

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**PROYECTO INTEGRADOR**

**DISEÑO DE UN PROTOTIPO DE MONITOREO Y CONTROL  
DE FACTOR DE POTENCIA PARA UN PARQUE SOLAR  
EMPLEANDO ALGORITMO DE APRENDIZAJE  
AUTOMATICO EN EL BORDE**

Previo la obtención del Título de:

**Ingeniero en Electrónica y Automatización**

Presentado por:

Adrián Vidal Bazurto Onofre

Daniel Robinson Gusnay Carrillo

GUAYAQUIL – ECUADOR

Año: 2021-2022

# DEDICATORIA

A Dios por haber bendecido a mi familia y a mí con salud durante el transcurso de mi carrera Universitaria.

A mi madre Maritza Adelaida Onofre que siempre estuvo a mi lado aconsejándome y sacrificándose, pasando muchas malas noches por mí. Mi padre Vidal Valentín Bazurto que no solo me dio su nombre, también que ha sido mi ejemplo para seguir, siempre tiene algo nuevo que enseñarme. Por último, pero no menos importante a mi hermana Bianca Valentina Bazurto, aunque no siempre nos llevemos bien siempre me apoyo en todo.

Fui capaz de llegar gracias a ellos, por lo que mi tesis y la culminación de esta etapa se le dedico.

**Adrián Vidal Bazurto Onofre**

# DEDICATORIA

En primer lugar, a Dios por darme salud y permitirme culminar este peldaño académico y por tener aún a mi familia junto a mí.

En segundo lugar, dedico esta tesis a toda mi familia Gusnay y Carrillo en especial a mi padre José Elías Gusnay y a mi madre María Juana Carrillo por creer en mí y apoyarme en cada momento en las decisiones y proyectos que llevé a cabo durante esta etapa.

Un agradecimiento a mis profesores quienes me brindaron de sus conocimientos aportándome en mi crecimiento profesional.

**Daniel Robinson Gusnay Carrillo**

## **AGRADECIMIENTOS**

Un agradecimiento sincero a mi Tutor de Tesis MSc. Víctor Asanza A. por su esfuerzo arduo durante la realización de esta tesis, por ayudarme a desarrollar mis habilidades y conocimiento en diferentes ámbitos laborales. A mi compañero de Tesis Daniel que nos sacó de más de un apuro con sus conocimientos.

A todos mis profesores, que pusieron su granito de arena para mi crecimiento académico, en especial a Ing. Víctor Arce, Ing. Tonny Toscano, Ing. María Carrillo y Doctora Rebeca Estrada, quienes me han apoyado, aconsejado y dado la oportunidad de explotar mis habilidades.

Por último, pero no menos importante quisiera agradecer a la señorita Natacha Thalía Suplewiche Pacha, que durante gran parte de mi vida universitaria estuvo a mi lado, haciéndome compañía en las largas jornadas y siempre viendo por mi bienestar.

**Adrián Vidal Bazurto Onofre**

## **AGRADECIMIENTOS**

En primer lugar, un agradecimiento sincero a mi Tutor de Tesis, el MSc. Víctor Asanza A., por su tiempo, ayuda y colaboración para la elaboración del presente proyecto.

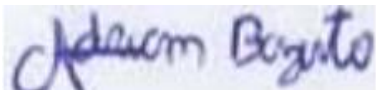
En segundo lugar, a mi compañero de tesis por su dedicación, esfuerzo y paciencia en la elaboración de este proyecto.

Un agradecimiento a mis amigos Dennys Pérez, Joel Calle, Byron Alvarado, Gabriel Velásquez, y Erick Pinargote, con quienes he compartido los buenos y malos momentos que he pasado durante estos años de mi carrera universitaria, así como reuniones de estudios y celebración por la culminación exitosa de otro semestre más.

**Daniel Robinson Gusnay Carrillo**

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; ADRIÁN VIDAL BAZURTO ONOFRE y DANIEL ROBINSON GUSNAY CARRILLO damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Adrián Vidal Bazurto  
Onofre



---

Daniel Robinson Gusnay  
Carrillo

# EVALUADORES



**PhD. Wilton Edixon Agila Gálvez**

PROFESOR DE LA MATERIA



Firmado electrónicamente por:  
**VICTOR MANUEL  
ASANZA ARMIJOS**

**MSc. Víctor Manuel Asanza A.**

PROFESOR TUTOR

## RESUMEN

El presente proyecto de titulación consiste en el diseño de un prototipo electrónico para monitorear y controlar el factor de potencia de la energía que suministra un parque solar empleando un algoritmo de machine learning, el mismo es un equipo compacto y eficiente que brinda el monitoreo en tiempo real del sensado de señales de potencias, fp, voltajes y corrientes del parque solar durante su operación, y controlar los bancos de capacitores para la corrección del fp de manera manual o automática con el modelo entrenado a partir de la ejecución del algoritmo ML desarrollado.

Previo al diseño, se realizó una consulta de las regulaciones en que se rigen los parques solares para la generación eléctrica, donde el prototipo fue adaptado para que el control automático del fp sea en el rango indicado en esas normas. Se divide en dos equipos: un módulo de monitoreo donde se empleó un sensor de energía por fase, 3 relés, un ESP-32 que envía los datos medidos y ejecuta el control de los relés por histéresis o control predictivo; y un segundo equipo para el entrenamiento del algoritmo que consta de un ESP-32 que recibe los datos sensados del módulo de monitoreo y que son enviados a un raspberry pi 4 que ejecuta el entrenamiento del algoritmo. Para la comunicación de ambos equipos se usó el protocolo ESP-NOW por su capacidad de comunicación entre equipos para el envío de datos sin estar conectados al WiFi.

Mediante el modelo recurrente escogido para el algoritmo del prototipo, se obtuvo un valor RMSE de 0.0000022 muy pequeño a comparación de los otros 3 modelos mencionados en la metodología, lo que permitió con un tiempo aceptable de entrenamiento en la raspberry hacer uso de este modelo para la predicción de factor de potencia. Finalmente, se constató que mediante un control predictivo con histéresis podemos realizar el control de factor de potencia de la energía suministrada sin que se dé cambios bruscos al momento que se dé la mejora de factor de potencia con la activación de los bancos de capacitores necesario para este mejoramiento.

**Palabras Clave:** Comunicación, factor de potencia, algoritmo, entrenamiento, predicción.



## **ABSTRACT**

*The following undergrad, consists of the design of an electronic prototype to monitor and control the power factor of the energy supplied by a solar park using a machine learning algorithm, it is a compact and efficient equipment that provides monitoring in time of the sensing of signals of powers, pf, voltages and currents of the solar park during its operation, and control the capacitor banks for the pf correction manually or automatically with the model trained from the execution of the developed algorithm ML.*

*Prior to the design, a consultation of the regulations governing solar parks for electricity generation was carried out, where the prototype was adapted so that the automatic control of the pf is within the range indicated in those regulations. It is divided into two pieces of equipment: a monitoring module where an energy sensor per phase was used, 3 relays, an ESP32 that sends the measured data and executes the control of the relays by hysteresis or predictive control; and a second device for training the algorithm that consists of an ESP32 that receives the data sensed from the monitoring module and that is sent to a raspberry pi 4 that executes the training of the algorithm. For the communication of both devices, the ESP-NOW protocol was used due to its communication capacity between teams for sending data without being connected to WiFi.*

*Through the recursive model chosen for the prototype algorithm, a very small RMSE value of 0.0000022 was obtained compared to the other 3 models mentioned in the methodology, which allowed, with an acceptable time of training in the raspberry, to make use of this model to power factor prediction. Finally, it was found that by means of a predictive control with hysteresis we can control the power factor of the energy supplied without sudden changes at the time the power factor improves with the activation of the capacitor banks necessary for this improvement.*

**Keywords:** *Communication, power factor, algorithm, training, prediction.*

# ÍNDICE GENERAL

EVALUADORES.....	7
RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS .....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS .....	XI
CAPÍTULO 1 .....	1
<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1 Descripción del problema .....	3
1.2 Justificación del problema.....	4
1.3 Objetivos.....	5
1.3.1 Objetivo General .....	5
1.3.2 Objetivos Específicos .....	5
1.4 Marco teórico .....	5
1.4.1 Energías Renovables .....	5
1.4.2 Parque Solar .....	8
1.4.3 Internet de las cosas (IoT).....	10
1.4.4 Algoritmos de Aprendizaje Automático.....	12
1.4.5 Redes neuronales .....	16
1.4.6 Protocolo ESP-NOW .....	17
1.4.7 Factor de Potencia .....	21
1.4.8 Raspberry Pi 4.....	23
1.4.9 ESP32 .....	24
CAPÍTULO 2.....	26

2	Metodología.....	26
2.1	Análisis de la empresa EnergyFutur Energías Renovables Ecuador S.A. ....	27
2.2	Central Brineforcorp-Cabal Energy.....	28
2.2.1	Módulo fotovoltaico modelo YL250p-29b .....	29
2.2.2	Inversor Freesun HESR-1001 .....	30
2.3	Variables que se miden en la salida del inversor.....	31
2.4	Selección de los sensores y dispositivos para el monitoreo. ....	32
2.4.1	Sensor de energía Voltaje de Corriente AC PZEM004 .....	32
2.4.2	PANTALLA NEXTION .....	34
2.4.3	Diseño del prototipo a partir de los dispositivos seleccionados.....	35
2.4.4	Diseño físico de la solución propuesta.....	36
2.5	Lenguaje de programación .....	37
2.5.1	Microcontrolador (ESP32): Lenguaje Arduino .....	37
2.5.2	Raspberry Pi: Python 3 .....	37
2.6	Protocolo de Comunicación.....	39
2.6.1	ESP-NOW .....	39
2.6.2	MQTT .....	40
2.6.3	ZigBee.....	41
2.6.4	Comparación de los protocolos descritos.....	42
2.7	Datos de entrenamiento.....	42
2.8	Proceso de la creación del modelo de predicción del prototipo. ....	44
2.8.1	Preparación entorno de trabajo .....	44
2.8.2	Procesamiento datos de entrenamiento.....	45
2.8.3	Propuestas de modelos para la predicción de factor de potencia .....	47
2.8.4	Pruebas y Selección Modelo.....	52
2.9	Proceso medición y envió datos .....	53

2.9.1	Diagrama general de bloques de la solución propuesta.....	53
2.9.2	Diagramas de bloques por etapas específicas de la solución propuesta. 54	
CAPÍTULO 3.....		59
3	Resultados Y ANÁLISIS .....	59
3.1	Etapas del entrenamiento del modelo para la predicción de fp .....	59
3.1.1	Pruebas de entrenamiento del modelo.....	59
3.1.2	Calibración parámetros del modelo.....	62
3.1.3	Proceso de validación del modelo.....	62
3.2	Etapas del entrenamiento del modelo en servidor principal Raspberry PI 4...	63
3.3	Transmisión datos medidor y servidor principal .....	64
3.4	Etapas de Monitoreo y control del factor de potencia .....	65
3.4.1	Pruebas de monitoreo .....	65
3.4.2	Interfaz gráfica medidor.....	69
3.4.3	Envío de datos por comunicación inalámbrica con ESP-NOW .....	70
3.4.4	Simulación prototipo etapa mejoramiento factor potencia .....	71
3.4.5	Alcances y Restricciones .....	72
3.5	Trabajos Futuros.....	73
CAPÍTULO 4.....		74
4	Conclusiones Y Recomendaciones .....	74
4.1	Conclusiones .....	74
4.2	Recomendaciones .....	75
ANEXOS.....		76
ETAPA ADQUISICION DATOS .....		76
ETAPA eleccion modelo red neuronal a utilizar .....		78
ETAPA PREDICION Y MEJORAMIENTO FACTOR POTENCIA.....		80
BIBLIOGRAFÍA.....		83

## ABREVIATURAS

A	Amperio
AC	Corriente Alterna
CENACE	Centro Nacional de Control de Energía
E.E.	Empresa Eléctrica
ESPOL	Escuela Superior Politécnica del Litoral
fp	Factor de Potencia
GPIO	General Purpose Input Output
GWh	Gigavatio-hora
IoT	Internet of Things
kW	Kilowatt
kVA	Kilo Voltio Amperio
kVAr	Kilo Voltio Amperio reactivo
MHz	Megahercios
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
ms	milisegundos
MW	Megawatts
M2M	Machine-to-Machine
ODS	Objetivo de Desarrollo Sostenible
ohm	ohmiaje
SNI	Sistema Nacional Interconectado
SoC	System On a Chip
UART	Transmisor Receptor Asíncrono Universal
V	Voltio
Vrms	Voltaje eficaz
W	Watts
°C	Grados Centígrados

## ÍNDICE DE FIGURAS

Figura 1.1 Estadística de producción de energía bruta en Ecuador. ARCONEL, 2018 [5].....	7
Figura 1.2 Mapa de producción de energía por empresas generadoras. ARCONEL, 2018 [5].....	8
Figura 1.3 Esquema de Parque Solar [16].....	9
Figura 1.4 Ejemplo de radiación solar diaria. Elaboración propia. ....	9
Figura 1.5 Arquitectura IoT de 7 niveles propuesto por CISCO [20]. ....	11
Figura 1.6 Arquitectura IIoT de 3 niveles [21]. ....	12
Figura 1.7 Algoritmos más usados en el Aprendizaje Automático [23]. ....	13
Figura 1.8 Diagrama de flujo en el Aprendizaje Supervisado [22]. ....	14
Figura 1.9 Aprendizaje no supervisado [23].....	15
Figura 1.10 Ejemplo de una red neuronal [28].....	17
Figura 1.11 Esquema Comunicación Unidireccional simple con ESP-NOW [31].....	19
Figura 1.12 Comunicación unidireccional un Maestro-varios Esclavos (izquierda) y un Esclavo-varios Maestros (derecha) [30].....	20
Figura 1.13 Comunicación bidireccional de múltiples placas ESP32 [30].....	20
Figura 1.14 Triangulo de Potencia (predominante inductivo).....	21
Figura 1.15 Raspberry Pi 4 model B [33]. ....	23
Figura 1.15 ESP32-DevKitC-32D [35].....	25
Figura 1.16 Pin Layout del ESP32-DevKitC-32D [35]. ....	25
Figura 2.1 Diagrama de bloques del modelo a seguir.....	26
Figura 2.2 Vista de la Planta fotovoltaica Brineforcorp [41]. ....	28
Figura 2.3 panel fotovoltaico Yingli YL250P-29b [42]. ....	30
Figura 2.4 E.T. prefabricado de hormigón donde se instalaron los inversores [41]. ..	31

Figura 2.5 Arquitectura del sensor de energía Voltaje de Corriente multímetro AC PZEM 004 [44].	32
Figura 2.6 PZEM-004t módulo de energía Voltaje de Corriente multímetro AC 80-260V 100A con transformador de corriente [47].	33
Figura 2.7 Nextion NX3224T028_011 [50].	34
Figura 2.8 Ejemplo de interfaz HMI en la pantalla Nextion NX3224T028_011 [50].	35
Figura 2.9 Esquema en diagrama de flujo de los elementos que conforma el prototipo electrónico.	36
Figura 2.10 Diseño físico del Módulo de monitoreo en tiempo real.	36
Figura 2.11 Diseño físico del Módulo del servidor principal y para el entrenamiento del algoritmo.	37
Figura 2.13 Python en Raspberry pi [53].	38
Figura 2.14 Logo de TensorFlow [53].	38
Figura 2.15 Comunicación Bidireccional entre dispositivos ESP32 [30]	39
Figura 2.16 Ejemplificación del Protocolo MQTT [56].	41
Figura 2.17 Esquema del sistema de planta solar empleado en Simulink para adquisición de datos.	43
Figura 2.18 Valor máximo de Potencia que pueda dar la planta solar simulada.	43
Figura 2.19 Logo de Google Colab [54].	44
Figura 2.20 Entorno de Google Colab [54].	44
Figura 2.21 Instalación de librería de TensorFlow y otras específicas.	45
Figura 2.22 Archivo CSV de los datos simulados en Matlab.	45
Figura 2.23 Gráfica del fp trifásico de un día al azar en el sistema de planta solar realizado en Simulink.	46
Figura 2.24 Propuesta de Modelo 1.	47
Figura 2.25 Capas de entrada y salida en el Modelo 1 propuesto.	47
Figura 2.26 CSV de los datos obtenidos a partir del Modelo 1.	48
Figura 2.27 Propuesta de Modelo 2.	48

Figura 2.28 Capas de entrada y salida en el Modelo 2 propuesto .....	49
Figura 2.29 CSV de los datos obtenidos a partir del Modelo 2. ....	49
Figura 2.30 Propuesta de Modelo 3.....	50
Figura 2.31 Capas de entrada y salida en el Modelo 3 propuesto .....	50
Figura 2.32 Propuesta de Modelo 4.....	51
Figura 2.33 Capas de entrada y salida en el Modelo 4 propuesto .....	51
Figura 2.34 CSV de los datos obtenidos a partir del Modelo 4. ....	52
Figura 2.35 Solución general del sistema en diagrama de flujo.....	54
Figura 2.36 Diagrama de flujo de las tareas del ESP32 del módulo de Monitoreo en tiempo real. ....	55
Figura 2.37 Diagrama de flujo de las tareas del Raspberry del servidor principal. ....	56
Figura 2.38 Diagrama de flujo del proceso del Web Server.....	57
Figura 2.39 Diagrama de flujo de la Tarea Relé definida.....	57
Figura 2.40 Diagrama de flujo de la Tarea Histéresis definida .....	58
Figura 2.41 Diagrama de flujo de la Tarea Control Predictivo definida. ....	58
Figura 3.1 Predicción del Fp trifásico con cada modelo propuesto.....	59
Figura 3.2 Valores RMSE de cada modelo propuesto. ....	60
Figura 3.3 Tiempo de entrenamiento de cada modelo propuesto.....	60
Figura 3.4 Error cuadrático medio.....	62
Figura 3.5 Máxima diferencia de factor de potencia. ....	63
Figura 3.6 Error cuadrático medio vs cantidad de muestras previas. ....	63
Figura 3.7 Entrenamiento del modelo en la Raspberry a partir de una muestra pequeña.....	64
Figura 3.8 Datos recibidos del ESP32 comunicado con ESPNOW al módulo de monitoreo.....	64
Figura 3.9 Envío de Predicciones de factor de potencia realizado servidor principal (Raspberry).....	65



Figura 3.10 Interfaz web para el monitoreo de potencias, factor de potencia, voltajes y corrientes. ....	65
Figura 3.11 Interfaz web monitoreando el FP trifásico. ....	66
Figura 3.12 Interfaz web para el control manual o automático de los relés, y para configurar el Setpoint del fp. ....	66
Figura 3.13 Interfaz web para la configuración del tiempo de muestreo y dirección MAC del dispositivo a enviar los datos. ....	67
Figura 3.14 Visualización de la Interfaz web mediante un móvil. ....	67
Figura 3.15 Visualización de los datos en el módulo de monitoreo y en un dispositivo móvil. ....	68
Figura 3.16 Web server del servidor principal para el monitoreo de potencia, corriente, voltaje y factor de potencia por fase. ....	68
Figura 3.17 Monitoreo de la potencia activa instantánea de cada fase. ....	69
Figura 3.18 Monitoreo de la potencia aparente instantánea de cada fase. ....	69
Figura 3.19 Monitoreo del factor de potencia instantáneo de cada fase. ....	70
Figura 3.20 Envío de datos por ESP-NOW. ....	70
Figura 3.21 Control manual del factor de potencia de la planta. ....	71
Figura 3.22 Control automático del factor de potencia de la planta. ....	71

## ÍNDICE DE TABLAS

Tabla 2.1 Características del módulo fotovoltaico modelo YL250p-29b [43]. .....	29
Tabla 2.2 Características del sensor de energía VoltajeCorriente AC PZEM004 [44].....	33
Tabla 2.3 Comparación entre los protocolos de comunicación descritos [30,54,57]..	42
Tabla 2.4 Procesamiento de los datos del CSV de los datos simulados. ....	46
Tabla 3.1 Parámetros importantes obtenidos en cada Modelo propuesto. ....	61

# CAPÍTULO 1

## 1. INTRODUCCIÓN

Las fuentes de energías renovables no convencionales (FERNC) tienen como misión proveer energía limpia y segura donde no se produce el gasto o agotamiento de su fuente generadora como la energía solar, energía eólica, la fuerza motriz del agua, entre otras, siendo una alternativa a las energías no renovables o convencionales que se basan en fuentes de energía de recurso limitado como el petróleo, gas natural, carbón, y energía nuclear, entre otros [1].

El aumento de la producción de bienes y crecimiento poblacional conlleva al consumo energético convencional global aumente en los últimos siglos, esto genera el incremento desmesurado de emisión de gases de efecto invernadero lo que acarrea a consecuencias negativas como el deshielo glaciar. Este problema denominado Cambio climático afecta al entorno actual y de las generaciones venideras, por ello, las energías renovables es una alternativa para reducir la emisión de contaminantes y ofrecer un mejor lugar de vida en el planeta [2].

La Ley 1715 establecida en el 2014 en el protocolo de Kioto y la Convención de la Naciones Unidas (ONU), establece para el contexto del Cambio Climático en iniciar y favorecer el desarrollo de nuevas tecnologías ambientalmente sostenibles para mejorar el cambio climático. Esta ley pretende que se reglamente de manera técnica la implementación de FERNC en países donde comúnmente no son usadas, explícitamente a sus municipios y localidades donde existe ausencia de abastecimiento eléctrico por no estar interconectadas al suministro eléctrico de energía del país. Otros objetivos con esta Ley, es la reducción de gases de efecto invernadero, seguridad en el abastecimiento, eficiencia y cambio en el consumo energético de la población a la energía renovable no convencional [3].

En la actualidad, en el Ecuador dado a los códigos y estándares que se han impuesto en las leyes para generación y distribución eléctrica se posee de varias fuentes generadoras de energía convencional y no convencionales [4], siendo en el 2018 la mayor la energía hidráulica con el 70.71% de producción energía bruta en todo el país, esta energía es considera energía renovable en nuestro país [5].

En nuestro país, se han planteado retos para la introducción de FERNC como los Proyectos Villonaco II y III para energía eólica con capacidad total de 22 MW y el Proyecto Aromo para energía solar fotovoltaica con capacidad aproximada de 200 MW. Según dijo Diego Jijón, Analista Técnico de Servicios Especializados, en una entrevista: “estos proyectos establecen nuevas oportunidades y retos en estas energías para el beneficio del país”, y señala que “los factores de rendimiento fotovoltaico oscilan entre un 15 y 20 % y el proyecto El Aromo sería cerca del 17%, donde si se multiplica las horas de sol al año por el factor de rendimiento, se tendría una idea de la generación eléctrica anual en este proyecto” [6].

Actualmente, existen 60 regulaciones vigentes, siendo las más importantes la ARCONEL 004/15 y CONELEC 006/00 para los requerimientos técnicos en la operación y control para la generación eléctrica [4]. Las centrales fotovoltaicas deben cumplir las regulaciones establecidas en el país para la generación y distribución de energías renovables como la Resolución No. ARCONEL-071/15 sobre los requerimientos técnicos de los generadores renovables no convencionales, y la Regulación No. ARCONEL 004/15 para los generadores de 100kW o mayor potencia nominal que se conectaran o están conectados a media o alta tensión, en esta se define los criterios y requisitos técnicos que deben seguir las empresas distribuidoras, CENACE, actores y participantes que operan proyectos de FERNC, todas estas regulaciones definen que se debe asegurar en todo tiempo los mismos niveles de voltajes y corrientes del sistema dentro de los rangos admisibles, la eficiencia y seguridad en su operación [7-9].

Teniendo en claro que estas empresas en la generación y distribución de energía deben cumplir regulaciones para la interconexión a un sistema de transmisión. Como el control de potencia activa, reactiva, voltaje y frecuencia, límites de desviación de voltaje y frecuencia [4]. Esto conlleva a que las empresas generadoras y distribuidoras de este tipo de fuente busquen mayor control en la calidad y eficiencia de energía que puedan generar. Para mejorar esto, se propone diseñar un prototipo de control y monitoreo del factor de potencia mediante algoritmo de Autoaprendizaje, con el fin de prevenir que el factor de potencia este fuera de los valores establecido en las regulaciones para la distribución de energía.

## 1.1 Descripción del problema

En la actualidad, existen industrias dedicadas a la venta de energía a los proveedores nacionales como CELEC EP, CNEL EP y sociedades anónimas como la E.E. Quito, E.E. de Ambato, etc. Estas industrias deben cumplir ciertos estándares como: mantener un factor de potencia y estabilidad de suministro [10]. Según el informe del ARCONEL 2018, detalla que las centrales fotovoltaicas de generación eléctrica poseen una potencia nominal y efectiva instalada de 27.63 MW y 26.74 MW, respectivamente. Donde de las 49 empresas generadoras por fuentes renovables que participaron en el 2018, de las centrales fotovoltaicas se registró que el 34.77 GWh fue de producción e importaciones SNI por plantas generadoras como Brineforcorp, Epfotovoltaica, Gran Solar, SurEnergy, Salinas y Santa Cruz Solar; y un 3.31 GWh por plantas distribuidoras como la central Isabela Solar de la E.E. Galápagos y la central Panel Fotovoltaico de la E.E. Ambato [5].

El incumplimiento de una regulación puede derivar en penalizaciones legales y económicas a este tipo de industria. De hecho, los grandes desafíos para esta industria es mantener un monitoreo constante y mejora del factor de potencia por medio de bancos de capacitores que por lo general lo realizan de manera manual, con el fin de cumplir lo que se detalla en las regulaciones, que la capacidad de control del factor de potencia sea desde 0.95 en atraso (predominante inductivo) y hasta 0.95 en adelanto (predominante capacitivo) [7-8]. Esta práctica provoca la necesidad de tener un operario vigilando constantemente dicho parámetro, cuyo tiempo de respuesta no siempre es el adecuado. Los parques solares es una alternativa de obtención de energía eléctrica que poseen 100 kW o más como capacidad nominal, en varias de este tipo de instalaciones durante el inicio de su operación y al llegar la noche presenta caídas considerables del factor de potencia, pudiendo llegar a valores muy por debajo de los 0.90, lo que afecta la calidad de energía que suministran, esto puede deberse por cambios parciales de la radiación solar que captan los paneles durante el transcurso del día. Se estima que la radiación solar se aprovecha entre las 6am a 6pm, sin embargo, entre las 5-10pm son las horas donde se da el mayor aporte energético por el parque solar, esto conlleva a una desventaja al no poder generar potencia activa durante esas horas pico de operación. [11].

Por ello, últimamente en el país estas industrias comienzan a enfocarse en proyectos para automatizar sus instalaciones, con el fin de mejorar la eficiencia de la generación de energía en sus plantas, siendo así de gran importancia el desarrollo de un sistema de monitoreo y control automatizado que permita a estas empresas generadoras incrementar el factor de potencia y la estabilidad del suministro.

## **1.2 Justificación del problema**

El presente prototipo de medición de energía propuesto para ser instalado en un sistema de generación de energía fotovoltaica es no invasivo, es decir, no se requiere realizar cambios de infraestructura del Parque Solar.

El diseño del prototipo electrónico integra sistemas embebidos que permiten controlar y monitorear el factor de potencia en tiempo real de un sistema eléctrico, con un tiempo de actuación rápida en el cambio de carga capacitiva adecuado para el mejoramiento de  $fp$  y ajustable en el tiempo para el monitoreo y la adquisición de datos de las variables eléctricas medidas, por ello no requerirá de una supervisión constante de un operario, pues el proceso del prototipo propuesto se encargará de manera automatizada la corrección de factor de potencia requerida en el sistema.

