

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería en Mecatrónica

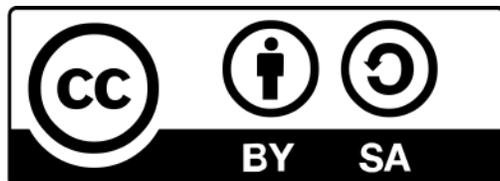


Diseño de un simulador de arreglo solar para un sistema
integrado de pruebas de potencia en CubeSats

Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Mecatrónica con el grado académico de Licenciatura

Facundo Daniel Mendoza Solano

Cartago, marzo de 2024



Esta obra está bajo una Licencia Creative Commons
Atribución-CompartirIgual 4.0 Internacional.

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Cartago, 11 de marzo 2024

Facundo Daniel Mendoza Solano
Céd: 1 1692 0062

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Facundo Daniel Mendoza Solano

Proyecto: Diseño de un simulador de arreglo solar para un sistema integrado de pruebas de potencia en CubeSats



Ing. Paula Elena Monge Chanto

Asesora

Cartago, 11 de marzo 2024

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Facundo Daniel Mendoza Solano

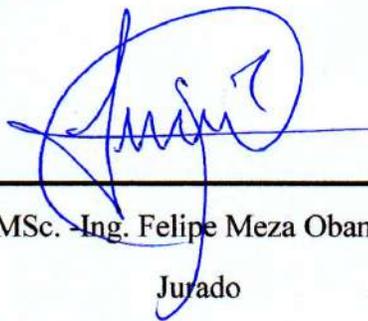
Proyecto: Diseño de un simulador de arreglo solar para un sistema integrado de pruebas de potencia en CubeSats

Miembros del jurado evaluador



Dr. -Ing. Carlos Adrián Salazar García

Jurado



MSc. -Ing. Felipe Meza Obando

Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 11 de marzo 2024

Resumen

Se presenta el proceso de diseño e implementación de un simulador de arreglo solar (SAS) para aplicaciones espaciales que forma parte del proyecto IPTC, el cual está destinado al desarrollo de una plataforma de pruebas para el sistema de potencia eléctrica (EPS) de CubeSats de 1U. El problema presentado consiste en que el EPS de los CubeSats presenta una tasa de fallos extremadamente alta, y las opciones comerciales de SAS disponibles en el mercado son muy costosas y están sobredimensionadas para las misiones CubeSat con fines académicos o de investigación. El producto derivado de este proyecto permite utilizar instrumentos de laboratorio menos costosos, como fuentes de poder y cargas electrónicas programables típicas, o bien una placa de potencia diseñada dentro del mismo marco del proyecto IPTC, a la cual se le diseñaron controladores digitales de tensión y corriente, para emular el comportamiento de arreglos solares presentes en CubeSats, con un costo de desarrollo mucho más bajo. Adicionalmente, se implementó una solución IoT que permite controlar y monitorear el SAS propuesto de manera remota, y una propuesta de chasis de bajo costo y de fácil manufactura. A través de múltiples pruebas experimentales, se comprobó el cumplimiento de las especificaciones más relevantes que fueron establecidas para validar la solución propuesta.

Palabras clave: simulador de arreglo solar, CubeSat, sistema de potencia eléctrico, internet de las cosas, chasis, controlador digital, SAS, IoT, IPTC, EPS

Abstract

The design and implementation process of a Solar Array Simulator (SAS) for space applications is presented as part of the IPTC project, which aims to develop a testing platform for the Electrical Power System (EPS) of 1U CubeSats. The problem addressed is that CubeSats' EPS has an extremely high failure rate, and the available commercial SAS alternatives in the market are too expensive and oversized for academic or research related CubeSat missions. The product derived from this project enables the use of less expensive laboratory instruments, such as typical programmable power sources and electronic loads, or a power board designed within the same framework of the IPTC project, for which digital voltage and current controllers were designed, to emulate the behavior of solar arrays present in CubeSats, with much lower development costs. Additionally, an IoT solution was implemented to remotely control and monitor the proposed SAS, along with a proposal for a low-cost, easy-to-manufacture chassis. Through multiple experimental tests, compliance with the most relevant specifications established to validate the proposed solution was confirmed.

Keywords: Solar Array Simulator, CubeSat, Electrical Power System, Internet of Things, chassis, digital controller, SAS, IoT, IPTC, EPS

Dedicatoria

Le dedico este proyecto mi familia por brindarme apoyo incondicional a lo largo de todas las etapas de mi vida, y a mi novia por acompañarme y motivarme durante los últimos cinco años.

Agradecimientos

Agradezco al Dr. Juan José Rojas Hernández por recibirme en el laboratorio, así como al MSc. Hugo Andrés Sánchez Ortiz, quienes me ofrecieron guía y conocimientos invaluable para el desarrollo exitoso del proyecto.

A la profesora Paula Monge Chanto, por haber sido una excelente asesora, cuyo criterio profesional, paciencia y dedicación fueron fundamentales durante este proceso.

A todos los profesores que fueron parte de mi formación académica y profesional.

A los compañeros y amigos que me acompañaron durante la carrera.

Índice general

Índice de tablas	V
Índice de figuras	VIII
Lista de acrónimos	XV
1. Introducción	1
1.1. Objetivos	4
1.1.1. Objetivo General	4
1.1.2. Objetivos Específicos	4
2. Marco Teórico	5
2.1. Fundamentos de sistemas fotovoltaicos	5
2.2. Modelos de rendimiento de celdas fotovoltaicas	7
2.2.1. Modelo ideal	7
2.2.2. Modelo estándar (de diodo único)	9
2.2.3. Modelo de dos diodos	11
2.3. Convertidores de conmutación DC-DC	12
2.3.1. Modelo de pequeña señal en tiempo discreto	13
2.3.2. Modelado de un convertidor reductor	17
2.4. Internet de las cosas	19
2.5. Conceptos de estadística y métricas de rendimiento	20
3. Metodología	23
3.1. Identificación y determinación de necesidades	23
3.2. Establecimiento de especificaciones objetivo	27
3.3. Generación de conceptos	35
3.4. Selección de conceptos	35
3.5. Desarrollo del concepto	36
3.6. Pruebas de concepto	36

4. Propuesta general de diseño	37
4.1. Selección de hardware	39
4.1.1. Microcontroladores	40
4.1.2. Convertidor AC-DC	42
4.1.3. Pantalla externa	43
4.1.4. Base de datos local	44
4.1.5. Interruptores	46
4.1.6. Puertos de comunicación	46
4.2. Plataformas para pruebas experimentales	47
5. Solución de Simulador de Arreglo Solar	49
5.1. Generación y selección de conceptos	51
5.1.1. Modelo analítico del arreglo solar	51
5.1.2. Guardado de datos	59
5.1.3. Pantalla externa	59
5.2. Desarrollo de conceptos	60
5.2.1. Modelo analítico: Componente #1	60
5.2.2. Modelo analítico: Componente #2	66
5.3. Validación de conceptos	83
5.3.1. Modelo analítico: Componente #1	83
5.3.2. Modelo analítico: Componente #2	103
6. Solución de controladores digitales	104
6.1. Selección de conceptos	105
6.2. Desarrollo de conceptos	105
6.2.1. Función de transferencia	105
6.2.2. Controladores digitales	109
6.3. Validación de conceptos	114
7. Solución de IoT	118
7.1. Generación y selección de conceptos	121
7.1.1. Protocolos de comunicación	121
7.1.2. Plataforma de IoT a utilizar	122
7.2. Desarrollo de conceptos	126
7.2.1. Panel de control	126
7.2.2. Base de datos	131

7.2.3. Panel de visualización	132
7.3. Validación de conceptos	136
8. Solución de Chasis	138
8.1. Generación y selección de conceptos	139
8.1.1. Selección del material de las paredes	139
8.1.2. Mecanismo de sujeción de paredes	140
8.1.3. Mecanismo de ventilación con filtrado de polvo	141
8.2. Desarrollo de conceptos	142
8.3. Validación del concepto	146
8.3.1. Prueba del método de sujeción y grosor	146
8.3.2. Prueba en la base del chasis	147
8.3.3. Prueba de las agarraderas	149
8.3.4. Desglose de materiales, peso y costos	149
9. Integración de soluciones de SAS e IoT	151
9.1. Pruebas en plataforma de SAS #1	151
9.1.1. Pruebas de diferentes dimensiones de arreglos solares	153
9.1.2. Pruebas de diferentes condiciones externas	157
9.1.3. Prueba de tiempos de refrescamiento	162
9.2. Pruebas en plataforma de SAS #2	163
9.2.1. Prueba para la creación de curvas características	164
9.2.2. Prueba de tiempos de refrescamiento	165
10. Análisis económico	167
11. Conclusiones y recomendaciones	171
11.1. Conclusiones	171
11.2. Recomendaciones	172
Bibliografía	173
A. Validación de las métricas establecidas	182
B. Repositorios del proyecto	187
C. Guías para usuarios	188

D. Recursos adicionales del chasis	189
E. Placa de desarrollo de SAS	191
F. Figuras de pruebas experimentales	195
F.1. Pruebas de diferentes dimensiones de arreglos solares	195
F.1.1. Arreglo 1x1	195
F.1.2. Arreglo 1x2	200
F.1.3. Arreglo 1x6	205
F.1.4. Arreglo 2x1	210
F.1.5. Arreglo 2x3	215
F.1.6. Arreglo 6x1	220
F.2. Pruebas de condiciones externas variables	225
F.2.1. Irradiancia	225
F.2.2. Temperatura	230
F.2.3. Ángulo de incidencia	235
G. Hojas de datos de paneles solares	240

Índice de tablas

3.1. Puntajes de importancia relativa para las necesidades.	24
3.2. Necesidades y su importancia relativa. Parte 1.	24
3.3. Necesidades y su importancia relativa. Parte 2.	25
3.4. Necesidades y su importancia relativa. Parte 3.	26
3.5. Jerarquización en necesidades primarias y secundarias.	27
3.6. Métricas y especificaciones establecidas. Parte 1.	28
3.7. Métricas y especificaciones establecidas. Parte 2.	29
3.8. Métricas y especificaciones establecidas. Parte 3.	30
3.9. Métricas y especificaciones establecidas. Parte 4.	31
3.10. Justificación de especificaciones establecidas. Parte 1.	32
3.11. Justificación de especificaciones establecidas. Parte 2.	33
3.12. Justificación de especificaciones establecidas. Parte 3.	34
4.1. Comparación de candidatos a pantalla externa.	43
5.1. Paneles solares de aplicaciones espaciales seleccionados.	55
5.2. Parámetros de entrada introducidos por el usuario.	57
5.3. Parámetros del panel solar seleccionado.	58
5.4. Parámetros de salida.	58
5.5. Puntos estudiados en el rango de condiciones externas.	64
5.6. Tiempos de ejecución del algoritmo de visión computacional.	88
5.7. Tiempos de ejecución del método de Hussein.	89
5.8. Parámetros del modelo de diodo único obtenidos mediante el método de Hussein.	91
5.9. Tiempos de ejecución de la función <i>Analytic_Model</i>	103
5.10. Tamaño de memoria de programas de la plataforma SAS #2	103
6.1. Parámetros de operación de la planta MCP.	105
6.2. Parámetros de los componentes reactivos de la planta MCP.	105

6.3. Parámetros de tiempos para el controlador digital.	107
7.1. Tamaño en Bytes de cada parámetro registrado en la base de datos.	120
7.2. Tiempos que tardan las plataformas de SAS en recibir respuesta del panel de control.	137
8.1. Precios estimados por ft ² de láminas de metal (en \$USD).	139
8.2. Precios por ft ² de materiales en el campus central del TEC (en \$USD).	139
9.1. Resultados de MAPE en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	155
9.2. Resultados de ARPE máximo de tensión en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	156
9.3. Resultados de ARPE máximo de corriente en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	156
9.4. Resultados de ARPE máximo de potencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	156
9.5. Resultados de ARPE máximo de resistencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	157
9.6. Resultados de MAPE en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales	159
9.7. Resultados de porcentaje de error relativo máximo de tensión en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales	160
9.8. Resultados de porcentaje de error relativo máximo de corriente en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales	160
9.9. Resultados de porcentaje de error relativo máximo de potencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales	160
9.10. Resultados de porcentaje de error relativo máximo de resistencia en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales	160
9.11. Tiempo de refrescamiento en la solución IoT con la plataforma de SAS #1.	162
9.12. Tiempo de refrescamiento en la solución IoT con la plataforma de SAS #2.	165
9.13. Tiempo de refrescamiento en la memoria Micro SD de la plataforma de SAS #2.	165
10.1. Costo de licencias de software utilizadas (en \$USD).	168
10.2. Valores de depreciación de elementos de hardware utilizados (en \$USD).	168

10.3. Costo de desarrollo del proyecto (en \$USD).	169
10.4. Costo total de la plataforma SAS #1 (en \$USD).	169
10.5. Costo total de la plataforma SAS #2 (en \$USD).	169
10.6. Comparación de plataformas con SAS comerciales para aplicaciones espaciales.	170
B.1. Enlaces de los repositorios del proyecto.	187
C.1. Enlaces de los recursos de la solución de chasis.	188
D.1. Enlaces de los recursos de la solución de chasis.	189
D.2. Denominaciones de los planos del chasis.	190
E.1. Ensamblaje de componentes en conectores.	191
E.2. Conexiones asociadas a interruptores.	191
E.3. Desglose de componentes y precios de placa de desarrollo.	194

Índice de figuras

2.1.	Representación de una celda solar [1].	5
2.2.	Espectro de radiación solar encima de la atmósfera y a nivel del mar [2].	6
2.3.	Circuito equivalente y curva característica de una celda solar ideal [3].	8
2.4.	Representación gráfica del factor de llenado FF [3].	9
2.5.	Circuito equivalente del modelo estándar de una celda solar [3].	9
2.6.	Efecto de la a) R_s y b) R_{sh} en el factor de llenado de la curva I-V de una celda solar [4].	10
2.7.	Circuito equivalente del modelo de dos diodos de una celda solar [3].	11
2.8.	(a) Circuito genérico de un convertidor de conmutación y (b) salida del interruptor [5].	12
2.9.	(a) Circuito de un convertidor reductor ideal y (b) su característica de control [5].	13
2.10.	Curvas que describen la derivación del modelo en tiempo discreto [6].	16
2.11.	Circuito equivalente de un convertidor reductor síncrono [5].	17
3.1.	Etapas para el diseño y desarrollo de productos [7].	23
3.2.	Diagrama de bloques. Adaptado de [7].	35
4.1.	Diagrama de bloques general.	37
4.2.	Especificaciones de candidatos a MCU_M.	40
4.3.	Matriz de selección de MCU_M.	40
4.4.	Especificaciones de candidatos a MCU_S2.	41
4.5.	Matriz de selección de MCU_S2.	41
4.6.	(a) NUCLEO-64 L073RZ [8] y (b) NodeMCU ESP8266 [9]	42
4.7.	Convertidor de conmutación AC-DC Mean Well IRM-60-24 [10].	43
4.8.	Matriz de selección para pantalla externa.	44
4.9.	Pantalla LCD1602 de SunFounder [11].	44
4.10.	Placa para tarjeta Micro SD de Adafruit [12].	45

4.11. Módulo de RTC DS3231. [13].	46
4.12. (a) Interruptor rotatorio [14] y (b) <i>rocker</i> [15].	46
4.13. (a) Conectores USB hembra a hembra [16], (b) PICKit 3 [17], y (c) cabezal de <i>jumpers</i> . [18].	47
4.14. Plataforma de SAS #1.	48
4.15. Plataforma de SAS #2.	48
5.1. Descomposición funcional de la plataforma de SAS #1.	49
5.2. Descomposición funcional de la plataforma de SAS #2.	50
5.3. Matriz de selección del modelo analítico ganador.	51
5.4. Comparación de diversos métodos para la resolución numérica del modelo de un diodo. Resultados tomados de [19].	52
5.5. Ejemplo de muestreo de una curva I-V para obtener su set de datos.	54
5.6. Componente #1 del modelo analítico propuesto.	56
5.7. Componente #2 del modelo analítico propuesto.	57
5.8. Ejemplo de datos mostrados en la pantalla externa.	59
5.9. Diagrama de flujo del algoritmo de visión computacional desarrollado. Parte 1.	60
5.10. Diagrama de flujo del algoritmo de visión computacional desarrollado. Parte 2.	61
5.11. Diagrama de flujo de la implementación del método de Hussein.	63
5.12. Diagrama de flujo del programa utilizado para confeccionar las hojas de datos de cada panel.	65
5.13. Diagrama de flujo de la función <i>Analytic_Model()</i>	66
5.14. Ciclo de control principal de la plataforma de SAS #1	69
5.15. Función para controlar instrumentos del laboratorio	70
5.16. Función para configurar el punto de operación en los instrumentos.	71
5.17. Ejemplo de corrección de un punto de operación. a) Fuente de poder y b) Carga electrónica	71
5.18. Función para corregir las salidas de los instrumentos del laboratorio.	72
5.19. Fotografía de la plataforma de SAS # 1.	73
5.20. Diagrama de flujo del ciclo de control del MCU_S2 (ESP8266).	74
5.21. Diagrama de flujo de la función <i>send_msgs_to_MCU_M()</i>	77
5.22. Diagrama de flujo del programa principal implementado en el MCU_M.	79
5.23. Diagrama de flujo de la función <i>check_msg_from_MCU_S1()</i>	79
5.24. Diagrama de flujo de la función <i>check_msg_from_MCU_S2()</i>	80
5.25. Diagrama de flujo del programa implementado en el MCU_S1.	81
5.26. Placa de desarrollo confeccionada.	82

5.27. Imágenes de curvas características originales. a) AZUR 3G30A b) AZUR 4G32C c) CESI CTJ30 d) SpectroLab UTJ.	83
5.28. Imágenes de curvas características editadas. a) AZUR 3G30A b) AZUR 4G32C c) CESI CTJ30 d) SpectroLab UTJ.	84
5.29. Comparación de curvas del panel solar AZUR 3G30A. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.	85
5.30. Comparación de curvas del panel solar AZUR 4G32C. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.	86
5.31. Comparación de curvas del panel solar CESI CTJ30. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas. . .	87
5.32. Comparación de curvas del panel solar SpectroLab UTJ. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.	88
5.33. Valores de RMSE y MAPE para diferentes valores de n en el AZUR 3G30A.	89
5.34. Valores de RMSE y MAPE para diferentes valores de n en el AZUR 4G32C.	90
5.35. Valores de RMSE y MAPE para diferentes valores de n en el CESI CTJ30. .	90
5.36. Valores de RMSE y MAPE para diferentes valores de n en el Spectrolab UTJ.	91
5.37. Diagramas de barras de los indicadores de rendimiento de los paneles solares estudiados.	92
5.38. Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar AZUR 3G30A.	93
5.39. Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar AZUR 4G32C.	94
5.40. Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar CESI CTJ30.	95
5.41. Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar Spectrolab UTJ.	96
5.42. Efectos del aumento de temperatura (a) e irradiancia (b) en la curva I-V de una celda solar [20].	98
5.43. Valores de la función coseno para diferentes ángulos de incidencia.	99
5.44. Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de temperatura de celda.	100
5.45. Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de irradiancia. . . .	101
5.46. Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de ángulo de incidencia.	102
6.1. Secuencia de pasos seguida para la obtención de los controladores digitales. .	104
6.2. Diagrama de tiempos para un controlador digital [6]	106
6.3. Diagrama de Bode para la función de transferencia en tiempo discreto de la tensión.	108

6.4. Diagrama de Bode para la función de transferencia en tiempo discreto de la corriente.	108
6.5. Controlador digital para un convertidor reductor [6].	109
6.6. Diagrama de bloques del controlador PID [6].	110
6.7. Diagrama de bloques del controlador PI [6].	112
6.8. Diagrama de bloques del controlador PID implementado en Simulink.	114
6.9. Diagrama de bloques del controlador PI implementado en Simulink.	114
6.10. Respuesta escalón unitario en controlador PID. Detalle de $\%SI$ y $t_{set2}\%$	115
6.11. Respuesta escalón unitario en controlador PID. Detalle de e_{ss}	115
6.12. Respuesta escalón unitario en controlador PI. Detalle de $\%SI$ y $t_{set2}\%$	116
6.13. Respuesta escalón unitario en controlador PI. Detalle de e_{ss}	116
7.1. Descomposición funcional de la solución de IoT.	118
7.2. Matriz de filtrado inicial de plataformas IoT.	124
7.3. Matriz de selección de plataforma IoT ganadora.	125
7.4. Panel de control donde el usuario ingresa los parámetros de entrada. La tensión es seleccionada como variable de control.	126
7.5. Tabla que muestra al usuario los valores máximos de tensión que puede introducir.	127
7.6. Panel de control donde el usuario ingresa los parámetros de entrada. La corriente es seleccionada como variable de control.	128
7.7. Mensaje que notifica al usuario que introdujo uno o varios errores erróneos en las entradas.	128
7.8. Panel de control con valores erróneos resaltados para facilitar al usuario la corrección de errores.	129
7.9. Tabla en la que el usuario puede modificar los parámetros de los paneles solares existentes, o introducir los de un panel nuevo.	130
7.10. Ejemplo de registros guardados en la base de datos en la nube.	132
7.11. Panel que permite al usuario visualizar en tiempo real los últimos valores conocidos de las entradas y salidas, así como los errores presentes en el conjunto de datos actual.	133
7.12. Gráficas en tiempo real que permiten al usuario visualizar el cambio de las salidas, los errores, y la forma de la curva característica del panel solar con el paso del tiempo.	134
7.13. Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función de la irradiancia.	135
7.14. Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función de la temperatura de celda.	135

7.15. Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función del ángulo de incidencia.	136
8.1. Secuencia de pasos seguida durante el proceso de diseño de la solución de chasis.	138
8.2. Secuencia de pasos seguida durante el proceso de diseño de la solución de chasis.	140
8.3. Método de sujeción cortando el contorno del tornillo en la lámina.	141
8.4. Método de sujeción con soporte interior de 90° de 3 vías.	141
8.5. Malla de PVC para filtración de polvo [21].	142
8.6. Vista frontal del chasis.	143
8.7. Vista trasera del chasis.	143
8.8. Vista lateral del chasis.	144
8.9. Vista superior del chasis.	145
8.10. Detalle del método de sujeción seleccionado.	145
8.11. Resultado de la simulación de factor de seguridad en el método de sujeción. .	147
8.12. Resultado de la simulación de factor de seguridad en la base del chasis. . . .	147
8.13. Resultado de la simulación de factor de seguridad en la base del chasis. . . .	148
8.14. Resultado de la simulación de factor de seguridad en la base del chasis. . . .	148
8.15. Resultado de la simulación de factor de seguridad en la base del chasis. . . .	149
8.16. Desglose del chasis. Pesos en kg y precios en \$USD	150
9.1. Instrumentos de laboratorio y ordenador ejecutando pruebas.	152
9.2. Dimensiones de arreglos solares evaluados.	153
9.3. Curvas I-V y P-V de un arreglo 2x1 (AZUR 3G30A).	154
9.4. Porcentajes de error en curvas características de un arreglo 2x1 (AZUR 3G30A).	154
9.5. MPPT a diferentes niveles de irradiancia (AZUR 3G30A).	158
9.6. MPPT a diferentes niveles de temperatura (AZUR 3G30A).	158
9.7. MPPT a diferentes ángulos de incidencia. (AZUR 3G30A).	159
9.8. Placa de desarrollo confeccionada.	163
9.9. Recreación de curvas usando la plataforma de SAS #2 vistas desde el panel de visualización de la plataforma IoT.	164
E.1. Diagrama de conexiones de la placa de desarrollo. Realizado con EasyEDA. .	192
E.2. Placa de desarrollo antes y después de colocar los componentes.	193
F.1. Curvas I-V y P-V de un arreglo 1x1 (AZUR 3G30A).	196
F.2. Porcentajes de error en curvas características de un arreglo 1x1 (AZUR 3G30A).	196
F.3. Curvas I-V y P-V de un arreglo 1x1 (AZUR 4G32C).	197

F.4. Porcentajes de error en curvas caract. de un arreglo 1x1 (AZUR 4G32C). . . 197

F.5. Curvas I-V y P-V de un arreglo 1x1 (CESI CTJ30). 198

F.6. Porcentajes de error en curvas características de un arreglo 1x1 (CESI CTJ30). 198

F.7. Curvas I-V y P-V de un arreglo 1x1 (Spectrolab UTJ). 199

F.8. Porcentajes de error en curvas caract. de un arreglo 1x1 (Spectrolab UTJ). . 199

F.9. Curvas I-V y P-V de un arreglo 1x2 (AZUR 3G30A). 201

F.10. Porcentajes de error en curvas características de un arreglo 1x2 (AZUR 3G30A). 201

F.11. Curvas I-V y P-V de un arreglo 1x2 (AZUR 4G32C). 202

F.12. Porcentajes de error en curvas caract. de un arreglo 1x2 (AZUR 4G32C). . . 202

F.13. Curvas I-V y P-V de un arreglo 1x2 (CESI CTJ30). 203

F.14. Porcentajes de error en curvas características de un arreglo 1x2 (CESI CTJ30). 203

F.15. Curvas I-V y P-V de un arreglo 1x2 (Spectrolab UTJ). 204

F.16. Porcentajes de error en curvas caract. de un arreglo 1x2 (Spectrolab UTJ). . 204

F.17. Curvas I-V y P-V de un arreglo 1x6 (AZUR 3G30A). 206

F.18. Porcentajes de error en curvas características de un arreglo 1x6 (AZUR 3G30A). 206

F.19. Curvas I-V y P-V de un arreglo 1x6 (AZUR 4G32C). 207

F.20. Porcentajes de error en curvas caract. de un arreglo 1x6 (AZUR 4G32C). . . 207

F.21. Curvas I-V y P-V de un arreglo 1x6 (CESI CTJ30). 208

F.22. Porcentajes de error en curvas características de un arreglo 1x6 (CESI CTJ30). 208

F.23. Curvas I-V y P-V de un arreglo 1x6 (Spectrolab UTJ). 209

F.24. Porcentajes de error en curvas caract. de un arreglo 1x6 (Spectrolab UTJ). . 209

F.25. Curvas I-V y P-V de un arreglo 2x1 (AZUR 3G30A). 211

F.26. Porcentajes de error en curvas características de un arreglo 2x1 (AZUR 3G30A). 211

F.27. Curvas I-V y P-V de un arreglo 2x1 (AZUR 4G32C). 212

F.28. Porcentajes de error en curvas caract. de un arreglo 2x1 (AZUR 4G32C). . . 212

F.29. Curvas I-V y P-V de un arreglo 2x1 (CESI CTJ30). 213

F.30. Porcentajes de error en curvas características de un arreglo 2x1 (CESI CTJ30). 213

F.31. Curvas I-V y P-V de un arreglo 2x1 (Spectrolab UTJ). 214

F.32. Porcentajes de error en curvas caract. de un arreglo 2x1 (Spectrolab UTJ). . 214

F.33. Curvas I-V y P-V de un arreglo 2x3 (AZUR 3G30A). 216

F.34. Porcentajes de error en curvas características de un arreglo 2x3 (AZUR 3G30A). 216

F.35. Curvas I-V y P-V de un arreglo 2x3 (AZUR 4G32C). 217

F.36. Porcentajes de error en curvas caract. de un arreglo 2x3 (AZUR 4G32C). . . 217

F.37. Curvas I-V y P-V de un arreglo 2x3 (CESI CTJ30). 218

F.38. Porcentajes de error en curvas características de un arreglo 2x3 (CESI CTJ30). 218

F.39. Curvas I-V y P-V de un arreglo 2x3 (Spectrolab UTJ). 219

F.40. Porcentajes de error en curvas caract. de un arreglo 2x3 (Spectrolab UTJ). . . 219

F.41. Curvas I-V y P-V de un arreglo 6x1 (AZUR 3G30A). 221

F.42. Porcentajes de error en curvas características de un arreglo 6x1 (AZUR 3G30A). 221

F.43. Curvas I-V y P-V de un arreglo 6x1 (AZUR 4G32C). 222

F.44. Porcentajes de error en curvas caract. de un arreglo 6x1 (AZUR 4G32C). . . 222

F.45. Curvas I-V y P-V de un arreglo 6x1 (CESI CTJ30). 223

F.46. Porcentajes de error en curvas características de un arreglo 6x1 (CESI CTJ30). 223

F.47. Curvas I-V y P-V de un arreglo 6x1 (Spectrolab UTJ). 224

F.48. Porcentajes de error en curvas caract. de un arreglo 6x1 (Spectrolab UTJ). . . 224

F.49. MPPT a diferentes niveles de irradiancia (AZUR 3G30A). 226

F.50. Porcentajes de error a diferentes niveles de irradiancia (AZUR 3G30A). . . . 226

F.51. MPPT a diferentes niveles de irradiancia (AZUR 4G32C). 227

F.52. Porcentajes de error a diferentes niveles de irradiancia (AZUR 4G32C). . . . 227

F.53. MPPT a diferentes niveles de irradiancia (CESI CTJ30). 228

F.54. Porcentajes de error a diferentes niveles de irradiancia (CESI CTJ30). 228

F.55. MPPT a diferentes niveles de irradiancia (Spectrolab UTJ). 229

F.56. Porcentajes de error a diferentes niveles de irradiancia (Spectrolab UTJ). . . 229

F.57. MPPT a diferentes niveles de temperatura (AZUR 3G30A). 231

F.58. Porcentajes de error a diferentes niveles de temperatura (AZUR 3G30A). . . . 231

F.59. MPPT a diferentes niveles de temperatura (AZUR 4G32C). 232

F.60. Porcentajes de error a diferentes niveles de temperatura (AZUR 4G32C). . . . 232

F.61. MPPT a diferentes niveles de temperatura (CESI CTJ30). 233

F.62. Porcentajes de error a diferentes niveles de temperatura (CESI CTJ30). . . . 233

F.63. MPPT a diferentes niveles de temperatura (Spectrolab UTJ). 234

F.64. Porcentajes de error a diferentes niveles de temperatura (Spectrolab UTJ). . . 234

F.65. MPPT a diferentes ángulos de incidencia. (AZUR 3G30A). 236

F.66. Porcentajes de error a diferentes ángulos de incidencia (AZUR 3G30A). . . . 236

F.67. MPPT a diferentes ángulos de incidencia. (AZUR 4G32C). 237

F.68. Porcentajes de error a diferentes ángulos de incidencia (AZUR 4G32C). . . . 237

F.69. MPPT a diferentes ángulos de incidencia. (CESI CTJ30). 238

F.70. Porcentajes de error a diferentes ángulos de incidencia (CESI CTJ30). 238

F.71. MPPT a diferentes ángulos de incidencia. (Spectrolab UTJ). 239

F.72. Porcentajes de error a diferentes ángulos de incidencia (Spectrolab UTJ). . . 239

Lista de acrónimos

AC	Corriente Alterna (<i>Alternating Current</i>)
ARPE	Porcentaje de Error Absoluto Relativo (<i>Absolute Relative Percentage Error</i>)
CNC	Control Numérico por Computadora (<i>Computer Numerical Control</i>)
CPU	Unidad de Procesamiento Central (<i>Central Processing Unit</i>)
DC	Corriente Directa (<i>Direct Current</i>)
EPS	Sistema de Potencia Eléctrica (<i>Electrical Power System</i>)
HTTP	Protocolo de Transferencia de Hipertexto (<i>Hypertext Transfer Protocol</i>)
I2C	Circuito Inter-Integrado (<i>Inter-Integrated Circuit</i>)
IoT	Internet de las Cosas (<i>Internet of Things</i>)
IPTC	Pruebas de Potencia Integradas para CubeSats (<i>Integrated Power Testing for CubeSats</i>)
LAIMI	Laboratorio Institucional de Microcomputadores del TEC
LCD	Pantalla de Cristal Líquido (<i>Liquid Crystal Display</i>)
MAPE	Porcentaje de Error Absoluto Medio (<i>Mean Absolute Percentage Error</i>)
MCP	Módulo Configurable de Potencia
MCU	Unidad de Microcontrolador (<i>Microcontroller Unit</i>)
MDF	Fibropanel de Densidad Media (<i>Medium-Density Fibreboard</i>)
MPP	Punto de Máxima Potencia (<i>Maximum Power Point</i>)
MPPT	Seguimiento del Punto de Máxima Potencia (<i>Maximum Power Point Tracking</i>)
MQTT	Transporte de Telemetría de Cola de Mensajes (<i>Message Queuing Telemetry Transport</i>)
NASA	Administración Nacional de Aeronáutica y el Espacio (<i>National Aeronautics and Space Administration</i>)
PI	Proporcional-Integral
PIC	Controlador de Interfaz Periférica (<i>Peripheral Interface Controller</i>)
PID	Proporcional-Integral-Derivativo
PWM	Modulación de Ancho de Pulso (<i>Pulse Width Modulation</i>)
RMSE	Raíz Cuadrada del Error Cuadrático Medio (<i>Root Mean Squared Error</i>)
SAS	Simulador de Arreglo Solar (<i>Solar Array Simulator</i>)
SETEC	Laboratorio de Sistemas Espaciales del TEC (<i>Space Systems Laboratory</i>)
SPI	Interfaz Periférica Serial (<i>Serial Peripheral Interface</i>)
TEC	Tecnológico de Costa Rica
UART	Receptor-Transmisor Asíncrono Universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
USB	Bus Serial Universal (<i>Universal Serial Bus</i>)

Capítulo 1

Introducción

Desde el lanzamiento del primer satélite artificial en 1957 [22], el interés, las aplicaciones y los avances que ha experimentado esta tecnología han crecido exponencialmente, a tal punto que hoy en día son herramientas fundamentales para la civilización humana. En la vida cotidiana, facilitan la navegación a través de sistemas de posicionamiento global (*Global Positioning System* o GPS), permiten el monitoreo del clima y la detección de situaciones de emergencia, así como posibilitar el acceso a Internet, televisión y llamadas telefónicas incluso en zonas rurales o muy remotas. Asimismo, en el ámbito científico y educativo son cada vez más utilizados en aplicaciones de astronomía, oceanografía y observación terrestre, siendo fundamentales para combatir el cambio climático.

La tendencia de lanzamientos de satélites ha aumentado de forma dramática en los últimos años [23], lo cual ha sido propiciado principalmente por el desarrollo de satélites cada vez más pequeños que le ha permitido a la industria espacial construir nuevos dispositivos a partir de componentes comerciales de bajo costo, baja potencia y con alta disponibilidad, posibilitando cada vez más la democratización del acceso al espacio.

Es en este contexto donde surge la tecnología CubeSat: pequeños satélites de dimensiones estandarizadas 1U, definidas por un cubo con aristas de 10 cm y un peso no mayor a 1.33 kg [24]. Un CubeSat de 1U puede operar por sí mismo como un satélite independiente, o bien, se pueden combinar varias de estas unidades para construir satélites más grandes. Originalmente surgieron como herramientas con fines didácticos y de demostración tecnológica. No obstante, en años recientes se han colocado como una alternativa real para misiones espaciales de bajo costo que presentan un alto valor de retorno comercial y de conocimientos científicos.

En nuestro país, esta tecnología ha desempeñado un papel importante desde el despliegue del proyecto Irazú [25], el cual fue el primer satélite puesto en órbita en la historia de Costa Rica y Centroamérica. Este fue un CubeSat de 1U desarrollado por el Laboratorio de Sistemas Espaciales del TEC (SETEC-Lab), que tenía como objetivos monitorear cómo los árboles de bosques de Costa Rica reaccionan ante el cambio climático y su impacto en la fijación de carbono, además de demostrar la capacidad institucional para el desarrollar proyectos de ingeniería espacial en el país. Posterior a este, también surgió el proyecto GW-Sat, un CubeSat de 3U desarrollado igualmente por SETEC-Lab, el cual será utilizado para monitorear las condiciones ambientales en los humedales ubicados en Palo Verde, Guanacaste [26].

Estos hitos han propiciado que dentro del TEC exista un inmenso interés por el desarrollo de tecnologías espaciales, tanto por parte de la comunidad docente como estudiantil. Por ejemplo, varios estudiantes de la institución han asistido a ediciones pasadas del Congreso Astronáutico Internacional (IAC) a presentar sus iniciativas [27], tales como el Proyecto MUSA, el cual dio origen a la empresa *Orbital Space Technologies* [28], responsable de la segunda misión espacial de Costa Rica después del proyecto Irazú.

Además de SETEC-Lab, de la Escuela de Ingeniería Electrónica, también existe el Laboratorio de Sistemas Espaciales Delta, de la Escuela de Ingeniería Electromecánica, el cual provee a cualquier estudiante interesado en el desarrollo de proyectos espaciales de un espacio físico, equipos, instrumentos, y asesoramiento técnico por parte de otros estudiantes más experimentados o de profesores altamente capacitados en estas áreas.

Este amplio interés ha permitido el surgimiento de nuevos proyectos que son apoyados y financiados por la Vicerrectoría de Investigación y Extensión del TEC, en los cuales estudiantes con diversos niveles de experiencia pueden colaborar como asistentes de investigación, y se han establecido convenios de cooperación mutua con universidades de países como Alemania, Japón y Estados Unidos.

Una de estas nuevas iniciativas es conocida como el proyecto IPTC (*Integrated Power Testing for CubeSats*), el cual tiene como objetivo desarrollar un sistema integrado para realizar pruebas de potencia en CubeSats, con el fin de detectar fallos relacionados con la recolección, distribución y almacenamiento de energía [29], ya que se ha determinado que el sistema de potencia eléctrica (*Electrical Power System* o EPS) es uno de los más propensos a fallar.

Aunque los satélites se diseñan teniendo en cuenta las condiciones extremas a las que se verán expuestos en el espacio, en estos se incorporan redundancias y se realizan extensas pruebas de calificación y rendimiento a nivel de parte, subsistema y sistema integrado [30] con el objetivo de reducir la tasa de fallas lo máximo posible, ya que una vez puestos en órbita, no se pueden reparar, modificar o cambiar posibles componentes defectuosos, por lo que en el caso de una falla crítica, la misión fracasaría por completo y no habría forma de remediarlo, implicando importantes pérdidas económicas. Además, a la hora de realizar las pruebas se debe tener en cuenta que las misiones en CubeSats por lo general duran de 3 a 12 meses [31], por lo que se debe garantizar que ningún subsistema falle antes de concluir por completo la misión.

El EPS es el subsistema encargado del procesamiento de la energía obtenida de las celdas fotovoltaicas, para posteriormente almacenarla y regularla para alimentar a los demás subsistemas del satélite [32], [33]. Esto lo convierte en un subsistema crítico, ya que un fallo en este significa que el satélite sería incapaz de cumplir con su misión. A partir de varias investigaciones sobre fallos en satélites, se concluyó que al menos el 45 % de estas pueden ser atribuidas a fallas eléctricas, donde el EPS es responsable del 27 % de ellas [34]. Además, en otras investigaciones se concluye que el EPS es responsable de un 28 % de las fallas durante el mismo día de lanzamiento del satélite, un 44 % de las fallas ocurridas posterior a los 30 días y un 36 % de las fallas luego de 90 días [30]. Es por esto que resulta imprescindible realizar pruebas cada vez más rigurosas para garantizar que la tasa de fallos en el EPS se vea considerablemente reducida.

Durante el desarrollo de un CubeSat, se sigue un proceso en el que primero se prueban los diferentes subsistemas por separado antes de integrarlos en el satélite. Sin embargo, actualmente no existe un módulo que pueda emular el consumo de potencia de los distintos subsistemas, lo cual representa un desafío importante para verificar el correcto funcionamiento del EPS. A partir de esta necesidad se plantea el proyecto IPTC, el cual permite llevar a cabo tres pruebas fundamentales para garantizar el correcto funcionamiento del EPS. Cada una de estas pruebas es realizada por un módulo configurable de potencia (MCP), el cual puede operar como un simulador de arreglo solar (*Solar Array Simulator* o SAS), un cargador y descargador de baterías (*Battery Charger and Discharger* o BCD) y una carga electrónica (*Electronic Load* o EL). Asimismo, debe contar con un módulo de control e interfaz (MCI), y un chasis adecuado para integrar los sistemas.

La solución actual abarca el desarrollo de un módulo SAS, el cual se implementa en una placa electrónica preexistente que contiene toda la fase potencia necesaria y que fue diseñada específicamente para los módulos del MCP. También se lleva a cabo el diseño de un chasis que contenga los elementos de hardware de la solución. Finalmente, se debe poder controlar el módulo de manera remota y monitorear algunos parámetros en tiempo real, por lo que se requiere una solución de Internet de las cosas (*Internet of Things* o IoT).

Un SAS es una fuente de alimentación eléctrica especial que, a diferencia de solo proporcionar tensión o corriente constante, intenta emular el comportamiento de un arreglo de celdas fotovoltaicas [35]. Este comportamiento se ve afectado por distintos factores, tales como la temperatura ambiente, la irradiancia solar y el ángulo de incidencia de los rayos solares sobre la superficie del panel [36]. No obstante, en el caso particular de un SAS enfocado en emular las condiciones que tendría un CubeSat colocado en órbita, resulta más complejo replicar el comportamiento de estas variables producto del constante cambio de orientación que experimenta el satélite, las temperaturas extremas a las que pueden estar sometidos, y periodos de eclipse [37]. Por lo tanto, desarrollar un sistema capaz de simular condiciones como estas que pueda luego ser integrado con los demás módulos del IPTC es imperativo para poder realizar las pruebas de potencia requeridas.

En el mercado existen soluciones comerciales de SAS altamente especializadas para emular las condiciones que se tendrían en el espacio, como por ejemplo la serie de SAS modulares MP4300A de Keysight [38] o el SAS Elgar de AMETEK [39]. No obstante, estas soluciones están dimensionadas para poder simular arreglos solares con potencias de entre 1 kW a 6 kW, además de que deben someterse a múltiples estándares de calidad de la industria, por lo que su mercado objetivo son instituciones o empresas con la capacidad de poner satélites en órbita de manera regular. Debido a esto, dichas soluciones pueden llegar costar decenas de miles de dólares estadounidenses, haciéndolas poco asequibles para fines plenamente académicos o para realizar pruebas en misiones espaciales de bajo costo, como lo fue el Proyecto Irazú.

Adicionalmente, la visión de los coordinadores del IPTC es que todo el conocimiento adquirido y los recursos desarrollados durante el diseño e implementación del proyecto sean de libre acceso para todo aquel que esté interesado en el desarrollo de la tecnología CubeSat, por lo que todas las soluciones desarrolladas deben ser de completa autoría de los colaboradores del proyecto y además deben estar debidamente documentadas para permitir el intercambio de conocimientos con miembros de otras entidades de educación o aficionados, de modo que la solución implementada pueda ser replicable.

1.1. Objetivos

1.1.1. Objetivo General

- Desarrollar un módulo de SAS que replique de manera precisa el comportamiento que tendría el arreglo solar de un CubeSat, que se pueda controlar y monitorear remotamente, y que utilice hardware preexistente y compatible con los demás módulos de potencia del proyecto IPTC.

1.1.2. Objetivos Específicos

1. Determinar las especificaciones requeridas para la solución a partir de criterios bibliográficos y de SETEC-Lab.
2. Diseñar una solución para la generación de las curvas características de un panel solar y el control de una planta preexistente.
3. Diseñar una solución de IoT para el envío de parámetros de entrada y la visualización de datos en tiempo real a través de una interfaz de usuario.
4. Diseñar un chasis para colocar los elementos de hardware de la solución de SAS propuesta.
5. Validar la integración de las componentes de la solución de SAS propuesta.

Capítulo 2

Marco Teórico

2.1. Fundamentos de sistemas fotovoltaicos

El principio de funcionamiento de las celdas solares se basa en el efecto fotovoltaico, es decir, la generación de una diferencia de potencial en la unión p-n de dos materiales semiconductores diferentes en respuesta a la radiación electromagnética, los cuales tienen electrones débilmente enlazados que ocupan una banda de energía llamada banda de valencia. Cuando a un electrón de valencia se le aplica energía que excede un cierto umbral, llamada energía de banda prohibida, los enlaces se rompen y el electrón puede moverse libremente hacia una nueva banda de energía llamada banda de conducción, donde puede “conducir” electricidad a través del material. De esta forma, los electrones libres en la banda de conducción están separados de la banda de valencia por la banda prohibida (medida en unidades de electron-voltios o eV) [40]. Esta energía necesaria para liberar el electrón puede ser suministrada por fotones, que en este caso son partículas de luz irradiadas por el Sol [1]. En la Figura 2.1 se presenta un representación de una celda solar, donde se puede apreciar cómo los fotones bombean electrones desde la banda de valencia a la banda de conducción.

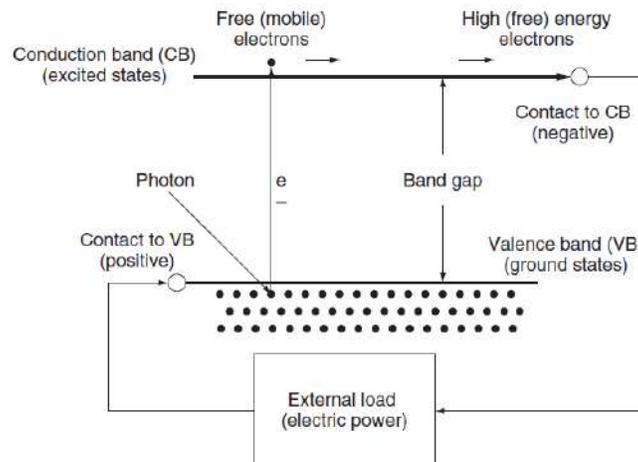


Figura 2.1: Representación de una celda solar [1].

Para utilizar la electricidad generada en dispositivos prácticos que requieren un voltaje o corriente particular para su funcionamiento, se deben conectar varias celdas solares entre sí, por lo general, en serie, para formar un panel solar, también llamado módulo fotovoltaico. No obstante, para obtener potencias aún mayores, los paneles solares se conectan entre sí, ya sea en serie, en paralelo o en una combinación de ambos, para formar un arreglo solar [40].

La eficiencia de las celdas solares depende de la capacidad de la tecnología para capturar la energía dentro del amplio espectro de radiación electromagnética que es la luz solar. Aunque el Sol emite todas las frecuencias de ondas electromagnéticas, excepto los rayos gamma, las longitudes de onda más cortas que la luz ultravioleta y más largas que la luz infrarroja tienen una irradiancia espectral muy baja, que es la energía por área de superficie [41]. Además, no toda la radiación llega a la superficie de la Tierra, porque las longitudes de onda ultravioleta son absorbidas por los gases de la atmósfera, principalmente por el ozono. De esta forma, la atmósfera actúa como un filtro de las bandas del espectro solar [2]. En la Figura 2.2, se puede apreciar el espectro de radiación solar de manera gráfica.

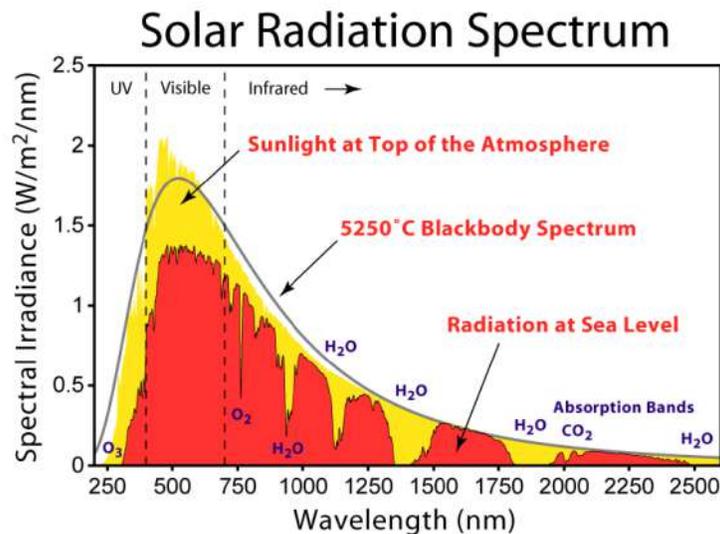


Figura 2.2: Espectro de radiación solar encima de la atmósfera y a nivel del mar [2].

Cada año existen nuevos avances en la tecnología fotovoltaica a medida que se vuelven más populares para aplicaciones terrestres, las cuales utilizan principalmente celdas de unión simple, que por lo general presentan eficiencias de alrededor del 20 %. Estas toman ventaja de no contar con limitaciones de área superficial y la naturaleza no crítica de la generación de energía [41]. No obstante, en aplicaciones espaciales sí se cuenta con una superficie limitada y la generación de energía es crítica para la misión, por lo que en este caso son preferibles celdas solares de uniones múltiples, que incorporan varias capas de material absorbente de luz que son capaces de convertir en energía más regiones de longitud de onda específicas del espectro solar, utilizando así un espectro más amplio de radiación solar. Estas van de 3 a 5 uniones basadas en elementos semiconductores del Grupo III-V (como el GaAs) [42].

Los CubeSats por lo general utilizan celdas solares de alto rendimiento que pueden proporcionar eficiencias superiores al 32 %, aunque tienen un costo sustancialmente más alto que las celdas solares de silicio para aplicaciones terrestres.

2.2. Modelos de rendimiento de celdas fotovoltaicas

Existen varios modelos matemáticos que permiten emular el comportamiento de las celdas solares en términos de su tensión y corriente de salida. Conforme aumenta la precisión del modelo, también aumenta la complejidad de la ecuación a resolver y con ello los recursos computacionales requeridos para encontrar la solución. A continuación, se presentan los tres modelos más comunes.

2.2.1. Modelo ideal

La forma más simple de modelar una celda solar consta de una fuente de corriente conectada en paralelo con un diodo rectificador [43]. La característica I-V correspondiente, se describe mediante las ecuaciones 2.1 y 2.2 de la celda solar de Shockley. En la Figura 2.3, se presenta el circuito equivalente de la celda solar ideal, así como un ejemplo de su curva característica I-V.

$$I = I_{ph} - I_D = I_{ph} - I_o \cdot \left[\exp \left(\frac{V}{n \cdot V_t} \right) - 1 \right] \quad (2.1)$$

$$V_t = \frac{N_{s,c} \cdot k \cdot T_c}{q} \quad (2.2)$$

Donde:

- I = corriente de salida de la celda [A]
- V = tensión de salida de la celda [V]
- I_{ph} = corriente fotoeléctrica [A]
- I_o = corriente de saturación del diodo [A]
- n = factor de idealidad del diodo (por lo general entre 1 y 2)
- V_t = tensión térmica [V]
- T_c = temperatura absoluta de la celda [K]
- $N_{s,c}$ = número de celdas en serie [K]
- q = carga del electrón [$1,602E-19 C$]
- k = constante de Boltzmann [$1,381E-23 J/K$]

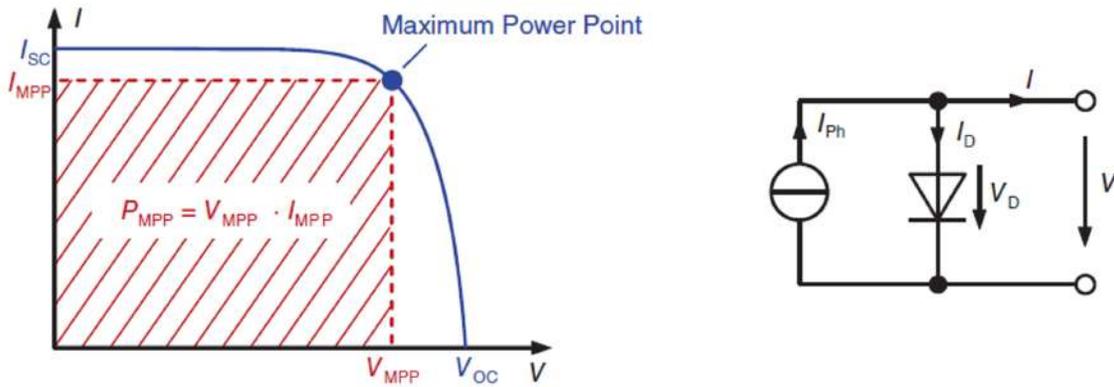


Figura 2.3: Circuito equivalente y curva característica de una celda solar ideal [3].

A continuación, se consideran algunos puntos importantes de la curva característica de una celda solar de la cual se derivan otros parámetros de interés:

- **Corriente de corto-circuito (I_{sc}):** corriente entregada por la celda solar cuando sus terminales están en corto-circuito y, por ende, su tensión V es igual a cero. Como se observa en la ecuación 2.3, esto da como resultado que I_{sc} sea equivalente a la corriente fotoeléctrica I_{ph} , y en consecuencia, I_{sc} también es proporcional a irradiancia G [3].

$$I_{sc} = I(V = 0) = I_{ph} - I_o \cdot [\exp(0) - 1] = I_{ph} \quad (2.3)$$

- **Tensión de circuito abierto (V_{oc}):** tensión resultante cuando la corriente se vuelve cero. A partir de la ecuación 2.1 y sabiendo que $I_{sc} = I_{ph}$, se obtiene la ecuación 2.4, de la cual se concluye que V_{oc} solo cambia con el logaritmo natural de la irradiancia G [3].

$$V_{oc} = V(I = 0) = n \cdot V_t \cdot \ln \left(\frac{I_{sc}}{I_o} + 1 \right) = n \cdot V_t \cdot \ln \left(\frac{I_{sc}}{I_o} \right) \quad (2.4)$$

- **Punto de máxima potencia (MPP):** es el punto de operación de la celda solar en el que se proporciona la máxima potencia. Dado que la potencia de un punto de operación siempre corresponde a la superficie $V \cdot I$, esta área debe ser la máxima en el caso del MPP, tal y como se observa en la Figura 2.3. Los valores de corriente y voltaje asociados con este punto se denominan I_{MPP} y V_{MPP} [3].
- **Factor de llenado (FF):** muestra la relación entre la potencia en el MPP con el producto entre I_{sc} y V_{oc} , como se observa en la ecuación 2.5. En la Figura 2.4, se puede apreciar que FF representa el tamaño del área bajo el punto de operación MPP comparada con el área $V_{oc} \cdot I_{sc}$ [3].

$$FF = \frac{V_{MPP} \cdot I_{MPP}}{V_{oc} \cdot I_{sc}} = \frac{P_{MPP}}{V_{oc} \cdot I_{sc}} \quad (2.5)$$

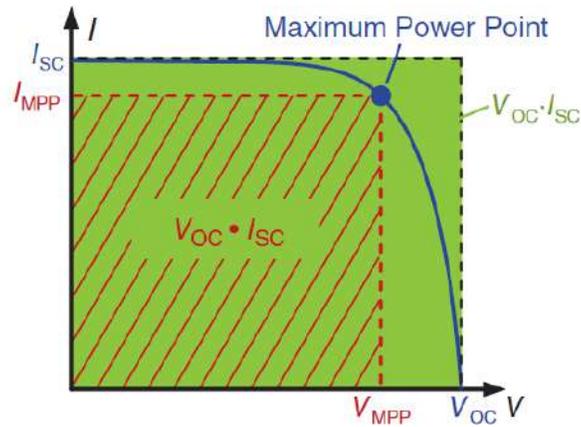


Figura 2.4: Representación gráfica del factor de llenado FF [3].

- **Eficiencia (η):** describe cuánta de la potencia óptica P_{Opt} que incide en la celda solar, se convierte en energía eléctrica P_{MPP} , tal y como se observa en la ecuación 2.6, donde A corresponde al área superficial de la celda [3].

$$\eta = \frac{P_{MPP}}{P_{Opt}} = \frac{FF \cdot V_{oc} \cdot I_{sc}}{G \cdot A} \quad (2.6)$$

2.2.2. Modelo estándar (de diodo único)

El modelo estándar, también llamado modelo de diodo único, profundiza en las pérdidas eléctricas existentes en la celda solar al agregar dos resistencias, y su circuito equivalente se presenta en la Figura 2.5.

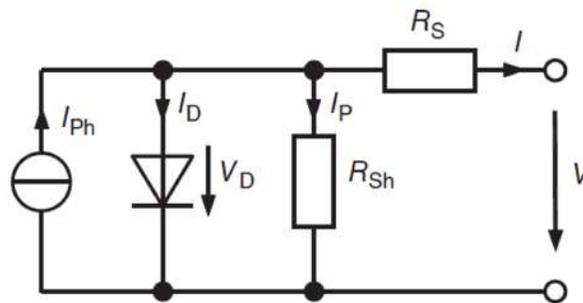


Figura 2.5: Circuito equivalente del modelo estándar de una celda solar [3].

La resistencia en serie, R_s , describe las pérdidas óhmicas en los contactos frontales de la celda solar y en la interfaz metal-semiconductor [3], [44]. En la curva característica I-V, se observa su efecto después del MPP, como se observa en la Figura 2.6.a, donde la línea azul corresponde a un comportamiento ideal, y en los demás la R_s es progresivamente más grande.

Por otra parte, la resistencia en paralelo, de derivación o *shunt*, R_{sh} , modela las corrientes de fuga en los bordes de la celda solar, así como cualquier cortocircuito puntual de la unión p-n [3], [45]. En la curva característica I-V de los paneles solares, esto se ve reflejado en la magnitud de la pendiente de la corriente antes del MPP, como se observa en la Figura 2.6.b, donde la línea azul corresponde a un comportamiento ideal, y en los demás la R_{sh} es progresivamente más pequeña.

Por lo tanto, entre más grande sea el valor de R_s y más pequeño sea el valor de R_{sh} , el factor de llenado FF se ve más reducido.

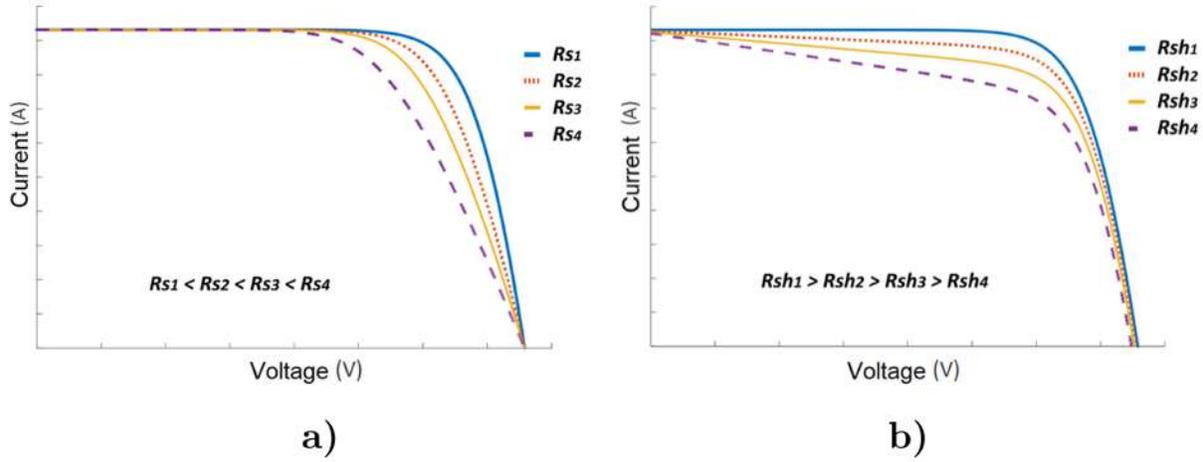


Figura 2.6: Efecto de la a) R_s y b) R_{sh} en el factor de llenado de la curva I-V de una celda solar [4].

Para obtener la característica I-V del modelo ideal, se parte de la corriente de salida $I = I_{ph} - I_D - I_{sh}$, donde I_{sh} se obtiene de:

$$I_{sh} = \frac{V_D}{R_{sh}} = \frac{V + I \cdot R_s}{R_{sh}} \quad (2.7)$$

Por lo tanto,

$$I = I_{ph} - I_o \left[\exp \left(\frac{V + I \cdot R_s}{n \cdot V_t} \right) - 1 \right] - \frac{V + I \cdot R_s}{R_{sh}} \quad (2.8)$$

Se puede apreciar que, dado que la corriente I aparece en ambos lados de la ecuación 2.8, esta debe ser resuelta numéricamente. Además, existen ciertos parámetros que por lo general los fabricantes de celdas solares no proporcionan en sus hojas de datos (n , I_{ph} , I_{ph} , R_s y R_{sh}), razón por la cual el modelo estándar también es conocido como el modelo de cinco parámetros.

2.2.3. Modelo de dos diodos

En el modelo de diodo único se supuso, por simplicidad, que no habría recombinación en la región de carga espacial del diodo a la hora de utilizar la ecuación de Shockley. Esto provoca que, en el caso de semiconductores con grandes bandas prohibidas, existan desviaciones significativas entre las curvas características reales y las modeladas. Por lo tanto, se utilizan un segundo diodo con el fin de tomar en cuenta tanto la corriente de difusión como la de recombinación [3]. El beneficio clave de utilizar el modelo de dos diodos es que ofrece una mayor precisión cuando los niveles de irradiancia solar son bajos [46]. No obstante, en este caso existen siete parámetros desconocidos, como se puede observar en la ecuación 2.9 de la característica I-V de la celda, por lo que resolverla resulta bastante más complejo. En la Figura 2.7, se presenta el circuito equivalente de este modelo.

$$I = I_{ph} - I_{o,1} \left[\exp \left(\frac{V + I \cdot R_s}{n_1 \cdot V_t} \right) - 1 \right] - I_{o,2} \left[\exp \left(\frac{V + I \cdot R_s}{n_2 \cdot V_t} \right) - 1 \right] - \frac{V + I \cdot R_s}{R_{sh}} \quad (2.9)$$

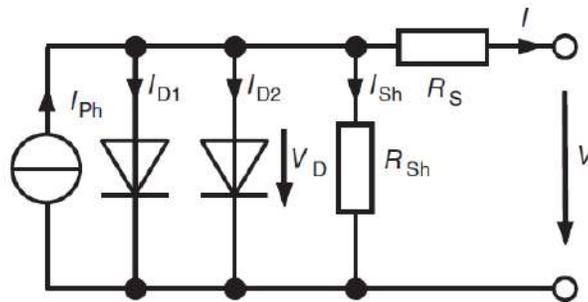


Figura 2.7: Circuito equivalente del modelo de dos diodos de una celda solar [3].

2.3. Convertidores de conmutación DC-DC

El principio de funcionamiento de este tipo de convertidores se basa en conmutar dispositivos semiconductores entre dos estados de muy baja disipación de potencia: encendido, donde la tensión existente es mínima, y apagado, donde el flujo de corriente es casi nulo. Esto permite obtener ondas pulsantes que luego se filtran con condensadores y bobinas de muy bajas pérdidas, dando lugar a tensiones de salida estables y con un rizado reducido. Al compararlos con los reguladores lineales convencionales, estos presentan eficiencias más altas, además de un tamaño y peso reducido. No obstante, la transición continua entre estados elevan el ruido y su diseño y control es más complejo [47].

Por lo tanto, estos convertidores cuentan con tres fases fundamentales: entrada de tensión, conmutación y filtrado. Dado que la fase de conmutación genera una conexión intermitente entre la entrada de tensión y el filtrado, se genera un ciclo de trabajo D que, según la topología del convertidor, transmite una tensión que puede ser mayor o menor a la que se tiene en la entrada.

En la Figura 2.8, se muestra el circuito de un convertidor convencional (a) y la curva de salida del interruptor $v_s(t)$ (b). En esta se puede apreciar que el interruptor genera una onda rectangular donde $v_s(t) = V_g$ cuando el interruptor se encuentra en la posición 1, y cero cuando está en la posición 2. El ciclo de trabajo D corresponde a la fracción de tiempo que el interruptor se encuentra en la posición 1, y se encuentra entre cero y uno, y el complemento del ciclo de trabajo D' , se define como $(1 - D)$. Además, se tiene la frecuencia de conmutación f_s que es inversa al periodo de conmutación T_s , y por lo general se encuentra en el rango de 1 kHz y 1 MHz, dependiendo de la velocidad de conmutación del dispositivo semiconductor en cuestión, que por lo general son transistores o diodos [5].

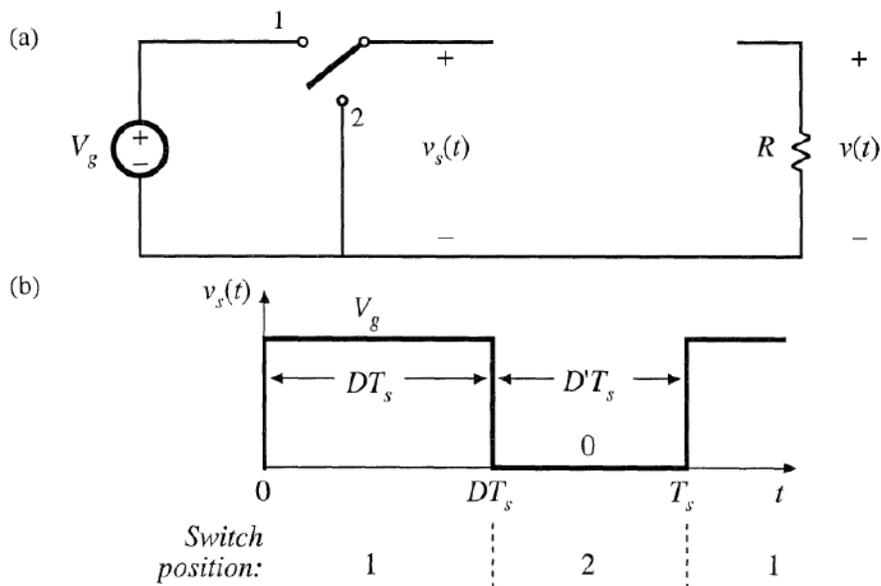


Figura 2.8: (a) Circuito genérico de un convertidor de conmutación y (b) salida del interruptor [5].

Para obtener una representación matemática de la tensión de salida del circuito, se utiliza el análisis de Fourier. Primero, se obtiene la tensión promedio utilizando la ecuación 2.10. Se puede apreciar entonces, que la componente DC promedio a la salida del convertidor, corresponde a la tensión de entrada V_g , multiplicada por un factor D . Dado que $0 < D < 1$, también se puede concluir que la tensión a la salida del interruptor, será menor a la que se tiene en la entrada.

$$\langle v_s \rangle = \frac{1}{T_s} \int_0^{T_s} v_s(t) dt = \frac{1}{T_s} (DT_s V_g) = DV_g \quad (2.10)$$

No obstante, debido a que la conmutación genera armónicos no deseados, es necesario colocar un filtro paso bajo como el de la Figura 2.9.a, resultando en la topología básica de un convertidor de tipo reductor ideal, cuya salida es esencialmente igual a la componente DC de $v_s(t)$, como se observa en la ecuación 2.11. Además, en la la Figura 2.9.b, se puede observar la característica de control del convertidor.

$$v \approx \langle v_s \rangle = DV_g \quad (2.11)$$

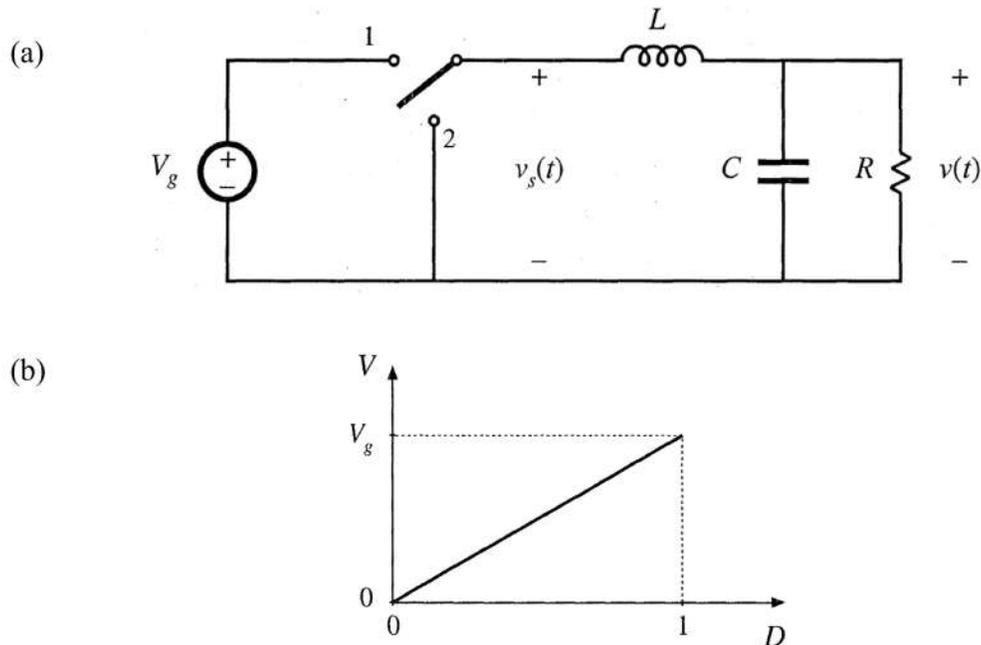


Figura 2.9: (a) Circuito de un convertidor reductor ideal y (b) su característica de control [5].

2.3.1. Modelo de pequeña señal en tiempo discreto

A continuación, se describe el modelado de un convertidor DC-DC utilizando la técnica de pequeña señal en tiempo discreto [6].

Se parte del modelado del sistema en el espacio de estados, considerando la operación del convertidor como una alternancia entre dos estados topológicos S_0 y S_1 (apagado y encendido), los cuales están descritos por un conjunto lineal de ecuaciones de espacio de estados:

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{v}(t) \\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t) + \mathbf{E}_c\mathbf{v}(t)\end{aligned}\quad (2.12)$$

Donde $c \in \{0, 1\}$, corresponde la señal PWM que denota el estado topológico, x y v representan los vectores de estado de entrada, y el de salida es representado por y . Dado que el modelado se centra en la dinámica de control a salida, el vector de entrada se asume como una constante $v(t) = \mathbf{V}$.

El modelado de señales pequeñas en tiempo discreto puede considerarse como un proceso de tres pasos:

1. Se expresa el vector de estado del convertidor muestreado x en el instante $k + 1$ en términos del vector de estado, el vector de entrada v y la entrada de control u en el instante k . Al hacerlo, generalmente se obtiene una ecuación de estado no lineal, donde f es una función vectorial no lineal.

$$\mathbf{x}[k + 1] = \mathbf{f}(\mathbf{x}[k], \mathbf{V}, u[k]) \quad (2.13)$$

2. El punto de operación Q del convertidor se determina resolviendo la ecuación mencionada anteriormente para un vector de estado muestreado constante $x[k + 1] = x[k] = \mathbf{X}$ y una entrada de control constante $u[k] = U$:

$$\mathbf{X} = \mathbf{f}(\mathbf{X}, \mathbf{V}, U) \quad (2.14)$$

Donde el punto de operación del convertidor Q se define como:

$$\mathbf{Q} = (\mathbf{X}, \mathbf{V}, U) \quad (2.15)$$

3. La ecuación de estado no lineal se perturba y se linealiza sucesivamente en las proximidades de Q , produciendo la descripción del espacio de estados de señal pequeña de la dinámica muestreada:

$$\begin{aligned}\hat{\mathbf{x}}[k + 1] &= \Phi\hat{\mathbf{x}}[k] + \gamma\hat{u}[k] \\ \hat{\mathbf{y}}[k] &= \delta\hat{\mathbf{x}}[k]\end{aligned}\quad (2.16)$$

Donde se tienen los componentes de pequeña señal del vector de estado muestreado, el comando de control y el vector de salida, en relación con sus componentes de CC \mathbf{X} , U e \mathbf{Y} , respectivamente.

$$\begin{aligned}\hat{\mathbf{x}}[k] &= \mathbf{x}[k] - \mathbf{X} \\ \hat{u}[k] &= u[k] - U \\ \hat{\mathbf{y}}[k] &= \mathbf{y}[k] - \mathbf{Y}\end{aligned}\quad (2.17)$$

Por otra parte, las matrices Φ y γ representan la matriz de estado de señal pequeña y la matriz de control a estado de señal pequeña, respectivamente, mientras que la matriz δ , representa la matriz de salida del convertidor perteneciente a la subtopología en la que se produce el muestreo. Cabe destacar también que las matrices \mathbf{E}_c no entran en el modelo de señal pequeña de control a salida.

$$\Phi = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{Q}} \quad (2.18)$$

$$\gamma = \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{\mathbf{Q}} \quad (2.19)$$

$$\delta = \begin{cases} \mathbf{C}_1 & \text{si ocurre en } S_1 \\ \mathbf{C}_0 & \text{si ocurre en } S_0 \end{cases} \quad (2.20)$$

En el dominio z , el conjunto de ecuaciones 2.16 se convierte en:

$$\begin{aligned} \hat{\mathbf{x}}(z) &= (z\mathbf{I} - \Phi)^{-1} \gamma \hat{u}(z) \\ \hat{\mathbf{y}}(z) &= \delta \hat{\mathbf{x}}(z) \end{aligned} \quad (2.21)$$

Mientras que la matriz de transferencia de control a salida de señal pequeña $\mathbf{W}(z)$, corresponde a:

$$\mathbf{W}(z) = \frac{\hat{\mathbf{y}}(z)}{\hat{u}(z)} = \delta (z\mathbf{I} - \Phi)^{-1} \gamma \quad (2.22)$$

Si se define al vector de salida en términos de la corriente del inductor y el voltaje de salida cuando $\mathbf{y} = [i_L \ v_o]^T$, $\mathbf{W}(z)$ se convierte en:

$$\mathbf{W}(z) = \begin{bmatrix} G_{iu}(z) = \frac{\hat{i}_L(z)}{\hat{u}(z)} \\ G_{vu}(z) = \frac{\hat{v}_o(z)}{\hat{u}(z)} \end{bmatrix} \quad (2.23)$$

La matriz de transferencia $\mathbf{W}(z)$ representa el resultado final del enfoque de modelado en tiempo discreto y el punto de partida para el diseño del compensador.

Ahora bien, para obtener el modelo resultante, se requiere calcular las matrices Φ y γ , partiendo del conjunto de ecuaciones 2.12. Primero, se debe considerar que dentro de un estado topológico dado S_c , la solución general del estado inicial $\mathbf{x}(t_0)$ es:

$$\mathbf{x}(t) = \exp(\mathbf{A}_c(t - t_0)) \cdot \mathbf{x}(t_0) + \int_{t_0}^t \exp(\mathbf{A}_c(t - \tau)) \cdot \mathbf{B}_c \cdot \mathbf{V} \cdot d\tau \quad (2.24)$$

En el primer paso de la derivación del modelado de señales pequeñas en tiempo discreto, simplemente se hace un uso repetido de la ecuación 2.24 a lo largo del intervalo de conmutación para expresar $\mathbf{x}[k + 1]$ en términos de $\mathbf{x}[k]$, \mathbf{V} y $u[k]$. Al hacerlo hay que tener en

cuenta que el vector de estado es una función continua del tiempo y, por tanto, no presenta discontinuidades en los instantes de conmutación.

Una vez que se dispone de una expresión para \mathbf{f} , se siguen los pasos dos y tres del procedimiento de modelado de señales pequeñas sin dificultades conceptuales. Para el escenario descrito anteriormente, el resultado para la matriz de estado y para la matriz de entrada, se presentan en las ecuaciones 2.25 y 2.26, respectivamente.

$$\Phi = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{Q}} = \exp(\mathbf{A}_0(T_s - t_d)) \cdot \exp(\mathbf{A}_1 DT_s) \cdot \exp(\mathbf{A}_0(t_d - DT_s)) \quad (2.25)$$

$$\gamma = \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{\mathbf{Q}} = \frac{T_s}{N_r} \cdot \exp(\mathbf{A}_0(T_s - t_d)) \cdot [(\mathbf{A}_1 \mathbf{X}_{\downarrow} + \mathbf{B}_1 \mathbf{V}) - (\mathbf{A}_0 \mathbf{X}_{\downarrow} + \mathbf{B}_0 \mathbf{V})] \quad (2.26)$$

El vector $\mathbf{X}_{\downarrow} = \mathbf{x}_s(t_{\downarrow})$ es el valor de $\mathbf{x}_s(t)$ en el borde modulado de $c_s(t)$, como se indica en la Figura 2.10. \mathbf{X}_{\downarrow} se deriva del punto de funcionamiento del convertidor, equivalente a:

$$\mathbf{X}_{\downarrow} = [\mathbf{I} - \exp(\mathbf{A}_1 DT_s) \cdot \exp(\mathbf{A}_0 D' T_s)]^{-1} \cdot [\exp(\mathbf{A}_1 DT_s) \cdot \mathbf{A}_0^{-1} (\mathbf{I} - \exp(\mathbf{A}_0 D' T_s)) \mathbf{B}_0 - \mathbf{A}_1^{-1} (\mathbf{I} - \exp(\mathbf{A}_1 DT_s)) \mathbf{B}_1] \mathbf{V} \quad (2.27)$$

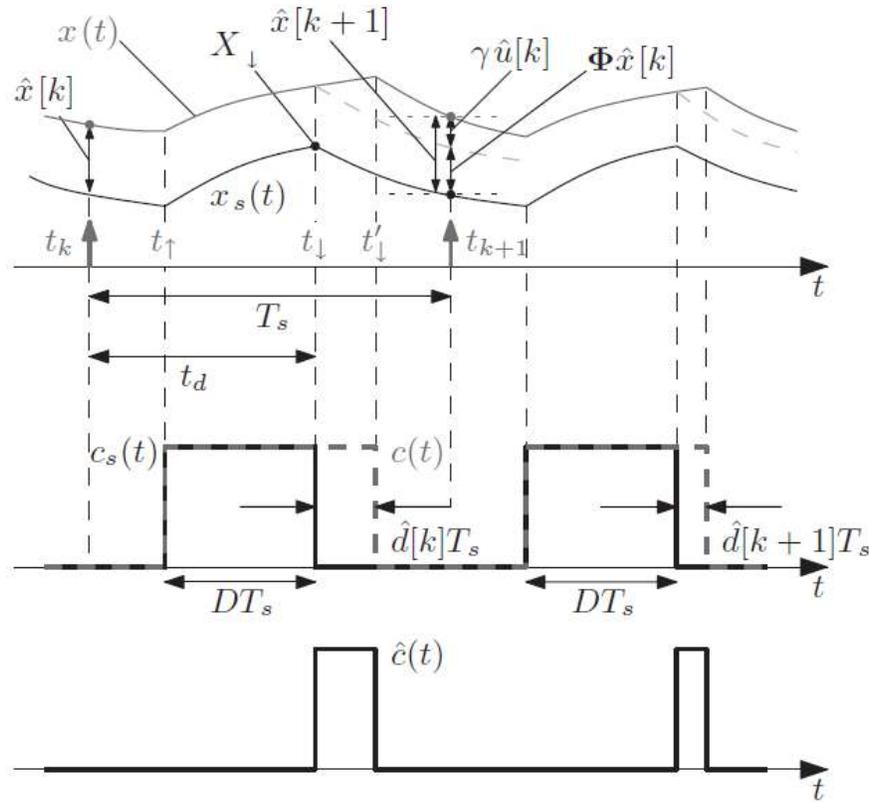


Figura 2.10: Curvas que describen la derivación del modelo en tiempo discreto [6].

2.3.2. Modelado de un convertidor reductor

En la Figura 2.11, se presenta el circuito equivalente de un convertidor reductor real. Se puede apreciar que, a diferencia del circuito ideal de la Figura 2.8, este cuenta con las resistencias r_L y r_C del inductor y del condensador, respectivamente.

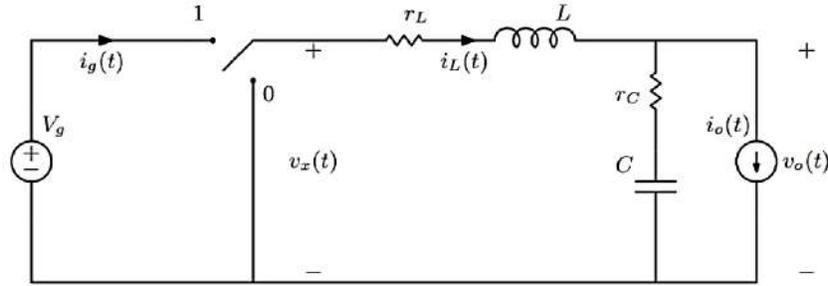


Figura 2.11: Circuito equivalente de un convertidor reductor síncrono [5].

Se procede con el planteamiento de las ecuaciones diferenciales de la corriente y la tensión de salida. A partir de análisis de mallas y de nodos cuando el interruptor se encuentra en la posición 1, se obtienen los conjuntos de ecuaciones 2.28 y 2.29:

$$\begin{aligned}
 0 &= -L \frac{d}{dt} i_L(t) + V_g - v_C(t) - i_L(t)(r_L + r_C) + I_o r_C \\
 \Rightarrow \frac{d(t)}{dt} i_L &= \frac{V_g}{L} - \frac{v_C(t)}{L} - \frac{i_L(t)(r_L + r_C)}{L} + \frac{I_o r_C}{L}
 \end{aligned} \tag{2.28}$$

$$\begin{aligned}
 0 &= C \frac{d}{dt} v_c - i_L(t) + I_o \\
 \Rightarrow \frac{d}{dt} v_c &= \frac{i_L(t)}{C} - \frac{I_o}{C}
 \end{aligned} \tag{2.29}$$

Se realiza el mismo procedimiento cuando el interruptor se encuentra en la posición 0, obteniendo los conjuntos de ecuaciones 2.30 y 2.31:

$$\begin{aligned}
 0 &= L \frac{d}{dt} i_L(t) + v_C(t) + i_L(t)(r_L + r_C) - I_o r_C \\
 \Rightarrow \frac{d}{dt} i_L(t) &= -\frac{v_C(t)}{L} - \frac{i_L(t)(r_L + r_C)}{L} + \frac{I_o r_C}{L}
 \end{aligned} \tag{2.30}$$

$$\begin{aligned}
 0 &= C \frac{d}{dt} v_c(t) - i_L(t) + I_o \\
 \Rightarrow \frac{d}{dt} v_c(t) &= \frac{i_L(t)}{C} - \frac{I_o}{C}
 \end{aligned} \tag{2.31}$$

Se puede observar que en estas ecuaciones diferenciales existen dos estados, $x_1(t)$ y $x_2(t)$, además de dos grupos de matrices para cuando el interruptor esté en las posiciones 0 y 1. A partir de la ecuación 2.12 y los estados definidos anteriormente, se obtienen las siguientes ecuaciones en el espacio de estados:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} &= \mathbf{A}_c \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{B}_c \begin{bmatrix} V_g \\ I_o \end{bmatrix} \\ \mathbf{y}(t) &= \mathbf{C}_c \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{E}_c \begin{bmatrix} V_g \\ I_o \end{bmatrix} \end{aligned} \quad (2.32)$$

$$\begin{aligned} \begin{bmatrix} \frac{d}{dt} i_L(t) \\ \frac{d}{dt} v_C(t) \end{bmatrix} &= \mathbf{A}_c \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \mathbf{B}_c \begin{bmatrix} V_g \\ I_o \end{bmatrix} \\ \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} &= \mathbf{C}_c \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \mathbf{E}_c \begin{bmatrix} V_g \\ I_o \end{bmatrix} \end{aligned} \quad (2.33)$$

Donde:

$$\mathbf{A}_0 = \mathbf{A}_1 = \begin{bmatrix} \frac{-(r_C + r_L)}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \quad (2.34)$$

$$\mathbf{B}_0 = \begin{bmatrix} 0 & \frac{r_C}{L} \\ 0 & -\frac{1}{C} \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} \frac{1}{L} & \frac{r_C}{L} \\ 0 & -\frac{1}{C} \end{bmatrix} \quad (2.35)$$

$$\mathbf{C}_0 = \mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ r_C & 1 \end{bmatrix} \quad (2.36)$$

$$\mathbf{E}_c = 0 \quad (2.37)$$

Una vez aplicado el modelado de pequeña señal en tiempo discreto descrito anteriormente, se toma la matriz Φ como la matriz \mathbf{A}_c , la matriz γ como la matriz \mathbf{B}_c , y la matriz δ como la matriz \mathbf{C}_c , para obtener las funciones de transferencia en tiempo discreto para la tensión y corriente de salida del convertidor. Esto se explica con mayor detalle en el Capítulo 6.

2.4. Internet de las cosas

El Internet de las cosas (IoT por sus siglas en inglés) se refiere a una red de objetos físicos que están integrados con sensores, software y conectividad de red que les permite recopilar y compartir datos. Estos dispositivos, también conocidos como “objetos inteligentes”, pueden ser simples dispositivos domésticos como termostatos, luces o electrodomésticos, dispositivos portátiles como relojes inteligentes o ropa con identificadores de radiofrecuencia, hasta complejas maquinarias industriales y sistemas de transporte [48]. El IoT permite que estos dispositivos se comuniquen entre sí, con otros dispositivos habilitados para Internet, como teléfonos inteligentes u ordenadores, y con la nube, creando una vasta red de dispositivos interconectados que pueden intercambiar datos y realizar una variedad de tareas de forma autónoma [49].

A continuación se describen algunos conceptos relevantes para el diseño de una solución de IoT:

- **Plataforma de IoT:** es una red masiva en la nube que admite dispositivos y aplicaciones de IoT. Esto incluye la infraestructura subyacente, los servidores y el almacenamiento necesarios para las operaciones y el procesamiento en tiempo real [50].
- **Interfaz de usuario:** las funciones mediante las que un usuario interactúa con un sistema informático. Esto incluye botones, iconos, cajas de texto, elementos gráficos, entre otros [51]. En términos de IoT, estas interfaces pueden utilizarse tanto para controlar el comportamiento de los dispositivos, como para visualizar los datos que estos envían.
- **Base de datos:** colección organizada de información o datos estructurados, que por lo general se modelan en filas y columnas en una serie de tablas para que el procesamiento y la consulta de datos sean eficientes [52].
- **HTTP:** el Protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés) se utiliza para estructurar solicitudes y respuestas a través de Internet, permitiendo que los datos se transfieran de un punto a otro a través de la red [53]. Un cliente realiza una solicitud HTTP a un anfitrión designado que se encuentra en un servidor, y el objetivo de la solicitud es acceder a un recurso en dicho servidor. Para realizar la solicitud, el cliente utiliza componentes de un Localizador Uniforme de Recursos (URL por sus siglas en inglés), que incluye la información necesaria para acceder al recurso [54].
- **MQTT:** el Transporte de Telemetría de Cola de Mensajes (MQTT por sus siglas en inglés) es un protocolo de mensajería ligero diseñado para dispositivos con recursos limitados y redes de bajo ancho de banda, alta latencia o poco confiables [55]. Su modelo basado en publicación y suscripción, permite una comunicación bidireccional fluida entre dispositivos, ya que separa al cliente que envía mensajes (publicador) del cliente que recibe mensajes (suscriptor). Los publicadores y suscriptores no necesitan establecer una conexión directa, ya que incorpora una tercera componente, conocida como intermediario o *broker*, que es responsable de enrutar y distribuir todos los mensajes [56].

- **Tiempo de respuesta:** métrica de rendimiento que mide el tiempo necesario para reaccionar ante una entrada [57]. En términos de servicios IoT en la nube, se podría interpretar como el tiempo que pasa desde que el usuario introduce una solicitud a través de la interfaz de control de la plataforma IoT, hasta que el dispositivo que se desea controlar recibe la información. También se podría interpretar como el tiempo que pasa desde que un dispositivo envía información a un servidor para que esta se guarde, hasta el momento en que dicha información se vea reflejada en una base datos.

2.5. Conceptos de estadística y métricas de rendimiento

A continuación, se explican algunos conceptos estadísticos y métricas de rendimiento relevantes para el proyecto. Asimismo, se presentan sus respectivas ecuaciones, en las cuales se consideran los siguientes parámetros:

x = variable de estudio aleatoria

y = valor teórico o de referencia

\hat{y} = valor experimental o calculado por modelo

n = número de valores estudiados

- **Promedio:** También conocido como media aritmética, consiste en la suma de todos los valores numéricos de un conjunto de datos dividida entre el número total de valores de dicho conjunto, con el fin de obtener un número que pueda representar de la mejor manera a todos los valores.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x \quad (2.38)$$

- **Desviación estándar (SD):** Indica qué tan dispersos están los datos con respecto a la media. Mientras mayor sea la desviación estándar, mayor será la dispersión de los datos.

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.39)$$

- **Desviación estándar relativa (RSD):** Indica qué tan grande o pequeña es la SD respecto a la media del conjunto de datos.

$$RSD = 100 \cdot \frac{s}{\bar{x}} \quad (2.40)$$

- **Error absoluto (AE) y Error Absoluto Medio (MAE):** El AE corresponde al valor absoluto de la diferencia entre el valor medido o predicho de una cantidad, y su valor real, teórico o de referencia. El promedio de los AE de un conjunto de datos se conoce como MAE.

$$AE = |y - \hat{y}| \quad (2.41)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.42)$$

- **Error Absoluto Relativo (ARE) y Error Absoluto Relativo Medio (MARE):** El ARE describe el cociente entre el error absoluto de la medición o predicción, y el valor real, teórico o de referencia. El promedio de ARE de un conjunto de datos se conoce como MARE.

$$\text{ARE} = \frac{AE}{y} = \frac{|y - \hat{y}|}{y} \quad (2.43)$$

$$\text{MARE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \quad (2.44)$$

- **Porcentaje de Error Absoluto Relativo (ARPE) y Porcentaje de Error Absoluto Medio (MAPE):** El ARE y el MARE también se suelen expresar como porcentajes, en cuyo caso se les denomina ARPE y MAPE, respectivamente.

$$\text{ARPE} = 100 \cdot \frac{|y - \hat{y}|}{y} \quad (2.45)$$

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \quad (2.46)$$

El MAPE es una de las métrica de rendimiento más utilizadas para medir la precisión de predicciones, ya que es simple de calcular y su interpretación es muy intuitiva, ya que expresa el error de pronóstico como un porcentaje del valor real. No obstante, también presenta desventajas significativas, ya que cuando el valor oscila en valores cercanos a cero, se genera un MAPE desproporcionadamente grande a pesar de que en realidad sí se esté siguiendo la tendencia general. Además, cuando se obtienen valores reales iguales a cero, el MAPE se indefine.

- **Error Cuadrático (SE), Error Cuadrático Medio (MSE) y Raíz Cuadrada del Error Cuadrático Medio (RMSE):** El SE es la diferencia cuadrática que existe entre el valor real o de referencia, y los valores predichos a partir de un modelo. El MSE es el promedio de los valores de SE y permite determinar qué tan cerca está una línea de regresión de un conjunto de puntos. La elevación al cuadrado permite eliminar cualquier signo negativo, además de castigar más fuertemente las diferencias grandes en comparación con otros indicadores como el MAPE. No obstante, esta también hace que el orden de la unidad resultante sea mayor que la unidad original de la variable de estudio. Debido a esto, es común calcular la raíz cuadrada del MSE para obtener el mismo orden unitario, lo cual se conoce como RMSE.

$$\text{SE} = (y_i - \hat{y}_i)^2 \quad (2.47)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.48)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.49)$$

- **Raíz Cuadrada del Error Cuadrático Medio Normalizado (nRMSE):** Una desventaja del RMSE es que este no proporciona una medida estandarizada, es decir, no está normalizada por ninguna escala, lo que dificulta la comparación entre diferentes conjuntos de datos o modelos. En el caso de modelos de rendimiento de celdas solares, normalizar el RMSE puede ser útil al comparar el rendimiento obtenido para diferentes paneles solares comerciales, los cuales pueden presentar variaciones considerables en sus rangos de operación de corriente y tensión. Si bien existen múltiples métodos de normalización, tomando en cuenta el contexto explicado anteriormente, se obtiene el nRMSE dividiendo el RMSE entre el rango de los datos.

$$\text{nRMSE} = \frac{\text{RMSE}}{\max_i(y_i) - \min_i(y_i)} \quad (2.50)$$

- **Coefficiente de Determinación (R^2):** Indica qué tan fuerte es el ajuste o correlación lineal entre los valores predichos o calculados, de los valores reales o de referencia. Entre más cercano sea el valor a 1, mejor es el rendimiento del modelo. Para calcularlo, primero es necesario obtener la suma los errores cuadráticos, también conocida como suma de cuadrados residual (SS_{res}), y la suma de cuadrados total (SS_{tot}).

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.51)$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (2.52)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (2.53)$$

Capítulo 3

Metodología

Se utiliza la metodología propuesta David Ulrich y Steven Eppinger en el libro *Diseño y desarrollo de productos* [7]. Esta describe un proceso sistemático e iterativo para diseñar y desarrollar productos, cuyas etapas se muestran en la Figura 3.1. Cabe destacar que previo a ellas, existe una etapa de planeación inicial, donde se define el problema a solucionar, los objetivos a cumplir y el alcance general del proyecto, obteniendo así la declaración de la misión; los resultados de esta etapa ya fueron presentados anteriormente en el Capítulo 1. Asimismo, al final de cada etapa del proceso se debe realizar un análisis y reflexión de los resultados obtenidos, con el fin de detectar posibles sesgos o puntos de mejora, y hacer una nueva iteración de los pasos, de ser necesario. A continuación, se explica cada etapa del proceso.

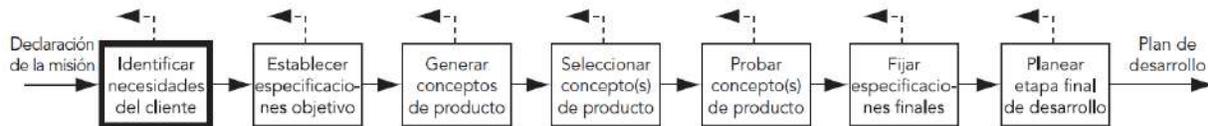


Figura 3.1: Etapas para el diseño y desarrollo de productos [7].

3.1. Identificación y determinación de necesidades

Esta etapa tiene como objetivo identificar el conjunto de requerimientos que se deben satisfacer para que la solución diseñada sea considerado como válida. Para lograr esto, se llevaron a cabo múltiples entrevistas presenciales y virtuales con profesores investigadores del proyecto IPTC. Luego, se interpretaron sus declaraciones para obtener la lista de necesidades y a cada una se les asignó un puntaje de importancia relativa, según la Tabla 3.1, lo cual permite identificar cuáles son críticas y cuáles pueden ser descartadas, en caso de ser necesario.

Tabla 3.1: Puntajes de importancia relativa para las necesidades.

Importancia relativa	Puntaje
La función es indeseable	1
La función no es importante	2
La función es deseable	3
La función es importante	4
La función es imprescindible	5

A continuación, en las Tablas 3.2, 3.3 y 3.4 se presentan las necesidades identificadas con su respectivo puntaje de importancia relativa.

Tabla 3.2: Necesidades y su importancia relativa. Parte 1.

No.	Necesidad	Imp.
1	El modelo es preciso para diferentes puntos de operación del panel solar en condiciones estándar	5
2	El modelo es aplicable a diferentes tipos de paneles solares	5
3	El modelo puede replicar el comportamiento del panel solar ante condiciones ambientales variadas	5
4	El modelo genera la hoja de datos con las respectivas curvas características varios paneles solares	4
5	El modelo se implementa en un software de código abierto	3
6	El modelo tiene un tiempo de ejecución bajo	3
7	El modelo requiere poca memoria	3
8	El modelo es aplicable a diferentes dimensiones de arreglos solares para CubeSats de 1 U	2
9	Los controladores estabilizan las señales de salida muy cerca de los valores esperados	5
10	Los controladores suprimen picos en las señales de salida cuando cambia el punto de operación	5
11	Los controladores estabilizan las señales de salida rápidamente	4
12	La plataforma permite al usuario introducir los niveles de irradiancia deseados	5
13	La plataforma permite al usuario introducir los niveles de temperatura deseados	5
14	La plataforma permite al usuario introducir los ángulos de incidencia deseados	5
15	La plataforma permite al usuario seleccionar el tipo de panel solar deseado	5
16	La plataforma permite al usuario controlar una variable de salida a la vez (tensión o corriente)	5
17	La plataforma permite al usuario iniciar, actualizar o detener una prueba	5
18	La plataforma se puede acceder desde una red de internet distinta a la del hardware	5

Tabla 3.3: Necesidades y su importancia relativa. Parte 2.

No.	Necesidad	Imp.
19	La solución de SAS puede recibir información desde la plataforma en poco tiempo	5
20	La interfaz permite visualizar los valores de salida medidos	5
21	La interfaz permite visualizar los valores de salida esperados	5
22	La interfaz cuenta con un tiempo de actualización bajo para los parámetros que monitorea	3
23	La interfaz crea una gráfica en función del tiempo con los valores de salida medidos	2
24	La interfaz permite visualizar el tiempo transcurrido desde que se inició una prueba	2
25	La plataforma permite realizar pruebas prolongadas sin el pago de una suscripción	5
26	La plataforma permite monitorear y controlar muchas variables sin el pago de una suscripción	5
27	La plataforma permite crear la interfaz de usuario deseada sin el pago de una suscripción	5
28	La solución diseñada se implementa en componentes de hardware presentes en el laboratorio	5
29	La plataforma IoT guarda los datos registrados en la nube	5
30	La solución de hardware guarda los datos registrados durante las pruebas en una memoria local	3
31	Se requiere de un manual de uso para operar el hardware	5
32	Se requiere de un manual de uso para operar el software del modelo analítico	5
33	Se requiere de un manual de uso para operar la plataforma IoT	5
34	Se requieren hojas de datos de varios paneles solares generadas por el modelo analítico	4
35	Se requiere de una guía con los pasos seguidos para implementar la plataforma IoT	3
36	El chasis tiene conectores externos para alimentación AC de 100 - 240 V a 50 / 60 Hz o DC de 12 - 30 V	5
37	El chasis tiene conectores externos para reprogramar los microcontroladores empleados	5
38	El chasis tiene conectores externos para leer los puertos seriales de los microcontroladores empleados	5
39	El chasis tiene conectores externos tipo banana para medir las entradas y salidas de la planta	5
40	El chasis tiene interruptores para seleccionar el tipo de alimentación o suspenderla	5
41	Los elementos externos del chasis están debidamente etiquetados	5

Tabla 3.4: Necesidades y su importancia relativa. Parte 3.

No.	Necesidad	Imp.
42	El chasis cuenta con una pantalla externa para visualizar los valores de entrada y salida de la planta	2
43	El chasis tiene ventilación adecuada que evita sobrecalentamiento de componentes internos	4
44	El chasis mantiene los componentes internos en una posición fija y estable	5
45	El chasis filtra la entrada de polvo en su interior	3
46	El chasis tiene un costo monetario de manufactura bajo	5
47	Los equipos y herramientas requeridos para manufacturar el chasis existen en laboratorios del TEC	4
48	El chasis puede ser ensamblado y desensamblado con un set básico de herramientas	4
49	El chasis puede ser manufacturado sin requerir de personal técnico de manufactura altamente calificado	3
50	Los materiales del chasis pueden limpiarse con agua y agentes desinfectantes comunes sin sufrir daños	5
51	Los materiales del chasis tienen un costo monetario bajo	5
52	Los materiales del chasis son suficientemente rígidos para soportar el peso de los componentes	5
53	Los materiales del chasis son resistentes a un amplio rango de temperaturas	4
54	El chasis es compacto	5
55	El chasis totalmente ensamblado es liviano	4
56	El chasis cuenta con elementos mecánicos que facilitan su transporte	2
57	Se requiere de planos mecánicos y de manufactura para el chasis	5
58	Se requiere de una lista de materiales y componentes para el chasis	5
59	Se requiere de un manual de ensamblaje para el chasis	4

Por otra parte, en la Tabla 3.5, las necesidades se organizan en una jerarquía que permite obtener un listado más reducido de necesidades generales o primarias, las cuales son desarrolladas en mayor detalle por necesidades secundarias.

Tabla 3.5: Jerarquización en necesidades primarias y secundarias.

No.	Primaria	Secundaria
1	Se emplea un modelo analítico para describir el comportamiento de un panel solar	1 a 8
2	Se emplean controladores digitales para regular las señales de salida del convertidor	9 a 11
3	La plataforma IoT permite el control remoto del hardware	12 a 19
4	La plataforma IoT cuenta con una interfaz para el monitoreo remoto del hardware	20 a 24
5	La plataforma IoT opera adecuadamente sin requerir el pago de una suscripción	25 a 27
6	La solución diseñada se implementa en hardware	28
7	La solución diseñada guarda los datos registrados durante las pruebas	29 y 30
8	La solución diseñada cuenta con documentación adecuada	31 a 35
9	El chasis para la solución diseñada cuenta con ciertos elementos externos	36 a 42
10	El chasis para la solución diseñada protege los componentes internos	43 a 45
11	El chasis para la solución diseñada presenta facilidades de manufactura y de ensamblaje	46 a 49
12	Los materiales del chasis para la solución diseñada cumplen ciertos requisitos	50 a 53
13	El chasis para la solución diseñada presenta facilidades para su transporte	54 a 56
14	El chasis para la solución diseñada presenta documentación adecuada	57 a 59

3.2. Establecimiento de especificaciones objetivo

Esta etapa tiene como fin cuantificar el grado de cumplimiento de las necesidades planteadas anteriormente. Para esto, se plantea una serie de métricas con sus respectivos valores objetivo, los cuales se presentan en las Tablas 3.6, 3.7, 3.8 y 3.9. Para establecer los valores ideales y marginales, se tomaron en cuenta las recomendaciones de los coordinadores basadas en su experiencia y en las características que requieren en la solución final, proyectos de graduación realizados previamente dentro del marco del IPTC, y revisiones bibliográficas. Además, en varias de las necesidades planteadas no existe una forma de cuantificar el grado de cumplimiento, simplemente se cumple o no se cumple con lo que indica la necesidad de manera explícita. Para tales casos, se utiliza una métrica binaria. En las Tablas 3.10, 3.11 y 3.12, se resumen los criterios utilizados para el establecimiento de los valores objetivo no binarios planteados anteriormente.

Tabla 3.6: Métricas y especificaciones establecidas. Parte 1.

No.	Métrica	Imp.	Unidad	Marginal	Ideal
1	Error porcentual absoluto medio (MAPE) respecto a valores de referencia	5	%	5	3
2	Número de paneles solares con los que se prueba que el modelo funciona adecuadamente	5	# de paneles	2	4
3	El modelo es sensible a cambios en las condiciones de irradiancia, temperatura y ángulo de incidencia	5	Binaria	Sí	Sí
4	Número de paneles a los que se les genera una hoja de datos exitosamente	4	# de paneles	2	4
5	El modelo es implementado en un software de código abierto	3	Binaria	Sí	Sí
6	Tiempo que le toma al modelo calcular la salida para un punto de operación dado	3	ms	100	50
7	Cantidad de memoria que requieren las funciones utilizadas por el modelo	3	KB	60	30
8	Cantidad de paneles solares que conforman el arreglo	2	# de paneles	6	6
9	Porcentaje de error en estado estacionario	5	%	<3	<1
10	Porcentaje de sobreimpulso	5	%	<15	<5
11	Tiempo de estabilización al 2%	4	ms	<100	<50
12	Rango de niveles de irradiancia en que el modelo funciona adecuadamente	5	W/m^2	0 - 1375	0 - 1375
13	Rango de niveles de temperatura en que el modelo funciona adecuadamente	5	°C	-80 a 80	-80 a 80
14	Rango de ángulos de incidencia en que el modelo funciona adecuadamente	5	°	0 a 90	0 a 90
15	Número de opciones de paneles solares que puede seleccionar el usuario	5	# de paneles	2	4
16	Se puede controlar el valor de tensión de salida o el valor de corriente de salida	5	Binaria	Sí	Sí

Tabla 3.7: Métricas y especificaciones establecidas. Parte 2.

No.	Métrica	Imp.	Unidad	Marginal	Ideal
17	Se puede iniciar, actualizar o detener una prueba cuando el usuario lo requiera	5	Binaria	Sí	Sí
18	Se puede acceder desde una red de internet distinta a la del hardware	5	Binaria	Sí	Sí
19	Tiempo que le toma al hardware recibir un mensaje desde la plataforma	3	s	10	0.5
20	Se pueden visualizar los valores de salida medidos (tensión, corriente, potencia)	5	Binaria	Sí	Sí
21	Se pueden visualizar los valores de salida esperados (tensión, corriente, potencia)	5	Binaria	Sí	Sí
22	Tiempo de actualización de los parámetros en la interfaz	3	s	10	0.5
23	Se crea una gráfica en función del tiempo con los valores de salida medidos	2	Binaria	Sí	Sí
24	Se puede visualizar el tiempo transcurrido desde que se inició una prueba	2	Binaria	Sí	Sí
25	Tiempo máximo de pruebas por día	5	hr	4	8
26	Número de variables que se pueden monitorear y controlar sin el pago de una suscripción	5	# de variables	20	40
27	Se puede crear la interfaz con las características deseadas sin el pago de una suscripción	5	Binaria	Sí	Sí
28	En la selección de componentes se consideran únicamente los disponibles en el laboratorio	5	Binaria	Sí	Sí
29	Los datos de las pruebas se guardan en la nube	5	Binaria	Sí	Sí
30	Los datos de las pruebas se guardan en una memoria local	3	Binaria	Sí	Sí
31	Existe un manual de uso del hardware	5	Binaria	Sí	Sí
32	Existe un manual de uso del software del modelo	5	Binaria	Sí	Sí
33	Existe un manual de uso de la plataforma IoT	5	Binaria	Sí	Sí

Tabla 3.8: Métricas y especificaciones establecidas. Parte 3.

No.	Métrica	Imp.	Unidad	Marginal	Ideal
34	Número de hojas de datos generadas	4	# de hojas	2	4
35	Existe una guía con los pasos de implementación de la plataforma IoT	3	Binaria	Sí	Sí
36	Existen conectores externos de alimentación AC y DC	5	Binaria	Sí	Sí
37	Todos los microcontroladores empleados cuentan con conectores para reprogramarlos	5	Binaria	Sí	Sí
38	Todos los microcontroladores empleados cuentan con conectores para leer sus puertos seriales	5	Binaria	Sí	Sí
39	Número de conectores externos tipo banana	5	# de conectores	4	4
40	Existen interruptores para seleccionar el tipo de alimentación y suspenderla	5	Binaria	Sí	Sí
41	Todos los elementos externos del chasis están debidamente etiquetados	5	Binaria	Sí	Sí
42	Existe una pantalla externa para visualizar los valores de entrada y salida	2	Binaria	Sí	Sí
43	Existe un método de ventilación en el chasis	4	Binaria	Sí	Sí
44	Todos los componentes internos cuentan con mecanismos para mantenerse en una posición fija	5	Binaria	Sí	Sí
45	Existe un mecanismo para filtrar polvo	3	Binaria	Sí	Sí
46	Costo total de manufactura del chasis	5	\$USD	<80	<50
47	Todos los equipos y herramientas considerados en el proceso de manufactura existen en el TEC	4	Binaria	Sí	Sí

Tabla 3.9: Métricas y especificaciones establecidas. Parte 4.

