

	GESTIÓN DE RECURSOS Y SERVICIOS BIBLIOTECARIOS		Código	FO-GS-15
			VERSIÓN	02
	ESQUEMA HOJA DE RESUMEN		FECHA	03/04/2017
			PÁGINA	1 de 1
ELABORÓ		REVISÓ		APROBÓ
Jefe División de Biblioteca		Equipo Operativo de Calidad		Líder de Calidad

RESUMEN TRABAJO DE GRADO

AUTOR(ES):

NOMBRE(S): JEFFERSON ARONI APELLIDOS: GARCÉS TORRES

NOMBRE(S): _____ APELLIDOS: _____

FACULTAD: INGENIERÍA

PLAN DE ESTUDIOS: INGENIERÍA ELECTRÓNICA

DIRECTOR:

NOMBRE(S): DARWIN ORLANDO APELLIDOS: CARDOZO SARMIENTO

CO-DIRECTOR:

NOMBRE(S): SERGIO ALEXANDER APELLIDOS: CASTRO CASADIEGO

TÍTULO DEL TRABAJO (TESIS): SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES, DESDE LA NUBE

RESUMEN

Este proyecto se basó en un sistema de supervisión para las variables en los paneles solares desde la nube. Para ello, se implementó una investigación tipo descriptiva, ya que se enfocó en la búsqueda de información, la cual se obtuvo mediante estudios especializados relacionados con el tema, libros y bases tecnológicas halladas en Internet. Primero, se logró recopilar la información sobre monitorización remota. Posteriormente, se realizó la programación de la aplicación Web Responsive en Firebase con conexión a Raspberry Pi. Finalmente, se realizaron las pruebas al sistema de supervisión, presentando resultados y beneficios de nuevas tecnologías digitales en instrumentación electrónica.

PALABRAS CLAVE: Sistema de supervisión, Raspberry Pi, Web Responsive en Firebase, Paneles solares.

CARACTERÍSTICAS:

PÁGINAS: 164 PLANOS: ILUSTRACIONES: CD ROOM: 1

SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES,
DESDE LA NUBE

JEFFERSON ARONI GARCÉS TORRES

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
FACULTAD DE INGENIERÍA
PLAN DE ESTUDIOS DE INGENIERÍA ELECTRÓNICA
SAN JOSÉ DE CÚCUTA

2022

SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES,
DESDE LA NUBE

JEFFERSON ARONI GARCÉS TORRES

Trabajo de grado presentado como requisito para optar el título de:

Ingeniero Electrónico

Director:

I.E. M.Sc. DARWIN ORLANDO CARDOZO SARMIENTO

Codirector:

I.E. M.Sc. SERGIO ALEXANDER CASTRO CASADIEGO

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTRÓNICA

SAN JOSÉ DE CÚCUTA

2022

ACTA DE SUSTENTACIÓN DE UN TRABAJO DE GRADO

Fecha: CÚCUTA, 15 DE MARZO DE 2022

Hora: 17:00

Lugar: SALON SC 301

Plan de Estudios: INGENIERÍA ELECTRÓNICA

Título de la Tesis: "SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES, DESDE LA NUBE"

Jurados: ANDRÉS EDUARDO PÁEZ PEÑA IE.MSc
MARCO AURELIO GARCÍA BERMÚDEZ IE.MSc

Director: DARWIN ORLANDO CARDOZO SARMIENTO, IE MSc

Codirector: SERGIO ALEXANDER CASTRO CASADIEGO,
IE.MSc.

Nombre del Estudiante:	Código:	Calificación:	
		Número	Letra
JEFFERSON ARONI GARCÉS TORRES	1161069	4,5	Cuatro, cinco

MERITORIO


ANDRÉS EDUARDO PÁEZ PEÑA


MARCO AURELIO GARCÍA BERMÚDEZ


ANGEL JOSEPH SOTO VERGEL
Coordinador Comité Curricular
Ingeniería Electrónica

Dedicatoria

Em lembrança de minha amada avó, a saudade quando olho o céu; porque as palavras nunca terão o amor que ela me deu, só espero escolher o caminho certo, para que ela se sinta orgulhosa de mim.

Jefferson Garcés

Agradecimientos

Agradezco inicialmente a Dios, por mostrarme su existencia y participación en mi vida a cada paso que doy; a mi abuela Teresa Herrera, que hizo tanto por mí y me enseñó el valor de la educación, sus esfuerzos seguirán acompañándome por el resto de mi vida, dándome fuerza cuando sea necesario; agradezco a mí hijo que cambio mi vida, y me ayudo a enfocar todo de mí, en culminar con éxito este reto; a mi esposa, mi suegra y su familia que creyeron en mí desde que me conocieron; a mi madre Luz Magaly ama de casa, mi padre Fabio Garcés licenciado en educación y mi hermana Marcela una gran profesional, que estuvieron allí, a pesar de las distancias y los diferentes enfoques de vida que tenemos cada uno; a mí Tío Martín, su esposa Sofía e hijo, que más allá del océano apoyaron mi causa años tras año; a mí Tía Laine Yamile, su esposo Wilson, y sus hijos, que me tendieron la mano y creyeron en mí, cuando más lo necesitaba; a mí Tío Alonso, su esposa Sandra e hijos, que vieron mi crecimiento; a mis compañeros, socios y amigos, Brayan, Duvar y Alexis, que son mi segunda familia, y nunca dudaron que esto fuera posible; a mis amadas instituciones SENA, y UFPS, que me formaron como profesional y a su gran equipo de profesionales.

Contenido

	pág.
Introducción	20
1. Problema	21
1.1 Título	21
1.2 Planteamiento del Problema	21
1.3 Justificación	22
1.3.1 Impacto esperado	23
1.3.2 Beneficios tecnológicos	23
1.3.3 Beneficios económicos	24
1.3.4 Beneficios sociales	24
1.3.5 Beneficios institucionales	24
1.4 Alcance	25
1.4.1 Tipo de proyecto	25
1.4.2 Resultados esperados	25
1.4.2.1 Resultados directos	26
1.4.2.2 Resultados indirectos	26
1.5 Limitaciones	26
1.6 Delimitaciones	27
1.7 Objetivos	28
1.7.1 Objetivo general	28
1.7.2 Objetivos específicos	28
2. Marco Referencial	29
2.1 Antecedentes	29

2.2 Marco Teórico	33
2.2.1 Energía solar	33
2.2.2 Eficiencia de un sistema fotovoltaico	33
2.2.3 Instrumentación electrónica y adquisición de datos	34
2.2.4 Voltaje, corriente y potencia	34
2.2.5 Sensor de Voltaje FZ0430 0.02445V – 25V	35
2.2.6 Sensor de voltaje SKU DFR0051	35
2.2.7 El sensor de corriente KS0185	36
2.2.8 Conversión Análogo/Digital (ADC)	37
2.2.9 Sistemas embebidos	37
2.2.10 Arduino	38
2.2.11 Raspberry Pi	38
2.2.12 Open source	39
2.2.13 Tecnologías de la información	39
2.2.14 Protocolo de comunicación Web	40
2.2.15 Lenguajes de desarrollo Web Html, Css y JavaScript	40
2.2.16 Hosting	41
2.2.17 Base de datos o DataBase	41
2.2.18 Json	42
2.2.19 Python	42
2.2.20 Firebase	42
2.2.21 Aplicación Web	43
2.2.22 Sistemas en tiempo real	43
2.2.23 Internet de las cosas	43

2.2.24 Industria 4.0	44
2.3 Marco Legal	44
2.3.1 Ley 1715 de 2014 Congreso de Colombia	44
2.3.2 Ley 697 de 2001 Congreso de Colombia	45
3. Diseño Metodológico	46
3.1 Tipo de Metodología	46
3.2 Fases del Proyecto	46
3.2.1 Fase 1. Selección de componentes de hardware	46
3.2.1.1 Desarrollo	46
3.2.2 Fase 2. Diseño de un sistema de instrumentación electrónica no invasivo para la captación, conversión análogo-digital y centralización de los datos en Raspberry Pi, de las variables en paneles solares	47
3.2.2.1 Desarrollo	48
3.2.3 Fase 3. Diseño, configuración y programación de una aplicación Web Responsive en Firebase con conexión a Raspberry Pi como centro de supervisión, procesamiento y envío de datos al Servidor de la DataBase	51
3.2.3.1 Desarrollo	51
3.2.4 Fase 4. Pruebas del sistema de supervisión, respaldo de la data en la nube, y descarga remota de las variables en los paneles solares	53
3.2.4.1 Desarrollo	53
3.2.5 Fase 5. Presentar resultados y beneficios de nuevas tecnologías digitales en instrumentación electrónica y aplicaciones Web para industria 4.0	57
3.2.5.1 Actividades	57

4. Resultados	59
4.1 Recopilar Información	60
4.2 Diseño del Sistema de Instrumentación	63
4.2.1 Selección de sistema de captación de datos	65
4.2.1.1 Calificación del microcontrolador	67
4.2.1.2 Calificación de comunicación del microcontrolador	67
4.2.1.3 Calificación de canales análogos del microcontrolador	69
4.2.1.4 Calificación de disponibilidad del microcontrolador	70
4.2.1.5 Calificación de precio del microcontrolador	71
4.2.1.6 Selección de microcontrolador	72
4.2.2 Análisis de sensores	74
4.2.2.1 Simulación de captación de datos	78
4.2.2.2 Configuración de Raspberry Pi con base de datos	80
4.2.3 Plataforma de backend	82
4.2.3.1 Conexión de Raspberry pi 3B con Firebase	86
4.3 Diseño de aplicación Web	89
4.3.1 Diseño de HTML	91
4.3.2 Diseño de CSS	93
4.3.3 Diseño de JavaScript	96
4.3.4 Conexión del Frontend con el Backend	98
4.4 Implementación del Prototipo	103
4.4.1 Implementación de la instrumentación	104
4.5 Divulgación de Resultados y Beneficios de Nuevas Tecnologías en Industria 4.0	125

5. Conclusiones	135
6. Recomendaciones	138
Referencias Bibliográficas	139
Anexos	146

Lista de Figuras

	pág.
Figura 1. Sensor de voltaje FZ0430 0.02445V - 25V	35
Figura 2. Sensor de voltaje SKU DFR0051	36
Figura 3. Sensor de corriente KS0185	37
Figura 4. Placas de desarrollo Arduino	38
Figura 5. Raspberry Pi 3B+	39
Figura 6. Descripción general de la arquitectura FCM en Firebase	43
Figura 7. Proceso de captación de datos por medio de la instrumentación	48
Figura 8. Esquemático de conexión del sensor de voltaje FZ0430	49
Figura 9. Instalación y configuración del sistema operativo en Raspberry Pi	50
Figura 10. Descarga de las dependencias de Raspberry Pi y creación de archivo Pitón	51
Figura 11. Brackets, editor de texto Open Source	52
Figura 12. Plataforma de desarrollo Backend, para aplicaciones Web	52
Figura 13. Proceso de los datos desde su captación, hasta la DataBase	54
Figura 14. Conexión y envío de datos desde la Raspberry Pi a la DataBase de Firebase	55
Figura 15. Proceso de interacción del usuario con la aplicación Web	56
Figura 16. Interacción y descarga remota de datos, desde aplicación Web	57
Figura 17. Filtro de aplicación Web para comparativas de Mouser Electronics ®	65
Figura 18. Circuito mínimo necesario para el funcionamiento del SCT013 YHDC	75
Figura 19. Circuito lector de corriente del KS0185	76
Figura 20. Circuito de sensores de voltaje, FZ0430 y SKU DFR0051	76
Figura 21. Código en Arduino para la lectura analógica de los sensores de voltaje	77
Figura 22. Modelo de diodo sencillo, con resistencia Shunt, y dependencia de la radiación	77

Figura 23. Lectura analógica del sensor de voltaje	78
Figura 24. Sensor de corriente KS0185	79
Figura 25. Código en Arduino para la lectura analógica del sensor de corriente	79
Figura 26. Lectura analógica del sensor de corriente	80
Figura 27. Creación de entorno Linux, para SO Debian de Raspberry Pi	81
Figura 28. Actualización de Raspberry Pi	81
Figura 29. Actualización manual de Raspberry Pi	82
Figura 30. Creación de proyecto en consola de Firebase	83
Figura 31. Consola del App, para los servicios de Firebase	84
Figura 32. Creación de espacio físico y protocolo de seguridad de Realtime DataBase	85
Figura 33. Generación de SDK para la administración de la base de datos	85
Figura 34. Advertencia de descarga de las credenciales	86
Figura 35. Desinstalación de la versión 2.7 de Python en la Raspberry Pi	87
Figura 36. Instalación del gestor de paquetes de instalación python3-pip de Raspberry Pi	87
Figura 37. Instalación de firebase-admin para acceso a Firebase desde la Raspberry Pi	88
Figura 38. Conexión de Raspberry Pi con Firebase	89
Figura 39. Inicio de la interfaz de la aplicación en HTML	91
Figura 40. Ventanas de interfaz gráfica de la App en HTML	92
Figura 41. Inicio de la interfaz gráfica de la App con CSS	93
Figura 42. Formatos Responsive de la interfaz gráfica de la aplicación con CSS.	95
Figura 43. Interfaz en diferentes pantallas de diferentes dispositivos	95
Figura 44. Ventanas de interfaz gráfica con CSS	96
Figura 45. Formulario de inicio de sesión del lado del cliente, con JavaScript	98
Figura 46. Conexión con el servidor de Firebase	99

Figura 47. Visualización desde la interfaz de los datos de Realtime DataBase	100
Figura 48. Configuración de los servicios de autenticación de Firebase	101
Figura 49. Gestión de usuarios con acceso a la aplicación	102
Figura 50. Descarga de histórico de datos	103
Figura 51. Componentes del prototipo	104
Figura 52. Panel fotovoltaico con resistencia en serie	105
Figura 53. Medición para la calibración de sensor de voltaje FZ0430	106
Figura 54. Comparación de calibración de sensor de voltaje FZ0430	107
Figura 55. Medición para la calibración de sensor de voltaje SKU SDFR 0051	108
Figura 56. Comparación de calibración de sensor de voltaje SKU DFR0051	108
Figura 57. Lectura de corriente con el sensor KS0185	109
Figura 58. Lectura de voltajes de calibración con el sensor KS0185	110
Figura 59. Lectura de corriente calibrada del sensor KS0185	111
Figura 60. Integración y calibración de la instrumentación	112
Figura 61. Comprobación de calibración de los sensores de voltaje	112
Figura 62. Comprobación de la calibración del sensor de corriente	113
Figura 63. Comprobación de la calibración del sensor de corriente	114
Figura 64. Conexión y envío de datos con Realtime DataBase desde la Raspberry Pi	114
Figura 65. Pruebas de funcionamiento en sitio del prototipo de supervisión remota	115
Figura 66. Comparativa de formatos de almacenamiento de histórico de datos	116
Figura 67. Organización de las variables para tabulación en el archivo Excel	116
Figura 68. Tabulación de datos de un día de lectura	117
Figura 69. Instalación de herramientas de despliegue de Firebase	118
Figura 70. Autorización de permisos del Firebase CLI	118

Figura 71. Inicio de sesión del Firebase CLI, en la cuenta de Gmail, para el despliegue	119
Figura 72. Opciones de despliegue en Hosting de Firebase	120
Figura 73. Despliegue de la aplicación Web en el Hosting	120
Figura 74. Lectura de datos en tiempo real, en aplicación Web desde la nube	121
Figura 75. Encuesta de evaluación de App Web	122
Figura 76. Resultados de primera pregunta	122
Figura 77. Resultados de segunda pregunta	123
Figura 78. Resultados tercera y cuarta pregunta	124
Figura 79. Resultados de quinta y sexta	124
Figura 80. Diagrama de conexión del sistema de supervisión	125
Figura 81. Mensaje de aceptación, y horario de presentación de la ponencia en el iii congreso internacional de investigación aplicada, creación e innovación: desarrollo sostenible y tendencias tecnológicas	126
Figura 82. Certificado de participación	127
Figura 83. Trasmisión en vivo, en Café de la mañana, del Canal TRO	128
Figura 84. Entrevista en vivo, en Café de la mañana, del canal TRO	128
Figura 85. Video de la entrevista en vivo, en Café de la mañana, del canal TRO, en el canal de YouTube	129
Figura 86. Tercera versión de la feria Cúcuta Digital Valley 2021	130
Figura 87. Stand para exposición del prototipo, en la tercera versión de la feria Cúcuta Digital Valley 2021	131
Figura 88. Presentación del prototipo a los funcionarios del Tecnoparque Nodo Cúcuta, gerente de CENS y asesores	132
Figura 89. Concurso de la segunda versión, INNOVACION 2021, CENS	133

Figura 90. Presentación del proyecto enfocado en la segunda versión, INNOVACION 2021, CENS	134
Figura 91. Final de segunda versión, INNOVACCION 2021, CENS	134

Lista de Tablas

	pág.
Tabla 1. Calificación de puntuación de viabilidad	63
Tabla 2. Calificación de microcontrolador	67
Tabla 3. Valorización de comunicación de los microcontroladores	68
Tabla 4. Calificación de comunicaciones de los microcontroladores	68
Tabla 5. Valorización de los canales análogos de los microcontroladores	69
Tabla 6. Calificación de los canales ADC	70
Tabla 7. Valorización de disponibilidad de los microcontroladores	70
Tabla 8. Calificación de disponibilidad de los microcontroladores	71
Tabla 9. Valorización de disponibilidad de los microcontroladores	71
Tabla 10. Calificación de disponibilidad de los microcontroladores	72
Tabla 11. Selección de sistema de captación de datos	73
Tabla 12. Datos de los sensores	74

Lista de Anexos

	pág.
Anexo 1. Manual técnico	147

Resumen

Este proyecto se basó en un sistema de supervisión para las variables en los paneles solares, desde la nube. Para ello, se implementó una investigación tipo descriptiva, ya que enfocó en la búsqueda de información y diseño de la instrumentación, para la captación y almacenamiento de datos. La información se obtuvo mediante estudios especializados relacionados con el tema, libros y bases tecnológicas halladas en Internet. Se logró desarrollar un prototipo de supervisión desde la nube, para las variables en los paneles solares. Seguidamente, se recopiló la información sobre monitorización remota, instrumentación, captación y respaldo de los datos de variables en paneles solares, la cual permitió diseñar el sistema de instrumentación electrónica no invasivo para la captación, conversión análogo-digital. Posteriormente, se realizó la programación de la aplicación Web Responsive en Firebase con conexión a Raspberry Pi como centro de supervisión, procesamiento y envío de datos al Servidor de la DataBase. Finalmente, se realizaron las pruebas al sistema de supervisión, respaldo de la data en la nube y descarga remota de las variables en los paneles solares, presentando resultados y beneficios de nuevas tecnologías digitales en instrumentación electrónica y aplicaciones Web para industria 4.0.

Introducción

El avance en Energía Solar es de gran interés para el Gobierno Nacional de Colombia, el cual considera la viabilidad de sistemas fotovoltaicos como alternativa al subsidio existente para los estratos 1, 2 y 3 en el consumo de energía eléctrica (Torres, 2016). En Cúcuta, Norte de Santander, la Universidad Francisco de Paula Santander (UFPS), se encuentra en desarrollo de sistemas fotovoltaicos, que obtengan mayor eficiencia, de los sistemas ya existentes. Con el uso de nuevas tecnologías y basados en el análisis de los datos tomados, buscan desarrollar nuevas técnicas que aumenten la conversión de la energía fotovoltaica en energía eléctrica aprovechable (Cepeda & Sierra, 2017).

Al integrar las tecnologías de la información (TIC), y sistemas embebidos Open Source de bajo coste, se diseñó y conectó, la electrónica de adquisición de datos, a los protocolos de comunicación Web, para visualizar y descargar la información en tiempo real. Los investigadores pueden tener acceso, a la información de los datos de las variables en los paneles de forma rápida y segura por medio de una aplicación Web, y realizar el acceso, con cualquier dispositivo conectado a Internet; para con ello concentrar sus recursos en el tratamiento de los datos y concluir sus investigaciones (Orellana & Sánchez, 2006).

En el presente trabajo se explica la utilidad del uso de nuevas tecnologías de la información y sistemas embebidos de bajo coste, como una solución viable a las instituciones que necesitan conectar su instrumentación a la nube, para realizar supervisión y respaldo de la información en tiempo real, desde cualquier dispositivo conectado a Internet.

1. Problema

1.1 Título

SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES, DESDE LA NUBE.

1.2 Planteamiento del Problema

La comunidad científica de la UFPS, tiene un proyecto para determinar el ángulo de mayor eficiencia, en los paneles de los sistemas fotovoltaicos instalados en la ciudad de Cúcuta. La mayor limitante es la supervisión remota, ya que, no hay una instrumentación que integre las diferentes variables de los paneles con una base de datos en la nube; y los dispositivos que hay en el mercado, solo centralizan datos de una instrumentación ya existente (por lo general del mismo fabricante), y los comunica con el servidor de la misma, obligando a la institución un pago periódico para el acceso a la información.

La medición de las variables se realiza por el Data Logger HOBO® 4-Channel Analog (UX120-006M) (ONSETM 2018), con disponibilidad de 4 canales y una resolución de 16 bits para registrar hasta 1,9 millones de eventos; permite la conexión de sensores de voltaje con un rango de medida de 0 - 20.1mA a una resolución de 0.3 μ y una temperatura entre 0° y 50°C (32° a 122°F). Un Data Logger muy potente en sitio, pero limitado en una toma de datos remota, e incapaz de conexión WIFI para el envío y respaldo de los datos en la nube, en tiempo real.

Por lo tanto, la toma de datos de las variables de los paneles solares de la UFPS es de difícil acceso remoto, y obliga a los investigadores a movilizarse al sitio y extraer la información por medio de memorias externas. El problema es que, no se tiene acceso a la información en tiempo

real, ni se respaldada inmediatamente en una base de datos; lo que obliga a un investigador a esta labor, consumiendo recurso humano y tecnológico, que puede ser aprovechado para el desarrollo de la investigación en curso, si se tuviera acceso a la información remotamente por parte de todos los investigadores.

Considerando la problemática anteriormente expuesta, se plantea la siguiente pregunta:

¿Cómo supervisar variables en paneles solares en tiempo real y con respaldo de la información?

1.3 Justificación

Para determinar si un sistema fotovoltaico es eficiente o no, se evalúa la potencia entregada a una carga (Halcafree, 2016). Al medir el voltaje junto con el valor resistivo de la carga, se determina la corriente, y usando la ley de Watts, se determina la potencia. Estos cálculos solo son posibles al tener acceso a los datos que actualmente son centralizados en el HOBO® Data Logger (UX120-006M).

La monitorización y descarga histórica de la información, no cuenta con una transmisión directa de datos a la nube, lo que impide el acceso y respaldo de la información en tiempo real. Esta limitante destina a un investigador a esta labor, consumiendo recurso humano y tecnológico en el proceso; siendo así necesario, el desarrollo de un prototipo funcional que permita la instrumentación, centralización, respaldo en la nube, y visualización de datos en tiempo real, desde cualquier dispositivo, con conexión a Internet para los diferentes investigadores.

Es de vital importancia que la instrumentación electrónica, sea de fácil conversión Análogo/Digital para la medición de variables analógicas en el microprocesador BCM2837 ARMv8, que tiene conexión LAN inalámbrica 802.11 ac bajo tecnología Raspberry Pi 3B

(Halcafree, 2016).

Este microprocesador sirvió de centro de datos y servidor, permitiendo la conexión directa a Internet, con un ejecutable desarrollado en Python. El envío de datos se realizó a una plataforma de Backend, con DataBase asíncrona, para su posterior visualización y descarga, desde una aplicación Web en cualquier dispositivo.

1.3.1 Impacto esperado. En la finalización de este proyecto, se obtuvo un prototipo que mejoró la toma y respaldo de datos de las variables en los paneles solares, dando a los investigadores acceso a la información desde cualquier dispositivo con acceso a Internet, y evitando la movilización al sitio, para liberar recursos humanos y tecnológicos, concentrándolos en el tratamiento de datos, y no en la captación de los mismos. También se aumentó la cantidad de canales de medida con un solo sistema de instrumentación, lo que redujo significativamente la inversión en el estudio de nuevos paneles solares, y con esto, se reconoció a la institución como pionera en investigación de tecnologías de industria 4.0 para la región, en el uso de aplicaciones Web para la monitorización remota de variables en tiempo real.

1.3.2 Beneficios tecnológicos. A continuación, se muestran los beneficios tecnológicos:

- Se desarrolló instrumentación no invasiva, de bajo coste y de fácil instalación, que obtiene resultados cercanos a la electrónica actual y tiene una mayor cantidad de canales de medición.
- Se integró nuevas tecnologías de la información, a la instrumentación electrónica que mide las variables de los paneles en tiempo real.
- Se centralizó, procesó y envió los datos a un Servidor con una aplicación Web, para el acceso a la información desde cualquier dispositivo con conexión a Internet.

1.3.3 Beneficios económicos. Los beneficios económicos se presentan a continuación:

- Se ahorró la compra de costosos Datalogger nuevos con tecnología IoT para el envío de datos a la nube.
- Se redujo el costo de implementación para la instrumentación y centralización de datos, de nuevos paneles solares.
- Se redujo la cantidad y compra de instrumentos de centralización de datos, al tener una mayor cantidad de canales de medida en un solo ordenador de placa simple con conectividad Wifi.

1.3.4 Beneficios sociales. Los beneficios sociales que se obtuvieron se evidencian de la siguiente manera:

- La aplicación Web de los paneles solares, contiene información de contacto que ofrece extensión a la UFPS, facilitando el conocimiento del proyecto a estudiantes, investigadores y científicos de otras instituciones de todo el mundo, y que deseen participar en los proyectos gestionados por el departamento de electricidad y electrónica.
- Una aplicación visible en Internet demostró la importancia que tiene el departamento de electricidad y electrónica de la UFPS, en el estudio de energías limpias en la región, incentivando el ingreso de nuevos investigadores a la UFPS.

1.3.5 Beneficios institucionales. Finalmente, los beneficios institucionales obtenidos fueron los siguientes:

- Se liberó recursos humanos y tecnológicos, para concentrarlos en el tratamiento de datos, y no, en la captación de los mismos; algo que los obliga a desplazarse y realizar las

funciones, que el sistema de supervisión remota, realizaría.

- La posibilidad de conexión desde cualquier dispositivo, facilitó a los investigadores, el acceso a la información en tiempo real, remotamente.
- El uso de base de datos en la nube, respalda la información en caso de pérdida en alguno de los discos de memoria, y facilitó, el acceso remoto a la información, y seguimiento de las variables de los paneles solares en tiempo real.

1.4 Alcance

El desarrollo de este proyecto, ofrece una alternativa económica a la monitorización remota desde la nube, de las diferentes variables en los paneles solares; y buscó por medio de las nuevas tecnologías de la información, ser una solución viable a las empresas que necesitan conectar su instrumentación a la nube, para realizar supervisión y respaldo de la información en tiempo real, desde cualquier dispositivo conectado a Internet.

1.4.1 Tipo de proyecto. El diseño y desarrollo del prototipo de, un sistema de instrumentación para la supervisión desde la nube de las variables en los paneles solares, es un tipo de investigación comparada con aplicación de conocimientos sobre un diseño experimental, utilizando el método científico y aplicando los conocimientos adquiridos durante el proceso de formación académica, en áreas de la electrónica, instrumentación, telecomunicaciones y tecnologías de la información.

1.4.2 Resultados esperados. En el desarrollo del proyecto, los resultados cumplieron con los objetivos del mismo. A continuación, se especifican los resultados directos e indirectos:

1.4.2.1 Resultados directos. El departamento de electricidad y electrónica, tiene la posibilidad de ampliar la cantidad de variables medidas, bajo el mismo sistema de captación de datos. El uso de la aplicación Web facilitó el acceso a la información desde cualquier dispositivo conectado a Internet, concentrando los recursos en el tratamiento de los datos. Además, tiene un respaldo de la información en una base de datos en la nube.

1.4.2.2 Resultados indirectos. La consolidación del conocimiento en esta investigación, permitió fomentar el interés en la monitorización por medio de tecnologías de la información, llevando nuevas tecnologías a la instrumentación electrónica, e interesando a la comunidad académica, instituciones y empresas, en la integración de industria 4.0 en los procesos de supervisión remota, de las variables necesarias en los diferentes procesos.

1.5 Limitaciones

En la realización de un prototipo de supervisión de datos, desde la nube en tiempo real, se enfrentaron varias limitaciones que se consideraron para que no afectaran la realización o la calidad del mismo.

Una importante limitante fue la falta de un software de simulación, que integre el diseño de la instrumentación, la captación y el envío de datos a una plataforma de Backend con DataBase en la nube, lo que impidió evaluar la totalidad del diseño antes de su implementación, poniendo en riesgo la lectura y captación de los datos; porque los software de simulación electrónica, no permitieron la emulación de sistemas informáticos, los software de emulación informática no permitieron la simulación de sistemas electrónicos, y las aplicaciones Web necesitan de servidores con acceso a Internet para su correcto funcionamiento, y en pruebas locales, no se puede comprobar la lectura y descarga desde cualquier dispositivo conectado a Internet.

Otra limitante es la falta de un sistema embebido de bajo coste, que realice todas las actividades integradas, lo que obligo el uso de varias tecnologías para el cumplimiento de cada etapa, buscando la compatibilidad entre ellas para la integración entre los sistemas. Al utilizarse varios componentes debió analizarse su compatibilidad, disponibilidad en el mercado, además de su precio, porque uno de los objetivos de este proyecto es que sea un sistema de bajo coste. El tiempo de envío de los dispositivos no disponibles en el mercado (o muy costosos a precio nacional) juega un papel importante en el desarrollo del prototipo, pues se necesitó de la disponibilidad total del sistema electrónico e informático para la realización de pruebas, por no disponerse de un software de simulación que integre todas las tecnologías.

En la selección de los sensores se descartó el sensor Sct013 YHDC, por no permitir la lectura de corriente directa, no disponer de un circuito acondicionador para su fácil integración con el conversor ADC, tener un costo elevado y una disponibilidad deficiente, para la ser utilizado.

En la etapa de realización de pruebas del prototipo, se dependió del clima de la ciudad, pues, por factores atmosféricos se retrasó o incluso se canceló la toma de datos, en algunos días.

1.6 Delimitaciones

Este proyecto, desarrolló un prototipo de supervisión remota de bajo coste, para la lectura de paneles en la ciudad de Cúcuta. En función de los recursos económicos, de tiempo, personal y tecnológicos, el sistema de supervisión se rigió solamente a lectura, centralización y respaldo de datos en una DataBase en la nube. Datos a los cuales se pudo acceder desde cualquier dispositivo conectado a Internet; pues conto con una aplicación Web desplegada en la nube, que se encargó de la visualización y descarga de datos, para cada investigador que tenga acceso de usuario a la aplicación.

1.7 Objetivos

Para el desarrollo de este proyecto se plantearon los siguientes objetivos tomando en cuenta la duración del mismo.

1.7.1 Objetivo general. Desarrollar un prototipo de supervisión desde la nube, para las variables en los paneles solares.

1.7.2 Objetivos específicos. Los objetivos específicos se muestran a continuación:

Recopilar información sobre monitorización remota, instrumentación, captación y respaldo de los datos de variables en paneles solares.

Diseñar un sistema de instrumentación electrónica no invasivo para la captación, conversión Análogo-Digital y centralización de los datos en Raspberry Pi, de las variables en paneles solares.

Programar aplicación Web Responsive en Firebase con conexión a Raspberry Pi como centro de supervisión, procesamiento y envío de datos al Servidor de la DataBase.

Realizar pruebas al sistema de supervisión, respaldo de la data en la nube y descarga remota de las variables en los paneles solares.

Presentar resultados y beneficios de nuevas tecnologías digitales en instrumentación electrónica y aplicaciones Web para industria 4.0.

2. Marco Referencial

Las presentes investigaciones, generaron aportes de interés para la realización del proyecto, y ayudaron en una mejor toma de decisiones, para la selección de hardware, y software.

Permitieron tener una idea general del cumplimiento de los objetivos, dando una visión global de como estas tecnologías son desarrolladas en otros países, y que aplicaciones están siendo exigidas por las instituciones, para la supervisión remota, con tecnologías de bajo coste.

2.1 Antecedentes

Schmalb (2021), en su trabajo titulado “Monitorização de uma rede de painéis fotovoltaicos usando um sistema embarcado de baixo custo”. Este proyecto pretendía ofrecer una herramienta para la administración de los recursos energéticos de pequeñas plantas fotovoltaicas en la región de Aracaju, Brasil, una de las zonas con mayores índice de radiación, y donde por miedo de sistemas embebidos de bajo costo, se pretendía ofrecer acceso a los datos de funcionamiento de los paneles fotovoltaicos, y con los cuales se buscaba estudiar posibilidades de mantenimiento, mejoras, o reajustes de diseño del sistema fotovoltaico, pues la finalidad de las pequeñas plantas, es reducir en la mayor cantidad posible, la dependencia de la red eléctrica convencional.

Esta investigación obtuvo como resultado, un módulo de bajo costo con el uso de microcontroladores de tecnologías Arduino y ESP8266, para la administración de información relevante en los paneles solares, y facilitó una herramienta para el seguimiento de alteraciones en el funcionamiento eficiente de pequeñas plantas fotovoltaicas.

Este trabajo aportó una idea del buen funcionamiento del sistema fotovoltaico, con captación de información por medio de un microcontrolador, ayudando en la selección de tecnologías de

captación de bajo costo (Schmalb, 2021).

Pontes & Souza (2021), en su trabajo titulado “Implantação de estação solarimétrica e de sistema supervisório com SCADABR e plataforma IoT em usina fotovoltaica na UNILAB-CE”. Las empresas generadoras de energía e instituciones privadas cada vez hacen mayor uso de energía fotovoltaica para suplir sus necesidades energéticas, u ofrecer este servicio en sitio; por lo que demandan herramientas, para la gestión de estos diseños, y prever mantenimientos que mejoren la eficiencia de los diseños fotovoltaicos implementados.

Como objetivo principal de este proyecto, se implementó un sistema de supervisión en SCADABR (Software libre para desarrollo de aplicaciones de automatización, control y adquisición de datos remoto) que permitió la monitorización remota de una estación fotovoltaica para la generación de energía a baja escala (254,2Kwp), en las instalaciones del “Campus das Auroras da Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB” en Ceará, Brasil.

Esta investigación culminó en el desarrollo de un módulo de gestión de datos remota IoT, desarrollado con tecnología NodeMcu ESP8266-12E, y el uso de tecnología remotas, por medio de Sockets que permitieron el almacenamiento y gestión de datos en dispositivos móviles.

Como aporte esta investigación permitió conocer una plataforma de monitorización con software libre, y ayudo a tener una perspectiva, de cómo el mercado necesita estos servicios a bajo costo, y como las empresas empiezan a solicitar este tipo de productos, para la generación en plantas de energía pequeñas (Pontes & Souza, 2021).

Mallegowda, Anithakanavlli & Amrutha (2020), en su trabajo titulado “Design and integration of middleware for IoT devices towards solar panel monitoring based on Raspberry Pi”. La supervisión de procesos, paso a un nuevo nivel de desarrollo, donde ya no se limita a aplicación nativas de los sistemas operativos, y donde se empieza a usar tecnologías de la información, para el desarrollo de aplicaciones de igual prestaciones, como lo son las desarrolladas en códigos nativos de las maquinas.

Este proyecto tuvo como objetivo investigar, desarrollar, probar e implementar un sistema de supervisión de bajo coste con todos los beneficios y aplicaciones del IoT, para la monitorización y control de las variables de un sistema fotovoltaico, donde por medio de desarrollo Frontend y Backend se visualizarán datos en dispositivos conectados a Internet; todo bajo tecnología compatible con Raspberry Pi.

Al concluir este proyecto se determinó que iniciar etapas secuenciales, ofrece garantía en la toma de decisiones, y facilita la corrección de errores en la integración de tecnologías de hardware y software.

Los resultados de este proyecto fomentaron una toma de componentes más detallada, pues es necesario un hardware que soporte las tecnologías informáticas que se usaron, y asegurar las plataformas que sean integrables entre sí, culminando con éxito, cualquier proyecto que aplique estas tecnologías (Mallegowda et al., 2020).

De Vasconcellos (2020), en su trabajo titulado “Desenvolvimento de um sistema de supervisão, controle e manutenção baseado em condição para usinas de geração solar fotovoltaicas usando o conceito IoT”. Las empresas que implementan sistemas fotovoltaicos, tienen interés en monitorizar las prestaciones energéticas de sus sistemas, y ofrecer tanto al

cliente, como a ellos, mediciones con mayor confianza, y con ellos familiarizar a los usuarios con la producción energética de sus sistemas fotovoltaicos, y se determine un criterio de eficiencia según el consumo para el que se diseñó el sistema.

Este proyecto tuvo como objetivo principal, desarrollar un sistema de supervisión, control y mantenimiento de bajo costo, para que las mayorías de consumidores de plantas de energía fotovoltaicas de baja escala; quienes usualmente no tienen los conocimientos especializados en producción energética, tengan una aplicación Web amigable soportada en los protocolos de comunicación HTTP en la nube, y puedan conocer el estado, y funcionamiento de su sistema fotovoltaico.

Con este proyecto se pudo observar las variables en las plantas de generación fotovoltaica, y como se debía enfocar la toma de datos de interés, que tienen los sistemas fotovoltaicos, y con ello organizar la información. También se logró tener una idea de cómo las instituciones, y usuarios de estos servicios de energías renovables, quieren determinar con anticipación, cualquier reajuste, o mantenimiento de los sistemas (De Vasconcellos 2020).

Castro (2020), en su trabajo titulado “Análisis de Eficiencia en Conversión/Almacenamiento de Energía Solar y Adquisición de Datos de Paneles Solares del Laboratorio de Física y Electrónica de la UNAD (Sede Nacional JCM) en el Marco de su Uso como Laboratorio Remoto”. La evolución de las tecnologías, facilita la integración de diferentes plataformas para llevar seguimiento remoto de los procesos de interés en áreas académicas, lo que interesa a las instituciones en integrar sus trabajos de investigación, para trabajar en sitios diferentes en los que se generan los procesos estudiados. Por eso la UNAD tuvo como interés hacer seguimiento a los procesos fotovoltaicos, en sus laboratorios, y como la eficiencia de carga de baterías puede

determinarse con datos tomados remotamente.

Este trabajo estudia la eficiencia del sistema fotovoltaico por medio de telemetría o telecontrol de equipos y dispositivos, y con esto poder realizar inspección y detección de fallas por medio de acceso a Internet.

Con este trabajo se pudo conocer, como las variables que son medidas, determinan la eficiencia del sistema fotovoltaico, y como por medio de telemetría o telecontrol de equipos y dispositivos, se puede realizar la inspección y detección de fallas solo con acceso a Internet, facilitando entender, como tener acceso remoto a los datos, juega un papel importante en la respuesta de reacción ante posibles fallos (Castro, 2020).