Por ello, el prototipo mediante la utilización de algoritmos machine learning permitirá que el mejoramiento del factor de potencia sea progresivo de tal manera que la energía que suministra el sistema fotovoltaico a las cargas conectadas a él nunca tenga el factor de potencia por debajo de los valores permitidos para su distribución eléctrica de acuerdo con las regulaciones por ARCONEL y CONELEC para FERNC [7-9]. Además, al ser un aprendizaje automático en el borde, no se requiere de conexión a internet para su continuo funcionamiento.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Implementar un prototipo electrónico para el monitoreo y corrección de factor de potencia en tiempo real, mediante algoritmos de predicción.

#### **1.3.2 Objetivos Específicos**

1. Diseñar el hardware del prototipo mediante componentes de sistemas embebidos de mayor alcance.
2. Diseñar el interfaz web para el monitoreo del suministro de energía en tiempo real.
3. Recolectar datos de una planta solar real para el control y monitoreo de las variables eléctricas del sistema.
4. Implementar algoritmos de aprendizaje automático en el borde para corrección de factor de potencia.

### **1.4 Marco teórico**

En este apartado se presenta la investigación realizada con los temas relacionados al proyecto propuesto.

#### **1.4.1 Energías Renovables**

De acuerdo con el informe (20124) del Banco Mundial, la energía debe ser un acceso universal a un nivel alcanzable, confiable y sostenible, siendo uno de los objetivos de la ODS, como eje para poner fin a la pobreza y hacer frente al cambio climático actual en el mundo. Hoy en día, alrededor de 1000 millones de personas en el mundo que todavía no disponen de este recurso y cientos de millones cuentan con un suministro eléctrico limitado y de poca confiabilidad. Se destaca también que alrededor de tres mil millones de personas emplean combustibles contaminantes tales como leña u otra biomasa para la calefacción o cocina en sus hogares, que causan impactos en la salud de la población pues los contaminantes se propagan en el aire de los espacios abiertos y cerrados de la zona [12].

La energía renovable o no convencional presenta diferentes tipos, según los recursos naturales que se usan para la producción de energía, de las cuáles se puede mencionar a la energía solar, solar fotovoltaica, eólica, mareomotriz, biomasa (por combustión de desechos vegetales o seres vivos), hidráulica, solar térmica, térmico biogás (proviene de residuos orgánicos), etc. Estas diversas fuentes sostenibles presentan inconvenientes tales como: dependencia geográfica y meteorológica, alto coste para su instalación como el caso de la hidráulica que se requiere construir grandes presas para una considerable producción de energía eléctrica, y sostenibilidad dado que algunas no son del todo aptas como es la geotérmica que en su operación puede desprender gases tóxicos que se encuentran dentro de la corteza terrestre que llega a afectar la capa de ozono del planeta. A pesar de los inconvenientes que pueden presentar, muchas de ellas cuentan de mayor ventaja al ser fuentes de energía limpia, segura, inagotable y con competitividad creciente en el entorno [2].

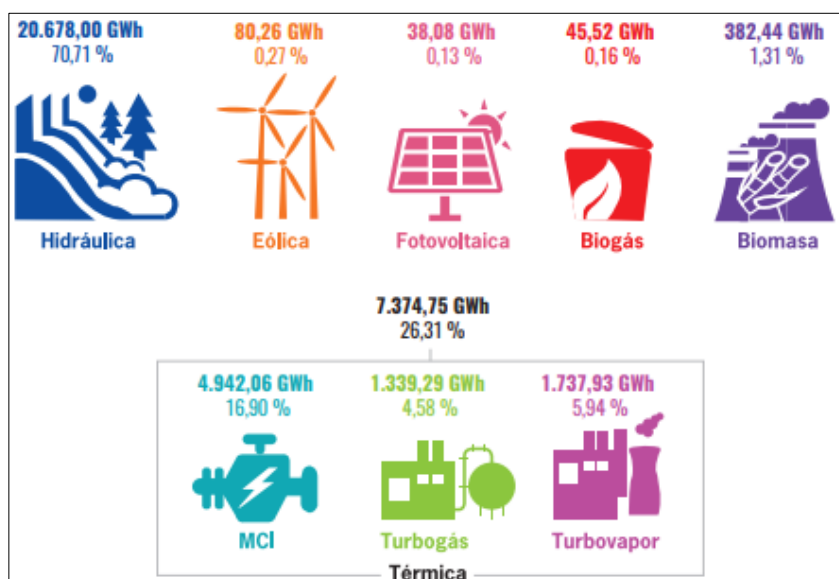
A nivel mundial, la energía renovable empieza a tomar más importancia, con el fin de que los países empiecen a usar sistemas energéticos modernos y con menor impacto ambiental. Muchos países con la implantación de políticas ambiciosas de ayuda a las energías renovables permiten que cada vez los costos de los sistemas que conforman a estas evolucionan a la baja de manera sostenida, lo que hace posible que en regiones con poco acceso a la energía puedan obtener una fuente renovable como los sistemas de energía solar domiciliario que cada vez son más eficientes y de más bajo costo [12].

La Agencia Internacional de la Energía (AIE), pronostica que las energías renovables irán aumentando su participación en el suministro eléctrico mundial, donde pasará del 26% en 2018 al 44% en 2040, donde las principales serán las energías eólica y fotovoltaica como mayor presencia en la generación de energía, de manera que así proporcione 2/3 del incremento de demanda energética registrado en ese período [13].

En territorio nacional, está presente la Energía Hidráulica, Energía Térmica Biomasa, Energía Eólica, Energía Térmica Biogás y Energía Fotovoltaica, siendo los retos al cambio de matriz energética hecha realidad, el cual se encamina a



alcanzar el 93% de energía limpia y renovable en el país mediante el aprovechamiento de estos recursos naturales. De acuerdo con la última estadística reportada por el ARCONEL del 2018, la energía total bruta generada fue de 29.243,59 GWh, siendo el 72,58% la generación por energía renovables, lo cual representa 21.224,31 GWh que se genera de energía eléctrica para ser distribuido en beneficio de la sociedad ecuatoriana (Ver Figura 1.1).



**Figura 1.1 Estadística de producción de energía bruta en Ecuador. ARCONEL, 2018 [5].**

Como se puede visualizar hay un cambio muy notable de generación de energía eléctrica a comparación de las estadísticas del ARCONEL del 2015 en donde el 51,78% era producción de energía renovable [14].

En la Figura 1.2, se presenta un mapa de la producción de energía de plantas generadoras según datos del Arconel 2018. En esta cartografía se destaca que la central hidroeléctrica que mayor producción en energía bruta fue la Coca Codo Sinclair con 6.200,71 GWh; de las centrales fotovoltaicas la de mayor producción de energía bruta fue la central Salinas de la empresa Granja Solar con 3,88 GWh; de las centrales de biogás la de mayor producción de energía bruta fue la central El Inga de la empresa Gasgreen con 40,25 GWh, y de central eólica está la Villanoco de la empresa CELEC-Gensur que produjo 73,70 GWh [5].

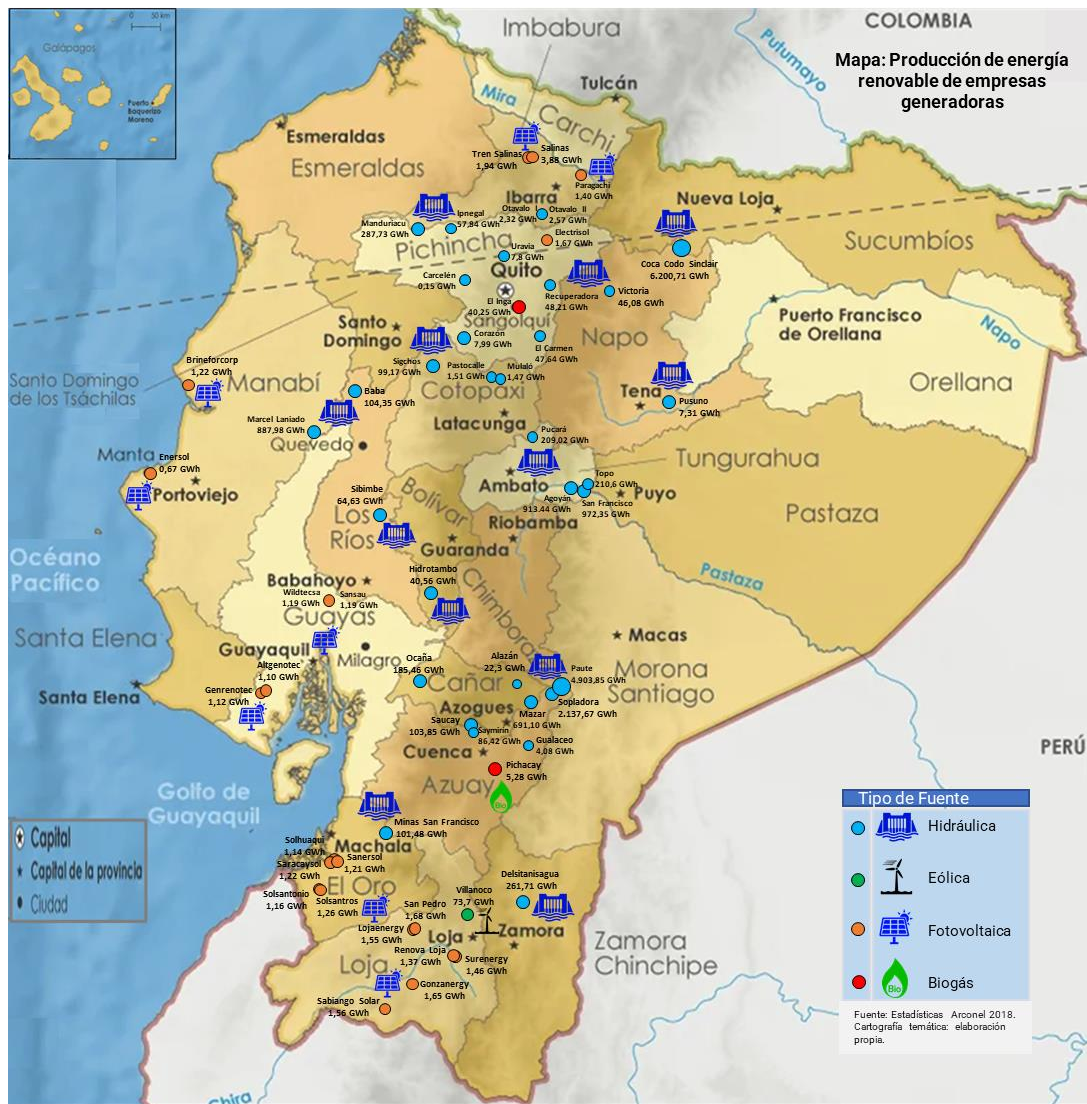


Figura 1.2 Mapa de producción de energía por empresas generadoras. ARCONEL, 2018 [5].

### 1.4.2 Parque Solar

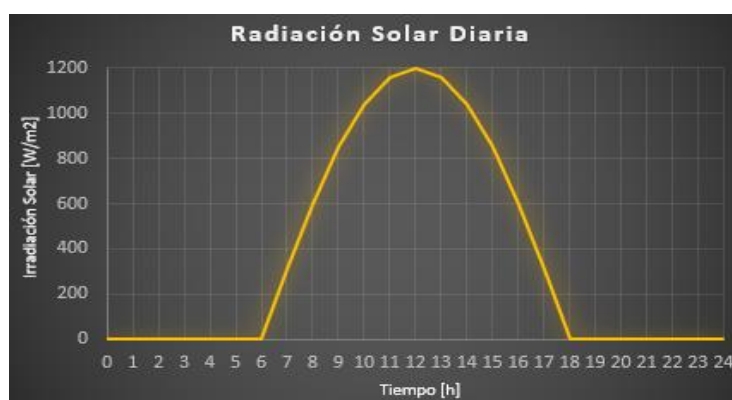
Un parque solar, es una alternativa de obtención de energía eléctrica limpia para la reducción de las emisiones de CO<sub>2</sub>, siendo una instalación de energía solar a gran escala, donde se establece múltiples paneles solares interconectadas que captan la energía solar para transformarla en energía eléctrica. Este tipo de instalaciones se emplean para generar miles de vatios de electricidad por segundo, de manera de abastecer a la demanda energética de industrias, universidades, ciudades, así como su venta de energía para redes de distribución eléctrica. En la Figura 1.3 se presenta el esquema de una planta solar, está requiere de una sala de control centralizada, transformadores de alta tensión e

inversores, este último debido a que los paneles solares transforman la energía solar en corriente directa, por lo que es necesario el inversor para que sea alterna y así poder realizar su distribución. Además, es importante indicar que también el parque solar puede almacenar en campos de baterías la energía solar captada, que luego puede ser usado en cualquier momento [15].



**Figura 1.3 Esquema de Parque Solar [16].**

Sin embargo, como descrito en el problema, los parques solares presentan caídas considerables en el factor de potencia entre las 5 de la tarde a 10 de la noche, esto se debe a que en esas horas ya la demanda del aporte energético incrementa debido a que la gente ya regresa del trabajo a su hogar, conllevando a actividades cotidianas que genera más demanda de energía reactiva a la planta solar, la que no puede generar potencia activa por estar en las horas donde la radiación solar es nula para que los paneles lleguen a generar energía útil.



**Figura 1.4 Ejemplo de radiación solar diaria. Elaboración propia.**

En nuestro país, los parques solares, generaron 37.62 GWh en el 2015, 38.08 GWh en el 2018 y 36.06 GWh en el 2019, de energía bruta en los cuáles se

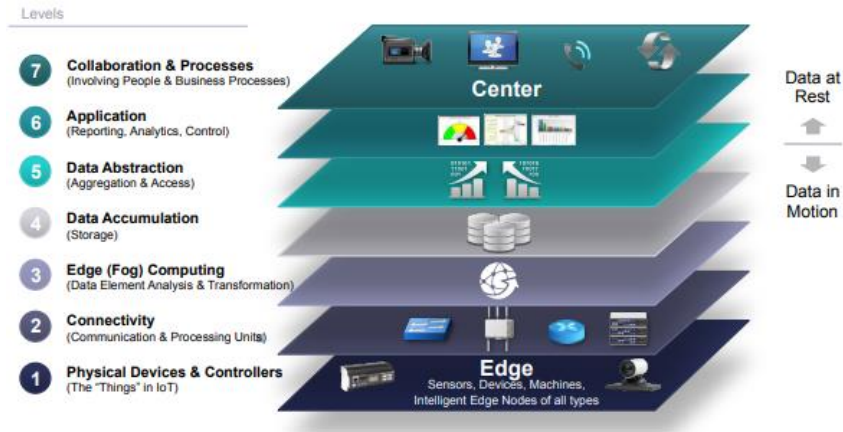
denota como hay una reducción en la producción de energía en estos parques desde el 2015-2019 [5,14,17], siendo importante mencionarlo debido a que esto puede deberse a factores como la radiación solar promedio anual, deficiencia en la producción de energía en las horas con poca o nula irradiancia solar, temperatura de operación en los paneles, entre otros.

### **1.4.3 Internet de las cosas (IoT)**

La tecnología IoT, es un término introducido por primera vez por el londinense Kevin Ashton (denominado como “el padre del internet de las cosas”) que describe mediante la identificación por radiofrecuencia y otros sensores a los objetos cotidianos se puede crear el IoT. Es un concepto que ha ido evolucionando y está en auge en los últimos años, pues permite la agrupación e interconexión de dispositivos, objetos y sistemas mediante el uso de una red (sea de carácter privado o Internet), dónde todos ellos podrían ser visibles e interaccionar [18]. Respecto al tipo de objetos o dispositivos podrían ser cualquiera, desde sensores y dispositivos mecánicos hasta objetos cotidianos como pueden ser el frigorífico, el calzado o la ropa. Cualquier cosa que se pueda imaginar podría ser conectada a internet e interaccionar sin necesidad de la intervención humana, el objetivo por tanto es una interacción de máquina a máquina, o lo que se conoce como una interacción M2M (machine to machine) o dispositivos M2M.

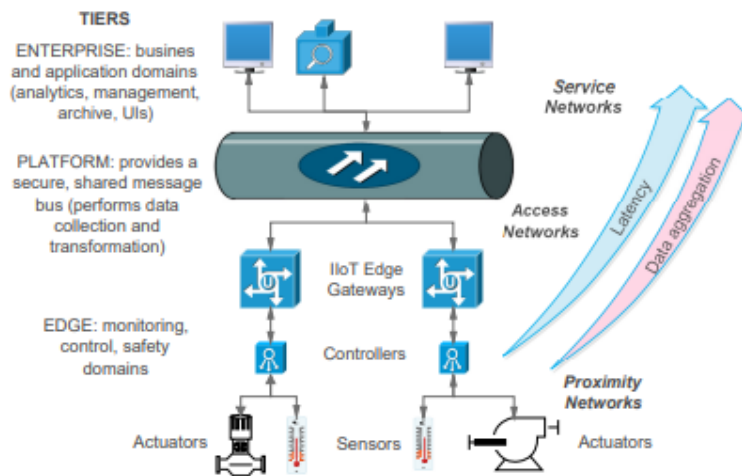
Los ejecutivos ponen al IoT como una tecnología emergente importante por encima de la robótica y la IA (inteligencia artificial). Según Daniel Burrus (director ejecutivo de Burrus Research), considera que “el IoT es una de las tendencias tecnológicas que causará la mayor interrupción y ofrecerá la mayor cantidad de oportunidades en la industria” [19]. Además, que se trata de cambiar los modelos antiguos, para crear nuevos productos y servicios, donde el IoT interrumpirá con el mismo impacto en todas las industrias imaginables (agricultura, seguridad, atención médica, energía, etc). Por ejemplo, en la industria petrolera al implementar tecnologías móviles de sensor a máquina para combatir los accidentes con mucha anticipación mediante un rápido análisis y acción inmediata cuando los sensores detectan un problema (como una fuga o tubería corroída). En la Figura 1.5, se ilustra un modelo de referencia de IoT y sus niveles, donde los datos fluyen en ambas direcciones; en un patrón de seguimiento los datos de

información van del nivel 1 (Dispositivos físicos y controladores) al nivel 7 (Colaboración y procesos), y en un patrón de control van del nivel 7 al nivel 1.



**Figura 1.5 Arquitectura IoT de 7 niveles propuesto por CISCO [20].**

El IoT, industrial IoT (IIoT) e industria 4.0, son conceptos muy relacionados, sin embargo, no pueden usarse indistintamente. En el IIoT, la comunicación consiste en enlaces M2M que deben cumplir estrictos requisitos en términos de confiabilidad y puntualidad. En la automatización de procesos industriales para las aplicaciones de control y monitoreo se lo puede agrupar en las subcategorías de control de bucle cerrado, monitoreo/supervisión, y en enclavamiento y control. La subcategoría de monitoreo/supervisión es menos sensible a la pérdida de paquetes y fluctuación, así como tolerante a retardos de transmisión en el segundo nivel, pero la subcategoría de control de bucle cerrado, enclavamiento y control, requieren que el retardo sea limitado a un nivel de 10-100 ms con 99,99% en confiabilidad de transmisión. La comparación de volumen de datos muestra que, mediante el IoT, los datos generados son dependientes en gran magnitud de la aplicación, en cambio a partir del IIoT, los datos se dirigen a la parte analítica, para mantenimiento predictivo y logística mejorada, por lo que se intercambian una gran cantidad de datos. En la Figura 1.6, se presenta gráficamente la arquitectura híbrida del IIoT, en el que se denota un incremento en la agregación de datos y en la latencia en los diferentes niveles [21].

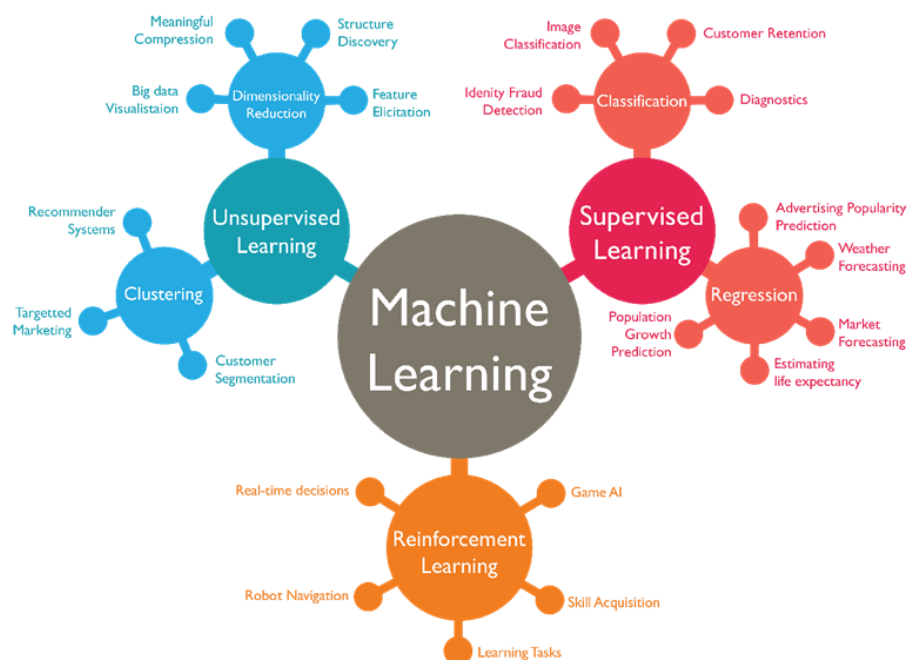


**Figura 1.6** Arquitectura IIoT de 3 niveles [21].

#### 1.4.4 Algoritmos de Aprendizaje Automático

El Aprendizaje Automático o ML por sus siglas en inglés, es un campo de estudio que posibilita que las computadoras tengan la capacidad de aprender sin ser intencionadamente programada que tiene como objetivo principal el aprender de los datos, diseñar y mejorar modelos matemáticos de entrenamiento único o continuo mediante datos relacionados al problema, esto permite que la máquina pueda deducir el futuro y decidir la mejor acción sin un conocimiento completo de todos los elementos que intervienen con el único fin de solucionar y observar los resultados de la mismas . La aplicación del ML es para enseñar a las máquinas como usar los datos de forma óptima y eficiente (mediante un enfoque de aprendizaje estadístico). Muchas veces no podemos interpretar la información de la base de datos que disponemos, por lo que en casos así es donde se aplica el ML en las computadoras. Hoy en día, la abundancia de conjuntos de datos permite que muchas empresas, industrias, universidades, entre otros, aumente la demanda de aplicar el aprendizaje automático en sus sistemas para extraer la información relevante que requieran [22].

El ML se basa en diversos algoritmos para resolver problemas de datos, se emplean dependiendo del tipo de problema a querer dar solución, el número de variables, el tipo de modelo que mejor se adapte a él, entre otros. En la Figura 1.7, se presenta los tipos de algoritmos por lo general usados en el ML.



**Figura 1.7 Algoritmos más usados en el Aprendizaje Automático [23].**

A continuación, se dará un detalle breve de los más utilizados en el ML:

- **Aprendizaje Supervisado:** se basa del concepto profesor o supervisor, que proporciona al ML una medida precisa de su error, comparable con los valores de salida. Es una función aplicada en algoritmos reales que hace uso de un conjunto o etiquetas de datos de entrenamiento conformado por parejas (entrada y salida correcta o esperada), además, también de ser capaz de funcionar con conjunto de datos nunca vistos. Su uso en el ML, cuando el algoritmo es lo más flexible y los datos son coherentes, permite que la precisión incremente y que el valor predicho y esperado sean lo más similar posible (diferencia de cero). El Aprendizaje Supervisado, se divide en tareas de Regresión y Clasificación. La tarea de **regresión** consiste en deducir a partir de los datos de entrenamiento una función basada en valores de salida continuas, en los cuáles sus resultados se intentan predecir mediante asignación de variables de entrada a alguna salida continua, por ejemplo, predecir el precio de un automóvil según el kilometraje y su marca, en este ejemplo, el precio es función del kilometraje y marca del auto, que es la producción continua. La tarea de **clasificación** es cuando en el proceso solo hay un número discreto de posibles resultados que se denomina categoría (por ejemplo, conjunto finito de clases, enumeración, entre otros), esta salida en valor categórico se predice en valores discretos como si o no, 0 o 1,

positivo o negativo, true or false, etc. En otras palabras, para el caso de regresión, las etiquetas de salida son continuas, mientras que son discretas en el caso de clasificación. Las aplicaciones comunes del aprendizaje supervisado son: detección de patrones, procesamiento natural de lenguaje, clasificación automática de imágenes, procesamiento de secuencia automático, etc [24].

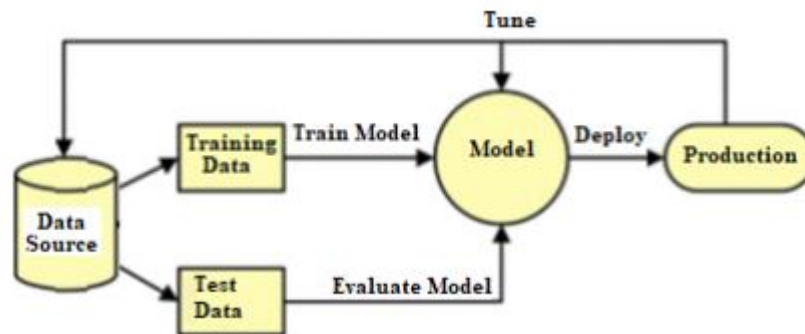
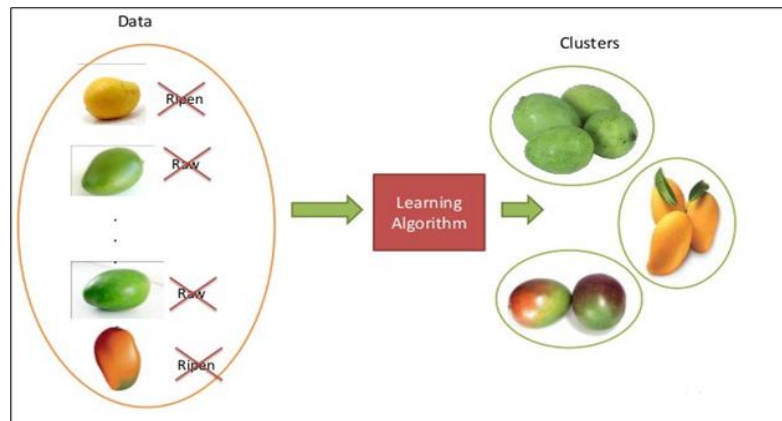


Figura 1.8 Diagrama de flujo en el Aprendizaje Supervisado [22].

➤ **Aprendizaje no supervisado:** se diferencia de la ausencia de cualquier supervisor, donde se proporciona un conjunto de datos en el cual no estamos interesados en ajustar pares de entrada y salida, solo en encontrar las estructuras de los datos disponibles (y datos futuros que se pueda dar por el mismo fenómeno). El Aprendizaje no supervisado, se divide principalmente en agrupación en clústeres y reducción de la dimensionalidad. El algoritmo de agrupación en **clústeres** (según su similitud o medida de distancia), consiste en agrupar los datos por la similitud de los mismos, por ejemplo, la segmentación de personas según las frutas compradas en un comisariato. El algoritmo de agrupación en **reducción de la dimensionalidad** consiste en simplificar la estructura de los datos reteniendo las características fundamentales de los mismos. Las aplicaciones comunes del aprendizaje no supervisado son: etiquetado automático, segmentación de objetos, detección de similitud [24].





