



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES

Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Autor
Elena Álvarez Castro

Directores
Juan Antonio López Villanueva
Salvador Rodríguez Bolívar

Departamento de Electrónica y Tecnología de computadores
Escuela Internacional de Posgrado

Granada, 15 de septiembre de 2020



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES

Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Autor
Elena Álvarez Castro

Directores
Juan Antonio López Villanueva
Salvador Rodríguez Bolívar



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

Título: CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES. Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Autor: Elena Álvarez Castro

Directores: Juan Antonio López Villanueva, Salvador Rodríguez Bolívar

Departamento: Electrónica y Tecnología de Computadores

Palabras clave:

Fotovoltaico, Célula Fotovoltaica, Característica I-V, Panel Solar, Modelo, LED, Estimación de Parámetros, HTML, Javascript, AJAX, JSON, SVG, Energías Renovables, Microcontrolador.

Resumen:

Este proyecto presenta un sistema de toma de muestras de un panel fotovoltaico, que, a través de un conjunto de circuitos y software de control, permite captar Curvas I-V bajo diferentes niveles de iluminación, ya sea natural o artificial. Cuenta también con un sistema Web de visualización de datos, que permite al usuario tomar muestras, cambiar las condiciones de iluminación artificial y comparar los valores experimentales con los parámetros obtenidos de forma teórica a partir de algoritmos de cálculo ya desarrollados por diferentes investigadores.



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

Título: CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES. Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Author: Elena Álvarez Castro

Advisors: Juan Antonio López Villanueva, Salvador Rodríguez Bolívar

Departament: Electrónica y Tecnología de Computadores

Keywords:

Photovoltaic, Photovoltaic Cell, I-V Curve, PV Modules, Model, LED, Parameters Estimation, HTML, Javascript, AJAX, JSON, SVG, Renewable Energy, Microcontroller.

Abstract:

This project uses an I-V Tracer and a sampling system to obtain information about a PV panel. A collection of circuits and multiple control programs allows us to capture I-V curves under alternative levels of lighting, whether natural or artificial. Likewise, it is complemented with a Web data visualization system that allows the user to take samples, change the artificial lighting conditions and contrast the experimental values with the theoretical parameters obtained. These adjustments are established theoretically through calculation algorithms that have already been developed by several researchers.



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

Título: CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES. Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Autor: Elena Álvarez Castro

Directores: Juan Antonio López Villanueva, Salvador Rodríguez Bolívar

Departamento: Electrónica y Tecnología de Computadores

Yo, **Elena Álvarez Castro**, alumna del Máster en Ingeniería Electrónica Industrial de la **Escuela Internacional de Posgrado de la Universidad de Granada**, con DNI 75576450-Z, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Master en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Firmado: Elena Álvarez Castro

Granada, 25 de septiembre de 2020



TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

Título: CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES. Sistema inalámbrico de toma de muestras de un panel fotovoltaico y su modelado teórico bajo diferentes condiciones de iluminación artificial y natural

Autor: Elena Álvarez Castro

Directores: Juan Antonio López Villanueva, Salvador Rodríguez Bolívar

Departamento: Electrónica y Tecnología de Computadores

D. **Juan Antonio López Villanueva**, Catedrático de Universidad del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

D. **Salvador Rodríguez Bolívar**, Catedrático de Universidad del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado **CARACTERIZACIÓN Y MODELADO DE CÉLULAS SOLARES DE SILICIO EN DIFERENTES CONDICIONES AMBIENTALES**, ha sido realizado bajo su supervisión por **Elena Álvarez Castro** y autorizamos la defensa de dicho trabajo ante el tribunal correspondiente

Y para que conste, expiden y firman el presente informe en Granada a 15 de septiembre de 2020,
Los directores:

Juan Antonio López Villanueva

Salvador Rodríguez Bolívar

Agradecimientos

Realizar este proyecto ha sido para mi todo un reto. Por este motivo quiero agradecer a las personas que me han permitido llegar hasta aquí.

A mi padre, mi madre y mi hermana, por apoyarme en todo momento. A mis amigos, los cuales puedo nombrar: Lucía, Alfonso, María y Elvira por creer siempre en mí y darme ánimos en mis momentos débiles.

Índice general

1. Objetivos	1
2. Introducción	3
3. Fundamento Teórico	7
3.1. Característica I-V de una célula fotovoltaica	7
3.2. Modelado de curvas I-V	8
4. Diseño del Sistema	13
4.1. Circuito Trazador de Curvas I-V	14
4.2. Sensor de Temperatura	15
4.3. Sistema de Iluminación Artificial	19
4.3.1. Prototipos	20
4.3.2. Diseño final	22
4.4. Sistema Completo	24
5. Visualización de datos y Sistema de comunicación	27
6. Datos Experimentales	33
6.1. Iluminación Natural	33
6.2. Iluminación Artificial	34
7. Conclusión	37
A. Código página visualización de datos vía web. Archivos HTML y javascript	39
B. Archivo de Código en C. Programa microcontrolador.	57
Bibliografía	69

1. Objetivos

El objetivo de este trabajo es implementar un sistema autónomo de toma de muestras de una placa solar (Curvas I-V), bajo diferentes niveles de iluminación, tanto artificial como natural. Igualmente comparar los resultados experimentales obtenidos con diferentes métodos teóricos de estimación de parámetros, propuestos en diferentes publicaciones.

En asignaturas relacionadas con dispositivos fotovoltaicos, este sistema puede utilizarse en el laboratorio para realizar prácticas y tomas de datos experimentales. Facilitando así la comprensión del funcionamiento de una placa solar fotovoltaica y los cambios que ocurren en la corriente generada dependiendo del tipo y nivel de iluminación, además de la temperatura de la propia placa.

Los objetivos específicos de este trabajo son:

- Diseñar un sistema autónomo de toma de muestras.
- Diseñar un sistema controlable de iluminación artificial.
- Visualizar los datos obtenidos vía web, que se ajuste a los siguientes puntos:
 - Realizar toma de muestras y su representación gráfica.
 - Regular el nivel de iluminación desde este interfaz.
 - Aplicar algoritmos para el cálculo del modelo teórico, visualizar los parámetros estimados y comparar resultados.
 - Mostrar contenido teórico del proyecto
- Diseño físico compacto, con estructura sólida para poder transportarlo.
- Basar el software en estándares de internet: tcp/ip, http, javascript y wi-fi. De esta forma será útil y accesible para cualquier usuario.

2. Introducción

La energía solar fotovoltaica es una de las energías renovables más utilizadas actualmente. La monitorización de los valores característicos de los paneles fotovoltaicos, es esencial para predecir su ciclo de vida y evitar problemas [6]. Esto es fundamental en grandes instalaciones, dónde existen varios paneles en serie y en paralelo, pues si el funcionamiento de uno de ellos no es correcto, afecta al rendimiento de toda la planta. Este es sólo un ejemplo de lo fundamental que es el estudio y la investigación en el campo de las energías renovables, para lograr aumentar su funcionalidad y eficiencia. El planeta debe avanzar hacia un desarrollo sostenible, dónde los combustibles fósiles dejen paso a las energías limpias, que aunque no lleguen a ser 100 % reciclables, disminuyen muy significativamente la contaminación y el vertido de residuos tóxicos.

En este trabajo se diseña y construye un sistema funcional de toma de muestras, de un panel fotovoltaico pequeño, pensado para realizar prácticas y estudios de laboratorio, mostrando que la monitorización de un panel solar se puede realizar de forma sencilla y obtener una gran cantidad de información y datos experimentales en tiempo real.

El sistema presentado parte de la base y la experiencia de un proyecto anterior, resultado de un trabajo fin de grado propio, entregado en junio de 2018¹[7]. Este proyecto era un prototipo menos complejo de un sistema trazador de curvas I-V, que a través de ethernet, conectaba con MatLab para mostrar gráficamente la característica I-V de un determinado panel fotovoltaico.

Sin embargo, a pesar de su buen funcionamiento, presentaba una serie de limitaciones:

- Todos los cálculos y representaciones gráficas se realizaban a través de MatLab, por tanto, sin un dispositivo con la versión adecuada de este entorno de cálculo no era posible la visualización de datos.
- La trasmisión de datos se realizaba a través de ethernet, conexión por cable, por lo que no era un sistema inalámbrico.
- No era posible medir la temperatura de la placa fotovoltaica.
- No contaba con una fuente de iluminación controlable, para regular el nivel de iluminación (artificial) incidente en el panel fotovoltaico.

Este nuevo sistema funciona de forma autónoma, ya que, está equipado con un microcontrolador que permite conexión inalámbrica, además de:

- Ser capaz de dibujar la característica I-V de un panel solar y medir su temperatura.
- Incluye una fuente de iluminación artificial controlable.
- Cuenta con una interfaz web específica para la visualización de datos en tiempo real y control del nivel de iluminación por el usuario (programación a través de HTML, javascript y gráficos SVG) desde cualquier dispositivo electrónico (PC, tablet, smartphone, etc).

¹El proyecto completo, con los archivos de código, se pueden encontrar en: enlace

El sistema completo, se divide en tres subsistemas independientes, pero comunicados entre sí:

- Sistema de toma de muestras
- Sistema de iluminación artificial
- Sistema de visualización de datos

El sistema de toma de muestras está formado por un circuito trazador de curva I-V y un sensor de temperatura. El sistema de iluminación artificial es desmontable, y permite la toma de muestras tanto en iluminación artificial como natural. Está formado por una serie de focos LED situados encima del panel solar mediante una estructura de varillas de aluminio y un circuito que se encarga de regular el nivel de iluminación que éstos proporcionan a la placa. Por último, la visualización de datos se realiza vía web, mediante comunicación inalámbrica. En la Figura 2.1 aparece un esquema general del sistema.

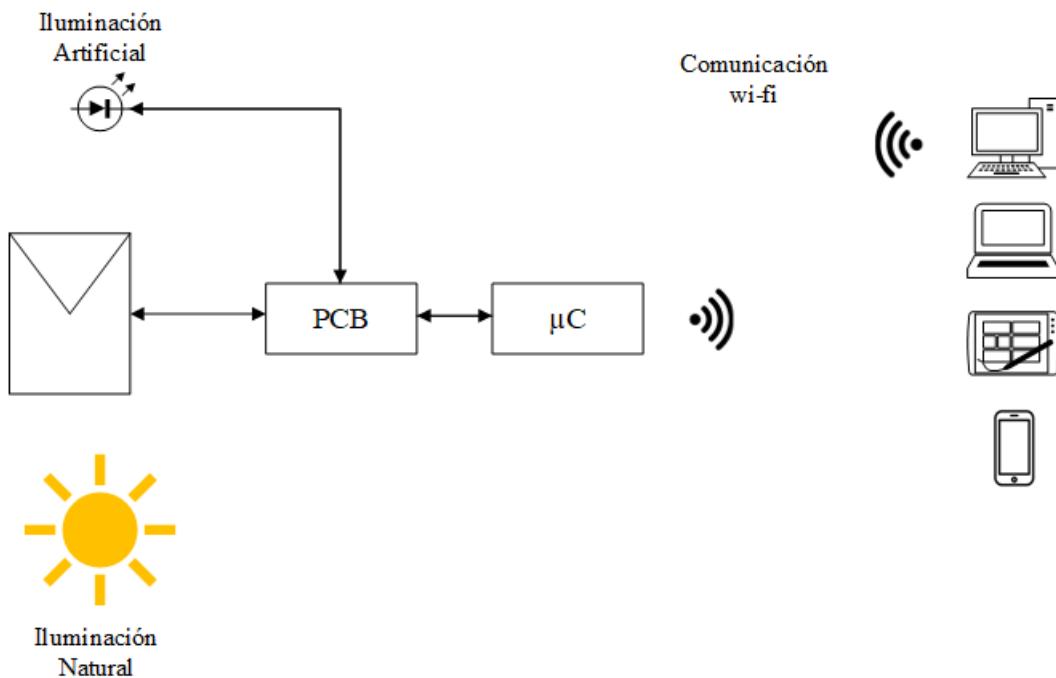


Figura 2.1.: Esquema del sistema diseñado. PCB: circuito trazador, conversor A/D, tarjeta SD, circuito sensor de temperatura, circuito regulador de iluminación.

En la página web es posible visualizar las muestras obtenidas de corriente y tensión, calcular los parámetros teóricos necesarios para modelar la curva experimental, a través de cuatro métodos diferentes de cálculo, observar datos experimentales tomados en iluminación natural (por medio de un repositorio de muestras, para facilitar su comparación con las muestras obtenidas en iluminación artificial), y controlar de forma simple la luz que los focos suministran al panel solar. Para realizar el modelado matemático de las curvas experimentales obtenidas dentro de la interfaz web, se han programado cuatro algoritmos de cálculo de parámetros en lenguaje javascript.

Todo el conjunto está construido sobre una bandeja de plástico resistente. Los circuitos están alojados en dos cajas diferentes, cerradas con tapaderas de plástico. Los focos LED se soportan gracias a una estructura de aluminio. Para que todo quede compacto, se utilizan bridas y tornillos para fijar todos los cables y placas de circuito a la base.

Introducción

Es difícil entender todas las aplicaciones que pueden llegar a tener conceptos como son el modelado de curvas I-V, los valores característicos o los parámetros teóricos, dentro del ámbito de investigación de las energías renovables. El sistema propuesto facilita su interpretación y ayuda a utilizar los conceptos visualizando datos reales.

3. Fundamento Teórico

Para comprender mejor el principio de funcionamiento del sistema y su fundamento teórico, a continuación, se realiza una introducción teórica a la característica I-V de una célula solar fotovoltaica y su modelado mediante un circuito eléctrico.

3.1. Característica I-V de una célula fotovoltaica

Una célula fotovoltaica es un dispositivo semiconductor, formado por una unión P-N [4]. Cuando la radiación solar incide en la célula, se generan portadores de carga que originan una corriente eléctrica (con los terminales de la célula conectados a una carga o en cortocircuito). A este proceso se le llama proceso fotovoltaico y se divide esencialmente en tres puntos:

1. La radiación solar es absorbida por la célula
2. Se generan portadores de carga
3. Estos portadores de carga se recolectan en los terminales de la célula, produciendo una corriente eléctrica

El principal material semiconductor que se utiliza para su fabricación es el Silicio. Los módulos fotovoltaicos están formados por un conjunto de células solares, las cuales se encuentran conectadas en serie o en paralelo. Las características del panel fotovoltaico utilizado son:

- Panel PV Logic, Solar Technology International
- 36 células en serie, de silicio monocristalino
- Apta para aplicaciones en exterior y resistente al agua

La curva I-V es la representación gráfica de la característica de salida de un panel fotovoltaico, su forma es la representada en la Figura 3.1. Muestra los valores de corriente y tensión que proporciona para diferentes cargas, desde un cortocircuito hasta circuito abierto. A partir de tres condiciones de operación principales, se definen sus valores característicos. Estas condiciones son [4][5]:

- Circuito abierto: terminales desconectados, por tanto la corriente suministrada es cero y el voltaje es máximo e igual al voltaje de circuito abierto V_{OC} .
- Cortocircuito: la placa está cortocircuitada, por tanto el voltaje es cero y la corriente es máxima e igual a la corriente de cortocircuito I_{SC} .

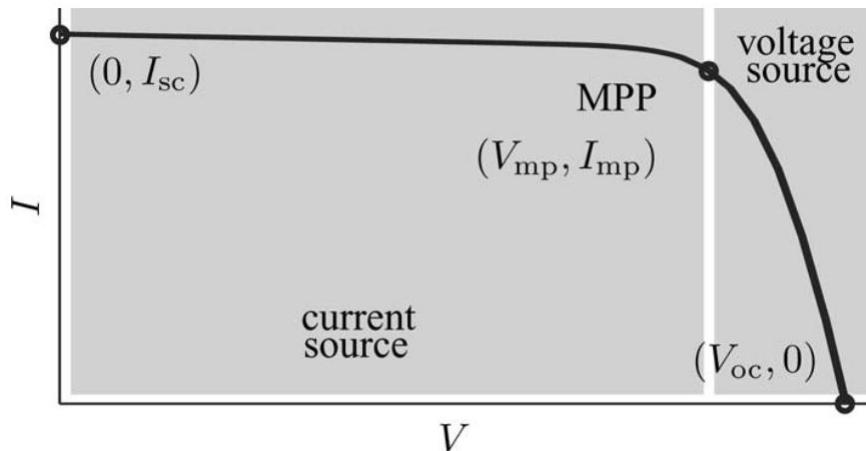


Figura 3.1.: Esquema Curva I-V [3].

- Punto de máxima potencia¹: el producto de la corriente por la tensión alcanza su valor máximo, la corriente y el voltaje en este punto siguen la nomenclatura I_{mp} y V_{mp} respectivamente.

Los valores característicos de un panel fotovoltaico son: voltaje de circuito abierto, V_{OC} , corriente de cortocircuito, I_{SC} , punto de máxima potencia, P_m , voltaje en el punto de máxima potencia, V_{mp} y corriente en el punto de máxima potencia, I_{mp} . Los fabricantes proporcionan estos valores para condiciones estándar². Los valores característicos que presenta el panel solar utilizado, aparecen la Tabla 3.1.

Potencia Máxima	5 W
Voltaje de Máxima Potencia V_{mp}	17.5 V
Intensidad de Máxima Potencia I_{mp}	0.29 A
Voltaje de Circuito Abierto V_{OC}	22 V
Intensidad de Cortocircuito I_{SC}	0.32 A
Temperatura de Operación	-40 a 85 °C
Dimensiones	306x218x25 mm

Cuadro 3.1.: Especificaciones de la Placa Solar Utilizada

3.2. Modelado de curvas I-V

Una célula solar fotovoltaica, se puede modelar idealmente como una fuente de corriente en paralelo con un diodo [3] (Figura 3.2), así se obtiene el modelo básico ideal que aparece en la Tabla 3.3. Dónde [6]:

- I_{PV} o I_g : corriente generada por la placa solar,
- I_S o I_{Sat} : corriente de saturación del diodo

¹Este punto es uno de los datos clave para los sistemas de seguimiento del punto de máxima potencia (MPPT).

²STC, Standard Test Conditions, irradiancia 1000 W/m², 25 °C y distribución espectral de 1.5.

- V_t está definido por la ecuación $V_t = \frac{AKT}{q}$, donde A es el factor de idealidad del diodo, q el valor absoluto de la carga eléctrica del electrón, T, la temperatura en Kelvin y K, la constante de Boltzman.

Existen varios modelos eléctricos para simular su funcionamiento y mostrar de forma más realista su comportamiento. Los más comunes añaden efectos de una resistencia serie y una resistencia en paralelo, dando lugar a los modelos de un único diodo. Al modelo que añade el efecto de R_s , se le llama modelo de cuatro parámetros, debido a que presenta cuatro incógnitas en el cálculo; siguiendo esta nomenclatura, al modelo que añade una resistencia en paralelo, se le conoce como modelo de cinco parámetros.

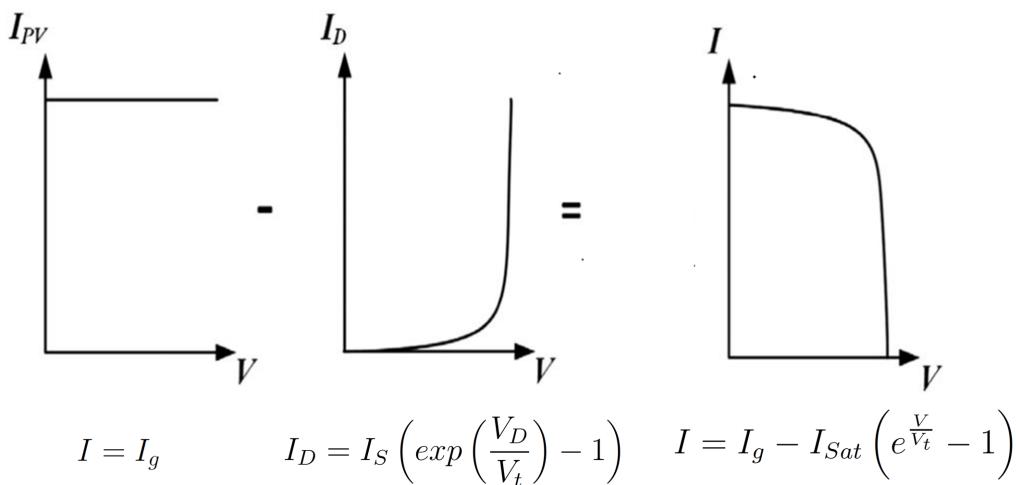


Figura 3.2.: Superposición de una fuente de corriente con un diodo de unión P-N

También existen modelos que añaden un segundo diodo, para tener en cuenta la corriente que se pierde durante la recombinación, proporcionando mayor exactitud en valores bajos de irradiancia. Sin embargo, la complejidad del cálculo es demasiada para determinadas aplicaciones dónde no se busca un nivel casi excelente de exactitud. El modelo de dos diodos, presenta siete incógnitas.

Los métodos matemáticos que se utilizan para la extracción de parámetros, son cuatro, dos de ellos basados en el modelo de un único diodo con el efecto de R_s y otros dos añadiendo el efecto de R_p . Cada uno utiliza diferentes aproximaciones y algoritmos de cálculo para encontrar los parámetros desconocidos (Tabla 3.2).

Para evaluar la exactitud de los métodos teóricos, se identifican tres zonas en la curva I-V. Un método teórico es bueno, cuando consigue ajustarse en estas tres zonas a la curva experimental tomada, independientemente de las condiciones de iluminación y temperatura. Las zonas son (Figura 3.3):

- Zona de corriente constante
- Zona de máxima potencia
- Zona de caída exponencial

Método teórico	Consideraciones para el cálculo de parámetros	Modelo que utiliza	Parámetros calculados
Randy Williams [6]	Resuelve de forma analítica un sistema de ecuaciones partiendo del funcionamiento del circuito, realizando una aproximación.	Un único diodo con efecto de R_S	I_g, I_{Sat}, R_S, A
Xiao [1]	Utiliza la ecuación resultante de que la derivada de la potencia con respecto al voltaje, en su punto máximo, es igual a cero.		
Silva [1]	Realiza un algoritmo de cálculo, realizando un barrido de posibles valores de A y R_S , calculando los demás parámetros de forma analítica.	Un único diodo con efecto de R_P	$I_g, I_{Sat}, R_S, R_P, A$
Villalva [1]	Busca que el punto de máxima potencia de curva modelada, sea igual al punto de máxima potencia de la curva experimental.		

Cuadro 3.2.: Métodos teóricos utilizados en el modelado.

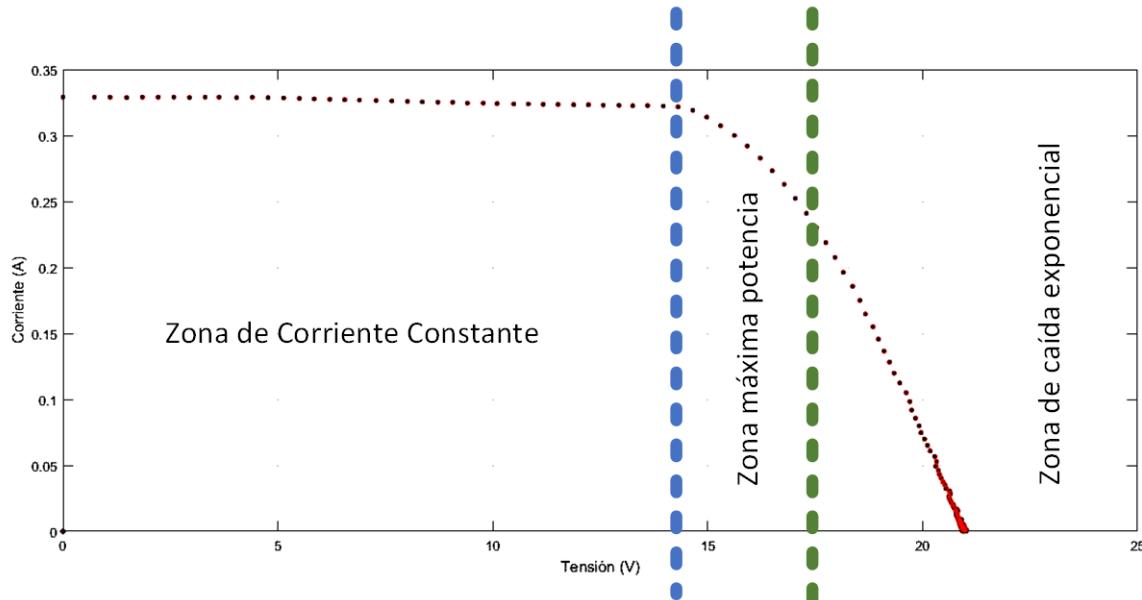


Figura 3.3.: Delimitación de zonas de una curva I-V.

Modelo	Circuito Equivalente	Ecuación
Ideal		$I = I_g - I_{Sat} \left(e^{\frac{V}{V_t}} - 1 \right)$
Un único diodo con efecto de R_s		$I = I_g - I_{Sat} \left[\exp \left(\frac{V+IR_s}{V_t} \right) - 1 \right]$
Un único diodo con efecto de R_p		$I = I_g - I_S \left[\left(e^{\left(\frac{V+IR_s}{V_t} \right)} - 1 \right) - \frac{V+IR_s}{R_P} \right]$
De dos diodos		$I = I_g - I_{S1} \left[\exp \left(\frac{V+IR_s}{Vt_1} \right) - 1 \right] - I_{S1} \left[\exp \left(\frac{V+IR_s}{Vt_2} \right) - 1 \right] - \frac{V+IR_s}{R_P}$

Cuadro 3.3.: Modelado de Células fotovoltaicas [3]

4. Diseño del Sistema

El sistema está controlado por el microcontrolador *MKR1000* (características y *Pinout* en Apéndice B). Incluye un circuito trazador de curvas I-V, un sensor de temperatura y un circuito encargado de regular el nivel de iluminación. Las placas de circuito implementadas, se agrupan en dos cajas¹. En la Figura 4.1, aparecen las dos cajas de conexiones cerradas (para evitar posibles problemas debido al polvo, humedad y demás factores).

