

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



Análisis del presupuesto de potencia para un satélite tipo  
CubeSat 2U

Trabajo de graduación presentado por Derek Javier Matheu González  
para optar al grado académico de Licenciado en Ingeniería Mecánica  
Industrial

Guatemala,  
2022







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



Análisis del presupuesto de potencia para un satélite tipo  
CubeSat 2U

Trabajo de graduación presentado por Derek Javier Matheu González  
para optar al grado académico de Licenciado en Ingeniería Mecánica  
Industrial

Guatemala,  
2022



Vo.Bo.:



(f) \_\_\_\_\_  
Ing. Víctor Hugo Ayerdi

Tribunal Examinador:



(f) \_\_\_\_\_  
Ing. Víctor Hugo Ayerdi



(f) \_\_\_\_\_  
Ing. Rony Herrarte



(f) \_\_\_\_\_  
Ing. Gustavo Barrera

Fecha de aprobación: Guatemala, 4 de enero de 2022.



<b>Lista de figuras</b>	XI
<b>Lista de cuadros</b>	XIV
<b>Resumen</b>	XV
<b>Abstract</b>	XVII
<b>1. Introducción</b>	1
<b>2. Objetivos</b>	3
2.1. Objetivos generales	3
2.2. Objetivos específicos	3
<b>3. Justificación</b>	5
<b>4. Marco teórico</b>	7
4.1. Nano satélites <i>CubeSat</i>	7
4.2. Proyecto Quetzal-1	8
4.2.1. Recolección y almacenamiento de energía	8
4.2.2. Presupuesto de potencia	9
4.2.3. Módulos de consumo de potencia	9
4.2.4. Análisis del primer escenario del presupuesto de potencia	15
4.2.5. Análisis del segundo escenario del presupuesto de potencia	16
4.2.6. Análisis del tercer escenario de presupuesto de potencia	18
4.3. Análisis de resultados del Quetzal-1	18
4.3.1. Limpieza de datos	19
4.3.2. Análisis de Potencia Real	21
4.3.3. Análisis de efectos de temperatura.	22
4.3.4. Análisis del comportamiento de los módulos	24
4.3.5. Análisis de datos atípicos.	27

<b>5. Metodología</b>	<b>29</b>
5.1. Análisis del presupuesto de potencia para un <i>CubeSat 2U</i>	30
5.1.1. Análisis de potencia real del Quetzal-1	30
5.1.2. Análisis de efectos de temperatura sobre las baterías del Quetzal-1	30
5.1.3. Validación del presupuesto de consumo de potencia del Quetzal-1	30
5.1.4. Desarrollo de un presupuesto de potencia para un <i>CubeSat 2U</i>	31
<b>6. Resultados</b>	<b>41</b>
6.1. Resultados de potencia real del Quetzal-1	41
6.1.1. Resultados de potencia bajo efectos de luz solar	41
6.1.2. Resultados de potencia del Quetzal-1 bajo eclipse	43
6.2. Resultados de efectos de temperatura sobre las baterías del Quetzal-1	44
6.3. Validación del presupuesto de consumo de potencia del Quetzal-1	45
6.3.1. Resultados del consumo de potencia del módulo <i>ADCS</i>	45
6.3.2. Resultados del consumo de potencia del módulo <i>COMMS</i>	46
6.3.3. Resultados del consumo de potencia del módulo <i>Payload</i>	47
6.3.4. Resultados del consumo de potencia del primer y segundo escenario	48
6.3.5. Resultados del consumo de potencia del tercer escenario	49
6.3.6. Resultados del estado de carga de las baterías del Quetzal-1	50
6.4. Desarrollo de un presupuesto de potencia para un <i>CubeSat 2U</i>	50
6.4.1. Simulación de generación de potencia total por los paneles solares para un <i>CubeSat 2U</i>	50
6.4.2. Simulación de los escenarios del presupuesto de potencia para un <i>CubeSat 2U</i>	51
6.4.3. Resultados del primer escenario del presupuesto de potencia para un <i>CubeSat 2U</i>	51
6.4.4. Resultados del segundo escenario del presupuesto de potencia para un <i>CubeSat 2U</i>	54
6.4.5. Resultados del tercer escenario del presupuesto de potencia para un <i>CubeSat 2U</i>	57
6.4.6. Resultados del cuarto escenario del presupuesto de potencia para un <i>CubeSat 2U</i>	59
<b>7. Discusión de resultados</b>	<b>63</b>
7.1. Análisis de potencia real del Quetzal-1	63
7.1.1. Análisis de generación de potencia por las baterías bajo efectos de luz solar	63
7.1.2. Análisis de consumo de potencia por las baterías bajo efectos de luz solar	64
7.1.3. Análisis de consumo de potencia por las baterías del Quetzal-1 bajo eclipse	65
7.2. Análisis de efectos de temperatura sobre las baterías del Quetzal-1	65
7.3. Validación del presupuesto de consumo de potencia del Quetzal-1	66
7.3.1. Análisis del consumo de potencia del módulo <i>ADCS</i>	66
7.3.2. Análisis del consumo de potencia del módulo <i>COMMS</i>	67
7.3.3. Análisis del consumo de potencia del módulo <i>Payload</i>	67
7.3.4. Análisis del consumo de potencia del primer y segundo escenario	67
7.3.5. Análisis del consumo de potencia del tercer escenario	68

7.3.6. Análisis del estado de carga de las baterías del Quetzal-1 . . . . .	68
7.4. Desarrollo de un presupuesto de potencia para un <i>CubeSat 2U</i> . . . . .	69
7.4.1. Simulación de generación de potencia total por los paneles solares para un <i>CubeSat 2U</i> . . . . .	69
7.4.2. Análisis del primer escenario del presupuesto de potencia para un <i>CubeSat 2U</i> . . . . .	69
7.4.3. Análisis del segundo escenario del presupuesto de potencia para un <i>CubeSat 2U</i> . . . . .	70
7.4.4. Análisis del tercer escenario del presupuesto de potencia para un <i>CubeSat 2U</i> . . . . .	70
7.4.5. Análisis del cuarto escenario del presupuesto de potencia para un <i>CubeSat 2U</i> . . . . .	70
<b>8. Conclusiones</b>	<b>73</b>
<b>9. Recomendaciones</b>	<b>75</b>
<b>10. Bibliografía</b>	<b>77</b>
<b>11. Anexos</b>	<b>79</b>



---

Lista de figuras

---

1.	Dimensiones de <i>CubeSat</i> desde 1U hasta 16U . . . . .	7
2.	Simulación del consumo de <i>ADCS</i> durante una órbita . . . . .	10
3.	Simulación de consumo de potencia <i>Payload</i> durante una órbita . . . . .	11
4.	Simulación del consumo del módulo <i>ADM</i> durante una órbita . . . . .	12
5.	Simulación de consumo de potencia del <i>COMMS</i> durante una órbita . . . . .	13
6.	Simulación de consumo de potencia de <i>OBC</i> durante una órbita . . . . .	14
7.	Simulación del consumo de potencia del Quetzal-1 durante el primer escenario . . . . .	15
8.	Simulación del estado de carga de las baterías durante primer escenario . . . . .	16
9.	Simulación del consumo de potencia del Quetzal-1 durante el segundo escenario . . . . .	17
10.	Simulación del estado de carga de las baterías durante segundo escenario . . . . .	17
11.	Simulación del consumo de potencia de Quetzal-1 durante el tercer escenario . . . . .	18
12.	Gráfica de datos inicial. . . . .	19
13.	Primera limpieza de datos . . . . .	20
14.	Datos de falla con subsistema <i>EPS</i> . . . . .	20
15.	Falla con el monitor de las baterías. . . . .	20
16.	Potencia generada bajo luz solar. . . . .	21
17.	Datos de los fotodiodos. . . . .	22
18.	Capacidad de carga durante almacenamiento prolongado, en función de la temperatura . . . . .	22
19.	Disminución momentanea de capacidad por enfriamiento de las celdas . . . . .	23
20.	Histograma de los registros del cambio de temperatura de las baterías. . . . .	24
21.	Histograma del estado de carga de las baterías. . . . .	24
22.	Frecuencia del consumo de corriente para el módulo <i>ADCS</i> . . . . .	25
23.	Frecuencia del consumo de corriente para el módulo <i>COMMS</i> . . . . .	26
24.	Frecuencia del consumo de corriente para el módulo <i>Payload</i> . . . . .	26
25.	Consumo de potencia de <i>ADCS</i> durante 30 segundos . . . . .	32
26.	Consumo de potencia de <i>Payload</i> en una órbita . . . . .	33
27.	Consumo de potencia de <i>ADM</i> . . . . .	34
28.	Consumo de potencia de <i>ADM</i> en una órbita . . . . .	34
29.	Consumo de potencia de <i>COMMS</i> durante 30 segundos . . . . .	35
30.	Consumo de potencia de <i>OBC</i> durante 30 segundos . . . . .	36
31.	Consumo de potencia de <i>EPS</i> durante 30 segundos . . . . .	37

32.	Potencia generada por las baterías bajo efectos de luz solar	41
33.	Consumo de potencia por las baterías bajo efectos de luz solar	42
34.	<i>Boxplot</i> del consumo de potencia por las baterías bajo efectos de luz solar	42
35.	Consumo de potencia por las baterías bajo eclipse	43
36.	<i>Boxplot</i> del consumo de potencia por las baterías bajo eclipse	43
37.	Histograma los rangos de temperatura de las baterías del Quetzal-1	44
38.	<i>Boxplot</i> los rangos de temperatura de las baterías del Quetzal-1	44
39.	<i>Boxplot</i> del consumo de potencia real del módulo <i>ADCS</i> durante 26 semanas	45
40.	Histograma del consumo de potencia teórico del módulo <i>ADCS</i> durante las simulaciones	45
41.	Histograma del consumo de potencia real del módulo <i>COMMS</i> durante 26 semanas	46
42.	Histograma del consumo de potencia real del módulo <i>COMMS</i> , en modalidad <i>RX</i> , durante 26 semanas	46
43.	Histograma del consumo de potencia real del módulo <i>COMMS</i> , en modalidad <i>TX-RX</i> , durante 26 semanas	46
44.	Histograma del consumo de potencia teórico del módulo <i>COMMS</i> durante las simulaciones	47
45.	<i>Boxplot</i> del consumo de potencia real del módulo <i>Payload</i> durante 26 semanas	47
46.	Histograma del consumo de potencia real del Quetzal-1 durante el primer y segundo escenario	48
47.	Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el primer escenario, durante una órbita	48
48.	Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el segundo escenario, durante una órbita	48
49.	Histograma del consumo de potencia real del Quetzal-1 durante el tercer escenario	49
50.	Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el tercer escenario, durante una órbita	49
51.	Histograma de la simulación de la generación de potencia de los paneles solares de un <i>Cubesat 2U</i> , durante un día	50
52.	Simulación del consumo de potencia total, por los módulos, durante una órbita	51
53.	Histograma del consumo de potencia durante el primer escenario	52
54.	Histograma de la potencia de entrada a las baterías	52
55.	<i>Boxplot</i> de la potencia de entrada a las baterías	53
56.	Gráfico de líneas del estado de carga de las baterías durante el primer escenario	53
57.	Consumo de potencia total durante el segundo escenario	54
58.	Histograma del consumo de potencia total durante el segundo escenario	54
59.	Histograma de la potencia generada por las baterías durante el segundo escenario	55
60.	<i>Boxplot</i> de la potencia generada por las baterías durante el segundo escenario	55
61.	Gráfico de líneas del estado de carga de las baterías durante el segundo escenario	56
62.	Consumo de potencia total durante el tercer escenario	57
63.	Histograma del consumo de potencia total durante el tercer escenario	57
64.	Histograma de la potencia generada por las baterías durante el tercer escenario	58
65.	<i>Boxplot</i> de la potencia generada por las baterías durante el tercer escenario	58
66.	Gráfico de líneas del estado de carga de las baterías durante el tercer escenario	59
67.	Consumo de potencia total durante el cuarto escenario	59
68.	Histograma del consumo de potencia total durante el cuarto escenario	60

69. Histograma de la potencia generada por las baterías durante el cuarto escenario	60
70. <i>Boxplot</i> de la potencia generada por las baterías durante el cuarto escenario	61
71. Gráfico de líneas del estado de carga de las baterías durante el cuarto escenario	61



1. Estimación de consumo de potencia de <i>ADCS</i> . . . . .	10
2. Estimación de consumo de potencia del <i>Payload</i> . . . . .	11
3. Estimación de consumo de potencia del <i>ADM</i> . . . . .	12
4. Estimación de consumo de potencia del <i>COMMS</i> . . . . .	13
5. Estimación de consumo de potencia del <i>OBC</i> . . . . .	14
6. Estimación de consumo de potencia del <i>EPS</i> . . . . .	15
7. Subsistemas activos durante primer escenario . . . . .	15
8. Subsistemas activos durante segundo escenario . . . . .	16
9. Subsistemas activos durante tercer escenario . . . . .	18
10. Parámetros de consumo de potencia de <i>ADCS</i> . . . . .	32
11. Parámetros de consumo de potencia de <i>Payload</i> . . . . .	33
12. Parámetros de consumo de potencia de <i>ADM</i> . . . . .	33
13. Parámetros de consumo de potencia de <i>COMMS</i> . . . . .	35
14. Parámetros de consumo de potencia de <i>OBC</i> . . . . .	36
15. Parámetros de consumo de potencia de <i>EPS</i> . . . . .	37
16. Parámetros de consumo de potencia de <i>Deorbit</i> . . . . .	38
17. Especificaciones de la celda solar utilizada en la simulación . . . . .	38
18. Especificaciones de las baterías de Polimero de litio . . . . .	39
19. Módulos que se encuentran activados durante la simulación de los escenarios, para presupuesto de potencia de un <i>CubeSat 2U</i> . . . . .	39
20. Estado de los módulos durante la simulación del primer escenario, para pre- supuesto de potencia de un <i>CubeSat 2U</i> . . . . .	40
21. Estado de los módulos durante la simulación del segundo escenario, para presupuesto de potencia de un <i>CubeSat 2U</i> . . . . .	40
22. Estado de los módulos durante la simulación del tercer escenario, para presu- puesto de potencia de un <i>CubeSat 2U</i> . . . . .	40
23. Estado de los módulos durante la simulación del tercer escenario, para el presupuesto de potencia de un <i>CubeSat 2U</i> . . . . .	40
24. Parámetros descriptivos de la potencia generada por baterías bajo efectos de luz solar . . . . .	41
25. Longitud de los datos de generación de potencia, bajo efectos de luz solar . . . . .	42

26. Parámetros descriptivos del consumo potencia por las baterías bajo efectos de luz solar . . . . .	42
27. Longitud de los datos de generación y consumo de potencia, bajo efectos de luz solar . . . . .	43
28. Parámetros descriptivos del consumo potencia por las baterías bajo eclipse . . . . .	43
29. Longitud de los datos de potencia generales del análisis . . . . .	44
30. Parámetros descriptivos los rangos de temperatura de las baterías del Quetzal-1 . . . . .	44
31. Resultados reales y teóricos del consumo de potencia del módulo <i>ADCS</i> . . . . .	45
32. Resultados reales y teóricos del consumo de potencia del módulo <i>COMMS</i> . . . . .	47
33. Resultados reales y teóricos del consumo de potencia del módulo <i>Payload</i> . . . . .	47
34. Parámetros descriptivos, reales y teóricos, del consumo potencia del Quetzal-1 durante el primer y segundo escenario . . . . .	49
35. Parámetros descriptivos, reales y teóricos, del consumo potencia del Quetzal-1 durante el tercer escenario . . . . .	50
36. Frecuencia de un estado de carga de las baterías igual al 100 %, durante la misión . . . . .	50
37. Parámetros teóricos descriptivos de la generación de potencia de un <i>Cubesat 2U</i> , durante un día . . . . .	51
38. Resumen de los escenarios analizados de las simulaciones . . . . .	51
39. Parámetros del consumo de potencia en el primer escenario . . . . .	52
40. Parámetros descriptivos de la potencia de entrada a las baterías . . . . .	53
41. Parámetros relevantes del análisis del estado de carga de las baterías . . . . .	54
42. Parámetros del consumo de potencia en el segundo escenario . . . . .	55
43. Parámetros descriptivos de la potencia de entrada a las baterías . . . . .	56
44. Parámetros relevantes del análisis del estado de carga de las baterías durante el segundo escenario . . . . .	56
45. Parámetros del consumo de potencia durante el tercer escenario . . . . .	57
46. Parámetros descriptivos de la potencia de entrada a las baterías . . . . .	58
47. Parámetros relevantes del análisis del estado de carga de las baterías durante el tercer escenario . . . . .	59
48. Parámetros del consumo de potencia durante el cuarto escenario . . . . .	60
49. Parámetros descriptivos de la potencia generada por las baterías . . . . .	61
50. Parámetros relevantes del análisis del estado de carga de las baterías durante el cuarto escenario . . . . .	62

Este trabajo consistió en realizar un análisis del presupuesto de potencia para un *CubeSat 2U*, se analizaron los resultados de la generación y consumo de potencia del Quetzal-1, los efectos de temperatura sobre las baterías y también se analizaron los resultados del presupuesto de potencia del Quetzal-1. Para obtener los resultados del presupuesto de potencia del Quetzal-1 se compararon los resultados teóricos contra los resultados reales del consumo de potencia de los módulos *ADCS*, *COMMS* y *Payload*.

Contando con los consumos de potencia de los módulos analizados, se utilizaron para realizar simulaciones del presupuesto de potencia de un *CubeSat 2U*, considerando la integración del módulo *Deorbit System*. Las simulaciones constaron de cuatro escenarios esperados durante el vuelo, en los cuales se obtuvieron los resultados del consumo de potencia, generación de potencia y el estado de carga de las baterías.

Se obtuvo que los resultados de la generación de potencia real del Quetzal-1 representan el 73.85 % del tiempo de las 26 semanas analizadas. En el caso del consumo de potencia del Quetzal-1, se obtuvo que no tiene un impacto negativo en el estado de carga de las baterías. Los rangos de temperatura de las baterías no tienen una relación lineal con la pérdida de energía. Los resultados del presupuesto de potencia para un *CubeSat 2U* muestran que se cuenta con la capacidad de generación de potencia para implementar nuevos componentes que sean utilizados en nuevos estudios.



This work consisted of carrying out an analysis of the power budget for a CubeSat 2U, the results of the generation and power consumption of the Quetzal-1, the effects of temperature on the batteries and the results of the Quetzal-1 power budget. To obtain the results of the Quetzal-1 power budget, the theoretical results were compared against the real results of the power consumption of the ADCS, COMMS and Payload modules.

Counting on the power consumption of the analyzed modules, they were used to simulate the power budget of a CubeSat 2U, considering the integration of the Deorbit System module. The simulations consisted of four scenarios expected during the flight, in which the results of power consumption, power generation and the state of charge of the batteries were obtained.

It was obtained that the results of the real power generation of Quetzal-1 represent 73.85 % of the time of the 26 weeks analyzed. In the case of the power consumption of the Quetzal-1, it was obtained that it does not have a negative impact on the state of charge of the batteries. Battery temperature ranges do not have a linear relationship with power loss. The results of the power budget for a CubeSat 2U show that the power generation capacity is available to implement new components that are used in new studies.



Los satélites tipo *CubeSat* son nanosatélites que se componen de múltiples módulos cúbicos de 10 cm x 10 cm x 10 cm, programados para realizar una misión específica. La simplificación en su diseño permite que su manufactura sea de bajo costo. El Quetzal-1 fue el primer CubeSat guatemalteco lanzado al espacio el 6 de marzo y fue liberado el 28 de abril del 2020, finalizando su misión 211 días después.

El satélite Quetzal-1, fue desarrollado para probar un sensor multiespectral y fue conformado por los siguientes módulos: A. Sistema de Determinación y Control de Actitud; éste tuvo el objetivo de estabilizar el Quetzal-1. B. Carga Útil; utilizó un motor que mueve un carrusel que cuenta con una cámara monocromática. C. Mecanismo de Despliegue de Antena; su función fue desplegar 4 antenas de comunicación. D. Comunicación; su función fue recibir y transmitir datos a la estación terrestre. E. Computadora Abordo; esta computadora operó los módulos del Quetzal-1. F. Sistema de Potencia Eléctrica; su función fue recolectar, almacenar y distribuir energía.

La recolección de energía fue mediante paneles solares que capturan y convierten parte de la energía solar en energía eléctrica. El almacenamiento de energía fue compuesto por un arreglo de 2 baterías conectadas en paralelo, las cuales suministraron energía a los componentes cuando no estuvo disponible la energía solar. El Quetzal-1 utilizó un calentador que tuvo la función de mantener la temperatura de las baterías entre 0 °C y 25 °C, debido a que las baterías presentan una disminución de la capacidad a medida que la temperatura de almacenamiento excede ambos límites.

Los satélites tipo *CubeSat* requieren de una selección del hardware apropiado para la composición de los módulos; por lo tanto, se deben determinar los requerimientos de potencia de estos componentes. Este procedimiento se denomina presupuesto de potencia. El presupuesto de potencia del Quetzal-1 consistió en realizar simulaciones para determinar el consumo de energía, para ello se consideraron tres escenarios por medio de los cuales se encontraron activados durante las simulaciones los siguientes módulos: Sistema de Potencia Eléctrica, Sistema de Determinación y Control de Actitud y Computadora Abordo.

Primer escenario: se simuló la activación de una secuencia del despliegue de las antenas de comunicación, Carga Útil se encontró desactivado y Comunicación se encontró en modalidad de recepción de información. Segundo escenario: el módulo de Despliegue de Antena se encontró desactivado, Carga Útil se encontró desactivado y Comunicación se encontró en la modalidad de recibir y transmitir información. Tercer escenario: el módulo de Despliegue de Antena se encontró desactivado, se activó la Carga Útil y Comunicación se encontró en la modalidad de recibir y transmitir información.

Previo al desarrollo de un *CubeSat 2U* fue necesario realizar simulaciones del presupuesto de potencia utilizando los componentes y los módulos del Quetzal-1, contando con la integración del módulo Deorbit System. Se realizó un análisis estadístico de los resultados de la información transmitida del Quetzal-1 en cuanto al consumo de potencia y el estado de carga de las baterías. Finalmente se plantearon escenarios para un nuevo presupuesto de potencia para el *CubeSat 2U*, utilizando los parámetros descriptivos de potencia obtenidos de los resultados del Quetzal-1.

#### 2.1. Objetivos generales

1. Presentar un presupuesto de consumo de potencia para un *CubeSat* 2U, que cumpla los requerimientos del satélite.

#### 2.2. Objetivos específicos

1. Analizar la potencia real obtenida por parte del Quetzal-1 para establecer el rango de valores en los que se encuentra.
2. Analizar el efecto de ciclos térmicos del satélite en órbita para determinar si existen variaciones significativas entre el estado de carga de las baterías.
3. Validar el presupuesto de potencia realizado para el Quetzal-1 en función al consumo real de los componentes del *CubeSat*.
4. Realizar el presupuesto de potencia para un *CubeSat* 2U considerando la integración del sistema de desorbitación.



La Universidad del Valle de Guatemala (UVG) inauguró el campo de la ciencia espacial en Guatemala, con el desarrollo del *CubeSat*, Quetzal -1. El proyecto contribuyó a nuevas oportunidades para ser exploradas al permitir el desarrollo de un *CubeSat*. El proyecto Quetzal-1 contó con un equipo de personas, entre ellos profesores, estudiantes de UVG y profesionales laborales externos. Durante el desarrollo surgió la necesidad de analizar el *CubeSat* con diferentes pruebas y la solución fue analizarlo con recursos proporcionados por UVG y otros de empresas externas. Entonces, el nuevo proyecto *CubeSat* también cuenta con estos recursos.

Los problemas a resolver del trabajo presentado en este documento son: Presentar un presupuesto de consumo de potencia para un *CubeSat 2U*. El *CubeSat* cuenta con diferentes módulos que ayudan a cumplir la misión y cada uno cuenta con los requerimientos específicos de energía eléctrica, por lo tanto el correcto análisis de los resultados del Quetzal-1 permiten la generación de nueva información para la simulación de nuevo presupuesto de potencia implementando un *Deorbit System*.

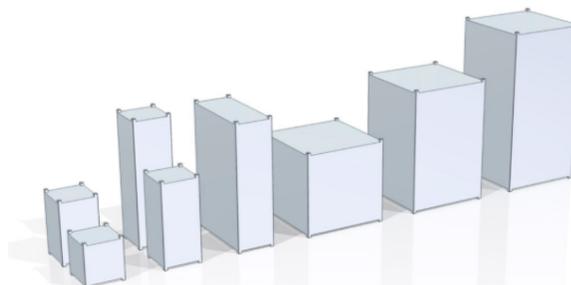
Para presentar el presupuesto de consumo de potencia es importante abordar el problema con la perspectiva de la ciencia de datos para analizar los datos de potencia obtenidos del Quetzal-1, además si los datos de carga de las baterías se ven afectados por los ciclos térmicos y validar el presupuesto de consumo de potencia realizado para el Quetzal-1. Los resultados del presupuesto de potencia proporcionan una nueva perspectiva respecto al consumo del *CubeSat* que permite analizar el comportamiento del *CubeSat* en diferentes escenarios.



#### 4.1. Nano satélites *CubeSat*

El desarrollo de nano satélites *CubeSat* permite realizar investigaciones a nivel científico en el espacio, probando la tecnología en órbita baja. El desarrollo de satélites está compuesto de diferentes componentes, se clasifican en: Estructura, térmico, determinación y control de actitud, comunicaciones, potencia y carga útil (García, 2017). Estos con la finalidad de realizar los objetivos de la misión. Cabe mencionar que las medidas de los CubeSat son de 10x10x10cm y se le denomina como 1U, se puede escalar a 2U (20x10x10cm), y 3U (30x10x10cm), junto con su límite de masa: 1U está restringido a 1.33Kg y 3U a 4Kg (Kulu, s.f.).

Figura 1: Dimensiones de *CubeSat* desde 1U hasta 16U



(Kulu, s.f.)



## 4.2. Proyecto Quetzal-1

### 4.2.1. Recolección y almacenamiento de energía

El área de recolección y almacenamiento de energía es parte del módulo de potencia eléctrica. En satélites tipo *CubeSat*, el sistema de generación de potencia está compuesto por el sistema de recolección de energía, sistema de almacenamiento de energía y sistema de distribución de energía (Garcia, 2017).

El sistema de recolección de energía más utilizado en un *CubeSat* es mediante paneles solares, que capturan la energía solar y convierten parte de ella en energía eléctrica. Para el proyecto Quetzal-1, se utilizaron las celdas solares *AZUR SPACE Solar Power GMBH* (Lee, 2019). La generación primaria de potencia es por el arreglo de los paneles solares. Cabe mencionar, que la corriente y el voltaje tienen una dependencia no lineal entre ellas. Por lo tanto, existe solo un punto donde se puede generar la potencia máxima (Bester y col., 2012). Además, aspectos como la eficiencia, el área disponible, de las celdas solares dependen mucho del diseño del *CubeSat*.

Con respecto al sistema de almacenamiento de energía del Quetzal-1, está compuesto por un arreglo de dos baterías conectadas en paralelo. Las baterías existentes son de diversas tecnologías, como por ejemplo, pueden ser de: Hidruro de níquel, Litio-ion, litio polímero, Litio-Fosfato, Níquel-Cadmio, etc. (Duann y col., 2020). La mayoría de los *CubeSat* han utilizado baterías basadas en tecnología de Litio debido a que poseen buena capacidad específica y densidad de energía a un costo accesible (Sanchez-Sanjuan y col., 2016). Para el proyecto del Quetzal-1, se utilizaron dos baterías tipo *Polymer Li-ion Rechargeable Battery model: DTP605068* (LIMITED, 2018).

Dadas las restricciones de masa de los *CubeSat*, limita la tecnología de las baterías que se pueden utilizar. Baterías con mejores propiedades cuentan con mayor energía por unidad de masa. Por lo tanto, para el Quetzal-1 no fue necesario una mayor capacidad específica. La distribución de la energía se lleva a cabo mediante un circuito eléctrico, este circuito presenta dos fuentes de energía, siendo la principal los paneles solares y la secundaria sería la batería. La batería es la que suministra la energía cuando no esté disponible la energía solar (Garcia, 2017).

La capacidad requerida del satélite depende también de la órbita en la cual se mueve el satélite. La trayectoria del satélite tiene dos estados. El primero es el estado de radiación o el periodo de luz solar disponible, donde el satélite tiene la posibilidad de generar energía y el segundo estado es donde no existe luz solar disponible, porque lo cubre la sombra de la tierra (Sanchez-Sanjuan y col., 2016). Ya que limita la cantidad de tiempo del requerimiento de energía de los módulos, también la energía disponible del sol. Estos aspectos se deben de tomar en cuenta para resolver la capacidad de energía de las baterías.

Para validar los análisis y selección de los componentes de este sistema, es necesario realizar pruebas para verificar el correcto funcionamiento de dichos componentes. Estas pruebas deben verificar el funcionamiento de las baterías y los paneles solares.

Las pruebas de los paneles solares consisten en revelar los daños a las celdas solares. A este procedimiento se le conoce como electroluminiscencia. Otras pruebas son de termografía en

la que los paneles se exponen a una fuente de calor y se debe de evaluar como es la disipación del calor. También es importante probar el desempeño de las celdas por medio de fuente de luz solar y medir la potencia eléctrica generada por los paneles (Lee, 2019).

Las Baterías requieren de pruebas como medir el voltaje, corriente y potencia. Además se deben de verificar el funcionamiento en el momento en que alcancen los límites superiores e inferiores de la carga. Es importante medir la masa antes y después de pruebas al vacío para medir cuanta de esta masa a se pierde (Garcia, 2017).

#### 4.2.2. Presupuesto de potencia

El presupuesto de consumo de potencia es un procedimiento que se realizó para determinar los requerimientos de potencia, escoger el hardware apropiado, y establecer parámetros de modulación para transmisión y recepción de la señal (Popescu, 2017). El presupuesto de consumo de potencia es un proceso iterativo que requiere de simulaciones para presentar los consumos de potencia de los módulos.

En el proyecto Quetzal-1, se realizó un presupuesto de consumo de potencia para asegurar que la capacidad de las baterías cumpliera con los requerimientos de la misión. Este se realizó para determinar el consumo del Quetzal-1 a lo largo de una órbita (Velásquez y col., 2019). Se consideraron tres escenarios, los cuales consideran posibles situaciones dependiendo del estado de la carga de las baterías y también dependiendo de la fase de la misión.

#### 4.2.3. Módulos de consumo de potencia

Para el proyecto Quetzal-1 se utilizaron diferentes módulos que consumen energía, que tienen la finalidad de completar la misión. Los módulos contaron con una tabulación de datos como la corriente y la potencia, ya sean las máximas o las nominales. El voltaje promedio y la cantidad de componentes del módulo respectivo. Cabe mencionar que los parámetros de consumo utilizados fueron determinados de forma teórica en las instalaciones de la Universidad del Valle de Guatemala. Los parámetros y funciones relevantes fueron los siguientes:

##### Módulo del sistema de determinación y control de actitud

Este módulo también se identifica por sus siglas en Inglés como *ADCS*. El *ADCS* fue módulo encargado de estabilizar el Quetzal-1. El Quetzal-1 utilizó un *ADCS* pasivo, que consiste en un imán y dos varillas de histéresis que su función fue alinear el satélite al campo magnético de la tierra (Lee, 2019).

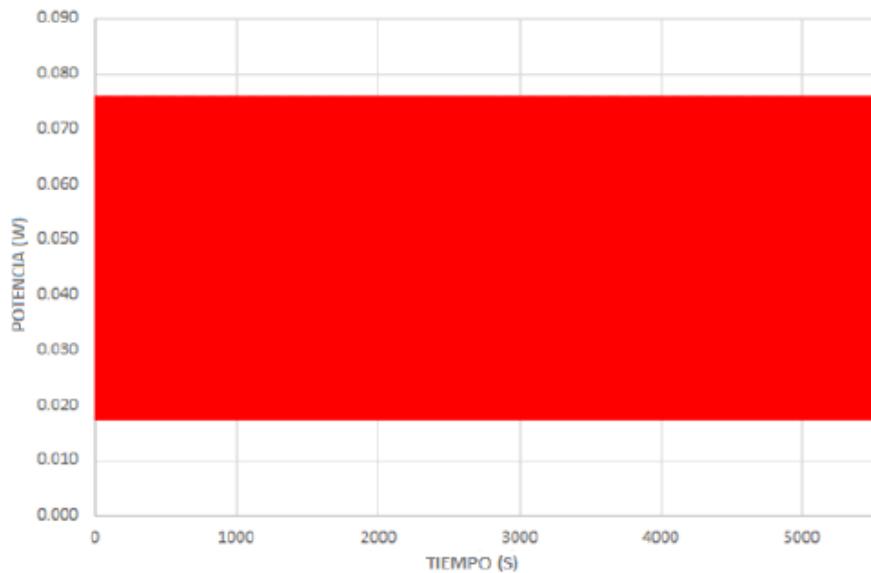
Cuadro 1: Estimación de consumo de potencia de *ADCS*

Componente	Cantidad	Voltaje (V)	Pico		Reposo	
			Corriente (mA)	Potencia (W)	Corriente (mA)	Potencia (W)
Subsistema <i>ADCS</i>	1	3.3	19.506	0.076	4.454	0.0175

(Velásquez y col., 2019)

Este subsistema recopiló datos de los componentes cada 10 segundos y para realizar las mediciones necesitó de 2 segundos. Durante esta medición, el módulo necesitó del consumo máximo de 0.076 W. Después de la medición los componentes del módulo permanecieron consumiendo la carga nominal equivalente a 0.0175 W (Velásquez y col., 2019).

Figura 2: Simulación del consumo de *ADCS* durante una órbita



(Velásquez y col., 2019)

### Módulo de carga útil

También conocido en inglés como *Payload*, este módulo fue encargado de llevar sensores multispectrales para cumplir la misión. Este módulo cuenta con la cámara monocromática *ArduCam*, carrusel de filtros y un motor piezoeléctrico *Tekceleo WLG-30*. Para mover el carrusel el motor rotó a una velocidad aproximada de 300 rpm. Que equivalen a 200 ms para rotar el carrusel 90 grados. La cámara utilizó un tiempo pico de 400 ms. En el presupuesto de potencia analizado, se tomaron cuatro fotografías por ciclo. Dejando un tiempo entre cada foto tomada para poder recopilar y almacenar los datos (Lee, 2019).

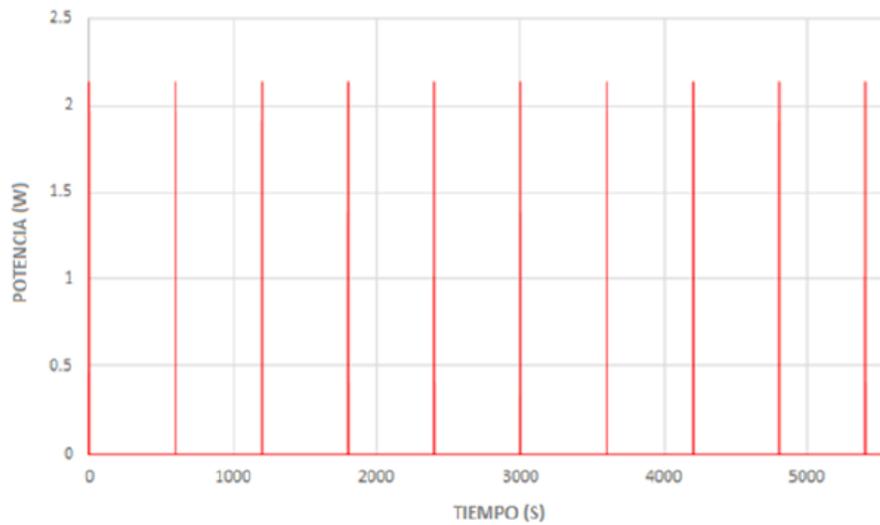
Cuadro 2: Estimación de consumo de potencia del *Payload*

Componente	Cantidad	Voltaje (V)	Pico		Reposo	
			Corriente (mA)	Potencia (W)	Corriente (mA)	Potencia (W)
<i>ArduCam</i>	1	3.3	249.2	1.246	60.4	0.302
<i>Tekceleo WLG-30</i>	1	7.5	760.30	5.780	98.3	0.747

(Velásquez y col., 2019)

Cabe mencionar que para el presupuesto de potencia se estableció la condición, que para activar el módulo *Payload*, el estado de carga del Quetzal-1 debe ser superior al 65 %. De tal forma, si la condición se cumple el módulo repetiría el ciclo, de tomar una fotografía, cada 600 segundos. Como resultado se comprobó que el Quetzal-1 contaba con la capacidad de trabajar con uso continuo del *Payload* (Velásquez y col., 2019).

Figura 3: Simulación de consumo de potencia *Payload* durante una órbita



(Velásquez y col., 2019)

### Módulo del mecanismo de despliegue de antena

Este módulo es también conocido como *ADM*. Es un mecanismo que tuvo la función de activar 4 resistencias, *Vishay Pro1*, que desplazarían las antenas. Las 4 antenas son las que permiten la correcta operación del módulo de comunicaciones (Lee, 2019).

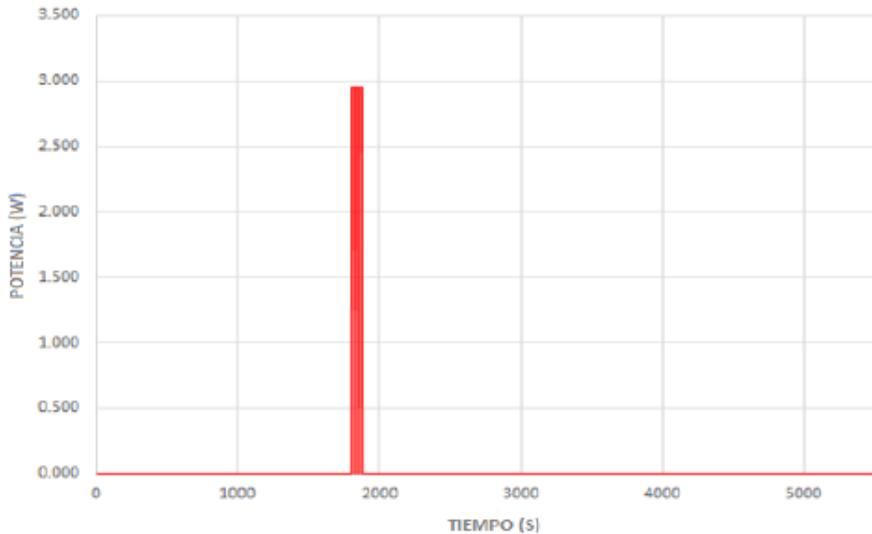
Cuadro 3: Estimación de consumo de potencia del *ADM*

Componente	Cantidad	Voltaje (V)	Pico		Reposo	
			Corriente (mA)	Potencia (W)	Corriente (mA)	Potencia (W)
RESISTENCIA <i>Vishay Pro1</i>	4	3.7	800	2.96	0.00	0.00

(Velásquez y col., 2019)

El mecanismo de despliegue de las antenas consistió en activar las 4 resistencias con la potencia máxima de 2.96 Watts durante 25 segundos, con un tiempo de espera de 5 segundos, para activar la siguiente resistencia al momento que la anterior dejara de consumir corriente. Al terminar el ciclo, el módulo fue programado para contar con un período de espera de 30 minutos. Ya que, en el caso que no se desplegara alguna de las antenas, se reiniciaría la secuencia de despliegue de las antenas. (Velásquez y col., 2019)

Figura 4: Simulación del consumo del módulo *ADM* durante una órbita



(Velásquez y col., 2019)

### Módulo de comunicación

Este módulo también es conocido como *COMMS*, tuvo la función de recibir y transmitir datos a la estación terrestre. En el módulo se decodificaron los datos y se obtuvo información del estado del Quetzal-1 (García, 2017).

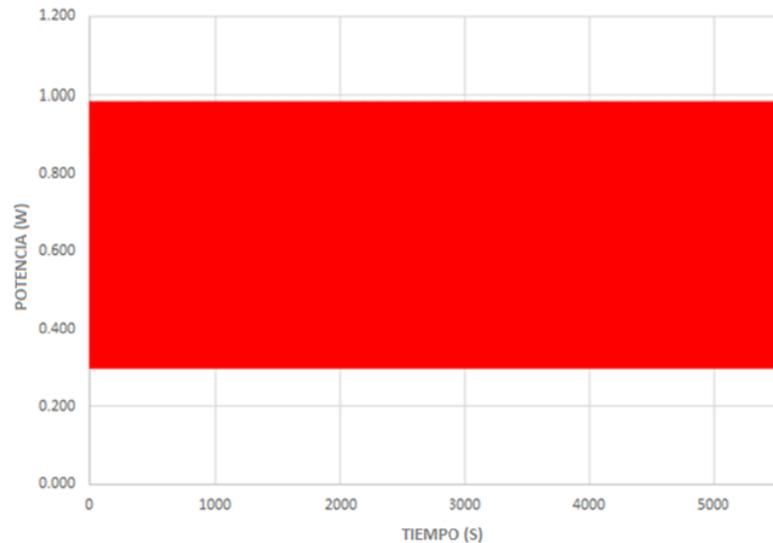
Cuadro 4: Estimación de consumo de potencia del *COMMS*

Componente	Cantidad	Voltaje (V)	Pico		Reposo	
			Corriente (mA)	Potencia (W)	Corriente (mA)	Potencia (W)
<i>NanoCom AX100</i>	1	3.3	998	3.293	90	0.297

(Velásquez y col., 2019)

El módulo de comunicaciones cuenta con dos estados de operación el estado de recepción de información (*RX*) y el estado de transferir información (*TX-RX*) a la estación de la tierra. Para su operación se contó con el componente *NanoCom AX100* que transmitió una baliza. Una baliza es una transmisión desde la estación espacial que contiene información del satélite. La baliza se transmite por un tiempo de 300 ms, y repitió el procedimiento cada 10 segundos. Para la transmisión de la baliza se utilizó un consumo de potencia de 3.293 W. De tal forma que, el consumo de corriente es nominal cuando no se encontraba transmitiendo información (Velásquez y col., 2019).

Figura 5: Simulación de consumo de potencia del *COMMS* durante una órbita



(Velásquez y col., 2019)

### Módulo de computadora a bordo

Este módulo también es identificado por sus siglas en Inglés *OBC*. Fue el módulo encargado de operar el Quetzal-1 mediante el envío de información por medio del componente *NanoMind*. Los comandos fueron procesados de tal forma que se distribuyeron a los otros sistemas del Quetzal-1, que afectaron en como el Quetzal-1 consume energía (García, 2017). El módulo *OBC* consumió energía en modo nominal y el consumo pico de 0.552 W que fue

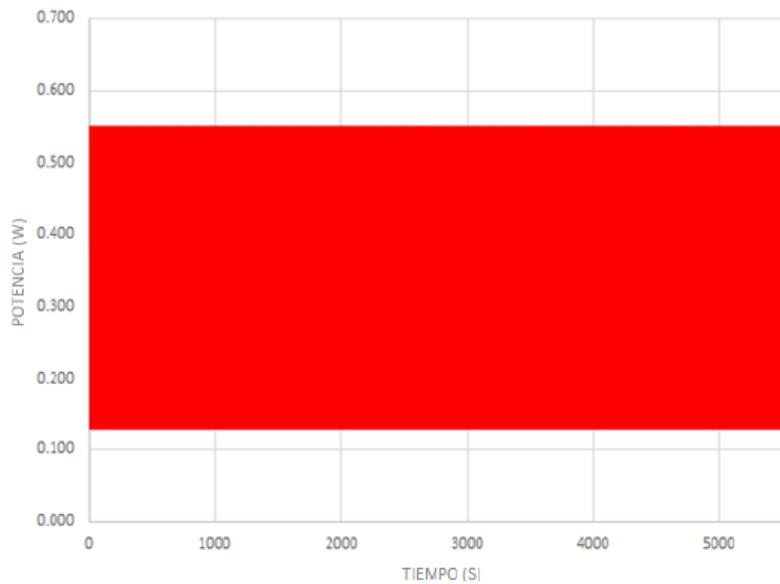
cuando transmitió una baliza cada 10 segundos (Velásquez y col., 2019).

Cuadro 5: Estimación de consumo de potencia del *OBC*

Componente	Cantidad	Voltaje (V)	Pico		Reposo	
			Corriente (mA)	Potencia (W)	Corriente (mA)	Potencia (W)
NanoMind A3200	1	3.3	38.7	0.552	38.7	0.128

(Velásquez y col., 2019)

Figura 6: Simulación de consumo de potencia de *OBC* durante una órbita



(Velásquez y col., 2019)

### Módulo de sistema de potencia eléctrica

Este módulo también es conocido como *EPS*. Su función fue de suministrar, almacenar, distribuir y controlar la energía eléctrica del Quetzal-1. Este subsistema funcionó de forma constante y permaneció en operación en sus parámetros picos de consumo.

