



**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL
CUSCO**

ESCUELA DE POSGRADO

MAESTRÍA EN CIENCIAS MENCIÓN FÍSICA

TESIS

**DISEÑO Y CONSTRUCCIÓN DE MÓDULOS
AUTOMATIZADOS, BASADOS EN ARDUINO PARA EL
DESARROLLO DE EXPERIMENTOS DE FÍSICA EN MECÁNICA
Y ELECTRICIDAD**

**PARA OPTAR AL GRADO ACADÉMICO DE MAESTRO EN
CIENCIAS MENCIÓN FÍSICA**

AUTOR

Br. YURI MORALES GARCÍA

ASESOR:

Mg. HENRY HOLGUÍN GALLEGOS

CÓDIGO ORCID: 0000-0003-0459-8700

CO ASESOR:

Dr. PASTOR RAÚL CHURA SERRANO

CUSCO - PERÚ

2 024

INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, asesor del trabajo de investigación/tesis titulado: DISEÑO Y CONSTRUCCIÓN DE MÓDULOS AUTOMATIZADOS, BASADOS EN ARDUINO PARA EL DESARROLLO DE EXPERIMENTOS DE FÍSICA EN MECÁNICA Y ELECTRICIDAD.

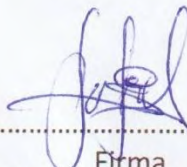
presentado por: YURI MORALES GARCÍA con Nro. de DNI: 23916846, para optar el título profesional/grado académico de MAESTRO EN CIENCIAS MENCIÓN FÍSICA. Informo que el trabajo de investigación ha sido sometido a revisión por 01 veces, mediante el Software Antiplagio, conforme al Art. 6° del *Reglamento para Uso de Sistema Antiplagio de la UNSAAC* y de la evaluación de originalidad se tiene un porcentaje de 08 % (Ocho por ciento)

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No se considera plagio.	X
Del 11 al 30 %	Devolver al usuario para las correcciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y **adjunto** la primera hoja del reporte del Sistema Antiplagio.

Cusco, 23 de octubre de 2024



Firma

Post firma.....HENRY HOLGUIN GALLEGOS

Nro. de DNI 23979948

ORCID del Asesor 0000-0003-0459-8700

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio:
<https://unsaac.turnitin.com/viewer/submissions/oid:27259:397284571?locale=es-MX>

NOMBRE DEL TRABAJO

DISEÑO Y CONSTRUCCIÓN DE MÓDULOS AUTOMATIZADOS, BASADOS EN ARDUINO PARA EL DESARROLLO DE EXPERIMENTOS DE FÍSICA EN MECÁNICA Y ELECTRICIDAD

AUTOR

Yuri Morales García

RECUENTO DE PALABRAS

30442 Words

RECUENTO DE CARACTERES

167578 Characters

RECUENTO DE PÁGINAS

179 Pages

TAMAÑO DEL ARCHIVO

43.3MB

FECHA DE ENTREGA

Oct 22, 2024 2:01 PM GMT-5

FECHA DEL INFORME

Oct 22, 2024 2:07 PM GMT-5

● 8% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 7% Base de datos de Internet
- Base de datos de Crossref
- 6% Base de datos de trabajos entregados
- 0% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Bloques de texto excluidos manualmente



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
ESCUELA DE POSGRADO

INFORME DE LEVANTAMIENTO DE OBSERVACIONES A TESIS

Dra. NELLY AYDE CAVERO TORRE, Directora (e) General de la Escuela de Posgrado, nos dirigimos a usted en condición de integrantes del jurado evaluador de la tesis intitulada **DISEÑO Y CONSTRUCCIÓN DE MÓDULOS AUTOMATIZADOS, BASADOS EN ARDUINO PARA EL DESARROLLO DE EXPERIMENTOS DE FÍSICA EN MÉCANICA Y ELECTRICIDAD** de la Br. BR. YURI MORALES GARCIA. Hacemos de su conocimiento que el (la) sustentante ha cumplido con el levantamiento de las observaciones realizadas por el Jurado el día **DIECINUEVE DE NOVIEMBRE DE 2024**.

Es todo cuanto informamos a usted fin de que se prosiga con los trámites para el otorgamiento del grado académico de MAESTRO EN CIENCIAS MENCIÓN FÍSICA.

Cusco, 04 diciembre 2024

MGT. DAVID TEOFILO VERA VICTORIA
Primer Replicante

MGT. GONZALO SUAREZ SANCHEZ
Segundo Replicante

DRA. YENY MARITZA ACCOSTUPA QUISPE
Primer Dictaminante

MGT. GIOVANNA GUZMAN CACERES
Segundo Dictaminante

PRESENTACIÓN.

Señor Director de la Escuela de Postgrado de la Universidad Nacional de San Antonio Abad del Cusco.

Señores miembros del Jurado:

En cumplimiento del Reglamento de Grados de la Escuela de Postgrado de la Universidad Nacional de San Antonio Abad del Cusco, Maestría en Ciencias con Mención en Física, pongo a vuestra consideración el presente trabajo de tesis intitulado “DISEÑO Y CONSTRUCCIÓN DE MÓDULOS AUTOMATIZADOS, BASADOS EN ARDUINO PARA EL DESARROLLO DE EXPERIMENTOS DE FÍSICA EN MECÁNICA Y ELECTRICIDAD”, para optar al Grado Académico de Maestro en Ciencias, mención Física.

En el presente trabajo, se han diseñado y construido módulos funcionales para automatizar experimentos de Física en Mecánica y Electricidad. Estos módulos pueden ser utilizados tanto en universidades como en colegios de nivel secundario, utilizan tecnología actual basada en microprocesadores, sensores, comunicación inalámbrica y hacen uso de dispositivos portátiles como Tablets, Smartphones, etc. con sistema operativo Android.

Como resultado de este trabajo se presentan módulos consistentes en sensores ópticos (fotopuertas) y un sensor piezoeléctrico, sistema de control (*Interface*), para los experimentos de Mecánica y para los experimentos de Electricidad y un software, además de la parte mecánica como son dos soportes universales, abrazaderas y varillas de sujeción. Para la prueba de estos módulos, se desarrollaron los experimentos de caída libre, péndulo simple, movimiento rectilíneo uniforme, ley de Ohm y Carga y descarga de un capacitor

AGRADECIMIENTO.

Quiero expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de esta tesis. A mi asesor y co asesor, por su invaluable orientación y apoyo a lo largo de este proceso. A mi familia, a mi esposa Marilú y a mis dos hijas, Gabriela y Giuliana, por su amor incondicional y por motivarme y animarme en esos momentos de incertidumbre y flaqueza. A mis hermanos, amigos y colegas de DAF, quienes me han brindado su apoyo y compartido sus conocimientos. Finalmente, quiero agradecer a mi padre y a mi madre por todo lo que pudieron brindarme; aunque ya no estén presentes físicamente, siempre están en mis pensamientos y acciones. Sin su ayuda, este trabajo no habría sido posible.

ÍNDICE

PRESENTACIÓN.....	i
AGRADECIMIENTO.....	ii
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE TABLAS.....	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN.....	xiv
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....	1
1.1. SITUACIÓN PROBLEMÁTICA.....	1
1.2. FORMULACIÓN DEL PROBLEMA OBJETO DE LA INVESTIGACIÓN.....	3
1.2.1. Problema General.....	3
1.2.2. Problemas Específicos.....	4
1.3. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN.....	4
1.4. OBJETIVOS DE LA INVESTIGACIÓN.....	7
1.4.1. Objetivo general.....	7
1.4.2. Objetivos específicos.....	7
CAPÍTULO II: MARCO TEÓRICO CONCEPTUAL.....	8
2.1. ANTECEDENTES DE LA INVESTIGACIÓN.....	8
2.1.1. Antecedentes Internacionales.....	8
2.1.2. Antecedentes Nacionales.....	10
2.1.2. Antecedentes Locales.....	12
2.2. BASES TEÓRICAS.....	12
2.2.1. Mecánica Clásica.....	12
2.2.1.1. Movimiento Rectilíneo Uniforme.....	13
2.2.1.2. Caída Libre.....	15
2.2.1.3. Péndulo Simple.....	17
2.2.2. Electricidad.....	20
2.2.2.1. Ley de Ohm.....	22
2.2.2.2. Constante de Tiempo de un Circuito RC.....	24

2.2.3. Plataformas de Desarrollo	26
2.2.3.1. Ordenador Personal (PC)	26
2.2.3.2. Teléfono Inteligente	27
2.2.3.3. Android	28
2.2.4. Hardware	31
2.2.4.1. Arduino	31
2.2.4.2. Sensores	38
2.2.4.3. Actuadores.....	38
2.2.5. Software	39
2.2.5.1. App Inventor	39
2.2.5.2. Java Script y HTML.....	43
2.3. MARCO CONCEPTUAL – DEFINICIÓN DE TÉRMINOS BÁSICOS.....	45
2.4. HIPÓTESIS	50
2.4.1. Hipótesis General	50
2.4.2. Hipótesis Específicas.....	50
2.5. IDENTIFICACIÓN DE VARIABLES E INDICADORES	50
2.6. OPERACIONALIZACIÓN DE LAS VARIABLES	50
 CAPÍTULO III: METODOLOGÍA	 51
3.1. TIPO Y DISEÑO DE INVESTIGACIÓN.....	51
3.2. UNIDAD DE ANÁLISIS	57
3.3. POBLACIÓN DE ESTUDIO	57
 CAPITULO IV: RESULTADOS Y DISCUSIÓN	 58
4.1. DISEÑO E IMPLEMENTACIÓN DE LOS ELEMENTOS DE CONTROL Y DETECCIÓN PARA LOS MÓDULOS.....	58
4.1.1. Actuadores.	58
4.1.2. Sensores.....	60
4.1.3. Carrito para el M.R.U.....	66
4.1.4. Interface.....	70
4.1.5. Otros componentes	77
4.2. DESARROLLO DEL SOFTWARE.....	80

4.2.1. Programación de la placa Arduino.	80
4.2.2. Programación en APP Inventor.	86
4.2.3. Graficas con CHARTJS.....	99
4.3. DESARROLLO DE EXPERIMENTOS DE PRUEBA	103
4.3.1. Ejecución del Software de inicio.....	103
4.3.2. Configuración Inicial del Aplicativo.....	104
4.3.3. Experimento de Caída Libre	108
4.3.3.1. Montaje del Equipo para Caída Libre	108
4.3.3.2. Aplicación para Caída Libre	110
4.3.4. Experimento de péndulo simple	113
4.3.4.1. Montaje del Equipo para Péndulo Simple.....	113
4.3.4.2. Aplicación para Péndulo Simple.....	114
4.3.5. Experimento del Movimiento Rectilíneo Uniforme (MRU)	118
4.3.5.1. Montaje del Equipo para MRU	118
4.3.5.2. Aplicación para el MRU	118
4.3.6. Experimento sobre la ley de Ohm	121
4.3.6.1. Montaje del Equipo para los Experimentos de Electricidad	121
4.3.6.2. Aplicación para la Ley de Ohm.....	121
4.3.7. Experimento sobre la Carga de un Capacitor	123
4.3.7.1. Montaje del Equipo para la Carga de un Capacitor	123
4.3.7.2. Aplicación para la Carga de un Capacitor.....	124
4.3.8. Experimento sobre la descarga de un capacitor.....	126
4.3.8.1. Montaje del Equipo para la Descarga de un Capacitor	126
4.3.8.2. Aplicación para la Descarga de un Capacitor	127
4.4. TRATAMIENTO DE LOS DATOS PARA EL CÁLCULO DE LOS PARÁMETROS DE LOS EXPERIMENTOS DEL PRESENTE TRABAJO.....	129
4.5. RESULTADOS OBTENIDOS.....	132
4.6. PRESENTACIÓN DE RESULTADOS	145
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES.....	154
CONCLUSIONES.....	154
RECOMENDACIONES.....	155
PROPUESTA PARA LA SOLUCIÓN DEL PROBLEMA.....	156

COSTOS DE IMPLEMENTACIÓN DE LA PROPUESTA	159
BENEFICIOS QUE APORTA LA PROPUESTA.....	160
REFERENCIAS BIBLIOGRÁFICAS	163
ANEXOS	1
ANEXO A. MATRIZ DE CONSISTENCIA	1
ANEXO B. SOFTWARE DE DISEÑO ELECTRÓNICO	2
ANEXO C. COTIZACIÓN DE EQUIPOS DE LABORATORIO DE LA EMPRESA PASCO SCIENTIFIC	3
ANEXO D. MANUAL DE USUARIO	4

ÍNDICE DE FIGURAS

Figura 1. <i>Caída Libre de una Esfera y una Pluma</i>	16
Figura 2. <i>Péndulo Simple</i>	18
Figura 3. <i>Péndulo Simple Idealizado</i>	18
Figura 4. <i>Grafica de la $V=f(I)$ para un Material Óhmico y no Óhmico</i>	24
Figura 5. <i>Curvas de Carga y Descarga de un Capacitor</i>	25
Figura 6. <i>Arduino Uno</i>	33
Figura 7. <i>Arduino Pro</i>	33
Figura 8. <i>Arduino Pro Mini</i>	34
Figura 9. <i>Leonardo</i>	34
Figura 10. <i>Micro</i>	34
Figura 11. <i>Mega</i>	35
Figura 12. <i>Mega ADK</i>	35
Figura 13. <i>Due</i>	35
Figura 14. <i>Ethernet</i>	36
Figura 15. <i>LilyPad</i>	36
Figura 16. <i>Esplora</i>	36
Figura 17. <i>Gestor de Proyectos App Inventor</i>	40
Figura 18. Diseñador de Proyectos App Inventor	40
Figura 19. <i>Herramienta blocks App Inventor</i>	41
Figura 20. <i>Conexión Wifi</i>	42
Figura 21. <i>Emulador</i>	42
Figura 22. <i>Conexión mediante USB</i>	43
Figura 23. <i>a) Solenoides b) Roseta de Red Rj45, c) Conexiones internas del actuador</i>	59
Figura 24. <i>Vista del actuador construido</i>	59
Figura 25. <i>Definición de pulsos</i>	61
Figura 26. <i>Circuito esquemático del sensor fotopuerta</i>	61
Figura 27. <i>Diseño del circuito impreso de la fotopuerta</i>	62
Figura 28. <i>Diseño CAD de la carcasa para el sensor fotopuerta</i>	62
Figura 29. <i>Diseño CAD para la Construcción de la Carcasa de la Fotopuerta</i>	63
Figura 30. <i>Fotopuerta construida</i>	63
Figura 31. <i>Efecto del uso del Schmitt Trigger (B) Uso de un Comparador (A)</i>	64
Figura 32. <i>Diseño del PCB del Adaptar de la Señal del Piezoeléctrico a la Interface</i> ..	65
Figura 33. <i>Adaptador para el Sensor con Piezoeléctrico con la Interface</i>	65

Figura 34. <i>Diseño CAD 3D del Sensor de Tiempo de Vuelo para el Experimento de Caída Libre</i>	66
Figura 35. <i>Sensor de Tiempo de Vuelo, Experimento de Caída Libre</i>	66
Figura 36. <i>Ciclos de Trabajo Generados por el PWM</i>	68
Figura 37. <i>Diseño CAD de la Carcasa del Carrito para el Experimento de MRU</i>	68
Figura 38. <i>Carrito ensamblado para el experimento de MRU</i>	69
Figura 39. <i>Diseño del PCB con Arduino Nano para el Control del Carrito</i>	70
Figura 40. <i>Diseño del Circuito Impreso del Teclado y Pantalla</i>	70
Figura 41. <i>Configuración de los Puertos de la Placa Arduino</i>	71
Figura 42. <i>Diseño Esquemático de la Interface para Experimento de Mecánica</i>	72
Figura 43. <i>Diseño PCB de la Interface para Experimento de Mecánica</i>	73
Figura 44. <i>Panel Frontal del Gabinete de la Interface</i>	73
Figura 45. <i>Vista Posterior de la Interface</i>	74
Figura 46. <i>Configuración de los Puertos de la Placa Arduino UNO</i>	74
Figura 47. <i>Diseño Esquemático de la Interface para Experimentos de Electricidad</i>	75
Figura 48. <i>Diseño PCB de la Interface para Experimentos de Electricidad</i>	76
Figura 49. <i>Panel Frontal de la Interface para Experimentos de Electricidad</i>	76
Figura 50. <i>Parte Superior de la Interface para Conectar Resistores o Capacitores</i>	77
Figura 51. <i>SopORTE Universal</i>	78
Figura 52. <i>Varilla de Acople, Diseño CAD y varilla Construida</i>	79
Figura 53. <i>Abrazadera de Ángulo Recto</i>	79
Figura 54. <i>IDE Arduino con un Fragmento del Código de la Aplicación</i>	80
Figura 55. <i>Diseño de la Pantalla de Bienvenida en App Inventor</i>	86
Figura 56. <i>Diseño de la Pantalla para la Aplicación Principal</i>	89
Figura 57. <i>Vista de la Pantalla Principal desde un Celular con Android</i>	92
Figura 58. <i>Vista de la Pantalla de Resultados de la Aplicación</i>	93
Figura 59. <i>Vista de la Pantalla de Datos</i>	98
Figura 60. <i>Librería chart.min.js en Carpeta del Proyecto</i>	100
Figura 61. <i>Grafica con Datos de Prueba</i>	102
Figura 62. <i>Generación del Archivo de Instalación App_PhysicsLab_F3.apk</i>	103
Figura 63. <i>Proceso de Instalación de la Aplicación en un Celular Inteligente</i>	104
Figura 64. <i>Pantalla de Bienvenida</i>	105
Figura 65. <i>Pantalla Principal de la Aplicación</i>	106
Figura 66. <i>Proceso para Activar el Bluetooth del Dispositivo Android</i>	107

Figura 67. <i>Proceso de Conexión del Dispositivo Android con la Interface</i>	108
Figura 68. <i>Diseño CAD para Caída libre</i>	109
Figura 69. <i>Montaje Mecánico para el Experimento de Caída Libre</i>	109
Figura 70. <i>Pantalla para el Ingreso de Datos</i>	110
Figura 71. <i>Validación de Datos</i>	111
Figura 72. <i>Pantalla de Resultados, Gráfico y Datos Guardados del Experimento de Caída Libre</i>	113
Figura 73. <i>Diseño CAD para Péndulo Simple</i>	114
Figura 74. <i>Montaje Mecánico para el Experimento de Péndulo Simple</i>	114
Figura 75. <i>Pantalla para el Ingreso de Datos para el Experimento del Péndulo Simple</i>	115
Figura 76. <i>Validación de Datos Ingresados para el Experimento del Péndulo Simple</i>	116
Figura 77. <i>Pantalla de Resultados, Gráfico y Datos Guardados del Experimento de péndulo simple</i>	117
Figura 78. <i>Montaje Mecánico para el Experimento de Movimiento Rectilíneo Uniforme</i>	118
Figura 79. <i>Pantalla para el Ingreso de datos para el Experimento de M.R.U.</i>	119
Figura 80. <i>Pantalla de Resultados, Gráfico y Datos para el Experimento sobre el M.R.U.</i>	120
Figura 81. <i>Interface de Electricidad con la resistencia para el experimento de la ley de Ohm</i>	121
Figura 82. <i>Pantalla para el Inicio del Experimento de ley de OHM</i>	122
Figura 83. <i>Pantalla de Resultados y Gráfico para el Experimento sobre la Ley de Ohm</i>	123
Figura 84. <i>Interface de Electricidad para el Experimento de Carga de un Capacitor</i>	124
Figura 85. <i>Pantalla para el Inicio del Experimento Carga de un Capacitor</i>	125
Figura 86. <i>Pantalla de Resultados y Gráfica para el Experimento de Carga de un Capacitor</i>	126
Figura 87. <i>Interface de Electricidad para el Experimento de Descarga del Capacitor</i>	127
Figura 88. <i>Pantalla para el Inicio del Experimento de Descarga de un Capacitor</i>	128
Figura 89. <i>Pantalla de Resultados y Gráfico para el Experimento de Descarga de un Capacitor</i>	129
Figura 90. <i>Gráfico altura en función del tiempo</i>	134

Figura 91. <i>Gráfico altura en función del tiempo al cuadrado</i>	134
Figura 92. <i>Grafica del Periodo en Función de la Longitud del Péndulo</i>	136
Figura 93. <i>Grafica del Periodo en Función de la Raíz Cuadrada de longitud del Péndulo</i>	136
Figura 94. <i>Gráfico de la Posición en Función del Tiempo</i>	138
Figura 95. <i>Grafica de la Tensión en Función de la Corriente</i>	140
Figura 96. <i>Grafica de la Tensión en Función del Tiempo para la Carga de un Capacitor</i>	141
Figura 97. <i>Grafica Linealizada de la Función Obtenida para la Carga de un Capacitor</i>	142
Figura 98. <i>Grafica de la Tensión en Función del Tiempo para la Descarga de un Capacitor</i>	144
Figura 99. <i>Grafica Linealizada de la Función Obtenida para la Descarga de un Capacitor</i>	144
Figura 100. <i>Hardware del Módulo de Mecánica, Incluye un Actuador, Interface, Sensore</i>	145
Figura 101. <i>Hardware del módulo de electricidad, incluye la Interface, resistencia y capacitores</i>	146
Figura 102. <i>Elementos mecánicos para montaje: varilla, soporte universal y abrazadera</i>	146
Figura 103. <i>Software desarrollado para un dispositivo Android</i>	147
Figura 104. <i>Resultados para el Experimento de Caída Libre</i>	148
Figura 105. <i>Resultados para el Experimento de Péndulo simple</i>	149
Figura 106. <i>Resultados para el Experimento de Movimiento Rectilíneo Uniforme</i>	150
Figura 107. <i>Resultados para el Experimento de la Ley de Ohm</i>	151
Figura 108. <i>Resultados para el Experimento de Carga de un capacitor</i>	152
Figura 109. <i>Resultados para el Experimento de Descarga de un capacitor</i>	153

ÍNDICE DE TABLAS

Tabla 1. <i>Resumen de las Características Generales del Sistema Operativo Android</i>	28
Tabla 2. <i>Tipos de Placas Arduino</i>	33
Tabla 3. <i>Datos del Experimento de Caída libre</i>	133
Tabla 4. <i>Datos del Experimento de Péndulo Simple.</i>	135
Tabla 5. <i>Datos del Experimento de MRU</i>	137
Tabla 6. <i>Datos del Experimento de la Ley de Ohm</i>	139
Tabla 7. <i>Datos del Experimento de Carga de un Capacitor</i>	141
Tabla 8. <i>Datos del Experimento de Descarga de un Capacitor</i>	143

RESUMEN

La presente tesis aborda el diseño y construcción de módulos automatizados para realizar experimentos de Física en las áreas de Mecánica y Electricidad, empleando la plataforma Arduino. El objetivo principal es la precisión y eficiencia en la ejecución experimental y en la recopilación de datos.

El proyecto se desarrolló en tres etapas: diseño conceptual, construcción del prototipo y validación. En la fase de diseño se definieron los requisitos esenciales, como el control de variables experimentales y la automatización de mediciones. Mediante técnicas de modelado y simulación, se desarrolló la arquitectura del sistema, que integra sensores, actuadores y un controlador central.

Durante la construcción, se ensamblaron placas Arduino, módulos Bluetooth, sensores, actuadores y circuitos electrónicos. Además, se diseñó una aplicación en App Inventor para controlar remotamente los módulos y registrar datos desde teléfonos inteligentes. La programación del sistema se realizó en el entorno IDE de Arduino.

Los prototipos fueron evaluados en seis experimentos de Física General: Caída Libre, Péndulo Simple, Movimiento Rectilíneo Uniforme, Ley de Ohm y Carga y Descarga de un Capacitor. Los resultados evidenciaron que los módulos automatizados superan significativamente los métodos manuales, proporcionando mayor precisión, consistencia en los datos y reduciendo los tiempos de ejecución.

En conclusión, los módulos automatizados optimizan los procedimientos experimentales y demuestran un alto potencial para su aplicación en otros contextos de investigación científica.

Palabras clave: módulo automatizado, automatización de experimentos, precisión experimental, Arduino.

ABSTRACT

This thesis explores the design and development of automated modules for conducting physics experiments in Mechanics and Electricity using the Arduino platform. The primary aim is to enhance the precision and efficiency of experimental execution and data collection.

The project was carried out in three main stages: conceptual design, prototype construction, and validation. In the design phase, critical requirements such as controlling experimental variables and automating measurements were identified. Modeling and simulation techniques were employed to define the system architecture, incorporating sensors, actuators, and a central control system.

During the construction phase, Arduino boards, Bluetooth modules, sensors, actuators, and electronic circuits were assembled. Additionally, an application was developed using App Inventor to enable remote control of the modules and data recording through smartphones. The system was programmed in the Arduino IDE.

The prototypes were tested in six General Physics experiments: Free Fall, Simple Pendulum, Uniform Rectilinear Motion, Ohm's Law, and the Charging and Discharging of a Capacitor. Results revealed that automated modules significantly outperform traditional manual methods, delivering higher precision, greater data consistency, and shorter execution times.

In summary, the automated modules enhance the efficiency of experimental procedures and exhibit significant potential for broader applications in scientific research.

Keywords: automated module, experiment automation, experimental accuracy, Arduino.

INTRODUCCIÓN

En la actualidad, el avance de la tecnología y la automatización ha transformado significativamente diversas áreas del conocimiento, incluyendo la investigación científica. En el campo de la Física experimental, la precisión en la recolección de datos y la eficiencia en la realización de experimentos son aspectos fundamentales para obtener resultados confiables. Sin embargo, muchos experimentos tradicionales aún se llevan a cabo de manera manual, lo que puede generar inconsistencias y aumentar el tiempo necesario para su ejecución.

En este contexto, la incorporación de herramientas tecnológicas que permitan automatizar estos procesos resulta crucial para mejorar la calidad de los resultados y optimizar los recursos disponibles. El uso de plataformas electrónicas accesibles, como Arduino, ofrece una alternativa eficaz para el desarrollo de dispositivos automatizados que faciliten la recolección de datos y el control de variables en experimentos físicos.

El presente trabajo de tesis se centra en el diseño y construcción de módulos automatizados para la realización de experimentos en Física, específicamente en las áreas de Mecánica y Electricidad. El objetivo principal es desarrollar un sistema que, mediante la automatización, aumente la precisión de las mediciones y reduzca el tiempo de ejecución de los experimentos.

Este documento se estructura en varias secciones. En primer lugar, se describe el proceso de diseño y construcción del prototipo, incluyendo la integración de sensores, actuadores y sistemas de control. Finalmente, se detallan las pruebas realizadas para evaluar la efectividad del módulo desarrollado, así como las conclusiones obtenidas a partir de los resultados.

Con esta investigación, se espera contribuir al campo de la enseñanza y la investigación experimental en Física, proporcionando una herramienta accesible y eficiente que pueda ser utilizada en diversos entornos académicos y de investigación.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. SITUACIÓN PROBLEMÁTICA

La educación basada en la enseñanza de las ciencias forma parte fundamental del avance de la sociedad, generando conocimientos científicos para el desarrollo de la tecnología y notables innovaciones sumamente beneficiosas para la humanidad (UNESCO, 1999).

Un laboratorio es una excelente herramienta pedagógica y, en muchos aspectos, un espacio esencial para la enseñanza de la ciencia a nivel introductorio. El laboratorio ofrece a los estudiantes la oportunidad de aprender a partir de sus propias experiencias. Además, puede y debe ser utilizado para estimular la curiosidad y el placer por la investigación y el descubrimiento. Proporciona a los alumnos la posibilidad de explorar, manipular, formular hipótesis, cometer errores, reconocerlos y, en consecuencia, aprender de ellos. Asimismo, se busca fomentar la elaboración de conjeturas razonables para explicar las observaciones realizadas, es decir, la construcción de modelos que puedan dar cuenta de dichas observaciones (Gil, 1997).

Tradicionalmente, los laboratorios de Física se diseñan según los requisitos de los planos de estudio, con el objetivo de proporcionar experiencias de aprendizaje. Es decir, permiten que los estudiantes descubran y midan los parámetros asociados a los fenómenos físicos, los cuales, de otro modo, quedarían restringidos al ámbito teórico-conceptual.

Un problema común en los laboratorios se relaciona con los instrumentos, que a menudo son limitados en número y no pueden ser utilizados al mismo tiempo por cada estudiante.

En muchas ocasiones, los profesores utilizan el equipo disponible para demostrar algunos efectos a los estudiantes que observan como si estuvieran en un museo, sin participar activamente. Ésto sucede debido a que para realizar experimentos de física se necesita un laboratorio equipado con muchos instrumentos costosos y complejos, los cuales no siempre están dentro de las posibilidades económicas de las instituciones educativas.

Las configuraciones experimentales pueden abarcar desde métodos muy simples, como el uso de un cronómetro para medir el tiempo de caída de un objeto, hasta configuraciones más complejas en laboratorios de investigación. Sin embargo, un elemento crucial para un aprendizaje efectivo es la participación activa de los estudiantes.

En los laboratorios de Física, es fundamental realizar mediciones tanto directas como indirectas de diversas magnitudes físicas. Estas mediciones son un requisito previo para analizar los fenómenos y aplicar el método científico, que constituye el objetivo principal del laboratorio.

Este proceso de recolección de datos, su análisis y la comprobación de las correspondientes hipótesis se ve limitado por el uso de instrumentos que, tras haber cumplido su tiempo de vida útil, ya no permiten tomar datos con la precisión y exactitud que poseían inicialmente. Además, debido al uso masivo de estos instrumentos, derivado de la gran cantidad de estudiantes que hacen uso de los laboratorios de Física General de la universidad, los dispositivos entran en un proceso acelerado de deterioro, sin que sea posible su restitución en un corto plazo.

En la actualidad, los laboratorios de Física General de la UNSAAC, Universidad Continental y de instituciones educativas de nivel secundario carecen de instrumentos que permitan recopilar y registrar datos de forma automática. Esta labor es realizada, por

ejemplo, por los equipos GLX de Pasco Scientific, los cuales son costosos, al igual que sus sensores y accesorios. Además, su reparación es complicada en caso de deterioro debido a su alto nivel de integración y a la falta de manuales técnicos y de diseño.

El presente trabajo podría ser una alternativa viable para mejorar el desarrollo de las prácticas de laboratorio, diseñando y construyendo módulos automatizados que funcionen como equipos complementarios a los ya existentes en los laboratorios.

La constante evolución de la tecnología electrónica, ha permitido desarrollar y simplificar el uso de los microcontroladores y más aún con el desarrollo de los IDE¹s, como el ya famoso Arduino. Esta tarjeta electrónica de código abierto es ampliamente utilizada como una tarjeta de adquisición de datos, de bajo costo. Debido a estas ventajas, es cada vez más común que los laboratorios de Física sean rediseñados para incorporar esta tecnología.

El empleo de la plataforma electrónica Arduino posibilita la creación de sistemas económicos, como aquellos que simplifican el registro y la transferencia de datos con otros dispositivos como computadoras o teléfonos inteligentes.

1.2. FORMULACIÓN DEL PROBLEMA OBJETO DE LA INVESTIGACIÓN

1.2.1. Problema General

¿Será posible diseñar y construir módulos automatizados basados en Arduino para el desarrollo de experimentos de Física en Mecánica y Electricidad?

¹ IDE siglas en inglés de **I**ntegrated Development Environment

1.2.2. Problemas Específicos

1. ¿Será posible diseñar la arquitectura del hardware del sistema y construirlo, asegurando la integración con los periféricos necesarios para implementar experimentos de Mecánica y Electricidad?
2. ¿Se podrá diseñar e implementar el software necesario para el control y operación de los módulos?
3. ¿Se podrá Probar el sistema hardware-software y verificar su funcionalidad con los experimentos de Física propuestos en este trabajo?

1.3. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

El desarrollo tecnológico y científico de una nación depende de una educación de calidad en sus distintos niveles. Nuestro país está rezagado y estancado en comparación con otras naciones de la región y del mundo (UNESCO, 2023). Una de las muchas causas de este retraso es la falta de material educativo tecnológico necesario en las áreas de ciencias básicas. Aunque existen numerosas empresas extranjeras dedicadas a la fabricación, desarrollo e implementación de laboratorios de ciencias básicas, que ofrecen una amplia variedad de dispositivos y sistemas tecnológicos, sus costos son muy elevados. Esto hace que, en muchos casos, sea imposible para las instituciones educativas adquirir estos equipos o reemplazarlos en caso de deterioro, además de requerir una gran cantidad de accesorios para realizar los distintos experimentos. No obstante, muchos de estos dispositivos pueden desarrollarse y fabricarse a nivel local, lo que permitiría, en algunos casos, subsanar la falta de ciertos instrumentos y, en otros, facilitar su reposición.

El avance de la ciencia y tecnología ha permitido la miniaturización de los sistemas digitales haciendo que cada vez las computadoras tengan dimensiones más

reducidas y su consumo sea masivo, esto hace que su costo sea accesible a todos aquellos usuarios interesados en realizar procesos de automatización y control.

Arduino, forma parte del concepto de hardware y software libre y está abierto para uso y contribución de toda la sociedad. Arduino es una plataforma de prototipos electrónicos y consiste básicamente en una placa electrónica que incorpora un microcontrolador, con un lenguaje de programación en un entorno de desarrollo que soporta entradas y salidas de datos y señales. Esta plataforma permite conectar el mundo físico analógico con el digital.

Dada la necesidad de modernizar y automatizar ciertos procesos experimentales en los laboratorios de Física, con el fin de obtener resultados más precisos y eficientes, este proyecto tiene como objetivo diseñar y construir módulos automatizados. Estos módulos estarán diseñados para ser comparables con los equipos comerciales existentes y se centrarán en la realización de experimentos en las áreas de Mecánica y Electricidad.

La implementación de estos módulos no solo mejorará la calidad de los experimentos en las asignaturas de Física I y Física III, sino que también contribuirá a mitigar la falta de instrumentos especializados en los laboratorios. De este modo, el proyecto elevará el nivel educativo y experimental, proporcionando a los estudiantes herramientas efectivas para su formación en ciencias físicas.

Estos módulos también podrían ser utilizados por universidades y colegios de nivel secundario tanto de la región como a nivel nacional y con la posibilidad de ampliar su uso y aplicación a otros laboratorios no considerados en el presente trabajo. Dentro de las principales características de los equipos que se han desarrollado en este trabajo están:

- **Bajo costo.** Los elementos necesarios para ensamblar los módulos están disponibles en la región y, en ocasiones, pueden ser importados con facilidad. Se empleará software de código abierto para la aplicación, lo que permitirá contar con equipos de laboratorio económicos en comparación con los desarrollados por empresas como PASCO, VERNIER y otras.

Como ejemplo de esta característica, podemos citar el sistema de caída libre de la empresa PASCO Scientific, de código ME-9889, cuyo costo es de S/. 4218.00 (ver cotización en el Anexo 2). Este costo supera la inversión realizada para la construcción de los módulos propuestos en este trabajo.

- **Arquitectura Abierta.** Los módulos se crearon con la capacidad de modificarlos tanto al nivel hardware como al nivel software, para ajustar y agregar mayor funcionalidad, con el objetivo de incrementar el número de experimentos.
- **Mantenimiento y reparación.** Se facilitarán gracias al diseño y construcción de las placas electrónicas, los componentes mecánicos de los sensores y el software de control, lo que permitirá reemplazar cualquier parte según sea necesario.
- **Uso de tecnología de conectividad inalámbrica,** como bluetooth y Wi-Fi, para la comunicación. Estas tecnologías están integradas en la mayoría de los dispositivos de uso común, como smartphones, Smart TVs, smartwatches, tablets y computadoras personales, lo que los hace portátiles y compatibles con las interfaces disponibles en cualquier institución educativa.
- **Uso del internet y la WEB.** Para la transferencia de datos recopilados y su posterior análisis con programas de computadora tales como Excel, Matlab, etc.

- **Viabilidad.** La viabilidad de este proyecto radica en que el diseño y construcción de un módulo didáctico para el laboratorio de Física General que se propone no requiere de un gran presupuesto económico. Además, las pruebas y calibración se llevaron a cabo en los laboratorios del Departamento Académico de Física de la UNSAAC.

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. Objetivo general

Diseñar y construir módulos automatizados, basados en Arduino, para el desarrollo de experimentos de Física en Mecánica y Electricidad.

1.4.2. Objetivos específicos

1. Diseñar la arquitectura del hardware del sistema y construirlo, asegurando la integración con los periféricos necesarios para implementar experimentos de Mecánica y Electricidad.
2. Diseñar e implementar el software necesario para el control y operación de los módulos.
3. Probar el sistema hardware-software y verificar su funcionalidad con los experimentos de Física en Mecánica y Electricidad propuestos en este trabajo.

CAPÍTULO II: MARCO TEÓRICO CONCEPTUAL

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Arduino ha revolucionado el campo de la electrónica y la programación, haciéndolos accesibles para estudiantes, educadores, investigadores y aficionados. En el ámbito de la educación en Física, Arduino ha permitido el desarrollo de experimentos interactivos y económicos que facilitan la comprensión de conceptos teóricos mediante la práctica. A continuación, se detallan algunos antecedentes y el desarrollo del uso de Arduino en experimentos de Física.

2.1.1. Antecedentes Internacionales

1.- Higuera, Guzmán y Rojas (2019) Implementando las metodologías STEAM² y ABP³ en la enseñanza de la Física mediante Arduino.

El objetivo principal fue implementar las metodologías STEAM y ABP a través del desarrollo de proyectos en clase, apoyados por la tecnología de software y hardware libre de Arduino, y con el respaldo de la documentación disponible en la web.

Mediante el uso de la tecnología Arduino y la programación de sensores específicos para cada proyecto, junto con la aplicación del aprendizaje basado en proyectos e integrado con la metodología STEAM, se ha observado que los estudiantes del curso de Física Mecánica del semestre A de 2019, junto con los integrantes del grupo de investigación GRUCEDI, lograron desarrollar hábitos de estudio, aumentar su disciplina y participar activamente en proyectos de investigación.

² STEAM siglas en inglés de **Science, Technology, Engineering, Arts & Mathematics**.

³ ABP siglas de **Aprendizaje Basado en Proyectos**

Se concluye que la implementación de estas metodologías en instituciones educativas puede ofrecer grandes ventajas tanto en el proceso de aprendizaje como en la optimización de recursos para los estudiantes. Así, es posible crear instrumentos de medición de calidad comparable a los más avanzados del mercado, pero con un costo considerablemente menor.

2.- Wurm, Ríos, Marinelli y Bernhardt (2018) Desarrollo de Instrumental de Laboratorio Controlado por Arduino.

El objetivo de este proyecto es desarrollar equipos de laboratorio de bajo costo que integren tecnología Arduino, facilitando la realización de actividades experimentales o demostrativas en la enseñanza de la Física. Desde la cátedra de Electrónica del Profesorado en Física, se plantea la necesidad de fomentar en los estudiantes, próximos a egresar, el uso de estas tecnologías emergentes. Con el fin de integrar conocimientos, se propone como proyecto final la elaboración de un dispositivo que combine programación, electrónica y física.

Se concluye que el desarrollo de este tipo de instrumentos permite a los estudiantes aplicar e interrelacionar conceptos de electrónica, programación y física, contando con el acompañamiento de los docentes. Esta experiencia en el aula promueve el aprendizaje basado en proyectos, enriqueciendo el proceso formativo.

3.- Silva (2024) Utilizando la plataforma Arduino en el proceso de enseñanza/aprendizaje de fundamentos de la electricidad en física.

La enseñanza de la Física ha enfrentado varios obstáculos, algunos de ellos relacionados con cuestiones estructurales y otros con factores culturales. Los problemas estructurales derivan tanto del diseño de la enseñanza de esta disciplina como de la falta de una infraestructura adecuada. En cuanto a las dificultades

culturales, estas provienen de un enfoque rígido en el cual el profesor es percibido como el único depositario del conocimiento, mientras que los estudiantes se limitan a memorizar y repetir información de manera mecánica. En este contexto, la metodología activa surge como una alternativa para transformar este enfoque, permitiendo que el estudiante asuma un rol central y participe de manera más activa, lo que genera múltiples beneficios.

Por tanto, el objetivo de este trabajo es implementar clases basadas en una metodología activa, utilizando la placa Arduino para la enseñanza de Física a estudiantes de secundaria, y realizar posteriormente una evaluación a través de métodos cualitativos y cuantitativos. Los resultados han sido satisfactorios, destacando que el 99% de los estudiantes recomendaría este tipo de clases para la enseñanza de Física, y el 100% calificó la metodología activa como una opción con gran potencial en comparación con la enseñanza tradicional.

2.1.2. Antecedentes Nacionales

1.- Centurión (2020) Capacidades del empleo de Arduino sobre la operatividad del laboratorio del curso de Física General.

El objetivo de este proyecto fue desarrollar un circuito utilizando una tarjeta Arduino con conectividad Bluetooth y una aplicación Android, con el fin de automatizar las mediciones de un experimento del Curso de Física General impartido en la Universidad Privada de la Selva Peruana. La metodología empleada fue de tipo descriptivo, con un diseño no experimental. Se utilizó una tarjeta Arduino equipada con sensores infrarrojos para capturar y procesar los datos, que luego fueron transmitidos por Bluetooth a una aplicación Android para su representación gráfica.

Se diseñó una práctica de laboratorio para el Curso de Física General, basada en un experimento que determinaba la aceleración de la gravedad (g) de manera indirecta, mediante la medición del periodo de un péndulo simple. Además, se aplicó una encuesta a siete docentes que imparten el curso de Física General o asignaturas similares a nivel universitario. Los resultados mostraron que los docentes consideraron que el dispositivo de medición es adecuado para la enseñanza sobre el péndulo simple, aunque señalaron que aún tiene margen para mejorar.

2.- Chang (2023) Arduino: medición y registro de temperaturas en cursos formativos para la carrera de meteorología en la UNALM 2022-I.

El objetivo del trabajo fue desarrollar las destrezas para que un estudiante de la carrera de meteorología/IMGRC pueda medir y registrar variables meteorológicas usando dispositivos electrónicos de bajo costo. Dos grupos de estudiantes voluntarios llevaron a cabo dos experimentos: en el primero, utilizaron termómetros de alcohol o mercurio para calcular la conductividad de la espuma de poliestireno y verificar la ley de enfriamiento; en el segundo, emplearon la plataforma Arduino con sensores digitales DS18B20 para medir la temperatura. Los estudiantes manifestaron satisfacción por el avance en sus conocimientos de electrónica, aprendieron las ventajas y limitaciones del uso de equipos electrónicos de bajo costo, y reconocieron el potencial de esta tecnología en sus futuras carreras.

3.- Sánchez y Cruz (2018) Diseño e implementación de un módulo didáctico de control de temperatura para la escuela profesional de ingeniería electrónica en la UNPR.

El propósito de este proyecto es diseñar e implementar un módulo didáctico de control de temperatura para el laboratorio de la escuela profesional de Ingeniería Electrónica de la UNPRG. Una vez completado, este módulo facilitará el desarrollo de conocimientos teóricos y prácticos en el área de instrumentación industrial, así como en el manejo de los equipos asociados. Para seleccionar los componentes del módulo, se priorizó tanto su funcionalidad como su costo de adquisición.

Una vez implementado, se obtuvo el modelo matemático de la planta y se desarrolló un algoritmo de control proporcional-integral (PI), ampliamente utilizado en la industria. Este algoritmo se aplicó mediante conceptos de control por computadora, que permitió que el sistema gestionara eficientemente los lazos de control dentro de un rango de temperaturas de 25.8 °C a 73.2 °C. Además, se diseñó una interfaz de usuario que facilita la visualización gráfica de las variables del sistema.

2.1.2. Antecedentes Locales

No se han encontrado trabajos relacionados con el presente proyecto en el repositorio de la Universidad Nacional de San Antonio Abad del Cusco, ni en los repositorios de otras universidades locales.

2.2. BASES TEÓRICAS

2.2.1. Mecánica Clásica.

La Mecánica Clásica es una rama de la Física que se encarga de analizar las leyes que rigen el movimiento de los cuerpos físicos a escala macroscópica, en contraste con la Mecánica Cuántica. Esta disciplina se aplica a velocidades bajas en comparación con la

velocidad de la luz en el vacío. En ella, se destacan tres principios clave: el tiempo es absoluto, los sistemas evolucionan temporalmente según el Principio de Mínima Acción, y las leyes que los gobiernan son completamente deterministas (Goldstein, Poole y Safko, 2002).

El primer enfoque de la Mecánica Clásica, conocido como mecánica Newtoniana, se basa en los fundamentos teóricos de Isaac Newton y los métodos matemáticos desarrollados por él, junto con figuras como Gottfried Wilhelm Leibniz, Joseph-Louis Lagrange y Leonhard Euler, en el siglo XVII (Serway & Jewett, 2018). Estos métodos describen el movimiento de los cuerpos físicos bajo la acción de fuerzas. Más adelante, se desarrollaron enfoques más abstractos, dando lugar a las reformulaciones de la Mecánica Clásica conocidas como Mecánica Lagrangiana y Mecánica Hamiltoniana, avances que se lograron en los siglos XVIII y XIX. Estas reformulaciones van más allá del enfoque de Newton, utilizando métodos analíticos más complejos, y, con adaptaciones, siguen siendo relevantes en diversas áreas de la Física Moderna (Marion & Thornton, 2004).

La Mecánica Clásica ofrece predicciones muy precisas cuando se aplica a objetos grandes, que no sean extremadamente masivos, y a velocidades considerablemente menores que la velocidad de la luz en el vacío (Goldstein, Poole y Safko, 2002).

