

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Diseño e implementación de un sistema de control vía RF y monitoreo vía GSM/GPRS de un robot para el secado de café

Trabajo de graduación presentado por Debora Raquel López Rodríguez para optar al grado académico de Licenciada en Ingeniería Mecatrónica

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



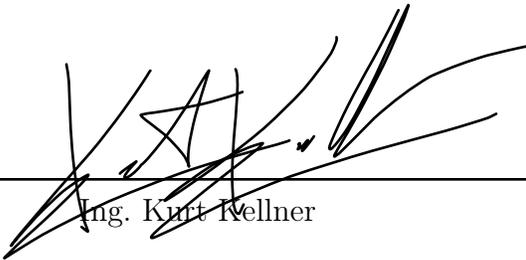
Diseño e implementación de un sistema de control vía RF y monitoreo vía GSM/GPRS de un robot para el secado de café

Trabajo de graduación presentado por Debora Raquel López Rodríguez para optar al grado académico de Licenciada en Ingeniería Mecatrónica

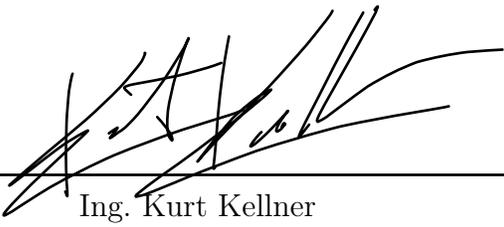
Guatemala,

2022

Vo.Bo.:

(f) 
Ing. Kurt Kellner

Tribunal Examinador:

(f) 
Ing. Kurt Kellner

(f) 
MAEB. Pablo Mazariegos

(f) 
M. Sc. Carlos Esquit

Fecha de aprobación: Guatemala, 21 de junio de 2022.

Este proyecto surge como continuación a la solución a una necesidad por parte de la finca El Injerto. Se vio la oportunidad de incluir herramientas para el monitoreo del proceso de secado al sol de café, logrando así que los caficultores puedan llevar un registro de las velocidades para lograr controlar con mayor precisión el tiempo adecuado para el secado del grano de café, donde el riesgo de daño al producto sea mínimo incrementando la producción y la productividad en el cultivo. Esta tesis no sería posible sin todo el apoyo y confianza por parte de la finca El Injerto.

Además, el diseño electrónico y el diseño CAD del control remoto se ha hecho bajo la asesoría del Ing. Kurt Kellner, quien ha brindado sus conocimientos y experiencia. Así mismo, el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la Universidad del Valle de Guatemala ha dado su apoyo constantemente al proyecto, ofreciendo soluciones para la continuación del proyecto durante la pandemia de COVID-19.

También quiero hacer un reconocimiento especial a mi familia y amigos que me apoyaron anímicamente en todo este proceso, y sin la ayuda de Dios jamás se hubieran alcanzado los buenos resultados de este trabajo.

Prefacio	v
Lista de figuras	XII
Lista de cuadros	XIII
Resumen	xv
Abstract	XVII
1. Introducción	1
2. Antecedentes	3
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	9
6. Marco teórico	11
6.1. Estándar GSM (Groupe Special Mobile)	11
6.2. Estándar GPRS (General Packet Radio Service)	12
6.3. Módulo GSM/GPRS	12
6.4. Manual de normas técnicas para el diseño ergonómico de puestos con pantallas de visualización	13
6.5. Norma DIN 33 402	14
6.6. Tarjeta SIM900 GSM/GPRS	15
6.7. Comandos AT Tarjeta GSM/GPRS	16
6.8. Adafruit IO	18