No.	Métrica	Imp.	Unidad	Marginal	Ideal
48	Todos los procedimientos de ensamblaje/desensamblaje se pueden realizar con un set básico de desatornilladores, alicates y pinzas	4	Binaria	Sí	Sí
49	Todos los procesos de manufactura se pueden realizar sin personal técnico altamente calificado	3	Binaria	Sí	Sí
50	Los materiales no se degradan con agua o agentes desinfectantes comunes	5	Binaria	Sí	Sí
51	Costo total de los materiales del chasis	5	\$USD	<80	<50
52	Factor de seguridad al hacer simulaciones de esfuerzos y deformaciones	5	adimensional	2	3
53	Temperaturas a las que se pueden ver expuestos los materiales del chasis	4	°C	0 a 60	0 a 70
54	Volumen del chasis	5	m^3	<0.5	<0.3
55	Peso del chasis totalmente ensamblado	4	kg	<10	<7
56	Existen elementos mecánicos que faciliten su transporte	2	Binaria	Sí	Sí
57	Existen planos mecánicos y de manufactura para el chasis	5	Binaria	Sí	Sí
58	Existe una lista de materiales y componentes para el chasis	5	Binaria	Sí	Sí
59	Existe un manual de ensamblaje para el chasis	4	Binaria	Sí	Sí

Tabla 3.10: Justificación de especificaciones establecidas. Parte 1.

No. Métrica	Justificación de valores objetivo
1	Se consultaron artículos [58], [19] en los que se compara el rendimiento de distintos modelos analíticos que describen el comportamiento de celdas solares. Estos presentan tablas donde, para cada modelo, se calculó el RSME y el MAPE al compararlos con valores de referencia. A partir de estos, se concluye que un MAPE $< 3\%$ es ideal y $< 5\%$ es aceptable.
2, 4, 15	Los coordinadores del proyecto IPTC solicitaron que la precisión del modelo fuera verificada con al menos 2 paneles solares comerciales para aplicaciones espaciales. Se considera como valor ideal el doble de paneles solares que el valor marginal.
6	Anteriormente se realizó un trabajo de graduación enfocado en la gestión de requerimientos para el IPTC [59], donde se establece que el sistema debe responder a cambios en las condiciones de tensión y corriente en un tiempo máximo de 500 <i>ms</i> . Dado que el tiempo que dura el modelo analítico en calcular el punto de operación es solo una fracción del tiempo total que toma procesar las solicitudes del usuario, enviar las señales de control a la planta del convertidor, y que los compensadores estabilicen las señales de salida, se requiere que el modelo analítico se ejecute en 1/5 del tiempo total disponible, como máximo.
7	Los principales candidatos a usar como microcontrolador central son el Arduino MEGA [60] y el Nucleo-6 [61], los cuales tienen 256 KB y 192 KB de memoria flash, respectivamente; se considera el caso más crítico donde la memoria disponible es de 192 KB. Como buena práctica, se asume que el programa no debe utilizar más del 80 % de la memoria disponible. Además, en el programa también se deben considerar otras funciones además del modelo analítico, como el procesamiento de señales provenientes de otros microcontroladores para el control de la planta. Por ende, al modelo se le asigna como máximo un 40 % de la memoria restante disponible, resultando en aproximadamente 60 KB. Se toma como valor ideal la mitad de este valor.

Tabla 3.11: Justificación de especificaciones establecidas. Parte 2.

No. Métrica	Justificación de valores objetivo
8	Los paneles solares comerciales para aplicaciones espaciales por lo general están dimensionados de forma tal que en cada cara de un CubeSat de 1 U, se puedan colocar dos de ellos [62], [63], [64], [65]. Si bien los CubeSats de 1U tienen 6 caras, solo es posible que 3 de ellas reciban radiación solar simultáneamente, y por ende 6 es el máximo de paneles solares que pueden generar potencia a la vez.
9, 10, 11	A partir de artículos en los que se proponen diseños de controladores digitales para convertidores DC-DC [66], [67], [68], y de trabajos previos realizados dentro del marco del IPTC [69], [70], se seleccionaron parámetros de control aceptables.
12	En un artículo donde se analizan las mediciones de radiómetros de irradiancia solar total en el espacio desde 1978 hasta 2010 [71], se puede observar que los valores máximos de irradiancia medidos no sobrepasan los $1375 W \cdot m^{-2}$; por ende, se toma este valor como máximo del rango. Además, dado que un SAS también debe considerar los periodos de eclipse que existen durante las misiones, el valor mínimo de irradiancia que se puede tener es de cero.
13	A partir de varios artículos donde se realizan análisis térmicos en los arreglos solares de satélites en distintas órbitas terrestres [72], [73], [74], se concluye que el rango típico de temperaturas se encuentra entre $-80 \text{ }^\circ\text{C}$ y $80 \text{ }^\circ\text{C}$.
14	Por trigonometría, se conoce que el rango de posibles ángulos de incidencia de los rayos solares sobre la superficie de la celda solar es de 0° a 90° .
22	Si bien se podrían lograr valores objetivo menores, se debe hacer un balance con respecto a los límites de volumen de datos que se pueden enviar a la plataforma IoT con una suscripción gratuita [75], [76], [77], [78]. Entre menor sea el tiempo de actualización de los datos en la interfaz, más rápido se llega a los límites establecidos y menor es la duración de las pruebas que se pueden monitorear. Por lo tanto, se considera como aceptable un tiempo de actualización de 1 s.
25	Las plataformas de IoT establecen límites en la cantidad de datos que se pueden transmitir por día [75]. Conociendo la cantidad de datos que se estarán enviando, su tiempo de actualización, y el límite de datos que se pueden transmitir por día, se calcula el tiempo aproximado que pueden tardar las pruebas antes de superar el límite de la plataforma.

Tabla 3.12: Justificación de especificaciones establecidas. Parte 3.

No. Métrica	Justificación de valores objetivo
26	En el Capítulo 7, se explican más detalladamente el número de variables que se desean controlar y monitorear. Se considera esto como una necesidad ya que algunas plataformas tienen límites en cuanto a la cantidad de variables que se pueden tener con la suscripción gratuita [77], [78].
39	El SAS tiene una entrada y una salida eléctricas, ambas DC, por lo que cada una requiere de un par de conectores. Se requiere que sean de tipo banana ya que son cables de uso típico en el laboratorio para tomar mediciones con multímetros, o conectarlos a fuentes de alimentación o cargas electrónicas programables.
46, 51, 52, 54, 55	Estos valores objetivo fueron establecidos a partir de recomendaciones de los coordinadores y de un trabajo de graduación previo enfocado en la gestión de requerimientos para el IPTC [59].
52	Esta métrica va enfocada en las partes del chasis que soportan el peso de los componentes electrónicos. Dado que la MCP del proyecto IPTC está conformada por tres módulos de potencia, a futuro se podría necesitar colocar tres plantas de convertidores DC-DC, una encima de la otra. Asumiendo que la mitad del peso de los componentes proviene de la planta, se requeriría entonces, como mínimo, un factor de seguridad de 2. Se toma como valor ideal un factor de seguridad igual o mayor a 3, en caso de que el peso de la planta sea mayor al asumido
53	Para el límite superior, se asume como caso más crítico aquel en el que un IC caliente hace contacto directo con materiales del chasis; relacionado directamente con la métrica 42. Para el límite inferior, se toma en cuenta las condiciones operacionales establecidas para el proyecto IPTC en [59].

3.3. Generación de conceptos

Esta etapa tiene como objetivo proponer diferentes candidatos de solución al problema existente. Por lo tanto, el primer paso consiste en realizar un análisis profundo del problema con el fin de realizar una descomposición funcional del mismo, obteniendo así un conjunto de subproblemas. A partir de esto, se puede construir un diagrama de bloques básico, como el de la Figura 3.2, el cual en capítulos posteriores será desarrollado en mayor detalle. En este, se tiene una entrada de energía, correspondiente a la alimentación eléctrica AC de 100-240 V a 50/60 Hz, y señales de entrada correspondientes a los comandos introducidos por el usuario desde una plataforma IoT. Por otra parte, se tiene una salida de energía correspondiente al valor DC regulado según el punto de operación del arreglo solar introducido por el usuario, y las señales de salida corresponden a los datos que se envían a la plataforma IoT, tales como los valores medidos en la salida del convertidor.

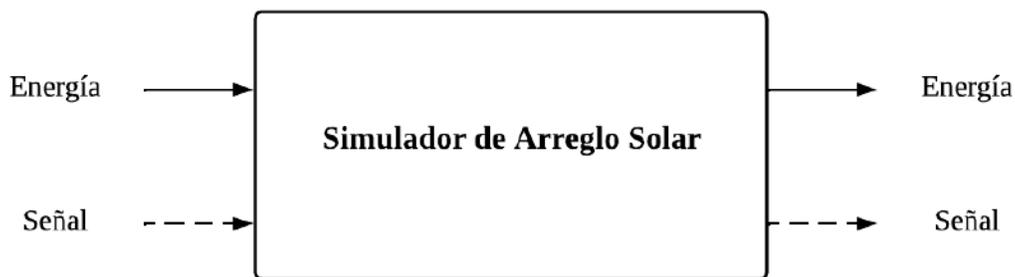


Figura 3.2: Diagrama de bloques. Adaptado de [7].

Una vez que se cuenta con la descomposición funcional del sistema, se debe realizar una búsqueda interna, relacionada con los conocimientos, creatividad y pensamiento crítico del ingeniero, y externa, lo cual requiere búsqueda exhaustiva de información proveniente de la literatura, patentes, expertos y de análisis comparativos contra posibles soluciones existentes en el mercado que asemejen a lo que se desea lograr. Esto permite la generación de una gran cantidad de conceptos, que se pueden organizar a partir de herramientas como árboles de clasificación y tablas de combinación.

3.4. Selección de conceptos

Con el fin de seleccionar cuáles de los conceptos propuestos pueden llegar a ser parte de la solución final de manera objetiva y evitando posibles sesgos. Para esto, se emplean las etapas de filtrado y de evaluación de conceptos, en las cuales se utilizan matrices de decisión utilizando criterios basados en las necesidades planteadas anteriormente. No obstante, la forma en que se implementan estas matrices varían levemente en ambas etapas.

En la etapa de filtrado, se busca descartar los conceptos claramente desfavorables, o bien hacer una combinación de dos o más conceptos. Para lograr esto, en la matriz de selección

se elige un concepto como referencia y los demás son evaluados con respecto a este según la siguiente simbología, la cual permite cuantificar el rendimiento neto de los conceptos y ordenarlos del más al menos apto.

- 0 : el rendimiento del concepto en el criterio evaluado es equivalente al del concepto de referencia
- + : el rendimiento del concepto en el criterio evaluado es superior al del concepto de referencia
- - : el rendimiento del concepto en el criterio evaluado es inferior al del concepto de referencia

En la etapa de filtrado, se le asigna un peso ponderado a cada criterio, tomando como base la importancia relativa asignada a cada necesidad. Nuevamente, se elige un concepto como referencia y los demás conceptos son evaluados según la siguiente escala numérica, la cual permite obtener una puntuación ponderada del rendimiento cada concepto y así elegir el ganador.

- 1 : el rendimiento del concepto en el criterio evaluado es notablemente menor al del concepto de referencia
- 2 : el rendimiento del concepto en el criterio evaluado es ligeramente menor al del concepto de referencia
- 3 : el rendimiento del concepto en el criterio evaluado es equivalente al del concepto de referencia
- 4 : el rendimiento del concepto en el criterio evaluado es ligeramente superior al del concepto de referencia
- 5 : el rendimiento del concepto en el criterio evaluado es notablemente superior al del concepto de referencia

3.5. Desarrollo del concepto

Una vez que se ha seleccionado el concepto ganador, se procede con el desarrollo del mismo para implementar un prototipo funcional.

3.6. Pruebas de concepto

Esta etapa tiene como objetivo validar que el prototipo funcional desarrollado cumple con las necesidades planteadas. Para esto, es necesario desarrollar diferentes experimentos que permitan cuantificar el rendimiento del prototipo y compararlo con las especificaciones planteadas.

Capítulo 4

Propuesta general de diseño

En este capítulo, se presenta una propuesta general de diseño, la cual se subdivide en tres componentes principales: solución de SAS, de IoT, y de chasis. Basado en las necesidades presentadas en el Capítulo 3, se plantea un diagrama de bloques con los componentes principales de la solución, como se observa en la Figura 4.1, el cual en capítulos posteriores se toma como base para la descomposición funcional en subproblemas.

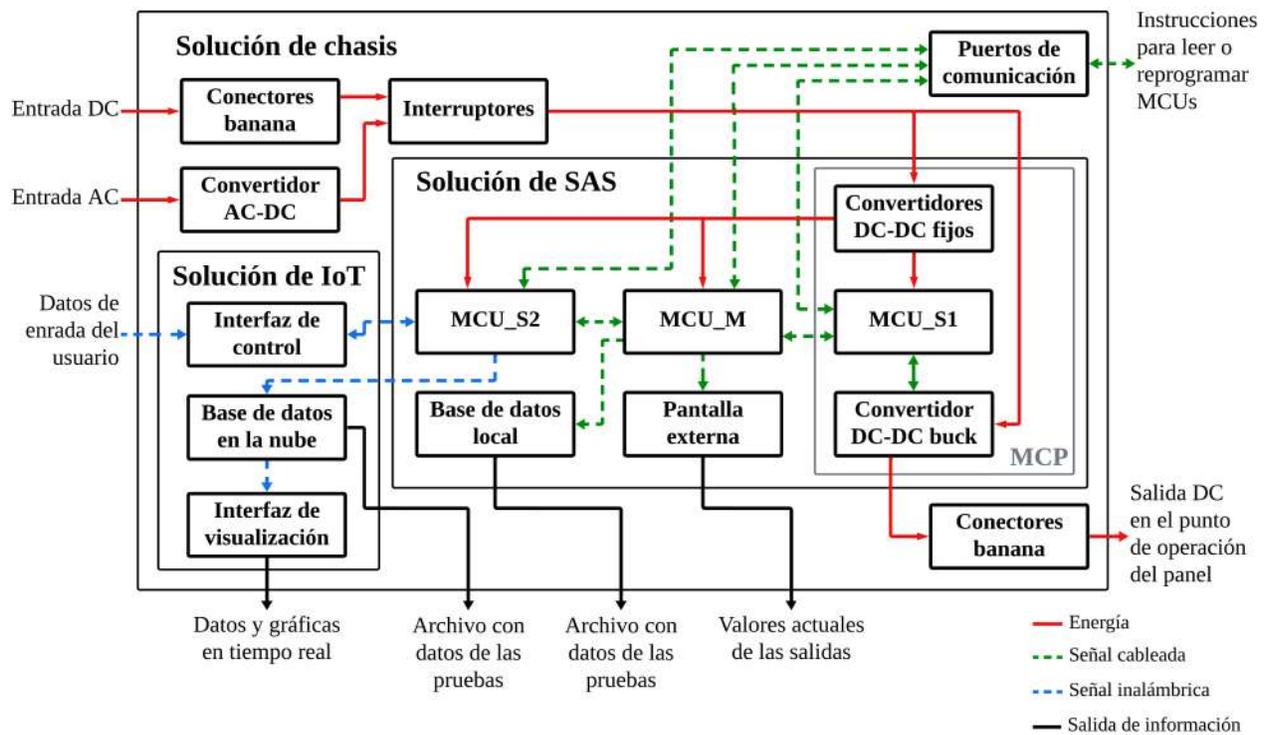


Figura 4.1: Diagrama de bloques general.

Cabe destacar que este diagrama contempla la interacción entre todos los subsistemas de un eventual producto final. No obstante, durante el desarrollo del presente proyecto de graduación, no se cuenta con un prototipo funcional de la planta MCP, ya que la planta existente en el laboratorio presenta problemas que impiden su uso. Actualmente otros investigadores dentro del proyecto IPTC se encuentran en la fase de diseño de una nueva versión de la planta para eventualmente implementar en el producto final. Por otra parte, tampoco se pretende proceder con la manufactura del chasis hasta contar con la placa PCB de esta nueva versión de la planta. Por lo tanto, el diseño del chasis se hará basado en las dimensiones y demás características de la versión actual de la planta MCP, y la validación de los indicadores se hace a través de simulaciones. Debido a estos dos factores, más adelante se presentan variaciones del anterior diagrama, que corresponden a las plataformas utilizadas para validar el cumplimiento de los indicadores de las soluciones de SAS e IoT a través de pruebas experimentales.

A continuación, se describen los principales componentes del diseño propuesto en la Figura 4.1:

- **Convertidor AC-DC:** Regula la entrada AC de 110 - 240 V a 50 / 60 Hz a un valor DC fijo por definir.
- **Interruptores:** Deben permitir al usuario encender o apagar el equipo, y seleccionar si la alimentación proviene de la salida del convertidor AC-DC, o de una alimentación DC externa a proveniente de conectores de tipo banana en el exterior del chasis.
- **Convertidores DC-DC fijos:** Regulan la alimentación DC que deje pasar el interruptor a 3.3 V y 5 V fijos, con el fin de que puedan usarse para alimentar a los microcontroladores. Se encuentran implementados en la planta del MCP.
- **Convertidor DC-DC reductor:** toda su fase de potencia se encuentra implementada en la planta del MCP. Esta, además, tiene sensores de corriente y tensión incorporados. Se alimenta de la señal DC que deja pasar el interruptor selector. Se tiene acceso a su salida a través de conectores banana, con el fin de que se les pueda conectar una carga o que se puedan medir con un multímetro. Es necesario diseñar controladores digitales de tensión y corriente.
- **MCU_S1:** Se encuentra presente en la planta del MCP y corresponde a un PIC16F1786 [79]. En este se implementan los controladores digitales, los cuales toman como realimentación las mediciones de los sensores de corriente y tensión. Mantiene una comunicación bidireccional con el MCU_M, del cual recibe el punto de operación deseado, y le envía la información leída de los sensores.
- **MCU_S2:** Tiene acceso a internet, por lo que se encarga de recibir las entradas introducidas por el usuario en una plataforma IoT y las envía al MCU_M. A dicha plataforma IoT envía las mediciones de los sensores para que se muestren en una interfaz gráfica. Además, durante las pruebas guarda de forma periódica el valor de las variables de interés en una base de datos en la nube, en archivos en formato .CSV o .JSON.

- **MCU_M:** Es el cerebro central de todo el diseño. Mantiene una comunicación bidireccional con los otros MCUs. Como ya se comentó, envía al MCU_S1 el punto de operación deseado en la salida del convertidor DC-DC *buck*, y recibe información de los sensores, la cual luego envía al MCU_S2. Además, del MCU_S2 recibe los datos de entrada del usuario. Tiene conectada una memoria flash externa donde guarda de manera periódica el valor de las variables de interés en un archivo de texto; esto para que sirvan como respaldo en caso de que el MCU_S2 pierda su conexión a internet y con ello el acceso a la base de datos en la nube. Asimismo, tiene conectada una pantalla, en la cual muestra los valores de salida del convertidor *buck* (tensión, corriente y potencia).

También es necesario aclarar qué se entiende exactamente por cada una de las tres componentes de la solución:

- **Solución de SAS:** Abarca la implementación de un modelo analítico que describe el rendimiento de arreglos solares, el diseño e implementación de un algoritmo para procesar la información proveniente del modelo analítico y enviar comandos a la planta del convertidor *buck*, y el diseño e implementación de controladores digitales para regular la tensión y corriente de salida de dicha planta. Además, las señales de salida deben poder visualizarse en una pantalla externa, y los datos registrados durante la operación del SAS deben guardarse en una memoria local.
- **Solución de IoT:** Abarca la selección de una plataforma IoT que permita al usuario acceder a una interfaz de control en la que se puedan modificar parámetros que afecten el comportamiento del sistema, así como una interfaz de visualización donde se puedan observar los valores actuales de las salidas del sistema, y su variación a través del tiempo de manera gráfica. Además, los datos registrados durante las pruebas deben guardarse en una base de datos en la nube.
- **Solución de chasis:** Abarca el diseño mecánico y el plan de manufactura de un chasis que contenga y proteja los componentes de hardware de las soluciones de SAS e IoT, además de contar con interruptores, conectores, puertos y una pantalla que permitan al usuario interactuar con el hardware desde el exterior. También se requiere de un diagrama eléctrico básico para interconectar los diferentes componentes y una guía de ensamble.

4.1. Selección de hardware

Tal y como se aprecia en las necesidades, es de importancia crítica que para la selección de los componentes de hardware que requiere la solución, se consideren solo opciones ya disponibles en el laboratorio.

4.1.1. Microcontroladores

En el caso de MCU_M, dado que debe contar con al menos dos puertos de comunicación serial UART, las opciones se reducen a una placa de desarrollo NUCLEO-64 de STMicroelectronics [8], [80], y un Mega 2560 de Arduino [81], [82], [83], [84]. En la Figura 4.2 se presenta una comparación de las especificaciones técnicas y el precio de cada placa, la cual se toma como base para crear la matriz de selección de la Figura 4.3, en la que se consideran varios criterios con sus respectivos pesos ponderados. Se aprecia que el claro candidato ganador a MCU_M es el NUCLEO-64.

Especificación	ST NUCLEO-64	Arduino Mega 2560 Rev3
CPU	STM32L073RZ	ATmega2560
Familia	32-bit ARM Cortex-M0+	8-bit AVR RISC
Frecuencia de reloj	32 MHz	16 MHz
Flash	192 KB	256 KB
SRAM	20 KB	8 KB
EEPROM	6 KB	4 KB
I/O	84	85
Puertos UART	4	4
Puertos SPI	6	1
Puertos I2C	3	1
Dimensiones	82.5 mm x 70 mm	101.5 mm x 53.3 mm
Peso	77 g	37 g
Precio	\$13.83	\$48.40

Figura 4.2: Especificaciones de candidatos a MCU_M.

Criterios de evaluación	Peso (%)	MCU_M			
		ST Nucleo-64		Arduino Mega 2560	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Velocidad de procesamiento	25	3	0.75	1	0.25
Memoria Flash	20	3	0.6	4	0.8
Memoria SRAM	20	3	0.6	1	0.2
Puertos seriales	10	3	0.3	2	0.2
Área	5	3	0.15	4	0.2
Peso	5	3	0.15	5	0.25
Precio	15	3	0.45	1	0.15
Puntaje			3.00		2.05
Lugar			1		2

Figura 4.3: Matriz de selección de MCU_M.

En el caso del MCU_S2, se requiere que pueda acceder de manera inalámbrica a Internet y que pueda mantener una comunicación bidireccional con el MCU_M a través de un puerto serial UART. Las opciones disponibles en el laboratorio, son una placa de desarrollo NodeMCU ESP8266 [9], [85], [86] y un Nano 33 IoT de Arduino [87], [88], [89]. En la Figura 4.4 se presenta una comparación de las placas, y posteriormente se realiza la matriz de selección de la Figura 4.5, en la que el claro candidato ganador a MCU_S2 es el NodeMCU ESP8266.

Especificación	NodeMCU ESP8266	Arduino Nano 33 IoT
CPU	Tensilica Xtensa LX106	SAMD21
Familia	32-bit RISC	32-bit ARM Cortex-M0+
Frecuencia de reloj	80 MHz	48 MHz
Flash	4 MB	256 KB
SRAM	64 KB	32 KB
Puerto UART	1	1
Puerto SPI	1	1
Puerto I2C	1	1
Dimensiones	49 mm x 26 mm	45 mm x 18 mm
Peso	20 g	5 g
Precio	\$9.95	\$25.50

Figura 4.4: Especificaciones de candidatos a MCU_S2.

Criterios de evaluación	Peso (%)	MCU_S2			
		Arduino Nano 33 IoT		NodeMCU ESP8266	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Velocidad de procesamiento	25	3	0.75	5	1.25
Memoria Flash	20	3	0.6	5	1
Memoria SRAM	20	3	0.6	5	1
Puertos seriales	10	3	0.3	3	0.3
Área	5	3	0.15	2	0.1
Peso	5	3	0.15	1	0.05
Precio	15	3	0.45	5	0.75
Puntaje			3.00		4.45
Lugar			2		1

Figura 4.5: Matriz de selección de MCU_S2.

En la Figura 4.6, se presentan imágenes de los candidatos ganadores a MCU_M y MCU_S2.

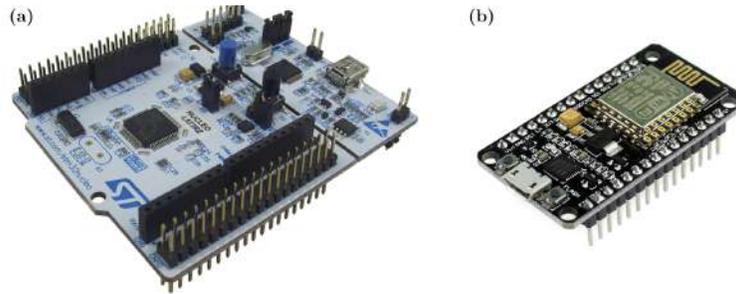


Figura 4.6: (a) NUCLEO-64 L073RZ [8] y (b) NodeMCU ESP8266 [9]

4.1.2. Convertidor AC-DC

En el año 2022, se compararon dos modelos de convertidores de conmutación Mean Well de la serie IRM-60 [10] para el proyecto IPTC, los cuales toman una entrada AC de 100 V a 240 V a una frecuencia de 50 a 60 Hz, y la convierten a una salida DC de 12 V a 5 A, en el caso del IRM-60-12, o de 24 V a 2.5 A, en el caso del IRM-60-24; ambos ofreciendo una potencia máxima de 60 W. Para elegir la mejor opción, se deben considerar los casos más críticos de tensión y de corriente que se pueden tener. Si bien el alcance de este proyecto se limita a simular solamente arreglos solares de CubeSats de 1U, a futuro se plantea ampliarlo hasta 3U, por lo que para la selección del convertidor se deben tomar en cuenta ambos casos.

Se realizó una selección de cuatro paneles solares para aplicaciones espaciales, en los cuales se tiene que los valores máximos de tensión, corriente y potencia, corresponden a 3.375 V, 0.548 A y 1.266 W, respectivamente. Los criterios utilizados para seleccionar dichos paneles se detallan en el Capítulo 5. Además, en un CubeSat de 1U se podrían tener hasta tres de estos paneles, conectados en serie o en paralelo, recibiendo irradiancia solar simultáneamente, mientras que en uno de 3U se tendrían 9. Por otra parte, el ángulo de incidencia de los rayos solares sobre la superficie de los tres paneles en que se logra la máxima potencia, corresponde a 54,7°. En cuanto a la potencia, cuando las tres caras del CubeSat tienen un ángulo de incidencia de 54,7°, se multiplica la potencia de salida por un factor de 1,73 al compararlo con un solo panel con un ángulo de incidencia de 0° [41]. Por lo tanto, se tienen los siguientes casos más críticos de tensión (todos los paneles en serie), corriente (todos los paneles en paralelo), y potencia, tanto para 1U, como para 3U.

$$V_{max,1U} = 3,375 \text{ V} \cdot 3 \cdot \cos(54,7^\circ) = 11,702 \text{ V}$$

$$V_{max,3U} = 3,375 \text{ V} \cdot 9 \cdot \cos(54,7^\circ) = 17,552 \text{ V}$$

$$I_{max,1U} = 0,548 \text{ A} \cdot 3 \cdot \cos(54,7^\circ) = 1,900 \text{ A}$$

$$I_{max,3U} = 0,548 \text{ A} \cdot 9 \cdot \cos(54,7^\circ) = 2,850 \text{ A}$$

$$P_{max,1U} = 1,266 \text{ W} \cdot 3 \cdot 1,73 = 13,137 \text{ W}$$

$$P_{max,3U} = 1,266 \text{ W} \cdot 9 \cdot 1,73 = 19,706 \text{ W}$$

Se puede observar que el Mean Well IRM-60-12 no cumple con la tensión máxima que se puede encontrar en un CubeSat de 3U. No obstante, el Mean Well IRM-60-24 sí cumple satisfactoriamente todos los casos críticos, y por lo tanto es el candidato escogido como convertidor AC-DC de la solución. Este se muestra en la Figura 4.7.



Figura 4.7: Convertidor de conmutación AC-DC Mean Well IRM-60-24 [10].

4.1.3. Pantalla externa

La función principal de la pantalla es mostrar los cambios de tensión, corriente y potencia medidos a la salida de la planta, con al menos dos decimales de precisión. No obstante, también es deseable que puedan mostrar caracteres alfanuméricos para notificar al usuario si existe algún problema, como por ejemplo que no se haya podido establecer conexión a internet o no se hayan podido recibir los parámetros de entrada del usuario. También es importante minimizar el precio, haciendo un balance entre la cantidad de caracteres que los candidatos pueden representar, y el área que ocupan.

En el laboratorio existen múltiples LEDs de 7 segmentos de un dígito y un punto decimal [90], [91], por lo que, según los valores máximos calculados anteriormente para la tensión, corriente y potencia, se necesitarían de al menos once de estos. También existen pantallas LCD de 16 por 2 caracteres (LCD1602) [11], [92], y de 20 por 4 caracteres (LCD2004) [93], [94]. En el caso de la LCD1602, esta incluye un módulo adicional que permite establecer comunicación con un MCU mediante el protocolo I2C, lo cual reduce significativamente el cableado necesario y facilita el prototipado de una placa de desarrollo en el que se tengan que soldar todas las vías manualmente.

En la Tabla 4.1 se realiza una comparación entre los candidatos a partir de los criterios de interés, y en la Figura 4.8, se presenta una matriz de selección.

Tabla 4.1: Comparación de candidatos a pantalla externa.

Característica	LEDs de 7 seg.	LCD1602 I2C	LCD2004
Precio (\$USD)	16.5	8.1	12.95
Área (cm ²)	26.3	28.8	58.8
Cantidad de caracteres	22	32	80
Alfanumérico	No	Sí	Sí

Criterios de evaluación	Peso (%)	Pantalla externa					
		LEDs de 7 segmentos		LCD1602 con I2C		LCD2004 (Ref.)	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Precio	45	1	0.45	4	1.8	3	1.35
Área	25	4	1	5	1.25	3	0.75
Cantidad de caracteres	20	1	0.2	2	0.4	3	0.6
Alfanumérico	5	1	0.05	3	0.15	3	0.15
Facilidad de cableado	5	1	0.05	5	0.25	3	0.15
Puntaje			1.75		3.85		3.00
Lugar			3		1		2

Figura 4.8: Matriz de selección para pantalla externa.

Se concluye que el candidato ganador es el LCD1602 con módulo I2C, el cual se muestra en la Figura 4.9.

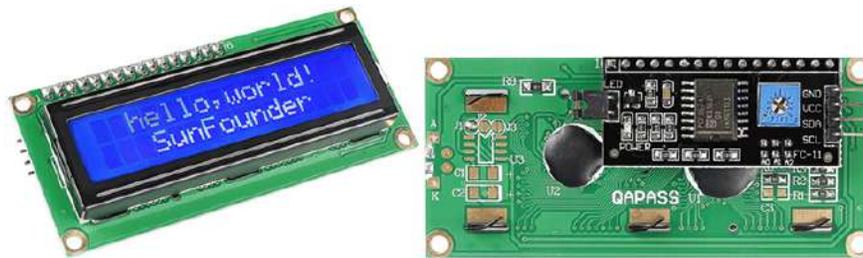


Figura 4.9: Pantalla LCD1602 de SunFounder [11].

4.1.4. Base de datos local

Se requiere seleccionar un componente que pueda guardar los registros de datos periódicos de múltiples pruebas diferentes en archivos de texto, sin la necesidad de tener que vaciarla frecuentemente por falta de memoria. Además, es deseable que los datos se puedan acceder desde un ordenador de manera rápida y simple.

Para calcular cuál es la capacidad de memoria mínima requerida, se considera que se realizan pruebas al menos 8 horas por día y se asume que se añaden registros nuevos cada 500 ms. Se tiene entonces que, al final del día, se tendrían 57600 líneas de registros. Se confecciona entonces un archivo en formato .TXT que contenga una línea de encabezado y 57600 líneas de registros, que tiene un peso de 8.95 MB y se puede encontrar en el repositorio de GitHub facilitado en el Anexo B. Si se asume que el equipo se debe poder usar durante los 7 días de la semana, durante al menos 4 semanas antes de que se acabe la capacidad de memoria disponible, se tiene entonces que dicha capacidad debe ser superior a 250.6 MB, o 2004.8 Mbit.

Existen opciones comerciales de módulos de memoria externa como EEPROM [95], FRAM [96] o Flash seriales [97], que se pueden conectar a los microcontroladores fácilmente a través de protocolos como SPI o I2C. No obstante, en todas estas la capacidad de memoria que presentan está por debajo de los 128 Mbit. Por lo tanto, debido al requisito de memoria antes calculado, dichas opciones quedan descartadas de manera directa.

Por otra parte, se tienen los lectores de tarjetas SD, un tipo de memoria flash que se encuentra en presentaciones desde los 2 GB en tarjetas SDHC, hasta incluso 2 TB en tarjetas SDXC [98]. Además, estas son fácilmente extraíbles y muchos ordenadores tienen lectores incorporados, o bien, se pueden adquirir lectores externos con interfaz USB por un bajo precio [99], permitiendo leer sus contenidos de manera rápida y simple. Por lo tanto, se concluye que este es el candidato ganador para base de datos local.

En el laboratorio se encuentra disponible una *Micro SD Card Breakout Board+* de Adafruit, y varias tarjetas Kingston Micro SDHC de 8 GB [100]. Dicha placa se puede comunicar con el MCU_M a través del protocolo SPI, y se muestra en la Figura 4.10.

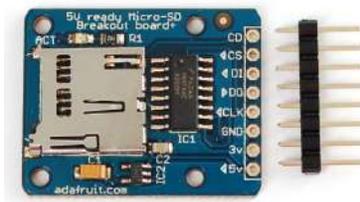


Figura 4.10: Placa para tarjeta Micro SD de Adafruit [12].

Reloj en tiempo real (RTC)

Una necesidad implícita para la base de datos local, es contar con acceso a la fecha y hora actual cada vez que se añada un nuevo registro. La placa NUCLEO-64 L073RZ, seleccionada como candidato ganador a MCU_M, cuenta con un RTC interno que se puede utilizar para suplir esta necesidad. No obstante, al conducir una prueba en la cual se estuvieron tomando registros periódicos, luego de 15 horas, a las 7:58:38 a.m., el RTC daba como hora actual las 9:43:05 a.m., es decir, se había adelantado aproximadamente 1 hora con 44 minutos; las evidencias se pueden encontrar en el repositorio de GitHub facilitado en el Anexo B. Si bien el RTC no desempeña una función crítica que pueda afectar a otros componentes del sistema, es deseable que sea más preciso. Por lo tanto, se consultó entre los componentes disponibles en el laboratorio la disponibilidad de algún módulo RTC externo, y se encontró un DS3231 [13], el cual es un módulo de alta precisión con una interfaz I2C. Cuenta además con una batería CR2032 que le permite continuar manteniendo la hora y fecha incluso si se retira la alimentación principal del sistema, y ofrece dos alarmas configurables, las cuales pueden usarse para activar interrupciones en el MCU_M de ser requerido (no es el caso para el desarrollo de este proyecto de graduación, pero puede ser útil a futuro). Es por esto que se decidió incorporar el módulo RTC DS3231 en la placa prototipo, el cual se muestra en la Figura 4.11.



Figura 4.11: Módulo de RTC DS3231. [13].

4.1.5. Interruptores

En cuanto a los interruptores, en el laboratorio se tienen de tipo *rocker* y rotatorios. Por lo tanto, para seleccionar la alimentación que el usuario desee, se utiliza un interruptor rotatorio LA38-20X3 [14], mostrado en la Figura 4.12.a, el cual permite girar una perilla y seleccionar uno de dos estados. Por otra parte, se utiliza un interruptor *rocker* KCD3 [15], mostrado en la Figura 4.12.b, que permite encender o apagar el equipo.



Figura 4.12: (a) Interruptor rotatorio [14] y (b) *rocker* [15].

4.1.6. Puertos de comunicación

Se tienen dos conectores USB 2.0 hembra a hembra con montaje en panel [16], mostrados en la Figura 4.13.a, los cuales permiten que los microcontroladores MCU_M y MCU_S2 se puedan conectar a un ordenador para leer sus puertos seriales o reprogramarlos con un cable USB macho a macho.

En cuanto al MCU_S1, dado que se trata de un PIC, para reprogramarlo o leer su puerto serial UART, se utiliza un PICKit 3 de Microchip Technology [101], que se conecta al microcontrolador con seis cables de tipo *jumper* y se conecta a un ordenador con un cable USB, como se observa en la Figura 4.13.b. Por lo tanto, en el exterior del chasis se deciden colocar cabezales de *jumpers* [18] como el mostrado en la Figura 4.13.c.

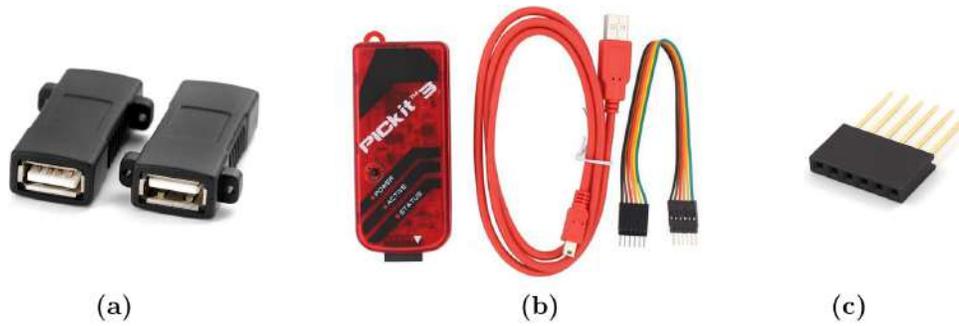


Figura 4.13: (a) Conectores USB hembra a hembra [16], (b) PICkit 3 [17], y (c) cabezal de *jumpers*. [18].

4.2. Plataformas para pruebas experimentales

Como se mencionó anteriormente, durante el desarrollo del presente proyecto de graduación, no se cuenta con una planta MCP funcional, y por consiguiente, tampoco se cuenta con un convertidor DC-DC reductor síncrono. A pesar de esto, se debe realizar el diseño teórico de controladores digitales para regular la tensión y corriente de salida del convertidor, los cuales se validan mediante simulaciones. [66]

Por otra parte, en el laboratorio sí existen instrumentos que pueden ser controlados utilizando las soluciones de SAS e IoT desarrolladas para poder emular las salidas de distintos arreglos solares. Estos corresponden a una fuente de poder programable RIGOL DP832 [102], cuya salida se conecta a una carga electrónica programable RIGOL DL3021 [103]. Sin embargo, el control de estos se puede hacer únicamente a través de un ordenador, por lo que no es posible incorporar el uso de microcontroladores y demás componentes electrónicos complementarios, como el RTC externo, el lector de tarjetas Micro SD y la pantalla LCD1602.

Por lo tanto, se diseñaron dos plataformas separadas para realizar las pruebas experimentales necesarias para validar el cumplimiento de todos los indicadores. En la Figura 4.14, se presenta el diagrama de bloques de la plataforma #1, la cual incorpora los instrumentos de laboratorio que permiten emular las salidas deseadas de arreglos solares. En la Figura 4.15, se presenta el diagrama de bloques de la plataforma #2, la cual permite validar la correcta interacción entre los microcontroladores y demás componentes electrónicos, asemejándose más a lo esperado en el producto final de la Figura 4.1. Cabe mencionar que se confeccionó una placa de desarrollo donde se soldaron manualmente los componentes electrónicos indicados; esta se muestra y explica con más detalle en capítulos posteriores. Por otra parte, dado que el MCU_S1 del producto final se encuentra ensamblado directamente en la planta MCP, a la cual no se tiene acceso, en ese caso se utiliza un microcontrolador Arduino UNO [104] que únicamente se encarga de recibir los puntos de operación enviados por el MCU_M, y se los devuelve con un poco de ruido, con el fin de emular la interacción esperada en el producto final, donde el MCU_M envía los puntos de operación al MCU_S1, este regula las salidas del convertidor y le devuelve al MCU_M las lecturas de los sensores de corriente y tensión incorporados en la planta. Finalmente, cabe mencionar que el convertidor AC-DC en este caso se trata de un convertidor genérico de 120 V @ 60 Hz a 5 VDC, el cual alimenta al Arduino UNO y este luego alimenta a los demás componentes electrónicos.

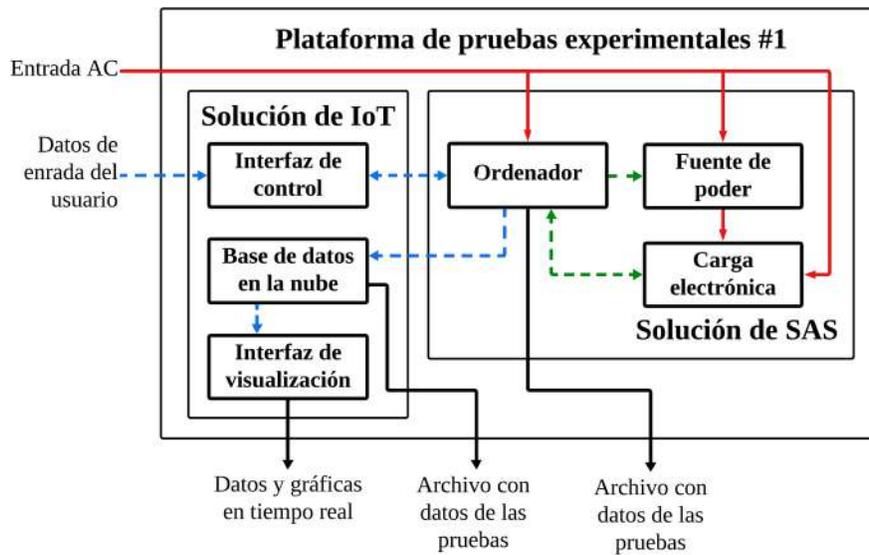


Figura 4.14: Plataforma de SAS #1.

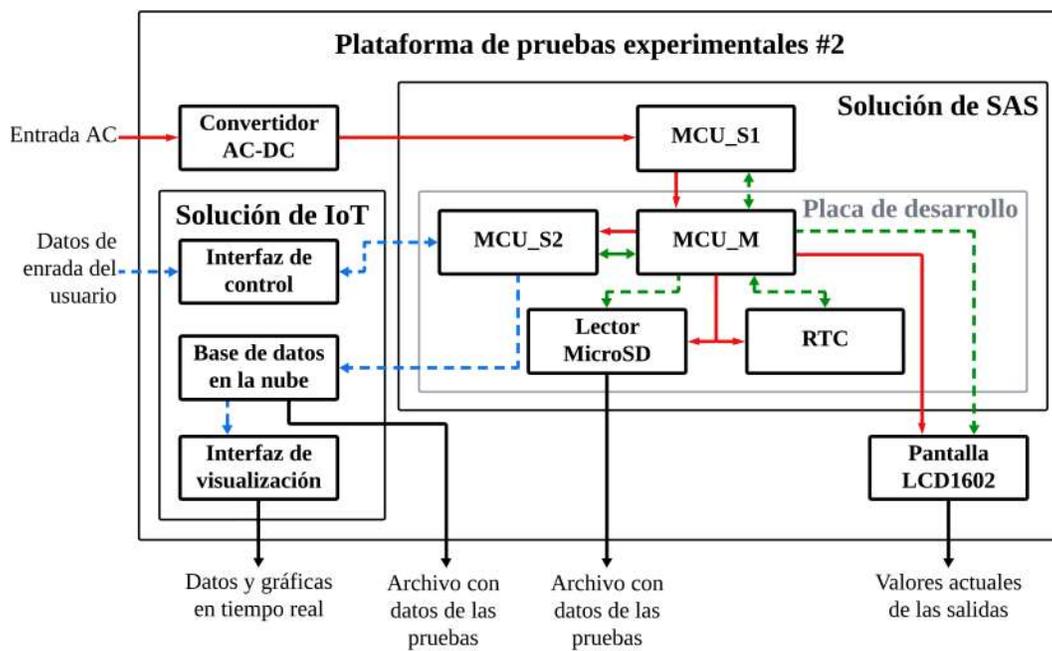


Figura 4.15: Plataforma de SAS #2.

Capítulo 5

Solución de Simulador de Arreglo Solar

En este capítulo se abarcan las fases de la metodología relacionadas con la generación, selección y desarrollo de conceptos para las diferentes componentes de la solución de SAS. También se lleva a cabo una validación parcial de la solución, ya que algunas componentes requieren ser integradas con la solución de IoT y por ende, su validación se realiza en el Capítulo 9.

Como se mencionó en la propuesta general del Capítulo 4, se diseñaron dos plataformas de pruebas separadas con el fin de facilitar la validación de todos los indicadores requeridos, por lo que ahora se debe proceder con la descomposición funcional de cada una.

En la Figura 5.1 se presenta la descomposición funcional de la plataforma de SAS #1.

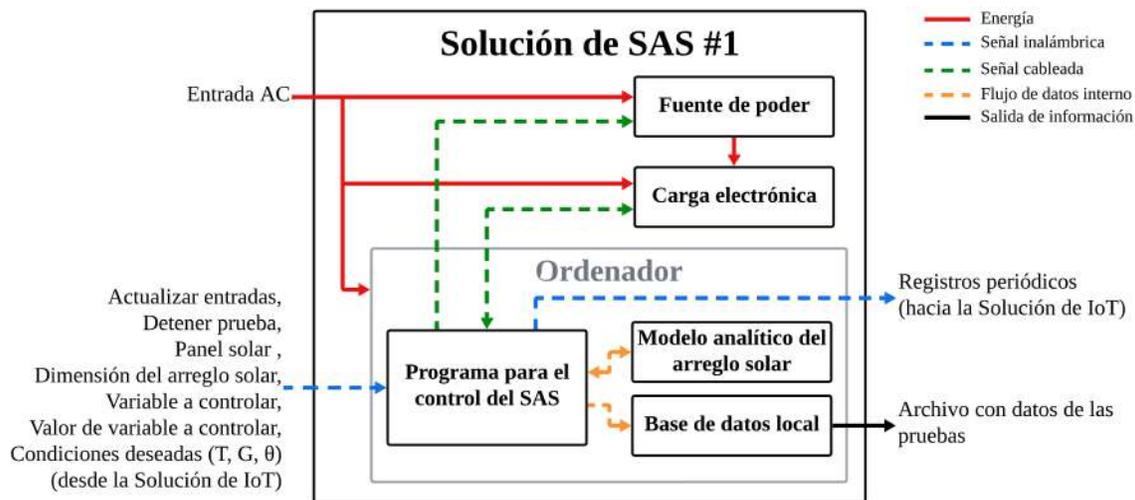


Figura 5.1: Descomposición funcional de la plataforma de SAS #1.

En este caso, se debe seleccionar e implementar un modelo matemático que pueda solucionarse de manera analítica, y que permita al usuario introducir valores de entrada y obtener como resultado el punto de operación (tensión y corriente) que tendría un arreglo solar real.

Además, se debe escribir un programa que permita al ordenador interactuar con la solución IoT para recibir múltiples entradas del usuario, y con base en ellas realizar las acciones correspondientes. Estas consisten en utilizar el modelo analítico para obtener el punto de operación deseado por el usuario y configurar los instrumentos del laboratorio para obtener dicho punto en sus salidas. Además, debe tomar registros periódicos que se guardan en una base de datos local dentro del mismo ordenador, y también enviarlos a la solución IoT para que se guarden en una base de datos en la nube.

En la Figura 5.2 se presenta la descomposición funcional de la plataforma de SAS #2.

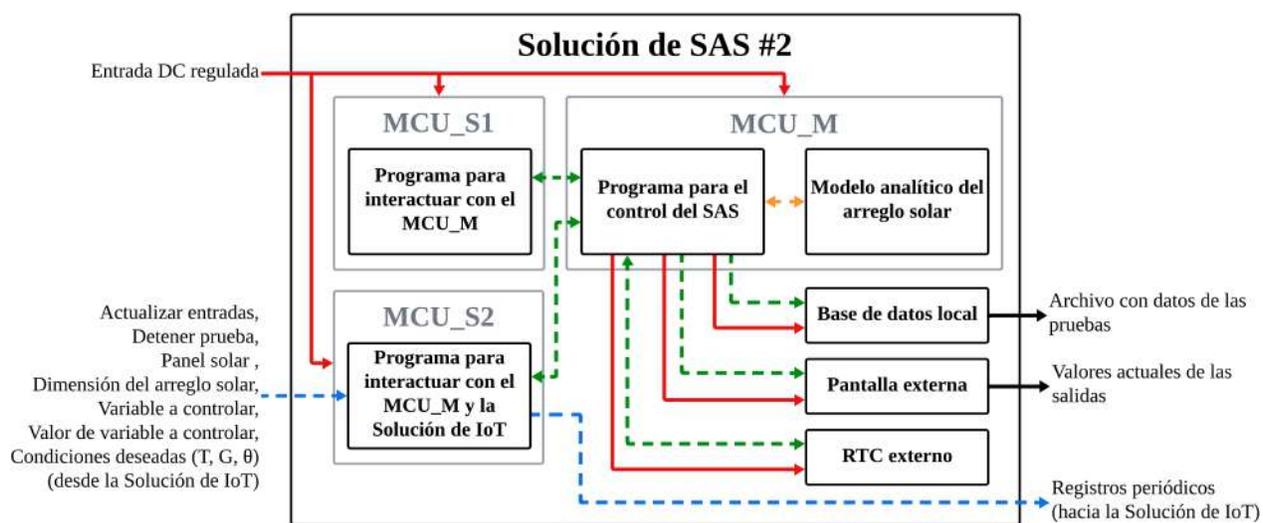


Figura 5.2: Descomposición funcional de la plataforma de SAS #2.

En este caso, se debe diseñar un sistema embebido que utiliza tres microcontroladores separados y otros componentes electrónicos complementarios para establecer comunicación con la solución IoT y obtener el punto deseado del arreglo solar que eventualmente debe recibir la planta MCP.

Primeramente, en el MCU_S2 se debe implementar un programa que se encargue de leer periódicamente las entradas del usuario en el panel de control de la solución IoT, analizarlas, y de cumplirse ciertas condiciones, enviar la información al MCU_M. Eventualmente, el MCU_M le envía de regreso registros periódicos que el MCU_S2 que enviar a la solución IoT para que se guarden en una base de datos en la nube.

Por otra parte, el programa de control implementado en el MCU_M primero debe configurar los componentes electrónicos complementarios, que en este caso corresponden a un RTC externo, una pantalla LCD y un lector de tarjetas Micro SD. Luego, una vez recibidas las entradas del usuario por parte del MCU_S2, se deben procesar para ejecutar el modelo analítico, el cual tiene un comportamiento idéntico al de la plataforma #1, que permita obtener el punto de operación del arreglo solar deseado. Una vez que se cuenta con este punto, se debe enviar al MCU_S1 para que este eventualmente devuelva las "lecturas" de corrien-

te y tensión, las cuales luego deben mostrarse en la pantalla LCD. Seguidamente, se deben guardar registros del estado actual del sistema en la tarjeta Micro SD, y a la vez enviarlos al MCU_S2.

Finalmente, se debe implementar un programa en el MCU_S1 que reciba los puntos de operación del MCU_M, les agregue ruido aleatorio (con el fin de emular las lecturas de corriente y tensión que eventualmente debe realizar el microprocesador de la planta MCP), y luego envíe de vuelta al MCU_M.

5.1. Generación y selección de conceptos

5.1.1. Modelo analítico del arreglo solar

El primer paso es seleccionar el modelo matemático a utilizar para describir el rendimiento de celdas solares. En el Capítulo 2, se explicaron los tres modelos más utilizados: el modelo ideal, el modelo de un diodo y el modelo de dos diodos.

Se realiza una matriz de selección en la que se consideran tres criterios a los que se les asigna un peso porcentual, tomando como base las necesidades planteadas en el Capítulo 3. Se le da prioridad a la precisión del modelo emulando el comportamiento de un panel solar real, seguido de la complejidad de resolución numérica (que está estrechamente relacionada con el tiempo que le toma al modelo calcular el punto de operación y con la memoria que requiere el programa), y la cantidad de literatura disponible, que si bien no se considera explícitamente en las necesidades, sigue siendo un criterio importante ya que facilita el proceso de implementación del modelo, así como su validación, pues permite comparar los resultados obtenidos con los de otros investigadores. En la Figura 5.3 se presentan los resultados, donde se observa que el modelo analítico ganador es el modelo estándar o de un diodo.

Criterios de evaluación	Peso (%)	Modelo analítico					
		Ideal		Un diodo (Ref.)		Dos diodos	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Precisión	40	1	0.4	3	1.2	5	2
Complejidad de resolución numérica	35	5	1.75	3	1.05	2	0.7
Literatura disponible	25	2	0.5	3	0.75	1	0.25
Puntaje			2.65		3.00		2.95
Lugar			3		1		2

Figura 5.3: Matriz de selección del modelo analítico ganador.

Seguidamente, se debe seleccionar un método adecuado para la resolución numérica del modelo de un diodo. En un artículo publicado en el 2019 [19], se realiza un comparación de diversos métodos de resolución del modelo de un diodo, donde se compara el rendimiento de cada uno a condiciones estándar de irradiancia, de 1000 W/m^2 , y a niveles más bajos, de 200 W/m^2 , lo cual requiere un escalado de los parámetros calculados. En la Figura 5.4, se puede apreciar que el modelo propuesto por Ahmed Hussein en el 2017 [105], presenta el MAPE más bajo de los métodos estudiados. Adicionalmente, en su artículo se explican de forma muy clara y concisa todos los pasos requeridos para la resolución del modelo analítico. Por lo tanto, se decidió implementar este método en la solución final del SAS.

Método	G (W/m^2)	MAPE (%)	MAPE promedio (%)
Hussein	1000	1.26	1.42
	200	1.58	
Cubas	1000	1.37	1.98
	200	2.59	
Villalva	1000	1.94	2.05
	200	2.16	
Ghani	1000	2.42	2.14
	200	1.85	
Celik	1000	1.28	2.27
	200	3.26	
Brano	1000	1.25	2.27
	200	3.28	
Laudani	1000	2.94	2.36
	200	1.78	

Figura 5.4: Comparación de diversos métodos para la resolución numérica del modelo de un diodo. Resultados tomados de [19].

El método de Hussein toma como entradas parámetros conocidos de un panel solar que son proporcionados por el fabricante para resolver la ecuación 2.8 del modelo de un diodo a condiciones ambientales conocidas de temperatura de celda $T_{c,ref}$ e irradiancia G_{ref} . Estos parámetros corresponden a V_{oc} , I_{sc} , V_{mp} , I_{mp} y N_s .

A continuación, se explica brevemente el método de Hussein, en el que se parte del conjunto de ecuaciones 5.1 y se asume un valor inicial de $n = 1$, con el fin de calcular el valor de R_s utilizando el el conjunto de ecuaciones 5.2 y el método numérico de Newton-Raphson, en la ecuación 5.3. Una vez que se conoce el valor de R_s , se calculan los demás parámetros utilizando el conjunto de ecuaciones 5.4.

$$\begin{aligned}
A &= \exp\left(\frac{V_{oc}}{V_t}\right) - 1 \\
B &= \exp\left(\frac{I_{sc}R_s}{V_t}\right) - 1 \\
C &= \exp\left(\frac{V_{mp} + I_{mp}R_s}{V_t}\right) - 1
\end{aligned} \tag{5.1}$$

$$\begin{aligned}
f(R_s) &= V_{mp}(C + 1)(I_{sc}V_{oc} - I_{sc}V_{mp} - I_{mp}V_{oc}) \\
&\quad - I_{sc}V_{mp}V_t(A - C) + I_{mp}V_{oc}V_t(B - C) + 2I_{mp}V_{mp}V_t(A - B) \\
f'(R_s) &= \frac{V_{oc}V_{mp}(C + 1)}{V_t}(I_{sc} - I_{mp}) + I_{sc}(B + 1)(V_{oc} - 2V_{mp}) \\
&\quad + (C + 1)\left(I_{sc}V_{mp} - I_{mp}V_{oc} - \frac{I_{sc}V_{mp}^2}{V_t}\right)
\end{aligned} \tag{5.2}$$

$$R_{s(i+1)} = R_{s(i)} - \frac{f(R_{s(i)})}{f'(R_{s(i)})} \tag{5.3}$$

$$R_{sh} = \frac{V_{oc}(C - B) - V_{mp}(A - B)}{I_{mp}(A - B) - I_{sc}(A - C)} - R_s$$

$$I_{ph} = \frac{I_{sc}\left(1 + \frac{R_s}{R_{sh}}\right)A - \frac{V_{oc}}{R_{sh}}}{A - B}$$

$$I_o = \frac{I_{ph}}{A} - \frac{V_{oc}}{AR_{sh}} \tag{5.4}$$

Una vez que se conocen todos los parámetros, se calcula el set de datos de la característica I-V del panel solar y se compara con un set de datos de referencia, de modo que se calcula el RMSE utilizando la ecuación 5.5. Este procedimiento se repite con diversos valores de n entre 1 y 2, de modo que se eligen los parámetros de la iteración en que el RMSE fue menor.

$$\text{RMSE} = \sqrt{\frac{\sum_0^i (I_i - I_{ref(i)})^2}{i}} \quad (5.5)$$

Obtención del set de datos

Como se mencionó anteriormente, se requiere de un set de datos de referencia de la característica I-V del panel solar, que, por lo menos para modelos comerciales de aplicaciones espaciales, los fabricantes no suelen proporcionar al público. No obstante, en artículos científicos y en hojas de datos es posible encontrar imágenes de dichas curvas para ciertos modelos, por lo que a partir de estas es posible extraer el set de datos requerido seleccionando múltiples puntos de muestreo (x, y) , como se muestra en la Figura 5.5.

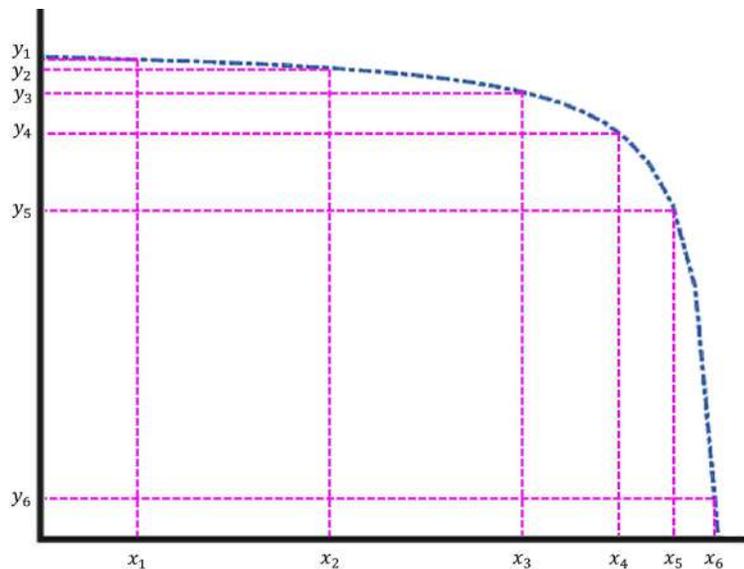


Figura 5.5: Ejemplo de muestreo de una curva I-V para obtener su set de datos.

Este procedimiento se puede realizar de forma manual, resultando en un proceso extremadamente lento y repetitivo si se desea un intervalo de muestreo pequeño, y por ende es propenso al error humano. No obstante, también puede ser realizado utilizando herramientas tecnológicas, como la visión computacional, la cual permite realizar el mismo proceso de manera mucho más precisa y para el tamaño de intervalo de muestreo que se desee, en cuestión de milisegundos. Es por esto que el claro concepto ganador es el de visión computacional, y para implementarlo se propone utilizar OpenCV (*Open Source Computer Vision*), una librería de visión computacional multi-plataforma disponible en Python, Java, C++ y MATLAB [106].

Selección de paneles solares

En cuanto a la selección de los paneles solares a utilizar, se tomó como referencia un listado de paneles solares de alto rendimiento para aplicaciones espaciales proporcionado por la NASA [42]. Luego, se buscaron las hojas de datos de los fabricantes para evaluar cuáles proporcionan imágenes de las curvas características, y también se buscaron artículos científicos donde los investigadores facilitaran dichas curvas. De este modo, se encontraron cuatro paneles solares comerciales con sus respectivas curvas características, presentados en la Tabla 5.1.

Tabla 5.1: Paneles solares de aplicaciones espaciales seleccionados.

Numeración	Fabricante	Panel solar	Referencias
1	AZUR Space	3G30A	[62], [107]
2	AZUR Space	4G32C	[63], [108]
3	CESI	CTJ30	[65]
4	Spectrolab	UTJ	[64]

Escalado de parámetros del modelo analítico

Por otra parte, una vez que son calculados los cinco parámetros desconocidos del modelo analítico, es posible utilizar ecuaciones encontradas en la literatura [41], [109], [110], [111], [112] para escalarlos a diferentes condiciones de irradiancia G , temperatura de celda T_c , y ángulo de incidencia θ , las cuales se presentan en el conjunto de ecuaciones 5.6. El parámetro E_g corresponde a la brecha de banda energética del material de la celda fotovoltaica, que en el caso de las tecnologías de multi-juntura, como la de los paneles solares seleccionados, corresponde a 1.6 eV [110].

$$\begin{aligned}
 G &= G_{\text{ref}} \cdot \cos(\theta) \\
 n &= n_{\text{ref}} \\
 I_{\text{ph}} &= \frac{G}{G_{\text{ref}}} \cdot [I_{\text{ph,ref}} + K_I \cdot (T_c - T_{c,\text{ref}})] \\
 I_o &= I_{o,\text{ref}} \cdot \left(\frac{T_c}{T_{c,\text{ref}}} \right)^3 \cdot \exp \left[\frac{qE_g}{nk} \cdot \left(\frac{1}{T_{c,\text{ref}}} - \frac{1}{T_c} \right) \right] \\
 R_s &= \frac{R_s \cdot G_{\text{ref}}}{G} \\
 R_{sh} &= \frac{R_{sh} \cdot G_{\text{ref}}}{G}
 \end{aligned} \tag{5.6}$$

Variantes del modelo analítico

Se concluye que solo es necesario utilizar el método de Hussein una única vez por cada panel solar comercial que se desee simular, ya que una vez que se conocen los cinco parámetros a condiciones ambientales de referencia, escalarlos a otras condiciones y utilizarlos para resolver la ecuación característica del modelo es mucho más simple y requiere menos recursos computacionales.

Es por esto que el modelo analítico propuesto tiene dos componentes separadas. La primera componente, presentada en la Figura 5.6, tiene como objetivo extraer el set de datos de una imagen de la curva característica del panel y luego emplear el método de Hussein para calcular los cinco parámetros desconocidos, los cuales luego pueden ser escalados a otras condiciones ambientales y generar los sets de datos respectivos de las curvas características I-V y P-V. Esto permite confeccionar una hoja de datos para cada panel solar que se estudie y generar gráficas donde se pueda apreciar la forma en que el modelo emula el comportamiento del panel en diferentes condiciones.

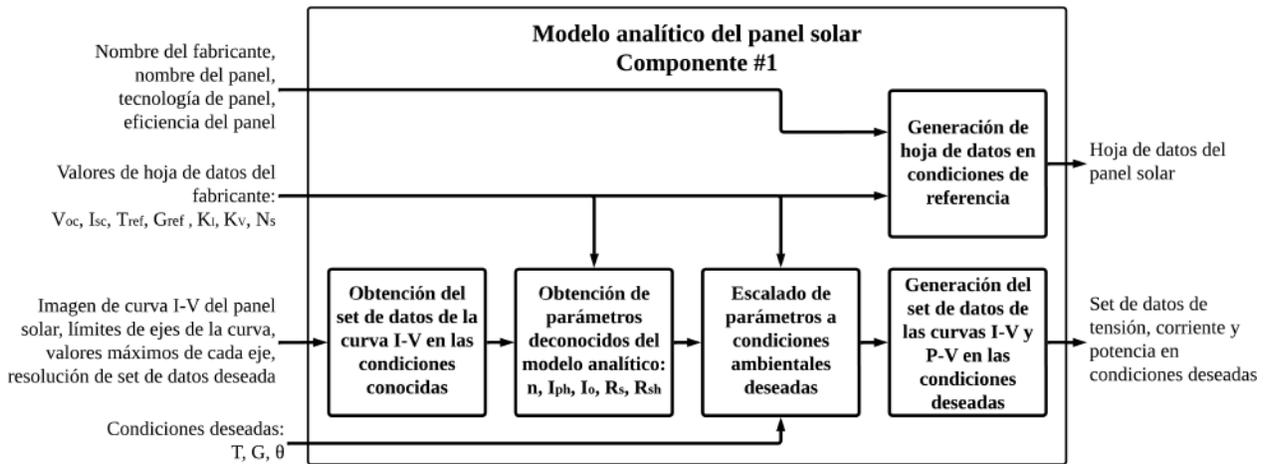


Figura 5.6: Componente #1 del modelo analítico propuesto.

La segunda componente del modelo analítico, presentada en la Figura 5.7, permite al usuario seleccionar un panel solar entre la lista de opciones predefinidas, de acuerdo a lo indicado en la Tabla 5.1, a las cuales ya se les calcularon los parámetros del modelo previamente y se les fue generada una hoja de datos. Asimismo, puede introducir las condiciones ambientales deseadas, las dimensiones del arreglo solar, y seleccionar el valor de tensión o corriente deseada a la salida, para que el modelo analítico use los datos conocidos para calcular su par y obtener así el punto de operación correspondiente. Cabe destacar que esta componente es la que se implementa en las plataformas de SAS para pruebas experimentales, descritas en el Capítulo 4.

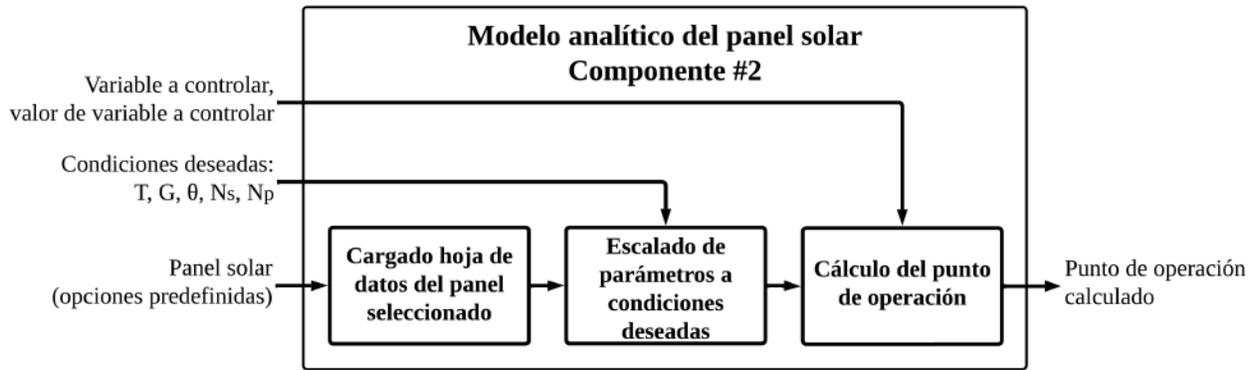


Figura 5.7: Componente #2 del modelo analítico propuesto.

Parámetros de entrada y salida del modelo analítico

A continuación, en la Tabla 5.2, se resumen los parámetros, con su respectiva simbología, que el usuario introduce desde la plataforma IoT y que luego son usadas por el modelo analítico. Cabe destacar el parámetro PV corresponde a un valor numérico entre 1 y 4, según la numeración que a cada panel se le asignó en la Tabla 5.1. Además, el parámetro CV corresponde a un valor binario; si el usuario desea introducir un valor de tensión en CVV , se tiene que CV es igual a 0, mientras que si desea introducir un valor de corriente, CV es igual a 1.

Tabla 5.2: Parámetros de entrada introducidos por el usuario.

Parámetro	Simbología
Panel solar	PV
Número de paneles solares en serie	N_s
Número de paneles solares en paralelo	N_p
Irradiancia (W/m^2)	G
Temperatura de celda ($^{\circ}C$)	T
Ángulo de incidencia ($^{\circ}$)	θ
Variable de control	CV
Valor de la variable de control	CVV

En la Tabla 5.3, se presentan los parámetros propios del panel solar, provenientes de hojas de datos, así como los cinco parámetros del modelo de un diodo, a condiciones de referencia, que se calcularon previamente utilizando el método de Hussein. Cabe aclarar que dentro de un solo módulo de panel solar pueden existir múltiples celdas en serie. A este valor es al que se hace referencia con $N_{s,pv}$, y no debe confundirse con el valor de N_s introducido por el usuario.

Tabla 5.3: Parámetros del panel solar seleccionado.

Parámetro	Simbología
Temperatura de la celda de referencia	T_{ref}
Irradiancia solar de referencia	G_{ref}
Tensión de circuito abierto	V_{oc}
Corriente de cortocircuito	I_{sc}
Tensión en el punto de máxima potencia	V_{mp}
Corriente en el punto de máxima potencia	I_{mp}
Potencia en el punto de máxima potencia	P_{mp}
Número de celdas en serie (en un solo panel solar)	$N_{\text{s,pv}}$
Coefficiente de temperatura de V_{oc}	K_v
Coefficiente de temperatura de I_{sc}	K_i
Brecha de banda energética	E_g
Factor de idealidad del diodo	n
Corriente fotoeléctrica de referencia	$I_{\text{ph,ref}}$
Corriente inversa de saturación del diodo de referencia	$I_{\text{o,ref}}$
Resistencia en serie de referencia	$R_{\text{s,ref}}$
Resistencia de derivación de referencia	$R_{\text{sh,ref}}$

Finalmente, en la Tabla 5.4, se presentan los parámetros de salida obtenidos luego de resolver el modelo analítico y medir la corriente y tensión, así como los ARPE asociados, según lo indicado en la ecuación 2.45. Cabe destacar que R_{out} únicamente es relevante para la plataforma de SAS #1, puesto que se necesita para configurar la resistencia de la carga electrónica. Asimismo, se presentan los parámetros del modelo de un diodo luego de ser escalados de acuerdo a las entradas del usuario, según el conjunto de ecuaciones 5.6, destacando que n permanece constante sin importar las condiciones externas.

Tabla 5.4: Parámetros de salida.

Dato	Simbología
Tensión de salida calculada y medida (V)	$V_{\text{out,calc}}, V_{\text{out,meas}}$
Corriente de salida calculada y medida (A)	$I_{\text{out,calc}}, I_{\text{out,meas}}$
Potencia de salida calculada y medida (W)	$P_{\text{out,calc}}, P_{\text{out,meas}}$
Resistencia de salida calculada y medida (Ω)	$R_{\text{out,calc}}, R_{\text{out,meas}}$
ARPE de la tensión, corriente, potencia y resistencia (%)	$Err_V, Err_I, Err_P, Err_R$
Factor de idealidad del diodo	n
Corriente fotoeléctrica escalada	I_{ph}
Corriente inversa de saturación del diodo escalada	I_{o}
Resistencia en serie escalada	R_{s}
Resistencia de derivación escalada	R_{sh}

5.1.2. Guardado de datos

Como se mencionó en el Capítulo 4, para la plataforma de SAS # 2, se utiliza una tarjeta Micro SD como base de datos local y se obtiene la fecha y hora actual de una placa RTC externa. Para lograrlo, se configura el MCU_M para que se comuniqué con el lector Micro SD a través del protocolo SPI y con el RTC a través de I2C, para escribir en un archivo de texto los parámetros de interés, separados entre sí por comas, tal y como se presenta a continuación:

$$Fecha, Hora, PV, N_s, N_p, CV, G, T, \theta, V_{out,calc}, V_{out,meas}, I_{out,calc}, I_{out,meas}, P_{out,calc}, P_{out,meas}, \\ Err_V, Err_I, Err_P, n, I_{ph}, I_o, R_s, R_{sh}$$

En el caso de la plataforma de SAS # 1, los datos son guardados en un archivo .CSV, y en este caso se deben agregar también los parámetros relacionados con la resistencia en la carga electrónica, tal y como se muestra a continuación:

$$Fecha, Hora, PV, N_s, N_p, CV, G, T, \theta, \\ V_{out,calc}, V_{out,meas}, I_{out,calc}, I_{out,meas}, P_{out,calc}, P_{out,meas}, R_{out,calc}, R_{out,meas}, \\ Err_V, Err_I, Err_P, Err_R, n, I_{ph}, I_o, R_s, R_{sh}$$

5.1.3. Pantalla externa

Como se mencionó en el Capítulo 4, se utiliza una pantalla LCD1602, configurada mediante protocolo I2C por el MCU_M, para mostrar los valores de salida medidos con dos decimales de precisión, tal y como se observa en la Figura 5.8.

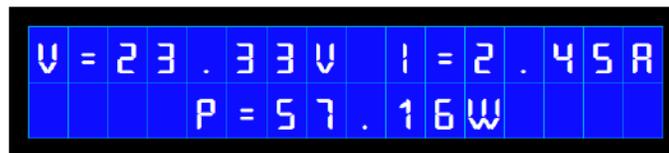


Figura 5.8: Ejemplo de datos mostrados en la pantalla externa.

5.2. Desarrollo de conceptos

5.2.1. Modelo analítico: Componente #1

Obtención de sets de datos con visión computacional

A continuación, se procede a explicar el algoritmo desarrollado en Python que utiliza la librería de visión computacional OpenCV para obtener el set de datos de la curva característica I-V de un panel, a partir de una imagen de la misma y conociendo los valores máximos de tensión y corriente que se pueden encontrar; es decir, los valores de V_{oc} e I_{sc} proporcionados en la hoja de datos del panel solar. Asimismo, se utiliza la librería NumPy para facilitar la manipulación de arreglos de datos.

En la Figura 5.9, se presenta la primera parte del diagrama de flujo del algoritmo.

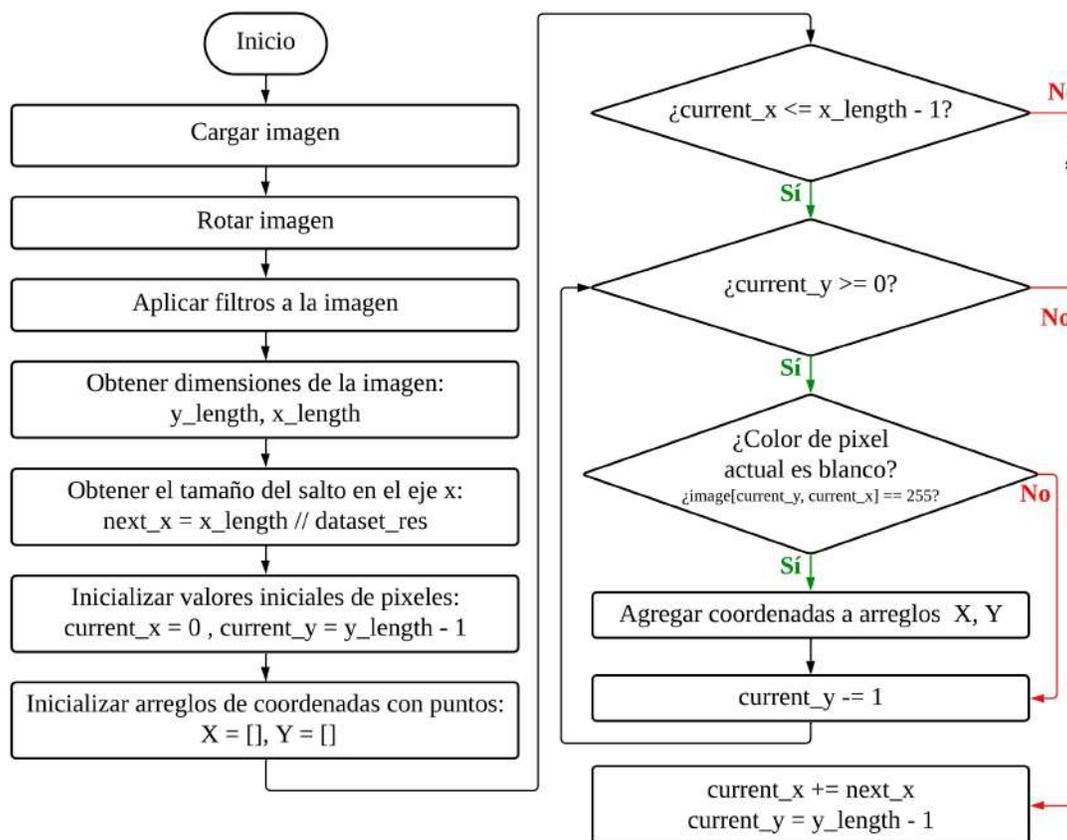


Figura 5.9: Diagrama de flujo del algoritmo de visión computacional desarrollado. Parte 1.

Se inicia cargando la imagen de la curva I-V de referencia, la cual luego se voltea a lo largo del eje x inferior, con el fin de que el origen de la curva I-V coincida con el origen del sistema de coordenadas de la imagen, el cual está en píxeles. Después, convierte la imagen a escala de grises y se aplica un umbral para convertir la imagen en una representación binaria, donde los píxeles blancos representan la curva y los píxeles negros representan el fondo.

El algoritmo procede a recorrer todos los píxeles en el eje x de la imagen. Para cada píxel en x , se recorren todos los píxeles correspondientes en y . Si se encuentra un píxel blanco (valor 255 en la escala de grises), se agregan sus coordenadas a las listas X , Y . Una vez que se escanea la columna completa en y para un píxel dado en x , se salta al siguiente píxel en x .

No obstante, debido al grosor de la curva en la imagen, se obtiene como resultado que para un mismo valor de x pueden existir múltiples valores de y . Este problema se soluciona con la siguiente parte del algoritmo, cuyo diagrama de flujo se presenta en la Figura 5.10.

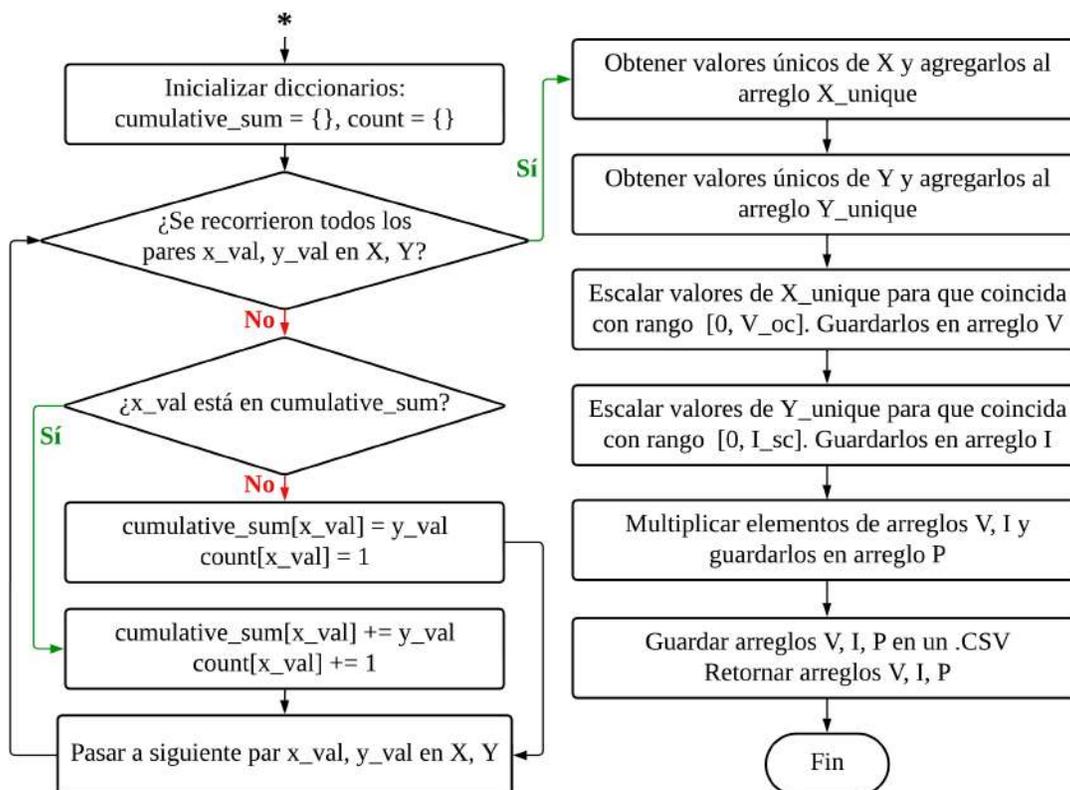


Figura 5.10: Diagrama de flujo del algoritmo de visión computacional desarrollado. Parte 2.

Primero, se inicializan dos diccionarios vacíos llamados *cumulative_sum* y *count*, los cuales se utilizan para almacenar la suma acumulativa de los valores de y y el recuento de cuántos puntos de la curva tienen cada valor de x , respectivamente.

Luego, se itera sobre cada par de valores de coordenadas almacenados en los arreglos X e Y . Para cada par, se verifica si el valor actual de x , almacenado en la variable x_val , ya está presente en el diccionario *cumulative_sum*. Si no está presente, se agrega x_val a *cumulative_sum*, con su correspondiente valor y_val , y el contador de ese valor de x se inicializa en 1 en el diccionario *count*. Si x_val ya está presente en *cumulative_sum*, se suma el valor actual de y_val al valor acumulado de la coordenada y correspondiente en *cumulative_sum*, y se incrementa en 1 el contador asociado a ese valor de x en *count*.

Después de completar esta iteración sobre todos los puntos de la curva, se obtienen los valores únicos de x y se guardan en el arreglo X_unique . Luego, se calcula el valor promedio de y para cada valor único de x , dividiendo la suma acumulativa de y por el recuento correspondiente de puntos de la curva con ese valor de x . Estos valores promedio se almacenan en el arreglo Y_unique .

Seguidamente, se escalan los valores de X_unique e Y_unique , los cuales están en píxeles, a los valores esperados de corriente y tensión, según los valores de V_{oc} e I_{sc} proporcionados en la hoja de datos del fabricante. Los resultados se guardan en los arreglos V e I , respectivamente. Luego, se multiplica cada valor de tensión por su respectivo par de corriente, para así obtener los valores de potencia, que se almacenan en el arreglo P .

Finalmente, se escriben los sets de datos de tensión, corriente y potencia obtenidos en un archivo .CSV, y se retornan los arreglos para que puedan ser usados por otras funciones dentro del programa.

Implementación del método de Hussein

En la Figura 5.11, se presenta el diagrama de flujo de la implementación del método de Hussein para obtener los cinco parámetros desconocidos de la ecuación del modelo de un diodo. Se inicia tomando los parámetros de entrada de la hoja de datos del fabricante, V_{oc} , I_{sc} , N_{s_pv} y T_{ref} , y los insumos provenientes del algoritmo de visión computacional, V_{mp} , I_{mp} , V_{arr_ref} e I_{arr_ref} . Asimismo, es necesario definir los valores iniciales de n y R_s , además de los saltos que se desean hacer en n y la tolerancia mínima requerida a la hora de resolver el método de Newton-Raphson. Se toma un valor inicial de n igual a 1, con el fin de explorar la totalidad del rango de posibles valores, y un R_s inicial igual a 0.3, ya que con valores más pequeños se tenían problemas de desbordamiento al iterar. Para la tolerancia, se tomó el mismo valor utilizado por el autor, correspondiente a 10^7 . Luego, se requiere inicializar algunas variables que se utilizan durante la ejecución de ciclos, como contadores o arreglos, además de las variables nV_t , correspondiente a la ecuación 2.2 multiplicada por n , y A, B y C, según el conjunto de ecuaciones 5.1.

Se procede con el ciclo principal, el cual comprueba que el valor actual de n para verificar que sea igual o menor a 2. Si se cumple, se procede a calcular el valor actual de la variable V_t y de $f(R_s)$ y $f'(R_s)$, según el conjunto de ecuaciones 5.2, las cuales dentro del programa se almacenan en las variables x e y , respectivamente. Se procede a verificar si el valor absoluto de x entre y es mayor al valor de tolerancia definido, y de ser el caso, se calcula el valor actual de las variables A, B, C, x , y , y R_s , según la ecuación 5.3. Cuando se logra obtener un valor inferior al de la tolerancia, se procede a calcular el valor de R_{sh} , I_{ph} e I_o , según el conjunto de ecuaciones 5.4.

Luego, se inicia un ciclo secundario para obtener el RMSE de la iteración actual. Para esto, se usa cada valor en V_{arr_ref} y los 5 parámetros obtenidos para resolver la ecuación 2.8 del modelo de diodo único y obtener la corriente I_{calc} , y cada uno de estos valores se guarda en el arreglo I_{arr_calc} . A partir de los arreglos I_{arr_ref} e I_{arr_calc} , es posible aplicar la ecuación 2.49 para obtener el RMSE. Adicionalmente, también se aplican las ecuaciones 2.50, 2.53 y 2.46 para obtener el nRMSE, R^2 y MAPE, respectivamente, con el fin de obtener más indicadores de rendimiento para evaluar el modelo.

Finalmente, se guardan los 5 parámetros y los indicadores de rendimiento de la iteración actual en un diccionario, que luego se escribe en un .CSV y se agrega a un arreglo, y se procede a reiniciar el ciclo principal con el siguiente valor de n que corresponda. Cuando n es mayor a dos, se procede a encontrar la iteración en la que se presentó el menor valor de RMSE y se extraen los parámetros correspondientes, dando por terminada la implementación del método de Hussein.

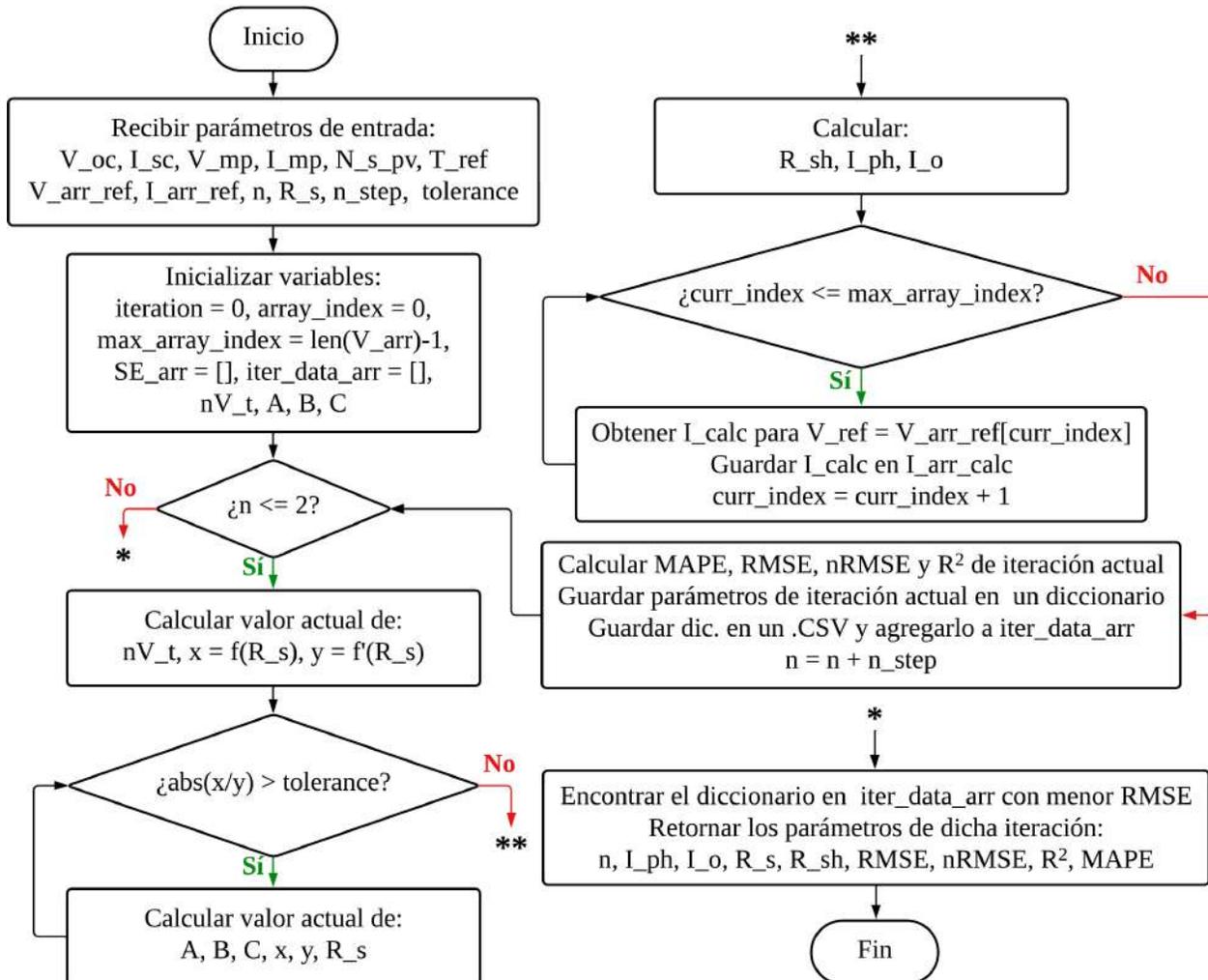


Figura 5.11: Diagrama de flujo de la implementación del método de Hussein.

Confección de hojas de datos

Se requiere confeccionar una tabla resumen para cada panel solar estudiado, en la que estén todos los parámetros conocidos, tanto los extraídos de de la hoja de datos del fabricante y de la literatura, como los calculados mediante el modelo de diodo único con los respectivos indicadores de rendimiento (RMSE, nRMSE, R^2 y MAPE).

Asimismo, se debe generar un set de datos que utilice los parámetros para calcular distintos puntos de la curva característica, y luego generar una gráfica donde se pueda comparar el set de datos de referencia, con el calculado. Luego, se debe aplicar el conjunto de ecuaciones 5.6 para escalar los parámetros obtenidos del modelo a, por lo menos, cinco puntos distintos dentro del rango establecido de irradiancia, temperatura y ángulo de incidencia, de modo que se pueda generar un set de datos de la curvas característica para cada punto estudiado y generar gráficas en donde se pueda apreciar la variación en el comportamiento eléctrico del panel en función de las condiciones externas estudiadas.

En la Tabla 5.5, se resumen los puntos de estudio escogidos, donde se aprecia que se parte del límite inferior de rango y se hacen incrementos equidistantes hasta llegar al límite superior del rango. En el caso del ángulo de incidencia, si se llega al límite superior (ángulo de 90°), las salidas son equivalentes a cero, por lo tanto, se decidió llegar hasta un ángulo de 80° .

Tabla 5.5: Puntos estudiados en el rango de condiciones externas.

Condición externa	Puntos estudiados
Irradiancia (W/m^2)	275, 550, 825, 1100, 1375
Temperatura de celda ($^\circ\text{C}$)	-80, -40, 0, 40, 80
Ángulo de incidencia ($^\circ$)	0, 20, 40, 60, 80

Finalmente, se organizan los resultados obtenidos en un solo documento que, para cada panel, contenga la tabla resumen de sus parámetros eléctricos, las gráficas generadas (tanto a condiciones de referencia, como a diferentes condiciones externas), y enlaces para descargar los sets de datos obtenidos.

En la Figura 5.12, se presenta el diagrama de flujo del programa principal de la componente # 1 del modelo analítico, de acuerdo a lo descrito anteriormente.

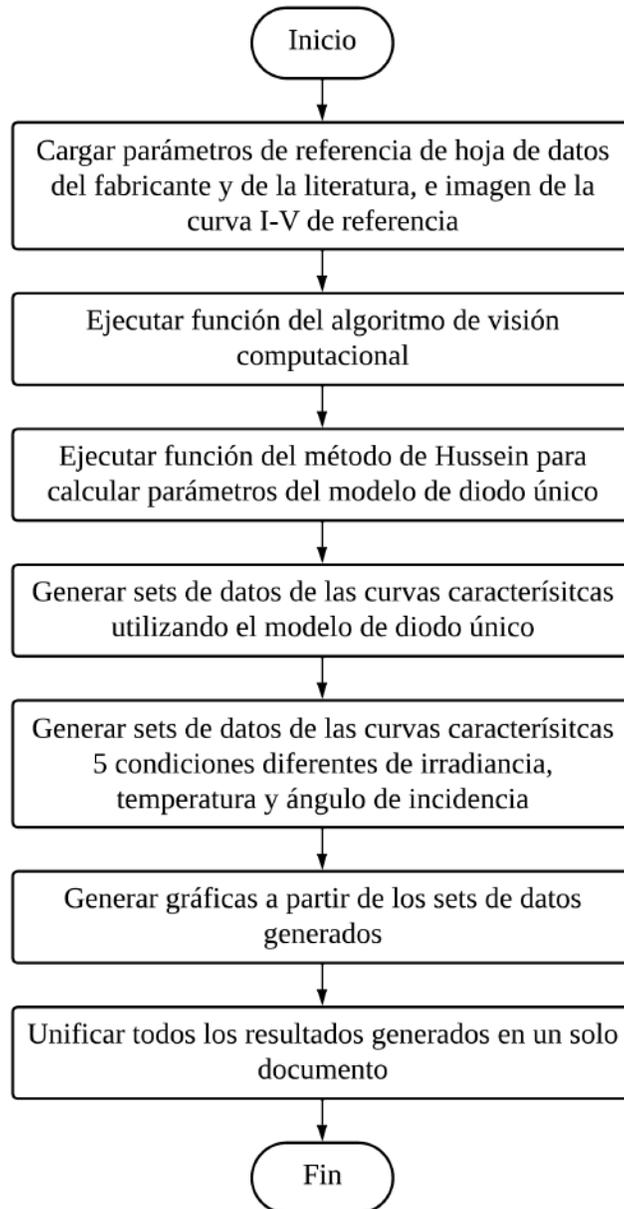


Figura 5.12: Diagrama de flujo del programa utilizado para confeccionar las hojas de datos de cada panel.

5.2.2. Modelo analítico: Componente #2

Como se mencionó en el Capítulo 4, se implementaron dos plataformas de SAS separadas para realizar pruebas experimentales. No obstante, el funcionamiento de la componente #2 de modelo analítico, propuesto en la Figura 5.7, es idéntico en ambas, y su diagrama de flujo se presenta en la Figura 5.13. En diagramas de flujo posteriores, se hace referencia a esta secuencia de pasos como la función *Analytic_Model()*.

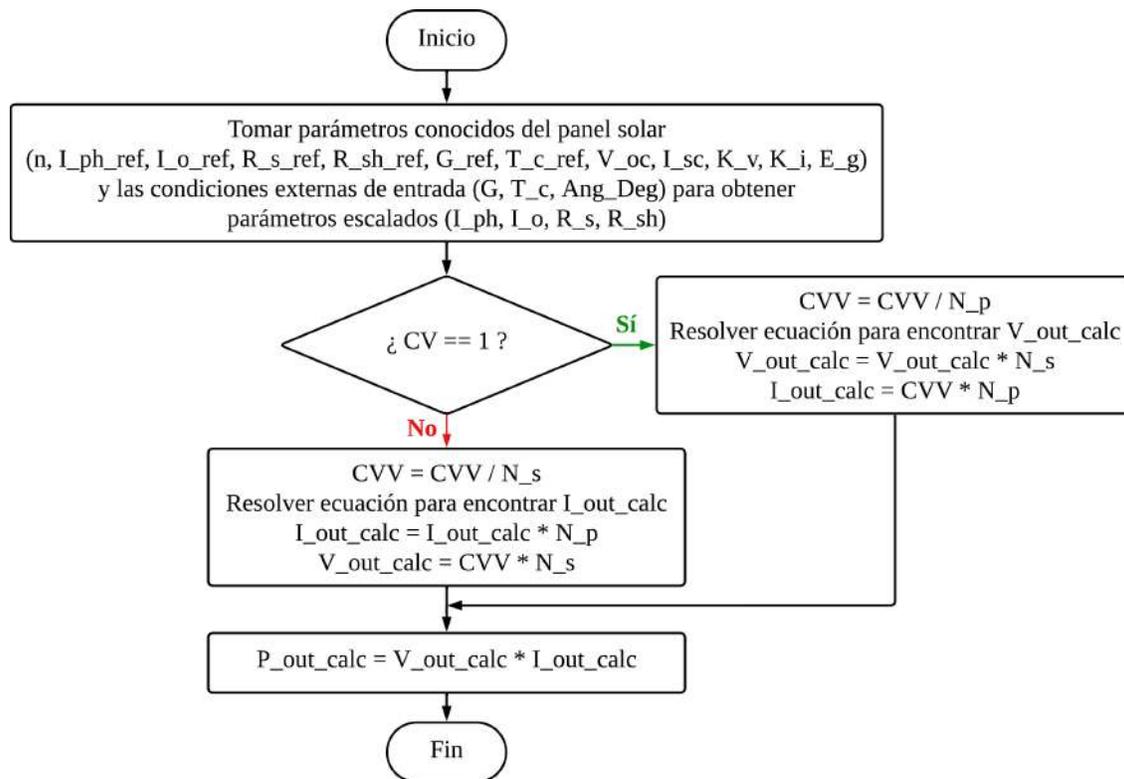


Figura 5.13: Diagrama de flujo de la función *Analytic_Model()*.

Se inicia asumiendo que, previamente, se recibieron desde la solución de IoT los parámetros de entrada listados en las Tablas 5.2 y 5.3. Se procede a utilizar esta información para escalar los cinco parámetros de referencia a las condiciones de irradiancia, temperatura de celda y ángulo de incidencia introducidas por el usuario, según lo indicado en el conjunto de ecuaciones 5.6.

Luego, se consulta el valor de la variable de entrada CV . Si este corresponde a 1, quiere decir que el usuario introdujo el valor de corriente de salida deseado, almacenado en la variable CVV . Se divide este valor entre el número de celdas en paralelo introducidas (ya que los valores de referencia de todos los parámetros del panel solar corresponden a un arreglo solar 1x1). Luego, se calcula la tensión de salida correspondiente resolviendo la ecuación 2.8 tomando a V como incógnita. Dado que este valor corresponde al de un arreglo 1x1, es necesario escalar tanto la corriente como la tensión a las dimensiones de arreglo indicadas por el usuario, por ende, se multiplica el valor de tensión obtenido por N_s y Cvv se multiplica por N_p (obteniendo el valor de corriente original).

Por otra parte, si el valor de CV corresponde a 0, quiere decir que el usuario introdujo el valor de tensión de salida deseado, y se realiza el mismo procedimiento descrito anteriormente, solo que en este caso el valor de CVV se divide entre el número de celdas en serie y la ecuación 2.8 se resuelve tomando a I como incógnita. Finalmente, se calcula el valor de la potencia de salida y se da por terminada la función.

A continuación, se explica en detalle el funcionamiento de cada plataforma de SAS con sus respectivos programas:

Plataforma de SAS #1

Como se mencionó anteriormente, se utiliza una fuente de poder programable RIGOL DP832 [102] cuya salida se conecta a una carga electrónica programable RIGOL DL3021 [103]. Para controlar los instrumentos, el fabricante recomienda utilizar el estándar conocido como *Arquitectura de Software de Instrumentos Virtuales* (o VISA, por sus siglas en inglés), el cual permite configurar, programar y solucionar problemas de sistemas de instrumentación que comprenden interfaces GPIB, VXI, PXI, Serial, Ethernet y/o USB [113]. Asimismo, ya existe un paquete de Python denominado PyVISA, que ya cuenta con la implementación de este estándar y permite acceder a los sistemas de bus utilizados por los instrumentos, por lo que fue la herramienta seleccionada para la implementación de la plataforma. Por ende, el cerebro de esta plataforma corresponde a un ordenador capaz de ejecutar programas en Python.

La fuente de poder cuenta con dos canales que pueden suministrar un máximo de 30 V y 3 A, cada uno. Dado que en algunos de los puntos de operación de los paneles solares seleccionados que se requieren evaluar, se sobrepasa el límite de 3 A, se optó por colocar los dos canales en paralelo, teniendo así como límites 30 V y 6 A. Además, se utilizaron los cables de medición de la carga electrónica para medir con alta precisión la caída de tensión recibida de la fuente de poder.

Por otra parte, se escribió una librería personalizada para enviar comandos de control a los instrumentos RIGOL presentes en el laboratorio, permitiendo así poder ejecutar funciones dentro del programa que configuraren los valores de tensión y corriente deseados en cada canal de la fuente de poder, y encender o apagar dichos canales. Asimismo, se puede configurar el modo de operación de la carga electrónica, los rangos de resistencia y tensión, y obtener las lecturas de sus sensores.

Se asume además que se cuenta con una plataforma de IoT funcional, desde la cual se pueden obtener los parámetros de entrada, listados en las Tablas 5.2 y 5.3, y a la cual se le pueden enviar los parámetros de salida listados en la Tabla 5.4, para que estos queden registrados en una base de datos en la nube.

En la Figura 5.14, se presenta el diagrama de flujo del ciclo principal del programa implementado en Python. Se inicia enviando una solicitud a la interfaz de control IoT para recibir un mensaje que proporciona información sobre las entradas del usuario y los parámetros del panel solar seleccionado; este contenido se almacena en la variable *UI_msg*.

Si *UI_msg* es distinto a cero, es diferente al último mensaje recibido, y el valor de la entrada *Start/Stop* (correspondiente al valor numérico ubicado en el índice 2 de *UI_msg*) es igual a 1, se procede a separar los parámetros de entrada del usuario de los parámetros del panel solar, y se ejecuta la función *load_user_inputs()*, la cual descompone el mensaje para obtener el valor numérico correspondiente a cada parámetro de entrada y los almacena en variables separadas dentro del programa. Luego, se verifica si el valor de *PV*, de acuerdo a la numeración asignada a cada panel en la Tabla 5.1, corresponde al último valor de *PV* conocido. Si se recibe un valor diferente, se procede a ejecutar la función *load_panel_params()*, la cual descompone el mensaje para obtener el valor numérico correspondiente a cada parámetro del panel solar y los almacena en variables separadas dentro del programa. De ser el mismo valor de *PV*, se asume que ya los parámetros del panel solar fueron cargados en una iteración previa del ciclo. Seguidamente, se ejecuta la función *Analytic_Model()* explicada anteriormente con la Figura 5.13, la cual escala los parámetros del modelo de un diodo de acuerdo a las entradas del usuario, y permite obtener el punto de operación de corriente y tensión deseado a la salida del arreglo solar. Estos parámetros son tomados por la función *Control_Instruments()*, la cual se encarga de configurar las salidas en los instrumentos de laboratorio utilizados, y una vez se reciben de vuelta las lecturas de los sensores, se guardan los registros, tanto en la base de datos de la solución IoT, como en una base de datos local, que en este caso corresponde a un archivo .CSV guardado en el ordenador. Esto da por terminada la iteración del ciclo.

Por otra parte, si *UI_msg* es igual al último mensaje recibido, almacenado en *last_UI_msg*, quiere decir que todo el procedimiento descrito en el párrafo anterior ya fue realizado en una iteración previa del ciclo, por lo que solo es necesario obtener las lecturas actuales de las salidas de los instrumentos, y guardar los registros en ambas bases de datos, dando por terminada la iteración actual del ciclo.

Si se da el caso de que la entrada *Start/Stop* es igual a 0, quiere decir que el usuario desea suspender las salidas del arreglo solar, por lo que se procede a apagar los canales de la fuente de potencia, dando por terminada la iteración actual del ciclo.

Finalmente, se puede apreciar que todas las acciones descritas anteriormente tienen como condición en común que *UI_msg* sea diferente a 0, ya que si esto no se cumple, quiere decir que el usuario no desea que el programa pueda obtener los valores de entrada introducidos en la interfaz de control ni que se guarden registros en las bases de datos. Por lo tanto, cuando *UI_msg* es igual a 0 no se debe ejecutar ninguna acción y se da por terminada la iteración actual del ciclo.