2.2.1.1. Movimiento Rectilíneo Uniforme

En el ámbito de la mecánica, se define un movimiento como rectilíneo uniforme cuando un objeto, el móvil, se desplaza en una línea recta a una velocidad constante, sin experimentar aceleración. A menudo, se abrevia este tipo de movimiento como MRU, aunque en ciertos lugares se le conoce como MRC, que significa movimiento rectilíneo constante (Serway & Jewett, 2018). El MRU se caracteriza por las siguientes propiedades:

- Es un movimiento a lo largo de una línea recta.
- Tanto la magnitud como la dirección y el sentido de la velocidad se mantienen invariables a lo largo del tiempo. Esto implica que el objeto se desplaza a una velocidad constante sin experimentar cambios en su rapidez ni en la dirección en la que se mueve.
- La magnitud de la velocidad se conoce como celeridad o rapidez, y en este caso, no hay aceleración presente (Tipler & Mosca, 2020).

El movimiento puede ocurrir en dos sentidos; una velocidad negativa indica un movimiento en sentido contrario a la convención adoptada como positiva.

Según la primera ley de Newton, una partícula puntual permanece en reposo o en MRU cuando no hay una fuerza externa neta actuando sobre ella. Esto se debe a que, en estas condiciones, las fuerzas actuantes están en equilibrio, lo que resulta en un estado de reposo o MRU. Sin embargo, esta es una situación ideal, ya que, en la práctica, siempre hay fuerzas presentes que tienden a alterar el movimiento de las partículas, lo que hace difícil encontrar un MRU sin la influencia de fuerzas adicionales que conduzcan a una fuerza neta nula (Serway & Jewett, 2018).

De acuerdo con las restricciones establecidas en las características del MRU, un objeto se desplaza en una trayectoria recta, en una única dirección, cubriendo distancias uniformes en intervalos de tiempo iguales. Durante todo su recorrido, mantiene una velocidad constante y no experimenta aceleración. Es importante destacar que la velocidad, siendo un vector, permanece invariable en cuanto a magnitud y dirección al ser constante (Halliday, Resnick, & Walker, 2017).

Para que un objeto experimente un movimiento rectilíneo uniforme (MRU), es esencial que se satisfaga la siguiente condición (Tipler & Mosca, 2020).

$$\left(v = \frac{x-x_0}{t-t_0} \right) = \textit{Constante} \quad (1)$$

Donde:

x es la posición en el espacio

t es el tiempo

De esta condición, llegamos a la ecuación del M.R.U.:

$$x = x_0 + v(t - t_0) \quad (2)$$

Donde:

x₀ es la posición en el instante t₀

2.2.1.2. Caída Libre

El estudio del movimiento de caída de cuerpos sobre la Tierra se remonta a las aportaciones del filósofo griego Aristóteles (384 a.C. – 322 a.C.), quien, basado principalmente en la intuición y observación del movimiento de los cuerpos en la naturaleza, concibió que los cuerpos más pesados caen más rápido que los menos pesados en una proporción igual a su peso. Esta idea de caída libre permaneció alrededor de dos mil años, hasta que el físico y matemático italiano Galileo Galilei (1564-1642 d.C.) demostró lo contrario, a partir de diversos experimentos que realizó al soltar varios cuerpos al mismo tiempo y desde alturas iguales. Como resultado de la experimentación, formuló que los cuerpos con diferentes pesos caen al mismo tiempo y que, por tanto, los tiempos de caída no dependen del peso de los cuerpos, como lo afirmaba Aristóteles, sino de la resistencia del aire que se opone a la caída de los objetos (Drake, 1989).

Esta concepción de la caída libre de cuerpos fue perfeccionada con el trabajo científico del físico y matemático inglés Isaac Newton (1642-1727 d.C.), en particular con la Ley de la Gravitación Universal. Experimentos como el mostrado en la Figura 1

revelan que se pueden omitir los efectos del aire si, además, la distancia de caída es pequeña en comparación con el radio terrestre, y si ignoramos los pequeños efectos debidos a la rotación de la Tierra. En estas condiciones, la aceleración es constante y corresponde a un movimiento uniformemente acelerado. El modelo idealizado que surge de tales supuestos se denomina caída libre, aunque también incluye el movimiento ascendente (Drake, 1989).

Figura 1.

Caída Libre de una Esfera y una Pluma



Nota. https://www.experimente.physik.unifreiburg.de/Mechanik_der_festen_Koerper/A3_Geradlinige_Bewegung_Kraft_Masse_Beschleunigung/fallversuche/fallimvakuum

La aceleración constante de un cuerpo en caída libre, se llama aceleración debida a la gravedad se la denota con la letra g , su valor se ha medido a nivel del mar y a una latitud de 45° , y es $32,17 \text{ ft/s}^2$ ó $9,806 \text{ m/s}^2$ (Tippens, 2011). El valor exacto varía según el lugar. En la superficie de la luna, la aceleración debida a la gravedad se debe a la fuerza de atracción de la Luna y no de la tierra, siendo su valor $1,624 \text{ m/s}^2$ (Cox, 2000). Cerca de la superficie del Sol, la aceleración es de $273,95 \text{ m/s}^2$ (Cox, 2000).

En general, para un cuerpo que se mueve con aceleración constante, la ecuación del desplazamiento está dada por:

$$x = x_o + v_o t + \frac{1}{2} a t^2 \quad (3)$$

Donde:

x_o es el desplazamiento inicial,

v_o velocidad inicial,

t tiempo y

a es la aceleración.

En el caso de los cuerpos en caída libre, es decir, objetos que caen únicamente bajo la influencia de la gravedad y que parten del reposo, la ecuación se simplifica. Dado que la velocidad inicial es cero y la aceleración es la aceleración de la gravedad g .

$$y = \frac{1}{2} g t^2 \quad (4)$$

Dónde:

y es el desplazamiento vertical y

g la aceleración de la gravedad.

2.2.1.3. Péndulo Simple.

Por definición, un péndulo simple consiste en una partícula de masa m suspendida por una cuerda inestirable, sin masa, de longitud L y ubicada en una región del espacio en la que hay un campo gravitacional uniforme, por ejemplo, cerca de la superficie de la tierra. La partícula suspendida se llama el bob del péndulo. Aquí discutimos el movimiento del bob. Si bien los resultados a revelar aquí son más precisos para el caso de una partícula puntual, son buenos siempre y cuando la longitud del péndulo (desde el extremo superior fijo de la cuerda hasta el centro de masa del bob) sea grande en comparación con una dimensión característica (como el diámetro si el bob es una esfera o la longitud de la arista si es un cubo) del bob. El uso de un bob de péndulo cuyo diámetro es 10% de la longitud del péndulo (a diferencia de una partícula puntual)

introduce un error de 0.05%. Hay que hacer el diámetro del bob 45% de la longitud del péndulo para obtener el error hasta 1%. (Schnick, 2007).

Figura 2.

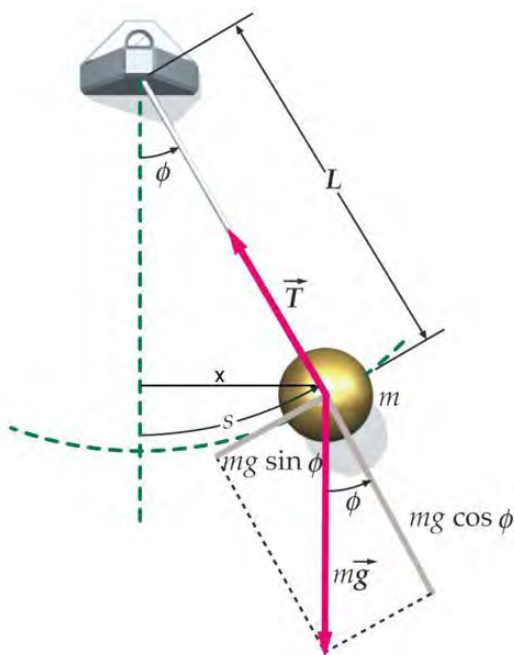
Péndulo Simple



Nota. <http://jesolesito1012.wixsite.com/fisica-mazuera-11/pndulo-simple>

Figura 3.

Péndulo Simple Idealizado



Nota. <https://planetario.buenosaires.gob.ar/sites/default/files/2018-08/galileo>

En la Figura 3 el bob se mueve con movimiento circular por consiguiente podemos aplicar la segunda ley de Newton para las rotaciones, es decir:

$$\mathcal{T} = I\alpha \quad (5)$$

Donde: \mathcal{T} : el momento de fuerzas o torque

I : momento de inercia

α : aceleración angular

El torque \mathcal{T} es negativo por el sentido horario de la rotación

$$\mathcal{T} = -Fb \quad (6)$$

La fuerza efectiva $F = mg\text{sen}\theta$ y el brazo $b = L$, en consecuencia

$$\mathcal{T} = -mgL\text{sen}\theta \quad (7)$$

El momento de inercia I de la masa m respecto al punto O

$$I = mL^2 \quad (8)$$

La aceleración angular de m : $\alpha = \ddot{\theta}$ (9)

Reemplazando las ecuaciones (2), (3) y (4) en la ecuación (1) se tiene:

$$-mgL\text{sen}\theta = (mL^2)\ddot{\theta}$$

Simplificando y trasponiendo términos, se expresa en la forma:

$$\ddot{\theta} + \frac{g}{L}\text{sen}\theta = 0 \quad (10)$$

Para pequeñas oscilaciones $\text{sen}\theta \approx \theta$ (rad), reemplazando esta aproximación en la ecuación (5):

$$\ddot{\theta} + \frac{g}{L}\theta = 0 \quad (11)$$

La ecuación (6) es similar a la ecuación del Movimiento Armónico Simple, cuya ecuación es:

$$\ddot{x} + \omega^2 x = 0 \quad (12)$$

De las ecuaciones (6) y (7) podemos implicar que:

$$\omega^2 = \frac{g}{L} \implies \omega = \sqrt{\frac{g}{L}} \quad (13)$$

Donde: ω Es la frecuencia angular de las pequeñas oscilaciones en el péndulo

$$\text{Finalmente } \omega = \frac{2\pi}{T} \implies T = 2\pi \sqrt{\frac{L}{g}} \quad (14)$$

Donde: T es el periodo del péndulo

2.2.2. Electricidad.

La electricidad es una forma de energía que surge a partir de los fenómenos físicos asociados a las cargas eléctricas, ya sean estacionarias o en movimiento. Más precisamente, se refiere al movimiento de los electrones en la capa exterior de los átomos de un material conductor. Una carga eléctrica, en tanto, es una de las propiedades fundamentales de la materia. Como sabemos, toda la materia está compuesta por átomos. Y dentro de un átomo, hay tres tipos de partículas más pequeñas: protones, neutrones y electrones (Tipler & Mosca, 2020).

Los protones y los electrones son partículas cargadas eléctricamente. Mientras que los protones tienen carga eléctrica positiva (+), los electrones tienen carga eléctrica negativa (-). Los neutrones, en tanto, no tienen carga eléctrica, por lo que son neutros. Los protones y neutrones se encuentran en el núcleo, en el centro de los átomos, pero en

la manifestación de la electricidad como fenómeno físico, la principal partícula involucrada es el electrón, que se encuentra fuera del núcleo (Halliday, Resnick, & Walker, 2017).

En un átomo, la cantidad de protones en el núcleo es igual a la cantidad de electrones alrededor y fuera de este, por lo que la cantidad de cargas positivas y negativas es igual, y el átomo se encuentra eléctricamente equilibrado. Esto hace que un átomo sea neutro (Serway & Jewett, 2018).

Sin embargo, determinados fenómenos logran movilizar a los electrones de las órbitas más externas del átomo, provocando un déficit de cargas negativas y convirtiendo al átomo en un ion positivo. Al producirse este movimiento y al trasladar un electrón de su órbita, se genera un espacio que atrae a un electrón de un átomo contiguo, desencadenando así un flujo de electrones en movimiento desde otros átomos contiguos (Tipler & Mosca, 2020).

La electricidad es la presencia o el flujo de estas partículas con carga, y las diversas manifestaciones de la electricidad son el resultado de la acumulación o el movimiento de los diferentes números de electrones (Halliday, Resnick, & Walker, 2017).

Diferencia de Potencial Eléctrico (V)

La diferencia de potencial eléctrico, o voltaje, es el trabajo externo requerido para trasladar una carga de una posición a otra dentro de un campo eléctrico. Representa el cambio en la energía potencial que experimenta una carga de prueba de $+1C$ (Serway & Jewett, 2018). La diferencia de potencial eléctrico se mide en unidades de julios por culombio (J/C). A esta diferencia de potencial se le llama voltaje, en honor a Alessandro Volta, el inventor de la batería (Tipler & Mosca, 2020).

Intensidad de Corriente Eléctrica (I)

Este término se refiere al flujo de corriente eléctrica, es decir, a la cantidad de carga eléctrica que atraviesa un material conductor en un determinado periodo de tiempo. El flujo de corriente eléctrica puede compararse con el caudal de un río, capaz de transportar cargas y realizar una cantidad de trabajo. Según el Sistema Internacional (SI), esta intensidad se mide en culombios por segundo (C/s), lo que equivale a un amperio (A), una unidad fundamental en el campo de la electricidad nombrada en honor al físico francés André-Marie Ampere. Para medir la intensidad de la corriente eléctrica se utilizan instrumentos como el galvanómetro o el amperímetro (Halliday, Resnick, & Walker, 2017).

Resistencia Eléctrica (R)

La resistencia eléctrica es una propiedad de los materiales que describe su oposición al flujo de corriente eléctrica. Se mide en ohmios (Ω) y es uno de los conceptos más fundamentales en el campo de la electricidad y la electrónica. Los ohmios reciben su nombre en honor a Georg Simon Ohm (Serway & Jewett, 2018).

2.2.2.1.Ley de Ohm

La corriente eléctrica I que fluye a través de un conductor es proporcional a la diferencia de potencial V aplicada. El físico alemán Georg Simon Ohm (1787-1854) fue el pionero en demostrar de forma experimental que la corriente en un alambre metálico es proporcional al voltaje aplicado:

$$I \propto V \quad (15)$$

Esta relación fundamental constituye la base de la ley de Ohm. Se puede interpretar como una relación de causa y efecto, donde la diferencia de potencial es la causa y la corriente

eléctrica es el efecto. Esta ley es empírica, es decir, se deriva de la observación experimental (Tipler & Mosca, 2020).

Ecuación Matemática de la ley de Ohm

La ley de Ohm es fundamental para comprender los principios básicos de los circuitos eléctricos. Estipula que la diferencia de potencial V aplicada a través de un conductor es directamente proporcional a la intensidad de corriente I que circula por dicho conductor. Ohm complementó esta ley al introducir el concepto de resistencia eléctrica R , que es el factor de proporcionalidad en la relación entre V e I (Serway & Jewett, 2018). La ley de Ohm se expresa matemáticamente de la siguiente manera:

$$I = \frac{1}{R} \Delta V \quad \text{ó} \quad \Delta V = RI \quad (16)$$

Donde:

V : es la tensión eléctrica entre los extremos del conductor

I : es la intensidad de corriente que pasa a través del conductor

R : es la resistencia eléctrica del conductor.

La ley de Ohm incorpora tres de las magnitudes fundamentales para trabajar con circuitos.

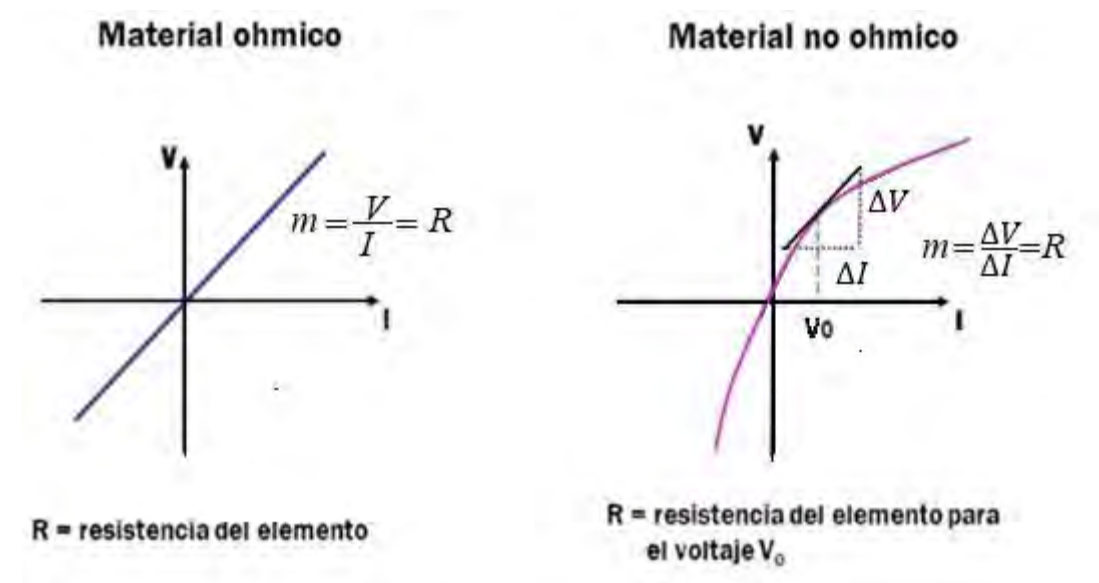
Conductores óhmicos

Los conductores óhmicos son aquellos que cumplen con la ley de Ohm.

El gráfico del voltaje vs. la intensidad de corriente en un conductor óhmico es una línea recta, lo que no ocurre con los conductores no óhmicos. (Giancoli, 2008)

Figura 4.

Grafica de la $V=f(I)$ para un Material Óhmico y no Óhmico



Nota. <https://www.unicoos.com/foro/fisica>

2.2.2.2. Constante de Tiempo de un Circuito RC.

La constante de tiempo en un circuito RC es un concepto fundamental en electrónica que describe el tiempo requerido para que un capacitor se cargue o descargue a través de una resistencia. La constante de tiempo τ se calcula utilizando la resistencia R y la capacitancia C del circuito, y tiene aplicaciones importantes en el diseño de filtros, osciladores y otros circuitos electrónicos (Serway & Jewett, 2018).

En un circuito RC compuesto por un resistor R y un capacitor C , la constante de tiempo es el producto de las magnitudes de estos dos componentes:

Resistencia (R): Es medida en ohmios (Ω) y representa la oposición al flujo de corriente en el circuito.

Capacitancia (C): Es medida en faradios (F) y describe la capacidad del capacitor para almacenar carga eléctrica.

La constante de tiempo se define como:

$$\tau = R \cdot C \quad (17)$$

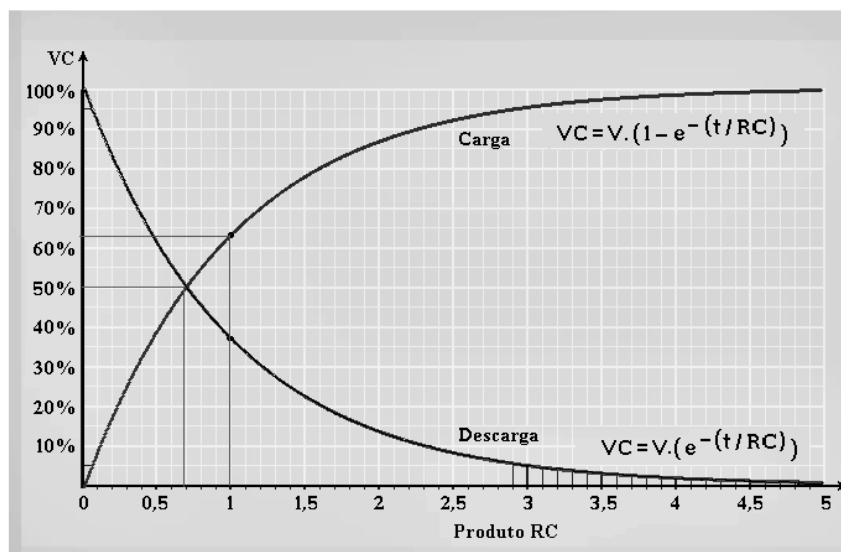
Significado Físico

La constante de tiempo se define como el tiempo necesario para que la tensión en el capacitor alcance aproximadamente el 63.2% de su valor final durante la carga o disminuya al 36.8% de su valor inicial durante la descarga. En términos generales, después de transcurrir 5 constantes de tiempo, el capacitor se encontrará completamente cargado o descargado (Tipler & Mosca, 2020), como se muestra en la Figura 5.

- **Carga:** Durante el proceso de carga, la tensión en el capacitor aumenta hasta V_0 , que es la tensión máxima aplicada al circuito RC. La ecuación que describe la carga es $V(t) = V_0 \cdot (1 - e^{-t/\tau})$, donde τ representa la constante de tiempo de carga del capacitor y t es el tiempo (Halliday, Resnick, & Walker, 2017).
- **Descarga:** En el proceso de descarga, la tensión en el capacitor disminuye exponencialmente, también regida por la constante de tiempo τ . La ecuación correspondiente a la descarga es $V(t) = V_0 \cdot e^{-t/\tau}$ (Halliday, Resnick, & Walker, 2017)

Figura 5.

Curvas de Carga y Descarga de un Capacitor



Nota. https://diegorl.neocities.org/mod2_p2p_obligatoria/condensadores

En consecuencia, la constante de tiempo τ constituye un parámetro fundamental que caracteriza la respuesta de un circuito RC a variaciones en la tensión, y se aplica en diversas áreas de la ingeniería electrónica. (Nilsson y Riedel, 2020)

2.2.3. Plataformas de Desarrollo

La plataforma de desarrollo de software es una herramienta que facilita a los desarrolladores la creación de aplicaciones y programas. Estas plataformas ofrecen un entorno de desarrollo integrado (IDE) que incluye todas las herramientas necesarias para el desarrollo, prueba y despliegue de aplicaciones. Además, proporcionan una variedad de herramientas de programación, bibliotecas y marcos de trabajo que ayudan a los desarrolladores a trabajar de manera más eficiente.

2.2.3.1. Ordenador Personal (PC)

Una computadora personal es un dispositivo electrónico que consta principalmente de un procesador, memoria y dispositivos de entrada/salida (E/S). También conocida como ordenador o computador personal, abreviada como PC (siglas en inglés de personal computer), se trata de una forma de microcomputadora diseñada originalmente para el uso individual, y comúnmente, el término PC se asocia a las computadoras compatibles con IBM PC. Normalmente, se utiliza una computadora personal por un único usuario, aunque existen sistemas operativos que permiten la interacción de varios usuarios al mismo tiempo, lo que se conoce como multiusuario. (Tanenbaum y Bos, 2014)

Una computadora personal típicamente está configurada para realizar las tareas habituales de la informática contemporánea, como la navegación por Internet, el estudio,

la creación de documentos de texto y tareas de oficina o educativas, como la edición de textos y bases de datos. Además, permite actividades de entretenimiento, como la reproducción de música, la visualización de videos y juegos, entre otras. En términos de portabilidad, podemos diferenciar entre computadoras de escritorio y computadoras portátiles. (Morley y Parker,2017).

2.2.3.2. Teléfono Inteligente.

Un teléfono inteligente, conocido como "Smartphone" en inglés, es un dispositivo móvil que posibilita la realización de llamadas telefónicas y añade funcionalidades que anteriormente solo estaban disponibles en asistentes personales digitales (PDA) o computadoras, como la edición de documentos, la navegación por Internet, la instalación de aplicaciones adicionales, entre otras. El término "inteligente" se utiliza mayormente con fines comerciales y generalmente se refiere a su capacidad para funcionar como una computadora de mano, llegando incluso a sustituir a una computadora.

- **Sistema Operativo.** Un smartphone típicamente utiliza un sistema operativo que posibilita la ejecución de aplicaciones, como iOS de Apple, Android de Google, entre otros.
- **Aplicaciones.**
A pesar de que todos los dispositivos móviles en la actualidad cuentan con algún tipo de software, un smartphone sobresale por su capacidad para realizar una amplia gama de funciones adicionales, como la creación, visualización o edición de documentos, la instalación de aplicaciones como asistentes personales, la edición de fotos, la reproducción de música, entre otras.
- **Acceso web.** Los Smartphone pueden acceder a la web a altas velocidades vía Wifi o redes 3G, 4G o 5G. (Gibson,2020).

2.2.3.3. Android

Es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, la compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Actualmente, los dispositivos Android tienen mayores ventas que las ventas combinadas de Windows Phone e iOS. (Baker y St. Laurent, 2011).

Tabla 1.

Resumen de las Características Generales del Sistema Operativo Android

Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX, GPRS, UMTS y HSDPA+.
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto, además del servicio de Firebase Cloud Messaging (FCM) siendo la nueva versión de Google Cloud Messaging (GCM) bajo la marca Firebase con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo.

Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y se ejecuta en la Máquina Virtual Dalvik, Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART). El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
Soporte multimedia	Soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
Soporte para hardware adicional	Soporta cámaras de fotos, vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno oficial es Android Studio, descargable desde la página oficial de desarrolladores de Android.
Google Play	Es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil.

Bluetooth	Soporte para A2DP y AVRCP, el envío de archivos (OPP) y la exploración del directorio telefónico y el marcado por voz junto con el envío de contactos entre teléfonos.
Videollamada	Soporta videollamada a través de Hangouts (ex-Google Talk).
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico.

Nota. Halliday, Resnick, & Walker, 2017

Arquitectura

Los componentes principales del sistema operativo de Android son (Baker y St. Laurent, 2011):

- **Aplicaciones:** Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas

son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

- **Runtime de Android:** Incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato *dex* por la herramienta incluida "*dx*". Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

2.2.4. Hardware

El término "hardware" se refiere a todos los componentes físicos que componen un sistema informático o electrónico. El hardware es fundamental para el funcionamiento de cualquier dispositivo electrónico o computadora, y se contrasta con el software, que comprende los programas y aplicaciones que se ejecutan en el hardware. (Tanenbaum y Bos, 2014)

2.2.4.1. Arduino

La información obtenida es una recopilación de la página web del proyecto y otras similares que hacen referencia al proyecto Arduino.



Arduino es una plataforma electrónica de código abierto que ofrece tanto software como hardware de fácil uso. Proporciona una herramienta de programación de código abierto y accesible para escribir y cargar código en sus placas. Las placas Arduino pueden leer entradas de diversos sensores, mensajes SMS, publicaciones en redes sociales, entre otros, y convertir estas entradas en salidas que activan motores, encienden luces, publican contenido en línea o realizan eventos externos (Arduino, 2023).

Desde su origen, Arduino ha impulsado el desarrollo de una amplia gama de proyectos, desde simples dispositivos cotidianos hasta complejos instrumentos científicos. La clave de su éxito reside en la dinámica comunidad que respalda este proyecto, compuesta por estudiantes, entusiastas, artistas, programadores y profesionales que se han unido en torno a esta plataforma de código abierto. Esta comunidad, que constantemente crea nuevo contenido, comparte conocimientos y resuelve dudas, ha sido esencial. Su aporte ha generado una gran cantidad de información accesible que resulta beneficiosa tanto para principiantes como para expertos.

Tipos de Placas Arduino

En el mercado existen diversos tipos de placas Arduino, cada una con características específicas que se adaptan a distintas aplicaciones. A continuación, en la Tabla 2, se presentan algunos modelos junto con sus especificaciones técnicas.

Tabla 2.*Tipos de Placas Arduino*

MODELO	DESCRIPCIÓN
 <p data-bbox="448 1070 746 1106">Figura 6. <i>Arduino Uno</i></p>	<p data-bbox="849 517 1390 842">Arduino Uno, es el modelo de referencia para la plataforma Arduino y es compatible con la mayoría de los shields existentes, tiene 14 Entradas/Salidas digitales (6 de las cuáles pueden ser usadas como salidas PWM), 6 entradas analógicas, velocidad de reloj de 16MHz, conexión USB, jack de alimentación y un botón de reset.</p> <p data-bbox="849 882 1390 1133">Contiene todos los componentes necesarios para el funcionamiento del microcontrolador. Para comenzar a utilizarlo, simplemente se debe conectar a una computadora mediante un cable USB o alimentarlo con un adaptador AC-DC o una batería (Shertz, 2023).</p>
 <p data-bbox="448 1729 746 1765">Figura 7. <i>Arduino Pro</i></p>	<p data-bbox="849 1202 1390 1787">El Arduino Pro está disponible en variantes de 3.3V/8MHz y 5V/16MHz. Aunque tiene la misma cantidad de pines y periféricos que el Arduino UNO, está basado en el mismo microcontrolador (ATmega328), pero no incluye un convertidor Serial-USB integrado. Por lo tanto, para programarlo, es necesario utilizar un cable FTDI externo. Esta característica lo hace una opción más económica en comparación con el Arduino UNO, especialmente cuando se planea utilizar varias tarjetas simultáneamente, ya que un solo cable FTDI puede ser usado para programar todas las tarjetas. (Shertz, 2023).</p>

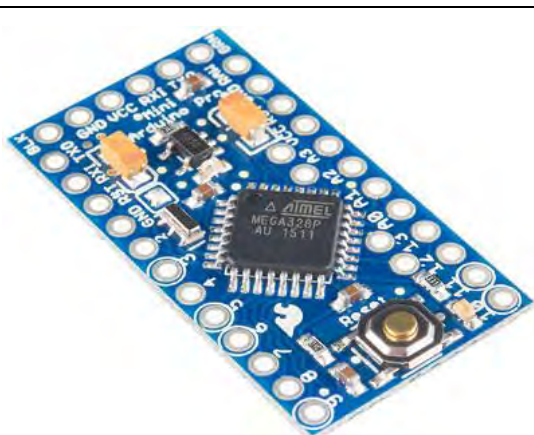


Figura 8. *Arduino Pro Mini*

Pro Mini, esta tarjeta es la versión miniatura del Arduino Pro, pensado para aplicaciones en las que el espacio sea muy reducido.

Requiere un cable FTDI para programarse y puede encajar en un protoboard si se sueldan pines de espaciamiento de 0.1 pulgadas.



Figura 9. *Leonardo*

Leonardo, esta tarjeta tiene una forma muy similar a la tarjeta Uno, así que es compatible con todos los Shields (a excepción de algunos por incompatibilidad de ciertos pines).

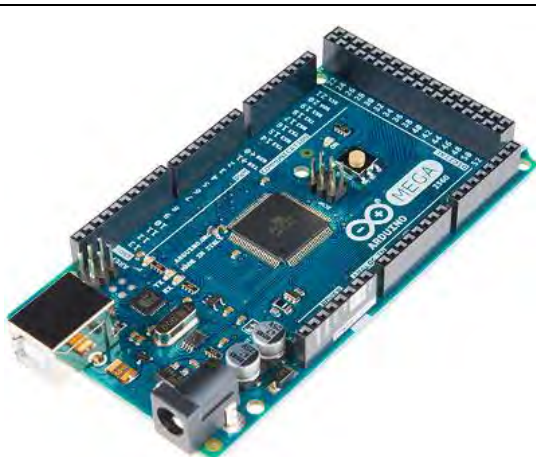
La gran ventaja de esta tarjeta es que está basada en el procesador ATmega32u4, el cual tiene comunicación USB integrada, por lo que no requiere de un convertidor Serial-USB ni de un cable FTDI para programarse, además de ser un poco más económico que el Arduino Uno por requerir menos componentes.

Gracias a sus capacidades USB integradas, puede emular las funciones de un mouse y un teclado. Esto permite realizar acciones como clic, doble clic, scroll o enviar texto de manera directa entre el Arduino y la computadora (Shertz, 2023).



Figura 10. *Micro*

Micro, esta tarjeta es la versión en miniatura del Arduino Leonardo, por lo que cuenta con sus mismas capacidades, tales como comunicación USB nativa y emulación de Mouse/Teclado, además de permitirte utilizarlo junto a un protoboard y reducir en gran medida el tamaño de tu circuito. (Shertz, 2023).

Figura 11. *Mega*

Mega, esta tarjeta es una excelente opción si necesitas potencia y, al mismo tiempo, compatibilidad con shields y código diseñado para Arduino Uno. Esta tarjeta cuenta con una cantidad mucho mayor de I/O que el Arduino Uno (54 vs 14 del Uno), además de tener 14 salidas PWM, 4 puertos UART, I2C y 16 entradas analógicas.

Además, tiene una memoria de mayor capacidad lo que te permitirá utilizarlo para códigos muy extensos o que requieran de una gran cantidad de variables (Shertz, 2023).

Figura 12. *Mega ADK*

Mega ADK, esta tarjeta es compatible pin a pin con el Arduino Mega, con la gran ventaja de incluir una interfaz Host USB, que te permite conectar tu Arduino a un teléfono con SO Android y comunicarte con él para acceder a la información de sus sensores o recibir órdenes de tu celular para controlar motores, LEDs, etc.

Con esta tarjeta puedes incluso recargar tu teléfono celular si requiere menos de 750mA y tienes una fuente de suficiente capacidad de corriente (Shertz, 2023).

Figura 13. *Due*

Due, esta tarjeta está basada en un procesador ARM-cortex de 32-bits a 84MHz, es la tarjeta Arduino con mayor capacidad y velocidad de procesamiento de esta lista.

Tiene un footprint similar al del Arduino Mega, pero tiene periféricos adicionales, como 2 convertidores DA y 2 pines de comunicación CAN.

Nota: A diferencia de otras tarjetas Arduino, esta tarjeta trabaja a 3.3V por lo que un voltaje mayor a 3.3v en sus pines de I/O puede dañar la tarjeta (Shertz, 2023).



Figura 14. *Ethernet*

Ethernet, esta tarjeta está basada en el microcontrolador ATmega328, al igual que el Arduino Uno, pero cuenta además con capacidad de conectarse a una red vía su puerto Ethernet. Cuenta además con un slot para tarjetas SD por lo que podrás guardar una gran cantidad de información y utilizarla cuando lo requieras, aún después de haber reseteado la tarjeta.

Nota: Se requieren de los pines 10 – 13 para la comunicación Ethernet, por lo que no deberán usarse para otros propósitos y el número de pines digitales disponibles se reduce a 9, contando con 4 salidas PWM (*Shertz, 2023*).

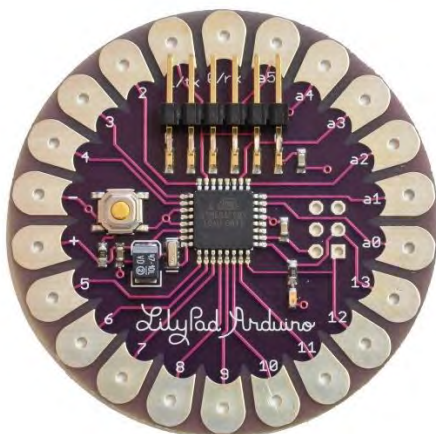


Figura 15. *LilyPad*

LilyPad, la tarjeta LilyPad está diseñada para ser utilizado en la ropa para crear e-textiles.

Tiene pines especiales para coser sensores y actuadores utilizando un hilo conductor (*Shertz, 2023*).



Figura 16. *Esplora*

Esplora, el Arduino Esplora está basado en el Leonardo y difiere de las anteriores tarjetas en que cuenta con un número de sensores integrados (de luz, temperatura, acelerómetro, joystick etc.) listos para usarse para interacción.

Está diseñado para gente que quiere tomarlo y comenzar a programar su Arduino sin tener que aprender electrónica previamente (*Shertz, 2023*).

¿Por Qué Usar Arduino?

Existen numerosos microcontroladores y plataformas de desarrollo disponibles, sin embargo, Arduino se destaca por no solo simplificar la tarea de programación, sino también por proporcionar:

- **Software Multiplataforma:** Puede trabajar en todas las plataformas (Mac, Windows, Linux).
- **Asequible:** Puedes encontrar placas por menos de 15 euros.
- **Entorno de programación simple y directo:** De fácil aprendizaje y uso, especialmente para principiantes.
- **Sencillez:** Es sencillo replicar y adaptar las placas, además es legal, ya que es Open Hardware y se distribuye bajo la licencia Creative Commons. Puedes ensamblarlo para crear tu propia placa, aunque no podrás utilizar el nombre "Arduino", que está registrado para las placas originales fabricadas en Italia.
- **Flexible:** Añadirle shields (módulos) en función del uso que se le vaya a requerir (conexión a Internet, control de motores, etc.) es muy fácil, y dispones de una gran cantidad de ellos para su compra online.
- **Software ampliable** y de código abierto, bajo licencia Creative Commons.

Arduino es una plataforma de código abierto que combina de manera sencilla el hardware y el software para la elaboración de proyectos electrónicos. Esta plataforma incluye una placa de circuito programable, comúnmente conocida como microcontrolador, y un software o entorno de desarrollo integrado (IDE) que se instala en tu computadora. A través de este IDE, puedes escribir y cargar el código en la placa física.

Las placas de Arduino pueden recibir diferentes tipos de señales, como la detección de luz a través de un sensor o la activación de un botón, y transformarlas en acciones, como el funcionamiento de un motor o la iluminación de un LED.

En resumen, puedes indicarle a la placa qué acciones realizar mediante el lenguaje de programación de Arduino (basado en Wiring) y el software de Arduino (IDE, basado en Processing). Con el tiempo, Arduino ha evolucionado para atender a una audiencia más amplia, adaptándose a diversas necesidades y desafíos. Ha ampliado su gama de productos, desde placas básicas de 8 bits hasta soluciones enfocadas en el Internet de las cosas (IoT), dispositivos portátiles, impresión 3D y sistemas embebidos. En conclusión, Arduino es una plataforma de desarrollo flexible, ideal para proyectos de automatización programable y hardware, que facilita la realización de ideas tanto a nivel personal como industrial. (Kurniawan, 2018).

2.2.4.2. Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, conocidas como variables de instrumentación, y convertirlas en variables eléctricas (Bentley, 2005). Estas variables de instrumentación pueden ser, por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, entre otras. Las magnitudes eléctricas pueden manifestarse como resistencia eléctrica (como en un termistor), capacidad eléctrica (como en un sensor de humedad), tensión eléctrica (como en un termopar), corriente eléctrica (como en un fototransistor), entre otras. (Fraden, 2016).

2.2.4.3. Actuadores

Los actuadores son dispositivos que convierten diferentes formas de energía en movimiento dentro de un sistema automatizado específico. Usualmente, transforman

energía eléctrica, neumática o hidráulica en fuerza mecánica para realizar diversas acciones.

Los actuadores electrónicos, por ejemplo, son controlados mediante corrientes eléctricas. Algunos de estos actuadores consumen una cantidad considerable de energía, por lo cual se emplean controladores especializados para gestionar su funcionamiento eficiente. (330ohms, 2023).

2.2.5. Software

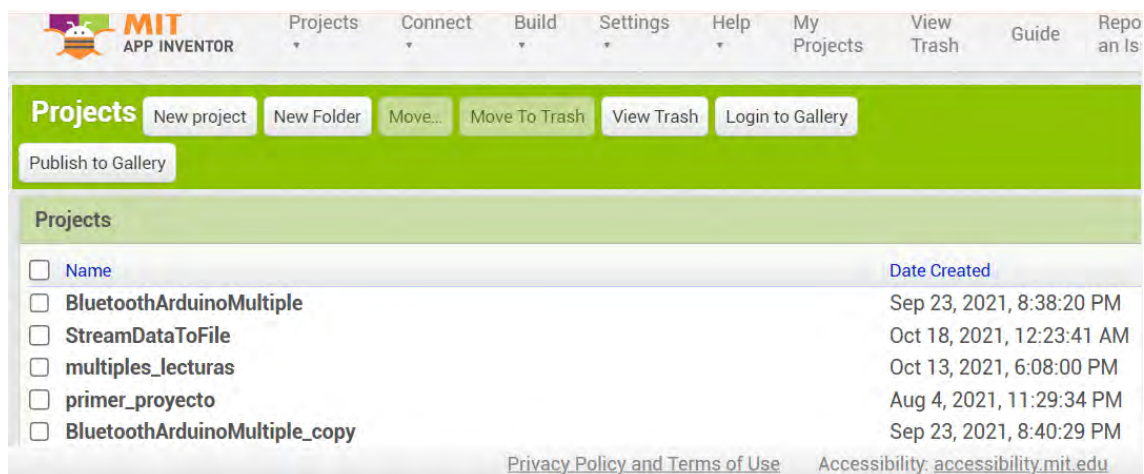
2.2.5.1. App Inventor

App Inventor, es una aplicación web de código abierto originalmente desarrollada por Google y mantenida ahora por el Instituto de Tecnología de Massachusetts (MIT). Utiliza una interfaz gráfica, muy similar a Scratch y la interfaz de usuario StarLogo TNG, que permite a los usuarios arrastrar y soltar objetos visuales para crear una aplicación que puede ejecutarse en dispositivos Android. Al crear App Inventor, Google se basó en una importante investigación previa en computación educativa, así como en el trabajo realizado en Google en entornos de desarrollo en línea. (MIT App Inventor. 2023)

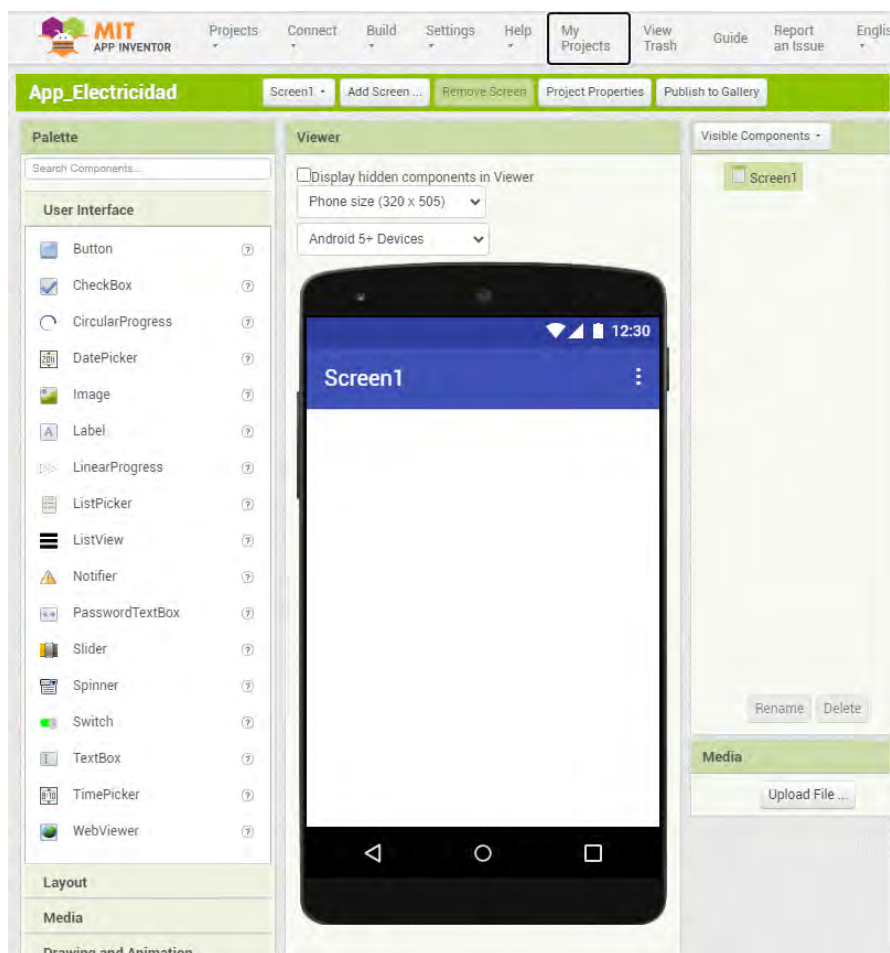
Interface App Inventor

Está formado por tres herramientas:

1. **Gestor de proyectos (My Projects)**. Donde se guardan todos nuestros proyectos. La ventana de los proyectos se carga apenas se accede al App Inventor, con esta herramienta podemos hacer un seguimiento a todos nuestros proyectos.

Figura 17.*Gestor de Proyectos App Inventor*

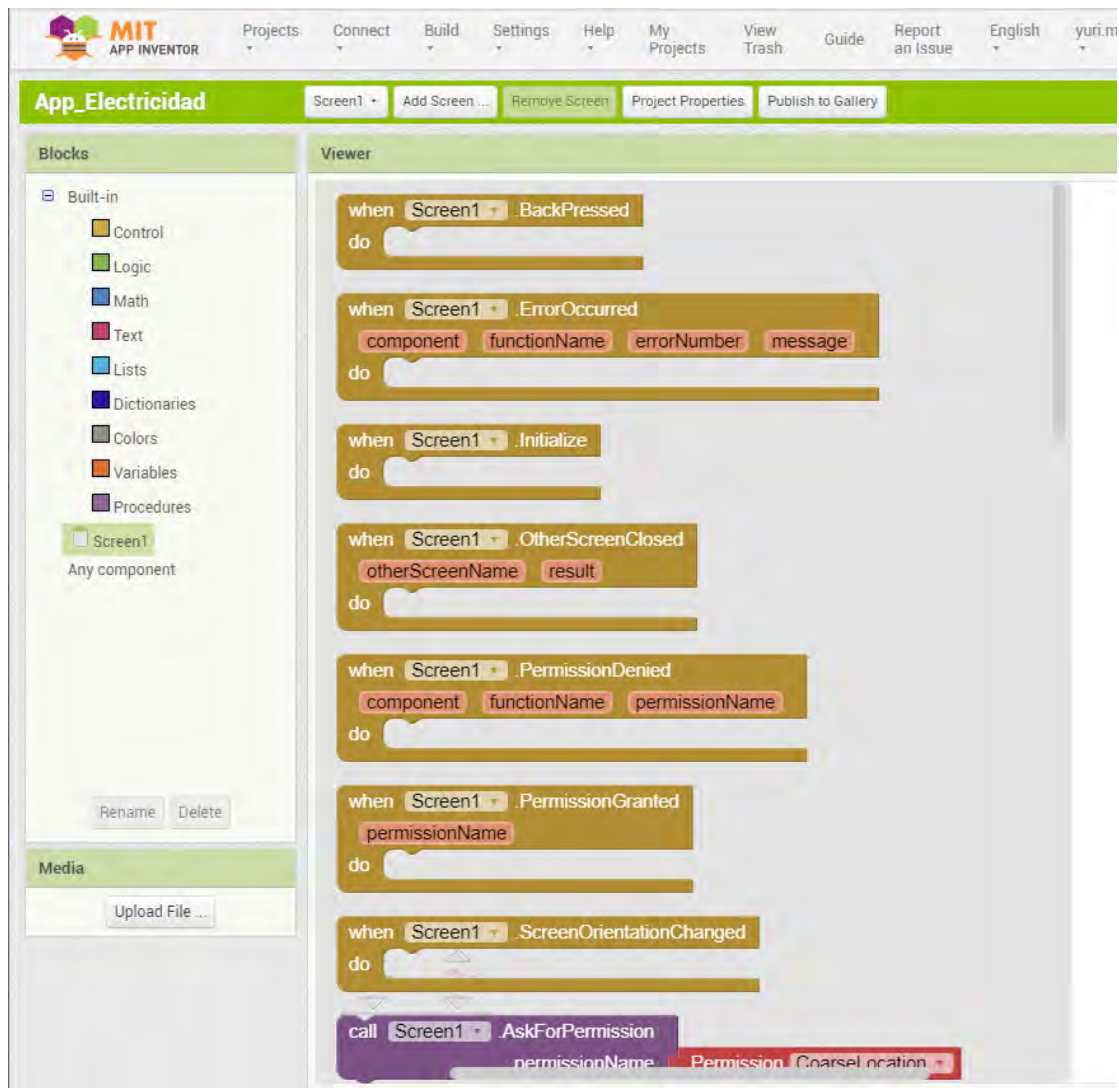
2. **Diseñador (Designer).** Esta herramienta se utiliza para diseñar la interfaz de las aplicaciones. Permite seleccionar componentes y definir la interfaz de usuario de la aplicación, incluyendo botones, etiquetas, cuadros de texto, y otros controles.