6.8.1. Feeds	18
6.8.2. Dashboards	19
6.9. Protocolo MQTT	19
6.9.1. Cliente	20
6.9.2. Servidor	20
6.9.3. Conexión	20
6.10. Protocolo HTTP/1.1/	22
6.11. Interfaz de usuario	25
6.11.1. React.js	25
6.11.2. React Native	25
6.11.3. Node.js	25
6.11.4. Firebase	25
6.12. Microcontrolador	26
6.13. Comunicación de radiofrecuencia	27
6.14. Cálculo de batería para el control remoto	27
7. Shield GSM/GPRS SIM900	29
7.1. Implementación	29
7.1.1. Hardware	29
7.1.2. Software	30
7.2. Pruebas y resultados	34
7.2.1. Comunicación Arduino Due -> Shield SIM900 -> Adafruit IO	34
8. Diseño y arquitectura del Back-end	37
8.1. Firebase	37
8.1.1. Autenticación de los usuarios	37
8.1.2. Hosting	39
8.2. Servidor MQTT	39
8.2.1. Cliente Adafruit IO	40
8.2.2. Cliente base de datos en MYSQL	41
8.2.3. Cliente aplicación móvil	42
8.3. Pruebas y resultados	42
8.3.1. Firebase	42
8.3.2. Servidor MQTT	42
9. Diseño y arquitectura del Front-end	45
9.1. Aplicación móvil	45
9.1.1. Prototipo final	45
9.2. Dashboard/aplicación web	50
9.2.1. Implementación del módulo de autenticación	51
9.2.2. Prototipo final de la pantalla principal	52
10. Control remoto	53
10.1. Funcionamiento	53
10.2. Implementación	54
10.2.1. Hardware	54
10.2.2. Módulo NRF24L01	55

10.3. Diseño asistido por computadora para el control remoto físico	56
10.3.1. Estudio ergonómico	56
10.4. Pruebas y resultados	58
10.4.1. <i>Hardware</i>	58
10.4.2. Diseño asistido por computadora para el control remoto físico	59
10.4.3. Módulo NRF24L01	66
11. Listado de materiales electrónicos completo y servicios	67
12. Conclusiones	71
13. Recomendaciones	73
14. Bibliografía	75
15. Anexos	77
15.1. Esquemáticos de los circuitos	77
15.1.1. Circuito del control remoto	77
15.2. PCBs	78
15.2.1. PCB control remoto	78
15.3. Mockups	79
15.3.1. Aplicación móvil	79
15.3.2. Aplicación web	80
16. Glosario	81

1. Arquitectura de red del sistema GSM.	11
2. Arquitectura de red del sistema GPRS.	12
3. Comparación módulos GSM/GPRS.	13
4. Medidas de los dedos según Norma DIN 33402 (Parte 1) [7]	14
5. Medidas de los dedos según Norma DIN 33402 (Parte 2) [7]	15
6. Descripción física de la tarjeta SIM900 GSM/GPRS.	16
7. Ejemplo de feed	18
8. Ejemplo de Dashboards	19
9. Codificación de datos en MQTT	20
10. Inicio de la conexión MQTT entre cliente y servidor.	21
11. Contenido del mensaje CONNACK.	22
12. Código del estado de conexión en el mensaje CONNACK.	22
13. Diagrama de secuencia de una interacción simple entre un cliente y un servidor.	23
14. Arduino DUE.	26
15. NRF24L01.	27
16. Constante de peukert para una batería de litio [19]	28
17. Serial selected	29
18. Software trigger	30
19. Consumo del shield SIM900 [20]	30
20. Esquemático SIM900 Y Arduino Due	31
21. Solicitud HTTP	31
22. Create Data Adafruit IO	33
23. Petición HTTP	33
24. Flujograma GPRS	34
25. Prueba de conexión TCP	35
26. Prueba de petición HTTP 1.1	35
27. Respuesta de la petición HTTP 1.1	36
28. Dashboard de Adafruit IO	36
29. Vista de los métodos de autenticación	38

30. Código del SDK de Firebase	38
31. Interfaz de HiveMQ MQTT Dashboard	39
32. Arquitectura del servido mqtt	40
33. Feeds del sistema	40
34. Feeds asignados de cada cliente	41
35. Estructura de la base de datos MYSQL	41
36. Prueba de los métodos de autenticación	42
37. Prueba del cliente adafruit io	43
38. Prueba del servidor MQTT/clientes	43
39. Prueba del cliente MYSQL	43
40. Prueba del cliente aplicacion móvil	44
41. Autenticación de la aplicación móvil	46
42. Registro de la aplicación móvil	47
43. Menú de la aplicación móvil	48
44. Dashboard de la aplicación móvil	49
45. WorkScreen de la aplicación móvil	50
46. Inicio de sesión	51
47. Flujo de registro de un nuevo usuario	52
48. Prototipo de Dashboard	52
49. CAD del control remoto	53
50. Diagrama de bloques control remoto	54
51. Prueba de funcionamiento de la pantalla Oled.	58
52. Vista frontal para la carcasa	59
53. Vista trasera para la carcasa	59
54. Vista frontal de riel para joystick	60
55. Vista trasera de riel para joystick	60
56. Vista frontal para la carcasa	61
57. Vista trasera para la carcasa	61
58. Vista frontal de riel para joystick 1	62
59. Vista frontal de riel para joystick 2	62
60. Vista frontal para la carcasa	63
61. Vista trasera para la carcasa 1	63
62. Vista trasera para la carcasa 2	64
63. Vista trasera para la carcasa 3	64
64. Vista frontal de riel para joystick	65
65. Riel ensamblado en el control remoto	65
66. Prueba de transmisión y recepción	66
67. Esquemático circuito control remoto	77
68. PCB circuito control remoto	78
69. Representación 3D del circuito control remoto	78
70. Mockup de aplicación móvil	79
71. Mockup de la aplicación web	80

