



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE INGENIERÍA ELECTRÓNICA

“Diseño e Implementación de dos bancos de pruebas móviles
para mediciones eléctricas y control domótico mediante IoT”

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTOR: PAÚL ARMANDO BOBADILLA LÓPEZ

TUTOR: ING. MSC. RAFAEL ENRIQUE PÉREZ ORDOÑEZ

GUAYAQUIL – ECUADOR

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORIA

Yo, Paúl Armando Bobadilla López con documento de identificación N°0931656011 manifiesto que:

Soy los autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 1 agosto del año 2023.

Atentamente,



Paúl Armando Bobadilla López

CI: 0931656011

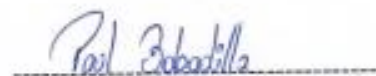
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, Paúl Armando Bobadilla López, con documento de identificación N° 0931656011, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado: "DISEÑO E IMPLEMENTACIÓN DE DOS BANCOS DE PRUEBAS MÓVILES PARA MEDICIONES ELÉCTRICAS Y CONTROL DOMÓTICO MEDIANTE IoT", el cual ha sido desarrollado para optar por el título de: INGENIERO ELECTRÓNICO, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que se hace la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 1 agosto del año 2023.

Atentamente,



Paúl Armando Bobadilla López

CI: 0931656011

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo, **Rafael Enrique Pérez Ordoñez**, con documento de identificación N°, 0916275076 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **"DISEÑO E IMPLEMENTACIÓN DE DOS BANCOS DE PRUEBAS MÓVILES PARA MEDICIONES ELÉCTRICAS Y CONTROL DOMÓTICO MEDIANTE IoT"**, realizado por **Paul Armando Bobadilla López** con documento de identificación N° 0931656011 obteniendo como resultado final el trabajo de titulación bajo la opción de trabajo de grado que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 1 agosto del año 2023.

Atentamente,



Rafael Enrique Pérez Ordoñez

CI: 0916275076

DEDICATORIA

Mi título universitario se lo dedico a Dios que sé que sin Él hubiera sido imposible lograrlo, ya que Él proveyó de los medios para que mis padres y mi persona pueda culminar con éxitos mis estudios.

Después de Dios dedico mi título a mis padres, hermana, hijo, familiares y amigos, quienes con paciencia y sabiduría supieron animarme día a día a seguir adelante hasta obtener mi tan deseado título de Ingeniero Electrónico.

Y especialmente le dedico este título a mi abuelo, que en paz descansa, que hasta lo estuvo pendiente de mis avances, por sus enseñanzas desde muy pequeño y lecciones de vida.

Paúl Armando Bobadilla López

AGRADECIMIENTO

La culminación del ciclo de los estudios universitarios atrae una amalgama de sentimiento, en donde me embarga la tristeza por los buenos y difíciles momentos vividos en el ámbito académico, pero me llena de alegría el saber que obtuve mi título universitario.

Agradezco a mis padres por su gran esfuerzo económico y por creer siempre en mí, que a pesar de una pandemia siempre buscaron la forma de que siga adelante.

Finalmente, y no menos importante a familiares y amigos que a lo largo de la carrera me ayudaron en lo académico y con palabras de aliento que me ayudaron a culminar este tan logrado objetivo.

Paúl Armando Bobadilla López

RESUMEN

AÑO	ALUMNO	DIRECTOR DE PROYECTO TÉCNICO	TEMA DE PROYECTO TÉCNICO
2023	Paúl Armando Bobadilla López	ING. Rafael Enrique Pérez Ordoñez MSc.	DISEÑO E IMPLEMENTACIÓN DE DOS BANCOS DE PRUEBAS MÓVILES PARA MEDICIONES ELÉCTRICAS Y CONTROL DOMÓTICO MEDIANTE IoT

El presente trabajo se centra en el diseño e implementación de dos bancos de pruebas móviles para mediciones eléctricas y control domótico mediante el uso de IoT. Estos bancos de pruebas están diseñados para brindar a los estudiantes de Ingeniería Electrónica y Automatización una herramienta de aprendizaje práctico que les permita realizar simulaciones de procesos industriales y actividades didácticas.

En el desarrollo de este proyecto se han considerado varios elementos, como la instrumentación electrónica, el diseño electrónico y la programación en pantallas inteligentes NEXTION. Los bancos de pruebas cuentan con sensores y dispositivos para realizar mediciones eléctricas precisas y también incorporan componentes para el control domótico, permitiendo a los estudiantes interactuar con dispositivos inteligentes.

El proyecto también incluye una interfaz desarrollada en NEXTION, que permitirá a los estudiantes visualizar en tiempo real los datos de las mediciones eléctricas y la salida de los controladores.

Palabras claves: lot, sensor, convertidor digital, nextion, microprocesador, efecto hall.

ABSTRACT

YEAR	STUDENTS	DIRECTOR OF TECHNICAL PROJECT	TECHNICAL PROJECT THEME
2023	Paúl Armando Bobadilla López	ING. Rafael Enrique Pérez Ordoñez MSc.	Design and Implementation of Two Mobile Test Benches for Electrical Measurements and Home Automation Control using IoT"

This work focuses on the design and implementation of two mobile test benches for electrical measurements and home automation control using IoT. These test benches are designed to provide students in Electronic and Automation Engineering with a practical learning tool that allows them to simulate industrial processes and educational activities.

Several elements have been considered in the development of this project, such as electronic instrumentation, electronic design, and programming on NEXTION smart displays. The test benches are equipped with sensors and measurement devices to perform precise electrical measurements. They also incorporate components for home automation control, enabling students to interact with smart devices.

The project also includes an interface developed on NEXTION, which allows students to visualize real-time data from electrical measurements and the output of the controllers.

Keywords: lot, sensor, digital converter, nextion, microprocessor, hall effect.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA	
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR	
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN	
DEDICATORIA	
AGRADECIMIENTO	
RESUMEN	
ABSTRACT	
ÍNDICE GENERAL	
ÍNDICE DE FIGURAS.....	
ÍNDICE DE TABLAS.....	
INTRODUCCIÓN.....	1
1. EL PROBLEMA.....	3
1.1. Antecedentes.....	3
1.2. Importancia y Alcances.....	3
1.3. Delimitación	4
1.3.1. Temporal	4
1.3.2. Espacial.....	4
1.3.3. Académica	4
1.4. Innovación.....	5
1.5. Objetivos.....	5
1.5.1. Objetivo general.....	5
1.5.2. Objetivos específicos	5
2. FUNDAMENTOS TEÓRICOS.....	6
2.1. Sistema de adquisición de datos.....	6
2.2. Tipos de tarjetas de adquisición de datos.....	6
2.2.1. Tarjetas de adquisición analógica.....	6
2.2.2. Tarjetas de adquisición digital	7
2.2.3. Tarjetas de adquisición mixta	7
2.2.4. Tarjetas de adquisición alta velocidad.....	7
2.2.5. Tarjetas de adquisición específicas	7
2.2.6. Tarjetas de adquisición con acondicionamiento de señal.....	8
2.3. Microcontroladores	8
2.3.1. Microcontroladores de propósito específico.....	8
2.3.1. Microcontroladores de bajo consumo.....	8
2.3.2. Microcontroladores con capacidad de red.....	9

2.3.3. Microcontroladores FPGA	9
2.3.4. Microcontroladores de alto rendimiento	9
2.4. Arduino	9
2.5. ESP32.....	10
2.6. Lectura digital	11
2.7. Lectura análoga.....	11
2.8. Conversión de analógico a digital	11
2.8.1. Muestreo y retención (Sample and Hold)	11
2.8.2. Aproximación sucesiva (Successive Approximation).....	11
2.8.3. Aproximación por rampa (Ramp)	12
2.8.4. Flash (o conversor paralelo)	12
2.8.5. Sigma-Delta	12
2.9. Conversión de digital a analógico	13
2.9.1. Convertidor digital a analógico por resistencias ponderadas R-2R DAC	13
2.9.2. Convertidor de aproximaciones sucesivas (Successive Approximation DAC)	13
2.9.3. Convertidor digital a analógico por escalera (R-2R Ladder DAC)	13
2.9.4. Convertidor digital a analógico por aproximación por corriente (Current Steering DAC).....	14
2.9.5. Convertidor digital a analógico por pulso de ancho modulado (Pulse Width Modulation DAC)	14
2.10. Sensor de corriente	14
2.10.1. Transformadores de corriente	14
2.10.2. Sensores de efecto Hall	15
2.10.3. Resistores shunt.....	15
2.10.4. Sensores de efecto Rogowski.....	16
2.11. Sensores de voltaje	16
2.11.1. Resistores de carga	17
2.11.2. Divisores de voltaje	17
2.11.3. Sensores basados en efecto Hall	18
2.11.4. Sensores de capacitancia.....	18
2.12. Sensores de potencia.....	19
2.13. Pantallas NEXTION	19
2.14. Relé de estado sólido (SSR)	19
2.15. MOSFET	20
2.16. TRIAC	21
2.17. DIAC.....	22
2.18. Internet de las cosas	22

2.19. Plataformas en la web para IoT	23
2.19.1. AWS IoT	23
2.19.2. Microsoft Azure IoT	24
2.19.3. Google Cloud IoT	24
2.19.4. IBM Watson IoT Platform	24
2.19.5. ThingWorx	24
2.19.6. Siemens MindSphere	24
2.19.7. Ubidots	25
2.20. Token en IOT	25
3. MARCO METODOLÓGICO	26
3.1. Sistema propuesto para el modulo	26
3.2. Diseño del proyecto	27
3.3. Planos del proyecto	28
3.4. Flujograma y funcionamiento del proyecto	29
3.5. Tarjetas pcb del modulo.....	31
3.6. Módulos y sensores del proyecto.....	34
3.7. Diagramas de conexiones internos del modulo	37
3.8. Enlace de comunicación entre NEXTION y embebidos	39
3.9. Enlace de comunicación entre embebido de adquisición y embebido de enlace a plataforma	43
3.10. Configuración en Plataforma Ubidots	46
4. MANUAL DE PRÁCTICAS	54
Práctica#1: Control y lectura de señal para dicroico	54
Práctica#2: Control y lectura de señal para punto electrico	55
Práctica#3: Control y lectura de señal para carga	56
Práctica#4: Control IoT en maleta didactica	57
Práctica#5: Control y lectura de maleta mediante IoT y nextion	58
RESULTADOS	59
CRONOGRAMA	62
PRESUPUESTO.....	63
CONCLUSIONES	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRAFICAS.....	66
ANEXOS	70
A. – Planos de la maleta.....	70
B. – Planos de las tarjetas de circuito impreso	71
C. Programación de embebido de pantalla	75
D. – Programación de embebido de enlace a plataforma Ubidots.....	84

E. – Programación en Ubidots.....	94
F. RESOLUCIÓN DE PRÁCTICA 1.....	97
G. RESOLUCIÓN DE PRÁCTICA 2.....	103
H. RESOLUCIÓN DE PRÁCTICA 3.....	109
I. RESOLUCIÓN DE PRÁCTICA 4.....	115
J. RESOLUCIÓN DE PRÁCTICA 5.....	125

ÍNDICE DE FIGURAS

Figura 1: Delimitación espacial del módulo del Proyecto Técnico	4
Figura 2: Diagrama de un sistema con Daq o Das.....	6
Figura 3: Arduino.....	10
Figura 4: Stm32.....	10
Figura 5: Transformadores de corriente	15
Figura 6: Sensor de efecto hall.....	15
Figura 7: Resistor de shunt	16
Figura 8: Sensores de efecto Rogowski.....	16
Figura 9: Resistor de carga	17
Figura 10: Sensores de efecto Rogowski	17
Figura 11: Sensores basados en efecto Hall.....	18
Figura 12: Sensores de capacitancia para voltaje ZMPT101B.....	19
Figura 13: Relé de estado sólido	20
Figura 14: MOSFET.....	21
Figura 15: TRIAC	21
Figura 16: DIAC.....	22
Figura 17: Comunicación entre microcontroladores y proyecto	26
Figura 18: Control para activación de relés solidos.....	26
Figura 19: Comunicación entre embebido y pantalla	27
Figura 20: Maleta de aluminio.....	27
Figura 21: Placa del proyecto	28
Figura 22: Dimensiones de placa	28
Figura 23: Selección de práctica	29
Figura 24: Funcionamiento de práctica 1	29
Figura 25: Funcionamiento de práctica 2.....	29
Figura 26: Funcionamiento de práctica 3.....	30
Figura 27: Funcionamiento de práctica 4.....	30
Figura 28: Funcionamiento de práctica 5.....	31
Figura 29: Esquemático de microcontrolador para sensores y pantalla	32
Figura 30: Esquemático de microcontrolador para enlace con plataforma	33
Figura 31: Vista de implementación parte interna del banco.....	33
Figura 32: Sensor de corriente Acs712 (naylormechanics, 2022)	34
Figura 33: Rele solido (ArduSpain, 2022)	35
Figura 34: Arduino nano con Atmel 32ua (descubrearduino, 2022)	35
Figura 35: Esp32 WROOM (upesy, 2022).....	36
Figura 36: Nextion y Arduino nano.....	36
Figura 37: Esquema del modulo.....	37
Figura 38: Componentes del modulo.....	38
Figura 39: Programa de Atmel 32 – Declaración de librerías	39
Figura 40: Programa de Atmel 32 – Declaración de objetos	39
Figura 41: Programa de Atmel 32 – Declaración de variables	40
Figura 42: Programa de Atmel 32 – método configuración.....	40
Figura 43: Programa de Atmel 32 – método principal.....	41
Figura 44: Programa de Atmel 32 – método nextion consultar	41
Figura 45: Programa de Atmel 32 – método nextion publicar.....	42
Figura 46: Programa de Atmel 32 – método obtener corriente	42
Figura 47: Programa de Atmel 32 – método publicar iot	43
Figura 48: Programa de Atmel 32 – método consultar iot.....	43
Figura 49: Programa en esp32 – declaración de librerías.....	44

Figura 50:	Programa en esp32 – declaración de variables.....	44
Figura 51:	Programa en esp32 – método callback.....	45
Figura 52:	Programa en esp32 – método de configuración	45
Figura 53:	Programa en esp32 – método de loop	46
Figura 54:	Programa en esp32 – método principal.....	46
Figura 55:	Creación de dispositivo en Ubidots.....	47
Figura 56:	Dispositivo en Ubidots llamado maleta.....	47
Figura 57:	Variabales en dispositivo maleta	48
Figura 58:	Espacio de trabajo para prácticas.....	49
Figura 59:	Espacio de trabajo en práctica 1	50
Figura 60:	Espacio de trabajo en práctica 2	50
Figura 61:	Espacio de trabajo en práctica 3	51
Figura 62:	Espacio de trabajo en práctica 4	52
Figura 63:	Espacio de trabajo en práctica 5	53
Figura 64:	Módulos didácticos	59
Figura 65:	Plataforma Ubidots	59
Figura 66:	Espacios de trabajo en Ubidots para supervisión de prácticas	60
Figura 67:	Distribución de sensores y relés.....	61
Figura 68:	Dimensiones de placa de bornes	70
Figura 69:	Esquemático de Atmel y sensores.....	71
Figura 70:	Tarjeta de circuito impreso para atmel32.....	72
Figura 71:	Esquemático para tarjeta esp32.....	73
Figura 72:	Tarjeta de circuito impreso para esp32	74
Figura 73:	Creación de dispositivo en Ubidots.....	94
Figura 74:	Creación de dispositivo en Ubidots.....	94
Figura 75:	Creación de dispositivo en Ubidots.....	95
Figura 76:	Creación de dispositivo en Ubidots.....	95
Figura 77:	Creación de dispositivo en Ubidots.....	95
Figura 78:	Creación de dispositivo en Ubidots.....	96
Figura 79:	Creación de dispositivo en Ubidots.....	96
Figura 80:	Diagrama de conexiones físicas – Práctica 1.....	98
Figura 81:	Conexiones físicas – Práctica 1.....	99
Figura 82:	Pantalla de inicio en Nextion – Práctica 1.....	99
Figura 83:	Panel en Nextion – Práctica 1.....	100
Figura 84:	Vista de panel en Nextion – Práctica 1.....	100
Figura 85:	Activación de foco dicroico 1 – Práctica.....	101
Figura 86:	Activación de foco dicroico 2 – Práctica.....	101
Figura 87:	Visualización de consumos en Nextion de los dicroicos – Práctica.....	102
Figura 88:	Visualización de consumos en pinza amperimétrica de los dicroicos – Práctica.....	102
Figura 89:	Diagrama de conexión – Práctica 2	104
Figura 90:	Conexiones físicas y a la red 110VAC – Práctica 2	105
Figura 91:	Panel de inicio en Nextion – Práctica 2.....	105
Figura 92:	Panel – Práctica 2.....	106
Figura 93:	Vista Física de panel – Práctica 2	106
Figura 94:	Activación de punto eléctrico mediante pantalla nextion – Práctica 2.....	107
Figura 95:	Conexiones físicas – Práctica.....	107
Figura 96:	Conexiones medición de pinza amperimétrica – Práctica 2	108
Figura 97:	Conexiones medición de pinza amperimétrica – Práctica 2	108
Figura 98:	Conexiones físicas – Práctica 3.....	110
Figura 99:	Conexiones físicas – Práctica 3.....	111
Figura 100:	Panel de inicio nextion – Práctica 3.....	111

Figura 101:	Panel – Práctica 3.....	112
Figura 102:	Conexiones físicas – Práctica 3	112
Figura 103:	Encendido de cargas – Práctica 3	113
Figura 104:	Consumo de cargas – Práctica 3	113
Figura 105:	Lectura y comparación con pinza amperimétrica – Práctica 3	114
Figura 106:	Conexiones físicas – Práctica 4	116
Figura 107:	Conexiones físicas – Práctica 4	117
Figura 108:	Panel de inicio – Práctica 4.....	117
Figura 109:	Panel – Práctica 4.....	118
Figura 110:	Panel en Ubidots sin conexión– Práctica 4	119
Figura 111:	Panel en Ubidots con conexión– Práctica 4.....	120
Figura 112:	Panel en Ubidots con conexión– Práctica 4.....	121
Figura 113:	Panel en Ubidots encendido de dicroico – Práctica 4.....	121
Figura 114:	Encendido de dicroico desde Ubidots– Práctica 4	122
Figura 115:	Panel nextion con señal desde Ubidots– Práctica 4	122
Figura 116:	Panel nextion con señal desde Ubidots– Práctica 4.....	123
Figura 117:	Panel nextion con señal desde Ubidots– Práctica 4	124
Figura 118:	Conexiones físicas – Práctica 5	126
Figura 119:	Conexiones físicas – Práctica 5	127
Figura 120:	Conexiones físicas – Práctica 5	127
Figura 121:	Panel – Práctica 5.....	128
Figura 122:	Panel en Ubidots sin conexión– Práctica 5	129
Figura 123:	Panel en Ubidots con conexión– Práctica 5.....	130
Figura 124:	Panel en Ubidots con conexión– Práctica 5.....	131
Figura 125:	Panel en Nextion– Práctica 5.....	131
Figura 126:	Panel en Ubidots encendido de dicroico – Práctica 5.....	132
Figura 127:	Encendido de dicroico desde Ubidots– Práctica 5	132
Figura 128:	Panel nextion con señal desde Ubidots– Práctica 5	133
Figura 129:	Panel nextion con señal desde Ubidots– Práctica 5.....	134

ÍNDICE DE TABLAS

Tabla 1 :Tabla de resultados	61
Tabla 2 :Tabla de resultados Práctica 1	102
Tabla 3 :Tabla de resultados Práctica 2	108
Tabla 4 :Tabla de resultados Práctica 3	114
Tabla 5 :Tabla de resultados Práctica 4	124
Tabla 6 :Tabla de resultados Práctica 5	134

INTRODUCCIÓN

El avance de la tecnología ha transformado significativamente la forma en que se interactúa con el entorno que nos rodea. En particular, el Internet de las cosas (IoT) ha emergido como una poderosa herramienta para la interconexión de dispositivos y sistemas, brindando un nuevo enfoque hacia la automatización y el control inteligente en diversos ámbitos. En el contexto de esta revolución tecnológica, el presente trabajo tiene como objetivo abordar el diseño e implementación de dos bancos de pruebas móviles, enfocados en mediciones eléctricas y control domótico, mediante la aplicación de soluciones IoT.

La necesidad de desarrollar estos bancos de pruebas surge de la creciente demanda de soluciones innovadoras y eficientes para evaluar y optimizar sistemas eléctricos y domóticos en distintos escenarios. La capacidad de medir y analizar de manera precisa y confiable los parámetros eléctricos y el funcionamiento de dispositivos domóticos es fundamental para asegurar la seguridad, eficiencia y sostenibilidad en entornos residenciales, comerciales e industriales.

Para llevar a cabo las mediciones eléctricas y el control domótico, se considerarán diversas variables críticas. En el caso de las mediciones eléctricas, se evaluarán parámetros como el voltaje, la corriente, la potencia activa y reactiva, el factor de potencia y otros aspectos relacionados con la calidad de la energía eléctrica. En cuanto al control domótico, se medirán variables como la temperatura ambiente, la humedad, el encendido y apagado de dispositivos, así como la interacción entre los elementos domóticos y la plataforma IoT.

La implementación de estos bancos de pruebas requerirá el uso de herramientas y tecnologías específicas. Se emplearán sensores y dispositivos de adquisición de datos que permitan capturar información en tiempo real. Además, se utilizarán microcontroladores y módulos de comunicación que faciliten la interconexión de los sistemas y la transmisión de datos hacia una

plataforma centralizada. Estas herramientas permitirán la gestión remota de los bancos de pruebas y la obtención de datos para su posterior análisis y evaluación.

Los resultados esperados de este trabajo son significativos. La implementación de los bancos de pruebas móviles posibilitará la realización de mediciones eléctricas y pruebas de control domótico en diferentes contextos, brindando una plataforma versátil y escalable para la investigación y el desarrollo de soluciones IoT aplicadas a la eficiencia energética y la automatización inteligente. Además, se espera contribuir a la mejora de la calidad de vida de las personas, al facilitar la implementación de sistemas eléctricos y domóticos más seguros, eficientes y adaptables a las necesidades de cada usuario.

Se proponen prácticas básicas y otras enfocadas en el tema de titulación. Se explican varios procesos y aplicaciones que un estudiante puede lograr con el módulo didáctico que demuestre sus destrezas para crear e innovar más procesos a nivel industrial.

El antecedente para el texto dado es un proyecto de diseño e implementación de dos bancos de pruebas móviles para mediciones eléctricas y control domótico utilizando IoT. Estos bancos de pruebas están destinados a brindar a los estudiantes de Ingeniería Electrónica y Automatización una herramienta práctica para realizar simulaciones de procesos industriales y actividades didácticas. En el desarrollo del proyecto, se han tenido en cuenta elementos como la instrumentación electrónica, el diseño electrónico y la programación en NEXTION. Los bancos de pruebas cuentan con sensores y dispositivos de medición para mediciones eléctricas precisas, así como componentes para el control domótico, lo que permite a los estudiantes interactuar con dispositivos inteligentes.

1. EL PROBLEMA

1.1. Antecedentes

El problema principal por el que se plantea el desarrollo de una maleta didáctica es poder visualizar el consumo de la potencia de energía eléctrica simulando un control sectorizado dentro de un hogar o industria.

En una implementación dentro de la universidad como trabajo de titulación en la carrera de electrónica y automatización incluye el uso de servidores locales para visualizar consumo de una carga eléctrica conectada a equipo mediante el uso de raspberry, el proyecto se estudia (Cifuentes Rubio, 2020), y tomando en cuenta dicho desarrollo se planteó el uso de una plataforma de código abierto como Ubidots.

1.2. Importancia y Alcances

En este trabajo se presenta el diseño e implementación de un banco didáctico móvil el cual permita realizar mediciones como el voltaje, corriente y la potencia de equipos o cargas a través de pruebas eléctricas que integren la parte teórica y experimental, permitiendo un aprendizaje en la parte eléctrica.

Debido a la continua evolución de los métodos de educación se ve la necesidad de implementar un módulo didáctico el cual permita el monitoreo de las variables como corriente, voltaje y potencia de las diferentes cargas, teniendo una interfaz de supervisión remota basada en el internet de las cosas, permitiendo la activación de las cargas y permitir el cálculo de las variables como la corriente, voltaje y potencia de cada elemento instalado.

1.3. Delimitación

1.3.1. Temporal

La implementación de este proyecto se realizó en un intervalo de seis meses a partir de la aprobación de este.

1.3.2. Espacial

El proyecto al finalizar se entregará a las instalaciones de algún laboratorio de la carrera Ingeniería Electrónica de la Universidad Politécnica Salesiana de la Sede Guayaquil como se detalla en la Figura 1.

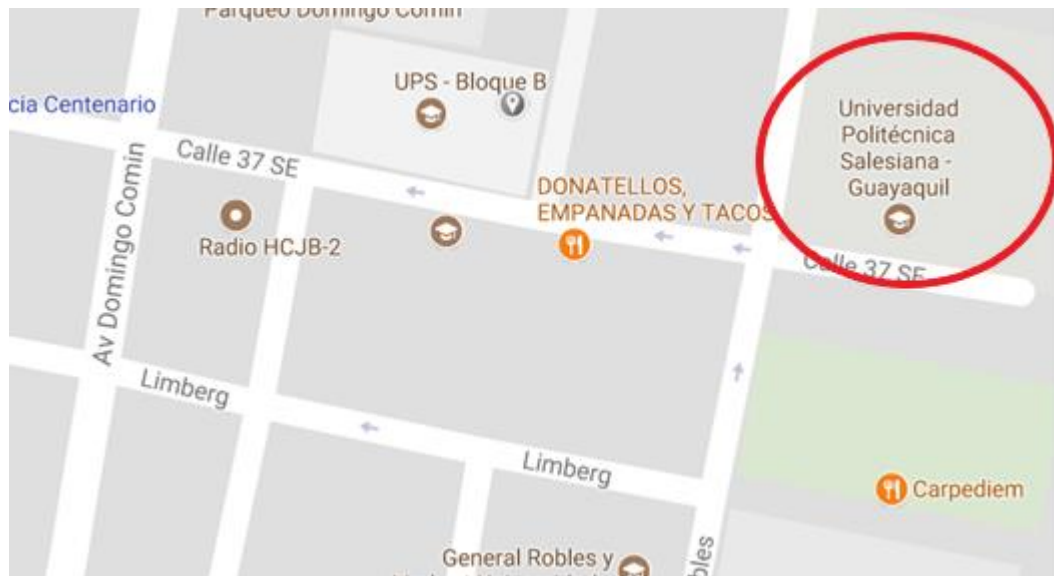


Figura 1: Delimitación espacial del módulo del Proyecto Técnico (Moreira, Rodríguez)

1.3.3. Académica

El módulo diseñado, provee un alcance académico que involucran conocimientos obtenidos en las siguientes materias: Electiva I, Electiva II, Teoría del Diseño, Electrónica de Potencia, Circuitos Eléctricos I, Procesamiento Digital de Señales e Informática Industrial.

1.4. Innovación

La finalidad de este proyecto de titulación es diseñar e implementar un módulo móvil donde se puedan integrar el internet de las cosas con la medición eléctrica de diferentes cargas.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar dos bancos de pruebas para mediciones eléctricas y control domótico mediante el internet de las cosas.

1.5.2. Objetivos específicos

- Realizar un sistema de supervisión para los datos de corriente total, voltaje y potencia total del módulo.
- Realizar módulos el cual permita la conexión de diferentes cargas mediante el internet de las cosas por medio de un microcontrolador de 32 bits.
- Desarrollar una guía que incluya 5 prácticas y buen uso de los bancos didácticos.

2. FUNDAMENTOS TEÓRICOS

2.1. Sistema de adquisición de datos

Un sistema de adquisición de datos es un grupo de sensores, componentes de hardware y programas de software que trabajan juntos con el fin de recoger las señales del mundo físico o real como se describe en la Figura 2, la unidad de procesamiento principal manipula los datos mediante el uso de software y hardware dedicados (Arrow , 2015).



Figura 2: Diagrama de un sistema con Daq o Das (Agüero, 2017)

2.2. Tipos de tarjetas de adquisición de datos

Según el trabajo de (Lizarzaburu Bazán, 2023) define que las tarjetas de adquisición de datos se dividen de acuerdo con la necesidad o entorno como lecturas de datos analógicos o digitales teniendo los siguientes tipos de tarjetas:

2.2.1. Tarjetas de adquisición analógica

Estas tarjetas permiten la captura de señales analógicas provenientes de sensores que miden variables como voltaje, corriente, temperatura, presión, entre otras, como diferentes rangos de entrada y resolución para adaptarse a las necesidades específicas de la aplicación (Duarte Acevedo, 2021).

2.2.2. Tarjetas de adquisición digital

Estas tarjetas se utilizan para capturar señales digitales, como pulsos o estados lógicos, provenientes de sensores o dispositivos que generan señales binarias, comúnmente utilizadas en aplicaciones de control, monitoreo o análisis de datos digitales (Cano Reyes, 2023).

2.2.3. Tarjetas de adquisición mixta

Estas tarjetas combinan la capacidad de adquisición tanto de señales analógicas como digitales, permiten capturar y sincronizar múltiples tipos de señales en una sola tarjeta, lo que resulta útil en aplicaciones que requieren la adquisición de datos mixtos o la correlación de eventos analógicos y digitales (Baño Baño, 2019).

2.2.4. Tarjetas de adquisición alta velocidad

Estas tarjetas están diseñadas para capturar señales a alta velocidad, generalmente en el rango de mega hertzios o giga hertzios, en aplicaciones como pruebas de comunicaciones, análisis de señales y procesamiento de imágenes en tiempo real.

2.2.5. Tarjetas de adquisición específicas

Algunas tarjetas están diseñadas para la adquisición de señales específicas, como señales de audio, señales de radiofrecuencia (RF), señales de video, señales de vibración, estas tarjetas suelen tener características especiales y adaptadas a la naturaleza de la señal que se va a adquirir (Sánchez Rohr, 2022).

2.2.6. Tarjetas de adquisición con acondicionamiento de señal

Estas tarjetas incluyen circuitos de acondicionamiento de señal integrados, que permiten amplificar, filtrar, aislar o adaptar la señal de entrada antes de ser digitalizada, para mejorar la calidad de la señal adquirida y facilitar su procesamiento posterior (Valencia, 2022).

2.3. Microcontroladores

Los microcontroladores son componentes esenciales en la electrónica moderna, diseñados específicamente para realizar tareas de control y procesamiento de datos en sistemas embebidos, se caracterizan por su tamaño compacto y su capacidad para ejecutar programas almacenados en su memoria interna (Fierro Lora, 2022).

2.3.1. Microcontroladores de propósito específico

Estos microcontroladores están diseñados para aplicaciones altamente especializadas, pueden incluir microcontroladores utilizados en sistemas de control industrial, automotriz, aeroespacial, médico, domótica, entre otros (Chan, 2011).

2.3.1. Microcontroladores de bajo consumo

Estos microcontroladores están optimizados para minimizar el consumo de energía, son ideales para aplicaciones que requieren una larga duración de la batería o funcionamiento con fuentes de energía limitadas, teniendo algunos ejemplos son la serie MSP430 de Texas Instruments y la familia ultra bajo consumo de los microcontroladores STM32L de STMicroelectronics (Mejía Vásquez, 2019).

2.3.2. Microcontroladores con capacidad de red

Estos microcontroladores están equipados con periféricos y protocolos de comunicación integrados para conectividad de red, pueden admitir Ethernet, Wi-Fi, Bluetooth, Zigbee y otros protocolos de comunicación, siendo adecuados para aplicaciones de Internet de las cosas (IoT), automatización del hogar y sistemas de monitoreo remoto (Herrera Calero, 2017).

2.3.3. Microcontroladores FPGA

Aunque técnicamente no son microcontroladores en sí mismos, los FPGA a menudo se utilizan en combinación con microcontroladores para agregar capacidad de procesamiento y flexibilidad, permiten implementar circuitos lógicos y periféricos personalizados, lo que los hace útiles en aplicaciones donde se requiere un alto grado de personalización (Vera Tasamá, 2017).

2.3.4. Microcontroladores de alto rendimiento

Estos microcontroladores están diseñados para aplicaciones que requieren un procesamiento intensivo, multitarea y alta velocidad de ejecución, teniendo arquitecturas de 32 bits y pueden incluir unidades de coma flotante, coprocesadores y amplias capacidades de memoria. Ejemplos notables son los microcontroladores ARM Cortex-M4 y Cortex-M7 utilizados en aplicaciones de alto rendimiento (CORTÉS OSORIO J. A., 2009).

2.4. Arduino

Es una placa electrónica de hardware libre que utiliza un microcontrolador reprogramable con una serie de pines que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (Cedeño-Rodríguez, 2020)



Figura 3:. Arduino (Cedeño-Rodríguez, 2020)

2.5. ESP32

Es una plataforma de desarrollo que integra un microcontrolador de 32 bits, basado en la arquitectura Xtensa LX6 de Espressif Systems. Este microcontrolador de bajo consumo tiene dos núcleos de procesamiento, lo que permite ejecutar múltiples tareas simultáneamente, donde cada núcleo puede funcionar a una velocidad de reloj de hasta 240 MHz y cuenta con una amplia gama de periféricos integrados (Pereira, 2022).



Figura 4:. Stm32 (Ordóñez, 2016).

2.6. Lectura digital

Es la lectura y procesamiento de señales o datos en forma de dígitos binarios (1 y 0), digital implica la conversión de señales analógicas a una representación digital mediante un proceso llamado muestreo y cuantificación (Pizarro Stuardo, 2020).

2.7. Lectura análoga

Es la medición y conversión de una señal analógica en una forma que pueda ser interpretada, se refiere al muestreo y cuantificación de una señal continua para obtener una representación digital que pueda ser procesada o visualizada por dispositivos electrónicos (CORTÉS OSORIO J. A., 2008).

2.8. Conversión de analógico a digital

La conversión analógica a digital (ADC) es un proceso fundamental en electrónica para capturar y representar una señal analógica en forma de datos digitales, son varios métodos utilizados en la conversión analógica a digital, cada uno con sus propias características y aplicaciones (Angulo Calle, 2009).

2.8.1. Muestreo y retención (Sample and Hold)

Este método implica tomar muestras instantáneas de la señal analógica a intervalos regulares y mantener cada muestra constante durante el tiempo de conversión. Utilizando un circuito de muestreo y retención, se captura la amplitud de la señal en momentos específicos, lo que permite su posterior conversión a una representación digital (CORTÉS OSORIO J. A., 2008)..

2.8.2. Aproximación sucesiva (Successive Approximation)

Este método es ampliamente utilizado debido a su eficiencia y precisión. Implica una serie de comparaciones sucesivas entre la señal analógica y un

valor digital estimado. Comienza con una estimación inicial, y mediante un proceso iterativo, se ajusta gradualmente el valor digital hasta que se alcance la mejor aproximación posible de la señal analógica (CORTÉS OSORIO J. A., 2008).

2.8.3. Aproximación por rampa (Ramp)

En este método, se genera una rampa lineal que se incrementa o decrementa a una tasa constante. La señal analógica se muestrea continuamente y se compara con la rampa. La conversión ocurre cuando la señal alcanza o cruza la rampa, y el tiempo necesario para alcanzar este punto se convierte en una representación digital de la señal analógica (CORTÉS OSORIO J. A., 2008).

2.8.4. Flash (o conversor paralelo)

Este método es rápido, pero requiere un hardware costoso. Consiste en una red de comparadores, donde cada comparador compara la señal analógica con un voltaje de referencia predefinido. Los resultados de estas comparaciones se utilizan para generar múltiples bits de salida simultáneamente, lo que permite una conversión rápida de la señal analógica a digital (CORTÉS OSORIO J. A., 2008).

2.8.5. Sigma-Delta

Este método se basa en la modulación por impulsos codificados (PCM) y utiliza técnicas de retroalimentación para mejorar la precisión. La señal analógica se compara con un valor de referencia y la diferencia se codifica en forma de impulsos. Estos impulsos se convierten en una secuencia digital que representa la señal analógica. El uso de la retroalimentación ayuda a mejorar la relación señal-ruido y la precisión de la conversión (CORTÉS OSORIO J. A., 2008).

2.9. Conversión de digital a analógico

La conversión digital a analógica (DAC) es el proceso inverso a la conversión analógica a digital. Implica tomar una señal digital y convertirla en una forma analógica continua (Paredes-Martínez, 2008).

Existen varios métodos utilizados para realizar esta conversión, cada uno con sus propias características y aplicaciones (Paredes-Martínez, 2008). A continuación, se describen algunos de los métodos más comunes:

2.9.1. Convertidor digital a analógico por resistencias ponderadas R-2R DAC

Este es uno de los métodos más utilizados. Consiste en una red de resistencias ponderadas que se conectan a los bits de la señal digital y las resistencias ponderadas generan una tensión proporcional al valor binario de cada bit, y todas las tensiones se suman para obtener la salida analógica (Serrano Guzmán, 2017).

2.9.2. Convertidor de aproximaciones sucesivas (Successive Approximation DAC)

Este método utiliza un proceso iterativo para aproximar el valor analógico deseado, que comienza con una estimación inicial y, mediante una serie de comparaciones sucesivas, se ajusta gradualmente el valor digital hasta que se alcance la mejor aproximación posible de la señal analógica deseada (Serrano Guzmán, 2017)..

2.9.3. Convertidor digital a analógico por escalera (R-2R Ladder DAC)

Este método utiliza una red de resistencias en forma de escalera, donde cada peldaño representa un bit de la señal digital, donde la corriente fluye a través

de los peldaños y se mezcla para generar una tensión de salida proporcional a la señal digital (Serrano Guzmán, 2017)..

2.9.4. Convertidor digital a analógico por aproximación por corriente (Current Steering DAC)

Este método se basa en la técnica de conmutación de corriente y se utiliza una red de interruptores controlados por los bits de la señal digital para direccionar corrientes proporcionales a cada bit, las corrientes se suman para generar una corriente analógica que se convierte en una tensión mediante una resistencia de carga (Serrano Guzmán, 2017)..

2.9.5. Convertidor digital a analógico por pulso de ancho modulado (Pulse Width Modulation DAC)

En este método, la señal digital se convierte en una señal de ancho de pulso (PWM) donde la duración del pulso varía proporcionalmente al valor digital y la señal PWM se filtra para obtener una forma de onda analógica (Serrano Guzmán, 2017).

2.10. Sensor de corriente

Los sensores de corriente se basan en diferentes principios de funcionamiento, pero en general, operan utilizando el efecto magnético de la corriente eléctrica, teniendo algunos de los tipos más comunes de sensores de corriente son:

2.10.1. Transformadores de corriente

Estos sensores utilizan el principio de la inducción electromagnética, poseen un devanado primario como se visualiza en la Figura 5, por donde circula la corriente a medir y un devanado secundario donde se induce una corriente proporcional a la corriente primaria, teniendo una relación entre la corriente

primaria y la corriente secundaria se establece mediante la relación de vueltas de los devanados (Hernández Areu, 2017).



Figura 5: Transformadores de corriente (Hernández Areu, 2017).

2.10.2. Sensores de efecto Hall

Estos sensores se basan en el principio del efecto Hall, que es la generación de una diferencia de potencial transversal a un conductor cuando se somete a un campo magnético y una corriente, convirtiendo este campo en una señal de salida proporcional (Guarella, 2011). Visualización física Figura 6.

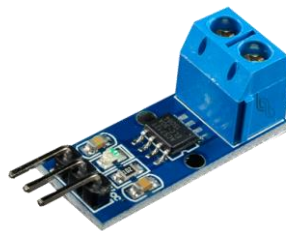


Figura 6: Sensor de efecto hall (Guarella, 2011).

2.10.3. Resistores shunt

Los resistores shunt son resistencias de baja resistividad que se colocan en serie con el conductor por donde circula la corriente. Al medir la caída de voltaje a través de la resistencia shunt, es posible determinar la corriente que la atraviesa utilizando la ley de Ohm (Bezunartea Oroz, 2021).