En la caja 1 se encuentra el circuito trazador de curvas, junto con el conversor D/A y la entrada para la tarjeta SD (ha sido necesario realizar una pequeña ranura en la tapadera para facilitar la extracción de ésta). Aparece con más detalle en la Figura 4.2a. En la caja 2, se sitúa el relé encargado de conectar la placa solar al circuito trazador, el circuito regulador del nivel de iluminación y el circuito de acondicionamiento del sensor de temperatura. La caja 2 sin cerrar aparece en la Figura 4.2b.

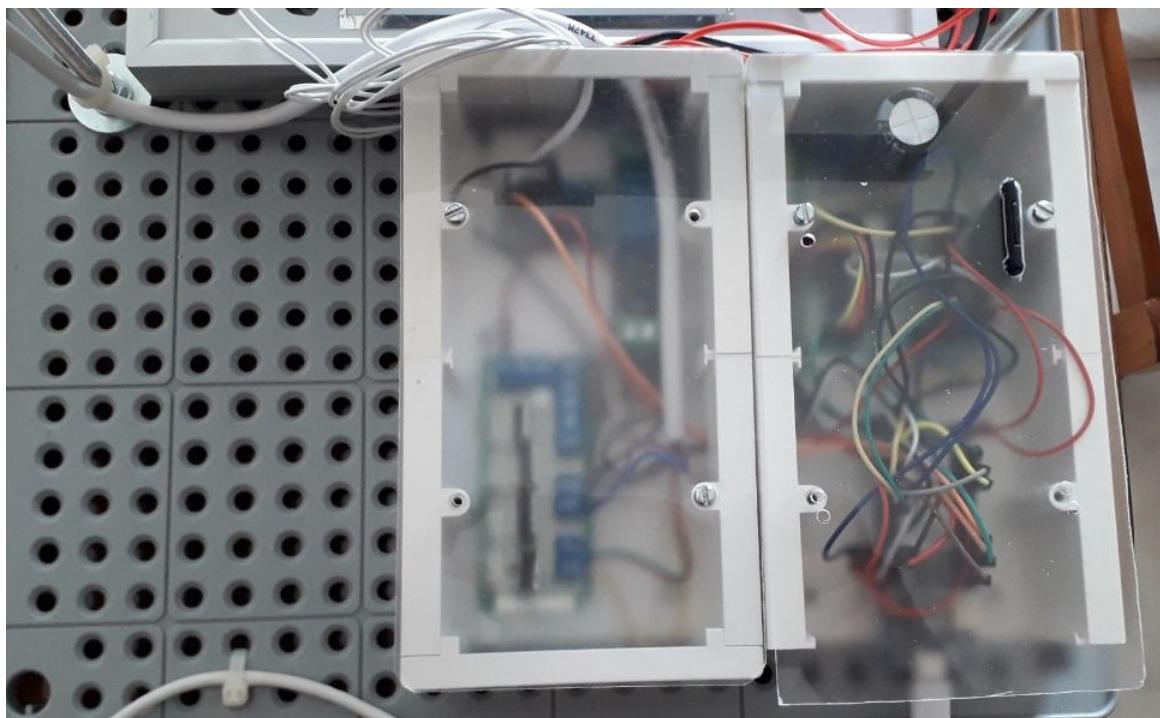


Figura 4.1.: Cajas de Conexiones del sistema cerradas. Caja 1, derecha. Caja 2, izquierda.

¹Se han utilizado cajas universales de instalación eléctrica en superficie.

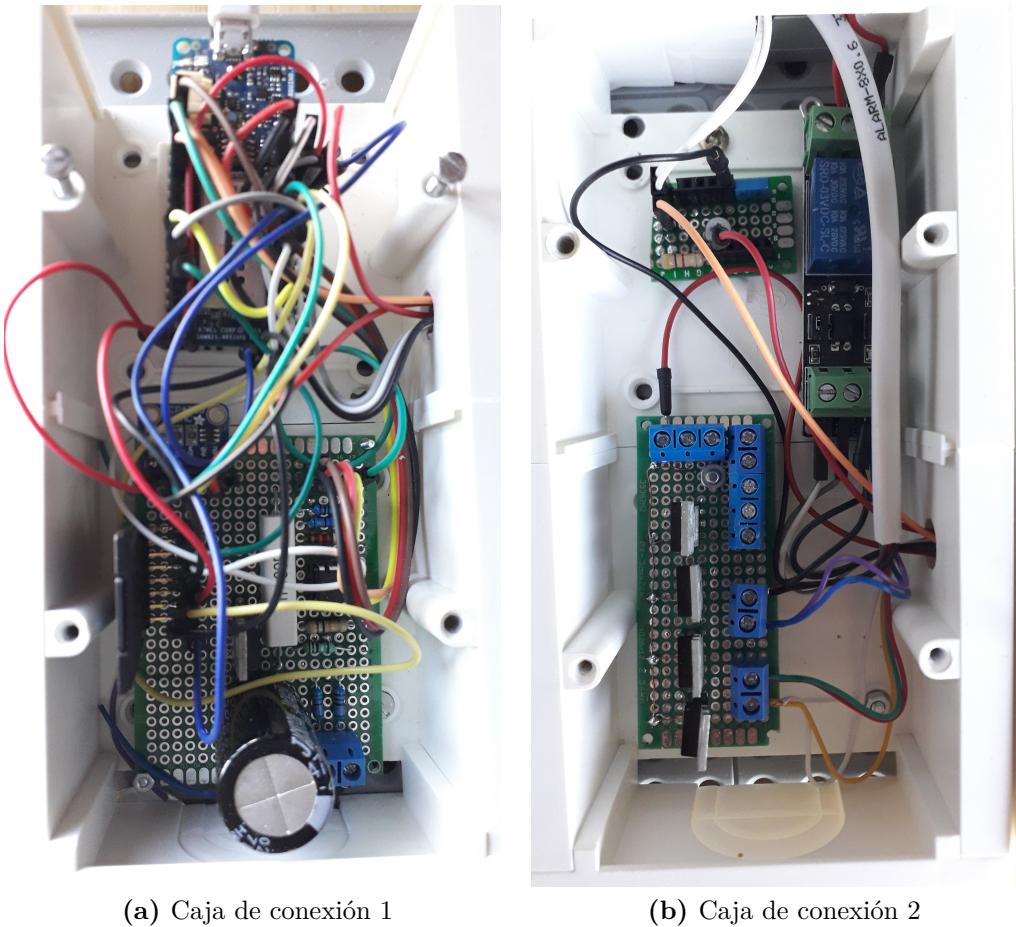


Figura 4.2.: Cajas de Conexiones del sistema sin cerrar.

4.1. Circuito Trazador de Curvas I-V

Mediante un sencillo circuito que utiliza el proceso de carga y descarga de un condensador para mostrar una carga variable en los terminales de la placa solar, es posible obtener la característica I-V de un módulo fotovoltaico. A este circuito se le llama circuito trazador de curvas I-V y obliga a trabajar al módulo en todo su rango de actuación [7]. Sigue el esquema electrónico de la Figura 4.3. El algoritmo de toma de muestras es el de la Figura 4.4. Se realizan como máximo 512 muestras, con el microcontrolador, el cual proporciona 12 bits de resolución.

Durante el proceso de carga, el divisor de tensión formado por las resistencias R_1 y R_2 permite la medida de la tensión entre los terminales de la placa y ajustar el valor medido al rango de entrada establecido por el microcontrolador. La resistencia R , en serie con el condensador, tiene el objetivo de medir la corriente suministrada por la placa y el amplificador operacional y su red de resistencias, ajusta la medida para obtener una tensión entre 0 y 3,3 V.

Los responsables del proceso de descarga son el transistor MOSFET y R_C , el cual se activa después de cada ciclo de medida, para descargar el condensador y que el circuito quede listo para una nueva toma de muestras. El circuito actúa según los siguientes puntos:

1. Situación inicial:

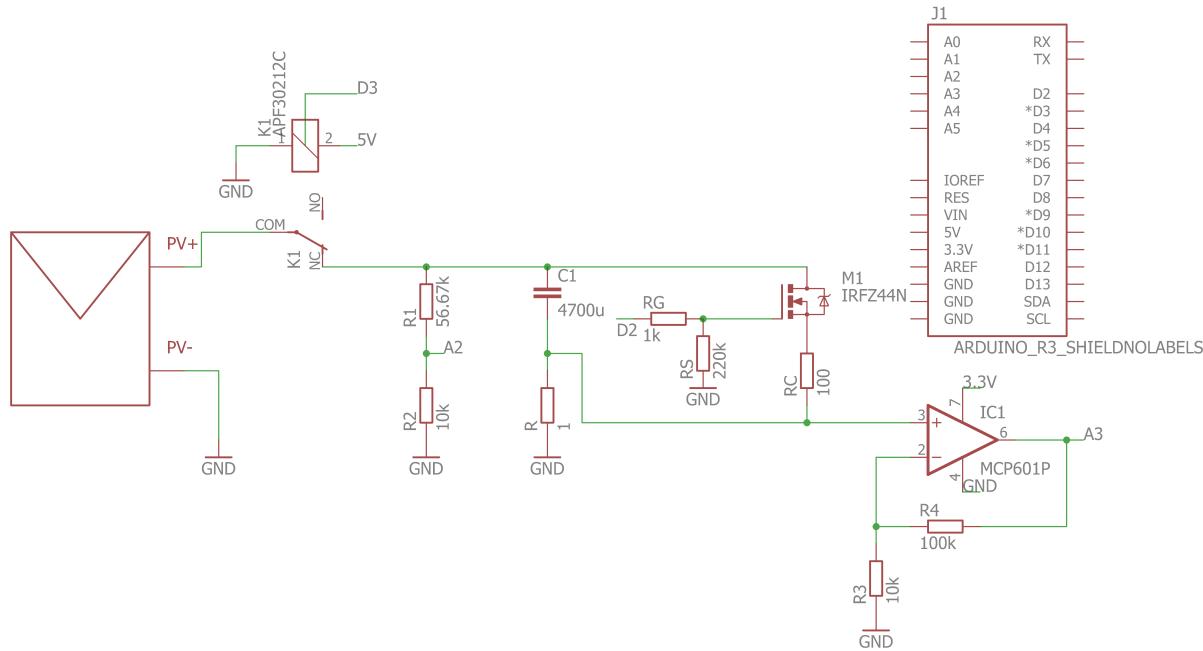


Figura 4.3.: Circuito Trazador de Curvas I-V[7]

- a) Condensador descargado.
 - b) Placa conectada al circuito a través del relé.
 - c) Transistor MOSFET abierto.
2. Toma de muestras
- a) Carga del condensador a través de la corriente proporcionada por la placa.
 - b) Toma de muestras a través de los circuitos de medida (red de resistencias R_1 y R_2 , resistencia R_C y amplificador operacional, con R_3 y R_4), hasta que la corriente es igual a cero.
3. Descarga del condensador
- a) Placa desconectada a través del relé.
 - b) MOSFET cerrado.
 - c) El condensador se descarga a través de la resistencia R_C .

En la placa del circuito trazador (Figura 4.5), se han incluido las conexiones de una tarjeta de memoria SD y de un conversor Digital/Analógico. La tarjeta SD se utiliza para guardar toda la información necesaria para la visualización de la página web y el conversor D/A en el circuito de regulación de iluminación.

4.2. Sensor de Temperatura

El sensor de temperatura utilizado es un termistor de $100\text{ }K\Omega$ NTC 3950 con cable alargador. Este sensor es del tipo *RepRap*, y se utiliza principalmente para medir temperatura en impresoras 3D. En este caso es utilizado para medir la temperatura de la placa solar y así realizar un mejor ajuste teórico y visualización de datos. Las características

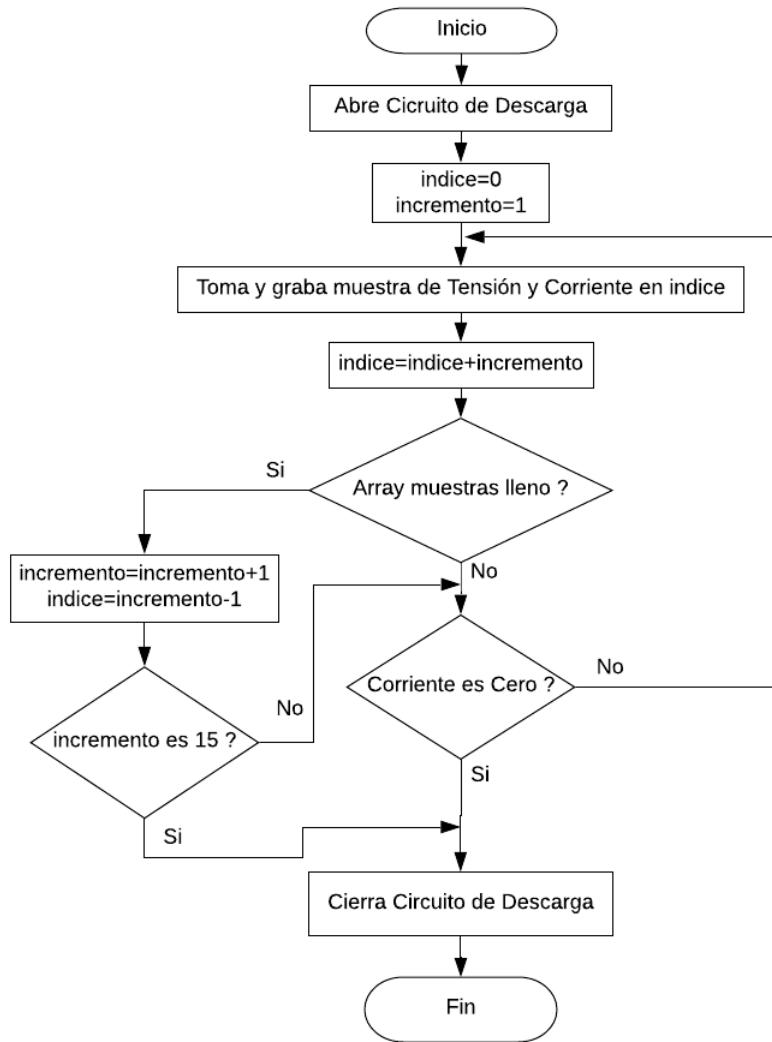


Figura 4.4.: Algoritmo de toma de muestras[7]

que presenta lo hacen adecuado para esta aplicación, debido a que es pequeño, flexible y está preparado para medir la temperatura en una superficie (cama caliente de impresoras 3D). Sus especificaciones técnicas son las siguientes:

- Resistencia: 100 Kohm + / -1 %
- B value: 3950 + / -1 %
- Precisión: + / -1 %
- Glass Packaging: 2.0 + / - 0.2mm
- Longitud: 32 + / - 5mm, Longitud de cable: 1 metro

Para obtener un valor de temperatura correcto, es necesario conectar un condensador en paralelo de $10 \mu F$ y una resistencia de $47 K\Omega$ en serie con la alimentación. Se elige este valor de resistencia para ampliar el rango de medida hasta aproximadamente $100 ^\circ C$, ya que la placa solar bajo iluminación natural, durante un tiempo prolongado puede alcanzar

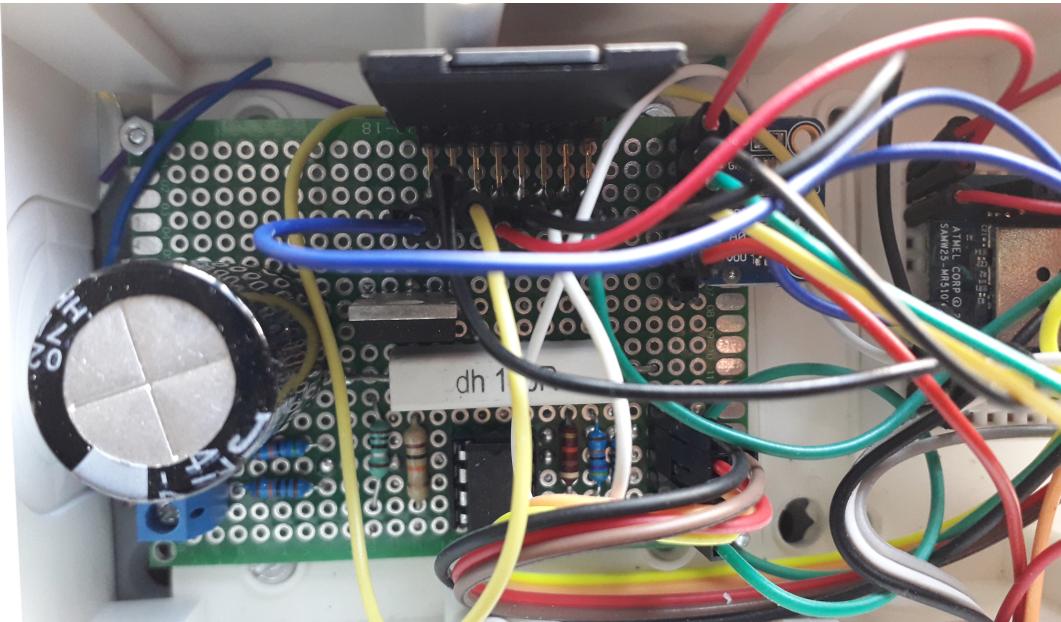


Figura 4.5.: Circuito trazador de curvas I-V, conversor D/A y adaptador tarjeta de memoria micro SD

este rango de temperatura. El diagrama de conexión se muestra en la Figura 4.6. En la Figura 4.8, se representa la tensión medida en el pin A3 y la resistencia que presenta el sensor, con la variación de temperatura, obtenidos a partir de los valores que proporciona su hoja de características. La tensión V_{out} , se calcula a partir del divisor de tensión formado por el propio sensor y la resistencia de $47 K\Omega$:

$$V_{out} = \frac{R_{th}}{R + R_{th}} \quad (4.1)$$

Al tener mayor rango de medida, se pierde algo de precisión, pero para esta aplicación en concreto, no es necesario tener un gran nivel de precisión para hacerse una idea de la temperatura que alcanza la superficie del panel solar. Con el objetivo de minimizar el espacio, disminuir el ruido y aumentar la eficiencia, y demás conflictos relacionados con malas conexiones, se ha integrado el circuito en una pequeña placa de circuito impreso (Figura 4.7).

A través del código fuente desarrollado para las impresoras 3D (que utiliza una tabla generada de valores en código python, para evitar realizar cálculos en tiempo real con el microcontrolador²), se configura el sensor dentro del código del sistema. Cuando se solicita una nueva toma de muestras, también se mide la temperatura en ese determinado momento, incluyendo ese valor en el array de datos enviados por el microcontrolador.

²El código para generar esta tabla, está disponible en: enlace

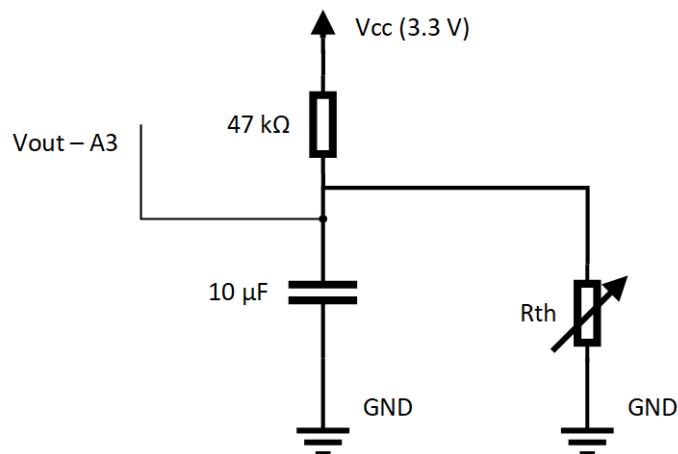
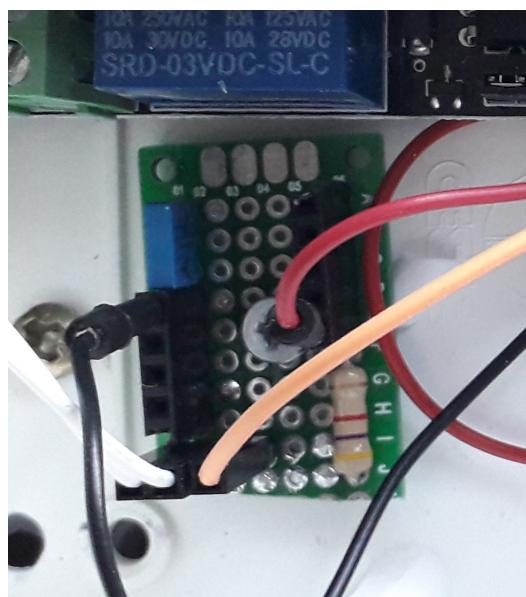
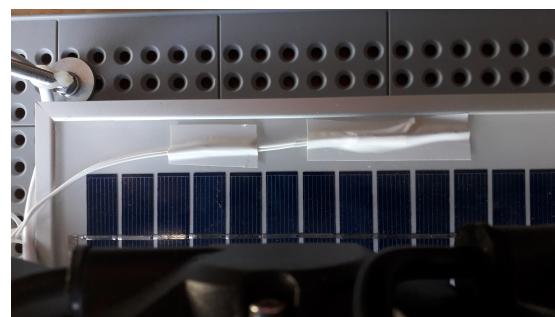


Figura 4.6.: Sensor de temperatura. Esquema de conexión



(a) Circuito de acondicionamiento del sensor de temperatura.



(b) Sensor de temperatura situado en la superficie del panel fotovoltaico.

Figura 4.7.: Sensor de Temperatura

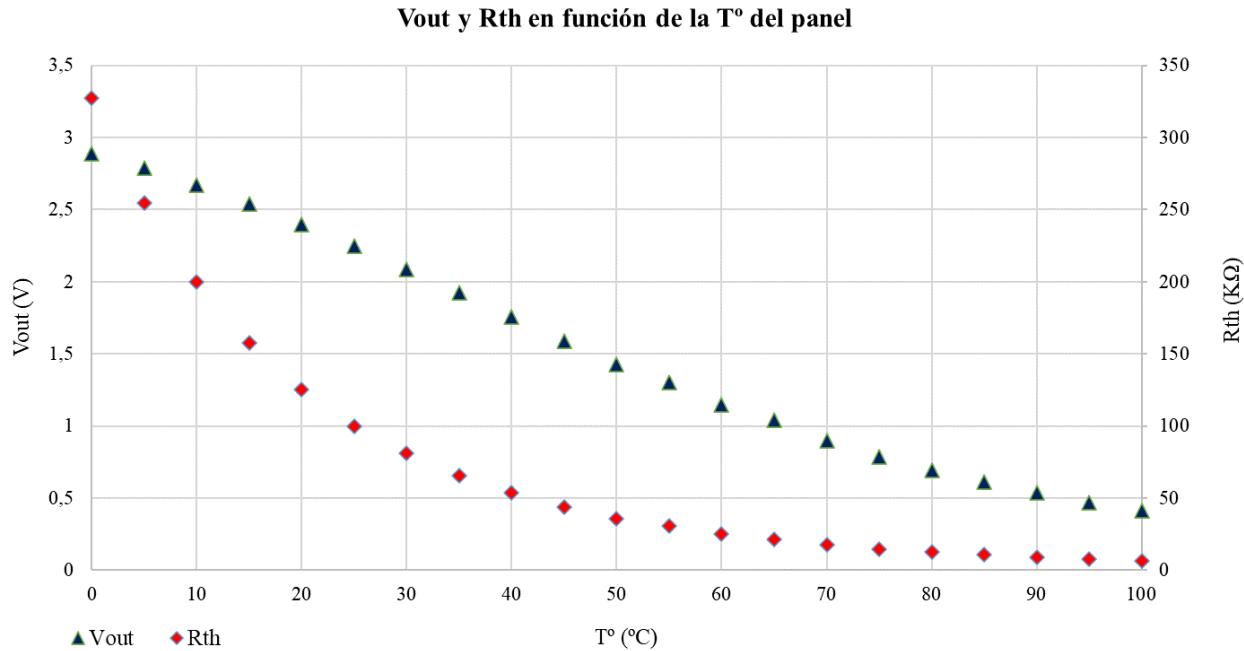


Figura 4.8.: Representación gráfica de la variación de la resistencia y la tensión Vout con la temperatura

4.3. Sistema de Iluminación Artificial

La estructura de iluminación artificial, se ha diseñado con el objetivo de tener el control sobre el nivel de iluminación incidente en la placa, y así tomar muestras en diferentes niveles de iluminación. Está formado por cuatro focos LED. Consta además de un circuito regulador, formado por un conversor D/A y transistores MOSFET, para poder controlar desde el microcontrolador el nivel de iluminación que los focos proporcionan. Éstos, se alimentan a partir de una fuente de alimentación externa de 12 V. Los focos LED elegidos tienen las siguientes características (Figura 4.9):

- 10000 Lúmenes
- Consumo 24 W
- Material: perfil de aluminio 6063, soporte de acero inoxidable y lente para PC.
- Protección IP68
- 32 LEDs
- Voltaje: 12V
- Vida útil: más de 50,000 horas
- Soporte ajustable, dirección y ángulo de la luz hasta 45 grados.