Figura 18.*Diseñador de Proyectos App Inventor*

3. **Editor de Bloques (Blocks).** En esta herramienta se programan las acciones que se realizarán con la interfaz previamente diseñada. Permite ensamblar bloques de programa que especifican cómo deben comportarse los componentes.

Figura 19.

Herramienta blocks App Inventor



Probando las Aplicaciones

Para observar el progreso de las aplicaciones que se están desarrollando en un dispositivo mientras se sigue construyendo (también conocido como 'Prueba en vivo'), existen las siguientes opciones.

- **Conexión Wifi.** Es la opción más recomendada cuando se tiene un ordenador, un dispositivo Android y una conexión Wifi, es la forma más sencilla de probar las aplicaciones. (MIT App Inventor. 2023)

Figura 20.

Conexión Wifi



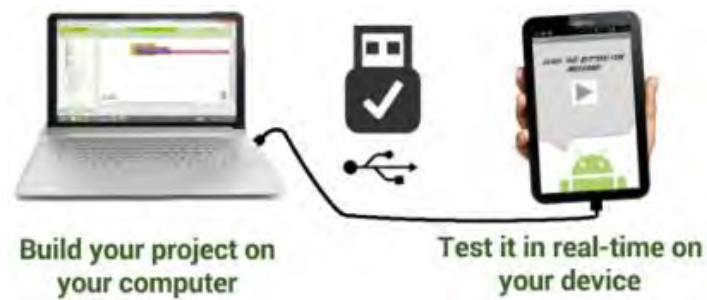
- **Emulador en App Inventor.** Incluso si no se dispone de un dispositivo Android, es posible desarrollar aplicaciones con App Inventor. Esta plataforma ofrece un emulador de Android que se muestra en la pantalla del ordenador. De esta manera, se pueden probar las aplicaciones en el emulador antes de distribuirlas a otros dispositivos. (MIT App Inventor. 2023)

Figura 21.

Emulador



- **Cable USB.** En caso de no contar con conexión Wifi, se puede utilizar un cable USB para realizar pruebas.

Figura 22.*Conexión mediante USB*

2.2.5.2. Java Script y HTML

JavaScript (JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript (Wikipedia C. , JavaScript, 2023). JavaScript se utiliza principalmente del lado del cliente (*client-side*), es decir, se ejecuta en el ordenador y no en el servidor; esto permite crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

Algunas personas tienden a confundir JavaScript con Java, aunque son lenguajes distintos con particularidades únicas. JavaScript puede integrarse en cualquier página web y ejecutarse sin necesidad de instalar software adicional para su visualización. Por otro lado, Java se destaca por ser un lenguaje independiente de la plataforma, lo que significa que los programas desarrollados en Java pueden ejecutarse en diversos sistemas operativos como Linux, Windows, y Apple. Además de su versatilidad, Java es ampliamente utilizado en el ámbito de internet. En resumen, JavaScript se caracteriza por ser un lenguaje interpretado y basado en prototipos, mientras que Java se orienta más hacia la programación orientada a objetos.

Principalmente, JavaScript se emplea para desarrollar páginas web con funciones como la creación de efectos como texto dinámico, animaciones, interacciones activadas por botones y mensajes emergentes para el usuario. Desde el punto de vista técnico, JavaScript es un lenguaje de programación interpretado, lo que significa que no requiere de una etapa de compilación para su ejecución. Los programas escritos en JavaScript pueden ser probados directamente en cualquier navegador sin la necesidad de realizar procesos adicionales. (MDN Web Docs. 2023)

Librerías JavaScript.

En general una librería es un archivo o conjunto de archivos que se usan para facilitar la programación. Las librerías en JavaScript consisten en archivos de código escritos en JavaScript que se insertan al principio de la página.

Se puede insertar cualquier archivo al principio de la página, sin embargo, no todos los archivos pueden considerarse librerías. Para que un archivo sea una librería, este tiene que poder usarse en distintas páginas, no sólo del mismo sitio, sino de varios sitios. Además, debe modificar el código de tal manera que nos facilite la inserción de nuevo código en la página.

La librería más conocida de JavaScript se llama JQuery, que permite simplificar la forma de interactuar con los documentos HTML, una de las ventajas es que es software libre y de código abierto, lo que permite su uso en proyectos libres y privados. Ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código.

Una librería JavaScript para la visualización de datos con HTML, que permite la creación de múltiples tipos de gráficos es *Chart.js*, en su versión 2.4, permite la

representación de nueve tipos de graficas diferentes, totalmente animadas y personalizables. Funciona con HTML5 por lo que soporta todos los navegadores web modernos incluidos los dispositivos móviles; no depende de librerías externas, es ligero y es una librería open source bajo Licencia MIT, su documentación es muy completa con muchos ejemplos. (Chart.js, 2023)

Para el presente trabajo se decidió usar esta librería por su facilidad para obtener los datos de archivos JSON (*JavaScript Object Notation*) que es un formato de texto ligero para el intercambio de datos, por la estética con que presenta los gráficos, su facilidad de uso, la amplia documentación con que cuenta y finalmente porque podemos utilizarlos en archivos HTML que funcionan offline sin conexión a internet.

2.3. MARCO CONCEPTUAL – DEFINICIÓN DE TÉRMINOS BÁSICOS

A. MODULO DIDÁCTICO:

Un módulo educativo, que también se conoce como módulo de enseñanza o módulo didáctico, es un recurso interactivo que engloba todos los componentes esenciales para que el estudiante adquiera conocimientos y habilidades a su propio ritmo, prescindiendo de una presencia constante del maestro, profesor o instructor. (Urrutia, 2020)

B. EXPERIMENTO:

Un experimento se refiere a un procedimiento diseñado para respaldar, refutar o validar una hipótesis. Los experimentos aportan información acerca de relaciones causa-efecto al mostrar qué resultados se obtienen cuando se manipula un factor específico. Los experimentos pueden variar significativamente en términos de sus objetivos y alcance, pero todos ellos se basan en la repetición de pasos y un análisis lógico de los resultados.

Los experimentos pueden ser realizados por individuos, incluso niños, para comprender conceptos básicos, como el efecto de la gravedad, o por equipos de científicos que pueden llevar a cabo investigaciones sistemáticas durante años para avanzar en su comprensión de un fenómeno. Los experimentos y actividades prácticas son elementos de gran relevancia en el aprendizaje de los estudiantes, especialmente en el entorno educativo de las ciencias. Estas actividades pueden mejorar el rendimiento académico y estimular el interés de los estudiantes en el material que están estudiando, sobre todo cuando se aplican de manera regular. (Montgomery,2020)

C. MICROCONTROLADOR:

Un microcontrolador es un chip integrado que se puede programar para ejecutar instrucciones almacenadas en su memoria. Este dispositivo contiene las tres partes esenciales de una computadora: la unidad central de procesamiento (CPU), las unidades de memoria (RAM y ROM), puertos de entrada y salida, y periféricos. Estas unidades se encuentran interconectadas en el interior del microcontrolador, formando lo que se conoce como una microcomputadora. La función principal de los microcontroladores es leer y llevar a cabo los programas escritos por el usuario. (Angulo, U., Romero, Y. y Angulo, M.,2000)

D. SENSOR:

Se trata de un dispositivo con la capacidad de identificar magnitudes físicas o químicas y convertirlas en otras magnitudes, generalmente de naturaleza eléctrica, que pueden medirse y manipularse. Las magnitudes físicas pueden incluir intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, entre otras. La magnitud eléctrica resultante puede manifestarse como una resistencia eléctrica (como en el caso de un termistor), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en

un termopar), una corriente eléctrica (como en un fototransistor) y similares. (Fraden, 2016).

E. ACTUADOR:

Un actuador es un mecanismo que tiene la capacidad de convertir energía proveniente de fuentes hidráulicas, neumáticas o eléctricas en acciones específicas destinadas a afectar un proceso automatizado. Este dispositivo recibe instrucciones de un regulador o controlador y, en función de estas directrices, emite comandos para poner en funcionamiento un componente final de control, como, por ejemplo, una válvula. Los actuadores son componentes que ejercen una influencia directa sobre la señal de salida del sistema automatizado, ajustando su intensidad de acuerdo con las indicaciones que provienen de la unidad de control. (330ohms, 2023)

F. SISTEMA DE ADQUISICIÓN DE DATOS:

Un sistema de adquisición de datos (DAQ) es un conjunto de dispositivos y software diseñados para capturar, procesar y almacenar señales provenientes de sensores y otros dispositivos físicos. Este sistema permite convertir señales físicas, como temperatura, presión, flujo, luz, entre otras, en señales digitales que pueden ser procesadas por una computadora u otro dispositivo de procesamiento.

Los componentes principales de un sistema de adquisición de datos incluyen:

Sensores o transductores, circuito de acondicionamiento de señal, Convertidor Analógico-Digital (ADC), unidad de procesamiento, software de control y análisis e Interfaces de comunicación

Los sistemas de adquisición de datos se utilizan en una amplia gama de aplicaciones, como control industrial, monitoreo ambiental, investigación científica, medicina, entre otras, donde es crucial la captura precisa y continua de datos del entorno físico para análisis y toma de decisiones. (National Instrument, 2023)

G. ORDENADOR PERSONAL:

En esencia, se trata de una microcomputadora concebida originalmente para el uso individual y que es compatible con el PC de IBM. Un computador personal por lo general es de tamaño mediano y está destinado a ser utilizado por una sola persona a la vez (aunque existen sistemas operativos que permiten la interacción de múltiples usuarios simultáneamente, lo que se conoce como multiusuario). Suele estar equipado para llevar a cabo las tareas comunes de la informática contemporánea, como realizar trabajos de oficina y navegar por Internet, entre otras funciones. (Tanenbaum y Bos, 2014)

H. DISPOSITIVO MÓVIL:

Se refiere a un dispositivo electrónico compacto, a menudo denominado computadora de mano o dispositivo de bolsillo, que posee capacidad de procesamiento y puede conectarse a Internet de manera constante o intermitente. Estos dispositivos suelen tener una capacidad de almacenamiento limitada y han sido diseñados principalmente para una función específica, aunque también pueden llevar a cabo otras tareas. Entre los ejemplos de dispositivos móviles se incluyen las computadoras móviles, los teléfonos inteligentes, las tabletas, los relojes inteligentes, entre otros. (Gibson,2020).

I. ANDROID:

Se trata de un sistema operativo móvil creado por Google, que se fundamenta en el núcleo Linux y está principalmente destinado a dispositivos móviles con pantallas táctiles, como smartphone y Tablet. Este sistema es de código abierto, lo que implica que su código puede ser modificado, y ofrece a los usuarios un amplio catálogo de aplicaciones que pueden descargarse de manera gratuita o realizando un pago. (Baker y St. Laurent, 2011).

J. PÉNDULO SIMPLE:

Un péndulo simple es un modelo idealizado en física para estudiar el movimiento oscilatorio. Consiste en una masa puntual suspendida en un hilo inextensible y de masa despreciable, fijado en un punto fijo. Este modelo permite analizar el comportamiento de los movimientos oscilatorios y se aplica en diversas áreas de la física para comprender y predecir fenómenos oscilatorios. (Schnick, 2007).

K. CAÍDA LIBRE:

La caída libre describe el movimiento de un objeto que está bajo la influencia exclusiva de la gravedad, sin la acción de otras fuerzas, como la resistencia del aire. Este modelo idealizado facilita el estudio de cómo los objetos caen en un campo gravitacional uniforme. (Drake, 1989)

L. MOVIMIENTO RECTILÍNEO UNIFORME:

El Movimiento Rectilíneo Uniforme (MRU) describe el desplazamiento de un objeto a lo largo de una línea recta con una velocidad constante. En este tipo de movimiento, el objeto se desplaza sin experimentar ninguna aceleración, manteniendo una velocidad constante durante todo el trayecto. (Serway y Jewett, 2018)

M. RESISTENCIA ELÉCTRICA:

La resistencia eléctrica es una propiedad fundamental de los materiales que describe la oposición que ofrecen al flujo de corriente eléctrica. Se mide en ohmios (Ω) y es esencial para entender cómo los circuitos eléctricos y electrónicos funcionan. (Giancoli, 2008)

N. CAPACITANCIA ELÉCTRICA:

La capacitancia eléctrica es una medida de la capacidad de un componente o sistema para almacenar carga eléctrica. Se mide en faradios (F) y es una propiedad fundamental en circuitos electrónicos, especialmente en componentes como los condensadores. (Giancoli, 2008)

2.4. HIPÓTESIS

2.4.1. Hipótesis General

Utilizando las tarjetas electrónicas Arduino se diseñarán y construirán módulos automatizados para el desarrollo de experimentos de Física en Mecánica y Electricidad.

2.4.2. Hipótesis Específicas

1. El diseño y la construcción del hardware integrará eficientemente los periféricos y accesorios necesarios para la implementación óptima de experimentos de Física, en las áreas de Mecánica y Electricidad.
2. El diseño e implementación del software controlará y operará eficientemente el funcionamiento del hardware de los módulos.
3. El desarrollo de los experimentos propuestos de Física en Mecánica y Electricidad, utilizando el sistema hardware-software, comprobará y validará su funcionalidad.

2.5. IDENTIFICACIÓN DE VARIABLES E INDICADORES

Debido a la naturaleza de la investigación aplicada, centrada en el diseño y construcción de prototipos de módulos para automatizar ciertos procesos experimentales en los laboratorios de Física General, tanto a nivel de hardware como de software, no fue posible identificar variables e indicadores específicos.

2.6. OPERACIONALIZACIÓN DE LAS VARIABLES

Como consecuencia de la imposibilidad de identificar las variables, no es factible su operacionalización.

CAPÍTULO III: METODOLOGÍA

3.1. TIPO Y DISEÑO DE INVESTIGACIÓN

Este trabajo de investigación es de tipo aplicada orientada al diseño y construcción de un prototipo de módulo automatizado para la realización de experimentos en Física General. El proceso sigue una serie de etapas específicas, destinadas a convertir una idea inicial en una solución tangible. Estas etapas abarcan desde la identificación del problema hasta la evaluación del prototipo final. A continuación, se describen las fases típicas de este tipo de investigación:

1.- Identificación del Problema y Definición de Requisitos

Definición del problema:

Este primer paso consistió en identificar y clarificar el problema o la necesidad que se busca resolver con el prototipo. Fue crucial entender el contexto en el que surge esta necesidad, ya sea a partir de dificultades actuales o de oportunidades de mejora en un proceso o sistema existente. En esta etapa, se llevó a cabo investigaciones preliminares para analizar la magnitud del problema, sus causas y las posibles consecuencias de no abordarlo, lo que permitió definir el objetivo del prototipo.

Requisitos y especificaciones:

Una vez definido el problema, se procedió a establecer los requisitos y especificaciones que nos guiaron en el diseño del prototipo. Estos incluyeron aspectos técnicos, funcionales y estéticos que el prototipo debe cumplir para ser efectivo. Entre los parámetros a definir se encuentran: el tamaño, los materiales a utilizar, las funcionalidades, el costo estimado y el rendimiento esperado. Esta etapa sentó las bases para el diseño y desarrollo del prototipo, asegurando que las soluciones propuestas sean viables y adecuadas a las necesidades planteadas.

2.- Revisión de la Literatura e Investigación Preliminar

Investigación de soluciones existentes:

En esta etapa, se realizó una revisión de las tecnologías y soluciones que ya existen y que guardan similitudes con el prototipo que se desea desarrollar. Este análisis nos permitió identificar si hay productos o dispositivos previos que puedan servir como referencia o inspiración. Además, ayuda a evitar la duplicación de esfuerzos al trabajar sobre conceptos ya desarrollados y abre la posibilidad de implementar mejoras en áreas que aún presenten deficiencias.

Estudio de materiales y tecnologías:

Esta fase implicó una investigación sobre los materiales y las tecnologías que podrían ser utilizados en el diseño y la construcción del prototipo. Fue fundamental elegir los materiales adecuados en función de los requisitos de resistencia, durabilidad, costo, sostenibilidad y disponibilidad. Igualmente, se evaluaron las tecnologías de fabricación y ensamblaje más idóneas, desde impresoras 3D hasta métodos de mecanizado o técnicas automatizadas, dependiendo de la complejidad del prototipo. La elección correcta de los materiales y las tecnologías influirá significativamente en el rendimiento, la eficiencia y el costo final del prototipo.

3.- Formulación de Objetivos y Diseño Inicial

Objetivos del prototipo:

Fue imprescindible establecer objetivos claros y concretos para el desarrollo del prototipo. Estos se alinearon con las necesidades identificadas durante la fase de definición del problema, orientando todo el proceso de diseño y construcción. Dichos objetivos incluyeron metas como mejorar la funcionalidad de un sistema existente, reducir costos de producción o implementación, optimizar la eficiencia energética, perfeccionar la ergonomía y facilitar el uso y mantenimiento del producto.

Diseño conceptual:

Una vez definidos los objetivos, se inició la fase de diseño conceptual, donde se esbozaron las primeras ideas visuales y técnicas del prototipo. Esta etapa incluyó la creación de bocetos, diagramas de flujo, planos y representaciones gráficas en 2D y 3D, que facilitaron la visualización de las diferentes soluciones propuestas antes de avanzar al desarrollo físico. Para lograr representaciones más detalladas y precisas, se emplearon herramientas de diseño asistido por computadora (CAD). Gracias al diseño conceptual, fue posible identificar posibles problemas antes de construir el prototipo, lo que permitió optimizar el uso de tiempo y recursos.

4.- Desarrollo del Prototipo**Diseño detallado:**

En esta etapa, se elaboraron especificaciones técnicas y planos que incluyeron dimensiones, tolerancias, materiales, y detalles sobre cómo se ensamblarán los componentes. También se definieron los sistemas eléctricos, mecánicos y electrónicos necesarios para el correcto funcionamiento del prototipo. El diseño contempló los aspectos de construcción, el uso de herramientas especializadas.

Construcción del prototipo:

Una vez finalizado y aprobado el diseño detallado, se procedió a la fase de construcción del prototipo. Se seleccionaron y adquirieron los materiales adecuados, y se utilizaron las técnicas de construcción previamente identificadas, que incluyeron programas CAD para el diseño de circuitos electrónicos, impresión 3D y corte láser. La construcción implicó tanto la creación de componentes individuales como su ensamblaje en una unidad funcional, además de la integración de sensores, microcontroladores y sistemas de control automatizado. Se verificó cada paso del

proceso para garantizar que el prototipo se construyera conforme a las especificaciones técnicas y cumpliera con los requisitos establecidos en fases anteriores.

5.- Pruebas y Evaluación

Pruebas iniciales:

Una vez construido el prototipo, se procedió a realizar pruebas iniciales para evaluar su rendimiento en relación con los requisitos y especificaciones previamente definidos. Se probaron sus funcionalidades con la finalidad de verificar que el prototipo realiza las tareas previstas y se llevaron a cabo pruebas de seguridad para garantizar que no presenta riesgos para los usuarios. Además, se identificaron posibles fallos, limitaciones o ajustes necesarios.

Evaluación de resultados:

En esta etapa, se analizaron los datos recopilados durante las pruebas para identificar cualquier problema técnico o de rendimiento que pueda requerir corrección. Asimismo, se revisaron aspectos como la funcionalidad, fiabilidad, eficiencia y seguridad, para asegurarnos de que el prototipo cumple con los objetivos establecidos al inicio del proyecto.

6.- Técnicas de recolección de información

a) Entrevistas estructuradas o semiestructuradas

Objetivo: Recopilar información cualitativa de expertos en Física, docentes y jefes de práctica.

Contenido:

Preguntas sobre los experimentos más comunes y útiles en la enseñanza de Mecánica y Electricidad.

Requerimientos técnicos y pedagógicos de los módulos.

Limitaciones actuales de los equipos existentes.

Aplicación: Entrevistas individuales a docentes, jefes de practica y estudiantes.

b) Observación directa

Objetivo: Analizar el contexto en el que se realizarán los experimentos.

Acciones:

Observar las prácticas de laboratorio para identificar dinámicas, limitaciones y necesidades.

Registre cómo los usuarios interactúan con los equipos actuales.

Identificar errores comunes o dificultades en la realización de los experimentos manuales.

7.- Técnicas de análisis e interpretación de la información

Análisis de Requerimientos y Necesidades

a) Análisis de Necesidades del Usuario

Objetivo: Identificar las expectativas y necesidades de los usuarios (estudiantes, jefes de practica) que utilizarán los módulos automatizados.

Método:

Análisis cualitativo: Interpretar los resultados de entrevistas y observaciones para detectar patrones comunes sobre los problemas actuales y las expectativas para los módulos automatizados.

Clasificación de funcionalidades: Priorizar las funcionalidades que los usuarios más requieren en los módulos, como la facilidad de uso, la precisión en la medición, o la capacidad de realizar múltiples experimentos.

Aplicación: Identificar qué experimentos físicos de mecánica y electricidad son más relevantes para los usuarios y qué aspectos del diseño de los módulos deben enfatizarse (interfaz de usuario, tipos de sensores, etc.).

b) Análisis de Requerimientos Técnicos

Objetivo: Determinar las especificaciones técnicas necesarias para que los módulos automatizados puedan realizar los experimentos de manera adecuada.

Método:

Estudio de viabilidad técnica: Analizar las especificaciones de hardware y software necesarios, incluyendo los sensores, actuadores, circuitos, fuentes de voltaje y microcontroladores.

Revisión de estándares: Asegurarse de que el diseño cumpla con los estándares de seguridad y precisión requeridos.

8.- Técnicas para demostrar la veracidad o falsedad de las hipótesis planteadas

Pruebas Experimentales

Objetivo: Evaluar si los módulos automatizados funcionan según lo previsto bajo condiciones controladas.

Método:

Diseñar experimentos controlados en los que se simulan los escenarios para los que el módulo está destinado.

Comparar los resultados obtenidos con valores teóricos conocidos.

Aplicación:

Si el módulo automatizado para un experimento de mecánica, como el de caída libre, muestra resultados consistentes con las ecuaciones teóricas, se puede confirmar la veracidad de la hipótesis de que el módulo puede simular con precisión dicho experimento.

Para experimentos de electricidad, se validan las mediciones de voltaje y corriente, comparándolas con las de un multímetro convencional.

3.2. UNIDAD DE ANÁLISIS

En esta tesis no se analizó ningún objeto, sujeto o característica, ya que se diseñaron y construyeron prototipos. Por lo tanto, este concepto no es aplicable

3.3. POBLACIÓN DE ESTUDIO

El presente trabajo, al ser una investigación aplicada en la que se diseñaron y construyeron prototipos, no requiere establecer una población de estudio. En consecuencia, no aplican los ítems subsiguientes en este capítulo de la estructura de tesis propuesta por la universidad.

CAPITULO IV: RESULTADOS Y DISCUSIÓN

4.1. DISEÑO E IMPLEMENTACIÓN DE LOS ELEMENTOS DE CONTROL Y DETECCIÓN PARA LOS MÓDULOS.

Los módulos que se expondrán en este trabajo han dado lugar al desarrollo de componentes de hardware y sistemas de control (interfaces) necesarios para la automatización de experimentos de Física General. Estos componentes, a su vez, deben ser combinados con software que habilite el control total del sistema. A continuación, se procederá a describir cada elemento de hardware y sus respectivas interfaces, proporcionando definiciones técnicas para cada uno de ellos y detallando el proceso de su creación.

4.1.1. Actuadores.

Cuando se realizan experimentos en el laboratorio que implican la medición del tiempo de cuerpos en movimiento, como la caída libre, el péndulo simple, el movimiento de proyectiles, entre otros; la precisión en la medición del tiempo inicial es un aspecto crítico que puede tener un impacto significativo en la interpretación de los resultados. Esto constituye la razón fundamental para la automatización del proceso de medición del tiempo inicial en los experimentos que se realizan con los prototipos. En este proyecto, se ha implementado un actuador con un electroimán que cumple la función de liberar el cuerpo en movimiento en los experimentos, permitiendo así que el movimiento comience de manera precisa y controlada.

En el caso del actuador, se han reutilizado solenoides⁴ reciclados provenientes de sistemas electrónicos, a los cuales se les ha proporcionado una carcasa adecuada y se les

⁴ Dispositivo físico que crea un campo magnético uniforme en su interior.

ha instalado un conector para la alimentación y el control. Para la carcasa, se ha empleado una roseta de red RJ45, fácilmente accesible localmente, y se ha adaptado para contener el solenoide y el conector hembra plug para la fuente de alimentación. Para el funcionamiento del actuador se utiliza la energía suministrada por la fuente de alimentación de la interfaz (12V), y la duración de su activación se regula mediante un circuito optoacoplador incorporado en la interfaz. La interfaz dispone de dos salidas designadas para la conexión de electroimanes, identificadas con indicadores luminosos de color rojo, provistos en el chasis de la interfaz para experimentos de mecánica. La figura 23 y figura 24 muestran los componentes para su construcción y una vista del actuador construido.

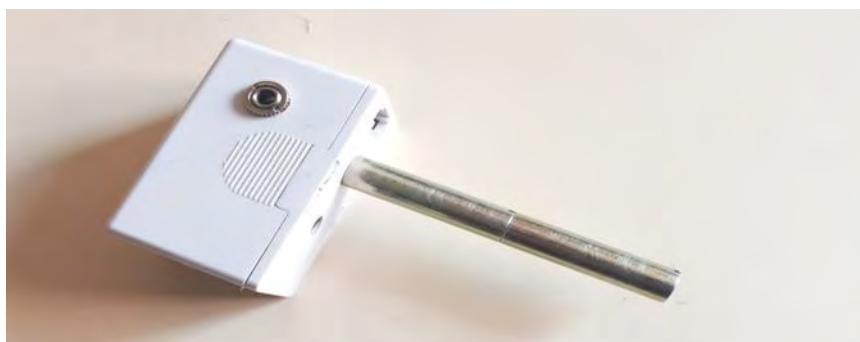
Figura 23.

a) Solenoides b) Roseta de Red Rj45, c) Conexiones internas del actuador



Figura 24.

Vista del actuador construido



4.1.2. Sensores.

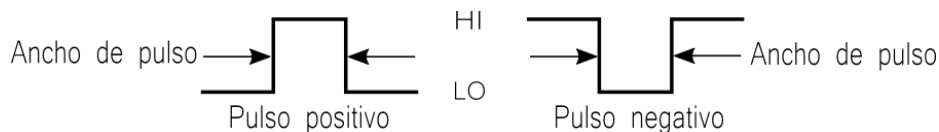
Fotopuertas.

En cualquier proceso de automatización, la utilización de sensores es esencial para recopilar información del entorno y transmitirla al sistema de control. En el contexto de los experimentos presentados, es crucial medir el tiempo que un objeto tarda en alcanzar un punto específico. Para abordar esta necesidad, se ha diseñado y construido un sensor fotopuerta (photogate en inglés), un dispositivo destinado a medir con alta precisión eventos de corta duración, especialmente aquellos de alta velocidad.

La fotopuerta se compone principalmente de dos elementos electrónicos: un emisor y un receptor de luz. Este dispositivo se activa cuando un objeto interrumpe el haz de luz que viaja desde el emisor al receptor. Estos componentes se instalan en una carcasa y se utilizan para investigar una variedad de conceptos físicos, como la caída libre, planos inclinados, carriles de aire, péndulos simples, entre otros.

El receptor de la fotopuerta emplea un fototransistor⁵ para medir la cantidad de luz proveniente de un haz de luz emitida por un LED. Debido a las fuentes de luz ambiental presentes en entornos normales de trabajo, se emplea un LED infrarrojo (LED IR) para minimizar la interferencia de señales de luz externa. Cuando un objeto bloquea el haz IR, el fototransistor cambia su estado de bajo a alto, marcando así el inicio del pulso. Posteriormente, cuando el haz IR se despeja, el fototransistor vuelve a su estado bajo inicial, indicando el final del pulso. La figura 25 ilustra la definición de los pulsos positivos y negativos.

⁵ Es un transistor activado por luz incidente en la región de base.

Figura 25.*Definición de pulsos*

En el circuito de la fotopuerta se utiliza un pulso positivo (LO-HI-LO). El tiempo transcurrido entre el inicio y el final del pulso representa la duración o ancho del pulso, y permite medir el tiempo durante el cual el haz IR estuvo bloqueado.

En la figura 26 se muestra el circuito esquemático y en la figura 27 el diseño del circuito impreso para la fotopuerta. Como componentes electrónicos se utilizó un fototransistor IR como receptor, un LED IR como emisor, un amplificador operacional para comparar la señal recibida por el fototransistor cuando se bloquea el haz IR con un voltaje de referencia, una resistencia variable de 10 k Ω para calibrar la sensibilidad de la fotopuerta y una resistencia de protección de 330 Ω para limitar el flujo de corriente hacia el LED IR. Para la conexión entre la fotopuerta y la interfaz se utilizó un conector de red telefónica (UTP RJ11 de 4 vías). La carcasa externa está equipada con un LED indicador que se enciende cuando el haz IR está bloqueado, permitiendo al usuario verificar el correcto funcionamiento del sensor.

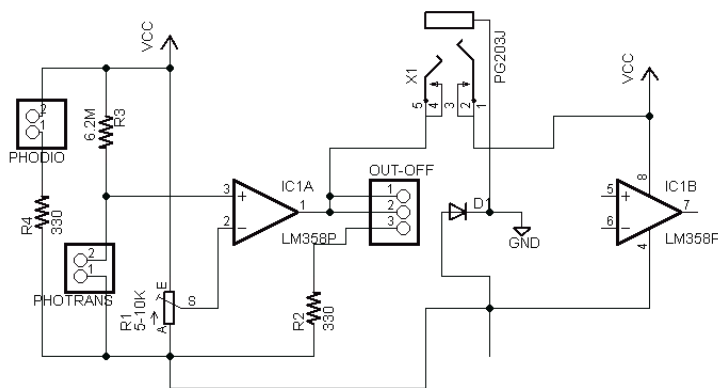
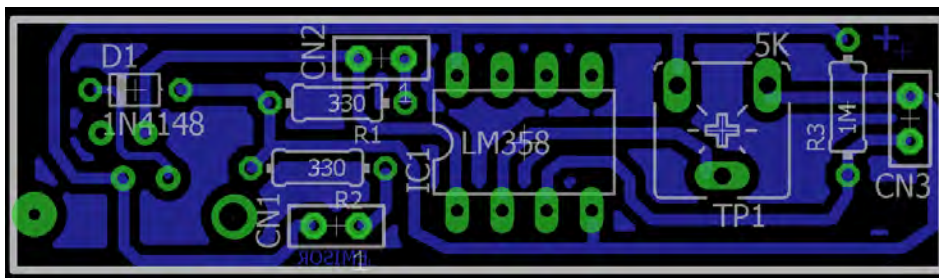
Figura 26.*Circuito esquemático del sensor fotopuerta*

Figura 27.

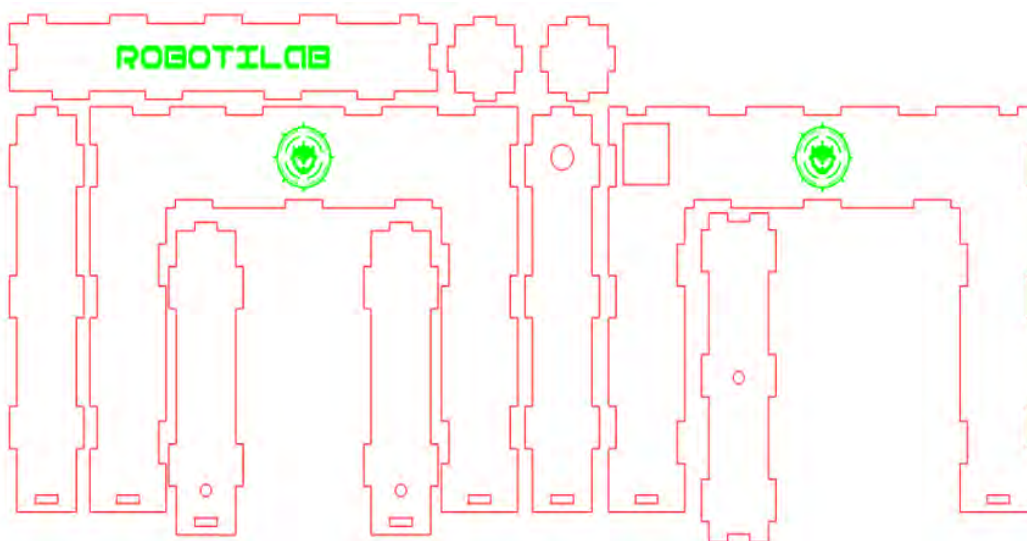
Diseño del circuito impreso de la fotopuerta



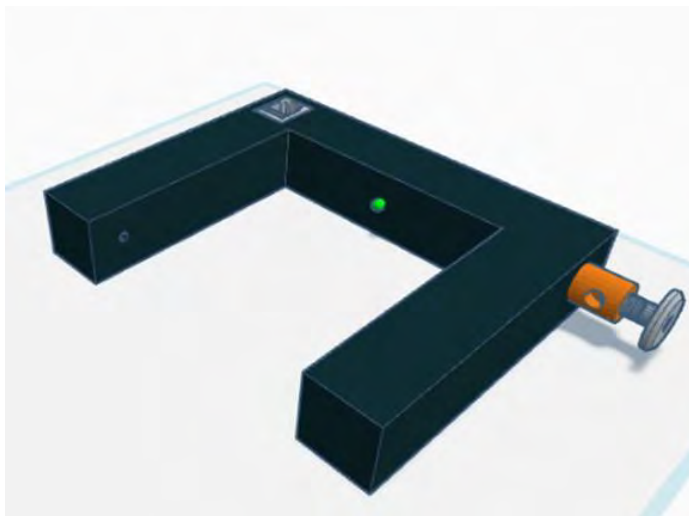
Para la construcción de la carcasa de la fotopuerta se utilizó acrílico. Se realizaron orificios en los extremos inferiores de la carcasa para colocar el emisor y el receptor LED IR. En la cara interior e inferior de la base del sensor se hizo un orificio para instalar un LED indicador, y en una de las caras laterales externas se colocó el conector Jack RJ11 para la conexión con la interfaz. El diseño de la carcasa se realizó utilizando un software CAD⁶ y el proceso de construcción de la fotopuerta se detalla en las figuras. 28, 29 y 30. Se utilizó una máquina de corte láser para obtener un acabado de alta precisión.

Figura 28.

Diseño CAD de la carcasa para el sensor fotopuerta



⁶ CAD (computer-aided design) diseño asistido por computadoras

Figura 29.*Diseño CAD para la Construcción de la Carcasa de la Fotopuerta***Figura 30.***Fotopuerta construida***Sensor de tiempo de vuelo con sensor piezoeléctrico**

Para la automatización del experimento de caída libre, utilizamos un sensor piezoeléctrico adherido a un acrílico de 3 mm de espesor, que constituye la superficie de impacto de las esferas metálicas. El impacto provoca una deformación en la superficie de contacto, la cual es detectada por el sensor piezoeléctrico. Debido a sus propiedades piezoeléctricas, el sensor genera una pequeña diferencia de potencial. Esta señal es luego acondicionada y amplificada digitalmente por un circuito integrado CD40106, el cual está

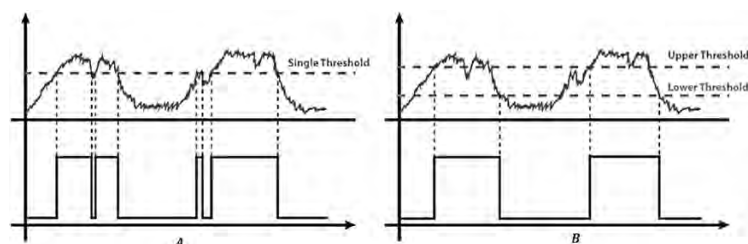
compuesto por 6 compuertas lógicas Schmitt Trigger. El circuito produce un pulso de señal cuadrada a la salida, lo que permite al sistema de control determinar el tiempo final de caída.

En el ámbito de la electrónica, un disparador Schmitt es un circuito comparador que utiliza retroalimentación positiva en la entrada no inversora de un comparador o amplificador diferencial. Este circuito convierte una señal de entrada analógica en una señal de salida digital. El término "disparador" se utiliza porque la salida mantiene su estado hasta que la entrada cambia lo suficiente para provocar un cambio en la salida. En la configuración no inversora, cuando la entrada supera un cierto umbral seleccionado, la salida es alta. Si la entrada desciende por debajo de otro umbral seleccionado (más bajo), la salida se vuelve baja, y si la entrada se encuentra entre los dos niveles, la salida mantiene su valor actual. Este efecto de doble umbral se llama histéresis, lo que implica que el disparador Schmitt tiene memoria y puede funcionar como un multivibrador biestable, también conocido como "latch" o "flip-flop". Existe una estrecha relación entre estos dos tipos de circuitos, ya que un disparador Schmitt se puede transformar en un "latch" y viceversa.

Los dispositivos de activación Schmitt se utilizan en aplicaciones de procesamiento de señales para filtrar el ruido de las señales que se emplean en circuitos digitales, especialmente en la corrección del rebote de contactos mecánicos en interruptores figura 31.

Figura 31.

Efecto del uso del Schmitt Trigger (B) Uso de un Comparador (A)



Nota. <https://madpcb.com/glossary/schmitt-trigger>

En la figura 32 se muestran el diseño del circuito impreso para el sensor con piezoeléctrico y en la figura 33 los componentes electrónicos y la carcasa que se utilizó.

Figura 32.

Diseño del PCB del Adaptar de la Señal del Piezoeléctrico a la Interface

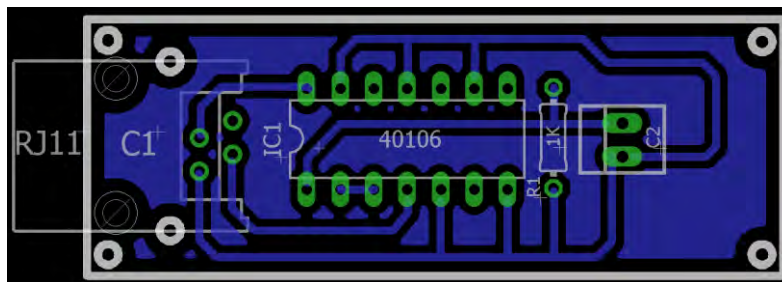
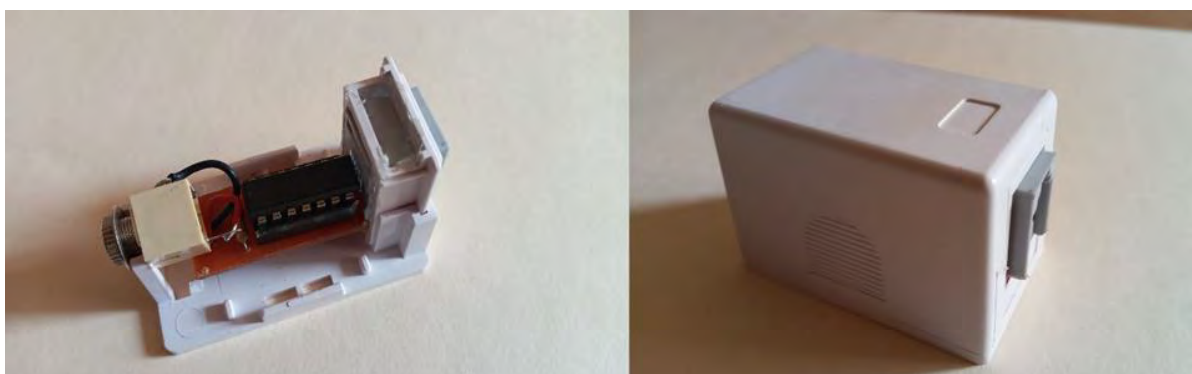


Figura 33.

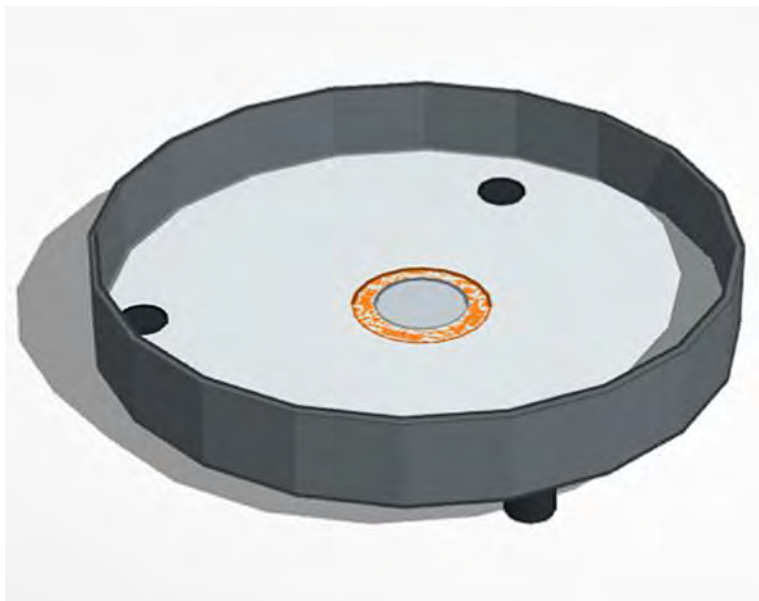
Adaptador para el Sensor con Piezoeléctrico con la Interface



Para la construcción de la carcasa del sensor con un piezoeléctrico, se utiliza acrílico junto con un tubo de cartón, cuya función principal es evitar que las esferas rueden. Además, se incorporaron tres gomas de jebe como patas para amortiguar los impactos. El sensor piezoeléctrico fue fijado con silicona en la parte central de la base, por su cara inferior, y se conectó a la interface mediante un circuito digital Schmitt Trigger integrado, el cual se encarga de reconformar la señal generada por el impacto y eliminar los rebotes. La figura 34 muestran el diseño en CAD y la figura 35 el sensor construido.

Figura 34.

Diseño CAD 3D del Sensor de Tiempo de Vuelo para el Experimento de Caída Libre

**Figura 35.**

Sensor de Tiempo de Vuelo, Experimento de Caída Libre



4.1.3. Carrito para el M.R.U.

Para el experimento de movimiento rectilíneo uniforme se diseñó y construyó un móvil con un solo motor para evitar la corrección de diferencias de velocidades, que podría ocurrir al utilizar dos motores. Se utilizó un Arduino Nano para el control de

velocidad mediante PWM (modulación por ancho de pulso), así como para el control mediante teclado y pantalla OLED. Esto nos permitió seleccionar entre tres velocidades y mostrar un conteo descendente al inicio del movimiento, activado por un receptor IR ubicado a un costado del carrito. También se incorporó un módulo encoder para controlar la distancia aproximada de su recorrido y evitar que exceda la longitud del carril.

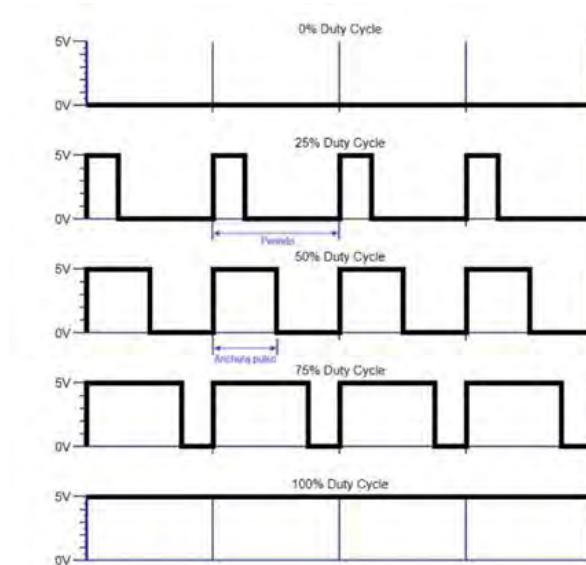
PWM, por sus iniciales en inglés que corresponden a Modulación por Ancho de Pulso, es una técnica esencial utilizada para la transmisión eficiente de señales, ya sean analógicas o digitales. La modulación resulta fundamental para garantizar que la señal pueda transmitirse sin pérdida de potencia o distorsión causada por interferencias.

Esta estrategia, conocida como PWM, se emplea específicamente en la transmisión de señales analógicas, convirtiéndolas en señales digitales para su posterior envío. En este método, se altera el ciclo de trabajo de una señal periódica, como una onda senoidal o cuadrada, con el propósito de transmitir información a través de un canal de comunicación o para regular la cantidad de energía dirigida a una carga. El ciclo de trabajo, expresado en porcentaje, representa el ancho de la parte positiva de la señal en relación con su período. De esta manera, un ciclo de trabajo (Duty Cycle) del 10% indicaría que la señal permanece en nivel alto durante el 10% del tiempo total del período, que se mide en relación con el 100%.