**Figura 1.9 Aprendizaje no supervisado [23].**

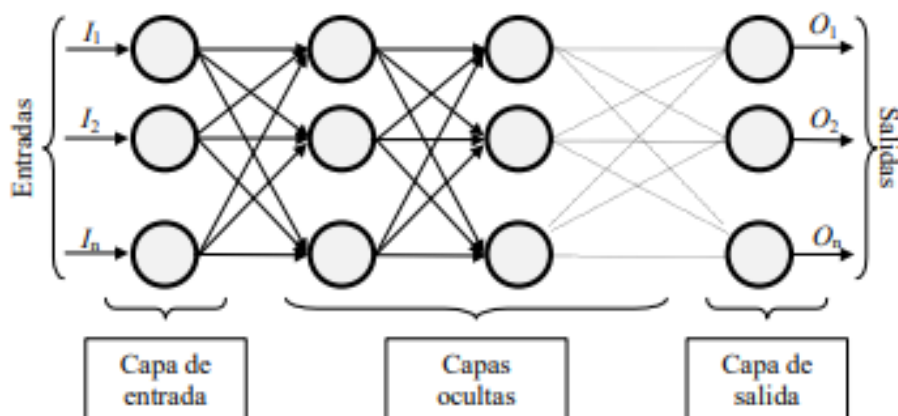
Por último, el algoritmo en el que nos basaremos para cumplir el diseño del prototipo para el control y monitoreo del factor de potencia en una planta fotovoltaica será el Autoaprendizaje o Self-Training, que es un método generalmente empleado en el algoritmo por Aprendizaje Semisupervisado en el ML en diversas aplicaciones como procesamiento natural de lenguaje detección y reconocimiento de objetos, entre otros.

➤ **Self-Training o Autoaprendizaje:** Este método aprovecha el poder del aprendizaje del aprendizaje semi-supervisado que es un algoritmo apto para ser utilizado en problemas de regresión, clasificación y predicción, así como de ser adecuados cuando en los modelos creados aparezcan conjuntos de datos sin etiquetar debido al alto coste del etiquetado o por la poca experiencia del desarrollador o diseñador. El Autoaprendizaje es iterativo que emplea sus propias predicciones para proporcionar etiquetas de salida a la mayoría de las observaciones o datos sin etiquetas en el modelo de algoritmo construido en el ML para una aplicación [25]. El Autoaprendizaje se basa de un clasificador que se entrena con una parte de observaciones etiquetadas (conjunto de entrenamiento), donde posteriormente se alimenta de datos que no poseen etiquetas, en el que realiza para este conjunto de datos su propia asignación de etiquetas en función de su predicción de alta confianza, y así estas etiquetas predichas pasan al conjunto de entrenamiento. Este procedimiento se repite más de ahí su nombre de Autoaprendizaje, pues el clasificador vuelve a entrenar con el nuevo conjunto de datos etiquetados y así va aprendiendo constantemente con cada iteración de procedimiento de entrenamiento hasta alcanzar una condición de parada [26].

### 1.4.5 Redes neuronales

Las redes neuronales son algoritmos de machine learning, que puede verse de manera análoga como el modelado del sistema nervioso de un ser vivo al realizar una tarea específica. Forman la base de la rama de Inteligencia Artificial o Deep Learning y la Data Science, que presenta resultados muy significativos en distintas aplicaciones como el reconocimiento de voz, la predicción del comportamiento de usuarios, en análisis de secuencias, detección y clasificación de objetos entre otras. La red neuronal es el conjunto de miles de unidades de procesamiento interconectadas denominadas neuronas, donde el ajuste de pesos (importancia asignada a la interconexión entre dos neuronas) es la manera más usada en el aprendizaje de estas redes [27].

Una red neuronal se conforma de capas (ver Figura 1.10). La primera capa es la **capa de entrada** en la ingresan los datos de entradas (información proveniente de fuentes externas de la red) que posteriormente pasa por la o las **capas ocultas** (capas internas de la red que no poseen contacto directo al exterior del entorno que dependiendo del modelo a usar puede ser una o varias capas interconectadas diversas formas), y salen hasta la **capa de salida** donde se encuentra las características que se quiere llegar a encontrar. Cada capa está conformada de uno o varias neuronas interconectadas a la siguiente (estas conexiones representan pesos), las cuales poseen un valor de sesgo y función de activación (lineal o no lineales) que devuelve una salida generada a partir de una o varias entradas conectadas a la neurona establecida esa función [28].



### **Figura 1.10 Ejemplo de una red neuronal [28].**

Hay varios tipos de redes neuronales: la red monocapa o perceptrón, redes de base radial (RBF), redes multicapa (MLP), redes convolucionales (CNN), y redes recurrentes (RNN).

Las redes neuronales convolucionales, es un tipo de red particular donde las neuronas corresponden a campos receptivos que de forma muy parecida inspira el funcionamiento de las neuronas de la corteza visual primaria del cerebro: Son generalmente usadas en problemas de visión artificial siendo posible su aplicación en el procesamiento de lenguaje naturales o para la clasificación de textos. Su proceso es parecido a las redes neuronales convencionales (un grafo acíclico) en el que reciben una entrada que se va transformando al pasar por varias capas de neuronas, pero en este caso en la convolucional la entrada es una imagen convertida en matriz tridimensional donde las capas se estructuran de forma tridimensional, y el resultado a partir de todas las transformaciones de la entrada por estas capas es lo que se conoce como clase o categoría perteneciente a dicha imagen procesada [29].

Las redes neuronales recurrentes, es un modelo recursivo, el cual se representa con un grafo que posee ciclos. Estos ciclos existentes poseen un impacto alto en la capacidad de aprendizaje de la red siendo principalmente aplicadas en el procesamiento de secuencias o series temporales. Pueden ser de estado discreto o estado continuo dependiendo las funciones de activación que se emplee [27]. En otras palabras, el modelo recursivo permite guardar la información de los datos salida, las cuales posteriormente son usadas como información para los próximos datos de entrada.

#### **1.4.6 Protocolo ESP-NOW**

Es un protocolo de comunicación desarrollado por Espressif Systems, siendo de corto alcance y baja potencia, se destaca por permitir la comunicación de múltiples dispositivos entre sí sin usar la conexión WiFi. Este protocolo es parecido a la conectividad inalámbrica de 2.4 GHz de baja potencia que se emplea en ratones inalámbricos. Antes de realizar la comunicación con este protocolo es

necesario primero que se realice el emparejamiento entre los dispositivos (se requiere saber la dirección MAC), una vez emparejados la conexión es segura (persistente), de igual a igual, sin requerir un protocolo de enlace, es decir, en caso de que uno de las dispositivos (o placas) pierda su alimentación o se reinicie, este automáticamente se conectará a los demás dispositivos una vez este energizado o se reinició por completo, de manera que continúe la comunicación entre todos los dispositivos emparejados [30].

Con el protocolo ESP-NOW, los datos de aplicación se encapsulan en un marco de acción específica que posteriormente son transmitidos de un dispositivo con WiFi a otro sin conexión WiFi. Para la protección del marco de acción específico del proveedor, emplea el método CCMP, que se describe en IEEE Std. 802.11-2012.

Este protocolo acepta las siguientes características:

- Comunicación unidifusión cifrada y no cifrada;
- Dispositivos homólogos cifrados y no cifrados mixtos;
- Se puede transportar una carga útil de hasta 250 bytes;
- Función de Devolución de llamada de envío y la posibilidad de configurarlo para que informe a la capa de aplicación sobre si fue exitosa o fracasó la transmisión.

El ESP-NOW también posee las siguientes limitaciones:

- La transmisión no es compatible;
- Pares cifrados limitados. Hasta 10 pares cifrados máximo en el modo Estación y hasta 6 máximo en el modo SoftAP o SoftAP + Estación;
- Acepta múltiples pares sin cifrar, pero su número total incluidos los pares cifrados, debe ser inferior a 20;
- La carga útil está limitada a 250 bytes.

El ESP-NOW es un protocolo de comunicación rápida que permite el intercambio de mensajes pequeños (hasta 250 bytes) entre placas de desarrollo ESP32. La velocidad de bits predeterminada de este protocolo es de 1Mbps [30]. Este protocolo versátil permite realizar la comunicación de forma unidireccional o bidireccional en varias configuraciones.

### 1.4.6.1 ESP-NOW: Comunicación Unidireccional

En este tipo de comunicación, se puede presentar varios escenarios, para diferentes tipos de aplicación, por ejemplo:

- **Una placa ESP32 enviando datos a otra placa ESP32:** este escenario es muy fácil de llevar a cabo su implementación y es óptimo para el envío de datos de una placa a otra, por ejemplo, las lecturas de medición de sensores o comandos de apagado e encendido para el control de un puerto GPIO (véase Figura 1.11) [31].



Figura 1.11 Esquema Comunicación Unidireccional simple con ESP-NOW [31].

- **Un ESP32 "maestro" que envía datos a múltiples "esclavos" ESP32:** este escenario es una configuración ideal en el desarrollo de un control remoto. El maestro o tablero ESP32 principal envía comandos iguales o diferentes a múltiples esclavos ESP32 (véase Figura 1.12).
- **Un "esclavo" ESP32 que recibe datos de múltiples "maestros":** este escenario es una configuración ideal cuando se recopila datos de varios nodos de sensores en una placa ESP32. El esclavo ESP32 se configura como una interfaz web que visualiza todos los datos de los demás tableros principales o maestros (véase Figura 1.12).



Figura 1.12 Comunicación unidireccional un Maestro-varios Esclavos (izquierda) y un Esclavo-varios Maestros (derecha) [30].

#### 1.4.6.2 ESP-NOW: Comunicación Bidireccional

En este tipo de comunicación cada placa o dispositivo emparejado puede ser un emisor y un receptor al mismo tiempo. Un escenario simple es la de dos placas comunicándose entre sí, otro escenario más complejo sería al agregar más placas a la comunicación tal como se observa en la Figura 1.13, esta configuración es similar a una red [31].

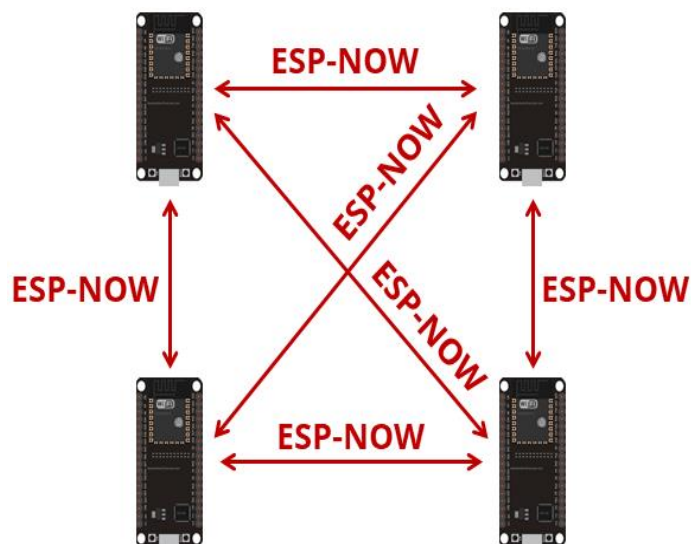


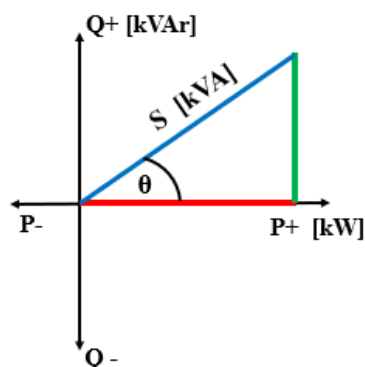
Figura 1.13 Comunicación bidireccional de múltiples placas ESP32 [30].

### 1.4.7 Factor de Potencia

Es una medida de eficiencia de un sistema eléctrico, que relaciona la cantidad de potencia activa y reactiva presente en la demanda energética de las cargas en el sistema, lo que se conoce como potencia aparente (S). La potencia activa (P) o energía útil, la usan los equipos eléctricos como maquinarias, electrodomésticos, entre otros, para realizar su operación tales como mover cargas, transporte en bandas, motores en ascensores, calefacción, refrigeración, etc. La potencia reactiva (Q) o “potencia perdida”, es la que se produce por los campos magnéticos necesarios que se generan para el funcionamiento de estos equipos eléctricos. El factor de potencia (fp) varía entre 0 y 1, donde a mayor potencia activa y menor potencia reactiva, el valor de fp tiende a 1 (lo ideal), y a menor potencia activa y mayor potencia reactiva, el valor de fp tiende a 0. Es por ello importante su control y monitoreo en los sistemas eléctricos, donde se recomienda mantenerlo a un mínimo de 0.95 con el fin de evitar consecuencias en la producción y la eficiencia del sistema, ya que esto puede perjudicar a daños posibles, por sobrecalentamiento de conductores, sobrecarga en las líneas de distribución, caídas de tensión, aumento de la planilla, daño o reducción de la vida útil de las cargas conectadas al sistema. El factor de potencia es determinado por la siguiente ecuación:

$$F_p = \cos(\theta) = P/S \quad (1.1)$$

La ecuación 1.1, es obtenida mediante el triángulo de potencias (ver Figura 1.14) que relaciona mediante fasores las potencias activa, reactiva y aparente presente en un sistema eléctrico.



**Figura 1.14 Triangulo de Potencia (predominante inductivo).**

Para la corrección de factor de potencia en sistemas industriales, se hace uso de dos alternativas; la primera alternativa consiste en hacer uso intermedio de la operación de motores síncronos que generen corriente adelantada de manera que genera la energía reactiva necesaria para corregir el factor de potencia, pero representa un costo considerable en su equipo así como su mantenimiento; y la segunda alternativa es mediante banco de capacitores o condensadores estáticos que es el método más sencillo y usado en industrias por su alcance en costo. Los condensadores permiten almacenar energía y genera la energía reactiva capacitiva necesaria para compensar la energía reactiva inductiva presente en el sistema energético [32].

Además, un banco de capacitores presenta ventajas tales como:

- Alto rendimiento (99% o más) para potencias pequeñas.
- La eficiencia de este equipo no depende del mantenimiento.
- Fácil conexión del equipo al circuito.
- No necesita combustible para su operación y
- La pérdida eléctrica por este equipo generalmente es por debajo de los 3 vatios por kVAr, etc.

En cuanto a las desventajas que presentan los bancos de capacitores son:

- La capacidad fija que impide que se adapten a las necesidades que requiera la carga
- Ondas armónicas presente cuando hay pequeñas cargas funcionando en la planta industrial debido a la resonancia del transformador con los capacitores, lo que produce un sobrecalentamiento en los capacitores, siendo necesario desconectarlos o reduciendo el menor número de capacitores para que no se dé la resonancia.

Aun existiendo las desventajas descritas en los bancos de capacitores, estas son mínimas y las cuáles cuentan con solución, por todo esto su aplicación en las industrias es muy alta [32].



### 1.4.8 Raspberry Pi 4

Es un microcontrolador, que consiste en uno de los ordenadores más básicos y comprados en el mundo. Usado para aprender a programar, como centro de medios y hogar inteligente, como cerebro de robots y núcleo de IA en red, etc. Entre las características de este dispositivo está: su operación es en forma silenciosa, consumo bajo de energía, redes rápidas, transferencia de datos 10 veces más rápida dado a su USB mejorada, variantes de memoria dependiendo a los proyectos a trabajar [33].

Entre características técnicas están:

- Cuenta con un procesador ARM Cortex-A72 de 4 núcleos a 1.5 GHz
- Permite la decodificación de video 4K a 60 fps.
- De dos puertos micro HDMI, dos puertos USB 2.0 y USB 3.0, un puerto USB-C de alimentación (debe ser de 15 W) en lugar del conector micro USB de alimentación, y 40 pines GPIO.
- Conectividad: Wi-Fi 802.11ac, Bluetooth 5.0, y Gigabit Ethernet.
- Memoria RAM en versiones de 2GB, 4GB y 8 GB.

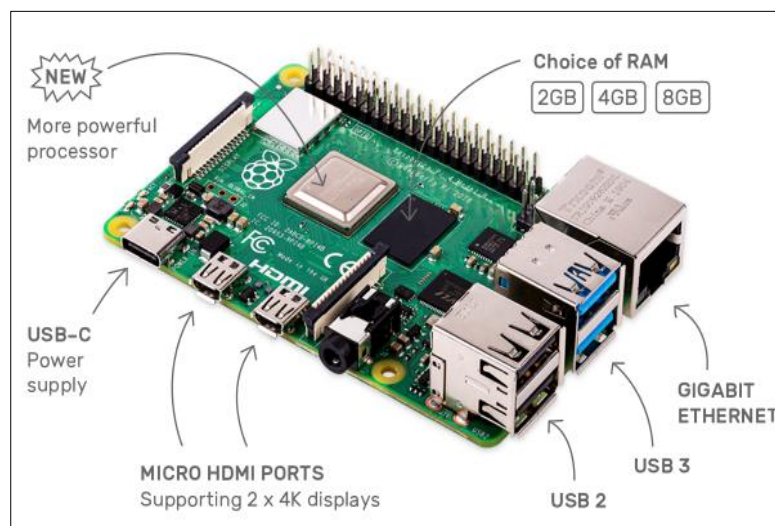


Figura 1.15 Raspberry Pi 4 model B [33].

### 1.4.9 ESP32

El microcontrolador ESP32, es un dispositivo de bajo costo y bajo consumo con sistema Wi-Fi/Bluetooth todo en uno, integrada y certificada que además de la radio inalámbrica, nos proporciona de un procesador integrado con interfaces para conectarse con varios periféricos. El integrado cuenta de dos núcleos de procesamiento con una arquitectura de 32 bits que se pueden controlar independientemente a frecuencias operativas entre 80 MHz y 240 MHz, memoria flash de 8 MB o 16 MB, y RAM de 512 KB. Los periféricos de este microcontrolador posibilitan la conexión a múltiples interfaces externas como:

- Transmisor receptor asíncrono universal (UART)
- Tarjetas SD
- Interfaz periférica serial (SPI)
- I<sup>2</sup>C
- I<sup>2</sup>S
- Ethernet
- Interfaces táctiles y capacitivas.

Existen varios módulos ESP32, según las necesidades de aplicación, por ello la selección adecuada de una placa de desarrollo ESP32 es vital para el diseño de una placa que se empleará en la producción o para el desarrollo de dispositivo en “bajo” o “alto” volumen en la planta de fabricación. Para llevar a cabo el prototipo se seleccionó el módulo ESP32-DEVKITC-32D (ver Figura 1.16), este módulo es una placa adaptadora sencilla para el microprocesador ESP32-WROOM-32D que facilita la programación y todos los circuitos de acondicionamiento que requiere un desarrollador o diseñador para empezar a realizar proyectos de aplicación de comunicación [34]. A continuación, se detalla las especificaciones técnicas de este módulo de ESP32:

- Procesador: ESP32-WROOM-32D (tecnología de 40nm)
- Voltaje de alimentación (por USB): 5V
- Rango de Frecuencia: 2.4 GHz a 2.5 GHz
- Conectividad: WiFi – 802.11 b/g/n, Bluetooth v4.2 BR/EDR
- Rango de temperatura de funcionamiento recomendado: -40°C a +85°C

- Posee 38 pines: 32 pines GPIO, 3 pines GND, 1 pin 5V, 1 pin 3.3V y un pin EN (Señal de activación del módulo – Activo alto).

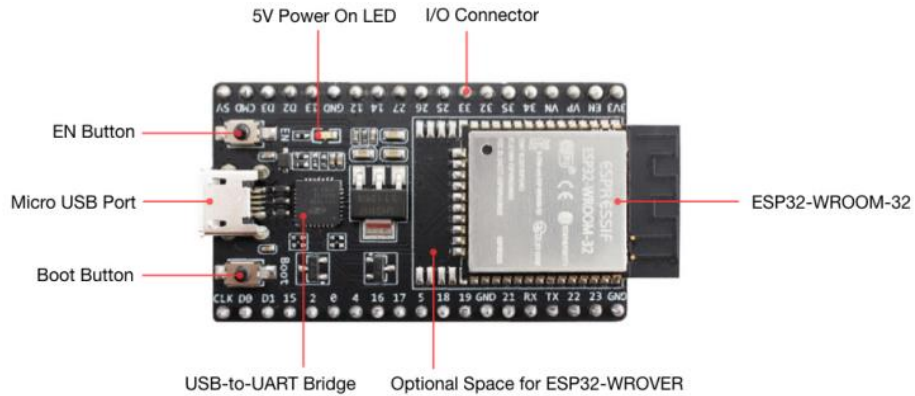


Figura 1.15 ESP32-DevKitC-32D [35].

En la Figura 1.16, se presenta la disposición de los pines en el módulo ESP32-DeviKitC-32D. La mayoría de los pines en este módulo permiten en la programación del ESP32 para habilitar varias funciones tales como ADC, DAC, PWM, I2S, I2C, SPI, entre otros, pero se recomienda no hacer uso de los pines GPIO6 al GPIO11 debido a que se puede interrumpir el acceso a la memoria flash SPI o SPI RAM integrado en el módulo, ya que su uso es de manera interna para la comunicación del ESP32 con la memoria flash SPI. Además, es importante indicar que los pines GPIO34, GPIO35, GPIO36 y GPIO39, su uso es solo para entrada, mientras que los demás pines GPIO son de entrada y salida [34].

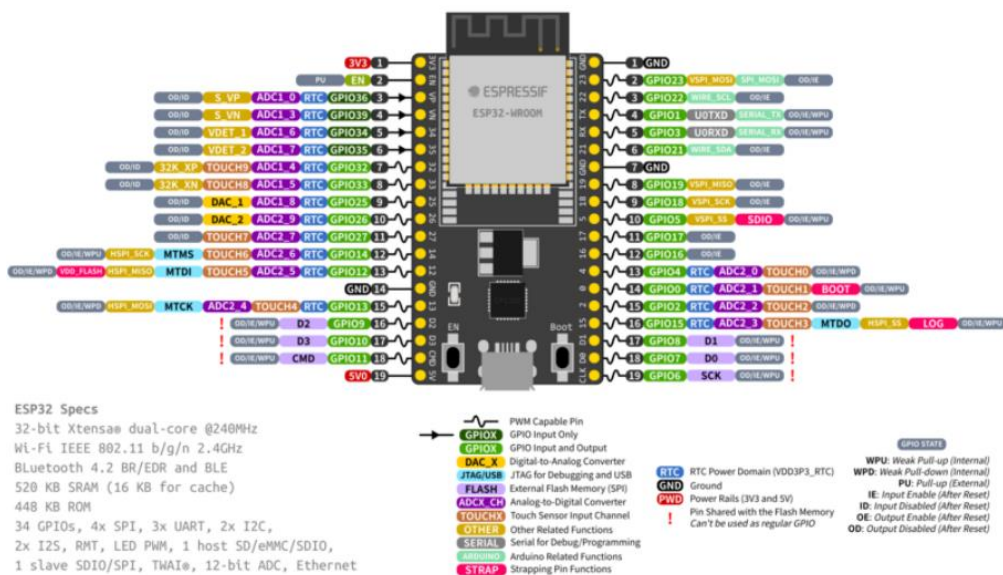


Figura 1.16 Pin Layout del ESP32-DevKitC-32D [35].

# CAPÍTULO 2

## 2 METODOLOGÍA

En esta sección se presenta la metodología a utilizar para la realización del proyecto, donde se especificará información clave para llevar a cabo el diseño del prototipo electrónico para la solución del problema a resolver. Para ello se realiza un análisis a la empresa EnergyFutur Energías Renovables Ecuador S.A., las variables a medir en la planta de acuerdo con los requerimientos de usuario/cliente, la selección de sensores y equipos a usar para el control y monitoreo del factor de potencia en tiempo real, la interfaz web para el monitoreo de las variables, el protocolo de comunicación y los lenguajes de programación a emplear en los microcontroladores. Por último, el proceso a utilizar para el desarrollo del algoritmo Machine Learning en el que se obtiene el modelo de predicción entrenado que será enviado y cargado al microcontrolador seleccionado que ejecutará el programa determinando el valor de carga capacitiva adecuado para la corrección de factor de potencia de la energía producida en la planta solar.

En la Figura 2.1, se detalla en un diagrama de bloques el modelo a seguir en el que se basa nuestro prototipo electrónico desarrollado.

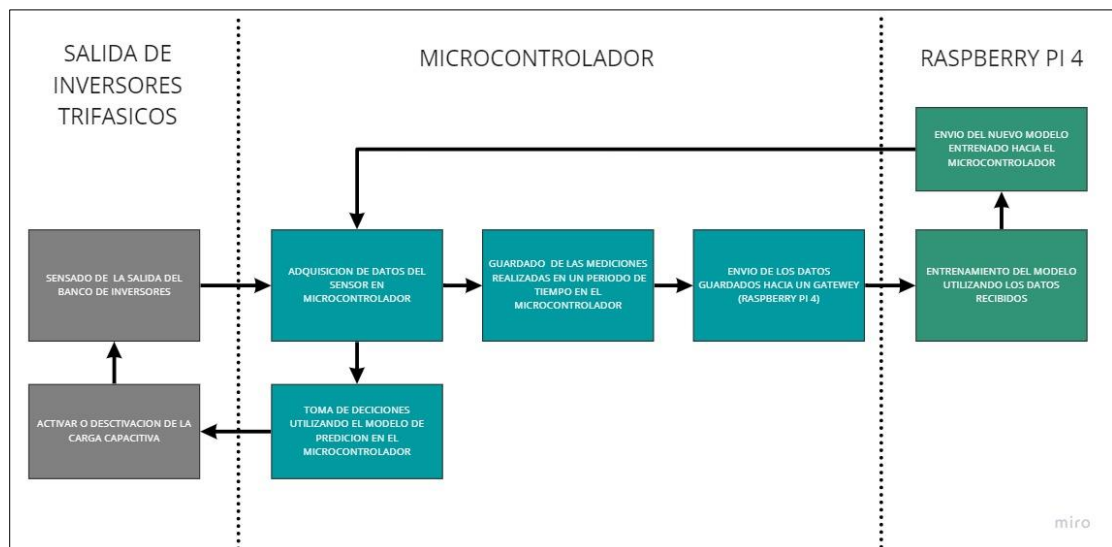


Figura 2.1 Diagrama de bloques del modelo a seguir.

## **2.1 Análisis de la empresa EnergyFutur Energías Renovables Ecuador S.A.**

La empresa Energyfutur Energías Renovables Ecuador S.A, es una empresa dedicada a la instalación y operación de generación de energía eléctrica, su matriz está ubicada en la Av. Francisco de Orellana, Villa: 27, en la ciudad de Guayaquil. Fundada el 18 de diciembre del año 2014 [36], enfoca el potencial humano en el ámbito de las energías renovables y la eficiencia energética, para el desarrollo, el cálculo, el diseño, el mantenimiento y proyección de instalaciones de Energía Solar Fotovoltaica, Energía eólica, Energía Biomasa y Geotermia, así como la investigación y difusión de FERNC [37]. Según menciona Francisco Vidal Pizarro, Director General de EnergyFutur Energía Renovables S.A, esta empresa verde actualmente en el mercado ecuatoriano cuenta con más de 5.5 MW<sub>peak</sub> instalado y de 5 plantas de fuente de energía solar fotovoltaica y entre otros proyectos que contribuyen al desarrollo sostenible del país mediante energía limpia que toma en consideración el entorno, tradiciones y la cultura de las zonas del país [38].

Es una empresa transnacional cuya matriz de origen es en el país de España. A continuación, se dará un detalle de la empresa EnergyFutur Energía Renovables S.L. y la empresa Brineforcorp que, en conjunto a esta empresa, trabajaron en el alza de la planta fotovoltaica Brineforcorp-Cabal Energy de 999 kW<sub>ac</sub> en Ecuador.

La empresa Energyfutur Energías Renovables S.L., es una empresa española fundada en el año 2008 que está dedicada y especializada en el sector de las energías renovables, desarrollando instalaciones de Energía Solar Fotovoltaica y Térmica, Energía Eólica y Biomasa. Su local se encuentra ubicado en la calle Mirandilla 54, Planta 1, oficina 2, ciudad de Benaguasil – España [39].