Figura 4.9.: Focos LED utilizados en el sistema de iluminación

4.3.1. Prototipos

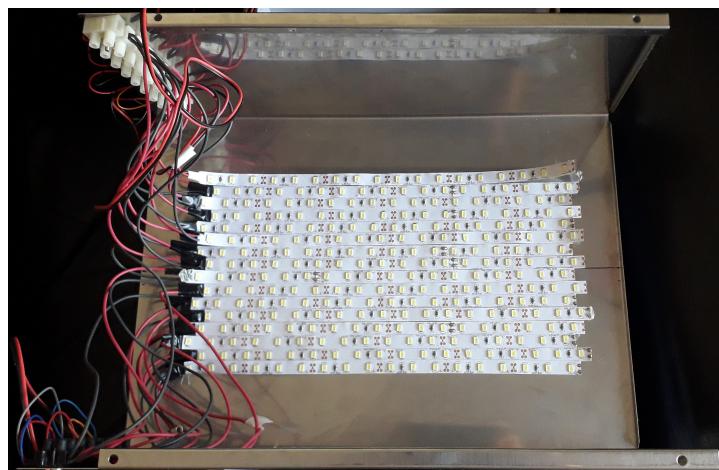
Para llegar al sistema de iluminación final y lograr un buen funcionamiento, ha sido necesario realizar varias pruebas de concepto y prototipos. En un principio, la idea fue conseguir una placa de circuito impreso. Sin embargo, debido a la situación de estado de alarma establecida desde el mes de Marzo, el acceso a ese material y a los laboratorios no fue posible (la soldadura de los LEDs en encapsulado SMD no era posible en un ambiente casero). Por ello, se decidió realizar un diseño más simple, partiendo de la idea principal de iluminar la placa solar desde arriba con una serie de LEDs.

El primer prototipo está formado por una tira de LEDs de 5 m de longitud (300 LEDs 56/30) y de la máxima potencia disponible en el mercado para este formato. Gracias al adhesivo que llevaba incorporada la cinta, se adhieren a una estructura de aluminio³. La tira de LEDs se divide en trozos de aproximadamente 15 LEDs y se conectan de forma alterna, mediante fichas de empalme para conseguir al menos tres niveles de iluminación: sólo dos tiras encendidas, cuatro encendidas, y así hasta todas las tiras encendidas para lograr el nivel máximo. Los LEDs estaban alimentados mediante una fuente externa de 12 V. Este prototipo se muestra en la Figura 4.10.

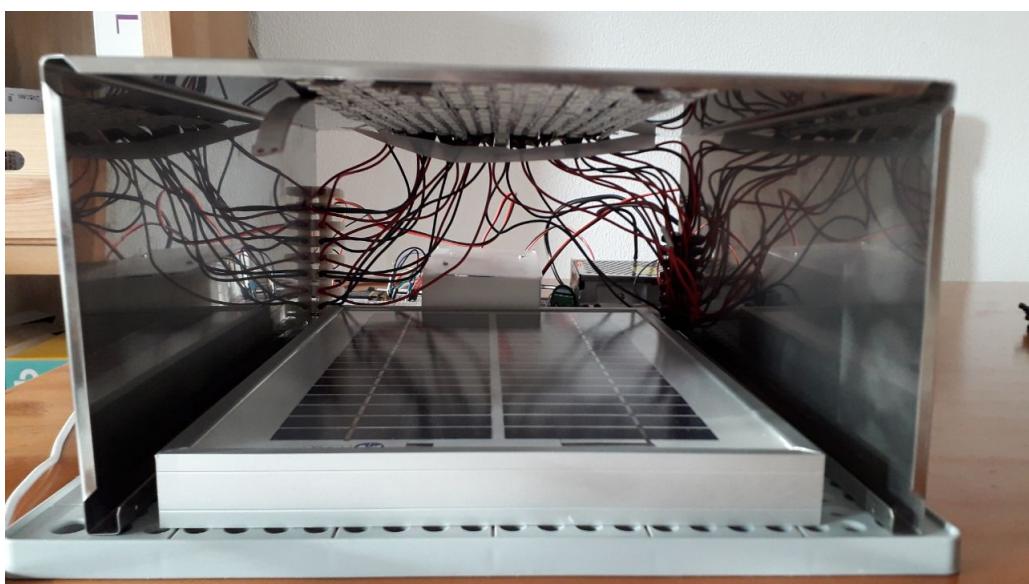
³La pieza de aluminio utilizada es una carcasa reciclada de un embellecedor de un extractor de cocina desmontado. Casualmente, sus proporciones y características son muy parecidas a las buscadas para esta aplicación.



(a) Vista superior. 1/2



(b) Vista superior. 2/2



(c) vista frontal del sistema

Figura 4.10.: Primer prototipo del sistema de iluminación.

A pesar de que la idea se ajustaba muy bien a la linea de trabajo inicial, estos LEDs no proporcionaban iluminación suficiente para que el circuito trazador pudiera captar una curva I-V. Por tanto, se descartó la idea.

Después del primer prototipo, estaba claro que se necesitaba más radiación para lograr al menos un rango parecido al de la radiación natural en un día soleado. Cambiando el concepto, se optó por dos focos LED de alta potencia (utilizados habitualmente para iluminación nocturna de maquinaria pesada), formados por 32 LEDs cada uno. Los resultados obtenidos fueron mucho mejores que el anterior prototipo y se lograba una intensidad de cortocircuito de hasta 0.05 A.

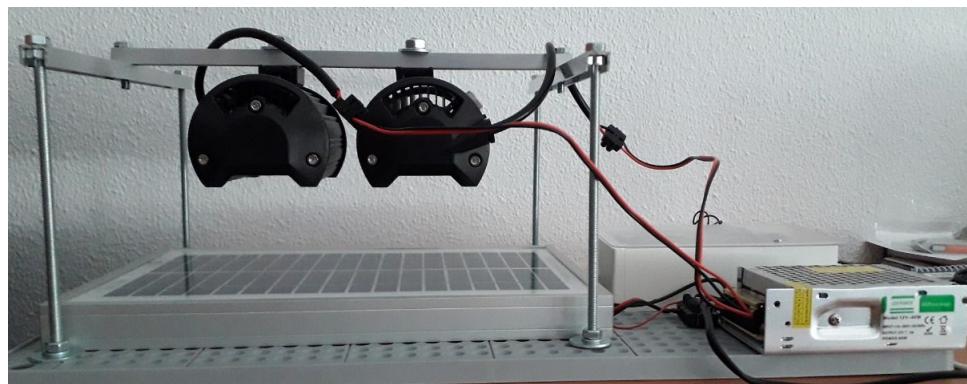


Figura 4.11.: Segundo prototipo de sistema de iluminación.

4.3.2. Diseño final

El diseño final está formado por cuatro focos LED de la marca WZTO, conectados a una fuente de alimentación externa de 12 V y controlados a través de MOSFET de potencia. La estructura es muy sólida y resistente. Está formada por varillas de aluminio fijadas a la base del sistema. El montaje permite la toma de muestras bajo iluminación artificial y también bajo iluminación natural, desmontando la parte de los focos LED. Tras las pruebas realizadas con este diseño, se verifica que las características de los focos elegidos son aptos para esta aplicación.

4.3.2.1. Circuito eléctrico

El circuito eléctrico de regulación está formado por cuatro MOSFETs IRFZ44N. Cada foco está conectado al drenador (*Drain*) de su transistor, las fuentes (*Source*) a tierra y las puertas (*Gate*) unidas y conectadas a la señal de control. Ésta última es proporcionada por un conversor D/A, que convierte un número entre 0 y 4096, proporcionado por el microcontrolador, en una señal analógica en voltios, que controla la cantidad de corriente que pasa por los focos, regulando así su nivel de iluminación.

4.3.2.2. Implementación física

Para evitar problemas de sobrecalentamiento y demás derivados, se ha realizado un montaje independiente del circuito en una placa de circuito impreso, incluyendo pines y conectores necesarios para centralizar las conexiones necesarias (Figura 4.14). Desde este circuito, se conectan los LEDs a través de una manguera de ocho cables, que llega hasta unas fichas de empalme situadas encima de la estructura, debido a que los cables de los focos LED son muy cortos y así se evitan posibles desconexiones (Figura 4.15).

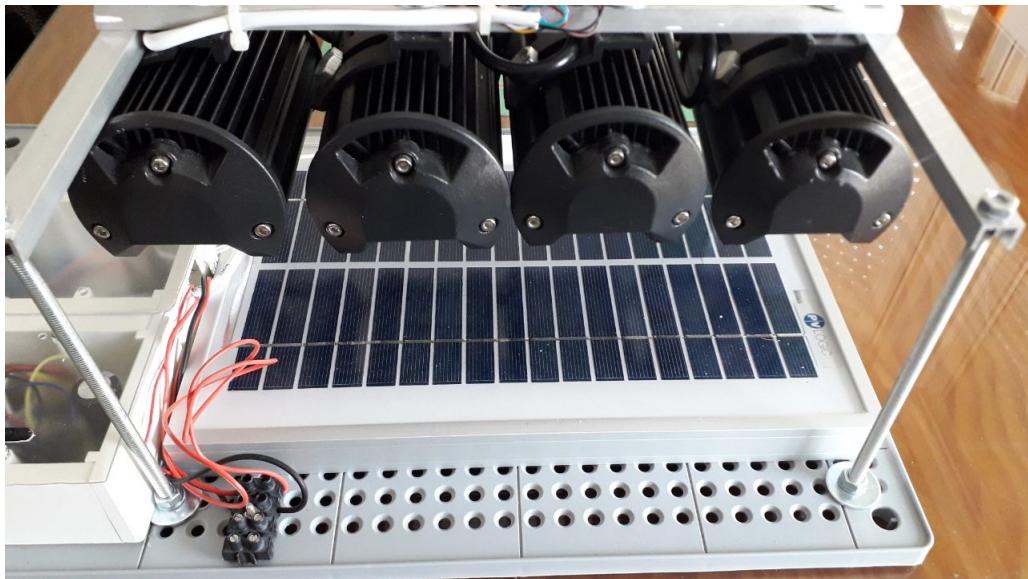


Figura 4.12.: Diseño final del sistema de iluminación.

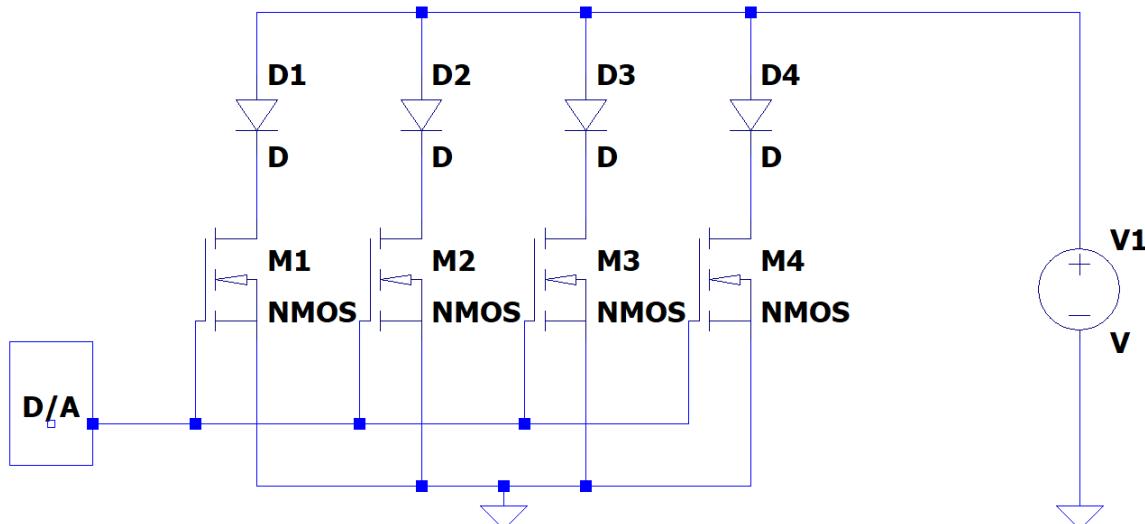


Figura 4.13.: Esquema eléctrico del circuito de regulación de la iluminación

El conversor D/A (Adafruit MCP4725 - 12-Bit DAC w/I2C), está situado junto al circuito trazador de curvas (Figura 4.5), así se ahorra espacio y está más cerca del microcontrolador para realizar las conexiones necesarias. Sólo la salida de control, es la que se conecta a este circuito regulador.

Los focos se conectan a una fuente de alimentación de 12 V y 20 A. Esta fuente se utiliza normalmente para alimentación de PC⁴. En la teoría estos focos necesitan cada uno sólo 2 A, por tanto, el sistema completo necesitaría 8 A de corriente para suministrar el máximo nivel de iluminación. Sin embargo, debido a que estas fuentes de alimentación son conmutadas, presentan un cierto valor de rizado, lo que hace que la luz que incide en la placa no sea del todo continua y las muestras no se tomen de forma correcta. Para evitar este tipo de problemas, la fuente de alimentación utilizada supera con creces la corriente que es necesaria para llegar hasta el máximo nivel de iluminación.

⁴Fuente de alimentación reciclada de un antiguo ordenador.

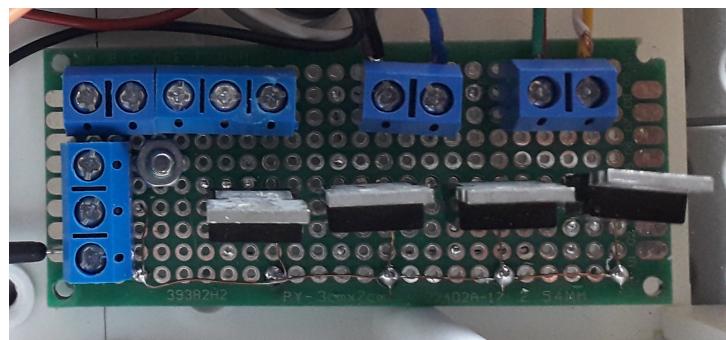


Figura 4.14.: PCB circuito regulador de iluminación

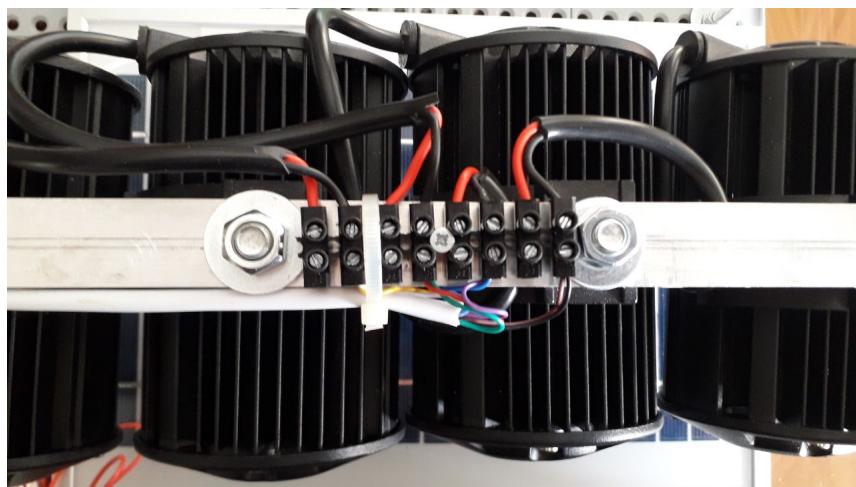


Figura 4.15.: Conexión Focos LEDs

4.3.2.3. Control nivel de iluminación

Para realizar el control del nivel de la iluminación desde el microcontrolador, únicamente es necesario inicializar los puertos de entradas y salidas, incluir la biblioteca *Adafruit* e iniciar el Bus I2C. A partir de ahí se escribe un número entre 0 y 4096 (12 bits de resolución) en el pin de control que está conectado al D/A. Éste se encarga de transformar esta señal en un valor analógico que entra a la puerta de los transistores MOSFET.

4.4. Sistema Completo

El sistema completo se muestra en la Figura 4.16 y está formado por los siguientes circuitos:

- Circuito trazador de curvas I-V (incluye relé), D/A y tarjeta de memoria SD
- Circuito regulador de iluminación
- Circuito acondicionador del sensor de temperatura

El control se realiza mediante lenguaje C a través del microcontrolador mkr1000. La tabla detallada de los componentes utilizados aparece en la Tabla 4.1.

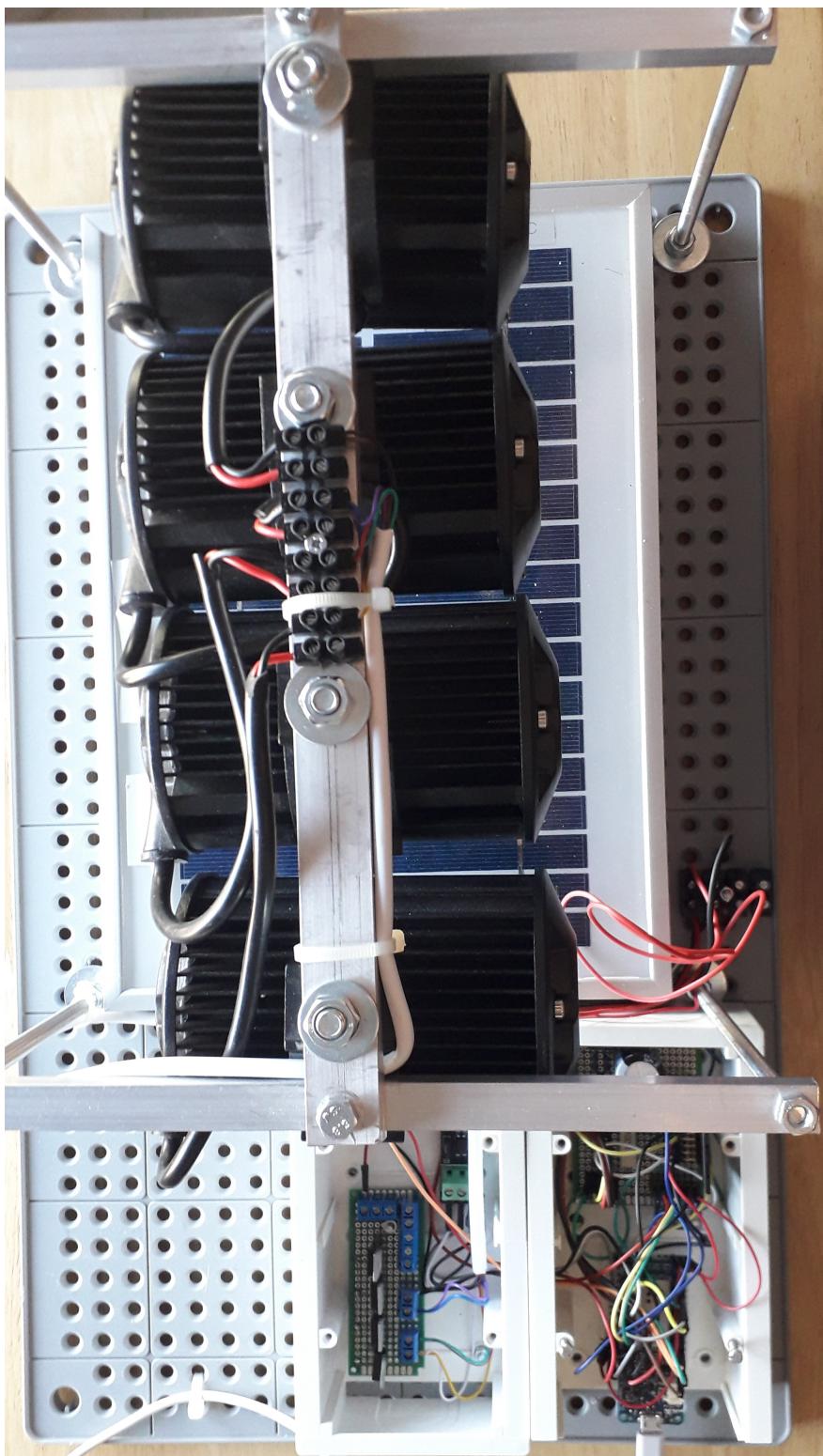


Figura 4.16.: Diseño final del sistema

Ud	Componente	Especificaciones
1	Microcontrolador	Arduino mkr1000
1	Placa solar	Panel PV Logic. 36 células en serie (Si)
1	Amplificador Operacional Rail-to-Rail	MCP601
4	Transistor MOSFET	IRFZ44N
1	Condensador Electrolítico	4700 uF, 35 V
8	Conectores	CON_TERMINAL_BLOCK_02-5MM
1	Relé	SRD-0 5V DC-SL-C, 10 A, 30 V DC
1	Resistencia 5 W	100 Ω
1	Resistencia 1/2 W	1 Ω
6	Resistencias 1/4 W	2x10K Ω , 56.6K Ω , 100K Ω , 1K Ω y 220K Ω
3	Placa Baquelita doble cara	Varios tamaños
1	Microcontrolador	Arduino mkr100
1	Resistencia 1/4 W	47K Ω
1	Condensador cerámico	10uF
1	Sensor de Temperatura	Termistor 100 K Ω NTC 3950
4	Focos LED	Focos LEDs WZTO, 96 W
2	Cajas para circuitería	caja universal de instalación eléctrica en superficie
9	Varillas de aluminio	4 varillas roscadas de métrica 6. 3 perfiles en forma de U
1	Tira de fichas de empalme y bridas	Varios tamaños
2	Tapaderas	Plástico semitransparente
1	Tarjeta miro SD con adaptador	SDCS2/16GB
1	Conversor D/A	Adafruit MCP4725 12-Bit DAC w/I2C

Cuadro 4.1.: Componentes del sistema

5. Visualización de datos y Sistema de comunicación

La visualización de datos se realiza a través del lenguaje HTML5¹, gráficos SVG², procesamiento de datos con javascript³ y AJAX⁴, y JSON⁵ para transferencia de datos. Por tanto, en este proyecto se han utilizado hasta tres lenguajes de programación diferentes: HTML, javascript y C. La interfaz gráfica sigue el esquema de la Figura 5.1 y permite:

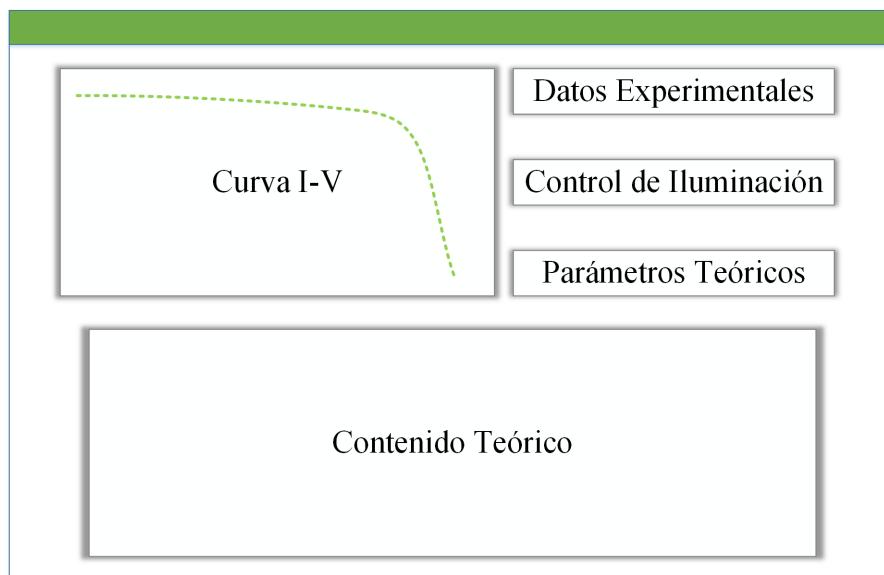


Figura 5.1.: Esquema de Página Web para visualización de datos.

- Toma de muestras en tiempo real, mostrando los valores experimentales obtenidos para I_{SC} , V_{OC} , I_{mp} , V_{mp} y T .
- Estimación de parámetros teóricos (I_g , I_{Sat} , A , R_S y R_P) mediante diferentes métodos de cálculo (Randy Williams, Xiao,Villava y Silva), visualizar la curva teórica para cada uno de los métodos y comparar los resultados teóricos con los experimentales.
- Ver contenido teórico del proyecto e introducción a las células solares, las curvas I-V y su modelado.

¹Última versión de HTML (Hypertext Markup Language). Es un lenguaje de marcado que sirve para definir la estructura y la semántica de una página web [8][2].

²Etiqueta del lenguaje HTML, para insertar gráficos y dibujos vectoriales escalables, basado en el estándar abierto SVG (Scalable Vector Graphics) [8].

³Lenguaje de programación interpretado y orientado a objetos. Se utiliza junto con HTML para el desarrollo de contenido web [8].

⁴Asynchronous JavaScript and XML. Herramienta web para generar aplicaciones dinámicas[8].

⁵JavaScript Object Notation. Formato para transferencia de datos [8].

La interfaz gráfica de la página web aparece en la Figura 5.2. En la gráfica aparecen representadas las muestras obtenidas y a la derecha una tabla con los valores experimentales y los parámetros calculados. Al avanzar hacia abajo en la página, aparece una introducción a la teoría de la característica I-V de una célula fotovoltaica, información del modelado de células y de los métodos que se han utilizado para la extracción de parámetros. En la Figura 5.3 se visualiza el aspecto de este contenido teórico.