Figura 7: Resistor de shunt (Bezunartea Oroz, 2021).

2.10.4. Sensores de efecto Rogowski

Los sensores de efecto Rogowski, también conocidos como bobinas de Rogowski, son dispositivos utilizados para medir la corriente eléctrica en un conductor a través de la bobina (Seuret Jiménez, 2014).

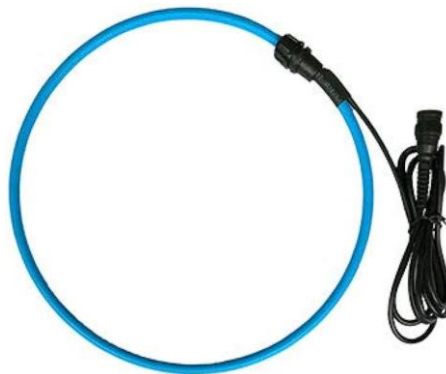


Figura 8: Sensores de efecto Rogowski (Seuret Jiménez, 2014).

2.11. Sensores de voltaje

Los sensores de voltaje pueden basarse en diferentes principios de funcionamiento, pero en general, operan midiendo la caída de voltaje o el potencial eléctrico en relación con una referencia de voltaje conocida. Algunos de los tipos más comunes de sensores de voltaje son:

2.11.1. Resistores de carga

Estos sensores se basan en la caída de voltaje que ocurre a través de un resistor cuando circula corriente a través de él. Al medir la caída de voltaje a través del resistor de carga, es posible determinar el voltaje aplicado (FORERO, 2011).

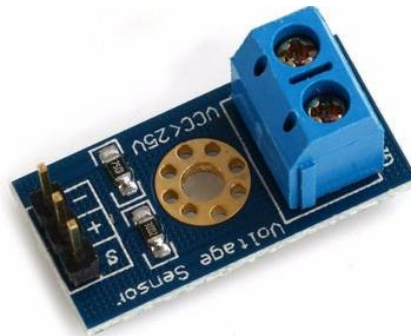


Figura 9:. Resistor de carga (FORERO, 2011).

2.11.2. Divisores de voltaje

Estos sensores utilizan una red de resistores conectados en serie o en paralelo para dividir el voltaje aplicado en proporciones conocidas. Midiendo el voltaje en un punto específico de la red de resistores, se puede determinar el nivel de voltaje total (Peña-Lecona, 2008).

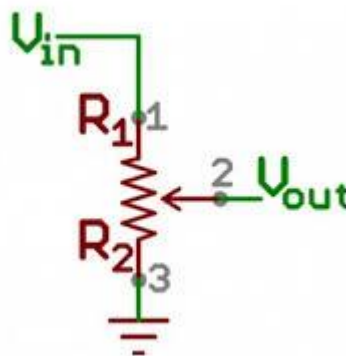


Figura 10:.. Sensores de efecto Rogowsk (Peña-Lecona, 2008).

2.11.3. Sensores basados en efecto Hall

Algunos sensores de voltaje utilizan el efecto Hall para medir campos magnéticos generados por corrientes eléctricas y, a partir de ahí, inferir el voltaje presente. Estos sensores aprovechan las propiedades magnéticas y eléctricas como se ilustra en Figura 11 de los materiales para realizar mediciones precisas de voltaje: (Peña-Lecona, 2008).

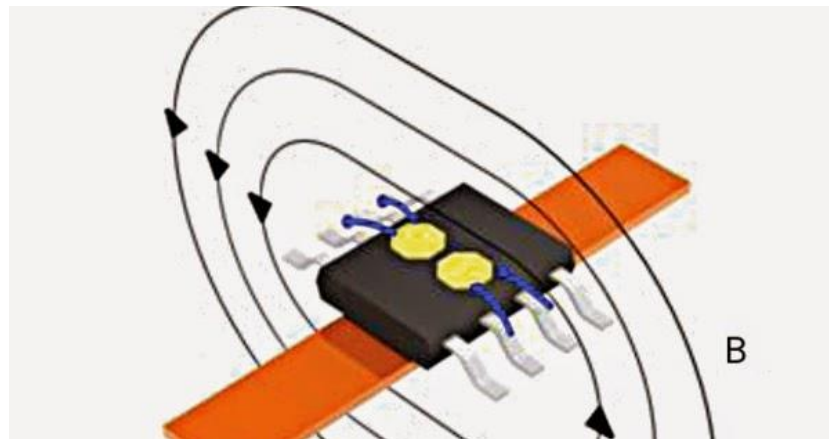


Figura 11: Sensores basados en efecto Hall (Peña-Lecona, 2008).

2.11.4. Sensores de capacitancia

Estos sensores miden cambios en la capacitancia de un condensador para determinar cambios en el voltaje aplicado. La capacitancia está relacionada con el voltaje, y al medir cambios en esta propiedad, se puede deducir el nivel de voltaje (Peña-Lecona, 2008)..

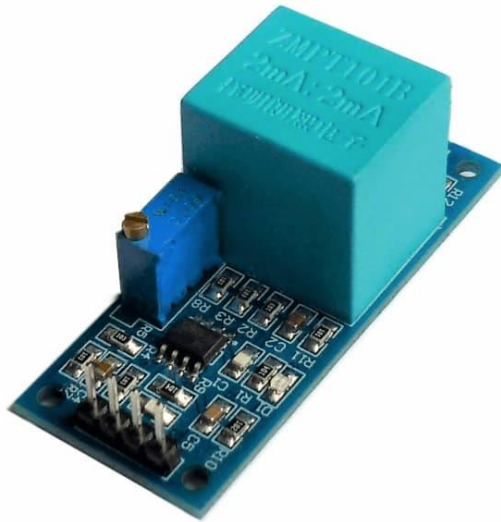


Figura 12: Sensores de capacitancia para voltaje ZMPT101B (Peña-Lecona, 2008).

2.12. Sensores de potencia

Los sensores de potencia desempeñan un papel importante en la gestión de la energía eléctrica, permitiendo la medición precisa y el monitoreo de la potencia en diferentes sistemas, como sistemas de distribución de energía, sistemas de control de energía y aplicaciones industriales (Salazar, 2021).

2.13. Pantallas NEXTION

Es un dispositivo de interfaz humano-máquina (HMI) que combina una pantalla táctil y un controlador integrado, permitiendo la visualización y control de datos en proyectos electrónicos. Proporciona una interfaz intuitiva para interactuar con el usuario, y su diseño simplificado y fácil programación la hacen una opción popular en aplicaciones donde se requiere una interfaz gráfica sencilla y personalizable (Villacrés Rodríguez, 2017).

2.14. Relé de estado sólido (SSR)

Un relé de estado sólido (SSR) es un dispositivo de conmutación electrónico que opera sin partes móviles. Utiliza componentes semiconductores, como optoacopladores y tiristores, para controlar la conexión o desconexión de la

corriente eléctrica en un circuito. Los SSR ofrecen ventajas como alta velocidad de conmutación, larga vida útil, aislamiento eléctrico y resistencia a golpes y vibraciones. Son utilizados en aplicaciones donde se requiere un control fiable y de alta precisión, como sistemas de automatización industrial, control de temperatura y equipos médicos (Fernández Angulo, 2022).



Figura 13: Relé de estado sólido (Fernández Angulo, 2022).

2.15. MOSFET

Es un dispositivo semiconductor que controla el flujo de corriente en un circuito mediante un campo eléctrico que actúa como un interruptor controlado por voltaje, permitiendo o bloqueando el paso de corriente según la tensión aplicada a su terminal de compuerta (Ramírez, 2022).

Los MOSFETs ofrecen una baja resistencia de encendido (ON) y una alta resistencia de apagado (OFF), lo que los hace ideales para aplicaciones de conmutación y amplificación de señales en electrónica, como en fuentes de alimentación, controladores de motores y circuitos integrados (Ramírez, 2022).

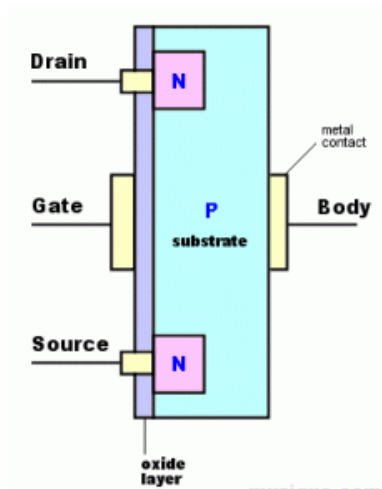


Figura 14: MOSFET (Ramírez, 2022).

2.16. TRIAC

Es un dispositivo de control de potencia bidireccional que permite la conducción de corriente en ambos sentidos en un circuito de corriente alterna (AC) que actúa como un interruptor electrónico que se activa mediante una pequeña corriente de compuerta (Álvarez-Moreira, 2022).

El TRIAC es ampliamente utilizado en aplicaciones de control de potencia AC, como reguladores de luz, controladores de velocidad de motores y control de temperatura, ya que puede regular la cantidad de energía que se entrega a una carga conectada al circuito AC (Álvarez-Moreira, 2022).

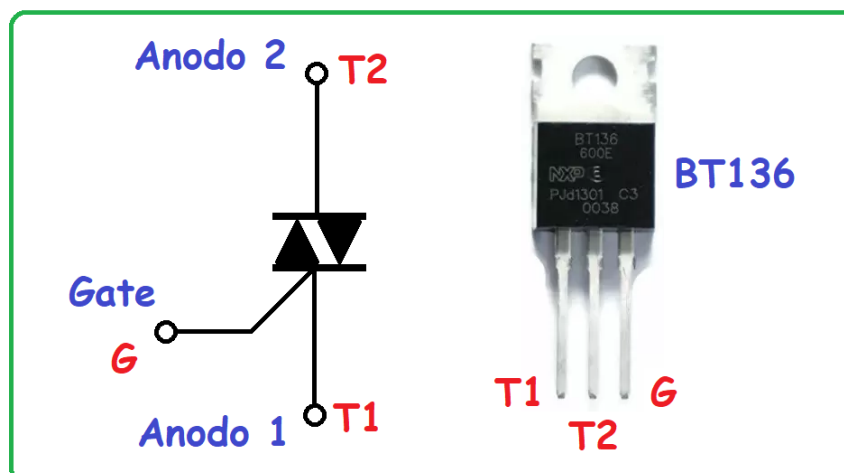


Figura 15: TRIAC (Álvarez-Moreira, 2022).

2.17. DIAC

Es un dispositivo semiconductor de disparo utilizado principalmente en aplicaciones de control de potencia, que es un tipo especial de diodo bidireccional que se dispara cuando la tensión alcanza un cierto nivel y permite el flujo de corriente en ambos sentidos (Flores Jacome, 2020).

El DIAC se utiliza comúnmente en circuitos de disparo de triac y en aplicaciones de control de encendido/apagado, como en temporizadores, reguladores de luz y sistemas de control de temperatura. Su función es proporcionar una activación precisa y confiable para el control de la corriente alterna (Flores Jacome, 2020).

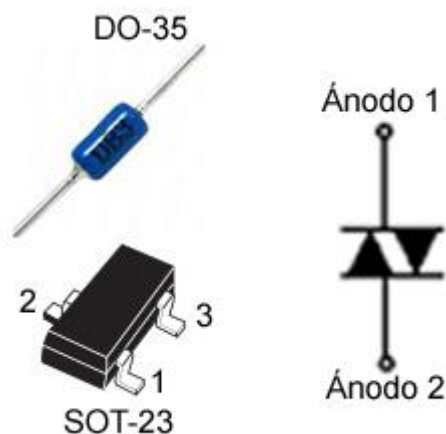


Figura 16:. DIAC (Flores Jacome, 2020).

2.18. Internet de las cosas

Es un concepto que se refiere a la interconexión de objetos cotidianos a través de Internet, permitiéndoles recopilar, intercambiar y actuar sobre datos. En el IoT, dispositivos y sensores incorporados en diversos objetos, desde electrodomésticos hasta vehículos y equipos industriales, se comunican entre sí y con aplicaciones y sistemas para recopilar información y realizar acciones automatizadas (Garrell, 2019).

Esto crea un ecosistema donde los datos se pueden analizar, optimizar procesos y mejorar la toma de decisiones en tiempo real, generando eficiencia, comodidad y nuevas oportunidades en diversos campos como hogares inteligentes, ciudades inteligentes, salud, agricultura y más (Viteri Castellano, 2022).

2.19. Plataformas en la web para IoT

Las plataformas en la web para IoT se dividen en distintos tipos según su funcionalidad. Las plataformas de conectividad se centran en facilitar la comunicación entre dispositivos IoT y servicios en la nube. Las plataformas de desarrollo ofrecen herramientas para crear aplicaciones y servicios personalizados para IoT (Muñoz Rodríguez, 2023).

Las plataformas de análisis se enfocan en el procesamiento y análisis de datos generados por los dispositivos IoT. Por último, las plataformas de gestión proporcionan herramientas para administrar y controlar los dispositivos y las soluciones de IoT de manera eficiente (Muñoz Rodríguez, 2023).

Estas plataformas son fundamentales para impulsar la adopción y el desarrollo de soluciones innovadoras en el ámbito del Internet de las cosas (Muñoz Rodríguez, 2023).

2.19.1. AWS IoT

Es una plataforma de Amazon Web Services que proporciona servicios y herramientas para conectar, gestionar y analizar dispositivos IoT a gran escala, además de permitir la integración con otros servicios de la nube de AWS (Supo Halanoca, 2022).

2.19.2. Microsoft Azure IoT

Es una plataforma de Microsoft que ofrece servicios de conectividad, almacenamiento y análisis de datos para IoT que permite el desarrollo de soluciones personalizadas y la integración con otros servicios de Azure (Lammini Rodríguez, 2023).

2.19.3. Google Cloud IoT

Es la plataforma de Google Cloud que brinda servicios para la conectividad, administración y procesamiento de datos de dispositivos IoT que incluye herramientas de análisis y machine learning para obtener insights valiosos (Vásquez Lema, 2022).

2.19.4. IBM Watson IoT Platform

Es una plataforma de IBM que combina la conectividad de dispositivos IoT, el análisis de datos y la inteligencia artificial de Watson que permite la creación de soluciones escalables y seguras para diversos sectores (González Suarez, 2023).

2.19.5. ThingWorx

Es una plataforma de PTC que ofrece capacidades de desarrollo rápido de aplicaciones IoT, que proporciona herramientas de conectividad, análisis y visualización de datos, así como funcionalidades de realidad aumentada y gemelo digital (Islas, 2022).

2.19.6. Siemens MindSphere

Es una plataforma de IoT de Siemens que permite la conexión de dispositivos y la gestión de datos en tiempo real, que proporciona herramientas para el análisis de datos, la simulación y la optimización de procesos industriales (Pulache Huertas, 2023).

2.19.7. Ubidots

Permite la recopilación, visualización y análisis de datos en tiempo real provenientes de sensores y dispositivos conectados. Ubidots proporciona una interfaz intuitiva y herramientas de arrastrar y soltar para crear paneles de control personalizados, gráficos, notificaciones y reglas automatizadas (Tejada Gutierrez, 2019).

Además, ofrece funcionalidades para el almacenamiento seguro de datos, integraciones con otros servicios y la posibilidad de crear aplicaciones y servicios personalizados utilizando su API (Tejada Gutierrez, 2019).

2.20. Token en IOT

Utilizado para autenticar y autorizar dispositivos y servicios en una red o plataforma IoT que funciona como una especie de "credencial" que permite que los dispositivos se comuniquen e interactúen de manera segura (Macías Valdivieso, 2022).

Un token en IoT puede ser una cadena de caracteres alfanuméricos generada por un sistema de autenticación o puede basarse en un estándar de seguridad como JSON Web Token (JWT). Estos tokens se utilizan para validar la identidad de un dispositivo o servicio, evitando accesos no autorizados y garantizando la integridad de la comunicación (Macías Valdivieso, 2022).

3. MARCO METODOLÓGICO

3.1. Sistema propuesto para el módulo

Para la implementación del sistema del módulo se utilizó los microcontroladores esp32 para la interacción con la plataforma iot y el Atmel 32UA para el manejo adquisición y reacondicionar los valores de los sensores en conjunto la activación de actuadores como se muestra en la Figura 17.

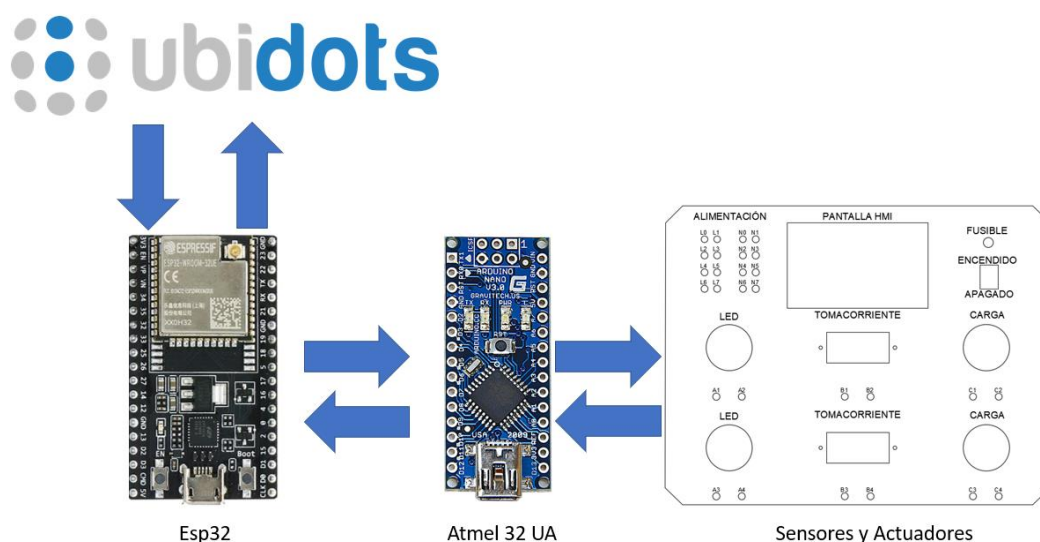


Figura 17: Comunicación entre microcontroladores y proyecto

En la aplicación respecto a la activación de los led, cargas y puntos eléctrico se optó por el uso de relé de estado sólido permitiendo que sea más eficaz al momento de su funcionamiento como se muestra en la Figura 18.



Figura 18: Control para activación de relés sólidos

Para la supervisión se instaló una pantalla de 7 pulgadas de la marca Nextion que utiliza comunicación serial con el atmel32Ua como se muestra en la Figura 19.

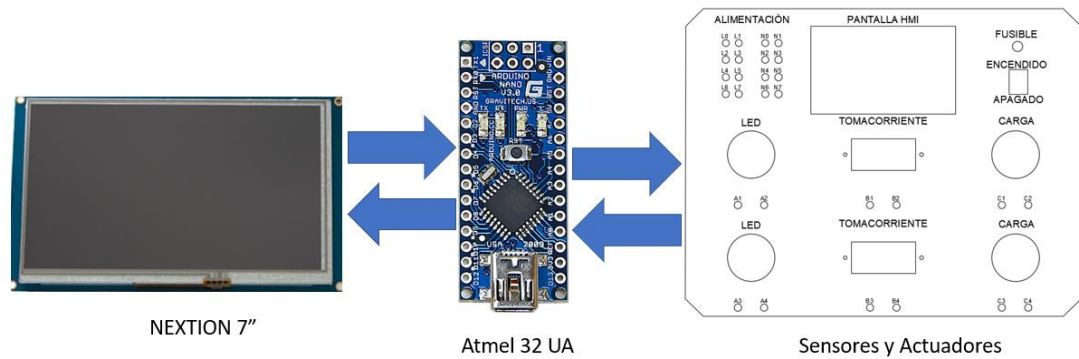


Figura 19: Comunicación entre embebido y pantalla

3.2. Diseño del proyecto

Para el diseño del módulo se optó por una maleta de aluminio de dimensiones 48x37 cm de exterior e interior unos 25 cm como se muestra en la Figura 20.



Figura 20: Maleta de aluminio

Se instalará una placa donde estarán los borneros y elementos como se muestra en la Figura 21.



Figura 21.: Placa del proyecto

3.3. Planos del proyecto

En la Figura 22 se detalla las dimensiones de la placa que contendrá los elementos como los leds, puntos eléctricos, boquillas para cargas, pantalla nextion y un conjunto de bornes de 4mm para la conexión entre cada etapa del módulo para más detalle revisará el Anexo A – Planos de maleta.

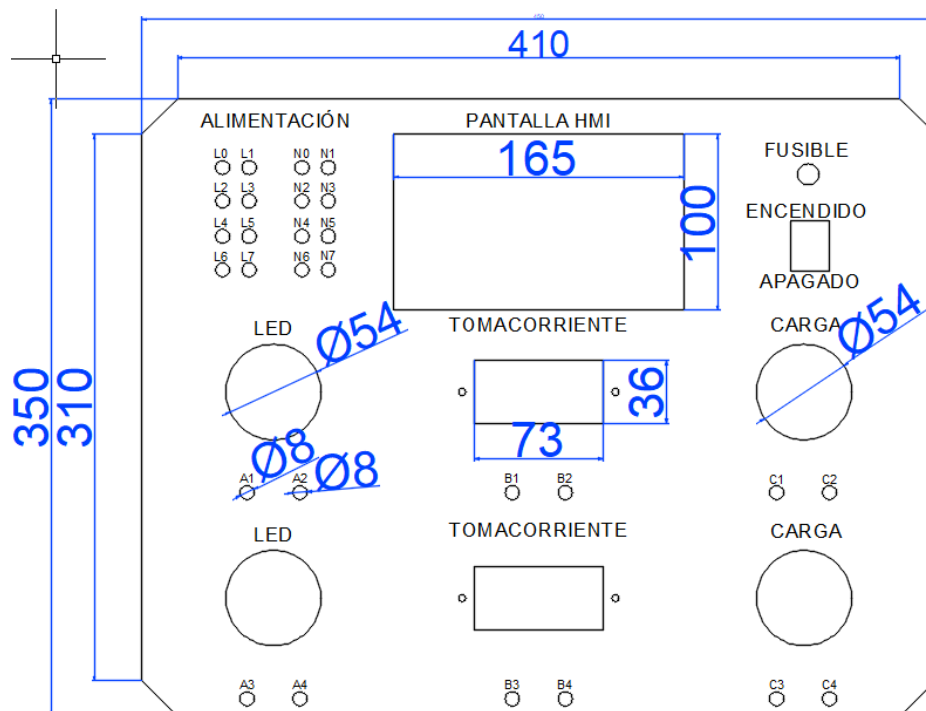


Figura 22.: Dimensiones de placa

3.4. Flujograma y funcionamiento del proyecto

En la maleta didáctica al encender se debe seleccionar en la pantalla NEXTION la práctica que se desea realizar siguiendo el flujo que se muestra en la Figura 23.

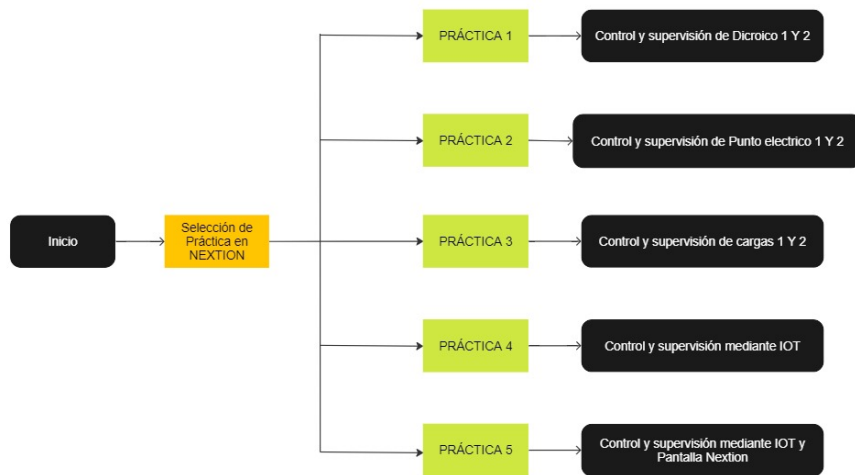


Figura 23:. Selección de práctica

En la práctica 1 se tiene como objetivo el encendido del led mediante la pantalla nextion y visualizar el dato del consumo de cada led como se muestra el mapa de la figura 24.

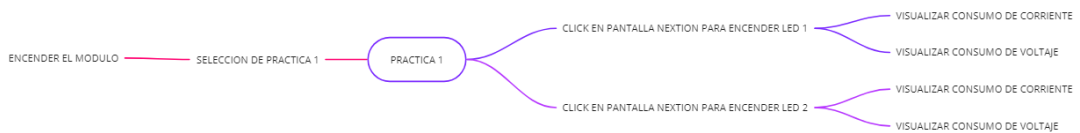


Figura 24:. Funcionamiento de práctica 1

En la práctica 2 se tiene como objetivo el encendido de los puntos eléctrico y verificar mediante la pantalla nextion y visualizar el dato del consumo de cada punto eléctrico como se muestra el mapa de la figura 25.



Figura 25:. Funcionamiento de práctica 2

En la práctica 3 se tiene como objetivo el encendido de cargas o foco incandescentes de 100 w y verificar mediante la pantalla nextion y visualizar el dato del consumo de cada punto eléctrico como se muestra el mapa de la figura 26.

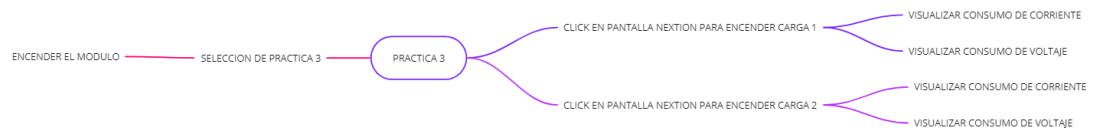


Figura 26:. Funcionamiento de práctica 3

En la práctica 4 se tiene como objetivo el encendido desde la plataforma IoT para los leds, puntos eléctricos, cargas y verificar mediante la pantalla nextion y visualizar el dato del consumo de cada elemento como se muestra el mapa de la figura 27.

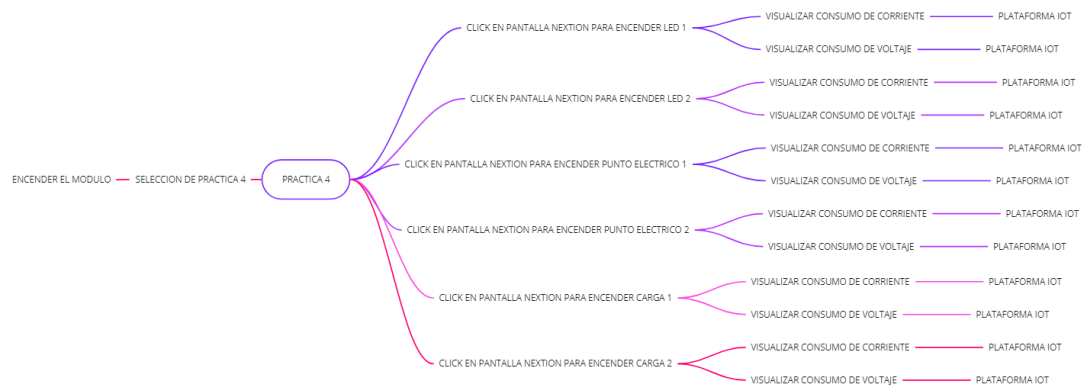


Figura 27:. Funcionamiento de práctica 4

En la práctica 5 se tiene como objetivo el encendido desde la plataforma IoT y desde la pantalla nextion para los leds, puntos eléctricos, cargas y verificar mediante la pantalla nextion y visualizar el dato del consumo de cada elemento como se muestra el mapa de la Figura 28.

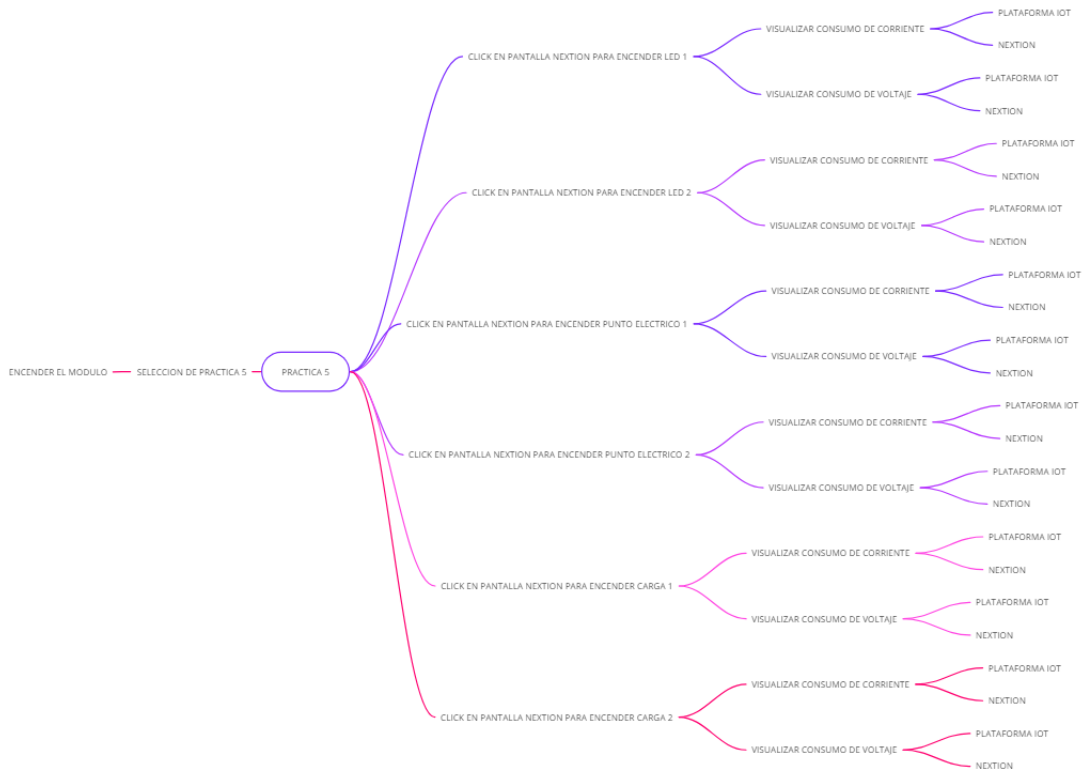


Figura 28:. Funcionamiento de práctica 5

3.5. Tarjetas pcb del modulo

Para la implementación del banco de pruebas se utilizó dos microcontroladores, en la Figura 29 se muestra el diagrama esquemático de las conexiones del microcontrolador que es para la adquisición de los valores de corriente mediante sensores analógicos de corriente y la comunicación con la pantalla nextion.

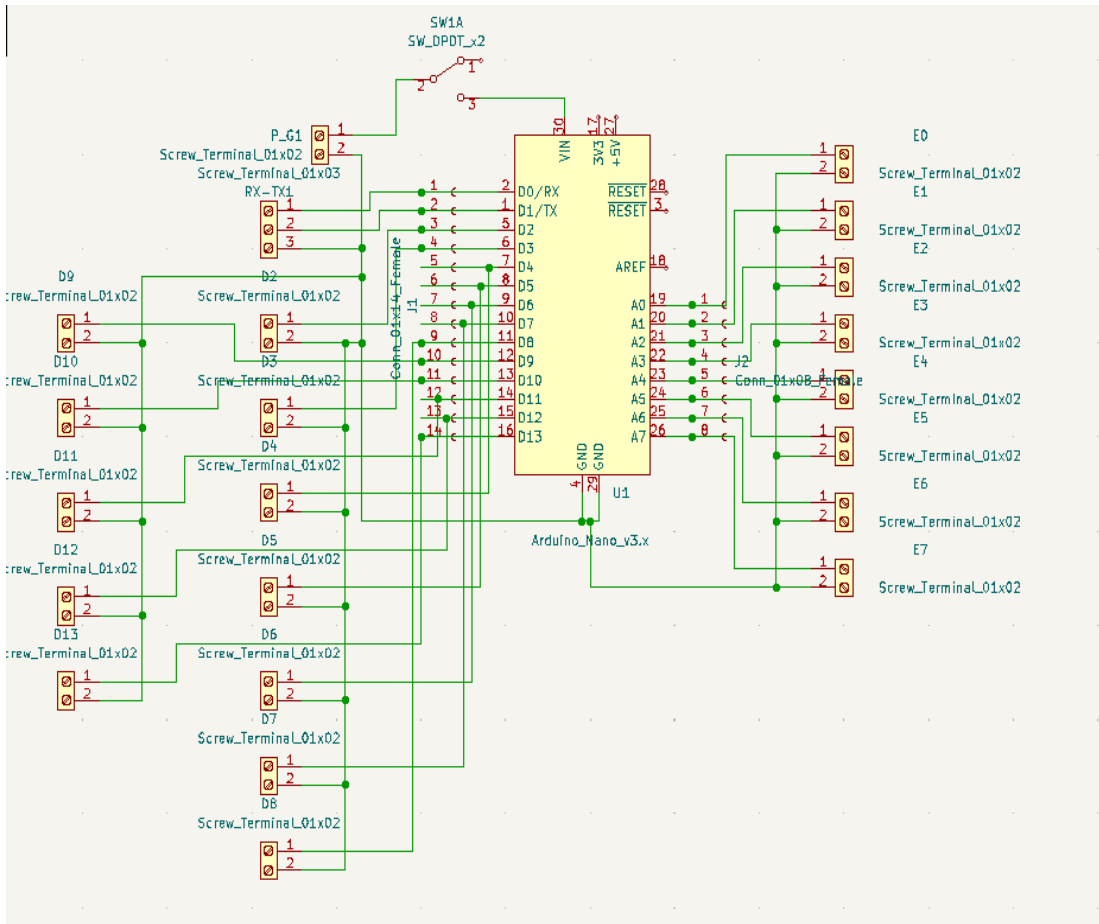


Figura 29: Esquemático de microcontrolador para sensores y pantalla

En la Figura 30 se muestra el microcontrolador que realiza el enlace a la plataforma IoT Ubidots mediante un protocolo TCP/IP y consulta los datos de la plataforma para la activación de los relés desde el IoT o la pantalla nextion.

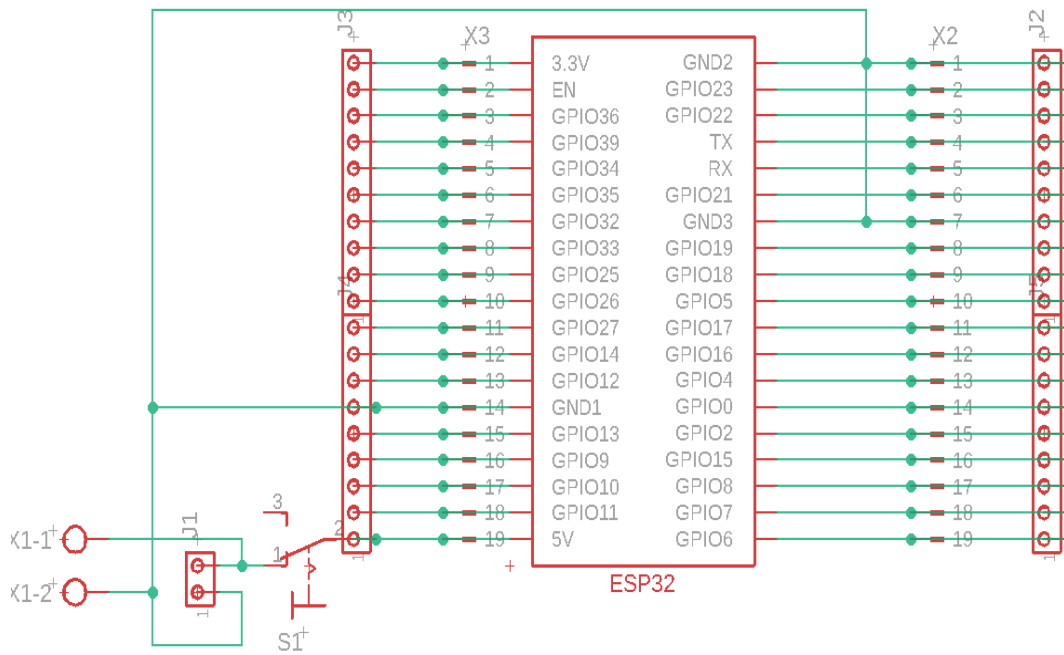


Figura 30: Esquemático de microcontrolador para enlace con plataforma

En la Figura 31 se detalla la implementación física dentro del banco de pruebas y la distribución de los elementos en una base de acrílico de 4 mm para evitar conductividad al trabajar en voltaje alterno.

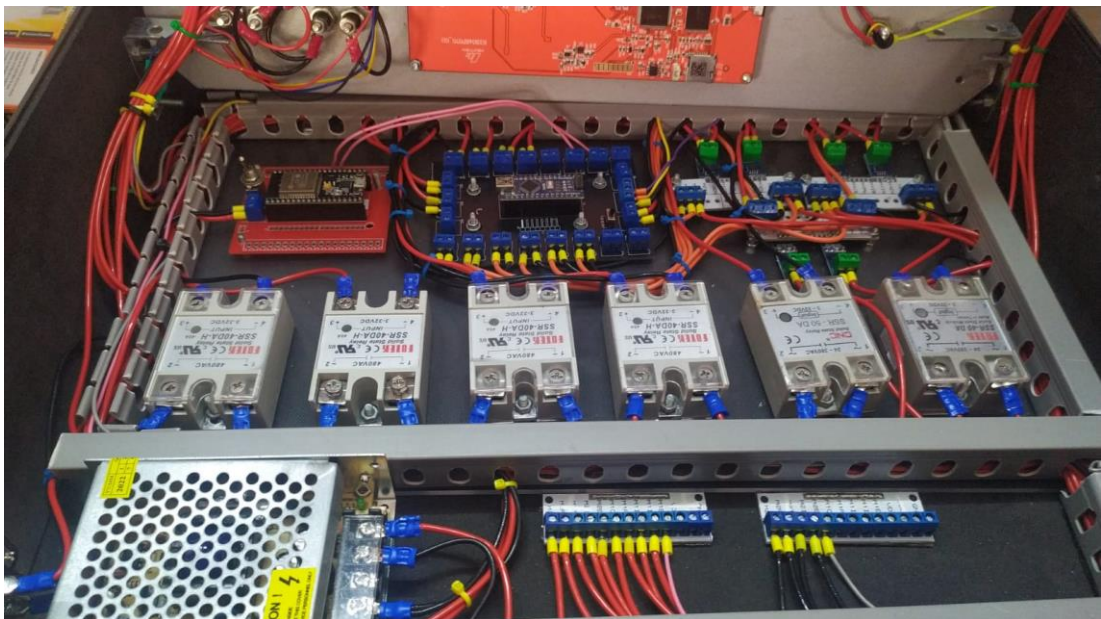


Figura 31: Vista de implementación parte interna del banco

3.6. Módulos y sensores del proyecto

Para la lectura de la corriente se utiliza un sensor ACS 712 que se muestra en la Figura 32 el cual consta de un efecto electromagnético el cual sirve para medir el voltaje proporcional a la corriente dependiendo la aplicación

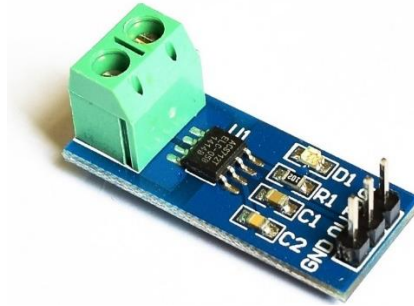


Figura 32: Sensor de corriente Acs712 (naylampmechatronics, 2022)

El sensor proporciona un valor inicial de 2.5 voltios cuando no hay corriente (0A), y a partir de ese punto, su salida de voltaje aumenta de manera proporcional según su sensibilidad. Esto significa que existe una relación lineal entre el voltaje de salida del sensor y la corriente.

