

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN



"SISTEMA AUTOMATIZADO DE  
CARACTERIZACIÓN DE PANELES  
SOLARES PARA CUBESATS"

TRABAJO FIN DE GRADO

Septiembre -  
2022

AUTOR: Raúl Gualda Martínez

DIRECTOR/ES: José Antonio Palazón Selva



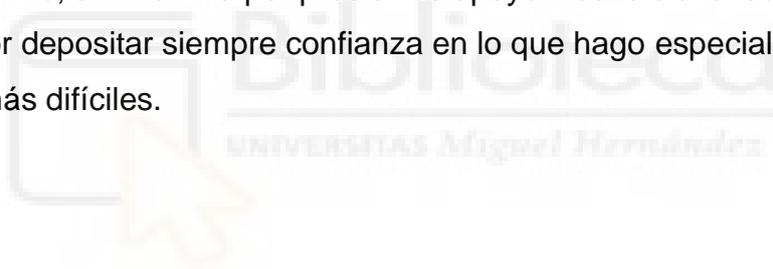
## Agradecimientos

Antes de comenzar el proyecto me gustaría mostrar mis agradecimientos a aquellas personas que han formado parte del trabajo gracias a su ayuda.

En primer lugar, al que ha sido mi tutor y compañero del trabajo, José Antonio Palazón, quien desde el primer momento me ha apoyado aportando ideas y soluciones cuando parecía que no las había para conseguir los mejores resultados.

En segundo lugar, agradecer a la empresa EMXYS y a todos sus empleados por darme la oportunidad de trabajar con ellos y por el apoyo recibido con la realización del trabajo de cada uno de ellos.

Por último, a mi familia por prestarme apoyo incondicional durante todo el proceso y por depositar siempre confianza en lo que hago especialmente en los momentos más difíciles.





# ÍNDICE

ÍNDICE DE FIGURAS .....	7
ÍNDICE DE TABLAS .....	8
ABREVIATURAS.....	9
1. INTRODUCCIÓN .....	10
1.1. Planteamiento del problema.....	16
1.2. Objetivos del proyecto .....	17
2. ESTADO DEL ARTE .....	18
3. DESCRIPCIÓN DEL DISEÑO.....	21
3.1. Diseño Hardware .....	23
3.1.1. Lectura del voltaje e intensidad del panel solar.....	23
3.1.2. Control del sistema de luz solar.....	31
3.1.3. Sensor de temperatura .....	34
3.1.4. Sensor de sol.....	37
3.1.5. Sistema de adquisición de datos y control.....	39
3.1.6. Etapa de alimentaciones .....	41
3.1.7. Interfaz eléctrica .....	44
3.1.8. Esquema de masas .....	45
3.2. Diseño software .....	47
3.2.1. Programa de caracterización en Python.....	47
3.2.2. Interfaz gráfica.....	51
4. IMPLEMENTACIÓN Y PUESTA EN MARCHA.....	55
4.1.1. Diseño y montaje de la PCB.....	55
4.1.2. Ensamblaje de la PCB .....	59
4.1.3. Integración del sistema .....	62
5. ANÁLISIS DE RESULTADOS.....	67
5.1. Resultados experimentales.....	67
6. CONCLUSIONES Y LÍNEAS FUTURAS.....	70
BIBLIOGRAFÍA .....	72
ANEXOS .....	74
I. Lista de materiales .....	74
II. Esquema eléctrico .....	75
III. Planos de la envolvente.....	77
IV. Diagrama de flujo para el método de selección de resistencia .....	80
V. Código fuente del módulo principal .....	81
VI. Código fuente del módulo gráfico .....	86



## ÍNDICE DE FIGURAS

Figura 1. Funcionamiento de una celda solar de triple unión [1].....	10
Figura 2. Método de resistencia variable para la caracterización.....	12
Figura 3. Curva característica I-V de un panel solar con 6 celdas en serie.....	13
Figura 4. Espectro de radiación de un cuerpo negro [2].....	14
Figura 5. Simulador de luz solar diseñado por EMXYS.....	15
Figura 6. Esquema del primer sistema de caracterización.....	16
Figura 7. Pico Solar Simulator de G2V Optics [3].....	18
Figura 8. Sun brick solar simulator [4].....	19
Figura 9. SINUS-300 de WVELABS [5].....	20
Figura 10. Hönle UV SOL 500 [6].....	21
Figura 11. Sistema de caracterización propuesto.....	22
Figura 12. Diagrama de bloques de la descripción del diseño.....	23
Figura 13. Terminales de un transistor MOSFET [7].....	24
Figura 14. Curva VGS de un transistor MOSFET.....	25
Figura 15. Circuito de carga dinámica basado en un MOSFET [8].....	26
Figura 16. Divisor resistivo de la tensión del panel.....	29
Figura 17. Esquemático del relé RT334006.....	32
Figura 18. Circuito de control del relé mediante una señal TTL [9].....	32
Figura 19. Sensor de temperatura AD590.....	35
Figura 20. Circuito de acondicionamiento del sensor de temperatura.....	36
Figura 21. Circuito de acondicionamiento del sensor de luz.....	38
Figura 22. Fuente de alimentación AD/DC TRACO TXL 035-12S [10].....	41
Figura 23. Montaje del regulador de tensión LM317LMX/NOPB.....	42
Figura 24. Circuitos reguladores de tensión.....	44
Figura 25. Plano de masas.....	47
Figura 26. Diagrama de flujo del programa en Python.....	50
Figura 27. Interfaz gráfica para la caracterización de paneles solares.....	51
Figura 28. Plantilla para el método de caracterización VGS Mode.....	53
Figura 29. Plantilla para el método de caracterización Resistance Mode.....	53
Figura 30. Ventana emergente de selección de plantilla.....	54
Figura 31. Visión 3D de la PCB diseñada en Circuit Studio.....	56
Figura 32. Layer de la cara superior de la PCB.....	57
Figura 33. Layer de la cara inferior de la PCB.....	57
Figura 34. Cara superior de la PCB fabricada.....	58
Figura 35. Cara inferior de la PCB fabricada.....	58
Figura 36. PCB montada.....	62
Figura 37. Base del soporte mecánico.....	63
Figura 38. Tapa del soporte mecánico.....	64
Figura 39. Sistema de carga dinámica integrado en el mecanizado.....	66
Figura 40. Resultados de la curva característica I-V para el panel 20210416.....	68

## ÍNDICE DE TABLAS

Tabla 1. Sistemas de adquisición de datos disponibles en el mercado.....	40
Tabla 2. Número de pines necesarios por cada señal .....	45
Tabla 3. Test de funcionamiento de los reguladores de tensión .....	59
Tabla 4. Test de funcionamiento del sensor de temperatura .....	60
Tabla 5. Test de funcionamiento del relé .....	60
Tabla 6. Test de funcionamiento del divisor de tensión .....	60
Tabla 7. Test de funcionamiento del fotodiodo.....	61
Tabla 8. Conexiones entre el DAQ y la PCB .....	65
Tabla 9. Conexiones entre el conector D-Sub y la PCB.....	65



**ABREVIATURAS**

Abreviatura	Descripción
DAQ	Data acquisition
$V_{OC}$	Tensión en circuito abierto del panel solar
$I_{SC}$	Corriente de cortocircuito del panel solar
$P_{mp}$	Punto de máxima potencia del panel solar
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
$V_{TH}$	Tensión umbral de conducción en un MOSFET
$I_D$	Corriente de drenador
$V_{GS}$	Caída de tensión entre la puerta y el surtidor
$V_{SA}$	Solar Array Voltage



## 1. INTRODUCCIÓN

Una de las principales etapas que forma un satélite es la etapa de potencia, encargada de obtener y gestionar la energía aportando autonomía al satélite a lo largo de la misión. Para obtener esta energía se utilizan paneles solares que transforman la energía solar en energía eléctrica.

Los paneles solares están formados por un conjunto de celdas solares conectadas entre sí en serie y/o paralelo. Estos componentes se encargan de obtener la energía solar y de ofrecer una cantidad de energía eléctrica al sistema en función de la irradiancia a la que están expuestas, la temperatura a la que se encuentran y la carga eléctrica conectada a sus terminales.

Las celdas solares utilizadas con más frecuencia en los paneles de satélites son celdas de triple unión. Este tipo de celdas son utilizadas en aplicaciones donde se requiere la mayor eficiencia posible ya que, al tener capas de diferentes materiales, responden a un mayor rango de longitudes de onda absorbiendo una mayor cantidad de radiación solar.

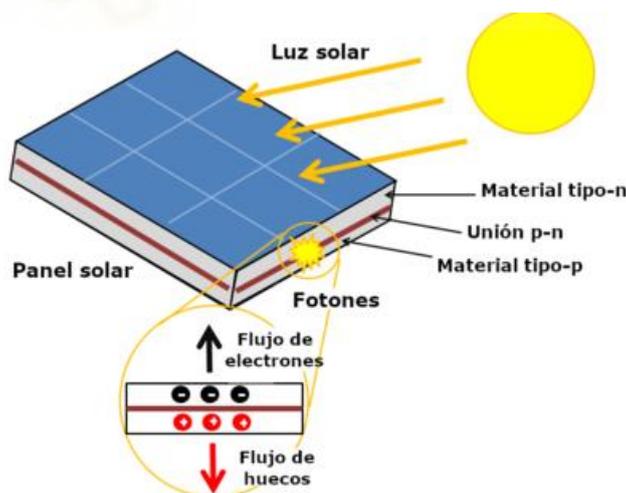


Figura 1. Funcionamiento de una celda solar de triple unión [1]

Como podemos ver en la figura superior, cuando un fotón irradia sobre la superficie de una celda solar, formada por un material tipo N, la energía que

aporta el fotón al material semiconductor se traduce en un cambio de potencial sobre el mismo generándose un flujo de electrones de un material hacia el otro. Aprovechando este flujo de electrones, colocando terminales metálicos en los extremos de la celda, se puede extraer esta energía eléctrica.

Las celdas solares tienen una serie de parámetros eléctricos característicos que describen su funcionamiento. Estos son los siguientes:

- Tensión de circuito abierto ( $V_{OC}$ ): voltaje que entrega la celda solar en sus terminales cuando no tiene ninguna carga conectada, es decir, cuando el circuito está abierto.
- Corriente de cortocircuito ( $I_{SC}$ ): corriente que entrega la celda solar cuando la carga conectada entre sus terminales es un cortocircuito.
- Tensión y corriente de máxima potencia ( $V_{mp}$ ,  $I_{mp}$ ): voltaje e intensidad para los que el panel entrega su máxima potencia.

Como se ha mencionado anteriormente, los paneles están formados por una o más celdas solares. La colocación de estas celdas en serie o paralelo afectará en las propiedades finales del panel solar. Esto es así ya que, al colar celdas en serie el voltaje total del panel será la suma del voltaje de todas ellas, mientras que, si las colocamos en paralelo, el parámetro que se sumará será la corriente. De esta forma los paneles solares diseñados tendrán los parámetros mencionados en la lista superior y vendrán dados en función de la colocación de celdas solares que los forman.

Para que la misión del cubesat se lleve a cabo de forma exitosa, es de vital importancia disponer de una caracterización eléctrica de los paneles solares ya que, en gran medida, la autonomía del sistema vendrá dada por la cantidad de energía que sean capaces de captar y entregar.

La caracterización de los paneles fotovoltaicos consiste en obtener los parámetros de voltaje e intensidad que ofrecen los paneles para unas condiciones constantes de irradiancia y temperatura, y para diferentes demandas de corriente a su salida.

Para obtener los valores de intensidad y voltaje se utiliza el método de resistencia variable. Este procedimiento consiste en colocar una resistencia en serie con el panel y variar su valor desde una resistencia lo suficientemente pequeña como para comportarse como un cortocircuito hasta una de un valor mayor considerada para el panel como un circuito abierto. De esta manera podremos obtener todos los puntos de la curva del panel, desde su corriente de cortocircuito ( $I_{sc}$ ) hasta su tensión de circuito abierto ( $V_{oc}$ ).

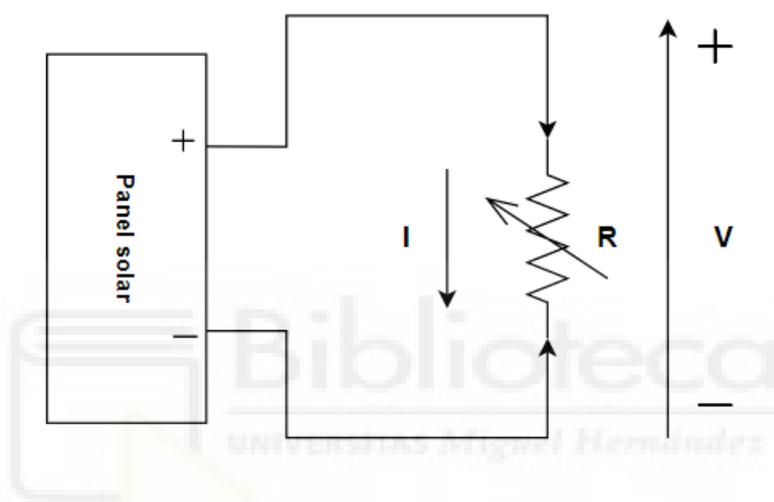


Figura 2. Método de resistencia variable para la caracterización

Una forma de representar gráficamente dicha información es mediante la curva I-V de caracterización, como la que se muestra en la siguiente figura, donde se representa los niveles de intensidad y voltaje del panel solar en función de la carga que tienen conectada a la salida.

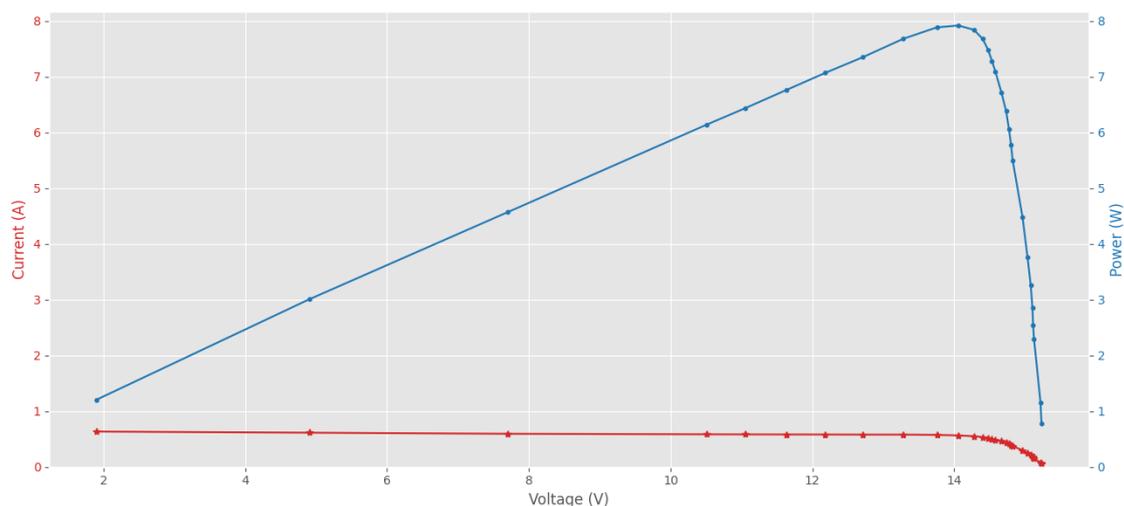


Figura 3. Curva característica I-V de un panel solar con 6 celdas en serie

Mediante el producto entre los valores de corriente y tensión, también se puede obtener también la curva de potencia. Mediante el estudio de la curva característica del panel solar podemos obtener los parámetros de interés mencionados anteriormente, así podemos obtener el punto de máxima potencia de nuestro panel para hacerlo funcionar en esa región y obtener la mayor cantidad de energía posible.

Para realizar estas curvas es necesario excitar el panel solar con luz solar o con un equipo que simule este tipo de luz con una irradiancia constante y que, a su vez, obtenga la curva característica I-V en función de esa irradiancia, la temperatura y la carga conectada. Las condiciones estándar para la caracterización de los paneles solares son de una irradiancia de  $1000 \text{ W/m}^2$ , una temperatura de  $25^\circ\text{C}$  y una distribución espectral AM1.5. Esta distribución espectral corresponde con la emitida por un cuerpo negro como es la luz solar [2]. En la siguiente figura se muestra una representación del espectro electromagnético emitido por el sol.

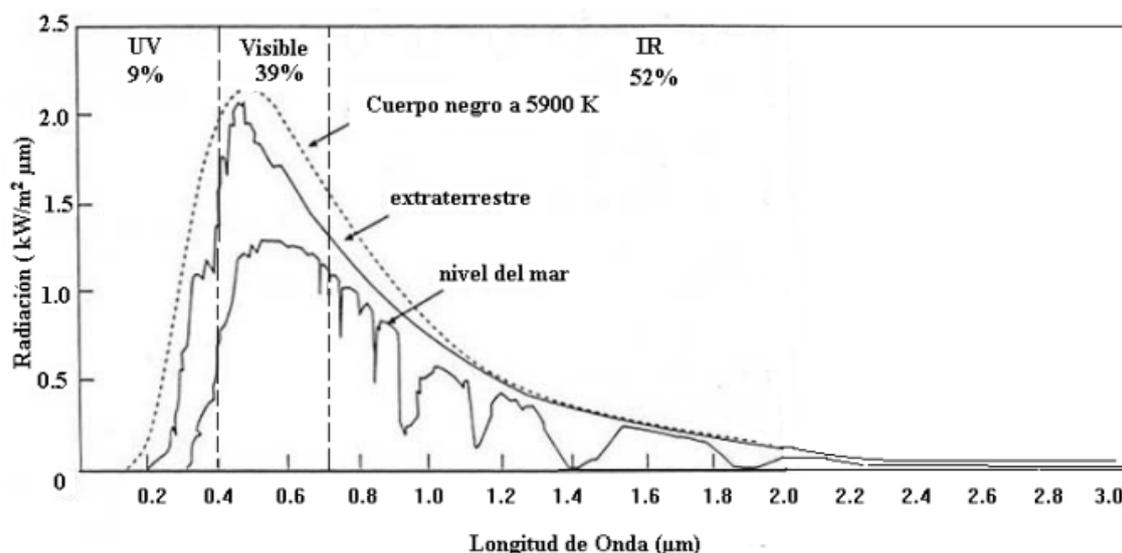


Figura 4. Espectro de radiación de un cuerpo negro [2]

Como se puede comprobar en la gráfica superior, el espectro de la luz solar comprende desde la radiación ultravioleta en menor medida, hasta los infrarrojos en mayor cantidad, pasando por el espectro visible. Además, en función de donde se encuentre el cuerpo que recibe la luz, la radiación a la que estará expuesto variará. De esta forma, el dispositivo que ejerza como simulador de luz solar deberá tener una distribución espectral que se asemeje a la ideal de un cuerpo negro, marcada en discontinuo en la gráfica superior.

La información extraída en la caracterización de cada panel permite verificar su rendimiento una vez fabricados. Además, permite validar procedimientos de fabricación y detectar degradaciones durante las pruebas de aceptación y calificación previas a su integración en cualquier satélite. Por este motivo, la línea de producción de paneles solares de EMXYS necesita disponer de un equipo que permita realizar la caracterización de forma fiable, rápida y segura durante su fabricación. Además de realizar dicha caracterización, el equipo deberá validar el funcionamiento de los sensores de temperatura y de detección de luz solar que incorpora.

EMXYS cuenta con el diseño de un simulador de luz solar utilizando un foco LED con una distribución que permite caracterizar paneles de cubesats bajo el estándar 3U (30x10 cm). Este tiene forma cúbica, con una compuerta en la

parte superior que nos permita introducir los paneles solares. En su interior se encuentran los focos que irradiarán los paneles y, también, los ventiladores con el fin de regular la temperatura de los focos y los paneles. En su carcasa externa, cuenta con dos interruptores, uno trasero y otro delantero. El trasero sirve para encender los ventiladores y proporcionar la potencia al resto del sistema, mientras que el interruptor delantero sirve para encender o apagar el sistema de iluminación. En la siguiente figura se muestra el simulador solar diseñado por EMXYS.



*Figura 5. Simulador de luz solar diseñado por EMXYS*

En cuanto al sistema de carga dinámica y adquisición de datos, la primera opción planteada para caracterizar los paneles solares consiste en utilizar una carga electrónica dinámica comercial. Este dispositivo, conectado a los terminales del panel solar, permite generar una carga resistiva y, a su vez, tomar los valores de tensión y corriente. La carga dinámica utilizada en este apartado es la RIGOL DL3021A, la cual, conectada con un USB a un ordenador, se puede programar mediante scripts de Python para fijar una resistencia en concreto y para tomar valores de voltaje e intensidad.

