

Universidad Tecnológica Nacional

Proyecto Final

**Trazador de curvas Corriente-Tensión para
evaluación de eficiencia energética con
conexión remota.**

Autores:

- *Cagnani, Joaquín Jesús*
- *Weiss, Augusto Rubén*

Director:

- *Gareis, Gabriel*

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero en Electrónica*

en la

Facultad Regional Paraná

Diciembre de 2021

Declaración de autoría:

Nosotros declaramos que el Proyecto Final “Trazador de curvas Corriente-Tensión para evaluación de eficiencia energética con conectividad remota.” y el trabajo realizado son propios.

Declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero en electrónica en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-
-

Fecha:

Agradecimientos:

En primer lugar, quisiéramos agradecer al Ingeniero Gabriel Gareis, quien nos ha acompañado a lo largo de este proyecto, brindándonos su tiempo y asesoramiento, sus opiniones y por sobre todo nos ha transmitido su conocimiento en el ámbito de las energías renovables, todo ello desde su generosidad.

Seguidamente, agradecemos a la institución Universidad Tecnológica Nacional Facultad Regional Paraná y a quienes la dirigen y administran y sobre todo a quienes enseñan en ella, incluyendo también al personal del laboratorio de electrónica general y del laboratorio de energías alternativas, quienes además nos han facilitado los instrumentos necesarios para llevar a cabo este proyecto final de carrera.

Por último, quisiéramos agradecer a nuestras familias: padres, hermanos, esposa y novia e hijo de cada uno de nosotros, a nuestros amigos y demás personas que han hecho posible también el transcurrir de nuestro paso por la institución, acompañándonos no solo en lo económico sino también en lo emocional.

Cagnani, Joaquín Jesús
Weiss, Augusto Rubén

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica/Civil/Electromecánica

**Trazador de curvas Corriente-Tensión para
evaluación de eficiencia energética con
conexión remota**

Cagnani, Joaquín Jesús

Weiss, Augusto Rubén

Abstract:

In order to provide to the scientific community a tool for studying the variation of energy efficiency due to a particular condition exerted on photovoltaic modules, we have developed a Current-Voltage curve tracer capable of carrying out the corresponding measurements, storing them and sending them to a server under MQTT protocol to make them available to the user for its remote usage and analysis.

It was implemented on a motherboard that contains different modules necessary for optimal operation. Some of them are the CPU (ESP32 microcontroller) which manages the general behavior of the unit and its connectivity, the MPPT modules that act as a electronic load and together with analog-digital converters perform the acquisition of the curves of

the photovoltaic modules, and temperature and solar radiation sensors for more information on the conditions of each curve obtained.

As a result, a very useful tool was obtained for the evaluation and analysis of the energy efficiency behavior of photovoltaic modules under conditions of interest and determined by the user.

Keywords:

Photovoltaic modules - Solar radiation - MQTT protocol - AP service - STA service

Resumen:

Con el fin de brindar a la comunidad científica una herramienta para el estudio de la variación de la eficiencia energética que ejerce una condición en particular sobre módulos fotovoltaicos, es que hemos desarrollado un trazador de curvas Corriente-Tensión capaz de realizar las mediciones correspondientes, almacenarlas y enviarlas a un servidor bajo protocolo MQTT para disponibilizarlas al usuario de forma remota y proceder al análisis de las mismas.

El mismo se implementó sobre una placa madre que contiene los diferentes módulos necesarios para un óptimo funcionamiento, entre los cuales podemos mencionar el CPU (microcontrolador ESP32) quien gestiona el comportamiento general de la unidad y la conectividad del mismo, los módulos MPPT que actúan como una carga electrónica y que junto a conversores analógico-digital realizan la adquisición de las curvas de los módulos fotovoltaicos, y sensores de temperatura y radiación solar para mayor información de las condiciones de cada curva obtenida.

Como resultado se obtuvo una herramienta de gran utilidad para la evaluación y el análisis del comportamiento de la eficacia energética de módulos fotovoltaicos ante condiciones de interés y determinadas por el usuario.

Palabras Clave:

Módulo fotovoltaico - Radiación solar - Protocolo MQTT - Servicio AP - Servicio STA

Índice:

Capítulo 1: Introducción	1
Capítulo 2: Desarrollo	6
2.1: Desarrollo del sistema.....	8
2.1.1: CPU, conversores ADC y periféricos.....	8
2.1.1.A: Investigación de componentes.....	8
2.1.1.B: Circuitos propuestos y su funcionamiento.....	14
2.1.1.C: Desarrollo del software.....	22
2.1.2: Módulo MPPT.....	23
2.1.2.A: Investigación de componentes.....	24
2.1.2.B: Circuito propuesto.....	25
2.1.2.C: Análisis de funcionamiento.....	26
2.1.3: Sensores de temperatura y radiación PAR.....	27
2.1.3.A: Investigación de componentes.....	30
2.1.3.B: Circuito propuesto y su funcionamiento.....	31
2.1.4: Placa base.....	34
2.1.4.A: Investigación.....	34
2.1.4.B: Ejecución del proyecto.....	35
2.1.4.C: Análisis de funcionamiento.....	38
2.1.5: Fuente de alimentación.....	39
2.1.5.A: Investigación de componentes.....	39
2.1.5.B: Circuito propuesto.....	43
2.1.5.C: Análisis de funcionamiento.....	45
2.1.6: Dispositivo en modo AP, módulo web y modos de funcionamiento Online y Sleep.....	46
2.1.6.A: Funcionamiento en modo online.....	47
2.1.6.B: Configuración del dispositivo.....	48
2.1.6.C: Testeo de variables y sensado de panel solar.....	49
2.1.7: Dispositivo en modo STA y broker MQTT.....	50
2.1.8: Servidor web y WebApp.....	57
2.1.8.A: Investigación de componentes.....	57
2.1.5.B: Análisis de funcionamiento.....	58
2.1.5.C: Desarrollo del software.....	67
2.2: Diseño final.....	78

Capítulo 3: Resultados	84
Capítulo 4: Análisis de costos	86
4.1: Costos de materiales y desarrollo.....	86
4.2: Plan de venta y comercialización.....	88
Capítulo 5: Discusión y conclusión	89
Capítulo 6: Literatura citada	90

Lista de Figuras:

Figura 1. <i>Detalle de consumo eléctrico per cápita de los últimos 50 años</i>	1
Figura 2. <i>Curvas Corriente-Tensión y Potencia-Tensión característicos de un módulo fotovoltaico</i>	4
Figura 3. <i>Efecto de la temperatura de trabajo sobre un módulo fotovoltaico</i>	5
Figura 4. <i>Diagrama general del sistema desarrollado</i>	6
Figura 5. <i>Diagrama en bloque del dispositivo</i>	7
Figura 6. <i>Diagrama en bloques de la fuente de alimentación</i>	8
Figura 7. <i>Microcontrolador NodeMCU ESP32</i>	9
Figura 8. <i>Estructura interna del microcontrolador ESP32</i>	10
Figura 9. <i>Diagrama de pin-out del microcontrolador ESP 32</i>	10
Figura 10. <i>Módulo ADC: ADS1115</i>	11
Figura 11. <i>Módulo RTC</i>	12
Figura 12. <i>Módulo micro SD</i>	13
Figura 13. <i>Display OLED</i>	13
Figura 14. <i>Convertor 3v3 - 5v</i>	14
Figura 15. <i>Esquema eléctrico pulsadores e indicadores LED</i>	15
Figura 16. <i>Conexión ESP32 - Micro SD</i>	16
Figura 17. <i>Ejemplo archivos guardados en micro SD</i>	16
Figura 18. <i>Ejemplo archivo .csv generado por el microcontrolador</i>	17
Figura 19. <i>Ejemplo de archivo logs del microcontrolador</i>	18
Figura 20. <i>Esquema eléctrico bus de datos I2C</i>	18
Figura 21. <i>Esquema eléctrico pin alarma por RTC</i>	19
Figura 22. <i>Mensaje encendido display OLED</i>	20
Figura 23. <i>Mensajes de inicio display OLED</i>	20
Figura 24. <i>Mensajes modo de funcionamiento en display OLED</i>	20
Figura 25. <i>Muestra de hora en display OLED</i>	20
Figura 26. <i>Mensajes “ejecutando barrido” en display OLED</i>	21
Figura 27. <i>Progressbar display OLED</i>	21
Figura 28. <i>Esquema eléctrico circuito desconexión de alimentación de módulos</i>	22
Figura 29. <i>Logica del CPU - Parte 1</i>	22
Figura 30. <i>Logica del CPU - Parte 2</i>	23
Figura 31. <i>MosFET utilizado como carga electrónica</i>	24
Figura 32. <i>Amplificador operacional</i>	24
Figura 33. <i>Esquema eléctrico módulo MPPT</i>	25

Figura 34. <i>Diseño PCB módulo MPPT</i>	25
Figura 35. <i>Módulo MPPT</i>	26
Figura 36. <i>Rampa diente de sierra proveniente del DAC</i>	26
Figura 37. <i>Curva I-V defectuosa</i>	27
Figura 38. <i>Esquema eléctrico módulo MPPT corregido</i>	28
Figura 39. <i>Curva I-V obtenida con módulo MPPT final</i>	28
Figura 40. <i>Comparación rampas de sierra del generador - Rshunt</i>	29
Figura 41. <i>Sensores de temperatura DS18B20</i>	29
Figura 42. <i>Sensor de radiación solar PAR</i>	30
Figura 43. <i>Esquema eléctrico bus de datos One-Wire</i>	31
Figura 44. <i>Error del sensor de temperatura</i>	32
Figura 45. <i>Placa base desarrollada</i>	34
Figura 46. <i>Diseño de la placa base - vista superior</i>	35
Figura 47. <i>Diseño de la placa base - vista inferior</i>	36
Figura 48. <i>Diseño de la placa base - vista 3d</i>	37
Figura 49. <i>Diseño de la placa base - formato gerber</i>	37
Figura 50. <i>Placa base en implementación</i>	38
Figura 51. <i>Modulo step-down LM2576HV</i>	39
Figura 52. <i>Baterías de litio utilizadas</i>	40
Figura 53. <i>Módulo BMS cargador de baterías</i>	41
Figura 54. <i>Módulo step-down LM2596</i>	41
Figura 55. <i>Módulo step-up MT3608</i>	41
Figura 56. <i>Esquema de la fuente de alimentación</i>	43
Figura 57. <i>Cargador de baterías de litio</i>	44
Figura 58. <i>Fuente de alimentación implementada</i>	45
Figura 59. <i>Diseño de discriminador de fuente de alimentación</i>	46
Figura 60. <i>Dispositivo móvil conectado a red WiFi del ESP32</i>	47
Figura 61. <i>Mensaje de conexión establecida</i>	48
Figura 62. <i>Aplicación web levantada por ESP32</i>	49
Figura 63. <i>Sección de test por aplicación web</i>	50
Figura 64. <i>Test realizado desde aplicación web</i>	50
Figura 65. <i>Sección Escanear por aplicación web</i>	51
Figura 66. <i>Lógica sitio web en modo local</i>	51
Figura 67. <i>Diagrama general de conexión remota</i>	52
Figura 68. <i>Estructura de tópicos MQTT</i>	53

Figura 69. <i>Interacción Front-end y Back-end</i>	53
Figura 70. <i>Pantalla de Login</i>	59
Figura 71. <i>Pantalla para registrarse como usuario</i>	59
Figura 72. <i>Mensaje de error de login</i>	60
Figura 73. <i>Dashboard personal vacío</i>	61
Figura 74. <i>Pantalla añadir dispositivo</i>	61
Figura 75. <i>Dashboard e información</i>	62
Figura 76. <i>Iconos de acciones sobre un dispositivo</i>	62
Figura 77. <i>Pantalla de visualización de curvas</i>	63
Figura 78. <i>Tile de radiación solar</i>	63
Figura 79. <i>Tile de temperatura ambiente</i>	64
Figura 80. <i>Tiles de temperaturas de trabajo de paneles solares</i>	64
Figura 81. <i>Tile de temperatura interna del dispositivo</i>	64
Figura 82. <i>Curvas en Web App</i>	65
Figura 83. <i>Selector de curvas</i>	65
Figura 84. <i>Indicador fecha y hora para gráfica de curvas</i>	65
Figura 85. <i>Pantalla para configurar un dispositivo de forma remota</i>	66
Figura 86. <i>Información de preconfiguración de dispositivo</i>	66
Figura 87. <i>Actualización de RTC de forma remota</i>	67
Figura 88. <i>Diagrama general de desarrollo de Front-End en Angular</i>	67
Figura 89. <i>Renderización de diferentes componentes</i>	71
Figura 90. <i>Estructura de comunicación Front-end y Back-end</i>	74
Figura 91. <i>Diagrama en bloques de Back-end</i>	75
Figura 92. <i>Diseño final - destapado</i>	78
Figura 93. <i>Diseño final - lateral inferior</i>	79
Figura 94. <i>Alargue MC4 para panel solar</i>	79
Figura 95. <i>Montaje del equipo completo</i>	80
Figura 96. <i>Montaje del equipo completo</i>	81
Figura 97. <i>Montaje del equipo completo</i>	82
Figura 98. <i>Comparación curvas Corriente-Tensión</i>	84
Figura 99. <i>Comparación curvas Potencia-Tensión</i>	85

Lista de Tablas

Tabla 1. Características del microcontrolador NodeMCU SP32.....	9
Tabla 2. Características módulo ADC ADS1115.....	11
Tabla 3. Resistencia según distancia sensor - microcontrolador.....	32
Tabla 4. Resolución sensor de temperatura.....	33
Tabla 5. Estructura tópicos MQTT de suscripción.....	54
Tabla 6. Estructura tópicos MQTT de publicación.....	55
Tabla 7. Modelo users en base de datos.....	68
Tabla 8. Modelo <i>devices</i> en base de datos.....	68
Tabla 9. Modelo <i>curve</i> en base de datos.....	69
Tabla 10. Modelo <i>point</i> en base de datos.....	69
Tabla 11. Análisis de costo de materiales.....	86
Tabla 12. Análisis de costo total.....	87

Lista de Abreviaciones y Símbolos

MQTT: MQ telemetric transport

MPPT: Maximum power point tracking

AP: access point

STA: station

ADC: analog to digital converter

DAC: digital to analog converter

RTC: real time clock

I-V: Corriente-Tensión

P-V: Potencia-Tensión

V: Voltaje

I: Amper

AO: amplificador operacional

Voc: Tensión de circuito abierto

Isc: Corriente de corto circuito

DC: Corriente directa (o continua)

MVF: Modulo fotovoltaico

API: Application programming interface

SN: serial number

Dedicado a:

Nuestras familias: padres, hermanos, esposa y novia e hijo de cada uno de nosotros, a nuestros amigos y personas que nos han acompañado en el proceso.

Capítulo 1: Introducción

Hoy día vivimos un mundo donde la energía eléctrica es un recurso de gran valor, valioso y necesario para nuestro vivir cotidiano. Podemos decir que se hace casi imposible pensar en un mundo en el cual vivimos sin ella.

Esto nos ha llevado a grandes consumos energéticos y a distintas formas de generación de la misma, algunas más invasivas y otras más limpias.

En la siguiente gráfica podemos ver como en los últimos 50 años el consumo eléctrico per cápita se ha casi triplicado, pasando de ser 1.2 KWh per cápita en el año 1970 a 3.13 KWh per cápita al año 2015.

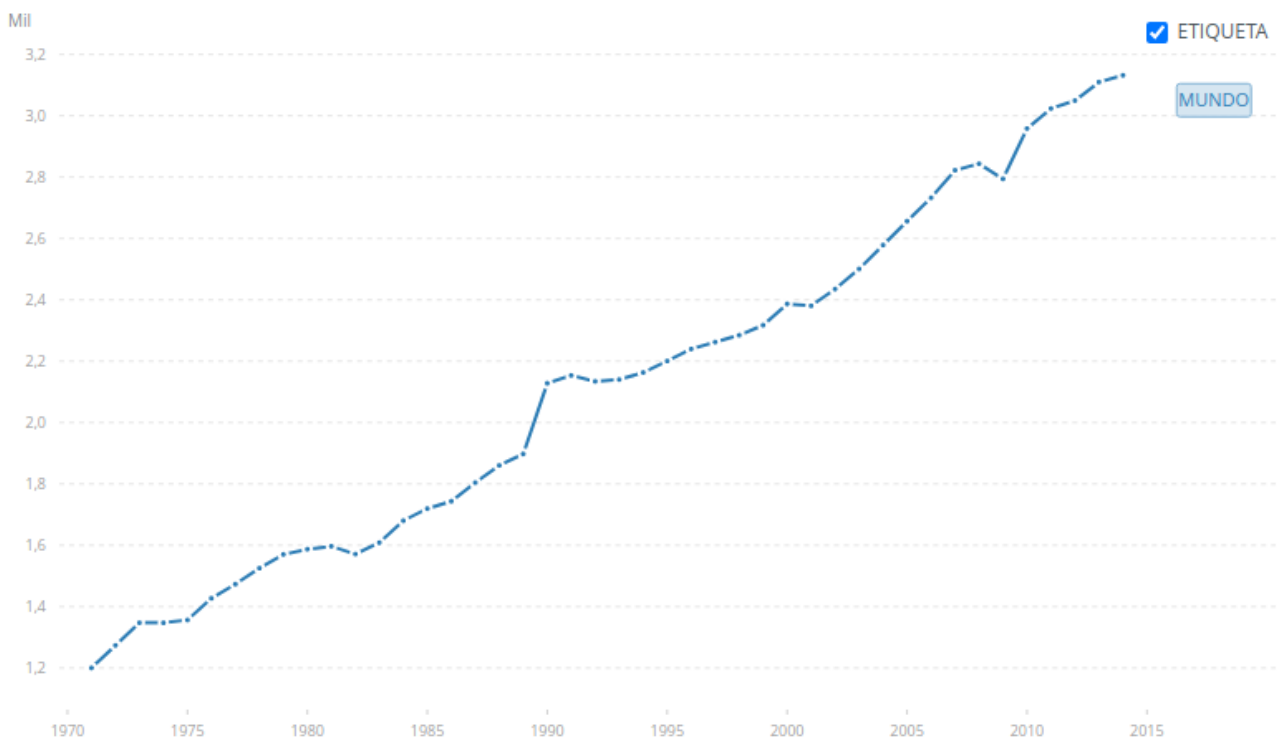


Figura 1. Detalle de consumo eléctrico per cápita de los últimos 50 años. [1]

Es así, como hace ya un tiempo, pero tomando gran auge en este último siglo, se enfatiza la necesidad de obtener energía de formas alternativas, más conocidas como renovables o energías verdes, y con ella la necesidad de incentivar la investigación en campos relacionados con ello.

El proyecto desarrollado, surge de la necesidad de disponer de un sistema que permita analizar el comportamiento de la eficiencia en generación de energía de módulos fotovoltaicos, comparando datos previos y/o datos del fabricante con ensayos continuos en un determinado lapso de tiempo.

Este trabajo está destinado principalmente al ámbito de investigación, y pretende dar a la comunidad científica una herramienta para el estudio del comportamiento y rendimiento energético de los módulos fotovoltaicos en condiciones impuestas por quien lo desee. Concretamente hoy desde el laboratorio de energías alternativas de nuestra facultad regional Paraná, se desea analizar la variación de la eficiencia de los módulos fotovoltaicos debido a la suciedad y la diferencia en el mantenimiento que los módulos fotovoltaicos reciben. Otros aspectos por estudiar pueden ser, por ejemplo: sombras, rejillas anti palomas, micro-rupturas, etc. Quedando completamente a disposición de la creatividad de los investigadores.

El mercado de hoy día no ofrece un equipo similar al desarrollado. Si bien podemos encontrar trazadores de curvas Corriente-Tensión para módulos o plantas fotovoltaicas, estos se destinan más a un análisis de funcionamiento y no a un estudio científico.

Es por eso que cabe destacar una de las características que diferencia a este desarrollo, la conectividad remota a través de protocolo MQTT y la capacidad de almacenar los datos en una base de datos sobre un servidor web para luego ser accedido desde cualquier navegador y desde cualquier ubicación, dándonos la posibilidad de estudiar un sistema fotovoltaico instalado en una estancia y poder estudiarlo desde un laboratorio en la universidad.

Algunos ejemplos comercializados hoy día son:

- **HT I-V500 w Curve Tracer:** I-V500w permite la detección en campo de la curva I-V, uno de los parámetros característicos principales tanto de un solo módulo como de cadenas de módulos para instalaciones fotovoltaicas hasta un máximo de 1500V y 10A o 1000V y 15A. Para medir la curva IV, I-V500w administra una base de datos interna.
- **Solmetric PVA-1000S:** PVA-1000S es un trazador de curvas I-V de 1000 voltios con modelado de rendimiento fotovoltaico incorporado e irradiación inalámbrica avanzada, temperatura y detección de inclinación. Proporciona un rendimiento y una precisión de medición sin precedentes y proporciona una visión profunda del rendimiento de sus arreglos.
- **Seaward PV200:** El PV200 es un trazador de curvas I-V compacto y rentable que utiliza una simple operación de botón pulsador, lo que lo convierte en un comprobador eficiente y versátil para sistemas fotovoltaicos. Se pueden almacenar 999 conjuntos de datos en el instrumento y, una vez descargados en el software SolarCert, se pueden realizar comparaciones con las condiciones de prueba estándar.

Es adecuado introducir en este punto diferentes conceptos de un módulo fotovoltaico y sus características principales y funcionales.

Los materiales y dispositivos fotovoltaicos convierten la luz solar en energía eléctrica, y las células fotovoltaicas se conocen comúnmente como células solares. Las células

fotovoltaicas son dispositivos de producción de electricidad fabricados con materiales semiconductores. La energía fotovoltaica puede traducirse literalmente como luz eléctrica.

Cuando la luz brilla sobre una célula fotovoltaica, puede reflejarse, absorberse o pasar a través del mismo. Pero sólo la luz absorbida genera electricidad. La energía de la luz absorbida se transfiere a los electrones en los átomos del material semiconductor de la célula fotovoltaica. Una propiedad eléctrica especial de la célula fotovoltaica llamada un campo eléctrico incorporado proporciona la fuerza, o voltaje, necesaria para conducir la corriente a través de una carga externa.

Un módulo fotovoltaico, no es más que un arreglo en serie y paralelos de células fotovoltaicas, y lo que define y caracteriza a un módulo fotovoltaico son sus parámetros eléctricos principales:

- Punto de máxima potencia.
- Corriente de corto circuito
- Tensión de circuito abierto
- Punto de operación óptimo en Corriente
- Punto de operación óptimo en Tensión

Existen otros parámetros que se pueden encontrar en una hoja de datos de un módulo fotovoltaico, pero hemos citado aquí los de mayor interés para el desarrollo de este proyecto.

Son estos parámetros los que definen las curvas que caracterizan a un módulo fotovoltaico, que ahora presentamos:

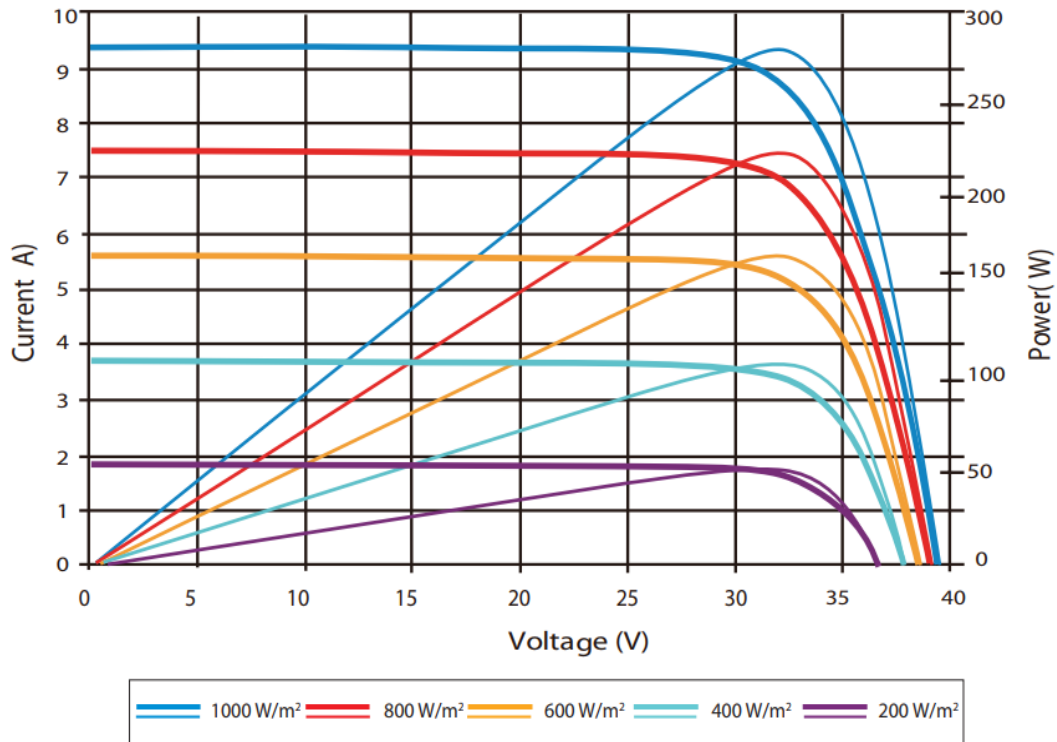


Figura 2. Curvas Corriente-Tensión y Potencia-Tensión características de un módulo fotovoltaico.[2]

En la imagen vemos diferentes curvas que se corresponden a diferentes puntos de Corriente-Tensión que entrega el módulo fotovoltaico, cada una de ellas correspondientes a una radiación solar diferente. Podemos deducir ya, la importancia que tiene la radiación solar al momento de generar energía eléctrica por medio de un módulo fotovoltaico. En la misma gráfica también pueden verse curvas de Potencia-Tensión las cuales definen un punto óptimo de funcionamiento o de mayor eficiencia.

Parámetros importantes y a tener en cuenta a la hora de analizar y estudiar el comportamiento de los módulos solares son la radiación solar que como vimos anteriormente a mayor radiación solar mayor potencia de salida tendrá el módulo fotovoltaico, pero no podemos dejar de lado la temperatura de trabajo del módulo en cuestión, ya que tienen gran influencia sobre la eficiencia a la hora de generar energía eléctrica.

A continuación, podemos observar su influencia sobre curvas características de un módulo fotovoltaico:

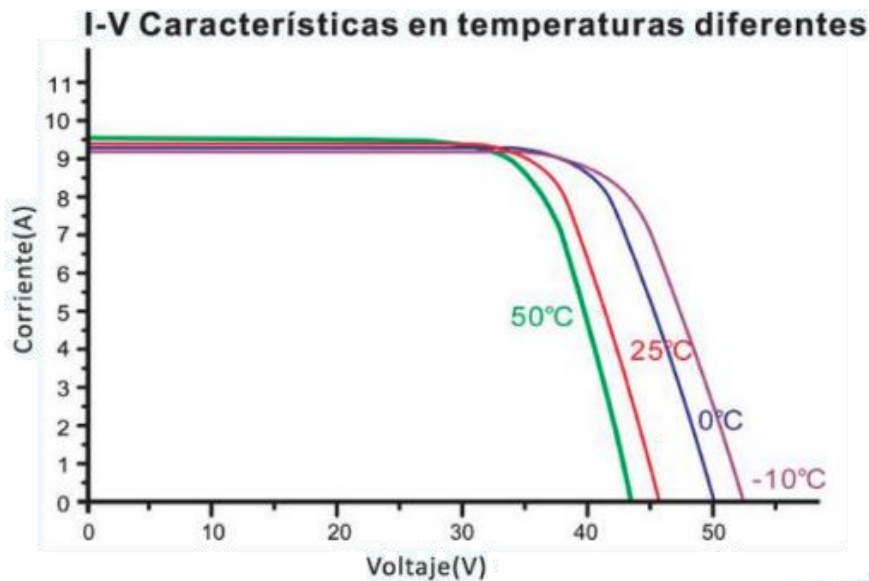


Figura 3. Efecto de la temperatura de trabajo sobre un módulo fotovoltaico.[3]

Como podemos observar en la imagen anterior, la temperatura de trabajo del módulo fotovoltaico tiene influencia sobre el nivel de tensión de salida en el módulo fotovoltaico, de manera que modifique los puntos correspondientes a Potencia-Tensión, esto deduce que a menor temperatura de trabajo mayor potencia de salida del módulo solar y vice versa.

Concluimos por lo tanto que tanto la radiación solar como la temperatura de trabajo en un módulo fotovoltaico son de gran importancia a la hora de interpretar una curva obtenida del mismo, es por ello que ambas se tienen en cuenta en el sistema desarrollado junto con la temperatura ambiente de donde sea instalado el mismo.

Capítulo 2: Desarrollo

El desarrollo completo del sistema se puede dividir en diferentes partes, dentro de ellas podemos mencionar al dispositivo principal que realiza entre sus tareas más importantes el proceso de medición sobre los módulos fotovoltaicos y también la presentación de datos y comunicación con el usuario llamado “Equipo MPPT”, al servicio Web que se utiliza para la configuración del equipo en modo local, al broker MQTT y servidor Web que proporciona la funcionalidad de monitoreo y configuración remota del dispositivo.

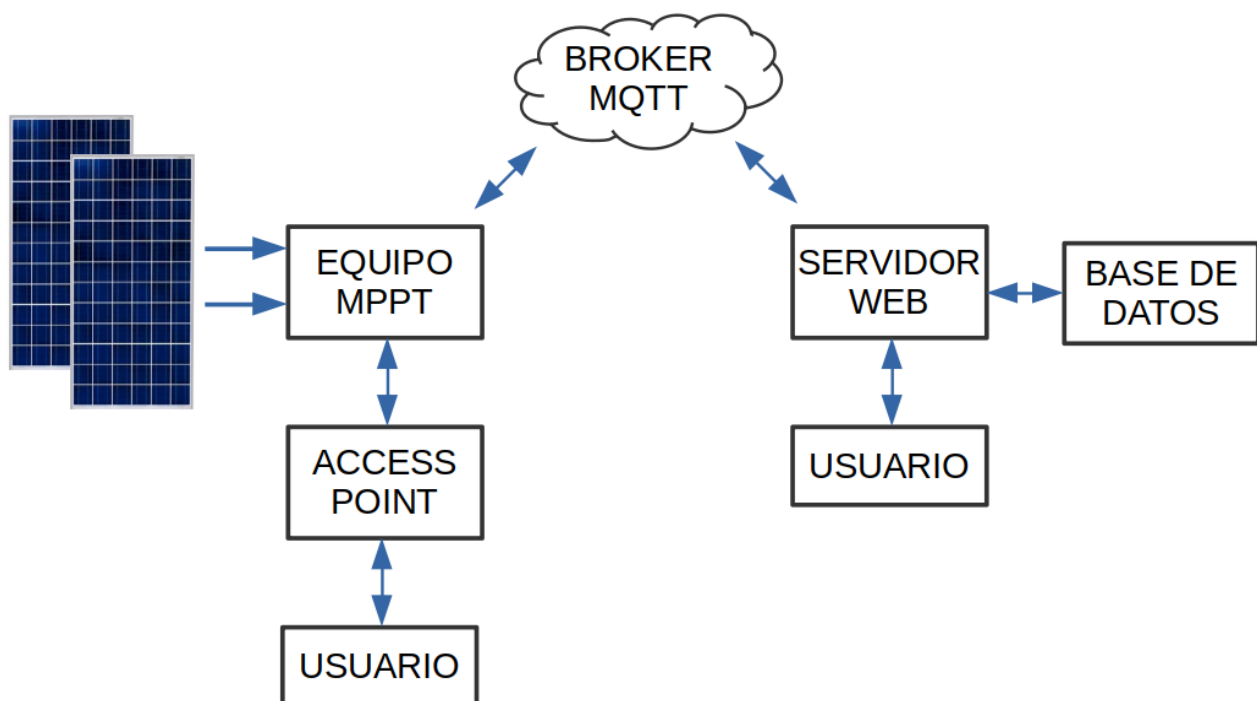


Figura 4. Diagrama general del sistema desarrollado.