Cuadro 6: Estimación de consumo de potencia del *EPS*

Componente	Cantidad	Voltaje (V)	Pico	
			Corriente (mA)	Potencia (W)
Subsistema <i>EPS</i>	1	3.3	6.11	0.02

(Velásquez y col., 2019)

#### 4.2.4. Análisis del primer escenario del presupuesto de potencia

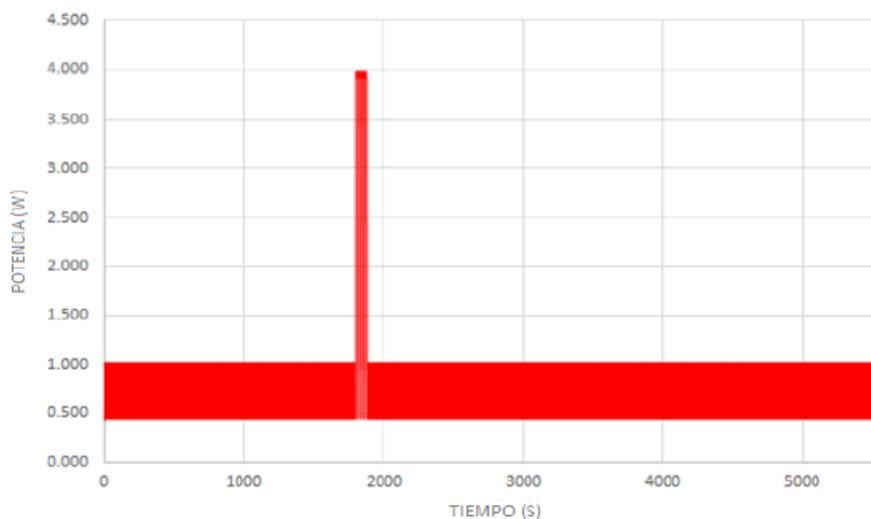
Este escenario fue en el cual el mecanismo de despliegue de antenas se activó 30 minutos después del lanzamiento del Quetzal-1 al espacio. Para el análisis solo se consideró un ciclo de despliegue de las antenas. Además el módulo *COMMS* estuvo en modo de recepción de información (*RX*). El módulo de carga útil estuvo desactivado durante este escenario. Se representó el consumo de potencia, bajo las siguientes condiciones que afectaron el estado de carga de las baterías durante una órbita en el primer escenario.

Cuadro 7: Subsistemas activos durante primer escenario

<i>ADCS</i>	<i>Payload</i>	<i>ADM</i>	<i>EPS</i>	<i>COMMS</i>	<i>OBC</i>
<i>ON</i>	<i>OFF</i>	<i>ON</i>	<i>ON</i>	<i>RX</i>	<i>ON</i>

(Velásquez y col., 2019)

Figura 7: Simulación del consumo de potencia del Quetzal-1 durante el primer escenario

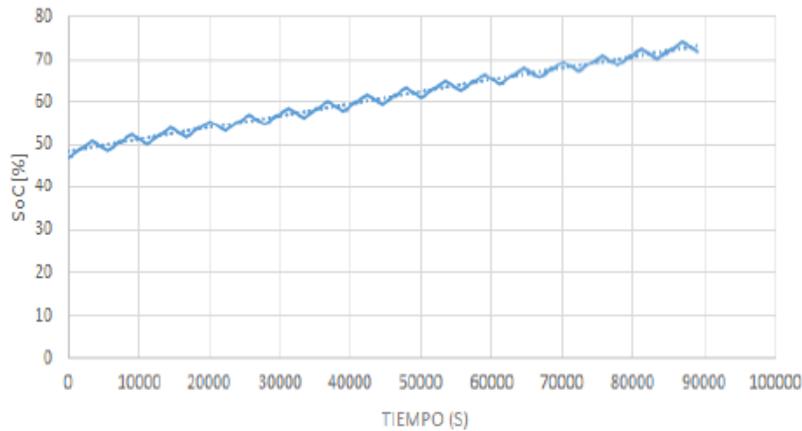


(Velásquez y col., 2019)

Ya que el voltaje de las baterías es de 3.7 V y están conectadas en paralelo se esperaba que el voltaje promedio durante la órbita fuera de 3.7 V. En esa órbita el consumo total de

energía estimado fue de 249.5 mAh. Recordando que este análisis se realizó bajo el supuesto que el estado inicial de la carga de las baterías, después del lanzamiento, es de 47%. Además se obtuvo una generación energía de 311.7 mAh. El resultado es que la tasa a la que se cargan las baterías es de aproximadamente 1.56%. Este resultado fue después de una simulación de un día, que equivale a 16 órbitas. Además se obtuvo como resultado un cambio positivo del 25.07%.

Figura 8: Simulación del estado de carga de las baterías durante primer escenario



(Velásquez y col., 2019)

#### 4.2.5. Análisis del segundo escenario del presupuesto de potencia

En este escenario se analizó el consumo de energía en una órbita después del lanzamiento y estabilización del satélite. Se utilizó la condición de que el estado de carga después del lanzamiento es de un 47%. El módulo de carga útil no se encontraba activo durante esta simulación, las antenas se desplegaron y el módulo *ADCS* se encontró activo también. El módulo *COMMS* se encontró en la función de transmisión de señal (*TX-RX*).

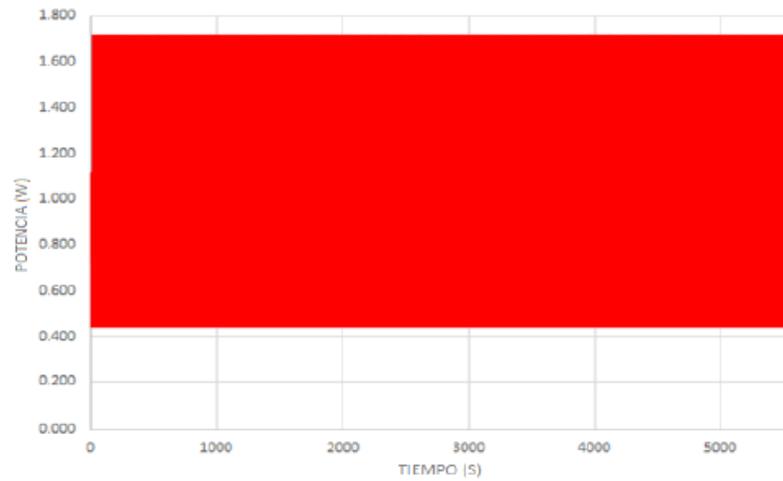
Cuadro 8: Subsistemas activos durante segundo escenario

<i>ADCS</i>	<i>Payload</i>	<i>ADM</i>	<i>EPS</i>	<i>COMMS</i>	<i>OBC</i>
<i>ON</i>	<i>OFF</i>	<i>OFF</i>	<i>ON</i>	<i>TX-RX</i>	<i>ON</i>

(Velásquez y col., 2019)



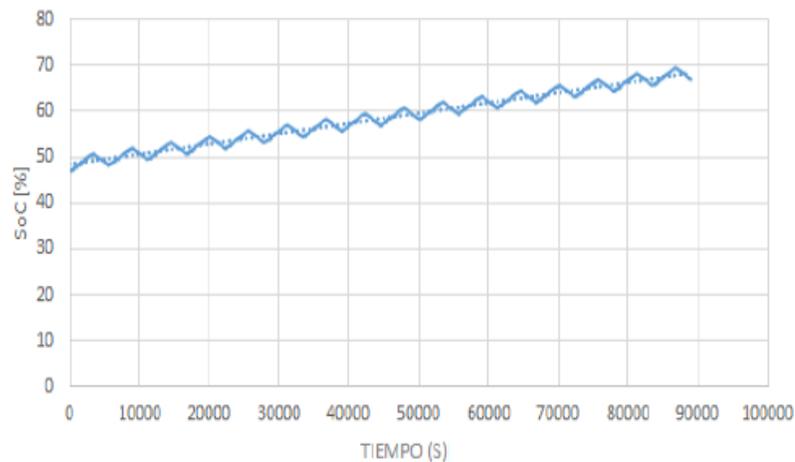
Figura 9: Simulación del consumo de potencia del Quetzal-1 durante el segundo escenario



(Velásquez y col., 2019)

Al ser la conexión de las baterías en paralelo se esperaba que el voltaje de descarga y carga fuera de 3.7V. Como la suposición del estado de carga al inicio de la misión, equivale a 1880 mAh. Al final de la simulación de una órbita, se obtuvo un consumo total de energía de 262.3mAh y una generación de energía promedio de 311.7 mAh. Esto da como resultado una tasa de carga positiva en cada órbita. Después de un día se calculó un cambio positivo de aproximadamente 19 %.

Figura 10: Simulación del estado de carga de las baterías durante segundo escenario



(Velásquez y col., 2019)



#### 4.2.6. Análisis del tercer escenario de presupuesto de potencia

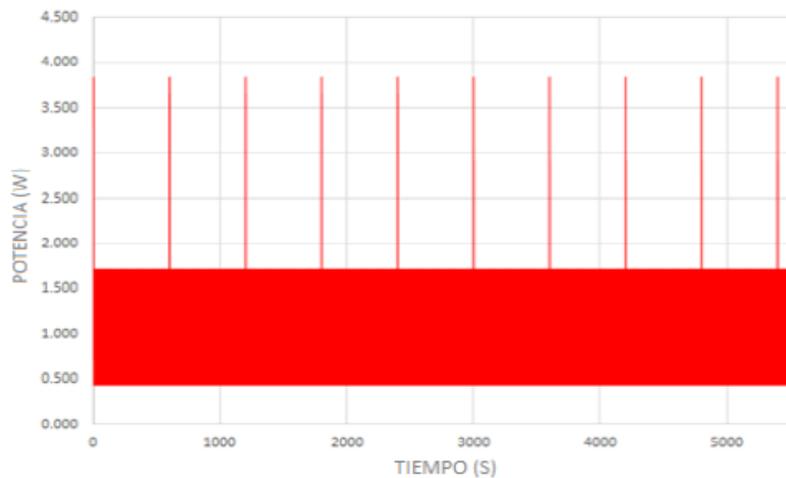
Para el tercer escenario se activó el módulo de la carga útil una vez cada 10 segundos, en el caso que la carga de las baterías fuera superior al 65 %. En este caso el módulo de *ADM* se encontró apagado, el módulo *COMMS* se encontró en modo de transmisión.

Cuadro 9: Subsistemas activos durante tercer escenario

<i>ADCS</i>	<i>Payload</i>	<i>ADM</i>	<i>EPS</i>	<i>COMMS</i>	<i>OBC</i>
<i>ON</i>	<i>ON</i>	<i>OFF</i>	<i>ON</i>	<i>TX-RX</i>	<i>ON</i>

(Velásquez y col., 2019)

Figura 11: Simulación del consumo de potencia de Quetzal-1 durante el tercer escenario



(Velásquez y col., 2019)

En este escenario se trabajó bajo la suposición que el estado de carga inicial después del lanzamiento sería del 60 %, con la finalidad de observar la variación del consumo cuando se activó la carga útil. Se obtuvo que, después de una órbita el consumo de energía sería de 263.4 mAh y una generación de 311 mAh. De esta manera se obtuvo que, incluso con el uso continuo de la carga útil, se genera la suficiente energía para cargar las baterías.

#### 4.3. Análisis de resultados del Quetzal-1

Los parámetros que representaron información relevante sobre el estado del Quetzal-1 se recopilaron y almacenaron, en hojas de cálculo. Se tomaron en cuenta 26 semanas de información siendo una hoja de cálculo por semana, las cuales abarcan desde el 28 de abril hasta el 27 de octubre del año 2020. Las hojas de cálculo contaron con un total de 92 variables. Para la realización de este trabajo de investigación se utilizaron un total de 25 variables para los análisis realizados. Las hojas de cálculo se utilizaron para el análisis de potencia real, análisis de efectos de temperatura y para la validación del presupuesto de potencia del Quetzal-1.

Previo a los análisis de datos se revisaron los tipos de variables en el lenguaje de *Python*, esto se realizó previo a cualquier uso de las variables ya que de no contar con el tipo de variable correspondiente pueden ocasionar errores en la implementación de código.

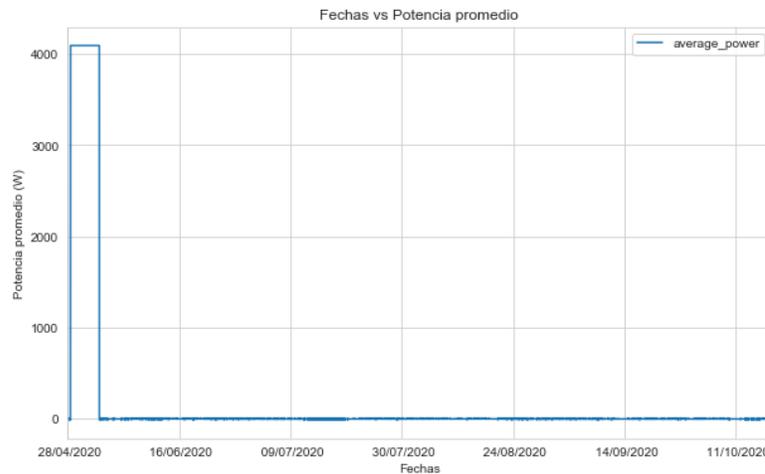
### 4.3.1. Limpieza de datos

Se realizó una búsqueda por los datos en el caso que existieran celdas vacías sin información. Los lenguajes de programación le asignan el parámetro N/A para llenar las celdas sin información. En el análisis de los datos no se encontraron celdas sin información.

Se procedió a identificar los datos que no brindan información de valor a los análisis, estos datos son datos atípicos, que representan errores en el registro de la información debido a un problema con el Quetzal-1. El primer problema que se puede observar en los datos son valores de 4,093 W que aparecen en las dos primeras semanas de datos y se debe a un fallo de comunicación entre *EPS* y *OBC*. En este caso el programa asignó estos valores para mostrar que no son valores reales.

Para analizar las hojas de cálculo de Excel se utilizó el lenguaje de programación *Python* y el siguiente paso fue vincular todos los datos en una sola hoja de cálculo, para ello se eligió la librería de *Python* denominada Pandas. Esta biblioteca de código es una de las herramientas más importantes para el análisis de datos de *Python*. Los datos con los valores de falla de comunicación se muestran en la Figura 12.

Figura 12: Gráfica de datos inicial.



Se realizó la limpieza de estos valores de 4,093 W. En la Figura 12 se grafican los datos sin estos valores, pero como se muestra en la figura hay negativos donde el satélite registra un valor de -8.5 W, estos valores corresponden a una falla con *EPS*, cuando esto sucede, las banderas de transmisión del *EPS*, que son sensores de energía eléctrica, registran un valor de cero que corresponde a datos corruptos. De la misma manera la temperatura del *EPS* muestra valores de -25 ° C, estos datos se pueden ver en la Figura 14.

Figura 13: Primera limpieza de datos

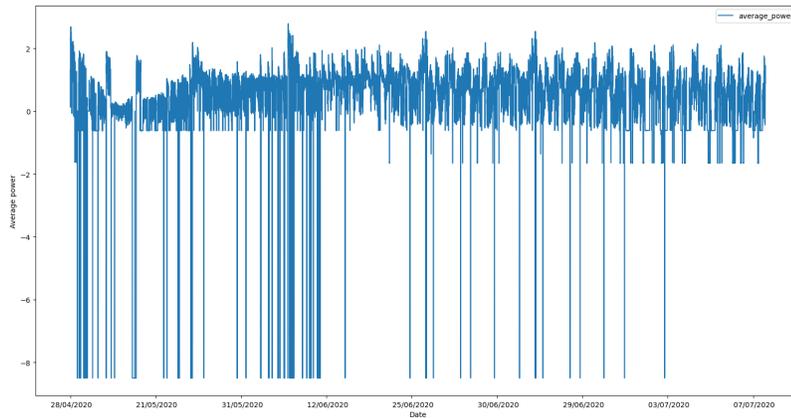


Figura 14: Datos de falla con subsistema *EPS*

	Date	average_power	battery_voltage	EPS_temperature	transmission_flags_EPS
163	29/04/2020	-8.5	2.5	-25.0	0
165	29/04/2020	-8.5	2.5	-25.0	0
172	29/04/2020	-8.5	2.5	-25.0	0
214	29/04/2020	-8.5	2.5	-25.0	0
217	29/04/2020	-8.5	2.5	-25.0	0
...	...	...	...	...	...
14308	28/06/2020	-8.5	2.5	-25.0	0
14544	29/06/2020	-8.5	2.5	-25.0	0
15102	30/06/2020	-8.5	2.5	-25.0	0
15582	01/07/2020	-8.5	2.5	-25.0	0
16524	03/07/2020	-8.5	2.5	-25.0	0

El último caso es cuando hay una falla con el monitor de las baterías. Cuando esto sucede, la forma de identificarlo es que el voltaje de las baterías tiene valores de 3.6 V. En la Figura 15 se puede ver este caso. Cabe mencionar que cuando el voltaje de las baterías es 3.6 V los valores de potencia son 0 Watts pero no son los únicos que aparecen en los datos por lo que son valores que no aportan al análisis.

Figura 15: Falla con el monitor de las baterías.

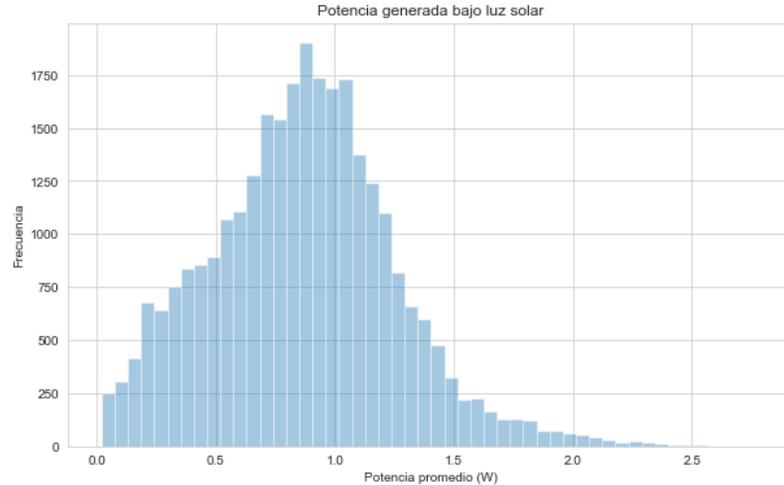
	Date	average_power	battery_voltage	EPS_temperature	transmission_flags_EPS
79	29/04/2020	0.0	3.6	3.65	31
162	29/04/2020	0.0	3.6	12.32	31
169	29/04/2020	0.0	3.6	11.19	31
180	29/04/2020	0.0	3.6	10.44	31
186	29/04/2020	0.0	3.6	9.68	31
...	...	...	...	...	...
17012	04/07/2020	0.0	3.6	2.14	31
17572	05/07/2020	0.0	3.6	2.90	31
18127	06/07/2020	0.0	3.6	2.90	31
18473	07/07/2020	0.0	3.6	7.80	31

### 4.3.2. Análisis de Potencia Real

En las hojas de cálculo, los datos de potencia se encuentran como la variable *average\_power* y los valores están en Watts. En los datos los valores de potencia se dividen en valores positivos y negativos, cuando los datos de potencia son positivos es cuando el Quetzal-1 estuvo generando energía y cuando son negativos el Quetzal-1 estuvo consumiendo energía. Estos resultados se obtienen mediante el cálculo de la potencia generada por los paneles solares menos el consumo total de los módulos del Quetzal-1.

Por lo que se debe de analizar las distribuciones de los datos de potencia, se debe realizar una separación entre los datos de consumo y generación de energía. Es importante mencionar que la potencia esta descrita también por voltaje que está registrado como *battery\_voltage* y la corriente eléctrica que está representada por la variable *average\_current*.

Figura 16: Potencia generada bajo luz solar.



Los resultados de potencia se pueden separar en base a cuando el Quetzal-1 se encontró bajo efectos de luz solar o eclipse. Los registros de los datos de potencia bajo luz solar están descritos por una distribución normal con un sesgo a la derecha. Dado que la distribución es normal, parámetros como la media de estadística descriptiva, representa un valor que describe el consumo de potencia de la distribución.

En los registros de los datos hay variables que corresponden al voltaje de los fotodiodos. Los fotodiodos son semiconductores sensibles a fluctuaciones de temperatura y flujos de radiación (Chuan Tan y col., 2013). Las variables se pueden encontrar en las hojas de cálculo como *ADC1\_CH 1-6* y *ADC2\_CH 1-6*. Si estos valores de voltaje de los fotodiodos son iguales a cero significa que el Quetzal-1 está en eclipse. De lo contrario el Quetzal-1 se encuentra bajo los efectos de luz solar. Como se muestra en la Figura 17. Cabe mencionar que no existe un resultado para la generación de potencia bajo eclipse debido a la falta de la luz del sol, por lo que no es posible generar energía bajo el eclipse.

Figura 17: Datos de los fotodiodos.

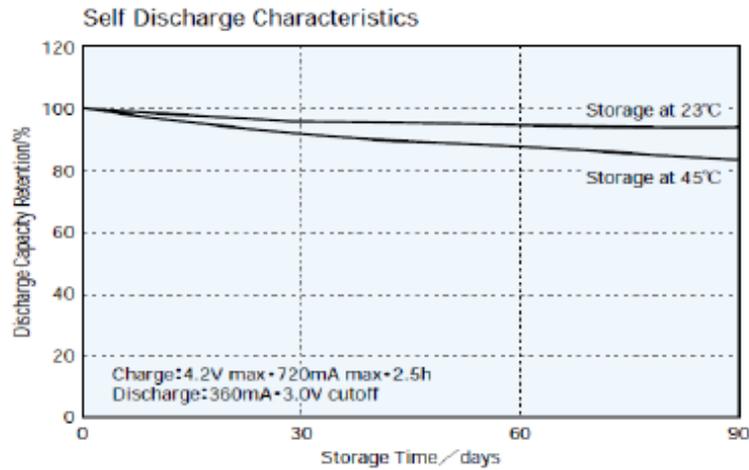
average_power	ADC1_CH1	ADC1_CH2	ADC1_CH3	ADC1_CH4	ADC1_CH5	ADC1_CH6	ADC2_CH1	ADC2_CH2	ADC2_CH3	ADC2_CH4	ADC2_CH5	ADC2_CH6
-1.59115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.62439	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.59115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.62439	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.62439	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
-0.61901	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.61901	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.61901	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.61901	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.61901	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

### 4.3.3. Análisis de efectos de temperatura.

Cuando se deja descargar una batería es afectada por la temperatura de almacenamiento y la profundidad de carga. La batería muestra una disminución de la capacidad bajo los factores de cuanto más cerca este del estado de la carga completa al momento del almacenamiento y cuanto mayor sea la temperatura ambiente.

Durante el almacenamiento de una batería recargable de iones de litio completamente cargada, normalmente pierde aproximadamente el 10 % de su capacidad en tres meses. Por estas razones, las baterías recargables de iones de litio deben tener un estado de carga aproximada del 60 % cuando se almacenan durante largos períodos de tiempo, y lo indicado es que se almacenen en un ambiente de baja temperatura (Cheng y col., 2017).

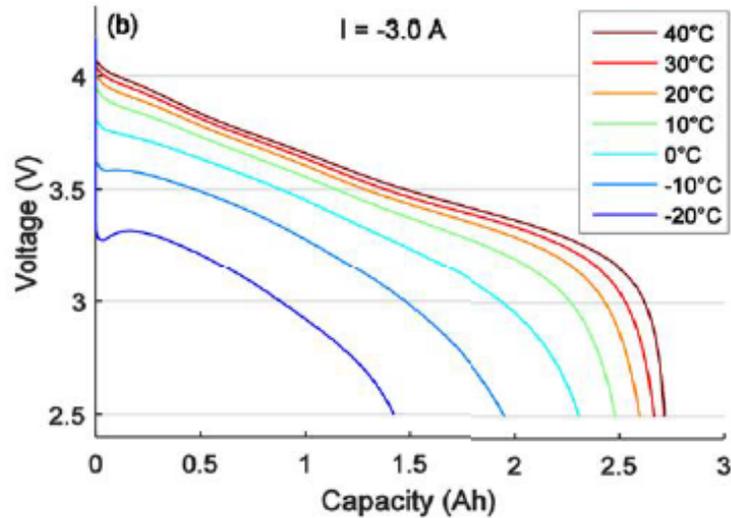
Figura 18: Capacidad de carga durante almacenamiento prolongado, en función de la temperatura



(Cheng y col., 2017)

A medida que la temperatura de almacenamiento es mayor, la carga de las baterías disminuye a mayor velocidad a lo largo del tiempo. Las baterías deben de ser almacenadas en un lugar seco y fresco ya que la humedad relativa no debe ser mayor a un 60 % HR. Las temperaturas bajas también afectan el estado de la carga de las baterías ya que entre más cercana, o incluso por debajo, a 0 °C disminuye la capacidad de la batería.

Figura 19: Disminución momentánea de capacidad por enfriamiento de las celdas



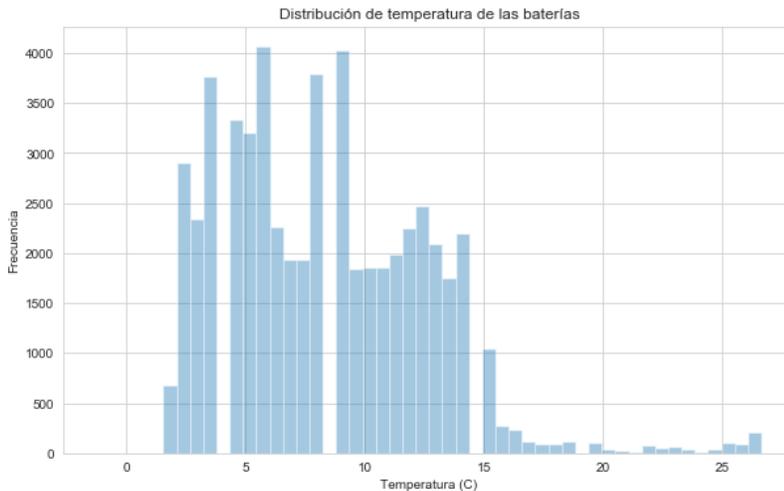
(University, s.f.)

Sin embargo, la alta resistencia interna causa un cierto efecto de calentamiento debido a la pérdida de eficiencia causada por la caída de voltaje cuando se aplica una corriente de carga. Aunque algunas baterías de litio pueden trabajar a temperaturas inferiores hasta de  $-40\text{ }^{\circ}\text{C}$ , estas solamente pueden usarse con una corriente de descarga reducida de aproximadamente  $0.2\text{ C}$  (University, s.f.).

Para realizar el análisis de los efectos de los cambios de temperatura sobre las baterías, se tomará en cuenta las variables de *EPS\_temperature* que nos indica las temperaturas a las que las baterías fueron expuestas durante la misión. De tal forma que es importante conocer las distribuciones de los datos y verificar que las lecturas registradas tengan sentido en el entorno que fueron programadas.

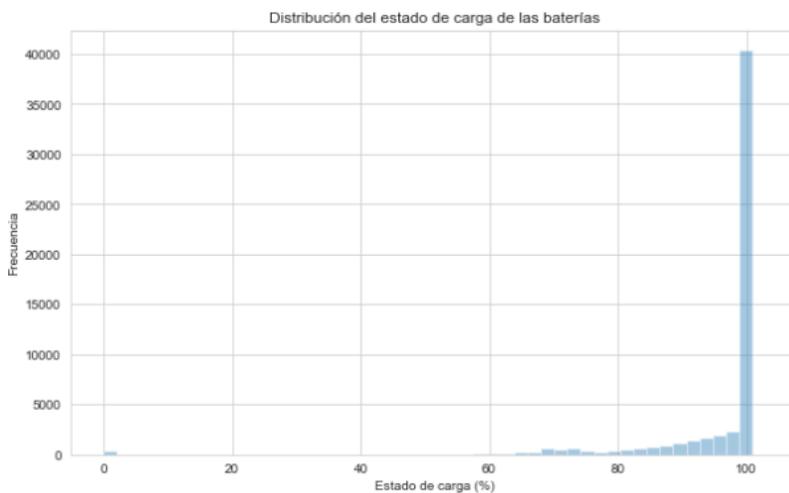
La temperatura de las baterías es controlada por un calentador, que contó con un sistema de control para mantener la temperatura de las baterías en rangos que se encuentren superiores a los  $0\text{ }^{\circ}\text{C}$  y los  $25\text{ }^{\circ}\text{C}$ . En la distribución podemos observar que los registros de la temperatura se encuentran en los parámetros de control, cabe mencionar que existen registros superiores a los  $25\text{ }^{\circ}\text{C}$ , sin embargo estos no son datos erróneos, estos existen ya que no es necesario que el sistema de control sea más eficiente. Existen registros que exceden del rango pero no de forma considerable.

Figura 20: Histograma de los registros del cambio de temperatura de las baterías.



La variable *state\_charge* indica el estado de carga de las baterías. Por lo tanto es relevante conocer la distribución del comportamiento del estado de carga de las baterías y verificar si los datos brindan información. En la siguiente figura se muestra la distribución de los datos del estado de carga en los cuales podemos observar que el rango de carga se encuentra de 0 a 100 % lo cual permite utilizar los datos para realizar análisis. Por lo tanto, se puede realizar un análisis de correlación entre las dos variables para comprobar el efecto de los cambios de temperatura sobre el estado de carga de las baterías.

Figura 21: Histograma del estado de carga de las baterías.

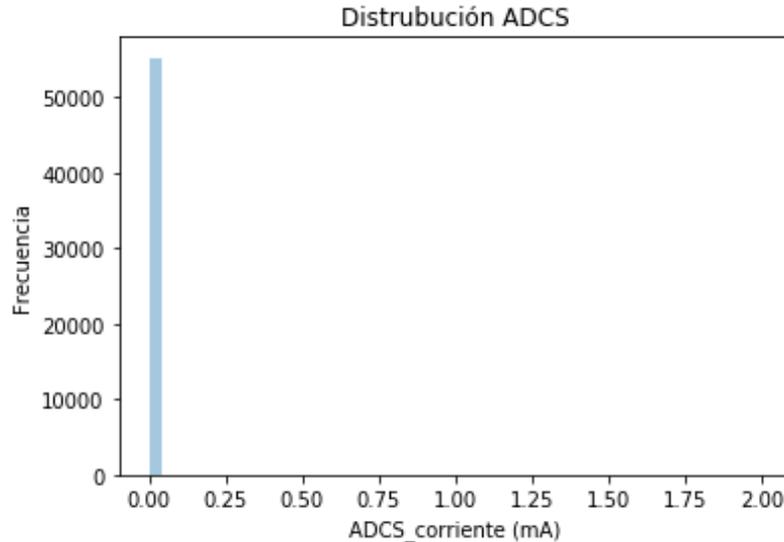


#### 4.3.4. Análisis del comportamiento de los módulos

En los resultados obtenidos por el Quetzal-1 se encuentran los consumos de corriente por parte de los módulos de *ADCS*, *COMMS*, y *Payload*. Los módulos *ADM*, *EPS*, y *OBC* no

cuentan con registros de corriente.

Figura 22: Frecuencia del consumo de corriente para el módulo *ADCS*

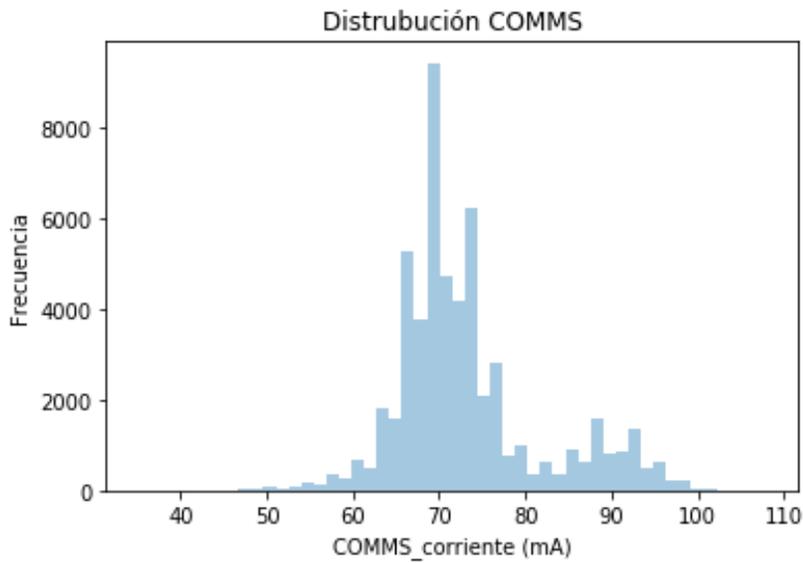


Para el módulo del *ADCS* se obtuvo que la mayor frecuencia de los datos se encuentra cuando el consumo es 0.00 mA. El módulo operó mediante de un imán y dos varillas de histéresis y fue programado para estar operando todo el tiempo. Por lo tanto, que los datos sean de 0.00 mA no se debe de considerar como que el módulo se encontraba desactivado.

En los datos de consumo de energía por parte de los módulos realizados en UVG, se puede observar que el consumo de corriente, del módulo *ADCS*, es bajo. Por lo tanto, los datos representan que el módulo consumió corriente durante la misión, sin embargo la computadora no cuenta con la capacidad para generar los decimales correspondientes al consumo real del módulo.

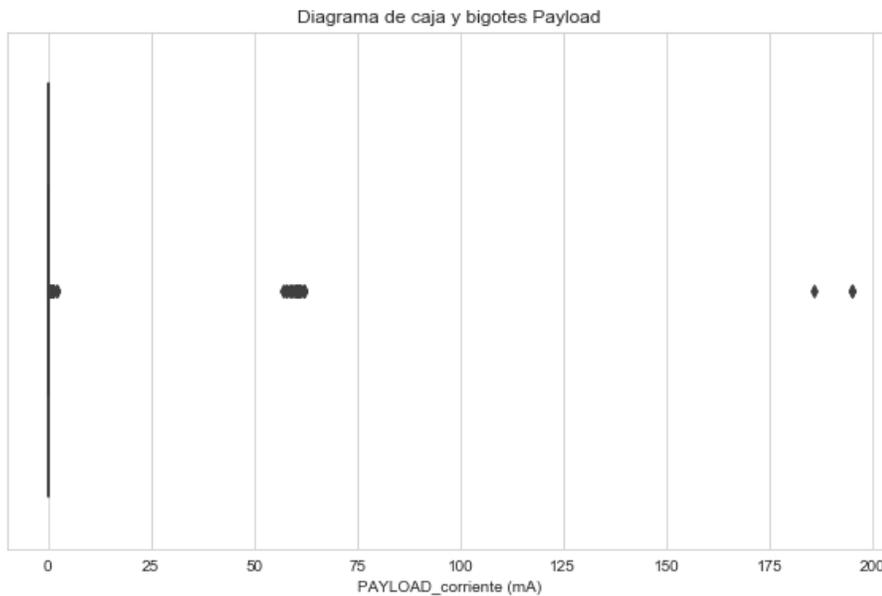
El módulo *COMMS* es uno de los módulos que siempre estuvo en operación sin embargo cuenta con dos estados de operación. El estado de recepción de información y el transmisión de información no son posibles de separar en la base de datos mediante una variable que indique de que estado es donde proviene la información. Por lo tanto, la siguiente figura el histograma muestra el comportamiento del módulo de comunicaciones durante la misión.

Figura 23: Frecuencia del consumo de corriente para el módulo *COMMS*.



El módulo *Payload* es uno de los módulos que no operan de forma continua, este se utilizó para tomar fotografías con la cámara monocromática por lo que se utilizó en momentos específicos, la mayor parte de los datos son de cuando no está consumiendo corriente.

Figura 24: Frecuencia del consumo de corriente para el módulo *Payload*



En la Figura 24 se graficó un diagrama de caja y bigotes, la línea vertical que se encuentra en cero es el diagrama, sin embargo no se muestra como debe, ya que a medida que el diagrama de caja y bigotes se cierra indica que los datos convergen en cero, debido a que el módulo se utilizó en momentos específicos. Los datos fuera del diagrama son los datos del comportamiento del módulo cuando fue utilizado. Se representan como datos atípicos por

su baja frecuencia durante la misión.

El módulo de *ADM* cuenta con la variable de *ADM\_status* que indica el estado de las 4 antenas, si las 4 antenas se desplegaron correctamente. Analizando los datos del consumo de potencia no es posible determinar los picos de consumo por parte del ADM, ya que los registros de los datos no son de forma continua, son discontinuos y no periódicos. Debido a que los datos del Quetzal-1 se obtuvieron cuando orbitó sobre alguna estación capaz de recibir esta señal, lo que explica la discontinuidad de la información. Además que el consumo del *ADM* es por 25 segundos, por lo que esta información no es posible de identificar entre los registros.

#### **4.3.5. Análisis de datos atípicos.**

Dadas las distribuciones de los datos, es importante mencionar que para cada gráfica de las distribuciones de los datos siempre se encontrarán datos atípicos en los diagramas de caja y bigote, como el de la Figura 24. Estos datos regularmente se eliminan ya que representan un sesgo en la información. Sin embargo, los datos que no son de utilidad ya fueron removidos en la sección de la limpieza de datos. Por lo tanto, los datos atípicos que se muestran en los diagramas son los registros con menor frecuencia durante la misión, estos son representados como los valores mínimos o máximos de las distribuciones. Lo que implica que los datos obtenidos de estas distribuciones representan información que se debe analizar.



La metodología de este trabajo consistió en lo siguiente: para el análisis de potencia real se localizaron y se removieron los datos que son fallas tanto en el *EPS*, de comunicación con la computadora y con el monitor de las baterías. Luego se analizaron los resultados de potencia dependiendo si el Quetzal-1 se encontraba bajo luz o eclipse solar y considerando el estado del calentador de las baterías, se obtuvieron los parámetros estadísticos y las distribuciones que describen el comportamiento de los datos. Para el análisis de los efectos de temperatura en las baterías, se obtuvo la distribución y los parámetros descriptivos de los datos de la temperatura con la finalidad de determinar los rangos máximos y mínimos alcanzados. Además, se obtuvo el coeficiente de correlación entre el estado de carga de las baterías y la temperatura.

Para la validación del presupuesto de potencia del Quetzal-1 se obtuvieron las distribuciones, parámetros nominales y pico del consumo de potencia de los módulos *ADCS*, *COMMS* y *Payload*; de tal forma que se calcularon los porcentajes de error respecto a los resultados teóricos de los módulos anteriormente mencionados. Adicionalmente, se analizaron los datos para obtener los resultados del consumo de potencia del Quetzal-1 durante los escenarios. En el caso del presupuesto de potencia de un *CubeSat 2U*, se tabuló el comportamiento del consumo de potencia de los módulos que lo componen, para lo cual se utilizó el simulador de la generación de potencia de los paneles solares realizado en UVG y se calcularon los resultados de la generación de potencia y estado de carga de las baterías.

## 5.1. Análisis del presupuesto de potencia para un *CubeSat 2U*

### 5.1.1. Análisis de potencia real del Quetzal-1

Para realizar el análisis de potencia real se separaron los datos en base a si el Quetzal-1 se encontraba bajo eclipse o bajo luz solar. Además, se analizaron otros casos relevantes involucrando el estado del calentador de las baterías. Al final se obtuvieron las distribuciones que representan el comportamiento de los datos y se tabularon sus parámetros descriptivos correspondientes. Los casos analizados fueron los siguientes:

1. Generación de potencia bajo efectos de luz solar
  - Potencia generada con el calentador de las baterías activado.
  - Potencia generada con el calentador de las baterías desactivado.
2. Consumo de potencia bajo efectos de luz solar.
  - Consumo de potencia con el calentador de las baterías activado.
  - Consumo de potencia con el calentador de las baterías desactivado.
3. Consumo de potencia bajo efectos de eclipse
  - Consumo de potencia con el calentador de las baterías activado.
  - Consumo de potencia con el calentador de las baterías desactivado.

### 5.1.2. Análisis de efectos de temperatura sobre las baterías del Quetzal-1

Se analizaron los rangos de temperatura a los que fueron expuestas las baterías del Quetzal-1. De tal forma, se obtuvo la distribución y un diagrama *boxplot* de los registros de la temperatura en grados centígrados. Se calcularon los parámetros descriptivos de la distribución de los resultados.

En este caso se realizó un análisis de correlación entre la temperatura y el estado de carga de las baterías. Se analizaron las distribuciones de los resultados de ambas variables mediante métodos estadísticos para obtener la información que describe las distribuciones. De esta forma se determinó la influencia de los cambios de temperatura sobre el almacenamiento de energía de las baterías.

### 5.1.3. Validación del presupuesto de consumo de potencia del Quetzal-1

Para la validación del presupuesto de potencia realizado para el Quetzal-1 se analizaron los resultados de consumo de potencia de los módulos del satélite, en el transcurso de las 26 semanas de recolección de datos. Para el módulo *ADCS* se determinaron el consumo nominal y pico, siendo el valor mínimo y máximo, que representan el comportamiento del módulo. De tal forma que se obtuvo el porcentaje de error utilizando los resultados del consumo nominal y pico teóricos del módulo.

Para el análisis del módulo *COMMS* se obtuvo la distribución de probabilidad del consumo de potencia, se determinó el valor esperado para el estado de *RX* y para el estado *TX-RX*. Se obtuvo el porcentaje de error utilizando los respectivos valores teóricos del módulo.

En el caso del módulo *Payload* se obtuvo una gráfica de la distribución de los resultados del consumo de potencia y se obtuvo el valor máximo. De tal forma que se obtuvo el porcentaje de error utilizando el consumo de potencia pico obtenido de forma teórica. Por lo tanto, los resultados reales, correspondientes, de los tres módulos analizados se utilizaron para el desarrollo del presupuesto de potencia de un *CubeSat 2U*.

Se realizó una investigación para determinar el comportamiento de los escenarios del consumo de energía planteados previo al lanzamiento. El primer y segundo escenario se analizó con un solo histograma ya que, de forma teórica, el primer escenario teórico representa el comportamiento del Quetzal-1 durante la secuencia de despliegue de las antenas. Como se mencionó anteriormente, no es posible determinar el consumo del módulo *ADM* ya que no se cuenta con una variable que registrara su consumo y no se puede delimitar la información para obtener de forma precisa un consumo aproximado. Por lo tanto, se discutieron los resultados comparando parámetros de estadística descriptiva.

Para el análisis del tercer escenario se obtuvo la distribución del consumo de potencia cuando el módulo *Payload* se encontraba activado. De tal forma que se determinaron parámetros de estadística descriptiva para discutir los resultados reales y teóricos.

#### **5.1.4. Desarrollo de un presupuesto de potencia para un *CubeSat 2U***

##### **Consumo de potencia por módulo para un *CubeSat 2U***

Para el desarrollo del presupuesto de potencia de un *CubeSat 2U* se contó con la integración de un nuevo módulo para el *CubeSat 2U* el cual es el sistema de Desorbitación. Considerando los resultados reales del consumo de potencia de los módulos en el Quetzal-1 se determinaron nuevos parámetros de consumo de potencia, pico y nominal, para los módulos *COMMS*, *Payload*, *ADCS*. Cabe mencionar que debido a la falta de información de los módulos *OBC*, *ADM* y *EPS*; se utilizaron los consumos de potencia determinados en pruebas de laboratorio en UVG para el Quetzal-1.

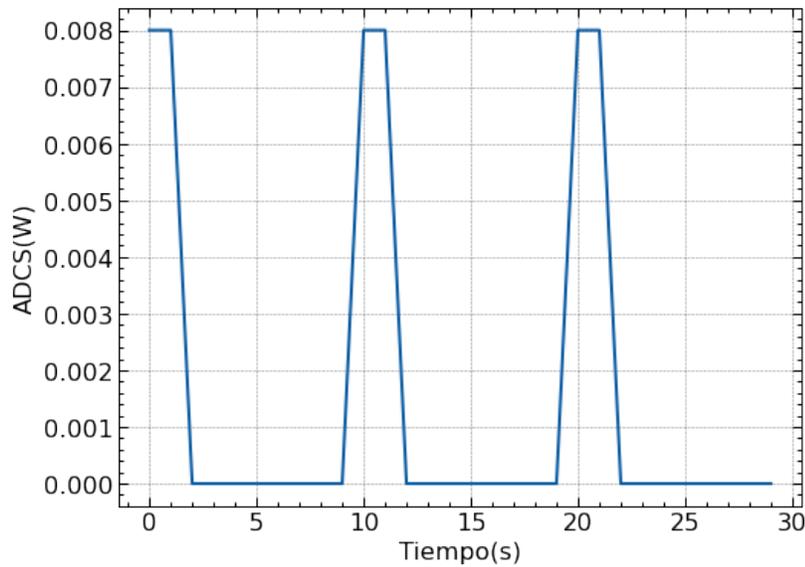
##### **Módulo *ADCS***

El módulo *ADCS* esta programado de tal manera que recopila datos de los componentes cada 10 segundos, las mediciones de los componentes requieren de 2 segundos para ser recopilados. Durante la medición de los datos es donde el módulo consume la carga pico de potencia. Después de este proceso es donde el módulo consume la carga nominal.

Cuadro 10: Parámetros de consumo de potencia de ADCS.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema ADCS	0.008	0.00

Figura 25: Consumo de potencia de *ADCS* durante 30 segundos



Por la alta frecuencia de los datos durante las simulaciones en una órbita, se presenta el consumo de potencia del módulo durante 30 segundos para apreciar el patrón de consumo que no se puede observar tomando en cuenta todos los segundos de una órbita completa.

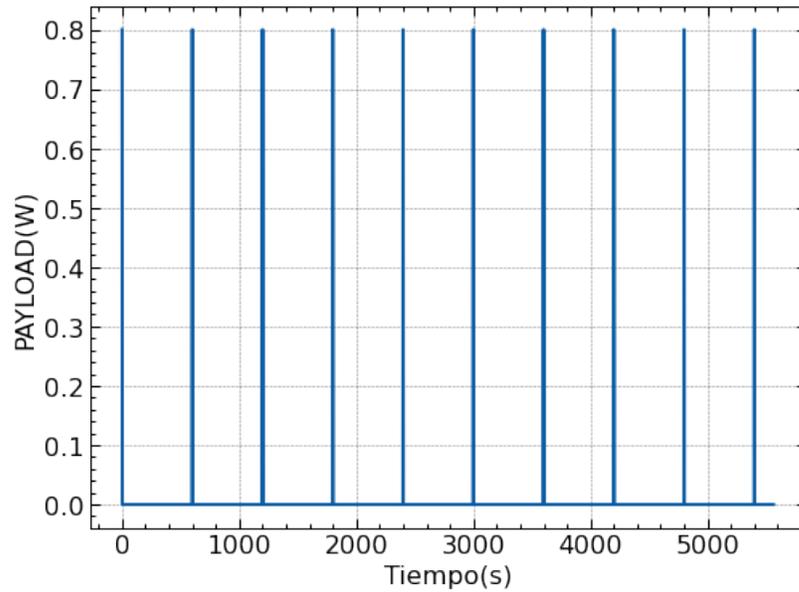
### Módulo *Payload*

El módulo *Payload* es el encargado de tomar fotografías, el módulo se activa en momentos específicos durante la misión. Por lo que, para el análisis del presupuesto de consumo de potencia del módulo, se simuló que el módulo tomaba una fotografía, consumiendo la carga pico, cada 600 segundos durante una órbita. Se analizó el impacto del uso continuo del *Payload*.