2.2 Marco Teórico

En la supervisión desde la nube de las variables de los paneles solares, se tuvo en cuenta, conceptos claves que llevaron al cumplimiento del proyecto, y que, son definiciones que dan claridad de los procesos de desarrollo del mismo, pues este, se reconoce por integrar varias áreas de la tecnología.

2.2.1 Energía solar. La energía solar, es aquella emitida por el Sol, que se considera inagotable por la perdurabilidad en el tiempo en función de la vida en la Tierra, y de la cual, ofrece un estimado de 10.000 veces más energía que todas las fuentes de energía utilizada por los seres humanos. Toda esta energía llega a la superficie terrestre de dos formas: radiación directa y radiación difusa (Hilcu, 2021).

2.2.2 Eficiencia de un sistema fotovoltaico. Los sistemas fotovoltaico fabricados de material semiconductor, aprovechan las propiedades físicas de las células solares de convertir la radiación

solar en energía eléctrica, y esta conversión se mide en eficiencia. Un 95% de estas células tienen una eficiencia del 17%, pero con los desarrollos de las últimas décadas han aumentado hasta en un 31% (De la Cruz & Bastidas, 2020)

2.2.3 Instrumentación electrónica y adquisición de datos. La instrumentación electrónica, es una rama de la ingeniería electrónica que sintetiza todos los conocimientos teóricos en la formación del ingeniero, en post de medir eficientemente las variables que intervienen en un proceso, y que, por medio de la caracterización de los sensores, ofrece al instrumentista, el criterio de instalar, calibrar, conectar y configurar los diferentes sistemas de medición al centro de datos. La adquisición de datos, se realiza aplicando una base sólida de la instrumentación a cada variable física, y por medio de los sensores debidamente calibrados, acondicionarlos al centro de datos digitales (Pallás, 2003).

2.2.4 Voltaje, corriente y potencia. El voltaje es conocido como el trabajo necesario para mover una carga de un punto a otro dentro del conductor, también, es llamado como diferencia de potencial o fuerza electromotriz (FEM), ya que, esta misma debe ser aplicada entre dos nodos en un circuito cerrado para la circulación de la carga dentro del circuito, y esta fuerza es medida en voltios (V), como unidad de medida (Charles & Matthewn, 2013).

La corriente es una característica exclusiva de las cargas eléctricas en movimiento; que pueden ser descritas como flujo de carga por unidad de tiempo, y medidas en Amperios (A) como unidad de medida; también puede ser interpretada como la velocidad de los electrones a través de una sección transversal de un conductor (Charles & Matthewn, 2013).

Potencia es la cantidad de energía consumida por un dispositivo en función del tiempo, y es necesaria, en cualquier circuito eléctrico o electrónico, ya que, es una media crítica para el

funcionamiento de cada componente (Charles & Matthew, 2013).

2.2.5 Sensor de Voltaje FZ0430 0.02445V – 25V. La medición de voltaje, en una resistencia de potencia en serie con el panel solar, permite realizar una medición de voltaje sin afectar el sistema, facilitando el diseño de un sistema de instrumentación no invasivo. El sensor de voltaje FZ0430 0.02445V – 25v (Figura 1) es un sensor configurable en un rango de trabajo de 0 – 16V con una alimentación de 3.3V, o un rango de 0 a 25V sin la necesidad de una alimentación de 5V (Carrod, 2021).

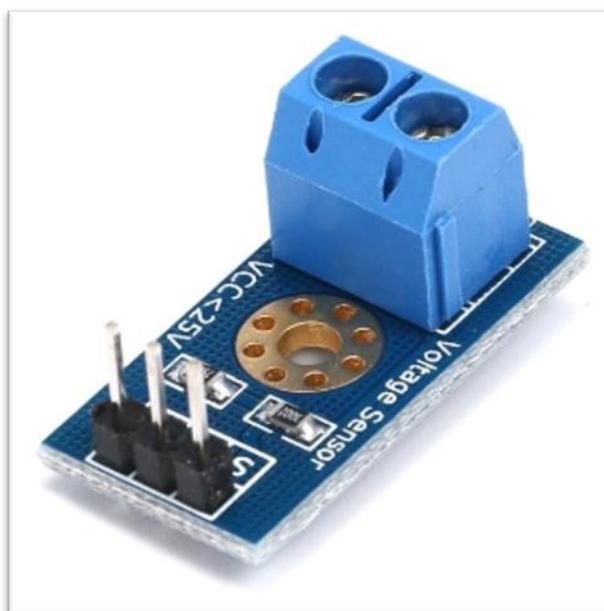


Figura 1. Sensor de voltaje FZ0430 0.02445V - 25V

Fuente: Carrod, 2021.

2.2.6 Sensor de voltaje SKU DFR0051. Este sensor realiza una lectura por división de voltaje, y puede medir rangos de lectura, entre 0.0245V – 25V (Figura 2), también es configurable, y permite una solución viable, para la lectura no invasiva del sistema (DFRobot, 2021).

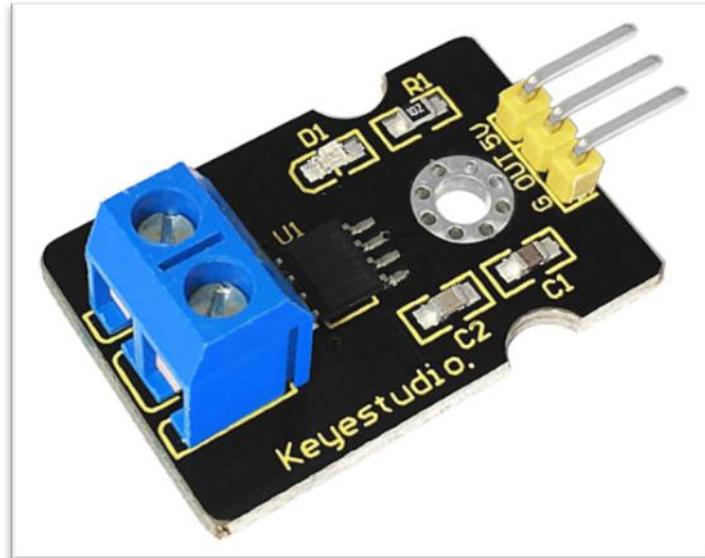


Figura 3. Sensor de corriente KS0185

Fuente: Keyestudio, 2021.

2.2.8 Conversión Análogo/Digital (ADC). Un conversor análogo/digital, es un sistema digital de procesamiento, donde las señales analógicas, ingresan al sistema y son interpretadas en una trama de datos, de manera digital; después esta trama de datos, son acondicionadas para su transmisión, según el formato de comunicación usado para la comunicación con la unidad de procesamiento central del dispositivo de control (Floyd, 2006).

2.2.9 Sistemas embebidos. Es llamado como sistema embebido, a un circuito electrónico que integra todas las características de software y hardware computacional, pero diseñado para la realización de una tarea específica. Todos los algoritmos de funcionamiento están guardados y operativos desde la unidad de procesamiento central del dispositivo de control o microcontrolador (Galeano, 2009).

2.2.10 Arduino. Nacido en 2005 en el Instituto de Diseño Iterativo de Ivrea (Italia), es una plataforma de desarrollo, que se soporta sobre un sistema embebido con placa de hardware libre (Figura 4), y microcontrolador reprogramable AVR de la familia Atmel. Su lenguaje de programación está basado en C++, es de bajo coste, y cuenta con una comunidad mundial que mantiene en constante actualización (Arduino, 2021).



Figura 4. Placas de desarrollo Arduino

Fuente: Arduino, 2021.

2.2.11 Raspberry Pi. Nacida en febrero del 2012, fue diseñada para la enseñanza de computación básica, pero debido a su practicidad como un ordenador de placa simple (Figura 5), bajo coste y tamaño compacto, comenzó a gobernar el mercado como un sistema embebido con microprocesador de sistema operativo basado en Linux, que permite la realización de tareas específicas, y procesamiento de señales. También, es usado como centro de control en supervisión remota, cuenta con una comunidad de mundial que se mantiene en actualización (Raspberry Pi Fountion, 2021).



Figura 5. Raspberry Pi 3B+

Fuente: Raspberry Pi Fountion, 2021.

2.2.12 Open source. Se refiere a toda la información, trabajos, investigaciones, software, hardware, aportes académicos y científicos de libre acceso para todo público, donde se permite, su descarga, manipulación, desarrollo e implementación, sin ninguna obligación mayor que respetar el copyright y su debido reconocimiento a los autores del desarrollo original. Bajo esta nueva cultura de acceso libre a la información y avances tecnológicos, se ha integrado año tras año, una nueva comunidad de investigadores, que dejan a libre público, sus descubrimientos y avances en cada área de su especialidad, por lo general en la electrónica y sistemas; mejorando cada día, el tiempo de respuesta ante fallos, y manteniendo actualizada a toda la comunidad en cada avance realizado (Spano, 2010).

2.2.13 Tecnologías de la información. Es la integración de desarrollos tecnológicos para la transmisión, almacenamiento y procesamiento la información en la red (Ministerio de Tecnologías de la Información y las Comunicaciones, 2021).

2.2.14 Protocolo de comunicación Web. El acceso libre a los protocolos TCP/IP (Transport Control Protocol/Internet Protocol) son los que facilitaron el crecimiento del Internet, donde la Web comenzó a sostener su éxito sobre dos pilares importantes que son: el protocolo HTTP, y lenguaje HTML, dejando una sin fin de páginas estáticas. El protocolo HTTP establece una comunicación TCP con el Servidor, lo que hace un canal de comunicación extremo a extremo entre cliente y Servidor. HTML ofrece enlaces entre páginas de forma fácil y simple, por medio de texto (Mateu, 2004).

2.2.15 Lenguajes de desarrollo Web Html, Css y JavaScript. HTML se conforma como un documento de texto plano, donde por medio de etiquetas (tags), se limitan los parámetros, y donde al inicio de cada etiqueta se pueden especificar atributos característicos de cada texto (Mateu, 2004). Ha de recordarse que este lenguaje solo enmarca el texto, y ofrece la información a la página, pero también redirecciona a los contenidos de la misma, la lógica general del HTML es: **<Etiqueta Atributo Atributo> “Aquí se escribe el texto” </Etiqueta>**.

CSS es la parte estética de las paginas, ya que complementa al HTML, para la apariencia de la misma, siendo un lenguaje de estilos obligatorio para diferenciar los componentes de presentación a los de información (Puig, 2004).

JavaScript (Js) es un lenguaje de programación interpretado por el cliente, desarrollado originalmente, porque se debía programar un Script por cada navegador, pero por la dificultad del mismo, se empezó a aplicar las API (Application Programming Interface), comunes para todos los navegadores. Facilitando por mucho el tiempo de desarrollo y mejorando la experiencia de los clientes ante páginas dinámicas (Puig, 2004).

Se debe enmarcar que, cada uno de los lenguajes puede ser codificado desde el mismo archivo HTML, pero se recomienda por organización de la programación, y detección de fallos más precisos, realizar documentos separados y llamarlos desde una etiqueta en HTML.

Estos lenguajes son los que componen el Frontend de una aplicación Web, siendo la interfaz gráfica que interactúa con el cliente; el Frontend se despliega en un Hosting, y desplegada se encarga de conectar con el Backend o Servidor, ofreciendo la gestión de bases de datos, autenticación, mensajes, o cualquier servicio programado del lado del Servidor.

2.2.16 Hosting. Hosting es el encargado de alojar, el despliegue de un sitio o aplicación Web, y donde cada archivo codificado en los lenguajes de desarrollo Web, se almacena para ejecutar las interacciones y operaciones de la aplicación. Los servicios de Hosting van de la mano del dominio, que son la dirección Web que el cliente usará para conectarse a la aplicación, pero normalmente son de pago y los servicios gratuitos, no ofrecen mucha seguridad, ni evitan caídas de la página; lo que obliga a los usuarios a contratar con una empresa que destina sus servidores para esta función.

Google ofrece su propio servicio de hosting y dominio gratuito, solo con acceso a un correo Gmail, este servicio es Firebase Hosting (Firebase, 2021a).

2.2.17 Base de datos o DataBase. Toda la información debe ser organizada para su posterior tratamiento, de esa forma es que una base de datos facilita este proceso, ya que es una colección estructurada de la información.

Google ofrece su propio servicio de DataBase gratuito, solo con acceso a un correo Gmail; se tiene acceso a dos diferentes tipos de DataBase: Firebase Realtime DataBase o Firestore

DataBase (Firebase, 2021b; Firebase, 2021c).

Para acceder a la DataBase, es necesario configurar el Backend, que permita la gestión de los datos del lado del servidor.

2.2.18 Json. JSON (JavaScript Object Notation) es una estructura de datos que facilita la creación y manejo de base de datos, ya que omite la obligación de bases relacionales y no relacionales, y trabaja en una sola compilación de las mismas, donde es más fácil la lectura para el usuario y la máquina (International Business Machines, 2021).

Hay que destacar, que por la facilidad de manejo de una base de datos con formato Json, Firebase Realtime DataBase almacena los datos en este formato y los sincroniza en tiempo real.

2.2.19 Python. Lenguaje interpretado e interactivo de alto nivel orientado a objetos, con tipado dinámico y robusto; se caracteriza por una sintaxis clara que favorece la interpretación por parte del programador. Es un software de código libre que permite la integración con productos copyright, y es soportado bajo todos los sistemas operativos y plataformas de desarrollo, lo que facilita la automatización de tareas y nuevas funciones de los programas, entre ellas los servicios de Backend, en los servidores (Borges, 2009).

2.2.20 Firebase. Plataforma de Backend en la nube, para desarrollo Web y Móvil (Figura 6), adquirida por Google en el 2014, y que junto a Google Cloud Platform , ofrece herramientas para desarrollo como lo son Develop en Firebase, Realtime DataBase, Autenticación de usuarios, Almacenamiento en la nube, Crash Reporting, Test Lab, Remote Config, Cloud Messaging y Hosting, entre otros, que reducen el tiempo de soporte en Backend (Firebase, 2021d).

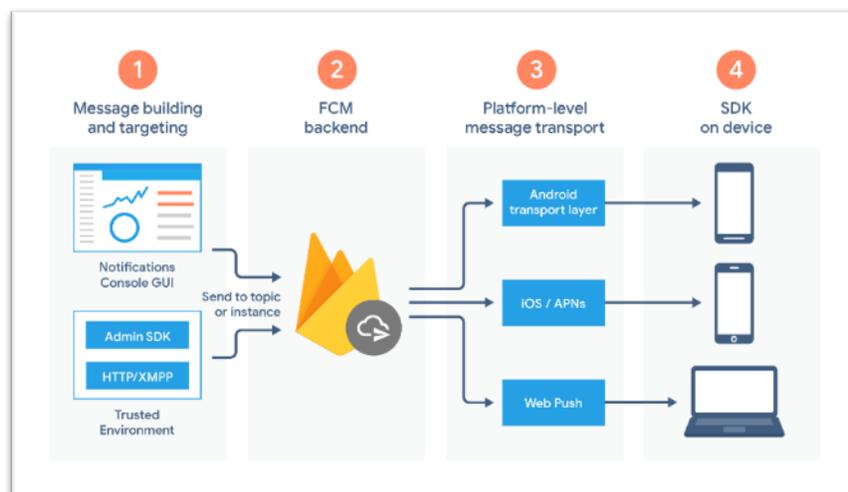


Figura 6. Descripción general de la arquitectura FCM en Firebase

Fuente: Firebase, 2021d.

2.2.21 Aplicación Web. Al dominarse el desarrollo de páginas estáticas, la creación Web paso a otro nivel, donde se empezaría a realizar cálculos, recibir peticiones, enviar y descargar datos, restringir los datos a usuarios permitidos; todo esto dinámico del lado del cliente con el Frontend, pero interactuando con el servidor por medio del Backend o lenguaje y aplicaciones de lado del servidor, diseñadas con este fin (Mateu, 2004).

2.2.22 Sistemas en tiempo real. Son sistemas caracterizados por tener relación directa en el cambio de sus variables en el proceso medido, y en la interacción directa con el procesamiento de sus datos e interfaz de la misma; por decirlo de alguna manera, la variable de cambio en la interfaz de usuario ha de cambiar y ser precisa al mismo tiempo que su variación en el proceso medido (Muñoz, 2009).

2.2.23 Internet de las cosas. Conocido como la integración de los sistemas informáticos en elementos de uso cotidiano. El Internet de las cosas vino a dar una nueva era a los sistemas

físicos cotidianos, teniendo acceso remoto, y todos los sistemas físicos y digitales interconectado (Rodríguez, 2019).

2.2.24 Industria 4.0. El termino se acuño cuando empezó a integrarse los sistemas informáticos con la industria manufacturera; supervisando, controlando e incluso realizando mantenimientos remotos apoyados en sistemas informáticos, o en ocasiones sin ninguna intervención humana (Ministerio de Tecnología de las Información y las Comunicaciones, 2019).

Esta tecnología, no se compone de una sola rama de ingeniería, sino que integra los conocimientos de electrónica, mecánica, control, telecomunicaciones e Internet de las cosas, para realizar todos los procesos conocidos en manufactura, pero con toda la red conectada entre sí, y con acceso remoto.

2.3 Marco Legal

El avance en Energía Solar ha posicionado a los países en la vanguardia de energías limpias, y buscando mitigar la huella ambiental dejada por los hidrocarburos a nivel mundial.

Colombia generó por ley incentivos al desarrollo de nuevas tecnologías, en proyectos de energías renovables no convencionales, y cubierto con beneficios tributarios de la Ley 1715 del 2014, que sigue regulado bajo los parámetros de la ley 697 de 2001.

2.3.1 Ley 1715 de 2014 Congreso de Colombia. Artículo 19. Desarrollo de la energía solar. El Gobierno Nacional considerará la viabilidad de desarrollar la energía solar como fuente de autogeneración para los estratos 1, .2 Y 3 como alternativa al subsidio existente para el consumo de electricidad de estos usuarios (Ministerio de Minas y Energía, 2022).

2.3.2 Ley 697 de 2001 Congreso de Colombia. Artículo 9. Promoción del uso de fuentes no convencionales de energía. El Ministerio de Minas y Energía formulará los lineamientos de las políticas, estrategias e instrumentos para el fomento y la promoción de las fuentes no convencionales de energía, con prelación en las zonas no interconectadas.

Artículo 10. El Gobierno Nacional a través de los programas que se diseñen, incentivará y promoverá a las empresas que importen o produzcan piezas, calentadores, paneles solares, generadores de biogás, motores eólicos, y/o cualquier otra tecnología o producto que use como fuente total o parcial las energías no convencionales (Congreso de Colombia, 2001).

3. Diseño Metodológico

Para el cumplimiento de los objetivos, se presentó una descripción detallada de la metodología realizada, que dio cumplimiento y solución a la problemática.

3.1 Tipo de Metodología

Como base fundamental de la metodología se consolidaron 5 fases que facilitó el desarrollo del prototipo, donde la fase inicial se apoyó en la búsqueda de información, para una etapa de selección que permitió cuantificar las características de las diferentes plataformas evaluadas, la segunda fase consistió en el diseño de la instrumentación, para la captación, almacenamiento y envío de la información a una base de datos, la tercera fase se dedicó a la creación de la interfaz gráfica en una aplicación Web Responsive, que se ejecutara en múltiples dispositivos, en la cuarta fase se integraron todas las tecnologías en un prototipo funcional, realizando pruebas de funcionamiento, y la última fase se dedicó a la divulgación de los resultados, de manera internacional, nacional y regional, respectivamente.

3.2 Fases del Proyecto

3.2.1 Fase 1. Selección de componentes de hardware. El desarrollo de la primera fase se evidencia a continuación:

3.2.1.1 Desarrollo. Analizando la información previa de las referencias de trabajos de investigación anteriores, se verificó como se han desarrollado sistemas de supervisión con plataformas de bajo coste, en otros países, y fue necesario una fase inicial de selección del hardware, que debió soportar las tecnologías necesarias, para la supervisión remota con tecnologías de la información, pues en el mercado se encuentran diferentes desarrollos

tecnológicos, y diferentes integraciones multiplataformas que dan un cumplimiento cercano a los objetivos buscados en este proyecto; pero ciertas tecnologías integradas, permitieron generar mayores beneficios, y fueron capaces de soportar un diseño con mejor viabilidad que otros ofrecidas para funciones similares.

En esta etapa se definieron los componentes según las utilidades, prestaciones, soporte, disponibilidad y precio, que aportaron en el desarrollo de la investigación. Para la selección se desarrolló un modelo de calificación, basado en la Matriz de Pugh, donde se pudo realizar comparativas cuantificadas, de los valores de mayores beneficios para el proyecto, en contraste con los de menor beneficio, y categorizando en porcentajes, las características de cada plataforma, en función de la de mayor beneficio; obteniéndose al final una evaluación cuantitativa de viabilidad de la plataforma, para el cumplimiento de los objetivos del proyecto.

Para poder tener centralizada la información se buscó un proveedor autorizado a nivel mundial de componentes electrónicos, para obtener los datos que se evaluaron en la etapa de selección.

3.2.2 Fase 2. Diseño de un sistema de instrumentación electrónica no invasivo para la captación, conversión análogo-digital y centralización de los datos en Raspberry Pi, de las variables en paneles solares. A continuación el desarrollo de la segunda fase:

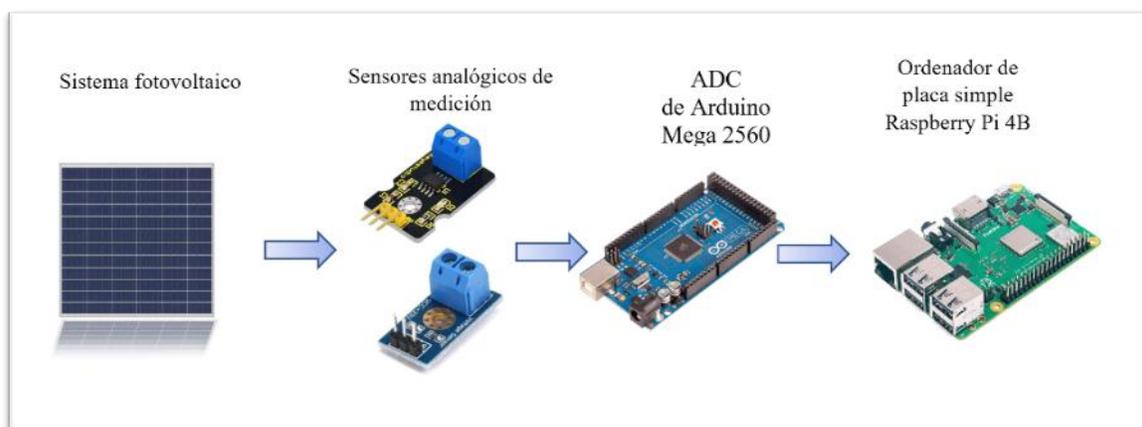


Figura 7. Proceso de captación de datos por medio de la instrumentación

3.2.2.1 Desarrollo. La medición de variables de corriente y potencia de los paneles, se realiza de manera indirecta al sensar el voltaje en una resistencia de potencia en serie con el panel. Por lo que se diseñó un circuito en el software de simulación Proteus Design Suite 8.10 ©, y donde se aprovechó las propiedades de simulación ofrecidas por el software, para configurar, y comprobar la lectura de los sensores de voltajes FZ0430, SKU DFR0051, de corriente de efecto Hall KS0185, y de la placa de captación de datos Arduino Mega 2560, seleccionados de la primera fase.

Como la lectura de las variables, dependen de la radiación, y en función practica del proyecto, se desarrolló un circuito del panel, basado en el modelo electrónico de diodo sencillo con resistencia Shunt del panel, el cual permitió la toma de datos variables, según la radiación que recibe el panel.

La lectura de voltaje, se realizó por medio de un diseño basado en el circuito eléctrico de los sensores de voltaje FZ0430 (ver figura 8), y SKU DFR0051, cuyo rango de medida es de 0V - 25V, y el cual se simuló utilizando el circuito eléctrico del sensor, para emular la captación del voltaje. Esta simulación cubrió el comportamiento ideal de los dos sensores de voltaje, pues su

construcción es la misma, por lo que su comportamiento ideal, también.

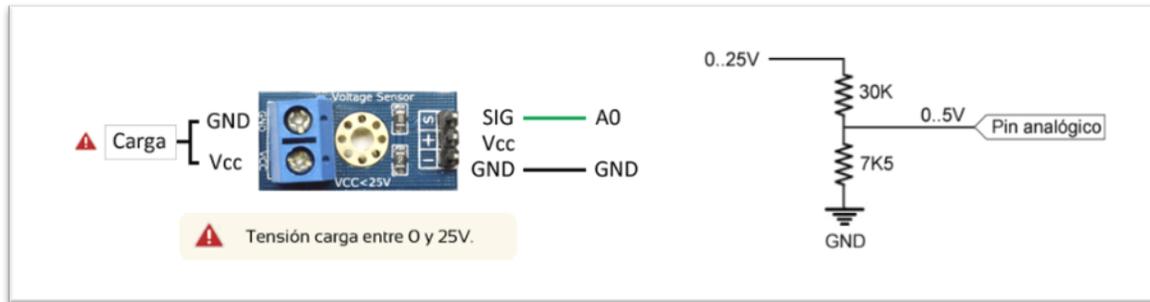


Figura 8. Esquemático de conexión del sensor de voltaje FZ0430

Fuente: Llamas, 2021.

Para la medición de corriente, se consideró inicialmente el sensor SCT013 YHDC, pero al analizarse su aporte a la investigación, se descubrió que no podía realizar la medición de corriente directa, la disponibilidad del sensor en el mercado local es deficiente, el precio es costoso, y necesita de un circuito de acondicionamiento adicional para ser interpretado por el sistema de instrumentación, lo que obligatoriamente lo descarto como una opción viable para el proyecto.

La solución de mayor viabilidad que se investigó, fue la que presentó el sensor de corriente KS0185, el cual se vende en el mercado local, en un formato embebido, que cuenta con el circuito de acondicionamiento para la lectura en microcontroladores, tiene un costo menor que la mitad del precio del SCT013 YHDC, y permite la lectura de corrientes directa y alterna, bajo la misma plataforma.

El circuito ACS172 está disponible en el software de simulación Proteus Design Suite 8.10 ©, lo que permitió, simular y comprobar la lectura de corriente generada en el panel.

Las señales recibidas de estos sensores son de tipo analógico, imposibles de interpretarse directamente en la Raspberry Pi; lo que obligó, la configuración de una etapa de conversión que adaptó las señales analógicas en formato digital, y que fueron interpretadas en el microprocesador de la Raspberry Pi.

La conversión analógico-digital, se realizó aprovechando el conversor ADC del Arduino Mega 2560, de 16 canales y una resolución de 10bits y un voltaje máximo de 5 voltios; una tarjeta que se acopla completamente con el ordenador de placa simple Raspberry Pi por medio de comunicación serial.

Como el sistema de simulación electrónica Proteus Design Suite 8.10 © no permite la emulación del sistema informático de la Raspberry Pi, se debió emular el comportamiento informático, en la plataforma de máquina virtual Oracle VM VirtualBox 6.1.18 ©; en ella se instaló el sistema operativo Rasbian 5.10 (Figura 9), y se desarrolló las conexiones, y el ejecutable para el envío de datos a la nube, y teniendo las bases computacionales configuradas se seleccionó la plataforma de Backend, que soporto la recepción, almacenamiento y envió de datos en la nube.

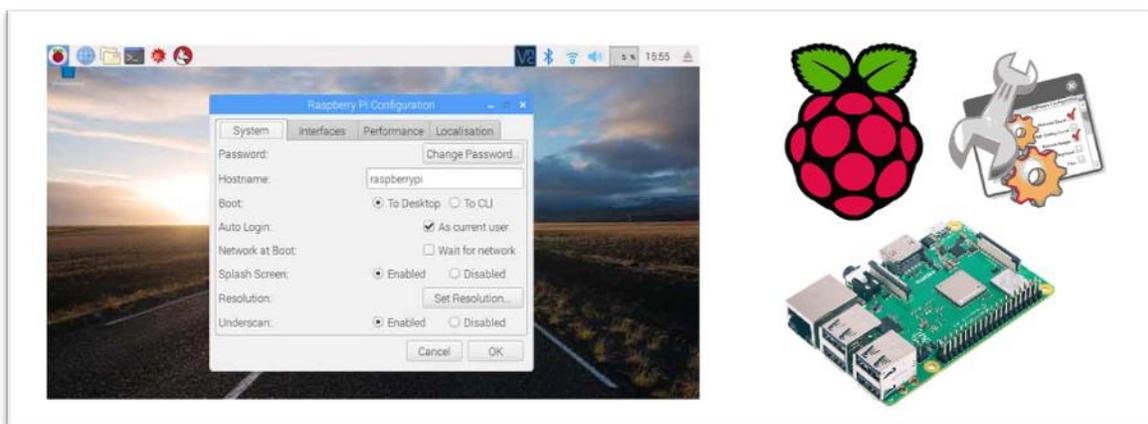


Figura 9. Instalación y configuración del sistema operativo en Raspberry Pi

La base de datos que se eligió, debió ser una DataBase asíncrona, que mantuviera su integridad, aun cuando el sistema de información no estuviera en uso por alguna razón; y esta se configuró en el ejecutable Python que administró la conexión de la instrumentación de Arduino, con la Raspberry Pi, y de la Raspberry Pi con la nube (ver figura 10).



Figura 10. Descarga de las dependencias de Raspberry Pi y creación de archivo Pitón

3.2.3 Fase 3. Diseño, configuración y programación de una aplicación Web Responsive en Firebase con conexión a Raspberry Pi como centro de supervisión, procesamiento y envío de datos al Servidor de la DataBase. El desarrollo de la tercera fase se evidencia a continuación:

3.2.3.1 Desarrollo. Se desarrolló una aplicación Web basada en los lenguajes de desarrollo HTML5, CSS3, JavaScript5, que son soportados por la mayoría de los navegadores actuales (por no decir que todos). Para el desarrollo de esta aplicación se utilizó el editor de texto Brackets 1.14 (Figura 11), ya que, es un editor liviano y con una herramienta de vista previa desde el Browser de Google Chrome, que permite la visualizar en tiempo real la interfaz Web.



Figura 11. Brackets, editor de texto Open Source

Finalizada la aplicación se integró con la plataforma Backend de desarrollo de aplicaciones Web, Firebase (ver figura 12). Plataforma de Google que ofrece gratuito en sus servidores, los servicios de Hosting, DataBase en tiempo real, Storage y Autenticación; herramientas necesarias en los objetivos de este proyecto. La conexión se realizó por medio de archivos programados en JavaScript5, que se alojaron en el Hosting de esta plataforma, y que, con la configuración de las API y SDK, permitieron a la aplicación, interacción entre el centro de datos, la DataBase y el usuario registrado.



Figura 12. Plataforma de desarrollo Backend, para aplicaciones Web

Fuente: Programa en Línea, 2022.

3.2.4 Fase 4. Pruebas del sistema de supervisión, respaldo de la data en la nube, y descarga remota de las variables en los paneles solares. El desarrollo de la cuarta fase se aprecia de la siguiente manera:

3.2.4.1 Desarrollo. Se conectó la instrumentación a un panel solar de prueba, y se comprobó la lectura, procesamiento y envío de datos desde la Raspberry Pi a la DataBase. En el algoritmo de funcionamiento para el procesamiento de los datos, se describe que los datos son números reales, que necesitan expresar los decimales de las variables analógicas.

La conexión de la instrumentación, debe trabajar con una conexión estable, de no ser así, debe verificarse y corregirse para iniciar mediciones. Cumpliendo con una conexión estable de la instrumentación, se verifica que los datos desde los sensores son correctos, y no valores nulos o ceros.

Con una conexión estable de la instrumentación, y la toma de datos correcta, se verifica la conexión con la base de datos para el almacenamiento de la información; en caso de no tener una correcta conexión debe verificarse y corregir para almacenar las mediciones (ver figura 13).

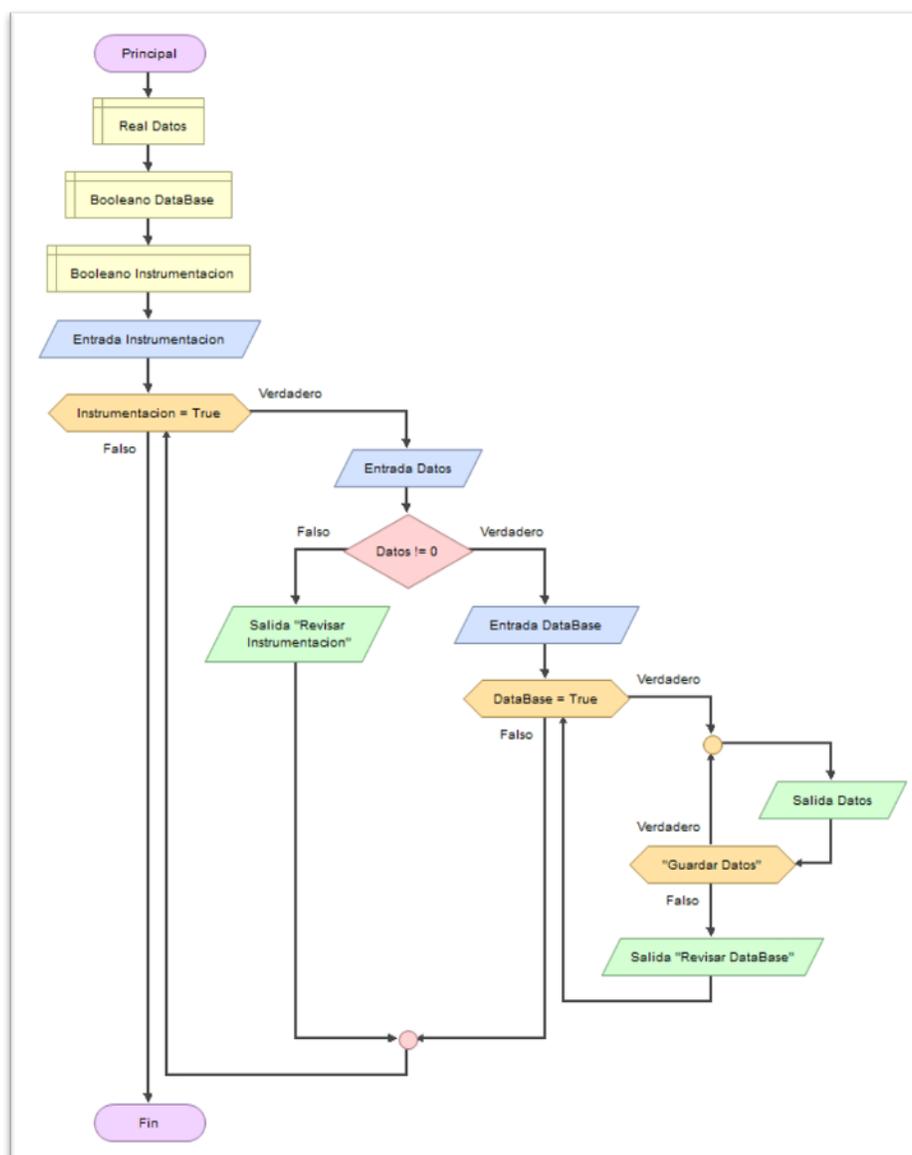


Figura 13. Proceso de los datos desde su captación, hasta la DataBase

Al obtener una lectura óptima de los datos se procedió al envío a través de Wifi, por medio del ejecutable Python al Servidor donde se alojó la aplicación Web (Figura14).



Figura 14. Conexión y envío de datos desde la Raspberry Pi a la DataBase de Firebase

Siendo posible el almacenamiento de los datos, tomados desde la instrumentación, se configuró varios usuarios con acceso a la aplicación, y se realizó la conexión desde diferentes dispositivos, con diferentes formatos de pantalla, para verificar la visualización correcta de los datos.

En el algoritmo de funcionamiento de la aplicación Web, se describe que se debe tener registro de los usuarios y claves de los mismos, pues solo es posible el acceso a los datos como usuario autorizado, de no ser verificado como usuario autorizado, solo tendrá acceso a la interfaz de la aplicación, pero no a la visualización ni descarga de los datos (figura 15).

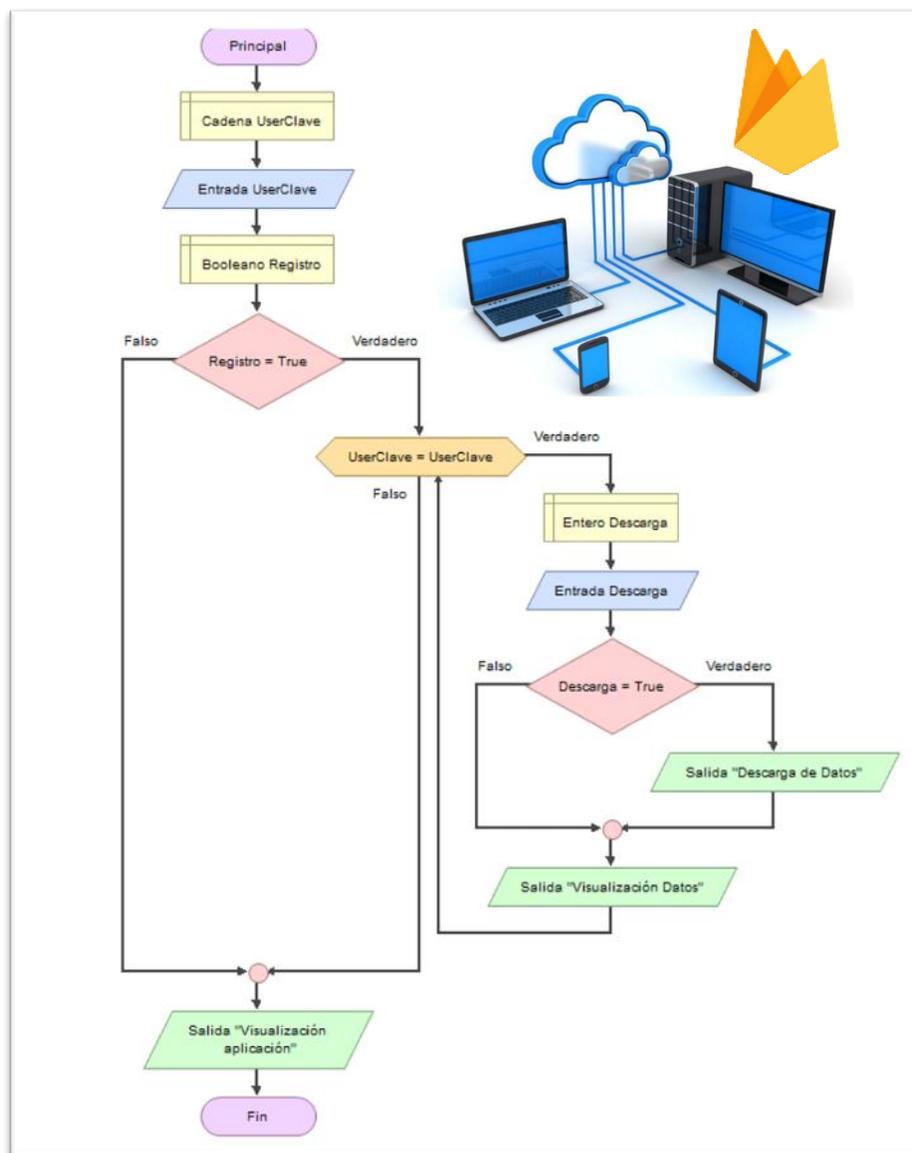


Figura 15. Proceso de interacción del usuario con la aplicación Web

Con el cumplimiento de las respectivas pruebas, se procedió a corrección de errores o modificaciones necesarias, y se puso a prueba el ingreso a la aplicación de los usuarios y la descarga remota de los datos (figura 16).



