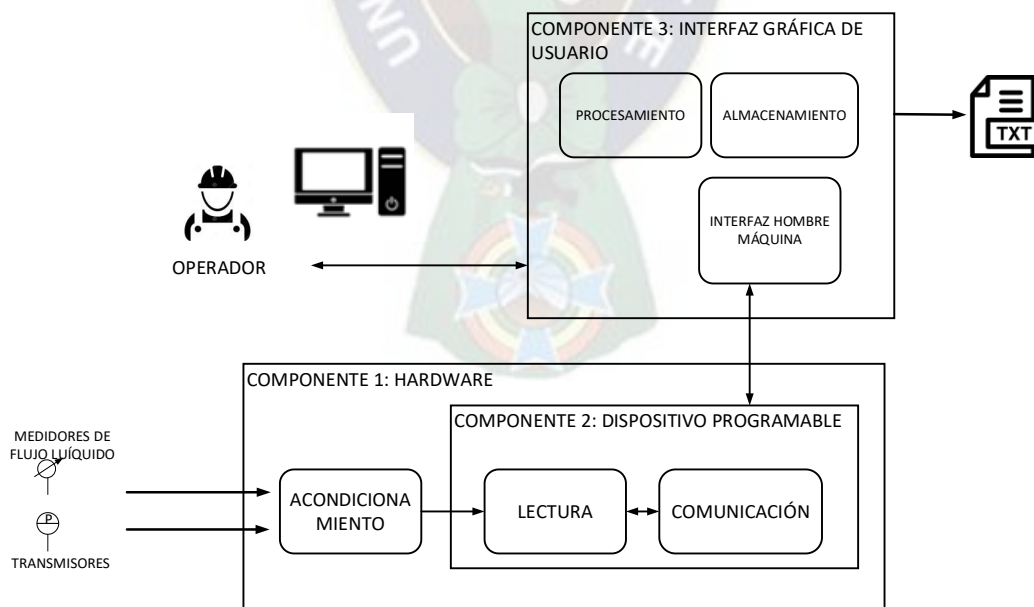


Figura 34. Arquitectura del sistema.

Fuente: Elaboración propia.

A continuación, se realiza el diseño de cada componente es decir Hardware; el programa para el DP; y la GUI.

#### 4.2.1. HARDWARE.

El hardware refiere a tarjetas de circuito impreso, donde se conectan físicamente Los terminales de los transmisores y señales de pulsos de los medidores, su fin es:

- Acondicionamiento de las señales analógicas de transmisores,
- Acondicionamiento señales de pulsos de los medidores,
- Contener circuitos necesarios para el funcionamiento del DP.

En la Figura 35 se muestra las entradas y salidas del hardware.

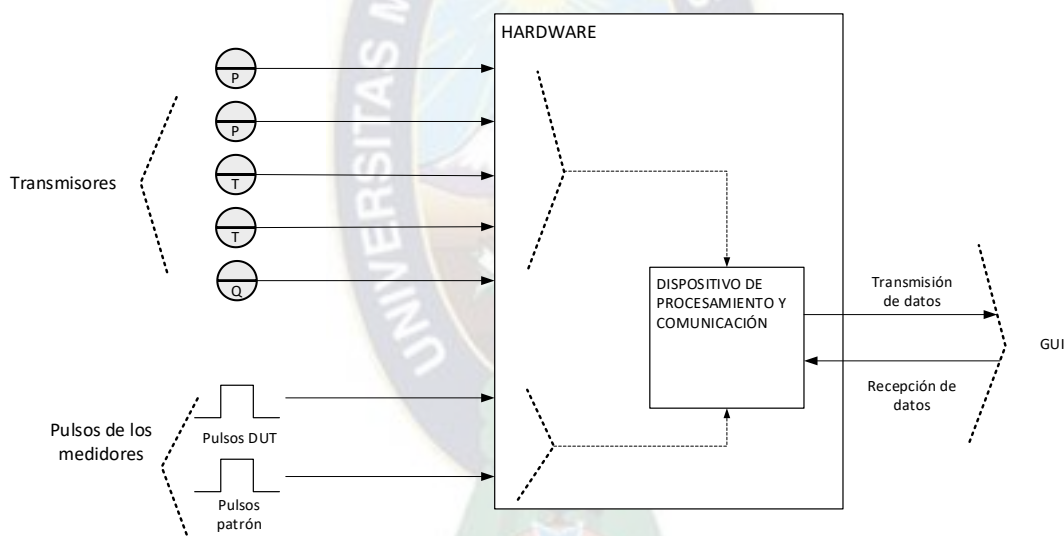


Figura 35. Entradas y salidas internas del hardware.

Fuente: Elaboración propia.

Como entradas se tiene

- Señales analógicas de los transmisores de la línea.
- Señales de pulsos de los medidores DUT y patrón.
- Señal de datos recibida en comunicación con la interfaz gráfica de usuario.

Como salida se tiene:

- Señal de datos transmitida en comunicación con la interfaz gráfica de usuario.

Como se describió en la sección 3.2 para medir la temperatura y presión en la línea se tiene 4 transmisores de 2 hilos. Para medir caudal se utiliza el transmisor integrado en el medidor patrón que



es de 4 hilos. Las señales de los medidores pueden ser de tipo activo o pasivo, para su conteo. También, para la comunicación se tienen 2 pines, uno de recepción y otro de transmisión de datos.

El hardware integra todos los componentes necesarios para conversión (corriente a voltaje o viceversa), aislamiento, filtrado, adaptación de nivel y otros que requieran las señales de entrada. También, contiene al dispositivo de procesamiento y los elementos necesarios para su funcionamiento.

Con lo descrito se considera las siguientes especificaciones para el hardware:

- Mínimamente 5 entradas para transmisores de 4 mA - 20 mA.
- Entradas de transmisores conmutables para 2 o 4 hilos.
- Mínimamente 2 entrada para señales de pulsos, para el DUT y patrón,
- Entradas para señales de pulsos conmutables para tipo activo o pasivo,
- Niveles de tensión para señales de pulsos 0 V – 24 V.

Con base en la descripción de requerimientos y especificaciones anterior, en un primer nivel se divide el hardware en 3 módulos: Acondicionamiento de las señales de transmisores (M1), acondicionamiento y sincronización de señales de pulsos (M2), dispositivo programable (M3). Estos módulos y su interconexión se muestran en la Figura 36.

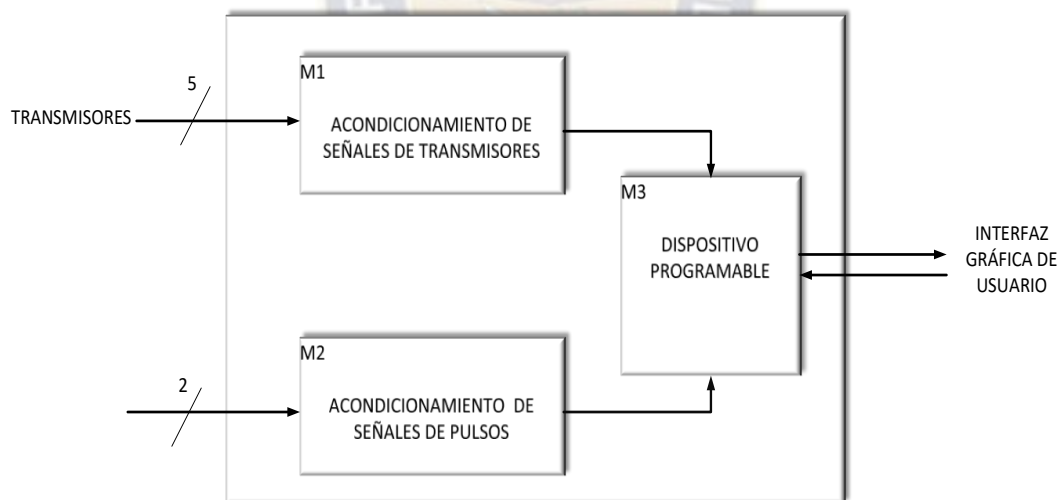


Figura 36. Módulos para el hardware.

Fuente: Elaboración propia.

En el módulo M1 se conectan las señales de los transmisores, donde se convierten a voltaje, se aíslan y filtran para su lectura con ADC. En el módulo M2 se conectan las señales de pulsos se filtran, se aíslan y se sincronizan para su conteo. En el módulo M3 se tiene el circuito necesario para el funcionamiento del DP. Bajo esta descripción, los módulos de M1 y M2 se dividen en sub módulos como se muestra en la Figura 37.



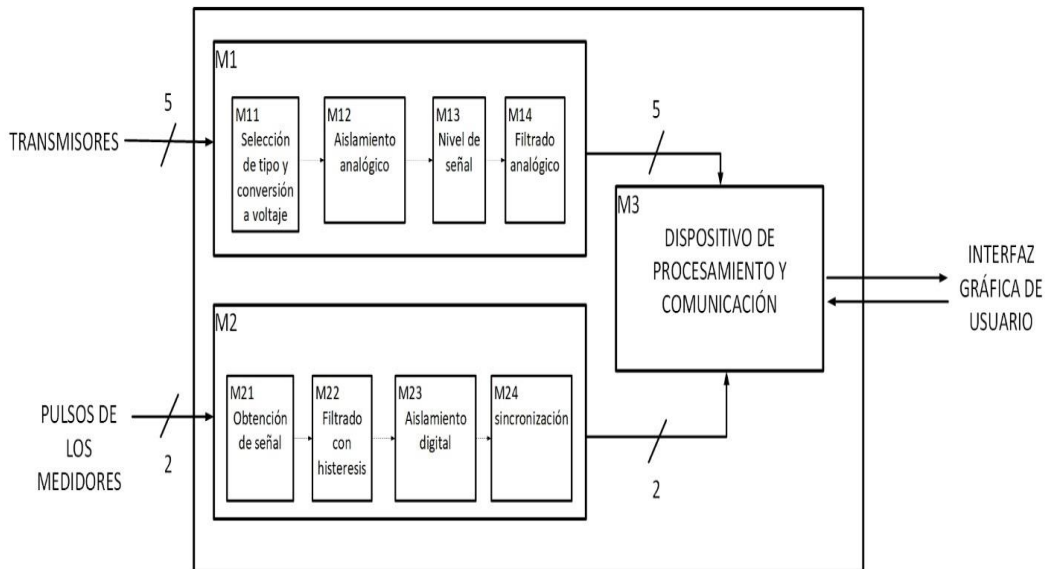


Figura 37. Módulos de primer y segundo nivel para el hardware.  
Fuente: Elaboración propia.

A continuación, se describe la funcionalidad e interconexión de los submódulos que componen los módulos M1, M2 y M3.

#### ACONDICIONAMIENTO DE SEÑALES DE TRANSMISORES (M1).

- M11 (Selección de tipo y conversión a voltaje): selecciona el tipo de transmisor (2 o 4 hilos) con un elemento de conmutación y convierte la señal de corriente de voltaje. Su función se define según la Ecuación 7.

$$V_1 = k * I_i \dots \dots (7) \quad 4 \text{ mA} \leq I_i \leq 20 \text{ mA}$$

Ecuación 7. Submódulo M11.

- M12 (Aislamiento analógico): Aísla la señal analógica de voltaje obtenida a partir de la corriente del transmisor. Al tratarse de aislamiento las fuentes de tensión de la señal de entrada y de salida no están conectadas al mismo nodo de masa. Su función se define según la Ecuación 8.

$$V_2 = k_1 * V_1 \dots \dots (8) \quad GND_o \neq GND_i$$

Ecuación 8. Submódulo M12.

- M13 (Nivel de señal): Amplifica o atenúa la señal aislada para que tenga los niveles adecuados para conectarse al dispositivo de procesamiento y comunicación. Su función se define según la Ecuación 9.

$$V_3 = k_2 * V_2 \dots \dots (9)$$

Ecuación 9, Submódulo M13.

- M14 (Filtrado analógico): refiere a un filtro analógico pasa bajo para eliminar componentes de ruido. Su función se define con la función de transferencia mostrada en la Ecuación 10.

$$\frac{V_o}{V_3} = \frac{1}{\tau s + 1} \dots \dots (10)$$

Ecuación 10. Submódulo M14.

Los módulos M11, M12, M13 y M14 se disponen uno a continuación de otro y de esta forma se tienen los elementos necesarios para una entrada analógica de transmisor de 2 o 4 hilos (conmutable) como se muestra en la Figura 38.



Figura 38. Submódulos para entrada de transmisor de 2 o 4 hilos.

Fuente: Elaboración propia

Según las especificaciones, se tienen mínimamente 5 entradas para transmisores de 2 o 4 hilos. Para conectar todos ellos, los elementos del módulo M1 se organizan en un sistema de medida multicanal con un ADC por canal, para 5 entradas como se muestra en la Figura 39.

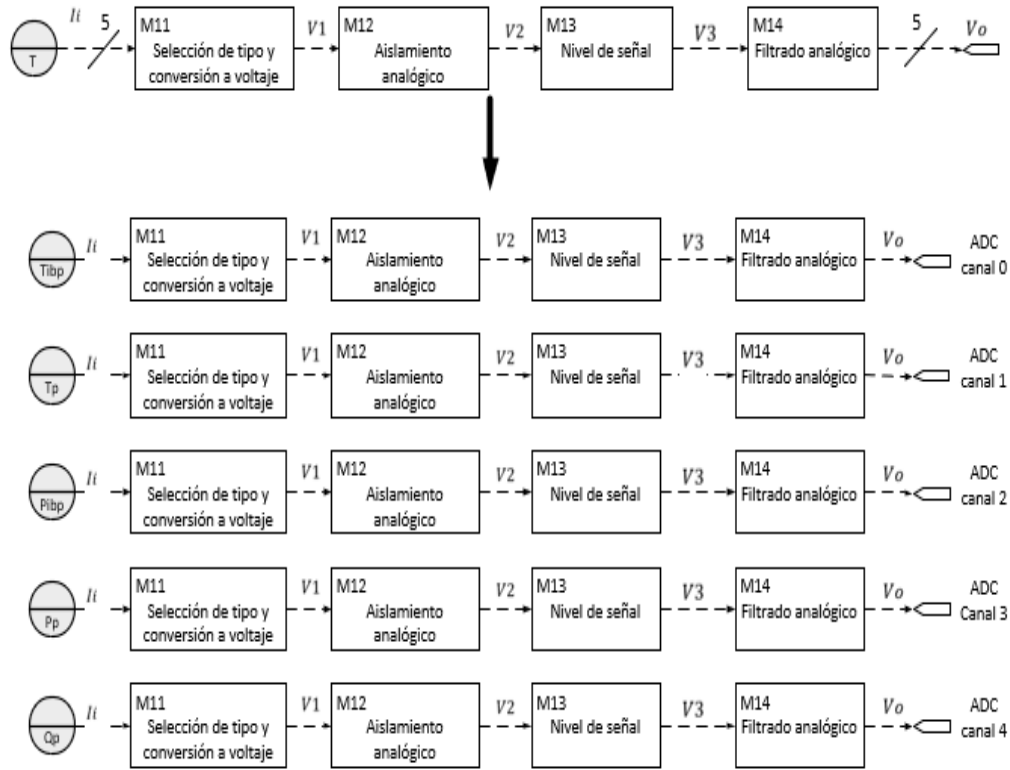


Figura 39. Acondicionamiento y medida de transmisores.

Fuente: Elaboración propia.

### ACONDICIONAMIENTO DE SEÑALES DE PULSOS (M2).

- M21 (Obtención de señal). Adquiere la señal de pulsos de los medidores. Según los requerimientos, contiene un elemento de conmutación que selecciona entrada de tipo activa o pasiva. De acuerdo las especificaciones, la señal de salida de este submódulo es una señal con niveles de 0 V a 24 V y está definida por la función:

$$V_1 = \begin{cases} 24; & 0 \leq t \leq \alpha T \\ 0; & \alpha T \leq t \leq T \end{cases}$$

$T = \frac{1}{f}$ ; es el periodo de la señal, propio del medidor.

Por lo general, esta señal es ruidosa y no es apta para conectar directamente a un dispositivo de conteo.

- M22(Comparador con histéresis). funciona como elemento de filtrado de la señal cuadrada obtenida en submódulo anterior (M21). Se compone de un comparador con histéresis y su función está definida por:

$$V_2 = \begin{cases} 24; & V_1 \geq V_a \uparrow \\ 0; & V_1 \leq V_b \downarrow \end{cases}$$

- M23(Aislamiento digital). Aísla la señal de pulsos filtrada. La señal de salida tiene la misma frecuencia que la señal de entrada; sin embargo, los niveles de tensión a la salida pueden variar. Su función se define por:

$$V_3 = \begin{cases} AV_2; & 0 \leq t \leq \alpha T \\ 0; & \alpha T \leq t \leq T \end{cases} \quad GND2 \neq GND3$$

- M24(Sincronización): sincroniza los pulsos del medidor bajo prueba y del patrón. Genera una ventana de conteo, donde, se deja pasar una cantidad N de pulsos del DUT y los pulsos del patrón, como se muestra en la Figura 40. El objeto de este elemento es que no se pierdan o ganen pulsos del DUT que pasaran a ser contados por el dispositivo de procesamiento y comunicación.

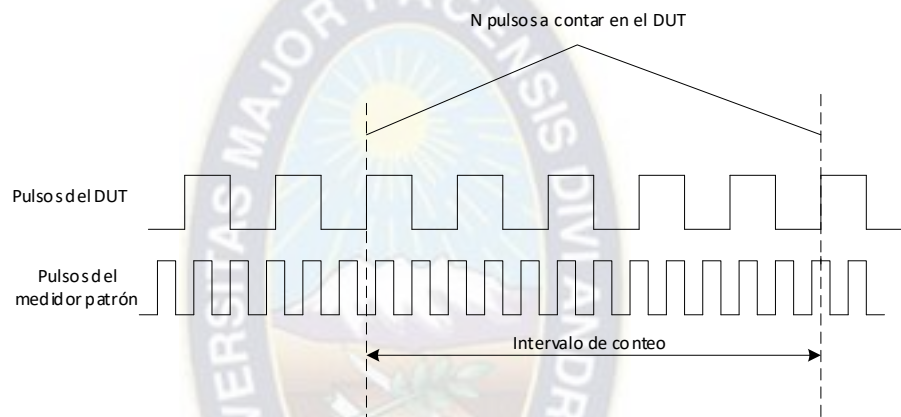


Figura 40. Conteo de pulsos en medidor bajo prueba y patrón.

Fuente: Elaboración propia.

Para lograr este conteo, nos apoyamos en señales auxiliares. Entonces, las características funcionales requeridas para este submódulo son:

- Una señal lógica denominada “habilitación” inicia y detiene el conteo de pulsos; es decir, funciona como señal de disparo. Esta señal será generada desde el dispositivo de procesamiento y comunicación.
- Cuando la señal de “habilitación” pasa de 0 a 1 lógico, se espera el primer flanco ascendente de los pulsos del medidor bajo prueba.
- Cuando se tiene el primer flanco ascendente de la señal de pulsos del medidor bajo prueba se habilita el conteo de los pulsos, en ambos medidores.
- Cuando se llega a contar una cantidad deseada de pulsos menos uno en el medidor bajo prueba; se espera un pequeño instante de tiempo y la señal “habilitación” pasa de 1 a 0 lógico.
- Después de que la señal “habilitación” halla pasado a 0 lógico, se espera el siguiente flanco ascendente de la señal de pulsos del DUT. Una vez que se tiene este flanco se detiene la cuenta inhabilitando las señales de pulsos del medidor bajo prueba y patrón; también, se pone a 1 lógico una señal denominada “finalización” para indicar al dispositivo de procesamiento y comunicación la culminación del conteo.

En Figura 41 se muestra el diagrama de tiempos para el funcionamiento descrito.

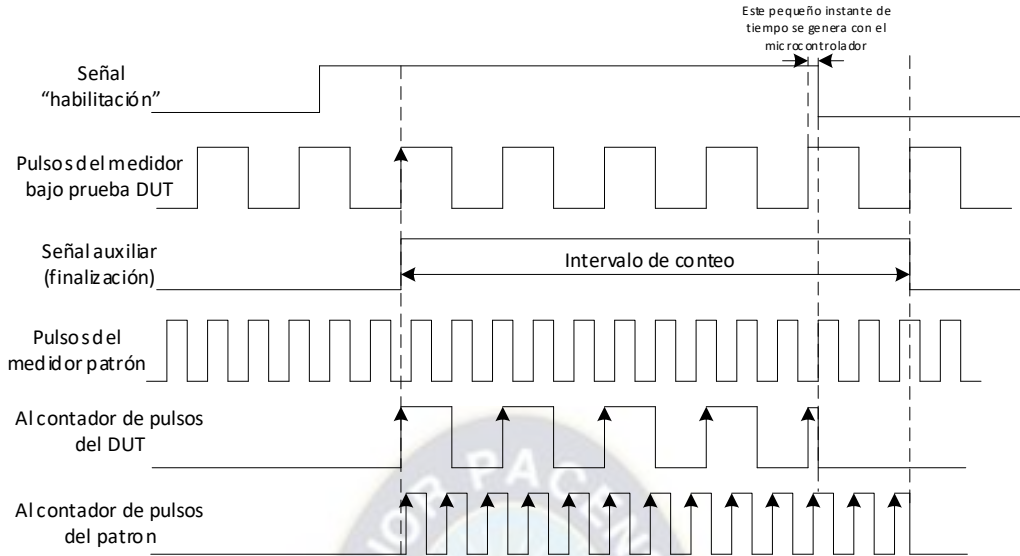


Figura 41. Diagrama de tiempos para la sincronización de los pulsos.

Fuente: Elaboración propia.

Las entradas de pulsos del medidor bajo prueba y patrón deben disponer de los elementos de obtención (M21), filtrado (M22) y aislamiento (M23) antes de ingresar al submódulo de sincronización (M24). La conexión de los submódulos M21, M22, M23 y M24, que componen al módulo M2, se muestra en la Figura 42.

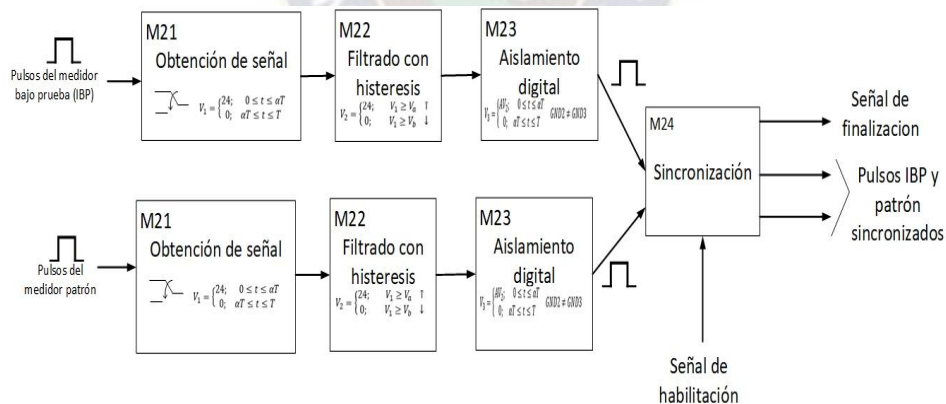


Figura 42. Elementos del módulo de M2.

Fuente: Elaboración propia.

### DISPOSITIVO PROGRAMABLE (M3).

Refiere a los componentes necesarios para el funcionamiento del dispositivo programable que está integrado en el hardware físicamente. Este módulo dependerá del DP elegido según las especificaciones necesarias para su programación y uso.

Con base en los requisitos, especificaciones y descripción de cada módulo (M1, M2 Y M3), se ordena cada elemento (módulos y submódulos) en el hardware como se muestra en la Figura 43.

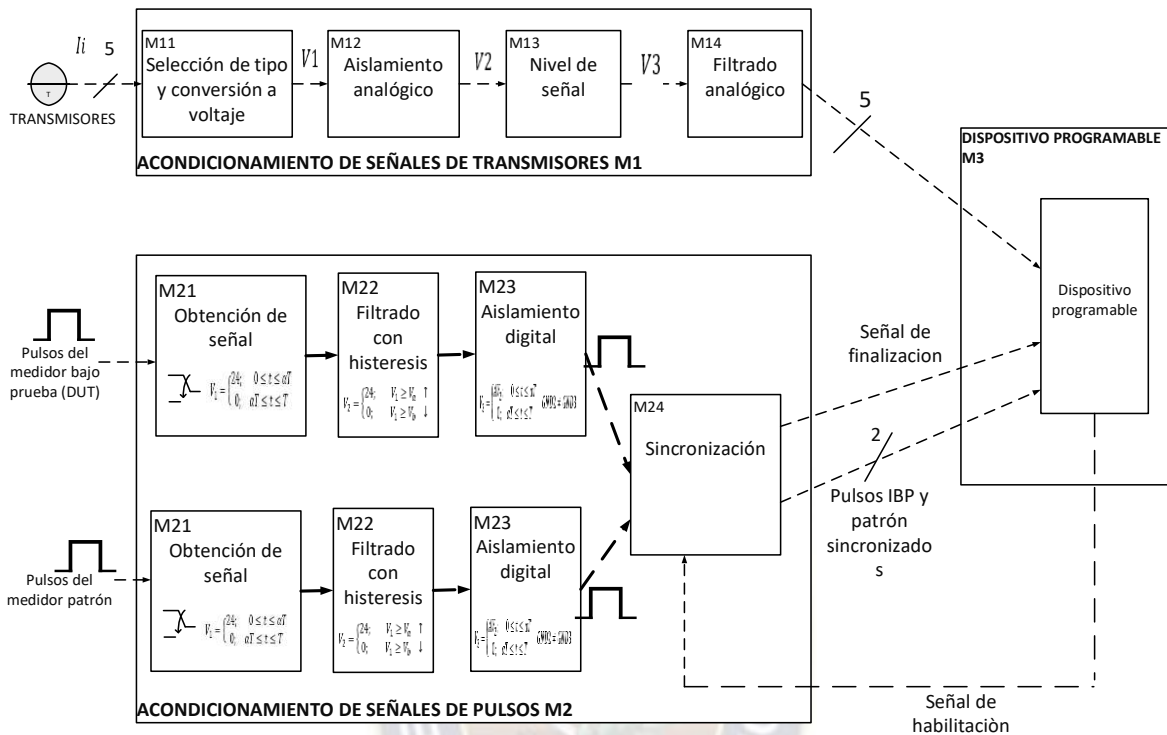


Figura 43. Organización de módulos y submódulos de Hardware.

Fuente: Elaboración propia.

#### 4.2.2. PROGRAMA PARA EL DISPOSITIVO PROGRAMABLE.

El dispositivo de programable (DP) es el nexo entre el hardware y la GUI. Este realiza la lectura de las señales acondicionadas (transmisores y medidores) en el hardware, y posteriormente envía los datos leídos a la GUI, a través de un protocolo de comunicación. También, recibe comandos desde la GUI para realizar las lecturas. Entonces, las entradas son las señales analógicas, pulsos de los medidores, señal de “finalización”, y datos recibidos (caracteres) desde la GUI. Las salidas son la señal “habilitación” para inicio y finalización del conteo, y los datos (cadena de caracteres) transmitidos hacia la GUI. Las entradas y salidas descritas se muestran en la Figura 44.



Figura 44. Entradas y salidas al DP.

Fuente: Elaboración propia.

Con base en lo explicado, se define las siguientes especificaciones para el DP:

- Debe contar con un módulo Conversor Análogo a Digital (*Analog to Digital Converter – ADC*).
  - Mínimamente 5 canales.
  - Resolución de 10 bits o mayor.
- Debe contar con módulo conteo.
  - Mínimamente 2 canales (contadores). Un canal para medidor bajo prueba y otro para el patrón.
- Debe contar con módulo de entradas y salidas de propósito general configurable, denominado módulo I/O.
  - Mínimamente 2 canales configurables. Uno para la señal de “habilitación” y otro para la señal de “finalización”.
- Debe contar con módulo de comunicación serial.
  - Manejo de protocolos estándar como ser RS232 o RS485.

En cuanto a especificaciones del programa se tiene:

- Debe tener capacidad de procesar e interpretar datos recibidos desde la interfaz gráfica de usuario. En este punto se define las tareas que debe realizar según el comando recibido como se detallan en la Tabla 4.

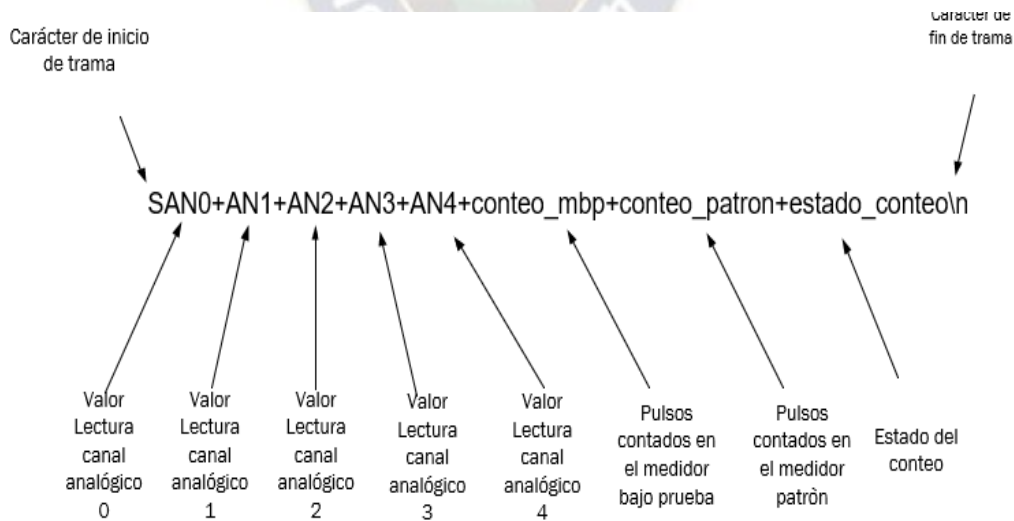


<i>Comando</i>	<i>Descripción</i>	<i>Tareas</i>
I	Inicio del conteo	Pone a 0 el valor de los contadores.  Pone a 1 la señal de “habilitación” para el conteo.
N	registrar cantidad de pulsos para conteo	Inicia el registro de pulsos a contar.
f	finalizar el registro de cantidad de pulsos para conteo	Finaliza el registro de pulsos a contar.
d	detener conteo ya iniciado	Pone a 0 la señal “habilitación”

*Tabla 4. Comandos para el dispositivo de procesamiento y comunicación.*

*Fuente: Elaboración propia.*

- El programa debe enviar la lectura de canales analógicos, conteo de pulsos y estado de conteo de forma periódica a la GUI. Para ello, se opta por ordenar los datos en una cadena de caracteres denominada trama, y posteriormente enviados a la GUI mediante comunicación serial. La trama tiene como inicio el carácter “S”, como fin el carácter “\n” y como separador de elementos el carácter “+”. El formato de la trama se muestra en la Figura 45.



*Figura 45. Trama generada.*

*Fuente: Elaboración propia.*

- Debe responder a los comandos recibidos desde la GUI en cualquier momento.



- Debe realizar el conteo de pulsos de forma paralela a los otros procesos, sin afectar ni verse afectado por los mismos.

A cada uno de los módulos, que se describieron, se asigna un código. Los módulos (M1, M2, M3 y M4) se interconectan a través de un módulo denominado de Configuración y Lectura (M5) como se muestra en la Figura 46.

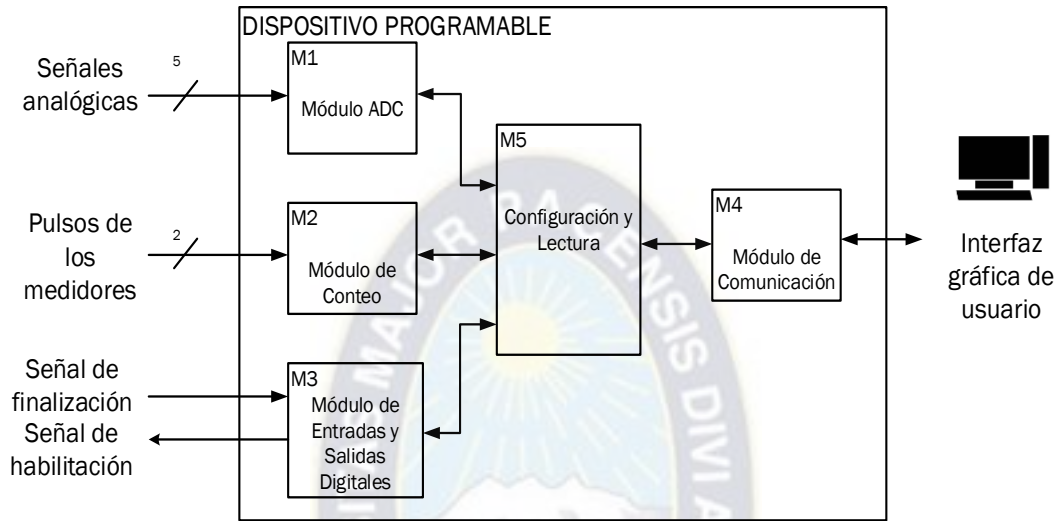


Figura 46. Módulos e interconexión en el DP.

Fuente: Elaboración propia.

A su vez, cada módulo requiere de funciones, las cuales se muestran en la Figura 47, Figura 48, Figura 49 y Figura 50.

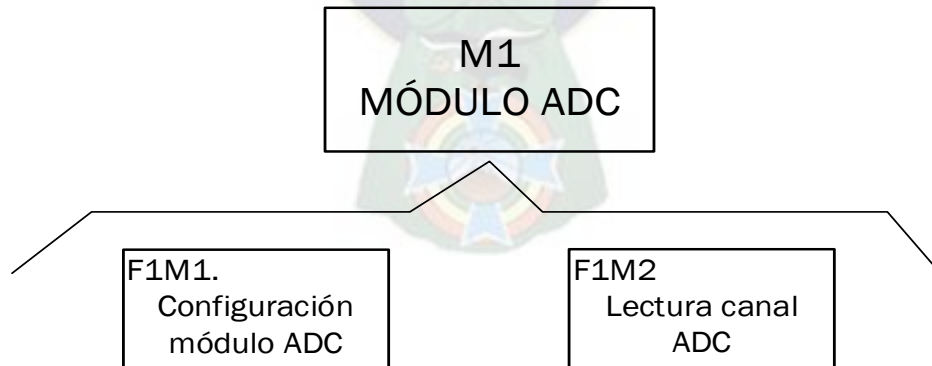


Figura 47. Funciones módulo M1.

Fuente: Elaboración propia.

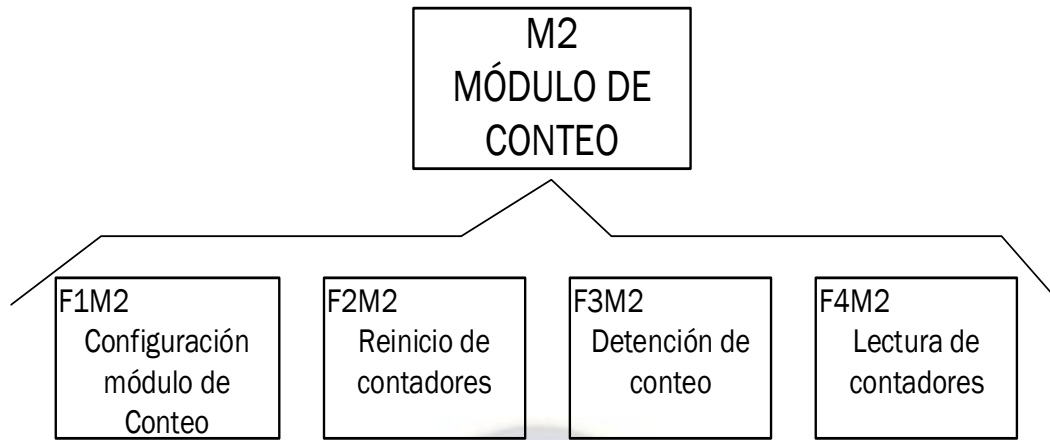


Figura 48. Funciones módulo M2.

Fuente: Elaboración propia.