En una gráfica de Voltaje versus Corriente, esta relación se representa mediante una línea recta, donde la pendiente de la recta es igual a la sensibilidad del sensor y la intersección en el eje Y corresponde a 2.5 voltios. De acuerdo con la ecuación 1 donde V es el voltaje del sensor y sensibilidad es el valor de acuerdo con el tipo del sensor.

$$I = \frac{V-2.5}{Sensibilidad} \quad (1)$$

Para la activación de los leds se utilizan relés de estado son los encargados de controlar el flujo de corriente hacia una carga eléctrica, funciona mediante la aplicación de una señal de control (generalmente una baja corriente y tensión) que activa los componentes de estado sólido dentro del SSR para abrir o cerrar el circuito principal como se ilustra en el Figura 33, permitiendo o interrumpiendo el flujo de corriente hacia la carga.

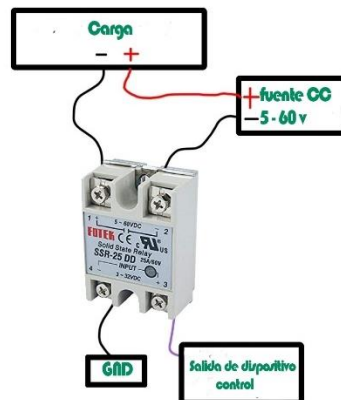


Figura 33:. Relé solido (ArduSpain, 2022)

La tarjeta de adquisición de datos es un Atmel 32 el cual viene integrado en un Arduino nano que se muestra en la Figura 34 permitiendo su programación mediante un cable USB, teniendo un ide de programación en el cual se pueden utilizar librerías para el manejo de la pantalla nextion.

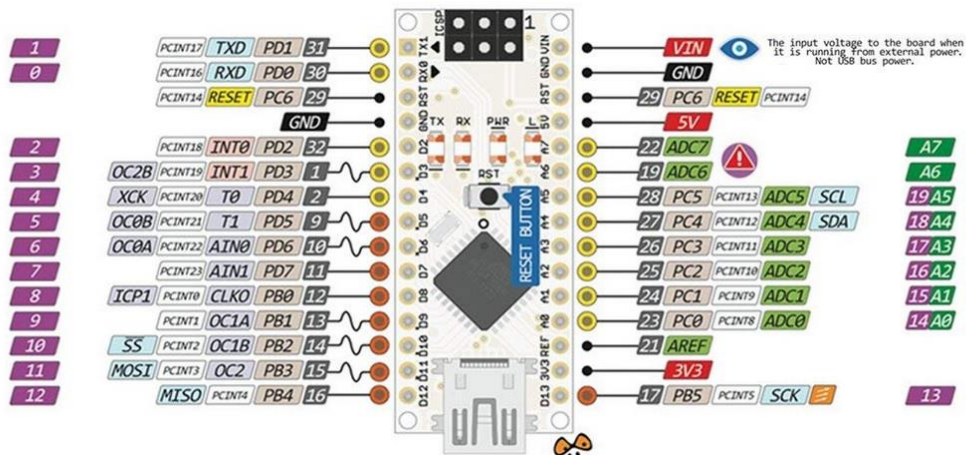


Figura 34:. Arduino nano con Atmel 32ua (descubrearduino, 2022)

La tarjeta para el enlace con la plataforma y consultar los datos en la misma es la esp32WROOM que se muestra en la Figura 35 que posee una conexión inalámbrica, en conjunto con un entorno de programación que integra el uso de librerías tales como UBIDOTS.

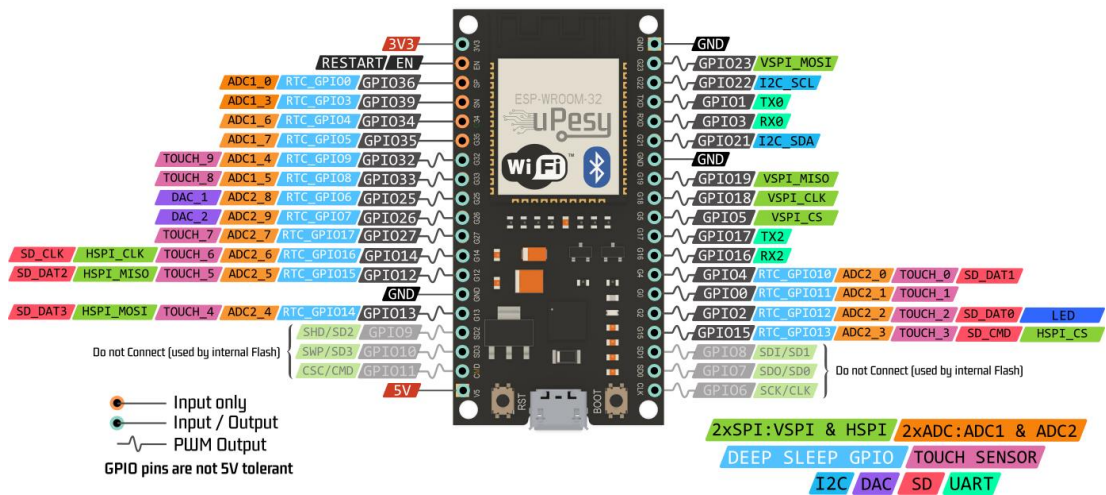


Figura 35:. Esp32 WROOM (upesy, 2022)

Para la visualización e interacción en el módulo se optó por la pantalla NEXTION modelo NX8048K070_011C que se conecta al microcontrolador para su interacción como se muestra en la Figura 36.



Figura 36:. Nextion y Arduino nano

3.7. Diagramas de conexiones internos del módulo

En la Figura 37 se detalla el diagrama de conexiones del módulo integrando los elementos como el embebido de adquisición, tarjeta de enlace, sensores y pantalla.

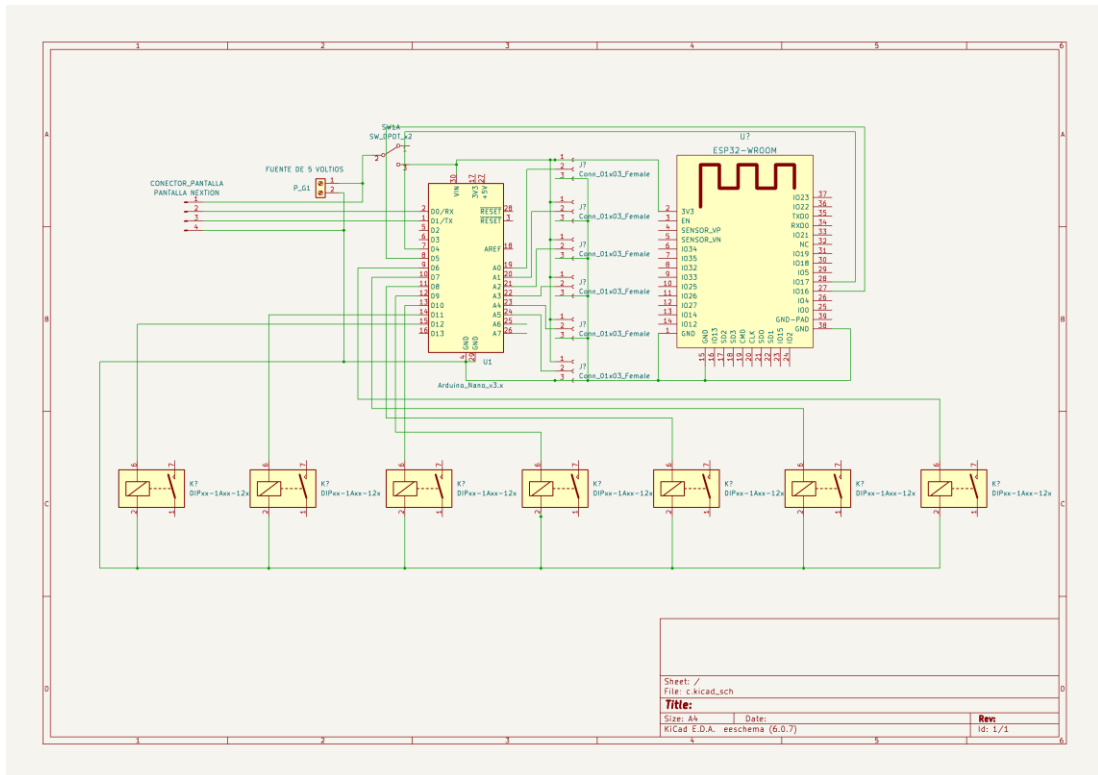


Figura 37.: Esquema del módulo

En la Figura 38 se muestra la implementación del módulo señalando que es cada elemento.

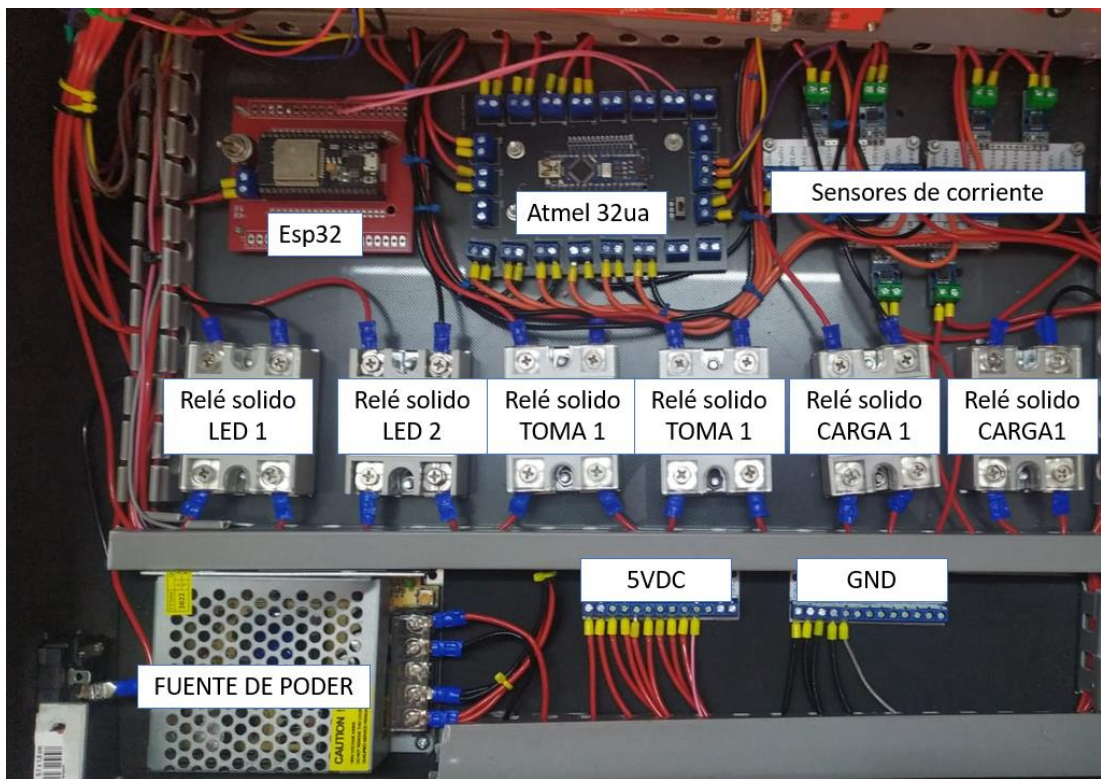


Figura 38:. Componentes del modulo

3.8. Enlace de comunicación entre NEXTION y embebidos

Este código es una implementación para controlar un sistema de medición y monitoreo de corriente eléctrica utilizando un microcontrolador y una pantalla táctil Nextion. El sistema mide la corriente en varios sensores y muestra los resultados en la pantalla táctil. Además, permite encender o apagar salidas en función de la corriente medida.

A continuación, se detalla el código en secciones:

Como se observa Figura 39 se incluyen las librerías necesarias, que son la librería "EasyNextionLibrary" para la comunicación con la pantalla Nextion y la librería "SoftwareSerial" para establecer una comunicación serial por software.

```
#include "EasyNextionLibrary.h" //  
#include <SoftwareSerial.h>
```

Figura 39:. Programa de Atmel 32 – Declaración de librerías

En la Figura 40 se crea un objeto myNex de la clase EasyNex para manejar la comunicación con la pantalla Nextion. También se crea un objeto portEnlace de la clase SoftwareSerial para la comunicación serial con otro dispositivo.

```
EasyNex myNex (Serial) ;  
SoftwareSerial portEnlace (5, 4) ;
```

Figura 40:. Programa de Atmel 32 – Declaración de objetos

En la Figura 41 se declaran las diversas variables para configurar el funcionamiento del sistema. Entre ellas se encuentran variables para almacenar la sensibilidad del sensor, un offset para el ruido, los valores de las salidas, los estados de los LEDs, y variables para almacenar los valores de corriente y potencia medidos.

```

float Sensibilidad = 0.139; //sensibilidad en V/A para nuestro sensor
float offset = 0.5; // Equivale a la amplitud del ruido
int salidaValue_1 = 12, salidaValue_2 = 11, salidaValue_3 = 10, salidaValue_4 = 9, salidaValue_5 = 8, salidaValue_6 = 7;
int ledState_1 = LOW, ledState_2 = LOW, ledState_3 = LOW, ledState_4 = LOW, ledState_5 = LOW, ledState_6 = LOW;
float voltajeSensor_1; //sensibilidad en V/A para nuestro sensor
float Ip_1 = 0, Ip_2 = 0, Ip_3 = 0, Ip_4 = 0, Ip_5 = 0, Ip_6 = 0; //obtenemos la corriente pico
float Irms_1 = 0, Irms_2 = 0, Irms_3 = 0, Irms_4 = 0, Irms_5 = 0, Irms_6 = 0; //Intensidad RMS = Ipico/(2^1/2)
float P_1 = 0, P_2 = 0, P_3 = 0, P_4 = 0, P_5 = 0, P_6 = 0;
float t_1 = 0, t_2 = 0;

```

Figura 41: Programa de Atmel 32 – Declaración de variables

En la función setup() Figura 42, se configuran las salidas de las salidas LED y se establece su estado inicial en LOW. Se inicia la comunicación con la pantalla Nextion y con el otro dispositivo mediante el puerto serial.

```

void setup()
{
  // analogReference(INTERNAL);
  pinMode(salidaValue_1, OUTPUT);
  pinMode(salidaValue_2, OUTPUT);
  pinMode(salidaValue_3, OUTPUT);
  pinMode(salidaValue_4, OUTPUT);
  pinMode(salidaValue_5, OUTPUT);
  pinMode(salidaValue_6, OUTPUT);
  digitalWrite(salidaValue_1, ledState_1);
  digitalWrite(salidaValue_2, ledState_2);
  digitalWrite(salidaValue_3, ledState_3);
  digitalWrite(salidaValue_4, ledState_4);
  digitalWrite(salidaValue_5, ledState_5);
  digitalWrite(salidaValue_6, ledState_6);
  myNex.begin(9600);
  portEnlace.begin(9600);
}

```

Figura 42: Programa de Atmel 32 – método configuración

El bucle loop() realiza diversas tareas en cada iteración. Primero, llama a la función get_corriente() para medir la corriente en los sensores y calcular los valores de corriente pico, intensidad RMS y potencia. Luego, llama a la función publicar_iot() para enviar los valores de corriente y estado de las salidas a otro dispositivo a través del puerto serial, como se describe en la Figura 43.

A continuación, llama a la función consultar_iot() para recibir comandos de encender o apagar las salidas desde el otro dispositivo.


```

void loop() {
    get_corriente();
    delay(1);
    publicar_iot();
    delay(1);
    consultar_iot();
    delay(1);
    nextion_publicar();
    delay(1);
    nextion_consultar();
    delay(1);
}

```

Figura 43: Programa de Atmel 32 – método principal

Las funciones `nextion_publicar()` y `nextion_consultar()`, se encargan de enviar y recibir datos desde la pantalla Nextion. Se utilizan comandos específicos de la pantalla para mostrar los valores de corriente y potencia en la interfaz gráfica, así como para recibir comandos de encender o apagar las salidas LED mediante toques en la pantalla.

```

void nextion_consultar()
{
    int x = myNex.readNumber("n12.val");
    switch (x)
    {
        case 32:
            ledState_6 = !ledState_6;
            digitalWrite(salidaValue_6, ledState_6);
            break;
        case 16:
            ledState_5 = !ledState_5;
            digitalWrite(salidaValue_5, ledState_5);
            break;
        case 8:
            ledState_4 = !ledState_4;
            digitalWrite(salidaValue_4, ledState_4);
            break;
        case 4:
            ledState_3 = !ledState_3;
            digitalWrite(salidaValue_3, ledState_3);
            break;
        case 2:
            ledState_2 = !ledState_2;
            digitalWrite(salidaValue_2, ledState_2);
            break;
        case 1:
            ledState_1 = !ledState_1;
            digitalWrite(salidaValue_1, ledState_1);
            break;
    }
}

```

Figura 44: Programa de Atmel 32 – método nextion consultar

```

void nextion_publicar()
{
    myNex.writeNum("x0.val", 110.00 * ledState_1 * 1000);
    myNex.writeNum("x1.val", Ip_1 * ledState_1 * 1000);
    myNex.writeNum("x2.val", (float)P_1 * ledState_1 * 1000);
    myNex.writeNum("x3.val", (float)110.00 * ledState_2 * 1000);
    myNex.writeNum("x4.val", (float)Ip_2 * ledState_2 * 1000);
    myNex.writeNum("x5.val", (float)P_2 * ledState_2 * 1000);
    myNex.writeNum("x6.val", (float)110.00 * ledState_3 * 1000);
    myNex.writeNum("x7.val", (float)Ip_3 * ledState_3 * 1000);
    myNex.writeNum("x8.val", (float)P_3 * ledState_3 * 1000);
    myNex.writeNum("x9.val", (float)110.00 * ledState_4 * 1000);
    myNex.writeNum("x10.val", (float)Ip_4 * ledState_4 * 1000);
    myNex.writeNum("x11.val", (float)P_4 * ledState_4 * 1000);
    myNex.writeNum("x12.val", (float)110.00 * ledState_5 * 1000);
    myNex.writeNum("x13.val", (float)Ip_5 * ledState_5 * 1000);
    myNex.writeNum("x14.val", (float)P_5 * ledState_5 * 1000);
    myNex.writeNum("x15.val", (float)110.00 * ledState_6 * 1000);
    myNex.writeNum("x16.val", (float)Ip_6 * ledState_6 * 1000);
    myNex.writeNum("x17.val", (float)P_6 * ledState_6 * 1000);
    t_1 = Ip_1 + Ip_2 + Ip_3 + Ip_4 + Ip_5 + Ip_6;
    t_2 = P_1 + P_2 + P_3 + P_4 + P_5 + P_6;
    myNex.writeNum("x22.val", (float)t_1 * 1000);
    myNex.writeNum("x23.val", (float)t_2 * 1000);
}

```

Figura 45: Programa de Atmel 32 – método nextion publicar

Esta función `get_corriente()` mide la corriente en los sensores durante medio segundo, utilizando las lecturas analógicas del sensor. Se calculan los valores de corriente pico, intensidad RMS y potencia para cada sensor.

```

void get_corriente()
{
    float voltajeSensor_1, voltajeSensor_2, voltajeSensor_3, voltajeSensor_4, voltajeSensor_5, voltajeSensor_6;
    float corriente_1 = 0, corriente_2 = 0, corriente_3 = 0, corriente_4 = 0, corriente_5 = 0, corriente_6 = 0;
    long tiempo = millis(); float Imax_1 = 0, Imax_2 = 0, Imax_3 = 0, Imax_4 = 0, Imax_5 = 0, Imax_6 = 0;
    float Imin_1 = 0, Imin_2 = 0, Imin_3 = 0, Imin_4 = 0, Imin_5 = 0, Imin_6 = 0;
    while (millis() - tiempo < 500) //realizamos mediciones durante 0.5 segundos
    {
        voltajeSensor_1 = analogRead(A0) * (5.0 / 1023.0); //lectura del sensor      voltajeSensor_2 = analogRead(A1) * (5.0 / 1023.0); //lectura del sensor
        voltajeSensor_3 = analogRead(A2) * (5.0 / 1023.0); //lectura del sensor      voltajeSensor_4 = analogRead(A3) * (5.0 / 1023.0); //lectura del sensor
        voltajeSensor_5 = analogRead(A4) * (5.0 / 1023.0); //lectura del sensor      voltajeSensor_6 = analogRead(A5) * (5.0 / 1023.0); //lectura del sensor
        corriente_1 = 0.9 * corriente_1 + 0.1 * ((voltajeSensor_1 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        corriente_2 = 0.9 * corriente_2 + 0.1 * ((voltajeSensor_2 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        corriente_3 = 0.9 * corriente_3 + 0.1 * ((voltajeSensor_3 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        corriente_4 = 0.9 * corriente_4 + 0.1 * ((voltajeSensor_4 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        corriente_5 = 0.9 * corriente_5 + 0.1 * ((voltajeSensor_5 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        corriente_6 = 0.9 * corriente_6 + 0.1 * ((voltajeSensor_6 - 3.527) / Sensibilidad); //Ecuación para obtener la corriente
        if (corriente_1 > Imax_1) Imax_1 = corriente_1;      if (corriente_2 > Imax_2) Imax_2 = corriente_2;
        if (corriente_3 > Imax_3) Imax_3 = corriente_3;      if (corriente_4 > Imax_4) Imax_4 = corriente_4;
        if (corriente_5 > Imax_5) Imax_5 = corriente_5;      if (corriente_6 > Imax_6) Imax_6 = corriente_6;
        if (corriente_1 < Imin_1) Imin_1 = corriente_1;      if (corriente_2 < Imin_2) Imin_2 = corriente_2;
        if (corriente_3 < Imin_3) Imin_3 = corriente_3;      if (corriente_4 < Imin_4) Imin_4 = corriente_4;
        if (corriente_5 < Imin_5) Imin_5 = corriente_5;      if (corriente_6 < Imin_6) Imin_6 = corriente_6;
        Ip_1 = ((Imax_1 - Imin_1) / 2) - offset; Ip_2 = ((Imax_2 - Imin_2) / 2) - offset;
        Ip_3 = ((Imax_3 - Imin_3) / 2) - offset; Ip_4 = ((Imax_4 - Imin_4) / 2) - offset;
        Ip_5 = ((Imax_5 - Imin_5) / 2) - offset; Ip_6 = ((Imax_6 - Imin_6) / 2) - offset;
        Ip_1 = Ip_1 * ledState_1; Ip_2 = Ip_2 * ledState_2; Ip_3 = Ip_3 * ledState_3; Ip_4 = Ip_4 * ledState_4; Ip_5 = Ip_5 * ledState_5; Ip_6 = Ip_6 * ledState_6;
        Irms_1 = Ip_1 * 0.707; Irms_2 = Ip_2 * 0.707; Irms_3 = Ip_3 * 0.707; Irms_4 = Ip_4 * 0.707; Irms_5 = Ip_5 * 0.707; Irms_6 = Ip_6 * 0.707; //Intensidad RMS = I pico/(2*1/2)
        P_1 = Irms_1 * 110.0; P_2 = Irms_2 * 110.0; P_3 = Irms_3 * 110.0; P_4 = Irms_4 * 110.0; P_5 = Irms_5 * 110.0; P_6 = Irms_6 * 110.0; // P=IV watts
    }
}

```

Figura 46: Programa de Atmel 32 – método obtener corriente

Funciones `publicar_iot()` y `consultar_iot()`: Estas funciones se encargan de enviar y recibir comandos a través del puerto serial con otro dispositivo. Los comandos enviados y recibidos son representados por letras (A, a, B, b, C, c, etc.) que corresponden a encender o apagar las salidas LED.

```

void publicar_iot() {
    portEnlace.print("A1:"); portEnlace.print(Ip_1); portEnlace.print(",");
    portEnlace.print("A2:"); portEnlace.print(Ip_2); portEnlace.print(",");
    portEnlace.print("A3:"); portEnlace.print(Ip_3); portEnlace.print(",");
    portEnlace.print("A4:"); portEnlace.print(Ip_4); portEnlace.print(",");
    portEnlace.print("A5:"); portEnlace.print(Ip_5); portEnlace.print(",");
    portEnlace.print("A6:"); portEnlace.print(Ip_6); portEnlace.print(",");
    portEnlace.print("A7:"); portEnlace.print(110 * ledState_1); portEnlace.print(",");
    portEnlace.print("A8:"); portEnlace.print(110 * ledState_2); portEnlace.print(",");
    portEnlace.print("A9:"); portEnlace.print(110.00 * ledState_3); portEnlace.print(",");
    portEnlace.print("A10:"); portEnlace.print(110.00 * ledState_4); portEnlace.print(",");
    portEnlace.print("A11:"); portEnlace.print(110.00 * ledState_5); portEnlace.print(",");
    portEnlace.print("A12:"); portEnlace.println(110.00 * ledState_6);
}

```

Figura 47:. Programa de Atmel 32 – método publicar IoT

```

void consultar_iot() {
    char comando;
    while (portEnlace.available() > 0) {
        comando = portEnlace.read();
        switch (comando) {
            case 'A': ledState_1 = LOW; digitalWrite(salidaValue_1, ledState_1); break;
            case 'a': ledState_1 = HIGH; digitalWrite(salidaValue_1, ledState_1); break;
            case 'B': ledState_2 = LOW; digitalWrite(salidaValue_2, ledState_2); break;
            case 'b': ledState_2 = HIGH; digitalWrite(salidaValue_2, ledState_2); break;
            case 'C': ledState_3 = LOW; digitalWrite(salidaValue_3, ledState_3); break;
            case 'c': ledState_3 = HIGH; digitalWrite(salidaValue_3, ledState_3); break;
            case 'D': ledState_4 = LOW; digitalWrite(salidaValue_4, ledState_4); break;
            case 'd': ledState_4 = HIGH; digitalWrite(salidaValue_4, ledState_4); break;
            case 'E': ledState_5 = LOW; digitalWrite(salidaValue_5, ledState_5); break;
            case 'e': ledState_5 = HIGH; digitalWrite(salidaValue_5, ledState_5); break;
            case 'F': ledState_6 = LOW; digitalWrite(salidaValue_6, ledState_6); break;
            case 'f': ledState_6 = HIGH; digitalWrite(salidaValue_6, ledState_6); break;
        }
    }
}

```

Figura 48:. Programa de Atmel 32 – método consultar IoT

3.9. Enlace de comunicación entre embebido de adquisición y embebido de enlace a plataforma

Este código es un programa para un dispositivo basado en ESP32 que se conecta a la plataforma Ubidots para publicar y recibir datos en tiempo real. Ubidots es una plataforma IoT (Internet of Things) que permite almacenar, visualizar y analizar datos de sensores en la nube.

A continuación, se detalla el código en secciones:

El código comienza incluyendo las bibliotecas necesarias. Entre ellas están UbidotsEsp32Mqtt.h para conectarse a Ubidots mediante MQTT (Message

Queuing Telemetry Transport) y HardwareSerial.h para comunicarse con el embebido a través del puerto serie hardware.

```
#include "UbidotsEsp32Mqtt.h"
#include <HardwareSerial.h>
HardwareSerial mySerial(2);
```

Figura 49:. Programa en esp32 – declaración de librerías

A continuación en la Figura 50, se definen varias constantes que se utilizan en el código, como el token de autenticación de Ubidots, las credenciales de Wi-Fi, las etiquetas de los dispositivos y variables de Ubidots, la frecuencia de publicación y un arreglo de sensores.

```
const char *UBIDOTS_TOKEN = "BBFF-CV3fPVTzSXropGZATRk1SvcsxsXHLY"
const char *WIFI_SSID = "Red_maleta"; // Put here
const char *WIFI_PASS = "ElectronicaUPS"; // Put h
const char *PUBLISH_DEVICE_LABEL = "maleta"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_1 = "ip_1"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_2 = "ip_2"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_3 = "ip_3"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_4 = "ip_4"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_5 = "ip_5"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_6 = "ip_6"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_7 = "v_1"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_8 = "v_2"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_9 = "v_3"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_10 = "v_4"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_11 = "v_5"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_12 = "v_6"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_13 = "p_1"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_14 = "p_2"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_15 = "p_3"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_16 = "p_4"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_17 = "p_5"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_18 = "p_6"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_19 = "t_1"; // Put here your
const char *PUBLISH_VARIABLE_LABEL_20 = "t_2"; // Put here your
const char *SUBSCRIBE_DEVICE_LABEL = "maleta"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_1 = "bt_1"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_2 = "bt_2"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_3 = "bt_3"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_4 = "bt_4"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_5 = "bt_5"; // Replace with
const char *SUBSCRIBE_VARIABLE_LABEL_6 = "bt_6"; // Replace with
```

Figura 50:. Programa en esp32 – declaración de variables

Esta función callback se invoca cada vez que llega un mensaje a través de MQTT desde Ubidots. Analiza el mensaje para obtener la etiqueta y el valor y realiza acciones en función de los datos recibidos.

```

void callback(char *topic, byte *payload, unsigned int length)
{
    String mensaje = "Message arrived [";
    mensaje += topic;
    mensaje += "] ";
    for (int i = 0; i < length; i++) {
        mensaje += (char)payload[i];
    }
    mensaje += "\n";
    int etiqueta = obtenerEtiqueta(mensaje).toInt();
    int valor = obtenerValor(mensaje).toInt();
    Serial.println("Etiqueta: " ); Serial.println(etiqueta);
    Serial.println(valor);
    switch (etiqueta) {
        case 1:      if (valor == 0.0) {mySerial.println("A");}
                    else {      mySerial.println("a");      }      break;
        case 2:      if (valor == 0.0) {mySerial.println("B");}
                    else {      mySerial.println("b");      }      break;
        case 3:      if (valor == 0.0) {mySerial.println("C");}
                    else {      mySerial.println("c");      }      break;
        case 4:      if (valor == 0.0) {      mySerial.println("D");      }
                    else {      mySerial.println("d");      }      break;
        case 5:      if (valor == 0.0) {      mySerial.println("E");      }
                    else {      mySerial.println("e");      }      break;
        case 6:      if (valor == 0.0) {      mySerial.println("F");      }
                    else {      mySerial.println("f");      }      break;
    }
}

```

Figura 51: Programa en esp32 – método callback

En la función setup, se inicia la comunicación serie, se establece la conexión Wi-Fi y se conecta a Ubidots. Luego, se suscribe a las últimas actualizaciones de las variables específicas que se definieron previamente.

```

void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    mySerial.begin(9600);
    // ubidots.setDebug(true); // uncomment this to make debug messages available
    ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
    ubidots.setCallback(callback);
    ubidots.setup();
    ubidots.reconnect();
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_1); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_2); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_3); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_4); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_5); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_6); // Insert the device and variable's Labels, respectively

    timer = millis();
}

```

Figura 52: Programa en esp32 – método de configuración

En la función loop, el dispositivo entra en un bucle infinito donde ejecuta dos funciones principales.

```

void loop()
{
  datos();
  principal();
}

```

Figura 53:. Programa en esp32 – método de loop

Esta función datos se encarga de leer los datos desde el embebido a través del puerto serie y almacena los valores de los sensores en el arreglo sensores. Esta función principal se encarga de publicar los datos almacenados en el arreglo sensores a Ubidots en intervalos regulares. También calcula la intensidad y potencia RMS basada en los datos de los sensores y los publica.

```

void principal()
{
  if (!ubidots.connected())
  {
    ubidots.reconnect();
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_1); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_2); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_3); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_4); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_5); // Insert the device and variable's Labels, respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL, SUBSCRIBE_VARIABLE_LABEL_6); // Insert the device and variable's Labels, respectively
  }
  if (abs(millis() - timer) > PUBLISH_FREQUENCY) // triggers the routine every 5 seconds
  {
    float value = analogRead(analogPin);
    ubidots.add(PUBLISH_VARIABLE_LABEL_1, sensores[0]);    ubidots.add(PUBLISH_VARIABLE_LABEL_2, sensores[1]); // Variable Corriente 2
    ubidots.add(PUBLISH_VARIABLE_LABEL_3, sensores[2]);    ubidots.add(PUBLISH_VARIABLE_LABEL_4, sensores[3]); // Variable Corriente 4
    ubidots.add(PUBLISH_VARIABLE_LABEL_5, sensores[4]);    ubidots.add(PUBLISH_VARIABLE_LABEL_6, sensores[5]); // Variable Corriente 6
    ubidots.publish(PUBLISH_DEVICE_LABEL);
    ubidots.add(PUBLISH_VARIABLE_LABEL_7, sensores[6]);    ubidots.add(PUBLISH_VARIABLE_LABEL_8, sensores[7]); // Variable Voltaje 2
    ubidots.add(PUBLISH_VARIABLE_LABEL_9, sensores[8]);    ubidots.add(PUBLISH_VARIABLE_LABEL_10, sensores[9]); // Variable Voltaje 4
    ubidots.add(PUBLISH_VARIABLE_LABEL_11, sensores[10]);  ubidots.add(PUBLISH_VARIABLE_LABEL_12, sensores[11]); // Variable Voltaje 6
    ubidots.publish(PUBLISH_DEVICE_LABEL);
    Irms_1 = sensores[0] * 0.707;    Irms_2 = sensores[1] * 0.707;    Irms_3 = sensores[2] * 0.707;    Irms_4 = sensores[3] * 0.707; //Intensidad RMS = I pico/(2^1/2)
    Irms_5 = sensores[4] * 0.707;    Irms_6 = sensores[5] * 0.707; //Intensidad RMS = I pico/(2^1/2)
    P_1 = Irms_1 * sensores[6];    P_2 = Irms_2 * sensores[7];    P_3 = Irms_3 * sensores[8];    P_4 = Irms_4 * sensores[9]; // P=IV watts
    P_5 = Irms_5 * sensores[10];    P_6 = Irms_6 * sensores[11]; // P=IV watts
    ubidots.add(PUBLISH_VARIABLE_LABEL_13, P_1);    ubidots.add(PUBLISH_VARIABLE_LABEL_14, P_2);    ubidots.add(PUBLISH_VARIABLE_LABEL_15, P_3); // Variable Potencia 3
    ubidots.add(PUBLISH_VARIABLE_LABEL_16, P_4);    ubidots.add(PUBLISH_VARIABLE_LABEL_17, P_5);    ubidots.add(PUBLISH_VARIABLE_LABEL_18, P_6); // Variable Potencia 6
    ubidots.publish(PUBLISH_DEVICE_LABEL);
    t_1= sensores[0]+ sensores[1]+ sensores[2]+ sensores[3]+ sensores[4]+ sensores[5];
    t_2=P_1+P_2+P_3+P_4+P_5+P_6;
    ubidots.add(PUBLISH_VARIABLE_LABEL_19, t_1);    ubidots.add(PUBLISH_VARIABLE_LABEL_20, t_2);    ubidots.publish(PUBLISH_DEVICE_LABEL);
    timer = millis();
  }
  ubidots.loop();
}

```

Figura 54:. Programa en esp32 – método principal

3.10. Configuración en Plataforma Ubidots

Para la configuración en la plataforma Ubidots es necesario crear un usuario en la plataforma, permitiendo entrar a dispositivos y crear el nuevo equipo que estará alojado en la nube como se muestra en la figura 55.

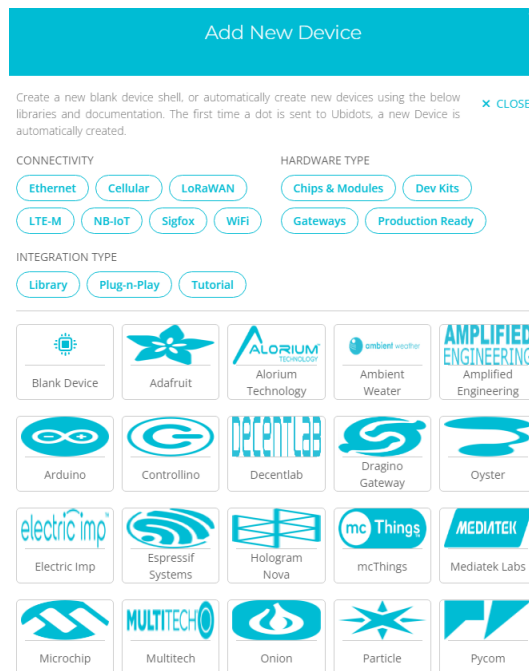


Figura 55: Creación de dispositivo en Ubidots

Posterior a la creación del dispositivo denominado maleta se mostrará lo q se observa en la Figura 56.

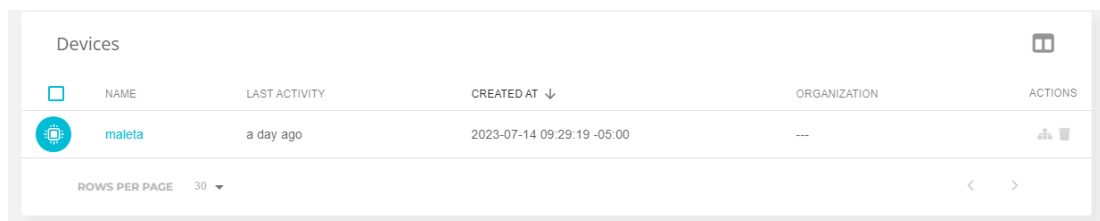


Figura 56: Dispositivo en Ubidots llamado maleta.

En el dispositivo maleta se proceden a crear las variables para los botones bt_1, bt_2, bt_3, bt_4, bt_5, para los valores de corriente ip_1, ip_2, ip_3, ip_4, ip_5, para los valores de voltaje vt_1, vt_2, vt_3, vt_4, vt_5, para los valores de las potencias p_1, p_2, p_3, p_4, p_5, para los valores de las sumas totales de corrientes y potencia llamado t_1, t_2 como se muestra en la Figura 57.