Figura 16. Interacción y descarga remota de datos, desde aplicación Web

3.2.5 Fase 5. Presentar resultados y beneficios de nuevas tecnologías digitales en instrumentación electrónica y aplicaciones Web para industria 4.0. A continuación el desarrollo de la quinta fase:

3.2.5.1 Actividades. En búsqueda de interesar a la región de Norte de Santander, e incentivar en la ciudad de Cúcuta, a las instituciones académicas, comerciales e industriales, en el uso de instrumentación electrónica, integrada con tecnologías de la información para aplicaciones en módulos de industria 4.0; se buscó eventos internacionales de divulgación científica con énfasis en industria 4.0 y energías renovables, espacios televisivos que promueven la visualización nacional de los desarrollos tecnológicos, ferias regionales de aplicaciones tecnológicas integrando tecnologías de la información, y concursos de innovación en energías renovables, que generen soluciones a las necesidades energéticas de la región.

El desarrollo del proyecto de un “Sistema de supervisión para las variables en los paneles solares, desde la nube”, obtuvo reconocimientos, como: ponencia seleccionada para el III Congreso Internacional de Investigación Aplicada, Creación e Innovación: Desarrollo Sostenible y Tendencias Tecnológicas, una entrevista en vivo para el canal TRO, en la sección de “Mentes ingeniosas” del magazín Café de la Mañana, también un Stand de participación en la tercera versión de la feria “Cúcuta Digital Valley 2021”, considerado el evento de tecnología más importante de la región, organizado por la Cámara de Comercio de Cúcuta, el Clúster Nortic y la Alcaldía de Cúcuta, y llegó a ser finalista en la segunda versión del concurso de creatividad, innovación tecnología y desarrollo de prácticas sostenibles, “INNOVACION 2021, desafíos que iluminan, soluciones que potencian nuestro crecimiento”, organizado por CENS, y sus aliados estratégicos, Secretaria del Banco del Progreso y la Alcaldía de Cúcuta.

4. Resultados

Cumpliendo el desarrollo de las fases elaboradas en la metodología, se obtuvieron resultados que llevaron al cumplimiento de los objetivos del proyecto, siempre considerando los parámetros de diseño, funcionalidad, integración de las plataformas, disponibilidad de las mismas en el mercado, y el valor presupuestal mínimo necesario, para obtener la mayor cantidad de beneficios, al compararse con otras tecnologías. Tomando decisiones que trajeron mayores beneficios, en el desarrollo del proyecto, a los planteados en la propuesta original.

En los resultados obtenidos en la primera fase, se diseñó un método de evaluación de las plataformas, que cuantifico las características de mayor aporte a la investigación, permitiendo conocer la viabilidad de factores como tiempo y precio, donde el de menor valor genera mayores beneficios, y el de mayor valor, genera menores beneficios.

En los resultados de la segunda fase, se descubrió un sensor que ofrecía mayor beneficio al proyecto y se descartó uno que se creían optimo, pero resultó el de menor viabilidad para prototipo.

En la tercera fase se encontró con tecnologías Web que superan en varias áreas las aplicaciones nativas y en la cuarta fase se logró implementar un prototipo funcional que cumplió todos los objetivos.

La fase final permitió la divulgación de los resultados ante diferentes instituciones, y generó mucho interés por diferentes comunidades en el desarrollo de industria 4.0 y energías renovables.

4.1 Recopilar Información

Para consolidar una buena toma de decisiones, se hizo la lectura de diferentes fuentes científicas y de investigación, como una base para el inicio de la etapa de selección que soportó el sistema de supervisión.

Navarrete (2021), en su trabajo titulado “Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi”. Esta investigación de titulación realiza la implementación de sensores inalámbricos con el módulo Esp32 LoRa y por conexión I2C realiza la medición de temperatura y humedad enviándolo por radiofrecuencia al sistema embebido Raspberry Pi que tiene un código en Python que realiza la lectura del puerto serial y envía los datos a una aplicación en la nube llamada Ubidots.

Jiménez & Villarreal (2021), en su trabajo titulado “Desarrollo e implementación de un sistema detector de radiación solar, temperatura y humedad en la empresa Red Digital Telecomunicaciones S.A.S. Girardot”. Este proyecto busca monitorear variables de nivel, temperatura, presión y flujo de la planta didáctica y realizar la comunicación con el usuario encargado de mantenimiento al generar alarmas cuando se superan los rangos establecidos. Las alarmas se verán desde una aplicación Web programada en Linux, Python y lenguaje C.

Montero, Norberto, Passos, Carvalho & Silveira (2020), en su trabajo titulado “Monitoramento em tempo real de temperatura de operação de uma planta FV”. Este artículo desarrolla un sistema de adquisición de datos (SAD) de bajo costo, donde en tiempo real, se realiza la medición de la temperatura de una planta fotovoltaica en operación conectada a la red eléctrica del laboratorio de energías alternativas (LEA) de la “Universidade Federal do Ceará (UFC)”; donde se usa la placa Raspberry Zero W como centro de datos de los diferentes

dispositivos.

Agurto (2020), en su trabajo titulado “Integración de un sistema de adquisición de datos mediante el uso de un Arduino Mega y Raspberry Pi3 como servidor Web y Base de Datos”. Este trabajo tiene como objetivo desarrollar un sistema de adquisición de datos de propósito general para industria o investigación, donde puedan ser guardados los datos en una base de datos y se tenga un entorno Web de visualización de los mismos, usando tecnologías libres: Arduino, Modulo wifi ESP8266 y Raspberry Pi3.

Soares, Carneiro & Oliveira (2020), en su trabajo titulado “Desenvolvimento de um registrador de dados para a aplicação em um sistema de monitoramento fotovoltaico”. Esta investigación desarrolla un sistema de captación de datos soportado en tecnología Raspberry Pi3, y con el cual se toman las variables de interés en una planta fotovoltaica y generando un histórico de datos que sentaran las bases para futuras investigaciones o mejoras del sistema.

Ezequiel, Villalta, Clara & Fuster (2019), en su trabajo titulado “Diseño del sistema de alimentación basado en energía solar de una estación de bombeo con control remoto”. El objetivo de este proyecto es realizar un sistema autónomo de bombeo fotovoltaico donde por medio de una aplicación Web puedan ser conocidas las variables del sistema, y controlar el giro de motor de la bomba.

Silvente (2019), en su trabajo titulado “Sistema de bajo coste para la monitorización de instalaciones fotovoltaicas: Aplicación de Arduino y Raspberry Pi”. Este artículo evalúa un sistema inalámbrico de monitorización de bajo costo basado en soluciones de código abierto, el cual permite monitorizar instalaciones fotovoltaicas a nivel del módulo fotovoltaico, brindando información detallada sobre el rendimiento de la planta de energía fotovoltaica.

Arquillo (2019), en su trabajo titulado “Gestión y monitorización de instalaciones eléctricas trifásicas mediante Arduino y Raspberry Pi”. Este trabajo presenta la monitorización de sistemas trifásicos en tiempo real, por medio de Hardware Open Source donde se puede estudiar el sistema trifásico y cada una de sus fases por separado con sensores de corriente no invasivos conectados a Arduino y Raspberry Pi.

Raafiu & Agus (2018), en su trabajo titulado “Smart Monitoring of Solar Panel System in Saving of the Electrical Power with Internet of Things”. Este trabajo presenta la monitorización de sistemas trifásicos en tiempo real, por medio de Hardware Open Source donde se puede estudiar el sistema trifásico y cada una de sus fases por separado con sensores de corriente no invasivos conectados a Arduino y Raspberry Pi.

Barón (2017), en su trabajo titulado “Monitorización de la energía consumida mediante Raspberry Pi para sistema domótico”. El objetivo de este proyecto es realizar un sistema de monitorización de energía consumida para domótica usando Raspberry Pi como centro de datos de 10 sensores de corriente simultáneamente.

González (2017), en su trabajo titulado “Automatización de un módulo fotovoltaico rotatorio de un eje con disponibilidad de datos de manera remota”. Esta investigación diseña un seguidor solar de eje, con disponibilidad de datos de forma remota, donde se podrá comparar la mayor radiación del Sol con respecto a una posición de referencia de 0° .

Larrea (2016), en su trabajo titulado “Registrador de datos de bajo coste y acceso remoto (Datalogger)”. Este proyecto usa sistemas embebidos de bajo coste para desarrollar un registrador de datos multicanal con una tasa de muestreo de 1khz/canal y de almacenamiento local, pero de administración remota.

4.2 Diseño del Sistema de Instrumentación

Para la definición de los componentes utilizados en el proyecto, se tomaron varios parámetros a considerar, y se calificaron en 5 puntuaciones que son definidas en porcentajes de 20% - 100%, en un rango de 1 – 5, donde cada uno tiene como categoría, Deficiente – Excelente. Método basado en la matriz de Pugh en Sejzer (2022), donde se tomaron las características de mayor aporte al proyecto y categorizo el resto cuantitativamente, en porcentaje de viabilidad en función de la plataforma de mayor beneficio, en cada categoría (ver tabla 1).

Tabla 1. Calificación de puntuación de viabilidad

Categoría	Deficiente	Malo	Regular	Bueno	Excelente
Calificación	1	2	3	4	5
Porcentaje (%)	0-20	20-40	40-60	60-80	80-100

Como primer paso, se seleccionó la plataforma de hardware en la que se desarrolló el diseño, ya que, este proyecto debía integrar la instrumentación, con tecnologías de la información. Se analizó la posibilidad de basar el diseño sobre microcontroladores, microprocesadores en ordenadores de placa simple, o plataformas computacionales completas; siendo las plataformas computacionales, demasiado costosas para su uso, pues no es viable usar un computador en sitio, para labores que pueden ser realizadas por equipos más económicos y eficientes.

Las plataformas de microcontroladores, fueron una idea atractiva, porque son plataformas económicas, de disponibilidad inmediata, y con sensores de fácil acoplamiento al dispositivo, para la captación, gestión y transferencia de datos; adicionalmente existen algunas con conexiones IoT, por medio de sockets, a servidores ya destinados a esta función. Pero los servidores públicos, no ofrecen seguridad o permanencia del servicio, y los servidores privados

requieren un pago que no es contemplado como posibilidad en el desarrollo de este proyecto.

Como opción final se seleccionó a los ordenadores de placa simple, sistemas soportados en microprocesadores, que ofrecen las prestaciones mínimas de plataformas computacionales, pero con un costo mucho menor y en formato embebido para aplicaciones en sitio. Cabe destacar que estos dispositivos cuentan con sistemas operativos y plataformas de conexiones a servidores, e incluso con la posibilidad de la creación de un servidor propio, sobre la misma plataforma, abaratando costos, y permitiendo el control total de las partes el proyecto.

Para tener una comparativa general entre diferentes dispositivos y fabricantes, se seleccionó a Mouser Electronics ®, proveedor autorizado a nivel mundial de semiconductores y componentes electrónicos, de las principales marcas (Mouser Electronics Colombia, 2021).

Las comparativas de los dispositivos son de acceso público, en español, y con énfasis en Colombia, o en el país de interés, lo que permitió una selección de fabricantes, componentes, tecnologías, costos, y tiempo de entrega, como se visualiza en la figura 17 (Mouser Electronics Colombia, 2021).

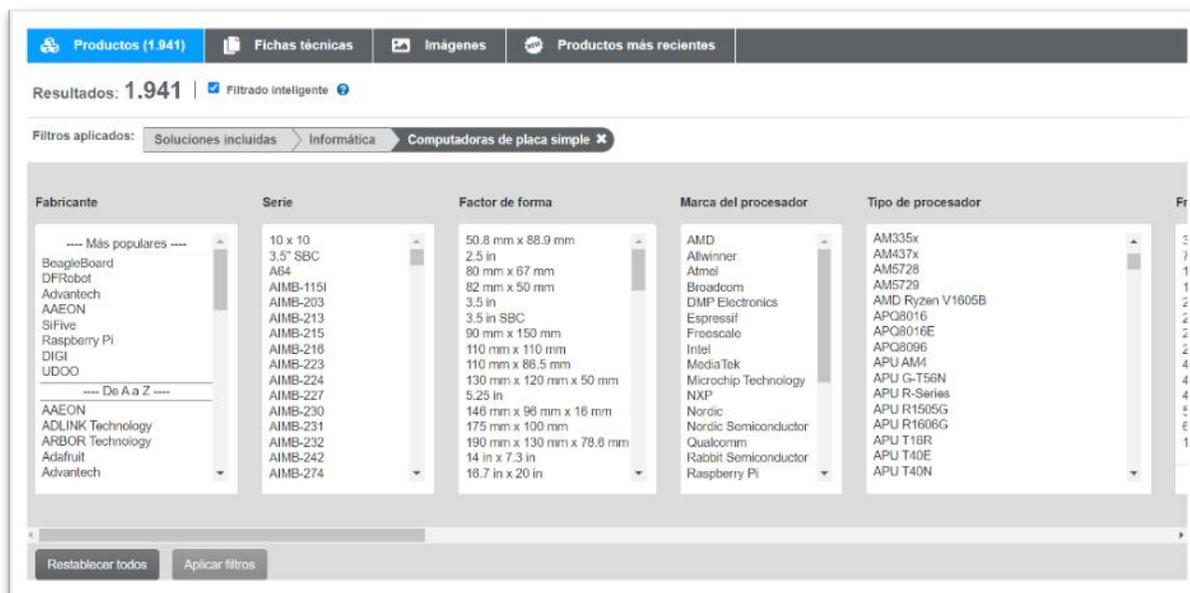


Figura 17. Filtro de aplicación Web para comparativas de Mouser Electronics ®

Fuente: Mouser Electronics Colombia, 2021.

En las diferentes opciones de los fabricantes, se encontraron plataformas demasiadas costosas para suplir el Datalogger HOBO® UX120-006M, que tiene un costo aproximado de 1 millón de pesos colombianos (Larrea, 2016; Sejzer, 2022). Se seleccionaron 5 opciones en precios inferiores a ese dispositivo, pero que, ofrecen una plataforma sólida para desarrollos IoT, industria 4.0 y conexión con tecnologías de la información, por medio APK y SDK, a servidores de Backend.

4.2.1 Selección de sistema de captación de datos. Ya seleccionada la plataforma computacional sobre la que se van a realizar la conexión del sistema con el servidor, se evalúa la posibilidad de conectar la instrumentación directamente con este dispositivo, lo que concluye que es posible solo bajo un circuito adaptador de la señal analógica de los sensores que sean necesarios usar; esto por el hecho, de que el microprocesador no consta de un conversor ADC

propio, y debe valerse de un circuito externo que cumpla esta función, comunicando por el puerto serial los datos de los sensores al microprocesador.

Como posibles opciones se tiene tres categorías evaluadas, que son circuito conversores análogo/digital, plataformas embebidas con conversores ADC integrados, y la posibilidad de diseñar uno, para los requerimientos del sistema, tomándose como decisión los términos de economía, viabilidad de canales de medición, facilidad de integración con la plataforma Raspberry Pi, y la posibilidad de gestión de datos y acciones, antes de ser enviados al microprocesador.

Se descartó la posibilidad de diseñar un conversor análogo/digital, pues esto llevaría a nuevos estudios, tiempos de desarrollo, pruebas y además costos, que están fuera de la viabilidad del proyecto. También se descartó la posibilidad de utilizar circuitos conversores análogo/digital, que, aunque ofrecían canales de conversión a la Raspberry pi, estos canales suelen ser limitado a un total de 2, 4 y máximo 8 canales, incrementando su precio en cada formato, y llegando a triplicar el costo, de opciones más económicas y viables, como plataformas embebidas con conversores análogos internos, incluso no cuentan con la posibilidad de realizar acciones diferentes a la conversión.

En la búsqueda de hardware sobre la cual basar el desarrollo del prototipo, se descartó las plataformas de microcontroladores como posibilidad de conexiones directa de la instrumentación, con tecnologías de la información, pues estos sistemas no permiten manejo o conexión con servidores remotos, ni la instalación de sistemas operativos computacionales que gestione los servicios de Backend. Pero se reconocieron como excelentes y económicas plataformas, para la captación, gestión y envío de datos.

Por eso, se tomó para comprar las plataformas: NodeMCU ESP8266, Arduino MKR1000, Arduino UNO R3, y Arduino Mega 2560, las más utilizadas en los proyectos de Raspberry Pi. Se descarto la WeMos D1 R2, por usar en microcontrolador ESP8266 y ofrecer las mismas características del NodeMCU.

4.2.1.1 Calificación del microcontrolador. Siendo un requerimiento la captación, tratamiento y envío de datos, mediante comunicación serial, todos los microcontroladores cumplen con las condiciones básicas de funcionamiento, por lo que recibieron una calificación de 5, y se enfocó la comparativa en sus diferencias, como la cantidad de canales análogos, posibilidad de comunicación de varios protocolos, precio y disponibilidad del dispositivo. Buscando, la mayor relación costo benéfico en el proyecto (ver tabla 2).

Tabla 2. Calificación de microcontrolador

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Calificación	5	5	5	5	5

4.2.1.2 Calificación de comunicación del microcontrolador. Como parámetro más importante, es que el microcontrolador, tenga comunicación serial de fácil adaptabilidad a la Raspberry Pi, y aunque es importante la conexión WIFI, el microprocesador, cumplirá con esta función, limitando al microcontrolador, como un dispositivo esclavo de captación y conversión de datos. También se tuvo en cuenta la posibilidad de los protocolos SPI, I2C y UART, como posibles conexiones de ser necesarias, pues se podrían adaptar pantallas LCD, para observar las variables en sitio (ver tabla 3).

Tabla 3. Valorización de comunicación de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
WIFI	1	1	1	0	0
USB / Serial	1	1	1	1	1
SPI	1	1	1	1	1
I2C	1	1	1	1	1
UART	0	1	1	1	1
TOTAL	4	5	5	4	4

Como el valor de protocolos evaluados es 5, este representó la categoría del 100%, y los demás se catalogan según su porcentaje comparado con este valor.

$$\text{Clasificación} = \frac{\text{valor obtenido}(100\%)}{\text{valor máximo}}$$

$$\text{NodeMCU ESP8266} = \frac{4(100\%)}{5} = 80\%$$

$$\text{Arduino WIFI Rev2} = \frac{5(100\%)}{5} = 100\%$$

$$\text{Arduino MKR1000} = \frac{5(100\%)}{5} = 100\%$$

$$\text{Arduino UNO R3} = \frac{4(100\%)}{5} = 80\%$$

$$\text{Arduino Mega 2560} = \frac{4(100\%)}{5} = 80\%$$

Con los porcentajes de cada plataforma, se realizó la clasificación, según la tabla 6, y considerando cada porcentaje, se le dio su calificación (ver tabla).

Tabla 4. Calificación de comunicaciones de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Calificación	4	5	5	4	4

4.2.1.3 Calificación de canales analógicos del microcontrolador. Para poder realizar un diseño competente con los ofrecidos en el mercado, un gran aporte, es una cantidad superior de canales de medición, y así ofrecer al usuario, la posibilidad de conectar mayor cantidad de sensores, con la misma plataforma de conexión a la nube, y bajo la misma aplicación; por eso, se analizó la cantidad de lecturas analógicas soportadas por los microcontroladores (ver tabla 5).

Tabla 5. Valorización de los canales analógicos de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Canales ADC	1	7	6	6	16
Canales I/O	9	8	14	14	54
TOTAL	10	15	20	20	70

Se visualiza la totalidad de los pines, pero evalúa solo la cantidad de pines analógicos, y como el valor máximo de canales analógicos es 16, este representó la categoría del 100%, y los demás se catalogan según su porcentaje comparado con este valor.

$$\text{Clasificación} = \frac{\text{valor obtenido}(100\%)}{\text{valor maximo}}$$

$$\text{NodeMCU ESP8266} = \frac{1(100\%)}{16} = 6.25\% \quad \text{Arduino WIFI Rev2} = \frac{6(100\%)}{16} = 37.50\%$$

$$\text{Arduino MKR1000} = \frac{7(100\%)}{16} = 43.75\% \quad \text{Arduino UNO R3} = \frac{6(100\%)}{16} = 37.50\%$$

$$\text{Arduino Mega 2560} = \frac{16(100\%)}{16} = 100\%$$

Con los porcentajes de cada plataforma, se realizó la clasificación, según la tabla 6, y considerando cada porcentaje, se le dio su calificación (ver tabla).

Tabla 6. Calificación de los canales ADC

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Calificación	1	3	2	2	5

4.2.1.4 Calificación de disponibilidad del microcontrolador. De la misma forma que se evaluó la disponibilidad de las plataformas de desarrollo computacional en los mercado local, regional o internacional, se analizó el tiempo de disponibilidad para los microcontroladores (ver tabla 7).

Tabla 7. Valorización de disponibilidad de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Disponibilidad	1 día	15 días	15 día	1 día	1 día

Retomando los cálculos realizados en la disponibilidad de las plataformas computacionales, se utilizaron de la misma forma en la disponibilidad de los microcontroladores; y con el valor máximo de días permitidos reducido a 30 días, porque el microcontrolador es importante en la etapa inicial de proyecto.

$$\text{Clasificación} = \left(1 - \frac{\text{valor días}}{\text{valor maximo días permitidos}}\right) 100\%$$

$$\text{NodeMCU ESP8266} = \left(1 - \frac{1}{30}\right) 100\% = 99.67\%$$

$$\text{Arduino MKR1000} = \left(1 - \frac{15}{30}\right) 100\% = 50\%$$

$$\text{Arduino WIFI Rev2} = \left(1 - \frac{15}{30}\right) 100\% = 50\%$$

$$\text{Arduino UNO R3} = \left(1 - \frac{1}{30}\right) 100\% = 99.67\%$$

$$\text{Arduino Mega 2560} = \left(1 - \frac{1}{30}\right) 100\% = 99.67\%$$

Con los porcentajes de cada plataforma, se realizó la clasificación, según la tabla 6, y considerando cada porcentaje, se le dio su calificación (ver tabla).

Tabla 8. Calificación de disponibilidad de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Calificación	5	3	3	5	5

4.2.1.5 Calificación de precio del microcontrolador. La limitante de costos del proyecto, obliga a viabilizar una tecnología según su precio, por lo que, se evaluó este parámetro como importante en la selección del microcontrolador (ver tabla 9).

Tabla 9. Valorización de disponibilidad de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Precio	\$35.000	\$162.039	\$191.036	\$117.265	\$171.847

Cabe destacar que el precio menor, es el más favorable, y sobre este, se calificaran los demás, dejando como precio máximo permitido, un total de \$200.000 pesos colombianos; lo que justifica la compra del artículo más costoso, y un 15% por costos de envío, o accesorios de ser necesario.

$$\text{Clasificación} = \left(1 - \frac{\text{Precio artículo}}{\text{Precio máximo permitido}}\right) 100\%$$

$$\text{NodeMCU ESP8266} = \left(1 - \frac{35\,000}{200\,000}\right) 100\% = 82.50\%$$

$$\text{Arduino MKR1000} = \left(1 - \frac{162\,039}{200\,000}\right) 100\% = 18.98\%$$

$$\text{Arduino WIFI Rev2} = \left(1 - \frac{191\,036}{200\,000}\right) 100\% = 4.48\%$$

$$\text{Arduino UNO R3} = \left(1 - \frac{117\,265}{200\,000}\right) 100\% = 41.37\%$$

$$\text{Arduino Mega 2560} = \left(1 - \frac{171\,847}{200\,000}\right) 100\% = 14.07\%$$

Con los porcentajes de cada plataforma, se realizó la clasificación, según la tabla 6, y considerando cada porcentaje, se le dio su calificación (ver tabla 10).

Tabla 10. Calificación de disponibilidad de los microcontroladores

Plataforma	NodeMCU ESP8266	Arduino MKR1000	Arduino WIFI Rev2	Arduino UNO R3	Arduino Mega 2560
Calificación	5	1	1	3	1

4.2.1.6 Selección de microcontrolador. Cuantificando las características de los diferentes microcontroladores, se pudo realizar una tabla comparativa y cuantificada que justificó la mejor selección (ver tabla 11).

$$\text{Promedio} = \frac{\text{Controlador} + \text{Comunicación} + \text{ADC} + \text{Disponibilidad} + \text{Precio}}{5}$$

Tabla 11. Selección de sistema de captación de datos

Plataforma	NodeMCU ESP8266	Arduino WIFI Rev2	Arduino MKR1000	Arduino UNO R3	Arduino Mega 2560
Microcontrolador	5	5	5	5	5
Comunicación	4	5	5	4	4
Canales ADC	1	3	2	2	5
Disponibilidad	5	3	3	5	5
Precio	5	1	1	3	1
Promedio	4	3.4	3.2	3.8	4

Categorizando y evaluando las características de interés que generan aportes al proyecto, se logró determinar a las plataformas Arduino Mega 2560 y NodeMCU ESP8266, como las plataformas de mayor viabilidad para usarse como sistema de captación de datos que conecte con la Raspberry Pi.

Se debe resaltar, que el NodeMCU ESP8266, es viable, solo cuando se usa una medida analógica, o es necesario la conexión esclavo maestro de varios al mismo tiempo; algo que aumenta los costos, y el desarrollo del proyecto. En cambio, el Arduino Mega 2560, a pesar de ser mucho más costoso, contiene una cantidad elevada de canales analógicos que ofertan un atractivo mayor a la instrumentación.

Como otra opción viable, está el Arduino UNO R3, que tiene características moderadas comparadas con el Arduino Mega 2560, y un incremento de 6 veces los canales de medición del NodeMCU ESP8266, lo que se oferta como una solución viable cuando se necesitan hasta un máximo de 6 mediciones analógicas.

Como apartado final de selección, cabe destacar que existen opciones compatibles de otros fabricantes, que usan las mismas tecnologías, y distribuyen una versión igual en características, pero más económica que la original; lo que reduce el costo de adquisición de una plataforma

Arduino Mega 2560 a \$70.000, y un Arduino Uno R3 a \$40.000, un precio mucho menor, que la mitad de la tarjeta original, pero con la misma funcionalidad y programación.

4.2.2 Análisis de sensores. El estudio previo de viabilidad de la plataforma de desarrollo Raspberry Pi 3B+, y el sistema de captación de datos Arduino Mega 2560, facilitó, seleccionar dispositivos compatibles que, ofrecen una gamma de tecnologías para la medición y envío de datos a bajo coste; y partiendo de esta premisa, se optó por utilizar los sensores de voltaje y corriente ofrecidos para la plataforma Arduino, los cuales pueden ser utilizados tanto en la placa Arduino Mega 2560, como la Arduino UNO R3, o incluso en la NodeMCU ESP8266, y obteniendo resultados replicables en cualquiera de las plataformas (ver tabla 12).

Tabla 12. Datos de los sensores

Sensor	Voltaje FZ0430	Voltaje SKU DFR0051	Corriente SCT013 YHDC	Corriente KS0185
Resolución	0.00489 V	0.00489 V	0.00489 V/A	0.185 V/A
Precisión	10Bits	10Bits	± 1%	± 1%
Rango	0.02445 – 25V	0.02445 – 25V	0 – 5Arms	-5A – 5A
Sensibilidad	0.0215846	0.0214851	~	0.185
Disponibilidad	Inmediata	30 días, bajo pedido	3 días, bajo pedido	Inmediata
Precio	\$ 5.000	\$ 20.000	\$ 49.000	\$ 21.000

En la selección de los sensores, al revisarse la documentación del sensor de corriente SCT013 YHDC, se descubrió que no permite la lectura de corriente directa, lo que es obligatorio para la lectura de la corriente generada por los paneles fotovoltaicos, que trabajan en corriente directa, algo que inmediatamente descarto la posibilidad de uso de estos sensores en el proyecto.

Estos sensores por su construcción en un formato de transformador reductor, con relación de espiras 1:2000, que aprovecha el efecto Hall generado en un conductor cuando es atravesado por corriente alterna, se puede recomendar como una solución para la lectura de cargas inductivas,

pero fue considerado demasiado costoso para ser práctico en un proyecto, pues la disponibilidad, es deficiente, ya que, en el mercado local, solo se encontró la versión con rango de medición de 0 - 100A, un rango demasiado elevado para ser utilizado en el diseño, y la versión 0 - 5A solo es disponible bajo pedido.

El formato de transformador reductor del SCT013 YHDC, no acondiciona la señal para ser leída por los microcontroladores, lo que incrementa el costo de diseño, e implementación. Se necesita una etapa de acondicionamiento que debe diseñarse, ajustarse e implementarse, con dispositivos discretos, incrementando el costo de implementación (ver figura 18).

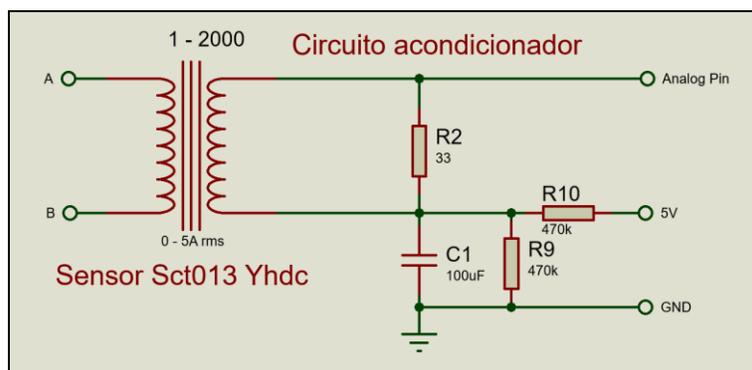


Figura 18. Circuito mínimo necesario para el funcionamiento del SCT013 YHDC

Al descartarse el sensor SCT013 YHDC, como sensor de corriente, se buscó una opción de mayor viabilidad, para tener la posibilidad de leer corriente generada en los paneles, por la que la de mayor viabilidad, fue el sensor KS0185, el cual se encuentra disponible en el mercado local, y tiene un precio mucho menor precio del SCT013 YHDC, se vende en un formato embebido que ya cuenta con el circuito acondicionador de la señal, y permite la lectura de corriente directa y alterna en la misma plataforma, lo que da mayor flexibilidad de lecturas al sensor (ver figura 19).

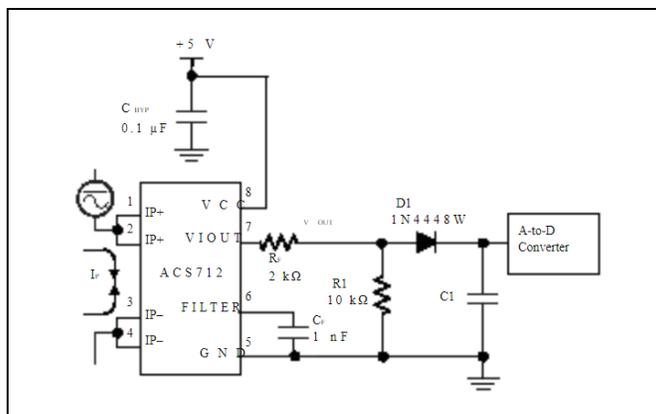


Figura 19. Circuito lector de corriente del KS0185

Fuente: All Data Sheet, 2021.

El sensor KS0185, es un circuito de efecto Hall, que es afectado por los diferentes campos electromagnéticos, y esto debe ser considerado en su uso, ya que, puede traer problemas de calibración en el momento de su instalación.

Con la selección de sensores de voltaje para la plataforma Arduino, se procedió a realizar el diseño en el software de simulación Proteus Design Suite 8.10 ©, basado en las características del sensor (ver figura 20).

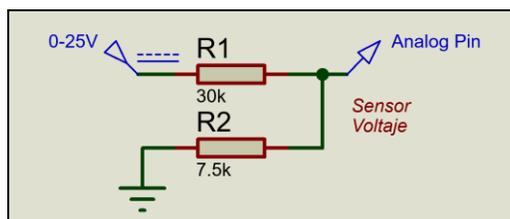


Figura 20. Circuito de sensores de voltaje, FZ0430 y SKU DFR0051

Con el voltaje máximo de medida, se configura la lectura del sensor en el microcontrolador; se debe destacar que la construcción de los dos sensores es similar, por lo que el comportamiento ideal es el mismo para los dos, y su código de lectura también (ver figura 21).

```

/*
 * Sensores de voltaje
 * FZ0430 y SKU DFR0051
 */
float sensor = 0; //Lectura del pin A0
float lectura = 0; //Conversión a voltaje
void setup() {
  Serial.begin(9600);
}

void loop() {
  sensor = analogRead(A0);
  lectura = (0.02445)*sensor;
  Serial.println(lectura);
  delay(500);
}

```

Figura 21. Código en Arduino para la lectura analógica de los sensores de voltaje

Al tener un código de lectura, se debió comprobar su funcionamiento sobre la plataforma Arduino, por lo que la simulación en Proteus Design Suite 8.10 © fue una excelente opción.

Para la medición, fue necesario un circuito que ofreciera la variación del voltaje en función de la radiación que recibe; por esto, se diseñó un circuito siguiendo el modelo electrónico de diodo sencillo con resistencia Shunt del panel, y emulando la radiación por medio de un divisor de voltaje con el TORCH_LDR, para trabajar la dependencia de la radiación (ver figura 22).

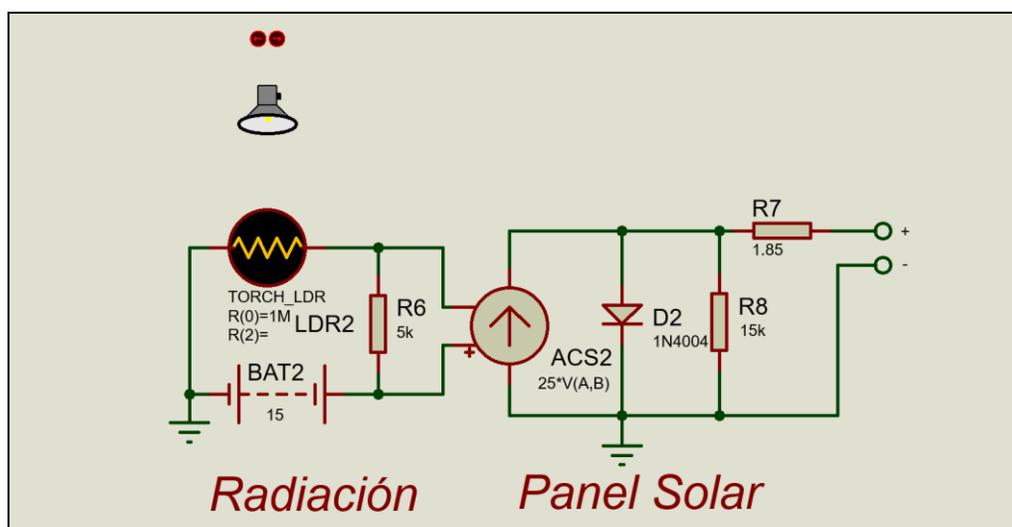


Figura 22. Modelo de diodo sencillo, con resistencia Shunt, y dependencia de la radiación

4.2.2.1 Simulación de captación de datos. Como se necesitaba emular los voltajes variables en un rango de 0V a 16V, se usó interpolación en los valores comunes de las resistencias, para encontrar la combinación adecuada, que permitieran la salida de voltajes de ese rango.

Con los valores encontrados, se pudo verificar la viabilidad de la lectura de voltaje, simulando la captación de datos de salida del panel, y comprobándolos con la tabla de la comunicación serial del Arduino configurado con el código de lectura, y un medidor de voltaje en la entrada del sensor (ver figura 23).

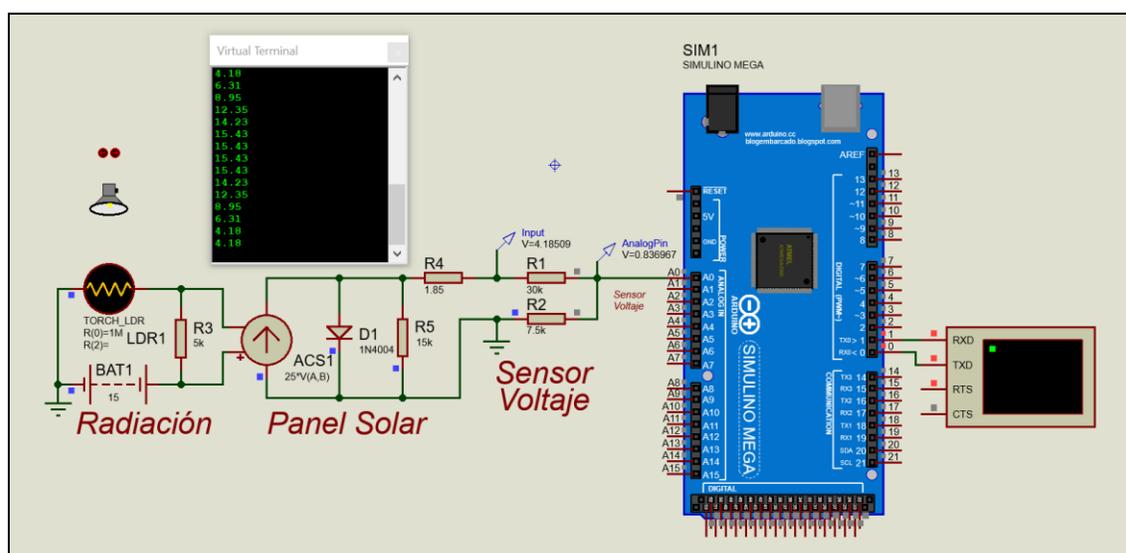


Figura 23. Lectura analógica del sensor de voltaje

Cumpliendo con el diseño de la lectura de voltaje, se procedió a verificar la lectura de corriente, usando el sensor que provee el software (ver figura 24).