El sistema completo necesita para su funcionamiento un dispositivo central que está compuesto por un módulo de conexión inalámbrica a internet, un punto de acceso inalámbrico local, un reloj interno para constatar fecha y hora, una pantalla para visualización, acceso a una tarjeta SD, sensores de temperatura, sensor de radiación, una carga electrónica encargada de evaluar el panel fotovoltaico y una fuente de alimentación dedicada para este dispositivo. A continuación, se detalla un diagrama en bloques para visualizar de cómo se encuentran relacionadas las distintas partes.

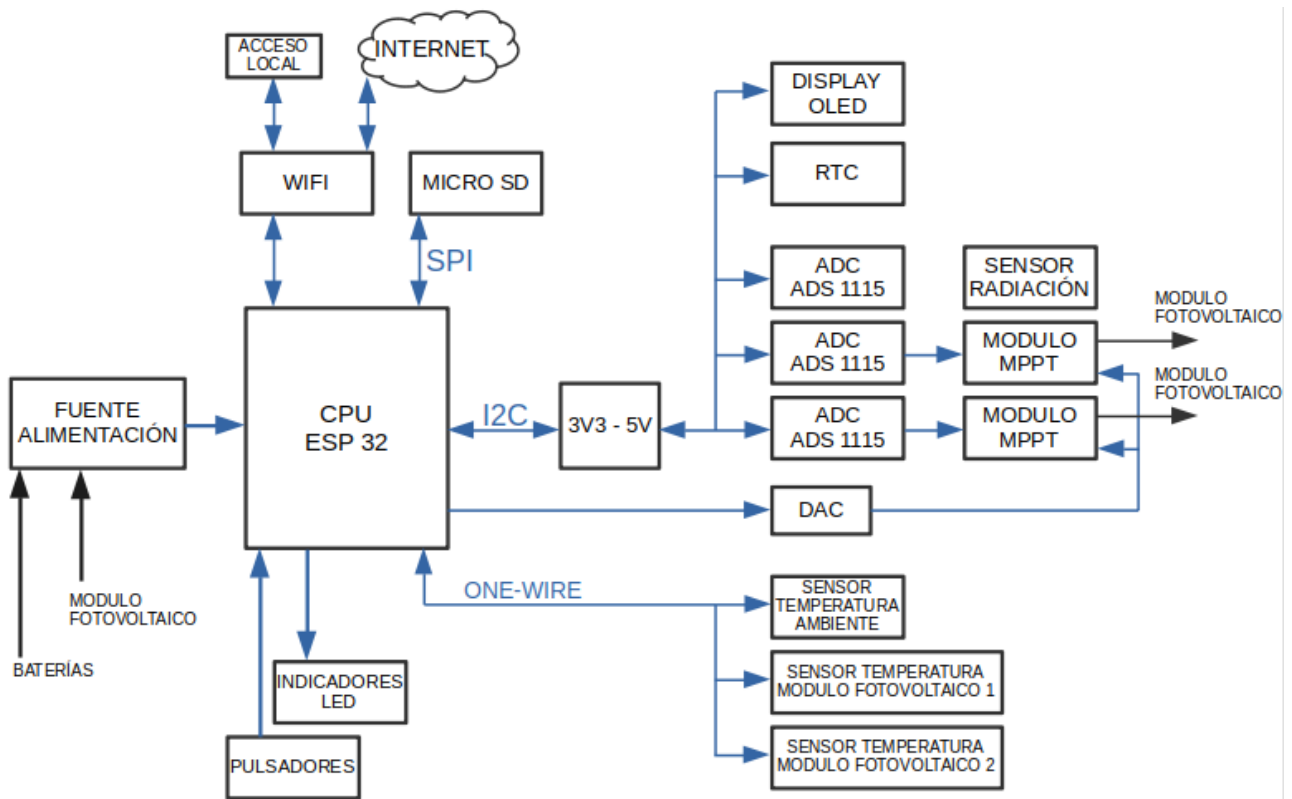


Figura 5. Diagrama en bloque del dispositivo.

El servicio web es una parte importante del sistema desarrollado, ya que brinda a los usuarios la interfaz gráfica necesaria para interactuar con los módulos fotovoltaicos instalados y también la capacidad de administrar los equipos, configurar diferentes sistemas y obtener datos de cada uno de ellos de forma remota para su posterior análisis.

El servicio se compone de una web app desarrollada por cuatro microservicios trabajando en conjunto, conocido como stack MEAN (por sus siglas):

- Angular framework
- MongoDB databases
- NodeJS
- Express

Cada uno de ellos se detallarán en la sección 2.1.7

La fuente de alimentación necesaria para el funcionamiento del Equipo MPPT, fue desarrollada para que tenga como primera medida la capacidad de poder conectarse en su entrada a un panel solar independientemente de la tecnología de este, obteniendo de allí la alimentación para el circuito, además debe poseer baterías para su uso nocturno y

desconexión del panel cuando el mismo se esté ensayando, de poseer alta eficiencia y entregar alimentación a 5V y 12V.

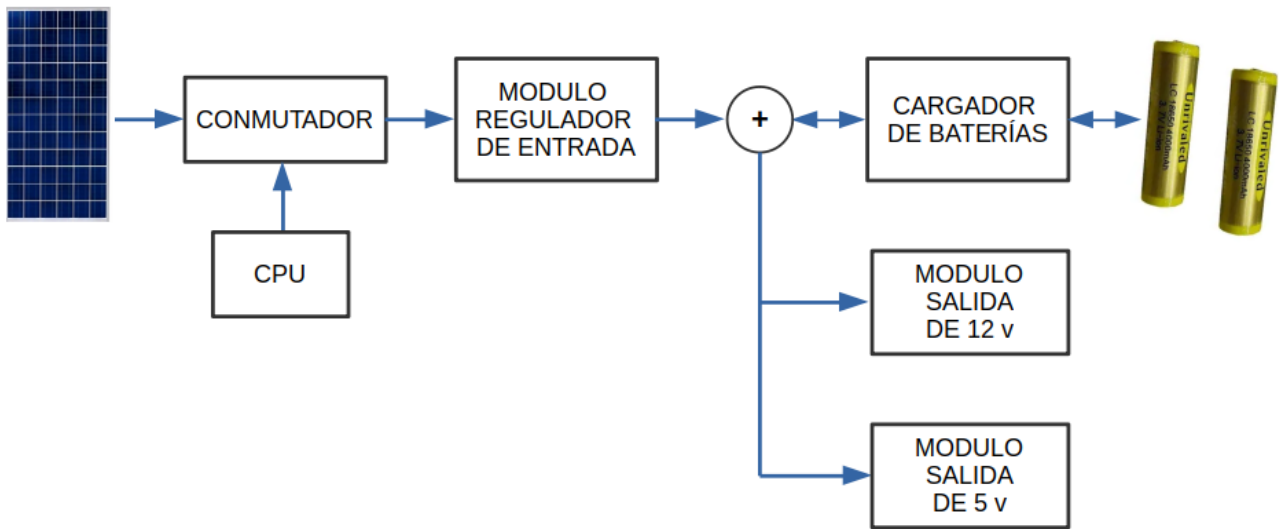


Figura 6. Diagrama en bloques de la fuente de alimentación.

2.1 Desarrollo del sistema

A continuación, se explicarán en detalle cada una de las partes intervinientes en el sistema completo.

2.1.1 CPU, conversores ADC y periféricos.

Nuestro sistema se diseñó en base a un dispositivo que conlleva un CPU central, encargado del sensado de las variables a evaluar, del control de los periféricos mencionados anteriormente en el diagrama en bloques, de la comunicación del dispositivo con las distintas partes del sistema y brindar un punto de acceso inalámbrico para la comunicación entre usuario y dispositivo. Todo lo mencionado anteriormente se montó sobre una placa base que se detalla más adelante.

A. Investigación de componentes

Dentro de la placa base se colocó como CPU central, encargado del funcionamiento completo del equipo y de brindar un punto de acceso inalámbrico vía WiFi, un NodeMCU ESP32 fabricado por Espressif. Este dispositivo posee un microprocesador de 32-bits Tensilica Xtensa LX6 (configurable como dual core o single core), comunicación inalámbrica Wifi y Bluetooth, incluye seguridad criptográfica por hardware, un coprocesador para el modo bajo consumo y periféricos como: I2C, SPI, CAN, PWM, entre otros.



Figura 7. Microcontrolador NodeMCU ESP32.

El mismo posee las características indicadas en la siguiente tabla.

Tensión del módulo	5V
Tensión del Microprocesador	2.2V a 3.6V
Microprocesador	Xtensa Dual-Core 32-bit LX6 con 600 DMIPS
Wi-Fi (802.11 b / g / n)	HT40
Bluetooth	Bluetooth 4.2 y BLE
Frecuencia de operación (valor típico)	Hasta 240 MHz de frecuencia de reloj (en el ide solo hasta 80MHz)
SRAM	448 KB
Flash	4 MB
GPIO	28
PWM (hardware)	No posee
PWM (software)	16 canales
SPI	3
I2C	2
I2S	2
UART	3
ADC	12-bits de resolución
CAN	Sí
Interfaz MAC Ethernet	Sí
Sensor de tacto	Sí
Sensor de efecto hall	Sí
Temperatura de trabajo	-40°C to 125°C

Tabla 1. Características del microcontrolador NodeMCU ESP32.

En la siguiente imagen, tomada de la hoja de datos, se muestran todos los bloques funcionales que conforman un SoC ESP32.

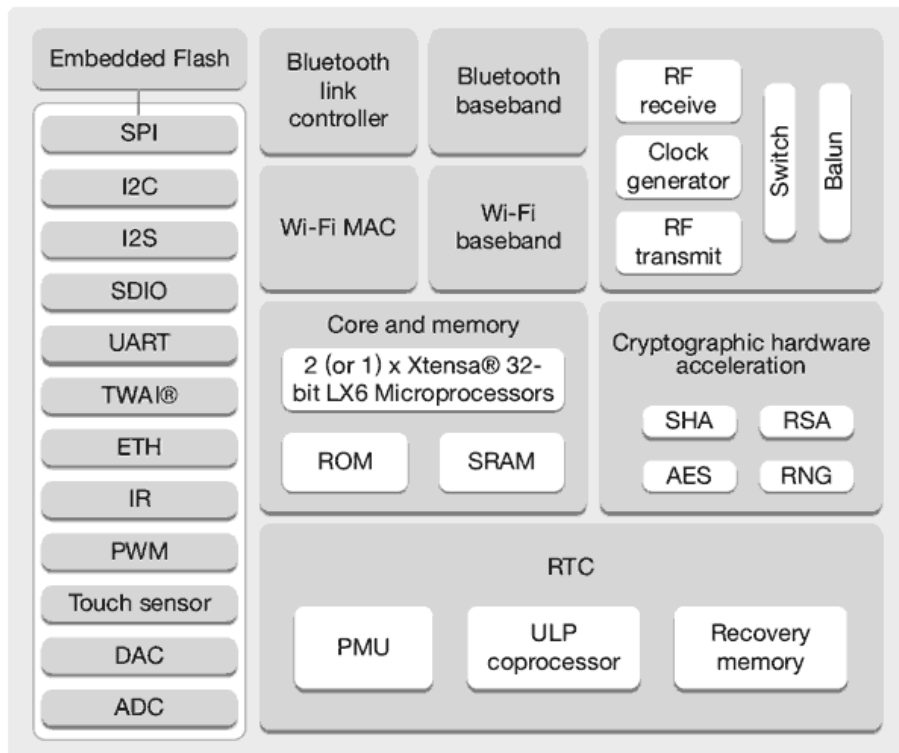


Figura 8. Estructura interna del microcontrolador ESP32.

Para nuestra implementación se consiguió el módulo Node-32s DEVKIT-C que posee el pinout mostrado en la siguiente imagen.[4]

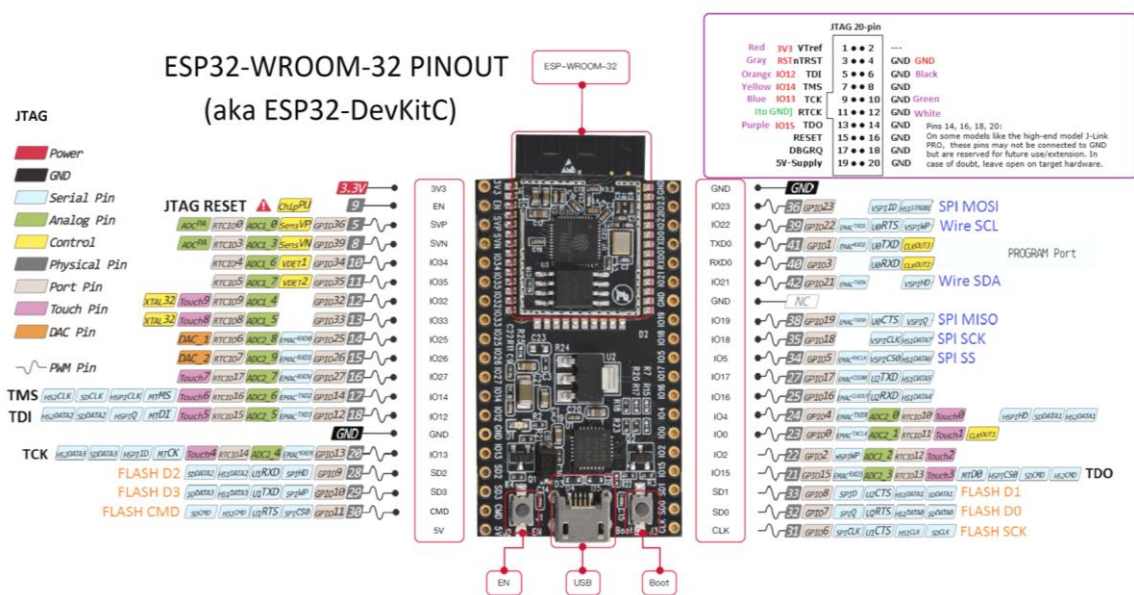


Figura 9. Diagrama de pin-out del microcontrolador ESP 32.[5]

Para el sensado de las variables analógicas se utilizó un módulo conversor analógico digital (ADC) ADS1115, el cual posee un rango de alimentación de 2V a 5V, proporciona una conversión de 16 bits de resolución (15 para la medición y un último para el signo), posee 4 canales de conversión simple con la posibilidad de realizar conversiones diferenciales tomando 2 pares de estos 4 canales, dispone de un modo comparador en el que el ADS1115 genera una alerta por un pin digital denominado ALERT y una de sus grandes ventajas es que incorpora un PGA que permite ajustar la ganancia desde 6,144V a 0,256V, esto permite obtener precisiones superiores cuando midamos tensiones inferiores a 5V. El mismo ADS1115 es configurable vía software, por comunicación serial por protocolo I2C.

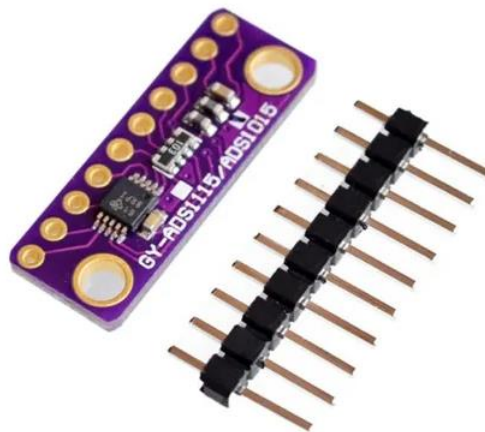


Figura 10. Módulo ADC: ADS1115.

El módulo posee las características mencionadas en la siguiente tabla.

Característica	Valor
Voltaje de operación	de 2V a 5,5V
Consumo de corriente	150 μ A (modo continuo)
Velocidad de muestreo programable	de 8sps a 860sps
Resolución	16-bit
Canales	4 canales de entrada o 2 diferenciales
Interfaz de comunicación	I2C (4 direcciones)
PGA programable	de 0,256V a 6,144V
Referencia interna de tensión	
Oscilador interno	

Tabla 2. Características módulo ADC ADS1115

Para el reloj interno del sistema se utilizó un módulo de reloj de tiempo real RTC de alta exactitud, compuesto por el integrado DS3231 línea Maxim, el cual posee una alimentación de 5V y se configura mediante comunicación serial por protocolo I2C.



Figura 11. Módulo RTC

Como características del mismo podemos mencionar:

- Fecha con segundos, minutos, horas, número día, día de la semana, mes y año.
- Compensación de año bisiesto.
- Formato de hora configurable en 12 o 24 horas
- 2 alarmas configurables
- Circuito compensador de temperatura para el voltaje de referencia interno
- Voltaje de alimentación: 3.3V - 5V DC
- RTC de alta precisión DS3231 con oscilador interno
- Exactitud Reloj: 2ppm
- Interfaz de comunicación digital: I2C
- Dirección I2C del DS3132: Read (11010001) Write(11010000)
- Memoria EEPROM AT24C32 (4K * 8bit = 32Kbit = 4KByte)
- Salida de onda cuadrada programable
- La batería puede mantener al RTC funcionando por 10 años.

El aparato se diseñó con la funcionalidad de poseer un almacenamiento de información utilizando tarjeta MicroSD, para realizar esta tarea se utilizó un Módulo Micro Sd Card 3v3 - Card Reader Arduino genérico, el cual posee una alimentación de 3.3V y se configura mediante comunicación serial por protocolo SPI.



Figura 12. Módulo micro SD

El dispositivo también tiene como funcionalidad la visualización de su trabajo en condición de uso local, para la tarea de visualización se utilizó un módulo display OLED de alto contraste compuesto por el driver SSD1306.



Figura 13. Display OLED

Esta pantalla posee 0.96 pulgadas de diagonal, muy visible, con una resolución de 128x64 puntos y tecnología OLED, no necesita retroiluminación y tiene un alto contraste incluso a plena luz del día. El driver interno SSD1306 se configura mediante comunicación serial por protocolo I2C y posee una alimentación de 5V.

El módulo contiene:

- Tecnología OLED de alto contraste.
- Resolución: 128x64 píxeles (Blanco)
- Controlador: SSD1306
- Interfaz I2C

- Ángulo de visión: $>160^\circ$
- Alimentación: 3 a 5V DC
- Dimensiones: 2.2 cm x 2.8 cm
- Temperatura de funcionamiento: -30 a +70 °C

Para todas las funcionalidades del dispositivo, se utilizan todos los periféricos mencionados hasta ahora conectados al CPU central, se combinaron módulos con alimentación de 5V conectados a un solo bus I2C al NodeMCU que posee una alimentación de 3.3V, por esta razón, es necesario adaptar la tensión de los canales de comunicación. Para la conversión de tensión del canal I2C se utilizó un módulo adaptador de nivel lógico TTL de 5V a 3.3V bidireccional genérico transistorizado.

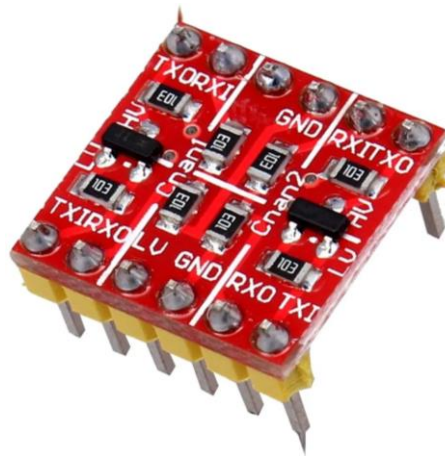


Figura 14. Conversor 3v3 - 5v

B. Circuitos propuestos y su funcionamiento

En primer lugar, el módulo Node-32s es el encargado de suministrar la alimentación de 3.3V al sistema, ya que en su interior contiene un regulador lineal AMS1117-3.3 que utiliza para su funcionamiento, de este pin de alimentación brinda la tensión de funcionamiento a todos los periféricos que se alimenten con 3.3V.

El sistema tiene como funcionalidad principal poder tener dos modos de funcionamiento que serán explicados más adelante, denominados “Modo Online” y “Modo Sleep”, el primero permite un funcionamiento directo entre el usuario y el equipo mientras que el segundo otorga la funcionalidad automática en base a las características de sensado y alarmas que se configuraron a este. Para poder discriminar entre un modo de operación u otro se implementó en el equipo el pulsador BTN1, el cual si mantenemos accionado mientras inicia el sistema nos dejará al equipo entrar en “Modo Online”, una vez allí el sistema tiene la opción de tener un funcionamiento directo, es posible con el equipo obtener el sensado de una curva I-V instantánea con las características que el

dispositivo tiene almacenadas, solo basta con accionar el pulsador BTN1 para obtener una curva de sensado y almacenarla en la tarjeta SD.

También es factible cuando el dispositivo esté en “Modo Online”, entrar en un “Modo Debugger” pudiendo extraer información del funcionamiento en tiempo real del sistema, el pulsador BTN2 tiene esta funcionalidad y permite cambiar el modo de operación para que esta información pueda ser extraída por Puerto Serie o Bluetooth. Además de esto último, el pulsador BTN2 en caso de mantenerlo presionado, genera un apagado de emergencia o reset que puede ser útil si el sistema entra en un modo no deseado o no responde.

Por último, podemos mencionar los indicadores lumínicos que el dispositivo posee, un total de 5 leds que tienen por objeto indicar al usuario los siguientes procesos:

- Si existe alimentación de 5V (Led D7)
- Si existe alimentación de 12V (Led D6)
- Si se está recibiendo información de algún medio (Led D5)
- Si se está enviando información hacia algún medio (Led D4)
- Si se está realizando alguna acción que requiera proceso del CPU, por ejemplo, el sensado de una Curva I-V (Led D3)

Lo descrito anteriormente se puede ver en el siguiente esquemático

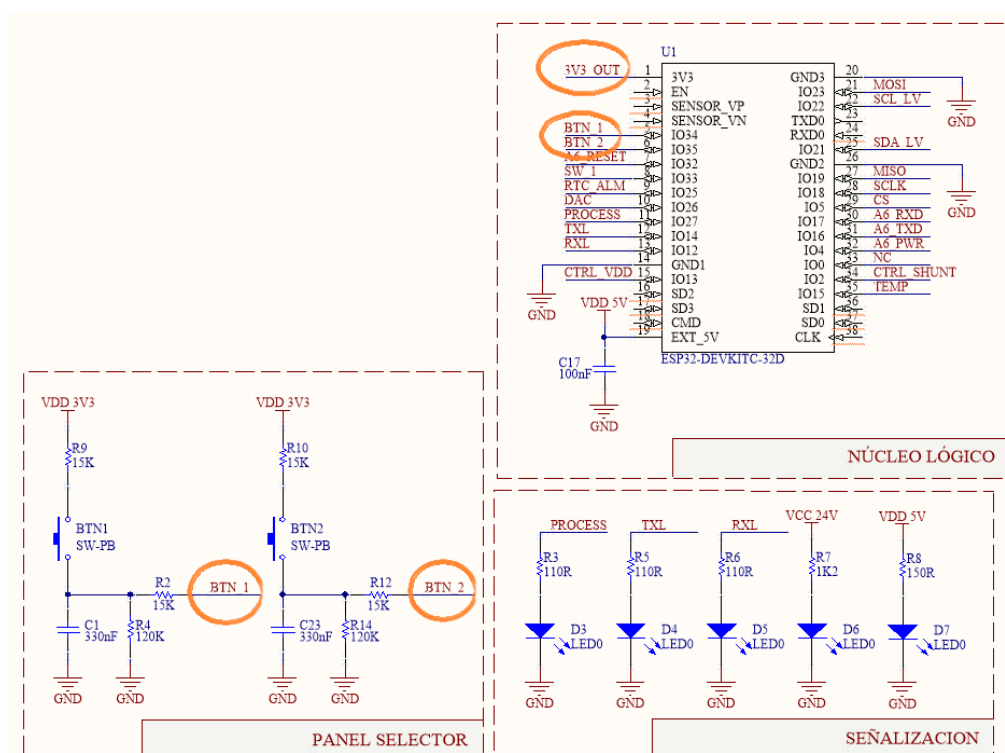


Figura 15. Esquema eléctrico pulsadores e indicadores LED

El sistema tiene como funcionalidad poder almacenar toda la información del sensado de una curva I-V, para ello se implementó como un periférico un módulo de

tarjeta MicroSD que se conecta por comunicación serial por protocolo SPI al CPU del equipo, la conexión de este módulo y su alimentación se pueden ver en el siguiente esquemático:

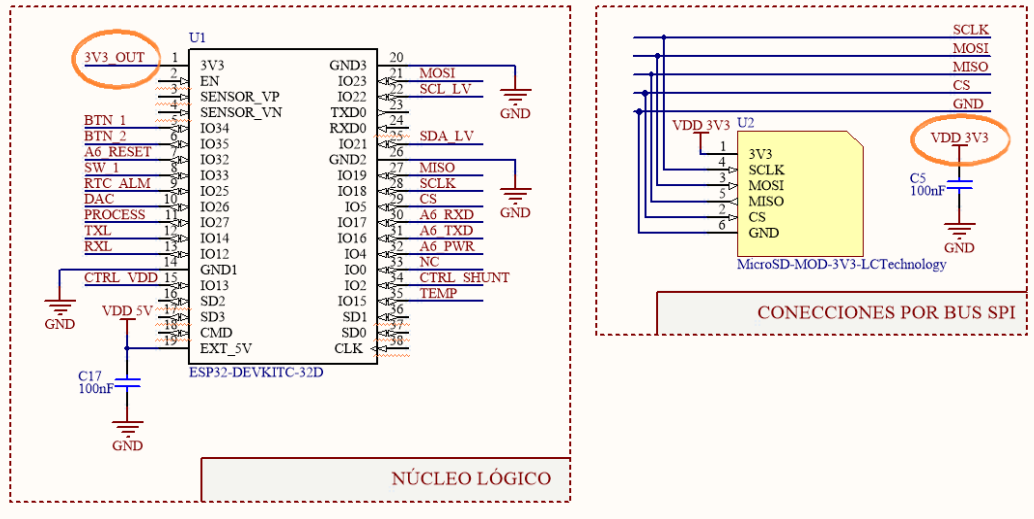


Figura 16. Conexión ESP32 - Micro SD

El CPU almacena dentro de una tarjeta MicroSD colocada en el módulo dos tipos de archivos de texto, el primero con extensión “.csv” es una hoja de cálculo que posee los datos del sensado de una curva I-V con todos sus parámetros.

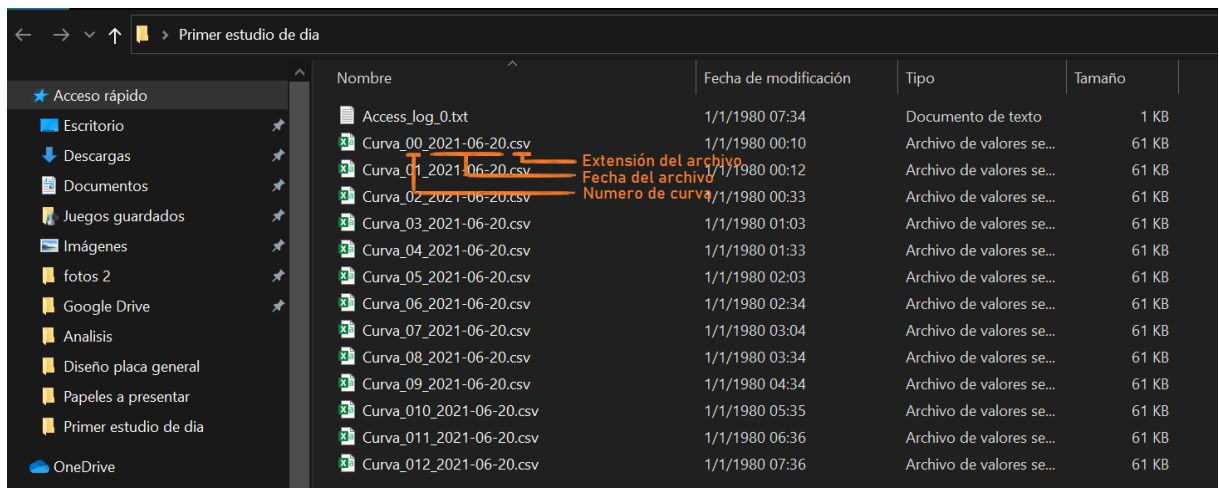


Figura 17. Ejemplo archivos guardados en micro SD

Si analizamos los datos dentro de este archivo, podemos encontrar la siguiente información:

- Fecha de realización del ensayo (Celda B3)
- Hora de inicio del proceso de sensado (Celda B6)
- Hora de finalización del proceso de sensado (Celda C6)
- Temperatura inicial del panel fotovoltaico 1 a ensayar (Celda B7)
- Temperatura final del panel fotovoltaico 1 a ensayar (Celda C7)

- Temperatura inicial del panel fotovoltaico 2 a ensayar (Celda B8)
- Temperatura final del panel fotovoltaico 2 a ensayar (Celda C8)
- Temperatura del equipo al iniciar el ensayo (Celda B9)
- Temperatura del equipo al finalizar el ensayo (Celda C9)
- Temperatura ambiente al iniciar el ensayo (Celda B10)
- Temperatura ambiente al finalizar el ensayo (Celda C10)
- Radiación al iniciar el ensayo (Celda B11)
- Radiación al finalizar el ensayo (Celda C11)
- Tiempo real del escaneo de curva I-V (Celda B14)
- Número de punto sensado (Columna B)
- Valor de tensión (Columna D)
- Valor de corriente (Columna H)

	A	B	C	D	E	F	G	H	I
1	IV Tracer - Comparacion entre Panel Solares								
2	FT1								
3	Fecha	20/6/2021							
4									
5	Parametro	Valor inicial	Valor final	Resolucion					
6	HRS	14:00:25	14:00:30	1[s]					
7	TMP_PFV0	34.13	34.06	0.0625[Â°C]					
8	TMP_PFV1	-0.01	-0.01	0.0625[Â°C]					
9	TMP_IV-Tracer	-0.01	-0.01	0.0625[Â°C]					
10	TMP_HMB	-0.01	-0.01	0.0625[Â°C]					
11	RAD	85	90						
12									
13	Cantidad de espera:	0							
14	Tiempo exploracion:	3859 [ms]							
15									
16	Panel_0								
17	Pto. caracteristico PanelFV-0 n:	1	Tension:	16.92 [V]			Corriente:	0 [mA]	
18	Pto. caracteristico PanelFV-0 n:	2	Tension:	16.8 [V]			Corriente:	202.39 [mA]	
19	Pto. caracteristico PanelFV-0 n:	3	Tension:	16.78 [V]			Corriente:	239.13 [mA]	
20	Pto. caracteristico PanelFV-0 n:	4	Tension:	16.75 [V]			Corriente:	275.54 [mA]	
21	Pto. caracteristico PanelFV-0 n:	5	Tension:	16.73 [V]			Corriente:	313.26 [mA]	
22	Pto. caracteristico PanelFV-0 n:	6	Tension:	16.71 [V]			Corriente:	349.67 [mA]	
23	Pto. caracteristico PanelFV-0 n:	7	Tension:	16.68 [V]			Corriente:	387.73 [mA]	
24	Pto. caracteristico PanelFV-0 n:	8	Tension:	16.66 [V]			Corriente:	424.46 [mA]	
25	Pto. caracteristico PanelFV-0 n:	9	Tension:	16.63 [V]			Corriente:	462.52 [mA]	
26	Pto. caracteristico PanelFV-0 n:	10	Tension:	16.61 [V]			Corriente:	497.94 [mA]	
27	Pto. caracteristico PanelFV-0 n:	11	Tension:	16.59 [V]			Corriente:	535.99 [mA]	
28	Pto. caracteristico PanelFV-0 n:	12	Tension:	16.56 [V]			Corriente:	573.73 [mA]	

Figura 18. Ejemplo archivo .csv generado por el microcontrolador

El segundo archivo almacenado, es llamado "Access_log_N.txt" que informa al usuario la cantidad de encendidos que tuvo el equipo, esto nos sirve para saber si el equipo se está reseteando constantemente de mala manera, si tuvo algún apagado fuera de término o si tiene algún otro tipo de problema. En caso de funcionar de manera correcta debemos tener un solo archivo "Access_log_0.txt" significando que la tarjeta SD funciona normalmente o el archivo no está corrompido y siempre debemos leer que el número de ingresos informados dentro del archivo sea el número de curvas generadas por el sistema, esto indica que la cantidad de encendidos del equipo fueron producidos por una alarma del despertador para realizar el sensado correspondiente, esto lo podemos ver en la figura siguiente.