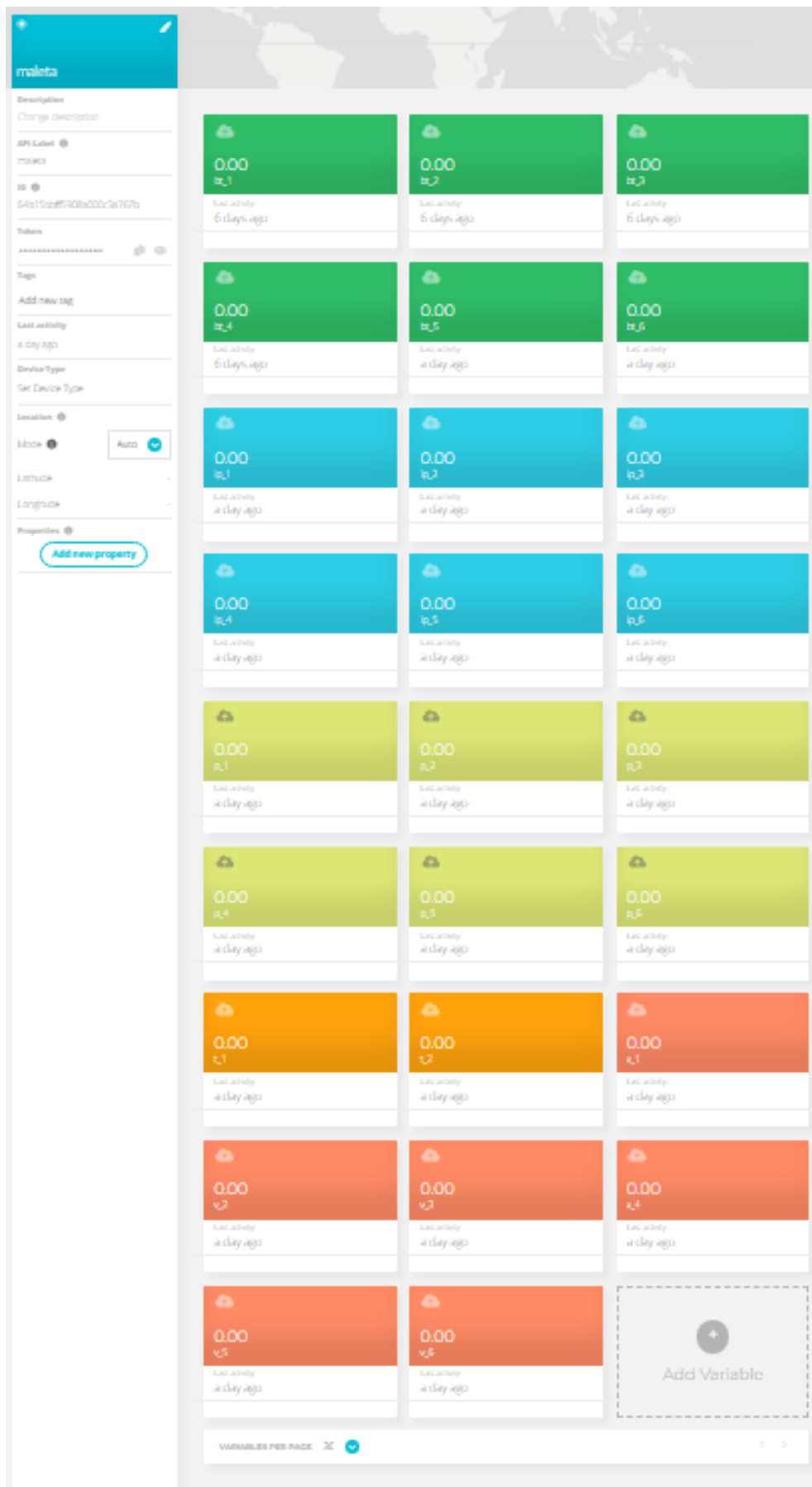
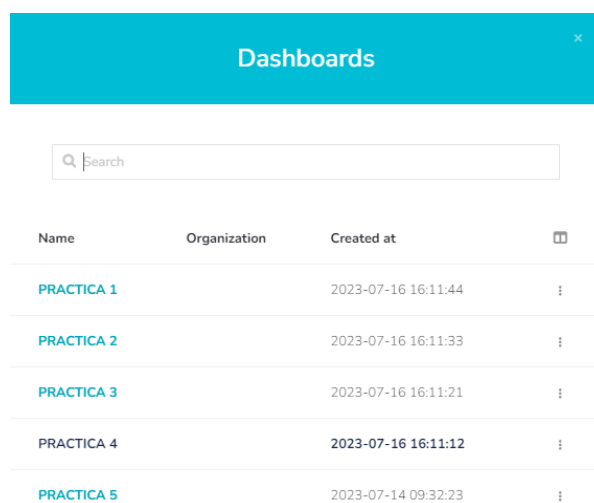


Figura 57: Variables en dispositivo maleta

El medio de visualización o espacio de trabajo en la plataforma se encuentra en la plantilla data se procede a crear 5 plantillas para cada práctica como se muestra en la Figura 58.



The screenshot shows a 'Dashboards' window with a search bar and a table of practice templates. The table has columns for Name, Organization, Created at, and an action menu icon.

Name	Organization	Created at	
PRACTICA 1		2023-07-16 16:11:44	⋮
PRACTICA 2		2023-07-16 16:11:33	⋮
PRACTICA 3		2023-07-16 16:11:21	⋮
PRACTICA 4		2023-07-16 16:11:12	⋮
PRACTICA 5		2023-07-14 09:32:23	⋮

Figura 58: Espacio de trabajo para prácticas

En el espacio de trabajo se procede a crear widget par el enlace con los botones que activaran los leds 1 y 2, en conjunto con los indicadores de corriente, voltaje y potencia, también la corriente y potencia total de los leds como se muestra en la Figura 59, también revisar el Anexo D para ver cómo crear widget y enlazar con variables.

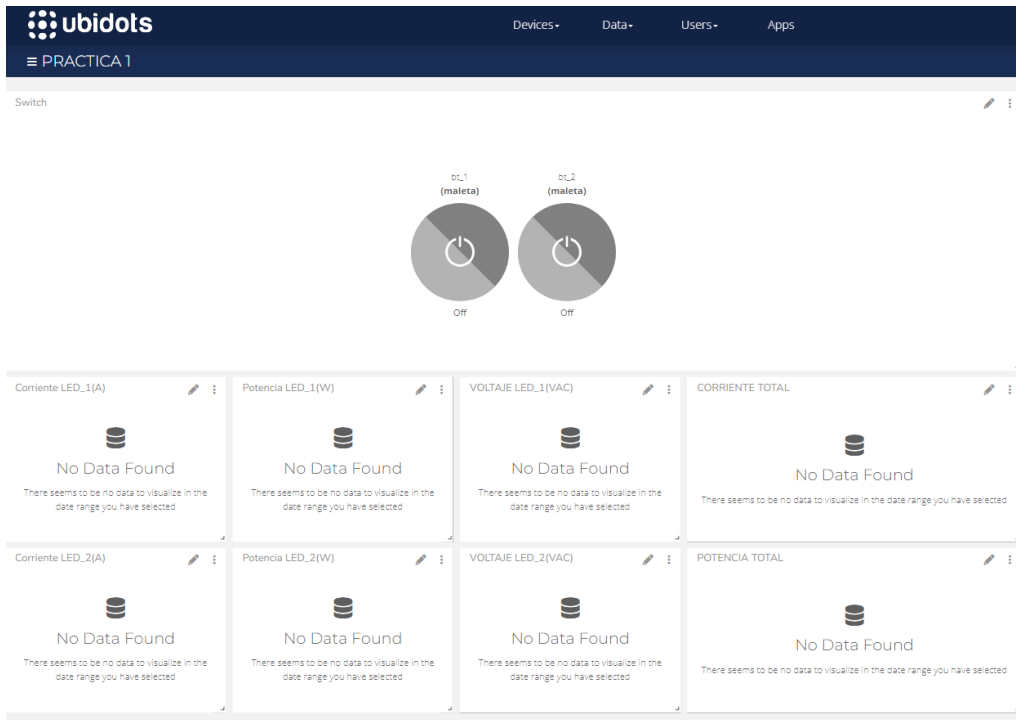


Figura 59:. Espacio de trabajo en práctica 1

En el espacio de trabajo se procede a crear widget par el enlace con los botones que activaran los puntos eléctricos 1 y 2, en conjunto con los indicadores de corriente, voltaje y potencia, también la corriente y potencia totales de los puntos eléctricos como se muestra en la Figura 60, también revisar el Anexo D para ver cómo crear widget y enlazar con variables.

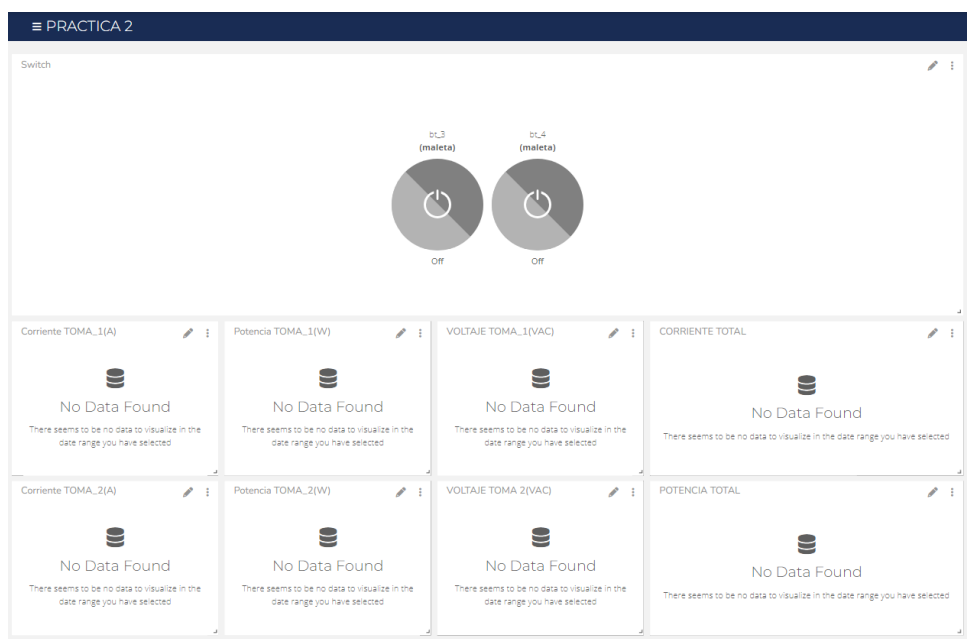


Figura 60:. Espacio de trabajo en práctica 2

En el espacio de trabajo se procede a crear widget par el enlace con los botones que activaran las cargas 1 y 2, en conjunto con los indicadores de corriente, voltaje y potencia, también la corriente y potencia totales de las cargas como se muestra en la Figura 61, también revisar el Anexo D para ver cómo crear widget y enlazar con variables.

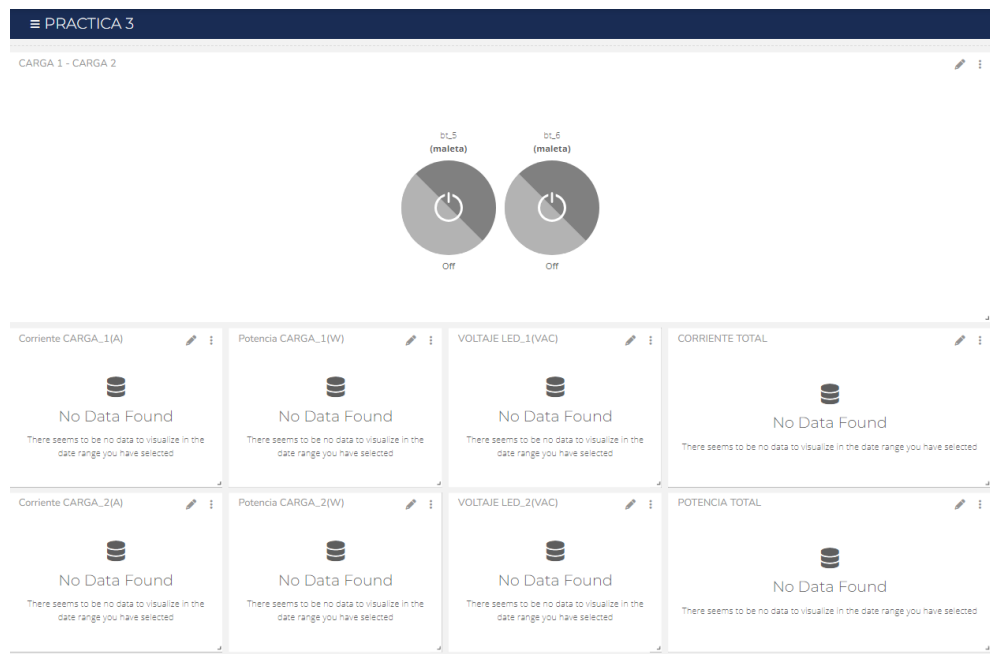


Figura 61:. Espacio de trabajo en práctica 3

En el espacio de trabajo se procede a crear widget par el enlace con los botones que activaran leds 1 y 2, puntos eléctricos 1 y 2, cargas 1y 2, en conjunto con los indicadores de corriente, voltaje y potencia, también la corriente y potencia totales como se muestra en la Figura 62, también revisar el Anexo D para ver cómo crear widget y enlazar con variables.

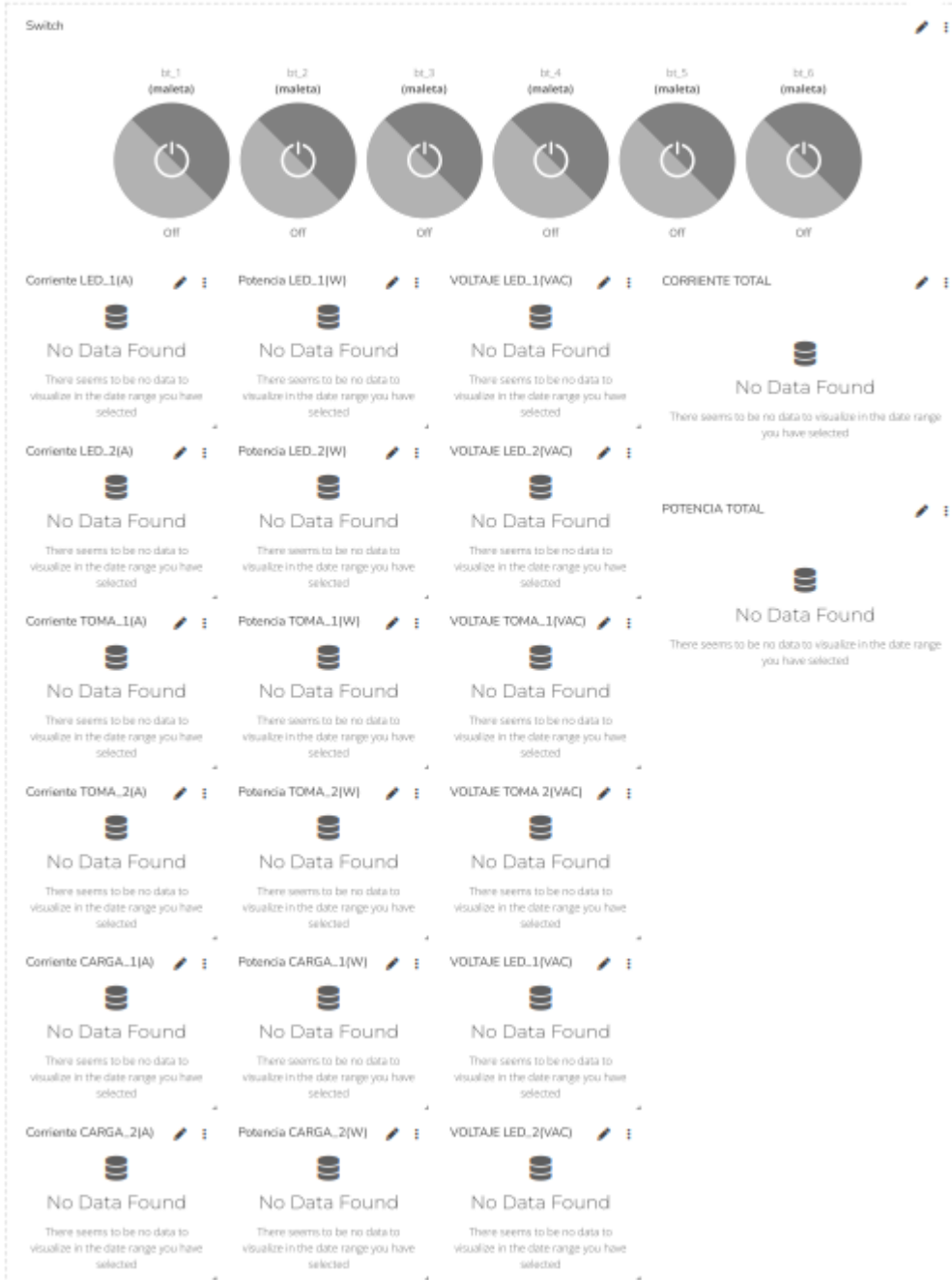


Figura 62:.. Espacio de trabajo en práctica 4

En el espacio de trabajo se procede a crear widget par el enlace con los botones que activaran leds 1 y 2, puntos eléctricos 1 y 2, cargas 1y 2, en conjunto con los indicadores de corriente, voltaje y potencia, también la corriente y potencia totales como se muestra en la Figura 63, también revisar el Anexo D para ver cómo crear widget y enlazar con variables.

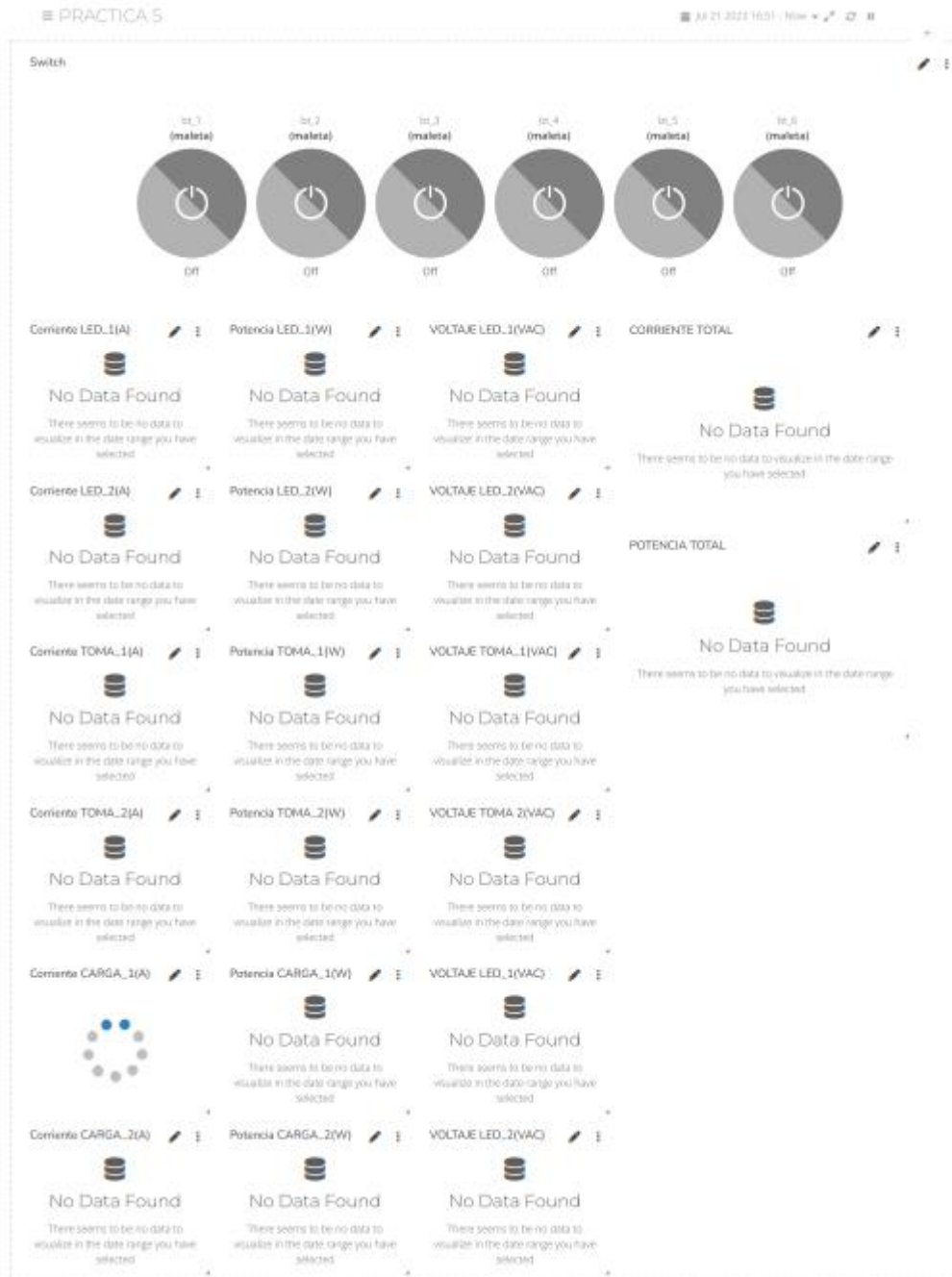



Figura 63: Espacio de trabajo en práctica 5

4. MANUAL DE PRÁCTICAS


Práctica#1: Control y lectura de señal para dicroico

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA:
No.:	1	TÍTULO: Control y lectura de señal para dicroico
OBJETIVOS: <ul style="list-style-type: none"> • Realizar las conexiones de la maleta en el tablero entre los bornes de alimentación y la sección de dicroico en I • Utilizar la plataforma Ubidots para visualizar el consumo de corriente en focos dicroicos. • Utilizar la plataforma Ubidots para visualizar la potencia en focos dicroicos. • Realizar la activación de los focos dicroicos desde la pantalla NEXTION 		
INSTRUCCIONES: <ol style="list-style-type: none"> 1. Analizar la solución propuesta de la práctica#1 ubicada en los anexos de la memoria técnica. 2. Conectar el módulo a la red eléctrica 110 VAC 3. Realizar las conexiones correctas del módulo de acuerdo con el esquemático planteado en la resolución del anexo de la PRÁCTICA 1. 		
ACTIVIDADES POR DESARROLLAR		
1. Realizar las conexiones pertinentes para la maleta		
2. Utilizar las herramientas de la nube para visualizar la corriente, voltaje y potencia en foco dicroico.		
3. Realizar la comparación entre una pinza amperimétrica y los valores dados por la maleta didáctica		
RESULTADOS: <ul style="list-style-type: none"> • Desarrollo de técnicas para aprendizaje en mediciones eléctricas. • Comprender el manejo de las herramientas de medición como multímetro. 		
CONCLUSIONES: <ul style="list-style-type: none"> • Al implementar la lectura de los valores de corriente y potencia se visualizan en una plataforma. • Para la visualización de los valores es necesario tener una conexión a la internet mediante red inalámbrica. 		
RECOMENDACIONES: <ul style="list-style-type: none"> • Revisar las conexiones de los bornes como indica el anexo de práctica 1. • Considerar la configuración de la práctica a realizarse en el módulo didáctico. • Revisar el marco metodológico y el anexo de programación para mayor aprendizaje del desarrollo y programación interna del modulo 		

Docente: ING. RAFAEL ENRIQUE PEREZ ORDÓÑEZ Msc.

Firma: _


Práctica#2: Control y lectura de señal para punto electrico

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA:
No.:	2	TÍTULO: Control y lectura de señal para punto eléctrico
OBJETIVOS: <ul style="list-style-type: none"> Realizar las conexiones de la maleta en el tablero entre los bornes de alimentación y la sección de los puntos eléctricos Utilizar la plataforma Ubidots para visualizar el consumo de corriente en los puntos eléctricos. Utilizar la plataforma Ubidots para visualizar la potencia en puntos eléctricos. Realizar la activación de los puntos eléctricos desde la pantalla NEXTION 		
INSTRUCCIONES: <ol style="list-style-type: none"> Analizar la solución propuesta de la práctica#2 ubicada en los anexos de la memoria técnica. Conectar el módulo a la red eléctrica 110 VAC Realizar las conexiones correctas del módulo de acuerdo con el esquemático planteado en la resolución del anexo de la PRÁCTICA 2 		
ACTIVIDADES POR DESARROLLAR		
1. Realizar las conexiones pertinentes para la maleta		
2. Utilizar las herramientas de la nube para visualizar la corriente, voltaje y potencia en puntos eléctricos.		
3. Realizar la comparación entre una pinza amperimétrica y los valores dados por la maleta didáctica		
RESULTADOS: <ul style="list-style-type: none"> Desarrollo de técnicas para aprendizaje en mediciones eléctricas. Comprender el manejo de las herramientas de medición. 		
CONCLUSIONES: <ul style="list-style-type: none"> Al implementar la lectura de los valores de corriente y potencia se visualizan en una plataforma. Para la visualización de los valores es necesario tener una conexión a la internet mediante red inalámbrica. 		
RECOMENDACIONES: <ul style="list-style-type: none"> Revisar las conexiones de los bornes como indica el anexo de práctica 2. Considerar la configuración de la práctica a realizarse en el módulo didáctico. Revisar el marco metodológico y el anexo de programación para mayor aprendizaje del desarrollo y programación interna del modulo 		

Docente: ING. RAFAEL ENRIQUE PEREZ ORDÓÑEZ Msc.

Firma: _


Práctica#3: Control y lectura de señal para carga

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA:
No.:	3	TÍTULO: <i>Control y lectura de señal para carga</i>
OBJETIVOS: <ul style="list-style-type: none"> • Realizar las conexiones de la maleta en el tablero entre los bornes de alimentación y la sección de las cargas • Utilizar la plataforma Ubidots para visualizar el consumo de corriente en la sección de las cargas • Utilizar la plataforma Ubidots para visualizar la potencia en la sección de las cargas • <i>Realizar la activación de las cargas desde la pantalla NEXTION</i> 		
INSTRUCCIONES:		
1. Analizar la solución propuesta de la práctica#3 ubicada en los anexos de la memoria técnica.		
2. Conectar el módulo a la red eléctrica 110 VAC		
3. Realizar las conexiones correctas del módulo de acuerdo con el esquemático planteado en la resolución del anexo de la PRÁCTICA 3.		
ACTIVIDADES POR DESARROLLAR		
1. Realizar las conexiones pertinentes para la maleta		
2. Utilizar las herramientas de la nube para visualizar la corriente, voltaje y potencia en las cargas led		
3. Realizar la comparación entre una pinza amperimétrica y los valores dados por la maleta didáctica		
RESULTADOS: <ul style="list-style-type: none"> • Desarrollo de técnicas para aprendizaje en mediciones eléctricas. • Comprender el manejo de las herramientas de medición como multímetro. 		
CONCLUSIONES: <ul style="list-style-type: none"> • Al implementar la lectura de los valores de corriente y potencia se visualizan en una plataforma. • Para la visualización de los valores es necesario tener una conexión a la internet mediante red inalámbrica. 		
RECOMENDACIONES: <ul style="list-style-type: none"> • Revisar las conexiones de los bornes como indica el anexo de práctica 3. • Considerar la configuración de la práctica a realizarse en el módulo didáctico. • Revisar el marco metodológico y el anexo de programación para mayor aprendizaje del desarrollo y programación interna del modulo 		

Docente: ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ Msc.

Firma: _


Práctica#4: Control IoT en maleta didáctica

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA:
No.:	4	TÍTULO: Control IoT en maleta didáctica
OBJETIVOS: <ul style="list-style-type: none"> • Realizar las conexiones de la maleta en el tablero entre los bornes de alimentación y las secciones de cargas, led y puntos eléctricos • Utilizar la plataforma Ubidots para visualizar el consumo de corriente de cada elemento en el modulo • Utilizar la plataforma Ubidots para visualizar la potencia total de cada elemento al momento de su activación. • Realizar la activación de los elementos o secciones desde la plataforma Ubidots 		
INSTRUCCIONES:		1. Analizar la solución propuesta de la práctica#4 ubicada en los anexos de la memoria técnica.
		2. Conectar el módulo a la red eléctrica 110 VAC
		3. Realizar las conexiones correctas del módulo de acuerdo con el esquemático planteado en la resolución del anexo de la PRÁCTICA 4.
ACTIVIDADES POR DESARROLLAR		
1. Realizar las conexiones pertinentes para la maleta		
2. Utilizar las herramientas de la nube para visualizar la corriente, voltaje y potencia en la plataforma Ubidots y controlar activación de las mismas.		
3. Realizar la comparación entre una pinza amperimétrica y los valores dados por la maleta didáctica		
RESULTADOS: <ul style="list-style-type: none"> • Desarrollo de técnicas para aprendizaje en mediciones eléctricas. • Comprender el manejo de las herramientas de medición como multímetro. 		
CONCLUSIONES: <ul style="list-style-type: none"> • Al implementar la lectura de los valores de corriente y potencia se visualizan en una plataforma. • Para la visualización de los valores es necesario tener una conexión a la internet mediante red inalámbrica. 		
RECOMENDACIONES: <ul style="list-style-type: none"> • Revisar las conexiones de los bornes como indica el anexo de práctica 5. • Considerar la configuración de la práctica a realizarse en el módulo didáctico. • Revisar el marco metodológico y el anexo de programación para mayor aprendizaje del desarrollo y programación interna del modulo 		

Docente: ING. RAFAEL ENRIQUE PEREZ ORDÓÑEZ Msc.

Firma: _

Práctica#5: Control y lectura de maleta mediante IoT y nextion

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA:
No.:	5	TÍTULO: <i>Control y lectura de maleta mediante IoT y nextion</i>
OBJETIVOS: <ul style="list-style-type: none"> • Realizar las conexiones de la maleta en el tablero entre los bornes de alimentación y cada elemento de acuerdo con el manual de práctica 5 • Utilizar la plataforma Ubidots para visualizar el consumo de corriente en la maleta • Utilizar la plataforma Ubidots para visualizar la potencia. • Realizar la activación de las cargas desde la pantalla NEXTION y desde la plataforma Ubidots 		
INSTRUCCIONES:		1. Analizar la solución propuesta de la práctica#5 ubicada en los anexos de la memoria técnica.
		2. Conectar el módulo a la red eléctrica 110 VAC
		3. Realizar las conexiones correctas del módulo de acuerdo con el esquemático planteado en la resolución del anexo de la PRÁCTICA 5.
ACTIVIDADES POR DESARROLLAR		
1. Realizar las conexiones pertinentes para la maleta		
2. Utilizar las herramientas de la nube para visualizar la corriente, voltaje y potencia en la maleta y su activación desde la plataforma o la pantalla nextion.		
3. Realizar la comparación entre una pinza amperimétrica y los valores dados por la maleta didáctica		
RESULTADOS: <ul style="list-style-type: none"> • Desarrollo de técnicas para aprendizaje en mediciones eléctricas. • Comprender el manejo de las herramientas de medición como multímetro. 		
CONCLUSIONES: <ul style="list-style-type: none"> • Al implementar la lectura de los valores de corriente y potencia se visualizan en una plataforma. • Para la visualización de los valores es necesario tener una conexión a la internet mediante red inalámbrica. 		
RECOMENDACIONES: <ul style="list-style-type: none"> • Revisar las conexiones de los bornes como indica el anexo de práctica 5. • Considerar la configuración de la práctica a realizarse en el módulo didáctico. • Revisar el marco metodológico y el anexo de programación para mayor aprendizaje del desarrollo y programación interna del modulo 		

Docente: ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ Msc.

Firma: _

RESULTADOS

La implementación del presente proyecto tiene como meta final la creación de dos maletas didácticas para realizar pruebas enfocadas en la medición de equipos eléctricos enfocados a la domótica como se observa en la Figura 64 los dos módulos.



Figura 64:.. Módulos didácticos

En el apartado de anexos F hasta anexo J se detalla el manual para utilizar las maletas como conexiones, credenciales, direcciones web para la plataforma Ubidots. En el apartado de la página web o plataforma es necesario ingresar la credencial **moduloups** y la contraseña **electronicaups** como se observa en la Figura 65 para poder ingresar al espacio de trabajo y realizar la supervisión.

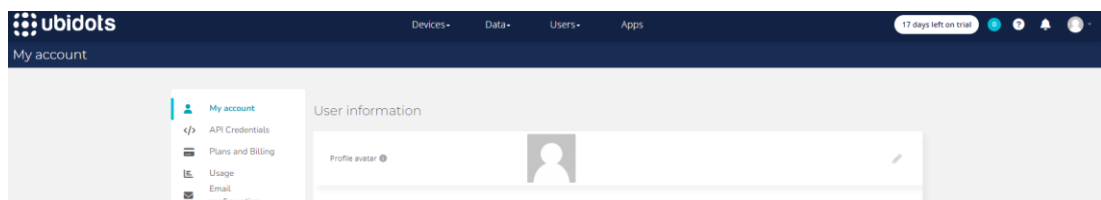


Figura 65:.. Plataforma Ubidots

Se registra cada una de las prácticas en la interface de ubidots como lo muestra la Figura 66.

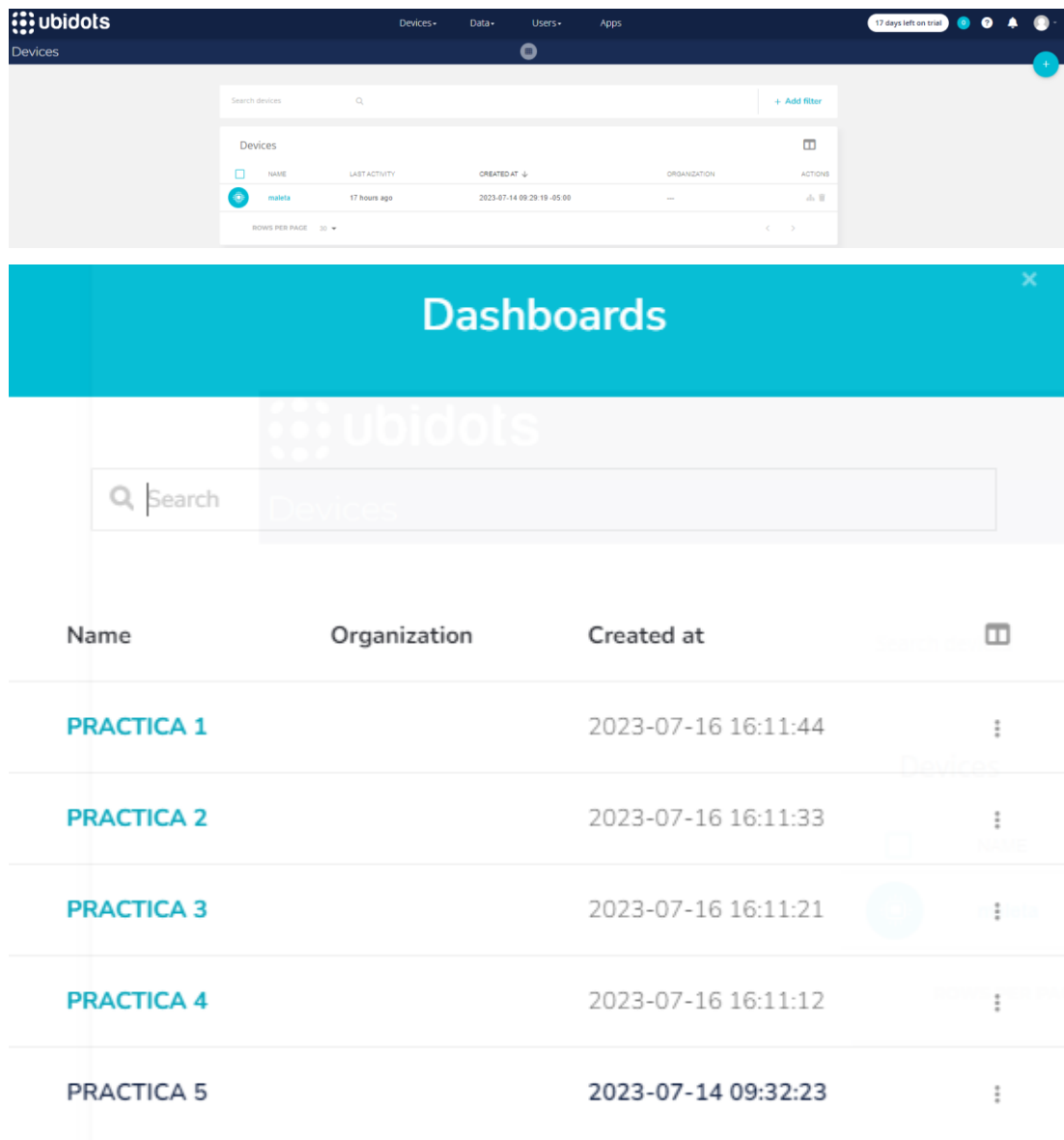


Figura 66: Espacios de trabajo en Ubidots para supervisión de prácticas

En la Figura 67 se detalla la implementación del sistema que para la lectura de corriente se utiliza los sensores ACS712 de voltaje y corriente conectados en serie a las cargas de manera independiente para su correcta lectura de esta manera obtener también mediante calculo por programa la potencia de cada equipo. También se detalla que el sistema para el realizar las cargas es un Atmel 32UA en conjunto con ESP32 ambos siendo controladores de 32 bits para esta implementación.

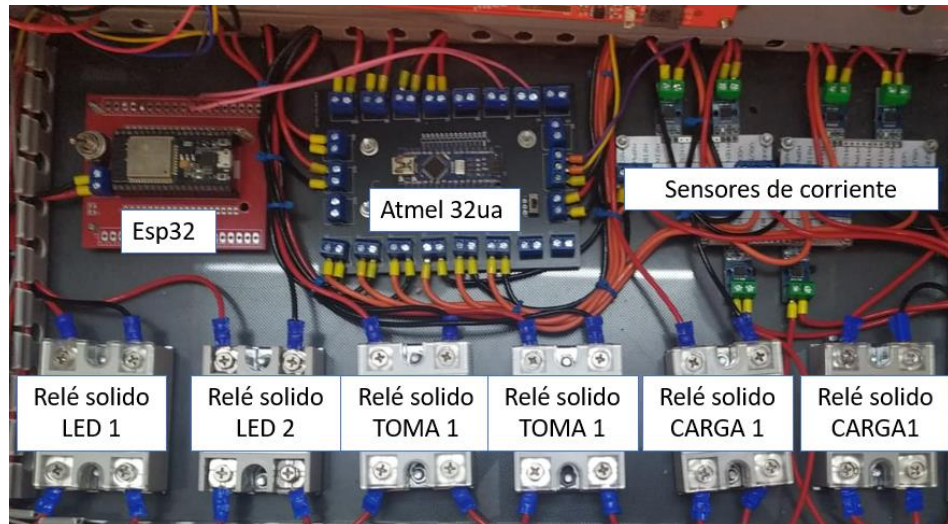


Figura 67:. Distribución de sensores y relés

En el anexo de resolución de práctica se detalla el uso de una pinza amperimétrica no invasiva (con certificación de la marca Fluke) para realizar una medición de corriente y comparar con los valores de lectura de los sensores ACS712. Se toma como ejemplo la implementación de la práctica 5 para los datos que se observan en la Tabla 1.

Elemento	Dato en maleta	Dato en pinza amperimétrica
Dicroico 1	0.31 Amperios	0.3 Amperios
Dicroico 2	0.30 Amperios	0.3 Amperios
Punto eléctrico 1	0.50 Amperios	0.5 Amperios
Punto eléctrico 2	0.71 amperios	0.7 amperios
Carga 1	0.92 Amperios	0.9 Amperios
Carga 2	0.52 Amperios	0.5 Amperios

Tabla 1 :Tabla de resultados

CRONOGRAMA

<i>DURACIÓN DEL PROYECTO</i>		
ITEM	ACTIVIDAD	MES
1	ADQUISICIÓN DE LOS COMPONENTES PARA EL MÓDULO DE ENTRENAMIENTO	1
2	DISEÑO DEL MÓDULO	1
3	DISEÑO DEL SISTEMA DE SUPERVISIÓN Y CONTROL MEDIANTE IOT	2
4	PRUEBAS DE CONEXIÓN Y MEDICIÓN DE CARGAS	4
5	RECOPIACIÓN DE INFORMACIÓN PARA REALIZACIÓN DE MONOGRAFÍA	4
6	PRUEBAS Y REVISIÓN DEL PROYECTO	5
7	CORRECCIÓN DE DOCUMENTO CURRICULAR Y TECNICO	6

PRESUPUESTO

Descripción	Cantidad	Valor	Total
Lámina de Aluminio 50cmx44cmx3mm	2	40	80
Maquinado de lamina	2	20	40
Rollo de cable rojo 18AWG 100 metros	1	40	40
Rollo de cable Negro 18AWG 100 metros	1	40	40
Serigrafía de lámina	2	10	20
Conectores tipo Banana 4mm Macho	40	0,85	34
Conectores tipo Banana 4mm Hembra	40	1,5	60
PCB 1 alimentación +5VDC	4	2,5	10
PCB 2 alimentación GND	4	2,5	10
PCB 3 Esp32	2	10	20
PCB 4 Controlador Nextion	2	10	20
Pantalla Nextion	2	120	240
Maleta de aluminio	2	100	200
Relés solidos	12	10	120
Sensores de corriente	12	12	144
Sensores de voltaje	12	12	144
Fuente conmutada	2	25	50
Soportes Foco dicroico	4	1	4
Soportes Foco	4	1	4
Tomacorriente	4	2,5	10
TOTAL			1290,00

CONCLUSIONES

- En la implementación de los bancos destinados a la medición eléctrica es necesario el uso de periféricos los cuales sean capaces de realizar lecturas de corriente y voltaje en corriente alterna.
- Mediante la implementación de la automatización sectorizada evitando sobre cargar el nivel de cómputo de un microcontrolador, se logró mediante el uso de un micro para procesamiento y otro para enlace con la plataforma IoT.
- Al realizar una guía para los estudiantes se obtiene un dispositivo capaz de ayudar a entender cómo funcionan los equipos de control domótico y de mediciones eléctricas.
- En la interacción de la plataforma y la maleta didáctica se desarrollo mediante microcontroladores de 32 bits ayudando a que el sistema sea rápido y eficaz.