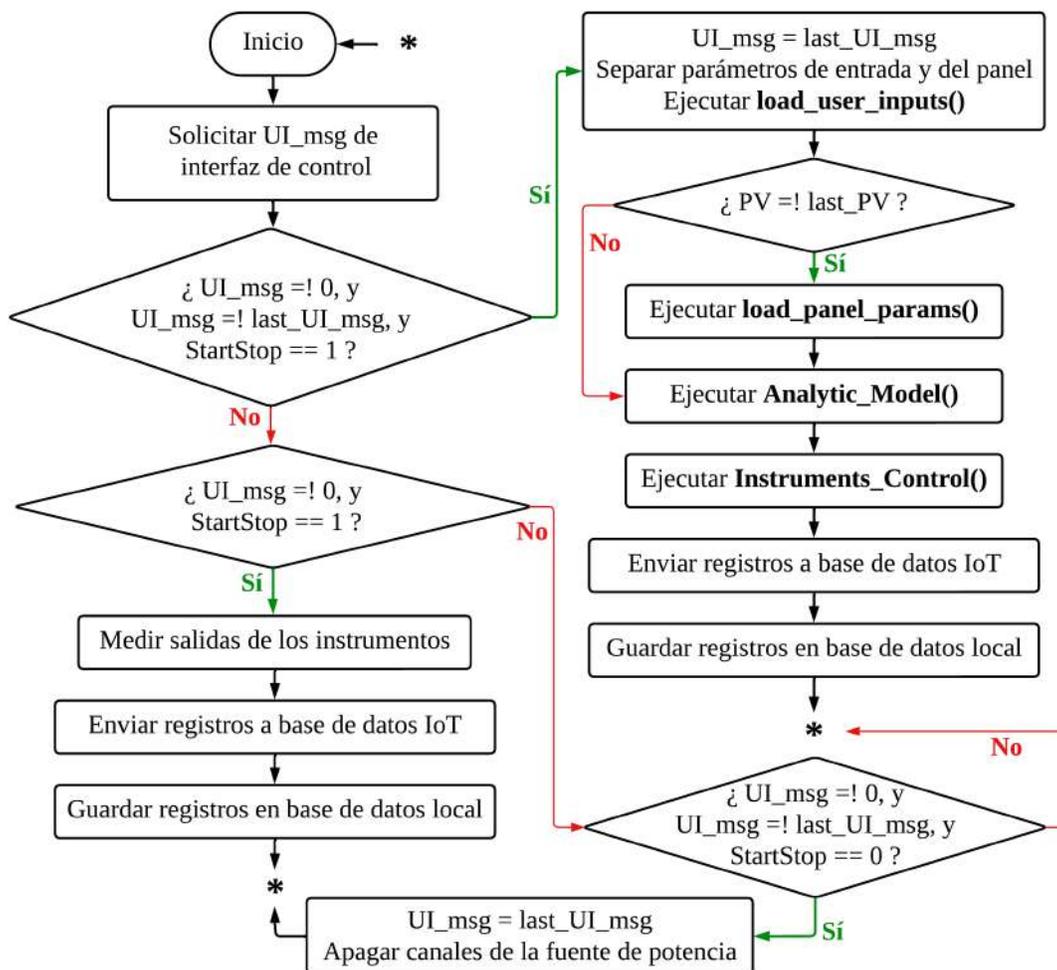


Figura 5.14: Ciclo de control principal de la plataforma de SAS #1

En la Figura 5.15, se presenta el diagrama de flujo de la función *Control_Instruments()*. Se inicia ejecutando la función *Op_Point_Config()*, la cual se encarga de configurar el punto de operación de corriente y tensión en la fuente de potencia, y el valor de resistencia en la carga electrónica. Luego, se obtienen las lecturas de los sensores de tensión y corriente, y con ellas se calculan indirectamente los valores de potencia y resistencia. Se calculan los porcentajes de error absoluto relativo para cada parámetro de salida, según la ecuación 2.43. Si el porcentaje de error obtenido para la potencia es superior a 0.5%, se procede a ejecutar la función *Op_Point_Correc()*, la cual hace ajustes en la configuración de los instrumentos hasta obtener porcentajes de error aceptables.

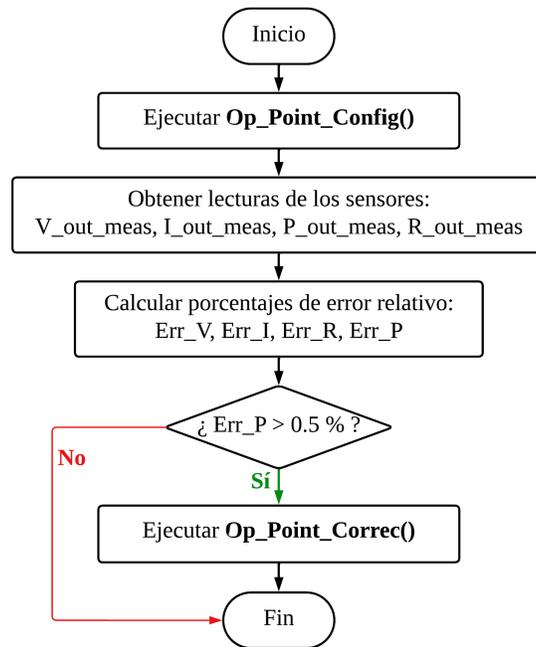


Figura 5.15: Función para controlar instrumentos del laboratorio

En la Figura 5.16, se presenta el diagrama de flujo de la función *Op_Point_Config()*. Cabe mencionar que la carga electrónica se opera en el modo de resistencia constante o CR. Este modo presenta dos rangos de operación, el rango mínimo corresponde a valores de resistencia menores a $15\ \Omega$ y de tensión menores a 10, y el rango máximo acepta valores de resistencia de hasta $15\ \text{k}\Omega$ y tensiones de hasta 155 V. Es necesario ajustar el rango de acuerdo al punto de operación deseado para obtener mayor precisión en las salidas, por lo tanto se consulta el valor de tensión y resistencia de salida calculados por el modelo analítico, almacenados en V_out_calc y V_out_calc , respectivamente, y el rango de resistencia actual, almacenado en RES_range , para hacer los ajustes necesarios en la fuente de potencia. Cabe destacar que cada vez que se hacen ajustes en el rango de resistencia de la carga, se deben apagar los canales de la fuente para evitar exceder los valores máximos que soporta la carga en ese momento.

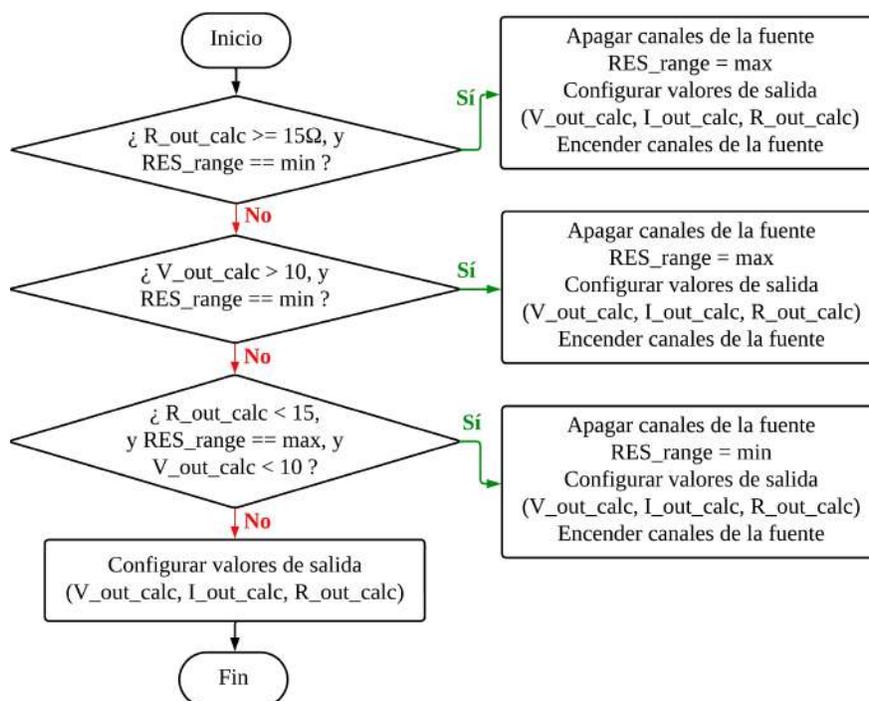


Figura 5.16: Función para configurar el punto de operación en los instrumentos.

De manera experimental se determinó que en la gran mayoría de los puntos de operación estudiados, la tensión y/o corriente configurada y que la fuente de poder dice entregar, no corresponde a los valores leídos por los sensores de precisión de la carga electrónica. El caso más común es que la tensión que dice entregar la fuente es considerablemente menor a la que lee la carga, lo que provoca discrepancias muy altas entre los valores de salida medidos y los esperados. Por lo tanto, es necesario ajustar los valores de tensión y/o corriente en la fuente hasta obtener valores aceptables en la carga electrónica. Por ejemplo, en la Figura 5.17, se puede apreciar cómo, luego de realizar las correcciones requeridas, la fuente de poder (izquierda) dice entregar 3.26 V y aproximadamente 1.11 A (el resultado de sumar las corrientes de ambos canales), pero los valores leídos en la carga (derecha) corresponden a 3.198 V y 1.108 A, siendo más cercanos a los valores esperados.



Figura 5.17: Ejemplo de corrección de un punto de operación. a) Fuente de poder y b) Carga electrónica

La función encargada de realizar estos ajustes de manera automática corresponde a *Op_Point_Correc()*, cuyo diagrama de flujo se presenta en la Figura 5.18. Esta realiza dos ciclos separados en donde se aumenta el valor de tensión y corriente en la fuente de poder, respectivamente.

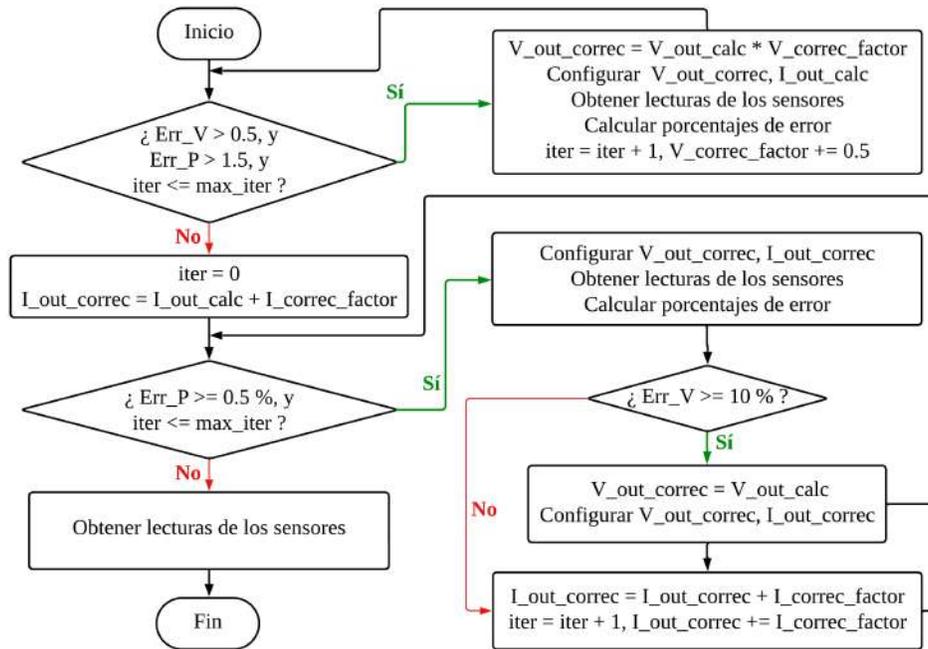


Figura 5.18: Función para corregir las salidas de los instrumentos del laboratorio.

Se inicia verificando que el porcentaje de error de la tensión y la potencia actuales son mayores que los mínimos requeridos, y que el número de iteraciones de ciclo actuales sea menor a un valor preestablecido; experimentalmente se determinó que por lo general después de 3 iteraciones no se aprecian cambios significativos en los porcentajes de error. Si se cumplen estas condiciones, se multiplica el valor de tensión calculado por el modelo analítico por un factor de corrección, que corresponde a 1.05 en un principio. Se configura el nuevo punto en la fuente de poder, se obtienen nuevas lecturas de los sensores y se calculan los porcentajes de error. Luego, se aumenta en uno el valor de iteraciones de ciclo, se añade un 5 % adicional al factor de corrección, y se vuelven a verificar las condiciones del ciclo para determinar si es necesario iniciar otra iteración. Cuando alguna de las condiciones deja de cumplirse, se resetea el valor del contador de iteraciones.

Seguidamente, se continúa con el ciclo para corregir el valor de corriente. Se inicia calculando el valor inicial de la corriente de corrección, el cual corresponde al valor de corriente calculado por el modelo analítico, más un valor de 0.01 A. Contrario a la tensión, en este caso se requieren hacer ajustes más pequeños, por lo que se opta por sumar un pequeño valor fijo de corriente. Para esto, se verifica si el porcentaje de error de la potencia sigue siendo más alto del ideal y si el número de iteraciones actuales es menor a máximo establecido, correspondiente a 5. Se configura el nuevo punto de operación en la fuente de poder, se obtienen lecturas de los sensores y se calcula el porcentaje de error. En algunos casos, resulta más conveniente solo ajustar la corriente y no la tensión, lo cual se manifiesta en un porcentaje

de error de tensión muy alto. Por lo tanto, se verifica si el porcentaje de error de la tensión supera el 10 %, y de ser el caso, de ahora en adelante se usa el valor de tensión calculado por el modelo analítico, y no el obtenido del ciclo de corrección de tensión. Luego, se procede a sumar 0.01 a la tensión de corrección actual, y se vuelven a verificar las condiciones del ciclo para determinar si es necesario realizar una nueva iteración. Una vez terminados ambos ciclos, se procede a obtener las lecturas finales de los sensores y se da por terminada la función.

En la Figura 5.19, se muestra una fotografía de la plataforma de SAS # 1, compuesta por los instrumentos del laboratorio (izquierda), un monitor donde se observa la interfaz de visualización y de control de la solución IoT (centro), y el ordenador donde se ejecuta el programa de Python (derecha).

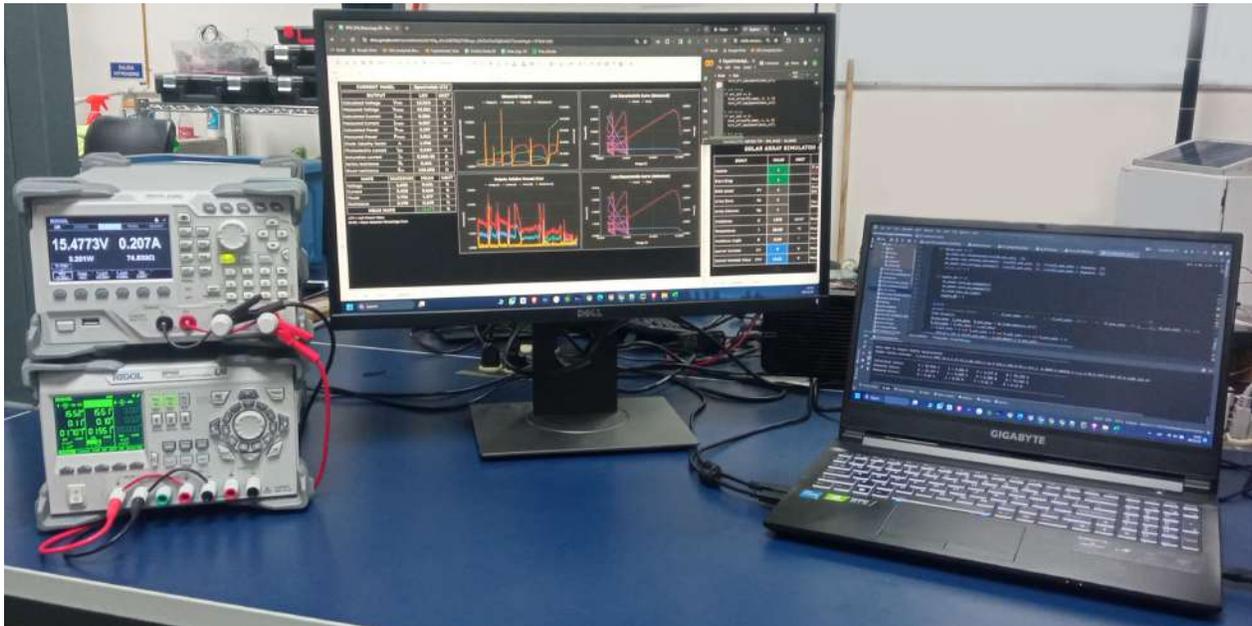


Figura 5.19: Fotografía de la plataforma de SAS # 1.

Plataforma de SAS #2

En la Figura 5.20, se presenta el diagrama de flujo del programa implementado en el MCU_S2, correspondiente a un NodeMCU ESP8266.

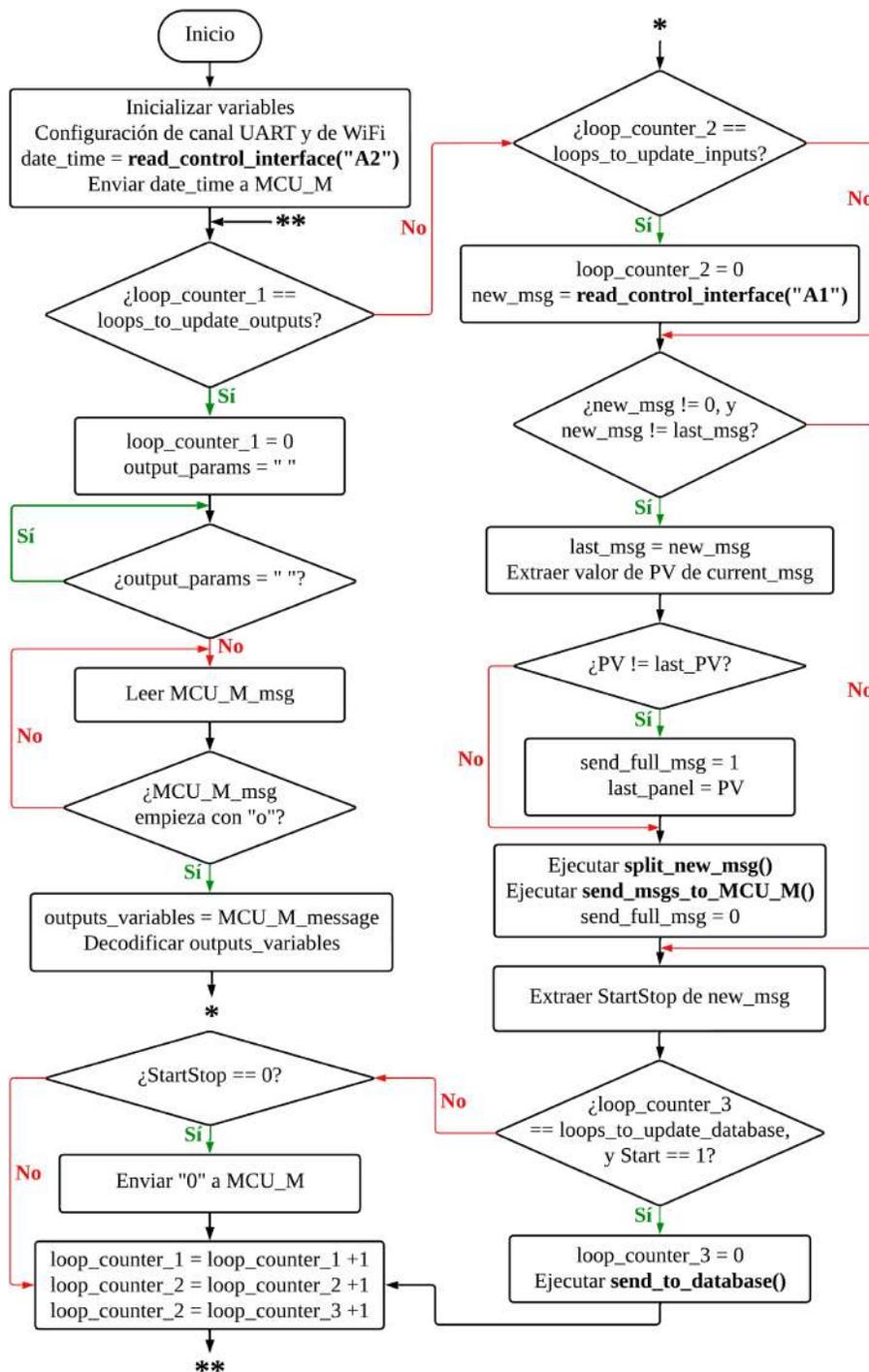


Figura 5.20: Diagrama de flujo del ciclo de control del MCU_S2 (ESP8266).

Se comienza inicializando las variables del programa, las cuales se encargan de almacenar los valores de las entradas ingresadas por el usuario, las salidas enviadas por el MCU_M y otros valores necesarios para la ejecución del ciclo, como contadores. También se realiza la configuración del canal UART que se utiliza para comunicarse con el MCU_M, así como la del WiFi para tener acceso a Internet. Luego, se envía una solicitud a la interfaz de control de la solución IoT para recibir el contenido de la celda A2, la cual contiene un mensaje como este: “t,14,1,24,10,33,40”, el cual indica el día, mes, año, hora, minuto y segundo, en ese orden, del momento en que se hizo la solicitud. Este mensaje se reenvía al MCU_M, con el fin de que se haga la configuración respectiva de fecha y hora en el RTC.

Luego, inicia propiamente el ciclo principal del programa. Primeramente, se verifica si el valor en el contador de ciclos #1 corresponde al número de iteraciones de ciclo necesarias para actualizar los valores de salida enviados por el MCU_M. De ser el caso, se resetea dicho contador y la variable *output_params* que se encarga de almacenar el texto enviado por el MCU_M (a través de UART) se coloca como vacía, de modo que se inicia un ciclo secundario que termina cuando se almacene un nuevo valor en dicha variable. Se monitorea el canal serial con el MCU_M hasta recibir un mensaje cuyo primer caracter es la letra “o”, la cual indica que el mensaje que está enviando contiene los valores de las salidas. Cuando llega este mensaje, se almacena en la variable indicada anteriormente, terminando el ciclo secundario. Luego, se decodifica el mensaje de las salidas, de modo que el valor de cada parámetro de salida se guarda en su respectiva variable dentro del programa. Más adelante, cuando se describe el funcionamiento del ciclo de control del MCU_M, se explica con más detalle el contenido de este mensaje.

Seguidamente, se verifica si el valor en el contador de ciclos #2 corresponde al número de iteraciones de ciclo necesarias para actualizar los valores de entrada del usuario. De ser el caso, se resetea dicho contador y se envía una solicitud a la interfaz de control de la solución IoT para recibir el contenido de la celda A2 y almacenarlo en la variable *new_msg*, el cual corresponde a un mensaje con el siguiente formato:

$$\begin{aligned}
 &“i, PV, N_s, N_p, G, T, \theta, CV, CVV, \\
 &d, T_{c,ref}, G_{ref}, V_{oc}, I_{sc}, V_{mp}, I_{mp}, P_{mp}, N_{s,pv}, K_v, K_i, E_g, \\
 &p, n, I_{ph,ref}, I_{o,ref}, R_{s,ref}, R_{sh,ref}”
 \end{aligned}$$

Cabe aclarar que se trata de valores numéricos, y las únicas letras en todo el mensaje corresponden a “i”, “d” y “p”, las cuales se utilizan como separadores entre los parámetros de entrada del usuario, los parámetros de la hoja de datos del panel solar seleccionado, y los cinco parámetros del modelo de un diodo (calculados en la componente #1 del modelo analítico con el método de Hussein), respectivamente.

Se verifica si *new_msg* es diferente a 0 (lo que indica que el usuario no desea que se actualicen los parámetros de entrada) y que sea diferente al último mensaje recibido del panel de control (lo que quiere decir que las entradas actuales ya fueron procesadas debidamente en un ciclo anterior). Si se cumplen ambas condiciones, se extrae el valor de *PV*, que indica el panel solar deseado por el usuario, y se compara con el último panel solar recibido. Si es igual, quiere decir que ya los parámetros del panel solar fueron enviados al MCU_M previamente, por lo que no es necesario enviar la totalidad de *new_msg* al MCU_M ya que los únicos parámetros que pueden haber cambiado están antes de la letra “d”. No obstante, si se trata de un nuevo panel, sí se debe enviar la totalidad de los parámetros, lo cual se indica cambiando el valor de la variable *send_full_msg* de 0 (su valor predeterminado) a 1. Luego, se ejecuta la función *split_new_msg()*, la cual se encarga de dividir el contenido de *new_msg* en tres variables separadas; esto debido a que el mensaje original es muy largo y se perdería parte de su contenido si se envía en un solo mensaje. Por ende, se obtienen las siguientes variables, junto a su respectivo contenido:

$$\begin{aligned} \text{input_params} &= “i, PV, N_s, N_p, G, T, \theta, CV, CVV” \\ \text{datasheet_params} &= “d, T_{c,ref}, G_{ref}, V_{oc}, I_{sc}, V_{mp}, I_{mp}, P_{mp}, N_{s,pv}, K_v, K_i, E_g” \\ \text{five_params} &= “p, n, I_{ph,ref}, I_{o,ref}, R_{s,ref}, R_{sh,ref}” \end{aligned}$$

Después, se ejecuta *send_msgs_to_MCU_M()*, la cual envía los mensajes que correspondan al MCU_M, de acuerdo al valor de *send_full_msg*; más adelante, se describe con mayor detalle la forma en se hace. Cuando termina de ejecutarse la función, se resetea el valor de *send_full_msg*.

Luego, se extrae el valor de la variable *StartStop* de *new_msg*, que indica si el usuario desea mantener en operación el sistema (valor igual a 1) o no (valor igual a 0). Si es igual a 1 y además se cumple con la condición de que el valor del contador de ciclos #3 corresponde al número de iteraciones establecidas para actualizar los registros de la solución IoT, se procede a ejecutar la función *send_to_database()* para enviar una solicitud con los valores actuales de todos los parámetros de interés. En caso de que el valor de *StartStop* sea igual a 0, se envía al MCU_M el mensaje “0” para que coloque las salidas del convertidor no entreguen potencia. Finalmente, se suma 1 a todos los contadores de ciclos usados en el programa y se inicia un nuevo ciclo.

En la Figura 5.21, se presenta el diagrama de flujo de la función *send_msgs_to_MCU_M()*. Se inicia verificando el valor de la variable *send_full_msg*. Si este corresponde a 0, Se inicia un ciclo principal, que termina cuando el MCU_M notifica que la transmisión de información fue completada mediante el mensaje “c”. Primeramente, el MCU_S2 envía el mensaje “s”, para notificarle al MCU_M que solo se va a transmitir un mensaje. Este, en respuesta envía el mensaje “r”, para notificar que está listo para recibir la información. Luego, el MCU_S2 envía el contenido de *input_params*, hasta que el MCU_M envía de vuelta el mensaje “c”, dando por terminado el ciclo. Luego, se resetean las variables utilizadas para guardar los mensajes, *waiting_r* y *MCU_M_msg*, y se da por terminada la función.

Por otra parte, si el valor de *send_full_msg* es igual a 1, se inicia un ciclo principal similar al anterior, que termina cuando se recibe el mensaje “c” por parte del MCU_M. En este caso, el MCU_S2 envía el mensaje “a”, para notificarle al MCU_M que se van a transmitir los tres mensajes correspondientes, y este nuevamente confirma que está listo para recibirlos enviando el mensaje “r”. Luego, el MCU_S2 procede a enviar el contenido de *input_params* hasta recibir el mensaje “i” por parte del MCU_M, confirmando que los recibió exitosamente. Luego, se realiza el mismo procedimiento para mandar el contenido de *datasheet_params* y *five_params*, en este caso recibiendo los mensajes de confirmación “d” y “p”, respectivamente, por parte del MCU_M. Cuando los tres mensajes fueron recibidos, se espera a que el MCU_M envíe el mensaje “c” para indicar que se completó la transmisión de todos los parámetros y se da por terminado el ciclo principal. Finalmente, se resetea el valor de las variables utilizadas para guardar mensajes, es decir, *waiting_r*, *waiting_i*, *waiting_d*, *waiting_p* y *MCU_M_msg*, y se da por terminada la función.

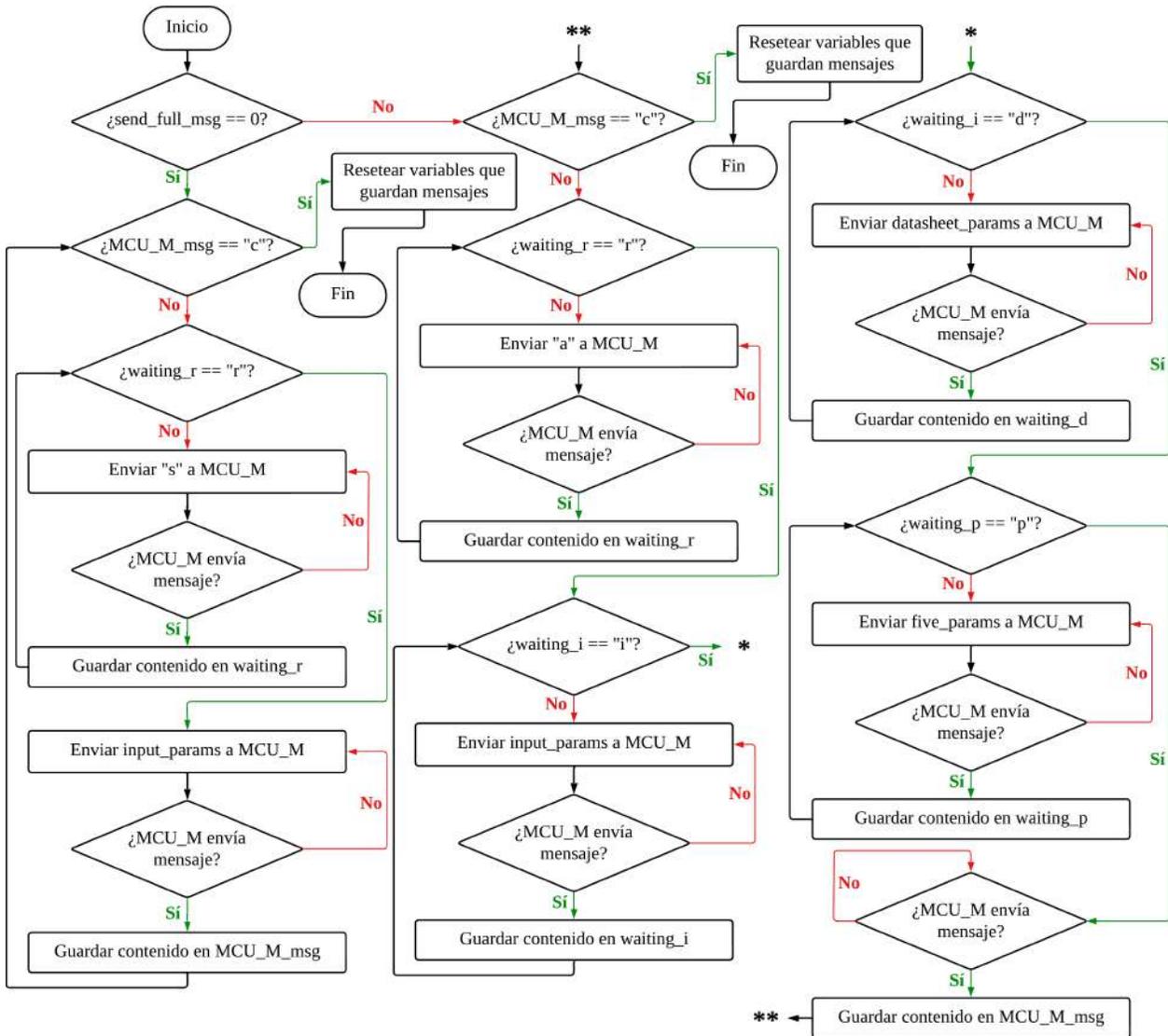


Figura 5.21: Diagrama de flujo de la función *send_msgs_to_MCU_M()*.

En la Figura 5.22, se presenta el diagrama de flujo del programa implementado en el MCU_M, correspondiente a un NUCLEO-64 L073RZ. Se comienza inicializando las variables que se utilizan para almacenar los valores de los parámetros provenientes del MCU_S2 y las lecturas de los sensores de tensión y corriente, provenientes del MCU_S1, así como aquellas utilizadas para la ejecución del programa, como contadores. Luego, se realiza la configuración de los distintos dispositivos conectados al MCU_M, como la placa del RTC, el lector Micro SD, la pantalla LCD, y los dos canales UART utilizados para mantener comunicación bidireccional con los otros microcontroladores.

Luego, se espera a que llegue un mensaje del MCU_S2 que inicie con la letra “t” y se ejecuta la función *RTC_config()*, en la que primero descompone el mensaje para extraer el valor de cada parámetro necesario para configurar la placa RTC externa. Una vez hecho esto, se imprime la fecha y hora en la pantalla LCD, permitiendo el usuario verificar que el MCU_S2 tiene acceso a Internet y que existe intercambio de información entre los microcontroladores.

Seguidamente, inicia el ciclo principal del programa. Primero, se ejecuta la función *check_msg_from_MCU_S2()*, en la cual se verifica si el MCU_S2 está solicitando el envío de mensajes, y de ser así, se realiza la secuencia de pasos requerida para recibirlos y hacer el procesamiento requerido para obtener todos los parámetros de entrada, provenientes de la solución IoT. Esta secuencia se explica con mayor detalle más adelante.

Luego, se ejecuta la función *Analytical_Model()*, explicada anteriormente con el diagrama de flujo de la Figura 5.13. Esta función permite obtener los valores de *I_out_calc* y *V_out_calc*, los cuales contienen el punto de operación esperado en las salidas del arreglo solar. Estas dos variables son enviadas al MCU_S1, con el fin de que este realice los ajustes necesarios en el convertidor DC-DC para obtener las salidas requeridas.

Después, se verifica si el MCU_S2 está enviando mensajes, en cuyo caso se realiza la secuencia de pasos requerida para recibirlos y procesarlos para obtener las lecturas de los sensores de corriente y tensión del convertidor, las cuales se almacenan en las variables *I_out_meas* y *V_out_meas*.

Se procede a verificar si el valor del contador de ciclos *loops_counter* es el necesario para actualizar los valores medidos a la salida del convertidor, en cuyo caso, se resetea el contador y se imprimen los valores de *I_out_meas*, *V_out_meas* y *I_out_meas* y *P_out_meas* (este último es calculado indirectamente con el valor de los dos primeros), se envía además todos los parámetros de salida al MCU_S2, lo cual incluye tanto los valores medidos por los sensores, como los calculados por el modelo analítico. Asimismo, se registran dichos parámetros en un archivo de texto dentro de la memoria Micro SD. Finalmente, se aumenta en 1 el valor de *loops_counter*, y se inicia un nuevo ciclo.

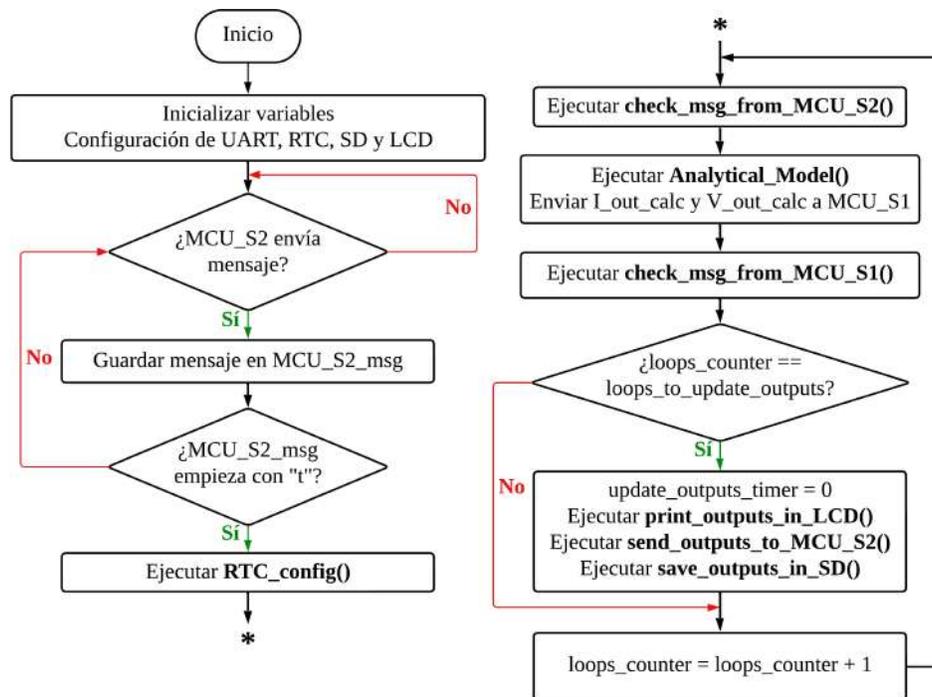


Figura 5.22: Diagrama de flujo del programa principal implementado en el MCU_M.

En la Figura 5.23, se presenta el diagrama de flujo de la función *check_msg_from_MCU_S1()*, la cual verifica si el MCU_S1 envía mensajes, y de ser caso, verifica que cumpla con un mínimo de caracteres antes de ejecutar la función *load_MCU_S1_msg()*, la cual extrae los valores de cada salida y las asigna a sus respectivas variables (*I_out_meas* y *V_out_meas*).

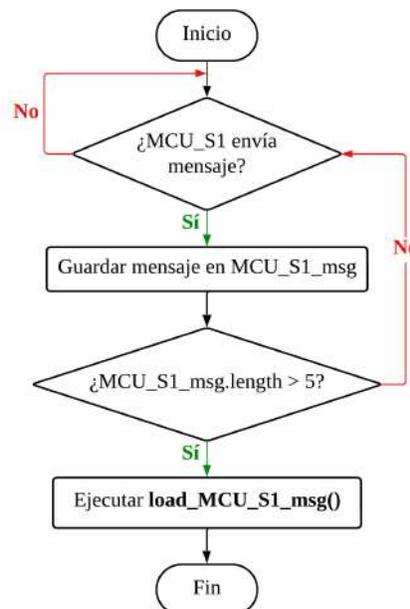


Figura 5.23: Diagrama de flujo de la función *check_msg_from_MCU_S1()*.

En la Figura 5.24, se presenta el diagrama de flujo de *check_msg_from_MCU_S2()*. Esta función es el complemento de la función *send_msgs_to_MCU_M()* de la Figura 5.21, por lo tanto, su funcionamiento ya fue explicado anteriormente. Al final de la función, se puede apreciar que, dependiendo del caso, se ejecuta la función *load_only_input_params()* o *load_all_params()*, las cuales extraen de los mensajes resultantes (input_params, datasheet_params y five_params) el valor de cada parámetro y lo asigna a su respectiva variable dentro del programa para que puedan ser usados por la función *Analytical_Model()*.

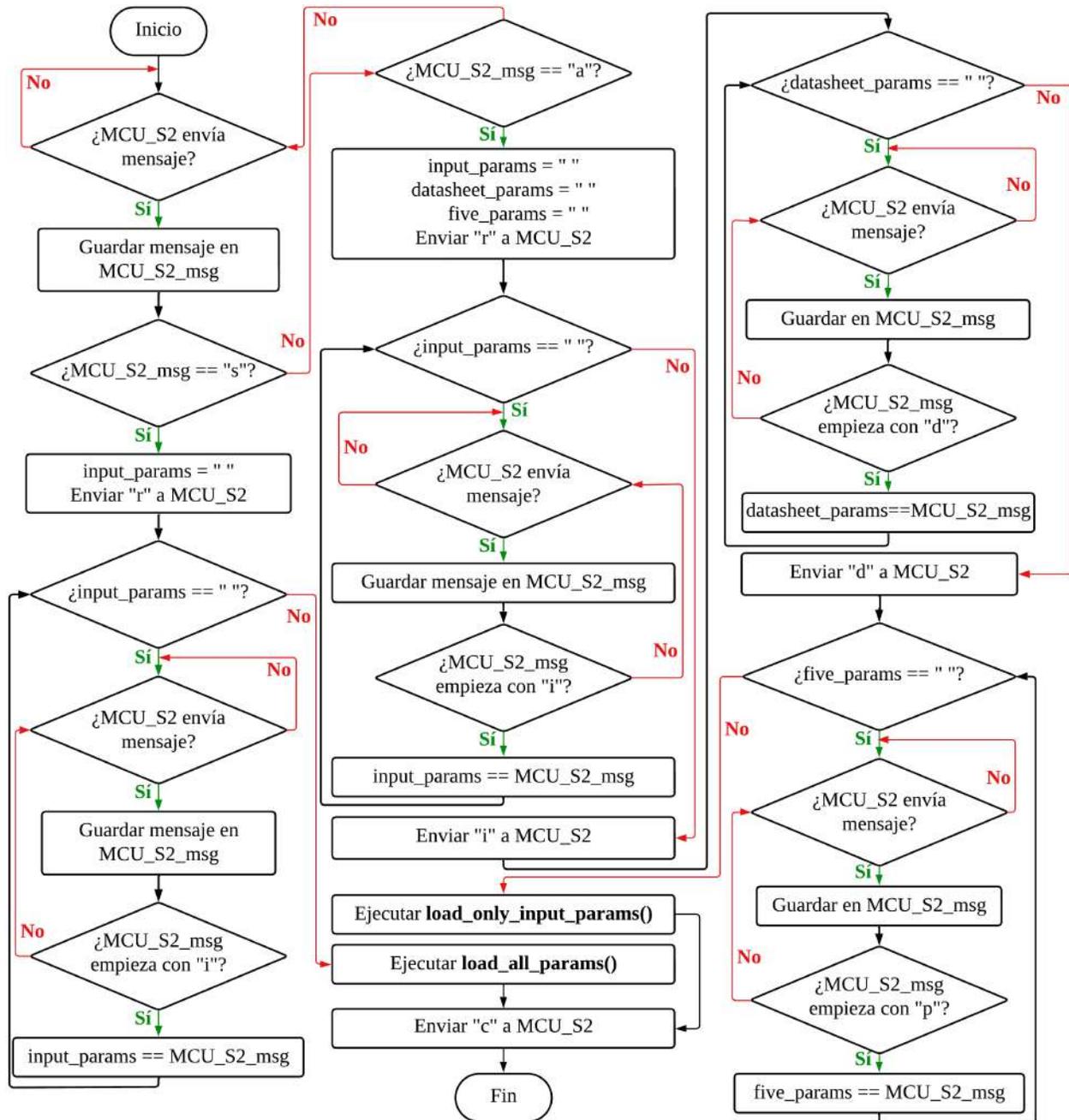


Figura 5.24: Diagrama de flujo de la función *check_msg_from_MCU_S2()*.

En la Figura 5.24, se presenta el diagrama de flujo del programa implementado en el MCU_S1. Primero, se inicializan las variables, se configura el canal UART con el que se mantiene comunicación con el MCU_M, y se establece la semilla para el generador de números aleatorio (o RNG, por sus siglas en inglés). Luego se inicia con el ciclo del programa, en el cual se verifica si llegan mensajes del MCU_M, y de ser el caso, se guarda su contenido en *MCU_M_msg* y se comprueba si cumple con una longitud mínima, con el fin de descartar que sea ruido. Si cumple, se ejecuta la función *load_MCU_M_msg*, la cual descompone el mensaje para obtener los valores correspondientes a V_{out} e I_{out} , y se cambia el valor de la variable *first_msg_received* a 1. Después, se verifica que el valor de dicha variable sea 1, y de ser el caso, se procede a generar dos números aleatorios entre 0 y el 0.5% de V_{out} e I_{out} , respectivamente. Luego, se les asigna un signo aleatorio, y se suman estos números aleatorios a los valores originales, almacenándolos en las variables V_{out_meas} e I_{out_meas} . Finalmente, se envían estos valores, separados entre sí por una coma, hacia el MCU_M, y se da por terminada la iteración del ciclo. De esta forma, se pretende emular el comportamiento esperado de la planta del MCP, la cual recibe un punto de operación del MCU_M, y le devuelve las lecturas de los sensores de tensión y corriente.

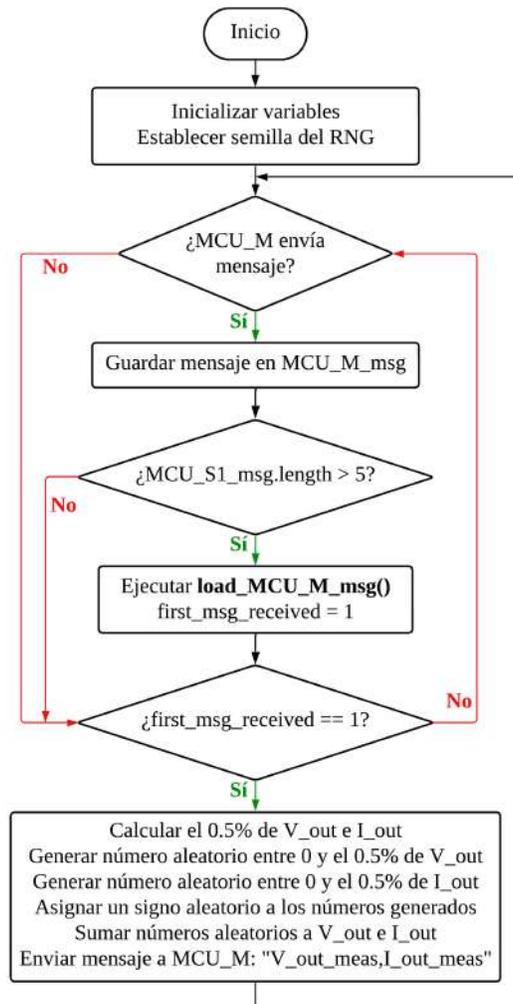


Figura 5.25: Diagrama de flujo del programa implementado en el MCU_S1.

En el Anexo B, se presenta el enlace de un repositorio de GitHub donde se encuentran todos los programas descritos anteriormente para ambas componentes del modelo analítico, y ambas plataformas de SAS.

En la Figura 5.26 se presenta una fotografía de la placa de desarrollo confeccionada, donde se soldaron a mano, en una placa perforada de 150 mm x 90 mm, conectores en fila tipo hembra en los que se pueden colocar o retirar los microcontroladores MCU_M y MCU_S2, el RTC y el lector Micro SD. Además, se agregaron dos interruptores tipo DIP, de dos bits cada uno, que permiten desconectar los canales de comunicación UART que el MCU_M mantiene con el MCU_S1 y MCU_S2, y dos interruptores de tipo SPDT, que permiten desconectar la entrada de potencia hacia el MCU_M y MCU_S2. Asimismo, se colocaron bloques de terminales para conectar la pantalla LCD, y otros conectores en fila tipo hembra donde se conecta la entrada de potencia y los canales UART provenientes del MCU_S1. En el Anexo E, se presenta el diagrama de conexiones de la placa, así como la lista de materiales y componentes con su respectivo costo.

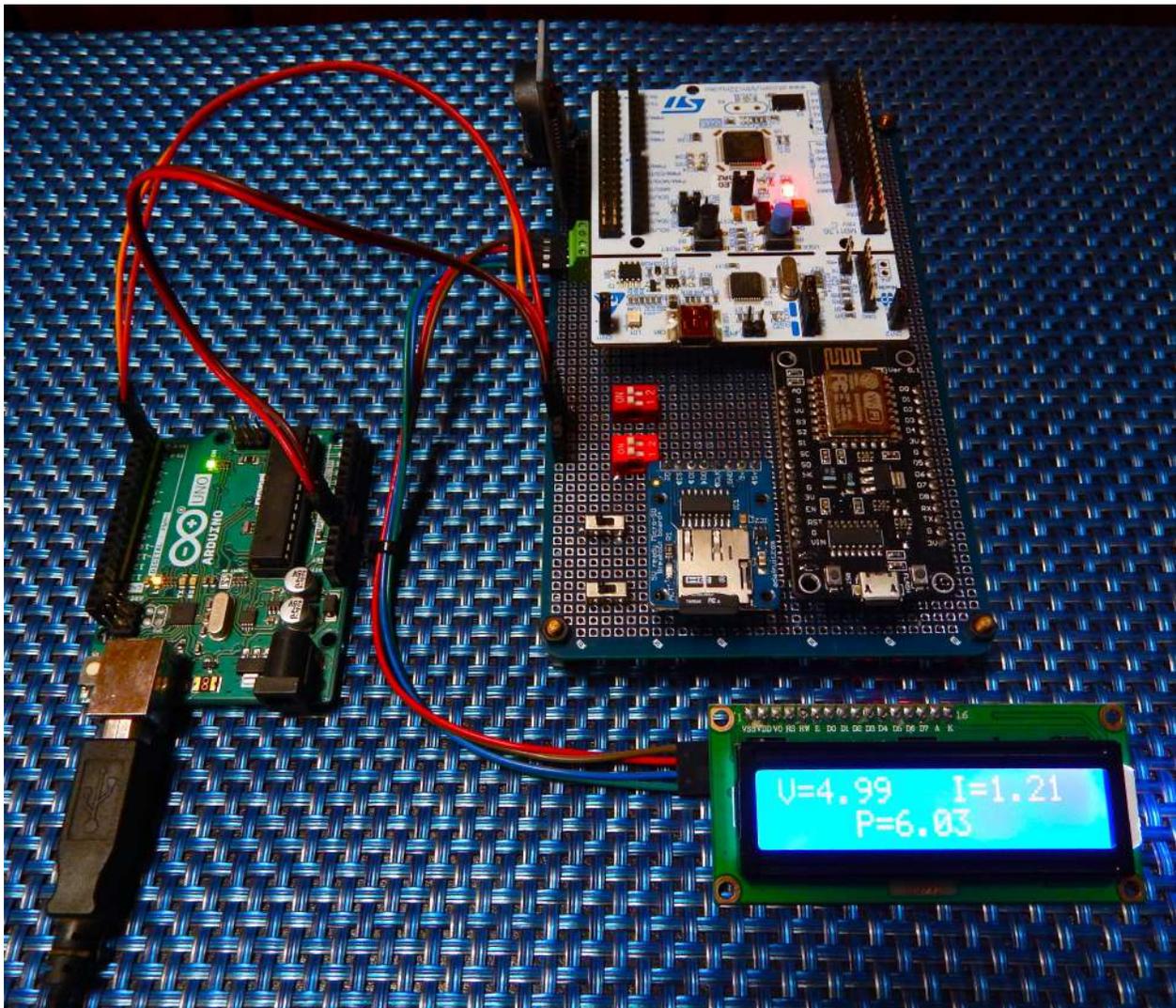


Figura 5.26: Placa de desarrollo confeccionada.

5.3. Validación de conceptos

5.3.1. Modelo analítico: Componente #1

Obtención de sets de datos con visión computacional

En la Figura 5.27, se presentan las imágenes de referencia de las curvas características I-V para panel solar, las cuales fueron tomadas de sus hojas de datos o de la literatura encontrada, según se indica en la Tabla 5.1.

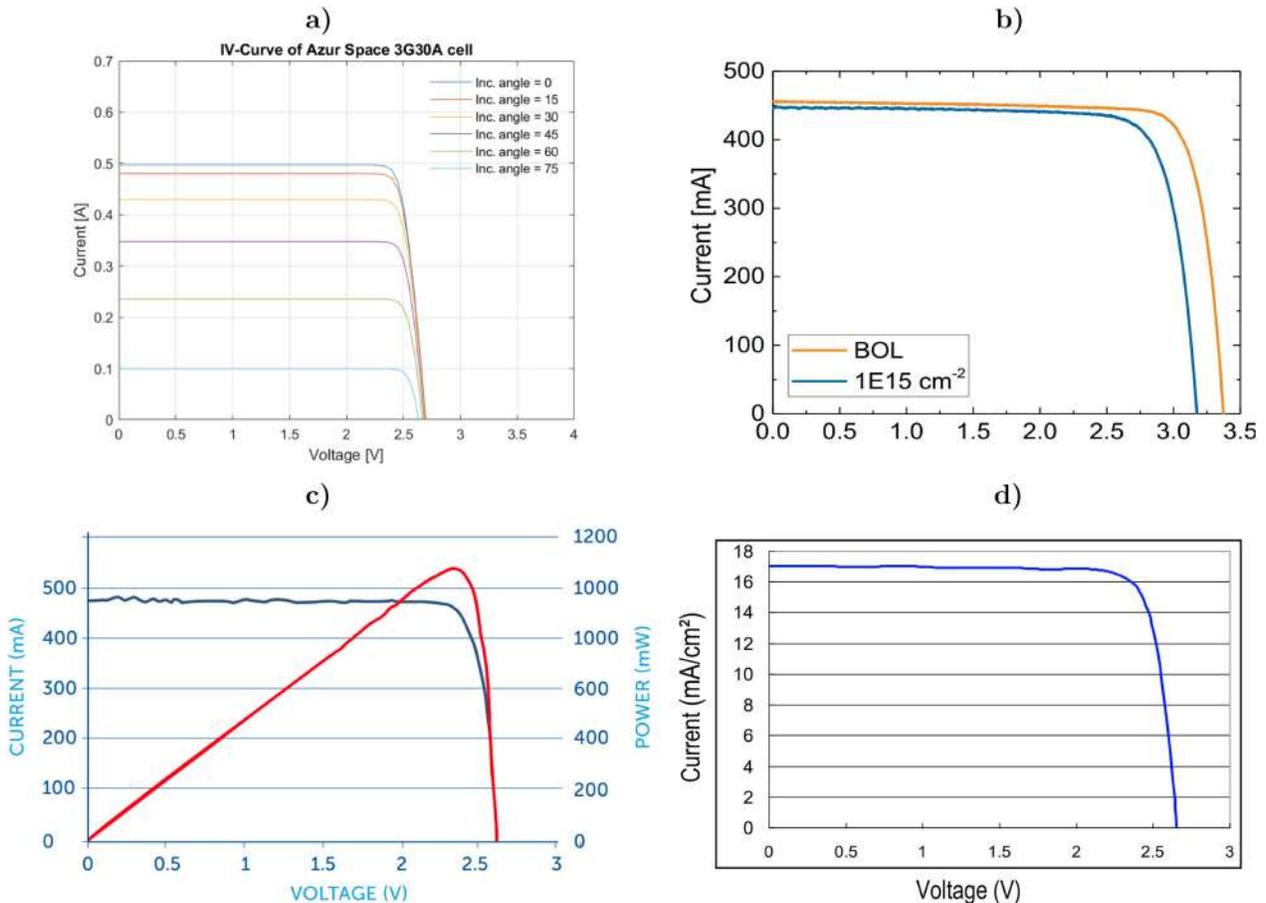


Figura 5.27: Imágenes de curvas características originales. a) AZUR 3G30A b) AZUR 4G32C c) CESI CTJ30 d) SpectroLab UTJ.

Es necesario hacer ajustes en cada imagen previo a utilizarlos en el algoritmo de visión computacional, con el fin de aislar únicamente la curva de interés, y además quitar los marcos de los ejes para que estos no sean contados como puntos de la curva.

En la imagen del AZUR 3G30A se puede apreciar que hay 6 curvas para diferentes valores de ángulo de incidencia. No obstante, la única de interés corresponde a la que tiene un ángulo de 0°, puesto que este es el ángulo estudiado en las hojas de datos. Asimismo, en la del AZUR 4G32C hay dos curvas, sin embargo la de interés corresponde a la de color amarillo, puesto que corresponde a la de una celda al inicio de su vida útil a condiciones estándar de irradiancia

de 1367 W/m^2 . En el caso del CESI CTJ30, se tiene en un mismo gráfico la curva I-V y la P-V, pero solo se utiliza la primera. Además, se puede apreciar que de 0 V a 2 V la curva presenta mucho ruido, lo cual puede causar problemas a la hora de calcular los parámetros por el método de Hussein, por lo tanto, al editar la imagen también se debe intentar corregir estas irregularidades. Finalmente, en la del SpectroLab UTJ, se deben remover las líneas color negro horizontales que demarcan los puntos del eje y . Adicionalmente, se debe notar que las unidades del eje y están en mA/cm^2 , por lo que se debe tomar el valor de área del panel, de la hoja de datos, para hacer la conversión a solo amperios.

En la Figura 5.28, se presentan las imágenes luego de ser editadas con el software Adobe Photoshop y listas para utilizarse como insumos del algoritmo de visión computacional. Para fines de visualización en este informe, se les coloca un marco negro para que sea más claro cuáles son los límites de cada imagen, no obstante, las imágenes a utilizar en el algoritmo no cuentan con ningún borde. Además, se puede observar que las imágenes del AZUR 3G30A y del CESI CTJ30 fueron recortadas para considerar como límites del eje y 0.5 A.

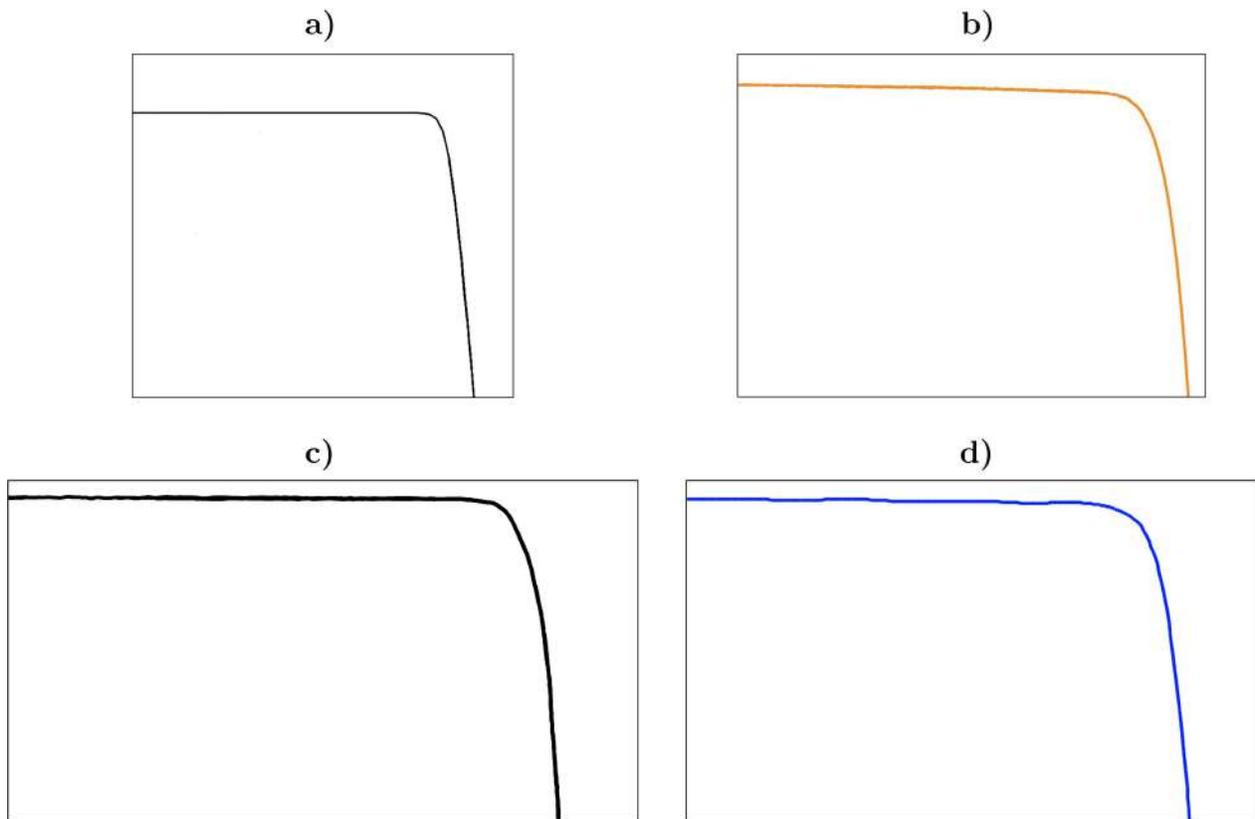


Figura 5.28: Imágenes de curvas características editadas. a) AZUR 3G30A b) AZUR 4G32C c) CESI CTJ30 d) SpectroLab UTJ.

En la Figura 5.29, se presenta la imagen editada del AZUR 3G30A utilizada como referencia, la gráfica creada a partir del set de datos extraído por el algoritmo de visión computacional, la cual cuenta con más de 500 puntos de muestreo, y ambas curvas superpuestas, con el fin de verificar visualmente que coincidan.

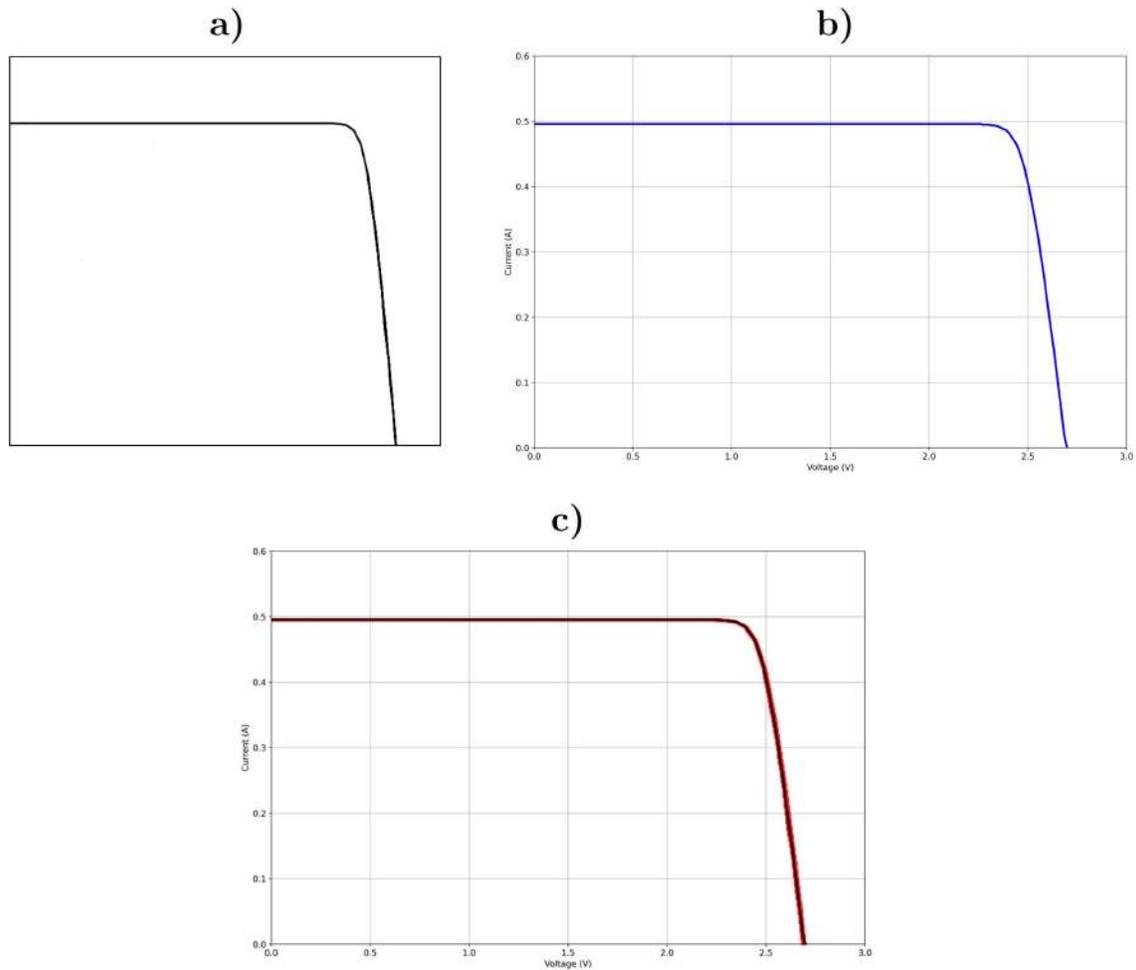


Figura 5.29: Comparación de curvas del panel solar AZUR 3G30A. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.

Se puede apreciar que a la curva de referencia se le aumentó el grosor y su color se cambió a rojo, con el fin de aumentar su visibilidad cuando se coloca la curva generada con el set de datos, la cual es de color negro. Se realizó esta misma comparación con los demás paneles solares, según se observa en las Figuras 5.30, 5.31 y 5.32.

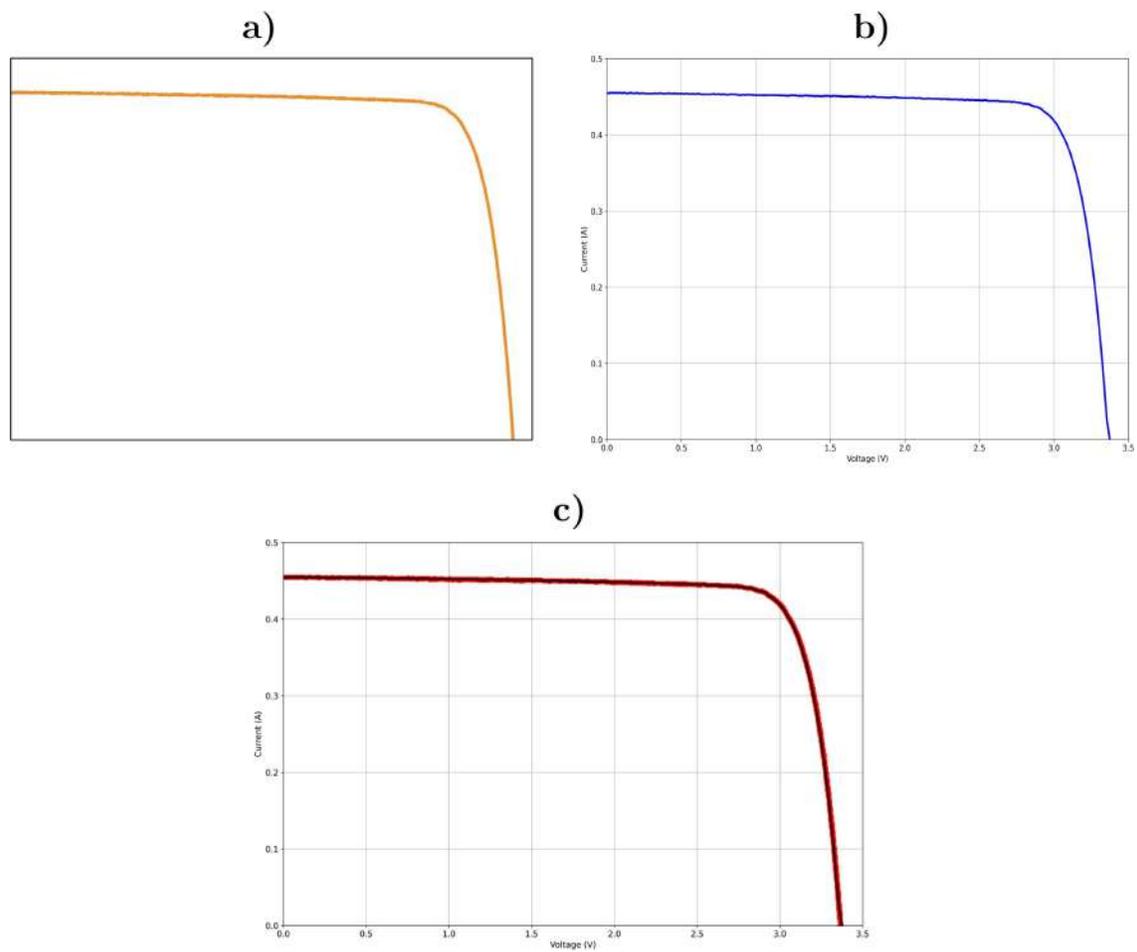


Figura 5.30: Comparación de curvas del panel solar AZUR 4G32C. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.

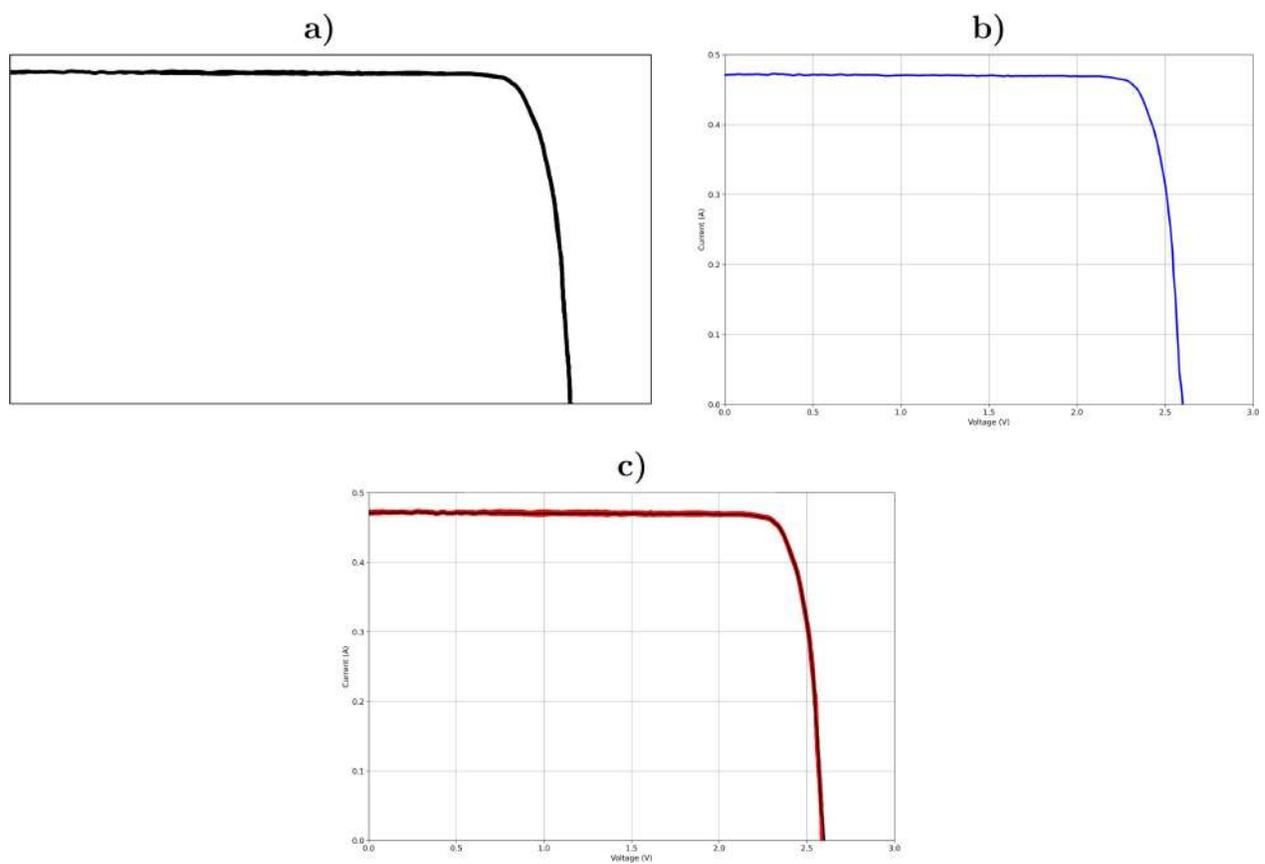


Figura 5.31: Comparación de curvas del panel solar CESI CTJ30. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.

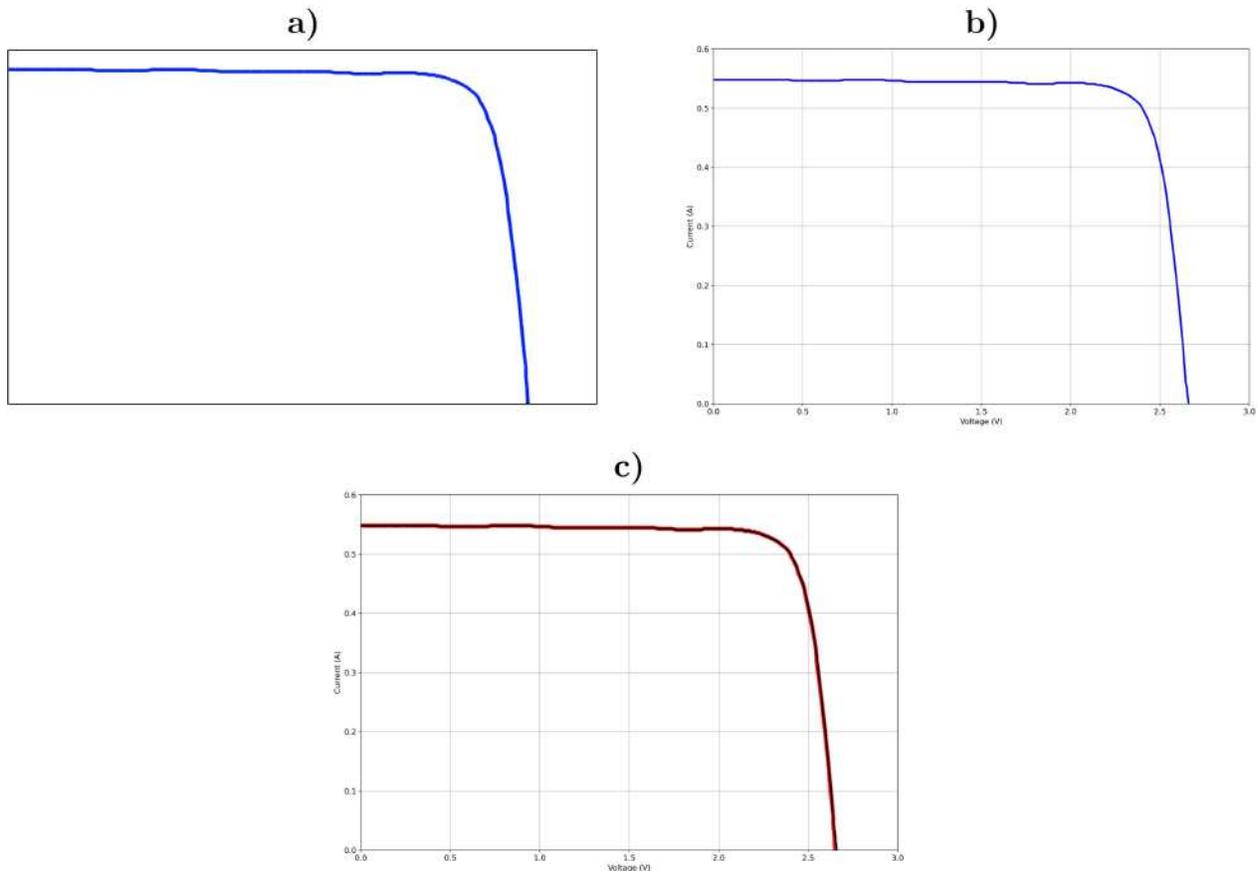


Figura 5.32: Comparación de curvas del panel solar SpectroLab UTJ. a) Curva I-V original b) Curva generada con el set de datos del algoritmo c) Curvas superpuestas.

En todos los casos, se puede apreciar que visualmente los sets de datos generados replican perfectamente las curvas deseadas, puesto que la línea negra se mantiene centrada sobre el contorno rojo. Por lo tanto, se considera la implementación del algoritmo de visión computacional como exitoso. Los sets de datos generados se pueden encontrar en la carpeta de *Google Drive* facilitada en el Anexo B.

Por otra parte, en la Tabla 5.6, se presentan los tiempos que le tomó al algoritmo de visión computacional extraer los sets de datos de cada imagen, donde se puede apreciar que entre mayor sea el número de píxeles de la imagen, mayor es el número de puntos que extrae y por ende tarda más en procesarse.

Tabla 5.6: Tiempos de ejecución del algoritmo de visión computacional.

Parámetro	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Segundos de ejecución	0.4500	0.4659	1.0444	0.7419
Cantidad de puntos	573	696	549	845
Píxeles de imagen	636 x 574	719 x 527	2498 x 1359	953 x 573

Implementación del método de Hussein

En la Tabla 5.7, se presentan los tiempos de ejecución del método de Hussein en cada uno de los paneles solares. Se puede notar que el tiempo está directamente relacionado con el número de puntos en el set de datos utilizado de referencia, según se mostró anteriormente en la Tabla 5.6.

Tabla 5.7: Tiempos de ejecución del método de Hussein.

Parámetro	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Segundos de ejecución	5.5706	7.4196	5.5010	8.6380

En la Figura 5.33, se presenta una gráfica donde se puede observar la forma en que el RMSE y MAPE varía en cada iteración del método de Hussein conforme se aumenta el factor de idealidad del diodo para el panel solar AZUR 3G30A. En ambas series se puede observar un comportamiento visiblemente parabólico. Además, se resaltan los valores mínimos de cada serie, lo cual permite identificar que el valor mínimo de MAPE se logra con un valor de n igual a 1.17, mientras que el valor mínimo de RMSE se logra con un n igual a 1.23. Dado que el RMSE fue el indicador utilizado por el autor para la selección del mejor conjunto de parámetros, se selecciona entonces el conjunto asociado al n de 1.23.

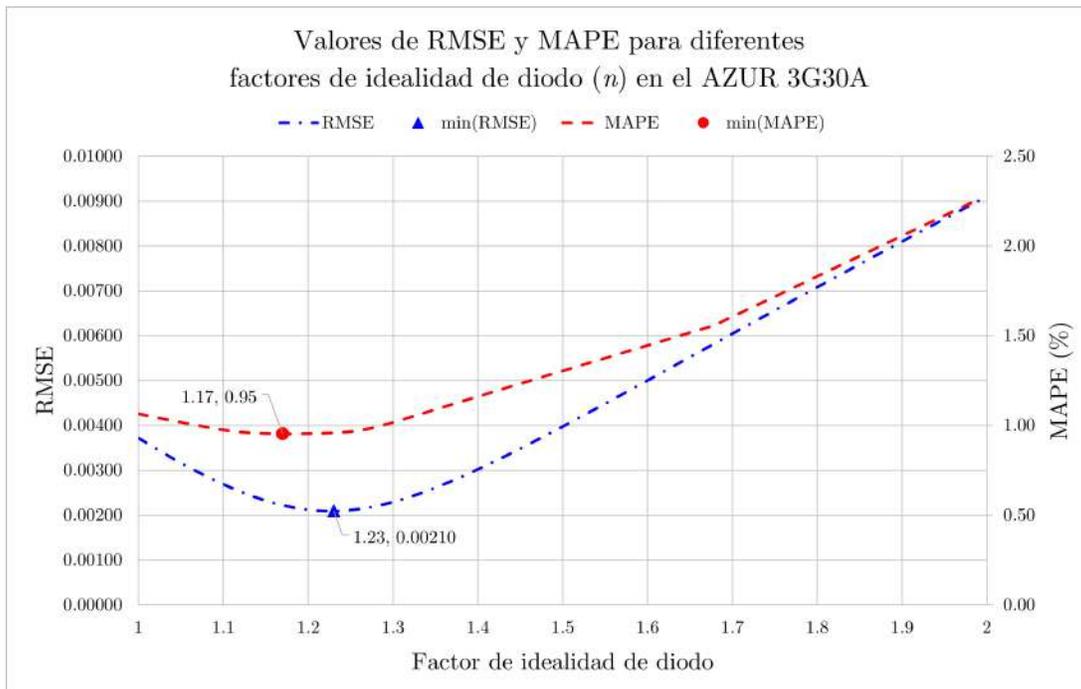


Figura 5.33: Valores de RMSE y MAPE para diferentes valores de n en el AZUR 3G30A.

En las Figuras 5.34, 5.35 y 5.36 se presentan las mismas gráficas pero para los paneles solares AZUR 4G32C, CESI CTJ30 y Spectrolab UTJ, respectivamente. En todas ellas se observa un comportamiento más lineal hasta encontrar los valores mínimos de MAPE y RMSE con un factor de idealidad igual a 1.99 en los tres casos.

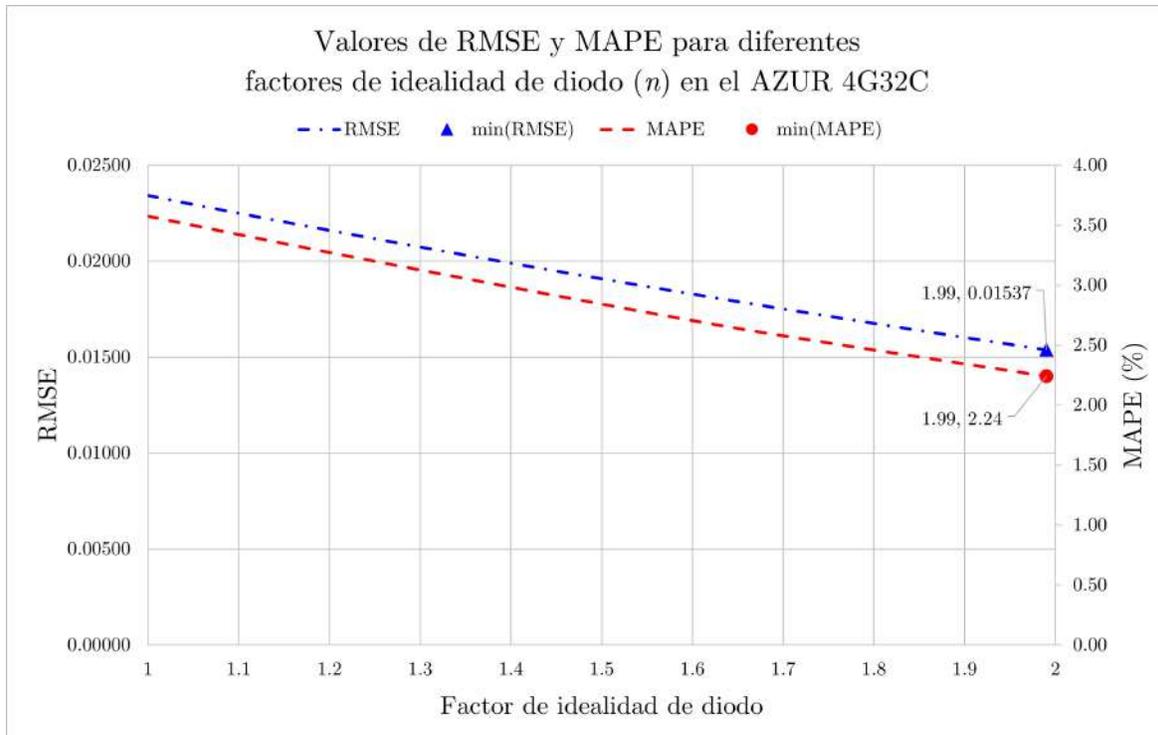


Figura 5.34: Valores de RMSE y MAPE para diferentes valores de n en el AZUR 4G32C.

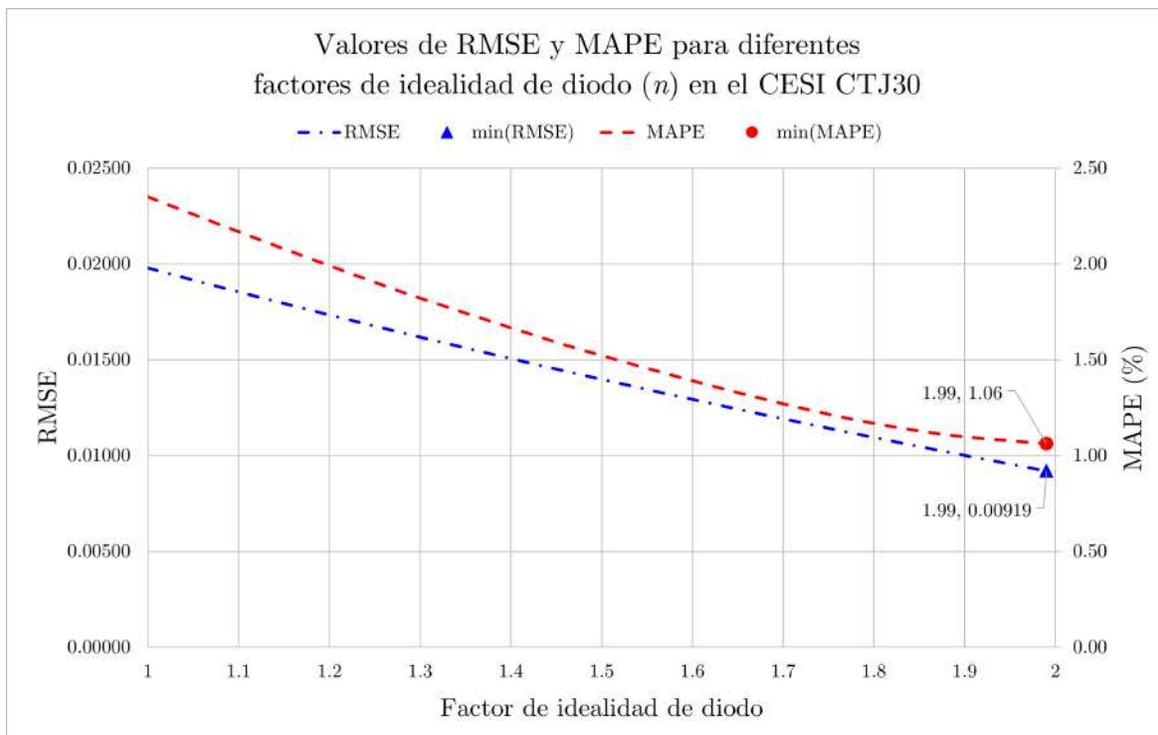


Figura 5.35: Valores de RMSE y MAPE para diferentes valores de n en el CESI CTJ30.

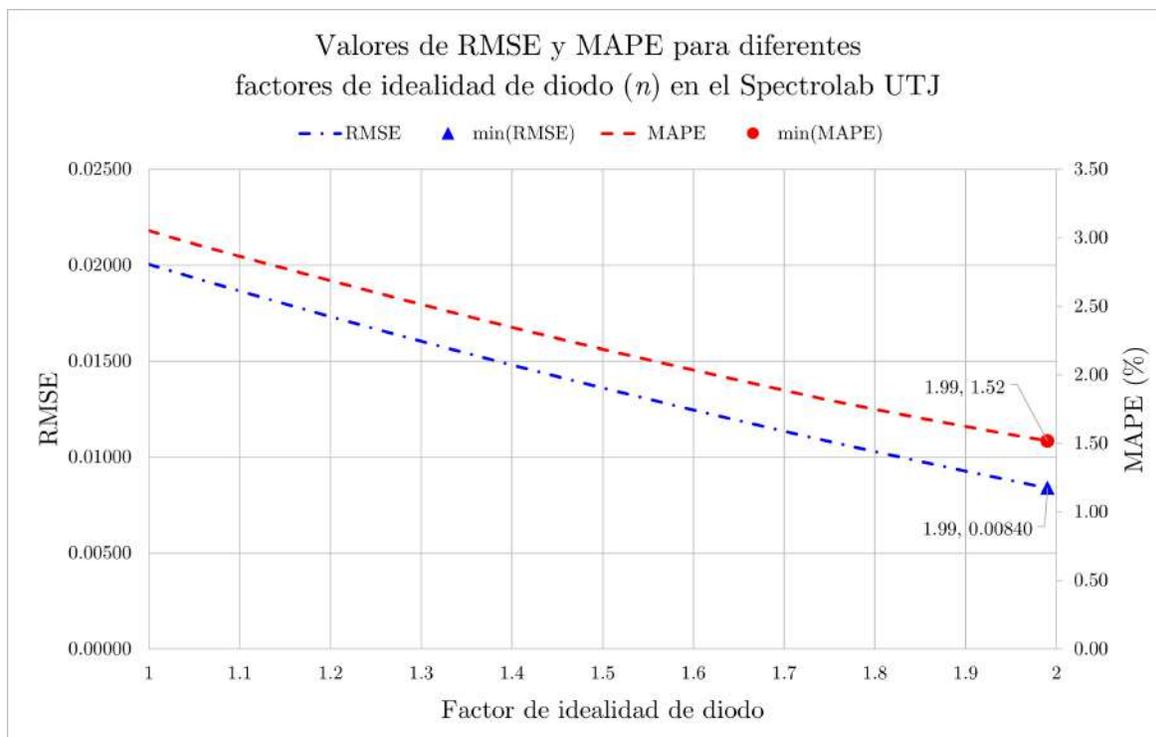


Figura 5.36: Valores de RMSE y MAPE para diferentes valores de n en el Spectrolab UTJ.

En la Tabla 5.8, se resumen los parámetros obtenidos para cada panel solar y los indicadores de rendimiento asociados. Se definió que el modelo analítico implementado debía presentar un MAPE inferior al 5%. Dado que el promedio de MAPE en los cuatro paneles es de 1.44%, se cumple satisfactoriamente con el objetivo.

Tabla 5.8: Parámetros del modelo de diodo único obtenidos mediante el método de Hussein.

Parámetros	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
n	1.23	1.99	1.99	1.99
I_{ph} (A)	0.4960	0.4562	0.4731	0.5490
I_o (A)	9.44E-38	9.45E-30	3.86E-23	2.26E-23
R_s (Ω)	0.3555	0.5705	0.2132	0.2408
R_{sh} (Ω)	996.98	207.91	677.70	138.29
RMSE	0.00210	0.01537	0.00919	0.00840
nRMSE	0.00423	0.03378	0.01944	0.01533
R^2	0.99931	0.94778	0.97751	0.98918
MAPE (%)	0.95947	2.24159	1.06213	1.51747

En la Figura 5.37, se presentan diagramas de barras que facilitan la comparación visual de los indicadores de rendimiento obtenidos para cada panel solar.

Comparación de indicadores de rendimiento entre los paneles solares estudiados

■ AZUR 3G30A ■ AZUR 4G32C ■ CESI CTJ30 ■ Spectrolab UTJ

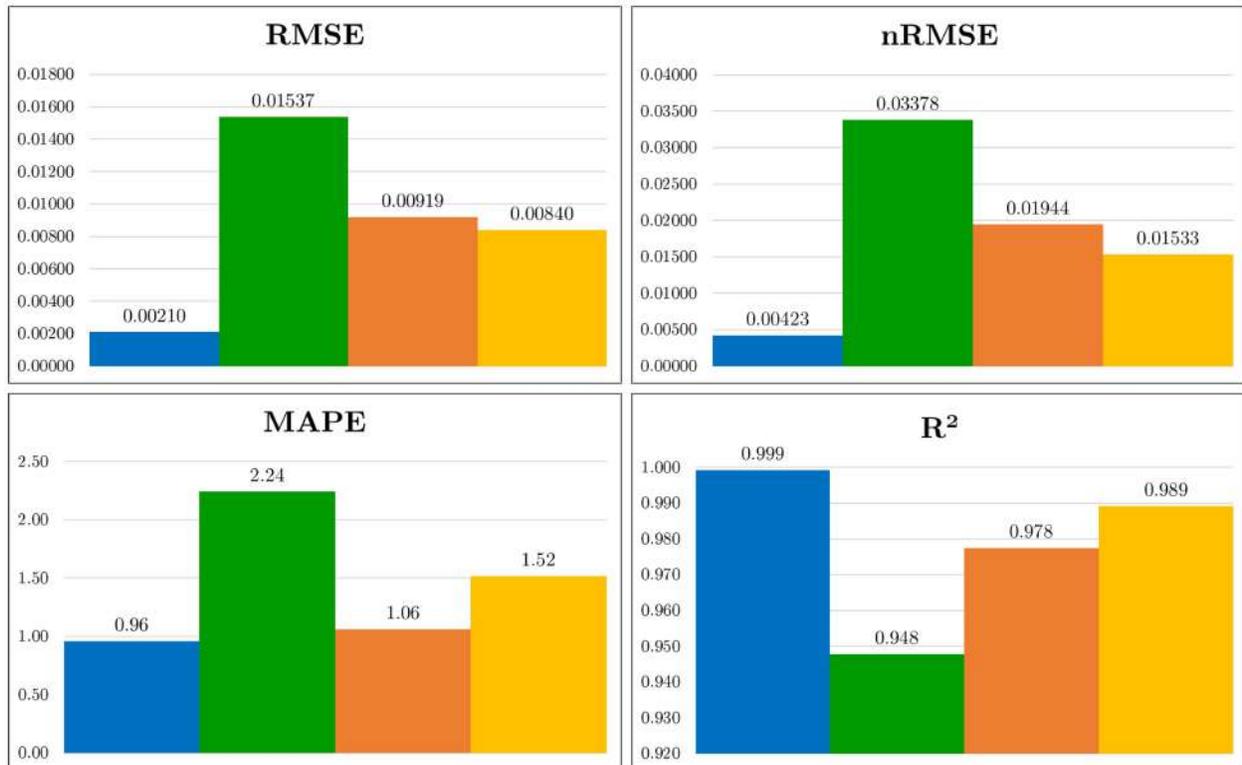


Figura 5.37: Diagramas de barras de los indicadores de rendimiento de los paneles solares estudiados.

Se puede apreciar que en todos los indicadores, el mejor y peor rendimiento lo presentan el AZUR 3G30A y el AZUR 4G32C, respectivamente. Por otra parte, en el RMSE se puede apreciar que el CESI CTJ presenta un rendimiento levemente superior al del Spectrolab UTJ, sin embargo, al hacer la normalización, la diferencia se hizo más significativa a favor del CESI CTJ, puesto que el rango de corriente del Spectrolab UTJ es más amplio. En el MAPE la diferencia es aún más notoria, aunque esto puede deberse a que en el Spectrolab fueron más prevalentes valores de ARPE muy altos cuando los valores se aproximan a cero, tal y como se observa más adelante cuando se presentan las curvas características. Finalmente, dado que los valores de R^2 son más sencillos de entender luego de ver el ajuste de las curvas calculadas respecto a las de referencia, esto también se discute más adelante.

En las Figuras 5.38, 5.39, 5.40 y 5.41, se presentan las curvas características I-V obtenidas para cada panel. En estas, se utilizan los sets de datos obtenidos mediante el algoritmo de visión computacional para trazar la curva de referencia, mientras que los parámetros del modelo de celda solar de diodo único obtenidos con el método de Hussein, se usan para calcular el valor de corriente para cada valor de tensión de referencia conocido, y así trazar la curva calculada analíticamente. Además, se presentan las curvas de ARPE y SE (error cuadrático), los cuales se utilizan para luego calcular el MAPE y RMSE, respectivamente.

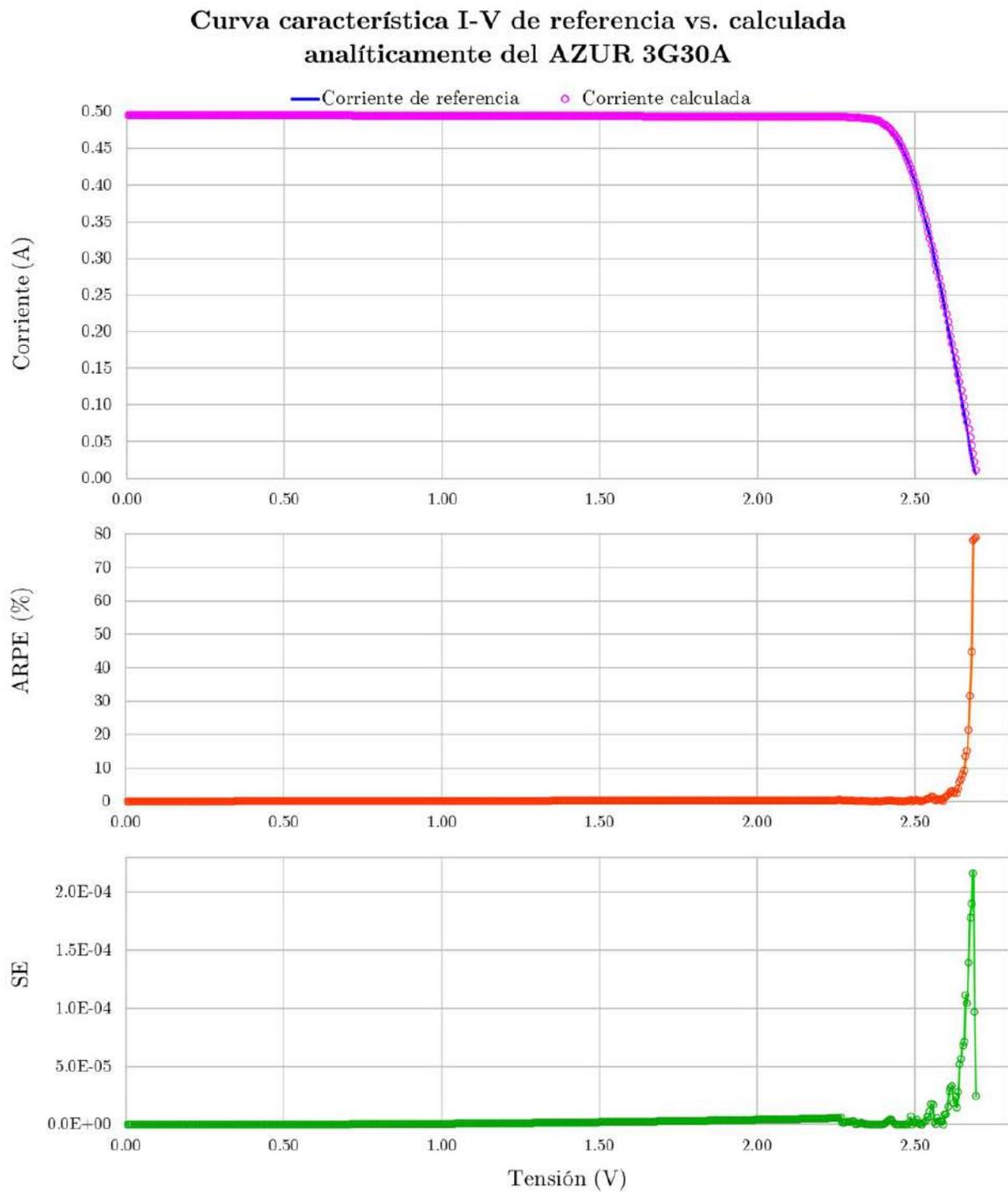


Figura 5.38: Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar AZUR 3G30A.

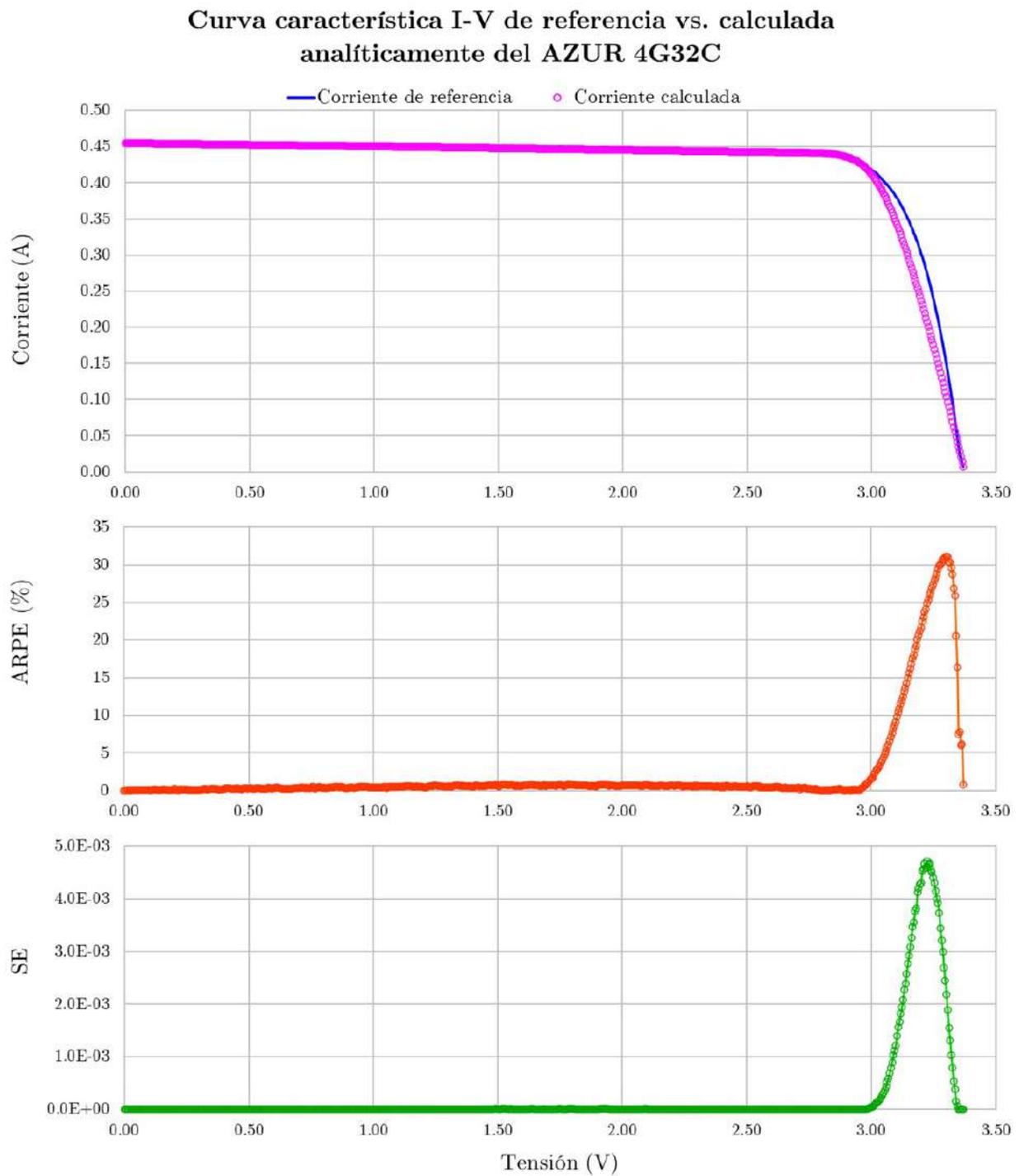


Figura 5.39: Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar AZUR 4G32C.

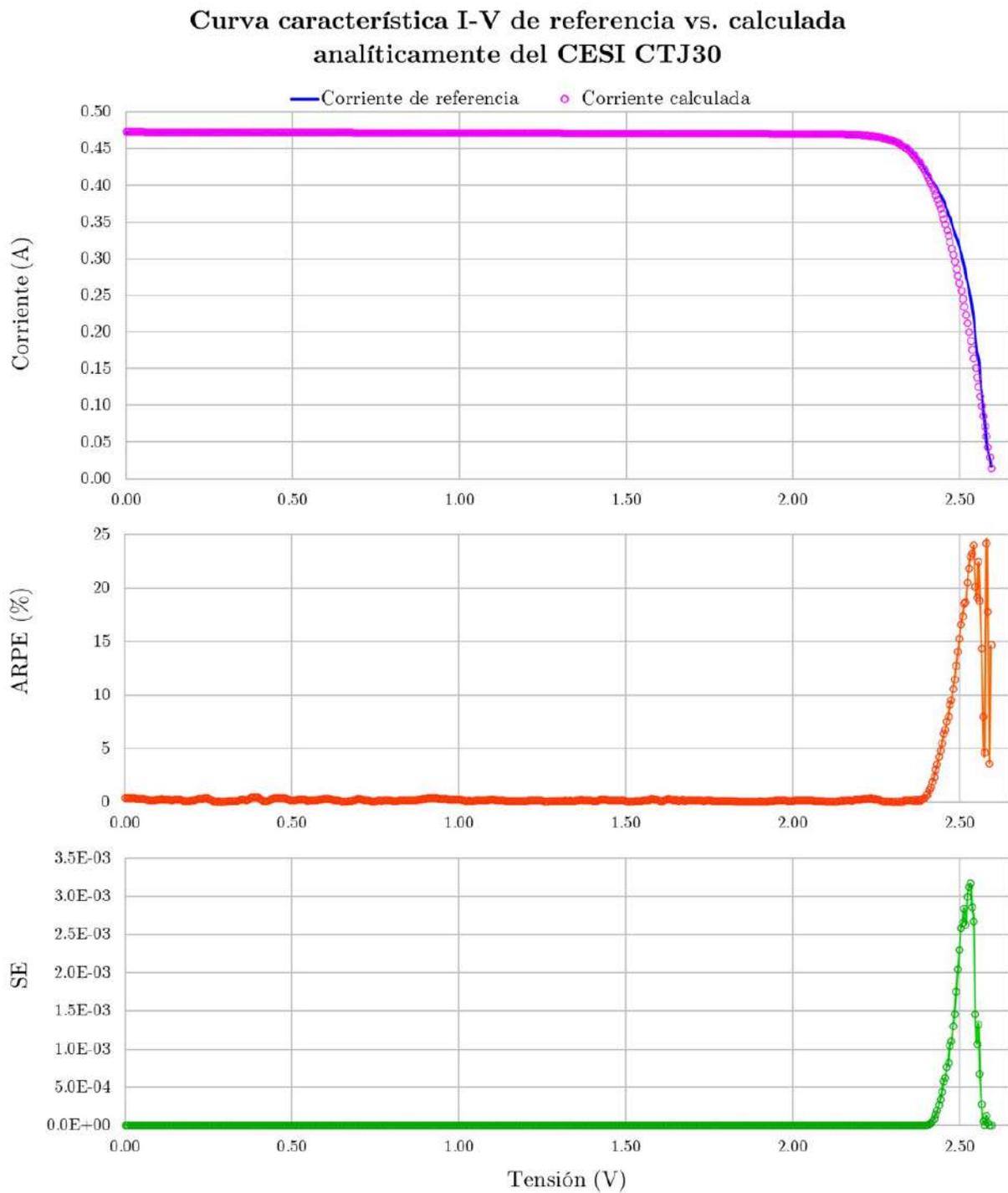


Figura 5.40: Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar CESI CTJ30.

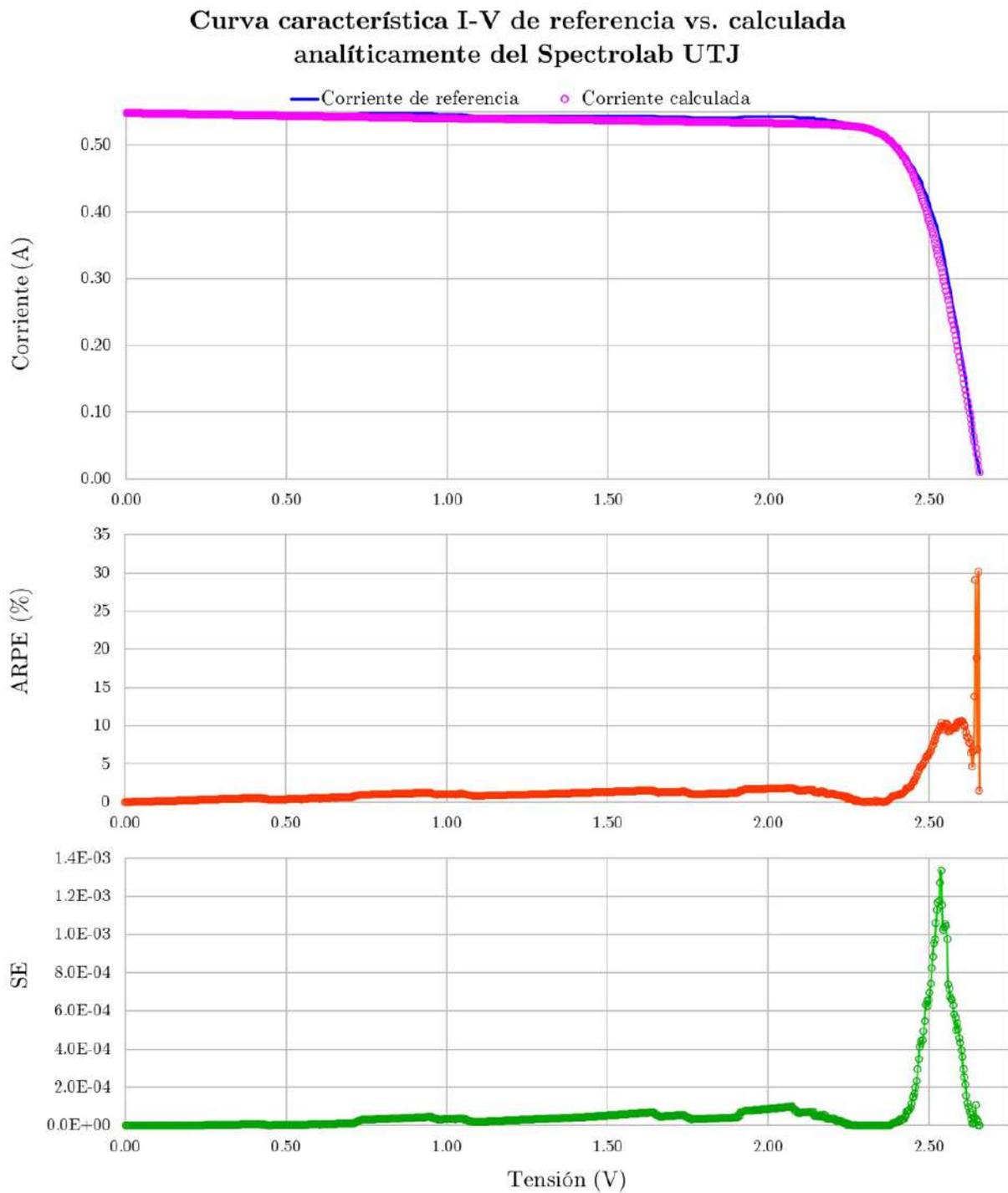


Figura 5.41: Curvas características I-V de referencia y calculada, y curvas de ARPE y SE para el panel solar Spectrolab UTJ.