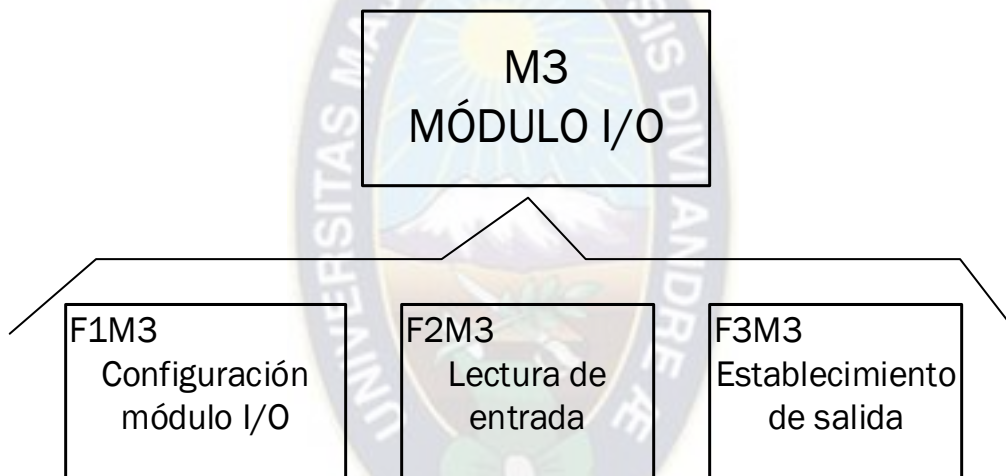


Figura 49. Funciones módulo M3.

Fuente: Elaboración propia.

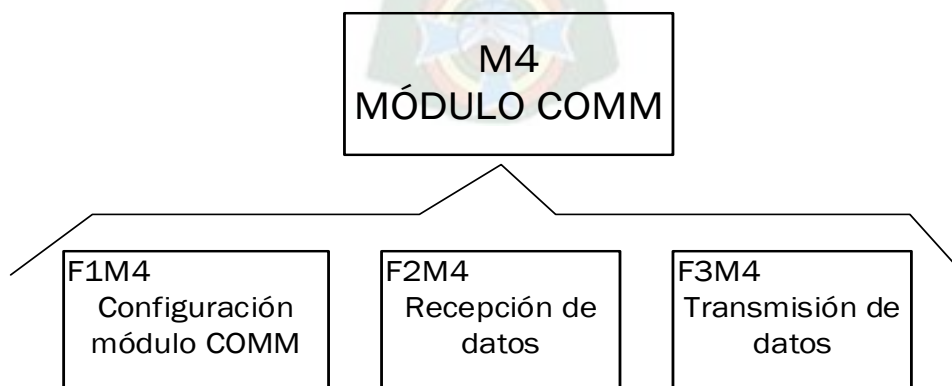


Figura 50. Funciones módulo M4.

Fuente: Elaboración propia.

Las funciones para el módulo ADC (M1) son:

- Configuración modulo ADC (F1M1): Se configura las características necesarias para el funcionamiento del módulo ADC.
- Lectura de canal ADC (F2M1): donde se lee el valor digital que corresponde a la señal analógica de alguno de los canales del módulo ADC.

Las funciones para el módulo de conteo (M2) son:

- Configuración módulo de conteo (F1M2): Se configura las características necesarias para el funcionamiento del módulo M2.
- Reinicio de contadores (F2M2): pone en 0 el valor de los contadores conectados a los pulsos de los medidores.
- Detención de conteo (F3M2): Detiene el conteo por finalización. Cuando se llega a contar una cantidad predefinida de pulsos del DUT, se debe detener el conteo.
- Lectura contadores (F4M2): realiza la lectura del valor acumulado en los contadores de pulsos.

Las funciones para el módulo de entradas y salidas I/O (M3) son:

- Configuración módulo I/O (F1M3): Se configura las características necesarias para el funcionamiento del módulo I/O.
- Lectura de entrada (F2M3): Obtiene el valor de una entrada digital, es decir, si está en estado lógico “0” o “1”. Esta función sirve para la interacción con la sincronización de pulsos (*submódulo M24 del Hardware*).
- Establecimiento de salida (F3M3): Establece el valor de una salida digital, es decir lo pone en estado “1” o “0” lógico. Esta función sirve para la interacción con la sincronización de pulsos (*submódulo M24 del Hardware*).

Las funciones para el módulo de comunicación COMM (M4) son:

- Configuración módulo COMM (F1M4): Se configura las características necesarias para el funcionamiento del módulo de comunicación.
- Recepción de dato COMM (F2M4): Recepción de datos que están en el buffer de comunicación. Esta función interactúa con la GUI y realiza procesamiento según los comandos detallados en la Tabla 4.
- Transmisión de dato COMM (F3M4): Envía datos por el puerto de comunicación. Esta función interactúa con la GUI y envía los datos con la estructura de trama mostrada en la Figura 45.

Las funciones de cada módulo, según las especificaciones, son ordenados en procesos y presentan jerarquía, orden y momentos de ejecución como se muestra en la Figura 51.

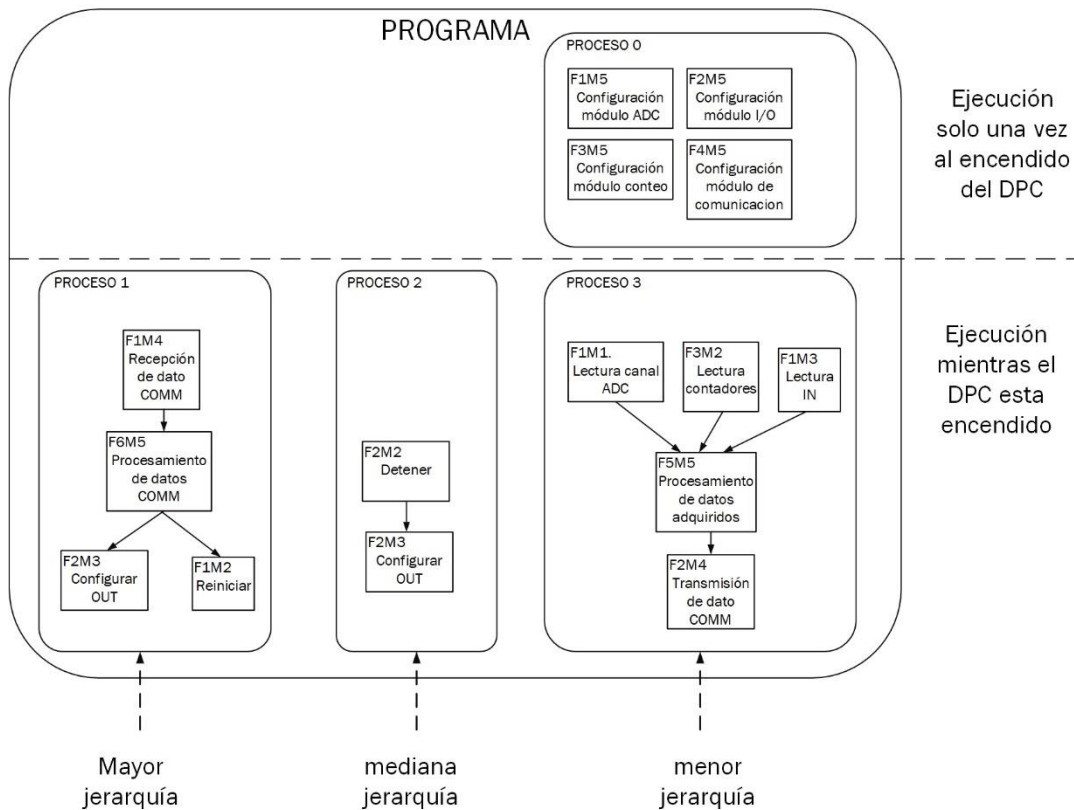


Figura 51. Procesos para el programa del DPC.

Fuente: Elaboración propia.

A continuación, se explica el algoritmo y flujograma de cada proceso.

### PROCESO 1.

Contempla las tareas relacionadas a la recepción de datos desde la GUI y su tratamiento. Este proceso se ejecuta mientras el DP este encendido después de la configuración de cada módulo. Así mismo, tiene la mayor jerarquía dado que se debe atender a las tareas que pueda solicitar un operador desde la GUI.

Básicamente, Verifica si existe algún carácter en la recepción de datos del módulo de comunicación. En caso afirmativo, se lee el comando y se ejecuta la tarea correspondiente según el detalle de la Tabla 4. Como se ve, las tareas que realiza son de configuración de pin de salida y reseteo de contadores. En la Tabla 5 se describe el pseudocódigo del proceso y en la Figura 52 se muestra el diagrama de flujo.

- 1) ¿Existe carácter en elemento de recepción de datos del módulo de comunicación?
- 2) Si existe, leer y ejecutar tarea en función del carácter recibido:
  - 'i': resetear los contadores, poner a 1 señal "habilitación"

- 'n': Iniciar registro de valor para conteo.
  - 'f': finalizar registro de valor para conteo.
  - 'd': poner a 0 señal de "habilitación".
- Si no saltar a 3).
- 3) Salir.

Tabla 5. Pseudocódigo del proceso 1.

Fuente; Elaboración propia.

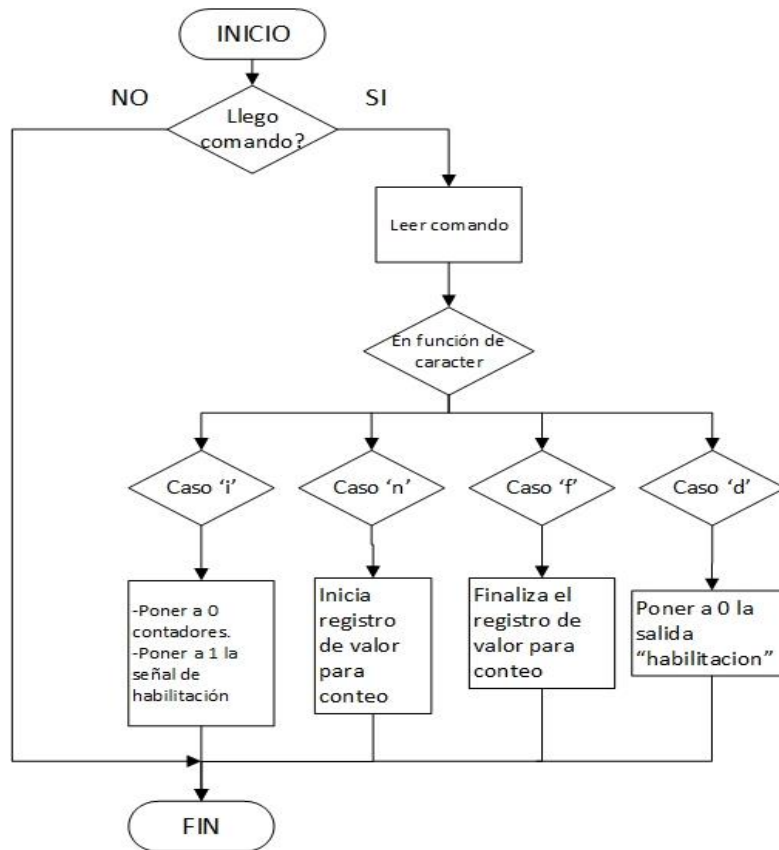


Figura 52. Diagrama de flujo en PROCESO 1.

Fuente: Elaboración propia.

## PROCESO 2.

Contempla las tareas de conteo de pulsos. Presenta jerarquía media. Este proceso se ejecuta mientras el DPC este encendido después de la configuración de cada módulo. Básicamente, incrementa el valor de los contadores con cada pulso a su entrada. En el caso del contador del medidor bajo prueba, también compara el valor contado con una cantidad programada de pulsos para detener el conteo. En la Tabla 6 se describe el pseudocódigo del proceso 2 y en la Figura 53 se muestra su diagrama de flujo.

- Para el contador de pulsos del DUT:
  - 1) Si se detecta pulso incrementar el valor de contador.
  - 2) Si el valor contado es igual al valor programado se pone a 0 la señal de “habilitación”.
- Para el contador de pulsos del medidor patrón:
  - 1) Si se detecta pulso incrementar el valor de contador.

Tabla 6. Pseudocódigo del proceso 2.

Fuente: Elaboración propia.

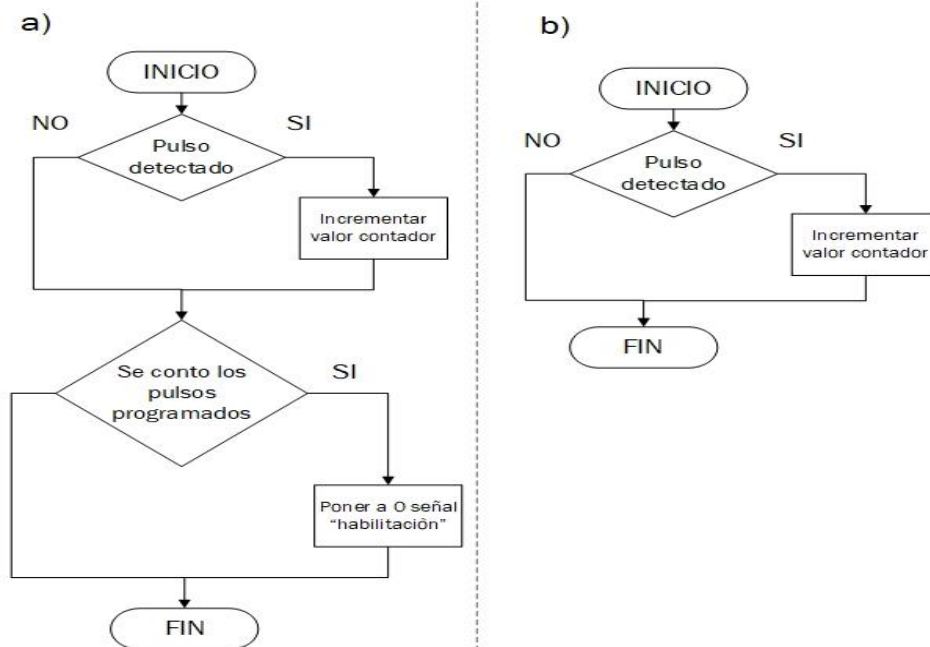


Figura 53. Diagrama de flujo para el PROCESO 2 en a) DUT b) medidor patrón.

Fuente: Elaboración propia.

### PROCESO 0 y 3.

Contempla las tareas de configuración de los módulos, lectura de datos y su transmisión. La configuración de los módulos se realiza por única vez al encendido del DP. La lectura de datos y su transmisión se ejecuta mientras el DP este encendido después de la configuración de cada módulo.

En este proceso se configura e inicializa los módulos; se realiza la lectura de canales analógicos, pulsos contados y estado de conteo; se genera y transmite la trama. La lectura de datos debe realizarse cada T ms. Donde T es el máximo tiempo de respuesta de los transmisores que están instalados en la

línea que se puede tomar de la descripción de los transmisores instalados en la línea (Tabla 9 del capítulo de Marco de Referencia).

En la Tabla 7 se describe el pseudocódigo para ambos procesos y en la Figura 54 se muestra su diagrama de flujo.

- 1) Inicializar y configurar el módulo de entradas y salidas I/O.
- 2) Inicializar y configurar el módulo ADC.
- 3) Inicializar y configurar el módulo de conteo.
- 4) Inicializar y configurar el módulo de comunicación.
- 5) Leer el estado de conteo, es decir, la señal de entrada “finalización”.
- 6) Leer el estado valor de los contadores, medidor bajo prueba y patrón.
- 7) Leer los canales analógicos.
- 8) Generar la trama con los valores leídos.
- 9) Enviar la trama generada.
- 10) Esperar T (s) y volver a 5).

*Tabla 7. Pseudocódigo para proceso 0 y 3.*

*Fuente; Elaboración propia.*





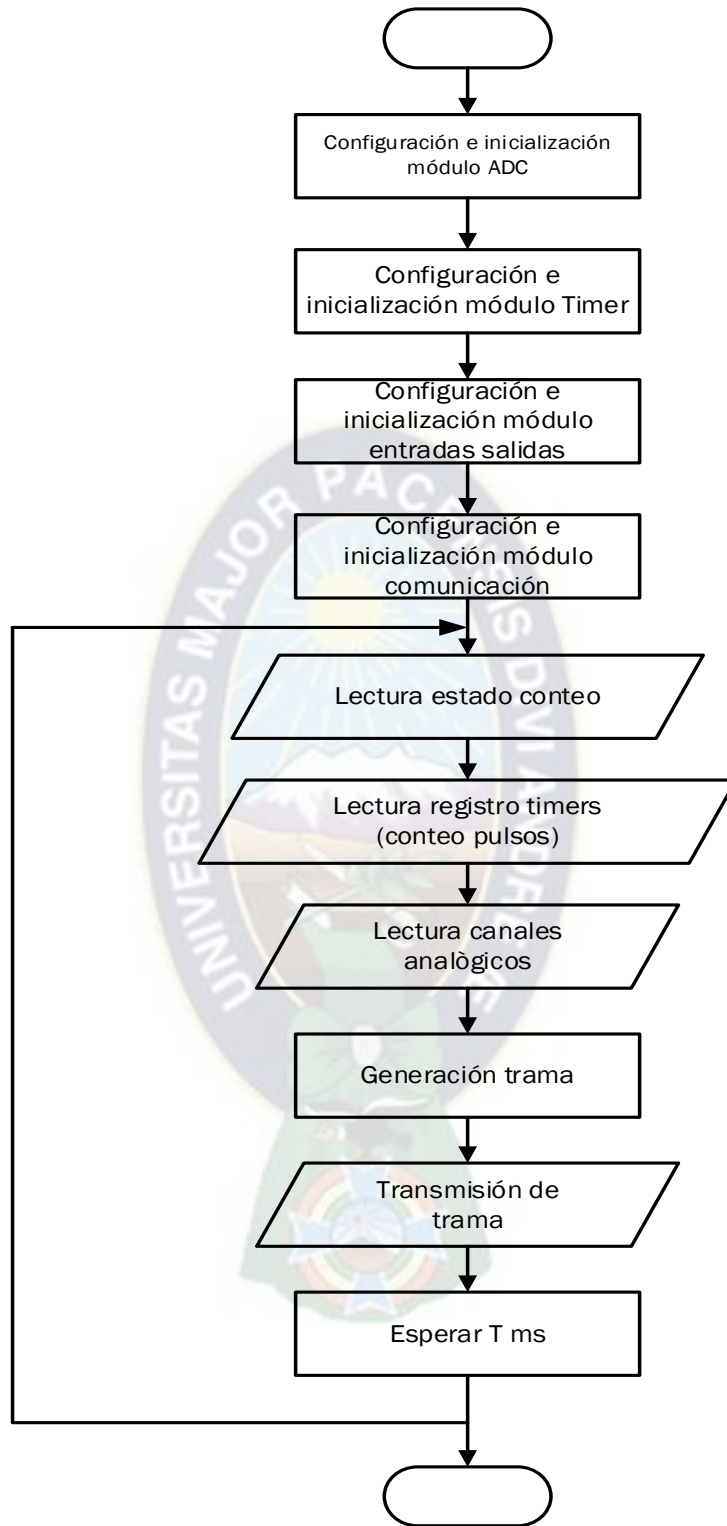


Figura 54. Diagrama de PROCESO 0 y 3.

Fuente: Elaboración propia.



### 4.2.3. INTERFAZ GRÁFICA DE USUARIO.

Refiere a una interfaz gráfica programada en computadora. La computadora se conecta al DP integrado en el hardware. En la interfaz, se visualiza las variables y datos en la planta de calibración, se configura el conteo de pulsos y se ingresa datos de condiciones ambientales por parte de un operador. Entonces, se constituye en el nexo entre operador y hardware como se muestra en la Figura 55.

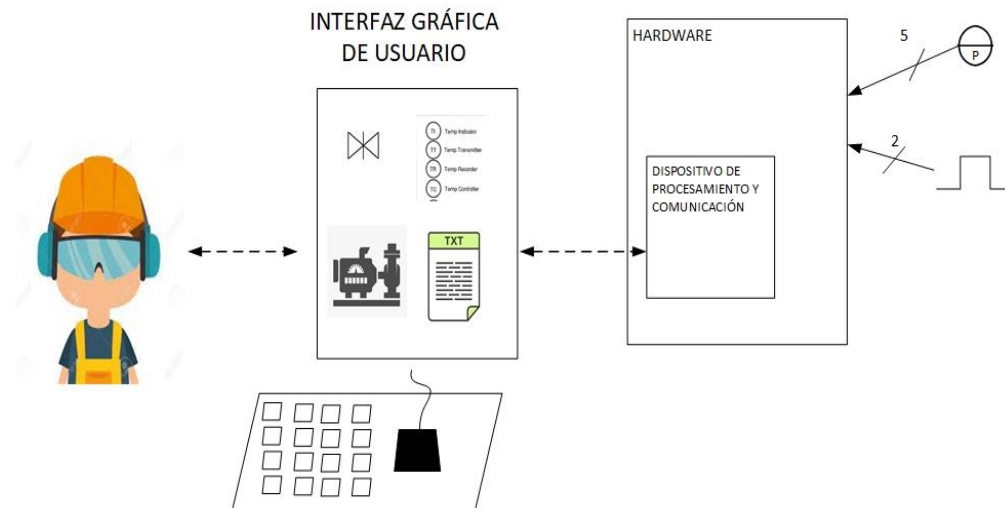


Figura 55. Interfaz hombre máquina.

Fuente: Elaboración propia.

La interfaz envía comandos hacia el DP, y recibe datos haciendo uso del puerto de comunicación. Los datos recibidos se procesan, y los valores obtenidos se muestran en indicadores numéricos, de texto y gráficos; y se almacenan los necesarios en un archivo de texto. Los comandos se envían cuando se activan botones o escriben datos sobre cuadros de texto. Entonces, la interfaz gráfica contempla los elementos visuales y el algoritmo de funcionamiento.

Las especificaciones para la GUI son:

- Debe contar con indicadores numéricos, de texto, luminosos y gráficos de los valores obtenidos por los transmisores para su visualización.
- Debe contar con un esquema de distribución de los elementos en la planta de calibración para interpretación de los datos del proceso.
- Debe contar con entradas numéricas, de texto y botones para la configuración, inicio y pausa del conteo y almacenamiento de datos.
- Debe contar con entradas numéricas para registrar los datos de condiciones ambientales en el proceso de calibración.
- Debe medir el tiempo de conteo de pulsos.
- Debe mostrar los resultados durante un conteo realizado, de forma que se realice ajuste en el medidor bajo prueba en caso de ser necesario.

- Debe mostrar el estado del proceso.
- Debe contar con un botón de finalización de todo el proceso.
- Debe contar con un selector e indicador luminoso para establecer comunicación con el hardware.
- El programa debe tener capacidad de manejo del puerto serial del computador.
  - Protocolos RS232 y/o RS485.
- El programa debe tener capacidad de visualizar las variables presentes en el proceso mientras responde a comandos del operador y procesa y almacena los datos de la trama enviados por el dispositivo de procesamiento y comunicación.
  - Los datos se guardan de forma tabulada en archivos de texto.

Según lo descrito, la interfaz gráfica de usuario es un programa que corre en un computador. Este integra elementos visuales en la pantalla y una lógica de programa. Los elementos visuales son: de monitoreo (gráficos, indicadores numéricos, luminosos o de texto) y activados por operador (botones, selectores, entradas numéricas o de texto). La lógica de programa realiza el envío de comandos y datos hacia el DP, el procesamiento y almacenamiento de datos. En la Figura 56 se muestra los elementos necesarios para el funcionamiento descrito.

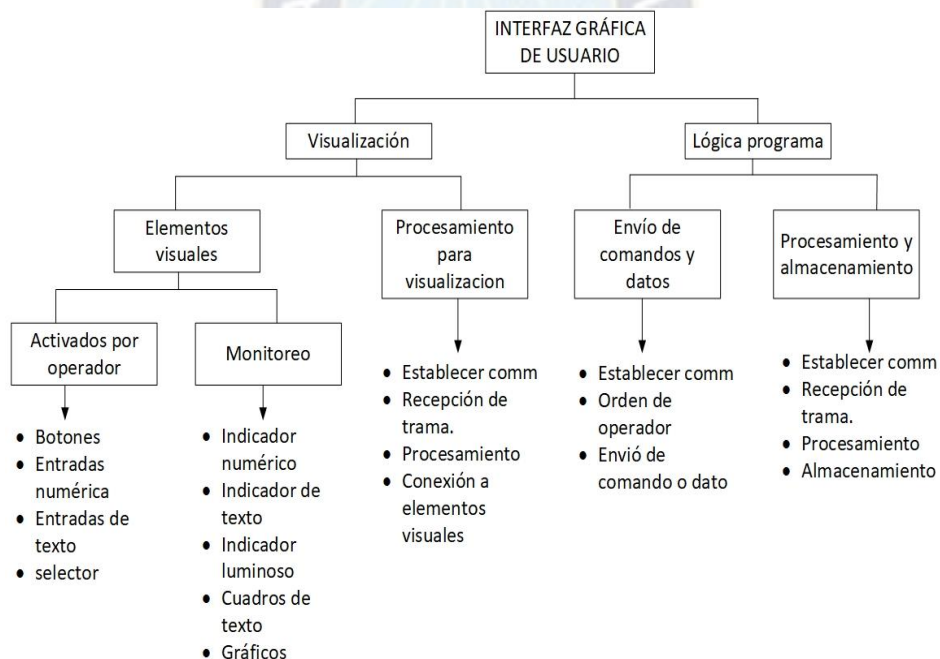


Figura 56. Elementos de la interfaz gráfica de usuario.

Fuente: Elaboración propia.

El envío de comandos y datos depende de los botones y entradas numéricas o de texto activados por el operador. El procesamiento y almacenamiento depende de la trama recibida y las órdenes del operador. El procesamiento para visualización recibirá los datos del elemento de procesamiento y almacenamiento. La conexión de elementos de la GUI se muestra en la Figura 57.

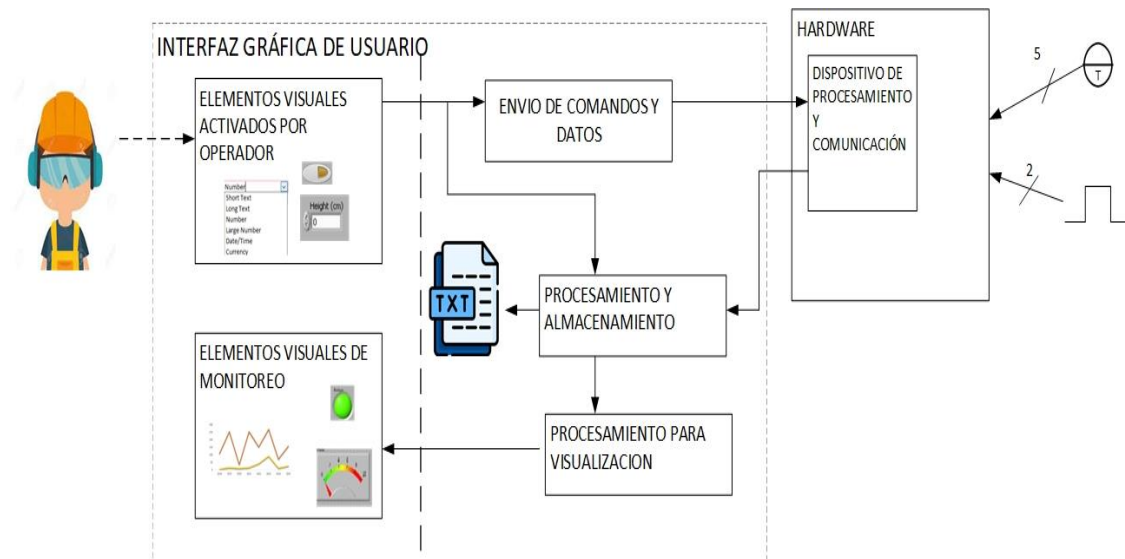


Figura 57. Conexión de elementos de la GUI.

Fuente: Elaboración propia.

Se diferencia 2 partes fundamentales de la GUI que son la visualización y la lógica de programa. A continuación, se detalla y esquematiza cada una de estas partes.

### Visualización.

Refiere a la distribución de elementos visuales en la pantalla del computador. Con base en los requerimientos y especificaciones, la pantalla se divide en 5 áreas:

- Esquema de la planta. presenta de forma estática la distribución de transmisores, bomba y medidores en la línea de tuberías, basado en el boceto mostrado en el punto 1.1 del capítulo de introducción. Cuenta con indicadores numéricos y de texto para exhibir los datos de los transmisores.
- Gráficos. Presenta en forma de gráfico el progreso de las variables en función del tiempo. De forma prioritaria, debe mostrar la temperatura, presión y caudal en la línea.
- Datos y configuración. En esta área el operador puede configurar la comunicación, conteo de pulsos e introducir datos de las condiciones ambientales. Presenta botones, cuadros para entrada de datos numérica o de texto y/o selectores entre otros.
- Estado. Presenta indicadores de texto y/o luminosos para que el operador pueda ver el estado del proceso.
- resultados. Presenta los resultados en el conteo de pulsos mediante indicadores o cuadros de texto.

En la Figura 58 se muestra la distribución de las áreas descritas.

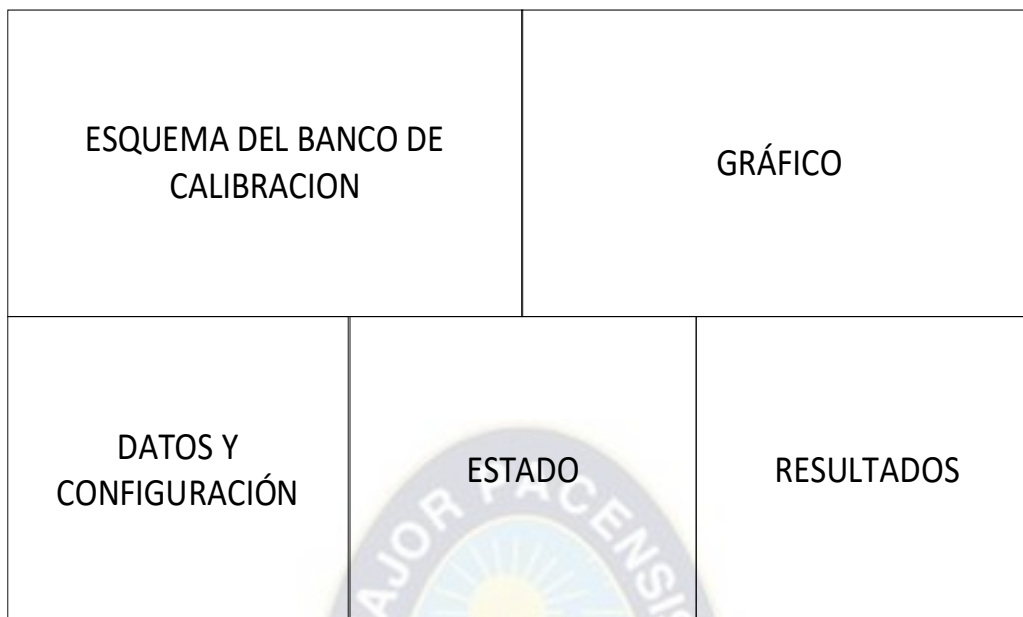


Figura 58. Áreas de la pantalla.

Fuente: Elaboración propia.

Los indicadores, botones y demás elementos visuales que componen cada una de estas áreas se detallan en la Tabla 8.

AREA	DESIGNACIÓN	TIPO DE ELEMENTO	DESCRIPCIÓN
DATOS Y CONFIGURACIÓN	Serial name	Selector	Selecciona el puerto de comunicación serial COM
	T Amb	Entrada de datos numérico	Para ingresar el valor de la temperatura ambiental
	HR Amb	Entrada de datos numérico	Para ingresar el valor de la humedad relativa ambiental
	P Amb	Entrada de datos numérico	Para ingresar el valor de la presión atmosférica ambiental
	VOLUMEN	Entrada de datos numérico	Programa el volumen que se desea que circule en la calibración
	Repeticiones	Entrada de datos numérico	Define el número de repeticiones para el conteo de pulsos a un determinado volumen de calibración
	k factor DUT	Entrada de datos numérico	Para introducir el valor de la constante de pulsos del DUT
	k factor Patrón	Entrada de datos numérico	Para introducir el valor de la constante de pulsos del patrón
	INICIAR CONTEO	Botón	Botón para iniciar el conteo de pulsos

	DETENER CONTEO	Botón	Botón para detener el conteo de pulsos
	DETENER PROCESO	Botón	Botón para detener todo el proceso
	N pulsos a contar	Indicador numérico	muestra la cantidad de pulsos para el conteo. Se calcula en función del volumen programado y la constante k del DUT.
ESTADO	Programación	indicador luminoso	Led para mostrar que se pueden programar elementos y mandar a iniciar el conteo
	Conteo	indicador luminoso	Led para mostrar que se está realizando conteo de pulsos.
	Generando reporte	indicado luminoso	Led para mostrar que se está generando reporte con datos adquiridos.
RESULTADOS	RESULTADOS DE ULTIMA CORRIDA	Indicador tipo texto	Texto donde se muestra los resultados obtenidos en el conteo de pulsos.
ESQUEMA DE LA PLANTA	temp dut	Indicador numérico	Muestra el valor obtenido del transmisor de temperatura para el DUT
	pres dut	Indicador numérico	Muestra el valor obtenido del transmisor de presión para el DUT
	temp patron	Indicador numérico	Muestra el valor obtenido del transmisor de temperatura para el patrón
	pres patron	Indicador numérico	Muestra el valor obtenido del transmisor de presión para el patrón
	N_DUT	Indicador numérico	Muestra la cantidad de pulsos contados por el DUT
	Vol_DUT	Indicador numérico	Muestra el volumen medido por el DUT
	N_Patrón	Indicador numérico	Muestra la cantidad de pulsos contados por el patrón
	Vol_Patrón	Indicador numérico	Muestra el volumen medido por el patrón
	Caudal_Q	Indicador numérico	Muestra el valor obtenido del transmisor de caudal a través del medidor patrón
GRÁFICO	caudal_graph	grafico	Muestra en un gráfico el comportamiento del caudal

	temp_graph	grafico	Muestra en un gráfico el comportamiento de los transmisores de temperatura
	pres_graph	grafico	Muestra en un gráfico el comportamiento de los transmisores de presión

Tabla 8. Elementos visuales por área de pantalla.

Fuente: Elaboración propia.

### Lógica de programa.

El programa envía comandos y datos al hardware, procesa y almacena los datos recibidos en la trama y almacena información para la visualización. El funcionamiento se basa en los comandos recibidos y la trama de datos. Para su programación, se distingue dos procesos fundamentales: el primer proceso agrupa los elementos de envío de comandos y datos y procesamiento y almacenamiento; el segundo es el procesamiento para visualización.

La arquitectura con la cual se puede atender a los procesos del programa es Productor Consumidor. El Productor se encarga de la comunicación, procesamiento, almacenamiento y generación de reporte. El consumidor realiza el procesamiento para la visualización conectando los datos obtenidos a los elementos visuales. Entre productor y consumidor existe una memoria. La memoria recibe datos del productor y los entrega al consumidor. En la Figura 59 se muestra en la arquitectura descrita para el programa.