RECOMENDACIONES

- Implementar un sistema de evaluación y retroalimentación constante por parte de los estudiantes que utilicen los bancos de pruebas.
- Buscar expandir la gama de prácticas que se pueden realizar con los bancos de pruebas móviles. Esto podría incluir aplicaciones más avanzadas de control domótico, simulaciones de procesos industriales complejos y situaciones reales que los estudiantes puedan enfrentar en el campo laboral.
- Mantenerse al día con los avances tecnológicos en IoT, control domótico y electrónica.
- Proporcionar a los estudiantes acceso a recursos de apoyo, como tutoriales, guías de usuario y material adicional, para que puedan aprovechar al máximo los bancos de pruebas y realizar sus proyectos de manera efectiva.

REFERENCIAS BIBLIOGRAFICAS

- Agüero, M. B. (2017). Obtenido de Sistema de adquisición de datos: <https://materias.df.uba.ar/mta2019c1/files/2014/08/Adquisicion-de-datos-Aguero.pdf>
- Álvarez-Moreira, P. J.-A. (2022). . *ACTIVO CORRIENTE EN LA RENTABILIDAD DE LA EMPRESA PROIMEC S.A. DEL CANTÓN MANTA, 2019-2020*. Revista Científica Multidisciplinaria Arbitrada YACHASUN, 6(11), 2-13. <https://doi.org/10.46296/yc.v6i11.0190>.
- Angulo Calle, J. P. (2009). *aptación de información de señales analógicas generada en investigación experimental*. Doctoral dissertation.
- ArduSpain. (2022). *Cómo conectar relé de estado solido Arduino*. Obtenido de Cómo conectar relé de estado solido Arduino: <https://arduino-spain.site/rele-estado-solido/>
- Arrow . (2015). Obtenido de DAS o DAQ: <https://www.arrow.com/es-mx/categories/data-acquisition/data-converters/data-acquisition-systems>
- Baño Baño, H. F. (2019). *istema empotrado de bajo costo para adquisición de datos basado en el estándar de comunicación industrial OPC-UA*. Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera Ingeniería Industrial en Procesos de Automatización.
- Bezunartea Oroz, X. (2021). *Dispositivo de medición y monitorización remota para centrales solares fotovoltaicas de autoconsumo con baterías de Arduino*.
- Cano Reyes, L. S. (2023). *Aplicación de un PLC al control de una banda transportadora*.
- Cedeño-Rodríguez, G. A.-G. (2020). *Informática libre en la Metodología de Aprendizaje basado en proyectos*. Polo del Conocimiento, 5(8), 1357-1367.
- Chan, F. N. (2011). *Análisis de sensibilidad al ruido en un sistema de comunicación sobre línea eléctrica para aplicaciones en Domótica empleando el protocolo X-10*. Ingeniería, 15(3), 147-156.
- Cifuentes Rubio, V. M. (2020). *Diseño e implementación para un servidor IOT de bajo costo utilizando STM 32 para el monitoreo de consumo eléctrico residencial*.
- CORTÉS OSORIO, J. A. (2008). *FUNDAMENTOS Y APLICACIÓN DEL MUESTREO EN SEÑALES UBICADAS EN LAS BANDAS ALTAS DEL ESPECTRO*. Scientia Et Technica, XIV(39), 37-42.

- CORTÉS OSORIO, J. A. (2009). *CRITERIOS DE SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO PARA MICROCONTROLADORES PIC*. *Scientia Et Technica*, XV(43), 187-192.
- descubrearduino. (2022). *Arduino Nano Pinout Funciones, Características y Donde Comprar*. Obtenido de Arduino Nano Pinout Funciones, Características y Donde Comprar: <https://descubrearduino.com/arduino-nano-pinout/>
- Duarte Acevedo, N. B. (2021). *esarrollo de sistema embebido para aplicar control adaptativo basado en redes neuronales a diferentes variables físicas*.
- Fernández Angulo, L. (2022). *ontrol de un relé de estado sólido mediante una aplicación Qt empleando Ethernet y Bus CAN*.
- Fierro Lora, J. E. (2022). *DISEÑO DE UN PLC BASADO EN UN MICROCONTROLADOR ARM CORTEX-M4 PARA SU APLICACIÓN ES UNA MÁQUINA DESTERRONADORA-CERNIDORA*.
- Flores Jacome, E. V. (2020). *mplementación de un módulo didáctico multifuncional para la asignatura de electrónica de potencia*. Bachelor's thesis, Ecuador: La Mana: Universidad Técnica de Cotopaxi UTC.
- FORERO, H. A. (2011). *IMPACTO DE LA INMERSIÓN DE TURBINAS EÓLICAS EN LA ESTABILIDAD DE VOLTAJE*. *Scientia Et Technica*, XVII(47), 29-33.
- Garrell, A. &. (2019). *La industria 4.0 en la sociedad digital*. Marge books.
- González Suarez, C. A. (2023). *Análisis comparativo de las plataformas web Ibm Cloud Y Alibaba Cloud*. Bachelor's thesis, Babahoyo: UTB-FAFI. 2023.
- Guarella, J. E. (2011). *Sensores y actuadores en motores*. .
- Hernández Areu, O. R. (2017). *Medición de la corriente de inrush en transformadores de distribución*. *Ingeniería Energética*, XXXVIII(2), 132-142.
- Herrera Calero, W. G. (2017). *Diseño de una red de automatización y control centralizada utilizando el protocolo inalámbrico ZIGBEE para el hogar*. Bachelor's thesis, Quito: Universidad de las Américas.
- Islas, L. R. (2022). *Gemelo virtual de un Brazo robótico colaborativo de 6 grados de libertad bajo la técnica de Realidad aumentada*. Doctoral dissertation, Tesis de Maestría Interinstitucional en Ciencia y Tecnología. Centro de Investigaciones en Óptica, AC Aguascalientes, Ag.
- Kang. (2000). MOSFET devices with polysilicon on single-layer. (págs. 35-38). IEEE Technical Digest. IEDM.
- Lammini Rodríguez, S. (2023). *Desarrollo de una plataforma de monitoreo de corredores en tiempo real con relojes inteligentes y Azure*.

- Lizarzaburu Bazán, P. &. (2023). *Desarrollo de un filtro digital para señales fotopletiográficas obtenidas de una tarjeta de adquisición de datos en un entorno de laboratorio.*
- Macías Valdivieso, C. (2022). *Creación de una botnet para dispositivos móviles.*
- Mejía Vásquez, E. J. (2019). *Predicción del consumo de energía eléctrica residencial de la Región Cajamarca mediante modelos Holt -Winters.* Ingeniería Energética, XL(3), 181-191.
- Muñoz Rodríguez, M. (2023). *Plataforma IoT para la provisión de servicios en procesos industriales.*
- naylampmechatronics. (2022). *naylampmechatronics.* Obtenido de TUTORIAL SENSOR DE CORRIENTE ACS712: https://naylampmechatronics.com/blog/48_tutorial-sensor-de-corriente-ac712.html
- Ordóñez, J. F. (2016). *Detector de bordes de imágenes usando un microcontrolador ARM.* Ingenius. Revista de Ciencia y Tecnología, (16), 30-35. <https://doi.org/10.17163/ings.n16.2016.04>.
- Paredes-Martínez, E. N.-I. (2008). *Convertidor digital a analógico de un bit.* Científica, 12(1), 25-30.
- Peña-Lecona, F. G.-Q.-M.-R.-R.-Z.-R.-G. (2008). *elemetría de alto voltaje con alta sensibilidad por medio de un sensor de campo eléctrico basado en fibra óptica y un cristal fotorefractivo.* Científica, 12(2), 87-95.
- Pereira, R. d. (2022). *Plataforma de enseñanza a distancia de microcontroladores e internet de las cosas.* ngenius. Revista de Ciencia y Tecnología, (28), 53-62. <https://doi.org/10.17163/ings.n28.2022.05>.
- Pizarro Stuardo, P. F. (2020). *Diseño y construcción de sistema de monitoreo de deformación estructural.*
- Pulache Huertas, D. J. (2023). *Diseño y simulación de una solución IoT industrial integrando equipos Siemens, Modbuspal y plataforma Node Red.*
- Pulido, M. Á. (2000). *Convertidores de frecuencia, controladores de motores y SSR.* Marcombo.
- Ramírez, J. O. (2022). *Control pasivo tolerante a aallos de sensado en dispositivos de compensación dinámicos - SVC mediante una estrategia híbrida.* Ingenius. Revista de Ciencia y Tecnología, (28), 34-43. <https://doi.org/10.17163/ings.n28.2022.03>.
- Salazar, F. N. (2021). *Arquitecturas de sistemas de captación de energía.* Enfoque UTE, 12(4), 45-57. <https://doi.org/10.29019/enfoqueute.777>.

- Sánchez Rohr, M. J. (2022). *Implementación sobre Raspberry-pi de protocolos de enrutamiento para redes de sensores.*
- Serrano Guzmán, M. F. (2017). *Método de sumas ponderadas para selección de sistemas energéticos no convencionales.* PROSPECTIVA, 15(2), 7-12. <https://doi.org/10.15665/rp.v15i2.913>.
- Seuret Jiménez, D. &. (2014). *Diseño y construcción de bobinas de Rogowski modificadas de bajo costo y alto desempeño.* Nova Scientia, 6(12), 219-233.
- Supo Halanoca, J. F. (2022). *Diseño e implementación de un sistema de monitoreo para una instalación fotovoltaica conectada a la red de 3.7 kW con convertidores CC/CC utilizando herramientas IoT y computación en la nube.*
- Tejada Gutierrez, J. M. (2019). *Diseño de un sistema prototipo domótica mediante la lectura de sensores y emisión de órdenes al sistema de actuadores con el apoyo de las IOT.*
- upesy. (2022). *Use the GPIO pins of the ESP32.* Obtenido de Use the GPIO pins of the ESP32: <https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide>
- Valencia, M. V. (2022). *Introducción a la adquisición y acondicionamiento de señales.* Instituto Tecnológico Metropolitano–ITM.
- Vásquez Lema, C. J. (2022). *Automatización de redes utilizadas PARA IoE: Análisis conceptual de la automatización de centro de datos para IOE .* Bachelor's thesis, Quito: EPN, 2022.
- Vera Tasamá, A. (2017). *Sistema informático basado en hardware embebido para la depuración de funciones electrónicas vía web.*
- Villacrés Rodríguez, M. G. (2017). *Diseño e instalación de un sistema de control, supervisión y monitoreo para la compuerta de restitución tipo vagón de la Central Hidroeléctrica San Francisco mediante el uso de un PLC de una interfaz humano–máquina.* Bachelor's thesis, Escuela Superior Politécnica de Chimborazo.
- Viteri Castellano, B. F. (2022). *Modelo de referencia de IoT para el diseño de casas inteligentes.* Master's thesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi .

ANEXOS

A. – Planos de la maleta

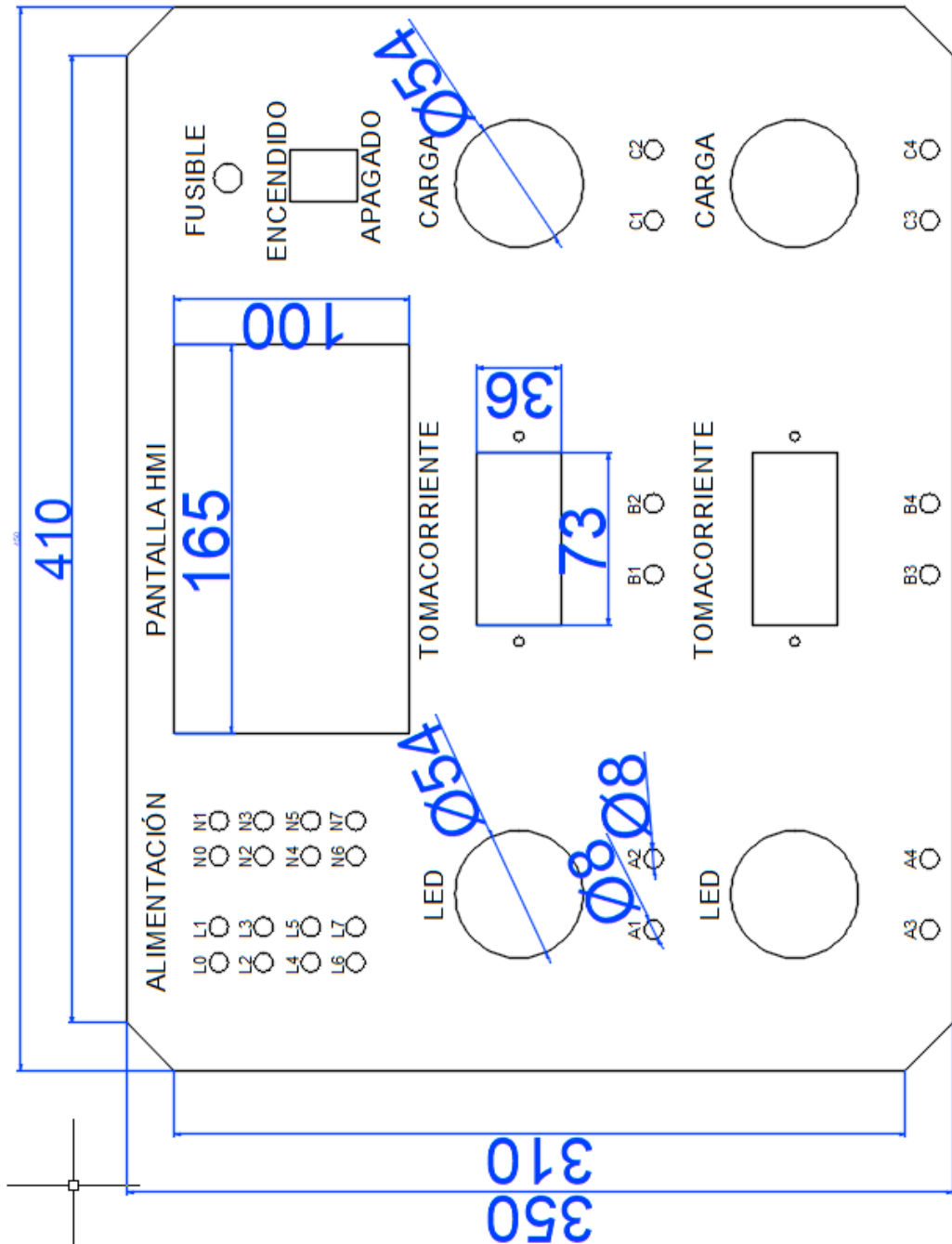


Figura 68: Dimensiones de placa de bornes

B. – Planos de las tarjetas de circuito impreso

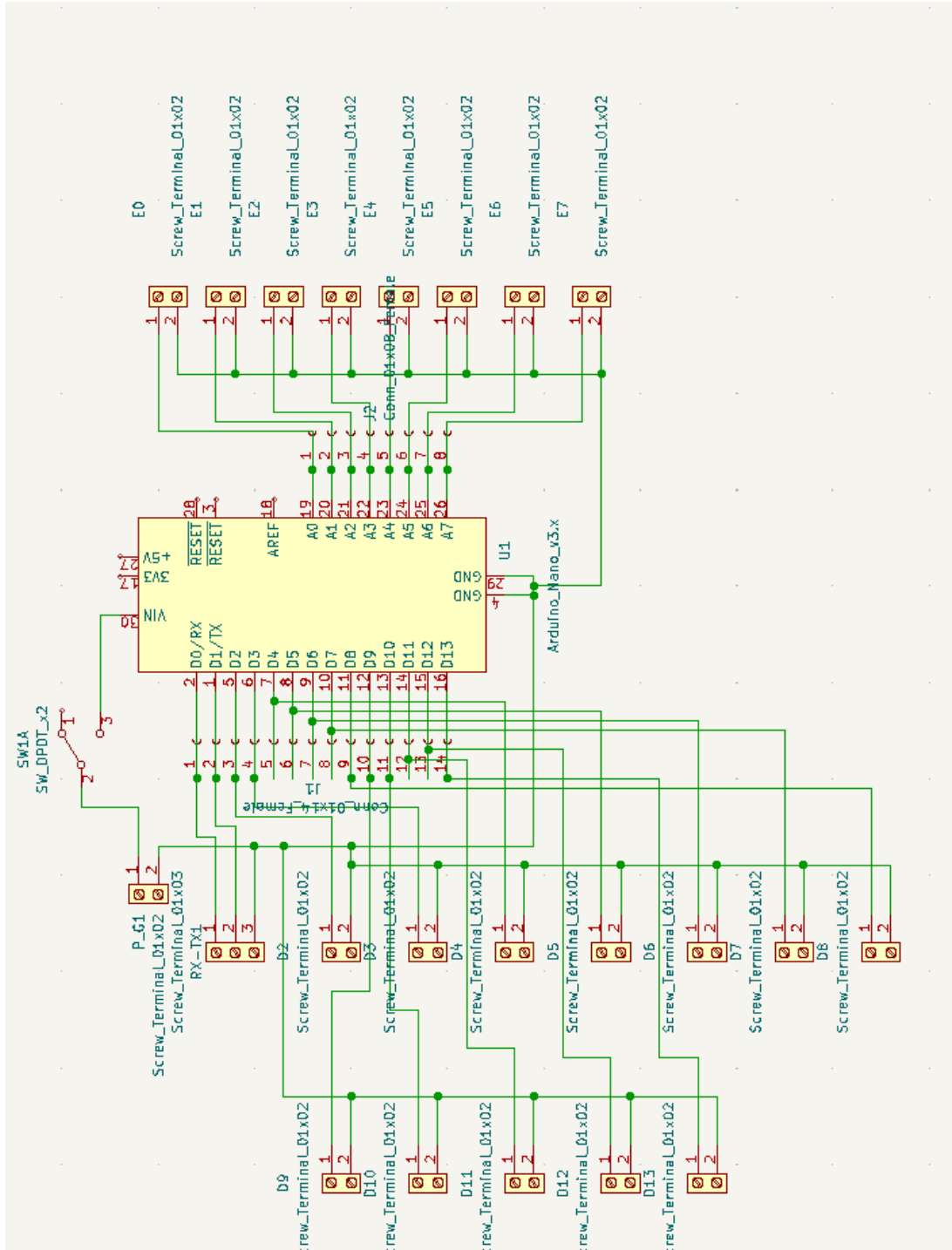


Figura 69.: Esquemático de Atmel y sensores

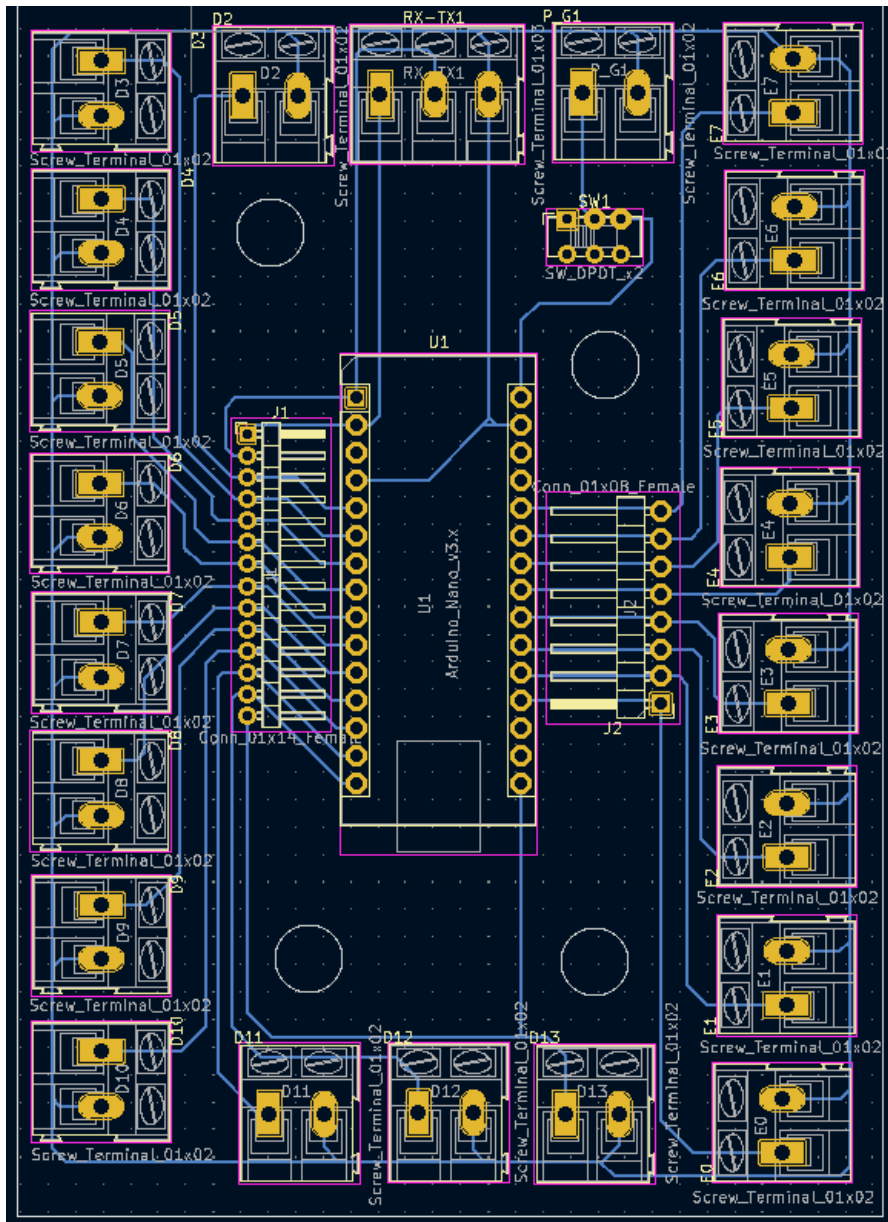


Figura 70.: Tarjeta de circuito impreso para atmel32

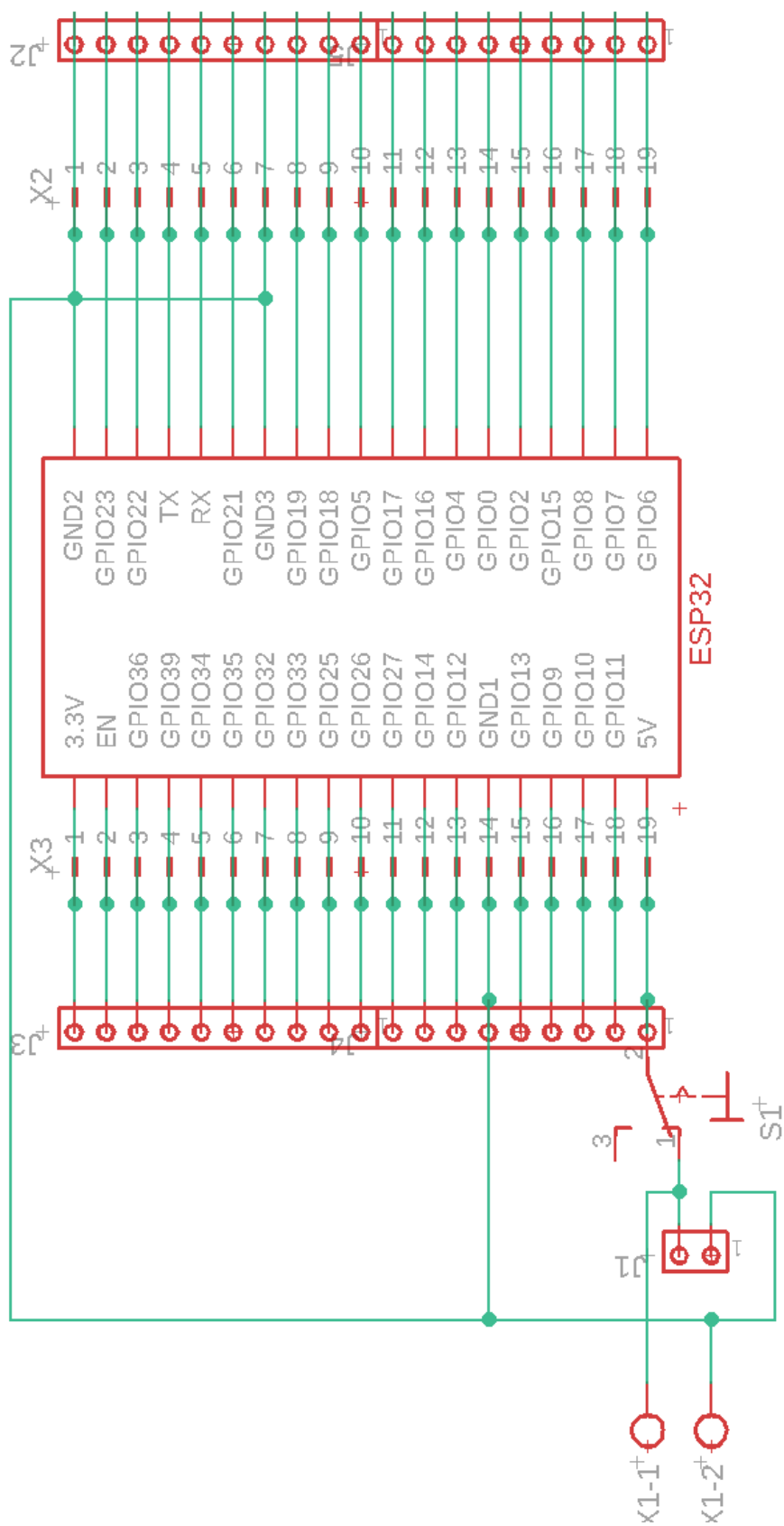


Figura 71.: Esquemático para tarjeta esp32

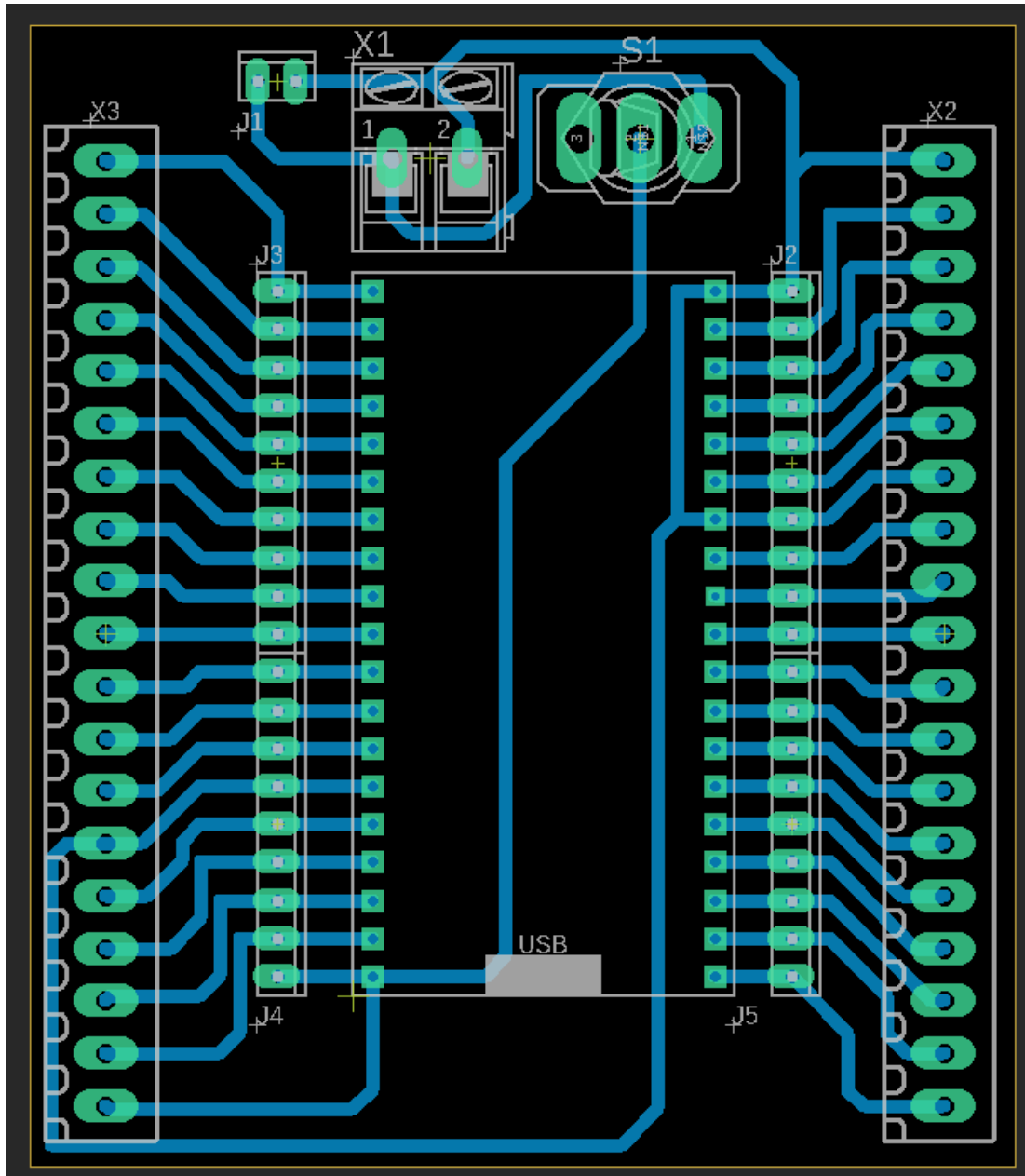


Figura 72: Tarjeta de circuito impreso para esp32

C. Programación de embebido de pantalla

```
#include "EasyNexionLibrary.h" // Include EasyNexionLibrary

EasyNex myNex(Serial); // Create an object of EasyNex class with the
name < myNex >
// Set as parameter the Hardware Serial you are going to use
#include <SoftwareSerial.h>
// software serial #1: RX = digital pin 5, TX = digital pin 4
SoftwareSerial portEnlace(5, 4);
float Sensibilidad = 0.139; //sensibilidad en V/A para nuestro sensor
float offset = 0.5; // Equivale a la amplitud del ruido

int salidaValue_1 = 12, salidaValue_2 = 11, salidaValue_3 = 10,
salidaValue_4 = 9, salidaValue_5 = 8, salidaValue_6 = 7;
int ledState_1 = LOW, ledState_2 = LOW, ledState_3 = LOW, ledState_4 =
LOW, ledState_5 = LOW, ledState_6 = LOW;
float voltajeSensor_1; //sensibilidad en V/A para nuestro sensor
float Ip_1 = 0, Ip_2 = 0, Ip_3 = 0, Ip_4 = 0, Ip_5 = 0, Ip_6 = 0; // se obtiene la
corriente pico
float Irms_1 = 0, Irms_2 = 0, Irms_3 = 0, Irms_4 = 0, Irms_5 = 0, Irms_6 = 0;
//Intensidad RMS = Ipico/(2^1/2)
float P_1 = 0, P_2 = 0, P_3 = 0, P_4 = 0, P_5 = 0, P_6 = 0;
float t_1 = 0, t_2 = 0;
void setup()
{
  // analogReference(INTERNAL);
  pinMode(salidaValue_1, OUTPUT);
  pinMode(salidaValue_2, OUTPUT);
  pinMode(salidaValue_3, OUTPUT);
  pinMode(salidaValue_4, OUTPUT);
  pinMode(salidaValue_5, OUTPUT);
```

```

pinMode(salidaValue_6, OUTPUT);
digitalWrite(salidaValue_1, ledState_1);
digitalWrite(salidaValue_2, ledState_2);
digitalWrite(salidaValue_3, ledState_3);
digitalWrite(salidaValue_4, ledState_4);
digitalWrite(salidaValue_5, ledState_5);
digitalWrite(salidaValue_6, ledState_6);
myNex.begin(9600);
portEnlace.begin(9600);
}

void loop() {
  get_corriente();
  delay(1);
  // Serial.print("Ip: "); Serial.print(lp_3);
  // Serial.print("A , Irms: "); Serial.print(Irms_3 ); Serial.print("A, Potencia: ");
  Serial.print(P_3); Serial.println("W");
  publicar_iot() ;
  delay(1);
  consultar_iot();
  delay(1);
  nextion_publicar();
  delay(1);
  nextion_consultar();
  delay(1);
}

void nextion_consultar()
{

  int x = myNex.readNumber("n12.val");
  switch (x)
  {

```

```

case 32:
    ledState_6 = !ledState_6;
    digitalWrite(salidaValue_6, ledState_6);
    break;
case 16:
    ledState_5 = !ledState_5;
    digitalWrite(salidaValue_5, ledState_5);
    break;
case 8:
    ledState_4 = !ledState_4;
    digitalWrite(salidaValue_4, ledState_4);
    break;
case 4:
    ledState_3 = !ledState_3;
    digitalWrite(salidaValue_3, ledState_3);
    break;
case 2:
    ledState_2 = !ledState_2;
    digitalWrite(salidaValue_2, ledState_2);
    break;
case 1:
    ledState_1 = !ledState_1;
    digitalWrite(salidaValue_1, ledState_1);
    break;
}

}

void nextion_publicar()
{

myNex.writeNum("x0.val", 110.00 * ledState_1 * 1000);
myNex.writeNum("x1.val", lp_1 * ledState_1 * 1000);

```

```

myNex.writeNum("x2.val", (float)P_1 * ledState_1 * 1000);
myNex.writeNum("x3.val", (float)110.00 * ledState_2 * 1000);
myNex.writeNum("x4.val", (float)Ip_2 * ledState_2 * 1000);
myNex.writeNum("x5.val", (float)P_2 * ledState_2 * 1000);
myNex.writeNum("x6.val", (float)110.00 * ledState_3 * 1000);
myNex.writeNum("x7.val", (float)Ip_3 * ledState_3 * 1000 );
myNex.writeNum("x8.val", (float)P_3 * ledState_3 * 1000);
myNex.writeNum("x9.val", (float)110.00 * ledState_4 * 1000);
myNex.writeNum("x10.val", (float)Ip_4 * ledState_4 * 1000);
myNex.writeNum("x11.val", (float)P_4 * ledState_4 * 1000);
myNex.writeNum("x12.val", (float)110.00 * ledState_5 * 1000);
myNex.writeNum("x13.val", (float)Ip_5 * ledState_5 * 1000);
myNex.writeNum("x14.val", (float)P_5 * ledState_5 * 1000);
myNex.writeNum("x15.val", (float)110.00 * ledState_6 * 1000);
myNex.writeNum("x16.val", (float)Ip_6 * ledState_6 * 1000);
myNex.writeNum("x17.val", (float)P_6 * ledState_6 * 1000);
t_1 = Ip_1 + Ip_2 + Ip_3 + Ip_4 + Ip_5 + Ip_6;
t_2 = P_1 + P_2 + P_3 + P_4 + P_5 + P_6;
myNex.writeNum("x22.val", (float)t_1 * 1000);
myNex.writeNum("x23.val", (float)t_2 * 1000);
}

```

```

void get_corriente()
{
    float voltajeSensor_1, voltajeSensor_2, voltajeSensor_3, voltajeSensor_4,
    voltajeSensor_5, voltajeSensor_6;
    float corriente_1 = 0, corriente_2 = 0, corriente_3 = 0, corriente_4 = 0,
    corriente_5 = 0, corriente_6 = 0;
    long tiempo = millis();
    float lmax_1 = 0, lmax_2 = 0, lmax_3 = 0, lmax_4 = 0, lmax_5 = 0, lmax_6
    = 0;
    float lmin_1 = 0, lmin_2 = 0, lmin_3 = 0, lmin_4 = 0, lmin_5 = 0, lmin_6 = 0;
    while (millis() - tiempo < 500) //Se realiza mediciones durante 0.5 segundos

```

```

{
  voltajeSensor_1 = analogRead(A0) * (5.0 / 1023.0); //lectura del sensor
  voltajeSensor_2 = analogRead(A1) * (5.0 / 1023.0); //lectura del sensor
  voltajeSensor_3 = analogRead(A2) * (5.0 / 1023.0); //lectura del sensor
  voltajeSensor_4 = analogRead(A3) * (5.0 / 1023.0); //lectura del sensor
  voltajeSensor_5 = analogRead(A4) * (5.0 / 1023.0); //lectura del sensor
  voltajeSensor_6 = analogRead(A5) * (5.0 / 1023.0); //lectura del sensor
  corriente_1 = 0.9 * corriente_1 + 0.1 * ((voltajeSensor_1 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  corriente_2 = 0.9 * corriente_2 + 0.1 * ((voltajeSensor_2 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  corriente_3 = 0.9 * corriente_3 + 0.1 * ((voltajeSensor_3 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  corriente_4 = 0.9 * corriente_4 + 0.1 * ((voltajeSensor_4 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  corriente_5 = 0.9 * corriente_5 + 0.1 * ((voltajeSensor_5 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  corriente_6 = 0.9 * corriente_6 + 0.1 * ((voltajeSensor_6 - 3.527) /
Sensibilidad); //Ecuación para obtener la corriente
  if (corriente_1 > Imax_1) Imax_1 = corriente_1;
  if (corriente_2 > Imax_2) Imax_2 = corriente_2;
  if (corriente_3 > Imax_3) Imax_3 = corriente_3;
  if (corriente_4 > Imax_4) Imax_4 = corriente_4;
  if (corriente_5 > Imax_5) Imax_5 = corriente_5;
  if (corriente_6 > Imax_6) Imax_6 = corriente_6;
  if (corriente_1 < Imin_1) Imin_1 = corriente_1;
  if (corriente_2 < Imin_2) Imin_2 = corriente_2;
  if (corriente_3 < Imin_3) Imin_3 = corriente_3;
  if (corriente_4 < Imin_4) Imin_4 = corriente_4;
  if (corriente_5 < Imin_5) Imin_5 = corriente_5;
  if (corriente_6 < Imin_6) Imin_6 = corriente_6;
}

```

```

Ip_1 = ((Imax_1 - Imin_1) / 2) - offset;
Ip_2 = ((Imax_2 - Imin_2) / 2) - offset;
Ip_3 = ((Imax_3 - Imin_3) / 2) - offset;
Ip_4 = ((Imax_4 - Imin_4) / 2) - offset;
Ip_5 = ((Imax_5 - Imin_5) / 2) - offset;
Ip_6 = ((Imax_6 - Imin_6) / 2) - offset;
Ip_1 = Ip_1 * ledState_1;
Ip_2 = Ip_2 * ledState_2;
Ip_3 = Ip_3 * ledState_3;
Ip_4 = Ip_4 * ledState_4;
Ip_5 = Ip_5 * ledState_5;
Ip_6 = Ip_6 * ledState_6;

Irms_1 = Ip_1 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_2 = Ip_2 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_3 = Ip_3 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_4 = Ip_4 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_5 = Ip_5 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_6 = Ip_6 * 0.707; //Intensidad RMS = Ipico/(2^1/2)
P_1 = Irms_1 * 110.0; // P=IV watts
P_2 = Irms_2 * 110.0; // P=IV watts
P_3 = Irms_3 * 110.0; // P=IV watts
P_4 = Irms_4 * 110.0; // P=IV watts
P_5 = Irms_5 * 110.0; // P=IV watts
P_6 = Irms_6 * 110.0; // P=IV watts
}

void publicar_iot() {
  portEnlace.print("A1:"); portEnlace.print(Ip_1); portEnlace.print(",");
  portEnlace.print("A2:"); portEnlace.print(Ip_2); portEnlace.print(",");
  portEnlace.print("A3:"); portEnlace.print(Ip_3); portEnlace.print(",");
  portEnlace.print("A4:"); portEnlace.print(Ip_4); portEnlace.print(",");
  portEnlace.print("A5:"); portEnlace.print(Ip_5); portEnlace.print(",");