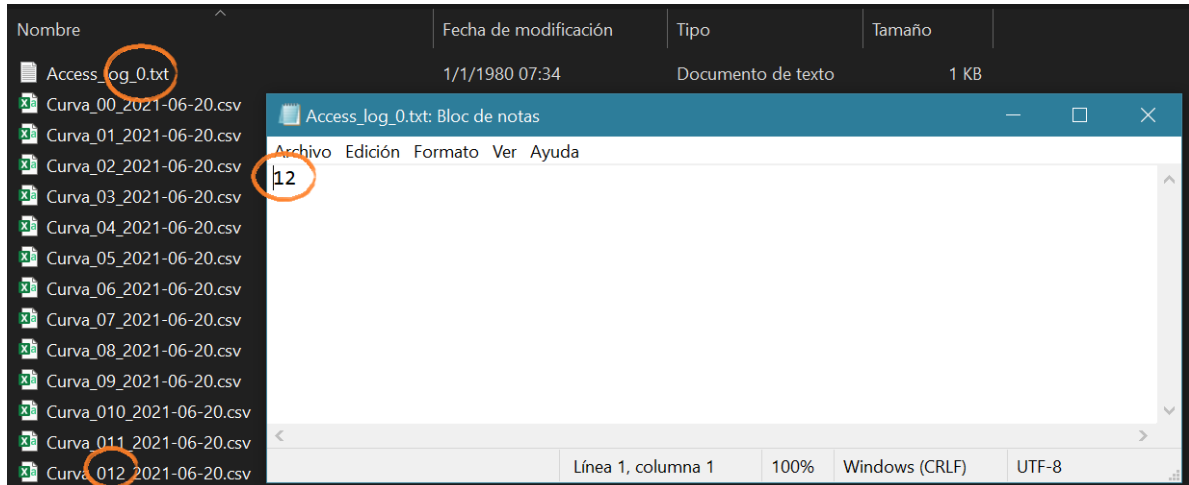


Figura 19. Ejemplo de archivo logs del microcontrolador

Todos estos archivos se almacenan en distintos directorios con el nombre “MPPT_fecha_log”, donde podremos encontrar en cada carpeta todas las curvas realizadas en ese día, por ejemplo, los archivos de la imagen anterior se encuentran en una carpeta llamada “MPPT_2021-06-20_log”.

El equipo tiene como funcionalidad poder registrar fecha y hora de los procesos, realizar el sensado de 5 variables analógicas y poder visualizar todo el proceso al usuario a través de algún tipo de pantalla. Para realizar esta tarea se implementaron periféricos conectados por comunicación serial por protocolo I2C a un mismo bus como se muestra en el esquemático siguiente:

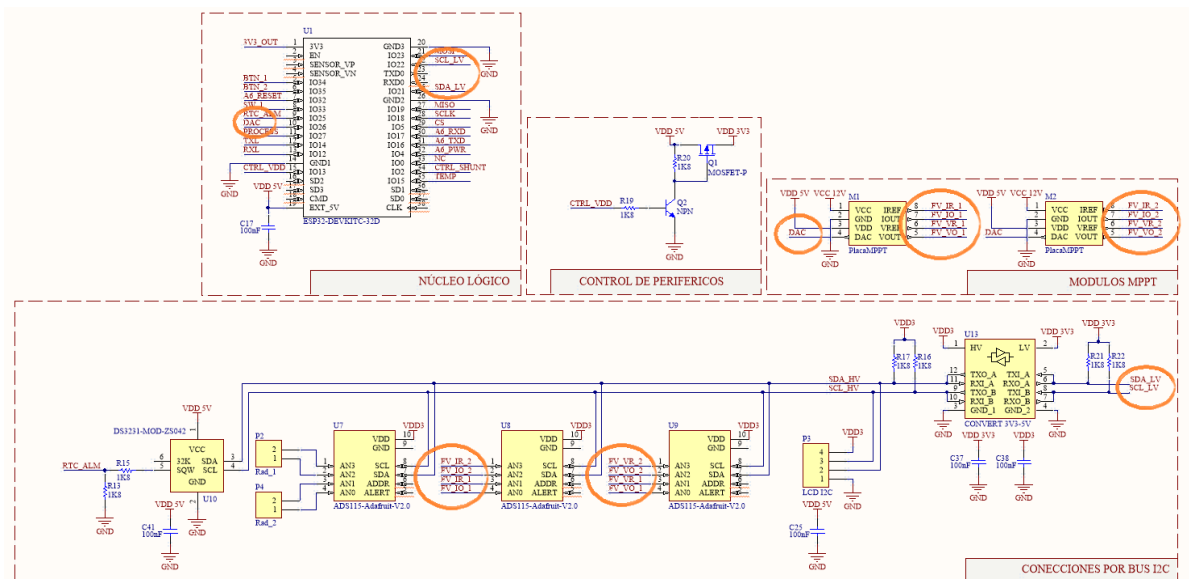


Figura 20. Esquema eléctrico bus de datos I2C

Comunicar encendido:

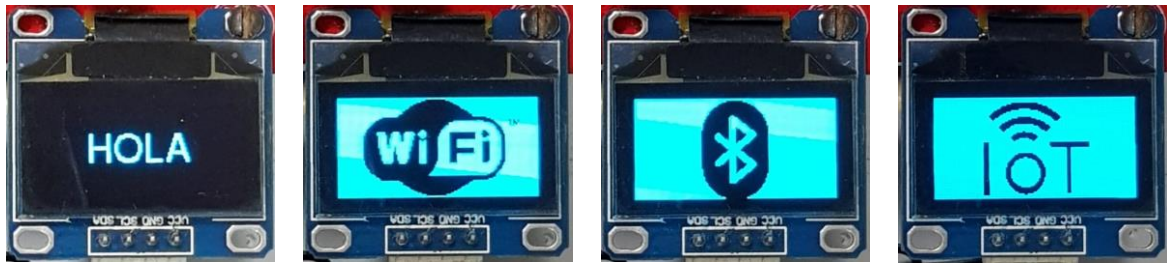


Figura 22. Mensaje encendido display OLED

Comunicar que se configuró el equipo correctamente:



Figura 23. Mensajes de inicio display OLED

Comunicar modo de funcionamiento:



Figura 24. Mensajes modo de funcionamiento en display OLED

Comunicar hora en modo Online



Figura 25. Muestra de hora en display OLED

Comunicar acción del usuario en modo Online:



Figura 26. Mensajes "ejecutando barrido" en display OLED

Comunicar levantado de curva



Figura 27. Progressbar display OLED

Por último, si vemos bien el esquemático se observa que todos los periféricos se encuentran alimentados a una tensión "VDD3" y no a 5V directamente, esto es así ya que el equipo posee internamente una llave electrónica implementada con el Mosfet Q1 que permite cortar la alimentación de los periféricos cuando el mismo entra en "Modo Sleep", esto se debe a la necesidad de reducir el consumo del dispositivo cuando se lo necesite y es controlada por el ESP32 a través del pin CTRL_VDD. Un detalle que podemos mencionar es que el único periférico que tiene alimentación directa de 5V es el módulo RTC ya que nunca debe apagarse para proporcionar la alarma para despertar el equipo.

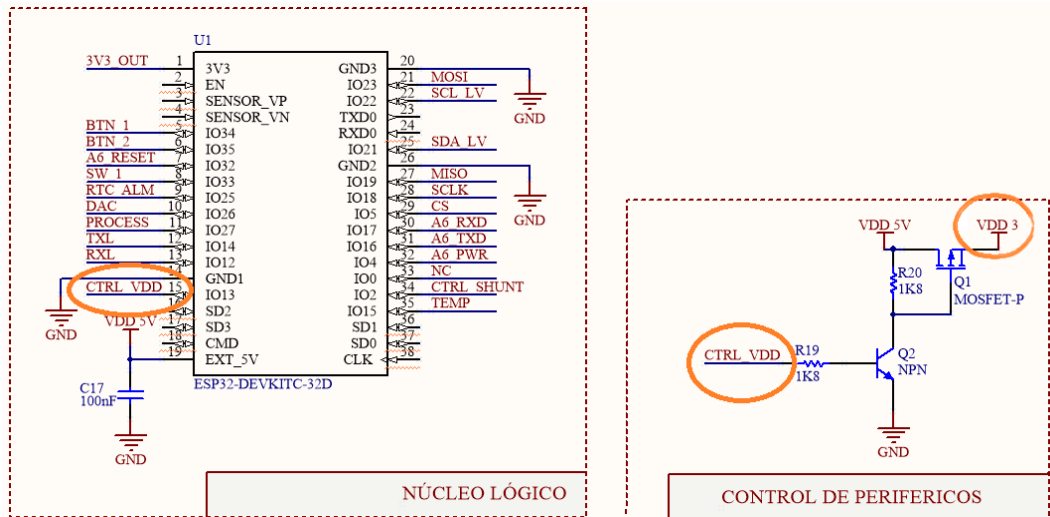


Figura 28. Esquema eléctrico circuito desconexión de alimentación de módulos

C. Desarrollo del software

La logica de programacion que utiliza el CPU del dispositivo es la siguiente:

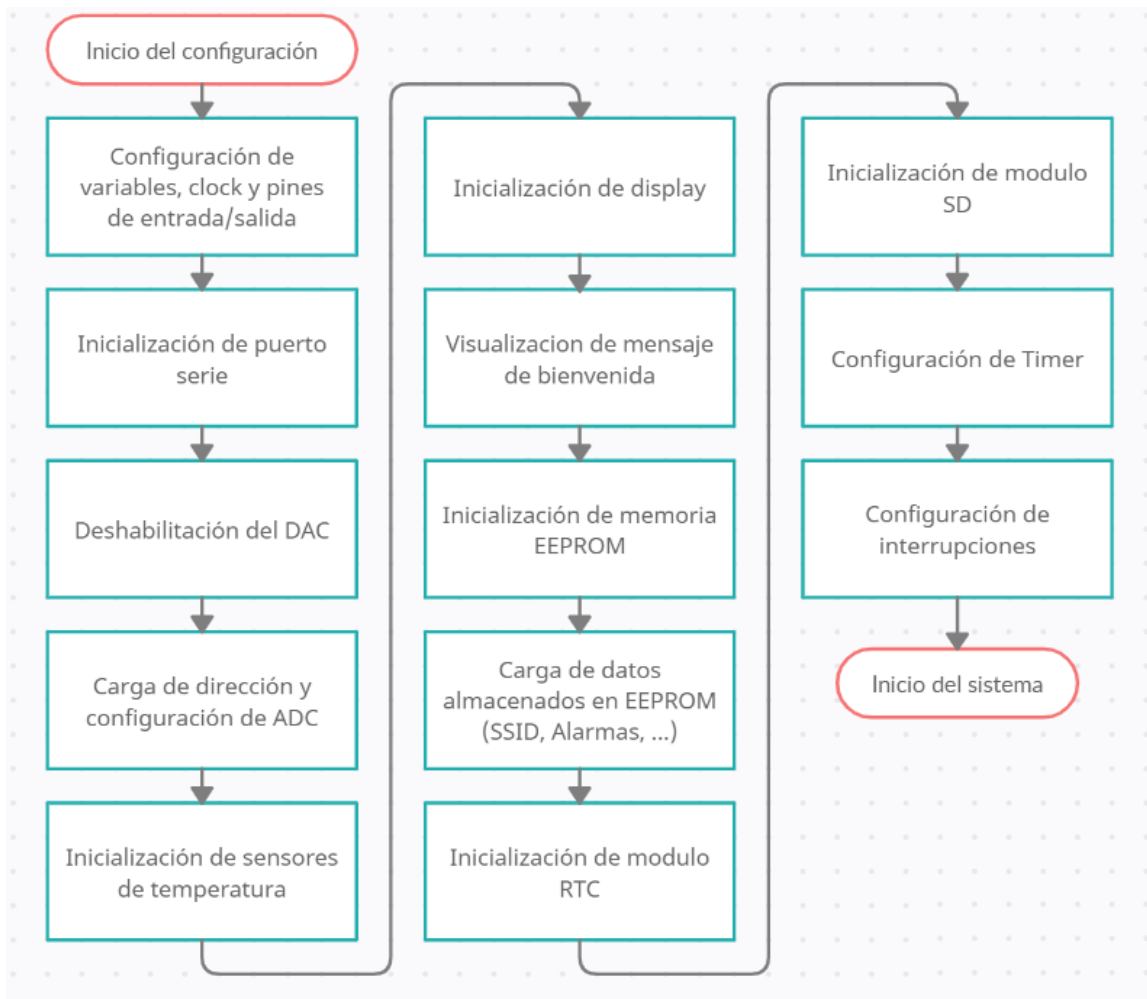


Figura 29. Logica del CPU - Parte 1

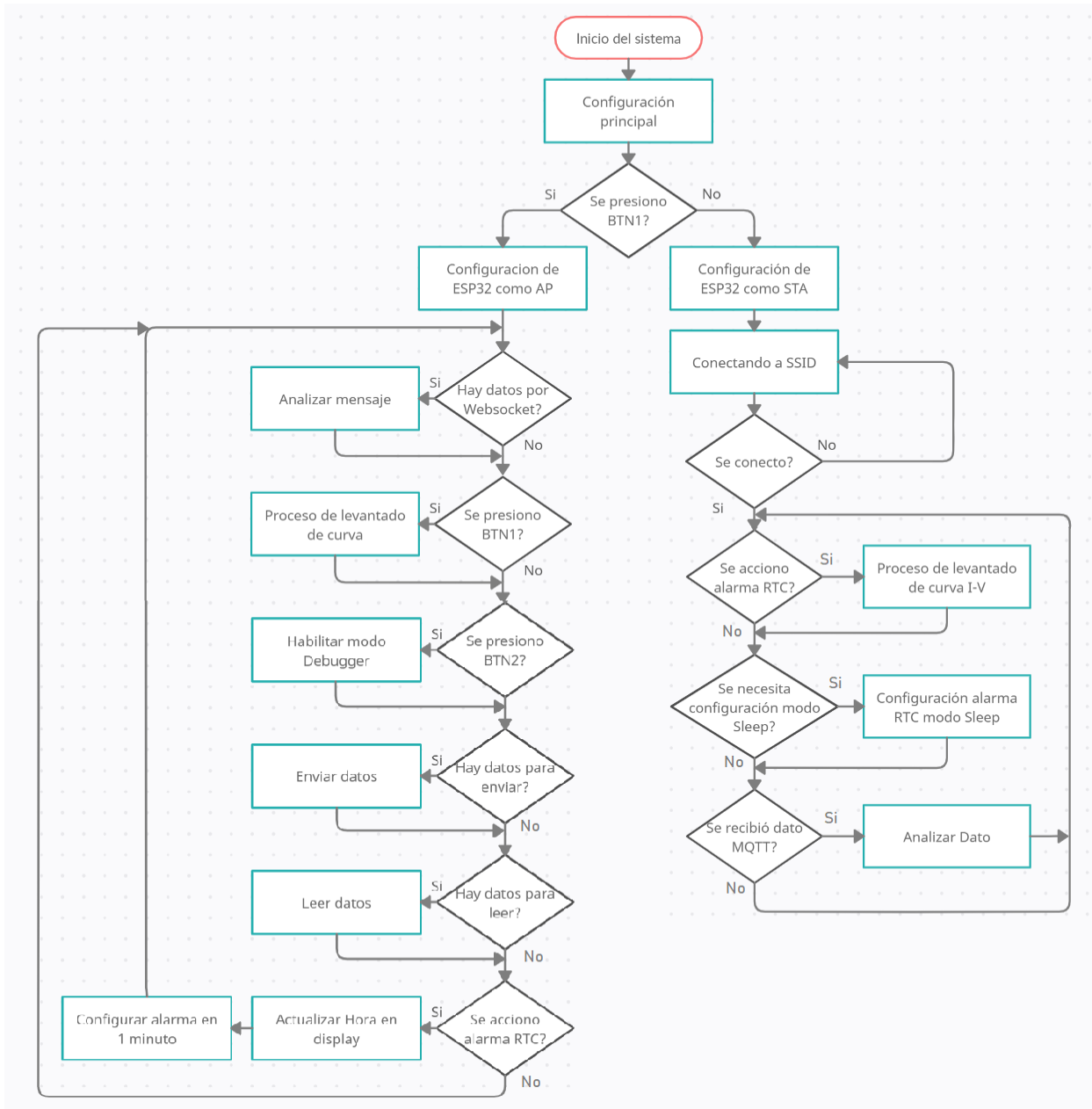


Figura 30. Logica del CPU - Parte 2

2.1.2 Módulos MPPT.

El módulo MPPT es una placa diseñada con el fin de ser para los módulos fotovoltaicos como una carga electrónica, la cual es controlada por el microcontrolador.

Para ello, se envía a esta placa una señal de tensión con forma de diente de sierra proveniente del DAC interno del ESP32, la cual por medio de un conversor V-I define la corriente de consumo que se demandará al módulo fotovoltaico para luego ser sensado.

A. Investigación de componentes

Para el control de corriente del panel se implementó una carga electrónica que funciona como fuente de corriente constante, la cual posee en su etapa de salida un MOSFET canal N, IRFP150N.



Figura 31. MosFET utilizado como carga electrónica

La elección de este transistor se basó en su baja $R_{DS(on)}$, alta corriente de trabajo, pero sobre todo por sus características disipativas y disponibilidad de mercado. Se adjunta datasheet para ver sus características.

El convertor V-I se diseñó en base a un amplificador operacional que cumple esta función, se realizó una investigación para determinar cuáles eran las características necesarias para que el dispositivo se adecue a nuestro diseño, se probó si era necesario que el mismo tenga cualidades como bajo ruido (low noise), baja corriente de bias (low bias input current) y alto rango de operación de tensión (rail to rail), donde se estudiaron operacionales como MCP2002, LMV772, TL072, entre otros. Como el funcionamiento del circuito no requiere altas prestaciones se optó por utilizar un A.O. de propósito general, económico, con gran disponibilidad en el mercado y que funcione con fuente de alimentación simple, para nuestro diseño utilizamos un LM358N.

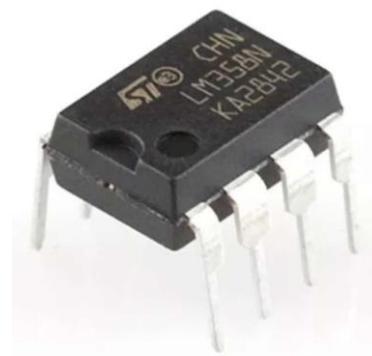


Figura 32. Amplificador operacional

B. Circuito propuesto

El módulo MPPT es una placa más pequeña diseñada con el fin de funcionar como carga electrónica conectada al panel solar. El circuito MPPT se diseñó como un módulo aparte con el objeto de ser reemplazable para el dispositivo, uno podría tranquilamente tomar este proyecto y cambiar a conveniencia este módulo particular, ya sea por otra tecnología de funcionamiento o por requerimiento del panel solar a ensayar, inclusive se podría remover el módulo y adaptar la conexión de este a un sistema aparte para evaluar plantas solares más grandes.

En nuestro caso se tomó como punto final poder ensayar paneles solares de hasta 50V de tensión en vacío VOC y corrientes de cortocircuito ISC de hasta 10A. El circuito propuesto es el mostrado a continuación.

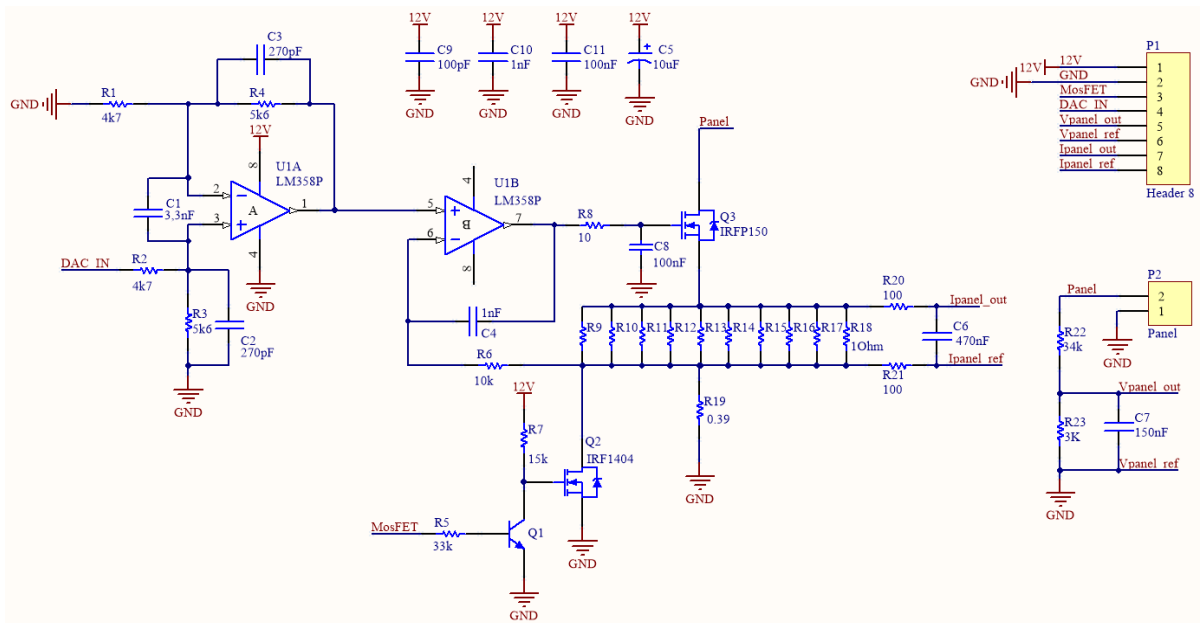


Figura 33. Esquema eléctrico módulo MPPT

El módulo funciona como convertor V-I, en la primera parte del diseño donde tenemos el primer A.O. con denominación U1A se encuentra un filtro combinado con un amplificador diferencial que funcionan como adaptador para la señal de excitación, la salida de este dispositivo ingresa a un convertor V-I implementado por un segundo A.O. con denominación U1B, el cual tiene como punto final el Mosfet canal N ya mencionado anteriormente. El panel solar se debe conectar a este módulo a través de la bornera P2.

Este dispositivo debe suministrar a la placa base los puntos de sensado de tensión y corriente del panel, estos mismos se realizan de manera diferencial con salida en los pines 5, 6, 7 y 8 del conector Sil-8 con denominación P1 de placa. Para el sensado de tensión se toma un divisor resistivo mediante R22 y R23 con filtro de salida y para el sensado de corriente se toma del array de resistencias de 1Ω formando una resistencia

R_{sens} de 0.1Ω con salida diferencial, de esta resistencia se coloca un filtro diferencial de orden 1 compuesto por R_{20} , R_{21} y C_6 antes de la salida del conector.

La resistencia R_{19} se utiliza como R_{shunt} para darle estabilidad al circuito y corriente máxima de sensado, la misma es cortocircuitada por el Mosfet Q_2 cuando se requiere la medición en cortocircuito.

La implementación del circuito anterior quedó fabricada en el siguiente módulo

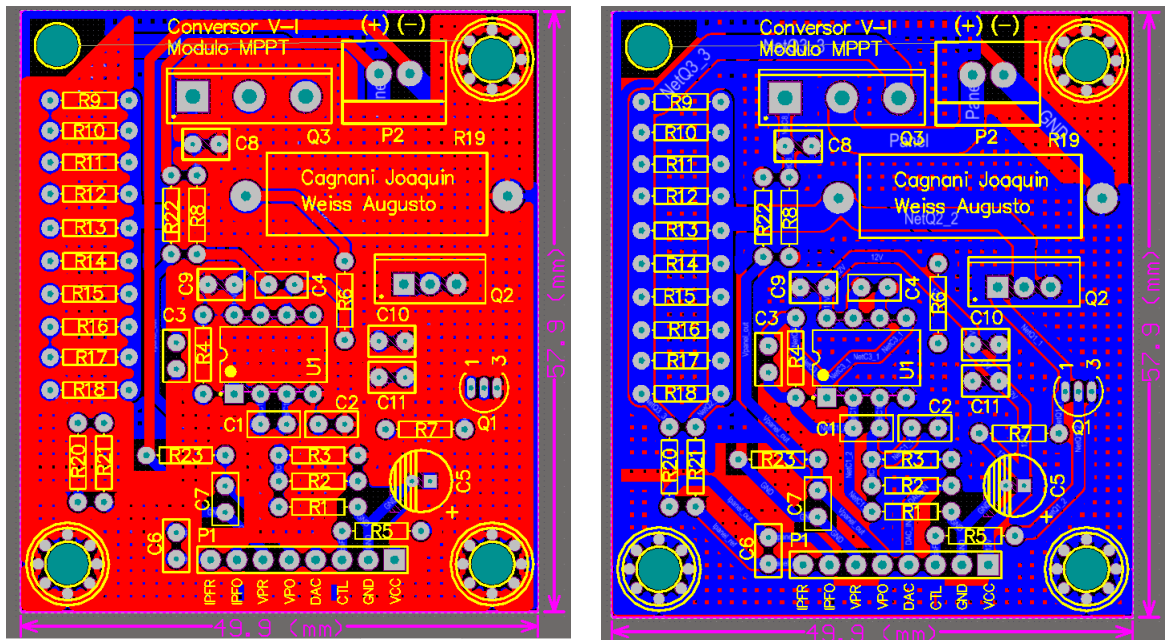


Figura 34. Diseño PCB módulo MPPT

La implementación de la placa quedó de la siguiente manera:

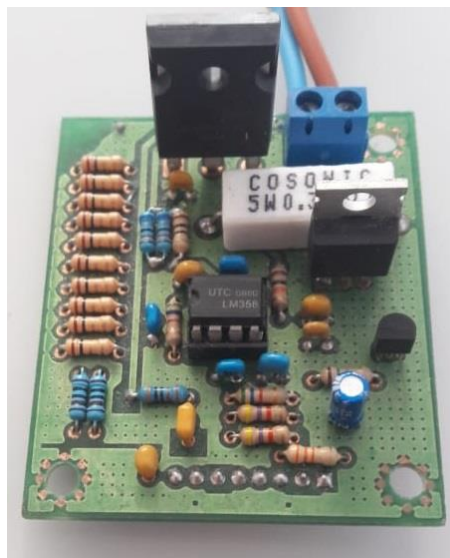


Figura 35. Módulo MPPT

C. Análisis de funcionamiento

El funcionamiento de este circuito se basa en convertir una señal de tensión proveniente del DAC interno del ESP32 a una señal de corriente a demandar en los paneles solares que se requieran ensayar. Se entrega una señal rampa creciente escalonada por el DAC al módulo MPPT y por cada punto del escalón de tensión, se toman los valores de tensión y corriente que ambos paneles solares entregan, por cada valor creciente de tensión de entrada, aumenta en valor fijo la corriente que proviene del panel. Esta demanda de corriente en aumento provoca una caída de tensión a la salida del panel siguiendo la curva característica que define el funcionamiento de cualquier panel solar, así podemos ensayar los módulos fotovoltaicos desde su valor en vacío con tensión VOC y corriente nula, a su valor en cortocircuito con corriente ISC y tensión nula. Como se ve en la imagen siguiente por cada valor de escalón, se registra un punto de corriente y tensión del panel, la cantidad de puntos es configurable por software y será explicada más adelante.

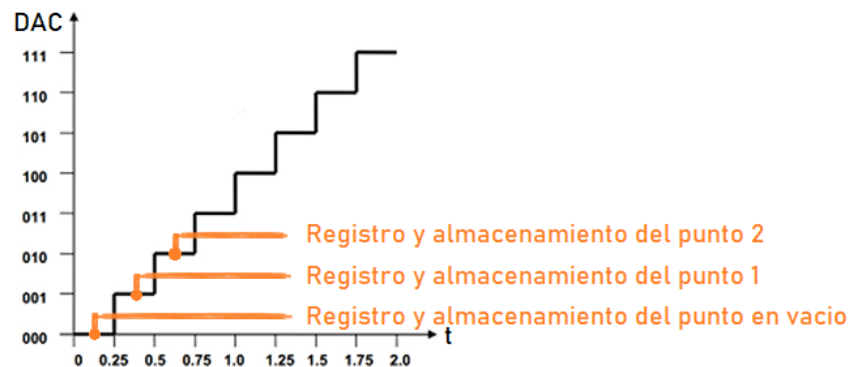


Figura 36. Rampa diente de sierra proveniente del DAC

Este módulo en forma general funcionó de buena manera, igualmente se tuvieron inconvenientes particulares para la implementación correcta de este circuito.

El primer inconveniente fue el sobrecalentamiento del Mosfet de potencia Q3 que maneja la corriente de consumo del panel, este problema se debía al offset entregado por el DAC interno del CPU del dispositivo, el DAC erróneamente entregaba una tensión muy baja del orden de los mV que generaba un consumo constante desde el panel solar hacia la carga electrónica, calentando el circuito a largo plazo. Este problema fue solucionado deshabilitando en el CPU el DAC cuando el mismo no se utiliza.