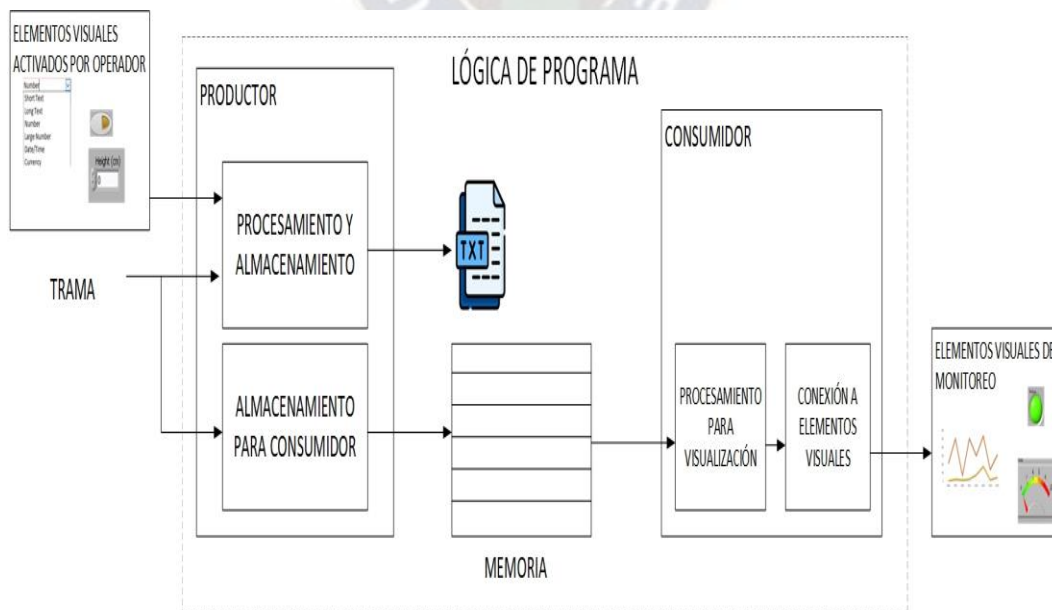


Figura 59. Arquitectura del programa de la GUI.

Fuente: Elaboración propia.



El productor contiene una máquina de estados para el procesamiento y almacenamiento. Este proceso depende de los elementos visuales activados por el operador y la trama de datos recibida. En la Figura 60 se muestra la máquina de estados para el cometido del productor.

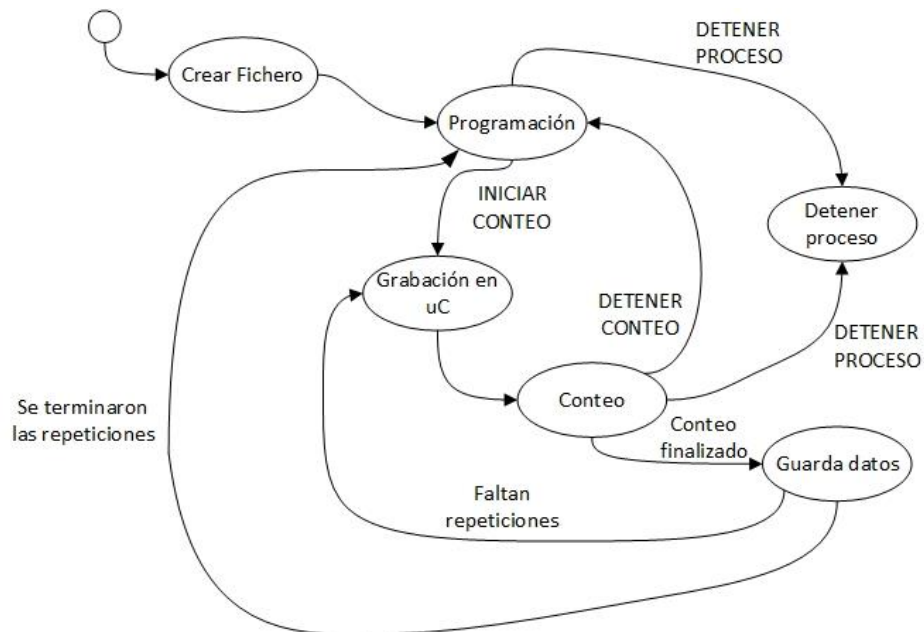


Figura 60. Máquina de estados para procesamiento y almacenamiento.

Fuente: Elaboración propia.

A continuación, se define las tareas que realiza cada estado y las condiciones de transición.

**Crear fichero.** Es el primer estado después del inicio. Las acciones y condiciones de transición son:

- Generar un archivo de texto para el almacenamiento de los datos adquiridos.
- Limpiar el buffer de comunicación serial.
- Realizar la transición al estado “programación” sin condición.

**Programación.** Se espera que el operador introduzca los datos necesarios para la calibración. Las acciones y condiciones de transición para este estado son:

- Poner a 0 el valor de repeticiones realizadas.
- Verificar si existen datos en el puerto serial, de ser así, almacena la trama en la memoria.
- Establece en uno (1) indicador luminoso “programación”.
- Si se oprime el botón “DETENER PROCESO” pasa al estado “detener proceso”.
- Si se oprime el botón “INICIAR CONTEO” pasa al estado “grabación en uC”.

**Grabación en uC.** Envía los comandos necesarios para el registro e inicio de conteo en el dispositivo de procesamiento y comunicación. También inicializa las variables necesarias antes del conteo. Las acciones y transiciones son:

- Inicializar el array donde se almacenan los datos de los transmisores en [0 0 0 0 0].
- Inicializar el contador de tiempo.
- Enviar el carácter 'r', la cantidad de pulsos a contar, el carácter 'f' e 'i', a través del puerto serial.
- Realizar la transición al estado “conteo” sin condición.

**Conteo.** Estado para el conteo de pulsos y almacenamiento de los valores de los transmisores. Las acciones y transiciones son:

- Verificar si se tiene datos en el puerto serial. De ser así, recibe el valor del puerto serial y almacena la trama en la memoria.
- Convertir los valores de lectura analógica recibidos en sus valores de temperatura y presión y almacenarlos en arreglos (*arrays*).
- Actualizar el valor de los conteos de pulsos.
- Contar el tiempo de conteo y mostrarlo en el indicador “tiempo”.
- Establecer en 1 el led indicador “conteo” y en 0 el led indicador “programación”.
- Si se oprime el botón “DETENER PROCESO” se realiza transición al estado “detener proceso”.
- Si se oprime el botón “DETENER CONTEO” envía el comando 'd' a través de puerto serie y se realiza transición al estado “programación”.
- Si se recibe la indicación de finalización de conteo en la trama se realiza transición al estado “guarda datos”.

**Guarda datos.** Añade filas de datos al archivo de texto con los datos obtenidos en el conteo. Las acciones y transiciones son:

- Obtener los valores promedios de los sensores durante el conteo.
- Mostrar y almacenar los datos adquiridos. En el caso de los transmisores los valores promedios. En el caso de los pulsos los últimos valores del conteo.
- Si se realizaron todas las repeticiones definidas en el control numérico “Repeticiones” se realiza la transición al estado “programación”, caso contrario la transición será al estado “grabación en uC” para empezar un nuevo conteo.

**Detener proceso.** Interrumpe la ejecución del programa. Las acciones y transiciones son:

- Enviar el comando 'd' al microcontrolador.
- Finalizar la ejecución del bucle productor.

Por otra parte, el consumidor realiza: obtención de la trama, separación de datos y conexión a elementos visuales. este funcionamiento se muestra como diagrama de flujo en la Figura 61.



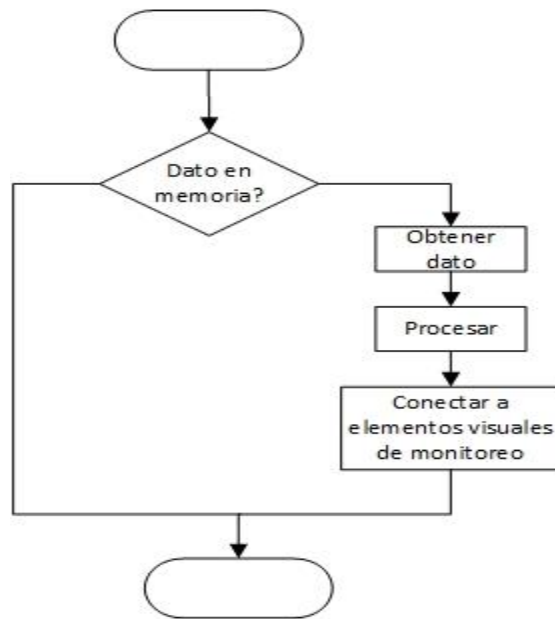


Figura 61. Diagrama de flujo para el consumidor.  
Fuente: Elaboración propia.



## 5. Capítulo 5 - Prototipo.

Cada uno de los elementos que componen al sistema, identificados en el punto 4.2, se implementan e integran en un prototipo. La implementación se realiza con base en los diseños del capítulo anterior. La integración se efectúa teniendo en cuenta la arquitectura de un sistema de adquisición basado en computador (Figura 34). A continuación, se explica este proceso para cada parte.

### 5.1. HARDWARE.

Como se mencionó en el punto 4.2.1 el Hardware necesario para el sistema se compone de 3 módulos los cuales son:

- Acondicionamiento de las señales de transmisores (M1),
- Acondicionamiento y sincronización de señales de pulsos (M2),
- Dispositivo de procesamiento y comunicación (M3).

A su vez, cada uno de estos módulos contiene submódulos. Estos últimos se pueden implementar mediante un circuito electrónico apropiado.

Para la construcción del prototipo, se considera componentes disponibles en las tiendas de electrónica de Bolivia. También, la disposición y configuración de dichos componentes puede estar dada en la hoja de datos (Datasheet) o corresponder a circuitos de aplicación típica.

A continuación, se detalla los circuitos, cálculo de componentes, diagrama esquemático, tarjeta de circuito impreso de los módulos de hardware M1, M2 y M3.

#### 5.1.1. ACONDICIONAMIENTO DE SEÑALES DE TRANSMISORES M1.

En la Figura 62 se muestra nuevamente los submódulos que componen al módulo M1.



Figura 62. Submódulos para entrada de transmisor de 2 o 4 hilos.

Fuente: Elaboración propia

A continuación, se explica el desarrollo de cada submódulo y finalmente el circuito resultante para el módulo M1.

### Submódulo M11.

El submódulo M11 tiene un elemento de selección para entrada de transmisor de 2 o 4 hilos y convierte la señal de corriente a voltaje. La relación de entrada y salida está dada por la Ecuación 7.

Para la selección de tipo de transmisor (2 o 4 hilos) se toma en cuenta los esquemas de conexión mostrados en el punto 2.4 del capítulo de marco teórico. Entonces, se usa un conmutador bipolar; de tal forma que acepte entrada de transmisor de 2 o 4 hilos dependiendo de la posición del conmutador. En la Figura 63 se muestra el esquema de conexión con el conmutador bipolar y los circuitos resultantes para cada posición.

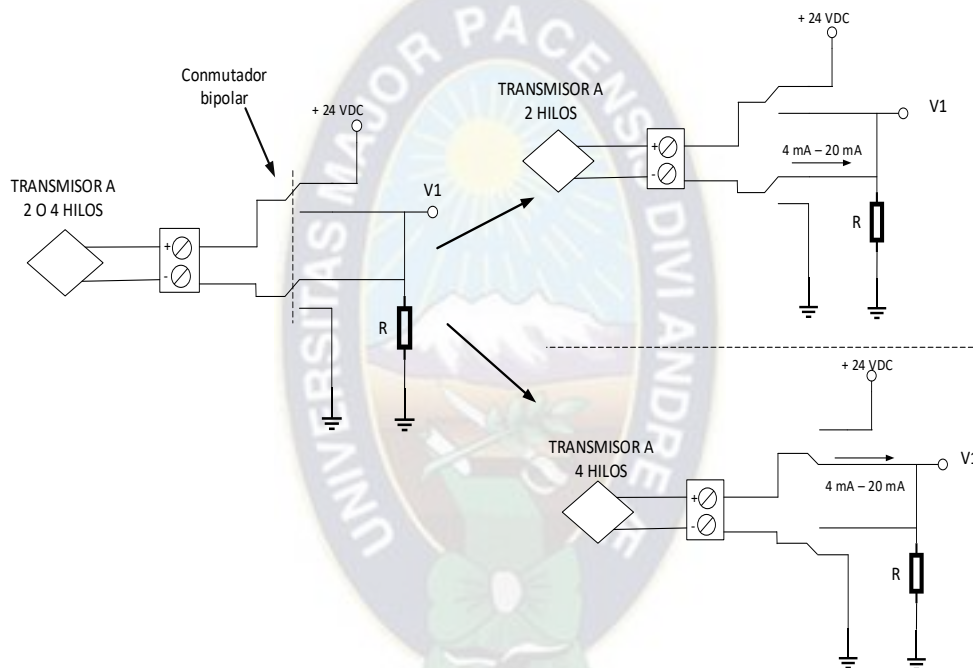


Figura 63. Selección de tipo de transmisor con conmutador bipolar.

Fuente: Elaboración propia

Como se puede ver en la figura anterior, al conectar el transmisor de 2 o 4 hilos, la entrada es una señal de corriente  $I_i$ . Dicha señal pasa por un resistor (R) y por la ley de ohm se tendrá:

$$V1 = R * I_i \dots \dots (11) \quad 4 \text{ mA} \leq I_i \leq 20 \text{ mA}$$

Ecuación 11. Relación submódulo M11.

Para la selección de la resistencia R, se toma en cuenta las características de los transmisores que se muestran en la Tabla 9.

Magnitud Medida	Marca	Modelo	Región de operación
Temperatura	Endress Hausser	TMT180	$0 \leq R \leq \frac{U - 10}{0.022} : [\Omega]$ $10 \leq U \leq 35 : [VDC]$
Presión	Yokogawa	EJA530E	$250 \leq R \leq \frac{U - 10,5}{0.0244} : [\Omega]$ $16,6 \leq U \leq 42 : [VDC]$
Caudal	Endress Hausser	IFC110	$15 \leq R \leq 500 : [\Omega]$ $U = 220 : [VAC]$

Tabla 9. Resumen de características de los transmisores.

Fuente: Elaboración propia.

Se usa una fuente de 24 VDC, es decir,  $U=24$  VDC. Con este valor se tiene las regiones de operación dadas por:

- $0 \leq R \leq 795,45 : [\Omega]$  (12)

*Ecuación 12. Región de operación transmisor de temperatura.*

- $250 \leq R \leq 553,28 : [\Omega]$  (13)

*Ecuación 13. Región de operación transmisor de presión.*

- $15 \leq R \leq 500 : [\Omega]$  (14)

*Ecuación 14. Región de operación transmisor de caudal.*

En base a los valores de rango de operación obtenidos en los transmisores se selecciona un valor de resistencia de carga que cumpla con la región de operación en todos los casos. Para ello, se intercepta las regiones de operación dados por las ecuaciones (12), (13) y (14).

$$0 \leq R \leq 795.45 \cap 250 \leq R \leq 553.28 \cap 15 \leq R \leq 500$$

$\rightarrow 250 \leq R \leq 500 : [\Omega]$

Se usa una resistencia de carga de  $R = 270[\Omega]$  por ser un valor comercial. Con este valor y aplicando la ley de OHM la salida de tensión se encuentra en el siguiente rango:

$$\begin{aligned} \text{si } I_i = 4 \text{ mA} &\rightarrow V1 = 1.08 \text{ V} \\ \text{si } I_i = 20 \text{ mA} &\rightarrow V1 = 5.4 \text{ V} \end{aligned}$$

Finalmente, el circuito para el submódulo M11 se muestra en la Figura 64.

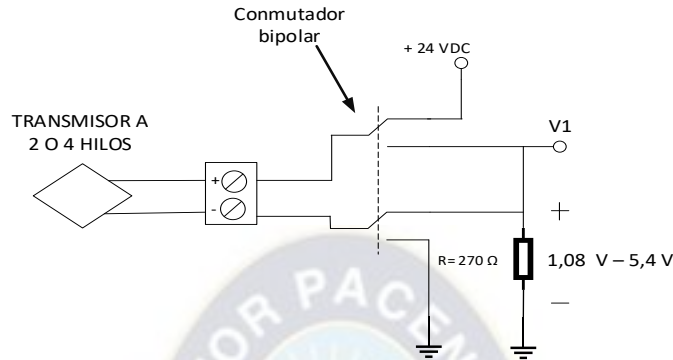


Figura 64. Circuito para submódulo M11.

Fuente: Elaboración propia.

### Submódulo M12.

El submódulo M12 realiza el aislamiento de la señal analógica, la relación de entrada a salida está dada por la Ecuación 8.

Se usa un dispositivo de aislamiento óptico. El diseño se basa en el circuito integrado SFH600. Este es un optoacoplador que presenta una zona de trabajo lineal entre 1 mA y 10 mA. En la Figura 65 se muestra el circuito de operación lineal.

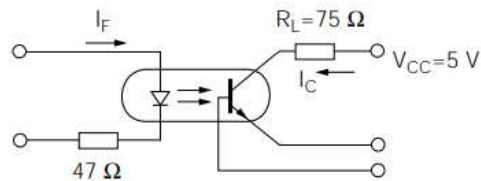


Figura 65. Operación lineal del SFH600.

Fuente: Elaboración propia.

El submódulo anterior (M11) tiene salida de voltaje en el intervalo  $1,8 \leq V1 \leq 5,4$ . Entonces, antes del circuito de aislamiento mostrado en la Figura 65 es necesaria una etapa de conversión de voltaje a corriente antes del optoacoplador. Dicha etapa deberá tener una salida en el intervalo  $1 \text{ mA} \leq I \leq 10 \text{ mA}$  para el uso del SFH600 en su zona de trabajo lineal. En la Figura 66 se muestra el circuito de conversión de voltaje a corriente usado con amplificador operacional AO LM358.

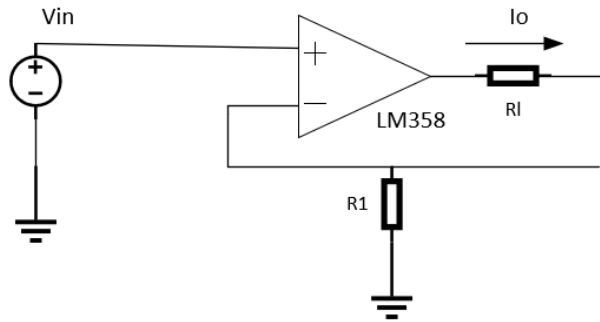


Figura 66. Circuito conversor de voltaje a corriente.

Fuente: Elaboración propia.

La corriente de salida  $I_o$ , en el circuito anterior, se define según:

$$I_o = \frac{V_1}{R_1} \quad (15)$$

Ecuación 15. Conversor voltaje a corriente.

El mayor nivel de voltaje en la salida del submódulo M11 es 5,4 V, lo deseable es que a este nivel se tengan 10 mA, a partir de la ecuación (15) se tiene:

$$R_1 = \frac{V_{in \max}}{10 \text{ mA}} = \frac{5,4 \text{ V}}{10 \text{ mA}} \rightarrow R_1 = 540 [\Omega]$$

Se toma el valor comercial  $R_1 = 510 [\Omega]$ , con este valor se tendrá:

$$\text{Si } V_{in} = 1,08 \text{ V} \rightarrow I_o = \frac{1,08 \text{ V}}{510 \Omega} = 2,12 \text{ mA}$$

$$\text{Si } V_{in} = 5,4 \text{ V} \rightarrow I_o = \frac{5,4 \text{ V}}{510 \Omega} = 10,6 \text{ mA}$$

De esta forma se trabaja en la zona de operación lineal del SFH600.

A la salida del circuito conversor de voltaje a corriente, se acopla el circuito de aislamiento mostrado en la Figura 67.

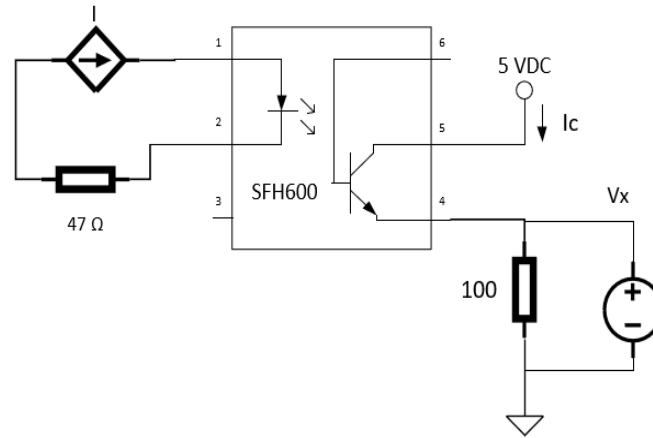


Figura 67. Circuito de aislamiento con SFH600.

Fuente: Elaboración propia.

Como se opera en la región lineal, la corriente de salida de la foto transistor ( $I_c$ ) será la misma que la corriente de entrada al fotodiodo. Del circuito anterior, se sabe que la corriente de entrada varía entre  $2,12 \text{ mA}$  y  $10,6 \text{ mA}$ . Para obtener una señal de voltaje se usa una resistencia de  $100 \text{ } [\Omega]$  entre el emisor y masa, aplicando la ley de Ohm el voltaje  $V_2$  será:

$$\text{Si } I_c = 2,12 \text{ mA} \rightarrow V_2 = 2,12 \text{ mA} * 100 \text{ } \Omega \rightarrow V_2 = 0.212 \text{ V}$$

$$\text{Si } I_c = 10,6 \text{ mA} \rightarrow V_2 = 10,6 \text{ mA} * 100 \text{ } \Omega \rightarrow V_2 = 1.06 \text{ V}$$

Finalmente, acoplando el circuito de aislamiento con SFH600 al circuito de conversión de voltaje a corriente, se tiene el circuito del submódulo M12 que se muestra en la Figura 68.

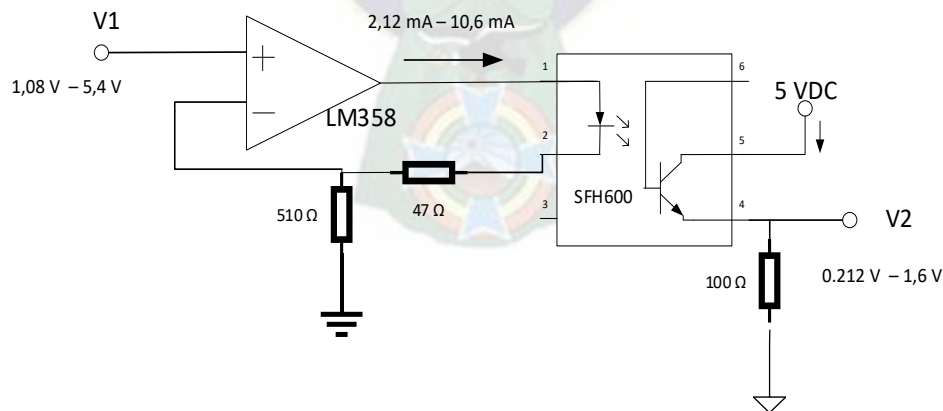


Figura 68. Circuito del submódulo M12.

Fuente: Elaboración propia.



### Submódulo M13.

El submódulo M13 reduce o eleva el nivel de señal de salida del submódulo de M12. Para este fin se usa un circuito amplificador no inversor con AO. Del submódulo anterior se tiene una señal de voltaje  $V_2$  entre  $0,212\text{ V}$  y  $1,06\text{ V}$ . Se amplifica esta señal para tener un rango mayor de tensión antes de que ingrese al ADC. En esta última etapa se usa el amplificador no inversor que se muestra en la Figura 69.

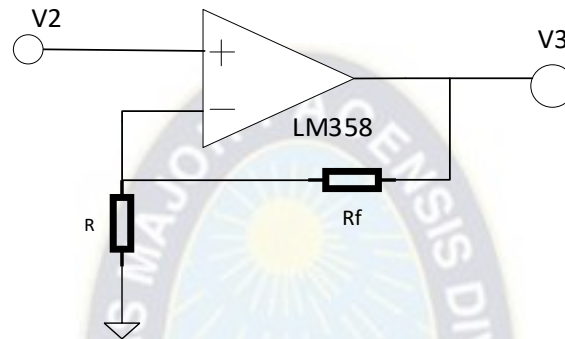


Figura 69. Circuito amplificador no inversor.

Fuente: Elaboración propia.

La ganancia del amplificador está dada por:

$$\frac{V_3}{V_2} = 1 + \frac{R_f}{R} \dots \dots (16)$$

Ecuación 16. Amplificador no inversor.

Se elige la ganancia  $\frac{V_3}{V_2} = 3$ ; entonces de la ecuación (16) se tiene:

$$2,5 = 1 + \frac{R_f}{R} \rightarrow R_f = 2 * R$$

Con  $R = 1\text{ k}[\Omega]$ , entonces  $R_f = 2\text{ k}[\Omega]$ .

Finalmente, como  $V_2$  varía entre  $0,212\text{ V}$  y  $1,06\text{ V}$ , con la ganancia del amplificador no inversor ( $\frac{V_3}{V_2} = 3$ ), el voltaje de salida del circuito estará entre  $0,636\text{ V}$  y  $3,18\text{ V}$ . El circuito para el submódulo M13 se muestra en la Figura 70.



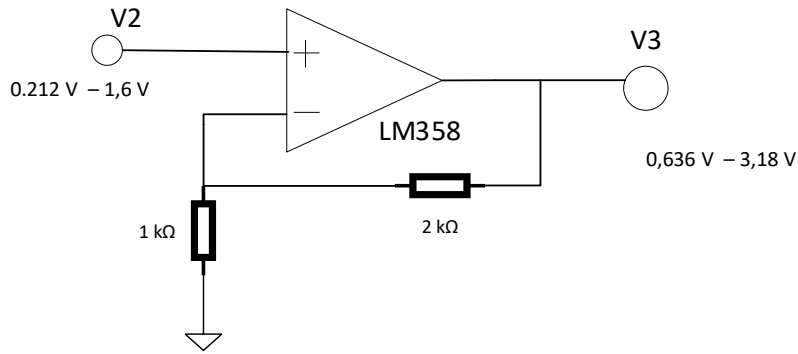


Figura 70. Circuito del submódulo M13.

Fuente: Elaboración propia.

### Submódulo M14.

El submódulo M14 refiere a un filtro pasa bajo. Su función de transferencia está dada por la Ecuación 10.

Para este caso se utiliza un circuito RC como el que se muestra en la Figura 71.

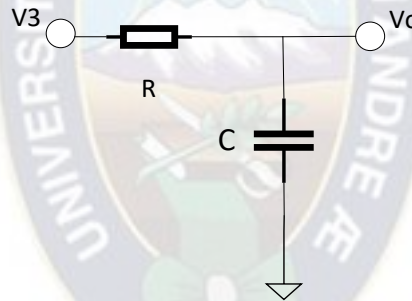


Figura 71. Filtro RC.

Fuente: Elaboración propia.

Para el circuito anterior la función de transferencia está definida por:

$$\frac{V_o}{V_3} = \frac{1}{SCR + 1} \dots \dots (17)$$

Ecuación 17. Filtro pasa bajo RC.

En la anterior ecuación la frecuencia de corte está definida por:

$$fc = \frac{1}{2\pi CR}$$

Se elige  $fc = 100 \text{ Hz}$  y  $C = 1 \mu\text{F}$  entonces se tiene

$$R = \frac{1}{2\pi Cfc} \rightarrow R = \frac{1}{2\pi * 10^{-6} * 100} \rightarrow R = 1,59 \text{ k}\Omega$$

Se toma el valor comercial  $R = 1,5 \text{ k}\Omega$ , entonces se tiene:

$$f_c = \frac{1}{2\pi CR} \rightarrow f_c = \frac{1}{2\pi * 10^{-6} * 1,5 * 10^3} \rightarrow f_c = 106 \text{ Hz}$$

Con esta frecuencia de corte se evita componentes de alta frecuencia (ruido). Finalmente, el circuito para el submódulo M14 se muestra en la Figura 72.

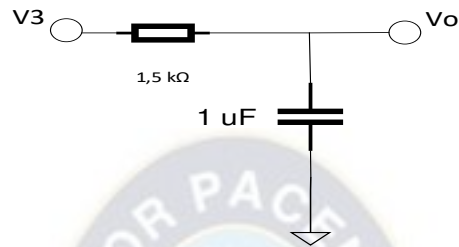


Figura 72. Circuito para submódulo M14.

Fuente: Elaboración propia.

### Modulo M1.

El módulo M1 para acondicionamiento de las señales de los transmisores, se logra uniendo los submódulos M11, M12, M13 y M14. El circuito resultante se muestra en la Figura 73.

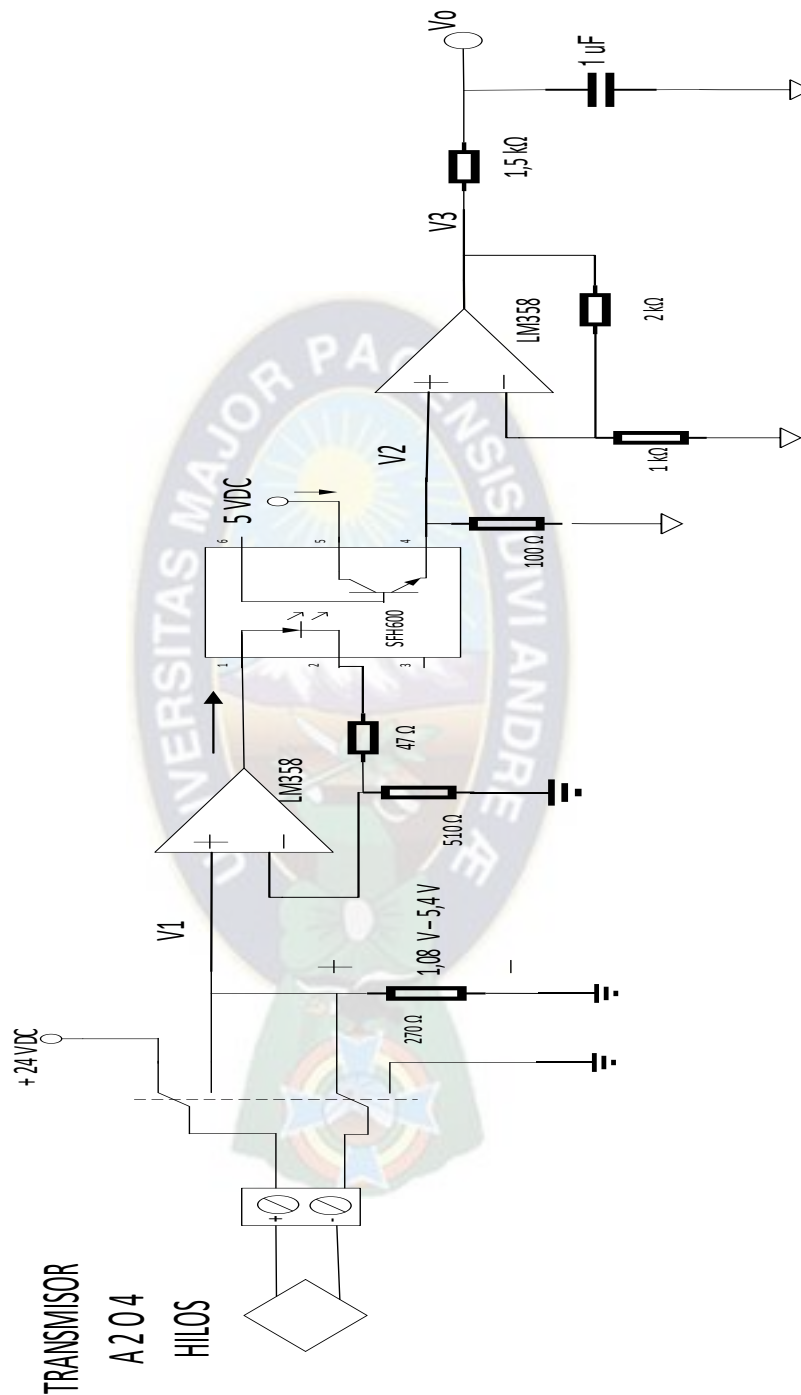


Figura 73. Circuito para módulo M1.

Fuente: Elaboración propia.

Cada canal deberá tener un circuito como el que se muestra en la Figura 73. Para obtener mediciones adecuadas en modulo M1, se realiza un análisis estableciendo la relación entre

corriente de entrada y voltaje de salida en cada canal. Para dicho estudio, se considera al circuito como caja negra según se muestra en la Figura 74. Para el análisis se toman medidas de corriente de entrada (I) y voltaje de salida (Vo) y se establece la relación con regresión. Las medidas y ecuaciones obtenidas para cada canal se muestran en el anexo E.

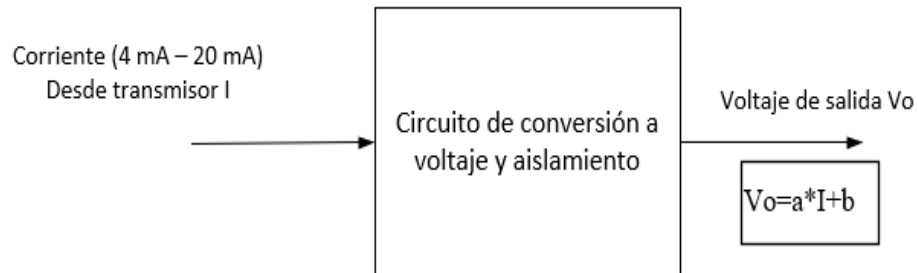


Figura 74. Relación voltaje corriente.

Fuente: Elaboración propia.

### 5.1.2. ACONDICIONAMIENTO Y SINCRONIZACIÓN DE SEÑALES DE PULSOS M2.

En la Figura 75 se muestra nuevamente los submódulos que componen al módulo M2.

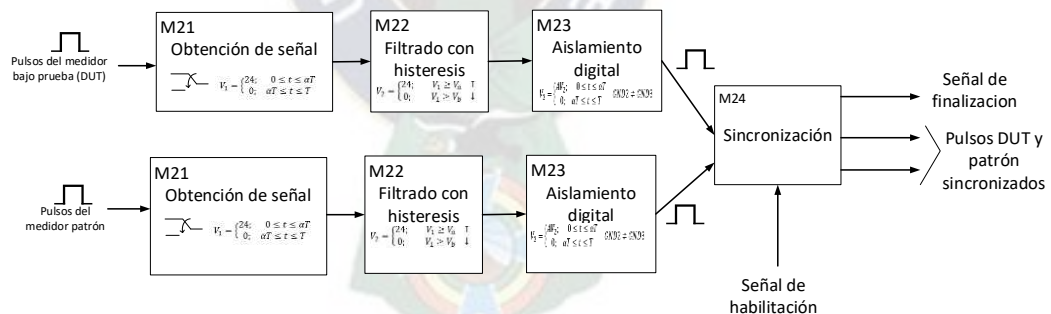


Figura 75. Elementos del módulo de M2.

Fuente: Elaboración propia.

A continuación, se explica el desarrollo de cada submódulo.

### Submódulo M21.

El submódulo M21 selecciona el tipo de entrada de pulsos, que puede ser activa o pasiva. Para este circuito se tiene en cuenta los esquemas mostrados en el punto 4.2.1. Se usa un conmutador bipolar para la selección de tipo de entrada de pulsos. En la Figura 76 se muestra el circuito para el submódulo M21 con los circuitos resultantes en las posiciones del conmutador.

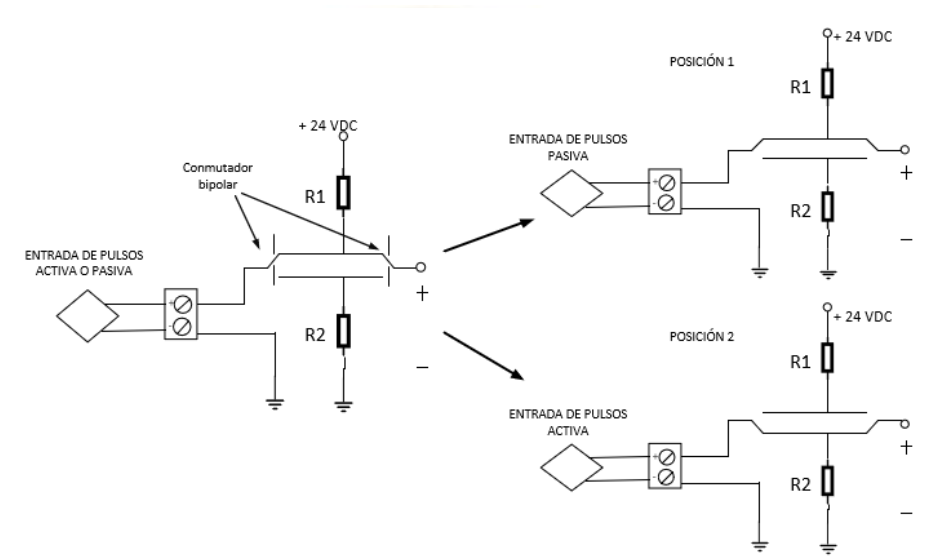


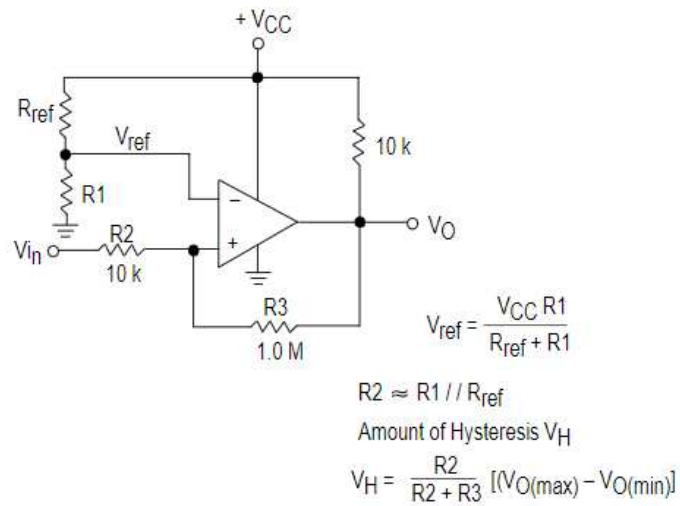
Figura 76. Circuito submódulo M21.