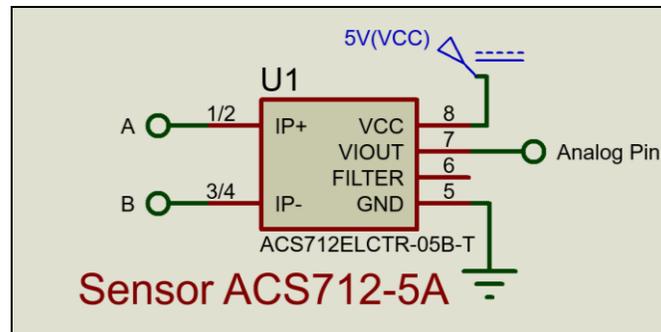


Figura 24. Sensor de corriente KS0185

Tomando como referencia la documentación sobre el sensor, se encuentra que ofrece una salida de 2.5V en una lectura de 0A, y se debe considerar en la elaboración del código de lectura (ver figura 25).

```

/*
 * Sensor de corriente
 * ACS712-5A
 */
float sensor = 0; //Lectura del pin A5
float lectura = 0; //Conversión a corriente
void setup() {
  Serial.begin(9600);
}
void loop() {
  sensor = analogRead(A1) * (5.0/1024.0);
  lectura = (sensor-2.5)/0.185;
  Serial.println(lectura, 4);
  delay(500);
}

```

Figura 25. Código en Arduino para la lectura analógica del sensor de corriente

Usando el diseño del panel, se acondiciono una carga de prueba de 100Ω, para generar una corriente en el orden de los miliamperios, y comprobar la lectura del sensor (ver figura 26).

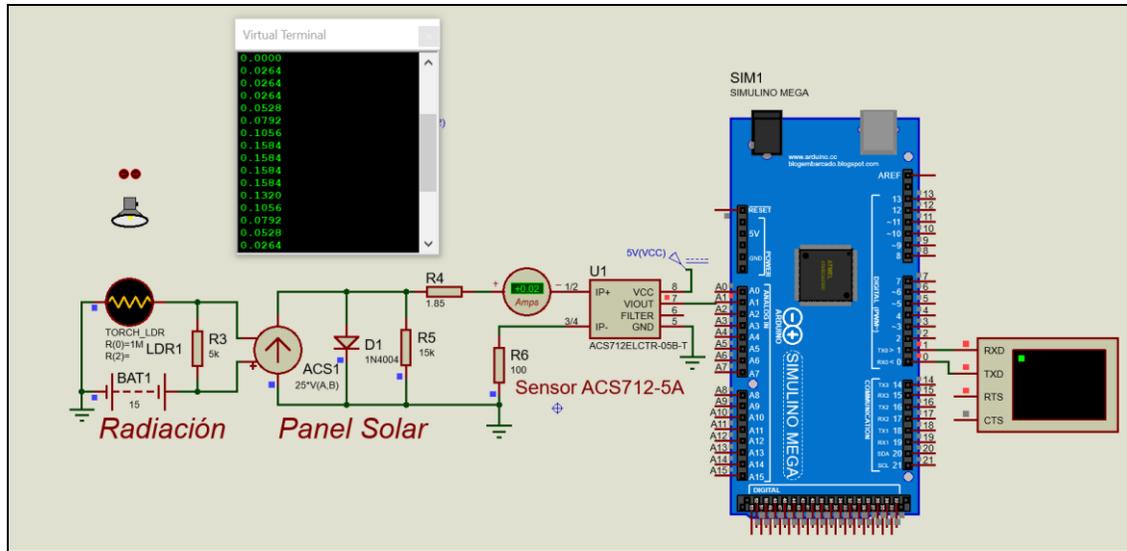


Figura 26. Lectura analógica del sensor de corriente

4.2.2.2 Configuración de Raspberry Pi con base de datos. Al tener funcionamiento de la lectura de los sensores, se procedió a la configuración de la plataforma computacional, que administrará el envío y conexión con la base de datos.

Como opción viable para el diseño, se encontró en la documentación de Raspberry Pi, que es posible su emulación informática en una versión de escritorio de Debian, soportada sobre máquina virtual, de la que se tomó como plataforma Oracle VM VirtualBox®, y donde se configura una máquina Linux, en la que se realiza la instalación del sistema operativo de la Raspberry Pi, y se configura para la conexión con la base de datos (ver figura 27).

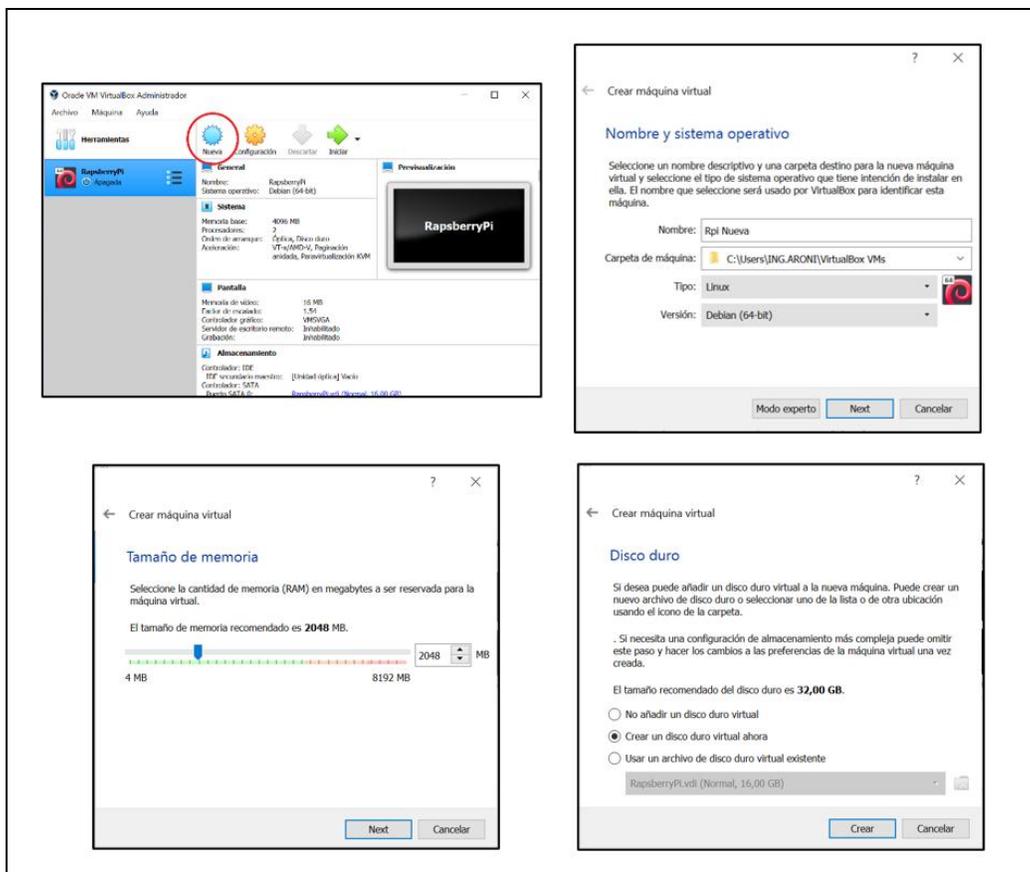


Figura 27. Creación de entorno Linux, para SO Debian de Raspberry Pi

Completando la instalación de Raspberry Pi en VM VirtualBox®, se le debió facilitar acceso a la red para actualizar y configurar la placa con las actualizaciones sugeridas (ver figura 28).

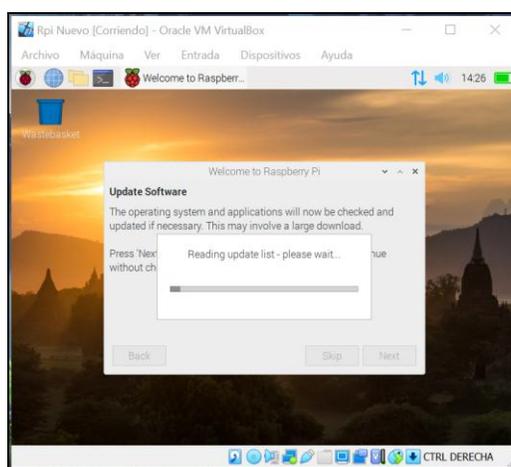
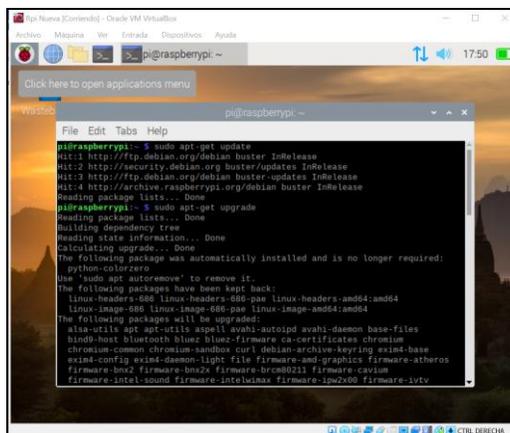


Figura 28. Actualización de Raspberry Pi

De no actualizarse por sugerencia misma de la placa, se puede realizar manualmente, desde la consola, con los comandos **apt-get** de actualización y descarga (ver figura 29):

\$ sudo apt-get update //Actualiza el repositorio

\$ sudo apt-get upgrade //Descarga actualizaciones



```

pi@raspberrypi:~$ sudo apt-get update
Hit:1 http://ftp.debian.org/debian buster InRelease
Hit:2 http://security.debian.org/debian-security InRelease
Hit:3 http://ftp.debian.org/debian buster-updates InRelease
Hit:4 http://archive.raspberrypi.org/debian buster InRelease
Reading package lists... Done
pi@raspberrypi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
python-colorzoro
Use 'sudo apt autoremove' to remove it.
The following packages have been kept back:
linux-headers-686 linux-headers-686-pae linux-headers-amd64:amd64
linux-image-686 linux-image-686-pae linux-image-amd64:amd64
The following packages will be upgraded:
alsa-utils apt apt-utils aspell avahi-autoipd avahi-daemon base-files
bind9-host bluetooth bluez bluez-firmware ca-certificates chromium
chromeos-common chromeos-sandbox curl debian-archive-keyring exim4-base
exim4-config exim4-daemon-light file firmware-amd-graphics firmware-atheros
firmware-brcm43xx firmware-brcm80211 firmware-devic
firmware-intel-sound firmware-intellinux firmware-ipw2x00 firmware-ivy

```

Figura 29. Actualización manual de Raspberry Pi

4.2.3 Plataforma de backend. En el momento de la selección de componentes, se dejó claro, que se necesitaba una plataforma de hardware que soportase un sistema operativo con la posibilidad de conexión a servidores, o incluso de la creación de uno propio, para la gestión de los datos en la nube.

La selección de la plataforma Raspberry Pi, fue un acierto en cuanto lenguajes de programación, sistema operativo, documentación, e integración con otras plataformas, se refiere; pues gracias a su comunidad Open Source, se ha podido establecer conexiones directas con plataformas robustas y seguras de desarrollo Backend; lo que permitió la gestión de los datos en la nube, sin la necesidad de la creación de servidores propios, o pago de licencias para los servidores privados destinados a esta función. Algo que cumple totalmente el objetivo

presupuestal de este proyecto.

Por eso se consultó, que servicios de Backend, con estas cualidades se ofrecían en el mercado, logrando conseguir la plataforma de desarrollo Firebase, la cual es adquisición de Google, y ofrece bajo sus servidores, los servicios necesarios para la gestión de Backend multiplataforma, y los servicios gratuitos de dominio, hosting, autenticación, almacenamiento, entre otros.

Todos los servicios de Firebase, son ofertados gratuitamente con una cuenta de Gmail, y tiene la documentación y SDK necesarias para la conexión multiplataforma. Para iniciar los servicios, se debió crear el proyecto desde la consola de Firebase con la cuenta Gmail, y darle el nombre al proyecto, el cual sirvió posteriormente como dominio de la aplicación Web (ver figura 30).

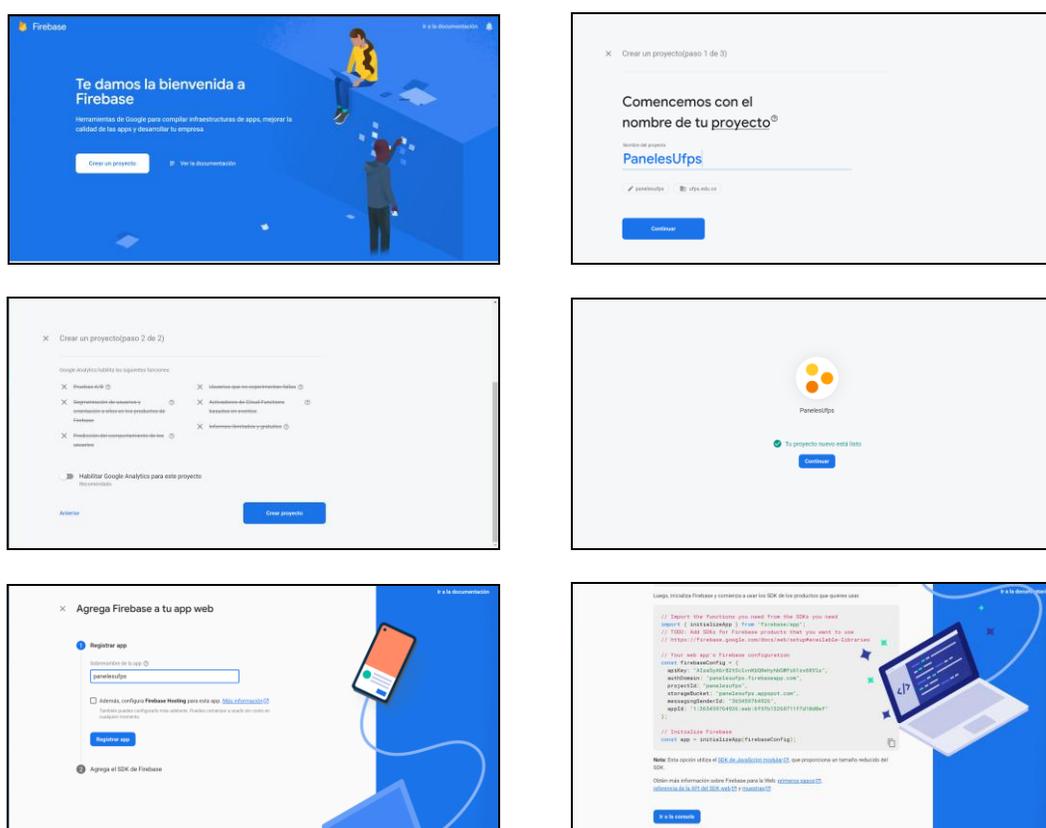


Figura 30. Creación de proyecto en consola de Firebase

Con la creación del proyecto, se tuvo acceso, a los diferentes servicios ofrecidos en esta plataforma, y en la lectura de la documentación, se especifica qué tipo de plataforma, y lenguajes de programación se desean utilizar, para que Firebase, ofrezca las claves y documentación necesaria para la conexión (ver figura 31).

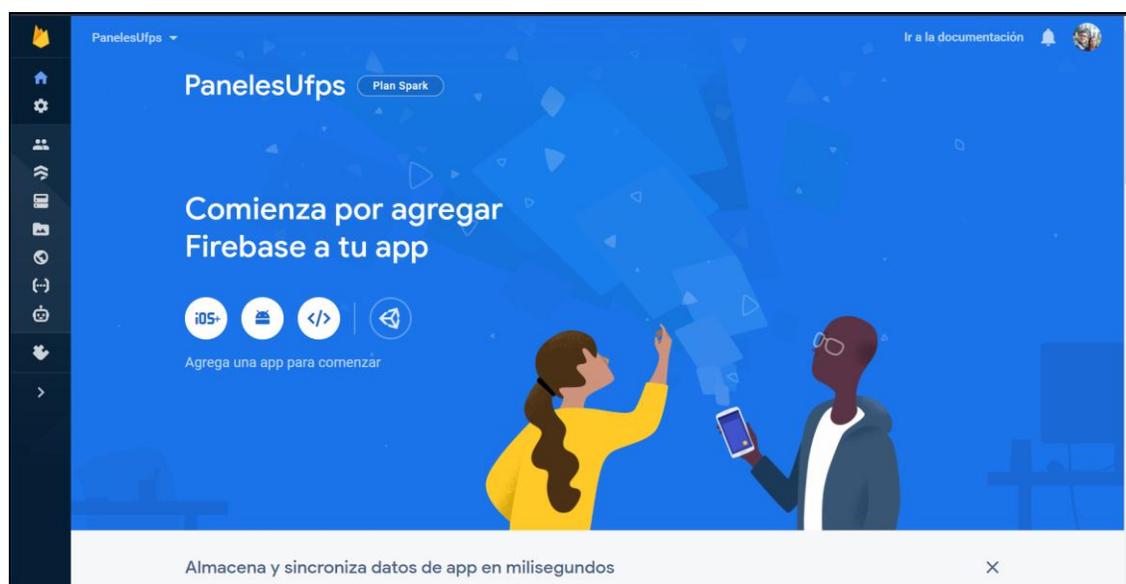


Figura 31. Consola del App, para los servicios de Firebase

Firestore DataBase y Realtime DataBase, ambas con almacenamiento en formato Json, y que, para función de este proyecto, se seleccionó Realtime DataBase; por ser una base de datos asíncrona, que conserva los datos almacenados, sin perder la integridad de la aplicación cuando se desconecta el sistema computacional que alimenta la información.

Para iniciar Realtime DataBase, fue necesario seleccionar el servidor físico que almacenara la información, recordando que son propiedad de Google, y gracias a ello, tendrá toda la seguridad y soporte ofrecidos por este proveedor, sin ningún costo.

Para efectos prácticos se seleccionó la posibilidad de lectura y escritura de terceros, solo con acceso de las claves de referencias ofrecidas por el proveedor, pero se destaca que la seguridad de la base de datos, se puede configurar desde la consola de Firebase, para ser estrictos en la conexión con la misma (ver figura 32).

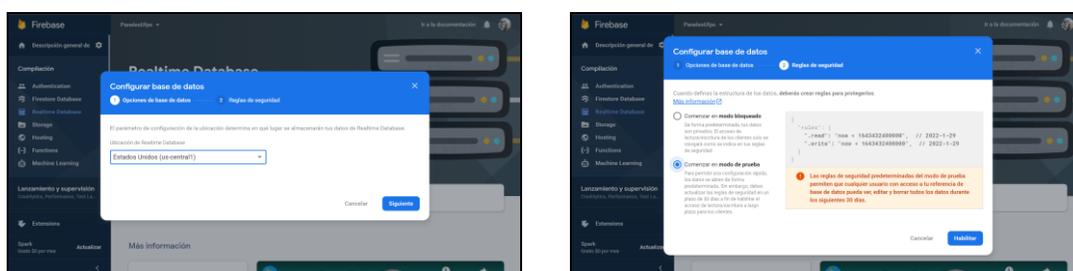


Figura 32. Creación de espacio físico y protocolo de seguridad de Realtime DataBase

Al culminar la configuración de la base de datos, se generaron las credenciales SDK, que fueron necesarias para la conexión desde la Raspberry Pi. Estas credenciales son propias y privadas de cada proyecto, y cada dispositivo. Los programas con las credenciales tienen acceso a la base de datos (ver figura 33).

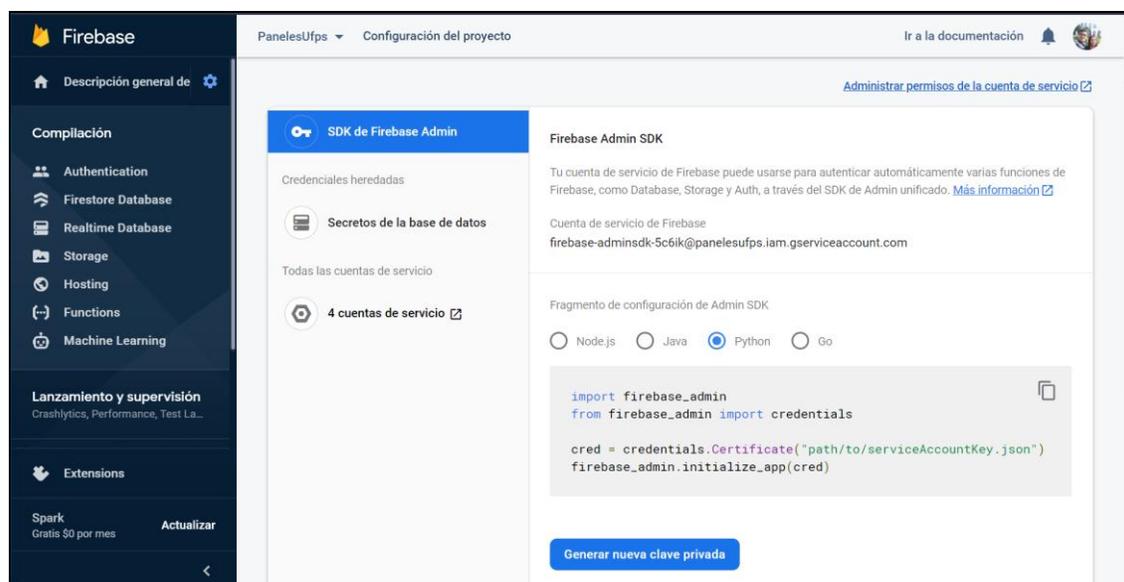


Figura 33. Generación de SDK para la administración de la base de datos

Un punto importante, es recordar almacenar el archivo “.json” con las credenciales SDK, este archivo solo puede ser generado una vez, y no se volverá a tener acceso desde la consola cuando se descargue. Este archivo debe ser almacenado en una ubicación accesible para el programa Python, ejecutado en la Raspberry Pi (ver figura 34).

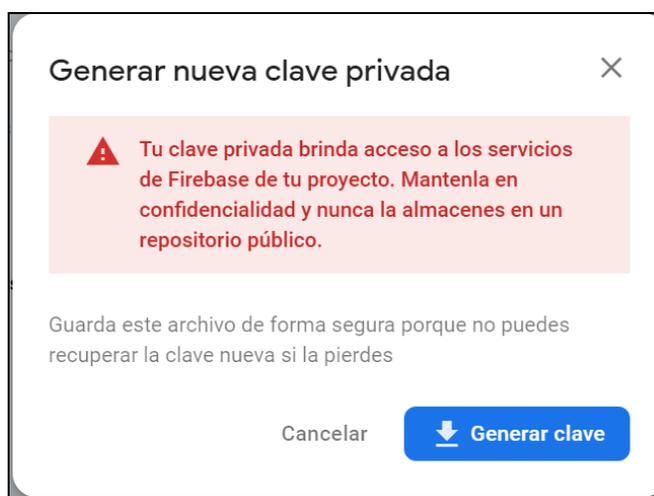


Figura 34. Advertencia de descarga de las credenciales

Al tener configurada la plataforma de Backend, se almacenó las credenciales en una ubicación accesible de la Raspberry Pi, para el funcionamiento del ejecutable Python.

4.2.3.1 Conexión de Raspberry pi 3B con Firebase. Durante el desarrollo del ejecutable Python para la conexión a Realtime DataBase, se descubrió que, para la gestión de los servicios de Firebase en la Raspberry Pi, es necesario el uso de una librería que se ejecuta, solo sobre la versión Python 3.6 o superior, lo que obliga a desinstalar la versión por defecto de Python 2.7 en la Raspberry Pi, por medio del comando apt-get remove, e instalar las dependencias de la nueva versión (ver figura 35).

```
$ sudo apt-get remove python2.7 //Remueve la versión 2.7
```

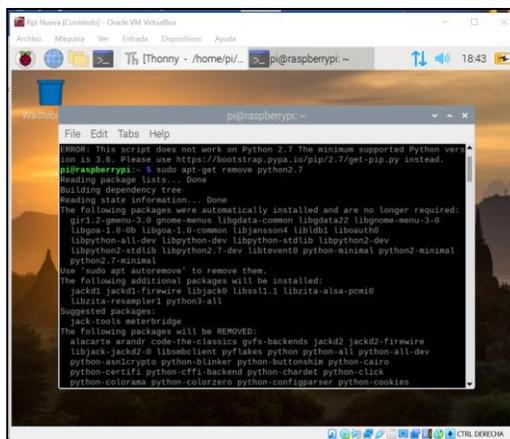


Figura 35. Desinstalación de la versión 2.7 de Python en la Raspberry Pi

Para saber que versión de Python está instalada en la Raspberry Pi, se puede usar los comandos `python3 --version` o `python3 -V` y, junto a la versión de Python instaladas, se debe actualizar el gestor de paquete de instalación `python3-pip` de la misma versión (ver figura 36).

`$ sudo python3 get-pip.py //Repositorio de librerías Python`

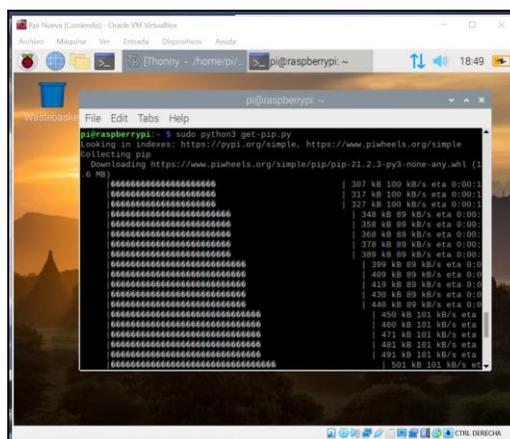
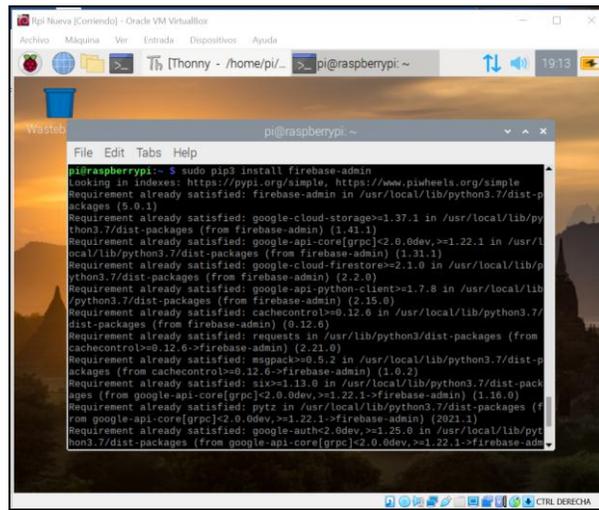


Figura 36. Instalación del gestor de paquetes de instalación python3-pip de Raspberry Pi

Con el gestor de paquetes de instalación de Python3, ya se tiene acceso al repositorio de la librería de firebase-admin. Por lo que se instaló (figura 37), y se procedió al desarrollo del ejecutable Python.

\$ sudo pip3 install firebase-admin //librería Firebase



```

pi@raspberrypi:~$ sudo pip3 install firebase-admin
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: firebase-admin in /usr/local/lib/python3.7/dist-packages (5.0.1)
Requirement already satisfied: google-cloud-storage>=1.37.1 in /usr/local/lib/python3.7/dist-packages (from firebase-admin) (1.41.1)
Requirement already satisfied: google-api-core[grpc]<2.0.0dev, >=1.22.1 in /usr/local/lib/python3.7/dist-packages (from firebase-admin) (1.31.1)
Requirement already satisfied: google-cloud-firestore>=2.1.0 in /usr/local/lib/python3.7/dist-packages (from firebase-admin) (2.2.0)
Requirement already satisfied: google-api-python-client>=1.7.8 in /usr/local/lib/python3.7/dist-packages (from firebase-admin) (2.15.0)
Requirement already satisfied: cachecontrol>=0.12.6 in /usr/local/lib/python3.7/dist-packages (from firebase-admin) (0.12.6)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from cachecontrol>=0.12.6->firebase-admin) (2.21.0)
Requirement already satisfied: msgpack>=0.5.2 in /usr/local/lib/python3.7/dist-packages (from cachecontrol>=0.12.6->firebase-admin) (1.0.2)
Requirement already satisfied: six>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from google-api-core[grpc]<2.0.0dev, >=1.22.1->firebase-admin) (1.16.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from google-api-core[grpc]<2.0.0dev, >=1.22.1->firebase-admin) (2021.1)
Requirement already satisfied: google-auth2.0dev, >=1.25.0 in /usr/local/lib/python3.7/dist-packages (from google-api-core[grpc]<2.0.0dev, >=1.22.1->firebase-admin) (2021.1)

```

Figura 37. Instalación de firebase-admin para acceso a Firebase desde la Raspberry Pi

En el desarrollo del ejecutable Python, se debieron tener en cuenta varios factores, que aportaban al desarrollo del proyecto; en los cuales están presentes, la conexión con la base de datos, la comunicación serial y el almacenamiento del histórico de datos.

Para la comunicación serial entre la Raspberry Pi 3B+, y el microcontrolador, es necesario la instalación de otra librería que permitiera la lectura serial de los dispositivos conectados.

\$ sudo apt-get install Python-serial //Comunicación Serial

Para el almacenamiento de histórico se usó las funciones: **open()** (“Nombre del archivo con la extensión”, “la operación de escribir”), **write()** (“Contenido del archivo”), y **close()**. Para el envío de los archivos se usó las funciones, **blob=**bucket.blob(“Nombre del Bucket del Storage de Firebase”), **outfile=**“Ubicación y nombre del archivo a subir” y, **blob.upload_from_filename(outfile)**. Funciones que dan acceso al ejecutable para las transferencias de archivos del almacenamiento local, al almacenamiento en la nube.

Todos los procedimientos se sintetizan en un ejecutable Python, que se inicia desde la Raspberry Pi, y que puede usar las credenciales para enviar la información y almacenarla (ver figura 38).

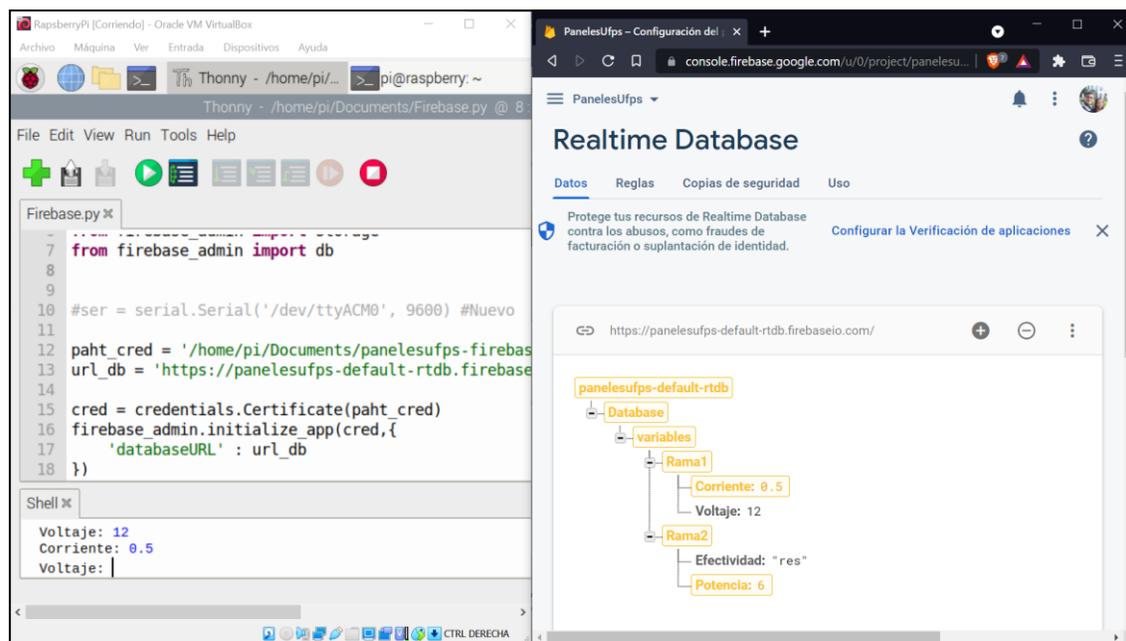


Figura 38. Conexión de Raspberry Pi con Firebase

4.3 Diseño de aplicación Web

Almacenar los datos desde la Raspberry Pi, no es suficiente para tener acceso a los mismos remotamente; por lo que fue obligatorio, diseñar una interfaz gráfica multiplataforma, que gestione el acceso a los datos desde cualquier dispositivo.

Inicialmente se consideró el desarrollo de una aplicación en Java, ya que, este lenguaje se caracteriza, por ser funcional en la mayoría de los dispositivos, pero consecuencia derivada de esto, es que la aplicación debía ser instalada y ejecutada sobre el dispositivo, y trae mayores consideraciones, como: las capacidades del dispositivo, la versión de Java que se usa, y una programación diferente para cada plataforma. Lo que no es conveniente en tiempo de desarrollo,

y recursos del proyecto.

Una mejor opción, la brindó el avance en tecnologías de la información, pues se basa en los navegadores del dispositivo, para la interacción de usuarios con aplicaciones Web; además permite en tiempo real tener acceso a los datos, y gestionar la seguridad por medio de usuarios autorizados. Por eso, para una interfaz multiplataforma, independiente de las capacidades del dispositivo, adaptativa en diferentes resoluciones, y que permita la gestión de usuarios, con acceso a los datos; se buscó como solución, diseñar una interfaz Web, o Frontend, que integrada con la gestión de la base de datos, o Backend de Firebase, permitiera la gestión de usuarios con acceso a la información, desde cualquier dispositivo conectado a Internet.

Esta solución permitió volcar el peso de la aplicación, en el motor del navegador y no en el mismo dispositivo, lo que es ajeno a las capacidades computacionales del mismo; y al diseñarse un Frontend Responsive, la interfaz sería adaptativa a cualquier tamaño de pantalla, lo que permite su correcta visualización en dispositivos móviles o de escritorio. Siendo superior esta solución a una aplicación nativa del equipo desarrollada en Java.

El desarrollo del Frontend de una aplicación Web, es sustentada bajo 3 lenguajes interactivos entre sí, HTML, CSS y JavaScript; donde cada uno ofrece un pilar de funcionamiento diferente en el navegador, y que, al integrarse generan una experiencia de usuario con las mismas capacidades de una aplicación nativa.

Para el desarrollo de estos lenguajes solo es necesario un editor de texto sencillo, pero para tener acceso a herramientas que facilitan la escritura de código, corrección de errores, compilación y ejecución del Script, se ofrecen en el mercado varios editores de texto de código fuente, que permiten un desarrollo más interactivo.

Para el desarrollo del Frontend fue seleccionado el editor de código fuente Open Source Brackets, el cual además de resaltar el código, marcar errores, tiene herramientas para autocompletado y ofrece vista previa en el navegador, permitiendo visualizar en tiempo real los cambios.

4.3.1 Diseño de HTML. La estructura sobre la que se sustenta el Frontend es el código HTML, ya que fue el primero desarrollado para Web. Es el encargado de contener la información dentro de etiquetas, además de llamar los otros archivos al navegador.

Para el desarrollo de la interfaz, se usó HTML5, y su función es la de contener la información, imágenes y enlaces, dentro de etiquetas (ver figura 39).

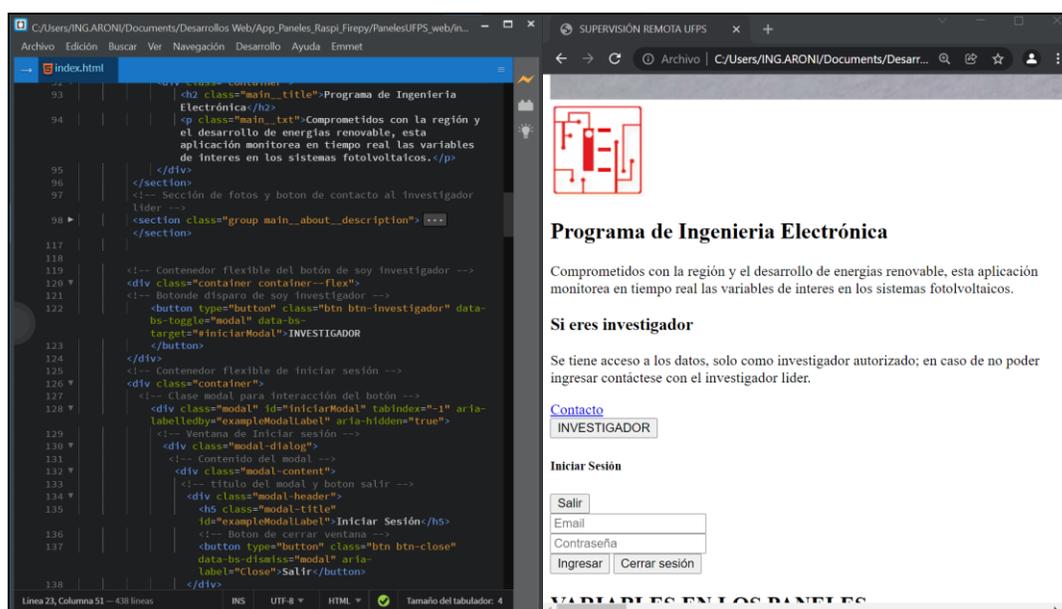


Figura 39. Inicio de la interfaz de la aplicación en HTML

Contener la información en etiquetas, permite darles atributos, o dinamismos, con los demás lenguajes, por lo que la secuencia de programación se concentra en agrupar la información, las imágenes y los enlaces, en las etiquetas de interés para la interfaz; desarrollando con ello, un

documento HTML independiente para cada ventana que necesitó la interfaz.

Para este proyecto, se desarrolló una ventana de inicio, donde ingresan los usuarios para tener accesos a los paneles, una ventana de nosotros con la información de contacto de los involucrados en el desarrollo, una ventana de galería, donde se visualizan imágenes de interés, una ventana de tecnología, donde se aprecia las plataformas usadas, una ventana de contacto, donde se tiene la información de la institución, y una ventana específica para cada panel, donde se tiene acceso a los datos, y los históricos de cada uno (ver figura 40).