Cuadro 11: Parámetros de consumo de potencia de Payload.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema Payload	0.8	0.00

Figura 26: Consumo de potencia de *Payload* en una órbita



### Módulo *ADM*

EL subsistema *ADM* es activado en momentos específicos durante la misión. Por lo que para realizar la simulación se programó de tal manera que el despliegue inicie en el segundo 1800 de la órbita. Cuando se activa, el módulo consume la potencia pico durante 30 segundos y cuenta con un tiempo de espera de 5 segundos para activar la siguiente resistencia. De tal forma que se despliegan en 4 ciclos, siendo un ciclo por cada resistencia.

Cuadro 12: Parámetros de consumo de potencia de ADM.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema ADM	2.96	0.00

Figura 27: Consumo de potencia de *ADM*

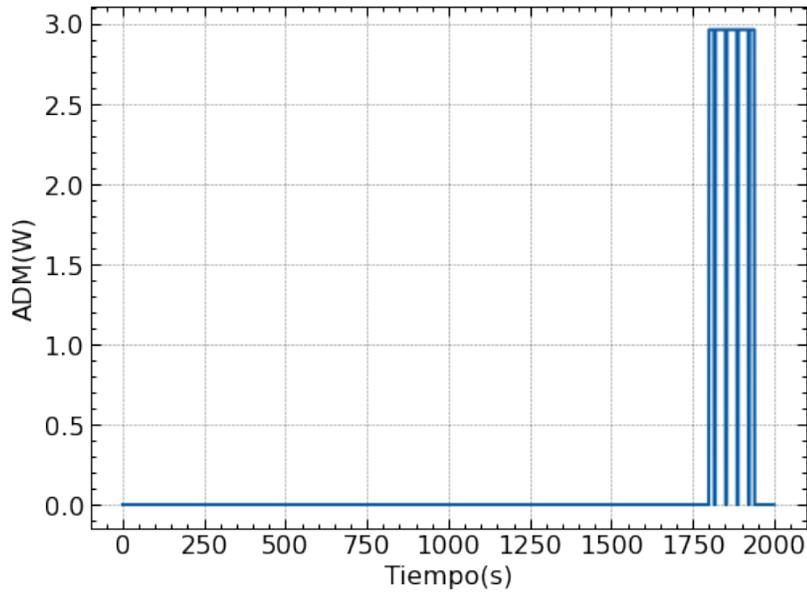
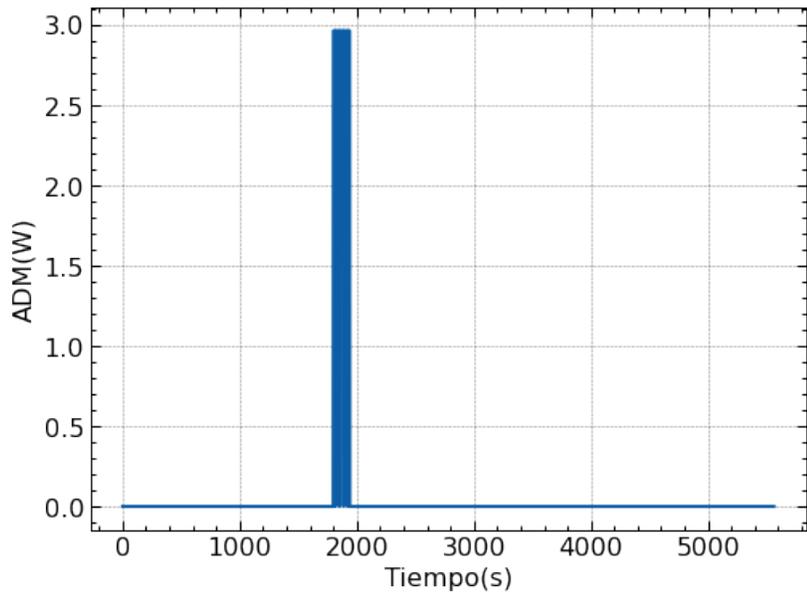


Figura 28: Consumo de potencia de *ADM* en una órbita



## Módulo *COMMS*

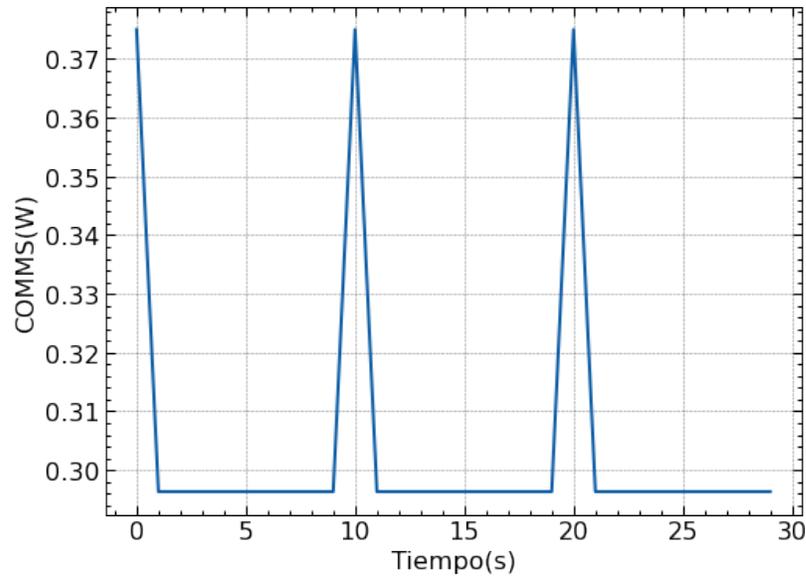
El módulo *COMMS* cuenta con dos estados de operación, el estado de recepción de información (*RX*) y el estado de transferencia de información (*TX-RX*). En el estado de recepción de información el módulo consume en carga nominal. El Quetzal-1 realizaba la transferencia de información que ocurre durante 300 ms, consumiendo la potencia pico. El proceso de transmitir la información se repite cada 10 segundos. Siendo el estado *TX-RX* en el que se encuentra operando el módulo en el espacio, debido a que la simulación es durante

una órbita, en la Figura 29 se muestra el consumo de potencia del módulo *COMMS* durante 30 segundos.

Cuadro 13: Parámetros de consumo de potencia de *COMMS*.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema <i>COMMS</i>	0.375	0.296

Figura 29: Consumo de potencia de *COMMS* durante 30 segundos



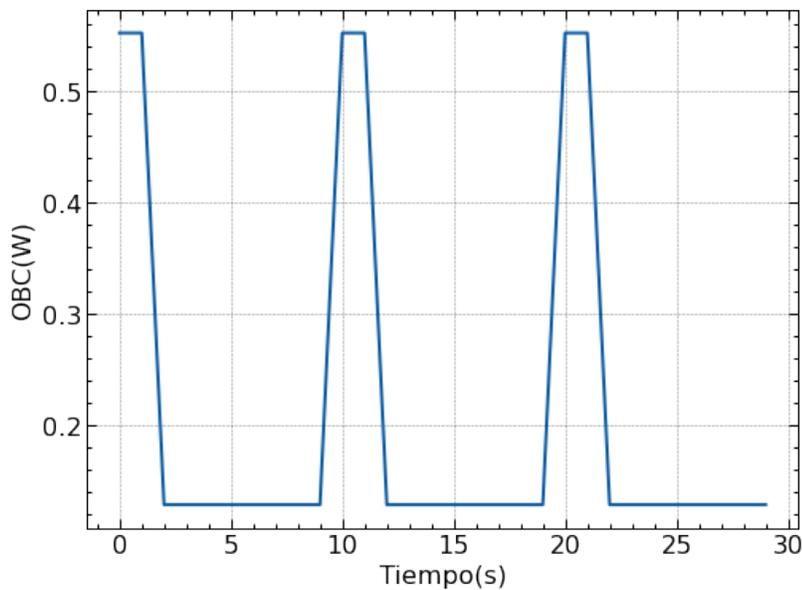
### Módulo *OBC*

La simulación del consumo de potencia del módulo *OBC* corresponde a que consume la carga pico cuando el módulo de comunicaciones se encuentra transmitiendo información, cuando el módulo *Payload* se encuentre tomando una fotografía o cuando el módulo *ADCS* recopila información del resto de módulos. De lo contrario, consume la carga nominal. Debido que a la simulación es durante una órbita en la Figura 30 se muestra el consumo de potencia del módulo *OBC* durante 30 segundos.

Cuadro 14: Parámetros de consumo de potencia de OBC.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema <i>OBC</i>	0.552	0.128

Figura 30: Consumo de potencia de *OBC* durante 30 segundos



## Módulo *EPS*

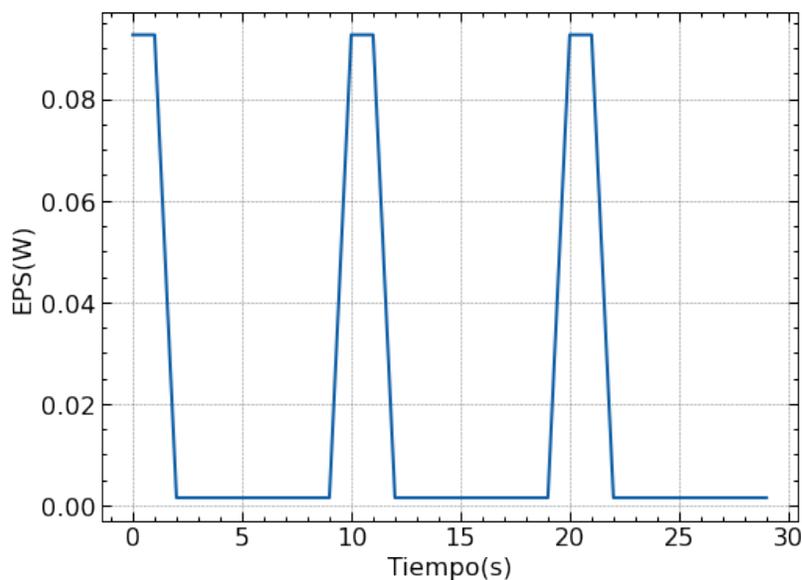
El módulo *EPS* cuenta con un micro controlador que tiene la función de encender y apagar el suministro de energía a los diferentes módulos. El modulo *EPS* consumirá la carga pico cuando módulo de comunicaciones se encuentra transmitiendo información, cuando el módulo *Payload* se encuentre tomando una fotografía o cuando el módulo *ADCS* recopila información del resto de módulos.

La mayor parte del tiempo el EPS operará en carga pico, durante la misión, ya que en el módulo de comunicaciones se encuentra en modalidad *TX-RX*. Debido que a la simulación es durante una órbita en la Figura 31 se muestra el consumo de potencia del módulo *EPS* durante 30 segundos.

Cuadro 15: Parámetros de consumo de potencia de EPS.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema EPS	0.093	0.002

Figura 31: Consumo de potencia de EPS durante 30 segundos



### Módulo *Deorbit System*

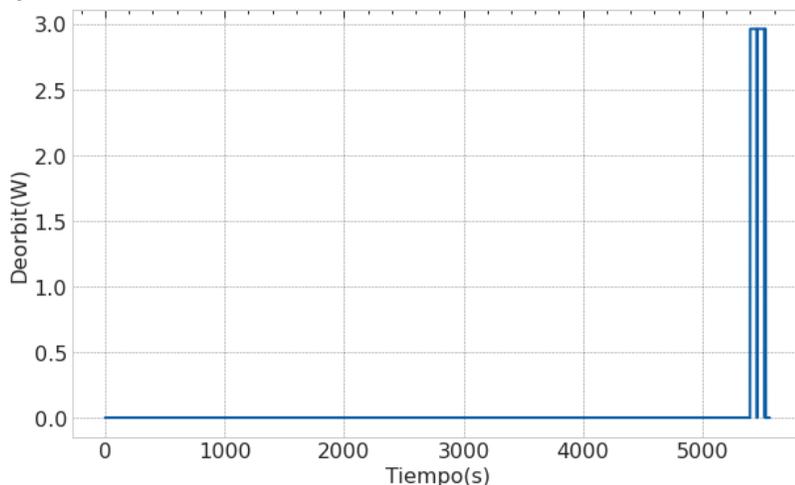
Para el correcto funcionamiento del *Deorbit System* se utilizarán 2 resistencias *Vishay Pro1 Resistor* para el despliegue de las dos velas del módulo. El módulo cumple su función en un momento determinado durante la misión por lo que se simuló cerca del final de una órbita, al segundo 5400. En el momento que se acciona la resistencia, consume la carga pico de potencia durante 1 minuto.

Después que la primera resistencia completa su ciclo existe un tiempo de espera de 5 segundos para activar la siguiente resistencia. El consumo de potencia por parte de las resistencias se debe a las pruebas realizadas en laboratorio para el mecanismo de despliegue de las antenas del módulo *ADM* para el Quetzal-1.

Cuadro 16: Parámetros de consumo de potencia de Deorbit.

Componente	Pico	Reposo
	Potencia (W)	Potencia (W)
Subsistema Deorbit	2.96	0.00

Figura 32: Consumo de potencia de *Deorbit System* en una órbita



### Simulación de generación de potencia para los paneles solares durante un día, para un *CubeSat 2U*

Para realizar la simulación del estado de carga de las baterías se obtuvo la generación de potencia de un *CubeSat 2U*. Se utilizó el simulador de generación de potencia empleado para el presupuesto de potencia del Quetzal-1, realizado en UVG. Para el cálculo de generación de potencia para un *CubeSat 2U* se utilizaron los parámetros de la celda solar que se utilizó en el Quetzal-1.

Cuadro 17: Especificaciones de la celda solar utilizada en la simulación

Fabricante	AZUR SPACE Solar Power GMBH
Área de celda solar	$3.018e-3m^2$
Eficiencia de celda solar	17.34%
Factor de temperatura de la celda solar	0.85

El parámetro de la eficiencia de la celda solar fue determinado mediante pruebas de laboratorio previo al lanzamiento del Quetzal-1. Los paneles solares del *CubeSat* están compuestos por 2 celdas solares conectadas en paralelo, tomando en cuenta que en la simulación se analizó un *CubeSat 2U*, los 4 laterales del *CubeSat* cuentan con 4 celdas solares. Además se tomó como referencia que la cara inferior del Quetzal-1 utilizó una celda solar para la correcta operación del riel y la cámara del módulo *Payload*.

## Simulación de los escenarios esperados durante el vuelo de un *CubeSat 2U*

Los escenarios consideran posibles situaciones dependiendo de qué módulos se encuentran desactivados o activados, dependiendo de qué procedimiento se está realizando en esa fase de la misión. Se determinó el valor esperado del consumo de potencia del *CubeSat 2U* en cada escenario. Utilizando el lenguaje de programación *Python*, se simuló cada escenario tomando en cuenta el consumo total de potencia durante una órbita representado por los módulos activos en cada escenario.

Debido a que la simulación de generación de potencia se realizó durante un día, el consumo de potencia de los escenarios se concatenó el equivalente en número de órbitas para poder analizar el cambio en el estado de carga de las baterías.

De tal forma que se obtuvo el consumo total de energía y se determinó la energía disponible, en Wh, siendo la diferencia entre la energía generada por los paneles solares y el consumo total del escenario analizado. Los cálculos realizados tuvieron la finalidad de obtener el estado de carga de las baterías. Para las simulaciones se utilizaron las especificaciones de las baterías, *Polymer Li-ion Rechargeable Battery model: DTP605068*, utilizadas en el *Quetzal-1*, tomando en cuenta que se simuló utilizando dos baterías conectadas en paralelo.

Cuadro 18: Especificaciones de las baterías de Polímero de litio

Fabricante	DTP
Capacidad nominal	2000 mAh
Voltaje nominal	3.7 V

Cabe mencionar que en las simulaciones de los módulos, los módulos que no cambian de modalidad durante la misión son *ADCS*, *EPS* y *OBC*. Por lo que, para los escenarios se deben de considerar como se muestran en el siguiente cuadro.

Cuadro 19: Módulos que se encuentran activados durante la simulación de los escenarios, para presupuesto de potencia de un *CubeSat 2U*

<i>ADCS</i>	<i>EPS</i>	<i>OBC</i>
<i>ON</i>	<i>ON</i>	<i>ON</i>

### Primer escenario

En el primer escenario se consideró que es el que ocurre después del lanzamiento, donde el *CubeSat 2U* no transmite información, por lo tanto el módulo *COMMS* se encuentra en modalidad *RX*; además, se activará la secuencia del despliegue de antenas del módulo *ADM*.

El *CubeSat* se entrega al transportador del satélite al espacio, con un 100% de estado de carga de las baterías. Sin embargo, para las simulaciones se consideró que se entrega al transportador con un 60% de carga, estimando una tasa de descarga del 13% cada 120 días por lo tanto, el estado de carga inicial es del 47%.

Cuadro 20: Estado de los módulos durante la simulación del primer escenario, para presupuesto de potencia de un *CubeSat 2U*

<i>Payload</i>	<i>ADM</i>	<i>COMMS</i>	<i>Deorbit System</i>
<i>OFF</i>	<i>ON</i>	<i>RX</i>	<i>OFF</i>

### Segundo escenario

Después de la liberación y de la activación de la secuencia del despliegue de antenas de comunicación, el módulo *COMMS* permanecerá en modalidad de transmisión de información. Por lo tanto, se empezará a recolectar información del estado del *CubeSat* además, se simuló el caso de un problema con el despliegue de las antenas por lo que se repetirá la secuencia en este escenario. Para la simulación del estado de carga de las baterías se utilizó un estado de carga del 47% nuevamente.

Cuadro 21: Estado de los módulos durante la simulación del segundo escenario, para presupuesto de potencia de un *CubeSat 2U*

<i>Payload</i>	<i>ADM</i>	<i>COMMS</i>	<i>Deorbit System</i>
<i>OFF</i>	<i>ON</i>	<i>TX-RX</i>	<i>OFF</i>

### Tercer escenario

En el tercer escenario las antenas fueron desplegadas correctamente y el módulo *COMMS* se encuentra recolectando y transmitiendo información por el resto de la misión. Se simuló la activación del módulo *Payload*, que toma fotografías. Se consideró el supuesto para las simulaciones del Quetzal-1, donde el *CubeSat* puede activar el módulo *Payload* cuando el estado de carga es igual o superior al 65%.

Cuadro 22: Estado de los módulos durante la simulación del tercer escenario, para presupuesto de potencia de un *CubeSat 2U*

<i>Payload</i>	<i>ADM</i>	<i>COMMS</i>	<i>Deorbit System</i>
<i>ON</i>	<i>OFF</i>	<i>TX-RX</i>	<i>OFF</i>

### Cuarto escenario

En el cuarto escenario se simuló el despliegue las dos velas del módulo *Deorbit System*, el cual se activará cerca del final de la misión. Se consideró que el estado de carga de las baterías sería del 30%.

Cuadro 23: Estado de los módulos durante la simulación del tercer escenario, para el presupuesto de potencia de un *CubeSat 2U*

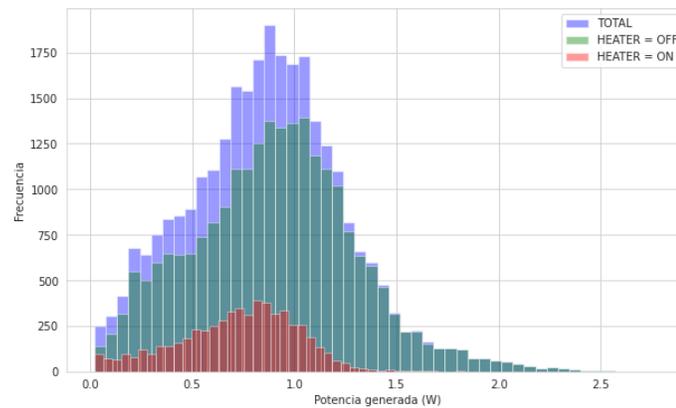
<i>Payload</i>	<i>ADM</i>	<i>COMMS</i>	<i>Deorbit System</i>
<i>OFF</i>	<i>OFF</i>	<i>TX-RX</i>	<i>ON</i>

## 6.1. Resultados de potencia real del Quetzal-1

### 6.1.1. Resultados de potencia bajo efectos de luz solar

#### Generación de potencia por las baterías bajo efectos de luz solar

Figura 32: Potencia generada por las baterías bajo efectos de luz solar



Cuadro 24: Parámetros descriptivos de la potencia generada por baterías bajo efectos de luz solar

Parámetro	Potencia total (W)	Potencia <i>HEATER ON</i> (W)	Potencia <i>HEATER OFF</i> (W)
Media	$0.864 \pm 0.0045$	$0.724 \pm 0.0075$	$0.900 \pm 0.0052$
<i>std</i>	0.389	0.292	0.402
Min	0.020	0.020	0.020
50 %	0.864	0.752	0.906
Max	2.787	1.832	2.788

Cuadro 25: Longitud de los datos de generación de potencia, bajo efectos de luz solar

Tipo de análisis	No. de datos	Porcentaje
Generación de P. total	28,931	100 %
Generación de P. <i>HEATER ON</i>	5,887	20.34 %
Generación de P. <i>HEATER OFF</i>	23,121	79.91 %

### Consumo de potencia por las baterías bajo efectos de luz solar

Figura 33: Consumo de potencia por las baterías bajo efectos de luz solar

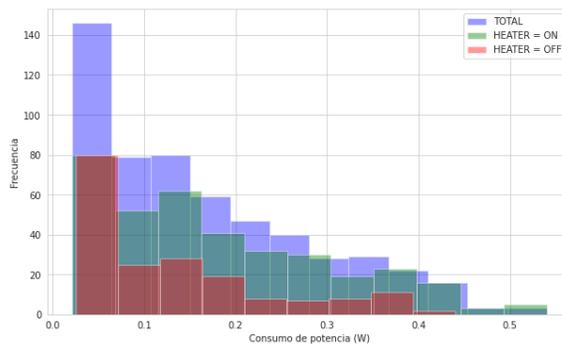
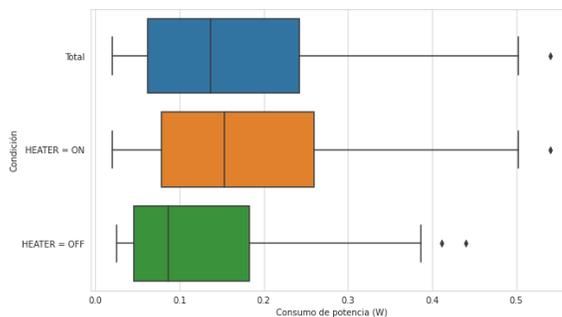


Figura 34: *Boxplot* del consumo de potencia por las baterías bajo efectos de luz solar



Cuadro 26: Parámetros descriptivos del consumo potencia por las baterías bajo efectos de luz solar

Parámetro	Potencia total (W)	Potencia <i>HEATER ON</i> (W)	Potencia <i>HEATER OFF</i> (W)
Media	0.163	0.180	0.129
Min	0.020	0.020	0.025
25 %	0.062	0.079	0.045
50 %	0.137	0.153	0.087
75 %	0.242	0.259	0.182
Max	0.540	0.540	0.440

Cuadro 27: Longitud de los datos de generación y consumo de potencia, bajo efectos de luz solar

Tipo de análisis	No. de datos	Porcentaje
Potencia bajo efectos de luz	29,483	100 %
Generación de potencia	28,931	98.12 %
Consumo de potencia	552	1.87 %

### 6.1.2. Resultados de potencia del Quetzal-1 bajo eclipse

Figura 35: Consumo de potencia por las baterías bajo eclipse

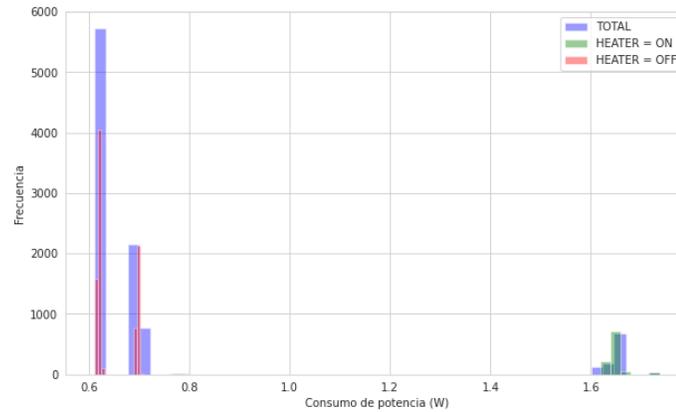
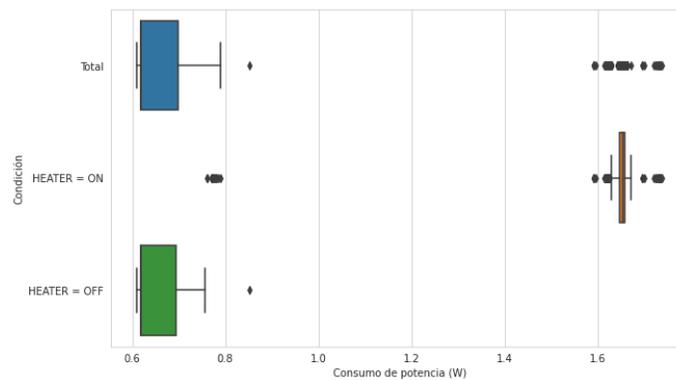


Figura 36: *Boxplot* del consumo de potencia por las baterías bajo eclipse



Cuadro 28: Parámetros descriptivos del consumo potencia por las baterías bajo eclipse

Parámetro	Potencia total (W)	Potencia <i>HEATER ON</i> (W)	Potencia <i>HEATER OFF</i> (W)
Min	0.610	0.760	0.610
25 %	0.619	1.645	0.619
50 %	0.619	1.653	0.619
75 %	0.697	1.657	0.693
Max	1.736	1.736	0.852

Cuadro 29: Longitud de los datos de potencia generales del análisis

Tipo de análisis	No. de datos	Porcentaje
Longitud total	39,175	100 %
Generación de P. luz solar	28,931	73.85 %
Consumo de P. eclipse	9,692	26.14 %

## 6.2. Resultados de efectos de temperatura sobre las baterías del Quetzal-1

Figura 37: Histograma los rangos de temperatura de las baterías del Quetzal-1

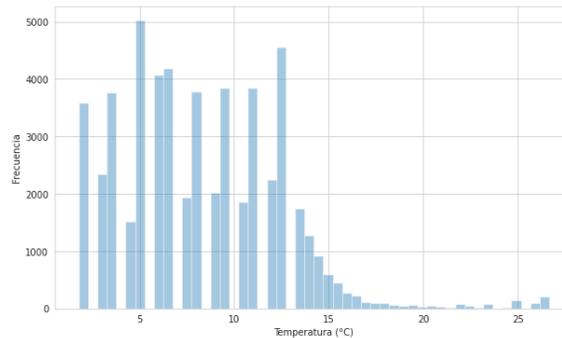
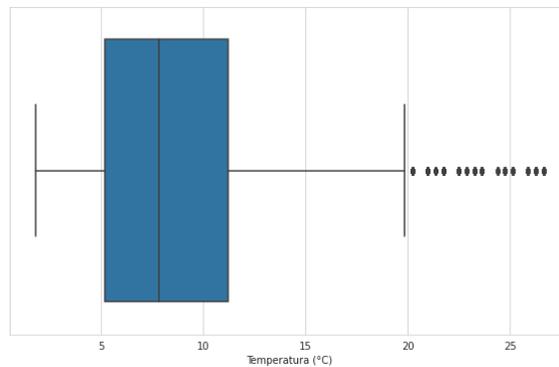


Figura 38: *Boxplot* los rangos de temperatura de las baterías del Quetzal-1



Cuadro 30: Parámetros descriptivos los rangos de temperatura de las baterías del Quetzal-1

Parámetro	Temperatura (°C)
Min	1.770
50 %	7.80
Max	26.65

## 6.3. Validación del presupuesto de consumo de potencia del Quetzal-1

### 6.3.1. Resultados del consumo de potencia del módulo *ADCS*

Figura 39: *Boxplot* del consumo de potencia real del módulo *ADCS* durante 26 semanas

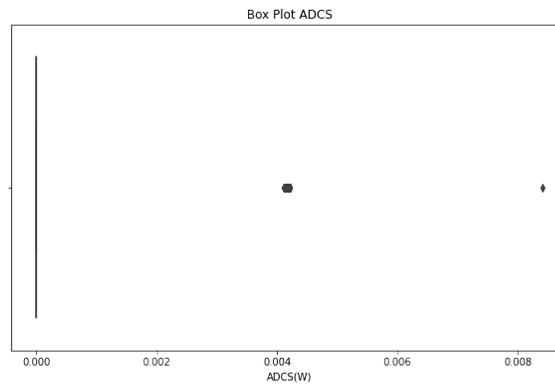
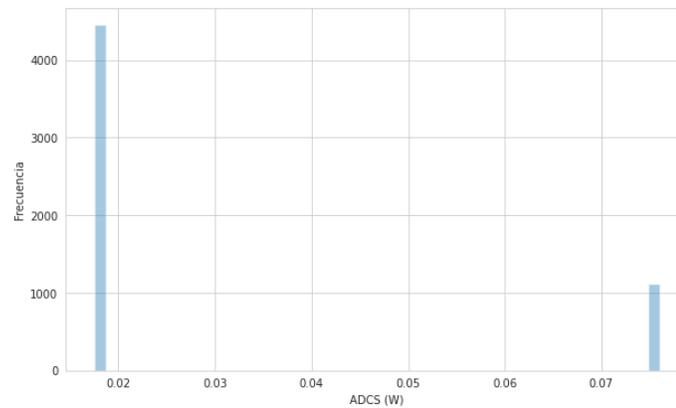


Figura 40: Histograma del consumo de potencia teórico del módulo *ADCS* durante las simulaciones



Cuadro 31: Resultados reales y teóricos del consumo de potencia del módulo *ADCS*

Potencia <i>ADCS</i>	Teórico (W)	Real (W)	Error (%)
Nominal	0.02	0.00	100
Pico	0.076	0.008	89

### 6.3.2. Resultados del consumo de potencia del módulo *COMMS*

Figura 41: Histograma del consumo de potencia real del módulo *COMMS* durante 26 semanas

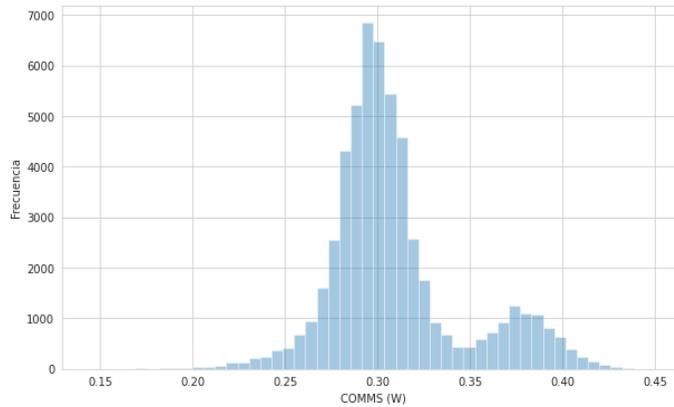


Figura 42: Histograma del consumo de potencia real del módulo *COMMS*, en modalidad *RX*, durante 26 semanas

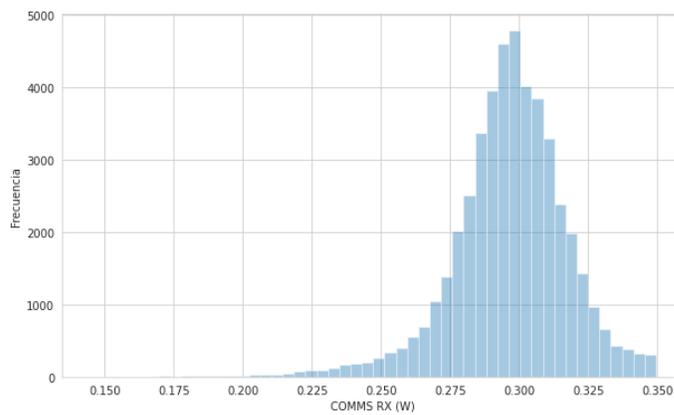


Figura 43: Histograma del consumo de potencia real del módulo *COMMS*, en modalidad *TX-RX*, durante 26 semanas

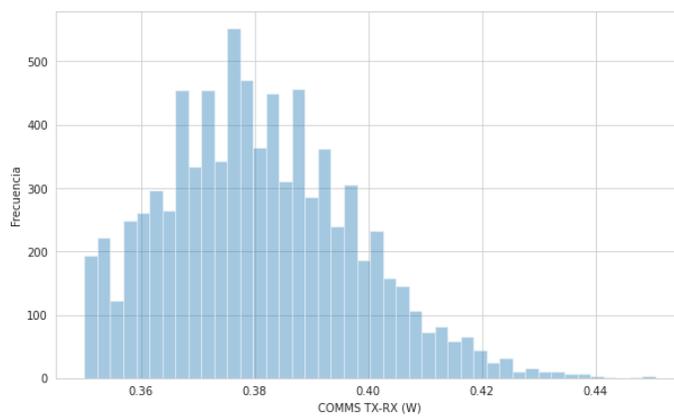
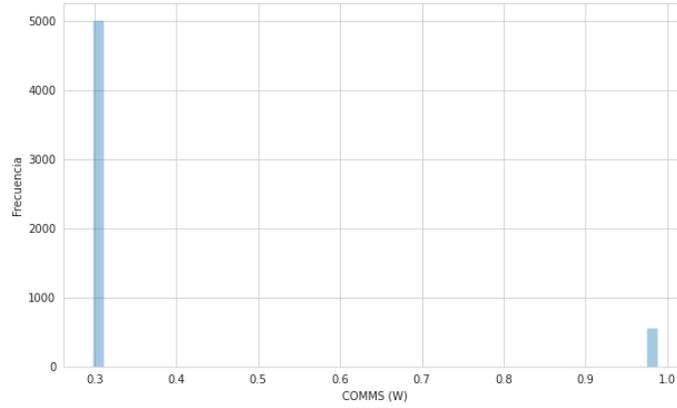


Figura 44: Histograma del consumo de potencia teórica del módulo *COMMS* durante las simulaciones

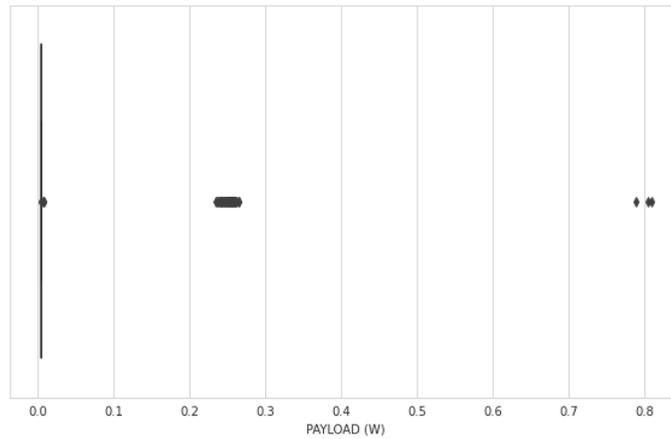


Cuadro 32: Resultados reales y teóricos del consumo de potencia del módulo *COMMS*

Potencia <i>COMMS</i>	Teórico (W)	Real (W)	Error (%)
RX	0.297	0.296	0.24
TX-RX	0.988	0.380	61.54

### 6.3.3. Resultados del consumo de potencia del módulo *Payload*

Figura 45: *Boxplot* del consumo de potencia real del módulo *Payload* durante 26 semanas



Cuadro 33: Resultados reales y teóricos del consumo de potencia del módulo *Payload*

Potencia <i>Payload</i>	Teórico (W)	Real (W)	Error (%)
Pico	2.137	0.809	62.14

### 6.3.4. Resultados del consumo de potencia del primer y segundo escenario

Figura 46: Histograma del consumo de potencia real del Quetzal-1 durante el primer y segundo escenario

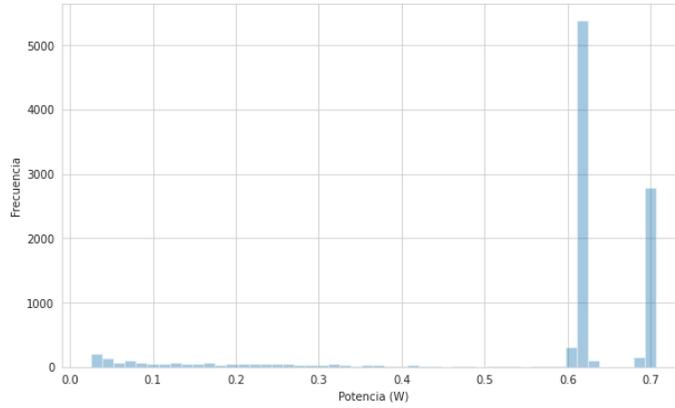


Figura 47: Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el primer escenario, durante una órbita

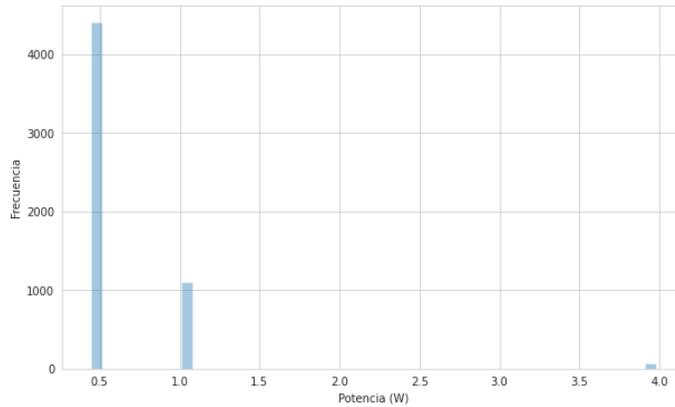
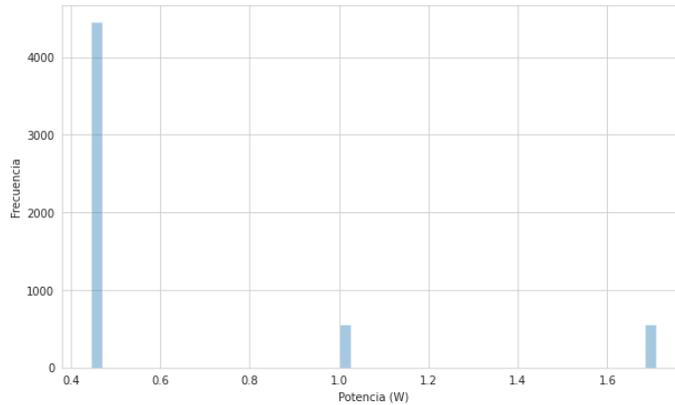


Figura 48: Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el segundo escenario, durante una órbita



Cuadro 34: Parámetros descriptivos, reales y teóricos, del consumo potencia del Quetzal-1 durante el primer y segundo escenario

Parámetro	Primer E. y Segundo E. (W)	Primer E. teórico (W)	Segundo E. teórico (W)
Media	0.57	0.597	0.628
Min	0.020	0.444	0.444
25 %	0.614	0.444	0.444
50 %	0.619	0.444	0.444
75 %	0.693	0.444	0.444
Max	0.706	3.977	1.708

### 6.3.5. Resultados del consumo de potencia del tercer escenario

Figura 49: Histograma del consumo de potencia real del Quetzal-1 durante el tercer escenario

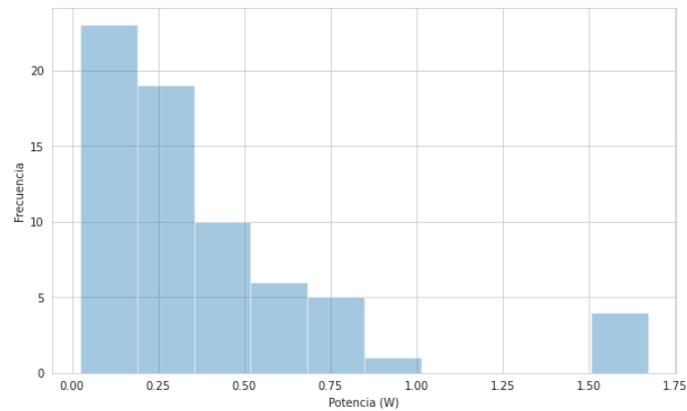
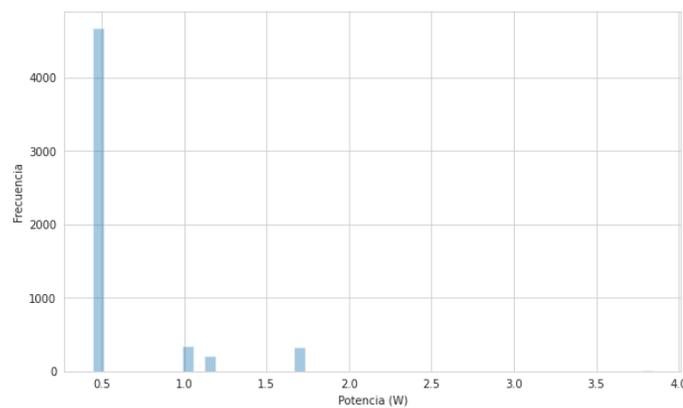


Figura 50: Histograma de la simulación del consumo de potencia teórico del Quetzal-1 en el tercer escenario, durante una órbita



Cuadro 35: Parámetros descriptivos, reales y teóricos, del consumo potencia del Quetzal-1 durante el tercer escenario

Parámetro	Tercer E. real (W)	Tercer E. teórico (W)
Media	0.336	0.587
Min	0.024	0.444
25 %	0.128	0.444
50 %	0.253	0.444
75 %	0.382	0.444
Max	1.670	3.843

### 6.3.6. Resultados del estado de carga de las baterías del Quetzal-1

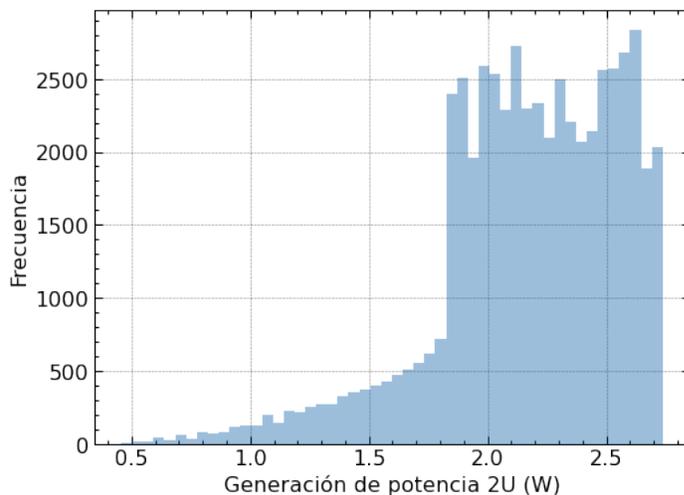
Cuadro 36: Frecuencia de un estado de carga de las baterías igual al 100%, durante la misión

Variable	No. de datos	Porcentaje
Estado de carga	39,227	71.13 %

## 6.4. Desarrollo de un presupuesto de potencia para un *CubeSat 2U*

### 6.4.1. Simulación de generación de potencia total por los paneles solares para un *CubeSat 2U*

Figura 51: Histograma de la simulación de la generación de potencia de los paneles solares de un *Cubesat 2U*, durante un día



Cuadro 37: Parámetros teóricos descriptivos de la generación de potencia de un *Cubesat 2U*, durante un día

Parámetros	Potencia (W)
Media	1.32
Min	0.00
25 %	0.00
50 %	1.88
VE	2.09
75 %	2.30
Max	2.74

#### 6.4.2. Simulación de los escenarios del presupuesto de potencia para un *CubeSat 2U*

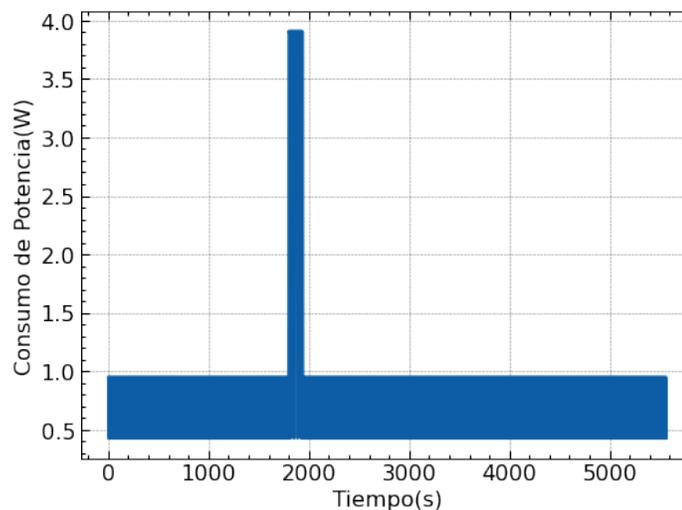
Cuadro 38: Resumen de los escenarios analizados de las simulaciones

Escenario	<i>ADCS</i>	<i>Payload</i>	<i>ADM</i>	<i>COMMS</i>	<i>EPS</i>	<i>OBC</i>	<i>Deorbit</i>
Primer Escenario	<i>ON</i>	<i>OFF</i>	<i>ON</i>	<i>RX</i>	<i>ON</i>	<i>ON</i>	<i>OFF</i>
Segundo Escenario	<i>ON</i>	<i>OFF</i>	<i>ON</i>	<i>TX-RX</i>	<i>ON</i>	<i>ON</i>	<i>OFF</i>
Tercer Escenario	<i>ON</i>	<i>ON</i>	<i>OFF</i>	<i>TX-RX</i>	<i>ON</i>	<i>ON</i>	<i>OFF</i>
Cuarto Escenario	<i>ON</i>	<i>OFF</i>	<i>OFF</i>	<i>TX-RX</i>	<i>ON</i>	<i>ON</i>	<i>ON</i>