En esencia, la técnica implica activar una salida digital durante un intervalo específico y mantenerla desactivada durante el resto del tiempo, creando pulsos positivos que se repiten de forma continua. En este proceso, la frecuencia permanece constante, determinada por el tiempo entre la emisión de pulsos, mientras que se ajusta la amplitud del pulso, conocida como ciclo de trabajo. El valor promedio de la tensión de salida a lo largo del tiempo resulta equivalente al valor analógico deseado.

Figura 36.

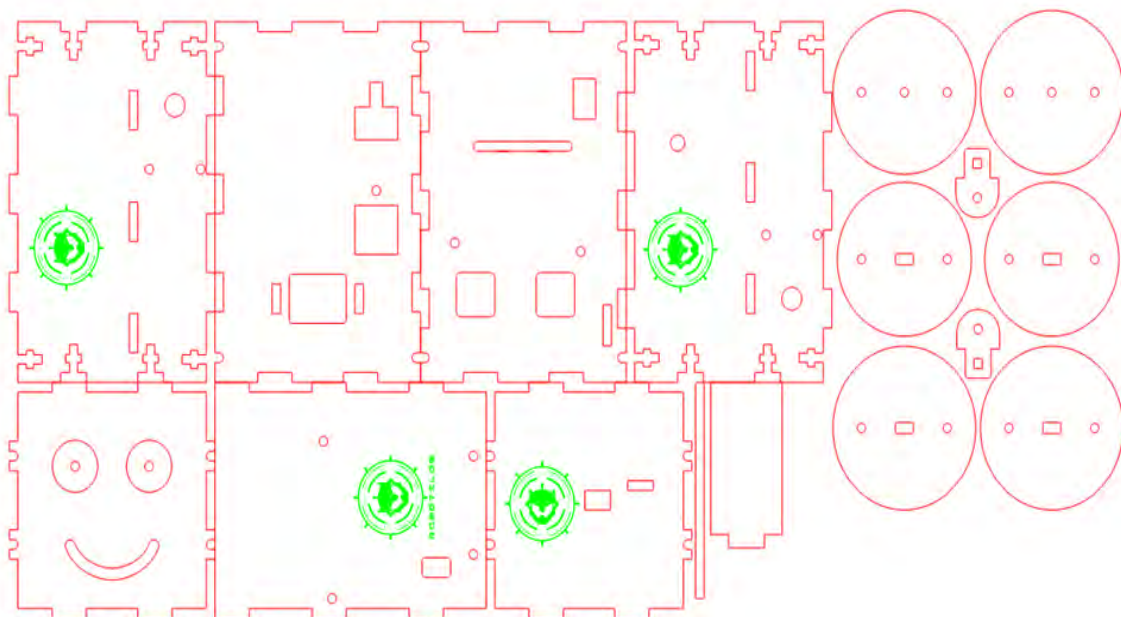
Ciclos de Trabajo Generados por el PWM



Para la construcción del carrito se utilizó MDF⁷ (Medium Density Fibreboard), la figura 37 corresponde al diseño en CAD de la carcasa del carrito y la figura 38 el carrito ensamblado.

Figura 37.

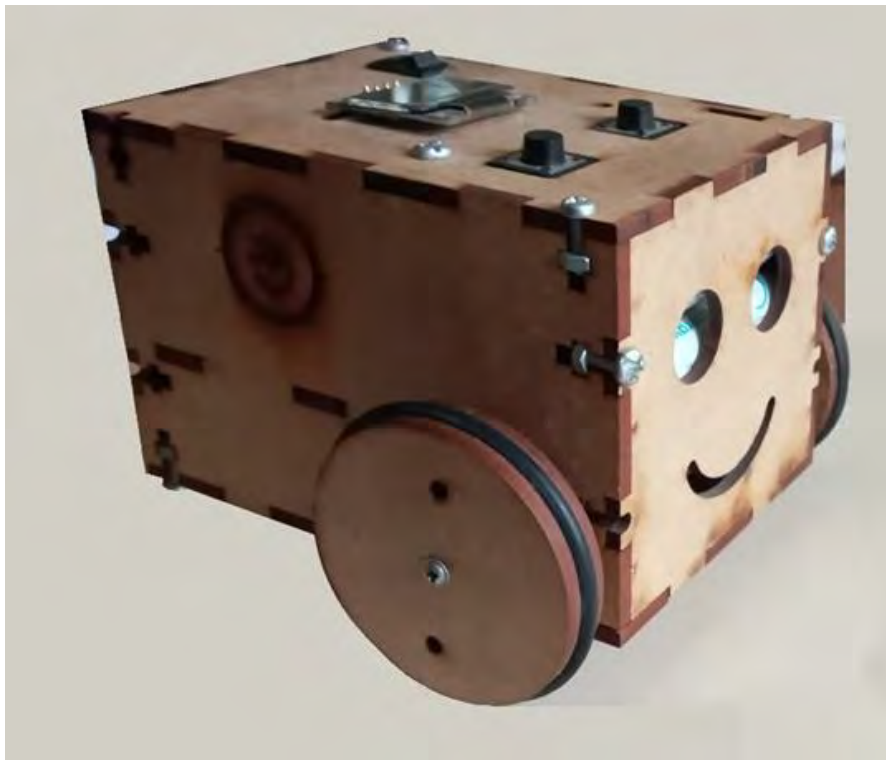
Diseño CAD de la Carcasa del Carrito para el Experimento de MRU



⁷ Tablero de fibras de madera y resinas sintéticas para proporcionar más densidad que la madera.

Figura 38.

Carrito ensamblado para el experimento de MRU

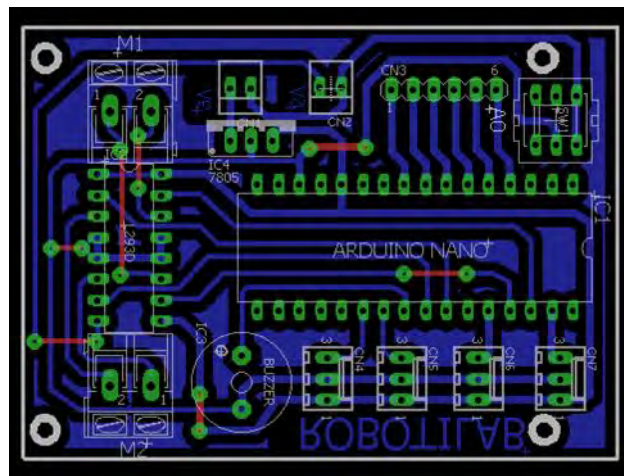


Para el control del carrito se diseñó una shield⁸ para el Arduino Nano. Esta permite conectar un motorreductor a través de un driver de motor, conectar el circuito de teclado y pantalla compuesto por dos pulsadores y una pantalla OLED, conectar el sensor IR para iniciar el movimiento del móvil tras un conteo descendente, y conectar un módulo encoder para controlar la distancia que recorrerá el móvil. Además, se utilizaron dos fuentes de voltaje: una para la electrónica del sistema y otra para el motor. También se incorporó un regulador de voltaje de 5V para la electrónica de control. En la figura 39 se muestra el diseño del circuito impreso de la shield para el Arduino Nano y en la figura 40 el circuito impreso para el teclado y pantalla.

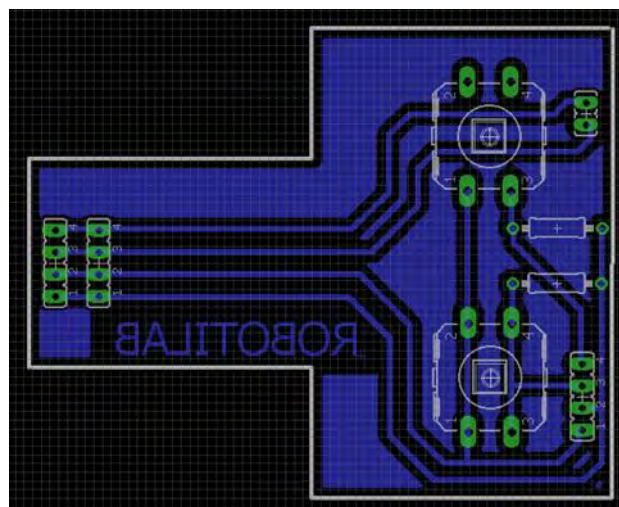
⁸ Placa de circuito que acopla a la placa Arduino conectándose a ella solo mediante sus pines sin necesidad de alguna otra conexión externa.

Figura 39.

Diseño del PCB con Arduino Nano para el Control del Carrito

**Figura 40.**

Diseño del Circuito Impreso del Teclado y Pantalla



4.1.4. Interface.

A través de la aplicación diseñada para el dispositivo Android⁹, el usuario tiene la tarea de supervisar los experimentos, lo cual implica establecer una comunicación entre el dispositivo Android, los sensores y los actuadores. Esta comunicación se efectúa mediante un sistema de control que adapta y transforma las señales generadas por los

⁹ Celular inteligente, tableta, ordenador portátil, etc. con Sistema Operativo Android.

sensores para que coincidan con las especificaciones de los puertos de entrada del dispositivo Android. De igual manera, convierte y acondiciona las señales de los puertos de salida para que sean compatibles con las características de los actuadores.

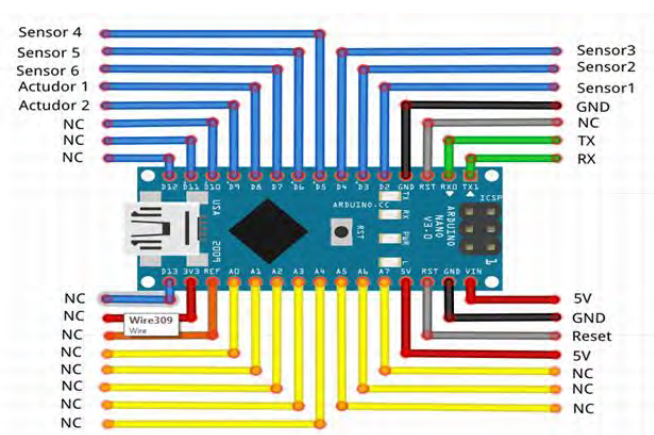
En los prototipos creados, la "Interfaz" es el componente responsable de unificar todos los sistemas de conversión y adaptación en un solo elemento, funcionando como el núcleo del sistema de control. En el presente proyecto se desarrollaron dos interfaces, una para los experimentos de mecánica y otra para los experimentos de electricidad. Las "Interfaces" fueron creadas mediante la utilización de placas Arduino¹⁰, las cuales se basan en un microcontrolador programable¹¹ ATmega328, junto con un entorno de desarrollo¹² (IDE).

Interface para los experimentos de mecánica.

Para el caso de la interface para experimentos de mecánica (MRU, Caída Libre y Péndulo Simple) se utilizó el Arduino nano. La figura 41 ilustra cómo se han dispuesto los puertos para los sensores, actuadores y alimentación, así como la configuración de la placa Arduino Nano, todo ello para la elaboración de la "Interface para mecánica".

Figura 41.

Configuración de los Puertos de la Placa Arduino



¹⁰ Plataforma de hardware libre basada en una placa con un microcontrolador y un entorno de desarrollo.

¹¹ Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

¹² Programa informático compuesto por un conjunto de herramientas de programación.

La shield diseñada para este Arduino permite conectar 6 sensores digitales y dos actuadores. Se incorporan optoacopladores con sus correspondientes transistores de acoplamiento para excitar a los electroimanes con la corriente necesaria para su funcionamiento. Tanto el Arduino como parte de la shield están alimentados por un adaptador externo de DC de 12 V a través de un conector USB tipo A. El Arduino y la shield trabajan con niveles de tensión TTL, para lo cual se ha incorporado un regulador de 5 voltios que proporciona suficiente corriente al sistema. Además, se ha añadido un zócalo para la inserción del módulo Bluetooth para la conectividad inalámbrica, como se ilustra en las figuras 42 y 43.

Figura 42.

Diseño Esquemático de la Interface para Experimento de Mecánica

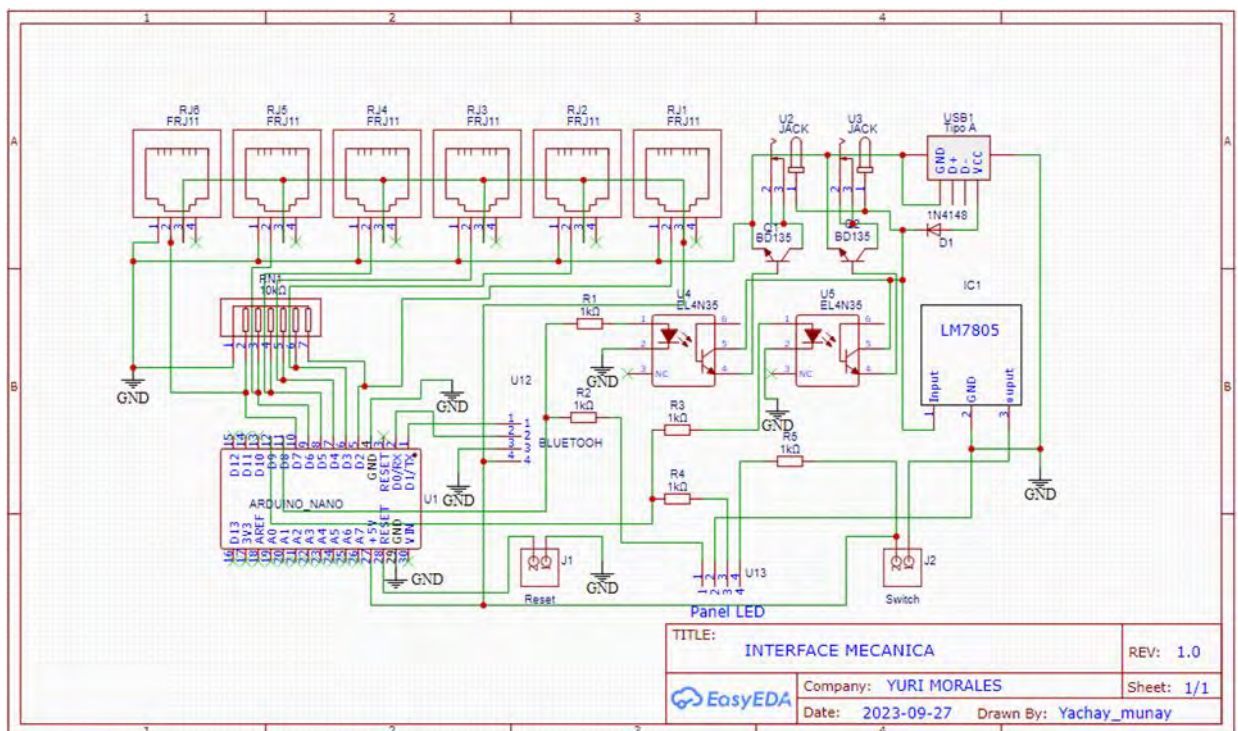
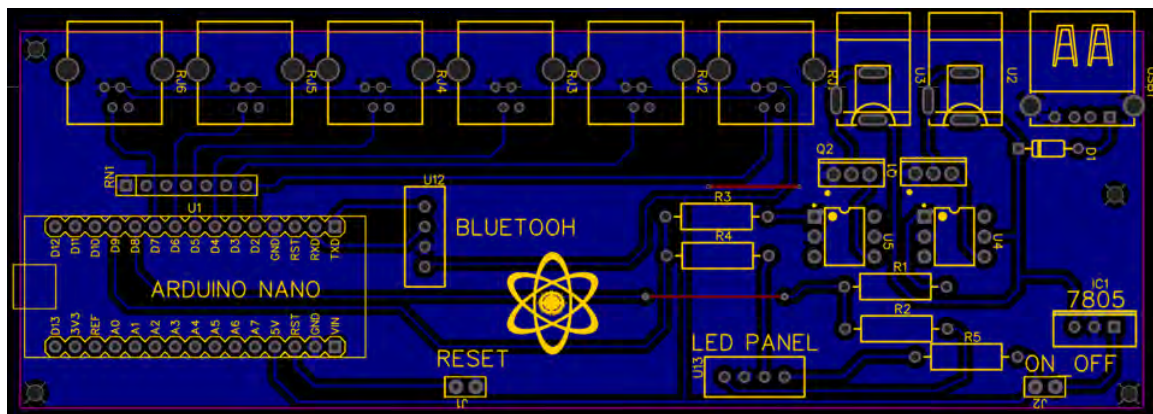


Figura 43.

Diseño PCB de la Interface para Experimento de Mecánica



En el panel frontal se han dispuesto tres indicadores ópticos: dos LED rojos que indican la actividad de los actuadores, y otro verde o azul para indicar la actividad de la interfaz, además del interruptor ON/OFF de la interfaz y un pulsador para reinicio, como se aprecia en la figura 44.

Figura 44.

Panel Frontal del Gabinete de la Interface



En la parte posterior del gabinete se encuentran seis conectores Jack RJ11 para los sensores, dos conectores Jack para los actuadores y un conector USB para el adaptador de 12 V. La carcasa se adaptó para albergar adecuadamente la tarjeta PCB de la interfaz,

permitiendo fácil acceso a los conectores y a los componentes interiores para reparaciones en caso de deterioro. La figura 45 muestra la parte posterior de la carcasa y los conectores.

Figura 45.

Vista Posterior de la Interface

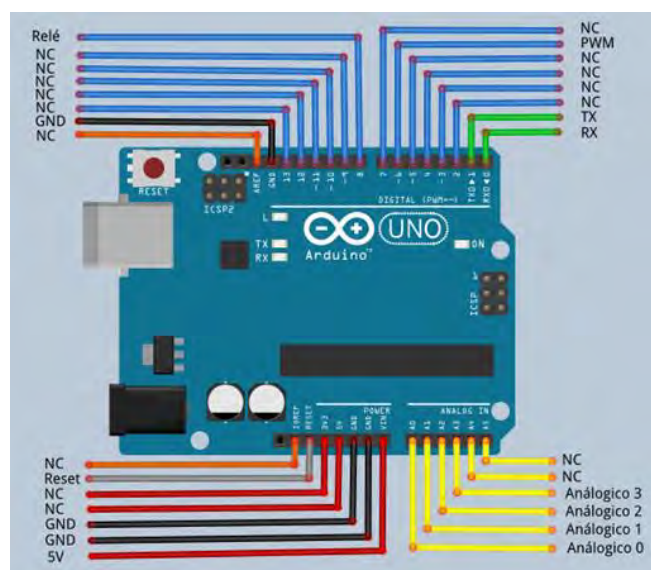


Interface para los experimentos de electricidad.

En el caso de la interfaz para experimentos de Electricidad, se utilizó Arduino UNO. La figura 46 ilustra cómo se han dispuesto los puertos para los sensores, actuadores y alimentación, así como la configuración de la placa Arduino UNO, todo ello para la elaboración de la interfaz.

Figura 46.

Configuración de los Puertos de la Placa Arduino UNO



La shield diseñada para este Arduino permite controlar un relé de dos circuitos, conectar cuatro entradas analógicas para medir potencial y corriente eléctrica, y proporcionar una salida digital para el experimento de tensión-corriente con una fuente de voltaje variable entre 0 y 5 voltios utilizando el pin 6, el cual puede configurarse como PWM. La shield y el Arduino están alimentados por un adaptador externo de DC de 12V a través de un conector Jack. El Arduino y parte de la shield operan con niveles de tensión TTL para los cuales se ha incorporado un regulador de 5 voltios que proporciona suficiente corriente al sistema. Para la conectividad inalámbrica se ha incorporado un zócalo para la inserción del módulo Bluetooth, como se ilustra en las figuras 47 y 48.

Figura 47.

Diseño Esquemático de la Interface para Experimentos de Electricidad

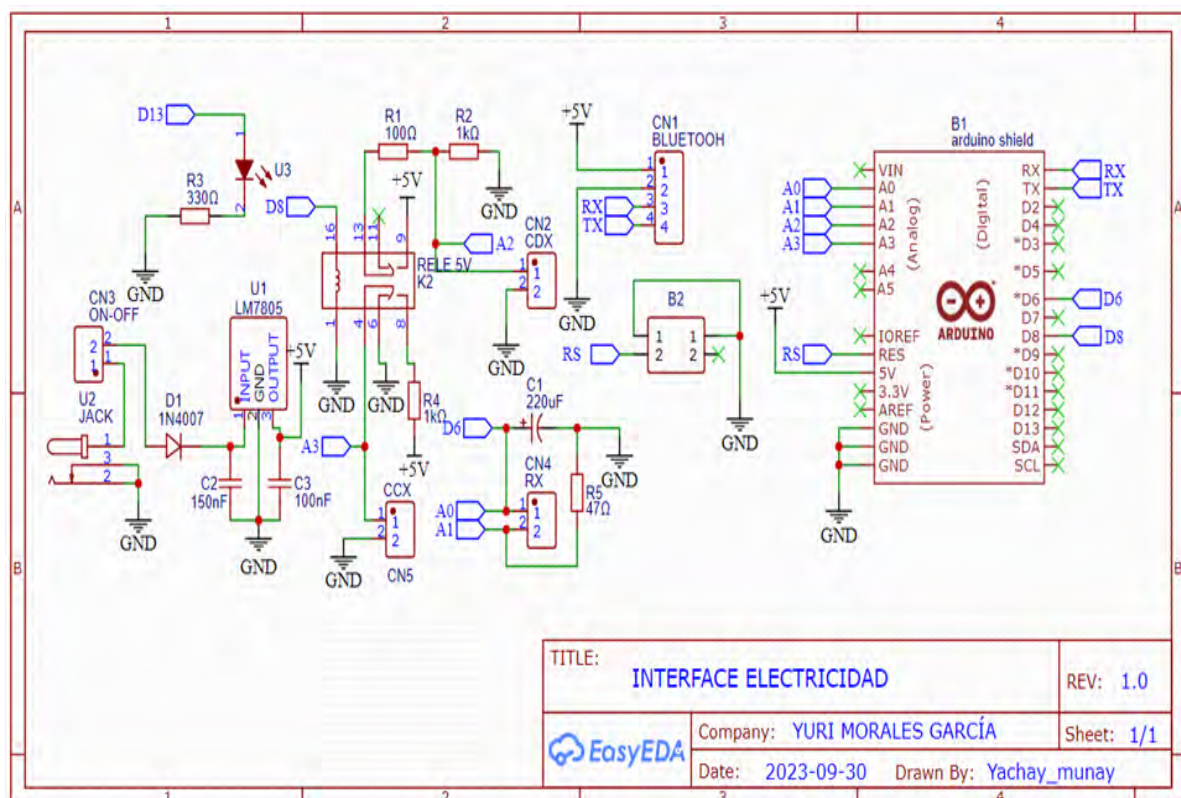
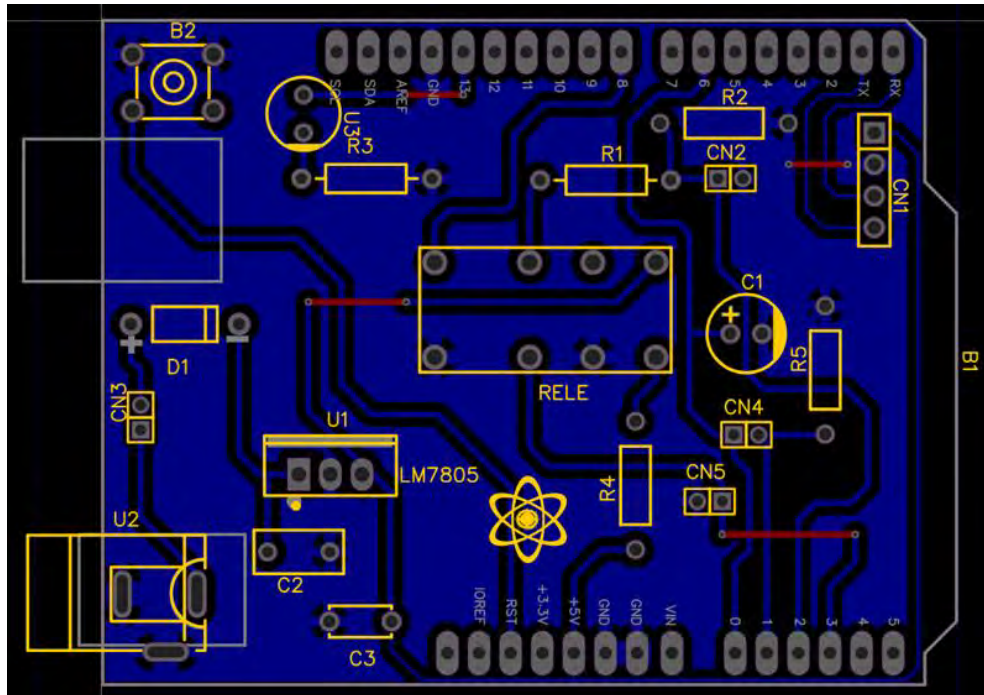


Figura 48.

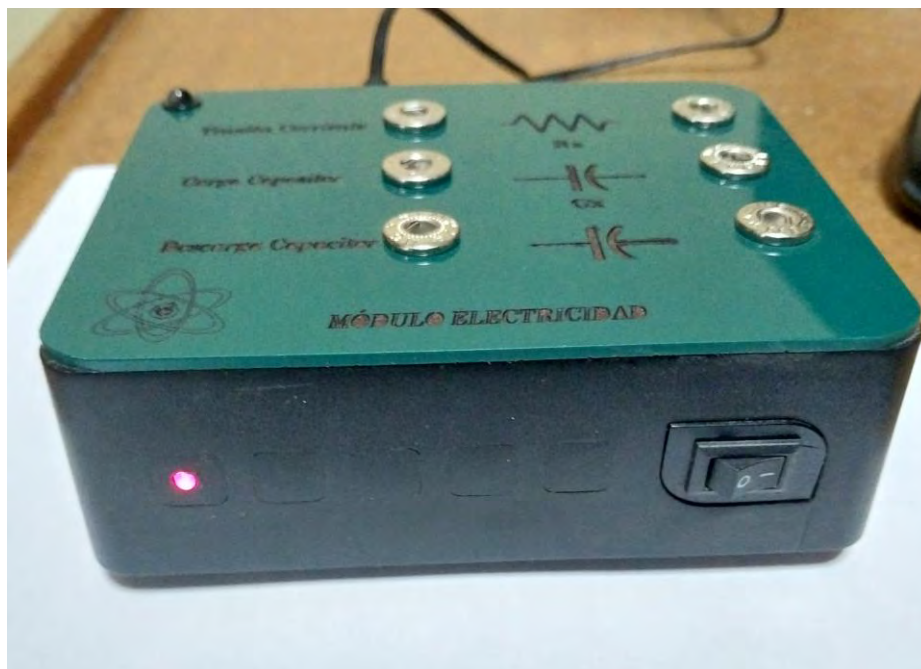
Diseño PCB de la Interface para Experimentos de Electricidad



En el panel frontal se ha dispuesto un LED rojo que indica la actividad de la interface además del interruptor ON / OFF como se muestra en la figura 49.

Figura 49.

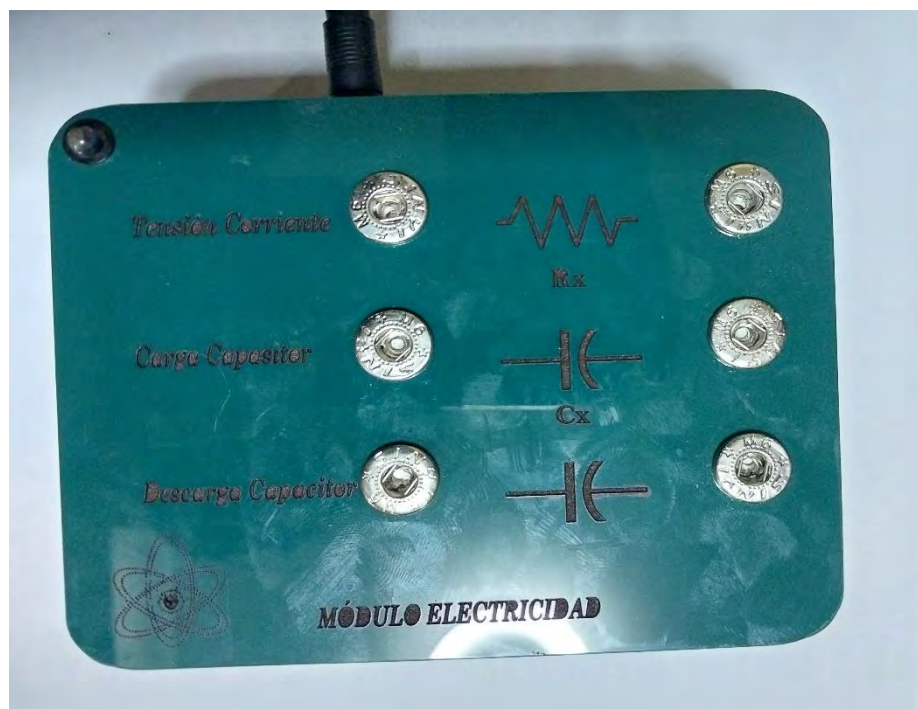
Panel Frontal de la Interface para Experimentos de Electricidad



En la parte superior del gabinete se han colocado terminales metálicos (Broches) para insertar los componentes de prueba (resistores o capacitores) con sus símbolos eléctricos correspondientes, gravados en bajo relieve para una mejor orientación, como se muestra en la figura 50.

Figura 50.

Parte Superior de la Interface para Conectar Resistores o Capacitores



4.1.5. Otros componentes

Soporte universal

Un soporte universal, también conocido como *Soporte de laboratorio*, es parte del equipamiento de un laboratorio que se utiliza para realizar diferentes montajes con los materiales presentes en un laboratorio para preparar una serie de experimentos. Básicamente está conformado por dos elementos, una base que puede ser triangular o rectangular y una varilla cilíndrica que permita sujetar materiales como pinzas, nueces,

doble nueces, etc. La figura 51 muestra el soporte universal de base rectangular adquirido para realizar los experimentos presentados con los módulos.

Figura 51.

Soporte Universal

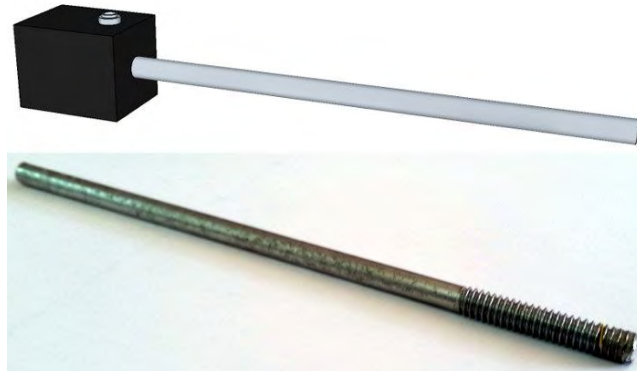


Varilla de acople.

La varilla de acople que se muestra en la figura 52 es un cuerpo cilíndrico metálico que forma parte del equipamiento para el montaje de los experimentos propuestos en el presente trabajo y permite sujetar diferentes elementos, como las fotopuertas y los actuadores. La varilla de acople es un cuerpo cilíndrico de un cuarto de pulgada de diámetro y aproximadamente quince centímetros de longitud. En uno de los extremos, tiene una rosca con una tuerca para sujetar los diferentes componentes.

Figura 52.

Varilla de Acople, Diseño CAD y varilla Construida



Abrazaderas

Las abrazaderas se consideran herramientas de equipamiento de laboratorio, generalmente construidas de metal. Sirven para sujetar diferentes objetos o realizar montajes elaborados. Para los módulos presentados se diseñó y construyó abrazaderas de ángulo recto (right angle clamp), como se muestra en la figura 53, que permite sujetar la varilla del soporte universal y las varillas que se acoplan a un actuador o un sensor en ángulo recto. El cuerpo de la abrazadera es una barra metálica cuadrada con dos agujeros circulares que forman un ángulo recto, uno del diámetro del cilindro del soporte y el otro del diámetro de la varilla de acople. En las caras laterales de la abrazadera se incluyen dos pernos de sujeción con tuercas tipo mariposa para mantener fijos el cilindro y la varilla.

Figura 53.

Abrazadera de Ángulo Recto



4.2. DESARROLLO DEL SOFTWARE.

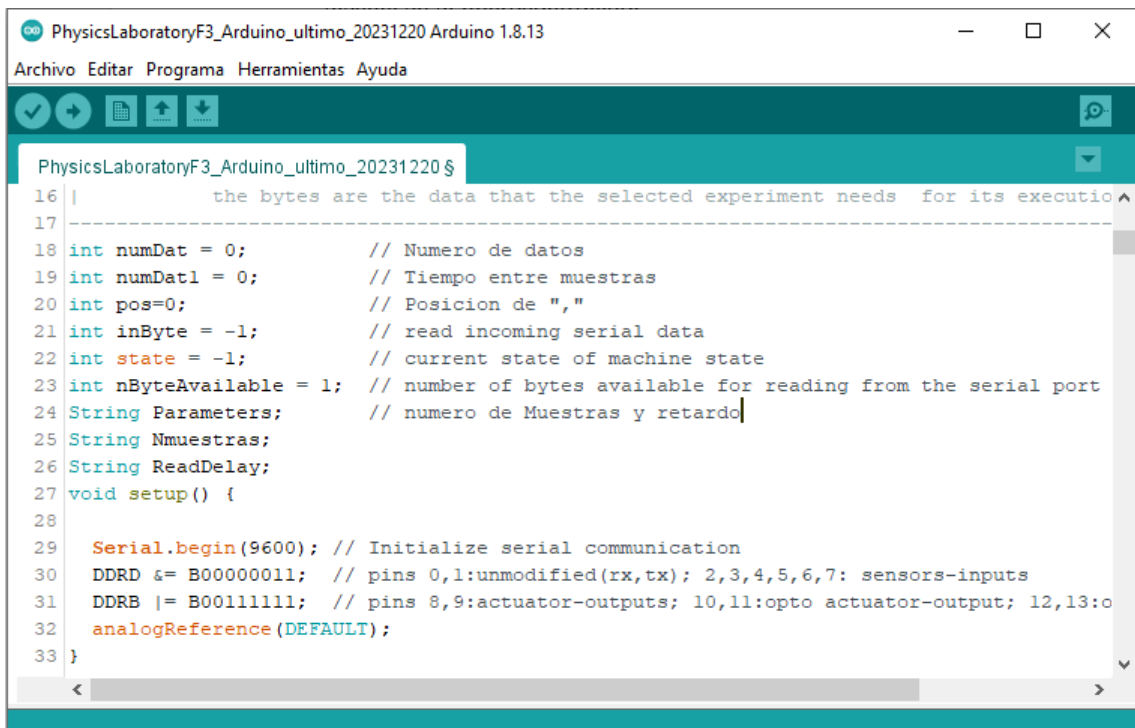
4.2.1. Programación de la placa Arduino.

La interfaz fue desarrollada utilizando las placas Arduino Uno R3 y Arduino Nano de la plataforma Arduino. Estas placas cuentan con catorce pines de entrada/salida, los cuales pueden programarse mediante el entorno de desarrollo integrado (IDE). Este entorno permite editar y compilar el código, que posteriormente se carga y ejecuta en el microcontrolador.

El software diseñado para las placas Arduino se compone de dos funciones principales, ambas de tipo *void* (sin retorno de valor): *setup()* y *loop()*. Estas funciones se describen a continuación, junto con otras funciones personalizadas que permiten la creación de fragmentos modulares de código reutilizables, lo que contribuye a una estructura más organizada y fácil de comprender. La figura 54 ilustra una vista del entorno de desarrollo integrado (IDE) con el código desarrollado para la placa Arduino.

Figura 54.

IDE Arduino con un Fragmento del Código de la Aplicación



```

PhysicsLaboratoryF3_Arduino_ultimo_20231220 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda
PhysicsLaboratoryF3_Arduino_ultimo_20231220 $
16 |         the bytes are the data that the selected experiment needs  for its executio
17 | -----
18 int numDat = 0;           // Numero de datos
19 int numDat1 = 0;         // Tiempo entre muestras
20 int pos=0;               // Posicion de ", "
21 int inByte = -1;        // read incoming serial data
22 int state = -1;         // current state of machine state
23 int nByteAvailable = 1; // number of bytes available for reading from the serial port
24 String Parameters;      // numero de Muestras y retardo
25 String Nmuestras;
26 String ReadDelay;
27 void setup() {
28
29     Serial.begin(9600); // Initialize serial communication
30     DDRD &= B00000011; // pins 0,1:unmodified(rx,tx); 2,3,4,5,6,7: sensors-inputs
31     DDRB |= B00111111; // pins 8,9:actuator-outputs; 10,11:opto actuator-output; 12,13:o
32     analogReference(DEFAULT);
33 }

```

La configuración inicial de la placa Arduino se realiza en la función *setup()*, que se ejecuta al inicio y solo una vez, inicializa la comunicación inalámbrica de forma serial y configura los puertos de entrada/salida, que para el caso de nuestras interfaces serían: dos pines para la comunicación inalámbrica por Bluetooth: pin 0 (*RX*), pin 1 (*TX*), que es la configuración por defecto de la placa.

Para el caso de la interfaz para experimentos de mecánica, se configuraron los pines digitales: seis pines como entradas para los sensores que son los pines 2, 3, 4, 5, 6, 7 y dos pines como salida para los actuadores, de los cuales los pines 8 y 9 son salidas con circuitos optoacopladores para protección, como se muestra en el siguiente código.

```
void setup() {
    Serial.begin(9600); // Initialize serial communication
    DDRD &= B00000011; // pins 0,1:unmodified(rx,tx)
    DDRB |= B00001111; // pins 8,9:actuator-outputs
}
```

En el caso de la interfaz para experimentos de electricidad se configuraron los pines digitales de forma similar para el caso de la interfaz mecánica, además de configurar todos los pines analógicos por defecto: seis pines, como se muestra en el siguiente código.

```
void setup() {
    Serial.begin(9600); // Initialize serial communication
    DDRD &= B00000011; // pins 0,1:unmodified(rx,tx)
    DDRB |= B00111111; // pins 8,9:actuator-outputs|
    analogReference(DEFAULT);
}
```

La función *loop()* contiene el código que se ejecutará continuamente y es aquí donde se establecen las acciones que realizará la placa Arduino mientras esté conectada a una fuente de voltaje. En la primera parte del código, Arduino está revisando continuamente si hay datos presentes para su lectura en el puerto serial. Una vez que se encuentre algún dato, el programa lee el dato e inicia una máquina de estados. El primer estado case 0: verifica la conexión entre la placa Arduino y el dispositivo Android, para

esto, cuando el dispositivo Android se conecta a la placa por Bluetooth, envía el código “00” a la placa y esta le devuelve el código “1”, lo que indica que la conexión fue correcta.

```

35 void loop() {
36
37   if ( Serial.available() > nByteAvailable ) {
38
39     // Whatever is available from the serial is read here
40     inByte = Serial.read();
41
42     //////////////////////////////////// BEGINING STATE MACHINE. ////////////////////////////////////
43     switch (state) {
44       case -1: // NOTHING RECEIVED YET
45         if ( inByte >= 48 && inByte <= 50 ) {
46           state = inByte - 48;
47         }
48         nByteAvailable = nByteAvailable - 1;
49         break; // state = -1 (initial state) taken care off
50
51       case 0: // VERIFY THE CONNECTION BETWEEN APPLICATION AND ARDUINO.
52         Serial.println('1');// The Application should read the value the '1'
53         state = -1;
54         nByteAvailable = 1;
55         break; // state = 0 taken care of

```

Para el segundo estado *case 1*: cuando la aplicación en el dispositivo Android abre la pantalla de resultados, dependiendo del experimento seleccionado por el usuario, el dispositivo Android envía un código a la placa para reconfigurar algunas entradas, salidas u otro parámetro.

```

57     case 1: // CONFIGURATE INPUT, OUTPUT
58       if (inByte == 48) { // Application send '10'
59         digitalWrite(13, HIGH);
60         delay(1000);
61         digitalWrite(13, LOW);
62       }
63       else if (inByte == 49) { // Application send '11'
64         digitalWrite(13, HIGH);
65         delay(2000);
66         digitalWrite(13, LOW);
67       }
68       else if (inByte == 50) { // Application send '12'
69         digitalWrite(13, HIGH);
70         delay(3000);
71         digitalWrite(13, LOW);
72       }

```

En el tercer estado *case 2*: se realiza el experimento y se ejecuta cuando el usuario, en el dispositivo Android, presiona el botón de *INICIAR* de la aplicación.

```

95     case 2:                // PERFORM THE EXPERIMENT
96         if (inByte == 48) { // Application send '20'
97             //
98             delay(100);
99             inByte = Serial.read();
100            if(inByte != -1){
101                TimeFreefall();
102            //          Serial.println(inByte);
103            }
104        }
105        else if (inByte == 49) { // Application send '21'
106            TimeSimplependulum();
107            Serial.println(21);
108        }
109        else if (inByte == 50) { // Application send '22'
110            MovimientoCircularUniforme()
111            Serial.println(22);
112        }

```

Para la interfaz mecánica, se han desarrollado tres funciones: ‘*FreeFallTime()*’, ‘*SimplePendulumTime()*’ y ‘*MRU_Time()*’, que ejecutan los experimentos de Caída libre, Péndulo simple y Movimiento Rectilíneo Uniforme, respectivamente. La función ‘*TimeFreeFall()*’ requiere que se ingrese como dato de entrada el número de las alturas desde donde se dejará caer una esfera sólida, pudiendo ser hasta veinte alturas, y retorna como datos de salida los tiempos en que el objeto impacta sobre la superficie donde se encuentra adherido el sensor. La función ‘*TimeSimplePendulum()*’ requiere que se ingrese como dato de entrada el número de oscilaciones y retorna como resultado los tiempos para cada oscilación. La función ‘*MRU_Time()*’ requiere que se ingrese como dato de entrada el número de sensores, que, por limitaciones de entradas de la placa solo se pueden colocar seis sensores como máximo, y retorna como datos de salida los tiempos en los que el móvil (carrito) pasa por cada uno de los sensores. El código para el experimento de Péndulo simple es:

```

196 void SimplePendulumTime(int numLong) { // Number of oscillations
197
198     int count = 0;
199     int numOsc = 10;
200     String txtResult;
201     bool bStatus = false;
202     unsigned int tempTime = 0;
203     //unsigned int waitTime = 10000;
204     unsigned int waitTime = 20000;
205     unsigned int elemagTime = 5000;
206     unsigned long startTime = 0;
207     unsigned long timeSensor1[numOsc];
208     unsigned long addTime = 0;
209
210     for (count = 0; count < numOsc; count++){ timeSensor1[count]=0; }
211
212     /* Activating the two electromagnet
213     (pin10 and pin 11)in millisecond */
214     PORTB |= B00001111;
215     delay(elemagTime);
216     PORTB &= B11110000;
217     waitTime += millis();
218
219     /* measure the time when the object passes
220     by each of the photogate sensors */
221     count = 0;
222     while ((count < 2*numOsc+1)&&(tempTime < waitTime)){
223         bStatus = false;
224         do{
225             tempTime = millis();
226             if((PIND & B11111100) != 0){
227                 if(count == 0){
228                     startTime = micros();
229                 }
230                 else if(count % 2 == 0){
231                     timeSensor1[int(0.5*count)-1] = micros() - startTime;
232                 }
233                 do{
234                     tempTime = millis();
235                     if((PIND & B11111100) == 0){ |
236                         bStatus = true;
237                     }

```

Para la interfaz de electricidad, se han desarrollado tres funciones: ‘*TensiónCorriente()*’, ‘*CargaCondensador()*’ y ‘*DescargaCondensador()*’, que ejecutan los experimentos de la Ley de Ohm, carga y descarga de un capacitor, respectivamente. La función ‘*TensiónCorriente()*’, no requiere ningún dato de entrada, pero retorna como datos de salida el potencial y la intensidad de corriente en la resistencia de prueba. Las

funciones ‘*CargaCondensador()*’ y ‘*DescargaCondensador()*’ requieren que se ingrese como dato de entrada el número de muestras, que como máximo solo pueden ser 21, limitado por el protocolo de comunicación para la transferencia de datos que utiliza Android, y el tiempo de muestreo (tiempo entre muestras), y retorna como datos de salida los tiempos y las potenciales eléctricos en el capacitor. El código para el experimento Carga del capacitor es:

```

212 void CargaCondensador(int numMuestras, int retardo) { // Number of samples
213     #define ledPin 13
214     #define relayPin 8
215     #define dischargePin A2
216     #define chargePin A3
217     int muestras = numMuestras;
218     int data[muestras];
219     unsigned long ref_time;
220     unsigned int times[muestras];
221     int datos;
222     const int delay_charge = 5000; // in milliseconds
223     int delay_read = retardo; // in milliseconds
224     int count;
225     int choice;
226     String cad;
227     String txtResult;
228     bool bStatus = false;
229     // the setup routine runs once when you press reset:
230     digitalWrite(relayPin, LOW);
231     delay(delay_charge);
232     digitalWrite(relayPin, HIGH);
233     digitalWrite(ledPin, HIGH);
234     ref_time = millis ();
235     for (count=0; count < muestras; count++)
236     {
237         data[count] = analogRead(3);
238         times[count] = millis() - ref_time;
239         delay(delay_read);
240     }
241     digitalWrite(ledPin, LOW);
242     if (count==muestras){bStatus = true;}
243
244     /* Results */
245     delay(100);
246     if(bStatus) // Data acquisition was correct
247         txtResult = 1 + "|";
248     else // Error in data acquisition
249         txtResult = 0 + "|";
250     digitalWrite(ledPin, HIGH);

```

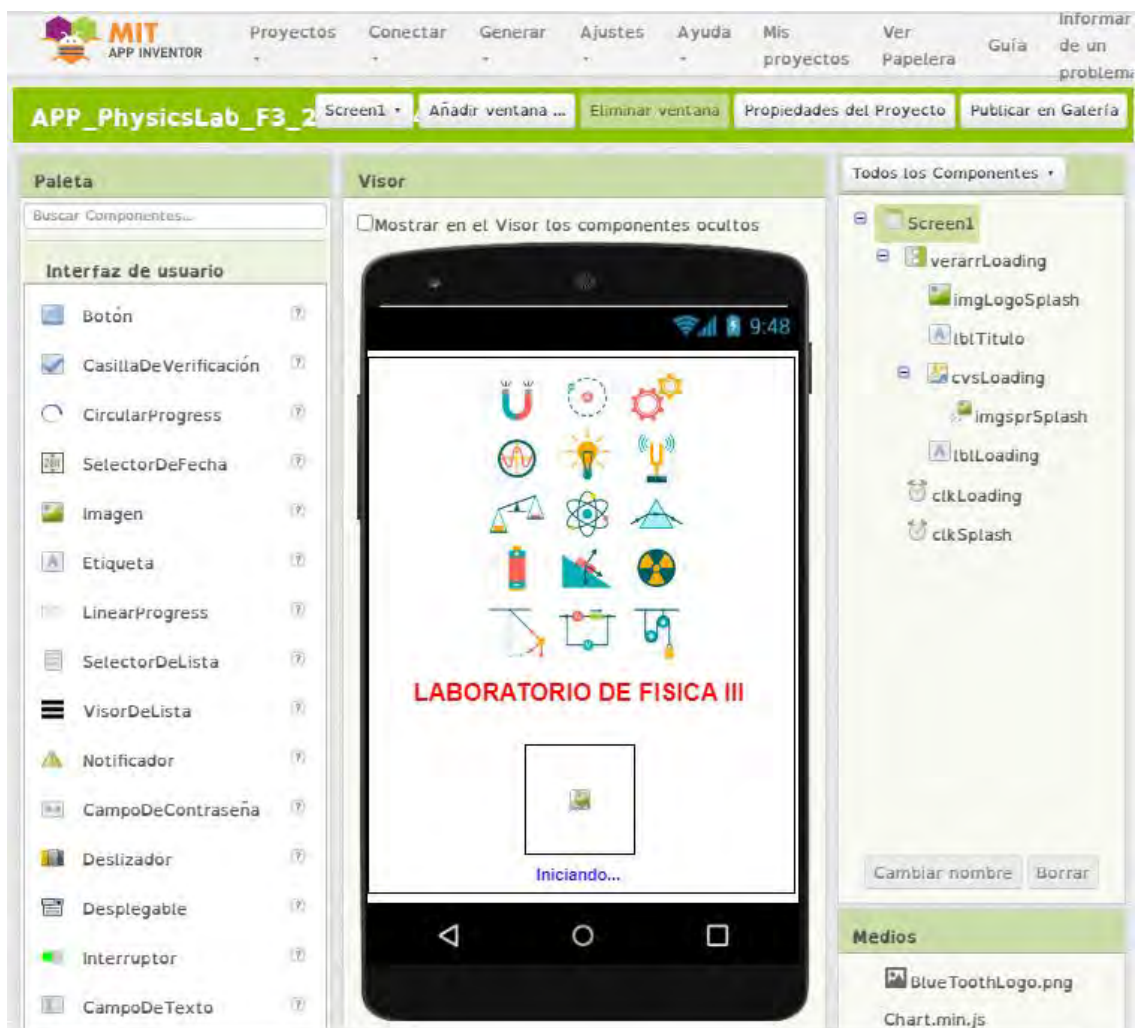
4.2.2. Programación en APP Inventor.