```



```

    portEnlace.print("A6:"); portEnlace.print(lp_6); portEnlace.print(",");
    portEnlace.print("A7:"); portEnlace.print(110 * ledState_1);
portEnlace.print(",");
    portEnlace.print("A8:"); portEnlace.print(110 * ledState_2);
portEnlace.print(",");
    portEnlace.print("A9:"); portEnlace.print(110.00 * ledState_3);
portEnlace.print(",");
    portEnlace.print("A10:"); portEnlace.print(110.00 * ledState_4);
portEnlace.print(",");
    portEnlace.print("A11:"); portEnlace.print(110.00 * ledState_5);
portEnlace.print(",");
    portEnlace.print("A12:"); portEnlace.println(110.00 * ledState_6);

}

```

```

void consultar_iot() {
    char comando;
    while (portEnlace.available() > 0) {
        comando = portEnlace.read();
        switch (comando) {
            case 'A':
                ledState_1 = LOW;
                digitalWrite(salidaValue_1, ledState_1);
                break;
            case 'a':
                ledState_1 = HIGH;
                digitalWrite(salidaValue_1, ledState_1);
                break;
            case 'B':
                ledState_2 = LOW;
                digitalWrite(salidaValue_2, ledState_2);
                break;
            case 'b':

```

```
    ledState_2 = HIGH;
    digitalWrite(salidaValue_2, ledState_2);
    break;
case 'C':
    ledState_3 = LOW;
    digitalWrite(salidaValue_3, ledState_3);
    break;
case 'c':
    ledState_3 = HIGH;
    digitalWrite(salidaValue_3, ledState_3);
    break;
case 'D':
    ledState_4 = LOW;
    digitalWrite(salidaValue_4, ledState_4);
    break;
case 'd':
    ledState_4 = HIGH;
    digitalWrite(salidaValue_4, ledState_4);
    break;
case 'E':
    ledState_5 = LOW;
    digitalWrite(salidaValue_5, ledState_5);
    break;
case 'e':
    ledState_5 = HIGH;
    digitalWrite(salidaValue_5, ledState_5);
    break;
case 'F':
    ledState_6 = LOW;
    digitalWrite(salidaValue_6, ledState_6);
    break;
case 'f':
    ledState_6 = HIGH;
```

```
digitalWrite(salidaValue_6, ledState_6);  
break;  
}  
}  
}
```

D. – Programación de embebido de enlace a plataforma Ubidots

```
/******  
    Include Libraries  
*****/  
  
#include "UbidotsEsp32Mqtt.h"  
#include <HardwareSerial.h>  
HardwareSerial mySerial(2);  
/******  
    Define Constants  
*****/  
  
const char *UBIDOTS_TOKEN = "BBFF-CV3fPVTzSXropGZATrk1SvcsxsXHLy";  
// Put here your Ubidots TOKEN  
  
const char *WIFI_SSID = "modulo_maleta";          // Put here your Wi-Fi SSID  
const char *WIFI_PASS = "ELECTRONICAUPS";        // Put here your Wi-Fi  
password  
  
const char *PUBLISH_DEVICE_LABEL = "maleta";     // Put here your Device label  
to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_1 = "ip_1";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_2 = "ip_2";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_3 = "ip_3";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_4 = "ip_4";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_5 = "ip_5";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_6 = "ip_6";  // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_7 = "v_1";   // Put here your Variable  
label to which data will be published  
  
const char *PUBLISH_VARIABLE_LABEL_8 = "v_2";   // Put here your Variable  
label to which data will be published
```

```
const char *PUBLISH_VARIABLE_LABEL_9 = "v_3"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_10 = "v_4"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_11 = "v_5"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_12 = "v_6"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_13 = "p_1"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_14 = "p_2"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_15 = "p_3"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_16 = "p_4"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_17 = "p_5"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_18 = "p_6"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_19 = "t_1"; // Put here your Variable
label to which data will be published
const char *PUBLISH_VARIABLE_LABEL_20 = "t_2"; // Put here your Variable
label to which data will be published
const char *SUBSCRIBE_DEVICE_LABEL = "maleta"; // Replace with the device
label to subscribe to
const char *SUBSCRIBE_VARIABLE_LABEL_1 = "bt_1"; // Replace with the
variable label to subscribe to
const char *SUBSCRIBE_VARIABLE_LABEL_2 = "bt_2"; // Replace with the
variable label to subscribe to
const char *SUBSCRIBE_VARIABLE_LABEL_3 = "bt_3"; // Replace with the
variable label to subscribe to
const char *SUBSCRIBE_VARIABLE_LABEL_4 = "bt_4"; // Replace with the
variable label to subscribe to
const char *SUBSCRIBE_VARIABLE_LABEL_5 = "bt_5"; // Replace with the
variable label to subscribe to
```

```
const char *SUBSCRIBE_VARIABLE_LABEL_6 = "bt_6"; // Replace with the
variable label to subscribe to
```

```
const int PUBLISH_FREQUENCY = 5000; // Update rate in millisecondsx
```

```
const int num_sensores = 12;
```

```
float sensores[12];
```

```
unsigned long timer;
```

```
uint8_t analogPin = 34; // Pin used to read data from GPIO34 ADC_CH6.
```

```
float Irms_1 = 0, Irms_2 = 0, Irms_3 = 0, Irms_4 = 0, Irms_5 = 0, Irms_6 = 0;
```

```
//Intensidad RMS = Ipico/(21/2)
```

```
float P_1 = 0, P_2 = 0, P_3 = 0, P_4 = 0, P_5 = 0, P_6 = 0;
```

```
float t_1 = 0, t_2= 0;
```

```
Ubidots ubidots(UBIDOTS_TOKEN);
```

```
/******
```

```
  Auxiliar Functions
```

```
*****/
```

```
void callback(char *topic, byte *payload, unsigned int length)
```

```
{
```

```
  String mensaje = "Message arrived [";
```

```
  mensaje += topic;
```

```
  mensaje += "] ";
```

```
  for (int i = 0; i < length; i++) {
```

```
    mensaje += (char)payload[i];
```

```
  }
```

```
  mensaje += "\n";
```

```
  int etiqueta = obtenerEtiqueta(mensaje).toInt();
```

```
  int valor = obtenerValor(mensaje).toInt();
```

```
  Serial.println("Etiqueta: "); Serial.println(etiqueta);
```

```
  Serial.println(valor);
```

```
  switch (etiqueta) {
```

```
    case 1:
```

```
      if (valor == 0.0) {
```

```
        mySerial.println("A");
```

```
    }  
    else {  
        mySerial.println("a");  
    }  
    break;  
case 2:  
    if (valor == 0.0) {  
        mySerial.println("B");  
    }  
    else {  
        mySerial.println("b");  
    }  
    break;  
case 3:  
    if (valor == 0.0) {  
        mySerial.println("C");  
    }  
    else {  
        mySerial.println("c");  
    }  
    break;  
case 4:  
    if (valor == 0.0) {  
        mySerial.println("D");  
    }  
    else {  
        mySerial.println("d");  
    }  
    break;  
case 5:  
    if (valor == 0.0) {  
        mySerial.println("E");  
    }  
    else {  
        mySerial.println("e");  
    }  
}
```

```

    break;
case 6:
    if (valor == 0.0) {
        mySerial.println("F");
    }
    else {
        mySerial.println("f");
    }
    break;
}
}

```

```

/*****

```

Main Functions

```

*****/

```

```

void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    mySerial.begin(9600);
    // ubidots.setDebug(true); // uncomment this to make debug messages available
    ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
    ubidots.setCallback(callback);
    ubidots.setup();
    ubidots.reconnect();
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_1); // Insert the device and variable's Labels,
respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_2); // Insert the device and variable's Labels,
respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_3); // Insert the device and variable's Labels,
respectively
}

```



```

    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_4); // Insert the device and variable's Labels,
respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_5); // Insert the device and variable's Labels,
respectively
    ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_6); // Insert the device and variable's Labels,
respectively

    timer = millis();
}

void loop()
{
    datos();

    principal();
}

void datos()
{
    if (mySerial.available() > 0) {
        // Lectura de los datos
        String datos = mySerial.readStringUntil('\n');

        // Extracción de los valores de los sensores
        int inicio = 0;
        for (int i = 0; i < num_sensores; i++) {
            String identificador = "A" + String(i + 1) + ":";
            int inicio_valor = datos.indexOf(identificador) + identificador.length();
            int fin_valor = datos.indexOf(',', inicio_valor);
            float valor = datos.substring(inicio_valor, fin_valor).toFloat();
            sensores[i] = valor;
        }
    }
}

```

```

    // Impresión de los valores de los sensores
    //   for (int i = 0; i < num_sensores; i++) {
    //       Serial.print("Sensor ");
    //       Serial.print(i + 1);
    //       Serial.print(": ");
    //       Serial.println(sensores[i], 4); // Imprimir con 4 decimales
    //   }
    }
}

void principal()
{

    // put your main code here, to run repeatedly:
    if (!ubidots.connected())
    {
        ubidots.reconnect();
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_1); // Insert the device and variable's Labels,
respectively
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_2); // Insert the device and variable's Labels,
respectively
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_3); // Insert the device and variable's Labels,
respectively
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_4); // Insert the device and variable's Labels,
respectively
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_5); // Insert the device and variable's Labels,
respectively
        ubidots.subscribeLastValue(SUBSCRIBE_DEVICE_LABEL,
SUBSCRIBE_VARIABLE_LABEL_6); // Insert the device and variable's Labels,
respectively
    }
}

```

```

if (abs(millis() - timer) > PUBLISH_FREQUENCY) // triggers the routine every 5
seconds
{
float value = analogRead(analogPin);
ubidots.add(PUBLISH_VARIABLE_LABEL_1, sensores[0]); // Variable Corriente
1
ubidots.add(PUBLISH_VARIABLE_LABEL_2, sensores[1]); // Variable Corriente
2
ubidots.add(PUBLISH_VARIABLE_LABEL_3, sensores[2]); // Variable Corriente
3
ubidots.add(PUBLISH_VARIABLE_LABEL_4, sensores[3]); // Variable Corriente
4
ubidots.add(PUBLISH_VARIABLE_LABEL_5, sensores[4]); // Variable Corriente
5
ubidots.add(PUBLISH_VARIABLE_LABEL_6, sensores[5]); // Variable Corriente
6
ubidots.publish(PUBLISH_DEVICE_LABEL);
ubidots.add(PUBLISH_VARIABLE_LABEL_7, sensores[6]); // Variable Voltaje 1
ubidots.add(PUBLISH_VARIABLE_LABEL_8, sensores[7]); // Variable Voltaje 2
ubidots.add(PUBLISH_VARIABLE_LABEL_9, sensores[8]); // Variable Voltaje 3
ubidots.add(PUBLISH_VARIABLE_LABEL_10, sensores[9]); // Variable Voltaje 4
ubidots.add(PUBLISH_VARIABLE_LABEL_11, sensores[10]); // Variable Voltaje 5
ubidots.add(PUBLISH_VARIABLE_LABEL_12, sensores[11]); // Variable Voltaje
6
ubidots.publish(PUBLISH_DEVICE_LABEL);
Irms_1 = sensores[0] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_2 = sensores[1] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_3 = sensores[2] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_4 = sensores[3] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_5 = sensores[4] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
Irms_6 = sensores[5] * 0.707; //Intensidad RMS = Ipico/(2^1/2)
P_1 = Irms_1 * sensores[6]; // P=IV watts
P_2 = Irms_2 * sensores[7]; // P=IV watts
P_3 = Irms_3 * sensores[8]; // P=IV watts
P_4 = Irms_4 * sensores[9]; // P=IV watts
P_5 = Irms_5 * sensores[10]; // P=IV watts

```

```

P_6 = Irms_6 * sensores[11]; // P=IV watts
ubidots.add(PUBLISH_VARIABLE_LABEL_13, P_1); // Variable Potencia 1
ubidots.add(PUBLISH_VARIABLE_LABEL_14, P_2); // Variable Potencia 2
ubidots.add(PUBLISH_VARIABLE_LABEL_15, P_3); // Variable Potencia 3
ubidots.add(PUBLISH_VARIABLE_LABEL_16, P_4); // Variable Potencia 4
ubidots.add(PUBLISH_VARIABLE_LABEL_17, P_5); // Variable Potencia 5
ubidots.add(PUBLISH_VARIABLE_LABEL_18, P_6); // Variable Potencia 6
ubidots.publish(PUBLISH_DEVICE_LABEL);
t_1= sensores[0]+ sensores[1]+ sensores[2]+ sensores[3]+ sensores[4]+
sensores[5];
t_2=P_1+P_2+P_3+P_4+P_5+P_6;
ubidots.add(PUBLISH_VARIABLE_LABEL_19, t_1); // Variable Potencia 5
ubidots.add(PUBLISH_VARIABLE_LABEL_20, t_2); // Variable Potencia 6
ubidots.publish(PUBLISH_DEVICE_LABEL);
timer = millis();
}

ubidots.loop();

}

```

```

String obtenerEtiqueta(String mensaje) {
    int inicioEtiqueta = mensaje.indexOf("/bt_") + 4;
    int finEtiqueta = mensaje.indexOf("/lv");

    if (inicioEtiqueta >= 0 && finEtiqueta >= 0 && finEtiqueta > inicioEtiqueta) {
        return mensaje.substring(inicioEtiqueta, finEtiqueta);
    } else {
        return "";
    }
}

```

```

String obtenerValor(String mensaje) {
    int inicioValor = mensaje.lastIndexOf("]") + 2;
    if (inicioValor >= 0 && inicioValor < mensaje.length()) {
        return mensaje.substring(inicioValor);
    }
}

```

```
} else {  
  return ""; } }
```

E. – Programación en Ubidots

Creación de variables en dispositivo

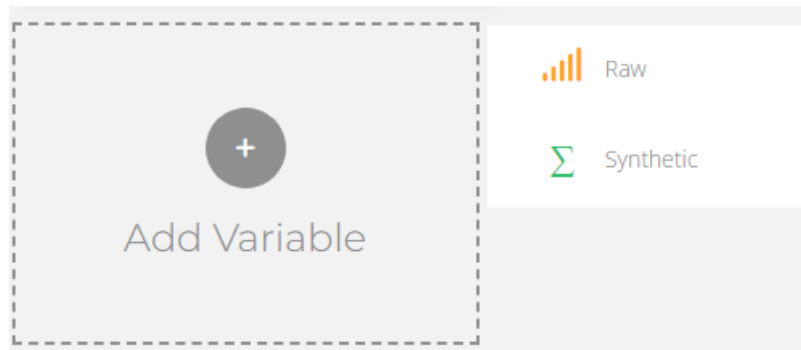


Figura 73: Creación de dispositivo en Ubidots

Asignar el nombre de la variable creada

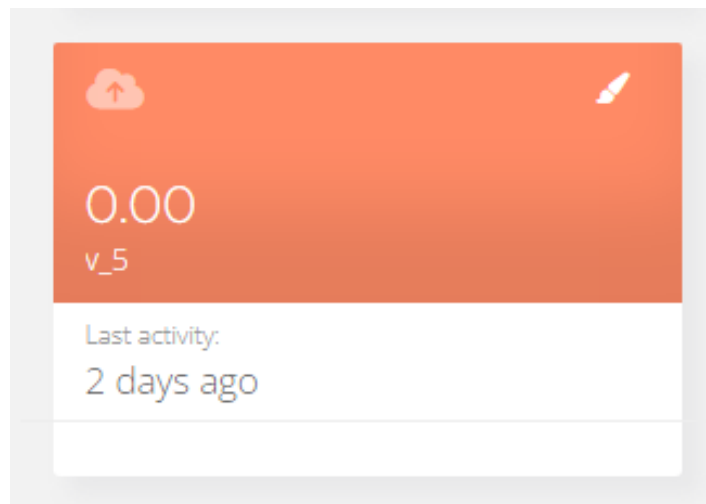


Figura 74: Creación de dispositivo en Ubidots

Enlace de variable desde espacio de trabajo a dispositivo.

Seleccionar el tipo de widget a utilizar como ejemplo se usará métrico para la visualización de datos numéricos.

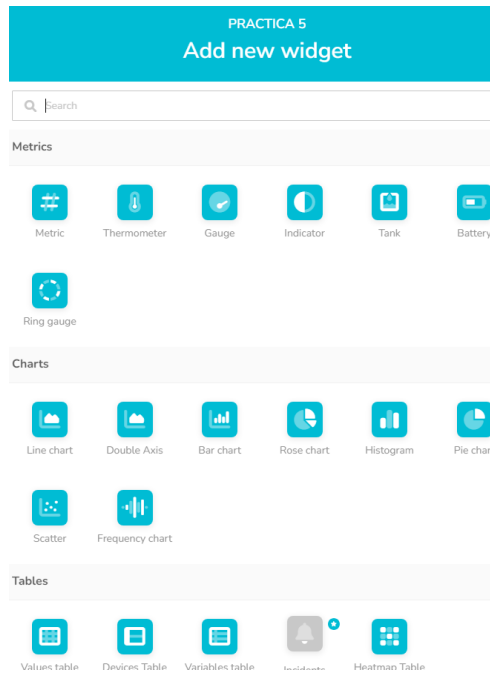


Figura 75:. Creación de dispositivo en Ubidots

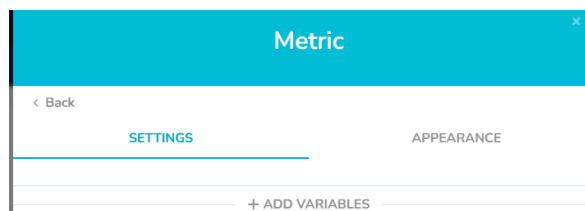


Figura 76:. Creación de dispositivo en Ubidots

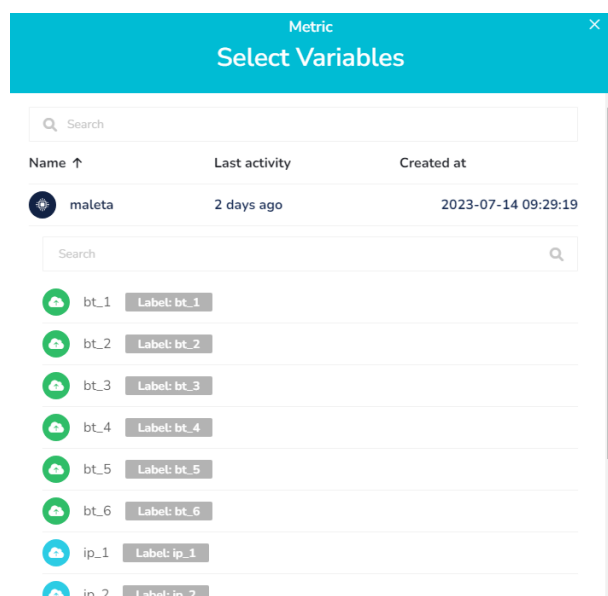


Figura 77:. Creación de dispositivo en Ubidots

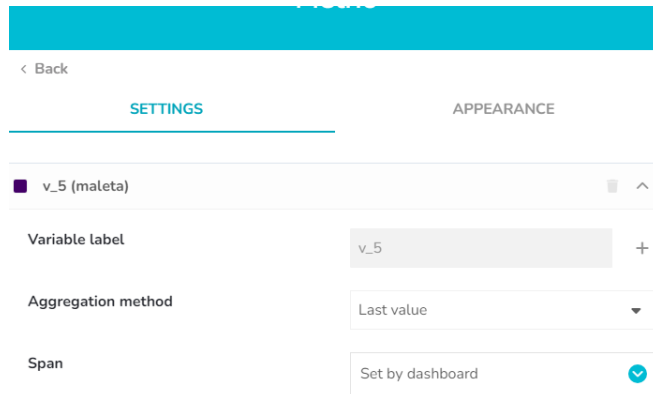


Figura 78: Creación de dispositivo en Ubidots

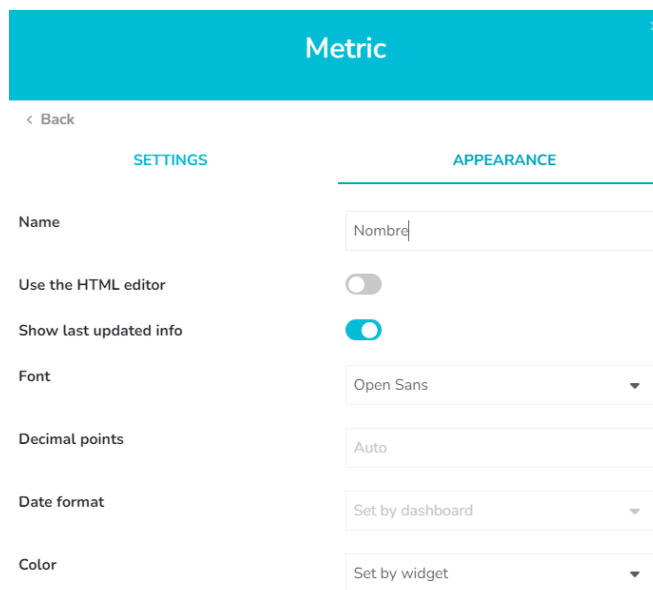


Figura 79: Creación de dispositivo en Ubidots

F. RESOLUCIÓN DE PRÁCTICA 1

PRÁCTICA 1

NÚMERO DE ESTUDIANTES: 2

DOCENTE

ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

TIEMPO ESTIMADO: 1 HORAS

**TEMA: “CONTROL Y LECTURA DE SEÑAL PARA
DICROICO.”**

A. Objetivo General

- Supervisar el consumo de diodo

B. Objetivos Específicos

- Realizar la lectura de la corriente de un foco diodo
- Controlar la activación del foco diodo desde la pantalla NEXTION

C. Marco Procedimental

1. Realizar la conexión de acuerdo con la Figura en la maleta didáctica.

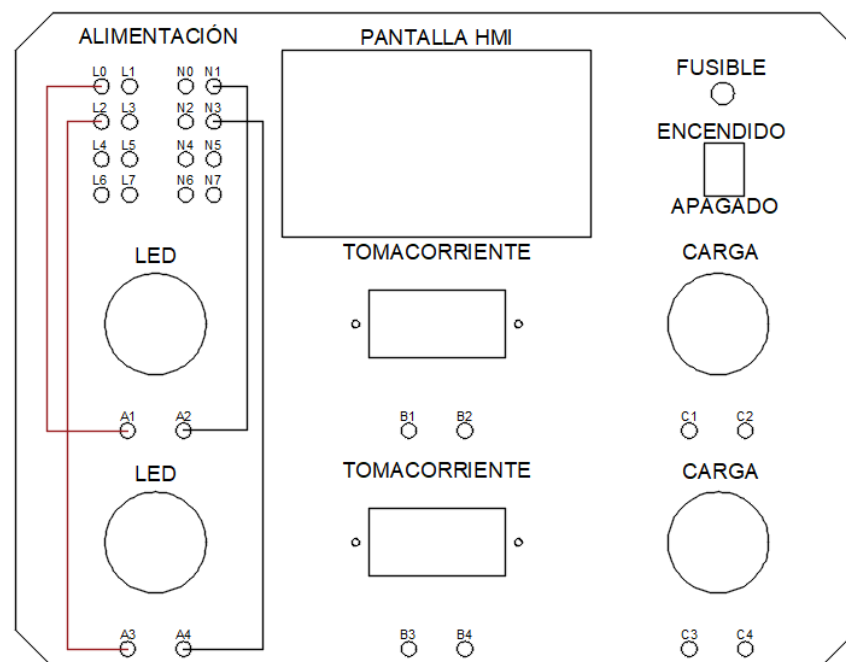


Figura 80.: Diagrama de conexiones físicas – Práctica 1

2. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido.

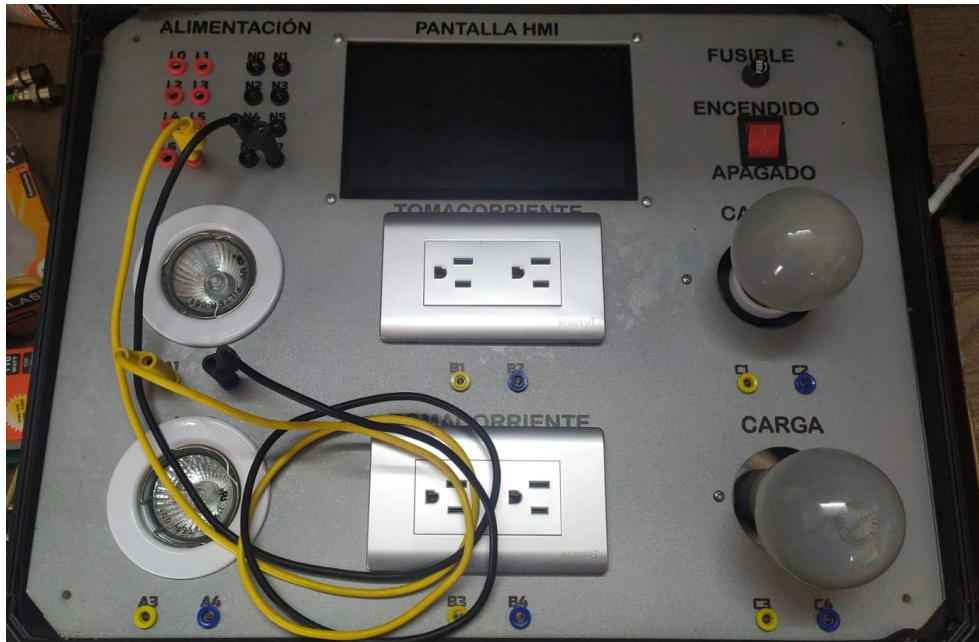


Figura 81:. Conexiones físicas – Práctica 1

- 3.** Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido. Proceder a seleccionar en el panel de inicio de la pantalla la práctica 1

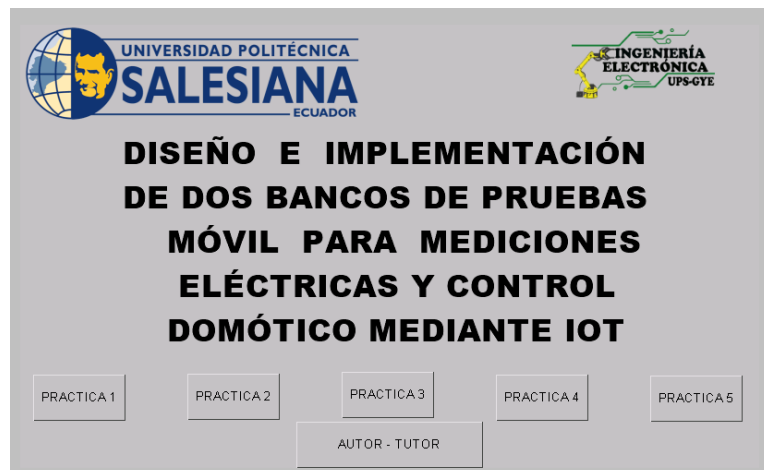


Figura 82:. Pantalla de inicio en Nextion – Práctica 1

- 4.** En el panel seleccionar la activación de cada led y visualizar el consumo de cada led en la pantalla.

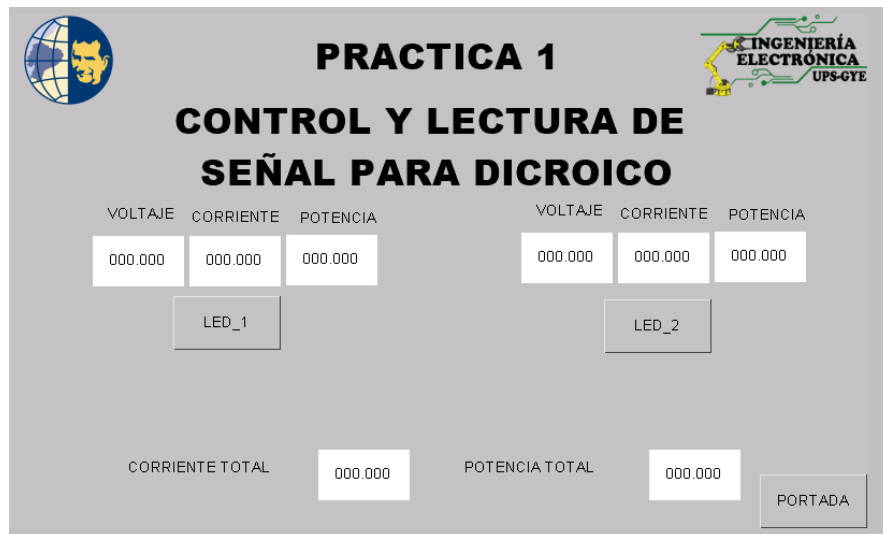


Figura 83:. Panel en Nextion – Práctica 1



Figura 84:. Vista de panel en Nextion – Práctica 1

5. Activar la carga 1 mediante los botones táctiles en la pantalla y visualizar la activación y consumo del led en los indicadores numéricos de la pantalla.

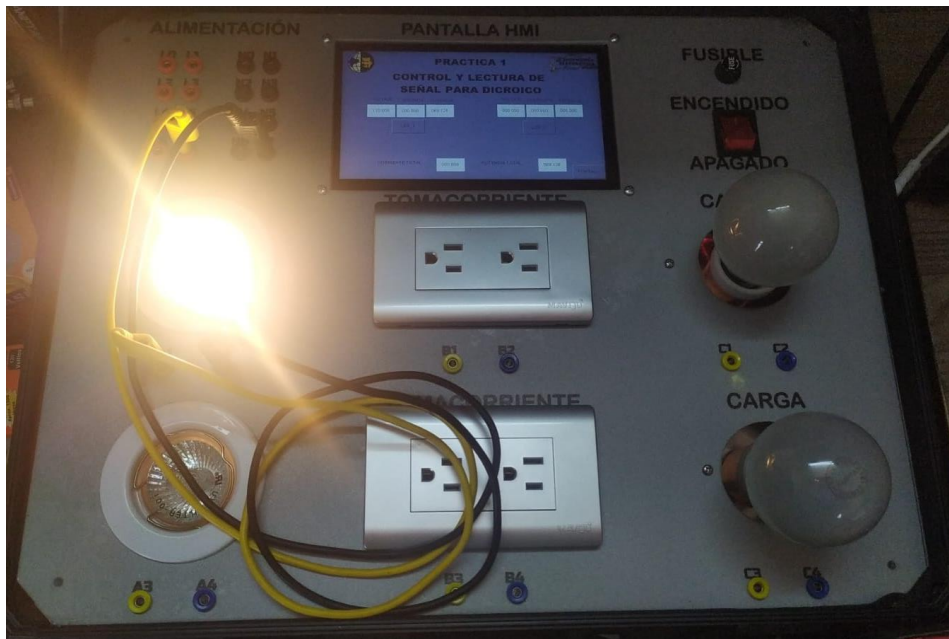


Figura 85: Activación de foco dicroico 1 – Práctica

- 6.** Activar la carga 2 mediante los botones táctiles en la pantalla y visualizar la activación y consumo del led en los indicadores numéricos de la pantalla.

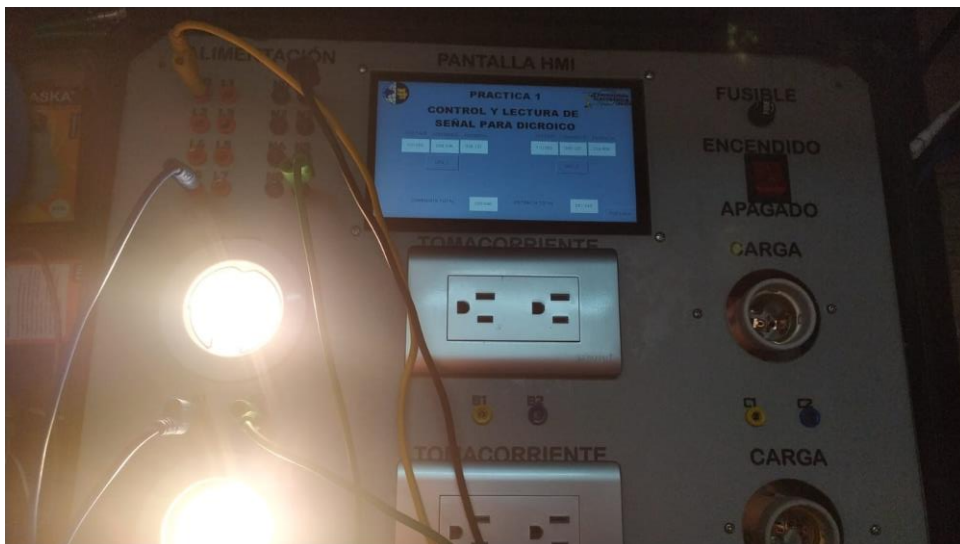


Figura 86: Activación de foco dicroico 2 – Práctica

- 7.** Visualizar consumo de cada foco dicroico en pantalla nextion.

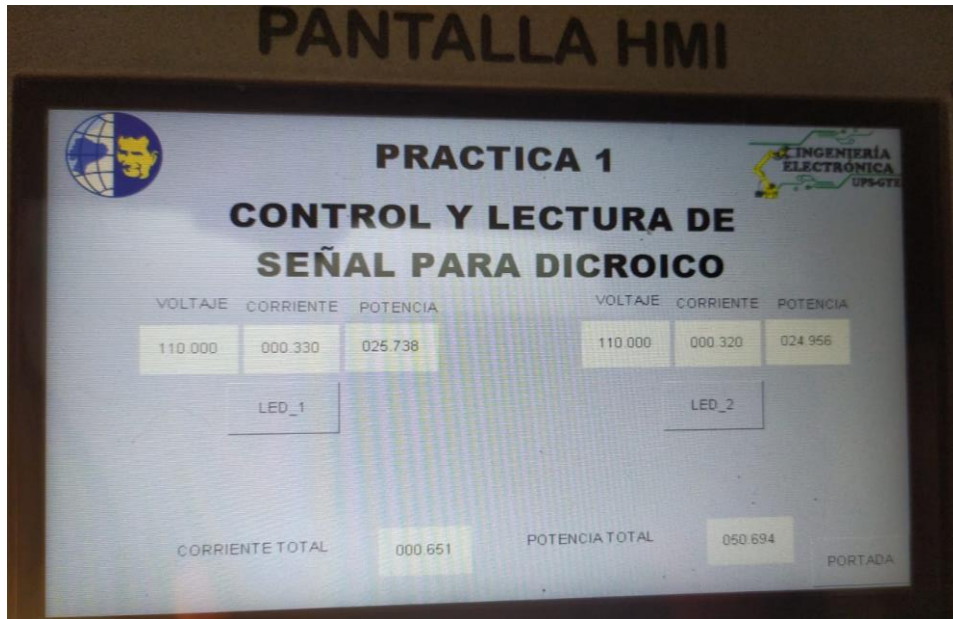


Figura 87: Visualización de consumos en Nextion de los dicroicos – Práctica

D. Registro de Resultados

8. Realizar la medición mediante una pinza para la lectura de amperaje y comparar resultado de lectura como se muestra en la Figura 88.

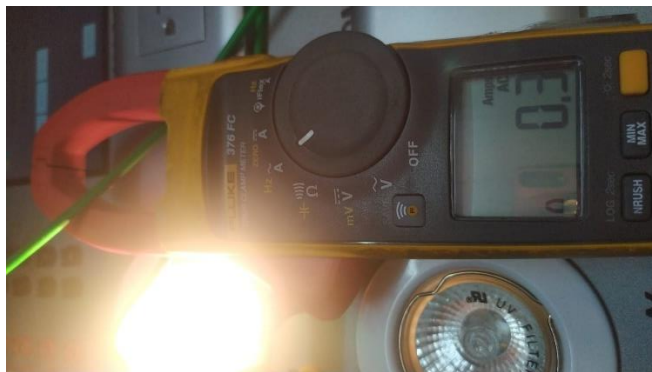


Figura 88: Visualización de consumos en pinza amperimétrica de los dicroicos – Práctica

Elemento	Dato en maleta	Dato en pinza amperimétrica
Dicroico 1	0.31 Amperios	0.3 Amperios
Dicroico 2	0.30 Amperios	0.3 Amperios

Tabla 2 :Tabla de resultados Práctica 1

G. RESOLUCIÓN DE PRÁCTICA 2

PRÁCTICA 2

NÚMERO DE ESTUDIANTES: 2

DOCENTE

ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

TIEMPO ESTIMADO: 1 HORAS

**TEMA: “CONTROL Y LECTURA DE SEÑAL PARA PUNTO
ELECTRICO.”**

A. Objetivo General

- Supervisar el consumo de un punto eléctrico

B. Objetivos Específicos

- Realizar la lectura de la corriente de un punto eléctrico
- Controlar la activación del punto eléctrico desde la pantalla NEXTION

C. Marco Procedimental

1. Realizar la conexión de acuerdo con la Figura en la maleta didáctica.

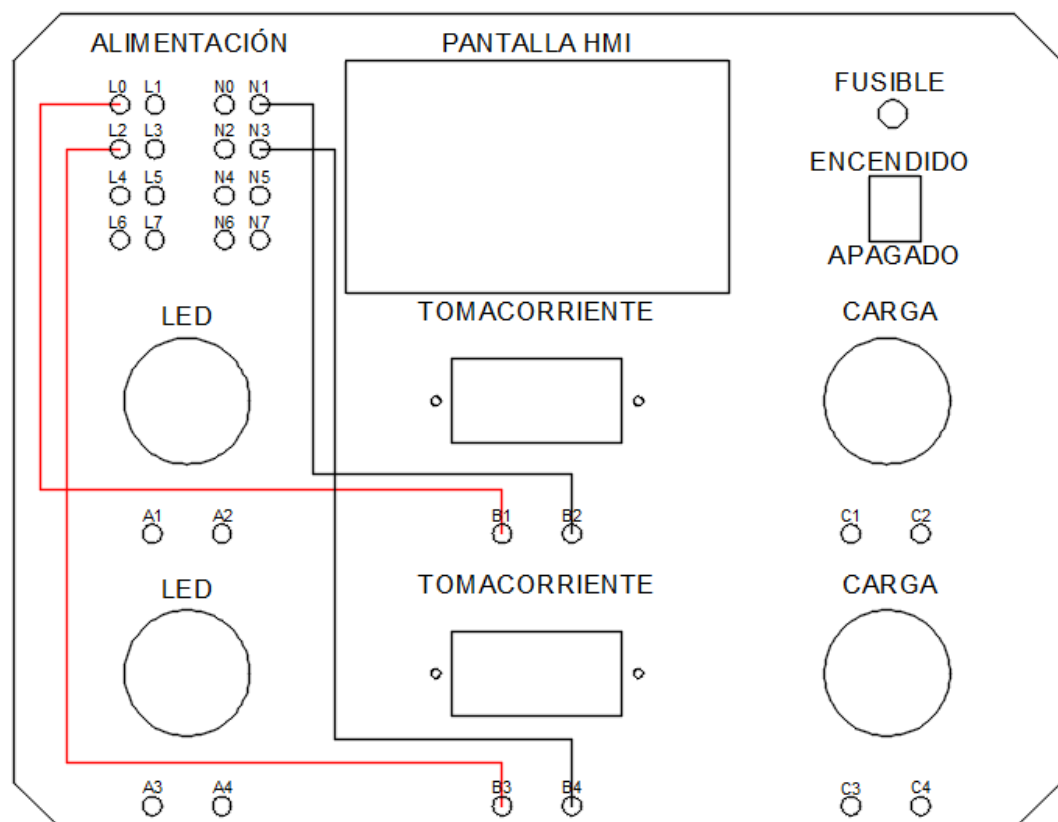


Figura 89.: Diagrama de conexión – Práctica 2

2. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido.