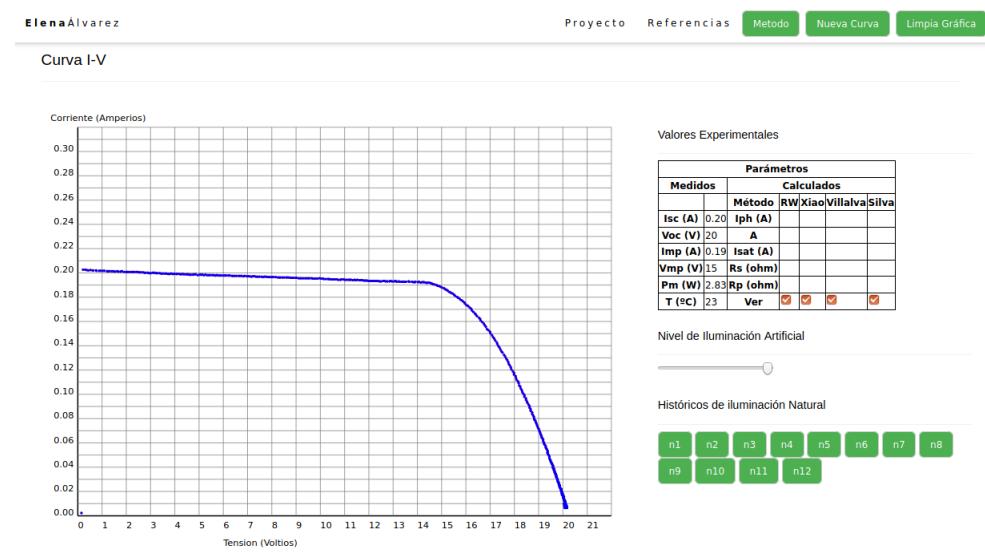


Figura 5.2.: Página Web para visualización de datos

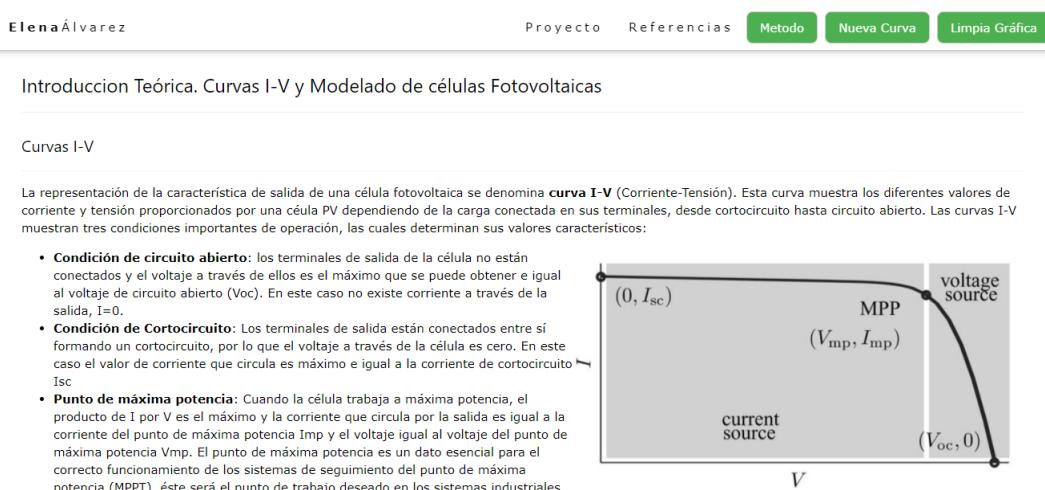


Figura 5.3.: Aspecto del inicio del contenido teórico de la página web del proyecto

El proceso para presentar una nueva muestra experimental sería el siguiente:

- El servidor detecta una nueva solicitud de muestreo
- Se reciben los datos en formato JSON en coma flotante

- Los datos se representan en un diagrama SVG a través de javascript y HTML

La comunicación se realiza a través del microcontrolador, que puede actuar como servidor y ser punto de acceso. Si se tiene control sobre el Router al que estará conectado el sistema, pueden darse varias opciones para acceder a la página web:

- El microcontrolador actúa como servidor y es un punto de acceso. De esta forma, no es necesaria la conexión a Internet para la visualización de datos
- Configurar el Router para asignarle al microcontrolador una dirección IP fija. En ese caso todos los dispositivos conectados al mismo Router que el sistema, pueden acceder a los datos
- Configurar direcciones IP y puerto 80 del Router dónde se encuentra el sistema. Así se puede acceder a los datos de forma remota, desde cualquier punto con conexión a Internet

En todos los casos para visualizar los datos en el navegador, se accede a través de la dirección IP del microcontrolador definida dependiendo de la configuración realizada. En la Figura 5.4 se muestra el diagrama de flujo del programa en C del sistema completo. El código fuente está incluido en el Apéndice B.

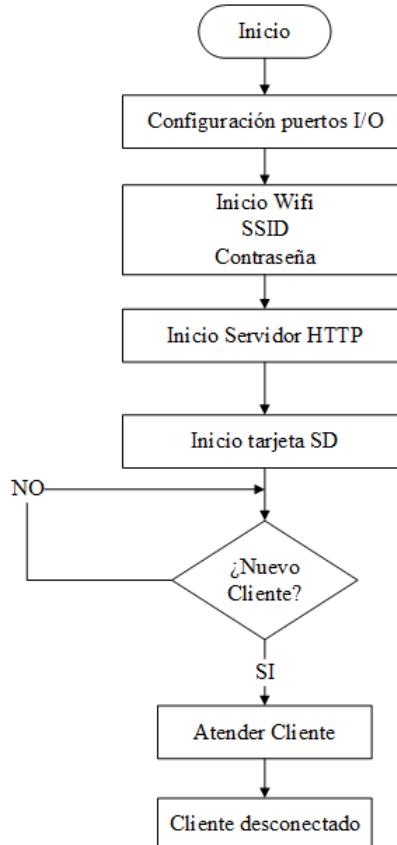
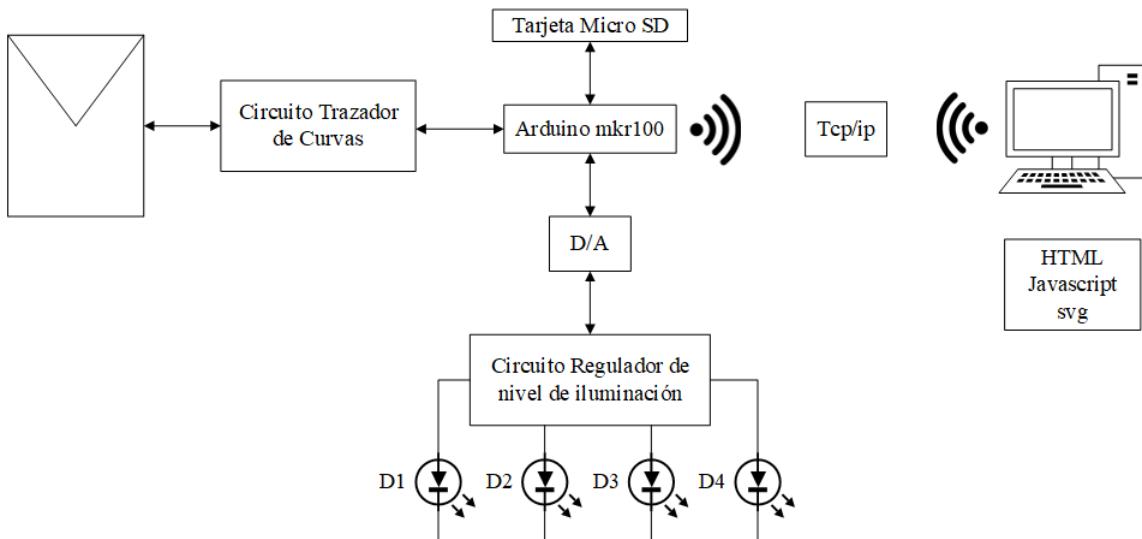


Figura 5.4.: Diagrama de flujo del sistema completo

En la Tabla 5.1 se muestran los protocolos de comunicación utilizados para cada capa y en la Figura 5.5, aparece un esquema visual de la conexión y comunicación de los diferentes elementos del sistema.

Capa	Protocolo de Comunicación
Físico y Enlace	wi-fi
Transporte y Red	tcp/ip
Aplicación	HTTP, SVG

Cuadro 5.1.: Niveles de comunicación y protocolos utilizados**Figura 5.5.:** Esquema de conexión y Comunicación

El sistema utiliza código HTML5 y hojas de estilo CSS⁶ para mostrar la interfaz gráfica. El gráfico dónde aparecen las curvas I-V es un gráfico dinámico creado en SVG, formato vectorial que contiene información de cada línea y de cada punto. Este gráfico se rellena de forma dinámica a través de código javascript, que se ejecuta en el cliente. La página web es interactiva y utiliza AJAX para actualizar los datos. Por tanto, la página web contiene código estático en HTML, el cual da forma a la página web, define la estructura, las imágenes y el contenido teórico. Tal y como aparece en la Figura 5.6, el código HTML hace llamadas a dos ficheros en código javascript, *iv.js* y *metodos.js*. Sus funciones aparecen descritas en la Tabla 5.2.

Cuando se lanza una nueva captura de muestras, el microcontrolador envía un array de datos en formato JSON, y el código javascript con SVG los convierte en una serie de datos legibles para el código principal de la página web, en HTML. Los ficheros de javascript y HTML están almacenados en la tarjeta de memoria, así como el fichero de estilo utilizado, los datos para el histórico de datos y las imágenes que se muestran en el contenido teórico.

El esquema de transmisión y visualización de datos, dónde el microcontrolador actúa como servidor, se muestra en la Figura 5.6. Tanto el código en HTML como en javascript se incluyen en el Apéndice A.

⁶Hoja de estilo basada en un estándar proporcionado por w3school [8].

Ficheros javascript	Funciones
iv.js	Función principal: se lanza cada vez que se carga la página, inicia liza la variable global, escala la gráfica I-V y lanza una primera captura de datos
	Función Nuevos Datos: lanza una nueva captura de datos a petición del cliente
	Función Crea Nueva Curva: se representan gráficamente las muestras tomadas de corriente y tensión y se muestran sus valores característicos
metodos.js	A partir de datos experimentales, ya sean en tiempo real o del históricos de datos, aplica los algoritmos correspondientes para calcular los parámetros teóricos a través de cuatro métodos. Cada método se ejecuta en una función diferente
	La función Curva Teórica: calcula la curva obtenida a partir de los parámetros teóricos y dibuja la curva teórica junto a los datos experimentales

Cuadro 5.2.: Funciones ficheros javascript

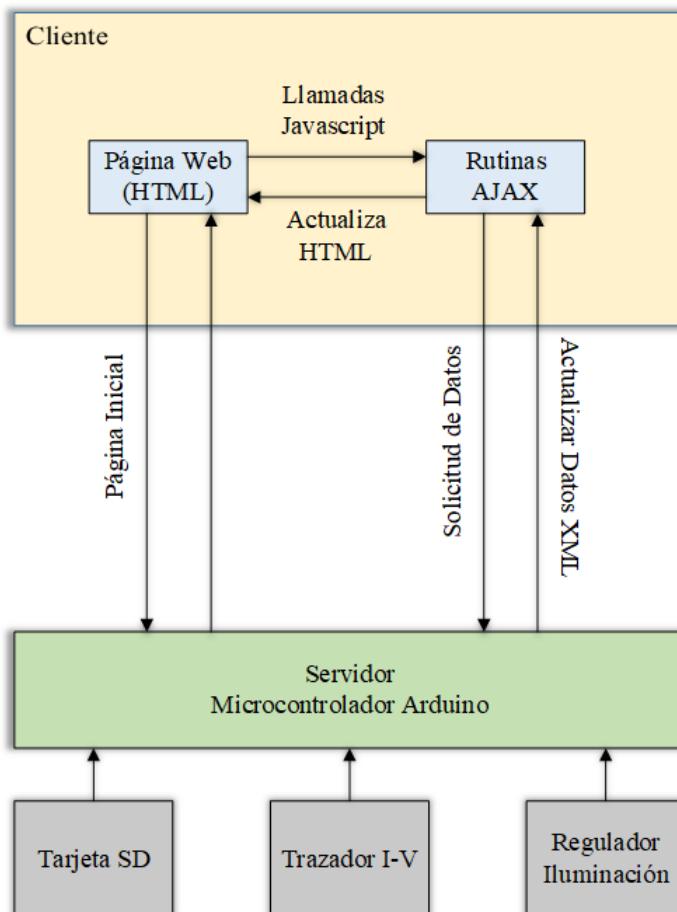


Figura 5.6.: Esquema de transmisión y visualización de Datos

6. Datos Experimentales

El sistema diseñado permite la toma de muestras, bajo diferentes niveles de iluminación artificial, y de iluminación natural, midiendo la temperatura de la placa en cada momento. Después, de realizar una serie de toma de muestras en todas estas condiciones, en este apartado se exponen los resultados obtenidos.

6.1. Iluminación Natural

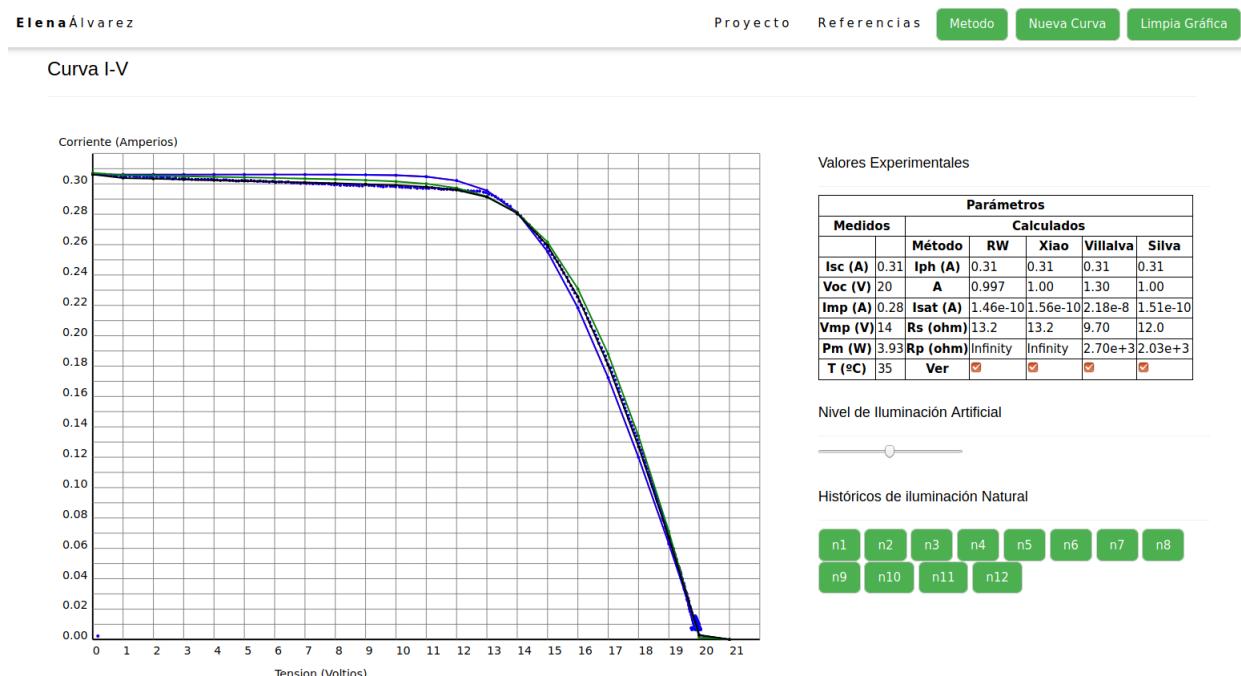


Figura 6.1.: Ejemplo curva obtenida en un día soleado al medio día y los parámetros teóricos calculados

Las curvas obtenidas bajo iluminación natural pueden ser visualizadas en la página web, a través de los botones habilitados en el aparado de histórico de datos, sin necesidad de desmontar el módulo de iluminación LED. En la Figura 6.1 aparece una curva experimental (en azul y en puntos) y cuatro curvas teóricas en diferentes colores. Esta curva corresponde a la muestra n12 del histórico de muestras, y presenta la máxima corriente de cortocircuito obtenida en iluminación natural. Al pulsar en Métodos (arriba a la derecha), se dibujan en la gráfica las cuatro curvas teóricas obtenidas. En la tabla que aparece a la derecha de la gráfica, se muestran los valores experimentales y los parámetros teóricos calculados. Mediante la opción Ver, se puede elegir qué curva teórica visualizar.

En un día soleado no muy caluroso, sobre el medio día y a una temperatura de aproximadamente 25 °C, la curva obtenida alcanza prácticamente los valores: $I_{SC} = 0,31\text{ A}$,

$V_{OC} = 20 V$. El algoritmo de toma de muestras funciona de forma excepcional con esta iluminación, proporcionando unos datos adecuados para realizar el cálculo teórico de parámetros. Cuando la iluminación natural es menor, se sigue obteniendo una buena curva, aunque no alcance valores altos de intensidad de cortocircuito y voltaje en circuito abierto. En este caso, la placa está expuesta directamente a la luz solar, y se calienta, llegando a superar los 30 °C. Este hecho afecta a su rendimiento, disminuyendo la V_{OC} proporcionada, es decir, la caída exponencial de la curva, se realiza más rápido, lo que da lugar a una disminución de su forma parabólica. Aún así, en un día totalmente soleado, a partir de las 12 de la mañana y orientando la placa correctamente, las curvas obtenidas rondan todas los 4-5 W de potencia.

6.2. Iluminación Artificial

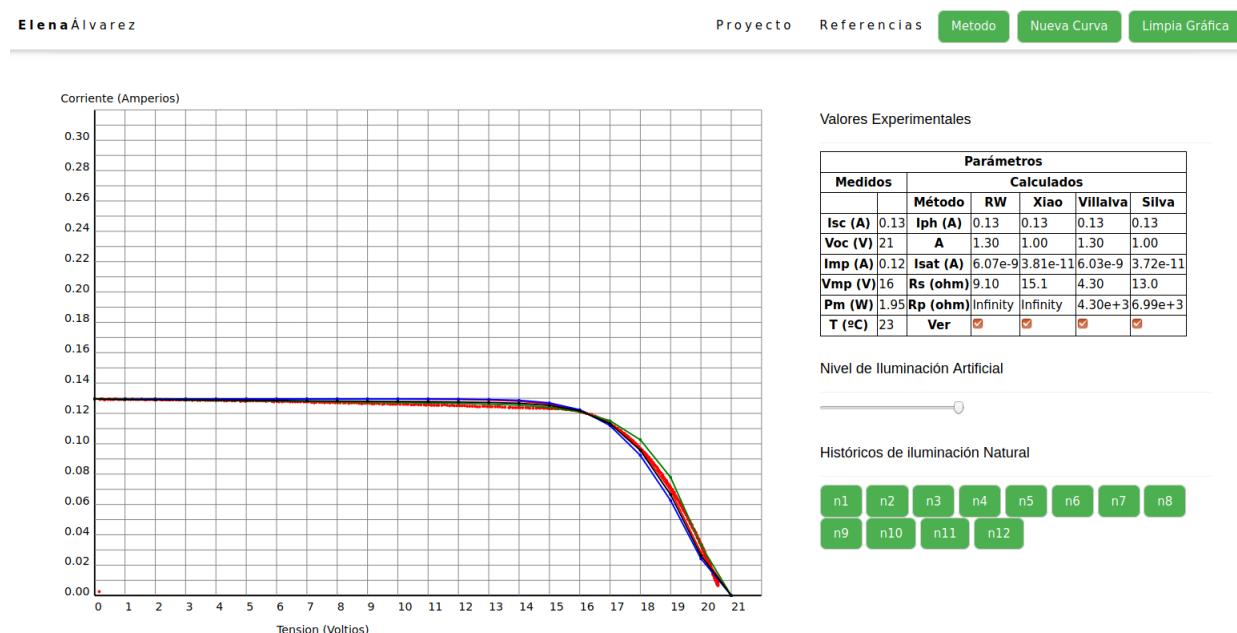


Figura 6.2.: Ejemplo curva obtenida con el máximo nivel de iluminación artificial y los parámetros teóricos obtenidos

En la Figura 6.2 aparece una curva experimental (en rojo y en puntos) y sus cuatro curvas teóricas calculadas en diferentes colores. Esta curva ha sido tomada bajo el máximo nivel de iluminación artificial que el sistema puede proporcionar.

A pesar de los esfuerzos realizados para que, mediante iluminación artificial, la placa solar alcance al menos la mitad de potencia suministrada, esto no ha sido posible. Bajo iluminación LED, la placa solar alcanza como mucho una $I_{SC} = 0,12 A$, un valor que no supera ni la mitad de su potencial ($I_{SC} = 0,32 A$). Respecto al voltaje en circuito abierto, puede llegar a alcanzar unos 21 V. En cuanto a la temperatura de la placa, bajo iluminación artificial, presenta valores mucho más pequeños, lo que hace que las curvas obtenidas, no presenten a penas deformación en la zona de caída exponencial y el voltaje de circuito abierto sea mayor. Sin embargo, debido a la cantidad de corriente que los focos LED necesitan para funcionar, y que, en bajos niveles caen en el circuito de regulación, hace que los transistores y demás componentes del sistema se calienten, lo que afecta también a la eficiencia.

Los métodos teóricos utilizados en la estimación de parámetros, bajo esta iluminación, a veces, presentan dificultades para seguir el algoritmo de cálculo. Esto es debido a que los focos LED, a fin de cuentas están conectados a una fuente de alimentación conmutada. Este hecho implica que existe un rizado en el voltaje suministrado, y esas pequeñas variaciones son captadas por las células solares.

7. Conclusión

El sistema diseñado y construido es capaz de obtener la característica I-V de un panel fotovoltaico bajo iluminación natural y artificial, medir su temperatura y mostrar los datos obtenidos en un entorno web accesible. En este entorno web se realiza la estimación de parámetros teóricos usando los algoritmos propuestos en cuatro artículos diferentes y compara gráficamente los resultados. Permite cambiar el nivel de iluminación artificial que incide en el panel, visualizar un histórico de muestras tomadas y una introducción teórica de los conceptos principales. Por tanto, todos los objetivos planteados se cumplen.

A lo largo del desarrollo del proyecto se han detectado una serie de posibles mejoras. Añadir un sensor de radiación solar para medir la radiación incidente, natural o artificial en la placa solar, y así poder comparar este valor con la energía obtenida. Se ha comprobado que el sistema de iluminación artificial mediante LED no genera ni la mitad de potencia que la radiación solar recibida en un día soleado. Esto es debido a que el espectro solar es muy diferente a la iluminación LED y tampoco ha sido posible concentrar mediante iluminación LED tanta energía como la recibida por el Sol en unidad de superficie.

Una solución para esta cuestión, puede ser utilizar las llamadas bombillas de espectro solar o espectro completo. Estas bombillas se utilizan en algunos invernaderos para acelerar el crecimiento de las plantas y vegetales, por ello proporcionan una luz algo más parecida a la luz solar.

El próximo paso para mejorar este sistema sería integrar todos los circuitos en una misma placa de circuito impreso y cerrar la estructura de iluminación artificial. Esto último permite integrar un ventilador que evite que los focos LED, los transistores y el sistema general alcance temperaturas elevadas, a parte de que lo haría mucho más compacto de lo que ya es actualmente.

Proporcionar herramientas prácticas para el aprendizaje de conceptos relacionados con las energías renovables, permite mostrar que los fundamentos teóricos no son tan complejos y que su utilización es más fácil de lo que parece. En este caso, visualizar las curvas I-V de un panel fotovoltaico concede información suficiente para realizar su modelado teórico, predecir su ciclo de vida y detectar posibles problemas internos y de degradación. Estos tres puntos son esenciales para el diseño de los seguidores del punto de máxima potencia, que se utilizan para maximizar el rendimiento de las grandes plantas fotovoltaicas, que, a fin de cuentas, son las que proporcionan realmente electricidad para las diferentes actividades del día a día y podrían sustituir en un futuro a los combustibles fósiles.