En la curva característica del AZUR 3G30A, Figura 5.38, se puede apreciar que, visualmente, en el AZUR 3G30A la curva calculada de corriente se ajusta de manera muy precisa a lo largo de todo el rango de tensión, lo cual se ve reflejado en su casi perfecto R^2 de 0.9993. No obstante, cabe destacar que, conforme la corriente se aproxima más a cero, los indicadores de rendimiento aumentan exponencialmente. En total, la curva cuenta con 571 puntos de operación. Si se toman en cuenta los primeros 549 puntos (96.15 % de la curva), se obtiene un MAPE de 0.24 % y un RMSE de 1.34E-03, mientras que en los últimos 22 puntos, el MAPE se eleva a 19 % y el RMSE a 8.3198E-03. Se puede concluir entonces que, tal y como se discutió en el Capítulo 2, el MAPE castiga más fuertemente las diferencias absolutas pequeñas cuando los valores son cercanos a cero, que el RMSE.

En la curva característica del AZUR 4G32C, Figura 5.39, el ajuste de la curva calculada con la de referencia después del MPP es bajo, lo que se ve reflejado en un R^2 de 0.9478, considerablemente más bajo que en todos los otros paneles solares. Sin embargo, en los últimos puntos de la curva el ajuste es mucho mejor, obteniendo valores de ARPE y SE más bajos que en el AZUR 3G30A.

En la curva característica del CESI CTJ30, Figura 5.40, se observa un comportamiento similar al del AZUR 4G32C, donde después del MPP el ajuste de la curva calculada baja, aunque en este caso de manera menos pronunciada, y conforme la corriente se aproxima a cero, el ajuste vuelve a mejorar, presentando un R^2 de 0.9775. En este caso se pueden observar oscilaciones muy pronunciadas en el ARPE de los últimos puntos, evidenciando nuevamente uno de los puntos débiles del MAPE como indicador de rendimiento.

Finalmente, en la curva característica del Spectrolab UTJ se presenta un ajuste muy bueno incluso después del MPP, y similar a como ocurrió en el AZUR 3G30A, los valores de ARPE y SE crecen exponencialmente en los últimos puntos. En este caso cabe destacar que antes del MPP, el ajuste de la curva se ve perjudicado debido a las irregularidades presentes en la curva utilizada como referencia, donde idealmente el comportamiento debería ser casi lineal. Si se hubiera realizado un proceso de edición de imagen similar al realizado en la curva del CESI CTJ30, se pudo haber obtenido un set de datos de referencia con un comportamiento más regular previo al MPP y esto habría beneficiado el MAPE y RMSE de la curva calculada analíticamente para este panel.

Escalado de parámetros a otras condiciones externas

Debido a que solo se cuentan con imágenes de las curvas características en condiciones estándar de irradiancia, temperatura, y ángulo de incidencia, no es posible obtener indicadores de rendimiento para las curvas características obtenidas bajo distintas condiciones al compararlas con valores de referencia.

No obstante, se puede verificar que el comportamiento que se observa en diferentes puntos de las curvas obtenidas, conforme se aumentan o reducen los niveles de irradiancia, temperatura o ángulo de incidencia, sea congruente con lo esperado según la teoría de celdas fotovoltaica.

Cuando se aumentan los niveles de temperatura de celda, se presentan las siguientes condiciones:

- La corriente aumenta.
- El rango de tensión disminuye.
- Se reduce la producción energética.

Por otra parte, cuando se aumenta la irradiancia, se dan las siguientes condiciones:

- La corriente aumenta con una resistencia constante.
- La tensión V_{oc} aumenta levemente.
- La tensión V_{mp} se reduce levemente.
- La potencia P_{mp} aumenta.

En la Figura 5.42, se presentan de manera gráfica los efectos descritos anteriormente.

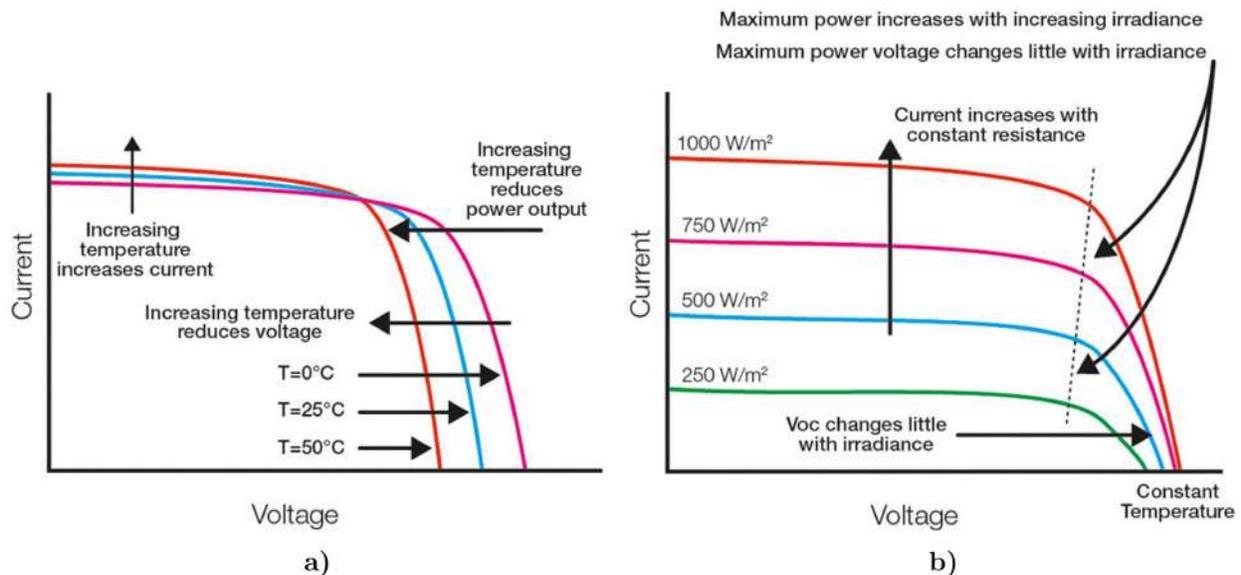


Figura 5.42: Efectos del aumento de temperatura (a) e irradiancia (b) en la curva I-V de una celda solar [20].

En cuanto al ángulo de incidencia, cabe recordar que este se refiere al ángulo en que los rayos solares impactan sobre la superficie de la celda fotovoltaica. El caso ideal es que los rayos impacten perpendicularmente a la superficie de la celda. No obstante, el aumentar el ángulo de incidencia en rango de 0° a 90° , lo que en esencia se hace es reducir la irradiancia que percibe el panel, y por ende los efectos percibidos en la curva característica son opuestos a los descritos anteriormente para el aumento de irradiancia. Además, como se observa en el conjunto de ecuaciones 5.6, el valor de irradiancia actual se multiplica por la función coseno del ángulo de incidencia, por lo tanto, se sigue la tendencia de la Figura 5.43.

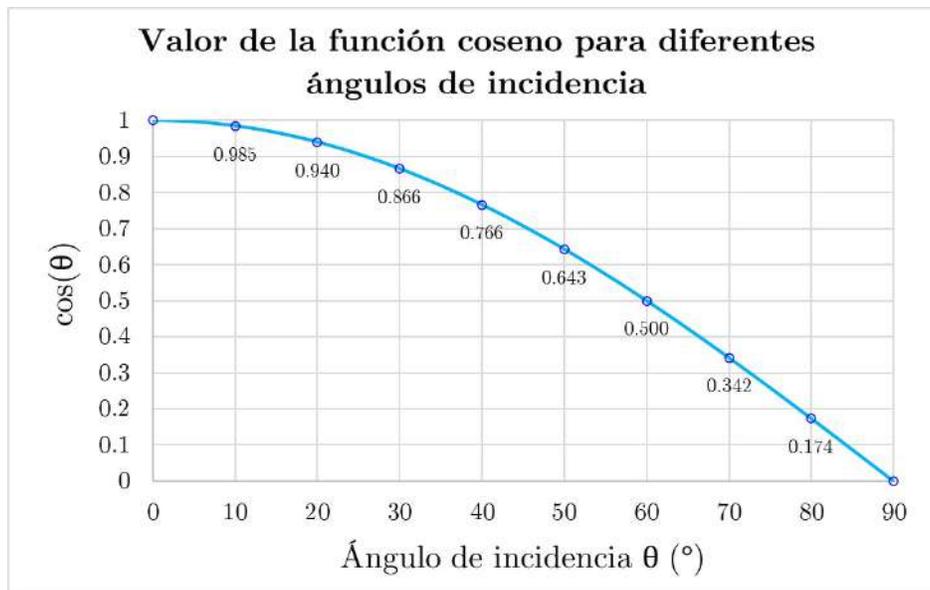


Figura 5.43: Valores de la función coseno para diferentes ángulos de incidencia.

A continuación, se presentan las curvas características del AZUR 3G30A obtenidas luego de escalar los parámetros del modelo de diodo único a cinco puntos diferentes por cada condición externa estudiada, según se indicó anteriormente en la Tabla 5.5.

En la Figura 5.44 se puede apreciar, en la curva I-V, que los incrementos constantes de temperatura producen un aumento de corriente y una disminución de tensión a lo largo de todo el rango. Además, en la curva P-V, se evidencia cómo la producción energética del panel solar se ve disminuida.

En la Figura 5.45 se puede observar cómo tanto las tensiones v_{oc} como las v_{mp} se ven levemente desplazadas a la izquierda conforme disminuye la irradiancia, mientras que se notan reducciones notorias de corriente y producción energética.

Finalmente, en la Figura 5.46 se pueden apreciar cómo los aumentos de ángulo de incidencia producen el efecto contrario que los aumentos de irradiancia. Asimismo, se puede apreciar el efecto de la función coseno en la forma en que varían las curvas con incrementos constantes de ángulo.

Todas las condiciones descritas anteriormente son consistentes en los demás paneles solares, por lo tanto, se considera el escalado de los parámetros a diferentes condiciones externas como exitoso.

En el Anexo G, se presentan las hojas de datos confeccionadas para cada uno de los paneles solares estudiados, donde se presentan sus respectivas tablas resúmenes, y curvas características. Además, debajo de cada conjunto de curvas se facilita un enlace para descargar los sets de datos correspondientes.

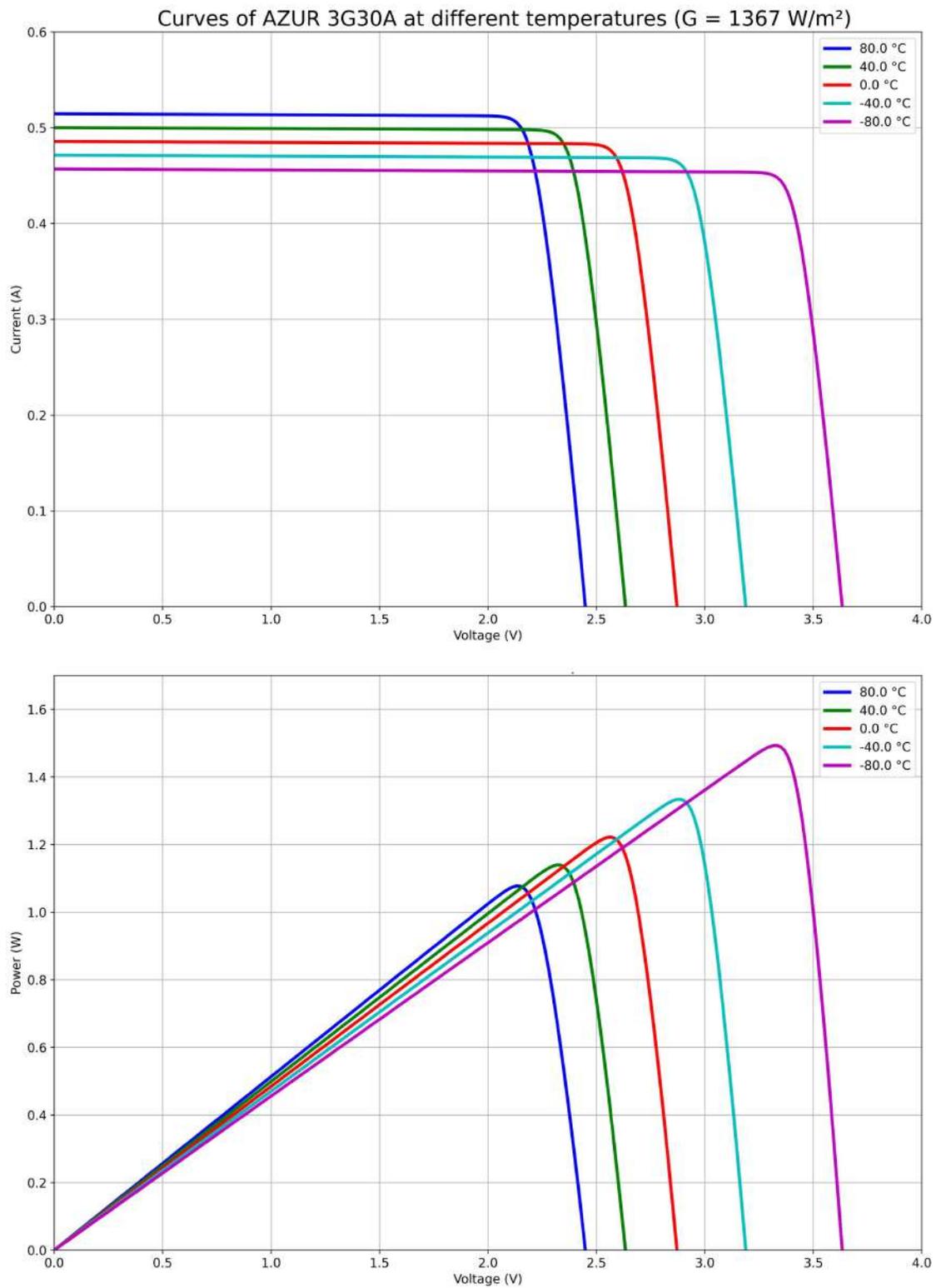


Figura 5.44: Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de temperatura de celda.

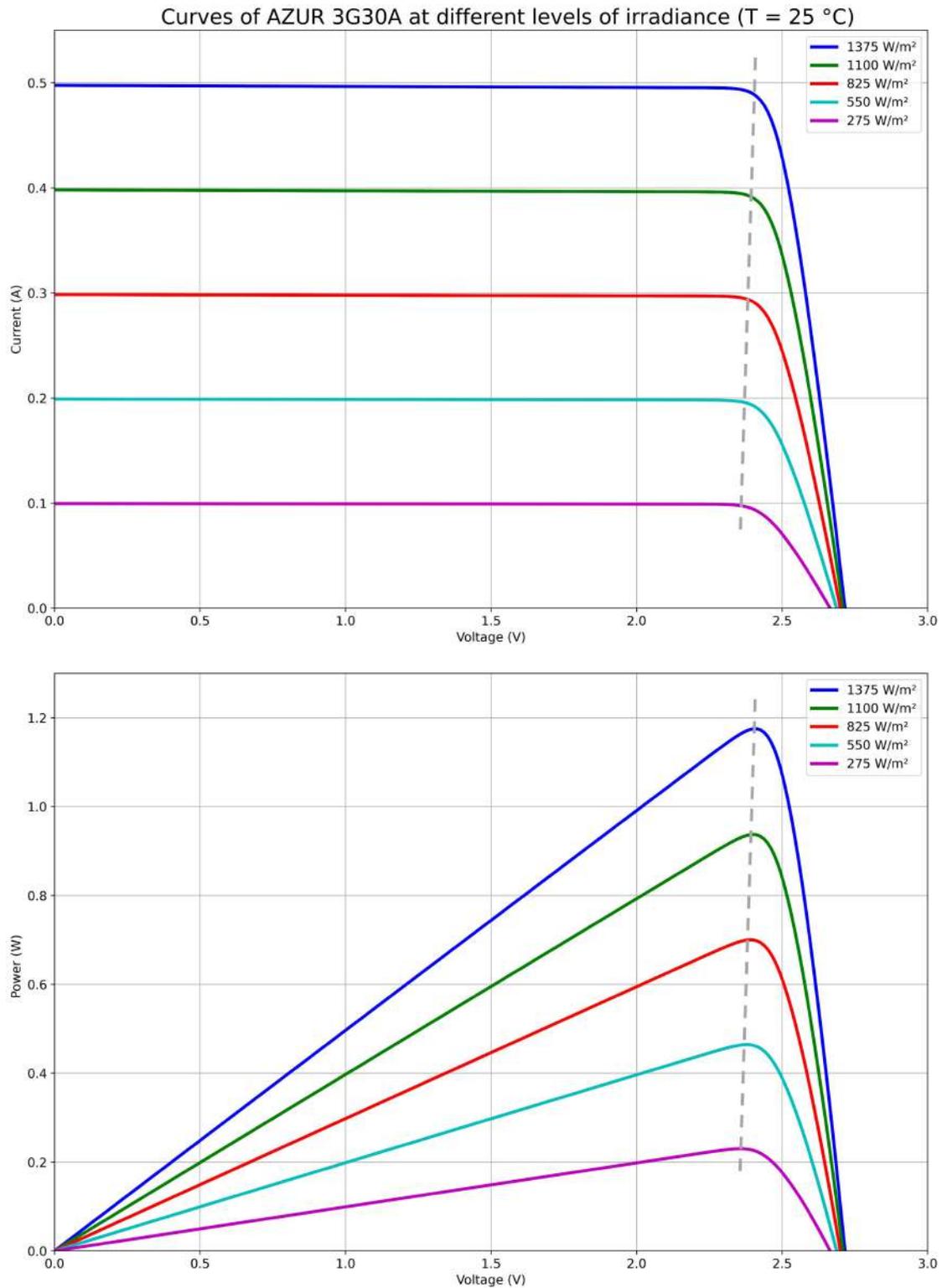


Figura 5.45: Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de irradiancia.

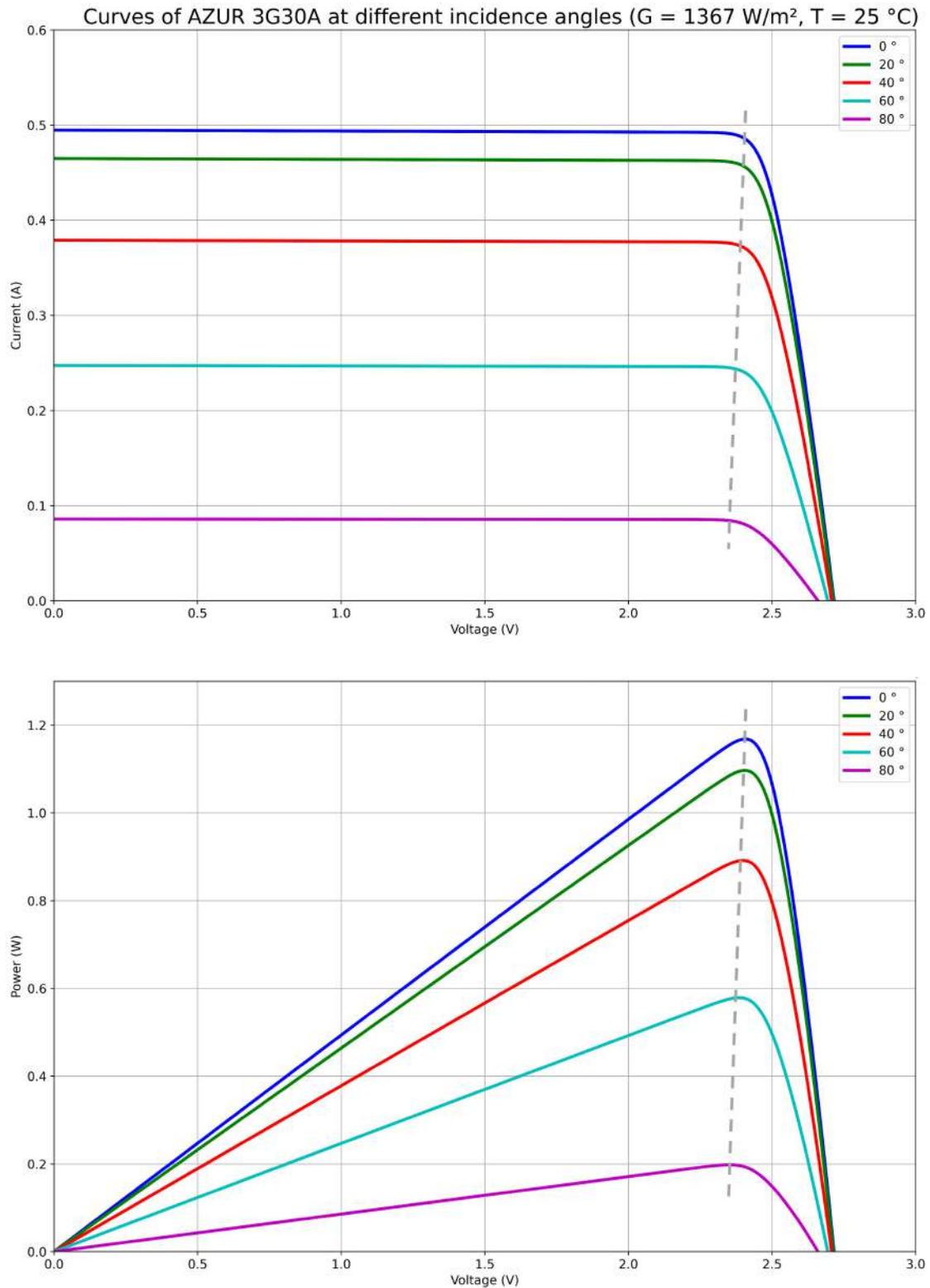


Figura 5.46: Curvas I-V y P-V del AZUR 3G30A a diferentes niveles de ángulo de incidencia.

5.3.2. Modelo analítico: Componente #2

En la Tabla 5.9, se resumen los tiempos de ejecución de la función *Analytic_Model* en cada plataforma de SAS, de acuerdo al diagrama de bloques de la Figura 5.13. Para esto, se desarrolló un programa secundario que utiliza las mismas funciones del programa implementado en el MCU_M, para calcular más de 1000 puntos de operación en donde se cambian aleatoriamente los valores de tensión y de las condiciones externas en un amplio rango. En las especificaciones de la Tabla 3.6, para este tiempo de ejecución se tiene como valor ideal que sea menor a 50 ms, por lo tanto, ambas plataformas de SAS lo cumplen con un amplio margen. Cabe mencionar que, dado que la plataforma #1 utiliza funciones de la librería SciPy de Python, el método utilizado para despejar la ecuación del modelo de diodo único está bastante más optimizado que la implementación manual en C/C++ del método de Newthon-Rhapson que se utilizó en la plataforma #2.

Tabla 5.9: Tiempos de ejecución de la función *Analytic_Model*.

Parámetro	Plataforma 1	Plataforma 2
Tiempo promedio (ms)	0.60090	2.09655
Desviación estándar (ms)	0.23329	0.43602
Tiempo máximo (ms)	1.52550	5.00000
Tiempo mínimo (ms)	0.25360	1.00000

Por otra parte, en la Tabla 5.10 se muestran los tamaños en KB de los programas implementados en el MCU_M y MCU_S2, así como el porcentaje de la memoria disponible que utilizan. También se coloca por aparte la memoria que abarca únicamente la implementación de *Analytic_Model*, ya que entre las especificaciones se estableció como valor ideal que utilizara menos de 30 KB, validando su cumplimiento.

Tabla 5.10: Tamaño de memoria de programas de la plataforma SAS #2

Parámetro	Tamaño (KB)	Porcentaje de memoria total (%)
Programa de MCU_M	84.668	43
Programa de MCU_S2	401.532	38
Función <i>Analytic_Model</i>	10.548	5

En el Capítulo 9, se presentan pruebas más extensas que permiten validar el correcto funcionamiento de ambas plataformas al ser integradas con la solución de IoT propuesta. En el Anexo C, se presentan guías detalladas de todo el software desarrollado y cómo configurar el hardware de ambas plataformas previo a realizar pruebas.

Capítulo 6

Solución de controladores digitales

En este capítulo se abarca el proceso de diseño de los controladores digitales para la tensión y corriente del convertidor de conmutación reductor síncrono de la planta MCP. Dado que no se cuenta con un prototipo funcional de la planta MCP, la validación del diseño se realiza a través de simulaciones.

En la Figura 6.1, se presenta la secuencia de pasos seguida.

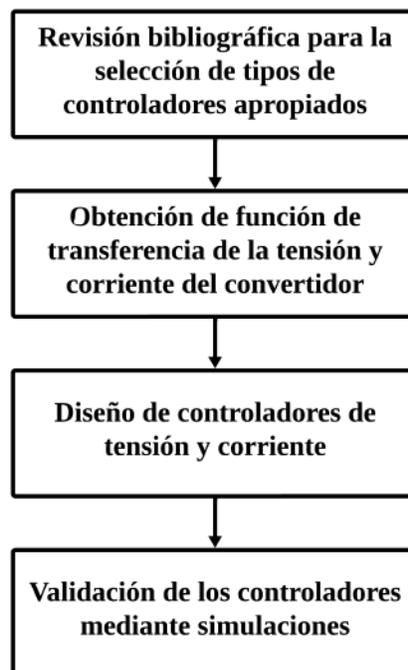


Figura 6.1: Secuencia de pasos seguida para la obtención de los controladores digitales.

6.1. Selección de conceptos

Se realizó una revisión bibliográfica sobre controladores digitales de convertidores de potencia de conmutación, donde se recomienda el uso controladores PID y PI para regular la salida de tensión y corriente, respectivamente, de convertidores reductores [6], como el presente en la planta MCP. Adicionalmente, se ofrece una guía teórica y práctica muy detallada que abarca todo el proceso de diseño e implementación de estos. Asimismo, en trabajos previos dentro del marco del proyecto IPTC se ha explorado el uso de este tipo de controladores para la carga y descarga de celdas electroquímicas [69], y para los diferentes modos de operación de una carga electrónica [70], obteniendo resultados prometedores. Por lo tanto, se selecciona un controlador PID para la tensión y un PI para la corriente.

6.2. Desarrollo de conceptos

6.2.1. Función de transferencia

En la Tabla 6.1 se definen ciertos valores de referencia para la obtención de las funciones de transferencia de la corriente y tensión del convertidor reductor. La frecuencia de conmutación fue definida en un trabajo de graduación previo dentro del marco del IPTC, validando que es un valor óptimo en una versión previa de la planta MCP [69]. Por otra parte, la tensión de entrada viene definida por el convertidor AC-DC seleccionado en el Capítulo 4, mientras que para la tensión y corriente de referencia, se toma el 90 % del los valores máximos que puede ofrecer dicho convertidor, con el fin de no sobrecargarlo.

Tabla 6.1: Parámetros de operación de la planta MCP.

Parámetro	Magnitud
Frecuencia de conmutación f_s	125 kHz
Tensión de entrada V_g	24 V
Tensión de referencia V_{ref}	21.6V
Corriente de referencia I_{ref}	3.6 A

Por otra parte, de acuerdo a la selección de componentes reactivos realizada en el mismo trabajo de graduación mencionado anteriormente, se tienen las características del capacitor e inductor de la planta MCP, listadas en la Tabla 6.2.

Tabla 6.2: Parámetros de los componentes reactivos de la planta MCP.

Parámetro	Capacitor	Inductor
Inductancia H	-	470 μ H
Capacitancia C	470 μ F	-
Resistencia serie equivalente ESR	20 m Ω	91.8 m Ω

Por otra parte, es necesario definir algunos parámetros relacionados con los tiempos de muestreo y de ejecución. En la Figura 6.2, se muestra el diagrama de tiempos para un controlador digital, donde k representa el instante de muestreo, $v_s(t)$ la tensión de salida del convertidor, t_{DPWM} el tiempo que toma obtener el comando de control $u[k]$ basado en el ciclo de muestreo y generar un tren de pulsos con ciclo de trabajo $d[k]$ proporcional a $u[k]$. El algoritmo encargado de realizar el control es programado y ejecutado por el MCU de la planta, por lo que t_{ctrl} depende del tiempo que toma realizar la conversión de la señal analógica a digital y del tiempo que le tome al MCU ejecutar el algoritmo, que se puede comparar con el periodo de conmutación. Asimismo, en la figura se puede apreciar que el periodo de muestreo y el de conmutación son iguales, cumpliendo así el teorema de Nyquist que establece que la frecuencia de muestreo debe ser mayor o igual al doble del ancho de banda ($f_s \geq 2B$).

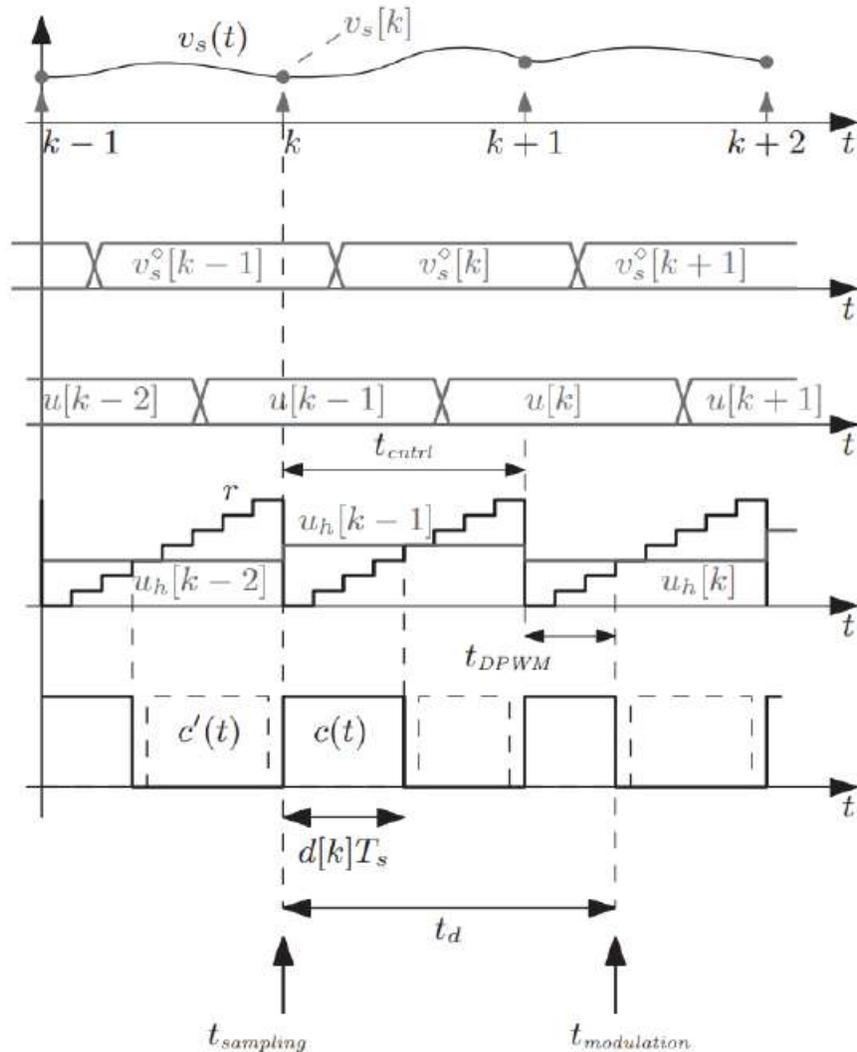


Figura 6.2: Diagrama de tiempos para un controlador digital [6]

Tomando el valor de f_s indicado en la Tabla 6.1, se obtienen los parámetros listados en la Tabla 6.3. Debido a lo indicado anteriormente, se tiene que t_{ctrl} es igual al periodo de conmutación T_s . Por otra parte, se tiene que t_{DPWM} es igual al producto del periodo de conmutación T_s y el ciclo de trabajo D , definido en 0.5 según se recomienda en la literatura [6]. Finalmente, el tiempo de retardo t_d es igual a la suma entre t_{DPWM} y t_{ctrl} .

Tabla 6.3: Parámetros de tiempos para el controlador digital.

Parámetro	Magnitud
Tiempo de ejecución del algoritmo de control t_{ctrl}	8 μ s
Periodo de conmutación T_s	8 μ s
Tiempo de retardo t_{DPWM}	4 μ s
Tiempo de retardo t_d	12 μ s

Ahora, se procede a utilizar el modelado de pequeña señal en tiempo discreto, descrito en detalle en el Capítulo 2, para obtener las funciones de transferencia de la tensión y corriente, tomando como base los parámetros establecidos anteriormente en las Tablas 6.1 y 6.3.

Las matrices Φ y γ , indicadas en las ecuaciones 2.25 y 2.26, respectivamente, son resueltas utilizando MATLAB. Por otra parte, tal y como se indica en la ecuación 2.20, la matriz δ corresponde a la matriz de salida C_c . De esta forma, se obtiene el conjunto de matrices 6.1.

$$\begin{aligned}\Phi &= \begin{bmatrix} 0,99795 & -0,01700 \\ 0,01700 & 0,99985 \end{bmatrix} \\ \gamma &= \begin{bmatrix} 0,40888 \\ -0,00347 \end{bmatrix} \\ \delta &= \begin{bmatrix} 1 & 0 \\ 0,02 & 1 \end{bmatrix}\end{aligned}\tag{6.1}$$

Utilizando las funciones incorporadas en MATLAB para obtener el espacio de estados [114] y la función de transferencia [115], $ss()$ y $tf()$, respectivamente, se toma la matriz \mathbf{A} como Φ y la matriz \mathbf{B} como δ para obtener la función de transferencia de la tensión, en la ecuación 6.2, y la corriente, en la ecuación 6.3.

$$G_v(z) = \frac{0,004699z + 0,002249}{z^2 - 1,998z + 0,9981}\tag{6.2}$$

$$G_i(z) = \frac{0,4089z - 0,4088}{z^2 - 1,998z + 0,9981}\tag{6.3}$$

Asimismo, utilizando la función $bode()$ [116] se obtienen los diagramas de Bode para cada función de transferencia, los cuales permiten apreciar cómo cambia la magnitud de la señal de salida del sistema con respecto a la frecuencia de la señal de entrada, así como el cambio de fase introducido por el sistema en varias frecuencias. En la Figura 6.3 se presenta el de la tensión y en la Figura 6.4 el de la corriente.

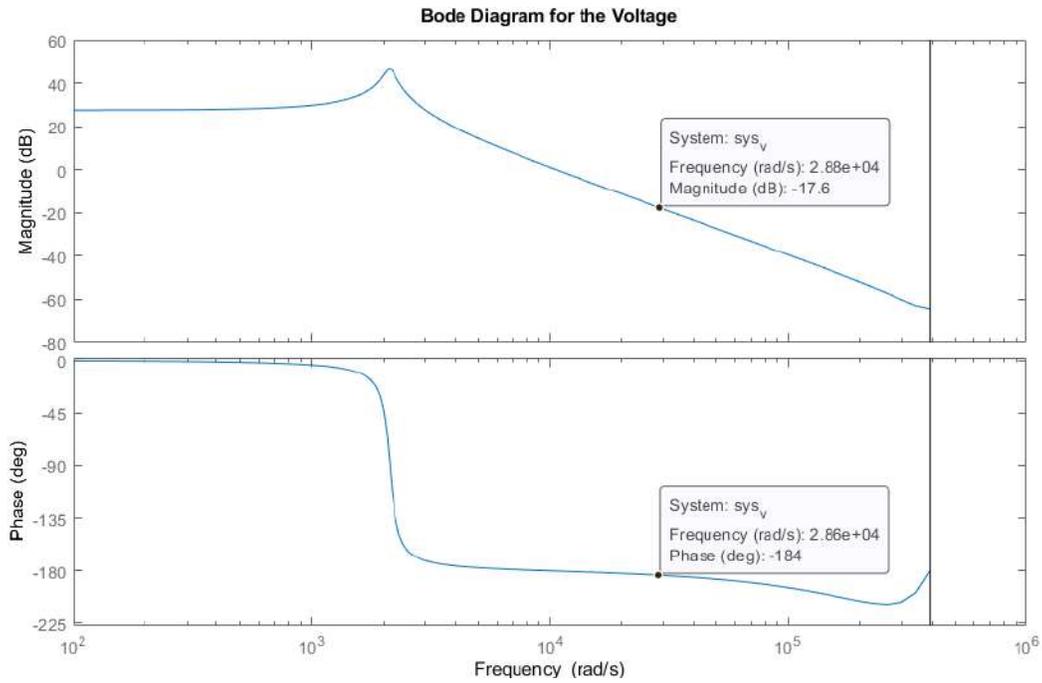


Figura 6.3: Diagrama de Bode para la función de transferencia en tiempo discreto de la tensión.

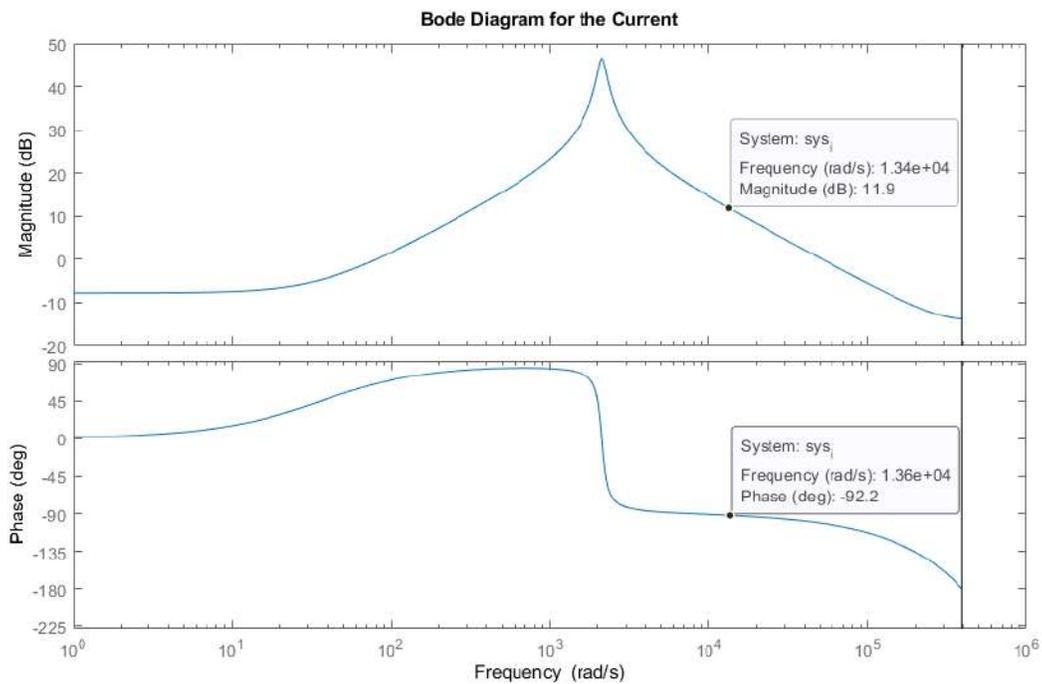


Figura 6.4: Diagrama de Bode para la función de transferencia en tiempo discreto de la corriente.

6.2.2. Controladores digitales

Se procede a diseñar los controladores de tensión y corriente que se derivan a partir de las funciones de transferencia obtenidas anteriormente. Como se mencionó en el Capítulo 2, el convertidor es controlado mediante una señal PWM que afecta la conmutación del interruptor. En la Figura 6.5, se presenta un diagrama en que se ilustra la forma en que la señal PWM es gestionada, así como la forma en que se implementa un controlador digital para la topología de convertidor de la planta.

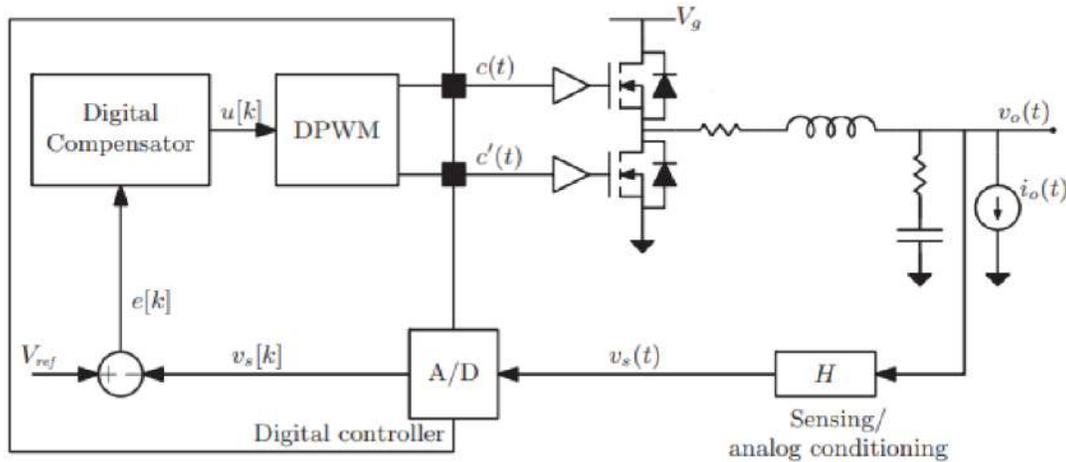


Figura 6.5: Controlador digital para un convertidor reductor [6].

Se puede observar que al conectar ambas salidas PWM a las compuertas de los transistores, dado que una está negada y la otra no, se asegura que si uno permite el paso de corriente, el otro lo bloquea y viceversa. También cabe destacar que se requiere de un ADC para digitalizar las lecturas y garantizar el correcto funcionamiento del controlador.

Controlador de tensión

En la Figura 6.6, se presenta el diagrama de bloques del controlador PID digital a implementar para regular la tensión, mientras que en el conjunto de ecuaciones 6.4 se presentan las expresiones en tiempo discreto que lo describen, siguiendo la estructura de Euler.

$$\begin{aligned}
 u_p[k] &= K_p e[k] \\
 u_i[k] &= u_i[k-1] + K_i e[k] \\
 u_d[k] &= K_d (e[k] - e[k-1]) \\
 u[k] &= u_p[k] + u_i[k] + u_d[k]
 \end{aligned} \tag{6.4}$$

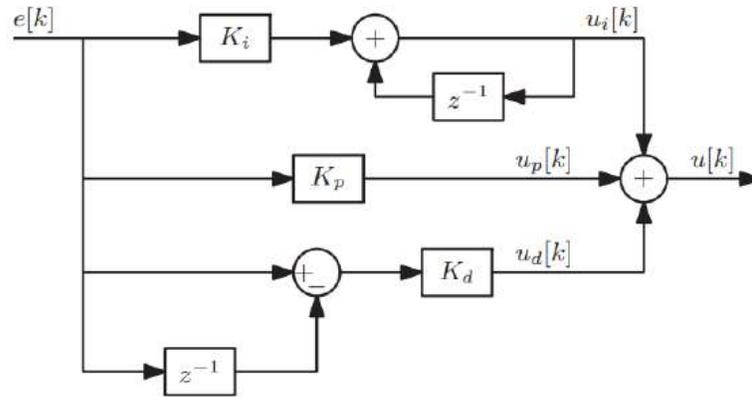


Figura 6.6: Diagrama de bloques del controlador PID [6].

Por otra parte, la función de transferencia de este compensador en el dominio de z se presenta en la ecuación 6.5

$$G_{PID}(z) = K_p + \frac{K_i}{1 - z^{-1}} + K_d \cdot (1 - z^{-1}) \quad (6.5)$$

Ahora, se procede a calcular las constantes del controlador PID. Primeramente, luego de consultar la literatura [6], se definen la frecuencia de cruce y el margen de fase como $f_c = \frac{f_s}{14} = 8928,57$ Hz y $\phi_m = 69^\circ$, respectivamente. En el caso del margen de fase, se indica que entre mayor sea, menor es el sobreimpulso durante la respuesta transitoria.

Al seleccionar dicho valor de f_c , se puede extraer del diagrama de Bode para la tensión, Figura 6.3, las especificaciones del conjunto de ecuaciones 6.6.

$$\begin{aligned} w_c &= 2\pi f_c = 8928,57 \text{ rad/s} \\ |T_{vu}(e^{jw_c T_s})| &= |T'_{vu}(jw'_c)| = -35,3 \text{ dB} \\ \angle T_{vu}(e^{jw_c T_s}) &= \angle T'_{vu}(jw'_c) = -188,015^\circ \\ w_p &= \frac{2}{T_s} = 250000 \text{ rad/s} \\ w'_c &= w_p \tan\left(\frac{w_c T_s}{2}\right) = 57061,00 \text{ rad/s} \end{aligned} \quad (6.6)$$

Seguidamente, se debe verificar que el margen de fase seleccionado cumple con las condiciones definidas en la literatura, como se indica en la inecuación 6.7.

$$\phi_{m,u} < \phi_m < \phi_{m,u} + \frac{\pi}{2} - \arctan\left(\frac{w'_c}{w_p}\right) \quad (6.7)$$

Los parámetros $\phi_{m,u}$ y $\arctan\left(\frac{w'_c}{w_p}\right)$ vienen definidos en el conjunto de ecuaciones 6.8.

$$\begin{aligned}\phi_{m,u} &= \pi + \angle T'_{vu}(jw'_c) = -188,0151^\circ + 180^\circ = -8,0151^\circ \\ \arctan\left(\frac{w'_c}{w_p}\right) &= 12,8566^\circ\end{aligned}\quad (6.8)$$

Por lo tanto, tal y como se muestra en la ecuación 6.9, el diseño de controlador PID es válido.

$$-8,0151^\circ < 69^\circ < 77,1434^\circ \quad (6.9)$$

Ahora, se procede a obtener las constantes del controlador a partir del conjunto de ecuaciones 6.10.

$$\begin{aligned}K_p &= G'_{PI\infty} G'_{PD0} \left(1 + \frac{w_{PI}}{w_{PD}} - \frac{2w_{PI}}{w_p}\right) \\ K_i &= 2G'_{PI\infty} G'_{PD0} \frac{w_{PI}}{w_p} \\ K_d &= \frac{G'_{PI\infty} G'_{PD0}}{2} \left(1 - \frac{w_{PI}}{w_p}\right) \left(\frac{w_p}{w_{PD}} - 1\right)\end{aligned}\quad (6.10)$$

En estas, se tiene que $G'_{PI\infty}$ corresponde a la ganancia integral del compensador y G'_{PD0} a la ganancia derivativa. Para un controlador de tensión PID, se tiene el siguiente conjunto de ecuaciones 6.11.

$$\begin{aligned}G'_{PI\infty} &= 1 \\ G'_{PD0} &= \frac{1}{|T'_{vu}(jw'_c)|} \frac{\sqrt{1 + \left(\frac{w'_c}{w_p}\right)^2}}{\sqrt{1 + \left(\frac{w'_c}{w_{PD}}\right)^2}} = 0,1330 \\ w_{PD} &= \frac{w'_c}{\tan\left(\phi_m - \phi_{m,u} + \arctan\left(\frac{w'_c}{w_p}\right)\right)} = 127,1998 \text{ Rad/s} \\ w_{PI} &= \frac{w_c}{20} = 2805 \text{ rad/s}\end{aligned}\quad (6.11)$$

Sustituyendo, se obtienen finalmente las constantes del conjunto de ecuaciones 6.12, cuya función de transferencia se muestra en la ecuación 6.13

$$K_p = 1,53100 \quad K_i = 0,00149 \quad K_d = 64,56482 \quad (6.12)$$

$$G_{c,v} = \frac{66,1z^2 - 130,7z + 64,56}{z^2 - z} \quad (6.13)$$

Controlador de corriente

En la Figura 6.7, se presenta el diagrama de bloques del controlador digital PI a implementar para regular la corriente, mientras que en el conjunto de ecuaciones 6.14 se presentan las expresiones en tiempo discreto que lo describen.

$$\begin{aligned} u_p[k] &= K_p e[k] \\ u_i[k] &= u_i[k-1] + K_i e[k] \\ u[k] &= u_p[k] + u_i[k] \end{aligned} \quad (6.14)$$

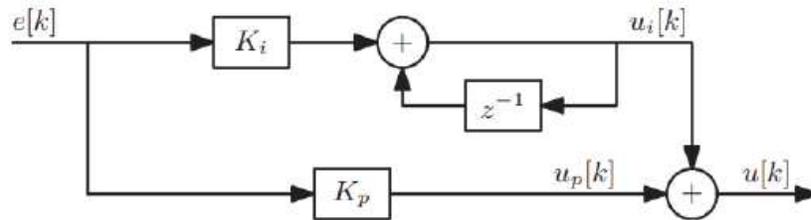


Figura 6.7: Diagrama de bloques del controlador PI [6].

Por otra parte, la función de transferencia de este compensador en el dominio de z se presenta en la ecuación 6.15.

$$G_{PI}(z) = K_p + \frac{K_i}{1 - z^{-1}} \quad (6.15)$$

Similar a como se realizó para el controlador de tensión, se requiere definir la frecuencia de cruce y el margen de fase. En este caso, se toma un $f_c = \frac{f_s}{18} = 6944,44$ Hz y $\phi_m = 70^\circ$. Nuevamente, a partir del valor de f_c , se puede extraer del diagrama de Bode de la corriente, Figura 6.4, las especificaciones del conjunto de ecuaciones 6.16.

$$\begin{aligned} w_c &= 43633,23 \text{ rad/s} \\ |T_{iu}(e^{jw_c T_s})| &= |T'_{iu}(jw'_c)| = 1,1811 \text{ dB} \\ \angle T_{vu}(e^{jw_c T_s}) &= \angle T'_{vu}(jw'_c) = -99,7371^\circ \\ w_p &= 250000 \text{ rad/s} \\ w'_c &= 44081,74 \text{ rad/s} \end{aligned} \quad (6.16)$$

Ahora, se debe verificar que el margen de fase seleccionado cumple con las condiciones indicadas en la literatura, según la inecuación 6.17.

$$\phi_{m,u} - \arctan\left(\frac{\omega'_c}{\omega_p}\right) < \phi_m < \phi_{m,u} \quad (6.17)$$

Los parámetros $\phi_{m,u}$ y $\arctan\left(\frac{\omega'_c}{\omega_p}\right)$ vienen definidos en el conjunto de ecuaciones 6.18.

$$\begin{aligned}\phi_{m,u} &= \pi + \angle T'_{iu}(j\omega'_c) = -99,7371^\circ + 180^\circ = 80,2629^\circ \\ \arctan\left(\frac{\omega'_c}{\omega_p}\right) &= 29,5532^\circ\end{aligned}\quad (6.18)$$

Por lo tanto, tal y como se muestra en la ecuación 6.19, el diseño de controlador PI es válido.

$$50,7097^\circ < 70^\circ < 80,2629^\circ \quad (6.19)$$

Se procede entonces a obtener las constantes a partir del conjunto de ecuaciones 6.20.

$$\begin{aligned}K_p &= G'_{PI\infty} \left(1 - \frac{\omega_{PI}}{\omega_p}\right) \\ K_i &= 2G'_{PI\infty} \frac{\omega_{PI}}{\omega_p} \\ \omega_{PI} &= \omega'_c \tan(\phi_{m,u} - \phi_m) \\ G'_{PI\infty} &= \frac{1}{|T'_i(j\omega'_c)|} \frac{1}{\sqrt{1 + \left(\frac{\omega_{PI}}{\omega'_c}\right)^2}} \\ \omega_{PI} &= 11863,6836 \text{ rad/s} \\ G'_{PI\infty} &= 0,8175\end{aligned}\quad (6.20)$$

A partir de esto, se obtienen las constantes del conjunto de ecuaciones 6.21, cuya función de transferencia se presenta en la ecuación 6.22

$$K_p = 1,85637 \quad K_i = 0,07759 \quad (6.21)$$

$$G_{ci}(z) = \frac{1,934z - 1,856}{z - 1} \quad (6.22)$$

6.3. Validación de conceptos

Para validar el rendimiento de los controladores, se implementaron en Simulink los diagramas de bloques indicados anteriormente para los controladores PID y PI en las Figuras 6.6 y 6.7, respectivamente, y se les colocó una entrada de escalón unitario, con el objetivo de visualizar la respuesta de su salida en el tiempo. Se configuró un tiempo de muestreo de $8 \mu\text{s}$.

En las Figuras 6.8 y 6.9 se presentan los diagramas de bloques implementados en Simulink para la tensión y la corriente, respectivamente.

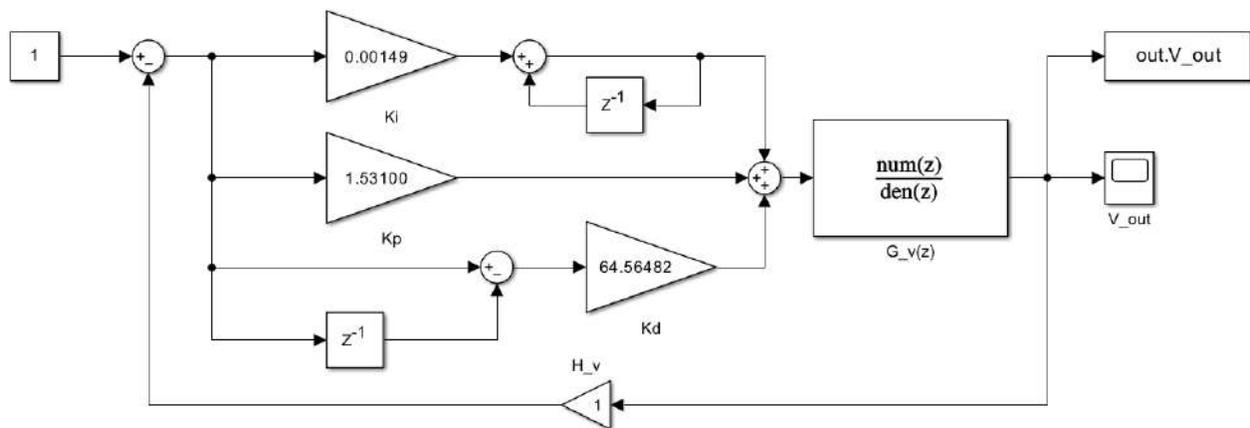


Figura 6.8: Diagrama de bloques del controlador PID implementado en Simulink.

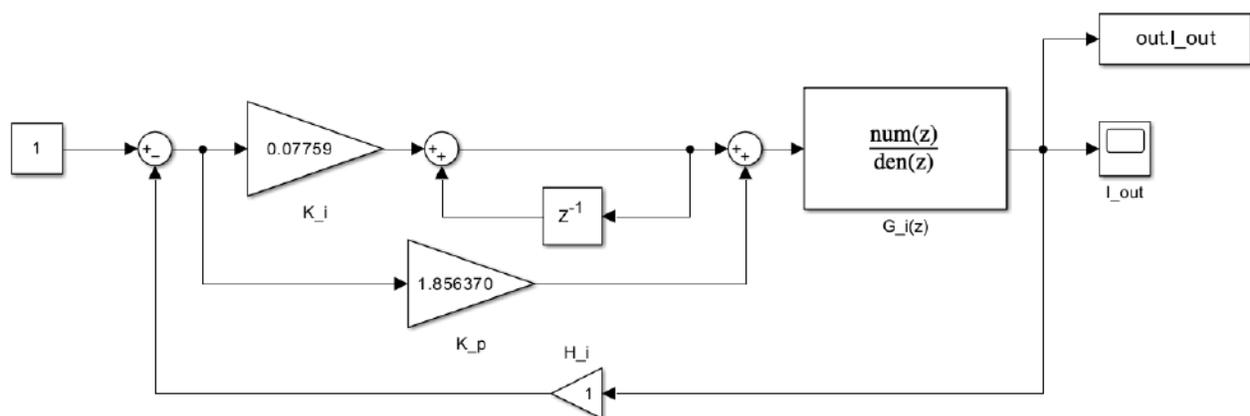


Figura 6.9: Diagrama de bloques del controlador PI implementado en Simulink.

En el controlador PID se obtiene la respuesta en la tensión de la Figura 6.10, donde se aprecia que se tiene un porcentaje de sobreimpulso $\%SI$ de 4.484% y un tiempo de estabilización de $t_{set2\%}$ de 264 μ s. Asimismo, en la Figura 6.11, se puede apreciar que pasados los 78.824 ms, se tiene un error en estado estacionario $e_{ss} = 0\%$.

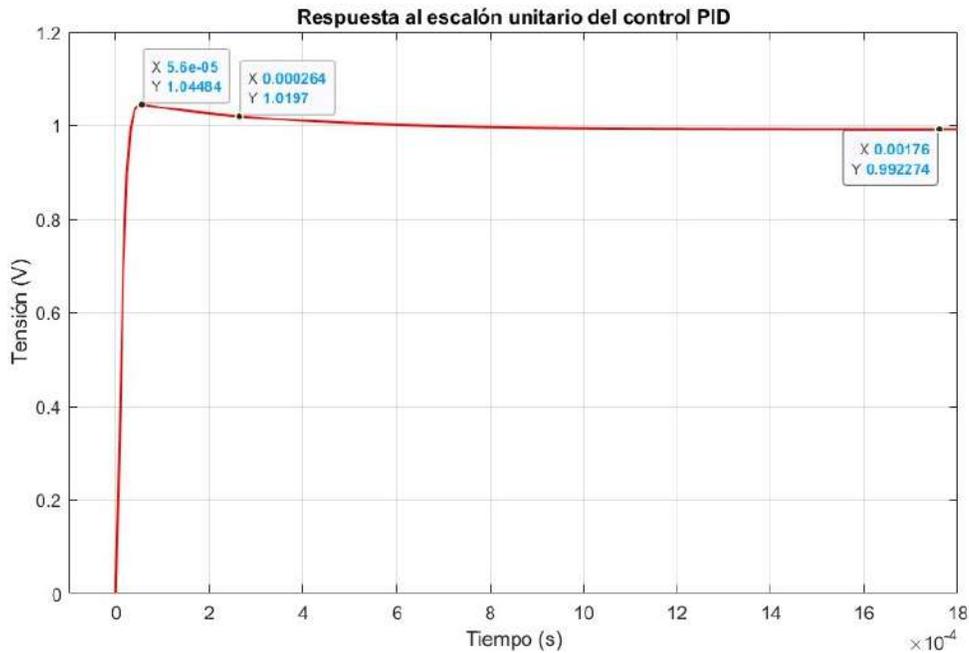


Figura 6.10: Respuesta escalón unitario en controlador PID. Detalle de $\%SI$ y $t_{set2\%}$.

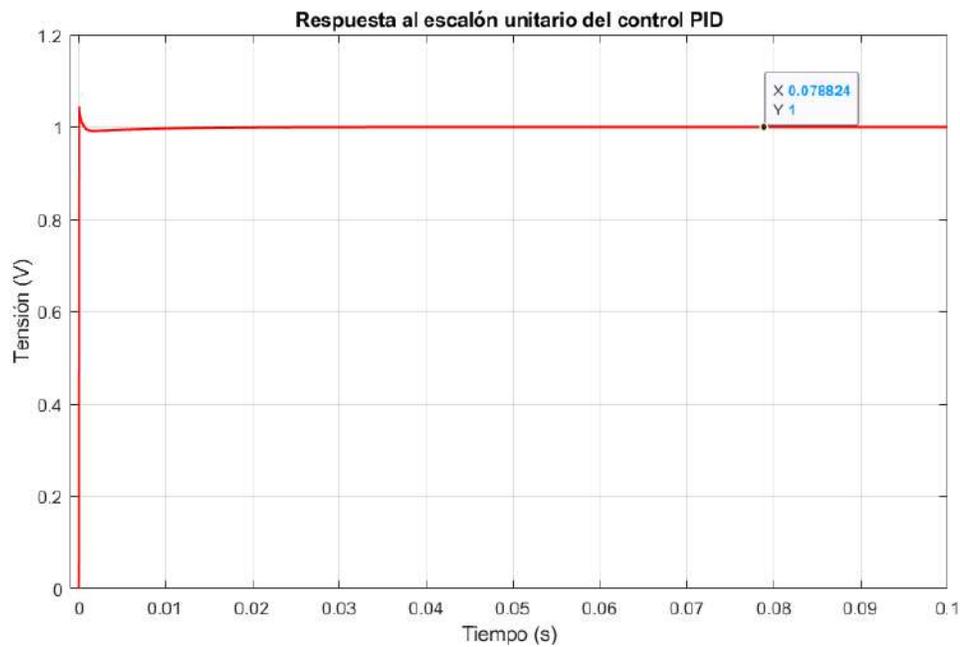


Figura 6.11: Respuesta escalón unitario en controlador PID. Detalle de e_{ss} .

En el caso del controlador PI, se obtiene la respuesta en la tensión de la Figura 6.12, donde se puede apreciar que se tiene un $\%SI$ de 4.478% y un $t_{set2\%}$ de 184 μ s. Asimismo, en la Figura 6.13, se puede apreciar que pasados los 287.376 ms, se tiene un error en estado estacionario $e_{ss} = 0\%$.

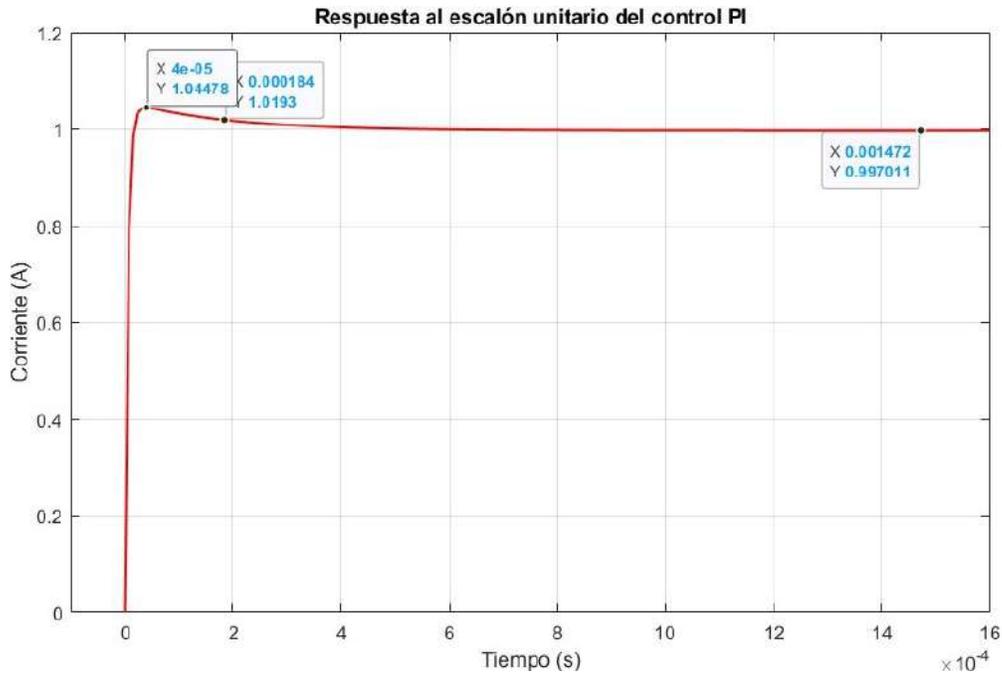


Figura 6.12: Respuesta escalón unitario en controlador PI. Detalle de $\%SI$ y $t_{set2\%}$.

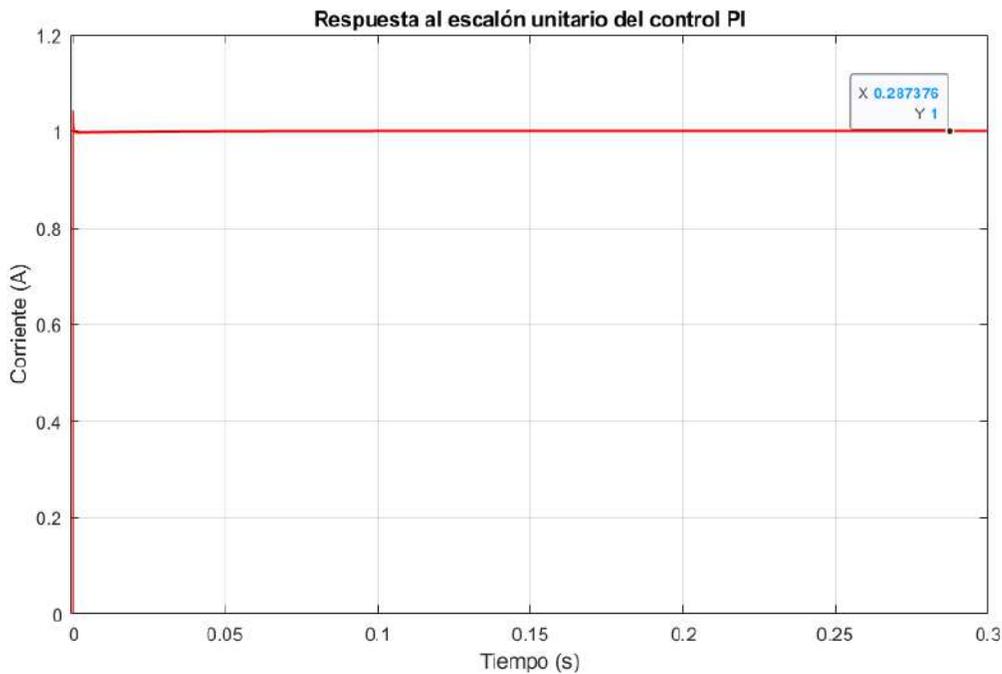


Figura 6.13: Respuesta escalón unitario en controlador PI. Detalle de e_{ss} .

En la Tabla 3.6, se definieron como valores ideales un $\%SI$ menor al 5%, un $t_{set2\%}$ menor a 50 ms, y un e_{ss} menor a 1%. Por lo tanto, las simulaciones permiten validar que ambos controladores digitales cumplen satisfactoriamente con todas las métricas de rendimiento definidas.

En el enlace de Google Drive facilitado en el Anexo B, se puede acceder a una carpeta que contiene todos los programas y simulaciones de MATLAB/Simulink que fueron utilizados en este capítulo.

Capítulo 7

Solución de IoT

En este capítulo se abarcan las fases de la metodología relacionadas con la generación, selección y desarrollo de conceptos para la solución de IoT. Dado que se requiere integrar la solución IoT con la solución de SAS, la validación de los conceptos se realiza en el Capítulo 9.

A continuación, se realiza la descomposición funcional de la solución:

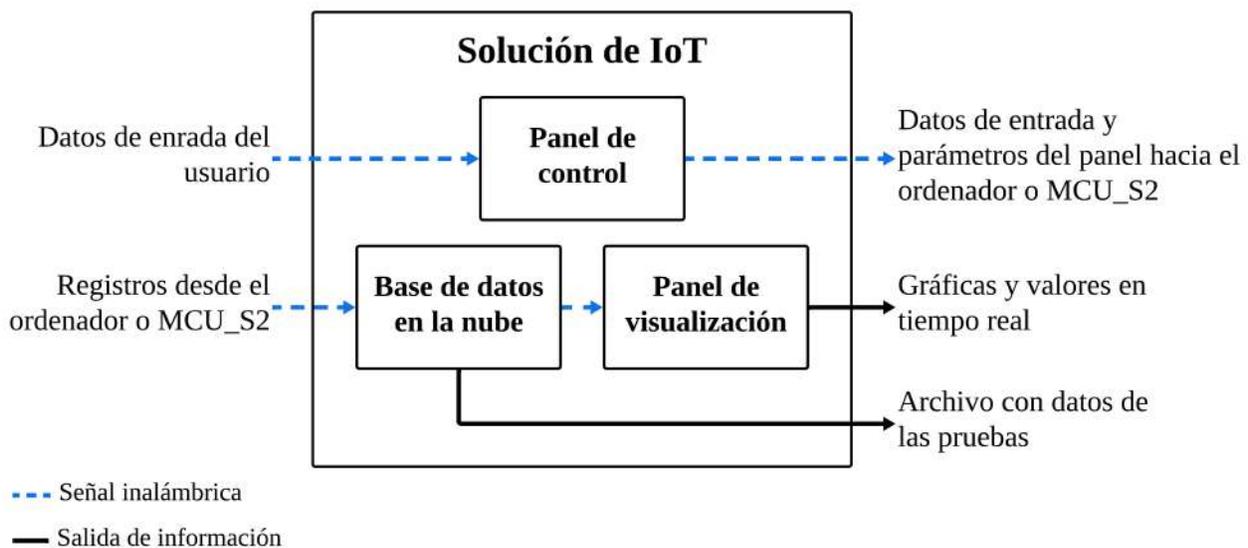


Figura 7.1: Descomposición funcional de la solución de IoT.

Se puede observar que la solución tiene tres componentes principales:

- **Panel de control:** consiste de una interfaz web que el usuario utiliza para introducir los parámetros de entrada que definen el comportamiento de la solución de SAS. Asimismo, es deseable que usuario pueda usar esta interfaz para modificar los parámetros de los paneles solares existentes, o bien agregar nuevos paneles solares a futuro. Por lo tanto, esta componente es el medio utilizado por el ordenador o MCU_S2, dependiendo de la plataforma de SAS que se esté usando, para obtener los parámetros de entrada y del panel solar, indicados previamente en las Tablas 5.2 y 5.3, respectivamente.

- **Base de datos en la nube:** consiste en un archivo que el ordenador o MCU_S2 accede a través de internet para guardar los registros periódicos del comportamiento del SAS, así como las entradas del usuario asociadas. Estos coinciden con los parámetros que se guardan en la memoria Micro SD, según se indicó en la sección 5.1.2.
- **Panel de visualización:** consiste de una interfaz web que accede a los registros guardados en la base de datos en la nube y permite al usuario observar de manera gráfica el comportamiento del SAS en tiempo real.

A continuación, se definen en mayor detalle los requisitos que cada componente de la plataforma IoT debe cumplir:

Panel de control

En la Tabla 5.2, se indicaron los parámetros de entrada que el usuario puede modificar y que luego son enviados a las plataformas de SAS desarrolladas para definir su comportamiento. A estas se le debe sumar la variable *Update*, que le permite al usuario definir si las plataformas SAS reciben o no parámetros del panel de control, y la variable *Start/Stop*, la cual permite activar o suspender la salida de potencia de las plataformas. Por lo tanto, en total se tienen 10 variables a controlar. Por otra parte, es deseable que el usuario pueda modificar los parámetros de los paneles solares a utilizar, listados en la Tabla 5.3, o bien, agregar nuevos paneles. Por ende, esto añadiría 16 variables adicionales por cada panel solar. Si se consideran los 4 paneles solares seleccionados, según la Tabla 5.1, más al menos un panel solar adicional, se tendrían 80 variables adicionales. Por lo tanto, el total de variables a controlar que deben poder enviarse desde el panel de control de la solución IoT, es igual a 90.

Por otra parte, las variables de salida que las plataformas de SAS envían a la solución IoT, con el fin de que estas se guarden en la base de datos, son listadas en la Tabla 5.4. Estas son en total 16 variables, pero se deben sumar además los 8 parámetros de entrada del usuario de la Tabla 5.2, los cuales están asociados directamente con las salidas. Por lo tanto, el total de variables distintas que se envían a la base de datos de la solución IoT, es igual a 24.

Base de datos

Como se mencionó anteriormente, en total se envían 24 variables diferentes a la base de datos, las cuales se listan en la Tabla 7.1 con su respectivo tamaño máximo en Bytes, considerando que cada carácter (ya sea letra, número o signo) equivale a 1 B. Se puede apreciar que el tamaño total del mensaje es de 153 B, o 176 B si se cuentan las comas que separan cada valor.

Tabla 7.1: Tamaño en Bytes de cada parámetro registrado en la base de datos.

Parámetro	Ejemplo	Tamaño (B)
PV	1	1
Ns	1	1
Np	1	1
G	1000.12	7
T	-10.12	6
Ang	10.12	5
CV	1	1
$V_{out,calc}$	10.12345	8
$V_{out,meas}$	10.12345	8
$I_{out,calc}$	1.12345	7
$I_{out,meas}$	1.12345	7
$P_{out,calc}$	10.12345	8
$P_{out,meas}$	10.12345	8
$R_{out,calc}$	10000.12	8
$R_{out,meas}$	10000.12	8
Err _V	10.12345	8
Err _I	10.12345	8
Err _P	10.12345	8
Err _R	10.12345	8
n	1.23	4
I_{ph}	1.12345	7
I_o	1.12E-123	9
R_s	1000.123	8
R_{sh}	10000.123	9
Total		153

Por otra parte, tal y como se detalla en la Tabla 3.7, el caso ideal es que se puedan enviar registros cada 0.5 segundos por 8 horas consecutivas al día, durante todos los días de un mes. El caso marginal es que se puedan enviar registros cada 10 segundos por 4 horas consecutivas al día, durante todos los días del mes. Por lo tanto, la base de datos debe tener la capacidad de recibir y guardar al menos 57600 registros por día o 1728000 por mes en el caso ideal, o al menos 1440 registros por día o 43200 por mes en el caso marginal. Además, dado que la fase de evaluación de desempeño de un CubeSat que es enviado al espacio puede tomar desde meses hasta años, se debe garantizar que la base de datos permanezca funcional por un tiempo indefinido.

Panel de visualización

Es deseable poder visualizar el último valor registrado de todos los parámetros de salida de la Tabla 5.4, y todos los de entrada de la Tabla 5.2, con excepción de CVV, puesto que este ya está cubierto en los valores de salida. Por lo tanto, en el panel de visualización se debe poder observar el último valor registrado de 23 variables distintas. También debe existir un contador que indique cuánto tiempo ha pasado entre el primer y último registro.

Por otra parte, deben existir gráficas que muestren el cambio de ciertas variables con respecto al tiempo. Debe haber una para las mediciones de tensión, corriente, potencia y resistencia, y otra para los ARPEs asociados a estas mediciones. De ser posible, también deben existir gráficas que muestren los valores de una o varias variables en función de otra. Por ejemplo, la corriente en función de la tensión y la potencia en función de la tensión, para poder visualizar como se trazan las curvas características en tiempo real. O bien, el cambio de la tensión, corriente, potencia y resistencia con respecto a la irradiancia, temperatura o ángulo de incidencia. Por lo tanto, el panel de visualización debe contener, como mínimo, dos gráficas distintas que monitoreen el cambio con respecto al tiempo de 8 variables.

7.1. Generación y selección de conceptos

7.1.1. Protocolos de comunicación

Debido a que la implementación de la solución IoT se debe hacer con un microcontrolador ESP8266, los protocolos de comunicación que se consideran son HTTP y MQTT, ya que son los más utilizados, con más recursos disponibles y que generalmente son soportados en plataformas de IoT. Cabe destacar que en esta subsección se discuten las ventajas y desventajas de cada protocolo, pero no se pretende hacer una selección definitiva. Esto debido a que primero se debe seleccionar la plataforma de IoT más apropiada, por lo que el objetivo de la discusión es determinar cuál es protocolo más apropiado para la solución de IoT a diseñar, y darle mayor prioridad a plataformas que ofrezcan la opción de utilizar este protocolo.

Como se mencionó en el Capítulo 2, HTTP funciona como un modelo cliente-servidor, donde cada transmisión se realiza a través de una conexión. Es especialmente eficiente para enviar o recibir grandes cantidades de datos, como imágenes, videos u otros archivos pesados. Sin embargo, HTTP sólo crea una conexión cuando se necesita enviar una solicitud [117]. Por otra parte, MQTT es un protocolo de publicación/suscripción utilizado frecuentemente en comunicaciones máquina-a-máquina (M2M) debido a su eficiencia y bajo consumo. A diferencia de HTTP, MQTT mantiene la conexión abierta, permitiéndole “reutilizar” la misma conexión para cada comunicación [118]. Esto es especialmente útil en aplicaciones donde se requiere una comunicación constante y en tiempo real entre los dispositivos y la plataforma IoT. En un estudio realizado donde se compara el rendimiento de ambos protocolos, MQTT resultó ser 20 veces más rápido y requiere 50 veces menos tráfico al publicar datos [119]. Por lo tanto, se concluye que si se desea obtener tiempos de refrescamiento muy bajos, como en el caso de este proyecto, el protocolo más adecuado es MQTT, y por ende es preferible seleccionar una plataforma que lo soporte.

7.1.2. Plataforma de IoT a utilizar

El primer paso consiste en realizar una búsqueda de las plataformas de IoT que se puedan utilizar libremente sin requerir el pago de una suscripción, y seleccionar la más apropiada de acuerdo a las necesidades establecidas.

A continuación, se resumen las características de algunas de las plataformas IoT que fueron consideradas:

Azure IoT Hub

Azure IoT Hub es una plataforma de IoT de Microsoft ofrece una suscripción gratuita permite transmitir hasta 8000 mensajes por día, con un tamaño máximo de 0.5 KB por mensaje [120]. En su documentación no especifica la existencia de un límite en cuanto a las variables que se pueden monitorear. Cuenta con herramientas para crear una interfaz de usuario con diversos elementos, como gráficos de línea multivariados, pulsadores, imágenes, y casillas donde se pueden introducir valores numéricos u observar el último valor conocido de las variables [121]. En su documentación tampoco se especifica un límite en cuanto al número de elementos gráficos que se pueden insertar. La plataforma permite la retención de los datos recibidos por un máximo de siete días. Este además permite a los dispositivos utilizar los protocolos MQTT y HTTP para establecer la comunicación entre los dispositivos y la nube.

AWS IoT Core

Esta plataforma de Amazon si bien se puede utilizar sin pagar, se considera más como un periodo de prueba que una suscripción gratuita, ya que posee límites que una vez son sobrepasados, no se vuelven a restablecer con el paso del tiempo. En total, se pueden enviar 500000 mensajes, o conectarse por 2250000 minutos [122]. Cabe destacar que dicho límite de mensajes imposibilita que se pueda cumplir con el valor ideal concerniente al número de registros que se pueden guardar en un mes, mientras que cumple con el valor marginal por menos de 12 meses. La documentación no especifica un máximo de tamaño por mensaje, límite de cuántas variables se pueden monitorear y tampoco un máximo de elementos gráficos que se pueden insertar en una interfaz de usuario. Tampoco hace mención de cuánto tiempo retiene la información enviada, por lo que se puede asumir que retiene la totalidad de los mensajes enviados durante el periodo de prueba. Soporta además los protocolos comunicación MQTT y HTTP.

Arduino Cloud

La versión gratuita de esta plataforma presenta limitaciones importantes, ya que solo se pueden monitorear un máximo de 5 variables y los datos enviados a la plataforma solo se retienen por un día [123]. Por otra parte, si bien no se especifica un límite de elementos gráficos que se pueden insertar, algunos elementos como gráficas de línea multivariadas están reservadas para la versión de paga. Asimismo, no se especifica un tamaño máximo de mensaje, ni un máximo de mensajes por día. Por otra parte, la programación del microcontrolador se puede hacer únicamente a través de su sitio web, por lo que también presenta un límite de compilaciones de programa por día y de tamaño de programa. Soporta los protocolos de comunicación MQTT y HTTP.

Adafruit IO

La versión gratuita únicamente permite el monitoreo de un máximo de 10 variables distintas y los datos son retenidos por un máximo de 30 días [124]. Por otra parte, si bien no se especifica un tiempo de refrescamiento mínimo, solo se puede actualizar el valor de un máximo de 30 variables por minuto, por lo tanto, dado que cada registro de la plataforma SAS está compuesta por 24 variables, solo permite enviar un registro por minuto. Por otra parte, incluye un máximo de 5 interfaces de usuario, y cada una puede tener un máximo de 10 elementos gráficos, que incluyen gráficos de líneas o de barras, medidores, interruptores, entre otros. Soporta los protocolos de comunicación MQTT y HTTP.

Google Workspace

La infraestructura de Google Workspace cuenta con la herramienta conocida como Google Apps Script. Esta permite utilizar JavaScript para el desarrollo aplicaciones que se pueden integrar con otras herramientas del entorno de Google Workspace, como Gmail, Drive, Calendario u hojas de cálculo [125]. Esta última es la más relevante para el contexto de este proyecto, ya que la implementación de todas las componentes de la solución IoT se puede hacer en hojas de cálculo y Google Apps Script se utiliza como intermediario para que el dispositivo pueda enviar o recibir información hacia o desde la hoja de cálculo, respectivamente. En la documentación de Google no se especifica ningún tipo de límite en cuanto al máximo de solicitudes que se pueden hacer por día, ni cuántas variables distintas se pueden enviar en un mismo mensaje. Además, ya que los registros periódicos de las variables se guardan directamente en una hoja de cálculo guardada en Google Drive, estos pueden permanecer guardados por un tiempo indefinido hasta que el usuario decida borrarlos. Por otra parte, dado que el espacio de almacenamiento disponible en una cuenta gratuita de Google Workspace es de 15 GB, resulta improbable que el usuario llegue a ese límite durante el periodo de pruebas de una misma misión de CubeSats. Además, el diseño de los paneles de control y de visualización se hace con herramientas gráficas incorporadas en las hojas de cálculo, por lo que no hay límite de cuántos se pueden implementar, ni de cuántas variables se pueden monitorear simultáneamente en una sola gráfica. También presenta la facilidad de usar comandos de Excel para facilitar la manipulación de la información que recibe el dispositivo o que este envía. Asimismo, el formateo condicional de celdas puede utilizarse para facilitar la navegación del usuario en el panel de control, por ejemplo, pudiendo notificar al usuario cuando el valor numérico que introduce en una variable es inválido o se encuentra fuera del rango permitido. La clara desventaja de esta plataforma es que no presenta soporte de protocolo MQTT, por lo que el tiempo de refrescamiento en la base de datos se puede ver perjudicado.

En la Figura 7.2, se presenta una matriz de filtrado inicial que permite descartar las plataformas que presentan limitaciones importantes que impiden satisfacer las necesidades planteadas para el proyecto.

Criterios de selección	Plataforma IoT				
	Azure IoT Hub (ref.)	AWS IoT Core	Arduino Cloud	Adafruit IO	Google Workspace
Funcionalidad a largo plazo	0	-	0	0	0
Cantidad de mensajes por año	0	-	+	-	+
Cantidad de variables permitidas	0	0	-	-	0
Días de retención de datos	0	+	-	+	+
Limitaciones para interfaz de usuario	0	0	-	0	0
Restricciones en tiempo de refrescamiento	0	-	0	-	0
Soporte de MQTT	0	0	0	0	-
Suma +	0	1	1	1	2
Suma 0	7	3	3	3	4
Suma -	0	3	3	3	1
Evaluación neta	0	-2	-2	-2	1
Lugar	2	3	3	3	1
¿Continuar?	Sí	No	No	No	Sí

Figura 7.2: Matriz de filtrado inicial de plataformas IoT.

AWS IoT Core debido a que la versión gratuita consiste en un periodo de prueba limitado principalmente por la cantidad de mensajes que se pueden tramitar, no puede garantizar que permanezca funcional durante todo el periodo de pruebas de un CubeSats. Si bien se podría reducir el número de mensajes que se tramitan por día para alargar su vida útil, esto afectaría directamente el tiempo de refrescamiento o el número de horas que puede utilizarse al día. Por lo tanto, es descartado de la siguiente fase de selección.

Por otra parte, las limitaciones en la cantidad de variables en Arduino Cloud y en Adafruit IO son una deficiencia fundamental, puesto que no permiten el monitoreo de todos los parámetros requeridos en la solución de SAS diseñada. Asimismo, en Adafruit IO las restricciones en cuanto al máximo de variables que se pueden actualizar en un minuto se traducen en un tiempo de refrescamiento inaceptable, mientras que el bajo periodo de retención de datos y las limitaciones de la interfaz gráfica, le restan más puntos a Arduino Cloud. Por lo tanto, estas dos plataformas también son descartadas.

Finalmente, se puede observar que Azure IoT Hub y Google Workspace tienen un rendimiento muy similar, donde este último presenta la desventaja de no presentar soporte para el protocolo MQTT, pero lo compensa con una cantidad de mensajes y un periodo de retención de datos ilimitados. Por lo tanto, estas dos plataformas continúan a la siguiente fase de selección.

En la Figura 7.3, se presenta la matriz de selección que permite obtener el concepto ganador para la plataforma IoT.

Criterios de evaluación	Peso (%)	Plataforma IoT			
		Azure IoT Hub (ref.)		Google Workspace	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Tiempo que puede operar por día	40	3	1.2	5	2
Tiempo de refrescamiento mínimo	30	3	0.9	1	0.3
Días de retención de datos	25	3	0.75	5	1.25
Complejidad de configuración	5	3	0.15	5	0.25
Puntaje			3.00		3.80
Lugar			2		1

Figura 7.3: Matriz de selección de plataforma IoT ganadora.

A pesar de que el tiempo de refrescamiento en Google Workspace se ve afectado por solo soportar el protocolo HTTP, la capacidad de garantizar que la plataforma no tenga ningún tipo de limitaciones en el tiempo que se puede utilizar por día termina resultando más relevante, ya que en el periodo de validación de un CubeSat puede resultar crítico realizar pruebas más extensas que sobrepasen el valor ideal establecido de 8 horas por día. Si bien se puede ajustar el tiempo de refrescamiento en el Azure IoT Hub para repartir los 8000 mensajes diarios en un periodo más extenso de 8 horas, eso termina afectando la única ventaja significativa que presenta sobre Google Workspace. Asimismo, el bajo periodo de retención de datos en Azure IoT Hub de tan solo 7 días, se eclipsado por el periodo prácticamente ilimitado que ofrece Google Workspace.

Finalmente, otra ventaja que ofrece Google Workspace es lo simple que resulta configurar las diferentes componentes de la solución con Google Apps Script, requiriendo solo unas pocas líneas de código en JavaScript para que las plataformas de SAS diseñadas puedan acceder a la información que requieren o hacer el envío de los registros, utilizando el menos complejo protocolo HTTP. Por otra parte, los paneles de control y de visualización pueden ser realizados y modificados de manera muy intuitiva por cualquier usuario con conocimientos básicos en comandos de hojas de cálculo. En contraste, el realizar la configuración inicial de la plataforma de Azure IoT Hub y establecer la comunicación bidireccional entre el microcontrolador y la nube puede resultar más complejo, requiriendo una revisión extensa y detallada de la documentación disponible, tanto de la ofrecida directamente por Microsoft, como por otros medios digitales. Por lo tanto, el claro ganador es Google Workspace.

7.2. Desarrollo de conceptos

7.2.1. Panel de control

En la Figura 7.4, se muestra la estructura principal del panel de control de la solución IoT implementada en Google Workspace.

SOLAR ARRAY SIMULATOR - CONTROL PANEL				
INPUT		VALUE	UNIT	GUIDELINES
Update		1		If you want to change multiple inputs at once, set this to 0 until all the changes are made.
Start/Stop		1		Must be either 0 = Stop or 1 = Start.
Solar panel	PV	1		Must be a value between 1 and 5. See Table 1 for reference.
Array Rows	Ns	2		Ns and Np must be positive integers. The product of Ns and Np must be equal to or less than 6.
Array Columns	Np	3		
Irradiance	G	1365	W/m ²	Must be a value between 0 and 1375.
Temperature	T	30.00	°C	Must be a value between -80 and 80.
Incidence Angle	θ	0.00	°	Must be a value between 0 and 90.
Control Variable	CV	0	V	Must be either 0 = Voltage or 1 = Current.
Control Variable Value	CVV	5.25	V	Must be a positive value equal to or less than the maximum voltage or current in Table 1.

Figura 7.4: Panel de control donde el usuario ingresa los parámetros de entrada. La tensión es seleccionada como variable de control.

Como se puede observar, se trata de una tabla en cuya primera columna se presenta el nombre y simbología de cada variable de entrada, en la segunda el valor que puede modificar el usuario, en la tercera la unidad de la variable, cuando corresponda, y en la cuarta la pautas que debe seguir el usuario para ingresar valores válidos en cada entrada.

La variable *Update* es binaria. Si es igual a 1, la plataforma SAS puede recibir todos los parámetros que requiera, si es igual a 0, el mensaje que llega a la plataforma es igual a "0", impidiendo que se envíen los valores actuales de las entradas. La pauta asociada a esta variable indica al usuario que, si desea cambiar el valor de múltiples variables al mismo tiempo, es recomendable colocar en 0 la Variable *Update*, ajustar las otras variables que requiera, y volver a colocar a *Update* en 1. De esta forma, todos los cambios llegan al SAS al mismo tiempo y se evita que el modelo analítico se calcule con la configuración parcial de los parámetros de entrada.

La variable *Start/Stop* igualmente es binaria. Si es igual a 0, indica que las salidas de tensión y corriente del SAS deben configurarse en 0 V y 0 A, respectivamente. Mientras que si es igual a 1, quiere decir que se debe proceder con la configuración del punto de operación indicado.

La variable *PV* indica cuál panel solar desea utilizar el usuario según la numeración indicada en la Tabla 5.1. El número 5 corresponde a un panel solar nuevo, no obstante, esto primero requiere que se agreguen sus parámetros, según se indica más adelante.

Las variables *Ns* y *Np*, indican el número de paneles solares en serie y en paralelo que el usuario desea, respectivamente. Las pautas son que ambos deben ser números enteros positivos, y que el producto de multiplicarlos debe ser igual o menor a 6. Esto debido a que se trata de un CubeSat de 1U, donde en cada cara puede haber un máximo de 2 paneles solares, y en un mismo instante únicamente 3 caras del CubeSat pueden estar expuestas a radiación solar.

Las variables *G*, *T* y θ , corresponden a los valores de irradiancia, temperatura de celda y ángulo de incidencia, respectivamente. Las pautas son que los valores introducidos deben encontrarse dentro del rango establecido para cada parámetro, según lo establecido en la Tabla 3.6.

La variable *CV* es binaria, y cuando es igual a 0 indica que el usuario desea introducir un valor de tensión para que el modelo calcule la corriente correspondiente, mientras que si es igual a 1 indica que se desea introducir un valor de corriente para calcular la tensión correspondiente.

La variable *CVV* es donde el usuario puede introducir el valor de tensión o corriente que desea. En las pautas se indica que debe ser un número positivo que no exceda el máximo permitido, según se indica en la Tabla 1 de ese documento, la cual se puede observar en la Figura 7.5.

Table 1: Solar panels and their maximum ratings*.

Solar panel	Value	Max. Voltage (V)	Max. Current (A)
AZUR 3G30A	1	5.37	1.49
AZUR 4G32C	2	6.67	1.36
CESI CTJ30	3	5.12	1.42
Spectrolab UTJ	4	5.30	1.66
<New solar panel>**	5	0.00	0.00

Figura 7.5: Tabla que muestra al usuario los valores máximos de tensión que puede introducir.

Utilizando fórmulas de hojas de cálculo, se calcula el V_{oc} y la I_{sc} de cada panel solar en las condiciones actuales establecidas por el usuario (según los valores de N_s , N_p , G , T y θ), permitiendo así poder generar la tabla con los valores máximos permisibles. Se puede observar además que se utiliza la herramienta de formato condicional para resaltar cuál es el valor máximo que corresponde, según los valores de PV y CV seleccionados.

Se puede observar también que se le asigna un color diferente a la tensión y a la corriente, siendo azul y magenta, respectivamente. En la Figura 7.6, se presenta cómo se ve el panel de control y la tabla asociada cuando CV es igual a 1.

SOLAR ARRAY SIMULATOR - CONTROL PANEL				
INPUT	VALUE	UNIT	GUIDELINES	
Update	1		If you want to change multiple inputs at once, set this to 0 until all the changes are made.	
Start/Stop	1		Must be either 0 = Stop or 1 = Start.	
Solar panel	PV	1	Must be a value between 1 and 5. See Table 1 for reference.	
Array Rows	N_s	2	N_s and N_p must be positive integers. The product of N_s and N_p must be equal to or less than 6.	
Array Columns	N_p	3		
Irradiance	G	1365	W/m ²	Must be a value between 0 and 1375.
Temperature	T	30.00	°C	Must be a value between -80 and 80.
Incidence Angle	θ	0.00	°	Must be a value between 0 and 90.
Control Variable	CV	1	A	Must be either 0 = Voltage or 1 = Current.
Control Variable Value	CVV	1.17	A	Must be a positive value equal to or less than the maximum voltage or current in Table 1.

Table 1: Solar panels and their maximum ratings*			
Solar panel	Value	Max. Voltage (V)	Max. Current (A)
AZUR 3G30A	1	5.37	1.49
AZUR 4G32C	2	6.67	1.36
CESI CTJ30	3	5.12	1.42
Spectrolab UTJ	4	5.30	1.66
<New solar panels>**	5	0.00	0.00

* This refers to V_{oc} and I_{sc} , not V_{mp} and I_{mp} .
 ** To use a new solar panel, you must go to Sheet2 and type the required parameters in its respective column. You can also modify the current solar panels' parameters, if required.

Figura 7.6: Panel de control donde el usuario ingresa los parámetros de entrada. La corriente es seleccionada como variable de control.

Por otra parte, cuando el usuario introduce valores inválidos en una o varias de las casillas, se muestra el mensaje de la Figura 7.7.

There are problems with your inputs that need to be fixed before any updates can be applied!

Figura 7.7: Mensaje que notifica al usuario que introdujo uno o varios errores erróneos en las entradas.

Además, se utiliza la herramienta de formato condicional para resaltar con color naranja las entradas que presentan problemas, con el fin de facilitar al usuario la detección y corrección de errores, tal y como se observa en la Figura 7.8.

INPUT	VALUE	UNIT	GUIDELINES	
Update	-1		If you want to change multiple inputs at once, set this to 0 until all the changes are made.	
Start/Stop	2		Must be either 0 = Stop or 1 = Start.	
Solar panel	PV	6	Must be a value between 1 and 5. See Table 1 for reference.	
Array Rows	Ns	-1	Ns and Np must be positive integers. The product of Ns and Np must be equal to or less than 6.	
Array Columns	Np	8.5		
Irradiance	G	1800	W/m ²	Must be a value between 0 and 1375.
Temperature	T	343.00	°C	Must be a value between -80 and 80.
Incidence Angle	θ	91.00	°	Must be a value between 0 and 90.
Control Variable	CV	2	V	Must be either 0 = Voltage or 1 = Current.
Control Variable Value	CVV	-6.00	V	Must be a positive value equal to or less than the maximum voltage or current in Table 1.

Figura 7.8: Panel de control con valores erróneos resaltados para facilitar al usuario la corrección de errores.

Adicionalmente, en la segunda página de la hoja de cálculo se encuentra la tabla de la Figura 7.9, en la cual se encuentran todos los parámetros de cada panel solar que son requeridos por el modelo analítico del SAS para calcular el punto de operación indicado por el usuario. Asimismo, hay una columna adicional para que el usuario pueda introducir los parámetros de un panel solar nuevo, si así lo requiere. También es posible modificar los parámetros de los paneles existentes.

Parameter	Symbol	Unit	AZUR 3G30A	AZUR 4G32C	CEST CTJ30	Spectrolab UTJ	<New solar panel>
Panel efficiency	η	%	30	32	29.5	28.3	
Cell temperature at ref. conditions	$T_{c,ref}$	$^{\circ}\text{C}$	28	25	25	28	
Solar irradiance at ref. conditions	G_{ref}	W/m^2	1367	1367	1367	1353	
Open Circuit Voltage	V_{oc}	V	2.699	3.375	2.6	2.66	
Short Circuit Current	I_{sc}	A	0.496	0.455	0.473	0.548	
MPP Voltage	V_{mp}	V	2.387	2.919	2.306	2.335	
MPP Current	I_{mp}	A	0.487	0.434	0.459	0.52	
MPP Power	P_{mp}	W	1.162	1.266	1.06	1.214	
Number of cells in series (in a single solar panel)	$N_{s,cell}$		1	1	1	1	
Voc temperature coefficient	K_{Voc}	$\text{V}/^{\circ}\text{C}$	-6.20E-03	-8.40E-03	-8.40E-03	-5.90E-03	
Isc temperature coefficient	K_{Isc}	$\text{A}/^{\circ}\text{C}$	3.60E-04	7.00E-05	7.00E-05	3.80E-05	
Band gap	E_g	eV	1.6	1.6	1.6	1.6	
Diode ideality factor	n		1.23	1.99	1.99	1.99	
Photoelectric current	I_{ph}	A	0.496	0.4562	0.4731	0.549	
Diode inverse saturation current	I_o	A	9.44E-38	9.45E-30	3.86E-23	2.26E-23	
Series resistance	R_s	Ω	0.3555	0.5705	0.2132	0.2408	
Shunt resistance	R_{sh}	Ω	996.98	207.91	677.7	138.29	
Root-Mean-Squared-Error	RMSE		0.0021	0.01537	0.00919	0.0084	

Figura 7.9: Tabla en la que el usuario puede modificar los parámetros de los paneles solares existentes, o introducir los de un panel nuevo.

En cuanto a cómo se envían los datos del panel de control a la plataforma SAS, primeramente se escribe en la celda A1 una fórmula que le permita tomar un valor de “0”, en caso de que *Update* sea igual a 0 o alguna de las entradas presente un valor inválido, o en caso contrario, listar todos los parámetros requeridos por el SAS en un solo mensaje, según el formato indicado anteriormente en el Capítulo 5:

$$\begin{aligned}
 &“i, PV, N_s, N_p, G, T, \theta, CV, CVV, \\
 &d, T_{c,ref}, G_{ref}, V_{oc}, I_{sc}, V_{mp}, I_{mp}, P_{mp}, N_{s,pv}, K_v, K_i, E_g, \\
 &p, n, I_{ph,ref}, I_{o,ref}, R_{s,ref}, R_{sh,ref}”
 \end{aligned}$$

Ahora se procede a explicar el funcionamiento del código desarrollado en JavaScript con la herramienta Google Apps Script. Se tiene una función *doGet* que se utiliza para procesar las solicitudes HTTP. Este extrae la hoja de cálculo activa y selecciona la página denominada “Control_Panel”. Luego, examina los parámetros enviados en la URL de la solicitud. Si el parámetro asociado a una acción es “readA1”, recupera el valor de la celda A1 de la página y lo retorna al cliente como texto sin formato. Este es el mensaje que contiene la información requerida por el SAS para procesar el modelo analítico.

Por otra parte, si la acción es “readA2”, primero escribe en la celda H2 (elegida aleatoriamente) la fecha y hora en que se recibió la solicitud. La celda A2 tiene una fórmula que permite tomar el valor de la celda H2 y formatearlo para que su contenido se muestre como

“ $t, \langle día \rangle, \langle mes \rangle, \langle año \rangle, \langle hora \rangle, \langle minuto \rangle, \langle segundo \rangle$ ”. Luego, la función *doGet()* recupera el valor de la celda A2 y lo devuelve al cliente como texto sin formato. Este es el mensaje que contiene la información que el MCU_M utiliza para configurar el RTC externo.

A continuación, se presenta el formato de las direcciones URL que permiten ejecutar las acciones antes descritas, donde $\langle GAS_ID \rangle$ se refiere al identificador único asignado a un proyecto de Google Apps Script.

```
“https://script.google.com/macros/s/<GAS_ID>/exec?action=readA1”
```

```
“https://script.google.com/macros/s/<GAS_ID>/exec?action=readA2”
```

Adicionalmente, se implementaron funciones que permiten cambiar el valor de las celdas donde se introducen los valores de entrada, que abarca de la celda D6 a la D15, a través de solicitudes HTTP. Esto con el objetivo de, por ejemplo, poder escribir un programa que pueda cambiar los valores de cada una de las entradas de manera periódica según se indique en un archivo .CSV, lo cual es de gran utilidad para conducir pruebas en el SAS.

Para lograr esto, se verifica si hay un parámetro llamado “D6” (o la celda que se desee modificar) en la solicitud HTTP. Si existe dicho parámetro, escribe el valor indicado para el parámetro “D6” en la celda D6. Este procedimiento se repite con todas las otras celdas que contienen los valores de las entradas.

A continuación, se presenta un ejemplo de dirección URL que permite configurar la variable G con un valor de 1367 W/m^2 en la celda D11, CV como tensión en la celda D14, y CVV como 4.2 V en la celda D15. Cabe destacar que se puede modificar el valor de las 10 variables de entrada con una sola dirección URL, si es requerido.

```
“https://script.google.com/macros/s/<GAS_ID>/exec?D11=1367&D14=0&D15=4.2”
```

7.2.2. Base de datos

Debido a que la plataforma de SAS #1 requiere del monitoreo de la resistencia de salida de la carga electrónica y la plataforma de SAS #2 no, se deben crear dos variantes de la base de datos. Por lo tanto, se decide mantenerlas en hojas de cálculo separadas entre sí y del panel de control.

En el código de Google Apps Script desarrollado para la base de datos nuevamente se utiliza una función *doGet()*, donde se accede a la hoja de cálculo activa y se selecciona la página denominada "Database". Luego, se busca la última fila que tiene contenido en dicha página y se agrega una nueva fila debajo. Luego, en las columnas A y B escribe la fecha y hora actual, respectivamente. Luego, se procesa cada uno de los parámetros recibidos, asignándolos a las columnas correspondientes en la fila.

Con el objetivo de reducir el tamaño del mensaje que envía la plataforma de SAS lo máximo posible, los valores correspondientes a la potencia y resistencia, tanto la calculada como la medida, se calculan directamente en el código a partir de los valores de tensión y corriente. Asimismo, los ARPEs asociados a la tensión, corriente, potencia y resistencia,

también se calculan directamente en el código. De esta forma, 8 de las variables que se guardan en cada registro son calculadas y escritas en la base de datos directamente por Google Apps Scripts, reduciendo el tamaño del mensaje que debe enviar la plataforma SAS de 175 B a 102 B, es decir, se da una reducción de aproximadamente 41.7%.

A continuación, se muestra un ejemplo de URL que puede ser enviada por cualquiera de las dos plataformas de SAS para guardar un registro en la base de datos.

```
“https://script.google.com/macros/s/<GAS_ID>/exec?
PV=1&Ns=2&Np=3&G=1367&T=28&Ang=0&CV=0&Vc=5&Vm=4.96&Ic=1.21&Im=1.22
&n=1.23&Iph=0.49&Io=9.44E-38&Rs=0.36&Rsh=996.98”
```

En la Figura 7.10, se muestra una captura de pantalla de cómo se ve la base de datos una vez que se han guardado varios registros.