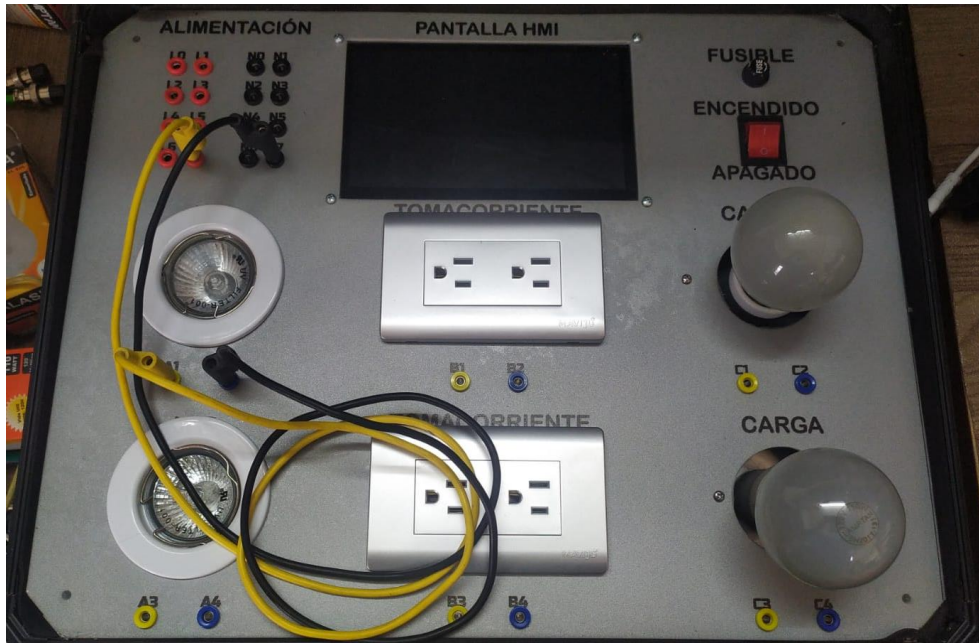


Figura 90:. Conexiones físicas y a la red 110VAC – Práctica 2

3. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido. Proceder a seleccionar en el panel de inicio de la pantalla la práctica 2

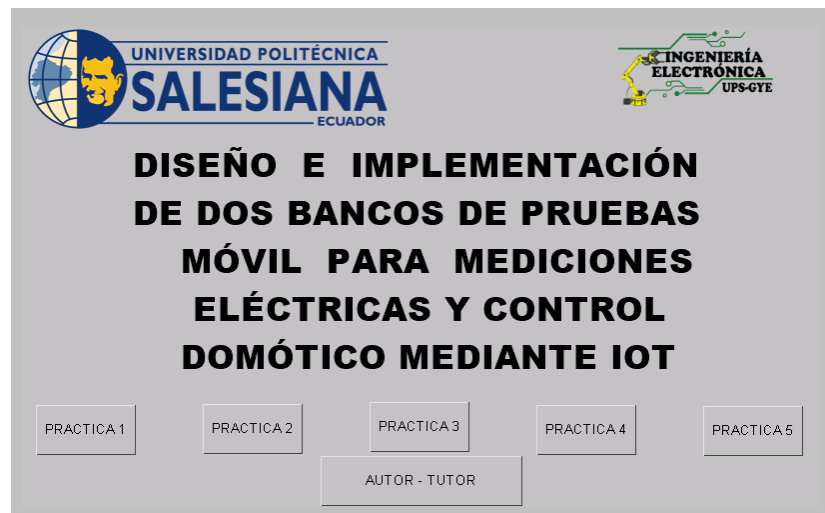


Figura 91:. Panel de inicio en Nextion – Práctica 2

4. En el panel seleccionar la activación de cada punto eléctrico mediante la pantalla táctil en un botón denominado Toma y visualizar el consumo de cada led en la pantalla.

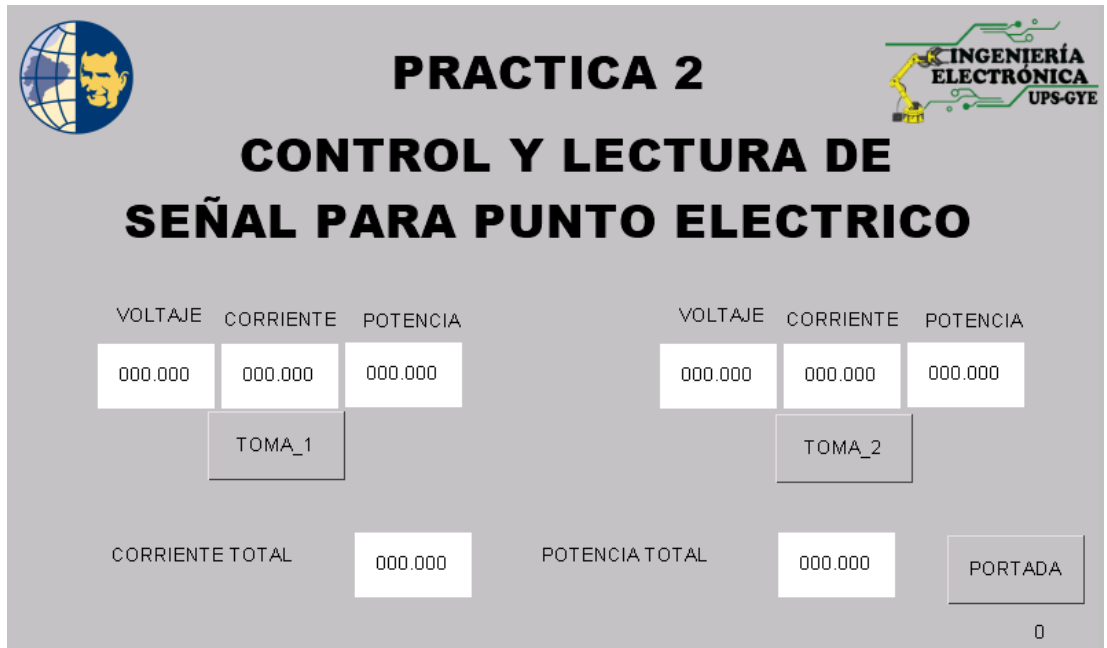


Figura 92:. Panel – Práctica 2

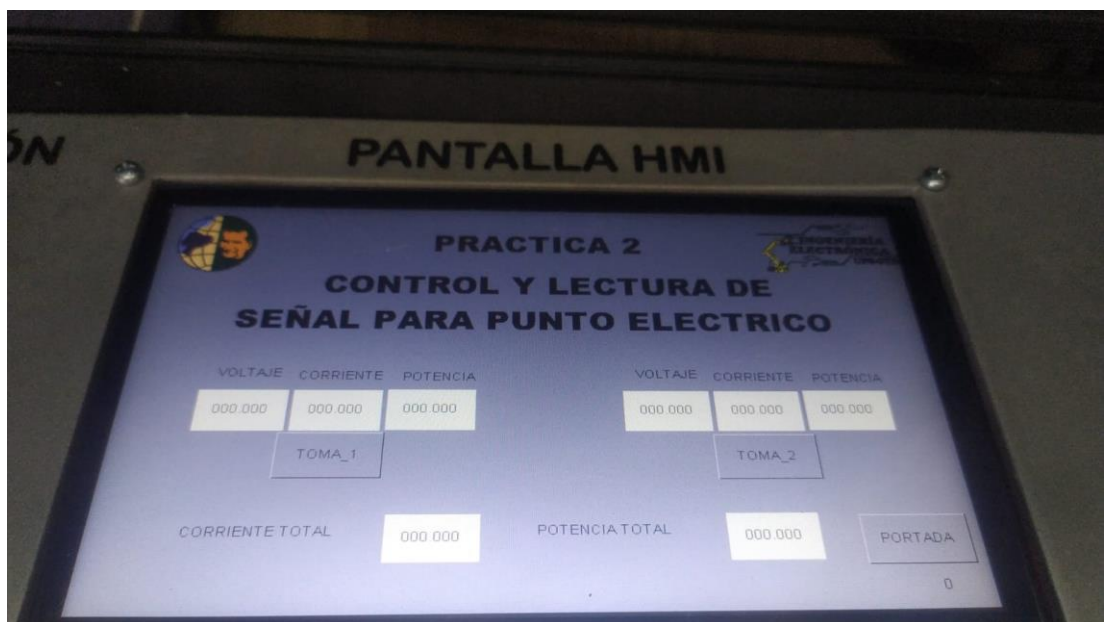


Figura 93:. Vista Física de panel – Práctica 2

5. Se procede a conectar equipos a los puntos eléctricos en esta práctica se utilizó un dremel (el equipo a conectarse puede ser cualquiera siempre que trabaje a un voltaje de 110VAC) . Se activa la alimentación en el punto eléctrico dando clic en la pantalla táctil Nextion.



Figura 94:. Activación de punto eléctrico mediante pantalla nextion –
Práctica 2

6. Al momento de activar el punto eléctrico y conectar algún equipo se visualiza en la pantalla nextion el consumo del equipo

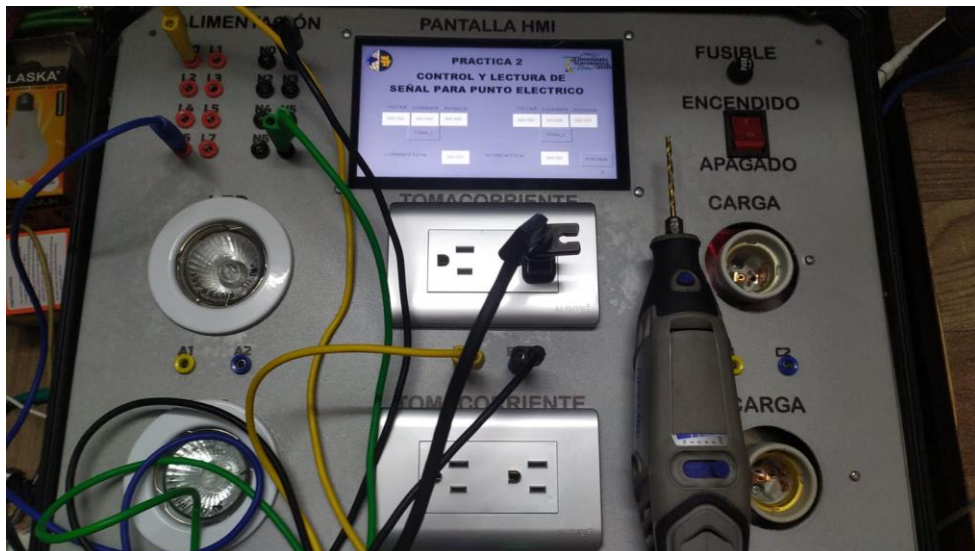


Figura 95:. Conexiones físicas – Práctica

D. Registro de Resultados

7. Al conectar la pinza de amperaje se observa en la Figura 92 la lectura del punto eléctrico y al compararse con la lectura en el módulo se observa el mismo valor de medición como se muestra en la Figura 93.

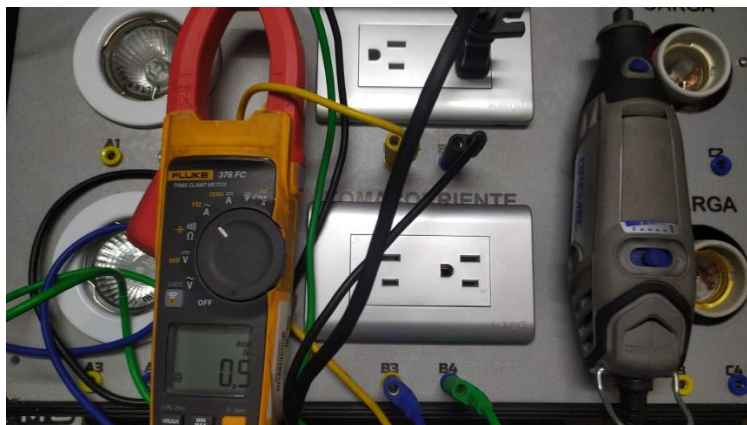


Figura 96:. Conexiones medición de pinza amperimétrica – Práctica 2

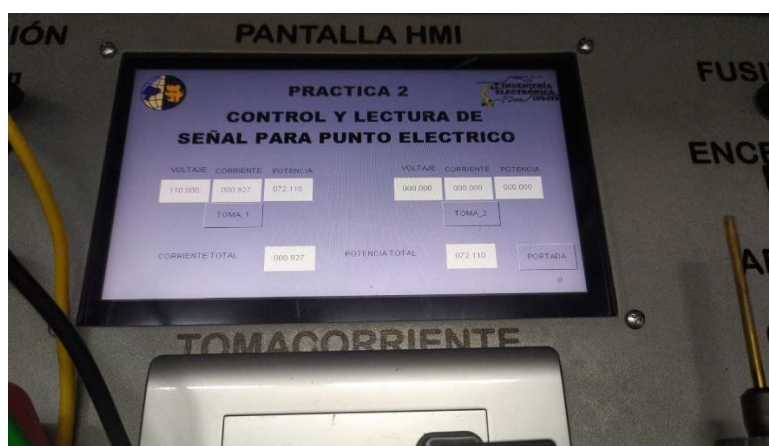


Figura 97:.. Conexiones medición de pinza amperimétrica – Práctica 2

En la Tabla 3 muestra la comparación de los datos dados por la maleta y una pinza amperimétrica.

Elemento	Dato en maleta	Dato en pinza amperimétrica
Punto eléctrico 1	0.15 Amperios	0.15 Amperios
Punto eléctrico 2	0 amperios	0 amperios

Tabla 3 :Tabla de resultados Práctica 2

H. RESOLUCIÓN DE PRÁCTICA 3

PRÁCTICA 3

NÚMERO DE ESTUDIANTES: 2

DOCENTE

ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

TIEMPO ESTIMADO: 1 HORAS

**TEMA: “CONTROL Y LECTURA DE SEÑAL PARA
CARGAS INCANDESENTE .”**

A. Objetivo General

- Supervisar el consumo de cargas incandescente

B. Objetivos Específicos

- Realizar la lectura de la corriente de cargas incandescente
- Controlar la activación de cargas incandescente desde la pantalla NEXTION

C. Marco Procedimental

1. Realizar la conexión de acuerdo con la Figura en la maleta didáctica.

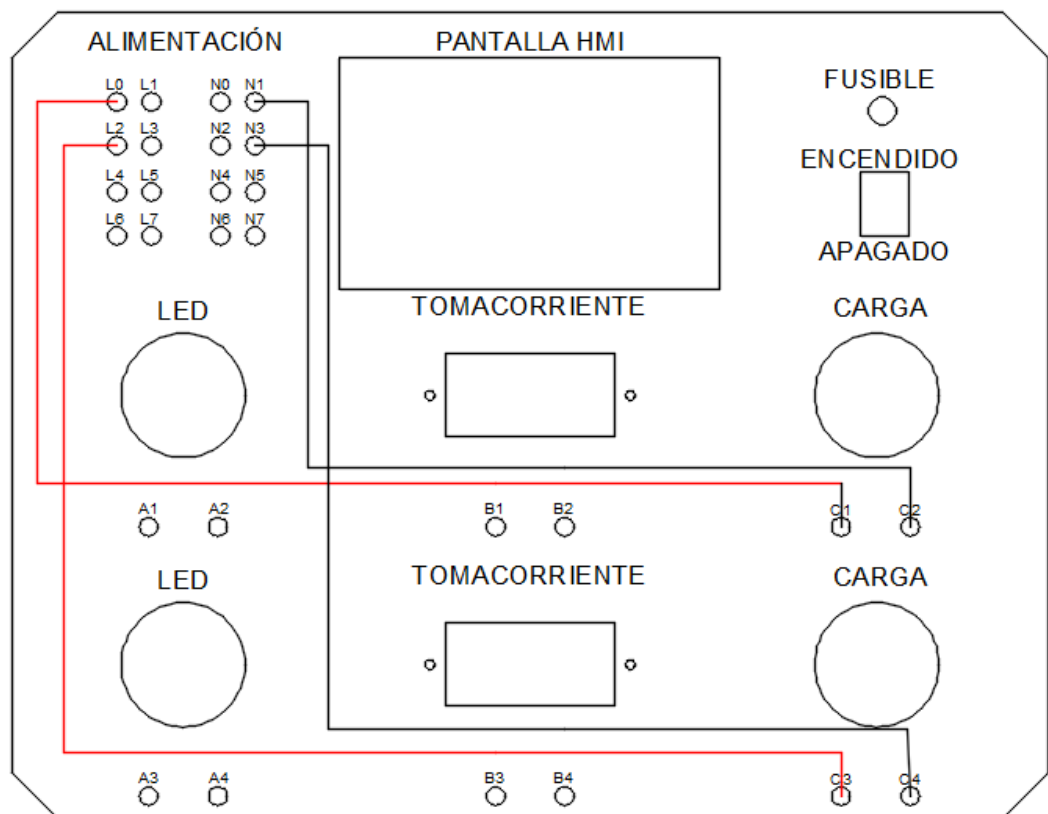


Figura 98:. Conexiones físicas – Práctica 3

- 2.** Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido.

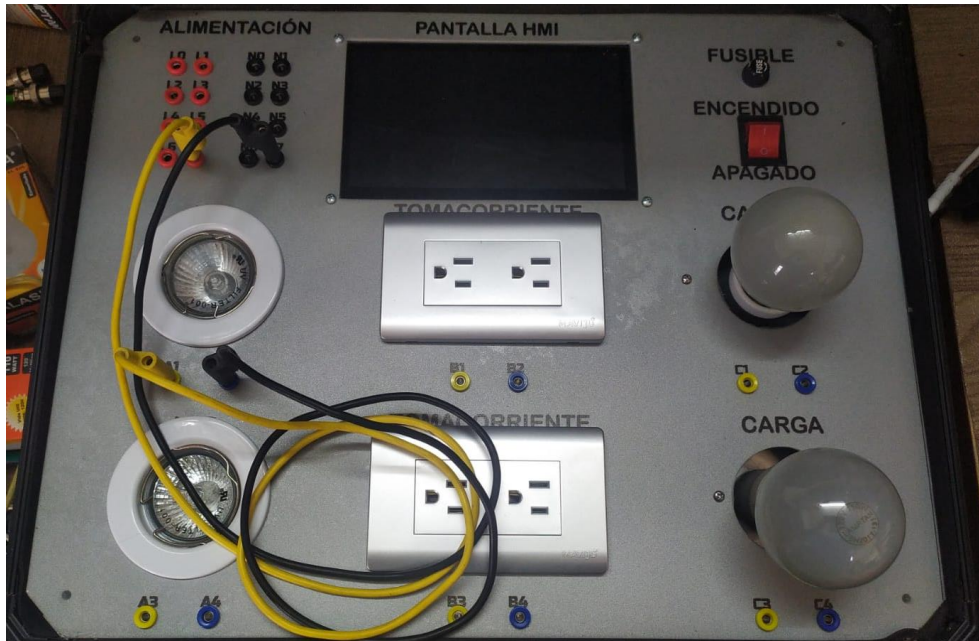


Figura 99.: Conexiones físicas – Práctica 3

- 3.** Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido. Proceder a seleccionar en el panel de inicio de la pantalla la práctica 3

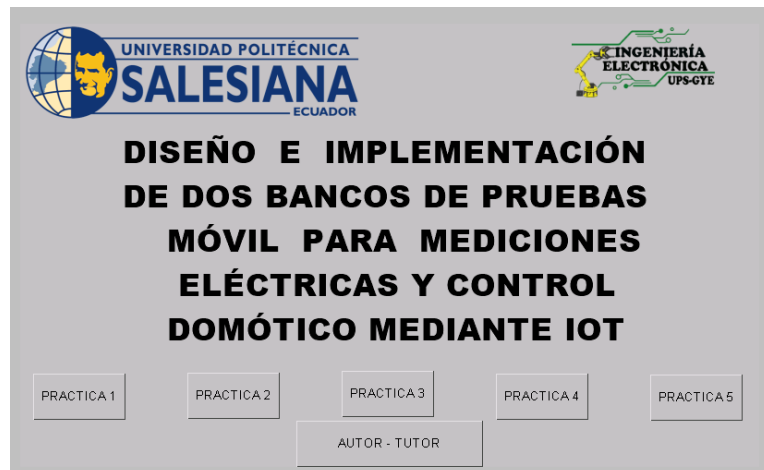


Figura 100.: Panel de inicio nextion – Práctica 3

- 4.** En el panel seleccionar la activación de cada led y visualizar el consumo de cada carga o foco incandescente en la pantalla.

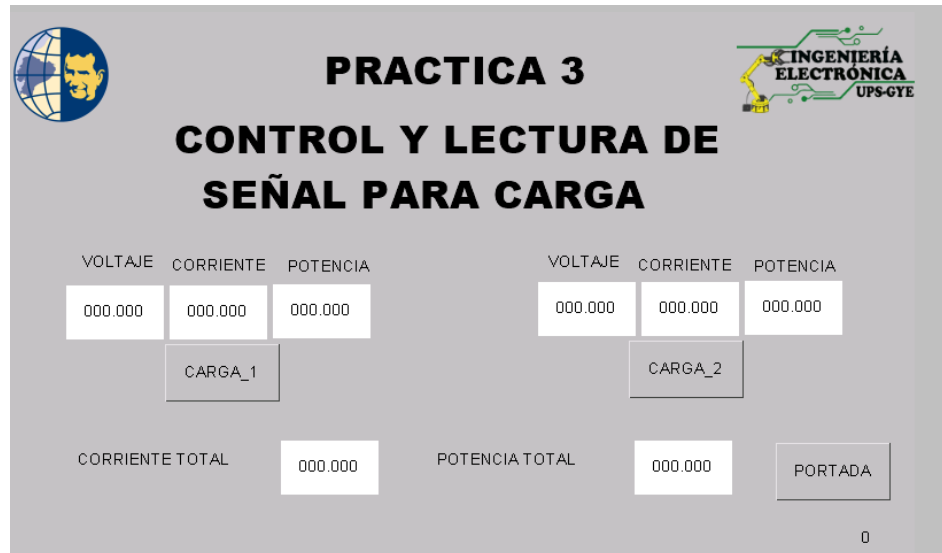


Figura 101:. Panel – Práctica 3



Figura 102:. Conexiones físicas – Práctica 3

- 5.** Activar la carga y visualizar la activación y consumo del led.



Figura 103:. Encendido de cargas – Práctica 3

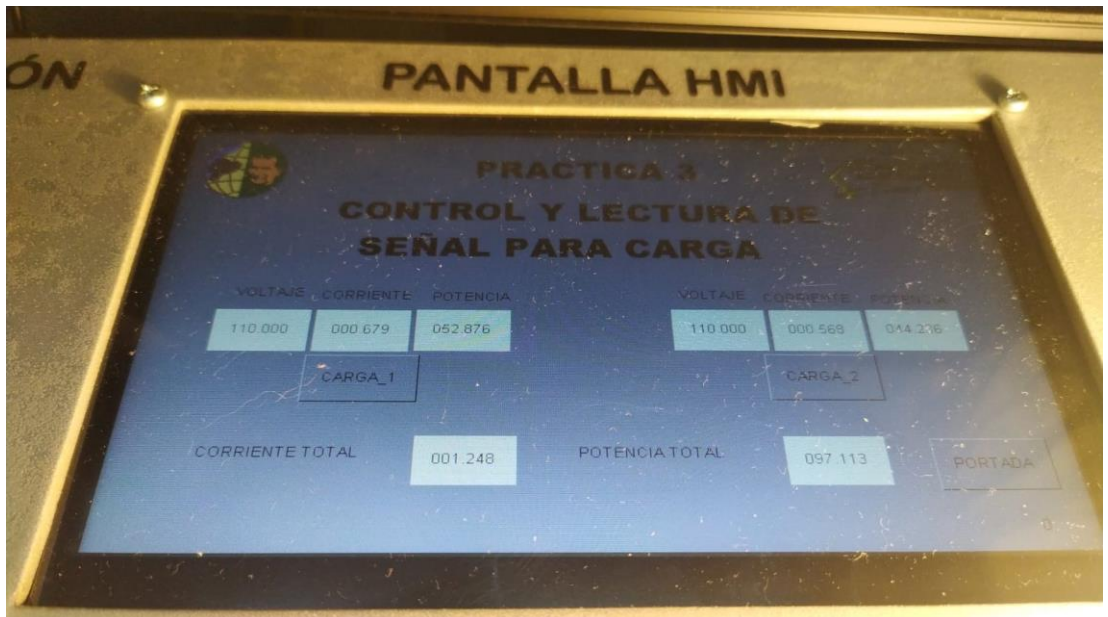


Figura 104:. Consumo de cargas – Práctica 3

D. Registro de Resultados

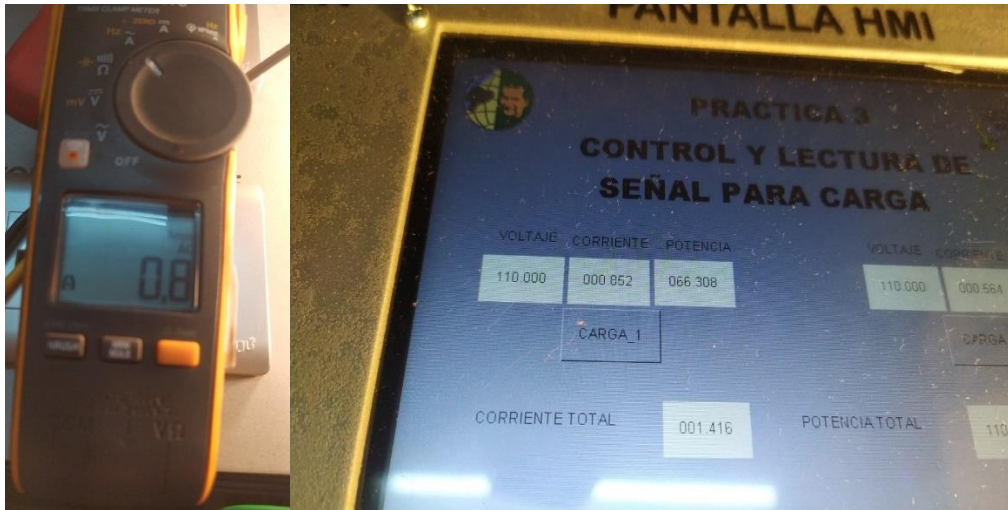


Figura 105:. Lectura y comparación con pinza amperimétrica – Práctica

En la Tabla 4 muestra la comparación de los datos dados por la maleta y una pinza amperimétrica.

Elemento	Dato en maleta	Dato en pinza amperimétrica
Carga 1	0.92 Amperios	0.9 Amperios
Carga 2	0.52 Amperios	0.5 Amperios

Tabla 4 :Tabla de resultados Práctica 3

I. RESOLUCIÓN DE PRÁCTICA 4

PRÁCTICA 4

NÚMERO DE ESTUDIANTES: 2

DOCENTE

ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

TIEMPO ESTIMADO: 1 HORAS

TEMA: “CONTROL Y LECTURA DESDE UBIDOTS.”

A. Objetivo General

- Realizar la activación desde plataforma Ubidots de los elementos en la maleta didáctica

B. Objetivos Específicos

- Realizar la lectura del consumo de los elementos
- Controlar la activación desde Ubidots y visualizar consumo en Nextion

C. Marco Procedimental

1. Realizar la conexión de acuerdo con la Figura en la maleta didáctica.

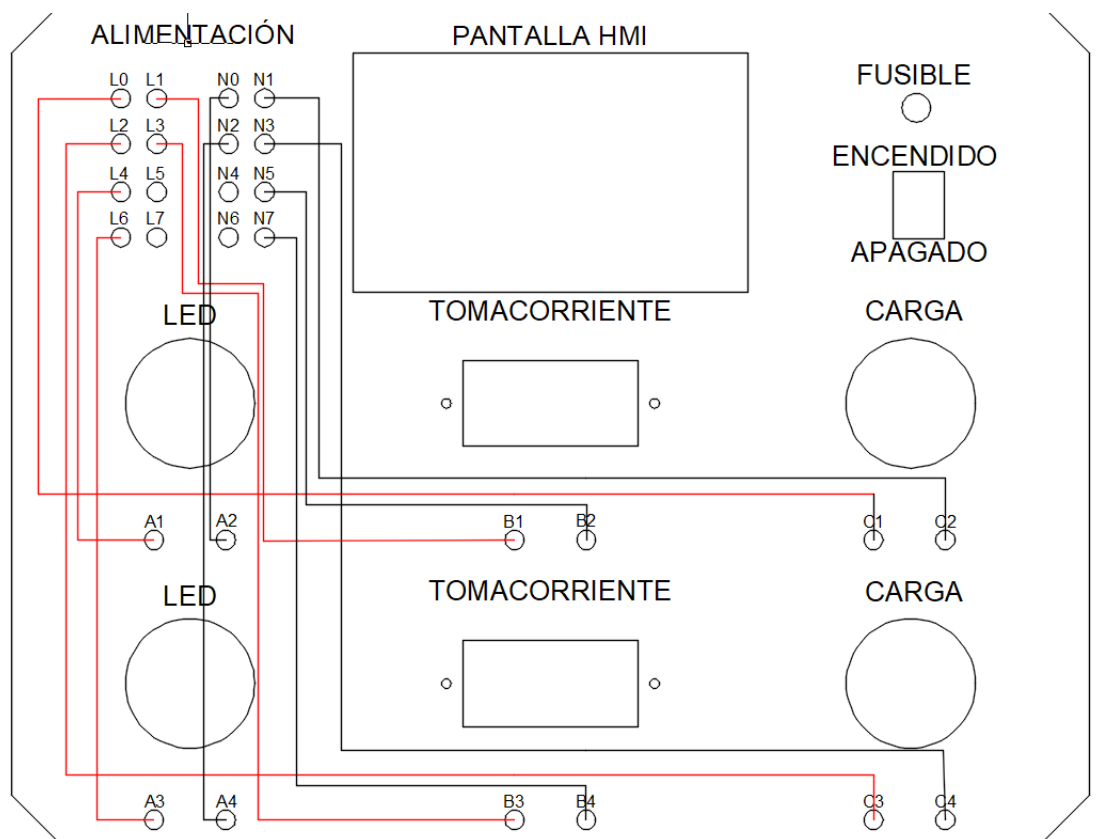


Figura 106:. Conexiones físicas – Práctica 4

2. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido.



Figura 107: Conexiones físicas – Práctica 4

3. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido. Proceder a seleccionar en el panel de inicio de la pantalla la práctica 4



Figura 108: Panel de inicio – Práctica 4

4. En el panel que se muestra en la Figura es donde se visualizarán el consumo de corriente y la potencia de los led dicroicos 1 y 2, puntos eléctricos 1 y 2, cargas incandescentes 1 y 2.

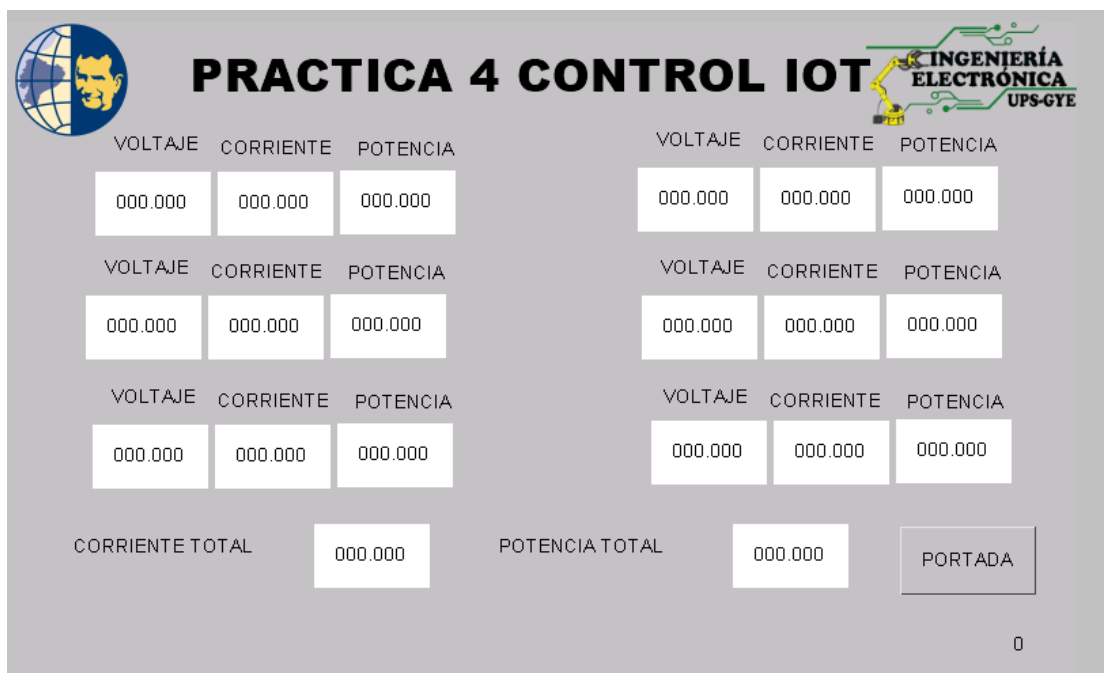


Figura 109:. Panel – Práctica 4

5. Se procede a ingresar al link de la práctica 4
<https://industrial.ubidots.com/app/dashboards/public/dashboard/f1rKXPJBRSKhXJHkhy-Uc6qWTXy2G55NgX00vrEDBPc>

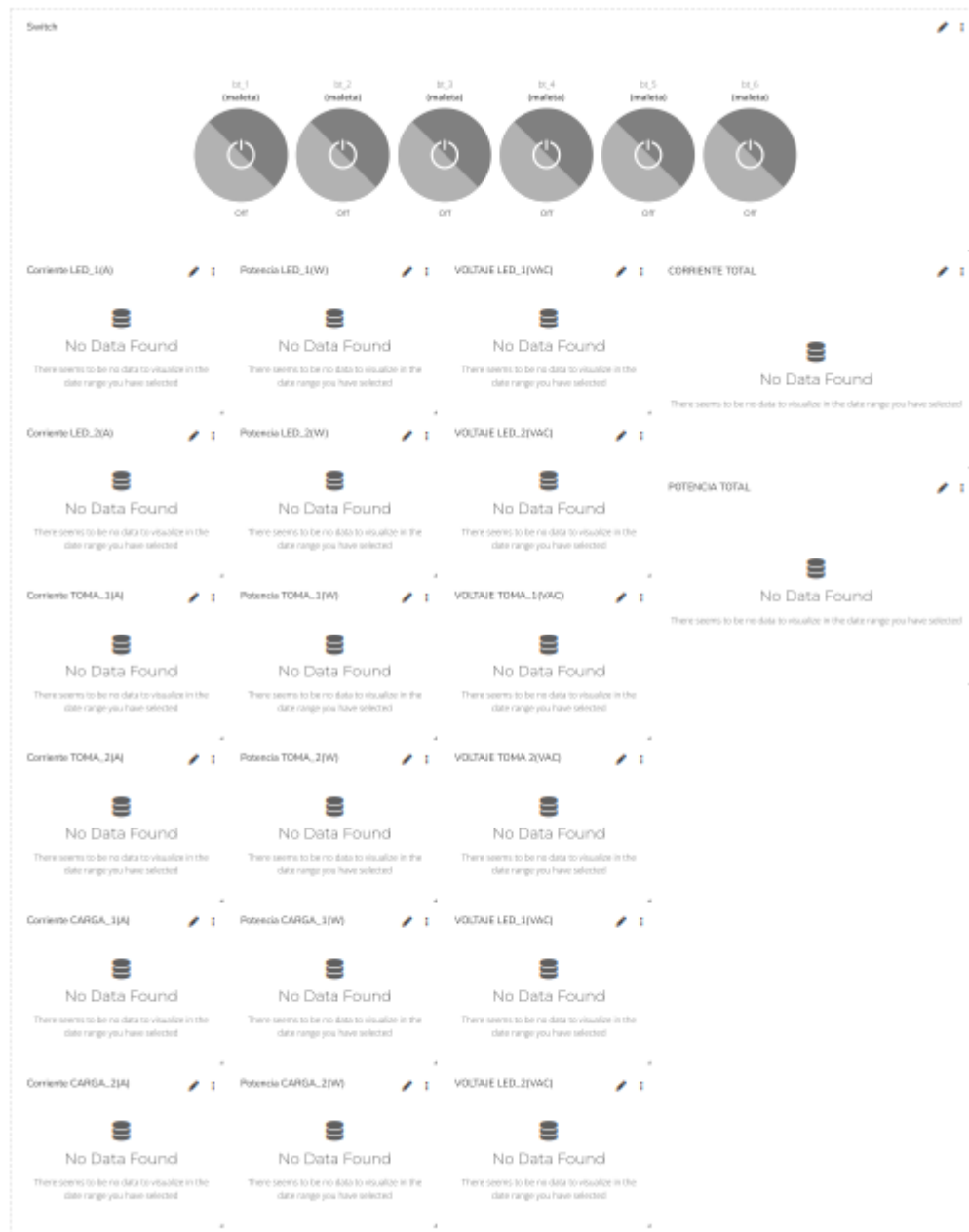


Figura 110:. Panel en Ubidots sin conexión– Práctica 4

6. Al iniciar con la práctica el panel en Ubidots cambiara de estado y se visualizaran los datos de corriente de cada elemento, en conjunto la potencia.

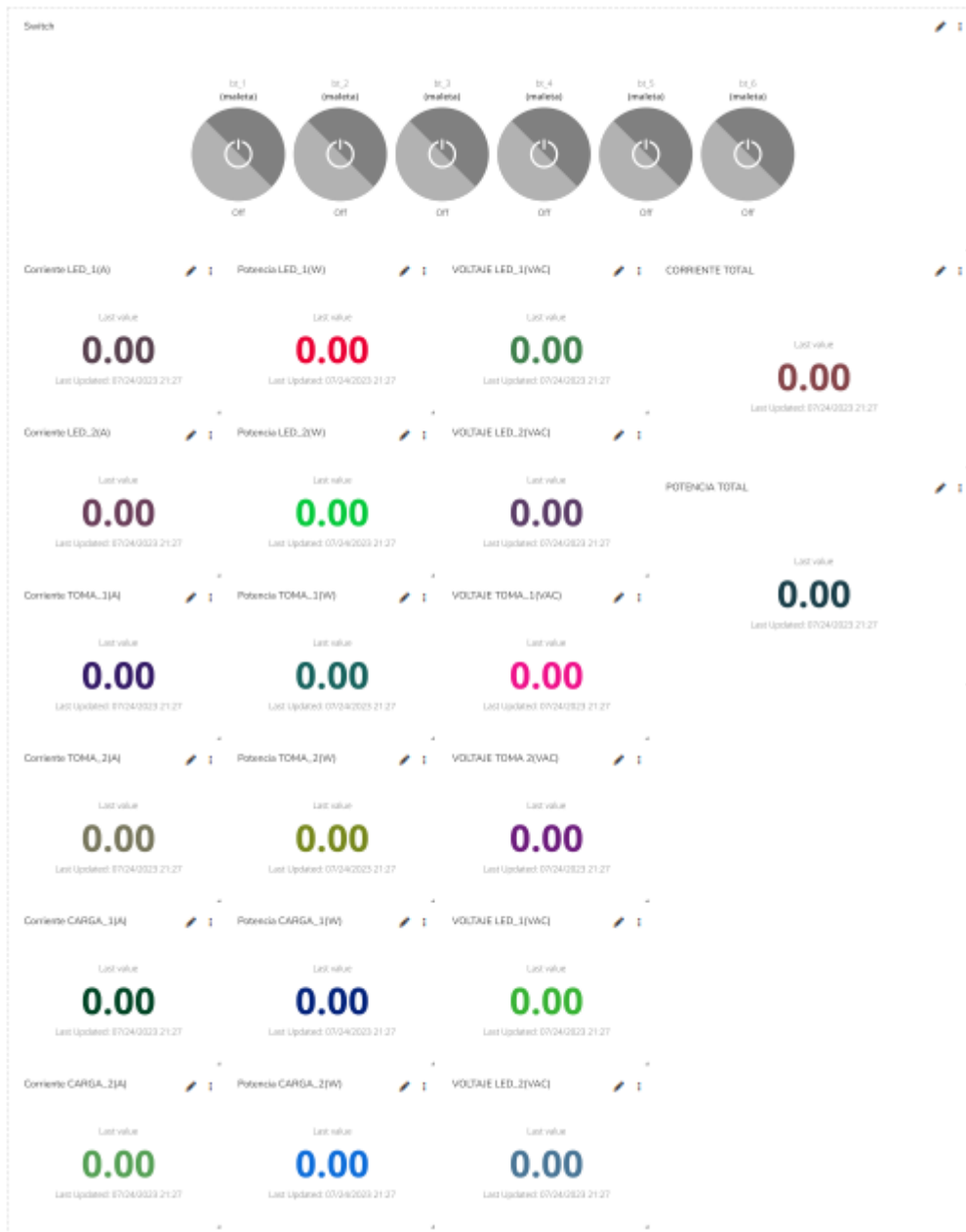


Figura 111.: Panel en Ubidots con conexión– Práctica 4

7. Se da clic en la pantalla de Ubidots en la parte superior se encontraran botones vinculados a cada elemento en la maleta como se observa en la figura.

