DISEÑO DE UN SISTEMA QUE PERMITA MONITOREAR LA ENERGÍA DE IMPACTO DURANTE LAS PRUEBAS REALIZADAS EN EL PÉNDULO CHARPY DEL LABORATORIO DE MATERIALES DE LA UNIVERSIDAD ANTONIO NARIÑO SEDE BUCARAMANGA

MIGUEL ÁNGEL ROJAS RAMÍREZ
GERSON OMAR ROJAS MURILLO

Universidad Antonio Nariño

Facultad de Ingeniería Electromecánica

Bucaramanga, Colombia

2020

DISEÑO DE UN SISTEMA QUE PERMITA MONITOREAR LA ENERGÍA DE IMPACTO DURANTE LAS PRUEBAS REALIZADAS EN EL PÉNDULO CHARPY DEL LABORATORIO DE MATERIALES DE LA UNIVERSIDAD ANTONIO NARIÑO SEDE BUCARAMANGA

MIGUEL ÁNGEL ROJAS RAMÍREZ
GERSON OMAR ROJAS MURILLO

Trabajo de grado presentado para optar por el título de:

Ingeniero Electromecánico.

Director de proyecto de Grado:

M.Sc. Camilo Leonardo Sandoval.

Universidad Antonio Nariño

Facultad de Ingeniería Electromecánica

Bucaramanga, Colombia

2020

NOTA DE ACEPTACIÓN

_____

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Agradecimientos

Dedicamos este trabajo de grado primeramente a Dios quien es aquel que nos ha dado la vida. A nuestros padres y esposas quienes nos apoyaron incondicionalmente en este camino y nos dieron ánimo en momentos de dificultad y desfallecimiento. A la Universidad Antonio Nariño la cual nos abrió sus puertas para formarnos profesionalmente. A nuestros docentes por brindarnos los diferentes métodos de enseñanza y guiarnos en este camino de ser profesionales, especialmente al M.Sc Camilo Sandoval por su interés, confianza y apoyo al realizar este trabajo de grado.

Contenido

Pág.

Resumen 13

Introducción. 15

Planteamiento del problema 16

Objetivos del proyecto 18

Objetivo general. 18

Objetivos específicos. 18

1. Marco conceptual. 19

1.1. Clasificación de las fracturas. 19

1.2. Tenacidad. 21

1.3. Norma ASTM E23. 21

1.4. Energía de impacto. 22

1.5. Sensores de posición. 22

1.5.1. Acelerómetros. 24

1.5.2. Giroscopio. 24

1.5.3. Potenciómetro angular. 25

1.6. Ensayo Charpy Instrumentado. 26

1.6.1. Software LabVIEW para Charpy instrumentado. 26

1.6.2. Comunicación serie en LabVIEW. 27

2. Revisión de características y modificaciones a realizar en el péndulo CHARPY de la UAN. 29
2.1. Operación actual del péndulo Charpy de la UAN. 29
2.1.1. Control realizado por el software LabVIEW. 29
2.1.2. Porque reemplazar los bloques LINX. 30
2.1.3. Lógica secuencial del martillo utilizando LINX. 31
2.1.4. Entradas y salidas digitales. 32
2.1.5. Circuitos de potencia y lógica de control cableada. 34
3. Algoritmos para Arduino Mega. 37
3.1. Características del nuevo algoritmo. 37
3.2. Algoritmo y programación para operación local. 38
3.3. Algoritmo y programación de nueva visualización local en LCD 40
3.4. Algoritmo y programación de nueva comunicación con LabVIEW. 42
3.5. Panel de control local 45
4. Alternativa para determinar los grados de inclinación. 47
4.1. Periféricos del microcontrolador Atmel 2560 47
4.1.1. Acelerómetro MMA845 47
4.1.2. Giroscopio POLOLU 2739 48
4.1.3. Potenciómetro. 48
4.2. Matrices para seleccionar sensor de posición angular. 49
5. Caracterización del sensor de posición y algoritmos para el microcontrolador ATMEL 2560. 52
5.1. Procedimiento para determinar ecuaciones. 53

5.1.1. Caracterizacion del ángulo 1. 54

5.1.2. Caracterización del ángulo 2. 55

5.2. Algoritmo para determinar ángulos. 56

6. Nuevo algoritmo para el péndulo basado en NI-VISA. 58

6.1. Modificaciones realizadas en el panel frontal. 58

6.2. Nuevo diagrama de bloques. 59

6.2.1. Recepción de datos en LabVIEW con NI-VISA 59

6.2.2. Trasmisión de datos en LabVIEW con NI-VISA. 62

7. Resultados y análisis. 64

7.1. Efectos en el circuito de control y potencia durante la operación desde el panel de control local. 64

7.1.1. Pruebas para subir y bajar martillo. 64

7.1.2. Anclaje y desanclaje del martillo, acciones en la lógica de control cableado 67

7.1.3. Visualización local y en LabVIEW de sensores de posición. 69

7.2. Simulación de secuencia de ascenso y anclaje de martillo. 71

7.3. Estimación de ángulos y cálculo de energía de impacto. 74

8. Conclusiones. 77

Bibliografía. 79

Anexo I. Código programado en el Arduino Mega. 81

Listado de figuras

Pág.

Figura 1 Indicador de ángulo inicial y final del péndulo Charpy 17

Figura 2 Fractura de acero 1040. 20

Figura 3 Tipos de entalla Charpy. 20

Figura 4 Esquema péndulo Charpy 22

Figura 5 Acelerómetro 24

Figura 6 Vectores que actúan sobre un giroscopio. 25

Figura 7 Potenciómetro. 26

Figura 8 Comunicación serie en LabVIEW. 27

Figura 9 Tablero de control del péndulo Charpy. 30

Figura 10 Diagrama de bloques utilizando bloques LINX. 31

Figura 11 Diagrama de flujo en operación automática. 32

Figura 12 Bloques utilizados actualmente por péndulo Charpy. 34

Figura 13 Circuito para invertir el sentido de giro del motor que eleva el martillo. 35

Figura 14 Circuito para invertir el giro del motor de anclaje del martillo. 35

Figura 15 Circuito de control cableado 36

Figura 16 Nuevas características del algoritmo del Microcontrolador. 37

Figura 17 Algoritmo para operación local. 38

Figura 18 Elementos en Proteus para emular entradas y salidas digitales. 39

Figura 19 Pantalla LCD gráfica 64x128 obtenida de Proteus 40

Figura 20 Algoritmo del Microcontrolador para controlar pantalla LCD. 41

Figura 21 Conexión entre Arduino Mega y pantalla LCD. 42

Figura 22 VSPE (Virtual Serial Port Emulator) 42

Figura 23 VSPE, seleccionado COM1 43

Figura 24 Configuración de puerto serial en Proteus 43

Figura 25 Algoritmo del Microcontrolador para comunicación serial. 44

Figura 26 Conexión entre Arduino Mega y COMPIM 45

Figura 27 Alternativa de panel a ubicar en el péndulo. 46

Figura 28 Acelerómetro 48

Figura 29 Giroscopio 48

Figura 30 Potenciómetro. 49

Figura 31 Facilidad para conectar el potenciómetro al eje del péndulo. 52

Figura 32 Ángulos para caracterización. 54

Figura 33 Relación voltaje y grados de inclinación del ángulo 1. 55

Figura 34 Relación voltaje y grados de inclinación del ángulo 2. 56

Figura 35 Algoritmo para determinar ángulos de inclinación. 57

Figura 36 (a) Panel frontal actual y (b) Panel frontal con modificaciones 58

Figura 37 Algoritmo de recepción de entradas digitales. 60

Figura 38 VISA Read e inicio de secuencia. 61

Figura 39 Información separada por segmentos para entradas digitales. 61

Figura 40 Información separada por segmentos para ángulos y energía de impacto. 62

Figura 41 Enviando información por puerto serial. 63

Figura 42 Posición de muletilla para subir martillo 64

Figura 43 R5 sube martillo. 65

Figura 44 Relés y contactores activados para subir martillo 66

Figura 45 Relés y contactores activados para bajar martillo o soltar guaya. 67

Figura 46 R7 ancla el martillo. 68

Figura 47 Relés y contactores activados para anclar y soltar martillo 69

Figura 48 Visualización de sensores de posición en LCD 70

Figura 49 Visualización de sensores de posición en LabVIEW 70

Figura 50 Visualización del estado de los pulsadores y finales de carrera en PROTEUS y Visualización del estado de la operación en LabVIEW 71

Figura 51 Simulación de subiendo martillo en Proteus. 72

Figura 52 Salida digital que emula el anclaje del martillo 73

Figura 53 Simulación de martillo anclado. 73

Figura 54 Simulación de impacto. 74

Figura 55 Martillo en posición vertical. 75

Figura 56 Visualización en LCD de ángulos y energía de impacto. 75

Figura 57 Visualización en LabVIEW de ángulos y energía de impacto. 76

Listado de tablas.

Pág.

Tabla 1 Sensores de diversas magnitudes físicas. 23

Tabla 2 Salidas digitales de la placa Arduino Mega. 33

Tabla 3 Entradas digitales de la placa Arduino Mega. 34

Tabla 4 Relación entre estado de entrada digital y visualización en LCD. 41

Tabla 5 Criterios de selección. 50

Tabla 6 Interfaz versus sensores. 50

Tabla 7 Voltaje versus sensores. 51

Tabla 8 Precio versus sensores. 51

Tabla 9 Matriz final para seleccionar el sensor de posición angular 51

Tabla 10 Comandos enviados por los diferentes botones. 63

Listado de ecuaciones.

Pág.

Ecuación 1 Energía de impacto..... 22

Ecuación 2 Factor de ponderación. 49

Ecuación 3 Peso de la opción..... 50

Ecuación 4 Relación voltaje y grados de inclinación 1. 55

Ecuación 5 Relación voltaje y grados de inclinación 2. 56

Resumen

Título: Diseño de un sistema que permita monitorear la energía de impacto durante las pruebas realizadas en el péndulo Charpy del laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga.

Autor(es): Miguel Ángel Rojas Ramírez, Gerson Omar Rojas Murillo

Descripción.

El presente trabajo de grado busca desarrollar un sistema de monitoreo para facilitar la adquisición de datos obtenidos en la medición de la energía de impacto durante la prueba realizada con el péndulo Charpy disponible en el laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga, esto será desarrollado a través de un algoritmo en LabVIEW que permitirá guardar las diferentes mediciones y además visualizar en un computador los valores obtenidos en la prueba, en la actualidad no se puede realizar la medición exacta de dicha variable porque la escala de los grados de visualización de desplazamiento del martillo es muy amplia y el cálculo se realiza de manera manual.

Palabras claves: Energía de impacto, LabVIEW, Microcontrolador, Péndulo Charpy.

Abstract

Title: Design of a system that allows monitoring the impact energy during the tests carried out on the Charpy pendulum of the materials laboratory of the Antonio Nariño university, Bucaramanga headquarters.

Author: Miguel Ángel Rojas Ramírez, Gerson Omar Rojas Murillo

Description.

The present degree work seeks to develop a monitoring system to facilitate the acquisition of data obtained in the measurement of impact energy during the test carried out with the Charpy pendulum available in the materials laboratory of the Universidad Antonio Nariño, headquarters Bucaramanga, this will be developed through an algorithm in LabVIEW that will allow saving the different measurements and also displaying the values obtained in the test on a computer. Currently, the exact measurement of this variable cannot be performed because the scale of the degrees of displacement visualization The hammer is very wide and the calculation is done manually.

Keywords: Impact Power, LabVIEW, Microcontroller, Charpy Pendulum.

Introducción.

Este trabajo cuyo alcance es diseñar un sistema que permita monitorear la energía de impacto durante las pruebas realizadas en el péndulo Charpy del laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga, utilizando un sensor de posicion con el cual se determinaran los grados de inclinacion del martillo utilizando un microcontrolador ATMEGA 2560 y visualizarla en un PC empleando LabVIEW, tiene como objetivos identificar la mejor alternativa para determinar los grados de inclinación, establecer una ecuación que permita relacionar los grados de inclinacion y desarrollar un algoritmo para realizar la operación del péndulo utilizando comandos NI-VISA, es por esta razón que se plantea la metodología mencionada

Fase 1. Seleccionar el sensor de posición. Inicialmente es necesario definir cuáles son las características más importantes del sensor que será utilizado para medir el ángulo de inclinación, incluyendo la facilidad para la instalación y el acople de dicha señal al sistema de adquisición de datos. Después de esto se entrarán a comparar las diferentes opciones para seleccionar la mejor alternativa.

Fase 2. Caracterización del sensor seleccionado. Se ejecutarán pruebas para realizar la puesta a punto del sistema de monitoreo desarrollado.

Fase 3. Desarrollar los algoritmos para la placa Arduino Mega 2560 y LabVIEW. Desde el IDE de Arduino se realizará el código que cumple las condiciones previamente establecidas en un algoritmo que permita leer el sensor de posición angular y calcular la energía de impacto.

Planteamiento del problema

En el año 2016 se realiza la construcción de un péndulo Charpy que permite determinar la energía de impacto de un material teniendo en cuenta la ley de conservación de energía [1], esto con el valor de aportar al laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga con el ánimo que los estudiantes realicen pruebas y afiancen sus conocimientos teóricos en la práctica. Sin embargo, dadas las dimensiones y peso del martillo, la manipulación de este equipo no era segura pues el anclaje se debía realizar de manera manual, es por eso que durante el primer semestre académico de 2019 se realiza una primera modificación en la operación de esta máquina que dio como resultado la implantación de un sistema controlado desde LabVIEW el cual actúa sobre unos motores eléctricos que automáticamente elevan el martillo y lo anclan [2]. Sin embargo, una condición por mejorar es lograr monitorear la energía de impacto del material durante la prueba, ya que en la actualidad esta máquina cuenta con un indicador del ángulo inicial y final (ver figura 1) cuya división de escala es de baja resolución lo cual aumenta el error al momento de realizar el cálculo de la energía de impacto por parte de los estudiantes. Por esta razón se plantea la pregunta de investigación ¿Cómo disminuir el error en las mediciones del ángulo inicial y final para determinar la energía de impacto del material en la prueba con el péndulo Charpy del laboratorio de materiales de la Universidad Antonio Nariño? Este presente trabajo de grado dará solución ya que conseguirá monitorear esta actividad mediante la instalación de una serie de sensores y modificaciones en los algoritmos en LabVIEW y de esta forma se asegurará una lectura fiable.

Figura 1 Indicador de ángulo inicial y final del péndulo Charpy
Fuente: Autores.

Objetivos del proyecto

Objetivo general.

Diseñar un sistema que permita monitorear la energía de impacto durante las pruebas realizadas en el péndulo Charpy del laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga.

Objetivos específicos.

- Identificar cual es la mejor alternativa para determinar los grados de inclinación del martillo del péndulo Charpy, utilizando matrices de selección donde se comparen diferentes sensores de posición.
- Establecer una ecuación que permita relacionar los ángulos inicial y final de la prueba simulada, a traves de un algoritmo desarrollado en una placa Arduino.
- Desarrollar un algoritmo para realizar la operación del péndulo Charpy utilizando comandos NI-VISA de LabVIEW simulando las señales de proceso a través del software Proteus.

1. Marco conceptual.

1.1. Clasificación de las fracturas.

Para iniciar el desarrollo de este trabajo es necesario hacer claridad sobre la clasificación de las fracturas de un cuerpo sólido en dos o más partes bajo la acción de esfuerzos. El proceso de fractura puede ser considerado por dos componentes, iniciación de la grieta y propagación de la grieta [3]. De acuerdo a la literatura se puede decir que existen cuatro maneras de fractura:

- Sobrecarga.
- Impacto.
- Fatiga.
- Fractura asistida por el medio.

Y los criterios comúnmente utilizados para clasificar las fracturas son basados en los siguientes aspectos:

- Velocidad de propagación de la grieta.
- Modo cristalográfico de la fractura.
- Aspecto macroscópico de la fractura.
- Deformación plástica macroscópica.
- Deformación plástica microscópica.
- Micromecanismo de fractura.

Según lo establece [4] las fracturas se pueden clasificar en dos categorías generales, fractura dúctil y fractura frágil. Una fractura dúctil se caracteriza por una deformación plástica apreciable antes y durante la propagación de la grieta. La fractura frágil en metales se caracteriza por una rápida velocidad de propagación de grieta, sin grandes deformaciones y muy poca

microdeformación. El rasgo característico de la fractura por clivaje son facetas planas como las indicadas en la figura 2.

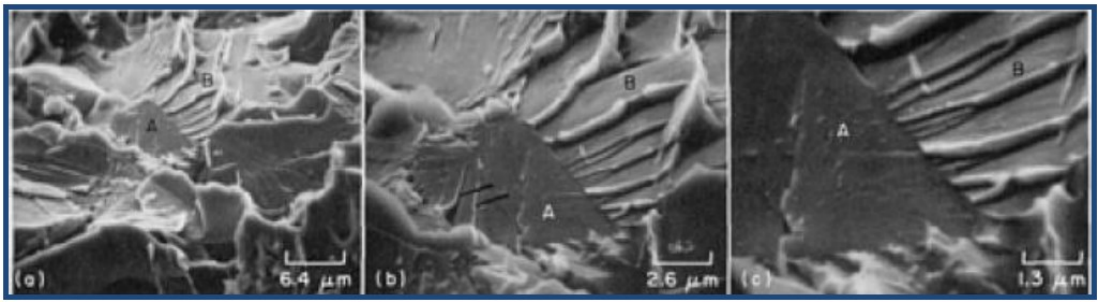

Figura 2 Fractura de acero 1040.
Fuente: DIETER, G.E. Mechanical Metallurgy, SI Metric Edition, McGraw-Hill, 1988, p. 265.

Las pruebas de impacto se realizan según normas internacionales en las cuales se detallan las dimensiones de las probetas empleadas, así como la forma de reporte de los resultados, las muestras que se utilizan en el ensayo Charpy tienen forma de paralelepípedo y se pueden utilizar tres tipos de entalla, tal como se muestra en la figura 3 [5].

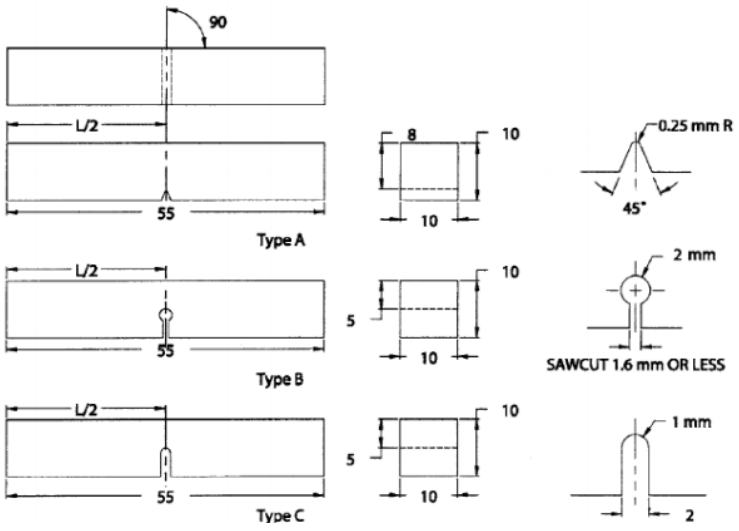

Figura 3 Tipos de entalla Charpy.
Fuente: ASTM E23- 07ae1. Standad Test Methods for Notched BarnImpact Testing of Metallic Materials, 2007.

1.2. Tenacidad.

Los materiales utilizados en la industria pasan por una serie de pruebas y requisitos para poder ser seleccionados y cumplir de manera óptima su función. Una de las propiedades puestas a prueba en los materiales es la tenacidad o la resistencia a la fractura, definida por [6] como la capacidad que tiene un material de absorber energía durante la falla o ruptura. Cada material tiene una tenacidad diferente y varía dependiendo de las imperfecciones que la muestra contenga. Esta capacidad de absorción de energía varía, depende principalmente de 6 factores [7]. Tamaño de los defectos del material: A mayor tamaño de los defectos, menor tenacidad tiene el material. La capacidad de deformación del material: Al incrementar la resistencia del material, por lo general se reduce la ductilidad obteniendo una menor tenacidad a la fractura. El grosor y la rigidez del material: A mayor grosor y más rigidez se obtiene una menor tenacidad a la fractura. La velocidad de aplicación de la carga: A mayor velocidad de impacto se reduce el nivel de tenacidad del material. La temperatura de ensayo: A mayor temperatura existe una menor tenacidad del material. La microestructura del material: Una estructura de granos pequeños presenta una mejora a la tenacidad a la fractura, en comparación de un material cuya estructura tenga gran cantidad de defectos puntuales y dislocaciones.

1.3. Norma ASTM E23.

Esta en esencia indica las características, consideraciones y recomendaciones que el péndulo, la muestra y el ensayo deben cumplir para que los resultados sean válidos. Además, incida el tamaño, los acabados de la muesca que debe tener la probeta [8]. En el ítem A. 2.3.5.2 se indican las ecuaciones que permiten determinar la energía de impacto y para ello es necesario tener en cuenta altura inicial y final, pero del martillo, longitud de la barra, ángulos inicial y final [9].

1.4. Energía de impacto.

La energía absorbida $\boldsymbol{E_a}$ por el material utilizado en la prueba de impacto del péndulo Charpy, se determina a través de la diferencia de energía potencial del péndulo antes y después del impacto; para ello se utiliza la ecuación 1 siendo $\boldsymbol{W}$ el peso del martillo y $\boldsymbol{L}$ la longitud tal como se observa en la figura 4 [10].