Pantalla de bienvenida.

Al iniciar el software desarrollado para las interfaces en un dispositivo Android, inicialmente la aplicación muestra una pantalla de bienvenida que consiste en un elemento gráfico de control, que incorpora una imagen con un ícono animado, señalando al usuario que el programa está comenzando. Además, presenta un texto con la versión del programa y un logotipo que identifica la aplicación. La figura 55 muestra el diseño de la pantalla de bienvenida en el App Inventor.

Figura 55.

Diseño de la Pantalla de Bienvenida en App Inventor



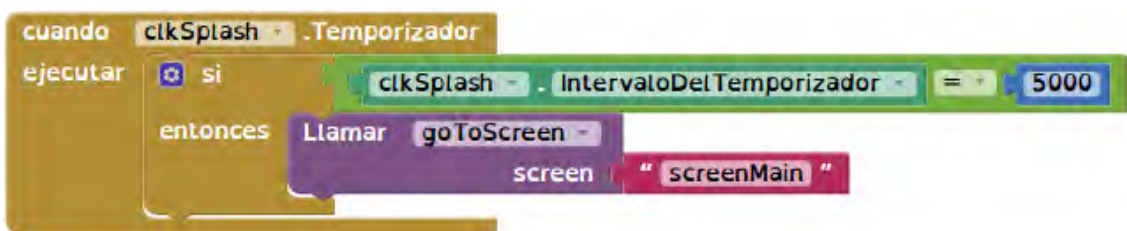
El código destinado a la pantalla inicial de la aplicación emplea la función *'Initialize()'*, la cual establece un color de fondo para la pantalla y activa dos componentes de reloj (*'Clock'*) en App Inventor. El primer reloj controla la animación del ícono de carga, mientras que el segundo reloj regula la duración de la pantalla de bienvenida.



La función *'clkLoading.Timer()'*, controla el componente reloj que genera la animación de la imagen *'imgLoading'*, haciendo una rotación de seis grados cada ocho milisegundos, tomando como eje el centro de la imagen, como muestra el siguiente código.



La función *'clkSplash()'*, cuyo código se presenta continuación, gestiona el componente de reloj que mantiene activa la pantalla de bienvenida en el dispositivo Android. Esta función se ejecuta cuando han transcurrido cinco segundos, que es el tiempo establecido para el temporizador del reloj. Una vez que se cumple el tiempo programado, se abre la pantalla principal de la aplicación.



Pantalla de la aplicación principal.

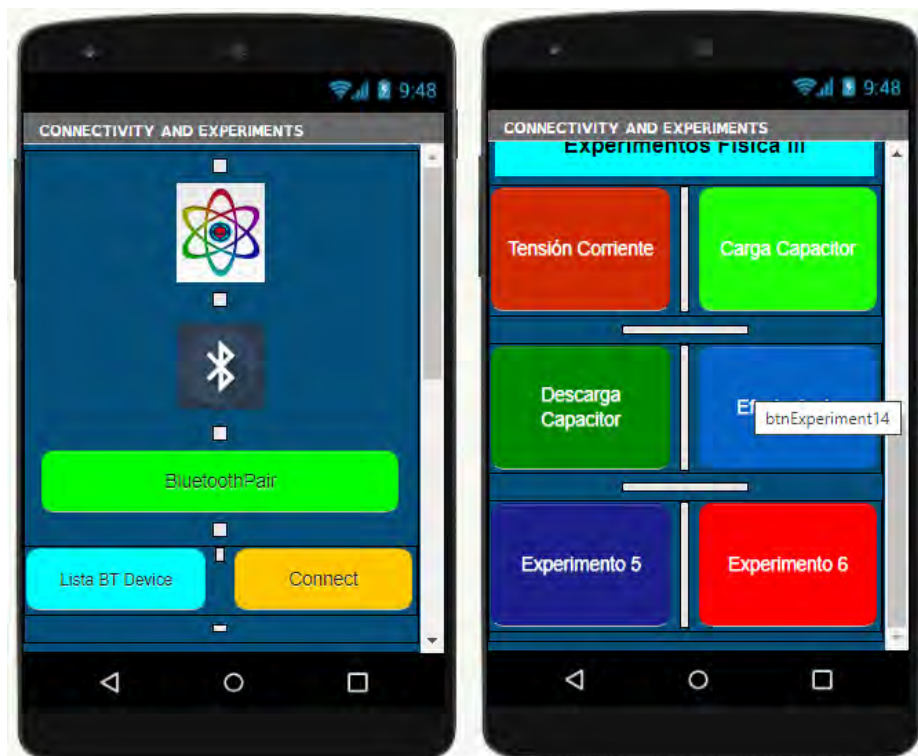
Una vez que se cierra la pantalla de bienvenida, la aplicación principal se inicia automáticamente y realiza dos acciones. En primer lugar, establece la conexión con la interfaz mediante comunicación inalámbrica a través de Bluetooth. Al mismo tiempo, muestra un menú que permite al usuario seleccionar el experimento que desea llevar a cabo.

Esta pantalla incluye un botón llamado "*BLUETOOTH PAIR*", que facilita el proceso de emparejamiento o vinculación entre el dispositivo Android y la interfaz. Durante este proceso, ambos dispositivos se identifican mutuamente mediante un nombre y una dirección interna, y se solicita una clave *PIN* para completar la conexión. Dado que puede haber múltiples interfaces disponibles, cada una con un módulo Bluetooth distinto identificado por una dirección única, la aplicación solo puede conectarse a una de ellas. Al seleccionar el botón "*LIST BT DEVICE*", se muestra una lista de los dispositivos Bluetooth vinculados al dispositivo Android, de entre los cuales el usuario debe elegir uno. Luego, al presionar el botón "*CONNECT*", se establece la conexión con el dispositivo Bluetooth previamente seleccionado, mientras que el botón "*EXIT APPLICATION*" finaliza la conexión con la interfaz y cierra la aplicación. Una vez que la conexión con la interfaz se realiza correctamente, el usuario puede seleccionar uno de los experimentos mostrados en la pantalla presionando el botón correspondiente al experimento deseado. La figura 56 muestra el diseño de la pantalla de la aplicación principal.

Tras el cierre de la pantalla de bienvenida, la pantalla de la aplicación principal se abre automáticamente.

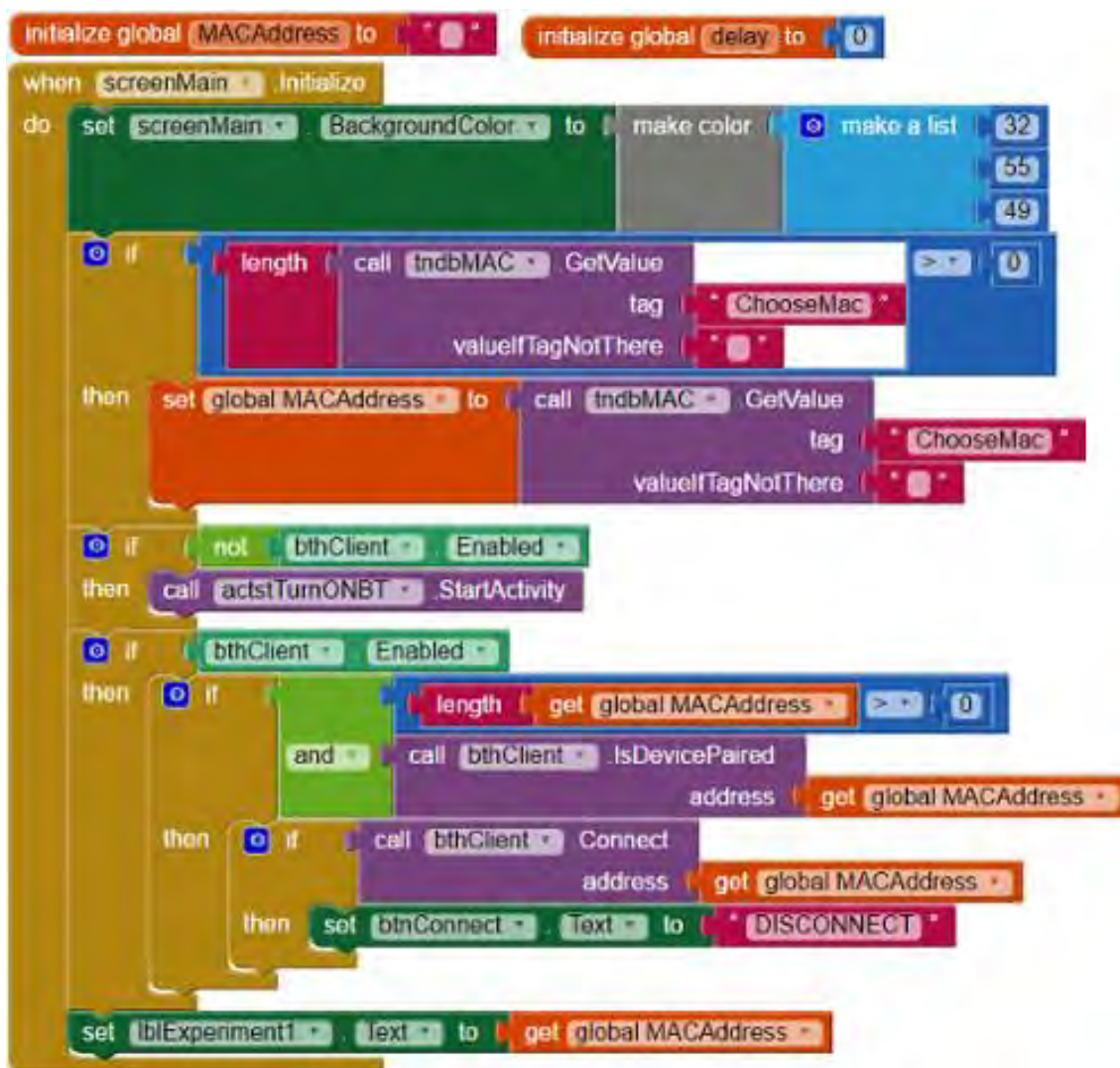
Figura 56.

Diseño de la Pantalla para la Aplicación Principal



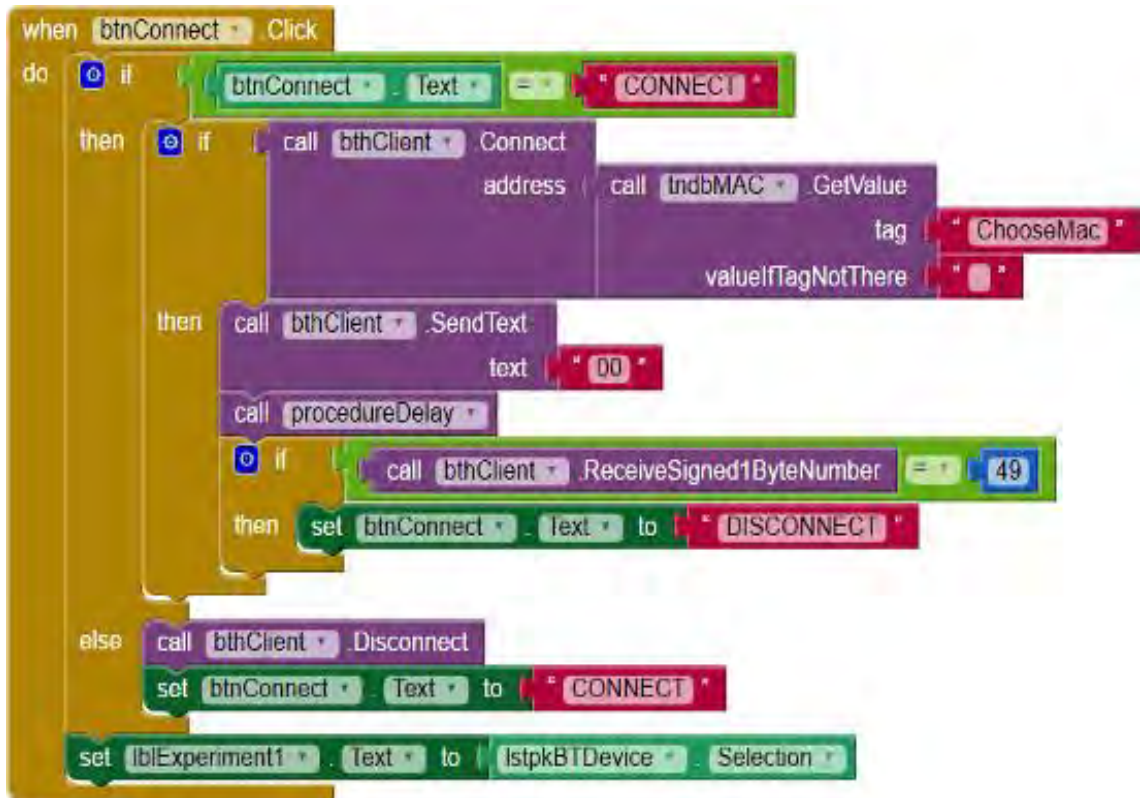
Se ha creado una base de datos que almacena la dirección de la interfaz que fue utilizada más recientemente por la aplicación en el dispositivo Android. Este registro permite que, en futuros usos con la misma interfaz, la conexión se configure automáticamente, evitando la necesidad de repetir el proceso de emparejamiento y conexión.

Antes de que aparezca la pantalla principal de la aplicación, el programa ejecuta la función `'screenMain.Initialize()'`. Esta función lleva a cabo varias acciones, que incluyen establecer un color de fondo en la pantalla, verificar si hay alguna dirección de interfaz almacenada en la base de datos, y comprobar si el Bluetooth del dispositivo Android está activado; de lo contrario, inicia el proceso de activación automáticamente. Finalmente, si hay una dirección de interfaz almacenada en la base de datos, procede a realizar la conexión de forma automática.

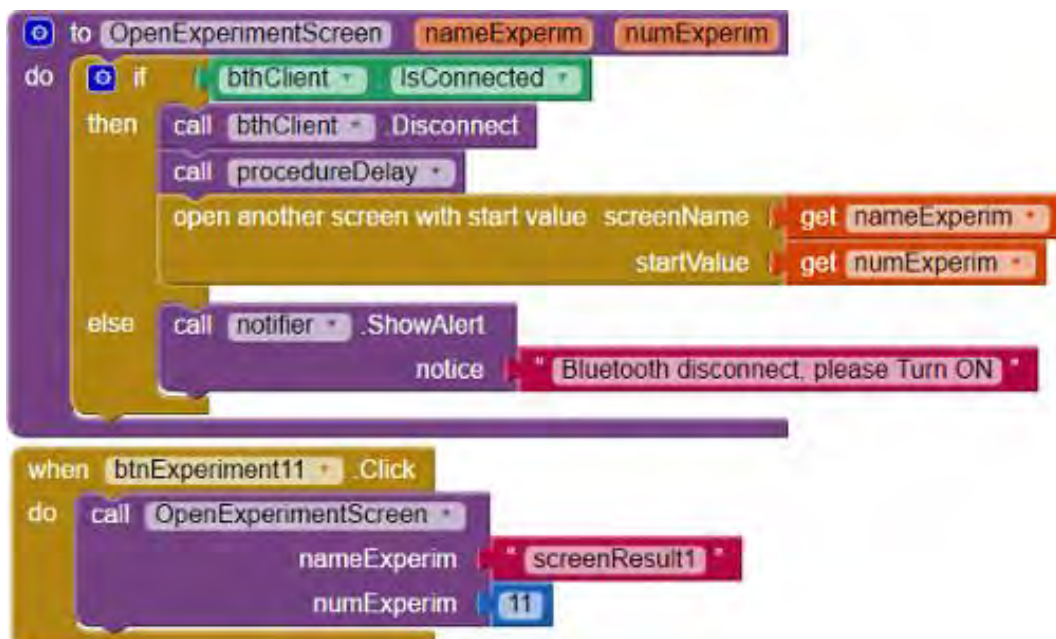


Cuando se ejecuta la aplicación por primera vez en un dispositivo Android o cuando se desea controlar una interfaz diferente a la última utilizada, es necesario llevar a cabo el proceso de emparejamiento y conexión con la interfaz utilizando los botones *BLUETOOTH PAIR*, *LIST BT DEVICE* y *CONNECT*. El botón *BLUETOOTH PAIR* facilita el emparejamiento o vinculación con la interfaz. El botón *LIST BT DEVICE* muestra una lista de todos los módulos Bluetooth (interfaces) vinculados a nuestro dispositivo dentro del alcance de éste. Por último, el botón *CONNECT* establece la conexión entre nuestro dispositivo Android y la interfaz. Para asegurarse de que la conexión sea exitosa, se envía un dato de prueba a la interfaz y, si la conexión es correcta, se recibe un dato de confirmación de la interfaz. A continuación, se muestra el código de

de la función que controla el botón *CONNECT*.



Después de que se haya establecido correctamente la conexión entre el dispositivo Android y la interfaz, se activa el menú que exhibe todos los experimentos disponibles. A partir de este punto, el usuario tiene la opción de seleccionar cualquier experimento, lo que generará una nueva pantalla para ingresar los datos correspondientes.



La figura 57 presenta una perspectiva de la pantalla inicial de la aplicación desde un teléfono inteligente equipado con el sistema operativo Android.

Figura 57.

Vista de la Pantalla Principal desde un Celular con Android



Pantalla de resultados.

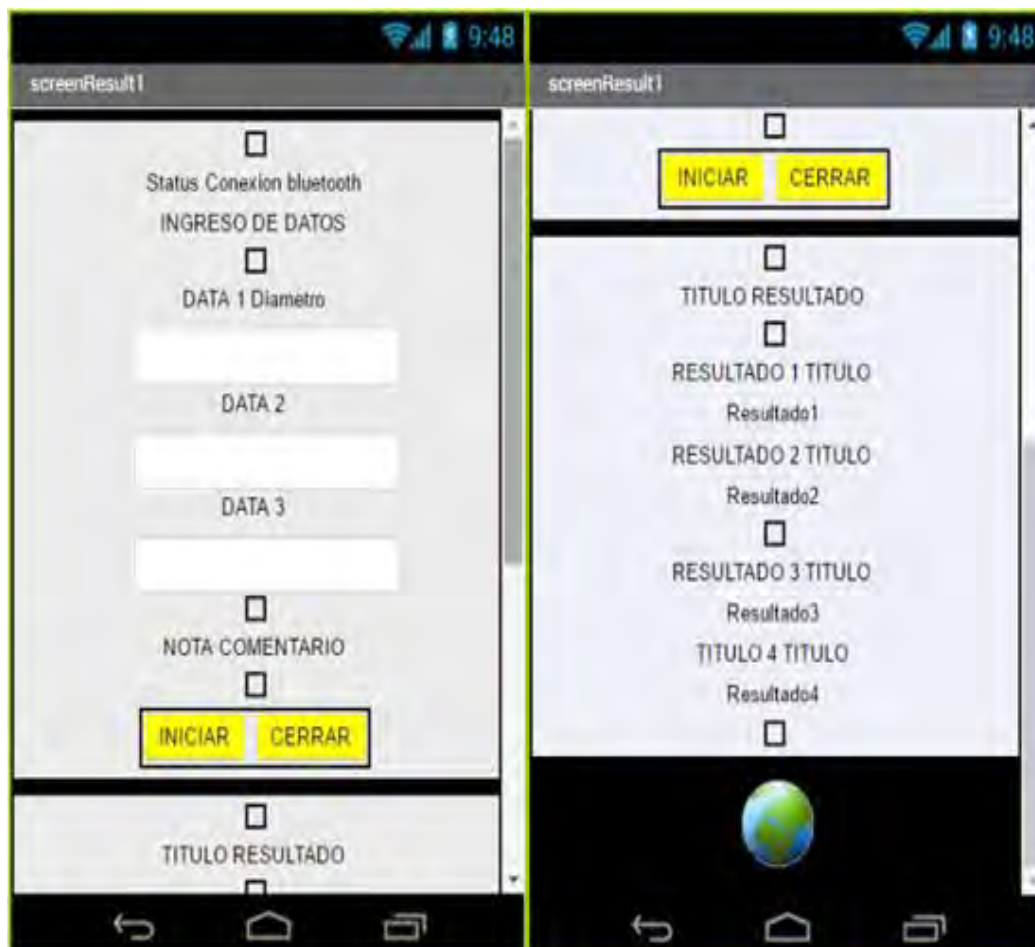
Después de que el usuario selecciona el botón correspondiente al experimento deseado en la pantalla principal de la aplicación, se despliega una nueva pantalla personalizada para cada experimento. En esta pantalla, se proporcionan etiquetas que indican al usuario los datos que debe ingresar para enviarlos a la interfaz. Posteriormente, la aplicación entra en modo de espera para recibir los resultados de la interfaz. Una vez recibidos, la aplicación realiza el análisis de los datos, muestra los resultados y genera la gráfica correspondiente.

La Figura 58 muestra la pantalla de resultados, que se compone de tres secciones. El primer módulo, destinado al ingreso de datos, incluye etiquetas y cajas de texto

genéricas que se adaptarán según el experimento en curso, así como botones que permiten iniciar el experimento, regresar a la pantalla principal y cerrar la aplicación. El botón "INICIAR" activa el experimento al enviar los datos necesarios a la interfaz; el botón "CERRAR" cierra la pantalla de resultados y retorna a la pantalla principal de la aplicación para que el usuario pueda seleccionar otro experimento; y el botón "SALIR DE LA APLICACIÓN" finaliza la aplicación. El segundo segmento de resultados exhibe los resultados del experimento seleccionado, presentando etiquetas personalizadas según el experimento en ejecución. Por último, el tercer segmento muestra la representación gráfica de los resultados del experimento utilizando un visor de páginas web y una librería JavaScript, sin necesidad de conexión a internet.

Figura 58.

Vista de la Pantalla de Resultados de la Aplicación



Antes de que la pantalla de resultados se muestre, el programa activa la función '*screenResult1.Initialize()*'. Esta función establece un color de fondo para la pantalla, inicializa los parámetros necesarios para mostrar la gráfica, verifica que la conexión Bluetooth con la interfaz sea correcta y coincida con el identificador inicializado en la pantalla principal correspondiente al experimento seleccionado, y también inicializa el texto de las etiquetas y las cajas de texto para el ingreso de datos. El código correspondiente es el siguiente.

```

cuando screenResult1 - .Iniciar
ejecutar
  poner global numExp a tomar el valor inicial
  poner verarrResults - .Visible como falso
  poner webvChart1 - .Visible como falso
  poner screenResult1 - .ColorDeFondo como crear color construir una lista 4 43 76
  poner webvChart1 - .UrlInicial como si tomar global development -
  entonces "file:///mnt/sdcard/Applinventor/assets/PlotchartP..."
  si no "file:///android_asset/PlotchartPhysicsLaboratory..."
  si llamar bthClientRes1 - .Conectar dirección llamar tndbMAC - .ObtenerValor
  etiqueta "ChooseMac"
  valorSiEtiquetaNoExiste ""
  entonces poner btnStart - .Visible como verdadero
  poner lblBTConnection - .Texto como "Conexión Bluetooth verificada"
  llamar InitializeScreenExperiments -
  sino poner btnStart - .Visible como falso
  poner lblBTConnection - .Texto como "Conexión Bluetooth incorrecta, verifique la cone..."

```

```

como InitializeScreenExperiments
ejecutar
  poner lblDataTitle1 - .Texto como "INGRESAR DATOS"
  poner lblNote1 - .Texto como ""
  si tomar global numExp == 11
  entonces poner screenResult1 - .Título como "Experimento sobre La Ley de OHM"
  poner lblData1 - .Visible como falso
  poner tboxData1 - .Visible como falso
  poner lblData2 - .Visible como falso
  poner tboxData2 - .Visible como falso
  poner lblData3 - .Visible como falso
  poner tboxData3 - .Visible como falso
  llamar bthClientRes1 - .EnviarTexto
  texto "10"

```

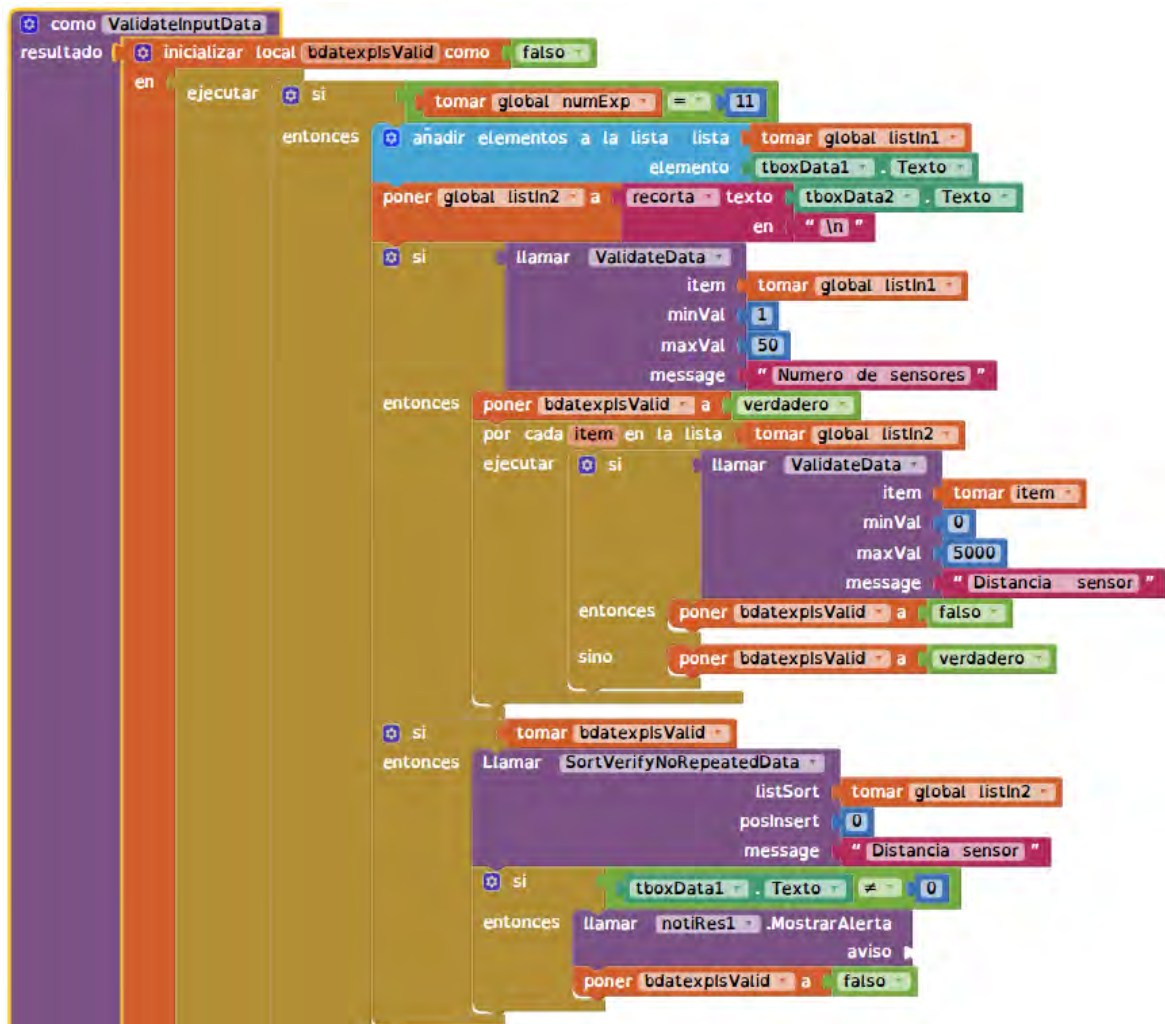
Después de que el usuario ingresa los datos del experimento y presiona el botón "INICIAR", se inicializan las variables y listas de datos correspondientes, así como las etiquetas para los títulos y subtítulos de los resultados esperados. Luego, se lleva a cabo un proceso de validación de los datos ingresados. En caso de que se detecten errores, se solicita al usuario que reintroduzca datos válidos a través de mensajes en pantalla. Además, se envía un código a la interfaz para reconfigurar el microcontrolador, si el experimento lo requiere. Finalmente, se envía una lista con todos los datos necesarios a la interfaz. El código necesario para realizar todas estas acciones se presenta a continuación.

```

cuando btnStart - .Clic
ejecutar
poner lblResult12 - .Texto - como " "
poner lblResult22 - .Texto - como " "
poner lblResult32 - .Texto - como " "
poner Etiqueta3 - .Texto - como " "
poner global listIn1 - a - crear una lista vacía
poner global listIn2 - a - crear una lista vacía
poner global listIn3 - a - crear una lista vacía
poner global Lista_X - a - crear una lista vacía
poner global Lista_Y - a - crear una lista vacía
poner global index_i - a - 0
poner global index_j - a - 0
si call ValidateInputData -
entonces si
tomar global numExp - = - 11
entonces
poner verarrResults - .Visible - como cierto
poner webvChart1 - .Visible - como cierto
poner lblResult41 - .Texto - como " GRAFICA "
poner lblResult42 - .Texto - como " Gráfica Tension Corriente "
poner lblResult11 - .Texto - como " Corriente (mA) "
poner lblResult21 - .Texto - como " Voltaje (V) "
poner Etiqueta2 - .Texto - como " Muestra "
poner global dataChain - a - " "
llamar bthClientRes1 - .EnviarTexto
texto " 20 "

```

Para validar los datos, se emplea la función *'ValidateInputData()'*. Esta función examina la lista de datos ingresados y comprueba lo siguiente: que se hayan introducido datos, que el dato ingresado sea un número y, en caso de serlo, que esté dentro de un rango de valores especificado. Posteriormente, mediante otra función, se verifica que no haya datos duplicados, dependiendo del experimento seleccionado por el usuario.



Cuando la interfaz recibe los datos enviados por la aplicación, la placa Arduino previamente programada ejecuta el experimento. Los datos obtenidos deben ser transmitidos de vuelta a la aplicación en el dispositivo Android para su procesamiento, análisis, presentación de los resultados y visualización de la gráfica correspondiente. La función *'clkResult.Timer()'* se encarga de verificar continuamente en cada ciclo de reloj

el puerto serial en busca de datos provenientes de la interfaz. Finalmente, esta función es responsable de enviar los comandos necesarios para mostrar la gráfica, tal como se ilustra en el siguiente código.

```

cuando clkResult1 . Temporizador
ejecutar si bthClientRes1 . Conectado
    llamar bthClientRes1 . BytesDisponiblesParaRecibir > 0
    entonces poner global receiveData a llamar bthClientRes1 . RecibirTexto
        númeroDeBytes (llamar bthClientRes1 . BytesDisponiblesParaRecibir)
        poner global dataChain a ""
        poner global listRes1 a recorta texto tomar global receiveData
            en ""
        por cada item en la lista tomar global listIn2
        ejecutar poner lblResult12 . Texto como unir lblResult12 . Texto
            " \n "
            tomar item
        por cada item en la lista tomar global listRes1
        ejecutar poner lblResult22 . Texto como unir lblResult22 . Texto
            " \n "
            tomar item
        poner lblResult32 . Texto como unir " Numero datos distancia: "
            longitud de la lista lista tomar global listIn2
            " , Numero de tiempos: "
            longitud de la lista lista tomar global listRes1
        por cada index desde 1
            hasta longitud de la lista lista tomar global listIn2
            en incrementos de 1
        ejecuta poner global dataChain a unir tomar global dataChain
            unir " {x:"
            seleccionar elemento de la lista tomar global listIn2
            indice tomar index
            " }:"
            seleccionar elemento de la lista tomar global listRes1
            indice tomar index
            " } "
        si tomar index <= longitud de la lista lista tomar global listIn2
        entonces poner global dataChain a unir tomar global dataChain
            " , "
        llamar RefreshPlotchart
    
```

Pantalla de datos.

Para guardar los datos experimentales mediante la aplicación, y permitir su posterior procesamiento y análisis, se debe presionar el botón *SAVE DATA* que aparece al final de los datos. La confirmación de que el evento se ha completado se realiza a través

de un mensaje que indica: “Datos guardados”. A continuación, se muestra el código para crear la base de datos.



Estos datos quedan registrados en una base de datos en el smartphone, donde se pueden visualizar en cualquier momento, exportar a una computadora mediante la web y eliminar si es necesario. Para ello, se debe presionar el botón SALIR en la pantalla de resultados, lo que retorna a la pantalla principal. Desde allí, se presiona el botón DATA VIEW para abrir la pantalla de datos.

La Figura 59 muestra la pantalla de datos, que consta de un espacio para presentar los datos, con sus respectivas etiquetas y cajas de texto, así como botones para exportar los datos, borrarlos y retornar a la pantalla principal.

Figura 59.

Vista de la Pantalla de Datos



Para exportar los datos a una hoja de cálculo de Google, se debe presionar el botón *EXPORT*. El programa ejecuta la función *btnExportData*, que exporta los datos a la web. Para ello, es necesario construir una URL que incluya la dirección de la hoja de cálculo, junto con los campos y nombres de las variables. El código para exportar y llamar a la hoja de cálculo es el siguiente:

```

cuando btnExportData .Clic
  ejecutar
    poner global numElement a longitud de la lista lista tomar global Lista_X
    por cada número desde 1
      hasta tomar global numElement
        en incrementos de 1
          ejecuta
            poner Web1 . Uri como unir " https://docs.google.com/forms/d/1BjNq4tRUpeMs50w... "
              seleccionar elemento de la lista tomar global Lista_X
                índice tomar número
              " "
              " &entry.1747858992= "
            seleccionar elemento de la lista tomar global Lista_Y
              índice tomar número
            llamar Web1 . Obtener

como LlamarExcel Acción
  ejecutar
    poner Web1 . Uri como " https://docs.google.com/forms/d/1BjNq4tRUpeMs50w... "
    llamar Web1 . PublicarTexto
      texto llamar Web1 . ConstruirSolicitudDeDatos
        lista construir una lista construir una lista " entry.556325523 "
          tomar Acción
  
```

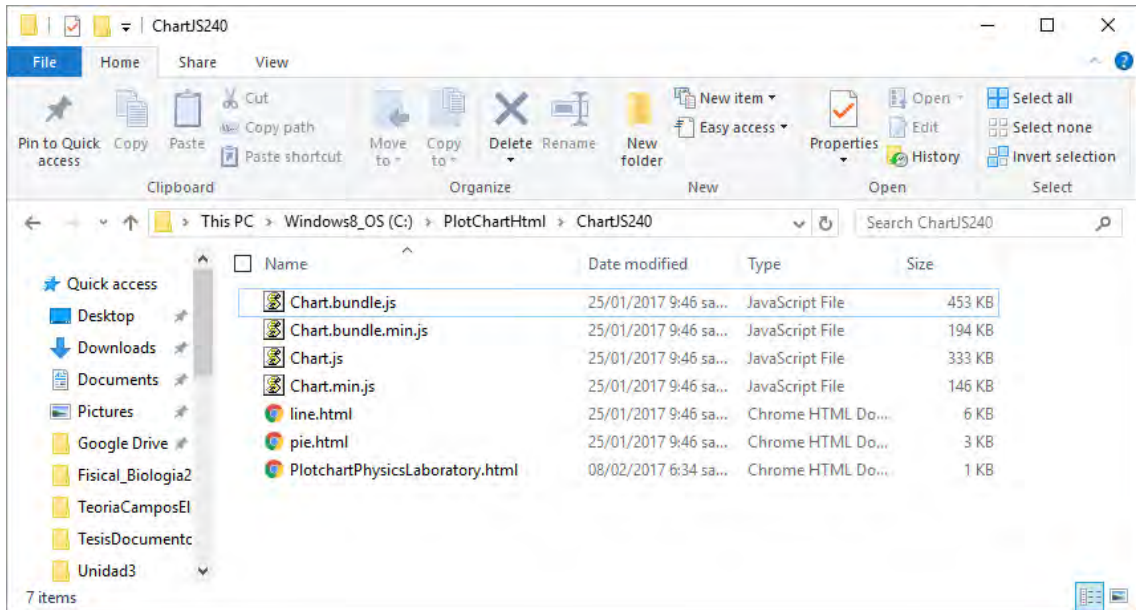
Para editar las celdas de los campos de la hoja de cálculo de Google de forma remota desde nuestra aplicación, debemos ingresar a Google Drive y crear un formulario para la hoja de cálculo con los campos necesarios para las variables. Posteriormente, recopilamos la dirección de la hoja y de sus campos, y estos se incluyen en el código de la aplicación.

4.2.3. Graficas con CHARTJS.

Para utilizar la librería **Chart.js** que permite visualizar datos de forma gráfica usando el elemento `<canvas>` de HTML5, primero se debe copiar el archivo *chart.min.js* en la carpeta JavaScript (JS) o directorio donde se va a realizar el proyecto.

Figura 60.

Librería *chart.min.js* en Carpeta del Proyecto



Se debe crear un archivo HTML en la carpeta del proyecto e incluir la librería **chart.js**.

```

1  <!doctype html>
2  <html>
3  <head>
4      <title>Physics Laboratory Chart</title>
5      <script src="Chart.min.js"></script>
6  </head>
7  <body>
8  </body>
9  </html>

```

Para realizar un gráfico, se debe crear un elemento ‘<canvas>’ de HTML5 en la sección ‘<body>’ del archivo HTML y luego crear un script para inicializar la clase gráfica (en este proyecto, se ha utilizado un gráfico de dispersión o *scatter chart*). A continuación, se debe recuperar el contexto 2D del lienzo donde se desea dibujar el gráfico.

```

16 <body>
17   <div style="width:90%; margin:auto;">
18     <canvas id="canvas"></canvas>
19   </div>
20   <script>
21     window.onload=function(){
22       var ctx = document.getElementById("canvas").getContext("2d");
23       window.myScatter = Chart.Scatter(ctx,JSON.parse(plotData))
24     }
25   </script>
26 </body>
27 </html>

```

Por último, es necesario insertar los datos que se desean graficar en formato JSON, el cual es un formato ligero de intercambio de datos fácilmente utilizable en JavaScript. Una de sus principales ventajas es que puede ser interpretado por cualquier lenguaje de programación. En la documentación de la librería Chart.js se encuentran diversas opciones de diseño para modificar parámetros como tamaño de fuente, color, estilos de la gráfica y otros disponibles para cada tipo de gráfico (ChartJS, 2016).

```

23 var plotData = { "type": "line",
24                 "data": {
25                   "datasets": [
26                     {
27                       "label": "Laboratory dataset",
28                       "borderColor": "rgb(255,99,132)",
29                       "backgroundColor": "rgba(255,99,132,0.2)",
30                       "data": [{"x":0,"y":0},{x":1,"y":1},{x":2,"y":4},
31                                {"x":3,"y":9},{x":4,"y":16},{x":5,"y":25}]
32                     }
33                   ]
34                 },
35                 "options": {
36                   "title": {
37                     "display": true,
38                     "text": "Physics Laboratory"
39                   }
40                 }
41               }

```

Luego de ejecutar todo el código generado anteriormente, obtenemos una gráfica como la que se muestra en la figura 61.

Figura 61.

Gráfica con Datos de Prueba



4.3. DESARROLLO DE EXPERIMENTOS DE PRUEBA

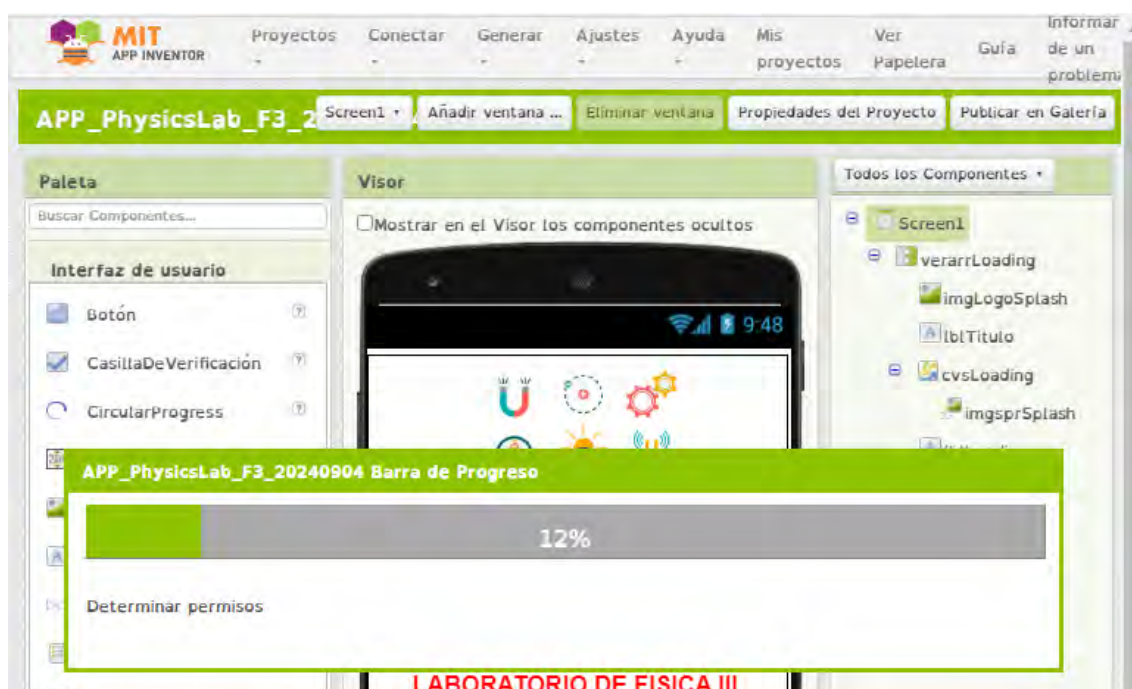
Posteriormente al diseño, construcción y pruebas de los módulos automatizados implementados con las tarjetas Arduino, se llevaron a cabo los experimentos propuestos en el presente trabajo, obteniendo resultados que permitieron cumplir con las expectativas del proyecto. A continuación, se presenta la secuencia de procedimientos realizados para el desarrollo de los experimentos.

4.3.1. Ejecución del Software de inicio

Se generaron dos archivos `App_PhysicsLab.apk` y `App_PhysicsLab_F3.apk` con extensión `.apk`, como se muestra en la Figura 62. Estos archivos se emplean para la distribución e instalación de componentes empaquetados diseñados para el sistema operativo Android, siendo compatible con diversos dispositivos móviles, tales como teléfonos inteligentes, tabletas, computadoras portátiles, relojes inteligentes, televisores inteligentes, entre otros.

Figura 62.