#### 6.4.3. Resultados del primer escenario del presupuesto de potencia para un *CubeSat 2U*

##### Consumo de potencia total durante una órbita

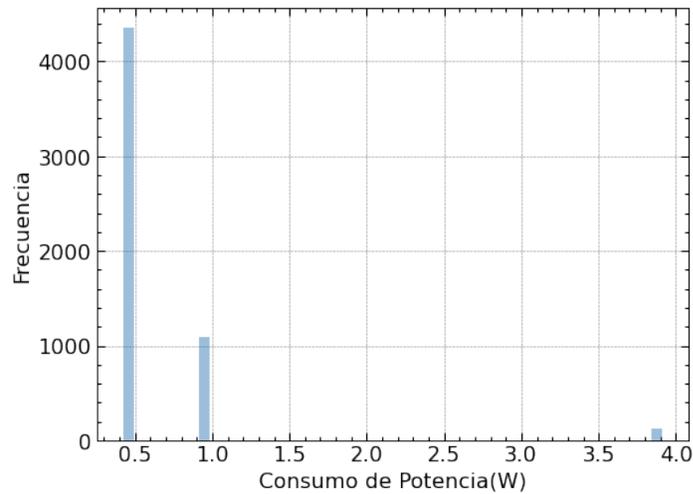
Figura 52: Simulación del consumo de potencia total, por los módulos, durante una órbita



Cuadro 39: Parámetros del consumo de potencia en el primer escenario

Parámetro	Potencia (W)
Valor esperado	0.43
Máximo por <i>ADM</i>	3.91
Máximo	0.95

Figura 53: Histograma del consumo de potencia durante el primer escenario



### Resultados del estado de carga de las baterías durante un día

Figura 54: Histograma de la potencia de entrada a las baterías

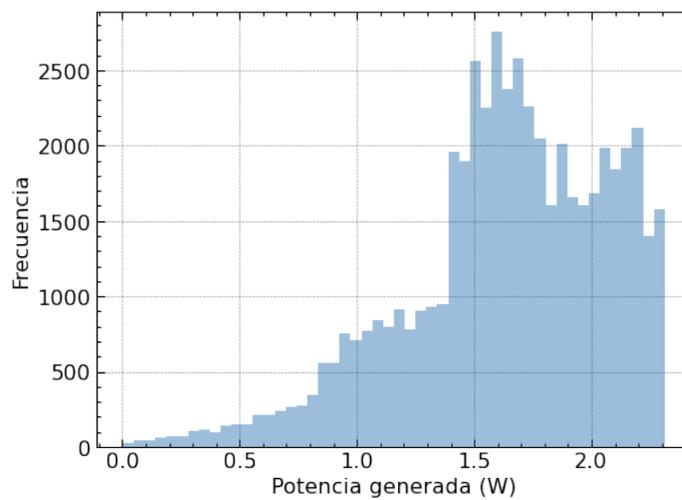
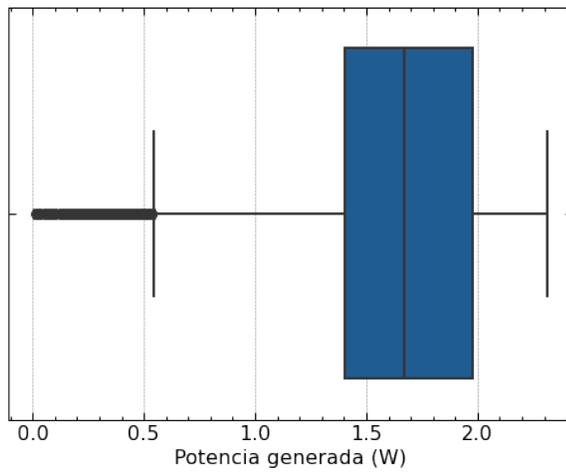


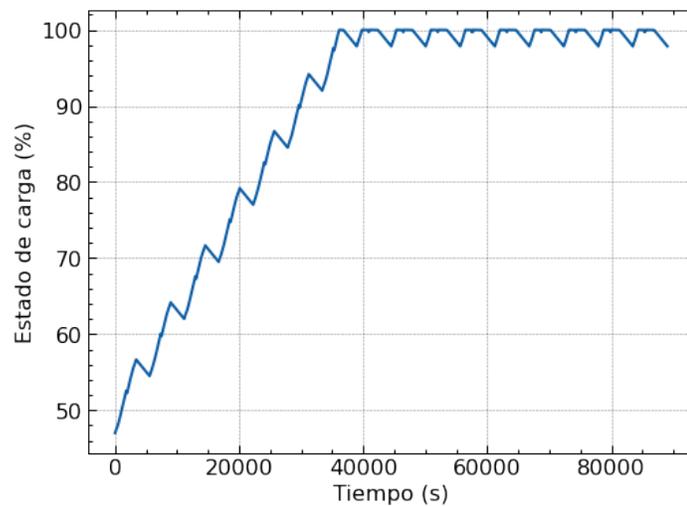
Figura 55: *Boxplot* de la potencia de entrada a las baterías



Cuadro 40: Parámetros descriptivos de la potencia de entrada a las baterías

Parámetro	Potencia(W)
Media	1.634
Min	0.003
25 %	1.404
50 %	1.670
75 %	1.977
Max	2.311

Figura 56: Gráfico de líneas del estado de carga de las baterías durante el primer escenario



Cuadro 41: Parámetros relevantes del análisis del estado de carga de las baterías

Parámetro	Estado de carga (%)
Máxima pérdida de energía	2.15
Tasa de carga total	50.84
Tasa de carga por órbita	3.17

#### 6.4.4. Resultados del segundo escenario del presupuesto de potencia para un *CubeSat 2U*

##### Consumo de potencia durante una órbita

Figura 57: Consumo de potencia total durante el segundo escenario

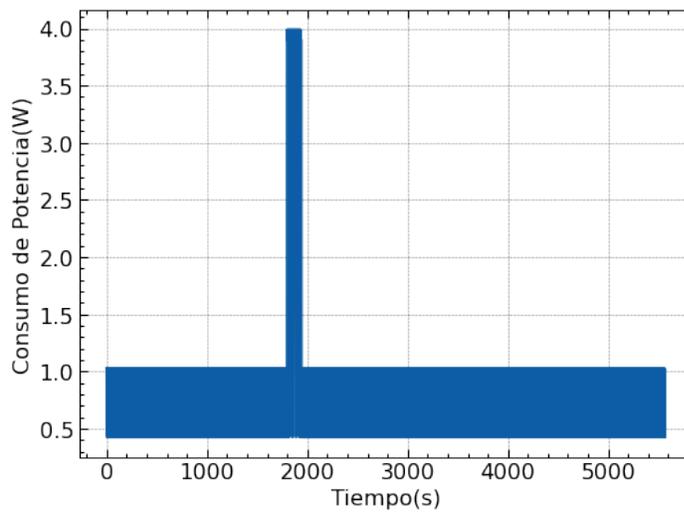
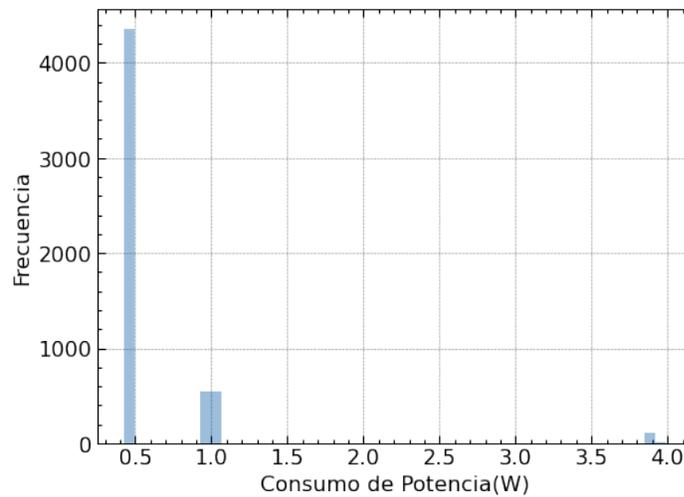


Figura 58: Histograma del consumo de potencia total durante el segundo escenario



Cuadro 42: Parámetros del consumo de potencia en el segundo escenario

Parámetro	Potencia (W)
Valor esperado	0.43
Máximo por <i>ADM</i>	3.99
Máximo	1.02

## Resultados del estado de carga de las baterías durante un día

Figura 59: Histograma de la potencia generada por las baterías durante el segundo escenario

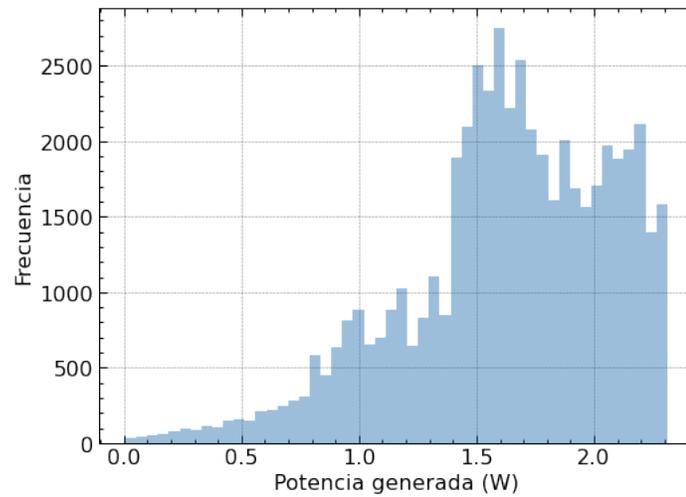
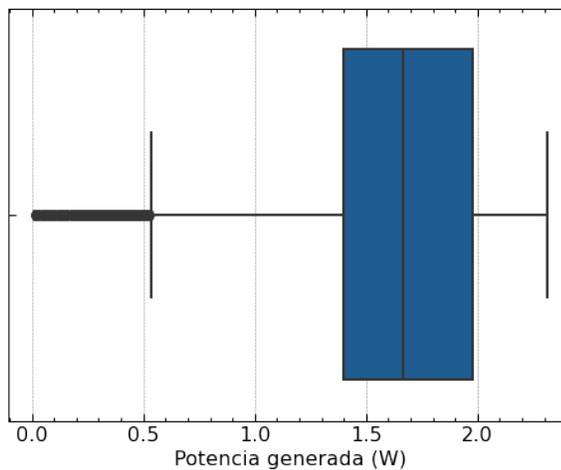


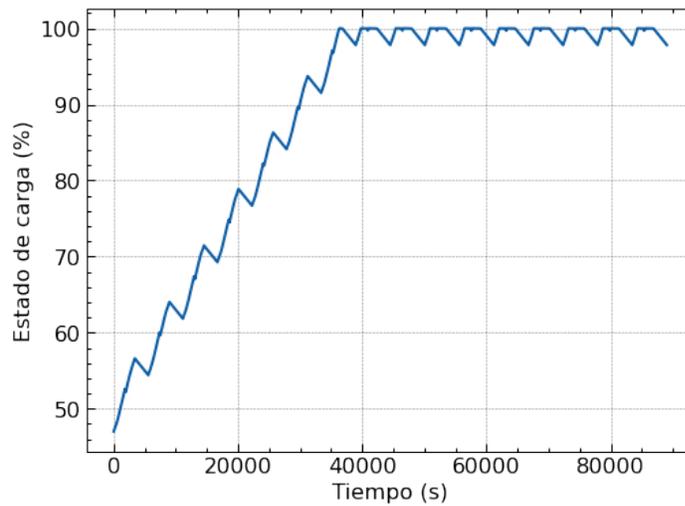
Figura 60: *Boxplot* de la potencia generada por las baterías durante el segundo escenario



Cuadro 43: Parámetros descriptivos de la potencia de entrada a las baterías

Parámetro	Potencia(W)
Media	1.635
<i>std</i>	0.440
Min	0.007
25 %	1.404
50 %	1.672
75 %	1.987
Max	2.311

Figura 61: Gráfico de líneas del estado de carga de las baterías durante el segundo escenario



Cuadro 44: Parámetros relevantes del análisis del estado de carga de las baterías durante el segundo escenario

Parámetro	Estado de carga (%)
Máxima pérdida de energía	2.18
Tasa de carga total	50.81
Tasa de carga por órbita	3.18

### 6.4.5. Resultados del tercer escenario del presupuesto de potencia para un *CubeSat 2U*

#### Consumo de potencia total durante una órbita

Figura 62: Consumo de potencia total durante el tercer escenario

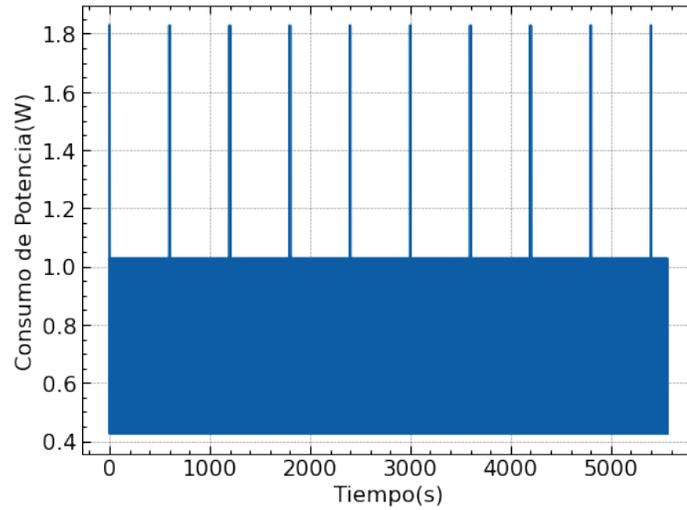
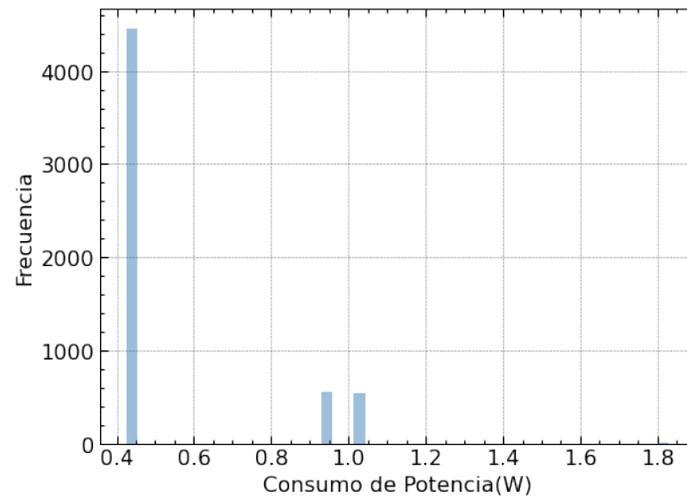


Figura 63: Histograma del consumo de potencia total durante el tercer escenario



Cuadro 45: Parámetros del consumo de potencia durante el tercer escenario

Parámetro	Potencia (W)
Valor esperado	0.43
Máximo	1.83

## Resultados del estado de carga de las baterías durante un día

Figura 64: Histograma de la potencia generada por las baterías durante el tercer escenario

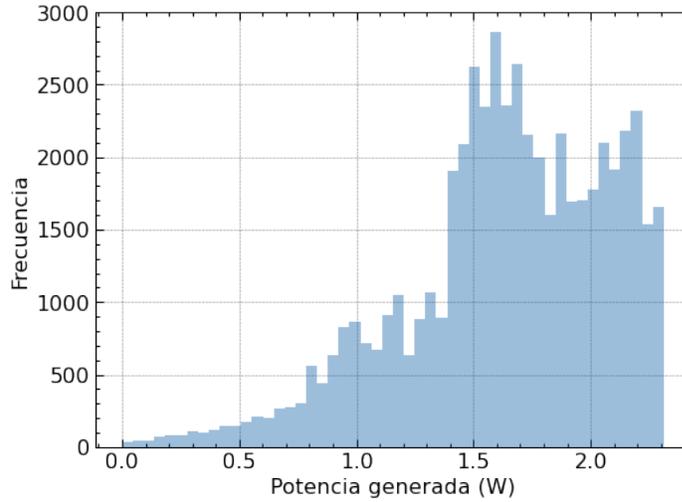
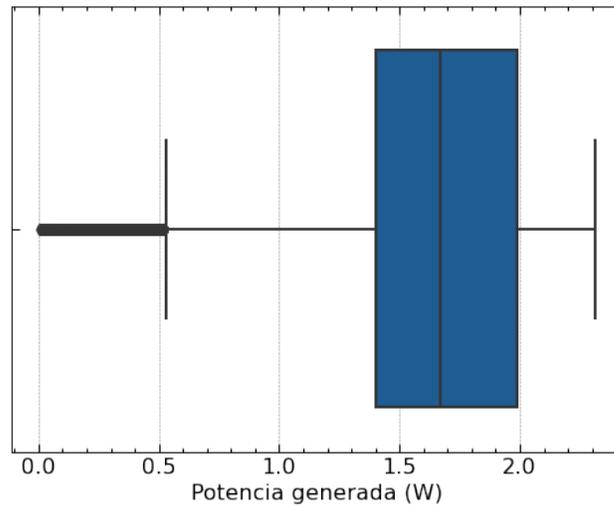


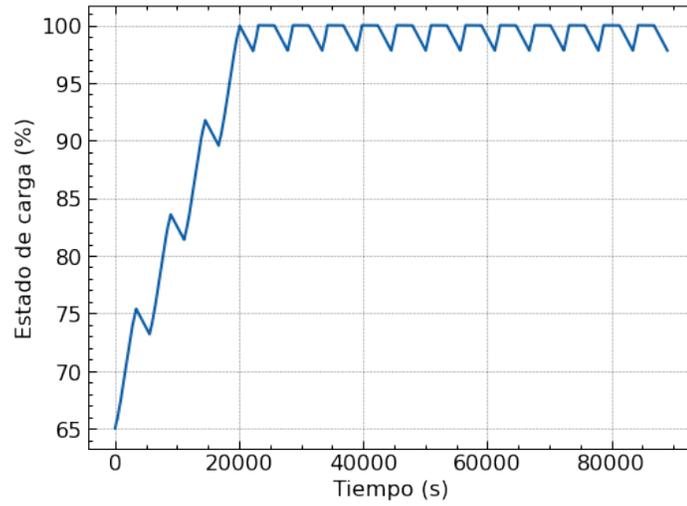
Figura 65: *Boxplot* de la potencia generada por las baterías durante el tercer escenario



Cuadro 46: Parámetros descriptivos de la potencia de entrada a las baterías

Parámetro	Potencia(W)
Min	0.0002
25 %	1.404
50 %	1.672
75 %	1.987
Max	2.311

Figura 66: Gráfico de líneas del estado de carga de las baterías durante el tercer escenario



Cuadro 47: Parámetros relevantes del análisis del estado de carga de las baterías durante el tercer escenario

Parámetro	Estado de carga (%)
Máxima pérdida de energía	2.19
Tasa de carga total	32.81
Tasa de carga por órbita	2.05

#### 6.4.6. Resultados del cuarto escenario del presupuesto de potencia para un *CubeSat 2U*

##### Consumo de potencia total durante una órbita

Figura 67: Consumo de potencia total durante el cuarto escenario

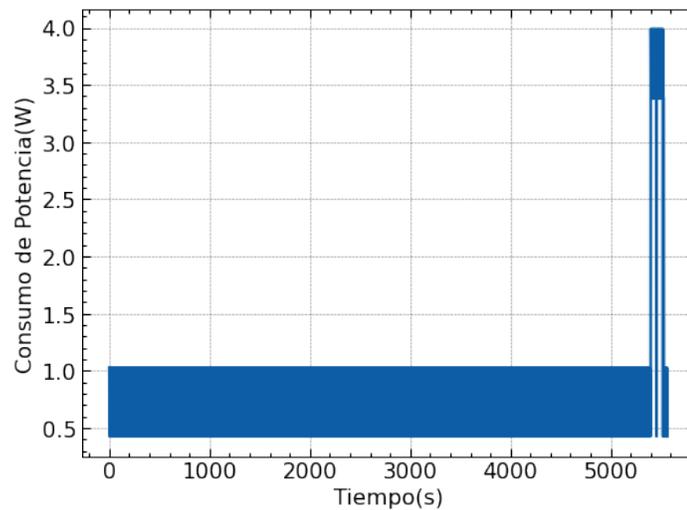
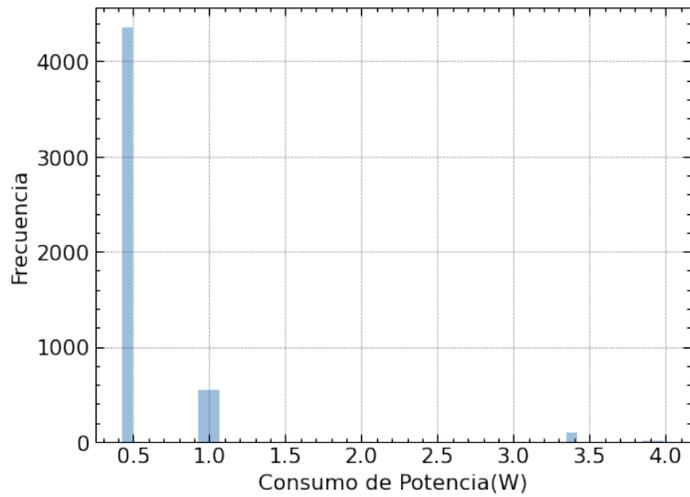


Figura 68: Histograma del consumo de potencia total durante el cuarto escenario



Cuadro 48: Parámetros del consumo de potencia durante el cuarto escenario

Parámetro	Potencia (W)
Valor esperado	0.43
Máximo por <i>Deorbit</i>	3.99
Máximo	1.02

### Resultados del estado de carga de las baterías durante un día

Figura 69: Histograma de la potencia generada por las baterías durante el cuarto escenario

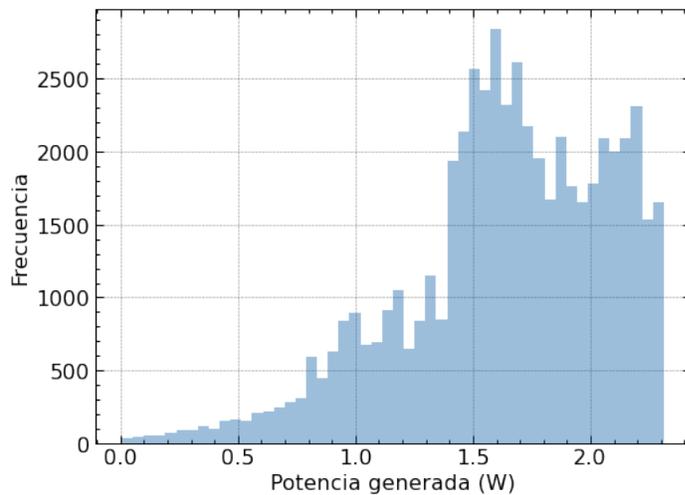
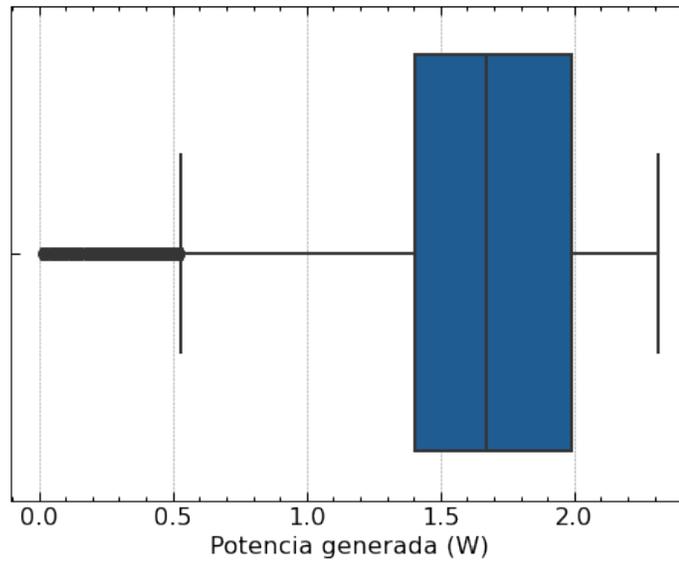


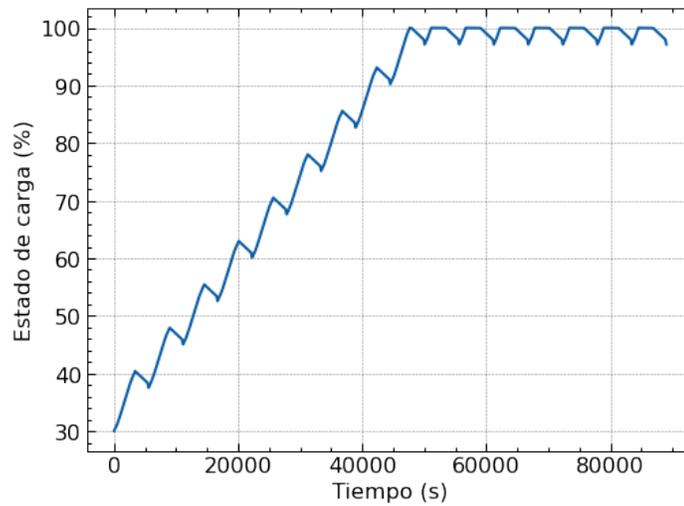
Figura 70: *Boxplot* de la potencia generada por las baterías durante el cuarto escenario



Cuadro 49: Parámetros descriptivos de la potencia generada por las baterías

Parámetro	Potencia(W)
Min	0.006
25 %	1.405
50 %	1.672
75 %	1.987
Max	2.311

Figura 71: Gráfico de líneas del estado de carga de las baterías durante el cuarto escenario



Cuadro 50: Parámetros relevantes del análisis del estado de carga de las baterías durante el cuarto escenario

Parámetro	Estado de carga (%)
Máxima pérdida de energía	2.86
Tasa de carga total	67.13
Tasa de carga por órbita	4.19

## 7.1. Análisis de potencia real del Quetzal-1

### 7.1.1. Análisis de generación de potencia por las baterías bajo efectos de luz solar

En la Figura 32 se muestra un sesgo hacia la izquierda de las distribuciones de los resultados de la potencia generada por las baterías del Quetzal-1, tomando en cuenta el estado del calentador de las baterías, siendo el color verde cuando el calentador se encontraba desactivado y color rojo cuando estaba activado.

Para el caso de la generación de potencia total se modeló como una distribución normal debido a que, como se puede observar en el Cuadro 24 la media y la mediana tienden a ser iguales. Se puede destacar el valor de la desviación estándar, que es 0.389 watts, el cual indica que los datos tienden a agruparse cerca de la media lo cual se puede ver en la tendencia leptocúrtica.

Con un nivel de confianza del 95 % la generación de potencia total por parte de las baterías, cuando el Quetzal-1 se encontraba bajo efectos de luz solar, es de  $0.86 \pm 0.0045$  watts. Otro parámetro relevante es el valor máximo, el cual es de 2.78 watts. Dado que la distribución muestra un sesgo hacia la izquierda, debido a que la cola de la distribución representa los valores atípicos respecto a los valores con menor frecuencia. En este caso, los datos de generación de potencia son aproximadamente mayores a 2 watts.

El análisis de la potencia generada por las baterías con el calentador activado fue modelado como una distribución normal. En este caso se obtuvo un resultado con un nivel de confianza del 95 % que la potencia generada es de  $0.724 \pm 0.0075$  watts. Como se puede observar el valor de la media y la mediana son significativamente diferentes, esto se debe a que

la tendencia de la distribución es platicúrtica en la cual se obtienen una baja concentración de datos en torno a su media.

En el caso de la desviación estándar se obtuvo que los datos se encuentran dispersos respecto a la media en 0.292 watts. Con un valor de generación de potencia máximo de 1.83 watts. Se puede observar en los resultados que cuando el calentador de las baterías se encuentra activado la generación de potencia se reduce, esto se debe a que el Quetzal-1 cuenta con un componente extra que está consumiendo potencia, por lo tanto el consumo total del Quetzal-1 es considerablemente mayor, por lo que la potencia promedio que cargan las baterías es significativamente menor respecto al resultado de generación de potencia total.

En el caso de la generación de potencia cuando el calentador de la baterías se encuentra desactivado, la media y la mediana tienden converger por lo que, se modeló como una distribución normal; además, se observa la tendencia mesocúrtica que presenta una concentración moderada de datos respecto a su media. Por lo tanto, con un nivel de confianza del 95 %, la potencia generada por las baterías con el calentador de las baterías desactivado, es de  $0.90 \pm 0.052$  watts, debido a que el calentador se encuentra desactivado la generación de potencia promedio es mayor lo cual se ve reflejado también en el valor máximo de potencia, de 2.78 watts.

En el Cuadro 25 se pueden observar las longitudes de los datos y su porcentaje representativo en el análisis. En el caso de la generación de potencia cuando el calentador se encuentra activado se puede observar que representa un 20.34 % de la información. Por lo que, el Quetzal-1 generó potencia con el calentador desactivado el 79.91 % de la información recolectada durante 26 semanas. Los resultados muestran que cuando el calentador de las baterías se encuentra activado, tiene un impacto en la reducción de la potencia generada. Sin embargo, la baja frecuencia de los datos muestran que no son representativos en comparación con los resultados de cuando el calentador se encuentra desactivado.

### 7.1.2. Análisis de consumo de potencia por las baterías bajo efectos de luz solar

En la Figura 33, se muestran las distribuciones de los resultados del consumo de potencia por las baterías del Quetzal-1, tomando en cuenta las condiciones del calentador. Para el análisis de consumo de potencia por las baterías total no se cuenta con una distribución normal. Cabe mencionar, que un diagrama *boxplot* permite una mejor visualización, ya que el 50 % de la agrupación de los datos se encuentra en valores de consumo cercanos a cero.

En el Cuadro 26 los parámetros que mejor describen el valor esperado son el primer cuartil y la mediana, ya que en la figura se puede observar que es el rango que mayor frecuencia tiene, respecto al resto de información. Por lo que, el consumo total esperado bajo los efectos de luz solar se encuentra entre 0.062 y 0.137 watts. Cabe mencionar que el valor máximo del consumo de potencia por las baterías es de 0.54 watts, que se encuentra como dato atípico del diagrama *boxplot*

Para el caso del consumo de potencia por las baterías cuando el calentador se encuentra activado, se puede observar en las Figuras 33 y 34, que el comportamiento de la distribución

de los datos es similar al análisis anterior. Para este caso, el consumo de potencia se encuentra entre 0.079 y 0.153 watts. Siendo los valores del primer cuartil y la mediana respectivamente. En el caso de cuando el calentador de las baterías se encuentra desactivado, de las Figuras 33 y 34, se puede observar que el consumo de potencia esperado se encuentra entre 0.045 y 0.087 watts, siendo los valores del primer cuartil y la mediana respectivamente.

Como se puede observar en el Cuadro 27 los resultados sobre el consumo de potencia por las baterías bajo los efectos de luz solar representan el 1.87% de los datos de potencia bajo los efectos de luz solar. Esto se puede apreciar también en la tendencia de los datos ya que las mayores frecuencias de los datos se encuentran cercanas a cero. Por lo tanto, a lo largo de las 26 semanas, el Quetzal-1 se encontró generando potencia el 98.12% del tiempo.

### 7.1.3. Análisis de consumo de potencia por las baterías del Quetzal-1 bajo eclipse

En la Figura 35, se pueden observar las distribuciones del consumo de potencia por las baterías bajo eclipse. Los análisis no se modelaron con un método de distribución de probabilidad, sin embargo el gráfico *boxplot* se obtuvo información relevante. Para el análisis de consumo de potencia total, el primer cuartil y la mediana convergen. Por lo tanto, el consumo de potencia total esperado, durante las 26 semanas analizadas, es de 0.619 watts. Como se puede observar los valores atípicos representan los valores máximos del consumo de potencia, siendo 1.73 watts el valor máximo.

El análisis del consumo de potencia con el calentador de las baterías activado, se puede observar en las Figuras 35 y 36. Los resultados de este análisis, corresponden a los datos atípicos del consumo de potencia total. Por lo tanto, los valores máximos del consumo de potencia por las baterías cuando el Quetzal-1 se encontraba bajo sombra corresponden a cuando el calentador se encontraba activado. Dado que, los valores del primer cuartil de 1.645 watts y el valor del tercer cuartil de 1.657 watts, convergen alrededor de la mediana. Por tanto, el consumo de potencia, con el calentador activado, es de 1.653 watts.

Para el caso del consumo de potencia por las baterías con el calentador desactivado, de las Figuras 35 y 36, se obtiene, nuevamente, el valor esperado del análisis del consumo de potencia total. Esto se debe a que la mediana, a diferencia de la normal, no es afectada por datos atípicos. Por lo tanto, el consumo de potencia por las baterías es de 0.619 watts. Cabe mencionar que, el resultado de consumo de potencia bajo eclipse con el calentador desactivado representa el consumo por parte de los módulos del Quetzal-1 ya que al estar bajo eclipse los paneles solares no pueden estar generando energía por lo tanto, las baterías solo pueden estar perdiendo energía bajo el consumo de los módulos.

## 7.2. Análisis de efectos de temperatura sobre las baterías del Quetzal-1

En la Figura 37 se muestra la distribución de los rangos de temperatura a los que se el calentador mantiene las baterías. Cabe mencionar que dentro de los datos de temperatura se

encuentran 4 registros de temperaturas menores a  $0\text{ }^{\circ}\text{C}$ , siendo el valor máximo de  $-1.25^{\circ}\text{C}$ . Además el valor máximo, positivo, registrado es de  $26.65\text{ }^{\circ}\text{C}$ . Estos datos fueron registrados, porque el calentador está programado para mantener la temperatura entre  $0$  y  $25\text{ }^{\circ}\text{C}$ . Por lo que, se debe esperar obtener una baja frecuencia de registros que se encuentren por encima de  $25\text{ }^{\circ}\text{C}$  y por debajo de  $0\text{ }^{\circ}\text{C}$ .

En el Cuadro 30 se encuentran los parámetros descriptivos de la distribución de los rangos de temperatura registrados. Dada la distribución de los datos, el parámetro que mejor describe la información es la mediana siendo un valor esperado de la temperatura de las baterías de  $7.80\text{ }^{\circ}\text{C}$ . Cabe mencionar que, el valor mínimo registrado es de  $1.77^{\circ}\text{C}$ . Como se puede observar en la Figura 38 los registros de temperatura con baja frecuencia, son los que representan la cola de la distribución. Estos valores no tienen un impacto significativo en el valor esperado.

Para el análisis de correlación entre las variables del estado de carga de las baterías y la temperatura de las baterías, se obtuvo el resultado de que el coeficiente de correlación es igual  $-0.01$ . En este caso el coeficiente es negativo, de tal forma que la relación es inversa. Sin embargo, al ser menor que  $-0.05$  indica que no tienen una relación lineal. Por lo tanto, la pérdida de energía por las baterías no se debe a los cambios de temperatura. Este resultado es el esperado ya que, como se mencionó anteriormente, el calentador de las baterías fue programado para mantener la temperatura en el rango, donde el impacto de la tasa de descarga de las baterías es el mínimo.

### 7.3. Validación del presupuesto de consumo de potencia del Quetzal-1

#### 7.3.1. Análisis del consumo de potencia del módulo *ADCS*

En la Figura 39 se puede observar que el módulo *ADCS*, durante las 26 semanas de recolección de datos, se obtuvo que el  $99.84\%$  de los datos corresponden a un consumo de potencia aproximado a cero watts. Esto se debe a que la computadora del Quetzal-1 no puede mostrar una mayor cantidad de decimales, debido a que el consumo se debe encontrar en diezmilésimas mayores que cero. En el caso que contara con una mayor capacidad, se podría apreciar el comportamiento real de la distribución del consumo de potencia.

Dadas las distribuciones del consumo de potencia real y teórico del módulo *ADCS* los parámetros estadísticos que mejor representan los datos son el máximo y mínimo. Para realizar la comparación con los datos teóricos del módulo se utilizó como consumo aproximado nominal de  $0.00$  watts. Para el caso del consumo pico se consideró el valor máximo que el módulo *ADCS* llegó a consumir, el cual fue un valor de  $0.008$  watts. Dada la distribución de los datos teóricos del módulo *ADCS*, de la Figura 40, la cual muestra el consumo con mayor frecuencia siendo de  $0.02$  watts y el consumo pico, de menor frecuencia, de  $0.076$  watts. Por lo tanto, el porcentaje de error en comparación con los datos teóricos contra los reales del consumo nominal es de  $100\%$ . Respecto al valor pico el porcentaje de error es de un  $89\%$ . Como se puede observar, en los resultados reales, el módulo *ADCS* no presenta un impacto negativo en el consumo del Quetzal-1.

### 7.3.2. Análisis del consumo de potencia del módulo *COMMS*

El comportamiento del módulo *COMMS*, se puede observar que en la Figura 41 presenta un asegunada tendencia en el lado derecho del histograma. Esto se debe a que comportamiento del módulo de comunicaciones presenta dos estados, el de recepción de información (*RX*) y el de transferencia de información (*TX-RX*). Como se puede observar en la Figura 42, se utilizaron los datos menores 0.35 watts en representación a la modalidad *RX*.

Dado que, la modalidad *TX-RX* transmite información cada 10 segundos, la modalidad *RX* es la que permanece en operación un intervalo mayor de tiempo. Por lo tanto, presenta una mayor frecuencia de datos alrededor de la modalidad *RX*. Como se puede observar, en la Figura 44 en los valores máximos y mínimos de la distribución de los datos teóricos. Por lo tanto, consumo esperado real del módulo *COMMS* en modalidad *RX* es de 0.296 watts.

El histograma de la modalidad *TX-RX* se muestra en la Figura 43, del cual se obtuvo un consumo de potencia esperado de 0.38 watts. Como se puede observar, en el Cuadro 32 el porcentaje de error, entre el consumo en modalidad *RX*, es de 0.24 %. Por lo tanto, no existe diferencia significativa entre el consumo real y el teórico. Con respecto a los datos de *TX-RX*, se obtuvo un porcentaje de error del 61.54 %, el cual se debe a que el módulo de comunicaciones el valor máximo de consumo de potencia es de 0.45 watts. Tomando en cuenta que el valor esperado durante las simulaciones fue de 0.988 watts.

### 7.3.3. Análisis del consumo de potencia del módulo *Payload*

Como se puede observar en el Cuadro 33 el porcentaje de error es de 62.14 %. Esto se debe a que existe una diferencia de 1.32 watts respecto a los resultados teóricos y experimentales. Cabe mencionar que no hay forma de determinar la diferencia entre el consumo del carrusel y la cámara, para así poder comparar los resultados con la estimación teórica.

Se debe principalmente a que no hay forma de clasificar los datos en busca de esa información. Sin embargo, no tiene un impacto negativo en el análisis. Ya que el carrusel rota 90 grados en 200 ms, como la recolección de los datos no fue de forma continua conforme al tiempo, no se puede realizar una suposición sobre qué datos del consumo corresponden al carrusel y la cámara. Tomando en cuenta que, el consumo real máximo registrado es significativamente bajo respecto a los resultados teóricos.

### 7.3.4. Análisis del consumo de potencia del primer y segundo escenario

Como se puede observar en la Figura 46, existe una alta frecuencia de aproximadamente 6000 datos registrados alrededor del valor de la mediana, que corresponde a un consumo de potencia de 0.619 watts. Como se mencionó anteriormente en la sección del consumo de potencia bajo los efectos de eclipse, el resultado del valor esperado del análisis es nuevamente de 0.619 watts. Esto se debe a que, la frecuencia de los datos del consumo del módulo *Payload* no representan un cambio significativo ya que existen 456 registros del consumo de potencia del módulo *Payload*.

En el caso del consumo de potencia esperado en las simulaciones de los escenarios, de las

Figuras 47 y 48, se obtuvo un consumo de potencia esperado de 0.444 watts. El cual presenta una diferencia de 0.175 watts menos respecto al valor esperado real. Con respecto al valor máximo real de consumo de potencia alcanzado, se obtuvo 0.706 watts. El cual también presenta una alta frecuencia en la distribución. En comparación con el primer escenario, realizado en las simulaciones durante una órbita, donde valor máximo es de 3.977 watts, proviene por el consumo de las antenas del *ADM*.

Por lo cual, dada la diferencia respecto a ambos valores se descarta la posibilidad de determinar el consumo real del módulo *ADM*. Respecto a la simulación del segundo escenario se esperaban valores máximos de consumo de 1.708 Watts. Se puede observar que, durante el primer y segundo escenario se obtuvo un valor máximo significativamente menor en comparación que en las simulaciones.

### 7.3.5. Análisis del consumo de potencia del tercer escenario

El histograma del consumo de potencia real, de la Figura 49, da como resultado un valor máximo de 1.67 watts y un valor, esperado durante el escenario, de 0.253 watts. Respecto al análisis teórico, de la simulación, se esperaba obtener valores máximos elevados de 3.84 watts, además de un valor esperado de 0.44 watts.

Los resultados se deben a que el consumo pico, por el módulo *Payload*, registrado por la computadora corresponde a 0.809 watts. Por lo tanto, al analizar el escenario respecto a cuándo el módulo *Payload* se encontraba activo, existe una diferencia significativa respecto a los resultados teóricos en cuanto al comportamiento del consumo de los módulos.

### 7.3.6. Análisis del estado de carga de las baterías del Quetzal-1

En la Figura 21, que se encuentra en la investigación del marco teórico, se obtuvo la distribución del comportamiento del estado de carga de las baterías. Se puede observar una diferencia significativa en el estado de carga del 100 %, en comparación al resto de datos. En el Cuadro 36 se obtuvo que el porcentaje de estos datos equivale al 71 % de la información dentro de la base de datos. Por lo tanto, se puede decir que el 71 % del tiempo durante 26 semanas en operación, el Quetzal-1 se encontró con un 100 % de carga de energía.

En los resultados de la potencia real del Quetzal-1, en el Cuadro 29, se observa que la generación de potencia por las baterías representa el 73.85 % y el consumo de potencia un 26.14 % de la muestra analizada de los datos. La diferencia entre la generación de potencia con el consumo implica que el Quetzal-1 tuvo una tasa de carga positiva, que respecto con los análisis teóricos de los escenarios también se obtuvo una tasa de carga positiva en los tres escenarios analizados.

Ambos resultados tienen relación con el estado de carga de las baterías ya que el Quetzal-1 estuvo generando más potencia que la potencia que consume. Por lo tanto, a medida que avanzó el tiempo, al llegar al 100 % solo podía perder energía si el consumo de los módulos fuera mayor que lo que generaban los paneles solares o si se encontraba bajo eclipse. Los resultados del estado de carga de las baterías del Quetzal-1 mostraron una tasa de carga positiva por lo tanto, alcanzaría un estado de carga del 100 %. Se puede decir que, el

presupuesto de potencia para el Quetzal-1 se simuló correctamente.

## 7.4. Desarrollo de un presupuesto de potencia para un *CubeSat 2U*

### 7.4.1. Simulación de generación de potencia total por los paneles solares para un *CubeSat 2U*

En el Cuadro 37 se puede observar que el valor de la media, de la generación de potencia de los paneles solares, es de 1.32. Sin embargo, en la Figura 51, el valor de la media se encuentra en una región la cual no describe un valor esperado para la distribución. Para obtener el valor esperado se modeló como una distribución uniforme la sección de mayor frecuencia dentro de la distribución.

Utilizando el rango de la mediana de 1.88 watts y el tercer cuartil de 2.30 watts. Dado que, la frecuencia alrededor de ambos valores supera los 2000 datos de frecuencia. De tal forma que, el valor esperado de la generación de potencia por los paneles solares es de 2.09 watts. Con un valor máximo de 2.74 watts.

### 7.4.2. Análisis del primer escenario del presupuesto de potencia para un *CubeSat 2U*

Como se puede observar en el Cuadro 39, para el primer escenario se obtuvo un consumo de potencia esperado por parte los módulos de 0.43 watts. Dado que en este escenario se activan las resistencias del módulo *ADM* se obtuvieron valores máximos de 3.91 watts. Sin embargo, dado que la frecuencia de los datos máximos por el *ADM* es significativamente inferior al resto de información, se esperaría alcanzar valores máximos 0.95 watts.

En las Figuras 54 Y 55 se puede observar el comportamiento de la generación de potencia por las baterías durante el primer escenario. El valor de generación de potencia esperado corresponde al valor de la mediana, el cual es de 1.67 watts. Como se puede observar en gráfico *boxplot* se observa el sesgo hacia el lado derecho, donde se obtiene que el 50 % de los datos se encuentran agrupados entre 1.4 y 1.97 watts. Además, el valor de generación de potencia máximo es de 2.31 watts.

Los resultados son los esperados ya que, la tendencia de la generación de potencia de los paneles solares es de una generación de potencia superior a los 2 watts. Por lo tanto, al obtener un valor esperado de consumo de potencia por el *CubeSat 2U* de 0.43 watts, se obtiene como resultado una tasa de carga de energía, por órbita, de 3.17 % y una tasa total de 50.84 %. En la figura 56 se puede observar el comportamiento del estado de carga de las baterías, el cual alcanza un estado de carga del 100 % durante la simulación.

### 7.4.3. Análisis del segundo escenario del presupuesto de potencia para un *CubeSat 2U*

Como se puede observar en el Cuadro 42, para el segundo escenario se obtuvo, nuevamente, un consumo de potencia esperado por parte los módulos de 0.43 watts. En este escenario es donde cambia la operación del módulo *COMMS* a *TX-RX*. Por lo tanto, obtiene un consumo máximo de 1.02 watts. Dado que en este escenario se consideró activar nuevamente el ciclo del sistema de despliegue de las antenas del *ADM*, es que se cuenta con registros de un consumo de potencia de 3.99 watts.