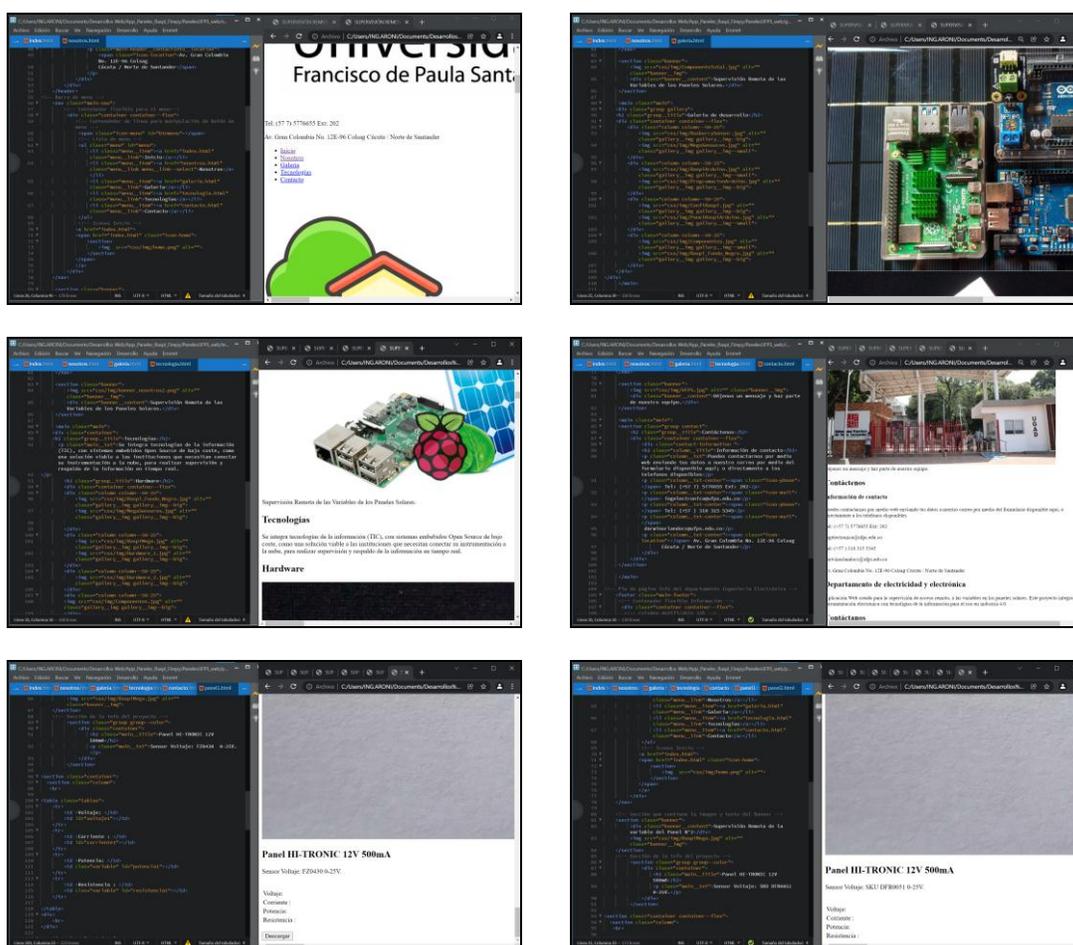


Figura 40. Ventanas de interfaz gráfica de la App en HTML

La estructura general del HTML, contiene dos importantes bloques, la cabeza (HEAD), y el cuerpo (BODY), para la interpretación de los caracteres del alfabeto español, se configuró el HEAD, con la etiqueta `<meta charset="UTF-8">`. Para identificar las pestañas en el navegador, se usó el icono con la etiqueta `<link rel="shortcut icon" href="ubicación de la imagen .ico">`, y para configurar luego con CSS3, la adaptabilidad a diferentes resoluciones se configura la etiqueta `<meta name="viewport" content="width=device-width, user-scalable=yes, initial-scale=1.0, maximum-scale=3.0, minimum-scale=1.0">`.

4.3.2 Diseño de CSS. Cuando evoluciono la Web, dejo de ser útil, solo la organización de información e imágenes; por lo que fue necesario, un lenguaje que le diera estilos al HTML, y con ello poder identificar la información relevante, de la información general.

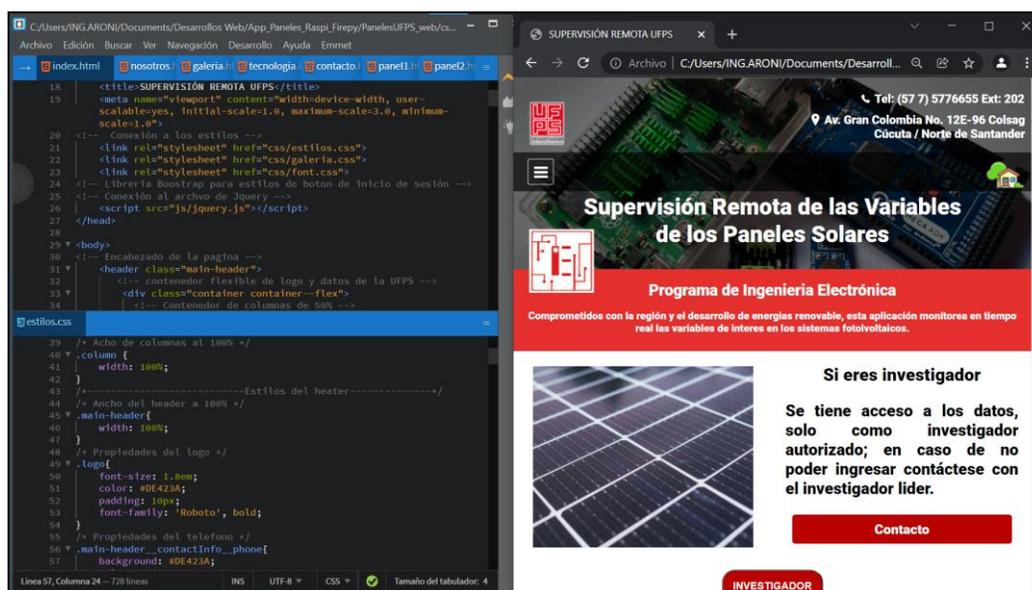


Figura 41. Inicio de la interfaz gráfica de la App con CSS

Para esta etapa de programación, se necesitó conocimientos en diseño gráfico y armonía del color (Mouser Electronics Colombia, 2021). Se usó la paleta de colores HTML, para encontrar el código hexadecimal del color representativo de la institución Universidad Francisco de Paula

Santander (2010), y con ello, tomar una combinación neutra como referencia, equilibrando visualmente la interfaz, sin sobrecargar las sensaciones del usuario, para una interacción cómoda con la aplicación (ver figura 41).

Dentro del documento HTML, puede ser elaborado todo el código necesario para una interfaz gráfica Web, pero se considera una buena práctica almacenar separadamente los archivos HTML, CSS, JavaScript, imágenes y demás; esto para asegurar, claridad y limpieza en el código, además de mantenimientos más rápidos, en caso de mejoras, o reconfiguraciones. También facilita al programador llevar un orden del desarrollo, y saber que parte del código debe abordar, en caso de fallo.

Aunque la estructura en etiquetas de HTML, y el formato en cascada de CSS, no permite modulación del código, ni creación de funciones, o herencias de objetos padres, como si sucede en lenguajes de programación como Java, la agrupación por clases, etiquetas o atributos id, permite manejar estilos generales para ciertos grupos en el documento HTML. Esta práctica es la que se usó para darle estilos a los banners, botones, columnas, títulos, footer, e incluso cajas de texto, inputs, tablas o enlaces que pertenezcan a la clase, etiqueta, o id seleccionados, sin necesidad de cambiar componente por componente.

Para la adaptabilidad de la aplicación en diferentes resoluciones, se debió configurar la etiqueta necesaria en el HEAD del documento HTML; y se procedió a escalar los tamaños ajustables necesarias para cada formato (ver figura 42).

```

334 }
335 }
336 /*-----Estilos del responsive-----*/
337 /* Tamaños de mínima pantalla */
338 @media screen and (min-width: 100px) { ... }
339 /* Propiedades del dinamismo de las imágenes dinámicas */
340 @keyframes publicidad_img { ... }
341 /* Tamaños de smartphone 7" */
342 @media screen and (min-width: 480px) { ... }
343 /* Pantalla tables */
344 @media screen and (min-width: 768px) { ... }
345 /* Pantallas Laptos */
346 @media screen and (min-width: 1024px) { ... }
347 /* Monitores de mesa */
348 @media screen and (min-width: 1180px) { ... }
349 /* Televisores */
350 @media screen and (min-width: 1600px) { ... }
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374

```

Figura 42. Formatos Responsive de la interfaz gráfica de la aplicación con CSS.

Las resoluciones seleccionadas, se probaron en diferentes pantallas para confirmar que se ajustaron correctamente, se tomaron referencias de las pantallas de Laptop, Tablet, y smartphone respectivamente (ver figura 43).



Figura 43. Interfaz en diferentes pantallas de diferentes dispositivos

Se programó una barra de menú desplegable en los formatos de pantalla más pequeños, y el que se debió programar para una interacción con el usuario con otro lenguaje de programación que le diera dinamismo e interacciones a la interfaz.

Durante el desarrollo de esta etapa, se debió evaluar qué información termina siendo importante y qué información se debe ajustar; pues en diferentes formatos la cantidad de texto puede alterar la integridad de la interfaz. También se debió evaluar qué tamaños son los correctos para las imágenes (ver figura 44).

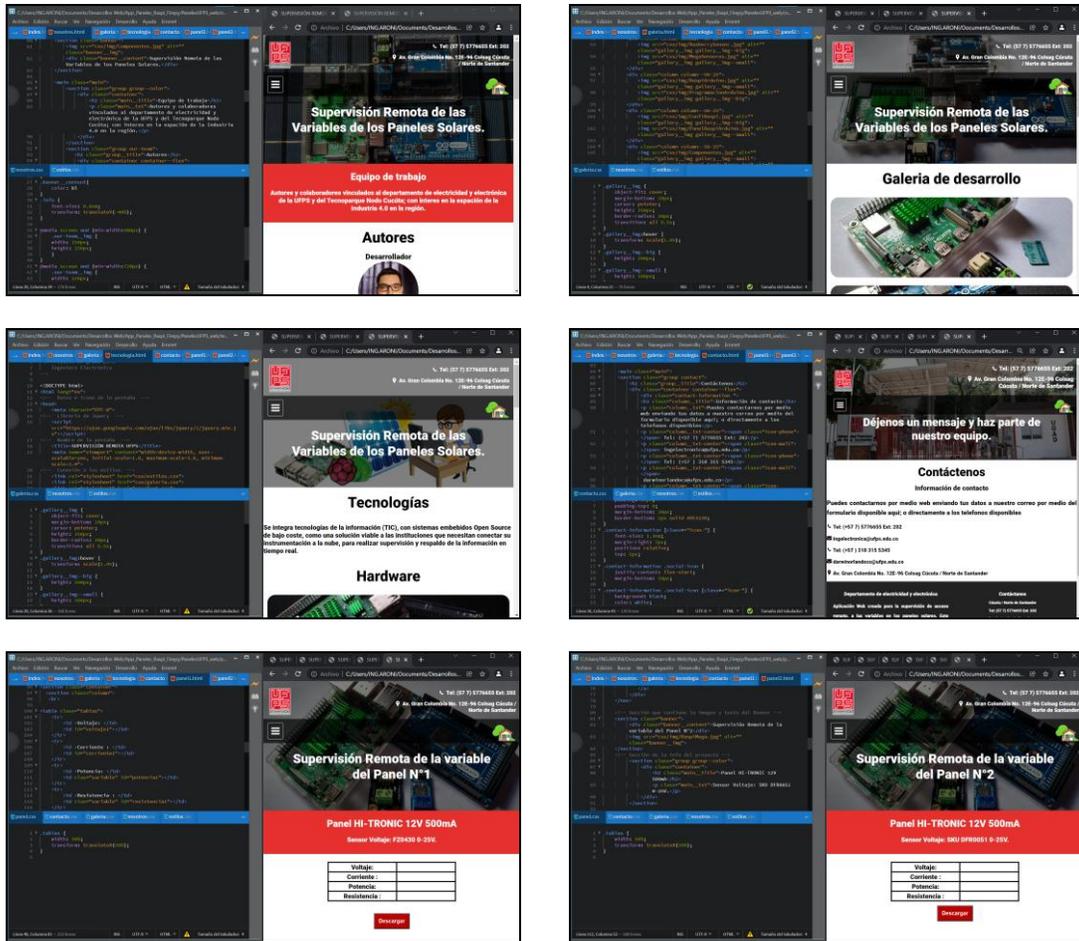


Figura 44. Ventanas de interfaz gráfica con CSS

La agrupación de las etiquetas por clases, facilito reutilizar el diseño original, en las demás clases con el mismo nombre, lo que redujo el tiempo de desarrollo, para las demás ventanas, y que enfoco los esfuerzos de diseño en la ventana principal, para replicar los resultados en las otras.

4.3.3 Diseño de JavaScript. Los documentos HTML y CSS, son suficientes para visualizar y organizar la información, de una forma agradable y atractiva para el usuario, pero no son suficientes, para mejorar la experiencia de usuario; por lo que se desarrolló un lenguaje del lado del cliente, que permitiera realizar peticiones interactivas, y le diera dinamismo a las páginas

Web; transformando las sencillas paginas estáticas que mostraban información, en aplicaciones capaces de cambiar según las acciones del usuario sobre ella, y así tengan acceso a servidores remotos, para acceder a los datos, y visualizarlos del lado del cliente.

JavaScript es un lenguaje de programación donde se pueden crear objetos, de los llamados selectores, que pueden ser etiquetas, clases o id declaradas en el HTML; se le dio un nombre id, a los botones, cajas de texto, y tablas, para con ellos poder interactuar, y realizar las acciones según las respuestas del cliente. Se debe resaltar que el id, es el selector de mayor uso, y que solo se permite un id por elemento; en caso de tener dos id similares, el Script leerá el primero, y ejecutará la acción sobre este, descartando todos los siguientes id con el mismo nombre. Se considera un error de programación usar el mismo id para dos elementos.

Sin los ejecutables JS (JavaScript), no se podría decir que se tiene una aplicación Web, y recibiría el nombre de página Web estática o dinámica según la interacción con el usuario, porque son las aplicaciones, las que permiten ejecutar acciones del lado del cliente, interactuando con el servidor.

Cabe destacar que el único lenguaje del lado del cliente interpretado por el usuario es JS, y se han desarrollado Frameworks, como JQuery, Angular o React, (marcos de trabajo), que realizan acciones bastante elaboradas con líneas menores de código. Estas acciones pueden ser desarrolladas solo JS, pero es necesario conocer a mayor profundidad la programación de este lenguaje.

Conociendo la programación en JS, se diseñó la interacción del formulario de ingreso, donde se solicitó usuario y contraseña; también se programó las acciones del evento click sobre, los botones e imágenes (ver figura 45).

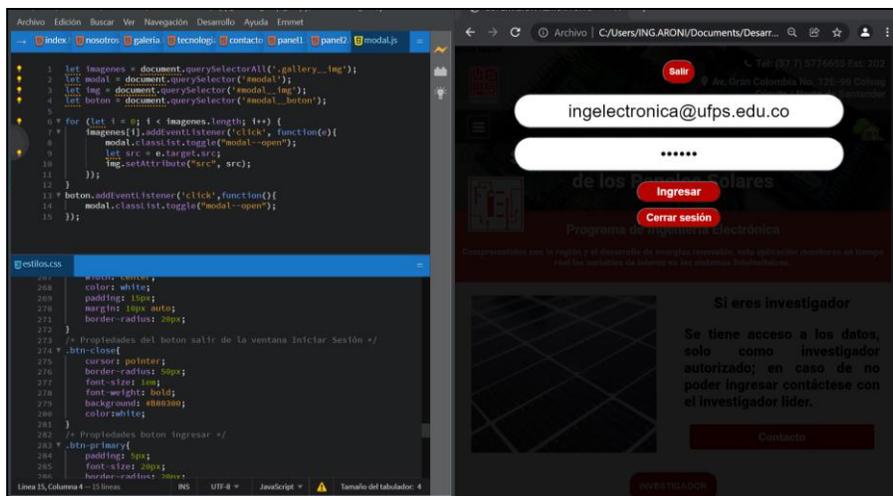


Figura 45. Formulario de inicio de sesión del lado del cliente, con JavaScript

Aunque se visualizan, botones, cajas de texto y tablas en el documento HTML, que interactúan con el cliente, estos elementos solo fueron completamente funcionales, cuando se configuraron las conexiones con la plataforma de Backend.

4.3.4 Conexión del Frontend con el Backend. Al culminar el desarrollo de la interfaz gráfica, o Frontend de la aplicación, se procedió a realizar la conexión con la base de datos del Backend. Se destaca que Firebase, fue seleccionada como plataforma de Backend, por ofrecer por medio de claves SDK, los servicios de autenticación, Realtime Database, almacenamiento y Hosting.

La conexión de la interfaz se realizó por una clave que es proporcionada por la misma plataforma de Firebase en su consola de administración.

Esta clave SDK es propia de cada proyecto, y es la que permite usar los servicios de Backend de Firebase, desde el HTML (ver figura 46). Firebase posee amplia información sobre sus servicios, y la forma de cómo utilizarlos, siempre es conveniente leer la información al respecto en la documentación de la consola, se recuerda que la clave es propia de cada proyecto, y para

efectos de libro fueron removidos.

The image shows a code editor on the left and a browser console on the right. The code editor displays the following JavaScript code for Firebase configuration and data retrieval:

```

365 <script>
366 // Your web app's Firebase configuration
367 const firebaseConfig = {
368   apiKey: "-----",
369   authDomain: "panalesufps.firebaseio.com",
370   databaseURL: "https://panalesufps-default-rtdb.firebaseio.com",
371   projectId: "panalesufps",
372   storageBucket: "panalesufps.appspot.com",
373   messagingSenderId: "-----",
374   appId: "-----"
375 };
376 // Initialize Firebase
377 firebase.initializeApp(firebaseConfig);
378
379 const auth = Firebase.auth();
380 var fs = firebase.database();
381
382 var userDataRef =
383   firebase.database().ref('Prueba/Base/Ram1').orderByKey();
384 //Prueba/Base/Ram1 toma los datos solo de eso
385   userDataRef.once("value").then(function(snapshot) {
386     var datos1 = snapshot.val().Corriente1;
387     var datos2 = snapshot.val().Voltaje1;
388     var datos3 = snapshot.val().Potencia1;
389     var datos4 = snapshot.val().Resistencia1;
390     var datos5 = snapshot.val().Corriente2;
391     var datos6 = snapshot.val().Voltaje2;
392     var datos7 = snapshot.val().Potencia2;
393     var datos8 = snapshot.val().Resistencia2;
394     var datos9 = snapshot.val().Corriente3;
395     var datos10 = snapshot.val().Voltaje3;
396     var datos11 = snapshot.val().Potencia3;
397     var datos12 = snapshot.val().Resistencia3;
398     var datos13 = snapshot.val().Corriente4;
399     var datos14 = snapshot.val().Voltaje4;
400     var datos15 = snapshot.val().Potencia4;
401     var datos16 = snapshot.val().Resistencia4;
402   });
403

```

The browser console on the right shows the Firebase configuration page with instructions to copy and paste the configuration into the body of the HTML document. The configuration code is repeated in the console:

```

<script type="module">
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebase
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libra

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: " ",
  authDomain: "panalesufps.firebaseio.com",
  databaseURL: "https://panalesufps-default-rtdb.firebaseio.c
  projectId: "panalesufps",
  storageBucket: "panalesufps.appspot.com",
  messagingSenderId: " ",
  appId: " "
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
</script>

```

Figura 46. Conexión con el servidor de Firebase

Al tener acceso a los servicios de Firebase, se activa los paquetes necesarios, según las necesidades del proyecto, vinculando las dependencias al código desarrollado en JS.

Con la conexión en el documento HTML, se programó la lectura y visualización de los datos en la interfaz; y se comprobó, la lectura de la base de datos desde la aplicación, lo que concluyo que la selección de las plataformas, fue adecuada para el diseño de supervisión remota, facilitando herramientas en la nube, que no se tenía con el uso de instrumentación local (ver figura 47).

The image shows two browser windows. The left window displays the Firebase Realtime Database console for 'panalesufps-default-rtdb'. The database structure is as follows:

```

panalesufps-default-rtdb
├── Database
│   └── Base
│       └── Rama1
│           ├── Corriente1: 0.1
│           ├── Corriente2: 0.17
│           ├── Corriente3: 0.024
│           ├── Corriente4: 0.058
│           ├── Potencia1: 1.5
│           ├── Potencia2: 2.89
│           ├── Potencia3: 0.288
│           ├── Potencia4: 0.841
│           ├── Resistencia1: 150
│           └── Resistencia2: 100
└── Ubicación de la base de datos: Estados Unidos (us-central1)
  
```

The right window shows the application's user interface. It features a red header with the text 'Programa de Ingeniería Electrónica' and a sub-header 'Supervisión Remota UFPS'. Below this is a section for investigators, followed by a table of variables for four panels:

PANEL 1		PANEL 2		PANEL 3		PANEL 4	
Voltaje:	15	Voltaje:	17	Voltaje:	12	Voltaje:	14.5
Corriente:	0.1	Corriente:	0.17	Corriente:	0.024	Corriente:	0.058
Potencia:	1.5	Potencia:	2.89	Potencia:	0.288	Potencia:	0.841
Resistencia:	150	Resistencia:	100	Resistencia:	500	Resistencia:	250

Figura 47. Visualización desde la interfaz de los datos de Realtime DataBase

Con el acceso a los datos desde la aplicación, se procedió a darle seguridad y restricción a los mismos, por lo que se crearon usuarios autorizados, utilizando los servicios de la misma plataforma, y aprovechando el formulario de ingreso desarrollado.

Para los servicios de autenticación, se debe activar el servicio en la consola de administración, y configurar los correos de usuarios que se desea, tengan acceso. La misma configuración le permite, seleccionar los métodos que se desea usar por los usuarios autorizados (ver figura 48).

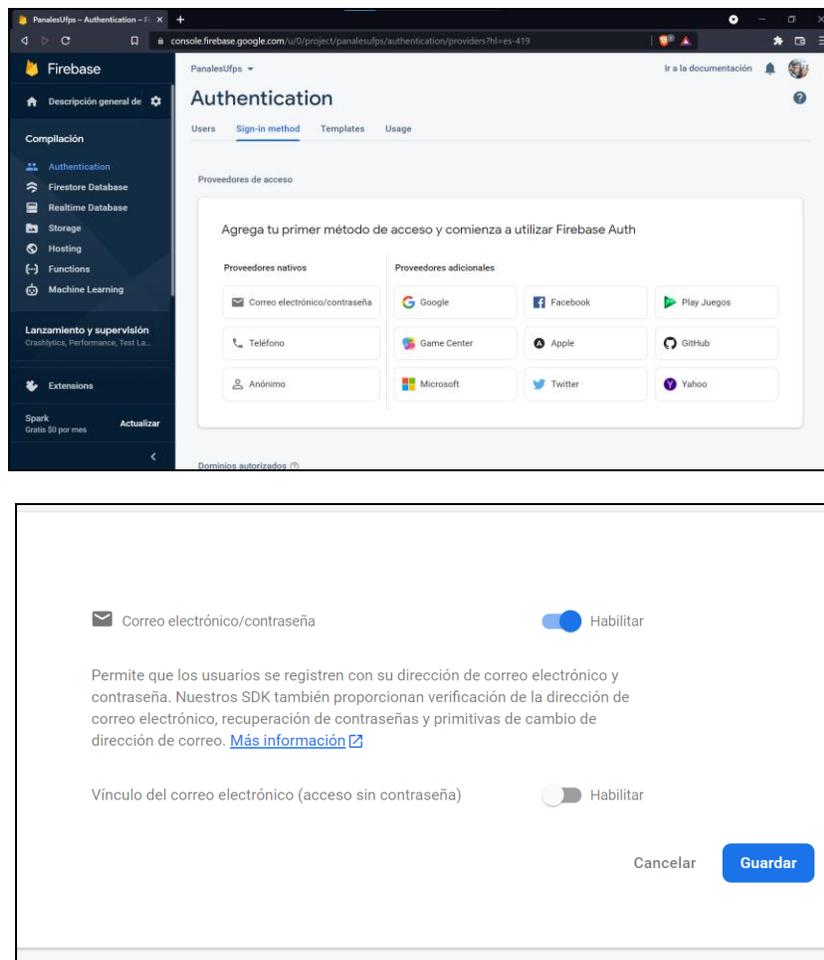


Figura 48. Configuración de los servicios de autenticación de Firebase

En la misma consola de Firebase se puede gestionar los usuarios con acceso a la aplicación, agregando a los que se consideraron necesarios. Se seleccionó el método de correo como usuario, y una contraseña otorgada por el administrador del proyecto, para asegurar con esto el control total de los usuarios y claves (ver figura 49).

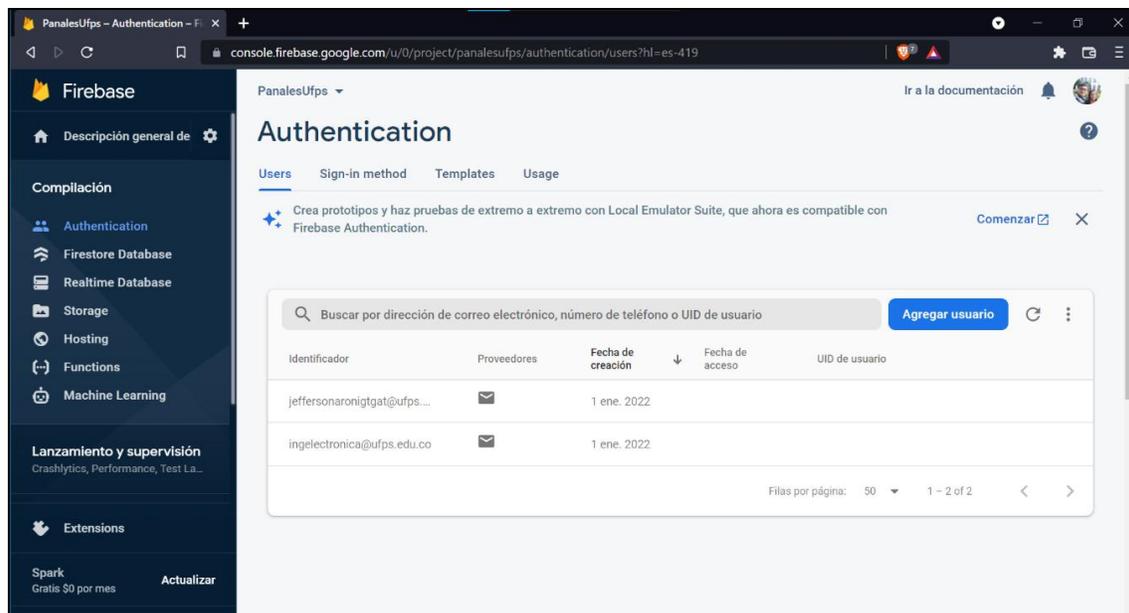


Figura 49. Gestión de usuarios con acceso a la aplicación

Culminado con la restricción de los datos, para dejar acceso a los mismos, solo a usuarios autorizados, se procedió a configurar el acceso a la descarga de los históricos, los cuales se almacenaron en el Storage de Firebase.

Para el histórico de datos, se generó una función que recorre todos los archivos almacenados, y muestra el nombre de los mismos; dejando al usuario la posibilidad de decidir cual desea descargar. Se dejó una caja de texto con un botón para confirmar la selección, y cuando se confirme la decisión, se visualiza el botón de descarga (ver figura 50).

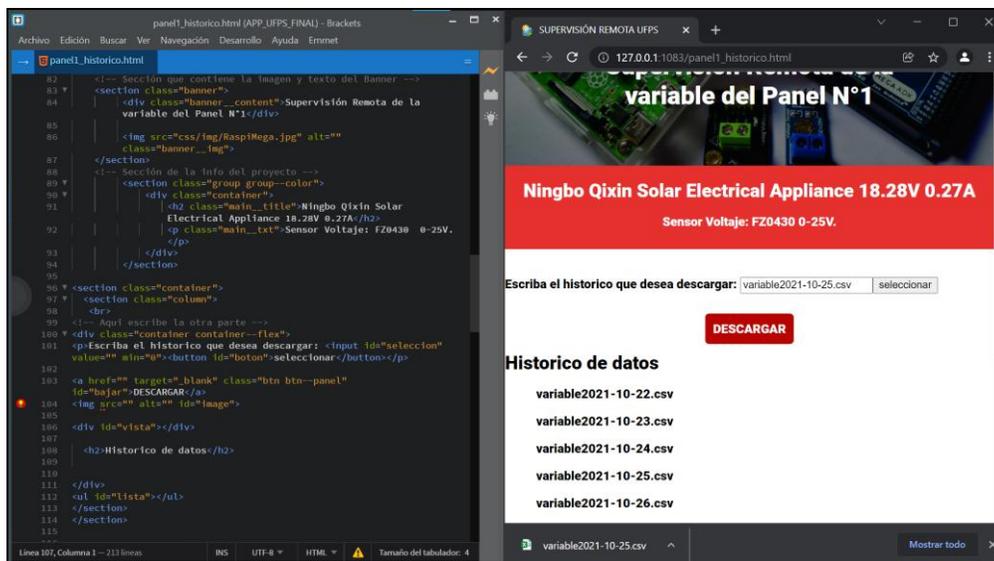


Figura 50. Descarga de histórico de datos

Con la visualización y descarga del histórico de datos, se demostró la efectividad del método de selección; además se verificó la integración de las plataformas de desarrollo seleccionadas, permitiendo llevar seguimiento, y cumplimiento de los objetivos, desde la toma de datos en los sensores, hasta la entrega de los datos e históricos a los usuarios.

En la culminación de esta etapa, se pudo comprobar el diseño de un prototipo de supervisión desde la nube, para las variables en los paneles solares, de bajo coste, y libre de pagos de licencias; lo que trajo mayores beneficios al proyecto.

4.4 Implementación del Prototipo

El desarrollo de un sistema de supervisión remota desde la nube, usando plataformas de acceso libre, tuvo como propósito general, el diseño de un prototipo que cumpliera con las características básicas de captación de datos del HOB0® UX120-006M; y que, realizará la entrega de datos, con acceso remoto, desde cualquier dispositivo conectado a Internet. Debido a la eficiencia del diseño realizado, se buscó desarrollar una implementación del prototipo que

integró todas las etapas de diseño, y que permitió, la captación de datos, la visualización y descarga de los históricos, remotamente, desde una aplicación desplegada en la nube.

4.4.1 Implementación de la instrumentación. Tomando las referencias de los sensores simulados, el microcontrolador para la captación de datos, el microprocesador como sistema de cómputo; se procedió a la búsqueda de los mismos en el mercado local, lo que permitió la adquisición de los sensores de voltaje FZ0430 y SKU DFR0051, sensor de corriente KS0185, microcontrolador Arduino Mega 2560, con su respaldo Arduino UNO R3 y el microprocesador Raspberry Pi 3B+, evaluados en la etapa de selección (ver figura 51).

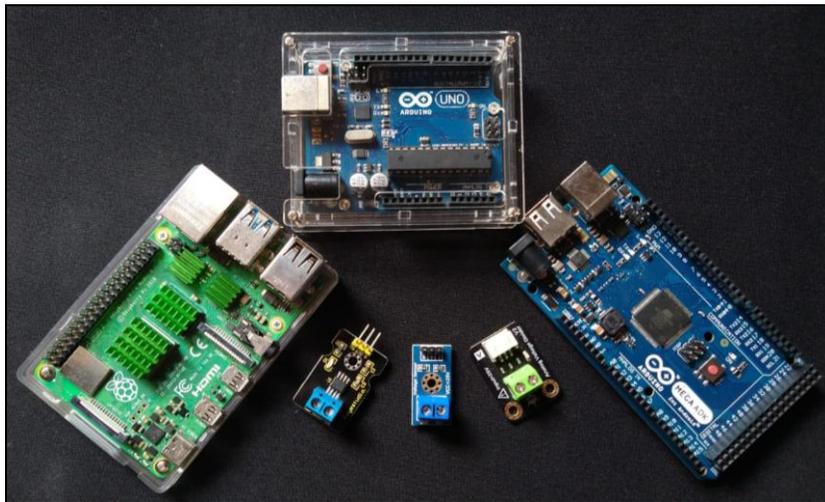


Figura 51. Componentes del prototipo

Confirmando la disponibilidad de los componentes del sistema de supervisión, se procedió a realizar la medición de las variables de un panel de intemperie, Ningbo Qixin Solar Electrical Appliance 18.36V - 0.27A de 5W de potencia; y donde se debió seleccionar una resistencia de potencia que permitió realizar las mediciones. Para la selección de la resistencia, se procedió a evaluar el mejor valor posible, para encontrar el equilibrio entre voltaje y corriente, y permitiera generar los valores más cercanos posible a los umbrales del panel.

$$Carga = \frac{V_{mp}}{I_{mp}}$$

$$Carga = \frac{18.36V}{0.27A} = 68\Omega$$

El valor de resistencia mínimo que se necesita, para que el panel en su máxima potencia generada, entregue los valores de corriente y voltaje máximos, es de 68Ω , pero por la naturaleza de dependencia con la radiación, para generar potencia; es sugerido un valor como mínimo del 25% superior al valor de resistencia encontrado, por lo que se utilizó un valor de 100Ω , que permitió una medición más intuitiva, y que cumplía con la descripción mencionada, además también se consideró una resistencia de 150Ω como opcional en caso de necesitarse medidas diferentes, o para alterar la resistencia de la carga, al necesitar medir corrientes mayores (ver figura 52).

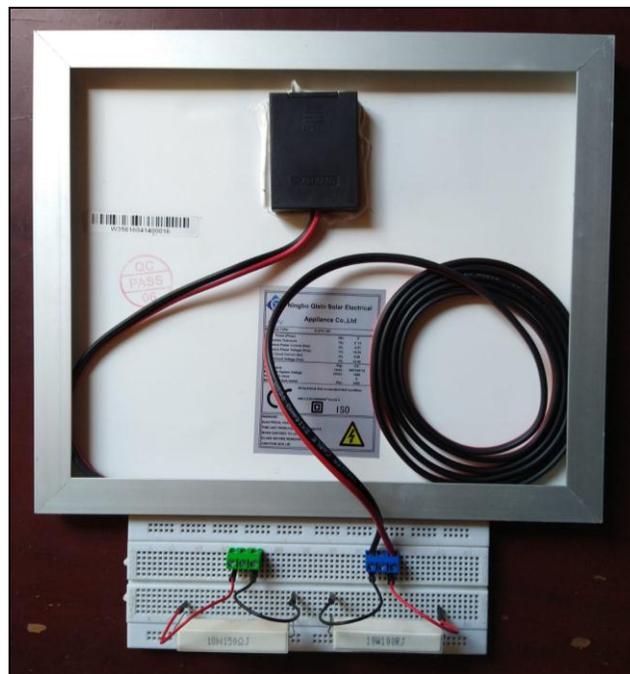


Figura 52. Panel fotovoltaico con resistencia en serie

Inicialmente se debió configurar la plataforma Arduino, para la lectura de los sensores, la organización de las lecturas, y el envío de la información por puerto serial. Se prefirió iniciar la programación en el Arduino UNO R3, por tener una Case protectora y permitir una mayor manipulación; pero debe resaltarse que los resultados son totalmente replicables en la tarjeta Arduino Mega 2560. Se debe resaltar que es conveniente el uso de la tarjeta Arduino UNO R3, mientras las mediciones analógicas, no superen un total de 6 canales de medición; recordando que el costo de adquisición de esta tarjeta es inferior a la del Arduino Mega 2560, y la programación es la misma, con unas ligeras variaciones de conexión.

Como se necesitó calibrar los sensores de voltaje, se midió el valor obtenido de un voltaje constante, generado en el pin digital 5, con un valor de salida de 4.36V que se interpreta por el sensor FZ0430, como 202; y el cual fue tomado como referencia para la sensibilidad, del sensor (ver figura 53).

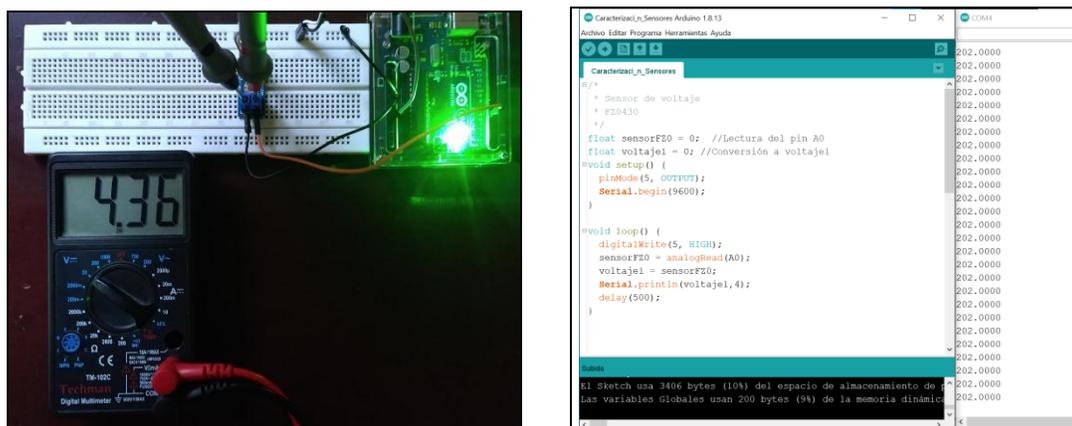


Figura 53. Medición para la calibración de sensor de voltaje FZ0430

La referencia se comparó con la medida mínima que fue 0V, y se interpreta como 0 en el Arduino, usándose la ecuación para encontrar el ajuste de la sensibilidad.

$$\text{sensibilidad} = \frac{4.36 - 0}{202 - 0} = 0.0215846$$

Al ajustarse la sensibilidad, se comparó otra medición constante, y se seleccionó el pin 3.3V, el cual tuvo una salida de 3.23V y se interpretó entre los valores de 3.2161V – 3.2377V del cual su promedio es 3.22685V; lo que se pudo aproximar a 3.23V. Siendo esta la medición interpretada por el multímetro (ver figura 54).



Figura 54. Comparación de calibración de sensor de voltaje FZ0430

La calibración del SKU DFR0051 se realizó de la misma forma, y se obtuvo un voltaje constante, del pin digital 5, con un valor de salida de 4.34V que se interpretó como 202, tomándose como referencia para el ajuste de la sensibilidad (ver figura 55).

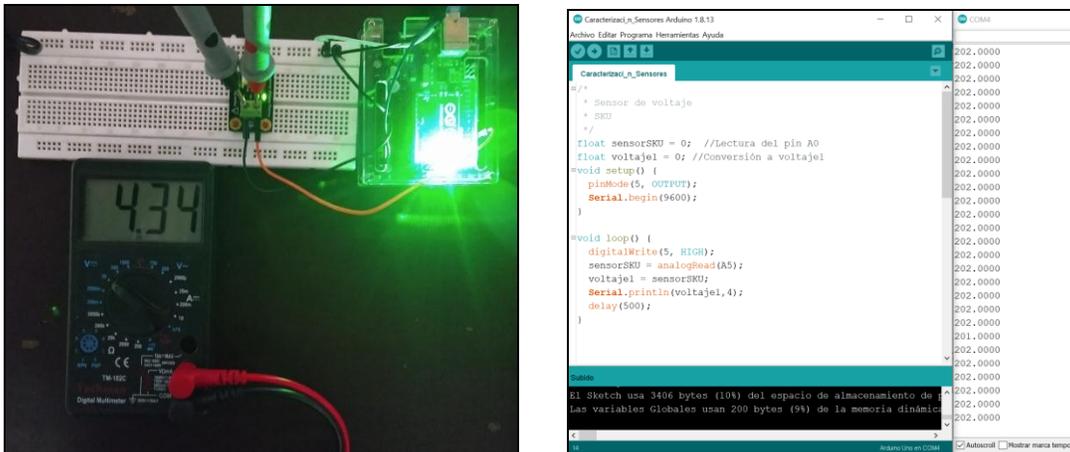


Figura 55. Medición para la calibración de sensor de voltaje SKU SDFR 0051

La referencia se comparó con el valor mínimo 0V, que se interpretó como 0 el Arduino, y usando la ecuación se ajustó la sensibilidad.

$$\text{sensibilidad} = \frac{4.34 - 0}{202 - 0} = 0.0214851$$

Al ajustarse la sensibilidad, se compara otra medición constante, y se seleccionó el pin 3.3V, el cual tuvo una salida de 3.23V y se interpretó con un valor de 3.2228V (ver figura 56).