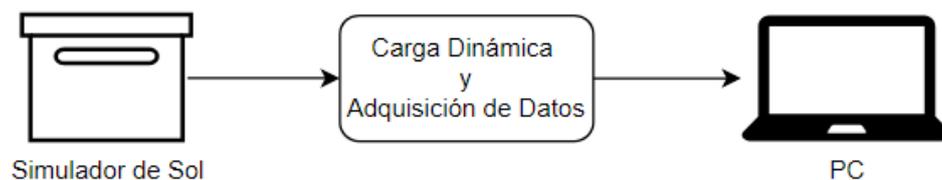


Figura 6. Esquema del primer sistema de caracterización

El método de caracterización consiste en que un operario, mediante un portátil conectado a la carga dinámica por USB, ejecute un script de Python que se comunica con este dispositivo estableciendo una carga constante y toma los datos de voltaje e intensidad del panel solar. A su vez, el operario ha de encender y apagar manualmente el simulador de luz solar irradiando el panel solar para poder obtener los valores I-V.

### 1.1. Planteamiento del problema

Sin embargo, el método de caracterización mencionado en el apartado anterior, tiene una serie de inconvenientes que hacen poco viable utilizarlo para obtener las propiedades eléctricas de los paneles solares de una forma óptima. Estos problemas son los siguientes:

- Es un proceso manual, es decir, se necesita de un operario que encienda y apague la luz del simulador a la vez que lanza los scripts de Python para cada una de las resistencias.
- Se requiere de una carga dinámica que sea capaz de establecer un valor de resistencia fijo, que mida la corriente y el voltaje que entrega el panel para esa resistencia y, que se pueda comunicar mediante USB con un ordenador mediante una librería de programación como por ejemplo Python. El instrumento utilizado es el RIGOL DL3021A con un precio de 849,95€ en SOTEL lo cual encarece notablemente el precio del sistema.
- El tiempo que pasa el simulador de luz solar encendido mientras la carga dinámica fija la resistencia y posteriormente adquiere los datos

de voltaje e intensidad es suficiente como para que los paneles solares se calienten. Durante este tiempo el panel se ve afectado por el calor de los focos y el calor generado por la conducción de las celdas en funcionamiento. Cuando la temperatura del panel solar aumenta, el rendimiento de este se reduce. Esto supone que las medidas de tensión e intensidad no son a una temperatura constante, lo cual es deseable.

- Para la lectura del sensor de temperatura y el fotodiodo de los paneles se necesitarían circuitos acondicionadores y sistemas de adquisición de datos externos a la carga dinámica.

## 1.2. Objetivos del proyecto

Con el fin de evitar los problemas mencionados en el apartado anterior se ha planteado una nueva alternativa que consiste en diseñar un sistema que esté embebido en el simulador de luz solar, sustituyendo a la carga dinámica y que sea capaz de lo siguiente:

- Obtener los pares de valores de voltaje e intensidad del panel solar para una temperatura constante y representar la curva característica.
- Reducir el tiempo que está expuesto el panel solar a la irradiación disminuyendo el periodo de adquisición de datos, evitando así que estos se sobrecalienten y la caracterización sea la correcta.
- Adquirir los datos de temperatura, así como realizar el test de funcionamiento del fotodiodo de los paneles solares.
- Automatizar el proceso de encendido y apagado del simulador de luz solar y el de adquisición y procesado de los datos.
- Reducir el coste que supondría utilizar un sistema externo encargado de realizar la adquisición de datos.
- Comunicarse con un ordenador para almacenar y procesar los datos.
- Tener todo el sistema (simulador de luz solar, carga dinámica y adquisición de datos) en un mismo elemento.

## 2. ESTADO DEL ARTE

Encontrar un sistema con las características mencionadas anteriormente no es algo común en el mercado. Sin embargo, debido al auge de la utilización de los sistemas fotovoltaicos y el desarrollo de la tecnología de nanosatélites, se han diseñado modelos comerciales de simuladores de sol para la caracterización y el testeado de módulos fotovoltaicos.

Un ejemplo de ellos es el modelo comercial Pico Solar Simulator del fabricante G2V Optics [3]. Este equipo es un simulador de luz solar con tecnología LED que opera en todo el espectro electromagnético con el estándar AM1.5. Cuenta con la posibilidad de variar el tipo y la cantidad de radiación que incide sobre el panel a testear así como la de escoger entre radiación terrestre o extraterrestre. Además, incluye un software de caracterización I-V que permite obtener las curvas características de los paneles solares de forma automática.



*Figura 7. Pico Solar Simulator de G2V Optics [3]*

El problema de este simulador es que su superficie es de 2.5x2.5 cm siendo insuficiente para caracterizar un panel 3U.

Sin embargo, el mismo fabricante dispone de un segundo simulador de luz solar con tecnología led con un área de irradiación mayor. Se trata del Sun Brick Solar Simulator [4] que dispone de las mismas características en cuanto a lo que se refiere a radiación, pero el área de iluminación es de 25x25 cm. Además, permite la unión de varios de estos dispositivos para generar mayores áreas de radiación. En cambio, este modelo no tiene un software de caracterización I-V como sí lo tenía el anterior, por lo que habría que implementarlo.



*Figura 8. Sun brick solar simulator [4]*

Un segundo modelo comercial es el SINUS-300 del fabricante WVELABS [5]. Al igual que el modelo anterior, se trata de un simulador de luz solar de tecnología LED con espectro AM1.5, el cual tiene una interfaz gráfica que nos permite modificar los parámetros de funcionamiento. En este caso el área de funcionamiento es más grande (24x24 cm) siendo, aún, insuficiente para caracterizar un panel de 3U o 6U diseñados en EMXYS. No obstante, el simulador cuenta con un software de caracterización de paneles mediante una carga dinámica. Gracias a este programa el simulador proporciona la curva característica del panel, así como sus parámetros más importantes.

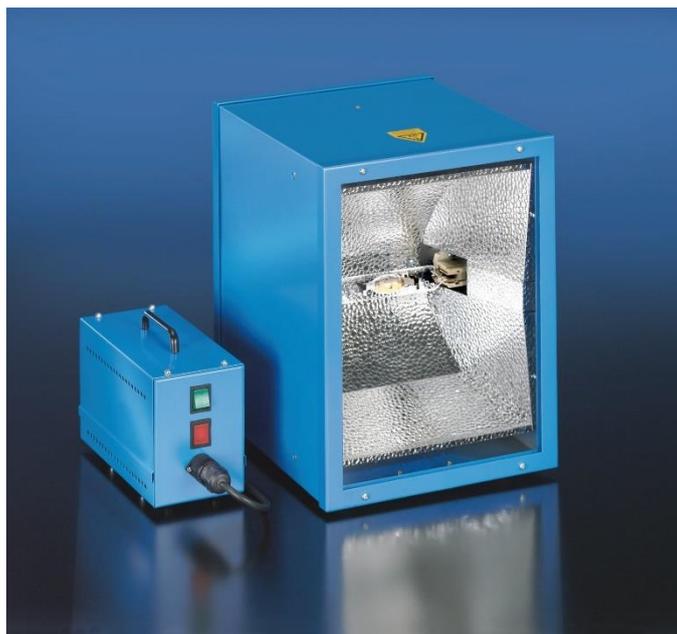


Figura 9. SINUS-300 de WAVELABS [5]

Como se puede comprobar, el principal problema que presentan estos modelos comerciales es el área de trabajo, ya que es insuficiente para caracterizar los paneles con estándar 3U (30x10cm). Además, en el caso del Sunbrick Solar Simulator de G2V, el precio del modelo más básico parte de 9600€ con la posibilidad de añadir el software de caracterizado I-V por 3240€ + IVA siendo muy elevado.

Una segunda opción que evite los problemas de dimensionado consiste en diseñar un simulador de luz solar utilizando uno o más focos. Estos han de ser capaces de generar una irradiancia con espectro AM1.5 en condiciones de caracterización estándar para el área necesitada según los paneles que se vayan a diseñar (3U en el caso de EMXYS).

Un ejemplo de un foco de estas características es el Hönle UV SOL 500 cuyo precio es de 3674€.



*Figura 10. Hönlé UV SOL 500 [6]*

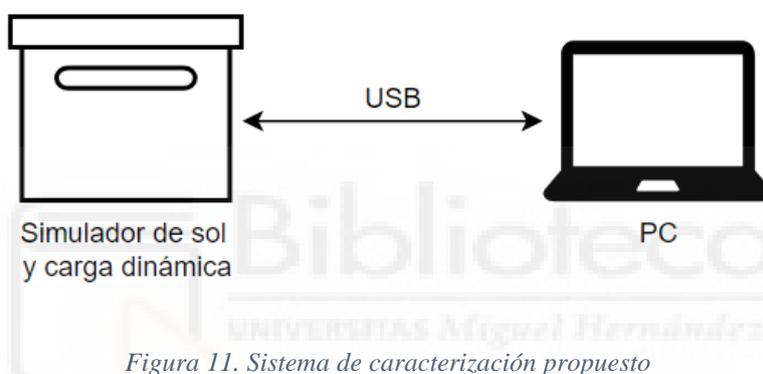
Teniendo en cuenta que EMXYS dispone de un foco de características similares al mencionado anteriormente, se ha optado por realizar el diseño completo del simulador de luz solar mediante este foco y que, a su vez, incluya la caracterización de los paneles solares, así como la adquisición de los datos de temperatura y del fotodiodo. De esta forma el sistema se adaptará a las necesidades de EMXYS en cuestiones como la superficie de caracterización, el espectro de radiación (AM1.5) y adquisición de datos.

### **3. DESCRIPCIÓN DEL DISEÑO**

Antes de comenzar con el diseño del sistema, se ha de hacer un dimensionado eléctrico, es decir, definir los parámetros máximos de tensión y corriente que ha de soportar el sistema de medida. Estos vendrán definidos por los propios paneles diseñados en EMXYS, por lo tanto, se tomará como tensión máxima de funcionamiento 30V y corriente máxima de funcionamiento 1.6A.

El sistema de caracterización al completo, tal y como podemos ver en la Figura 11, estará compuesto por tres secciones:

- Simulador de luz solar: panel de focos y bombillas encargados de proporcionar la irradiancia necesaria al panel solar.
- Carga dinámica y adquisición de datos: sistema que, fija la carga eléctrica conectada al panel solar en serie, toma los pares de datos de voltaje e intensidad para ese valor de resistencia y adquiere los datos del sensor de temperatura y del fotodiodo.
- Ordenador: sistema encargado de controlar la carga dinámica, la adquisición de datos y, posteriormente, procesa y almacena los datos obteniendo la curva característica del panel.



*Figura 11. Sistema de caracterización propuesto*

Por lo tanto, tendremos que diseñar un sistema que se pueda incrustar en el simulador solar y que realice lo siguiente: ejercer como carga variable, ocuparse de la adquisición de datos y del control del sistema (encendido y apagado del foco) y que se comuniquen con un ordenador externo para enviar los comandos de control. En la Figura 12 se muestra un diagrama con los bloques principales que formarán nuestro sistema.

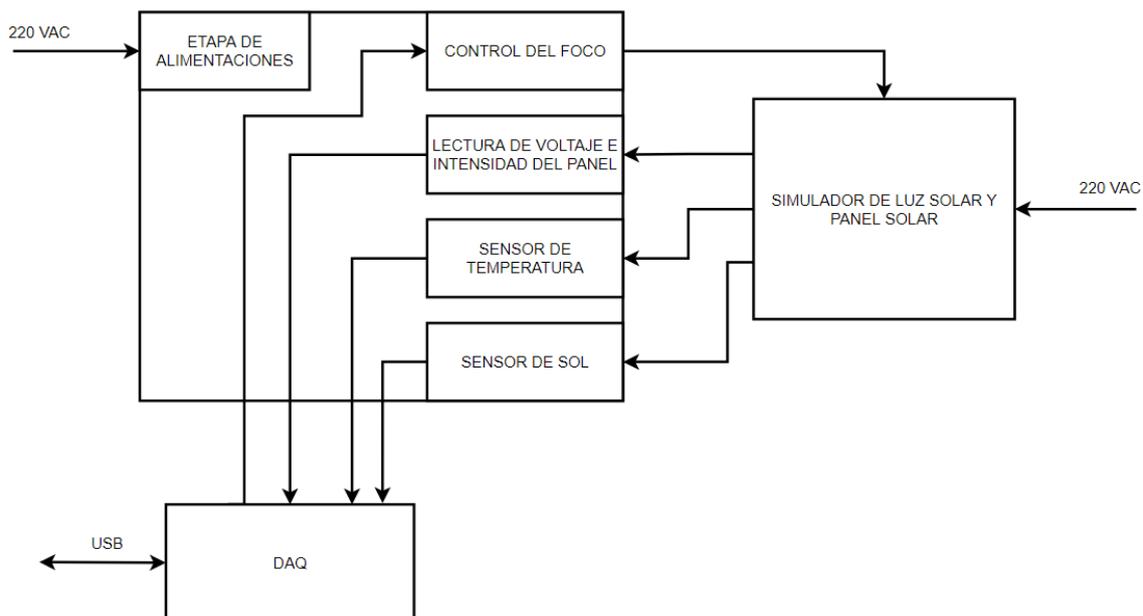


Figura 12. Diagrama de bloques de la descripción del diseño

### 3.1. Diseño Hardware

La primera etapa del diseño de nuestro sistema será la Hardware. En este apartado modelizaremos e implementaremos cada uno de los bloques que forma el sistema de caracterización al completo.

#### 3.1.1. Lectura del voltaje e intensidad del panel solar

En este bloque del sistema se diseñará la carga dinámica programable que nos permita, por un lado, fijar una resistencia eléctrica variable conectada en serie al panel y que, a la vez, obtenga los datos de voltaje e intensidad del panel solar.

### 3.1.1.1. Circuito de carga dinámica

Para diseñar la carga dinámica que cumpla con los requisitos mencionados anteriormente, se hará funcionar un transistor MOSFET como resistencia variable.

Este tipo de transistores están formados por 3 terminales: el drenador (D), el surtidor (S) y la puerta (G). Una de las principales características de estos transistores es que, bajo el terminal de la puerta, existe una capa de óxido que impide el paso de corriente. Esto significa que el control del componente se realiza mediante la aplicación de una tensión y no de corriente.

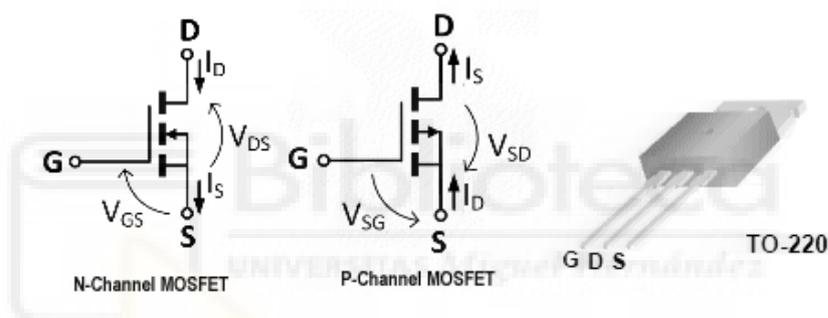


Figura 13. Terminales de un transistor MOSFET [7]

En un MOSFET de canal N, al aplicar una tensión positiva a la puerta, se genera un campo eléctrico permitiendo el paso de electrones entre el drenador y surtidor. Por lo tanto, la cantidad de corriente que fluye del drenador al surtidor ( $I_D$ ) vendrá dada en función de la tensión que hay entre puerta y surtidor ( $V_{GS}$ ). Sin embargo, no se generará un campo eléctrico hasta que la tensión  $V_{GS}$  supere un valor de tensión umbral denominado comúnmente  $V_{TH}$ , donde se pueden diferenciar dos casos:

- $V_{GS} \leq V_{TH}$ : el transistor estará en zona de corte por lo que la corriente que atraviesa el drenador es 0 ( $I_D=0$ ). En estas condiciones el MOSFET se comporta como un circuito abierto.

- $V_{GS} \geq V_{TH}$ : el transistor comienza a conducir pasando a la zona óhmica por lo que la corriente de drenador a surtidor es distinta de cero.

Una vez el transistor se encuentra en la zona óhmica, este tiene un comportamiento resistivo. Cuanto mayor sea la tensión que cae de puerta a surtidor, la corriente que atraviesa el drenador será mayor, es decir, la resistencia será menor. La cantidad de corriente que atraviesa el drenador en función de la tensión  $V_{GS}$  nos la proporciona el fabricante del transistor.

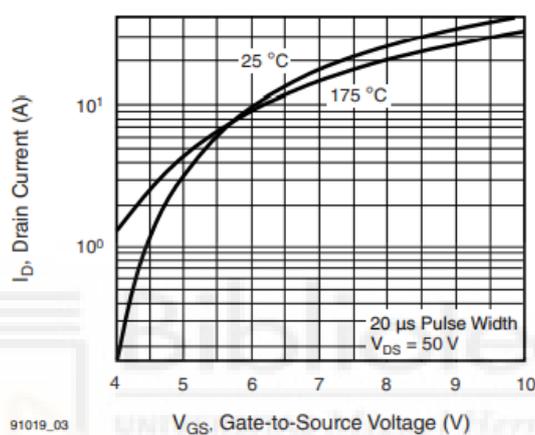


Figura 14. Curva VGS de un transistor MOSFET

Sabiendo de la posibilidad de utilizar el transistor MOSFET como resistencia variable, el siguiente paso consiste en diseñar el circuito que nos permita modificar la tensión  $V_{GS}$ , de forma que podamos obtener los valores de voltaje e intensidad de los paneles solares de forma controlada. El circuito propuesto es el siguiente:

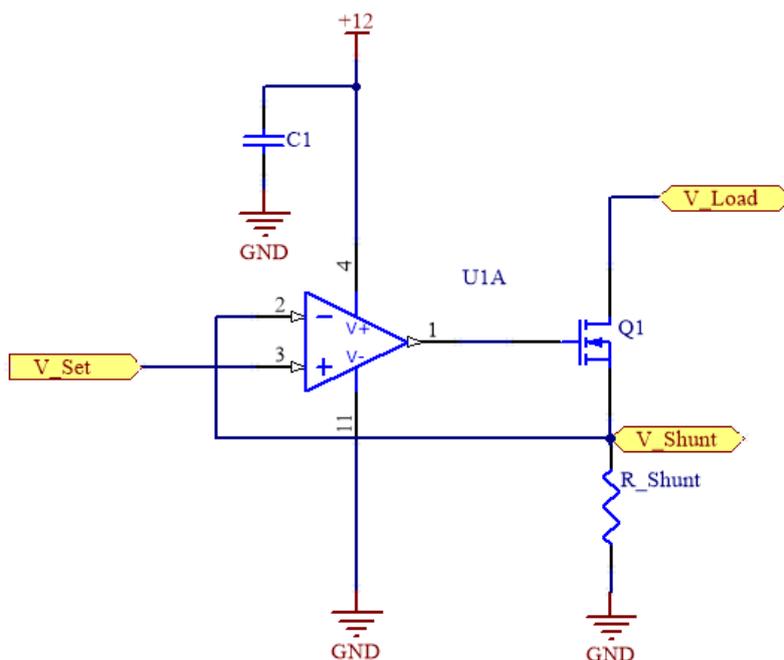


Figura 15. Circuito de carga dinámica basado en un MOSFET [8]

El funcionamiento del circuito se basa en la teoría de cortocircuito virtual de los amplificadores operacionales, donde se sabe que la tensión aplicada en ambas entradas es la misma. De esta forma, podremos crear un bucle de realimentación negativa que nos permita controlar la corriente que le demandamos al panel solar en forma de tensión.

Para ello, conectaremos la entrada negativa del amplificador operacional al surtidor del MOSFET seguido de una resistencia de Shunt de un valor conocido. Así, fijando la tensión en la entrada positiva podremos determinar indirectamente la corriente solicitada al panel solar con la caída de tensión en la resistencia de Shunt ( $V_s$ ). Por otro lado, la salida del amplificador se conectará a la puerta del transistor ( $V_G$ ), de forma que este se encargará de ajustar la tensión de puerta de forma que el transistor entre en conducción y deje pasar la corriente solicitada, entre drenador y surtidor, en forma de tensión en el surtidor ( $V_s$ ). Por lo tanto, la corriente que le pidamos al panel solar vendrá dada por la siguiente ecuación:

$$I_{panel} = \frac{V_{set}}{R_{shunt}}$$

En nuestro caso, el transistor utilizado es un MOSFET de potencia canal N IRF530PBF dada su disponibilidad en el stock de EMXYS y su baja resistencia ( $R_{DS(ON)}$ ) cuando se encuentra en conducción. Además, al circular corriente de drenador a surtidor se disipará potencia en el componente en forma de calor por lo tanto, se colocará un disipador de calor pegado al encapsulado evitando que este se dañe.