Fuente: Elaboración propia.

Para el circuito se usa una fuente de 24 VDC, la corriente que circula se puede limitar a 10 mA, entonces para ambos casos (R1 y R2) se debe usar una resistencia de  $R1 = R2 = \frac{24 V}{10 mA} \rightarrow R1 = R2 = 2.4 k\Omega$ .

### Submódulo M22.

Como se explicó en capítulo de diseño las señales de pulsos obtenidas en el submódulo M21 pueden presentar ruido entonces se hace necesario el filtro de histéresis. Para la implementación de esta etapa se opta por un circuito integrado LM 339 que cuenta con aplicación típica de comparación por histéresis. Este dispositivo soporta niveles tensión de hasta 36 V. En la Figura 77 se muestra el circuito de aplicación comparador con histéresis no inversor.



*Figura 77. Circuito comparador con histéresis no inversor.  
Fuente: Tomado de hoja de datos LM339.*

De circuito mostrado en la Figura 77 el valor de referencia y cantidad de histéresis se definen por la Ecuación 18 y Ecuación 19 respectivamente.

$$V_{ref} = \frac{V_{CC} R_1}{R_{ref} + R_1} \dots \dots (18)$$

*Ecuación 18. Valor de referencia en comparador con histéresis.*

$$V_H = \frac{R_2}{R_2 + R_3} [V_{O_{max}} - V_{O_{min}}] \dots \dots (19)$$

*Ecuación 19. Cantidad de histéresis para comparador.*

Tomando los valores de las resistencias:

$$R_{ref} = 10 \text{ k}\Omega; R_1 = 10 \text{ k}\Omega; R_3 = 1 \text{ M}\Omega$$

Con estos valores se tiene en la ecuación (18) se tiene:

$$V_{ref} = \frac{V_{CC} R_1}{R_{ref} + R_1} \rightarrow V_{ref} = \frac{24 \text{ V} * 10 \text{ k}\Omega}{10 \text{ k}\Omega + 10 \text{ k}\Omega}$$

$$\rightarrow V_{ref} = 12 \text{ V}$$

$$R_2 = R_1 // R_3 = 10 \text{ k}\Omega // 10 \text{ k}\Omega = 5 \text{ k}\Omega$$

$$\rightarrow R_2 \approx 5,1 \text{ k}\Omega$$

Entonces la cantidad de histéresis dado en la ecuación (19) queda:

$$V_H = \frac{R_2}{R_2 + R_3} [V_{Omax} - V_{Omin}] \rightarrow V_H = \frac{5,1 \text{ k}\Omega}{5,1 \text{ k}\Omega + 1 \text{ M}\Omega} [24 \text{ V} - 0 \text{ V}]$$

$$\rightarrow V_H = 0,12 \text{ V}$$

Finalmente, el circuito comparador no inversor con histéresis para el submódulo M22 se muestra en la Figura 78.

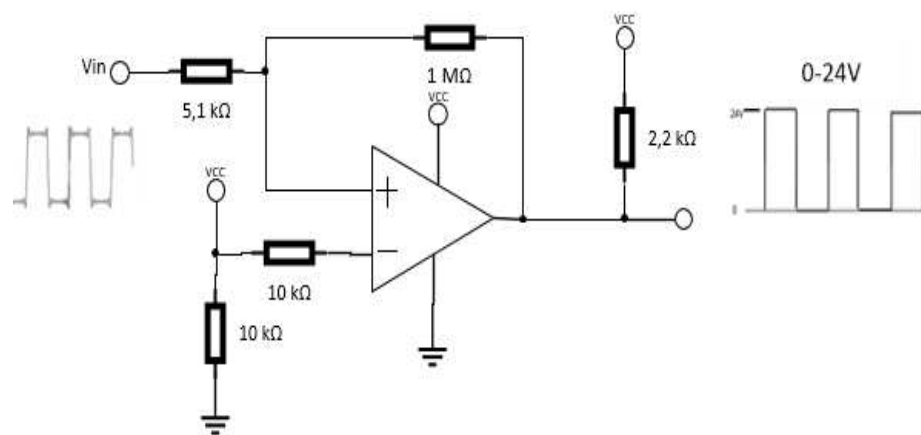


Figura 78. Circuito del submódulo M22.

Fuente: Elaboración propia.

### Submódulo M23.

Para el aislamiento digital se usa un optoacoplador 4N35 seguido de un comparador con histéresis Schmitt trigger 7414. El 4N35 consta de un diodo emisor y un fototransistor que se satura en el ciclo positivo de la señal de pulsos, usando una fuente de 5 VDC en el fototransistor del 4N35 se reduce los niveles. Además, este componente aísla las señales entre los medidores de flujo líquido y el circuito de sincronización. Adicionalmente se pone un diodo Zener en la entrada del 4N35 como componente de protección en caso de sobretensión. El circuito integrado 7414 estabiliza la señal a los niveles de 0 V – 5 V. El circuito propuesto para el módulo M23 se muestra en la Figura 79.

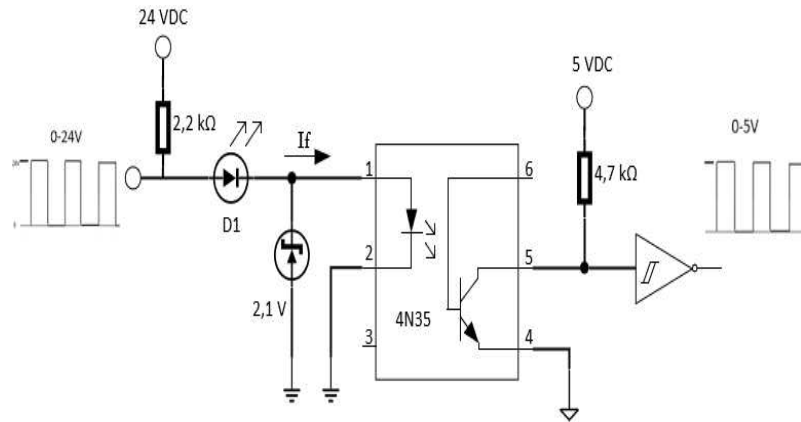


Figura 79. Circuito del módulo M23.

Fuente: Elaboración propia.

#### Submódulo M24.

En la Figura 80 se muestra nuevamente el diagrama de tiempos para el funcionamiento descrito en el punto 4.2.1 del capítulo de arquitectura y diseño.

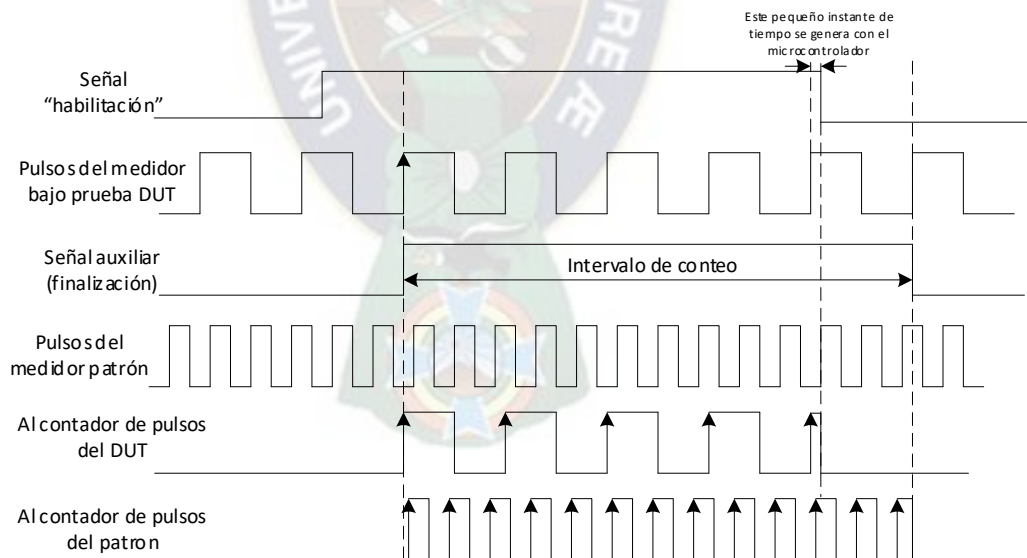


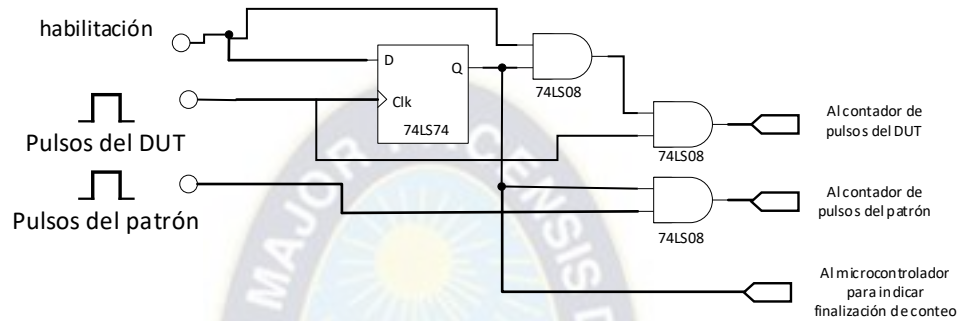
Figura 80. Diagrama de tiempos para la sincronización de los pulsos.

Fuente: Elaboración propia.

De acuerdo al diagrama de tiempos mostrado en la Figura 80 se usa un circuito de sincronización que tiene tres (3) entradas: La señal de “habilitación”, la señal de pulsos del medidor bajo prueba (IBP) y la señal de pulsos del medidor patrón; y una salida para indicar al microcontrolador la



finalización del conteo. Para generar la señal auxiliar donde está el intervalo de conteo se usó flip flop D 74LS74; el flanco descendente de esta señal indicara al microcontrolador la finalización del conteo. Para generar las señales que ingresan a los contadores de pulsos se utilizaron compuertas lógicas AND 74LS08. El circuito para la sincronización se muestra en la Figura 81.



*Figura 81. Circuito para sincronización de los pulsos del medidor bajo prueba y patrón.  
Fuente: Elaboración propia.*

## Módulo M2.

Con base en el esquema de la Figura 75 se une los circuitos de cada submódulo (M21, M22, M23 y M24) y se obtiene el circuito para el módulo M2 resultando en el circuito mostrado en la Figura 82.

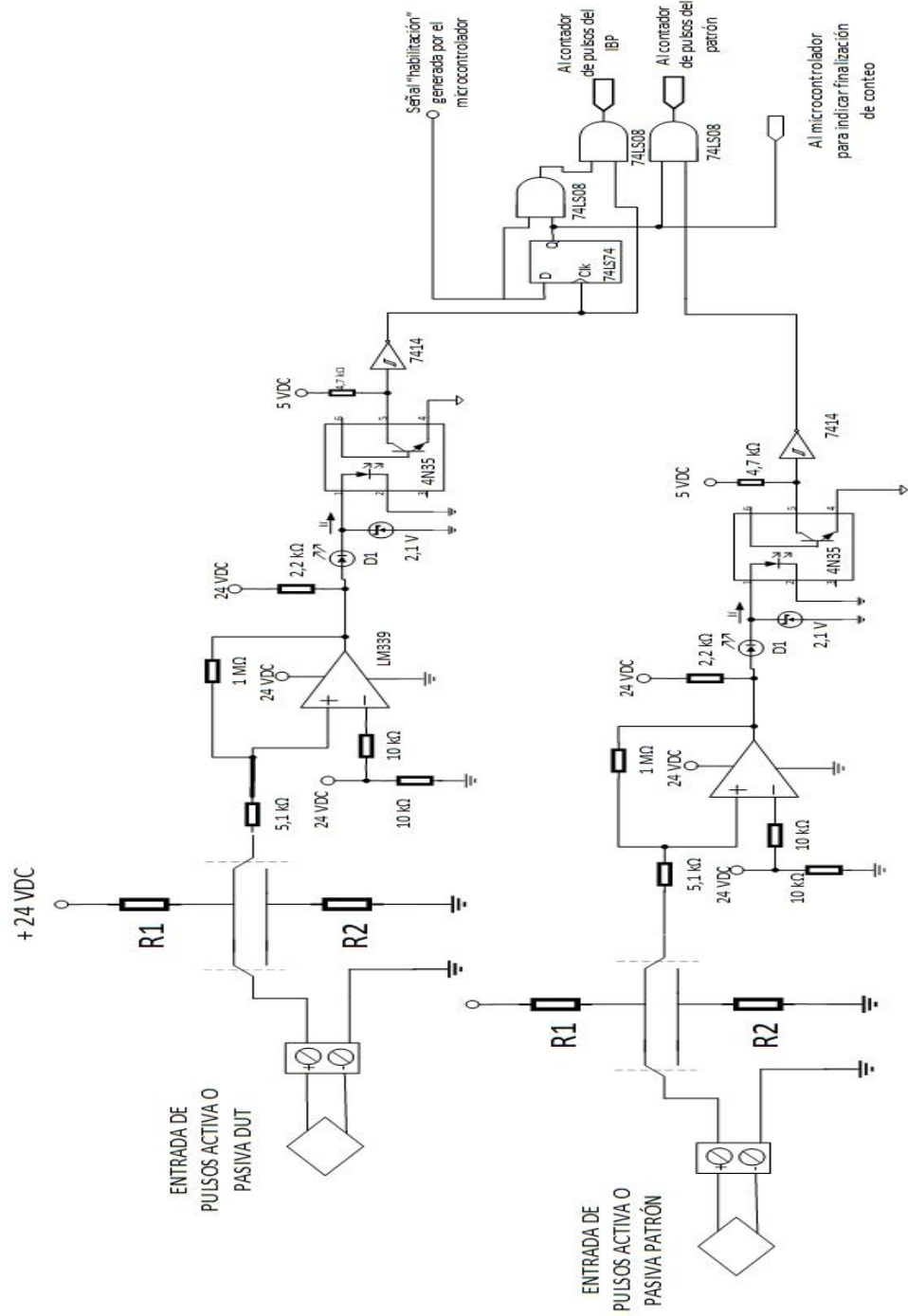


Figura 82. Circuito de módulo M2.

Fuente: Elaboración propia.

## 5.2. PROGRAMACION DEL DP.

El DP que puede ser un microcontrolador, microprocesador, FPGA (*Field Programmable Gate Array*) o un controlador lógico programable (*Programmable Logic Controller – PLC*) siempre y cuando posea los módulos que se describieron en el punto 4.2.2. que son:

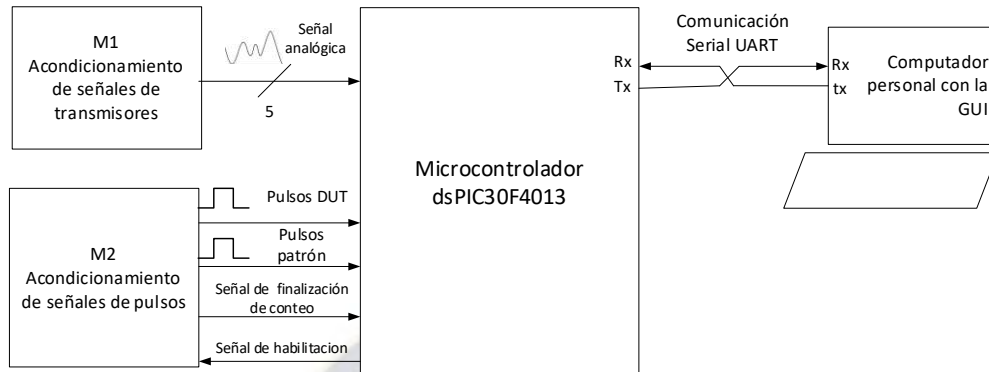
- Módulo ADC con 5 canales como mínimo y resolución mayor a 10 bits (M1).
- Módulo de conteo con 2 canales como mínimo de 32 bits (M2).
- Módulo de entradas y salidas digitales de propósito general de 2 canales como mínimo (M3).
- Módulo de comunicación serial con protocolos RS232 o RS485 (M4).

Tomando en cuenta las tareas descritas en el punto 4.2.2 para el DP, se tienen 5 entradas analógicas, 3 entradas digitales, 1 entrada de recepción de datos (Rx), 1 salida digital y una salida de transmisión de datos (Tx). Debido a sus módulos periféricos (Tabla 10) se elige un microcontrolador DSPIC30F4013 como DP. En la Figura 83 se muestra las entradas y salidas del DSPIC30F4013

Pines	Temporizadores (Timer) 16-bit	Salida PWM/Comparación	Conversor A/D 12 bit	UART	SPI	I2C	CAN
40/44	5	4	13	2	1	1	1

Tabla 10. Periféricos del dsPIC30F4013.

Fuente: Hoja de datos del DSPIC30F4013.



**Figura 83. Esquema de interacción de microcontrolador y elementos del sistema.**  
*Fuente: Elaboración propia.*

Antes de escribir el código de programación es necesario elegir los módulos y pines a usar para un correcto funcionamiento. Esta designación se hace en función a las entradas y salidas mostradas en la Figura 83 y con ayuda de lo explicado en el punto 2.7 del Marco Teórico. En la Tabla 11 se muestra los módulos y pines seleccionados para el microcontrolador. Se debe mencionar que el DSPIC30F4013 cuenta con la posibilidad de configurar un Timer en modo contador de 32 bits el cual se asigna para el conteo de pulsos del DUT. En el caso del contador para los pulsos del patrón se asigna un timer de 16 bits.

<i>Entrada / Salida</i>	<i>Señal que corresponde</i>	<i>periférico designado</i>	<i>nombre del pin</i>	<i>pin que corresponde en el microcontrolador</i>	<i>Descripción</i>
Entrada	Pulsos DUT	Timer 2 /3	T2CK	15	El timer 2/3 de 32 bits se usa para en modo contador para los pulsos del DUT.
Entrada	Pulsos patrón	Timer 1	T1CK	16	El timer 1 de 16 bits se usa en modo contador para los pulsos del patrón.
Entrada	Finalización de conteo	Interrupción Externa 2	INT2	18	Se identificará la finalización del conteo con el flanco

					descendente de esta señal.
Entrada	Señal analógica de los transmisores	ADC canal 0	AN0	2	Conversión de la señal que proviene de los transmisores.
Entrada	Señal analógica de los transmisores	ADC canal 1	AN1	3	Conversión e la señal que proviene de los transmisores.
Entrada	Señal analógica de los transmisores	ADC canal 2	AN2	4	Conversión de la señal que proviene de los transmisores.
Entrada	Señal analógica de los transmisores	ADC canal 3	AN3	5	Conversión de la señal que proviene de los transmisores.
Entrada	Señal analógica de los transmisores	ADC canal 4	AN4	6	Conversión de la señal que proviene de los transmisores.
Salida	Señal habilitación	Pin I/O de propósito general	RA11	17	Es la señal para habilitación del conteo de pulsos que va al circuito de acondicionamiento y sincronización de pulsos
Entrada	Señal Rx	UART 2	U2RX	28	Recepción de datos del módulo de comunicación serial
Salida	Señal Tx	UART 2	U2TX	27	Transmisión de datos del módulo

					de comunicación serial
--	--	--	--	--	------------------------

Tabla 11. Designación de módulos y pines del microcontrolador.  
Fuente: Elaboración propia.

En la Figura 84 se muestra los pines designados sobre el microcontrolador y las señales de entrada o salida a las que corresponden.

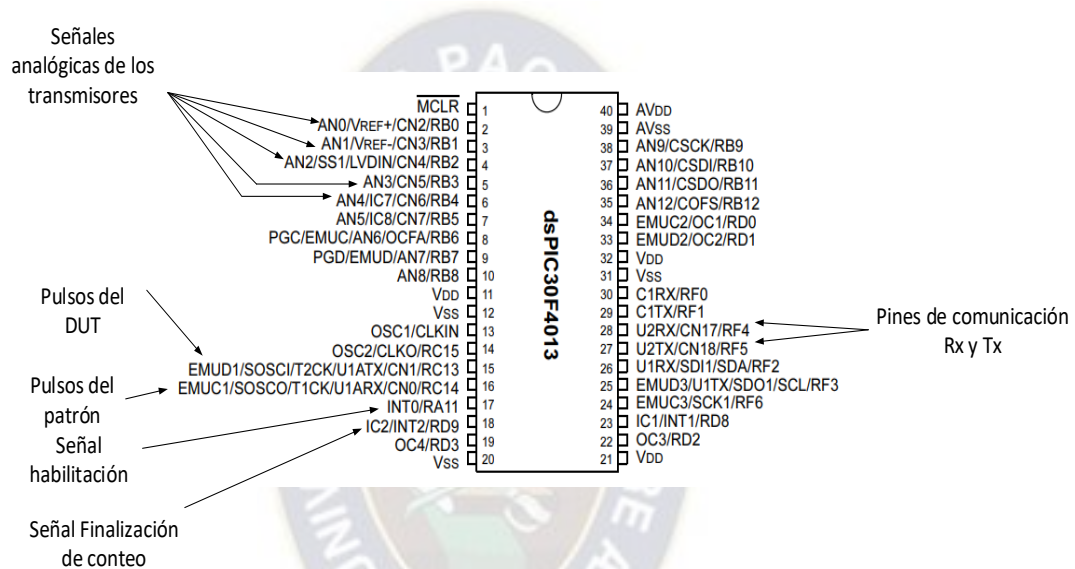


Figura 84. Designación de pines en el microcontrolador.  
Fuente: Elaboración propia.

Para la programación del microcontrolador se sigue la estructura mostrada en la Tabla 12.

<p><i>Definición de librerías y archivos de cabecera;</i></p> <p><i>Declaración de constantes;</i></p> <p><i>Declaración de variables;</i></p> <p><i>Definición de funciones;</i></p> <p><i>Funciones de servicio de atención a interrupciones;</i></p> <p><i>Función principal "main";</i></p> <p><i>Configuración de módulos;</i></p> <p><i>Habilitación de interrupciones;</i></p> <p><i>Bucle infinito "While (1)";</i></p> <p><i>Descripción de funciones;</i></p>
---

*Tabla 12. Estructura para programación del microcontrolador DSPIC30F4013*  
*Fuente: Elaboración propia.*

En la escritura del código inicialmente se define las variables necesarias, se continua con las funciones de los módulos de microcontrolador y posteriormente se ordena dichas funciones. Para ello, se sigue las descripciones del punto 4.2.2 del capítulo de Desarrollo del Sistema y del punto 2.7 del capítulo de Marco Teórico.

### **5.2.1.DEFINICION DE VARIABLES PARA EL PROGRAMA.**

Para la comprensión del código es necesario definir las variables que intervienen y son necesarias en el programa. Las mismas se detallan en la Tabla 13.

<b>VARIABLE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
an0	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
an1	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
an2	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
an3	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
an4	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
an5	Entero 16 bits	Para almacenamiento de la lectura del canal 1 del módulo ADC
count_dut	Entero 32 bits	Para almacenamiento de la lectura del contador de pulsos del DUT
count_patron	Entero 32 bits	Para almacenamiento de la lectura del contador de pulsos del Patrón
count_patron_high	Entero 16 bits	Como se asignó un Timer de 16 bits, esta variable permite realizar cuentas de hasta 32 bits
cuentas_programadas	Entero 32 bits	la cantidad de pulsos a contar antes de detener el conteo



conteo_str	Cadena de caracteres	Almacena el dato de pulsos a contar como cadena de caracteres
rxbyte	Carácter 8 bits	Almacena el carácter recibido en el puerto de comunicación
estado1	Entero 8 bits	variable auxiliar para la recepción de cadena de caracteres
trama	Cadena de caracteres	Almacena la cadena de caracteres que se enviara a la GUI

*Tabla 13. Definición de variables para el programa.*

*Fuente: Elaboración propia.*

## **5.2.2. FUNCIONES DE LOS MÓDULOS.**

Con las variables definidas y pines designados se puede escribir el código para cada función de los módulos que se explicaron en el punto 4.2.2 del capítulo de Desarrollo del Sistema.

### **5.2.2.1. MODULO ADC.**

La función denominada F1M1 para configuración del módulo ADC se renombra como “init\_adc” para mejor legibilidad. No tiene parámetros de entrada ni salida, su código se muestra en la Tabla 14. Para esta parte de código se considera la siguiente configuración:

- Niveles de referencia entre VDD y VSS, es decir, los valores de alimentación del microcontrolador (0 V - 5 V).
- Conversión inicia automáticamente después del muestreo.
- Resolución de 12 bits, es decir valores de lecturas analógicas entre 0 y 4095.
- No se usan interrupciones para este módulo.

```

void init_adc() {
    TRISBbits.TRISB0 = 1; //pin B0 como entrada
    TRISBbits.TRISB1 = 1; //pin B1 como entrada
    TRISBbits.TRISB2 = 1; //pin B2 como entrada
    TRISBbits.TRISB3 = 1; //pin B3 como entrada
    TRISBbits.TRISB4 = 1; //pin B4 como entrada
    TRISBbits.TRISB5 = 1; //pin B5 como entrada
    ADPCFG = 0xFFC0; //pines B0-B5 como analógicos
    ADCON1 = 0x0000;
    ADCON1bits.SSRC = 0b111; //inicia conversión
    ADCON2 = 0x0000;
    ADCON3bits.ADCS = 0b011010; //para garantizar Tad=667ns
    ADCON3bits.SAMC = 5; //15Tad entre muestreo y conversión
    ADCHSbits.CH0NA = 0; //negativo del ADC a Vref-
    ADCSSL = 0x0000; //no se realiza escaneo
}

```

Tabla 14. Código de función *init\_adc*.

Fuente: Elaboración propia.

La función denominada F2M1 para lectura de canal del módulo ADC se renombra como “leer\_adc”. Como parámetro de entrada recibe el número de canal y como parámetro de salida el valor de lectura analógica. Su código se muestra en la Tabla 15.

```

int leer_adc(int canal) {
    int lectura = 0;
    switch (canal) { //se elige el canal AN positivo del ADC
        case 0: ADCHSbits.CH0SA = 0b0000; // si la entrada es 0 elige canal 0
            break;
        case 1: ADCHSbits.CH0SA = 0b0001; // si la entrada es 1 elige canal 1
            break;
        case 2: ADCHSbits.CH0SA = 0b0010; // si la entrada es 2 elige canal 2
            break;
        case 3: ADCHSbits.CH0SA = 0b0011; // si la entrada es 3 elige canal 3
            break;
        case 4: ADCHSbits.CH0SA = 0b0100; // si la entrada es 4 elige canal 4
            break;
        case 5: ADCHSbits.CH0SA = 0b0101; // si la entrada es 5 elige canal 5
            break;
        default: ADCHSbits.CH0SA = 0b0000; // por defecto se elige canal 0
            break;
    }
    ADCON1bits.SAMP = 1; //inicia muestreo entonces 31Tad después la
    conversión
    while (!ADCON1bits.DONE); //verifica si la conversión esta lista
    lectura = ADCBUF0;
    return lectura; //retorna el valor de la lectura
}

```

Tabla 15. Código de función leer\_adc.

Fuente: Elaboración propia.

#### 5.2.2.2. MODULO DE CONTEO.

La función denominada F1M2 para configuración del módulo de conteo se realiza para el contador de pulsos del DUT y patrón. Entonces se tienen dos funciones que se nombran como “init\_timer230)” para los pulsos del DUT e “init\_timer10)” para el patrón. Ambas funciones no tienen parámetros de entrada ni salida. En la Tabla 16 y Tabla 17 los códigos para las funciones mencionadas.

```

void init_timer23() {
  TRISCbits.TRISC13 = 1; //T2CK como entrada
  T2CON = 0;
  T2CONbits.T32 = 1; //timer 23 modo 32bits
  T2CONbits.TCS = 1; //fuente de clock externa
  IEC0bits.T3IE = 1; //habilita interrupción por timer
  TMR2 = 0; //Se limpia timer 2
  TMR3 = 0; //se limpia timer 3
  PR2 = 0xFFFF; //se pone al maximo valor de conteo
  PR3 = 0xFFFF;
  T2CONbits.TON = 1; //se habilita el timer 2
}

```

Tabla 16. Código de función `init_timer23`.

Fuente: Elaboración propia.

```

void init_timer1() {
  TRISCbits.TRISC14 = 1; //T1CK como entrada
  T1CON = 0;
  T1CONbits.TCS = 1; //fuente de reloj externa
  T1CONbits.TSYNC = 1; //modo contador sincronizado
  PR1 = 0xFFFF; //interrupción por desborde en la
  máxima cuenta
  TMR1 = 0x0000; //se limpia el valor del timer 1
  IEC0bits.T1IE = 1; //se habilita interrupción del timer 1
  IFS0bits.T1IF = 0; //flag del timer 1 inicialmente en 0
  T1CONbits.TON = 1; //timer 1 activado
}

```

Tabla 17. Código de función `init_timer23`.

Fuente: Elaboración propia.

La función denominada F2M3 para reinicio de los contadores, consiste en poner a 0 el valor de los contadores. Para ello, las variables y registros de Temporizadores del microcontrolador para el conteo de pulsos son igualados a 0 como se muestra en el código de la Tabla 18.

```

count_patron_high = 0;
TMR1 = 0; //el timer 1 se pone a 0
TMR2 = 0; //el timer 2 se pone a 0
TMR3 = 0; //el timer 3 se pone a 0

```

Tabla 18. Código para reinicio de contadores.

Fuente: Elaboración propia.

La función denominada F3M2 para detener el conteo cuando se llega a contar la cantidad programada de pulsos se ejecuta como interrupción por comparación del contador del DUT con el valor programado de pulsos “cuentas\_programadas”. Se vio conveniente que esta función sea una rutina de atención a la interrupción generada cuando se iguala el valor de pulsos del DUT con el de las “cuentas\_programadas”, el código se muestra en la Tabla 19.

```

void __attribute__((__interrupt__, __auto_psv__)) _T3Interrupt(void) {
    __delay32(10); // se genera pequeño retardo
    habilitacion = 0; //Se pone a 0 la señal para habilitación de conteo
    flag_timer3 = 0; //limpiamos el flag para otra interrupción del timer 3
}

```

Tabla 19. Código para reinicio de contadores.

Fuente: Elaboración propia.

La función denominada F4M2 para lectura del valor acumulado se ejecuta con el código mostrado en la Tabla 20.

```

count_ibp = ((unsigned long int) TMR3 << 16) | (unsigned
long int) TMR2;
count_patron = ((unsigned long int) count_patron_high <<
16) | (unsigned long int) TMR1;

```

Tabla 20. Código para lectura de valores de los contadores.

Fuente: Elaboración propia.

### 5.2.2.3. MODULO I/O.

La función denominada F1M3 para configuración del módulo I/O se renombra como “init\_io” para mejor legibilidad. No tiene parámetros de entrada ni salida, su código se muestra en la

```

void init_io() {
    ADPCFG = 0xFFFF; //Define todos los pines
    como entradas
    TRISAbits.TRISA11 = 0; //definir como salida A11
    "señal habilitación"
    TRISDbits.TRISD9 = 1; // se defina el D9 como
    entrada para la señal de finalización "señal
    finalización"
}

```

Tabla 21. Código para función *init\_io*.

Fuente: Elaboración propia.

En cuanto a las funciones de lectura y establecimiento de entrada o salida digital respectivamente, al trabajar con registros, para establecer solo es necesario poner a "1" o "0" la salida correspondiente y en la lectura asignar a alguna variable el valor de la entrada. En la tabla 22 se muestra como realizar el establecimiento y lectura de entradas y salidas digitales.

```

habilitacion = 1; //Establece la señal habilitación en "1"
lógico
estado_conteo=finalizacion; // Lee el valor de la señal
finalización y lo almacena en la variable estado conteo

```

Tabla 22. Código para lectura y establecimiento de pines I/O.

Fuente: Elaboración propia.

#### 5.2.2.4. MODULO COMM.

La función denominada F1M4 para configuración del módulo de comunicación se renombra como "init\_uart". Esta función no tiene parámetros de entrada ni salida, su código se muestra la Tabla 23. Para esta parte de código se considera la siguiente configuración:

- Tasa de Baudios de 115200.
- Datos de 8 bits, sin paridad.
- 1 bit de parada.
- Interrupción en la recepción de caracteres.



```

void init_uart2() {
    U2MODE = 0; //Ponemos a 0 el registro de control del UART2
    U2BRG = BRGVAL; // se carga valor para genera el baudrate
115200 baudios
    U2MODEbits.PDSEL = 0b00; // 8 bits, sin paridad
    U2MODEbits.STSEL = 0; //1 bit de stop
    U2MODEbits.ABAUD = 0; //Auto-Baud, Deshabilitado
    U2STAbits.URXISEL = 0; //interrupción cuando un carácter
es recibido en RX
    IPC6bits.U2RXIP = 1; //PRIORIDAD BAJA DE LA
INTERRUPCION
    IEC1bits.U2RXIE = 1; //habilita interrupción por recepción
UART 2
    IFS1bits.U2RXIF = 0; //se limpia la bandera de la interrupción
de recepción de carácter
    U2MODEbits.UARTEN = 1; //habilita el UART
    U2STAbits.UTXEN = 1; //habilita la transmisión Tx del UART
}

```

Tabla 23. Código de función `init_uart2`.

Fuente: Elaboración propia.

La función denominada F2M4 para recepción de datos se ejecuta como una rutina de atención a la interrupción generada cuando se tienen datos en el buffer de comunicación del microcontrolador. Se sigue el algoritmo descrito en la Tabla 5 y Figura 52. El código se muestra en la Tabla 24.

```

void __attribute__((__interrupt__, no_auto_psv))
_U2RXInterrupt(void) {
    rxbyte = U2RXREG; // se lee el valor del registro
    if (estado1 == 0) {
        switch (rxbyte) {

            case 'i': //comando para iniciar el conteo de pulsos
                count_patron_high = 0;
                TMR1 = 0; //el timer 1 se pone a 0
                TMR2 = 0; //el timer 2 se pone a 0
                TMR3 = 0; //el timer 3 se pone a 0
                habilitacion = 1;
                break;

```

```

    case 'd': //comando para detener un conteo ya
    iniciado
        habilitacion = 0;
        break;

    case 'n': //inicio de registro para conteo de pulsos
    {
        estado1 = 1;
        int z = 0;
        for (z = 0; z <= 9; z++) conteo_str[z] = "0";
        i = 0;
    }
    break;
}
} else if (rxbyte == 'f') { //finalización del registro de cuentas
    estado1 = 0;
    cuentas_programadas = atol(conteo_str);
    PR2=cuentas_programadas&0x0000FFFF;
    PR3 = (cuentas_programadas & 0xFFFF0000) >> 16;
} else {
    char *p; //puntero para almacenar en char
    p = &conteo_str[i];
    *p = rxbyte;
    i++;
}
IFS1bits.U2RXIF = 0; //limpiamos para otra interrupción
}

```

Tabla 24. Código para función de recepción de datos.

Fuente: Elaboración propia.