El segundo inconveniente fue un mal funcionamiento del circuito producido por la saturación del amplificador operacional debido al ruido que ingresaba por el Mosfet Q3 y la no referencia en la realimentación del circuito convertidor V-I y su salida, produciendo curvas características como las siguientes:

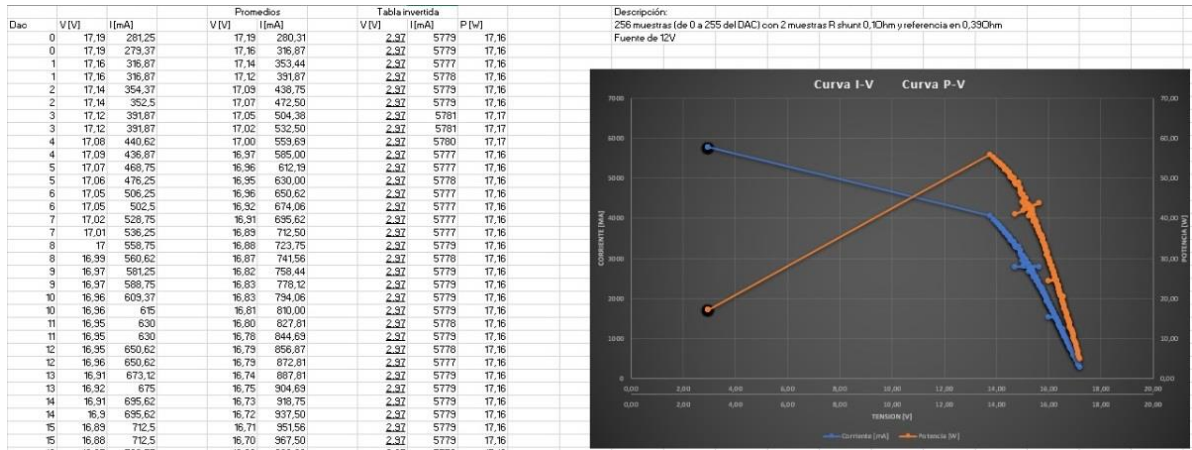


Figura 37. Curva I-V defectuosa

Este problema fue solucionado con la implementación de filtros de realimentación y filtro de salida en el convertor V-I evitando el ingreso de ruido al amplificador y la saturación del mismo. Las modificaciones son las mostradas con rojo en el siguiente esquema.

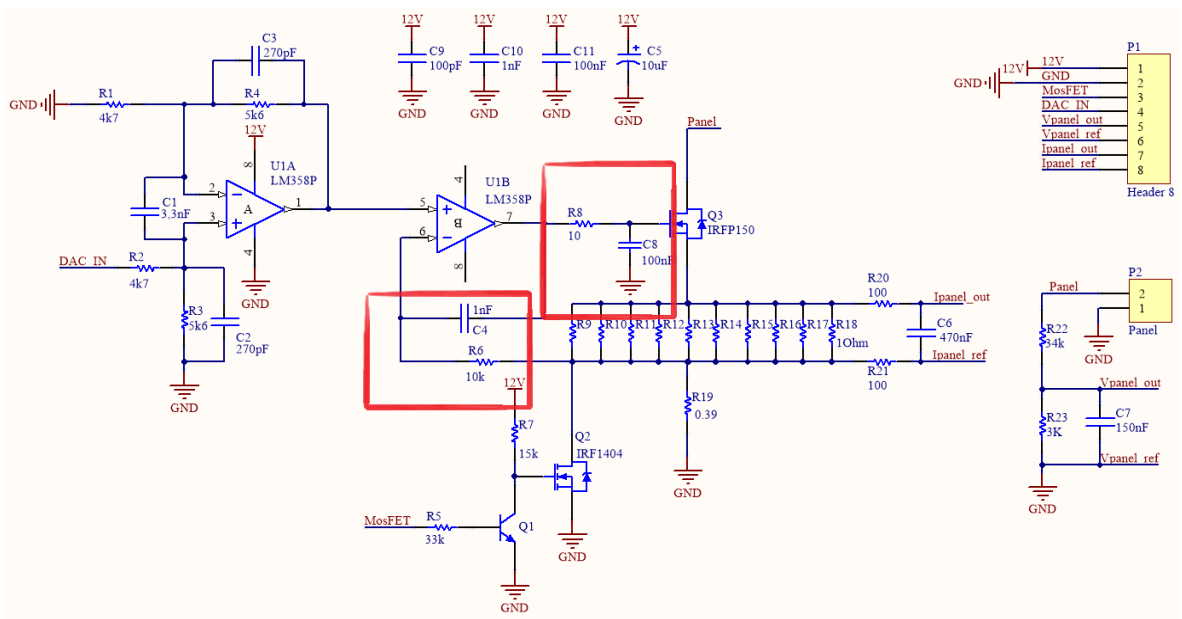


Figura 38. Esquema eléctrico módulo MPPT corregido.

Gracias a estas modificaciones el sensado de curva se pudo realizar sin perturbaciones en forma lineal obteniendo el siguiente resultado

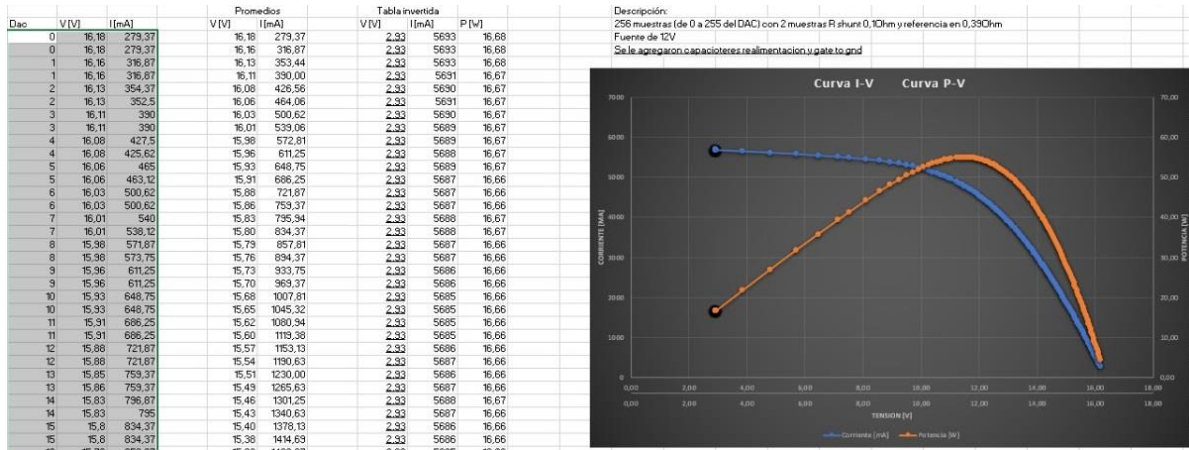


Figura 39. Curva I-V obtenida con módulo MPPT final.

Para evaluar el desempeño del módulo MPPT se verificó la linealidad del sistema, el ensayo consiste en excitar el mismo con una señal de tensión con forma de onda en diente de sierra y registrar la señal en corriente que la carga electrónica consume en su salida.

En la salida del circuito luego del filtro diferencial podemos encontrar la siguiente respuesta:



Figura 40. Comparación rampas de sierra del generador - Rshunt

La señal de color azul es la proveniente de nuestro generador de funciones, en este caso tenemos una señal de amplitud de 2,57[V] a lo que nuestro sistema responde con una señal en corriente con una amplitud de 2,98[V] sobre una Rshunt de 0,39[Ω], señal amarilla.

De este análisis podemos rescatar, en primera medida que el sistema posee una buena linealidad al circular por todo el circuito, obteniendo a la salida una señal rampa con excelentes características.

En segunda instancia podemos calcular una ganancia del módulo donde:

$$G_{MPPT} = \frac{\frac{2,98}{0,39}}{2,57} = 2,97$$

Es decir que si excitamos el circuito con una tensión de 3,3[V] de entrada (salida de tensión máxima de nuestro sistema digital) podemos obtener una corriente de carga de $G_{mppt} \cdot V_{in} = 9,8[A]$.

Esta medida no es tan exacta ya que no se midió la resistencia R_{shunt} de $0,39\Omega$ con exactitud y además no es un componente de precisión, la medición en corriente se realiza de otra manera sobre la resistencia R_{sens} , esta R_{shunt} solo garantiza la corriente máxima de carga que tenemos en nuestro dispositivo. Si esta resistencia variará a $0,38\Omega$ el cálculo nos quedaría:

$$G_{MPPT} = \frac{2,98/0,38}{2,57} = 3,05 \quad ; \quad V_{DAC} = 3,3[V] \quad ; \quad I_{SHUNT} = G_{MPPT} \cdot V_{DAC} = 10,065[A]$$

Como vemos carece de importancia el valor de esta R_{shunt} a los cálculos de medición, solo nos importa que nuestro sistema nos permite registrar una corriente máxima de $10[A]$, de manera lineal.

2.1.3 Sensores de temperatura y radiación PAR.

A. Investigación de componentes

El dispositivo tiene objeto sensar 4 diferentes temperaturas de manera precisa, para el sensado de ellas se emplearon sensores DS18B20 fabricados por DALLAS semiconductor, el cual posee una configuración digital por comunicación serial a través de protocolo 1 Wire, con una alimentación de 3.3V.

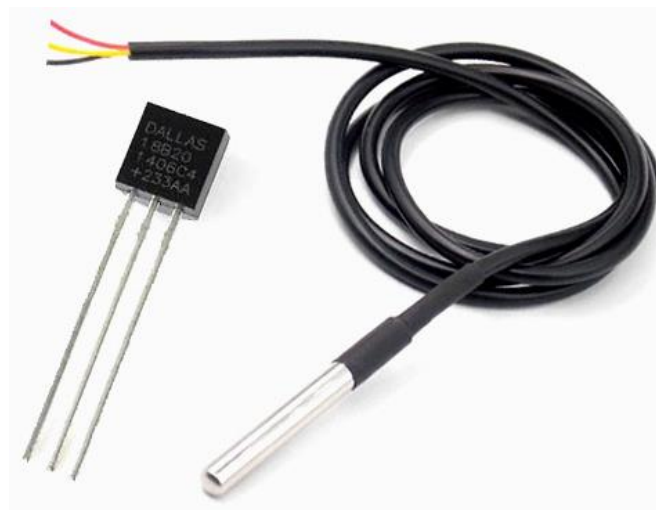


Figura 41. Sensores de temperatura DS18B20

El sensor contiene:

- Sensor de temperatura digital programable
- Se comunica mediante el método de 1 cable
- Voltaje de funcionamiento: 3V a 5V
- Rango de temperatura: -55 ° C a + 125 ° C
- Precisión: $\pm 0,5$ ° C
- Resolución de salida: de 9 bits a 12 bits (programable)
- La dirección única de 64 bits permite la multiplexación
- Tiempo de conversión: 750 ms a 12 bits
- Opciones de alarma programables
- Disponible como To-92, SOP e incluso como sensor impermeable

La medición más importante del equipo es la medición de radiación solar, para este punto se utilizó un sensor PAR provisto por el Laboratorio LEA de UTN-FRP.



Figura 42. Sensor de radiación solar PAR

El sensor PAR mide la radiación fotosintéticamente activa, que se define como la radiación comprendida en el intervalo de 400nm a 700nm. Esta radiación es aquella capaz de producir actividad fotosintética en plantas y microorganismos.

La respuesta de este sensor es proporcional a la irradiancia recibida, es decir que para medir la radiación solar debemos digitalizar la tensión de salida del radiómetro.

En particular, este sensor posee una constante de $10,68 \mu\text{V} / \text{W} / \text{m}^2$, es decir que para una irradiancia de unos $900 \text{ W} / \text{m}^2$ deberemos digitalizar una señal de unos 9.61 mV

B. Circuitos propuestos y su funcionamiento

Este sistema tiene como funcionalidad poder informarnos de las siguientes temperaturas de manejo:

- Temperatura de trabajo del panel fotovoltaico 1
- Temperatura de trabajo del panel fotovoltaico 2
- Temperatura ambiente
- Temperatura del equipo

Es importante conocer el valor de las temperaturas de trabajo, ya que es uno de los parámetros que influyen en el rendimiento de un módulo fotovoltaico, es decir si nosotros estamos comparando dos paneles solares es importante que la temperatura de ambos paneles sea similar, ya que temperaturas muy diferentes entregan resultados dispares que carecen de sentido en la comparación. También podemos mencionar que otro de los puntos a interes es ensayar los módulos fotovoltaicos con distintas temperaturas de exposición, aquí es donde interviene el uso del sensor de temperatura ambiente, mientras que el sensor de temperatura del equipo sirve para saber si nuestro sistema presenta alguna falla o está expuesto a demasiado calor, lo que a largo plazo dañaría nuestro sistema.

Para esta tarea se implementó en nuestro circuito el uso de sensores DS18B20, el cual es un sensor digital de temperatura que utiliza comunicación serial mediante protocolo One-Wire, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus. Esto se puede ver en el esquemático mostrado a continuación.

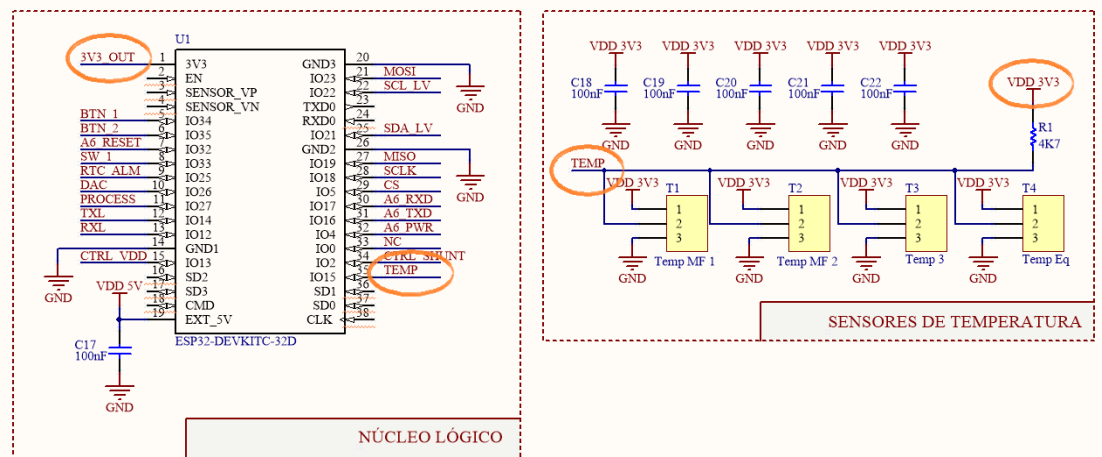


Figura 43. Esquema eléctrico bus de datos One-Wire

Este bus One-Wire necesita para su funcionamiento una resistencia Pull-Up en función de a distancia del sensor, la misma queda definida por la siguiente tabla:

Resistencia Pull-Up	Distancia del sensor al Microcontrolador
4,7 kΩ	0 m a 5 m
3,3 kΩ	5 m a 10 m

2,2 k Ω	10 m a 20 m
1,2 k Ω	20 m a 50 m

Tabla 3. Resistencia según distancia sensor - microcontrolador

En nuestro caso implementamos un resistor de 4.7K Ω con denominación R1.

Cada sensor tiene una dirección única de 64 bits establecida de fábrica, esta dirección sirve para identificar al dispositivo con el que se está comunicando, ya que dentro del bus 1-wire nosotros tenemos 4 sensores de temperatura. Para acceder a uno de ellos necesitamos primero tener comunicación con el sensor que queremos evaluar mediante esta dirección.

Las direcciones de nuestros sensores son:

- Temperatura del panel fotovoltaico 1:
0x28, 0x23, 0xA7, 0x41, 0x4B, 0x18, 0x07, 0xAD
- Temperatura del panel fotovoltaico 2:
0x28, 0x32, 0xAF, 0x52, 0x2E, 0x24, 0x11, 0xD1
- Temperatura ambiente:
0x28, 0x25, 0xE9, 0x33, 0x1A, 0x13, 0x01, 0x6D
- Temperatura del equipo:
0x28, 0x59, 0xBC, 0x22, 0x65, 0x20, 0x01, 0xB0

Estos sensores poseen la propiedad de ajustar la resolución de conversión de los mismos, cabe destacar que a mayor resolución mayor el tiempo de conversión. Para nuestro dispositivo los sensores son ajustados cada vez que el equipo se enciende a la mayor resolución posible informada en la siguiente tabla.

Resolución de conversión	Discriminacion de temperatura
9 bit	0,5°C
10 bit	0,25°C
11 bit	0,125°C
12 bit	0,0625°C

Tabla 4. Resolución sensor de temperatura

Por último, podemos mencionar el rango de temperatura y su error. Este sensor es posible medir temperatura desde los -55°C hasta los 125°C, para valores entre -10°C y 85°C el sensor posee un error de $\pm 0,5^\circ\text{C}$ y para resto de temperaturas entre -55°C y 125°C el error es de $\pm 2^\circ\text{C}$.



Figura 44. Error del sensor de temperatura

Para el funcionamiento del sensor de radiación solo tenemos que configurar el módulo ADC con el PGA en la escala más baja posible y digitalizar el mismo en modo diferencial como las variables de tensión y corrientes extraídas de los módulos fotovoltaicos.

2.1.4 Placa base.

La placa base se diseñó con el fin de contener todos los módulos utilizados y mencionados anteriormente, está pensada de tal manera que cualquier componente sea fácilmente reemplazable.

Como principal ventaja, esta placa derivó en estabilidad en el funcionamiento a la hora de realizar las diferentes pruebas.

A. Investigación

Para la fabricación de la placa, se realizó una investigación en el mercado local de empresas que estén destinadas a la realización de este servicio, por una cuestión económica y de aprendizaje se optó por utilizar una empresa extranjera para la elaboración del proyecto. Este proceso aportó en el aprendizaje y en la obtención de experiencia a la hora de realizar un desarrollo particular, efectuar un pedido en el extranjero, ejecutar una compra internacional, llevar cabo la exportación de un diseño local en archivos Gerber genéricos para la fabricación del producto en industria, llevar a la práctica la comunicación y logística en este tipo de trabajos.

Como resultado se obtuvo una placa base de muy buena calidad y con las características mencionadas a continuación.

- Tamaño: 200mm x 150mm
- Doble face
- Tecnología Through-hole: Si
- Mascara antisoldante: Sí color rojo

- Serigrafía de componentes: Sí color blanco
- Material: FR4
- Espesor: 1.6mm
- Terminación: HASL con plomo

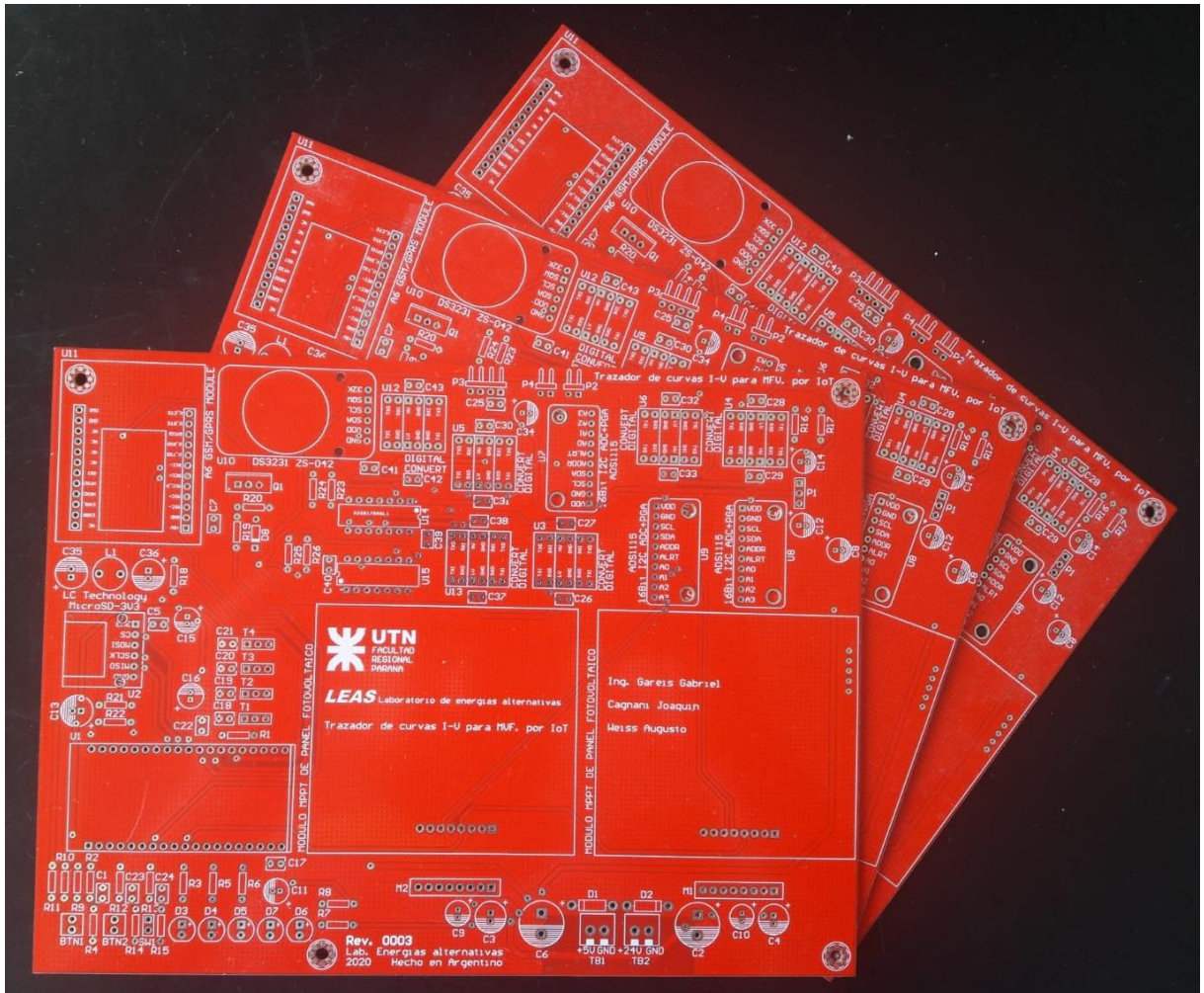


Figura 45. Placa base desarrollada

B. Ejecución del proceso

En primer lugar, se realizó el diseño del esquemático de las distintas partes ya mencionadas, luego se formalizó el circuito completo en una placa base con el diseño del siguiente PCB.

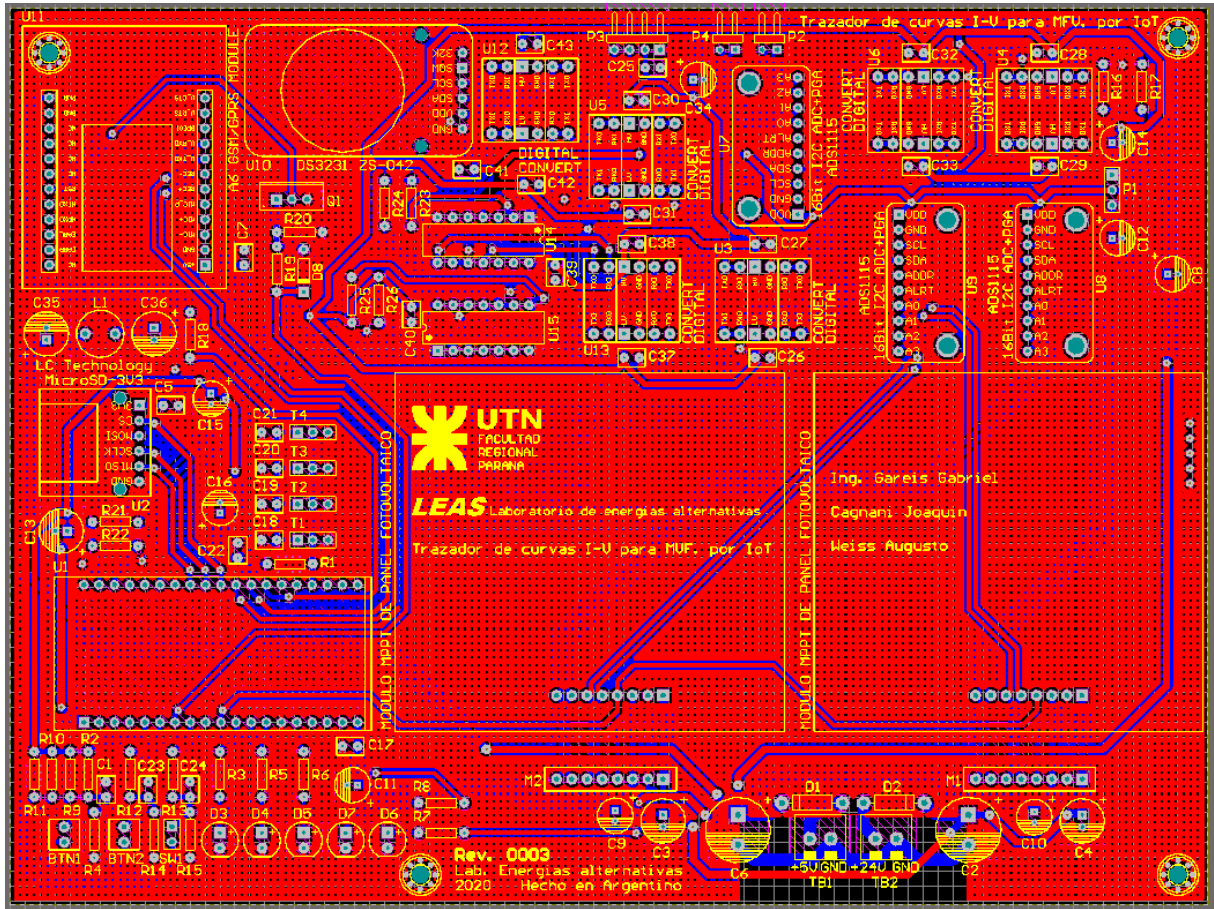


Figura 46. Diseño de la placa base - vista superior

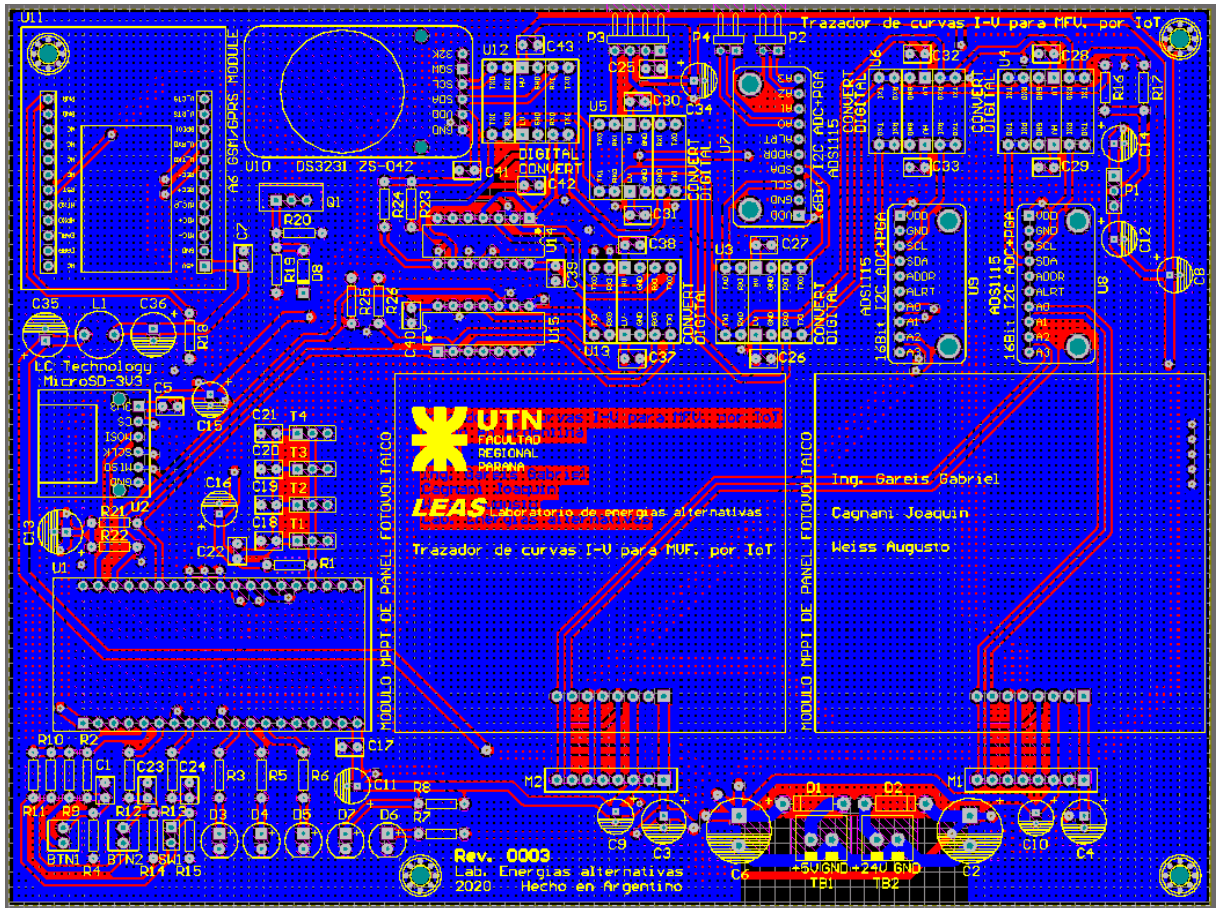


Figura 47. Diseño de la placa base - vista inferior

Para verificar el tamaño de la placa y la conformidad del montaje de los componentes en ella, se corroboró mediante modelado 3D la disposición de cada componente a colocar.

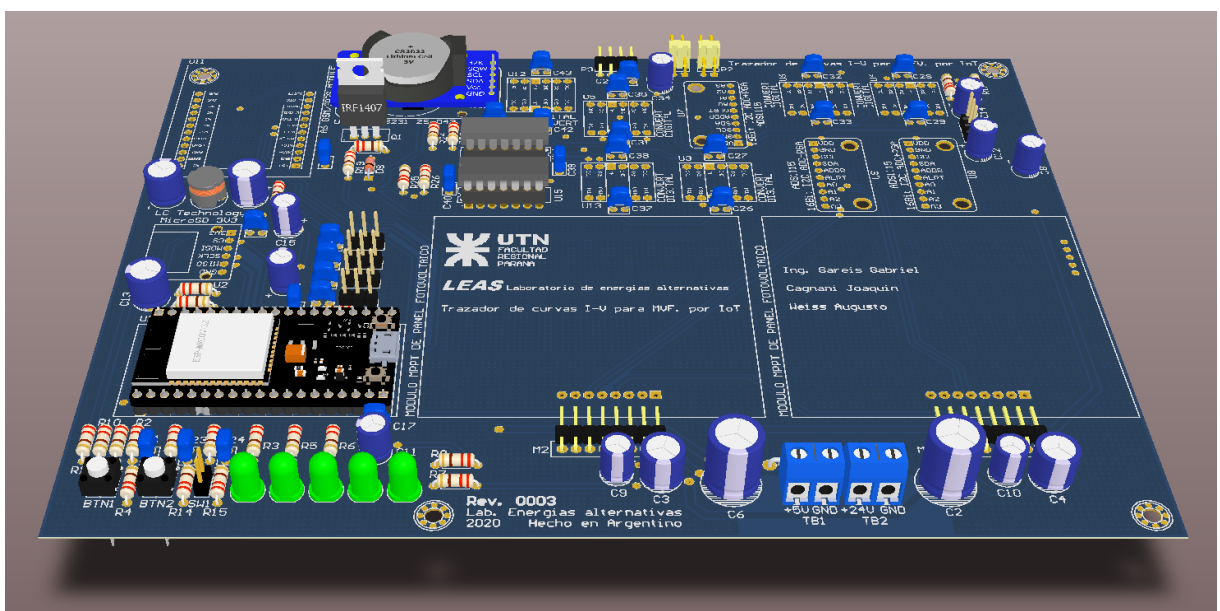


Figura 48. Diseño de la placa base - vista 3d

Por último, una vez que el desarrollo fue finalizado, se procedió a la exportación del proyecto en formato de archivos gerber, para la fabricación de esta placa.