La empresa Brineforcorp S.A., es una empresa fundada en el año 2012, dedicada a instalaciones de equipos para la generación de energía eléctrica ya sea térmica, nuclear, mareomotriz, por turbina de diésel o gas y de otras fuentes renovables (solar fotovoltaica, hidroeléctrica, biomasa, etc) [40]. La empresa se encuentra ubicada en la Vía San Vicente-canoa Km 11 Y Vía Briceño-La Unión en el cantón San Vicente (Manabí – Ecuador).

En el siguiente apartado se explicará sobre el proyecto de la planta fotovoltaica Brineforcorp.

## **2.2 Central Brineforcorp-Cabal Energy**

La central fotovoltaica Brineforcorp-Cabal Energy, ubicada en sitio Briceño del cantón San Vicente, Manabí – Ecuador, su infraestructura está cerca del núcleo poblacional Nuevo Briceño en los terrenos de la familia Santos-González (entre plantaciones de teca y piscinas de cultivo de camarón). Es la segunda planta solar fotovoltaica de gran potencia alzada en esta provincia, posee una potencia nominal de campo fotovoltaico de 1.118 kWp (kilowatt peak o kilowatt potencia). Con este proyecto la provincia de Manabí cuenta de dos únicas plantas fotovoltaicas siendo una la central fotovoltaica Enersol y la otra la central fotovoltaica Brineforcorp donde está última es la de mayor potencia y energía bruta que genera en la provincia. La instalación de este parque solar se conforma del uso de 4.472 módulos fotovoltaicos del modelo YL250p-29b de la marca Yingli Solar. La potencia máxima es de 999 kWac en la conexión al SNI del país, donde se hace uso de inversores Freesun HESR-1001 de la marca Power Electronics. Este proyecto fue diseñado y dirigido por la empresa española Energyfutur Energías Renovables S.L., es el quinto proyecto en el que la firma interviene en nuestro país. Por otro lado, la empresa Energyfutur Energías Renovables Ecuador S.A. es la encargada de llevar el mantenimiento de esta planta fotovoltaica. Se estima que la producción de energía sea de 1.283.484 MWh anuales (aprox. 1,29 GWh), lo que equivale a satisfacer la demanda energética de más de 300 viviendas de la zona. Además, se recalca que es uno de los 22 proyectos de generación de energía que posee una potencia menor a los 1 MWac (o 1000 kWac) en el país [41].



**Figura 2.2 Vista de la Planta fotovoltaica Brineforcorp [41].**

A continuación, se describe al módulo fotovoltaico y el inversor que fueron seleccionados y empleados en el parque solar por la empresa española.

### 2.2.1 Módulo fotovoltaico modelo YL250p-29b

Este módulo fotovoltaico es parte de la serie YGE 60 CELL de Yingli Solar, está conformado de celdas o células solares policristalinos de tonalidad azulada y alta eficiencia con vidrio texturizado de alta transmisión (véase Figura 2.3) y que ofrece hasta 15.3% de eficiencia máxima, reduciendo costos de instalación y mayor salida de energía kWh del sistema por unidad de área [42].

La tabla 2.1, presenta las características más relevantes del módulo fotovoltaico modelo YL250p-29b de Yingla Solar.

**Tabla 2.1 Características del módulo fotovoltaico modelo YL250p-29b [43].**

<b>Potencia nominal STC</b>	250 W
<b>Potencia nominal PTC</b>	226.2 W
<b>Tolerancia de potencia</b>	0%/+2%
<b>Potencia por unidad de área STC</b>	153 W/m <sup>2</sup>
<b>Eficiencia máxima</b>	15.3%
<b>Número de células</b>	60
<b>Corriente a máxima potencia (I<sub>mp</sub>)</b>	8.24 A
<b>Voltaje a máxima potencia (V<sub>mp</sub>)</b>	30.4 V
<b>Corriente de cortocircuito (I<sub>sc</sub>)</b>	8.79 A
<b>Voltaje en circuito abierto (V<sub>oc</sub>)</b>	38.4 V
<b>Temperatura normal de funcionamiento de la célula (NOCT)</b>	46°C
<b>Voltaje máximo del sistema</b>	600 V
<b>Dimensiones (largo-ancho-profundidad)</b>	1650 mm – 990 mm – 40 mm
<b>Peso</b>	19.1 kg
<b>Método de instalación</b>	Montado en bastidor

La empresa EnergyFutur optó por este modelo y realizó la evaluación de los parámetros técnicos, garantías y certificados de Yingla Solar en este módulo fotovoltaico. En total 4472 paneles fueron empleados y unas 6 unidades más que son para reserva. Su selección se basó en mayor medida al factor del PID, donde

estos paneles demuestran resistencia al PID de acuerdo la norma IEC 62804, además, de ser factible económicamente para cumplir las necesidades de proyectos de fuentes generadoras de gran escala y ofrecer garantía de producto de 10 años y tolerancia de potencia positiva de 0 – 5 W [41].



**Figura 2.3** panel fotovoltaico Yingli YL250P-29b [42].

### **2.2.2 Inversor Freesun HESR-1001**

EnergyFutur para la selección de los inversores elaboró una tabla de selección de inversores, tomando en cuenta las características técnicas y costo del producto a cada modelo de inversor. Optó por los inversores Freesun HESR-1001 de Power Electronics debido a sus aspectos en rendimiento, asistencia servicio técnico, tiempo de entrega, garantía y coste. Para la implementación de estos inversores hicieron uso de un formato de sistema de Edificio Técnico (E.T.) prefabricado en hormigón con el fin de conservar el equipo de las condiciones de salinidad por proximidad al mar, además, que en su interior dispone de las medidas de protecciones en ac y dc (ver Figura 2.4). Este inversor de 1000 kVA se conforma de 8 módulos de 125 kW, un transformador en aceite de 1100 kVA y un conjunto de celdas de Media Tensión (M.T.) de 2L+P que posee SF6 (Hexafloruro de Azufre) que garantiza el corte y aislamiento eléctrico de todas las funciones en Alta Tensión (A.T.). El E.T. para su protección a las lluvias y radiación solar dispone de una envoltura a base de teja metálica junto a madera de teca. Este equipo por parte del fabricante cuenta de un String Supervisor SFS24 como concentrador de los strings y del Freesun Data center para el monitoreo del equipo, de esta manera mejora la eficiencia de las plantas y la protección (con fusibles de polos positivo y negativo, de sobre tensión integrada) de las series de módulos de inversores en la E.T [41].





Figura 2.4 E.T. prefabricado de hormigón donde se instalaron los inversores [41].

### 2.3 Variables que se miden en la salida del inversor.

En cuanto a los requerimientos del usuario/cliente, nuestra propuesta de prototipo electrónico para automatizar el proceso de corrección de fp del parque solar, realiza las siguientes mediciones que también pueden ser monitoreadas mediante una interfaz web empleada en un HMI. Estas variables se dividen en:

- ❖ **Variables medidas:** Voltaje, Corriente, Potencia, Energía [KWh], Frecuencia, Factor de potencia. Estas variables serán medidas mediante el sensor seleccionado (véase la sección 2.4.1) para el diseño del prototipo electrónico propuesto.
- ❖ **Variables calculadas:** Voltajes de línea a línea, potencias del triángulo de potencia trifásica. Variables que son determinadas en la programación mediante los datos adquiridos por el sensor seleccionado (véase la sección 2.4.1).

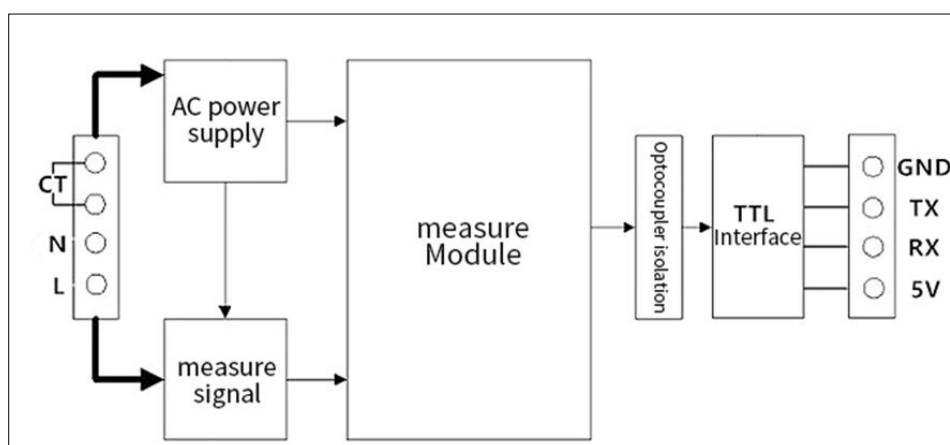
De las variables medidas, se creará un conjunto de datos (datos set) que serán los datos de entrenamiento a usar en el algoritmo de autoaprendizaje para su posterior entrenamiento para la obtención del modelo de predicción adecuado para el control de factor de potencia en la energía generada por la planta.

## 2.4 Selección de los sensores y dispositivos para el monitoreo.

En este apartado se describe el sensor y dispositivos seleccionados para el prototipo electrónico, tomando en consideración sus características técnicas, satisfacción de los requerimientos de las variables, la comunicación y factibilidad económica.

### 2.4.1 Sensor de energía Voltaje de Corriente AC PZEM004

El módulo de energía AC PZEM004T de la marca Peacefair, mide voltaje AC, corriente, potencia activa, energía, frecuencia y factor de potencia. La medición realizada puede ser consultada al módulo por medio de protocolo serie TTL.



**Figura 2.5 Arquitectura del sensor de energía Voltaje de Corriente multímetro AC PZEM 004 [44].**

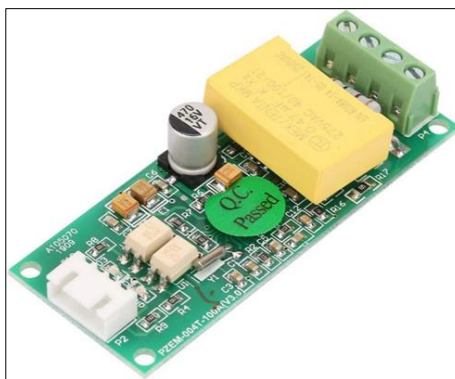
En la Figura 2.5, podemos observar la arquitectura que maneja nuestro módulo, este emplea una medición de corriente por bobina, lo que nos permite obtener mediciones de corriente alterna de manera no invasiva, es decir, no necesitamos abrir el circuito para colocar el sensor. Además de tener un aislamiento con respecto a la carga, debido que la corriente a medir no circula por el módulo de corriente.

La Tabla 2.2 muestra las características más importantes de los parámetros que podemos medir con el sensor AC PZEM004t.

**Tabla 2.2 Características del sensor de energía VoltajeCorriente AC PZEM004 [44].**

Parámetro	Rango medición	Resolución	Precisión
Voltaje	80~260V	0.1V	0.5%
Corriente	0~100A	0.02A*	0.5%
Potencia Activa	0~23kW	0.1W	0.5%
Energía Activa	0~9999.99kWh	1Wh	0.5%
Frecuencia	45~65Hz	0.1Hz	0.5%
Factor de Potencia	0.00~1.00	0.01	1%

La ventaja fundamental de este módulo AC, es que viene integrado con un microcontrolador el cual se encarga de todo el procesamiento de la medición de corriente alterna, de tal manera que, si deseamos obtener un parámetro, solo bastará con realizar una petición al módulo y este nos devolverá el valor de dicho parámetro. Ahorrándonos tener que realizar todo el cálculo nosotros mismo, tal como pasa con el sensor de voltaje Zmpt101b [45] y el sensor de corriente ACS712 [46], que para obtener los datos de voltaje y corriente necesitamos hacer cálculos rms, promedio y cruce por cero.



**Figura 2.6 PZEM-004t módulo de energía Voltaje de Corriente multímetro AC 80-260V 100A con transformador de corriente [47].**

Actualmente en el mercado podemos encontrar módulos de energía AC que comparten las características de nuestro modulo, por ejemplo, el sensor PZEM-014 que puede medir el mismo rango de voltaje y corriente, sea 80-260V y 0-100A respetivamente. Sin embargo, decidimos utilizar el PZEM-004T (ver Figura 2.6) debido al protocolo de comunicación que utiliza, y su precio está alrededor de los \$14.40 [47]. Además, en la Figura 2.5 podemos observar que el módulo tiene

una interfaz TTL, esta nos permite la comunicación serial con nuestro microcontrolador (ESP32 explicado en el capítulo anterior) a utilizar, obteniendo una comunicación directa con el módulo, a comparación del PZEM-014/016 AC [48] el cual maneja una comunicación RS485 la cual es útil en muchos aspectos de la automatización industrial, para microcontroladores no es el caso debido que se necesitaría de un módulo adicional para convertir el RS485 en una comunicación que pueda utilizar el microcontrolador sea I2C, UART, SPI.

#### 2.4.2 PANTALLA NEXTION

La interfaz de usuario es fundamental en el desarrollo del proyecto, esta permitirá mantener una comunicación con el usuario, logrando, mostrando información importante o también configuración del sistema. Para nuestro proyecto utilizamos las pantallas de la familia Nextion, puntualmente el modelo NX3224T028\_011.

Estas pantallas Nextion tienen la particularidad de ser completamente táctiles, lo que facilita la interacción hombre máquina, además de embeber en un solo entorno actuadores como botones o potenciómetros [49].

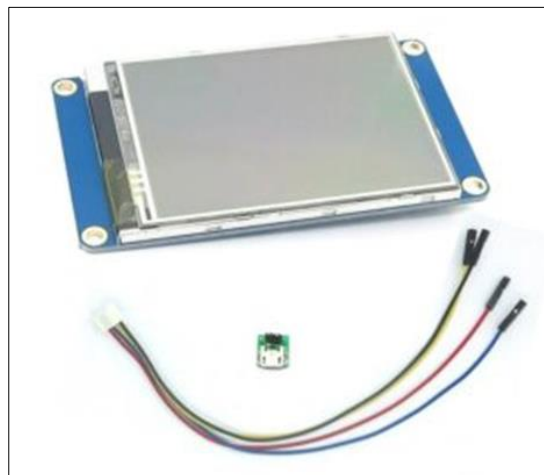
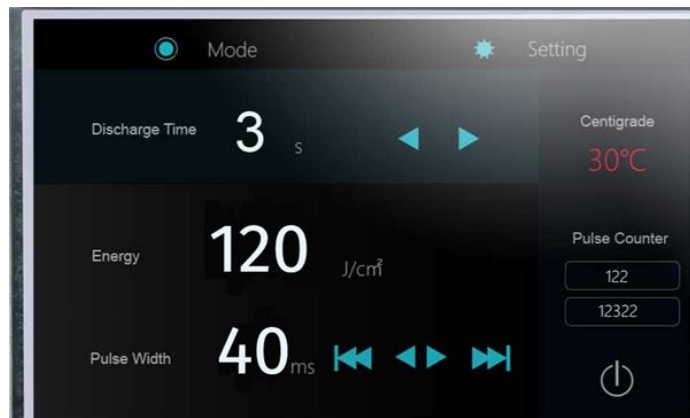


Figura 2.7 Nextion NX3224T028\_011 [50].



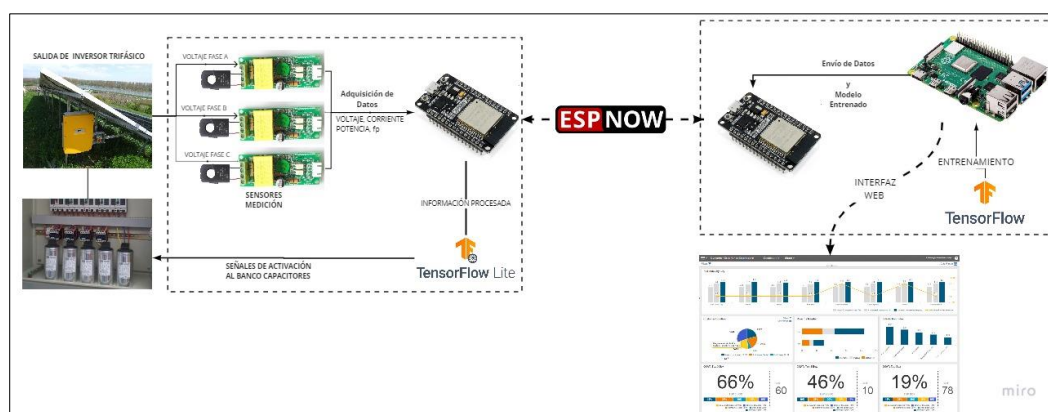
**Figura 2.8 Ejemplo de interfaz HMI en la pantalla Nextion NX3224T028\_011 [50].**

En comparación con otros HMI en el mercado, las pantallas Nextion tienen la particularidad de tener un microcontrolador programable dedicado para las funciones de la pantalla. Nuestro modelo NX3224T028\_011 utiliza un STM32 [50]. Esto conlleva una gran ventaja para nuestro proyecto, debido que la pantalla se programa por separado y lo único que se debe programar en nuestro proyecto es el envío y la recepción de datos en la pantalla. Esto conlleva a quitar procesamiento a nuestro microcontrolador, el cual puede ser utilizado en otras tareas. Por todas estas razones se decidió utilizar este tipo de pantalla antes que las OLED o display genéricos en los cuales todo el procesamiento de la imagen debe ser realizado en el microcontrolador.

Finalmente, estos dispositivos se integra al prototipo electrónico junto al microcontrolador ESP32 y a la Raspberry pi 4 especificado en el capítulo anterior, para su diseño a emplear y la respectiva programación para el modelo de predicción que permita al prototipo cumplir con la función de mejorar el factor de potencia en la producción de energía de la planta solar.

### **2.4.3 Diseño del prototipo a partir de los dispositivos seleccionados.**

Una vez seleccionados los dispositivos que conformaran el prototipo, se realizó un diagrama de flujo (ver Figura 2.9) que presenta de manera simple el proceso de la adquisición de datos, la comunicación entre dispositivos, monitoreo y control en tiempo real que ejecutará nuestro prototipo electrónico para el mejoramiento de factor de potencia de la energía generada de la planta fotovoltaica.



**Figura 2.9 Esquema en diagrama de flujo de los elementos que conforma el prototipo electrónico.**

#### 2.4.4 Diseño físico de la solución propuesta

El diseño físico del prototipo electrónico consiste en dos equipos, uno por bloque como se observa en la Figura 2.9. El primer equipo (bloque izquierdo), es el módulo de monitoreo en tiempo real (ver Figura 2.10) el cual contiene los sensores de voltaje-corriente, un módulo esp32, 3 relés, una antena, y la pantalla Nextion, todo esto dentro de una chasis compacto y seguro con protección IP65 para polvo y humedad. El segundo equipo (bloque derecho), es el que está conformado por un esp32 y el raspberry pi 4 adaptados con un módulo, este equipo es para el entrenamiento del algoritmo a partir del dataset que reciba de las mediciones del primer equipo, además cuenta con una antena para la comunicación con el otro equipo (ver Figura 2.11).



**Figura 2.10 Diseño físico del Módulo de monitoreo en tiempo real.**



**Figura 2.11** Diseño físico del Módulo del servidor principal y para el entrenamiento del algoritmo.

## **2.5 Lenguaje de programación**

En este apartado, se especifica el tipo de lenguaje de programación se empleó en cada dispositivo, los cuales tienen la tarea de ejecutar el algoritmo de predicción propuesto para la solución del proyecto.

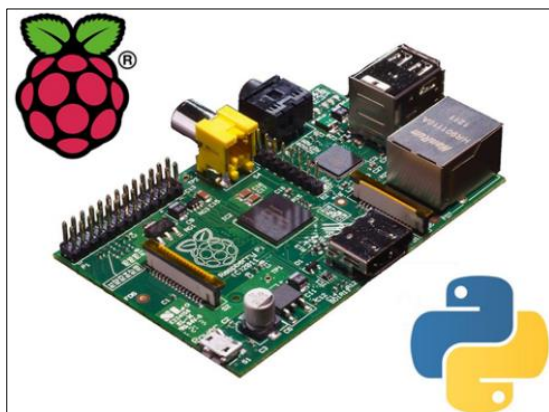
### **2.5.1 Microcontrolador (ESP32): Lenguaje Arduino**

Se escogió el lenguaje de programación de Arduino porque actualmente es el más utilizado en microcontroladores, está basado en C++ sin embargo su entorno de programación es más amigable a un lenguaje humano. Al ser openSource la comunidad ha creado un sinnúmero de librerías que nos ayudaran en la ejecución de nuestro proyecto. Su entorno de programación Arduino IDE al ser de código abierto, permitió que se crearan soportes para varios tipos de microcontroladores, de tal manera que podemos programar con Arduino, aunque nuestro microcontrolador no sea uno [51].

### **2.5.2 Raspberry Pi: Python 3**

Python es un lenguaje de programación de alto nivel, del tipo interpretado en el que emplea el tipado dinámico, esto quiere decir, cada instrucción es ejecutada sin previa compilación. Actualmente es uno de los lenguajes de programación más usados a nivel mundial, debido a que es un lenguaje versátil

que destaca por su sintaxis de código legible y limpio. Es un lenguaje multiplataforma, es decir, puede correr en diferentes sistemas operativos, entre ellos Linux, Android, entre otros. La Raspberry Pi tiene una distribución de Linux llamada Raspbian, en la cual se puede ejecutar código Python desde scripts muy sencillos hasta proyectos muy grandes [52].

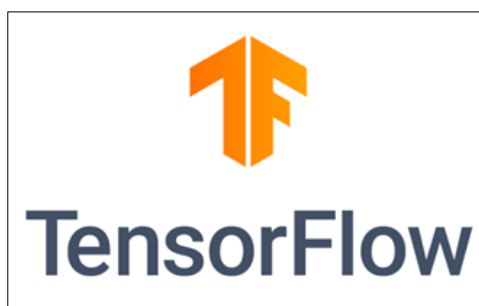


**Figura 2.13 Python en Raspberry pi [53].**

A continuación, se especifica la librería TensorFlow que puede ser usada en Python, para realizar aplicaciones de machine learning.

### **2.5.2.1 TensorFlow**

TensorFlow es una librería dedicada a la creación, entrenamiento y ejecución de redes neuronales. Su versatilidad permite ser ejecutado en varios tipos de arquitecturas CPU o del tipo GPU. En nuestro caso particular la utilizaremos en el lenguaje Python, con esta librería crearemos los algoritmos de predicción que serán previamente utilizados en los microcontroladores [53].



**Figura 2.14 Logo de TensorFlow [53].**



## 2.6 Protocolo de Comunicación

Para la comunicación de los dispositivos ESP32 y el Raspberry pi 4 (Gateway), se seleccionó el protocolo ESP-NOW debido a sus prestaciones en los requisitos de recursos del dispositivo, ancho de banda bajo de la red con bajo consumo de energía. Por consiguiente, se explica sobre que trata este protocolo, además, de otras alternativas de protocolos que puede emplearse para la comunicación de datos.

### 2.6.1 ESP-NOW

Es un protocolo de comunicación de corto alcance y baja potencia, que permite la comunicación de múltiples dispositivos entre sí sin usar la conexión WiFi. Es similar a la conectividad inalámbrica de 2.4 GHz de baja potencia. Antes de realizar la comunicación con este protocolo es necesario primero hacer el emparejamiento entre los dispositivos, una vez emparejados la conexión es segura (persistente), cada tarjeta de ESP32 tiene una dirección MAC única. Además, ofrece el envío de mensajes pequeños de hasta 250 bytes entre placas de desarrollo ESP32, una velocidad de bits predeterminada de 1Mbps, y un alcance de hasta 190 metros en la comunicación sin distorsión entre dispositivos en campo abierto [30].

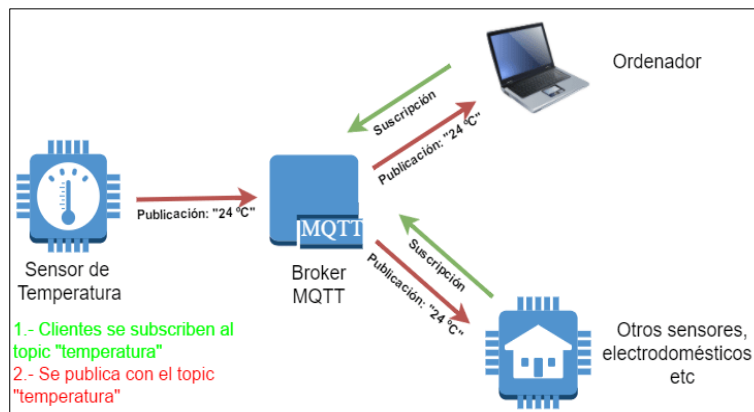
Se puede realizar la comunicación unidireccional (relación maestro-esclavo o viceversa) y comunicación bidireccional, en esta última los dispositivos actúan al mismo tiempo como emisor y receptor.



Figura 2.15 Comunicación Bidireccional entre dispositivos ESP32 [30]

## 2.6.2 MQTT

Este protocolo de comunicación M2M de tipo message queue, se basa de la pila TCP/IP como base para la comunicación y de estándar abierto OASIS e ISO (ISO/IEC PRF 20922) para protocolo de transporte de mensajería tipo publicación-suscripción, cliente-servidor. En MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación, donde su ejecución es sobre TCP/IP u otros protocolos de red que proporcionen conexiones bidireccionales ordenadas sin pérdidas. El Mqtt garantiza una entrega confiable en la transmisión de datos en redes de poca confianza o con bajo ancho de banda en la red. Además, es ideal para entornos restringidos por sus características en la implementación al ser liviana, simple y fácil de usar [54]. El protocolo MQTT brinda tres calidades de servicio para la entrega de mensajes: El **QoS 0**, que es un servicio de entrega de mensajes "como máximo una vez" según el entorno operativo, donde la pérdida de mensajes es posible, el uso del QoS 0 es cuando no importa si la conexión con los datos del sensor de lectura de la aplicación se pierde por un tiempo; otra calidad de servicio es el **QoS 1**, que es un servicio de entrega de mensajes "al menos una vez", pero que se puede dar el caso de duplicidad; y por último la calidad de servicio **QoS 2**, que garantiza que el servicio de entrega de mensajes sea de "exactamente una vez", además permite reducir el tráfico de red en los intercambios de los protocolos. El QoS 2, es apropiado para notificar a las partes interesadas (clientes y suscriptores) cuando se presenta una desconexión inusual, por ejemplo, es útil en sistemas de facturación donde la redundancia de la pérdida de mensajes genera la imposición de cargos incorrectos en estos sistemas. El MQTT tiene tres componentes constituyentes: El primer componente es el Editor/Productor (Cliente MQTT), el segundo es el Broker (un servidor MQTT), y el último componente es el Consumidor / Suscriptor (Cliente MQTT) [55].



**Figura 2.16 Ejemplificación del Protocolo MQTT [56].**

### 2.6.3 ZigBee

El protocolo ZigBee es una tecnología de comunicación de corto alcance y baja velocidad de datos, que opera en bandas de frecuencias de 868 MHz, 915 MHz y 2.4 GHz. Ofrece transferencia de datos de hasta 250 Kbps.

ZigBee es un estándar desarrollado por la Alianza ZigBee adaptada al estándar IEEE 802.15.4 en sus dos primeras capas, la capa física y la subcapa de acceso al medio (MAC) así también en la capa de red y de aplicación. Sus principales características son:

- Hardware y software de bajo costo
- Brinda baja latencia
- Bajo costo de dispositivos, instalación y su mantenimiento
- Eficiencia energética alta.
- Alcance típico corto menor a 50m.

ZigBee al igual que otros estándares que se usan en redes de corto alcance como Bluetooth y 802.11, es un protocolo desarrollado para una determinada clase de aplicación. Actualmente es el estándar más aceptado en redes de sensores y actuadores en el que su operación deba ser a batería [57].