La función denominada F3M4 para transmisión de datos se renombra como “enviar\_string”. Esta función recibe como parámetro de entrada la trama de datos y no tiene parámetros de salida, su código se muestra en la tabla 25.



```

void enviar_string(char *str) {//función para enviar string
    char *p; // define variable tipo puntero
    p = str; // apuntamos a la direccion de la cadena de
caracteres
    while (*p) enviar_char(*p++); //se envía cada uno de
los caracteres del string
}

```

Tabla 25. Código para función enviar\_string.

Fuente: Elaboración propia.

Como se muestra en la tabla 25 la función “enviar\_string” requiere de otra función denominada “enviar\_char” cuyo código se muestra en la Tabla 26.

```

void enviar_char(char ch) {
    while (U2STAbits.UTXBF == 1); //espera a
que buffer de transmisión este vacío
    U2TXREG = ch; //carga el carácter en el
buffer de transmisión
}

```

Tabla 26. Código para función enviar\_char.

Fuente: Elaboración propia.

### 5.2.3. ORDENAMIENTO DEL CÓDIGO.

El código completo para el microcontrolador se ordena de acuerdo a la Figura 51 mostrado en el punto 4.2.2 del capítulo de Desarrollo del Sistema y se siguen los flujogramas mostrados en la Figura 52, Figura 53 y Figura 54. El código completo se muestra en el Anexo F.

### 5.3. PROGRAMACIÓN DE LA GUI.

La interfaz de usuario tiene los controles, visualizadores y gráficos necesarios en el monitoreo y toma de datos para una calibración. El programa se comunica con el microcontrolador a través de comandos y caracteres. La finalidad es que el operador pueda programar la toma de mediciones, es decir, el conteo de pulsos, monitorear las variables de influencia en la línea y almacenar los datos adquiridos. LabVIEW es un entorno de programación para sistemas de adquisición e instrumentación, por tal razón se elige esta plataforma para la programación de la GUI.

La implementación de la GUI se realiza con base en las descripciones del punto 4.2.3 del capítulo de Desarrollo del sistema. Inicialmente se programa los elementos visuales y posteriormente la lógica del programa.

### 5.3.1. ELEMENTOS VISUALES.

Los elementos visuales como indicadores, botones y gráficos se distribuyen según lo establecido en la Figura 58, tomando en cuenta los elementos descritos en la Tabla 8 la interfaz gráfica toma el aspecto que se muestra en la Figura 85.

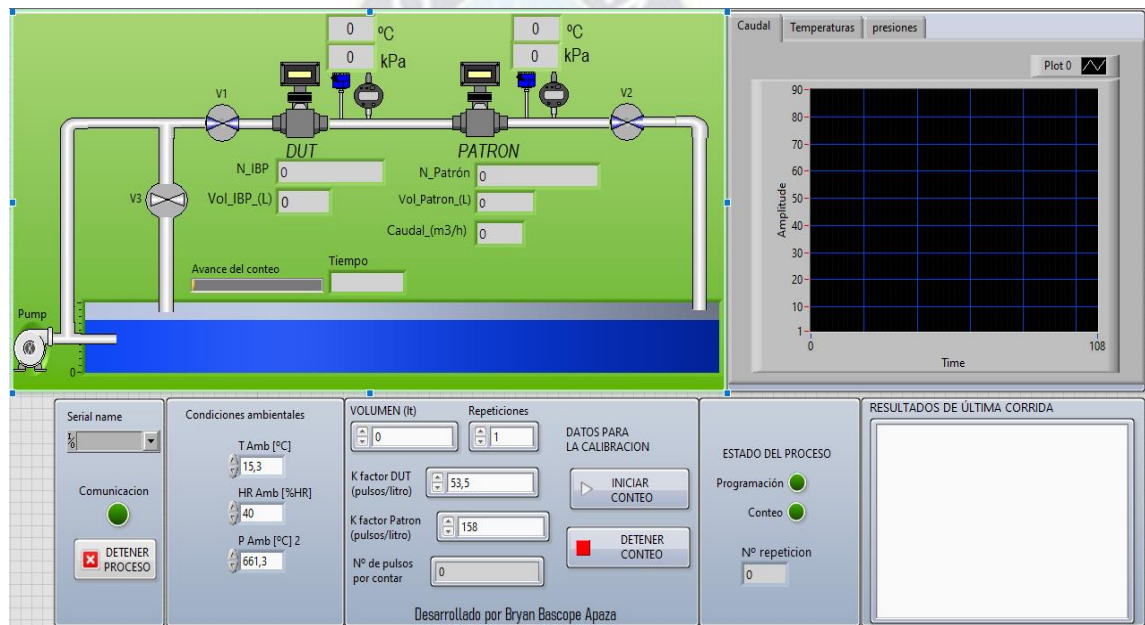
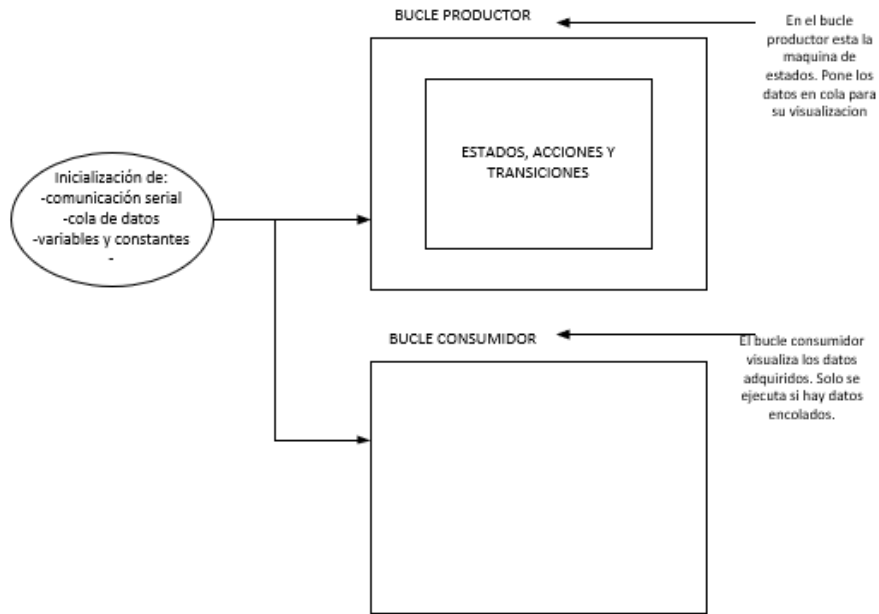


Figura 85. Interfaz de usuario.

Fuente: Elaboración propia.

### 5.3.2. PROGRAMACIÓN DEL DIAGRAMA DE BLOQUES.

Como se explicó en el punto 4.2.3 la GUI cuenta con un elemento de lógica que corresponde al diagrama de bloques en LABVIEW. El mismo debe ser capaz de responder a los comandos, generar reportes en archivos de texto, almacenar y visualizar los datos. Por esta razón, se optó por usar la arquitectura productor consumidor. La estructura para el diagrama de bloques, se basa en la arquitectura productor consumidor de la Figura 59 y se muestra en la Figura 86.



*Figura 86. Esquema de bloques en el programa de LabVIEW.  
Fuente: Elaboración propia.*

El bucle productor responde a los elementos de la interfaz, almacena datos y genera reportes. El bucle consumidor se encarga de realizar la visualización de los datos obtenidos. A continuación, se muestra los diagramas de bloque, para el productor y consumidor, que se desarrollaron en LabVIEW.

### **5.3.2.1. BUCLE PRODUCTOR.**

Dentro del bucle productor se programa la máquina de estados mostrada en la Figura 60. La máquina de estados responderá a los eventos de la interfaz con acciones específicas y permitirá una mejor organización de los bloques.

A continuación, se muestra el diagrama de bloques de cada estado definido en el punto 4.2.3 y se toma en cuenta las transiciones que se muestran en la Figura 60.

**Crear fichero.** Es el primer estado después del inicio de la máquina de estados. Las acciones y condiciones de transición son:

- Mostrar un cuadro de dialogo para introducir datos de la calibración. En base a los datos introducidos se define el nombre y crea el archivo de texto donde se guardan los datos adquiridos.

- Imprimir la cabecera en la primera fila del archivo de texto creado.
- Limpiar el buffer de comunicación serial.
- Realizar la transición al estado “programación” sin condición.

En la Figura 87 se muestra el diagrama de bloques del estado “crear fichero”.

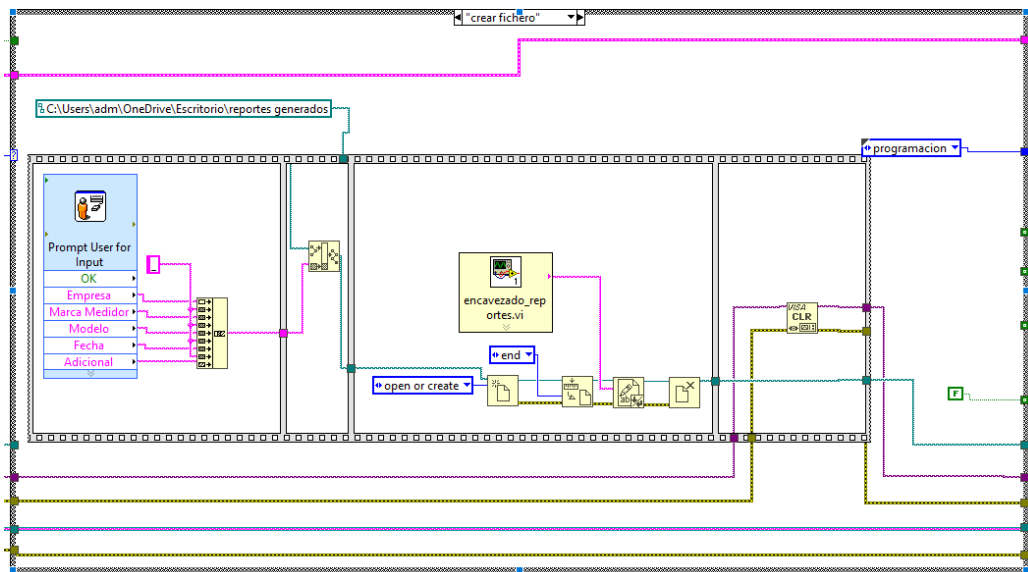


Figura 87. Diagrama de bloques del estado Crear fichero.

Fuente: Elaboración propia.

**Programación.** Se espera que el operador introduzca los datos necesarios para la calibración. Las acciones y condiciones de transición para este estado son:

- Poner a 0 el valor de repeticiones realizadas.
- Verificar si existen datos en el puerto serial, de ser así, separa los datos adquiridos. Actualiza el valor de conteo de pulsos y pone a la cola la cadena de caracteres recibido.
- Establece ren 1 el led indicador “programación” y en 0 el led indicador de conteo.
- Si se oprime el botón “DETENER PROCESO” pasa al estado “detener proceso”.
- Si se oprime el botón “INICIAR CONTEO” pasa al estado “grabación en uC”.

En la Figura 88 se muestra el diagrama de bloques del estado “programación”.

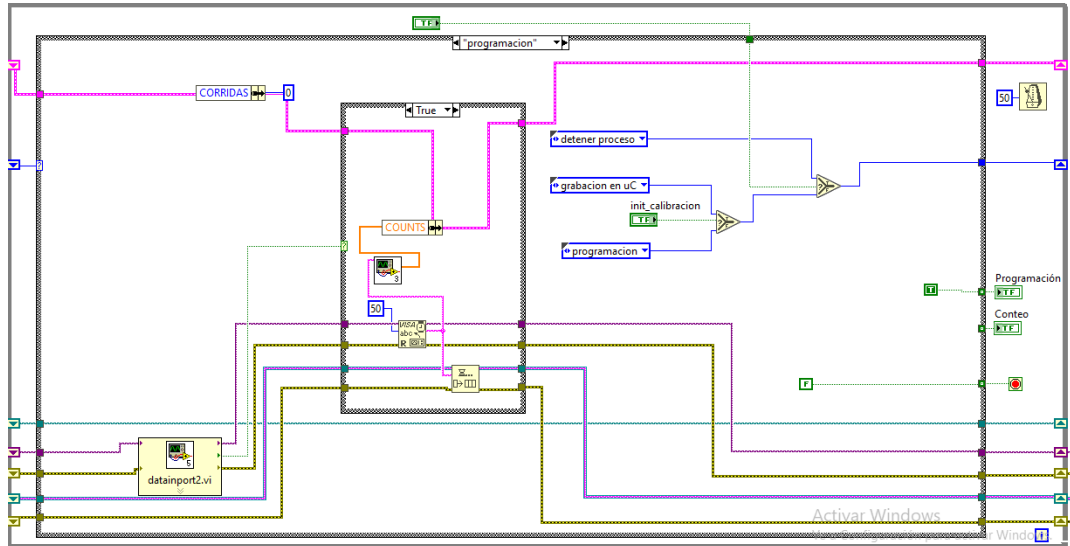


Figura 88. Diagrama de bloques del estado "programación".  
 Fuente: Elaboración propia.

**Grabación en uC.** Envía los comandos necesarios para el registro e inicio de conteo en el microcontrolador. También inicializa las variables necesarias antes del conteo. Las acciones y transiciones son:

- Inicializar el array donde se almacenan los datos de los transmisores en [0 0 0 0 0].
- Inicializar el contador de tiempo.
- Enviar el carácter 'r', la cantidad de pulsos a contar, el carácter 'f' e 'i', a través del puerto serial, al microcontrolador.
- Realizar la transición al estado "conteo" sin condición.

En la Figura 88 se muestra el diagrama de bloques del estado "grabación en uC".

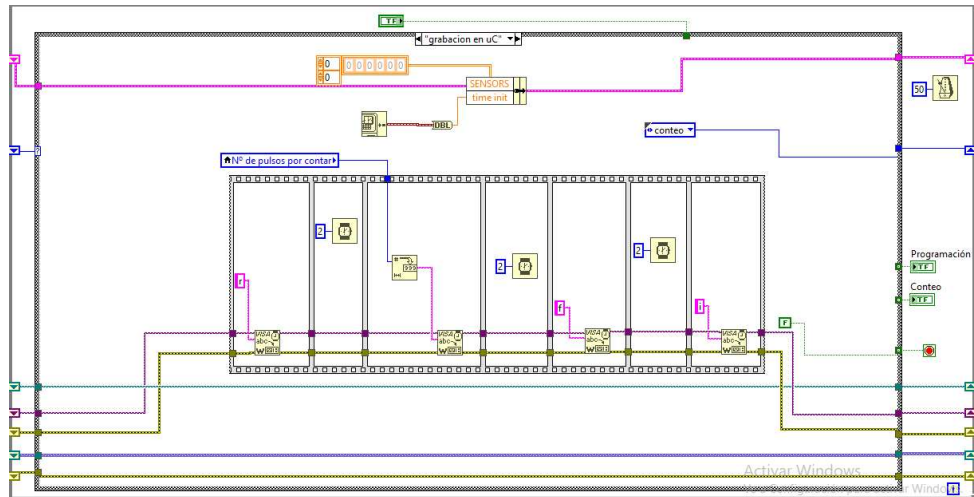


Figura 89. Diagrama de bloques del estado "grabación en uC".

Fuente: Elaboración propia.

**Conteo.** Estado para el conteo de pulsos y almacenamiento de los valores de los transmisores. Las acciones y transiciones son:

- Verificar si se tiene datos en el puerto serial. De ser así, recibe el valor del puerto serial y lo pone en la cola de datos
- Convertir los valores de lectura analógica recibidos en sus valores de temperatura y presión y almacenarlos en arrays.
- Actualizar el valor de los conteos de pulsos.
- Contar el tiempo de conteo y mostrarlo en el indicador "tiempo".
- Establecer en 1 el led indicador "conteo" y en 0 el led indicador "programación".
- Si se oprime el botón "DETENER PROCESO" se realiza transición al estado "detener proceso".
- Si se oprime el botón "DETENER CONTEO" envía el comando 'd' a través de puerto serie y se realiza transición al estado "programación".
- Si se recibe la indicación de finalización de conteo se realiza transición al estado "guarda dato".

En la Figura 90 se muestra el diagrama de bloques del estado "conteo".



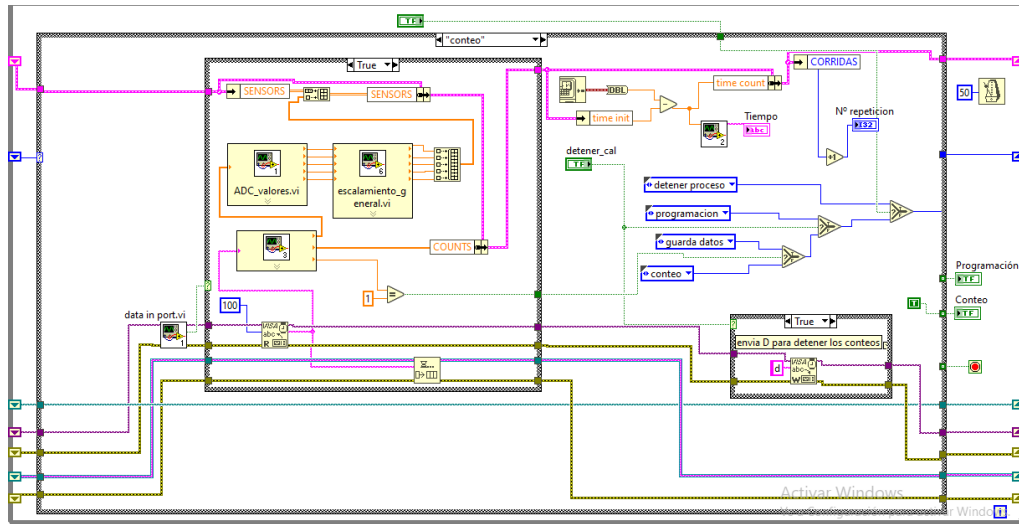


Figura 90. Diagrama de bloques del estado "grabación en uC".  
Fuente: Elaboración propia.

**Guarda datos.** Añade filas de datos al archivo de texto con los datos obtenidos en el conteo. Las acciones y transiciones son:

- Obtener los valores promedios de los sensores durante el conteo.
- Mostrar y almacenar los datos adquiridos. En el caso de los transmisores los valores promedios. En el caso de los pulsos los últimos valores del conteo.
- Si se realizaron todas las repeticiones definidas en el control numérico "Repeticiones" se realiza la transición al estado "programación", caso contrario la transición será al estado "grabación en uC" para empezar un nuevo conteo.

En la Figura 91 se muestra el diagrama de bloques del estado "guarda dato".



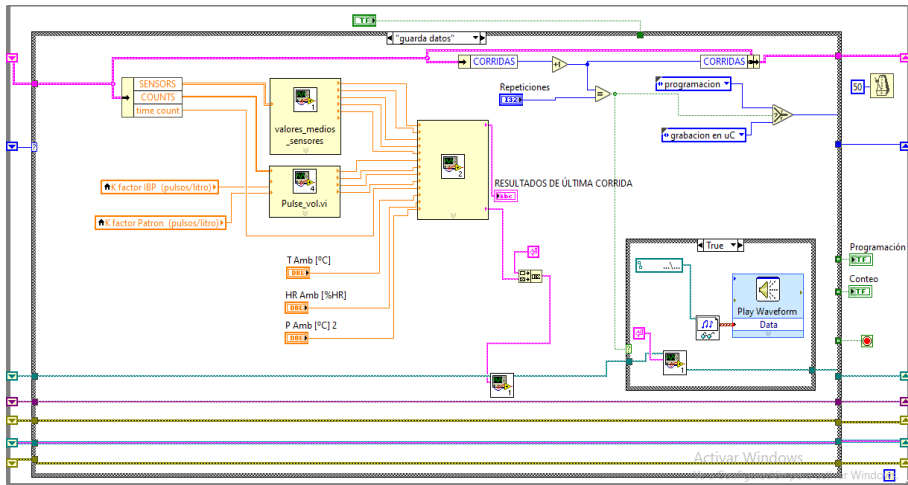


Figura 91. Diagrama de bloques del estado "guarda dato".  
Fuente: Elaboración propia.

**Detener proceso.** Interrumpe la ejecución del programa. Las acciones y transiciones son:

- Enviar el comando 'd' al microcontrolador.
- Finalizar la ejecución del bucle productor.

En la Figura 91 se muestra el diagrama de bloques del estado "detener proceso".

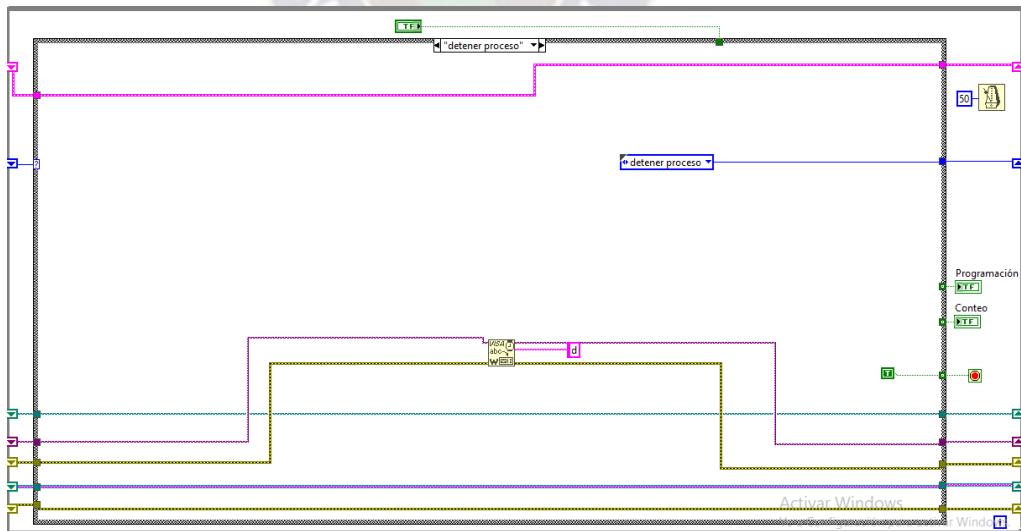


Figura 92. Diagrama de bloques del estado "detener proceso".  
Fuente: Elaboración propia.

### 5.3.2.2. BUCLE CONSUMIDOR.

Realiza la visualización de los datos obtenidos. Se ejecuta cada vez que se tienen datos en la cola. Se recibe una cadena de caracteres en el formato de la trama definido en el punto 4.2.2 y se convierte a sus valores de temperatura, presión, caudal y conteo de pulsos. en la Figura 93 se muestra el diagrama de bloques.

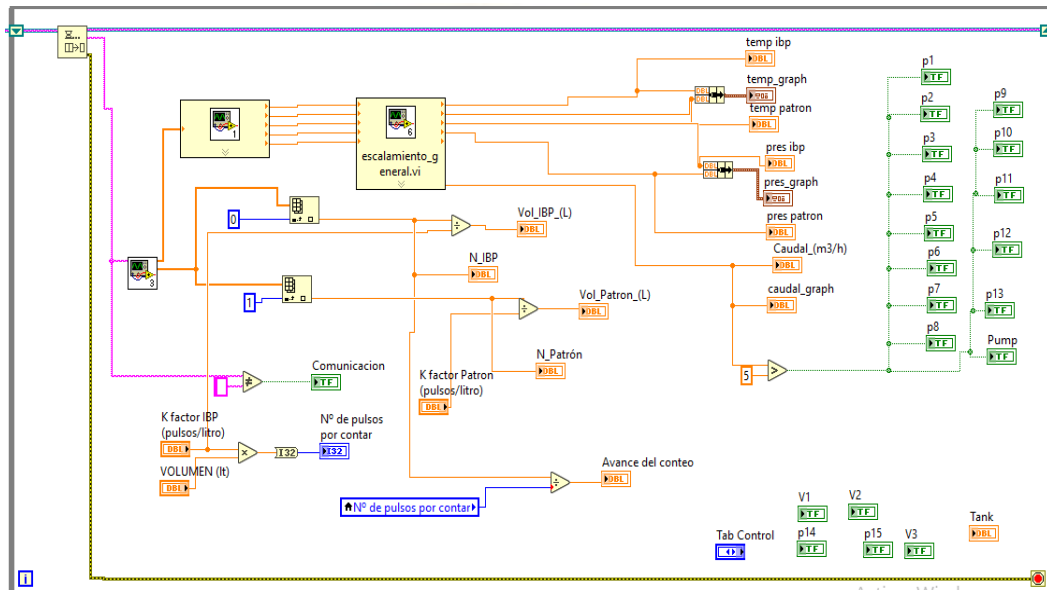


Figura 93. Diagrama de bloques de bucle consumidor.

Fuente: Elaboración propia.

### 5.3.2.3. PROGRAMA.

Con base en el esquema de la Figura 86 se unen los bucles productor y consumidor, la distribución y conexiones resultantes se muestran en la Figura 94. Los subVI's utilizados en la programación se puede ver en el anexo G.

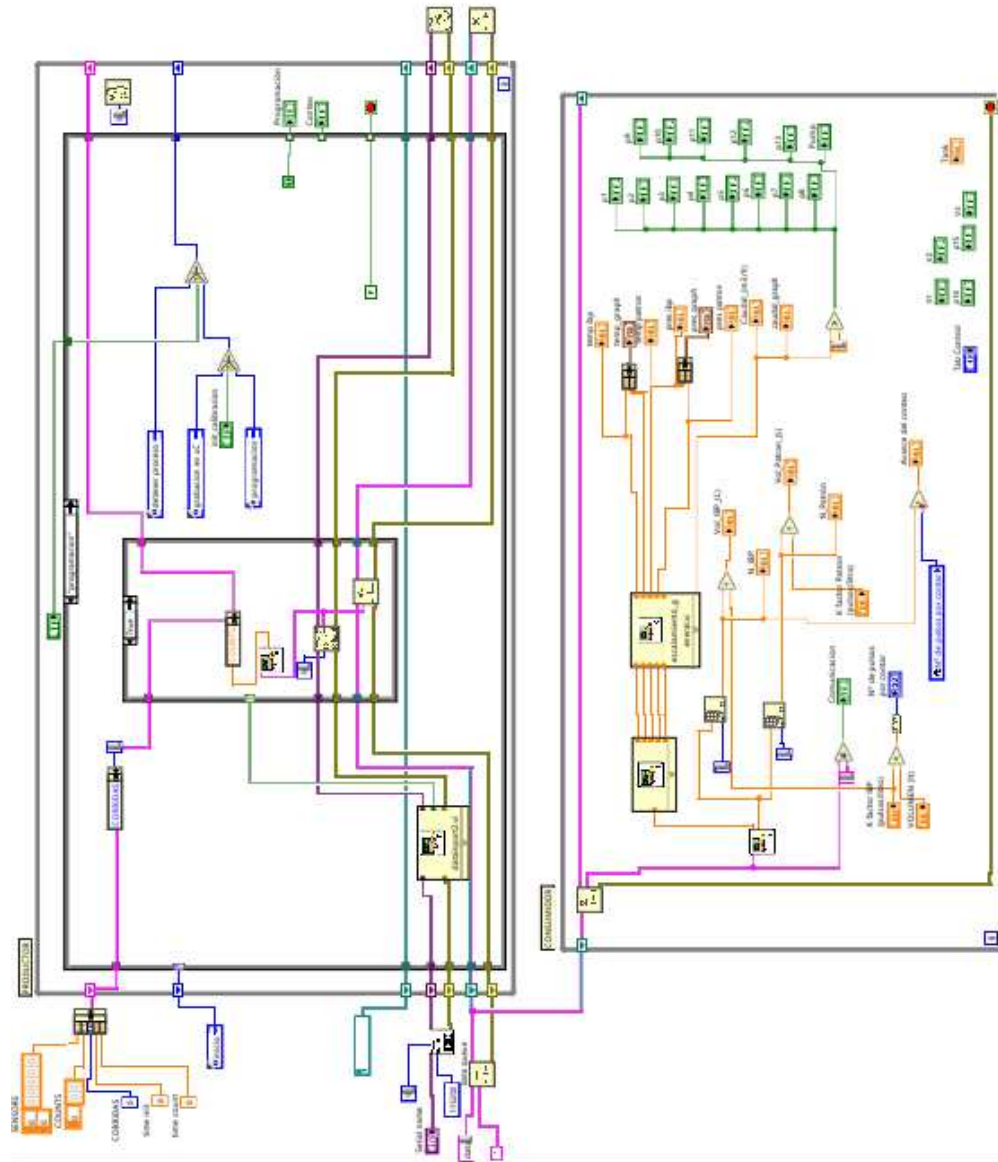


Figura 94. diagrama de bloques del programa en LabVIEW.  
Fuente: Elaboración propia.

#### **5.4. INTEGRACIÓN.**

Se realiza la integración de Hardware y la GUI. Para ello inicialmente se construyeron tarjetas de circuito impreso (PCB) con base en los circuitos mostrados en el punto 5.1 y teniendo en cuenta se usa un microcontrolador DSPIC30F4013. Posteriormente, las tarjetas se conectan al computador donde se ejecuta la GUI desarrollada en LAbVIEW.

Para el Hardware se construyeron 3 placas de circuito impreso (PCB) la cuales contienen los circuitos que se desarrollaron para los módulos M1 y M2 y M3 el Hardware. Las denominaciones de las tarjetas y los circuitos (módulos o submódulos) que contienen se detallan a continuación:

- Tarjeta 1: Contiene los circuitos del módulo M1 de forma íntegra para los 5 canales analógicos que se identificaron como necesarios. Entonces el circuito usado para esta tarjeta es el de la Figura 73. En el anexo H.1 se muestra el esquemático, layout y la PCB resultante con los componentes soldados.
- Tarjeta 2: Contiene los circuitos de los submódulos M21 y M22 necesarios para la conexión de las señales de pulsos de los medidores. Entonces el circuito usado para esta tarjeta es el de la Figura 76 y Figura 78. En el anexo H.2 se muestra el esquemático, layout y la PCB resultante con los componentes soldados.
- Tarjeta 3: Contiene los circuitos de los submódulos M23, M24 para la sincronización de los pulsos, y el módulo M3 para el funcionamiento del microcontrolador. Entonces el circuito usado para esta tarjeta es el de la Figura 79 y Figura 81 juntamente con el circuito de funcionamiento del DSPIC30F4013. En el anexo H.3 se muestra el esquemático, layout y la PCB resultante con los componentes soldados.

En la Figura 95 se muestra la interconexión de las tarjetas y mencionadas.

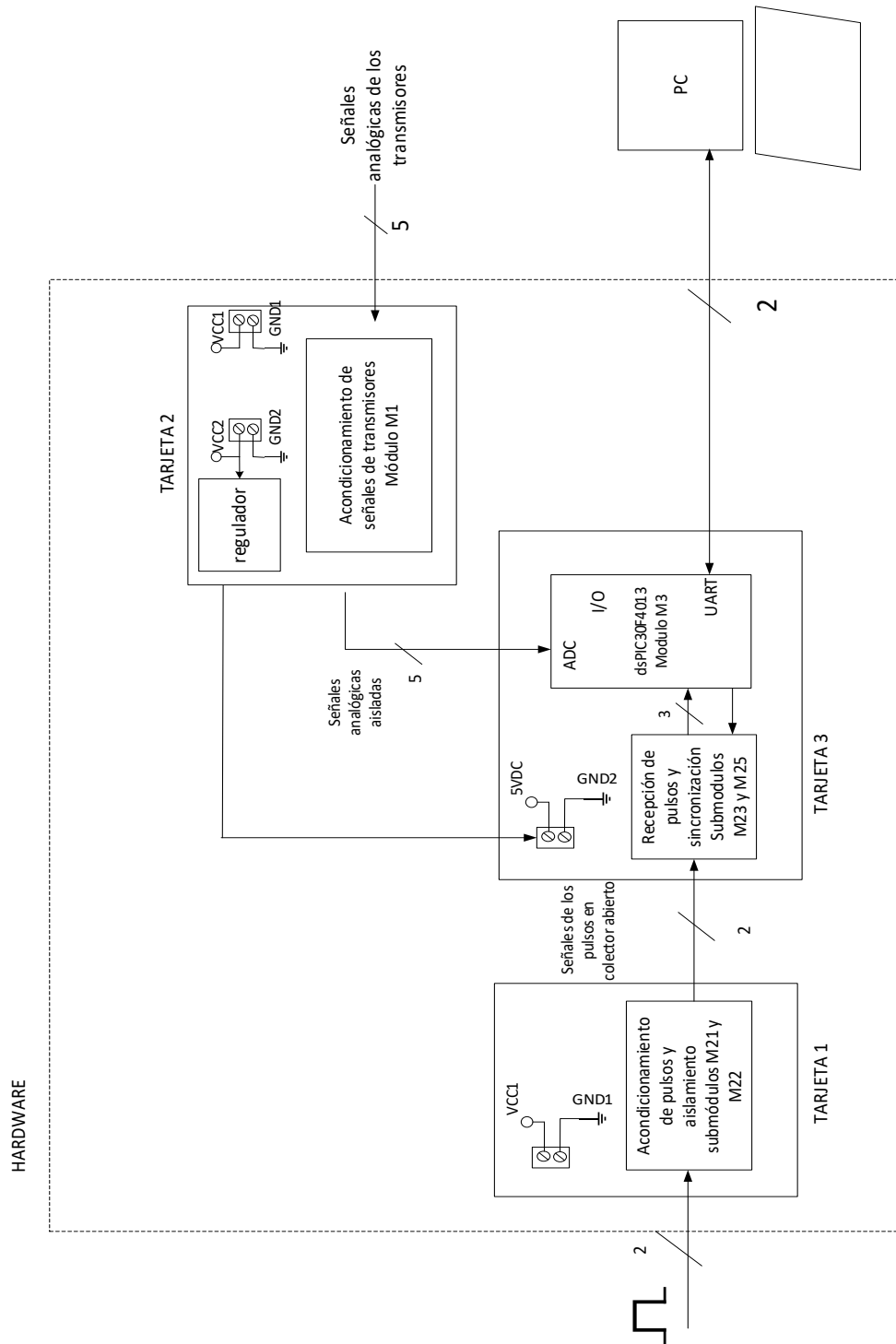


Figura 95. Esquema de tarjetas y su conexión.  
Fuente: Elaboración propia.

Para conectar el Hardware a la computadora donde se ejecuta la GUI programada en LabVIEW se utiliza un cable de adaptación para de puerto USB a Serial, cuyo modelo es PL2303 que se muestra en la Figura 96.



*Figura 96. Cable PL2303 para conexión de Hardware con la GUI.  
Fuente: Elaboración propia.*

## 6. Capítulo 6 – Validación del diseño.

En el presente capítulo se muestra las pruebas y mediciones realizadas en la adquisición de datos para la calibración de un medidor de flujo líquido con el prototipo del sistema. Se empieza realizando pruebas de funcionamiento fuera y dentro de laboratorio y se verifica los recursos que demandan la toma de medidas. Finalmente, se realiza el análisis de consistencia y repetibilidad de los datos obtenidos para verificar que **las mediciones** presentan confiabilidad

### 6.1. FUNCIONAMIENTO DEL SISTEMA.

#### Pruebas fuera de laboratorio.

Inicialmente, se realiza pruebas emulando el comportamiento de los transmisores con potenciómetros y los pulsos de los medidores con generadores basados en circuito integrado 7414. En el software primero se elige el puerto de comunicación COM. Después, al correr al programa, primero se pide los datos para el nombre del archivo de texto donde se registran los datos adquiridos como se muestra en la Figura 97.