A. Código página visualización de datos vía web. Archivos HTML y javascript

En este anexo se incluyen los archivos de código correspondientes a:

- Código HTML desarrollado para mostrar los datos en una interfaz web.
- Código javascript desarrollado para mostrar gráficamente la curva I-V tomada y realizar los cálculos de parámetros teóricos. Se incluyen dos archivos: iv.js y metodos.js.
- Fichero CSS de estilo que da formato al contenido web.

```

1 <!-- TFM Elena Álvarez Castro -->
2 <!-- Página web para la visualización de datos -->
3 <!-- Página principal - Código HTML -->
4
5 <!DOCTYPE html>
6
7 <html>
8   <head>
9     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
10    <title>Diagrama IV</title>
11    <meta name="viewport" content="width=device-width, initial-scale=1">
12    <link rel="stylesheet" href="w3.css">
13    <!-- Para dibujar curva I-V experimental -->
14    <script src="iv.js"></script>
15    <!-- Para métodos teóricos -->
16    <script src="metodos.js"></script>
17  </head>
18 <body onload="Principal()">
19
20 <!-- Cabecera de la página -->
21   <div class="w3-top">
22     <div class="w3-bar w3-white w3-wide w3-padding w3-card">
23       <a href="#" class="w3-bar-item w3-button">Elena</a>
24       <!-- Float links to the right. Hide them on small screens -->
25       <div class="w3-right w3-hide-small">
26         <button class="w3-button w3-green w3-border w3-round-large"
27           type="button" onclick="nuevosDatosF()">>Metodo</button>
28         <button class="w3-button w3-green w3-border w3-round-large"
29           type="button" onclick="nuevosDatos()">>Nueva Curva</button>
30         <button class="w3-button w3-green w3-border w3-round-large"
31           type="button" onclick="LimpiaDatos()">>Limpia Gráfica</button>
32         <a href="#projects" class="w3-bar-item w3-button">Proyecto</a>
33         <a href="#about" class="w3-bar-item w3-button">Referencias</a>
34       </div>
35     </div>
36   </div>
37 </div>
38
39 <!-- Header -->
40 <header class="w3-display-container w3-content" style="max-width:1500px;" id="home">
41
42 <!-- Contenido principal -->
43
44   <div class="w3-content w3-padding" style="max-width:1564px">
45     <div class="w3-container w3-padding-32">
46       <h3 class="w3-border-bottom w3-border-light-grey w3-padding-16">Curva I-V</h3>
47     </div>
48   <!-- Grafica -->
49   <img id="IV" alt="Grafica de la curva I-V experimental"/>
50   <div class="w3-third">
51     <table border="1" style="width:100%; border-collapse: collapse; border: none; margin-bottom: 10px;">
52       <thead>
53         <tr>
54           <th>Parámetros</th>
55           <th>Medidos</th>
56           <th>Calculados</th>
57         </tr>
58       </thead>
59       <tbody>
60         <tr>
61           <td>Tensión (Volts)</td>
62           <td>Corriente (Ampérios)</td>
63           <td>Resistencia (ohm)</td>
64         </tr>
65       </tbody>
66     </table>
67     <table border="1" style="width:100%; border-collapse: collapse; border: none; margin-bottom: 10px;">
68       <thead>
69         <tr>
70           <th>Valores Experimentales y parámetros teóricos-->
71           <th>Parámetros</th>
72           <th>Medidos</th>
73           <th>Calculados</th>
74         </tr>
75       </thead>
76       <tbody>
77         <tr>
78           <td>Isc (A)</td>
79           <td>Isc</td>
80           <td>igRw</td>
81           <td>igV</td>
82           <td>igS</td>
83         </tr>
84         <tr>
85           <td>Voc (V)</td>
86           <td>Voc</td>
87           <td>ax</td>
88           <td>av</td>
89           <td>as</td>
90         </tr>
91         <tr>
92           <td>Imp (A)</td>
93           <td>Imp</td>
94           <td>isRw</td>
95           <td>isv</td>
96           <td>isS</td>
97         </tr>
98         <tr>
99           <td>Vmp (V)</td>
100          <td>Vmp</td>
101          <td>rsRw</td>
102          <td>rsX</td>
103          <td>rsV</td>
104          <td>rsS</td>
105        </tr>
106        <tr>
107          <td>Pm (W)</td>
108          <td>Pm</td>
109          <td>rpRw</td>
110          <td>rpX</td>
111          <td>rpV</td>
112          <td>rpS</td>
113        </tr>
114        <tr>
115          <td>Rp (ohm)</td>
116          <td>Rp</td>
117          <td>rpRw</td>
118          <td>rpX</td>
119          <td>rpV</td>
120          <td>rpS</td>
121        </tr>
122        <tr>
123          <td>T (°C)</td>
124          <td>T</td>
125          <td>Ver</td>
126          <td><input type="checkbox" checked="" id="verRw" onclick="VerOcultarLineaMetodo('verRw')"/></td>
127          <td><input type="checkbox" checked="" id="verX" onclick="VerOcultarLineaMetodo('verX')"/></td>
128          <td><input type="checkbox" checked="" id="verV" onclick="VerOcultarLineaMetodo('verV')"/></td>
129          <td><input type="checkbox" checked="" id="vers" onclick="VerOcultarLineaMetodo('verS')"/></td>
130        </tr>
131      </tbody>
132    </table>
133  </div>
134
135 <!-- Nivel iluminación artificial -->
136 <div class="w3-border-bottom w3-border-light-grey w3-padding-16">Nivel de Iluminación Artificial</div>
137   <input id="barraLuz" onchange="ActualizaLuz()" type="range" min="1" max="100" value="50">
138
139 <!-- Históricos de iluminación natural-->
140 <div class="w3-border-bottom w3-border-light-grey w3-padding-16">Históricos de iluminación Natural</div>
141
142   <table border="1" style="width:100%; border-collapse: collapse; border: none; margin-bottom: 10px;">
143     <thead>
144       <tr>
145         <th>Historico</th>
146         <th>Valor</th>
147       </tr>
148     </thead>
149     <tbody>
150       <tr>
151         <td>n0930.txt</td>
152         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n1" onclick="nuevosDatosHistorico('nF0930.txt')"/></td>
153       </tr>
154       <tr>
155         <td>n0935.txt</td>
156         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n2" onclick="nuevosDatosHistorico('nF0935.txt')"/></td>
157       </tr>
158       <tr>
159         <td>n0945.txt</td>
160         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n3" onclick="nuevosDatosHistorico('nF0945.txt')"/></td>
161       </tr>
162       <tr>
163         <td>n1000.txt</td>
164         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n4" onclick="nuevosDatosHistorico('nF1000.txt')"/></td>
165       </tr>
166       <tr>
167         <td>n1015.txt</td>
168         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n5" onclick="nuevosDatosHistorico('nF1015.txt')"/></td>
169       </tr>
170       <tr>
171         <td>n1040.txt</td>
172         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n6" onclick="nuevosDatosHistorico('nF1040.txt')"/></td>
173       </tr>
174       <tr>
175         <td>n1055.txt</td>
176         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n7" onclick="nuevosDatosHistorico('nF1055.txt')"/></td>
177       </tr>
178       <tr>
179         <td>n1100.txt</td>
180         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n8" onclick="nuevosDatosHistorico('nF1100.txt')"/></td>
181       </tr>
182       <tr>
183         <td>n1115.txt</td>
184         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n9" onclick="nuevosDatosHistorico('nF1115.txt')"/></td>
185       </tr>
186       <tr>
187         <td>n1130.txt</td>
188         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n10" onclick="nuevosDatosHistorico('nF1130.txt')"/></td>
189       </tr>
190       <tr>
191         <td>n1145.txt</td>
192         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n11" onclick="nuevosDatosHistorico('nF1145.txt')"/></td>
193       </tr>
194       <tr>
195         <td>n1200.txt</td>
196         <td><input type="button" class="w3-button w3-green w3-border w3-round-large" value="n12" onclick="nuevosDatosHistorico('nF1200.txt')"/></td>
197       </tr>
198     </tbody>
199   </table>
200 </div>

```

```

131
132
133 </div> <!-- termina parte derecha de la grafica -->
134
135
136 <!-- Contenido teórico de la pagina -->
137
138 <div class="w3-content w3-padding" style="max-width:1500px">
139
140 <div class="w3-container w3-padding-32" id="projects">
141   <h3 class="w3-border-bottom w3-border-light-grey w3-padding-16">Proyecto</h3>
142   <p>
143     Este proyecto está formado por un sistema trazador de curvas I-V de una placa solar fotovoltaica. Mediante un sistema de iluminación artificial,
144     es posible cambiar el nivel de iluminación, para apreciar los cambios que se realizan en la característica I-V de la placa. También cuenta con un sensor de temperatura,
145     para visualizar la temperatura en °C que tiene la placa y así, observar su influencia en las curvas I-V.
146
147     Por otro lado, están implementados cuatro métodos teóricos de extracción de parámetros. Cada uno de ellos calcula una curva teórica a través de
148     un sistema de cálculo iterativo.
149   </p>
150
151 <!-- Introducción teórica a las curvas IV y al modelado de células -->
152 <div class="w3-container w3-padding-32" id="projects">
153   <h3 class="w3-border-bottom w3-border-light-grey w3-padding-16">Introducción Teórica. Curvas I-V y Modelado de células Fotovoltaicas</h3>
154
155   <h4 class="w3-border-bottom w3-border-light-grey w3-padding-16">Curvas I-V</h4>
156
157   <p>
158
159     La representación de la característica de salida de una célula fotovoltaica se denomina <b>curva I-V</b> (Corriente-Tensión).
160     Esta curva muestra los diferentes valores de corriente y tensión proporcionados por una célula PV dependiendo de la carga
161     conectada en sus terminales, desde cortocircuito hasta circuito abierto.
162     Las curvas I-V muestran tres condiciones importantes de operación,
163     las cuales determinan sus valores característicos:
164   </p>
165
166   <div>
167     
168   </div>
169
170   <ul>
171     <li>
172       <b>Condición de circuito abierto</b>: los terminales de salida de la célula no están conectados y el voltaje
173       a través de ellos es el máximo que se puede obtener e igual al voltaje de circuito abierto (Voc).
174       En este caso no existe corriente a través de la salida, I=0.
175     </li>
176     <li>
177       <b>Condición de Cortocircuito</b>: Los terminales de salida están conectados entre sí formando un cortocircuito,
178       por lo que el voltaje a través de la célula es cero. En este caso el valor de corriente que circula es máximo e igual
179       a la corriente de cortocircuito Isc
180     </li>
181     <li>
182       <b>Punto de máxima potencia</b>: Cuando la célula trabaja a máxima potencia, el producto de I por V es el máximo y
183       la corriente que circula por la salida es igual a la corriente del punto de máxima potencia Imp y el voltaje igual
184       al voltaje del punto de máxima potencia Vmp.
185       El punto de máxima potencia es un dato esencial para el correcto funcionamiento de los sistemas de seguimiento
186       del punto de máxima potencia (MPPT), éste será el punto de trabajo deseado en los sistemas industriales de generación
187       de energía eléctrica.
188     </li>
189   </ul>
190
191   <p>
192     Las características anteriores también dependen de la radiación solar y la Temperatura. Fundamentalmente, la radiación hace que
193     aumente la corriente de cortocircuito (Isc) proporcionada por la placa, de forma que a mayor radiación mayor corriente.
194     Un aumento de la temperatura provoca una disminución en la zona de máxima potencia MPP. Los fabricantes suelen proporcionar
195     estos valores para unas condiciones estándar (CEM o STC, en inglés, Standard Test Conditions. Se refieren a una irradiancia de 1000 W/m2,
196     distribución espectral de 1.5 y 25°C de temperatura) de funcionamiento.
197
198   </p>
199
200   <p>
201     Con estas tres condiciones de operación se obtienen los valores característicos de la curva I-V de una célula PV: Isc, Voc, Pm, Imp, Vmp.
202     Para hallar los valores de la curva de un módulo fotovoltaico, basta con multiplicar estos valores por el número de células en serie
203     y en paralelo.
204   </p>
205
206 <h4 class="w3-border-bottom w3-border-light-grey w3-padding-16">Modelado de Células Fotovoltaicas</h4>
207   <p>
208
209     El modelado de células fotovoltaicas se refiere al cálculo de la curva I-V partiendo de datos
210     obtenidos de forma experimental o de las hojas características proporcionadas por el fabricante, de una forma teórica
211     mediante algoritmos matemáticos.
212   </p>
213
214   <p>
215     Un modelado teórico es esencial para la simulación, diseño, evaluación, control y optimización de los sistemas
216     fotovoltaicos; siendo crucial para los sistemas de <b>seguimiento del punto máxima potencia</b> y para los cálculos
217     relacionados con la eficiencia. El problema de encontrar los parámetros
218     adecuados para el modelado teórico de células solares se denomina <i>Problema de Estimación de Parámetros del Modelo
219     de una Célula Fotovoltaica</i> [5]. La búsqueda de estos parámetros es uno de los grandes retos que presenta el campo
220     de las energías renovables.
221   </p>
222
223   <p>
224     Para que un modelo sea aceptable, debe incluir un circuito equivalente adecuado, siendo sus parámetros correspondientes
225     suficientemente exactos. Un método de estimación de parámetros para las células fotovoltaicas debe tener los siguientes
226     requisitos:
227   </p>
228   <ul>
229     <li>
230       Debe proporcionar parámetros lo más exactos posibles. La curva teórica obtenida debe ajustarse de forma correcta a
231       los datos experimentales.
232     </li>
233     <li>
234       Debe ser válido, proporcionando resultados prácticamente iguales cuando es aplicado varias veces para el modelado
235       de una misma curva experimental.
236     </li>
237     <li>
238       Debe proporcionar resultados de igual calidad para cualquier curva a modelar.
239     </li>
240     <li>
241       Debe tener un tiempo de cálculo aceptable.
242     </li>
243   </ul>
244   <p>
245     Existen diferentes modelos de Circuitos para la simulación del comportamiento de los paneles PV.
246   </p>
247
248 <!-- Modelos de circuitos para la obtención de parámetros -->
249
250   <div class="w3-row-padding w3-grayscale">
251     <div class="w3-col l3 m6 w3-margin-bottom">
252       <img alt="John" style="width:68%">
253       <h3>Modelo Ideal</h3>
254       <p class="w3-opacity">Modelo de tres parámetros: Ig, A e Isat</p>
255
256       <p> Sin la presencia de la irradiancia solar, una célula solar se comporta como un diodo de unión PN simple,
257       y la curva I-V está dada por la ecuación de Shockley, donde ID y VD son la corriente y el voltaje del
258       diodo respectivamente.
259       </p>
260       <p>
261         <img alt="Ecación Modelo Ideal" style="width:85%">
262       </p>

```

```

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392

```

<p>

-
- Is o Isat: corriente de saturación del diodo
-
-
- A: factor de Idealidad del diodo
-
-
- q: valor absoluto de la carga eléctrica del electrón
-
-
- T: temperatura en Kelvin
-
-
- K:constante de Boltzman
-

</p>

<p>

El modelo ideal no es utilizado en procesos de simulación, sólo para fines teóricos y explicativos.

</p>

<p><button class="w3-button w3-light-grey w3-block">Contact</button></p>

Modelo de un Único Diodo.

<p class="w3-opacity">Con efecto de resistencia serie Rs. Modelo de cuatro parámetros: Ig, A, Isat y Rs. </p>

<p>En cualquier panel PV existen resistencias óhmicas de las conexiones y contactos, las cuales dependen de la temperatura, y resistencias internas de la unión P-N de las células que forman el módulo PV, las cuales aumentan cuando la irradiancia disminuye [4]. Estás pérdidas resitivas se modelan como una resistencia en serie Rs. A pesar de que el modelo de un diodo con una resistencia serie imita el comportamiento físico de una célula solar mejor que el modelo ideal, sigue sin ser demasiado exacto.

</p>

<p>

</p>

<p><button class="w3-button w3-light-grey w3-block">Contact</button></p>

Modelo de un Único Diodo.

<p class="w3-opacity"> Con efecto de resistencia en paralelo Rp.Modelo de cinco parámetros:Ig, A, Isat, Rs y Rp.</p>

<p>En este caso, para obtener mayor exactitud, se tiene en cuenta una corriente de pérdidas en la unión P-N, representada como una resistencia en paralelo Rp, que se añade al circuito, además de la Rs [5]. El modelo resultante es el más utilizado en el modelado de células solares debido a que ofrece una gran exactitud y su cálculo es relativamente simple. Sin embargo, la exactitud para valores muy bajos de irradiancia no llega a ser aceptable.

</p>

<p>

</p>

<p><button class="w3-button w3-light-grey w3-block">Contact</button></p>

Modelo de dos diodos.

<p class="w3-opacity">Modelo de siete parámetros:Ig, Rs, rp, A1,Isat1, A2, Isat2.</p>

<p>Este modelo añade el efecto de la pérdida de corriente en el proceso de recombinación, como un segundo diodo D2. La ventaja es que a diferencia de los modelos de un solo diodo, proporciona exactitud cuando hay valores bajos de irradiancia [5]

</p>

<p><button class="w3-button w3-light-grey w3-block">Contact</button></p>

Comparación de Modelos

<h4 class="w3-border-bottom w3-border-light-grey w3-padding-16">Comparación de Modelos</h4>

<p>

La ecuación que describe de forma matemática el comportamiento de las células solares depende del modelo elegido para la célula solar y hay que tener en cuenta las siguientes cuestiones [1]:

-
- Los modelos de cinco parámetros son los que más se utilizan, debido a su relación entre exactitud y complejidad.
-
-
- El modelo de extracción de parámetros más simple es el de tres parámetros (Modelo Ideal), pero sólo se aplican en explicaciones básicas de funcionamiento o para fines teóricos.
-
-
- El modelo de dos diodos es mucho más exacto que el de un solo diodo, sin embargo la complejidad de sus algoritmos no suele salir rentable para ciertas condiciones de iluminación.
-

</div>

<!-- Metodos teóricos -->

Método Randy Williams

<p>Este método no considera el efecto de una resistencia en paralelo.

Utiliza los puntos característicos de una curva I-V; Isc, Voc, Vmp e Imp para realizar la aproximación de estos parámetros, realizando la aproximación $I = I_{pv} - I_{sc}$, resuelve de forma analítica ecuaciones,

<p><button class="w3-button w3-light-grey w3-block">Referencia</button></p>

Método Xiao

<p>Este método desprecia el efecto de la resistencia en paralelo.

Se basa en que en un módulo PV la derivada de la potencia con respecto al voltaje en el punto de máxima potencia es cero, definiendo a partir de este hecho, los cuatro parámetros mediante ecuaciones analíticas.

<p><button class="w3-button w3-light-grey w3-block">Referencia</button></p>

Método Silva

<p class="w3-opacity">Método de cinco parámetros</p>

```

393     <p>Realiza un barrido completo de los posibles valores de A (desde 1 hasta 2 en pasos de 0.01) y de Rs (desde 0 a 2 ohm en pasos de 1 ohm, aunque este rango depende del área de la célula). Los otros tres parámetros Ig, Isat y Rp se calculan de forma analítica.</p>
394     </p>
395     <p><button class="w3-button w3-light-grey w3-block">Referencia</button></p>
396 </div>
397 <div class="w3-col l3 m6 w3-margin-bottom">
398     
399     <h3>Método Villalva</h3>
400     <p class="w3-opacity">Método de cinco parámetros</p>
401     <p>En este caso, intenta ajustar los valores de Rs y de Rp, para que el punto de máxima potencia de la curva modelada, sea igual al punto de máxima potencia de la curva experimental.</p>
402     <p><button class="w3-button w3-light-grey w3-block">Referencia</button></p>
403 </div>
404 </div>
405 </div>
406 <!-- Panel solar utilizado -->
407 <div class="w3-container w3-padding-32" id="projects">
408     <h3>Características del panel solar utilizado</h3>
409
410     El prototipo que se desarrolla en este proyecto, se ha llevado a cabo mediante un panel de la serie PV Logic, de Solar Thecnology International (disponible en su página oficial). Formado por 36 células en serie, de silicio monocristalino. Presenta las siguientes características:
411     <ul>
412         <li>Alta eficiencia.</li>
413         <li>Construcción robusta que la hace apta para aplicaciones de exterior y resistente al agua.</li>
414         <li>Incluye un diodo que trabaja en inversa, para evitar la retroalimentación.</li>
415         <li>La degradación de las células PV no será mayor del 20% en 20 años.</li>
416     </ul>
417
418     Valores característicos en condiciones estándar de medida:
419
420     <table style = "border: 1px solid black; border-collapse: collapse;">
421
422         <tr> <th>Potencia Máxima (Pm)</th> <th>5 W</th> </tr>
423         <tr> <th>Voltagje de Máxima Potencia (Vmp)</th> <th>17.5 V</th> </tr>
424         <tr> <th>Intensidad de Máxima Potencia (Imp)</th> <th>0.29 A</th> </tr>
425         <tr> <th>Intensidad de Cortocircuito (Isc)</th> <th>0.32 A</th> </tr>
426         <tr> <th>Voltagje de Circuito abierto (Voc)</th> <th>22 V</th> </tr>
427         <tr> <th>Temperatura de Operación</th> <th>-40 a 85 °C</th> </tr>
428         <tr> <th>Dimensiones</th> <th>306x218x25 mm</th> </tr>
429     </table>
430
431     <!-- /p> -->
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451     </div>
452 </div>
453
454 <!-- Fin contenido teórico -->
455 </header>
456
457
458 <!-- Footer -->
459 <footer class="w3-center w3-black w3-padding-16">
460     <p>Powered by <a href="https://www.w3schools.com/w3css/default.asp" title="W3.CSS" target="_blank" class="w3-hover-text-green">w3.css</a></p>
461 </footer>
462
463 </body></html>
464

```

```

/*
 * TFM Elena Álvarez Castro
 * Página web para la visualización de datos
 * Extracción de datos y representación gráfica de la Curva I-V - Código javascript
 * Fichero iv.js
 */

var gIV;
var temperaturaCentigrados;
var muestrasF = []; //array de muestras en coma flotante
                    //cada elemento tiene v,i,p
                    //Tension, corriente, potencia calculada

var colores = ["red", "blue", "yellow", "cyan", "magenta",
               "orange", "purple", "aqua", "aquamarine", "blueviolet", "brown", "chocolate"];
var indiceColor=0;
function CambiaTamanoGrafica(grSvg) {
    grSvg.setAttribute("width", 300);
    grSvg.setAttribute("height", 400);
}
function Principal() {

    gIV = document.getElementById("IV");
    gIV.setAttribute("padding-top", "20%");
    //gIV.setAttribute("width", 1000);
    //gIV.setAttribute("height", 700);

    CreaLineasEscala();
    // Pedimos una curva inicial
    nuevosDatos();
}

function CreaLineasEscala(){
    var contLineaScala=document.getElementById("lineasEscala");

    for (var i=100; i<980; i+=40){
        var newLinea = document.createElementNS("http://www.w3.org/2000/svg",
                                                'line');
        newLinea.setAttribute("x1",i);
        newLinea.setAttribute("y1",670);
        newLinea.setAttribute("x2",i);
        newLinea.setAttribute("y2",30);
        contLineaScala.appendChild(newLinea);
    }
    for (var i=30; i<651; i+=20){
        var newLinea = document.createElementNS("http://www.w3.org/2000/svg",
                                                'line');
        newLinea.setAttribute("x1",60);
        newLinea.setAttribute("y1",i);
        newLinea.setAttribute("x2",940);
        newLinea.setAttribute("y2",i);
        contLineaScala.appendChild(newLinea);
    }
    for (var i=60,j=0; i<940; i+=40, j++) {
        var newTexto = document.createElementNS("http://www.w3.org/2000/svg",
                                                'text');
        newTexto.setAttribute("x",i);
        newTexto.setAttribute("y",690);
        newTexto.textContent= j.toString();
        gIV.appendChild(newTexto);
    }

    for (var i=670, j=0.00; i>31; i-=40, j=j+0.02000) {
        var newTexto = document.createElementNS("http://www.w3.org/2000/svg",
                                                'text');
        newTexto.setAttribute("x",20);
        newTexto.setAttribute("y",i);
        newTexto.textContent= j.toFixed(2);
        gIV.appendChild(newTexto);
    }
}

function nuevosDatos() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            //document.getElementById("demo").innerHTML +=
            //this.responseText;
            var myArr = JSON.parse(this.responseText);
            temperaturaCentigrados = myArr.temperatura;
            if (temperaturaCentigrados==undefined) //Ojo
                temperaturaCentigrados = 23;
            CreaNuevaCurva(myArr.muestras);
        }
    };
    xhttp.open("GET", "nDatosFloat.txt", true);
    xhttp.send();
}

//Historico de datos con iluminacion Natural
function nuevosDatosHistorico(nombreFichero) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {

```

```

        var myArr = JSON.parse(this.responseText);
        temperaturaCentigrados = myArr.temperatura;
        if (temperaturaCentigrados === undefined)
            temperaturaCentigrados = 23;
        CreaNuevaCurva(myArr);
    }
};

nuevoNombre = nombreFichero;
xhttp.open("GET", nuevoNombre, true);
xhttp.send();
}

function LimpiaDatos() {
    gIV = document.getElementById("IV");
    var listaHijos=gIV.getElementsByTagName("circle");

    for (var i = 0; i < listaHijos.length; i++) {
        listaHijos[i].remove();
    }
}

function CreaNuevaCurva(arr) {
    muestrasF=arr;
    //ordenamos el array
    muestrasF.sort(function(a,b){
        return(a.v-b.v);
    });
    //Calculo de valores experimentales para poner en la tabla
    for(i = 0, pMax=0, iMax=0, vMax=0; i < muestrasF.length; i++){
        muestrasF[i].p=muestrasF[i].v*muestrasF[i].i;
        corriente[i] =muestrasF[i].i;
        tension[i] =muestrasF[i].v;
        potencia[i] =muestrasF[i].v*muestrasF[i].i;
        if (potencia[i] > pMax){
            pMax=potencia[i];
            Imp =muestrasF[i].i;
            Vmp =muestrasF[i].v;
        }
        if (corriente[i]>iMax) iMax = corriente [i];
        if (tension[i]>vMax) vMax = tension [i];
    }
    Isc = iMax;
    Voc = vMax;

    document.getElementById("Isc").innerHTML = Isc.toPrecision(2);
    document.getElementById("Voc").innerHTML = Voc.toPrecision(2);
    document.getElementById("Imp").innerHTML = Imp.toPrecision(2);
    document.getElementById("Vmp").innerHTML = Vmp.toPrecision(2);
    document.getElementById("T").innerHTML = temperaturaCentigrados;
    document.getElementById("Pm").innerHTML = pMax.toPrecision(3);

    //
    for(i = 0; i < arr.length; i++){
        yPixel = arr[i].i*2000; //(640/0.32)
        yPixel = 670-yPixel;
        xPixel = arr[i].v*(880.0/22.0);
        xPixel +=60;
        var newCirculo = document.createElementNS("http://www.w3.org/2000/svg", 'circle'); //
        newCirculo.setAttribute("fill", colores[indiceColor]);
        newCirculo.setAttribute("cx",xPixel);
        newCirculo.setAttribute("cy",yPixel);
        newCirculo.setAttribute("r",2); //Set path's data
        gIV.appendChild(newCirculo);
    }
    indiceColor++;
    if (indiceColor==colores.length) indiceColor = 0;
}

//Control de nivel iluminación artificial

function ActualizaLuz(){
    var nivel = document.getElementById("barraLuz").value;
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var respuesta = JSON.parse(this.responseText);
        }
    };
    var url="nL.txt?nv="+nivel;
    xhttp.open("GET", url, true);
    xhttp.send();
}