## 2.6.4 Comparación de los protocolos descritos.

En este apartado se presenta la comparación entre los protocolos descritos, evidenciando en que aspectos brinda mejor fiabilidad para su uso en la aplicación del prototipo propuesto.

**Tabla 2.3 Comparación entre los protocolos de comunicación descritos [30,54,57].**

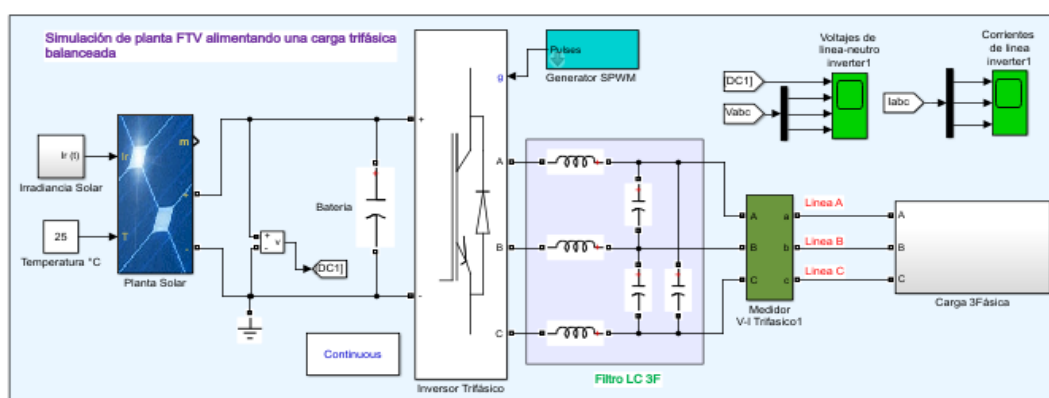
Protocolo	ESP-NOW	MQTT	ZigBee
Características			
Estándar basado	En las capas 1 y 2 del Modelo OSI	En las capas del 5 al 7 del Modelo OSI	En las dos primeras capas del IEEE 802.15.4
Consumo de energía	Bajo	Bajo	Bajo
Frecuencia de operación	2.4 GHz	-	868 MHz, 915 MHz y 2.4GHz
Velocidad de datos	Predeterminada de 1 Mbps	256 Mbps	Hasta 250 kbps
Alcance de transferencia	Menor a 190 metros	-	Menor a 50 metros
Requiere Internet para la comunicación entre dispositivos	No	Si	Si

En la Tabla 2.3 podemos observar algunas características, donde denotamos que el protocolo ESP-NOW seleccionado para el proyecto está basado en la capa física y enlace de datos del modelo OSI, justo al nivel de los actuadores y sensores, además podemos ver permite la comunicación sin tener conexión a WiFi siendo esto útil en la adquisición de datos de diferentes nodos de sensores cuando hay falta de un acceso a una red inalámbrica local.

## 2.7 Datos de entrenamiento

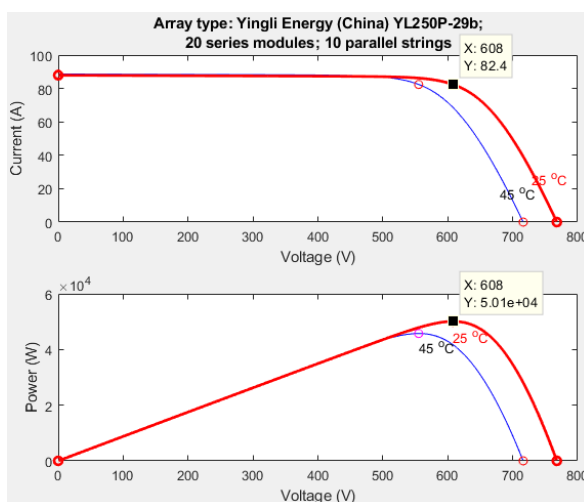
Debido a los inconvenientes presentados por la Pandemia, no nos fue posible ir a la planta solar para la recolección de datos a partir de la medición con el prototipo llevado a físico. Es por ello que, para llevar a cabo el entrenamiento del algoritmo de predicción a desarrollar (ver sección 2.8), se realizó una obtención de un data set mediante la ayuda del software simulink-matlab en donde se simuló una

planta solar de 200 paneles solares de la marca Yingla Solar especificado en la Tabla 2.1, estos paneles fueron configurados así: 20 en serie y 5 ramas en paralelos, siendo los valores en potencia máxima de 50KW cuando el voltaje es 608 Vdc y la corriente es 82.5 Adc. Con esta planta alimentando una carga trifásica, se realizó la simulación de 30 días variando la impedancia de la carga entre días con el fin de que el comportamiento sea un poco distinto al día anterior (ver Anexos 1 y 2). Los datos obtenidos corresponden a un mes de simulación, siendo mayor los cambios de potencia en los fines de semana con respecto a los días particulares.



**Figura 2.17 Esquema del sistema de planta solar empleado en Simulink para adquisición de datos.**

En la Figura 2.18, se presenta los valores de voltaje y corriente para un valor máximo de Potencia de 50 kW que puede entregar la planta de 200 módulos fotovoltaicos operando a una temperatura de operación de 25°C.



**Figura 2.18 Valor máximo de Potencia que pueda dar la planta solar simulada.**

## 2.8 Proceso de la creación del modelo de predicción del prototipo.

En este apartado se detalla el proceso realizado de inicio a fin en la programación del algoritmo machine learning para la obtención del modelo de predicción del prototipo, que ofrece de forma automatizada una solución óptima y eficiente del mejoramiento del factor de potencia en la energía suministrada en la planta solar durante las horas pico de demanda energética, horario donde el sistema fotovoltaico no puede generar la potencia activa necesaria para el aporte energético lo que se evidencia en una caída del factor de potencia en la energía generada. Para ello mediante diagramas de flujos se explicará el proceso que se llevó para el desarrollo de la solución propuesta del proyecto.

### 2.8.1 Preparación entorno de trabajo



Figura 2.19 Logo de Google Colab [54].

Para el proceso de creación del modelo se utilizó la herramienta Google Colab [58], este es un producto de la compañía Google que permite ejecutar código sin necesidad de instalar programas en nuestra máquina, además al ser un entorno virtual el código se ejecuta en los equipos de la compañía permitiéndonos obtener resultados rápidos en equipos de bajos recursos computacionales.

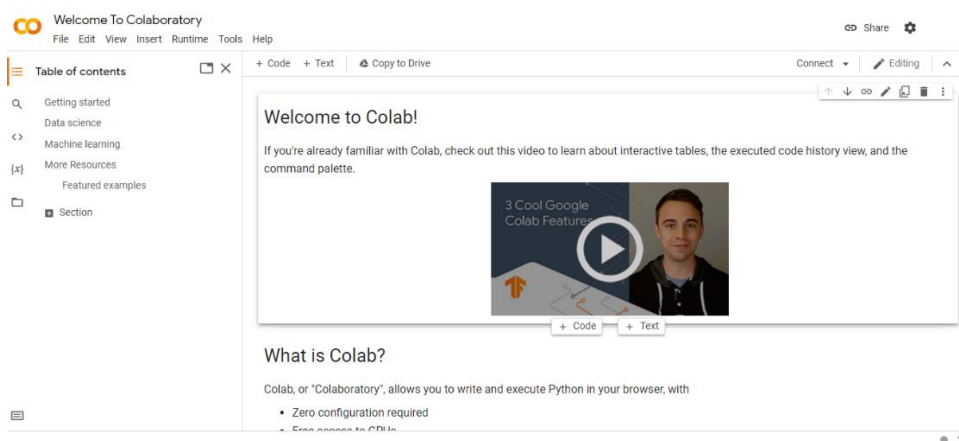


Figura 2.20 Entorno de Google Colab [54].

Como podemos observar en la Figura 2.20, Google Colab nos ofrece un entorno sencillo en donde podremos gestionar nuestros proyectos de una manera fácil e intuitiva. Para utilizarla debemos ingresar con nuestra cuenta Gmail. De manera nativa viene con soporte para Python con un sin números de librerías preinstaladas.

En nuestro caso particular al utilizar la librería TensorFlow requerimos tener versiones específicas de algunas librerías por lo que al inicio de nuestro proyecto procedemos a instalar las versiones, como podemos visualizar en la Figura 2.21.

```

!pip install tensorflow-gpu~=2.1.1
!pip install numpy~=1.19.5
!pip install pandas~=1.3.5
!pip install matplotlib~=3.1.3
!pip install hexdump~=3.3
!pip install pyserial~=3.5

```

**Figura 2.21 Instalación de librería de TensorFlow y otras específicas.**

Una vez terminado el proceso podemos iniciar con la creación de nuestro modelo a partir de machine learning.

### 2.8.2 Procesamiento datos de entrenamiento

Para el entrenamiento se creó un CSV en donde se recopiló todos los datos simulados en Matlab, se procedió a descargarlos en nuestro proyecto, dándonos una tabla con toda la información, como podemos observar en Figura 2.22 a continuación.

	day	weekday	second_hora	VA	IA	FPA	VB	IB	FPB	VC	IC	FPC	FP3
0	1	2	2.16	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	NaN
1	1	2	12.16	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	NaN
2	1	2	22.16	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	NaN
3	1	2	32.16	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	NaN
4	1	2	42.16	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
156266	31	8	50362.16	121.797031	0.001251	0.863339	120.203471	0.001263	0.862262	122.936589	0.001279	0.872363	0.998659
156267	31	8	50372.16	121.871520	0.001251	0.864471	120.005134	0.001258	0.861573	122.543463	0.001277	0.871926	0.998842
156268	31	8	50382.16	121.961238	0.001253	0.865398	119.998407	0.001256	0.861317	122.245969	0.001276	0.871267	0.999092
156269	31	8	50392.16	122.092467	0.001255	0.866033	120.119254	0.001255	0.861282	122.109749	0.001276	0.870678	0.999288
156270	31	8	50402.16	122.281338	0.001257	0.866519	120.273709	0.001256	0.861195	122.093285	0.001276	0.870283	0.999407

156271 rows x 13 columns

**Figura 2.22 Archivo CSV de los datos simulados en Matlab.**

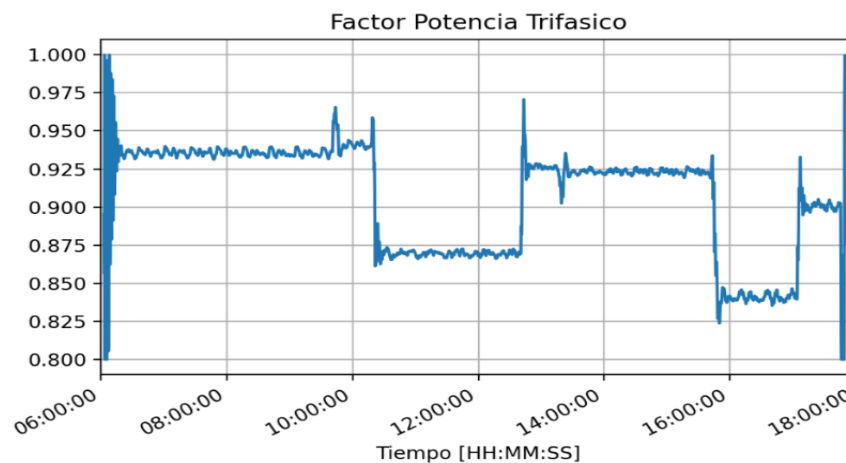
Lo primero que notamos fue que teníamos datos corrompidos tales como NAN, esto se esperaba debido a los elementos utilizados en la simulación. Por lo que antes de usarlos se procesaron la información (ver Anexo 3), eliminando todas las filas cuyos valores no estuvieran definidos, dándonos la siguiente Tabla 2.4.

**Tabla 2.4 Procesamiento de los datos del CSV de los datos simulados.**

day	weekday	second_hora	VA	IA	FPA	VB	IB	FPB	VC	IC	FPC	FP3	
17	1	2	21772.16	0.801235	0.082278	0.620309	0.468186	0.087618	0.531812	0.786852	0.106296	0.844455	0.874028
18	1	2	21782.16	1.275625	0.115637	0.824830	0.605347	0.096066	0.074327	0.981904	0.162075	0.898706	0.851470
19	1	2	21792.16	1.878474	0.195535	0.925141	1.097659	0.093291	-0.000812	1.065483	0.226651	0.879538	0.999996
20	1	2	21802.16	2.501236	0.315222	0.951126	1.937990	0.114402	0.513473	1.033608	0.283957	0.729613	0.800000
21	1	2	21812.16	3.011560	0.459830	0.949513	3.038630	0.216096	0.835615	1.184722	0.317360	0.280310	0.800000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
156266	31	8	71962.16	121.797031	0.001251	0.863339	120.203471	0.001263	0.862262	122.936589	0.001279	0.872363	0.998659
156267	31	8	71972.16	121.871520	0.001251	0.864471	120.005134	0.001258	0.861573	122.543463	0.001277	0.871926	0.998842
156268	31	8	71982.16	121.961238	0.001253	0.865398	119.998407	0.001256	0.861317	122.245969	0.001276	0.871267	0.999092
156269	31	8	71992.16	122.092467	0.001255	0.866033	120.119254	0.001255	0.861282	122.109749	0.001276	0.870678	0.999288
156270	31	8	72002.16	122.281338	0.001257	0.866519	120.273709	0.001256	0.861195	122.093285	0.001276	0.870283	0.999407

155744 rows x 13 columns

En donde todos los valores son válidos para el entrenamiento, para validar esto se procedió a graficar el factor de potencia trifásico de un día al azar y verificar que todo este correcto.



**Figura 2.23 Gráfica del fp trifásico de un día al azar en el sistema de planta solar realizado en Simulink.**



### 2.8.3 Propuestas de modelos para la predicción de factor de potencia

En este apartado se detalla las propuestas para el modelo a utilizar para la predicción de factor de potencia trifásico, se describe el procesamiento de los datos. Cada uno representa una arquitectura diferente con diferentes entradas y capas ocultas, compartiendo en todos los casos una capa de salida que se busca obtener el factor de potencia en el futuro.

#### 2.8.3.1 Modelo 1

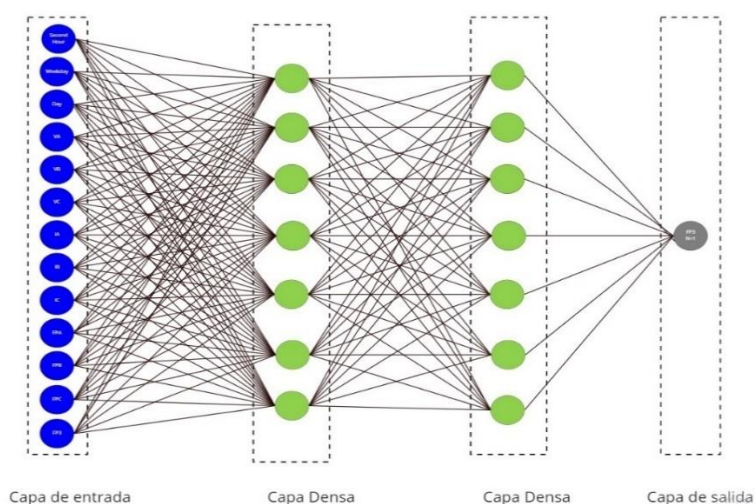


Figura 2.24 Propuesta de Modelo 1.

Nuestra primera propuesta (ver Figura 2.24), fue el modelo más sencillo, un secuencial con una capa de entrada que recibe parámetros de fecha, hora y datos eléctricos de la carga, por medio de capas densas se llega a nuestra capa de salida. En este modelo el objetivo es predecir el comportamiento futuro en base del comportamiento actual.

```
Model: "Model_1"
-----
Layer (type)                Output Shape          Param #
-----
Capa_Entrada (Dense)        (None, 32)            448
Capa_Ocultas_1 (Dense)      (None, 16)            528
Capa_Ocultas_2 (Dense)      (None, 8)             136
Capa_Salida (Dense)         (None, 1)              9
-----
Total params: 1,121
Trainable params: 1,121
Non-trainable params: 0
```

Figura 2.25 Capas de entrada y salida en el Modelo 1 propuesto.

La Figura 2.25 corresponde al modelo 1 propuesto para nuestro proyecto, donde podemos visualizar la cantidad de neuronas empleadas en cada capa oculta desde los datos de entrada a la salida. Como solo necesitamos de una salida utilizamos la estrategia descendente en donde comenzamos con una gran cantidad de neuronas y procedemos disminuir hasta llegar a la salida necesaria.

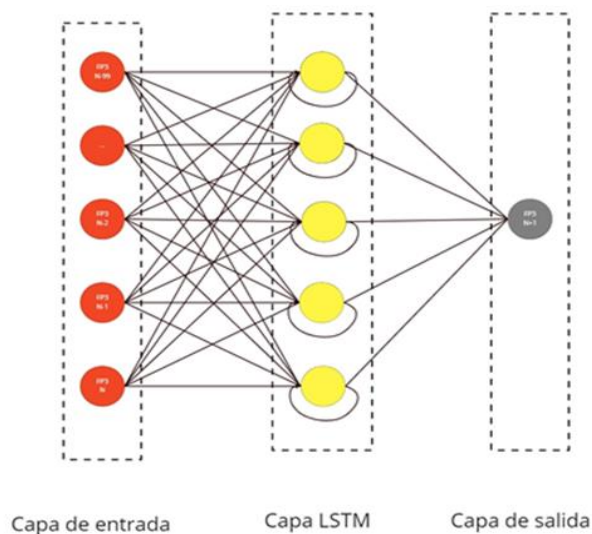
	day	weekday	second_hora	VA	VB	VC	IA	IB	IC	FPA	FPB	FPC	FP3
0	11.0	4.0	30832.16	153.021961	152.886362	152.888512	24.115510	24.100383	24.107434	0.582978	0.582595	0.582969	0.911011
1	7.0	8.0	28852.16	149.143959	149.116372	149.188730	24.121980	24.128065	24.142286	0.633055	0.632767	0.633204	0.935712
2	6.0	7.0	43402.16	158.169194	158.178808	158.034544	23.824020	23.792214	23.793972	0.507115	0.506830	0.505928	0.868808
3	3.0	4.0	26292.16	145.801558	145.543828	145.754952	23.981225	23.977345	24.010437	0.670464	0.669833	0.671180	0.952508
4	8.0	1.0	36962.16	163.605081	163.938500	163.581901	23.050924	23.055405	23.008021	0.413169	0.414007	0.412133	0.811371
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69626	4.0	5.0	71632.16	117.978495	116.961705	116.223037	18.308859	18.123478	18.360282	0.564121	0.546347	0.554336	0.895206
69627	4.0	5.0	35282.16	163.688609	164.027163	163.659075	23.050273	23.041917	22.983169	0.411747	0.413341	0.410624	0.810096
69628	6.0	7.0	51652.16	150.832530	150.735203	150.969985	24.048241	24.087404	24.101518	0.606018	0.605867	0.607504	0.924263
69629	2.0	3.0	68572.16	150.067094	149.342685	149.299169	23.266205	23.146837	23.278766	0.559417	0.553831	0.556436	0.897412
69630	2.0	3.0	64092.16	154.795747	154.155952	154.611774	23.983393	23.949917	24.056844	0.557649	0.554875	0.558238	0.899991

69631 rows x 13 columns

**Figura 2.26 CSV de los datos obtenidos a partir del Modelo 1.**

Los datos de entrada para el entrenamiento lo podemos observar en la Figura 2.26, en donde podemos destacar que a diferencia de los datos obtenidos del CSV de la Figura 2.22 estos se encuentran completamente en desorden. Se utilizó esta técnica porque en este modelo cada predicción se basa solamente en el ahora, esto evita que el modelo aprenda patrones y sea más eficiente.

### 2.8.3.2 Modelo 2



**Figura 2.27 Propuesta de Modelo 2.**

En esta propuesta se optó por dejar de lado los modelos simples para dar paso a un modelo recursivo (ver Anexo 4 y 5), como se describió en el marco teórico, los modelos recursivos guardan información y los datos de salida sirven de información para los siguientes datos de entrada. A diferencia del primer modelo que cada predicción es individual sin importar sucesos pasados, en este modelo se busca diferenciar patrones y comportamientos en la carga.

```

Model: "Model_2"
-----
Layer (type)                Output Shape                Param #
-----
Capa_Ocultas (LSTM)         (None, 50)                 10400
-----
Capa_Salida (Dense)         (None, 1)                 51
-----
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
  
```

**Figura 2.28 Capas de entrada y salida en el Modelo 2 propuesto**

La Figura 2.28, corresponde al modelo 2 creado en nuestro proyecto, donde podemos visualizar como la cantidad de neuronas en la capa oculta (LSTM, tipo de dato para redes neuronales recurrentes) que emplea está modelo.

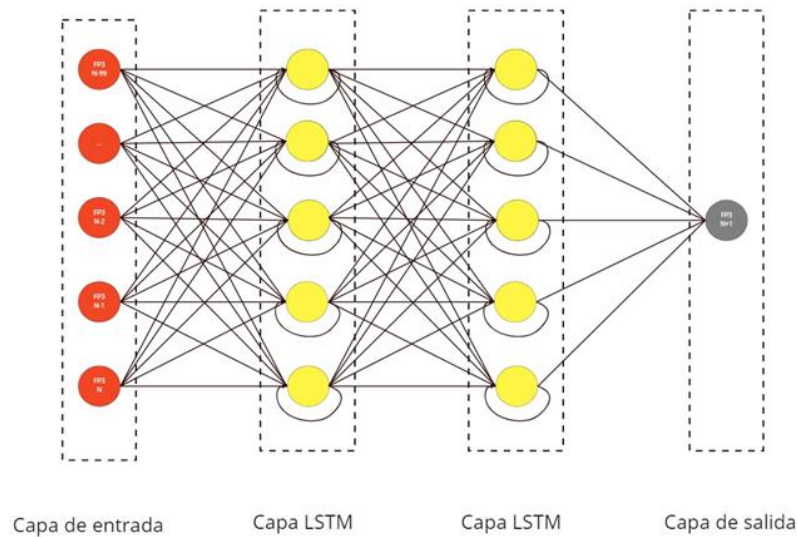
	(N-39)	(N-38)	(N-37)	(N-36)	(N-35)	(N-34)	(N-33)	(N-32)	(N-31)	(N-30)	(N-29)	(N-28)	(N-27)	(N-26)	(N-25)	(N-24)	(N-23)	(N-22)
0	0.874028	0.851470	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000	0.800000	0.901902
1	0.851470	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000	0.800000	0.901902	0.999125
2	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000	0.800000	0.901902	0.999125	0.968808
3	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000	0.800000	0.901902	0.999125	0.968808	0.955017
4	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000	0.800000	0.901902	0.999125	0.968808	0.955017	0.992195
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
70291	0.999963	0.999995	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052	0.999013	0.999169
70292	0.999995	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052	0.999013	0.999169	0.999457
70293	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052	0.999013	0.999169	0.999457	0.999736
70294	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052	0.999013	0.999169	0.999457	0.999736	0.999900
70295	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052	0.999013	0.999169	0.999457	0.999736	0.999900	0.999954

70296 rows x 40 columns

**Figura 2.29 CSV de los datos obtenidos a partir del Modelo 2.**

A diferencia del modelo 1, esta capa recibe como entrada los factores de potencia actual y pasados, de tal manera que utiliza la información anterior para predecir la nueva. Esto lo podemos ver mejor en la Figura 2.29, en donde N representa la muestra actual y N-39 representa 39 muestras antes. En este caso los datos no se colocaron al azar debido a que se busca encontrar un patrón.

### 2.8.3.3 Modelo 3



**Figura 2.30 Propuesta de Modelo 3.**

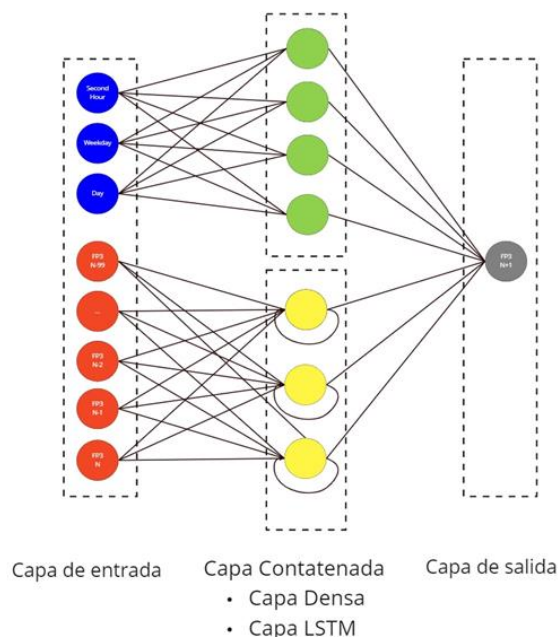
En esta propuesta el modelo sigue siendo un recurrente con la diferencia de agregar una capa más, las capas LSTM tienen la oportunidad de pasar sus estados a la siguiente capa, para que puedan compartir información entre ellas. En este modelo se sigue buscando encontrar un patrón en el comportamiento del factor de potencia.

```
Model: "Modelo_3"
-----
Layer (type)                Output Shape          Param #
-----
Capa_Ocultas_1 (LSTM)       (None, 50, 50)       10400
Capa_Ocultas_2 (LSTM)       (None, 25)            7600
Capa_Salida (Dense)         (None, 1)             26
-----
Total params: 18,026
Trainable params: 18,026
Non-trainable params: 0
```

**Figura 2.31 Capas de entrada y salida en el Modelo 3 propuesto**

Para este modelo se utilizó la misma matriz de entrada del modelo 2. En la Figura 2.31 podemos visualizar como la cantidad de neuronas en cada capa oculta (LSTM) que emplea está modelo.

### 2.8.3.4 Modelo 4



**Figura 2.32 Propuesta de Modelo 4**

En el modelo 1 se buscaba obtener una predicción en base a información proporcionada del ahora, en los modelos 2 y 3 se buscaba obtener la predicción a comportamientos pasados. Para el modelo 4 se buscó obtener las ventajas de los dos tipos. En ese modelo se tiene una capa concatenada en donde unos datos de la entrada se ubican en una capa Densa y los otros en una capa LSTM, de tal manera que tenemos una capa más compleja. Con esto se buscó que el modelo encuentre patrones y además tener en consideración parámetros de tiempo día.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 53, 1)]	0	
lambda (Lambda)	(None, 3, 1)	0	input_1[0][0]
dense_16 (Dense)	(None, 3, 50)	100	lambda[0][0]
lambda_1 (Lambda)	(None, 50, 1)	0	input_1[0][0]
flatten (Flatten)	(None, 150)	0	dense_16[0][0]
lstm (LSTM)	(None, 50)	10400	lambda_1[0][0]
concatenate (Concatenate)	(None, 200)	0	flatten[0][0] lstm[0][0]
dense_17 (Dense)	(None, 1)	201	concatenate[0][0]
Total params: 10,701			
Trainable params: 10,701			
Non-trainable params: 0			

**Figura 2.33 Capas de entrada y salida en el Modelo 4 propuesto**

Como se observa en la Figura 2.33, este modelo es más complejo debido que se diseñó los diferentes partes del modelo y se realizó la capa concatenación para unir dos capas de Densa y LSTM.

	Day	WeekDay	Second_hour	(N-42)	(N-41)	(N-40)	(N-39)	(N-38)	(N-37)	(N-36)	(N-35)	(N-34)	(N-33)	(N)	
0	1.0	2.0	22172.16	0.874028	0.851470	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066
1	1.0	2.0	22182.16	0.851470	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117
2	1.0	2.0	22192.16	0.999996	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925
3	1.0	2.0	22202.16	0.800000	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000
4	1.0	2.0	22212.16	0.800000	0.800000	0.800000	0.800000	0.970722	0.932685	0.825741	0.875066	0.996117	0.898925	0.800000	0.800000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
70291	14.0	7.0	71962.16	0.999963	0.999995	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787
70292	14.0	7.0	71972.16	0.999995	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643
70293	14.0	7.0	71982.16	0.999929	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456
70294	14.0	7.0	71992.16	0.999881	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236
70295	14.0	7.0	72002.16	0.999904	0.999956	0.999987	0.999995	0.999988	0.999958	0.999892	0.999787	0.999643	0.999456	0.999236	0.999052

70296 rows x 43 columns

**Figura 2.34 CSV de los datos obtenidos a partir del Modelo 4.**

Con respecto a la entrada, se realizó el mismo proceso del modelo 2 y 3, en el que consistió obtener los n datos antes del actual, agregando los datos de día, semana y horas para la parte de capas densa.