En las Figuras 59 Y 60 se puede observar el comportamiento de la generación de potencia por las baterías durante el segundo escenario. Para este análisis se obtuvo nuevamente un valor esperado de generación de potencia por las baterías de 1.67 watts. Como se puede observar en gráfico *boxplot* se observa nuevamente, el sesgo hacia el lado derecho. Además, el valor de generación de potencia máximo es de 2.311 watts.

En la Figura 61 se puede observar el comportamiento del estado de carga de las baterías, en el cual se alcanza nuevamente un estado de carga del 100 % durante 16 órbitas. Además, se obtuvo una tasa de carga de energía por órbita de 3.18 % y una tasa total del 50.81 %. Cabe mencionar, debido a la transición de efectos de luz a eclipse, se pierde un 2.19 % del porcentaje de carga de las baterías.

### 7.4.4. Análisis del tercer escenario del presupuesto de potencia para un *CubeSat 2U*

En la Figura 62 se puede observar el consumo máximo de 1.83 watts, debido a la toma de fotografías cada 600 segundos. Dada la distribución, de la Figura 63, se obtiene un valor esperado de 0.433 watts. En las Figuras 64 y 65 se puede observar, la distribución de generación de potencia por las baterías durante un día. La distribución cuenta con un sesgo hacia la derecha, representado por los valores del primer cuartil de 1.40 watts y tercer cuartil de 1.98 watts. En el Cuadro 46, se puede observar que el valor de generación de potencia esperado es de 1.672 watts, con un valor máximo de 3.31 watts.

Como se puede observar el uso continuo del módulo *Payload*, de 10 veces por órbita, no afecta significativamente la generación de potencia por las baterías. Como se puede observar en el Cuadro 47, se obtiene una tasa de carga positiva por órbita del 2.05 % y 32.81 % para una tasa de carga total durante un día. De tal forma que durante la simulación se alcanza el 100 % de la carga de las baterías. Además, el porcentaje máximo de pérdida de energía es de 2.2 % debido al uso continuo de los módulos durante el tiempo de sombra.

### 7.4.5. Análisis del cuarto escenario del presupuesto de potencia para un *CubeSat 2U*

En la Figura 67 se puede observar el consumo máximo de 3.99 watts, debido al despliegue de las dos antenas del módulo *Deorbit System*. Dada la distribución, de la Figura 68, se obtiene un valor esperado de 0.433 watts. Teniendo en cuenta que el consumo máximo de

1.02 watts, antes del despliegue de las antenas, se debe por la transmisión de información cada de 10 segundos por el módulo *COMMS*.

En las Figuras 69 y 70 se puede observar, la distribución de la generación de potencia por las baterías durante un día, la cual cuenta con un sesgo hacia la derecha, representado por los valores del primer cuartil de 1.405 watts y tercer cuartil de 1.987 watts. En el Cuadro 49, se puede observar que el valor de generación de potencia esperado es de 1.672 watts, con un valor máximo de 3.31 watts.

Como se puede observar el uso continuo del módulo *Payload*, de 10 veces por órbita, no afecta significativamente la generación de potencia por las baterías. En el Cuadro 50 se puede observar que se obtiene una tasa de carga positiva por órbita del 2.05 % y 32.81 % para una tasa de carga total durante un día. De tal forma que durante la simulación se alcanza el 100 % de la carga de las baterías. Además, el porcentaje máximo de pérdida de energía es de 2.86 % debido al uso continuo de los módulos durante el tiempo de sombra.



1. La generación de potencia real, bajo efectos de luz solar, del Quetzal-1 representa el 73.85 % del tiempo durante las 26 semanas de recolección de datos del estado del Quetzal-1, con un valor promedio de  $0.86 \pm 0.0045$  watts.
2. El consumo de potencia en conjunto de los módulos *COMMS*, *ADCS*, *EPS* y *OBC* es de 0.619 watts, siendo este el consumo de potencia esperado del Quetzal-1 bajo efectos de eclipse.
3. Los resultados de las distribuciones de consumo de potencia bajo eclipse no tienen un impacto negativo en el estado de carga de las baterías.
4. Los registros del cambio de temperatura de las baterías se encuentran dentro del rango teórico de 0 °C a 25 °C, esto minimiza la tasa de descarga de las baterías.
5. El análisis de correlación entre la temperatura y el estado de carga de las baterías muestra que la temperatura no tuvo una relación lineal con la pérdida de energía del Quetzal-1.
6. El consumo de potencia del módulo *ADCS* no tuvo un impacto significativo en la distribución del consumo de potencia del Quetzal-1.
7. De los módulos del Quetzal-1, *COMMS*, *ADCS* y *Payload* se obtuvo que los porcentajes de error de los consumos de potencia pico fueron superiores al 61 %.
8. La generación de potencia de los paneles solares fue mayor que el consumo esperado del Quetzal-1 por lo tanto, el estado de carga de las baterías permaneció en 100 % durante el 71 % de la misión.
9. El Quetzal-1 contó con una correcta implementación del presupuesto de potencia, capaz de demostrar que el estado de carga de las baterías se encontraría aumentando en cada escenario.
10. Para el *CubeSat 2U* se espera que los paneles solares generen una potencia de 2.09 watts.

11. Para el *CubeSat 2U* se espera que el consumo de potencia total de los módulos *COMMS*, *ADCS*, *EPS* y *OBC*; sea de 0.43 watts.
12. El consumo de potencia de 2.96 watts de los módulos *ADM* y *Deorbit System* no tienen un impacto significativo en la pérdida de energía del *CubeSat 2U*.
13. El uso del módulo *Payload* cada 600 segundos no tiene un impacto significativo en el estado de carga de las baterías.
14. Para el *CubeSat 2U* la pérdida máxima de carga de las baterías es de 2.86 %.
15. Para un *CubeSat 2U* se cuenta con la capacidad de generación de potencia para implementar nuevos componentes que sean utilizados en nuevos estudios.

---

## Recomendaciones

---

En el desarrollo de un próximo *CubeSat*, debería recolectarse la información del consumo de potencia real de los módulos *ADCS*, *Payload*, *ADM*, *COMMS*, *EPS*, *OBC*, *Deorbit System*; para poder obtener un análisis con mayor precisión del consumo total del *CubeSat*, debido a que el consumo de potencia de los módulos puede presentar una diferencia significativa respecto a los cálculos teóricos. Además, se podría analizar los resultados de la generación de potencia de los paneles solares, si se cuenta con registros de todos los módulos.

Para futuros análisis se recomienda la implementación de herramientas estadísticas de datos. Calcular únicamente un valor promedio no brinda suficiente información. Como se ha mencionado anteriormente, el valor de la media no siempre describe la distribución analizada. Por lo tanto, para futuros análisis podrían utilizarse las herramientas de visualización, tales como el histograma y diagrama *boxplot*, los cuales brindan mejor visualización del comportamiento de los datos, que un diagrama de líneas. El diagrama de líneas es útil para visualizar picos máximos y mínimos que presentan baja frecuencia. Sin embargo, pueden representar información que no describe el comportamiento real de los datos.



- Bester, J., Groenewald, B. B. & Wilkinson, R. H. (2012). Electrical power system for a 3U CubeSat nanosatellite incorporating peak power tracking with dual redundant control.
- Cheng, X.-B., Zhang, R., Zhao, C.-Z. & Zhang, Q. (2017). Toward Safe Lithium Metal Anode in Rechargeable Batteries: A Review. *Chemical Reviews*, 117(15), 10403-10473. <https://doi.org/10.1021/acs.chemrev.7b00115>
- Chuan Tan, Y., Chandrasekara, R., Cheng, C. & Ling, A. (2013). Silicon avalanche photodiode operation and lifetime analysis for small satellites.
- Duann, Y., Chang, L. C., Chao, C.-K., Chiu, Y.-C., Tsai-Lin, R., Tai, T.-Y., Luo, W.-H., Liao, C.-T., Liu, H.-T., Chung, C.-J., Duann, R., Kuo, C.-L., Liu, J.-Y., Yang, Z.-M., Gacal, G. F., Chandran, A., Priyadarshan, H., Verma, A., Fang, T.-W. & Srivastava, S. (2020). IDEASSat: A 3U CubeSat mission for ionospheric science. *Advances in Space Research*, 66(1), 116-134. <https://doi.org/10.1016/j.asr.2020.01.012>
- Garcia, D. (2017). *CubeSat Fase 4 Modulo; Potencia* (Megaproyecto). Universidad del Valle de Guatemala. Guatemala.
- Kulu, E. (s.f.). What is a CubeSat? Consultado el 28 de noviembre de 2020, desde <https://www.nanosats.eu/cubesat.html>
- Lee, M. F. (2019). *CubeSat Phase: 5 Module: Solar Panels* (Megaproject). Universidad del Valle de Guatemala. Guatemala.
- LIMITED, D. P. T. (2018). Polymer Li-ion Rechargeable Battery [Model:DTP605068].
- Popescu, O. (2017). Power Budgets for CubeSat Radios to Support Ground Communications and Inter-Satellite Links. *IEEE Access*, 5, 12618-12625. <https://doi.org/10.1109/ACCESS.2017.2721948>
- Sanchez-Sanjuan, S., Gonzalez-Llorente, J. & Hurtado-Velasco, R. (2016). Comparison of the Incident Solar Energy and Battery Storage in a 3U CubeSat Satellite for Different Orientation Scenarios. *Journal of Aerospace Technology and Management*, 8(1), 91-102. Consultado el 28 de noviembre de 2020, desde <http://www.jatm.com.br/ojs/index.php/jatm/article/view/531>

University, B. (s.f.). What causes Lithium-ion to die? – Battery University. Consultado el 31 de mayo de 2021, desde [https://batteryuniversity.com/learn/article/what\\_causes\\_lithium\\_ion\\_to\\_die](https://batteryuniversity.com/learn/article/what_causes_lithium_ion_to_die)

Velásquez, A., Martínez, D., Gómez, H. & Ramírez, M. (2019). *CuebSat Fase 6* (Megaproyecto). Universidad del Valle de Guatemala. Guatemala.

# CAPÍTULO 11

---

Anexos

---

### **ANEXO 1: Ubicación de los códigos de Python utilizados en este trabajo**

<b>No. Anexo</b>	<b>Drive</b>	<b>Designación</b>	<b>Procedimiento</b>	<b>Ubicación</b>
2	3U CubeSat	TAB-PWR- 082	Análisis de Potencia Real	3U CubeSat / 5. POWER / 3. SPREADSHEETS
3	3U CubeSat	TAB-PWR- 083	Análisis de Efectos de Temperatura	
4	3U CubeSat	TAB-PWR-084	Validación del Presupuesto de Potencia del Quetzal-1	
5	3U CubeSat	TAB-PWR- 085	Presupuesto de Potencia de un CubeSat 2U	

## Anexo 2. Código en Python para Análisis de Potencia Real

October 28, 2021

```
[1]: import pandas as pd
      from google.colab import files
      uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving Power\_ten\_weeks.xls to Power\_ten\_weeks.xls

```
[2]: import io
      import numpy as np
      import seaborn as sns
      import matplotlib.pyplot as plt
      from matplotlib import rcParams
      rcParams['figure.figsize'] = 10,6
      df = pd.read_excel(io.BytesIO(uploaded['Power_ten_weeks.xls']))
```

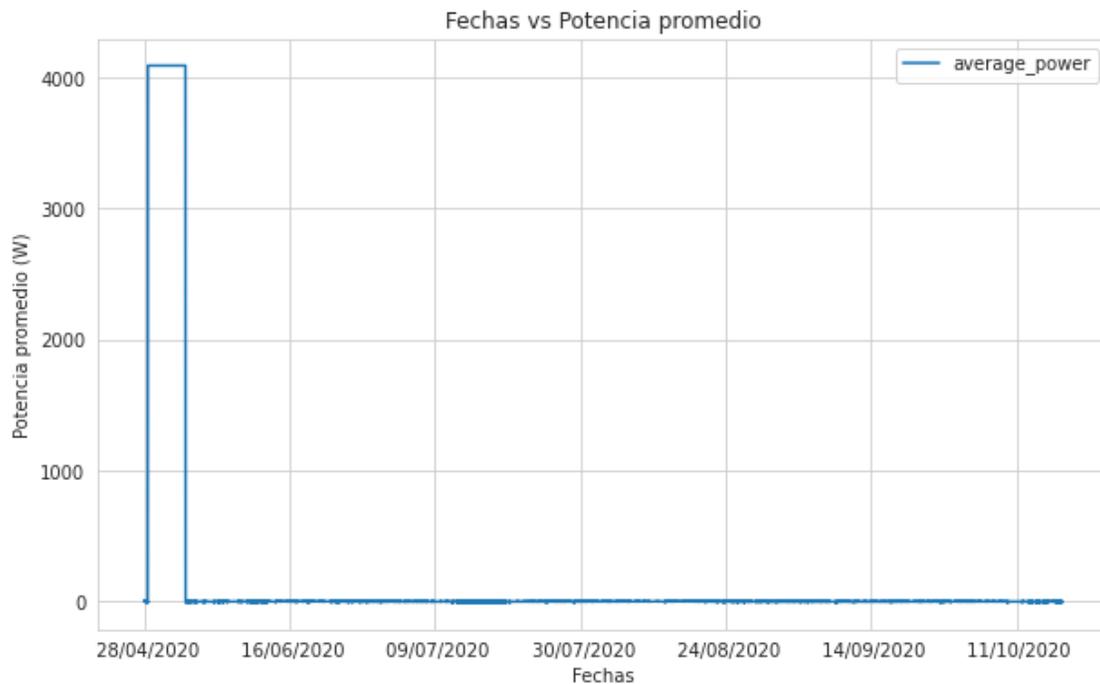
```
[3]: df.dtypes
```

```
[3]: Date                object
      average_power      float64
      battery_voltage    float64
      average_current    float64
      EPS_temperature    float64
      remaining_capacity  float64
      state_charge       int64
      state_health       int64
      transmission_flags_EPS  int64
      HEATER_status      int64
      ADC1_CH1           float64
      ADC1_CH2           float64
      ADC1_CH3           float64
      ADC1_CH4           float64
      ADC1_CH5           float64
      ADC1_CH6           float64
      ADC2_CH1           float64
      ADC2_CH2           float64
      ADC2_CH3           float64
      ADC2_CH4           float64
```

```
ADC2_CH5          float64
ADC2_CH6          float64
dtype: object
```

## 1 Raw Data

```
[5]: sns.set_style('whitegrid')
df.plot("Date", "average_power")
plt.title("Fechas vs Potencia promedio")
plt.xlabel('Fechas')
plt.ylabel('Potencia promedio (W)')
plt.legend()
#plt.figure(figsize=(20,10))
plt.show()
```



Photodiodes list

```
[4]: List =_
↳ ['ADC1_CH1', 'ADC1_CH2', 'ADC1_CH3', 'ADC1_CH4', 'ADC1_CH5', 'ADC1_CH6', 'ADC2_CH1',_
↳ #List of photodiodes voltage variables
    'ADC2_CH2', 'ADC2_CH3', 'ADC2_CH4', 'ADC2_CH5', 'ADC2_CH6']
#df.loc[(df[List] == 0).all(axis=1)]
```

```
[ ]: # df.loc[df['average_power'] == 4093]
# df.loc[df['battery_voltage'] == 3.6]
# df.loc[df['average_power'] < 0]
#df.loc[df['average_power'] == 0]
#df.loc[((df[List] == 0).all(axis=1))& (df['average_power'] > 10)]
#df.loc[((df[List] == 0).all(axis=1))& (df['average_power'] < 0)]
# df.loc[((df[List] == 0).all(axis=1))& (df['average_power'] == 0)]
```

## 2 Data Cleaning

```
[6]: #Datos de consumo y generacion en la misma graf
df0 = df[df["average_power"] != 4093]
df0 = df0[df0['transmission_flags_EPS'] != 0]
df0 = df0[df0['EPS_temperature'] != -25]
df0 = df0[df0['battery_voltage'] != 3.6]
```

## 3 Power generated when is under sunlight

```
[9]: df1 = df0[(df0[List] != 0).all(axis=1)]
df1 = df1[df1.average_power > 0] # The negative values are when is consuming,
↳power, and this analysis is for power generation.
df1.average_power.describe()
```

```
[9]: count      28931.000000
mean          0.864527
std           0.389731
min           0.020770
25%           0.602390
50%           0.864130
75%           1.100930
max           2.787630
Name: average_power, dtype: float64
```

```
[11]: minimum = df1.average_power.min()
maximum = df1.average_power.max()
print("The minimum value of power is = " + str(round(minimum,3)) + " Watts")
print("The maximum value of power is = " + str(round(maximum,2)) + " Watts")
x_bar = df1.average_power.mean()
print("The actual power consumed is = " + str(round(x_bar,2)) + " Watts")
sigma = df1.average_power.std()
print("The standard deviation is = " + str(round(sigma,2)))
n = len(df1)
coef = 0.95
alpha = 1. - coef
print(n)
```

```

import scipy.stats as stats
z_critical = stats.norm.ppf(q = 0.975)
zinterval = stats.norm.interval(alpha=coef)
zinterval
#Standard error
standard_error = sigma / np.sqrt(n)

CI_lower = x_bar - z_critical * standard_error
CI_upper = x_bar + z_critical * standard_error
pluslow = round(z_critical * standard_error,4)
print(pluslow)
round(CI_lower,3),round(CI_upper,3)

```

The minimum value of power is = 0.021 Watts  
 The maximum value of power is = 2.79 Watts  
 The actual power consumed is = 0.86 Watts  
 The standard deviation is = 0.39  
 28931  
 0.0045

[11]: (0.86, 0.869)

```

[ ]: #Esto se justifica porque tenemos una distribucion normal (Gaussiana)
int_sup = x_bar + sigma #Intervalo superior
int_inf = x_bar - sigma #Intervalo inferior
rango_conf = int_sup - int_inf #Rango de 69% de los datos
rango_conf

```

[ ]: 0.7794620378125323

```

[ ]: df.loc[df['HEATER_status'] == 102] # verifying manual inputs for the heater
df.loc[df['HEATER_status'] == 102]

```

[ ]: Empty DataFrame  
 Columns: [Date, average\_power, battery\_voltage, average\_current, EPS\_temperature, remaining\_capacity, state\_charge, state\_health, transmission\_flags\_EPS, HEATER\_status, ADC1\_CH1, ADC1\_CH2, ADC1\_CH3, ADC1\_CH4, ADC1\_CH5, ADC1\_CH6, ADC2\_CH1, ADC2\_CH2, ADC2\_CH3, ADC2\_CH4, ADC2\_CH5, ADC2\_CH6]  
 Index: []

## 4 Power generated when is under sunlight, Heater = ON

```

[12]: df4 = df1[df1['HEATER_status'] == 153]
df4['average_power'].describe()

```

[12]: count 5787.000000  
 mean 0.722497

```

std          0.292333
min          0.020770
25%         0.535920
50%         0.751960
75%         0.930600
max          1.832110
Name: average_power, dtype: float64

```

```

[ ]: minimum4 = df4.average_power.min()
maximum4 = df4.average_power.max()
x_bar4 = df4.average_power.mean()
sigma4 = df4.average_power.std()
n4 = len(df4)
coef = 0.95
alpha = 1. - coef
z_critical = stats.norm.ppf(q = 0.975)
zinterval = stats.norm.interval(alpha=coef)

standard_error = sigma4 / np.sqrt(n4)
standard_error

CI_lower4 = x_bar4 - z_critical * standard_error
CI_upper4 = x_bar4 + z_critical * standard_error
pluslow = round(z_critical * standard_error,4)
print(pluslow)
round(CI_lower4,3),round(CI_upper4,3)

```

0.0075

[ ]: (0.715, 0.73)

## 5 Power generated when is under sunlight, Heater = OFF

```

[13]: df5 = df1[df1['HEATER_status'] == 150]
df5['average_power'].describe()

```

```

[13]: count      23121.000000
mean          0.900210
std           0.402868
min           0.020770
25%          0.627320
50%          0.905670
75%          1.150780
max           2.787630
Name: average_power, dtype: float64

```

```
[ ]: minimum5 = df5.average_power.min()
maximum5 = df5.average_power.max()
x_bar5 = df5.average_power.mean()
sigma5 = df5.average_power.std()
n5 = len(df5)
coef = 0.95
alpha = 1. - coef
z_critical = stats.norm.ppf(q = 0.975)
zinterval = stats.norm.interval(alpha=coef)

standard_error = sigma5 / np.sqrt(n5)
standard_error

CI_lower5 = x_bar5 - z_critical * standard_error
CI_upper5 = x_bar5 + z_critical * standard_error
pluslow = round(z_critical * standard_error,4)
print(pluslow)
round(CI_lower5,3),round(CI_upper5,3)
```

0.0052

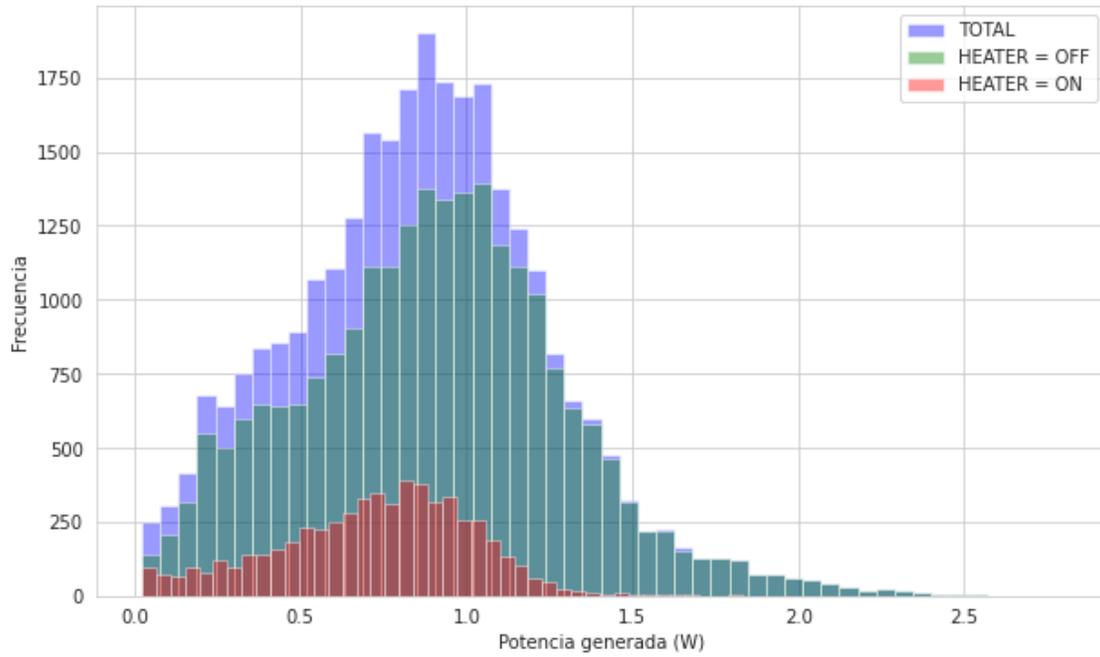
```
[ ]: (0.895, 0.905)
```

## 6 Power Generation Distribution when is under sunlight

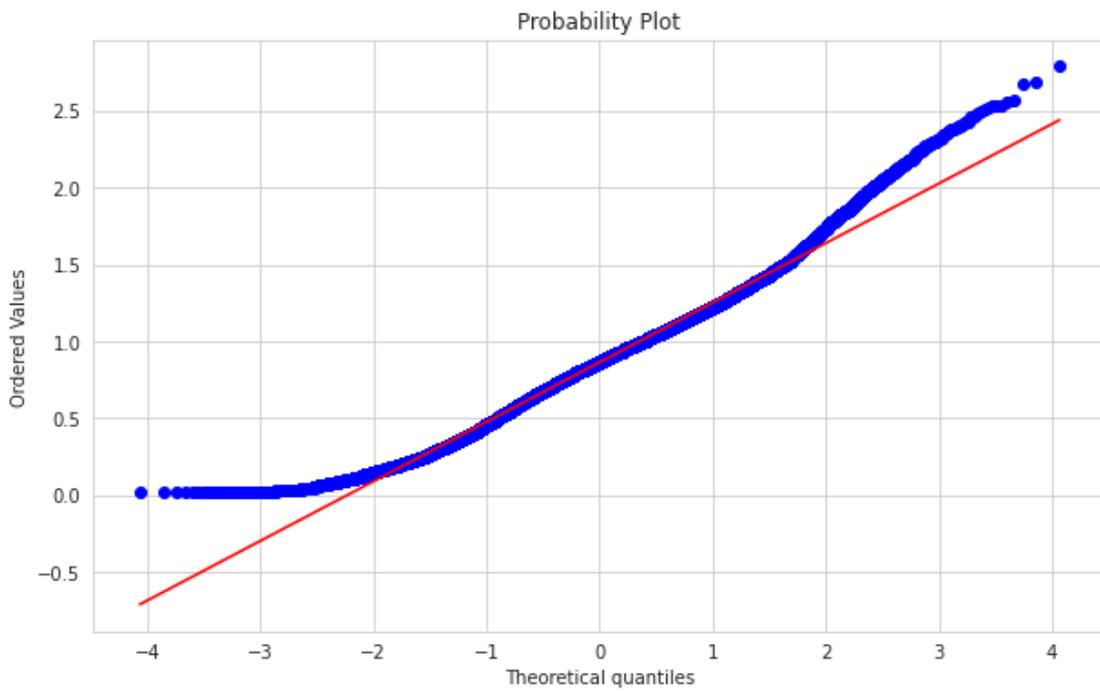
```
[ ]: sns.distplot(a=df1['average_power'], label="TOTAL",color= "blue", kde = False)
sns.distplot(a=df5['average_power'], label="HEATER = OFF",color= "green", kde =_
↳False)
sns.distplot(a=df4['average_power'], label="HEATER = ON",color= "red", kde =_
↳False)
plt.xlabel('Potencia generada (W)')
plt.ylabel('Frecuencia')
plt.legend()
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).

```
warnings.warn(msg, FutureWarning)
```

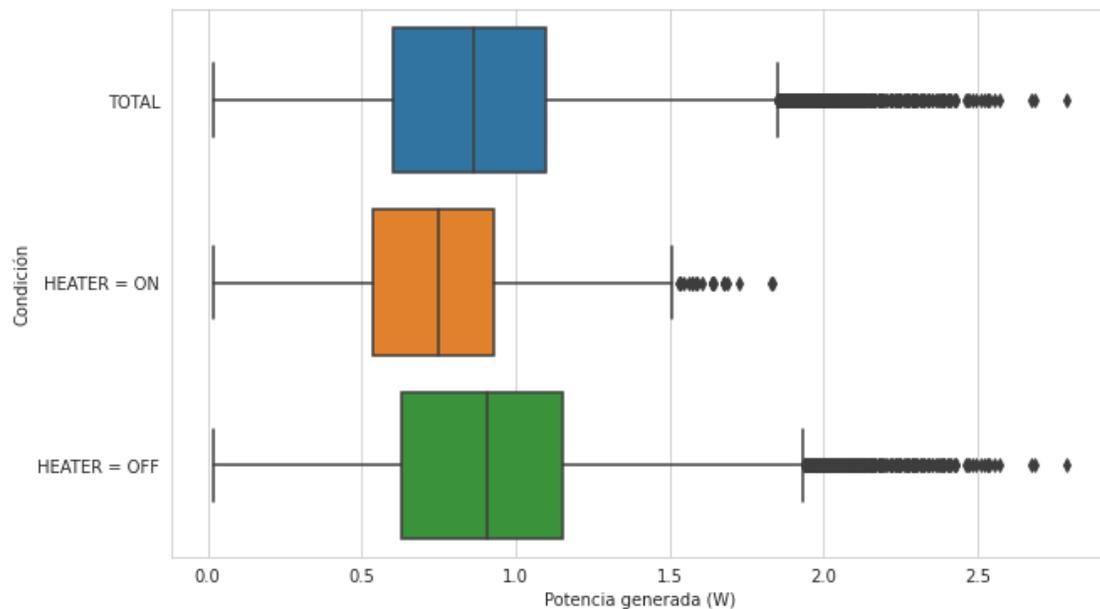


```
[14]: import scipy.stats
scipy.stats.probplot(df1["average_power"], plot = plt)
plt.show()
```



```
[ ]: df1["Condición"] = "TOTAL"
df11 = df1.copy()
df11['Condición'] = np.where(df11["HEATER_status"] == 153, "HEATER = ON",
↪ "HEATER = OFF")
cdf = pd.concat([df1, df11])
sns.boxplot(data=cdf,x='average_power', y='Condición')
plt.xlabel('Potencia generada (W)')
```

```
[ ]: Text(0.5, 0, 'Potencia generada (W)')
```



## 7 Total power consumed Analysis when is under sunlight

```
[15]: df2 = df[df["average_power"] != 4093]
df2 = df2[df2.average_power != 0]
df2 = df2[df2['transmission_flags_EPS'] != 0]
df2 = df2[df2['EPS_temperature'] != -25]
df2 = df2[(df2[List] != 0).all(axis=1)]
df2 = df2[df2.average_power < 0]
df2['average_power'] = df2['average_power'].abs()
df2['average_power'].describe()
```

```
[15]: count    552.000000
mean         0.163619
std          0.119706
```

```
min      0.020770
25%     0.062320
50%     0.137100
75%     0.241998
max      0.540080
Name: average_power, dtype: float64
```

#Total power consumed when is under sunlight, HEATER =ON

```
[16]: df6 = df2[df2['HEATER_status'] == 153]
df6['average_power'].describe()
```

```
[16]: count      363.000000
mean        0.180930
std         0.123683
min         0.020770
25%         0.078930
50%         0.153710
75%         0.259655
max         0.540080
Name: average_power, dtype: float64
```

#Total power consumed when is under sunlight, HEATER =OFF

```
[17]: df7 = df2[df2['HEATER_status'] == 150]
df7['average_power'].describe()
```

```
[17]: count      188.000000
mean        0.129539
std         0.103823
min         0.024930
25%         0.045700
50%         0.087240
75%         0.182800
max         0.440370
Name: average_power, dtype: float64
```

## 8 Power Consumed Distributions when is under sunlight

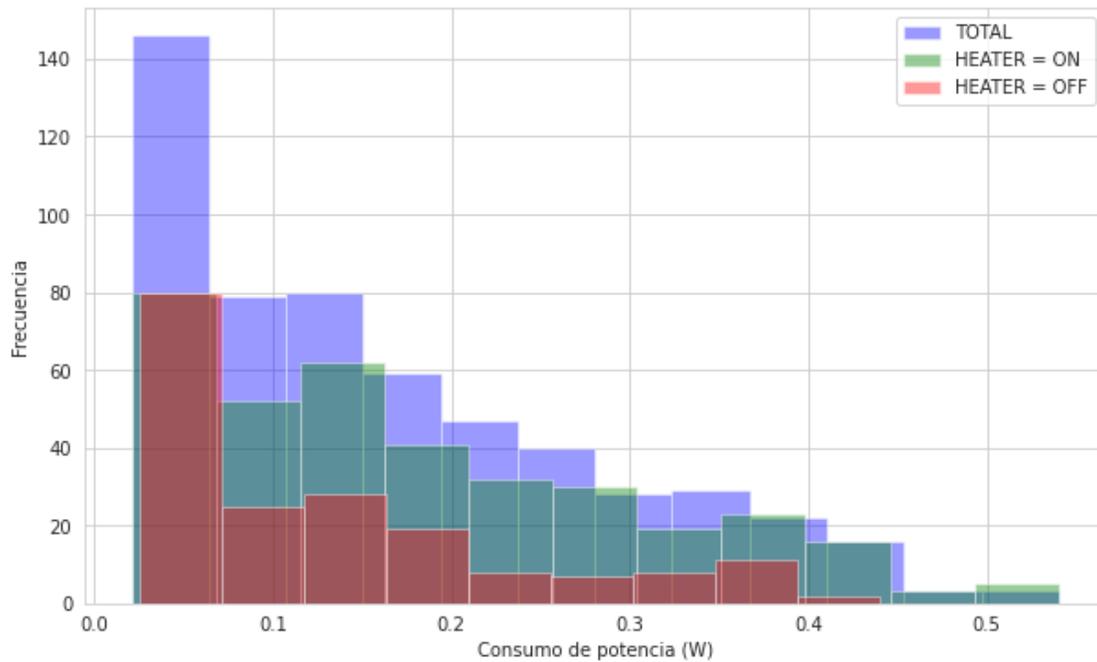
```
[ ]: sns.distplot(a=df2['average_power'], label="TOTAL",color='blue', kde = False)
sns.distplot(a=df6['average_power'], label="HEATER = ON",color='green',kde =_
↳False)
sns.distplot(a=df7['average_power'], label="HEATER = OFF",color = 'red', kde =_
↳False)

plt.xlabel('Consumo de potencia (W)')
plt.ylabel('Frecuencia')
```

```
plt.legend()
plt.show()
```

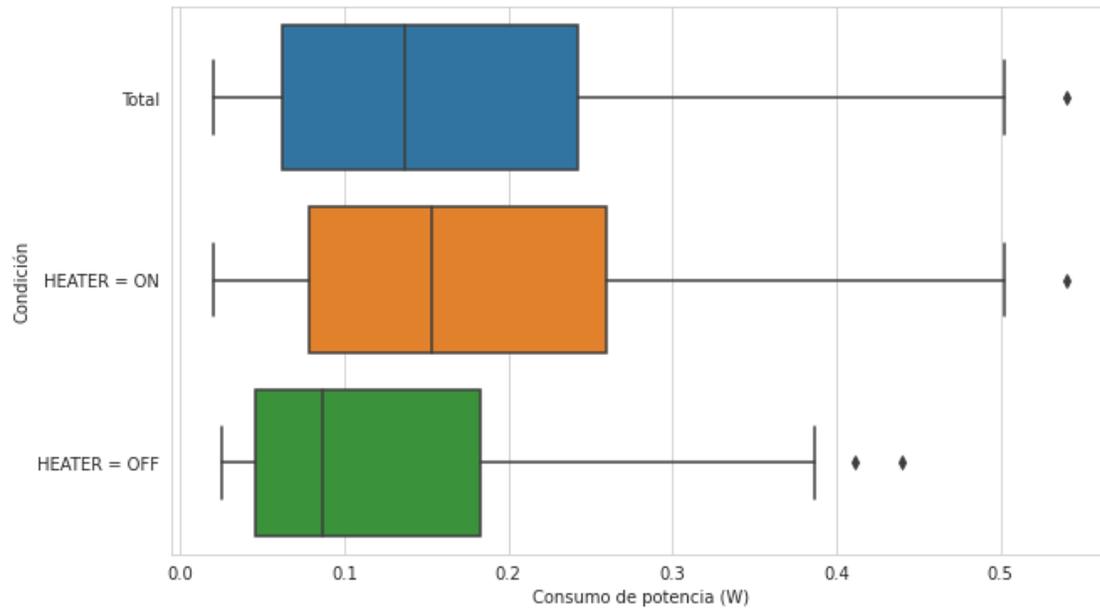
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



```
[ ]: df2["Condición"] = "Total"
df10 = df2.copy()
df10['Condición'] = np.where(df10["HEATER_status"] == 153, "HEATER = ON",
    ↪ "HEATER = OFF")
cdf = pd.concat([df2, df10])
sns.boxplot(data=cdf, x='average_power', y='Condición')
plt.xlabel('Consumo de potencia (W)')
```

```
[ ]: Text(0.5, 0, 'Consumo de potencia (W)')
```



#Power consumed when is in eclipse

```
[18]: df3 = df[df["average_power"] != 4093]
df3 = df3[df3.average_power != 0]
df3 = df3[df3['transmission_flags_EPS'] != 0]
df3 = df3[df3['EPS_temperature'] != -25]
df3 = df3[(df3[List] == 0).all(axis=1)]
df3 = df3[df3.average_power < 0]
df3['average_power'] = df3['average_power'].abs()
df3['average_power'].describe()
```

```
[18]: count    9692.000000
mean         0.752202
std          0.311716
min          0.610700
25%         0.619010
50%         0.619010
75%         0.697950
max          1.736560
Name: average_power, dtype: float64
```

#Total power consumed when is in eclipse, HEATER =ON

```
[19]: df8 = df3[df3['HEATER_status'] == 153]
df8['average_power'].describe()
```

```
[19]: count    1051.000000
      mean     1.632316
      std      0.124039
      min     0.760260
      25%     1.645160
      50%     1.653470
      75%     1.657620
      max     1.736560
      Name: average_power, dtype: float64
```

#Total power consumed when is in eclipse, HEATER =OFF

```
[22]: df9 = df3[df3['HEATER_status'] == 150]
      df9['average_power'].describe()
```

```
[22]: count    8641.000000
      mean     0.645155
      std      0.037831
      min     0.610700
      25%     0.619010
      50%     0.619010
      75%     0.693790
      max     0.851660
      Name: average_power, dtype: float64
```

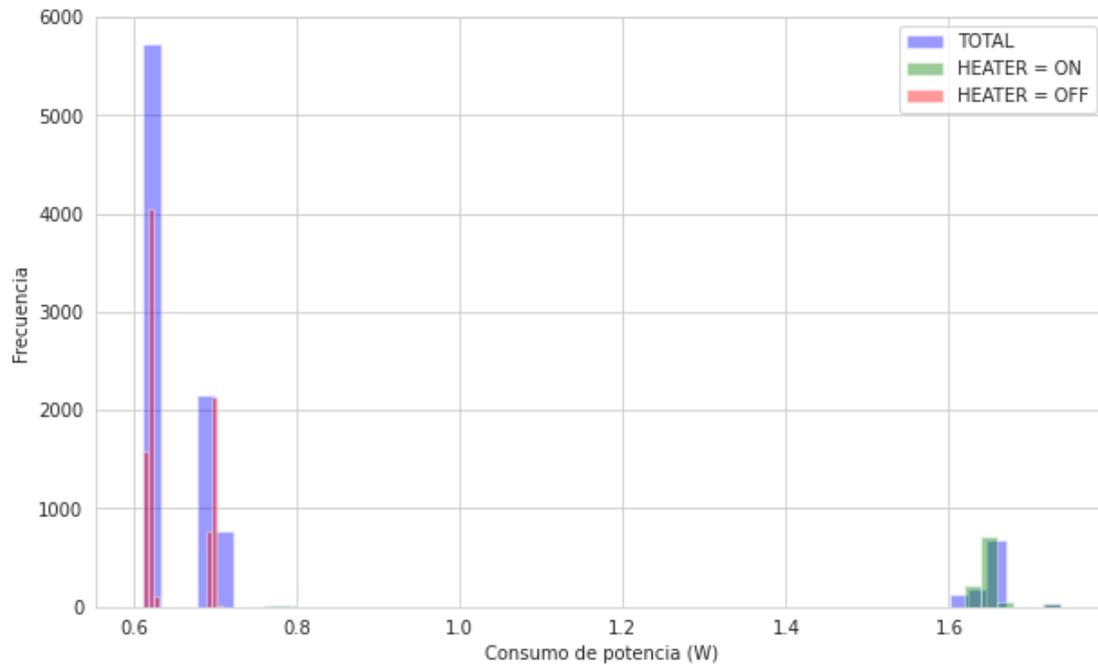
#Power Consumed Distribution when is in Eclipse

```
[ ]: sns.distplot(a=df3['average_power'], label="TOTAL",color='blue', kde = False)
      sns.distplot(a=df8['average_power'], label="HEATER = ON",color='green',kde =_
      ↪False)
      sns.distplot(a=df9['average_power'], label="HEATER = OFF",color = 'red', kde =_
      ↪False)
      plt.xlabel('Consumo de potencia (W)')
      plt.ylabel('Frecuencia')
      plt.legend()
      plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:

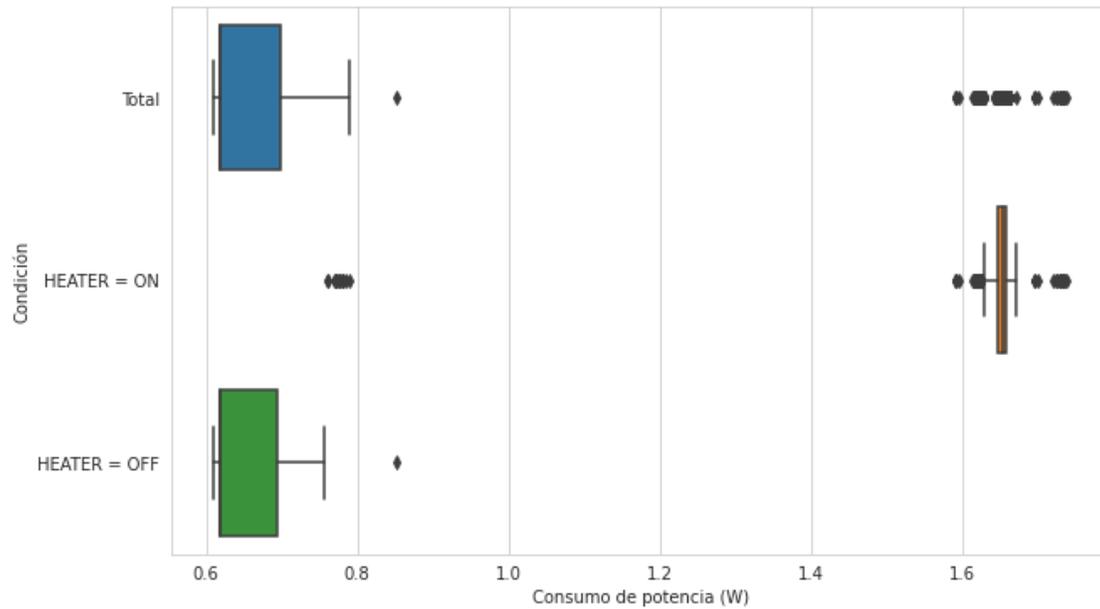
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



```
[ ]: df3["Condición"] = "Total"
df11 = df3.copy()
df11['Condición'] = np.where(df11["HEATER_status"] == 153, "HEATER = ON",
↪ "HEATER = OFF")
cdf = pd.concat([df3, df11])
sns.boxplot(data=cdf,x='average_power', y='Condición')
plt.xlabel('Consumo de potencia (W)')
```

```
[ ]: Text(0.5, 0, 'Consumo de potencia (W)')
```



```
[ ]: # data =
      ↳[['Min', round(minimum,2), round(maximum2,2), round(maximum3,2)], ['Mean', round(x_bar,2), round(
#           ['S.
      ↳Deviation', round(sigma,2), round(sigma2,2), round(sigma3,2)], ['Lower
      ↳Interval', round(CI_lower,2), round(CI_lower2,2), round(CI_lower3,2)], ['Upper
      ↳Interval', round(CI_upper,2), round(CI_upper2,2), round(CI_upper3,2)]]
# dftable = pd.DataFrame(data, columns = [' ', 'Power Generated Under Sunlight
      ↳(W)', 'Power 2', 'Power 3'], index=None)
# dftable
```

## Anexo 3. Código en Python para Análisis de efectos de temperatura

October 28, 2021

```
[ ]: import pandas as pd
      from google.colab import files
      uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving Power\_Budget.xls to Power\_Budget.xls

```
[ ]: import io
      import matplotlib.pyplot as plt
      import numpy as np
      import seaborn as sns
      df = pd.read_excel(io.BytesIO(uploaded['Power_Budget.xls']))
```

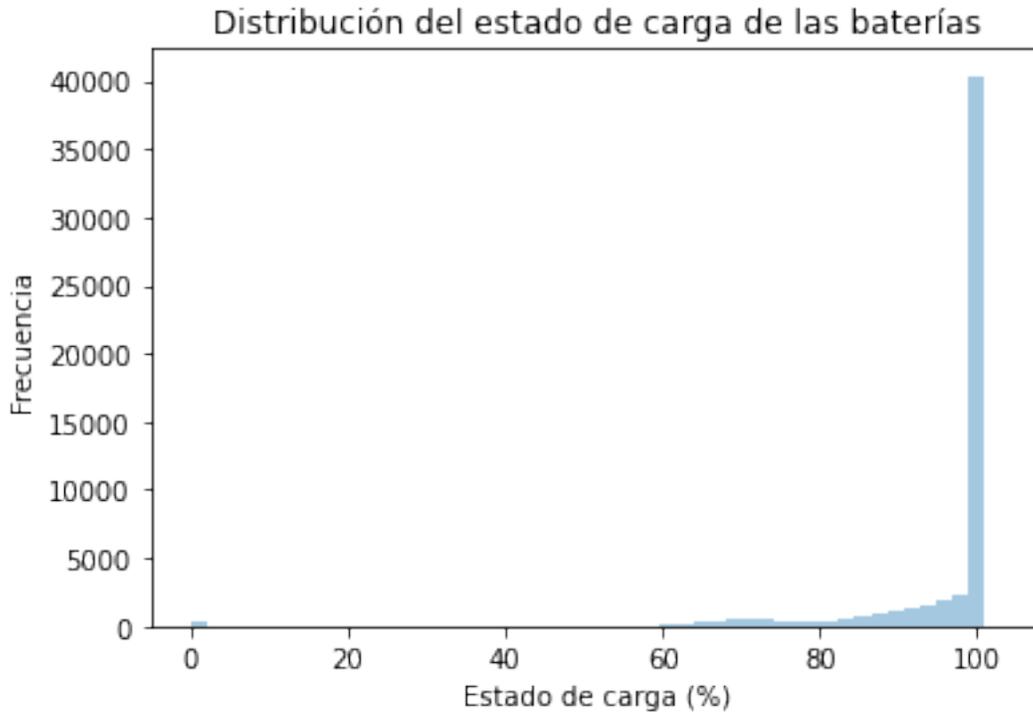
```
[ ]: df0 = df[df["average_power"] != 4093]
      df0 = df0[df0['transmission_flags_EPS'] != 0]
      df0 = df0[df0['EPS_temperature'] != -25]
      df0 = df0[df0['battery_voltage'] != 3.6]
      df0 = df0[df0['EPS_temperature'] >= 0]
      len(df0.loc[df0['HEATER_status'] == 144])
      #df0.loc[df0['HEATER_status'] == 144]
```

```
[ ]: 25
```

```
[ ]: sns.distplot(a=df0['state_charge'], label='state_charge', kde = False);
      plt.title("Distribución del estado de carga de las baterías")
      plt.xlabel('Estado de carga (%)')
      plt.ylabel('Frecuencia')
      plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

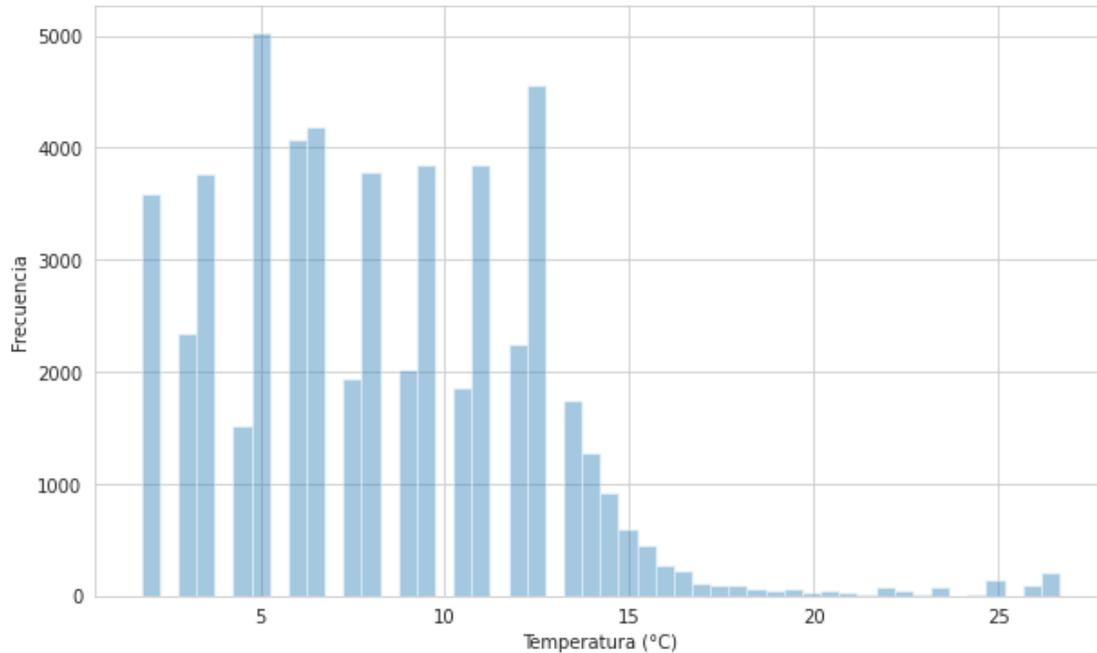
```
warnings.warn(msg, FutureWarning)
```