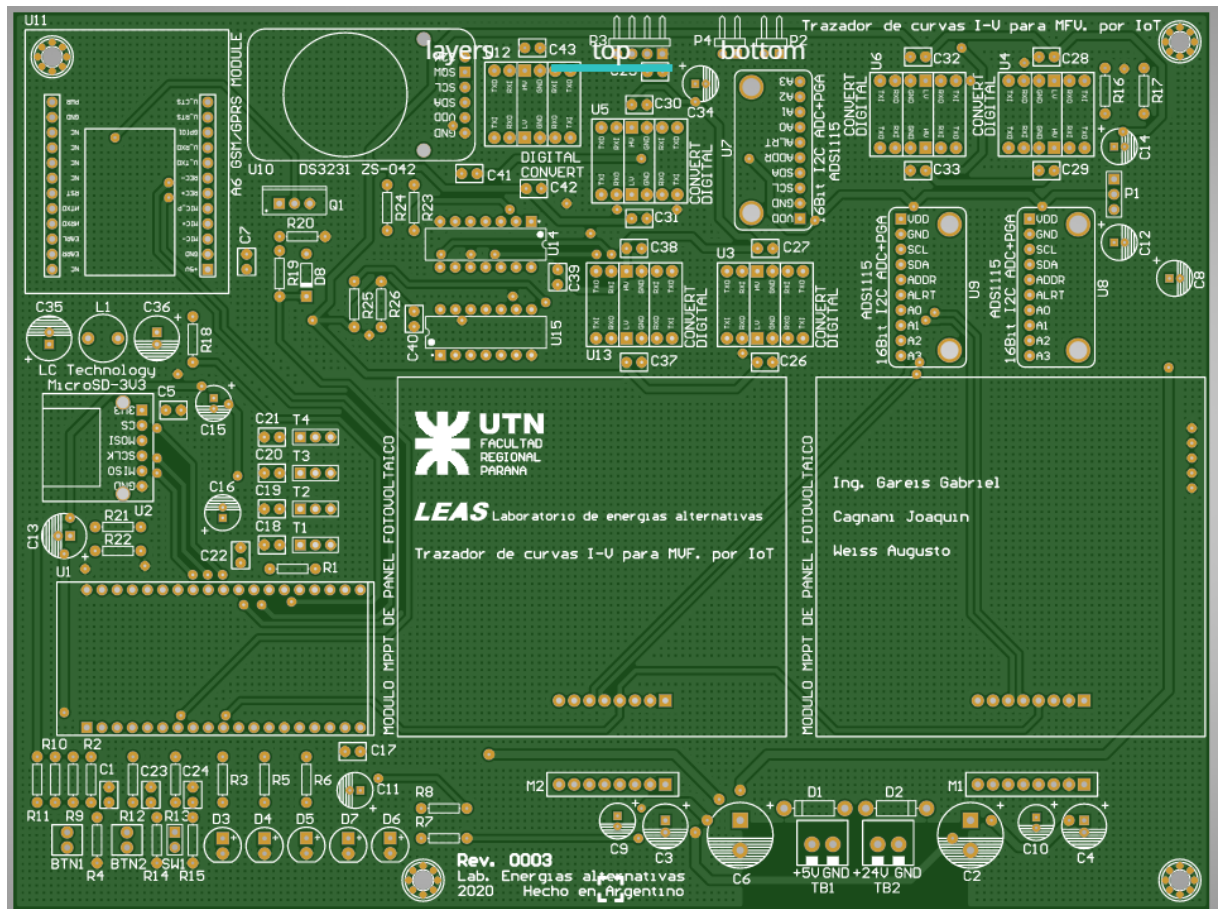


Figura 49. Diseño de la placa base - formato gerber

C. Análisis de funcionamientos

El funcionamiento de la placa logró gran estabilidad al sistema y permitió la mejora del mismo en distintas partes del circuito, estos cambios ya fueron agregados al esquemático general y a los distintos módulos mencionados anteriormente, la placa base terminó montada con sus modificaciones como se muestra a continuación.

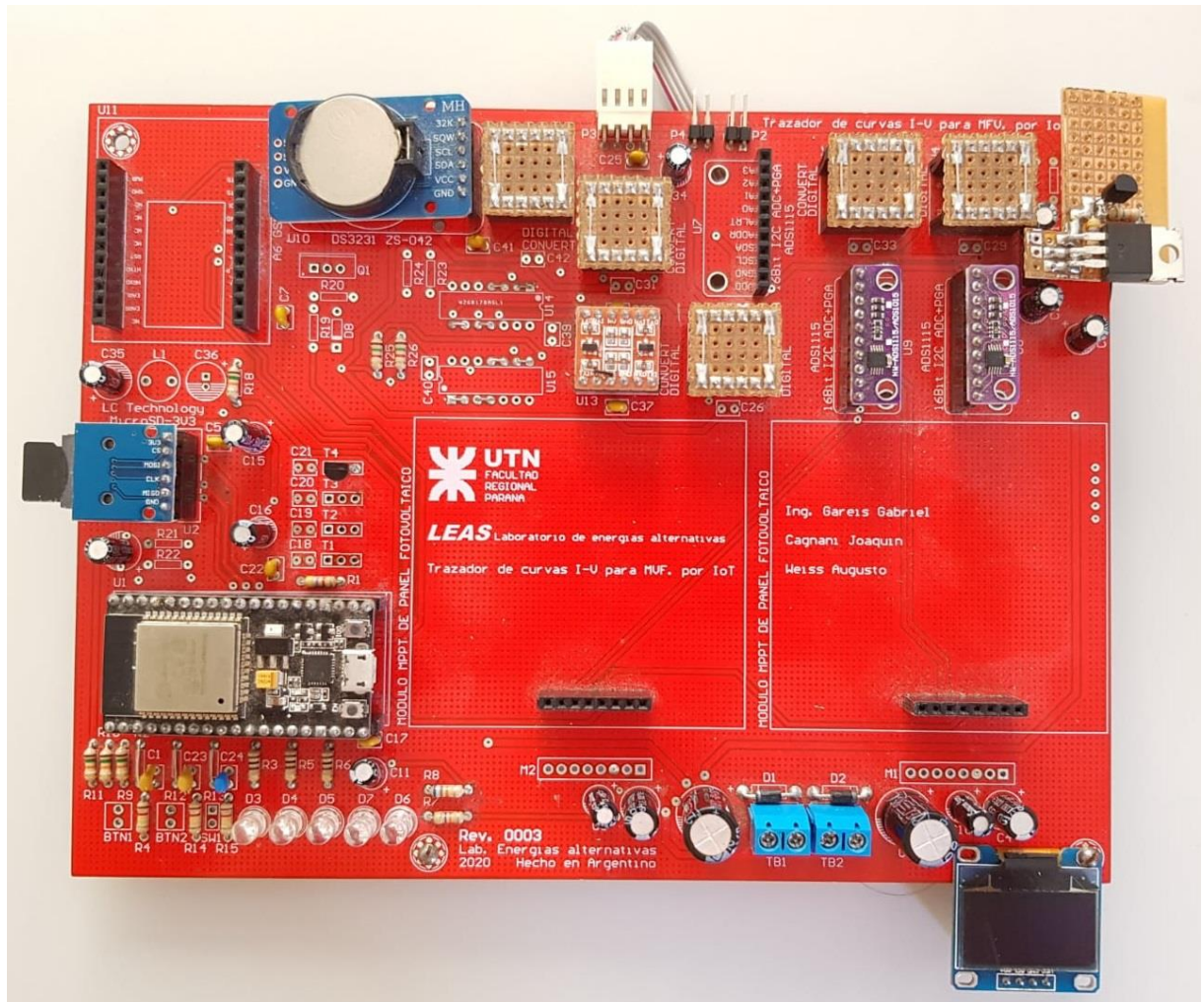


Figura 50. Placa base en implementación

2.1.5 Fuente de alimentación.

Para la alimentación del dispositivo se investigó la disponibilidad en el mercado de distintos módulos de reguladores de tensión con distintas características, cada bloque necesita determinadas cualidades que hacen al funcionamiento del dispositivo, todas ellas serán explicadas con la descripción del circuito.

A. Investigación de componentes

Para la fuente de alimentación la conexión al panel es uno de los puntos más importantes, ya que el regulador principal que proporcione tensión tanto a la salida como al cargador de baterías tiene que soportar hasta 50 V de entrada como mínimo, para este trabajo se eligió un regulador step-down DC-DC implementado con un LM2576HV, en donde se ingresará con una tensión de 5V a 60V DC y permite regular la tensión de salida en un rango de 1.25V a 26V.

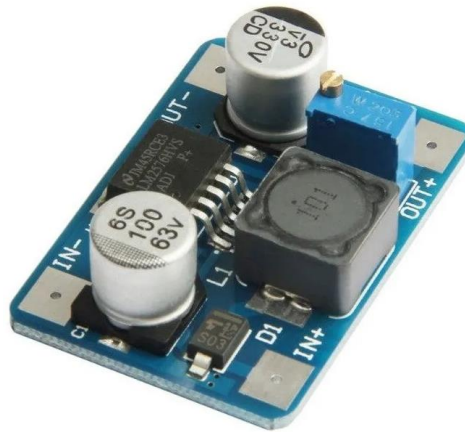


Figura 51. Modulo step-down LM2576HV

La elección de este regulador se da exclusivamente por la tensión de entrada que soporta, eficiencia del mismo y disponibilidad en el mercado, la gran mayoría de los reguladores que se encontraron poseen tensión de entrada de 40V. Las características del regulador son las siguientes.

Características:

- Eficiencia de conversión: hasta 92%
- Frecuencia de conmutación: 150kHz
- Voltaje de entrada: 5 - 60V DC
- Voltaje de salida: 1.25 - 26V DC
- Corriente de salida: nominal 2A, (máximo 3A con disipador de calor adicional)
- Ondulación de salida: <30mA
- Rectificador: rectificación no sincrónica
- Propiedades del módulo: módulo reductor no aislado (regulador buck)
- Protección contra cortocircuitos: limitación de corriente, desde la recuperación
- Temperatura de funcionamiento: -40 a 85 ° C
- Aumento de temperatura a plena carga: 40 ° C
- Regulación de carga: $\pm 0.5\%$
- Regulación de voltaje: $\pm 0.5\%$
- Velocidad de respuesta dinámica: 5% 200 μ S
- Dimensiones: 44 x 30 x 14 mm
- Peso: 14g

El sistema posee batería para su uso cuando el panel no pueda entregar alimentación suficiente al sistema y también con el objeto de poder alimentar al dispositivo cuando se necesite separar al panel solar del sistema para su evaluación. Para ello se utilizaron 2 baterías de litio modelo 18650 conectadas en serie, estas tienen por carácter ser muy ligeras, con una elevada capacidad energética y resistencia a la descarga.



Figura 52. Baterías de litio utilizadas

Las características de esta batería de litio son las mostradas a continuación:

- Batería Recargable
- Tensión nominal de 3.7V
- Capacidad: 4000mAh
- Tecnología Li-ion
- Largo aproximado 6.5cm

Para la carga de las baterías mencionadas anteriormente, debido a que las mismas están conectadas en serie en nuestro sistema, es necesario constar de un cargador de batería balanceado para mantener a dichas baterías a misma tensión de carga. Para realizar esto último se utilizó un Módulo Bms Protector Para 2 Baterías 18650 de 2 Celdas 8A, HX-2S-A10.

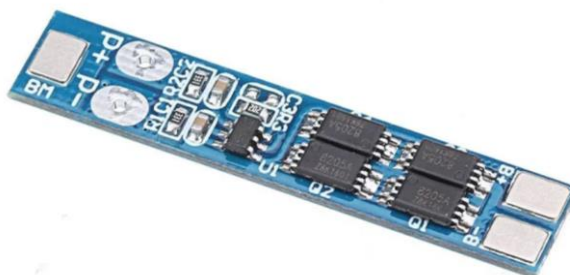


Figura 53. Módulo BMS cargador de baterías

Las características de este módulo son las mostradas a continuación:

- Protección contra cortocircuito
- Protección contra sobrecarga
- Protección contra sobredescarga
- Protección contra sobrecorriente
- Voltaje de entrada: 7.4V a 8.4V
- Rango de protección contra sobrecarga por celda: 4.25V a 4.35V +/- 0.05V
- Rango de protección contra sobredescarga por celda: Típico: 2.30V a 3.0V +/- 0.05V
- Corriente de trabajo máxima (Carga y descarga): 8A
- Protección contra sobrecorriente: 10A
- Dimensiones: 41mm x 8mm

La fuente de alimentación posee una salida de 5V necesaria para alimentar a todo el sistema, como la tensión entregada por las baterías ronda alrededor de los 7V a 8V utilizamos para este trabajo un módulo regulador step-down DC-DC implementado con un LM2596 de alta eficiencia, muy necesaria ya que el dispositivo completo conlleva un consumo elevado.



Figura 54. Módulo step-down LM2596

Las características de este módulo son mencionadas a continuación:

- Voltaje de entrada: 4.0V a 40V
- Voltaje de salida: 1.23V a 35V Ajustable (el voltaje de entrada debe tener al menos 1.5V más que la salida).
- Corriente máxima de salida: 2A (Es posible llegar a 3A si se utiliza el disipador correspondiente).
- Potencia de salida: 50-70W, utilizar disipador
- Rectificador: rectificación no sincrónica
- Propiedades del módulo: módulo reductor no aislado (regulador buck)
- Eficiencia de conversión: 92%

- Frecuencia de Trabajo: 150 KHz
- Ripple en la salida: 30mV (máx.)
- Protección de cortocircuito y sobre-temperatura.
- Dimensiones: 43mm x 21mm

La fuente de alimentación posee una salida de 12V necesaria para el sistema de medida implementado con el módulo MPPT, utilizamos para este trabajo un módulo regulador step-up DC-DC implementado con un MT3608 de alta eficiencia.



Figura 55. Módulo step-up MT3608

Las características del módulo anterior son las siguientes:

- Tensión de entrada: 2V a 24V
- Salida de tensión: 5V a 28V ajustable
- Corriente máxima de entrada: 2A
- Corriente máxima de salida: 2A
- Rectificador: rectificación no sincrónica
- Propiedades del módulo: módulo reductor no aislado (regulador boost)

B. Circuito propuesto

El circuito propuesto en base a los módulos estudiados es el mostrado a continuación.

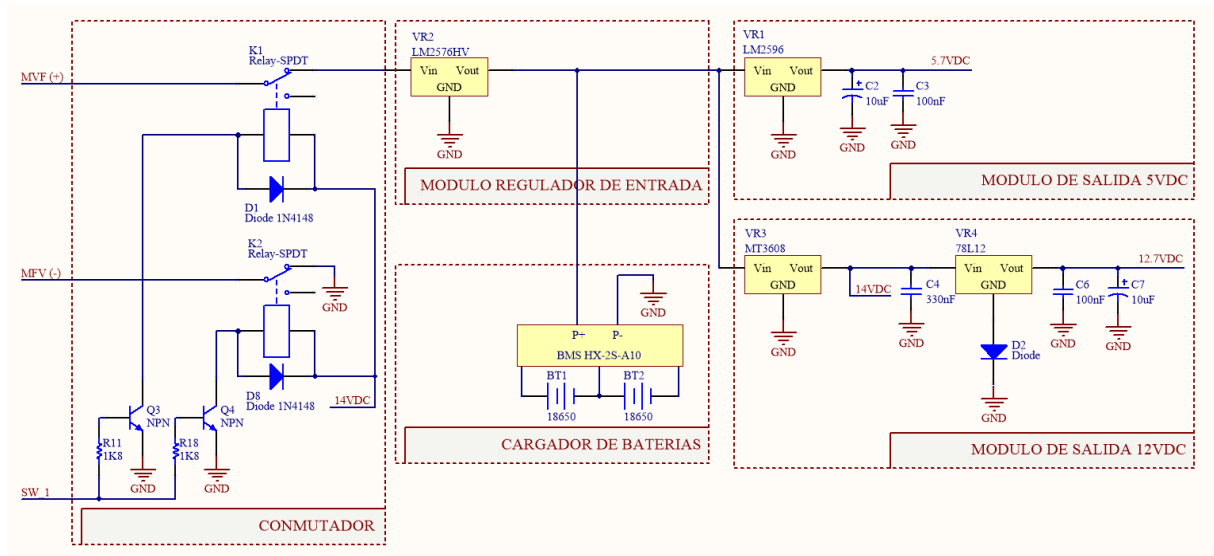


Figura 56. Esquema de la fuente de alimentación

En primera medida la fuente de alimentación arranca en el “módulo regulador de entrada” implementado con un LM2576HV, este permite una tensión de alimentación máxima de 60V dejando al dispositivo conectar cualquier fuente de alimentación externa que esté por debajo del valor mencionado. Como la fuente se alimenta del mismo panel, el uso de este módulo nos permite conectar distintas variedades de módulos fotovoltaicos con distintas tensiones nominales, así como también una fuente externa.

En la entrada al LM2576HV, se encuentran 2 relay que son los encargados de desconectar la entrada de los paneles solares, el objeto de esto es que cuando se busque analizar un panel solar, el mismo se desconecte de la fuente de alimentación dejando al equipo suministrado a baterías y poder ensayar el/los paneles solares en estado libre logrando una medición en vacío. Estos relés son controlados por el CPU del dispositivo y poseen una lógica normal cerrada, es decir si el equipo se queda sin baterías, los mismos paneles solares podrán alimentar al dispositivo y cargar las baterías cuando los relays estén no accionados.

La etapa “módulo regulador de entrada” tiene como funcionamiento poder otorgar la alimentación al cargador de baterías, así como también alimentar los reguladores de salida. El módulo “cargador de batería” implementado con el BMS HX-2S-A10, se alimenta directamente del “módulo regulador de entrada”, este componente tiene por objeto cargar de forma simétrica 2 baterías de litio 18650 y proporcionar las protecciones antes mencionadas hacia las mismas.

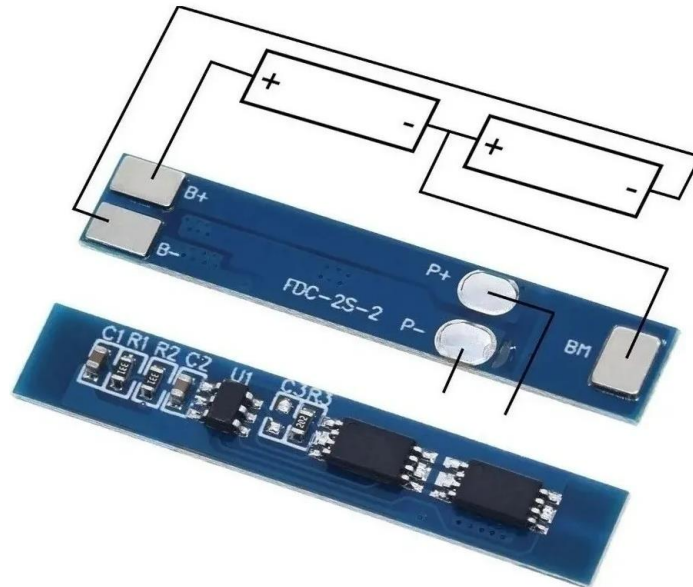


Figura 57. Cargador de baterías de litio

La tensión de salida del cargador de batería es extraída desde los pines P+ y P- también encargados de inyectar la tensión para la carga de las baterías 18650. Esto significa que el módulo en sí posee un discriminador en su interior para saber si hay o no alimentación externa, permitiendo conectar este componente directamente a los módulos de salida sin ningún tipo de conexión extra.

El módulo de salida de 5V tiene por objeto ser de alta eficiencia ya que alimentan al CPU del equipo y todos sus periféricos, es por esto que solo está implementado con un regulador switching, además de esto el módulo está pensado para poder tener una alimentación variable de entrada ya sea por el primer regulador o por la variación de las descargas de la batería.

El módulo de salida de 12V tiene por objeto ser estable, ya que es el encargado de alimentar el bloque MPPT, es por esto que su implementación se realizó con un módulo que eleva la tensión de salida hasta 15V y luego se reduce la misma con un regulador lineal 78L12 a 12V, de esta forma cualquier fluctuación en el elevador de tensión es filtrada por el regulador de salida, el cual se buscó que tenga un valor de PSRR aceptable.

C. Implementación

La implementación de estos módulos se realizó sobre una placa preperforada quedando de la siguiente manera

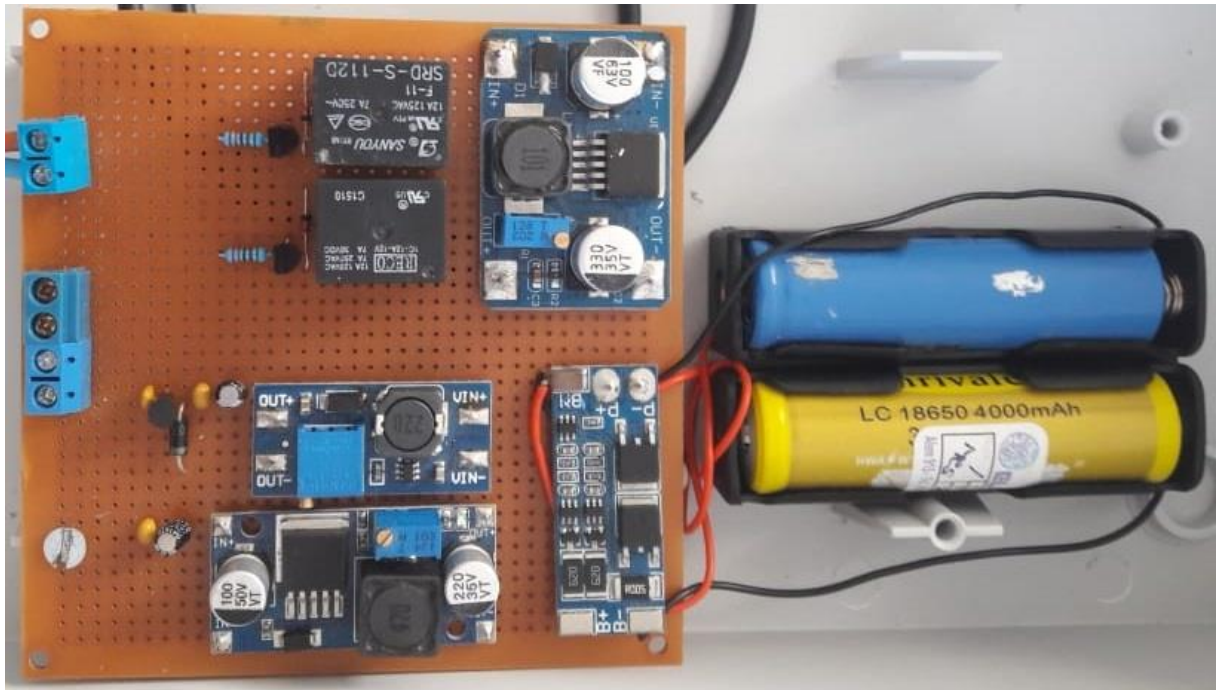


Figura 58. Fuente de alimentación implementada

D. Análisis de funcionamientos

En primera medida se implementó un sistema que posee una sola batería 18650 con un cargador TP4056, desde el punto de vista circuital, por un lado, se conectaba el “módulo regulador de entrada” y por otro lado se extraía la alimentación de las baterías hacia nuestro dispositivo, este tipo de conexión derivó en el uso de un discriminador. La etapa “módulo regulador de entrada” tiene como funcionamiento poder otorgar la alimentación al cargador de baterías, así como también alimentar los reguladores de salida, es por esto que se le colocó un bloque denominado “Discriminador”, el cual fue implementado con un Mosfet canal P y tiene por lógica de funcionamiento discriminar cuando haya tensión proveniente por el módulo principal. La idea fue detectar si hay alimentación en la primera parte de este dispositivo y no permitir la conexión entre las baterías y los módulos de salida, así esta primera parte no solo alimenta a los módulos de salida sino también carga las baterías del equipo, podemos resumir que el sistema queda energizado por alimentación externa. En caso de no existir alimentación en la primera etapa de este circuito, el Mosfet conmuta y conecta las baterías a los módulos de salida, podemos resumir que el sistema queda energizado por baterías.

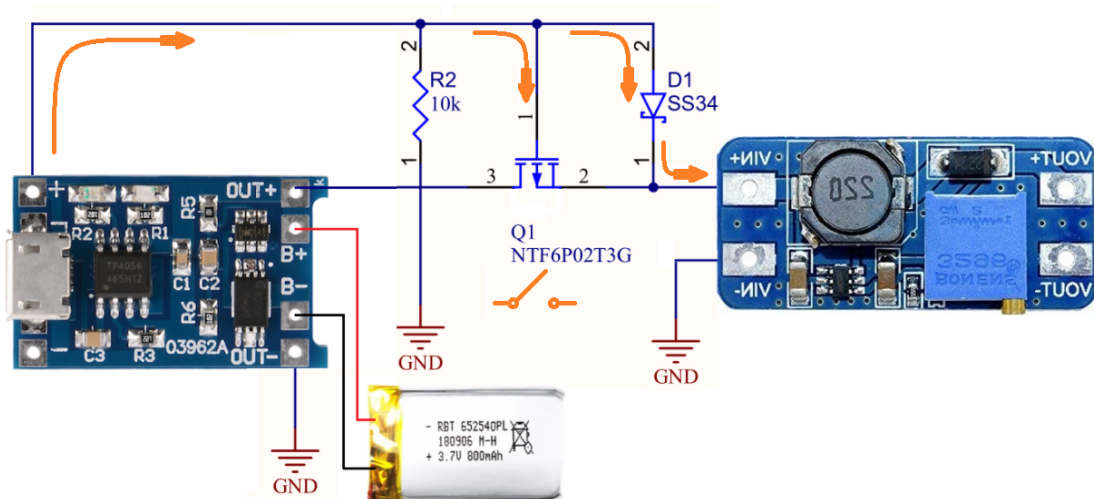


Figura 59. Diseño de discriminador de fuente de alimentación

El diseño funcionó eléctricamente muy bien pero no consiguió tener una eficiencia en cuanto a tiempo de rendimiento. El sistema solo funcionaba 6 horas prendido en “Modo Online” y no alcanzaba a funcionar 24 horas de corrido en “Modo Sleep” con las 10 alarmas activas. Debido a lo explicado con anterioridad el diseño de la fuente de alimentación se modificó para colocar dos baterías de litio en serie cargados con el módulo BMS nombrado con anterioridad y removiendo del esquema el módulo discriminador.

El funcionamiento de la fuente de alimentación fue satisfactorio, consiguiendo un no calentamiento de los componentes en ella, un uso protegido de las baterías y una duración de más de 24 horas si el sistema se mantenía en Modo Sleep con las 10 alarmas en funcionamiento. Si el sistema se lo deja en Modo Online alimentado sólo a baterías se consigue una duración de 20 horas aproximadas de funcionamiento.

2.1.6 Dispositivo en modo AP, módulo web y modos de funcionamiento Online y Sleep.

Nuestro dispositivo posee dos modos de funcionamiento llamados “Modo Online” y “Modo Sleep”, el primero sirve para brindar a un usuario particular el uso del equipo en forma local, mientras que el segundo está destinado al uso global donde más de un usuario puede hacer uso del dispositivo en forma remota.

El “Modo Online” tiene por funcionalidad principal poder configurar el equipo para luego ser dejado en “Modo Sleep” por largos periodos de tiempo, pero también le sirve al usuario para realizar tareas de testeo de las distintas variables que el equipo maneja y ensayo de los módulos fotovoltaicos levantando una curva I-V y P-V de manera instantánea con características dinámicas configuradas por el usuario.

Por defecto el equipo ingresa siempre en “Modo Sleep”, es por ello que para entrar en “Modo Online” el usuario debe encender el equipo y presionar el pulsador BTN1 de nuestro circuito cuando se escriba en la pantalla la descripción “Iniciando” y no deberá soltarlo hasta que en la pantalla aparezca la descripción “Modo Online”. Es indistinto si el usuario presiona el botón BTN1 ni bien enciende el dispositivo o cuando aparezca la leyenda antes mencionada, si es importante que lo mantenga presionado hasta que el dispositivo informe el modo de operación.

A. Funcionamiento del dispositivo en “Modo Online”

Cuando el dispositivo ingresa en “Modo Online”, nuestro NodeMCU no intentara conectarse a ninguna red disponible, se configura en modo Access Point (AP) y brindara al usuario un punto de acceso inalámbrico vía WiFi con las siguientes características:

- SSID: IV_Tracert_001
- PSW: limon001

Tanto el SSID y el PSW no serán configurables por el usuario y el número “001” varía en función del número del equipo.

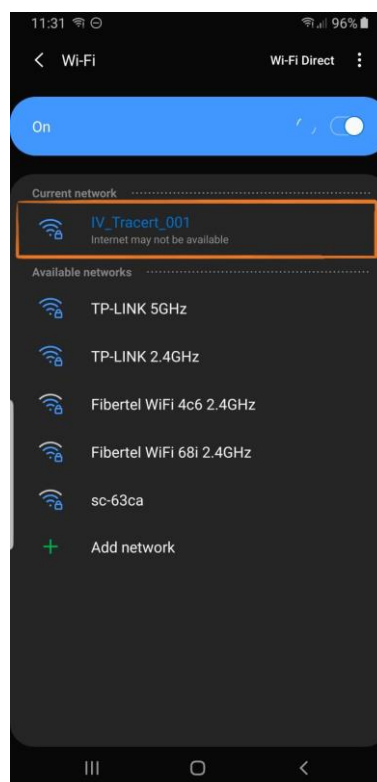


Figura 60. Dispositivo movil conectado a red WiFi del ESP32

Una vez que el usuario se conecte al equipo con algún aparato (Celular, PC, Tablet, etc), el dispositivo funcionará como Web Server Asíncrono ofreciendo al usuario un sitio Web accesible por cualquier navegador al que llamaremos “Módulo Web”. Para acceder al

Módulo Web el usuario deberá buscar al dispositivo con la dirección ip 192.168.120.3 dentro del navegador.

Cabe destacar que este dispositivo posee una ip fija que no será configurable por el usuario, por otro lado, mencionamos también que la comunicación entre el ESP32 y el Módulo Web se realizará a través de lo que se conoce como un WebSocket.

Una vez que el usuario intenta conectarse al dispositivo, se abrirá un sitio Web informando que se realizó una conexión correcta y el CPU enviará al usuario toda la información guardada en la memoria EEPROM de este, la cual podemos mencionar:

- SSID a la cual se conectará el dispositivo cuando esté en “Modo Sleep”
- PSW a la cual se conectará el dispositivo cuando esté en “Modo Sleep”
- Alarmas almacenadas en memoria.