```

```

/*
 * TFM Elena Álvarez Castro
 * Página web para la visualización de datos
 * Calculo de parametros teoricos - Código javascript
 * Fichero metodos.js
 *
 */
///////////////////////////////
var gmIV; //elemento grafico para dibujar

var colores = ["red", "blue", "yellow", "cyan", "magenta",
               "orange", "purple", "aqua", "aquamarine", "blueviolet", "brown", "chocolate"];
var indiceColor=0;

var corriente = []; // Array con corrientes
var tension   =[];
var potencia  =[];

//lineas de representacion de los metodos
//Para visualizar.
var lMetodoRW;
var lMetodoX;
var lMetodoV;
var lMetodoS;

q=1.6022e-19;

function MetPrincipal() {

    gmIV = document.getElementById("IV");
}

function nuevosDatosF() {
    if (muestrasF.length == 0){
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                var myArr = JSON.parse(this.responseText);

                MetCreaNuevaCurva(myArr);
            }
        };
        xhttp.open("GET", "nDatosFloat.txt", true);
        xhttp.send();
    }
    else
        MetCreaNuevaCurva(muestrasF);
}

function MetCreaNuevaCurva(arr) {

    muestrasF = arr; // muestrasF es global
    // calculamos la potencia de cada muestra
    for(i = 0, pMax=0, iMax=0, vMax=0; i < muestrasF.length; i++){
        muestrasF[i].p=muestrasF[i].v*muestrasF[i].i;
        corriente[i] =muestrasF[i].i;
        tension[i]  =muestrasF[i].v;
        potencia[i] =muestrasF[i].v*muestrasF[i].i;
        if (potencia[i] > pMax){
            pMax=potencia[i];
            Imp =muestrasF[i].i;
            Vmp =muestrasF[i].v;
        }
        if (corriente[i]>iMax) iMax = corriente [i];
        if (tension[i]>vMax)   vMax = tension [i];
    }
    Isc = iMax;
    Voc = vMax;

    //Imp y Vmp ya estan calculado en el bucle anterior.
    valores=MetodoRW(Isc,Voc,Imp,Vmp);
    vTeoricos=CurvaTeorica(valores.Ig,valores.A,valores.Isat,valores.Rs,valores.Rp,Voc,"red");
    lMetodoRW = DibujaLineasGrafica(vTeoricos.Imetodo,vTeoricos.Vmetodo,"red","RW");

    valores = MetodoXiao(Isc,Voc,Imp,Vmp);
    vTeoricos=CurvaTeorica(valores.Ig,valores.A,valores.Isat,valores.Rs,valores.Rp,Voc,"blue");
    lMetodoX = DibujaLineasGrafica(vTeoricos.Imetodo,vTeoricos.Vmetodo,"blue","Xiao");

    valores = MetodoVillalva(Isc,Voc,Imp,Vmp);
    vTeoricos=CurvaTeorica(valores.Ig,valores.A,valores.Isat,valores.Rs,valores.Rp,Voc,"green");
    lMetodoV = DibujaLineasGrafica(vTeoricos.Imetodo,vTeoricos.Vmetodo,"green","Villalva");

    valores = MetodoSilva(Isc,Voc,Imp,Vmp);
    vTeoricos=CurvaTeorica(valores.Ig,valores.A,valores.Isat,valores.Rs,valores.Rp,Voc,"black");
    lMetodoS = DibujaLineasGrafica(vTeoricos.Imetodo,vTeoricos.Vmetodo,"black","Villalva");
}

function MetodoRW(Isc,Voc,Imp,Vmp){
    var valoresFormula = new Object();

```

```

Iph=Isc;
numeradorVt = (2*Vmp-Voc) * (Isc-Imp);
denominadorVt=Isc+((Isc-Imp)*Math.log(1-(Imp/Isc)));
Vt=numeradorVt/denominadorVt;
Io=Isc*Math.exp(-Voc/Vt);
terminoLog = Math.log(1-(Imp/Isc));
Rs=(Vt*terminoLog)+Voc-Vmp;
Rs=Rs/Imp;
Rs=Rs/Imp;

Ig=Iph;
Isat=Io;
T=302;
K=1.38065e-23; //Constante de Bolzam
Ns=36 //Numero de celdas en serie
q=1.6022e-19;
A=Vt*q/(Ns*K*T);
//llama a curva teorica
var Rp=Infinity; // Estimado
actualizaTablaMetodos("Rw",Ig,A,Isat,Rs,Rp);
valoresFormula.metodo = "Rw";
valoresFormula.Ig=Ig;
valoresFormula.A = A;
valoresFormula.Isat = Isat;
valoresFormula.Rs = Rs;
valoresFormula.Rp=Rp;
return(valoresFormula);
//CurvaTeorica(Ig,A,Isat,Rs,Rp,Voc);
}

function MetodoSilva(Isc, Voc, Imp, Vmp){
var valoresFormula = new Object();
var matrizMAEP = [];
var valoresMetodo = [];

T=302;
K=1.38065e-23;
Ns = 36;
q= 1.6022e-19;

Ig= Isc;
columnaMaep = 1;
indiceMaep = 0;
for (A=1; A<2.5; A=A+0.5){
    for (Rs=0.5; Rs<15; Rs=Rs+0.5){
        resistenciaSerie=Rs;
        Vt=(Ns*A*K*T)/q;
        RpTerminoExp=(Math.exp((Vmp+Imp*Rs)/Vt)-1)/(Math.exp(Voc/Vt)-1);
        RpNumerador=(Voc*RpTerminoExp)-Vmp-(Imp*Rs);
        RpDenominador=Imp+Ig*(RpTerminoExp-1);
        Rp=RpNumerador/RpDenominador;
        if(Rp<0){
            Rp=Rp;
            continue;
        }
        Isat = Ig-(Voc/Rp);
        Isat = Isat/(Math.exp(Voc/Vt)-1);
        //Calculamos un array con tensiones y corrientes según formulas
        //Tomamos las tensiones de las muestras

        for (i=0; i<muestrasF.length; i++){
            valoresMetodo[i]=new Object();
            valoresMetodo[i].v=muestrasF[i].v;
            valoresMetodo[i].i=NaN;
        }
        //Calculamos la intensidad asociada a cada tension según formula
        valoresMetodo[0].i=Isc;
        for (i=1; i<valoresMetodo.length; i++){
            Ip=valoresMetodo[i-1].i;
            while(isNaN(valoresMetodo[i].i)){
                Inew = Ig - Isat * (Math.exp((valoresMetodo[i].v+Ip*Rs)/Vt)-1)-
                    (valoresMetodo[i].v+Ip*Rs)/Rp;
                if (Inew>Ip)
                    valoresMetodo[i].i=0;
                else{
                    actualError = Ip - Inew;
                    if (actualError < 0.0001)
                        if (Inew < 0)
                            valoresMetodo[i].i = 0;
                        else
                            valoresMetodo[i].i = Inew;
                }
                Ip = Ip - 0.00001;
            }
        }
        //valoresMetodo=CurvaTeorica(Ig,A,Isat,Rs,Rp,Voc,"green");
        for(i=0; i<valoresMetodo.length;i++){
            valoresMetodo[i].p = valoresMetodo[i].i*valoresMetodo[i].v;
        }
        for(i=0,MAEP=0; i<valoresMetodo.length;i++){
            valoresMetodo[i].error = valoresMetodo[i].p-muestrasF[i].p;
            valoresMetodo[i].error = Math.abs(valoresMetodo[i].error);
            MAEP += valoresMetodo[i].error;
        }
    }
}

```

```

        MAEP=MAEP/valoresMetodo.length;

        //grabamos en la matriz maep los valores actuales.

        matrizMAEP[indiceMaep] = new Object();
        matrizMAEP[indiceMaep].Maep = MAEP;
        matrizMAEP[indiceMaep].a = A;
        matrizMAEP[indiceMaep].Rs = Rs;
        matrizMAEP[indiceMaep].Rp = Rp;
        matrizMAEP[indiceMaep].Ig = Ig;
        matrizMAEP[indiceMaep].Isat = Isat;
        indiceMaep++;
    }
}

// Ordenamos la matriz MAEP
matrizMAEP.sort(function(a,b){return a.Maep - b.Maep});;

valoresFormula.metodo = "S";
valoresFormula.Ig = matrizMAEP[0].Ig;
valoresFormula.A = matrizMAEP[0].a;
valoresFormula.Isat = matrizMAEP[0].Isat;
valoresFormula.Rs = matrizMAEP[0].Rs;
valoresFormula.Rp = matrizMAEP[0].Rp;

actualizaTablaMetodos("S",valoresFormula.Ig,valoresFormula.A,valoresFormula.Isat,valoresFormula.Rs,valoresFormula.Rp);

return(valoresFormula);
}

function MetodoVillalva(Isc,Voc,Imp,Vmp){
var valoresFormula = new Object();

T=302;
K=1.38065e-23; //Constante de Bolzam
Ns=36; //Numero de celdas en serie
q=1.6022e-19;

A=1.3;
Ig=Isc;
Vt=(Ns*A*K*T)/q;
Isat = Ig/(Math.exp(Voc/Vt)-1);
//Proponemos
Rs=0;
Rpmin=(Vmp/(Isc-Imp))-((Voc-Vmp)/Imp);
Rp=Rpmin;

pMaxMuestra=Imp*Vmp; //Potencia maxima de la muestra

//INicio del bucle
errorPmax = 100;
while ((errorPmax > 0.005) && (Rs<50)){
    Ig=Isc;
    //solve eq 15
    exponenteIsat=(Vmp+(Rs*Imp))/Vt;
    RpNumerador =Vmp*(Vmp+Rs*Imp);
    RpDenom=Vmp*(Ig-Isat*(Math.exp(exponenteIsat)-1));
    RpDenom=RpDenom-pMaxMuestra;
    Rp=RpNumerador/RpDenom;

    if(Rp<0){
        Rs=Rs+0.1;
        continue;
    }
    //solve eq 16
    Ig=(Rp+Rs)*Isc/Rp;
    valoresTeoricos=CurvaTeorica(Ig,A,Isat,Rs,Rp,Voc,"black");
    pMaxTeorica=0;
    for (i=0;i<valoresTeoricos.Imetodo.length;i++){
        potencia=valoresTeoricos.Imetodo[i]*valoresTeoricos.Vmetodo[i];
        if (potencia>pMaxTeorica)
            pMaxTeorica=potencia;
    }
    errorPmax=Math.abs(pMaxTeorica-pMaxMuestra);
    Rs=Rs+0.1;
}

actualizaTablaMetodos("V",Ig,A,Isat,Rs,Rp);
valoresFormula.metodo = "V";
valoresFormula.Ig=Ig;
valoresFormula.A = A;
valoresFormula.Isat = Isat;
valoresFormula.Rs = Rs;
valoresFormula.Rp=Rp;
return(valoresFormula);
}

function MetodoXiao(Isc,Voc,Imp,Vmp){
/* Metodo de cuatro parámetros. Desprecia el efecto de la resistencia
 * en paralelo.
 * Se basa en que en un modulo PV la derivada de la potencia respecto
 * al voltaje es cero en el punto de máxima potencia.
 */
var valoresFormula = new Object();

```

```

var Vmetodo = [];
var Imetodo = [];
var VIPmetodo = []; //tension,corriente,potencia v,i,p
var errorMatriz = [];

T=302;
K=1.38065e-23; //Constante de Bolzam
Ns=36; //Numero de celdas en serie
var Ig,Isat,A,Rs,Rp;

Ig=Isc;

for (i = 0; i< muestrasF.length; i++){
    Vmetodo[i] = muestrasF[i].v; //tomamos las tensiones
}
Vmetodo.sort(function(a, b){return a - b}); //ordenamos tensiones

indiceError = 0;
for (A=1.0; A<2; A+=0.05){
    Vt   =(Ns*A*K*T)/q; //Eq 2
    Isat = Ig/(Math.exp(Voc/Vt)-1);
    Rs=(Vt*Math.log((1-Imp/Ig)*Math.exp(Voc/Vt)+Imp/Ig))-Vmp)/Imp;

    index = 0;
    Imetodo[index] = Ig;
    for (j=0; j < Vmetodo.length; j++){
        V  = Vmetodo[j];
        if (index == 0)
            Ip = Ig;
        else
            Ip = Imetodo[index-1];
        while(Imetodo.length < index+1){
            Inew=Ig-Isat*(Math.exp((V+Ip*Rs)/Vt)-1);
            if (Inew > Ip)
                Imetodo[index] = 0;
            else{
                actualError=Ip-Inew;
                if(actualError < 0.0001)
                    if (Inew < 0)
                        Imetodo[index]=0;
                    else
                        Imetodo[index]=Inew;
            }
            Ip = Ip - 0.00001;
        }//busqueda de Imetodo(V)
        index++;
    }
    for (i = 0; i<Vmetodo.length; i++){
        var vip = {};
        vip.v = Vmetodo[i];
        vip.i = Imetodo[i];
        vip.p = Vmetodo[i] * Imetodo[i];

        VIPmetodo[i] = vip;
    }
    //Ordenamos por potencia decreciente el array de VIP
    VIPmetodo.sort(function(a,b){return b.p - a.p});

    exponente = (VIPmetodo[0].v+VIPmetodo[0].i*Rs)/Vt;
    numerador = (Isat/Vt)*Math.exp(exponente);
    denominador = 1+(Isat*Rs/Vt)*Math.exp(exponente);
    t1=numerador/denominador;
    t2=VIPmetodo[0].i/VIPmetodo[0].v
    errorTotal = t2-t1;

    var objError = {};
    objError.error = errorTotal;
    objError.A      = A;

    errorMatriz[indiceError]=objError;
    errorMatriz[indiceError]=objError;
    indiceError++;
} //bucle A
//ordenamos de menor a mayor la matriz de errores
errorMatriz.sort(function(a,b){return a.error - b.error});

Aoptimo = errorMatriz[0].A;

A  = Aoptimo;
Vt = (Ns*Aoptimo*K*T)/q;
Isat = Ig/(Math.exp(Voc/Vt)-1);
Rs  = (Vt*Math.log((1-Imp/Ig)*Math.exp(Voc/Vt)+Imp/Ig))-Vmp)/Imp;

var Rp=Infinity; // Estimado
actualizaTablaMetods("X",Ig,A,Isat,Rs,Rp);
valoresFormula.metodo = "X";
valoresFormula.Ig=Ig;
valoresFormula.A = A;
valoresFormula.Isat = Isat;
valoresFormula.Rs = Rs;
valoresFormula.Rp=Rp;
return(valoresFormula);
//CurvaTeorica(Ig,A,Isat,Rs,Rp,Voc);

```

```

}

function CurvaTeorica(Ig=0.32,A=1.5000,Isat=8.3e-7,Rs=15,Rp=1000000,Voc=20,colorCurva) {
    var valoresCurva = new Object();
    var Vmetodo = []; //array con Tensiones
    var Imetodo = [];

    var T=302; //Temperatura
    var K=1.38065e-23; //Constante de Bolzam
    var Ns=36; //Numero de celdas en serie
    var q=1.6022e-19; // Carga del electron

    Vt=Ns*A*K*T/q;

    for (j=0; j<Voc+1; j++) {
        Vmetodo[j] = j;
    }

    Imetodo[0] = Ig;
    var index=0;
    for (j=0; j<Vmetodo.length; j++) {
        V = Vmetodo[j];
        if (index == 0)
            Ip = Ig;
        else
            Ip = Imetodo[index-1];
        while(Imetodo.length<(index+1)){
            Inew=Ig-Isat*(Math.exp((V+Ip*Rs)/Vt)-1)-(V+Ip*Rs)/Rp;
            if (Inew>Ip)
                Imetodo[index]=Ip;
            else{
                actualError=Ip-Inew;
                if(actualError < 0.0001)
                    if (Inew < 0) Imetodo[index] = 0;
                    else Imetodo[index] = Inew
                Ip = Ip -0.00001;
            }
        }
        index++;
    }
    valoresCurva.Imetodo = Imetodo;
    valoresCurva.Vmetodo = Vmetodo;
    return(valoresCurva);
    //DibujaLineasGrafica(Imetodo,Vmetodo,colorCurva);
}
function DibujaLineasGrafica(iFloat, vFloat,colorCurva,idMetodo){
    //agrupamos elementos
    gIV = document.getElementById("IV");
    var newG = document.createElementNS("http://www.w3.org/2000/svg", "g");
    newG.setAttribute('id', idMetodo);
    //newG.setAttribute("fill", colorCurva);

    var puntosPolilinea="";
    //Pasamos los arrays de corriente y tension a pixels.
    for (var i=0; i<iFloat.length; i++){
        yPixel = iFloat[i]*2000; //(640/0.32)
        yPixel = 670-yPixel;
        xPixel = vFloat[i]*(880.0/22.0);
        xPixel +=60;

        var newCirculo = document.createElementNS(
            "http://www.w3.org/2000/svg", 'circle'); //
        newCirculo.setAttribute("fill",colorCurva);
        newCirculo.setAttribute("cx",xPixel);
        newCirculo.setAttribute("cy",yPixel);
        newCirculo.setAttribute("r",2); //Set path's data
        newG.appendChild(newCirculo);
        puntosPolilinea += xPixel + "," + yPixel+ " ";
    }
    var newPolilinea = document.createElementNS(
        "http://www.w3.org/2000/svg", 'polyline'); //
    newPolilinea.setAttribute("points",puntosPolilinea);

    newPolilinea.setAttribute("fill","none");
    newPolilinea.setAttribute("stroke-width","2");
    newPolilinea.setAttribute("stroke",colorCurva);
    newG.appendChild(newPolilinea);
    gIV.appendChild(newG);

    return (newG);
}

function actualizaTablaMetods(metodo,Ig,A,Isat,Rs,Rp) {
    document.getElementById("ig"+metodo).innerHTML = Ig.toPrecision(2);
    document.getElementById("a"+metodo).innerHTML = A.toPrecision(3);
    document.getElementById("is"+metodo).innerHTML = Isat.toExponential(2);
    document.getElementById("rs"+metodo).innerHTML = Rs.toPrecision(3);
    document.getElementById("rp"+metodo).innerHTML = Rp.toPrecision(3);
}
function VerOcultarLineaMetodo (nombreMetodo){
    var checkBox = document.getElementById(nombreMetodo);

```

```
var lMetodo;
switch (nombreMetodo) {
    case "verRw": lMetodo = lMetodoRW; break;
    case "verX": lMetodo = lMetodoX; break;
    case "verS": lMetodo = lMetodoS; break;
    case "verV": lMetodo = lMetodoV; break;
}
if (checkBox.checked == true)
    lMetodo.style.visibility = "visible";
else
    lMetodo.style.visibility = "hidden";
//lMetodoRW.style.visibility = "hidden"
}
```

```

/*TFM Elena Álvarez Castro
Página web para la visualización de datos
Fichero de Estilo CSS*/

/* W3.CSS 4.13 June 2019 by Jan Egil and Borge Refsnes */
html{box-sizing:border-box},*:before,:after(box-sizing:inherit)
/* Extract from normalize.css by Nicolas Gallagher and Jonathan Neal git.io/normalize */
html{-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}body{margin:0}
article,aside,details,figcaption,figure,footer,header,main,menu,nav,section{display:block}summary{display:list-item}
audio,canvas,progress,video{display:inline-block}progress{vertical-align:baseline}
audio:not([controls)]{display:none;height:0}[hidden],template{display:none}
a{background-color:transparent}a:active,a:hover{outline-width:0}
abbr[title]{border-bottom:none;text-decoration:underline;text-decoration:underline dotted}
b,strong{font-weight:bolder}dfn{font-style:italic}mark{background:#ff0;color:#000}
small{font-size:80%}sub,sup{font-size:75%;line-height:0;position:relative;vertical-align:baseline}
sub{bottom:-0.25em}sup{top:-0.5em}figure{margin:1em 40px}img{border-style:none}
code,kbd,pre,samp{font-family:monospace,monospace;font-size:1em}hr{box-sizing:content-box;height:0;overflow:visible}
button,input,select,textarea,optgroup{font:inherit;margin:0}optgroup{font-weight:bold}
button,input{overflow:visible}button,select{text-transform:none}
button,[type=button],[type=reset],[type=submit]{-webkit-appearance:button}
button::-moz-focus-inner,[type=button]:-moz-focus-inner,[type=reset]:-moz-focus-inner,[type=submit]:-moz-focus-inner{border-style:none;padding:0}
button:-moz-focusring,[type=button]:-moz-focusring,[type=reset]:-moz-focusring,[type=submit]:-moz-focusring{outline:1px dotted ButtonText}
fieldset{border:1px solid #c0c0c0;margin:0 2px;padding:.35em .625em .75em}
legend{color:inherit;display:table;max-width:100%;padding:0;white-space:normal}textarea{overflow:auto}
[type=checkbox],[type=radio]{padding:0}
[type=number]:-webkit-inner-spin-button,[type=number]:-webkit-outer-spin-button{height:auto}
[type=search]{-webkit-appearance:textfield;outline-offset:-2px}
[type=search]:-webkit-search-decoration{-webkit-appearance:none}
::-webkit-file-upload-button{-webkit-appearance:button;font:inherit}
/* End extract */
html,body{font-family:Verdana,sans-serif;font-size:15px;line-height:1.5}html{overflow-x:hidden}
h1{font-size:36px}h2{font-size:30px}h3{font-size:24px}h4{font-size:20px}h5{font-size:18px}h6{font-size:16px}.w3-serif{font-family:serif}
h1,h2,h3,h4,h5,h6{font-family:"Segoe UI",Arial,sans-serif;font-weight:400;margin:10px 0}.w3-wide{letter-spacing:4px}
hr{border:0;border-top:1px solid #eee;margin:20px 0}
.w3-image{max-width:100%;height:auto}img{vertical-align:middle}a{color:inherit}
.w3-table,.w3-table-all{border-collapse:collapse;border-spacing:0;width:100%;display:table}.w3-table-all{border:1px solid #ccc}
.w3-bordered tr,.w3-table-all tr{border-bottom:1px solid #ddd}.w3-striped tbody tr:nth-child(even){background-color:#f1f1f1}
.w3-table-all tr:nth-child(odd){background-color:#fff}.w3-table-all tr:nth-child(even){background-color:#f1f1f1}
.w3-hoverable tbody tr:hover,.w3-ul.w3-hoverable li:hover{background-color:#ccc}.w3-centered tr th,.w3-centered tr td{text-align:center}
.w3-table td,.w3-table th,.w3-table-all td,.w3-table-all th{padding:8px 8px;display:table-cell;text-align:left;vertical-align:top}
.w3-table th:first-child,.w3-table td:first-child,.w3-table-all th:first-child,.w3-table-all td:first-child{padding-left:16px}
.w3-btn,.w3-button{border:none;display:inline-block;padding:8px 16px;vertical-align:middle;overflow:hidden;text-decoration:none;color:inherit;background-color:inherit;text-align:center;cursor:pointer;white-space:nowrap}
.w3-btn:hover{box-shadow:0 8px 16px 0 rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19)}
.w3-btn,.w3-button{-webkit-touch-callout:none;-webkit-user-select:none;-khtml-user-select:none;-moz-user-select:none;-ms-user-select:none;user-select:none}
.w3-disabled,.w3-btn:disabled,.w3-button:disabled{cursor:not-allowed;opacity:0.3}.w3-disabled *,:disabled *{pointer-events:none}
.w3-btn.w3-disabled:hover,.w3-btn:disabled:hover{box-shadow:none}
.w3-badge,.w3-tag{background-color:#000;color:#fff;display:inline-block;padding-left:8px;padding-right:8px;text-align:center}.w3-badge{border-radius:50%}
.w3-ul{list-style-type:none;padding:0;margin:0}.w3-ul li{padding:8px 16px;border-bottom:1px solid #ddd}.w3-ul li:last-child{border-bottom:none}
.w3-tooltip,.w3-display-container{position:relative}.w3-tooltip .w3-text{display:none}.w3-tooltip:hover .w3-text{display:inline-block}
.w3-ripple{active{opacity:0.5}.w3-ripple{transition:opacity 0s}}
.w3-input{padding:8px;display:block;border:none;border-bottom:1px solid #ccc;width:100%}
.w3-select{padding:9px 0;width:100%;border:none;border-bottom:1px solid #ccc}
.w3-dropdown-click,.w3-dropdown-hover{position:relative;display:inline-block;cursor:pointer}
.w3-dropdown-hover:hover .w3-dropdown-content{display:block}
.w3-dropdown-hover:first-child,.w3-dropdown-click:hover{background-color:#ccc;color:#000}
.w3-dropdown-hover: hover > .w3-button:first-child,.w3-dropdown-click: hover > .w3-button:first-child{background-color:#ccc;color:#000}
.w3-dropdown-content{cursor:auto;color:#000;background-color:#fff;display:none;position:absolute;min-width:160px;margin:0;padding:0;z-index:1}
.w3-check,.w3-radio{width:24px;height:24px;position:relative;top:6px}
.w3-sidebar{height:100%;width:200px;background-color:#fff;position:fixed;z-index:1;overflow:auto}
.w3-bar-block.w3-dropdown-hover,.w3-bar-block.w3-dropdown-click{width:100%}
.w3-bar-block.w3-dropdown-hover .w3-dropdown-content,.w3-bar-block.w3-dropdown-click .w3-dropdown-content{min-width:100%}
.w3-bar-block.w3-dropdown-hover .w3-button,.w3-bar-block.w3-dropdown-click .w3-button{width:100%;text-align:left;padding:8px 16px}
.w3-main,#main{transition:margin-left .4s}
.w3-modal{z-index:3;display:none;padding-top:100px;position:fixed;left:0;top:0;width:100%;height:100%;overflow:auto;background-color:rgb(0,0,0);background-color:rgba(0,0,0,0.4)}
.w3-modal-content{margin:auto;background-color:#fff;position:relative;padding:0;outline:0;width:600px}
.w3-bar{width:100%;overflow:hidden}.w3-center .w3-bar{display:inline-block;width:auto}
.w3-bar.w3-bar-item{padding:8px 16px;float:left;width:auto;border:none;display:block;outline:0}
.w3-bar.w3-dropdown-hover,.w3-bar.w3-dropdown-click{position:static;float:left}
.w3-bar.w3-button{white-space:normal}
.w3-bar-block.w3-bar-item{width:100%;display:block;padding:8px 16px;text-align:left;border:none;white-space:normal;float:none;outline:0}
.w3-bar-block.w3-center .w3-bar-item{text-align:center}.w3-block{display:block;width:100%}