```
[ ]: from matplotlib import rcParams
rcParams['figure.figsize'] = 10,6
sns.set_style('whitegrid')

sns.distplot(a=df0['EPS_temperature'], label='EPS_temperature', kde = False);
plt.xlabel('Temperatura (°C)')
plt.ylabel('Frecuencia')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

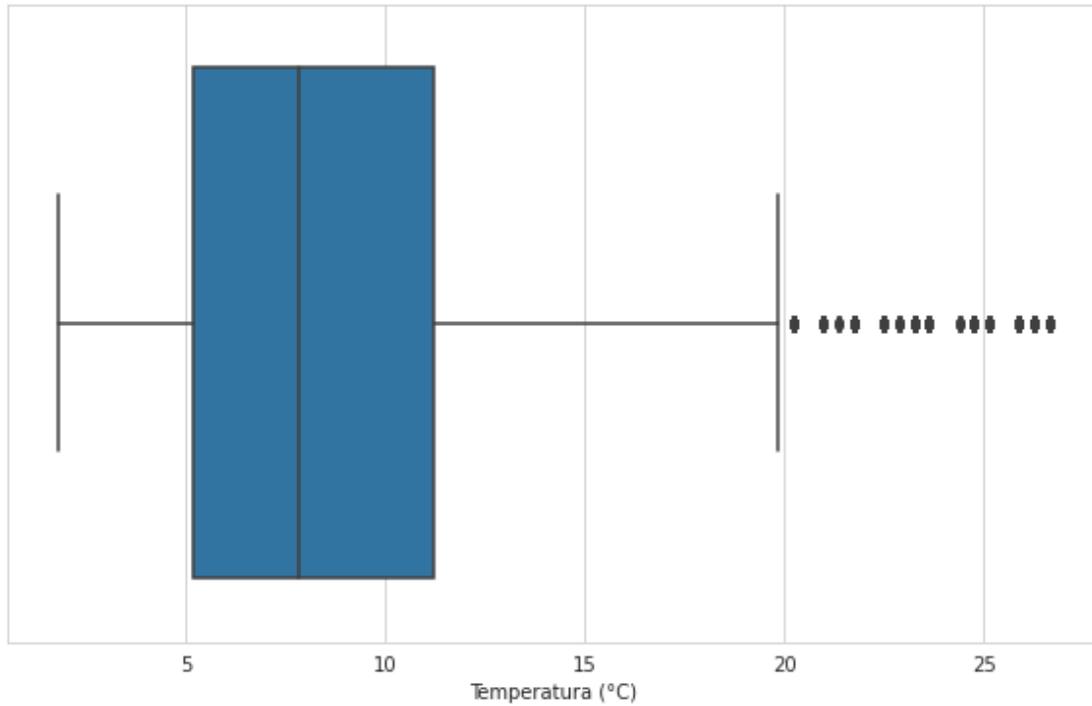


```
[ ]: sns.boxplot("EPS_temperature", data = df0)
plt.xlabel('Temperatura (°C)')
```

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

```
[ ]: Text(0.5, 0, 'Temperatura (°C)')
```



```
[ ]: df0['EPS_temperature'].describe()
```

```
[ ]: count    55313.000000
      mean      8.317385
      std       4.266851
      min       1.770000
      25%       5.160000
      50%       7.800000
      75%      11.190000
      max      26.650000
      Name: EPS_temperature, dtype: float64
```

```
[ ]: from matplotlib import rcParams
      rcParams['figure.figsize'] = 10,6
      sns.set_style('whitegrid')

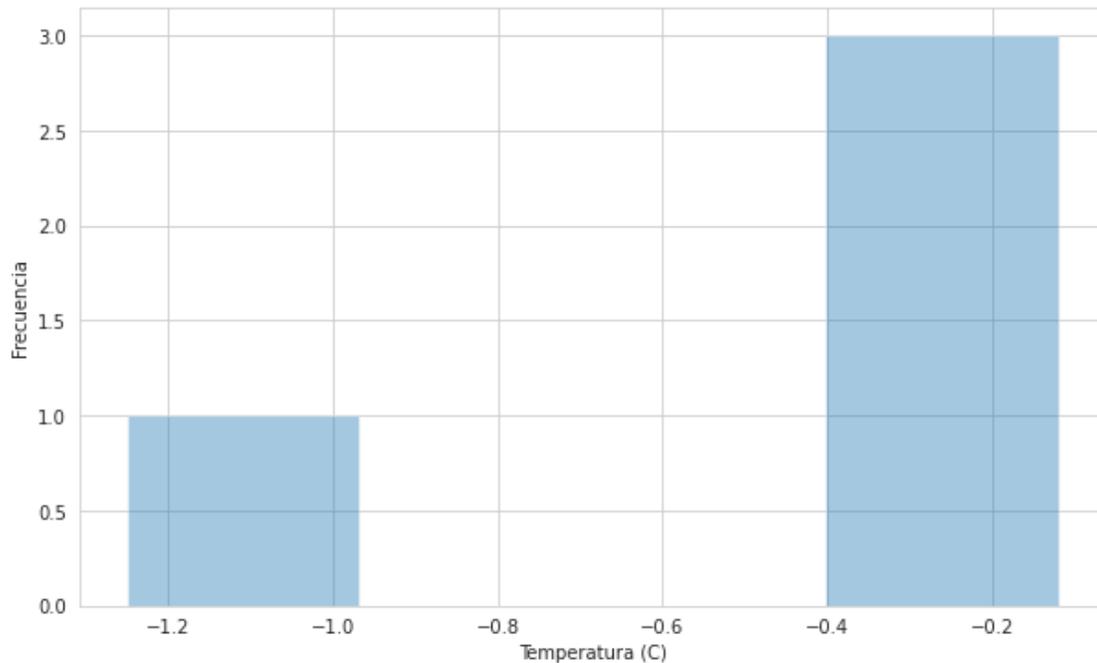
      df1 = df[df["average_power"] != 4093]
      df1 = df1[df1['transmission_flags_EPS'] != 0]
      df1 = df1[df1['EPS_temperature'] != -25]
      df1 = df1[df1['battery_voltage'] != 3.6]
      df1 = df1[df1['EPS_temperature'] < 0]
      sns.distplot(a=df1['EPS_temperature'], label='EPS_temperature', kde = False);
      plt.xlabel('Temperatura (C)')
      plt.ylabel('Frecuencia')
```

```
len(df1)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: 4
```



```
[ ]: df1['EPS_temperature'].describe()
```

```
[ ]: count    4.0000  
     mean    -0.4025  
     std     0.5650  
     min    -1.2500  
     25%    -0.4025  
     50%    -0.1200  
     75%    -0.1200  
     max    -0.1200  
     Name: EPS_temperature, dtype: float64
```

```
#Pearson's Correlation
```

```

[ ]: List =
↳ ['ADC1_CH1', 'ADC1_CH2', 'ADC1_CH3', 'ADC1_CH4', 'ADC1_CH5', 'ADC1_CH6', 'ADC2_CH1',
↳ #List of photodiodes voltage variables
    'ADC2_CH2', 'ADC2_CH3', 'ADC2_CH4', 'ADC2_CH5', 'ADC2_CH6']
df3 = df[df["average_power"] != 4093]
df3 = df3[df3.average_power != 0]
df3 = df3[df3['transmission_flags_EPS'] != 0]
df3 = df3[df3['EPS_temperature'] != -25]
df3 = df3[(df3[List] == 0).all(axis=1)]
df3 = df3[df3.average_power < 0]
df3['average_power'] = df3['average_power'].abs()
# df0.corr() # correlation for all variables
from scipy.stats import pearsonr
corr, _ = pearsonr(df3['EPS_temperature'], df3['state_charge'])
corr

```

```

[ ]: -0.014054539870667565

```

## Anexo 4. Código en Python para Validación del presupuesto de potencia del Quetzal-1

November 2, 2021

```
[ ]: import pandas as pd
      from google.colab import files
      uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving Power\_Budget.xls to Power\_Budget.xls

```
[ ]: import io
      import matplotlib.pyplot as plt
      import numpy as np
      import seaborn as sns
      from matplotlib import rcParams
      rcParams['figure.figsize'] = 10,6
      df = pd.read_excel(io.BytesIO(uploaded['Power_Budget.xls'])) # Excel with 26
      ↪ weeks of Quetzal-1 results
```

Data cleaning

```
[ ]: df0 = df[df["average_power"] != 4093]
      df0 = df0[df0['transmission_flags_EPS'] != 0]
      df0 = df0[df0['EPS_temperature'] != -25]
      df0 = df0[df0['battery_voltage'] != 3.6]
```

```
[ ]: List = []
      ↪ ['ADC1_CH1', 'ADC1_CH2', 'ADC1_CH3', 'ADC1_CH4', 'ADC1_CH5', 'ADC1_CH6', 'ADC2_CH1',
      ↪ #List of photodiodes voltage variables
      'ADC2_CH2', 'ADC2_CH3', 'ADC2_CH4', 'ADC2_CH5', 'ADC2_CH6']
```

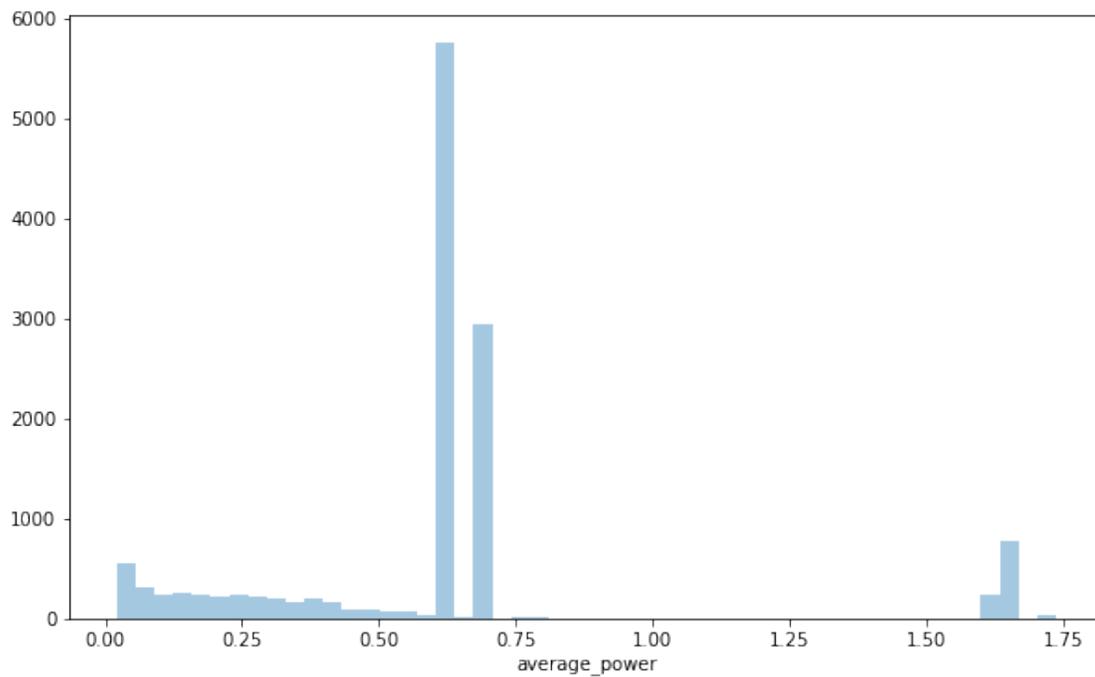
```
[ ]: prueba = df0[df0['PAYLOAD_current'] == 0]
      prueba = prueba[prueba["average_power"] < 0]
      prueba = prueba[prueba["ADM_status"] != 0]
      # prueba = prueba[prueba['HEATER_status'] == 153]
      prueba['average_power'] = prueba['average_power'].abs()

      sns.distplot(a= prueba['average_power'], label="prueba", kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7284d5610>
```



```
[ ]: prueba['average_power'].describe()
```

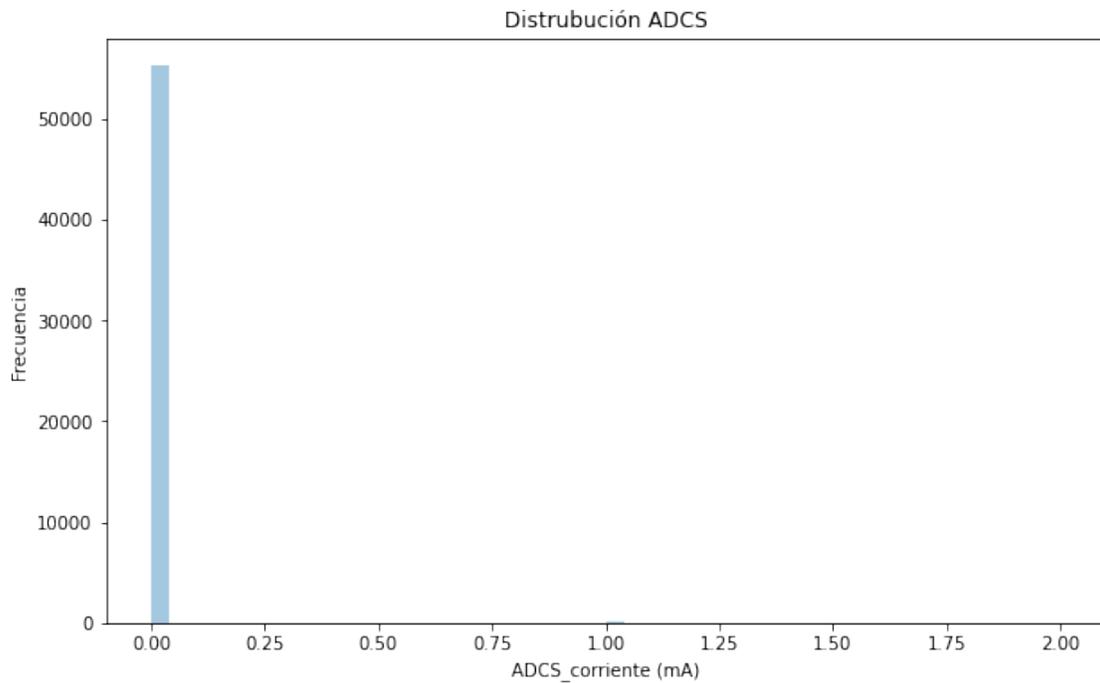
```
[ ]: count    13145.000000
      mean     0.619022
      std     0.364494
      min     0.020770
      25%    0.556700
      50%    0.619010
      75%    0.693790
      max     1.736560
      Name: average_power, dtype: float64
```

```
[ ]: sns.distplot(a=df0['ADCS_current'], label="ADCS_current", kde = False)
      plt.title("Distribución ADCS")
      plt.xlabel('ADCS_corriente (mA)')
      plt.ylabel('Frecuencia')
```

```
plt.show()
```

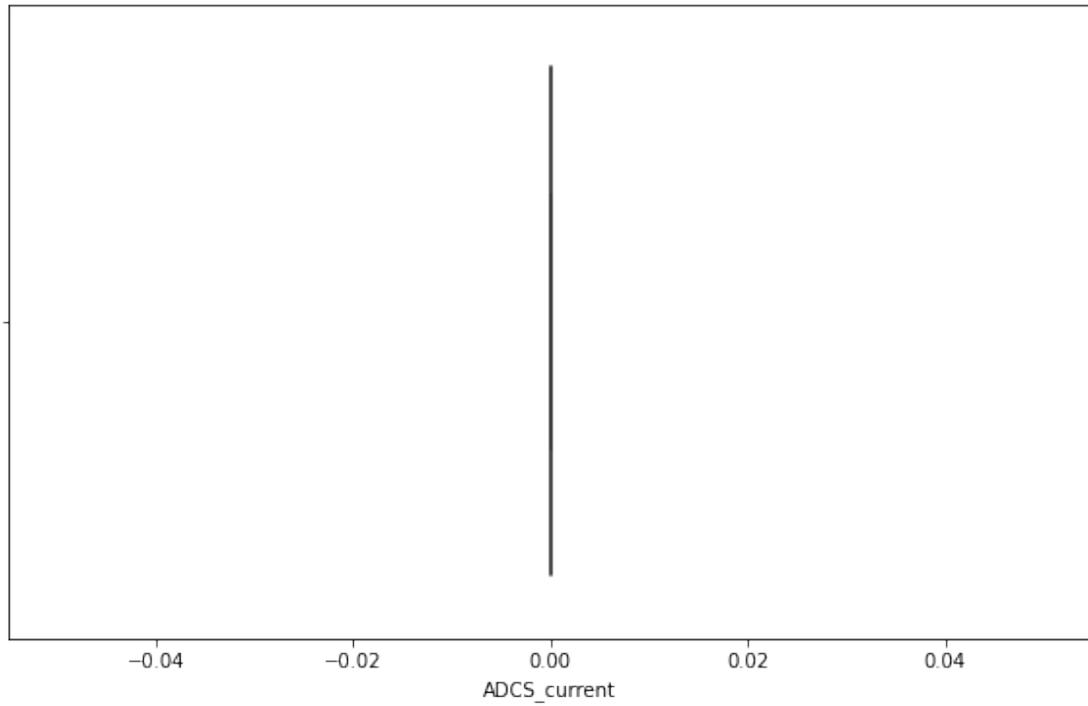
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```



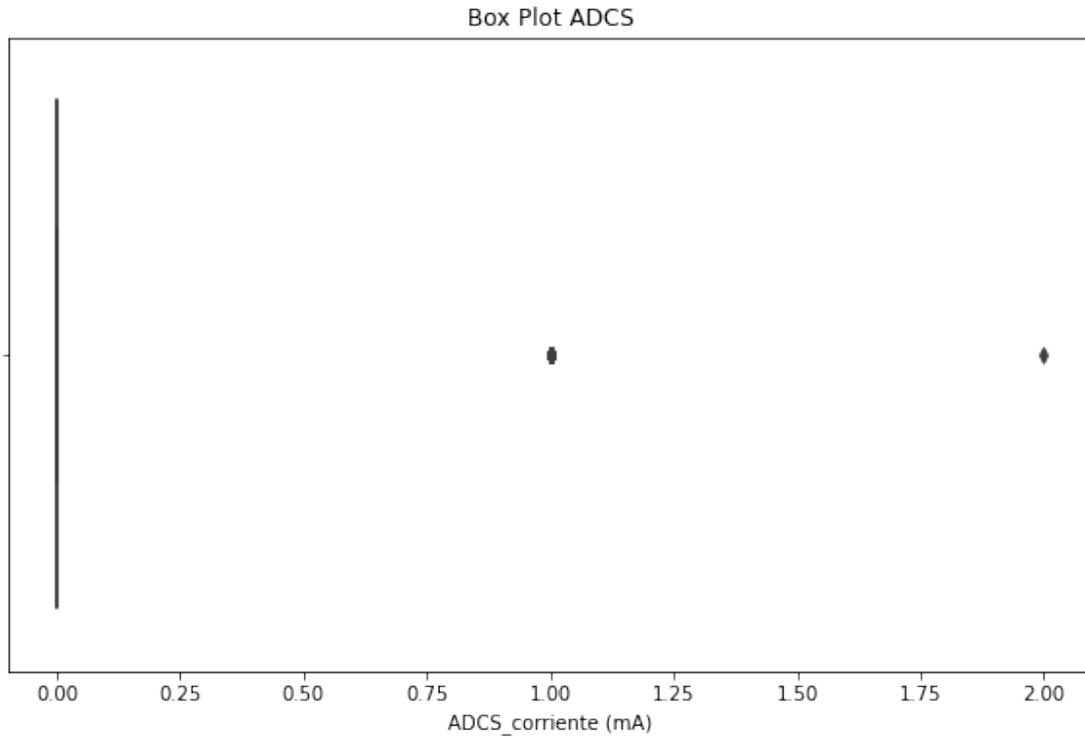
```
[ ]: adcs = df0[df0['ADCS_current'] < 1]  
sns.boxplot(x='ADCS_current', data = adcs)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe719d3a1d0>
```



```
[ ]: sns.boxplot(x='ADCS_current', data = df0)
plt.title("Box Plot ADCS")
plt.xlabel('ADCS_corriente (mA)')

plt.show()
```

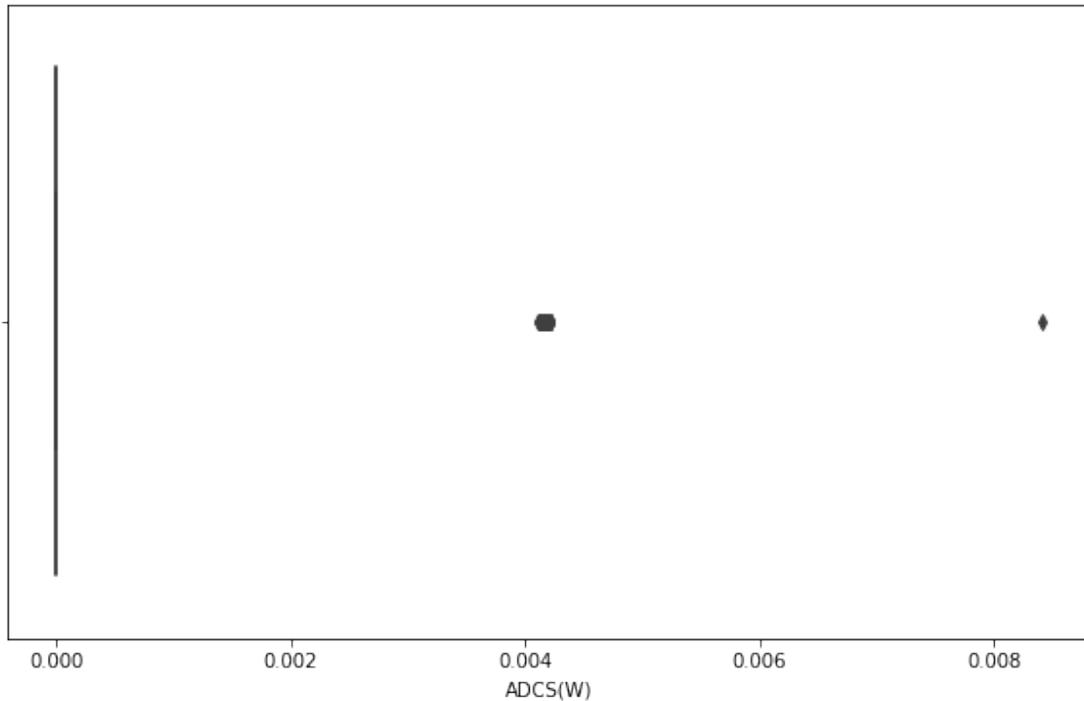


ADCS power distribution

```
[ ]: ADCS = df0[df0['ADCS_current'] >= 0]
      ADCS_power = (ADCS.ADCS_current/1000) * ADCS.battery_voltage
      sns.boxplot(x=ADCS_power, data = ADCS)
      plt.xlabel('ADCS(W)')
      print(ADCS_power.max())
      print(ADCS_power.min())
```

0.00842

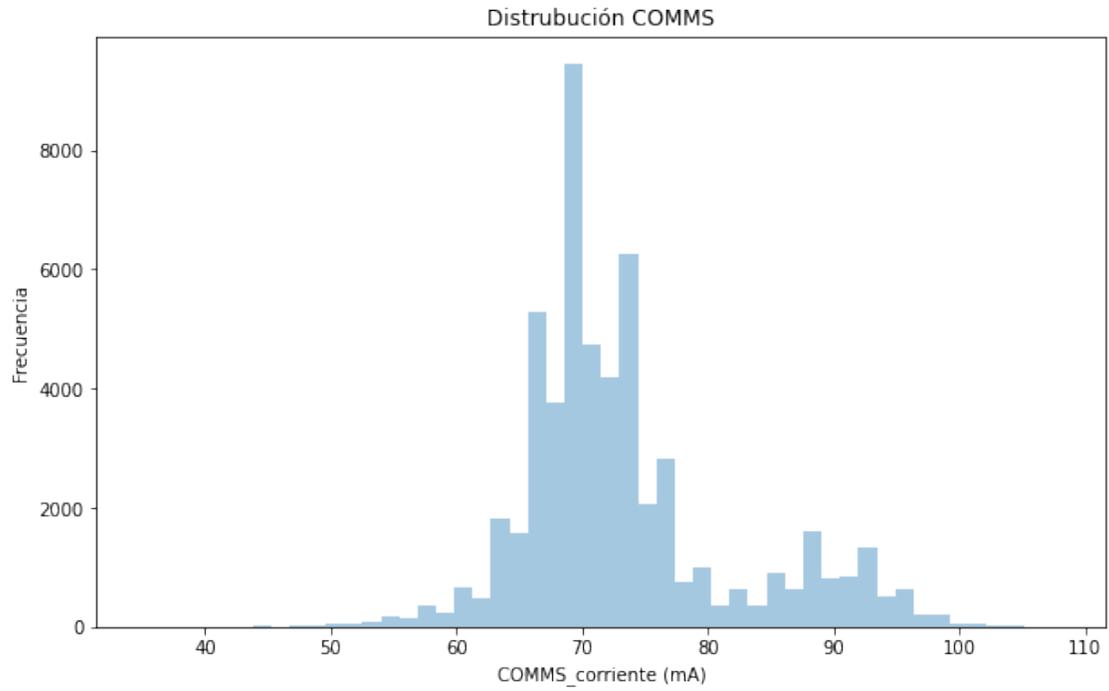
0.0



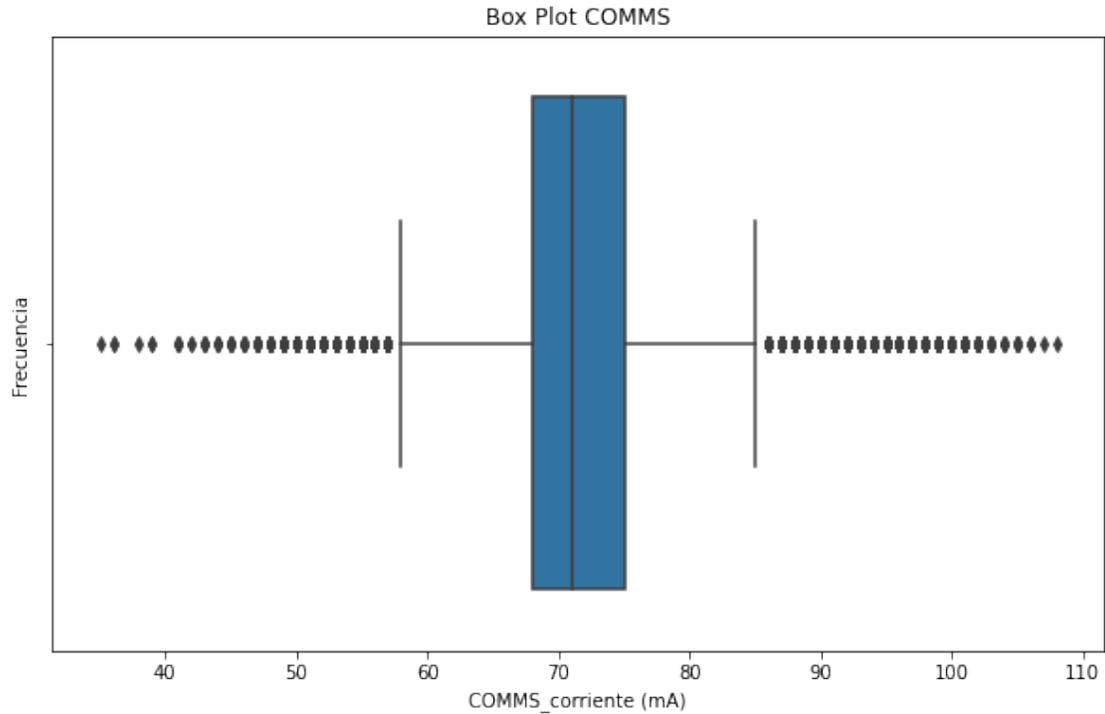
```
[ ]: sns.distplot(a=df0['COMMS_current'], label="COMMS_current", kde = False)
plt.title("Distribución COMMS")
plt.xlabel('COMMS_corriente (mA)')
plt.ylabel('Frecuencia')
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



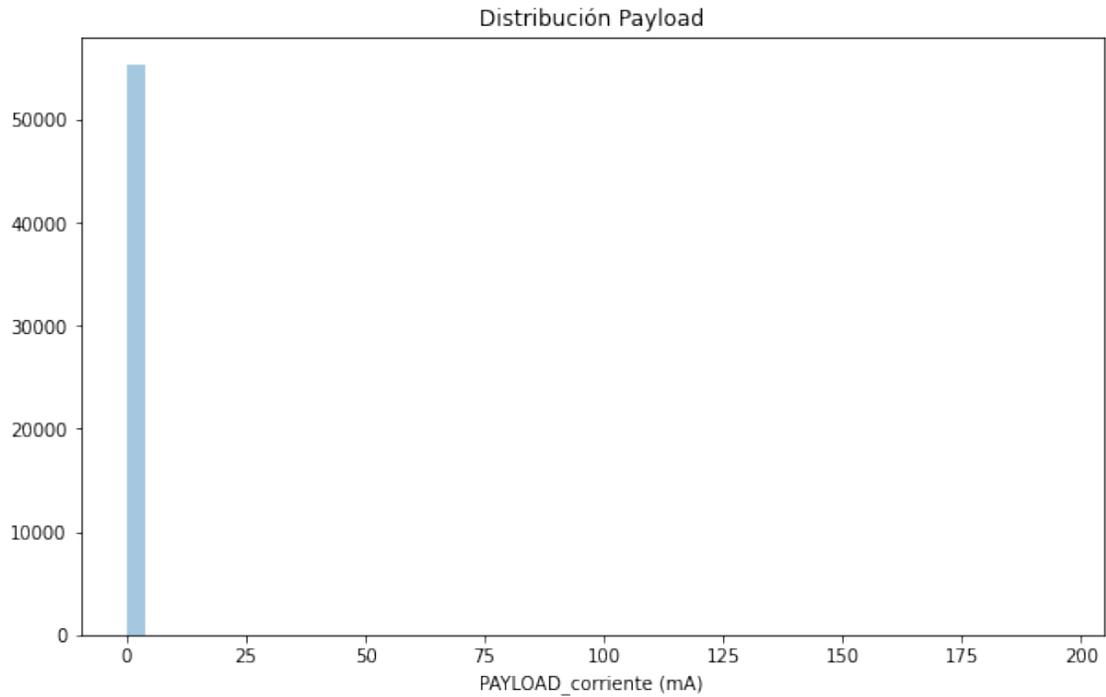
```
[ ]: sns.boxplot(x='COMMS_current', data = df0)
plt.title("Box Plot COMMS")
plt.xlabel('COMMS_corriente (mA)')
plt.ylabel('Frecuencia')
plt.show()
```



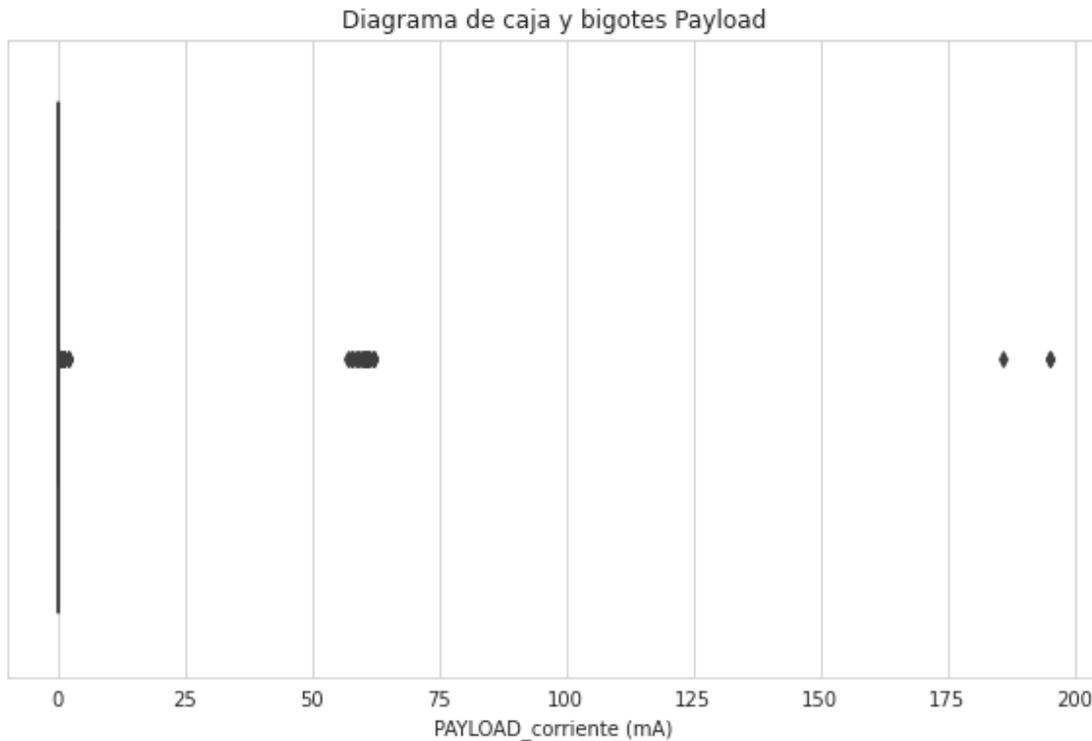
```
[ ]: sns.distplot(a=df0['PAYLOAD_current'], label="PAYLOAD", kde = False)
plt.title("Distribución Payload")
plt.xlabel('PAYLOAD_corriente (mA)')
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



```
[ ]: rcParams['figure.figsize'] = 10,6
sns.set_style('whitegrid')
sns.boxplot(x='PAYLOAD_current', data = df0)
plt.title("Diagrama de caja y bigotes Payload")
plt.xlabel('PAYLOAD_corriente (mA)')
plt.show()
```

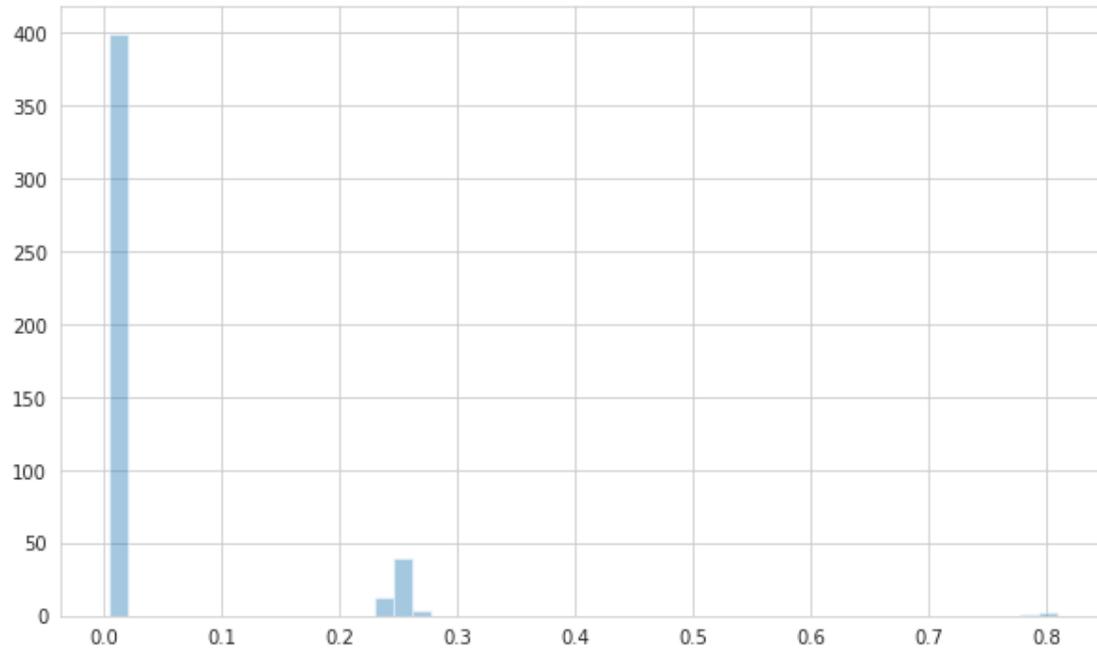


#Payload Power

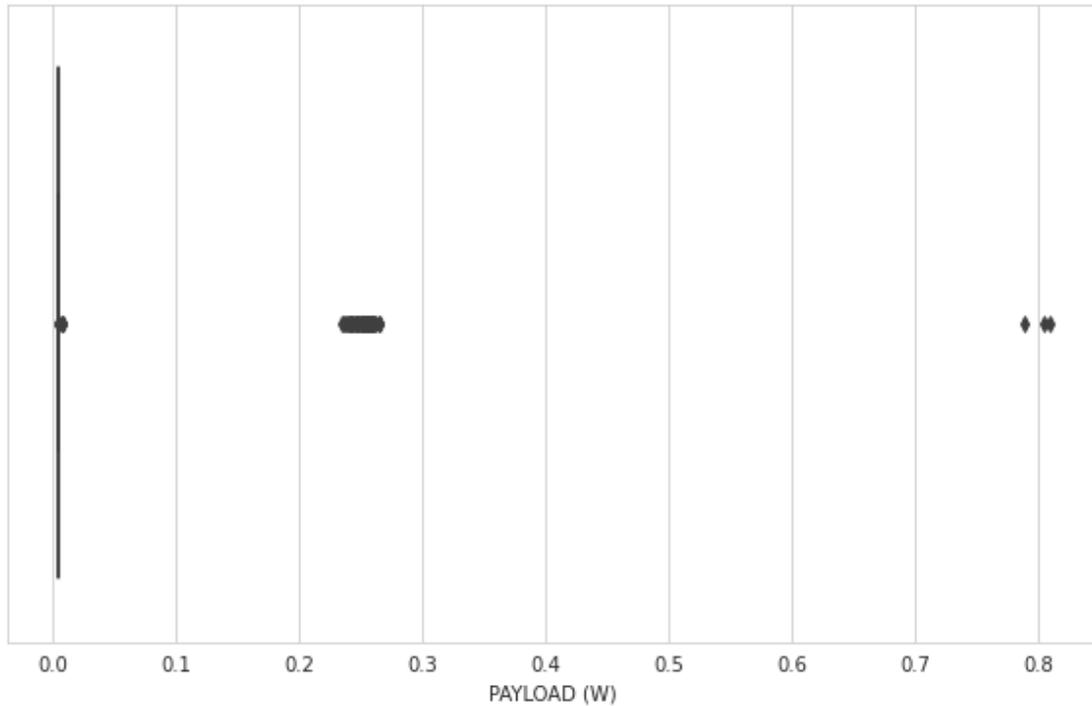
```
[ ]: payload = df0[df0["PAYLOAD_current"] != 0]
      payload_power = (payload.PAYLOAD_current/1000) * payload.battery_voltage

      sns.distplot(a=payload_power, label="payload_power", kde = False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```



```
[ ]: sns.boxplot(x=payload_power, data = payload_power)
plt.xlabel('PAYLOAD (W)')
plt.show()
print(payload_power.max())
print(payload_power.min())
print(len(payload_power != 0))
```

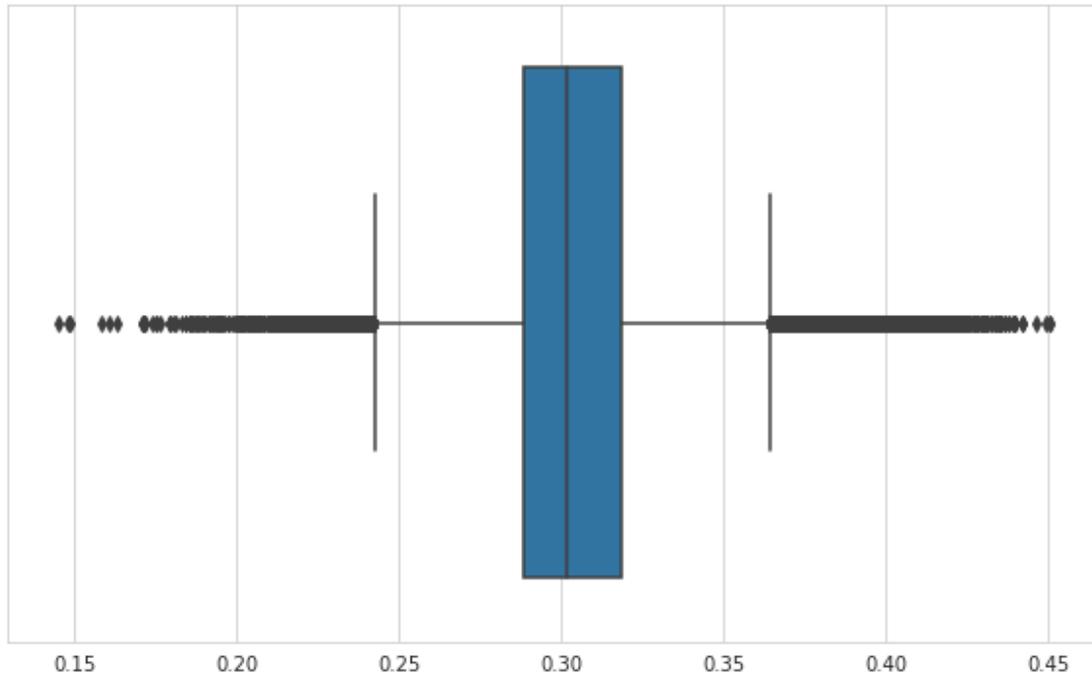


```
0.8092500000000001
0.0039900000000000005
456
```

```
#COMMS Power
```

```
[ ]: COMMS = df0[df0["COMMS_current"] != 0]
COMMS_power = (COMMS.COMMS_current/1000) * COMMS.battery_voltage
sns.boxplot(x=COMMS_power, data = COMMS )
print(COMMS_power.max())
print(COMMS_power.mean())
print(COMMS_power.min())
```

```
0.45047
0.30896565685053035
0.1449
```