A la hora de seleccionar el amplificador operacional se ha de tener en cuenta que el voltaje a la entrada de su terminal positiva es el que fija la corriente del panel solar según lo mostrado en la ecuación superior. Es por esto por lo que el amplificador que escojamos ha de ser rail to rail tanto de entrada como de salida. Esto quiere decir que, el rango de voltajes de entrada y salida podrá variar en todo el rango de alimentación del amplificador. De esta forma podremos utilizar valores de voltaje pequeños para solicitar valores pequeños de corriente sin ningún problema, de igual forma que la salida del amplificador no estará limitada a un valor de tensión y podrá saturar al valor de la alimentación. El componente seleccionado que cumple las características es el amplificador operacional LMP7704MTX/NOPB, proporcionando además 4 amplificadores operacionales en un mismo integrado.

Por otro lado, sabiendo el valor de la resistencia de Shunt, podremos calcular la corriente que entrega el panel solar tomando el voltaje que cae entre sus terminales. Teniendo en cuenta que por ella circularán grandes cantidades de corriente, su valor ha de ser lo suficientemente pequeño como para disipar la mínima potencia posible. Sin embargo, si el valor de resistencia es muy pequeño, las variaciones de tensión que tendremos que aplicar para modificar la corriente que solicitemos al panel serán muy pequeñas y se necesitaría mucha precisión.

Por ejemplo, si escogemos una resistencia de  $0.1\Omega$  y queremos una precisión de 10mA, aplicando la ley de Ohm, necesitaríamos un sistema con una precisión de 1mV como mínimo. De esta forma, el valor de la resistencia

escogida es de  $3\Omega$  ya que es un valor comercial, ganamos resolución y no es un valor muy elevado. Sabiendo el valor de la resistencia y el valor de corriente máximo que pasará por ella, calcularemos la potencia máxima que soportará mediante la ley de Ohm:

$$P_{MAX} = V * I = R * I^2 = 3 * 1.6^2 = 7.68W$$

El modelo de resistencia escogido es el PWR220T-20-3R00F, soportando una potencia máxima de 20W.

Para alimentar el amplificador se ha de tener en cuenta el rango de tensiones que necesitamos para cumplir con los requisitos de diseño. Es de especial interés el caso donde mayor corriente se extrae al panel, ya que, en nuestro caso será cuando circulen 1.6A, y, para que se de esta condición, la tensión que cae en la resistencia de Shunt ( $V_s$ ) ha de ser 5V. La tensión umbral ( $V_{GS}$ ) de conducción que nos proporciona el fabricante del transistor es de aproximadamente 4.5V. Por lo tanto, a la puerta del MOSFET, o lo que es lo mismo, a la salida del amplificador, deberemos tener una tensión mayor a (4.5 + 6) V. Así, la salida podrá saturar al valor máximo de alimentación, permitiendo que el transistor opere en todo el rango de corrientes.

Por lo tanto, la alimentación positiva del integrado será a 12V, así como la alimentación negativa estará llevada a tierra. Además, utilizaremos un condensador de desacoplo de 100nF a la entrada de la alimentación de 12V para eliminar ruido.

#### 3.1.1.2. Acondicionamiento del voltaje del panel solar

Gracias a la resistencia de Shunt mencionada en el apartado anterior, podemos conocer la corriente que entrega el panel solar. Además, se sabe que el voltaje máximo que se medirá entre los terminales de esta resistencia será de

5V, valor inferior al máximo soportado por un sistema de adquisición de datos comercial. Sin embargo, la tensión máxima de funcionamiento de nuestros paneles solares será de 30V, pero el sistema de adquisición de datos utilizado (apartado 3.4) soporta entradas analógicas de hasta 10V. Por lo tanto, tendremos que diseñar un circuito que nos permita medir todo el rango de voltajes posibles sin dañar el sistema de adquisición de datos.

Para evitar dañar la electrónica del módulo de adquisición de datos, se necesita un circuito que reduzca la tensión en 3 veces su valor, de forma que para el valor máximo de salida del panel solar (30V) a la entrada del DAQ obtengamos el valor máximo permitido (10V). El punto de medida del voltaje del panel será en el drenador del transistor. El circuito propuesto se encuentra representado en la Figura 16.

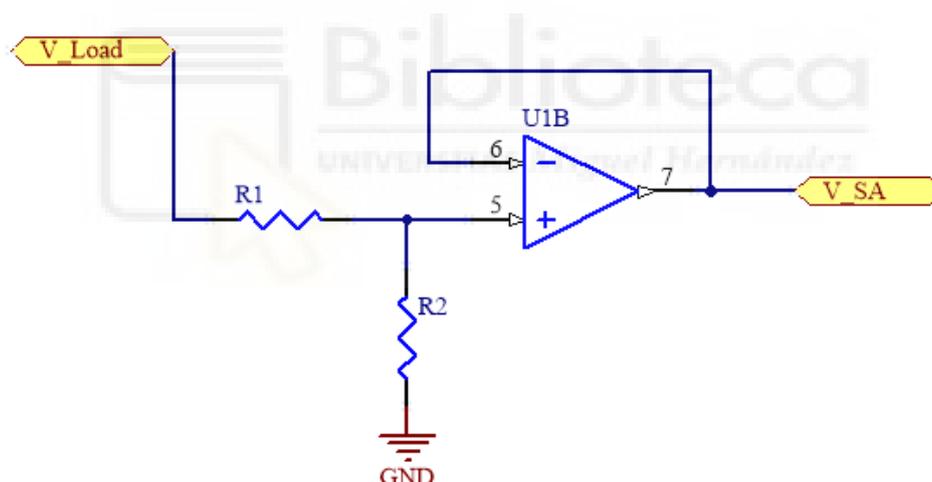


Figura 16. Divisor resistivo de la tensión del panel

Para reducir la tensión se utilizará un divisor resistivo donde la relación entre el voltaje de salida y el de entrada viene dado por la siguiente relación de resistencias:

$$V_{out} = \left( \frac{R2}{R1 + R2} \right) * V_{in}$$

Sabiendo que el factor de reducción que han de cumplir las resistencias es de 1/3, si resolvemos la ecuación obtenemos que:

$$R_1 = 2 * R_2$$

Las resistencias escogidas que cumplen con los requisitos son  $R_1 = 10K\Omega$  y  $R_2 = 4K99\Omega$ .

Para leer el voltaje del panel solar con el factor de reducción deseado, si conectásemos el voltaje de salida del divisor de tensión directamente al sistema de medida, estaríamos modificando el factor de reducción ya que el sistema de medida se comportaría como una resistencia en paralelo a la resistencia a  $R_2$  alterando el factor de reducción.

Para solventar este problema se ha utilizado un seguidor de tensión aprovechando uno de los amplificadores operacionales del LMP7704MTX/NOPB. La tensión de salida del divisor resistivo se conecta a la entrada positiva del amplificador operacional, mientras que la entrada negativa se conecta a la salida de este. Puesto que la impedancia de entrada de los amplificadores operacionales es muy elevada, la tensión de salida del divisor resistivo no se va a ver modificada, teniendo el factor de reducción original. Al tener la misma tensión en la entrada positiva que en la negativa, a la salida del amplificador tendremos la tensión preparada para ser medida.

De esta forma la tensión que llega al sistema de medida ( $V_{SA}$ ) viene dada por la siguiente ecuación:

$$V_{SA} = \left( \frac{R_2}{R_1 + R_2} \right) * V_{LOAD} = \left( \frac{4k99}{10k + 4k99} \right) * V_{LOAD} = 0.333 * V_{LOAD}$$

A la hora de obtener el valor real de la tensión del panel solar para el procesado de datos habrá que tener en cuenta el factor de reducción, invirtiéndolo para obtener el valor real del voltaje.

### **3.1.2. Control del sistema de luz solar**

Como se ha mencionado anteriormente, uno de los principales objetivos del proyecto es automatizar el proceso de caracterización de los paneles solares. Para ello necesitamos diseñar una sección de control que se encargue de gestionar el apagado y encendido del iluminado del simulador de luz solar.

El primer paso es sustituir el interruptor de encendido y apagado de las luces, por uno o varios relés que actúen como interruptores automáticos activados por una señal digital. Como la alimentación de los focos y las bombillas, es de corriente alterna, se ha decidido utilizar dos relés, uno para la fase y otro para el neutro, asilándolos el uno del otro. Los relés escogidos tienen que cumplir los siguientes requisitos:

- Soportar tensiones de hasta 220V en alterna.
- La tensión de excitación de la bobina ha de ser continua y con un voltaje de funcionamiento de 6V.
- Ha de ser sin enclavamiento, para que cuando se desactive la señal de control cambie de estado el relé y no se quede enganchado en un mismo estado.

A la hora de seleccionar el modelo de relé, aparte de tener en cuenta los requisitos anteriores, se escogerá aquel que soporte la mayor corriente posible para que, en los picos de corriente en el encendido del sistema de iluminación, el relé no se dañe.

Teniendo todo esto en cuenta, el modelo de relé seleccionado es el RT334006 de TE Connectivity, soportando corrientes de hasta 16A en sus contactos.

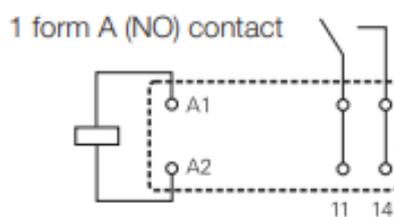


Figura 17. Esquemático del relé RT334006

Según la hoja de datos del fabricante la bobina del relé se excita con una tensión en continua de 6V cambiando de estado a normalmente abierto, mantiene el estado para valores superiores a 4.2V y cambia de estado a normalmente cerrado cuando la tensión se reduce hasta 0.6V.

La señal que activa o desactiva el relé es una señal digital que, por lo general, tendrá un nivel de tensión de 5V o 3.3V. Estos niveles de tensión digitales son insuficientes para activar la bobina del relé que trabaja a 6V. Para evitar este problema, utilizaremos un transistor BJT con la siguiente configuración:

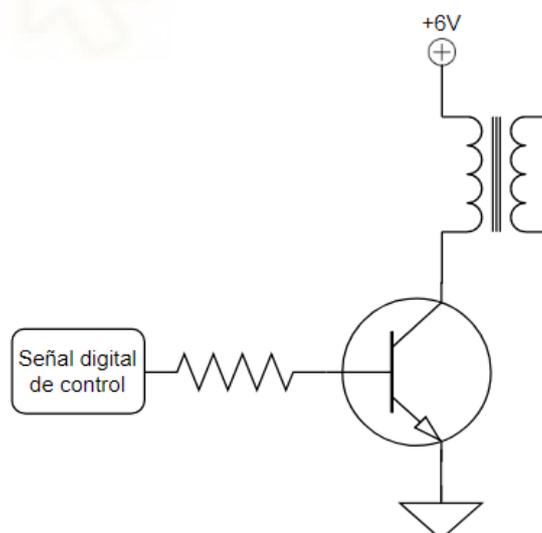


Figura 18. Circuito de control del relé mediante una señal TTL [9]

Un transistor NPN de este tipo tiene 4 tipos de funcionamiento:

- Activa: el transistor conduce y la corriente que atraviesa de colector a emisor es proporcional a la corriente que atraviesa la base de este.
- Inversa: se trata del mismo funcionamiento que en activa salvo que la corriente fluye inversamente, es decir, de emisor a colector.
- Corte: el transistor se comporta como un circuito abierto ya que no conduce.
- Saturación: el transistor se comporta como un corto circuito donde la corriente fluye de colector a emisor.

En nuestro caso nos interesa que el transistor funcione o en corte o en saturación. Si se encuentra en corte, el transistor no conducirá y por tanto no habrá corriente que atraviese la bobina del relé y este no se activará. Sin embargo, si el transistor está en su estado de saturación, fluirá corriente del colector al emisor, o lo que es lo mismo, habrá corriente en la bobina lo que hará que el relé se active. Es decir, funcionará como un interruptor.

Un transistor NPN pasa a encontrarse en saturación cuando la tensión que cae entre la base y el emisor supera un umbral. En la mayoría de los casos esta tensión es de aproximadamente 0.6V. Por el contrario, cuando la tensión entre base y emisor es menor que el umbral, el transistor pasa a estado de corte.

Utilizando este circuito conseguimos aislar la tensión de control del relé de la tensión de la alimentación del mismo, ya que, aplicando una tensión de un nivel inferior en la base del transistor, conseguiremos hacer que conduzca la corriente por la bobina y, por tanto, cambiaremos su estado.

El último paso para completar el diseño del circuito es obtener el valor de la resistencia de base que, para la tensión de control, permita entrar al transistor en su modo de conducción, es decir, que la tensión  $V_{BE}$  sea mayor de 0.6V.

La ecuación que rige el funcionamiento de nuestro circuito de control es la siguiente:

$$V_{Control} = I_b * R_b + V_{BE}$$

O lo que es lo mismo:

$$V_{Control} = \frac{I_c}{\beta} * R_b + V_{BE}$$

Sabiendo que la corriente de colector es la misma que la corriente que atraviesa la bobina del relé (80mA medidos en conducción), que la tensión de control es de 5V, que la  $\beta$  de los transistores se puede aproximar a 100 y que la tensión de base a emisor ha de ser mayor de 0.6V, se puede calcular el valor máximo de resistencia para el cual el transistor conduce.

$$R_{b,m\acute{a}x} = \frac{(V_c - V_{BE}) * \beta}{I_c} = \frac{(5 - 0.6) * 100}{0.08} = 5.5k\Omega$$

Puesto que el valor mencionado es el valor máximo de resistencia que podemos escoger para que funcione el circuito, la resistencia escogida para este es de 2.2k $\Omega$ .

### 3.1.3. Sensor de temperatura

Los paneles diseñados y fabricados por EMXYS tienen integrado en su parte trasera (interna al satélite) un sensor de temperatura. El modelo utilizado para monitorizar su temperatura en todo momento es el AD590. Este integrado se comporta como una fuente de corriente que proporciona 1 $\mu$ A/K para cualquier voltaje de alimentación superior a los 4V, midiendo temperaturas entre -55 $^{\circ}$ C hasta +150 $^{\circ}$ C.

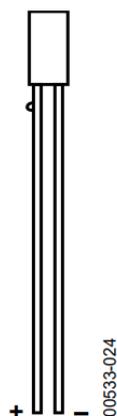


Figura 19. Sensor de temperatura AD590

Para alimentar el AD590 utilizaremos los 6V disponibles de uno de nuestros reguladores de tensión (apartado 3.2.4). Este nivel de tensión será llevado al terminal positivo del sensor mediante un conector externo.

Al comportarse como una fuente de corriente el sensor de temperatura, colocando una resistencia de un valor conocido en serie con este, podremos calcular la tensión que cae en la resistencia y, de forma implícita, la temperatura del sensor.

Puesto que las variaciones de tensión en la resistencia en función de la temperatura son muy pequeñas, se utilizará un condensador en paralelo a la resistencia con el fin de filtrar el ruido y tener la mejor precisión posible. Además, utilizaremos un diodo Zener de 10V, también en paralelo a la resistencia, para evitar que, en el caso de producirse una sobretensión mayor a este valor, se dañe el DAQ.

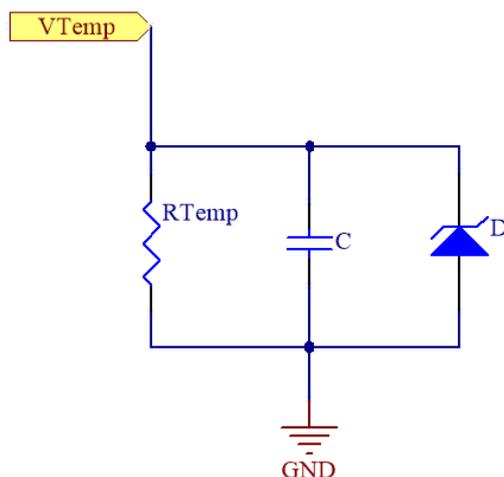


Figura 20. Circuito de acondicionamiento del sensor de temperatura

De esta forma, conociendo el valor de la resistencia, las ecuaciones que rigen la temperatura en función de la tensión y viceversa son las siguientes:

$$V_{Temp} = (T_{AD590(^{\circ}C)} + 273.2) * (I_{AD590} * R_{Temp}) \quad (1)$$

$$T(^{\circ}C) = \left( \frac{V_{temp}}{R_{Temp}} \right) * 10^6 - 273.2 \quad (2)$$

Sabiendo que el valor de tensión que leeremos es proporcional a la resistencia que coloquemos en serie con el sensor, tendremos que elegir una resistencia que nos proporcione unos valores de voltaje aceptables para todo el rango de temperaturas.

En nuestro caso utilizaremos una resistencia de  $2490\Omega$  siendo el voltaje mínimo de  $0.543V$  para la temperatura mínima ( $-55^{\circ}C$ ) y el máximo  $1.054V$  para la máxima temperatura ( $150^{\circ}C$ ). Por último, el condensador escogido será un condensador pequeño, escogiendo  $100nF$  en nuestro caso.

Con el circuito ya diseñado, una vez obtengamos el valor de salida en tensión del circuito acondicionador del AD590, usando la segunda ecuación superior se puede obtener la temperatura del panel en grados centígrados.

No obstante, al obtener la temperatura interesa tener una resolución de, al menos, 1°C. Para ello, calcularemos la diferencia de voltajes que hay para dos temperaturas que difieran en 1°C y de esta forma sabremos la resolución necesaria que ha de tener el sistema de adquisición de datos para poder distinguir temperaturas en un grado.

$$V_{0^{\circ}\text{C}} = (0 + 273.2) * \left( \frac{1\mu\text{A}}{\text{K}} * 2490 \right) = 0.680268 \text{ V}$$

$$V_{1^{\circ}\text{C}} = (1 + 273.2) * \left( \frac{1\mu\text{A}}{\text{K}} * 2490 \right) = 0.682758 \text{ V}$$

$$\text{Resolución} = V_{1^{\circ}\text{C}} - V_{0^{\circ}\text{C}} = 0.0025 \frac{\text{V}}{^{\circ}\text{C}}$$

Como conclusión, el sistema de medida ha de ser capaz de medir variaciones de tensión de 2.5mV como mínimo, para tener una resolución de 1°C.

#### 3.1.4. Sensor de sol

Al igual que con el sensor de temperatura, cada uno de los paneles solares tiene incrustado en su superficie un fotodiodo. Este se comporta como un sensor de luz ambiental que es sensible a la irradiación solar. Cuando una fuente lumínica lo irradia comienza a conducir corriente. Por lo tanto, para su correcto funcionamiento ha de polarizarse en inversa permitiendo el flujo de electrones una vez recibe la luz solar en nuestro caso.

Con los datos del sensor de luz de cada uno de los 4 paneles que forman el cubesat, se puede determinar la orientación del mismo en función de si los fotodiodos detectan luz o no en el panel.

El problema de este componente reside en que, una vez el fotodiodo recibe luz solar, este comienza a conducir una corriente muy pequeña que es imposible medir con nuestro sistema de adquisición de datos ni el del propio satélite. Es por esto por lo que se ha diseñado un circuito que amplifique esta señal de corriente y proporcione una señal de tensión medible cuando el fotodiodo se encuentre en conducción. Así, midiendo un valor de voltaje podremos determinar si el panel está recibiendo luz solar o no, y si el sensor lumínico funciona correctamente. En la Figura 21 se muestra el circuito diseñado.

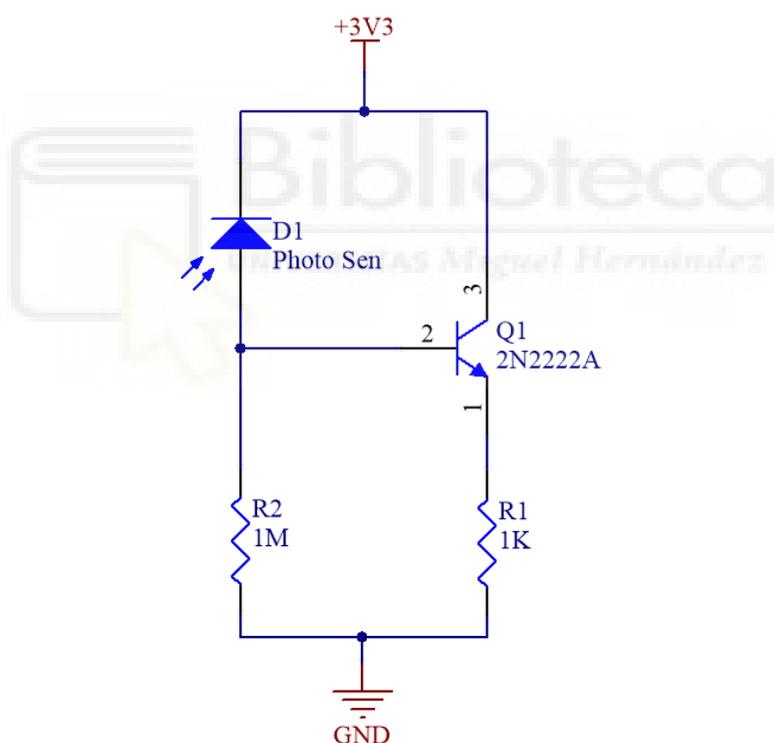


Figura 21. Circuito de acondicionamiento del sensor de luz

No obstante, a la hora de diseñar el circuito para nuestro sistema, hay que tener en cuenta que el fotodiodo se encuentra en el panel solar. Por lo tanto, utilizaremos un conector de 2 pines que nos permita conectar las señales externas del fotodiodo provenientes del panel solar en nuestro circuito de acondicionamiento.