Ecuación 1 Energía de impacto.

$$E_a = W * L(\cos(\beta) - \cos(\theta))$$

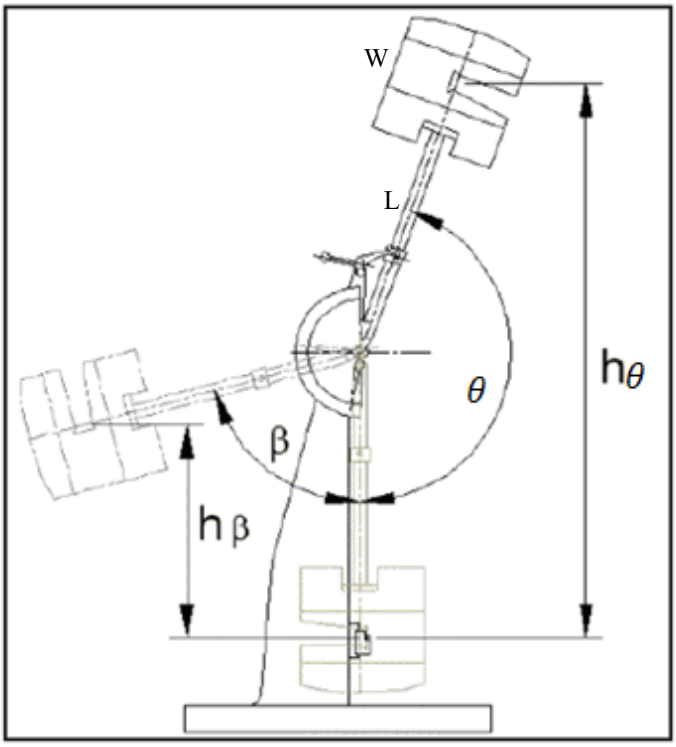

Figura 4 Esquema péndulo Charpy
Fuente: Viracocha, C. (2014).

1.5. Sensores de posición.

De acuerdo a lo indicado por (Domingo, 2003, p. 38) los sensores son elementos electrónicos que permiten conocer la variable de interés de algún proceso, y se pueden clasificar como:

- Pasivos: Modifican la impedancia con determinadas condiciones físicas o químicas y provocan variaciones en la tensión o corriente del circuito en el cual operan.
- Activos: Son generadores eléctricos que no necesitan de fuentes de alimentación excepto la de la fuente de los circuitos de amplificación.

Según el mismo autor, estos sensores también se pueden clasificar según el tipo de señal de salida como se menciona:

- Análogos: La salida toma un valor de tensión o corriente variable de forma continua dentro del campo de medida.
- Digitales: Salida codificada en forma de pulsos.
- Todo o nada: Indican únicamente cuando una variable detecta un límite de operación.

Adicional a esto también se pueden clasificar según la magnitud física a detectar y en la tabla 1 se presenta una variedad de sensores utilizados en la industria.

Magnitud	Transductor	Característica
Posición angular	Potenciómetro	Analógico
	Encoder	Digital
	Acelerómetro, giroscopio	Analógico.
Pequeños desplazamientos o deformaciones	Galga	Analógico.
	Transformador diferencial	Analógico.
Velocidad lineal o angular	Dinamo	Analógico.
	Encoder	Digital
	Acelerómetro	Analógico.

Tabla 1 Sensores de diversas magnitudes físicas.
Fuente: Viracocha, C. (2014).

Para obtener la posición angular del péndulo del martillo es posible mediante sensores como los mencionados a continuación.

1.5.1. Acelerómetros.

Miden fuerzas que son aplicadas a una masa en movimiento y pueden ser capacitivos o mecánicos. Los acelerómetros mecánicos emplean una masa conocida que se encuentra suspendida dentro de un cilindro o entorno que limita su movimiento lineal [11]. Un acelerómetro capacitivo trabajan bajo el mismo principio que los acelerómetros mecánicos, con la diferencia que estos miden el desplazamiento de una masa (posición) utilizando un condensador diferencial, que consta de dos condensadores variables colocados de tal manera que su capacitancia cambie del mismo modo pero con signo contrario, adicional a esto se agrega un circuito acondicionador de la señal de salida, para obtener una medida lineal (ver figura 5).

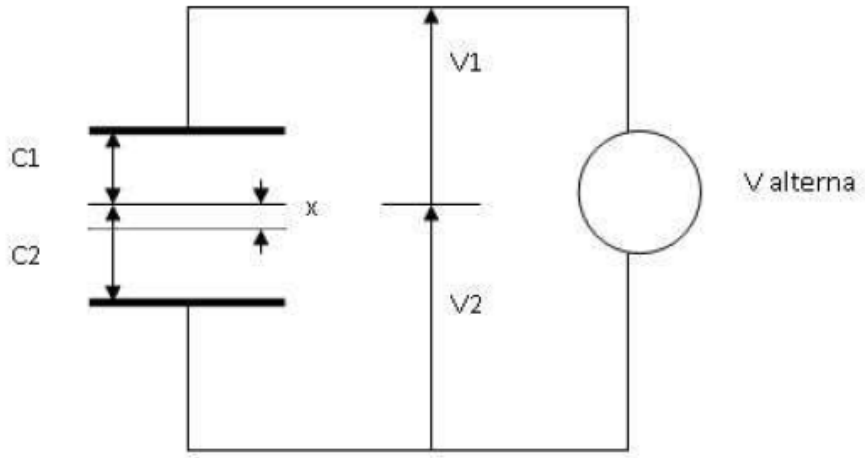

Figura 5 Acelerómetro
Fuente: (Moore, 2007).

1.5.2. Giroscopio.

Es un mecanismo simétrico de rotación que gira alrededor de su eje de simetría y que permite determinar su orientación o variación angular a través de un comportamiento intuitivo (ver figura 6) [12].

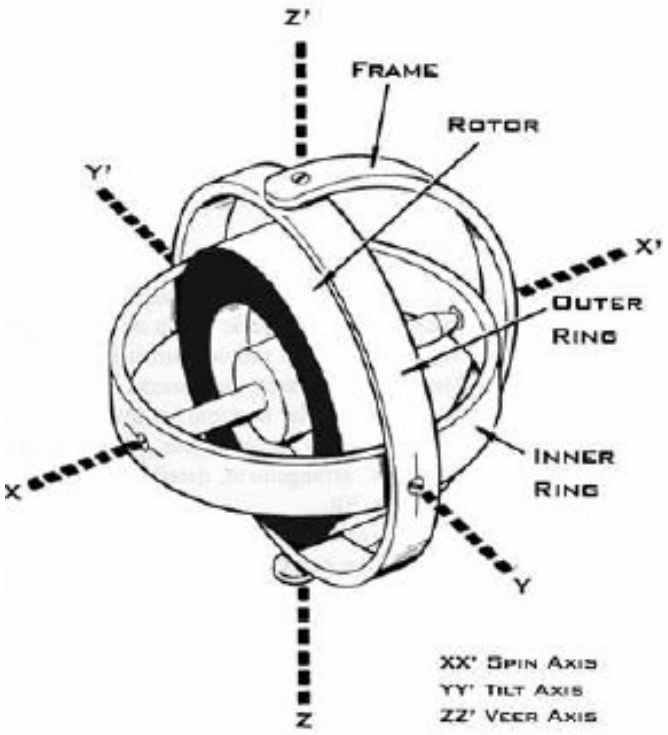

Figura 6 Vectores que actúan sobre un giroscopio.
Fuente: (Guerrero, 2013)

1.5.3. Potenciómetro angular.

Los potenciómetros angulares son dispositivos que convierten la posición angular en una salida analógica, consta de una bobina sobre un material conductor, distribuida a lo largo de un arco que funge como soporte y un apuntador único a un eje de salida que se desplaza sobre dicho conductor (ver figura 7). El movimiento angular provoca cambios de resistencia de forma que, al aplicar el potencial constante en los extremos, una tensión proporcional al ángulo de giro del origen podrá ser medida en la salida del potenciómetro [13].

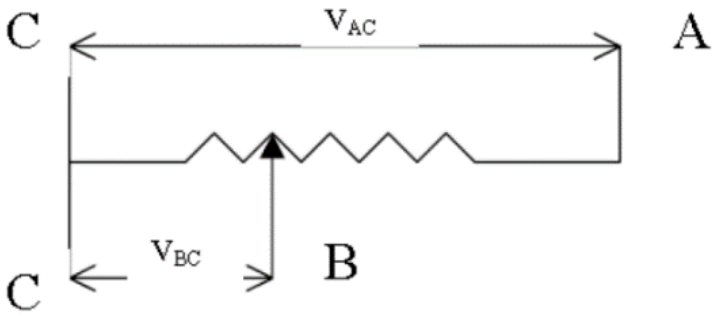

Figura 7 Potenciómetro.
Fuente: Autores.

1.6. Ensayo Charpy Instrumentado.

Este ensayo fue concebido pensando en tener información más precisa sobre el proceso de fractura dinámica que ofrece el ensayo de impacto convencional, por medio de dispositivos electrónicos adaptados al martillo de impacto que transforman la energía mecánica en energía eléctrica que es transmitida a un sistema de adquisición de datos para posteriormente ser llevada a un software que procesa la información [14]. Para el caso de este trabajo ha pensado en adaptar el péndulo Charpy del laboratorio de materiales de la UAN utilizando el software LabVIEW.

1.6.1. Software LabVIEW para Charpy instrumentado.

Como ya se ha mencionado en el ítem 1.5 de este documento, existen distintos tipos de sensores que permiten relacionar la posición con una variable eléctrica que puede ser tomada por un sistema de adquisición de datos, sin embargo, ahora se entra a detallar como se realizara la comunicación con el software. Las comunicaciones que se pueden establecer entre varios dispositivos se pueden clasificar de acuerdo a muchos parámetros desde el punto de vista de forma de transición, pero para el caso de este trabajo solo se entrara a evaluar la comunicación serial, la cual es una trama de bits transmitidos desde el start al stop y cuyas principales características son:

- La velocidad, medida en baudios.
- Numero de bits de datos.
- Tipo de paridad.

1.6.2. Comunicación serie en LabVIEW.

Las versiones actuales de LabVIEW incluyen unos VI de la librería VISA como la indicada en la figura 8, el bloque llamado Visa Configure Serial Port el cual permite configurar todos los parámetros de la comunicación serial.

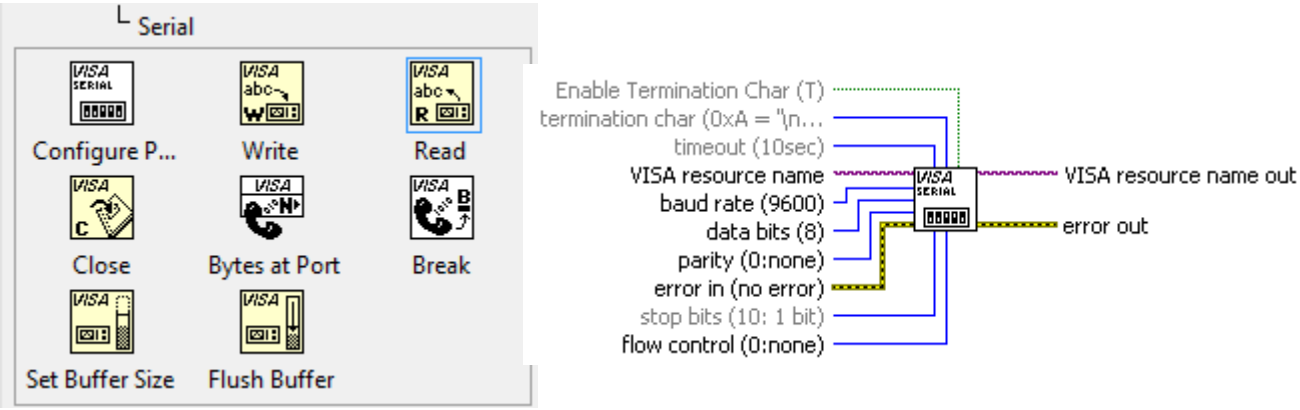

Figura 8 Comunicación serie en LabVIEW.
Fuente: Autores.

Los bloques utilizados para realizar la comunicación serial utilizando LabVIEW y que se observan en la figura 8 se describen a continuación:

- Visa Write, se encarga de escribir datos desde el buffer hasta el otro dispositivo.
- Visa Read, es el contrario al anterior, a este es necesario indicarle el número de bytes que debe leer el buffer, para darle un valor a este parámetro.

Como se mencionó en los objetivos se hará uso de NI-VISA que es la implementación de National Instrument de este estándar, puede establecer comunicaciones a través de GPIB, serie PXI, o Ethernet. VISA es una API o librería desarrollada por varios fabricantes de equipos que

proporcionan un estándar de software para las operaciones de lectura y escritura. Mediante NI VISA también se puede acceder a dispositivos de otros fabricantes.

2. Revisión de características y modificaciones a realizar en el péndulo CHARPY de la UAN.

Para lograr operar el péndulo Charpy utilizando los bloques de comunicación NI-VISA de LabVIEW, es necesario contar con la información necesaria y pertinente que lleven a un diseño y simulación correcta y esto será lo analizado en este capítulo, todo esto en busca de tener un péndulo instrumentado.

2.1. Operación actual del péndulo Charpy de la UAN.

Lo primero que se entra a evaluar es como opera la máquina Charpy después de la última actualización realizada en el año 2019. De esta revisión es posible identificar los aspectos de mayor relevancia:

- El control es realizado por el software LabVIEW.
- Las salidas digitales del microcontrolador son acopladas a una tarjeta de relés de estado sólido lo cual aísla eléctricamente el control con la potencia.

2.1.1. Control realizado por el software LabVIEW.

De esta revisión lo que se pretende es identificar cuáles son los mecanismos de programación y hardware utilizados para realizar la visualización de las variables del proceso de tal forma que se logren reemplazar los VI de LINX utilizados por bloques NI-Visa. Tal como lo indica el documento de (Manosalva & Picón, 2019). De esta revisión se encuentra que el Hardware utilizado para operar los motores está basado en la Placa Arduino Mega (ver figura 9), la programación de esta placa se basa en la herramienta LINX que permite controlar las entradas y salidas digitales y monitorear las señales analógicas.

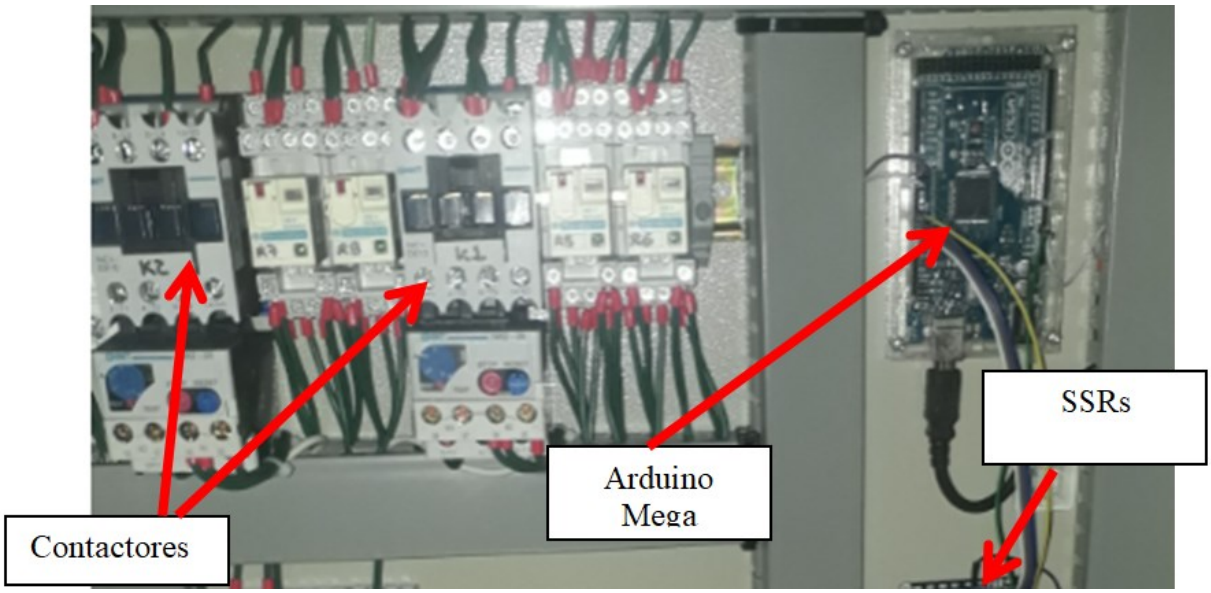

Figura 9 Tablero de control del péndulo Charpy.
Fuente: (Manosalva & Picón, 2019, p. 58).

2.1.2. Porque reemplazar los bloques LINX.

En la figura 10 se observa un fragmento de la programación realizada en el diagrama de bloques de LabVIEW del sistema actual de control del péndulo Charpy. La mejora que se plantea es utilizar bloques NI-Visa de tal forma que se logren realizar las actividades de control directamente desde la placa Arduino Mega y adicional a ello agregar un algoritmo que permita realizar el cálculo de la energía de impacto. Las razones por las cuales se plantea modificar el algoritmo de LabVIEW son las siguientes:

- El control dependerá exclusivamente de los bloques LINX, de esta forma se crea una barrera para operar el péndulo Charpy si no existirá un computador disponible.
- Los bloques LINX tienen limitaciones en cuanto al uso de sensores de posición.
- Utilizando bloques LINX no es posible realizar simulaciones que permitan identificar si es posible determinar la energía de impacto utilizando un sensor de posición angular.

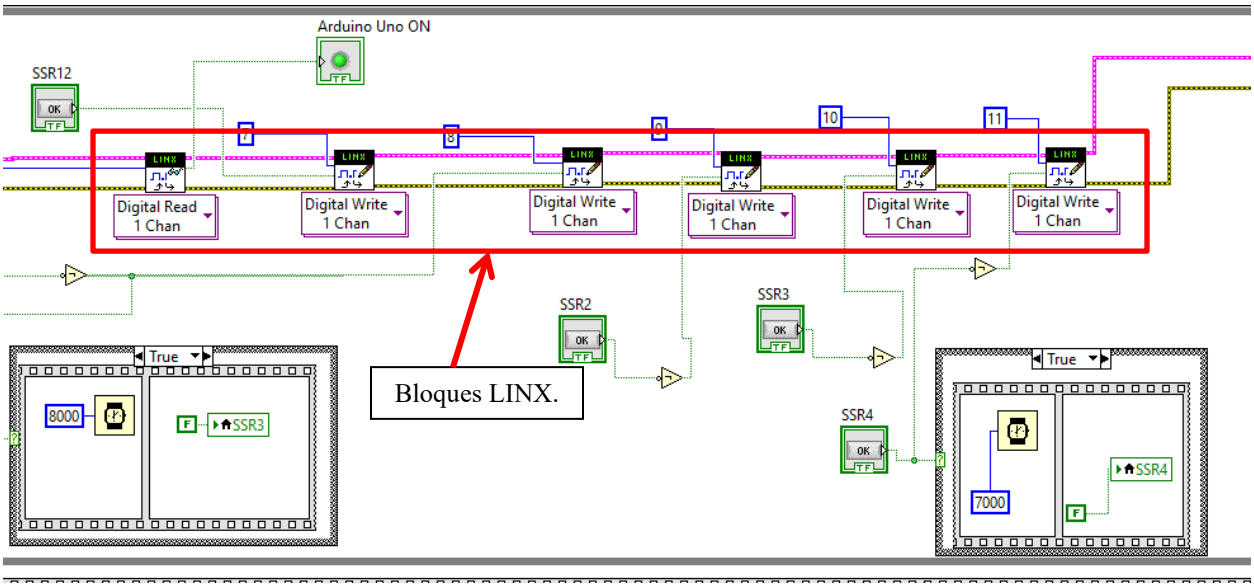

Figura 10 Diagrama de bloques utilizando bloques LINX.
Fuente: (Manosalva & Picón, 2019, p. 58).

2.1.3. Lógica secuencial del martillo utilizando LINX.

La intención de este ítem es saber cuáles son las acciones que deberá desarrollar el algoritmo que será implementado en la placa Arduino Mega para efectos de la simulación y deben ser realizados de manera secuencial, este proceso se observa en la figura 11 (Manosalva & Picón, 2019, p. 45).

- Activar motor en el sentido de giro que permite el ascenso del martillo.
- Cuando alcanza el nivel superior bien posicionado apaga el motor que está activado.
- Pasado un tiempo activa el motor que ancla el martillo.
- Cuando el martillo está anclado se envía la señal que apaga el martillo.
- Se invierte el giro que sube el martillo para destensionar la guaya, esta acción se realiza de forma automática pasado un tiempo el cual permita realizar la operación segura del péndulo.

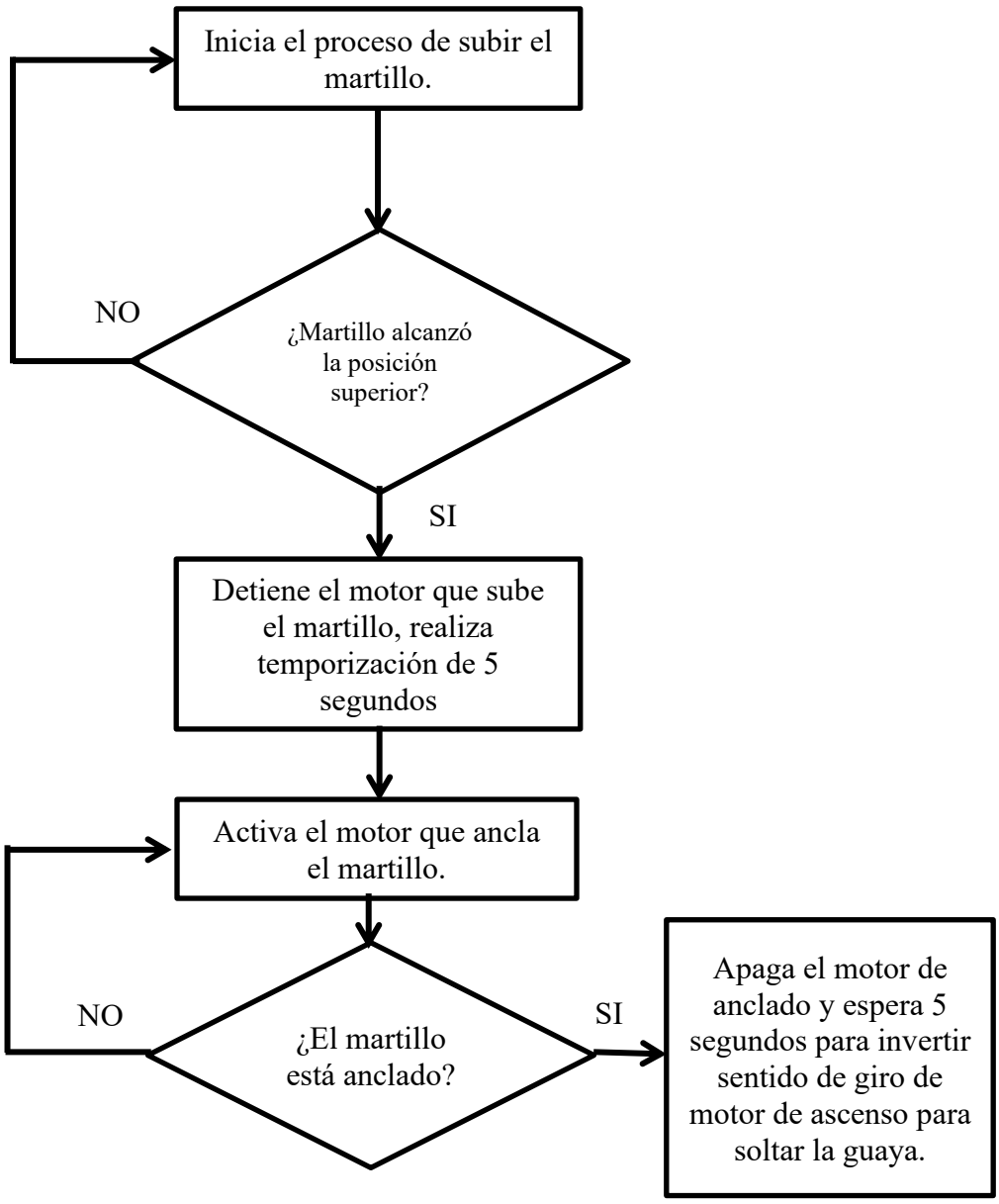

Figura 11 Diagrama de flujo en operación automática.
Fuente: (Manosalva & Picón, 2019, p. 45).