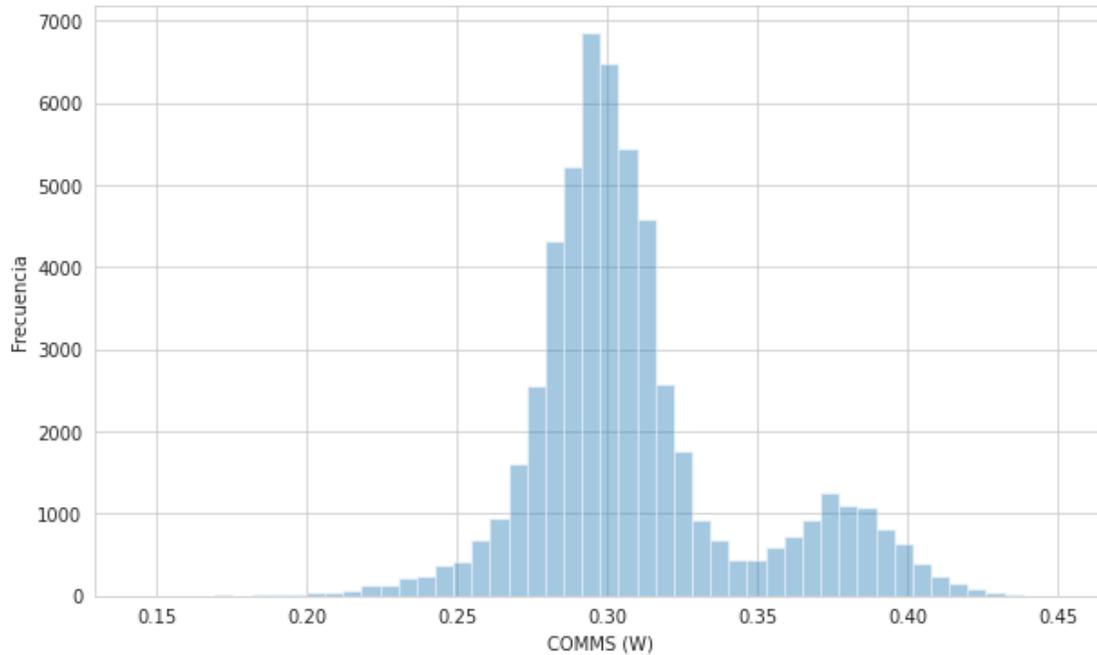


```
[ ]: sns.distplot(a=COMMS_power, label="COMMS_power", kde = False);
plt.xlabel('COMMS (W)')
plt.ylabel('Frecuencia')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```

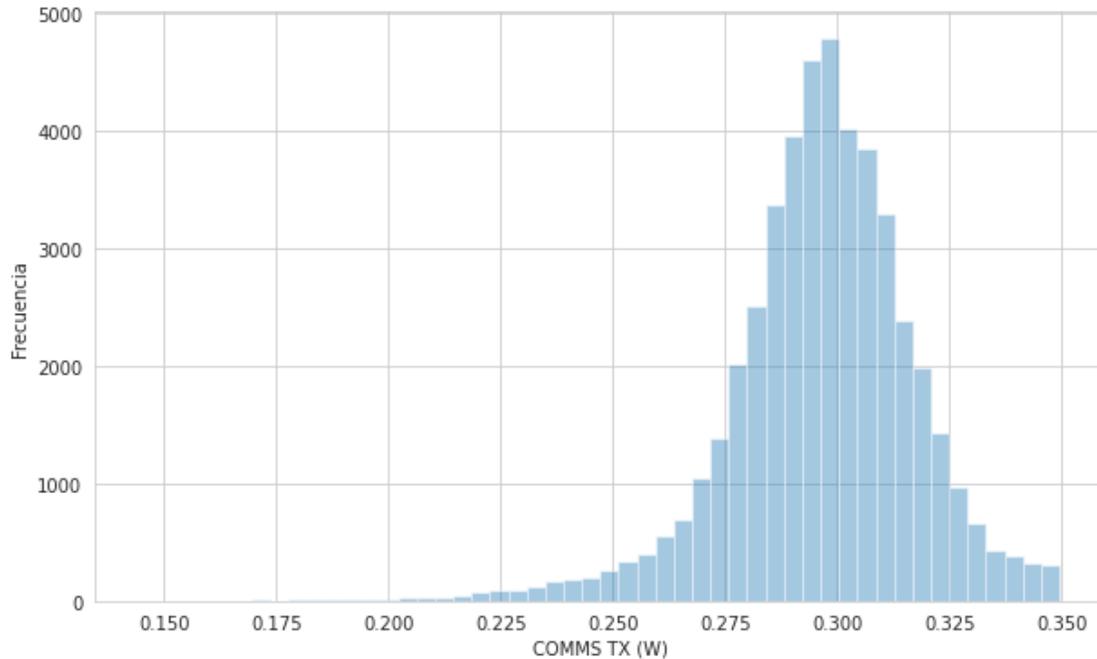


```
[ ]: COMMS_TX = COMMS_power[COMMS_power < 0.35]
sns.distplot(a=COMMS_TX, label="COMMS_TX", kde = False);
# print(COMMS_TX.max())
# print(COMMS_TX.mean())
# print(COMMS_TX.min())
plt.xlabel('COMMS TX (W)')
plt.ylabel('Frecuencia')
# COMMS_TX
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



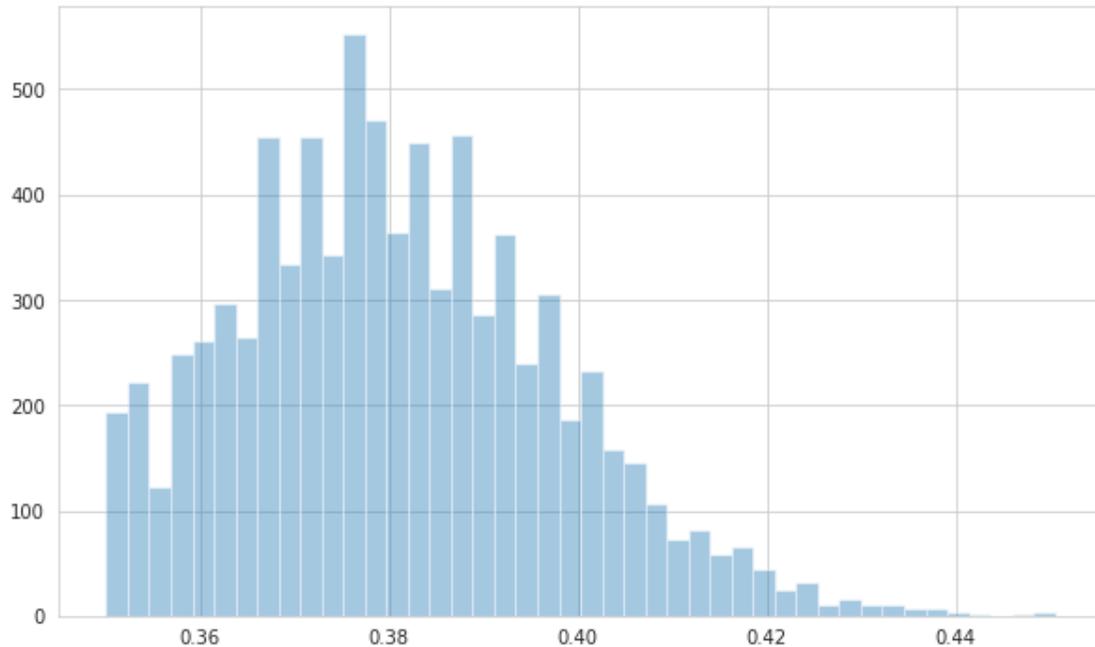
```
[ ]: # df1.average_power.describe()
COMMS_TX
```

```
[ ]: 0      0.28712
      1      0.27936
      2      0.27264
      3      0.26202
      4      0.29230
      ...
      63146  0.25620
      63147  0.26397
      63148  0.31080
      63149  0.26397
      63150  0.29400
      Length: 47052, dtype: float64
```

```
[ ]: COMMS_RX = COMMS_power[COMMS_power > 0.35]
      sns.distplot(a=COMMS_RX, label="COMMS_RX", kde = False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```



#Análisis del primer escenario

```
[ ]: #ADCScurrent, comms current, payload current RX es el modo de recepcion de
      ↳COMMS
first_scenary = df0[df0["PAYLOAD_current"] == 0] # Se separan los datos
      ↳respecto a que el consumo de payload es 0
first_scenary = first_scenary[first_scenary['average_power'] < 0]

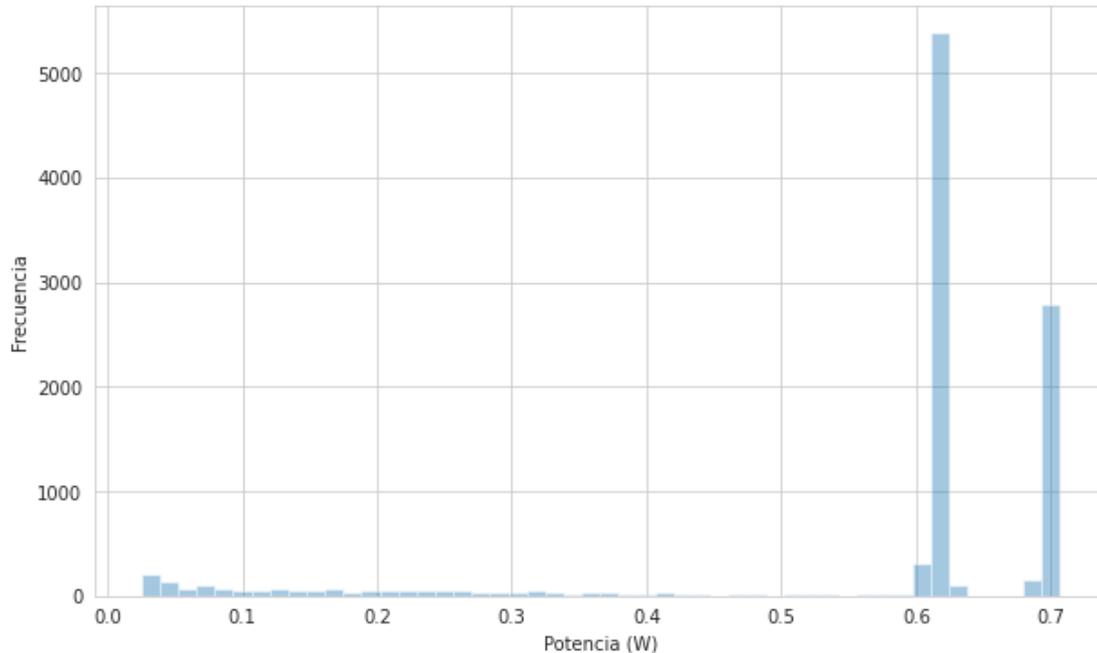
first_scenary = first_scenary[first_scenary['HEATER_status'] == 150]
#first_scenary = first_scenary[first_scenary['COMMS_current'] < 0.35]
first_scenary['average_power'] = first_scenary['average_power'].abs()

sns.distplot(a=first_scenary['average_power'], label="average_power", kde =
      ↳False);
plt.xlabel('Potencia (W)')
plt.ylabel('Frecuencia')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



```
[ ]: first_scenary['average_power'].describe()
```

```
[ ]: count    10354.000000
      mean      0.572702
      std       0.179739
      min       0.024930
      25%       0.614860
      50%       0.619010
      75%       0.693790
      max       0.706260
      Name: average_power, dtype: float64
```

#Análisis del segundo escenario

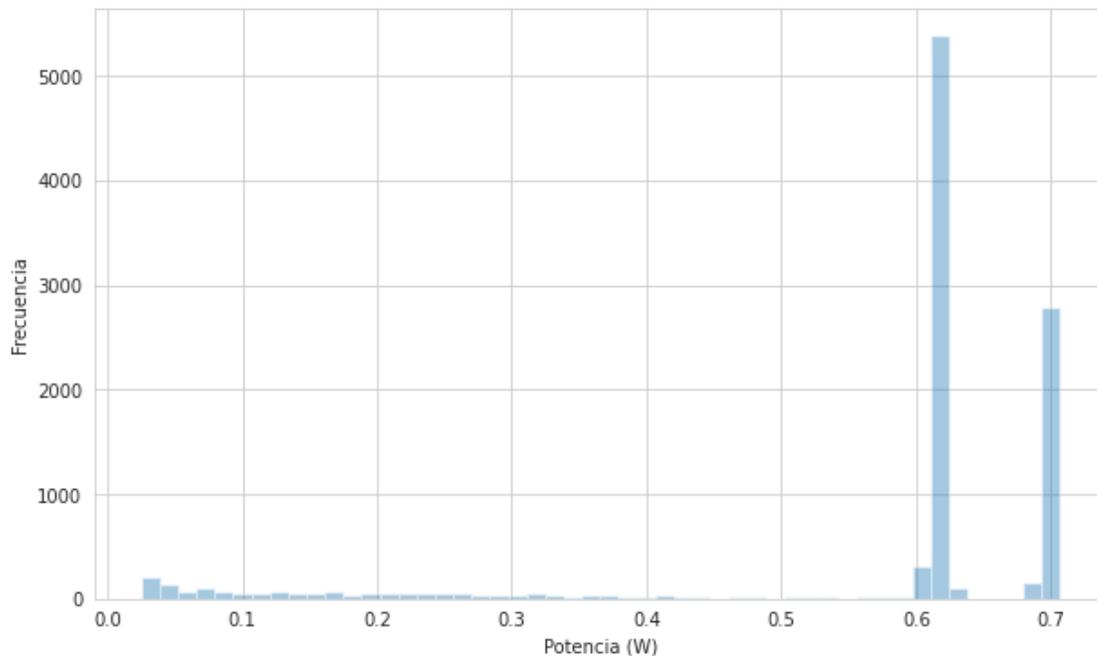
```
[ ]: second_scenary = df0[df0['PAYLOAD_current'] == 0]
      second_scenary = second_scenary[second_scenary['average_power'] < 0]
      second_scenary = second_scenary[second_scenary['HEATER_status'] == 150]
      second_scenary['average_power'] = second_scenary['average_power'].abs()
      sns.distplot(a=second_scenary['average_power'], label="average Power", kde =_
      →False)
      plt.xlabel('Potencia (W)')
      plt.ylabel('Frecuencia')
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level

function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



#Thrid Escenary

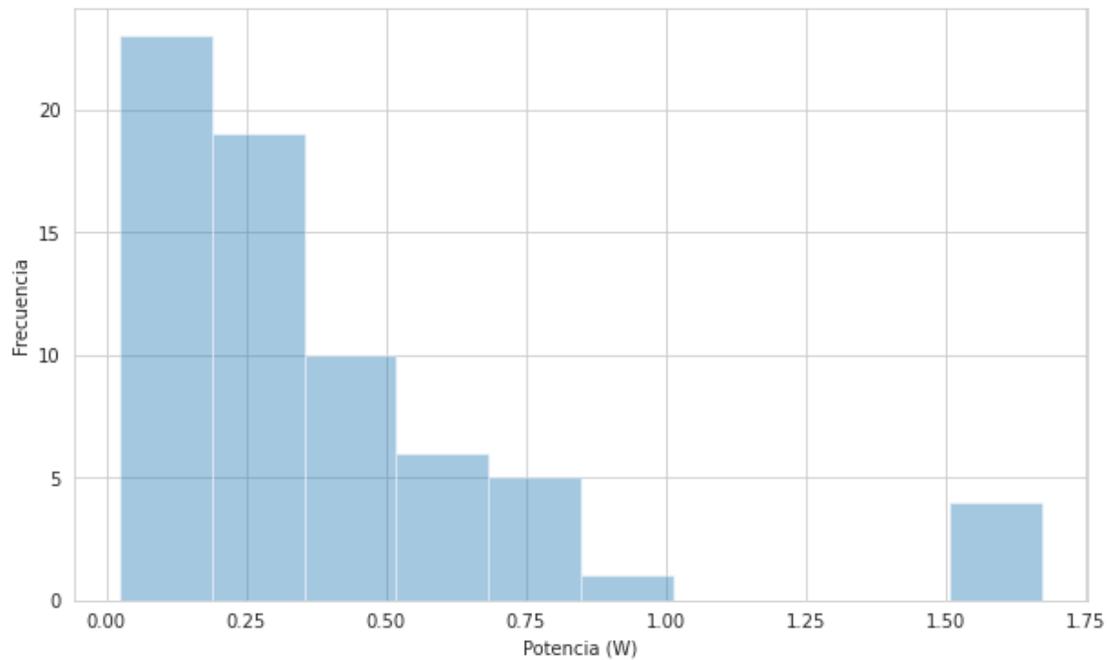
```
[ ]: #Tercer Escenario
third_scenary = df0[df0['PAYLOAD_current'] != 0]
# third_scenary = third_scenary[third_scenary['HEATER_status'] == 153]
third_scenary = third_scenary[third_scenary['average_power'] < 0]
third_scenary['average_power'] = third_scenary['average_power'].abs()
sns.distplot(a=third_scenary['average_power'], label="average Power", kde =_
→False)
plt.xlabel('Potencia (W)')
plt.ylabel('Frecuencia')
len(third_scenary)
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:

FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: 68
```



```
[ ]: third_scenary.average_power.describe()
```

```
[ ]: count    57.000000
      mean     0.336219
      std     0.392729
      min     0.024930
      25%    0.128790
      50%    0.253420
      75%    0.382210
      max     1.670090
      Name: average_power, dtype: float64
```

```
#State of charge distribution
```

```
[ ]: print(len(df0[df0['state_charge'] == 100])) #71% esta en 100% de carga
      print(len(df0))
```

```
39227
```

```
55317
```

```
#Resultados teóricos de los módulos del Quetzal-1.
```

```
[ ]: uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

Saving Teoricos.xlsx to Teoricos.xlsx

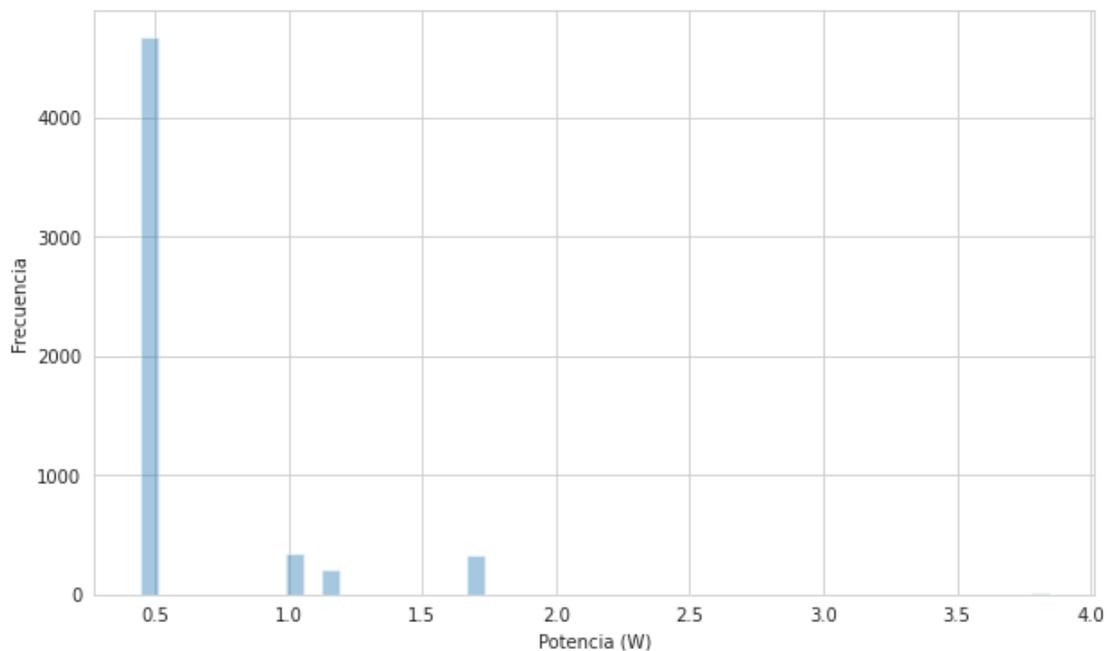
```
[ ]: teoricos = pd.read_excel(io.BytesIO(uploaded['Teoricos.
→xlsx']), sheet_name='escenario3') # Excel de los resultados del simulador del
→Qetzal-1 con cada escenario
# en diferentes páginas
```

```
[ ]: sns.distplot(a=teoricos['TOTAL (W)'], label="teoricos", kde = False)
plt.xlabel('Potencia (W)')
plt.ylabel('Frecuencia')
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



```
[ ]: teoricos['TOTAL (W)'].describe()
```

```
[ ]: count    5564.000000
      mean      0.587502
      std      0.366480
```

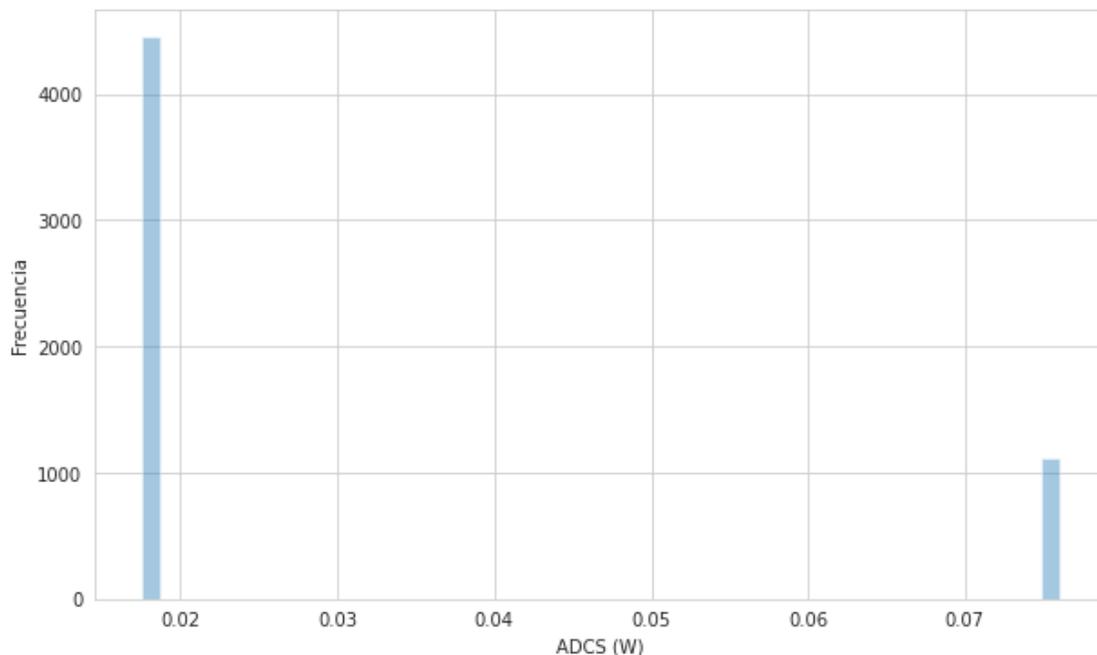
```
min          0.444018
25%          0.444018
50%          0.444018
75%          0.444018
max          3.843521
Name: TOTAL (W), dtype: float64
```

```
[ ]: sns.distplot(a=teoricos['ADCS (W)'], label="ADCS (W)", kde = False)
plt.xlabel('ADCS (W)')
plt.ylabel('Frecuencia')

plt.show()
```

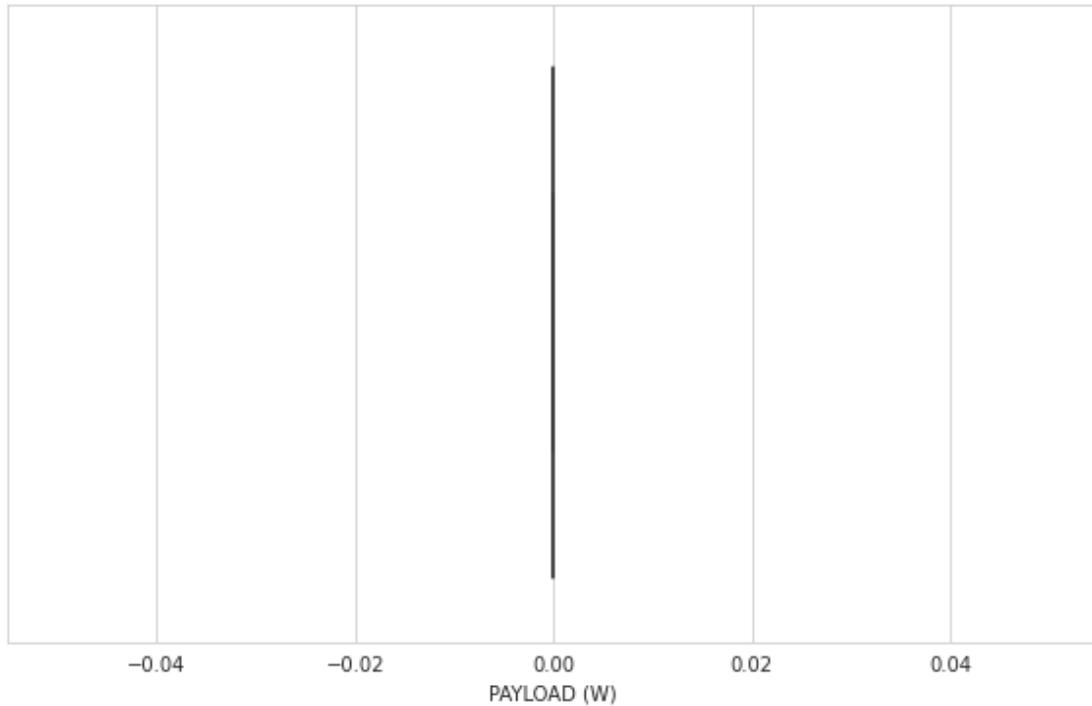
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



```
[ ]: sns.boxplot(x= 'PAYLOAD (W)', data = teoricos)
plt.xlabel('PAYLOAD (W)')
```

```
[ ]: Text(0.5, 0, 'PAYLOAD (W)')
```

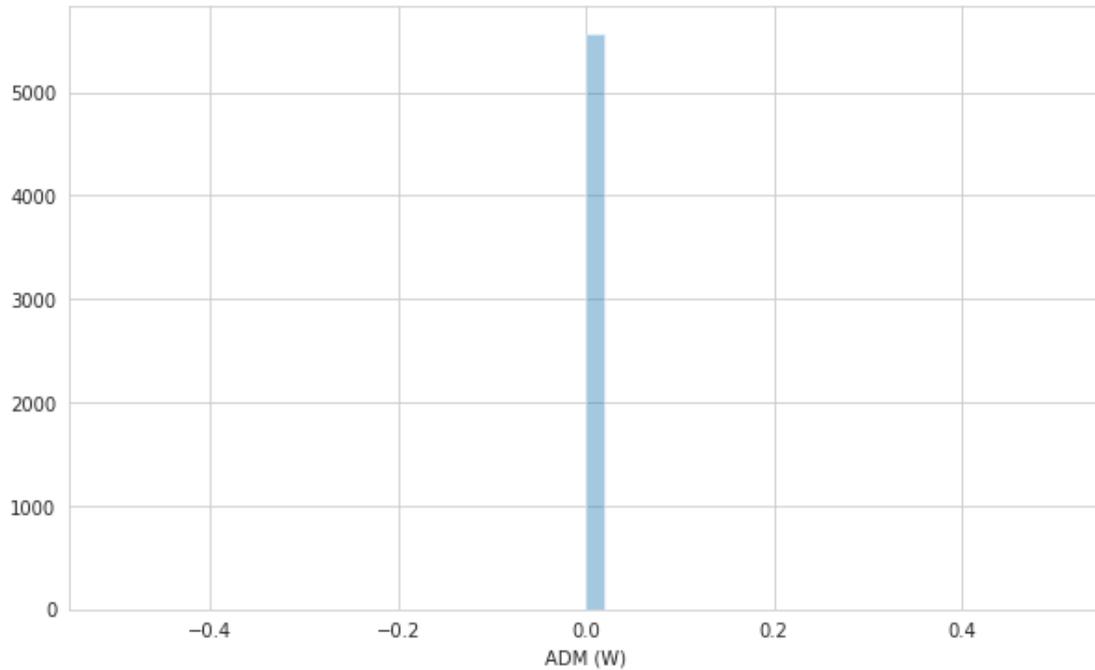


```
[ ]: sns.distplot(a=teoricos['ADM (W)'], label="ANTENNA (W)", kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe718f7c490>
```

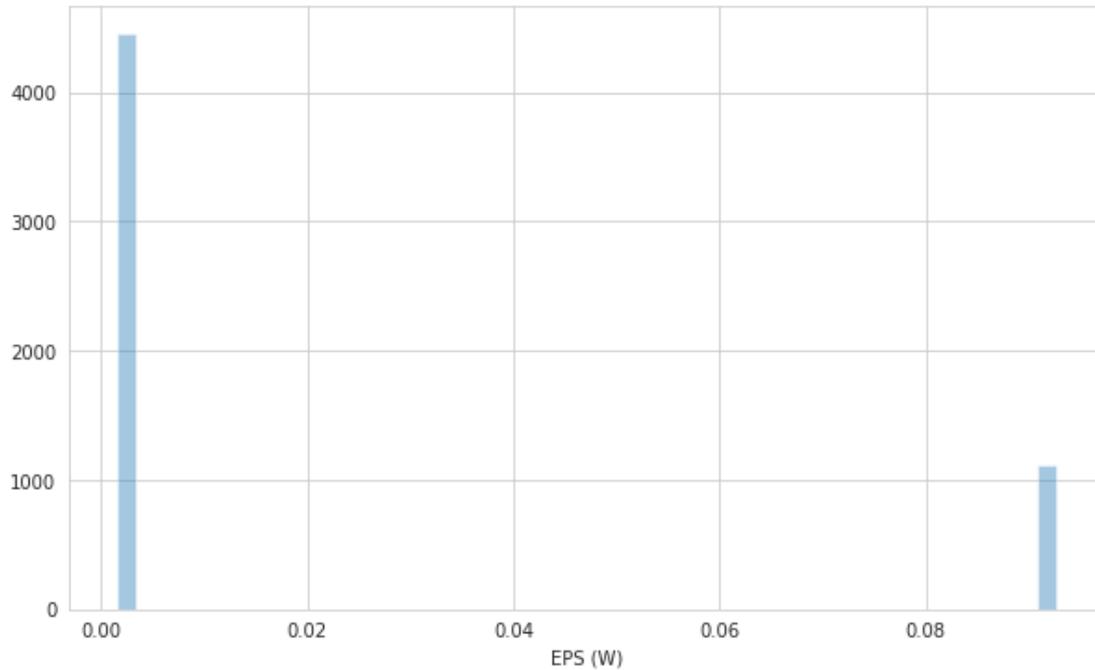


```
[ ]: sns.distplot(a=teoricos['EPS (W)'], label="EPS (W)", kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe713e3abd0>
```

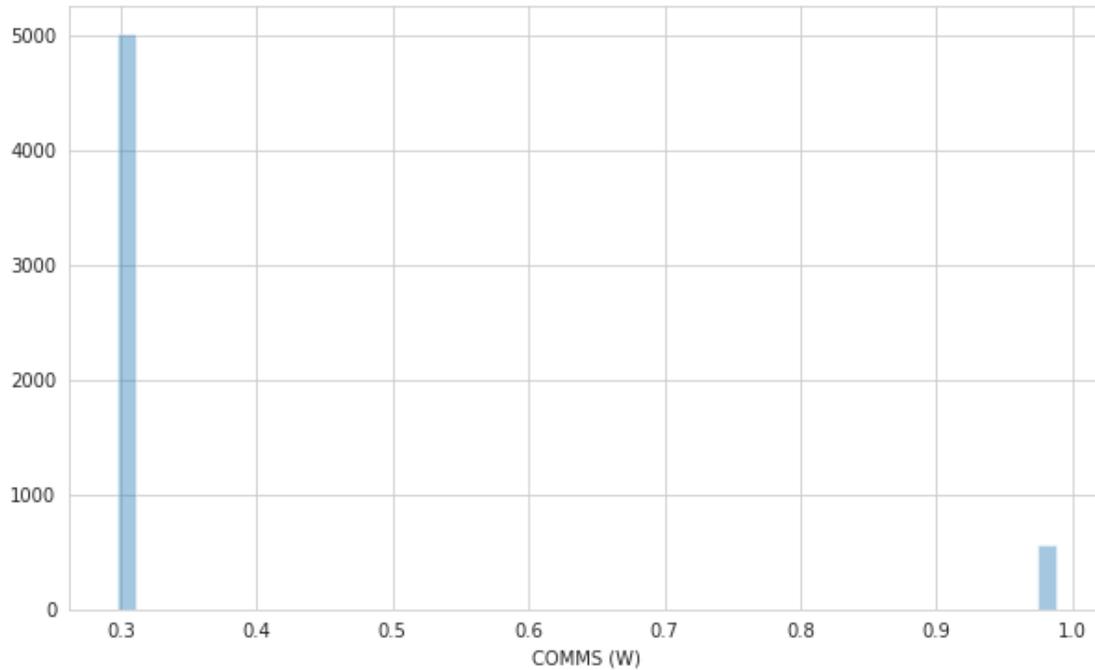


```
[ ]: sns.distplot(a=teoricos['COMMS (W)'], label="COMMS (W)", kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe719c96490>
```

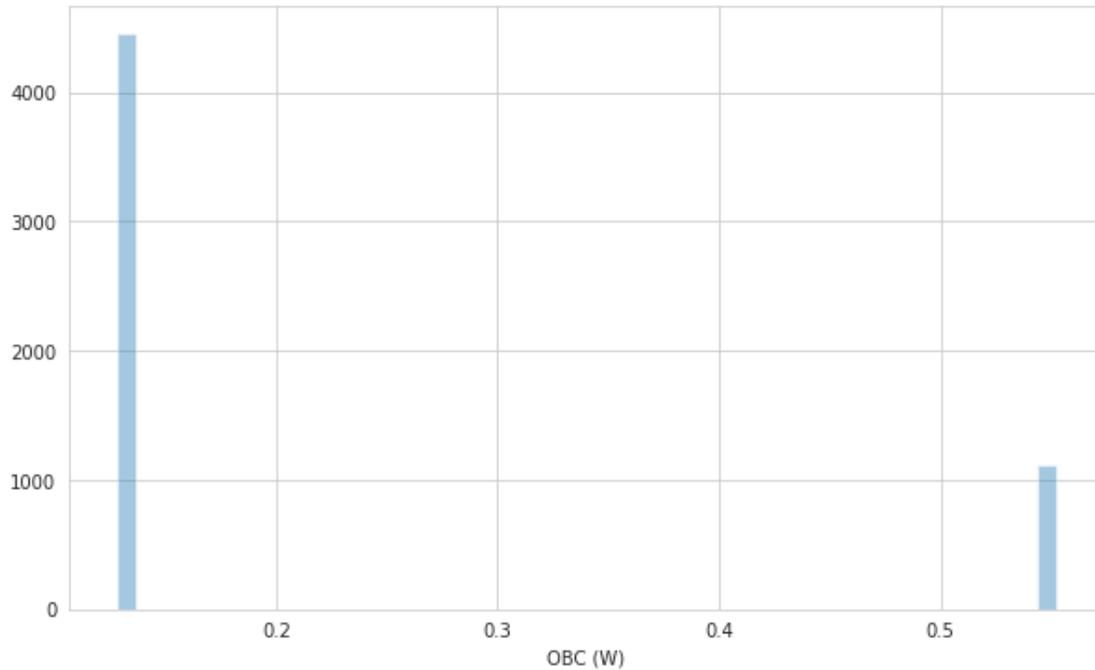


```
[ ]: sns.distplot(a=teoricos['OBC (W)'], label="OBC (W)", kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe719918690>
```



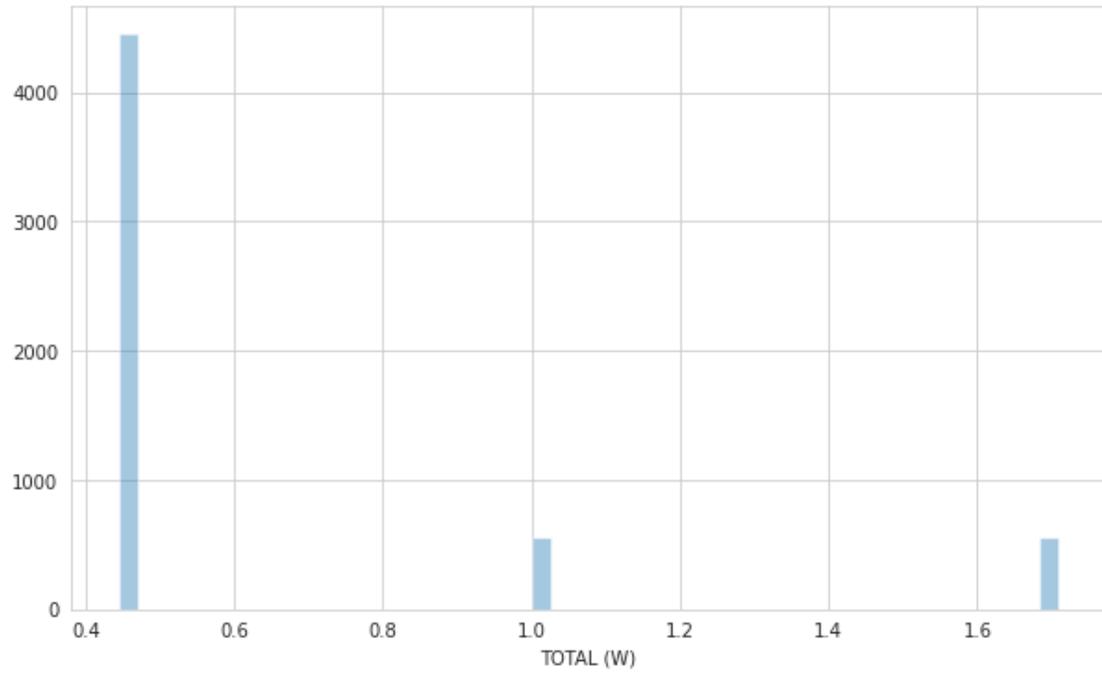
```
[ ]: sns.distplot(a=teoricos['TOTAL (W)'], label="TOTAL (W)", kde = False)
print(teoricos['TOTAL (W)'].max())
print(teoricos['TOTAL (W)'].min())
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
1.7086275
```

```
0.44401787
```



# Anexo 5. Código en Python para Presupuesto de potencia CubeSat 2U

November 2, 2021

## 1 Power Budget CubeSat 2U

En este Google Collab notebook se realiza el presupuesto de potencia para un CubeSat 2U, se recomienda revisar los siguientes trabajos realizados del presupuesto de potencia del Quetzal-1: “TAB-PWR-079 Modules Consumption and Scenarios.xlsx”, “TAB-PWR-078 Power Budget.xlsx” y ” TAB-PWR-081 Solar Panels Power Generation Simulator.m”; que se encuentran en la carpeta del Quetzal-1/ Power/ Power Budget.

#Libraries

```
[ ]: import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
rcParams['figure.figsize'] = 10,6
# sns.set_style('whitegrid')
```

```
[ ]: !pip install SciencePlots
```

```
Requirement already satisfied: SciencePlots in /usr/local/lib/python3.7/dist-packages (1.0.9)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from SciencePlots) (3.2.2)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->SciencePlots) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->SciencePlots) (2.4.7)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages (from matplotlib->SciencePlots) (1.19.5)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->SciencePlots) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->SciencePlots) (0.10.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycler>=0.10->matplotlib->SciencePlots) (1.15.0)
```

```
[ ]: plt.style.reload_library()
plt.style.use(['science', 'notebook', 'grid'])
```

#Scenario setup

```
[ ]: status_adcs = "ON"
status_payload = "OFF"
status_adm = "ON"
status_comms = "RX" #TX-RX
#EPS = ON
status_obc = "ON"
status_deorbit = "OFF"
```

#Constantes importantes del simulador de generacion de potencia de los paneles soalres

```
[ ]: N = 5564 #período orbital. una orbita = 5564 s y un dia = 89009 s
SOC = 47 # estado de carga de las baterías.
#Earth Constants
Radio_E = 6378 #Earth radio [km]
mu_E = 398600 #Earth gravitational constant [km3/s2]

#ISS orbit constant
h_ISS = 408 #ISS height [km]
inclination_ISS = math.radians(51.64) #orbit inclination [°]

#CubeSat Movement
Radio_CubeSat = Radio_E + h_ISS #radio of the CubeSat orbit [km]
rot_z = 3 # angular speed z given by a pasive ADCS [°/s]
rot_z_rad = math.radians(rot_z) # angular speed z given by a pasive ADCS [rad/s]
V_tan = math.sqrt(mu_E/Radio_CubeSat) # tangential speed [km/s]
w = V_tan/Radio_CubeSat # angular speed of the CubeSat [rad/s]
Orbital_periodo = 5563 #[seg]
n_orbit = 16
total_orbit = n_orbit * Orbital_periodo
p = 2*(math.pi*Radio_CubeSat)/V_tan
#CubeSat Solar Panel Unit Vectors and location from Earth
SP_top = np.array([[0], [0], [1]])
SP_bottom = np.array([[0], [0], [-1]])
SP_fpp1 = np.array([[0], [-1], [0]])
SP_side1 = np.array([[0], [1], [0]])
SP_fpp2 = np.array([[1], [0], [0]])
SP_side2 = np.array([[ -1], [0], [0]])
CubeSat = np.array([[0], [Radio_CubeSat], [0]]) # location of the cubesat at  $t_0$ 
↳ local reference of Earth

#Solar Cells constant
efficiency = 17.34 #[%]
```

```
area = 0.003018 # [m^2]
temp_fac = 0.85 # [%] is a temperature factor

# battery Type: Polymer Li-ion Rechargeable Battery
Qn = 4 # the nominal capacity of the 2 batteries array in Ah.
Va = 3.7 # s the voltage batteries array in V, since the batteries array is in
↳parallel.
```

## #ADCS POWER CONSUMPTION

```
[ ]: adcs_nominal = 0.00
adcs_peak = 0.008
```

## 1.1 PAYLOAD POWER CONSUMPTION

```
[ ]: payload_nominal = 0.00
payload_peak = 0.8
motor_movement_cycle = 1
time_between_cycles = 600
motor_movement_time = 220 #ms
#payload camera data
soc_photos = 65
t_cycles = 25
cam_peak_time = 400
```

## 1.2 ADM POWER CONSUMPTION

```
[ ]: adm_nominal = 0.00
adm_peak = 2.96
deployment_duration = 30
deployment_start = 1800
deployment_cycles = 4
wait_time = 5
```

## 1.3 COMMS POWER CONSUMPTION

```
[ ]: comms_nominal = 0.2963
comms_peak = 0.375
comms_times_per_cycle = 1
comms_tbetween = 10
comms_tx_time = 300 # ms
```

## 2 OBC POWER CONSUMPTION

```
[ ]: obc_nominal = 0.12800
obc_peak = 0.552
```

## 2.1 EPS POWER CONSUMPTION

```
[ ]: eps_nominal = 0.0270737
     eps_peak = 0.0926075
     eps_low = 0.0015179
```

## 3 Deorbit System

```
[ ]: deorbit_nominal = 0
     deorbit_peak = 2.96
     resis_deployment_duration = 60
     resis_deployment_start = 5400
     resis_deployment_cycles = 2
     resis_wait_time = 5
```

#Equations used in Solar Panels Power Generation Simulator

```
[ ]: #Function to get the position of the CubeSat on the orbit rotating the x
     #axis
     def Orbit_x(vector_CubeSat,angle):
         mati = np.mat([[1, 0, 0],[0, math.cos(angle), -math.sin(angle)],[0, math.
         ↪sin(angle), math.cos(angle)]])
         mati = mati.round(2)

         return np.dot(mati, vector_CubeSat)
     #Function to get the position of the CubeSat on the orbit rotating the z
     #axis
     def Orbit_z(vector_CubeSat,angle):
         mati = np.mat([[ math.cos(angle), -math.sin(angle), 0],[ math.sin(angle),
         ↪math.cos(angle), 0], [0, 0, 1]])
         mati = mati.round(2)

         return np.dot(mati, vector_CubeSat)
     #Function to get the latitud angle of the CubeSat on the Earth
     #this latitud with help to find the nadier angle of the magnetic field
     def Latitud_funcion(v):
         return math.atan(v[2]/np.linalg.norm(v))
     #Function to get Angle Between Magnetic Field Vector and Local Nadier
     def Nadier_function(angle_lat):
         return 3.842 * math.cos(0.6077 * angle_lat + 2.286) + 4.11
     #Function to get the angle between the solar panel and the sun rays
     def Angle_sun(panel):
         sun = np.array([[0], [-1368], [0]])
         dot0 = panel[0] * sun[0]
         dot1 = panel[1] * sun[1]
         dot2 = panel[2] * sun[2]
         dotFinal = dot0[0] + dot1[0] + dot2[0]
```

```

    return math.acos(dotFinal/(np.linalg.norm(panel) * np.linalg.norm(sun)))
# Function to get the efficiency of the solar cells
def Eff_function(angle):
    return 17.34 * (math.cos(angle)/100)

```

## 4 Power consumption per module

```

[ ]: column_names = ["time(s)", "ADCS(W)", "PAYLOAD(W)", "ADM(W)", "COMMS(W)",
↳"EPS(W)", "OBC(W)",'Deorbit(W)', 'Power Consumption(W)', 'Energy_
↳Consumption(Wh)', 'total(Wh)']

df = pd.DataFrame(columns = column_names)
df['time(s)'] = np.arange(N)

```

```

[ ]: adcs_w = []
payload_w = []
adm_w = []
comms_w = []
eps_w = []
obc_w = []
deorbit_w = []
total_w = []
ec_wh = []
total_wh = []
for i in range(Orbital_periodo + 1):
#Iteraciones de columna de consumo de potencia de ADCS
    if status_adcs == "ON":
        if i % 10 == 0 or i % 10 == 1:
            adcs_w.append(adcs_peak)
        else:
            adcs_w.append(adcs_nominal)
    else:
        adcs_w.append(0)
#Iteraciones de columna de consumo de potencia de PAYLOAD
    if status_payload == 'ON':
        if i % time_between_cycles == 0:
            payload_w.append(payload_nominal + payload_peak)
        else:
            payload_w.append(0)
    else:
        payload_w.append(0)
#Iteraciones de columna de consumo de potencia de ADM
    if status_adm == "ON":
        if i >= deployment_start and i < deployment_start + (deployment_cycles *
↳(deployment_duration + wait_time)):

```

```

        if i >= int(i/(deployment_duration + wait_time)) * (deployment_duration +
↪wait_time) and i <= deployment_duration + int(i/(deployment_duration +
↪wait_time)) * (deployment_duration + wait_time):
            adm_w.append(adm_peak)
        else:
            adm_w.append(0)
        else:
            adm_w.append(0)
        else:
            adm_w.append(0)
#Iteraciones de columna de consumo de potencia de COMMS
        if status_comms == 'TX-RX':
            if i >= int(i/comms_tbetween) * comms_tbetween and i <=
↪(comms_times_per_cycle/2) + int(i/comms_tbetween) * comms_tbetween:
                comms_w.append(comms_peak) #originalmente era (comms_peak*comms_tx_time/
↪1000)
            else:
                comms_w.append(comms_nominal)
        else:
            comms_w.append(comms_nominal)
#Iteraciones de columna de consumo de potencia de EPS
        if adcs_w[i] == adcs_peak or payload_w[i] != 0 or adm_w[i] != 0 or
↪comms_w[i] == (comms_peak): # originalmente era: comms_w[i] ==
↪(comms_peak*comms_tx_time/1000)
            eps_w.append(eps_peak)
        elif adcs_w == adcs_nominal:
            eps_w.append(eps_low)
        else:
            eps_w.append(eps_low)
#Iteraciones de columna de consumo de potencia de OBC
        if status_obc == 'ON':
            if adcs_w[i] == adcs_peak or payload_w[i] != 0 or adm_w[i] != 0 or
↪comms_w[i] == (comms_peak): # originalmente era: comms_w[i] ==
↪(comms_peak*comms_tx_time/1000)
                obc_w.append(obc_peak)
            elif adcs_w == adcs_nominal:
                obc_w.append(obc_nominal)
            else:
                obc_w.append(obc_nominal)
        else:
            obc_w.append(0)
# Iteraciones de columna de consumo de potencia de Deorbit System
        if status_deorbit == 'ON':
            if i >= resis_deployment_start and i < resis_deployment_start +
↪(resis_deployment_cycles * (resis_deployment_duration + resis_wait_time)):

```