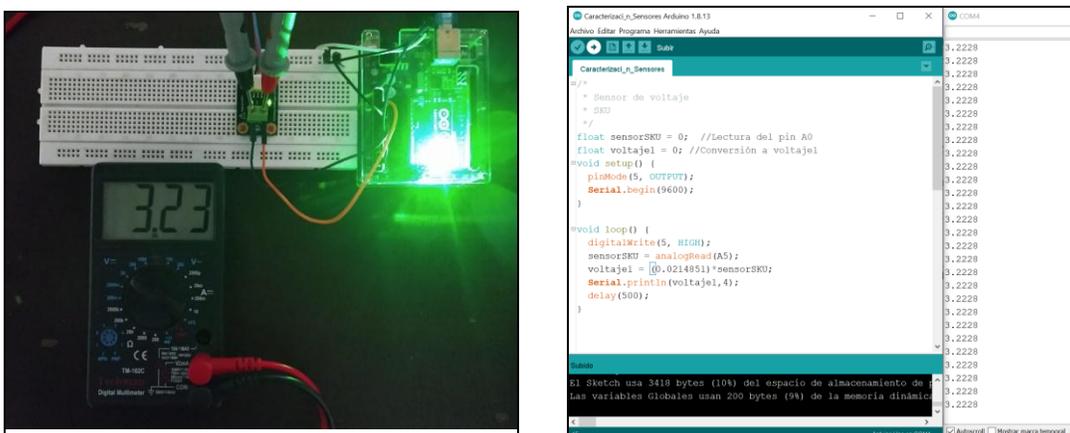


Figura 56. Comparación de calibración de sensor de voltaje SKU DFR0051

Terminado de calibrar los sensores, de voltaje, se determinó, que el sensor FZ0430 tiene mayor exactitud, y el SKU DFR0051, mayor precisión, pero la diferencia entre sus resultados no varía lo suficiente, como para considerarse diferentes. Por lo que no es viable usar el sensor SKU DFR0051, si el FZ0430 que tiene un precio 4 veces menor, y ofrece la misma calidad de medida.

Como se concluyó en la etapa de selección de sensores, el sensor de corriente SCT013 YHDC, no fue considerado para su utilización en el prototipo, y se justifica, porque su viabilidad es deficiente en cuanto aportes al proyecto; además se tiene el sensor KS0185, que cumple con todos los parámetros exigidos para el prototipo.

Se inicio las lecturas de corriente con el sensor KS0185, descubriéndose que era necesario una calibración, pues lo valores medidos no coincidían con los del multímetro; también fue necesario para aumentar la corriente, utilizar una resistencia menor que se obtuvo de dos resistencias en paralelo (ver figura 57).



Figura 57. Lectura de corriente con el sensor KS0185

En la lectura de corriente, los datos variaban demasiado para considerarse una medida estable, por lo que este sensor funciona mejor con un promedio de medición. Se programó un

ciclo para promediar las mediciones, y poder tener un valor estable que se pueda calibrar para la lectura de corriente. El ciclo se programó con 350 interacciones con retardo de 15 milisegundos entre ellas, lo que facilitó una lectura rápida, pero ajustada, para entregar los valores estables.

Para la calibración es necesario mínimo dos puntos de medición y la sensibilidad del sensor, el cual es ofrecido por el fabricante. Los puntos de calibración pueden ser los voltajes del sensor, los valores de 0 – 1024 interpretados por el ADC, o los valores de corriente.

En la medición se deben ajustar los puntos de medición del sensor a tierra, y en lo posible con la misma tierra del sensor, confirmando la lectura de cero voltios, sin interferencias de nodos flotantes (ver figura 58).

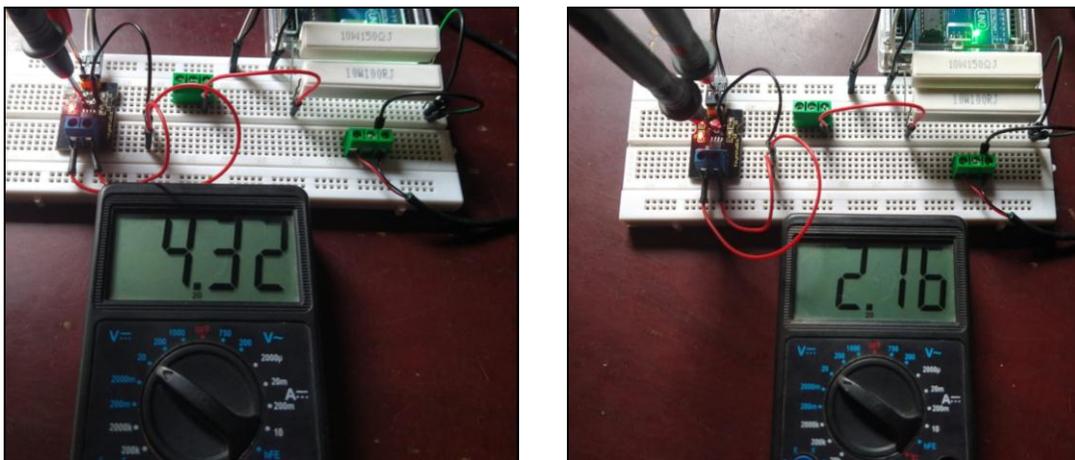


Figura 58. Lectura de voltajes de calibración con el sensor KS0185

En la calibración se descubrió que los mejores valores para los puntos de calibración son los voltajes en el sensor, donde se debió ajustar el valor máximo, representativo de 1024, en 4,315V y no en 5V como se había considerado idealmente, y el valor de 0A, que tiene un valor de 2,167V en la salida del sensor, y no de 2.5V, también considerado ideal. Se aclara que se mantenía una pequeña variación en las lecturas, por lo que se ajustó en el código de programación a valores

promedios que permitieran una lectura con mayor exactitud, a los valores de corriente (ver figura 59).

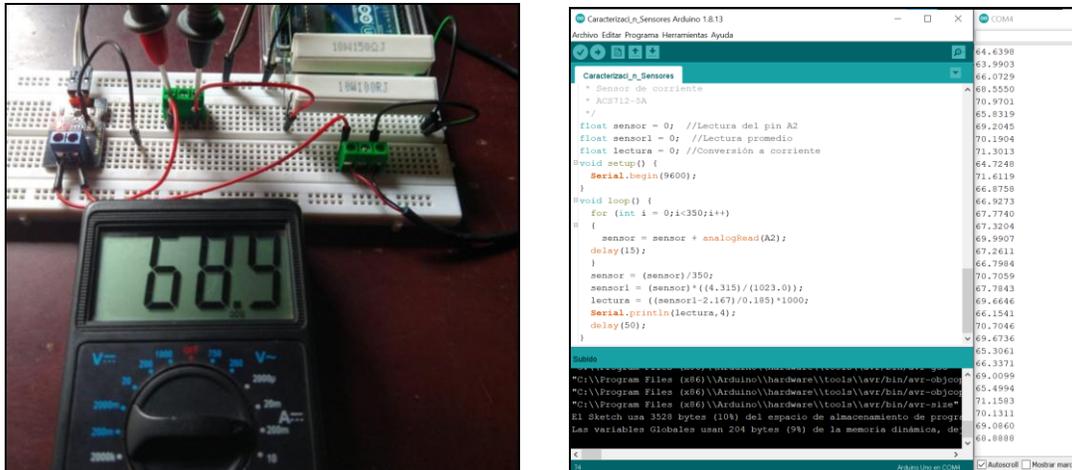


Figura 59. Lectura de corriente calibrada del sensor KS0185

Al comprobarse la lectura de corriente en el sensor, se procedió a implementar el sistema integrado de los sensores de instrumentación, y se utilizó un voltaje constante en la resistencia de carga, para compararlos con el medido, y realizar los ajustes de calibración necesarios, para el funcionamiento de los 3 sensores.

Los sensores de voltaje no necesitan ser alimentados, pero si referenciados a tierra, con el microcontrolador, y poder realizar las comparaciones de voltajes; el sensor de corriente si necesitó ser alimentado y se utilizó el pin de 5V del Arduino, como fuente de alimentación del sensor. Para la calibración del sensor de corriente KS0185 se debió considerar el voltaje de alimentación y calibrar los valores máximos, que dependen de la alimentación; luego se debió medir el voltaje de referencia de cero amperios, y ajustarlo para la interpretación de corriente de 0A.

En el código necesitó un formato de organización de los datos para fácil interpretación en la Raspberry Pi 3B+, y se seleccionó el carácter “/” como separador de las mediciones de los diferentes sensores, los cuales se organizaron en funciones con el nombre del sensor, y son llamadas dentro del void loop del Arduino, imprimiendo los valores del FZ0430, SKU DFR0051 y el KS0185, respectivamente. Se dejó el valor Vout del ACS, como auxiliar para la calibración (ver figura 60).

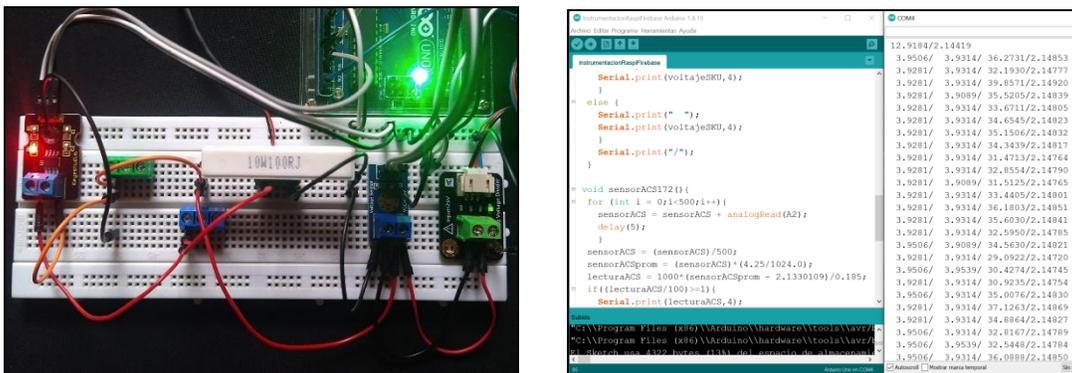


Figura 60. Integración y calibración de la instrumentación

Se realizó la verificación de los valores medidos de voltaje en el multímetro, con los interpretados por el microcontrolador, demostrándose la correcta calibración de los sensores de voltaje (ver figura 61).

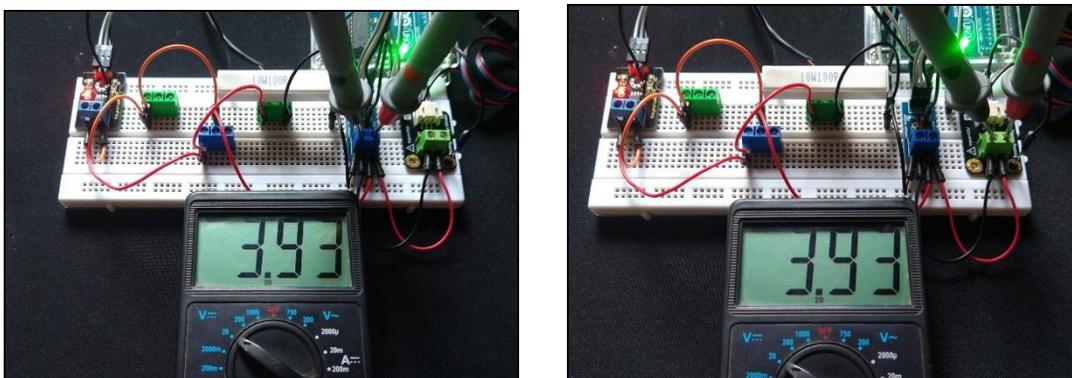


Figura 61. Comprobación de calibración de los sensores de voltaje

En la lectura de corriente, se debió considerar que el sensor no estuviera expuesto a variaciones de campos magnéticos, o variaciones de voltajes en la alimentación, porque esto traería errores en las medidas. Se utilizó la medición del multímetro para confirmar la medición de corriente, y se comprobó la correcta calibración del sensor (ver figura 62).

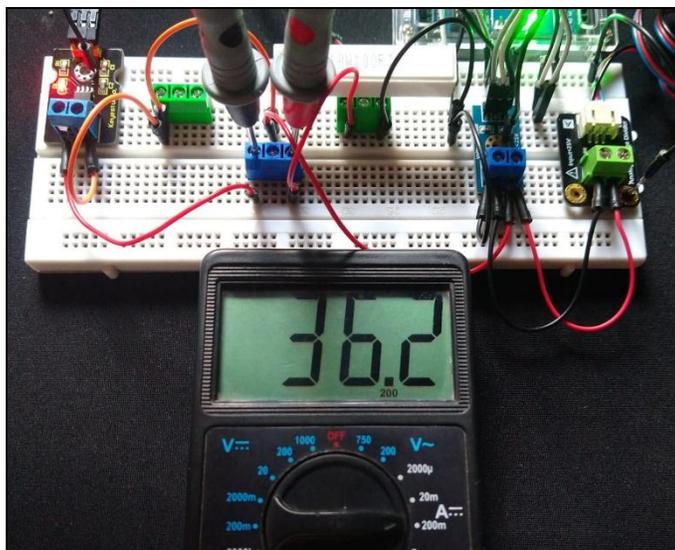


Figura 62. Comprobación de la calibración del sensor de corriente

Con el sistema de instrumentación calibrado, se procedió a la configuración de la Raspberry Pi 3B+, usando el mismo procedimiento que de la etapa de configuración de Raspberry Pi, actualizando la versión de Python, e instalando las librerías de comunicación serial, y de Firebase (ver figura 63).



Figura 63. Comprobación de la calibración del sensor de corriente

Con la Raspberry Pi 3B+ configurada, se procedió a desarrollar el ejecutable Python que comunica los datos del microcontrolador, a la base de datos en Firebase, y se utilizó la conexión inalámbrica de la interfaz de la Raspberry Pi 3B+, por VNC, para verificar la lectura y envío de datos, comparándolos y confirmando el almacenamiento, en la plataforma en la nube (ver figura 64).

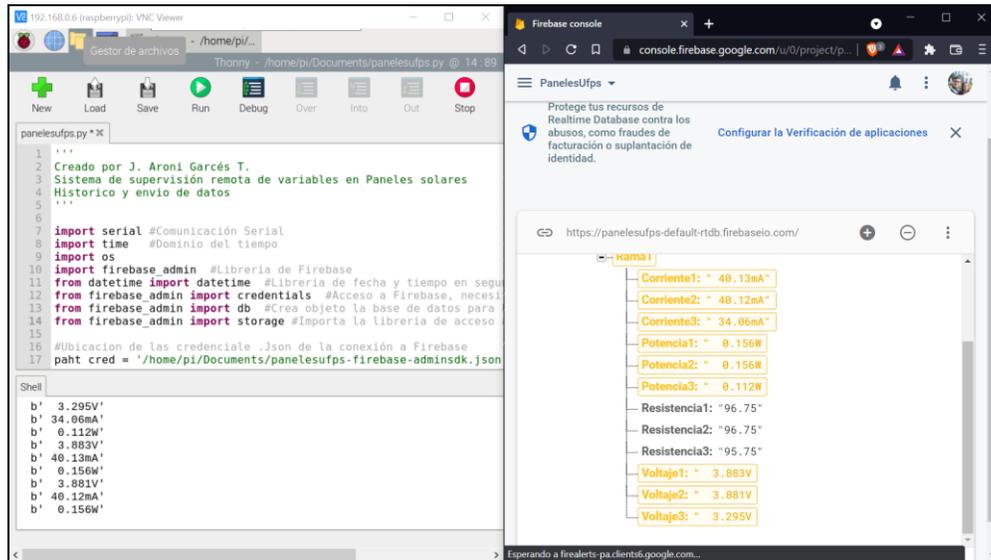


Figura 64. Conexión y envío de datos con Realtime DataBase desde la Raspberry Pi

Con la instrumentación calibrada, conectada y confirmando la gestión de datos desde la Raspberry Pi 3B+, se realizó la instalación del prototipo en un lugar abierto donde recibió la lectura directa del sol, y se pudo comprobar la lectura de los datos, por parte de la Raspberry Pi (ver figura 65).



Figura 65. Pruebas de funcionamiento en sitio del prototipo de supervisión remota

Al realizarse la lectura constante del panel, se permitió iniciar la generación de históricos donde se tuvo que considerar que formato era el más óptimo para la gestión de la información. Inicialmente se propuso el texto plano .txt, el cual solo permitía la visualización, por lo que se procedió a realizar el cambio por el formato .csv el cual es interpretado en Excel, y posee varias herramientas de gestión de datos, como la separación en columnas, que permitió separar los datos del tipo de variable (ver figura 66).

	A	B	C	D
1	Hora:			
2	11:57:39.42			
3	Voltaje: b'23.25V'			
4	Corriente: b'243.97mA'			
5	Hora:			
6	11:58:01.74			
7	Voltaje: b'23.14V'			
8	Corriente: b'243.85mA'			
9	Hora:			
10	11:59:07.23			
11	Voltaje: b'23.18V'			
12	Corriente: b'243.90mA'			
13	Hora:			
14	12:0:12.60			
15	Voltaje: b'23.20V'			
16	Corriente: b'243.92mA'			
17	Hora:			
18	12:01:17.78			
19	Voltaje: b'23.20V'			
20	Corriente: b'243.92mA'			
21	Hora:			
22	12:02:23.08			
23	Voltaje: b'23.18V'			
24	Corriente: b'243.90mA'			

Figura 66. Comparativa de formatos de almacenamiento de histórico de datos

Con la separación de los datos, se descubrió, que no era necesario declarar el tipo de variable medida, pues dificultaba tabular los datos individuales en cada casilla, y se organizó el almacenamiento de los datos secuenciales, con la separación del carácter “;”, para tabular la información con las herramientas de Excel (ver figura 67).

```
if cont==0:
    archivo.write("Hora;Voltaje FZ0430;Corriente KS0185; Voltaje SKUDFR \n")
    archivo.write(str(hora)+";"+str(numvolt1)+";"+str(corr1)+";"+str(volt2)+"\n")
```

Figura 67. Organización de las variables para tabulación en el archivo Excel

Con la información organizada, se inició el almacenamiento de las variables y la tabulación de los datos en el archivo Excel, donde se puede separar los datos numéricos, y realizar graficas de comportamiento de las medidas, pudiendo identificar la curva de generación del panel durante el día, las mediciones iniciaron el primer día del mes de noviembre del año 2021, donde se tuvo una radiación constante que generó una curva, que permite apreciar el comportamiento del panel durante 10 horas de luz, las horas que se tomaron son desde las 7:30 am hasta las 5:25 pm (ver figura 68).

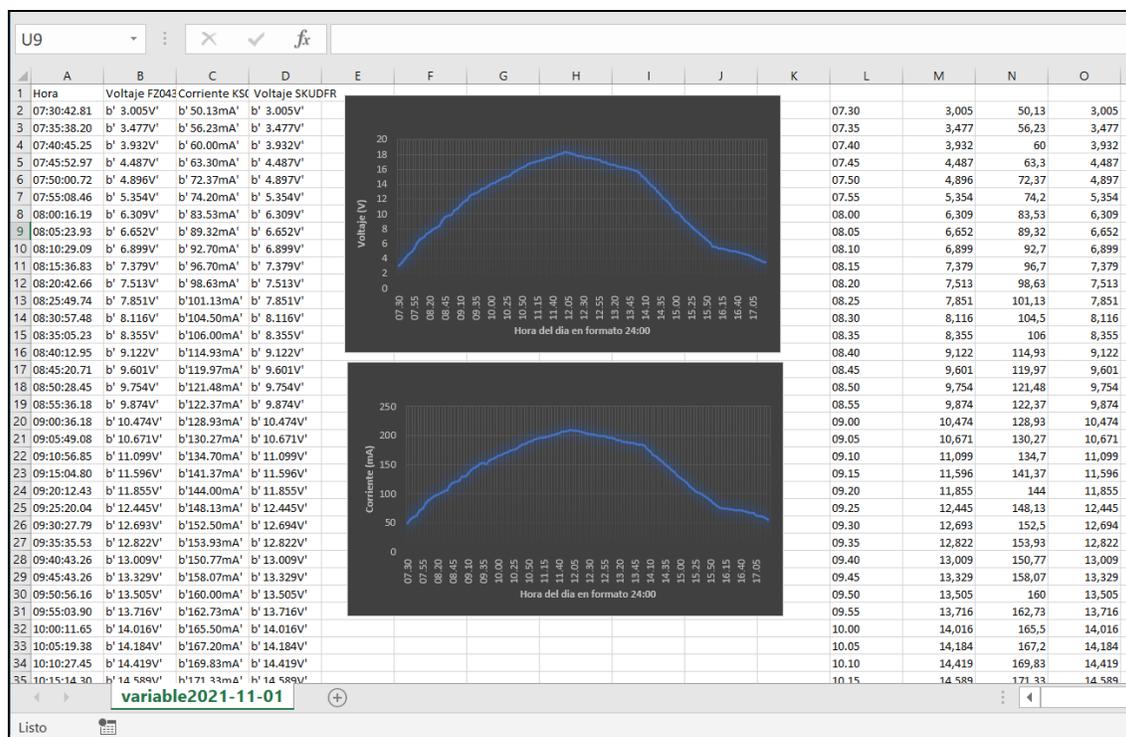


Figura 68. Tabulación de datos de un día de lectura

Con los datos almacenados en la plataforma en la nube, se procedió a desplegar la aplicación Web, en el Hosting, y con ello poder tener acceso desde cualquier dispositivo remotamente.

Al iniciar el servicio de Hosting en la plataforma de Firebase, se ofrece un paso a paso a seguir, que inicia con la instalación de las herramientas de servidor de Firebase, donde es obligatorio tener Nodejs instalado (ver figura 69).

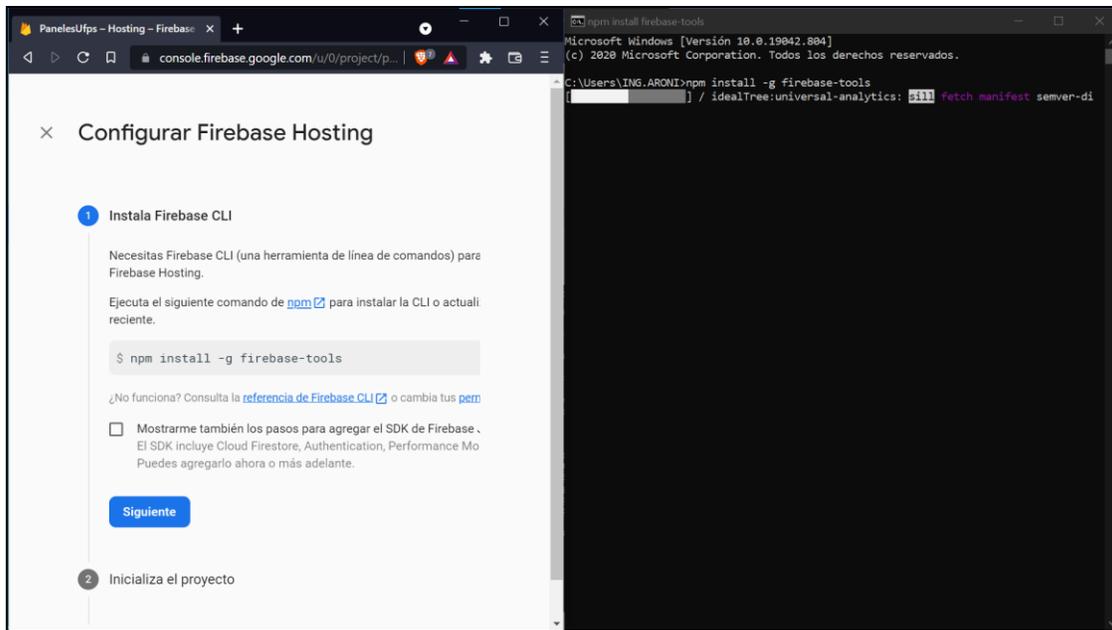


Figura 69. Instalación de herramientas de despliegue de Firebase

Para acceder a los servicios de Hosting, es necesario, permitir el acceso por parte del CLI de Firebase, a las dependencias conectadas a la cuenta de Gmail, en el gestor de despliegue en el CMD de Windows (ver figura 70).

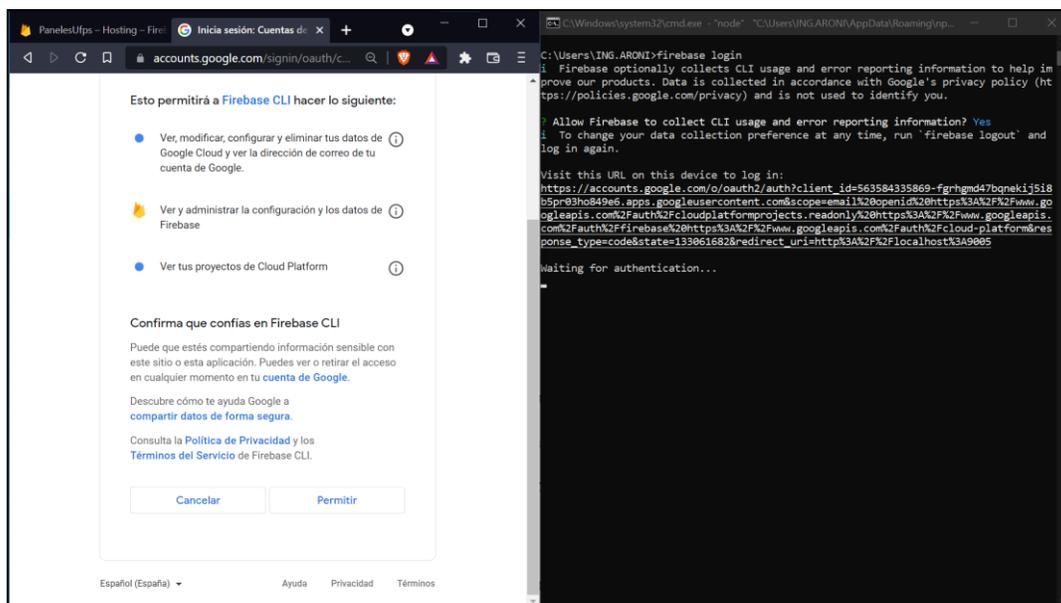


Figura 70. Autorización de permisos del Firebase CLI

Completando el acceso de parte de Firebase CLI, se inicia sesión con acceso de la cuenta Gmail, usada para los servicios de Hosting (ver figura 71).

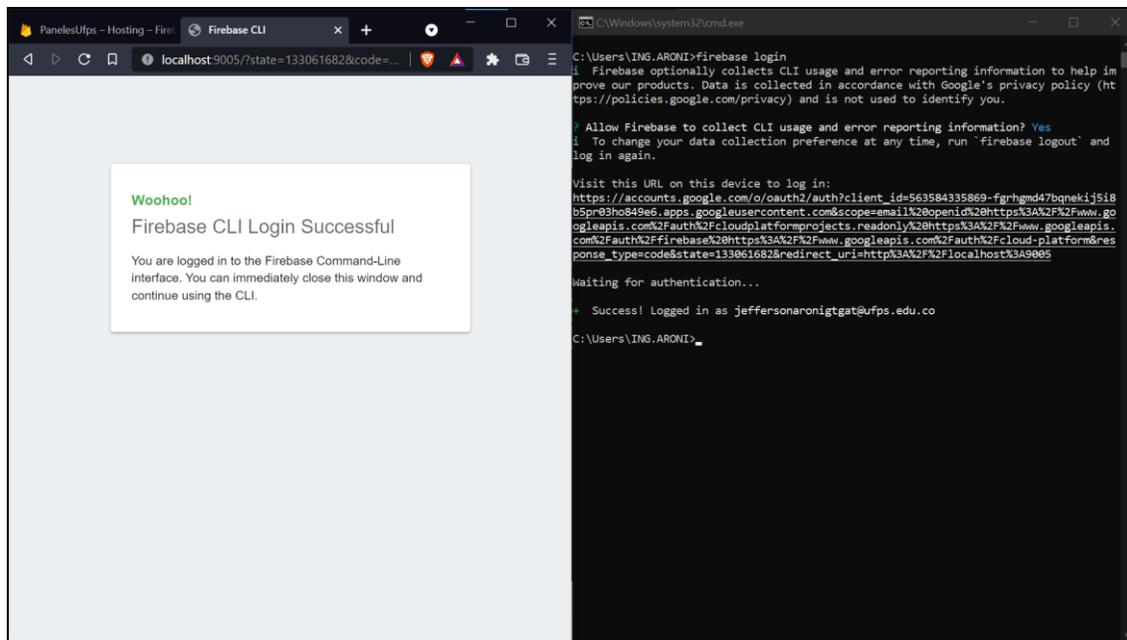


Figura 71. Inicio de sesión del Firebase CLI, en la cuenta de Gmail, para el despliegue

Con el acceso al gestor de despliegue se permite seleccionar la opción viable para el mismo, de la cual se seleccionó el despliegue en Hosting, con la opción de despliegue en GitHub, de lo contrario se realizaría el despliegue en las dos plataformas.

Es de importancia tener almacenado en la carpeta public los documentos de la aplicación Web, pues esta carpeta es a la que se tiene acceso con NodeJs, para realizar el despliegue (ver figura 72).

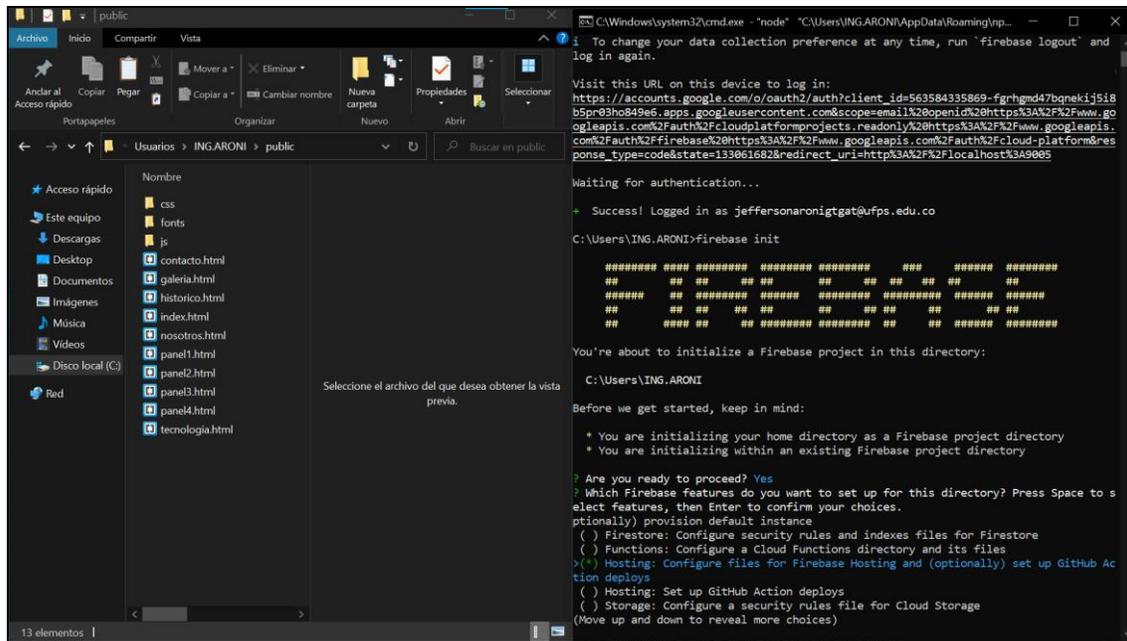


Figura 72. Opciones de despliegue en Hosting de Firebase

Como paso final se realiza el despliegue con el comando `firebase deploy` (ver figura 73).

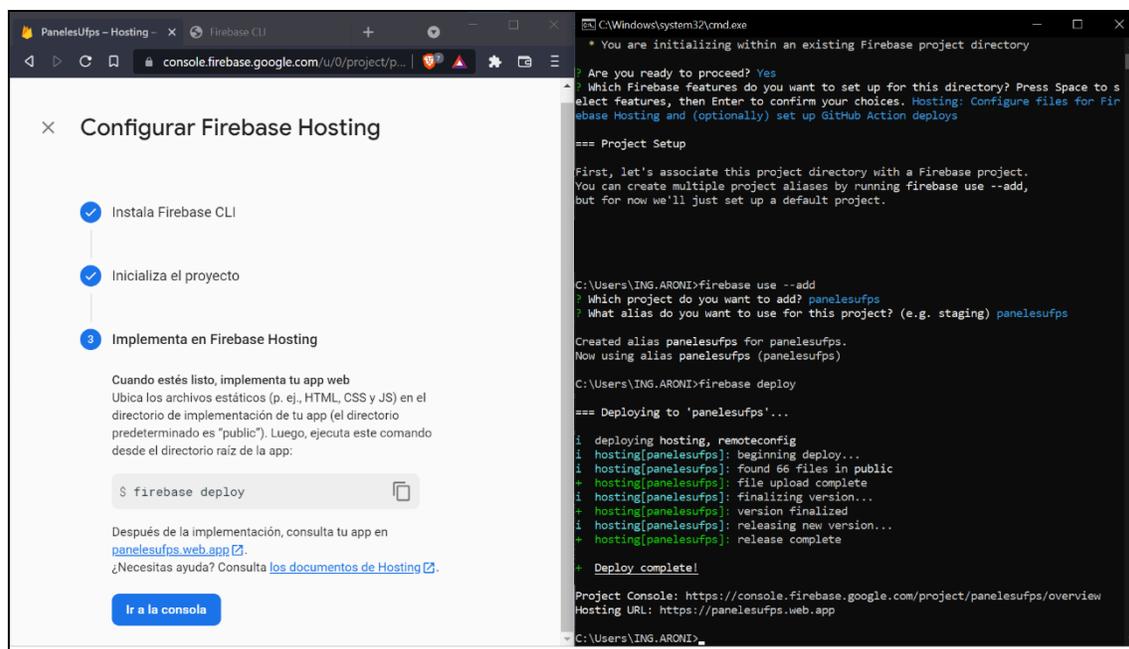


Figura 73. Despliegue de la aplicación Web en el Hosting

De no realizar la correcta autorización de NodeJs, y Firebase CLI a las dependencias necesarias, puede causar algún fallo en el despliegue, que impide el uso del Hosting, por lo que se recomienda remitirse a la realización de los pasos con mayor detenimiento para corregir el error, o revisar la documentación de Firebase, por si se anexaron nuevos procedimientos, o modificaciones para el despliegue de las App en los servicios de Hosting.

En ocasiones algunos archivos no son soportados por el hosting, o impiden el despliegue, por lo que es recomendable verificar los documentos e imágenes de la App; los formatos de estos pueden no ser soportados, por lo que es necesario cambiar los formatos para continuar.

Al culminar la etapa de despliegue de la aplicación en el Servidor, se genera el dominio <https://panelesufps.web.app/>, este dominio es la dirección Web con la que se tendrá acceso a la aplicación desde cualquier dispositivo y donde se visualiza la interfaz (ver figura 74).

The image shows a VNC viewer window on the left displaying a Raspberry Pi terminal. The terminal shows the execution of a Python script named 'panelesufps.py'. The script outputs real-time data for four solar panels (PANEL 1 to PANEL 4), including voltage, current, power, and resistance. The data is as follows:

PANEL 1	PANEL 2	PANEL 3	PANEL 4
Voltaje: 3.996V	Voltaje: 4.039V	Voltaje: 8.429V	Voltaje:
Corriente: 41.30mA	Corriente: 41.75mA	Corriente: 87.13mA	Corriente:
Potencia: 0.165W	Potencia: 0.169W	Potencia: 0.734W	Potencia:
Resistencia: 96.75	Resistencia: 96.75	Resistencia: 96.75	Resistencia:
Fecha: 2022-01-11	Fecha: 2022-01-11	Fecha: 2022-01-11	Fecha: 2022-01-11
Hora: 12:54:01.69	Hora: 12:54:01.69	Hora: 12:54:01.69	Hora: 12:54:01.69

The right side of the image shows a web browser displaying the application's interface. The title is 'Supervisión Remota de las Variables de los Paneles Solares'. It features a navigation menu with 'INVESTIGADOR' selected. Below the menu, there is a section titled 'Si eres investigador' with a 'Contacto' button. At the bottom, there is a table titled 'VARIABLES EN LOS PANELES' which displays the real-time data for the four panels, matching the data shown in the terminal.

Figura 74. Lectura de datos en tiempo real, en aplicación Web desde la nube

Con la aplicación desplegada se realizó una encuesta de “Evaluación de Interfaz Web de App para Paneles de la UFPS”, donde accedieron con su correo institucional 31 personas entre, estudiantes, docentes e investigadores del departamento de Ingeniería Electrónica, que interactuaron con la aplicación y respondieron el cuestionario (ver figura 75):

Evaluación de Interfaz Web de APP para Paneles de la UFPS

Evaluación de facilidad de uso, institucionalidad, y acceso a los datos de la interfaz Web de la App:

<https://panelesufps.web.app/>

Usuario: prueba@gmail.com

contraseña: 123456

Figura 75. Encuesta de evaluación de App Web

La primera pregunta se enfocó en conocer el dispositivo del que se accedió a la aplicación, donde la mayoría con 41,9% de los usuarios utilizó un Smartphone, para conectarse (ver figura 76)



Figura 76. Resultados de primera pregunta

La segunda pregunta se enfocó en conocer desde que sistema operativo se accedió a la aplicación, y aunque, puede tenerse un contraste por la cantidad superior de Smartphones

comparadas con equipos de cómputo, se recuerda que los teléfonos inteligentes tienen normalmente 3 sistemas operativos, Android, IOS y Windows Phone, por lo que conocer el dispositivo no es suficiente para determinar los sistemas operativos de mayor uso para la aplicación; por lo que es necesario también conocer los sistemas operativos de donde se accedió, donde el 35,5% accedió desde un dispositivo con sistema operativo Windows (ver figura 77).