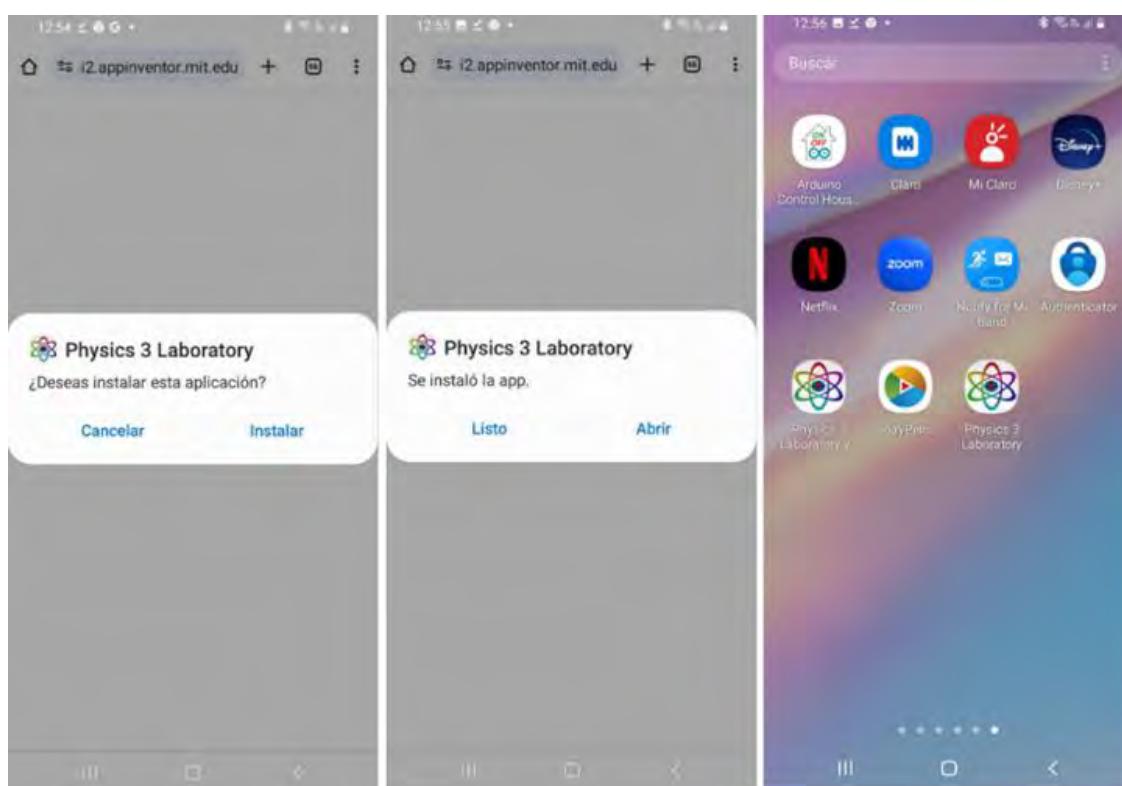
Generación del Archivo de Instalación `App_PhysicsLab_F3.apk`



El archivo .apk generado se instala en cualquier dispositivo móvil con sistema operativo Android. La Figura 63 muestra el proceso de instalación en un celular inteligente con la última versión del sistema operativo Android (versión 8.0.0), donde los íconos representados por el modelo del átomo de Sommerfeld corresponden a las aplicaciones instaladas.

Figura 63.

Proceso de Instalación de la Aplicación en un Celular Inteligente



4.3.2. Configuración Inicial del Aplicativo

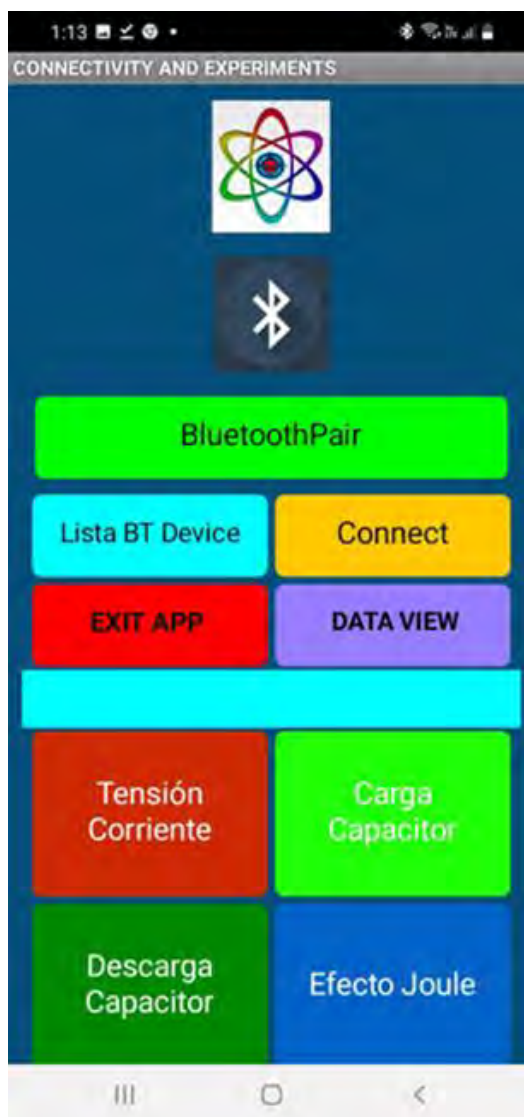
Cuando la aplicación se inicia en un dispositivo Android, se muestra una pantalla de bienvenida durante unos segundos. Esta pantalla incluye elementos como una imagen que proporciona una representación visual de la diversidad experimental de la Física, y un ícono dinámico que señala la carga interna de los recursos necesarios, como se observa en la Figura 64.

Figura 64.*Pantalla de Bienvenida*

Tras la pantalla de bienvenida, la aplicación despliega la pantalla principal (figura 65), donde el usuario configura los parámetros necesarios para realizar los experimentos. Estas configuraciones incluyen el emparejamiento de la conexión inalámbrica Bluetooth, el establecimiento de la conexión y la selección del experimento.

Figura 65.

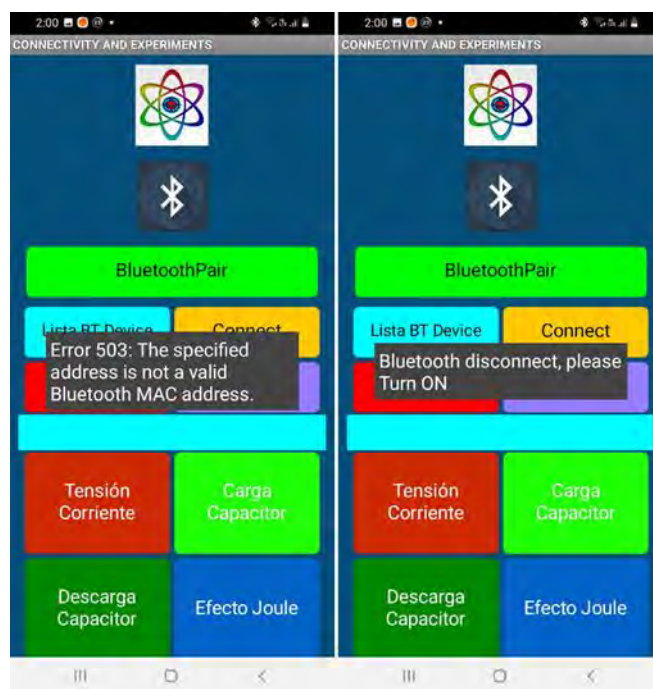
Pantalla Principal de la Aplicación



La aplicación utiliza Bluetooth como protocolo de comunicación para interactuar con la interfaz. Bluetooth está diseñado para la comunicación inalámbrica entre dispositivos de bajo consumo y con un alcance corto. La primera acción consiste en verificar si el Bluetooth está activado en el dispositivo móvil donde está instalada la aplicación. En caso de que esté inactivo, la aplicación solicitará al usuario que siga los pasos necesarios para activarlo, como se ilustra en la figura 66.

Figura 66.

Proceso para Activar el Bluetooth del Dispositivo Android

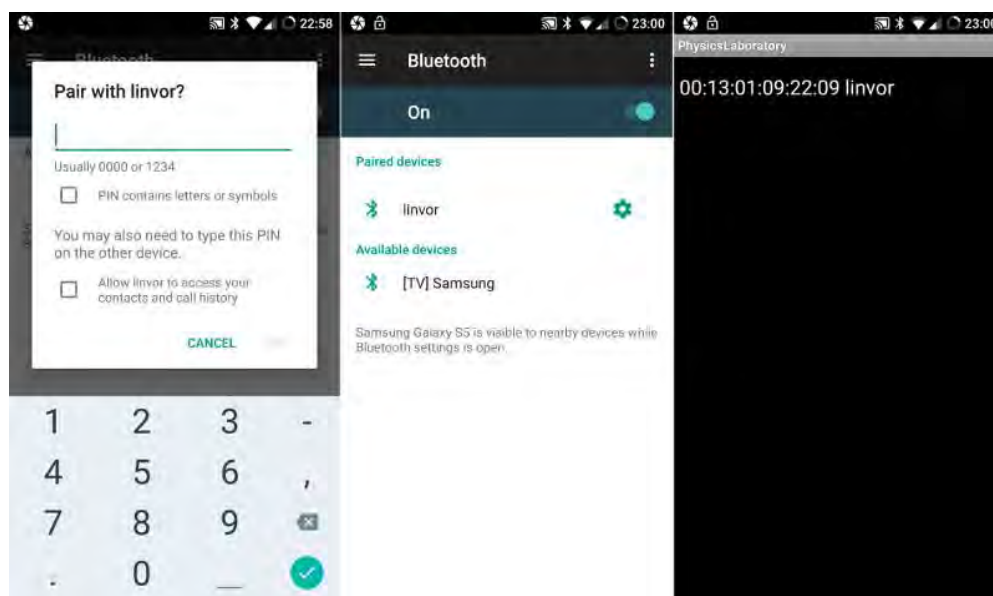


Una vez que se ha activado el Bluetooth en el dispositivo móvil, el siguiente paso es vincular o emparejar el dispositivo Android con la interfaz. Para realizar este procedimiento, se presiona el botón *"BLUETOOTH PAIR"*, que abre una pantalla que facilita la búsqueda de todos los dispositivos Bluetooth disponibles. Se identifica el dispositivo correspondiente a la interfaz deseada y se inicia el proceso de vinculación. Durante este procedimiento, se solicita introducir una contraseña, generalmente establecida por defecto como "0000" o "1234". Al seleccionar el botón *"LISTA DE DISPOSITIVOS BT"*, se despliega una pantalla que muestra todos los dispositivos actualmente vinculados. Se elige el dispositivo asociado a la interfaz con la que se desea trabajar. Finalmente, al presionar el botón *"CONECTAR"*, se lleva a cabo la conexión entre el dispositivo Android y la interfaz, permitiendo la realización de los experimentos presentados en la pantalla. Todo este procedimiento, detallado en la figura 67, es ejecutado por el usuario mediante un proceso de validación que ofrece notificaciones en caso de errores en los pasos. Es importante destacar que la conexión entre el dispositivo

Android y la interfaz solo se realiza una vez; en ejecuciones posteriores de la aplicación con la misma interfaz, la conexión se establece automáticamente, simplificando así el proceso.

Figura 67.

Proceso de Conexión del Dispositivo Android con la Interface



Hasta que se establezca correctamente la conexión entre el dispositivo Android y la interfaz, la aplicación impide al usuario seleccionar cualquiera de los experimentos disponibles en el menú.

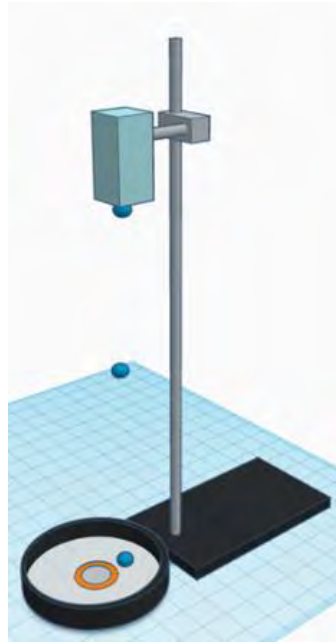
4.3.3. Experimento de Caída Libre

4.3.3.1. Montaje del Equipo para Caída Libre

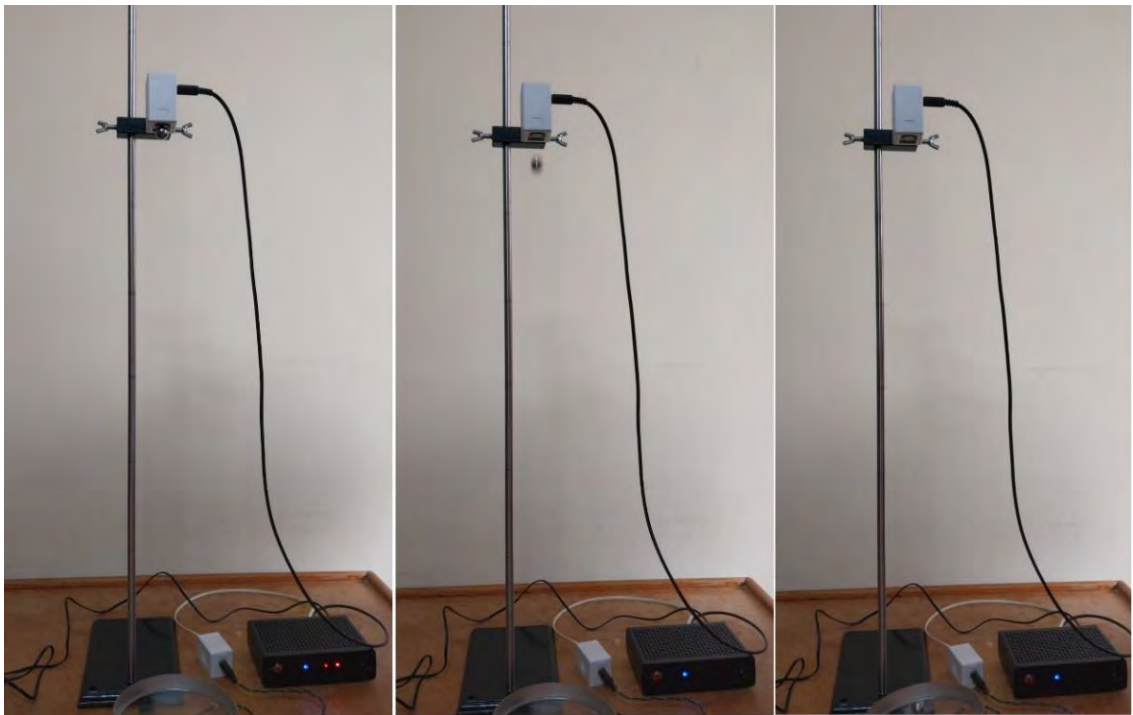
El montaje mecánico de este experimento incluye un soporte universal, abrazaderas, un electroimán, un sensor con piezoeléctrico, una interfaz, un circuito para la adaptación de la señal piezoeléctrica a digital (con eliminador de rebotes) y los cables de conexión. La figura 68 presenta el diseño CAD que sirve como guía para el armado del experimento, mientras que la figura 69 muestra una vista del montaje completo con todos los componentes.

Figura 68.

Diseño CAD para Caída libre

**Figura 69.**

Montaje Mecánico para el Experimento de Caída Libre



4.3.3.2. Aplicación para Caída Libre

En la pantalla principal de la aplicación *App_PhysicsLab.apk*, al presionar el botón *CAÍDA LIBRE*, se abre una pantalla personalizada para el ingreso de datos, como muestra la figura 70. Primero se muestra un mensaje que indica si la conexión con la Interface es correcta. Luego, solicita ingresar el Número de medidas que se desea realizar, con un máximo de 21 medidas, limitado por el número de caracteres que se pueden enviar o recibir, según el protocolo de comunicación. A continuación, el programa solicita la Altura para cada medida, que es la distancia entre el electroimán y la esfera en relación con la posición del sensor.

Figura 70.

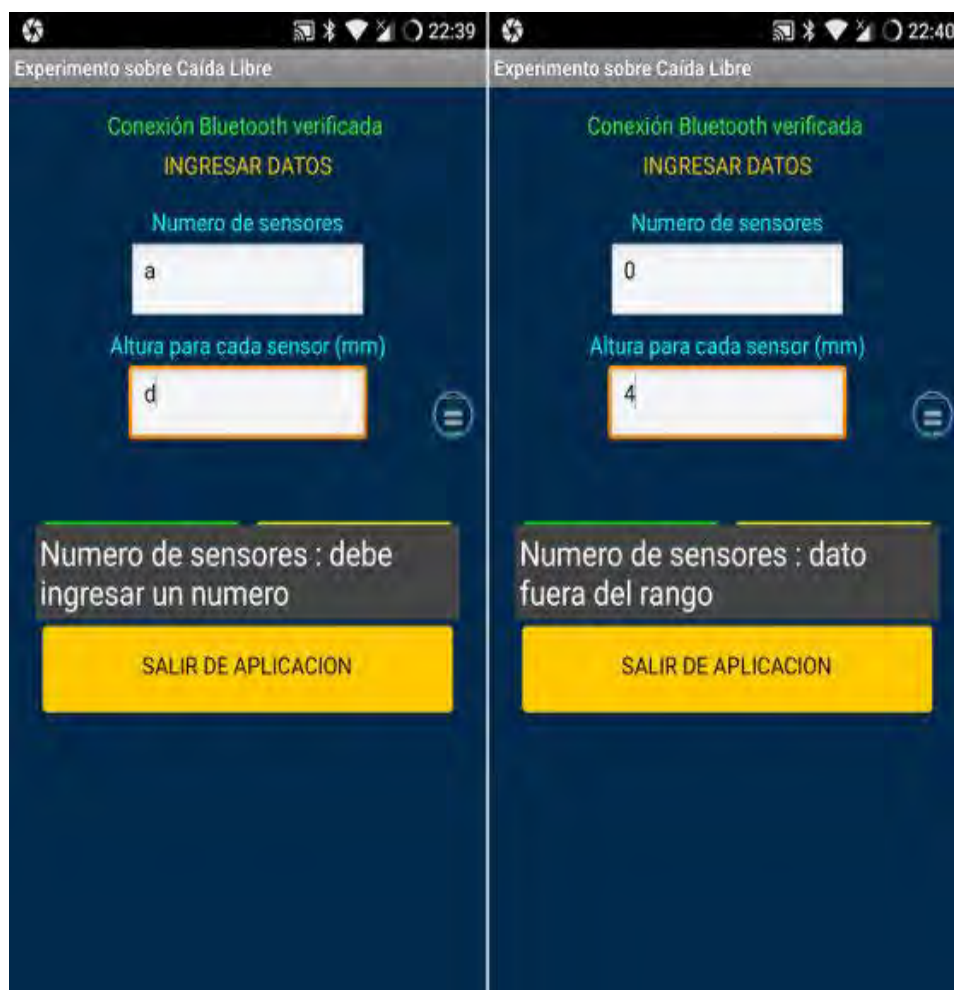
Pantalla para el Ingreso de Datos



Al presionar el botón "INICIAR", la aplicación realiza inicialmente una validación de los datos ingresados. Si encuentra errores, notifica al usuario mediante mensajes que aparecen en la pantalla durante unos segundos, como se ilustra en la figura 71. Estos errores pueden deberse a la omisión de datos esenciales, la introducción de letras en lugar de números, datos fuera de rango, entre otros. Una vez que los datos han sido validados correctamente, la aplicación envía la información necesaria a la interfaz, dando inicio a la ejecución del experimento. Mientras tanto, la aplicación entra en modo de espera, aguardando los resultados que la interfaz debe enviar.

Figura 71.

Validación de Datos



Cuando los datos llegan a la interfaz, se activa el electroimán conectado a esta durante un intervalo de cinco segundos. Durante este período, los LED indicadores correspondientes se iluminan para permitir al usuario realizar una verificación visual y colocar el cuerpo metálico en la base del electroimán. Tras desactivar el electroimán, se registra el tiempo inicial y el cuerpo comienza su caída libre. Cada impacto con el sensor se registra en un vector de datos, que posteriormente se enviará a la aplicación para su análisis.

La interfaz realiza, de forma independiente a la aplicación en Android, un proceso de verificación y validación de los datos capturados. Este procedimiento abarca situaciones como la ausencia de detección del cuerpo por parte del sensor, para lo cual se establece un tiempo de espera, o la conexión incorrecta de los sensores a la interfaz, entre otros posibles casos.

Una vez que la aplicación Android recibe correctamente los datos enviados por la interfaz, se habilita una pantalla donde se presentan los resultados junto con la gráfica correspondiente, como se ilustra en la figura 72. En esta pantalla, se muestra una tabla con tres columnas: la primera identifica cada dato, la segunda registra el momento en que el cuerpo activa el sensor, y la tercera indica la altura desde la cual se dejó caer la esfera metálica. La gráfica complementaria representa la relación entre la distancia y el tiempo, generada a partir de los datos almacenados.

Figura 72.

Pantalla de Resultados, Gráfico y Datos Guardados del Experimento de Caída Libre



4.3.4. Experimento de péndulo simple

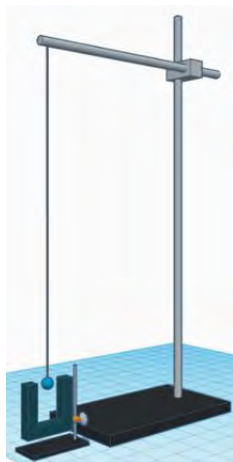
4.3.4.1. Montaje del Equipo para Péndulo Simple

Para este experimento, el montaje mecánico incluye un soporte universal, abrazaderas, un electroimán, un sensor de fotopuerta, una interfaz y los cables de conexión.

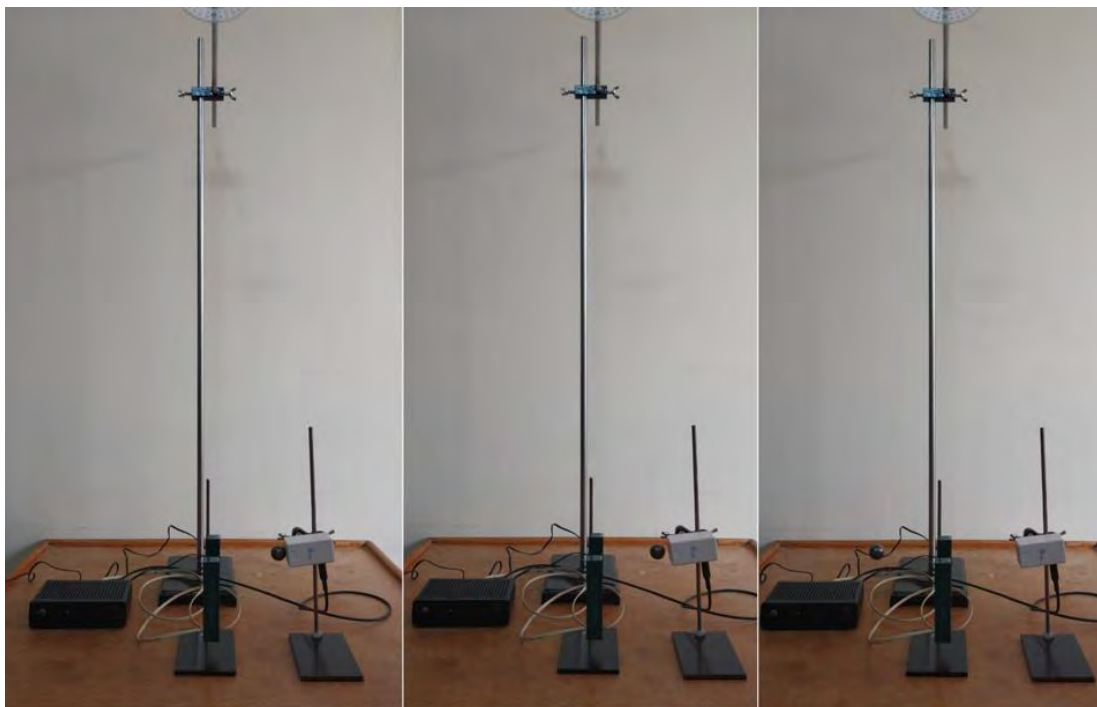
Al igual que en el caso anterior, la figura 73 presenta un diseño CAD que sirve como guía para el armado del experimento, mientras que la figura 74 muestra el montaje completo con todos sus componentes.

Figura 73.

Diseño CAD para Péndulo Simple

**Figura 74.**

Montaje Mecánico para el Experimento de Péndulo Simple



4.3.4.2. Aplicación para Péndulo Simple

En la pantalla principal de la aplicación *App_PhysicsLab.apk*, al presionar el botón "PÉNDULO SIMPLE", la aplicación despliega una nueva pantalla personalizada para el ingreso de datos, mostrando un mensaje que indica si la conexión con la interfaz es correcta, como se ilustra en la figura 75.

En esta pantalla, el primer dato a ingresar es el "Número de oscilaciones". A continuación, se debe introducir la "Longitud de la cuerda", para lo cual es necesario medir la longitud de la cuerda más el radio de la esfera. Este ajuste es importante para que, en el análisis de los datos, el cuerpo sea considerado como una masa puntual.

Figura 75.

Pantalla para el Ingreso de Datos para el Experimento del Péndulo Simple

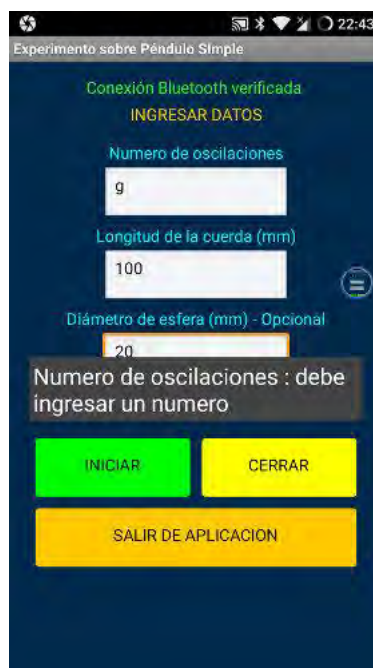


Al igual que en el escenario anterior de la caída libre, al presionar el botón "INICIAR", el programa realiza una verificación de los datos ingresados. En caso de detectar datos incorrectos, como la omisión de información esencial, la introducción de letras en lugar de números, datos fuera del rango esperado, entre otros, la aplicación notifica al usuario para que corrija los datos mediante mensajes de notificación, como se muestra en la figura 76. Una vez que los datos han sido validados, la aplicación envía la

información necesaria a la interfaz y comienza la ejecución del experimento. Durante este proceso, la aplicación entra en modo de espera, aguardando los resultados que la interfaz debe enviar para llevar a cabo el registro de datos y presentarlos.

Figura 76.

Validación de Datos Ingresados para el Experimento del Péndulo Simple



Cuando los datos llegan a la interfaz, se activa un electroimán durante cinco segundos para que el usuario pueda colocar el cuerpo metálico en su posición correspondiente, a uno de los lados del sensor, formando un ángulo menor a diez grados con la vertical. Después de desactivar el electroimán, el cuerpo comienza a moverse, y se registra el tiempo inicial una vez que el cuerpo pasa por el sensor por primera vez. Posteriormente, cada vez que el cuerpo pasa por el sensor dos veces, lo que corresponde a una oscilación completa, se registra el tiempo y se almacena en un vector de datos que será enviado a la aplicación para su análisis posterior.

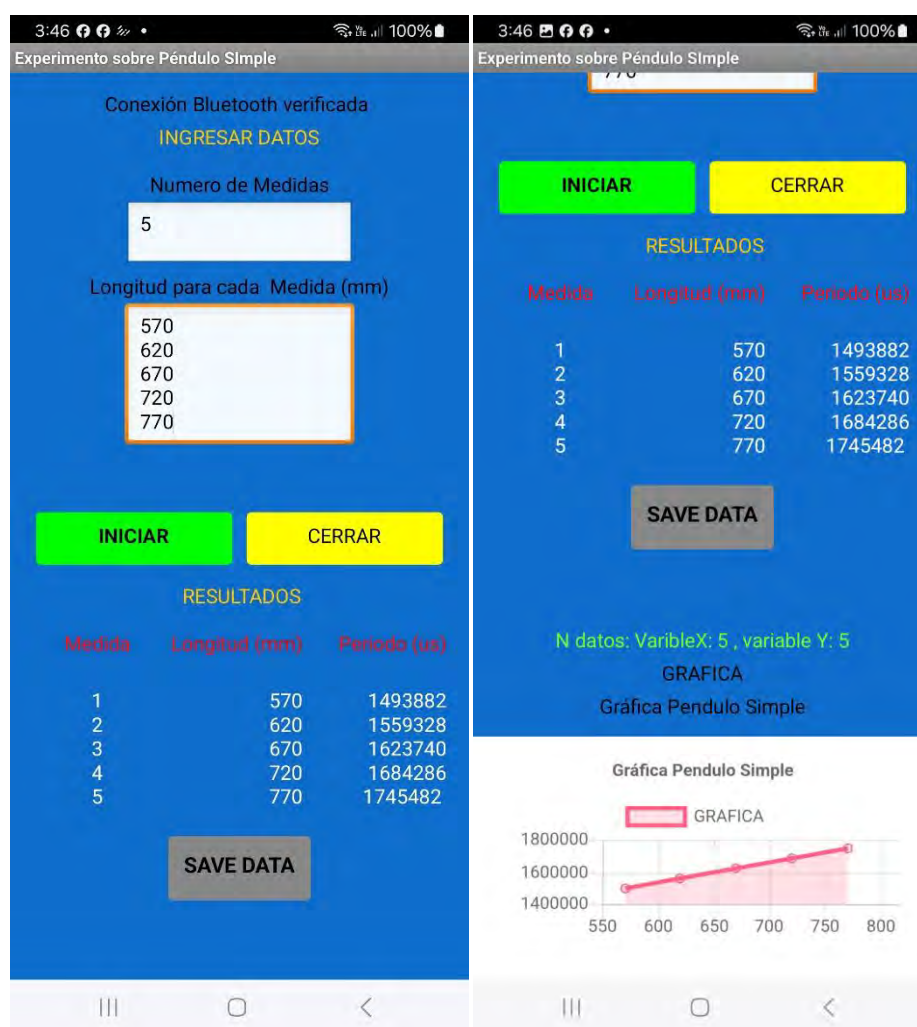
Además de la aplicación en Android, la interfaz también realiza una verificación y validación de los datos capturados en casos como: si el cuerpo no ha pasado por el

sensor dentro de un tiempo de espera determinado o si el cuerpo no ha completado todas las oscilaciones indicadas por el usuario, entre otros casos.

Una vez que la aplicación recibe correctamente los datos enviados por la interfaz, activa una pantalla para mostrar los resultados y generar la gráfica correspondiente, como se ilustra en la figura 77. La pantalla de resultados presenta una tabla con tres columnas: la primera indica el número de oscilación, la segunda muestra el período de cada oscilación, y la tercera muestra la longitud de la cuerda, considerando el cuerpo como una masa puntual.

Figura 77.

Pantalla de Resultados, Gráfico y Datos Guardados del Experimento de péndulo simple



4.3.5. Experimento del Movimiento Rectilíneo Uniforme (MRU)

4.3.5.1. Montaje del Equipo para MRU

Aunque el móvil cuenta con un solo motor y las llantas están alineadas, en ocasiones puede desviarse ligeramente. Para garantizar que mantenga una trayectoria recta durante el experimento de MRU, se incluyó un carril como accesorio, además de los siguientes elementos: un carrito, un emisor IR ubicado en el punto de partida y conectado a la interfaz mediante un puerto configurado para actuadores, seis sensores de fotopuerta, la interfaz y los cables de conexión. Como en el caso anterior, la figura 78 muestra el arreglo experimental completamente ensamblado con todos los componentes.

Figura 78.

Montaje Mecánico para el Experimento de Movimiento Rectilíneo Uniforme



4.3.5.2. Aplicación para el MRU

En la pantalla principal de la aplicación App_PhysicsLab.apk, cuando se activa el botón MRU, la aplicación muestra una nueva pantalla personalizada para el ingreso

de datos, con un mensaje que indica si la conexión con la Interfaz es correcta, como se muestra en la figura 79. El primer dato a ingresar es el número de sensores; luego se debe ingresar la posición de los sensores, para lo cual se debe medir la distancia entre los sensores.

Figura 79.

Pantalla para el Ingreso de datos para el Experimento de M.R.U.



Al igual que en los escenarios anteriores de caída libre y péndulo simple, al activar el botón "INICIAR", el programa lleva a cabo una verificación de los datos ingresados. En caso de detectar datos incorrectos, la aplicación notifica al usuario para que los corrija. Una vez que los datos han sido validados, la aplicación envía la información necesaria a la interfaz y comienza la ejecución del experimento. Durante este proceso, la aplicación entra en modo de espera, aguardando los resultados que la interfaz debe enviar.

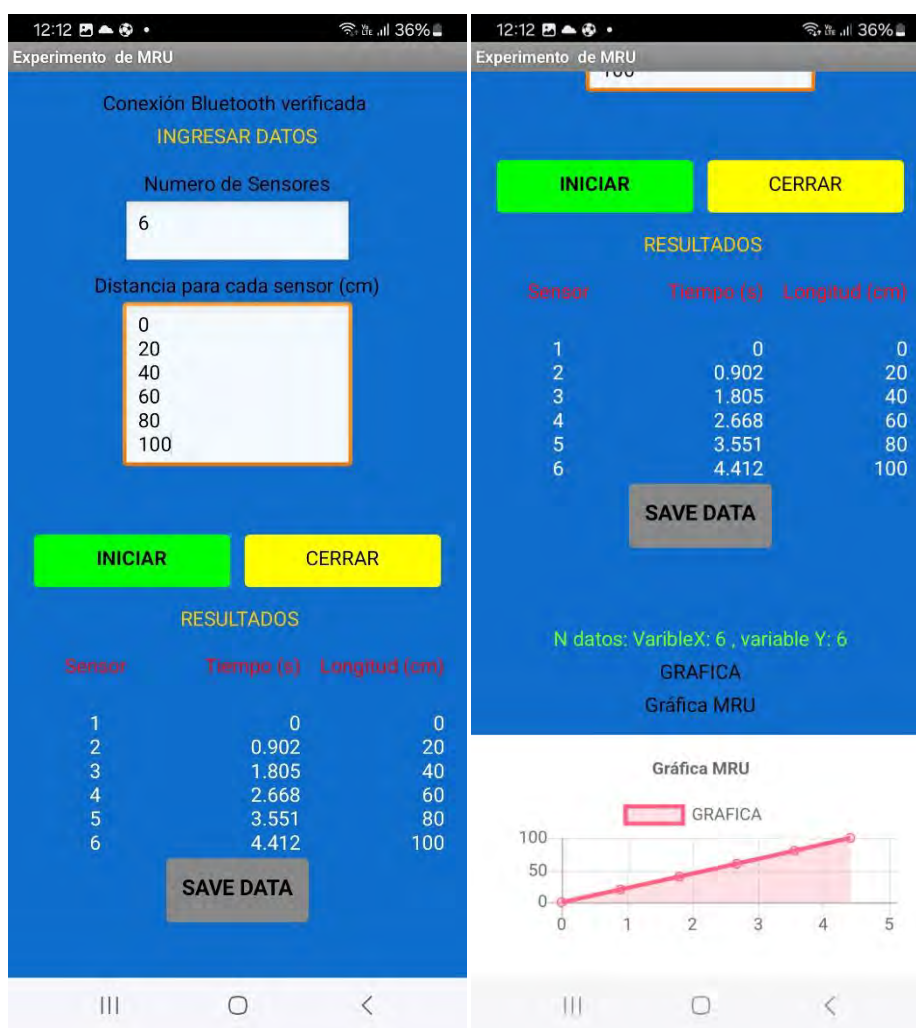
Cuando los datos llegan a la interfaz, se activa un LED IR durante cinco segundos, iniciando el conteo regresivo para el movimiento del móvil. Una vez que el

conteo llega a cero, el móvil comienza a desplazarse. Cada vez que pasa por uno de los sensores, se registra el tiempo correspondiente, que se almacena en un vector de datos para su posterior envío a la aplicación y análisis.

Cuando la aplicación recibe los datos enviados correctamente por la interfaz, se activa una pantalla que muestra los resultados y genera el gráfico correspondiente, como se ilustra en la figura 80. La pantalla de resultados incluye una tabla con tres columnas: la primera indica el sensor, la segunda registra el tiempo asociado a cada sensor, y la tercera muestra la distancia recorrida. El gráfico representa la relación entre la distancia recorrida y el tiempo transcurrido para cubrir dichas distancias.

Figura 80.

Pantalla de Resultados, Gráfico y Datos para el Experimento sobre el M.R.U.



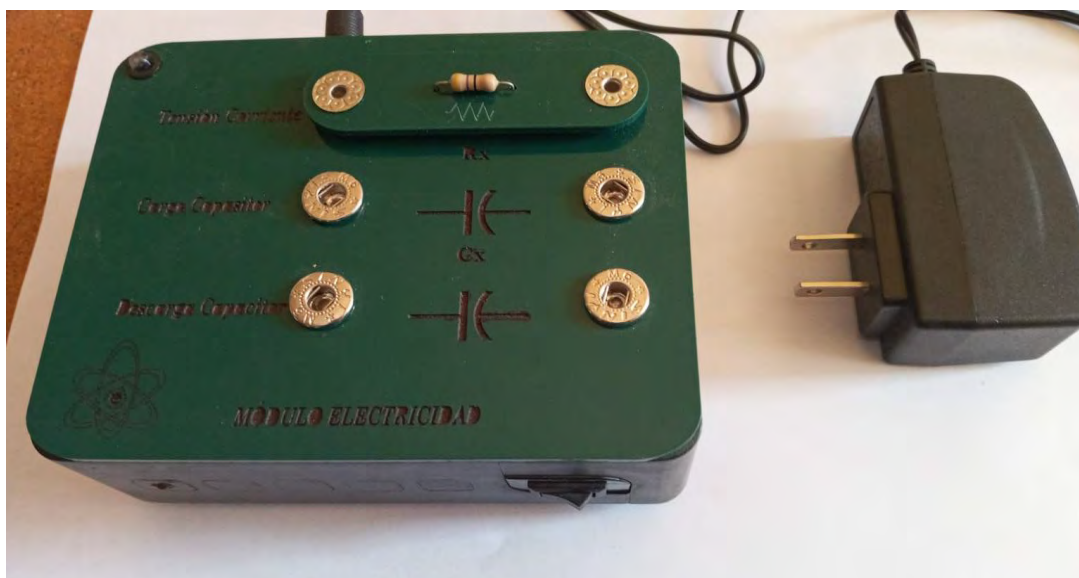
4.3.6. Experimento sobre la ley de Ohm

4.3.6.1. Montaje del Equipo para los Experimentos de Electricidad

En estos experimentos, la fuente de poder, el voltímetro y el amperímetro se han integrado a la interfaz, creando un sistema ordenado y eficiente. Los componentes de prueba, como resistencias y capacitores, se conectan de manera práctica a una placa diseñada con terminales metálicos (broches), que facilitan su montaje y conexión. Esta placa ha sido adherida a la caja de la interfaz, lo que se ilustra claramente en la Figura 83. Además, la Figura 84 destaca el lugar exacto donde debe conectarse el resistor de prueba, esencial para llevar a cabo el experimento sobre la ley de Ohm.

Figura 81.

Interface de Electricidad con la resistencia para el experimento de la ley de Ohm



4.3.6.2. Aplicación para la Ley de Ohm

En la pantalla principal de la aplicación *App_PhysicsLab_F3.apk*, al presionar el botón *TENSIÓN CORRIENTE*, se abre una nueva pantalla personalizada para el ingreso de datos, donde se muestra un mensaje que confirma si la conexión con la interfaz es correcta, como se ilustra en la figura 82. En este caso, no se requiere ingresar datos.

Figura 82.

Pantalla para el Inicio del Experimento de ley de OHM



De manera similar a los escenarios previos de caída libre, péndulo simple y MRU, al pulsar el botón "INICIAR", la aplicación transmite la información necesaria a la interfaz y comienza el experimento. La aplicación entra en modo de espera mientras aguarda los resultados que la interfaz debe enviar. Cuando los datos llegan correctamente a la aplicación, se activa una pantalla para mostrar los resultados y generar el gráfico correspondiente, como se observa en la figura 83. Esta pantalla de resultados incluye una tabla con tres columnas: la primera para el dato, la segunda para la corriente y la tercera para la tensión. El gráfico refleja la relación entre la tensión y la intensidad de la corriente, así como los datos almacenados en la base de datos.

Figura 83.

Pantalla de Resultados y Gráfico para el Experimento sobre la Ley de Ohm



Ciertos dispositivos, como los focos, Leds, son no óhmicos, lo que implica que sus gráficos de diferencia de potencial eléctrico en función de la corriente no siguen una línea recta. En el caso de dispositivos no óhmicos, no es posible emplear la ley de Ohm.

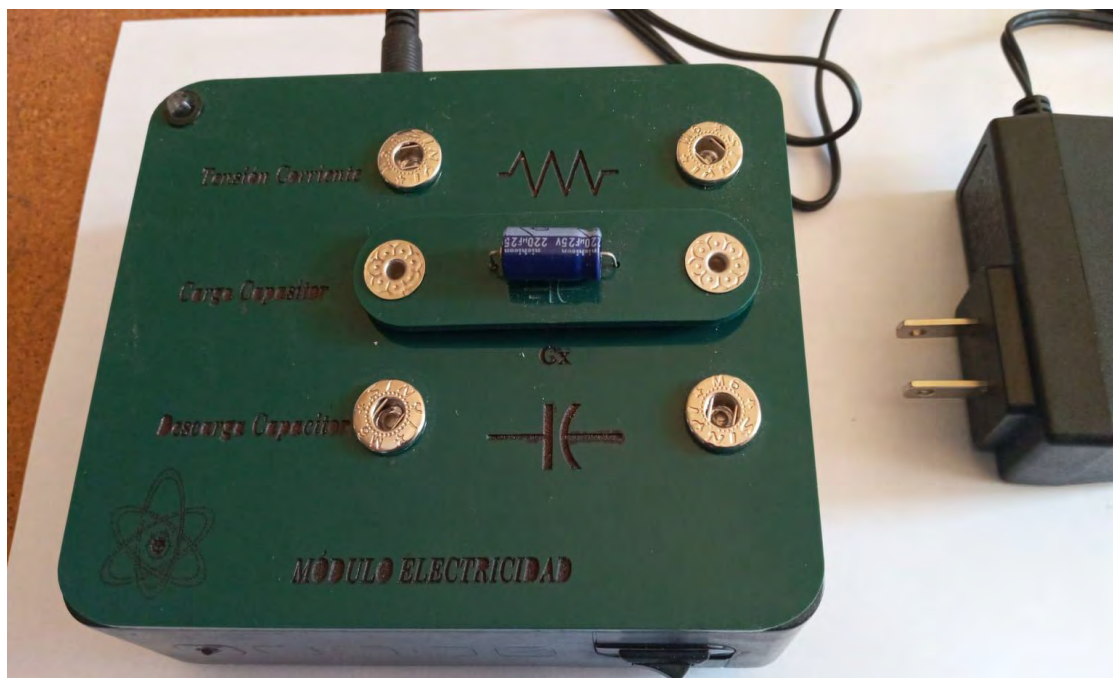
4.3.7. Experimento sobre la Carga de un Capacitor

4.3.7.1. Montaje del Equipo para la Carga de un Capacitor

Para llevar a cabo el experimento de carga de un capacitor, se conecta el capacitor de prueba en la segunda ubicación de la placa de la interfaz, asegurándose de respetar la polaridad indicada. La figura 84 ilustra el modo, la posición y la orientación del componente de prueba.

Figura 84.

Interface de Electricidad para el Experimento de Carga de un Capacitor



4.3.7.2. Aplicación para la Carga de un Capacitor

En la pantalla principal de la aplicación *App_PhysicsLab_F3.apk*, al presionar el botón *CARGA CAPACITOR*, se abre una pantalla personalizada para el ingreso de datos. En esta nueva pantalla, se muestra un mensaje que confirma si la conexión con la interfaz es correcta, tal como se observa en la figura 85.

El primer dato que se debe ingresar es el número de muestras, con un máximo permitido de 21. A continuación, se introduce el tiempo de muestra, que corresponde al intervalo entre cada muestra. Para un capacitor de $220 \mu F$, se sugiere un tiempo de muestreo de 25 ms, ya que permite observar adecuadamente el comportamiento no lineal del experimento.

Figura 85.