2.1.4. Entradas y salidas digitales.

Para efectos de la simulación adicional a identificar los algoritmos de la operación del péndulo, es necesario saber cuáles son los pines utilizados como entradas y salidas digitales que permiten la operación de los motores y leer los finales de carrera, esto para programar la placa

Arduino Mega y para efectos de lograr una simulación acorde a las conexiones reales del péndulo Charpy. Para el caso de las entradas digitales se describen los pines y sus funciones en la tabla 2, aquí los llamados SSR son los que funcionan como elementos para realizar la conexión entre la salida de control de 5V de la placa Arduino Mega con el circuito de control cableado a 110V.

Pin a programar en la placa Arduino Mega como salida digital.	Pines en nueva programación	Señal sobre la cual actúa la salida digital.
7	40	Activa la placa de SSR.
8	41	SSR1 Subir martillo
9	42	SSR2 Bajar Guaya
10	43	SSR3 Anclar martillo
11	44	SSR4 Impacto o desanclar martillo.

Tabla 2 Salidas digitales de la placa Arduino Mega.
Fuente: Autores.

En cuanto a las entradas digitales se utilizan los pines indicados en la tabla 3, que fueron identificados directamente del diagrama de bloques programados en LabVIEW, como se observa en la figura 12 en la cual se indica que fueron utilizados los bloques “Digital Read”, adicional a estas señales se agregaran las señales que permitirán asegurar la operación del péndulo para saber cuál es el estado del sistema que ancla el martillo e indicarle al microcontrolador que se desea realizar si la operación se realizara sin contar con LabVIEW.

Pin a programar en la placa Arduino Mega como entrada digital	Señal sobre la cual actúa la salida digital.
3	Martillo Arriba
2	La placa Arduino Uno que conecta con la Aplicación móvil se encuentra operando.
30	Martillo desanclado (Nueva)
31	Martillo anclado (Nueva)
32	Subir martillo (Nueva)
33	Bajar martillo (Nueva)
34	Impacto (Nueva)

Tabla 3 Entradas digitales de la placa Arduino Mega.
Fuente: Autores.

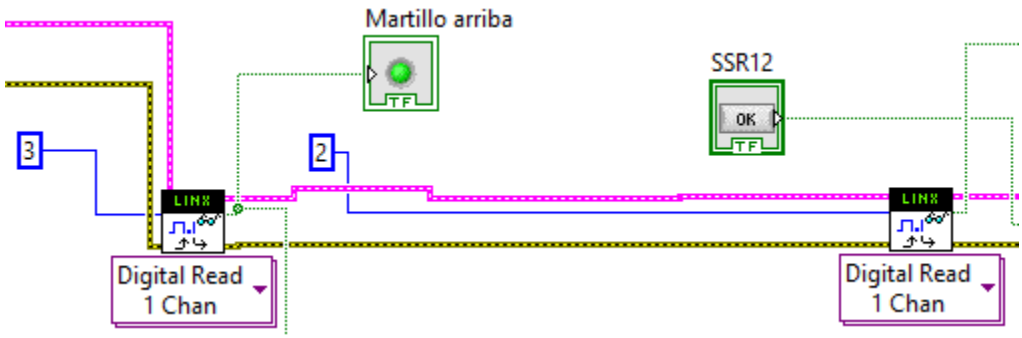

Figura 12 Bloques utilizados actualmente por péndulo Charpy.
Fuente: Autores.

2.1.5. Circuitos de potencia y lógica de control cableada.

Los circuitos de potencia son utilizados para controlar el sentido de giro de los motores eléctricos y aquí lo que se analizara es la lógica que se debe seguir para llevar a cabo las acciones necesarias a ejecutar por el péndulo Charpy. Para el caso del motor que se encarga de subir el martillo se deberán activar los relés indicados en la figura 13 y la secuencia es la descrita a continuación:

- Subir motor: Se activará el relevador R5 que es controlador por el SSR1.
- Soltar guaya: Activa el relevador R6 que es controlado por el SSR2.

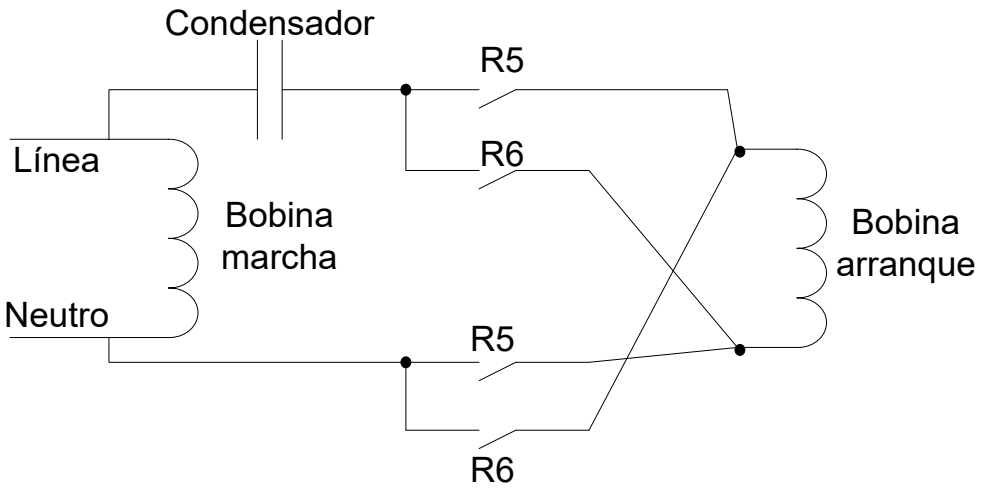

Figura 13 Circuito para invertir el sentido de giro del motor que eleva el martillo.
Fuente: (Manosalva & Picón, 2019, p. 51).

Para el caso del motor que se encarga de anclar el martillo se realizaron las conexiones de la figura 14, y la lógica para operar estas acciones son las descritas a continuación:

- Anclar martillo: Se activará el relevador R7 a través del SSR3.
- Impacto: Se activará el relevador R8 a través del SS4.

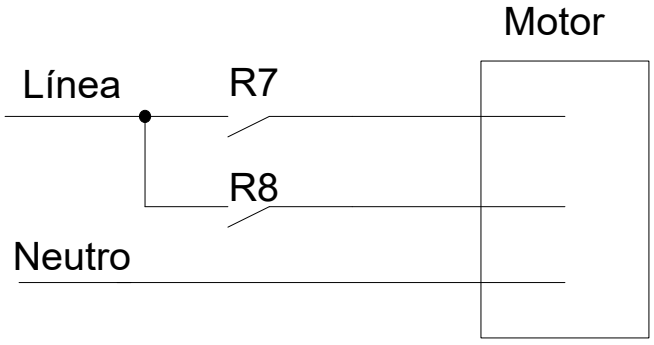

Figura 14 Circuito para invertir el giro del motor de anclaje del martillo.
Fuente: (Manosalva & Picón, 2019, p. 51).

La lógica de control cableado indicada en la figura 15 es importante tenerla presente, a través de este se logra controlar el péndulo Charpy y aquí se tienen en cuenta nuevos elementos como los contactores K1 y K2 que se encargan de activar el motor para elevar el martillo y

anclar el martillo respectivamente, este será utilizado para comprobar la operación de las simulaciones.

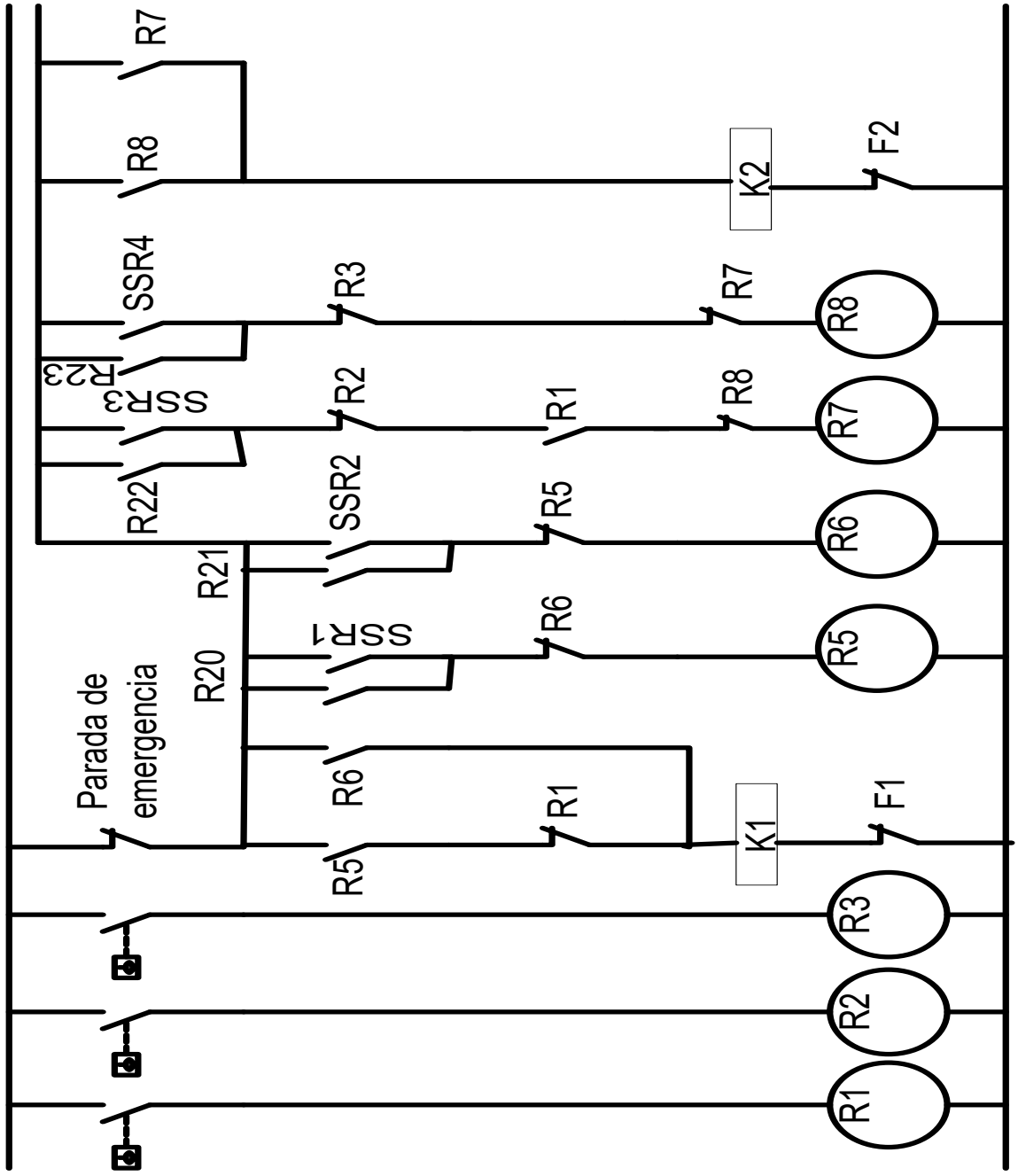

Figura 15 Circuito de control cableado
Fuente: (Manosalva & Picón, 2019, p. 54).

3. Algoritmos para Arduino Mega.

Teniendo en cuenta que el control ahora planteado y el cálculo de la energía de impacto se realizará directamente por el microcontrolador de la placa Arduino Mega y que LabVIEW será utilizado para visualizar la operación del péndulo. En este capítulo se desarrollarán los algoritmos pertinentes que posteriormente serán simulados utilizando el software Proteus. El código completo está disponible en el Anexo I.

3.1. Características del nuevo algoritmo.

En la figura 16 se resumen las nuevas características que se incluirán en la programación del Microcontrolador de la placa Arduino Mega, esto permitirá ahora a la operación del péndulo Charpy que se realizase de forma paralela con la aplicación móvil y LabVIEW.

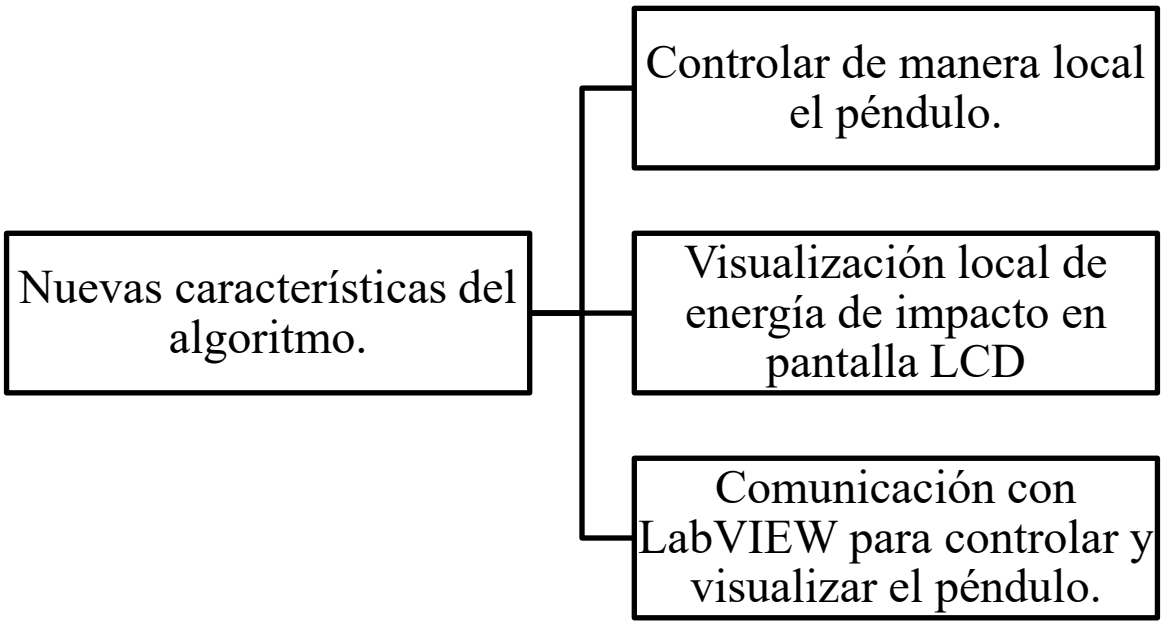

Figura 16 Nuevas características del algoritmo del Microcontrolador.
Fuente: Autores.

3.2. Algoritmo y programación para operación local.

En esta etapa de la programación se deben tener en cuenta que se han añadido tres señales que permitirán operar el péndulo cuando no se cuenta con la aplicación móvil ni LabVIEW, pero igualmente el péndulo operará de forma automática para anclar el martillo y para ello se realizará la lógica planteada en la figura 17.

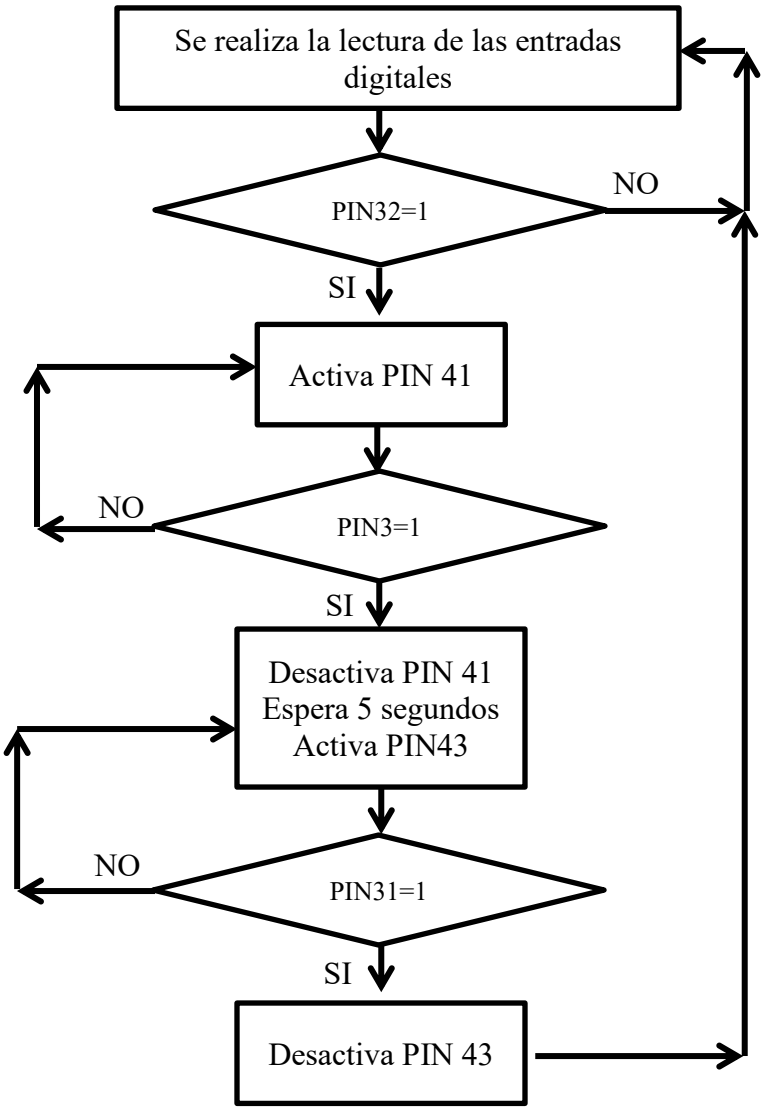

Figura 17 Algoritmo para operación local.
Fuente: Autores.

Para emular el estado de las entradas digitales se dispone en Proteus de una serie de indicadores pulsadores y para las salidas digitales indicadores LED (ver figura 18), de esta forma al realizar el análisis de los resultados más adelante estas serán las señales que indicarán que acción se encuentra realizando el microcontrolador y determinar si cumple la función que permite operar el péndulo Charpy.

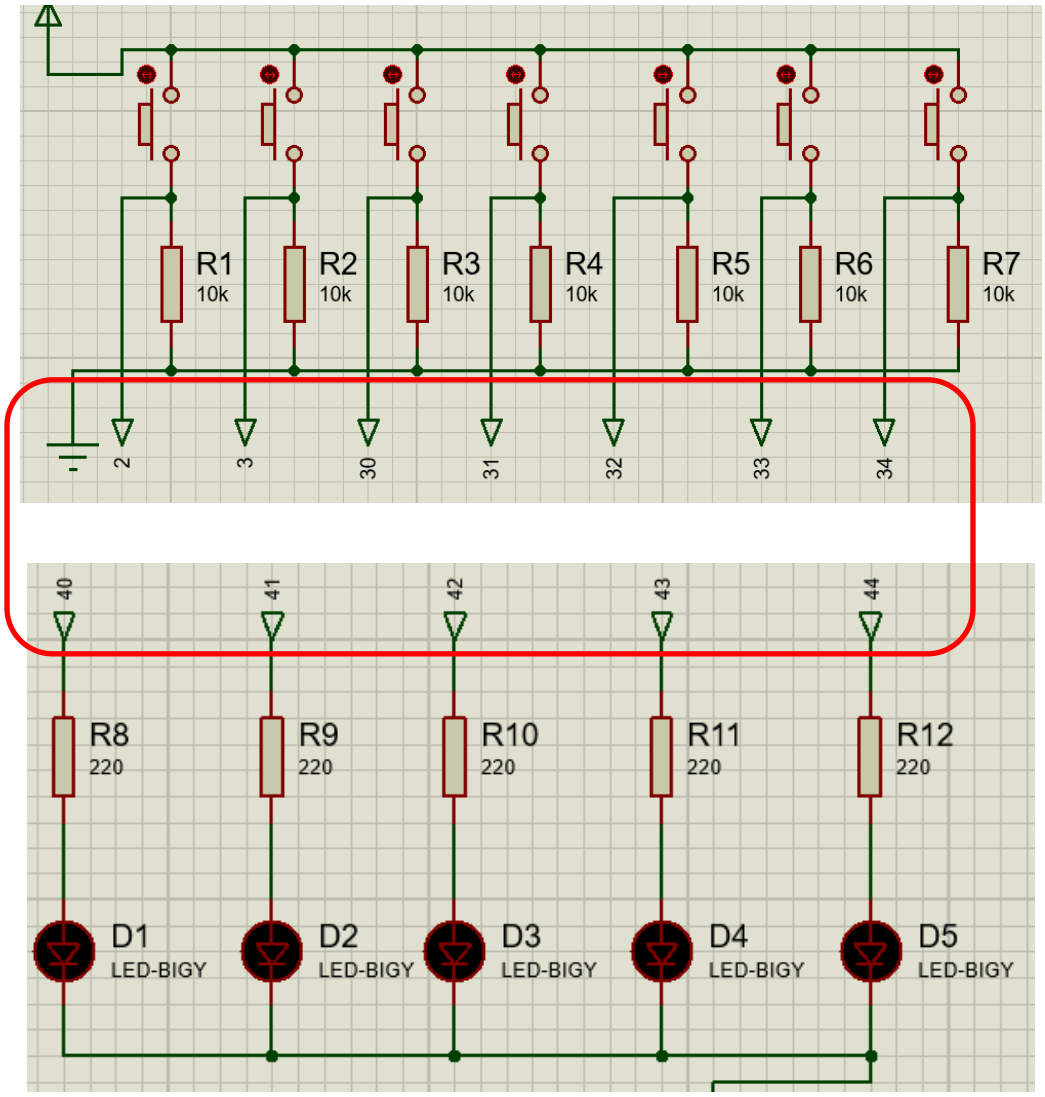

Figura 18 Elementos en Proteus para emular entradas y salidas digitales.
Fuente: Autores.