```

        if i >= int(i/(resis_deployment_duration + resis_wait_time)) *␣
↳(resis_deployment_duration + resis_wait_time) and i <=␣
↳resis_deployment_duration + int(i/(resis_deployment_duration +␣
↳resis_wait_time)) * (resis_deployment_duration + resis_wait_time):
            deorbit_w.append(deorbit_peak)
        else:
            deorbit_w.append(0)
        else:
            deorbit_w.append(0)
        else:
            deorbit_w.append(0)
# calculo total(W)
total_w.append( adcs_w[i] + payload_w[i] + adm_w[i] + comms_w[i] + eps_w[i] +␣
↳obc_w[i] + deorbit_w[i])
# Energy consumption
ec_wh.append(total_w[i]/3600)
# Total(Wh)
if i == 0:
    total_wh.append(ec_wh[i])
else:
    total_wh.append(total_wh[i-1] + ec_wh[i])
df['ADCS(W)'] = adcs_w
df['PAYLOAD(W)'] = payload_w
df['ADM(W)'] = adm_w
df['COMMS(W)'] = comms_w
df['EPS(W)'] = eps_w
df['OBC(W)'] = obc_w
df['Deorbit(W)'] = deorbit_w
df['Power Consumption(W)'] = total_w
df['Energy Consumption(Wh)'] = ec_wh
df['total(Wh)'] = total_wh
df

```

```

[ ]:      time(s)  ADCS(W)  ...  Energy Consumption(Wh)  total(Wh)
0         0         0.008  ...                0.000264  0.000264
1         1         0.008  ...                0.000264  0.000527
2         2         0.000  ...                0.000118  0.000645
3         3         0.000  ...                0.000118  0.000764
4         4         0.000  ...                0.000118  0.000882
...      ...      ...      ...                ...      ...
5559     5559     0.000  ...                0.000118  0.935206
5560     5560     0.008  ...                0.000264  0.935470
5561     5561     0.008  ...                0.000264  0.935733
5562     5562     0.000  ...                0.000118  0.935852
5563     5563     0.000  ...                0.000118  0.935970

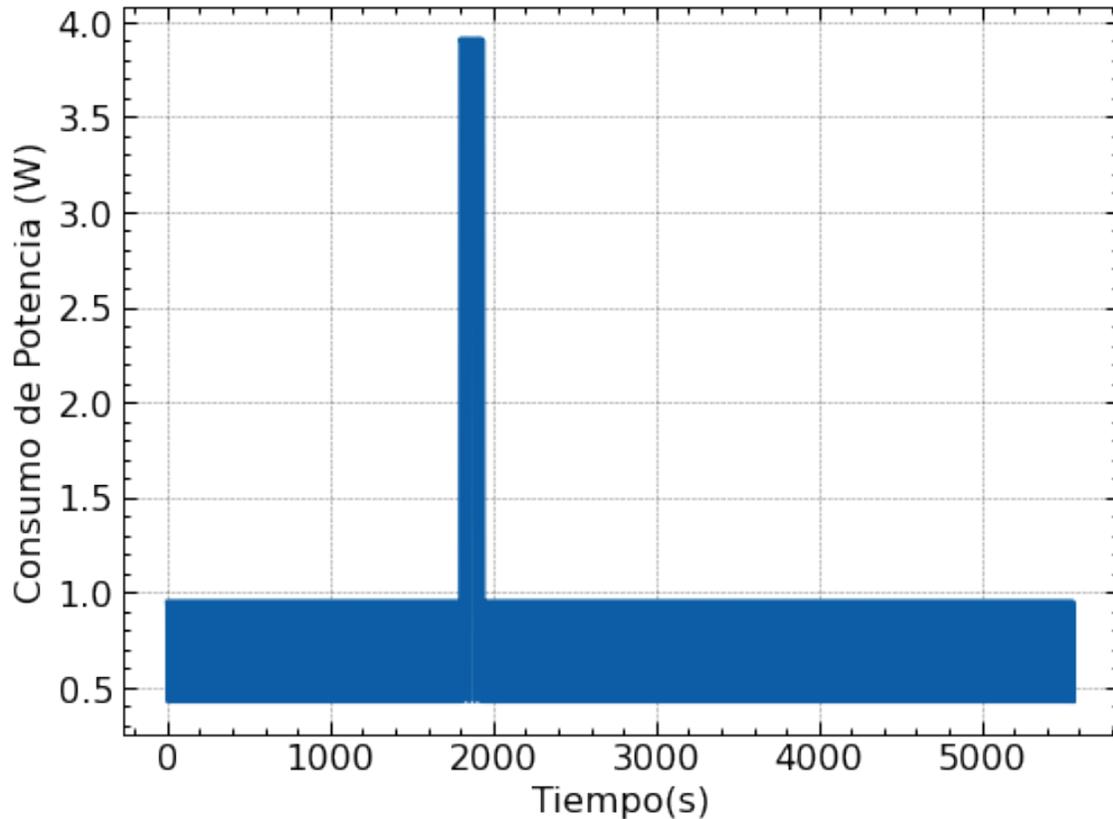
```

```
[5564 rows x 11 columns]
```

```
[ ]: sns.lineplot(x = 'time(s)', y='Power Consumption(W)',data= df);
plt.ylabel('Consumo de Potencia (W)')
plt.xlabel('Tiempo(s)')

# close = df[df['time(s)'] < 30]
# sns.lineplot(x = 'time(s)', y='EPS(W)',data= close);
# plt.xlabel('Tiempo(s)')
```

```
[ ]: Text(0.5, 0, 'Tiempo(s)')
```

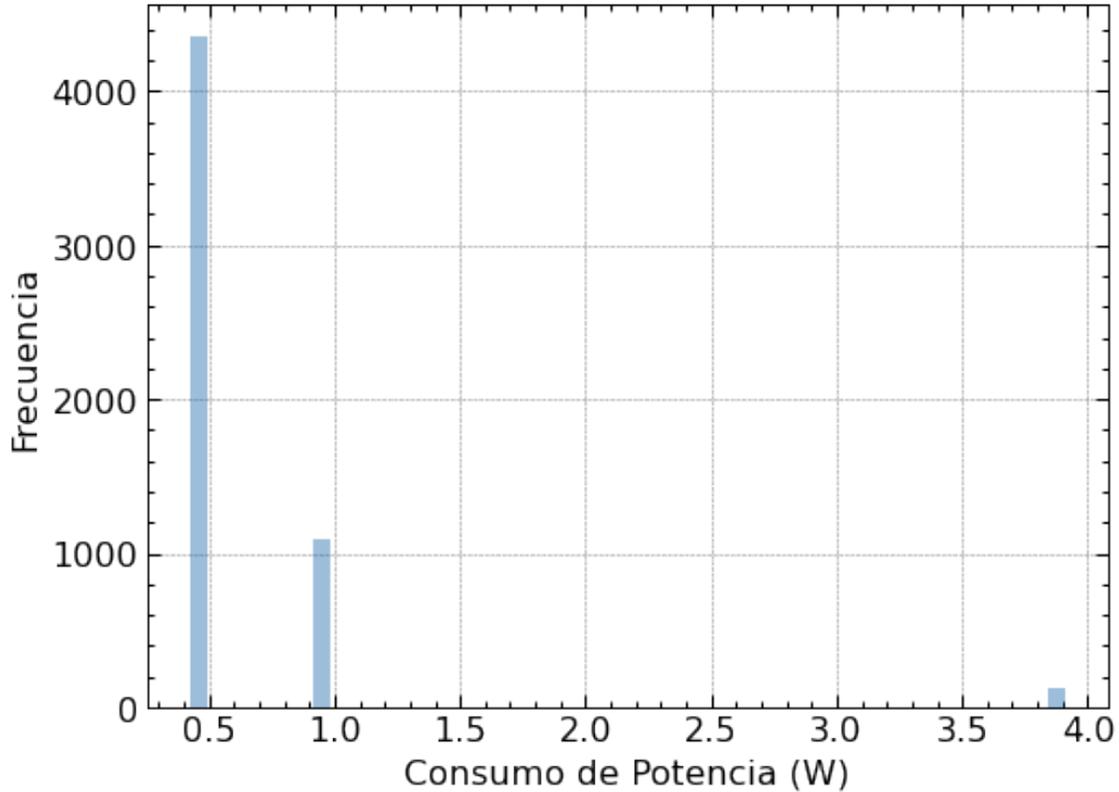


```
[ ]: sns.distplot(a=df['Power Consumption(W)'], label="Power Consumption(W)", kde =_
↪False)
plt.xlabel('Consumo de Potencia (W)')
plt.ylabel('Frecuencia')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



```
[ ]: summary = df['Power Consumption(W)'].describe()
summary
```

```
[ ]: count    5564.000000
mean         0.605588
std          0.539104
min          0.425818
25%          0.425818
50%          0.425818
75%          0.425818
max          3.908908
Name: Power Consumption(W), dtype: float64
```

#Solar panels power generation simulator

```
[ ]: column_names0 = [
    'Time (s)',
    'Orbit Angle (°)',
    'Rot Z (°)',
    'Latitud (°)',
```

```

'Nadier (°)',
'Sun Angle Top(°)',
'Sun Angle Bottom(°)',
'Sun Angle FPP1(°)',
'Sun Angle Side1(°)',
'Sun Angle FPP2(°)',
'Sun angle Side2(°)',
'Power Top(W)',
'Power Bottom(W)',
'Power FPP1(W)',
'Power Side1(W)',
'Power FPP2(W)',
'Power Side2(W)',
'Total Generation(W)'
]
df0 = pd.DataFrame(columns = column_names0)

```

```

[ ]: Time = [] #1
Orbit_Angle = [] #2
Rot_Z = [] #3
Latitud = [] #4
Nadier = [] #5
Sun_Angle_Top = [] #6
Sun_Angle_Bottom = [] #7
Sun_Angle_FPP1 = [] #8
Sun_Angle_Side1 = [] #9
Sun_Angle_FPP2 = [] #10
Sun_angle_Side2 = [] #11
Power_Top = [] #12
Power_Bottom = [] #13
Power_FPP1 = [] #14
Power_Side1 = [] #15
Power_FPP2 = [] #16
Power_Side2 = [] #17

counter = 0
counter_orbit = 1
n_orbit = 16
total_orbit = n_orbit * Orbital_periodo

while counter <= total_orbit:
    p_z = 1.2236 + counter * w # rotational angle for z axis global
    p_z_cubesat = rot_z_rad * counter # rotational angle for z axis of the CubeSat
    CubeSat_r = Orbit_z(CubeSat,p_z) # CubeSat is rotated from z axis global
    CubeSat_r = Orbit_x(CubeSat_r,inclination_ISS) # CubeSat is rotated from x_
    ↪axis global

```

```

lat = Latitud_funcion(CubeSat_r) # latitud angle of the location of the
↳cubesat for nadier angle [rad]
angle_nadier = Nadier_function(lat) # Nadier angle of the CubeSat [rad]
Time.append(counter) # the time is entered into the matrix
Orbit_Angle.append(round(math.degrees(p_z),2)) # position is entered into the
↳matrix
Rot_Z.append(round(math.degrees(p_z_cubesat),2)) # rotation of the cubesat is
↳entered into de matrix
Latitud.append(round(math.degrees(lat),2)) # latitud of the cubesat is
↳entered into de matrix
Nadier.append(round(math.degrees(angle_nadier),2)) # nadier angle of the
↳cubesat is entered to de matrix
if not (p_z >= (1.2236 + 2 * math.pi * (counter_orbit-1)) and (p_z <= 5.0595
↳+ 2* math.pi *(counter_orbit - 1))):
    Sun_Angle_Top.append(0)
    Sun_Angle_Bottom.append(0)
    Sun_Angle_FPP1.append(0)
    Sun_Angle_Side1.append(0)
    Sun_Angle_FPP2.append(0)
    Sun_angle_Side2.append(0)
    Power_Top.append(0)
    Power_Bottom.append(0)
    Power_FPP1.append(0)
    Power_Side1.append(0)
    Power_FPP2.append(0)
    Power_Side2.append(0)
else:
# CubeSat its rotated form x axis beacuse the Magnetic Field Vector and Local
↳Nadier
    SP_top_r = Orbit_x(SP_top,-angle_nadier) # Solar Panel Top is rotated from
↳x axis
    SP_bottom_r = Orbit_x(SP_bottom,-angle_nadier)# Solar Panel Bottom is
↳rotated from x axis
    SP_fpp1_r = Orbit_x(SP_fpp1,-angle_nadier) # Solar Panel FPP1 is rotated
↳from x axis
    SP_fpp2_r = Orbit_x(SP_fpp2,-angle_nadier) # Solar Panel FPP2 is rotated
↳from x axis
    SP_side1_r = Orbit_x(SP_side1,-angle_nadier) # Solar Panel Side1 is rotated
↳from x axis
    SP_side2_r = Orbit_x(SP_side2,-angle_nadier) # Solar Panel Side2 is rotated
↳from x axis
# CubeSat its rotated from z axis beacuse the rotation angular speed z given by
↳a pasive ADCS
    SP_top_r = Orbit_z(SP_top_r,p_z_cubesat) # Solar Panel Top is rotated from
↳z axis

```

```

    SP_bottom_r = Orbit_z(SP_bottom_r,p_z_cubesat) # Solar Panel Bottom is
↳rotated from z axis
    SP_fpp1_r = Orbit_z(SP_fpp1_r,p_z_cubesat) # Solar Panel FPP1 is rotated
↳from z axis
    SP_fpp2_r = Orbit_z(SP_fpp2_r,p_z_cubesat) # Solar Panel FPP2 is rotated
↳from z axis
    SP_side1_r = Orbit_z(SP_side1_r,p_z_cubesat) # Solar Panel Side1 is rotated
↳from z axis
    SP_side2_r = Orbit_z(SP_side2_r,p_z_cubesat) # Solar Panel Side2 is rotated
↳from z axis
# finding the angle between the panel and the sun's rays
    a_SP_top = Angle_sun(SP_top_r) # Solar Panel Top is rotated from z axis
    Sun_Angle_Top.append(math.degrees(a_SP_top)) # SOLAR ANGLE PANEL TOP
    a_SP_bottom = Angle_sun(SP_bottom_r) # Solar Panel Bottom is rotated from z
↳axis
    Sun_Angle_Bottom.append(math.degrees(a_SP_bottom)) # SOLAR ANGLE PANEL
↳BOTTOM
    a_SP_fpp1 = Angle_sun(SP_fpp1_r) # Solar Panel FPP1 is rotated from z axis
    Sun_Angle_FPP1.append(math.degrees(a_SP_fpp1)) # SOLAR ANGLE PANEL FPP1
    a_SP_side1 = Angle_sun(SP_side1_r) # Solar Panel Side1 is rotated from z
↳axis
    Sun_Angle_Side1.append(math.degrees(a_SP_side1)) # SOLAR ANGLE PANEL SIDE1
    a_SP_fpp2 = Angle_sun(SP_fpp2_r) # Solar Panel FPP2 is rotated from z axis
    Sun_Angle_FPP2.append(math.degrees(a_SP_fpp2)) # SOLAR ANGLE PANEL FPP2
    a_SP_side2 = Angle_sun(SP_side2_r) # Solar Panel Side2 is rotated from z
↳axis
    Sun_angle_Side2.append(math.degrees(a_SP_side2)) # SOLAR ANGLE PANEL SIDE2
# finding if top solar panel is pointing to the sun or not
    if (math.degrees(a_SP_top) > 89) or ((1368 * area * 2
↳*Eff_function(a_SP_top) * temp_fac) < 0):
        Power_Top.append(0) # power generated from top solar panel is entered
↳into the matrix
    else:
        Power_Top.append(1368 * area * 2 *Eff_function(a_SP_top) * temp_fac)
# finding if bottom solar panel is pointing to the sun or not
    if (math.degrees(a_SP_bottom) > 89) or ((1368 * area *
↳Eff_function(a_SP_bottom) * temp_fac) < 0):
        Power_Bottom.append(0) # power generated from bottom solar panel is
↳entered into the matrix
    else:
        Power_Bottom.append(1368 * area * Eff_function(a_SP_bottom) * temp_fac)
# finding if fpp1 solar panel is pointing to the sun or not
    if math.degrees(a_SP_fpp1) > 89 or ((1368 * area * 4
↳*Eff_function(a_SP_fpp1) * temp_fac) < 0):
        Power_FPP1.append(0) # power generated from fpp1 solar panel is entered
↳into the matrix

```

```

else:
    Power_FPP1.append(1368 * area * 4 *Eff_function(a_SP_fpp1) * temp_fac)
# finding if fpp2 solar panel is pointing to the sun or not
    if math.degrees(a_SP_fpp2) > 89 or ((1368 * area * 4 *
↳*Eff_function(a_SP_fpp2) * temp_fac) < 0):
        Power_FPP2.append(0) # power generated from fpp2 solar panel is entered
↳into the matrix
    else:
        Power_FPP2.append(1368 * area * 4 *Eff_function(a_SP_fpp2) * temp_fac)
# finding if side1 solar panel is pointing to the sun or not
    if math.degrees(a_SP_side1) > 89 or ((1368 * area * 4 *
↳*Eff_function(a_SP_side1) * temp_fac) < 0):
        Power_Side1.append(0) # power generated from side1 solar panel is entered
↳into the matrix
    else:
        Power_Side1.append(1368 * area * 4 *Eff_function(a_SP_side1) * temp_fac)
# finding if side2 solar panel is pointing to the sun or not
    if math.degrees(a_SP_side2) > 89 or ((1368 * area * 4 *
↳Eff_function(a_SP_side2) * temp_fac) < 0):
        Power_Side2.append(0)
    else:
        Power_Side2.append(1368 * area * 4 * Eff_function(a_SP_side2) * temp_fac)

if counter == Orbital_periodo * counter_orbit:
    counter_orbit += 1
counter += 1

df0['Time (s)'] = Time
df0['Orbit Angle (°)'] = Orbit_Angle
df0['Rot Z (°)'] = Rot_Z
df0['Latitud (°)'] = Latitud
df0['Nadier (°)'] = Nadier
df0['Sun Angle Top(°)'] = Sun_Angle_Top
df0['Sun Angle Bottom(°)'] = Sun_Angle_Bottom
df0['Sun Angle FPP1(°)'] = Sun_Angle_FPP1
df0['Sun Angle Side1(°)'] = Sun_Angle_Side1
df0['Sun Angle FPP2(°)'] = Sun_Angle_FPP2
df0['Sun angle Side2(°)'] = Sun_angle_Side2
df0['Power Top(W)'] = Power_Top
df0['Power Bottom(W)'] = Power_Bottom
df0['Power FPP1(W)'] = Power_FPP1
df0['Power Side1(W)'] = Power_Side1
df0['Power FPP2(W)'] = Power_FPP2
df0['Power Side2(W)'] = Power_Side2
df0['Total Generation(W)'] = 0.75 * (df0['Power Top(W)'] + df0['Power
↳Bottom(W)'] + df0['Power FPP1(W)'] + df0['Power Side1(W)'] + df0['Power
↳FPP2(W)'] + df0['Power Side2(W)'])

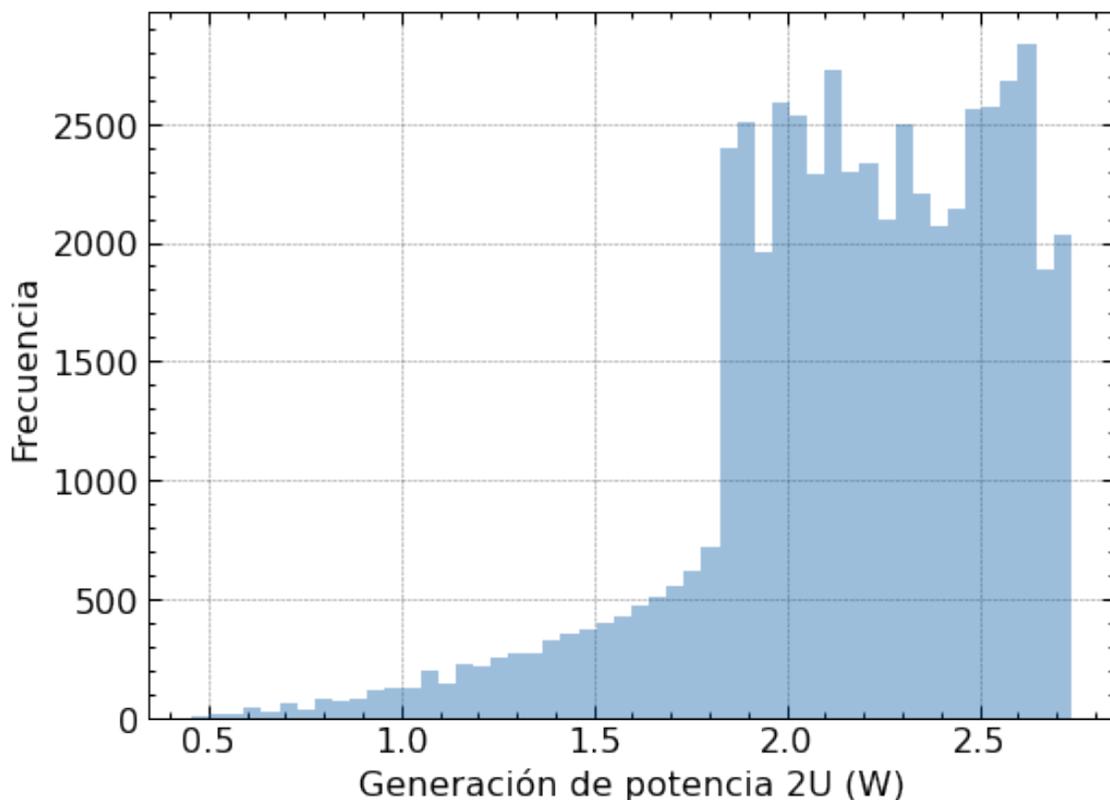
```

```
[ ]: zeros = df0[df0['Total Generation(W)'] != 0]
sns.distplot(a=zeros['Total Generation(W)'], label="Total Generation(W)", kde =_
↳False)
plt.xlabel('Generación de potencia 2U (W)')
plt.ylabel('Frecuencia')
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```

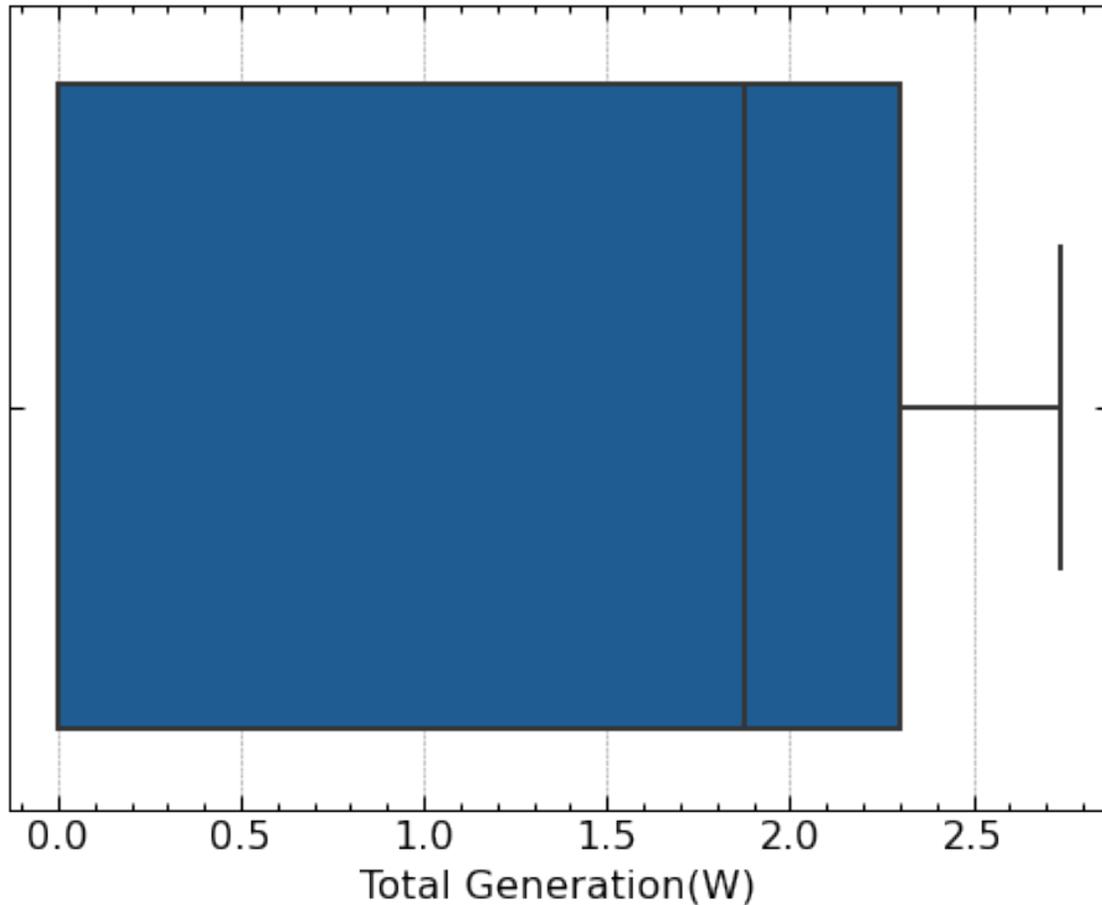


```
[ ]: sns.boxplot("Total Generation(W)", data = df0)
```

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only  
valid positional argument will be `data`, and passing other arguments without an  
explicit keyword will result in an error or misinterpretation.

```
FutureWarning
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6898bd40d0>
```



```
[ ]: df0['Total Generation(W)'].describe()
```

```
[ ]: count      89009.000000
      mean        1.324474
      std         1.100242
      min         0.000000
      25%         0.000000
      50%         1.878913
      75%         2.300783
      max         2.737431
      Name: Total Generation(W), dtype: float64
```

#State of Charge Analysis

```
[ ]: df1 = df0[["Time (s)", "Total Generation(W)"]]
      df1['Consumption(W)'] = pd.concat([df['Power Consumption(W)']] * n_orbit,
      ↪ ignore_index=True)
```

```

df1['Available Energy(W)'] = df1['Total Generation(W)'] - df1['Consumption(W)']
df1['Total Generation(Wh)'] = df1['Total Generation(W)']/3600
df1['Consumption(Wh)'] = df1['Consumption(W)']/3600
df1['Available Energy(Wh)'] = df1['Total Generation(Wh)'] -
↳df1['Consumption(Wh)']
df1['Available (Ah)'] = df1['Available Energy(Wh)']/Va
soc = 30
socArray = []
#Iteraciones de estado de carga de las baterias.
for i in range(89009):
    if soc + ((df1['Available (Ah)'][i]/Qn) * 100) > 100:
        soc = 100
    elif soc + ((df1['Available (Ah)'][i]/Qn) * 100) <= 0:
        soc = 0
    else:
        soc = soc + ((df1['Available (Ah)'][i]/Qn) * 100)
    socArray.append(soc)
df1['SoC'] = socArray
df1 # Print the table with each column

```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:2:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

This is separate from the ipykernel package so we can avoid doing imports until

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:4:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

after removing the cwd from sys.path.

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:5:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
"""
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
import sys
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:19:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
[ ]:      Time (s)  Total Generation(W)  ...  Available (Ah)      SoC
0         0         1.132693  ...      0.000014  30.000345
1         1         1.223277  ...      0.000021  30.000860
2         2         1.307121  ...      0.000066  30.002514
3         3         1.411318  ...      0.000074  30.004364
4         4         1.491524  ...      0.000080  30.006364
...      ...      ...      ...
89004     89004         0.000000  ...     -0.000032  97.850384
89005     89005         0.000000  ...     -0.000032  97.849585
89006     89006         0.000000  ...     -0.000032  97.848785
89007     89007         0.000000  ...     -0.000032  97.847986
89008     89008         0.000000  ...     -0.000032  97.847187
```

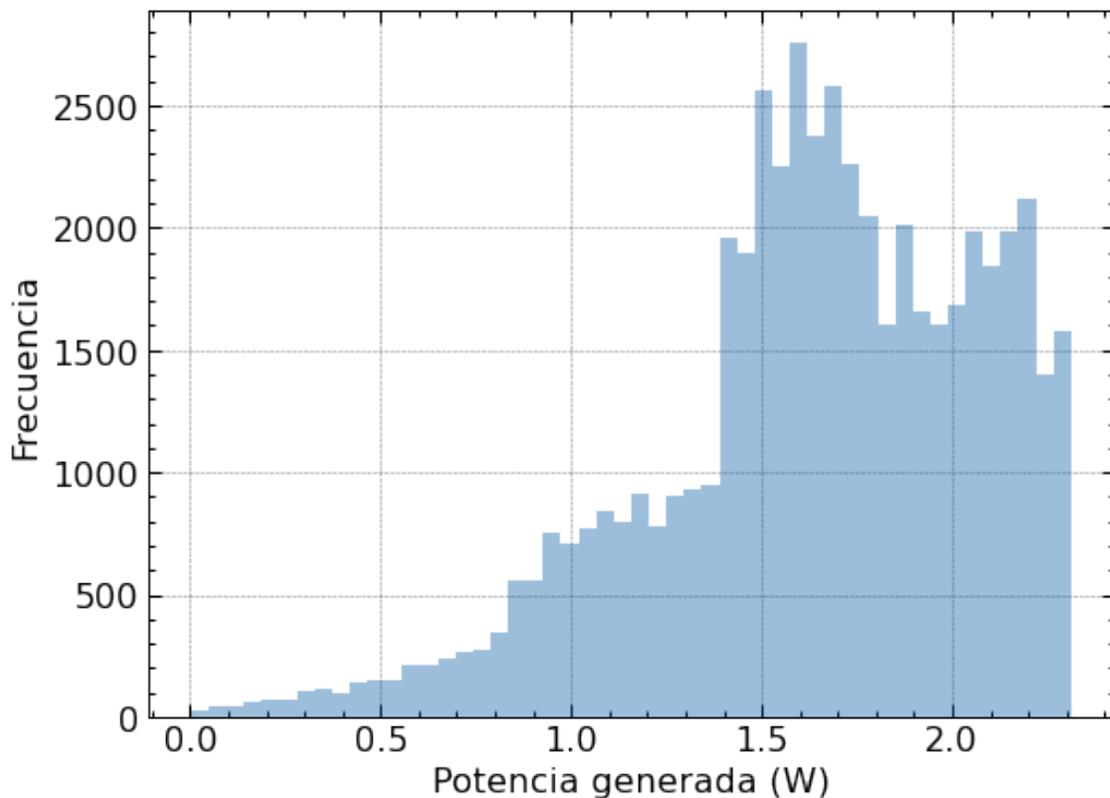
[89009 rows x 9 columns]

```
[ ]: power_generation = df1[df1['Available Energy(W)'] > 0]
sns.distplot(a=power_generation['Available Energy(W)'], label="Available_
↪Energy(W)", kde = False)
```

```
plt.xlabel('Potencia generada (W)')
plt.ylabel('Frecuencia')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

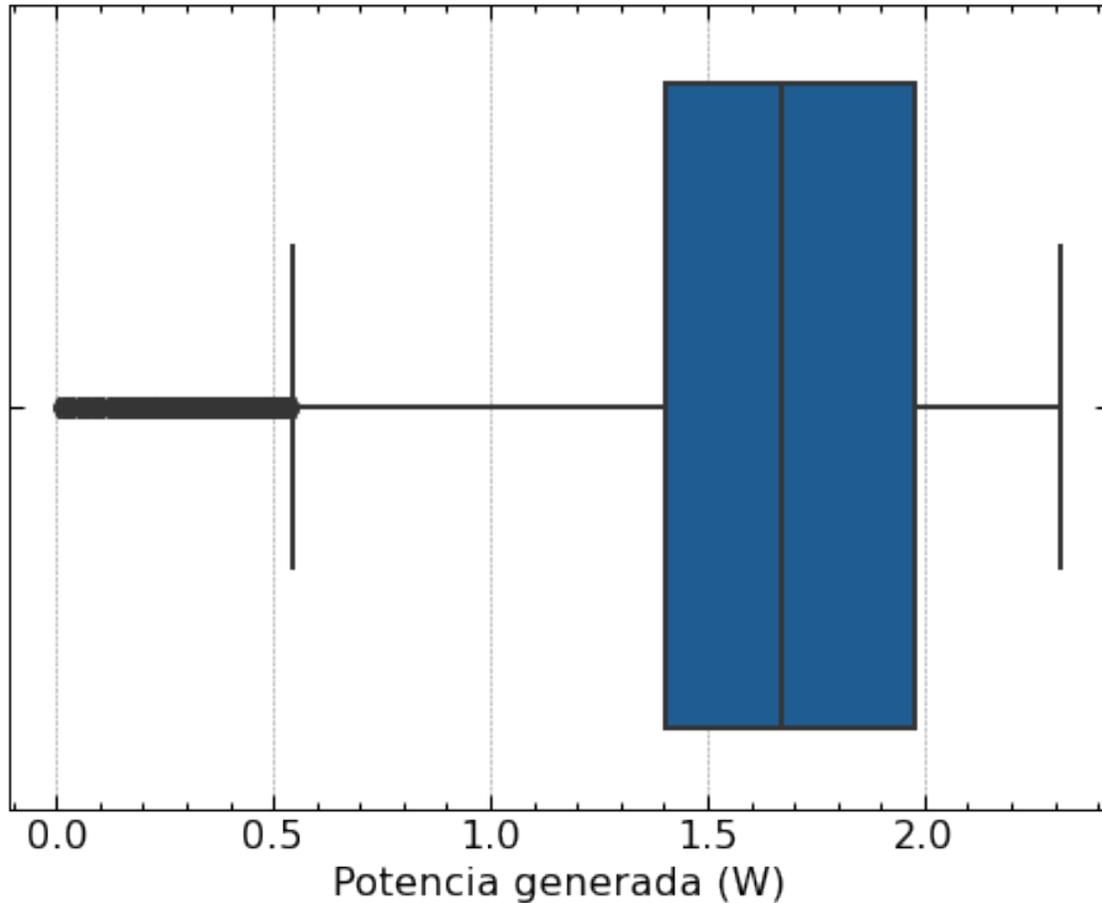
```
[ ]: Text(0, 0.5, 'Frecuencia')
```



```
[ ]: sns.boxplot("Available Energy(W)", data = power_generation)
plt.xlabel('Potencia generada (W)')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

```
[ ]: Text(0.5, 0, 'Potencia generada (W)')
```



```
[ ]: power_generation['Available Energy(W)'].describe()
```

```
[ ]: count    52246.000000
      mean      1.634330
      std       0.434840
      min       0.003698
      25%      1.404437
      50%      1.670514
      75%      1.976858
      max       2.311613
      Name: Available Energy(W), dtype: float64
```

```
[ ]: power_consumption = df1[df1['Available Energy(W)'] < 0]
      power_consumption['Available Energy(W)'] = power_consumption['Available_
      →Energy(W)'].abs()
      sns.distplot(a=power_consumption['Available Energy(W)'], label="Available_
      →Energy(W)", kde = False)
      plt.xlabel('Consumo de potencia disponible de las baterías (W)')
```

```
plt.ylabel('Frecuencia')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
```

```
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

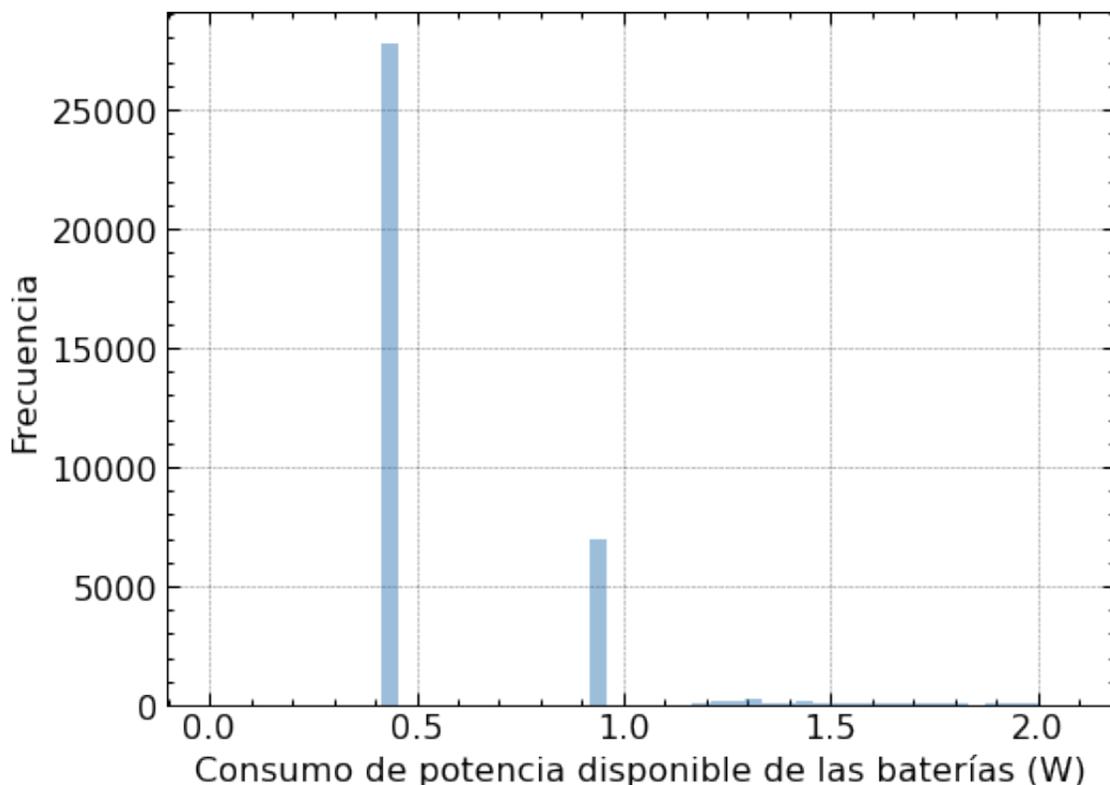
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
[ ]: Text(0, 0.5, 'Frecuencia')
```



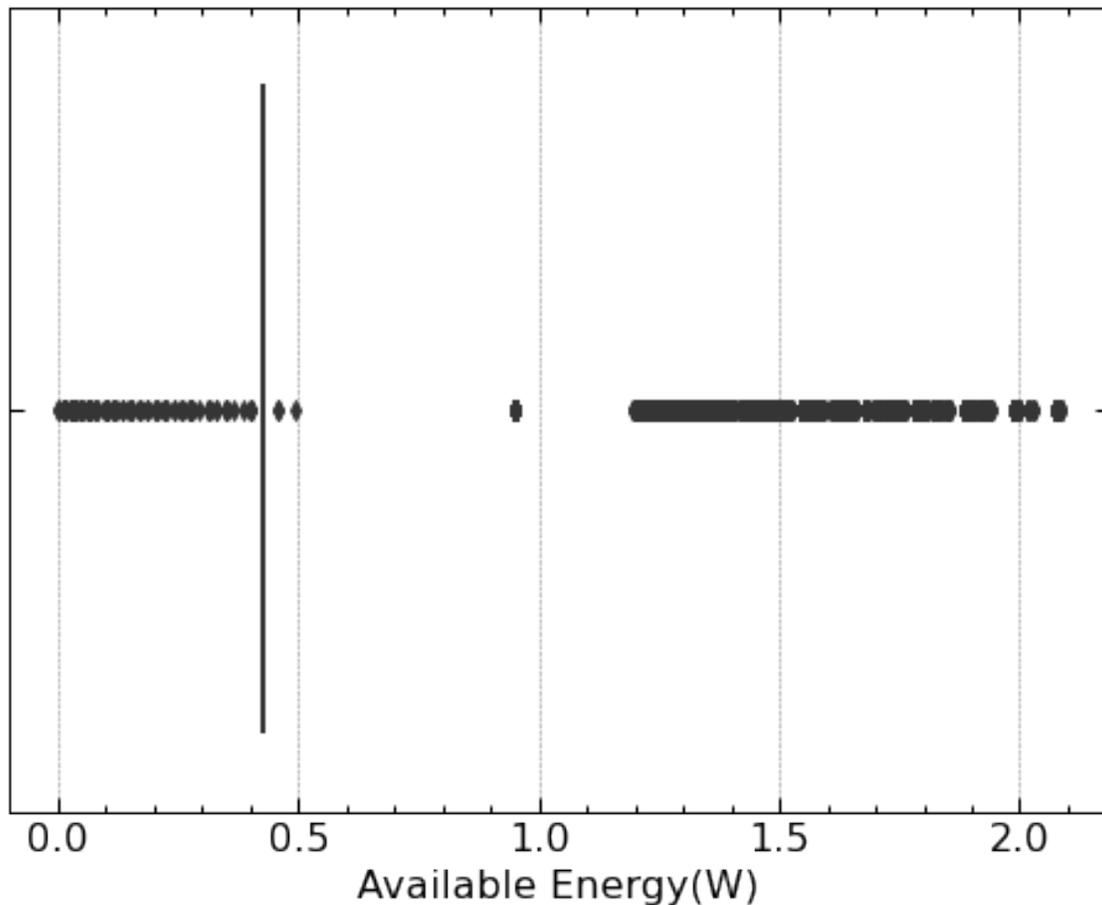
```
[ ]: sns.boxplot("Available Energy(W)", data = power_consumption)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only
```

valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f689891b9d0>
```

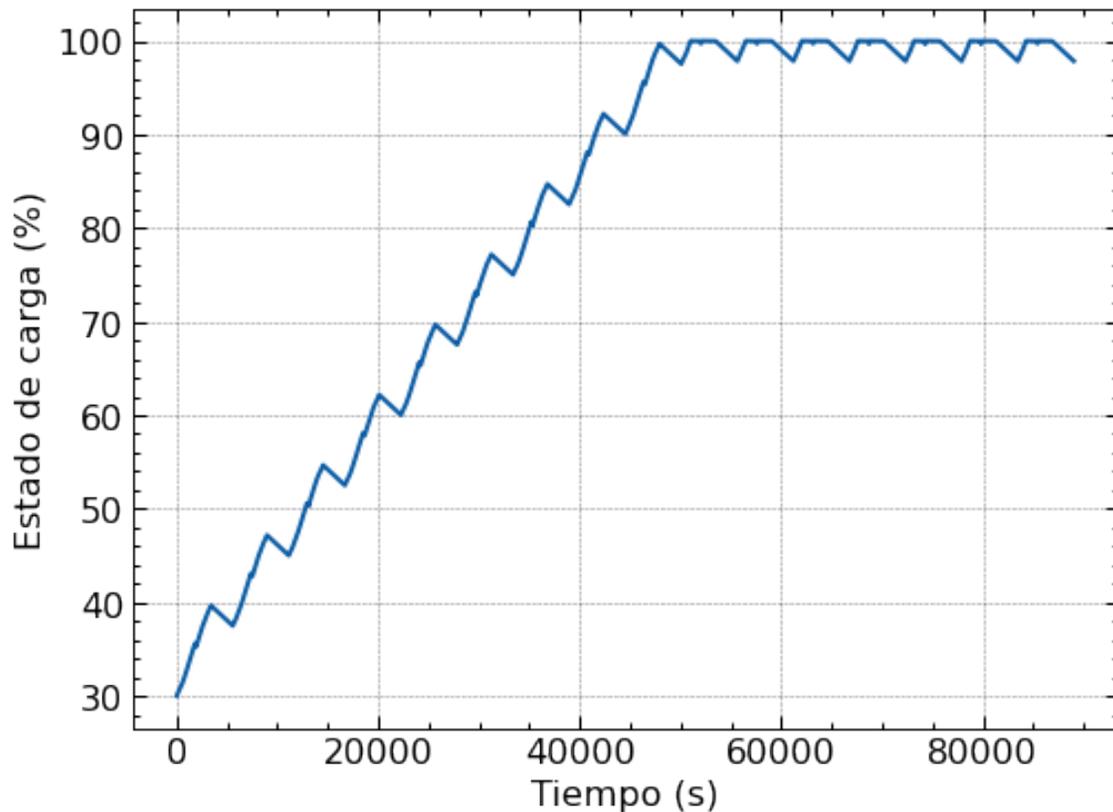


```
[ ]: power_consumption['Available Energy(W)'].describe()
```

```
[ ]: count      36763.000000
      mean        0.582122
      std         0.306330
      min         0.000120
      25%         0.425818
      50%         0.425818
      75%         0.425818
      max         2.083354
      Name: Available Energy(W), dtype: float64
```

```
[ ]: final_SoC = df1.SoC[89008]
SoC_difference = final_SoC - df1.SoC[0]
Average_SoC_change_per_orbit = SoC_difference / n_orbit
sns.lineplot(x = 'Time (s)', y='SoC',data= df1);
plt.xlabel('Tiempo (s)')
plt.ylabel('Estado de carga (%)')
```

```
[ ]: Text(0, 0.5, 'Estado de carga (%)')
```



```
[ ]: print(final_SoC)
print(SoC_difference)
print(Average_SoC_change_per_orbit)
print(df1['SoC'].max() - final_SoC)
```

```
97.84718689020245
67.8468419472284
4.240427621701775
2.152813109797549
```