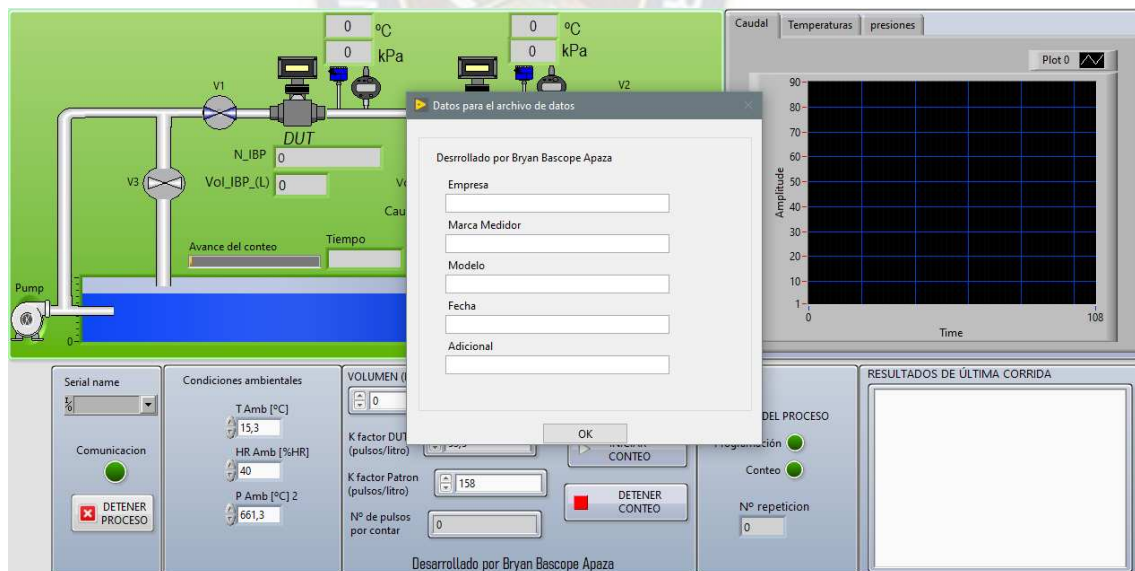


Figura 97. Inicio del programa.

Fuente: Elaboración propia.

Se verifica la visualización de los datos de transmisores emulados con potenciómetros y el conteo de pulsos. Se realizan conteos configurando el volumen y numero de repeticiones como se muestra en la Figura 98.



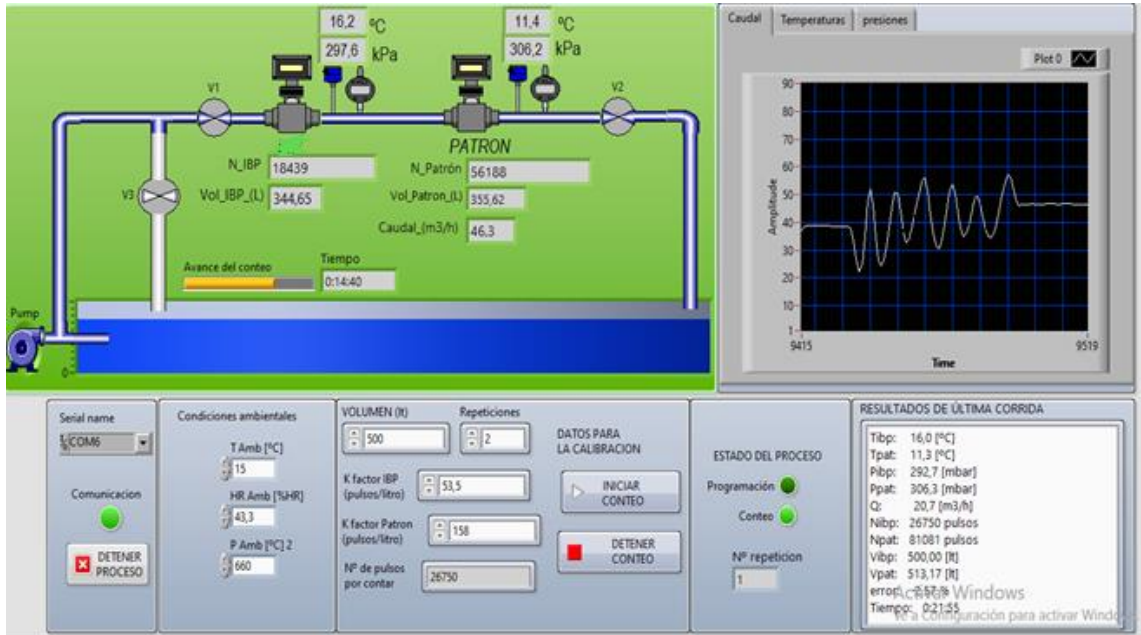


Figura 98. Visualización y conteo en pruebas fuera de laboratorio.  
Fuente: Elaboración propia.

En la Figura 99 se muestra el archivo de texto con los datos adquiridos en forma tabulada, en consecuencia, se obtiene los datos esperados.

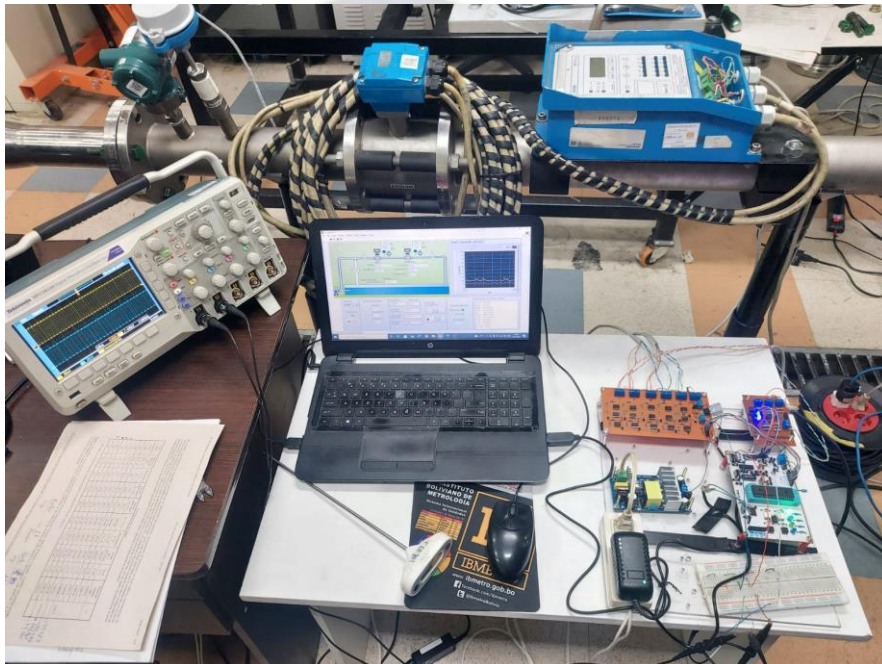
Bryan\_Yokogawa\_YG10\_20210130\_Prueba0.txt: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda						
Q	T amb	HR amb	P amb	N_dut	T_dut	P_dut	N_p	T_p	P_p	
[m <sup>3</sup> /h]	[°C]	[%HR]	[mbar]	pulse	[°C]	[kPa]	pulse	[°C]	[kPa]	
43,8	15,0	43,3	660,0	26750	16,2	296,5	81533	11,4	308,2	
46,5	15,0	43,3	660,0	26750	16,2	296,1	81558	11,4	307,6	
46,4	15,0	43,3	660,0	26750	16,1	295,0	82076	11,3	306,7	
46,3	15,0	43,3	660,0	53500	16,1	294,8	163610	11,3	305,9	
46,2	15,0	43,3	660,0	53500	16,1	293,8	163369	11,3	305,5	

Figura 99. Datos adquiridos en pruebas fuera de laboratorio.  
Fuente: Elaboración propia.

### **Pruebas en laboratorio.**

Se realizan pruebas en el laboratorio de Flujo de IBMETRO. Se conectan los transmisores y medidores a las tarjetas desarrolladas. En la Figura 100 se muestran la tarjeta con las conexiones funcionando en el laboratorio.



*Figura 100. Sistema funcionando en laboratorio.*

*Fuente: Elaboración propia.*

Inicialmente se verificaron los valores obtenidos de los transmisores. Para la temperatura se usa un termómetro digital, para presión la pantalla de los transmisores y para caudal el visualizador del medidor patrón. En la se muestra los datos visualizados en la interfaz y las mediciones con otros instrumentos.

TEMPERATURA			
Instrumento		En interfaz de LabVIEW	
Temperatura termómetro [°C]		Temperatura transmisor Tdut [°C]	Temperatura transmisor Tp [°C]
15,5		15,7	15,2
PRESIÓN			
Instrumento		En interfaz de LabVIEW	
Pantalla de transmisor Pdut [kPa]	Pantalla transmisor Pp [kPa]	Presión de transmisor Pdut [kPa]	Presión de transmisor Pp [kPa]
20,6	13,1	20,2	13,6
CAUDAL			
Instrumento		En interfaz de LabVIEW	
pantalla medidor patrón Q [m3/h]		Caudal desde el medidor patrón [m3/h]	
33,5		32,2	

Tabla 27. Valores indicados en la interfaz e instrumentos en la línea.  
Fuente: Elaboración propia.

Se verifica que los pulsos obtenidos sean estables y con los niveles deseados 0 V – 5 V, para esto se utiliza el osciloscopio de laboratorio. En la Figura 101 se puede ver la señal de pulsos del DUT (amarillo) y del Patrón (azul).

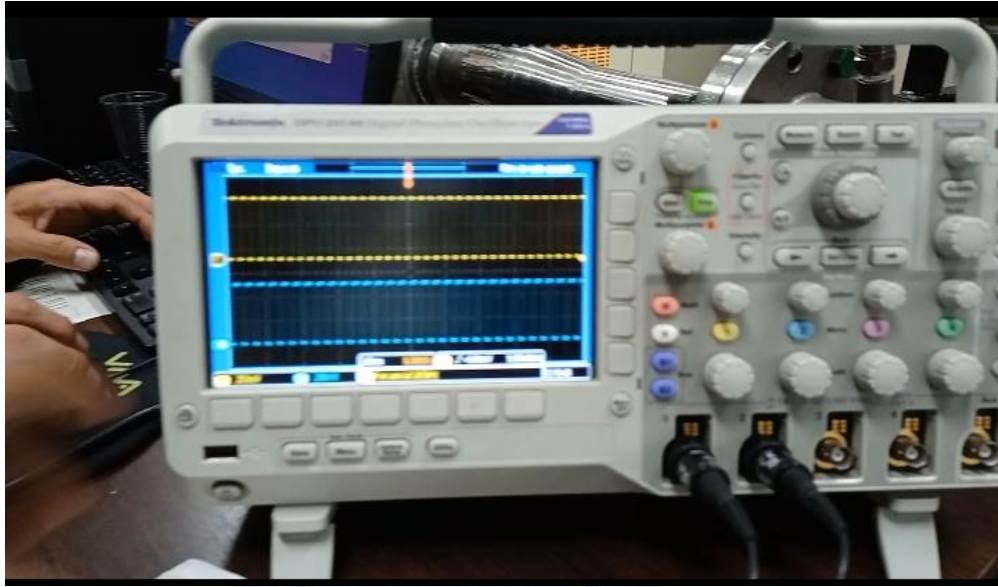


Figura 101. Pulsos del DUT y patrón.

Fuente: Elaboración propia.

Para analizar el funcionamiento se realizaron conteos con 3 repeticiones para 3 volúmenes (300 L, 500 L y 800 L) y caudal de  $33 \text{ m}^3/\text{h}$  aproximadamente. El medidor usado como DUT en las pruebas es de marca HENDRESS HAUSER modelo PROMASS 83E. En la Figura 102 se muestra el archivo de texto con los datos. Se debe mencionar que para la toma de mediciones solo es necesario un operador el cual puede monitorear las magnitudes de influencia en la medición sin tener que realizar operaciones de apertura y cierre de válvulas o apunte de valores de los transmisores.

Archivo	Edición	Formato	Ver	Ayuda						
Q	T amb	HR amb	P amb	N_dut	T_dut	P_dut	N_p	T_p	P_p	
[m <sup>3</sup> /h]	[°C]	[%HR]	[mbar]	pulse	[°C]	[kPa]	pulse	[°C]	[kPa]	
33,4	15,3	40,0	661,3	59989	15,7	20,4	49891	15,3	13,1	
33,4	15,3	40,0	661,3	59989	15,7	20,7	49886	15,3	12,9	
33,4	15,3	40,0	661,3	59989	15,7	20,7	49887	15,3	13,2	
33,1	15,3	40,0	661,3	99981	15,7	20,7	83162	15,3	12,9	
33,0	15,3	40,0	661,3	99981	15,7	20,6	83130	15,2	12,7	
33,2	15,3	40,0	661,3	99981	15,6	20,6	83123	15,2	12,9	
33,1	15,3	40,0	661,3	159970	15,6	20,6	133039	15,2	12,8	
33,2	15,3	40,0	661,3	159970	15,6	20,8	133030	15,2	12,7	
33,2	15,3	40,0	661,3	159970	15,6	20,5	133034	15,2	12,9	

Figura 102. Datos obtenidos en las pruebas de laboratorio.

Fuente: Elaboración propia.

## 6.2. ANALISIS DE CONFIABILIDAD DE LAS MEDICIONES.

Como se mencionó en el punto 3.4 del marco referencial, la confiabilidad en las mediciones está asociada al análisis cuantitativo de consistencia y repetibilidad. Para ello, se tomaron medidas a tres (3) cantidades de volumen diferente (300 L, 500L y 800 L) a un caudal de  $33 \text{ m}^3/\text{h}$  por tres (3) diferentes operadores. Los datos obtenidos con el prototipo del sistema, se muestran en el anexo H.

Como se explicó, la cantidad de pulsos contados para el DUT es constante por lo cual su variación es nula. En el caso del patrón, las mediciones obtenidas varían por lo cual se usarán las mismas para el análisis de consistencia y repetibilidad, tomando en cuenta 10 repeticiones. Se toma en cuenta que la constante de pulsos del medidor patrón (KHRONE IFC 110) es  $166 \frac{\text{pulso}}{\text{L}}$ . Los pulsos y volumen obtenidos por cada operador para cada cantidad de volumen se muestran en la Tabla 28, Tabla 29 y Tabla 30.



Volumen medido=300 L					
Operador A		Operador B		Operador C	
N patrón	V patrón (L)	N patrón	V patrón (L)	N patrón	V patrón (L)
49887	300,52	49891	300,55	49893	300,56
49888	300,53	49887	300,52	49885	300,51
49887	300,52	49887	300,52	49889	300,54
49892	300,55	49888	300,53	49886	300,52
49883	300,50	49886	300,52	49885	300,51
49890	300,54	49885	300,51	49893	300,56
49884	300,51	49889	300,54	49884	300,51
49888	300,53	49886	300,52	49885	300,51
49886	300,52	49885	300,51	49887	300,52
49889	300,54	49886	300,52	49890	300,54

Tabla 28. Pulsos y Volumen obtenidos en el patrón para medición de 300 L.  
Fuente: Elaboración propia.

Volumen medido=500 L					
Operador A		Operador B		Operador C	
N patrón	V patrón (L)	N patrón	V patrón (L)	N patrón	V patrón (L)
83097	500,58	83159	500,96	83092	500,55
83153	500,92	83134	500,81	83152	500,92
83115	500,69	83112	500,67	83187	501,13
83134	500,81	83125	500,75	83156	500,94
83114	500,69	83108	500,65	83167	501,01
83133	500,80	83137	500,83	83131	500,79
83152	500,92	83156	500,94	83129	500,78
83122	500,73	83142	500,86	83147	500,89
83129	500,78	83140	500,84	83131	500,79
83111	500,67	83124	500,75	83158	500,95

Tabla 29. Pulsos y Volumen obtenidos en el patrón para medición de 500 L.  
Fuente: Elaboración propia.

Volumen medido=800 L					
Operador A		Operador B		Operador C	
N patrón	V patrón (L)	N patrón	V patrón (L)	N patrón	V patrón (L)
133041	801,45	133032	801,40	133026	801,36
133037	801,43	133032	801,40	133030	801,39
133031	801,39	133034	801,41	133030	801,39
133032	801,40	133035	801,42	133035	801,42
133041	801,45	133035	801,42	133030	801,39
133032	801,40	133034	801,41	133040	801,45
133032	801,40	133029	801,38	133034	801,41
133034	801,41	133036	801,42	133038	801,43
133029	801,38	133041	801,45	133032	801,40
133029	801,38	133044	801,47	133036	801,42

*Tabla 30. Pulsos y Volumen obtenidos en el patrón para medición de 800 L.  
Fuente: Elaboración propia.*

#### **6.2.1. CONSISTENCIA DE LAS MEDICIONES.**

Se analiza la consistencia de las mediciones, de forma cuantitativa, con base en los fundamentos mostrados en el punto 3.4.1 del marco referencial. En la Tabla 31, Tabla 32 y Tabla 33 se muestra los valores de consistencia obtenidos en base a las mediciones de volumen del patrón.



PUNTO 300 L	Promedio $\bar{v}$	Desviación estándar $s$	Grados de libertad $v = n - 1$	$v * S^2$	Consistencia intra ensayos		Valores atípicos de dispersión	
					$k$	$k < k_{\text{critico}}$	$C = \frac{S_i^2}{\sum_{i=1}^N S_i^2}$	$C < C_{\text{critico}}$
Operador A	300,5299	0,0160	9	2,315E-03	0,97582	Datos consistentes	0,317	Datos aceptados
Operador B	300,5272	0,0124	9	1,387E-03	0,75541	Datos consistentes	0,190	Datos aceptados
Operador C	300,5326	0,0200	9	3,591E-03	1,21537	Datos consistentes	0,492	Datos aceptados
	N	n	$\sum_{i=1}^n S_i^2$	$\sum_{i=1}^n (n-1)$	$k_{\text{critico}}$ (95%)	Critico (95%)		
	3	10	0,00081	27	1,29	0,871		

Tabla 31. Análisis de consistencia en mediciones a 300 L.  
Fuente: Elaboración propia.

PUNTO 300 L	Promedio $\bar{v}$	Desviación estándar $S$	Grados de libertad $v = n - 1$	$v * S^2$	Consistencia intra ensayos		Valores atípicos de dispersión	
					$k$	$k < k_{\text{critico}}$	$C = \frac{S_i^2}{\sum_{i=1}^N S_i^2}$	$C < C_{\text{critico}}$
Operador A	500,7621	0,1073	9	1,036E-01	0,86599	Datos consistentes	0,250	Datos aceptados
Operador B	500,8087	0,1018	9	9,334E-02	0,82194	Datos consistentes	0,225	Datos aceptados
Operador C	500,8766	0,1555	9	2,175E-01	1,25478	Datos consistentes	0,525	Datos aceptados
	N	n	$\sum_{i=1}^N S_i^2$	$\sum_{i=1}^N (n-1)$	$k_{\text{critico}}$ (95%)	Critico (95%)		
	3	10	0,04605	27	1,29	0,871		

Tabla 32. Análisis de consistencia en mediciones a 500 L.  
Fuente: Elaboración propia

PUNTO 300 L	Promedio $\bar{v}$ [L]	Desviación estándar $S$	Grados de libertad $\nu = n - 1$	$\nu * S^2$	Consistencia intra ensayos		Valores atípicos de dispersión	
					$k$	$k < k_{\text{critico}}$	$C = \frac{S_i^2}{\sum_{i=1}^N S_i^2}$	$C < C_{\text{critico}}$
Operador A	801,4110	0,0267	9	6,399E-03	1,01418	Datos consistentes	0,343	Datos aceptados
Operador B	801,4196	0,0267	9	6,406E-03	1,01473	Datos consistentes	0,343	Datos aceptados
Operador C	801,4071	0,0255	9	5,859E-03	0,97044	Datos consistentes	0,314	Datos aceptados
	N	n	$\sum_{i=1}^N S_i^2$	$\sum_{i=1}^N (\nu - 1)$	$k_{\text{critico}}$ (95%)	Critico (95%)		
	3	10	0,00207	27	1,29	0,871		

Tabla 33. Análisis de consistencia en mediciones a 800 L.  
Fuente: Elaboración propia.

Como se observa en las tablas anteriores todas las mediciones presentan consistencia y son válidas según los criterios establecidos.

### 6.2.1. REPETIBILIDAD DE LAS MEDICIONES.

Se analiza la repetibilidad de las mediciones con base en los fundamentos explicados en el punto 2.1.1 del marco de referencia. Se debe tener en cuenta que para el criterio de repetibilidad se considera las especificaciones del fabricante del medidor patrón (KHRONE IFC110) que en su manual indica, la repetibilidad es  $0.1 \% * Valor\ de\ la\ medicion$ . Por ejemplo, si se mide 300 L con el medidor patrón entonces la repetibilidad deberá ser siempre menor a  $0,1 \% * 300 = 0,3$  y se establece este valor como límite para la repetibilidad. En la Tabla 34, Tabla 35 y Tabla 36 se resume el análisis de repetibilidad para los volúmenes medidos con el medidor patrón.

Volumen medido: 300 L					
Operador	Volumen medido $\bar{V}$	Desviación estándar $S$	Repetibilidad $r$	Límite repetibilidad $L_r$	Criterio
A	300,53	0,0160	0,0505	0,3	Repetibilidad Valida
B	300,53	0,0124	0,0391	0,3	Repetibilidad Valida
C	300,53	0,0200	0,0629	0,3	Repetibilidad Valida

Tabla 34. Análisis de repetibilidad en las mediciones a 300 L.  
Fuente: Elaboración propia.

Volumen medido: 500 L					
Operador	Volumen medido $\bar{V}$	Desviación estándar $S$	Repetibilidad $r$	Limite repetibilidad $L_r$	Criterio
A	500,76	0,1073	0,3381	0,5	Repetibilidad Valida
B	500,81	0,1018	0,3209	0,5	Repetibilidad Valida
C	500,88	0,1555	0,4899	0,5	Repetibilidad Valida

Tabla 35. Análisis de repetibilidad en las mediciones a 500 L.  
Fuente: Elaboración propia.

Volumen medido: 800 L					
Operador	Volumen medido $\bar{V}$	Desviación estándar $S$	Repetibilidad $r$	Limite repetibilidad $L_r$	Criterio
A	801,41	0,0267	0,0840	0,8	Repetibilidad Valida
B	801,42	0,0267	0,0841	0,8	Repetibilidad Valida
C	801,41	0,0255	0,0804	0,8	Repetibilidad Valida

Tabla 36. Análisis de repetibilidad en las mediciones a 800 L.  
Fuente: Elaboración propia.

Como se ve en las tablas anteriores, en cada punto de calibración y para cada operador la repetibilidad es válida con el prototipo de sistema.

# **7. Capítulo 7 – Conclusiones y Recomendaciones.**

## **7.1. CONCLUSIONES.**

Con base en la experiencia lograda durante desarrollo del proyecto, se presenta cada objetivo específico acompañado de su respectiva conclusión.

- Investigar el procedimiento y datos requeridos para la calibración de medidores de flujo líquido, utilizando señales de pulsos.

*Conclusión:* Se estudiaron los aspectos que inherentes a la calibración de medidores de flujo líquido usando señales de pulsos. Se enfatizó en el aspecto referido a que datos se deben adquirir y como realizar el tratamiento de los pulsos para las mediciones (Sección 3.1.1 del capítulo 3)

- Identificar los requerimientos para una toma de mediciones de forma automática, para un diseño adecuado del sistema.

*Conclusión:* Se reconocieron los requerimientos para el diseño del sistema con base en el procedimiento de calibración por comparación usando señales de pulsos (Sección 4.1 del capítulo 4). Para ello se tomaron en cuenta las recomendaciones de los fabricantes sobre buenas prácticas, factores de influencia en las mediciones y los equipos instalados en el banco de calibración de medidores de flujo líquido de METROLAB (Secciones 3.2.1 y 3.3 del capítulo 3).

- Diseñar un sistema para la adquisición de datos y generación de reportes. Para ello, se realiza un análisis general, de requerimientos, arquitectura y/o interconexión de elementos necesarios.

*Conclusión:* Se diseñó el sistema de adquisición de datos y generación de reportes (sección 4.2 del capítulo 4). Dicho diseño se realizó con base en los requerimientos previamente definidos y que brindaron la base para las especificaciones del sistema.

- Validar el diseño, mediante la construcción de un prototipo que permita realizar la adquisición de datos en la calibración de un medidor de flujo líquido.

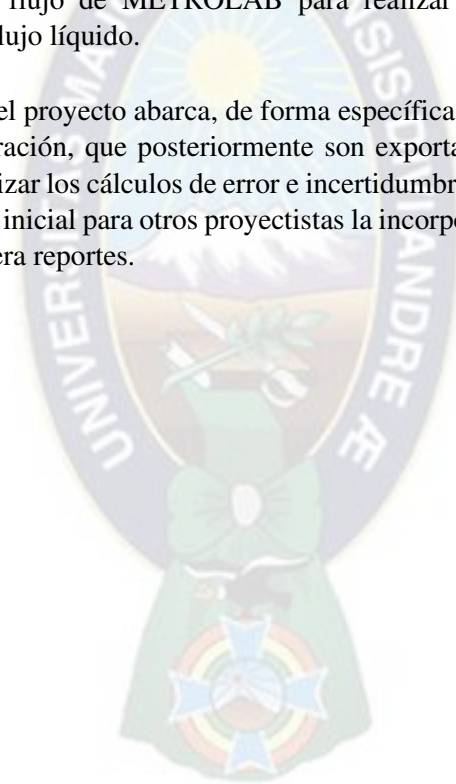
*Conclusión:* Se validó el diseño, mediante la construcción de un prototipo (Capítulo 5); con el cual, inicialmente se realizaron pruebas de funcionamiento y se tomaron datos de un medidor de flujo líquido (sección 6.1 del capítulo 6) y finalmente se verificó la confiabilidad de las mediciones obtenidas con el sistema (sección 6.2 del capítulo 6). Se observa que para realizar las mediciones con el sistema solo se requiere de un operador y

ya no son necesarias operaciones manuales con las válvulas y apunte de datos; así mismo, las medidas presentan confiabilidad.

## **7.2. RECOMENDACIONES.**

A partir de los resultados obtenidos en el desarrollo del proyecto, se pueden señalar las siguientes recomendaciones:

- Dado que el sistema obtiene medidas confiables evitando los inconvenientes presentados en el punto 1.3 del capítulo 1; además, de ser funcional en condiciones de laboratorio como se mostró en el capítulo 6. Entonces, se recomienda el uso de este sistema en el laboratorio de flujo de METROLAB para realizar mediciones en la calibración de medidores de flujo líquido.
- El desarrollo del proyecto abarca, de forma específica, la adquisición de datos necesarios para una calibración, que posteriormente son exportados a una hoja de cálculo en MS Excel para realizar los cálculos de error e incertidumbre para un reporte final. Por lo tanto, puede ser tarea inicial para otros proyectistas la incorporación de una etapa que realiza los cálculos y genera reportes.





## 8. Bibliografía

- [1] METROLAB, «METROLAB S.R.L.» [En línea]. Disponible en: <http://metrolab.com.bo/>.
- [2] CENTRO ESPAÑOL DE METROLOGIA CEM, Vocabulario Internacional de Metrología, JCGM, 2012.
- [3] TUVNEL LTDA, GOOD PRACTICE GUIDE THE CALIBRATION OF FLOW METERS, GLASGOW.
- [4] *Bolivia Decreto Supremo Nro. 24498, 1997.*
- [5] M. Kruger, «How automated calibration can help service companies be more competitive,» 2019. [En línea]. Disponible en: <https://blog.beamex.com/how-automated-calibration-can-help-service-companies-be-more-competitive>.
- [6] L. Renda, «Flowmeter calibration: Five best practices you need to know,» 13 Febrero 2020. [En línea]. Disponible en: <https://us.flukecal.com/blog/flowmeter-calibration-five-best-practices-you-need-know>.
- [7] N. Methies, «Calibration Technical fundamentals,» 2013.
- [8] YOKOGAWA FGP-140, «Pressure Transmitter Response Time,» 2008.
- [9] D. A. Snyder, Understanding 4-20 mA loops, Dakota, 2008.
- [10] J. M. Carvallo, Catalogo de Organismos Acreditados y Homologados, La Paz, 2020.
- [11] Organización de las naciones unidas contra la droga y el delito, Glosario de terminos sobre garantia de calidad y buenas practicas de laboratorio, Viena, 2012.
- [12] SERVICIO ECUATORIANO DE NORMALIZACIÓN, EXACTITUD (VERACIDAD Y PRECISIÓN) DE RESULTADOS Y SUS MÉTODOS DE MEDICIÓN - PARTE 1: PRINCIPIOS GENERALES Y DEFINICIONES, Quito Ecuador, 2015.
- [13] H. G. Postigo, «Estudio de Presición,» *Validación y Verificación de Métodos*, 2021.
- [14] BRITISH STANDARD, Measurement of fluid flow - Methods of specifying flowmeter performance BS ISO 11631, 1998.
- [15] S. Chun, B.-R. Yoon, H.-M. Choi y Y. Bong Lee, «Water Flow Meter Calibration with a Master Meter Method,» *INTERNATIONAL JOURNAL OF PRECISION ENGINEERING AND MANUFACTURING*, vol. 18, nº 8, 2017.

- [16] D. A. Loza Guerrero , *Curso Teórico Práctico - Metrología de Flujo*, Mexico: CENAM, 2006.
- [17] SCIENTIFIC COLUMBUS, SC10V Watthour portable standard user`s manual, 1998.
- [18] M. Wentao y S. Bin, «The Calibration System of Gas Flow meter Based on Standar Flow meter Method,» *Trans Tech Publications*, vol. 468, pp. 3122-3125, 2012.
- [19] M. P. Garcia, J. A. Anton y J. C. Rodriguez, *Instrumentación Electrónica*, Thomson.
- [20] J. L. C. Zenteno, *SISTEMA INTEGRADO DE ANÁLISIS-EVALUACIÓN VERIFICACIÓN DE LA CALIDAD DE ENERGIA ELÉCTRICA EN BOLIVIA*, La Paz, 2012.
- [21] DEWESOFT, «Que es Adquisición de Datos,» 2020. [En línea]. Disponible en: <https://dewesoft.com/es/daq/que-es-adquisicion-de-datos>.
- [22] T. Mortenson, «What are 2-wire and 4-wire Transmitter Output Loops?,» 25 Mayo 2020. [En línea]. Disponible en: <https://realpars.com/transmitter-wiring/>.
- [23] «Learning Instrumentation and Control Engineering,» 2020. [En línea]. Disponible en: <https://www.instrumentationtoolbox.com/2016/08/transmitter-voltage-and-power.html>.
- [24] «Back to Basics: The Fundamentals of 4-20 mA Current Loops,» 2022. [En línea]. Disponible en: <https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops>.
- [25] J. R. Lajara Vizcaíno y J. Pelegrí Sebastiá, *LabVIEW Entorno gráfico de programacion*, Catalanes: Macombo, 2007.
- [26] R. W. Larsen, *LabVIEW for Engineers*, New Jersey: Prentice Hall, 2011.
- [27] G. W. Johnson y R. Jennings, *LabVIEW Graphical Programming*, Chicago: MC Graw-Hill, 2006.
- [28] R. Bitter, T. Mohiuddin y M. Nawrocki, *LABVIEW ADVANCED PROGRAMMING TECHNIQUES*, New York: CRC Press, 2001.
- [29] C. J. Moreno Gomez, «LabVIEW core I y LabVIEWcore II,» udemy, 2017. [En línea]. Disponible en: <https://www.udemy.com/course/labview-university-de-cero-a-la-primera-certificacion/learn/lecture/10056166#overview>.
- [30] NATIONAL INSTRUMENTS, «Arquitectura de productor/consumidor en LabVIEW,» 11 Febrero 2021. [En línea]. Disponible en: <https://www.ni.com/es-cr/support/documentation/supplemental/21/producer-consumer-architecture-in-labview0.html>.

- [31] MICROCHIP, *dsPIC30F Family Reference Manual*, U.S.A., 2006.
- [32] MICROCHIP, dsPIC30F3014, dsPIC30F4013 Data Sheet, USA: MICROCHIP, 2004.
- [33] KHRONE, *Instrucciones de instalacion y funcionamiento ALTOFLUX IFC 110 F*.
- [34] ENDRES+HAUSER, *Technical Information iTEMP TMT 180*, 2018.
- [35] YOKOGAWA, *User's Manual Absolute Pressure and Gauge Pressure Transmitters*, 16 ed., 2020.

## ***ANEXO A – Periféricos y registros del microcontrolador DSPIC30F4013.***

Se brinda breve descripción sobre los periféricos de entradas y salidas digitales, entradas analógicas, timer y comunicación serial del microcontrolador dsPIC30F4013.

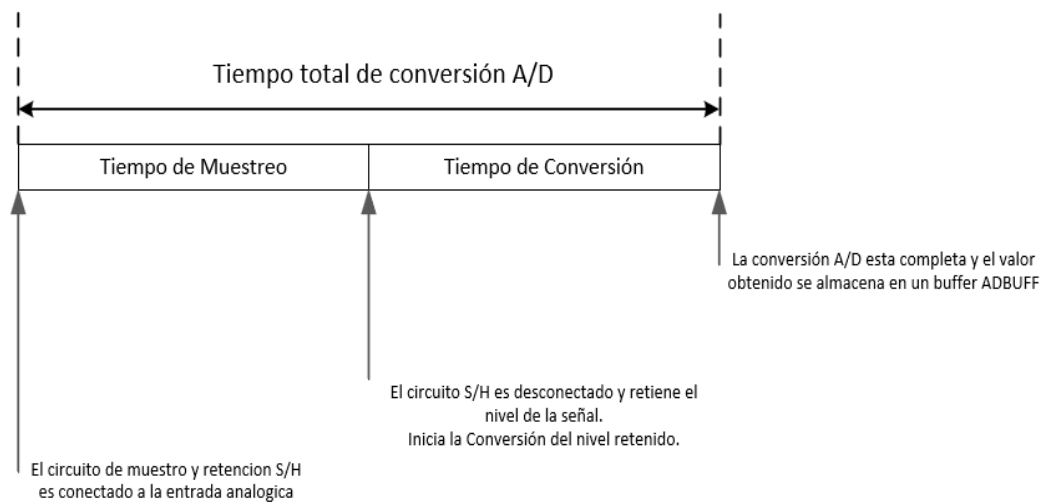
### **PINES DE ENTRADA SALIDA I/O.**

Para la configuración y manejo de un pin como entrada o salida se tienen 4 registros asociados: TRISx, PORTx y LATx; donde x toma algún valor según el puerto y numero de pin. El registro TRIS define un pin como entrada o salida, el registro PORT lee el estado de un pin y el registro LAT configura el estado de un pin de salida. Los registros para configurar los pines del microcontrolador son: TRISA, PORTA, LATA, TRISB, PORTB, LATB, TRISC, PORTC, LATC, TRISD, PORTD, LATD, TRISF, PORTF y LATF.

### **CONVERSION ANALÓGICO A DÍGITAL.**

El conversor analógico digital A/D del microcontrolador permite convertir una entrada analógica en una señal digital. En el dsPIC30F4013 la conversión se realiza a palabras digitales de 12 bits. Se pueden disponer de hasta 13 entradas analógicas (AN0, AN1, ..., AN12) en este microcontrolador. Los niveles de referencia analógicos son configurables por software a los niveles de alimentación (AVDD, AVSS) o a los niveles conectados en los pines ( $V_{REF+}/V_{REF-}$ ).

La obtención de la palabra digital que representa al nivel analógico de algún pin se realiza en dos etapas: Muestro y Retención (S/H por sus siglas en ingles) y la conversión. El muestreo y retención se encarga de seleccionar el canal analógico y obtener la señal muestreada. La etapa de conversión se encarga de la cuantificación y codificación, es decir, que asigna una palabra de 12 bits al valor muestreado. Una vez obtenido el valor digital este se almacena en un buffer. En la Figura 103 se muestra la línea de tiempo en la conversión.



*Figura 103. Tiempos de conversión Analógico a digital*

*Fuente: Hoja de datos DSPIC30F4013 [32].*

Se debe tener cuidado en cumplir con algunas especificaciones de tiempo para la conversión. Según el manual del dsPIC30F4013 el tiempo de conversión se define en unidades de TAD (Time Acquisition). El mínimo valor para un TAD es 667 ns. Este valor se puede configurar mediante los registros de control del microcontrolador.

Como el convertor es de 12 bits, los valores obtenidos en la conversión pueden ir desde 0 hasta  $2^{12} - 1$ , es decir, desde 0 hasta 4095. Así mismo, la resolución de la conversión estará definida por los niveles de tensión de referencia y la cantidad de bits del convertor A/D. Por ejemplo, si se selecciona los valores de alimentación como niveles de referencia es decir 0 V a 5 V, y como el conversor del dsPIC30F4013 es de 12 bits, se tendrá:  $Resolucion = \frac{5V-0V}{2^{12}} \rightarrow Resolucion = 0.805 mV$ .