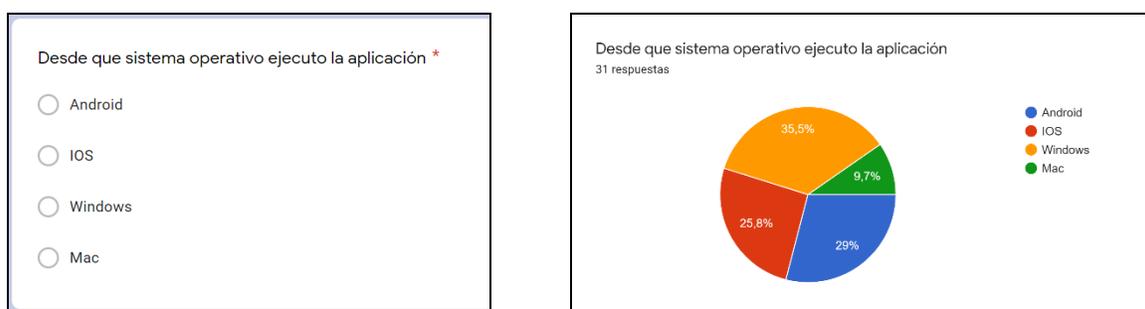


Figura 77. Resultados de segunda pregunta

La tercera y cuarta pregunta, se enfocaron en saber si la aplicación contenía algún error de programación, donde se accediera a los datos, sin ningún registro autorizado, o si la interfaz no se adaptaba al dispositivo. Por eso se preguntó si al acceder a la aplicación se tuvo acceso a los datos cuando no se había realizado el inicio de sesión, donde ningún usuario tuvo acceso a ellos, teniendo un 100% de seguridad a los datos; en cuanto a si la aplicación se ejecutaba correctamente en su dispositivo, solo un usuario reportó que la aplicación no se ejecutó correctamente, lo que se representó como un 3,2% de fallos en la aplicación, pero se aclara que este fallo fue generado por la mala conexión de Internet en el dispositivo, lo que evita la correcta visualización de la aplicación en el Browser, que es lo que sucede cuando se usa el navegador con malas conexiones de red. El 96,8% de los usuarios tenían una buena conexión que permitió la correcta ejecución de la aplicación (ver figura 78).

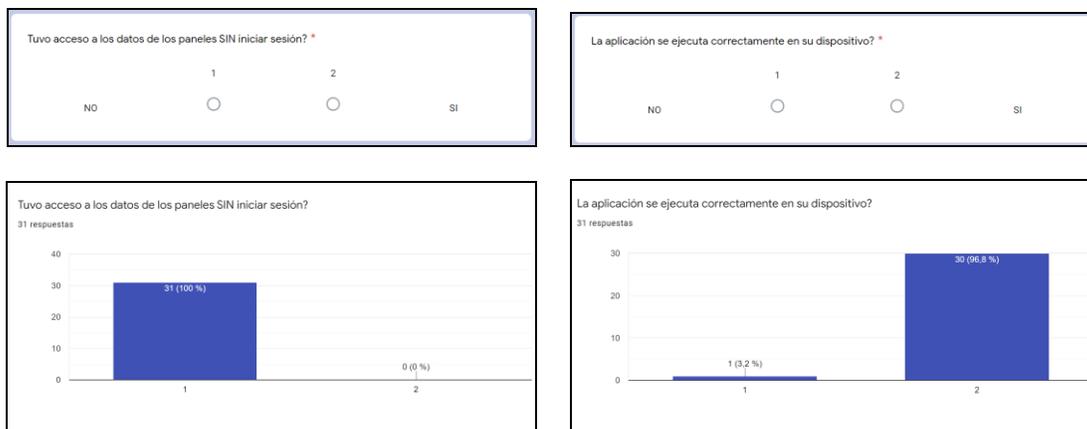


Figura 78. Resultados tercera y cuarta pregunta

Las últimas preguntas se enfocaron en la apariencia de la aplicación donde el 93,5% de los usuarios tuvo una experiencia satisfactoria de la aplicación, la cual se adaptó correctamente a los tamaños de pantalla de sus dispositivos, y evaluaron los colores, como agradables para la aplicación (ver figura 79).

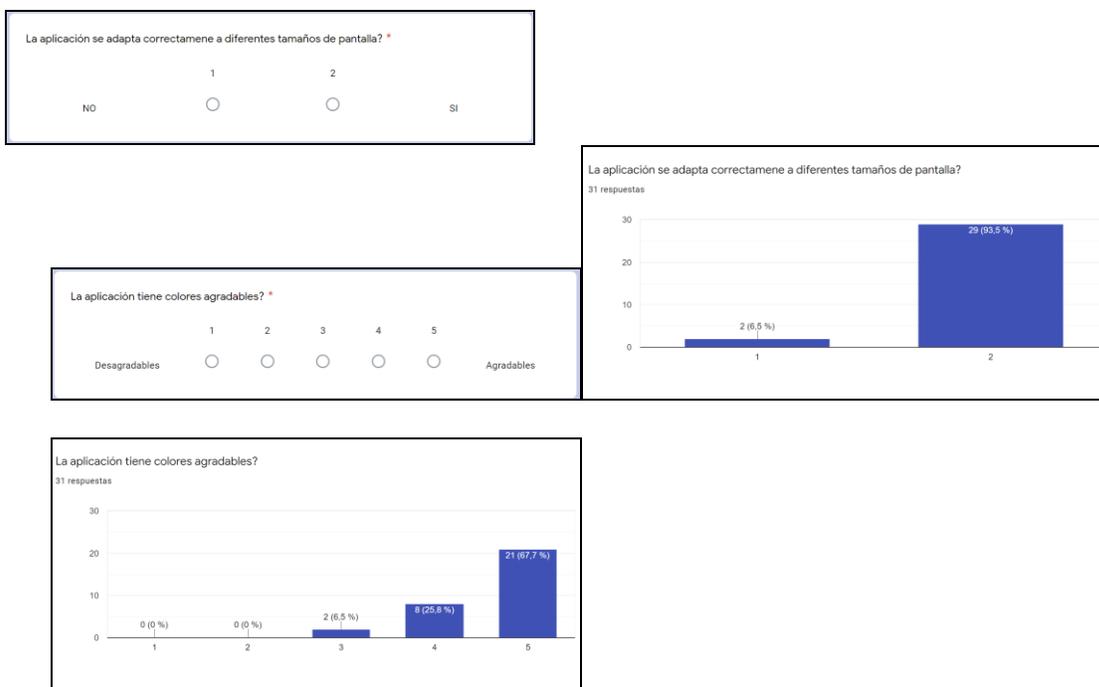


Figura 79. Resultados de quinta y sexta

Culminando con las pruebas de funcionamiento en la implementación del prototipo, se describe el diagrama de conexión del mismo (ver figura 80).

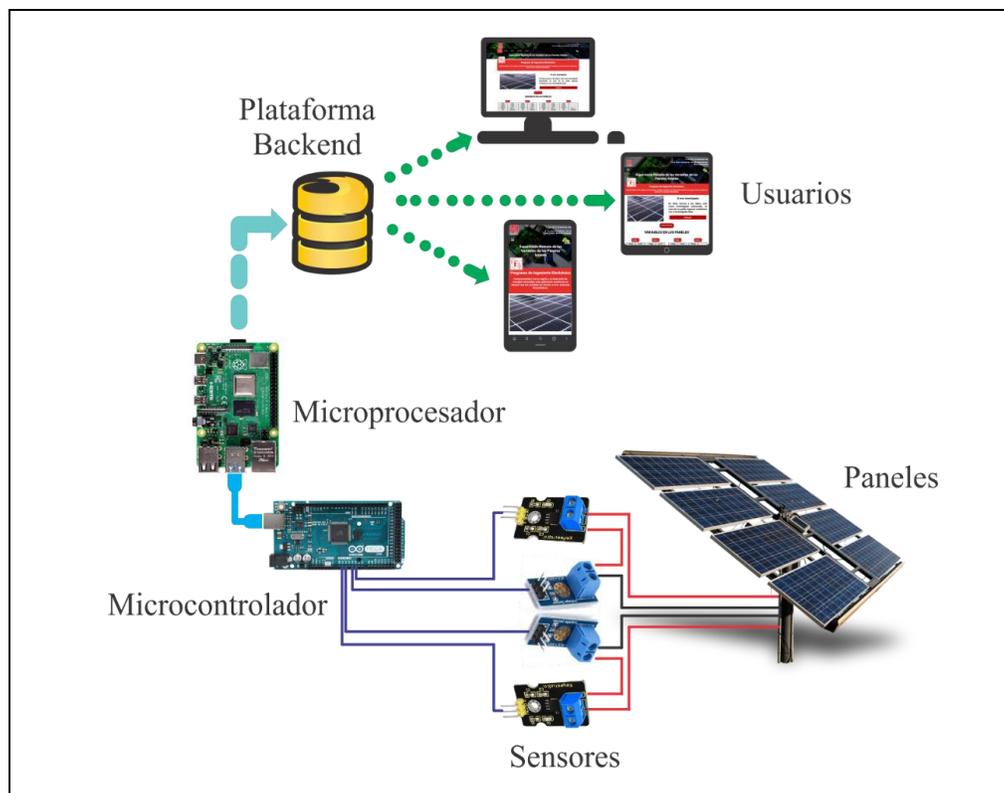


Figura 80. Diagrama de conexión del sistema de supervisión

4.5 Divulgación de Resultados y Beneficios de Nuevas Tecnologías en Industria 4.0

Uno de los objetivos de este proyecto investigativo, fue promover el desarrollo de instrumentación electrónica, integrada con aplicaciones remotas gestionadas con tecnologías de la información; generando soluciones viables a las instituciones de la región, que necesitan conectar su instrumentación a la nube, e interesándolas en el desarrollo de tecnologías 4.0.

La viabilidad del diseño, y la comprobación de funcionamiento en un prototipo funcional que gestiona el acceso a los datos de forma remota, con restricción de los mismos, solo a usuarios autorizados, facilitó el interés en la divulgación de estos resultados, a comunidades científicas de

incidencia internacional, medios de comunicación de incidencia nacional, y los principales eventos regionales desarrollados para tecnología, innovación y energías renovables.

El Sistema de supervisión para las variables en los paneles solares, desde la nube, fue aceptada como ponencia para el III Congreso Internacional de Investigación Aplicada, Creación e Innovación: Desarrollo Sostenible y Tendencias Tecnológicas (ver figura 81).

Bogotá, D.C septiembre de 2021

Respetado ponente

En nombre del Servicio Nacional de Aprendizaje SENA, Centro de Gestión de Mercados, Logística y Tecnología de la Información Regional Distrito Capital (Bogotá-Colombia), le extendemos un cordial saludo. Es para nosotros muy grato informarle la aceptación de su participación en el III CONGRESO INTERNACIONAL DE INVESTIGACIÓN APLICADA, CREACIÓN E INNOVACIÓN: DESARROLLO SOSTENIBLE Y TENDENCIAS TECNOLÓGICAS.

Eje Temático	Eje Temático Telenformática
Modo	Virtual via StreamYard, transmitido por YouTube
Fecha	Septiembre 21 de 2021
Horarios	Jornada tarde de 2:00pm a 5:00 pm
Enlace de ponentes.	https://streamyard.com/loqz66x6
Plantillas de presentación	Descargar
Ponentes	2:00 pm - 2:20pm José Miguel Cárdenas Serpa – Jeyber Adrán Gómez García 2:20 pm - 2:40pm Ivón Cristela Benítez González 2:40pm -3:00pm Jefferson Aroni Garcés Torres 3:00 pm - 5:00pm Alfredo Ordóñez Casanova*
Recomendaciones	Conexión a internet estable, preferiblemente por cable. Excelente iluminación fondo de color claro y probar previamente sonido. Ingresar min 15 min antes del desarrollo de su sesión.

Figura 81. Mensaje de aceptación, y horario de presentación de la ponencia en el iii congreso internacional de investigación aplicada, creación e innovación: desarrollo sostenible y tendencias tecnológicas

Se logro la divulgación de los resultados ante comunidad científica, nacional e internacional, interesados en el uso de nuevas tecnologías conectadas y gestionadas desde la nube, para la aplicación de industria 4.0, obteniéndose certificado de participación (ver figura 82).



Figura 82. Certificado de participación

La investigación del Sistema de supervisión para las variables en los paneles solares, desde la nube, despertó interés de medios audiovisuales interesados en la divulgación de tecnología e innovación, recibiendo una entrevista en vivo, el día 4 de noviembre del 2021 para el canal TRO, en la sección de “Mentes ingeniosas” del magazín Café de la Mañana (ver figura 83), canal televisivo de la región de los Santanderes, conocido por divulgación de las novedades, y desarrollos de la región; mostrando al país, como los investigadores de la región trabajan para llevar los nuevos desarrollos tecnológicos a las instituciones.

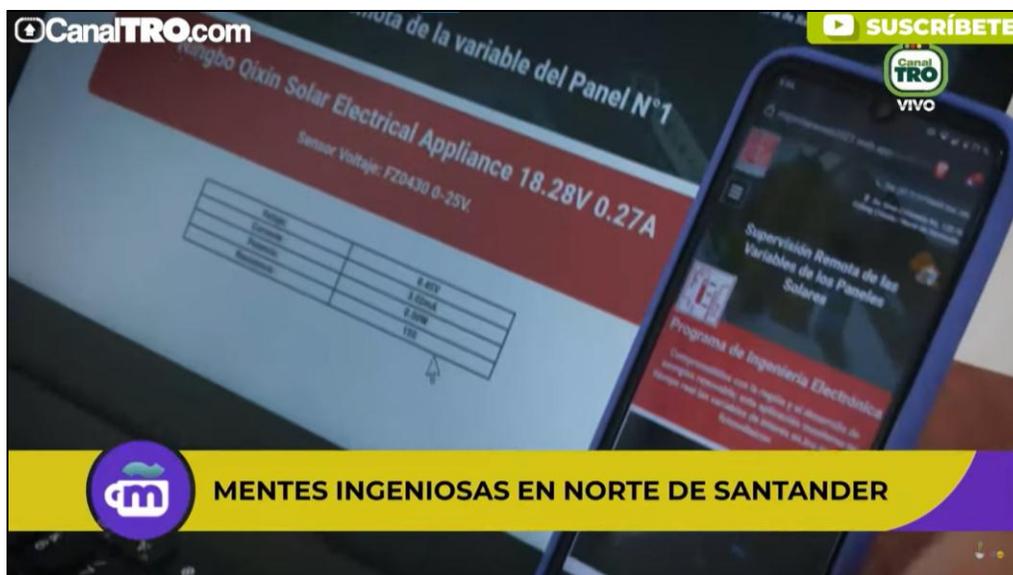


Figura 83. Trasmisión en vivo, en Café de la mañana, del Canal TRO

La transmisión en vivo de la entrevista, se realizó en televisión nacional el día 4 de noviembre del 2021, por medio del canal TRO, en el horario de 6:30 am, transmisión matutina (ver figura 84).



Figura 84. Entrevista en vivo, en Café de la mañana, del canal TRO

Se dejó referencia audiovisual en el canal de YouTube oficial del programa, que es posible ver en el enlace <https://www.youtube.com/watch?v=eHZiZq09Sr0> que, durante el desarrollo de

este libro, tuvo más de 650 visualizaciones, posicionándolo como el video más visto de la sección, y el segundo con más vistas en la historia del canal (ver figura 85).



Figura 85. Video de la entrevista en vivo, en Café de la mañana, del canal TRO, en el canal de YouTube

Cabe destacar que, con el fin de ampliar la divulgación, se compartió la información en grupos y redes sociales, generando en diferentes comunidades interés en estos nuevos desarrollos.

Gracias a la difusión de la investigación, y la acogida positiva por las comunidades e instituciones, el proyecto recibió la invitación a su presentación en un Stand de la tercera versión de la feria “Cúcuta Digital Valley 2021”, considerado el evento de tecnología más importante de la región, organizado por la Cámara de Comercio de Cúcuta, el Clúster Nortic y la Alcaldía de Cúcuta figura 86 (Diario la Opinión, 2021).



Figura 86. Tercera versión de la feria Cúcuta Digital Valley 2021

En esta feria se tuvo un espacio el día viernes 12 de noviembre del 2021, en las instalaciones de la biblioteca pública Julio Pérez Ferrero, en el horario de 1:00 pm hasta las 6:00 pm donde se tuvo acceso por parte de empresarios, estudiantes, investigadores, y personas del común, para conocer los detalles del desarrollo de la investigación, e interactuar más de cerca con el funcionamiento y las tecnologías utilizadas en el desarrollo del proyecto, mostrando de forma directa y presencial los beneficios del uso de la instrumentación electrónica, integrada con tecnología de la información, para la realización de módulos de industria 4.0 (ver figura 87).



Figura 87. Stand para exposición del prototipo, en la tercera versión de la feria Cúcuta Digital Valley 2021

Incentivados por los resultados obtenidos en la investigación, se recibió por medio del Tecnoparque Nodo Cúcuta, una invitación a la presentación de proyectos de innovación, con José Miguel Gonzales C. el gerente de CENS y su equipo de asesores, para conocer más a detalle, el prototipo, su funcionalidad, tecnologías usadas y como por medio de este tipo de investigaciones, se puede llevar energía solar, a sitios de difícil acceso, o regiones aisladas sin conectividad a la red convencional (ver figura 88).



Figura 88. Presentación del prototipo a los funcionarios del Tecnoparque Nodo Cúcuta, gerente de CENS y asesores

También el proyecto se postuló a la segunda versión del concurso de creatividad, innovación, tecnología y desarrollo de prácticas sostenibles, “INNOVACION 2021, desafíos que iluminan, soluciones que potencian nuestro crecimiento”, organizado por CENS, y sus aliados estratégicos, secretaria del Banco del Progreso y la Alcaldía de Cúcuta (ver figura 89).

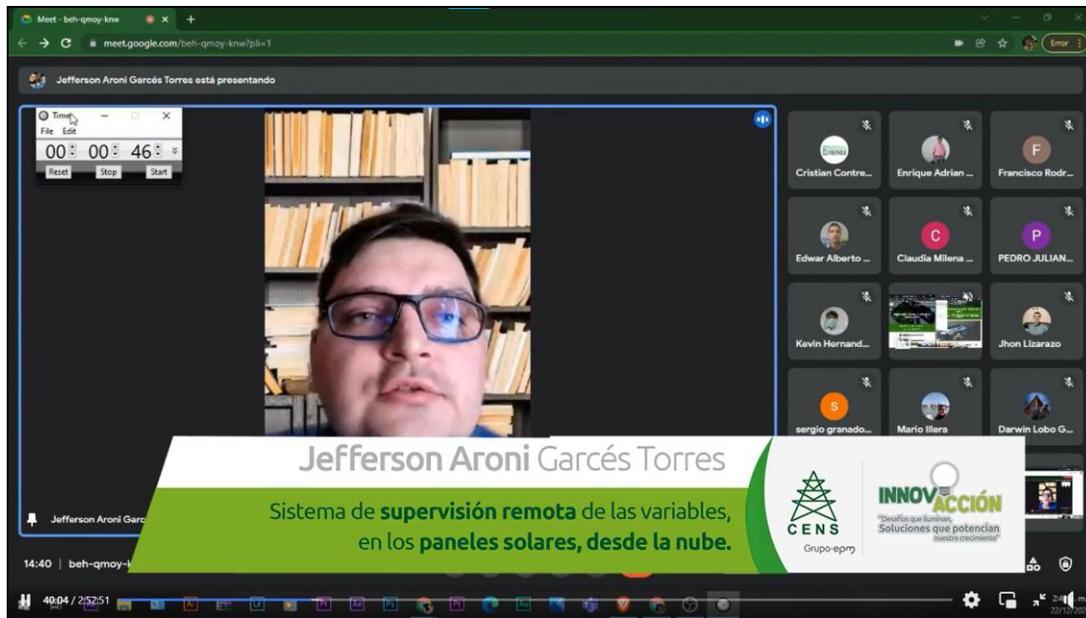


Figura 89. Concurso de la segunda versión, INNOVACION 2021, CENS

El evento se transmitió en vivo en los canales oficiales de CENS, y en Facebook Live, el día 22 de diciembre del 2021, en el horario de 2:00 pm, donde quedo registro del concurso en la página social de Facebook de la empresa y el cual es accesible en el siguiente enlace <https://www.facebook.com/CENSGrupoEPM/videos/1627334584276156>.

El proyecto tuvo su aparición, en el minuto 38:00, en un tiempo máximo de 2 minutos para realizar la presentación del proyecto, y donde se realizaba una sección de preguntas, por parte de los jurados. Uno de los beneficios que más se resaltó, es que era modular, y brinda la oportunidad de adaptarse a la institución que lo requiere (ver figura 90).

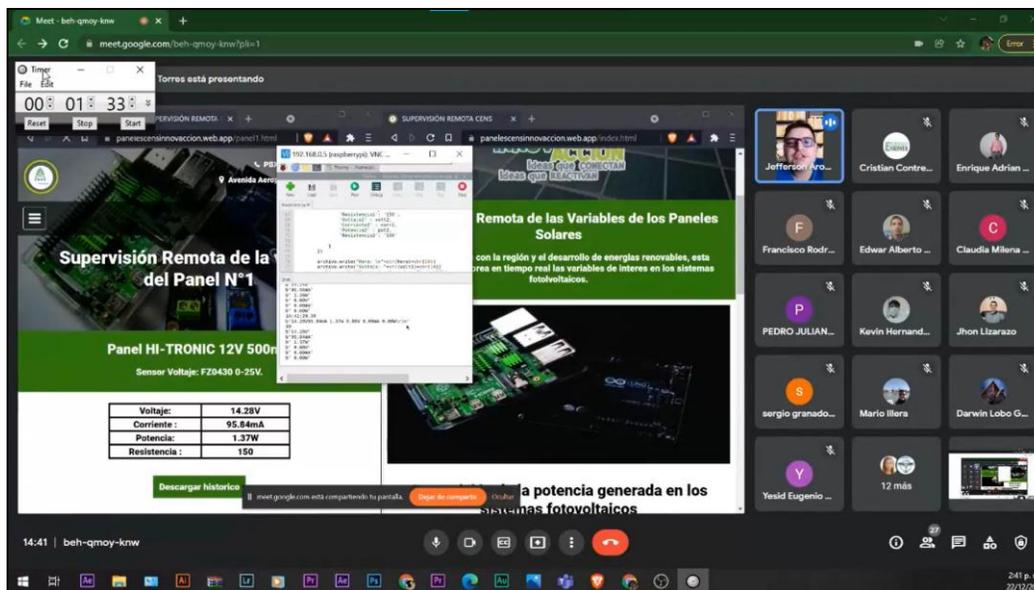


Figura 90. Presentación del proyecto enfocado en la segunda versión, INNOVACION 2021, CENS

El proyecto fue seleccionado como finalista, demostrando el interés por parte de la empresa comercializadora de energía, en este tipo de tecnologías, y para la supervisión remota de la generación en sistemas fotovoltaicos (ver figura 91).

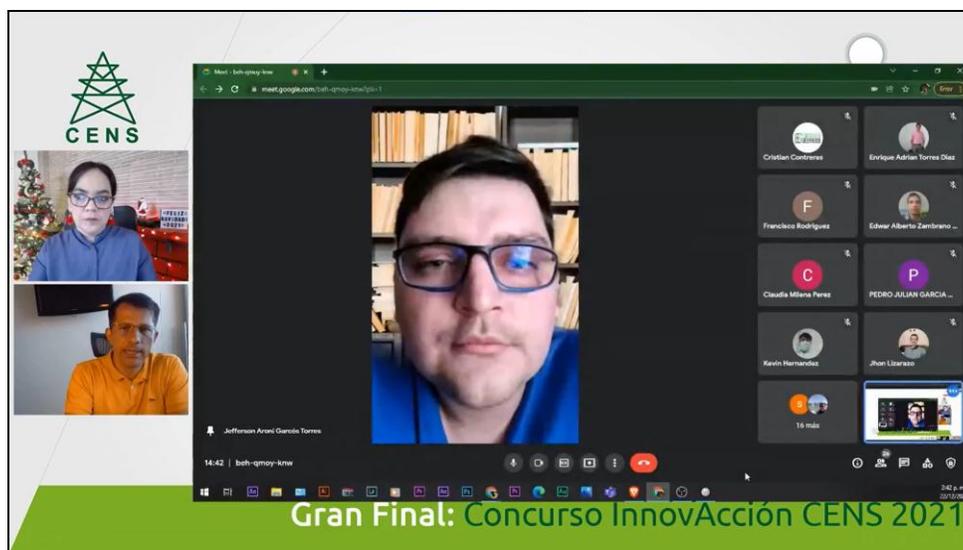


Figura 91. Final de segunda versión, INNOVACCION 2021, CENS

5. Conclusiones

La recopilación y estudio de información permitió determinar qué tipo de aplicaciones son demandadas por las empresas en el mercado, y como la electrónica de bajo costo, sistemas embebidos y tecnologías de la información, está siendo aplicados para la gestión energética de pequeñas plantas solares, monitorización de eficiencia en los paneles y diagnóstico de funcionamiento para diseños fotovoltaicos.

Tomando como referencia la matriz Pugh se puede elaborar un método de evaluación que facilite la selección de cada plataforma secuencialmente, y descarte las de menor viabilidad, sobre la que más beneficios ofrece al proyecto, para el diseño de un prototipo.

La Raspberry Pi 3b+ tiene un 94,2% de viabilidad, como plataforma computacional, comparada con plataformas similares de otros fabricantes, y es superior por su cantidad de memorias de almacenamiento y RAM, protocolos de comunicación, y disponibilidad en los mercados locales, además de su precio.

El Arduino Mega 2560 tiene un 80% de viabilidad, como plataforma de instrumentación que cuenta con un conversor ADC de hasta 16 canales analógicos, de medición, tiene como respaldo la placa Arduino UNO R3, que permite las mismas labores, para un máximo de 6 canales analógicos de medición requeridos.

El Sensor KS0185 es la opción de mayor viabilidad para la lectura de corrientes a bajo costo, pues permite la lectura de corrientes directa y alterna, bajo la misma plataforma.

El avance en tecnologías de la información, ofrece tecnologías multiplataformas que, permiten el desarrollo de aplicaciones Web con las mismas o mejores prestaciones que las

aplicaciones nativas de los dispositivos.

Las aplicaciones Web, son ejecutadas en los motores de búsqueda de Internet (navegadores o Browser), y no sobre el sistema operativo; lo que permite la ejecución de potentes interfaces, sobre plataformas de bajos recursos computacionales, y adaptando la aplicación a las características del dispositivo.

El diseño de interfaces Web, no es regido por las normas ANSI/ISA de diseño HMI convencional, y son desarrolladas bajo criterios de minimalismo, en diseño gráfico enfocado a marketing y publicidad, generando experiencias intuitivas, y agradables en el usuario, que lo llevan a la frecuencia de uso.

La calibración de los sensores se debe realizar con fuentes constantes antes de ser conectados al panel fotovoltaico, pues la dependencia de la radiación resulta en una variación permanente de las mediciones, lo que impide la calibración de la instrumentación.

Para que la visualización de los datos sea en tiempo real, la actualización de los mismos en la interfaz se configuro cada 15 segundos, lo que es menor a la frecuencia de almacenamiento, que se promedió en 5 minutos para un total de 120 mediciones, que representan 10 horas de luz Sol.

Los archivos .cvs permiten la manipulación de la información en software de gestión de datos comunes para los investigadores, y que brindan herramientas para tabular, organizar, y procesar la información según la necesidad de la investigación.

En la divulgación de los resultados se comprobó el interés de parte de las instituciones de la región, por módulos de supervisión remota en la medición energética de los sistemas fotovoltaicos, y con ello tener herramientas de acceso remoto para la gestión energética en

sistemas fotovoltaicos.

Los canales de televisión que divulgan desarrollos tecnológicos, acercan los resultados de investigaciones científicas, a diferentes comunidades que normalmente no tiene acceso a estos conocimientos, y facilita a las instituciones a conocer avances en las tecnologías de interés para la solución de sus necesidades.

6. Recomendaciones

Para la administración de la aplicación Web, se recomienda designar un investigador líder, que gestione los usuarios autorizados, y el manejo de la consola de Firebase; evitando con ellos, darle acceso a personas que puedan alterar las bases de datos, o los usuarios con acceso a la información.

En la calibración del sensor de efecto Hall KS0185, se debe considerar los campos magnéticos del sitio, que, de no considerarse, alteran las mediciones.

Para el módulo de supervisión remota, puede utilizarse la placa Arduino UNO R3, cuando los variables de medición analógica, no superan un total de 6 canales, lo que reduce mucho más el costo de implementación de este tipo de tecnologías.

La aplicación Web, puede ser rediseñada, para futuros convenios, trabajos, o proyectos que necesiten un modulo de supervisión remota, usando nuevas ventanas de información, o configurando las actuales.

La instrumentación utilizada, permite la conexión de otros sensores de interés para los investigadores, con lo que se podrá leer radiación y temperatura, la cual también determina la eficiencia del sistema fotovoltaico.

Referencias Bibliográficas

- Agurto, D. (2020). *Integración de u sistema de adquisición de datos mediante el uso de un Arduino Mega y Raspberry Pi3 como servidor Web y Base de Datos*. Tesis de grado. Universidad Nacional de Piura. Piura, Perú.
- All Data Sheet. (2021). *ACS712 Datasheet (PDF) - Allegro MicroSystems*. Recuperado de: <https://pdf1.alldatasheet.com/datasheet-pdf/view/168326/ALLEGRO/ACS712.html>
- Arduino. (2021). *¿Qué es Arduino?*. Recuperado de: <https://arduino.cl/que-es-arduino/>.
- Arquillo, R. (2019). *Gestión y monitorización de instalaciones eléctricas trifásicas mediante Arduino y Raspberry Pi*. Tesis de grado. Universidad de Sevilla. Sevilla, España.
- Barón, A. (2017). *Monitorización de la energía consumida mediante Raspberry Pi para sistema domótico Alejandro Barón Cuevas*. Tesis de grado. Universidad Carlos III de Madrid. Madrid, España.
- Borges, L. (2009). *Python para desarrolladores*. Rio de Janiero: Creative Commons.
- CARROD. (2021). *Módulo Sensor FZ0430 Medidor de Voltaje Breakout semiconductor refacciones electronicas componentes electronicos todo en electronica tienda de electronica en Torreon tienda en linea en Mexico*. Recuperado de: <https://www.carrod.mx/products/modulo-sensor-fz0430-medidor-de-voltaje-breakout>
- Castro, J. (2020). *Análisis de Eficiencia en Conversión/Almacenamiento de Energía Solar y Adquisición de Datos de Paneles Solares del Laboratorio de Física y Electrónica de la UNAD (Sede Nacional JCM) en el Marco de su Uso como Laboratorio Remoto*. Universidad

Nacional Abierta y a Distancia. Bogotá, Colombia.

Cepeda, J. & Sierra, A. (2017). Aspectos que afectan la eficiencia en los paneles fotovoltaicos y sus potenciales soluciones. *Revista NAFA SCR Doc*, 7(80), 1-17.

Charles, K. & Matthew, O. (2013). *Fundamentos de circuitos eléctricos*. Porto Alegre: McGraw Hill.

Congreso de Colombia. (2001). *Ley 697 de 2001. Mediante la cual se fomenta el uso racional y eficiente de la energía, se promueve la utilización de energías alternativas y se dictan otras disposiciones*. Bogotá: Diario Oficial No. 44.573.

De la Cruz, J. & Bastidas, Y. (2020). *Evaluación de tres modelos matemáticos de la eficiencia de un panel fotovoltaico*. Tesis de grado. Universidad Nacional del Centro del Perú. Huancayo, Perú.

De Vasconcellos, A. (2020). *Desenvolvimento de um sistema de supervisão, controle e manutenção baseado em condição para usinas de geração solar fotovoltaicas usando o conceito IoT*. Universidade de Brasília. Brasília, Brasil.

DFRobot. (2021). *Gravity_Analog Voltage Divider V2-DFRobot*. Recuperado de:
<https://www.dfrobot.com/product-90.html>.

Diario la Opinión. (2021). Hoy inicia 'cúcuta digital valley 2021'. *La Opinión*. Recuperado de:
<https://www.laopinion.com.co/cucuta/hoy-inicia-cucuta-digital-valley-2021>.

Ezequiel, A., Villalta, L., Clara, T. & Fuster, P. (2019). *Diseño del sistema de alimentación basado en energía solar de una estación de bombeo con control remoto*. Tesis de grado.

Universidad Politecnica de Valencia. Valencia, España.

Firestore. (2021a). *Firestore Hosting*. Recuperado de:

<https://firebase.google.com/docs/hosting/>
<https://firebase.google.com/docs/hosting?hl=es-419>.

Firestore. (2021b). *Firestore Realtime Database*. Recuperado de:

https://firebase.google.com/docs/database?hl=es-419#how_does_it_work.

Firestore. (2021c). *Cloud Firestore*. Recuperado de:

<https://firebase.google.com/docs/firestore?authuser=0>.

Firestore. (2021d). *Firestore*. Recuperado de: <https://firebase.google.com/>

Floyd, T. (2006). *Fundamentos de sistemas digitales*. Madrid: Pearson Prentice Hall.

Galeano, G. (2009). *Programación de sistemas embebidos en C*. México: Alfaomega.

González, K. (2017). *Automatización de un módulo fotovoltaico rotatorio de un eje con disponibilidad de datos de manera remota*. Tesis de grado. Instituto Tecnológico de Costa Rica. San José, Costa Rica.

Halcafree, G. (2016). *Raspberrypi guía del usuario 1*. Recuperado de:

<https://solectroshop.com/img/cms/Productos%20DICIEMBRE/Gu%C3%ADa%20de%20Raspberrypi%20Pi%20desde%20cero%20-%20Solectro.pdf>

Hilcu, M. (2021). ¿Qué es y cómo funciona la energía solar?. Recuperado de:

<https://www.otovo.es/blog/energia/que-es-la-energia-solar/>

International Business Machines. (2021). *Formato JSON (JavaScript Object Notation)* -

Documentación de IBM. Recuperado de:

<https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format>.

Jiménez, R. & Villarreal, C. (2021). *Desarrollo e implementación de un sistema detector de radiación solar, temperatura y humedad en la empresa Red Digital Telecomunicaciones S.A.S. Girardot*. Tesis de grado. Universidad Piloto de Colombia. Bogotá, Colombia.

Keystudio. (2021). *Free shipping! keystudio ACS712-5A Current Sensor for Arduino*

Compatible. Recuperado de: <https://www.keystudio.com/free-shipping-keystudio-ac712-5a-current-sensor-for-arduino-compatible-p0165.html>.

Larrea, A. (2016). *Registrador de datos de bajo coste y acceso remoto (Datalogger)*. Tesis de grado. Universidad Politecnica de Valencia. Valencia, España.

Llamas, L. (2021). *Medir voltajes de hasta 25V con Arduino y FZ0430*. *Luís Llamas*. Recuperado de: <https://www.luisllamas.es/medir-voltajes-de-hasta-25v-con-arduino-y-fz0430/>.

Mallegowda, M., Anithakanavlli, D. & Amrutha, M. (2020). Design and integration of middleware for IoT devices towards solar panel monitoring based on Raspberry Pi. *Journal Seybold Rep*, 4(2), 1-19.

Mateu, C. (2004). *Desarrollos de aplicaciones Web*. Catalunya: FUOC.

Ministerio de Minas y Energía. (2022). *RETIE, “Energía Eléctrica” – GM*. Recuperado de: <https://www.minenergia.gov.co/retie>.

Ministerio de Tecnología de las Información y las Comunicaciones (2019). *Aspectos básicos de la industria 4.0*. Recuperado de: https://colombiatic.mintic.gov.co/679/articles-124767_recurso_1.pdf.

Ministerio de Tecnologías de la Información y las Comunicaciones. (2021). *Tecnologías de la información y las comunicaciones*. Recuperado de: <https://www.mintic.gov.co/portal/inicio/Glosario/T/5755:Tecnologias-de-la-Informacion-y-las-Comunicaciones-TIC>

Montero, I., Norberto, D., Passos, A., Carvalho, P. & Silveira, S. (2020). Monitoramento em tempo real de temperatura de operação de uma planta FV. *VIII Congresso Brasileiro de Energia Solar–Fortaleza, 1(2)*, 1-11.

Mouser Electronics Colombia. (2021). *Distribuidor de componentes electronicos*. Recuperado de: <https://co.mouser.com/>

Muñoz, J. (2009). *Sistemas empotrados en tiempo real*. California: Creative Commons.

Navarrete, J. (2021). *Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi*. Tesis de grado. Universidad Católica de Santiago de Guayaquil. Guayaquil, Ecuador.

ONSET. (2018). *HOBO 4-Channel Analog Data Logger*. Recuperado de: <http://www.onsetcomp.com/products/data-loggers/ux120-006m>.

Orellana, D. & Sánchez, M. (2006). Técnicas de recolección de datos en entornos virtuales más usadas en la investigación cualitativa. *Revista de Investigación Educativa, 24(1)*, 205–22.

Pallás, R. (2003). *Sensores y Acondicionadores de Señal*. Bogotá: Alfaomega.

Pontes, W. & Souza, C. (2021). *Implantação de estação solarimétrica e de sistema supervisorio com SCADABR e plataforma IoT em unsina fotovoltaica na UNILAB-CE*. Universidade Federal do Ceará. Fortaleza, Brasil.

Programa en Línea. (2022). *Plataforma de desarrollo Backend, para aplicaciones Web*.

Recuperado de: <https://www.programaenlinea.net/firebase-plataforma-desarrollo-google/>

Puig, J. (2004). *CSS3 y Javascript avanzado*. Catalunya: FUOC.

Raafiu, B. & Agus, P. (2018). Smart Monitoring of Solar Panel System in Saving of the Electrical Power with Internet of Things. *Instrumentation, Control, Protection & Internet of Things*, 8(2), 156–165.

Raspberry Pi Fountion. (2021). *¿Que es Raspberry Pi?*. Recuperado de:

<https://raspberrypi.cl/que-es-raspberry/>.

Rodríguez, R. (2019). Internet de las cosas: Futuro y desafío para la epidemiología y la salud pública. *Revista Universitaria y Salud*, 21(3), 253–260. Doi: 10.22267/rus.192103.162.

Schmalb, R. (2021). *Monitorização de uma rede de painéis fotovoltaicos usando um sistema embarcado de baixo custo*. Tesis de grdo. Universidade de Évora. Évora, Portugal.

Sejzer, R. (2022). Calidad Total: La Matriz de Pugh para la toma de decisiones Recuperado de:

<http://ctcalidad.blogspot.com/2016/10/la-matriz-de-pugh-para-la-toma-de.html>.