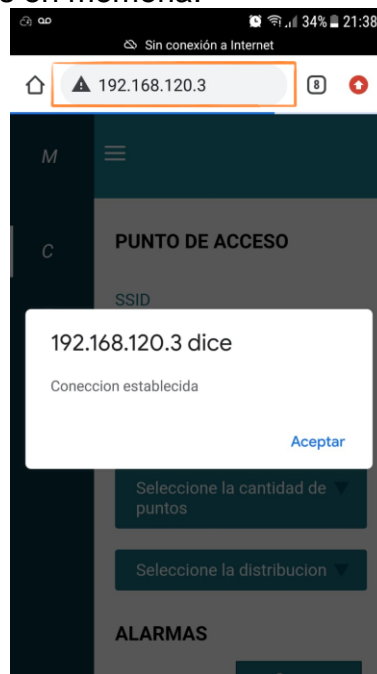


Figura 61. Mensaje de conexión establecida

B. Configuración del dispositivo

Para configurar el dispositivo basta con elegir dentro del sitio Web el SSID y el password al que el equipo se debe conectar cuando esté trabajando en “Modo Sleep”, elegir la resolución que deseamos tener en la curva característica y su distribución (Número de puntos de la gráfica y en qué zona ubicarlos) y por último ver qué alarmas tiene el dispositivo guardadas en su memoria, pudiendo editarlas o borrarlas.

Finalizada la estructura deseada solo debemos dar click al botón cargar para enviar la configuración al CPU.

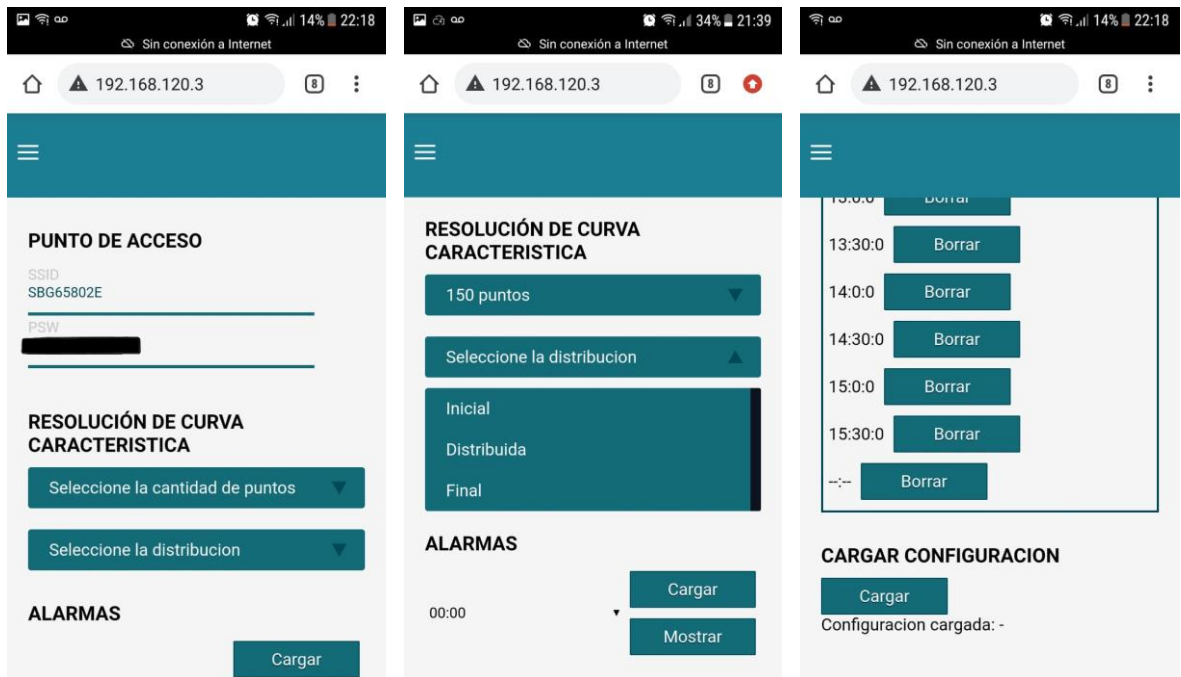


Figura 62. Aplicación web levantada por ESP32

C. Testeo de variables y sensado del panel solar

Es posible para un uso particular del dispositivo poder ensayar el panel solar al que estamos conectados, esto nos sirve para evaluar malas instalaciones o posibles daños en el módulo fotovoltaico, así como también garantizar una buena instalación de este aparato.

Para testear las variables analógicas, asociadas al dispositivo solo tenemos que entrar a la pestaña llamada Testeo dentro del sitio Web y dar click en el botón Test.



Figura 63. Sección de test por aplicación web



Figura 64. Test realizado desde aplicación web

Para realizar el ensayo de los módulos fotovoltaicos es necesario informar la configuración del escaneo seteando la resolución de la curva I-V y su distribución entrando en la pestaña llamada Escanear dentro del sitio Web y dar click en el botón Escanear.



Figura 65. Sección Escanear por aplicación web.

La resolución de una curva I-V se define como el número de puntos que la misma posee, mientras que la distribución de la curva I-V hace referencia a la ubicación de dichos puntos. Así podemos tener por ejemplo una curva de 100 puntos distribuida donde la gráfica tendrá 100 puntos desde su valor en vacío hasta su valor en cortocircuito, otro ejemplo sería tener una gráfica de 20 puntos al final, donde el dispositivo sólo evaluará los

20 puntos más cercanos a la condición de cortocircuito del panel, con el fin de estudiar bien cómo se comporta este en esta condición.

La logica de programacion del sitio Web es la siguiente:

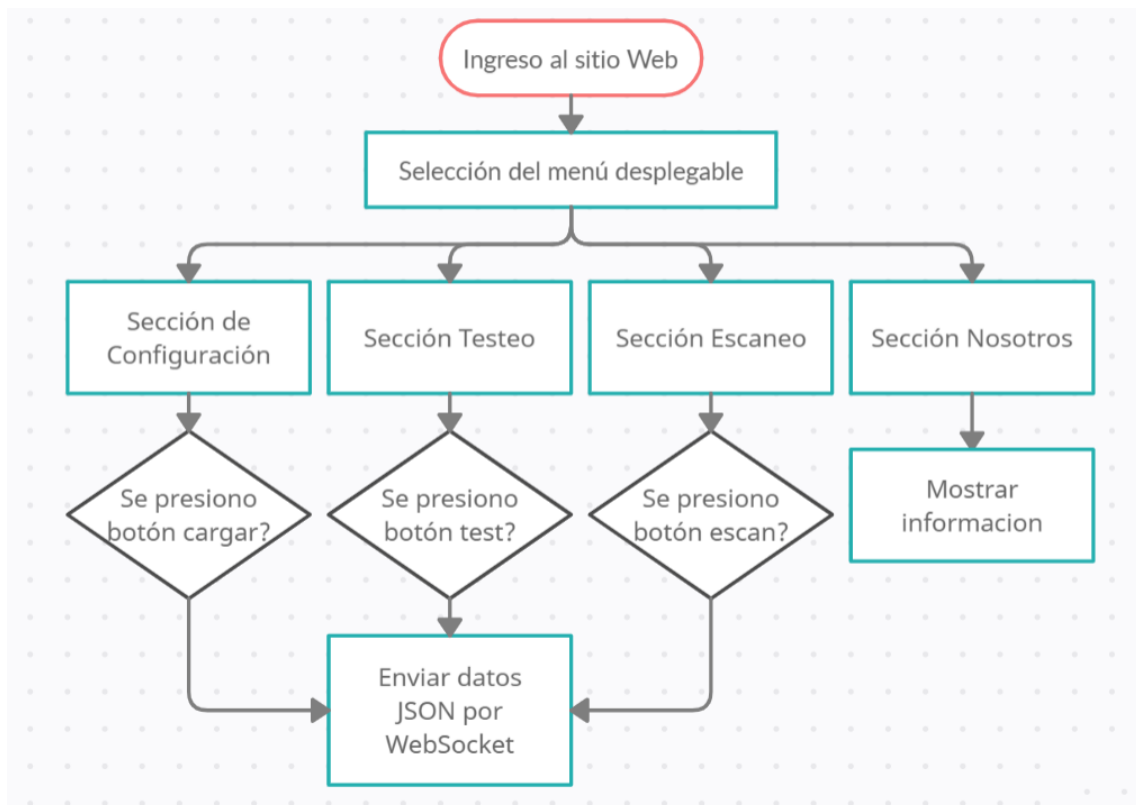


Figura 66. Lógica sitio web en modo local

2.1.7 Dispositivo en modo STA y broker MQTT.

El dispositivo tiene como funcionamiento ordinario un modo de trabajo denominado “Modo Sleep”, aquí el NodeMCU se configura en modo Station (ST) e intenta conectarse a una red WiFi con SSID y Password previamente configuradas en “Modo Online”.

Terminada la configuración del CPU para este modo de trabajo, el dispositivo discrimina si fue despertado manualmente, por una alarma previamente configurada o por un error del sistema. Si el dispositivo fue despertado por una alarma el equipo procede a realizar el sensado de una curva I-V, el almacenamiento de información y el envío de la misma al Servidor Web que se diseñó para su trabajo, es decir el sistema Web que será desarrollado en la sección 2.1.8.

El dispositivo se comunica con el Sistema Web a través del protocolo MQTT. MQTT es un protocolo de comunicación enfocado a la conectividad máquina a máquina (M2M), basado en una trama de datos de mensajería con un enfoque del tipo publicación/suscripción, a diferencia del protocolo HTTP que posee un enfoque de

petición/respuesta. Uno de sus puntos más fuertes es ser extremadamente simple y ligero.

Para la implementación de este protocolo en el envío de información al servidor Web se utilizó para nuestro sistema un broker MQTT gratuito. El broker es el servidor que acepta mensajes publicados por clientes y los difunde entre los clientes suscritos. Para nuestro caso, tenemos al dispositivo y al sistema Web suscritos como clientes al broker donde cada uno publicará un mensaje en caso de enviar información al otro.

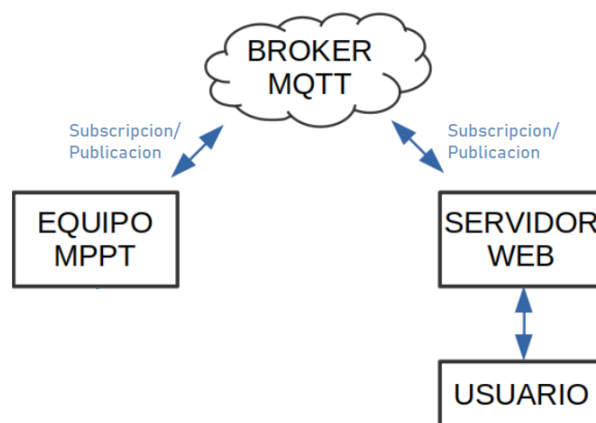


Figura 67. Diagrama general de conexión remota.

Para la mensajería entre el dispositivo y el Sistema Web se utiliza el empleo de tópicos específicos, el tópico o “Topic” en MQTT es una cadena de texto UTF-8, de longitud máxima de 65536 caracteres (aunque lo normal es que sea mucho menor), que distingue entre mayúsculas y minúsculas y se utiliza para el empaquetado de información en base a niveles de incumbencia separados entre sí por una barra inclinada ‘/’.

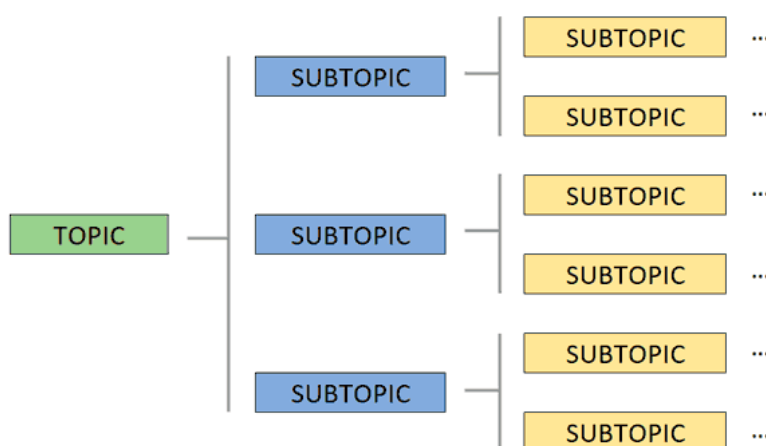


Figura 68. Estructura de tópicos MQTT

En nuestro caso la comunicación entre el dispositivo y el servidor Web tendrá la siguiente estructura de tópicos desde el punto de vista del dispositivo:

Dispositivo:

- Tópico Raiz: 2HVwqw4RkoBUYnY
- Nombre de usuario del dispositivo: ivt001
- ID del dispositivo en modo cliente: random

Suscripción:

Tópico	Tipo de mensaje	Descripción
TopicRaiz/Nombre/sens/temp	dato (String)	Lea descripcion 1.1
TopicRaiz/Nombre/sens/mfv	dato (String)	Lea descripcion 1.2
TopicRaiz/Nombre/date/hora	dato (String)	Lea descripcion 1.3
TopicRaiz/Nombre/date/fecha	dato (String)	Lea descripcion 1.4
TopicRaiz/Nombre/alrm/set	dato (String)	Lea descripcion 1.5
TopicRaiz/Nombre/alrm/del	dato (String)	Lea descripcion 1.6
TopicRaiz/Nombre/feat	dato (String)	Lea descripcion 1.7

Tabla 5. Estructura tópicos MQTT de suscripción

Descripción 1.1: Suscripción del dispositivo para recibir el pedido del sensado de una temperatura particular. El dato es un String que indica cual es la temperatura a sensar, siendo posible para el equipo sensar los siguientes tipos de temperatura:

Dato “amb”: Procede a medir temperatura ambiente.

Dato “ivt”: Procede a medir temperatura del equipo.

Dato “mvf1”: Procede a medir temperatura del mvf 1.

Dato “mvf2”: Procede a medir temperatura del mvf 2.

Descripción 1.2: Suscripción del dispositivo para recibir el pedido de ensayo de los módulos fotovoltaicos. El dato recibido es un String que indica la distribución y la resolución de la curva de la siguiente forma “100;D”, donde 100 es la cantidad de puntos y D es su distribución.

Descripción 1.3: Suscripción del dispositivo para recibir el pedido de cambio de hora. El dato es un String con el formato “13:00:00”

Descripción 1.4: Suscripción del dispositivo para recibir el pedido de cambio de fecha. El dato es un String con el formato “22-11-2021”, evitar las barras para la fecha en este mensaje.

Descripción 1.5: Suscripción del dispositivo para recibir el pedido del almacenado de nueva alarma. El dato es un String con formato “10:00:00”

Descripción 1.6: Suscripción del dispositivo para recibir el pedido del borrado de una alarma particular o de todo el conjunto de alarmas. El dato es un String con formato número de alarma “3” o con formato “T” para el borrado de todo el conjunto.

Descripción 1.7: Suscripción del dispositivo para recibir una nueva configuración de funcionamiento. El dato recibido es un String que indica la distribución y la resolución de la curva de la siguiente forma “D;100”, donde 100 es la cantidad de puntos y D es su distribución.

Publicación:

Tópico	Tipo de mensaje	Descripción
TopicRaiz/Nombre/temp/amb	dato (String)	Lea descripcion 2.1
TopicRaiz/Nombre/temp/ivt	dato (String)	Lea descripcion 2.2
TopicRaiz/Nombre/temp/mvf1	dato (String)	Lea descripcion 2.3
TopicRaiz/Nombre/temp/mvf2	dato (String)	Lea descripcion 2.4
TopicRaiz/Nombre/conf	dato (JSON)	Lea descripcion 2.5
TopicRaiz/Nombre/plot(+)/mvf1(+)/punto	dato(JSON)	Lea descripcion 2.6
TopicRaiz/Nombre/plot(+)/mvf1(+)/time	dato(JSON)	Lea descripcion 2.7
TopicRaiz/Nombre/plot(+)/mvf1(+)/rad	dato(JSON)	Lea descripcion 2.8
TopicRaiz/Nombre/plot(+)/mvf1(+)/temp	dato(JSON)	Lea descripcion 2.9

Tabla 6. Estructura tópicos MQTT de publicación

Descripción 2.1: Publicación del dispositivo sobre la temperatura ambiente sensada.

Descripción 2.2: Publicación del dispositivo sobre la temperatura del equipo sensada.

Descripción 2.3: Publicación del dispositivo sobre la temperatura de trabajo del módulo fotovoltaico 1 sensada.

Descripción 2.4: Publicación del dispositivo sobre la temperatura de trabajo del módulo fotovoltaico 2 sensada.

Descripción 2.5: Publicación del dispositivo sobre su configuración.

Descripción 2.6: Publicación del dispositivo sobre la información de un punto medido, en dicha información se tienen los valores de número de punto, valor de tensión y valor de corriente, estructurados en un dato JSON. Cabe destacar que este tópico puede variar en su subtema plot al subtema test para diferenciar la publicación de un sensado por alarma o por petición, así como también este tópico puede variar en su subtema mvf1 al subtema mvf2 para diferenciar los paneles ensayados. Por ejemplo:

Destino: TopicRaiz/Nombre/test/mvf2/punto - Mensaje: Dato(JSON)

Descripción 2.7: Publicación del dispositivo sobre la información de los tiempos de ejecución medidos en el proceso de ensayo de los paneles solares, en dicha información se tienen los valores de fecha del ensayo, tiempo de comienzo, tiempo de finalización, tiempo real en ms del proceso de levantado de curva I-V, estructurados en un dato JSON. Cabe destacar que este tópico puede variar en su subtema plot al subtema test para diferenciar la publicación de un sensado por alarma o por petición, así como también este tópico puede variar en su subtema mvf1 al subtema mvf2 para diferenciar los paneles ensayados.

Descripción 2.8: Publicación del dispositivo sobre la información de la radiación medida en el proceso de ensayo de los paneles solares, en dicha información se tienen los valores de radiación al comienzo del ensayo y radiación al final del ensayo, estructurados en un dato JSON. Cabe destacar que este tópico puede variar en su subtema plot al subtema test para diferenciar la publicación de un sensado por alarma o por petición, así como también este tópico puede variar en su subtema mvf1 al subtema mvf2 para diferenciar los paneles ensayados.

Descripción 2.9: Publicación del dispositivo sobre la información de las temperaturas de funcionamiento medidas en el proceso de ensayo de los paneles solares, en dicha información se tienen los valores de temperatura de trabajo del módulo fotovoltaico al iniciar el ensayo, temperatura de trabajo del módulo fotovoltaico al finalizar el ensayo, temperatura ambiente al iniciar el ensayo, temperatura ambiente al finalizar el ensayo, temperatura del equipo al iniciar el ensayo, temperatura del equipo al finalizar el ensayo, estructurados en un dato JSON. Cabe destacar que este tópico puede variar en su subtema plot al subtema test para diferenciar la publicación de un sensado por alarma o por petición, así como también este tópico puede variar en su subtema mvf1 al subtema mvf2 para diferenciar los paneles ensayados.

2.1.8 Servidor web y WebApp.

El servidor web, también conocido como Back-End, tiene varios objetivos y funcionalidades necesarias para la usabilidad del equipo desarrollado.

Algunas de ellas, las más importantes podríamos suponer, son:

- Administrar todas las peticiones que se hagan desde el Front-End hacia el Back-End y responder a las mismas brindando también la información que se requiera. Las cuales pueden ser necesarias para obtener información de los usuarios o información de los sistemas instalados.
- Es el encargado de centralizar la información en la base de datos, enviar las consultas, obtener los datos, realizar operaciones muy típicas como filtros y ordenamientos, también aplicada a usuarios o sistemas instalados.
- Y también será quien a través del protocolo MQTT se comunique con los diferentes equipos instalados, recibiendo datos o enviando configuraciones.

La Web App por su parte es la encargada de brindar al usuario la interfaz gráfica para interactuar con los equipos, ver los datos, y otras funcionalidades que se detallarán en el apartado de funcionamiento.

A. Investigación de componentes

La aplicación web se desarrolló en un ambiente conocido como stack MEAN, por las siglas de los microservicios utilizados en la misma, los cuales son:

- Angular framework: Aquí desarrollamos lo que se conoce como Front-end, es decir lo que el usuario vé en su pantalla y a través de la cual puede interactuar con los equipos y con la información de ellos.

Angular es un framework para aplicaciones web desarrollado en TypeScript, es de código abierto y es mantenido por Google. Se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con la capacidad de Modelo Vista Controlador (MVC).

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

- Node JS: es un entorno de ejecución asíncrono en tiempo real, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor de Google, es multiplataforma y de código abierto. Fue desarrollado para la capa del servidor basado en el lenguaje de programación JavaScript, y fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como son los servidores web.

- Express: es un entorno de desarrollo minimalista para Node.js que permite estructurar una aplicación de una manera ágil. Su objetivo es que no tengamos que reinventar la rueda cada vez que queramos crear una aplicación web, ofreciéndonos soporte para las principales necesidades en este tipo de aplicaciones: gestión de peticiones y respuestas, cabeceras, rutas, vistas, etc.

- MongoDB: es un sistema de base de datos NoSQL, orientado a documentos y también de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

B. Análisis de funcionamiento



Figura 69. Interacción Front-end y Back-end [9]

Como hemos mencionado anteriormente, y complementado con la imagen superior, el usuario interactúa con el sistema desarrollado y con la información que este haya generado a través de la WebApp desarrollada en el Front-End con Angular.

Cuando realizamos una solicitud para ingresar al servidor la primera pantalla que se nos presenta es la siguiente:

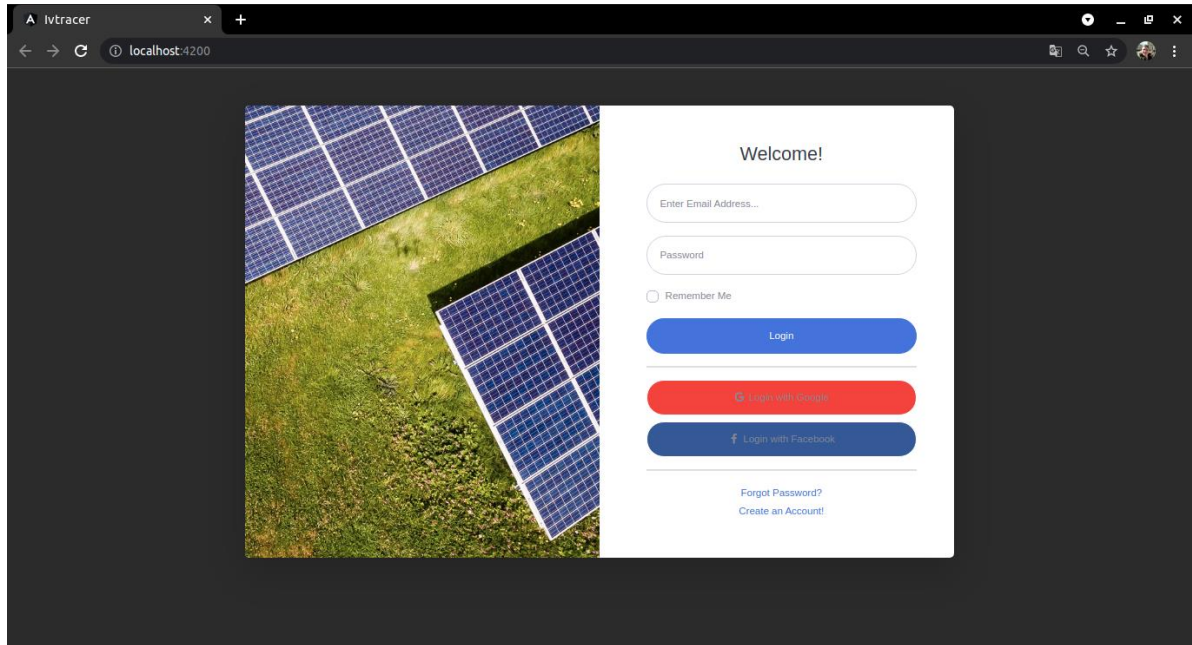


Figura 70. Pantalla de Login

Desde aquí podremos registrarnos (si es nuestra primera vez en el sistema) o loguearnos con nuestro usuario y contraseña.

Si vamos a registrarnos, se cargará una pantalla como la que vemos en la imagen siguiente:

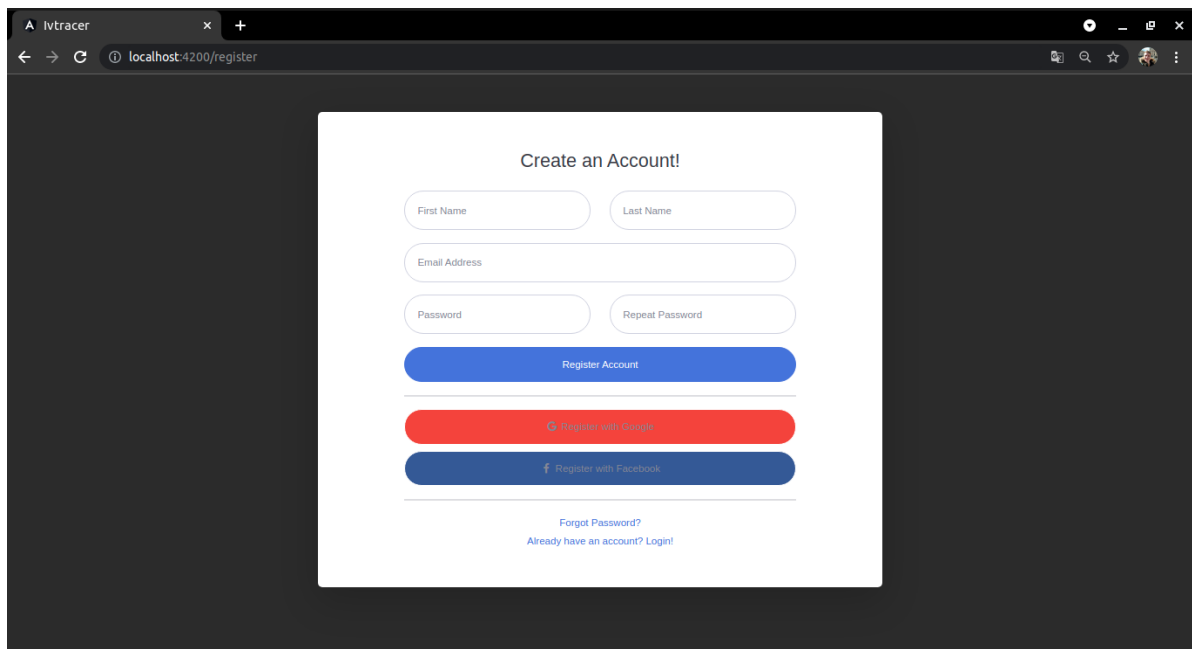


Figura 71. Pantalla para registrarse como usuario

Esta nos pedirá unos datos, los cuales se almacenarán en una tabla llamada users dentro de nuestra base de datos que contiene toda la información referida a los usuarios registrados.

Aquí en caso de querer registrarnos con un usuario que ya existe, obtendremos un mensaje de error y el formulario se resetea para nuevo ingreso.

Una vez registrados, podremos loguearnos en el sistema y así se cargará nuestro dashboard o panel de control. El cual estará vacío hasta que agreguemos un nuevo dispositivo.

En caso de introducir datos erróneos obtendremos un mensaje de error y como ha de suponerse no entrará en el sistema, sino que quedará en la pantalla de login, como se muestra en la imagen siguiente:

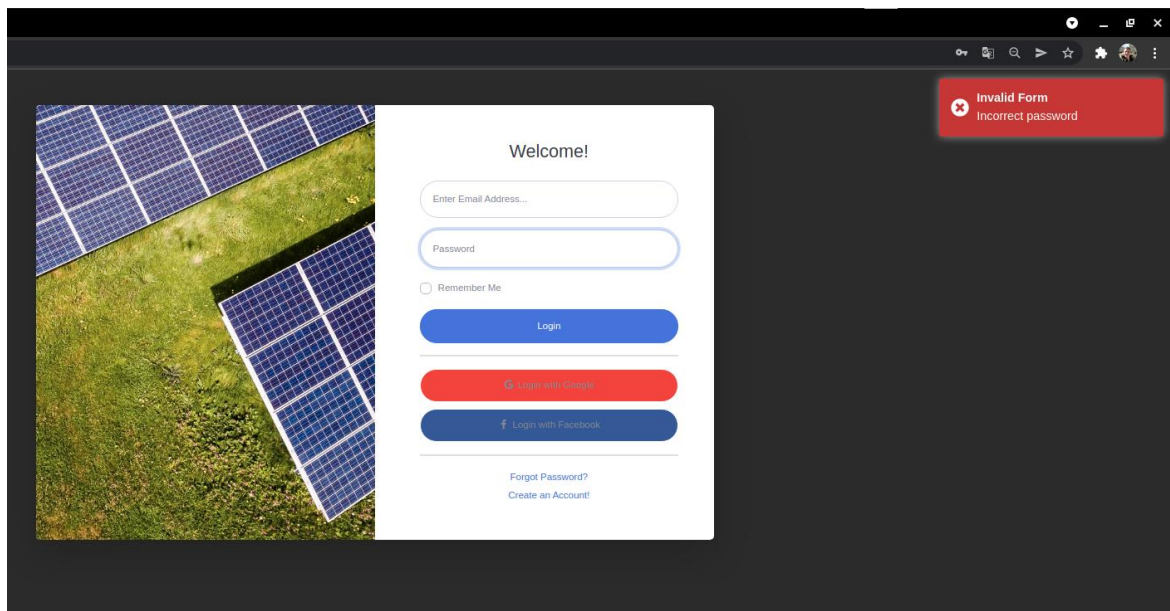


Figura 72. Mensaje de error de login

En la siguiente imagen vemos nuestro dashboard en el cual podemos ver:

- Un mensaje de que nos hemos logueado correctamente.
- El menú de navegación de forma lateral ubicado a la izquierda
- Una barra de sesión en el lado superior
- Un botón para agregar dispositivos.

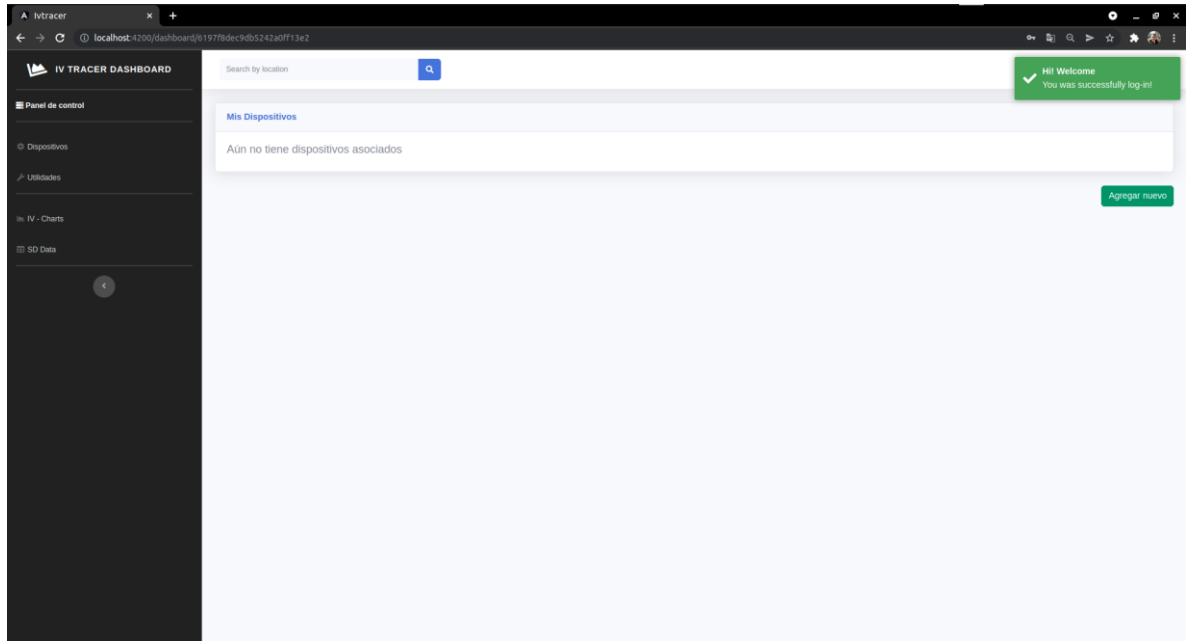


Figura 73. Dashboard personal vacío

Como hemos dicho, nuestro dashboard se encontrará vacío, clickeando el botón verde Agregar nuevo, podremos registrar un nuevo equipo para su posterior seguimiento.