3.3. Algoritmo y programación de nueva visualización local en LCD

En este trabajo se plantea utilizar una pantalla LCD grafica de 64x128 pixeles cuyo controlador es el KS0108 y en Proteus tiene la referencia LGM12641BS1R (ver figura 19).

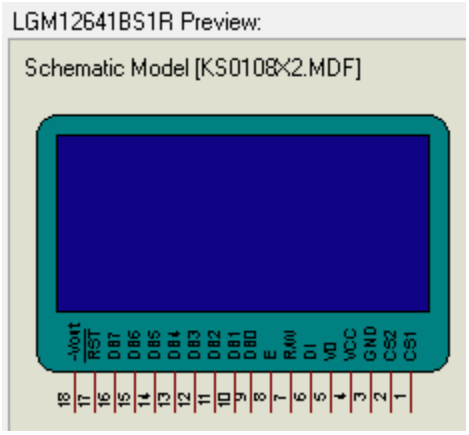

Figura 19 Pantalla LCD gráfica 64x128 obtenida de Proteus
Fuente: Autores.

Antes de desarrollar el algoritmo de la figura 20 es necesario establecer cuál será la información aquí presentada y teniendo en cuenta que la idea central de este proyecto es determinar la energía de impacto este será un dato a indicar, adicional a ello se indicarán los ángulos θ y β (ver ecuación 1), también se presentará si el martillo ubicado está en la posición superior, si el martillo está anclado o desanclado. Para controlar esta pantalla se incluirá inicialmente la librería “basicGLCD.h”. La relación que existe entre el estado mostrado en la pantalla y la condición de operación del péndulo se presenta en la tabla 4 dejando claro entonces que los estados dependerán de las entradas digitales.

Estado de entrada digital	Indicación en LCD
3	Si esta entrada digital esta activada se indicara que el martillo se encuentra arriba bien posicionado
30	Si esta activada esta entrada se indicara que el martillo está desanclado
31	Si esta activada esta entrada se indicara que el martillo está anclado

Tabla 4 Relación entre estado de entrada digital y visualización en LCD.
Fuente: Autores.

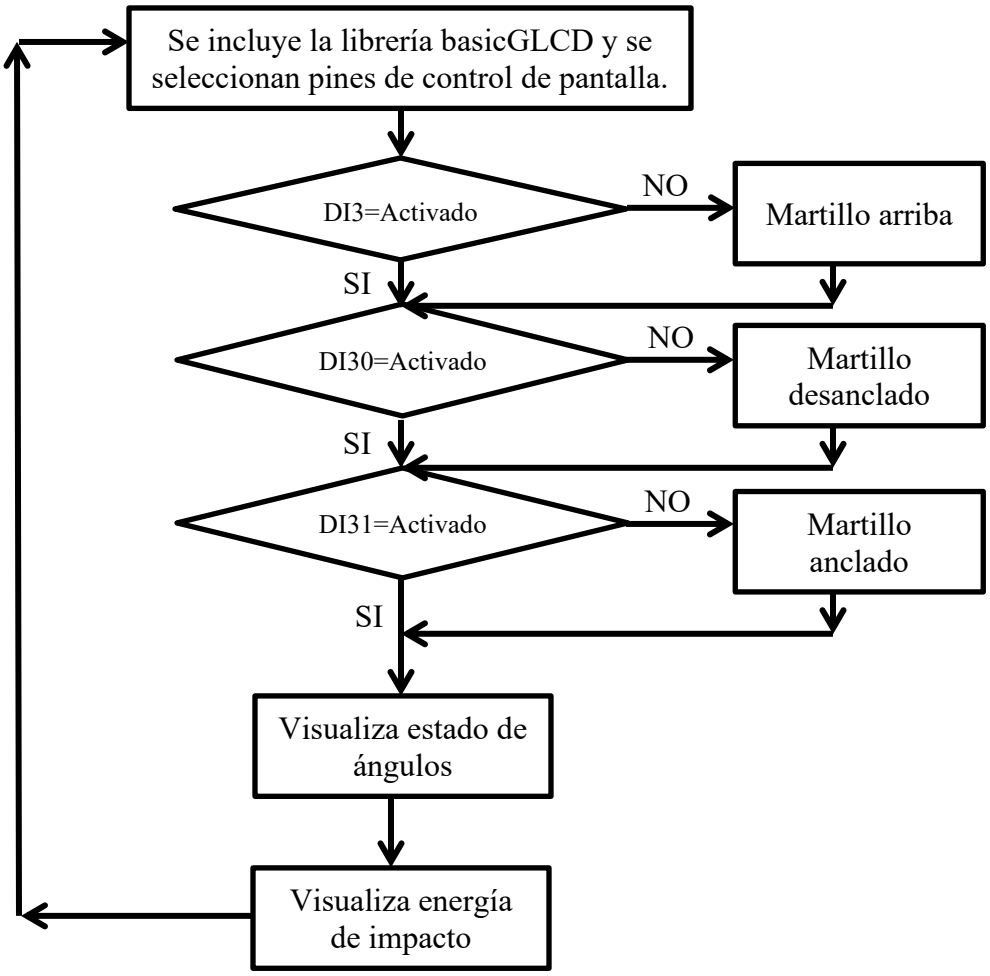

Figura 20 Algoritmo del Microcontrolador para controlar pantalla LCD.
Fuente: Autores.

Para finalizar este ítem se deja presente a través de la figura 21 como se realiza la conexión entre el Arduino Mega y la pantalla LCD, la función de cada uno de los pines se describen en el Anexo I donde se presenta el detalle de la programación realizada.

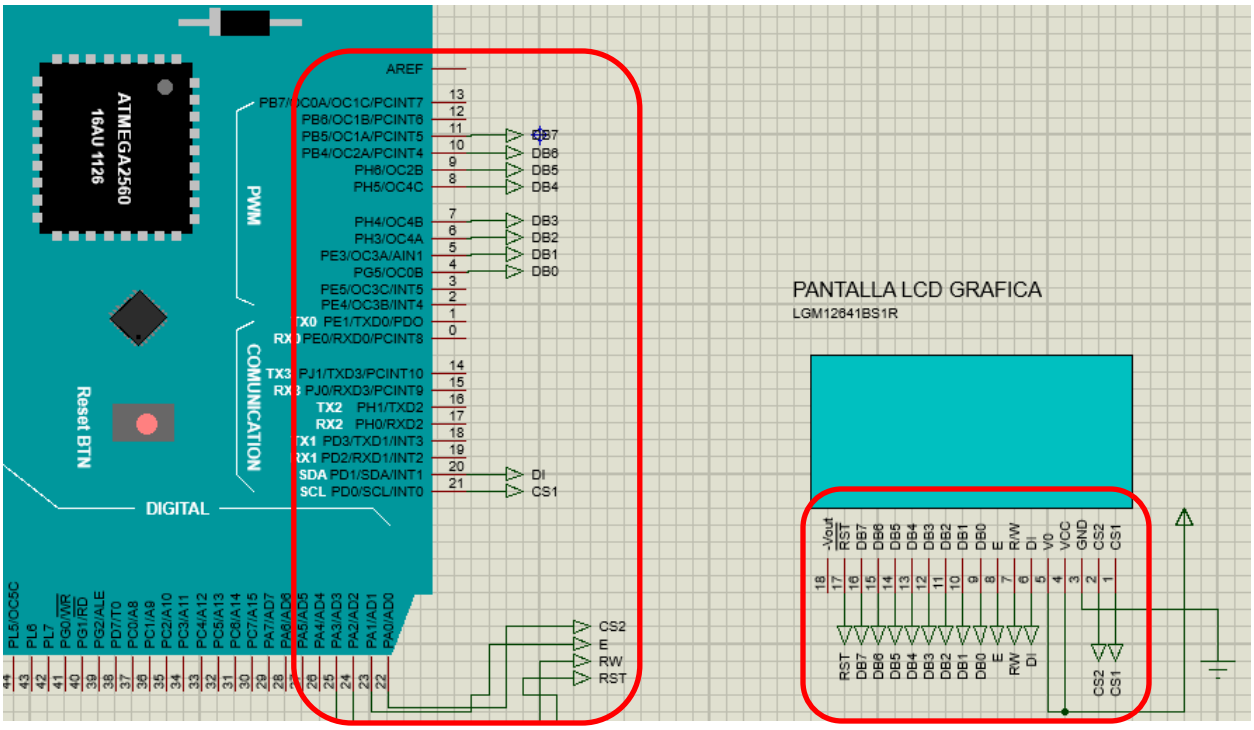

Figura 21 Conexión entre Arduino Mega y pantalla LCD.
Fuente: Autores.

3.4. Algoritmo y programación de nueva comunicación con LabVIEW.

Antes de iniciar con el desarrollo de los algoritmos se hace referencia al software VSPE (Virtual Serial Port Emulator) cuyo objetivo es crear un puerto virtual que servirá como enlace entre Proteus y LabVIEW, utilizando la opción Device type (Conector) como lo indica a figura 22 y posterior a esto seleccionando el nombre de puerto que se ira a utilizar que para el caso de este trabajo es COM1 (ver figura 23).

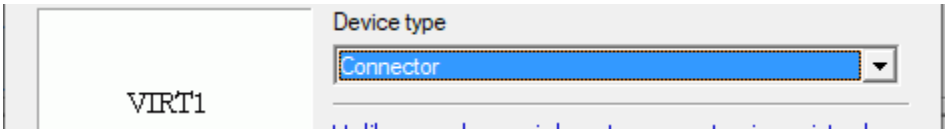

Figura 22 VSPE (Virtual Serial Port Emulator)
Fuente: Autores.

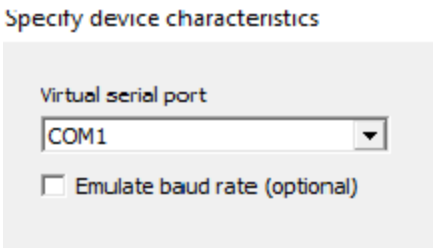

Figura 23 VSPE, seleccionado COM1.
Fuente: Autores.

Adicional a esto Proteus cuenta con una herramienta llamada "COMPIM" con este es posible transmitir la información que se encuentra en el puerto serial del Arduino Mega a LabVIEW utilizando como enlace el VSPE, la configuración de este se debe realizar como quedo establecido en la figura 24 indicando la velocidad de transmisión y seleccionando el puerto COM1, todos estos pasos son necesarios para logar una simulación exitosa.

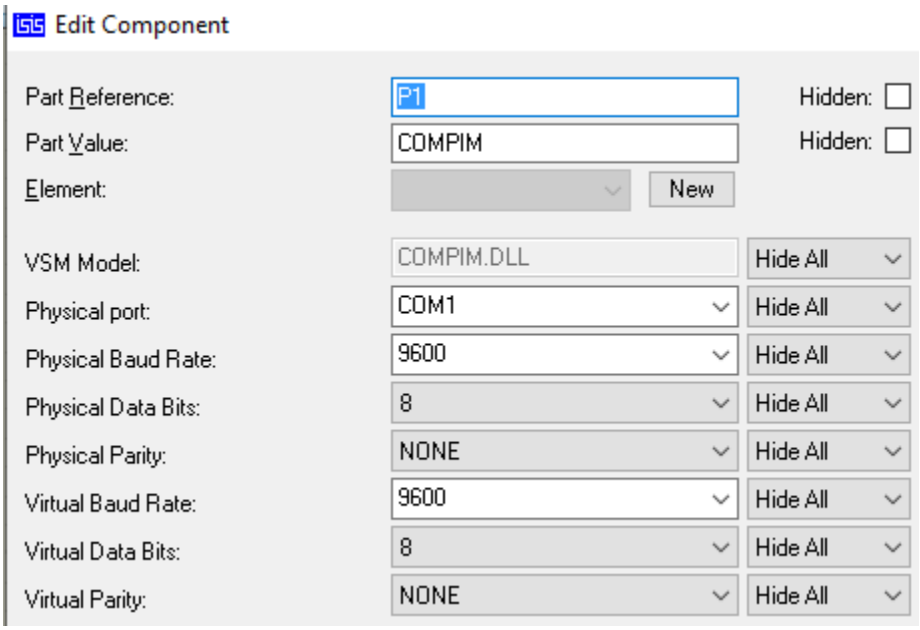

Figura 24 Configuración de puerto serial en Proteus
Fuente: Autores.

El algoritmo de la figura 25 cumple las funciones de transmitir el estado de las entradas digitales y la posición del martillo que para el caso son sus respectivos ángulos θ y β (ver ecuación 1), también recibirá información de cuál acción se desea ejecutar desde LabVIEW.

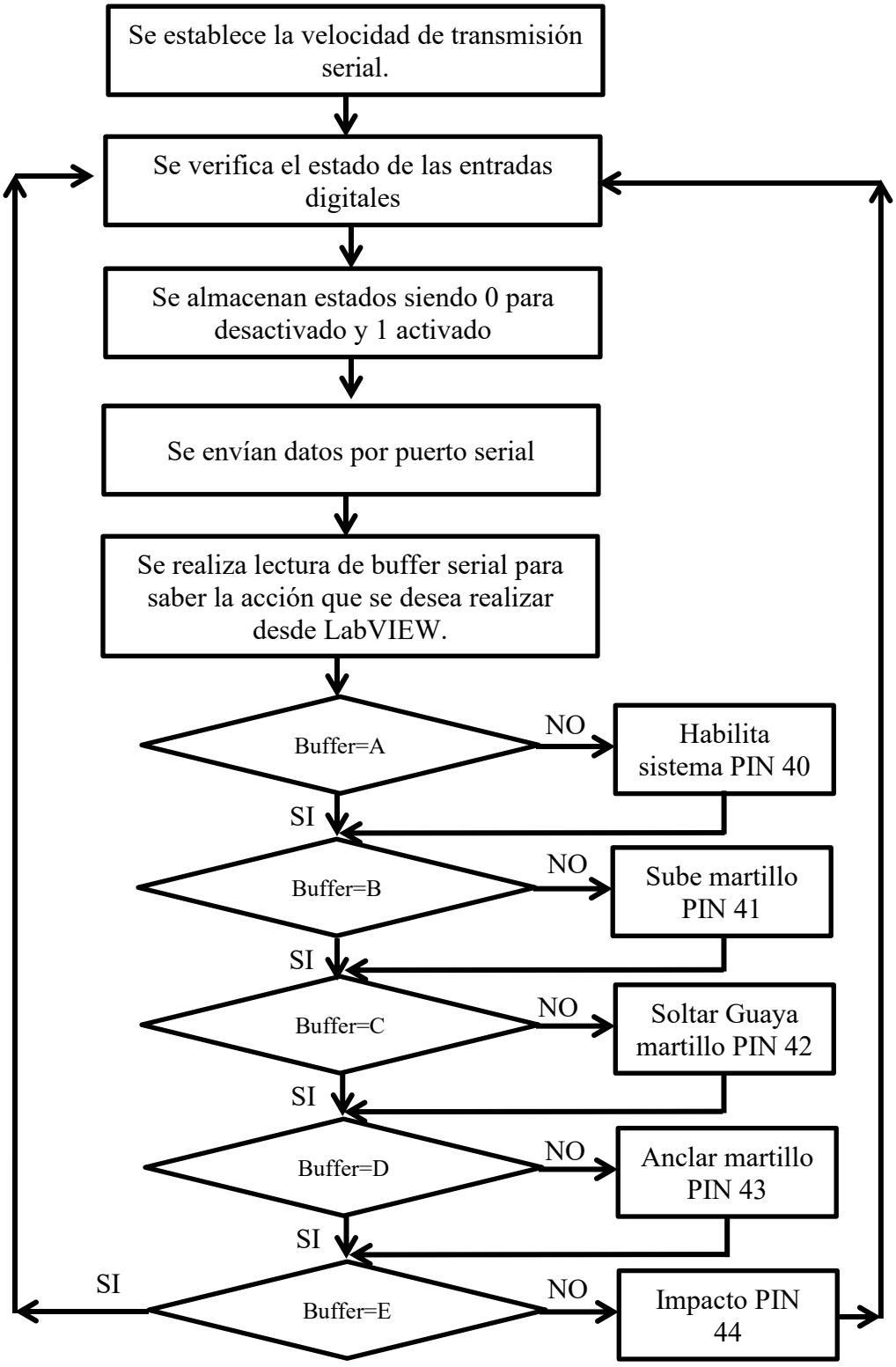

Figura 25 Algoritmo del Microcontrolador para comunicación serial.
Fuente: Autores.

Los elementos utilizados para realizar esta comunicación se realizan como lo indica la figura 26, y hacen parte del conjunto para emular la operación del péndulo Charpy para determinar la energía de impacto y operar los motores utilizando bloques VISA.

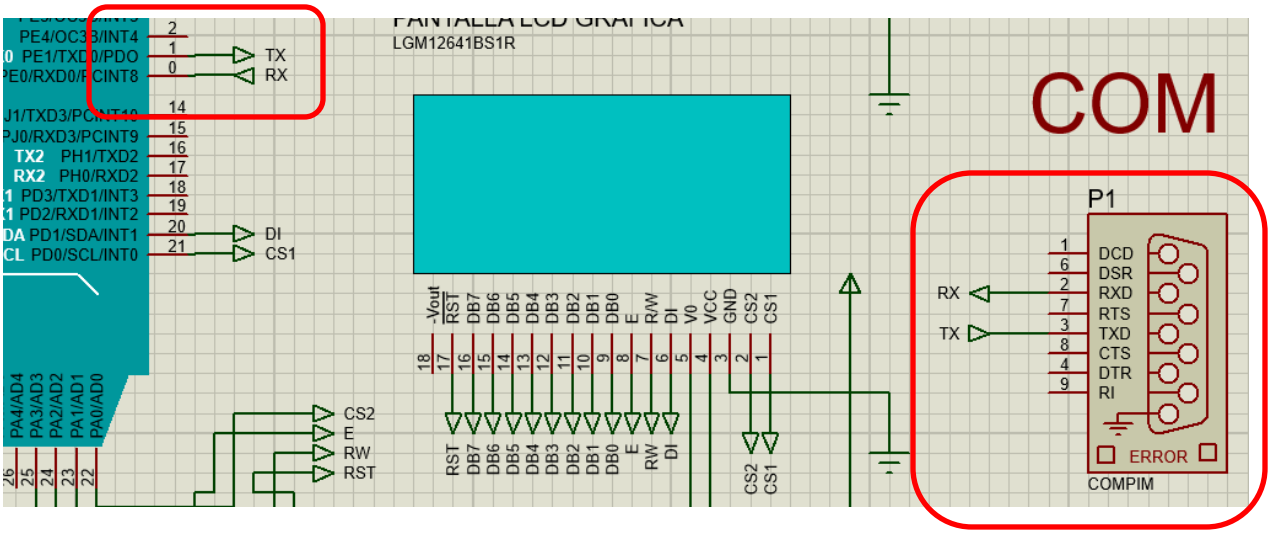

Figura 26 Conexión entre Arduino Mega y COMPIM
Fuente: Autores.

3.5. Panel de control local

El panel frontal ubicado en el péndulo contará con una pantalla LCD gráfica, una muletilla de tres posiciones para subir y bajar martillo, además de un pulsador cuya función será desanclar el martillo y hace referencia a la función impacto, en la figura 27 es posible observar esta alternativa.

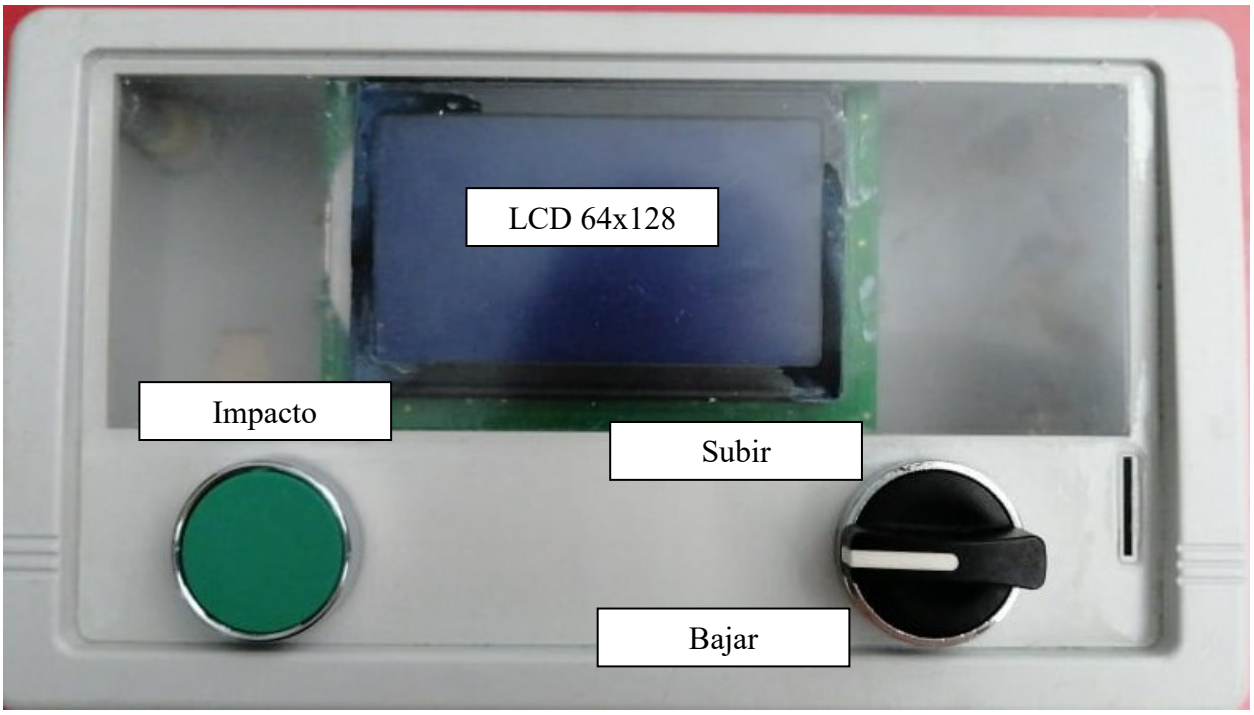

Figura 27 Alternativa de panel a ubicar en el péndulo.
Fuente: Autores.