### 3.1.5. Sistema de adquisición de datos y control

Una de las partes más importantes de nuestro sistema es aquella que se encargará de la adquisición de datos, como por ejemplo el voltaje y la intensidad del panel, y la gestión de las señales de control, tanto digitales como analógicas.

En este apartado se abren dos posibilidades. Una de ellas sería diseñar el sistema desde cero, donde tendríamos que incluir circuitos con conversores analógico-digitales y viceversa, circuitos para utilizar señales digitales, así como, implementar una interfaz de comunicación de datos que nos permita comunicar nuestro sistema con un ordenador externo para interactuar con él. La segunda posibilidad es utilizar una tarjeta de adquisición de datos comercial que nos permita programarlo y comunicarnos con él mediante una interfaz ya definida.

En nuestro caso, se ha optado por la segunda opción ya que, a pesar de que el coste de un módulo de estas características puede ser elevado, reduce el tiempo total de diseño del sistema. Además, este tipo de módulos permiten su programación con Python utilizando librerías propias simplificando notablemente la adquisición de datos.

Antes de escoger el DAQ se han de conocer las necesidades del sistema para poder establecer los requisitos que ha de cumplir. Estos son los siguientes:

- **4 entradas analógicas:** encargadas de muestrear los voltajes del panel solar, de la resistencia de shunt, de la temperatura y del fotodiodo.
- **1 salida analógica:** encargada de fijar el voltaje en el surtidor del MOSFET.
- **1 salida digital:** encargada de general la señal digital que activa el relé.

Una vez establecidos los requisitos se ha realizado una búsqueda de diferentes módulos comerciales para realizar una comparativa y seleccionar el que mejores prestaciones proporcione en relación a su precio.

Fabricante	Nombre	Entradas Analógicas	Salidas Analógicas	Resolución (bits)	Tasa de Muestreo	I/O Digitales	Software	Interfaz de Comunicación	Precio (€)	Stock
National Instruments	USB-6001	8	2	14	20KS/s	13	ANSI C, Python, LabVIEW	USB 2.0	270	Sí
Measurement Computing	USB-202	8	2	12	100KS/s	8	Python, Linux, Labview	USB 2.0	146	Sí
Measurement Computing	USB-205	8	2	12	500KS/s	8	Python, Linux, Labview	USB 2.0	192,38	Sí
Omega	OM-USB-1408FS	8	2	14	50KS/s	8	Visual Studio, Labview	USB 2.0	320	Sí

*Tabla 1. Sistemas de adquisición de datos disponibles en el mercado*

La tarjeta elegida es la USB-205 de Measurement Computing ya que cumple todos los requisitos necesarios, y, a pesar de ser más costosa que su versión anterior (USB-202) la tasa de muestreo es mucho más elevada, proporcionándonos mucha más velocidad a la hora de muestrear las diferentes señales. Otro motivo de su elección es que proporcionan una librería para Python con las funciones necesarias para comunicarse mediante USB con el módulo. Las especificaciones completas del DAQ USB-205 proporcionadas por el fabricante son las siguientes:

- **Entradas Analógicas:**
  - Número de canales: 8.
  - Resolución: 12 bits.
  - Tasa de Muestreo: hasta 500kS/s.
  - Muestreo: multiplexado.
  - Rango: +/-10V.
- **Salidas Analógicas:**
  - Número de canales: 2.
  - Resolución: 12 bits.
  - Tasa de Muestreo: hasta 250S/s.
  - Rango: 0-5V.
- **I/O Digitales:**
  - Número de Canales: 8.
  - Counter/Timer: 1.

### 3.1.6. Etapa de alimentaciones

La alimentación externa que utilizaremos para nuestro sistema completo es de 12V. Para proveer este nivel de tensión utilizaremos una fuente de alimentación AC/DC externa de 12V que nos permita encender nuestro dispositivo conectándolo a la red eléctrica.

La fuente seleccionada es la TRACO POWER TXL 035-12S con las siguientes características:

- **Modelo:** TXL 035-12S
- **Entrada:**
  - Voltaje: 100-240VAC.
  - Corriente: 1.0A máx.
  - Frecuencia: 47-63Hz.
- **Salida:**
  - Voltaje: 12.0VDC.
  - Corriente: 3.0A.
  - Potencia: 36W máx.



Figura 22. Fuente de alimentación AD/DC TRACO TXL 035-12S [10]

Sin embargo, en la placa hay diferentes etapas que trabajan con niveles de tensión inferiores. Por un lado, tenemos la alimentación de los relés y del sensor de temperatura a 6V, mientras que por el otro tenemos la alimentación del circuito acondicionador del fotodiodo, que es de 3.3V

Para obtener estos niveles de voltaje independientemente de la tensión de entrada, se han utilizado reguladores de tensión lineales ajustables. El regulador escogido es el LM317LMX/NOPB. Este integrado permite reducir la tensión a su salida independientemente de la de entrada, utilizando una configuración de resistencias.

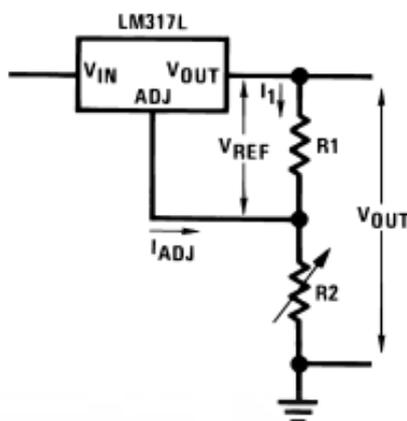


Figura 23. Montaje del regulador de tensión LM317LMX/NOPB

La ecuación que rige el voltaje de salida, según la hoja de datos del componente es la siguiente:

$$V_{OUT} = V_{REF} * \left(1 + \frac{R_2}{R_1}\right) + I_{ADJ} * R_2$$

Siguiendo las especificaciones del fabricante, sabiendo que la tensión de referencia  $V_{REF}$  es de 1.25V y que la corriente de ajuste es lo suficientemente pequeña como para poder despreciar el término  $I_{ADJ} * R_2$ , si fijamos el valor de la resistencia  $R_1$  y el voltaje de salida deseado ( $V_{OUT}$ ), podemos obtener el valor de la resistencia  $R_2$  despejando la ecuación superior de la siguiente forma:

$$R_2 = \left( \frac{V_{OUT}}{V_{REF}} - 1 \right) * R_1$$

Puesto que necesitamos dos niveles de tensión distintos a partir de los 12V, utilizaremos dos reguladores de tensión, uno para cada tensión de salida buscada. Estos estarán colocados en cascada, de forma que, primero obtendremos los 6V a partir de los 12V y, a partir de los 6V del primer regulador, se sacarán los 3.3V. Sabiendo las tensiones de salidas que se buscan, el siguiente paso es calcular la relación de resistencias para obtener esos voltajes.

En nuestro caso, fijaremos la resistencia  $R_1$  de ambos reguladores a  $240\Omega$  según lo aconsejado por el fabricante.

$$R_{2, 6V} = \left( \frac{6V}{1.25V} - 1 \right) * 240 \quad R_{2, 6V} = 912\Omega$$
$$R_{2, 3.3V} = \left( \frac{3.3V}{1.25V} - 1 \right) * 240 \quad R_{2, 3.3V} = 394\Omega$$

Los valores de resistencia reales que más se acercan a los obtenidos teóricamente son:

$$R_{2, 6V} = 910\Omega \quad R_{2, 3.3V} = 402\Omega$$

Por último, el fabricante recomienda utilizar condensadores de desacoplo de 100nF entre el voltaje de entrada y tierra para mejorar la respuesta transitoria. Los circuitos reguladores de tensión resultantes se muestran en la Figura 24.

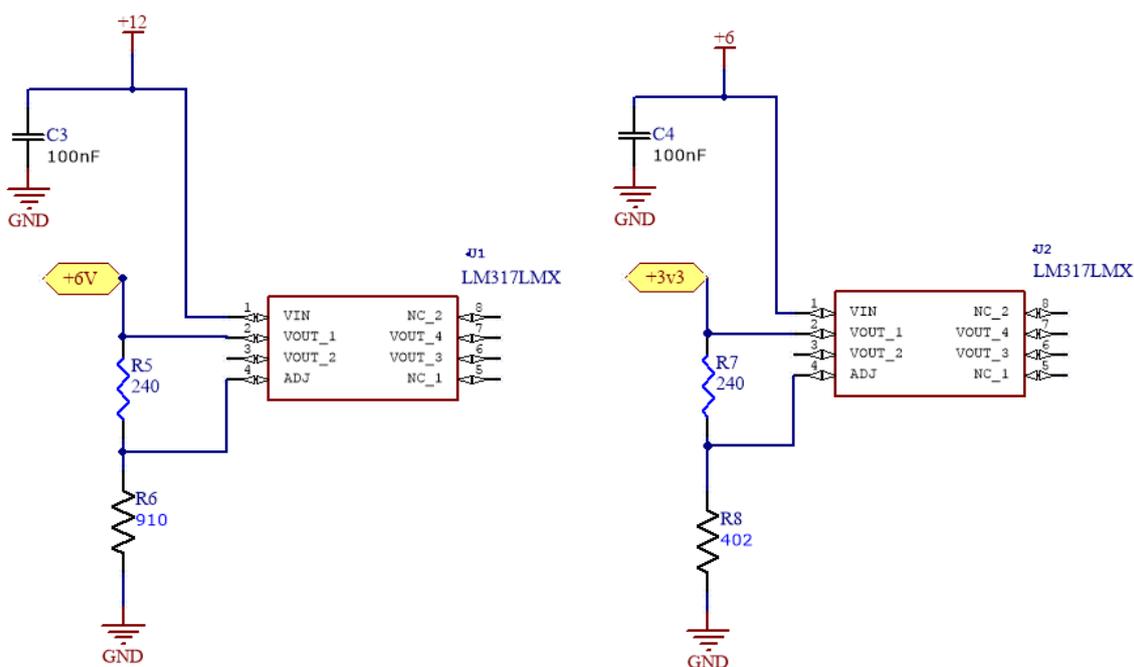


Figura 24. Circuitos reguladores de tensión

### 3.1.7. Interfaz eléctrica

Cabe destacar que, en nuestro sistema, nos encontramos con señales que salen de la PCB que diseñaremos, como por ejemplo la alimentación de 6V del sensor de temperatura, al igual que las hay que provienen del exterior como la tensión del panel solar o la de alimentación. Por este motivo, a la hora de diseñar el esquemático y la PCB, tendremos que añadir conectores en aquellas señales que necesitemos llevar o traer del exterior.

Para saber cuántos y qué tipo de conectores necesitamos, se ha hecho la siguiente lista:

Descripción	Nº de pines
Alimentación 12V	2
Entrada y salida de la fase en el relé	2
Entrada y salida del neutro en el relé	2
Sensado de la tensión de Shunt	1

Entrada analógica de voltaje para fijar la corriente	1
Terminales del panel solar	2
Señal digital de control	2
Tensión de salida del fotodiodo	1
Tensión de salida del sensor de temperatura	2
Sensado del voltaje del panel solar	1
Ánodo del fotodiodo	1
6V, 3V3	2
GND	3

*Tabla 2. Número de pines necesarios por cada señal*

En todos los casos utilizaremos conectores de cable a placa permitiéndonos conectar los cables y desconectarlos fácilmente.

Para la alimentación de 12V y las entradas y salidas de la fase y el neutro utilizaremos terminales que nos permitan utilizar cables de mayor grosor, mientras que para el resto de las señales primaremos reducir el tamaño al máximo posible ya que en esos casos no tenemos señales de potencia.

### **3.1.8. Esquema de masas**

Puesto que el sistema está formado por diferentes subsistemas, es de vital importancia contar con un plano de masas que detalle cada una de las referencias a masa de los subsistemas que forman el proyecto.

Para empezar, para cada uno de los diferentes elementos que forman nuestro sistema por separado, diferenciaremos entre las referencias a tierra que tienen diferenciando entre corriente alterna o continua.

- Simulador de luz solar: en este sistema contamos únicamente con la corriente alterna que se encarga de aportar energía a los focos y a los ventiladores.

- Fuente de alimentación AC/DC: este dispositivo tiene, por un lado, corriente alterna que viene de la red eléctrica, mientras que por otro tiene la corriente continua con la que alimentaremos la carga dinámica.
- DAQ: el módulo de adquisición trabaja exclusivamente en continua, teniendo dos masas diferentes para señales analógicas y para señales digitales.
- PCB: cuenta con un único plano de masa general para corriente continua, aislado de la corriente alterna que pasa por los relés.
- Panel solar: funcionan en corriente continua, teniendo un terminal positivo y otro negativo.

En cuanto a las señales analógicas de continua (aquellas que forman parte de la carga dinámica), es importante que estén referenciadas a una misma masa común. De lo contrario, encontraríamos señales que se encuentran flotando y, a la hora de realizar mediciones, los valores obtenidos no serían los correctos. Por ejemplo, si la masa del sistema de adquisición de datos es diferente a la del panel solar, tomará medidas de voltaje e intensidad desvirtuados. Para ello, aprovecharemos el plano de masas de la PCB para referenciar todas las masas analógicas y digitales de aquellos dispositivos que las tengan. En la siguiente figura se muestra el plano de masas utilizado.

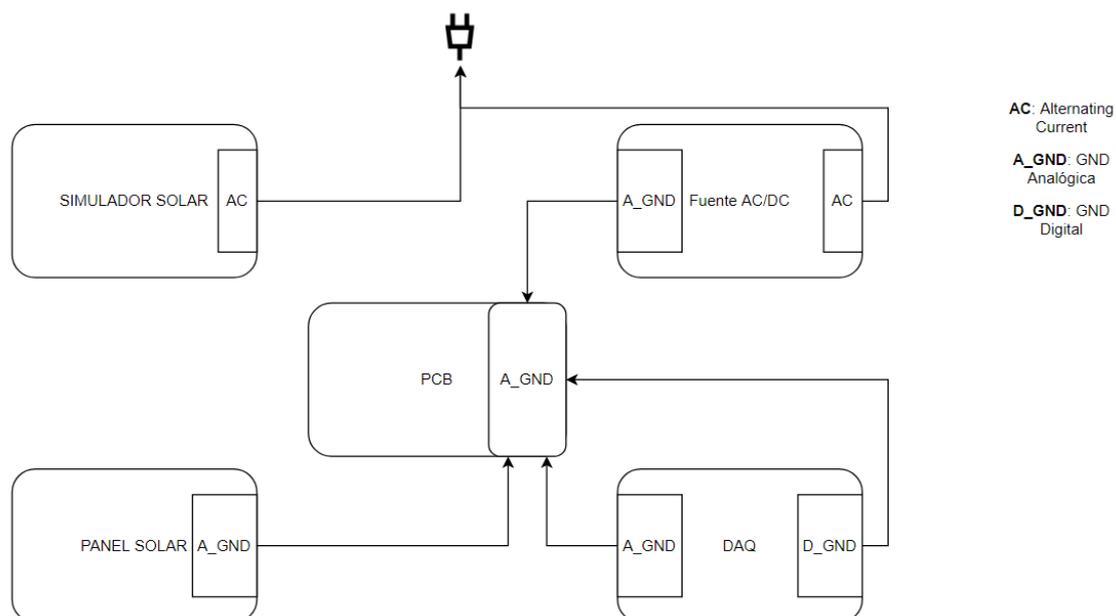


Figura 25. Plano de masas

### 3.2. Diseño software

La siguiente etapa de diseño del proyecto es la etapa software. En este apartado diseñaremos un programa que se encargue de comunicarse con nuestro sistema controlándolo y obteniendo los datos para la caracterización de los paneles y para las pruebas de temperatura y del fotodiodo.

El programa estará implementado en el lenguaje Python ya que la tarjeta de adquisición de datos incluye una librería en este lenguaje que permite la programación del módulo y la adquisición de datos con una interfaz USB. Esta librería permite el uso de funciones como la lectura de entradas analógicas o la generación de señales de salida tanto analógicas como digitales [11].

#### 3.2.1. Programa de caracterización en Python

Antes de comenzar a codificar el programa necesitamos establecer las necesidades que ha de cumplir este para completar todos los test necesarios y realizar la correcta lectura y almacenamiento de los datos.

Para realizar la curva característica I-V del panel solar necesitamos obtener los datos de voltaje e intensidad, los cuales lo haremos mediante la lectura y muestreo analógico de las entradas del DAQ. Sin embargo, para obtener todos los valores de la curva, tenemos que modificar la tensión en el surtidor del MOSFET cambiando así, la resistencia que hay conectada en el panel solar. Para conseguir estos valores, realizaremos un barrido de tensión con la salida analógica del DAQ.

Los valores que necesitamos para realizar el barrido de tensión y obtener los pares I-V son los siguientes:

- Voltaje máximo en el surtidor del MOSFET, o lo que es lo mismo, intensidad máxima que se le va a pedir al panel solar. Podemos obtener ambos valores gracias a que se conoce de antemano el valor de la resistencia de Shunt.
- Voltaje mínimo en el surtidor del MOSFET o corriente mínima que le pediremos al panel solar.
- Número de muestras que queremos tomar entre el valor de intensidad máximo y el valor de intensidad mínimo solicitadas al panel.

Con estos tres parámetros podemos definir el rango de corrientes que le solicitaremos al panel solar para realizar su curva característica. De esta forma, podremos almacenar cada uno de los puntos con los pares de voltaje e intensidad leídos para posteriormente representar la curva.

En el caso del sensor de temperatura y el fotodiodo su lectura es mucho más sencilla puesto que, al tener el circuito de acondicionamiento diseñado, únicamente hay que leer su voltaje de salida mediante las entradas analógicas del DAQ y almacenar sus valores.

Cabe destacar que el proceso de adquisición de estos datos se realizará una vez el simulador de luz solar esté encendido. Por lo tanto, el programa ha de gestionar también el encendido y apagado del foco.

Una vez se hayan obtenido los datos de voltaje e intensidad, los del sensor de temperatura y los del fotodiodo, el siguiente paso será almacenarlos para

poder interpretarlos y tener acceso a ellos posteriormente. Para ello generaremos una hoja de cálculo donde guardaremos una fila con cada una de las muestras que tomemos de los datos a conocer: voltaje, intensidad, potencia, temperatura y voltaje del fotodiodo. Además, este fichero contará con el número de referencia del panel que se ha caracterizado como elemento identificativo.

Por último, con todas las muestras de voltaje e intensidad, generaremos una gráfica donde se mostrará la curva I-V y la curva de potencia del panel solar y la guardaremos en forma de imagen junto al fichero con todos los datos.

A la hora de desarrollar el programa, hay que tener en cuenta que para ejecutarse necesitaremos introducirle los parámetros deseados por el usuario, es decir, el número de referencia, la intensidad máxima y mínima del panel y el número de muestras que se quieren tomar.

En la Figura 26 se muestra un diagrama completo del funcionamiento interno del programa.