Date	Time	PV	Ns	Np	G	T	θ	CV	Vcalc	Vmesa	Icalc	Imesa	Pcalc	Pmesa	Icalc	Rmesa	Err V	Err I	Err P	Err R	n	Iph	Io	Rs	Rsh
10/11/2023	20:37:57	1	2	1	1367	28	0	0	4.78000	4.77198	0.48542	0.48054	2.32510	2.33130	0.02886	0.76783	0.16776	0.43542	0.26893	0.60995	1.23	0.40660	9.44E-38	0.36	996.98
10/11/2023	20:38:01	1	2	1	1367	28	0	0	4.78000	4.77238	0.48542	0.48049	2.32510	2.33128	0.02886	0.77224	0.15937	0.42576	0.26571	0.55402	1.23	0.40660	9.44E-38	0.36	996.98
10/11/2023	20:38:06	1	2	1	1367	28	0	0	4.78000	4.77223	0.48542	0.48068	2.32510	2.33200	0.02886	0.75871	0.16247	0.46138	0.30018	0.71382	1.23	0.40660	9.44E-38	0.36	996.98
10/11/2023	20:38:13	1	2	1	1367	28	0	0	4.78000	4.77346	0.48542	0.48076	2.32510	2.33386	0.02886	0.76696	0.15378	0.47963	0.34259	0.51359	1.23	0.40660	9.44E-38	0.36	996.98
10/11/2023	20:38:31	1	2	1	1367	28	2.5	0	4.78000	4.77562	0.48595	0.48051	2.32285	2.33283	0.03634	0.76879	0.08155	0.52500	0.43377	0.60880	1.23	0.40528	9.44E-38	0.36	997.93
10/11/2023	20:38:35	1	2	1	1367	28	2.5	0	4.78000	4.77182	0.48595	0.48048	2.32285	2.33096	0.03634	0.77368	0.17115	0.52107	0.34802	0.63708	1.23	0.40528	9.44E-38	0.36	997.93
10/11/2023	20:38:40	1	2	1	1367	28	2.5	0	4.78000	4.77823	0.48595	0.48852	2.32285	2.33427	0.03634	0.78120	0.03809	0.52785	0.48137	0.50867	1.23	0.40528	9.44E-38	0.36	997.93
10/11/2023	20:38:45	1	2	1	1367	28	2.5	0	4.78000	4.77748	0.48595	0.48883	2.32285	2.33536	0.03634	0.77334	0.05282	0.58165	0.53851	0.64058	1.23	0.40528	9.44E-38	0.36	997.93
10/11/2023	20:39:02	1	2	1	1367	28	5	0	4.78000	4.76739	0.48450	0.48048	2.31613	2.31922	0.06490	0.79987	0.26379	0.38804	0.13320	0.65922	1.23	0.40413	9.44E-38	0.36	1000.79
10/11/2023	20:39:06	1	2	1	1367	28	5	0	4.78000	4.76012	0.48450	0.48855	2.31613	2.32040	0.06490	0.80322	0.22755	0.41250	0.18441	0.62527	1.23	0.40413	9.44E-38	0.36	1000.79
10/11/2023	20:39:13	1	2	1	1367	28	5	0	4.78000	4.77324	0.48450	0.48658	2.31613	2.32256	0.06490	0.80846	0.14142	0.41950	0.27740	0.57209	1.23	0.40413	9.44E-38	0.36	1000.79
10/11/2023	20:39:17	1	2	1	1367	28	5	0	4.78000	4.77135	0.48450	0.48655	2.31613	2.32148	0.06490	0.80655	0.18107	0.41250	0.23108	0.59149	1.23	0.40413	9.44E-38	0.36	1000.79
10/11/2023	20:39:36	1	2	1	1367	28	7.5	0	4.78000	4.77371	0.48221	0.48460	2.30494	2.31335	0.01280	0.80081	0.13151	0.48687	0.38271	0.62528	1.23	0.401797	9.44E-38	0.36	1005.58
10/11/2023	20:39:42	1	2	1	1367	28	7.5	0	4.78000	4.77650	0.48221	0.48465	2.30494	2.31489	0.01280	0.80568	0.07325	0.58683	0.43119	0.57746	1.23	0.401797	9.44E-38	0.36	1005.58
10/11/2023	20:39:47	1	2	1	1367	28	7.5	0	4.78000	4.77586	0.48221	0.48463	2.30494	2.31494	0.01280	0.80670	0.08667	0.48194	0.38496	0.58507	1.23	0.401797	9.44E-38	0.36	1005.58
10/11/2023	20:40:05	1	2	1	1367	28	19	0	4.78000	4.76739	0.47893	0.48053	2.28930	2.29880	0.80951	0.92102	0.26379	0.33417	0.06950	0.59901	1.23	0.400465	9.44E-38	0.36	1012.36
10/11/2023	20:40:10	1	2	1	1367	28	19	0	4.78000	4.76824	0.47893	0.48064	2.28930	2.29179	0.80951	0.92064	0.24813	0.35598	0.16888	0.59892	1.23	0.400465	9.44E-38	0.36	1012.36
10/11/2023	20:40:15	1	2	1	1367	28	19	0	4.78000	4.76965	0.47893	0.48079	2.28930	2.29175	0.80951	0.91422	0.27929	0.38741	0.16704	0.65421	1.23	0.400465	9.44E-38	0.36	1012.36
10/11/2023	20:40:21	1	2	1	1367	28	10	0	4.78000	4.76784	0.47893	0.48083	2.28930	2.29251	0.80951	0.91588	0.25448	0.39576	0.14928	0.64774	1.23	0.400465	9.44E-38	0.36	1012.36
10/11/2023	20:40:39	1	2	1	1367	28	12.5	0	4.78000	4.76856	0.47474	0.47655	2.26925	2.27244	10.06871	10.00649	0.23933	0.38103	0.14979	0.61800	1.23	0.404243	9.44E-38	0.36	1021.19
10/11/2023	20:40:43	1	2	1	1367	28	12.5	0	4.78000	4.76959	0.47474	0.47644	2.26925	2.27101	10.06871	10.06453	0.28952	0.35912	0.07759	0.63744	1.23	0.404243	9.44E-38	0.36	1021.19
10/11/2023	20:40:48	1	2	1	1367	28	12.5	0	4.78000	4.76837	0.47474	0.47663	2.26925	2.27273	10.06871	10.00338	0.24333	0.30788	0.15359	0.64892	1.23	0.404243	9.44E-38	0.36	1021.19

Figura 7.10: Ejemplo de registros guardados en la base de datos en la nube.

7.2.3. Panel de visualización

El panel de visualización de cada plataforma de SAS fue implementado en páginas secundarias de la misma hoja de cálculo donde están sus respectivas bases de datos. En este caso no es necesario escribir código en JavaScript, puesto que únicamente se utilizan fórmulas de celda, reglas de formato condicional y gráficos incorporados en hojas de cálculo típicas.

En la Figura 7.11, se presenta una división del panel donde se puede observar el tiempo transcurrido entre el primer y último registro, el último valor conocido de cada una de las entradas y salidas, y una sección donde se hace un análisis en tiempo real de los errores, mostrando, para cada una de las salidas, el máximo ARPE que se ha presentado y el MAPE, además de calcular el MAPE medio de todas las salidas.

Time elapsed		00 h, 19 min, 23 s	
INPUT		LKV	UNIT
Solar Panel	PV	AZUR 3G30A	
Array dimensions	$N_s \times N_p$	2 x 1	
Irradiance	G	1367.00	W/m ²
Temperature	T	28.00	°C
Incidence Angle	θ	87.50	°
Control Variable	CV	Voltage	V
OUTPUT		LKV	UNIT
Calculated Voltage	V_{calc}	4.58000	V
Measured Voltage	V_{meas}	4.57556	V
Calculated Current	I_{calc}	0.02122	A
Measured Current	I_{meas}	0.02098	A
Calculated Power	P_{calc}	0.09720	W
Measured Power	P_{meas}	0.09600	W
Diode Ideality factor	n	1.23000	
Photoelectric current	I_{ph}	0.02164	A
Saturation current	I_o	9.44E-38	A
Series resistance	R_s	8.150	Ω
Shunt resistance	R_{sh}	22856.350	Ω
OUTPUT ERRORS	MAXIMUM	MAPE	UNIT
Voltage	0.618	0.244	%
Current	1.888	0.415	%
Power	1.984	0.426	%
Resistance	2.254	0.588	%
MEAN MAPE		0.418	%
LKV = Last Known Value			
MAPE = Mean Absolute Percentage Error			

Figura 7.11: Panel que permite al usuario visualizar en tiempo real los últimos valores conocidos de las entradas y salidas, así como los errores presentes en el conjunto de datos actual.

En otra división del panel, se presentan gráficas que se van actualizando en tiempo real, tal y como se presenta en la Figura 7.12. Estas permiten visualizar el cambio con respecto al tiempo de las salidas medidas y los ARPEs asociados a ellas (gráficas a la izquierda). Así mismo, se presentan gráficas que permiten observar el cambio de la corriente y potencia en función de la tensión, permitiendo observar cómo se van formando las curvas características en tiempo real. Se presentan dos de estas gráficas, una toma los valores calculados o de referencia, y otra los medidos (gráficas a la derecha).

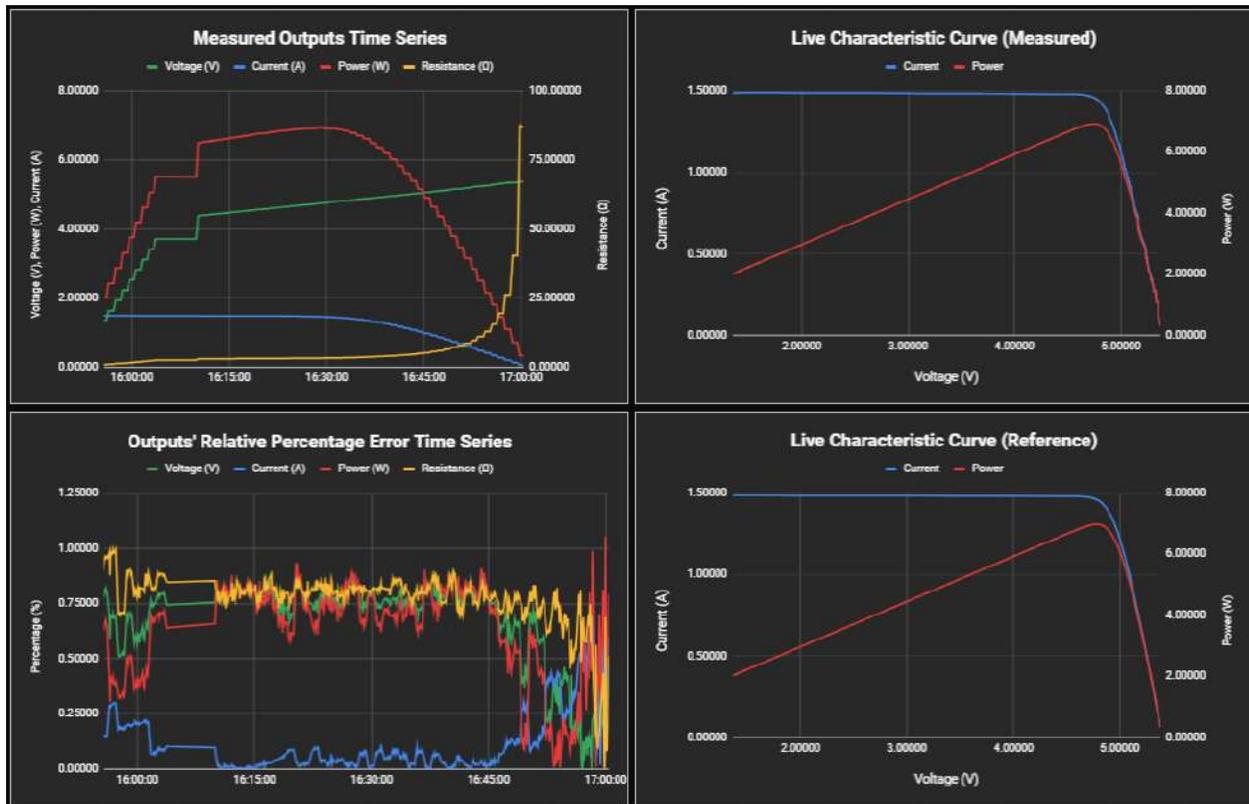


Figura 7.12: Gráficas en tiempo real que permiten al usuario visualizar el cambio de las salidas, los errores, y la forma de la curva característica del panel solar con el paso del tiempo.

En páginas aparte de la misma hoja de cálculo, también se pueden observar gráficas que presentan el cambio de las salidas en función de las condiciones externas introducidas por el usuario. En la Figura 7.13 se presenta la gráfica en función de la irradiancia, en la Figura 7.14 en función de la temperatura de celda, y en la Figura 7.15 en función del ángulo de incidencia.

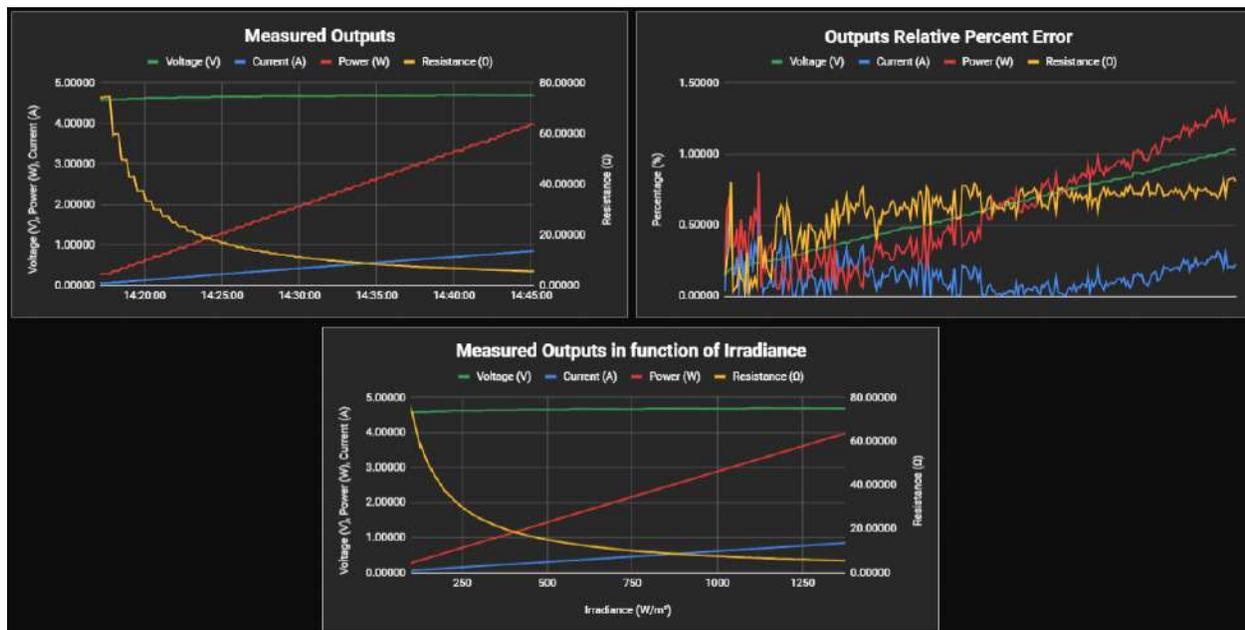


Figura 7.13: Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función de la irradiancia.

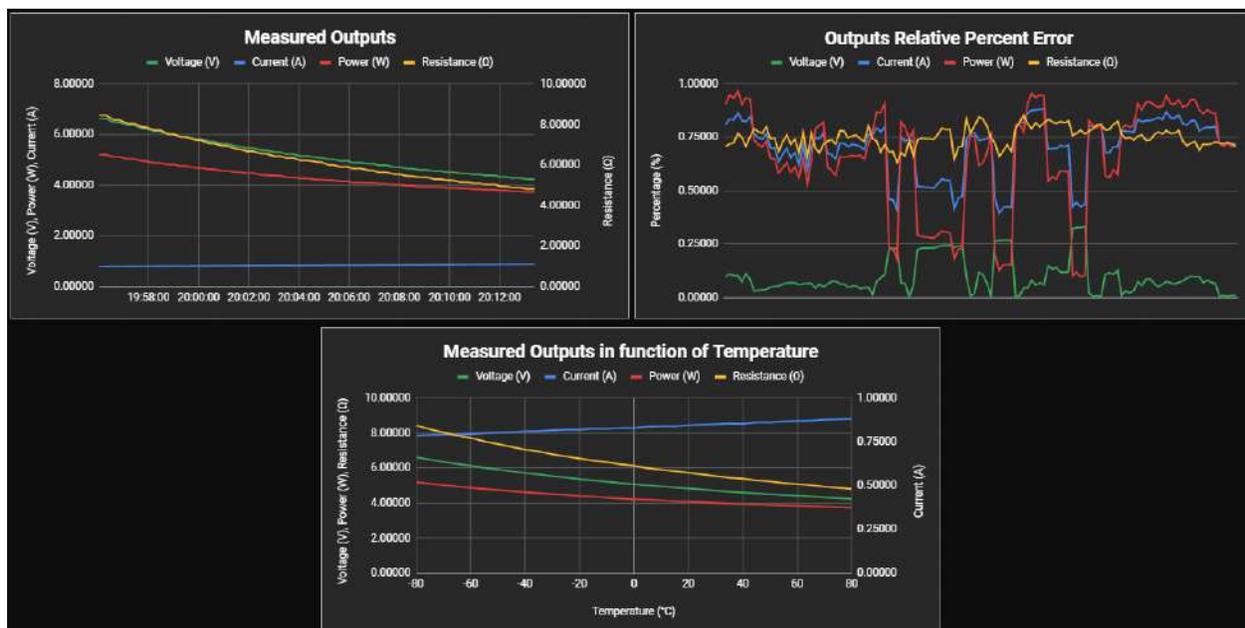


Figura 7.14: Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función de la temperatura de celda.

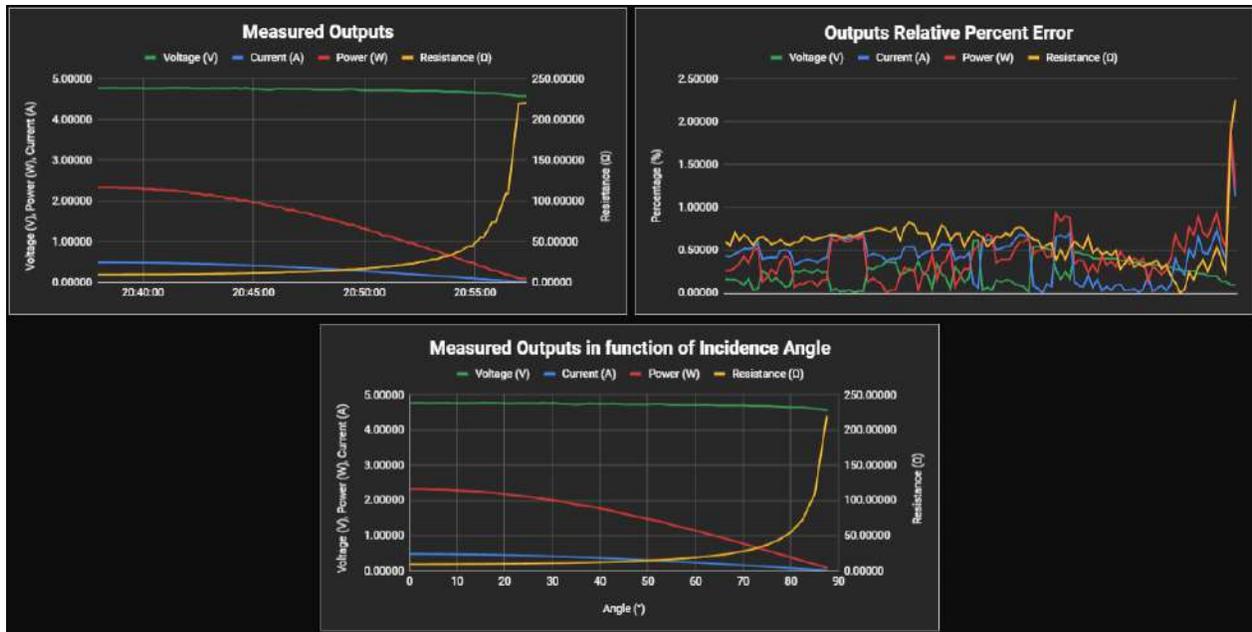


Figura 7.15: Gráfica en tiempo real que permiten al usuario visualizar el cambio de las salidas en función del ángulo de incidencia.

En el Anexo B, se presentan enlaces a repositorios donde se puede acceder los códigos desarrollados en JavaScript, y en el Anexo C se presenta una guía detallada de cómo configurar una hoja de cálculo para integrarla con *Google Apps Script*, y otra de cómo utilizar las diferentes componentes de la solución IoT desarrollada.

7.3. Validación de conceptos

Se desarrolló un programa adicional en cada plataforma de SAS para medir el tiempo que tarda en procesarse una solicitud hacia el panel de control, con el fin de recibir el contenido de la celda A1, el cual corresponde a la información de las entradas del usuario y los parámetros del panel. En total se tomaron más de 1000 muestras para cada plataforma. Además, se colocó un contador de solicitudes, el cual permitió validar que el 100% de las solicitudes realizadas fueron procesadas con éxito, ya que en todas se recibió el mensaje esperado. En la carpeta de *Google Drive*, cuyo enlace se facilita en el Anexo B, se presentan los programas utilizados y resultados crudos obtenidos.

En la Tabla 7.2, se presentan los resultados obtenidos de estas pruebas. Entre las especificaciones establecidas, se tiene como valor ideal que el tiempo promedio que toma en procesarse cada solicitud sea menor a 0.5 segundos, y como valor marginal que sea menor a 10 segundos. Por lo tanto, dado que se cumple con un amplio margen el valor marginal en ambas plataformas, y la importancia asignada a la especificación es de 3, se valida el cumplimiento del objetivo.

Tabla 7.2: Tiempos que tardan las plataformas de SAS en recibir respuesta del panel de control.

Parámetro	Plataforma 1	Plataforma 2
Tiempo promedio (s)	2.1853	4.7927
Desviación estándar (s)	0.6338	0.6945
Tiempo máximo (s)	15.4200	15.1170
Tiempo mínimo (s)	1.5710	3.9070

En el Capítulo 9, se presentan más pruebas realizadas en conjunto con la solución de SAS, permitiendo validar otros conceptos de la solución IoT.

Capítulo 8

Solución de Chasis

En este capítulo se abarca el proceso de diseño de un chasis para los distintos componentes de hardware de las soluciones de SAS e IoT. Cabe destacar que durante el desarrollo del presente proyecto de graduación no se procedió con la manufactura de un prototipo funcional, por lo que la validación del diseño se limita a simulaciones.

En la Figura 8.1, se presenta la secuencia de pasos seguida.

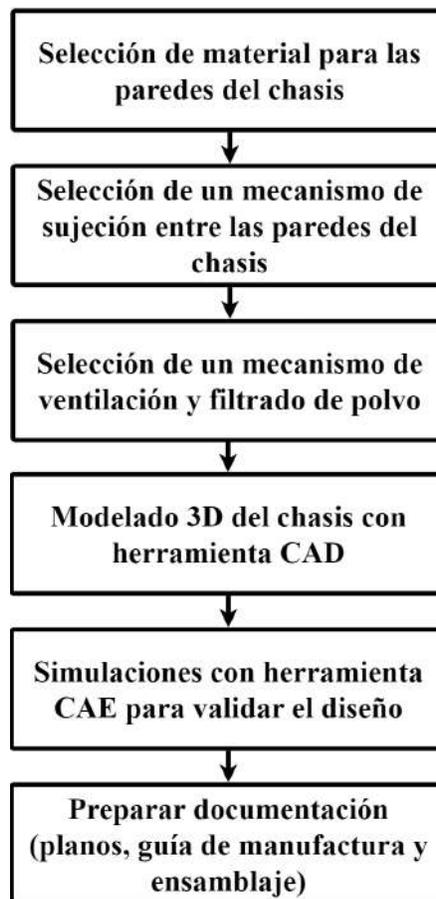


Figura 8.1: Secuencia de pasos seguida durante el proceso de diseño de la solución de chasis.

8.1. Generación y selección de conceptos

8.1.1. Selección del material de las paredes

Según se plasma en las necesidades y especificaciones del Capítulo 3, el cliente desea generar una propuesta de chasis económica y fácil de manufacturar. De preferencia, los materiales y métodos de manufactura utilizados deben encontrarse dentro de las instalaciones de campus central del TEC. A pesar de ser económico, debe ser robusto y fácil de transportar, por lo que se debe hacer un balance entre el peso y la rigidez del mismo. También debe ser fácil de ensamblar, requiriendo únicamente un kit de herramientas básico, debe poder ser limpiado frecuentemente sin que sus materiales se degraden, y debe poseer cualidades térmicas que garantice que los componentes internos no se vean afectados por temperaturas altas.

Algunos metales utilizados frecuentemente para el chasis de instrumentos electrónicos comerciales son láminas de aluminio 5052 o acero inoxidable 304 [126]. En el proveedor consultado, OnlineMetals, los precios de láminas de 0.02 pulgadas de grosor se listan en la Tabla 8.1. No obstante, para hacer envíos de EUA a Costa Rica, la tarifa es de aproximadamente \$90 adicionales. Se estima que comprando la lámina en un importador local, la tarifa de envío se puede reducir en un 50 %, obteniendo así un estimado más cercano a la realidad de cuánto pueden llegar a costar.

Tabla 8.1: Precios estimados por ft² de láminas de metal (en \$USD).

Lámina	Sin importar	Importado de EUA	Importador local
Aluminio 5052	4.91	27.41	16.16
Acero inoxidable 304	16.99	39.49	28.24

También es importante considerar los métodos de manufactura que se requieren. Si bien la Escuela de Mantenimiento Industrial del campus central de TEC tiene un taller de manufactura con herramientas que pueden permitir el corte de láminas metálicas, su uso es más restringido y puede necesitar de un permiso especial para su uso. Adicionalmente, se debe coordinar con un profesor o funcionario del taller para que supervise las labores.

Por otra parte, en el LAIMI del campus central del TEC se ofrecen láminas de MDF y acrílico de diversos grosores. En la Tabla 8.2, se presenta un resumen de sus precios. Cabe destacar que estas se venden en colones costarricenses (CRC), por lo que se aplicó una conversión de divisas tomando un tipo de cambio de \$USD 1 = CRC 520.

Tabla 8.2: Precios por ft² de materiales en el campus central del TEC (en \$USD).

Lámina	\$USD
Acrílico 3 mm de grosor	3.18
Acrílico 6 mm de grosor	5.37
MDF 3 mm de grosor	0.45
MDF 6 mm de grosor	0.59

Tanto el MDF como el acrílico pueden ser cortados mediante una cortadora láser CNC operada por estudiantes que realizan horas de asistencia en las instalaciones del LAIMI, el cual, en periodo lectivo, está abierto de lunes a viernes, de 7:30 a. m. a 12:00 a. m., y los sábados de 8:00 a. m. a 12:00 p. m. El MDF presenta el problema de que, si bien es más barato, es más propenso a ser dañado por la humedad, lo que complica su proceso de limpieza. Además, la madera suele mantener las temperaturas a lo interno del chasis más elevadas, lo cual es perjudicial para los componentes electrónicos internos. Por otra parte, el acrílico es resistente a la humedad, no es buen conductor de calor y además puede utilizarse en rangos de temperatura entre $-30\text{ }^{\circ}\text{C}$ a $90\text{ }^{\circ}\text{C}$ sin deformarse, y tiene un punto de fusión $160\text{ }^{\circ}\text{C}$ [127], por lo que no es factible que se vea dañado por las temperaturas de los componentes internos.

A partir de estos criterios, se realiza la matriz de decisión de la Figura 8.2, donde se observa que el claro material ganador es el acrílico. Inicialmente se selecciona el grosor más alto, de 6 mm, con el objetivo de garantizar mayor rigidez a la estructura y reducir la posibilidad de fallas mecánicas. No obstante, si en la etapa de validación se comprueba que los factores de seguridad obtenidos son muy elevados, se puede considerar reducir el grosor a 3 mm.

		Material de paredes del chasis							
		Acero inoxidable		Aluminio (Ref.)		Acrílico		MDF	
Criterios de evaluación	Peso (%)	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Costo del material	30	2	0.6	3	0.9	4	1.2	5	1.5
Facilidad de manufactura	30	3	0.9	3	0.9	5	1.5	5	1.5
Resistencia a la humedad	20	3	0.6	3	0.6	3	0.6	1	0.2
Cualidades térmicas	20	3	0.6	3	0.6	3	0.6	1	0.2
Puntaje			2.70		3.00		3.90		3.40
Lugar			4		3		1		2

Figura 8.2: Secuencia de pasos seguida durante el proceso de diseño de la solución de chasis.

8.1.2. Mecanismo de sujeción de paredes

Se requiere desarrollar un método de sujeción que garantice que el chasis mantenga su rigidez cuando se transporte, y que sus paredes se mantengan unidas.

Un método de sujeción comúnmente utilizado para lograr esto en acrílico, consiste en cortar el contorno del tornillo y la tuerca en una de las láminas, y en otra cortar un orificio para que pase el tornillo, tal y como se observa en la Figura 8.3. Si bien es un método económico, dado que no requiere piezas adicionales, presenta el problema de que si se aplica fuerza excesiva al apretar el tornillo, resulta muy fácil terminar quebrando la lámina. Por lo tanto, esta opción fue descartada.

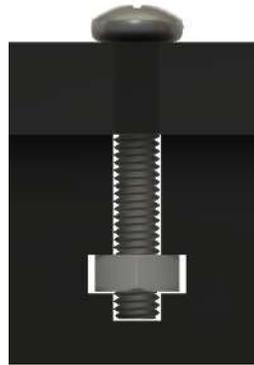


Figura 8.3: Método de sujeción cortando el contorno del tornillo en la lámina.

Otro método de sujeción propuesto consiste en involucrar elementos externos que faciliten la unión de los panales, de modo que en las dos láminas que se deseen unir únicamente se deba cortar un agujero. En la Figura 8.4, se muestra un soporte interior de 90° de 3 vías con roscado para tornillo M4, el cual es ampliamente utilizado para unir extrusiones 2020 de aluminio en aplicaciones de CNC. Si se coloca esta unión en cada una de las esquinas internas del chasis, se garantiza una unión robusta entre cada pared sin comprometer la integridad del material. Por lo tanto, aunque significa un costo mayor, se selecciona como método de sujeción.

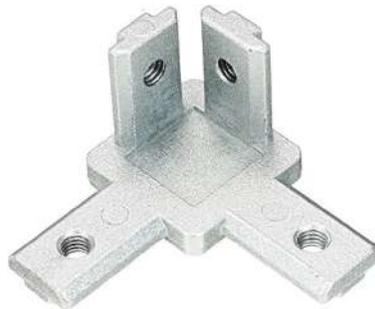


Figura 8.4: Método de sujeción con soporte interior de 90° de 3 vías.

8.1.3. Mecanismo de ventilación con filtrado de polvo

Dado que aún no se cuenta con el prototipo funcional de la planta MCP, no es posible determinar con precisión qué tanta ventilación se requiere en el chasis para mantener temperaturas bajas en los componentes electrónicos de potencia que se encuentran en dicha planta. No obstante, a partir de experiencias de otros investigadores del laboratorio que trabajaron con una versión previa de la planta, se concluye que no es necesario colocar componentes eléctricos como ventiladores, sino que solamente añadiendo orificios que permitan el flujo de aire hacia el interior del chasis es suficiente.

No obstante, también es importante considerar que debe existir un mecanismo que filtre la entrada de polvo. En ordenadores de escritorio modernos, es común encontrar mallados de PVC como el de la Figura 8.5, lo cuales se colocan sobre mallados con orificios más grandes directamente cortados en las paredes del chasis, con el fin de reducir la entrada de polvo pero a la vez permitir la ventilación del sistema. Es posible encontrar conjuntos de estos mallados que permiten cortarlos a la medida que se requiera, y por un bajo costo inferior a \$USD 10.

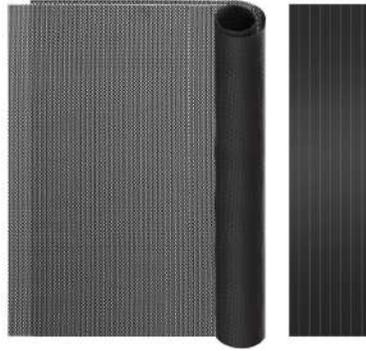


Figura 8.5: Malla de PVC para filtración de polvo [21].

Por lo tanto, se seleccionada como método de ventilación un mallado con orificios circulares que se corta directamente sobre el panel de acrílico (se omiten geometrías más complejas como hexágonos porque requieren un mayor tiempo de corte y por ende resultan más costoso), sobre el cual se colocan mallas de PVC para filtrar el polvo.

8.2. Desarrollo de conceptos

En la Figura 8.6, se presenta una imagen del frente del chasis diseñado. Se puede observar que diseño cuenta con todos elementos requeridos por el usuario: interruptor de encendido y apagado, interruptor rotatorio para seleccionar la entrada de potencia deseada (del convertidor AC-DC interno o de una fuente DC externa), la pantalla LCD1602 donde se muestran las lecturas de tensión, corriente y potencia, cuatro conectores banana (dos para conectar la entrada de potencia, y otros dos para tomar la salida del SAS), dos conectores USB para reprogramar al MCU_M y MCU_S2, y un conector apilable de 6 pines para reprogramar al MCU_S1 presente en la eventual planta MCP. También se puede observar que en cada panel lateral se encuentran orificios que permiten al usuario levantar el chasis más fácilmente.



Figura 8.6: Vista frontal del chasis.

En la Figura 8.7, se presenta una imagen trasera del chasis diseñado, donde se puede observar que se colocó un conector IEC320, el cual permite conectarse directamente a un enchufe con alimentación AC de 120 V a 60Hz.



Figura 8.7: Vista trasera del chasis.

En la Figura 8.8, se puede observar la implementación del mallado cortado sobre la superficie de acrílico de los paneles laterales, el cual se extiende sobre un área de 200 mm x 80 mm, y cuyos orificios tienen un diámetro de 2 mm y están separados entre sí por 1 mm. Por otra parte, el mallado de PVC se corta en dimensiones de 220 mm x 100 mm, y sobre los bordes metálicos se utiliza un taladro para cortar cuatro agujeros de 3 mm de diámetro cada uno, lo cual permite fijar el mallado en el chasis con tornillos y tuercas M3. En ambos paneles laterales se encuentra este mecanismo de ventilación y filtrado de polvo.

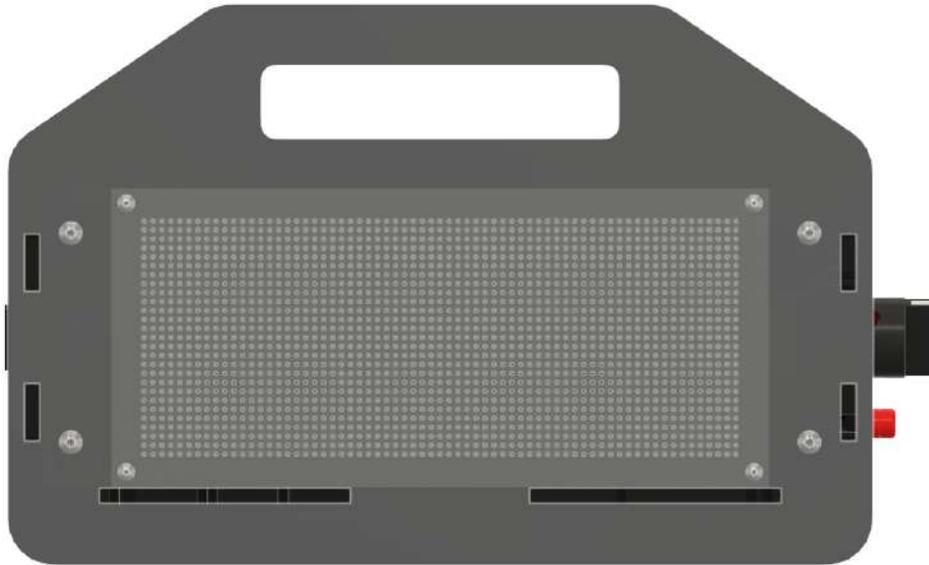


Figura 8.8: Vista lateral del chasis.

En la Figura 8.9, se puede observar la forma en que se organizaron los componentes de hardware dentro del chasis cuando se remueve el panel superior, los cuales fueron fijados con tornillos M3 a la base del chasis. Además, se garantiza que existan al menos 10 mm de distancia entre dos componentes, o entre los componentes con las paredes o uniones internas.

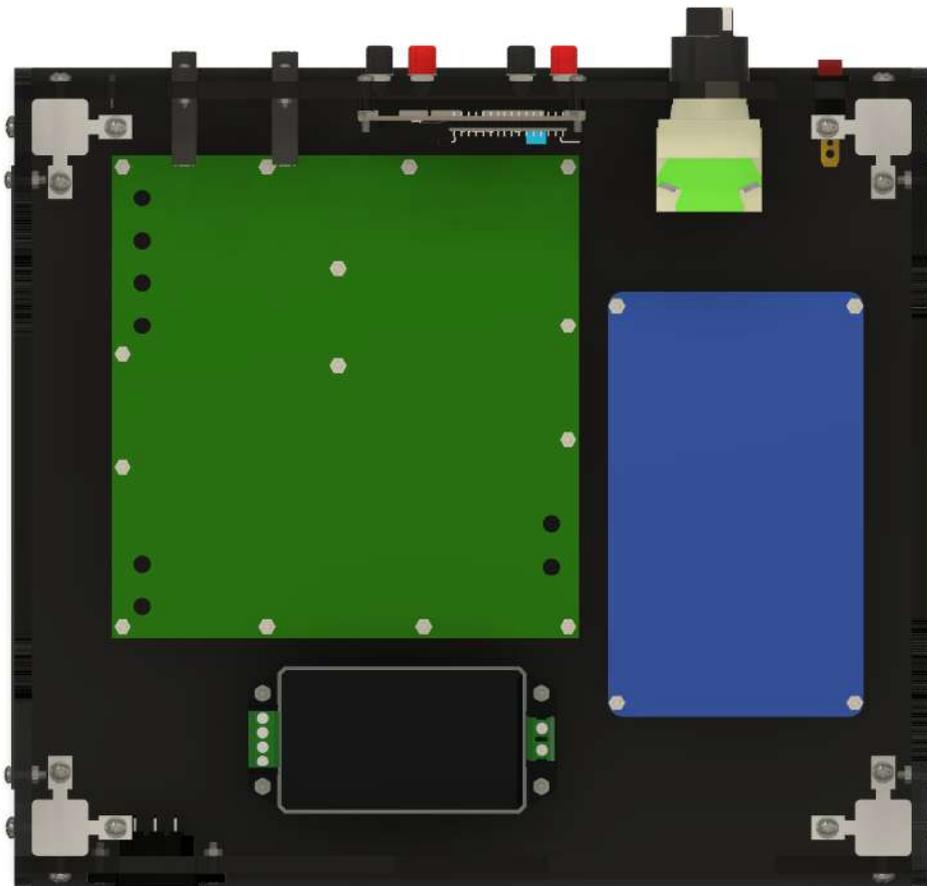


Figura 8.9: Vista superior del chasis.

Finalmente, en la Figura 8.10, se puede observar con mayor detalle el método de sujeción seleccionado para mantener todas las paredes del chasis unidas.

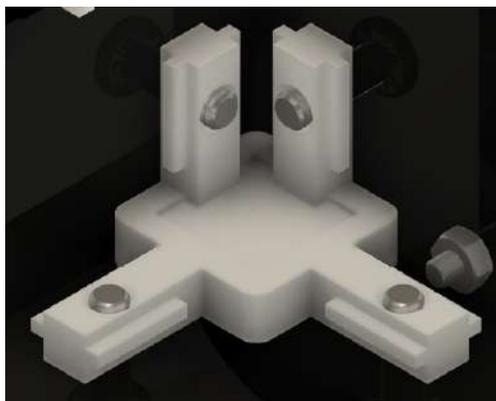


Figura 8.10: Detalle del método de sujeción seleccionado.

8.3. Validación del concepto

8.3.1. Prueba del método de sujeción y grosor

Para validar que las láminas de acrílico son capaces de soportar la presión generada al apretar los tornillos M4 y con el orificio roscado del soporte interior de 3 vías, es necesario calcular la fuerza de sujeción requerida, según se indica en la ecuación 8.1 [128]:

$$F = \frac{T}{K \cdot D} \quad (8.1)$$

Donde T es el torque de sujeción, K el factor de tuerca (que por convención es igual a 0.2) y D el diámetro nominal del tornillo. Cabe destacar que esta fórmula está diseñada para unidades del sistema inglés, por lo tanto se debe hacer la conversión correspondiente.

Para tornillos M4 de cabeza de botón, se recomienda usar un T igual a 3.1 Nm [129]. No obstante, se recomienda disminuir este torque cuando se intenta hacer la sujeción en materiales plásticos, con el fin de no dañarlos [130]. Dado que no se encontró un factor de reducción específico, se decidió utilizar el 50 % del torque de sujeción que se usaría en metales.

Al resolver la ecuación 8.1 con los valores indicados anteriormente, se obtiene la siguiente fuerza de sujeción:

$$F = 379,36 \text{ lb} = 172,08 \text{ kg} \quad (8.2)$$

Esta fuerza es ejercida en ambos lados de la lámina de aluminio. En el modelo, la superficie de la cabeza del tornillo y a la superficie del soporte interior de 3 vías que hacen contacto con el acrílico se separan, con el fin de conocer el área y de esta forma aplicar la presión que cada elemento genera sobre el acrílico únicamente en esas partes. Esto da como resultado que, del lado de la cabeza del tornillo se genere una presión de 7.26 MPa, y del lado del soporte de 1.12 MPa.

En la Figura 8.11, se presentan los resultados obtenidos de la simulación, donde se comprueba que el factor de seguridad obtenido es igual a 2.033, cumpliendo marginalmente con el indicador objetivo de 2. Por lo tanto, se recomienda disminuir en un mayor porcentaje el torque de sujeción aplicado. Por otra parte, dado que con una lámina de acrílico con un grosor 6 mm se pasó marginalmente esta prueba, se puede asumir que con un grosor menor de 3mm se hubiera presentado una falla, por lo tanto la selección de grosor fuera acertada.

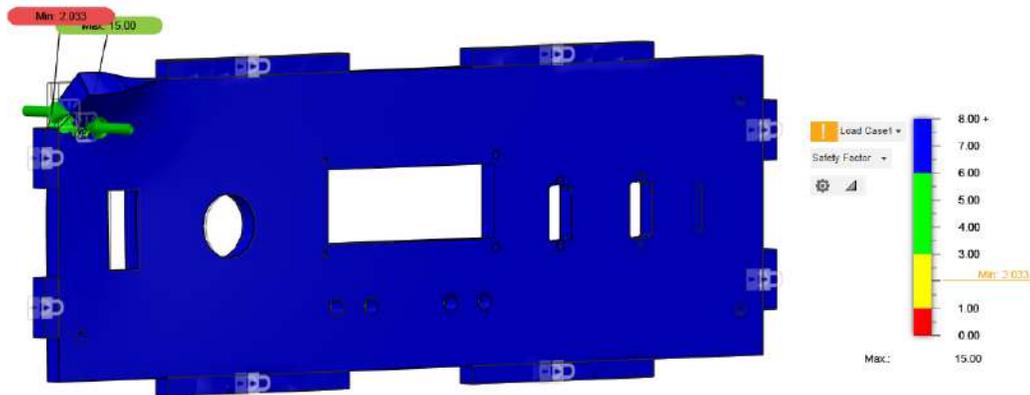


Figura 8.11: Resultado de la simulación de factor de seguridad en el método de sujeción.

8.3.2. Prueba en la base del chasis

Se realizó una simulación en la que se verifica que el factor de seguridad, la tensión mecánica y la elongación presentes en la lámina de acrílico, se encuentran dentro de rangos óptimos. Para esto, se colocó el peso cada componente de hardware presente en la base, incluyendo los tornillos, tuercas y separadores que los mantienen en posición fija.

El factor de seguridad obtenido fue de 15, mientras que el valor ideal definido entre los indicadores del proyecto para considerar el diseño como válido es de 3; en la Figura 8.12 se presentan los resultados de la simulación. La máxima tensión mecánica obtenida fue de 2.195 MPa, y en la Figura 8.13 se presentan los resultados. Finalmente, en la elongación máxima obtenida fue de 0.061 mm, y en la Figura 8.14 se presentan los resultados. Por lo tanto, se comprueba bajo diversos indicadores que la base de acrílico del chasis no debería experimentar fallas mecánicas durante su uso.

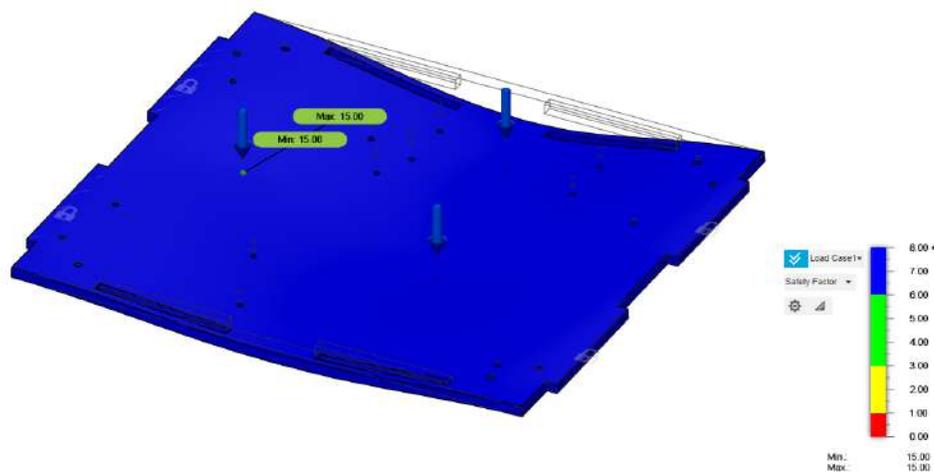


Figura 8.12: Resultado de la simulación de factor de seguridad en la base del chasis.

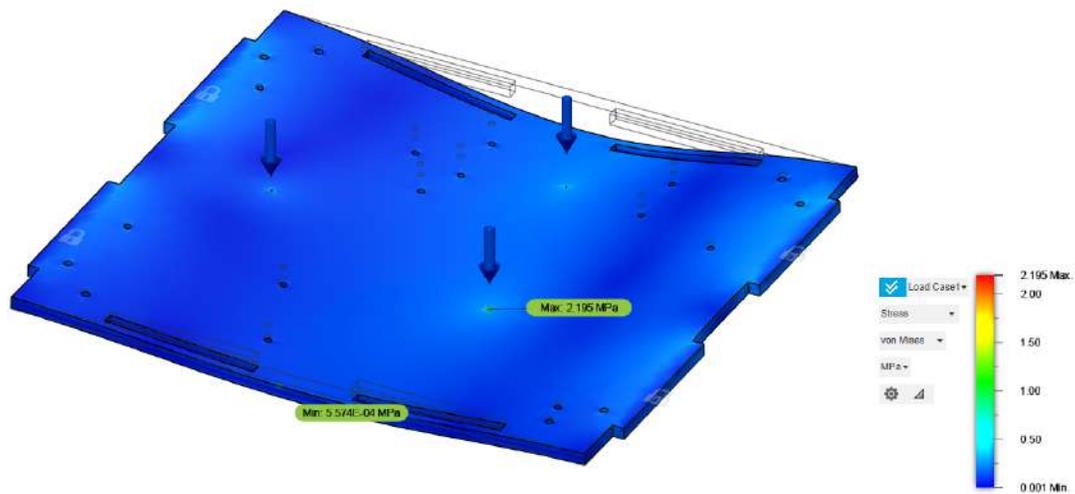


Figura 8.13: Resultado de la simulación de factor de seguridad en la base del chasis.

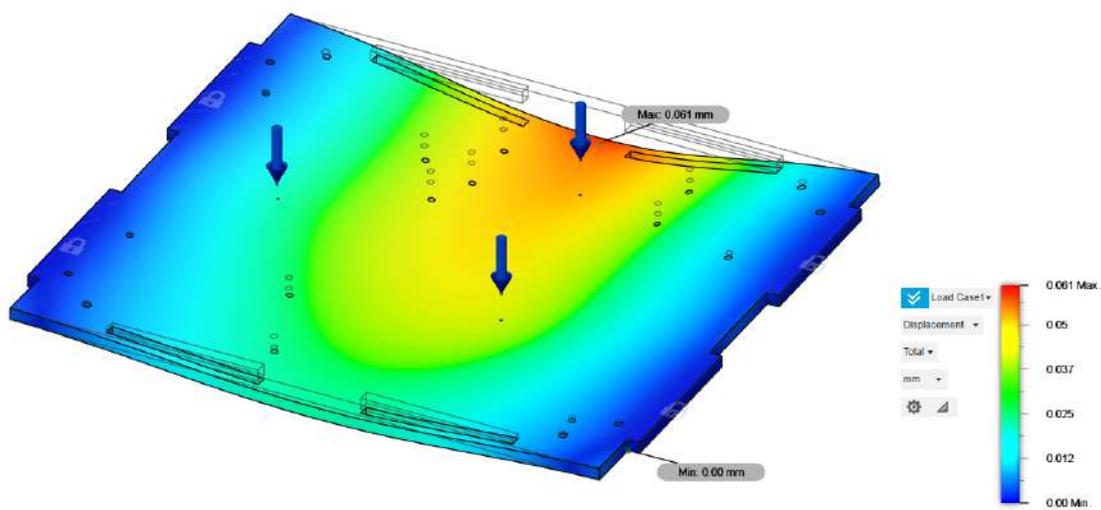


Figura 8.14: Resultado de la simulación de factor de seguridad en la base del chasis.

8.3.3. Prueba de las agarraderas

Se colocaron agarraderas en los paneles laterales para que el usuario pueda levantar y transportar el chasis fácilmente. Para garantizar que estas no sufran fallas mecánicas durante su uso, se realizó una simulación en la que se deja fija la base del panel y se aplica una carga equivalente a 3.5 kg, que es el peso estimado del chasis con todos sus componentes internos (más adelante se presenta el desglose de pesos). En la Figura 8.15, se aprecia que las agarraderas son capaces de soportar esa carga con un factor de seguridad de 15, superando con un amplio margen el valor ideal esperado de 3.

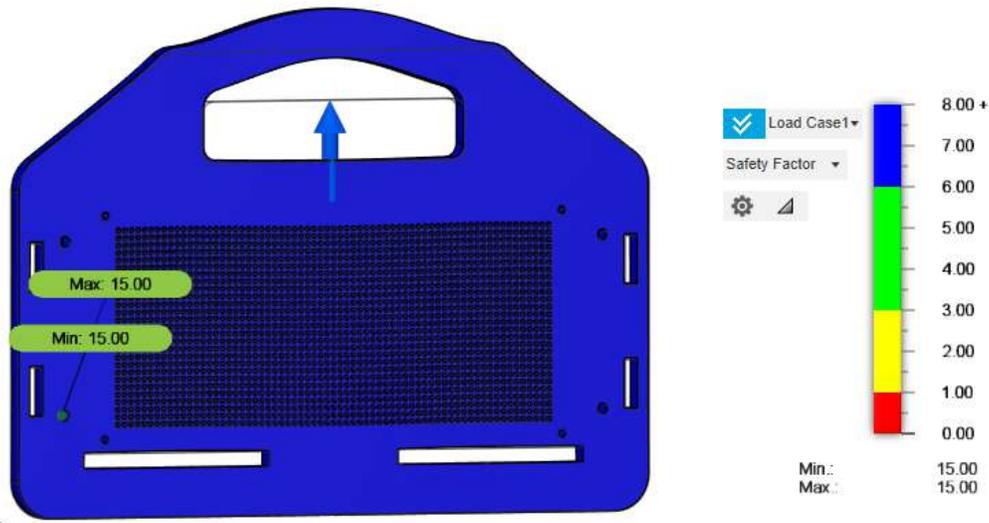


Figura 8.15: Resultado de la simulación de factor de seguridad en la base del chasis.

8.3.4. Desglose de materiales, peso y costos

Las paredes del chasis fueron diseñadas para ser manufacturadas a partir de láminas de acrílico transparente de 6 mm de grosor. En el LAIMI 2 del campus central del TEC, se pueden adquirir este tipo de láminas en dimensiones de 800 mm por 500 mm, por un precio aproximado de 12000 colones. En total se debe comprar 1.25 láminas para poder cortar las 6 paredes del chasis. En el Anexo D, se facilitan enlaces para los planos de cada pieza de acrílico, los modelos 3D de todo el diseño, las simulaciones, y los documentos en formato PDF para realizar el cortado láser. Se estima un costo aproximado de 10000 colones por este corte. Además, en el Anexo C, se facilita el enlace a una guía de manufactura y ensamble del chasis, incluyendo las conexiones eléctricas que se deben hacer.

En la Figura 8.16, se presenta el desglose de materiales y componentes del chasis, con sus respectivos pesos en kg y costos en dólares (\$USD), ya que esta fue la moneda solicitada por el cliente. El tipo de cambio utilizado fue de \$USD 1 = CRC 520.

Elemento	Cantidad	Peso Unitario	Peso Total	Precio Unitario	Precio Total
Panel frontal (acrílico de 6mm)	1	0.21569	0.21569	48.08	48.08
Panel trasero (acrílico de 6mm)	1	0.23792	0.23792		
Panel lateral (acrílico de 6mm)	2	0.32995	0.65990		
Panel superior (acrílico de 6mm)	1	0.62521	0.62521		
Panel inferior (acrílico de 6mm)	1	0.63895	0.63895		
Anillo espaciador (acrílico de 6mm)	4	0.00010	0.00042		
Malla PVC para filtrado de polvo	1	0.01300	0.01300	10.00	10.00
Convertidor AC-DC Mean Well IRM-60-24ST	1	0.20500	0.20500	27.00	27.00
Conector de alimentación AC IEC320	1	0.04200	0.04200	1.95	1.95
Conector banana hembra NB-418 rojo	2	0.00200	0.00400	1.00	2.00
Conector banana hembra NB-418 negro	2	0.00200	0.00400	1.00	2.00
Conector apilable de 6 pines de 2.54 mm	1	0.00100	0.00100	0.45	0.45
Interruptor rocker KCD3	1	0.00799	0.00799	3.50	3.50
Interruptor rotatorio LA38-20X3	1	0.03997	0.03997	13.95	13.95
Conector USB hembra-hembra	2	0.03203	0.06407	3.95	7.90
Cable MicroUSB 80 cm	1	0.01000	0.01000	4.95	4.95
Cable MiniUSB 80 cm	1	0.01000	0.01000	4.95	4.95
Soporte interior de 90° de 3 vías con roscado M4	8	0.01040	0.08318	1.95	15.60
Separador hexagonal M3 macho x hembra 15 mm	18	0.00303	0.05461	0.40	7.20
Tornillo M2 x 14 mm	4	0.00039	0.00154	0.20	0.80
Tornillo M3 x 12 mm	18	0.00094	0.01685	0.07	1.26
Tornillo M3 x 18 mm	8	0.00120	0.00962	0.08	0.64
Tornillo M4 x 12 mm	32	0.00175	0.05600	0.10	3.20
Tuerca M2	4	0.00014	0.00056	0.20	0.80
Tuerca M3	36	0.00038	0.01368	0.06	2.16
Arandela M3	4	0.00008	0.00031	0.02	0.08
		Total (kg)	3.02	Total (\$)	158.47

Figura 8.16: Desglose del chasis. Pesos en kg y precios en \$USD

Entre los requerimientos del chasis se tiene que el costo combinado de la compra de materiales y su manufactura, debe ser idealmente inferior a \$USD 100. En esto se contempla únicamente el material de las paredes, la malla para el filtrado de polvo, y los mecanismos utilizados para unirlos, es decir, los tornillos, tuercas, y soportes interiores. Se tiene entonces que el costo total fue de \$USD 89.82, cumpliendo satisfactoriamente con el requerimiento.

Por otra parte, el chasis cuenta con un peso de aproximadamente 3.02 kg sin incluir el peso de la placa de desarrollo y la planta MCP, las cuales le suman aproximadamente 350 g. Por lo tanto, también se cumple con el peso ideal requerido inferior a 7 kg.

El proceso de ensamblaje también resulta sencillo, ya que solo necesita de apretar tornillos M2, M3 y M4, y hacer 8 orificios con una broca de 3 mm, por lo que se puede hacer con un kit de herramientas básico, como el presente en el laboratorio. La manufactura no requiere de personal técnico altamente capacitado, ya que el corte láser es realizado por estudiantes que laboran como operarios en el LAIMI. Igualmente, de ser necesario se pueden capacitar a investigadores del laboratorio para poder hacer el corte sin ayuda externa.

Capítulo 9

Integración de soluciones de SAS e IoT

Este capítulo tiene como objetivo integrar ambas plataformas de SAS diseñadas con la solución IoT y conducir pruebas que permitan validar el grado de cumplimiento de las necesidades establecidas.

9.1. Pruebas en plataforma de SAS #1

Para cada uno de los cuatro paneles solares comerciales para aplicaciones espaciales seleccionados, listados en la Tabla 5.1, se utilizó una fuente de poder programable cuya salida se conecta a una carga electrónica programable. La fuente de poder utilizada cuenta con dos canales que pueden suministrar un máximo de 30 V y 3 A. Dado que en algunos de los puntos de operación que se requieren evaluar se sobrepasa el límite de 3 A, se optó por colocar los dos canales en paralelo, teniendo así como límites 30 V y 6 A. Además, se utilizaron los cables de medición de la carga electrónica para medir con alta precisión la caída de tensión recibida de la fuente de poder.

Ambos equipos se conectaron a un ordenador mediante cables USB para que, posteriormente, a través de Python y la librería PyVISA, se puedan enviar comandos para controlar sus salidas y modos de operación. Asimismo, se utilizó la interfaz de control de la solución IoT para enviar de manera automatizada los diferentes parámetros a evaluar (modelo de panel solar, dimensiones del arreglo solar, condiciones externas, y el punto de operación de tensión), luego, el modelo analítico procesa dichos parámetros para calcular la corriente correspondiente, y se envían los valores a los instrumentos. Luego, el algoritmo de control desarrollado realiza las correcciones requeridas de acuerdo a la retroalimentación obtenida del voltímetro interno de la carga electrónica para regular las salidas hasta obtener porcentajes de error mínimos. Finalmente, se envían las mediciones y demás parámetros de interés a la base de datos de la solución IoT, de modo que esta se vaya actualizando en tiempo real y se puedan observar los cambios en la interfaz de visualización.

Cabe destacar que, si bien se requiere que un ordenador esté encendido y conectado a los instrumentos mientras ejecuta un programa de Python que recibe periódicamente la información de la interfaz de control, esta se puede acceder y operar desde un ordenador o dispositivo móvil en cualquier otra ubicación física, siempre y cuando tenga acceso a internet, así como la interfaz de visualización y la base de datos. De esta forma se obtiene la componente de control y monitoreo remoto deseada.

Para realizar estas pruebas, dado que se requerían evaluar miles de puntos de operación distintos, se desarrolló un programa adicional en Python que permite modificar de forma periódica los valores de entrada en la interfaz de control de acuerdo a las necesidades del usuario. Para esto, el usuario puede subir archivos .CSV que contengan los diferentes puntos de operación que se desean evaluar, en los cuales además puede especificar el modelo de panel, dimensiones del arreglo y las condiciones externas deseadas. Asimismo, puede definir cuánto tiempo desea que se mantenga cada punto de operación a la salida de los equipos. Cuando todos los puntos de operación fueron evaluados, el programa envía el comando correspondiente a la interfaz de control para que se apague el equipo. Este programa se implementó en una Jupyter Notebook en Google Collab, por lo que puede ser ejecutado tanto en el mismo ordenador que controla los instrumentos, como en otro ordenador o dispositivo móvil con acceso a internet en una ubicación física distinta, ya que viene incluido entre las herramientas de Google Workspace. El enlace a dicha Notebook se presenta en el Anexo B.

En la Figura 9.1, se muestra una fotografía de los instrumentos y el ordenador que los controla, mientras se ejecutan pruebas.

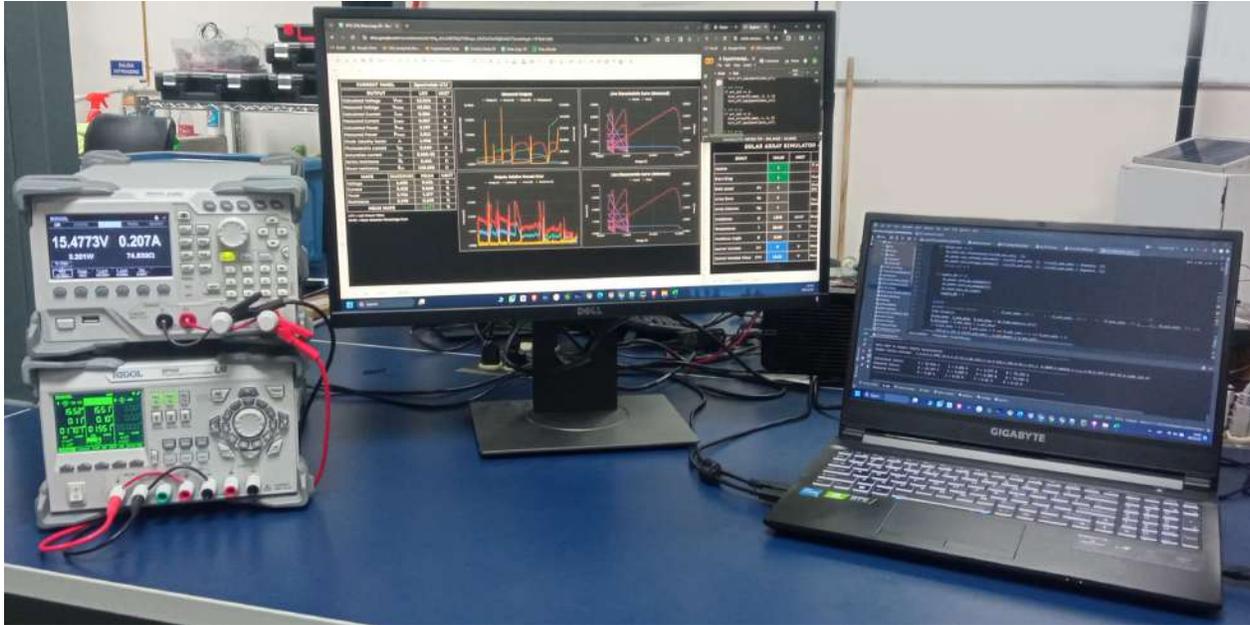


Figura 9.1: Instrumentos de laboratorio y ordenador ejecutando pruebas.

9.1.1. Pruebas de diferentes dimensiones de arreglos solares

Estas pruebas tienen como objetivo recrear las curvas características I-V y P-V de cada modelo de panel solar, evaluando alrededor de 60 puntos de operación de tensión y corriente. Este proceso se llevó a cabo para seis arreglos solares de diferentes dimensiones, los cuales se pueden apreciar de forma gráfica en la Figura 9.2.

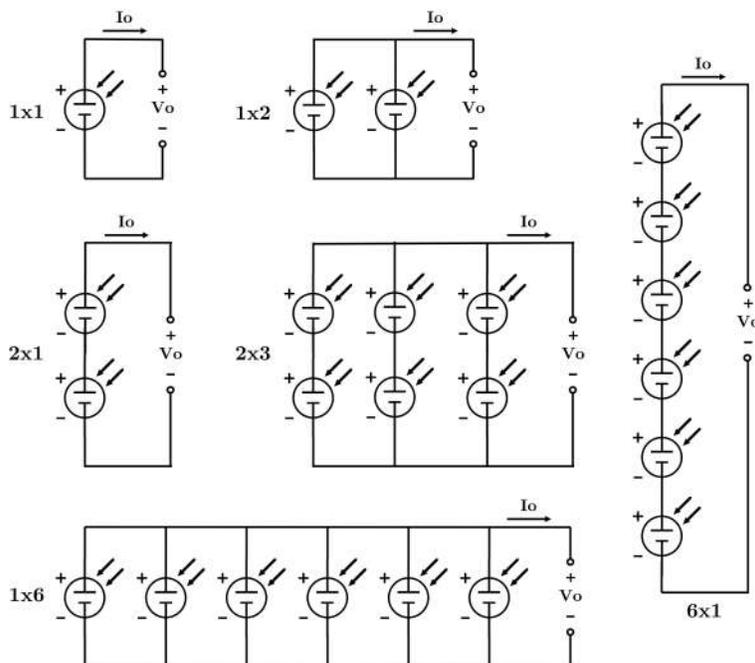


Figura 9.2: Dimensiones de arreglos solares evaluados.

Los parámetros de interés son la tensión, corriente y potencia proporcionados por la fuente de poder, además de la resistencia en la carga electrónica, por lo que para cada uno de estos se obtiene el ARPE en cada punto de operación, el cual permite comparar los valores de referencia calculados por el modelo analítico, con los medidos por la carga electrónica.

Dado que se evaluaron seis arreglos solares para cuatro modelos comerciales de paneles solares, se realizaron en total 24 pruebas. En la Figura 9.3, se presenta un ejemplo de las curvas características generadas, que en este caso corresponden a un arreglo 2x1 de paneles solares AZUR 3G30A. En esta se puede observar cómo las mediciones realizadas se aproximan de manera muy precisa al comportamiento esperado, indicado por las curvas punteadas de referencia, obteniendo ARPEs inferiores al 1% en la mayoría de la curva, tal y como se observa en la Figura 9.4. Se puede observar también que en los últimos puntos de la curva el ARPE aumenta rápidamente. No obstante, esto se puede atribuir a la problemática discutida en capítulos anteriores, donde conforme la corriente se aproxima a valores cercanos a cero, el MAPE (correspondiente al promedio de cada uno de los valores de ARPE a lo largo de la curva) se ve afectado por errores desproporcionados, a pesar de que sí se siga la tendencia esperada, como es el caso en la Figura 9.3. En el Anexo F, se presentan las curvas características de cada una de las pruebas realizadas, donde se discute con mayor detalle el comportamiento de cada una y se comparan los resultados obtenidos entre los cuatro paneles, para cada dimensión de arreglo solar estudiada.

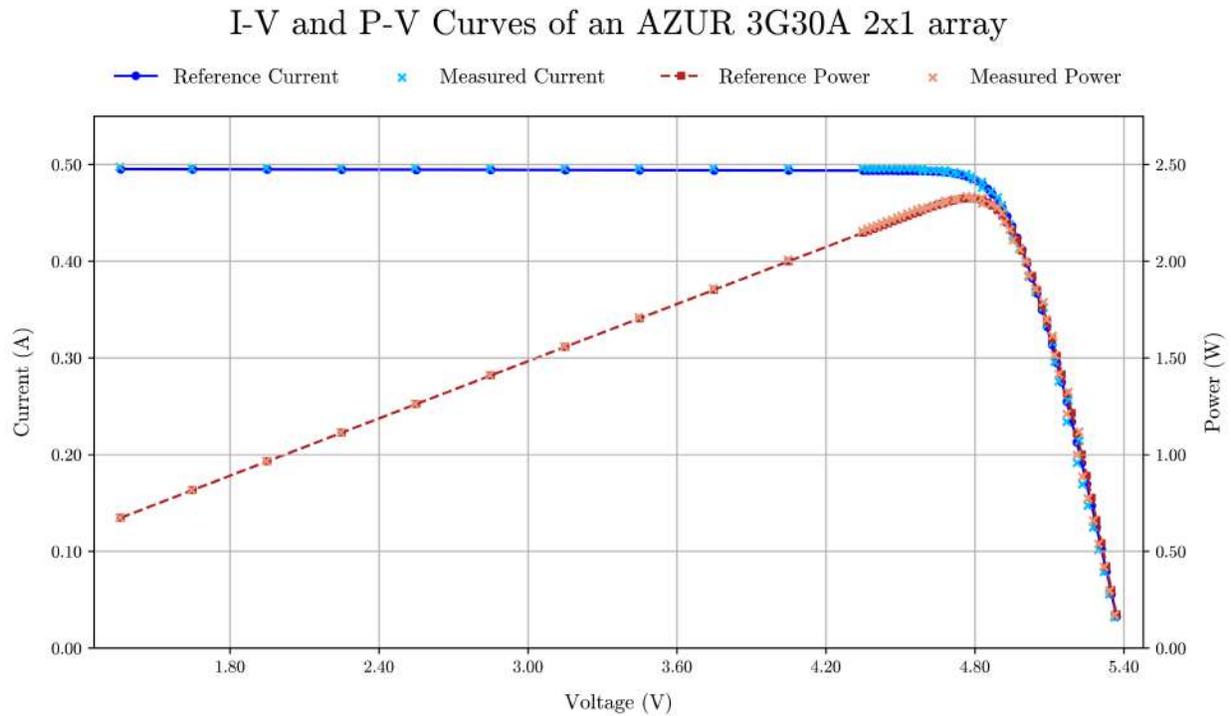


Figura 9.3: Curvas I-V y P-V de un arreglo 2x1 (AZUR 3G30A).

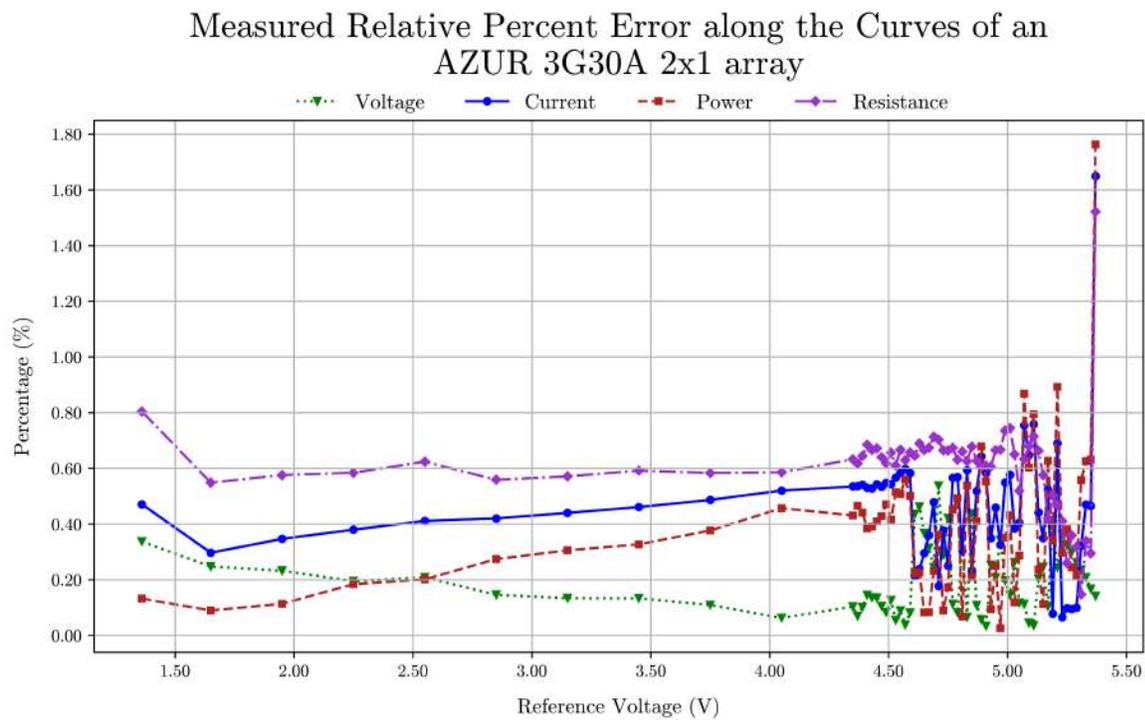


Figura 9.4: Porcentajes de error en curvas características de un arreglo 2x1 (AZUR 3G30A).

A continuación, en la Tabla 9.1 se presenta el MAPE obtenido en cada prueba, correspondiente al promedio del ARPE de cada punto de operación estudiado. También se calcula el MAPE promedio para cada panel solar y dimensión de arreglo solar, y así como el promedio de estos para obtener el MAPE global a lo largo de todas las pruebas. El MAPE máximo permitido en estas pruebas es de 5 %, según se detalló en las métricas de la Tabla 3.6, por lo que se puede observar que en el 100 % de las pruebas se cumplió con este requerimiento con un amplio margen, siendo el máximo presentado de 0.5970 %, y el promedio de 0.4767 %. Esto permite concluir que tanto el modelo analítico, como el programa desarrollado para controlar los instrumentos del laboratorio fueron exitosos.

Tabla 9.1: Resultados de MAPE en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Arreglo	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ	Promedio
1x1	0.5970	0.4730	0.5220	0.4760	0.5170
1x2	0.5060	0.4460	0.4470	0.4890	0.4720
1x6	0.5830	0.5860	0.5530	0.5650	0.5718
2x1	0.4140	0.4450	0.4660	0.4350	0.4400
2x3	0.5470	0.5100	0.5090	0.2540	0.4550
6x1	0.3900	0.4220	0.4040	0.4010	0.4043
Promedio	0.5062	0.4803	0.4835	0.4367	0.4767

Por otra parte, a partir de los promedios calculados para cada dimensión de arreglo solar, se puede apreciar que el MAPE más alto se presenta cuando se tienen seis paneles en paralelo, lo que hace que se presenten valores de corriente muy altos respecto los de tensión. En contraste, el MAPE más bajo ocurre cuando se tienen seis paneles en serie, lo que se traduce en valores de tensión muy altos respecto a los de corriente. Por lo tanto, se puede concluir que los controladores internos de la fuente de poder utilizada presentan mayores problemas de estabilidad cuando se tienen corrientes muy elevadas respecto a la tensión.

No obstante, cabe destacar que la práctica más común en CubeSats es colocar dos paneles en serie por cara, y luego conectar los arreglos de cada cara en paralelo, es decir, es un arreglo 2x3. Se puede observar que el MAPE presentado en este caso, igual a 0.4550 %, es inferior al MAPE global obtenido. Por lo tanto, se puede concluir que para fines de misiones CubeSats reales, donde no es común utilizar arreglos donde más arreglos en serie que en paralelo, el rendimiento de esta plataforma es bastante óptimo.

Adicionalmente, en las Tablas 9.2, 9.3, 9.4 y 9.5, se evalúan los porcentajes de error relativo máximos encontrados en cada una de las pruebas en términos de tensión, corriente, potencia y resistencia, respectivamente. En cada una de las tablas se resalta el mayor ARPE encontrado. Se puede observar entonces, que el máximo porcentaje de error relativo encontrado a lo largo de todas las pruebas fue de 2.1251 %, correspondiente a la potencia. Se puede observar entonces, que incluso el máximo porcentaje de error encontrado en las 24 pruebas realizadas sigue estando considerablemente por abajo del 5 % permitido.

Tabla 9.2: Resultados de ARPE máximo de tensión en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Arreglo	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
1x1	1.1427	0.8066	0.8185	0.8312
1x2	1.2302	0.7827	0.7346	0.8021
1x6	0.9782	0.9788	0.9441	0.9296
2x1	0.5374	0.8247	0.5733	0.5542
2x3	0.8582	0.8537	0.8699	0.6801
6x1	0.7501	0.7424	0.7225	0.7607

Tabla 9.3: Resultados de ARPE máximo de corriente en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Arreglo	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
1x1	1.5423	1.8659	0.8974	0.8456
1x2	1.5910	0.8434	0.8120	0.8723
1x6	0.7096	0.4947	0.8649	0.8118
2x1	1.6491	1.1851	0.8515	0.7881
2x3	0.5690	0.6675	0.7683	0.9662
6x1	1.0914	1.9215	0.8494	0.5930

Tabla 9.4: Resultados de ARPE máximo de potencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Arreglo	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
1x1	1.8991	2.1251	1.1749	1.1738
1x2	1.9532	0.9401	1.0585	1.2378
1x6	1.0271	1.0371	1.0104	1.0783
2x1	1.7628	1.2746	0.9228	0.9209
2x3	0.8765	0.9077	0.8882	1.0118
6x1	1.1568	1.9754	0.9064	1.0372

Tabla 9.5: Resultados de ARPE máximo de resistencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Arreglo	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
1x1	1.2256	1.5419	0.8247	0.8152
1x2	1.1429	0.8703	0.8866	0.9126
1x6	1.1365	1.0165	1.0111	1.1106
2x1	1.5217	1.1071	0.8392	0.8128
2x3	0.9713	0.8928	0.8675	1.1575
6x1	1.3061	1.8840	0.8656	0.7905

9.1.2. Pruebas de diferentes condiciones externas

Los CubeSats modernos por lo general tienen subsistemas que garantizan que el microsatélite se oriente de forma tal que se obtenga la máxima captación de potencia a través de sus arreglos solares. Estos se conocen como seguidores del punto de máxima potencia (MPPT).

Dado que se desea poder observar de forma evidente cómo las condiciones externas afectan el rendimiento de los arreglos solares, estas pruebas se enfocan en evaluar siempre el punto de máxima potencia del arreglo solar, el cual varía en función de la irradiancia, temperatura y ángulo de incidencia. Se realizan entonces tres pruebas diferentes por cada modelo comercial de panel solar, resultando en 12 pruebas en total. En cada una de estas, una de las condiciones externas cambia a lo largo de su respectivo rango de operación, los cuales fueron establecidos en las métricas del Capítulo 3.

Al igual que en las pruebas de diferentes dimensiones de arreglos solares, se monitorean cuatro parámetros a partir de los sensores internos de la carga electrónica, y se calculan los ARPEs en cada punto de operación evaluado. En las Figuras 9.5, 9.6 y 9.7, se presentan ejemplos de curvas que muestran el cambio de los parámetros en función de la irradiancia, temperatura y ángulo de incidencia, respectivamente. El objetivo de estas pruebas es validar que los instrumentos del laboratorio, en conjunto con el programa de control desarrollado, sean capaces de emular de manera precisa el comportamiento de un arreglo solar real cuando es sometido a los diferentes rangos de condiciones externas establecidos. Por lo tanto, estas gráficas permiten observar de manera visual qué tanto se apegan las mediciones a los valores de referencia esperados. En el Anexo F, se presentan y se discuten el conjunto de curvas generado para los cuatro modelos de paneles solares evaluados, incluyendo también las curvas de ARPE relativo asociadas.

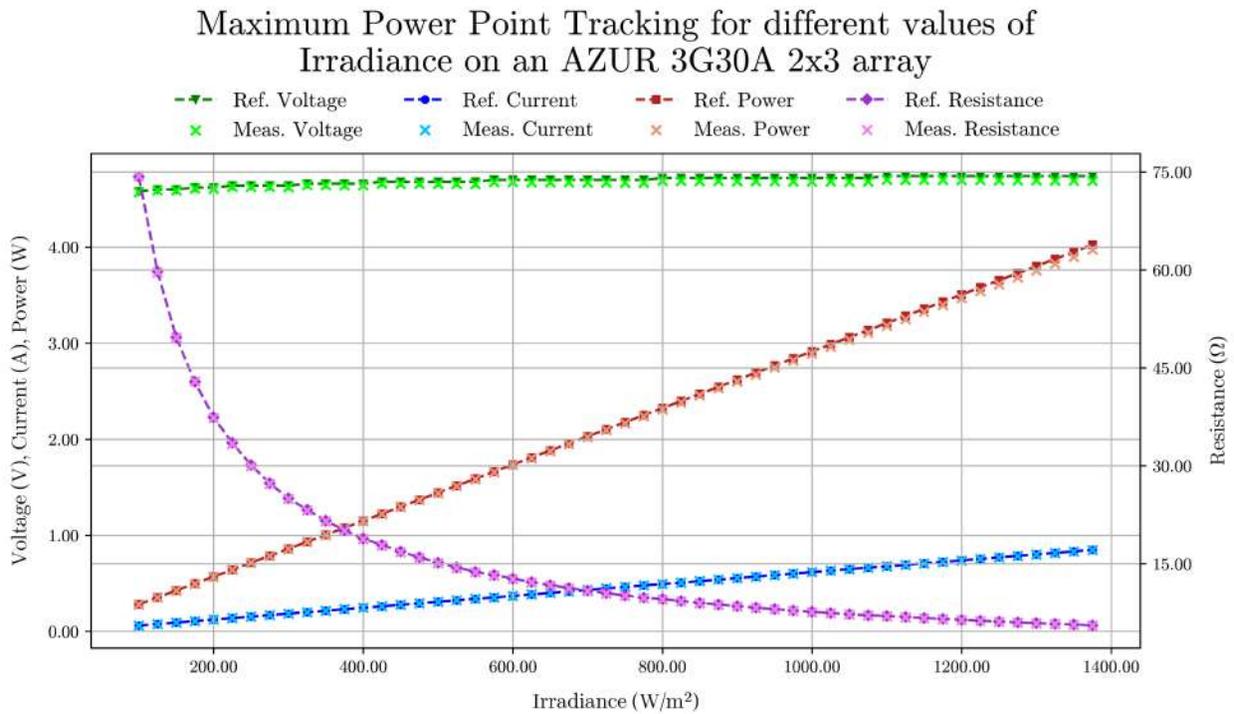


Figura 9.5: MPPT a diferentes niveles de irradiancia (AZUR 3G30A).

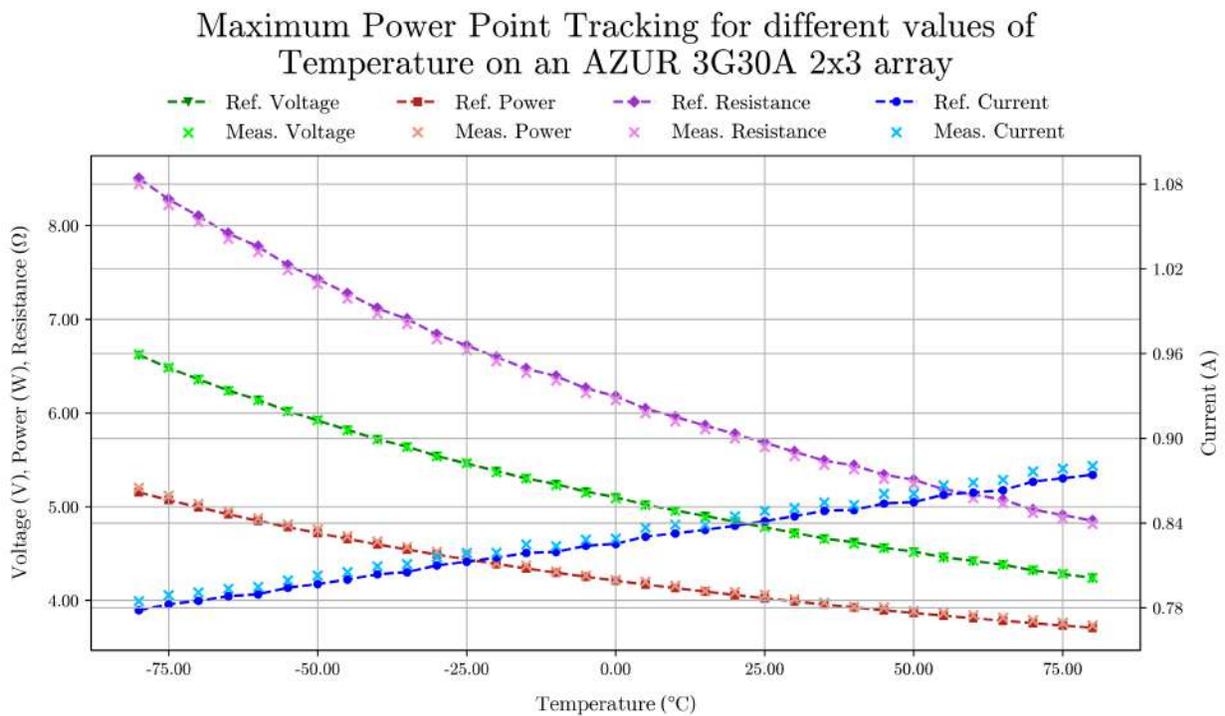


Figura 9.6: MPPT a diferentes niveles de temperatura (AZUR 3G30A).

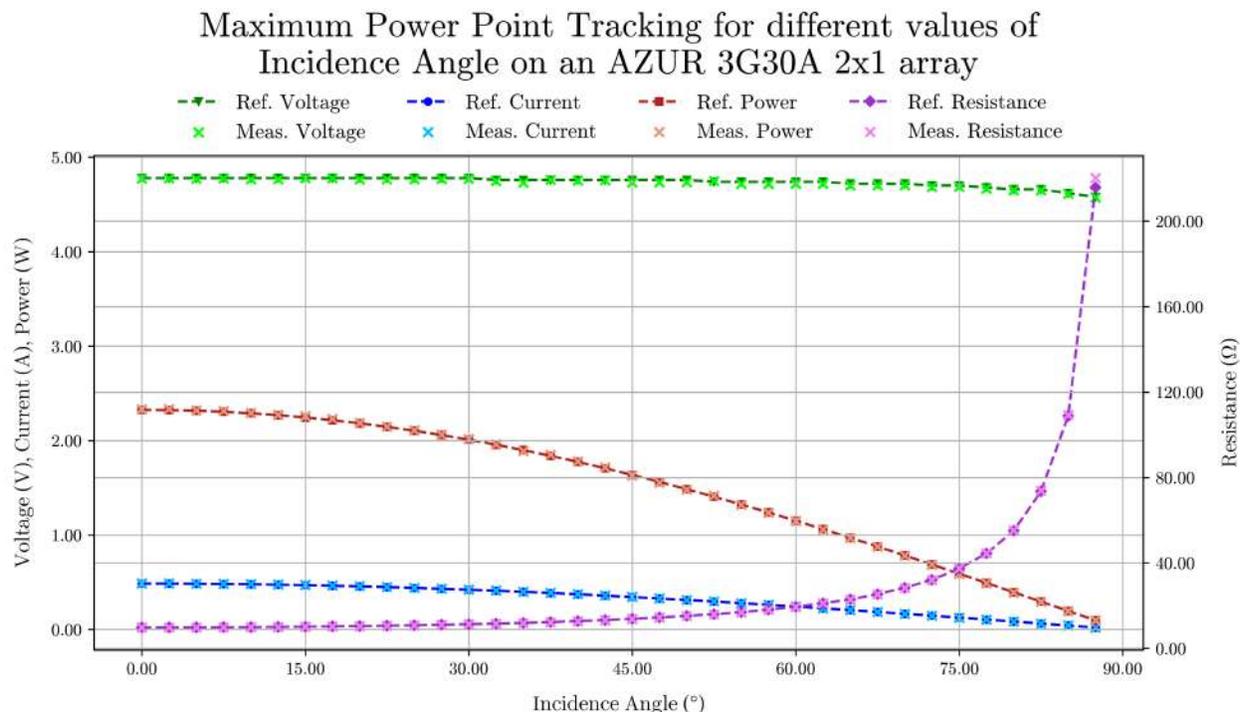


Figura 9.7: MPPT a diferentes ángulos de incidencia. (AZUR 3G30A).

A continuación, en la Tabla 9.6 se presenta el MAPE obtenido en cada prueba, así como el MAPE promedio para cada modelo de panel solar y condición externa variable, y el MAPE global a lo largo de todas las pruebas, el cual corresponde a 0.4645 %. Nuevamente se puede observar que en todas las pruebas se logra obtener un MAPE considerablemente menor al permitido, de 5%.

Tabla 9.6: Resultados de MAPE en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales

Cond. variable	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ	Promedio
Irradiancia	0.4910	0.4640	0.4950	0.4960	0.4865
Temperatura	0.5510	0.6000	0.4920	0.4340	0.5193
Áng. de inc.	0.4250	0.4470	0.4580	0.2210	0.3878
Promedio	0.4890	0.5037	0.4817	0.3837	0.4645

Por otra parte, los MAPE más altos que se presentaron fue durante la prueba del rango de temperatura. Tal y como se observa en la Figura 9.6, la particularidad de esta prueba respecto a las otras, es que la tensión disminuye y la corriente aumenta conforme se sigue aumentando la temperatura, mientras que en las pruebas de irradiancia y ángulo de incidencia, la tensión siempre se mantiene en valores elevados. Por lo tanto, nuevamente se evidencia cómo los controladores internos de la fuente de poder presentan mayores problemas en mantenerse estables en el punto de operación indicado cuando la tensión es baja relativa a la corriente.

Adicionalmente, en las Tablas 9.7, 9.8, 9.9 y 9.10, de nuevo se evalúan los ARPEs máximos encontrados en cada una de las pruebas en términos de tensión, corriente, potencia y resistencia, respectivamente, destacando el valor máximo encontrado en cada caso. Se puede observar entonces, que el máximo porcentaje de error relativo encontrado a lo largo de todas las pruebas fue de 2.0380 %, correspondiente a la resistencia de la carga. Por lo tanto, una vez más se comprueba que incluso el punto de operación con mayor porcentaje de error encontrado se encuentre considerablemente por abajo del 5 % permitido.

Tabla 9.7: Resultados de porcentaje de error relativo máximo de tensión en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales

Condición variable	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Irradiancia	1.0300	0.8680	0.8610	0.8610
Temperatura	0.3280	0.8930	0.4530	0.4870
Ángulo de incidencia	0.6180	0.5020	0.6430	0.7040

Tabla 9.8: Resultados de porcentaje de error relativo máximo de corriente en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales

Condición variable	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Irradiancia	0.3620	0.8880	0.8620	0.8990
Temperatura	0.8760	0.8590	0.8490	0.5550
Ángulo de incidencia	1.5110	1.7950	1.7990	1.2910

Tabla 9.9: Resultados de porcentaje de error relativo máximo de potencia en distintas dimensiones de arreglos solares para distintos modelos de paneles solares comerciales

Condición variable	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Irradiancia	1.2760	0.9850	1.0340	1.0740
Temperatura	0.9380	1.0750	0.9570	0.3140
Ángulo de incidencia	1.6070	1.9090	1.9610	1.0400

Tabla 9.10: Resultados de porcentaje de error relativo máximo de resistencia en pruebas de condiciones externas variables para distintos modelos de paneles solares comerciales

Condición variable	AZUR 3G30A	AZUR 4G32C	CESI CTJ30	Spectrolab UTJ
Irradiancia	0.8230	0.7900	0.8120	0.8280
Temperatura	0.8240	0.8240	0.8280	0.8380
Ángulo de incidencia	2.0380	1.7110	1.6620	1.5190

A partir de las 36 pruebas conducidas en la plataforma de SAS #1, se tiene entonces que el MAPE global obtenido es de 0.4706 %, el máximo MAPE obtenido en una prueba fue de 0.6 %, y el ARPE máximo obtenido fue de 2.1251 %. Por lo tanto, se concluye que en el 100 % de las pruebas y de los puntos de operación evaluados, se cumplió con un amplio margen el rendimiento esperado de obtener porcentajes de error inferiores al 5 %.

De este modo se logró validar entonces que el uso de los instrumentos de laboratorio con el programa de control desarrollado es un medio óptimo para emular el comportamiento de arreglos solares de CubeSats de 1U, pudiendo replicar de manera adecuada las curvas características de distintas dimensiones de arreglos, así como de hacer el seguimiento del MPP en los rangos de irradiancia, temperatura y ángulo de incidencia esperados en misiones reales de CubeSats en el espacio.

Por otra parte, el tiempo total que tomó realizar estas 36 pruebas fue 23 horas, 14 minutos y 13 segundos, transmitiendo 9814 registros. En esto no se cuentan las decenas de pruebas que fueron descartadas por mejoras que se hicieron al programa de control, pero se estima que en total se dedicaron al menos 70 horas en total realizando pruebas en esta plataforma. Adicionalmente, en el caso del panel solar AZUR 3G30A, todas las pruebas de dimensiones de arreglos solares fueron tomadas en un solo día, consecutivamente, tomando en total un aproximado de 7 horas, y en otra ocasión, con el panel solar Spectrolab UTJ, también se realizaron pruebas consecutivas por 5 horas con 16 minutos. Los resultados crudos de cada prueba se encuentran en el enlace de Google Drive presentado en el Anexo B. A lo largo de aproximadamente 8 semanas, utilizando la plataforma 5 días por semana, por al menos 8 horas diarias, no se presentaron problemas de interrupción relacionados con fallas al contactar el servidor de la solución IoT. Asimismo, en algunos casos se dejó la plataforma operando a lo largo de la noche mientras que se monitoreaba su comportamiento a través del panel de visualización o se hacían modificaciones en el panel de control estando a kilómetros de distancia del laboratorio, bajo otra red WiFi, e incluso utilizando dispositivos móviles, puesto que las hojas de cálculo pueden ser abiertas y modificadas con la aplicación de *Google Drive* en un teléfono inteligente.

Por lo tanto, este conjunto de pruebas también permitieron comprobar la validez de cada componente de la solución IoT al hacer pruebas reales a lo largo de periodos prolongados iguales o superiores a la métrica ideal de 8 horas diarias, permitiendo controlar remotamente la plataforma desde una ubicación geográfica y red de Internet distinta, y permitiendo monitorear en tiempo real el comportamiento de esta a través del panel de visualización. Asimismo, la base de datos fue capaz de recibir como mínimo 20000 registros entre las 8 semanas de uso que se le dio sin presentar ningún tipo de limitaciones, lo cual no hubiera sido posible con otras de las plataformas IoT, como por ejemplo la de Azure IoT Hub debido a su límite de 8000 mensajes mensuales.

9.1.3. Prueba de tiempos de refrescamiento

La prueba consistió en enviar un mismo punto de operación aleatorio hacia la plataforma IoT, por un tiempo total de 19 horas, 20 minutos y 12 segundos consecutivos, enviando un total de 13359 registros. Se envía un mismo punto con el objetivo de que el tiempo que le tome al modelo analítico calcular nuevos puntos de operación no afecte el tiempo de refrescamiento en la base de datos. Asimismo, en las especificaciones se tiene como valor ideal el poder conducir pruebas por al menos 8 horas diarias. De esta forma, la prueba también permitió validar que la plataforma es capaz de funcionar de manera consecutiva por más del doble del tiempo esperado sin presentar ningún tipo de problema.

A continuación, en la Tabla 9.11, se presentan los resultados obtenidos en cuanto al tiempo de refrescamiento. Cabe recordar que el valor ideal para este parámetro es de 0.5 s, y el marginal de 10 s, mientras que el grado de importancia de estas especificaciones es equivalente a 3, siendo 5 el valor máximo de importancia.

Tabla 9.11: Tiempo de refrescamiento en la solución IoT con la plataforma de SAS #1.

Métrica	Tiempo (s)
Promedio	5.2195
Desviación estándar	1.8943
Mínimo	3
Máximo	59

Por lo tanto, con un tiempo de refrescamiento de 5.2195 ± 1.8943 segundos, se puede observar que no se cumple con el valor ideal esperado, pero sí con el marginal. Como se discutió en el Capítulo 7, el hecho de que la plataforma IoT únicamente soportara el protocolo HTTP y no el MQTT, iba a tener un impacto notable en el tiempo de refrescamiento. Asimismo, se puede observar que el tiempo máximo que se registro fue de 59 segundos, lo cual, dado que es mucho más alto que el promedio, puede atribuirse a problemas relacionados con la red WiFi o una sobrecarga inusual en los servidores de Google Workspace. No obstante, dado que tiempo promedio cumple con el valor marginal, y el grado de importancia asignado a la especificación era no crítico, se puede concluir que la solución IoT implementada a la hora de integrarse con la plataforma de SAS #2 cumple con el requerimiento.

Nuevamente, los datos crudos de esta prueba se encuentran en el enlace de Google Drive presentado en el Anexo B. En este también se presentan los datos recompilados en el archivo .CSV que funciona como base de datos local. Debido a la forma que fue programada la función para enviar los datos a la plataforma IoT, el programa espera a que la plataforma confirme si llegó o no el mensaje, por lo que durante ese tiempo no se ejecuta ninguna otra acción dentro del programa. Por lo tanto, los tiempos de refrescamiento en dicho archivo .CSV coinciden con los de la base de datos en la nube. No obstante, si se desea puede omitirse ese tiempo de espera, resultando en que la base de datos local se actualice más frecuentemente que la de la nube.

9.2. Pruebas en plataforma de SAS #2

Como se ha descrito en capítulos anteriores, se confeccionó una placa de desarrollo que contiene los microcontroladores y demás componentes electrónicos seleccionados durante la fase de diseño presentada en el Capítulo 4. El flujo de trabajo planteado es que el MCU_M le envíe a la planta MCP el punto de operación deseado para que esta regule las salidas del convertidor DC-DC, utilizando los controladores digitales diseñados, y luego envíe de regreso al MCU_M los valores de tensión y corriente medidos por sus sensores internos. No obstante, dado que por factores externos no fue posible utilizar esta planta durante el desarrollo del proyecto. Sin embargo, todas las otras componentes de la solución ya se encontraban implementadas en un prototipo funcional, porque lo que es posible evaluar el rendimiento del modelo analítico, así como la intercomunicación entre los distintos microcontroladores y la solución IoT.

Es por esto que, en vez de conectar la planta MCP, se utilizó un microcontrolador Arduino UNO, el cual recibe el punto de operación proveniente del MCU_M y le envía de regreso el mismo punto de operación pero añadiéndole ruido que varía de manera aleatoria periódicamente. De esta forma, es posible emular el comportamiento esperado de la planta MCP desde el punto de vista del MCU_M, y así poder evaluar el funcionamiento de las demás componentes. Cabe destacar que, dado que este microcontrolador adicional solo funciona como un reemplazo temporal de la planta MCP y no forma parte del prototipo funcional real del proyecto IPTC, es irrelevante hacer una matriz de selección, ya que no es parte del diseño. En la Figura 9.8, nuevamente se presenta una fotografía del prototipo funcional utilizado para estas pruebas.

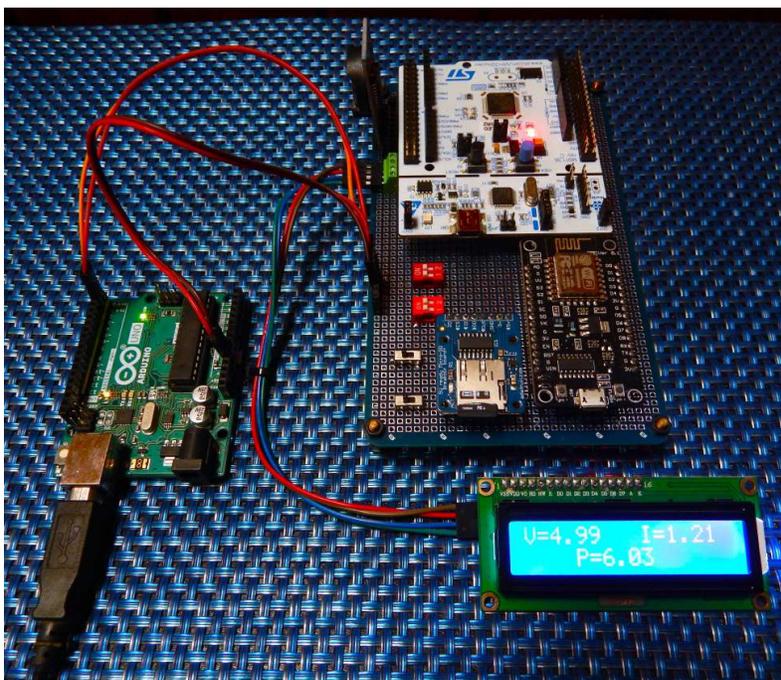


Figura 9.8: Placa de desarrollo confeccionada.

9.2.1. Prueba para la creación de curvas características

Esta prueba tiene como objetivo validar la correcta implementación de la plataforma de SAS #2 para comunicarse con la plataforma IoT y recibir distintos puntos de operación y parámetros del panel solar desde la interfaz de control, procesar el modelo analítico, y enviar los registros a la base de datos. La manera de validar el funcionamiento, es revisando que en la base de datos existan registros asociados a cada uno de los puntos de operación que se enviaron, y que las lecturas de los sensores tengan ARPEs entre 0% y 1%, ya que así fue programado el MCU_S1.

Al cabo de 2 horas, con 28 minutos y 20 segundos, se recibieron 2018 registros, y se verificó que entre ellos se incluyeran los 407 puntos de operación diferentes que fueron enviados. Estos puntos permitieron recrear las curvas características para las diferentes dimensiones de arreglos solares descritas en la Figura 9.2 para el panel solar AZUR 3G30A. En la Figura 9.9, se muestran las gráficas generadas en el panel de visualización IoT, permitiendo verificar exitosamente que se cumple con todos los objetivos de la prueba, puesto que la gráfica de los ARPEs permite apreciar que los valores oscilaban en el rango esperado, y las curvas características fueron replicadas perfectamente. Los datos crudos se encuentran en el enlace de Google Drive presentado en el Anexo B.

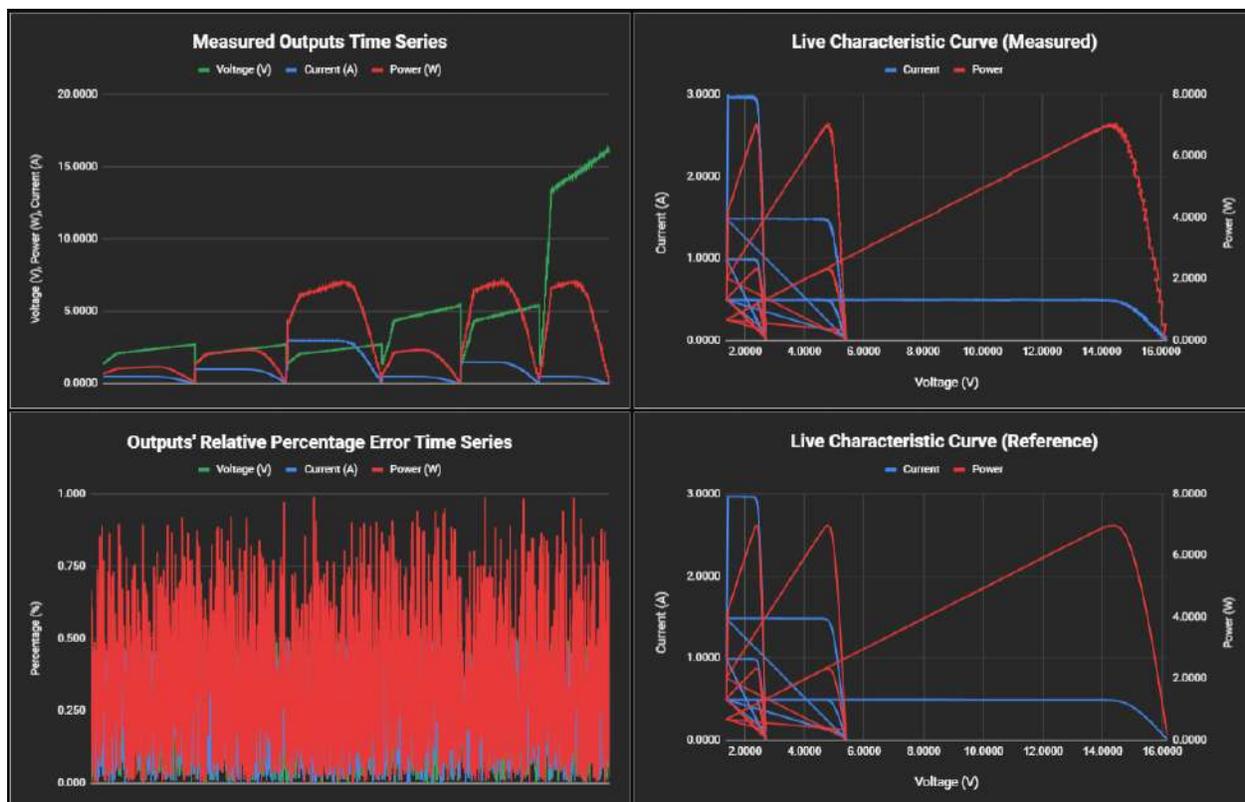


Figura 9.9: Recreación de curvas usando la plataforma de SAS #2 vistas desde el panel de visualización de la plataforma IoT.

9.2.2. Prueba de tiempos de refrescamiento

Se realizó exactamente la misma prueba realizada en la plataforma de SAS #1, de modo que nuevamente se envió un mismo punto de operación aleatorio hacia la plataforma IoT. En esta ocasión, la prueba se extendió por un tiempo total de 16 horas, 12 minutos y 1 segundos consecutivos, enviando un total de 14325 registros. Nuevamente, la prueba también permitió validar que la plataforma es capaz de funcionar de manera consecutiva por el doble del tiempo ideal esperado sin presentar problemas.

A continuación, en la Tabla 9.12, se presentan los resultados obtenidos en cuanto al tiempo de refrescamiento.

Tabla 9.12: Tiempo de refrescamiento en la solución IoT con la plataforma de SAS #2.

Métrica	Tiempo (s)
Promedio	4.023
Desviación estándar	2.040
Mínimo	1
Máximo	17

En este caso, con un tiempo de refrescamiento de 4.023 ± 2.040 segundos, se puede observar una reducción de aproximadamente 1 segundo en el tiempo de refrescamiento respecto a la otra plataforma de SAS. Esto puede atribuirse a que el microcontrolador ESP8266 se enfoca únicamente en establecer comunicación con la plataforma IoT, mientras que en la plataforma de SAS#1, dado que se utiliza un computador con sistema operativo de Windows, es posible que existan procesos secundarios ocultos que previenen que el módulo WiFi se enfoque únicamente en ejecutar el programa.

A pesar de esta reducción, la plataforma IoT sigue sin cumplir con el valor ideal esperado, pero sí con el marginal con un margen más amplio. Nuevamente, dado que el grado de importancia asignado a la especificación era no crítico, se puede concluir que la solución IoT implementada a la hora de integrarse con la plataforma de SAS #2 cumple con el requerimiento.

Por otra parte, en este caso la forma en que se programó el MCU omite el tiempo de espera para confirmar si se recibe o no el mensaje, resultando en que la base de datos local, que en este caso consta de una memoria MicroSD, se actualiza más frecuentemente que la base de datos en la nube. A continuación, se listan los parámetros obtenidos en la Tabla 9.13.

Tabla 9.13: Tiempo de refrescamiento en la memoria Micro SD de la plataforma de SAS #2.

Métrica	Tiempo (s)
Promedio	1.2153
Desviación estándar	0.4110
Mínimo	1
Máximo	2

Se puede observar que, con 1.2153 ± 0.4110 segundos, el tiempo de refrescamiento obtenido fue considerablemente menor que el de la base de datos en la nube, puesto que no estaba limitado por el tiempo que la plataforma IoT toma en procesar las solicitudes HTTP. Por lo tanto, además de validar que los datos se guardan adecuadamente, se puede concluir que en caso de que sea necesario obtener registros más frecuentes que los que ofrece la plataforma IoT, extraer los datos de la memoria Micro SD es una alternativa óptima.

Capítulo 10

Análisis económico

Para determinar la validez de la solución diseñada, se debe analizar su rendimiento a nivel económico. En este capítulo se hace un análisis de los recursos invertidos durante el desarrollo del proyecto y los principales beneficios que este proporciona. Cabe destacar que el proyecto fue primordialmente concebido para su uso durante el periodo de pruebas de misiones CubeSat con fines académicos o de investigación, por lo que el análisis se centra en los beneficios que brinda la herramienta en centros de educación superior, como es el caso de SETEC-Lab y Delta-Lab en el Tecnológico de Costa Rica. Se complementa el análisis con las estimaciones del costo de las plataformas de SAS #1 y #2 propuestas en este proyecto, donde esta última se trata de una eventual implementación física del chasis y la placa de desarrollo diseñados en este proyecto, junto con una planta MCP funcional, y se realiza una comparación con plataformas de SAS comerciales utilizadas en la industria y en centros de investigación.

En la Tabla 10.1, se presentan las licencias de software que fueron utilizadas durante el desarrollo del proyecto. Cabe destacar que tanto JetBrains como Autodesk ofrecen licencias estudiantiles completamente gratuitas con fines meramente académicos. Por otra parte, la licencia académica de MATLAB que se utilizó pertenece al laboratorio donde se desarrolló el proyecto, por lo que también es ampliamente utilizada por otros investigadores. En cuanto a Adobe Creative Cloud (la herramienta utilizada fue Adobe Photoshop), la licencia utilizada fue accedida desde una computadora del LAIMI 2 del Tecnológico de Costa Rica, por lo que también puede ser utilizada por cualquier estudiante de la institución. No obstante, se considera la tarifa de la licencia estudiantil de Adobe Creative Cloud por 1 mes, correspondiente al periodo que fue necesario utilizarla. Si bien se utilizaron otras herramientas de software, como Arduino IDE y herramientas de Google Workspace, el uso de estas no tiene ningún costo monetario.