4. Alternativa para determinar los grados de inclinación.

Teniendo claro los aspectos técnicos que permiten controlar los motores del péndulo Charpy se inicia a plantear cuál es la mejor opción o alternativa para determinar la posición del martillo.

4.1. Periféricos del microcontrolador Atmel 2560

Lo importante de presentar estos periféricos es mostrar que cada uno de los sensores que se van a evaluar se podrían conectar al microcontrolador Atmel 2560. Aquí se evalúan tres tipos de sensores con los cuales se puede determinar la posición y son, acelerómetro, giroscopio, potenciómetro. Los periféricos son:

- Entradas y salidas digitales.
- Entradas análogas.
- Interfaz RS232.
- Interfaz I2C.
- Interfaz SPI.

4.1.1. Acelerómetro MMA845

Este acelerómetro funciona en los tres ejes y tiene una resolución de 12 bits. Este dispositivo permite seleccionar las escalas de intensidad de fuerzas G en: +/-2g, +/-4g, +/-8g con filtros pasa alto, en la figura 28 se observan los pines de salida del sensor los cuales pueden ser conectados directamente a los pines SDA y SCL del Arduino Mega ya que este cuenta con la interfaz I2C, además éste pude trabajar con voltajes entre 2 - 3.3V

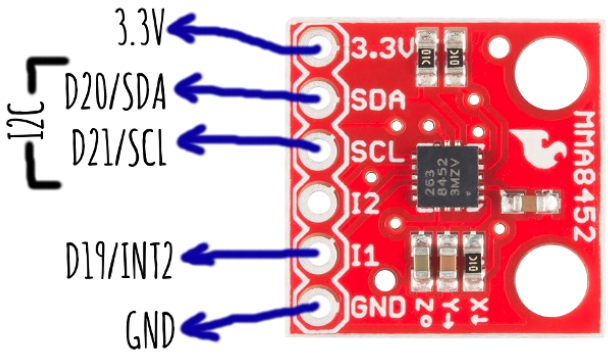

Figura 28 Acelerómetro
Fuente: Acelerómetro MMA845. Recuperado de < http://interactingobjects.com/tag/rotary-encoder/>

4.1.2. Giroscopio POLOLU 2739

Si la solución fuese un giroscopio este podría ser conectado al Arduino Mega utilizando la interfaz I2C. El sensor de referencia POLOLU 2739, el cual integra una unidad de medida inercial, el altímetro, giroscopio y acelerómetro, magnetómetro y barómetro digital, puede ser alimentado desde 2.5V hasta 5.5V (ver figura 29)

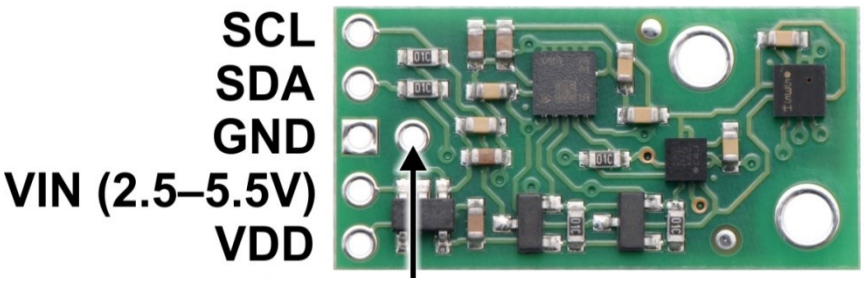

Figura 29 Giroscopio
Fuente: Giroscopio POLOLU 2739. Recuperado de < https://www.sigmaelectronica.net/producto/pololu-2739/>

4.1.3. Potenciómetro.

Otra alternativa viable para determinar la posición angular del martillo es un potenciómetro (ver figura 30) conectado al eje donde actualmente se encuentra conectado el sensor de aguja de medición local. Este sensor convierte la posición en una resistencia variable

que al ser polarizada entregaría un voltaje proporcional a la posición angular. La señal de salida puede ser conectada a una de las 16 entradas análogas del Arduino Mega.

Figura 30 Potenciómetro.
Fuente: Potenciómetro. Recuperado de < https://www.ecobadajoz.es/potenciometros/potenciometro-multivuelta-2k-2w-eje-635mm-lineal-pot2218m2-2k.html>

4.2. Matrices para seleccionar sensor de posición angular.

Después de saber cuáles son las alternativas para determinar la posición angular se entra a seleccionar cuál de ellas presenta la mejor opción. Para iniciar se establecen los criterios que se desean comparar que son, interfaz, voltaje de alimentación, precio. Después de esto se establece cual es el criterio de mayor importancia y para ello se utiliza la ecuación 2 que determina el factor de ponderación, en la tabla 5 se aplica esta ecuación siendo 1 poco importante y 10 muy importante.

$$FP = \frac{Suma\ parcial}{Suma\ total}$$

Ecuación 2 Factor de ponderación.

	Interfaz	Voltaje	Precio	Suma	Factor de ponderación
Interfaz		10	10	20	**0,98**
Voltaje	0,1		0,1	0,2	0,01
Precio	0,1	0,1		0,2	0,01
	Total			**20,4**	

Tabla 5 Criterios de selección.
Fuente: Autores.

De la tabla anterior es claro que el criterio de mayor importancia para determinar cuál es el sensor para determinar la posición angular es la interfaz que permite establecer la comunicación con el Arduino Mega. Ahora en las tablas 6, 7 y 8 se presentan las comparaciones de los diferentes sensores comparando cada uno de los criterios y mediante la ecuación 3 se determina el peso de la opción.

$$Peso\ de\ la\ opción = \frac{Suma\ parcial}{Suma\ total}$$

Ecuación 3 Peso de la opción.

Interfaz	Potenciómetro	Acelerómetro	Giroscopio	Suma	Peso de la opción
Potenciómetro		10	10	20	**0,90**
Acelerómetro	0,1		1	1,1	0,05
Giroscopio	0,1	1		1,1	0,05
	Total			**22,2**	

Tabla 6 *Interfaz versus sensores.*
Fuente: Autores

Voltaje	Potenciómetro	Acelerómetro	Giroscopio	Suma	Peso de la opción
Potenciómetro		1	1	2	**0,33**
Acelerómetro	1		1	2	0,33
Giroscopio	1	1		2	0,33
	Total			**6**	

Tabla 7 *Voltaje versus sensores.*
Fuente: Autores

Precio	Potenciómetro	Acelerómetro	Giroscopio	Suma	Peso de la opción
Potenciómetro		1	1	2	**0,33**
Acelerómetro	1		1	2	0,33
Giroscopio	1	1		2	0,33
	Total			**6**	

Tabla 8 *Precio versus sensores.*
Fuente: Autores

Finalmente se realiza la tabla 9 donde se multiplica el peso de la opción por el factor de ponderación y se suman cada uno de los valores obtenidos por cada criterio de este análisis se tienen que el potenciómetro presenta la mejor opción, por esta razón de ahora en adelante todos los diseños y análisis se basarán en este sensor de posición angular.

Sensor	Interfaz			Voltaje			Precio			Puntaje final
	FP	PO	**Valor**	FP	PO	**Valor**	FP	PO	**Valor**	
Potenciómetro	0,98	0,90	**0,88**	0,01	0,33	**0,00327**	0,01	0,33	**0,00327**	0,89
Acelerómetro	0,98	0,05	**0,05**	0,01	0,33	**0,00327**	0,01	0,33	**0,00327**	0,06
Giroscopio	0,98	0,05	**0,05**	0,01	0,33	**0,00327**	0,01	0,33	**0,00327**	0,06

Tabla 9 *Matriz final para seleccionar el sensor de posición angular*
Fuente: Autores

5. Caracterización del sensor de posición y algoritmos para el microcontrolador ATMEL 2560.

Antes de realizar la caracterización del sensor de posición que para el caso es un potenciómetro se plantea la alternativa para realizar la conexión del mismo al eje y estructura del péndulo, para ello será necesario elaborar un accesorio que conecte el eje con el potenciómetro tal como lo indica la figura 31, este se ubicará donde actualmente se encuentra ubicado el indicador de aguja.

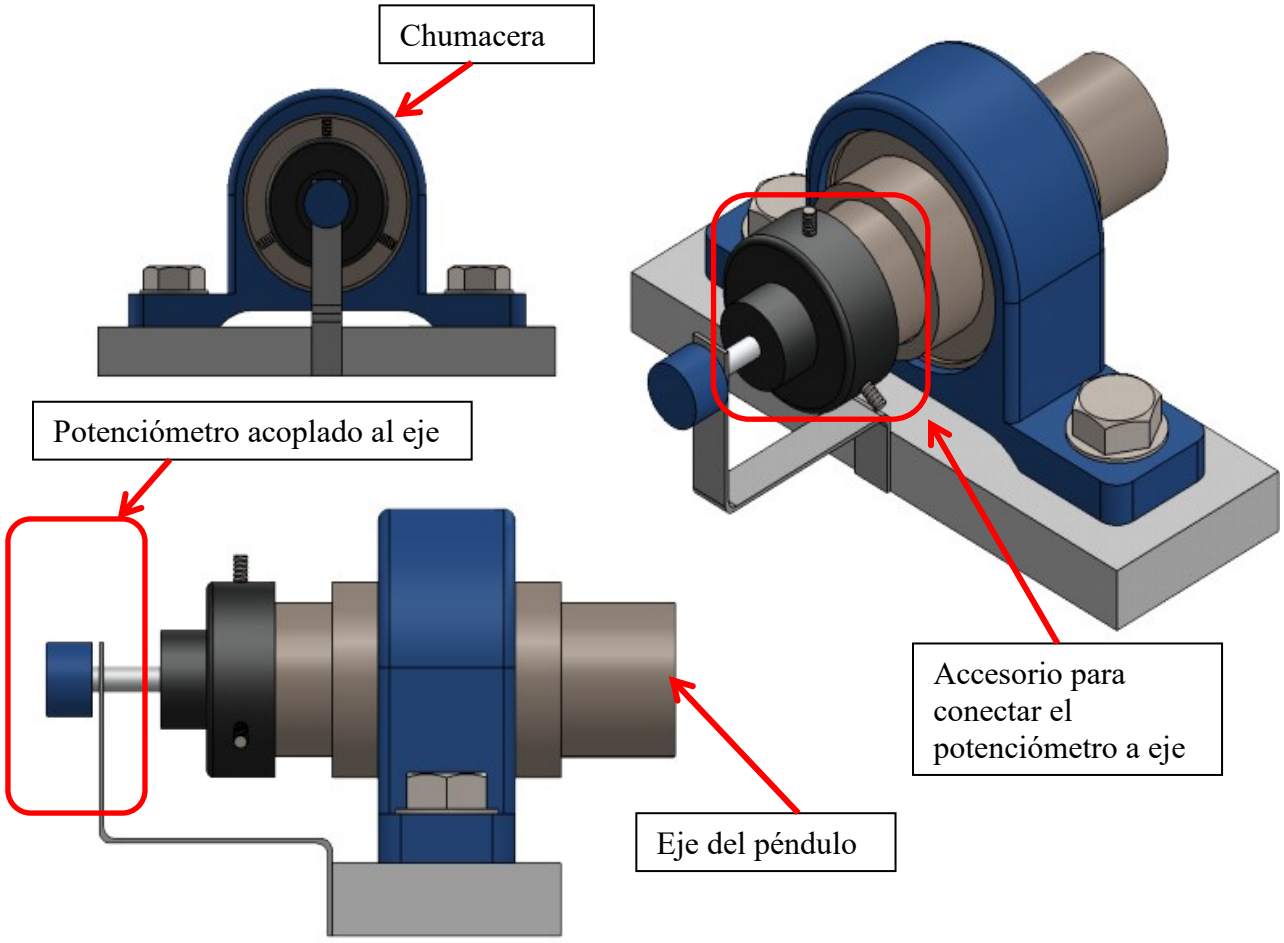

Figura 31 Facilidad para conectar el potenciómetro al eje del péndulo.
Fuente: Autores.

5.1. Procedimiento para determinar ecuaciones.

Para realizar la caracterizacion del sistema e identificar las ecuaciones del ángulo 1 y 2 se utilizará la figura 32 y se seguirán los pasos descritos a continuación:

- Se ubica el martillo en posición completamente vertical y para ello se utilizará un nivel de gota fija de tal forma que el voltaje a la salida del potenciómetro sea 2.5V.
- Se eleva el martillo hasta alcanzar la posición de anclaje siendo este ángulo de 120°, este valor es tomado del documento de (Manosalva & Picón, 2019).
- Estando anclado el martillo se registra el voltaje que para efectos de la simulación se tomara como 4V.
- Se ubica nuevamente el martillo en posición vertical y hasta que el voltaje a la salida del potenciómetro sea 2.5V.
- Se eleva el martillo hasta alcanzar la posición máxima de elevación que serán 180°.
- Se registra el voltaje a la salida del potenciómetro y para efectos de la simulación se tomará como 0V.

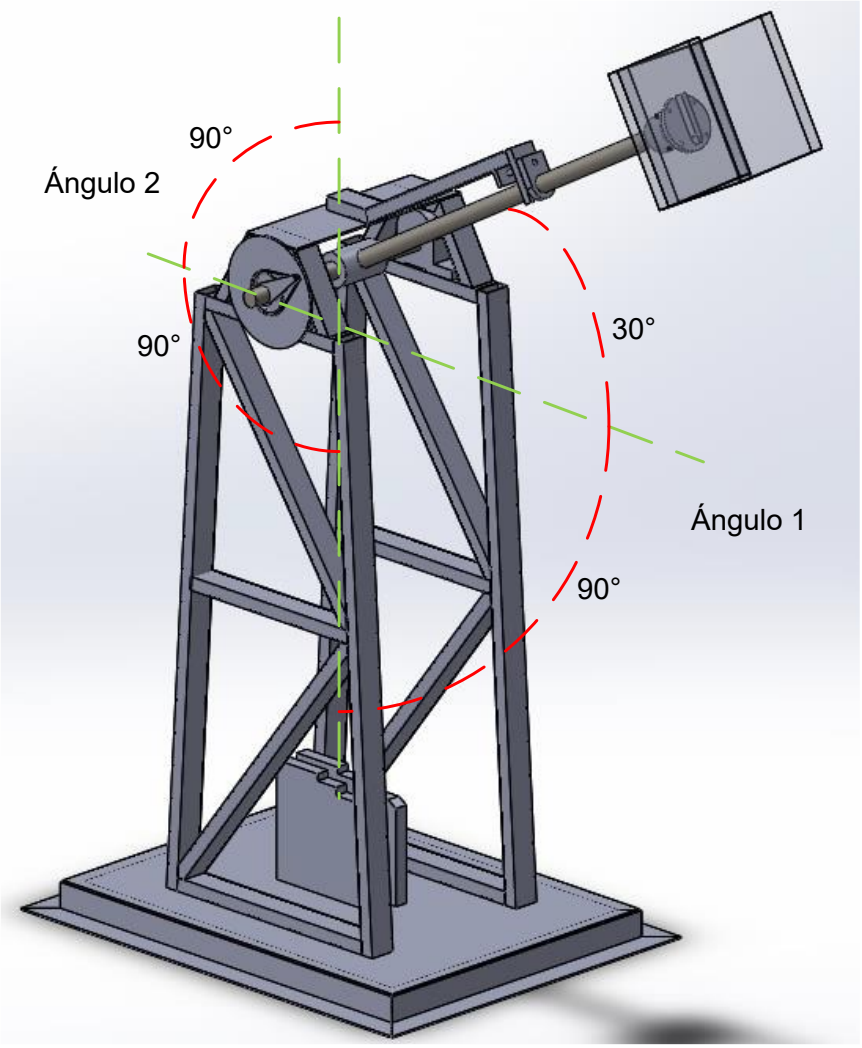

Figura 32 Ángulos para caracterización.
Fuente: Autores.

5.1.1. Caracterizacion del ángulo 1.

Para obtener la ecuación que relaciona el ángulo 1 de inclinación con el voltaje de salida del potenciómetro se siguen los siguientes pasos:

- Se debe tener en cuenta que 2.5V corresponden a 0°
- Se debe tener en cuenta que 4V corresponden a 120°.

- Se ingresan estos datos a Excel siendo el eje X voltaje y el eje Y grados de inclinación obteniendo como resultado la gráfica de la figura 33.

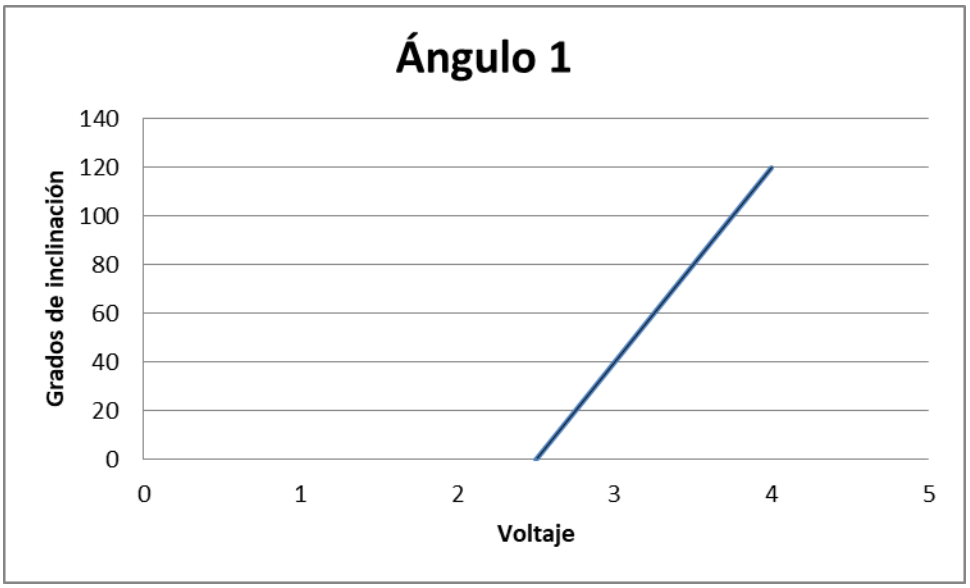

Figura 33 Relación voltaje y grados de inclinación del ángulo 1.
Fuente: Autores.

Utilizando la función que permite agregar la línea de tendencia en Excel se tiene la ecuación **4**, de tal forma que al tener un voltaje de 4V los grados de inclinación serán 120°.

$$Angulo1 = 80 * (Voltaje - 2.5)$$

Ecuación 5 Relación voltaje y grados de inclinación 1.
Fuente: Autores.

5.1.2. Caracterización del ángulo 2.

Para obtener la ecuación que relaciona el ángulo 2 de inclinación con el voltaje de salida del potenciómetro se siguen los siguientes pasos:

- Se debe tener en cuenta que 2.5V corresponden a 0°
- Se debe tener en cuenta que 0V corresponden a 180°.

- Se ingresan estos datos a Excel siendo el eje X voltaje y el eje Y grados de inclinación obteniendo como resultado la gráfica de la figura 34.

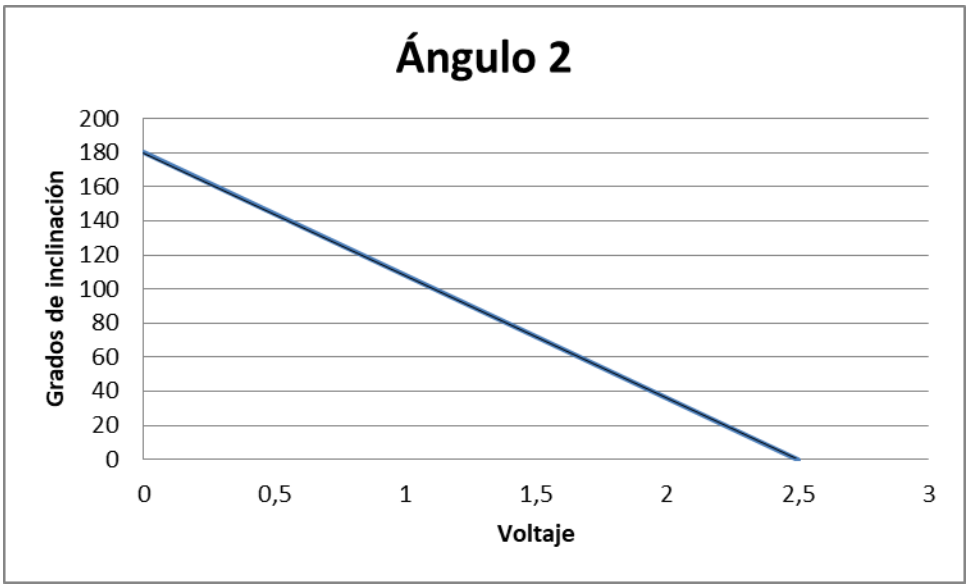

Figura 34 Relación voltaje y grados de inclinación del ángulo 2.
Fuente: Autores.

Utilizando la función que permite agregar la línea de tendencia en Excel se tiene la ecuación **6**, de tal forma que al tener un voltaje de 0V los grados de inclinación serán 180°.

$$Angulo2 = 180 - (72 * Voltaje)$$

Ecuación 7 Relación voltaje y grados de inclinación 2.
Fuente: Autores.

5.2. Algoritmo para determinar ángulos.

Para que el microcontrolador identifique cómo detetminar los ángulos se utilizará el algoritmo de la figura 35, estando el martillo en posición completamente vertical y se da la orden de subir, se inicia el cálculo del ángulo cuyo valor final se dará cuando se alcance la posición superior y se active la entrada digital 3, este dato se almacenrará como ángulo 1, al dar clic en impacto se inicia el descenso del martillo, y cuando este llega a la posición completamente

vertical la salida de voltaje del sensor será 2.5V y continuará descenciendo es ahí donde se inician a guardar datos del ángulo 2 y cuyo valor final se determinará cuando se termine de disminuir voltaje y se inicie nuevamente a aumentar. La programación que cumple esta función se puede ver en el Anexo I.