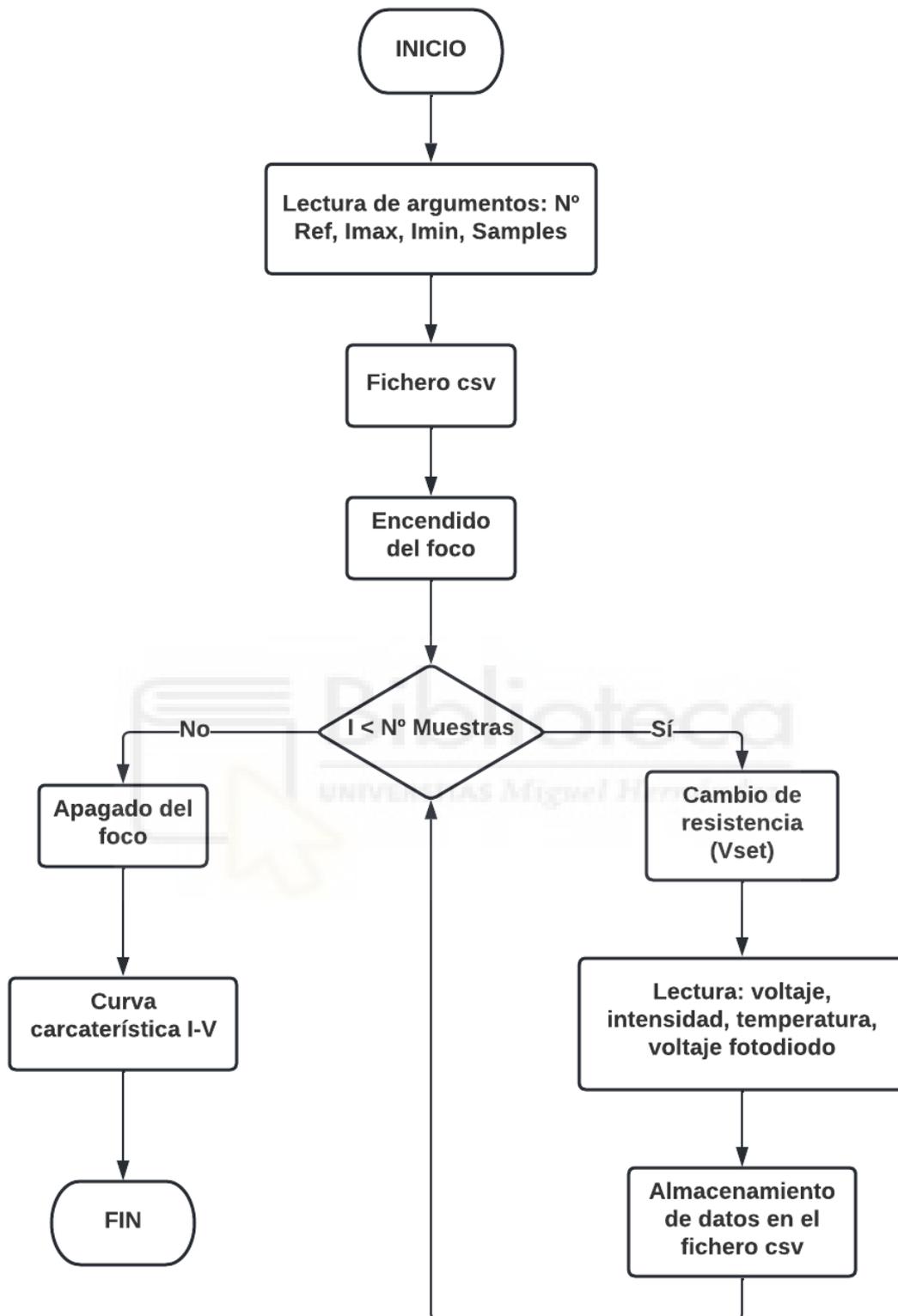


Figura 26. Diagrama de flujo del programa en Python

### 3.2.2. Interfaz gráfica

Con el fin de facilitar el manejo del programa de caracterización de los paneles solares, así como hacerlo más intuitivo y funcional para el operario que realice las pruebas de funcionamiento no necesite tener conocimientos de programación en Python, se ha diseñado una interfaz gráfica. Esta aplicación se ha desarrollado utilizando la librería gratuita Tkinter de Python. Este módulo permite la creación de interfaces gráficas de una forma muy sencilla.



Figura 27. Interfaz gráfica para la caracterización de paneles solares

Tal y como podemos ver en la Figura 27 la interfaz de caracterización consta de una única pantalla donde tenemos 3 partes diferenciadas. Por un lado, tenemos los valores de voltaje, corriente y temperatura del panel solar en tiempo real.

Por otro lado, tenemos un pequeño panel de control con una serie de botones y desplegables con diferentes funciones. Los botones de circuito abierto y cortocircuito, al pulsarlos, encienden el simulador de luz solar y fijan un valor de resistencia en la carga dinámica de circuito abierto o cortocircuito respectivamente. De esta forma, el operario podrá tomar los valores de tensión

de circuito abierto y la corriente de cortocircuito instantáneamente y sin necesidad de caracterizar el panel. Seguido de ellos, encontramos el botón de apagar que permite apagar el foco del simulador después de haberlo encendido.

Más a la derecha, nos encontramos con un desplegable que nos permitirá elegir el método de caracterización que deseamos utilizar. Hay dos modos de caracterización:

- **VGS Mode:** en este método se realiza un barrido de tensión sobre el surtidor del MOSFET, o lo que es lo mismo, un barrido de resistencias recorriendo los valores de corriente para el rango que ha introducido el usuario. Se trata del mismo método que el utilizado en el apartado anterior.
- **Resistance Mode:** en este método el usuario introduce una lista de resistencias que desea colocar en serie con el panel solar. Sabiendo la resistencia a fijar, el sistema aplica el voltaje en el surtidor del MOSFET de forma que la resistencia equivalente entre los bornes del panel sea la pedida por el usuario. La finalidad de este modo de caracterización es que el usuario tenga la posibilidad de elegir las resistencias que desea colocar en serie con el panel y que la medida para cada una de ellas sea a la misma temperatura. En el Anexo IV se muestra un diagrama de flujo del método a seguir para utilizar este tipo de caracterización.

Puesto que en ambos métodos de caracterización la aplicación necesita que el usuario le introduzca parámetros de entrada, se ha optado por crear dos plantillas con formato Excel donde el usuario introduzca estos parámetros y posteriormente pueda leer el fichero desde la aplicación. En el caso de utilizar “*VGS Mode*”, el usuario ha de introducir el número de referencia del panel solar, la corriente máxima y mínima que desea solicitarle y, por último, el número de muestras que desea tomar para la curva I-V. Sin embargo, si selecciona el método “*Resistance Mode*”, el usuario debe introducir la lista y el número total de resistencias que desea colocar en serie con el panel, así como el número de referencia del panel solar. En las figuras 28 y 29 se muestra un ejemplo de las plantillas.

	A	B	C
1	VGS Characterization Mode		
2			
3	Reference Number:		
4	IMAX (A):		$\leq 2$
5	IMIN (A):		$\geq 0$
6	Samples:		

Figura 28. Plantilla para el método de caracterización VGS Mode

	A	B	C
1	Resistance Characterization Mode		
2			
3	Reference Number:		
4	Resistances Values:		
5	Number of Resistances:		
6			

Figura 29. Plantilla para el método de caracterización Resistance Mode

Una vez se ha seleccionado el método de caracterización, pulsaremos el botón “Comenzar Test”. Esto abrirá una nueva pestaña que nos permitirá seleccionar la plantilla con los parámetros del panel, que habrá configurado previamente el operario. Cabe destacar que el formato del archivo seleccionado ha de ser exclusivamente *.xls*, en caso contrario el programa lanzará un mensaje de error al ser incapaz de abrirlo.

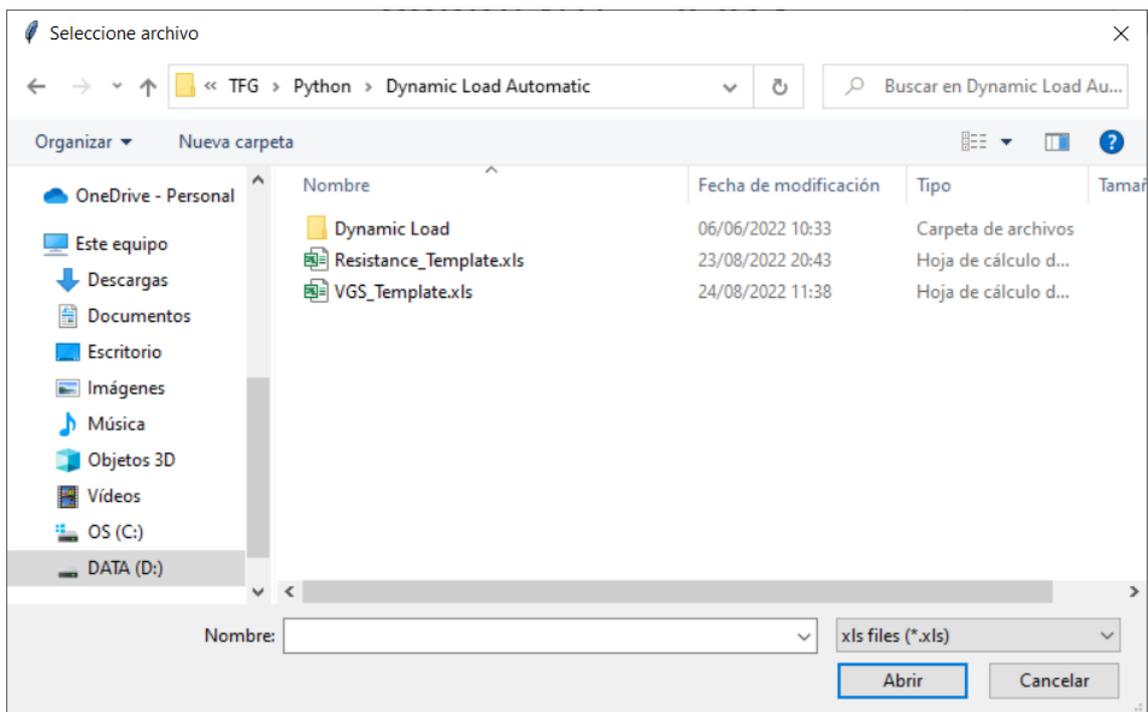


Figura 30. Ventana emergente de selección de plantilla

Por último, el programa se encargará de leer los parámetros del fichero y de realizar la caracterización del panel solar. Al igual que en el programa del apartado anterior, la aplicación almacena los datos de voltaje, corriente, potencia, temperatura y tensión del fotodiodo para cada una de las muestras que toma en un archivo con formato csv. Una vez terminada la caracterización el programa guarda este archivo con todos los datos y también una imagen con la curva característica I-V del panel. Estos dos archivos se guardarán en el mismo directorio desde el cual se está ejecutando el programa.

Además, una vez se ha generado la imagen con la curva, se mostrará el resultado de esta misma en la parte inferior de la interfaz, permitiendo al usuario ver la curva desde la aplicación y, también, realizar nuevas medidas que se sobrepondrán a la anterior para así poder comprobar diferentes situaciones.

## 4. IMPLEMENTACIÓN Y PUESTA EN MARCHA

### 4.1.1. Diseño y montaje de la PCB

Una vez tenemos diseñadas todas las etapas de nuestro sistema, diseñaremos una PCB que contenga la electrónica. Para el diseño se ha utilizado el software Circuit Studio cuya licencia ha sido proveída por EMXYS.

Antes de comenzar con el esquemático, haremos una lista de todos los componentes que necesitamos y comprobaremos cuáles de ellos se encuentran en el stock de EMXYS y cuáles hay que comprarlos. Esta lista se muestra en el Anexo I.

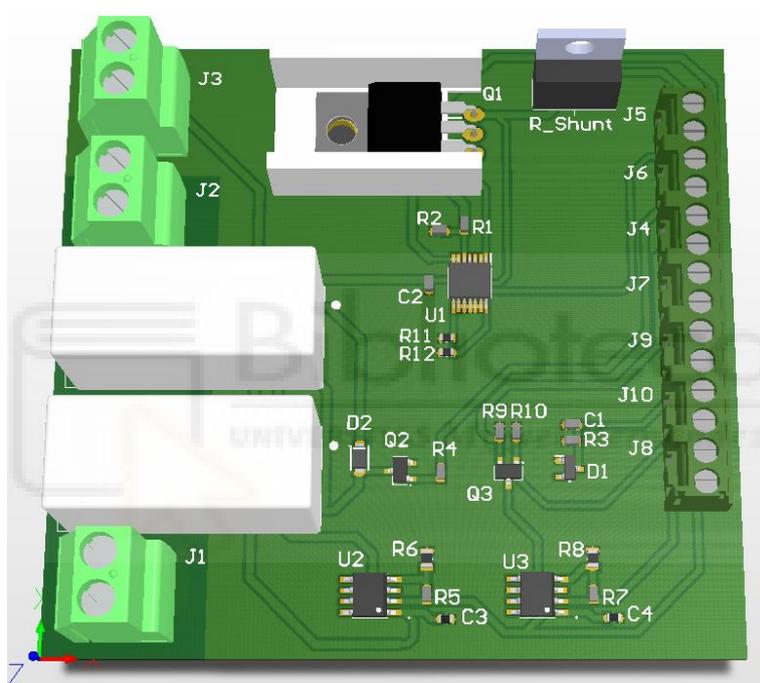
Una vez se dispone de todos los componentes o se han pedido a los proveedores, comenzaremos generando una librería que contenga el esquemático y la huella de cada uno de ellos y montaremos el esquemático completo de la PCB (Anexo II).

Con el esquemático finalizado pasaremos al diseño del layout de nuestra PCB. En cuanto a sus dimensiones, trataremos de hacerla lo más pequeña posible para facilitar su integración. Hay que poner especial atención en la altura de esta ya que es la mayor limitación que tenemos ya que la ranura donde se colocará el sistema nos permite una altura de 5cm. Teniendo todo esto en cuenta el layout de la PCB se ha definido de 80x75cm, mientras que su altura máxima vendrá dada por el componente más alto de todos. En nuestro caso este componente se trata del conector de la alimentación de 12V y de las fases y neutros de los relés, con una suma total de 2cm.

Puesto que la PCB no tiene un número muy elevado de componentes y la gran mayoría de ellos son de montaje superficial, se ha optado por colocar todos ellos sobre su cara superior facilitando el posterior proceso de soldadura y reduciendo la altura que tendrá la PCB. A la hora de colocar los componentes se han ido posicionando por etapas, es decir, por un lado, tendremos el circuito de carga dinámica, por otro los acondicionadores y reguladores de tensión y, de forma aislada, los relés con la parte de alterna. Con la finalidad de evitar pérdidas de potencia y sobrecalentamiento en las señales de nuestro circuito, las pistas

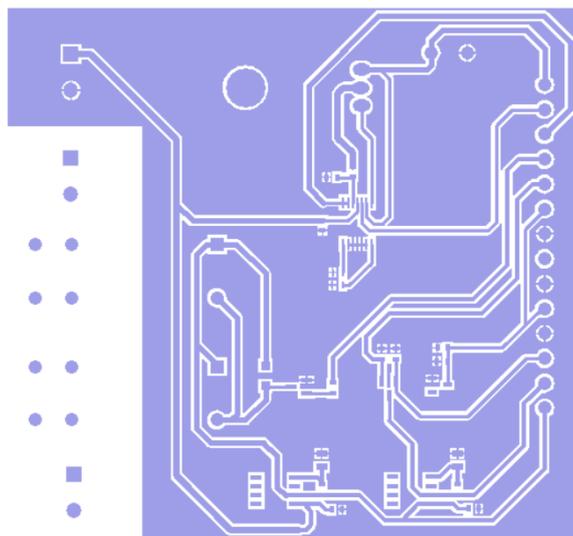
se han creado con el mayor ancho posible especialmente para aquellas por las que fluye gran cantidad de potencia. Para la generación de los planos de masa, tanto en la cara superior como en la inferior, se ha aislado por completo la parte de alterna de la de continua.

Cuando el diseño del layout está completo, gracias al visor 3D de Circuit Studio podemos obtener una vista preliminar de cómo será el resultado de nuestra placa una vez montada con todos los componentes.

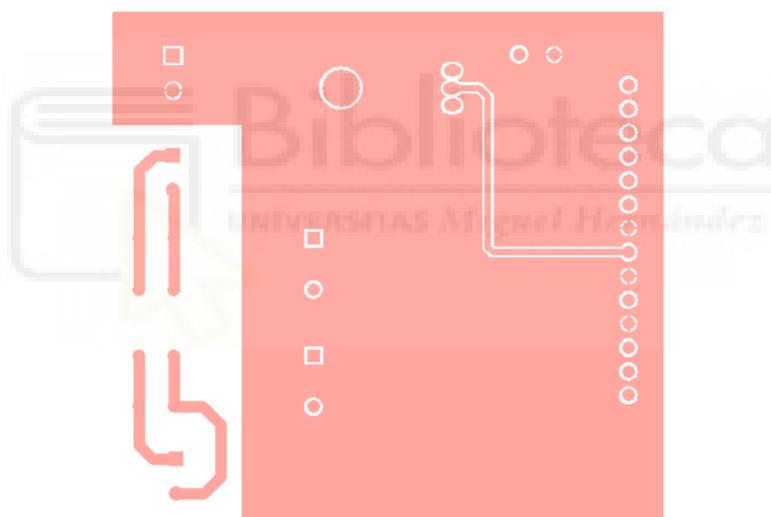


*Figura 31. Visión 3D de la PCB diseñada en Circuit Studio*

No obstante, el plano superior e inferior de la placa se muestran en las siguientes imágenes:



*Figura 32. Layer de la cara superior de la PCB.*



*Figura 33. Layer de la cara inferior de la PCB*

Con el diseño del layout concluido el siguiente paso es generar los gerbers necesarios para enviárselos al fabricante de la placa. Las Figuras 34 y 35 muestran los resultados de la placa ya fabricada.

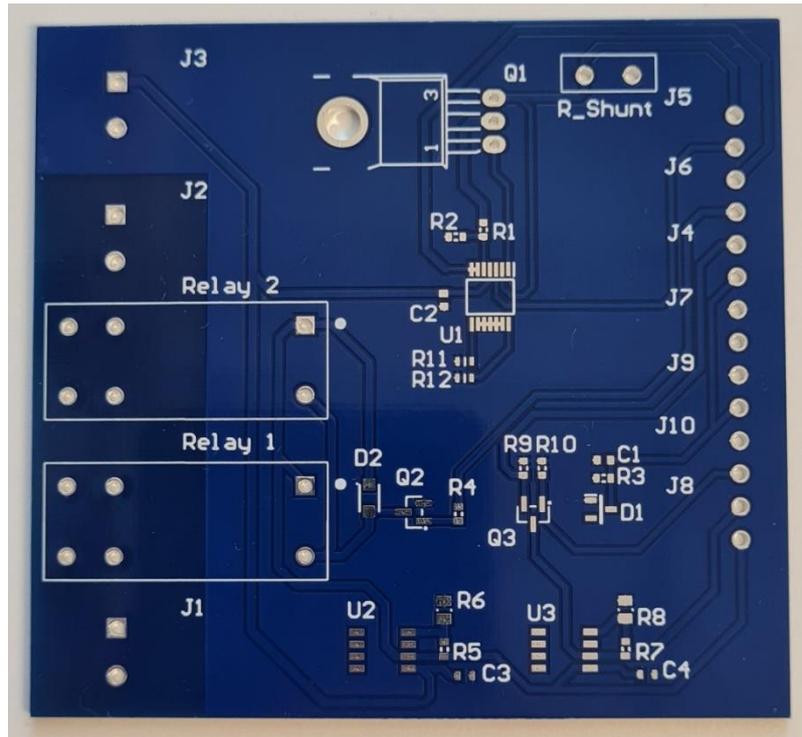


Figura 34. Cara superior de la PCB fabricada

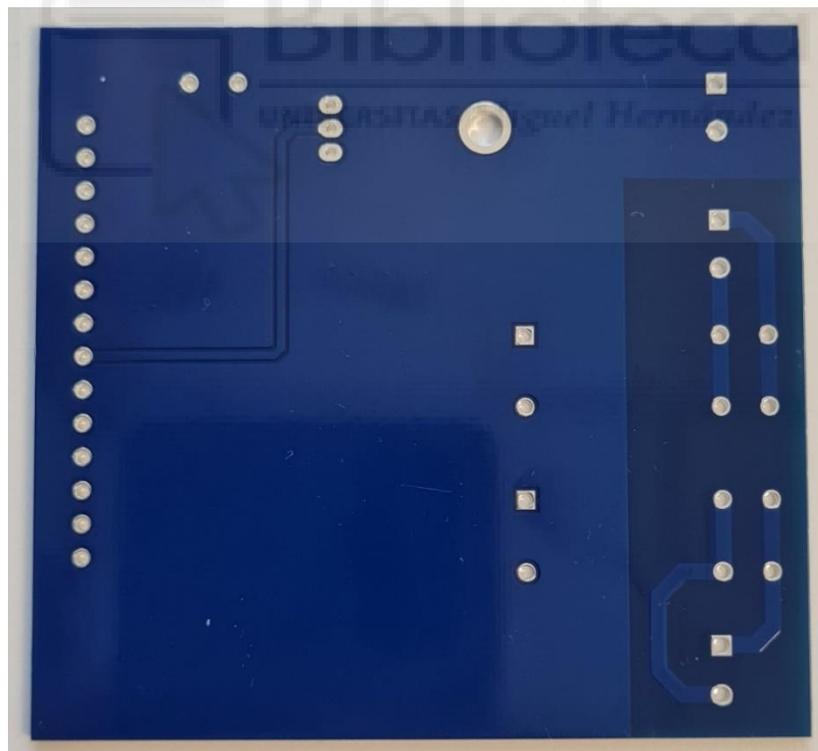


Figura 35. Cara inferior de la PCB fabricada

#### 4.1.2. Ensamblaje de la PCB

Para el montaje de la placa, se ha optado por realizar todas las soldaduras de forma manual, tanto las de orificio pasante como las superficiales. Para comprobar el funcionamiento de cada etapa del diseño, se irá soldando cada una de ellas a la vez que se le realizarán las pruebas de funcionamiento para comprobar que su ensamblaje y funcionamiento es el correcto. Antes de montar ninguna etapa, soldaremos todos los conectores para tener acceso a las señales de alimentación y datos.