A continuación, presentamos la pantalla para agregar nuestro dispositivo:

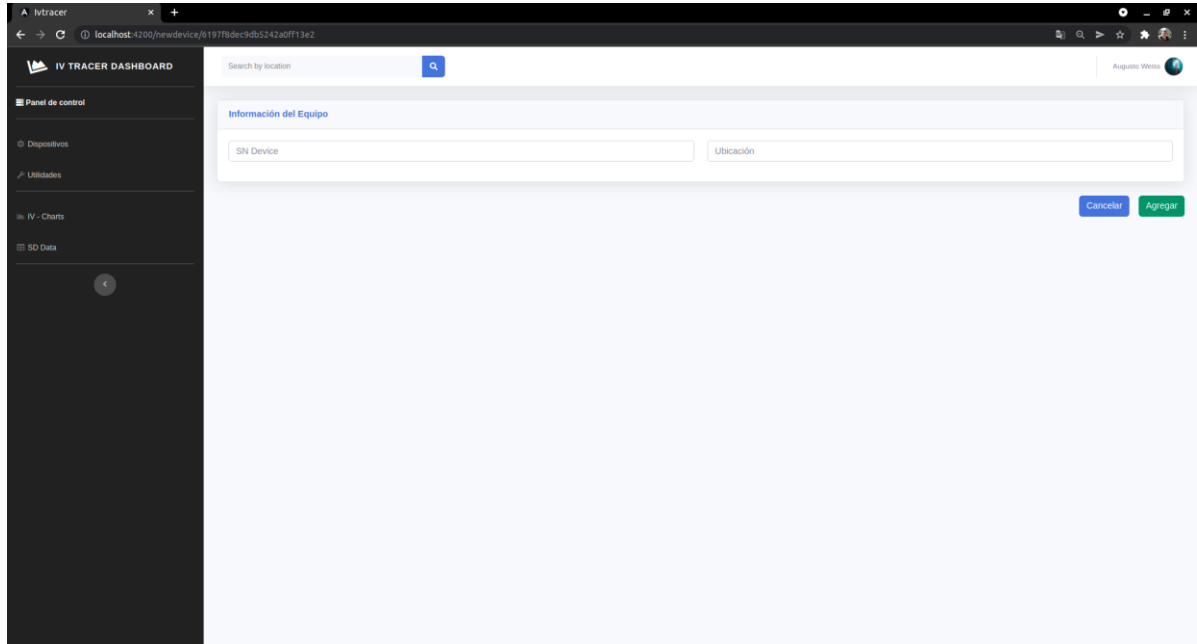


Figura 74. Pantalla añadir dispositivo

En ella podremos agregar nuestro dispositivo a seguir completando el campo SN device, el cual hace referencia a un código (Serial Number) característico y único de cada dispositivo/sistema instalado el cual el usuario conocerá de antemano.

También nos encontramos con el campo Ubicación el cual es de referencia para el usuario, pensando en un usuario que tenga más de un sistema instalado y en diferentes localidades o ubicaciones.

Una vez agregado el mismo, ya podremos visualizarlo en nuestro panel de control o dashboard.

En la siguiente imagen vemos como el dispositivo se agregó correctamente y hemos agregado otro más de manera que quede representado el caso en el que un usuario tiene asociado más de un dispositivo. También podemos ver algunos datos asociados al mismo, como son el SN device, la ubicación, el tipo de distribución y la cantidad de puntos por curva que tiene configurado el dispositivo.

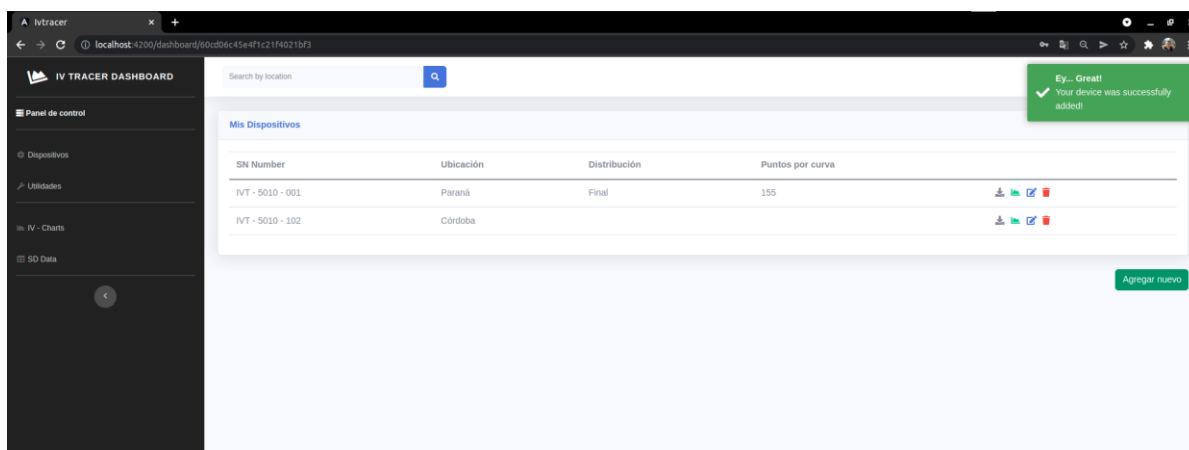


Figura 75. Dashboard e información

Ahora que ya tenemos un dispositivo asociado a nuestro usuario podremos interactuar con él, y para ello tenemos cuatro iconos a la derecha de cada uno de los dispositivos que explicaremos a continuación.

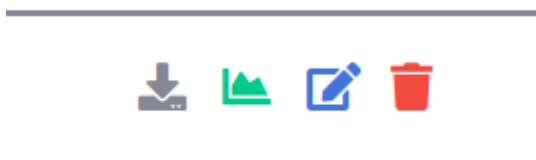


Figura 76. Iconos de acciones sobre un dispositivo

El primer icono que vemos, en color gris, es un icono de descarga. El mismo descargará en un archivo de tipo csv, toda la información asociada a curvas tomadas por el equipo y que estén guardadas en la base de datos del servidor. En este punto si todo ha ido bien deberíamos tener la misma información que hay en la tarjeta de memoria SD interna del equipo instalado.

El segundo icono nos llevará a una pantalla para visualizar las curvas obtenidas.

En el tercer icono encontramos la pantalla que nos dará la posibilidad de realizar modificaciones sobre la configuración de un dispositivo en concreto.

Por último, el cuarto dispositivo, en color rojo, desvincula al dispositivo de nuestra cuenta. De esta manera no lo veremos más en nuestro dashboard.

A continuación, iremos a la pantalla de Gráficos, icono verde antes nombrado.

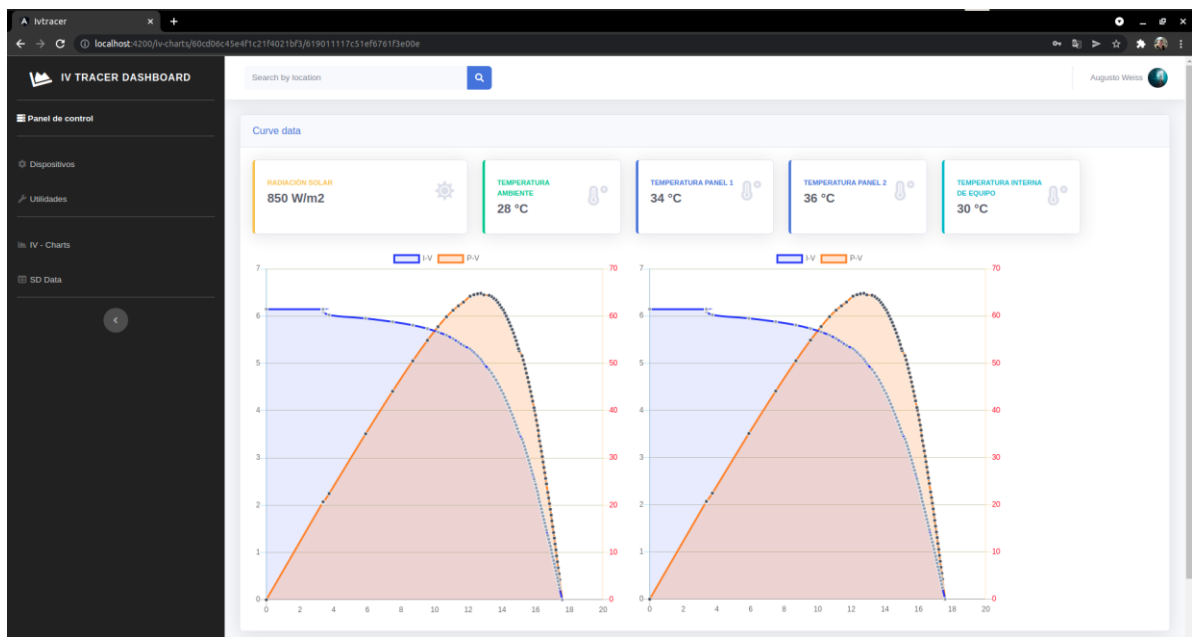


Figura 77. Pantalla de visualización de curvas

Cuando ingresamos a la pantalla antes nombrada, se nos carga en ella la última gráfica obtenida por el dispositivo referida a los dos módulos fotovoltaicos.

Podemos ver en la parte superior cinco tiles o tarjetas con la información obtenida de la medición que refieren a la situación ambiente donde se encuentran los módulos fotovoltaicos instalados.

El primero corresponde a la medición de la radiación solar existente en el momento de obtener la curva.

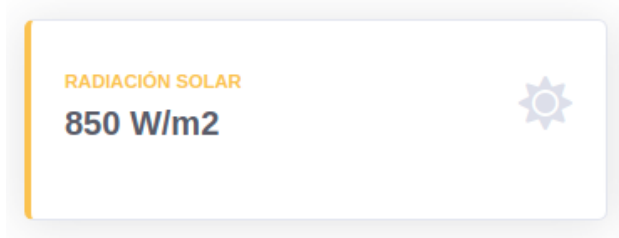


Figura 78. Tile de radiación solar

Luego tenemos el indicador de temperatura ambiente de dicho momento.



Figura 79. Tile de temperatura ambiente

Seguido de ello encontramos las temperaturas de trabajo de celda de los módulos fotovoltaicos.

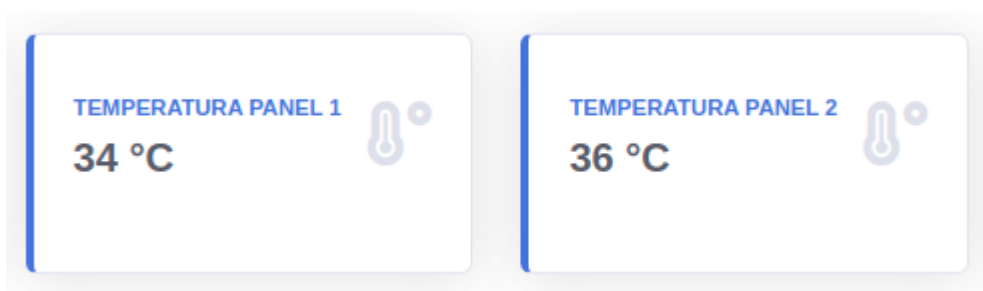


Figura 80. Tiles de temperaturas de trabajo de paneles solares

Y, por último, la medición de temperatura del equipo internamente.



Figura 81. Tile de temperatura interna del dispositivo

Seguido de los tiles antes descritos, nos encontramos con las dos gráficas obtenidas de los módulos fotovoltaicos.

Las cuales representan la parte más importante del desarrollo, ya que representan los puntos de *Corriente-Tensión* y *Potencia-Tensión* obtenidos en cada módulo fotovoltaico.

En la gráfica izquierda hemos ocultado la curva *Corriente-Tensión* a modo de mostrar esta funcionalidad que puede ser de utilidad a la hora de analizar una curva. Puede también ocultarse la curva de *Potencia-Tensión* y dejar visible solo la de *Corriente-Tensión*.

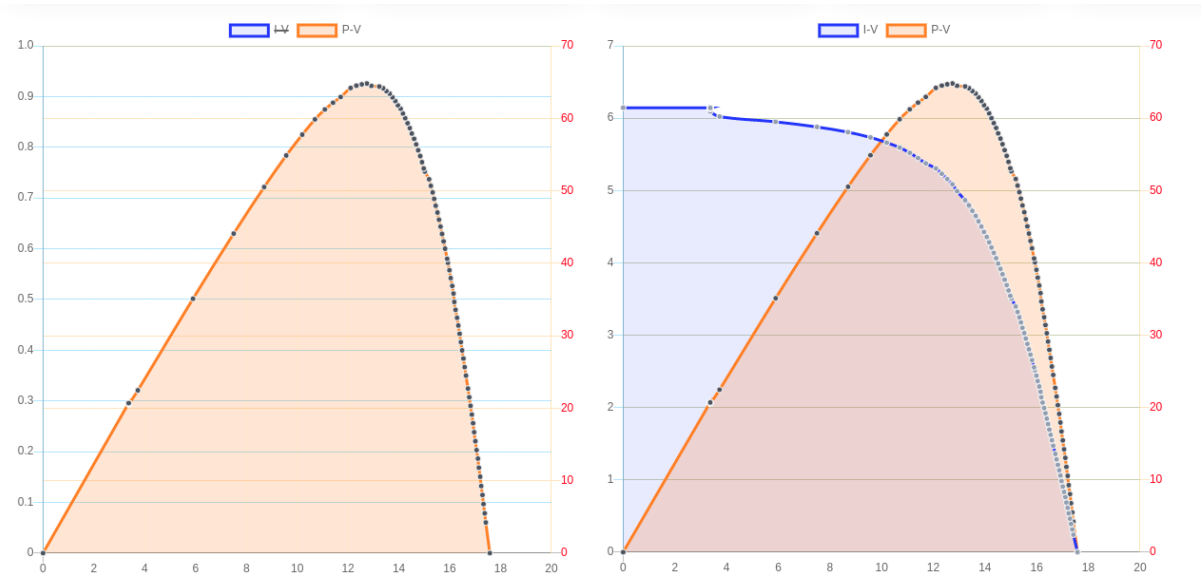


Figura 82. Curvas en Web App

Por último, pero no menos importante, luego de las gráficas se encuentra un pequeño menú para acceder a otras gráficas:

Figura 83. Selector de curvas

En este menú podemos seleccionar fecha y hora de la curva que queremos obtener el gráfico para analizar.

Figura 84. Indicador fecha y hora para gráfica de curvas

Volviendo a la pantalla anterior, nuestro dashboard, podemos ingresar al icono para configurar nuestro dispositivo.

A continuación, veremos en la imagen dicha pantalla.

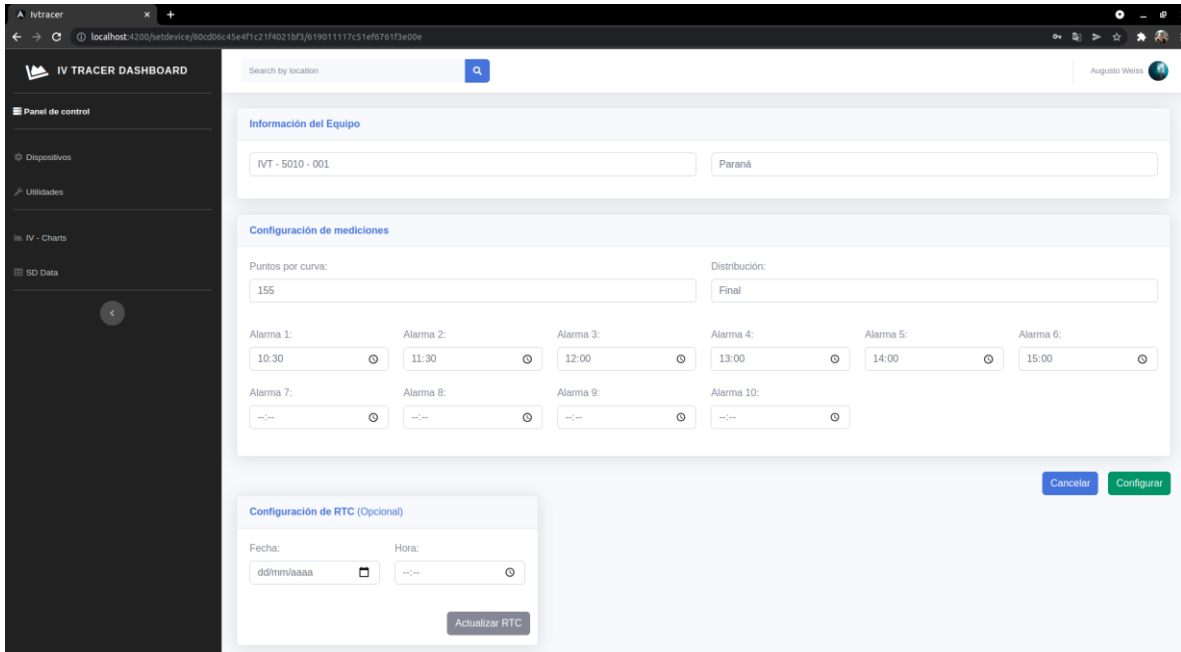


Figura 85. Pantalla para configurar un dispositivo de forma remota

En ella podemos ver que se han cargado algunos datos, ellos corresponden a la configuración cargada en la base de datos y que es idéntica a la configuración que hay en el dispositivo.

En primer lugar, vemos el número de serie del dispositivo y la ubicación, de forma similar a como veíamos en la pantalla para agregar un nuevo dispositivo.

La segunda parte corresponde a las configuraciones de las curvas y las alarmas configuradas para que el dispositivo tome curvas de los módulos fotovoltaicos.

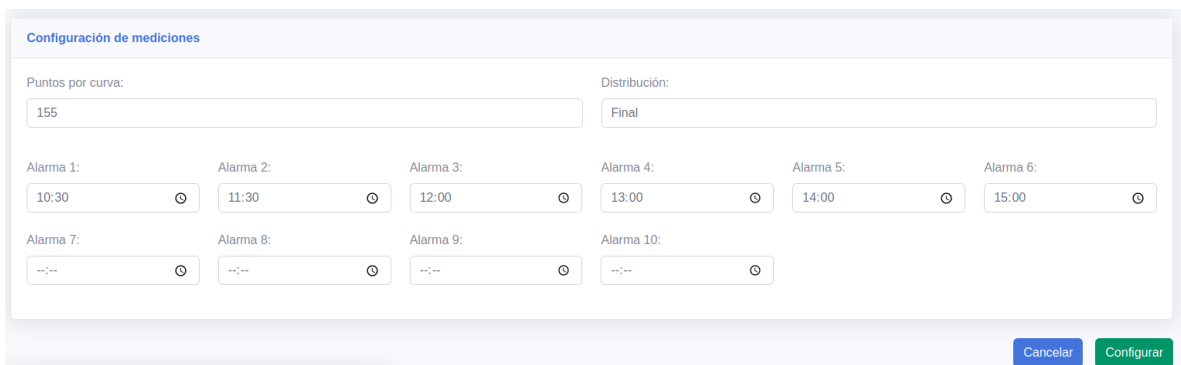


Figura 86. Información de preconfiguración de dispositivo

Aquí podemos ver como nuestro equipo se configuró para tomar 155 puntos por cada curva con distribución de tipo final y también podemos ver seis alarmas configuradas para obtener las curvas, es decir, este equipo obtendrá 6 curvas por día y por módulo fotovoltaico, en total unas 12 curvas.

En la imagen anterior también vemos el botón para confirmar la configuración, y el botón para cancelar y volver a nuestro dashboard.

Por debajo de las opciones de configuración del dispositivo encontramos una configuración opcional para la RTC, este puede cargarse y enviarse hacia el dispositivo el cual tomará estos valores y actualizará la RTC en caso de haberse desconfigurado y tener datos de fecha y hora incorrectos.

Figura 87. Actualización de RTC de forma remota

Hasta aquí hemos visto el funcionamiento y algunas de las características de esta Web app, a continuación, en el siguiente módulo desarrollaremos detalles del funcionamiento interno del servidor y la Web app.

C. Desarrollo del software

Como ya hemos mencionado anteriormente esta sección se encuentra bien definida por lo que es el Front-End del sistema y lo que es el Back-End del mismo.

Por ello es que analizaremos el desarrollo de cada uno de forma diferenciada.

Comenzando por el Front-End, el cual ya sabemos que ha sido desarrollada en *Angular*, un framework con estructura de trabajo basada en *Modelo - Vista - Controlador*.

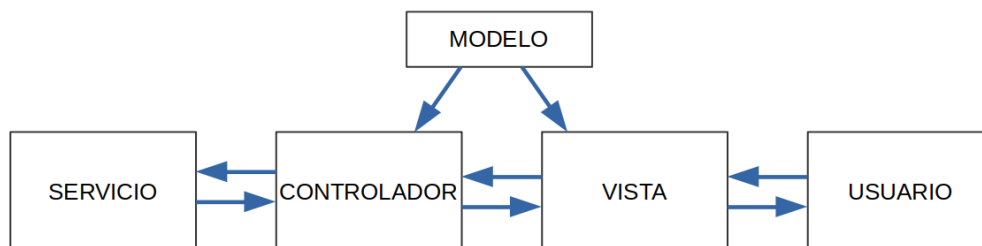


Figura 88. Diagrama general de desarrollo de Front-End en Angular

- Modelos:
Es la capa donde definimos la estructura de datos para enviar y/o recibir entre los controladores y las vistas, pero también a ser enviados hacia el Back-End a través de la API que comunica a estos dos sistemas.

En nuestro esquema de desarrollo nos encontramos con cuatro módulos, los cuales tienen cierta correlatividad con las cuatro tablas definidas en la base de datos.

Los modelos definidos son:

Usuario	_id
	Nombre
	Apellido
	Email
	Contraseña

Tabla 7. Modelo users en base de datos

devices	_id
	id usuario
	SN_device
	Ubicación
	Puntos
	Distribución
	Alarma 1
	Alarma 2
	Alarma 3
	Alarma 4

	Alarma 5
	Alarma 6
	Alarma 7
	Alarma 8
	Alarma 9
	Alarma 10

Tabla 8. Modelo devices en base de datos

Curva	_id
	SN_device
	Radiación
	Temperatura ambiente
	Temperatura módulo FV 1
	Temperatura módulo FV 2
	Temperatura interna

Tabla 9. Modelo curve en base de datos

Punto	_id
	id curva
	Valor de corriente
	Valor de tensión

Tabla 10. Modelo point en base de datos

- Vistas:

Para definir las vistas es necesario introducir el concepto de componentes.

Un componente en Angular está compuesto por cuatro archivos:

- Un archivo html que contiene el maquetado de la información que se visualizará en la página que se muestre.
- Un archivo css el cual define el estilo (colores, formas, etc) que en nuestro caso no usamos porque definimos uno global para toda la aplicación.
- Un archivo spec.ts para diferentes test del componente.
- Un archivo ts (typescript) que es el controlador del componente. Se desarrollará próximamente.

Angular, consta de un componente padre en el cual pueden definirse parámetros, como por ejemplo se nombró anteriormente el archivo de estilos css que luego cada componente hijo heredará.

En el componente padre, constamos también de un archivo de ruteo el cual irá redirigiendo la página hacia distintos componentes que se irán renderizando en la pantalla de nuestro navegador web.

En nuestro archivo de ruteo contamos con siete direcciones de ruteo para renderizar componentes, los cuales son:

- Login: renderiza el componente de login o pantalla de inicio.
- Dashboard: renderizar en pantalla el panel de control o dashboard.
- Register: renderizar en pantalla el componente registro de usuario.
- NewDevice: renderizar en pantalla el componente para añadir un dispositivo a un usuario.
- SetDevice: renderiza el componente para configurar un dispositivo.
- iv-charts: renderiza el componente que grafica las curvas obtenidas.
- path '**': es una ruta que define que ante cualquier error en la url nos dirige a una pantalla definida, en nuestro caso hemos definido la pantalla de inicio

Como hemos visto cada dirección de ruteo en este caso renderiza algún componente desarrollado, pero además cada componente es capaz de renderizar a otro componente tomándolo como hijo, dando esto la ventaja de reutilizar código.

Tomando esta ventaja, es que se desarrollaron componentes extras además de los antes nombrados que son:

- Nav: el componente nav es la barra superior que vemos a lo largo de las diferentes pantallas que se renderizan.
- Sidebar: es el componente que renderiza el menú lateral izquierdo.

Así sumamos un total de ocho componentes que heredan del componente padre.

Ejemplo:



Figura 89. Renderización de diferentes componentes

- **Controladores:**

Los controladores tienen por objetivo darle dinámica a los datos que vamos a ver en pantalla, para ello es necesario interactuar con la base de datos, pero a su vez para poder realizar esto es necesario comunicarse con nuestro Back-End el cual tiene las configuraciones necesarias para obtener diferentes datos de diferentes tablas en la base de datos.

La comunicación hacia el Back-End se realiza a través de una API (application programming interface). Del lado del Front-End en Angular tenemos la posibilidad de realizar estas peticiones o request con protocolo http hacia el Back-End a través de la definición de servicios.

Con esta breve introducción sobre los servicios pasamos a desarrollar cada controlador.

 - **Controlador login:**

El controlador del componente login obtiene los datos del formulario que se renderiza en la pantalla de login (usuario y contraseña).

Con ellos hacemos una consulta o request hacia el Back-End a través de nuestra API definida por el servicio de usuarios la cual nos responderá con un status 200 si el usuario existe y la contraseña es correcta y además nos devolverá los datos del usuario, así podremos pasar los datos al componente dashboard y renderizar su pantalla.

En caso de no existir o ser incorrecta la contraseña, entonces obtenemos la notificación de error en la pantalla de login.
 - **Controlador register:**

El controlador del componente register, muy similar al controlador login obtiene los datos del formulario que se renderiza en la pantalla de registro (nombre, apellido, usuario y contraseña).

Con ellos es capaz de realizar una consulta o request hacia el Back-End el cual nos responderá con un status 200 si el usuario no existe y la contraseña

es correcta, para luego renderizar el componente login, o informar a través de una notificación que hubo un problema al querer registrar el usuario, pudiendo ser que el usuario ya existe o que la confirmación de contraseña es incorrecta.

- Controlador dashboard:

El controlador del dashboard inicialmente con los datos del usuario obtiene a través de un servicio dedicado a dispositivos los dispositivos asociados al usuario, por ello podemos ver todos nuestros dispositivos, con ello además completar la tabla de datos que vemos en la vista del dashboard.

Otras funcionalidades que tiene este controlador es realizar tomar los eventos que se den en los botones o iconos de la pantalla y rutear así los diferentes componentes, por ejemplo, si clickeamos sobre el icono de configuración de un dispositivo, nos realiza el ruteo a tal componente enviando también los parámetros necesarios, siendo en este ejemplo el id del dispositivo a configurar.

Este controlador es el encargado también de poder exportar los datos de las curvas de un dispositivo generando el archivo csv y su posterior descarga.

- Controlador add-device:

El controlador del componente add-device toma los datos del formulario que se renderiza en la pantalla add-device (SN device y ubicación).

Con ellos realiza una petición o request hacia el Back-End a través del servicio definido para dispositivos, y con la respuesta del mismo podremos mostrar el dispositivo añadido o un mensaje de error, en este último caso reseteando los valores del formulario.

- Controlador set-device:

Este controlador, lo primero que resuelve teniendo el id del dispositivo que va a configurarse es completar los datos del formulario, de esta manera el usuario puede conocer si el dispositivo ya contaba con alguna configuración. En caso de no existir alguna configuración, el formulario quedará vacío.

En ambos casos el usuario puede modificar los campos del formulario y una vez confirmados este controlador enviará los datos al servicio desarrollado para dispositivos para luego ser almacenados en base de datos y transmitidos hacia el equipo que quiere configurar.

De forma muy similar este controlador es capaz de recolectar los datos para actualizar la fecha y hora de la RTC del dispositivo.

- Controlador iv-charts:

El controlador del componente iv-chart tiene como tarea, resolver la parte gráfica de las curvas.

Para ello, una vez obtenidos los valores de las curvas a analizarse ordenan los valores de forma que en la pantalla la curva quede bien expuesta, esto se debe a que, una vez hecha la petición a la base de datos, la misma devuelve los registros encontrados de forma aleatoria.

Todos estos puntos obtenidos son almacenados en vectores que luego será argumento de la función que grafica los puntos, pero también en este controlador se genera el vector correspondiente a los puntos de potencia ya

que estos no se miden en el equipo, sino que se obtienen por Ley de Ohm, $P = V \times I$ (Potencia igual a Tensión por Corriente).

Por otro lado, este controlador, contiene las definiciones de las características de los gráficos, como son colores, valores en los ejes, la leyenda y otros.

- Servicios:

Los servicios en Angular son los encargados de comunicarse con el Back-End de nuestra aplicación, a través de la API diseñada por nosotros.

Las peticiones se componen de cuatro operaciones básicas conocidas como CRUD por sus siglas en inglés: Create, Read, Update and Delete. Las cuales además son correlativas a las peticiones http conocidas como post, get, put, delete respectivamente.

En nuestro sistema hemos diseñado cuatro servicios diferentes, uno para cada modelo antes definido.

- `user.services`: diseñado para realizar las peticiones http referidas a usuarios. En ella encontramos peticiones como:
 - Obtener un usuario
 - Registrar un usuario
 - Actualizar un usuario (cuando se le asocia un nuevo dispositivo)

Las peticiones de tipo *delete* quedan reservadas al administrador del servidor.

- `device.services`: diseñado para realizar las peticiones http referidas a dispositivos. Algunas de las peticiones son:
 - Obtener un dispositivo
 - Añadir un dispositivo
 - Actualizar un dispositivo
 - Actualizar RTC del dispositivo
 - Eliminar un dispositivo

- `curve.services`: diseñado para realizar solo una petición http referida a las curvas almacenadas y es la de obtener cierta curva. Obviamente desde el lado del usuario no es posible crear una curva, esto solo puede hacerlo un dispositivo.

- `points.services`: al igual que el servicio para curvas, está diseñado para realizar solo una petición http referida a los puntos almacenados y relacionado a cierta curva.