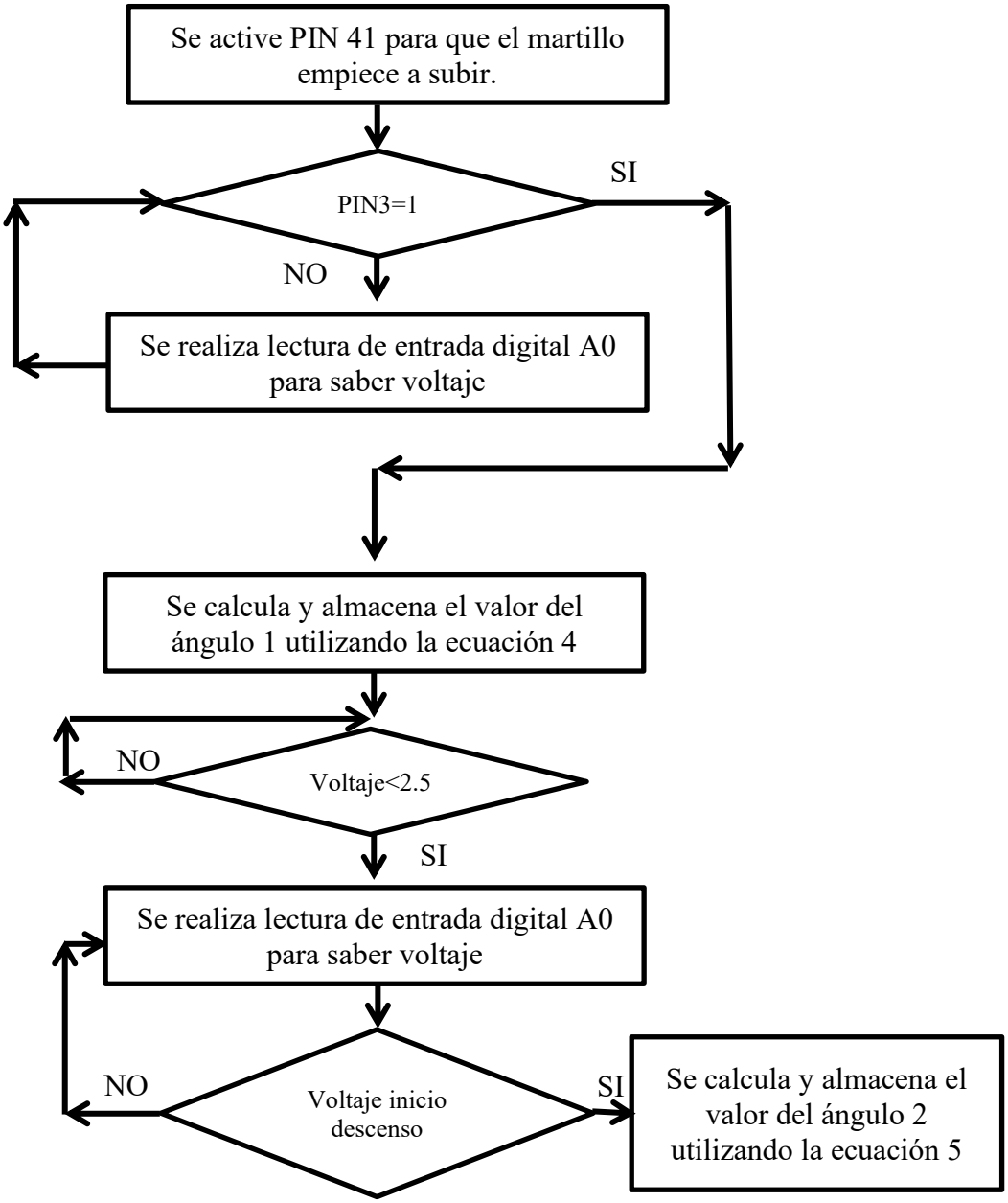

Figura 35 Algoritmo para determinar ángulos de inclinación.
Fuente: Autores.

6. Nuevo algoritmo para el péndulo basado en NI-VISA.

En este capítulo se presentan las modificaciones realizadas al diagrama de bloques y al panel frontal que fue desarrollado en LabVIEW en el cual se utilizaron los bloques NI-VISA.

6.1. Modificaciones realizadas en el panel frontal.

En la figura 36 (a) se observa el panel frontal con el cual funciona actualmente el péndulo, y en la misma figura (b) se observan los botones que fueron incluidos. El botón que tiene como nombre DETENER MOTORES tiene la función de apagar todas las salidas y detener la operación del péndulo, adicional a eso los indicadores LED dejan ver si el martillo está anclado, desanclado, subiendo o bajando y finalmente los indicadores donde se presentan los ángulos θ y β que corresponden a ángulo 1 y ángulo 2 respectivamente y la energía de impacto de la probeta en prueba.

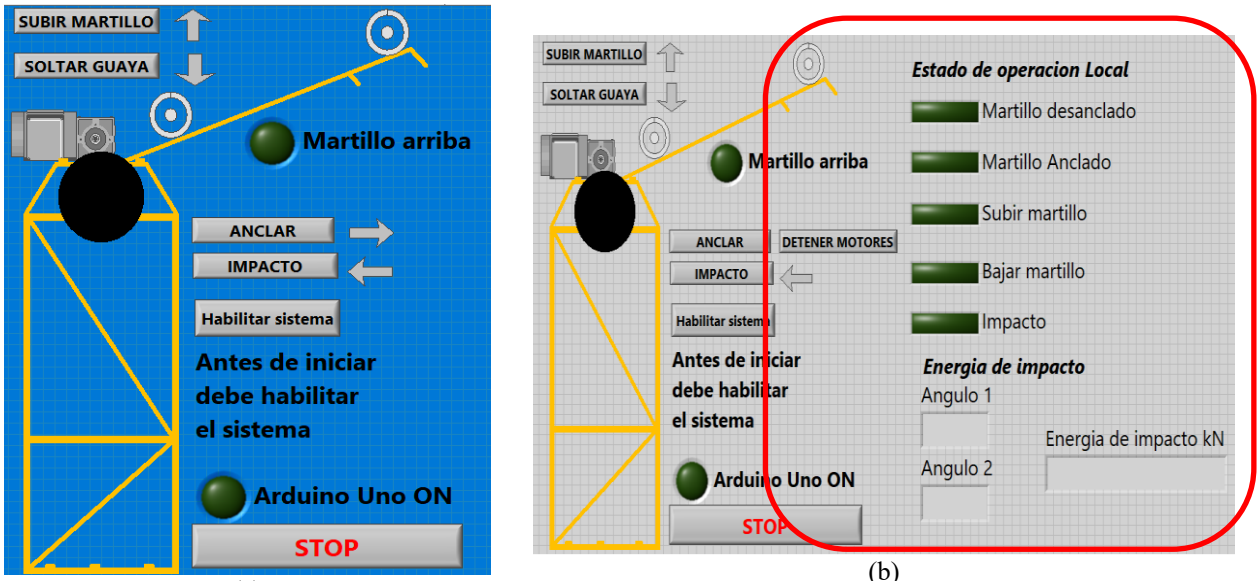

Figura 36 (a) Panel frontal actual y (b) Panel frontal con modificaciones
Fuente: Autores.

6.2. Nuevo diagrama de bloques.

El diagrama de bloques se dividira en dos etapas, la primera de ellas es recepción de datos donde se evidenciarán aspectos técnicos de programación que permiten evidenciar que es posible visualizar el estado operacional del péndulo utilizando funciones NI-VISA y la segunda es la trasmisión de datos donde se registra como es posible controlar de forma paralela la operación del péndulo sin que sea necesario contar continuamente con LabVIEW.

6.2.1. Recepción de datos en LabVIEW con NI-VISA

La recepción de datos también se divide en dos etapas, la primera de ellas se encarga de realizar la lectura de las entradas digitales y la segunda de presentar los ángulos y la energía de impacto resultado de la prueba. La primera de ellas obedece a la secuencia de llegada de datos de cada una de las señales digitales tal como se programó desde el IDE de Arduino (Ver Anexo I). En el algoritmo de la figura 37 es posible identificar como se agrupan los datos para su posterior trasmisión, inicialmente se envía el identificador 110 para LabVIEW identifique que la secuencia inicia siempre con la misma información, después de esto se envía el estado de la entrada digital dos para determinar si el Arduino Uno que se conecta a la App está funcionando, seguido a esto va el dato que determina si el martillo está en la posición superior, después el estado de los finales de carrera que indican el estado anclado y desanclado, para finalizar con la información de subir y bajar martillo de impacto.

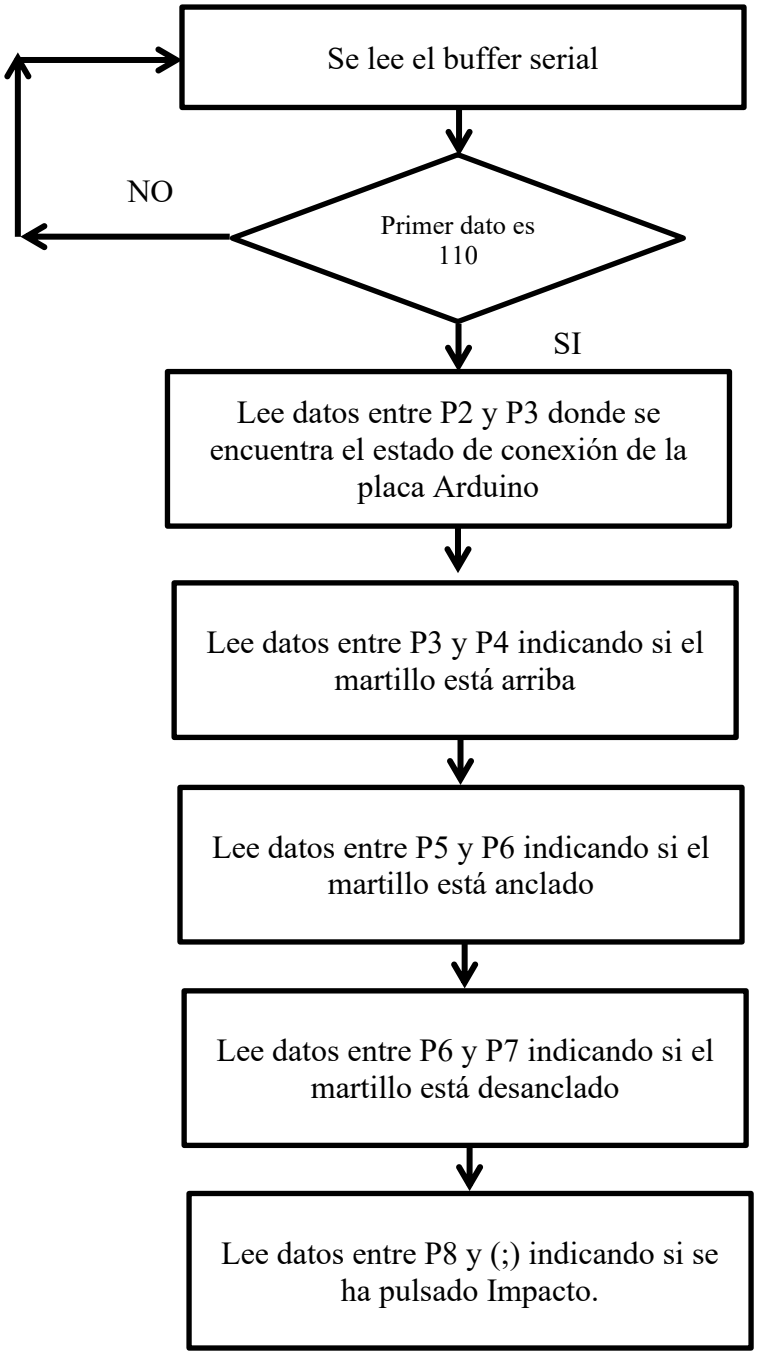

Figura 37 Algoritmo de recepción de entradas digitales.
Fuente: Autores.

Al pasar al diagrama de bloques de LabVIEW lo primero que se utiliza es el bloque NI-VISA Read (ver figura 38) la salida de información de este bloque pasa inicialmente a una secuencia de bloques que permiten identificar si la primera información es 110 entregando un

TRUE a la salida del comparador, si es verdadero se ejecutará entonces la información de la estructura CASE.

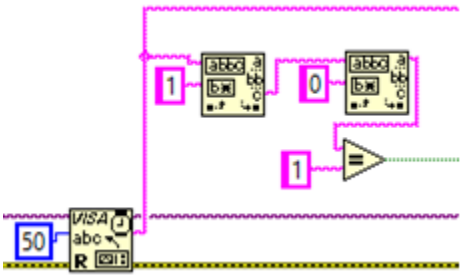

Figura 38 VISA Read e inicio de secuencia.
Fuente: Autores.

Para saber cuál es el estado de cada una de las señales digitales que para efectos de la simulación serán accionados por pulsadores, se utiliza la secuencia de la figura 39 aquí se toma la información que viene del NI-VISA Read para después iniciar a separarla por segmentos, si el valor recibido es cero esto indicará que la señal digital no está activada, si el valor recibido es uno indicara que está activado.

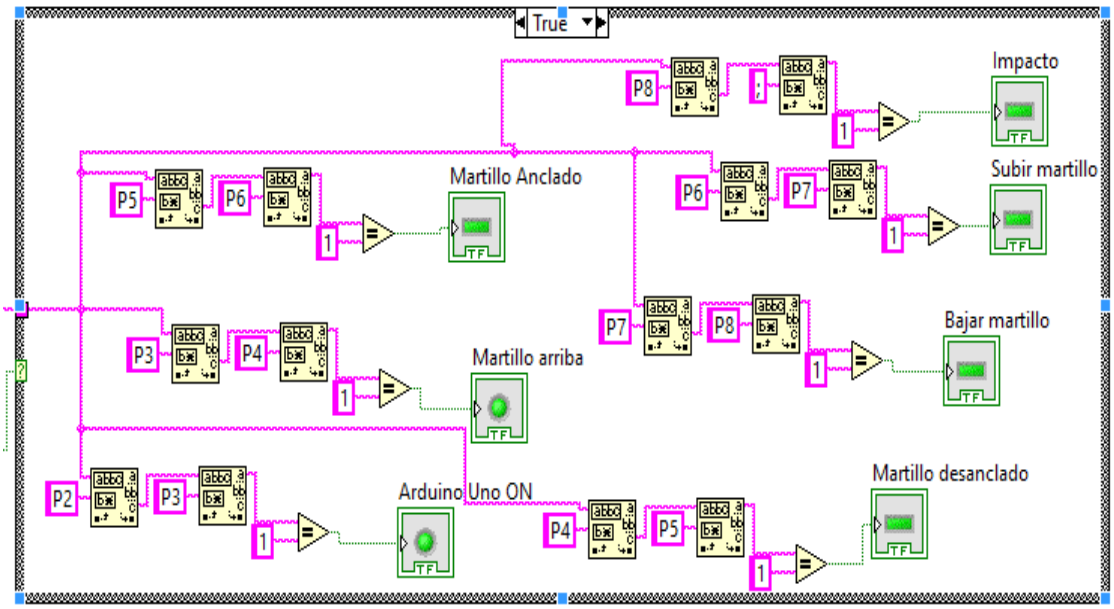

Figura 39 Información separada por segmentos para entradas digitales.
Fuente: Autores.

En cuanto a la visualización de los ángulos y la energía de impacto es importante inicialmente señalar que se podrán leer de forma local en la pantalla LCD y para el caso de LabVIEW el microcontrolador del Arduino Mega una vez estime estos valores los comunicara a través del puerto serial, en la figura 40 se aprecia que se utilizan los mismos bloques usados para leer las entradas digitales solo que en este caso no es necesario utilizar los comparadores y la información se visualiza directamente en indicadores tipo string.

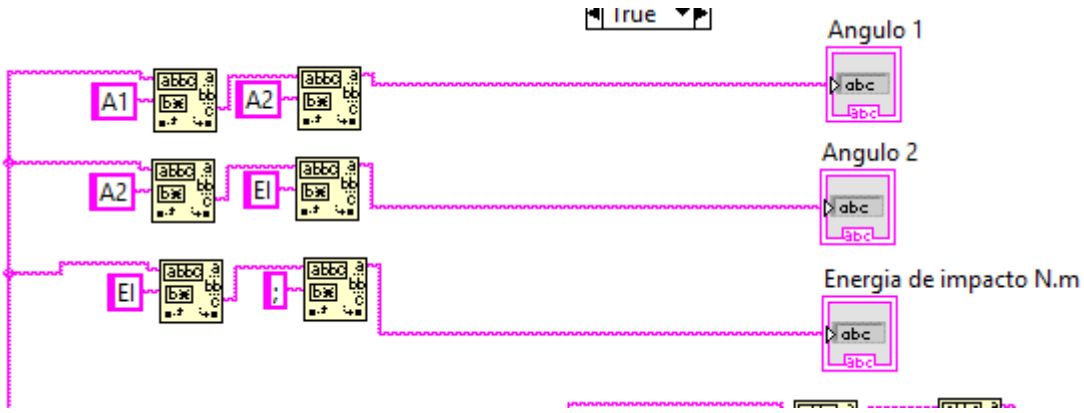

Figura 40 Información separada por segmentos para ángulos y energía de impacto.
Fuente: Autores.

6.2.2. Trasmisión de datos en LabVIEW con NI-VISA.

La lógica utilizada para enviar informacion desde LabVIEW hasta el microcontrolador se basa en enviar una serie de letras que llevarán a que el algoritmo accione los motores y ejecute la secuencia de subir el martillo y anclarlo. En la tabla 10 se describen las acciones que se ejecutan ahora utilizando las funciones NI-VISA cuando se pulsan los difentes botones.

Botón	Comando enviado
SUBIR MARTILLO	Envía la letra B
SOLTAR GUAYA	C
ANCLAR	D
IMPACTO	E
Habilitar sistema	A
DETENER MOTORES	F

Tabla 10 Comandos enviados por los diferentes botones.
Fuente: Autores.

En la figura 41 se presenta el bloque NI-VISA Write que permite enviar los comandos a través de puerto serial y que serán leídos por el microcontrolador, utilizando inicialmente un comparador booleano que detecta si se está pulsando el botón correspondiente a la acción que se desea ejecutar.

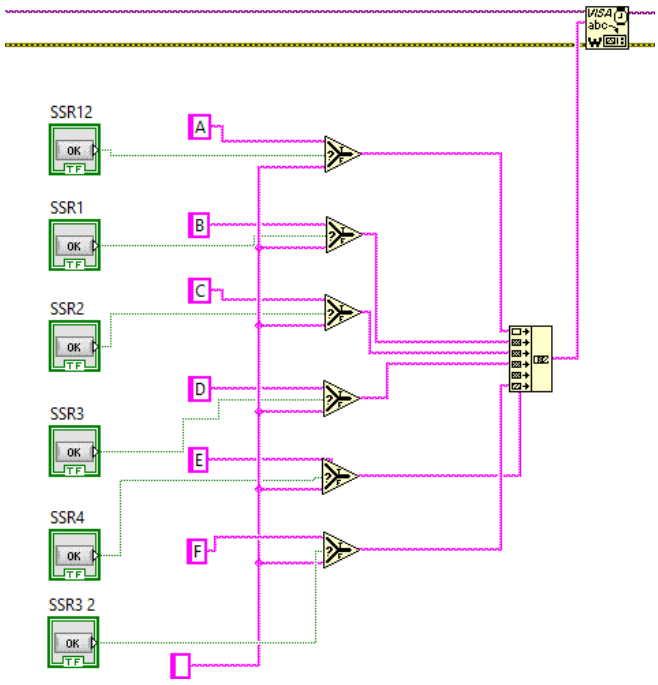

Figura 41 Enviando información por puerto serial.
Fuente: Autores.

7. Resultados y análisis.

Los resultados que se analizarán a continuación presentan que es posible modificar los algoritmos de LabVIEW y el microcontrolador para operar el péndulo Charpy utilizando funciones NI-VISA y mediante el sensor de posición angular que para el caso es un potenciómetro es posible determinar la posición del martillo y calcular la energía de impacto, sin embargo, se iniciará presentando la operación del péndulo desde el panel de control local.

7.1. Efectos en el circuito de control y potencia durante la operación desde el panel de control local.

El análisis que se entra a desarrollar en este apartado es el efecto que se tiene cuando se actúa sobre la muletilla que se plantea ubicar en el panel de control local (ver figura 27). La intención de esta prueba es mostrar que a través de las salidas digitales que actuarán sobre los relés de estado sólido será posible controlar el sentido de giro de los motores.

7.1.1. Pruebas para subir y bajar martillo.

Cuando se desea subir el martillo se manipula la muletilla como se indica en la figura 42, esto enviará una señal al pin 32 y tal como está programado el algoritmo ocasionará entonces que se active la salida del pin 41.

Figura 42 Posición de muletilla para subir martillo
Fuente: Autores.

El efecto que esta acción tendría dentro de la lógica de control cableada es, activar la salida 41 donde se encuentra conectado el SSR1. Como se observa en la figura 43 esto ocasionará que se active el relevador R5 que le indicará al motor que debe girar en el sentido que permite subir el martillo. Como se observa en la figura 44 también activará el contactor K1 que se encarga de energizar el motor, aquí es importante mencionar que cuando se alcanza la posición superior se activará el relevador R1 el cual al abrir el contacto que se encuentra en serie con R5 desenergizará K1 apagando de esta forma el motor de manera automática.

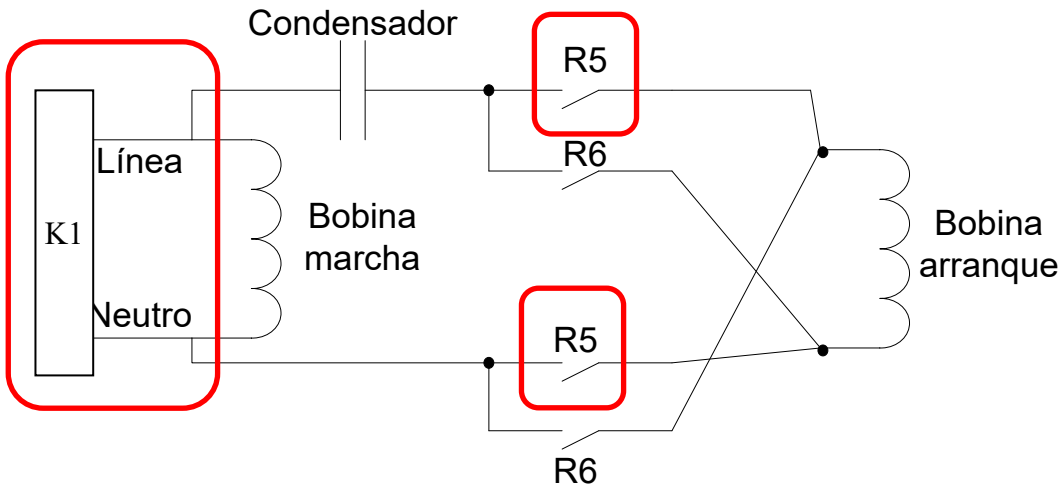

Figura 43 R5 sube martillo.
Fuente: Autores.