##### 4.1.2.1. Reguladores de tensión

Comenzaremos por los reguladores de tensión los cuales, una vez montados y alimentados, deberán proporcionar tensiones de 6 y 3.3V. En la siguiente tabla se muestra su funcionamiento:

Salida	Valor esperado (V)	Valor medido (V)	PASS/FAIL
+6V	6.00 ± 0.1	6.02	PASS
+3V3	3.30 ± 0.1	3.34	PASS

*Tabla 3. Test de funcionamiento de los reguladores de tensión*

##### 4.1.2.2. Sensor de temperatura

El siguiente circuito que montaremos es el que se encarga de la medida del sensor de temperatura. Para comprobar que el circuito acondicionador funciona bien, utilizaremos un sensor AD590 alimentado con los 6V de la placa, mediremos la temperatura de este sensor con una sonda externa de referencia y, con la relación de temperatura a voltaje comentada en el apartado 3.2.2 comprobaremos si el voltaje es el esperado para la temperatura de referencia.

Temp. Referencia (°C)	Valor esperado (V)	Rango de valor esperado (V)	Valor medido (V)	PASS/FAIL
25.2	0.743	± 0.01	0.740	PASS

Tabla 4. Test de funcionamiento del sensor de temperatura

#### 4.1.2.3. Circuito de control

Seguimos ensamblando los relés y el sistema de control de encendido y apagado del simulador de luz solar. Para comprobar que funciona correctamente, generaremos una señal digital que hará cambiar de estado a los relés y, por tanto, habrá continuidad entre sus patillas. Cuando la señal vuelva a estado bajo, los terminales del relé estarán abiertos.

TTL (V)	Pin	Pin	Continuidad	PASS/FAIL
+5V	PH1	PH1	Cortocircuito	PASS
	PH2	PH2	Cortocircuito	PASS
0V	PH1	PH1	Abierto	PASS
	PH2	PH2	Abierto	PASS

Tabla 5. Test de funcionamiento del relé

#### 4.1.2.4. Acondicionamiento del voltaje del panel solar

El siguiente circuito que probaremos de forma aislada es el que se encarga de reducir la tensión del panel solar, donde comprobaremos que el factor de reducción es el correcto.

Tensión del panel solar (V)	Valor esperado (V)	Valor medido (V)	PASS/FAIL
15.0	5.0 ± 0.1	5.05	PASS

Tabla 6. Test de funcionamiento del divisor de tensión

4.1.2.5. Circuito de acondicionamiento del fotodiodo

La última etapa que montaremos para comprobar su funcionamiento es el circuito que extrae una tensión cuando el fotodiodo recibe luz solar.

Señal	Simulador de luz solar (Stado)	Valor esperado (V)	Valor medido (V)	PASS/FAIL
Vout	OFF	$0.00 \pm 0.1$	0.00	PASS
	ON	$2.70 \pm 0.1$	2.66	PASS

*Tabla 7. Test de funcionamiento del fotodiodo*

Por último, soldaremos el circuito que funciona como la carga variable, es decir, el amplificador operacional, el MOSFET con su disipador de calor y la resistencia de SHUNT. Su funcionamiento será comprobado una vez se realice la primera caracterización de un panel solar.

En la Figura 36 se muestra el resultado final de la placa con todos sus componentes ensamblados.

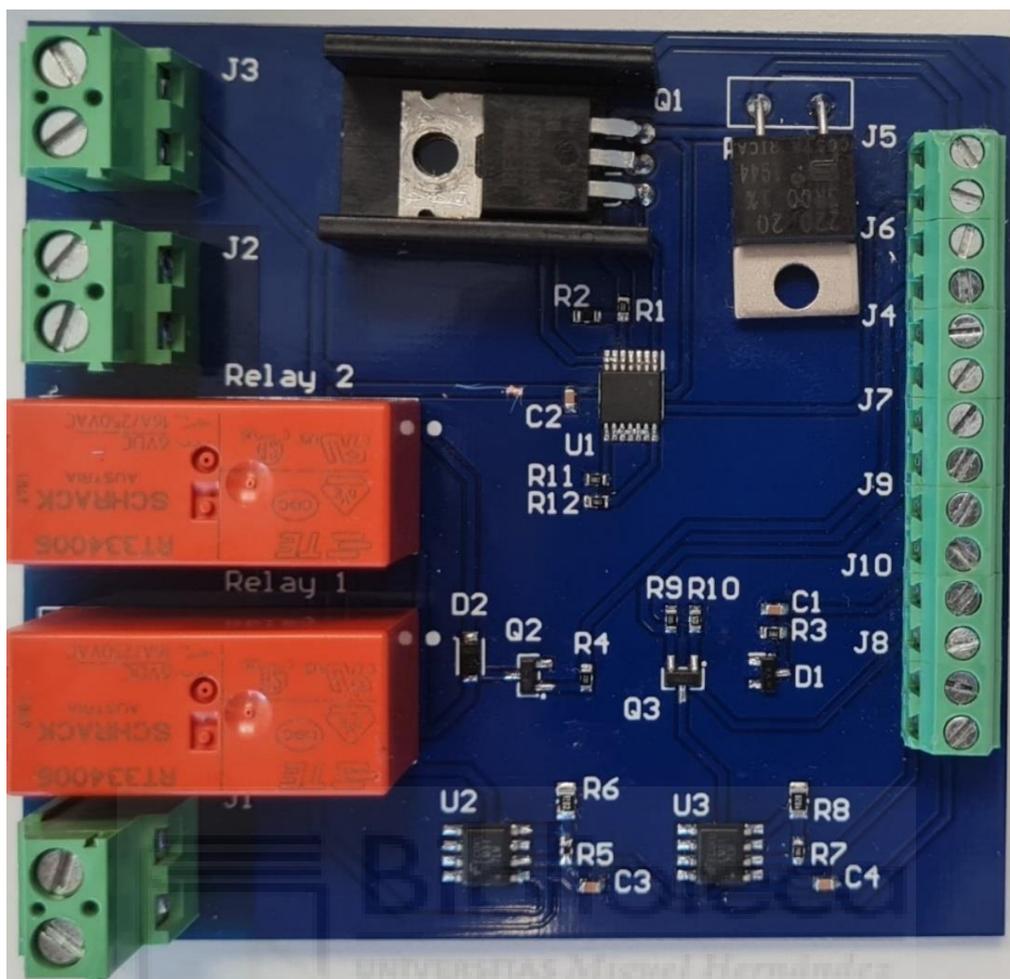


Figura 36. PCB montada

#### 4.1.3. Integración del sistema

Uno de los objetivos marcados al comienzo del proyecto era tener todo el sistema (carga dinámica, adquisición de datos y simulador de luz solar) en un mismo elemento. Por lo tanto, el último paso antes de pasar al diseño software, consiste en la integración del dispositivo diseñado en el simulador de luz solar.

Se utilizará la ranura que hay debajo del simulador de luz solar con espacio vacío (5 cm aproximadamente) para colocar el sistema de adquisición de datos, la carga dinámica y la fuente de alimentación al conjunto. Para facilitar la integración de estos 3 elementos, se diseñará un mecanizado en forma de caja mediante impresión 3D donde fijaremos cada uno de los sistemas y realizaremos las conexiones pertinentes.

A la hora de diseñar el soporte mecánico debemos tener en cuenta la limitación de altura mencionada anteriormente y la necesidad de añadir ranuras a los laterales del soporte para poder conectar la fuente de alimentación, la PCB y el sistema de adquisición de datos con el exterior. Además, al mecanizado tendremos que añadirle un conector que permita interconectar las señales del panel solar con nuestro sistema. Para ello utilizaremos un conector D-Sub de 9 posiciones en uno de sus laterales. Los planos realizados para el diseño del soporte mecánico se muestran en el Anexo III.

El resultado de la base del mecanizado y su tapa se muestra en las figuras 37 y 38.



*Figura 37. Base del soporte mecánico*



*Figura 38. Tapa del soporte mecánico*

El siguiente paso consiste en fijar los elementos del sistema (PCB, DAQ y fuente de alimentación) al soporte y realizar las conexiones necesarias.

Comenzaremos con las conexiones de la fuente de alimentación. Para alimentarla, se utiliza un enchufe universal y sus cables de fase y neutro, de esta forma conseguiremos encender o apagar el sistema completo conectando o desconectando el enchufe de la red eléctrica. La salida de 12V la conectaremos directamente a la entrada de alimentación de la PCB.

Por otro lado, a la entrada de los relés conectaremos de forma simétrica la fase y el neutro provenientes del interruptor externo del simulador de luz solar, mientras que a la salida de los relés conectaremos la fase y neutro de los focos a encender.

Por último, resta conectar las señales de datos y control entre la tarjeta de adquisición de datos y la PCB y entre el conector D-Sub y la PCB.

Pin (DAQ)	Pin (PCB)	Etiqueta
CH0	J6_1	V_SA
CH1	J5_1	V_Shunt
AGND	J7_1	GND
CH2	J4_2	V_Temp
CH3	J4_1	V_Out
AOUT0	J5_2	V_Set
DO0	J6_2	V_Control
DGND	J9_1	GND

Tabla 8. Conexiones entre el DAQ y la PCB

Pin (D-Sub)	Pin (PCB)	Etiqueta
1	J8_2	+6V
2	J8_2	+6V
3	J8_1	+3V3
4	J9_2	V_Temp
5	J10_2	V_Anode
6	J10_2	-V_Load
7	J10_2	-V_Load
8	J7_2	+V_Load
9	J7_2	+V_Load

Tabla 9. Conexiones entre el conector D-Sub y la PCB

Cabe destacar que el para el conector D-Sub se ha colocado uno hembra en el soporte físico del sistema, por lo tanto, se utilizará un conector D-Sub de las mismas posiciones, pero macho, con las señales externas al sistema de carga dinámica y que vienen del panel solar.

El resultado final del sistema integrado en el mecanizado diseñado se muestra en la siguiente figura.



*Figura 39. Sistema de carga dinámica integrado en el mecanizado*

## 5. ANÁLISIS DE RESULTADOS

Durante el proceso de fabricación y puesta en marcha del sistema, las pruebas de funcionamiento se han realizado sobre los paneles diseñados por EMXYS. Estos paneles tienen las dimensiones correspondientes a una plataforma de cubesat 3U y están formados por la unión de 6 celdas solares en serie.

### 5.1. Resultados experimentales

En este apartado mostraremos los resultados experimentales de la caracterización de uno de los paneles diseñados en EMXYS. El método de caracterización utilizado por los operarios para cada uno de los 4 paneles que forman el cubesat, ha sido "VGS Mode". De esta forma, los resultados mostrados corresponden únicamente a este método de caracterización.

El panel utilizado tiene como referencia número de referencia el 20210416. En la Figura 40 se muestra, en rojo, su curva característica I-V, y en azul su curva de potencia entregada a la carga.

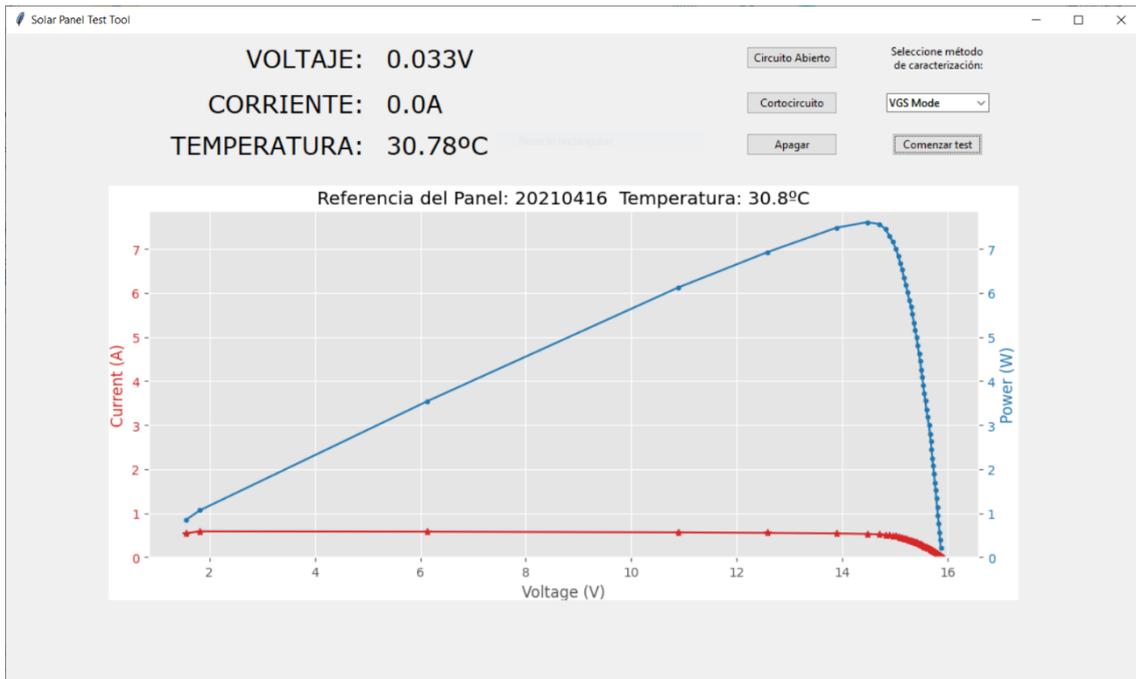


Figura 40. Resultados de la curva característica I-V para el panel 20210416

De los resultados del fichero, por un lado, podemos obtener los valores de voltaje e intensidad de máxima potencia, así como el punto de máxima potencia. Por otro lado, podemos obtener el incremento de la temperatura con la diferencia entra la final y la inicial y, por último, los valores de tensión del fotodiodo para comprobar si funciona correctamente con el voltaje esperado. Estos valores se muestran en la siguiente tabla:

<b><math>V_{OC}</math> (V)</b>	15.886
<b><math>I_{SC}</math> (A)</b>	0.587
<b><math>V_{mp}</math> (V)</b>	14.379
<b><math>I_{mp}</math> (A)</b>	0.538
<b><math>P_{mp}</math> (W)</b>	7.742
<b><math>\Delta_{temp}</math> (°C)</b>	1.3
<b><math>V_{fotodiodo}</math> (V)</b>	2.744

Tabla 10. Propiedades eléctricas del panel solar 20210416

Una vez realizada la misma medida para los 3 paneles restantes que forman el cubesat, se ha podido comprobar como las propiedades eléctricas obtenidas para cada uno de ellos son muy similares entre sí. Esto es un buen indicativo de que los paneles han sido diseñados y están funcionando tal y como se esperaba sin tener parámetros de funcionamiento dispares ya que interesa que todos tengan un comportamiento similar.

Por otro lado, en la tabla superior podemos ver como el incremento de temperatura del panel no es elevado, obteniendo un rendimiento del panel mejor que si lo fuera y, además, la tensión del fotodiodo es la esperada por lo que se extrae que está funcionando correctamente.



## 6. CONCLUSIONES Y LÍNEAS FUTURAS

Una vez diseñado y puesto en marcha el sistema, se ha podido comprobar como hemos cumplido con los objetivos del proyecto, obteniendo como resultado un sistema automático y preciso de caracterización de paneles solares. Gracias a este proyecto, EMXYS podrá tener un proceso de caracterización de los paneles solares en serie reduciendo el tiempo de caracterización de todos ellos y, por tanto, el tiempo que tiene que dedicar un empleado a este proceso. Además, contará con la posibilidad de convertir el sistema en un producto que pueda ser comercializado y, por tanto, obtener ingresos por ello.

Igualmente, el hecho de incluir una interfaz gráfica para el usuario hace que el proceso de caracterización sea mucho más sencillo, rápido e intuitivo, de forma que la persona a cargo de la caracterización de los paneles solares no tenga por qué tener conocimientos sobre programación ni sobre el funcionamiento de los paneles solares ni el sistema diseñado.

Por otro lado, durante el proceso de diseño del sistema, por un lado he podido poner en práctica todos los conocimientos adquiridos a lo largo del Grado, mientras que por otro me ha servido como introducción al mundo laboral gracias a que el proyecto ha sido desarrollado en su totalidad en EMXYS.

En cuanto a las líneas abiertas del trabajo, hablaremos de las mejoras que puede introducir nuestro sistema tanto en la parte hardware como en la parte software. En el primero de los casos algunas mejoras que podríamos introducir son las siguientes:

- Sobredimensionar el sistema para poder caracterizar paneles solares con mayores voltajes y corrientes de funcionamiento. De esta forma, EMXYS tendría mayor libertad a la hora de diseñar paneles y podría, por ejemplo, caracterizar paneles como los que se han mencionado en apartados anteriores, colocados en serie y/o paralelo.
- Diseñar un convertidor de corriente alterna a corriente continua para la alimentación de los 12V que esté embebido en la PCB diseñada, ahorrando espacio y evitando la dependencia de un modelo comercial.

- Diseñar la etapa de adquisición de datos y control partiendo desde cero con el uso de convertidores analógicos a digital y viceversa, así como una interfaz de datos que permita controlar esta etapa mediante un ordenador externo.
- Introducir en el sistema un ordenador embebido, como por ejemplo una Raspberry Pi, que se encargue de realizar lo que hacía hasta ahora el ordenador externo conectado por USB al DAQ. Esta mejora sería interesante ya que tendríamos todo el sistema en un único elemento y, podríamos extraer los datos de caracterización, por ejemplo, utilizando un pen drive conectado a este ordenador.
- Mejorar la estética del entorno gráfico y añadir funcionalidades como por ejemplo mostrar los valores  $V_{sc}$ ,  $I_{oc}$  y  $P_{mp}$  una vez realizada la caracterización.



**BIBLIOGRAFÍA**

- [1] Enciclopedia de Energía, “Célula fotovoltaica”, [En línea]. Disponible: [https://energyeducation.ca/Enciclopedia\\_de\\_Energia/index.php/C%C3%A9lula\\_fotovoltaica](https://energyeducation.ca/Enciclopedia_de_Energia/index.php/C%C3%A9lula_fotovoltaica). [Último acceso: 2022].
- [2] Norma Rosario Flores Rivera y Miguel Ángel Domínguez Ramírez, “Medición de la eficiencia energética de los paneles solares de silicio.”, pp. 16-18.
- [3] “Pico Solar Simulator”, G2V Optics, [En línea]. Disponible: <https://g2voptics.com/products/pico-solar-simulator/>. [Último acceso: 2022].
- [4] “Sunbrick Solar Simulator”, G2V Optics, [En línea]. Disponible: <https://g2voptics.com/products/sunbrick-solar-simulator/>. [Último acceso: 2022].
- [5] “SINUS-300”, WAVELABS, [En línea]. Disponible: <https://wavelabs.de/products/sinus-220/>. [Último acceso: 2022].
- [6] “Hönle UV SOL 500”, Hönle, [En línea]. Disponible: <https://www.hoenle.com/products/solar-simulation-systems-and-light-fixtures/sol-light-fixtures#>. [Último acceso: 2022].
- [7] Vicente García, “El Transistor MOSFET”, 2012, [En línea]. Disponible: <https://www.diarioelectronicohoy.com/blog/el-transistor-mosfet>. [Último acceso: 2022].
- [8] Electronoobs, “DIY Constant Load Controller”, 2020, pp. Schematic, [En línea]. Disponible: <https://electronoobs.io/tutorial/161#>. [Último acceso: 2022].
- [9] Inventable.eu, “Cómo controlar un relé con un transistor”, [En línea]. Disponible: <https://www.inventable.eu/controlar-rele-con-transistor/>. [Último acceso: 2022].
- [10] “TXL 035-12S”, TRACO POWER, [En línea]. Disponible: <https://www.tracopower.com/int/es/model/txl-035-12s>. [Último acceso: 2022].

- [11] Measurement Computing, “Universal Library Help”, [En línea]. Disponible: [https://www.mccdaq.com/PDFs/Manuals/Mcculw\\_WebHelp/ULStart.htm](https://www.mccdaq.com/PDFs/Manuals/Mcculw_WebHelp/ULStart.htm). [Último acceso: 2022].