Tabla 10.1: Costo de licencias de software utilizadas (en \$USD).

Licencia de software	Costo
Licencia estudiantil de JetBrains PyCharm Community Edition	0
Licencia estudiantil de Autodesk Fusion 360	0
Licencia estudiantil de Autodesk AutoCAD	0
Licencia académica anual de MathWorks MATLAB/Simulink	275
Licencia estudiantil mensual de Adobe Creative Cloud	19.99
Total	294.99

En cuanto a elementos de hardware que se utilizaron durante el desarrollo del proyecto, estas se listan en la Tabla 10.2. Se trata de los instrumentos del laboratorio utilizados de la marca RIGOL, un monitor marca DELL que se encuentra en el laboratorio, y un computador portátil personal marca Gigabyte. Se presenta además la vida útil estimada de cada elemento, el tiempo de uso que llevan, el costo de compra, y el costo residual, el cual fue calculado asumiendo que al final de su vida útil, cada elemento se deprecia en un 70 %.

Tabla 10.2: Valores de depreciación de elementos de hardware utilizados (en \$USD).

Componente de hardware	Años útiles	Años de uso	Costo de compra	Costo residual
Computadora personal	7	2	1000	800
Monitor secundario	10	5	90	58.5
Fuente de poder programable RIGOL DP832	10	5	709.5	461.175
Carga electrónica programable RIGOL DL3021	10	5	748.5	486.525

En la Tabla 10.3, se estima el costo del desarrollo del proyecto. Se puede observar que el mayor recurso invertido en este proyecto fue el tiempo de desarrollo. El estudiante dedicó aproximadamente 40 horas semanales para la ejecución del proyecto, y dado que se le concedió una prórroga por factores externos que retrasaron el desarrollo del mismo, en total el periodo de desarrollo fue de aproximadamente 7 meses. El pago por hora en asistencias especiales en el Tecnológico de Costa Rica es de aproximadamente \$2.00. No obstante, el máximo de horas que se pueden nombrar por semana son 20, y el nombramiento fue realizado por un total de 20 semanas. También se toma en cuenta la depreciación de los componentes de hardware que se dio durante los 7 meses, y que la licencia de Adobe y Mathworks fueron utilizadas por 1 y 2 meses, respectivamente.

Tabla 10.3: Costo de desarrollo del proyecto (en \$USD).

Detalle	Costo
Licencias de software	65.82
Depreciación de hardware	121.54
Pago de asistencia especial	800.00
Total	987.37

En la Tabla 10.4, se desglosa el costo total de la plataforma de SAS #1, tomando en cuenta el costo de los instrumentos nuevos, y una Raspberry Pi 4 con un monitor y periféricos, que es la alternativa más económica a la computadora portátil personal utilizada, pero que perfectamente cumple con la función que está desarrollando para el funcionamiento de la plataforma.

Tabla 10.4: Costo total de la plataforma SAS #1 (en \$USD).

Componente	Costo
Fuente de poder programable RIGOL DP832	709.5
Carga electrónica programable RIGOL DL3021	748.5
Raspberry Pi 4	100
Monitor y periféricos	100
Total	1658

En la Tabla 10.5, se desglosa el costo total de la plataforma de SAS #2, en el cual se utiliza el costo estimado que tendrá la planta MCP que en este momento se encuentra en desarrollo. Cabe destacar que la solución IoT que controla a la plataforma puede ser accedida desde un teléfono inteligente de gama baja, por lo que su precio aproximado también se toma en cuenta en el desglose.

Tabla 10.5: Costo total de la plataforma SAS #2 (en \$USD).

Componente	Costo
Planta MCP	300
Placa de desarrollo	84.68
Chasis	158.47
Teléfono inteligente	160
Total	703.15

En la Tabla 10.6, se presenta una comparación de las plataformas SAS desarrolladas con opciones comerciales disponibles para aplicaciones espaciales [39], [38]. Se puede notar que estas alternativas están diseñadas para simular arreglos solares mucho más grandes que los presentes en CubeSats de 1U o 3U, que son contemplados para el proyecto IPTC.

Tabla 10.6: Comparación de plataformas con SAS comerciales para aplicaciones espaciales.

Especificación	Keysight MP4300A	AMETEK Elgar	Plataforma 1	Plataforma 2
Potencia máxima(W)	1000	1000	90	60
Tensión máxima (V)	60	200	30	24
Corriente máxima (A)	10	15	6	4
Error en puntos de operación (%)	0.05	1	0.47	No definido
Precio (\$USD)	10000 - 15000	10000 - 15000	1658.00	703.15

A partir de estos datos, se puede concluir que el proyecto cumplió con el objetivo de diseñar una plataforma que permita emular el comportamiento de los arreglos solares de CubeSats de 1U con alta precisión y sin tener que incurrir en gastos exorbitantes.

En el caso de la plataforma de SAS #1, se utilizó equipo ya disponible en el laboratorio, que además es ampliamente utilizado en otros proyectos de investigación, por lo que la inversión en nuevos componentes de hardware fue nula. A pesar de esto, se logró obtener una precisión altamente competitiva, ya que su porcentaje de error medio es incluso menor que el especificado por AMETEK. Cabe destacar que, dado que todos los recursos creados durante el desarrollo de este proyecto están destinados a ser de código abierto, esto permite a otras universidades, centros de investigación o entusiastas de microsátélites el acceso a una herramienta de pruebas mucho más accesible y que probablemente también puedan utilizar sin requerir de una inversión de capital, puesto que este tipo de fuentes de poder y cargas electrónicas programables son de uso común, y suelen ser compatibles con la librería de PyVISA utilizada. No obstante, incluso si requirieran comprar el equipo, la inversión sería hasta 9 veces menor que la que harían con una opción de SAS comercial.

En el caso de la plataforma de SAS #2, a pesar de que la placa MCP aún se encuentra en desarrollo y por ende no es posible validar el rendimiento final del sistema, el presente proyecto de graduación logró desarrollar el cerebro central que se encarga de controlar dicha planta, validando la correcta intercomunicación de todos los componentes de hardware utilizados y desarrollando una solución de IoT completamente gratuita y sin limitaciones en cuanto a horas de uso o capacidad de almacenamiento, siendo su única desventaja el no poder ofrecer tiempos de refrescamiento muy bajos (menores a 3 segundos). Además, una vez que se tenga un prototipo funcional de todo el sistema propuesto por el proyecto IPTC, debido a su bajo costo que se estima ser hasta 21 veces menor que el de las opciones comerciales disponibles, aún más investigadores y entusiastas tendrán acceso a una herramienta que les permita conducir las pruebas de potencia necesarias en el EPS para que sus misiones CubeSat tengan mayores probabilidades de éxito.

Adicionalmente, el desarrollo de trabajos de investigación como el presente pueden dar lugar a publicaciones que le brinden mayor visibilidad a la iniciativa IPTC, potencialmente atrayendo a más colaboradores y/o futuros usuarios que permitan acelerar el desarrollo del mismo.

Capítulo 11

Conclusiones y recomendaciones

11.1. Conclusiones

1. Se realizó un estudio de las necesidades del cliente, las cuales resultaron en un conjunto de especificaciones que permitieron guiar el desarrollo de las distintas componentes de la solución diseñada.
2. Se desarrolló un modelo analítico que emula el comportamiento de un arreglo solar para CubeSats de 1U, obteniendo un MAPE medio de 1.44 % en las pruebas realizadas y siendo capaz de escalar las salidas de acuerdo a las condiciones externas que se configuren. Además, se implementaron dos plataformas para posteriormente integrarlas con la solución IoT.
3. Se diseñaron controladores digitales para el convertidor reductor de conmutación síncrona presente en la planta MCP. Su validación se realizó con Simulink, donde el controlador PID de tensión cuenta con un % SI de 4.484 %, un t_{set2} de 264 μ s y un e_{ss} de 0 %, y el controlador PI de corriente cuenta con un % SI de 4.478 %, un t_{set2} de 184 μ s, y un e_{ss} de 0 %. Fueron diseñados para proporcionar salidas de hasta 21.6 V y 3.6 A, respectivamente.
4. Se desarrolló e implementó una solución IoT que permite controlar de manera remota las plataformas de SAS desarrolladas, guardar registros periódicos de dichas plataformas en una base de datos en la nube, y desplegar un panel de visualización en donde se puede observar el cambio de diferentes variables en tiempo real. Se validó que los tiempos que toman en procesarse las solicitudes HTTP sean menores a 10 segundos.
5. Se diseñó y validó una propuesta de chasis para colocar los diferentes elementos de hardware de la plataforma de SAS #2, satisfaciendo, como mínimo, el factor de seguridad marginal de 2 en todas las simulaciones realizadas, y presentando un costo de materiales y manufactura menor al presupuestado.

6. Se realizó la integración de las soluciones de SAS e IoT, permitiendo llevar a cabo pruebas experimentales en las que se validó el rendimiento al utilizar instrumentos de laboratorio para emular el comportamiento de arreglos solares, obteniendo un MAPE de 0.47%. Asimismo, se validó que la intercomunicación entre los distintos componentes de hardware a implementar en el producto final del proyecto IPTC se da de forma adecuada mediante la implementación física de una placa de desarrollo.

11.2. Recomendaciones

1. Obtener acceso a sets de datos reales tomados con un SAS comercial a condiciones de referencia y con condiciones externas variadas para validar a un nivel más profundo el rendimiento de modelo analítico desarrollado. En la Universidad de Anhalt, en Alemania, existen colaboradores de la iniciativa IPTC con acceso a un SAS comercial y un panel solar AZUR 3G30A que permitirían obtener los datos requeridos.
2. La propuesta de chasis fue realizada tomando en cuenta las dimensiones de la versión de la planta MCP presente en el laboratorio. Cuando se tenga la versión nueva, se debe verificar si las dimensiones coinciden o sino hacer los ajustes necesarios antes de manufacturar el chasis. Asimismo, se debe tener precaución al apretar los tornillos que mantienen unidas las paredes del chasis, puesto que es posible provocar una fractura si se usa fuerza excesiva.

Bibliografía

- [1] A. Luque and S. Hegedus, *Handbook of Photovoltaic Science and Engineering*. Wiley, 2003.
- [2] E. Babatunde, *Solar Radiation*. IntechOpen, 2012.
- [3] K. Mertens, *Photovoltaics: Fundamentals, Technology, and Practice*. Wiley, 2018.
- [4] M. Kitayama da Silva, M. Gul, and H. Chaudhry, “Review on the Sources of Power Loss in Monofacial and Bifacial Photovoltaic Technologies,” *Energies*, vol. 14, p. 7935, 11 2021.
- [5] R. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, ser. Online access with purchase: Springer. Springer US, 2001.
- [6] L. Corradini, D. Maksimovic, P. Mattavelli, and R. Zane, *Digital Control of High-Frequency Switched-Mode Power Converters*, ser. IEEE Press Series on Power and Energy Systems. Wiley, 2015.
- [7] D. Ulrich and S. Eppinger, *Diseño y desarrollo de productos*. México: McGraw-Hill, 2013.
- [8] STMicroelectronics, “STM32 Nucleo-64 development board with STM32L073RZ MCU, supports Arduino and ST morpho connectivity,” <https://www.st.com/en/evaluation-tools/nucleo-l073rz.html>.
- [9] Components101, “NodeMCU ESP8266,” <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>.
- [10] Mean Well, “60W AC-DC PCB-Mount Green Power Module,” <https://www.meanwell.com/Upload/PDF/IRM-60/IRM-60-SPEC.PDF>.
- [11] SunFounder, “I²C LCD1602,” http://wiki.sunfounder.cc/index.php?title=I%C2%B2C_LCD1602#:~:text=a%20pullup%20resistor.-,I2C%20Address,cases%20it%20may%20be%200x3F.
- [12] Adafruit, “Micro SD Card Breakout Board Tutorial,” <https://learn.adafruit.com/adafruit-micro-sd-breakout-board-card-tutorial/arduino-wiring>.
- [13] MicroJPM, “DS3231 Real Time Clock Module [RTC],” <https://www.microjpm.com/products/ds3231-at24c32-iic-modulo-de-reloj-en-tiempo-real/>.

- [14] —, “LA38-20X3 Rotary Switch Knob 22mm,” <https://www.microjpm.com/products/ad70325/>.
- [15] —, “KCD3 Red Waterproof Rocker Switch With Light,” <https://www.microjpm.com/products/ad71231/>.
- [16] —, “USB 2.0 Standard Female To Female Socket Panel Mount,” <https://www.microjpm.com/products/ad37276/>.
- [17] Amazon, “MPLAB PICkit 3 PIC Microcontroller Chip in-Circuit Debugger Programmer,” <https://www.microchip.com/en-us/development-tool/pg164130>.
- [18] MicroJPM, “6 Pin Stackable Header 2.54mm,” <https://www.microjpm.com/products/stackable-header-6-pins-2-54mm/>.
- [19] S. Bader, X. Ma, and B. Oelmann, “One-diode photovoltaic model parameters at indoor illumination levels—A comparison,” *Solar Energy*, vol. 180, pp. 707–716, 2019.
- [20] Seaward Electronic, “How does temperature and irradiance affect I-V curves?” <https://www.seaward.com/gb/support/solar/faqs/00797-how-does-temperature-and-irradiance-affect-i-v-curves/>.
- [21] Amazon, “MoKo 400x300mm DIY PC Case Dust Mesh Filter,” <https://t.ly/YVI3h>.
- [22] J. Lucero, “Recent satellite history,” Jun 2022. [Online]. Available: <https://www.orbitaleos.com/satellite-history/>
- [23] S. Chakrabarti, “How many satellites are orbiting Earth?” Sep 2021. [Online]. Available: <https://www.space.com/how-many-satellites-are-orbiting-earth>
- [24] A. Poghosyan and A. Golkar, “CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions,” *Progress in Aerospace Sciences*, vol. 88, 11 2016.
- [25] Tecnológico de Costa Rica, “Proyecto Irazú,” Nov 2018. [Online]. Available: <https://www.tec.ac.cr/proyectos/proyecto-irazu-0>
- [26] J. Umaña, “TEC se asocia con Universidad de George Washington para lanzamiento de una nueva misión espacial,” Oct 2018. [Online]. Available: <https://www.tec.ac.cr/hoyeneltec/2018/10/25/tec-se-asocia-universidad-george-washington-lanzamiento-nueva-mision-espacial>
- [27] —, “72 costarricenses figuran en el Congreso Internacional de Astronáutica,” Sep 2022. [Online]. Available: <https://www.tec.ac.cr/hoyeneltec/2022/06/07/universitarios-buscan-fondos-representar-pais-francia-proyecto-espacial-biodomo>
- [28] —, “Segunda misión costarricense llegó con éxito al espacio,” Nov 2022. [Online]. Available: <https://www.tec.ac.cr/hoyeneltec/2022/11/23/segunda-mision-costarricense-llego-exito-espacio>

- [29] J. Carvajal, “Desarrollo de un sistema integrado para la prueba de sistemas de potencia en CubeSats,” 2022.
- [30] M. Langer and J. Bouwmeester, “Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward,” 08 2016.
- [31] Canadian Space Agency, “CubeSats in a nutshell,” Abr 2017. [Online]. Available: <https://www.asc-csa.gc.ca/eng/satellites/cubesat/what-is-a-cubesat.asp>
- [32] D. D. R. E. S. y. D. H. M. Aybar, L. Hidalgo, “Análisis de los modos de fallo del sistema de gestión de energía de un Cubesat,” 05 2021.
- [33] J. Gonzalez-Llorente, S. Munevar, and S. Sánchez-Sanjuán, “Análisis de la Capacidad de Almacenamiento de las Baterías para un Satélite CubeSat 3U,” 11 2015.
- [34] J. Bouwmeester, M. Langer, and E. Gill, “Survey on the implementation and reliability of CubeSat electrical bus interfaces,” *CEAS Space Journal*, vol. 9, 09 2016.
- [35] D. Nagy, “3-Channel Solar Array Simulator for CubeSat Power Budget Verification,” Master’s thesis, Technische Universität München, 2018.
- [36] R. Venkateswari and S. Sreejith, “Factors influencing the efficiency of photovoltaic system,” *Renewable and Sustainable Energy Reviews*, vol. 101, pp. 376–394, 03 2019.
- [37] H. Thorvardarson, F. Gøttsche, and F. Tonicello, “A New European High Fidelity Solar Array Simulator for Near Earth and Deep Space Applications,” *E3S Web of Conferences*, vol. 16, p. 14002, 01 2017.
- [38] Keysight, “MP4300 series solar array simulator system.” [Online]. Available: <https://www.keysight.com/us/en/products/dc-power-supplies/dc-power-solutions/mp4300-series-solar-array-simulator-system.html>
- [39] AMETEK, “Solar Array Simulator.” [Online]. Available: <https://www.programmablepower.com/products/engineered-systems/solar-array-simulator-sas>
- [40] A. Smets, K. Jäger, O. Isabella, R. van Swaaij, and M. Zeman, *Solar Energy: The Physics and Engineering of Photovoltaic Conversion, Technologies and Systems*. UIT Cambridge, 2016.
- [41] “A Guide to CubeSat Mission and Bus Design: 5.5 Power Generation,” Mar 2022. [Online]. Available: <https://pressbooks-dev.oer.hawaii.edu/epet302/chapter/5-5-power-generation/#:~:text=If%20there%20are%20two%20panels,power%20of%20a%20single%20panel>.
- [42] “State-of-the-Art of Small Spacecraft Technology: 3.0 Power,” 2022. [Online]. Available: <https://www.nasa.gov/smallsat-institute/sst-soa/power>
- [43] A. McEvoy, T. Markvart, and L. Castaner, *Practical Handbook of Photovoltaics: Fundamentals and Applications*. Elsevier Science, 2011.

- [44] PV Education, “Series Resistance,” <https://www.pveducation.org/pvcdrom/solar-cell-operation/series-resistance>.
- [45] —, “Shunt Resistance,” <https://www.pveducation.org/pvcdrom/solar-cell-operation/shunt-resistance>.
- [46] H. M. Ridha, “Parameters extraction of single and double diodes photovoltaic models using Marine Predators Algorithm and Lambert W function,” *Solar Energy*, vol. 209, pp. 674–693, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0038092X20309968>
- [47] A. Escobar-Molero, “Diseño de un Convertidor Buck Síncrono Controlado Digitalmente,” Ph.D. dissertation, 09 2016.
- [48] IBM, “What is the internet of things (IoT)?” <https://www.ibm.com/topics/internet-of-things>.
- [49] Amazon, “What is IoT (Internet of Things)?” <https://aws.amazon.com/what-is/iot/>.
- [50] Arm Limited , “What Is IoT Cloud?” <https://www.arm.com/glossary/iot-cloud#:~:text=An%20IoT%20cloud%20is%20a,real%2Dtime%20operations%20and%20processing>.
- [51] Trackinno, “How IoT Works – Part 4: User Interface,” <https://trackinno.com/iot/how-iot-works-part-4-user-interface/>.
- [52] Oracle, “What Is a Database?” <https://www.oracle.com/database/what-is-database/>.
- [53] Codecademy, “HTTP Requests,” <https://www.codecademy.com/article/http-requests>.
- [54] IBM, “HTTP Requests,” <https://www.ibm.com/docs/en/cics-ts/5.3?topic=protocol-http-requests>.
- [55] Amazon Web Services, “What Is the MQTT Protocol and How Does it Work?” <https://www.emqx.com/en/blog/the-easiest-guide-to-getting-started-with-mqtt>.
- [56] EMQ Technologies, “What is MQTT?” <https://aws.amazon.com/what-is/mqtt/>.
- [57] phoenixNAP, “What Is Response Time?” [https://phoenixnap.com/glossary/response-time#:~:text=Response%20time%20is%20a%20performance,service%20time%20\(request%20fulfillment\)](https://phoenixnap.com/glossary/response-time#:~:text=Response%20time%20is%20a%20performance,service%20time%20(request%20fulfillment)).
- [58] V. Stornelli, M. Muttillio, T. De Rubeis, and I. Nardi, “A new simplified five-parameter estimation method for single-diode model of photovoltaic panels,” *Energies*, vol. 12, no. 22, p. 4271, 2019.
- [59] C. A. Rodríguez-Delgado, “Gestión de requerimientos para el desarrollo de un sistema integrado de pruebas para CubeSats,” 06 2023.

- [60] Arduino, “Arduino Mega 2560 Rev3.” [Online]. Available: <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us>
- [61] “STM32 Nucleo-64 development board with STM32L073RZ MCU.” [Online]. Available: https://www.st.com/en/product/nucleo-l073rz?ecmp=tt9470_gl_link_feb2019&rt=db&id=DB2196
- [62] “Data Sheet 3G30GA,” *AZUR SPACE*. [Online]. Available: https://www.azurspace.com/images/products/0003401-01-01_DB_3G30A.pdf
- [63] AZUR SPACE, “Data Sheet 4G32C - Advanced.” [Online]. Available: https://www.azurspace.com/images/0005979-01-01_DB_4G32C_Advanced.pdf
- [64] SpectroLab, “Data Sheet Ultra Triple Junction (UTJ).” [Online]. Available: https://www.spectrolab.com/photovoltaics/UTJ-CIC_Data_Sheet.pdf
- [65] CESI, “Data Sheet CTJ30.” [Online]. Available: <https://www.cesi.it/app/uploads/2020/03/Datasheet-CTJ30-1.pdf>
- [66] M. Tran, Y. Sun, Y. Kobori, A. Kuwana, and H. Kobayashi, “Overshoot Cancellation Based on Balanced Charge-Discharge Time Condition for Buck Converter in Mobile Applications,” in *2019 IEEE 13th International Conference on ASIC (ASICON)*. IEEE, 2019, pp. 1–4.
- [67] N. D. Bhat, D. B. Kanse, S. D. Patil, and S. D. Pawar, “DC/DC Buck Converter Using Fuzzy Logic Controller,” in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 182–187.
- [68] D. Guiza, D. Ounnas, Y. Soufi, and A. Bouden, “DC-DC Buck Converter Control Improvement,” in *2021 18th International Multi-Conference on Systems, Signals Devices (SSD)*, 2021, pp. 1419–1424.
- [69] K. A. Gómez-Villagra, “Diseño de un sistema embebido para el control y regulación del proceso dinámico de carga y descarga de celdas electroquímicas y su posterior validación para satélites CubeSat de 1U,” 06 2022.
- [70] P. C. C. Villalobos, “Diseño de una carga electrónica programable para el desarrollo de un sistema de pruebas de potencia en CubeSats,” 06 2023.
- [71] K. Yeo, N. Krivova, and S. Solanki, “Solar cycle variation in solar irradiance,” *Space Science Reviews*, vol. 186, pp. 137–167, 2014.
- [72] H. K. Kim and C. Y. Han, “Analytical and numerical approaches of a solar array thermal analysis in a low-earth orbit satellite,” *Advances in Space Research*, vol. 46, no. 11, pp. 1427–1439, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0273117710005818>
- [73] J. Banik, S. Kiefer, M. LaPointe, and P. LaCorte, “On-orbit validation of the roll-out solar array,” in *2018 IEEE Aerospace Conference*, 2018, pp. 1–9.

- [74] K.-B. Shin, C.-G. Kim, C.-S. Hong, and H.-H. Lee, “Thermal distortion analysis of orbiting solar array including degradation effects of composite materials,” *Composites Part B: Engineering*, vol. 32, no. 4, pp. 271–285, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1359836801000208>
- [75] “Azure IoT Hub pricing.” [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/iot-hub/>
- [76] “AWS Free Tier with AWS IoT Core.” [Online]. Available: <https://aws.amazon.com/iot-core/pricing/>
- [77] “Arduino Cloud - Plans.” [Online]. Available: <https://cloud.arduino.cc/plans>
- [78] “Adafruit IO Plus,” <https://io.adafruit.com/plus>.
- [79] Microchip Technology, “PIC16F1786,” <https://www.microchip.com/en-us/product/PIC16F1786>.
- [80] STMicroelectronics, “NUCLEO-L073RZ,” <https://estore.st.com/en/nucleo-l073rz-cpn.html>.
- [81] Arduino, “Mega 2560 Rev3 Overview,” <https://docs.arduino.cc/hardware/mega-2560>.
- [82] —, “Arduino Mega 2560 Rev3,” <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us>.
- [83] MicroJPM, “Arduino Mega 2560 R3,” <https://www.microjpm.com/products/arduino-mega-2560/>.
- [84] CRCibernética, “Official Arduino Mega 2560 REV 3,” <https://www.crcibernetica.com/official-arduino-mega-2560-rev-3/>.
- [85] MicroJPM, “NodeMCU ESP8266 Development Board,” <https://www.microjpm.com/products/ad34917/>.
- [86] CRCibernética, “NodeMcu V3 Lua LoLin ESP8266 Dev Board,” <https://www.crcibernetica.com/nodemcu-v3-lua-lolin-esp8266-dev-board/>.
- [87] Arduino, “Nano 33 IoT Overview,” <https://docs.arduino.cc/hardware/nano-33-iot>.
- [88] —, “Arduino Nano 33 IoT,” <https://store-usa.arduino.cc/products/arduino-nano-33-iot>.
- [89] MicroJPM, “Arduino Nano 33 IoT with Headers,” <https://www.microjpm.com/products/ad68134/>.
- [90] NINGBO, “LED DIGIT DISPLAY,” <https://c1555f5ec9.clvaw-cdnwnd.com/34662fcf1f1e607c561442431023ac8e/200000287-c7a3dc89cc/5611A%20Datasheet.pdf>.

- [91] MicroJPM, “Display LED Rojo 7 Segmentos 1 Dígito Anodo Comun 0.56,” <https://www.microjpm.com/products/display-led-rojo-7-segmentos-1-digito-anodo-comun-0-56/>.
- [92] —, “Blue Display 16x2 w/ Serial Interface,” <https://www.microjpm.com/products/ad30797/>.
- [93] SunFounder, “LCD2004 Module,” http://wiki.sunfounder.cc/index.php?title=LCD2004_Module.
- [94] MicroJPM, “LCD2004 20x4 Character LCD Display Module Blue Backlight,” <https://www.microjpm.com/products/ad56548/>.
- [95] SparkFun, “SparkFun Qwiic EEPROM Breakout - 512Kbit,” <https://www.sparkfun.com/products/18355>.
- [96] Adafruit, “Adafruit I2C Non-Volatile FRAM Breakout - 256Kbit / 32KByte,” <https://www.adafruit.com/product/1895>.
- [97] SparkFun, “Serial Flash Breakout - Assembled 128Mbit,” <https://www.sparkfun.com/products/17115>.
- [98] Kingston Technology, “A guide to SD and microSD card types,” <https://www.kingston.com/en/blog/personal-storage/microsd-sd-memory-card-guide>.
- [99] MicroJPM, “microSD USB Reader,” <https://www.microjpm.com/products/microsd-usb-reader/>.
- [100] Compubetel, “KINGSTON 8GB MICROSDHC,” <https://compubetel.com/producto/kingston-8gb-microsdhc-class-4-2/>.
- [101] Microchip Technology, “PICKIT™ 3 IN-CIRCUIT DEBUGGER.” [Online]. Available: <https://www.microchip.com/en-us/development-tool/pg164130>
- [102] “DP800 SERIES | HIGH PERFORMANCE LINEAR DC POWER SUPPLIES, author=RIGOL TECHNOLOGIES,” <https://www.rigolna.com/products/dc-power-loads/dp800/>.
- [103] R. TECHNOLOGIES, “DL3000 SERIES | PRECISION DC ELECTRONIC LOADS,” <https://www.rigolna.com/products/dc-power-loads/dl3000/>.
- [104] Arduino, “Arduino Uno Rev3,” <https://store-usa.arduino.cc/products/arduino-uno-r3>.
- [105] A. Hussein, “A simple approach to extract the unknown parameters of pv modules,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 25, no. 5, pp. 4431–4444, 2017.
- [106] R. Kulhary, “OpenCV – Overview,” <https://www.geeksforgeeks.org/opencv-overview/>.

- [107] J. Dalbins and K. Allaje, “Development of Maximum Power Point Tracking for ESTCube-2.”
- [108] W. Guter, F. Dunzer, L. Ebel, K. Hillerich, W. Köstler, T. Kubera, M. Meusel, B. Postels, and C. Wächter, “Space solar cells–3G30 and next generation radiation hard products,” in *E3S Web of Conferences*, vol. 16. EDP Sciences, 2017, p. 03005.
- [109] H. Tian, F. Mancilla-David, K. Ellis, E. Muljadi, and P. Jenkins, “Detailed performance model for photovoltaic systems,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2012.
- [110] W. De Soto, S. A. Klein, and W. A. Beckman, “Improvement and validation of a model for photovoltaic array performance,” *Solar energy*, vol. 80, no. 1, pp. 78–88, 2006.
- [111] V. L. Brano, A. Orioli, G. Ciulla, and A. Di Gangi, “An improved five-parameter model for photovoltaic modules,” *Solar Energy Materials and Solar Cells*, vol. 94, no. 8, pp. 1358–1370, 2010.
- [112] C. H. Matthew T. Boyd, “Single Diode Equivalent Circuit Models,” *PV Performance Modeling Collaborative*, 2012.
- [113] PyVISA, “PyVISA: Control your instruments with Python,” <https://pyvisa.readthedocs.io/en/latest/>.
- [114] MathWorks, “State-space model,” <https://au.mathworks.com/help/control/ref/ss.html>.
- [115] —, <https://www.mathworks.com/help/control/ref/tf.html>, title=Transfer function model.
- [116] —, <https://www.mathworks.com/help/ident/ref/dynamicsystem.bode.html>, title=Bode plot of frequency response.
- [117] IoT Consulting, “MQTT vs HTTP – Qué elegir para tu proyecto IoT,” <https://iotconsulting.tech/mqtt-vs-http-que-elegir-para-tu-proyecto-iot/>.
- [118] L. Llamas, “¿Qué es MQTT? Su importancia como protocolo IoT,” <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>.
- [119] Flespi, “HTTP vs MQTT performance tests,” <https://flespi.com/blog/http-vs-mqtt-performance-tests>.
- [120] Microsoft, “Azure IoT Hub pricing,” <https://azure.microsoft.com/en-us/pricing/details/iot-hub/>.
- [121] —, “Tutorial: Customize the dashboard and manage devices in Azure IoT Central,” <https://learn.microsoft.com/en-us/azure/iot-central/retail/tutorial-in-store-analytics-customize-dashboard>.

- [122] Amazon Web Services, “AWS IoT Core pricing,” <https://aws.amazon.com/iot-core/pricing/>.
- [123] Arduino, “Arduino Cloud Plans,” <https://cloud.arduino.cc/plans>.
- [124] Adafruit, “The internet of things for everyone,” <https://io.adafruit.com/>.
- [125] Google for Developers, “Google Apps Script overview,” <https://developers.google.com/apps-script/overview>.
- [126] Protocase, “Electronic Enclosure Design 101 – Sheet Metal Design Guide,” <https://www.protocase.com/blog/2017/02/16/electronic-enclosure-design-101/>.
- [127] S. Naghiloo, “Polycarbonate vs. Acrylic | Melting Point & Uses,” <https://study.com/academy/lesson/polycarbonate-vs-acrylic.html>.
- [128] Engineers Edge, “Fastener Scre Torque Clamp Force Design Calculator,” <https://www.engineersedge.com/torque.htm>.
- [129] Monster Bolts, “Metric Fasteners - Socket Screw Torque and Specifications,” <https://monsterbolts.com/pages/metric-socket-screw-torque-specs>.
- [130] Melco Tech, “Torque Specifications,” https://www.melco-service.com/docs/AMAYA_OS_v10/AMAYA/Torque_Specifications.htm.
- [131] TEC, “CORTE Y GRABADO LÁSER,” https://www.tec.ac.cr/sites/default/files/media/doc/corte_laser_-_condiciones_del_servicio.pdf.

Anexo A

Validación de las métricas establecidas

Tabla de validación de especificaciones

No.	Métrica	Imp.	Unidad	Valor marginal	Valor ideal	Valor obtenido
1	Error porcentual absoluto medio (MAPE) respecto a valores de referencia	5	%	< 5	< 3	0.47
2	Número de paneles solares con los que se prueba que el modelo funciona adecuadamente	5	# de paneles	2	4	4
3	El modelo es sensible a cambios en las condiciones de irradiancia, temperatura y ángulo de incidencia	5	Binaria	Sí	Sí	Sí
4	Número de paneles a los que se les genera una hoja de datos exitosamente	4	# de paneles	2	4	4
5	El modelo es implementado en un software de código abierto	3	Binaria	Sí	Sí	Sí (Python y C/C++)
6	Tiempo que le toma al modelo calcular la salida para un punto de operación dado	3	ms	1000	500	0.6009 ± 0.2333 (plataforma #1) 2.0965 ± 0.4360 (plataforma #2)
7	Cantidad de memoria que requieren las funciones utilizadas por el modelo	3	KB	60	30	10.54
8	Cantidad de paneles solares que conforman el arreglo	2	# de paneles	6	6	6
9	Porcentaje de error en estado estacionario	5	%	< 3	< 1	0 (PID y PI)
10	Porcentaje de sobreimpulso	5	%	< 15	< 5	4.484 (PID) 4.478 (PI)
11	Tiempo de estabilización	4	ms	< 100	< 50	0.264 (PID) 0.184 (PI)
12	Rango de niveles de irradiancia en que el modelo funciona adecuadamente	5	W/m ²	0 a 1375	0 a 1375	0 a 1375
13	Rango de niveles de temperatura en que el modelo funciona adecuadamente	5	°C	-80 a 80	-80 a 80	-80 a 80
14	Rango de ángulos de incidencia en que el modelo funciona adecuadamente	5	°	0 a 90	0 a 90	0 a 90
15	Número de paneles solares que puede seleccionar el usuario	5	# de paneles	2	4	5 (el usuario debe proporcionar los parámetros del quinto panel solar)
16	Se puede controlar el valor de tensión de salida o el valor de corriente de salida	5	Binaria	Sí	Sí	Sí

No.	Métrica	Imp.	Unidad	Valor marginal	Valor ideal	Valor obtenido
17	Se puede iniciar, actualizar o detener una prueba cuando el usuario lo requiera	5	Binaria	Sí	Sí	Sí
18	Se puede acceder desde una red de internet distinta a la del hardware	5	Binaria	Sí	Sí	Sí
19	Tiempo que le toma al hardware recibir un mensaje desde la plataforma	3	s	10	0.5	2.1853 ± 0.6338 (plataforma #1) 4.7927 ± 0.6945 (plataforma #2)
20	Se pueden visualizar los valores de salida medidos (tensión, corriente, potencia)	5	Binaria	Sí	Sí	Sí
21	Se pueden visualizar los valores de salida esperados (tensión, corriente, potencia)	5	Binaria	Sí	Sí	Sí
22	Tiempo de actualización de los parámetros en la base de datos	3	s	10	0.5	5.2195 ± 1.8943 (plataforma #1) 4.023 ± 2.040 (plataforma #2)
23	Se crea una gráfica en función del tiempo con los valores de salida medidos	2	Binaria	Sí	Sí	Sí
24	Se puede visualizar el tiempo transcurrido desde que se inició una prueba	2	Binaria	Sí	Sí	Sí
25	Tiempo máximo de pruebas por día	5	h	4	8	Sin límite. Máximo validado: 19.33
26	Número de variables que se pueden monitorear y controlar sin el pago de una suscripción	5	# de variables	20	40	Sin límite, máximo validado: 90
27	Se puede crear la interfaz con las características deseadas sin el pago de una suscripción	5	Binaria	Sí	Sí	Sí
28	En la selección de componentes se consideran únicamente los disponibles en el laboratorio	5	Binaria	Sí	Sí	Sí
29	Los datos de las pruebas se guardan en la nube	5	Binaria	Sí	Sí	Sí
30	Los datos de las pruebas se guardan en una memoria local	3	Binaria	Sí	Sí	Sí
31	Existe un manual de uso para operar el hardware	5	Binaria	Sí	Sí	Sí
32	Existe un manual de uso para operar el software del modelo	5	Binaria	Sí	Sí	Sí
33	Existe un manual de uso para operar la plataforma IoT	5	Binaria	Sí	Sí	Sí

No.	Métrica	Imp.	Unidad	Valor marginal	Valor ideal	Valor obtenido
34	Número de paneles solares a los que se le generan hojas de datos con el modelo analítico	4	# de hojas	2	4	4
35	Existe una guía con los pasos de implementación de la plataforma IoT	3	Binaria	Sí	Sí	Sí
36	Existe un conector externo para alimentación AC	5	Binaria	Sí	Sí	Sí
37	Todos los microcontroladores empleados cuentan con conectores para reprogramarlos	5	Binaria	Sí	Sí	Sí
38	Todos los microcontroladores empleados cuentan con conectores para leer sus puertos seriales	5	Binaria	Sí	Sí	Sí
39	Número de conectores externos tipo banana	5	# de conectores	4	4	4
40	Existe un interruptor para encender y apagar el hardware	5	Binaria	Sí	Sí	Sí
41	Todos los elementos externos del chasis están debidamente etiquetados	5	Binaria	Sí	Sí	Sí
42	Existe una pantalla externa para visualizar los valores de entrada y salida	2	Binaria	Sí	Sí	Sí
43	Existe un método de ventilación en el chasis	3	Binaria	Sí	Sí	Sí
44	Todos los componentes internos cuentan con mecanismos para mantenerse en una posición fija	5	Binaria	Sí	Sí	Sí
45	Existe un mecanismo para filtrar polvo	3	Binaria	Sí	Sí	Sí
46	Costo total de manufactura del chasis	5	\$USD	< 80	< 50	19.23
47	Todos los equipos y herramientas considerados en el proceso de manufactura existen en el TEC	4	Binaria	Sí	Sí	Sí
48	Todos los procedimientos de ensamblaje y desensamblaje se pueden realizar con un set básico de desatornilladores, alicates y pinzas	4	Binaria	Sí	Sí	Sí

No.	Métrica	Imp.	Unidad	Valor marginal	Valor ideal	Valor obtenido
49	Todos los procesos de manufactura se pueden realizar sin requerir personal técnico altamente calificado	3	Binaria	Sí	Sí	Sí
50	Los materiales no se degradan con agua o agentes desinfectantes comunes	5	Binaria	Sí	Sí	Sí
51	Costo total de los materiales del chasis	5	\$USD	< 80	< 50	70.58
52	Factor de seguridad al hacer simulaciones de esfuerzos y deformaciones	5	N/A	> 2	> 3	> 2.03
53	Temperaturas a las que se pueden ver expuestos los materiales del chasis	4	°C	0 a 60	0 a 70	-30 a 90
54	Volumen del chasis	5	m ³	< 0.5	< 0.3	0.02
55	Peso del chasis totalmente ensamblado	4	kg	< 10	< 7	3.02
56	Existen elementos mecánicos que faciliten su transporte	2	Binaria	Sí	Sí	Sí
57	Existen planos mecánicos y de manufactura para el chasis	5	Binaria	Sí	Sí	Sí
58	Existe una lista de materiales y componentes para el chasis	5	Binaria	Sí	Sí	Sí
59	Existe un manual de ensamblaje para el chasis	4	Binaria	Sí	Sí	Sí

Anexo B

Repositorios del proyecto

En la Tabla B.1, se pueden encontrar los enlaces a un repositorio de GitHub, una carpeta de Google Drive y una *Notebook* de Google Colab, así como la descripción del contenido de cada uno.

Tabla B.1: Enlaces de los repositorios del proyecto.

Enlace	Contenido
GitHub	Controladores digitales para la planta MCP: programas desarrollados para el modelado del convertidor reductor síncrono, y el diseño y simulación de controladores PI y PID para la corriente y tensión. Programado en MATLAB y Simulink.
	Componente #1 del SAS: algoritmo de visión computacional, método de Hussein, y generación de hojas de datos con tablas resumen y gráficas. Programado en Python.
	Componente #2 del SAS, plataforma de SAS #1: programas para ejecutar con un ordenador y controlar instrumentos de laboratorio RIGOL. Programado en Python.
	Componente #2 del SAS, plataforma de SAS #2: programas para implementar en microcontroladores de la placa de desarrollo. Programado en C/C++.
	Solución de IoT: programas para implementar en Google Apps Script y establecer un enlace con Google Sheets. Programado en JavaScript.
Google Drive	Todos los programas desarrollados, modelos 3D, guías, y resultados de todas las pruebas experimentales y simulaciones realizadas para las soluciones de SAS, IoT, chasis y controladores digitales.
Google Colab	<i>Notebook</i> utilizada para cambiar los parámetros de la interfaz de control periódicamente, de acuerdo a lo indicado en archivos .CSV de entrada. Programado en Python.

Anexo C

Guías para usuarios

En la Tabla C.1 se proporcionan enlaces para acceder a las guías creadas para orientar a los usuarios en el uso de los recursos desarrollados.

Tabla C.1: Enlaces de los recursos de la solución de chasis.

Guía	Contenido
Software de solución de SAS	Resumen de las funciones presentes en cada programa de software desarrollado.
Hardware de solución de SAS	Secuencia de pasos a seguir para configurar los componentes de hardware de ambas plataformas de SAS antes de realizar pruebas.
Uso de plataforma IoT	Guía para usar las componentes de la plataforma de IoT desarrollada.
Configuración de plataforma IoT	Secuencia de pasos a seguir para crear un programa de <i>Google Apps Script</i> desde cero.
Manufactura y ensamble de chasis	Secuencia de pasos detallada para manufacturar y ensamblar el chasis, incluyendo sus conexiones eléctricas.

Anexo D

Recursos adicionales del chasis

En la Tabla D.1 se proporcionan enlaces para acceder a los anexos preparados para la solución de chasis:

Tabla D.1: Enlaces de los recursos de la solución de chasis.

Enlace	Contenido
Planos	Planos de las piezas a cortar en acrílico, así como el ensamblaje del sistema de filtrado de polvo.
Modelos 3D	Archivos de todos los modelos 3D creados en formatos compatibles con Autodesk Fusion 360, Autodesk Inventor y STEP.
Simulaciones	Archivos de todas las simulaciones realizadas en Autodesk Fusion 360 para validar el diseño del chasis.
Corte láser	Dos archivos PDF que contienen los contornos con el formato de colores y grosores establecido por el LAIMI para ser cortados en láminas de acrílico [131]. También se coloca el archivo DWG donde se pueden editar.
Lista de materiales	Hoja de cálculo con enlaces en los que se pueden comprar los componentes requeridos para el chasis.

Por otra parte, en la Tabla D.2 se presentan las denominaciones asignadas a cada plano, junto con la pieza o ensamble asociados.

Tabla D.2: Denominaciones de los planos del chasis.

Denominación	Plano
IPTC-SAS-01	Ensamble general
IPTC-SAS-02	Panel frontal
IPTC-SAS-03	Panel trasero
IPTC-SAS-04	Panel lateral
IPTC-SAS-05	Panel inferior
IPTC-SAS-06	Panel superior
IPTC-SAS-07	Ensamble de filtro de polvo
IPTC-SAS-08	Anillo espaciador

Anexo E

Placa de desarrollo de SAS

En la Figura E.1, se presenta el diagrama de conexiones de la placa de desarrollo utilizada para la plataforma de SAS # 2. En la Tabla E.1, se aclara cuáles componentes se ensamblan en cada elemento del diagrama, y en la Tabla E.2 las conexiones asociadas a cada interruptor. La letra “H” hace referencia a conectores en fila tipo hembra, “TB” a bloques terminales, y “SW” a interruptores.

Tabla E.1: Ensamblaje de componentes en conectores.

Connector	Conexión
H1, H2	MCU_M (NUCLEO-64 L073RZ)
H3, H4	MCU_S2 (NodeMCU ESP8266)
H5	Micro SD Card Breakout Board+
H6	MCU_S1 (Arduino UNO) I/O
H7	RTC DS3231
TB1	Entrada de 5V
TB2	LCD1602

Tabla E.2: Conexiones asociadas a interruptores.

Switch	Descripción
SW1	tipo SPDT, 5V hacia MCU_M
SW2	tipo SPDT, 5V hacia MCU_S2
SW3	tipo DIP, UART entre MCU_M y MCU_S2
SW4	tipo DIP, UART entre MCU_M y MCU_S1

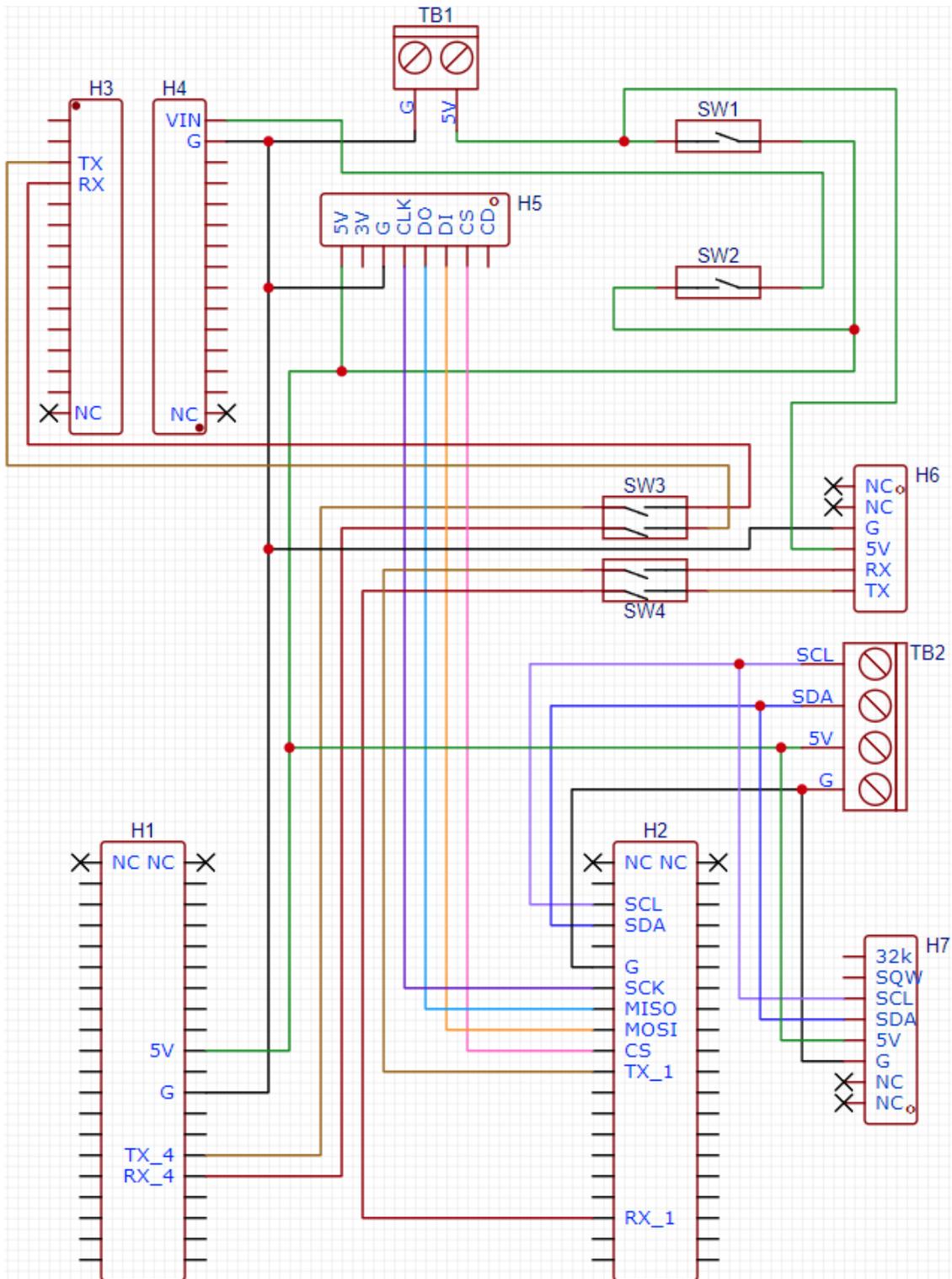


Figura E.1: Diagrama de conexiones de la placa de desarrollo. Realizado con EasyEDA.

En la Figura E.2, Se presenta una comparación de la placa de desarrollo antes y después de ensamblar los componentes.

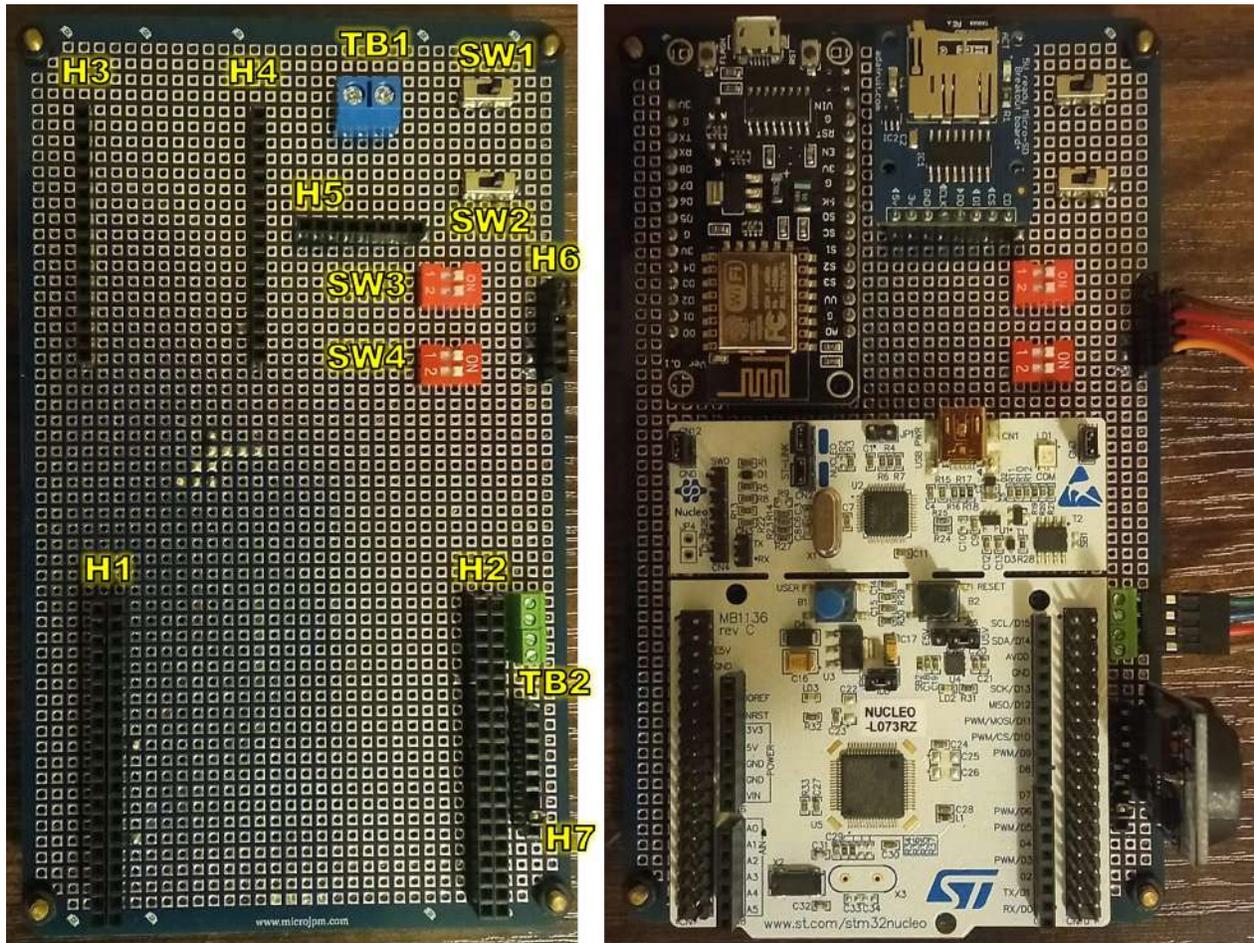


Figura E.2: Placa de desarrollo antes y después de colocar los componentes.

Cabe aclarar que los conectores H1, H2, H3, H4, H6 y H7 cuentan con terminales donde no se hace ninguna conexión, tal y como se indica en la Figura E.1 con el símbolo en forma de “X” y la notación “NC”. Esto se debe a que se utilizaron únicamente los conectores en fila disponibles en el laboratorio, por lo que en algunos casos no se encontraron conectores con el número exacto de terminales requeridos.

No obstante, en la Tabla E.3 se hace el desglose de los componentes y su respectivo precio, en donde se colocan los conectores con el número adecuado de terminales que se necesitan, a excepción del conector para el MCU_M, puesto que no se fabrican conectores de 19 terminales a doble fila.

Tabla E.3: Desglose de componentes y precios de placa de desarrollo.

Componente	Cantidad	Precio unitario	Precio total	Expendedor
NUCLEO-64 L073RZ	1	\$ 21.00 (*)	\$ 21.00	Mouser
NodeMCU ESP8266	1	\$ 10.95	\$ 10.95	MicroJPM
Adafruit Micro SD Card Breakout Board+	1	\$ 19.95	\$ 19.95	MicroJPM
LCD1602 w/ I2C interface	1	\$ 8.50	\$ 8.50	MicroJPM
DS3231 Real Time Clock Module	1	\$ 10.95	\$ 10.95	MicroJPM
SPDT Slide Switch	2	\$ 0.30	\$ 0.60	MicroJPM
2-Bit Slide Type DIP Switch	2	\$ 0.50	\$ 1.00	MicroJPM
DG128-2 Terminal Block 2 Poles 5mm	1	\$ 0.58	\$ 0.58	MicroJPM
DG350-2 Terminal Block 2 Poles 3.5mm	2	\$ 0.50	\$ 1.00	MicroJPM
4 Pin Female Single Row 2.54mm Header	1	\$ 0.25	\$ 0.25	MicroJPM
6 Pin Female Single Row 2.54mm Header	1	\$ 0.30	\$ 0.30	MicroJPM
8 Pin Female Single Row 2.54mm Header	1	\$ 0.35	\$ 0.35	MicroJPM
15 Pin Female Single Row 2.54mm Header	2	\$ 0.40	\$ 0.80	MicroJPM
2x20 2.54mm Pin Double Row Female Header	2	\$ 0.90	\$ 1.80	MicroJPM
90mm x 150mm Perforated Copper Plate with Dots	1	\$ 3.85	\$ 3.85	MicroJPM
Hook-up Wire (24 AWG)	1	\$ 2.80	\$ 2.80	MicroJPM
Total			\$ 84.68	

* Costo estimado con envío, puesto que no se encontró a la venta en Costa Rica.

Anexo F

Figuras de pruebas experimentales

En este anexo se analizan en mayor detalle cada una de las curvas obtenidas después de las pruebas experimentales realizadas en la plataforma de SAS #2.

F.1. Pruebas de diferentes dimensiones de arreglos solares

F.1.1. Arreglo 1x1

Se inicia analizando las curvas obtenidas para los arreglos 1x1 de cada panel. En el caso del AZUR 3G30A, Figura F.5, se puede observar que después de cierto punto una vez pasado el punto de MPP, los ARPES aumentan considerablemente, debido a que conforme se disminuye la corriente, el modo de operación de los controladores de la fuente empiezan a oscilar. A pesar de esto, el MAPE resultante se encuentra dentro de rangos aceptables. En el caso del AZUR 4G32C, Figura F.3, se evidencia un comportamiento más estable, a pesar de que nuevamente existen oscilaciones una vez la corriente empieza a disminuir. Asimismo, se observa que los ARPEs en los últimos puntos son desproporcionadamente altos a pesar que se sigue la tendencia esperada. En el caso del CESI CTJ30, Figura F.5, y del Spectrolab UTJ, Figura F.7, se observan resultados similares.

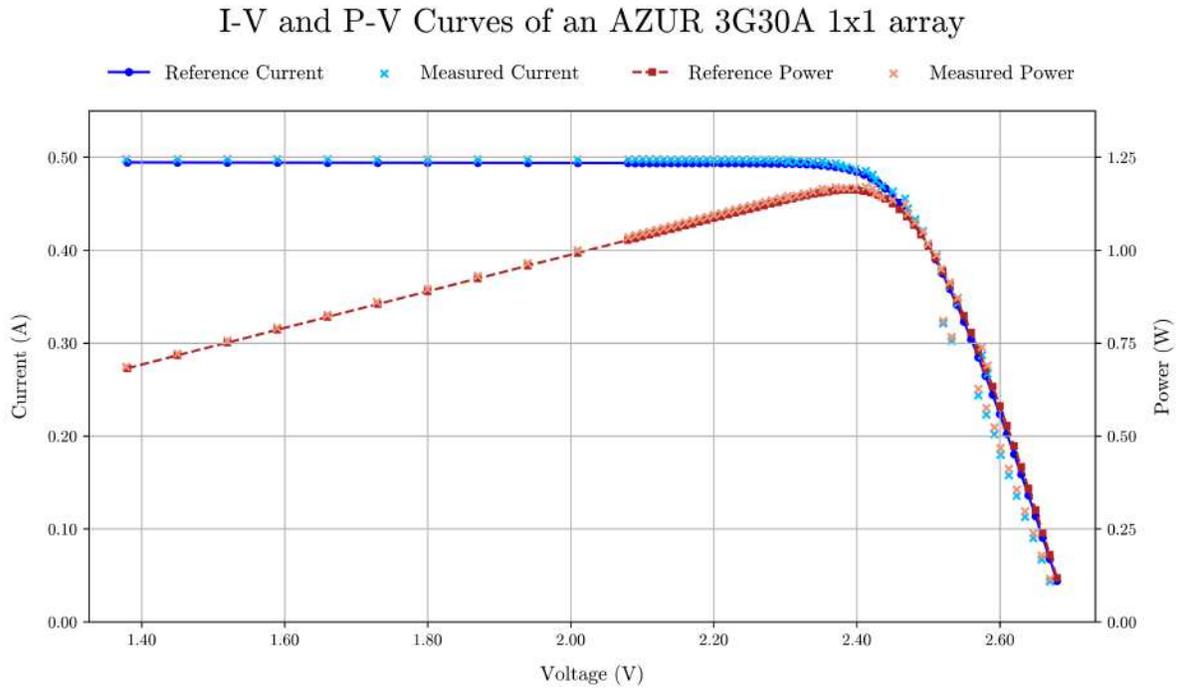


Figura F.1: Curvas I-V y P-V de un arreglo 1x1 (AZUR 3G30A).

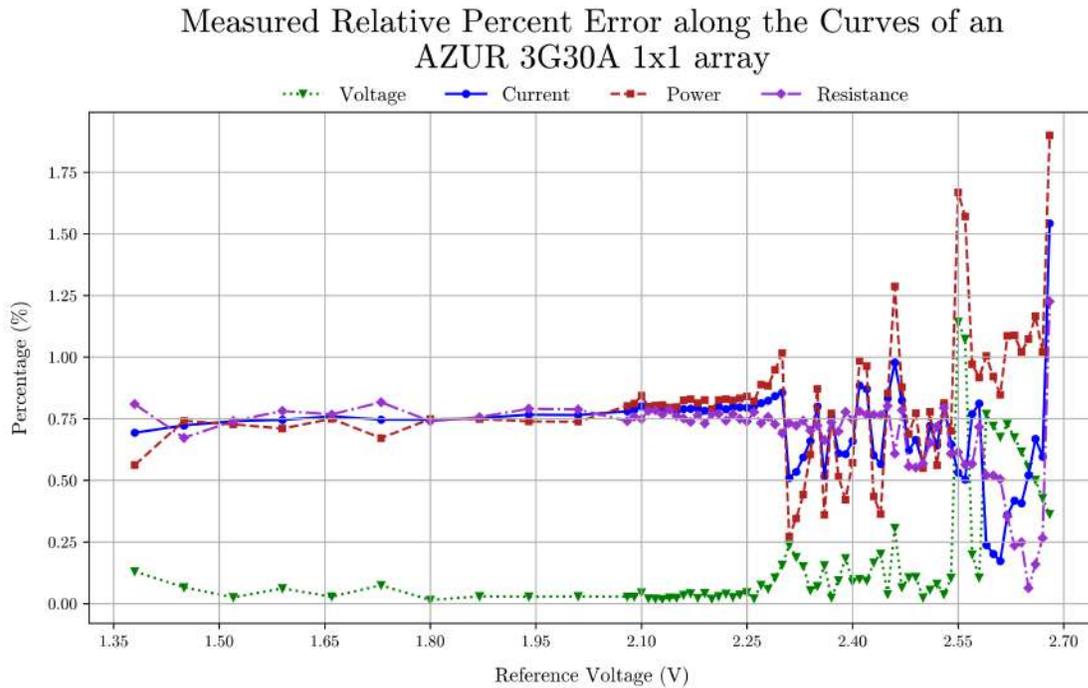


Figura F.2: Porcentajes de error en curvas características de un arreglo 1x1 (AZUR 3G30A).

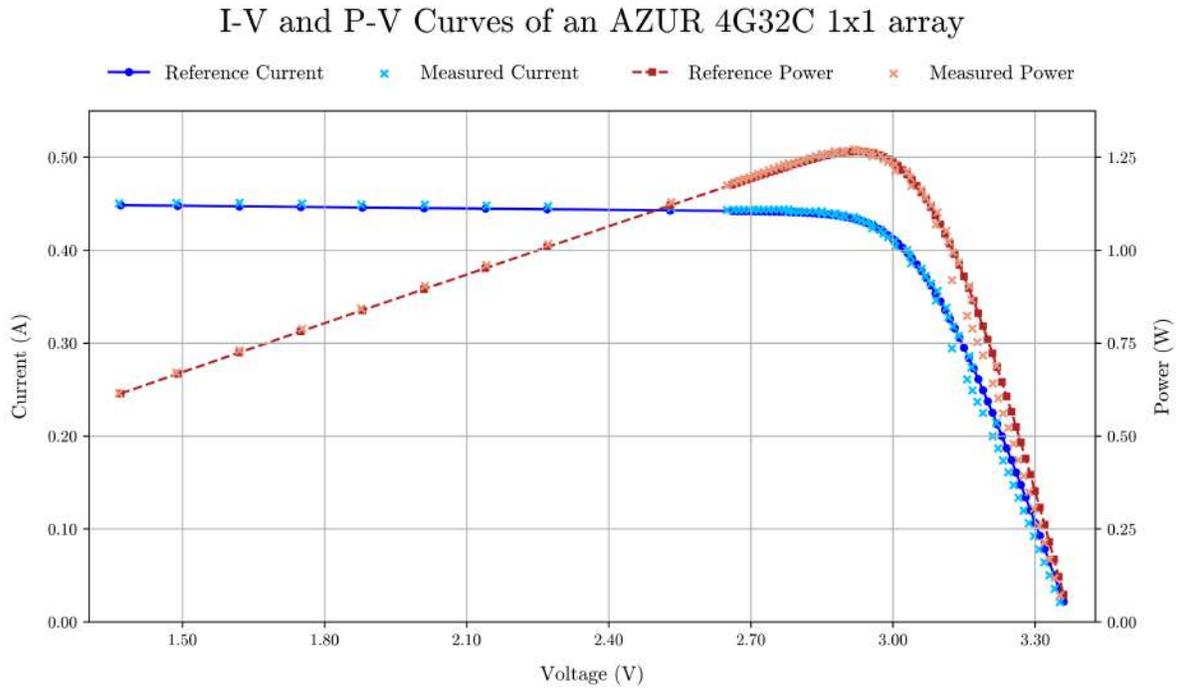


Figura F.3: Curvas I-V y P-V de un arreglo 1x1 (AZUR 4G32C).

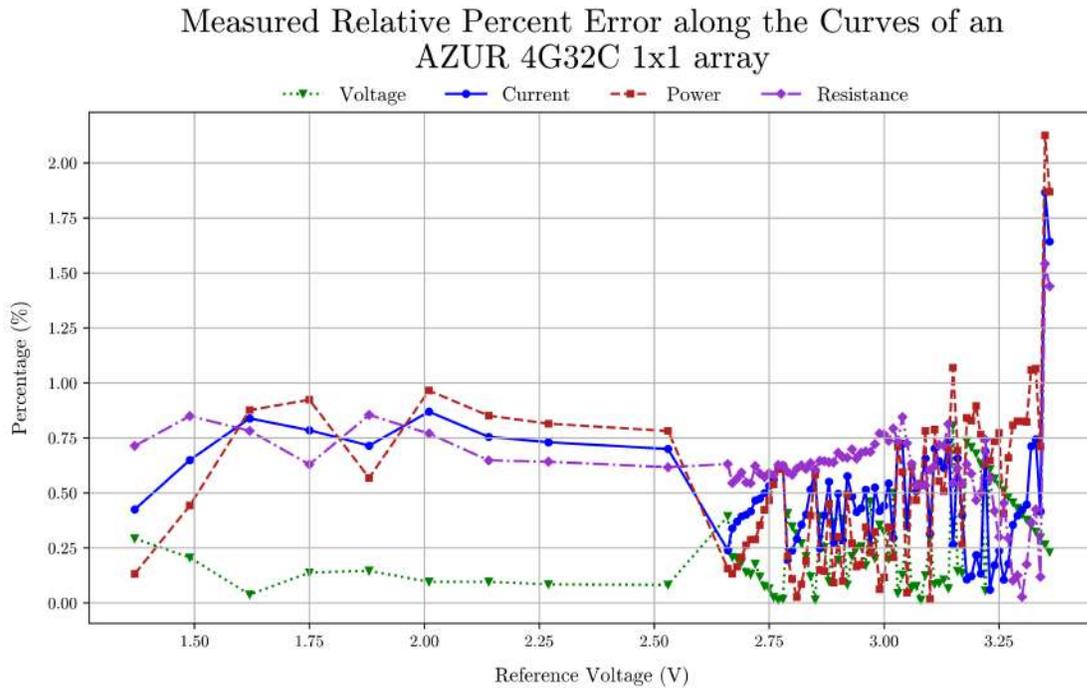


Figura F.4: Porcentajes de error en curvas caract. de un arreglo 1x1 (AZUR 4G32C).

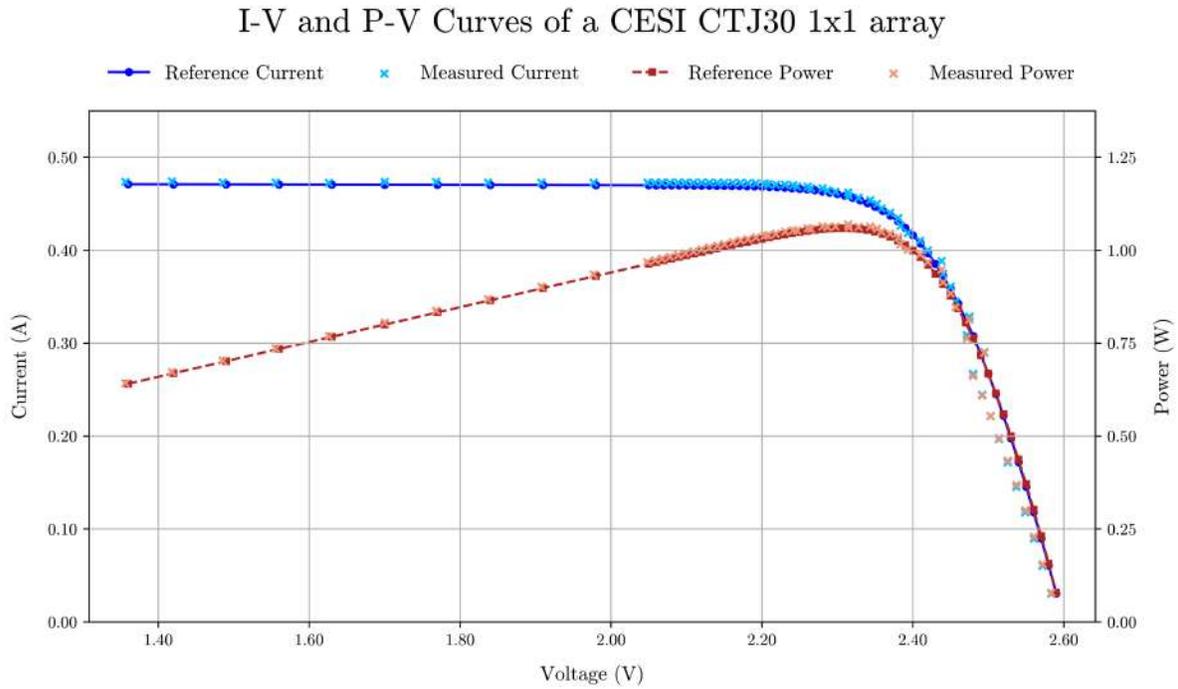


Figura F.5: Curvas I-V y P-V de un arreglo 1x1 (CESI CTJ30).

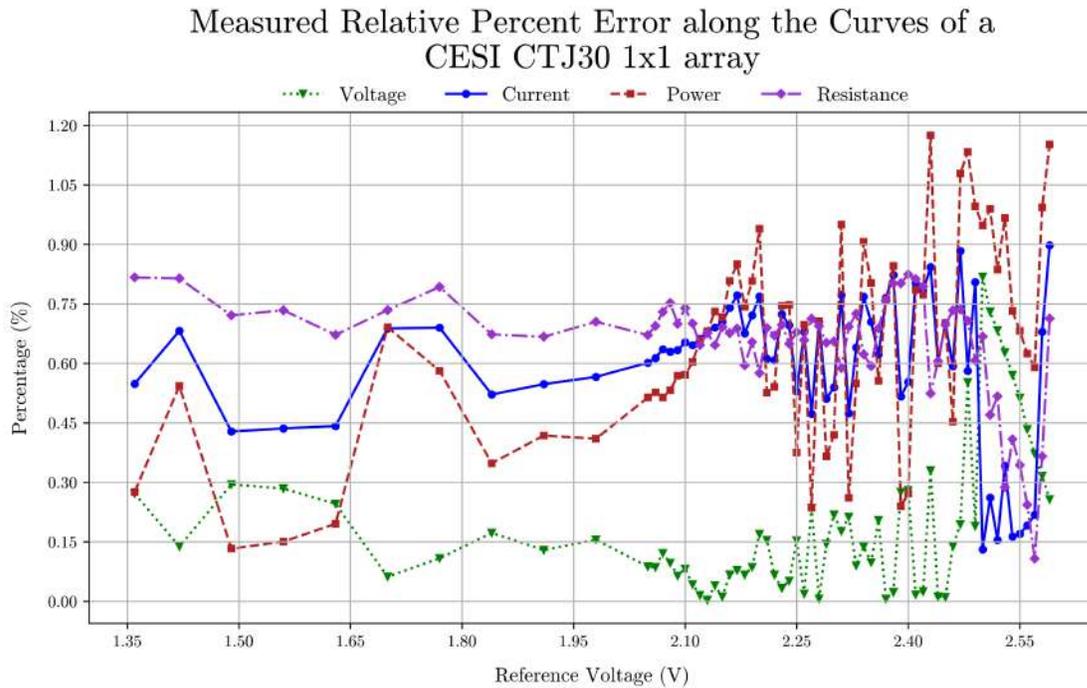


Figura F.6: Porcentajes de error en curvas características de un arreglo 1x1 (CESI CTJ30).

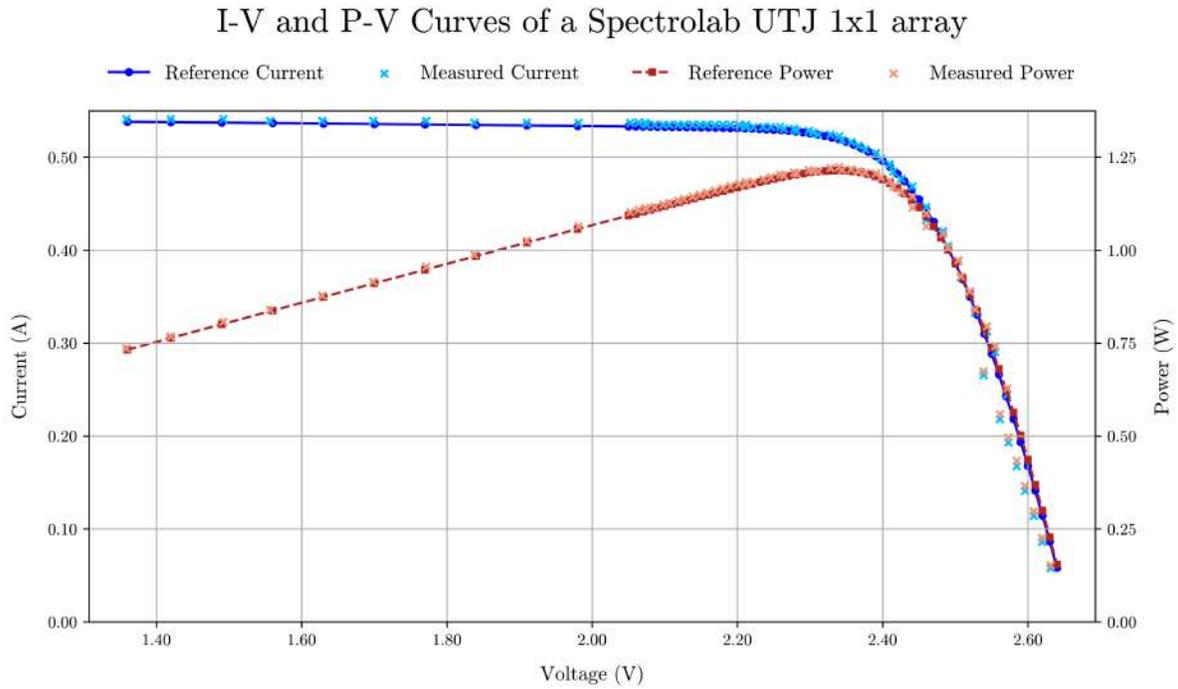


Figura F.7: Curvas I-V y P-V de un arreglo 1x1 (Spectrolab UTJ).

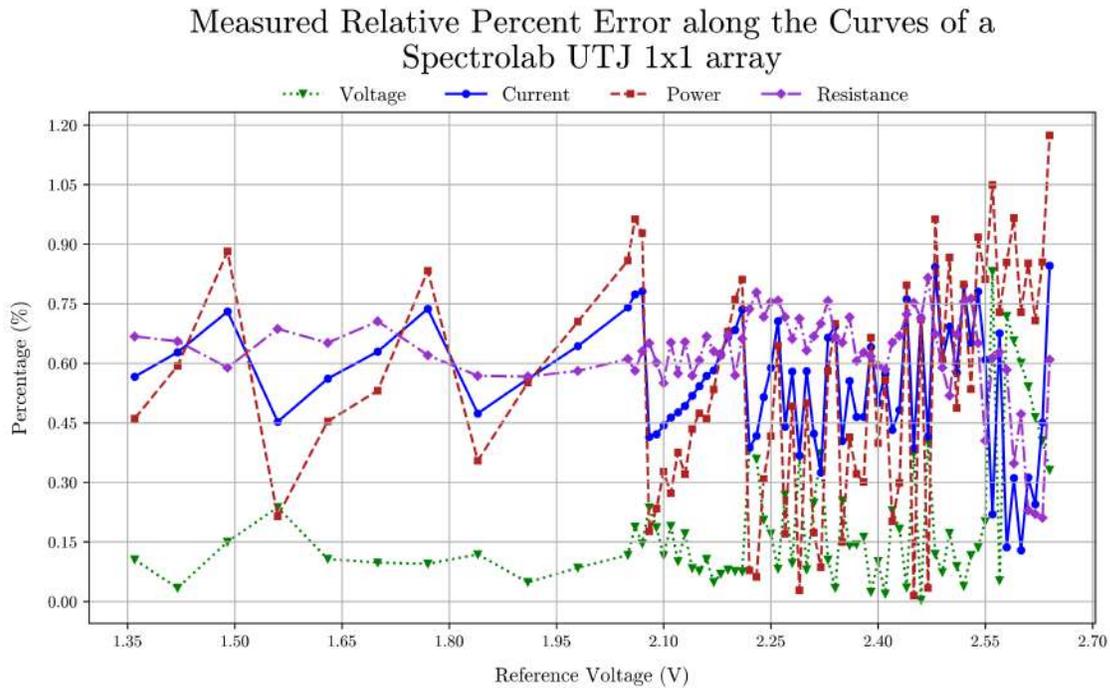


Figura F.8: Porcentajes de error en curvas caract. de un arreglo 1x1 (Spectrolab UTJ).

F.1.2. Arreglo 1x2

Ahora se procede con las curvas obtenidas para los arreglos 1x2 de cada panel. En el caso del AZUR 3G30A, Figura F.9, se puede observar un comportamiento idéntico al del 1x1, donde una vez pasado el punto de MPP, los ARPES aumentan considerablemente, debido a la oscilación de los modos de control de la fuente de poder. En el caso del AZUR 4G32C, Figura F.11, y el CESI CTJ30, Figura F.9, si bien sí existe dicha oscilación, se presenta un mejor rendimiento, manteniéndose siempre por debajo del ARPE de 1 %, inclusive conforme la corriente es muy cercana a 0. En el Spectrolab UTJ, Figura F.15, se observa un comportamiento similar que en los anteriores, pero este sí se ve más afectado por un ARPE desproporcionado en el último punto evaluado.

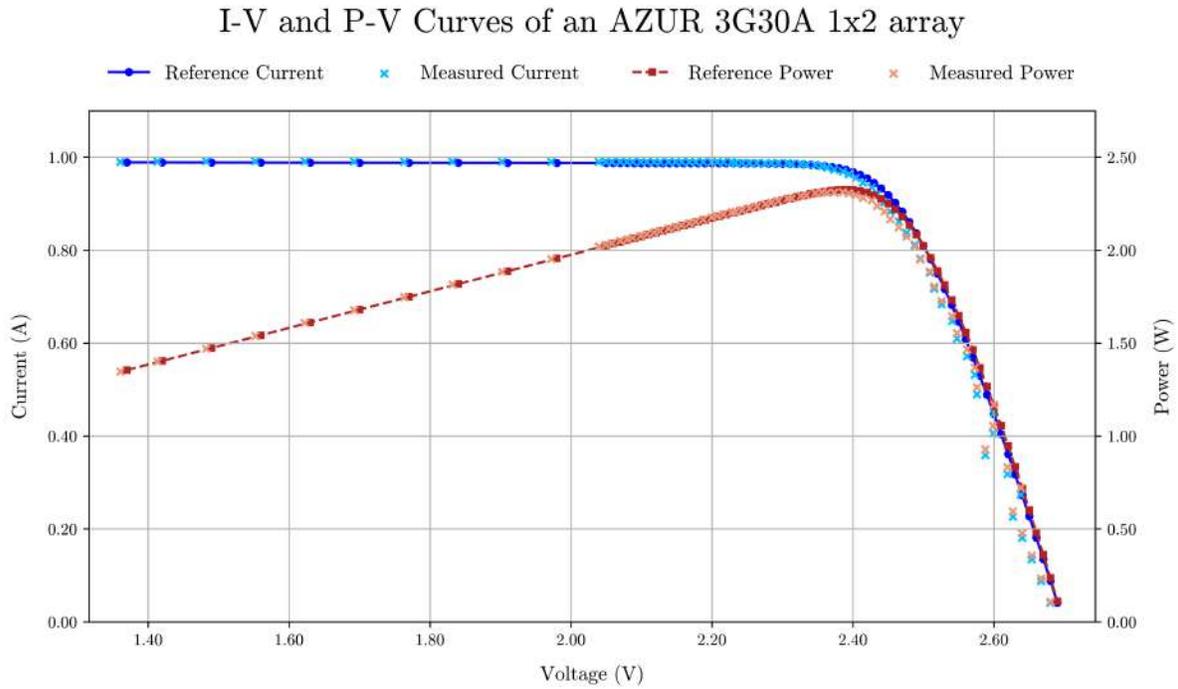


Figura F.9: Curvas I-V y P-V de un arreglo 1x2 (AZUR 3G30A).

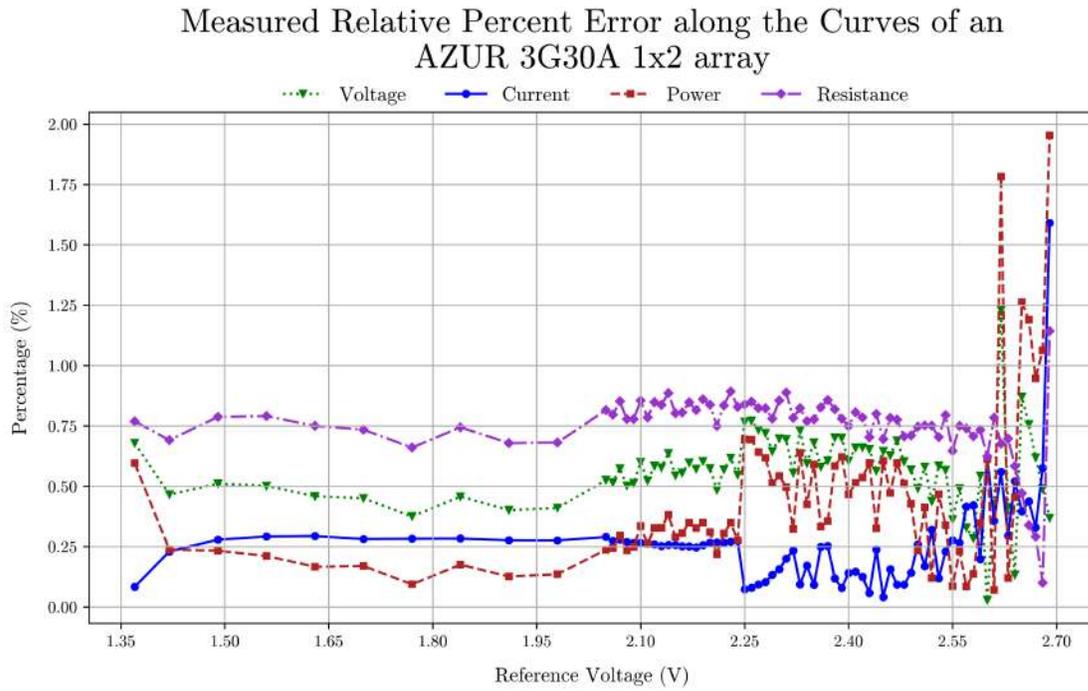


Figura F.10: Porcentajes de error en curvas características de un arreglo 1x2 (AZUR 3G30A).

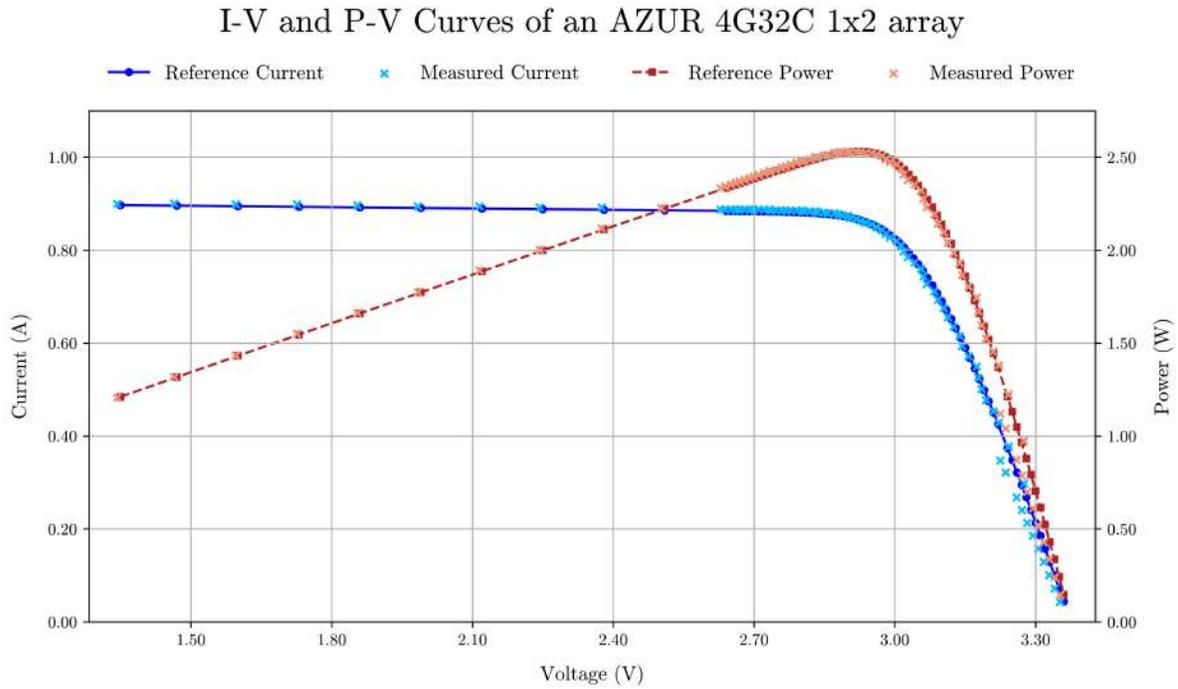


Figura F.11: Curvas I-V y P-V de un arreglo 1x2 (AZUR 4G32C).

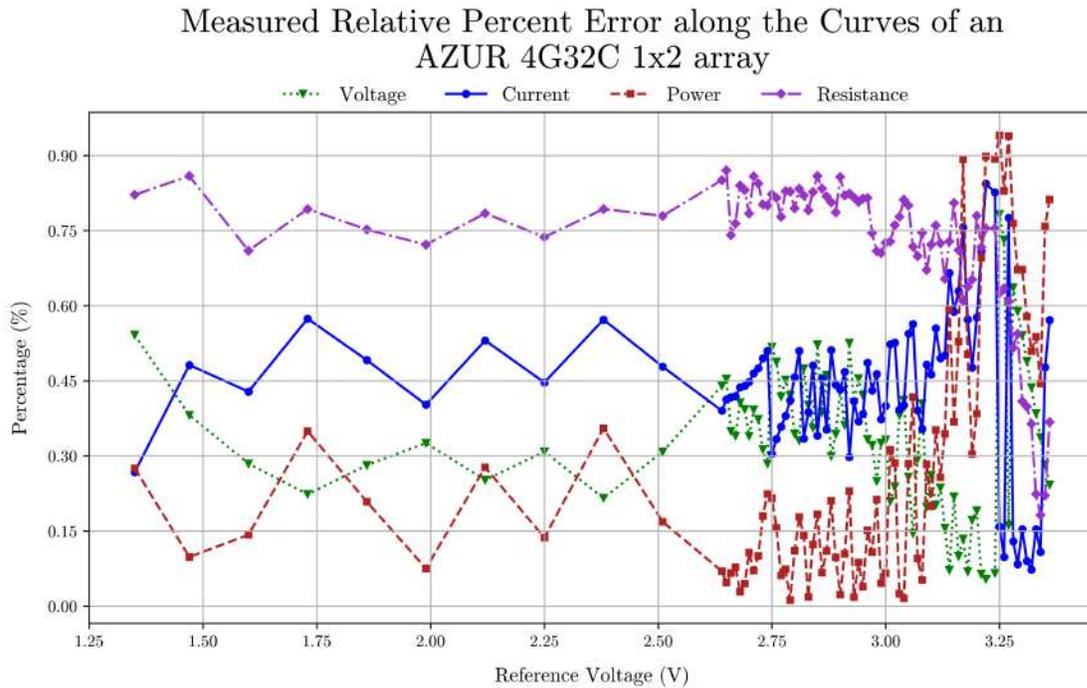


Figura F.12: Porcentajes de error en curvas caract. de un arreglo 1x2 (AZUR 4G32C).

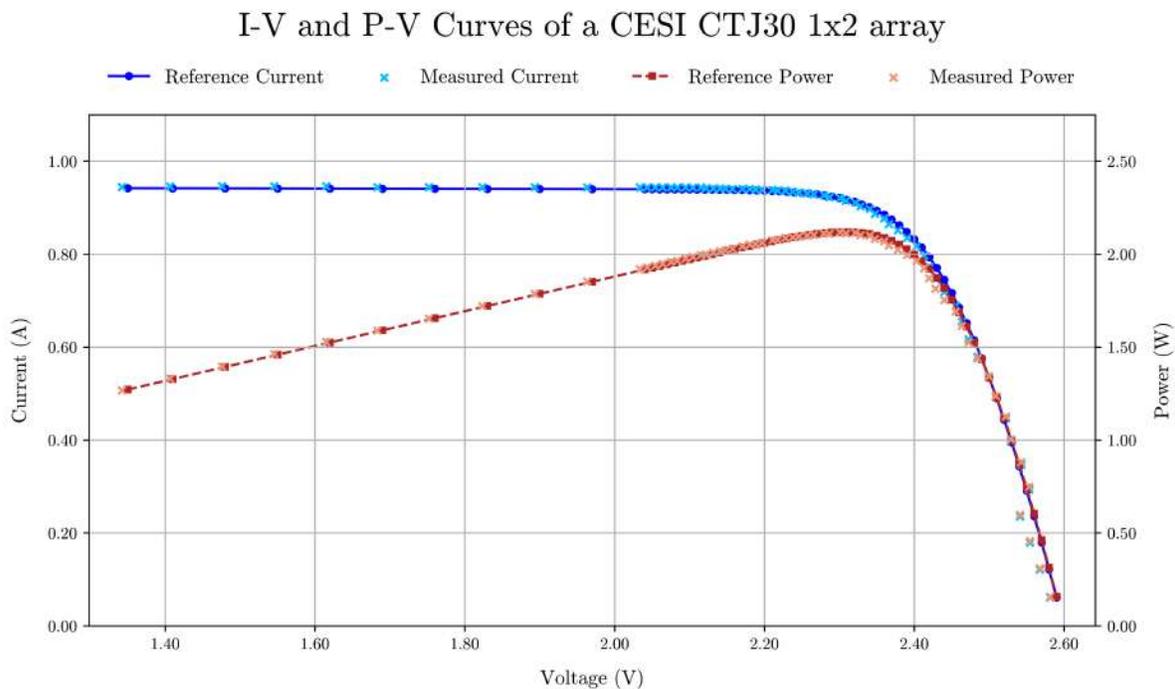


Figura F.13: Curvas I-V y P-V de un arreglo 1x2 (CESI CTJ30).

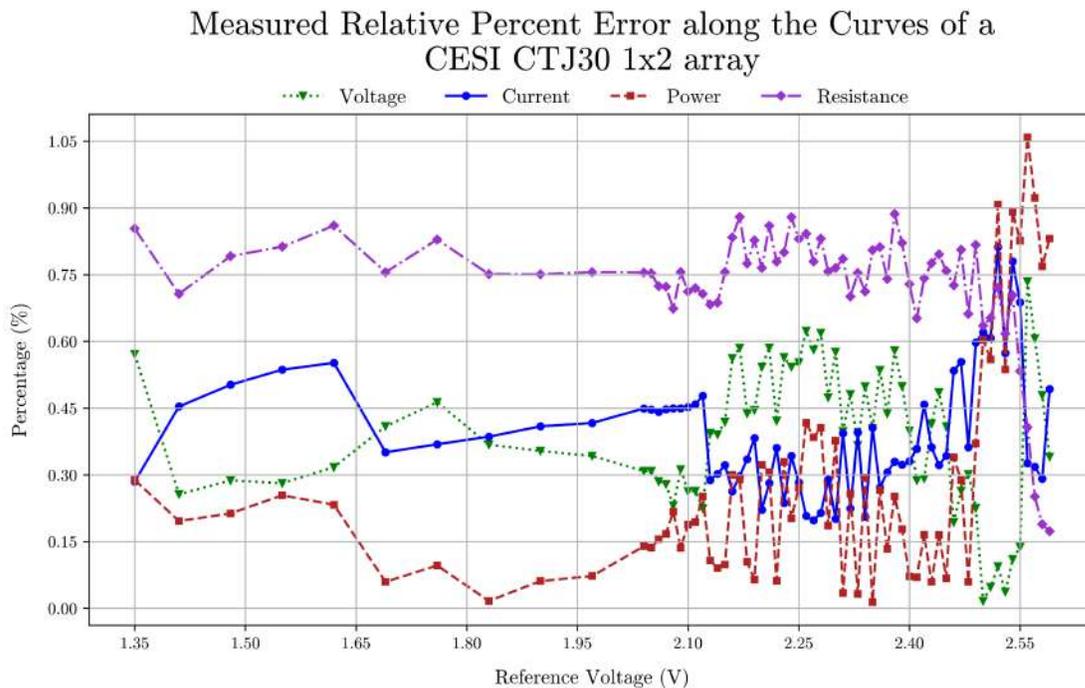


Figura F.14: Porcentajes de error en curvas características de un arreglo 1x2 (CESI CTJ30).

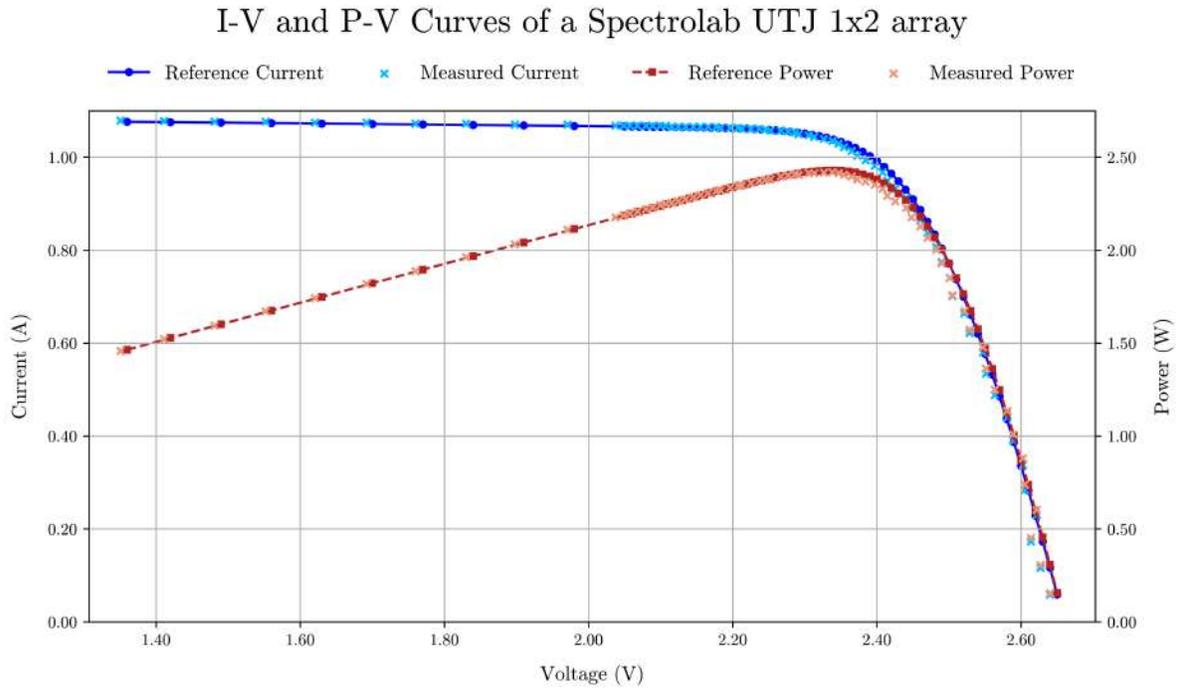


Figura F.15: Curvas I-V y P-V de un arreglo 1x2 (Spectrolab UTJ).

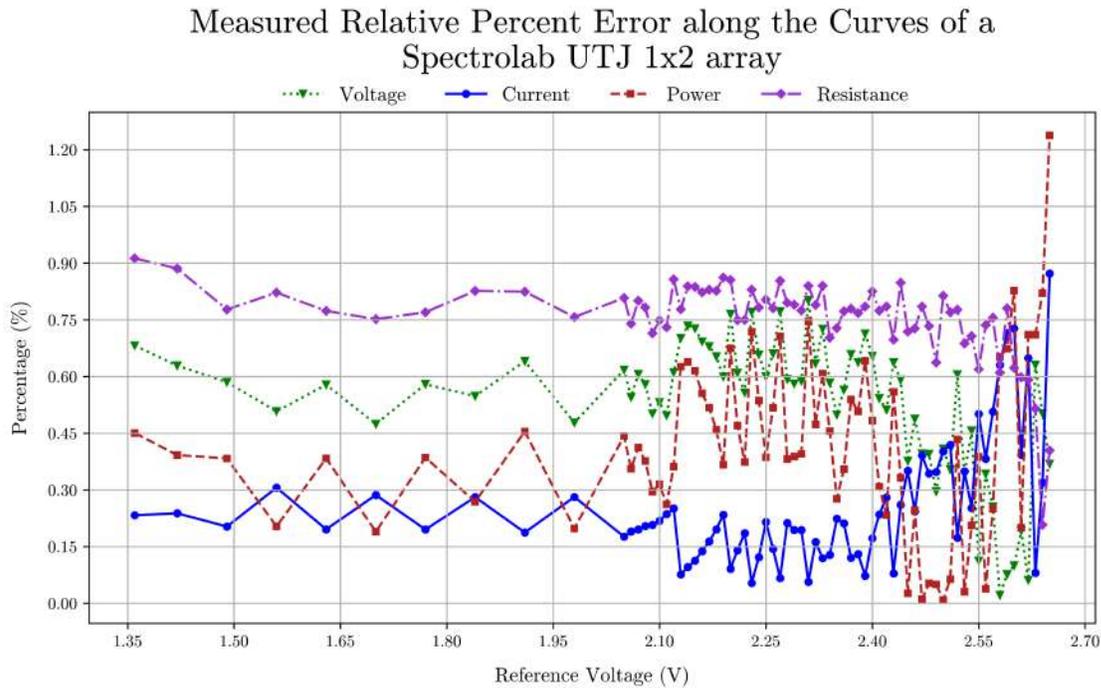


Figura F.16: Porcentajes de error en curvas caract. de un arreglo 1x2 (Spectrolab UTJ).

F.1.3. Arreglo 1x6

Ahora se procede con las curvas obtenidas para los arreglos 1x6 de cada panel. En el caso del AZUR 3G30A, Figura F.17, se puede observar que las oscilaciones empiezan incluso antes del punto de MPP, lo que se puede atribuir a la alta corriente con respecto a la tensión, como fue discutido anteriormente. En este caso los ARPEs no aumentan desproporcionadamente en los últimos puntos. En el caso del AZUR 4G32C, Figura F.19, el CESI CTJ30, Figura F.17, y el Spectrolab UTJ, Figura F.23, también es consistente la oscilación incrementada antes del MPP, y tampoco se ven afectados por los ARPEs en los últimos puntos.

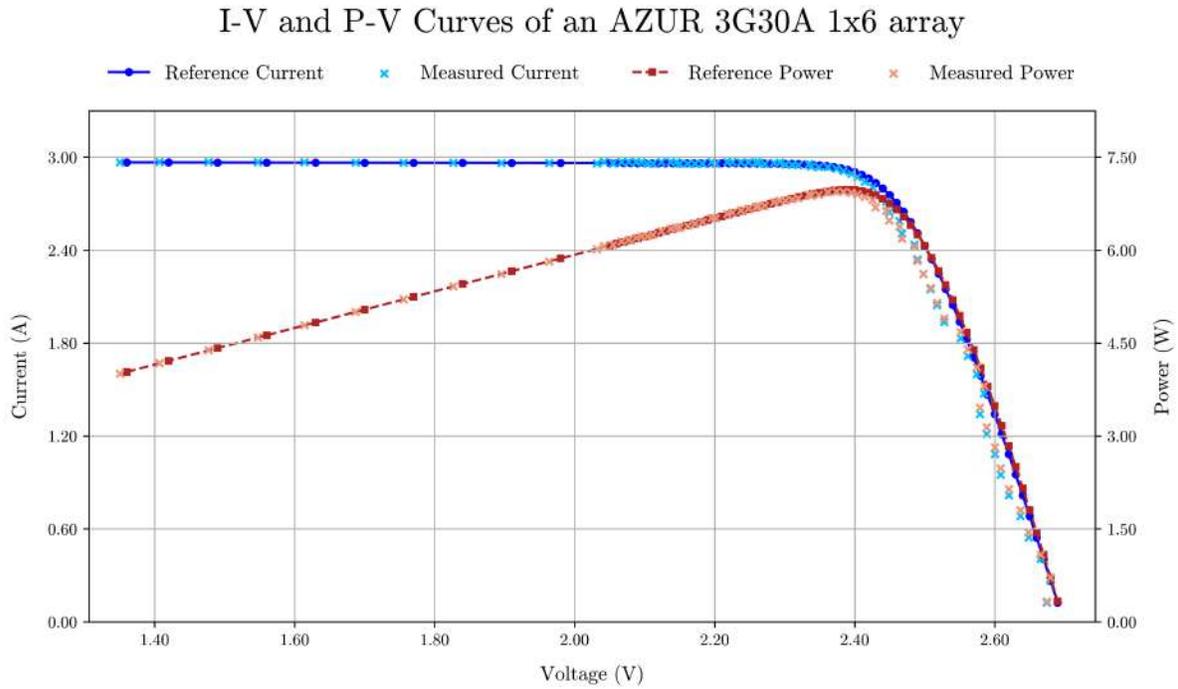


Figura F.17: Curvas I-V y P-V de un arreglo 1x6 (AZUR 3G30A).

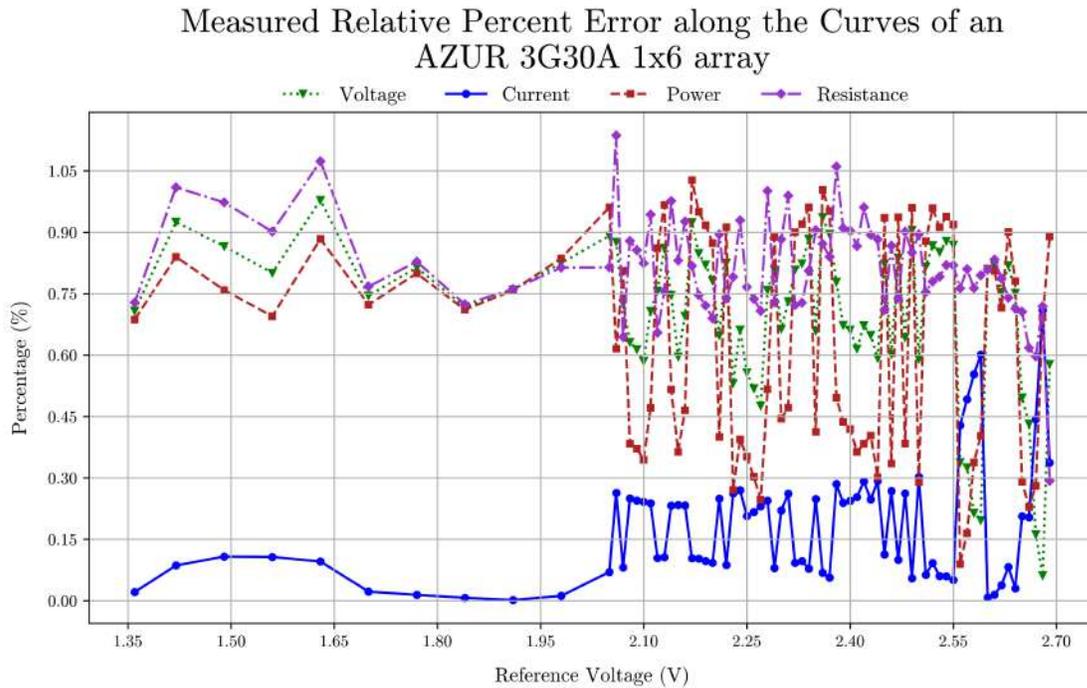


Figura F.18: Porcentajes de error en curvas características de un arreglo 1x6 (AZUR 3G30A).

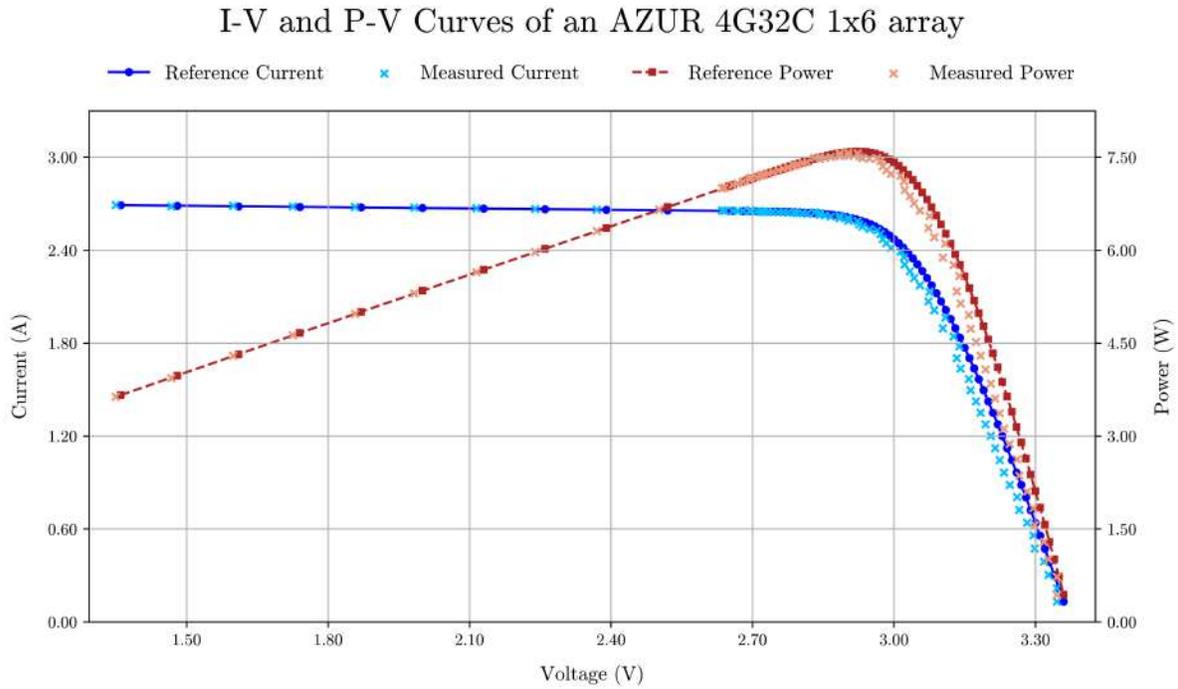


Figura F.19: Curvas I-V y P-V de un arreglo 1x6 (AZUR 4G32C).