Para la programación y funcionamiento del módulo A/D en el dsPIC30F4013 se tiene 6 registros: ADCON1, ADCON2, ADCON3, ADCHS, ADPCFG y ADCSSL. Los registros de control ADCON1, ADCON2 y ADCON3 definen la operación del convertor A/D. el registro ADCHS selecciona los canales de entrada a convertir. El registro ADPCFG define los pines del microcontrolador que funcionaran como entradas analógicas. El registro ADCSSL sirve si se desea

realizar un escaneo de los canales. Después de la conversión los valores obtenidos se almacenan en un buffer ADCBUF0 ADCBUF1 ... ADCBUFF.

Los pasos para realizar una conversión se resumen a continuación [31]:

- 1) Configurar el módulo A/D:
  - Configurar pines analógicos, niveles de referencia.
  - Selecciona el canal de entrada analógica a convertir.
  - Definir el tiempo de conversión en base al TAD.
  - Iniciar la secuencia de conversión A/D
- 2) Mandar señal de disparo para inicio de muestreo y retención.
- 3) Esperar mientras se realiza el muestreo y retención.
- 4) Cuando ocurre el disparo por finalización de muestreo, iniciar la conversión.
- 5) Esperar a que la conversión finalice.
- 6) Leer el resultado del Buffer A/D.

### **TIMER's.**

El módulo Timer (temporizador) es un contador integrado en el microcontrolador dsPIC30F4013. Para este modelo específico de microcontrolador se tienen 5 Timer's de 16 bits designados como TIMER1, TIMER2, TIMER3, TIMER4 Y TIMER5.

Los Timer son clasificados en 3 tipos de acuerdo a ciertas características únicas:

- Tipo A. Puede ser usado en aplicaciones de Real-Time Clock (RTC)
- Tipo B. Se pueden combinar con Timer de tipo C para formar Timer de hasta 32 bits.
- Tipo C. Se puede combinar con Timer tipo B para formar Timer de 32 bits. Así mismo, tiene la capacidad de disparar una conversión A/D.

Cada uno de los Timer's puede operar en modo contador. En este modo, una señal externa actúa como reloj para el Timer. La señal externa se debe conectar al pin externo denotado como TXCK, donde X puede tomar el valor de 1, 2 o 4, dependiendo de que timer se usara para contar dicha señal. En la figura 17, más arriba, se puede ver los pines correspondientes a T1CK y T2CK en las patillas 16 y 15 respectivamente.

A continuación, se da una breve descripción de los registros necesarios para configuración y funcionamiento de los módulos Timer. Se debe tener en cuenta que x puede tomar los valores de 1, 2, 3, 4 y 5 dependiendo del Timer a utilizar.

- TMRx, registro que guarda el valor actual de conteo del Timer.
- PRx, registro de periodo asociado con el Timer.
- TxCON, registro de control asociado con el Timer.

Los Timers pueden generar interrupciones por comparación. Estas interrupciones ocurren cuando el valor de conteo en TMRx es igual al valor guardado en el registro PRx. Cuando se cumple la condición  $TMRx = PRx$ , un bit TxIF se pone en 1 indicando la ocurrencia del evento. Para la habilitación de interrupción en el Timer se debe poner a 1 el bit TxIE.

Como se explicó los Timers del dsPIC30F4013 pueden usarse como contadores. De esta forma se puede conectar una señal externa a los pines del microcontrolador y de esta forma obtener el valor del conteo. Así mismo se puede combinar dos Timers y obtener un Timer de 32 bits [31].

## **MÓDULO UART.**

El módulo UART (Universal Asynchronous Receiver Transmitter) del dsPIC30F4013 permite la comunicación con otros dispositivos como ser computadoras, módulos periféricos y otros microcontroladores. La comunicación que se puede realizar con este módulo es Full dúplex, Asíncrona, protocolos RS232 y RS485. Así mismo, este microcontrolador tiene 2 módulos UART disponibles para su uso.

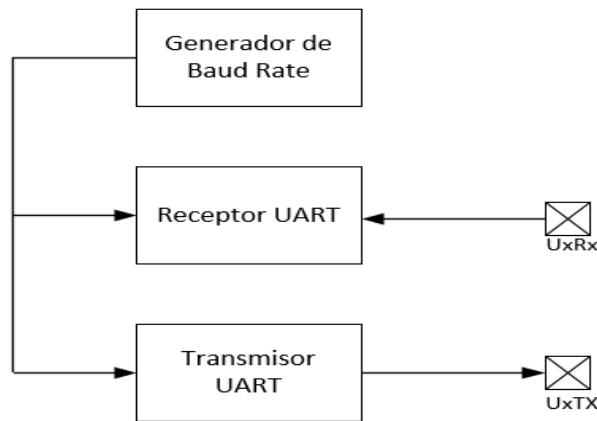
Las características principales del módulo UART son:

- Comunicación *Full duplex* de 8 o 9 bits.
- Opciones de paridad par, impar o no paridad.
- Uno o dos bits de stop.
- Generador de tasa de Baudios completamente integrado.
- Tasas de baudios desde 29 bps hasta 1.875 Mbps.
- Buffer de transmisión de 4 palabras de tipo FIFO (*Fist Input First Output* – Primero en entrar primero en salir).
- Buffer de recepción de 4 palabras tipo FIFO.
- Capacidad de generar interrupciones por transmisión o recepción de datos.

Un diagrama de bloques simplificado del módulo UART se muestra en la figura 19. Los pines para recepción y transmisión de datos están denotados con UxRX y UxTX respectivamente, donde x puede valer 1 o 2. También, los elementos clave para su funcionamiento son:



- Generador de Baud Rate. Se encarga de generar una señal de reloj con una frecuencia tal que sea compatible con la tasa de bits por segundo en recepción y transmisión.
- Transmisor asíncrono. Transmite los datos del buffer de transmisión.
- Receptor asíncrono. Es el encargado de recibir los datos y enviarlos al buffer de recepción.



*Figura 104. Diagrama de bloques simplificado de módulo UART.*

*Fuente: Manual de referencia DSPIC30F4013[31].*

En la comunicación con otros dispositivos, las características como ser Baudios, paridad, bit stop, cantidad de bits deben ser las mismas. Para la programación del módulo UART se tienen los siguientes registros: UxMODE, UxSTA, UxTXTEG, UxRXREG y UxBRG.

A continuación, se detalla los pasos a seguir para programar el módulo UART en modo de transmisión y recepción:

### ***Transmisión.***

- 1) Configurar el módulo UART. Se debe definir la cantidad de bits, paridad, velocidad de transmisión, tasa de Baudios, pines I/O para transmisión y recepción, entre otros.
- 2) Habilitar el módulo UART.
- 3) Habilitar la transmisión en el módulo UART.
- 4) Escribir los bytes a ser transmitidos en el buffer de transmisión.
- 5) Al finalizar la transmisión, se puede generar una interrupción dependiendo de la aplicación deseada.

### ***Recepción.***

- 1) Configurar el módulo UART. Se debe definir la cantidad de bits, paridad, velocidad de transmisión, tasa de Baudios, pines I/O para transmisión y recepción, entre otros.
- 2) Habilitar el módulo UART.
- 3) Se puede habilitar una interrupción cada vez que se tengan uno o mas bytes en el buffer de recepción. Todo depende de la aplicación y algoritmo utilizado en la programación.
- 4) Leer el dato recibido del buffer de recepción.

El detalle y designación de bits de los registros asociados a los módulos del dsPIC30F4013 se encuentra en su hoja de datos [32].

**ANEXO B – Valores de  $k$  crítico para análisis de consistencia de datos.**

Indicadores para "h" de Mandel y estadístico "K" a 5% de nivel de confianza											
		k									
		n									
p	h	2	3	4	5	6	7	8	9	10	
3	1.15	1.65	1.53	1.45	1.4	1.37	1.34	1.32	1.3	1.29	
4	1.42	1.76	1.59	1.5	1.44	1.4	1.37	1.35	1.33	1.31	
5	1.57	1.81	1.62	1.53	1.46	1.42	1.39	1.36	1.34	1.32	
6	1.66	1.85	1.64	1.54	1.48	1.43	1.4	1.37	1.35	1.33	
7	1.71	1.87	1.66	1.55	1.49	1.44	1.41	1.38	1.36	1.34	
8	1.75	1.88	1.67	1.56	1.5	1.45	1.41	1.38	1.36	1.34	
9	1.78	1.90	1.68	1.57	1.5	1.45	1.42	1.39	1.36	1.35	
10	1.80	1.90	1.68	1.57	1.5	1.46	1.42	1.39	1.37	1.35	

p = número de laboratorios o analistas a un nivel dado  
n = número de réplicas en cada uno de los laboratorios a ese nivel

Tabla 37. Valores de  $k$  crítico.

Fuente: Estudio de Precisión [13].

***ANEXO C – Valores de C crítico para análisis de datos atípicos.***

TABLA 4 - Valores críticos para prueba de Cochran						
p	n=2		n=3		n=4	
	1%	5%	1%	5%	1%	5%
2	-	-	0,995	0,975	0,979	0,939
3	0,993	0,967	0,942	0,871	0,883	0,798
4	0,968	0,906	0,864	0,768	0,781	0,684
5	0,928	0,841	0,788	0,684	0,696	0,598
6	0,883	0,781	0,722	0,616	0,626	0,532
7	0,838	0,727	0,664	0,561	0,568	0,48
8	0,794	0,68	0,615	0,516	0,521	0,438
9	0,754	0,638	0,573	0,478	0,481	0,403
10	0,718	0,602	0,536	0,445	0,447	0,373

p = número de laboratorios o analistas a un nivel dado.  
n = número de desviaciones estándar para el estudio.

*Tabla 38. Valores de C crítico.*

*Fuente: Estudio de Precisión [13].*

## ***ANEXO D – Valores de $t_{95}$ para cálculo de repetibilidad.***

En la Tabla 39 se detallan los valores de  $t_{95}$  necesarios para realizar el cálculo de repetibilidad.

Grados de Libertad (n-1)	$t_{95}$
1	12,706
2	4,303
3	3,182
4	2,776
5	2,571
6	2,447
7	2,365
8	2,306
9	2,262
10	2,228
11	2,201
12	2,179
13	2,16
14	2,145
15	2,131

*Tabla 39. Valores  $t_{95}$  para cálculo de repetibilidad.*

*Fuente: Methods of Specifying Flowmeter Performance [14].*

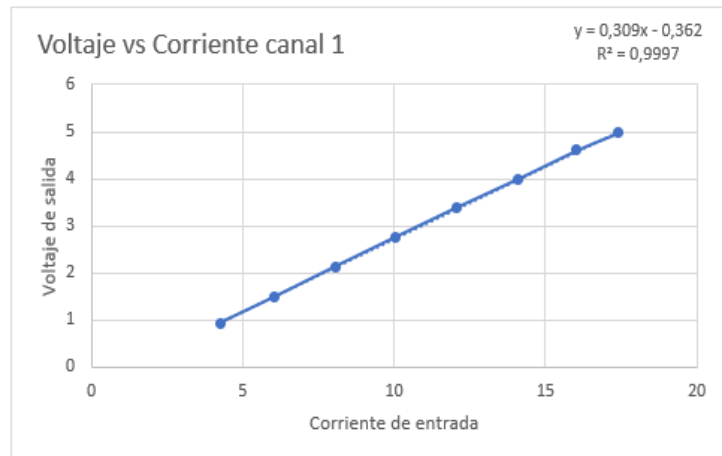
## ***ANEXO E – Relación Voltaje-Corriente de los canales del módulo M1 del Hardware.***

Se muestra los datos obtenidos en las medidas de voltaje de salida y corriente de entrada, la gráfica de tendencia de los datos y la ecuación de relación obtenida por regresión; en cada canal del circuito de medida electrónico.

### **Canal 1.**

#### **Datos.**

Corriente de entrada (mA)	Voltaje de salida (V)
4,26	0,937
6,06	1,493
8,06	2,137
10,04	2,762
12,05	3,388
14,09	3,997
16,03	4,603
17,40	4,973
19.97	5.102



*Figura 105. Relación voltaje corriente canal 1.  
Fuente: Elaboración propia.*

*Tabla 40. Datos de voltaje-corriente en canal 1.  
Fuente: Elaboración propia.*

**Ecuaciones:**  $V_o = 0,3089 * I - 0,3617 \leftrightarrow I = 3,2374 * V_o + 1,1709$

$$R^2 = 0,9997$$

## Canal 2.

### Datos

Corriente de entrada (mA)	Voltaje de salida (V)
4,25	0,663
6,05	1,059
7,99	1,499
10,06	1,957
12,08	2,409
14,01	2,845
16,02	3,289
18,01	3,718
20,50	4,284

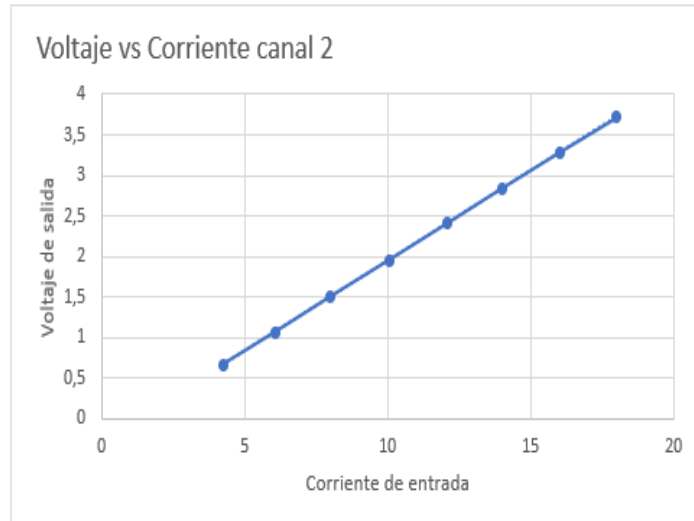


Figura 106. Relación voltaje corriente canal 2.  
Fuente: Elaboración propia.

Tabla 41. Datos de voltaje-corriente en canal.  
Fuente: Elaboración propia.

**Ecuaciones:**  $V_o = 0,2228 * I - 0,2839 \leftrightarrow I = 4,4890 * V_o + 1,2742$

$$R^2 = 1$$

### Canal 3.

#### Datos

Corriente de entrada (mA)	Voltaje de salida (V)
4,24	0,511
6,04	0,879
7,99	1,293
10,00	1,738
12,02	2,205
14,01	2,649
16,02	3,108
17,99	3,577
19,97	4,013

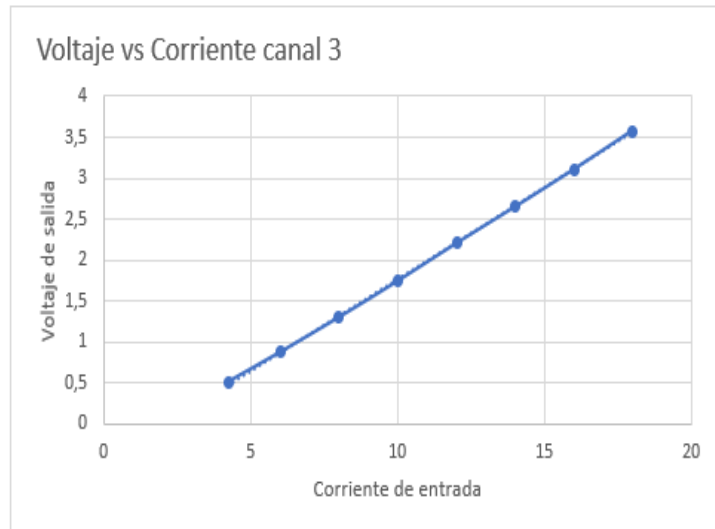


Figura 107. Relación voltaje corriente canal 3.  
Fuente: Elaboración propia.

Tabla 42. Datos de voltaje-corriente en canal 3.  
Fuente: Elaboración propia.

$$\text{Ecuaciones: } V_o = 0,3089 * I - 0,3617 \quad \leftrightarrow \quad I = 3,2374 * V_o + 1,1709$$

$$R^2 = 0,9996$$



#### Canal 4.

#### Datos

Corriente de entrada (mA)	Voltaje de salida (V)
4,25	0,8113
6,08	1,2871
8,10	1,8123
10,04	2,3167
12,11	2,8549
14,05	3,3593
16,12	3,8975
17,99	4,3837
20,3	4,9843

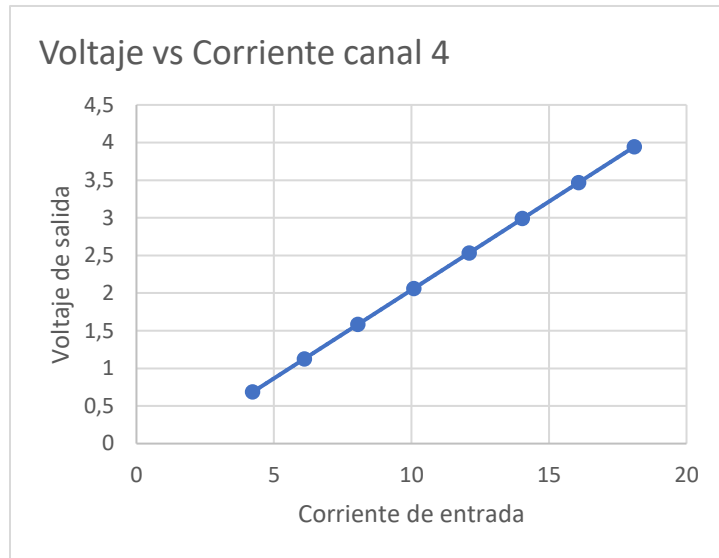


Figura 108. Relación voltaje corriente canal 4.  
Fuente: Elaboración propia.

Tabla 43. Datos de voltaje-corriente en canal 4.  
Fuente: Elaboración propia.

**Ecuaciones:**  $V_o = 0,2600 * I - 0,2937 \leftrightarrow I = 3,8455 * V_o + 1,1294$

$$R^2 = 1$$

## Canal 5.

### Datos

Corriente de entrada (mA)	Voltaje de salida (V)
4,23	0,686
6,11	1,127
8,05	1,583
10,09	2,062
12,10	2,534
14,04	2,990
16,08	3,469
18,11	3,946
20,1	4,414

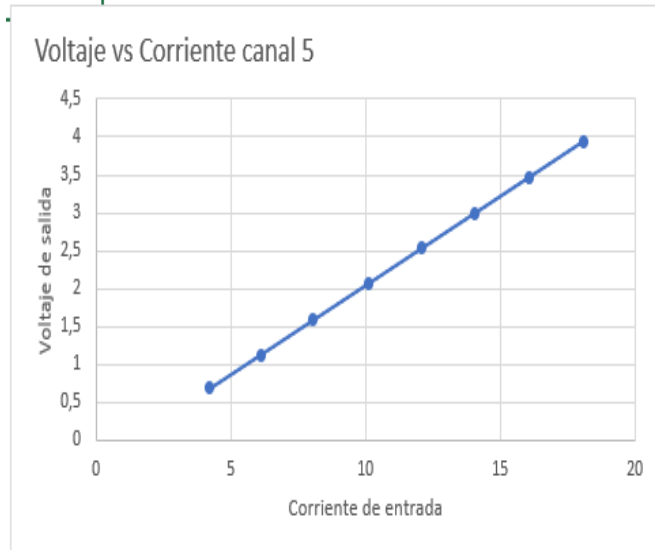


Figura 109. Relación voltaje corriente canal 5.  
Fuente: Elaboración propia.

Tabla 44. Datos de voltaje-corriente en canal 5.  
Fuente: Elaboración propia.

$$\text{Ecuaciones: } V_o = 0,2349 * I - 0,3079 \quad \leftrightarrow \quad I = 4,2557 * V_o + 1,3103$$

$$R^2 = 1$$

## ***ANEXO F – Código completo del microcontrolador.***

```
#include <xc.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h> // librería para manejar cadenas de caracteres
#include <math.h> // librería para realizar operaciones matemáticas
#include "proyecto_grado_v1_header.h"
#include <libpic30.h>
#include <p30F4013.h> // librería para retardos del dsPIC30F4013

//*****DEFINICION DE CONSTANTES PARA EL
PROGRAMA*****
//****constantes para los timers (conteo de pulsos)*****
#define habilitacion LATAbits.LATA11 //señal para habilitar el conteo de pulsos
#define flag_timer3 IFS0bits.T3IF //bandera de interrupcion del timer 3
#define flag_timer1 IFS0bits.T1IF //bandera de interrupcion del timer 1
#define flag_int2 IFS1bits.INT2IF

//*****
//****algunas constantes adicionales para el programa*****
#define indicador LATBbits.LATB9 // auxiliar para indicacion
#define Freq 20000000ULL //frecuencia del oscilador 20MHz
#define delay_ms(x) __delay32((Freq/1000)*x) //funciona para retardo delay
#define T 50 //retardo para el bucle infinito

//*****
//*****FIN DEFINICION DE CONSTANTES PARA EL
PROGRAMA*****
//*****DEFINICION DE FUNCIONES PARA EL
PROGRAMA*****
//****funciones para el uso del módulo ADC*****
void init_adc(); // funcion para configuracion de ADC
int leer_adc(int canal); //funcion para leer algun pin AN

//*****
//**funciones para el uso de los timers (conteo de pulsos)*****
void init_timer23(); //funcion para inicio del timer 2/3
void init_timer1(); //funcion para inicio del timer 1
```

```

//*****
//*****funciones para el uso del módulo de comunicación UART2*****
void init_uart2(); //función para configurar los registros del UART
void enviar_char(char ch); //función para enviar carácter desde el puerto serial
void enviar_string(char *str); //función para enviar un string desde el puerto serial
void init_int2();//función para inicializar la interrupción externa 2

//*****FIN      DEFINICION      DE      FUNCIONES      PARA      EL
PROGRAMA*****

//*****DEFINICION      DE      VARIABLES      PARA      EL
PROGRAMA*****
/**variables para el uso del ADC*****
int an0 = 0,an1 = 0,an2 = 0,an3 = 0,an4 = 0,an5 = 0,an6 = 0; // variable para la
lectura analogica
//*****
/**variables para el uso de timers (conteo de pulsos)*****
unsigned int count_patron_high = 0; //parte alta del contador patron
unsigned long int cuentas_programadas = 0; //donde se almacenaran las cuentas
a realiza
unsigned long int count_ibp = 0, count_patron = 0; //contadores del ibp y patron
int end_count=0;

//*****
/**variables para el uso del modulo de comunicacion UART2*****
char conteo_str[] = "0000000000"; //donde se almacenara los caracteres
recibidos para el conteo
int BRGVAL = 10; //para baudrate de 115200 con 1.5% de error
char rxbyte; //byte que se recibe del puerto serie
int estado1 = 0; //estado para recepción de caracteres
int i = 0; //potencia para la recepción del valor de valor a conta
char trama[50]; //para enviar los datos obtenidos en lecturas analógicas y de
pulsos

//*****
//*****FIN      DEFINICION      DE      VARIABLES      PARA      EL
PROGRAMA*****

```

```

****interrupcion cuando se recibe dato dep uart2 ****
void __attribute__((__interrupt__, no_auto_psv)) _U2RXInterrupt(void) {
    rxbyte = U2RXREG; // se lee el valor del registro
    if (estado1 == 0) {
        switch (rxbyte) {

            case 'i': //comando para iniciar el conteo de pulsos
            {
                count_patron_high = 0;
                TMR1 = 0;//el timer 1 se pone a 0
                TMR2 = 0;//el timer 2 se pone a 0
                TMR3 = 0;//el timer 3 se pone a 0
                habilitacion = 1;
            } break;

            case 'd': //comando para detener un conteo ya iniciado
                habilitacion = 0;
                break;

            case 'r'://inicio de registro para conteo de pulsos
            {
                estado1 = 1;
                int z = 0;
                for (z = 0; z <= 9; z++) conteo_str[z] = "0";
                i = 0;
            } break;
        }
    } else if (rxbyte == 'f') { //finalizacion del registro de cuentas
        estado1 = 0;
        cuentas_programadas = atol(conteo_str);
        PR2=cuentas_programadas&0x0000FFFF;
        PR3 = (cuentas_programadas & 0xFFFF0000) >> 16;
    } else {
        char *p; //puntero para almacenar en char
        p = &conteo_str[i];
        *p = rxbyte;
        i++;
    }
    IFS1bits.U2RXIF = 0; //limpiamos para otra interrupcion
}

```

```

//*****
//*****interrupción del timer 3 para contador de 32bits*****
void __attribute__((__interrupt__, __auto_psv__)) _T3Interrupt(void) {
    __delay32(10);
    habilitacion = 0; //inhabilitamos la señal para conteos
    flag_timer3 = 0; //limpiamos el flag para otra interrupcion del timer 3
}

//*****
//*****interrupción timer 1*****
void __attribute__((__interrupt__, __auto_psv__)) _T1Interrupt(void) {
    count_patron_high++; //incrementa la parte alta del contador para el patron
    flag_timer1 = 0; //limpiamos el flag para otra interrupcion del timer 3
}

//*****
//*****interrupción externa 2 que indica que ya acabo el conteo*****
void __attribute__((interrupt, no_auto_psv)) _INT2Interrupt(void) {
    delay_ms(5);
    end_count=1; //indica que conteo finalizo se procesa en el loop
    flag_int2=0; //limpiamos la bandera de la interrupcion externa 2
}

//*****
int main(void) {

    //init_io();
    TRISBbits.TRISB9 = 0; //definir como salida B12(pin para señal indicador)
    indicador=1; //para indicar el inicio de la configuracion

    init_adc(); //inicializar el modulo ADC(configuracion)
    ADCON1bits.ADON = 1; //habilita el modulo ADC

    TRISAbits.TRISA11 = 0; //definir como salida A11 (pin para señal habilitacion)
    init_timer23(); //inicializa el timer 23 modo contador 32 bits
    init_timer1(); //inicializa el timer 1 modo contador 16 bits
    habilitacion = 0; //la señal de habilitacion de conteo inicialmente en 0

    init_uart2(); //inicializa el modulo UART2para comunicacion serial

```

```

init_int2();//inicializacmo la int ext 2 para ver cuando finaliza el conteo y enviar
la info al software labview
delay_ms(100);
indicador=0;//para indicar el final de configuracion del microcontrolador

while (1) {

    indicador=!indicador;

    //lectura de canales analogicosanalogicos
    *****

    an0 = leer_adc(0); delay_ms(20);
    an1 = leer_adc(1); delay_ms(20);
    an2 = leer_adc(2); delay_ms(20);
    an3 = leer_adc(3); delay_ms(20);
    an4 = leer_adc(4); delay_ms(20);
    //parpadear(2);
    //*****

    /**lectura de los contadores de pulsos del patron e ibp
    count_ibp = ((unsigned long int) TMR3 << 16) | (unsigned long int) TMR2;
    count_patron = ((unsigned long int) count_patron_high << 16) | (unsigned
long int) TMR1;
    //*****

    /**envio de datos al puerto serial para
interfaz*****
    sprintf(trama, "S%d+%d+%d+%d+%d+%lu+%lu+%d\n", an0, an1, an2, an3,
an4, count_ibp, count_patron,0); //se envia la trama de datos
    enviar_string(trama);
    //*****

    /****preguntamos si finalizo un conteo iniciado*****
    if(end_count){
    sprintf(trama, "S%d+%d+%d+%d+%d+%lu+%lu+%d\n", an0, an1, an2, an3,
an4, count_ibp, count_patron,1);
    enviar_string(trama);
    end_count=0;

```

```

    }
    ///*****
    delay_ms(T);
}
return 0;
}
///*****funciones para el uso del ADC*****

void init_adc() {
    TRISBbits.TRISB0 = 1; //pin B0 como entrada
    TRISBbits.TRISB1 = 1;
    TRISBbits.TRISB2 = 1;
    TRISBbits.TRISB3 = 1;
    TRISBbits.TRISB4 = 1;
    TRISBbits.TRISB5 = 1;
    ADPCFG = 0xFFC0; //pinES B0-B5 como analogico
    ADCON1 = 0x0000; //
    ADCON1bits.SSRC = 0b111; //conversion inicia automaticamente despues de
muestreo
    ADCON2 = 0x0000; //
    ADCON3bits.ADCS = 0b011010; //para garantizar Tad=667ns
    ADCON3bits.SAMC = 5; //15Tad entre muestreo y conversion
    ADCHSbits.CH0NA = 0; //negativo del ADC a Vref-
    ADCSSL = 0x0000; //ningun canal sera escanead
}

int leer_adc(int canal) {
    int lectura = 0;
    switch (canal) { //se elige el canal AN positivo del ADC
        case 0: ADCHSbits.CH0SA = 0b0000;
            break;
        case 1: ADCHSbits.CH0SA = 0b0001;
            break;
        case 2: ADCHSbits.CH0SA = 0b0010;
            break;
        case 3: ADCHSbits.CH0SA = 0b0011;
            break;
        case 4: ADCHSbits.CH0SA = 0b0100;
            break;
    }
}

```



```

        case 5: ADCHSbits.CH0SA = 0b0101;
            break;
        default: ADCHSbits.CH0SA = 0b0000;
            break;
    }
    ADCON1bits.SAMP = 1; //inicia muestreo entonces 31Tad despues la
conversion
    while (!ADCON1bits.DONE); //pregunta si la conversion esta lista
    lectura = ADCBUF0;
    return lectura;
}

//*****
//***funciones para el uso de los timer 1 (16bits) y timer 2/3 en modo 32
bits*****
void init_timer23() {
    TRISCbits.TRISC13 = 1; //T2CK como entrada
    T2CON = 0;
    T2CONbits.T32 = 1; //timer 23 modo 32bits
    T2CONbits.TCS = 1; //fuente de clock externa
    IEC0bits.T3IE = 1; //habilita interrupcion por timer
    TMR2 = 0; //Se limpia timer 2
    TMR3 = 0; //se limpia timer 3
    PR2 = 0xFFFF; //se pone al maximo valor de conteo
    PR3 = 0xFFFF;
    T2CONbits.TON = 1; //se habilita el timer 2
}

void init_timer1() {
    TRISCbits.TRISC14 = 1; //T1CK como entrada
    T1CON = 0;
    T1CONbits.TCS = 1; //fuente de reloj externa
    T1CONbits.TSYNC = 1; //modo contador sincronizado
    PR1 = 0xFFFF; //interrupcion por desborde en la maxima cuenta
    TMR1 = 0x0000; //se limpia el valor del timer 1
    IEC0bits.T1IE = 1; //se habilita interrupcion del timer 1
    IFS0bits.T1IF = 0; //flag del timer 1 inicialmente en 0
    T1CONbits.TON = 1; //timer 1 activado
}

```

```

//*****
//*****funciones para el uso del UART2*****
void init_uart2() {
    U2MODE = 0; //Ponemos a 0 el registro de control del UART2
    U2BRG = BRGVAL; // se carga valor para genera el baudrate 115200 baudios
    U2MODEbits.PDSEL = 0b00; // 8 bits, sin paridad
    U2MODEbits.STSEL = 0; //1 bit de stop
    U2MODEbits.ABAUD = 0; //Auto-Baud, Deshabilitado
    U2STAbits.URXISEL = 0; //interrupción cuando un carácter es recibido en RX
    IPC6bits.U2RXIP = 1; //PRIORIDAD BAJA DE LA INTERRUPCION
    IEC1bits.U2RXIE = 1; //habilita interrupción por recepción UART 2
    IFS1bits.U2RXIF = 0; //se limpia la bandera de la interrupción de recepción de
carácter
    U2MODEbits.UARTEN = 1; //habilita el UART
    U2STAbits.UTXEN = 1; //habilita la transmisión Tx del UART
}

void enviar_string(char *str) { //funcion para enviar string
    char *p; // define variable tipo puntero
    p = str; // apuntamos a la direccion de la cadena de caracteres
    while (*p) enviar_char(*p++); //se envia cada uno de los caracteres del string
}

void enviar_char(char ch) {
    while (U2STAbits.UTXBF == 1) ; //espera a que buffer de transmisión este vacío
    U2TXREG = ch; //carga el carácter en el buffer de transmisión
}

//*****definición de funciones para la interrupción externa 2*****
void init_int2(){
    TRISDbits.TRISD9 = 1; // se defina el D9 como entrada para la int ext 2
    flag_int2=0; //borramos el flag de la int2
    INTCON2bits.INT2EP=1; //polaridad negativa de la int2
    IEC1bits.INT2IE=1; // habilitamos la int ext 2
}

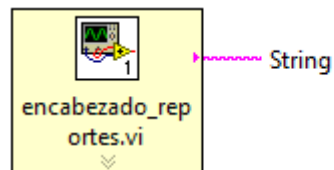
//*****
//*****

```

## ANEXO G – Funciones del programa en LabVIEW.

Se muestran una descripción y el diagrama de bloques de los subVI's utilizados en el programa de LabVIEW.

### encabezado\_reportes.



Este subVI genera el encabezado del reporte, se imprime antes de los valores

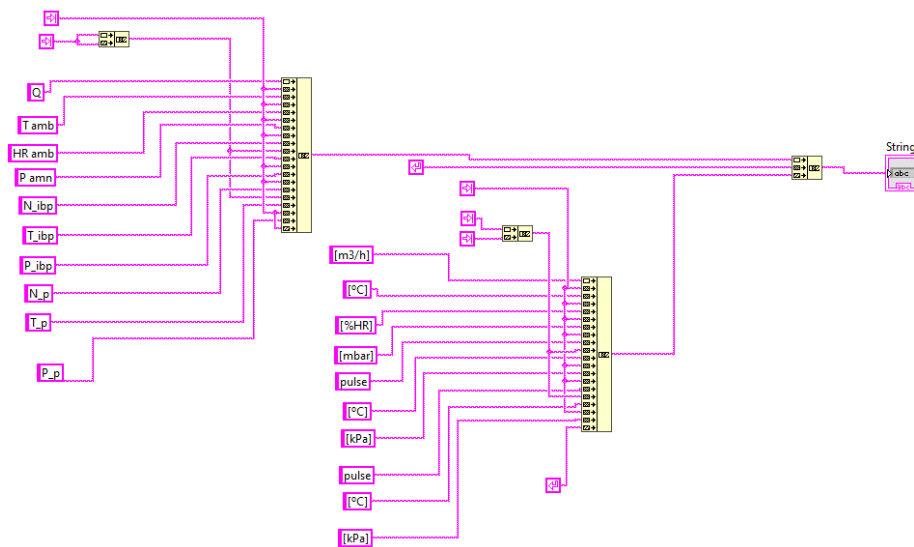


Figura 110. SubVI encabezado\_reportes.

Fuente: Elaboración propia.

## Separa datos.



este subVI separa los datos en formato String que vienen del microcontrolador y los convierte en arrays, de esta forma se tiene un array para los sensores, 1 array para los valores de los conteos y 1 elemento numerico que indica en que estado esta actualmente el sistema

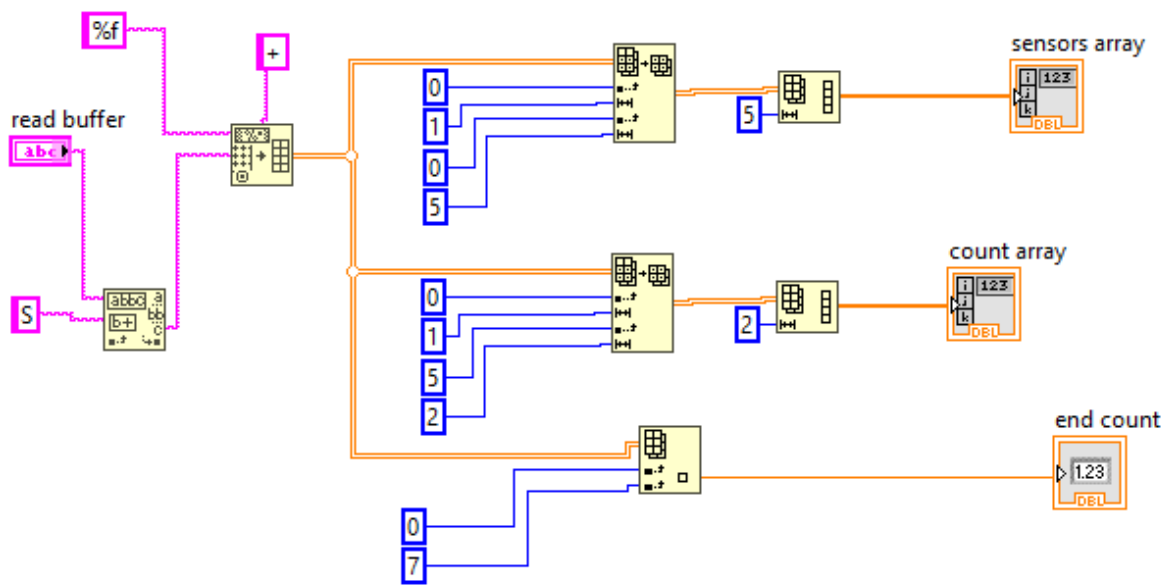
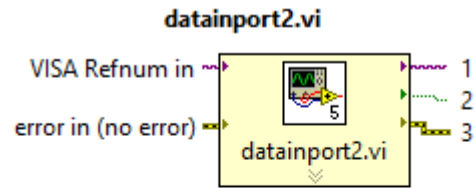


Figura 111. SubVI separa datos.