Silvente, G. (2019). Sistema de bajo coste para la monitorización de instalaciones fotovoltaicas:

Aplicación de Arduino y Raspberry Pi. *23rd Int. Proceedings from the International*

Congress on Project, 10(2), 1407–1417.

Soares, I., Carneiro, I. & Oliveira, D. (2020). Desenvolvimento de um registrador de dados para a aplicação em um sistema de monitoramento fotovoltaico. *VIII Congresso Brasileiro de Energia Solar–Fortaleza*, 4(2), 1-7.

Spano, D. (2010). *El Open Source como facilitador del Open Access*. Buenos Aires Argentina: Comité Coordinador del Grupo Latinoamericano de Greenstone Prodigio Consultores.

Torres, F. (2016). *Análisis del marco normativo del sector eléctrico colombiano, impactos en la regulación eléctrica de la ley 1715 de 2014*. Tesis de grado. Universidad Nacional de Colombia. Bogotá, Colombia.

Universidad Francisco de Paula Santander. (2010). *Manual de imagen corporativa*. Recuperado de: <https://ww2.ufps.edu.co/public/archivos/pdf/34377aed8b11d395e95f15cb30c86085.pdf>.

Whelan, B. (1994). *Color Harmony 2: A Guide to Creative Color Combinations*. Recuperado de: <https://findit.library.yale.edu/catalog/digcoll:2785034>

Anexos

Anexo 1. Manual técnico

MANUAL TECNICO DE ADMINISTRADOR Y DE USUARIO

SISTEMA DE SUPERVISIÓN PARA LAS VARIABLES EN LOS PANELES SOLARES,
DESDE LA NUBE.

AUTOR

JEFFERSON ARONI GARCÉS TORRES

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

SAN JOSÉ DE CÚCUTA

2022

Contenido

	pag.
1. Adición de Nuevos Sensores al Sistema de Instrumentación	150
2. Conexión remota con Raspberry Pi por VNC.....	151
3. Administración de Consola de Firebase.....	154
4. Herramientas de Interfaz Gráfica de la App Web	157

Lista de Figuras

	pag.
Figura 1. Creación de las variables, y la función del sensor	150
Figura 2. Creación de la función y organización de los datos del nuevo sensor.....	150
Figura 3. Aplicaciones necesarias para la conexión inalámbrica.....	152
Figura 4. Escaneo de dispositivos conectados en la Red Wifi	152
Figura 5. Creación de conexión de VNC	153
Figura 6. Conexión con interfaz de Raspberry Pi	153
Figura 7. Consola de Firebase	154
Figura 8. Herramienta de Autenticación de Firebase	155
Figura 9. Creación de nuevo usuario.....	155
Figura 10. Usuario nuevo agregado	156
Figura 11. Menú de navegación de la aplicación	157
Figura 12. Página principal de la aplicación	157
Figura 13. Botones de contacto y de ingreso	158
Figura 14. Página de inicio y botones	158
Figura 15. Acceso a los datos de los paneles	159
Figura 16. Botón y tabla de datos del panel	160
Figura 17. Ventana dedicada de cada panel	160
Figura 18. Ventana de históricos.....	161
Figura 19. Descarga de históricos	161
Figura 20. Ventana nosotros.....	162
Figura 21. Ventana galería	162
Figura 22. Imagen seleccionada	163
Figura 23. Ventana de tecnologías	163
Figura 24. Ventana contacto.....	164
Figura 25. Menú en formato móvil	164

1. Adición de Nuevos Sensores al Sistema de Instrumentación

El módulo del sistema de supervisión remota de las variables en los paneles solares, cuenta con la posibilidad de conectar hasta un total de 16 sensores analógicos en sus entradas, por lo que, la implementación en sitio debe calibrar la instrumentación y conectarla con la placa de captación de datos.

El personal técnico que desee conectar más sensores, debe considerar crear las variables del nuevo sensor, y escribir el algoritmo que procese la información, y ajuste los valores a un total de 6 caracteres, (Figura).

```
//NUEVO SENSOR
void sensorNUEVO();
float sensorNUEVO = 0; //Lectura del pin AX(PIN NUEVO)
float sensorNUEVO = 0; //Lectura promedio
float corrienteNUEVO = 0; //Conversión a corriente
float voltajeNUEVO = 0; //Conversión a voltaje
float potenciaNUEVO = 0; //Conversión a potencia
```

Figura 1. Creación de las variables, y la función del sensor

La función del sensor debe contener la calibración del mismo, y la organización de los datos, para almacenar siempre en un máximo de 6 caracteres el valor, esto aprovecha los recursos del Arduino Mega 2560 y libera al procesador de la Raspberry Pi 3B+ de hacer esta tarea, (Figura 2).

```
InstrumentacionRaspiFirebase
void loop() {
  sensorACS172();
  sensorFZ0430();
  sensorSKUDFR();
  sensorNUEVO();
}
void sensorNUEVO(){
  sensorNUEVO = analogRead(AX);
  voltajeNUEVO = (SENSIBILIDAD)*sensorNUEVO;
  corrienteNUEVO = (FACTOR-INCREMENTO)*(voltajeNUEVO/resistencia);
  potenciaNUEVO = voltajeNUEVO*(corrienteNUEVO/(FACTOR DE AJUSTE));
  //Voltaje NUEVO
  if((voltajeNUEVO/100)>=1){
    Serial.print(voltajeNUEVO,3);
    Serial.print("V");
  }
  else if((voltajeNUEVO/10)>=1 && (voltajeNUEVO/10)<10){
    Serial.print(" ");
    Serial.print(voltajeNUEVO,3);
    Serial.print("V");
  }
  else {
    Serial.print(" ");
    Serial.print(voltajeNUEVO,3);
    Serial.print("V");
  }
}
//corriente NUEVO
```

Figura 2. Creación de la función y organización de los datos del nuevo sensor

2. Conexión remota con Raspberry Pi por VNC.

Para iniciar el procesamiento de los datos, realizar alguna modificación, seguimiento de fallos, o reconfiguración de la organización y envío de los datos, se debe interactuar con la Raspberry Pi, por lo que se aconseja hacerlo de manera remota con un dispositivo que permita la visualización de la interfaz, y cuente con los periféricos para la gestión de las acciones.

El método convencional es conectar una pantalla, mouse y teclado a la Raspberry Pi, pero este método en ocasiones puede ser engorroso, por necesitar de varios componentes para realizarlo, ya que una opción recomendable, es utilizar un dispositivo móvil que permita la gestión de la Raspberry Pi inalámbricamente.

Para la conexión remota, es necesario mínimo dos programas que permitan conocer la dirección IP de la Raspberry Pi 3b+, y con esta dirección realizar la conexión por protocolo VNC, el cual es activado en la configuración inicial de la plataforma.

Los programas sugeridos están en versión gratuita móvil, que permite al personal técnico tener las herramientas en su dispositivo Smartphone, facilitando la gestión de los recursos de la plataforma, sin requerir de otros componentes, a los que usualmente tienen a su disposición el personal técnico.

Las aplicaciones están en la Play Store y son descargables gratuitamente, (Figura 3).

Fing -Escaner de red: La versión usada es la gratuita 11.6.0. con actualización del 10 de enero del 2022, y con un tamaño de 41Mb.

VNC Viewer - Remote: La versión usada es la gratuita 3.7.1.44443 con actualización del 19 de marzo del 2021, y con un tamaño de 24Mb.

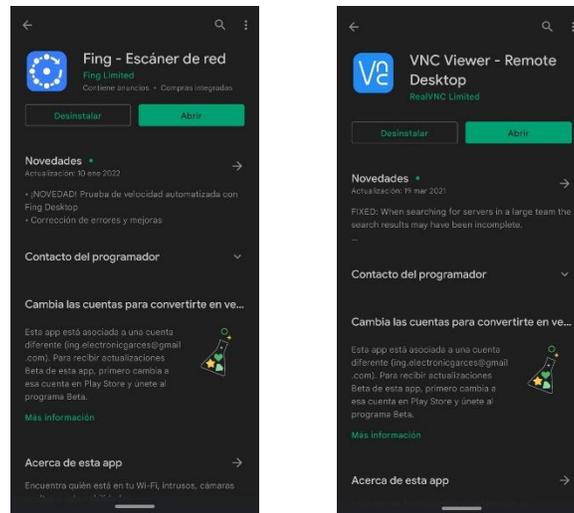


Figura 3. Aplicaciones necesarias para la conexión inalámbrica

Para iniciar la conexión se recuerda que el dispositivo debe estar conectado en la misma Red de internet de la Raspberry Pi 3b+, o no podrá ser encontrado por la aplicación.

La interfaz de **Fing** tiene la opción de escanear dispositivos conectados a la misma Red Wifi en la que se encuentra conectado el dispositivo, (Figura 4).

Buscar Dispositivos: Realiza un escaneo que visualiza las direcciones Ip de los dispositivos conectados a la misma Red de Wifi.

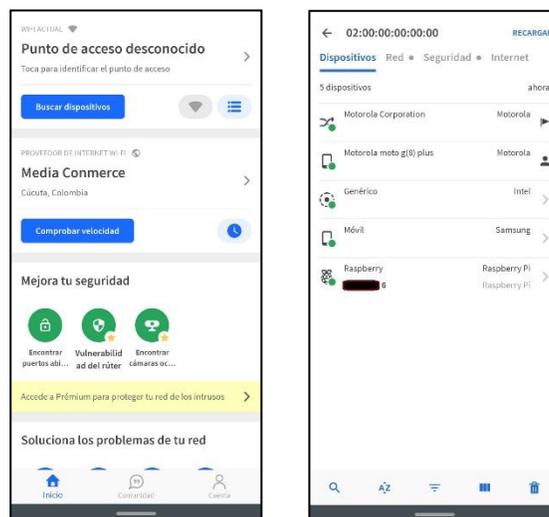


Figura 4. Escaneo de dispositivos conectados en la Red Wifi

Conociendo la dirección IP de la Raspberry Pi 3b+, ya se puede realizar la conexión por medio de VNC, ingresando a la aplicación y escribiendo la dirección del dispositivo, (Figura 5).

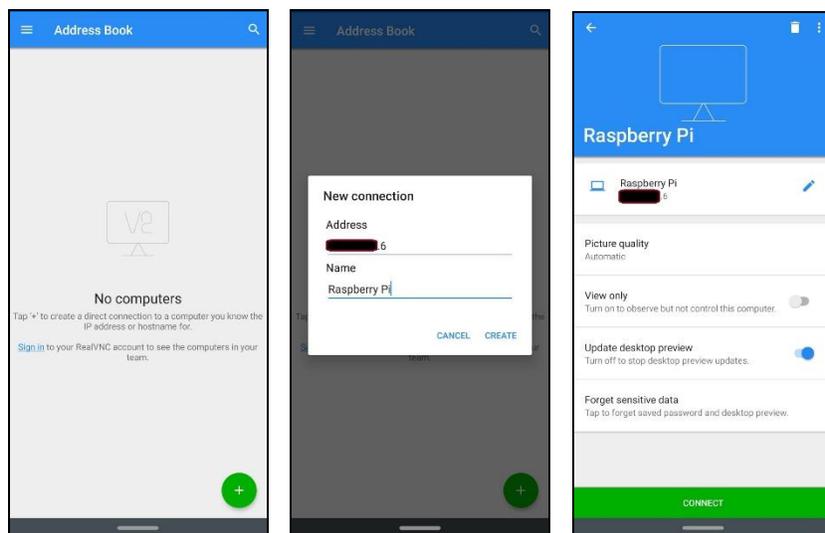


Figura 5. Creación de conexión de VNC

Con la creación del ambiente de conexión, se conecta a la Raspberry Pi 3B+ con el nombre y contraseña de usuario configurados en la misma, (Figura 6).

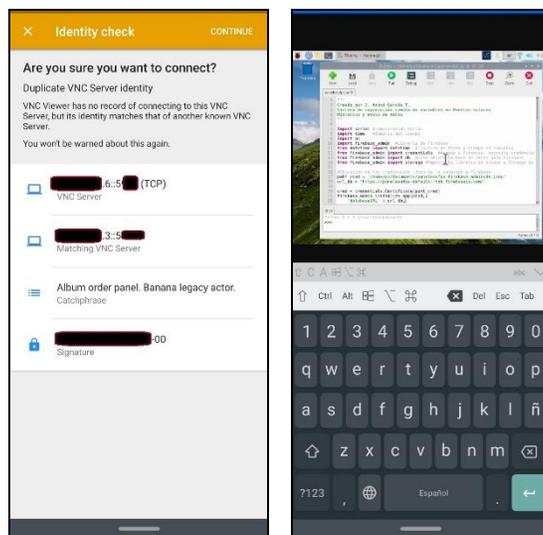


Figura 6. Conexión con interfaz de Raspberry Pi

Con la conexión se puede gestionar la Raspberry Pi 3b+ desde el dispositivo móvil, el cual ofrece las herramientas de pantalla táctil, teclado digital, ajuste de pantalla y Zoom; pero si el personal técnico prefiere, puede conectar un teclado a la Raspberry Pi 3b+ o al Smartphone, y gestionar solo la visualización desde la aplicación.

3. Administración de Consola de Firebase

La administración de los usuarios, sus contraseñas, los archivos y datos almacenados en la plataforma de Backend, son gestionados, desde la consola de Firebase, a la cual se le recomienda el acceso a una persona designada como el administrador del proyecto, pues desde esta plataforma se pueden eliminar o crear usuarios con acceso a la aplicación, o borrar información que dejará de ser publica para los usuarios autorizados.

El ingreso a la consola se realiza con el correo de Gmail usado para los servicios de Firebase; en el panel principal aparecen los proyectos que se administran con la cuenta, y donde se accede para los servicios de la misma, (Figura 7).

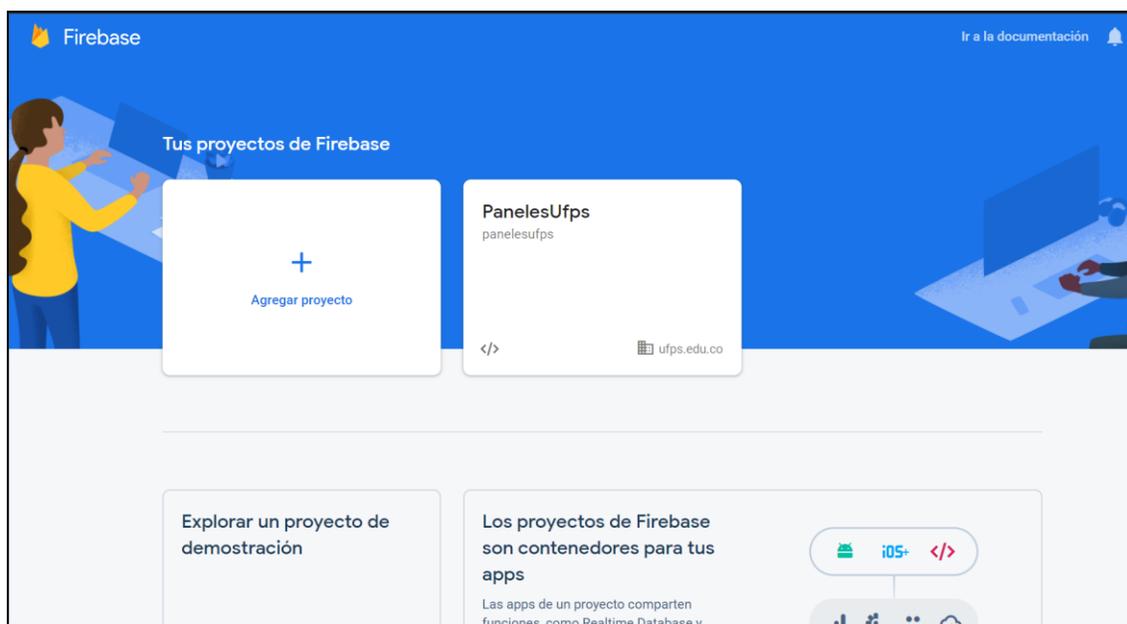


Figura 7. Consola de Firebase

Iniciando los servicios de Firebase, se debe seleccionar la herramienta de Autenticación, donde se gestionan los usuarios autorizados y sus permisos para con la aplicación, (Figura 8).

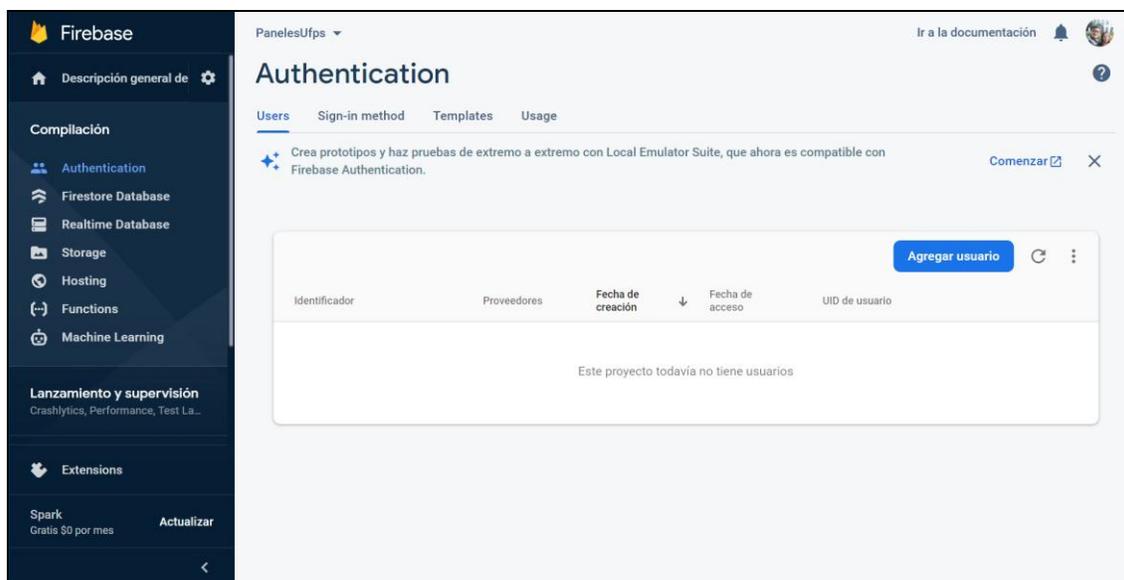


Figura 8. Herramienta de Autenticación de Firebase

Se resalta que en la ventana aparecen los usuarios que tiene acceso a la aplicación, y el que tenga acceso a la consola, puede agregar o eliminar los usuarios que desee, por lo que es recomendable que solo una persona, gestione estos servicios.

Para agregar un nuevo usuario, se debe anexar los datos del correo y contraseña, (Figura 9).

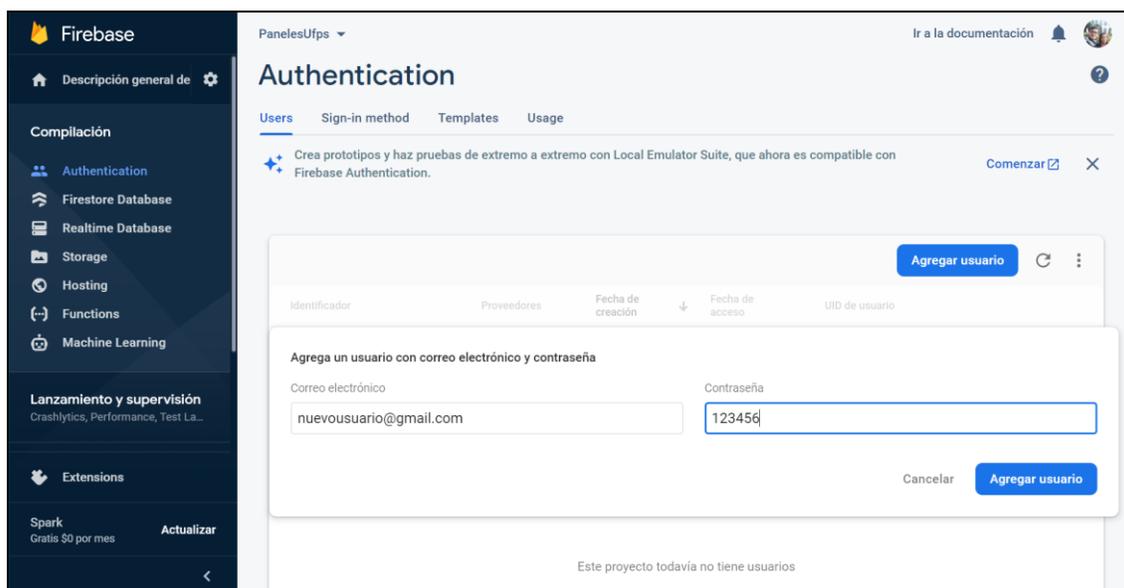


Figura 9. Creación de nuevo usuario

Al terminar la creación del nuevo usuario quedará registrado dentro de la aplicación, y recibirá los permisos automáticamente, mientras se mantenga como usuario autorizado, (Figura 10).

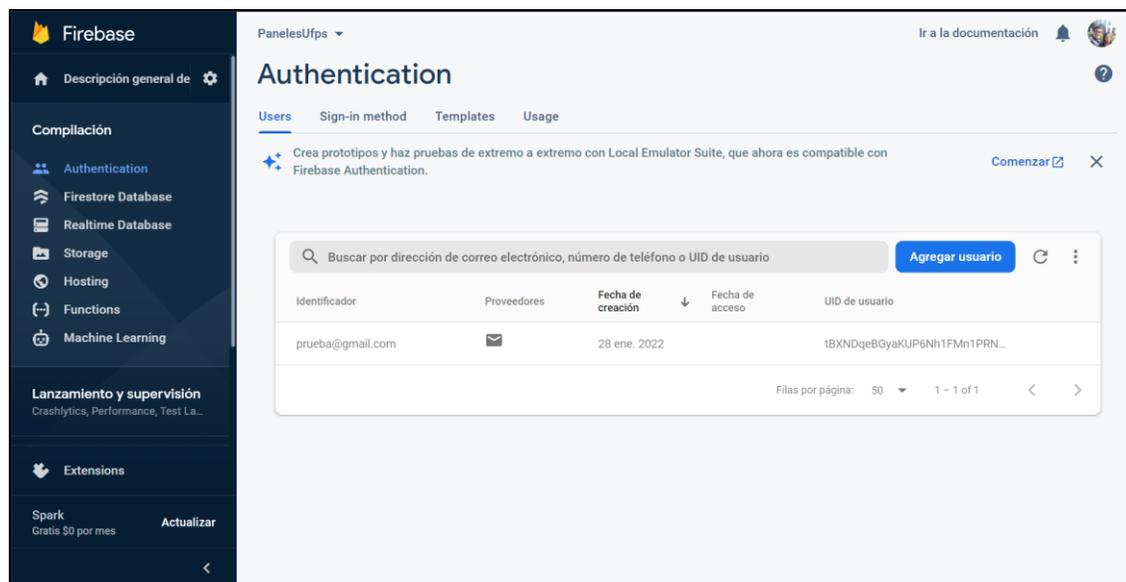


Figura 10. Usuario nuevo agregado

Por motivos de seguridad, la aplicación se configuró para la contraseña no sea cambiada por el usuario, sino suministrada por el administrador, y con ello, tener total control de los permisos a la información.

4. Herramientas de Interfaz Gráfica de la App Web

La aplicación se desarrolló siguiendo los protocolos generales de una aplicación Web, donde se tiene un menú de enlaces principales, que sectorizan la información en ventanas, para que el usuario pueda dirigirse a la que consideré de interés; también se incluyó un icono de inicio, que dirige al usuario a la pantalla principal desde cualquier otra, (Figura 11).

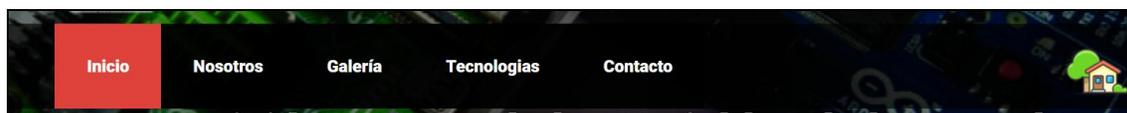


Figura 11. Menú de navegación de la aplicación

La ventana principal es la que mayor cantidad de información tiene, y en la que se tiene los botones de acceso y de contacto, (Figura 12).

Figura 12. Página principal de la aplicación

En la pantalla general, se tiene dos botones que le permiten interacción al usuario con la aplicación. El botón de **Contacto** le permite remitirse a la ventana con los datos del investigador líder, y de la institución para comunicarse por alguna novedad; el botón de **INVESTIGADOR** le permite el inicio de sesión, (Figura 13).



Figura 13. Botones de contacto y de ingreso

El botón de **INVESTIGADOR** genera un formulario de ingreso donde son solicitados el correo y contraseña del usuario permitido, los cuales tienen un texto que solicita los datos respectivamente. En el formulario se tiene tres botones adicionales que permiten la gestión del mismo; el botón **Salir** cierra el formulario, el botón **Ingresar** valida la información, y el botón **Cerrar Sesión**, saca al usuario de los permisos autorizado al ingresar, (Figura 14).

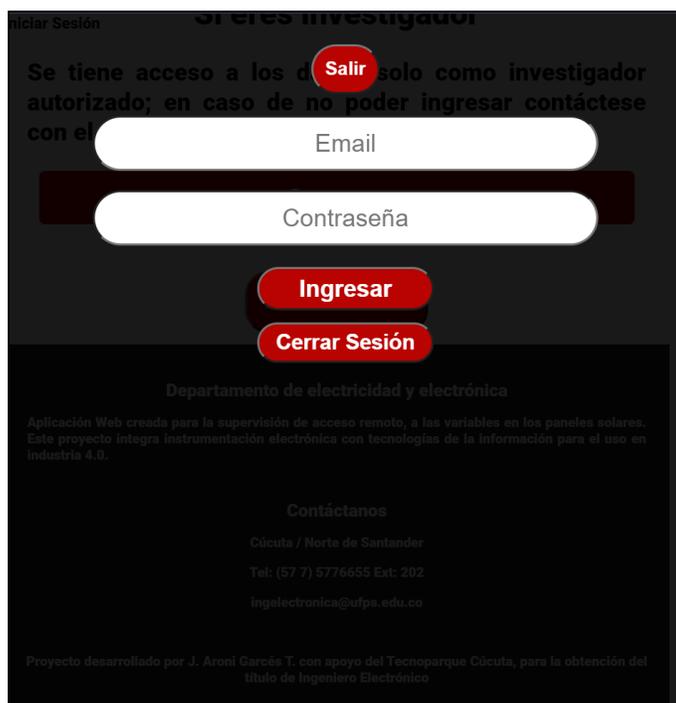


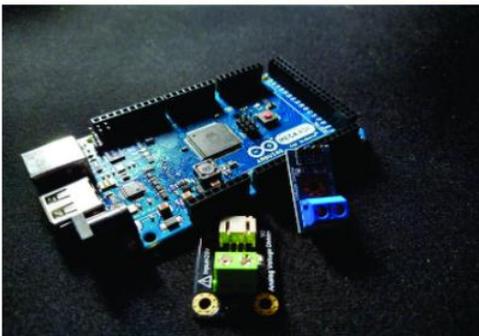
Figura 14. Página de inicio y botones

Al iniciar sesión se tiene acceso a nuevos elementos dentro de la página de inicio, donde se visualiza los datos de los sensores conectados en una tabla, y se genera un botón por cada sensor, o panel supervisado, (Figura 15).

Supervisión Remota de las variables de los
Paneles Solares

Programa de Ingeniería Electrónica

Comprometidos con la región y el desarrollo de energías renovable; esta aplicación monitorea en tiempo real las variables de interés en los sistemas fotovoltaicos.



Si eres investigador

Se tiene acceso a los datos, solo como investigador autorizado; en caso de no poder ingresar contáctese con el investigador líder.

Contacto

INVESTIGADOR

VARIABLES EN LOS PANELES

PANEL 1

Voltaje:	
Corriente :	
Potencia:	
Resistencia:	
Fecha:	
Hora:	

PANEL 2

Voltaje:	
Corriente :	
Potencia:	
Resistencia:	
Fecha:	
Hora:	

PANEL 3

Voltaje:	
Corriente :	
Potencia:	
Resistencia :	
Fecha:	
Hora:	

PANEL 4

Voltaje:	
Corriente :	
Potencia:	
Resistencia :	
Fecha:	
Hora:	

Departamento de electricidad y electrónica
Contáctanos

Figura 15. Acceso a los datos de los paneles

Cada tabla de datos posee su propio botón **Panel #** que permite redirigirse a una ventana dedica a la visualización en tiempo real de los datos.

VARIABLES EN LOS PANELES

PANEL 1

Voltaje:	
Corriente :	
Potencia:	
Resistencia:	
Fecha:	
Hora:	

Figura 16. Botón y tabla de datos del panel

La ventana dedicada, contiene datos del sensor y del panel, para profundizar que, y con que se está midiendo; adicionalmente contiene el botón **HISTORICO**, que permite el acceso a los históricos almacenados en la base de datos, (Figura 17).

Tel: (57 7) 5776655 Ext: 202

Av. Gran Colombia No. 12E-96 Colsag Cúcuta / Norte de Santander

Inicio
Nosotros
Galería
Tecnologías
Contacto

Supervisión Remota de la variable del Panel Sensor N°1

Ningbo Qixin Solar Electrical Appliance 18.28V 0.27A

Sensor Voltaje: FZ0430 0-25V.

Voltaje:	
Corriente:	
Potencia:	
Resistencia:	
Fecha:	
Hora:	

HISTORICO

Figura 17. Ventana dedicada de cada panel

La ventana de históricos, visualiza una lista de los archivos que están disponibles, para la descarga, y cuenta con una caja de texto para escribir el nombre del archivo de interés, el botón para descargar está oculto, y solo es visible hasta que es seleccionado un archivo, (Figura 18).



Figura 18. Ventana de históricos

Al seleccionarse un archivo, presionando el botón **seleccionar**, aparece el botón **DESCARGAR**, que permite descargar el archivo en el dispositivo, (Figura 19).

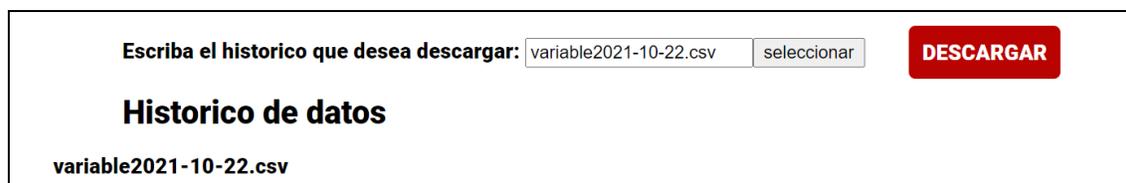


Figura 19. Descarga de históricos

La pestaña **Nosotros** tiene la información del desarrollador y colaboradores del proyecto, (Figura 20).

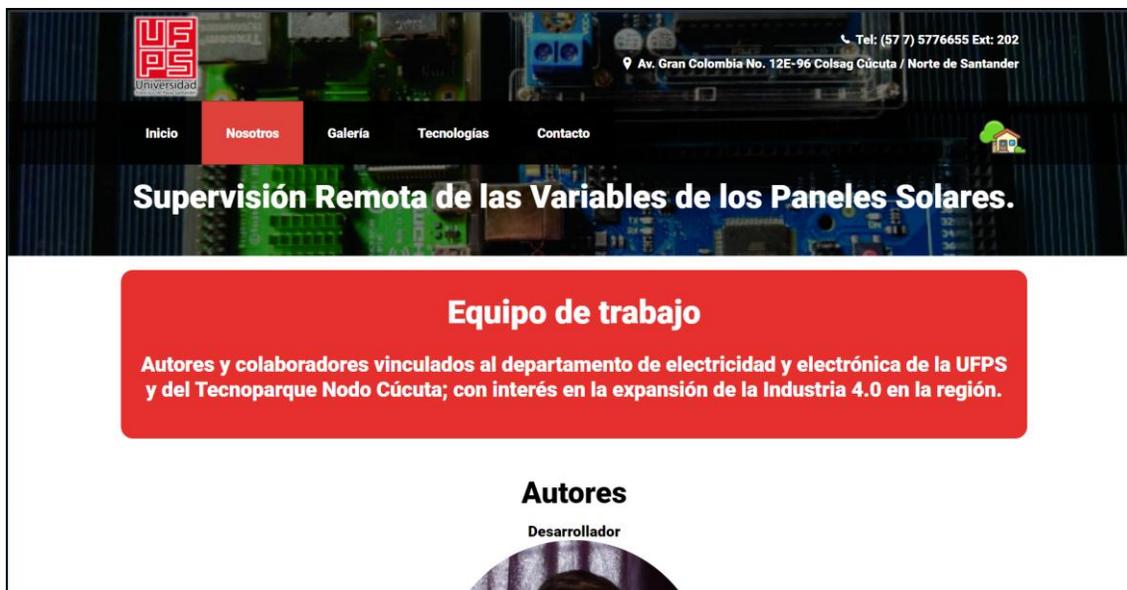


Figura 20. Ventana nosotros

La pestaña **Galería** tiene imágenes del proyecto, y de algunos componentes, (Figura 21).



Figura 21. Ventana galería

Las imágenes pueden ser seleccionadas para verlas a detalle, y aparece un botón **X** de cerrar la imagen, (Figura 22).



Figura 22. Imagen seleccionada

La pestaña **Tecnologías** tiene imágenes de las tecnologías usadas en el proyecto, y también permite la selección de las imágenes, como la ventana de **Galería**, (Figura 23).

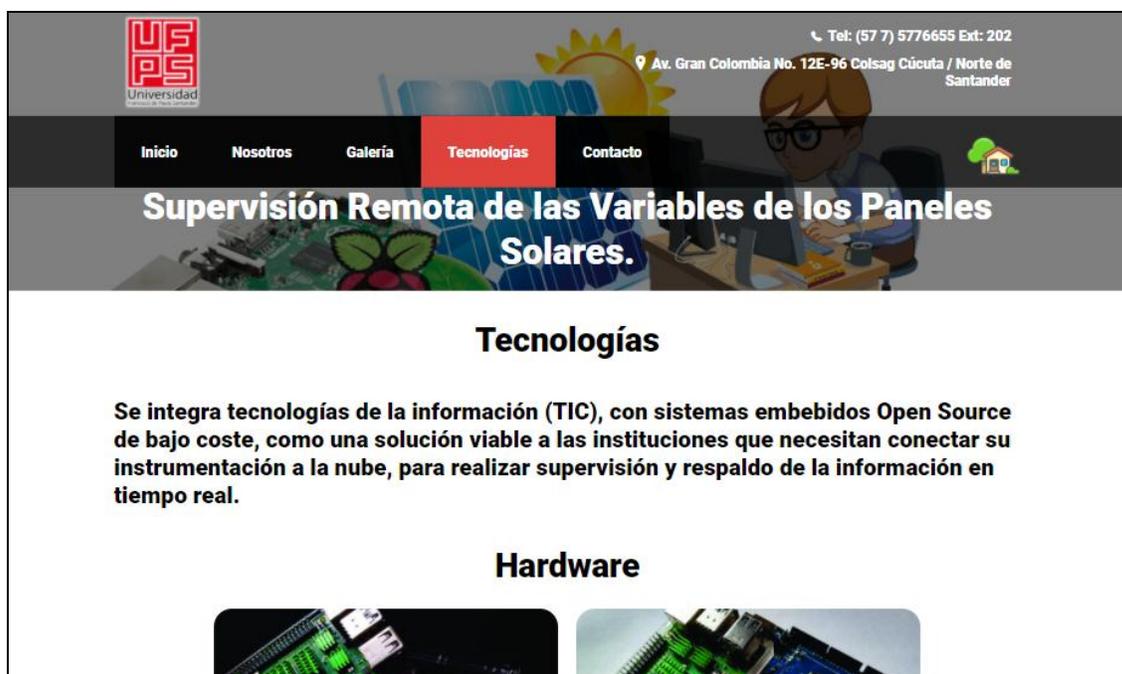


Figura 23. Ventana de tecnologías

La pestaña de **Contacto** tiene dos formas para ser accedida, ya sea por el menú, o por el botón de la pantalla de inicio, (Figura 24).

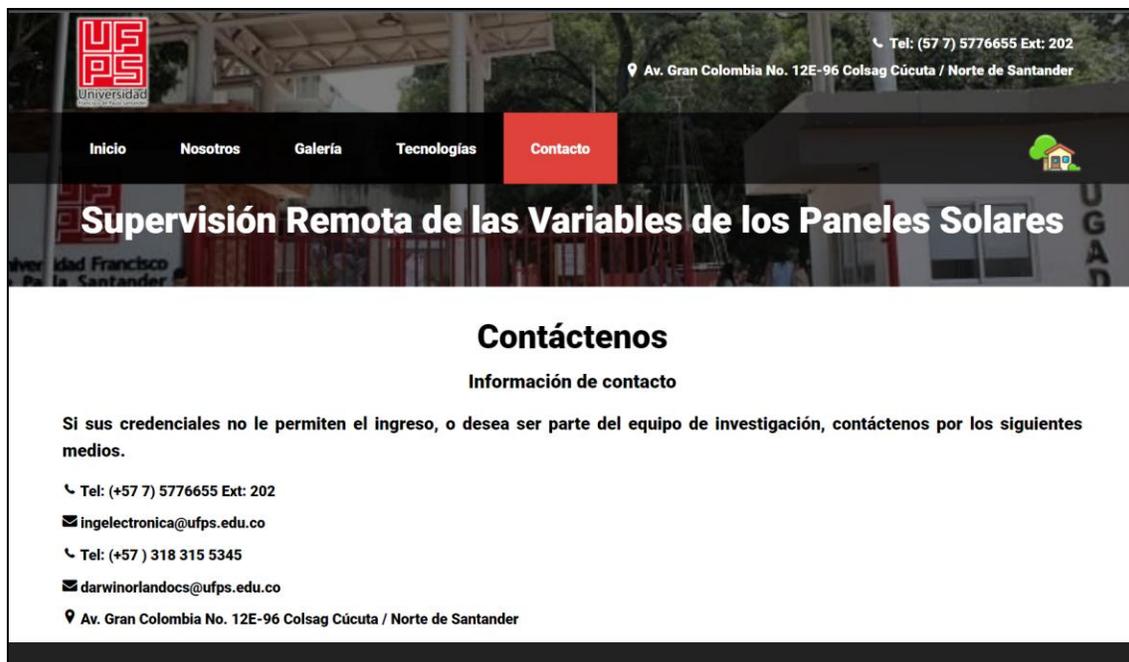


Figura 24. Ventana contacto

Lo único que difiere con formato móvil, es la barra de menú compacta, que tiene un botón de menú, que se despliega cuando es seleccionado, (Figura 25).



Figura 25. Menú en formato móvil