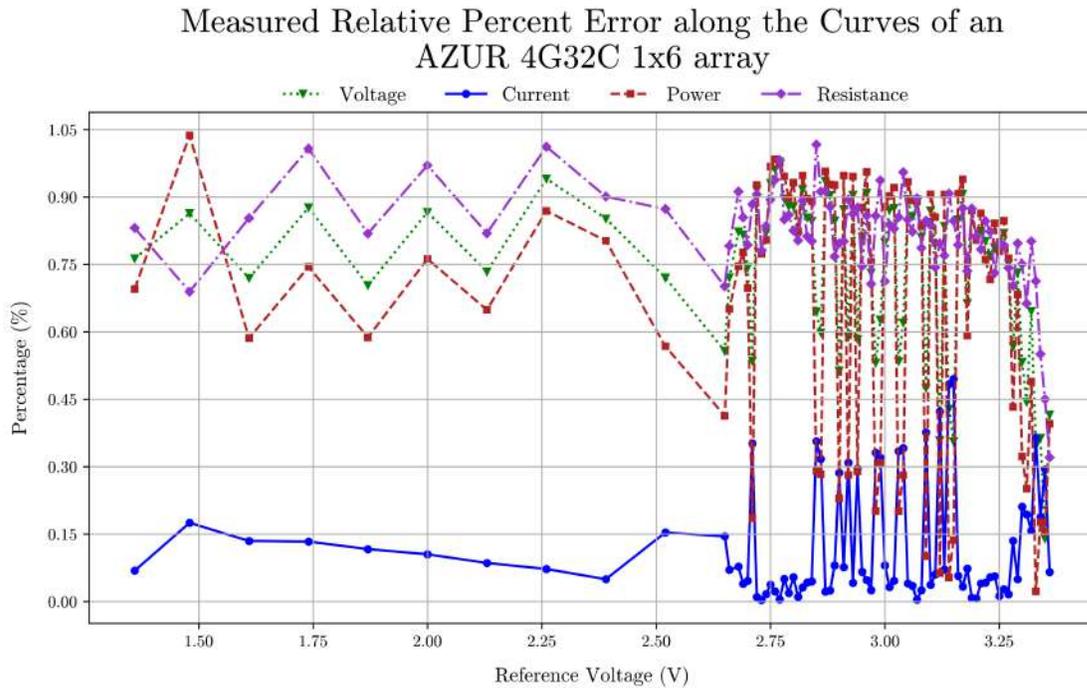


Figura F.20: Porcentajes de error en curvas caract. de un arreglo 1x6 (AZUR 4G32C).

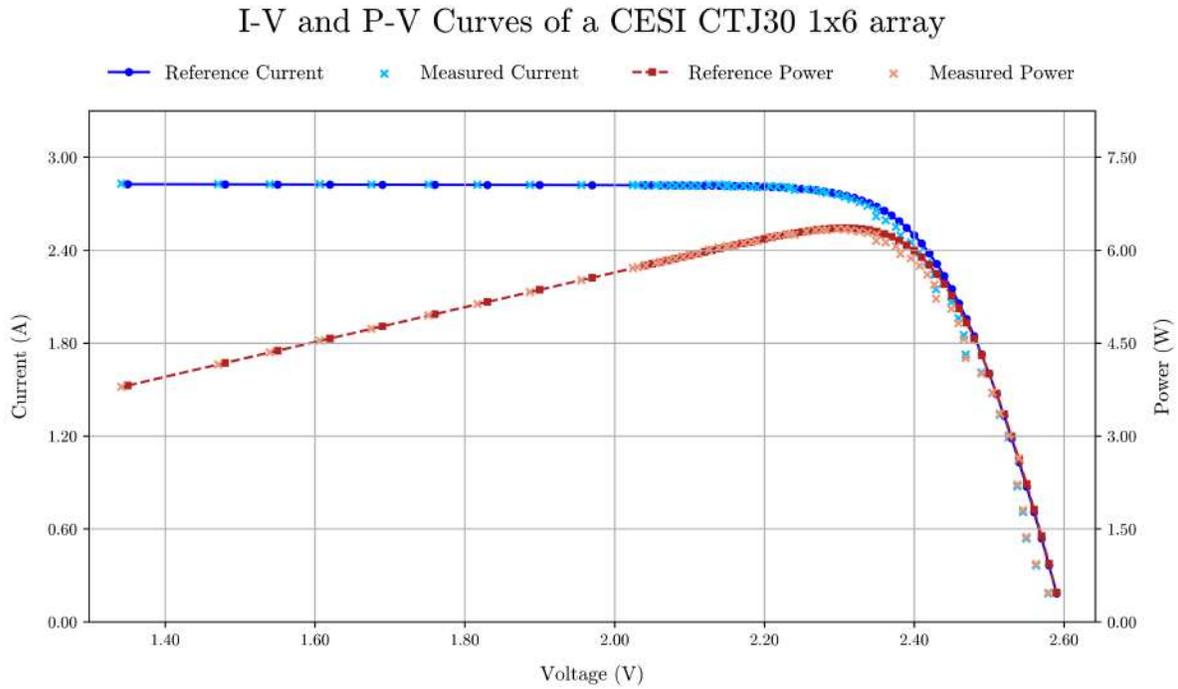


Figura F.21: Curvas I-V y P-V de un arreglo 1x6 (CESI CTJ30).

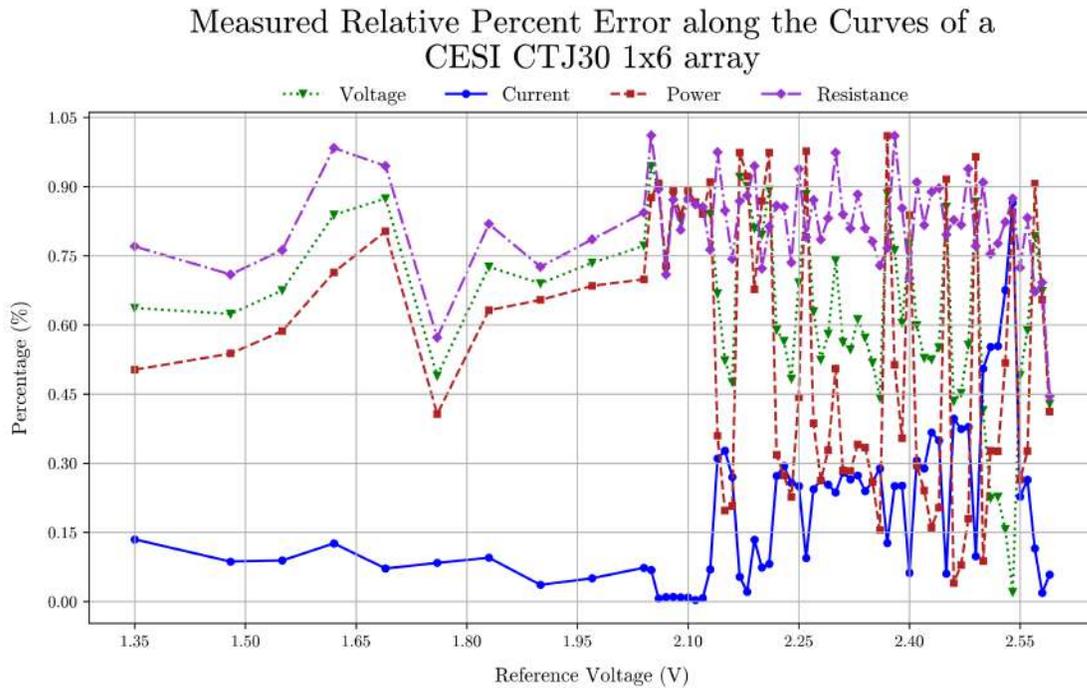


Figura F.22: Porcentajes de error en curvas características de un arreglo 1x6 (CESI CTJ30).

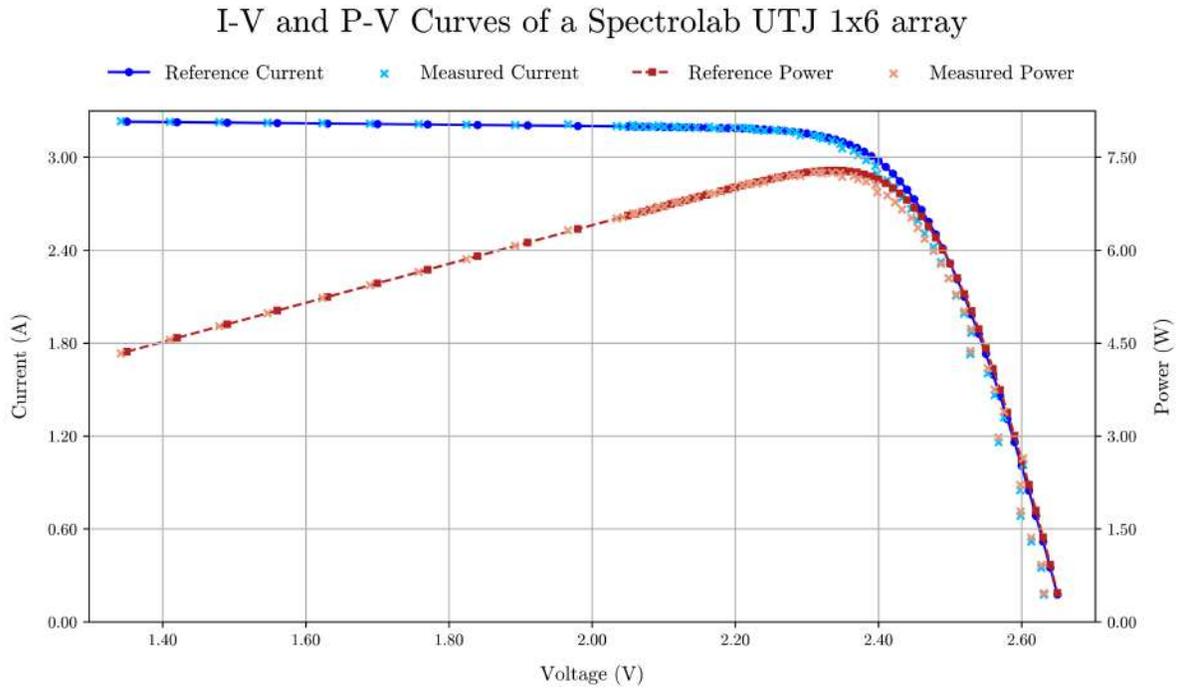


Figura F.23: Curvas I-V y P-V de un arreglo 1x6 (Spectrolab UTJ).

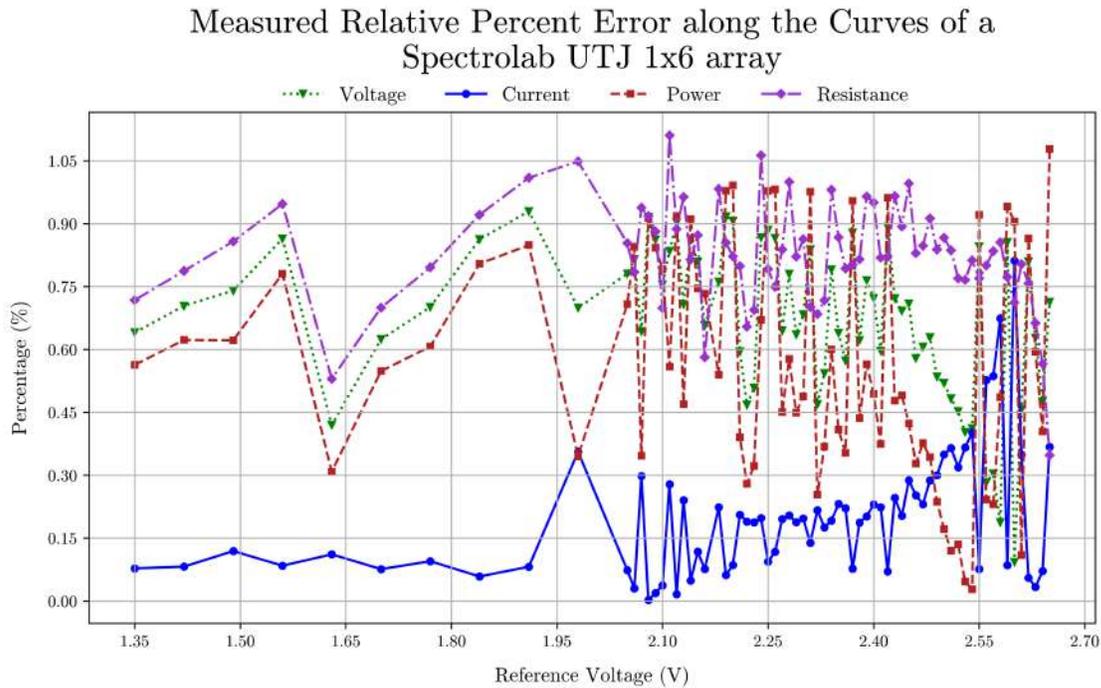


Figura F.24: Porcentajes de error en curvas caract. de un arreglo 1x6 (Spectrolab UTJ).

F.1.4. Arreglo 2x1

Ahora se procede con las curvas obtenidas para los arreglos 2x1 de cada panel. En el caso del AZUR 3G30A, Figura F.25, su rendimiento es uno de los mejores por el momento en las primeras porciones de la curva, manteniéndose por debajo del ARPE igual a 0.9%. No obstante, se ve afectado por un ARPE desproporcionado en el último punto. En el caso del AZUR 4G32C, Figura F.27, se puede observar que en tres puntos al inicio de la curva se presentan errores considerablemente altos, lo cual es inconsistente con el comportamiento esperado hasta ahora. Luego del MPP se presenta la típica oscilación en el modo de control pero se mantiene en rangos aceptables. En los últimos puntos se ve bastante afectado por ARPEs elevados conforme se aproxima a 0 la corriente. En el caso de el CESI CTJ30, Figura F.25, y el Spectrolab UTJ, Figura F.31, el rendimiento es muy bueno en toda la curva, manteniendo el ARPE por debajo del 0.9%.

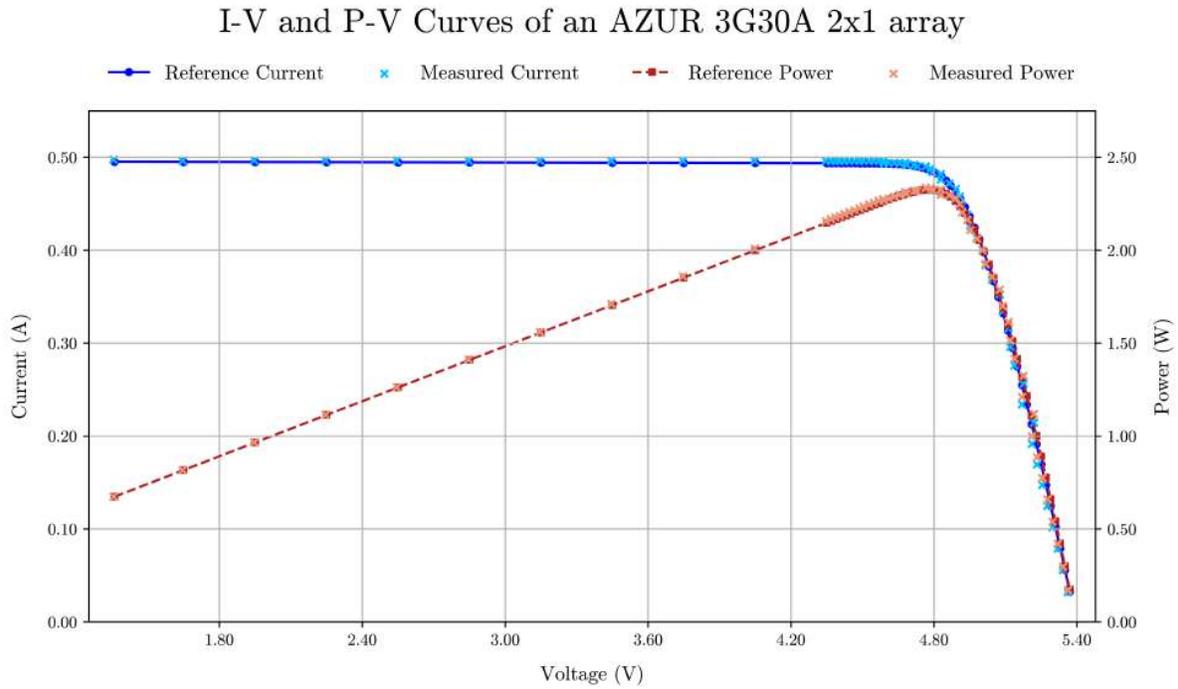


Figura F.25: Curvas I-V y P-V de un arreglo 2x1 (AZUR 3G30A).

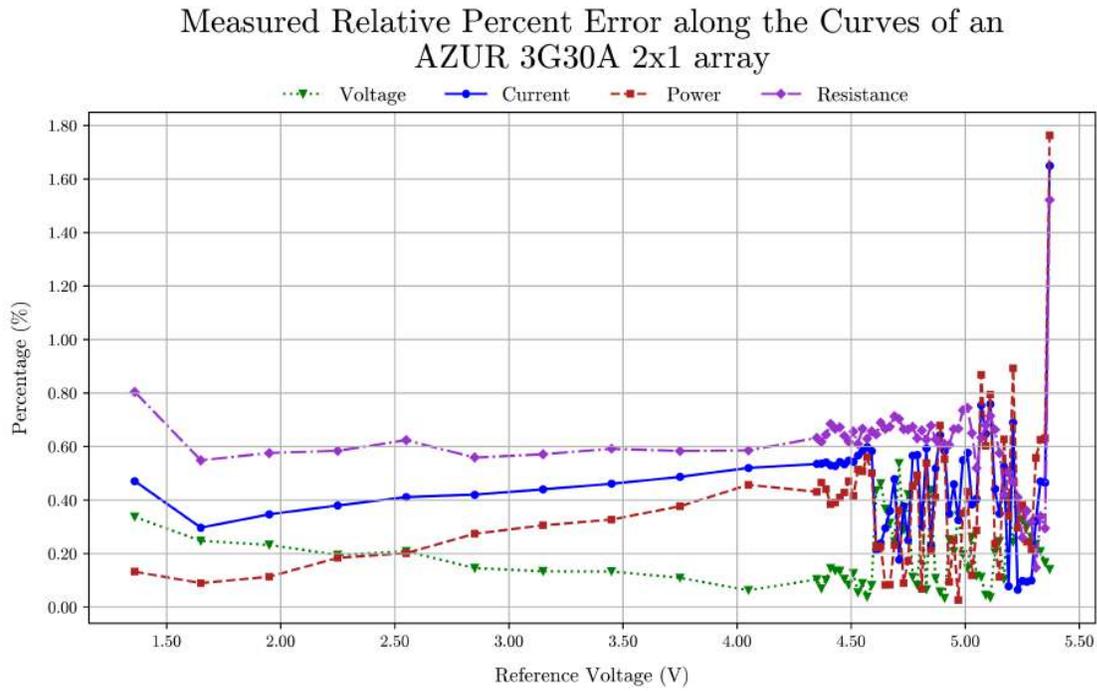


Figura F.26: Porcentajes de error en curvas características de un arreglo 2x1 (AZUR 3G30A).

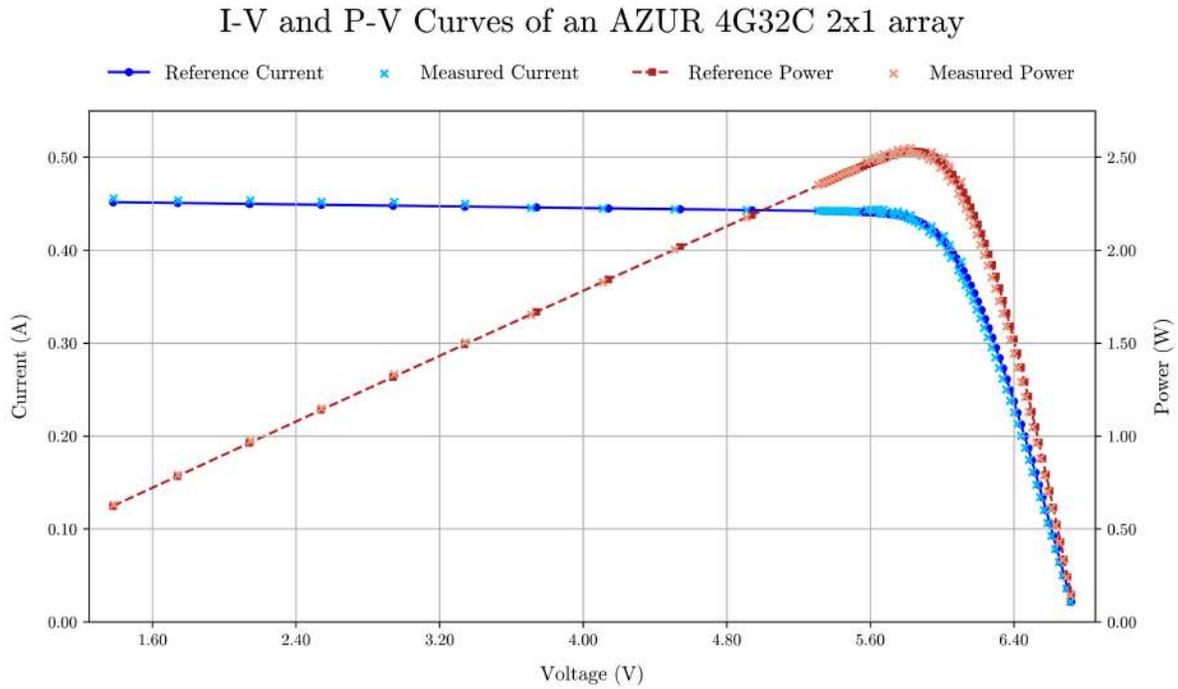


Figura F.27: Curvas I-V y P-V de un arreglo 2x1 (AZUR 4G32C).

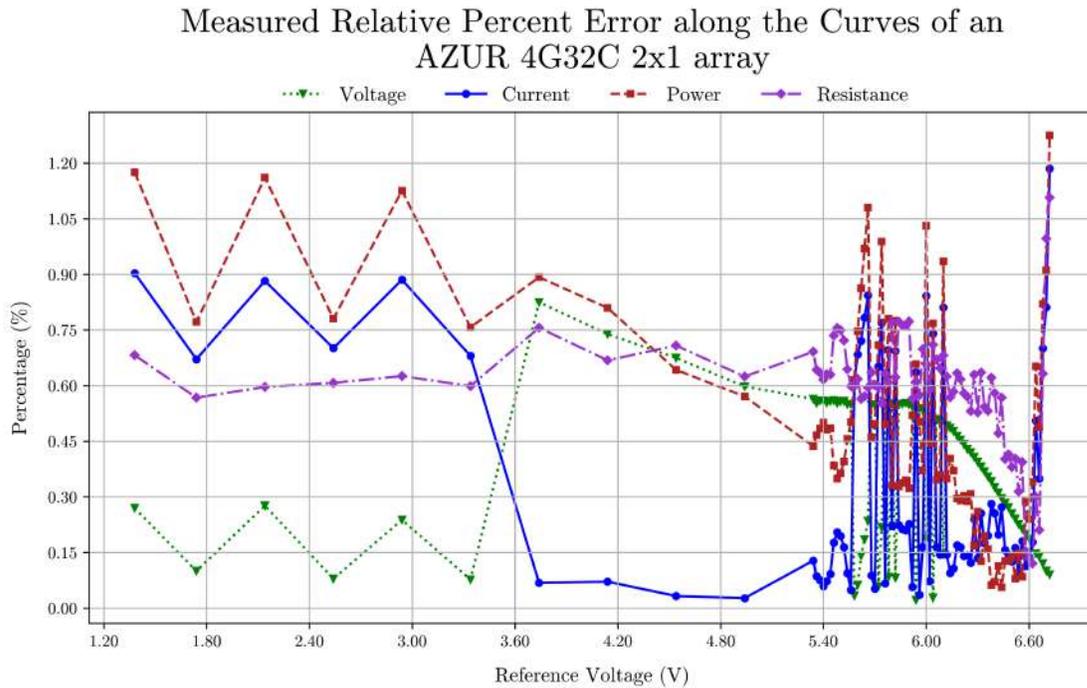


Figura F.28: Porcentajes de error en curvas caract. de un arreglo 2x1 (AZUR 4G32C).

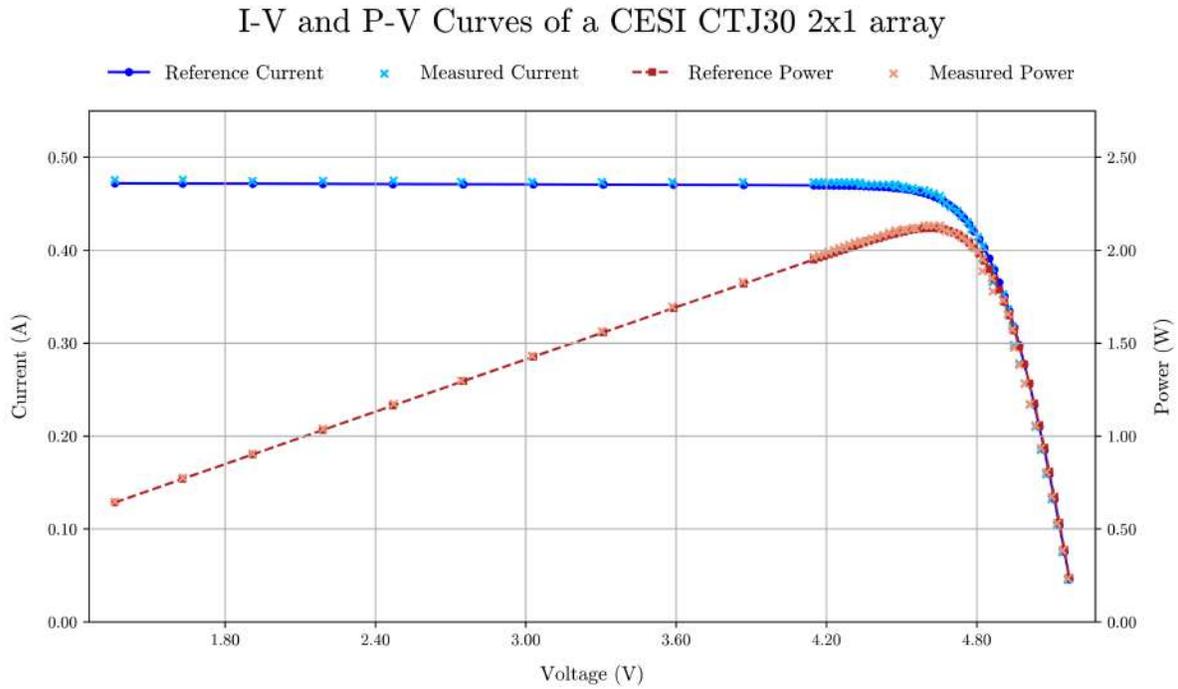


Figura F.29: Curvas I-V y P-V de un arreglo 2x1 (CESI CTJ30).

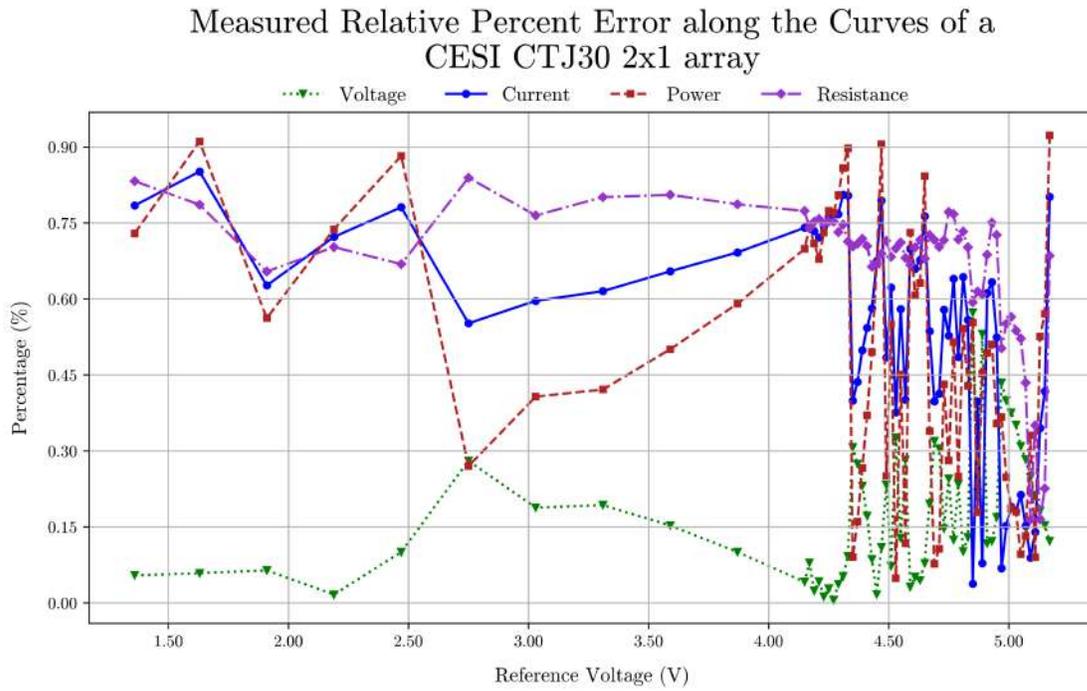


Figura F.30: Porcentajes de error en curvas características de un arreglo 2x1 (CESI CTJ30).

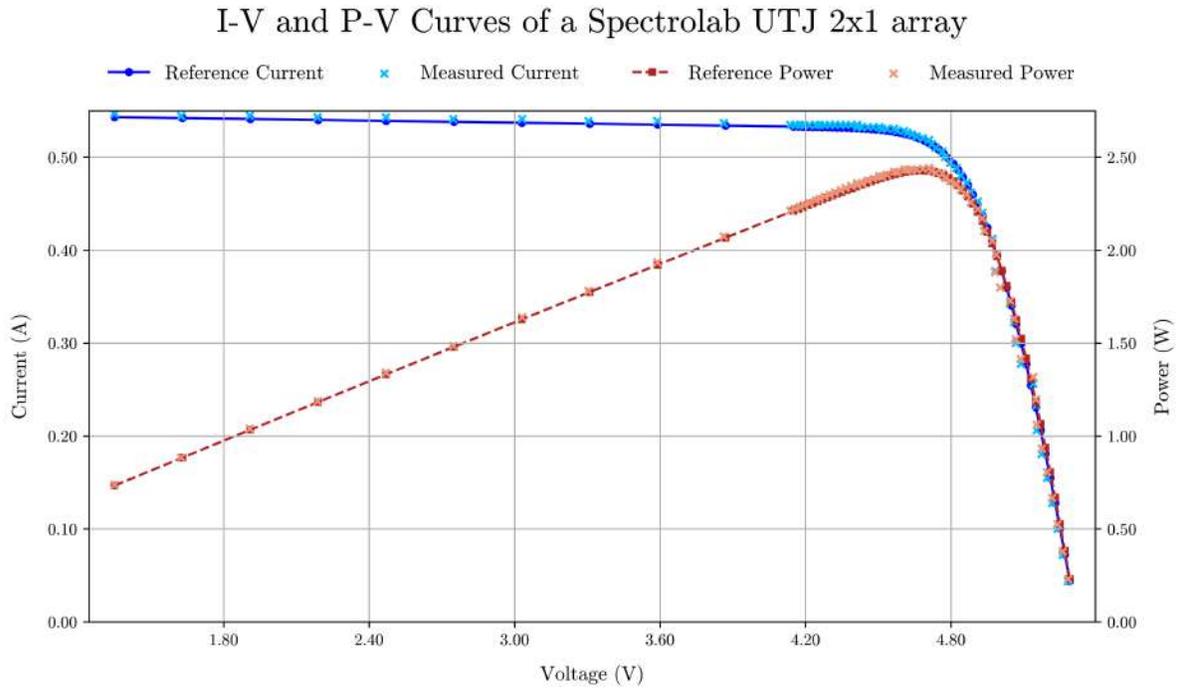


Figura F.31: Curvas I-V y P-V de un arreglo 2x1 (Spectrolab UTJ).

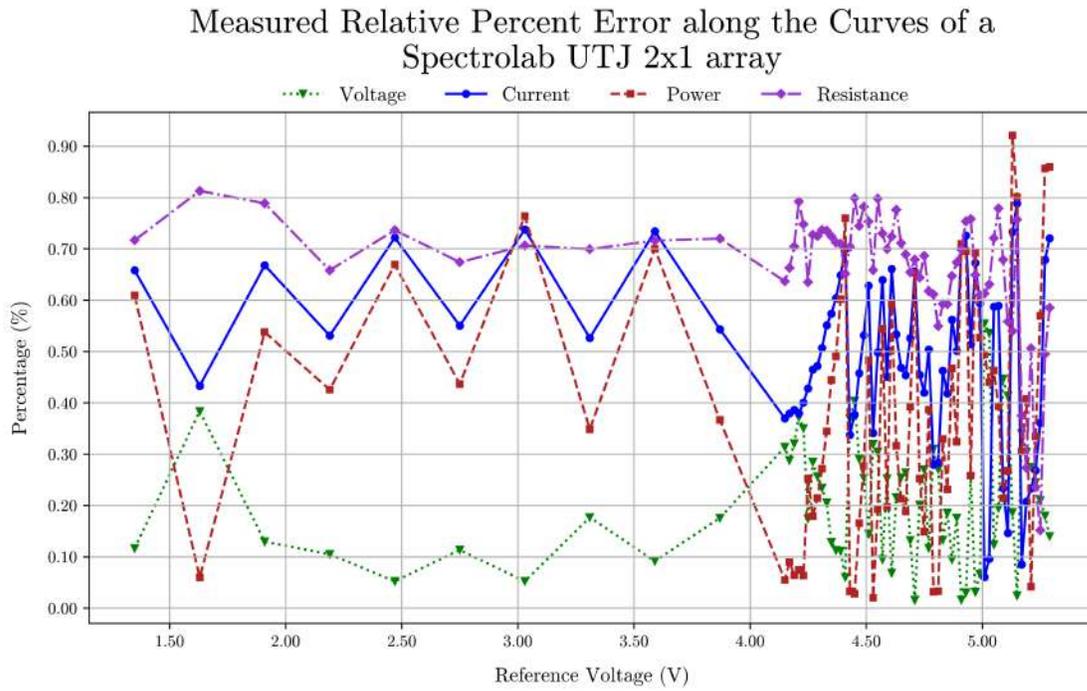


Figura F.32: Porcentajes de error en curvas caract. de un arreglo 2x1 (Spectrolab UTJ).

F.1.5. Arreglo 2x3

Ahora se procede con las curvas obtenidas para los arreglos 2x3 de cada panel. En el caso del AZUR 3G30A, Figura F.33, AZUR 4G32C, Figura F.35, y CESI CTJ30, Figura F.33, su rendimiento es bastante bueno, manteniendo ARPEs inferiores al a 0.9%. El Spectrolab UTJ, Figura F.39, presento ARPEs por arriba del 1% en puntos cercanos al MPP, pero luego se estabilizó.

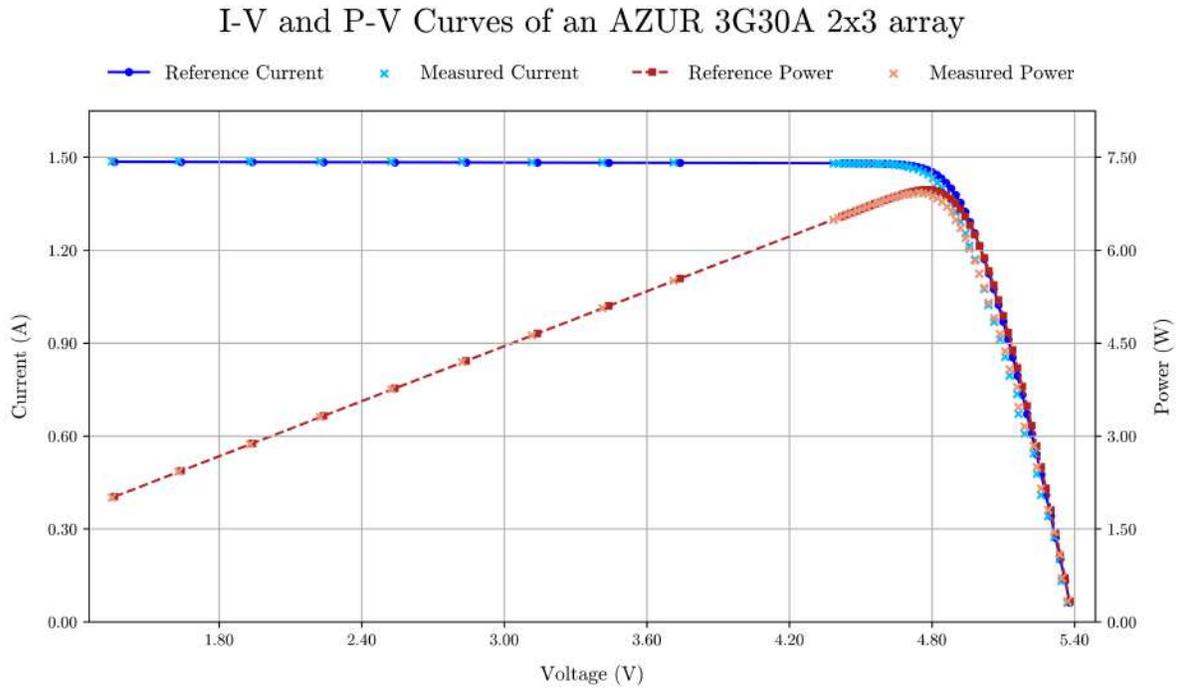


Figura F.33: Curvas I-V y P-V de un arreglo 2x3 (AZUR 3G30A).

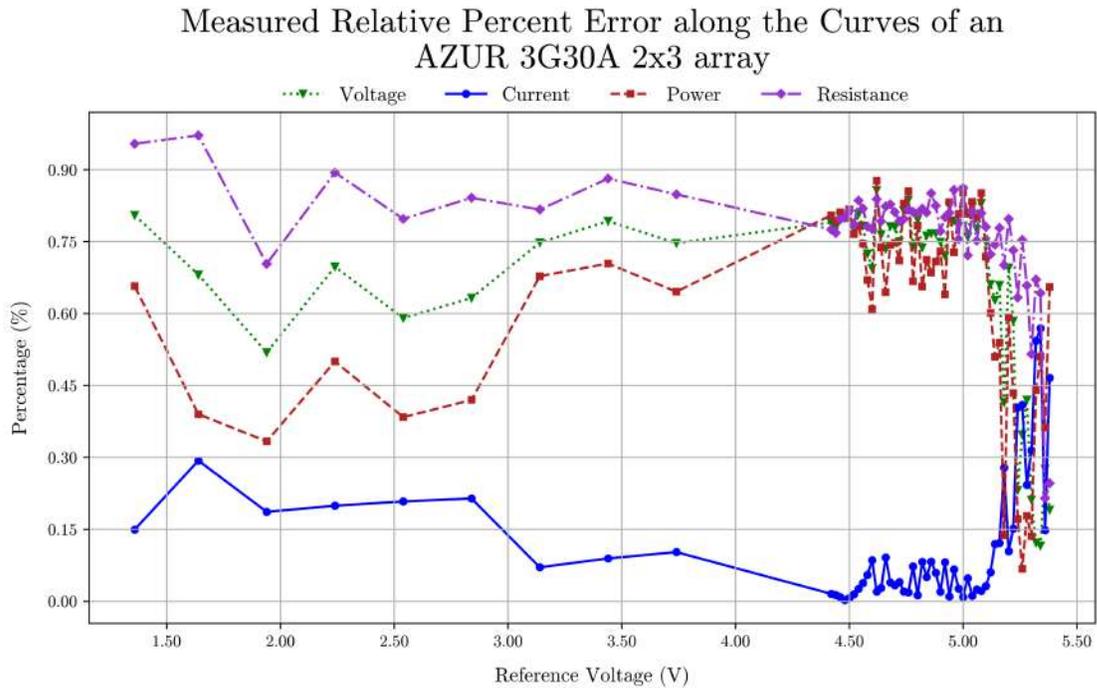


Figura F.34: Porcentajes de error en curvas características de un arreglo 2x3 (AZUR 3G30A).

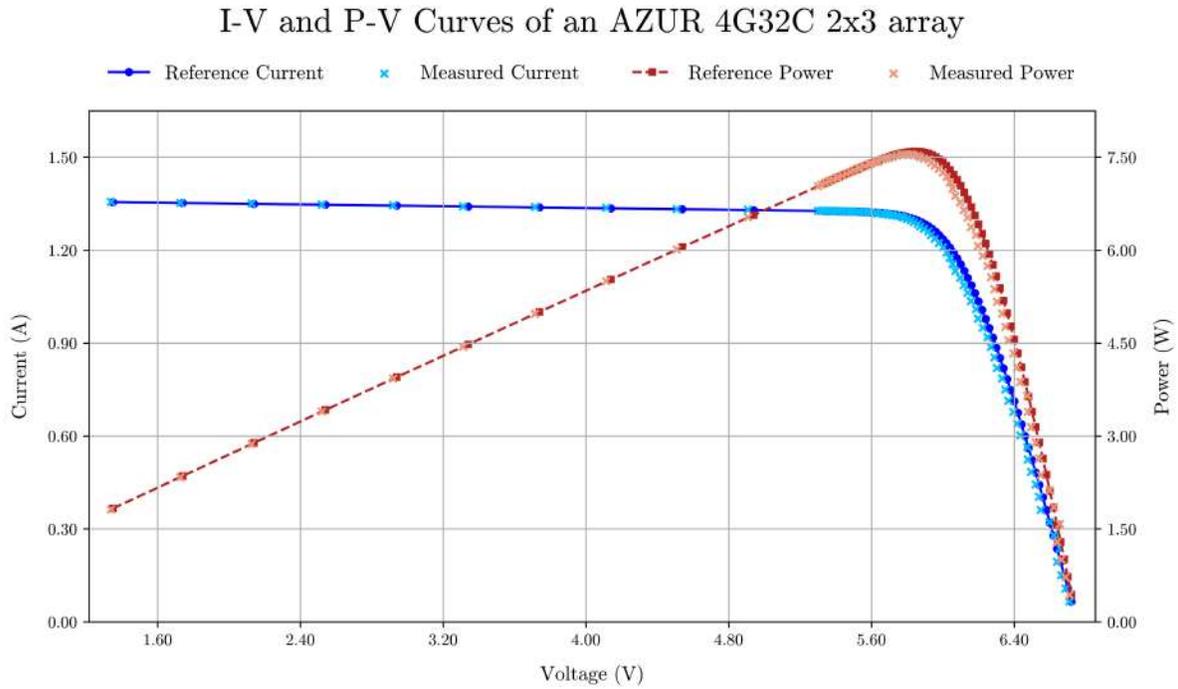


Figura F.35: Curvas I-V y P-V de un arreglo 2x3 (AZUR 4G32C).

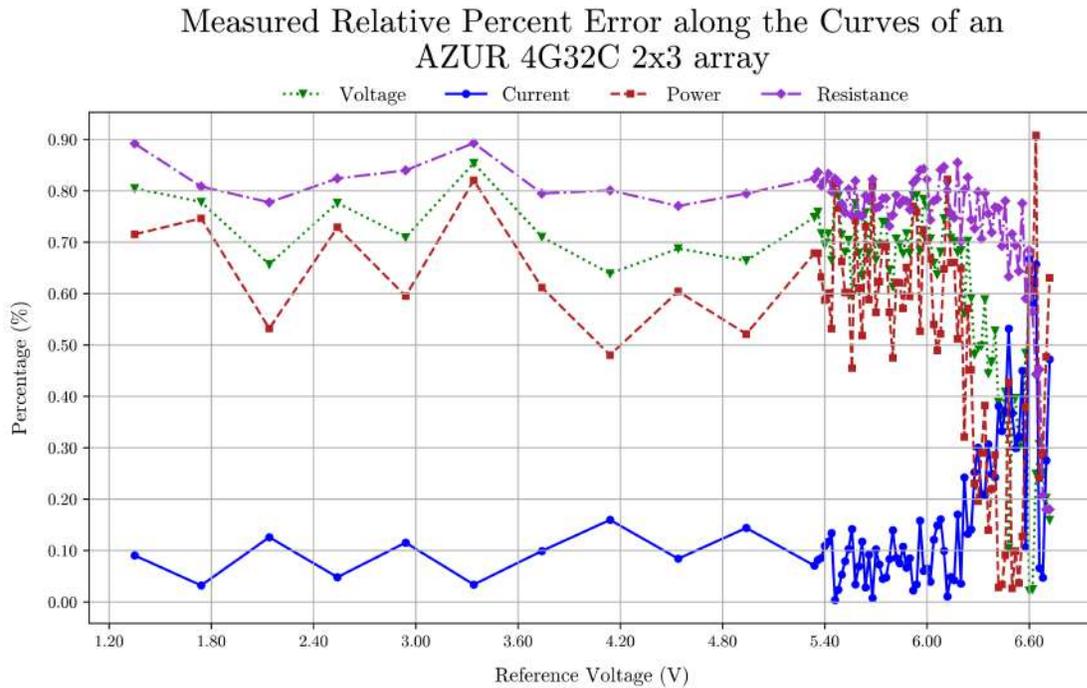


Figura F.36: Porcentajes de error en curvas caract. de un arreglo 2x3 (AZUR 4G32C).

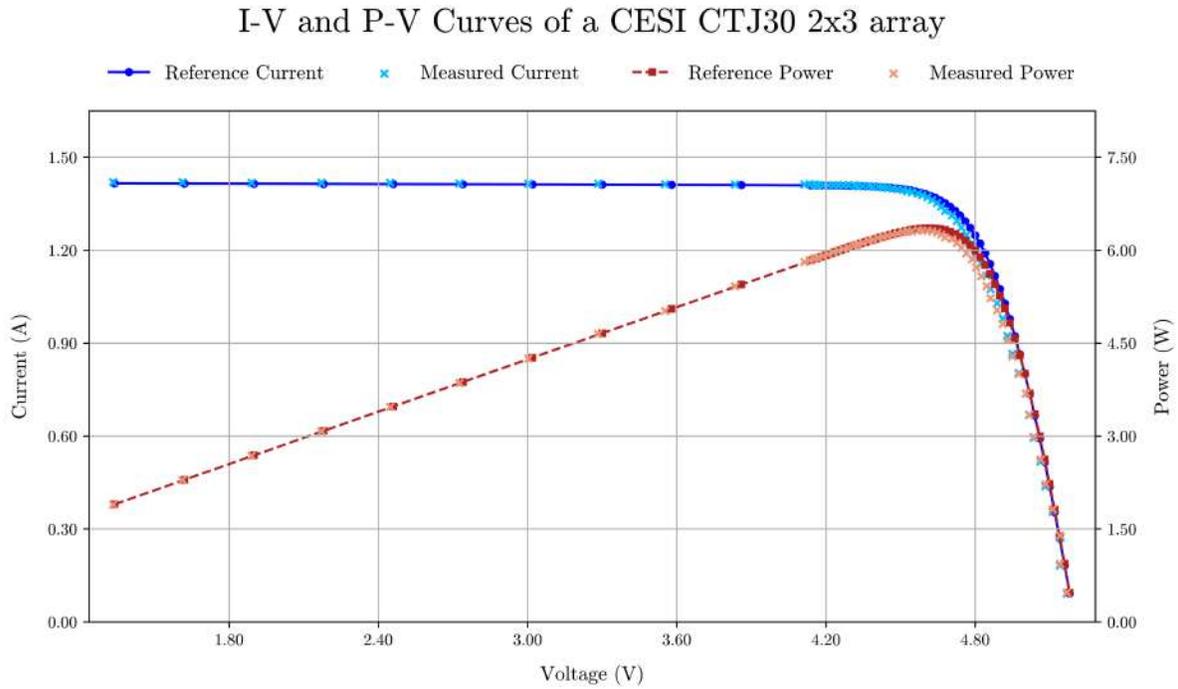


Figura F.37: Curvas I-V y P-V de un arreglo 2x3 (CESI CTJ30).

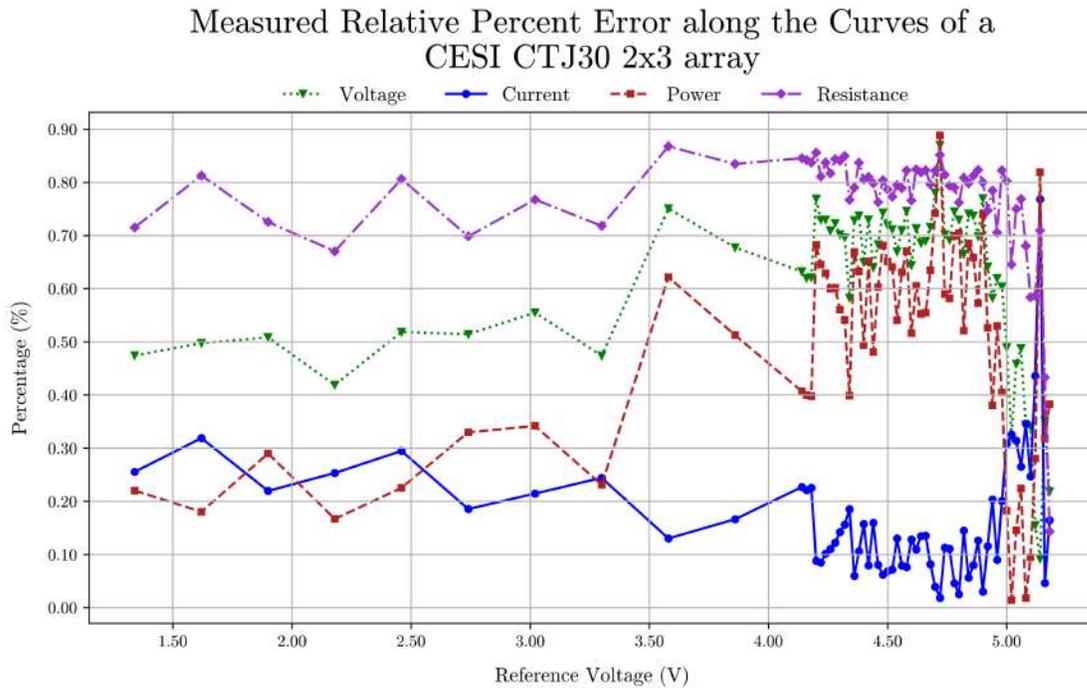


Figura F.38: Porcentajes de error en curvas características de un arreglo 2x3 (CESI CTJ30).

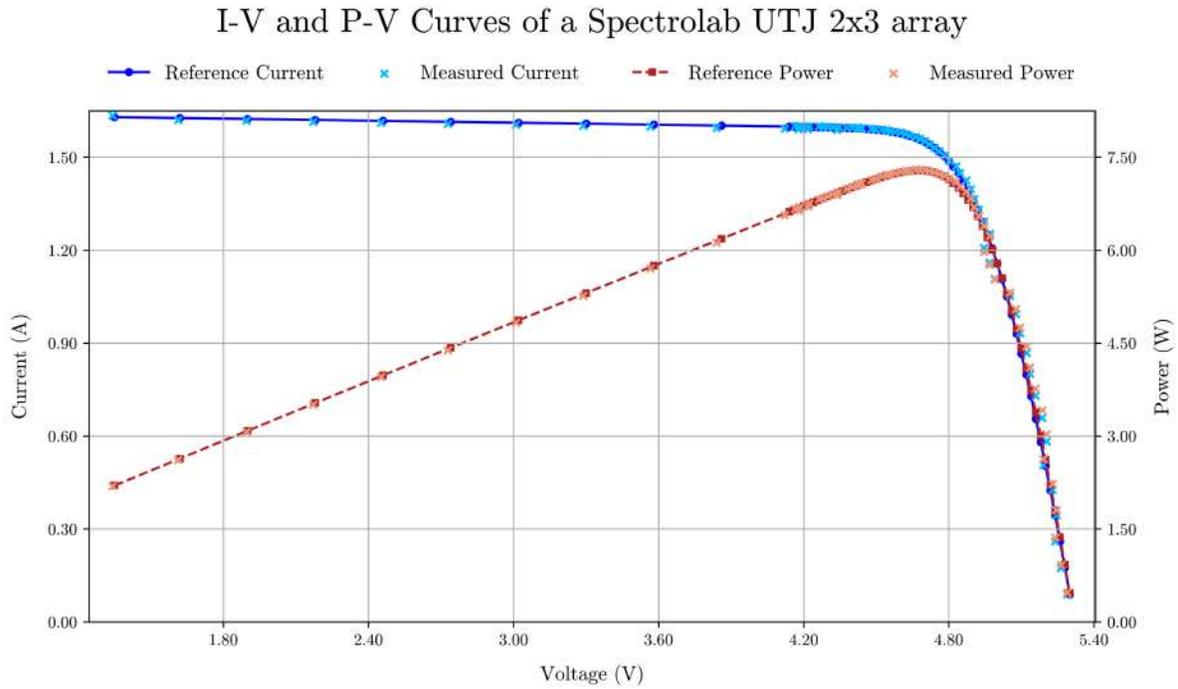


Figura F.39: Curvas I-V y P-V de un arreglo 2x3 (Spectrolab UTJ).

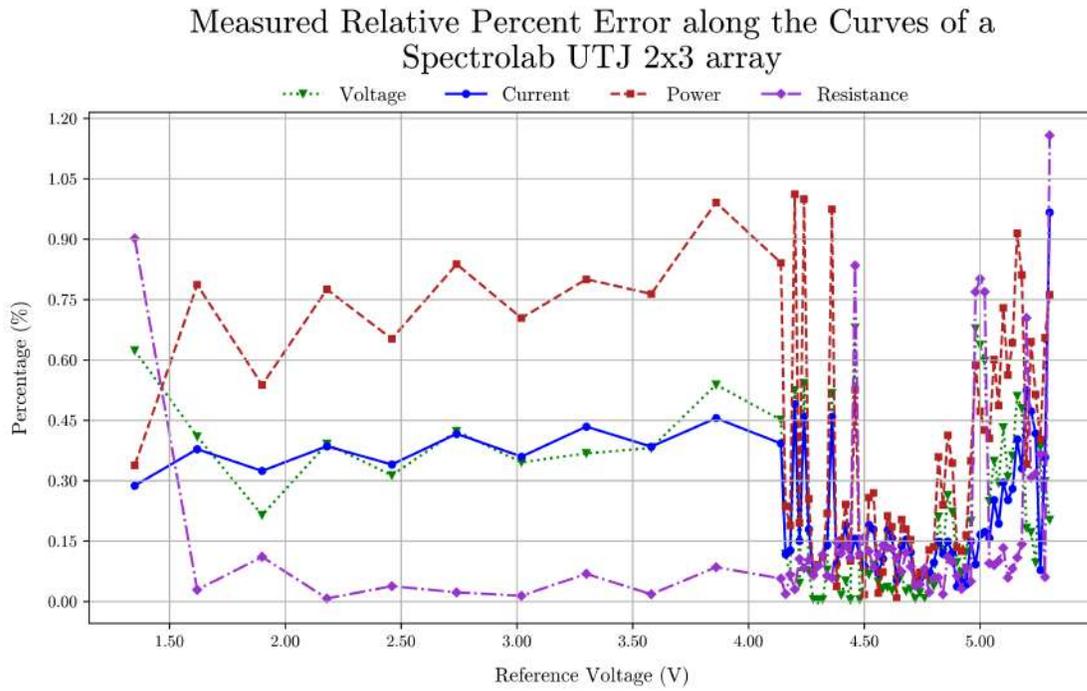


Figura F.40: Porcentajes de error en curvas caract. de un arreglo 2x3 (Spectrolab UTJ).

F.1.6. Arreglo 6x1

Finalmente, se procede con las curvas obtenidas para los arreglos 6x1 de cada panel. Cabe recordar que este es el arreglo más favorable en todos los paneles, puesto que los controladores digitales de la fuente de poder están mejor optimizado para regular tensiones altas cuando la corriente es baja. En el caso del AZUR 3G30A, Figura F.41, su rendimiento es muy bueno, con excepción de los últimos puntos donde se ve afectado por ARPEs elevados conforme la corriente se acerca a 0. En el AZUR 4G32C, Figura F.43, se tiene un comportamiento similar, presentando un rendimiento excelente antes de verse afectado por el ARPE aún más elevado que en el caso anterior en el último punto. El mismo comportamiento es consistente para el CESI CTJ30, Figura F.41, y el Spectrolab UTJ, Figura F.47, aunque en estos casos los ARPEs al final no se disparan tanto.

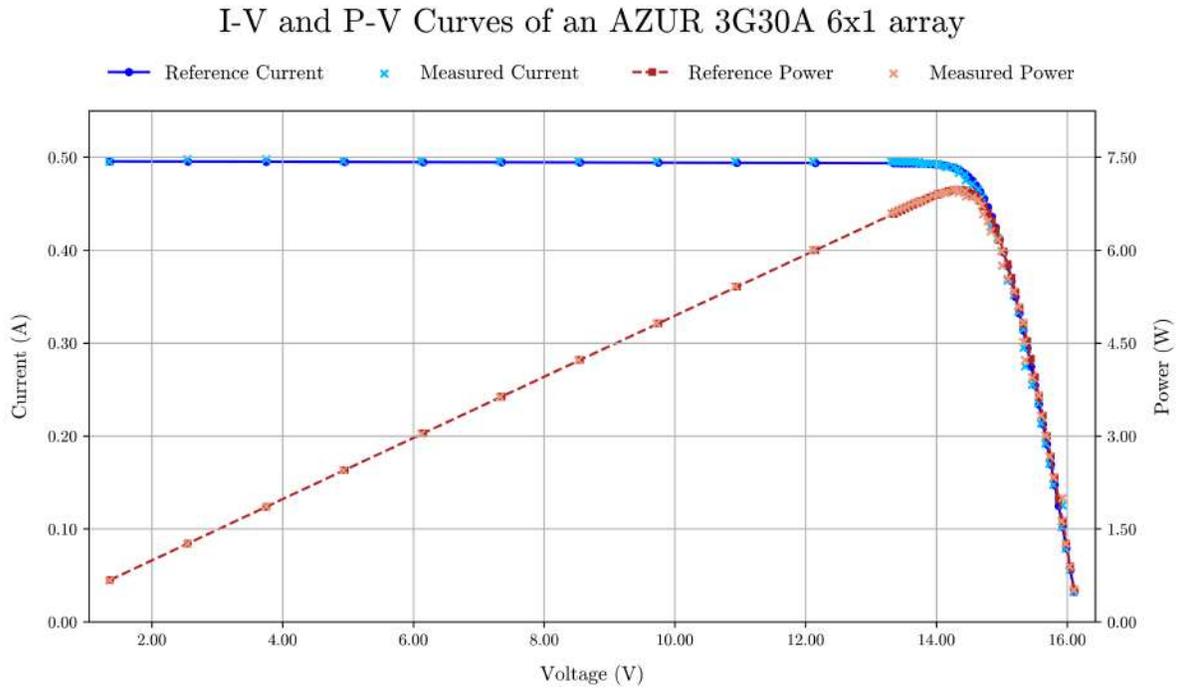


Figura F.41: Curvas I-V y P-V de un arreglo 6x1 (AZUR 3G30A).

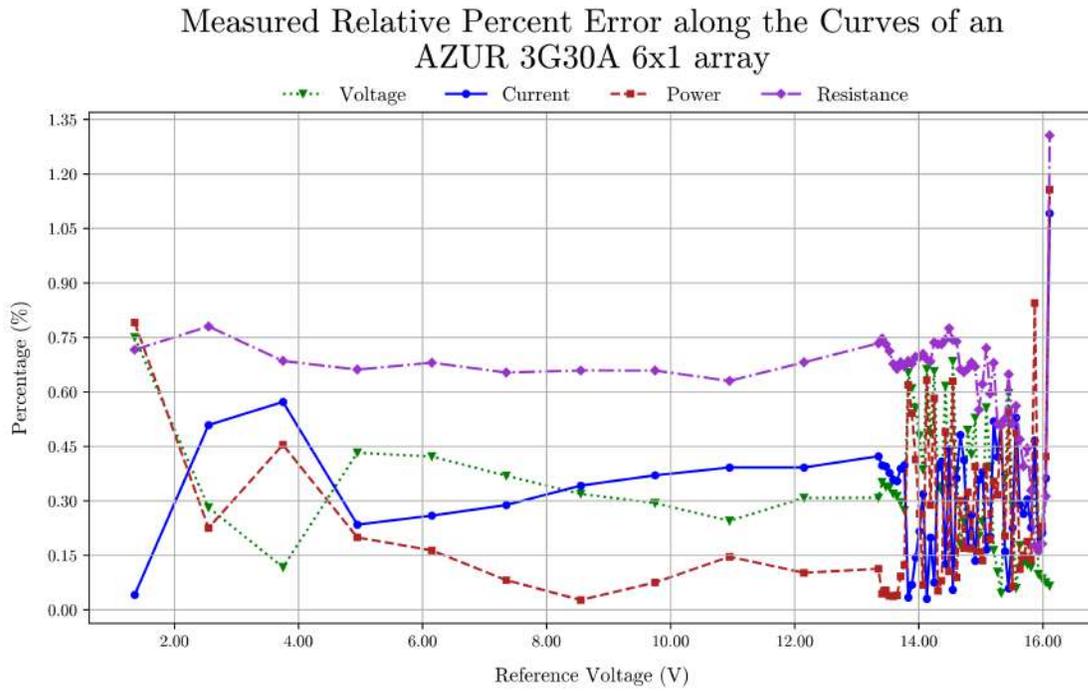


Figura F.42: Porcentajes de error en curvas características de un arreglo 6x1 (AZUR 3G30A).

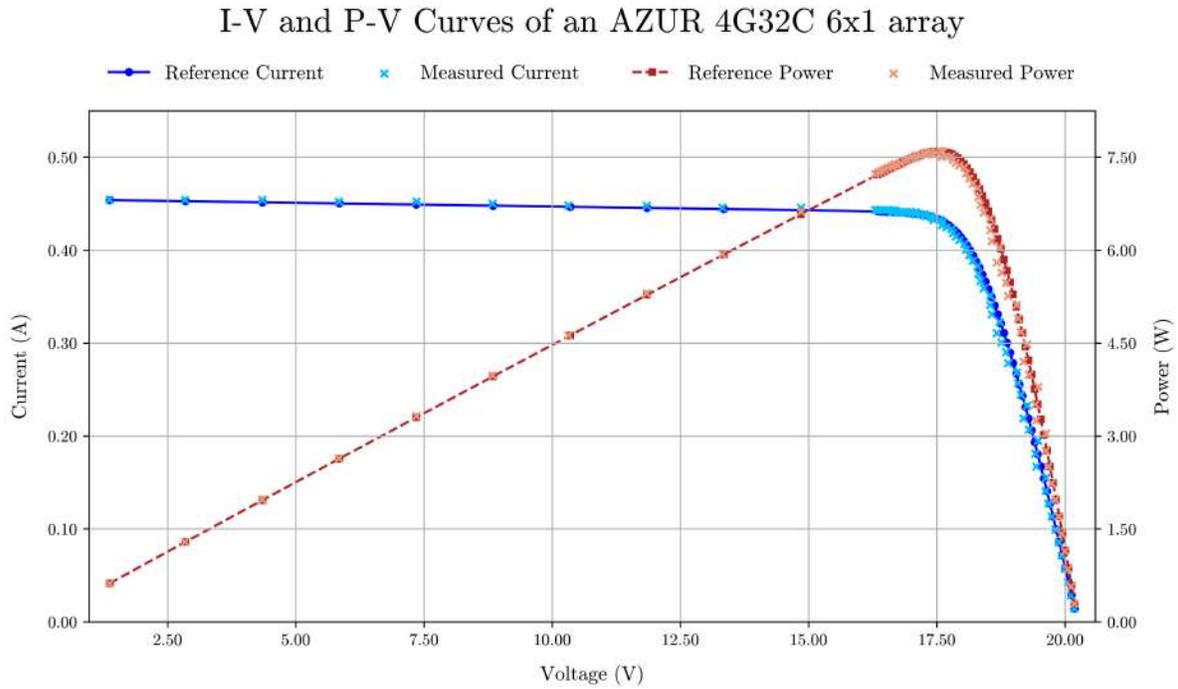


Figura F.43: Curvas I-V y P-V de un arreglo 6x1 (AZUR 4G32C).

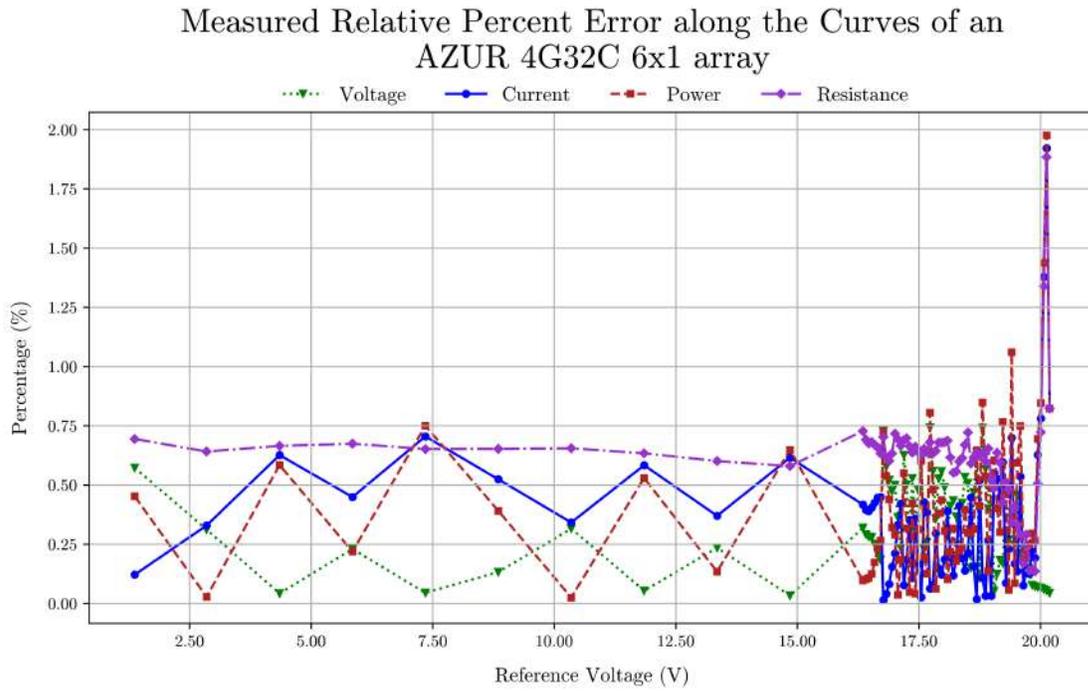


Figura F.44: Porcentajes de error en curvas caract. de un arreglo 6x1 (AZUR 4G32C).

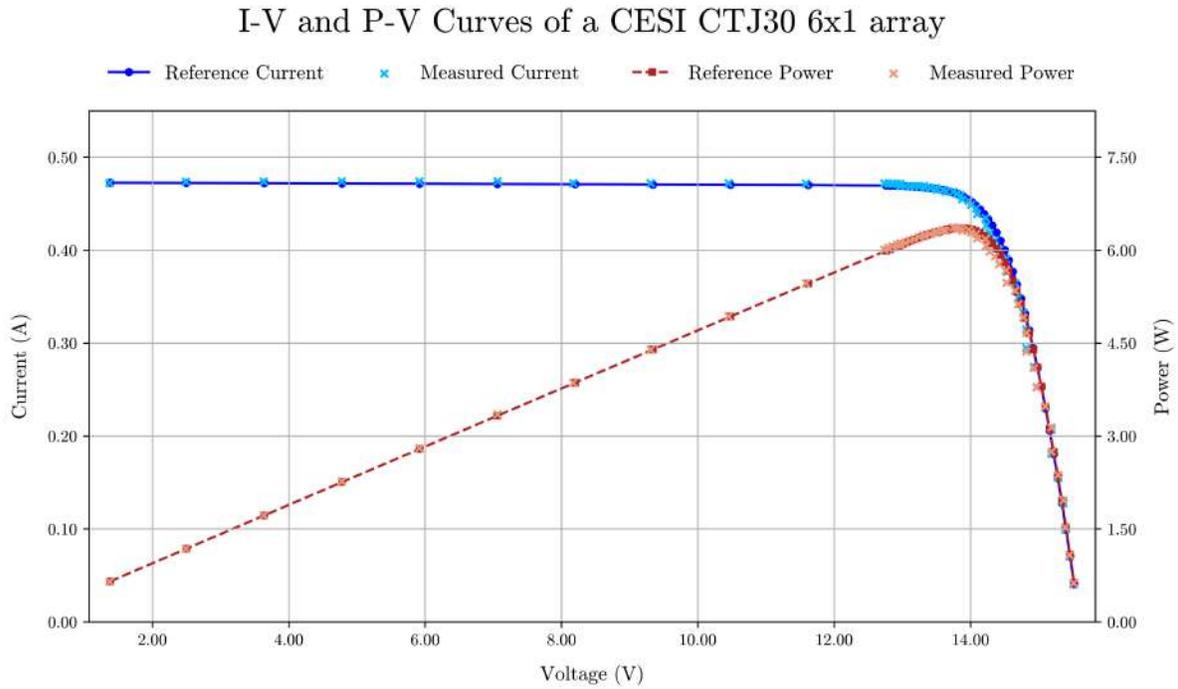


Figura F.45: Curvas I-V y P-V de un arreglo 6x1 (CESI CTJ30).

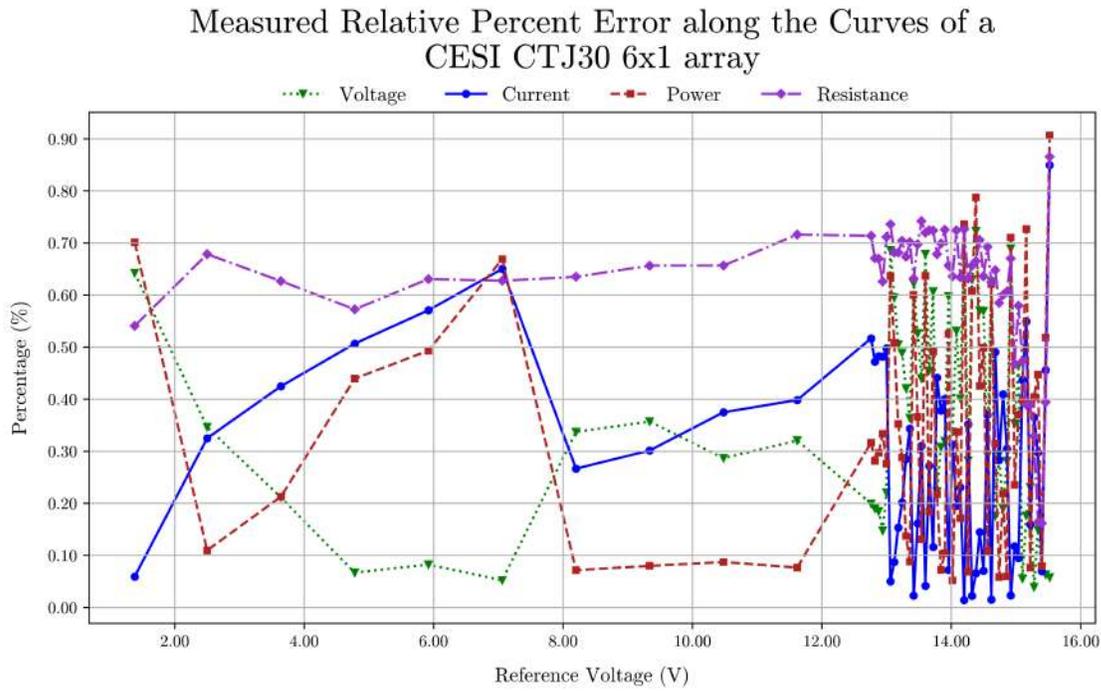


Figura F.46: Porcentajes de error en curvas características de un arreglo 6x1 (CESI CTJ30).

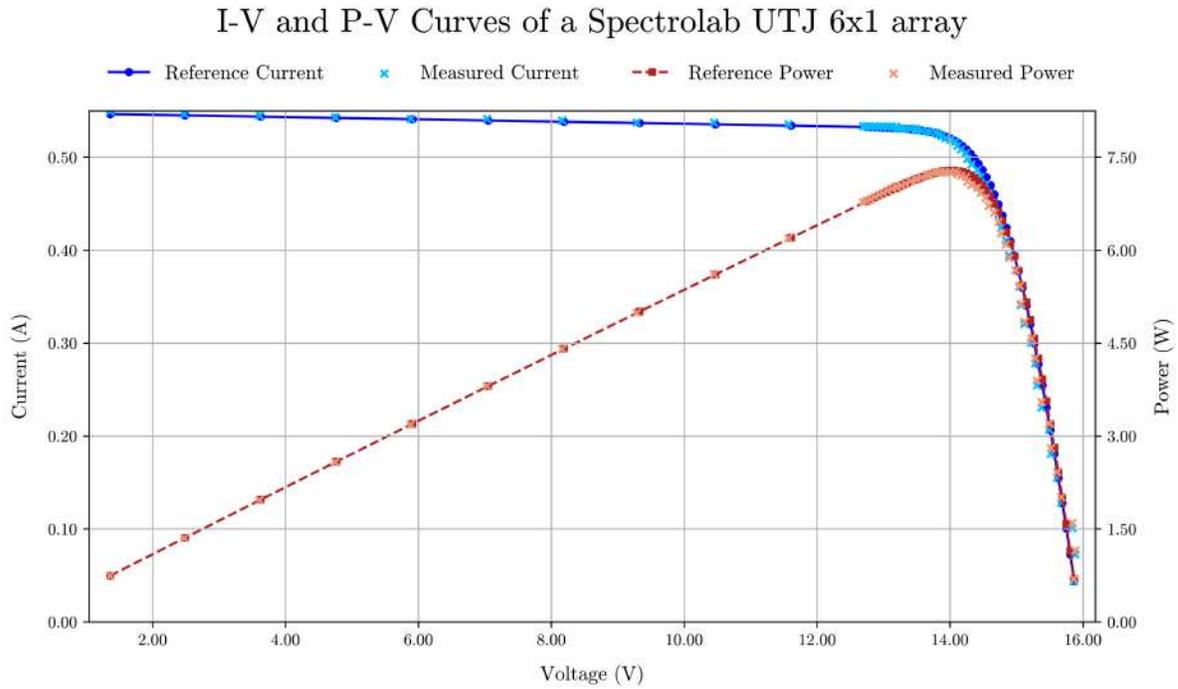


Figura F.47: Curvas I-V y P-V de un arreglo 6x1 (Spectrolab UTJ).

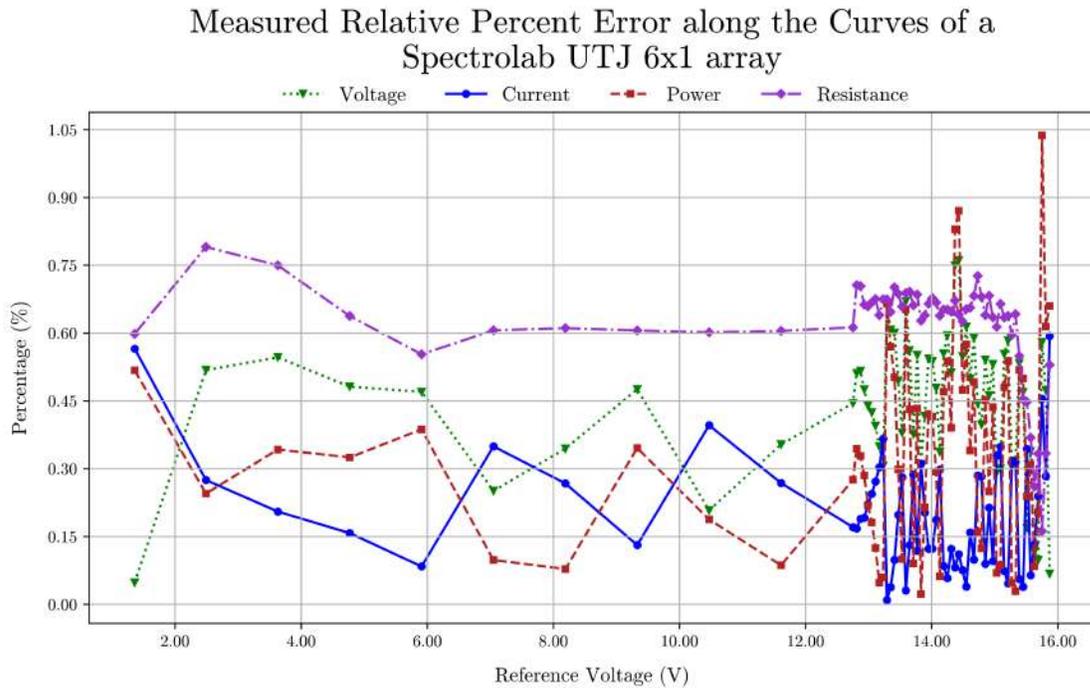


Figura F.48: Porcentajes de error en curvas caract. de un arreglo 6x1 (Spectrolab UTJ).

F.2. Pruebas de condiciones externas variables

F.2.1. Irradiancia

Ahora se procede a analizar las curvas obtenidas cuando se realiza un MPPT en todo el rango de las condiciones externas, iniciando con la irradiancia. En todos los casos se observa que, conforme aumenta la irradiancia, los ARPEs también aumentan. En el caso del AZUR 3G30A, Figura F.49, y el AZUR 4G32C, Figura F.51, el comportamiento es prácticamente lineal, mientras que en el caso del CESI CTJ30, Figura F.53, y el Spectrolab UTJ, Figura F.55, se presentan mayores oscilaciones a partir de la mitad del rango. Esto podría deberse a que, dado que tanto la tensión como la corriente aumentan continuamente, a los controladores de la fuente de poder se le dificulta más mantener ambos parámetros estables.

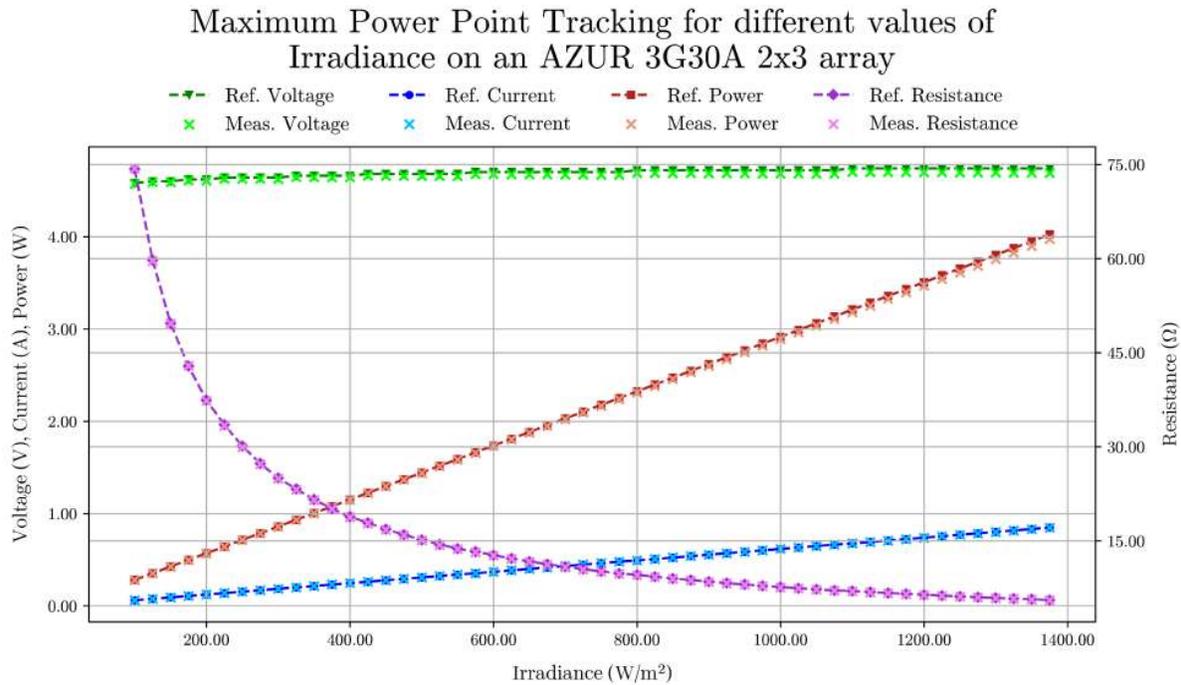


Figura F.49: MPPT a diferentes niveles de irradiancia (AZUR 3G30A).

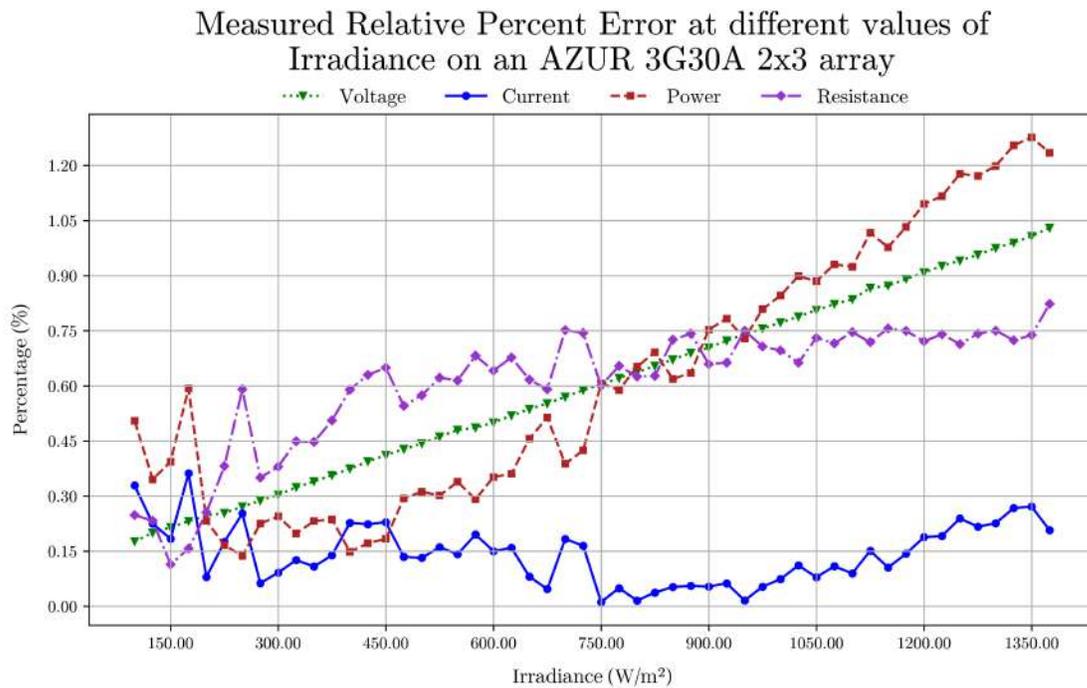


Figura F.50: Porcentajes de error a diferentes niveles de irradiancia (AZUR 3G30A).

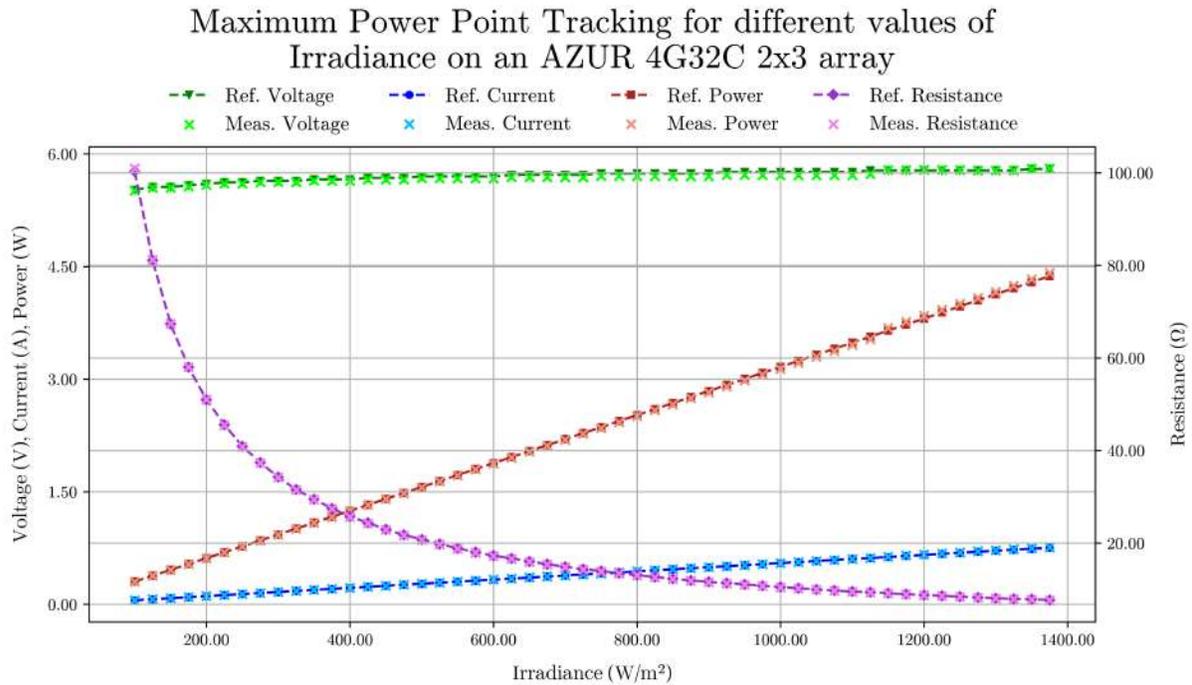


Figura F.51: MPPT a diferentes niveles de irradiancia (AZUR 4G32C).

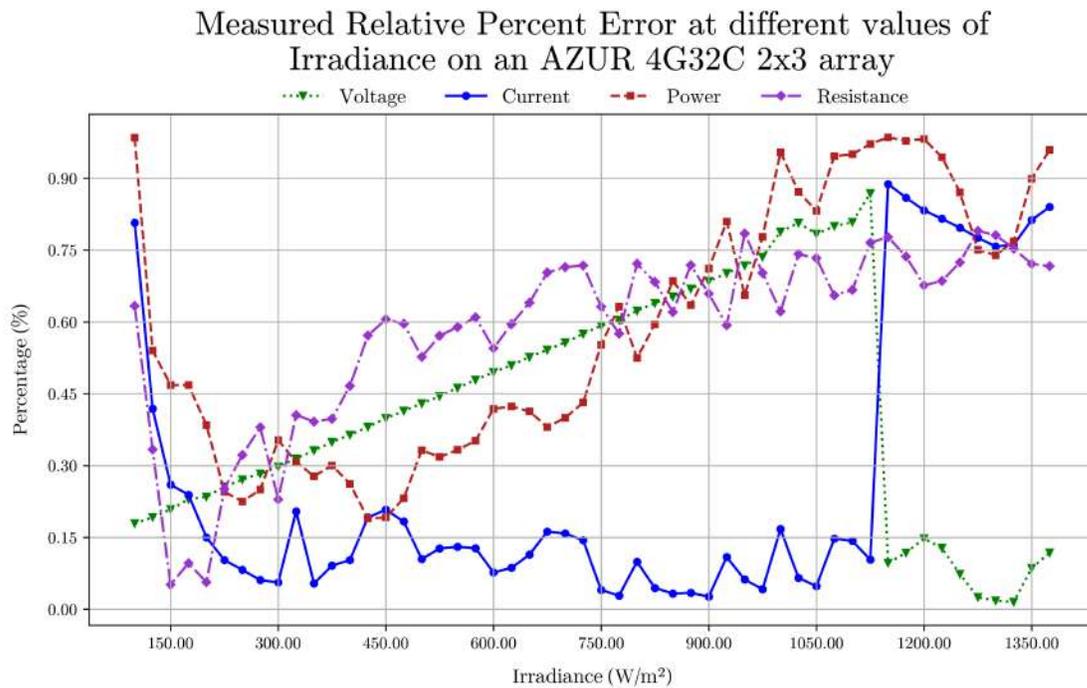


Figura F.52: Porcentajes de error a diferentes niveles de irradiancia (AZUR 4G32C).

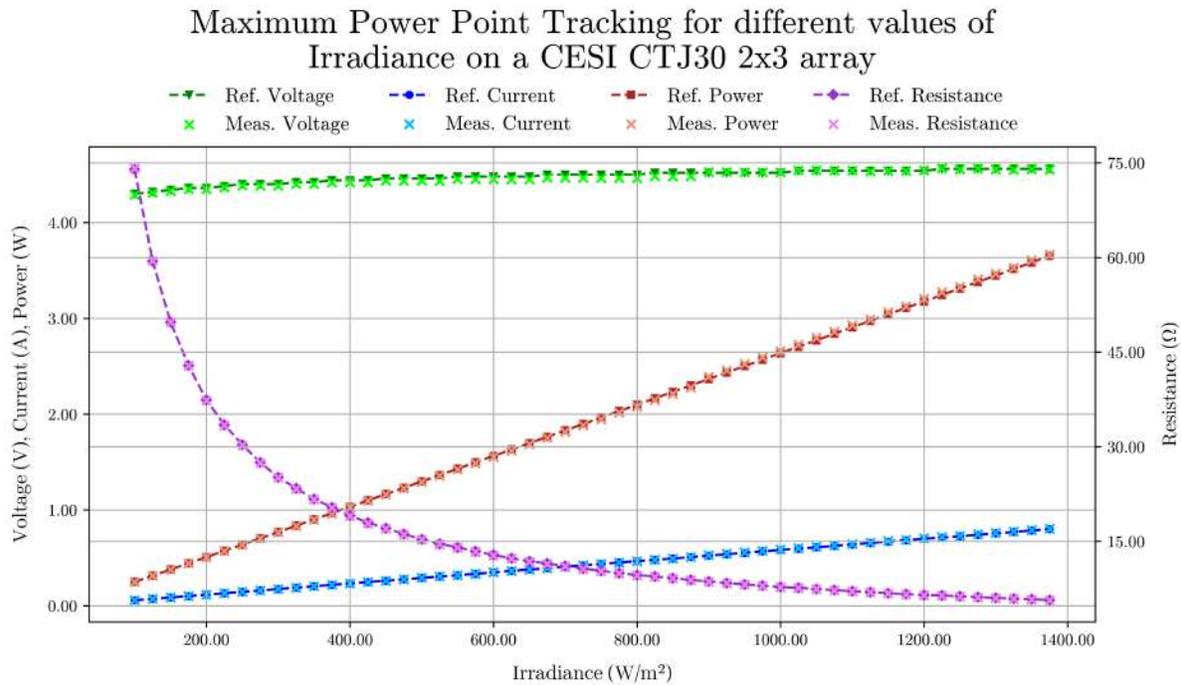


Figura F.53: MPPT a diferentes niveles de irradiancia (CESI CTJ30).

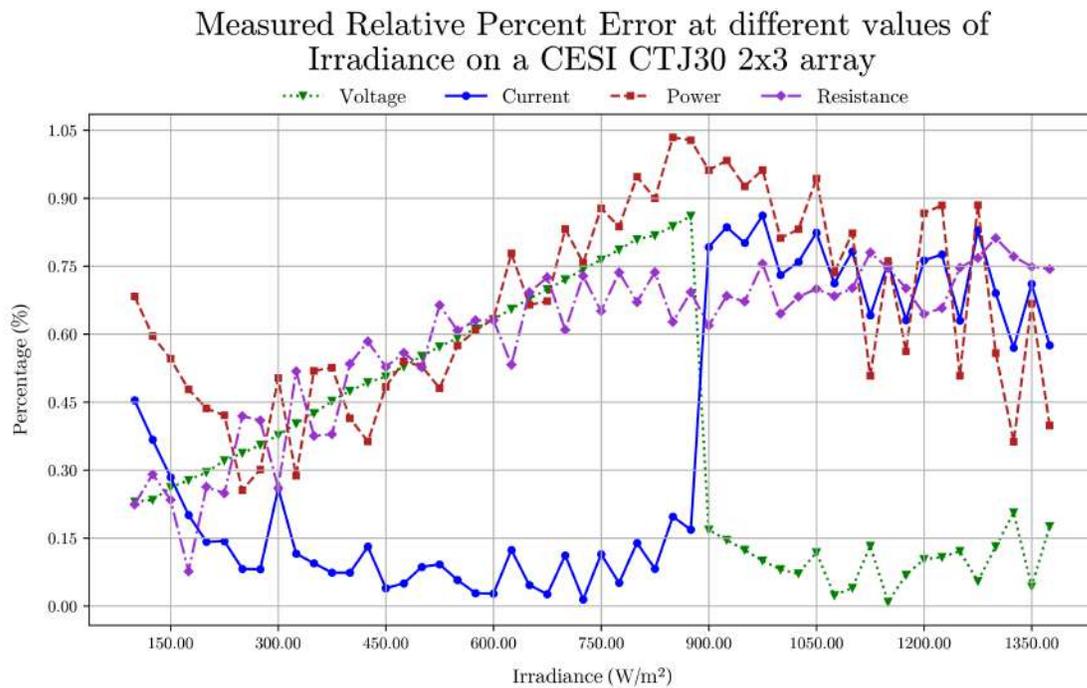


Figura F.54: Porcentajes de error a diferentes niveles de irradiancia (CESI CTJ30).

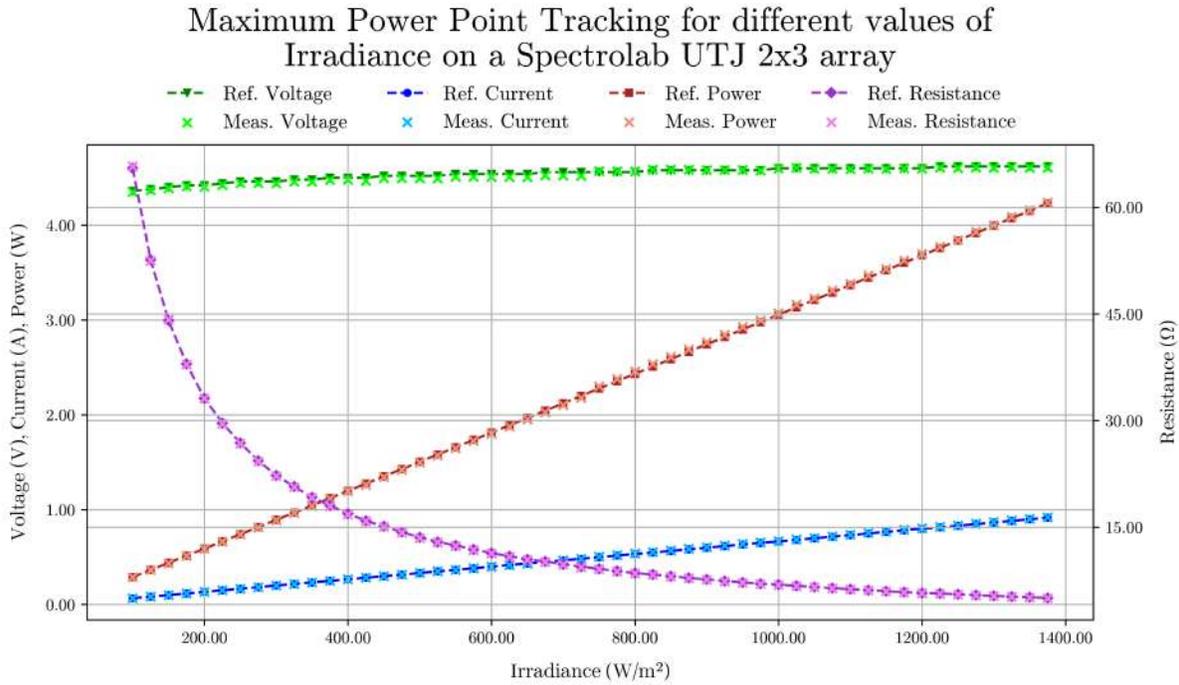


Figura F.55: MPPT a diferentes niveles de irradiancia (Spectrolab UTJ).

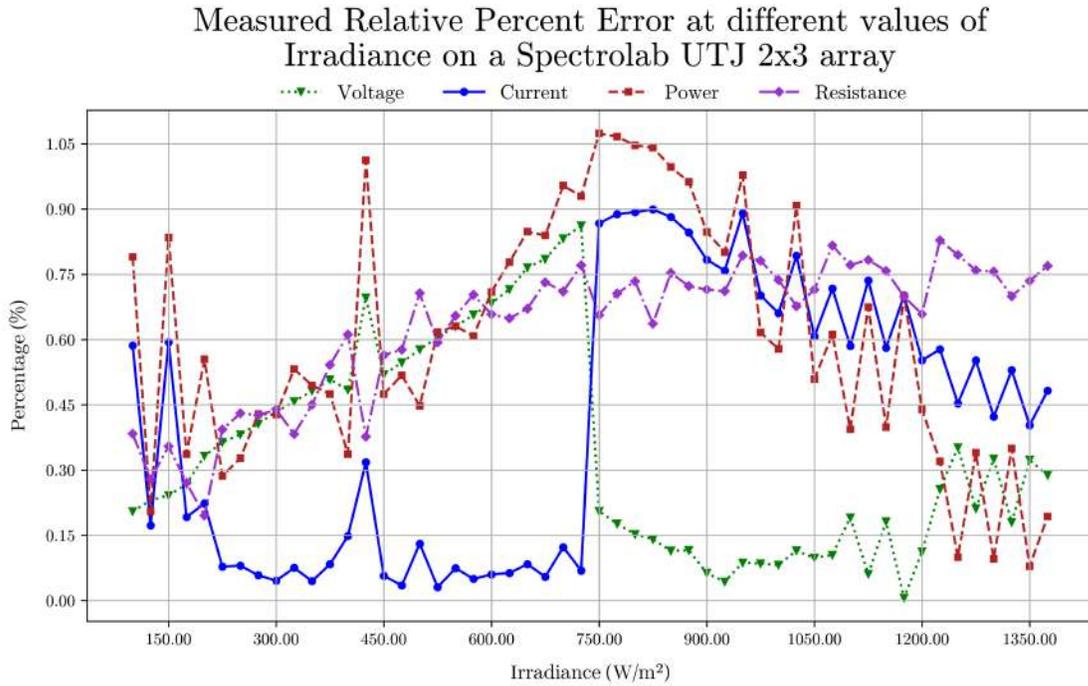


Figura F.56: Porcentajes de error a diferentes niveles de irradiancia (Spectrolab UTJ).

F.2.2. Temperatura

Ahora se procede con las curvas obtenidas al recorrer el rango de temperatura. La particularidad de este caso es que la tensión se hace cada vez más pequeña, mientras que la corriente aumenta, lo cual se ha visto que resulta problemático para los controladores de la fuente. Por lo tanto, a diferencia del caso de la irradiancia, donde se tenían ARPES muy bajos al inicio y luego aumentaban, en este caso se tienen ARPES más cercanos al 1% en todo el rango, lo cual es consistente en los cuatro paneles: AZUR 3G30A, Figura F.57, AZUR 4G32C, Figura F.59, CESI CTJ30, y Spectrolab UTJ, Figura F.63.

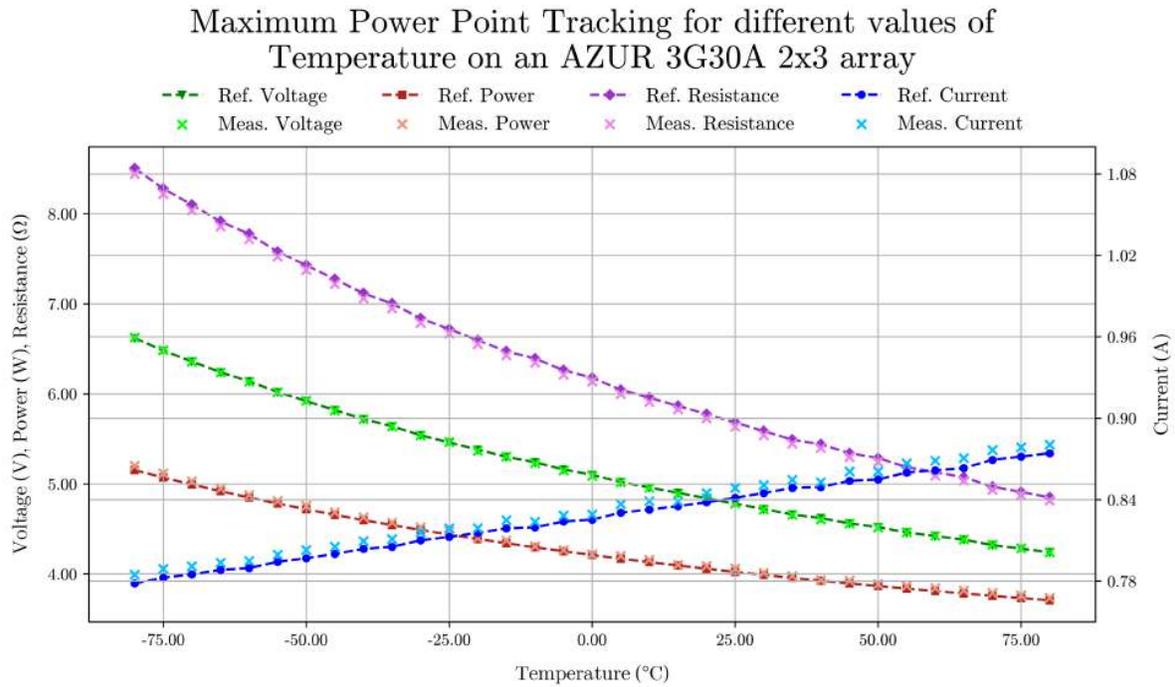


Figura F.57: MPPT a diferentes niveles de temperatura (AZUR 3G30A).

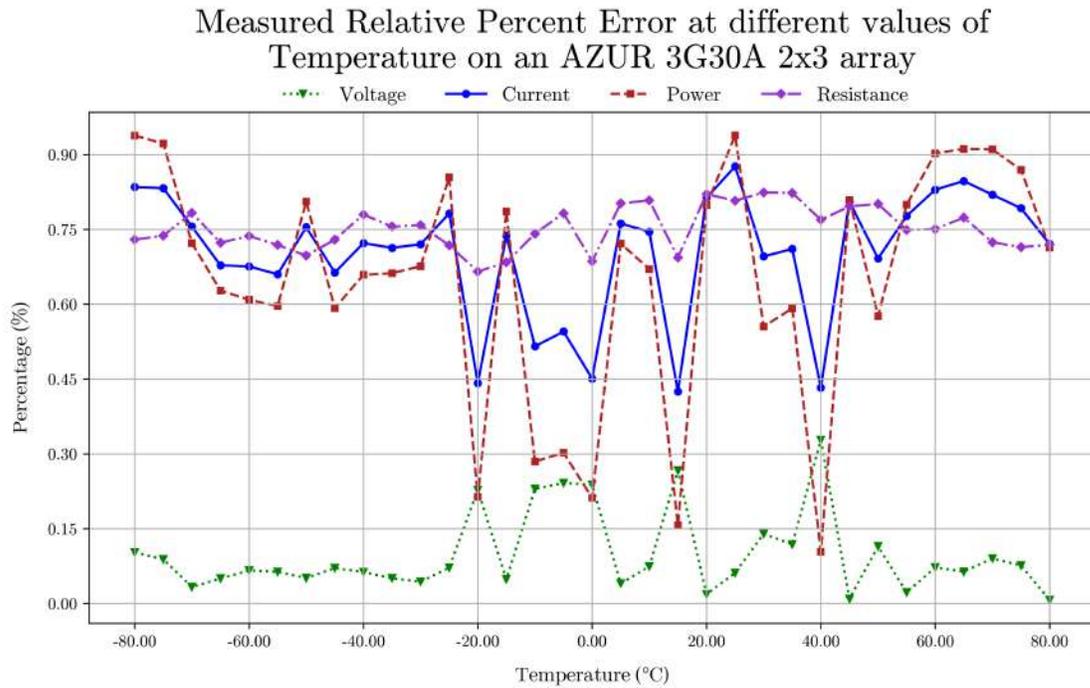


Figura F.58: Porcentajes de error a diferentes niveles de temperatura (AZUR 3G30A).

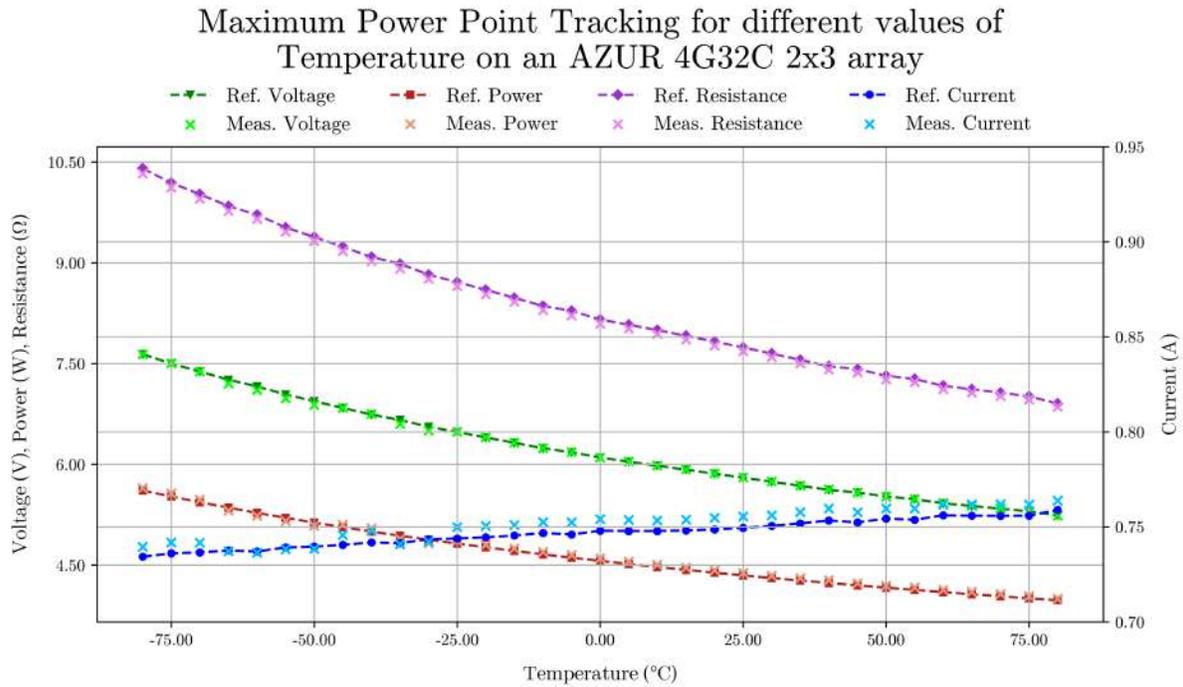


Figura F.59: MPPT a diferentes niveles de temperatura (AZUR 4G32C).