```

```

.w3-responsive{display:block;overflow-x:auto}
.w3-container:after,.w3-container:before,.w3-panel:after,.w3-panel:before,.w3-row:after,.w3-row:before,.w3-row-
padding:after,.w3-row-padding:before,
.w3-cell-row:before,.w3-cell-row:after,.w3-clear:after,.w3-clear:before,.w3-bar:before,.w3-
bar:after{content:"";display:table;clear:both}
.w3-col,.w3-half,.w3-third,.w3-twothird,.w3-threequarter,.w3-quarter{float:left;width:100%}
.w3-col.s1{width:8.33333%}.w3-col.s2{width:16.66666%}.w3-col.s3{width:24.99999%}.w3-col.s4{width:33.33333%}
.w3-col.s5{width:41.66666%}.w3-col.s6{width:49.99999%}.w3-col.s7{width:58.33333%}.w3-col.s8{width:66.66666%}
.w3-col.s9{width:74.99999%}.w3-col.s10{width:83.33333%}.w3-col.s11{width:91.66666%}.w3-col.s12{width:99.99999%}
@media (min-width:601px){.w3-col.m1{width:8.33333%}.w3-col.m2{width:16.66666%}.w3-col.m3,.w3-quarter{width:24.99999%}.w3-
col.m4,.w3-third{width:33.33333%}
.w3-col.m5{width:41.66666%}.w3-col.m6,.w3-half{width:49.99999%}.w3-col.m7{width:58.33333%}.w3-col.m8,.w3-
twothird{width:66.66666%}
.w3-col.m9,.w3-threequarter{width:74.99999%}.w3-col.m10{width:83.33333%}.w3-col.m11{width:91.66666%}.w3-
col.m12{width:99.99999%}
@media (min-width:993px){.w3-col.l1{width:8.33333%}.w3-col.l2{width:16.66666%}.w3-col.l3{width:24.99999%}.w3-
col.l4{width:33.33333%}
.w3-col.l5{width:41.66666%}.w3-col.l6{width:49.99999%}.w3-col.l7{width:58.33333%}.w3-col.l8{width:66.66666%}
.w3-col.l9{width:74.99999%}.w3-col.l10{width:83.33333%}.w3-col.l11{width:91.66666%}.w3-col.l12{width:99.99999%}}
.w3-rest{overflow:hidden}.w3-stretch{margin-left:-16px;margin-right:-16px}
.w3-content,.w3-auto{margin-left:auto;margin-right:auto}.w3-content{max-width:980px}.w3-auto{max-width:1140px}
.w3-cell-row{display:table;width:100%}.w3-cell{display:table-cell}
.w3-cell-top{vertical-align:top}.w3-cell-middle{vertical-align:middle}.w3-cell-bottom{vertical-align:bottom}
.w3-hide{display:none!important}.w3-show-block,.w3-show{display:block!important}.w3-show-inline-block{display:inline-
block!important}
@media (max-width:1205px){.w3-auto{max-width:95%}}
@media (max-width:600px){.w3-modal-content{margin:0 10px;width:auto!important}.w3-modal{padding-top:30px}
.w3-dropdown-hover.w3-mobile .w3-dropdown-content,.w3-dropdown-click.w3-mobile .w3-dropdown-content{position:relative}
.w3-hide-small{display:none!important}.w3-mobile{display:block;width:100%!important}.w3-bar-item.w3-mobile,.w3-dropdown-
hover.w3-mobile,.w3-dropdown-click.w3-mobile{text-align:center}
.w3-dropdown-hover.w3-mobile,.w3-dropdown-hover.w3-mobile .w3-btn,.w3-dropdown-hover.w3-mobile .w3-button,.w3-dropdown-
click.w3-mobile,.w3-dropdown-click.w3-mobile .w3-btn,.w3-dropdown-click.w3-mobile .w3-button{width:100%}
@media (max-width:768px){.w3-modal-content{width:500px}.w3-modal{padding-top:50px}}
@media (min-width:993px){.w3-modal-content{width:900px}.w3-hide-large{display:none!important}.w3-sidebar.w3-
collapse{display:block!important}
@media (max-width:992px) and (min-width:601px){.w3-hide-medium{display:none!important}}
@media (max-width:992px){.w3-sidebar.w3-collapse{display:none}.w3-main{margin-left:0!important;margin-
right:0!important}.w3-auto{max-width:100%}
.w3-top,.w3-bottom{position:fixed;display:none;width:100%;height:100%;top:0;left:0;right:0;bottom:0;background-
color:rgba(0,0,0,0.5);z-index:2}
.w3-display-topleft{position:absolute;left:0;top:0}.w3-display-topleft{position:absolute;right:0;top:0}
.w3-display-bottomleft{position:absolute;left:0;bottom:0}.w3-display-bottomright{position:absolute;right:0;bottom:0}
.w3-display-middle{position:absolute;top:50%;left:50%;transform:translate(-50%,-50%);-ms-transform:translate(-50%,-50%)}
.w3-display-left{position:absolute;top:50%;left:0%;transform:translate(0%,-50%);-ms-transform:translate(0%,-50%)}
.w3-display-right{position:absolute;top:50%;right:0%;transform:translate(0%,50%);-ms-transform:translate(0%,50%)}
.w3-display-topmiddle{position:absolute;left:50%;top:0;transform:translate(-50%,0%);-ms-transform:translate(-50%,0%)}
.w3-display-bottommiddle{position:absolute;left:50%;bottom:0;transform:translate(-50%,50%);-ms-transform:translate(-50%
%,50%)}
.w3-display-container:hover .w3-display-hover{display:block}.w3-display-container:hover span.w3-display-
hover{display:inline-block}.w3-display-hover{display:none}
.w3-display-position{position:absolute}
.w3-circle{border-radius:50%}
.w3-round-small{border-radius:2px}.w3-round,.w3-round-medium{border-radius:4px}.w3-round-large{border-radius:8px}.w3-
round-xlarge{border-radius:16px}.w3-round-xxlarge{border-radius:32px}
.w3-row-padding,.w3-row-padding>.w3-half,.w3-row-padding>.w3-third,.w3-row-padding>.w3-twothird,.w3-row-padding>.w3-
threequarter,.w3-row-padding>.w3-quarter,.w3-row-padding>.w3-col{padding:0 8px}
.w3-container,.w3-panel{padding:0.01em 16px}.w3-panel{margin-top:16px;margin-bottom:16px}
.w3-code,.w3-codespan{font-family:Consolas,"courier new";font-size:16px}
.w3-code{width:auto;background-color:#fff;padding:8px 12px;border-left:4px solid #4CAF50;word-wrap:break-word}
.w3-codespan{color:crimson;background-color:#f1f1f1;padding-left:4px;padding-right:4px;font-size:110%}
.w3-card,.w3-card-2{box-shadow:0 2px 5px 0 rgba(0,0,0,0.16),0 2px 10px 0 rgba(0,0,0,0.12)}
.w3-card-4,.w3-hover-shadow:hover{box-shadow:0 4px 10px 0 rgba(0,0,0,0.2),0 4px 20px 0 rgba(0,0,0,0.19)}
.w3-spin{animation:w3-spin 2s infinite linear}@keyframes w3-spin{0%{transform:rotate(0deg)}100%
{transform:rotate(359deg)}}
.w3-animate-fading{animation:fading 10s infinite}@keyframes fading{0%{opacity:0}50%{opacity:1}100%{opacity:0}}
.w3-animate-opacity{animation:opac 0.8s}@keyframes opac{from{opacity:0} to{opacity:1}}
.w3-animate-top{position:relative;animation:animatetop 0.4s}@keyframes animatetop{from{top:-300px;opacity:0}
to{top:0;opacity:1}}
.w3-animate-left{position:relative;animation:animateleft 0.4s}@keyframes animateleft{from{left:-300px;opacity:0}
to{left:0;opacity:1}}
.w3-animate-right{position:relative;animation:animatoright 0.4s}@keyframes animatoright{from{right:-300px;opacity:0}
to{right:0;opacity:1}}
.w3-animate-bottom{position:relative;animation:animatebottom 0.4s}@keyframes animatebottom{from{bottom:-300px;opacity:0}
to{bottom:0;opacity:1}}
.w3-animate-zoom {animation:animatezoom 0.6s}@keyframes animatezoom{from{transform:scale(0)} to{transform:scale(1)}}
.w3-animate-input{transition:width 0.4s ease-in-out}.w3-animate-input:focus{width:100%!important}
.w3-opacity,.w3-hover-opacity:hover{opacity:0.60}.w3-opacity-off,.w3-hover-opacity-off:hover{opacity:1}
.w3-opacity-max{opacity:0.25}.w3-opacity-min{opacity:0.75}
.w3-greyscale-max,.w3-greyscale-max,.w3-hover-greyscale:hover,.w3-hover-greyscale:Hover{filter:grayscale(100%)}
.w3-greyscale,.w3-grayscale{filter:grayscale(75%)}.w3-greyscale-min,.w3-grayscale-min{filter:grayscale(50%)}
.w3-sepia{filter:sepia(75%)}.w3-sepia-max,.w3-hover-sepia:hover{filter:sepia(100%)}.w3-sepia-min{filter:sepia(50%)}
.w3-tiny{font-size:10px!important}.w3-small{font-size:12px!important}.w3-medium{font-size:15px!important}.w3-large{font-
size:18px!important}
.w3-xlarge{font-size:24px!important}.w3-xxlarge{font-size:36px!important}.w3-xxxlarge{font-size:48px!important}.w3-
jumbo{font-size:64px!important}
.w3-left-align{text-align:left!important}.w3-right-align{text-align:right!important}.w3-justify{text-
align:justify!important}.w3-center{text-align:center!important}
.w3-border-0{border:0!important}.w3-border{border:1px solid #ccc!important}
.w3-border-top{border-top:1px solid #ccc!important}.w3-border-bottom{border-bottom:1px solid #ccc!important}
.w3-border-left{border-left:1px solid #ccc!important}.w3-border-right{border-right:1px solid #ccc!important}
.w3-topbar{border-top:6px solid #ccc!important}.w3-bottombar{border-bottom:6px solid #ccc!important}
.w3-leftbar{border-left:6px solid #ccc!important}.w3-rightbar{border-right:6px solid #ccc!important}

```

```

.w3-section, .w3-code{margin-top:16px!important; margin-bottom:16px!important}
.w3-margin{margin:16px!important}.w3-margin-top{margin-top:16px!important}.w3-margin-bottom{margin-bottom:16px!important}
.w3-margin-left{margin-left:16px!important}.w3-margin-right{margin-right:16px!important}
.w3-padding-small{padding:4px 8px!important}.w3-padding{padding:8px 16px!important}.w3-padding-large{padding:12px
24px!important}
.w3-padding-16{padding-top:16px!important; padding-bottom:16px!important}.w3-padding-24{padding-
top:24px!important; padding-bottom:24px!important}
.w3-padding-32{padding-top:32px!important; padding-bottom:32px!important}.w3-padding-48{padding-
top:48px!important; padding-bottom:48px!important}
.w3-padding-64{padding-top:64px!important; padding-bottom:64px!important}
.w3-left{float:left!important}.w3-right{float:right!important}
.w3-button:hover{color:#000!important; background-color:#ccc!important}
.w3-transparent, .w3-hover-none:hover{background-color:transparent!important}
.w3-hover-none:hover{box-shadow:none!important}
/* Colors */
.w3-amber, .w3-hover-amber:hover{color:#000!important; background-color:#ffc107!important}
.w3-aqua, .w3-hover-aqua:hover{color:#000!important; background-color:#00ffff!important}
.w3-blue, .w3-hover-blue:hover{color:#fff!important; background-color:#2196F3!important}
.w3-light-blue, .w3-hover-light-blue:hover{color:#000!important; background-color:#87CEEB!important}
.w3-brown, .w3-hover-brown:hover{color:#fff!important; background-color:#795548!important}
.w3-cyan, .w3-hover-cyan:hover{color:#000!important; background-color:#00bcd4!important}
.w3-blue-grey, .w3-hover-blue-grey:hover, .w3-blue-gray, .w3-hover-blue-gray:hover{color:#fff!important; background-
color:#607d8b!important}
.w3-green, .w3-hover-green:hover{color:#fff!important; background-color:#4CAF50!important}
.w3-light-green, .w3-hover-light-green:hover{color:#000!important; background-color:#8bc34a!important}
.w3-indigo, .w3-hover-indigo:hover{color:#fff!important; background-color:#3f51b5!important}
.w3-khaki, .w3-hover-khaki:hover{color:#000!important; background-color:#f0e68c!important}
.w3-lime, .w3-hover-lime:hover{color:#000!important; background-color:#cdcc39!important}
.w3-orange, .w3-hover-orange:hover{color:#000!important; background-color:#ff9800!important}
.w3-deep-orange, .w3-hover-deep-orange:hover{color:#fff!important; background-color:#ff5722!important}
.w3-pink, .w3-hover-pink:hover{color:#fff!important; background-color:#e91e63!important}
.w3-purple, .w3-hover-purple:hover{color:#fff!important; background-color:#9c27b0!important}
.w3-deep-purple, .w3-hover-deep-purple:hover{color:#fff!important; background-color:#673ab7!important}
.w3-red, .w3-hover-red:hover{color:#fff!important; background-color:#f44336!important}
.w3-sand, .w3-hover-sand:hover{color:#000!important; background-color:#fdf5e6!important}
.w3-teal, .w3-hover-teal:hover{color:#fff!important; background-color:#009688!important}
.w3-yellow, .w3-hover-yellow:hover{color:#000!important; background-color:#ffeb3b!important}
.w3-white, .w3-hover-white:hover{color:#000!important; background-color:#fff!important}
.w3-black, .w3-hover-black:hover{color:#fff!important; background-color:#000!important}
.w3-grey, .w3-hover-grey:hover, .w3-gray, .w3-hover-gray:hover{color:#000!important; background-color:#9e9e9e!important}
.w3-light-grey, .w3-hover-light-grey:hover, .w3-light-gray, .w3-hover-light-gray:hover{color:#000!important; background-
color:#f1f1f1!important}
.w3-dark-grey, .w3-hover-dark-grey:hover, .w3-dark-gray, .w3-hover-dark-gray:hover{color:#fff!important; background-
color:#616161!important}
.w3-pale-red, .w3-hover-pale-red:hover{color:#000!important; background-color:#ffddcc!important}
.w3-pale-green, .w3-hover-pale-green:hover{color:#000!important; background-color:#ddffdd!important}
.w3-pale-yellow, .w3-hover-pale-yellow:hover{color:#000!important; background-color:#ffffcc!important}
.w3-pale-blue, .w3-hover-pale-blue:hover{color:#000!important; background-color:#ddffff!important}
.w3-text-amber, .w3-hover-text-amber:hover{color:#ffc107!important}
.w3-text-aqua, .w3-hover-text-aqua:hover{color:#00ffff!important}
.w3-text-blue, .w3-hover-text-blue:hover{color:#2196F3!important}
.w3-text-light-blue, .w3-hover-text-light-blue:hover{color:#87CEEB!important}
.w3-text-brown, .w3-hover-text-brown:hover{color:#795548!important}
.w3-text-cyan, .w3-hover-text-cyan:hover{color:#00bcd4!important}
.w3-text-blue-grey, .w3-hover-text-blue-grey:hover, .w3-text-blue-gray, .w3-hover-text-blue-
gray:hover{color:#607d8b!important}
.w3-text-green, .w3-hover-text-green:hover{color:#4CAF50!important}
.w3-text-light-green, .w3-hover-text-light-green:hover{color:#8bc34a!important}
.w3-text-indigo, .w3-hover-text-indigo:hover{color:#3f51b5!important}
.w3-text-khaki, .w3-hover-text-khaki:hover{color:#b4aa50!important}
.w3-text-lime, .w3-hover-text-lime:hover{color:#cdcc39!important}
.w3-text-orange, .w3-hover-text-orange:hover{color:#ff9800!important}
.w3-text-deep-orange, .w3-hover-text-deep-orange:hover{color:#ff5722!important}
.w3-text-pink, .w3-hover-text-pink:hover{color:#e91e63!important}
.w3-text-purple, .w3-hover-text-purple:hover{color:#9c27b0!important}
.w3-text-deep-purple, .w3-hover-text-deep-purple:hover{color:#673ab7!important}
.w3-text-red, .w3-hover-text-red:hover{color:#f44336!important}
.w3-text-sand, .w3-hover-text-sand:hover{color:#fdf5e6!important}
.w3-text-teal, .w3-hover-text-teal:hover{color:#009688!important}
.w3-text-yellow, .w3-hover-text-yellow:hover{color:#d2be8c!important}
.w3-text-white, .w3-hover-text-white:hover{color:#fff!important}
.w3-text-black, .w3-hover-text-black:hover{color:#000!important}
.w3-text-grey, .w3-hover-text-grey:hover, .w3-text-gray, .w3-hover-text-gray:hover{color:#757575!important}
.w3-text-light-grey, .w3-hover-text-light-grey:hover, .w3-text-light-gray, .w3-hover-text-light-
gray:hover{color:#f1f1f1!important}
.w3-text-dark-grey, .w3-hover-text-dark-grey:hover, .w3-text-dark-gray, .w3-hover-text-dark-
gray:hover{color:#3a3a3a!important}
.w3-border-amber, .w3-hover-border-amber:hover{border-color:#ffc107!important}
.w3-border-aqua, .w3-hover-border-aqua:hover{border-color:#00ffff!important}
.w3-border-blue, .w3-hover-border-blue:hover{border-color:#2196F3!important}
.w3-border-light-blue, .w3-hover-border-light-blue:hover{border-color:#87CEEB!important}
.w3-border-brown, .w3-hover-border-brown:hover{border-color:#795548!important}
.w3-border-cyan, .w3-hover-border-cyan:hover{border-color:#00bcd4!important}
.w3-border-blue-grey, .w3-hover-border-blue-grey:hover, .w3-border-blue-gray, .w3-hover-border-blue-gray:hover{border-
color:#607d8b!important}
.w3-border-green, .w3-hover-border-green:hover{border-color:#4CAF50!important}
.w3-border-light-green, .w3-hover-border-light-green:hover{border-color:#8bc34a!important}
.w3-border-indigo, .w3-hover-border-indigo:hover{border-color:#3f51b5!important}
.w3-border-khaki, .w3-hover-border-khaki:hover{border-color:#f0e68c!important}
.w3-border-lime, .w3-hover-border-lime:hover{border-color:#cdcc39!important}
.w3-border-orange, .w3-hover-border-orange:hover{border-color:#ff9800!important}
.w3-border-deep-orange, .w3-hover-border-deep-orange:hover{border-color:#ff5722!important}
.w3-border-pink, .w3-hover-border-pink:hover{border-color:#e91e63!important}

```

```
.w3-border-purple,.w3-hover-border-purple:hover{border-color:#9c27b0!important}
.w3-border-deep-purple,.w3-hover-border-deep-purple:hover{border-color:#673ab7!important}
.w3-border-red,.w3-hover-border-red:hover{border-color:#f44336!important}
.w3-border-sand,.w3-hover-border-sand:hover{border-color:#fdf5e6!important}
.w3-border-teal,.w3-hover-border-teal:hover{border-color:#009688!important}
.w3-border-yellow,.w3-hover-border-yellow:hover{border-color:#ffeb3b!important}
.w3-border-white,.w3-hover-border-white:hover{border-color:#ffff!important}
.w3-border-black,.w3-hover-border-black:hover{border-color:#000!important}
.w3-border-grey,.w3-hover-border-grey:hover,.w3-border-gray,.w3-hover-border-gray:hover{border-color:#9e9e9e!important}
.w3-border-light-grey,.w3-hover-border-light-grey:hover,.w3-border-light-gray,.w3-hover-border-light-gray:hover{border-color:#f1f1f1!important}
.w3-border-dark-grey,.w3-hover-border-dark-grey:hover,.w3-border-dark-gray,.w3-hover-border-dark-gray:hover{border-color:#616161!important}
.w3-border-pale-red,.w3-hover-border-pale-red:hover{border-color:#ffe7e7!important}.w3-border-pale-green,.w3-hover-border-pale-green:hover{border-color:#e7ffe7!important}
.w3-border-pale-yellow,.w3-hover-border-pale-yellow:hover{border-color:#ffffcc!important}.w3-border-pale-blue,.w3-hover-border-pale-blue:hover{border-color:#e7ffff!important}
```


B. Archivo de Código en C. Programa microcontrolador.

La placa de desarrollo utilizada es Arduino MKR1000. Incluye conectividad Wi-Fi y se basa en el Atmel ATSAMW25 SoC (System on Chip), que forma parte de la familia SmartConnect de dispositivos Atmel Wireless, diseñados específicamente para proyectos y dispositivos de IoT. Más información en la página oficial de Arduino (enlace).

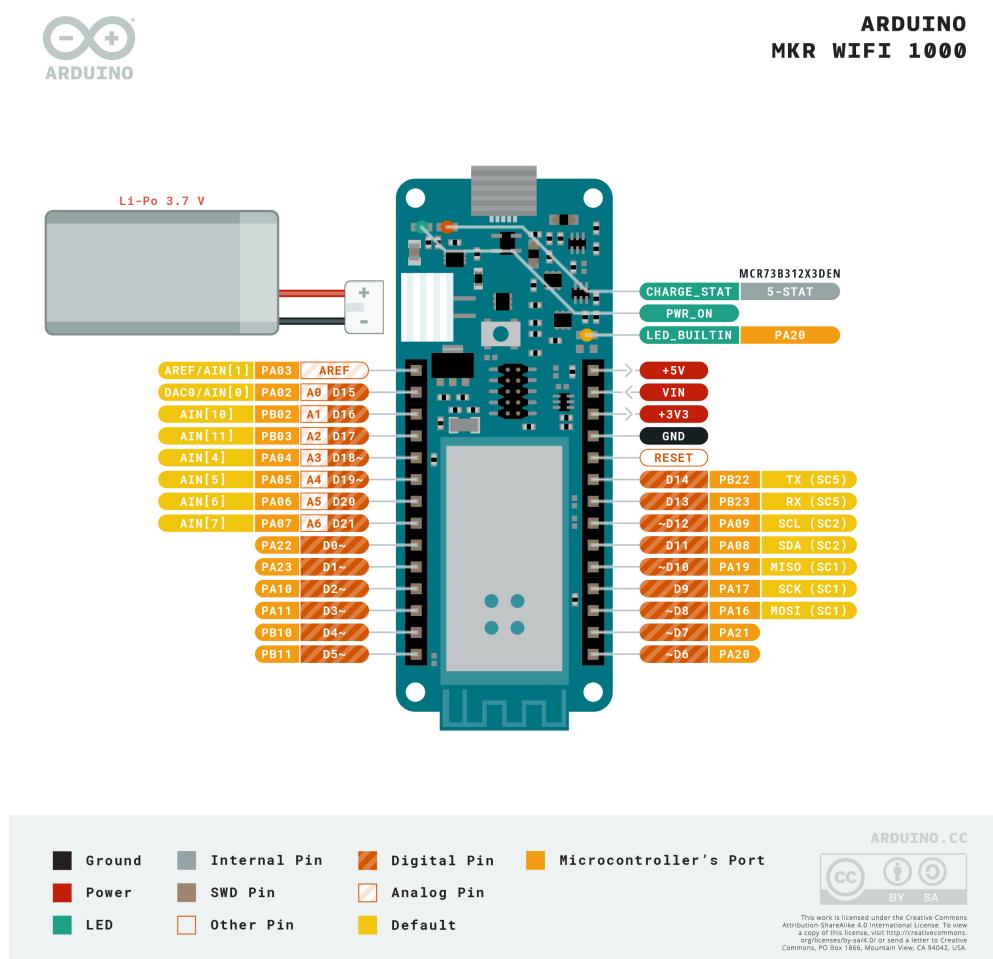


Figure B.1.: Arduino MKR1000. Pinout

```

/*
 * TFM Elena Álvarez Castro
 * Código microcontrolador Arduino mkr1000
 */

#include <SPI.h>
#include <WiFi101.h>
#include <SD.h>
#include <avr/dtostrf.h>
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

#define MOSFET 2
#define RELE 3

#define PAGINA_INICIAL 0
#define NUEVA_TRAZA 1
#define FAVICON 2
#define TEXTOSD 3
#define NUEVO_FICHERO 4
#define NUEVA_TRAZA_FLOAT 5
#define NUEVO_NIVEL_LUZ 6
#define MAXIMA_LONGITUD_NOMBRE_FICHERO 16
int peticion;
char ficheroSolicitado[MAXIMA_LONGITUD_NOMBRE_FICHERO]; // 8.3+null

//mi wifi
char ssid[] = "Andared"; // your network SSID (name)
char pass[] = "llevalatararaunvestidoblancoenodecascabeles";

int keyIndex = 0; // your network key Index number (needed only for WEP)

int led = LED_BUILTIN;
int status = WL_IDLE_STATUS;
WiFiServer server(80);

void CortocircuitaPlacaSolar(boolean);
void EnviaDatosCliente(WiFiClient);
void EnviaDatosNuevaTraza(WiFiClient);
void EnviaFicheroSD(WiFiClient);
int TomaMuestras(void);
void EnviaHtml(char *);
void AnalizaActualPeticion(String);
void ConversorDA();

#define NUMERO_MUESTRAS 512
unsigned int arrCorriente[NUMERO_MUESTRAS];
unsigned int arrTension [NUMERO_MUESTRAS];
char cadenaF[16] = " ";
float corrienteF;
float tensionF;
char miBuffer[1048];

//Sensor de Temperatura
// Tabla generada
#define NUMTEMPS 20
short temptable[NUMTEMPS][2] = {
    {1, 713},
    {54, 236},
    {107, 195},
    {160, 172},
    {213, 157},
    {266, 144},
    {319, 134},
    {372, 125},
    {425, 117},
    {478, 110},
    {531, 103},
    {584, 96},
    {637, 89},
    {690, 83},
    {743, 75},
    {796, 68},
    {849, 59},
    {902, 48},
    {955, 34},
    {1008, 3}
};

int THERMISTOR_PIN = A3;

const char nombre[] = "hola\
gIV = document.getElementById(\"IV\");\
{hola}";

File ficheroSd;
void setup() {
  Serial.begin(9600);