## 2.8.4 Pruebas y Selección Modelo

Los 4 modelos propuestos fueron entrenados con los primeros 30 días de la semana como datos de entrada, para cada caso el entrenamiento obtuvo los datos esperados según su arquitectura. Una vez finalizado el entrenamiento se realizó un Test utilizando el último día del mes siendo este el 31, de esta manera se evaluaría las perdiciones y no la memoria del modelo (ver Anexo 6).

Para la selección del modelo se realizaron las siguientes validaciones:

- Visual una gráfica donde se puede ver el comportamiento real y el que simula el modelo, de esta manera se puede observar las diferencias.
- **Error cuadrático medio:** calculo estadístico entre valor real y precedido, en donde se desea visualizar la eficiencia del modelo.
- **Error absoluto:** operación con la que se busca observar la diferencias entre el valor real y el simulado, con este podremos obtener el mayor y menor error de factor de potencia.

- **Tiempo entrenamiento:** Como su nombre lo indica, en este Test se logra saber que tanto el módulo se acerca al valor real y cuál es el coste en segundos que tarde en realizar el entrenamiento.

## **2.9 Proceso medición y envió datos**

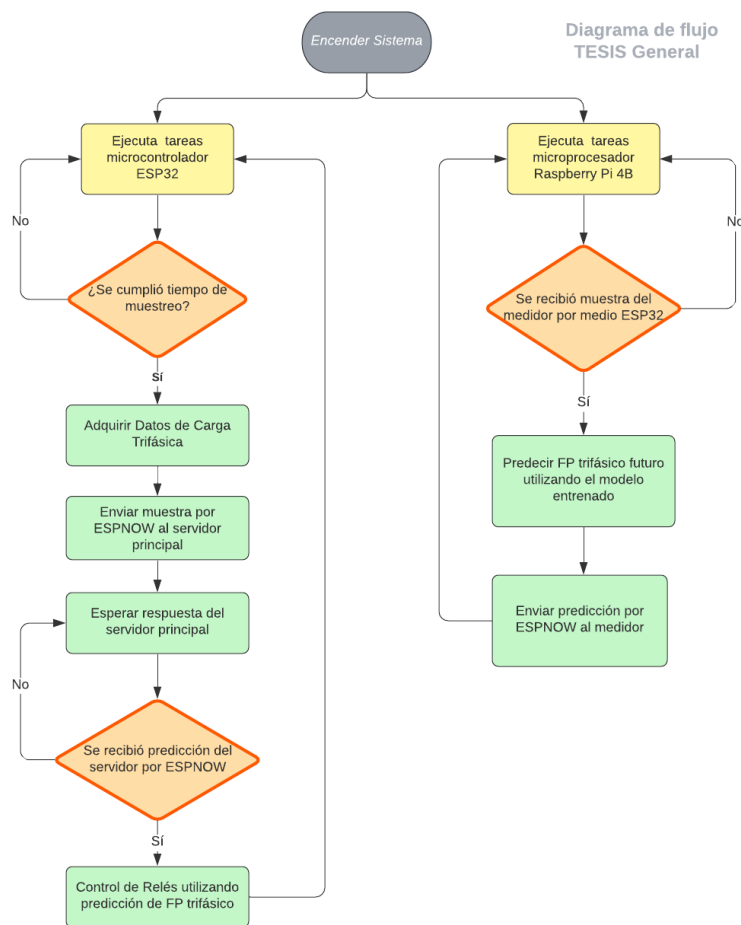
En este apartado, se presenta en diagrama de flujos el proceso de medición, predicción y envió de datos que realiza el sistema de los equipos una vez inicializado su operación.

### **2.9.1 Diagrama general de bloques de la solución propuesta.**

En la Figura 2.35 se muestra el diagrama de flujo general del presente proyecto, el cuál ejemplifica el proceso que lleva a cabo el equipo de monitoreo y el servidor principal mencionado anteriormente (ver sección 2.4.4). El desarrollo inicia una vez encendido el sistema, posteriormente pasa a la ejecución de tareas en cada equipo

Iniciado la ejecución de tareas en el microcontrolador del equipo de monitoreo en tiempo real, y una vez verificado el tiempo de muestreo, procede a realizar la adquisición de datos de la medición eléctrica de cada fase en la salida del inversor, que luego son enviados al servidor principal mediante la comunicación por ESP-NOW (ver Anexo 7), el cuál realiza un guardado de estas muestras en una base datos SQL Lite para su posterior uso cuando se inicie el entrenamiento a partir de un dataset seleccionado (ver Figura 2.38).

Finalmente, con el modelo entrenado predictivo, predice el fp trifásico futuro que luego es enviado al ESP32 del módulo de monitoreo que realizará de ser necesario el control de los relés para corregir el fp de la energía suministrada (más detalle en la Figura 2.37).



**Figura 2.35 Solución general del sistema en diagrama de flujo.**

### 2.9.2 Diagramas de bloques por etapas específicas de la solución propuesta.

La primera etapa para mencionar, son las tareas que ejecuta el ESP32 del módulo de monitoreo. El proceso empieza con la configuración e iniciación de la librería freeRTOS con el que se lleva la programación para configurar, el servidor Web del medidor, la comunicación con ESP-NOW, y la obtención de datos por los sensores voltaje Corriente AC (ver Figura 2.36).

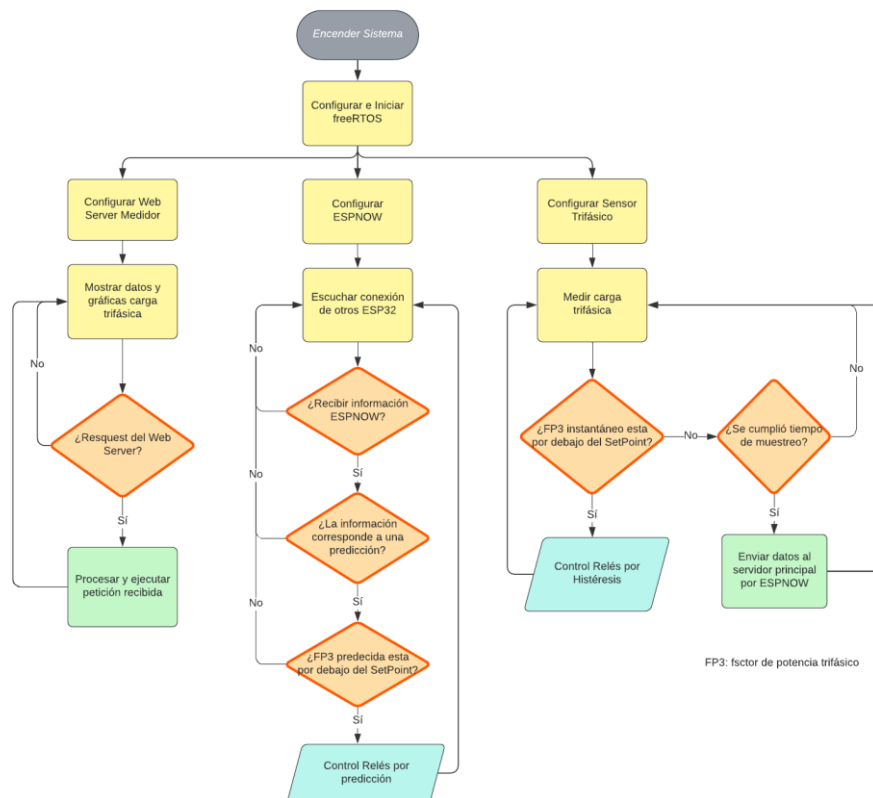
Configurado el servidor web, procede a mostrar los datos y gráfica medidas en tiempo real en la pantalla del módulo de monitoreo, en el cual el personal a cargo podrá seleccionar que medida eléctrica visualizar.

Configurado la comunicación con ESPNOW, se procede a establecer la comunicación con el ESP32 del servidor principal, para ello mediante una bocina se informa cuando la conexión se ha establecido. Recibe la información del fp



trifásico predicha con el modelo predictivo y posteriormente toma la acción de ejecutar el control predictivo de relés cuando está por debajo del valor deseado.

Configurado los sensores voltajes corriente AC, se procede a empezar la medición eléctrica en la salida del inversor. Cuando el fp trifásico instantáneo medido está por debajo del valor deseado (0.95 o más), el microcontrolador manda una señal para el control de los relés por histéresis, caso contrario esos datos son enviados al servidor principal si están dentro del tiempo de muestreo definido.

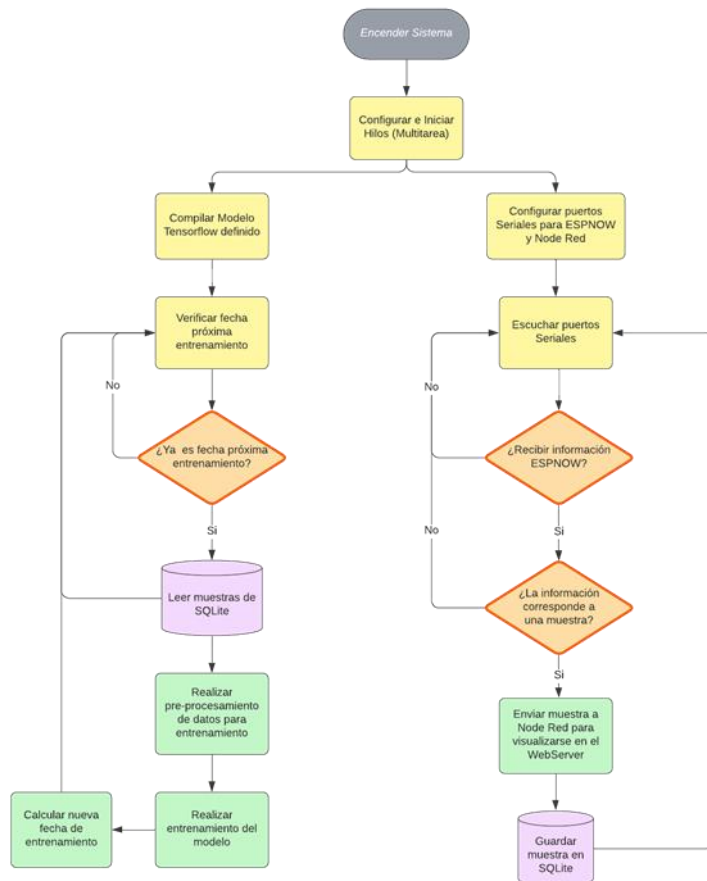


**Figura 2.36 Diagrama de flujo de las tareas del ESP32 del módulo de Monitoreo en tiempo real.**

La segunda etapa para mencionar, son las tareas que ejecuta el Raspberry del servidor principal. El proceso empieza con la configuración e iniciación de hilos (Multitarea) con el que se lleva la programación para compilar el modelo desarrollado con la librería de TensorFlow, y para configurar los puertos seriales que se usaran para ESPNOW y Node-RED (ver Figura 2.37).

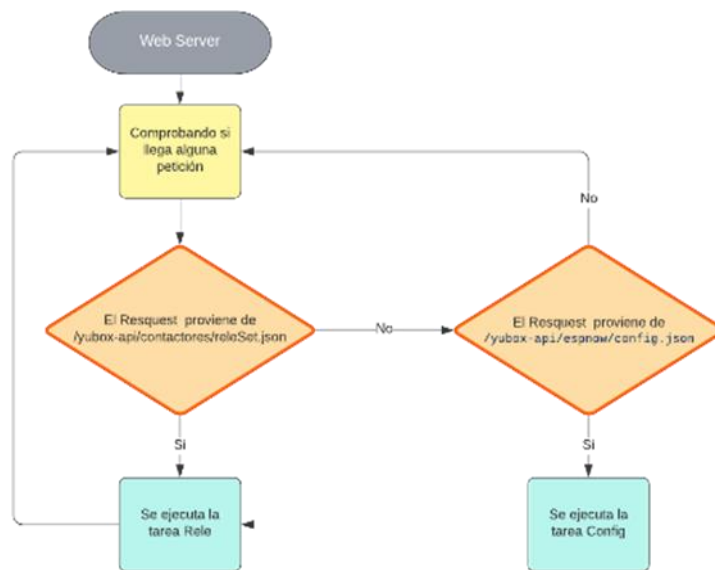
Compilado el modelo desarrollado, se procede a realizar la lectura de muestras y preprocesamiento de esos datos para inicializar el entrenamiento del modelo siempre que sea la fecha de entrenamiento establecida en la programación.

Configurado los puertos seriales, se procede a recibir la información (ver Anexo 8), donde si está es una muestra es enviada a Node-RED para su visualización en el web Server (ver Figura 2.38) y posteriormente son guardados en una base de datos en SQLite.



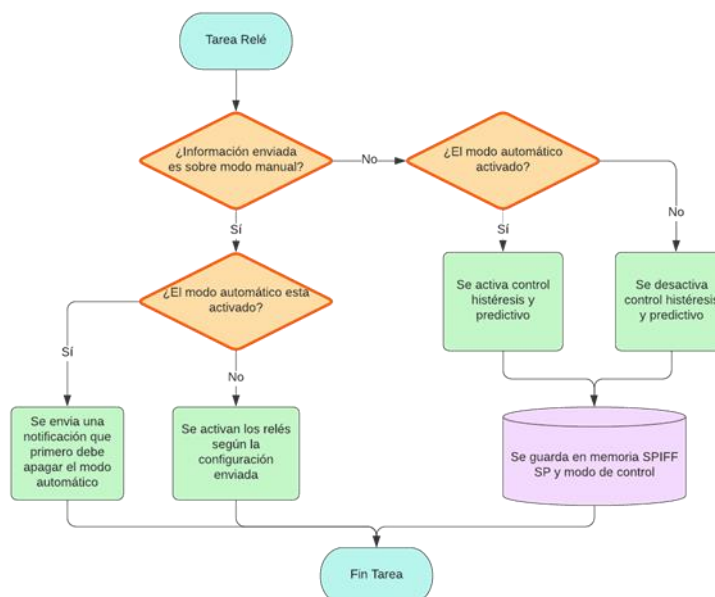
**Figura 2.37 Diagrama de flujo de las tareas del Raspberry del servidor principal.**

En la Figura 2.38, se presenta configuración del web server del ESP32 del módulo de monitoreo, donde al comprobar alguna petición proveniente en el control para los contactores activa la Tarea Relé (ver Figura 2.39), o proveniente del control predictivo del fp enviado por ESPNOW que ejecuta la configuración y control de los relés de acuerdo con la acción que lleguen a tomar estas tareas a partir del fp predicho.



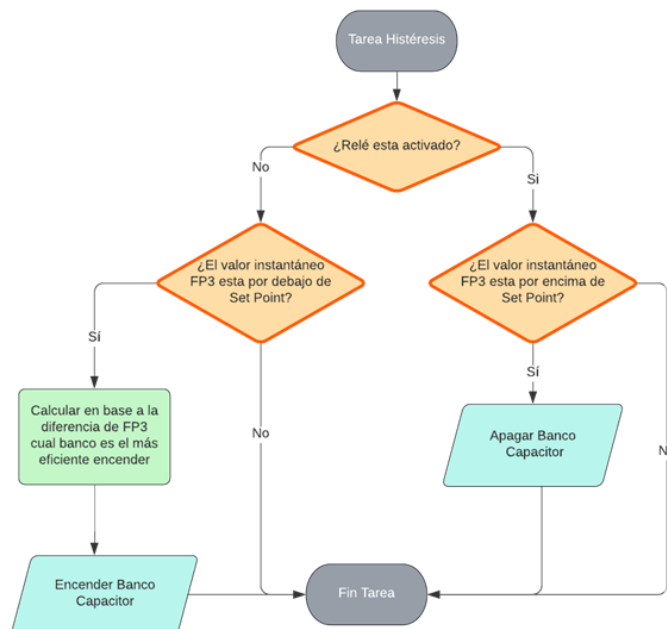
**Figura 2.38 Diagrama de flujo del proceso del Web Server.**

La Tarea Relé (ver Figura 2.39), consiste en la configuración manual o automático de los relés (o contactores) de cada banco de capacitores que se usarán para la corrección del fp de la energía suministrada. El modo manual activa los relés según lo que define el usuario, además de contar con el envío de una notificación si el modo automático está encendido durante la operación de esto. El modo automático activa los relés por control de histéresis y predictivo, y esto es guardado en una memoria SPIFFS.

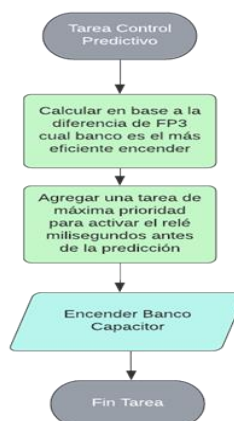


**Figura 2.39 Diagrama de flujo de la Tarea Relé definida.**

La tarea de control por Histéresis (ver Figura 2.40), se encarga de apagar el banco de capacitor (abrir el relé que lo conecta) cuando el valor de fp trifásico instantáneo está por arriba de los 0.95, y encender el banco de capacitor (cerrar el relé que lo conecta) adecuado cuando dicho parámetro está por debajo del valor mencionado. Este control es instantáneo, de no ser suficiente para el mejoramiento del fp, la tarea de control predictivo (ver Figura 2.41) se encarga de determinar el banco de capacitor más adecuado que mejore el fp predicho, el cual cuenta también con una tarea de máxima prioridad que activa el relé conectado a ese banco milisegundos antes del valor fp predicho (ver Anexo 9).



**Figura 2.40 Diagrama de flujo de la Tarea Histéresis definida**



**Figura 2.41 Diagrama de flujo de la Tarea Control Predictivo definida.**

# CAPÍTULO 3

## 3 RESULTADOS Y ANÁLISIS

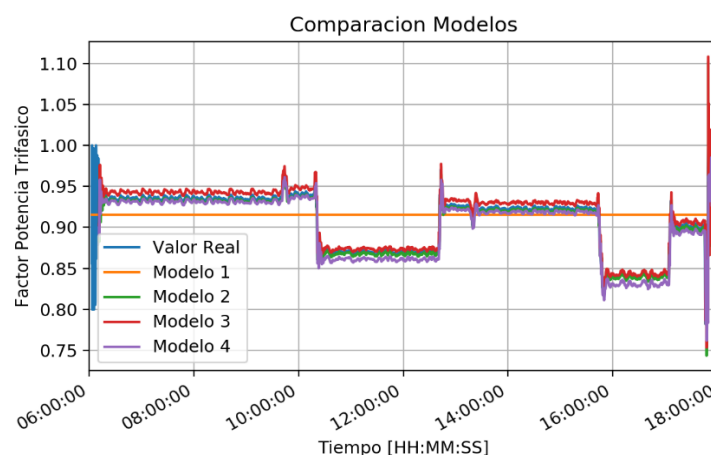
En este capítulo se procede a presentar el detalle de los resultados que se obtuvo del presente proyecto previo a la obtención del título. Los resultados se mostrarán de manera gráfica con respecto a la etapa de entrenamiento del modelo de predicción, calibración, proceso de validación, restricción.

### 3.1 Etapa del entrenamiento del modelo para la predicción de fp

En este apartado, se presenta los resultados obtenidos en la fase de entrenamiento del algoritmo predictivo desarrollado para el control del factor de potencia.

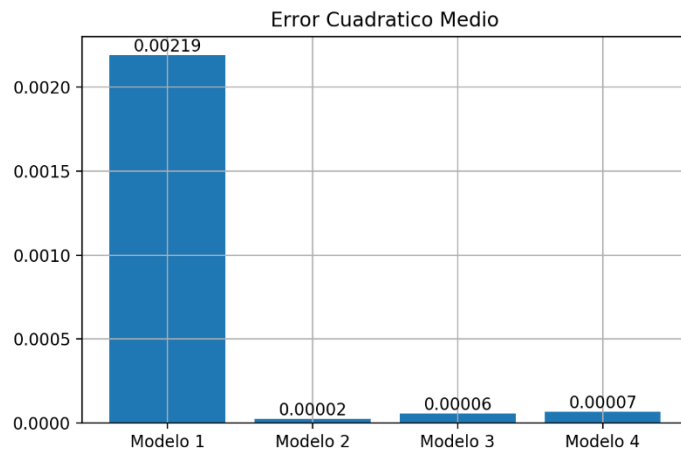
#### 3.1.1 Pruebas de entrenamiento del modelo

Una vez realizado el proceso de entrenamiento se graficó las predicciones realizadas en cada modelo, dándonos los siguientes resultados:



**Figura 3.1 Predicción del Fp trifásico con cada modelo propuesto.**

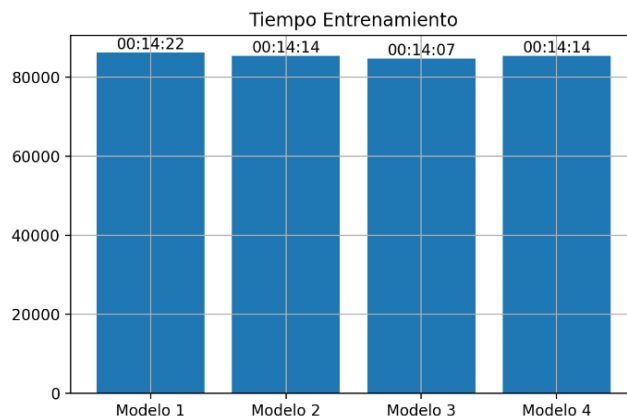
En la Figura 3.1, podemos observar la comparación de los modelos, arrojándonos que los modelos 2, 3 y 4 logran predecir el comportamiento de las gráficas, visualizándose como rayas casi iguales las unas de las otras, con sus respectivos errores. Con respecto al modelo 1, demuestra que el factor de potencia no puede ser determinado usando datos actuales y se necesita un apartado recurrente para tener mejores resultados.



**Figura 3.2 Valores RMSE de cada modelo propuesto.**

En la Figura 3.2 tenemos la comparación de los errores cuadráticos medios (RMSE), en donde nos arroja que el modelo 2 al final de los entrenamientos obtuvo el error más bajo, seguido del modelo 3 y 4. Estos valores muy bajos reafirma lo visto en la imagen anterior que los 3 modelos obtienen un comportamiento muy parecido al valor real.

Los tiempos de entrenamiento es sumamente importante, debido que nuestro algoritmo va a ejecutarse en una Raspberry, que a diferencia de Colab no tiene muchos recursos para el entrenamiento, lo que realizaría un tiempo mayor de espera. El modelo más rápido fue el número tres como puede verlo en la Figura 3.3. Lo que se puede deducir de este resultado es



**Figura 3.3 Tiempo de entrenamiento de cada modelo propuesto.**

A continuación, se adjunta una tabla con los parámetros más importantes obtenidos en la comparación de los modelos.

**Tabla 3.1 Parámetros importantes obtenidos en cada Modelo propuesto.**

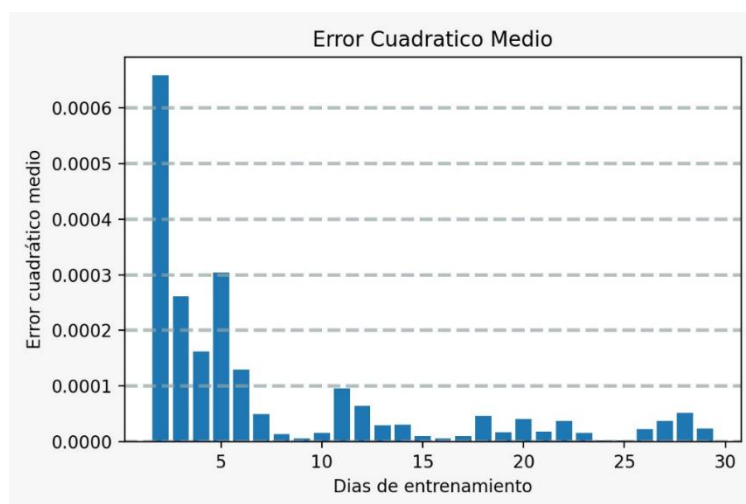
Modelos	Modelo 1	Modelo 2	Modelo 3	Modelo 4
Error Absoluto Máximo	0.115467	0.105654	0.103552	0.103652
Error Absoluto Mínimo	0.000198	0.00032	0.000153	0.000154
Error Cuadrático Medio	0.00219	0.000022	0.000057	0.000067

Con ella logramos determinar que los modelos 2,3,4 son aptos para la realización de la predicción, los 3 fueron capaces de predecir con un error muy bajo. Por otro lado, el modelo 1 no llegó a predecir y obtuvo el error más grande de los tres. Esto concluye que el factor de potencia trifásico tiene un patrón que no podremos obtener con una red ordinaria, además en caso de haber salido un mejor rendimiento se habría descartado por la complejidad para llegar a un buen resultado, se tuvo que realizar varias pruebas para encontrar unos parámetros que den un comportamiento deseado. Esto no paso con los demás modelos debido que solo se necesitó establecer unos parámetros en específico.

Por todo esto detallado se tomó la decisión de descartar por completo el modelo 1, el modelo que seleccionamos como final y utilizaremos para el resto del trabajo fue el modelo 2. Si bien se mencionó que los tres modelos restantes tenían un comportamiento parecido y los tres dieron buenos resultados la decisión del modelo dos se tomó por ser el modelo con el menor error cuadrático y al tiempo de entrenamiento. Aunque diferencia en el tiempo de ejecución entre los tres solo fue de segundos recordar que los modelos tres y cuatro fueron más complejos que el dos, en el tercero un sistema de dos recurrencias y el ultimo una capa concatenada. El modelo dos nos otorga menores tiempo menor tiempo de ejecución y menos recursos del dispositivo. A esto agregar que todos los modelos se entrenaron en Google Colab cuyos recursos son de una computadora de escritorio. La Raspberry posee muchas limitaciones provocando que los pocos segundos de diferencia se pueden traducir en minutos o hasta hora según los datos de entrada.

### 3.1.2 Calibración parámetros del modelo

En este apartado explicaremos a detalle el proceso para la calibración de parámetros del modelo, al seleccionar un modelo recursivo el tiempo de muestras es importantes debido que en base a esto el modelo determinara un valor futuro. Para la prueba se utilizó el modelo dos y se entrenó utilizando diferente número de entradas, en donde cada caso el último dato representaba el factor de potencia actual y los demás eran factores de potencia anteriores hasta un k valor. Se entreno con los valores de 10 a 100 entradas, con saltos de 10 unidades, es decir, 10, 20, 30, ..., 100. Para cada caso se calculó el error cuadrático medio obteniendo la siguiente grafica.