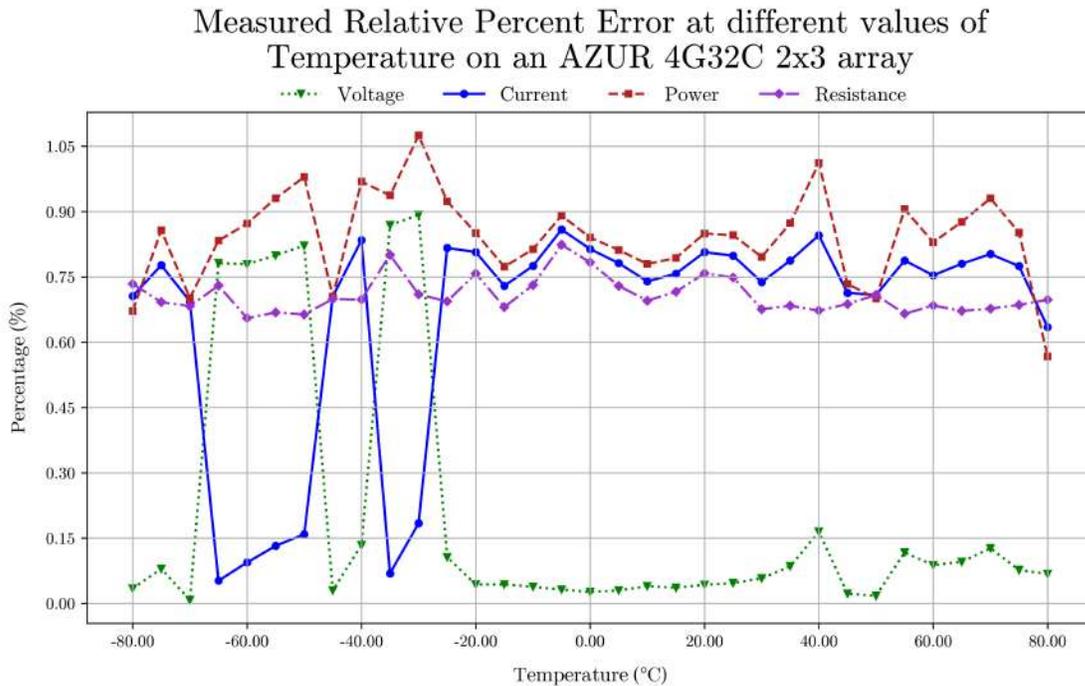


Figura F.60: Porcentajes de error a diferentes niveles de temperatura (AZUR 4G32C).

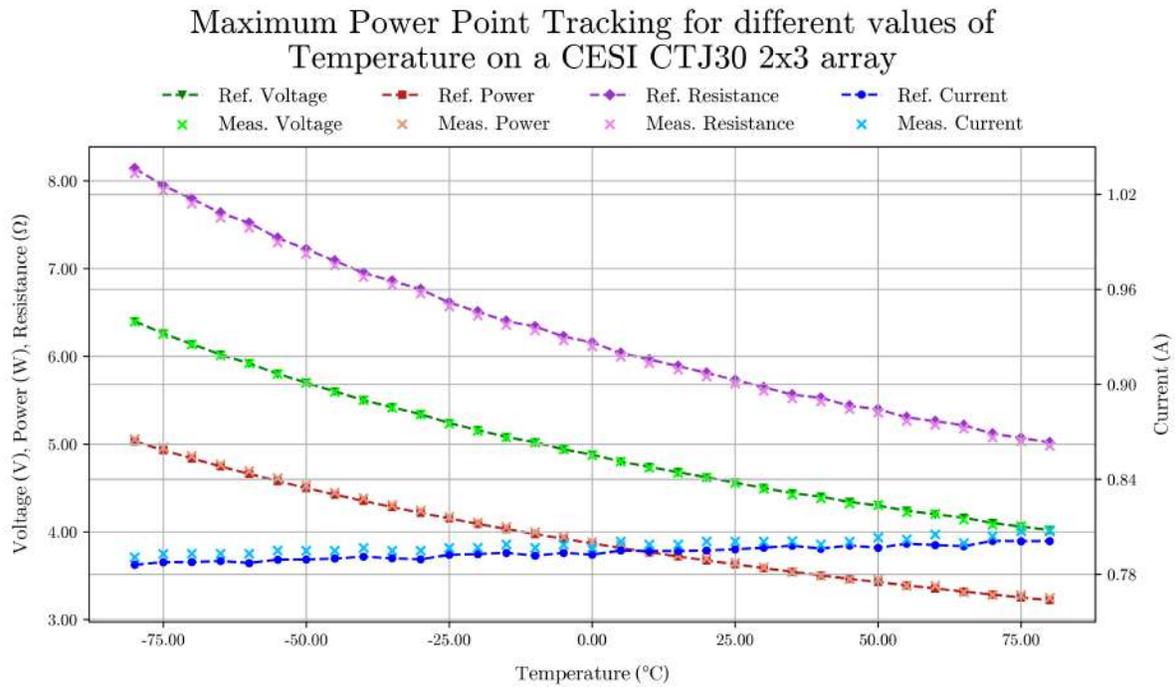


Figura F.61: MPPT a diferentes niveles de temperatura (CESI CTJ30).

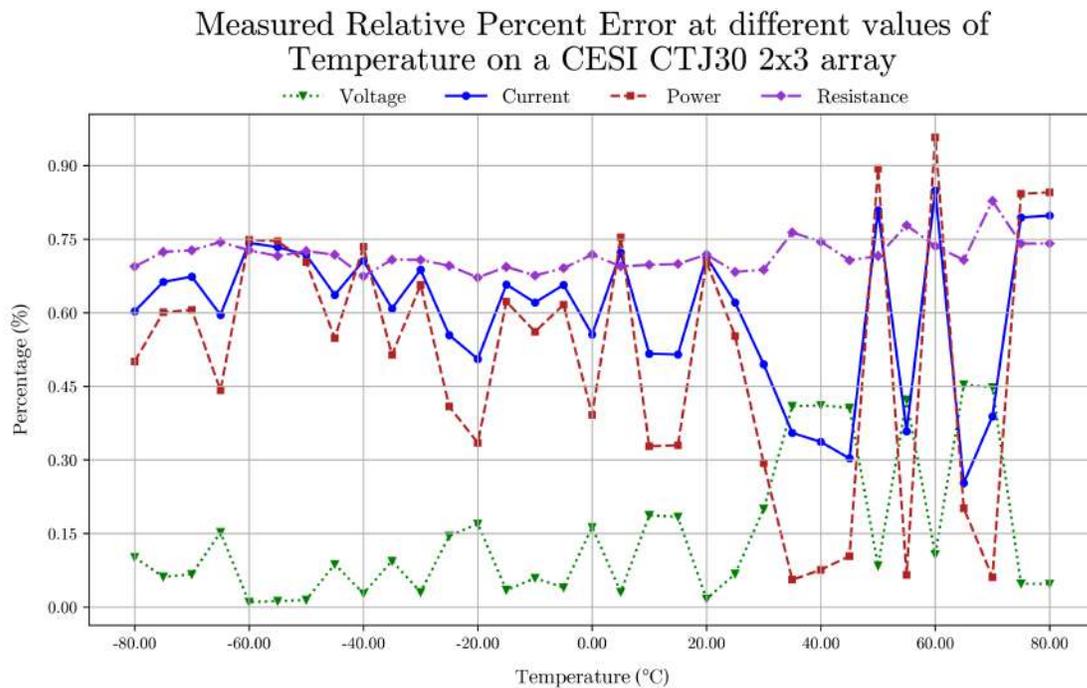


Figura F.62: Porcentajes de error a diferentes niveles de temperatura (CESI CTJ30).

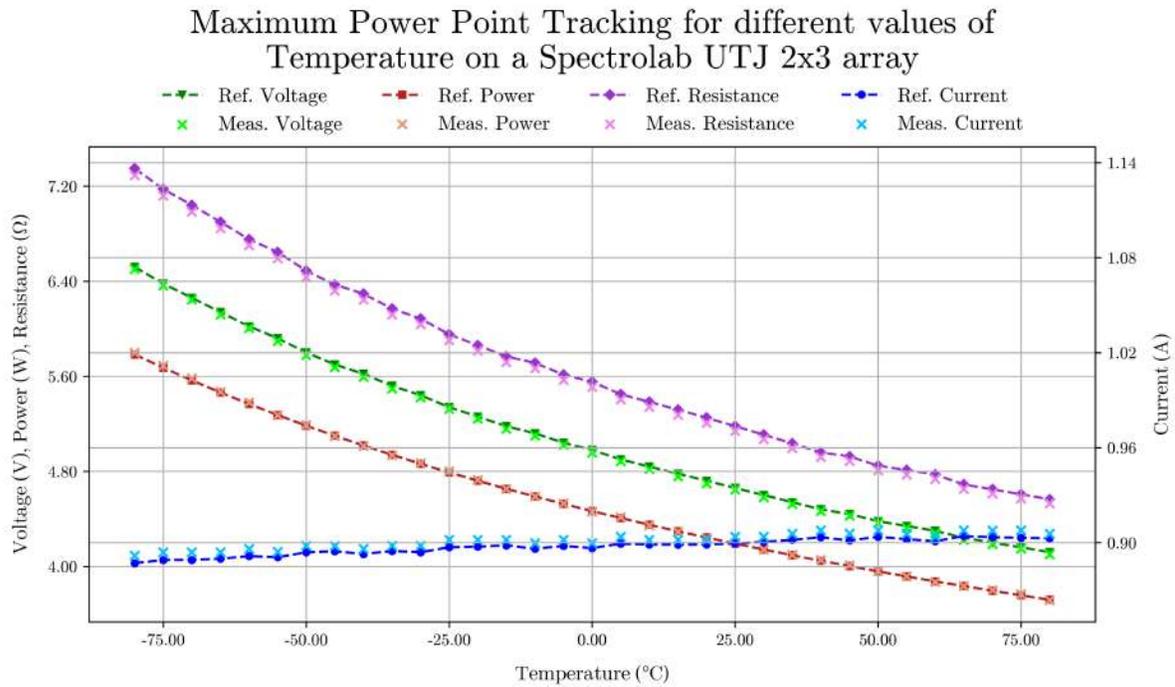


Figura F.63: MPPT a diferentes niveles de temperatura (Spectrolab UTJ).

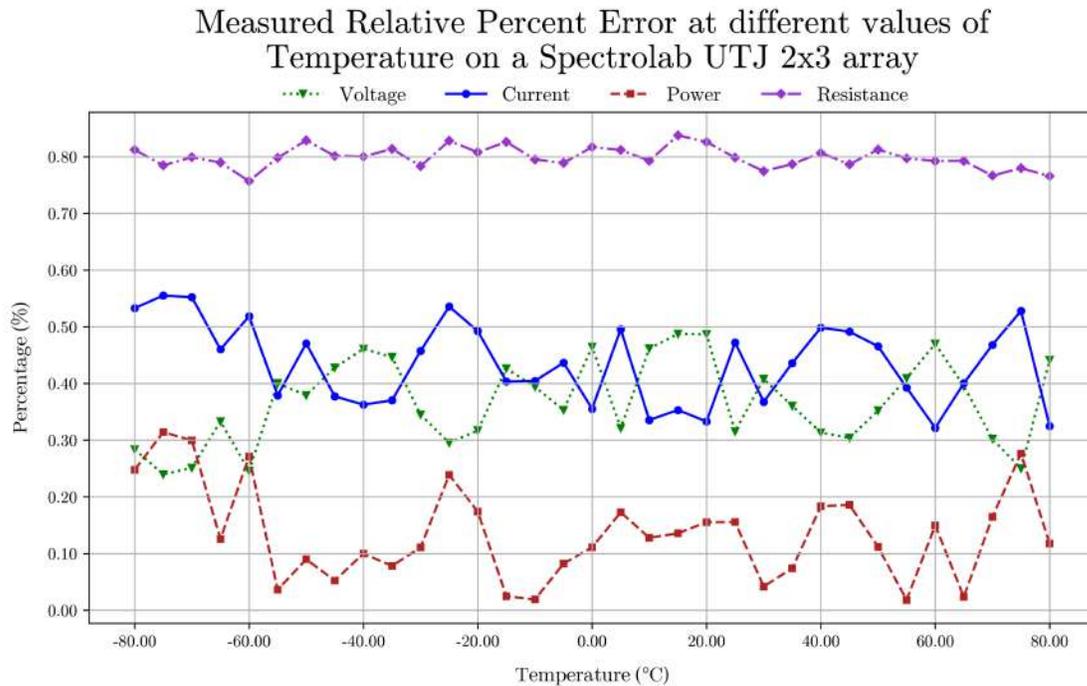


Figura F.64: Porcentajes de error a diferentes niveles de temperatura (Spectrolab UTJ).

F.2.3. Ángulo de incidencia

Finalmente, se procede con el rango de ángulos de incidencia. Este caso es bastante similar al de la irradiancia, solo que esta vez la corriente empieza siendo alta y disminuye en los últimos puntos, presentado esta vez un comportamiento menos lineal debido a la función coseno, según se explicó en el Capítulo 5. Los ARPES aumentan rápidamente en los últimos puntos, lo que se puede atribuir a la forma exponencial que aumenta la resistencia de la carga, causando inestabilidad en los controladores de la fuente. Esto es consistente en los cuatro paneles: AZUR 3G30A, Figura F.65, AZUR 4G32C, Figura F.67, CESI CTJ30, y Spectrolab UTJ, Figura F.71.

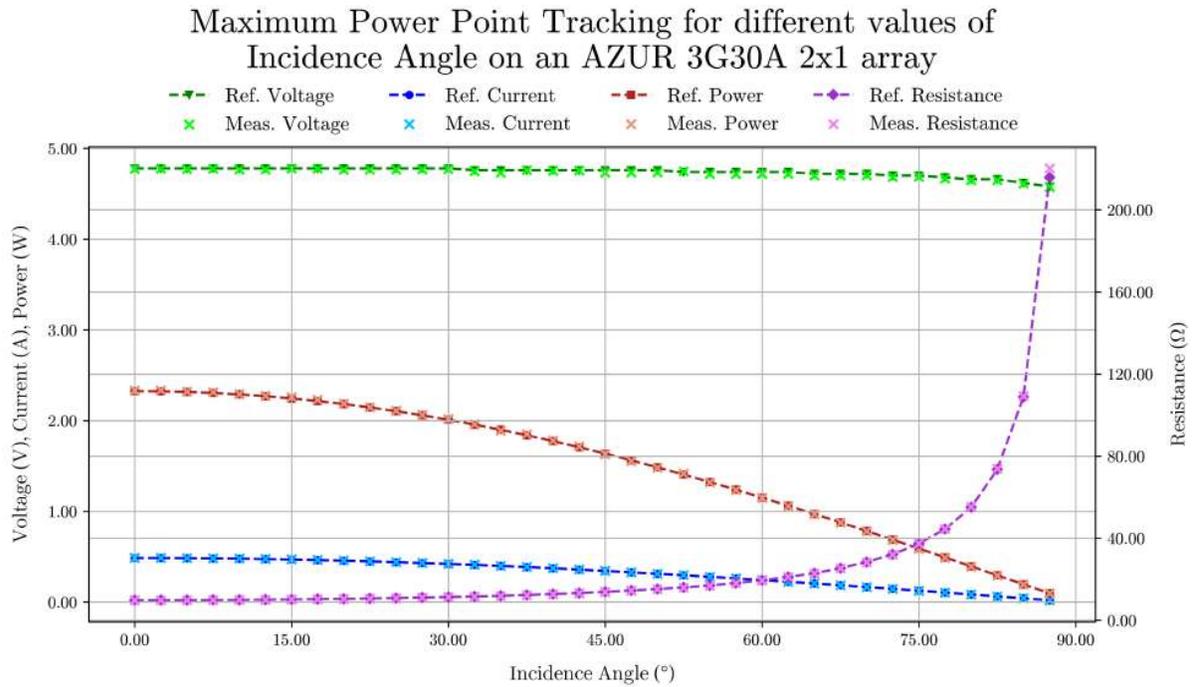


Figura F.65: MPPT a diferentes ángulos de incidencia. (AZUR 3G30A).

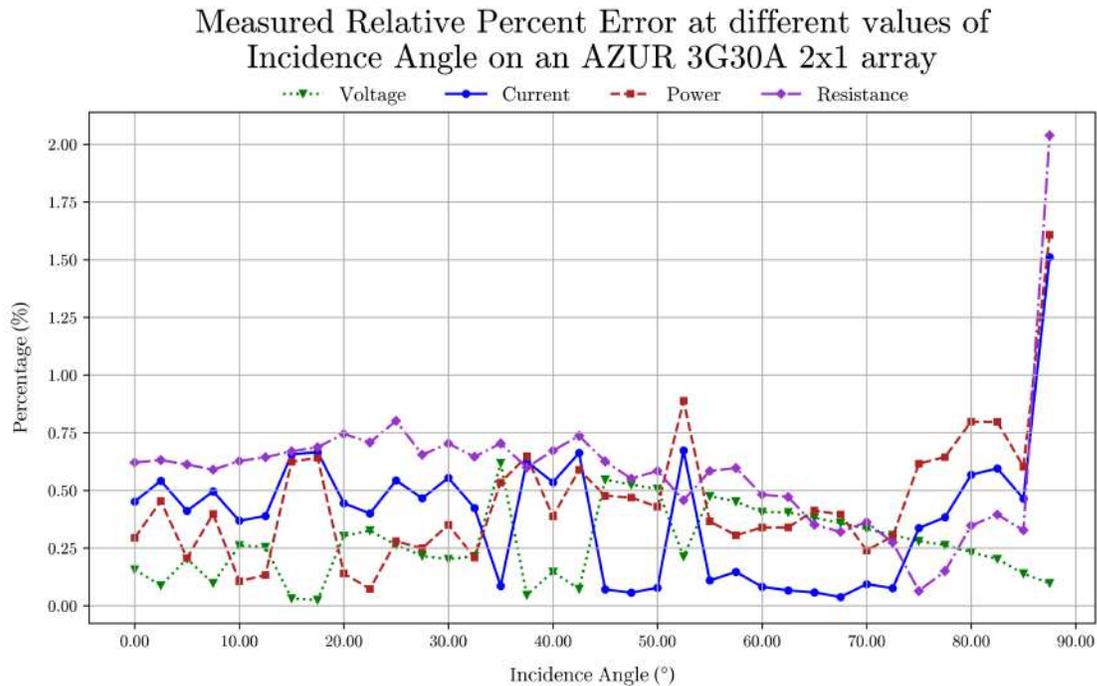


Figura F.66: Porcentajes de error a diferentes ángulos de incidencia (AZUR 3G30A).

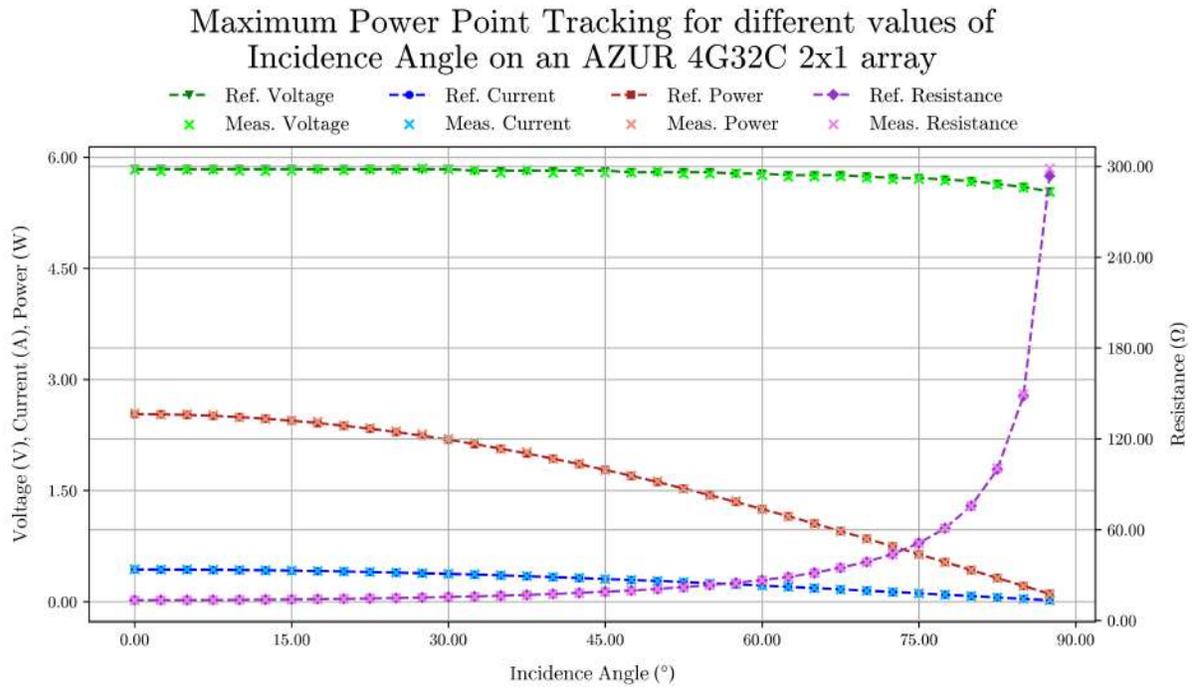


Figura F.67: MPPT a diferentes ángulos de incidencia. (AZUR 4G32C).

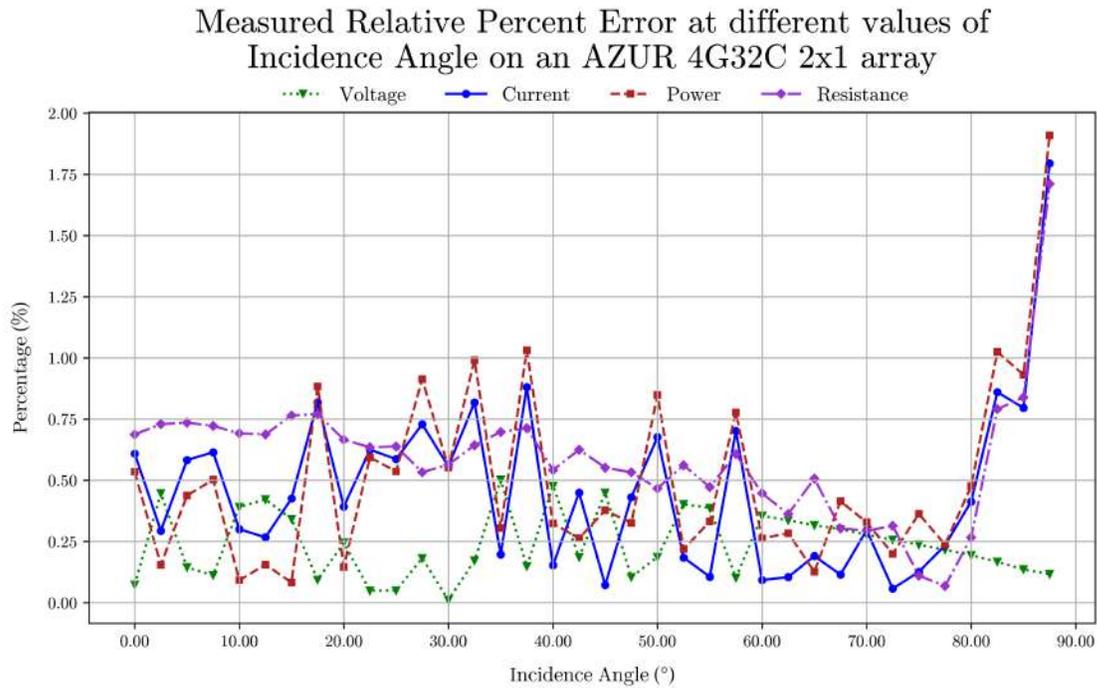


Figura F.68: Porcentajes de error a diferentes ángulos de incidencia (AZUR 4G32C).

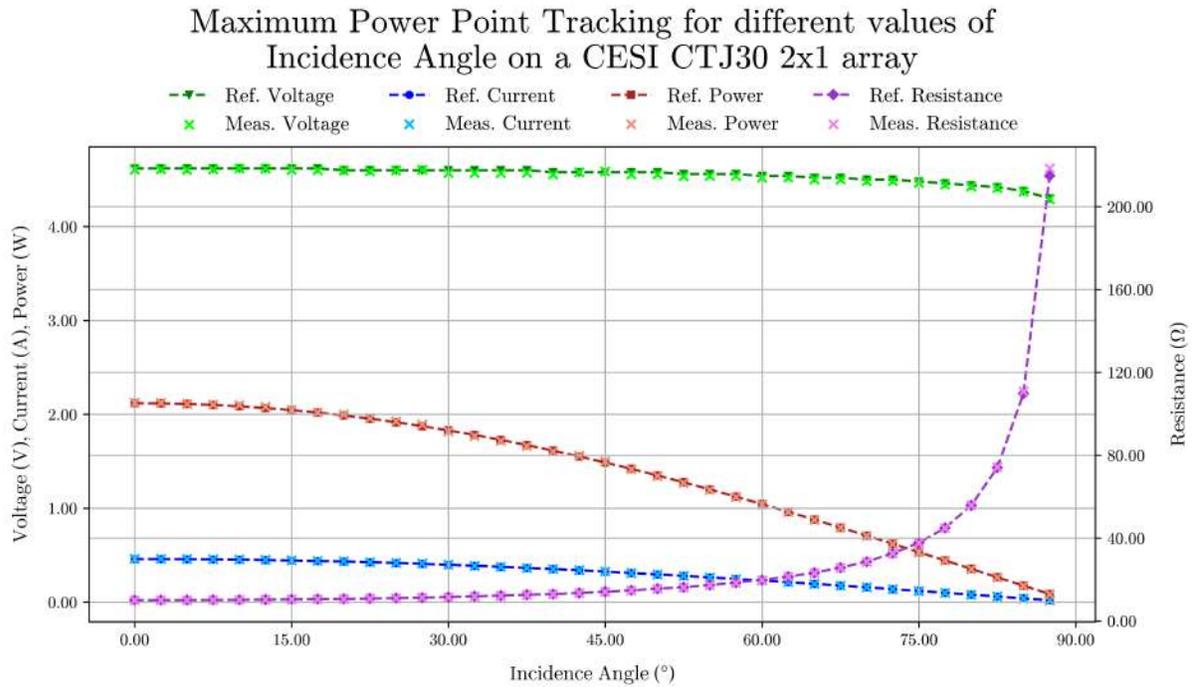


Figura F.69: MPPT a diferentes ángulos de incidencia. (CESI CTJ30).

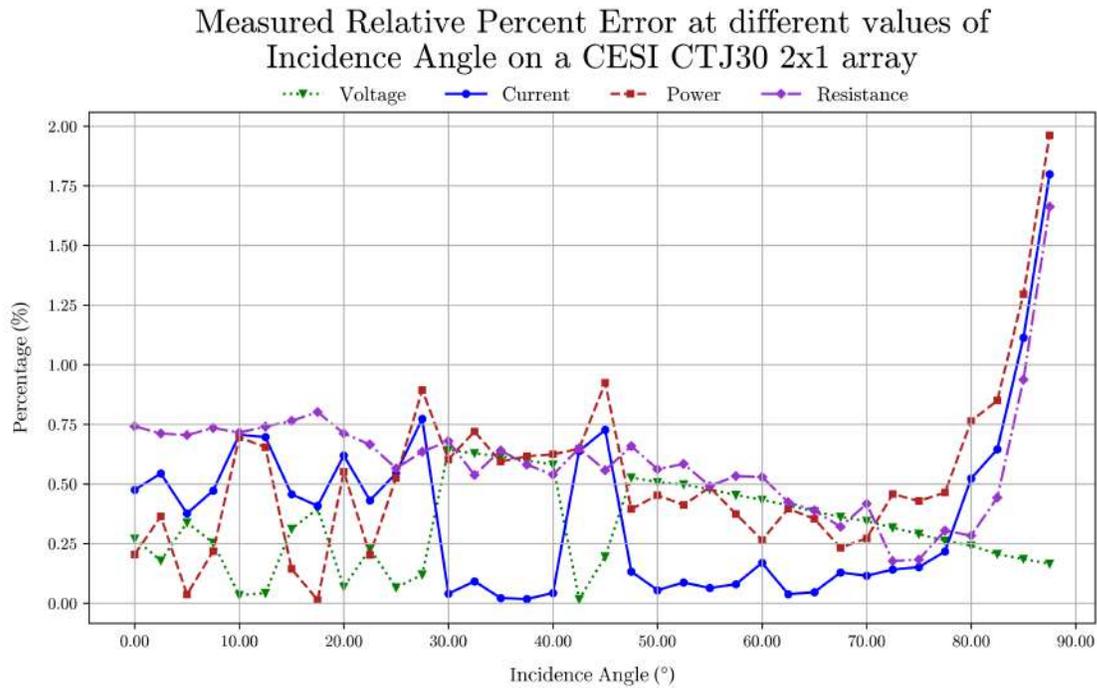


Figura F.70: Porcentajes de error a diferentes ángulos de incidencia (CESI CTJ30).

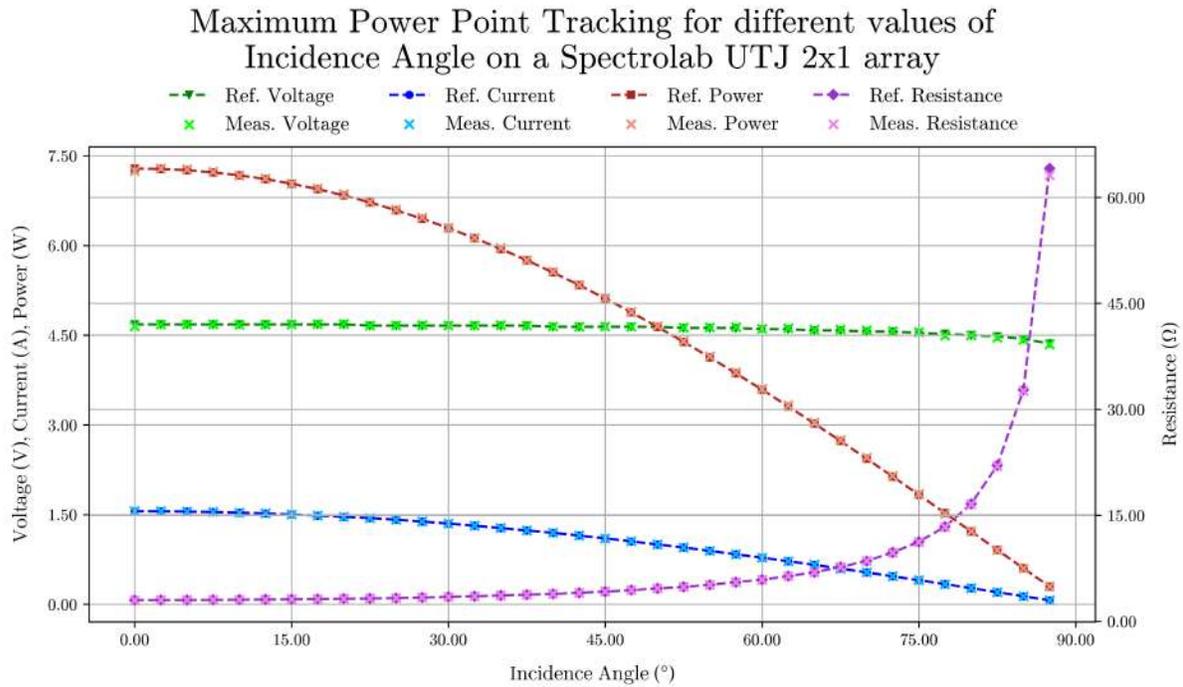


Figura F.71: MPPT a diferentes ángulos de incidencia. (Spectrolab UTJ).

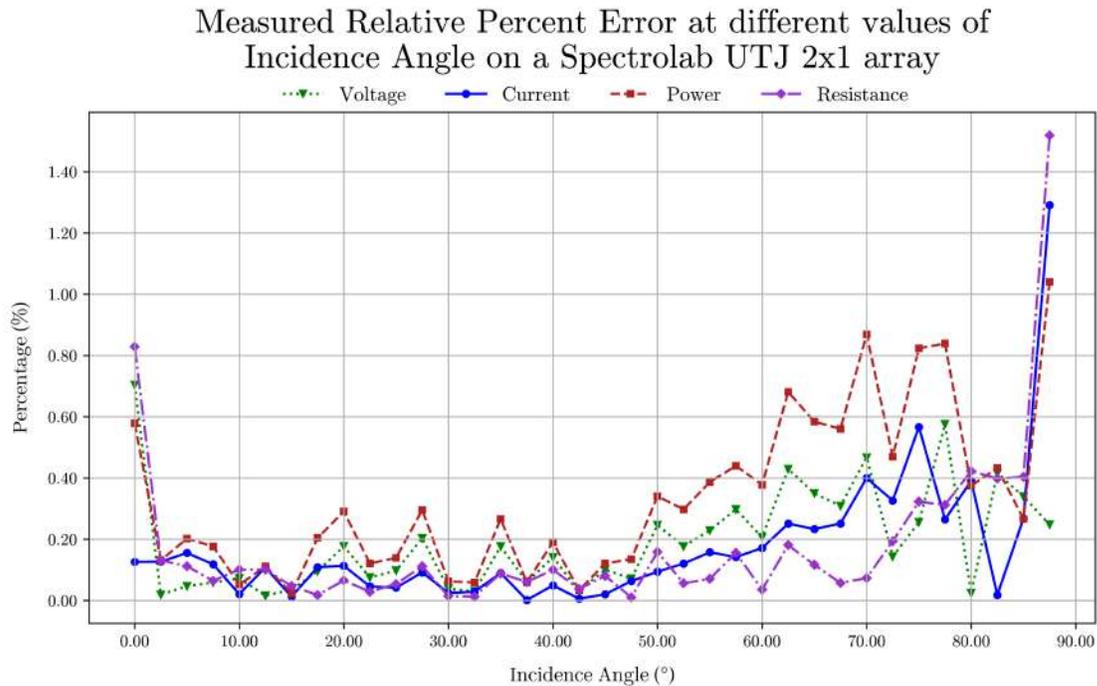


Figura F.72: Porcentajes de error a diferentes ángulos de incidencia (Spectrolab UTJ).

Anexo G

Hojas de datos de paneles solares

Nota: En este [enlace](#) se puede visualizar y descargar el documento en formato .PDF, en el cual los hipervínculos para la descarga de los sets de datos sí funcionan correctamente.

AZUR SPACE 3G30A

30% Efficiency Triple Junction Solar Cell

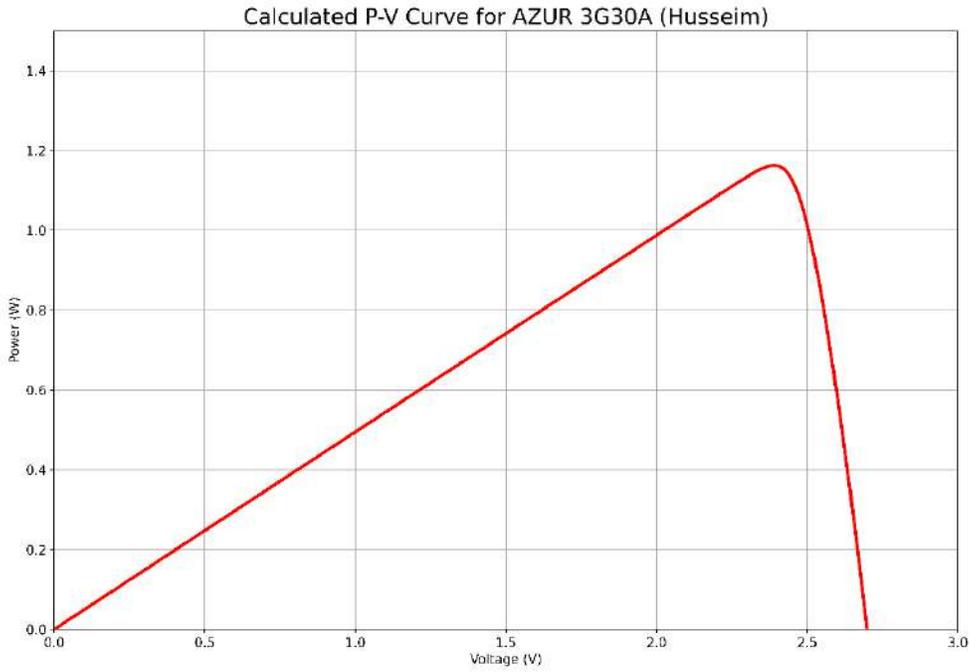
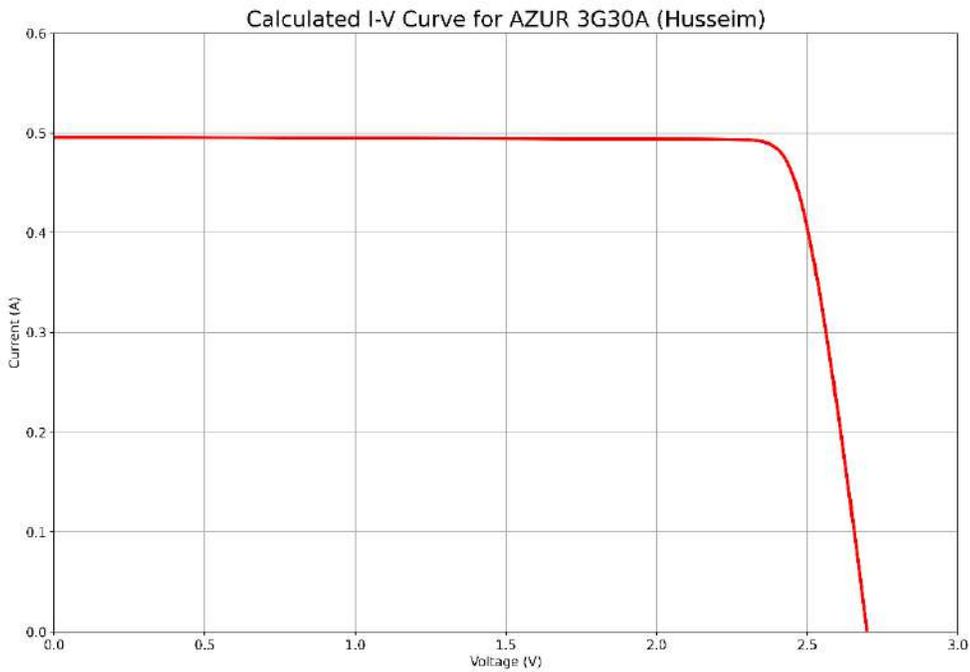
Electrical Data

Open Circuit Voltage	V_{oc}	2.699 V
Short Circuit Current	I_{sc}	0.496 A
MPP Voltage	V_{mp}	2.387 V
MPP Current	I_{mp}	0.487 A
MPP Power	P_{mp}	1.162 W
Number of cells in series	N_s	1
V_{oc} temperature coefficient	K_v	-6.2E-03 V/°C
I_{sc} temperature coefficient	K_i	3.6E-04 A/°C
Band gap	E_g	1.6 eV
Diode ideality factor	n	1.23
Photoelectric current	I_{ph}	0.4960 A
Diode inverse saturation current	I_o	9.44E-38 A
Series resistance	R_s	0.3555 Ω
Shunt resistance	R_{sh}	996.98 Ω
Root-Mean-Squared-Error*	RMSE	0.00210
Normalized-Root-Mean-Squared-Error*	nRMSE	0.00423
Coefficient of Determination*	R^2	0.99931
Mean-Absolute-Percentage-Error*	MAPE	0.95947 %

Spectrum: 1367 W/m²; T = 28 °C

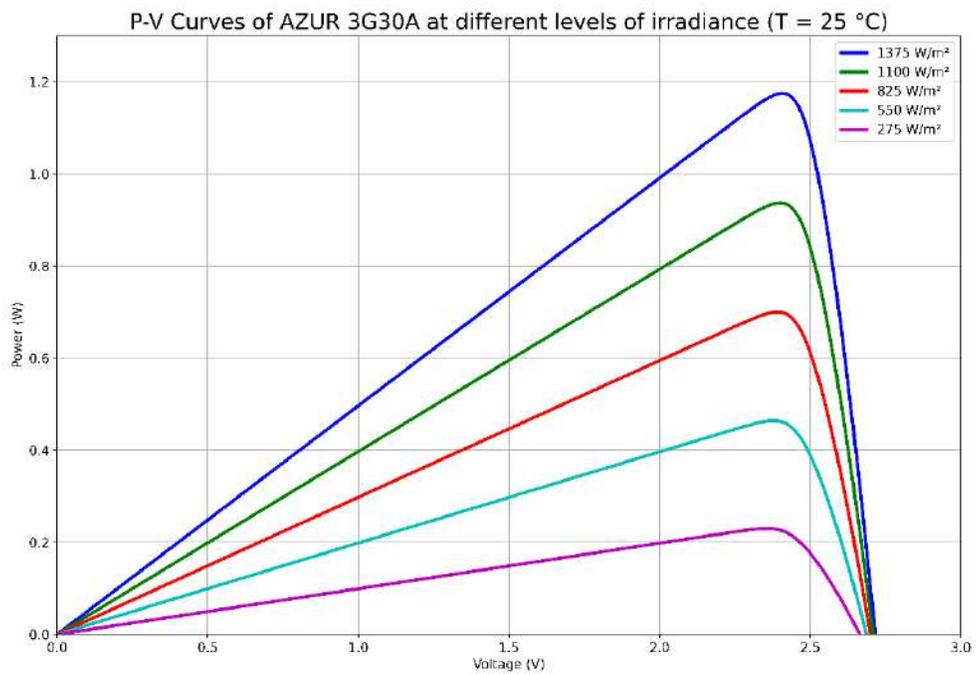
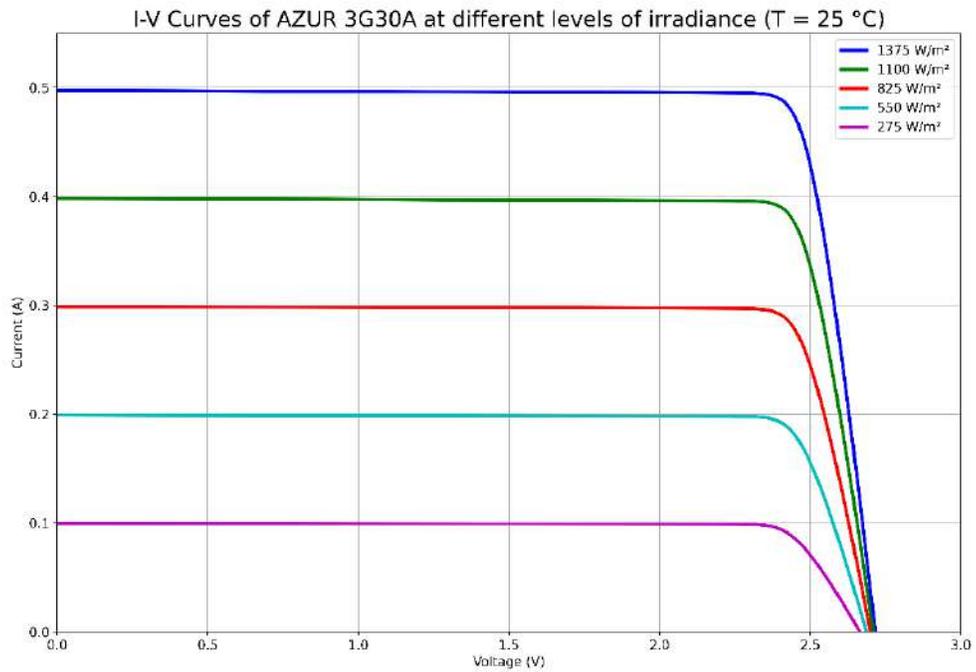
*Applies to the estimated values of n, I_{ph} , I_o , R_s and R_{sh}

Performance Data at Reference Conditions



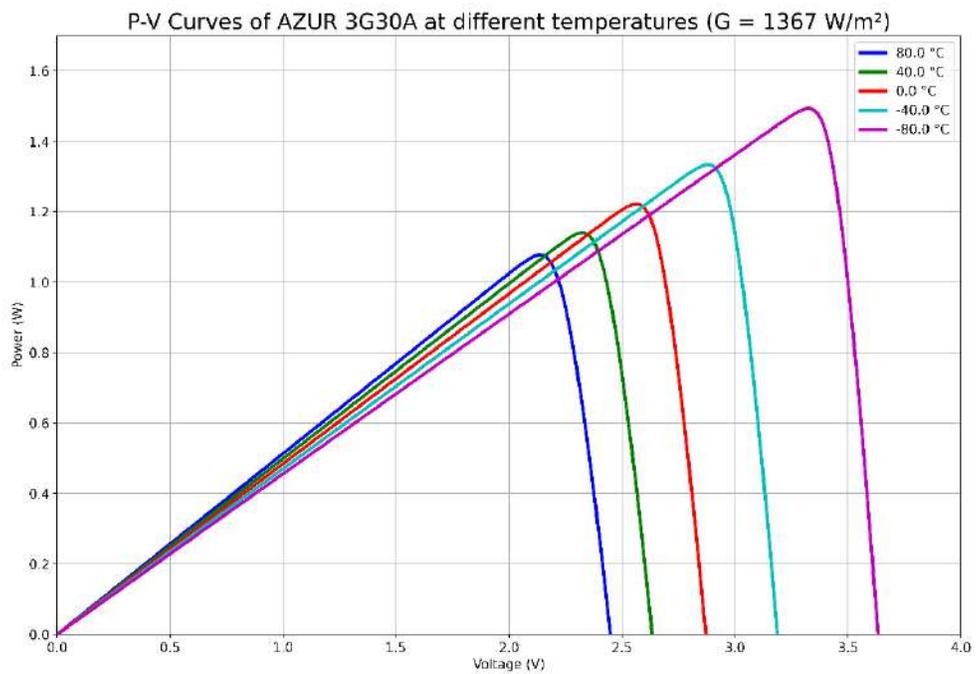
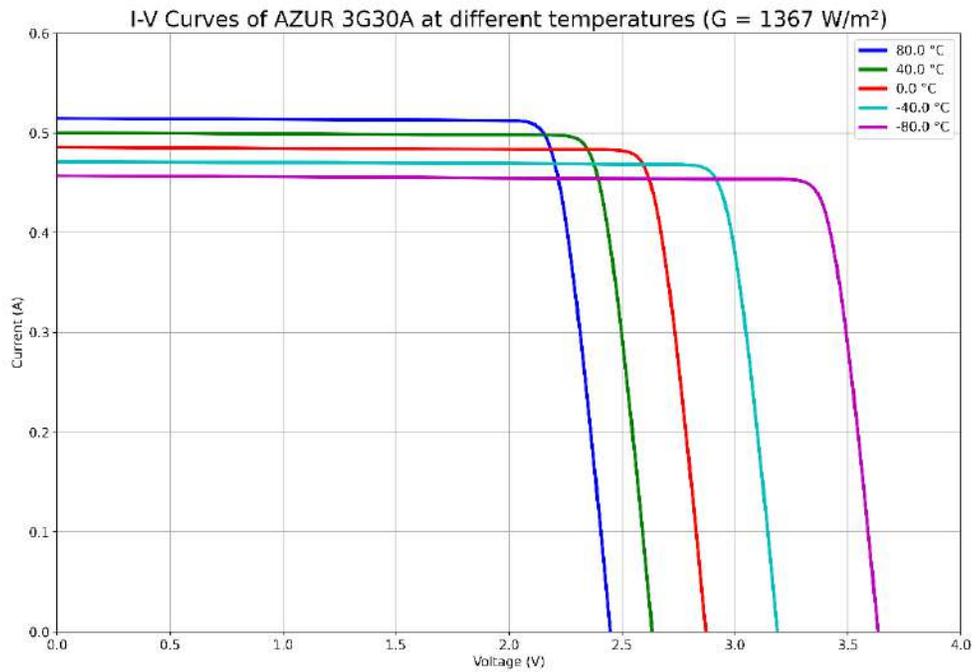
[Download datasets](#)

Performance Data at Different Conditions



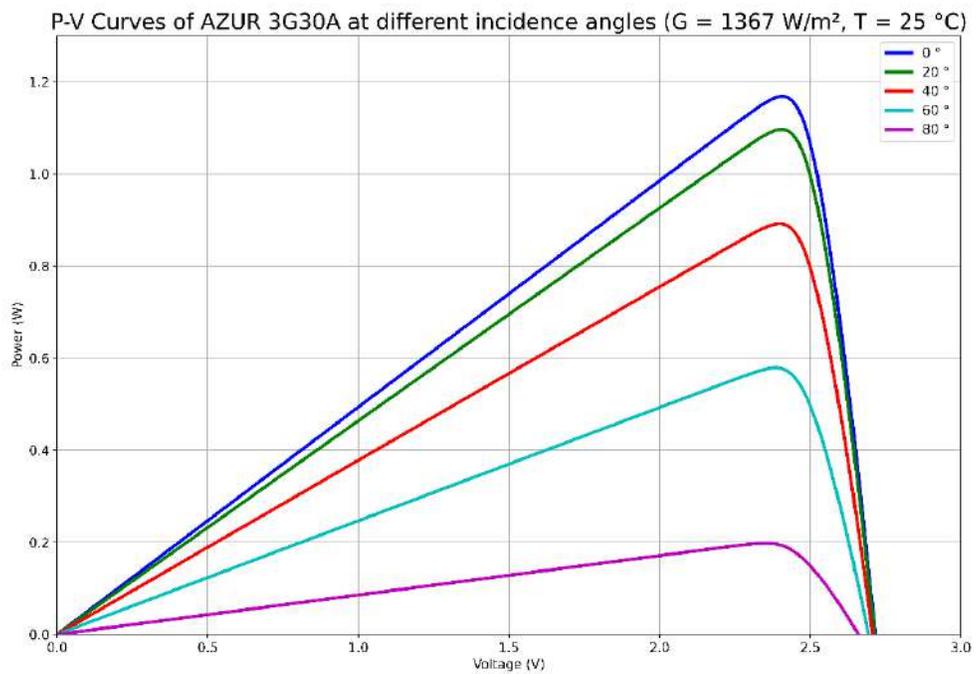
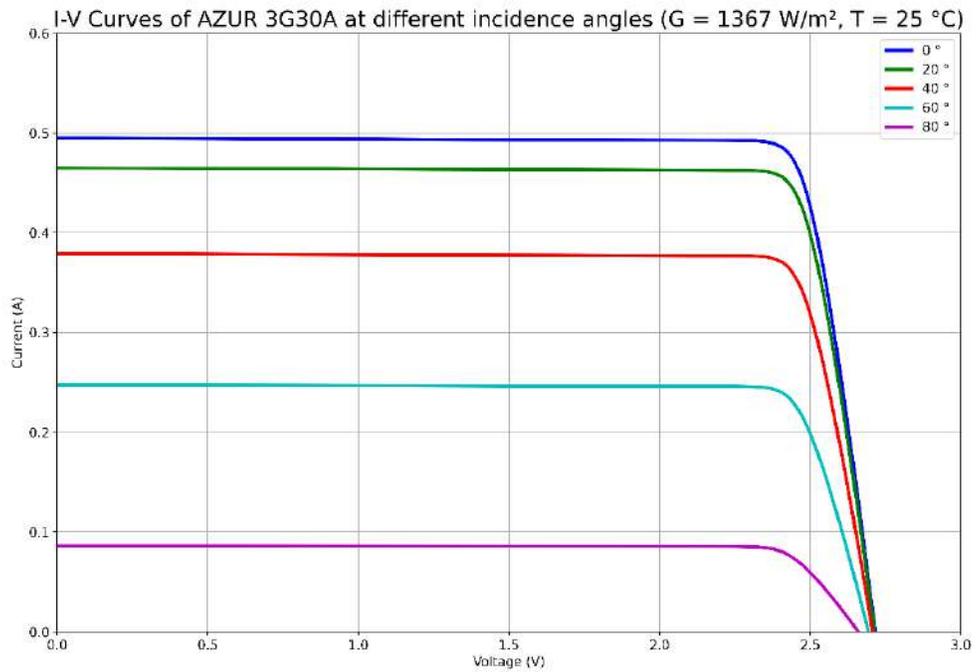
[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

AZUR SPACE 4G32C

32% Efficiency Quadruple Junction Solar Cell

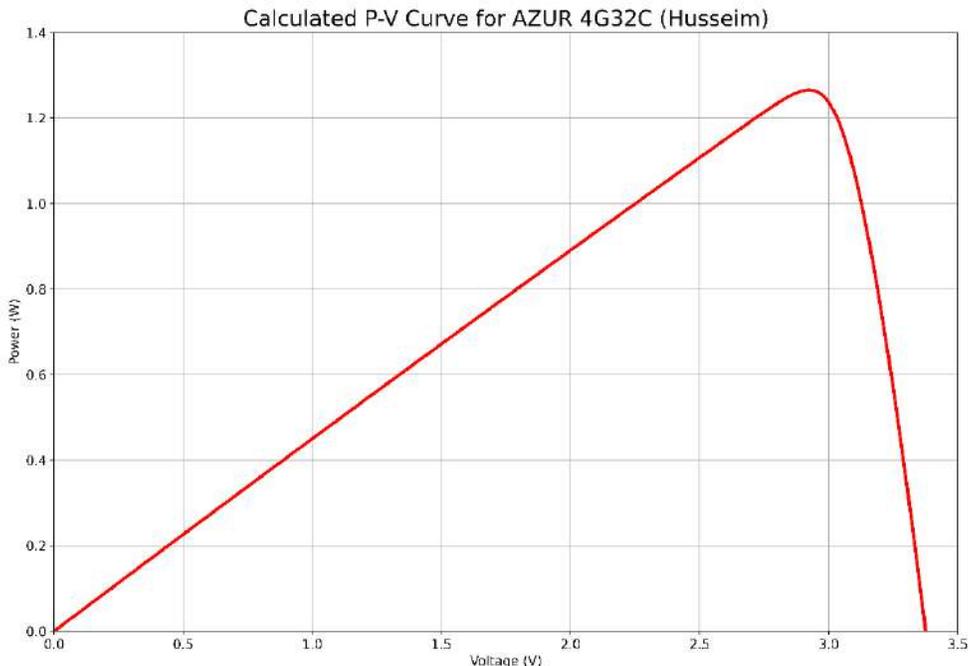
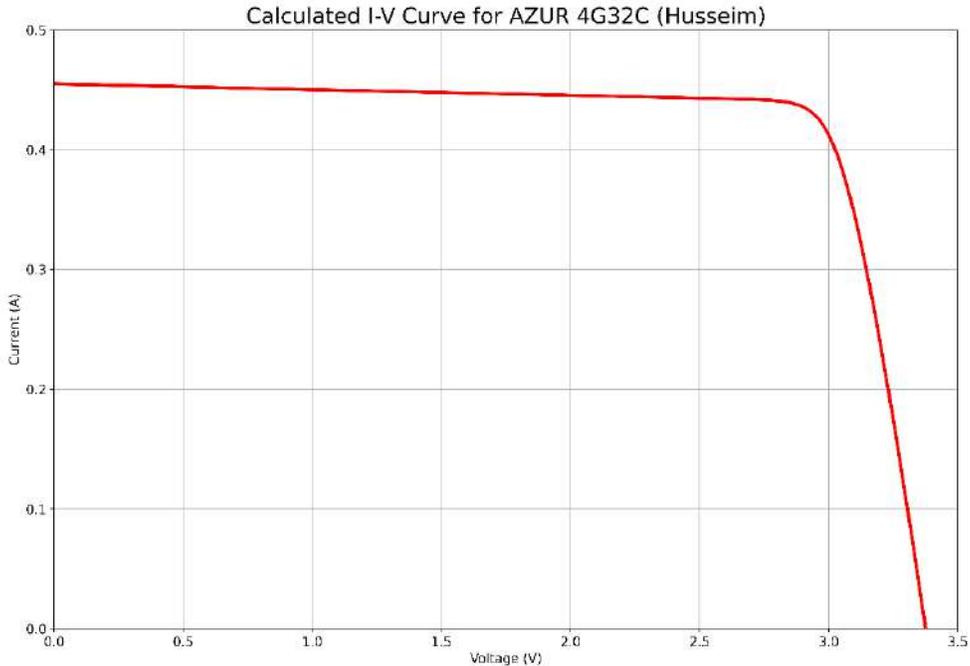
Electrical Data

Open Circuit Voltage	V_{oc}	3.375 V
Short Circuit Current	I_{sc}	0.455 A
MPP Voltage	V_{mp}	2.919 V
MPP Current	I_{mp}	0.434 A
MPP Power	P_{mp}	1.266 W
Number of cells in series	N_s	1
V_{oc} temperature coefficient	K_v	-8.4E-03 V/°C
I_{sc} temperature coefficient	K_i	7.0E-05 A/°C
Band gap	E_g	1.6 eV
Diode ideality factor	n	1.99
Photoelectric current	I_{ph}	0.4562 A
Diode inverse saturation current	I_o	9.45E-30 A
Series resistance	R_s	0.5705 Ω
Shunt resistance	R_{sh}	207.91 Ω
Root-Mean-Squared-Error*	RMSE	0.01537
Normalized-Root-Mean-Squared-Error*	nRMSE	0.03378
Coefficient of Determination*	R^2	0.94778
Mean-Absolute-Percentage-Error*	MAPE	2.24159 %

Spectrum: 1367 W/m²; T = 25 °C

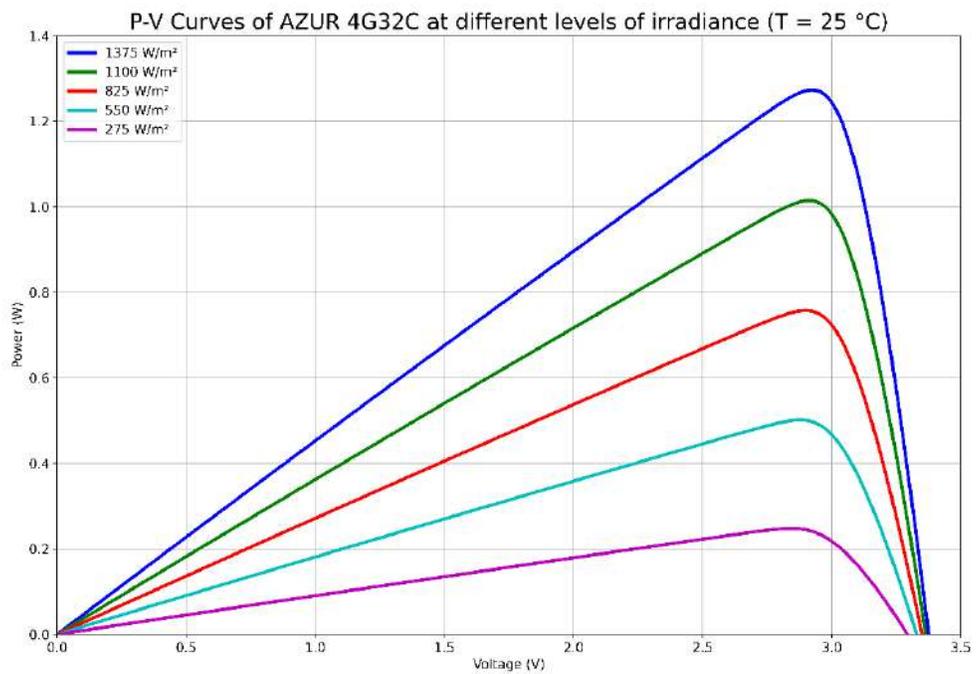
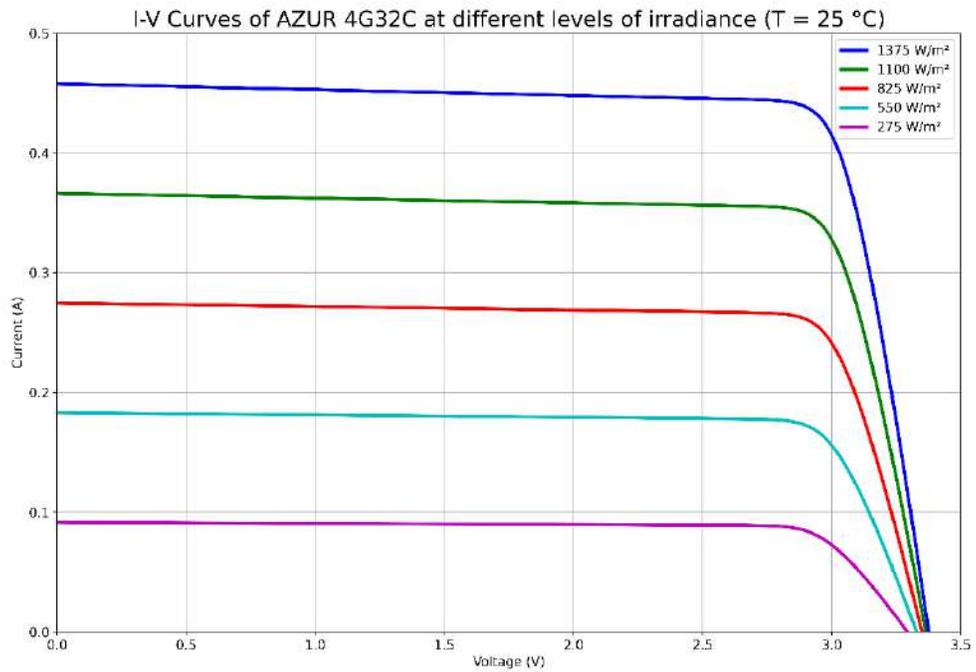
*Applies to the estimated values of n, I_{ph} , I_o , R_s and R_{sh}

Performance Data at Reference Conditions



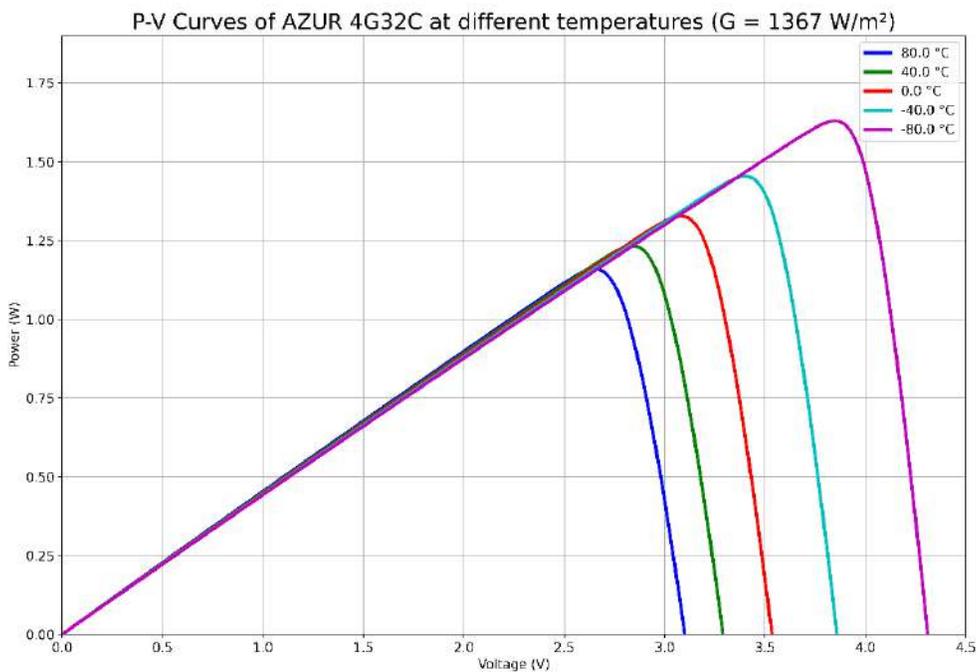
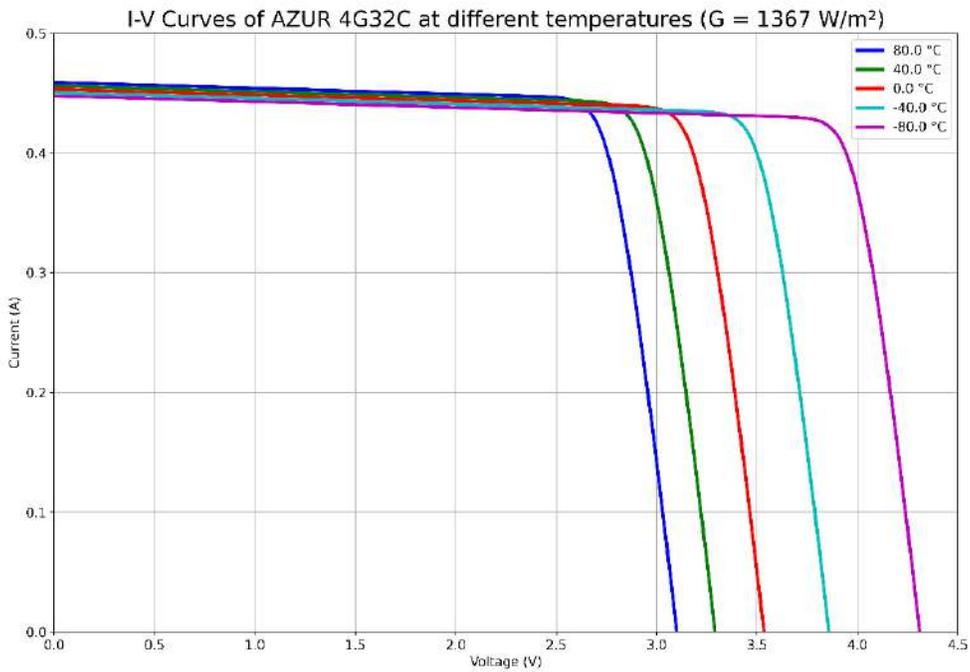
[Download datasets](#)

Performance Data at Different Conditions



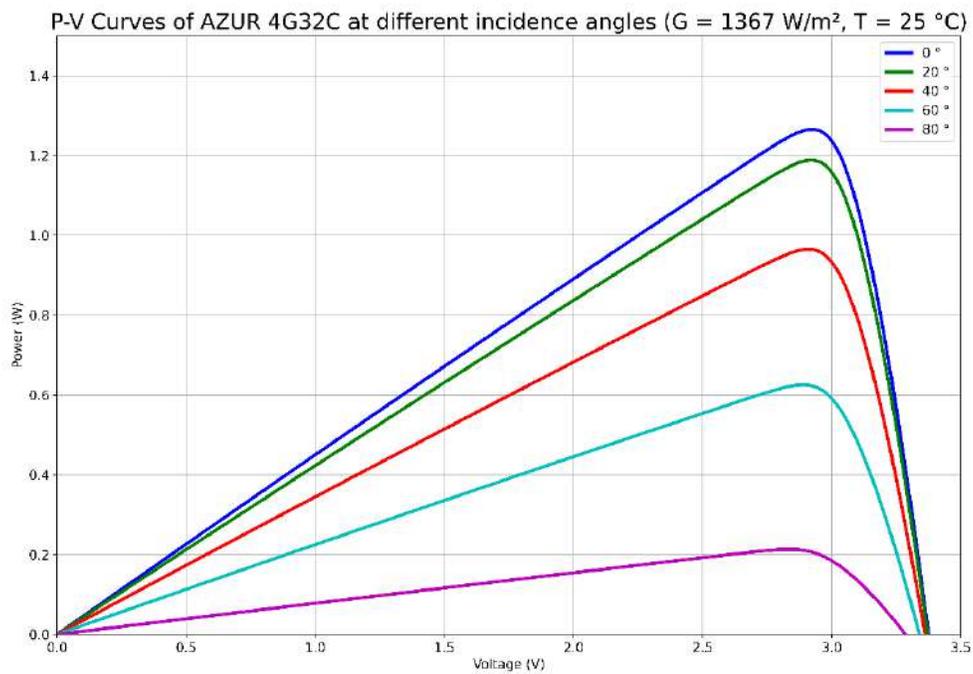
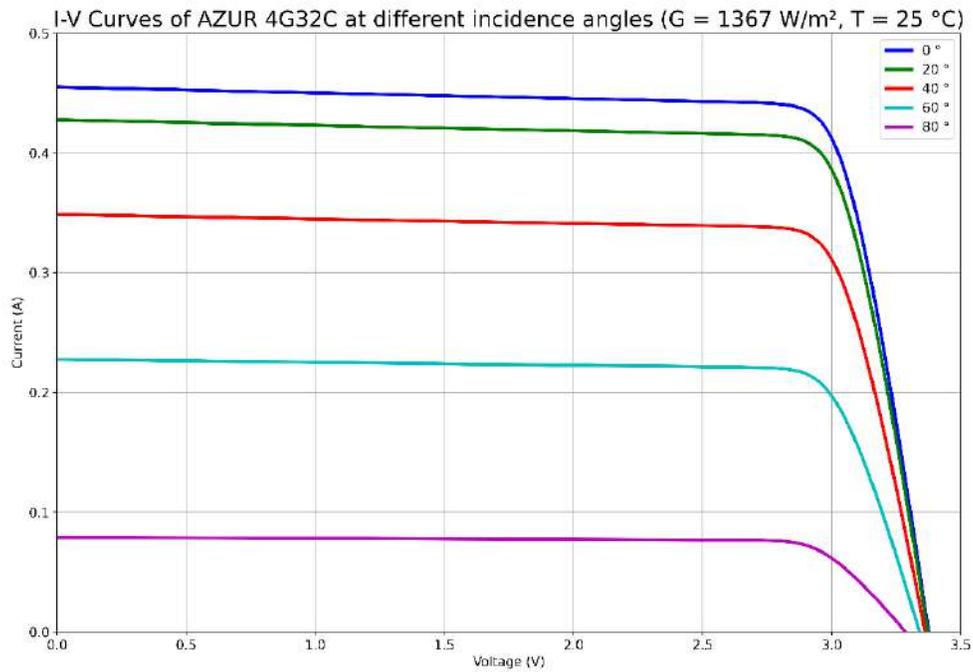
[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

CESI CTJ30

29.5% Efficiency Triple Junction Solar Cell

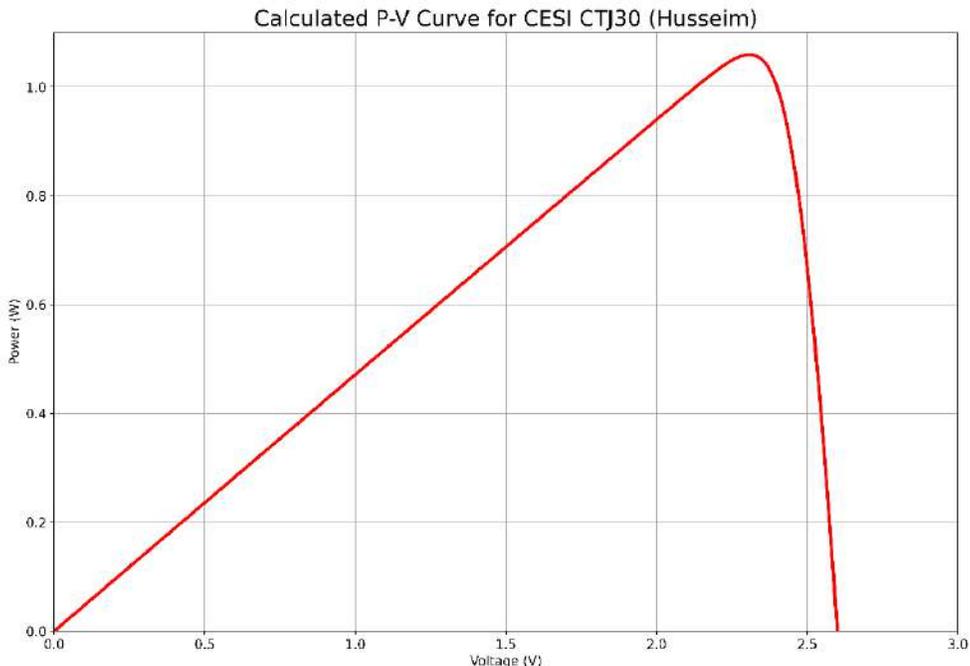
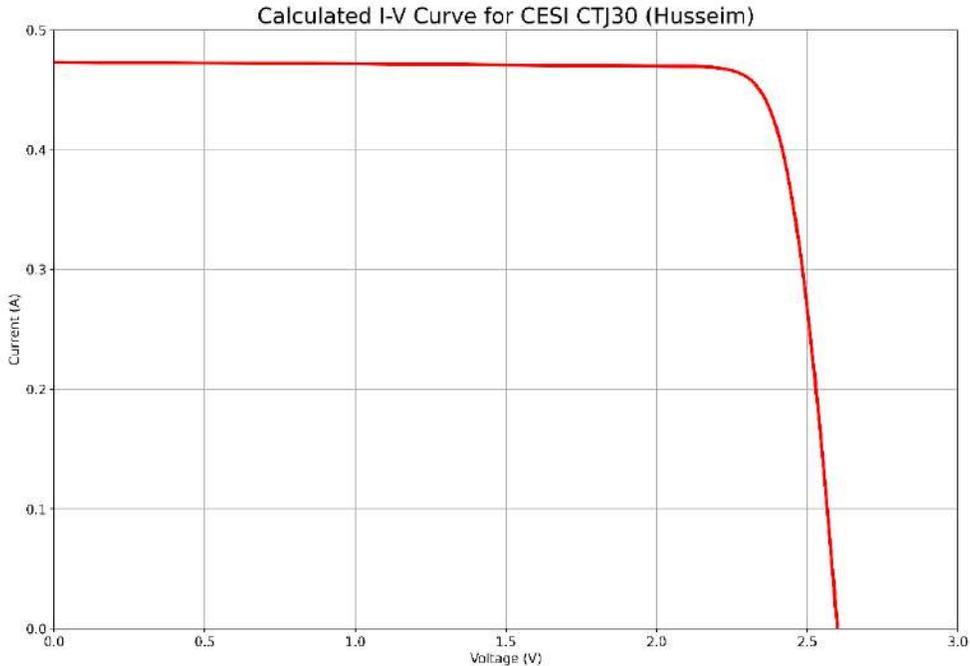
Electrical Data

Open Circuit Voltage	V_{oc}	2.600 V
Short Circuit Current	I_{sc}	0.473 A
MPP Voltage	V_{mp}	2.306 V
MPP Current	I_{mp}	0.459 A
MPP Power	P_{mp}	1.060 W
Number of cells in series	N_s	1
V_{oc} temperature coefficient	K_v	-8.4E-03 V/°C
I_{sc} temperature coefficient	K_i	7.0E-05 A/°C
Band gap	E_g	1.6 eV
Diode ideality factor	n	1.99
Photoelectric current	I_{ph}	0.4731 A
Diode inverse saturation current	I_o	3.86E-23 A
Series resistance	R_s	0.2132 Ω
Shunt resistance	R_{sh}	677.70 Ω
Root-Mean-Squared-Error*	RMSE	0.00919
Normalized-Root-Mean-Squared-Error*	nRMSE	0.01944
Coefficient of Determination*	R^2	0.97751
Mean-Absolute-Percentage-Error*	MAPE	1.06213 %

Spectrum: 1367 W/m²; T = 25 °C

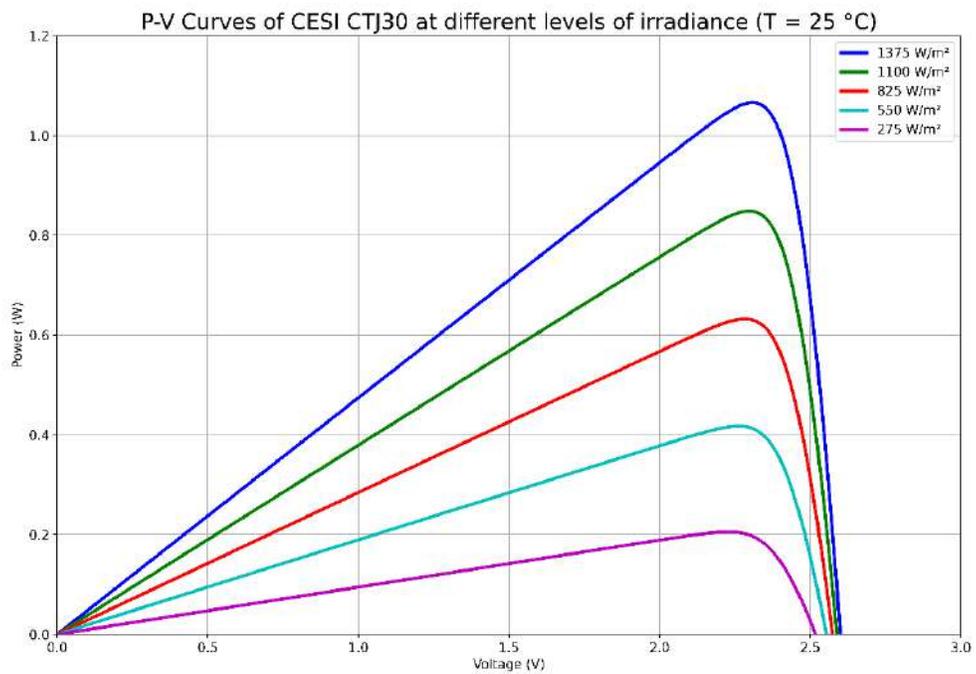
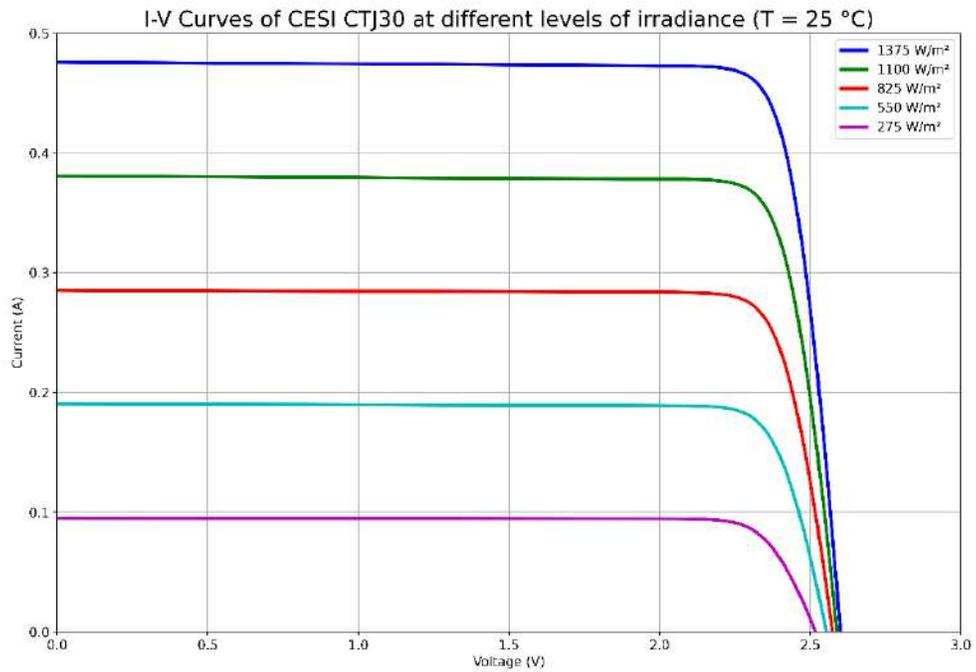
*Applies to the estimated values of n, I_{ph} , I_o , R_s and R_{sh}

Performance Data at Reference Conditions



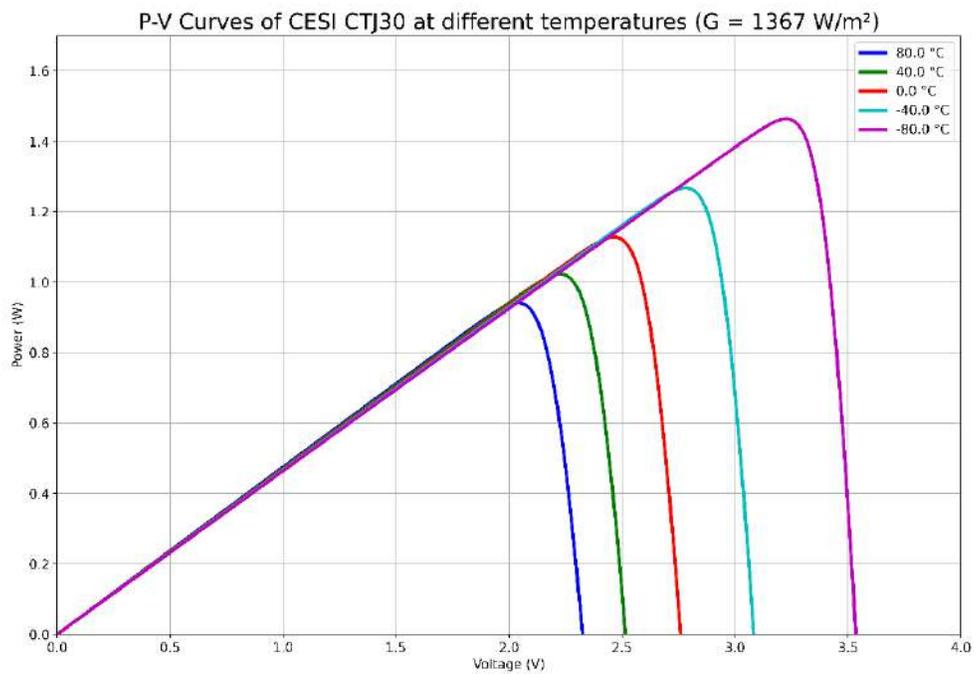
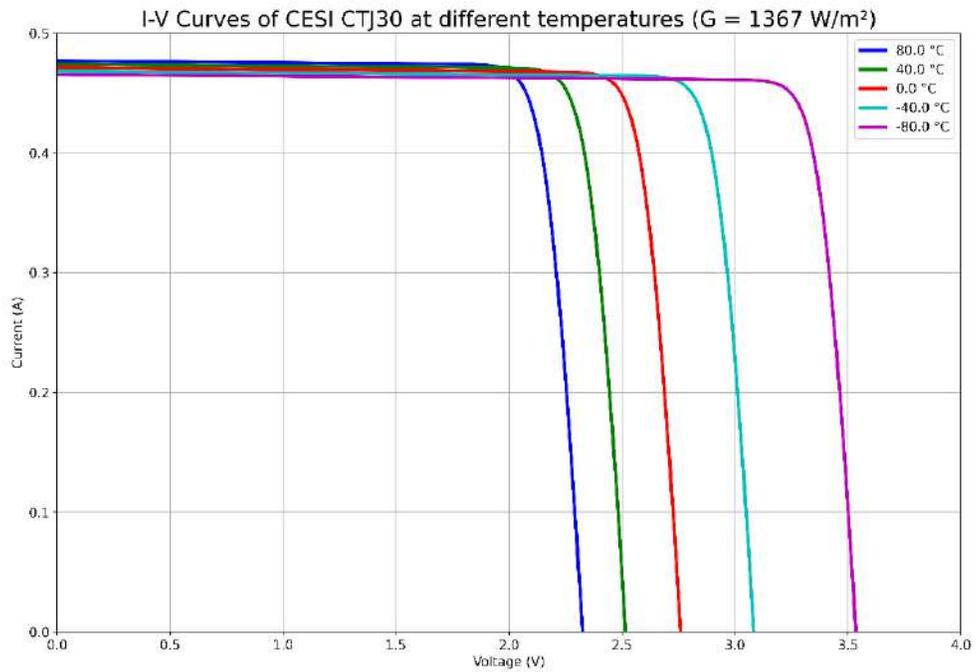
[Download datasets](#)

Performance Data at Different Conditions



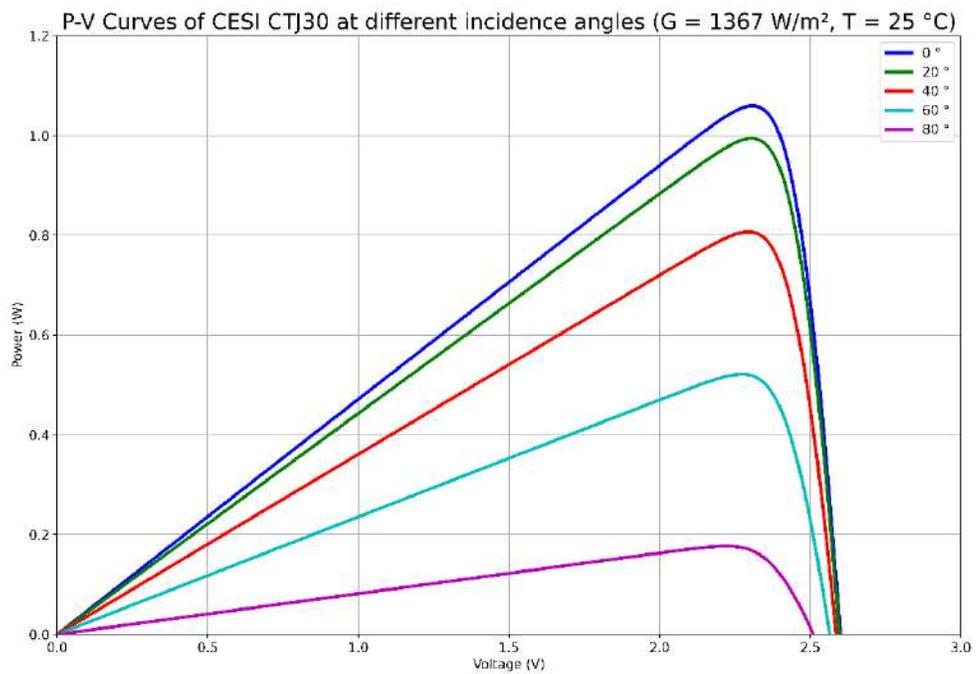
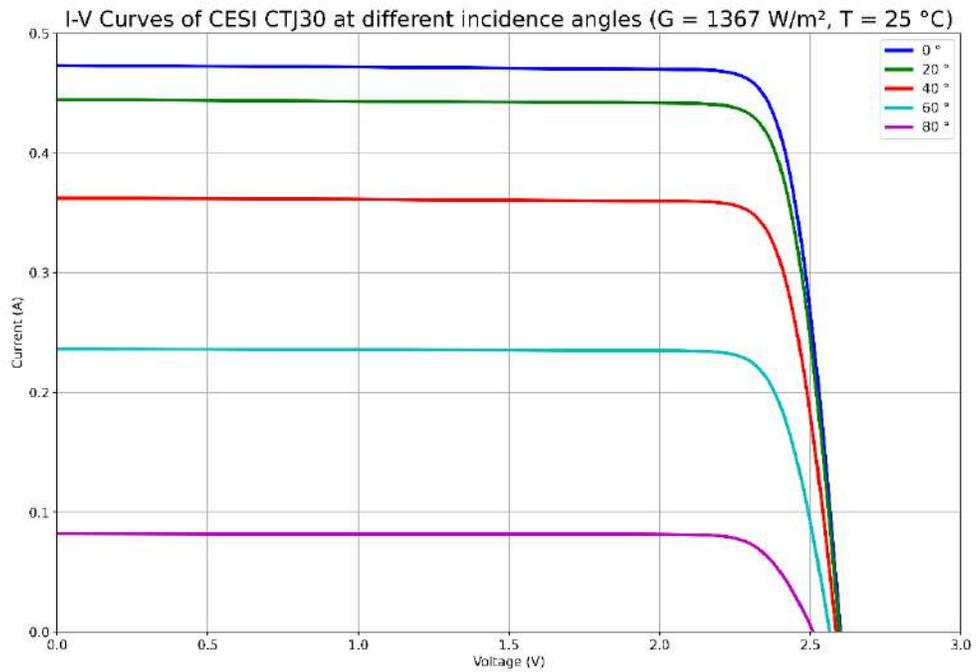
[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

SPECTROLAB UTJ

28.3% Efficiency Triple Junction Solar Cell

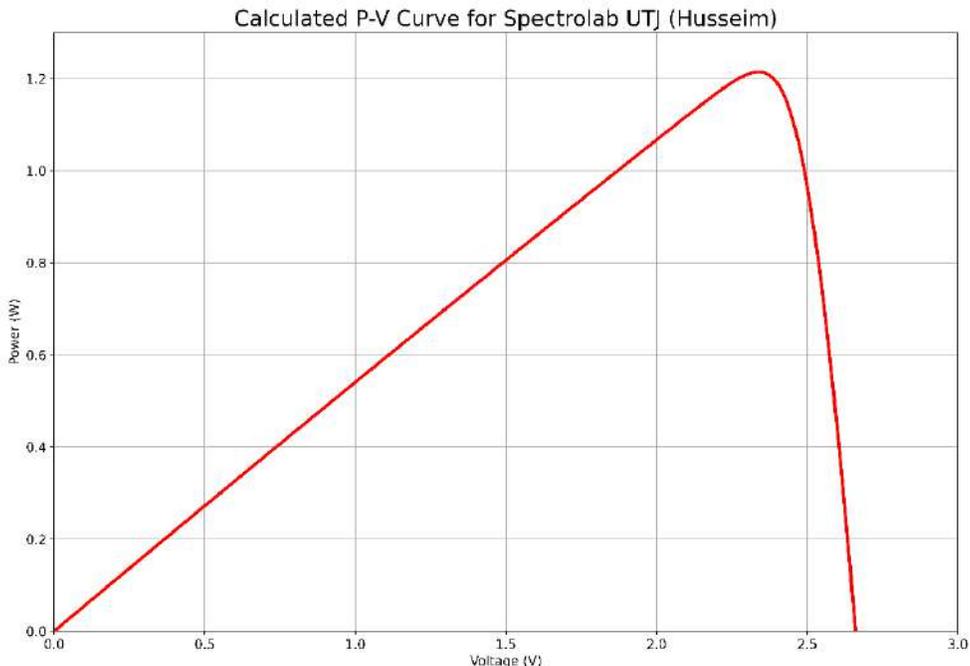
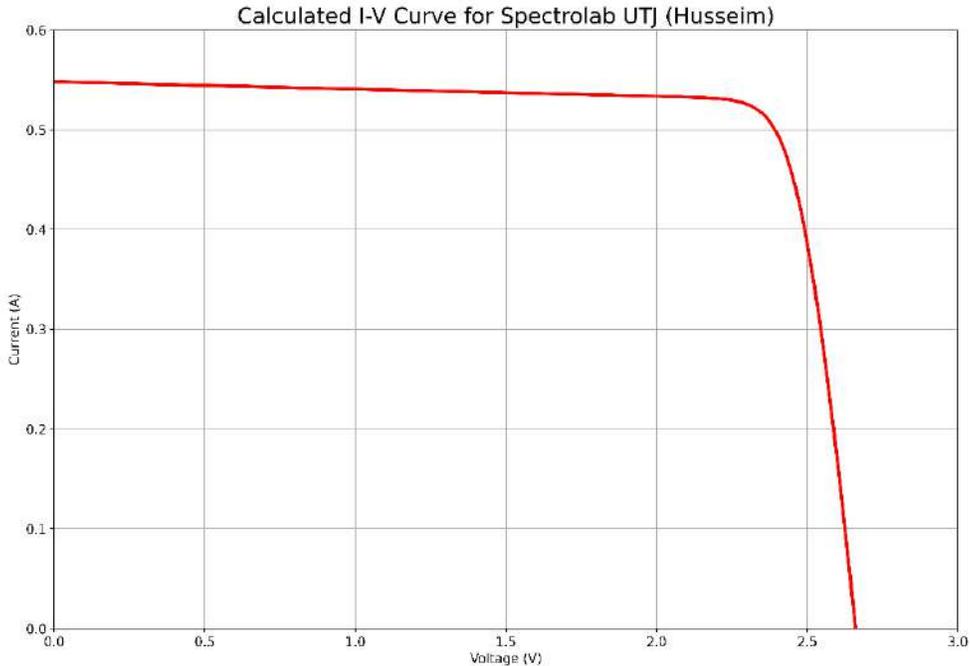
Electrical Data

Open Circuit Voltage	V_{oc}	2.600 V
Short Circuit Current	I_{sc}	0.548 A
MPP Voltage	V_{mp}	2.335 V
MPP Current	I_{mp}	0.520 A
MPP Power	P_{mp}	1.214 W
Number of cells in series	N_s	1
V_{oc} temperature coefficient	K_v	-5.9E-03 V/°C
I_{sc} temperature coefficient	K_i	3.8E-05 A/°C
Band gap	E_g	1.6 eV
Diode ideality factor	n	1.99
Photoelectric current	I_{ph}	0.5490 A
Diode inverse saturation current	I_o	2.26E-23 A
Series resistance	R_s	0.2408 Ω
Shunt resistance	R_{sh}	138.29 Ω
Root-Mean-Squared-Error*	RMSE	0.00840
Normalized-Root-Mean-Squared-Error*	nRMSE	0.01533
Coefficient of Determination*	R^2	0.98918
Mean-Absolute-Percentage-Error*	MAPE	1.51747 %

Spectrum: 1353 W/m²; T = 28 °C

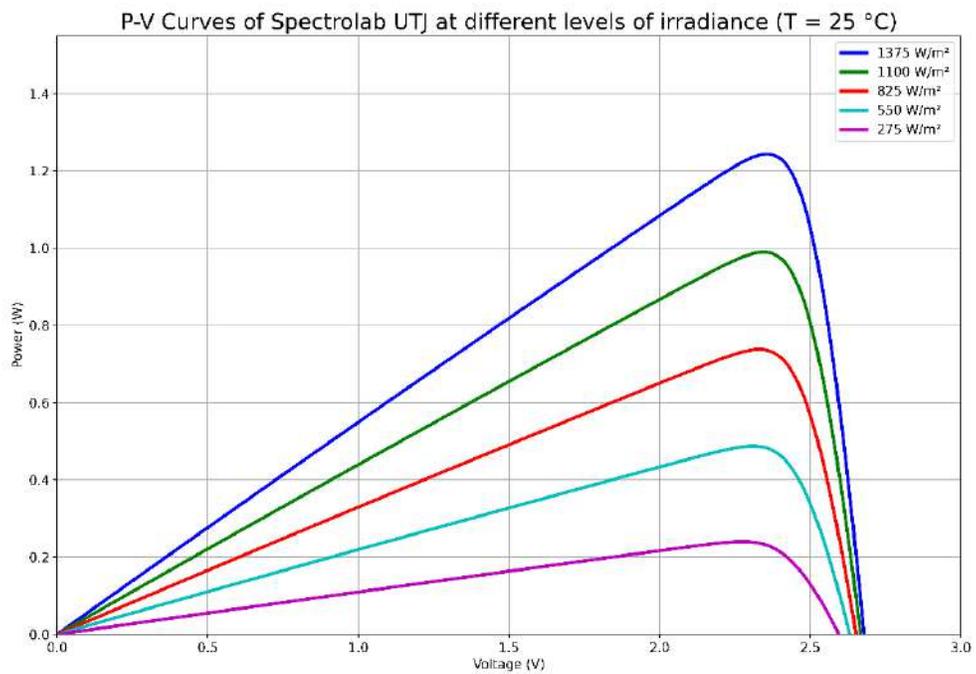
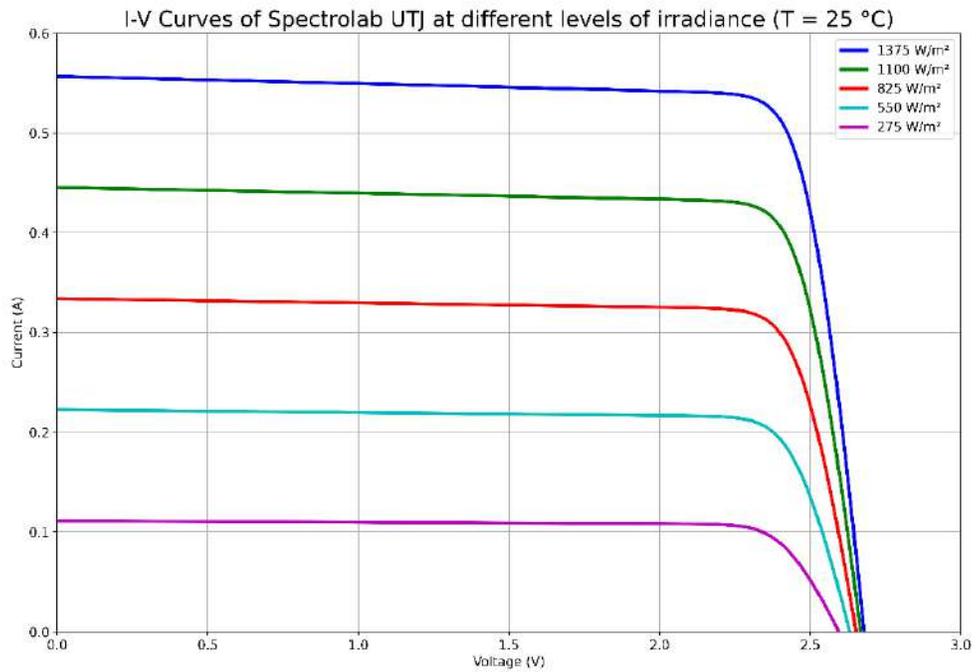
*Applies to the estimated values of n , I_{ph} , I_o , R_s and R_{sh}

Performance Data at Reference Conditions



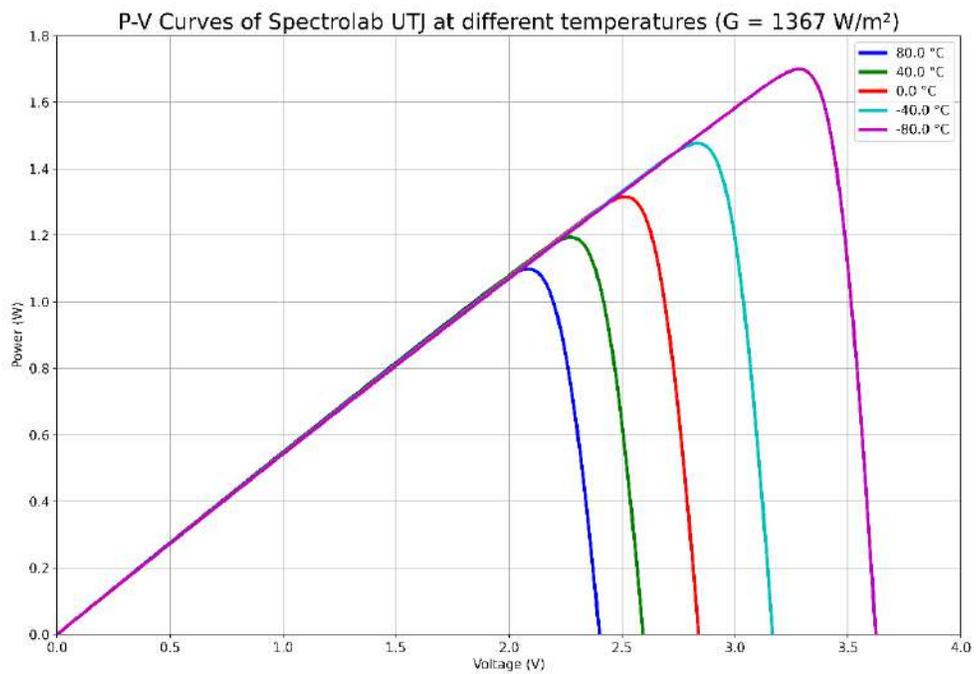
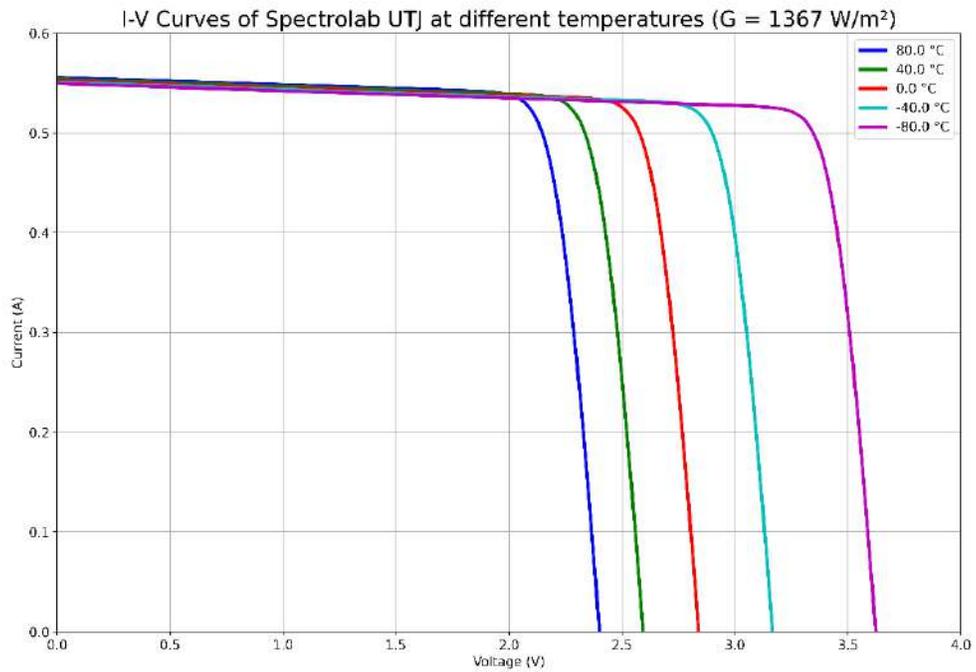
[Download datasets](#)

Performance Data at Different Conditions



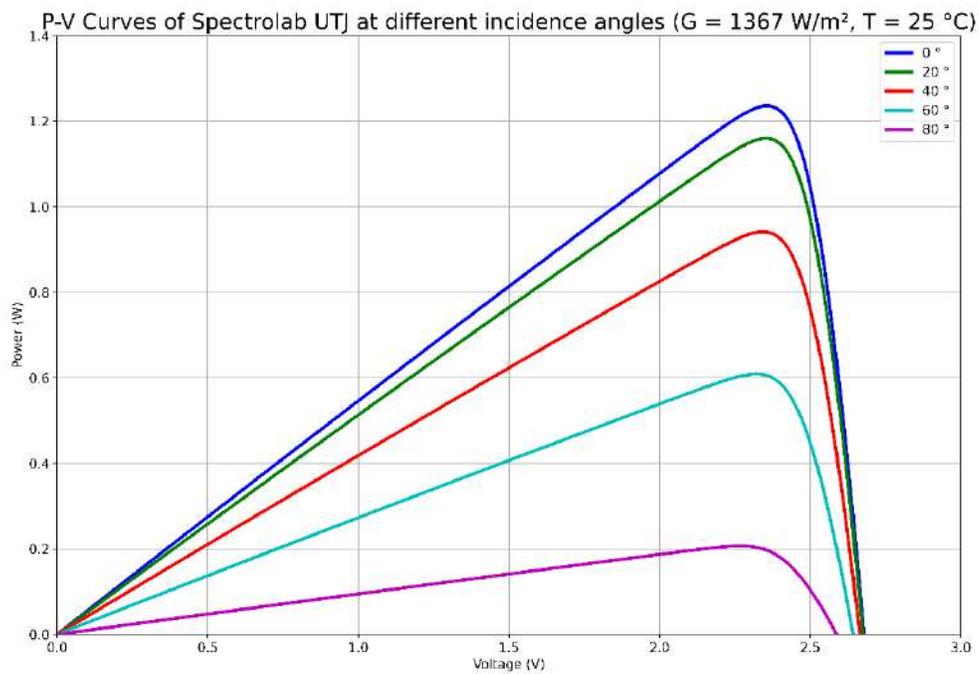
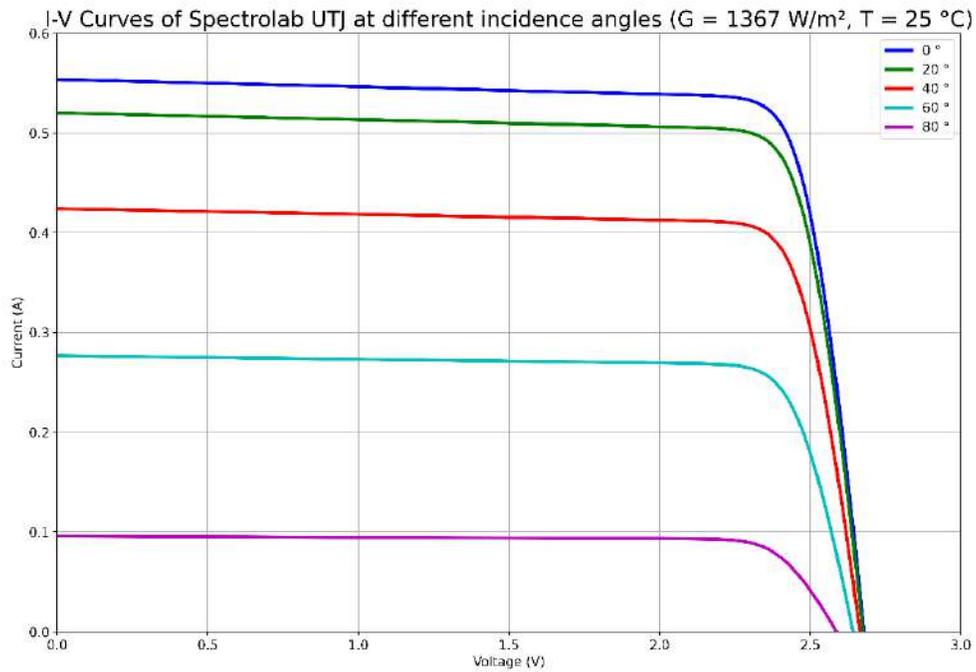
[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)

Performance Data at Different Conditions



[Download datasets](#)