Hemos comentado hasta este punto el proceso, los aspectos y las funcionalidades del desarrollo hecho para el lado del Front-End, a partir de ahora comenzaremos con el desarrollo del lado del Back-End.

Para ello traemos nuevamente a colación la siguiente imagen, a modo de recordar cómo se compone el sistema completo.



Figura 90. Estructura de comunicación Front-end y Back-end [9]

En este momento nos concentramos en el servidor el cual define cómo se comporta la API, y quien consulta la base de datos opera los resultados y devuelve al Front-End un resultado.

Dentro del entorno de Node JS encontramos tres funcionalidades muy importantes y específicas que se corren en un script:

- 1- El servicio de Express: desde aquí administramos las peticiones que se realicen hacia el servidor, donde definimos como trabaja nuestra API.
- 2- La conexión a la base de datos: aquí se alojan los datos necesarios para conectarse al *cluster* de mongoDB y poder interactuar con los datos almacenados.
- 3 - El servicio MQTT: el cual se compone de dos servicios, uno para publicar y otro para suscribirse.

Express es un entorno de desarrollo para Node JS que permite estructurar una aplicación de una manera sencilla y rápida, ofreciéndonos soporte para las principales necesidades en este tipo de aplicaciones: gestión de peticiones y respuestas, cabeceras, rutas, vistas, etc.

Dentro del mismo encontramos los archivos de ruteo para las peticiones http, la definición de los modelos tal como ya hemos definido anteriormente, y los controladores, quienes acceden a la base de datos, y según lo que se requiera se devuelve la información obtenida a través de una respuesta o response.

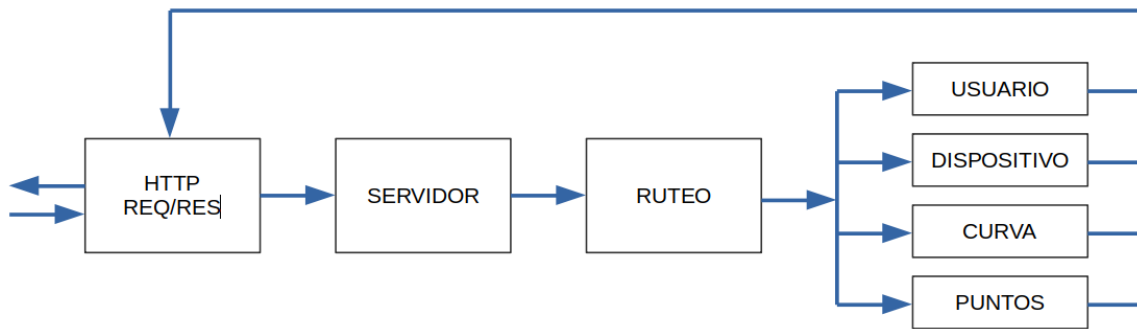


Figura 91. Diagrama en bloques de Back-end

- **Modelos:**
Los modelos que aquí se definen son cuatro y son idénticos a los definidos en el Front-End, que se corresponden a cada una de las tablas definidas en la base de datos. Ellos son: modelos usuarios, dispositivos, curvas y puntos.
- **Ruteo:**
Dentro del ruteo encontramos también cuatro archivos de ruteo, uno por cada controlador y tabla de la base de datos.
Estos reciben las peticiones http con los métodos CRUD y rutean cada petición hacia el método correspondiente que se haya definido en el controlador del servidor, para luego tomar el resultado y devolverlo por medio de la API al controlador del Front-End.
- **Controladores:**
Cada controlador se definió para acceder a una tabla diferente de la base de datos, es por ello que contamos con cuatro controladores distintos, y con métodos definidos para cada necesidad del desarrollo, aunque muy similar entre ellos.
 - **Controlador de usuario: *UserController***
Aquí encontraremos los métodos definidos para registrarnos, loguearnos, obtener los datos de un usuario, actualizar datos y eliminar datos.

Registro de usuarios:

A través del cuerpo de la petición, mayormente conocido como *body request* obtenemos los datos que el controlador del Front-End nos ha enviado y podemos crear el registro dentro de la base de datos.

Como en el modelo de usuarios el campo email se ha marcado como *único*, el controlador podrá crear el registro en la base de datos solo si no se encuentra ya registrado.

Si pudo registrarse correctamente entonces retornamos en el *response* un *status 200*.

Login de usuarios:

Al igual que para registros de usuarios, se obtienen el usuario y contraseña que llegan en el body de la request.

Aquí realizamos una consulta a la base de datos, y si se encuentra el usuario, entonces procedemos a chequear que la contraseña ingresada sea igual a la contraseña almacenada en la base de datos.

Si ambos datos son correctos, entonces se devuelven los datos del usuario a través del *body response*.

Caso contrario obtendremos un *status 404* indicando que los datos son incorrectos.

Obtener datos de un usuario:

Esta petición es más sencilla ya que en los parámetros de la request se define un id, el cual será utilizado para buscar en la base de datos.

Aquí no hay ningún tipo de lógica, solo devolver el registro de usuario que se requiere. Esto es para uso ya interno en la aplicación web.

Eliminar un usuario:

Este método, al igual que el anterior realiza la consulta a la base de datos mediante el id del registro, para luego dar la instrucción de eliminarlo.

Es un método utilizado a nivel de administrador de la base de datos y no por los usuarios de la aplicación web.

Actualizar usuario:

Por medio de este método, y mediante un id de usuario, podemos buscar tal registro en la base de datos y escribir todos o alguno de sus campos.

Si la actualización pudo ejecutarse correctamente entonces obtendremos como respuesta un *status 200* y en el body de la *response* los datos actualizados del usuario.

- Controlador de dispositivos: *device Controller*

Aquí encontraremos los métodos definidos para agregar un dispositivo nuevo, obtener los dispositivos asociados a un usuario, obtener los datos de un dispositivo en concreto, actualizar la configuración de un dispositivo, actualización de la RTC y el método para eliminar un dispositivo.

Agregar dispositivo:

A través de este método es que podemos agregar un nuevo dispositivo a la base de datos, almacenando sus datos de configuración y ubicación, estando asociado a un SN device.

Dispositivos asociados a un usuario:

Esta petición recibe por medio del body de la request un id de usuario, para con el buscar los registros de dispositivos guardados en la base de datos que estén asociados al id recibido y poder responder con una lista de dispositivos por medio de la *response*.

Obtener datos de un dispositivo:

Similar al método antes descrito, esta petición recibe por medio del body de la request un id de un dispositivo, para luego responder a través del body de la *response* la configuración y datos correspondientes al mismo.

Actualizar dispositivo:

Por medio de este método, y mediante un id de dispositivo, se busca un registro en la base de datos y se escribe alguno o todos sus campos.

Si la actualización pudo ejecutarse correctamente entonces obtendremos como respuesta un *status* 200 y en el body de la *response* los datos actualizados del usuario.

Actualizar RTC:

Este método tiene por objetivo, recibir los datos de fecha y hora indicados por el usuario, para luego ser enviados por protocolo MQTT hacia el dispositivo MPPT y así poder actualizar sus datos en caso de tener diferencias significativas en los mismos.

Eliminar un dispositivo:

La petición de eliminar un dispositivo recibe un id de dispositivo para luego eliminar de la base de datos los registros del dispositivo correspondiente.

- Controlador de curvas: *curveController*

Dentro de este controlador encontramos los métodos definidos para agregar y obtener los datos de una curva.

En este controlador no se implementan los métodos de eliminar y de actualizar una curva, ya que no son funcionalidades que requiera el usuario, dado que esta información proviene del dispositivo MPPT a través de protocolo MQTT.

- Controlador de puntos: *pointController*

De igual manera que en el controlador de curvas, dentro de este controlador encontraremos los métodos definidos para agregar y obtener los datos correspondientes a un punto de una curva.

En este controlador tampoco se implementan los métodos de eliminar y de actualizar un punto de una curva, ya que no son funcionalidades que requiera el usuario, dado que esta información es proveniente del dispositivo MPPT a través del protocolo MQTT.

2.2 Diseño final

El prototipo final se montó sobre una caja estanca de tamaño 250mm x 300mm donde se pudo ubicar la placa base, la fuente de alimentación y las baterías. En ella se ubicó una llave que permite encender y apagar manualmente el dispositivo y encontrarán dos prensacables plásticos por el cual saldrán 4 cables correspondiente a, sensor de temperatura del módulo fotovoltaico 1, sensor de temperatura del módulo fotovoltaico 2, sensor de temperatura ambiente y sensor de radiación PAR.

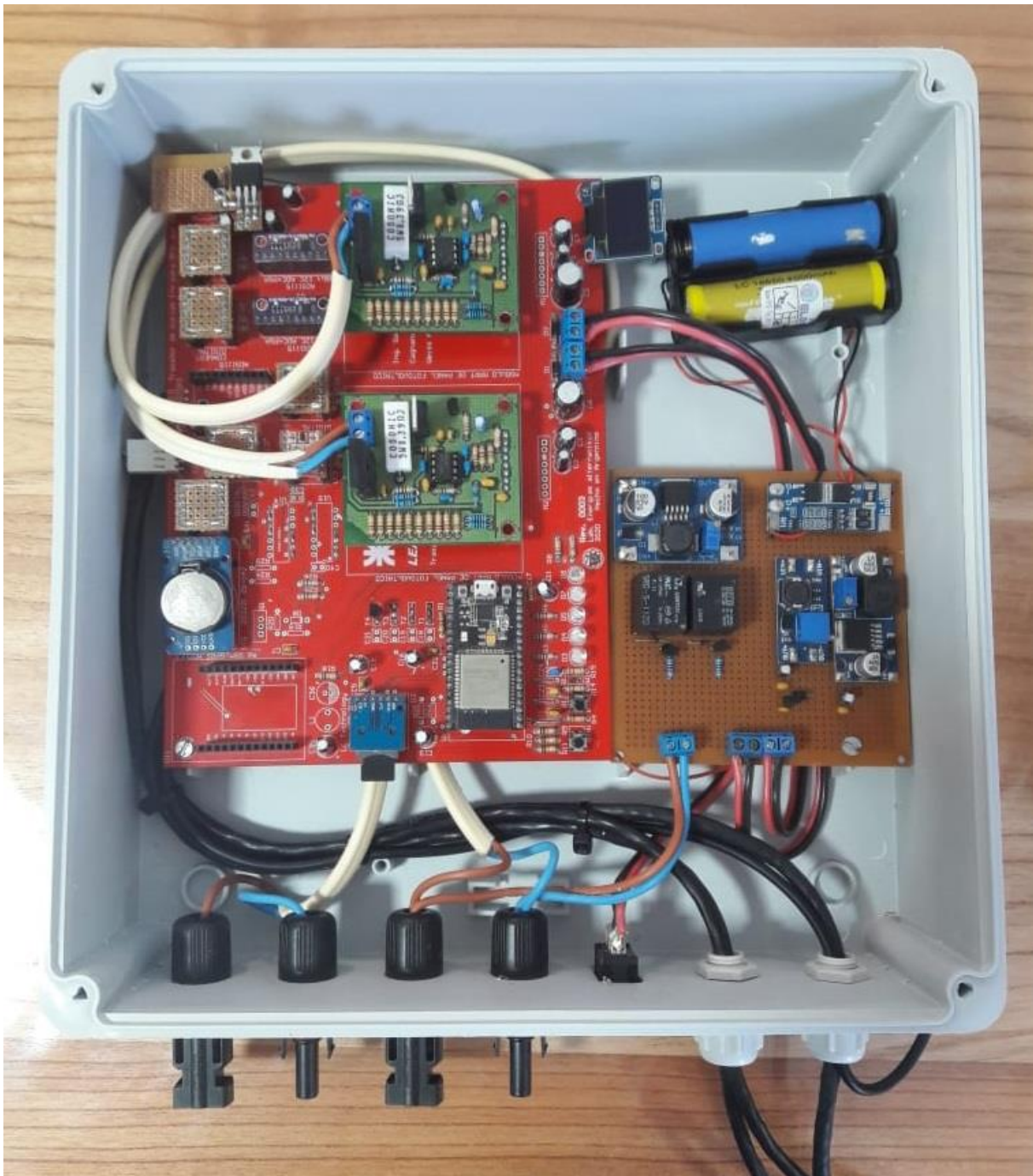


Figura 92. Diseño final - destapado



Figura 93. Diseño final - lateral inferior

Para la conexión entre el dispositivo y el módulo fotovoltaico se utilizaron dos pares de conectores MC4, el mismo conector se utiliza en la caja de conexiones del panel solar. Con el fin de ubicar el dispositivo a conveniencia se utiliza un cable una a una con el mismo conector.



Figura 94. Alargue MC4 para panel solar

Por el lado del sensor de temperatura, se utilizó una pieza dedicada para poder montar el sensor detrás de una celda del panel solar como se muestra en la siguiente figura.



Figura 95. Montaje del equipo completo

Por último, mostramos cómo termina montado el dispositivo al panel solar:

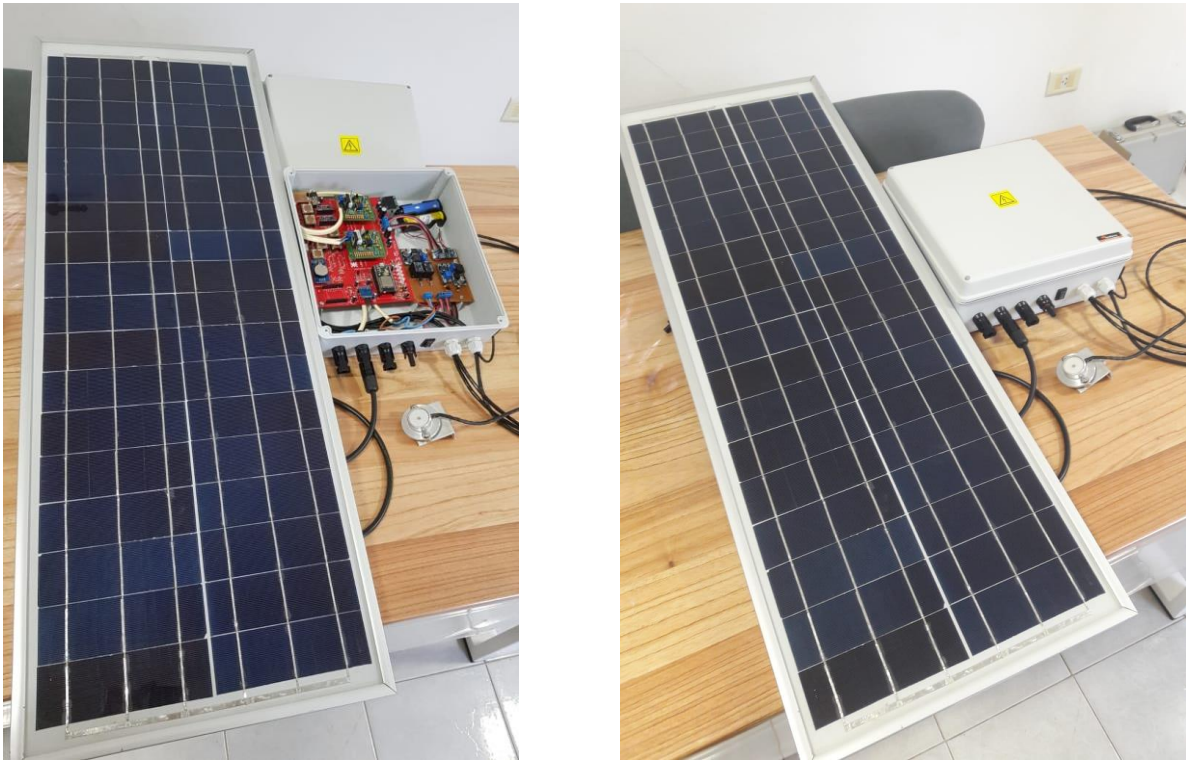


Figura 96. Montaje del equipo completo



Figura 97. Montaje del equipo completo

Operación del equipo:

- Paso 1: En primera medida el usuario que maneja el dispositivo deberá abrir el equipo, reiniciarlo manualmente desde la llave principal y presionar el botón BTN1 de la placa hasta que salga la leyenda “Modo Online” en la pantalla de visualización
- Paso 2: En este modo, el dispositivo habilitó un punto de acceso inalámbrico mediante una red WiFi con SSID: IV_Tracert_001, debemos conectarnos a ella con algún dispositivo.
- Paso 3: Conectados a la red, debemos buscar el dispositivo desde algún navegador WEB con la dirección ip 192.168.120.3, una vez allí tendremos acceso al Módulo Web ya explicado en la sección 2.1.6. Debemos configurar el equipo según los requerimientos que nosotros deseamos, en caso de necesitar un uso local del equipo, nos mantendremos en este punto para la operación del mismo.
- Paso 4: Terminada la configuración, se debe resetear el equipo ya sea manteniendo presionado el pulsador BTN2 o apagándose desde la llave central, luego de esto ya podremos cerrar el dispositivo. A modo de control sugerimos esperar a que en la pantalla de visualización indique la condición “Modo Sleep” y se apague, en este punto sabremos que el dispositivo entró en el modo correspondiente, se conectó a la red WiFi que fue seteada en su configuración, precargó la alarma seteada en su módulo RTC y entro en modo ahorro de consumo. En caso de tener un error y nunca conectarse a la red correspondiente, quedará con la pantalla de visualización encendida indicando “Modo Sleep”.
- Paso 5: El dispositivo ya está configurado y podremos implementar su uso desde la WebApp que se diseñó para su manejo.

Capítulo 3: Resultados

Como resultado de este “Proyecto final de carrera” hemos obtenido una herramienta dedicada al análisis del comportamiento de módulos fotovoltaicos y su variación en la eficiencia al momento de generar energía eléctrica.

Diferenciado de equipos comerciales por tener la capacidad de transmitir los datos hacia un servidor web y así analizar los datos de forma remota, hemos cumplido nuestros objetivos en lograr un equipo que pueda permanecer instalado junto a dos módulos fotovoltaicos y ensayarlos día a día según las configuraciones hechas por el usuario.

A continuación, presentaremos un ejemplo de resultados obtenidos durante el proceso de pruebas realizadas con el sistema completo:

En las siguientes imágenes veremos un adelanto de cómo se podrán analizar dos mediciones realizadas sobre el mismo panel, estando en la primera instancia completamente limpio y luego parcialmente sucio, y si bien los resultados no son totalmente significativos aún, a simple vista puede verse la diferencia que existe entre ellos.

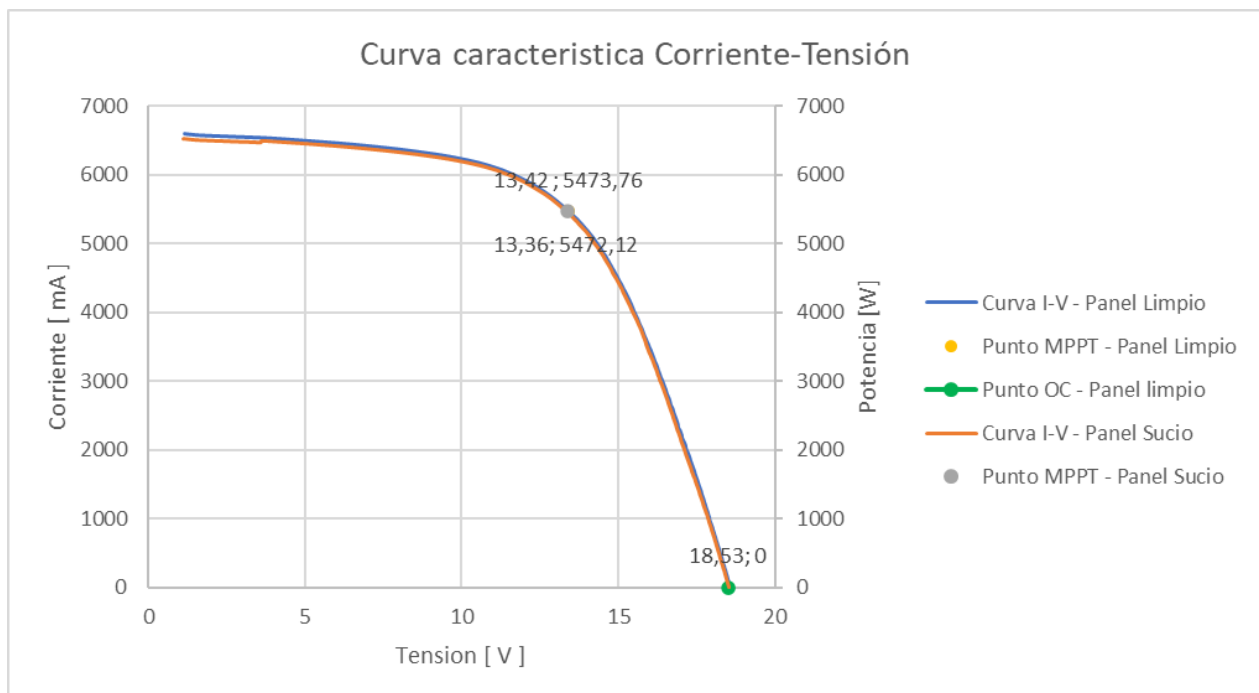


Figura 98. Comparación curvas Corriente-Tensión

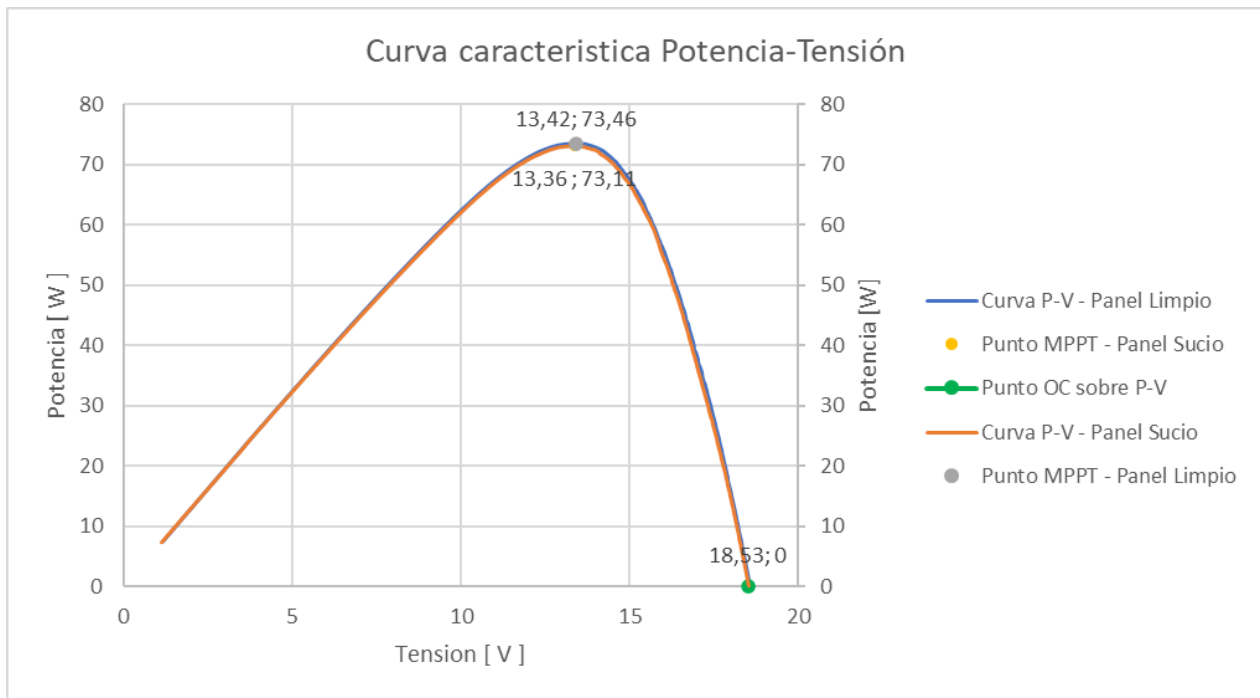


Figura 99. Comparación curvas Potencia-Tensión

Capítulo 4: Análisis de Costos

En el presente capítulo, se verá un análisis de costos en materiales más un análisis de costos de diseño y desarrollo. Se justificará el costo total del desarrollo enmarcando al mismo en precios del mercado, y comentaremos la situación que encarará el desarrollo implementado respecto a posibles planes de ventas.

4.1 Costos de materiales y desarrollo

A continuación, ha de detallarse la lista de materiales incluidos para el desarrollo del dispositivo MPPT completo:

Descripción	Cantidad	Precio unitario	Precio total
Placa PCB 15 x 20 cm	1	\$ 2.100	\$ 2.100
Placa PCB 5 x 6 cm	2	\$ 850	\$ 1.700
Placa preperforada	1	\$ 265	\$ 265
ESP 32 - NodeMCU	1	\$ 1.245	\$ 1.245
Módulo Micro SD	1	\$ 169	\$ 169
Módulo RTC	1	\$ 360	\$ 360
Convertor 3v3 - 5v	1	\$ 135	\$ 135
ADC ADS1115	3	\$ 1439	\$ 4.317
Display OLED	1	\$ 785	\$ 785
Step-down DC-DC LM2576HV	1	\$ 1089	\$ 1089
Módulo Bms cargador de baterías	1	\$ 778	\$ 778
Step-down LM2596	1	\$ 334	\$ 334
Step-up MT3608	1	\$ 223	\$ 223
Baterías de litio	2	\$ 713	\$ 1.426
MosFET IRFP150N	2	\$ 456	\$ 912
Amplificador operacional LM358N	2	\$ 163	\$ 326
MosFet IRF1404	3	\$ 372	\$ 1.116
Sensores DS18B20	4	\$ 346	\$ 1.384

Sensor de radiación PAR	1	\$ 60.934	\$ 60.934
Conectores MC4 (par macho-hembra)	2	\$ 262	\$ 524
Prensacables	2	\$ 60	\$ 120
Caja estanca 25x30 cm	1	\$ 1890	\$ 1890
Switch on-off	1	\$ 135	\$ 135
Capacitores varios	-	\$ 6600	\$ 6600
Resistencias varias	-	\$ 1100	\$ 1100
Miscellaneous	-	\$ 9000	\$ 9000
Total			\$ 98.967

Tabla 11. Análisis de costo de materiales

El monto total de la lista de materiales corresponde a \$98.967 (pesos noventa y ocho mil novecientos sesenta y siete).

Sumamos a este monto, las horas contempladas de diseño y desarrollo por un saldo de \$120.000 (pesos ciento veinte mil)

Materiales	\$ 98.967
Diseño y desarrollo	\$ 120.000
Total	\$ 218.967

Tabla 12. Análisis de costo total

El resultado final del análisis de costo se resuelve en \$218.967 (pesos doscientos dieciocho mil novecientos sesenta y siete).

El resultado del análisis de costo final puede parecer a primera vista elevado, pero es necesario contextualizar este resultado, teniendo en cuenta los valores de mercado no solo de los equipos que realizan tareas similares a las del sistema desarrollado nombrados en la introducción, sino también los valores de cotización de módulos fotovoltaicos y los elementos asociados a una instalación fotovoltaica.

También podemos destacar que en el sistema se incluye un sensor de radiación PAR con un costo de \$60.934 (pesos sesenta mil novecientos treinta y cuatro), el cual es el más económico del mercado, pudiendo elevarse hasta valores de \$170.000 (pesos ciento setenta mil).

4.2 Plan de venta y comercialización

El equipo desarrollado no tiene altas perspectivas de venta, el origen del mismo no fue la comercialización y la generación de ganancias económicas. Es necesario recordar que el mismo se desarrolló con el fin de brindar a la comunidad científica una herramienta de análisis de comportamientos y rendimientos de módulos fotovoltaicos.

De igual manera, puede ser eventualmente comercializado.

En caso de comercializarse, nos encontramos con dos caminos posibles:

- vender el diseño
- vender el producto.

En el primer caso tendrá mayor peso el valor monetario de las horas de trabajo, más un valor estimado de la ganancia a la que se estaría renunciando por la venta de la propiedad intelectual.

En el caso de venderse el producto, deberá analizarse el mercado con mayor profundidad y establecer un plan de marketing adecuado para detectar posibles compradores. Con ello, se deberá analizar la relación entre clientes y cantidad mínima de dispositivos a fabricar para recuperar la inversión hecha en los dispositivos.

Capítulo 5: **Discusión y Conclusión.**

Como resultado de este desarrollo que hemos llevado a cabo bajo la forma de “Proyecto final de carrera” hemos de concluir que el producto obtenido será una herramienta totalmente innovadora y de gran utilidad para la comunidad científica del ámbito de las energías renovables, en especial de generación solar.

Cumpliendo los objetivos que nos hemos propuesto al comenzar el mismo, y felices del resultado obtenido, nos conmueve el orgullo de sabernos partícipes y colaboradores de futuros grupos de investigación que contando con el sistema desarrollado serán generadores de información y divulgación científica.

La posibilidad de mantener conectados dos módulos fotovoltaicos y poder realizar en ellos diferentes tratamientos o imponer diferentes condiciones externas, hacen que sumado a que como ya hemos nombrado con anterioridad, la capacidad de monitorear la información de forma remota y disponibilizar siempre que se cuente con un dispositivo con internet, se diferencie del mercado y brinde así la mayor prioridad al usuario en el análisis de los datos a obtenerse.

Un punto que no podemos dejar de mencionar, son los problemas y dificultades que hemos enfrentado durante el transcurso del desarrollo de este “Proyecto final de carrera”. A continuación, citaremos algunos de ellos:

- Error en los buses de comunicación.
- Ruido en la señal diente de sierra que controla la resistencia de la carga electrónica MosFET en la placa MPPT.
- Sobrecarga de memoria RAM en el microcontrolador ESP32

Para cerrar este capítulo hemos de listar una serie de mejoras posibles que pueden realizarse para hacer de este desarrollo una herramienta con mayor funcionalidad, mejor desempeño y mayor seguridad.

- Integración con módulo GPRS para dar conectividad al equipo cuando este se encuentre en zonas rurales.
- Incorporar mayor capacidad en las baterías del equipo.
- Encriptado de datos de los usuarios en el servidor web.
- Optimización de la base de datos y en la forma de relacionar las tablas.

Capítulo 6: Literatura Citada.

- [1] <https://datos.bancomundial.org/indicador/EG.USE.ELEC.KH.PC?end=2019&start=1970&view=chart>
- [2] <http://www.lv-energy.com/#!/paneles-solares/>
- [3] <http://www.lv-energy.com/#!/paneles-solares/>
- [4] https://www.exploreembedded.com/wiki/Overview_of_ESP32_features._What_do_they_practically_mean%3F
- [5] <https://components101.com/microcontrollers/esp32-devkitc>
- [6] <https://www.ht-instruments.com/en/products/photovoltaic-testers/i-v-curve-tracers/i-500w/>
- [7] <http://www.solmetric.com/pva1000s.html>
- [8] <http://www.seawardsolar.com/userfiles/pv200-solar-pv-tester-curve-tracer.php>
- [9] <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT-ROP4y55gdFSOWMpVwsBBhdaMf89aYz58iQ&usqp=CAU>
- [10] <https://angular.io/>
- [11] <https://nodejs.org/en/>
- [12] <https://expressjs.com/>
- [13] <https://www.mongodb.com/>
- [14] https://en.wikipedia.org/wiki/Representational_state_transfer
- [15] <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

ANEXO