**Figura 3.4 Error cuadrático medio.**

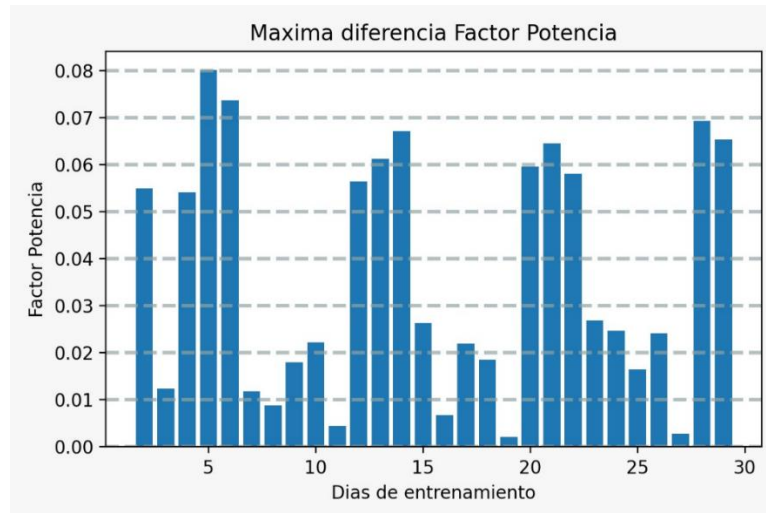
En donde se pudo apreciar que a medida que el mes avanza el módulo obtenido nos entrega un menor error cuadrático necesitando solo 11 días para ser totalmente eficiente en la predicción con un error menor al 0.0001. Además del error cuadrático medio se obtuvo el error máximo absoluto que podemos observar en la imagen.

### 3.1.3 Proceso de validación del modelo

Una vez establecido el modelo con su respectiva capa de entrada, se procedió a realizar diferentes pruebas para observar eficiencia de nuestro modelo. Para esta validación se realizó un entrenamiento en cada día del mes, en donde se utilizaron para el entrenamiento los datos obtenidos hasta el día actual de la prueba, para la verificación se utilizó el día siguiente a la prueba, de tal manera

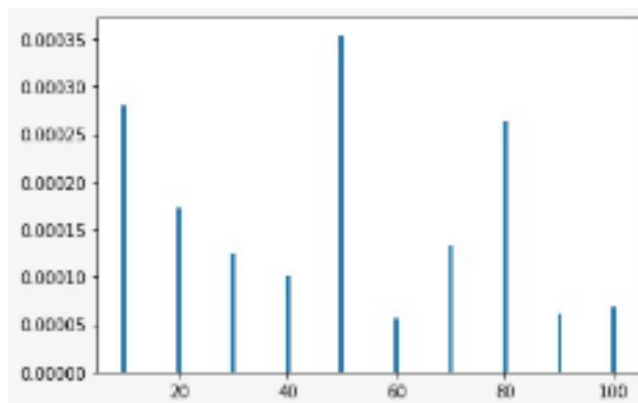


que cada día se obtuvo un error cuadrático medio el cual podemos observar en la Figura 3.4.



**Figura 3.5 Máxima diferencia de factor de potencia.**

Este fue calculado en base a obtener el error máximo durante el día, en el podemos visualizar como el máximo factor de potencia erróneo fue de 0.08, sin embargo, al tener la imagen de error cuadrático medio (ver Figura 3.4) nos queda claro que al inicio de la ejecución el error será alto, pero disminuirá a lo largo de mes.



**Figura 3.6 Error cuadrático medio vs cantidad de muestras previas.**

### 3.2 Etapa del entrenamiento del modelo en servidor principal Raspberry PI 4

Una vez definido el modelo y validado su funcionamiento se procedió a montarlo en nuestro servidor principal en la Raspberry. Durante este proceso se pudo contrastar el tiempo real de simulación. En donde el entrenamiento con una

muestra pequeña tardó alrededor de 9 minutos en entrenar tal como se puede verificar en la Figura 3.7

```

Model: "Model_3"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, 50)                10400
-----
dense (Dense)                (None, 1)                  51
-----
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
-----
Train on 13525 samples
 988/13525 [=>.....] - ETA: 9:19 - loss: 0.0024 - accuracy: 0.0000e+00

```

**Figura 3.7 Entrenamiento del modelo en la Raspberry a partir de una muestra pequeña.**

Una vez terminado el proceso de entrenamiento se procedió a realizar las predicciones del factor de potencia.

### 3.3 Transmisión datos medidor y servidor principal

En la figura 3.8, se visualiza los datos de medición que va recibiendo el raspberry del ESP32 enlazado con el ESP32 del módulo de monitoreo, esto durante cada tiempo de muestreo definido. Podemos observar que recibe los valores de voltaje, corriente y factor de potencia de cada línea de energía de la salida del inversor de la planta solar, además, de también recibir el factor de potencia trifásico.

```

Datos recibidos:
{'cmd': 'muestra', 'VA': 277.0584717, 'VB': 278.2703857, 'VC': 277.4612427, 'IA': 46.63391113, 'IB': 46.60368729, 'IC': 46.55472565, 'FPA': 0.721388996, 'FPB': 0.721696019, 'FPC': 0.721882999, 'FP3': 0.970537841, 'MAC': '3c:61:05:13:81:34'}

Datos recibidos:
{'cmd': 'muestra', 'VA': 277.0584717, 'VB': 278.2703857, 'VC': 277.4612427, 'IA': 46.63391113, 'IB': 46.60368729, 'IC': 46.55472565, 'FPA': 0.721388996, 'FPB': 0.721696019, 'FPC': 0.721882999, 'FP3': 0.970537841, 'MAC': '3c:61:05:13:81:34'}

Datos recibidos:
{'cmd': 'muestra', 'VA': 277.0584717, 'VB': 278.2703857, 'VC': 277.8547363, 'IA': 46.63391113, 'IB': 46.60368729, 'IC': 46.81370926, 'FPA': 0.721388996, 'FPB': 0.721696019, 'FPC': 0.721760988, 'FP3': 0.969451964, 'MAC': '3c:61:05:13:81:34'}

```

**Figura 3.8 Datos recibidos del ESP32 comunicado con ESPNOW al módulo de monitoreo**

Esos datos recibidos pasan a almacenarse en una base de datos en SQLite, que posteriormente mediante el preprocesamiento de datos, ya pueden ser usados para un nuevo entrenamiento del algoritmo del modelo seleccionado.

Una vez que se obtiene el modelo entrenado, se procede a realizar la ejecución del mismo para predecir el factor de potencia futuro dependiendo del tiempo de predicción definido en el algoritmo, para las predicciones mostradas en la Figura 3.9, se empleó un tiempo de predicción de 5 segundos al futuro.

```
[I][ESPnow.cpp:38] sentCallback(): Last Packet Sent to: 98:f4:ab:1b:1e:90
[I][ESPnow.cpp:39] sentCallback(): Last Packet Send Status: Delivery Success
Received message from: 98:f4:ab:1b:1e:90 - {'cmd': 'predict', 'valor': 0.9779326}

[I][ESPnow.cpp:38] sentCallback(): Last Packet Sent to: 98:f4:ab:1b:1e:90
[I][ESPnow.cpp:39] sentCallback(): Last Packet Send Status: Delivery Success
Received message from: 98:f4:ab:1b:1e:90 - {'cmd': 'predict', 'valor': 0.97756827}

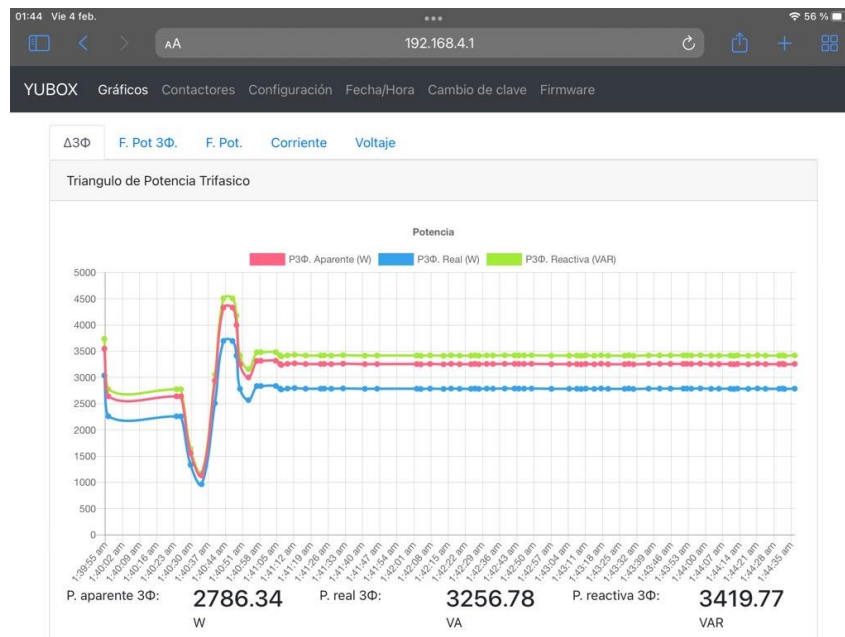
[I][ESPnow.cpp:38] sentCallback(): Last Packet Sent to: 98:f4:ab:1b:1e:90
[I][ESPnow.cpp:39] sentCallback(): Last Packet Send Status: Delivery Success
Received message from: 98:f4:ab:1b:1e:90 - {'cmd': 'predict', 'valor': 0.97731256}
```

**Figura 3.9 Envío de Predicciones de factor de potencia realizado servidor principal (Raspberry)**

### 3.4 Etapa de Monitoreo y control del factor de potencia

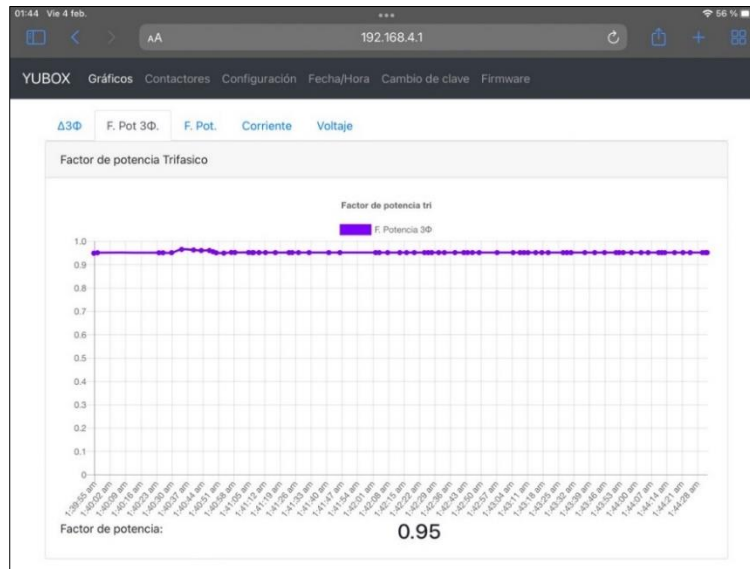
En este apartado se muestra los resultados obtenidos en el monitoreo y control de factor de potencia a partir de los datos obtenidos por el módulo de monitoreo y la ejecución del modelo entrenado en raspberry.

#### 3.4.1 Pruebas de monitoreo



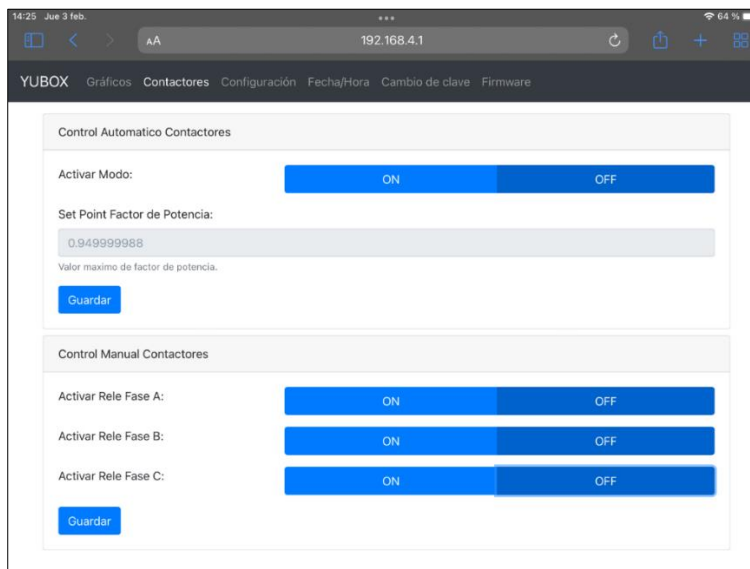
**Figura 3.10 Interfaz web para el monitoreo de potencias, factor de potencia, voltajes y corrientes.**

En la Figura 3.10, se visualiza las gráficas de potencias obtenidas con la planta solar simulada de un día particular, donde podemos también denotar que permite ver los valores instantáneos que entrega a la carga trifásica.



**Figura 3.11 Interfaz web monitoreando el FP trifásico.**

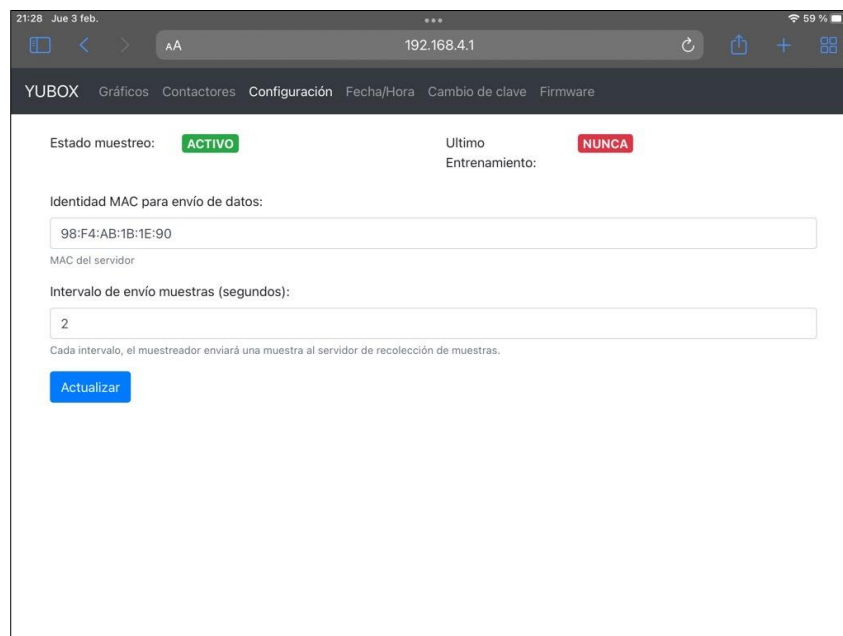
En la Figura 3.11, se visualiza la gráfica de factor de potencia trifásico de la energía que suministra la planta solar a una carga trifásica en un día particular.



**Figura 3.12 Interfaz web para el control manual o automático de los relés, y para configurar el Setpoint del fp.**

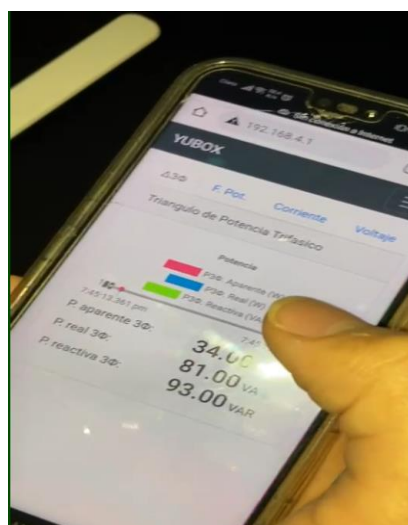
Tal como se puede observar en la Figura 3.12, la interfaz web permite configurar el modo de control de los relés, siendo está manual o automático,

dependiendo de la elección del usuario, donde podemos ver en el control automático se puede definir y guardar el Setpoint que se quiere en el factor de potencia.

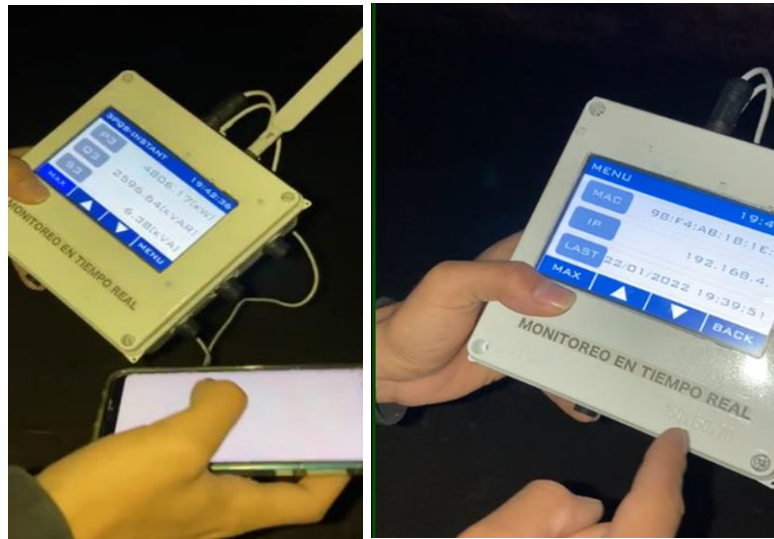


**Figura 3.13 Interfaz web para la configuración del tiempo de muestreo y dirección MAC del dispositivo a enviar los datos.**

En la Figura 3.13, se da la configuración del tiempo de muestreo de los datos a enviar al dispositivo MAC configurado su dirección para la respectiva comunicación del web server y ese dispositivo.



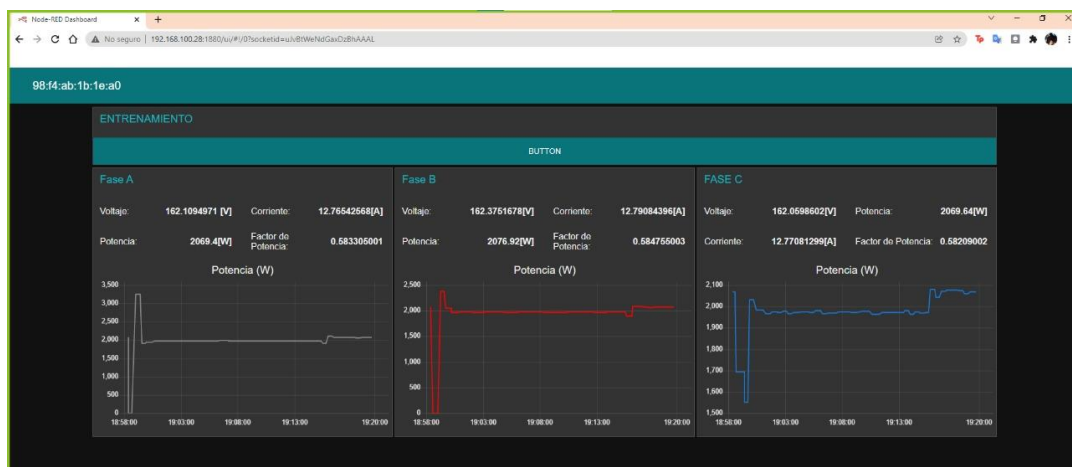
**Figura 3.14 Visualización de la Interfaz web mediante un móvil.**



**Figura 3.15 Visualización de los datos en el módulo de monitoreo y en un dispositivo móvil.**

En la Figura 3.14, podemos visualizar como mediante un móvil se puede monitorear la misma interfaz web diseñada mostrada en las figuras anteriores. Donde en la Figura 3.15, se procedió a comprobar la lectura de datos en el módulo de monitoreo y en lo que se observa en el móvil en ese mismo instante.

Además, se muestra en la Figura 3.16, que también mediante el uso de Node-RED, se realizó una interfaz web para el servidor principal donde podemos monitorear los datos eléctricos por fase que entrega la salida del inversor trifásico de la planta solar a la carga trifásica.



**Figura 3.16 Web server del servidor principal para el monitoreo de potencia, corriente, voltaje y factor de potencia por fase.**

Esta interfaz, permite en una misma interfaz web el visualizar las gráficas por fase de cada uno de los módulos de monitoreo, en caso de agregar más módulos en la comunicación de equipos con el servidor principal.

### 3.4.2 Interfaz gráfica medidor

Se presenta en las siguientes Figuras 3.17 al 3.19, imágenes referentes a la interfaz gráfica del módulo de monitoreo diseñado y llevado a cabo su implementación, como se puede observar nos presenta los valores instantáneos de potencia activa, aparente y factor de potencia de cada fase, teniendo también las opciones de visualizar la potencia reactiva, voltajes y corrientes que está entregando en esos instantes la fuente de energía a la carga.

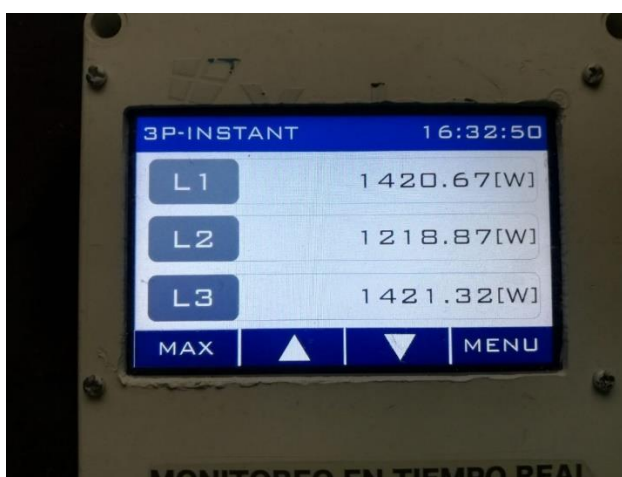


Figura 3.17 Monitoreo de la potencia activa instantánea de cada fase.



Figura 3.18 Monitoreo de la potencia aparente instantánea de cada fase.



Figura 3.19 Monitoreo del factor de potencia instantáneo de cada fase.

### 3.4.3 Envío de datos por comunicación inalámbrica con ESP-NOW

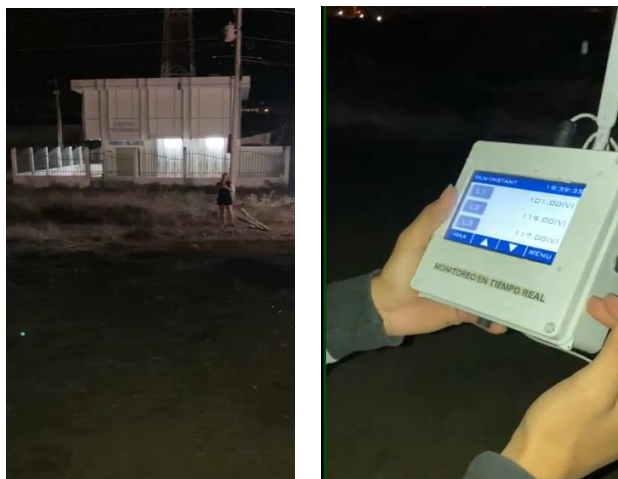
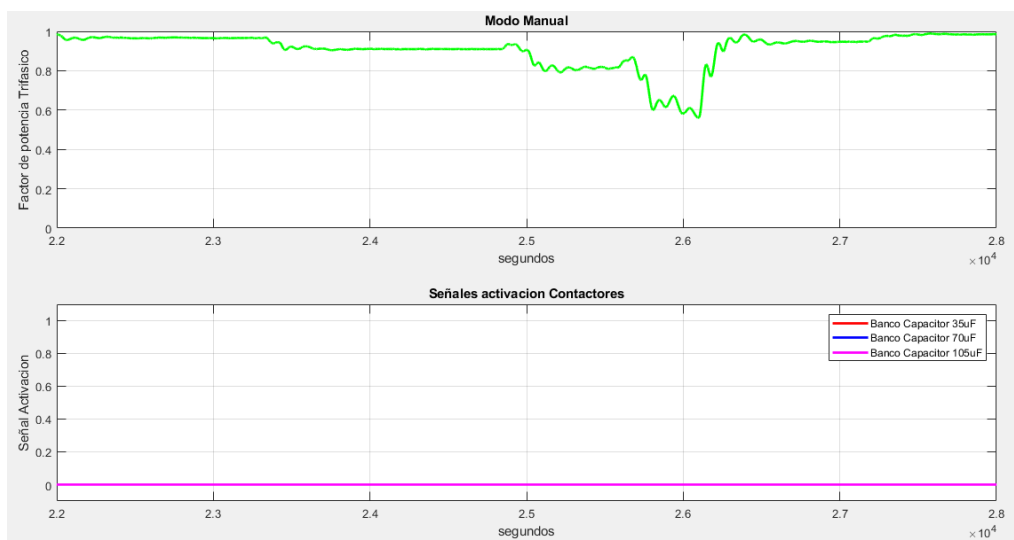


Figura 3.20 Envío de datos por ESP-NOW.

En la Figura 3.20, se presenta la respectiva prueba que se hizo para comunicar los dispositivos ESP32 del módulo de monitoreo y del servidor principal mediante ESP-NOW sin que estén conectados a una red WiFi, donde a una distancia de 10 metros, se pudo notar que no hubo problemas con la transmisión de los datos. Sin embargo, se destaca que hubo un poco de problemas en el envío de datos ya cuando la distancia era más de 10 metros, esto es debido al alcance de las antenas usadas en ambos equipos y posibles interferencias electromagnéticas que pudieron estar presente en el lugar donde se probó.

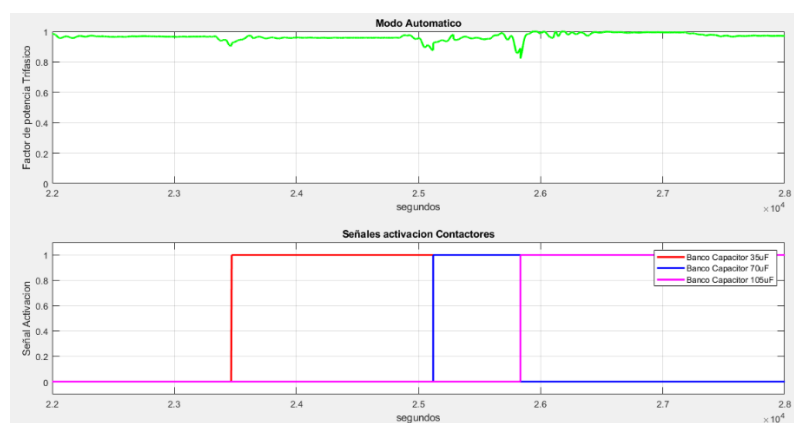


### 3.4.4 Simulación prototipo etapa mejoramiento factor potencia



**Figura 3.21 Control manual del factor de potencia de la planta.**

En la Figura 3.21, se presenta la simulación del factor de potencia de la energía suministrada de la planta simulada, en donde este control debe ser realizado de modo manual para la corrección de factor de potencia activando los bancos de capacitores con los relés, por lo que al no realizarse podemos visualizar como el factor de potencia es mucho menor a 0.95 en ciertos instantes, lo que solo es la medición de la planta sin emplear ningún control, esto indica que la energía que está suministrando la planta está fuera del rango establecidos en las regulaciones mencionadas anteriormente.



**Figura 3.22 Control automático del factor de potencia de la planta.**

En la Figura 3.22, ya se presenta el control de factor de potencia de la energía suministrada de la planta simulada, pero ya con el modo automático

encendido. En este modo, vemos ya como se da los cambios respectivos de la activación de bancos de capacitores para mejorar el factor de potencia a mayor o igual a 0.95, siendo así que mediante los bancos empleados vemos como se diferencia la señal de factor de potencia del modo manual en los tiempos de 2300 segundos a 2800 segundos en simulación pasada al tiempo real, con respecto a los que se visualiza en el factor de potencia con control automático, siendo este mismo control el desarrollado mediante el algoritmo de control predictivo que predice el valor de fp futuro y a base de eso se determina que valor de capacitancia es adecuado valor compensar la demanda de energía reactiva que tenga la carga a la fuente y así el factor incrementa a más de 0.95.

Es importante recalcar, que los bancos de capacitores dependiendo de los fp predichos se mantendrá cierto tiempo activos como se observa en las señales de activación de los contactores en la Figura 2.22.

### **3.4.5 Alcances y Restricciones**

#### **Alcance**

- Este prototipo puede implementarse en otros tipos de fuentes a parte de la solar, como puede ser la eólica, siempre y cuando la salida de voltajes y corrientes que suministren estas fuentes a las cargas que están conectadas no superen los valores mínimos y máximos de medición de los sensores del módulo de monitoreo.
- El módulo de monitoreo es con IP65, lo que implica que es resistente a condiciones de humedad y polvo.
- El sistema de prototipo con ESPNOW, permite que sea modular, es decir, se puede emplear más módulos de monitoreo en tiempo real, a cada inversor que disponga la planta, y así comunicar todos los equipos mediante la comunicación bidireccional que ofrece este protocolo.