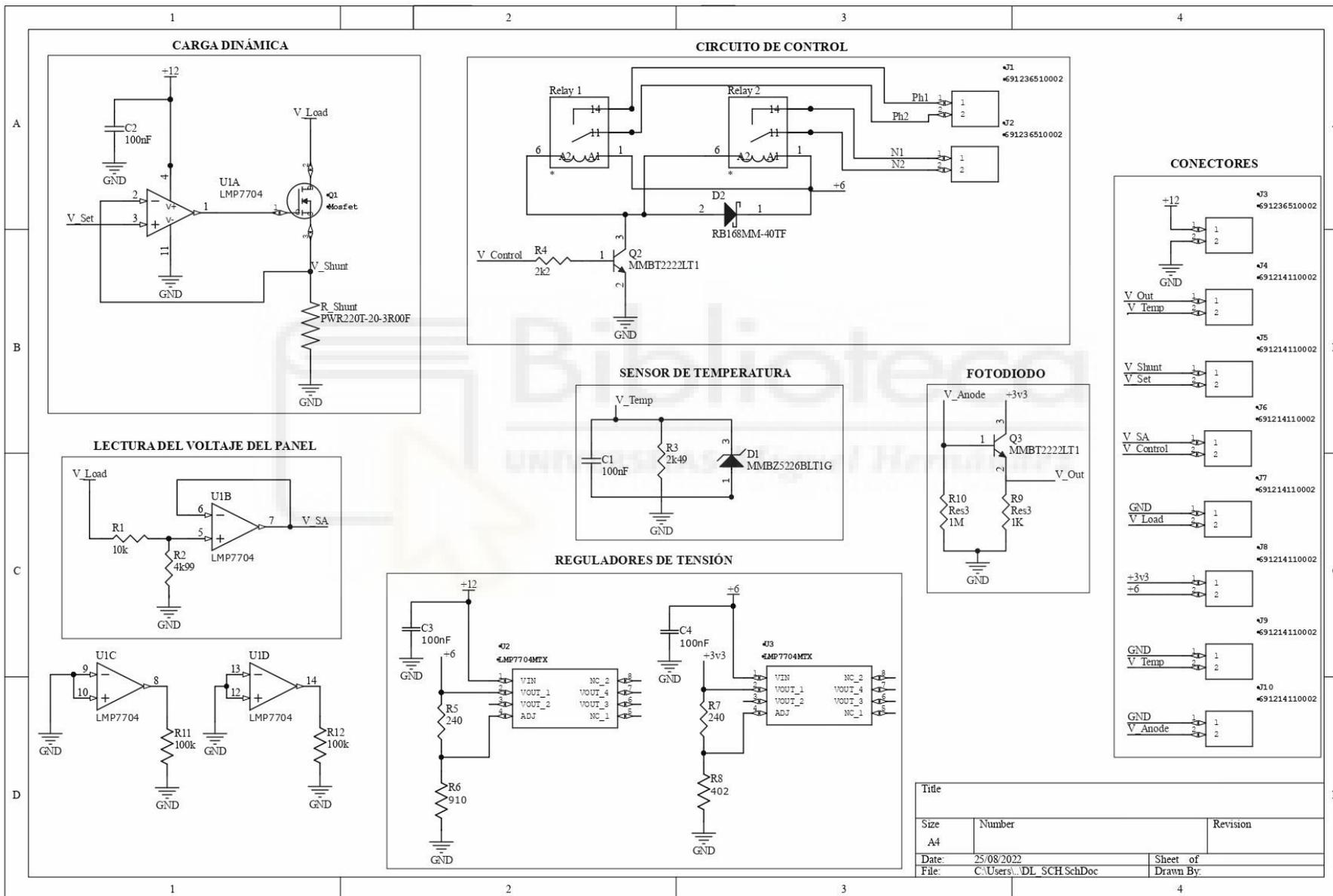
## ANEXOS

### I. Lista de materiales

Nombre	Descripción	Valor	Cantidad	Stock
U1	OP AMP	LMP7704MTX/NOPB	1	Sí
Q1	Mosfet N channel	IRF530PBF	1	Sí
Q2, Q3	BJT	SMMBT2222ALT1G	2	Sí
R1	Resistor 0603	10k	1	Sí
R2	Resistor 0603	4k99	1	Sí
R_Shunt	12W Resistor	PWR220T-20-3R00F	1	Pedido
R3	Resistor 0603	2k49	1	Sí
R4	Resistor 0603	2k2	1	Sí
R5, R7	Resistor 0603	240	2	Sí
R6	Resistor 0805	910	1	Sí
R8	Resistor 0805	402	1	Sí
R9	Resistor 0603	1k	1	Sí
R10	Resistor 0603	1M	1	Sí
R11, R12	Resistor 0603	100k	2	Sí
C1, C2, C3, C4	Capacitor 0603	100nF	4	Sí
D1	Zener 3.3V	MMBZ5226BLT1G	1	Sí
J1, J2, J3	Connector	CTB0709	3	Sí
J4, J5, J6, J7, J8, J9, J10	Connector	6,91214E+11	7	Pedido
U2	Regulator	LM317LMX	2	Sí
D2	Schottky Diode	RB168MM-40TFTR	1	Sí
Relay	Relay	RT334006	2	Pedido
Dissipator	Heat Sink	HSE-B2111-038	1	Pedido

## II. Esquema eléctrico

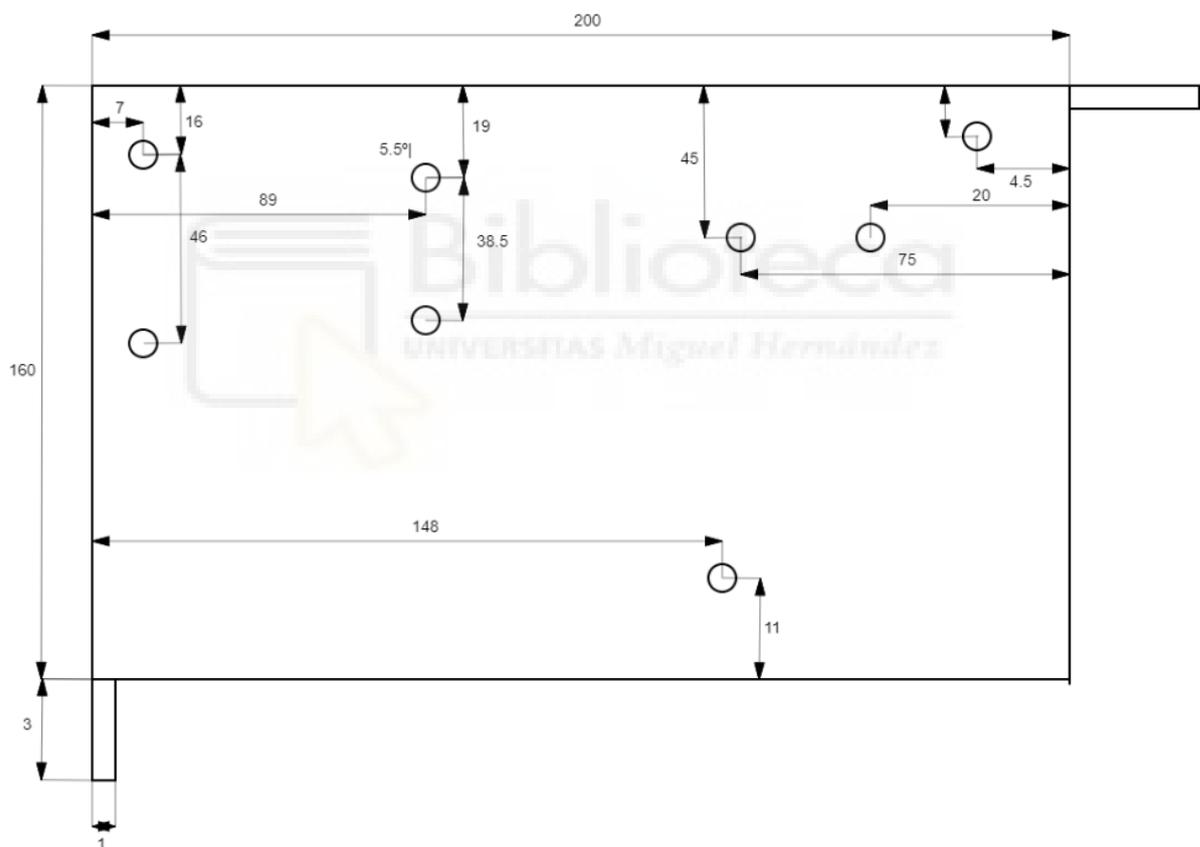




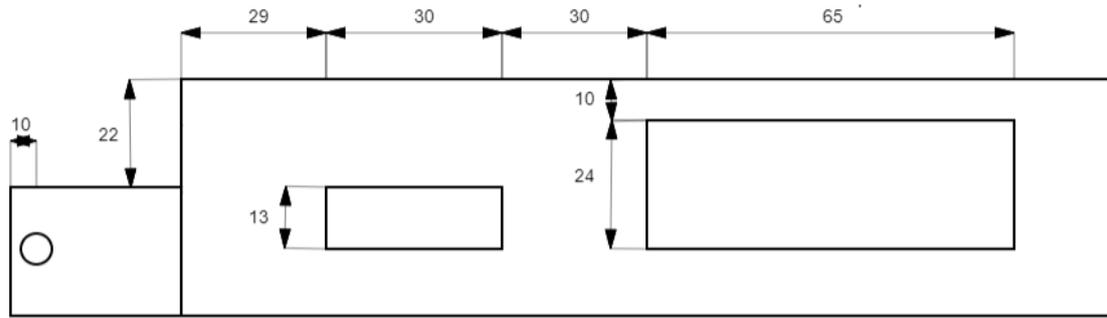
### III. Planos de la envolvente

Antes de mostrar los planos diseñados, cabe destacar que todos los tornillos utilizados son de métrica 3, por lo que el diámetro de todos los agujeros es de 3mm.

Comenzaremos mostrando el plano inferior del mecanizado. En este añadiremos agujeros en las posiciones de los tornillos de los sistemas a fijar en las posiciones deseadas.



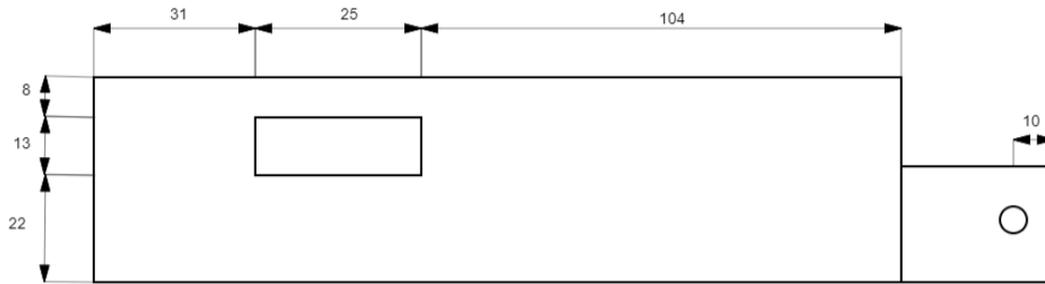
En el siguiente plano nos encontramos con uno de los 2 planos laterales interiores donde tenemos dos ranuras: una para introducir los cables de fase y neutro de los relés para encender y apagar los focos y otra para acceder a la alimentación y la salida de tensión de la fuente de alimentación.



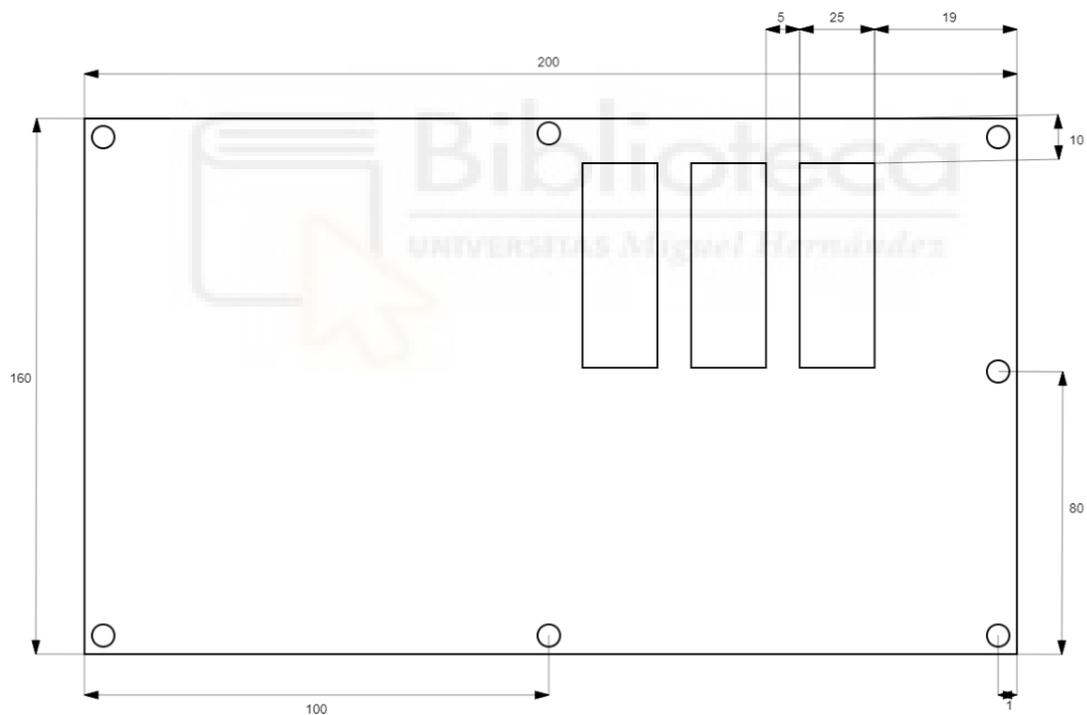
En el siguiente plano lateral interior incluye una ranura con 2 tornillos para conectar el conector D-Sub.



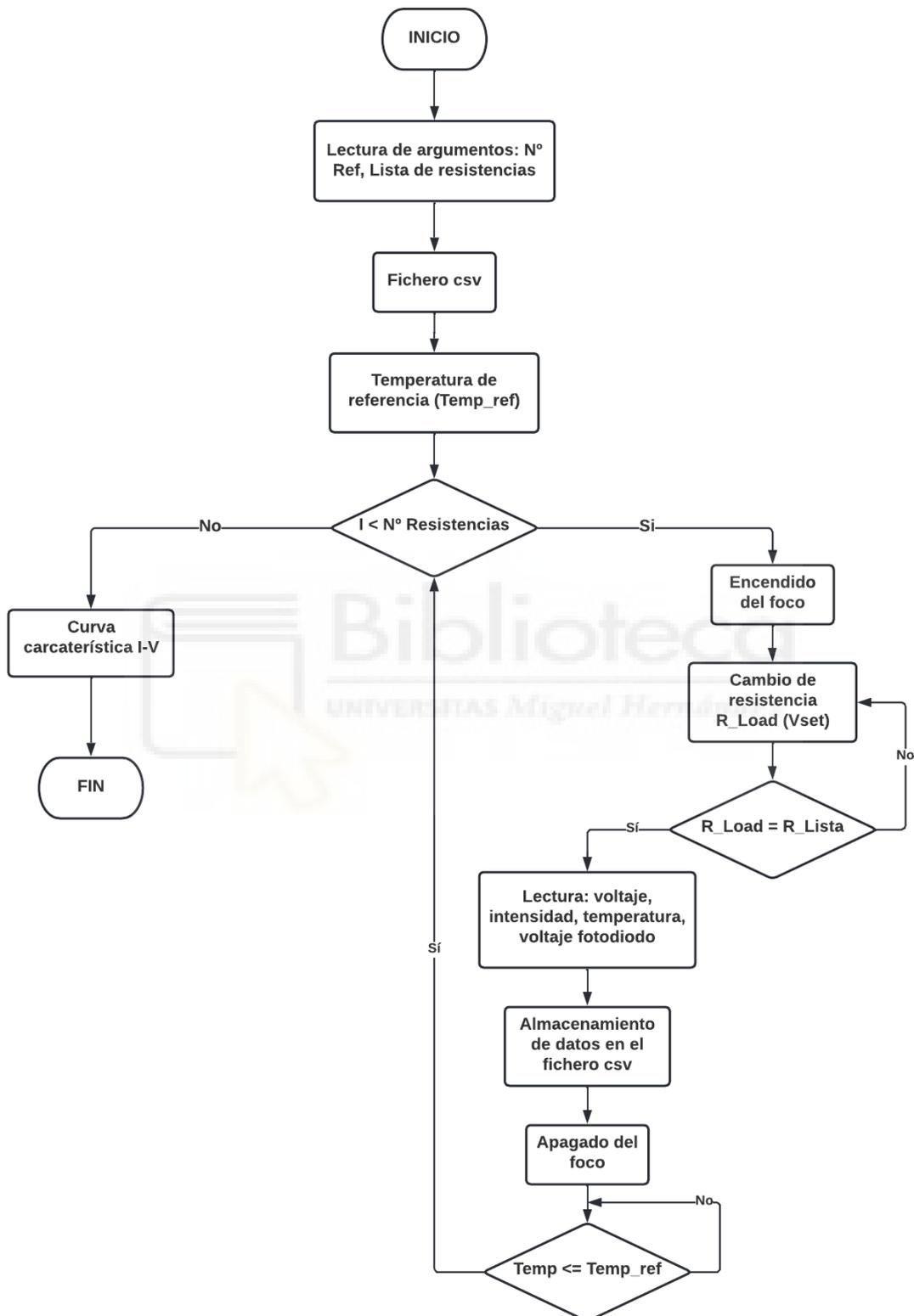
En el tercer lateral, añadimos una ranura donde encontraremos el USB del módulo de adquisición de datos. Este lateral dará al exterior del simulador de luz solar para poner conectar el ordenador con el sistema mediante un cable USB.



Por último, se muestra el plano de la tapa superior del mecanizado. Aquí se ha añadido una ranura para favorecer la ventilación de la fuente de alimentación.



#### IV. Diagrama de flujo para el método de selección de resistencia



## V. Código fuente del módulo principal

```
###--- LIBRERÍAS ---###
from mcculw import ul
from mcculw.enums import ULRange
from mcculw.ul import ULError
from mcculw.enums import DigitalPortType
import time
import numpy as np
from pathlib import Path
import sys
import csv
import matplotlib.pyplot as plt
from os import listdir

plot_data_IV=[]
plot_data_W=[]

###--- RANGOS DE VOLTAJE DEL DAQ ---###
ao_range = ULRange.UNI5VOLTS
voltage_range = ULRange.BIP10VOLTS
current_range = ULRange.BIP10VOLTS

###--- CONSTANTES ---###
board_num = 0
channel = 0
channel_out = 0
voltage_ch = 0
current_ch = 1
temp_ch = 2
diode_ch = 3

imput_mode = 16

r_shunt = 3
r_temp = 2490
multiplier = 3 #-Factor de escala para la lectura del voltaje del panel

###--- FUNCIÓN PARA LECTURAS DE VOLTAJE ---###
def average_voltage(samples, channel):
    voltage = 0
    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, channel, voltage_range))
    avg_voltage = voltage/samples
    return avg_voltage
```

```
###--- FUNCIÓN PARA LECTURAS DE TEMPERATURA ---###
def average_temperature(samples, channel):
    voltage = 0
    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, channel, voltage_range))
    avg_voltage = voltage/samples
    temp = (avg_voltage/r_temp)*1000000-273.15
    return temp

if (len(sys.argv) < 4): #-Comprobación del número de argumentos
    print('Número de parámetros incorrecto')
    print('Introduzca: [NºRef] [Imax] [Imin] [Nºmuestras]')
    quit()

while len(ul.get_daq_device_inventory(1, number_of_devices=10)) < 1: #-
    Comprobación de la conexión del DAQ
        print('Error, conecte la tarjeta de adquisición de datos')
        time.sleep(1)

###--- ALMACENAMIENTO DE LOS PARÁMETROS DE CORRIENTE Y OBTENCIÓN DEL RANGO DE
MEDIDA ---###
panel_ref = sys.argv[1]
i_max = float(sys.argv[2])
i_min = float(sys.argv[3])
samples = int(sys.argv[4])

if i_max < i_min or i_max > 2 or i_min < 0: #-Comprobación de la validez de
los parámetros de corriente
    print('Parámetros de corriente inválidos')
    quit()

v_max = i_max * r_shunt
v_min = i_min * r_shunt
increment = (v_max - v_min)/samples

###--- CONFIGURACIÓN DEL PUERTO DIGITAL DEL DAQ ---###
port_type = DigitalPortType.AUXPORT
ul.d_config_port(board_num, port_type, 1)
ul.d_out(board_num, port_type, 0)
time.sleep(3)

###--- GENERACIÓN DEL ARCHIVO CSV ---###
date = str("{0:02d}").format(time.localtime().tm_mday)
date += str("{0:02d}").format(time.localtime().tm_mon)
date += str("{0:04d}").format(time.localtime().tm_year)
```

```

t = str("{0:02d}{1:02d}{2:02d}").format(time.localtime().tm_hour,
time.localtime().tm_min, time.localtime().tm_sec)

file_name = str('I-V_Measures'+ '_' + date + '_' + t + '.csv')
print(file_name)

header = ('Voltaje;Corriente;Potencia;Temperatura;Fotodiodo')

with open(file_name, 'a+') as fs:
    fs.write('Nº Ref;' + str(panel_ref) + ';\n')
    fs.write(';\n')
    fs.write(str(header) + ';\n')

time.sleep(1)

temp1 = average_temperature(50, temp_ch)
ul.d_out(board_num, port_type, 1) #-ENCENDIDO DEL FOCO
time.sleep(0.25)

###--- BUCLE DE ADQUISICIÓN Y ALMACENAMIENTO DE DATOS ---###
z = v_max
for i in range(samples):
    ul.v_out(board_num, channel_out, ao_range, z) #-Modificación de la
resistencia que ve el panel solar
    z = z - increment

    time.sleep(0.001)

    voltage = average_voltage(500, voltage_ch)*multiplier + 0.6 #-Voltaje
del panel
    current = average_voltage(500, current_ch)/r_shunt #-Corriente del panel
    temp = average_temperature(100, temp_ch) #-Temperatura del panel
    photo_diode = average_voltage(100, diode_ch) #-Tensión del fotodiodo
    power = voltage * current #-Potencia del panel

    print('Voltaje: ' + str(np.round(voltage,3)))
    print('Corriente: ' + str(np.round(current,3)))
    print('Potencia: ' + str(np.round(power,3)) + '\n')

    data_folder = Path('./')
    file_path = data_folder / file_name

    line = str(np.round(voltage,3)) + ';' + str(np.round(current,3)) + ';' +
str(np.round(power,3)) + ';' + str(np.round(temp,1)) + ';' +
str(np.round(photo_diode,3))
    with open(file_name, 'a+') as fs:
        fs.write(str(line) + ';\n') #-Almacenamiento de los datos en el fichero
csv