```

```

delay(1000); //Para poder continuar aunque no haya puerto serie.
Serial.println("Access Point Web Server");

pinMode(led, OUTPUT);
pinMode(RELE, OUTPUT);
pinMode(MOSFET, OUTPUT);

digitalWrite(RELE, LOW);
digitalWrite(MOSFET, HIGH);

pinMode(A1, INPUT);
pinMode(A2, INPUT);

if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
}

Serial.print("Creating access point named: ");
Serial.println(ssid);
/* Codigo para modo punto de acceso
// Create open network. Change this line if you want to create an WEP network:
status = WiFi.beginP(ssid);
if (status != WL_AP_LISTENING) {
    Serial.println("Creating access point failed");
    // don't continue
    while (true);
}

// wait 10 seconds for connection:
delay(10000);
*/
while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);

    // wait 10 seconds for connection:
    delay(10000);
}
// start the web server on port 80
server.begin();

// you're connected now, so print out the status
printWiFiStatus();

Serial.print("Initializing SD card...");

if (!SD.begin(4)) {
    Serial.println("initialization SD failed!");
}
else
    Serial.println("initialization SD done.");

dac.begin(0x62);
ConversorDA(2000);
}

void loop() {
if (status != WiFi.status()) {
    // it has changed update the variable
    status = WiFi.status();

    if (status == WL_CONNECTED) {
        byte remoteMac[6];

        Serial.print("Device connected to AP, MAC address: ");
        WiFi.APClientMacAddress(remoteMac);
        printMacAddress(remoteMac);
    } else {

        Serial.println("Device disconnected from AP");
        status = WiFi.begin(ssid, pass);
        Serial.println("Conectando de nuevo ...");
        delay(1000);
        server.begin();
    }
}
}

WiFiClient client = server.available(); // listen for incoming clients

if (client) { // if you get a client,
    Serial.println("new client"); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    //int peticion = PAGINA_INICIAL; // solicitud del cliente
    int escritos = 0;
    char ficheroSolicitado[MAXIMA_LONGITUD_NOMBRE_FICHERO]; // 8.3+null
    unsigned int longitudCadenaGet;
    int i,j;

```

```

while (client.connected()) {
    if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n') {
            // if the current line is blank, you got two newline characters in a row.
            // that's the end of the client HTTP request, so send a response:
            if (currentLine.length() == 0) {
                // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
                // and a content-type so the client knows what's coming, then a blank line:
                client.println("HTTP/1.1 200 OK");
                //client.println("Connection: close"); // the connection will be closed after completion of the re
                //client.println("Content-type:text/html");
                //client.println("Connection: close"); // the connection will be closed after completion of the re
                //client.println();
                switch (peticion){
                    case NUEVO_FICHERO:
                        EnviaFicheroSD(client);
                    case PAGINA_INICIAL:
                        /*
                        Serial.println("Pide Pagina");
                        EnviaHtml(client, codigoCabecera);
                        EnviaHtml(client, codigoJavaScript);
                        EnviaHtml(client, codigoBody);
                        EnviaDatosCliente(client);
                        EnviaHtml(client, codigoFinBody);
                        client.println();
                        */
                        break;
                    case NUEVA_TRAZA:
                        EnviaDatosNuevaTraza(client);
                        //EnviaHtml(client, cabeceraNuevaTraza);
                        client.println();
                        break;
                    case NUEVA_TRAZA_FLOAT:
                        EnviaDatosNuevaTrazaFloat(client);
                    case FAVICON:
                        break;
                    case TEXTOSD:
                        EnviaFicheroSD(client);
                        break;
                    case NUEVO_NIVEL_LUZ:
                        client.println();
                        break;
                }
                // break out of the while loop:
                break;
            }
            else { // if you got a newline, then clear currentLine:
                AnalizaActualPeticion(currentLine);
                currentLine = "";
            }
        }
        else if (c != '\r') { // if you got anything else but a carriage return character,
            currentLine += c; // add it to the end of the currentLine
        }
    }

    // Check to see if the client request was "GET /H" or "GET /L":
    if (currentLine.endsWith("GET /H")) {
        digitalWrite(led, HIGH); // GET /H turns the LED on
    }
    if (currentLine.endsWith("GET /L")) {
        digitalWrite(led, LOW); // GET /L turns the LED off
    }
    if (currentLine.endsWith("GET /nDatos.txt")) {
        peticion = NUEVA_TRAZA;
    }
}

}
// close the connection:
client.stop();
Serial.println("client disconnected");
}

//ConversorDA(2500);
/*
Serial.print("Temperatura:");
int celsius = read_temp();
Serial.println(celsius);
*/
}

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print your WiFi shield's IP address:
    IPAddress ip = WiFi.localIP();
}

```

```

Serial.print("IP Address: ");
Serial.println(ip);

// print the received signal strength:
long rssi = WiFi.RSSI();
Serial.print("signal strength (RSSI):");
Serial.print(rssi);
Serial.println(" dBm");
// print where to go in a browser:
Serial.print("To see this page in action, open a browser to http://");
Serial.println(ip);

}

void printMacAddress(byte mac[]) {
    for (int i = 5; i >= 0; i--) {
        if (mac[i] < 16) {
            Serial.print("0");
        }
        Serial.print(mac[i], HEX);
        if (i > 0) {
            Serial.print(":");
        }
    }
    Serial.println();
}

void CortocircuitaPlacaSolar (boolean conmuta){
    if (conmuta){
        digitalWrite(3, LOW);      //Rele
        delay(25);
        digitalWrite(LED_BUILTIN, HIGH);
        digitalWrite(2, HIGH);     //Mosfet
    }
    else{
        digitalWrite(2, LOW);      //Mosfet
        digitalWrite(LED_BUILTIN, LOW);
        digitalWrite(3, HIGH);     //Rele
    }
}

void EnviaDatosNuevaTrazaFloat(WiFiClient cliente){
    int indiceBuffer = 0;
    char sIntensidad []= "{\"i\":\"";
    char stension []= "\",\"v\":\"";
    char sTemperatura[]= "\"temperatura\":\"";
    int temperatura;
    int numeroMuestras;

    /*Envia los datos de la traza en formato float json */
    /* Preparamos los datos en un buffer antes de enviarlos */

    Serial.println("EnviaDatosNuevaTrazaFloat");
    numeroMuestras = TomaMuestras();
    Serial.println ("enviando Muestras");
    cliente.println("Content-type:text/html");
    cliente.println("Connection: close");
    cliente.println();

    //cliente.print("[");
    miBuffer[indiceBuffer+1]='{';
    strcpy(miBuffer+indiceBuffer,sTemperatura,strlen(sTemperatura));
    indiceBuffer += strlen(sTemperatura);

    temperatura = read_temp();
    sprintf(miBuffer+indiceBuffer, "%d", temperatura);
    indiceBuffer+=strlen(miBuffer);
    miBuffer[indiceBuffer+1]=',';

    strcpy(miBuffer+indiceBuffer, "\"muestras\":\"");
    indiceBuffer+=strlen(miBuffer);

    miBuffer[indiceBuffer+1]= '[';

    for (int i = 0; i<numeroMuestras; i++){ //formato float
        if (i) //cliente.print(',');
        miBuffer[indiceBuffer+1]=',';

        //cliente.print("{\"i\":\"");
        strcpy(miBuffer+indiceBuffer,sIntensidad,strlen(sIntensidad));
        indiceBuffer +=strlen(sIntensidad);

        corrienteF = arrCorriente[i]*3.22265*0.0001;
        dtostrf(corrienteF,-9,5,cadenaF);//dato, anchoTotal, decimales, buffer
        //cliente.print(cadenaF);
        strcpy(miBuffer+indiceBuffer,cadenaF,strlen(cadenaF));
        indiceBuffer +=strlen(cadenaF);

        //cliente.print(",\"v\":\"");
        strcpy(miBuffer+indiceBuffer,stension,strlen(stension));
        indiceBuffer +=strlen(stension);

        tensionF = arrTension[i]*3.22265*0.001*6.667;
    }
}

```

```

dtosstrf(tensionF,-9,5,cadenaF);//dato, anchoTotal, decimales, buffer
//cliente.print(cadenaF);
strncpy(miBuffer+indiceBuffer,cadenaF,strlen(cadenaF));
indiceBuffer +=strlen(cadenaF);

//cliente.print(")");
miBuffer[indiceBuffer++]='}';

if(indiceBuffer > 1000){
    cliente.write(miBuffer, indiceBuffer);
    indiceBuffer = 0;
}
}

if(indiceBuffer){
    cliente.write(miBuffer, indiceBuffer);
    indiceBuffer = 0;
}
cliente.println("]");

}

void EnviaDatosNuevaTraza(WiFiClient cliente){
int yPixel;
int xPixel;
Serial.println(".....EnviaDatosNuevaTraza.....");
TomaMuestras();
Serial.println ("nvviendo Muestras");
cliente.println("Content-type:text/html");
cliente.println("Connection: close");
cliente.println();

cliente.print("[");
for (int i = 1; i<NUMERO_MUESTRAS; i++){ //
    //Serial.println(arrCorriente[i]);
    corrienteF = arrCorriente[i]*3.22265*0.0001;
    //Serial.println(corrienteF);
    tensionF = arrTension[i]*3.22265*0.001*6.667;
    //dtosstrf(corrienteF,-9,5,cadenaF);//dato, anchoTotal, decimales, buffer
    yPixel = corrienteF * 2000; //yPixel = (int) corrienteF *(640.0/0.32);
    //Serial.println(yPixel);
    yPixel = 670-yPixel;
    xPixel = (int) (tensionF*(880.0/22.0));
    xPixel += 60; //El centro esta en el punto 60,670

    cliente.print("{\"cx\"::");
    cliente.print(xPixel);
    cliente.print(",\"cy\"::");
    cliente.print(yPixel);
    cliente.print("},");
    //cliente.print(cadenaF);
    //cliente.print(" ");
}
cliente.println("{\"cx\": 110,\"cy\": 200}]");
/*
cliente.print("[{\"cx\": 110,\"cy\": 200},");
cliente.print("{\"cx\": 150,\"cy\": 250}]");
*/
}

void EnviaDatosCliente(WiFiClient cliente){
unsigned int maxArrCorriente = 0; // Para ajustar escalas
unsigned int maxArrTension = 0; //
int xPixel;
int yPixel;
char lineaEje[80];
//cliente.println("Hola");
//Para octave
//cliente.print("corriente");
//cliente.print(numeroSerie);
/**/
* Datos de la placa para ajuste de escala
* Voc = 22 V
* Isc = 0.32A
*/
TomaMuestras();
cliente.print(F("<svg id=\"IV\" height=\"750\" width=\"1000\" style=\"border:1px solid black\">\n"));

//Cada pixel en x corresponde (880/22)*voltios
//Cada pixel en y corresponde a (640/0.32)*corriente
//el origen de coordenadas estará en los pixles 60,30+640
// <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
// por cada punto dibujaremos un circulo
cliente.print(
    "<line x1=\"60\" y1=\"670\" x2=\"940\" y2=\"670\" stroke=\"black\" stroke-width=\"2\" />\n"); //eje x
cliente.print(
    "<line x1=\"60\" y1=\"670\" x2=\"60\" y2=\"30\" stroke=\"black\" stroke-width=\"2\" />\n"); //eje y
cliente.print(
    "<text x=\"300\" y=\"720\" fill=\"black\">Tension (Voltios)</text>\n");
cliente.print(
    "<text x=\"15\" y=\"500\" fill=\"black\" style=\"writing-mode: sideways-lr;\">Corriente (Amperios)</text>\n");

// Atributos de lineas horizontales y verticales
cliente.print ("<g stroke=\"grey\" stroke-width=\"1\" >\n");
for (int i = 1; i< 23; i++){ //Lineas verticales
    sprintf(lineaEje,

```

```

        "<line x1=\"%d\" y1=\"670\" x2=\"%d\" y2=\"30\" />\n",
        i*(880/22)+60,
        i*(880/22)+60
    );
    cliente.print(lineaEje);
}
for (int i = 1; i< 33; i++){ //Lineas horizontales
    sprintf(lineaEje,
        "<line x1=\"60\" y1=\"%d\" x2=\"940\" y2=\"%d\" />\n",
        670-(i*(640/32)+0),
        670-(i*(640/32)+0)
    );
    cliente.print(lineaEje);
}
cliente.print ("</g>\n");

for (int i = 0; i< 23; i++){ //leyenda lineas verticales
    //<text x="0" y="15">I love SVG!</text>
    sprintf(lineaEje,
        "<text x=\"%d\" y=\"690\">%d</text>\n",
        i*(880/22)+60,
        i
    );
    cliente.print(lineaEje);
}

for (int i = 0; i< 33; i++){ //Leyenda Lineas horizontales
    sprintf(lineaEje,
        "<text x=\"20\" y=\"%d\">0.%02d</text>\n",
        670-(i*(640/32)+0),
        i
    );
    cliente.print(lineaEje);
}

// Atributos de punto de grafica
cliente.print ("<g stroke-width=\"0\" fill=\"green\" >\n");
for (int i = 1; i<NUMERO_MUESTRAS; i++){
    Serial.println(arrCorriente[i]);
    corrienteF = arrCorriente[i]*3.22265*0.0001;
    //Serial.println(corrienteF);
    tensionF = arrTension[i]*3.22265*0.001*6.667;
    //dtostrf(corrienteF,-9,5,cadenaF); //dato, anchoTotal, decimales, buffer
    yPixel = corrienteF * 2000; //yPixel = (int) corrienteF *(640.0/0.32);
    //Serial.println(yPixel);
    yPixel = 670-yPixel;
    xPixel = (int) (tensionF*(880.0/22.0));
    xPixel += 60; //El centro esta en el punto 60,670

    cliente.print("<circle cx=\"");
    cliente.print(xPixel);
    cliente.print("\" cy=\"");
    cliente.print(yPixel);
    cliente.print("\" r=\"2\" />\n");
    //cliente.print(cadenaF);
    //cliente.print(" ");
}
cliente.print ("</g>\n");
//cliente.println(";");
//cliente.print("tension");
//cliente.print(numeroSerie);
/*
cliente.print(" ");
for (int i = 1; i<NUMERO_MUESTRAS; i++){
    tensionF = arrTension[i]*3.22265*0.001*6.667;
    dtostrf(tensionF,-9,5,cadenaF);
    cliente.print(cadenaF);
    cliente.print(" ");
}

*/
cliente.println(F("</svg>\n"));

}
// Retorna el numero de muestras significativas en el array de muestras.
int TomaMuestras(void){
    unsigned long microsAntes;
    unsigned long microsDespues;
    unsigned long ultimaMuestraCorriente;
    int nMuestra = 0;
    int incremento = 1;
    int mayorIndiceMuestras=0;

    //CortocircuitaPlacaSolar(false);
    microsAntes = micros();

    ultimaMuestraCorriente = 1; //Para que entre en el bucle
    nMuestra = 0; //Primera muestra que vamos a tomar
    incremento = 1; //Incremento inicial para llenar el
                    // array de muestras
}

```

```

        for (int i = 0; i<NUMERO_MUESTRAS; i++){
            arrTension [i] = 0;
            arrCorriente [i] = 0;
        }
    /*
     * Tomaremos muestras hasta tener corriente cero.
     * Si antes de tener corriente cero, hemos llenado el array de muestras
     * volveremos a recorrer el array de muestras
     * insertando las nuevas tomas en incrementos de 2,3,4, ... y así
     * sucesivamente hasta llegar a un máximo de 20 pasadas para no provocar
     * un time out en el cliente.
    */
    digitalWrite(MOSFET,LOW);
    digitalWrite(RELE,HIGH);
    //delay(1); //Para dar tiempo al rele
    while (ultimaMuestraCorriente>20 || (nMuestra<5)) {
        arrTension [nMuestra] = analogRead(A1);
        ultimaMuestraCorriente = arrCorriente[nMuestra] = analogRead(A2);
        nMuestra +=incremento;
        if(nMuestra > mayorIndiceMuestras)
            mayorIndiceMuestras = nMuestra;

        if (nMuestra > NUMERO_MUESTRAS-1){
            incremento++;
            nMuestra=incremento+5;
        }
        if (incremento == 15) break;
        delay(10);
    }
    digitalWrite(RELE,LOW);
    digitalWrite(MOSFET,HIGH);

    //CortocircuitaPlacaSolar(true);
    microsDespues = micros();

    Serial.print("Tiempo de muestreo en microsegundos:");
    Serial.println(microsDespues-microsAntes);
    Serial.print(F("Numero Muestra en corte: "));
    Serial.println(nMuestra);

    Serial.print(F("Incremento: "));
    Serial.println(incremento);

    Serial.print(F("ultimaMuestraCorriente: "));
    Serial.println(ultimaMuestraCorriente);

    int presentaColumnas=0;

    if(presentaColumnas){
        Serial.println("Corriente Tension");
        for (int i = 0; i<NUMERO_MUESTRAS; i++){
            Serial.println("Corriente --- Tensión");
            Serial.print(arrCorriente[i]);
            Serial.print(" --- ");
            Serial.println(arrTension[i]);
        }
        for (int i = 0; i<NUMERO_MUESTRAS; i++){
            //Tension de referencia a 3.2 V
            //
            //3.3/1024 = 3.22265 exp -3 Para Vref = 3.3V

            /***** Medimos caida de tension en una R de 1 Ohmio
            * a la salida de un amplificador con Av=10
            * Valor Real de la corriente. Av=10, R=1 ohmio
            * Por ello dividimos por 10
            */
            corrienteF = arrCorriente[i]*3.22265*0.0001;
            /* Valor Real de Tension:
            *   Divisor de tension
            *   Vi Tension en la Placa
            *   Vo Tension medida
            *   R1=56.67 Mohm
            *   R2=10 Mohm
            *   Vo=R2/(R1+R2) * Vi
            *   Vi=(R1+R2)/R2 * Vo
            *
            *   Vi=(56.67+10)/10 *Vo
            *   Vi=66.67/10 * Vo
            *   Vi = 6.667*Vo
            */
            tensionF = arrTension[i]*3.22265*0.001*6.667;

            //Presentamos resultados
            dtostrf(corrienteF,-9,5,cadenaF);//dato, anchoTotal, decimales, buffer
            Serial.print(cadenaF);
            Serial.print(" ");
            dtostrf(tensionF,-9,5,cadenaF);
            Serial.println(cadenaF);
        }
    }
    return (mayorIndiceMuestras);
}

```

```

}

void EnviaHtml(WiFiClient cliente, const char *cadenaHtml){
    int escritos;

    escritos = cliente.print(cadenaHtml);
    Serial.print("Escritos=");
    Serial.println(escritos);
    Serial.print("Longitud=");
    Serial.println(strlen(cadenaHtml));
    Serial.println(cadenaHtml);
}

void EnviaFicheroSD(WiFiClient cliente){
    int leidos;
    int indicePunto;
    int i=0;

    for (i= 0; i<strlen(ficheroSolicitado); i++){
        if (ficheroSolicitado[i] == '.')
            indicePunto = i;
    }

    if (strcmp(ficheroSolicitado+i, "htm") == 0)
        cliente.println("Content-type:text/html");
    if (strcmp(ficheroSolicitado+i, "txt") == 0)
        cliente.println("Content-type:text/css");
    if (strcmp(ficheroSolicitado+i, "css") == 0)
        cliente.println("Content-type:text/css");
    if (strcmp(ficheroSolicitado+i, "jpg") == 0)
        cliente.println("Content-type:image/jpeg");
    if (strcmp(ficheroSolicitado+i, "pdf") == 0)
        cliente.println("Content-type:application/pdf");

    cliente.println("Connection: close");
    cliente.println();
    Serial.print("Enviando Fichero ");
    Serial.println(ficheroSolicitado);
    ficheroSd = SD.open(ficheroSolicitado);
    if (ficheroSd) {
        Serial.println(ficheroSolicitado);
        // read from the file until there's nothing else in it:
        while (ficheroSd.available()) {
            leidos = ficheroSd.read(miBuffer, 1024);
            Serial.print("Datos Leidos:");
            Serial.println(leidos);
            //Serial.write(ficheroSd.read());
            cliente.write(miBuffer, leidos);
        }
        // close the file:
        ficheroSd.close();
    } else {
        // if the file didn't open, print an error:
        cliente.println("ERROR 404");
        Serial.println("error opening fichero");
    }
}
void AnalizaActualPeticion(String linea){
    int longitudCadenaGet;
    long nuevoNivel;

    int i,j;

    //Cuando pide un fichero se recibe
    // GET /nombre.abc /HTTP1.1 hay que extraer el nombre.abc

    Serial.print("Linea Completa:");
    Serial.println(linea);
    if (linea.substring(0,5)=="GET /"){
        peticion = NUEVO_FICHERO;
        longitudCadenaGet = linea.length();
        Serial.print("Longitu Cadena Get =");
        Serial.println(longitudCadenaGet);
        //Linea pidiendo pagina
        if (longitudCadenaGet > 5){
            for (i=0,j=0;
                 i<(MAXIMA_LONGITUD_NOMBRE_FICHERO - 1) & j<longitudCadenaGet;
                 i++,j++){
                if (linea.charAt(j+5)==' ') break; //!!! OJO...
                ficheroSolicitado[i]=linea.charAt(j+5);
            }
            ficheroSolicitado[i]=0; //null
        }
        if(strlen(ficheroSolicitado)==0){
            //Fichero index.htm
            strcpy(ficheroSolicitado, "index.htm");
        }
        if (strcmp (ficheroSolicitado, "nDatos.txt") == 0){
            peticion = NUEVA_TRAZA;
        }
        else if (strcmp (ficheroSolicitado, "nDatosFloat.txt") == 0){
            peticion = NUEVA_TRAZA_FLOAT;
        }
        else if (strncmp (ficheroSolicitado, "nL.txt", strlen("nL.txt"))==0){

```

```

        Serial.print("Pide nuevo nivelLuz:");
        Serial.println(ficheroSolicitado);
        //La peticion viene como nL.txt?nv=nivel
        nuevoNivel=atoi(ficheroSolicitado+strlen("nL.txt?nv="));
        Serial.print("Nuevo Nivel Solicitado:");
        Serial.println(nuevoNivel);
        peticion = NUEVO_NIVEL_LUZ;
        //Nivel bajo 1200
        //Nivel alto
        ConversorDA((nuevoNivel*28.95)+1200);
    }
    else {
        peticion = NUEVO_FICHERO;
    }
    Serial.print("Fichero Solicitado =");
    Serial.println(ficheroSolicitado);
} // GET /
}

void ConversorDA(long nivel){

    Serial.print("nivel recibido = ");
    Serial.println(nivel);
    dac.setVoltage(nivel, false);

}
int read_temp()
{
    int rawtemp = analogRead(THERMISTOR_PIN);
    int current_celsius = 0;

    byte i;
    for (i=1; i<NUMTEMPS; i++)
    {
        if (temptable[i][0] > rawtemp)
        {
            int realtemp = temptable[i-1][1] + (rawtemp - temptable[i-1][0]) * (temptable[i][1] - temptable[i-1][1])
            if (realtemp > 255)
                realtemp = 255;

            current_celsius = realtemp;
            break;
        }
    }

    // Overflow: We just clamp to 0 degrees celsius
    if (i == NUMTEMPS)
    current_celsius = 0;

    return current_celsius;
}

```

Índice de figuras

2.1. Esquema del sistema diseñado. PCB: circuito trazador, conversor A/D, tarjeta SD, circuito sensor de temperatura, circuito regulador de iluminación.	4
3.1. Esquema Curva I-V [3]	8
3.2. Superposición de una fuente de corriente con un diodo de unión P-N	9
3.3. Delimitación de zonas de una curva I-V.	10
4.1. Cajas de Conexiones del sistema cerradas. Caja 1, derecha. Caja 2, izquierda.	13
4.2. Cajas de Conexiones del sistema sin cerrar.	14
4.3. Circuito Trazador de Curvas I-V[7]	15
4.4. Algoritmo de toma de muestras[7]	16
4.5. Circuito trazador de curvas I-V, conversor D/A y adaptador tarjeta de memoria micro SD	17
4.6. Sensor de temperatura. Esquema de conexión	18
4.7. Sensor de Temperatura	18
4.8. Representación gráfica de la variación de la resistencia y la tensión Vout con la temperatura	19
4.9. Focos LED utilizados en el sistema de iluminación	20
4.10. Primer prototipo del sistema de iluminación.	21
4.11. Segundo prototipo de sistema de iluminación.	22
4.12. Diseño final del sistema de iluminación.	23
4.13. Esquema eléctrico del circuito de regulación de la iluminación	23
4.14. PCB circuito regulador de iluminación	24
4.15. Conexión Focos LEDs	24
4.16. Diseño final del sistema	25
5.1. Esquema de Página Web para visualización de datos.	27
5.2. Página Web para visualización de datos	28
5.3. Aspecto del inicio del contenido teórico de la página web del proyecto	28
5.4. Diagrama de flujo del sistema completo	29
5.5. Esquema de conexión y Comunicación	30
5.6. Esquema de transmisión y visualización de Datos	31
6.1. Ejemplo curva obtenida en un día soleado al medio día y los parámetros teóricos calculados	33
6.2. Ejemplo curva obtenida con el máximo nivel de iluminación artificial y los parámetros teóricos obtenidos	34
B.1. Arduino MKR1000. Pinout	57

Bibliografía

- [1] Marcelo C. Calvalcanti Aguinaldo J. Nascimento Jr. Emmerson A.Silva, Fabricio Bra-daschia. Parameter estimation method to improve the accuracy off photovoltaics electrical model. *IEEE Journal of Photovoltaics*, 6(1):8, January 2016.
- [2] Christian Van Der Henst John Freddy Vega. Guia html5. el presente de la web. html5, css3 y javascript. 2011.
- [3] A. Rezaee Jordehi. Parameter estimation of solar photovoltaic (pv) cells: A review. *Renewable and Sustainable Energy Reviews*, (61):354–371, 2016.
- [4] Eladio Duran Aranda Miguel A. Martinez Bohorquez Jose M. Andujar Marquez, Juan M. Enrique Gomez. Sistema para la generacion automatica de curvas i-v, p-v y monitiorizacion de modulos fotovoltaicos. *XXV Jornadas de Automatica*.
- [5] Samkeliso Shongwe and Moin Hanif. Comparative analysis of different single-diode pv modeling methods. *IEEE Journal of Photovoltaics*, 5(3):938–946, May 2015.
- [6] Randy Williams Faustino Chenlo Joao L. Afonso Vicente Leite, Jose Batista. Determination of models parameters of pv modules using a low cost i-v tracer. *Instituto Politecnico de Bragan  a, Escola Superior de Tecnologia e Gest  o*, pages 53–60, 2012.
- [7] Elena Alvarez Castro. Tutor: Juan Antonio Lopez Villanueva. Caracterizacion y modelado de celulas solares de silicio en diferentes condiciones de iluminacion. Technical report, Trabajo Fin de Grado. Departamento de Electronica y Tecnologia de Computadores, 2018.
- [8] The World's Largest Web Developer Site w3schools.com.