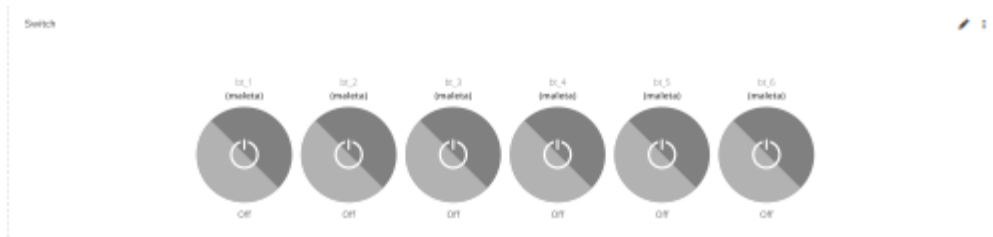


Figura 112:. Panel en Ubidots con conexión– Práctica 4

8. Se procede a dar clic en el botón 1 para encender el dicroico como se muestra en la Figura , posterior se visualizara la activación como en Figura

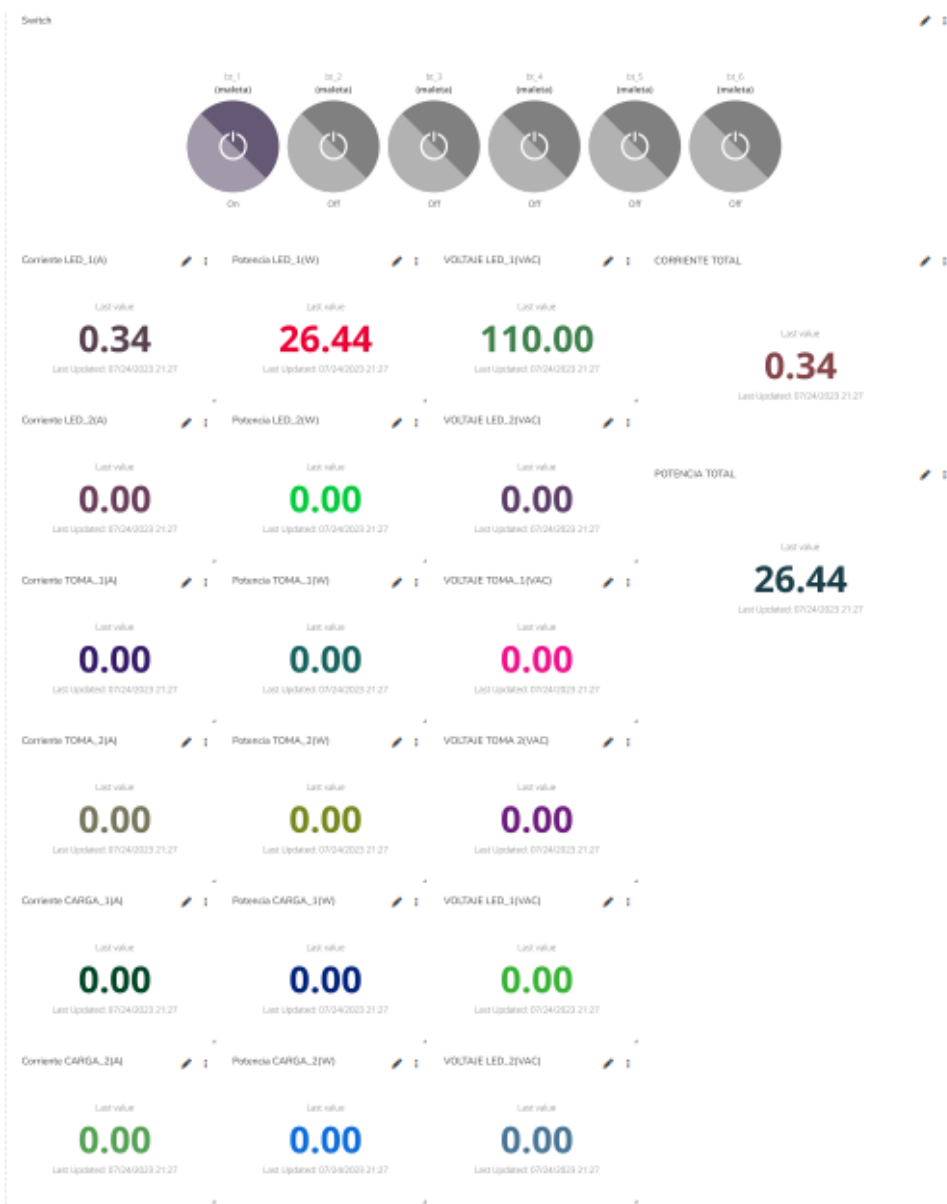


Figura 113:. Panel en Ubidots encendido de dicroico – Práctica 4

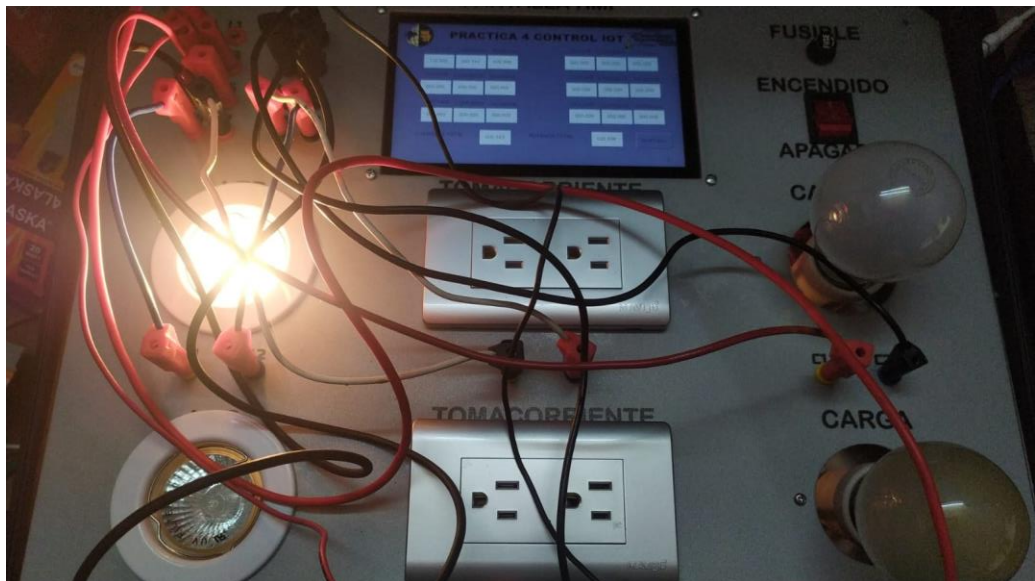


Figura 114:. Encendido de dicrico desde Ubidots– Práctica 4

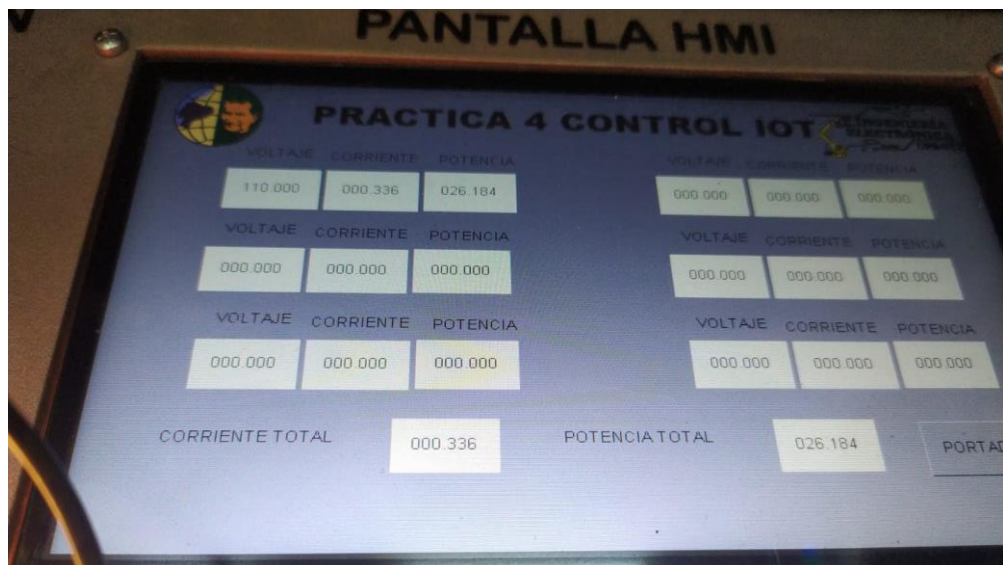


Figura 115:. Panel nextion con señal desde Ubidots– Práctica 4

9. Se procede a encender todo los elementos y visualizar el consumo en la plataforma como se muestra en la Figura .

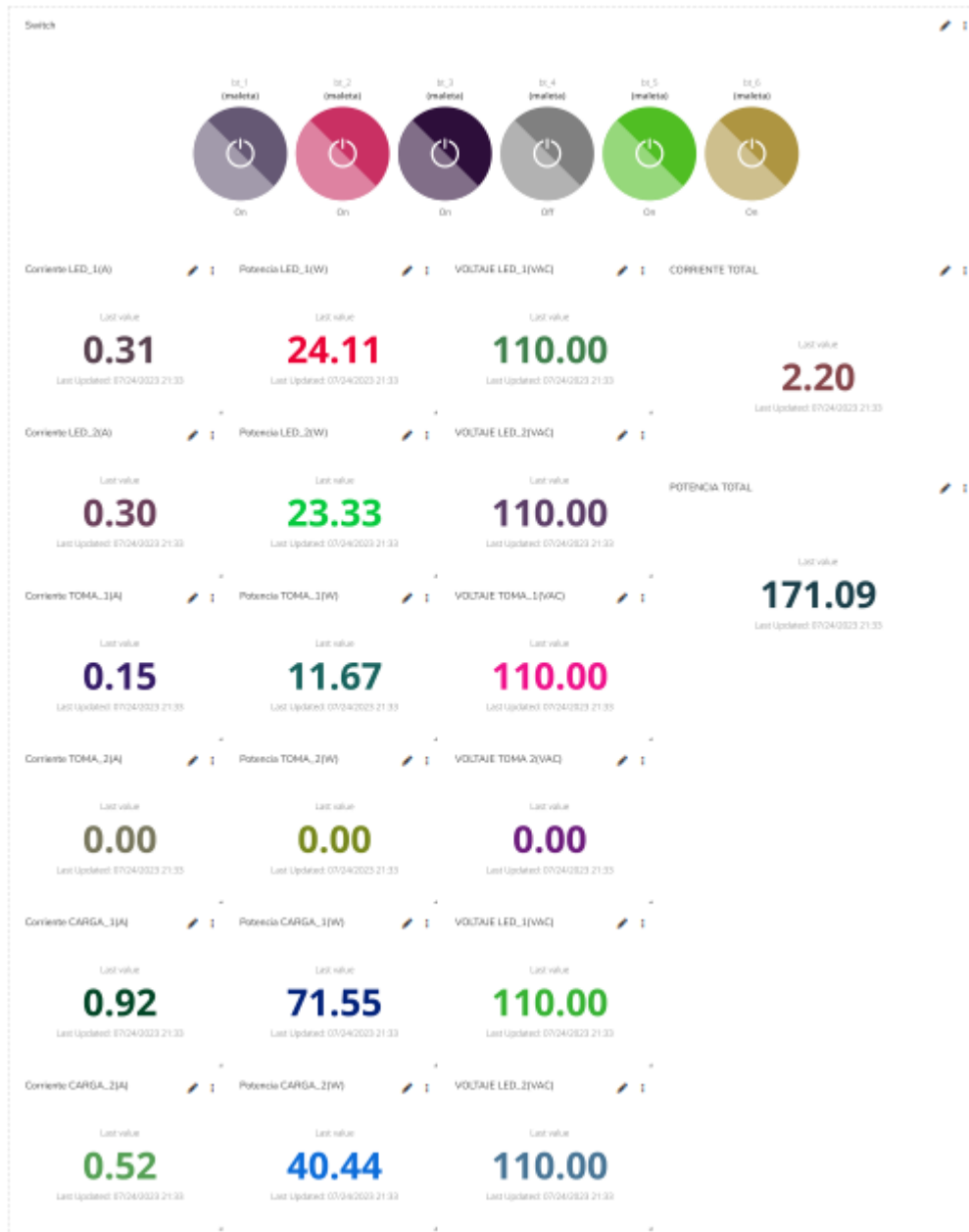


Figura 116.: Panel nextion con señal desde Ubidots– Práctica 4

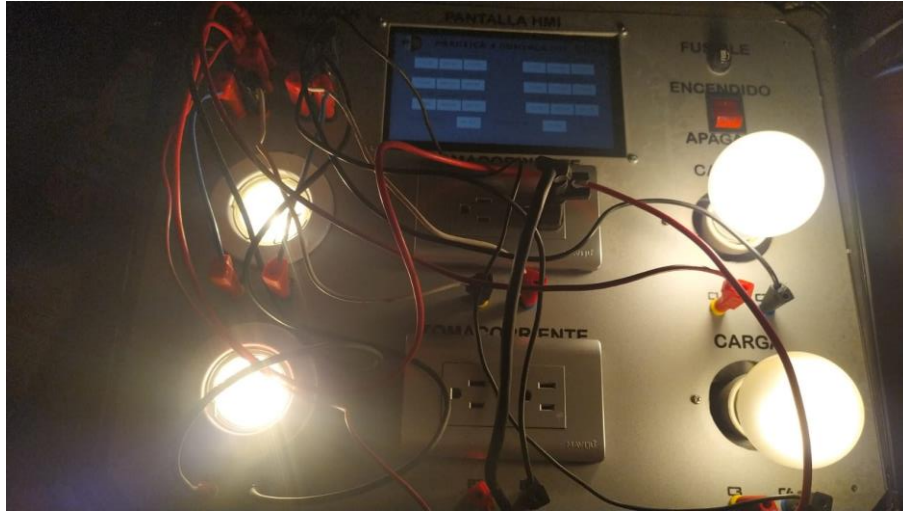


Figura 117: Panel nextion con señal desde Ubidots– Práctica 4

D. Registro de Resultados

En la Tabla 5 muestra la comparación de los datos dados por la maleta y una pinza amperimétrica.

Elemento	Dato en maleta	Dato en pinza amperimétrica
Dicroico 1	0.31 Amperios	0.3 Amperios
Dicroico 2	0.30 Amperios	0.3 Amperios
Punto eléctrico 1	0.15 Amperios	0.15 Amperios
Punto eléctrico 2	0 amperios	0 amperios
Carga 1	0.92 Amperios	0.9 Amperios
Carga 2	0.52 Amperios	0.5 Amperios

Tabla 5 :Tabla de resultados Práctica 4

J. RESOLUCIÓN DE PRÁCTICA 5

PRÁCTICA 5

NÚMERO DE ESTUDIANTES: 2

DOCENTE

ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

TIEMPO ESTIMADO: 1 HORAS

**TEMA: “CONTROL Y LECTURA DE MALETA
MEDIANTE IoT Y NEXTION.”**

A. Objetivo General

- Supervisar el consumo de la maleta

B. Objetivos Específicos

- Realizar la activación de los elementos mediante la pantalla NEXTION
- Visualizar el consumo de los elementos en la pantalla NEXTION
- Realizar la activación de los elementos mediante la plataforma Ubidots
- Visualizar el consumo de los elementos en la plataforma Ubidots

C. Marco Procedimental

1. Realizar la conexión de acuerdo con la Figura en la maleta didáctica.

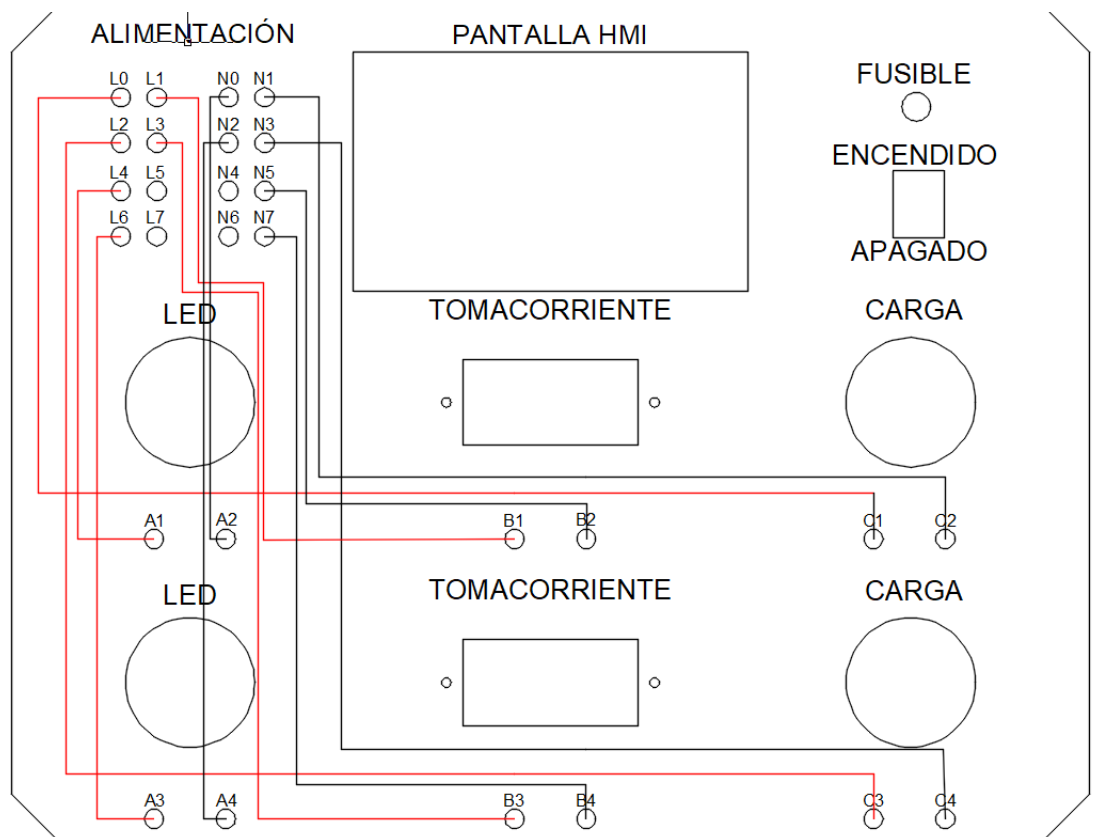


Figura 118.: Conexiones físicas – Práctica 5

2. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido.



Figura 119:. Conexiones físicas – Práctica 5

3. Conectar el módulo a una alimentación de 110VAC y cambiar el estado del switch a encendido. Proceder a seleccionar en el panel de inicio de la pantalla la práctica 5



Figura 120:. Conexiones físicas – Práctica 5

4. En el panel que se muestra en la Figura es donde se visualizarán el consumo de corriente y la potencia de los leds dicroicos 1 y 2, puntos eléctricos 1 y 2, cargas incandescentes 1 y 2.



Figura 121: Panel – Práctica 5

5. Se procede a ingresar al link de la práctica 4

<https://industrial.ubidots.com/app/dashboards/public/dashboard/EZfWUyYxVWsmmt4Fr-4AFPRyGFrLw6AiRxfQXgAFCpl>

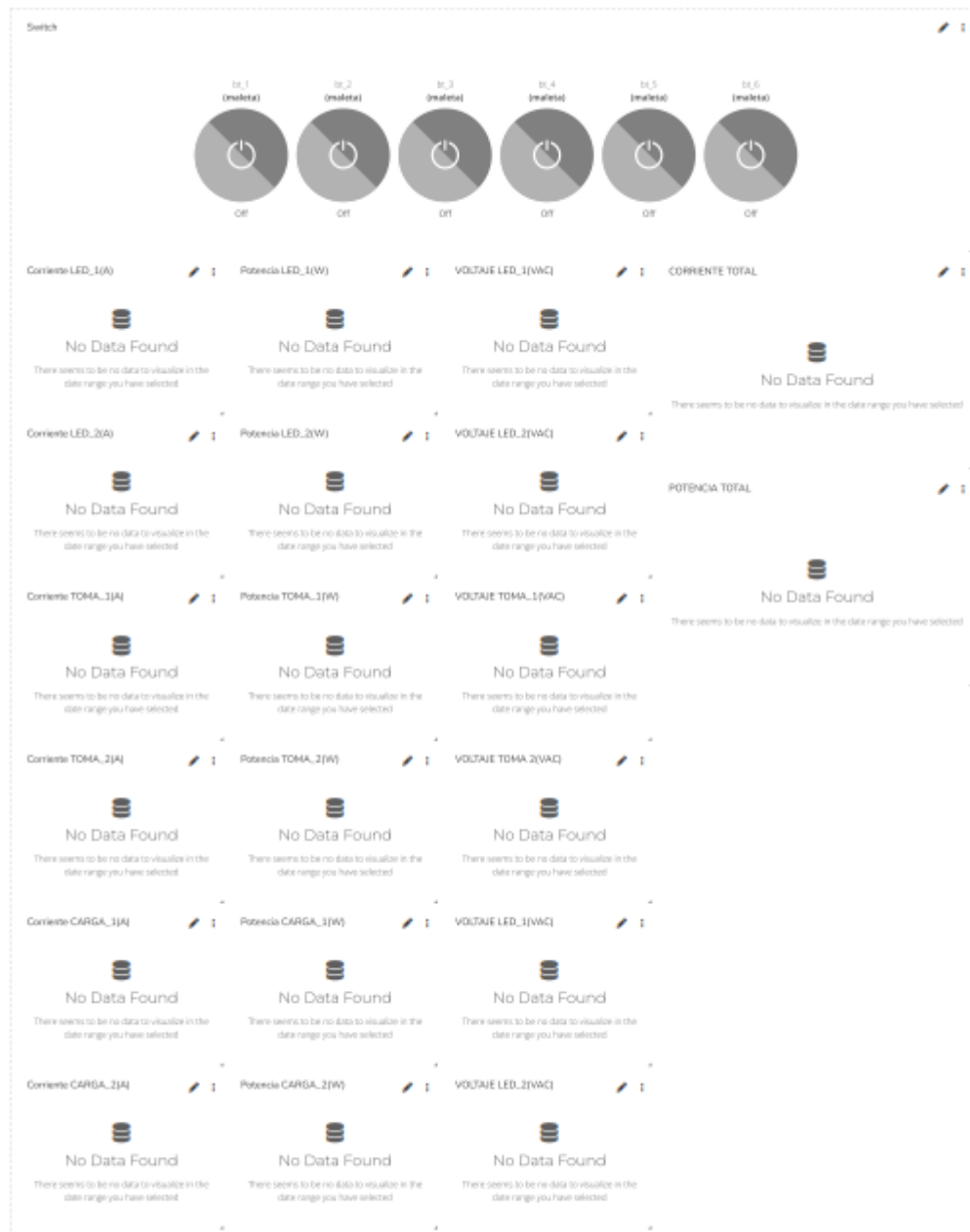


Figura 122:. Panel en Ubidots sin conexión– Práctica 5

6. Al iniciar con la práctica el panel en Ubidots cambiara de estado y se visualizaran los datos de corriente de cada elemento, en conjunto la potencia.

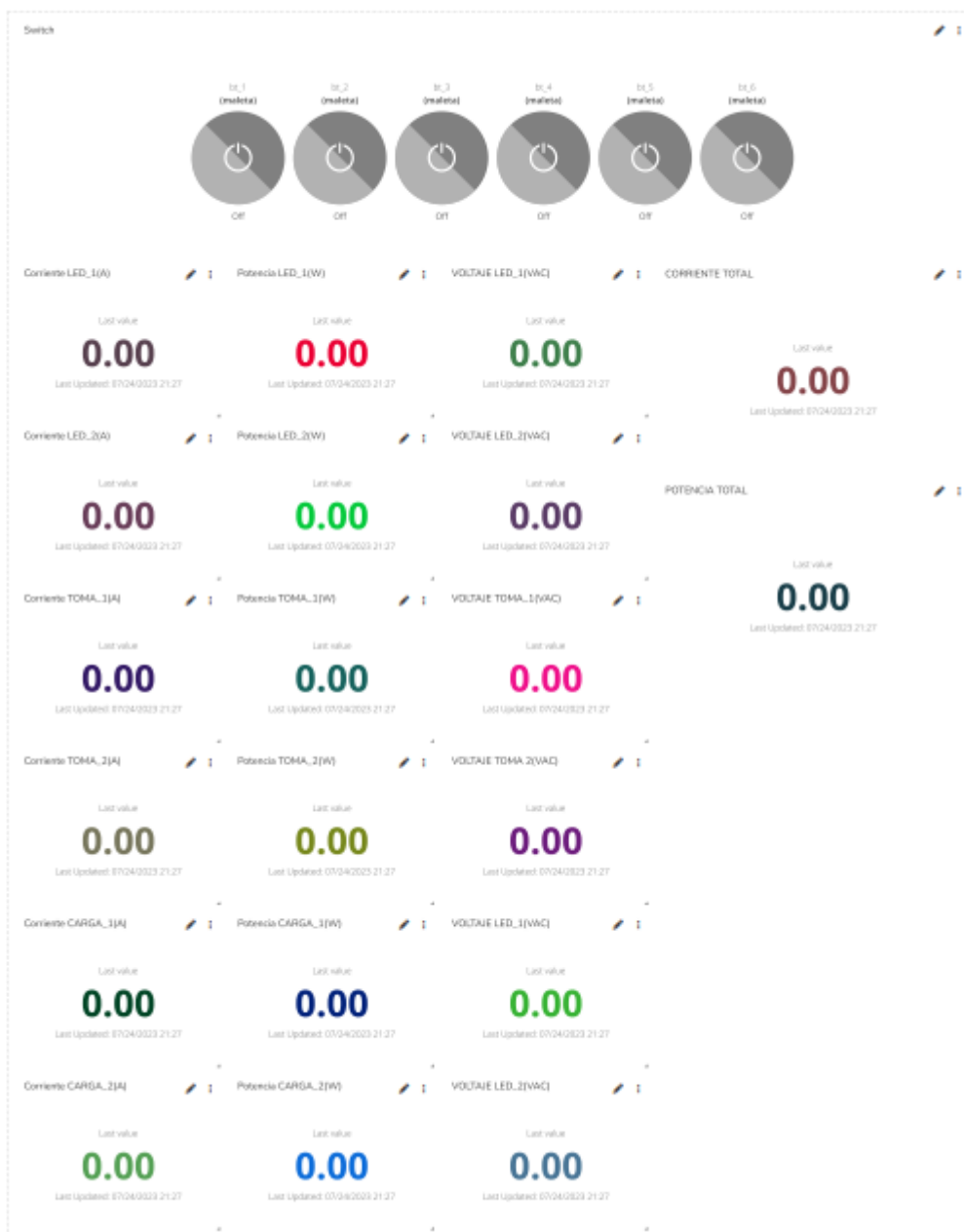


Figura 123:. Panel en Ubidots con conexión– Práctica 5

7. Se da clic en la pantalla de Ubidots en la parte superior se encontrarán botones vinculados a cada elemento en la maleta como se observa en la figura.

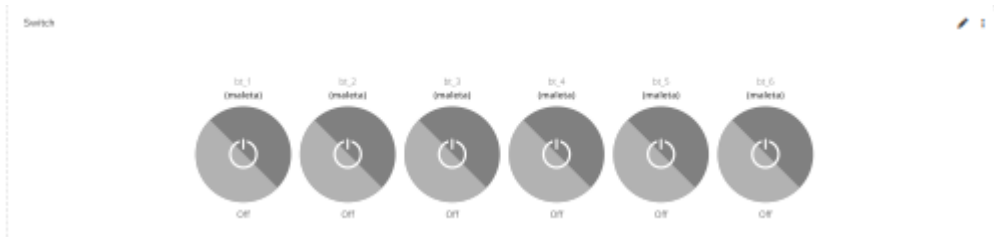


Figura 124: Panel en Ubidots con conexión– Práctica 5

8. Se da clic en la pantalla de NEXTION en la parte superior se encontrarán botones vinculados a cada elemento en la maleta como se observa en la figura.



Figura 125: Panel en Nextion– Práctica 5

9. Se procede a dar clic en el botón 1 para encender el dicroico como se muestra en la Figura , posterior se visualizara la activación como en Figura

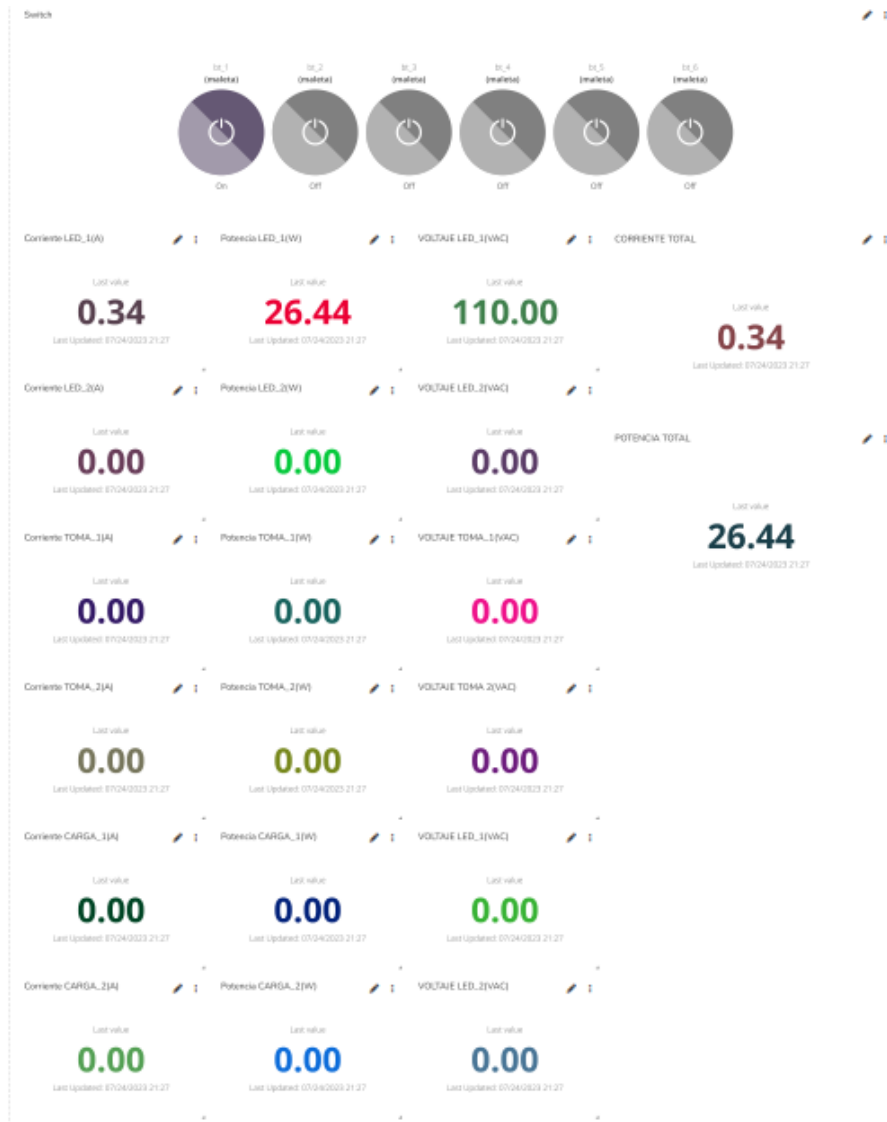


Figura 126: Panel en Ubidots encendido de microico – Práctica 5

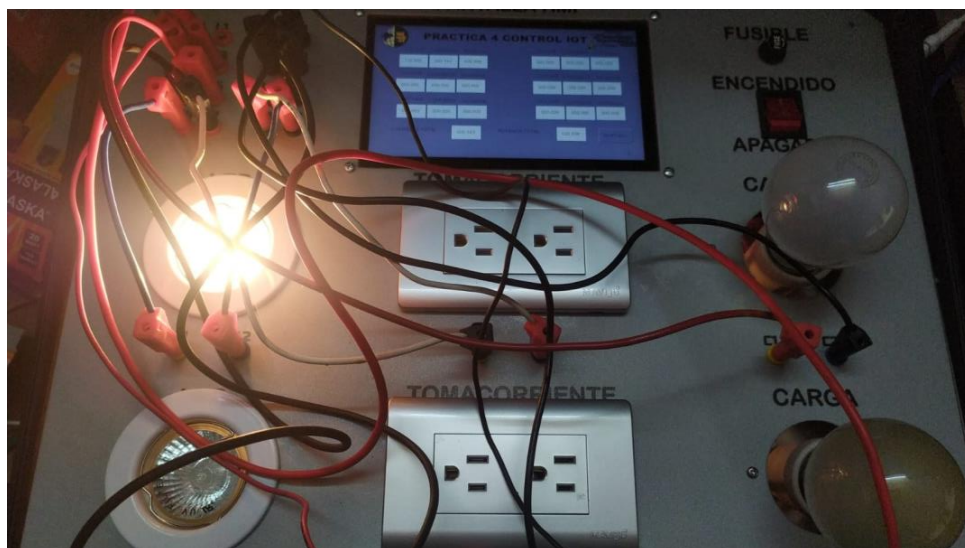


Figura 127: Encendido de microico desde Ubidots– Práctica 5

10. Se procede a encender todo los elementos y visualizar el consumo en la plataforma como se muestra en la Figura 128.

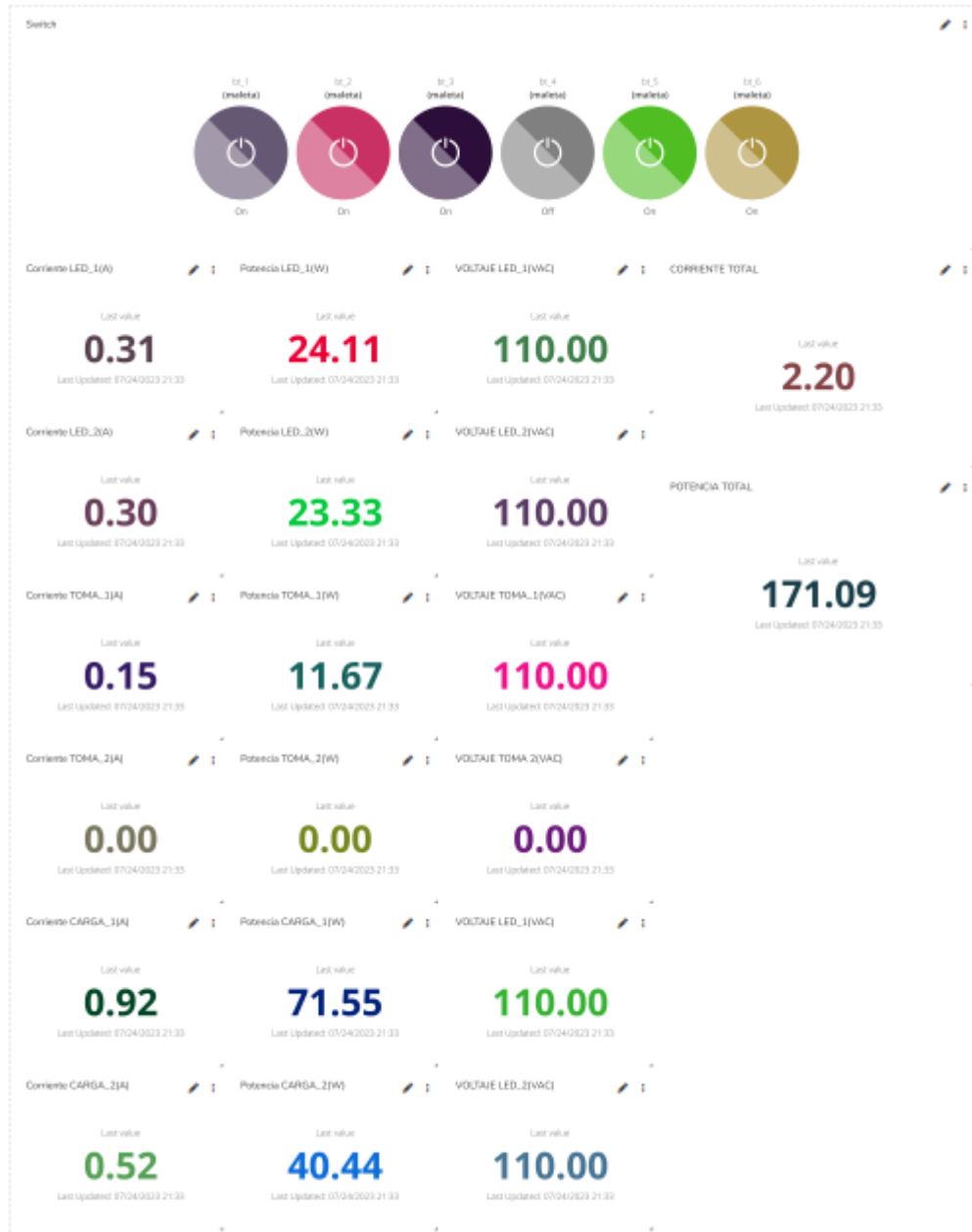


Figura 128: Panel nextion con señal desde Ubidots– Práctica 5

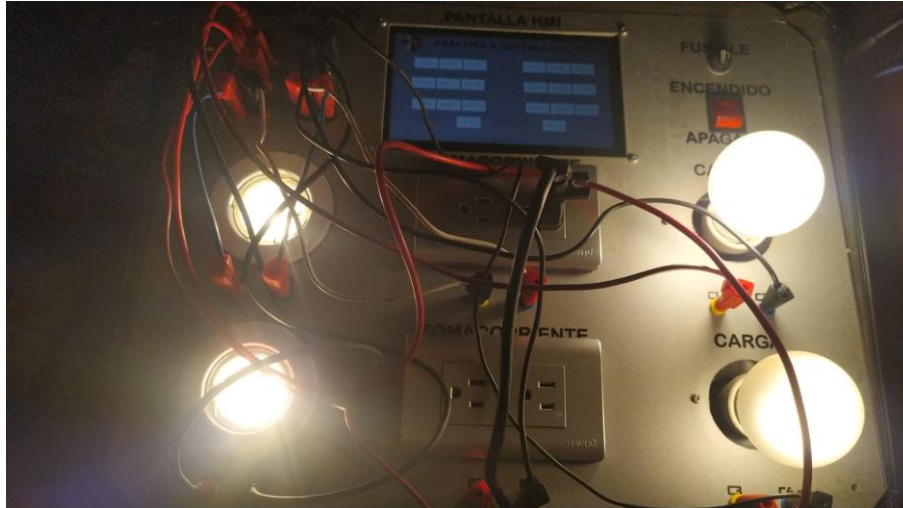


Figura 129:.. Panel nextion con señal desde Ubidots– Práctica 5

D. Registro de Resultados

En la Tabla 6 muestra la comparación de los datos dados por la maleta y una pinza amperimétrica.

Elemento	Dato en maleta	Dato en pinza amperimétrica
Dicroico 1	0.31 Amperios	0.3 Amperios
Dicroico 2	0.30 Amperios	0.3 Amperios
Punto eléctrico 1	0.15 Amperios	0.15 Amperios
Punto eléctrico 2	0 amperios	0 amperios
Carga 1	0.92 Amperios	0.9 Amperios
Carga 2	0.52 Amperios	0.5 Amperios

Tabla 6 :Tabla de resultados Práctica 5