```

```
voltage_avg = round(voltage,3)
current_avg = round(current,3)
power_avg = round(voltage*current,3)

pt_IV=[voltage_avg, current_avg]
pt_W=[voltage_avg, power]

plot_data_IV.append(pt_IV)
plot_data_W.append(pt_W)

temp2 = average_temperature(50, temp_ch)

time.sleep(0.5)

ul.v_out(board_num, channel_out, ao_range, 0)
ul.d_out(board_num, port_type, 0) #-Apagado del foco

print('Temperatura inicial:' + str(temp1))
print('Temperatura final:' + str(temp2))
print('Aumento de temperatura de:' + str(temp2 - temp1))

plot_matrixIV = np.array(plot_data_IV)
plot_matrixIV = np.transpose(plot_matrixIV)

plot_matrixW = np.array(plot_data_W)
plot_matrixW = np.transpose(plot_matrixW)

###---GRÁFICA CON LA CURVA I-V---###
plt.style.use('ggplot')
fig = plt.figure(figsize=(13,6))

ax1 = fig.add_subplot(111)
color = 'tab:red'
ax1.set_title('Panel: ' + str(panel_ref) + ' Temperatura: ' +
str(np.round(temp1,1)) + '°C')
ax1.set_xlabel('Voltage (V)')
ax1.set_ylabel('Current (A)', color=color)
ax1.tick_params(axis='y', labelcolor=color)
plt.plot(plot_matrixIV[0], plot_matrixIV[1],color=color,label='I-
V',marker="*")
plt.grid()

ax2 = ax1.twinx()
color = 'tab:blue'
ax2.set_ylabel('Power (W)', color=color)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0)
```

```
ax1.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0)
ax1.grid()
plt.plot(plot_matrixW[0],
plot_matrixW[1],color=color,label='Power',marker=".")

fig.tight_layout()

fig.show()
fig.set_size_inches(10, 4.6)

plt.savefig("figure_output.png")
```



## VI. Código fuente del módulo gráfico

```
###--- LIBRERÍAS ---###
from mcculw import ul
from mcculw.enums import ULRange
from mcculw.ul import ULError
from mcculw.enums import DigitalPortType

import time
import numpy as np
import xlrd
from pathlib import Path
import sys
import csv
import matplotlib.pyplot as plt
from os import listdir
from pylab import *
from PIL import ImageTk, Image
from random import randint

from tkinter import *
from tkinter.ttk import *
from tkinter import filedialog

plot_data_IV=[]
plot_data_W=[]

###--- RANGOS DE VOLTAJE DEL DAQ ---###
ao_range = ULRange.UNI5VOLTS
voltage_range = ULRange.BIP10VOLTS
current_range = ULRange.BIP10VOLTS
port_type = DigitalPortType.AUXPORT

###--- CONSTANTES ---###
board_num = 0
channel = 0
channel_out = 0
voltage_ch = 0
current_ch = 1
temp_ch = 2
diode_ch = 3

imput_mode = 16

r_shunt = 3
r_temp = 2490
```

```

multiplier = 3 #-Factor de escala para la lectura del voltaje del panel

###--- FUNCIÓN PARA LECTURAS DE VOLTAJE ---###
def average_voltage(samples, channel):
    voltage = 0
    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, channel, voltage_range))
    avg_voltage = voltage/samples
    return avg_voltage

###--- FUNCIÓN PARA LECTURAS DE TEMPERATURA ---###
def average_temperature(samples, channel):
    voltage = 0
    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, channel, voltage_range))
    avg_voltage = voltage/samples
    temp = (avg_voltage/r_temp)*1000000-273.15
    return temp

###--- LECTURA DE LA CORRIENTE EN TIEMPO REAL ---###
def current_read():
    voltage = 0
    samples = 100

    while len(ul.get_daq_device_inventory(1, number_of_devices=10)) < 1:
        print('Error, conecte la tarjeta de adquisicion de datos')

    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, current_ch, voltage_range))
    avg_voltage = voltage/samples

    current = (avg_voltage/r_shunt)
    if current >= 0 :
        text_label = str(np.round(current, 3)) + 'A'
        currentLabel['text']= text_label
    else:
        text_label = str(np.round(0.0, 3)) + 'A'
        currentLabel['text']= text_label

    root.after(500, current_read)

###--- LECTURA DE LA TEMPERATURA EN TIEMPO REAL ---###
def temperature_read():
    voltage = 0
    samples = 100

    while len(ul.get_daq_device_inventory(1, number_of_devices=10)) < 1:

```

```

    print('Error, conecte la tarjeta de adquisicion de datos')

for i in range(samples):
    voltage = voltage + (ul.v_in(board_num, temp_ch, voltage_range))
avg_voltage = voltage/samples

temp = (avg_voltage/r_temp)*1000000-273.15
text_label = str(np.round(temp, 2)) + '°C'
templabel['text']= text_label

root.after(500, temperature_read)

###--- LECTURA DEL VOLTAJE EN TIEMPO REAL ---###
def voltage_read():
    voltage = 0
    samples = 100

    while len(ul.get_daq_device_inventory(1, number_of_devices=10)) < 1:
        print('Error, conecte la tarjeta de adquisicion de datos')

    for i in range(samples):
        voltage = voltage + (ul.v_in(board_num, voltage_ch, voltage_range))

    avg_voltage = (voltage/samples)*multiplier

    text_label = str(np.round(avg_voltage, 3)) + 'V'
    voltslabel['text']= text_label

    root.after(500, voltage_read)

###--- BOTÓN CIRCUITO ABIERTO ---###
def opc_command():
    ul.v_out(board_num, channel_out, ao_range, 0)
    time.sleep(0.05)
    ul.d_config_port(board_num, port_type, 1)
    ul.d_out(board_num, port_type, 1)

###--- BOTÓN CORTOCIRCUITO ---###
def shc_command():
    ul.v_out(board_num, channel_out, ao_range, 6)
    time.sleep(0.05)
    ul.d_config_port(board_num, port_type, 1)
    ul.d_out(board_num, port_type, 1)

###--- BOTÓN APAGADO ---###
def offcommand():
    ul.d_config_port(board_num, port_type, 1)
    ul.d_out(board_num, port_type, 0)

```

```
###--- VENTANA PARA ABRIR FICHERO EXCEL ---###
def open_file():
    name_file=filedialog.askopenfilename(title = "Seleccione
archivo",filetypes = (("xls files","*.xls"),("todos los archivos","*..*")))
    return name_file

###--- BOTÓN PARA LANZAR EL TEST DE CARACTERIZACIÓN ---###
def launchcommand():

    if combo.current() == -1:
        print('Seleccione un metodo...')
        errorLabel['text']= "ERROR: Seleccione un método..."

    elif combo.current() == 0:
        errorLabel['text']= ""
        template_file = open_file()

        csv_read = xlrd.open_workbook(template_file)
        sheet = csv_read.sheet_by_index(0)
        panel_ref = int (sheet.cell_value(2,1))
        i_max = float(sheet.cell_value(3,1))
        i_min = float(sheet.cell_value(4,1))
        samples = int(sheet.cell_value(5,1))

        print('Referencia del panel:' + str(sheet.cell_value(2,1)))
        print('Valor max:' + str(sheet.cell_value(3,1)))
        print('Valor min:' + str(sheet.cell_value(4,1)))
        print('Samples:' + str(sheet.cell_value(5,1)))

        if float(i_max) > float(i_min) and float(i_max) <= 2 and float(i_min)
>= 0:
            errorLabel['text']= ""
            vgs_test(panel_ref, i_max, i_min, samples)

        else:
            print('ERROR: Los parámetros de corriente son incorrectos')
            errorLabel['text']= "ERROR: Los parámetros de corriente son
incorrectos"

    elif combo.current() == 1:
        errorLabel['text']= ""
        template_file = open_file()

        csv_read = xlrd.open_workbook(template_file)
        sheet = csv_read.sheet_by_index(0)
        n_ref = int(sheet.cell_value(2,1))
        n_res = int(sheet.cell_value(4,1))
        print(n_res)
```

```

r_list=[]
for i in range (n_res):
    # r_list[i] = int(sheet.cell_value(3,i+1))
    r_list.append(int(sheet.cell_value(3,i+1)))

resistance_test(n_ref, r_list)

###--- COMPROBACIÓN DE TEMPERATURA DE REFERENCIA ---###
def wait_for_temp(temperature_reference):
    aux_temp = average_temperature(100, 2)
    print('Temperatura actual: ' + str(aux_temp))
    print('Diferencia de temperatura: ' + str(aux_temp-temperature_reference))
    time.sleep(1)
    while aux_temp > temperature_reference:
        aux_temp = average_temperature(100, 2)
        print('Diferencia de temperatura: ' + str(aux_temp-
temperature_reference))
        time.sleep(1)

###--- MÉTODO DE CARACTERIZACIÓN VGS ---###
def vgs_test(n_reference, i_max, i_min, samples):
    date = str("{0:02d}").format(time.localtime().tm_mday)
    date += str("{0:02d}").format(time.localtime().tm_mon)
    date += str("{0:04d}").format(time.localtime().tm_year)

    t = str("{0:02d}{1:02d}{2:02d}").format(time.localtime().tm_hour,
time.localtime().tm_min, time.localtime().tm_sec)

    file_name = str('I-V_Measures'+ '_' + date + '_' + t + '.csv')
    print(file_name)

    header = ('Voltaje;Corriente;Potencia;Temperatura;Fotodiodo')

    with open(file_name, 'a+') as fs:
        fs.write('Nº Ref;' + str(n_reference) + ';\n')
        fs.write(';\n')
        fs.write(str(header) + ';\n')

    time.sleep(3)

    v_max = i_max * r_shunt
    v_min = i_min * r_shunt

    increment = (v_max - v_min)/samples

    temp1 = average_temperature(50, temp_ch)
    port_type = DigitalPortType.AUXPORT
    ul.d_config_port(board_num, port_type, 1)
    ul.d_out(board_num, port_type, 1)

```

```
time.sleep(0.25)

z = v_max

for i in range(samples):
    ul.v_out(board_num, channel_out, ao_range, z)
    z = z - increment

    time.sleep(0.001)

    voltage = average_voltage(200, voltage_ch)*multiplier
    current = average_voltage(200, current_ch)/r_shunt
    temp = average_temperature(50, temp_ch)
    photo_diode = average_voltage(50, diode_ch)
    power = voltage * current

    print('Voltaje: ' + str(np.round(voltage,3)))
    print('Corriente: ' + str(np.round(current,3)))
    print('Potencia: ' + str(np.round(power,3)) + '\n')

    data_folder = Path('./')
    file_path = data_folder / file_name

    line = str(np.round(voltage,3)) + ';' + str(np.round(current,3)) + ';' +
+ str(np.round(power,3)) + ';' + str(np.round(temp,1)) + ';' +
str(np.round(photo_diode,3))
    with open(file_name, 'a+') as fs:
        fs.write(str(line) + '\n')

    voltage_avg = round(voltage,3)
    current_avg = round(current,3)
    power_avg = round(voltage*current,3)

    pt_IV=[voltage_avg, current_avg]
    pt_W=[voltage_avg, power]

    plot_data_IV.append(pt_IV)
    plot_data_W.append(pt_W)

temp2 = average_temperature(50, temp_ch)

time.sleep(0.5)

ul.v_out(board_num, channel_out, ao_range, 0)
ul.d_out(board_num, port_type, 0) #-Apagado del foco

print('Temperatura inicial:' + str(temp1))
print('Temperatura final:' + str(temp2))
print('Aumento de temperatura de:' + str(temp2 - temp1))
```

```

plot_matrixIV = np.array(plot_data_IV)
plot_matrixIV = np.transpose(plot_matrixIV)

plot_matrixW = np.array(plot_data_W)
plot_matrixW = np.transpose(plot_matrixW)

###---FIGURE---###
plt.style.use('ggplot')
fig = plt.figure(figsize=(13,6))

ax1 = fig.add_subplot(111)
color = 'tab:red'
ax1.set_title('Referencia del Panel: ' + str(n_reference) + ' Temperatura:
' + str(np.round(temp1,1)) + '°C')
ax1.set_xlabel('Voltage (V)')
ax1.set_ylabel('Current (A)', color=color)
ax1.tick_params(axis='y', labelcolor=color)
plt.plot(plot_matrixIV[0], plot_matrixIV[1],color=color,label='I-
V',marker="*")
plt.grid()

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
color = 'tab:blue'
ax2.set_ylabel('Power (W)', color=color)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0) #Comment to plot
I-V and W with different scales
ax1.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0) #Comment to plot
I-V and W with different scales
ax1.grid()
plt.plot(plot_matrixW[0],
plot_matrixW[1],color=color,label='Power',marker=".")

fig.tight_layout()

# fig.show()
fig.set_size_inches(10, 4.6)

plt.savefig("figure_output.png")

time.sleep(0.25)

img = PhotoImage(file = "figure_output.png")
photoLabel.config(image=img)
photoLabel.image = img

###--- MÉTODO DE CARACTERIZACIÓN RESISTANCE ---###
def resistance_test(n_reference, resistances):

```

```

date = str("{0:02d}").format(time.localtime().tm_mday)
date += str("{0:02d}").format(time.localtime().tm_mon)
date += str("{0:04d}").format(time.localtime().tm_year)

t      =      str("{0:02d}{1:02d}{2:02d}").format(time.localtime().tm_hour,
time.localtime().tm_min, time.localtime().tm_sec)

file_name = str('I-V_Measures'+ '_' + date + '_' + t + '.csv')
print(file_name)

header = ('Voltaje;Corriente;Potencia;Temperatura;Fotodiodo;Resistencia')

with open(file_name, 'a+') as fs:
    fs.write('Nº Ref;' + str(n_reference) + ';\n')
    fs.write(';\n')
    fs.write(str(header) + ';\n')

temp1 = average_temperature(50, temp_ch)
port_type = DigitalPortType.AUXPORT
ul.d_config_port(board_num, port_type, 1)
ul.d_out(board_num, port_type, 0)

increment = 0.002

print(resistances)
time.sleep(1)

for iteration in range(len(resistances)):
    ###---Setting selected resistance---###
    r_load = resistances[iteration]
    z = 0.1
    r_eq = 1000000

    ul.d_out(board_num, port_type, 1)
    time.sleep(0.1)

    while r_eq > r_load:
        z = z + increment
        ul.v_out(board_num, channel, ao_range, z)

        voltage = average_voltage(100, voltage_ch)*multiplier
        current = average_voltage(100, current_ch)/r_shunt
        power = voltage * current
        r_eq = voltage/current

    print(r_eq)

temp = average_temperature(10, temp_ch)
photo_diode = average_voltage(10, diode_ch)

ul.d_out(board_num, port_type, 0)
print('Resistencia equivalente: ' + str(r_eq))

```

```

time.sleep(0.5)

print('Voltaje: ' + str(np.round(voltage,3)))
print('Corriente: ' + str(np.round(current,3)))
print('Potencia: ' + str(np.round(power,3)) + '\n')

data_folder = Path('./')
file_path = data_folder / file_name

line = str(np.round(voltage,3)) + ';' + str(np.round(current,3)) + ';'
+ str(np.round(power,3)) + ';' + str(np.round(temp,1)) + ';' +
str(np.round(photo_diode,3)) + ';' + str(np.round(photo_diode,3))
with open(file_name, 'a+') as fs:
    fs.write(str(line) + ';\n')

voltage_avg = round(voltage,3)
current_avg = round(current,3)
power_avg = round(voltage*current,3)

pt_IV=[voltage_avg, current_avg]
pt_W=[voltage_avg, power]

plot_data_IV.append(pt_IV)
plot_data_W.append(pt_W)

wait_for_temp(temp1)

plot_matrixIV = np.array(plot_data_IV)
plot_matrixIV = np.transpose(plot_matrixIV)

plot_matrixW = np.array(plot_data_W)
plot_matrixW = np.transpose(plot_matrixW)

###---FIGURE---###
plt.style.use('ggplot')
fig = plt.figure(figsize=(13,6))

ax1 = fig.add_subplot(111)
color = 'tab:red'
ax1.set_title('Referencia del Panel: ' + str(n_reference) + ' Temperatura:
' + str(np.round(temp1,1)) + '°C')
ax1.set_xlabel('Voltage (V)')
ax1.set_ylabel('Current (A)', color=color)
ax1.tick_params(axis='y', labelcolor=color)
plt.plot(plot_matrixIV[0], plot_matrixIV[1],color=color,label='I-
V',marker="*")
plt.grid()

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
color = 'tab:blue'

```

```

ax2.set_ylabel('Power (W)', color=color)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0) #Comment to plot
I-V and W with different scales
ax1.set_ylim(top=max(plot_matrixW[1]) + 0.25, bottom=0) #Comment to plot
I-V and W with different scales
ax1.grid()
plt.plot(plot_matrixW[0],
plot_matrixW[1],color=color,label='Power',marker=".")

fig.tight_layout()

# fig.show()
fig.set_size_inches(10, 4.6)

plt.savefig("figure_output.png")

time.sleep(0.25)

img = PhotoImage(file = "figure_output.png")
photoLabel.config(image=img)
photoLabel.image = img

###---GRAPHIC LOOP---###
root=Tk()

root.title("Solar Panel Test Tool")
root.geometry("1250x750")

frame = Frame()
frame.pack()

onButton = Button(frame, text="Circuito Abierto", command=opc_command,
width=15)
onButton.grid(row=0, column=3, padx=10, pady=10, sticky="e")

offButton = Button(frame, text="Cortocircuito", command=shc_command, width=15)
offButton.grid(row=1, column=3, padx=10, pady=10, sticky="e")

voltageEntry = Button(frame, text="Apagar", command=offcommand, width=15)
voltageEntry.grid(row=2, column=3, padx=10, pady=10, sticky="e")

combo = Combobox(frame, state="readonly", values=["VGS Mode", "Resistance
Mode"], width=15)
combo.grid(row=1, column=4, padx=10, pady=10)

infoLabel = Label(frame, text="Seleccione método \n de caracterización:",
width=17)
infoLabel.grid(row=0, column=4, padx=10, pady=10)

```

```
launchButton = Button(frame, text="Comenzar test", command=launchcommand,
width=15)
launchButton.grid(row=2, column=4, padx=10, pady=10)

photoLabel = Label(frame)
photoLabel.grid(row=3, column=0, columnspan=5, padx=10, pady=20)

errorLabel = Label(frame, text="")
errorLabel.grid(row=0, column=5, padx=10, pady=5, sticky = "e")

name1Label = Label(frame, text="VOLTAJE:", font=("Verdana",20))
name1Label.grid(row=0, column=0, padx=10, pady=5, sticky = "e")

voltsLabel = Label(frame, text="NA V", font=("Verdana",20), width=10)
voltsLabel.grid(row=0, column=1, padx=10, pady=5, sticky = "w")
voltage_read()

name2Label = Label(frame, text="CORRIENTE:", font=("Verdana",20))
name2Label.grid(row=1, column=0, padx=10, pady=5, sticky = "e")

currentLabel = Label(frame, text="NA A", font=("Verdana",20))
currentLabel.grid(row=1, column=1, padx=10, pady=5, sticky = "w")
current_read()

name3Label = Label(frame, text="TEMPERATURA:", font=("Verdana",20))
name3Label.grid(row=2, column=0, padx=10, pady=5, sticky = "e")

tempLabel = Label(frame, text="NA °C", font=("Verdana",20))
tempLabel.grid(row=2, column=1, padx=10, pady=5, sticky = "w")
temperature_read()

root.mainloop()
```