Fuente: Elaboración propia.

## datainport2.



### Outputs

- 1 VISA Refnum out
- 2 Boolean
- 3 error out

Este Sub Vi consulta si hay datos disponibles en el puerto serial

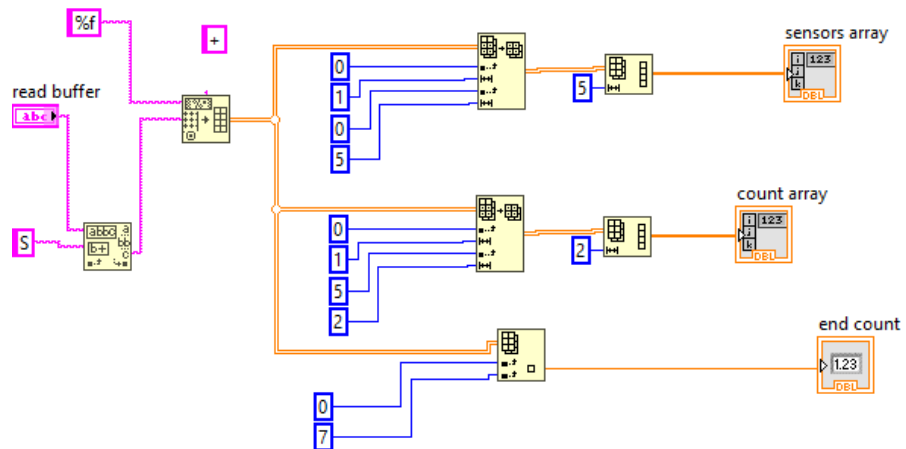


Figura 112. SubVI datainport2.

Fuente: Elaboración propia.

## ADC\_valores.

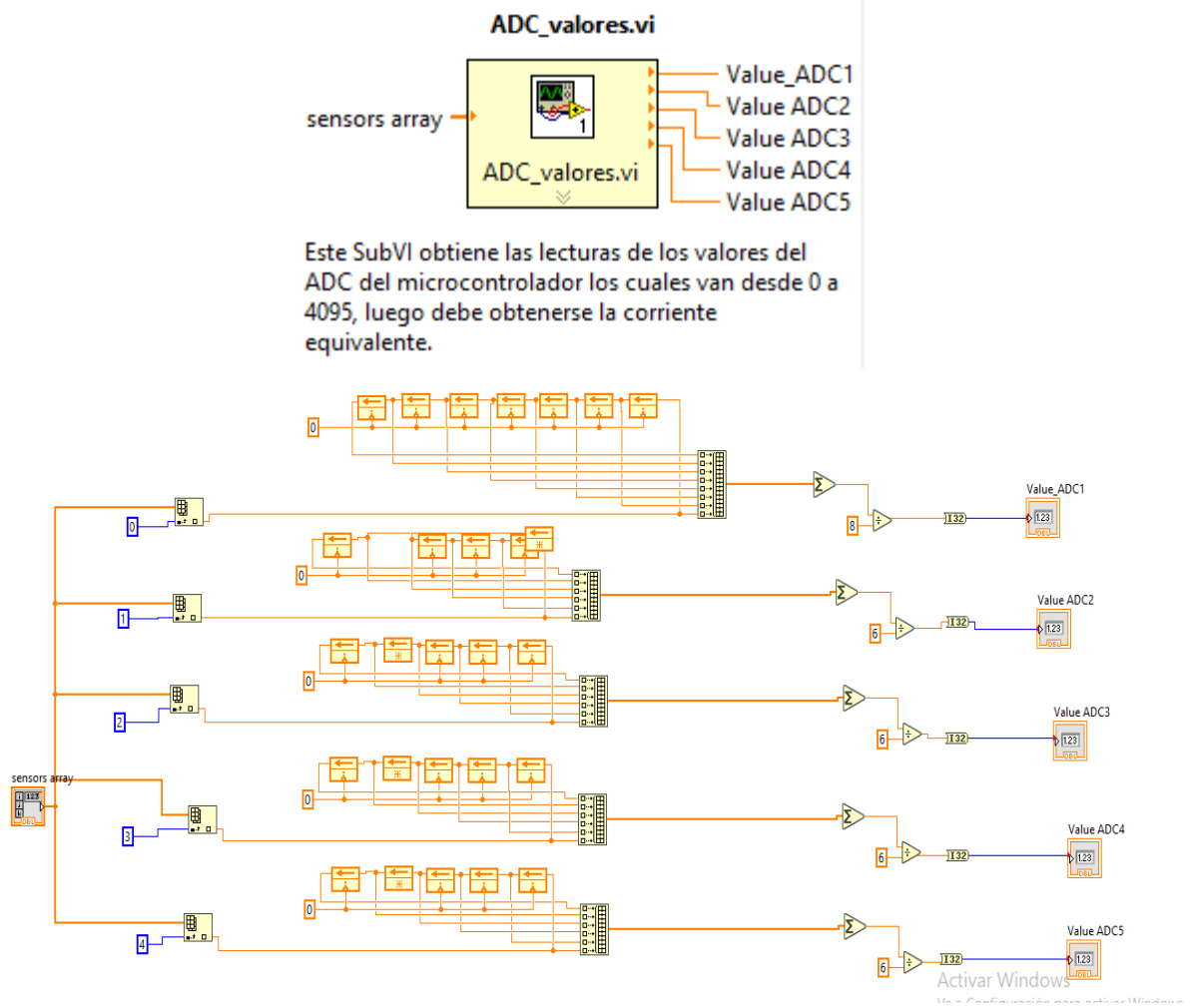
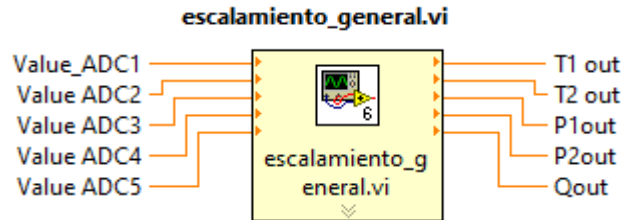


Figura 113. SubVI ADC\_valores.

Fuente: Elaboración propia.

## escalamiento general.



Este Sub VI lee los valores del ADC del microcontrolador y los convierte en corriente, posteriormente realiza el escalamiento de las variables de temperatura, presión y caudal y se tiene las variables escaladas en los límites especificados por constantes, acá se realiza la calibración de la lectura de los sensores.

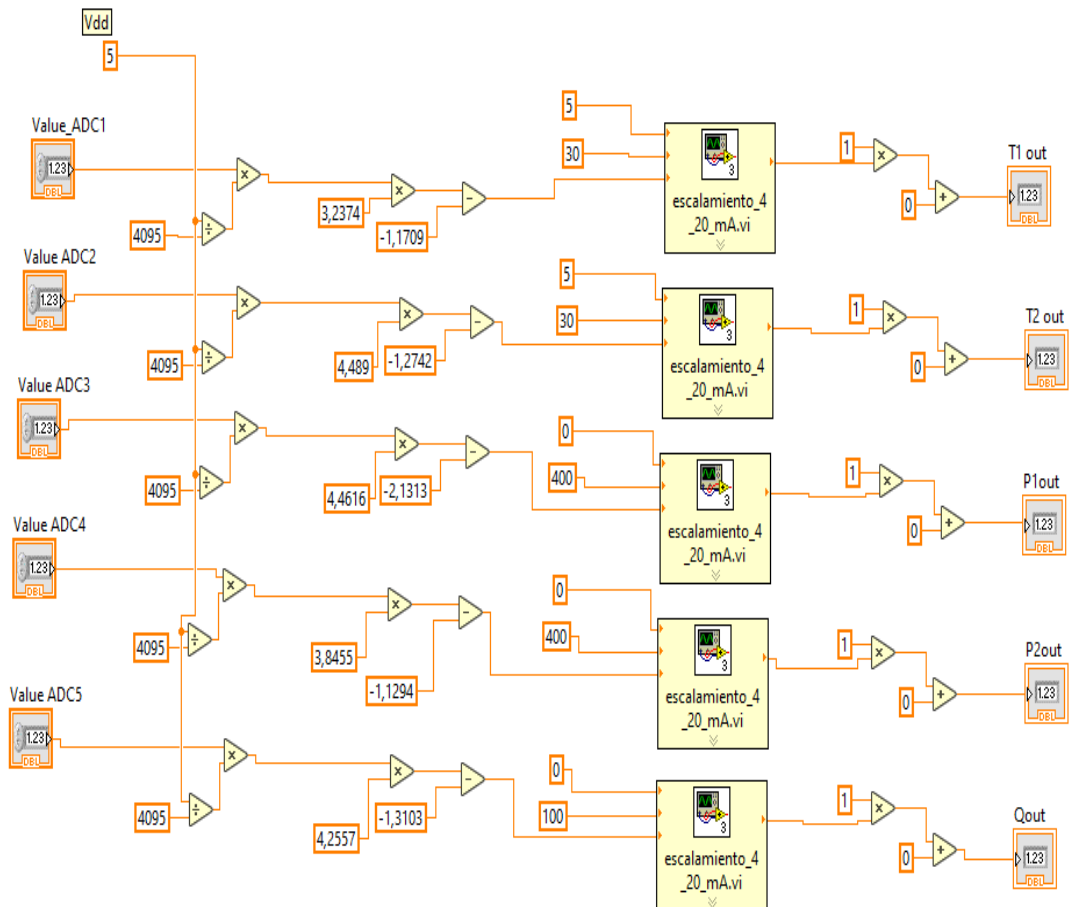
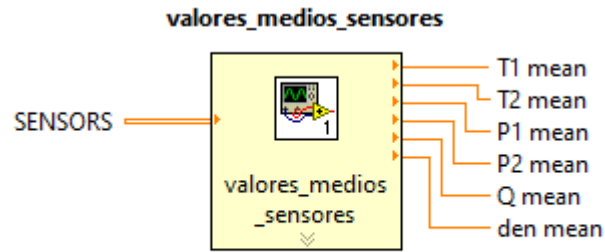


Figura 114. SubVI escalamiento\_general.

Fuente: Elaboración propia.

valores\_medio\_sensores.



Este sub VI obtiene el valor medio de los valores provenientes guardados de una corrida

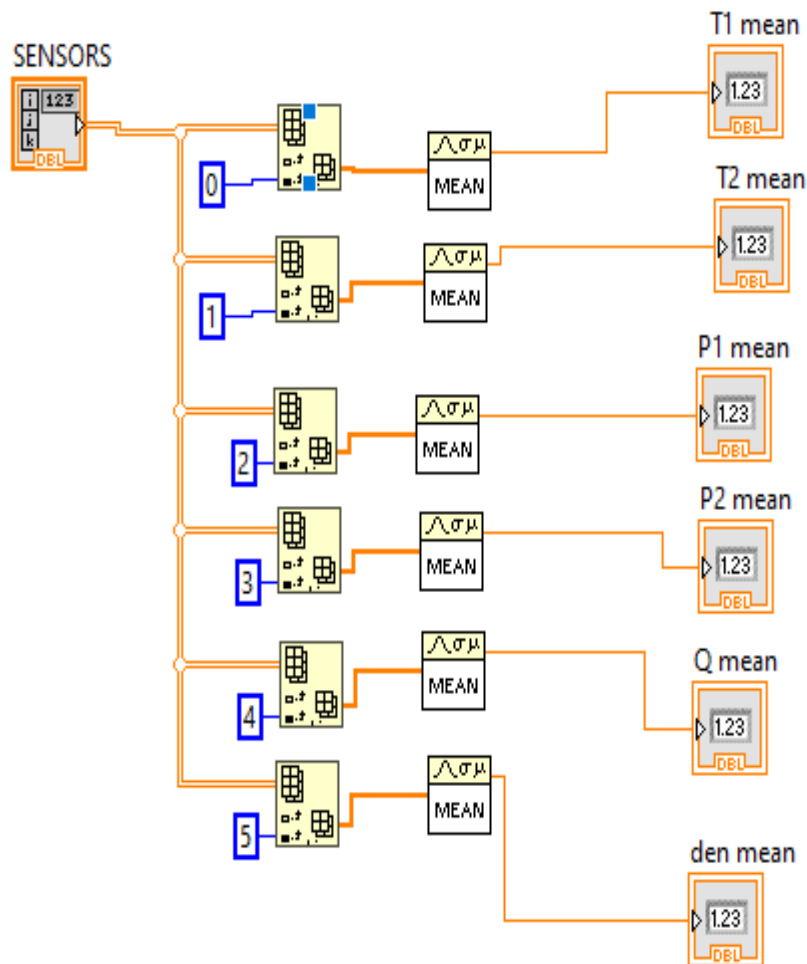
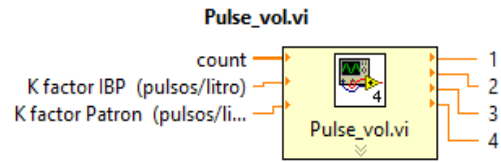


Figura 115. SubVI valores\_medio\_sensores.

Fuente: Elaboración propia.



## Pulse\_vol.



### Outputs

- 1 DBL Pulsos ibp
- 2 DBL Pulsos pat
- 3 DBL Vol ibp
- 4 DBL Vol pat

Este SubVI obtiene los datos de pulsos del ibp, patron y sus volúmenes con su k-factor

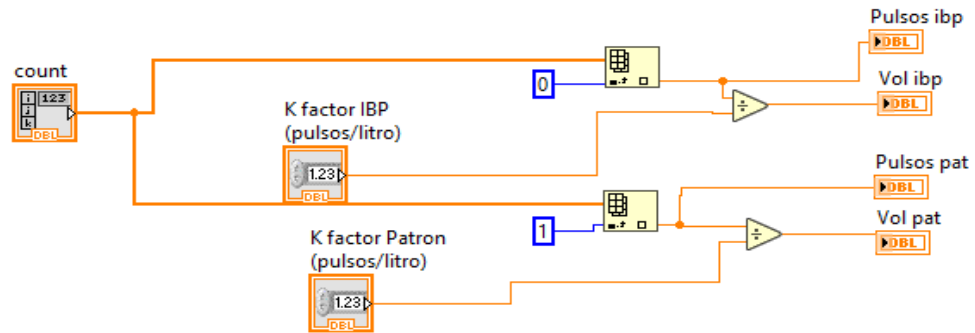
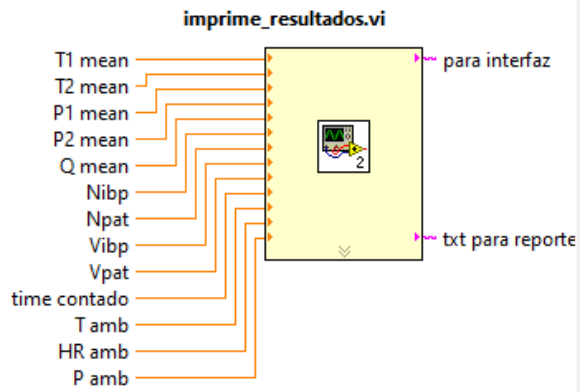


Figura 116. SubVI Pulse\_vol.

Fuente: Elaboración propia.

**imprime\_resultados.**



Este SubVI agrupa los resultados obtenidos en txt para mostrarlos en la pantalla

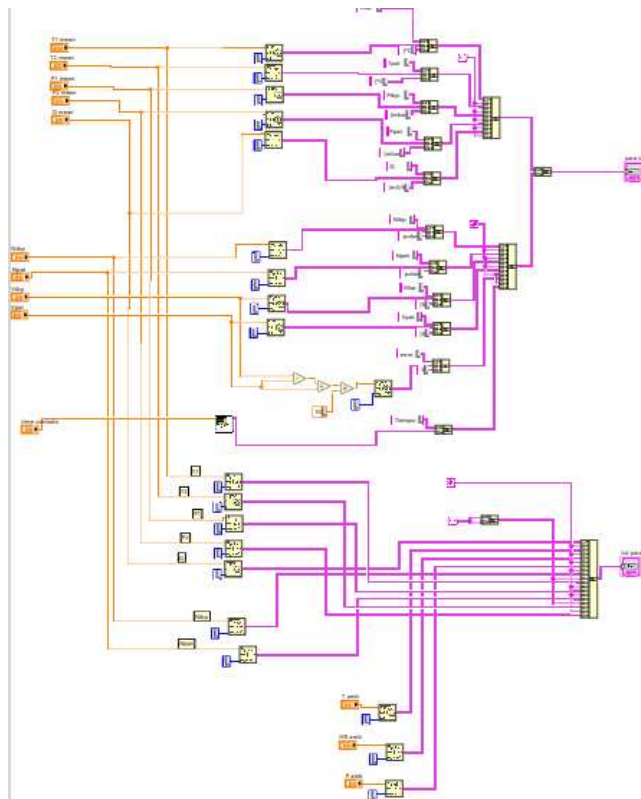
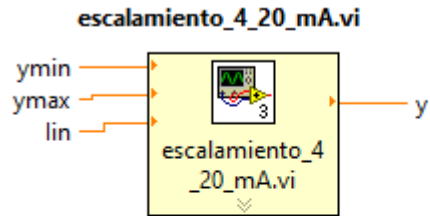


Figura 117. SubVI. *imprime\_resultados.*

Fuente: *Elaboración propia.*

## escalamiento\_4\_20\_mA.



Este SubVI hace el escalamiento de los valores de corriente adquiridos a sus correspondientes valores de temperatura, presión o caudal, según el transmisor que correspondaa. Las entradas son los rangos de medición de los sensores y la corriente medida.

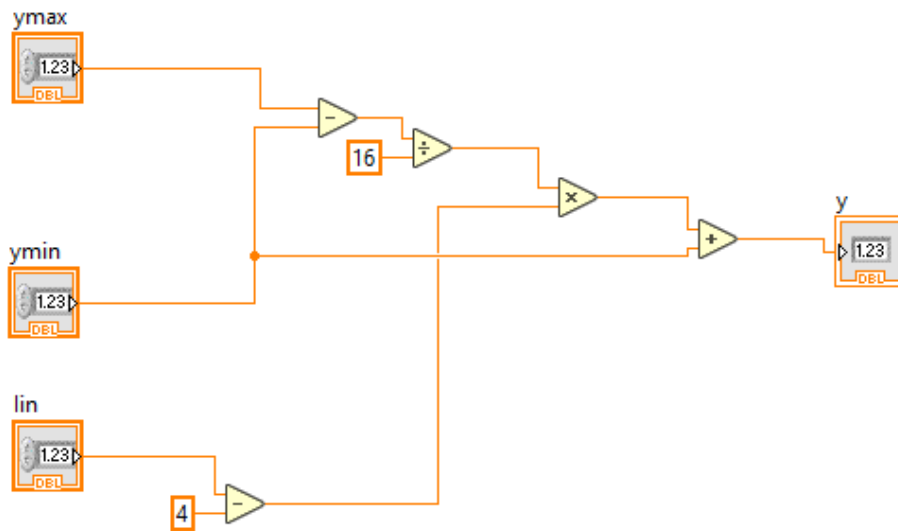
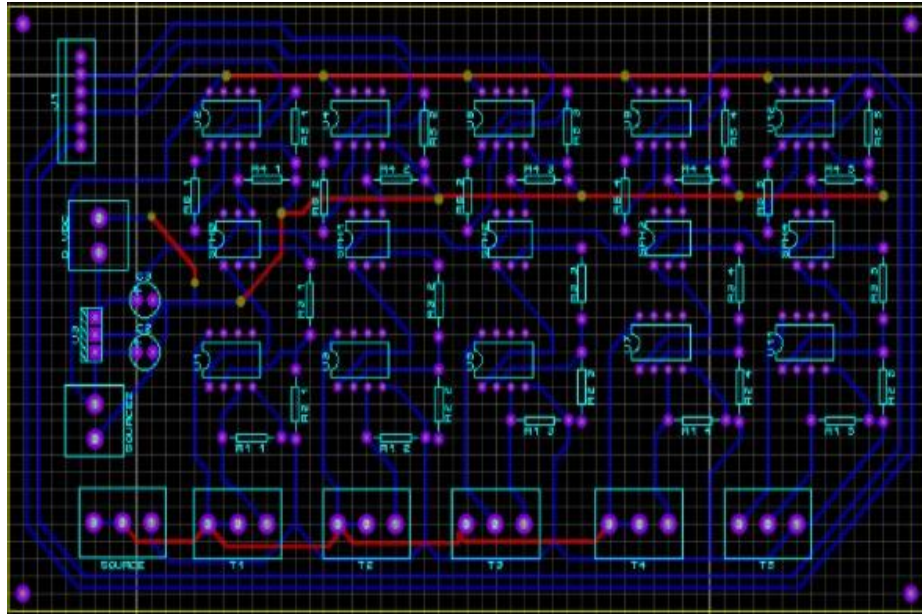


Figura 118. SubVI escalamiento 4\_20\_mA.  
Fuente: Elaboración propia.

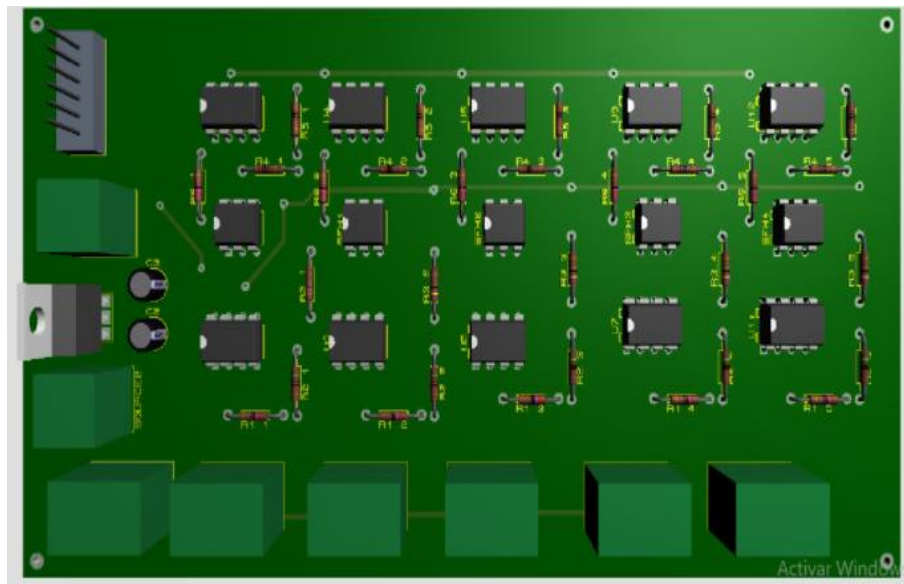
## ***ANEXO H – Tarjetas del Hardware.***

### ***ANEXO H.1. TARJETA 1.***



*Figura 119. Esquemático Tarjeta 1.*

*Fuente: Elaboración propia.*



*Figura 120. Simulación Tarjeta 1.*

*Fuente: Elaboración propia.*

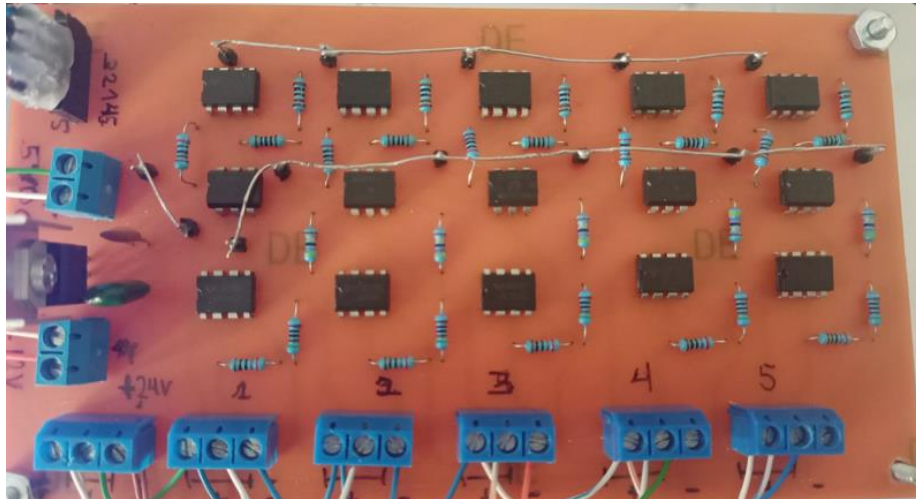


Figura 121. Tarjeta 1.

Fuente: Elaboración propia.

## ANEXO H.2. TARJETA 2.

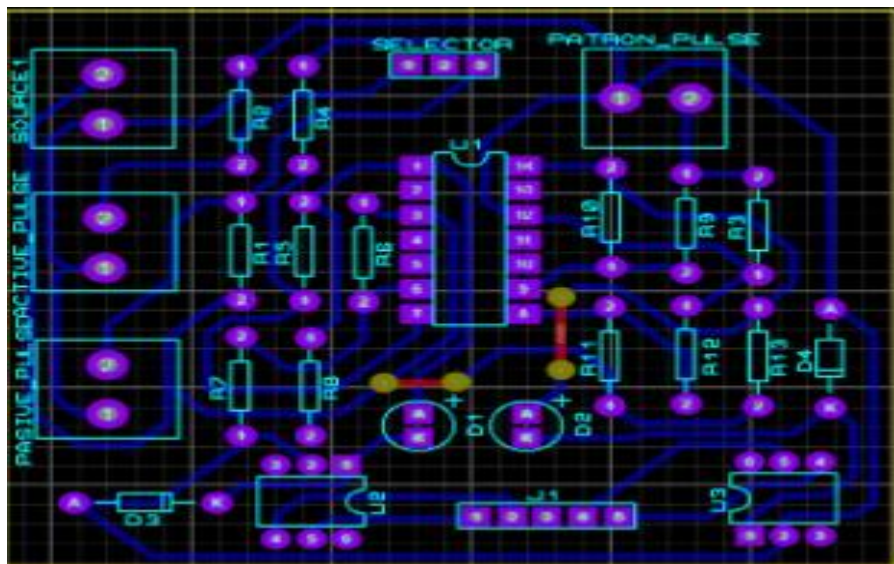


Figura 122. Esquemático Tarjeta 2.

Fuente: Elaboración propia.



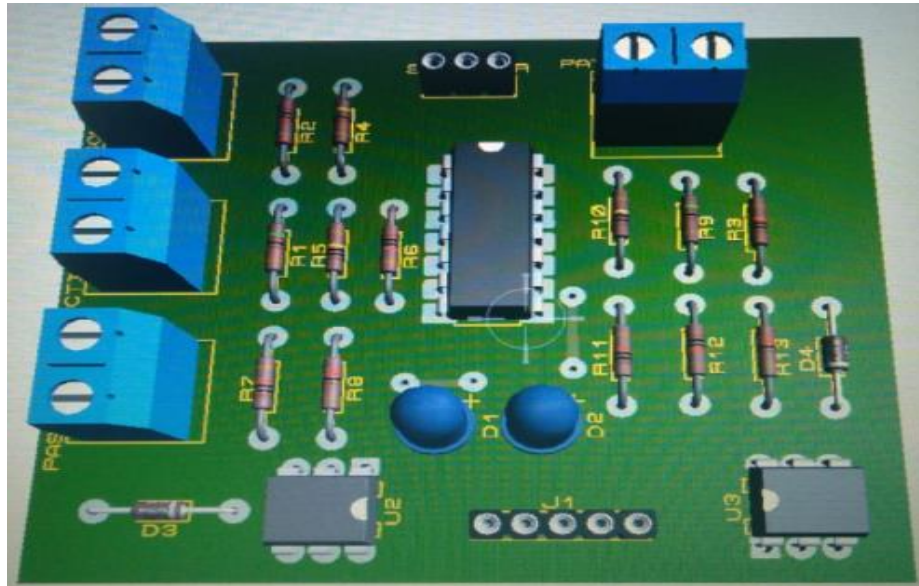


Figura 123. Simulación Tarjeta 2.

Fuente: Elaboración propia

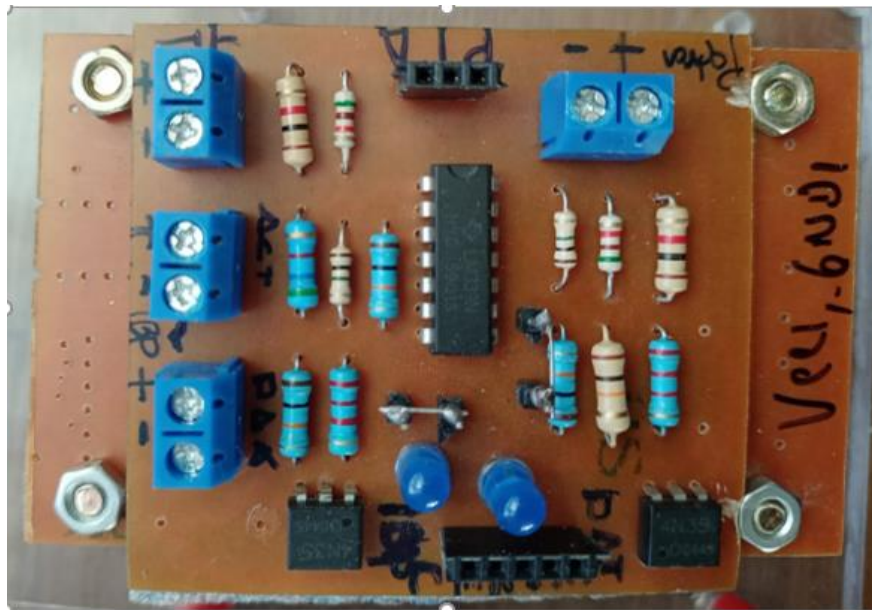


Figura 124. Tarjeta 2.

Fuente: Elaboración propia

**ANEXO H.3. TARJETA 3.**

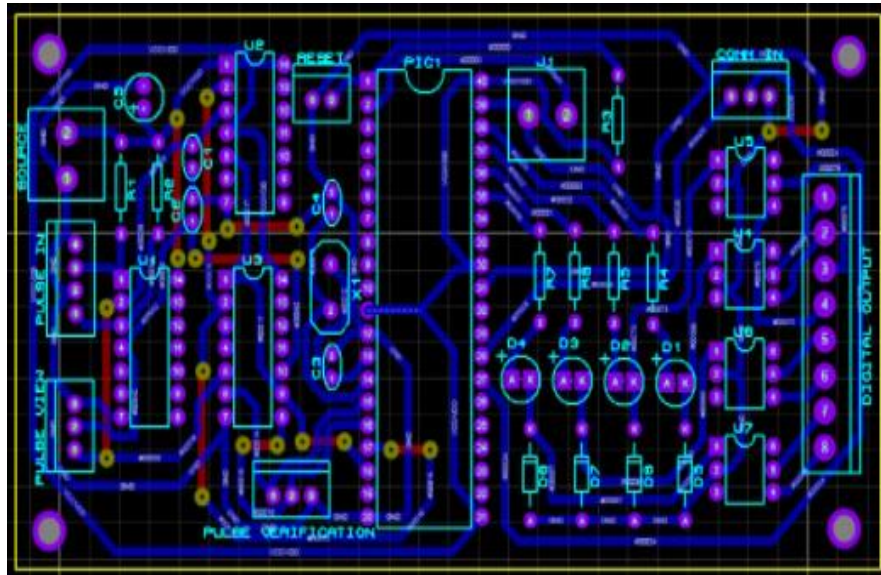


Figura 125. Esquemático Tarjeta .3.

Fuente: Elaboración propia.

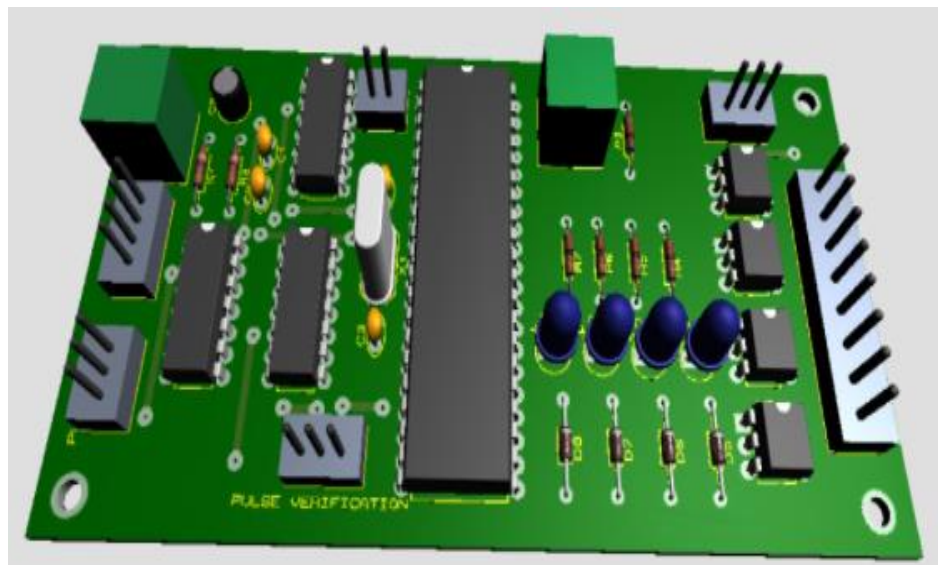


Figura 126. Simulación Tarjeta 3.

Fuente: Elaboración propia

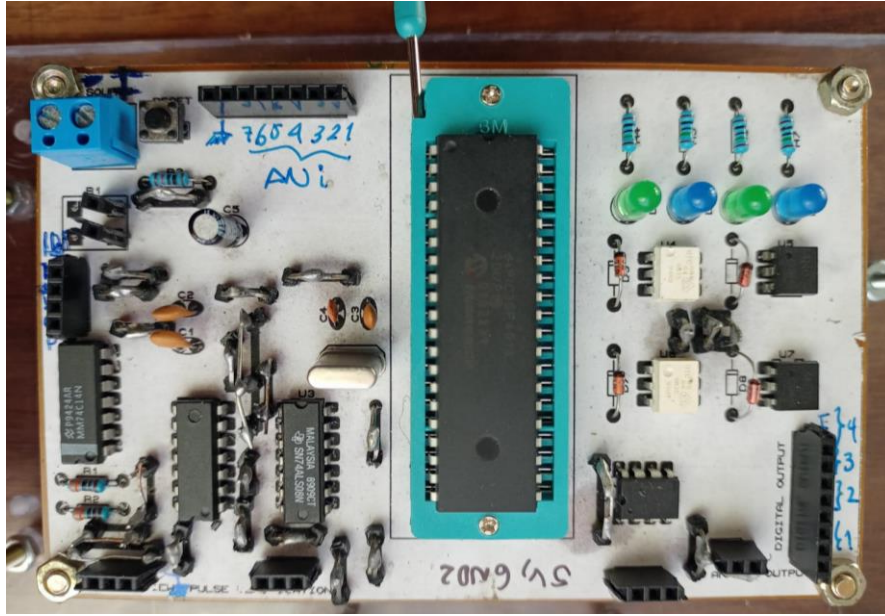


Figura 127. Tarjeta 3.

Fuente: Elaboración propia



## ***ANEXO I – Cartas.***



**Figura 128. Carta de consentimiento METROLAB.**  
**Fuente: METROLAB SRL.**

CORREO ELECTRÓNICO: [bryanjh.bass@gmail.com](mailto:bryanjh.bass@gmail.com)

NRO. DE CELULAR: 67101331