Pantalla para el Inicio del Experimento Carga de un Capacitor



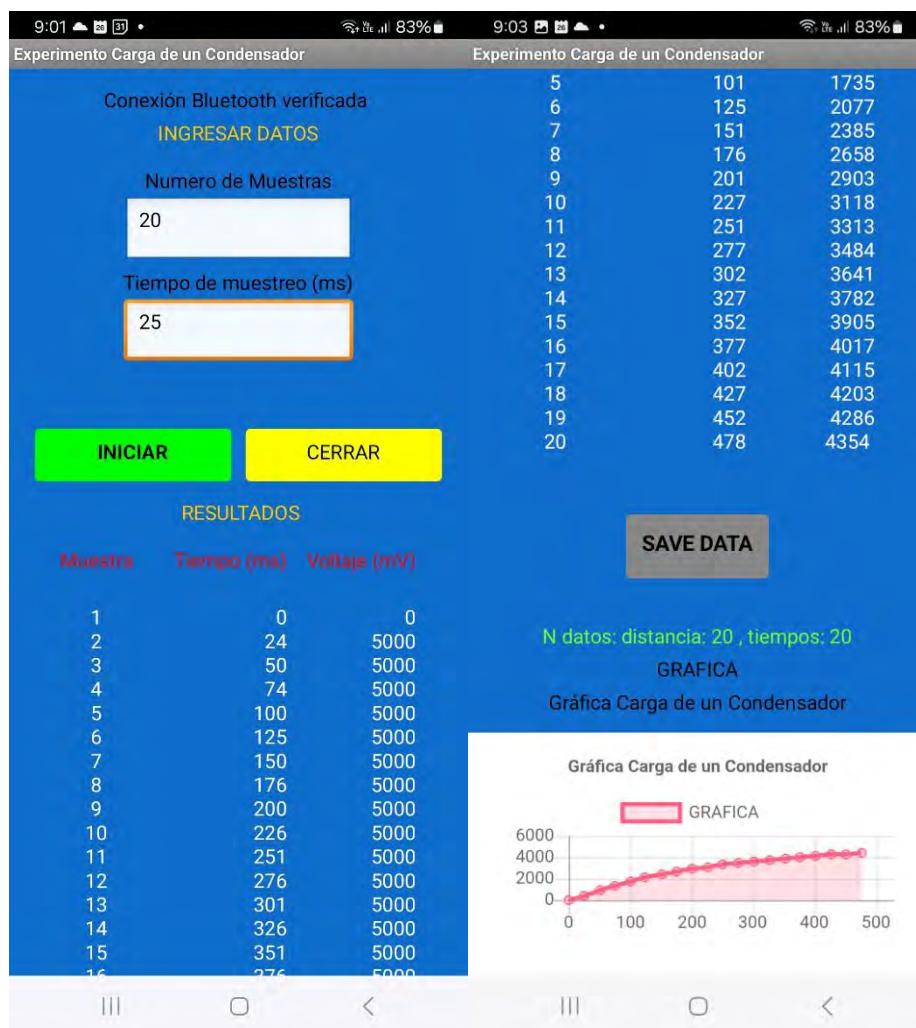
Al igual que en los escenarios anteriores, al pulsar el botón *INICIAR*, la aplicación envía la información necesaria a la interfaz y da inicio a la ejecución del experimento. Durante este proceso, la aplicación entra en modo de espera, guardando los resultados enviados por la interfaz.

Todas las aplicaciones desarrolladas en este trabajo incluyen la validación de los datos ingresados. Una vez que los datos son recibidos correctamente desde la interfaz, la aplicación activa una pantalla donde se presentan los resultados y se genera la gráfica correspondiente, tal como se muestra en la figura 86. La pantalla de resultados presenta una tabla con tres columnas: la primera indica el dato, la segunda muestra el tiempo, y

la tercera la tensión del capacitor. La gráfica ilustra la relación entre la tensión y el tiempo, así como los datos almacenados en la base de datos.

Figura 86.

Pantalla de Resultados y Gráfica para el Experimento de Carga de un Capacitor



4.3.8. Experimento sobre la descarga de un capacitor

4.3.8.1. Montaje del Equipo para la Descarga de un Capacitor

Para el experimento de la descarga de un capacitor, este se conecta en la tercera ubicación de la placa de la interfaz, respetando la polaridad indicada. La figura 87 muestra el modo, la posición y la orientación del componente de prueba.

Figura 87.

Interface de Electricidad para el Experimento de Descarga del Capacitor



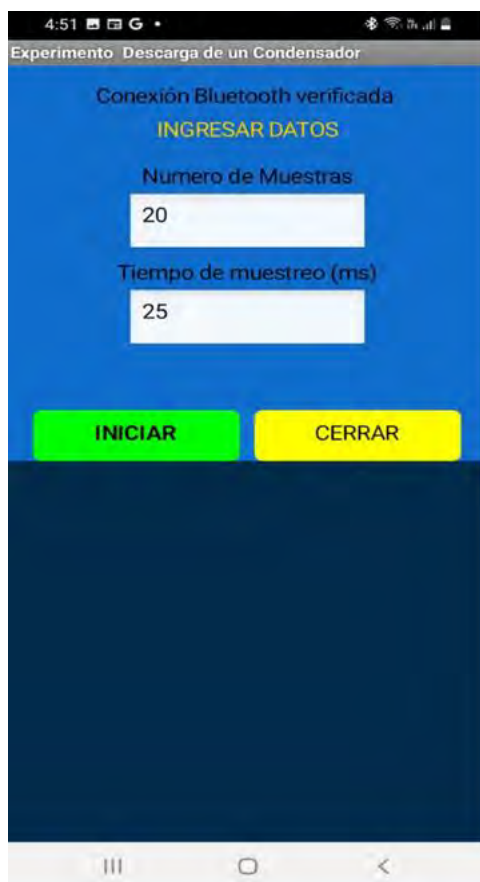
4.3.8.2. Aplicación para la Descarga de un Capacitor

La inclusión de un botón para la carga y otro para la descarga del capacitor en el diseño de la aplicación responde a que, para realizar la descarga, el capacitor debe estar previamente cargado con la máxima diferencia de potencial suministrada por la fuente. Además, con el objetivo de simplificar el diseño de la interfaz, se incorporan dos circuitos que se activan de manera independiente.

Por lo tanto, en la pantalla principal de la aplicación *App_PhysicsLab_F3.apk*, al presionar el botón *DESCARGA CAPACITOR*, se abre una nueva pantalla personalizada para el ingreso de datos. En esta pantalla, se muestra un mensaje que confirma si la conexión con la interfaz es correcta, como se observa en la figura 88. Esta pantalla es similar a la empleada para la carga del capacitor. El primer dato a ingresar es el número de muestras, con un máximo de 21. Posteriormente, se introduce el tiempo de muestreo, que corresponde al intervalo entre muestras.

Figura 88.

Pantalla para el Inicio del Experimento de Descarga de un Capacitor

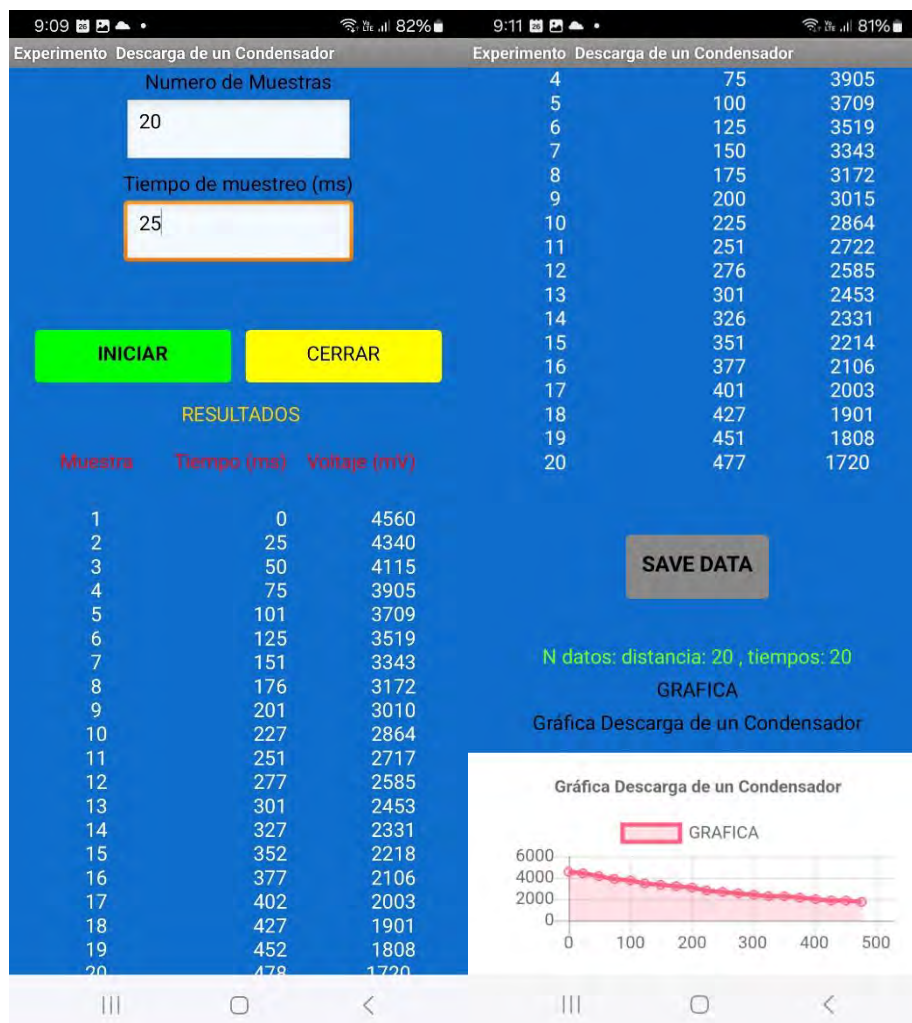


El procedimiento es similar para todos los experimentos: al pulsar el botón *INICIAR*, la aplicación envía los datos necesarios a la interfaz y da inicio a la ejecución del experimento.

Todas las aplicaciones desarrolladas en este trabajo realizan una validación de los datos ingresados. Una vez que la aplicación recibe correctamente la información enviada desde la interfaz, se activa una pantalla donde se muestran los resultados y se genera el gráfico correspondiente, como se ilustra en la figura 89. En la pantalla de resultados se presenta una tabla con tres columnas: la primera muestra los datos registrados, la segunda indica el tiempo y la tercera presenta la tensión del condensador. Además, el gráfico generado ilustra la relación entre la tensión y la corriente.

Figura 89.

Pantalla de Resultados y Gráfico para el Experimento de Descarga de un Capacitor



4.4. TRATAMIENTO DE LOS DATOS PARA EL CÁLCULO DE LOS PARÁMETROS DE LOS EXPERIMENTOS DEL PRESENTE TRABAJO.

Si se tiene una muestra de valores X e Y medidos sobre n individuos: (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) y se quiere estimar los valores de Y según el modelo $Y = AX + B$, donde A , B son inicialmente desconocidos. De entre todas las rectas, se debe encontrar la que mejor se ajuste a los datos observados, es decir, se busca aquellos valores para A y B que hagan mínimos los errores de estimación.

Para un valor x_i , el modelo estima un valor para Y igual a $y_i = A x_i + B$ y el valor observado en Y es y_i , con lo que en este caso el error de estimación sería $e_i = y_i - (A x_i + B)$. Entonces tomaremos como estimaciones para A y B , los valores que minimicen la suma de los errores al cuadrado dada por:

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - (A x_i + B))^2 \quad (17)$$

Este método de estimación se le llama *método de mínimos cuadrados*, la solución se obtiene derivando esta función con respecto a cada uno de los parámetros e igualando a cero. Las estimaciones para A y B dan como resultado:

$$A = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sum_{i=1}^n x_i^2 - n \bar{x}^2} \quad (18)$$

$$B = \bar{y} - A \bar{x} \quad (19)$$

A la recta $Y = AX + B$, se le llama recta de regresión lineal de Y sobre X .

Caída Libre.

En el experimento de caída libre, se aplica la regresión lineal a los datos y versus t^2 , donde t representa los tiempos registrados por la interfaz cuando el cuerpo impacta en la superficie con el sensor piezoeléctrico y y son las alturas introducidas por el usuario en la aplicación, medidas desde el punto de caída del cuerpo hasta la base del soporte donde se encuentra el sensor. La aceleración de la gravedad se estima a partir del parámetro A de la recta de regresión, y se calcula mediante la siguiente relación:

$$g = 2A \quad (20)$$

Péndulo Simple.

En el experimento del péndulo simple, se aplicó la regresión lineal a los datos T versus \sqrt{L} donde, T son los tiempos obtenidos por la Interface cada vez que el cuerpo pasa por la fotopuerta tras completar una oscilación, y L es la longitud de la cuerda introducida por el usuario en la aplicación. La aceleración de la gravedad se estima a partir del parámetro A de la recta de regresión y se calcula mediante la siguiente relación:

$$g = \frac{4\pi^2}{A^2} \quad (21)$$

Movimiento Rectilíneo Uniforme (M.R.U.).

En el Movimiento Rectilíneo Uniforme (M.R.U.), al aplicar el método de mínimos cuadrados a los datos $d=f(t)$, donde t representa los tiempos registrados por la interfaz cada vez que el móvil pasa por una fotopuerta, y d son las distancias ingresadas por el usuario en la aplicación, medidas desde el punto de partida del móvil hasta el final del carril, el parámetro A de la recta de regresión corresponde a la velocidad del móvil.

$$v = A \quad (22)$$

Ley de Ohm

En el caso de la Ley de Ohm, se aplica el método de mínimos cuadrados a los datos $V=f(I)$, donde I representa la corriente y V las caídas de potencial en la resistencia, ambos obtenidos a través de la interfaz cuando la fuerza electromotriz (fem) interna genera niveles de voltaje entre 0 y 5 voltios. El valor de la resistencia se determina al estimar el parámetro A de la recta de regresión, el cual está dado por:

$$R = A \quad (23)$$

De esta manera, A representa la pendiente de la recta en el gráfico de V contra I y está pendiente corresponde al valor experimental de la resistencia del resistor en prueba.

Carga y Descarga de un Capacitor

Para analizar la carga y descarga de un capacitor, es necesario primero linealizar la solución de las ecuaciones del circuito $V=f(t)$, donde V representa el potencial en el capacitor en distintos instantes de tiempo t , ambos parámetros obtenidos a través de la interfaz del sistema.

Las expresiones linealizadas se describen en general como $V'=g(t)$, aplicando el método de mínimos cuadrados para determinar el parámetro A . Este parámetro está directamente relacionado con las constantes de tiempo tanto de carga como de descarga del capacitor, y se puede utilizar para calcular dichas constantes de manera experimental.

$$\tau = \frac{1}{A} \quad (24)$$

De esta forma, el parámetro A se obtiene a partir de la pendiente de la recta de la expresión linealizada $V'=g(t)$, y esta pendiente está inversamente relacionada con la constante de tiempo del capacitor.

4.5. RESULTADOS OBTENIDOS

Los datos presentados en las siguientes tablas fueron registrados y posteriormente exportados por la aplicación a través de la web a una hoja de cálculo en Google Sheets. Utilizando las herramientas de esta aplicación, graficamos y realizamos el análisis de datos, empleando funciones de ajuste de curvas y de mínimos cuadrados para obtener los siguientes resultados.

Caída Libre.

Utilizando el montaje experimental ilustrado en la figura 69 (montaje mecánico para caída libre), se realizaron varias pruebas con el equipo, variando la altura desde 100 mm hasta 600 mm en incrementos de 100 mm. En cada prueba, se registró el tiempo que tarda el objeto en caer hasta el sensor piezoeléctrico, que sirve como nivel de referencia. En total, se midieron seis alturas y sus respectivos tiempos, como se muestra en la Tabla 3. Con la posibilidad de registrar más datos para alturas en el rango de 0 a 1 metro.

Tabla 3.

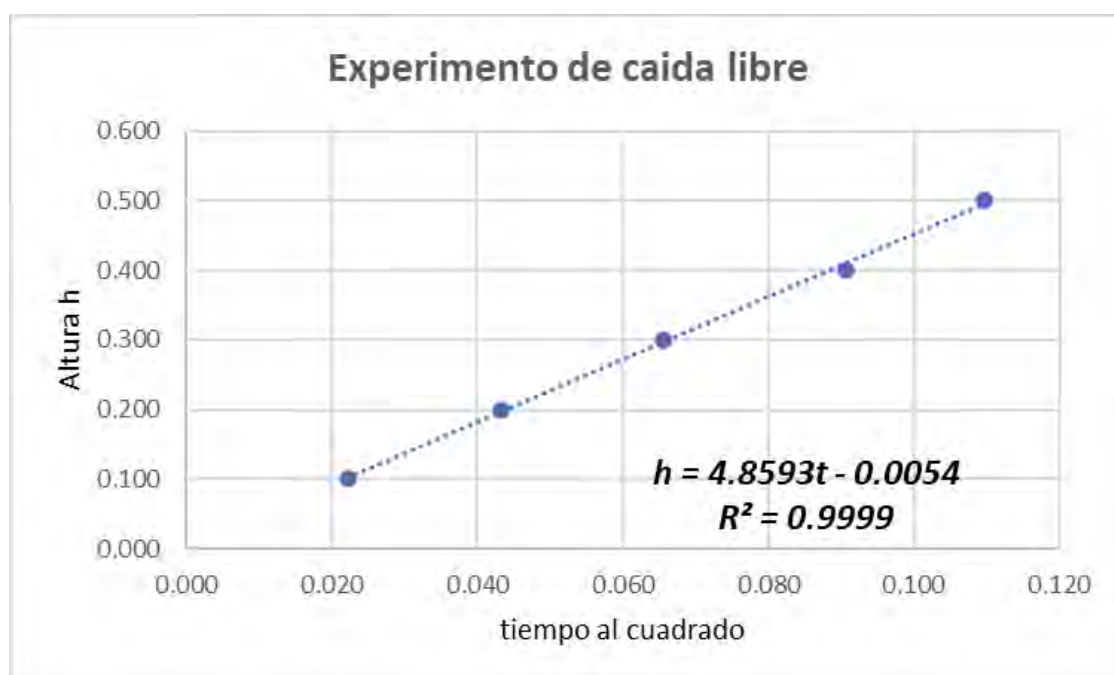
Datos del Experimento de Caída libre

	altura h (m)	tiempo sensor (s)	t ²
1	0.100	0.149	0.022
2	0.200	0.208	0.043
3	0.300	0.256	0.066
4	0.400	0.301	0.091
5	0.500	0.331	0.110
6	0.600	0.355	0.126

La figura 90, muestra la representación gráfica de los datos (h,t) , donde h representa la altura y t el tiempo que tarda el objeto en caer desde esa altura. Estos datos sugieren una relación entre ambas variables, la cual puede expresarse como $h = f(t)$, ajustándose a una curva de tipo cuadrática.

Figura 90.*Gráfico altura en función del tiempo*

La figura 91 muestra la gráfica linealizada junto con la ecuación característica obtenida mediante el método de mínimos cuadrados y la correlación de los datos.

Figura 91.*Gráfico altura en función del tiempo al cuadrado*

El valor de la pendiente es $A = 4.8593$ como se muestra en la Figura 91, de donde se obtiene un valor de la aceleración de la gravedad $g = 2A = (9.719 \pm 0.049) \text{ m/s}^2$. El valor del coeficiente de determinación obtenido fue de $R^2 = 0.9999$, lo que indica que la relación entre h y t^2 es lineal con muy buena aproximación.

Péndulo Simple.

Utilizando el montaje experimental mostrado en la figura 74 (montaje mecánico para péndulo simple), se realizaron pruebas para medir el período de 10 oscilaciones del péndulo con una longitud fija de la cuerda. Este procedimiento se repitió para cinco longitudes diferentes de la cuerda. A continuación, se presenta un resumen de los datos obtenidos en la tabla 4.

Tabla 4.

Datos del Experimento de Péndulo Simple.

N°	Longitud L (m)	\sqrt{L}	Periodo T(s)
1	0.7700000	0.8774964	1.7699850
2	0.7200000	0.8485281	1.7123248
3	0.6700000	0.8185353	1.6527624
4	0.6200000	0.7874008	1.5886104
5	0.5700000	0.7549834	1.5247568

La Figura 92 muestra una representación del promedio del período de 10 oscilaciones (T) en función de la longitud (L) de la cuerda del péndulo. Al igual que en el experimento anterior, se observa una relación no lineal (de tipo potencial) entre T y L .

Figura 92.

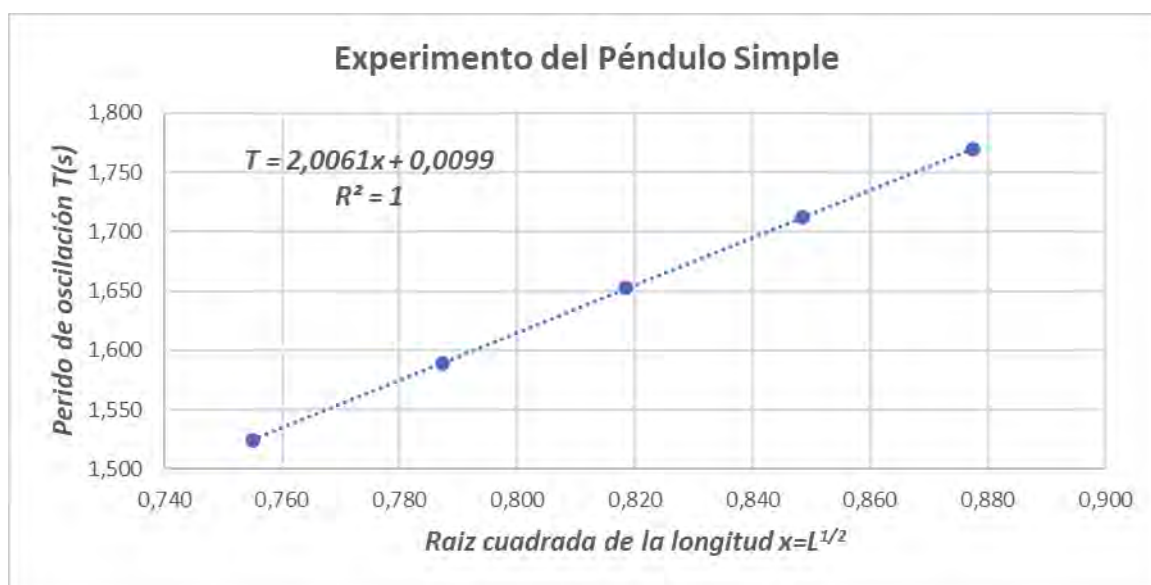
Gráfica del Periodo en Función de la Longitud del Péndulo



La figura 93 muestra la gráfica linealizada $T = f(\sqrt{L})$, junto con la ecuación característica obtenida mediante el método de mínimos cuadrados y la correlación de los datos. Para determinar la aceleración de la gravedad, utilizamos el coeficiente A y la ecuación 21.

Figura 93.

Gráfica del Periodo en Función de la Raíz Cuadrada de longitud del Péndulo.



El valor de la pendiente es $A = 2.0061$ como se muestra en la ecuación de la figura 96, lo que permite calcular el valor de la aceleración de la gravedad $g = 9.80967 \text{ m/s}^2$. El coeficiente de correlación lineal obtenido fue de $R = 1$, lo que indica que la relación entre T y \sqrt{L} es lineal con muy buena aproximación. En conclusión, el valor de la aceleración de la gravedad g obtenido por ambos métodos se encuentran dentro del 5% del valor teórico para Cusco, que es de 9.7729 m/s^2 (Castro, 2009).

Movimiento Rectilíneo Uniforme (MRU).

Utilizando el montaje de la Figura 78 (montaje para el Movimiento Rectilíneo Uniforme) con seis sensores, se realizaron varias pruebas con el equipo, variando las distancias entre sensores, teniendo en cuenta que la longitud del carril es de 120 cm. Se registró el tiempo que el móvil tardó en pasar por cada sensor. En el experimento, se analizó el movimiento del móvil a partir del momento en que pasó por el primer sensor. Los datos obtenidos se muestran en la tabla 5.

Tabla 5.

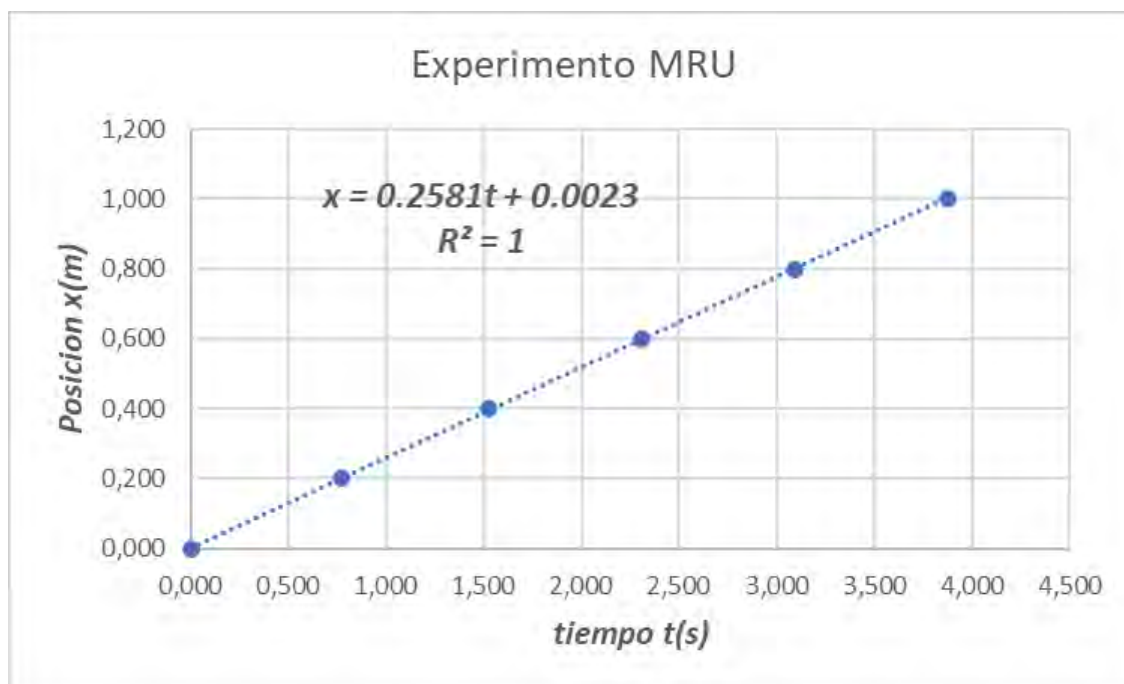
Datos del Experimento de MRU

Sensor	Tiempo $t(s)$	Distancia $x(m)$
1	0.000	0.000
2	0.770	0.200
3	1.525	0.400
4	2.311	0.600
5	3.091	0.800
6	3.875	1.000

La figura 94, muestra la gráfica lineal de $x = f(t)$, junto con la ecuación característica obtenida mediante el método de mínimos cuadrados y la correlación de los datos.

Figura 94.

Gráfico de la Posición en Función del Tiempo



El valor de la pendiente es $A = 0.2581$ como se muestra en la ecuación de la figura 94, lo que corresponde a la velocidad del móvil $v = A$. La correlación entre la distancia x y el tiempo t fue de $R = 1$, lo que demuestra que la relación entre x y t es lineal con muy buena aproximación. En conclusión, el movimiento corresponde a un Movimiento Rectilíneo Uniforme con una velocidad de 0.2581 m/s.

Ley de Ohm.

Utilizando el montaje de la figura 81 (montaje para el experimento de la Ley de Ohm), se realizaron varias pruebas con el equipo. Durante estas pruebas, se registraron los datos de la diferencia de potencial y la intensidad de corriente en un resistor de 47Ω . Los datos obtenidos se presentan en la tabla 6.

Tabla 6.*Datos del Experimento de la Ley de Ohm*

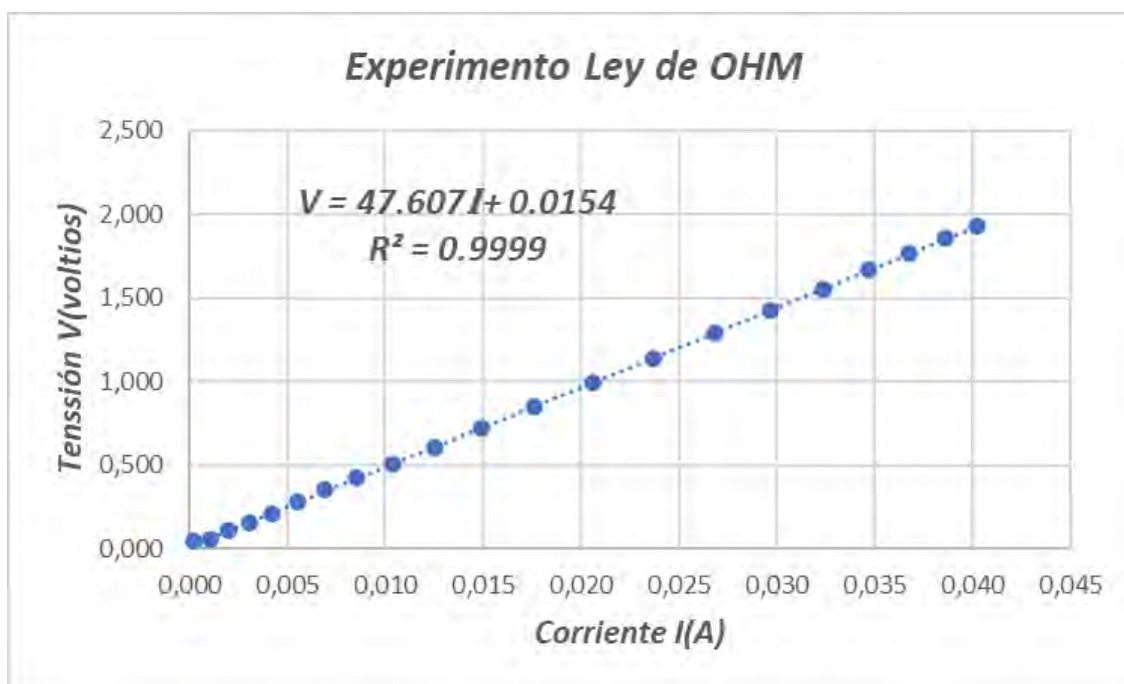
Dato	Corriente $I(A)$	Tensión $V(V)$
1	0.00025	0.05000
2	0.00110	0.06000
3	0.00204	0.11000
4	0.00308	0.16000
5	0.00423	0.21000
6	0.00552	0.28000
7	0.00696	0.35000
8	0.00858	0.43000
9	0.01042	0.51000
10	0.01252	0.61000
11	0.01492	0.72000
12	0.01762	0.85000
13	0.02060	0.99000
14	0.02372	1.14000
15	0.02682	1.29000
16	0.02971	1.43000
17	0.03237	1.55000
18	0.03468	1.67000
19	0.03676	1.77000
20	0.03858	1.86000
21	0.04019	1.93000

Con los datos obtenidos experimentalmente, se determinó el valor de la resistencia del resistor en prueba R mediante el método de mínimos cuadrados, $R = A$ de acuerdo a la ecuación 22.

La figura 95, muestra los resultados de la regresión lineal de $V=f(I)$.

Figura 95.

Grafica de la Tensión en Función de la Corriente.



El valor de la pendiente es de 47.607 como se muestra en la Figura 98, que corresponde al valor de la resistencia. El coeficiente de determinación entre la Tensión Eléctrica y la Intensidad de Corriente fue de $R^2 = 0.9999$, lo que indica una relación lineal con una buena precisión entre V e I . El valor obtenido para el resistor es 47.607Ω , lo que se compara favorablemente con el valor teórico de la resistencia, que es $(47 \pm 2.35)\Omega$.

Carga de un Capacitor.

Utilizando el montaje de la figura 84 (montaje para el experimento de carga del capacitor), se realizaron varias pruebas con el equipo. Se registraron los datos de la tensión eléctrica y el tiempo durante el proceso de carga de un capacitor de $220\mu\text{F}$ a través de un resistor de $1\text{k}\Omega$. Los datos obtenidos se muestran en la Tabla 7.

Tabla 7.

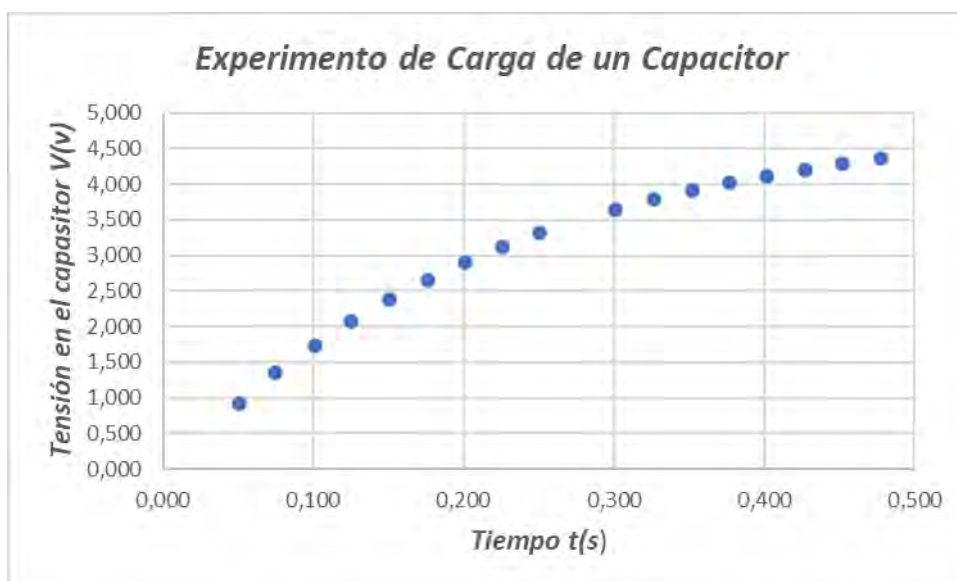
Datos del Experimento de Carga de un Capacitor

Dato	Tiempo $t(s)$	Tensión $V(v)$	$LN(1 - V/V_0)$
1	0.051	0.918	-0.20285085
2	0.075	1.353	-0.31553300
3	0.101	1.735	-0.42617815
4	0.125	2.082	-0.53853946
5	0.151	2.385	-0.64817381
6	0.176	2.658	-0.75843265
7	0.201	2.898	-0.86654864
8	0.226	3.118	-0.97710287
9	0.251	3.313	-1.08648611
10	0.301	3.641	-1.30268878
11	0.327	3.782	-1.41222774
12	0.352	3.905	-1.51868355
13	0.377	4.012	-1.62151049
14	0.402	4.115	-1.73160555
15	0.427	4.203	-1.83633851
16	0.452	4.281	-1.93933183
17	0.478	4.354	-2.04639369

La figura 96 muestra una representación de la tensión en el capacitor (en voltios, V) en función del tiempo t , considerando que la fuerza electromotriz (fem) del circuito es de $V_0 = 5$ voltios y la carga inicial del condensador es de 0 culombios.

Figura 96.

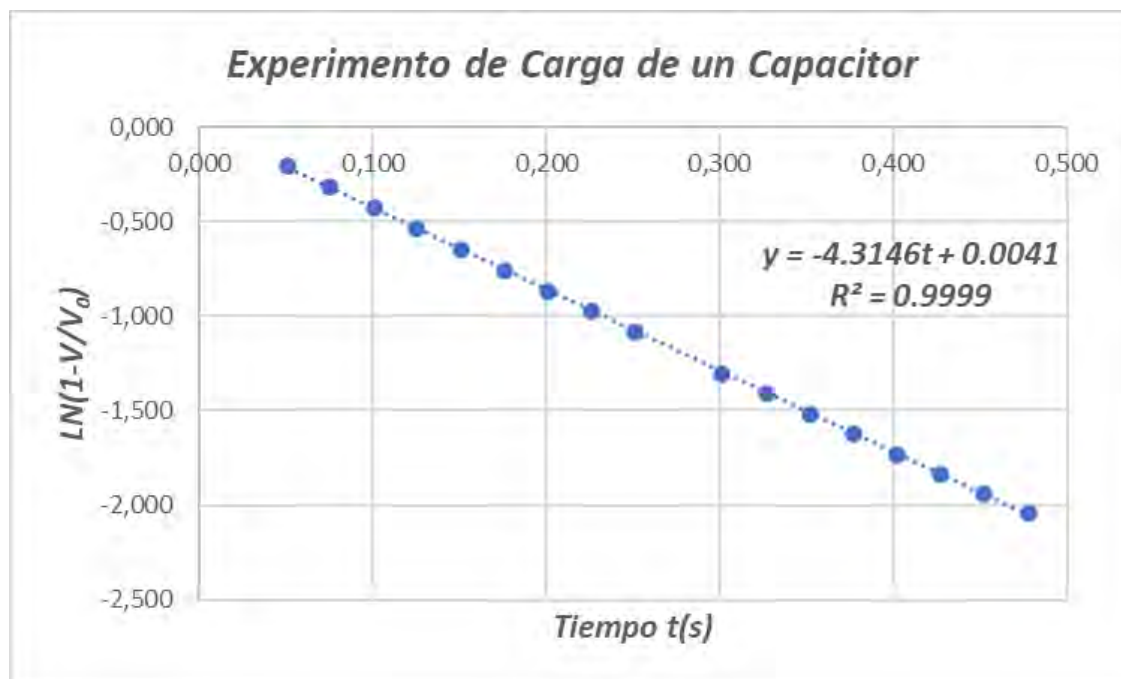
Grafica de la Tensión en Función del Tiempo para la Carga de un Capacitor.



La figura 97, muestra la gráfica linealizada, la ecuación característica obtenida por mínimos cuadrados y la correlación de los datos.

Figura 97.

Gráfica Linealizada de la Función Obtenida para la Carga de un Capacitor



El valor de la pendiente que corresponde al valor de la inversa de la constante de tiempo τ , es 4.3146 como se muestra en la Figura 100. El coeficiente de determinación entre la $\ln(1-V/V_0)$ y el tiempo es de $R^2 = 0.9999$, lo que indica que la relación es lineal con una buena aproximación. El valor obtenido para la constante de tiempo τ fue de 0.232s, en comparación con el valor teórico $\tau = RC = 0.220s$.

Descarga de un Capacitor.

Utilizando el montaje de la figura 87 (montaje para el experimento de descarga del capacitor), se realizaron varias pruebas con el equipo, registrando los datos de la tensión eléctrica y el tiempo en un capacitor de $470 \mu F$ con un resistor de descarga de $1 k\Omega$. Los datos obtenidos se muestran en la Tabla 8.

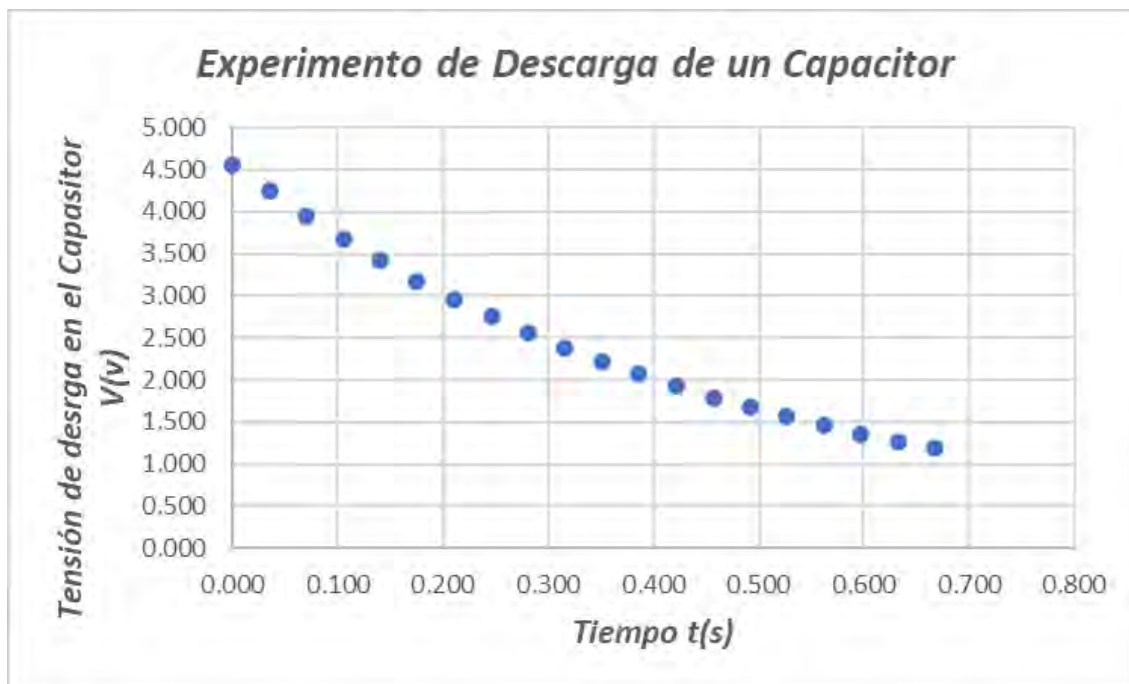
Tabla 8.*Datos del Experimento de Descarga de un Capacitor*

Dato	Tiempo $t(s)$	Tensión $V(v)$	$LN(V/V_0)$
1	0.035	4.252	-0.069933163
2	0.386	2.072	-0.788808299
3	0.070	3.949	-0.143860241
4	0.106	3.675	-0.215769491
5	0.632	1.265	-1.282250501
6	0.562	1.456	-1.141629674
7	0.210	2.956	-0.433485620
8	0.421	1.930	-0.859802621
9	0.492	1.676	-1.000912621
10	0.527	1.559	-1.073278033
11	0.281	2.561	-0.576924816
12	0.141	3.416	-0.288852348
13	0.597	1.358	-1.211309594
14	0.000	4.560	0.000000000
15	0.457	1.779	-0.941271215
16	0.667	1.182	-1.350114705
17	0.316	2.385	-0.648123499
18	0.352	2.223	-0.718464989
19	0.175	3.176	-0.361700080
20	0.246	2.751	-0.505358142

La figura 98 muestra una representación de la tensión en el capacitor en voltios (V) en función del tiempo (t). En este experimento, la descarga del capacitor se realiza a través del resistor de descarga. Inicialmente, el capacitor está cargado al máximo, con su diferencia de potencial al mismo nivel que la *fem* del circuito, es decir, 5 voltios. A medida que el tiempo transcurre, se observa cómo la tensión en el capacitor disminuye hasta llegar a cero, lo que indica que el capacitor se ha descargado por completo. Esta gráfica es fundamental para analizar la dinámica de descarga del capacitor y verificar la relación exponencial esperada en este proceso.

Figura 98.

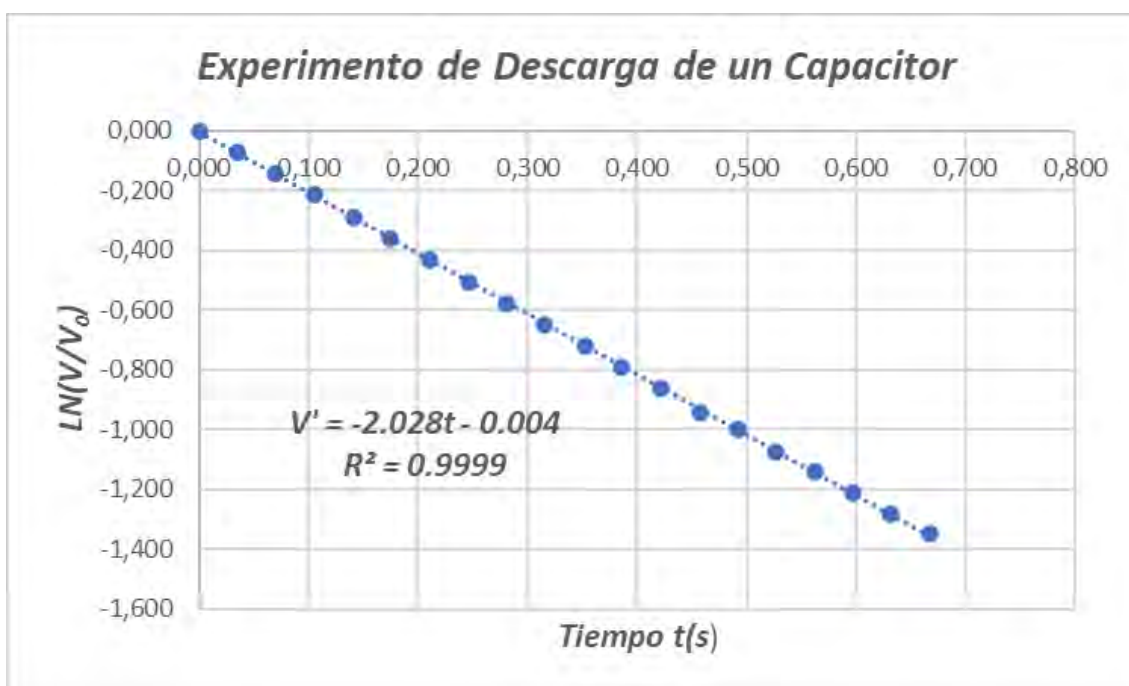
Gráfica de la Tensión en Función del Tiempo para la Descarga de un Capacitor



La figura 99, muestra la gráfica linealizada, la ecuación característica obtenida por mínimos cuadrados y la correlación de los datos.

Figura 99.

Gráfica Linealizada de la Función Obtenida para la Descarga de un Capacitor



El valor de la pendiente es $A = 2.028$ como se muestra en la ecuación de la figura 99, que corresponde al valor de la inversa de la constante de tiempo τ . El coeficiente de determinación entre la $\ln(V/V_0)$ y el tiempo es de $R^2 = 0.9999$, lo que indica que la relación es lineal con una buena aproximación. El valor de la constante de tiempo τ obtenido fue de $0.493s$. comparado con el valor teórico de la constante $\tau = RC = 0.470s$.

4.6. PRESENTACIÓN DE RESULTADOS

En las siguientes figuras se muestran el resumen todos los módulos que componen el prototipo presentado en el presente trabajo.

La figura 100 muestra imágenes del hardware diseñado y construido para los experimentos de Mecánica. En estas imágenes se pueden apreciar el actuador (electroimán), los sensores (fotopuerta y piezoeléctrico), la interfaz, el carrito para el MRU y el circuito anti rebote del sensor piezoeléctrico.

Figura 100.

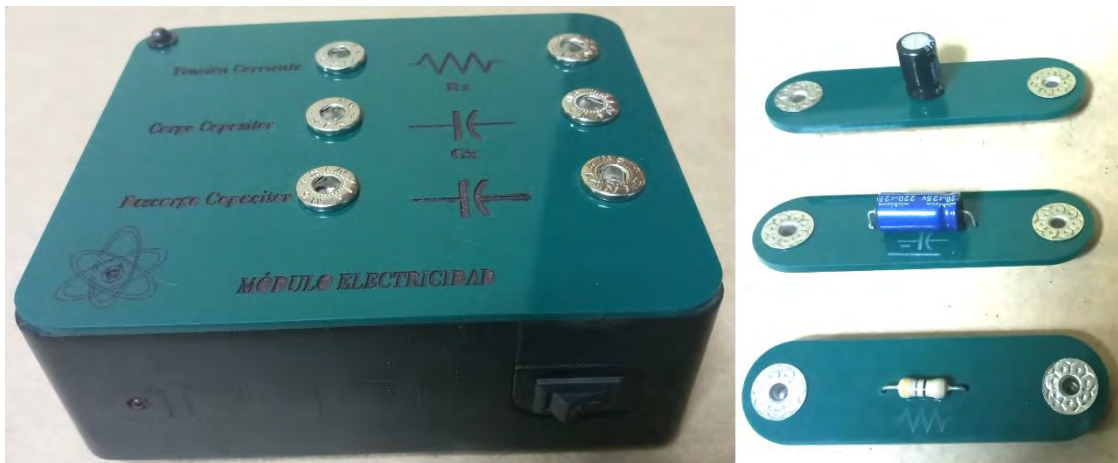
Hardware del Módulo de Mecánica, Incluye un Actuador, Interface, Sensores



La Figura 101 muestra imágenes del hardware diseñado y construido para los experimentos de Electricidad. En estas imágenes se puede apreciar la interfaz y los componentes utilizados para las pruebas.