#### **Restricciones**

- Dado a la pandemia, nos resultó complicado en ir al parque solar para realizar recolección de datos y pruebas del prototipo, por lo que para poder verificar el

modelo entrenado se hizo uso de datos obtenidos mediante una planta solar simulada en el software Simulink.

- Dado a los recursos computacionales la simulación de un día de operación en la planta, se la realizó en un tiempo de simulación de 2.4 segundos en Simulink, representando cada hora del tiempo real en 0.1 segundos en Simulink, ya que la ejecución de la planta solar con el uso del prototipo tardaba demasiado para colocar más tiempo de simulación.
- La distancia entre equipos a comunicar con el protocolo no debe ser muy extensa, dado a que a largas distancias ya empiezan a generarse problemas en el envío de datos en equipos ESP32.

### **3.5 Trabajos Futuros**

El proyecto se llevó a cabo mediante simulación, por lo que, para trabajos futuros, ponemos a criterio que se considere los siguientes puntos:

- Realizar las pruebas con el prototipo en una planta real, una vez que se obtenga una buena parte de datos recolectados en la medición de energía en la planta, y así ser usados para el entrenamiento del algoritmo.
- Probar la comunicación con otros protocolos de mayor alcance y transmisión de bits en caso de contar de una red local fija y estable el sitio donde estará el prototipo.
- De contar con mayor presupuesto, puede optarse por emplear un medidor de energía que registre la energía activa y reactiva que suministra la planta una vez empieza la operación el prototipo, y así poder comparar los datos almacenados en SQLite con los del medidor. De esta manera comprobar que el funcionamiento del modelo entrenado está trabajando correctamente o de requerirse un nuevo entrenamiento antes de la fecha establecida para el próximo.

# CAPÍTULO 4

## 4 CONCLUSIONES Y RECOMENDACIONES

En esta sección describirá la importancia del trabajo desarrollado, las fortalezas y las debilidades. Debe explicar lo que los resultados significan, la importancia respecto a otros trabajos realizados. Las implicaciones de este trabajo y los posibles trabajos futuros.

### 4.1 Conclusiones

- ❖ Los modelos propuestos tienen similar tiempo de entrenamiento alrededor de 14 minutos, sin embargo, el cambio que más se denota es en el error cuadrático medio donde el modelo 2 presenta el menor valor de este indicador de exactitud, el cual nos afirma que el comportamiento del fp predicho es lo más parecido a lo real.
- ❖ Luego de las pruebas a los modelos establecidos se pudo contrastar que para señales tipo factor de potencia que tienen un patrón en el tiempo, las redes recurrentes son la mejor opción gracias a las capas interconectadas a sí mismo, permitiendo guardar informaciones previas.
- ❖ No es suficiente un control de redes neuronales para el factor de potencia, debido que ante cambios bruscos el sistema tarda un poco en responder al no ser un patrón conocido para él, sin embargo, esto fue solucionado complementando la activación con un control por histéresis.
- ❖ La implementación de web server para la industria eléctrica es de suma importancia, permitiendo la generación de reportes gráficas y base de datos para toma de futuras decisiones, además al tener un control inalámbrico se salvaguarda la seguridad de los operadores al no tener que acercarse a los bancos de capacitores.

## 4.2 Recomendaciones

- ❖ De disponer una conexión local estable y segura para el funcionamiento de los equipos del prototipo se puede optar por otros protocolos de comunicación con mayor rapidez en la tasa de bits o velocidad de datos como LoraWAN y así conseguir mayor calidad en la transmisión en vivo de los datos entre dispositivos.
- ❖ El esquema de la planta solar simulada es simple, debido a la falta de recursos computacionales del ordenador, se recomienda en caso de que se disponga de mayores recursos computacionales, realizar un nuevo esquema alimentando varias cargas balanceadas y desbalanceadas, de manera de conseguir datos más reales que los datos a partir de solo cargas balanceadas.
- ❖ No utilizar la versión reciente de TensorFlow, aunque las nuevas versiones solucionan bugs y mejoran rendimientos, esta librería es bien conocida por traer problemas de compatibilidad o versiones específicas para otras librerías, por lo que se recomienda utilizar una versión anterior estable, que no tenga mucho bug incluidos.

# ANEXOS

## ETAPA ADQUISICION DATOS

### ANEXO 1

#### Definir Variables planta

```
% 1 Segundo -> 0.001
% 10 Segundos -> 0.01

%Equivale a un segundo
gain_time = 0.0001;

%Equivale a 10 segundos
ts_save = gain_time *10;

%Equivalente a 12Horas -> 6AM a 8PM
horas_simular = 14;

hora_inicio = 6*60*60*gain_time;
ocultarSol = 12*60*60*gain_time; %5PM

total_sim = (horas_simular*60*60)*gain_time

%% Iniciamos variables importantes

%Variables de envio
ts_send = 0.005
ASCII_send = '%f,%f,%f,%f,%f,%f,\n'

ts_read = 0.0055
ASCII_read = '%f|%f|%f\n'
```

## ANEXO 2

Bucle para simular una planta solar de 30 días, en cada día la carga se modificaba un poco

```
%% Datos Carga Trifasica
clc
data = [];

%weekday
weekday = 1;

for dia = 1:31
    day = dia;
    if (weekday>7)
        weekday = 1
    else
        weekday = weekday +1
    end

    %Seed Ruido Solar
    speed_solar = [round(rand()*100000)]
    %Luz solar
    duty_solar = (4*rand())-(3*rand()+90)
    %Datos de carga RL balanceada
    R=10;
    %Asumir que T=0 es las 6am
    time_hour = 60*60*gain_time;
    |
    if (weekday<6)
        T1= 0.8 *time_hour + rand()*time_hour ; % 8am
        T2= 3 *time_hour + rand()*time_hour; % 9am
        T3= 5 *time_hour + rand()*time_hour;% 11am
        T4= 7 * time_hour + rand()*time_hour ;% 1pm
        T5= 10 * time_hour + rand()*time_hour;% 4pm
        T6 = 10; % 6am del día siguiente

        L1=6.5e-03 + rand()*1e-04; %se activa de 0-T1
        L2=12e-03 + rand()*1e-04; %se activa de T1-T2
        L3=19e-03 + rand()*1e-04; %se activa de T2-T3
        L4=18e-03 + rand()*1e-04; %se activa de T3-T4
        L5=12e-03 + rand()*1e-04; %se activa de T4-T5
        L6=13e-03+ rand()*1e-04;%se activa de T5-T6

    else
        T1= 3*time_hour + rand()*time_hour; % 9 am
        T2= 4*time_hour + rand()*time_hour; % 10 am
        T3= 6*time_hour + rand()*time_hour; % 12pm
        T4= 9*time_hour + rand()*time_hour; % 3pm
        T5= 11*time_hour + rand()*time_hour; % 5pm
        T6= 11.5*time_hour + rand()*time_hour; % 5:30pm del día siguiente

        T1R = 10;
        T2R = 10;
        L1=10e-03+ rand()*1e-04; %se activa de 0-T1
        L2=9.5e-03+ rand()*1e-04; %se activa de T1-T2
        L3=15e-03+ rand()*1e-04; %se activa de T2-T3
        L4=11e-03+ rand()*1e-04; %se activa de T3-T4
        L5=17e-03+ rand()*1e-04; %se activa de T4-T5
        L6=13e-03+ rand()*1e-04;%se activa de T5-T6
    end

    msg = 'Iniciando Simulacion'
    sim('Planta_FTV_load_balanceada.slx')
    msg = 'Terminando Simulacion'

    data = [data;dataSet]

    t = 'Terminado guardado'
end
```

# ETAPA ELECCION MODELO RED NEURONAL A UTILIZAR

## ANEXO 3

Extracción, limpieza y preprocesamiento datos obtenidos de la planta en Matlab.

```
#Base datos Obtenida Matlab
csv_url = 'https://raw.githubusercontent.com/avbazurt/RedNeuronalTesis/Simulatic

#Creamos dataframe con estos Datos
df = pd.read_csv(csv_url, error_bad_lines=False)

df

#

data = df[df['FP3'].notna()]

#Limpiamos los datos corrompidos
data["FP3"][data["FP3"]<0.8] = 0.8

#Por cuestiones de simulacion se definio 0->6AM, debemos agregar dicho valor
data["second_hora"][:] = data["second_hora"][:] + 6*60*60

data["weekday"][:] = data["weekday"][:] - 1

#Extraemos dos dataframe uno utilizaremos para todos los entrenamientos
data_train = data[data["day"]<5]
data_test = data[data["day"]==5]

data_json = {
    "modelos": ["Modelo 1", "Modelo 2", "Modelo 3", "Modelo 4"],
    "tiempo": [0, 0, 0, 0],
    "mean_squared_error": [0, 0, 0, 0],
    "error_max": [0, 0, 0, 0],
    "error_min": [0, 0, 0, 0]
}

#Creo el Json a utilizar mas adelante
with open('data.json', 'w') as file:
    json.dump(data_json, file)

data
```



## ANEXO 4

**Función Utilizada para dar formato específico para utilizarlos en el entrenamiento del modelo.**

```
#Preparar datos
def DataProcesorModel2(_datos,timeSteps):
    #Entrenamiento
    fp3_train = list(_datos["FP3"][:])
    second_hora = list(_datos["second_hora"][:])

    num_train = len(fp3_train)

    xTrain = []
    yTrain = []
    second = []

    for i in range(0, num_train-timeSteps):
        #Lista - [FP3-actual]
        fp3 = fp3_train[i:i+timeSteps]

        train = list(fp3)
        xTrain.append(train)

        yTrain.append(fp3_train[i+timeSteps])
        second.append(second_hora[i+timeSteps])

    xTrain, yTrain = np.array(xTrain), np.array(yTrain)

    xTrain = np.reshape(xTrain, (xTrain.shape[0], xTrain.shape[1], 1))

    return [xTrain,yTrain,second]
```

## ANEXO 5

**Creación del modelo recurrente a utilizar en nuestra solución y su respectivo entrenamiento**

```
model_2 = Sequential(name="Model_2") # Inicializa el modelo

timeSteps = 50

#Capa Entrada
model_2.add(layers.Input(shape=(timeSteps,1), name="Capa_Entrada"))

#Capa Oculta 1
model_2.add(LSTM(units=50, name="Capa_Oculta"))

#Capa Salida
model_2.add(Dense(units=1,name="Capa_Salida"))

""" compilar modelo """
model_2.compile(optimizer='rmsprop',
                loss='mse') # mse = mean_squared_error

model_2.summary()
```

Layer (type)	Output Shape	Param #
Capa_Oculta (LSTM)	(None, 50)	10400
Capa_Salida (Dense)	(None, 1)	51

Total params: 10,451  
Trainable params: 10,451  
Non-trainable params: 0

```
trian_model_2 = DataProcesorModel2(data_train,timeSteps)
model_2.fit(trian_model_2[0], trian_model_2[1], epochs=1, batch_size=1, verbose=True)
```

## ANEXO 6

### Comparación de datos reales con las predicciones otorgadas por el modelo

```
#Test Propuesta
test_model_2 = DataProcesorModel2(data_test,timeSteps)

second_hora = test_model_2[2]
valor_predict = model_2.predict(test_model_2[0])
valor_real = test_model_2[1]

plt.title("Validacion Modelo 2")
GraficarDatos(second_hora,[valor_real,valor_predict])
plt.legend(["Valor Real","Predicción"])
plt.ylabel("Factor Potencia Trifasico")
plt.show()

#Calculamos el error_mean_squared

print("Error Cuadratico Medio")
print(mean_squared_error(valor_real,valor_predict))

#Calculo Error
error = np.abs(np.subtract(valor_real,valor_predict[:,0]))
plt.title("Error Absoluto Modelo 2")
GraficarDatos(second_hora,[error])
plt.show()

#Calculamos error maximo y minimo
print("Maximo Error Detectado")
print(max(error))

print("Minimo Error Detectado")
print(min(error))

#Calculamos el error_mean_squared
print("Error Cuadratico Medio")
print(mean_squared_error(valor_real,valor_predict))
```

## ETAPA PREDICCIÓN Y MEJORAMIENTO FACTOR POTENCIA

## ANEXO 7

### Envió datos por protocolo ESP-NOW

```
void espnowMuestrearFases(void)
{
#ifdef Test_Yubox
  sensor->GetMedicionTrifasica();
  yuboxMuestrearFases();
#endif

  DynamicJsonDocument json_data(JSON_OBJECT_SIZE(14));
  json_data["cmd"] = "muestra";
  json_data["VA"] = sensor->DatosFaseA.VLN;
  json_data["VB"] = sensor->DatosFaseB.VLN;
  json_data["VC"] = sensor->DatosFaseC.VLN;
  json_data["IA"] = sensor->DatosFaseA.I;
  json_data["IB"] = sensor->DatosFaseB.I;
  json_data["IC"] = sensor->DatosFaseC.I;
  json_data["FPA"] = sensor->DatosFaseA.FP;
  json_data["FPB"] = sensor->DatosFaseB.FP;
  json_data["FPC"] = sensor->DatosFaseC.FP;
  json_data["FP3"] = sensor->TriFase.fp3;

  String json_output;
  serializeJson(json_data, json_output);

  status_muestreo = now->sendData(MAC_servidor, json_output);
}
```

## ANEXO 8

### Recepción datos en Raspberry Pi

```
try:
    #Convertir String en Dic
    dic = json.loads(texto)
    print("Datos recibidos:")
    print(dic)
    if (dic["cmd"] == "muestra"):

        #Enviar datos a Node Red
        NodeRed.WritePort(texto)

        #Verificar mac correcta
        if (dic["MAC"]=="c8:2b:96:a7:e3:e0"):

            #Realizar prediccion
            predic = RedNeuronal.Prediccion(float(dic["FP3"]))

            #Enviar respuesta ESPNOW
            parametros = lambda mensaje: str({
                "cmd": "now",
                "MAC": "C8:2B:96:A7:E3:E0",
                "msg": str(mensaje)
            })

            mensaje = {"cmd":"predict","valor":predic}
            #print(dic)
            #print(mensaje)
            #print("")
            ESP32.WritePort(parametros(mensaje))
            print(mensaje)

            print("")
else:
    print("No concuerda")
```

## ANEXO 9

### Control Relés con predicción enviada desde Raspberry Pi

```
// Cuando rele esta activado

if (on_rele_A or on_rele_B or on_rele_C)
{
    if (fp3 > set_point_fp + 0.25)
    {
        Serial.println("Reset Rele");
        on_rele_A = false;
        on_rele_B = false;
        on_rele_C = false;

        // Desactivo banderas
        bool bandera_A = false;
        bool bandera_B = false;
        bool bandera_C = false;
    }
}
```

```
if ((fp3 >= set_point_fp and !mejor_fp))
{
  Serial.println("Listo para otro capacitor");
  mejor_fp = true;
}
```

```
else if (fp3 < set_point_fp and mejor_fp)
{
  if (delta_fp > 0)
  {
    if (bandera_C)
    {
      log_i("Fp muy bajo");
    }
    else if (bandera_B)
    {
      // Activamos el rele C
      on_rele_A = false;
      on_rele_B = false;
      on_rele_C = true;

      Serial.println("Rele C");
      bandera_C = true;
      mejor_fp = false;
    }

    else if (bandera_A)
    {
      // Activamos el rele B
      on_rele_A = false;
      on_rele_B = true;
      on_rele_C = false;
      Serial.println("Rele B");

      // Bandera
      bandera_B = true;
      mejor_fp = false;
    }
    else
    {
      on_rele_A = true;
      on_rele_B = false;
      on_rele_C = false;
    }
  }
}
```

## Anexo 10

Link del Repositorio del proyecto: <https://github.com/avbazurt/RedNeuronalTesis>

# BIBLIOGRAFÍA

- [1] Revista Energía. (2012). Energías Renovables No Convencionales. [En línea]. Available: <https://www.revistaenergia.com/1135/>. [Último acceso: 12 11 2021].
- [2] López Merodio, I. (2019). La energía renovable: Importancia de su implantación y desarrollo.
- [3] Marzola Loaiza, M. C. (2017). Fuentes de energía renovables no convencionales.
- [4] Proaño Fuentes, J. F. (2020). Análisis comparativo de los códigos eléctricos de energía solar fotovoltaica de Chile, Puerto Rico y Ecuador (Bachelor's thesis).
- [5] ARCOTEL. (2018). Estadística Anual y Multianual del Sector Eléctrico Ecuatoriano. [En línea]. Available: <https://www.regulacioneolica.gob.ec/wp-content/uploads/downloads/2019/08/Estad%C3%ADsticaAnualMultianual2018.pdf>. [Último acceso: 01 11 2021].
- [6] GeoEnergíaEc. (2020). Retos y oportunidades de las energías renovables en el Ecuador. [En línea]. Available: <https://www.geoenergia.gob.ec/retos-y-oportunidades-de-las-energias-renovables-no-convencionales-en-ecuador/>. [Último acceso: 13 11 2021].
- [7] ARCONEL. (2015). Regulación No. ARCONEL-004/15. [En línea]. Available: <https://www.regulacioneolica.gob.ec/wp-content/uploads/downloads/2015/12/Regulacion-No.-ARCONEL-004-15.pdf>. [Último acceso: 15 11 2021].
- [8] ARCONEL. (2016). Resolución No. ARCONEL-071/15. [En línea]. Available: <https://www.regulacioneolica.gob.ec/wp-content/uploads/downloads/2016/07/071-15-Requerimientos-t%C3%A9cnicos-generadores-renovables-no-convencionales.pdf>. [Último acceso: 15 11 2021].
- [9] CONELEC. (2016). Resolución No. CONELEC-004/02. [En línea]. Available: <https://www.regulacioneolica.gob.ec/wp-content/uploads/downloads/2016/02/Regulacion-No.-CONELEC-004-02.pdf>. [Último acceso: 15 11 2021].

- [10] CENACE. (2021). INFORME ANUAL 2020. [En línea]. Available: <http://www.cenace.gob.ec/wp-content/uploads/downloads/2021/04/Informe-Anual-CENACE-2020-Parte-1.pdf>. [Último acceso: 12 11 2021].
- [11] Novoa Guamán, E. G. (2015). Manual de operación y mantenimiento preventivo y correctivo para Parques fotovoltaicos en el Ecuador (Bachelor's thesis, Quito, EPN, 2015.).
- [12] Pillajo Amagua, T. O. (2019). Análisis del estado actual de la implementación de energías renovables en el Ecuador. Propuesta de una guía de uso eficiente de la energía eléctrica.
- [13] IRENA. (2020). Global Renewables Outlook: Energy transformation 2050. [En línea]. Available: <https://irena.org/publications/2020/Apr/Global-Renewables-Outlook-2020>. [Último acceso: 30 10 2021].
- [14] ARCONEL. (2015). Estadística Anual y Multianual del Sector Eléctrico Ecuatoriano. [En línea]. Available: <https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2016/11/Estad%C3%ADsticaAnualyMultianual2015.pdf>. [Último acceso: 30 10 2021].
- [15] Cordero, F. (2018). Parques solares: análisis descriptivo y constructivo de energías renovables no convencionales.
- [16] EfenAR Solar. (s.f.). Huertos Solares. [En línea]. Available: <http://www.efenarsolar.com/index.php/es/huertos-y-parques-solares>. [Último acceso: 30 10 2021].
- [17] ARCONEL. (2019). Estadística Anual y Multianual del Sector Eléctrico Ecuatoriano. [En línea]. Available: [https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2020/06/Est\\_2019\\_Borrador\\_08-06-2020\\_1606.pdf](https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2020/06/Est_2019_Borrador_08-06-2020_1606.pdf). [Último acceso: 30 10 2021].
- [18] Nord, J. H., Koohang, A., & Paliszkievicz, J. (2019). The Internet of Things: Review and theoretical framework. *Expert Systems with Applications*, 133, 97-108.
- [19] Ben-Daya, M., Hassini, E., & Bahroun, Z. (2019). Internet of things and supply chain management: a literature review. *International Journal of Production Research*, 57(15-16), 4719-4742.

- [20] Cisco. The Internet of Things Reference Model. [En línea]. Available: [http://cdn.iotwf.com/resources/71/IoT\\_Reference\\_Model\\_White\\_Paper\\_June\\_4\\_2014.pdf](http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf). [Último acceso: 05 11 2021].
- [21] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, 14(11), 4724-4734.
- [22] Mahesh, B. (2020). Machine Learning Algorithms-A Review. *International Journal of Science and Research (IJSR)*. [Internet], 9, 381-386.
- [23] Rajesh Khadka. (2017). Machine Learning Types. [En línea]. Available: <https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1>. [Último acceso: 05 11 2021].
- [24] Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- [25] Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (Vol. 1142, No. 1, p. 012012). IOP Publishing.
- [26] Tanha, J., van Someren, M., & Afsarmanesh, H. (2017). Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8(1), 355-370.
- [27] Pérez-Ortiz, J. A. (2002). Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto.
- [28] Salas, R. (2004). *Redes neuronales artificiales*. Universidad de Valparaíso. Departamento de Computación, 1, 1-7.
- [29] Núñez Sánchez-Agustino, F. J. (2016). Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional.
- [30] Pasic, R., Kuzmanov, I., & Atanasovski, K. (2021). ESP-NOW communication protocol with ESP32. *Journal of Universal Excellence*, 6(1), 53-60.
- [31] Random Nerd Tutorials. (s.f.). Getting Started with ESP-NOW (ESP32 with Arduino IDE). [En línea]. Available: <https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>. [Último acceso: 20 01 2022].

- [32] Barros Saldaña, W. (1981). Corrección del factor de potencia en sistemas industriales (Bachelor's thesis, QUITO/EPN/1981).
- [33] Raspberry. (s.f.). Raspberry Pi 4 model B. [En línea]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. [Último acceso: 01 11 2021].
- [34] Jacob Beningo. (2020). Cómo seleccionar y usar el módulo ESP32 con Wi-Fi/Bluetooth adecuado para una aplicación de IoT industrial. [En línea]: Available: <https://www.digikey.com/es/articles/how-to-select-and-use-the-right-esp32-wi-fi-bluetooth-module>. [Último acceso: 05 11 2021].
- [35] Espressif Systems. (2021). ESP32 ESP-IDF Programming Guide. [En línea]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/esp-idf-en-v5.0-dev-346-g7af719e-esp32.pdf>. [Último acceso: 14 11 2021].
- [36] EMIS. (2020). ENERGYFUTUR ENERGÍAS RENOVABLES ECUADOR S.A. [En línea]. Available: [https://www.emis.com/php/company-profile/EC/Energyfutur\\_Energias\\_Renovables\\_Ecuador\\_SA\\_es\\_5471294.html](https://www.emis.com/php/company-profile/EC/Energyfutur_Energias_Renovables_Ecuador_SA_es_5471294.html). [Último acceso: 01 12 2021].
- [37] Educaedu. (s.f.). ENERGYFUTUR Guayaquil Información. [En línea]. Available: <https://www.educaedu.com.ec/centros/energyfutur-uni2275>. [Último acceso: 01 12 2021].
- [38] Francisco Vidal. (2021). EnergyFutur, Energías Renovables Ecuador S.A. [En línea]. Available: <https://ec.linkedin.com/in/fvidalp>. [Último acceso: 02 12 2021].
- [39] EnergyFutur. (s.f.). Información de EnergyFutur. [En línea]. Available: <http://www.energyfutur.com/>. [Último acceso: 02 12 2021].
- [40] EMIS. (2020). BRINEFORCORP S.A. (ECUADOR). [En línea]. Available: [https://www.emis.com/php/company-profile/EC/Brineforcorp\\_SA\\_es\\_3982432.html](https://www.emis.com/php/company-profile/EC/Brineforcorp_SA_es_3982432.html). [Último acceso: 03 12 2021].
- [41] Salvador Molina. (2014). Reportaje de planta fotovoltaica Brineforcorp-Cabal Energy ubicada en Manabí. [En línea]. Available:



[http://www.webdom.es/projectfiles/Enenergyfutur\\_Ecuador.pdf](http://www.webdom.es/projectfiles/Enenergyfutur_Ecuador.pdf). [Último acceso: 03 12 2021].

[42] SolarElectric. (2020). Paneles solares Yingli YL250P-29b. [En línea]. Available: <https://www.solarelectricsupply.com/yingli-yl250p-29b-solar-panels>. [Último acceso: 03 12 2021].

[43] SolarDesignTool (s.f.). Características Yingli Solar YL250P-29b. [En línea]. Available: <http://www.solardesigntool.com/components/module-panel-solar/Yingli-Solar/2147/YL250P-29b/specification-data-sheet.html>. [Último acceso: 03 12 2021].

[44] Peacefair. (s.f.). PZEM-004T V3.0 User Manual. [En línea]. Available: <https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf>. [Último acceso: 05 12 2021].

[45] ETC2. (s.f.). ZMPT101B(ZMPT107) voltage transformer operating guide (Datasheet). [En línea]. Available: <https://html.alldatasheet.com/html-pdf/1131993/ETC2/ZMPT101B/112/1/ZMPT101B.html>. [Último acceso: 05 12 2021].

[46] Allegro MicroSystems. (s.f.). ACS712 Datasheet. [En línea]. Available: <https://html.alldatasheet.com/html-pdf/168326/ALLEGRO/ACS712/293/1/ACS712.html>. [Último acceso: 05 12 2021].

[47] Amazon. (2021). Precio de PZEM-004t módulo de energía Voltaje de Corriente multímetro AC 80 – 260 V 100 A con transformador de corriente. [En línea]. Available: <https://amazon.com/-/es/pzem-004t-Corriente-mult%C3%ADmetro-transformador-corriente/dp/B074QGWMKH>. [Último acceso: 05 12 2021].

[48] Peacefair. (s.f.). PZEM-014/016 AC communication module. [En línea]. Available: <https://images-na.ssl-images-amazon.com/images/I/81GtkIOyZaL.pdf>. [Último acceso: 05 12 2021].

[49] Nextion. (2021). Introducción a Nextion. [En línea]. Available: <https://nextion.tech/>. [Último acceso: 05 12 2021].

- [50] Nextion. (2021). Nextion NX3224T028\_011. [En línea]. Available: <https://nextion.tech/>. [https://nextion.ca/datasheets/nx3224t028\\_011/](https://nextion.ca/datasheets/nx3224t028_011/). [Último acceso: 05 12 2021].
- [51] Tapia Ayala, C. H., & Manzano Yupa, H. M. (2013). Evaluacion de la plataforma Arduino e implementacion de un sistema de control de posicion horizontal (Bachelor's thesis).
- [52] Charly. (2016). Introducción a python. [En línea]. Available: <https://www.prometec.net/rpi-introduccion-python/>. [Último acceso: 05 12 2021].
- [53] Conde Ortiz, D. (2018). Inteligencia artificial con TensorFlow para predicción de comportamientos.
- [54] Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *IEEE Access*, 8, 201071-201086.
- [55] Vicente Lahoz. (2019). El protocolo MQTT. [En línea]. Available: <https://www.securityartwork.es/2019/02/01/el-protocolo-mqtt-impacto-en-espana/>. [Último acceso: 27 11 2021].
- [56] Indelmar. (2017). ¿QUÉ ES MQTT? [En línea]. Available: <https://www.indelmar.com/?p=1140>. [Último acceso: 06 12 2021].
- [57] Dignani, J. P. (2012). Análisis del protocolo ZigBee (Doctoral dissertation, Universidad Nacional de La Plata).
- [58] Google. (s.f.). Google Colab. [En línea]. Available: <https://colab.research.google.com/>. [Último acceso: 30 01 2021].