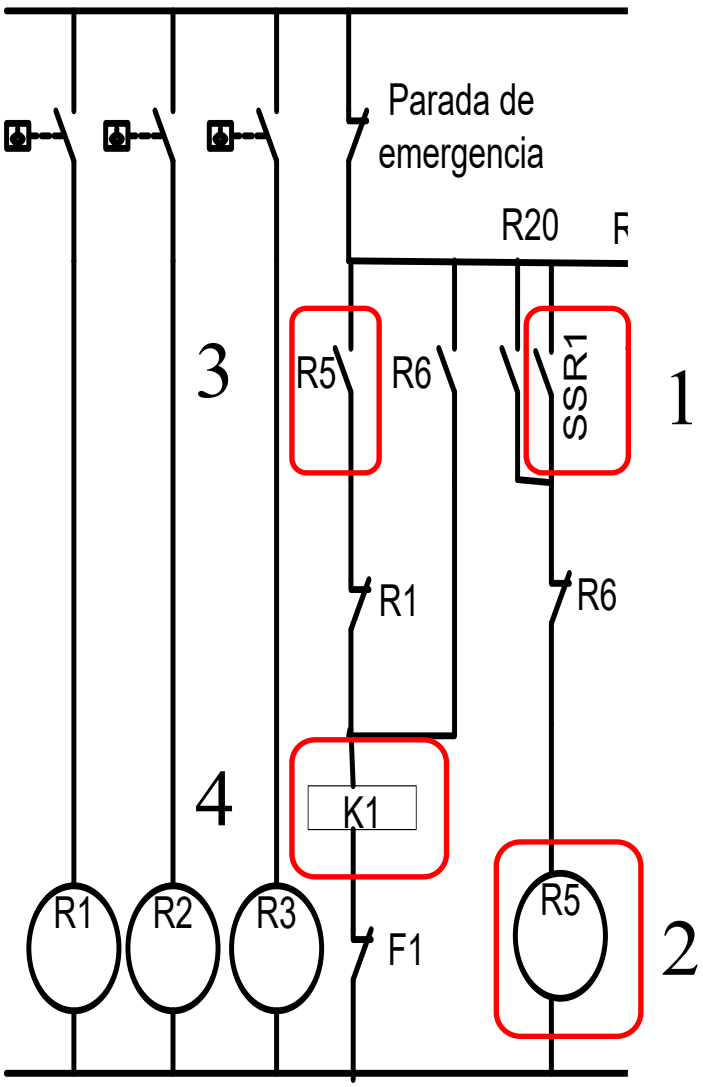

Figura 44 Relés y contactores activados para subir martillo
Fuente: Autores.

Si se llega a desear bajar el martillo o simplemente soltar la guaya para iniciar otra prueba se accionará la muletilla como se indica en la figura 45, en esta ocasión se activará la salida 42 que activa el SSR2 que se encargará de energizar R6 y este energizará K1.

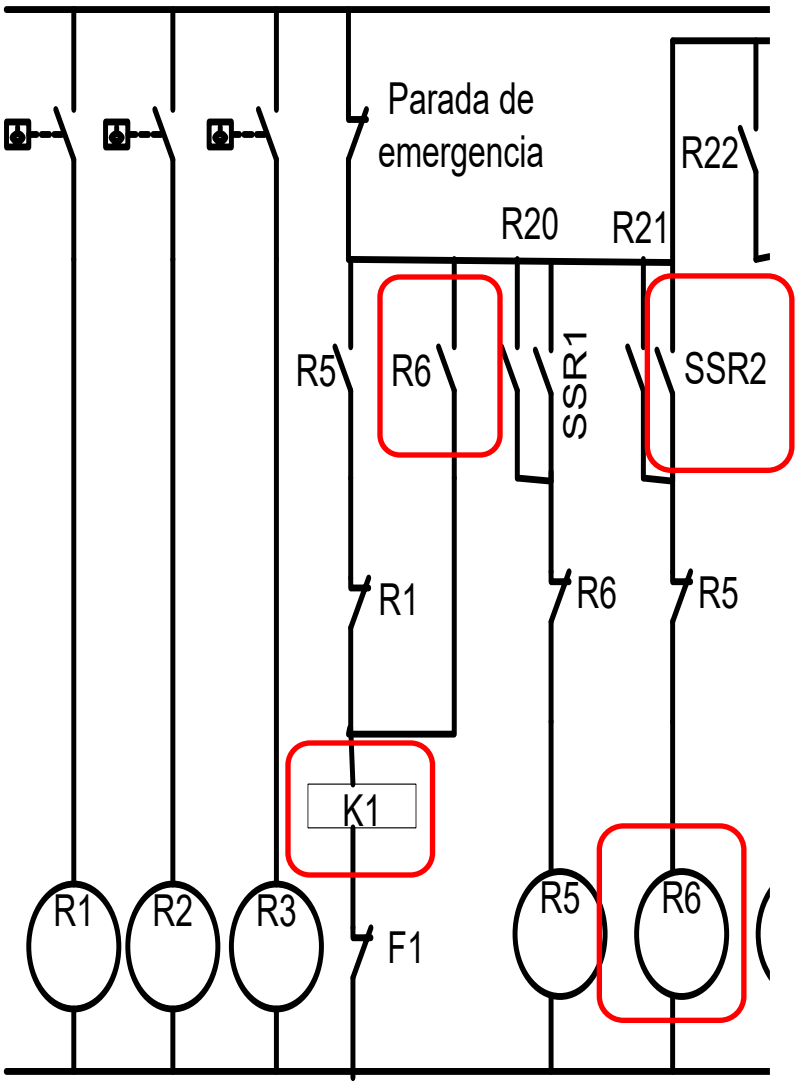

Figura 45 Relés y contactores activados para bajar martillo o soltar guaya.
Fuente: Autores.

7.1.2. Anclaje y desanclaje del martillo, acciones en la lógica de control cableado

Si se desea anclar el martillo se activará el relevador R7 y si se desea realizar el impacto una vez este anclado se activará R8 y para energizar el motor se activará K2 (ver figura 46).

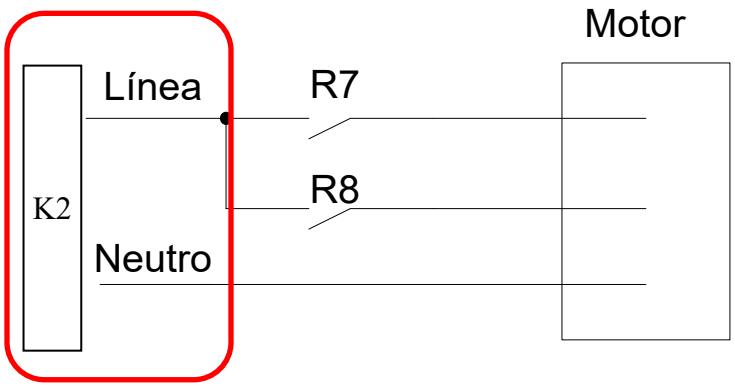

Figura 46 R7 ancla el martillo.
Fuente: Autores.

Para tener de forma ordenada cuales son los pasos a seguir cuando se realiza el anclaje del martillo se deben tener en cuenta las siguientes apreciaciones las cuales se presentan en la figura 47.

1. Cuando el martillo alcanza la posición superior se activa R1.
2. Como el martillo está anclado se encuentra activado el relevador R3.
3. El algoritmo activará la salida donde está conectado el SSR3 y esto traerá como resultado que se active R7 y aquí se inicia el proceso de anclaje.
4. Cuando el martillo se encuentra anclado se activará R2 y esto desactivará R7 apagando el motor.
5. Una vez se esté seguro de realizar el impacto activará la salida donde esta conectado el SSR4 y esto activara R8.
6. Cuando el sistema de anclaje active R3 el motor se apagará automáticamente.

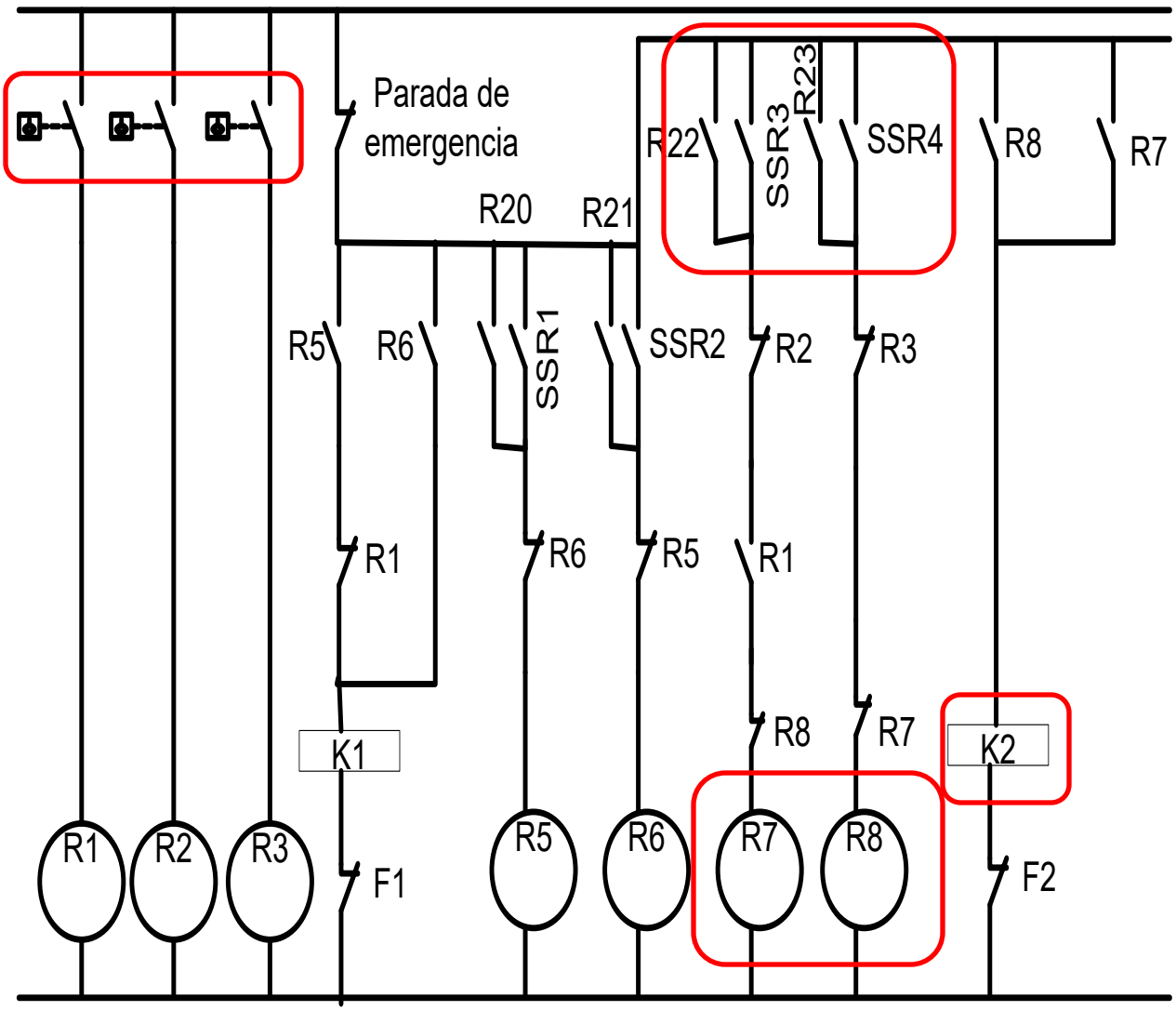

Figura 47 Relés y contactores activados para anclar y soltar martillo
Fuente: Autores.

7.1.3. Visualización local y en LabVIEW de sensores de posición.

Cuando se habla de sensores de posición se hace referencia a los que indican cual es la ubicación del martillo arriba, anclado o desanclado, para ello en Proteus se ubicaron tres pulsadores que se conectan en los pines 3, 30 y 31; aquí se hace referencia a PM como Posición Martillo (1: arriba, 0 abajo), MA Martillo anclado (1: anclado), MD Martillo desanclado (1: Desanclado) respectivamente. Como se ha mencionado a lo largo del documento en la pantalla

LCD se visualizarán estos estados para que el operador del péndulo tenga presente cuales acciones se están ejecutando, todo esto se observa en la figura 48.

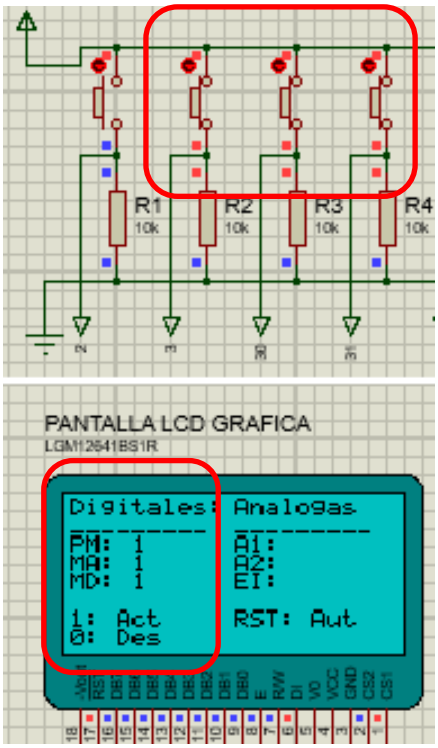

Figura 48 Visualización de sensores de posición en LCD
Fuente: Autores.

De manera paralela a la información presentada en la pantalla LCD en LabVIEW también es posible visualizar el estado de los pulsadores y finales de carrera mejorando de esta forma la operación del péndulo Charpy, (ver figura 49 y 50).

Figura 49 Visualización de sensores de posición en LabVIEW
Fuente: Autores.

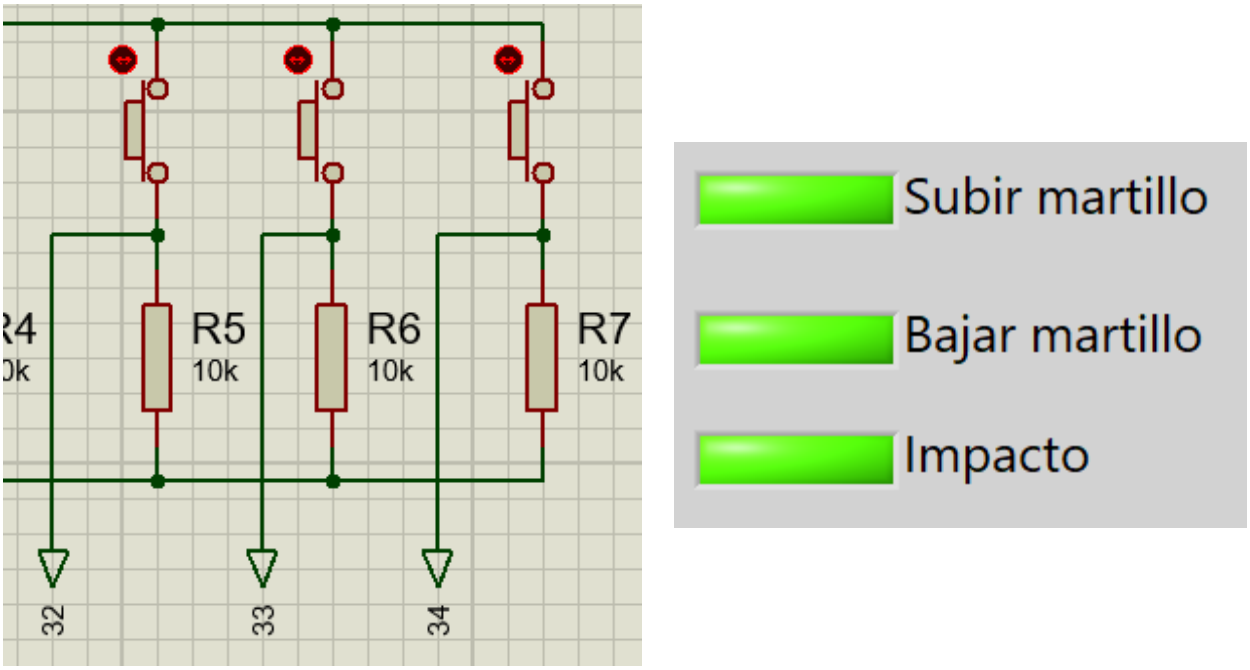

Figura 50 Visualización del estado de los pulsadores y finales de carrera en PROTEUS y Visualización del estado de la operación en LabVIEW
Fuente: Autores.

Esta prueba permite mostrar inicialmente que el algoritmo que realiza la lectura de las entradas digitales en el microcontrolador tal como se había establecido, adicional a esto el algoritmo en LabVIEW que captura la información presenta estos resultados en el panel frontal, entonces empieza a ser evidente que el uso de las funciones o bloques NI-VISA permiten automatizar el péndulo Charpy utilizando LabVIEW.

7.2. Simulación de secuencia de ascenso y anclaje de martillo.

Después de ver que con los algoritmos desarrollados tanto en LabVIEW como en el microcontrolador permiten leer estado de los finales de carrera, ahora se entra a emular el ascenso del martillo y la secuencia de anclaje. Lo primero que se hará entonces es activar el pulsador 32 en Proteus y una condición inicial será que el martillo este desanclado por esta razón

se activa también el pulsador 30, esto ocasionara que se active el led conectado en la salida 41 (ver figura 51) la cual envía la señal al circuito de control cableado que permite subir el motor.

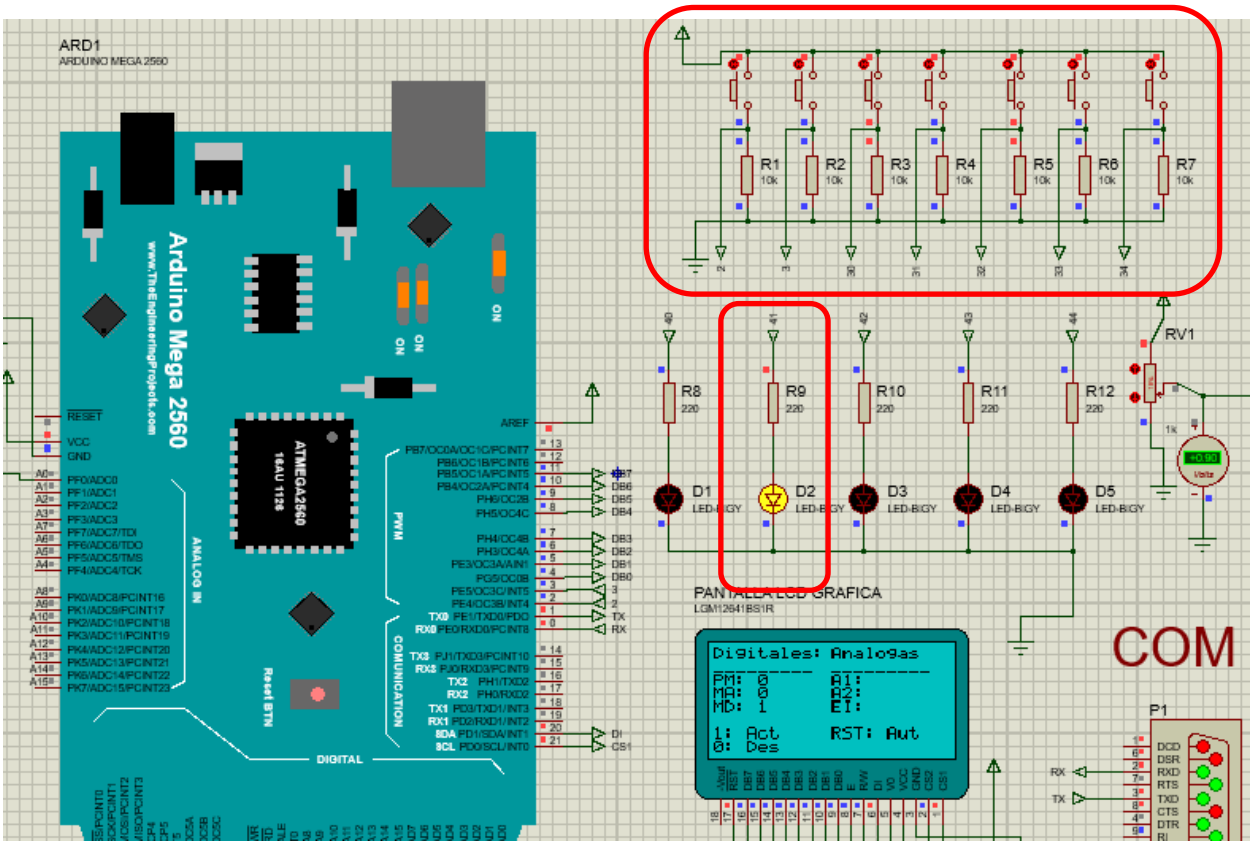

Figura 51 Simulación de subiendo martillo en Proteus.
Fuente: Autores.

Cuando se alcanza la posición superior se activa la entrada 3 y esto trae como resultado que se desactive la salida 41 es decir el martillo deja de subir, después de esto se activara la salida 43 (ver figura 52) que acciona el motor que realiza el anclaje del martillo y cuando se alcanza el final de carrera que activa la señal de pin 31 se apagara la salida 43 porque en este momento el martillo ya se encuentra anclado (ver figura 53).

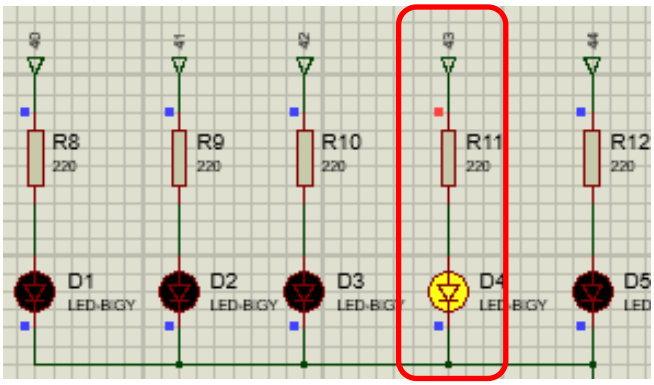

Figura 52 Salida digital que emula el anclaje del martillo.
Fuente: Autores.