Figura 101.

Hardware del módulo de electricidad, incluye la Interface, resistencia y capacitores.



La Figura 102 muestra imágenes de los elementos utilizados para sujetar y fijar sensores, actuadores y otros componentes del instrumental de laboratorio.

Figura 102.

Elementos mecánicos para montaje: varilla, soporte universal y abrazadera



La Figura 103 muestra imágenes de la aplicación implementada para el control de los experimentos mediante un smartphone. Se pueden apreciar la pantalla de inicio, la pantalla de configuración del Bluetooth, el menú de los experimentos implementados y la pantalla que muestra los datos registrados en la base de datos.

Figura 103.

Software desarrollado para un dispositivo Android



La figura 104 muestra imágenes de la aplicación implementada para el control del experimento de Caída Libre mediante un smartphone. Se pueden apreciar la pantalla de ingreso de datos, la pantalla con los datos medidos y el gráfico de los datos ajustados a una curva de tendencia, así como la pantalla que muestra los datos guardados en la base de datos y un botón para exportar a una hoja de cálculo de Google a través de la web.

Figura 104.

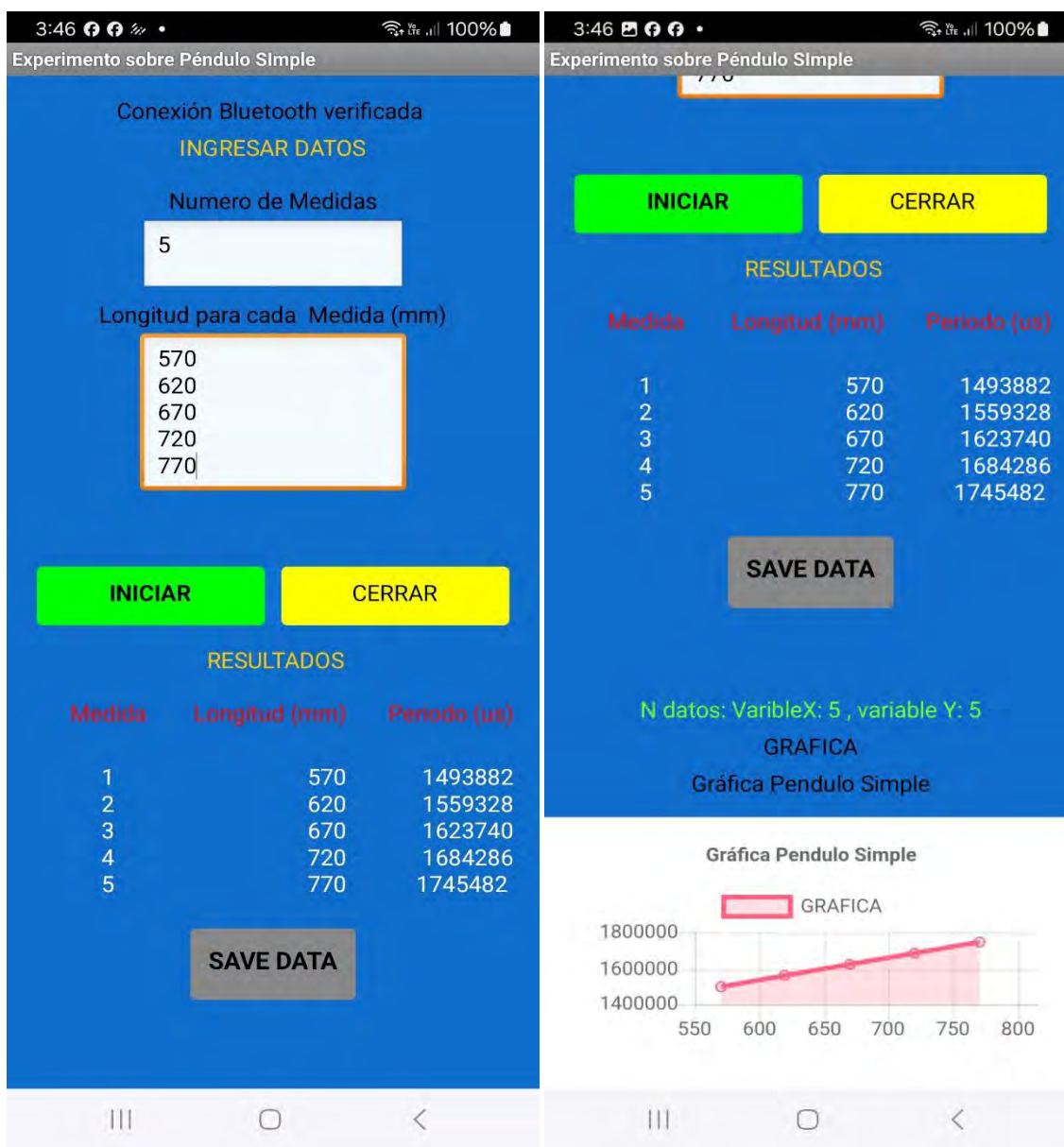
Resultados para el Experimento de Caída Libre



La Figura 105 muestra la aplicación para el experimento de Péndulo Simple. Incluye las pantallas para ingresar datos, visualizar los datos medidos y acceder a los datos guardados en la base de datos, junto con un botón para exportar a una hoja de cálculo de Google a través de la web.

Figura 105.

Resultados para el Experimento de Péndulo simple



La figura 106 muestra la aplicación para el experimento de M.R.U. Incluye la pantalla de ingreso de datos, los datos medidos con su gráfico ajustado a una curva de tendencia, y la pantalla con los datos guardados en la base de datos, además de un botón para exportar a una hoja de cálculo de Google.

Figura 106.

Resultados para el Experimento de Movimiento Rectilíneo Uniforme



La figura 107 muestra imágenes de la aplicación implementada para el control del experimento sobre la Ley de Ohm mediante un smartphone. La primera imagen presenta la pantalla de inicio, donde se observa que el experimento está diseñado para operar sin la necesidad de introducir datos iniciales. La segunda imagen muestra la pantalla con los datos medidos en tiempo real, junto con un gráfico que ilustra estos datos ajustados a una

La Figura 108 y la Figura 109 muestran imágenes de las aplicaciones implementadas para el control de los experimentos sobre la Carga y Descarga de un Capacitor mediante un smartphone. La primera imagen presenta la pantalla de ingreso de datos, donde se configuran los parámetros iniciales del experimento. La segunda imagen muestra la pantalla con los datos medidos en tiempo real, junto con un gráfico que ilustra estos datos ajustados a una curva de tendencia. Finalmente, se visualiza la pantalla donde se muestran los datos almacenados en la base de datos y se incluye una opción para exportar los resultados a una hoja de cálculo de Google a través de la web, facilitando la revisión y análisis posterior.

Figura 108.

Resultados para el Experimento de Carga de un capacitor

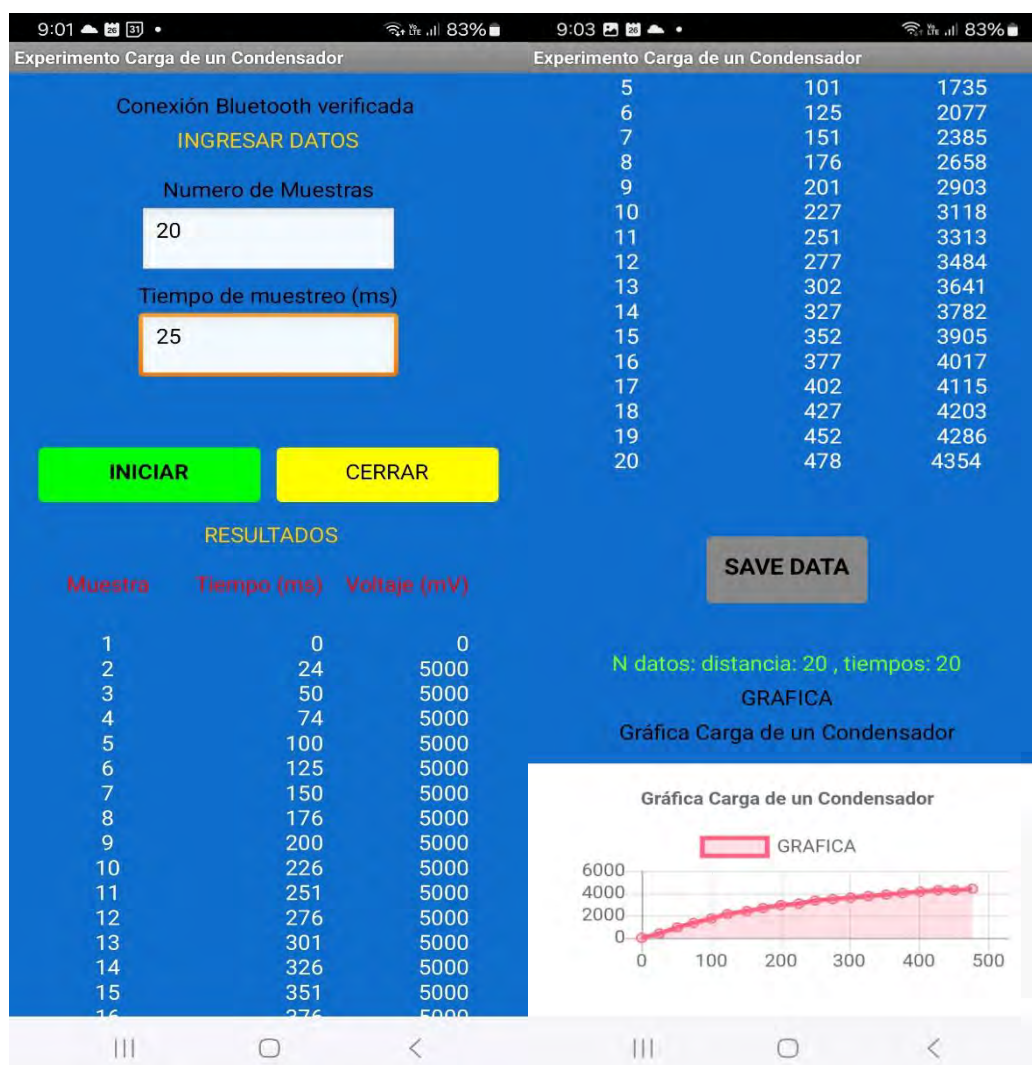
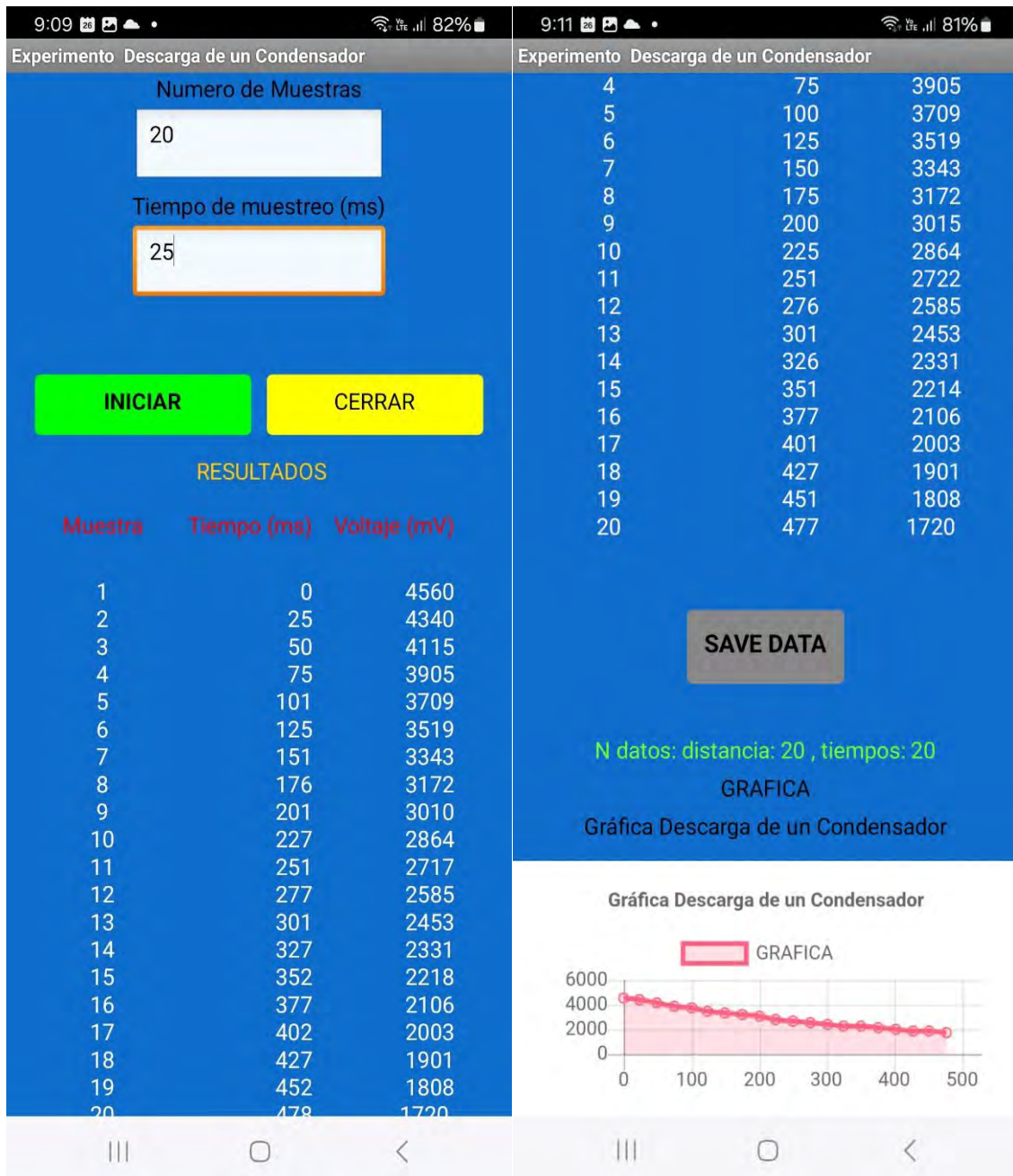


Figura 109.

Resultados para el Experimento de Descarga de un capacitor



CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES.

1. En esta tesis se ha demostrado la viabilidad de diseñar y construir módulos automatizados basados en Arduino para el desarrollo de experimentos de Física en las áreas de Mecánica y Electricidad. Aprovechando la tecnología actual como microcontroladores, sensores y conectividad inalámbrica, lo que permite su control desde dispositivos móviles con sistema operativo Android. Este logro cumple con el objetivo general del proyecto.
2. Se diseñó la arquitectura hardware del sistema y se construyó asegurando una integración con los periféricos necesarios para implementar experimentos de Mecánica y Electricidad.
3. Se diseño e implemento el software necesario para el control y operación de los módulos.
4. Se probaron y validaron los módulos verificando la consistencia de los resultados con las ecuaciones y los valores teóricos correspondientes a los experimentos de Física en Mecánica y Electricidad propuestos en este trabajo.

RECOMENDACIONES.

Luego de la finalización del proyecto, que ha incluido la construcción de módulos automatizados basados en la plataforma Arduino para el desarrollo de experimentos de Física General, (Mecánica y Electricidad) destinados a su uso en colegios y universidades; se hacen las siguientes recomendaciones:

- **Crear un manual de usuario:** Es esencial proporcionar instrucciones detalladas sobre cómo utilizar los módulos. Este manual debe estar dirigido tanto a estudiantes como a profesores, permitiéndoles aprovechar al máximo su funcionalidad.
- **Incorporar herramientas multimedia:** Se recomienda la creación de video tutoriales que ofrezcan una guía visual sobre la realización de experimentos con los módulos construidos. Estos videos deben proporcionar demostraciones paso a paso sobre cómo utilizar los módulos y llevar a cabo los experimentos descritos en este trabajo.
- **Elaborar guías de laboratorio:** Se deben desarrollar guías de laboratorio que implementen los módulos presentados, facilitando la realización de los experimentos propuestos.
- **Implementar nuevos sensores:** Se recomienda incorporar otros tipos de sensores que permitan desarrollar un mayor número de experimentos, ampliando así las posibilidades educativas del proyecto.
- **Desarrollar nuevos experimentos:** Se deben explorar y crear nuevos experimentos, tales como el M.R.U.V. (Movimiento Rectilíneo Uniformemente Variado), Conservación de Energía Mecánica, resonancia de un circuito RLC, entre otros.

- **Desarrollar software para otros sistemas operativos:** Es importante crear versiones del software para otros sistemas operativos además de Android, como iOS (dispositivos Apple), Windows y Linux (PCs), lo cual se plantea como trabajo futuro.

PROPUESTA PARA LA SOLUCIÓN DEL PROBLEMA

La falta de equipos en los laboratorios de física de la universidad se puede abordarse de manera efectiva utilizando Arduino, que ofrece una solución económica, versátil y educativa. Aquí algunas propuestas específicas:

Equipamiento Básico con Arduino

Descripción: Proveer a los laboratorios de kits de Arduino que incluyan placas, sensores, y módulos básicos.

- Beneficios: Bajo costo, gran versatilidad, facilidad de uso.
- Acciones:
 - Identificar las necesidades específicas de los laboratorios.
 - Comprar kits de inicio de Arduino y componentes adicionales.
 - Ofrecer talleres de capacitación para profesores y estudiantes.

Desarrollo de Experimentos Personalizados

Descripción: Diseñar experimentos y prácticas de laboratorio que utilicen sensores y módulos de Arduino para cubrir temas específicos de física.

- Beneficios: Personalización del aprendizaje, alineación con el currículo.
- Acciones:
 - Crear una base de datos de experimentos que se puedan realizar con Arduino.
 - Desarrollar manuales y guías de usuario detalladas.
 - Integrar estos experimentos en el plan de estudios de física.

Programa de Capacitación y Soporte Técnico

Descripción: Implementar un programa continuo de capacitación y soporte técnico para profesores y estudiantes.

- Beneficios: Mejora en el uso de la tecnología, mantenimiento y actualización constante.
- Acciones:
 - Ofrecer talleres periódicos y sesiones de formación en línea.
 - Crear un equipo de soporte técnico para resolver dudas y problemas.
 - Establecer una comunidad de práctica para compartir experiencias y soluciones.

Desarrollo de Proyectos Interdisciplinarios

Descripción: Fomentar proyectos que integren física, programación y electrónica mediante Arduino.

- Beneficios: Aprendizaje integrado, desarrollo de habilidades técnicas y de resolución de problemas.
- Acciones:
 - Promover competencias y ferias de ciencias con proyectos basados en Arduino.
 - Establecer colaboraciones con otras facultades como ingeniería y ciencias de la computación.
 - Ofrecer créditos académicos por la participación en estos proyectos.

Colaboraciones y Patrocinios

Descripción: Establecer alianzas con empresas y organizaciones para la donación de equipos y financiación de proyectos.

- Beneficios: Aumento de recursos, actualización de equipos.
- Acciones:
 - Buscar patrocinios de empresas tecnológicas y educativas.

- Solicitar donaciones de equipos obsoletos pero funcionales.
- Organizar eventos para presentar los proyectos realizados con Arduino y atraer patrocinadores.

Implementación de Laboratorios Virtuales y Simuladores

Descripción: Usar software de simulación junto con Arduino para ampliar las capacidades de los laboratorios.

- Beneficios: Acceso a más recursos, posibilidad de realizar experimentos sin restricciones físicas.
- Acciones:
 - Integrar software de simulación como Tinkercad o Proteus en el currículo.
 - Desarrollar laboratorios virtuales donde los estudiantes puedan practicar antes de trabajar con hardware real.
 - Ofrecer acceso remoto a equipos Arduino mediante plataformas en línea.

Evaluación y Mejora Continua

Descripción: Evaluar regularmente la efectividad de las soluciones implementadas y realizar ajustes según sea necesario.

- Beneficios: Adaptación constante a las necesidades y mejoras continuas.
- Acciones:
 - Recoger feedback de estudiantes y profesores sobre el uso de Arduino en los laboratorios.
 - Analizar los resultados académicos y la calidad de los experimentos.
 - Ajustar los programas y equipos según los resultados obtenidos.

Implementar estas propuestas puede transformar significativamente la experiencia educativa en los laboratorios de física, proporcionando a los estudiantes herramientas modernas y prácticas que fomenten el aprendizaje activo y la innovación.

COSTOS DE IMPLEMENTACIÓN DE LA PROPUESTA

La siguiente tabla muestra el presupuesto utilizado en materiales y mano de obra externa, para la construcción de cuatro interfaces, seis sensores fotopuerta, un sensor piezoeléctrico, tres actuadores y equipo de laboratorio (soportes universales, varillas de acople, abrazaderas, etc.), para el armado de los experimentos mostrados en el presente trabajo.

N°	DESCRIPCIÓN	COSTO UNITARIO	CANTIDAD	COSTO
1	Arduino Uno R3	S/.100.00	1	s/. 100.00
2	Arduino Nano	S/.80.00	4	s/. 320.00
3	Sensores y transductores	S/.150.00	1	s/.150.00
4	Componentes electrónicos	S/.200.00	1	s/. 200.00
5	Conectores de diferentes tipos	S/.200.00	1	s/. 200.00
6	Electroimanes	S/.30.00	3	s/. 90.00
7	Construcción de soporte tipo nuez	S/.30.00	6	s/. 180.00
8	Construcción de interfaces	S/.150.00	4	s/. 600.00
9	Construcción de sensores	S/.50.00	10	s/. 500.00
10	Construcción de actuadores	S/.30.00	3	s/. 90.00
11	Construcción de placas electrónicas	S/.50.00	5	s/. 250.00
12	Construcción de cables	S/.10.00	20	s/. 200.00
13	Soportes universales de laboratorio	S/.150.00	2	s/. 300.00
14	Varios	S/.1,000.00	1	s/. 1,000.00
TOTAL				s/.4180.00

BENEFICIOS QUE APORTA LA PROPUESTA

Implementar instrumentos de laboratorio de enseñanza de Física General con Arduino ofrece una serie de beneficios significativos que pueden mejorar la calidad de la educación y la experiencia de los estudiantes. Aquí algunos de los beneficios más destacados:

Costos Reducidos

Beneficio: Los equipos de laboratorio tradicionales pueden ser muy costosos. Arduino, por otro lado, es una plataforma de hardware de bajo costo.

- Impacto: Permite a las universidades equipar sus laboratorios con múltiples kits de Arduino sin un gasto significativo, facilitando el acceso a equipos para más estudiantes.

Versatilidad y Adaptabilidad

Beneficio: Arduino es extremadamente versátil y puede utilizarse para una amplia gama de experimentos y proyectos.

- Impacto: Los mismos kits pueden ser reutilizados en diversas prácticas de física, desde mecánica hasta electricidad y magnetismo, lo que maximiza el valor de la inversión inicial.

Promoción del Aprendizaje Práctico y Activo

Beneficio: Arduino fomenta un enfoque práctico y activo del aprendizaje.

- Impacto: Los estudiantes pueden diseñar y construir sus propios experimentos, lo que refuerza la comprensión teórica a través de la aplicación práctica.

Fomento de Habilidades Técnicas y de Programación

Beneficio: Trabajar con Arduino incluye la programación y el diseño de circuitos.

- Impacto: Los estudiantes desarrollan habilidades valiosas en programación, electrónica y resolución de problemas, que son altamente demandadas en el mercado laboral actual.

Incentivo a la Creatividad y la Innovación

Beneficio: Arduino permite a los estudiantes experimentar y crear soluciones innovadoras.

- Impacto: Fomenta un entorno educativo donde la creatividad es valorada y los estudiantes son alentados a pensar de manera innovadora para resolver problemas científicos.

Fácil Integración y Uso

Beneficio: Arduino es conocido por su facilidad de uso y abundante documentación y recursos en línea.

- Impacto: Los profesores pueden integrar fácilmente Arduino en sus planes de estudio y encontrar apoyo en comunidades en línea, lo que facilita la implementación y el uso continuo.

Acceso a una Gran Comunidad y Recursos

Beneficio: Arduino tiene una gran comunidad global que comparte proyectos, códigos y soluciones.

- Impacto: Los estudiantes y profesores tienen acceso a una vasta cantidad de recursos y pueden colaborar con otros usuarios en todo el mundo para resolver problemas y mejorar sus proyectos.

Facilidad para la Colaboración Interdisciplinaria

Beneficio: Arduino puede ser utilizado en una variedad de disciplinas, no solo en física.

- Impacto: Fomenta proyectos interdisciplinarios que integran conocimientos de física, informática, ingeniería y otras áreas, enriqueciendo la educación de los estudiantes.

Adaptabilidad a Tecnologías Emergentes

Beneficio: Arduino es compatible con una amplia gama de sensores y tecnologías emergentes.

- Impacto: Los laboratorios pueden mantenerse actualizados con las últimas tendencias tecnológicas, como el Internet de las Cosas (IoT), lo que prepara a los estudiantes para el futuro tecnológico.

Facilitación de la Evaluación Continua y el Feedback

Beneficio: Arduino permite la recolección y análisis de datos en tiempo real.

- Impacto: Los estudiantes pueden obtener feedback inmediato sobre sus experimentos y ajustarlos en tiempo real, lo que mejora el proceso de aprendizaje y comprensión.

La implementación de instrumentos de laboratorio de enseñanza de física general con Arduino no solo hace que la educación sea más accesible y asequible, sino que también mejora la calidad de la enseñanza al hacerla más interactiva, práctica y alineada con las necesidades del mercado laboral actual. Además, fomenta un entorno educativo que valora la creatividad, la innovación y la colaboración interdisciplinaria.

REFERENCIAS BIBLIOGRÁFICAS

- 330ohms, E. (noviembre 2023). *¿Qué son los actuadores electrónicos?*
<https://www.330ohms.com/blogs/blog/85020868-que-son-los-actuadores-electronicos>
- 5hertz, E. (Junio 2023). *5Hertz Electronica*. Obtenido de
<http://5hertz.com/tutoriales/?p=571#1.1QAC>
- Android, E. (Junio 2023). *Android Studio*. Obtenido de
<https://developer.android.com/studio/intro/index.html?hl=es-419>
- Angulo, U., Romero, Y. & Angulo, M.(2000). *Microcontroladores PIC Diseño practico de aplicaciones* (2da ed.) Mc Graw Hill
- Arduino, E. (Junio 2023). *Arduino*. Obtenido de
<https://www.arduino.cc/en/Guide/Introduction>
- Arion, D. N. (2000). *Case-study experiments in the introductory physics curriculum*.
 The Physics Teacher 38, 373.
- BasicX, E. (19 de Noviembre de 2023). *BasicX Home*. Obtenido de
<http://www.basicx.com/>
- Baker, R., & St. Laurent, A. (2011). *Android Essentials*. Apress.
- Bentley, J. P. (2005). *Principles of Measurement Systems*. Pearson Education.
- ChartJS, E. (2023). *Get Started*. Obtenido de <http://www.chartjs.org/docs/latest/>
- Centurión Wu Acuy. (2020). *Capacidades del empleo de Arduino sobre la operatividad del laboratorio del curso de Física general* [Tesis de grado. Universidad Privada de la Selva Peruana]. <http://repositorio.ups.edu.pe/handle/UPS/106>

Cox, A. N. (2000). *Allen's astrophysical quantities*. Springer: SAO/NASA ADS Astronomy Abstract Service.

MDN Web Docs. (2023). *JavaScript*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Drake, S. (1989). *Galileo at Work: His Scientific Biography*. University of Chicago Press.

Halliday, D., Resnick, R., & Walker, J. (2017). *Fundamentals of physics* (10.^a ed.). Wiley.

HandyBoard, E. (19 de Noviembre de 2023). *The Handy Board*. Obtenido de <http://handyboard.com/hb/Inventor>, W. (2015).

Higuera S., Guzmán R., J. & Rojas G. (2019). *Implementando las metodologías steam y abp en la enseñanza de la física mediante Arduino*. Memorias De Congresos UTP, 133-137. <https://revistas.utp.ac.pa/index.php/memoutp/article/view/2304>

Fraden, J. (2016). *Handbook of Modern Sensors: Physics, Designs, and Applications* (5th ed.). Springer.

Giancoli, D. C. (2008). *Physics: Principles with Applications* (7th ed.). Pearson.

Gibson, C. (2020). *Smartphones and Beyond: Lessons from the Remarkable Rise and Fall of Symbian*. Cambridge University Press.

Goldstein, H., Poole, C., & Safko, J. (2002). *Classical Mechanics* (3rd ed.). Addison-Wesley.

Kurniawan, A. (2018). *Arduino Programming with .NET and Sketch*. Apress.

- Marion, J. B., & Thornton, S. T. (2004). *Classical Dynamics of Particles and Systems* (5th ed.). Brooks Cole.
- McDermott, L. S. (November 2000). Preparing teachers to teach physics and physical science by inquiry. *Physics Education, Volume 35, Number 6*, 411-420.
- Menéndez, J. R. (1992). Laboratorio Automatizado de Física, Departamento de Física del Instituto Superior Politécnico “José A. Echavarría” de la Habana, Cuba.
- MIT App Inventor. (n.d.). *About Us*. Retrieved from <https://appinventor.mit.edu/about>
- Montgomery, DC (2004). *Diseño y análisis de experimento*. Editorial Limusa
- Morley, D., & Parker, C. (2017). *Understanding Computers: Today and Tomorrow* (16th ed.). Cengage Learning.
- Murillo-Soto, L. D. (enero-marzo 2023). *Automatización de pequeña escala con Open Hardware*. Tecnología en Marcha. Vol. 28, N° 1, 15-23.
- National Instruments (diciembre 2023). *¿Qué es Adquisición de Datos?* <https://www.ni.com/es/shop/data-acquisition.html>
- Nilsson, J. W., & Riedel, S. A. (2020). *Electric Circuits* (11th ed.). Pearson.
- Parallax, E. (noviembre de 2023). *Parallax. The Basic Stam*. Obtenido de <https://www.parallax.com/catalog/microcontrollers/basic-stamp>.
- Phidgets, E. (noviembre 2023). *Products for USB Sensing and Control*. Obtenido de <http://www.phidgets.com/>

Reyes, L. O. (2006). *Diseño de un sistema de bajo costo para la automatización de medidas en experimentos de física. "LabGIMC"*. Revista de la Sociedad Colombiana de Física, ISSN-e 0120-2650, Vol. 38, N°. 2, 850-853.

Sánchez C. y Cruz I. (2018). *Diseño e implementación de un Módulo Didáctico de Control de Temperatura para la Escuela Profesional de Ingeniería Electrónica en a UNPRG* [Tesis de grado. Universidad Nacional Pedro Ruiz Gallo - Lambayeque]. <https://repositorio.unprg.edu.pe/handle/20.500.12893/2092>

Salvador G. (2016). *Experimentos de física, De bajo costo usando TIC's*. UNSAN Buenos Aires. <https://www.researchgate.net/profile/Salvador-Gil-2>

Schnick, J. W. (2007). *Calculus-Based Physics I*. Saint Anselm College.

Serway, R. A., & Jewett, J. W. (2018). *Physics for Scientists and Engineers with Modern Physics* (10th ed.). Cengage Learning.

Tanenbaum, A. S., & Bos, H. (2014). *Modern Operating Systems* (4th ed.). Pearson.

Tipler, P. A., & Mosca, G. (2020). *Physics for scientists and engineers* (7.^a ed.). W.H. Freeman and Company

Tippens, P. E. (2011). *Física, conceptos y aplicaciones Séptima edición revisada*. Lima: McGraw-Hill/Interamericana Editores, S.A.

UNESCO. (1999). *Declaración sobre la Ciencia y el Uso del Saber Científico*.

Conferencia General, 30th, https://unesdoc.unesco.org/ark:/48223/pf0000116994_spa

UNESCO. (2023). <https://www.unesco.org/es/articles/la-unesco-hace-un-llamado-tomar-acciones-en-el-sector-educativo-tras-los-bajos-resultados-de-america>

Urrutia, R. (2020) *Diseño y Aplicación de Módulos Didácticos en la Educación Moderna*. Editorial Educa.

Watts, M. (1991). *The science of problem solving : a practical guide for science teachers*. London: Cassell.

Young, H. F. (2009). *Física Universitaria*. Mexico: Pearson Educacion.

ANEXOS

ANEXO A. MATRIZ DE CONSISTENCIA

Título: Diseño y construcción de módulos automatizados, basados en Arduino para el desarrollo de experimentos de Física en Mecánica y Electricidad

Autor: Yuri Morales García

PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLES	METODOLOGÍA
<p><u>General</u> ¿Será posible diseñar y construir módulos automatizados basados en Arduino para el desarrollo de experimentos de Física en Mecánica y Electricidad?</p> <p><u>Específicos</u></p> <ol style="list-style-type: none"> ¿Será posible diseñar la arquitectura del hardware del sistema y construirlo, asegurando la integración con los periféricos necesarios para implementar experimentos de Mecánica y Electricidad? ¿Se podrá diseñar e implementar el software necesario para el control y operación de los módulos? ¿Se podrá Probar el sistema hardware-software y verificar su funcionalidad con los experimentos de Física propuestos en este trabajo? 	<p><u>General</u> Diseñar y construir módulos automatizados, basados en Arduino, para el desarrollo de experimentos de Física en Mecánica y Electricidad.</p> <p><u>Específicos</u></p> <ol style="list-style-type: none"> Diseñar la arquitectura del hardware del sistema y construirlo, asegurando la integración con los periféricos necesarios para implementar experimentos de Mecánica y Electricidad. Diseñar e implementar el software necesario para el control y operación de los módulos. Probar el sistema hardware-software y verificar su funcionalidad con los experimentos de Física en Mecánica y Electricidad propuestos en este trabajo. 	<p><u>General</u> Utilizando las tarjetas electrónicas Arduino se diseñarán y construirán módulos automatizados para el desarrollo de experimentos de Física en Mecánica y Electricidad.</p> <p><u>Específicos</u></p> <ol style="list-style-type: none"> El diseño y la construcción del hardware del sistema integrara eficientemente los periféricos y demás accesorios necesarios para la implementación óptima de experimentos de Física en Mecánica y Electricidad. El diseño e implementación del software controlara y operara eficientemente el funcionamiento del hardware de los módulos. El desarrollo de los experimentos propuestos de Física en Mecánica y Electricidad, utilizando el sistema hardware - software, comprobara y validara su funcionalidad. 	<p>Por las características del presente trabajo no es pertinente establecer variables</p>	<p>Este trabajo es una investigación aplicada. Elegimos este enfoque porque el sistema que desarrollamos y proponemos requiere de diseño, construcción y programación de hardware y software, con el fin de integrarse en el desarrollo de experimentos de enseñanza de Física General, tanto a nivel universitario como secundario. Además, buscamos que los resultados obtenidos sean comparables con los de equipos comerciales que ofrecen funcionalidades similares a las que aspiramos lograr con este proyecto.</p>

ANEXO B. SOFTWARE DE DISEÑO ELECTRÓNICO

Para el diseño de circuitos electrónicos y PCBs se utilizó el software EasyEDA Pro que es una herramienta avanzada para la automatización del diseño electrónico (EDA) que proporciona varias características útiles para la captura de esquemas, diseño de PCB, simulación y colaboración. A continuación, se mencionan algunas de sus características más destacadas:

- Captura de Esquemas
- Diseño de PCB
- Biblioteca de Componentes
- Simulación
- Basado en la Nube
- Multiplataforma
- Interfaz de Usuario Avanzada:



ANEXO C. COTIZACIÓN DE EQUIPOS DE LABORATORIO DE LA EMPRESA

PASCO SCIENTIFIC

Lima, 11 de setiembre del 2023

Cotización No. 129-2023-AGBV/OPERON LAB S.A.C.



Asunto : COTIZACION EQUIPOS DE OPTICA
 Sres : UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
 Atencion : Sr. Jose Olintho de la Torre Cruz
 Email : olintho.delatorre@unsaac.edu.pe

Presente.-

Es muy grato dirigirme a Usted, para saludarlo y a la vez atender su requerimiento de:

ITEM	DESCRIPCION	CANT.	CODIGO	MARCA	IMAGEN REFERENCIAL	P. UNIT.	SUB TOTAL
1	<p>SISTEMA DE CAÍDA LIBRE El sistema de caída libre se puede utilizar para dejar caer casi cualquier objeto pequeño conectando una pequeña arandela de acero con una pequeña almohadilla adhesiva (ambas están incluidas en el sistema). Usando un interruptor eléctrico, el tiempo se inicia automáticamente, justo cuando se deja caer el objeto. Y la plataforma de tiempo de vuelo detiene el tiempo cuando el objeto lo golpea. Incluye: 1x Caja de caída 1x Cable de control 1x Caja de control 1x Adaptador de CA 1x Almohadilla receptora de tiempo de vuelo 1x Interruptor de temporizador 10x Arandelas de liberación 50x Liberar etiquetas para fijar arandelas al objeto 1x Bola de nylon pequeña 1x Bola de plástico grande 1x Pelota de golf 1x Pelota de golf hueca 1x Bola de acero de 1" 1x Bola de acero de 5/8"</p>	1	ME-9889	PASCO		S/4,218.00	S/4,218.00
2	<p>CONTROLADOR DE ONDA Este controlador rediseñado ofrece mejoras en la fuerza motriz y la amplitud, así como una guía de cuerdas incorporada para que las piezas pequeñas no se pierdan. Es ideal para vibrar ondas en una cuerda, conducir placas Chladni para mostrar los modos de vibración o para demostrar vibraciones de resonancia. Utilice un generador de funciones (no incluido) para alimentar este controlador de ondas. Incluye: 1x Controlador de onda con abrazadera de varilla incorporada y soporte de cuerda 1x Cadena de onda de muestra</p>	1	WA-9855	PASCO		S/1,195.10	S/1,195.10
3	<p>ABRAZADERA DE MESA UNIVERSAL Fije esta abrazadera universal de mesa a mesas o estantes de hasta 6,0 cm (2 3/8 pulgadas) de grosor. También se puede montar en un soporte de anillo. Monte varillas en la abrazadera vertical u horizontalmente. Las varillas se sujetan de forma segura mediante contactos estables de tres puntos. Use varillas de laboratorio sin roscar estándar, de 9,5 mm (3/8 de pulgada) a 12,7 mm (1/2 pulgada), vertical u horizontalmente. Use varilla de laboratorio roscada 1/2-13 verticalmente.</p>	1	ME-9376B	PASCO		S/717.06	S/717.06
4	<p>SUPER POLEA CON ABRAZADERA Mejora tu tabla de fuerzas y experimentos de plano inclinado. Con su abrazadera integral, la Super Polea facilita la configuración y la alineación. La altura de la polea es totalmente ajustable, lo que le permite rozar la parte superior de una mesa de fuerza para obtener lecturas sin paralelaje, al tiempo que mantiene la fuerza paralela a la pista en un plano inclinado. Se adapta a mesas de hasta 2,0 cm (13/16 pulg.) de grosor.</p>	1	ME-9448B	PASCO		S/302.29	S/302.29
5	<p>JUEGO DE MASAS Y GANCHOS Este robusto conjunto de masa enganchada está hecho de hierro fundido y recubierto con esmalte. Incluye: 1x 1000g Masa 1x 500g Masa 2x 200g Masa 1x 100g Masa 1x 50g Masa 2x 20g Masa 1x 10g Masa 1x Soporte de masa moldeado</p>	1	SE-8759	PASCO		S/506.16	S/506.16
6	<p>HILO NEGRO (PAQUETE DE 3) Hilo de nylon negro Incluye tres carretes de hilo de nylon negro</p>	1	ME-9875	PASCO		S/112.48	S/112.48
7	<p>VASO DE PRECIPITADOS, 1000 ML (PAQUETE DE 6) Estos vasos de precipitados de 1000 ml son borosilicatos, cristalería de grado estudiantil para su uso en el laboratorio de física. Paquete de 6. La cristalería de borosilicato de alta calidad y de grado estudiantil resistirá años de uso en laboratorio.</p>	1	SE-7288	PASCO		S/604.58	S/604.58

ANEXO D. MANUAL DE USUARIO

INTRODUCCIÓN

Los módulos desarrollados en este trabajo están programados exclusivamente para realizar los experimentos de Mecánica y Electricidad propuestos en este estudio. No obstante, existe la posibilidad de implementar otros experimentos mediante la modificación del código de la tarjeta Arduino, así como de la aplicación instalada en el smartphone.

La aplicación implementada en App Inventor es compatible únicamente con teléfonos inteligentes que utilizan el sistema operativo Android, comprobada desde la versión 8.0 hasta las versiones más recientes.

PRECAUCIONES DE SEGURIDAD

Para el uso correcto de los módulos en los experimentos de Mecánica y Electricidad, se recomienda primero revisar la información presentada en este trabajo, especialmente lo relacionado con el desarrollo de los experimentos de prueba.

- En el caso de la interfaz para los experimentos de Mecánica y Electricidad, utilice únicamente la fuente de voltaje adaptada específicamente para estos circuitos.
- Para el módulo destinado a los experimentos de Electricidad, no conecte componentes distintos a los indicados en la serigrafía de la placa de conexiones diseñada para los componentes de prueba.

ESPECIFICACIONES

ESPECIFICACIONES DE LA INTERFACE PARA EXPERIMENTOS DE MECÁNICA

- Conectores para 6 entradas digitales compatibles con sensores de nivel TTL.
- Conectores para 2 salidas digitales destinadas a actuadores, con un nivel de tensión de 12 V.
- Indicadores ópticos para mostrar la actividad de los actuadores.
- Pulsador de reinicio de la interface
- Resolución del temporizador en *ms* y μs

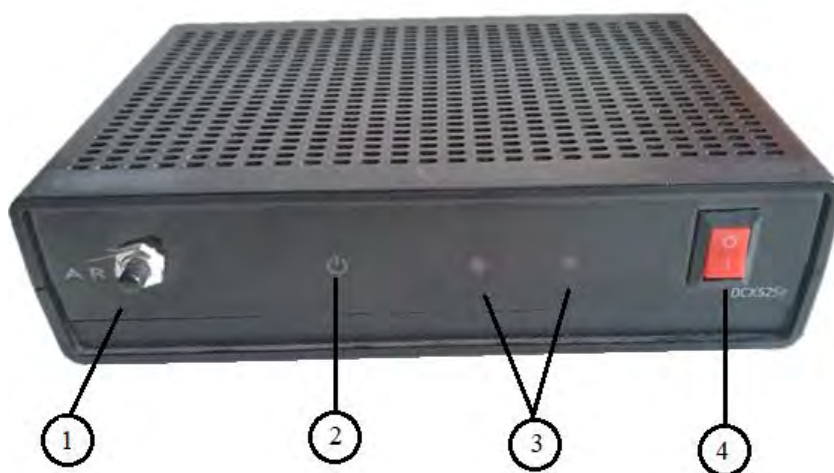
ESPECIFICACIONES DE LA INTERFACE PARA EXPERIMENTOS DE ELECTRICIDAD

- Conectores para componentes de prueba, Resistores y Capacitores
- Indicadores ópticos para mostrar la actividad de la interface
- Resolución del conversor análogo a digital de 10 bit

CONTROLES, INDICADORES Y CONECTORES

INTERFACE PARA EXPERIMENTOS DE MECÁNICA

CONTROLES E INDICADORES



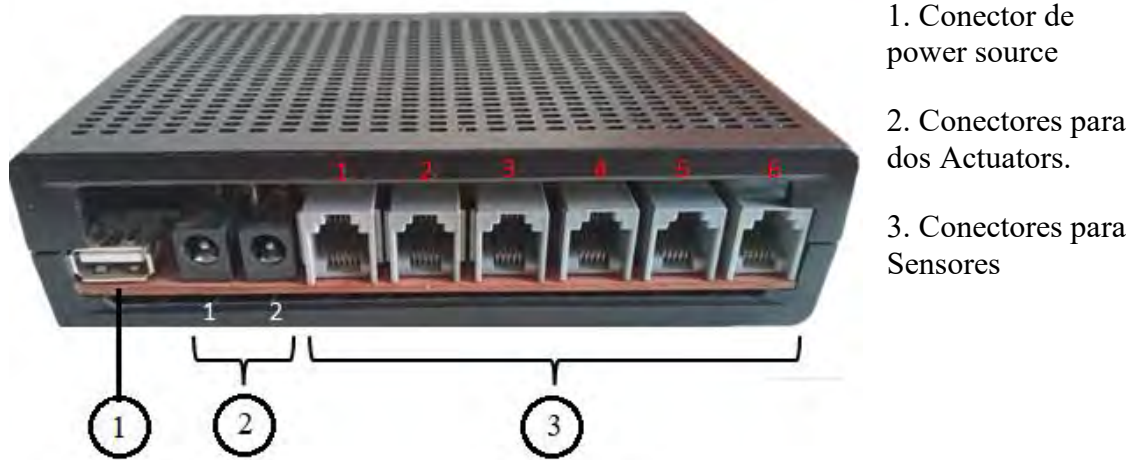
1. Pulsador RESET

2. Led de ON/OFF

3. Led de Actuators

4. Switch ON/OFF

CONECTORES DE ENTRADA – SALIDA



INTERFACE PARA EXPERIMENTOS DE ELECTRICIDAD