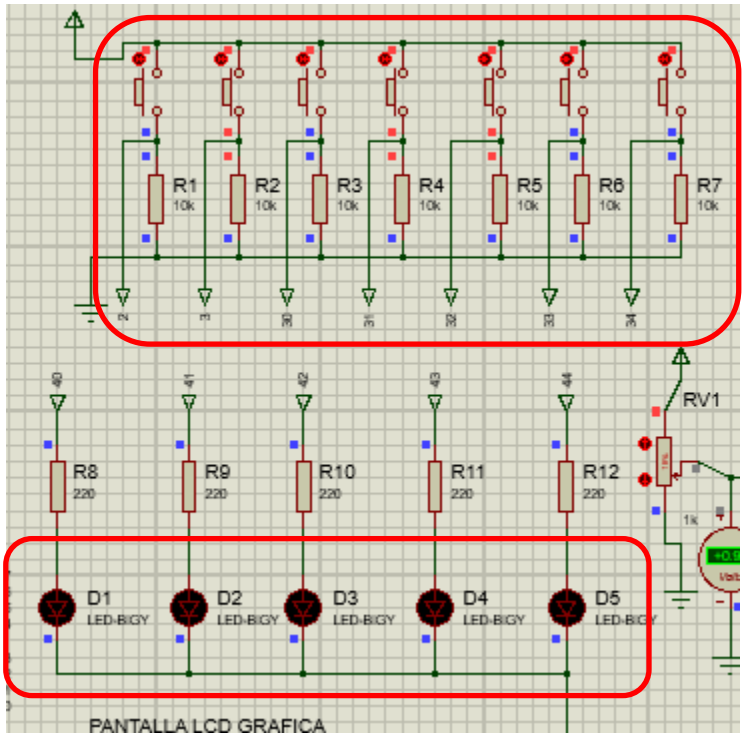

Figura 53 Simulación de martillo anclado.
Fuente: Autores.

Si se activa en este punto el pulsador impacto (pin 34) se activara la salida 44 (ver figura 54) que energizara el motor que desancla el martillo según el circuito de potencia que se identificó en el capítulo 2 de este documento, de esta forma es claro que la secuencia se ejecutará tal como se estableció según la lógica bajo la cual opera el péndulo.

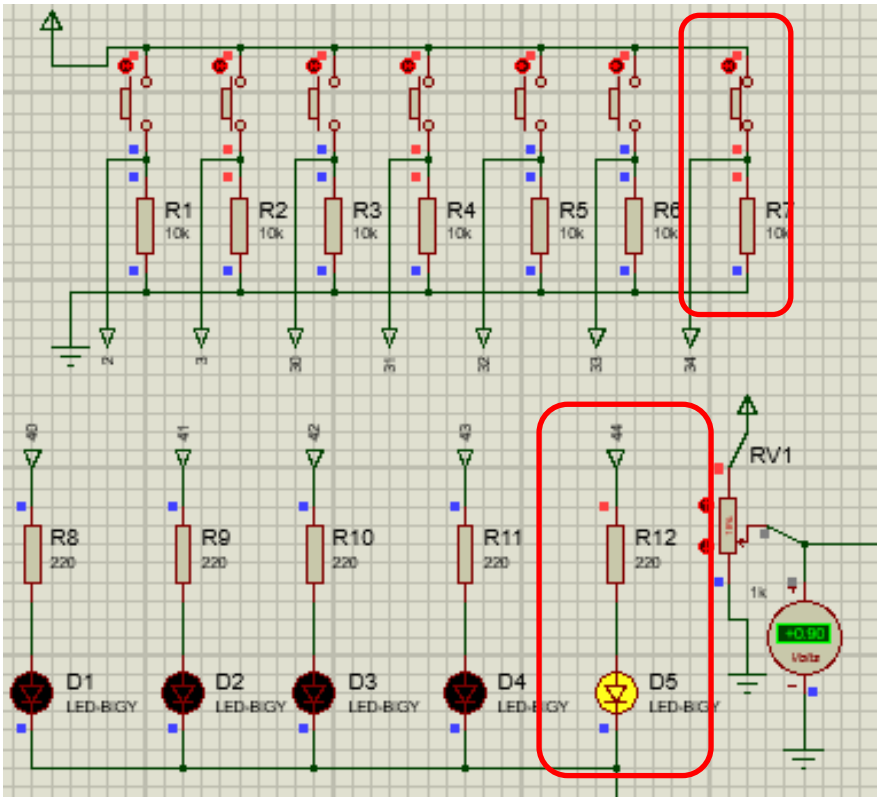

Figura 54 Simulación de impacto.
Fuente: Autores.

7.3. Estimación de ángulos y cálculo de energía de impacto.

La estimación de los ángulos y el cálculo de la energía de impacto busca únicamente mostrar que los algoritmos se ejecutan correctamente y que se visualizan los datos en la LCD gráfica como en LabVIEW, aquí no se entra a realizar pruebas con probetas. Para ello se realizarán los siguientes pasos.

- Es necesario simular el ascenso del martillo dado que es una condición de inicio para calcular la energía de impacto.
- Para iniciar se debe garantizar que el martillo este completamente vertical y ello se simulara manteniendo la salida de voltaje de potenciómetro en 2.5V (ver figura 55).

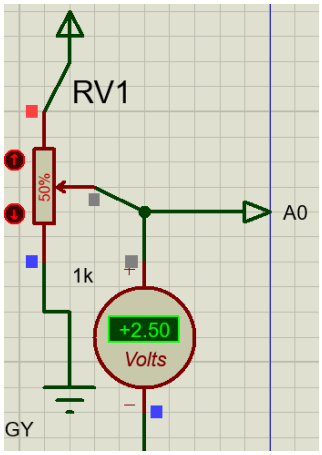

Figura 55 Martillo en posición vertical.
Fuente: Autores.

- Se pulsa la señal 30 y después de esto se empieza a aumentar el voltaje hasta alcanzar 4V que equivalen a 120°.
- Después de esto se da clic en el botón impacto e iniciara el descenso y para efecto de la simulación se lleva el voltaje a la salida del potenciómetro a 1V que equivale a105° aproximadamente, entonces el algoritmo que determina los ángulos y la energía de impacto se ejecuta y para este caso se visualizarán los datos en el panel LCD (ver figura 56) y en LabVIEW en la figura 57. W=30kg L=0.76m

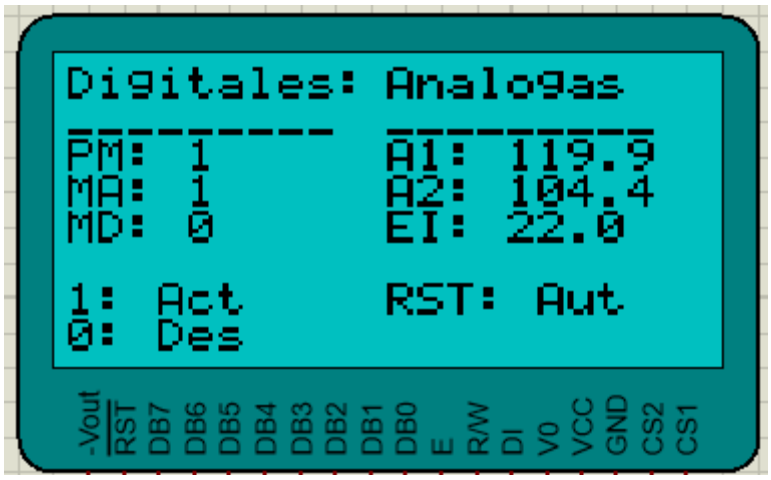

Figura 56 Visualización en LCD de ángulos y energía de impacto.
Fuente: Autores.

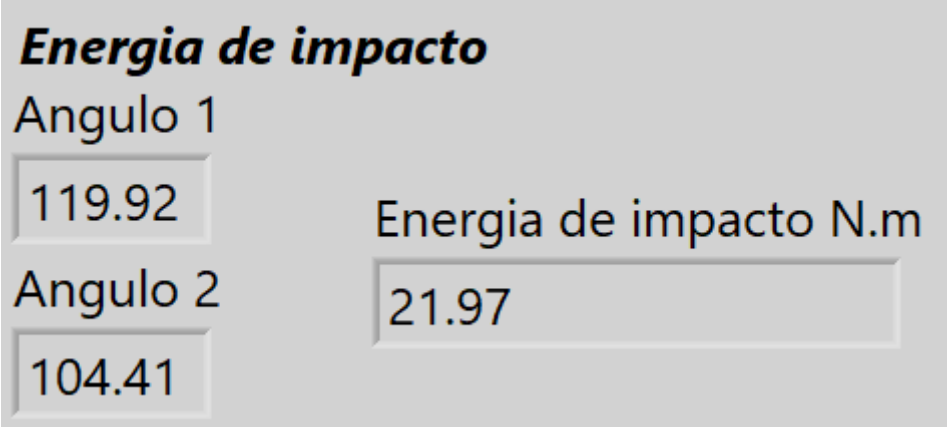

Figura 57 Visualización en LabVIEW de ángulos y energía de impacto.
Fuente: Autores.

8. Conclusiones.

Para establecer cuál es la mejor alternativa que permite determinar la posición del martillo se elabora una matriz de selección donde se comparan el giroscopio POLULO 2737, el acelerómetro MMA845 y un potenciómetro, se compararon estas opciones porque cada uno de ellos puede ser conectado a la placa Arduino mega dado que esta cuenta con periféricos como I2C, SPI y entradas análogas; de este análisis se obtuvo que la mejor opción es el potenciómetro ya que tuvo una ponderación de 0.89 adicionalmente puede ser conectado a una de las entradas análogas y puede ser acoplado directamente al eje en el cual se encuentra instalado actualmente el sistema de visualización local, de esta manera se logra mejorar la resolución de la medición siendo esta 0.12°, de esta manera se asciende a un 98.8% la eficiencia en la adquisición de los datos de los ángulos 1 y 2 para el cálculo automático de la energía de impacto.

En cuanto a la obtención de los ángulos 1 y 2 se logran obtener dos ecuaciones lineales donde es posible relacionar el voltaje de salida del potenciómetro con la inclinación angular del martillo, adicional a esto se plantea el algoritmo que permite determinar cuándo iniciar la lectura de los ángulos y como determinar los valores máximos de los mismos.

En LabVIEW se desarrolló un algoritmo que mantuvo la misma presentación en el panel frontal, agregando la visualización de todas las señales digitales y adicional a ello la indicación de los ángulos de inclinación y energía de impacto, esto fue posible porque en el diagrama de bloque se incluyeron los bloques NI-VISA que se comunican con el Arduino Mega el cual tiene un algoritmo que permite realizar la operación completa del péndulo.

Al realizar las simulaciones empleando Proteus es posible evidenciar que, se logra realizar la visualización local del estado de las señales digitales y la señal análoga que relaciona los ángulos 1 y 2, también se logra observar cómo interactúan las señales digitales que actúan

sobre los SSR que finalmente conmutan los relevadores o contactores de la lógica de control cableada para accionar el motor que eleva el martillo y lo ancla.

Bibliografía.

[1]. ASTM E23- 07ae1. Standad Test Methods for Notched BarnImpact Testing of Metallic Materials, 2007.

[2]. Jaimes, J & Jaimes D. (2016). Implementación mecánica de un equipo para pruebas de impacto por péndulo Charpy en el laboratorio de materiales de la universidad Antonio Nariño sede Bucaramanga.

[3]. DA Cruz Noruega, Influencia do raio de ponta do entalhe do tipo de carregamento e da microestrotura no proceso a fratura do aco estrutural ABNT-4340, p. 53.

[4]. DIETER, G.E. MEchanical Metallurgy, SI Metric Editio, McGraw-Hill, 1988, p. 265.

[5]. Manosalva D & Vásquez M. (2019). Implementación de un posicionador para el péndulo Charpy que hace parte del laboratorio de materiales de la Universidad Antonio Nariño sede Bucaramanga.

[6]. Mott, R. (2009). Diseño de elementos de máquinas. Editorial PEARSON, p. 266.

[7]. Askeland, D. R. (2004). Ciencia e Ingeniería de los Materiales. México: International Thomson.

[8]. Cortez, D & Jaramillo K. (2016). Diseño de un péndulo de impacto tipo Charpy para el laboratorio de materiales de la pontificia universidad Javeriana de Cali.

[9]. Stantad test methods for Notched bar impact testing of metallic materials ASTM 23.

[10]. Viracocha, C. (2014). Diseño y construcción de una máquina de ensayo de impacto Charpy, para el laboratorio de electromecánica de la universidad técnica de cotopaxi en el periodo 2014”

[11]. DOMINGO PEÑA, J., y otros. 2003. Introducción a los autómatas programables. Aragón: Editorial UOC, 2003.

[12]. K. Moore, A. Agur, and M. Moore, Anatomía con orientación clínica. Editorial Médica Panamericana Sa de, 2007. P. 22 – 25.

[13]. Guerrero, E. (2013). Instrumentación de un robot bípedo de 12 GDL: sensores de posición, presión e ingercial.

[14]. Arnold J, & Vieco, J. Determinacion de la tenacidad a la fractura por medio del ensayo charpy instrumentado en un acero A283 grado C envejecido en un ambiente de hidrogeno.

Anexo I. Código programado en el Arduino Mega.

```
#include <basicGLCD.h>

// definicion de salidas digitales
#define     SSR     40          // habilitar placa SSR
#define     SSR1    41          // Subir martillo
#define     SSR2    42          // Bajar martillo
#define     SSR3    43          // Anclar martillo
#define     SSR4    44          // impacto

// definicion de entradas digitales

#define     PIN3    3           // Martillo arriba
#define     PIN2    2           // Arduino uno conectado
#define     PIN30   30          // martillo desanclado
#define     PIN31   31          // martillo anclado
#define     PIN32   32          // subir martillo
#define     PIN33   33          // Bajar martillo
#define     PIN34   34          // impacto

//String A1;
//String A2;
//String EPS;

int         button2 =   0;
int         button3 =   0;
int         button30 =  0;
int         button31 =  0;
int         button32 =  0;
int         button33 =  0;
int         button34 =  0;
int         anclado=    0;
int         subiendo=   0;
int         dato=       0;
int         Serial_SSR1=  0;
int         Serial_SSR2=  0;
int         Serial_SSR3=  0;
int         Serial_SSR4=  0;
int         estado_pendulo= 0;
int         contador=   0;
int         inicio_prueba = 0;
int         fin_prueba  = 0;


float       voltaje     = 0;
float       voltaje2    = 0;
float       angulo1     = 0;
float       angulo2     = 0;
float       L           = 0.76;
float       W           = 30;
float       grados      =0;
float       EI          =0;
```

```
basicGLCD GLCD;

void setup() {

	Serial.begin(9600);			// velocidad de transmision

	pinMode(PIN3, INPUT);
	pinMode(PIN2, INPUT);
	pinMode(PIN30, INPUT);
	pinMode(PIN31, INPUT);
	pinMode(PIN32, INPUT);
	pinMode(PIN33, INPUT);
	pinMode(PIN34, INPUT);

	pinMode(SSR, OUTPUT);
	pinMode(SSR1, OUTPUT);
	pinMode(SSR2, OUTPUT);
	pinMode(SSR3, OUTPUT);
	pinMode(SSR4, OUTPUT);


	GLCD.setDataPins(4,5,6,7,8,9,10,11);
	GLCD.setRSPin(20);
	GLCD.setCSPins(21, 22);
	GLCD.setENPin(23);
	GLCD.setRWPin(24);
	GLCD.setRSTPin(25);
	while(GLCD.init() != 1) {
			Serial.println("You have to set all pins");
			return 0;
	}
 // Pantalla principal de LCD
	GLCD.setPage(LEFT);
	GLCD.setCursor(0, 0);
	GLCD.sendString("Digitales:");
	GLCD.setPage(LEFT);
	GLCD.setCursor(0, 1);
	GLCD.sendString("_________");
	GLCD.setCursor(0, 2);
	GLCD.sendString("PM:");
	GLCD.setCursor(0, 3);
	GLCD.sendString("MA:");
	GLCD.setCursor(0, 4);
	GLCD.sendString("MD:");
	GLCD.setCursor(0, 6);
	GLCD.sendString("1: Act");
	GLCD.setCursor(0, 7);
	GLCD.sendString("0: Des");

	GLCD.setPage(RIGHT);
	GLCD.setCursor(0, 0);
	GLCD.sendString("Analogas");
	GLCD.setPage(RIGHT);
	GLCD.setCursor(0, 1);
```

```
        GLCD.sendString("_________");
        GLCD.setCursor(0, 2);
        GLCD.sendString("A1:");
        GLCD.setCursor(0, 3);
        GLCD.sendString("A2:");
        GLCD.setCursor(0, 4);
        GLCD.sendString("EI:");
        GLCD.setCursor(0, 6);
        GLCD.sendString("RST: Aut");                    // El reset se realiza de forma automatica cuando el martillo
        vuelve a alcanzar la posicion superior.

}

void loop() {

        button2 = digitalRead(PIN2);
        button3 = digitalRead(PIN3);
        button30 = digitalRead(PIN30);          // martillo desanclado
        button31 = digitalRead(PIN31);          // MArtillo anclado
        button32 = digitalRead(PIN32);          // subir martilo
        button33 = digitalRead(PIN33);          // bajar martillo
        button34 = digitalRead(PIN34);          // impacto

    if (button32 == HIGH or Serial_SSR1==1  and button3 == LOW) {

                                digitalWrite(SSR1, HIGH);
                                subiendo=1;
                                EI=0;
                                grados=0;
                                angulo1=0;
                                angulo2=0;
                                fin_prueba=0;

     }

    if (subiendo == 1 and button3 == HIGH and button30 == HIGH) {
                                digitalWrite(SSR1, LOW);
                                delay(1000);
                                digitalWrite(SSR3, HIGH);
     }

     if(button31 == HIGH){
                    digitalWrite(SSR3, LOW);
                    subiendo=0;


       }

     if(button31 == HIGH and button34 == HIGH or Serial_SSR4 ==1){
                                 digitalWrite(SSR4, HIGH);
                                 anclado=1;
                                 digitalWrite(SSR3, LOW);

     }

     if(anclado ==1 and button30 == HIGH){
```

```
                        digitalWrite(SSR4, LOW);
                        anclado=0;
                        Serial_SSR1 =0;
                        Serial_SSR2 =0;
                        Serial_SSR3 =0;
                        Serial_SSR4 =0;

    }


    if(button31 == HIGH){
                GLCD.setPage(LEFT);
                GLCD.setCursor(4, 3);
                GLCD.sendString("1");
    }

    if(button31 == LOW){
                GLCD.setPage(LEFT);
                GLCD.setCursor(4, 3);
                GLCD.sendString("0");
    }

    if(button30 == HIGH){

      GLCD.setPage(LEFT);
      GLCD.setCursor(4, 4);
      GLCD.sendString("1");

    }

    if(button30 == LOW){

      GLCD.setPage(LEFT);
      GLCD.setCursor(4, 4);
      GLCD.sendString("0");

    }

    if(button3 == LOW){

      GLCD.setPage(LEFT);
      GLCD.setCursor(4, 2);
      GLCD.sendString("0");

    }

    if(button3 == HIGH){

      GLCD.setPage(LEFT);
      GLCD.setCursor(4, 2);
      GLCD.sendString("1");
      digitalWrite(SSR1, LOW);
    }

//recibir datos puerto serial
```

```
if(Serial.available())

        {
        dato=Serial.read();
        switch (dato){

                case 'A':           // Habilita sistema Activa pin 40
                digitalWrite(SSR, HIGH);
                break;

                case 'B':           // Sube martillo activa pin 41
                Serial_SSR1=1;
                break;

                case 'C':           // Bajar martillo activa pin 42
                digitalWrite(SSR2, HIGH);
                break;

                case 'F':           // Bajar martillo activa pin 42
                digitalWrite(SSR1, LOW);
                digitalWrite(SSR2, LOW);
                digitalWrite(SSR3, LOW);
                digitalWrite(SSR4, LOW);
                break;


                case 'D':           // anclar martillo activa pin 43
                digitalWrite(SSR3, HIGH);
                break;

                case 'E':           // desanclar martillo activa pin 44
                Serial_SSR4=1;
                break;


                }

        }

if(contador>25){
        Serial.print("110");
        Serial.print("P2");
        Serial.print(button2);
        Serial.print("P3");
        Serial.print(button3);
        Serial.print("P4");
        Serial.print(button30);
        Serial.print("P5");
        Serial.print(button31);
        Serial.print("P6");
        Serial.print(button32);
        Serial.print("P7");
        Serial.print(button33);
        Serial.print("P8");
        Serial.print(button34);
        Serial.println(";");
```

```
contador=0;
}

delay(10);
contador=contador+1;

if(subiendo==1){

          voltaje = (analogRead(A0)*5.0/1024.0);          // la multiplicacion se hace para convertir el numero a
        un valor entre 0-5V
          if(button3==1){                          // cuando esta condicione
                    angulo1 = 80.0*(voltaje - 2.5);
                    inicio_prueba=1;
          }
}

voltaje = (analogRead(A0)*5.0/1024.0);
delay(50);

if(inicio_prueba ==1 and voltaje < 2.5){

                    voltaje2 = (analogRead(A0)*5.0/1024.0);
                    if(voltaje2 > voltaje){
                              inicio_prueba=0;
                              angulo2 = 180.0 - (72*voltaje2);
                              fin_prueba=1;
                    }


}

if(fin_prueba==1){                                   // aqui se calcula la energia de impacto
          grados = angulo2 - angulo1;
          EI = W*L*cos((grados*PI)/180.0);
  String A1;    A1= String(angulo1, 1);
  String A2;    A2= String(angulo2, 1);
  String EPS;   EPS= String(EI, 1);
          GLCD.setPage(RIGHT);
          GLCD.setCursor(4, 2);
          GLCD.sendString(A1);
          GLCD.setCursor(4, 3);
          GLCD.sendString(A2);
          GLCD.setCursor(4, 4);
          GLCD.sendString(EPS);
}
 }
```