Lista de cuadros

1. Comandos AT para GSM/GPRS	17
2. Comandos TCP	32
3. Comandos APN	32
4. Bytes enviados por el control remoto	55
5. Bytes enviados por el robot móvil	56
6. BOM completo del control remoto y del sistema de recopilación de datos GSM/GPRS	68
7. Servicio de Authentication y Hosting de Firebase	69

El presente proyecto se enfoca en dotar de herramientas al diseño original del robot móvil para lograr monitorear el tiempo y velocidad adecuados para el secado del grano de café, basados en el supuesto de que en la finca no se dispone de una red de internet inalámbrica.

Se realizó como primera propuesta la implementación del módulo GSM/GPRS que cuenta con un chipset de operación SIM900 de SIMCom Wireless Solutions. Este módulo ofrece una solución completa para transmisión inalámbrica ya que posee una configuración cuatri-banda que puede trabajar en las frecuencias GSM/GPRS 850/900/1800/1900 MHz para transmitir y recibir voz, SMS y datos. Con el cual se obtuvieron resultados satisfactorios en la comunicación UART con el Arduino Due.

Además de la implementación de Adafruit IO como servicio de nube principal, la cual proporciona la recepción de la información en tiempo real proveniente del robot móvil, del que se obtuvieron resultados satisfactorios. Se realizó una Interfaz web por medio de react.js y una interfaz móvil por medio de react native, brindando una autenticación de usuario al utilizar firebase como host de nuestras aplicaciones con las funciones de desarrollador que nos ofrece la plataforma.

Por último, se realizó el diseño y el circuito del control remoto, que es la forma en que el usuario puede tomar control del robot móvil cuando no desea que trabaje de forma autónoma.

The project in question focuses on providing tools for the original design of the mobile robot, this is done to monitor the proper time and speed for the coffee's drying process based on the assumption that there is no internet access on the locale.

Firstly, implementing the use of a module GSM/GPRS was proposed. This module contains an operational chipset SIM900 from SIMCom Wireless Solutions. The GSM module offers a complete solution for wireless transmission due to it allowing for a 4-band configuration which is able to use frequencies GSM/GPRS 850/900/1800/1900 MHz used for transmitting and receiving voice, SMS and other data. With this, obtaining satisfactory results was possible with the use of UART communication with the DUE Arduino.

Apart from the implementation of Adafruit IO as the main cloud service, providing the tools to receive real time information from the mobile robot from which, very positive results were gathered. A web interface was created with the use of react.js and a mobile interface with react native, providing, in return, a user authentication when firebase was used as host for our applications with the developer functions given by the platform.

Lastly, a design and the circuit for the remote control were made. With the use of this controller, the user gets control of the mobile robot whenever that manual control is needed.

El presente proyecto se enfoca en el diseño de un sistema de monitorización y el diseño asistido por computadora del control remoto de un robot móvil autónomo para el secado de café guatemalteco de calidad. Surge por la falta de innovación y uso de tecnología en el proceso de producción de café guatemalteco; específicamente el proceso de secado al sol. Para este se debe hacer mover el café constantemente para que los rayos del sol sequen todos los granos equitativamente. Se expondrán el conjunto de elementos hardware y software que serán necesarios, tanto para manipular determinadas acciones del robot móvil, como acceder a ciertos datos de forma remota empleando la red GSM/GPRS.

Para la implementación del prototipo y por las características que presenta la plataforma Arduino se seleccionó la placa Arduino DUE, siendo la más adecuada para la conexión del sistema de control vía RF y monitoreo vía GSM/GPRS. En conjunto con la placa de Arduino Due se estará usando el Shield GSM/GPRS SIM900, debido al soporte de los principales protocolos de comunicación que estaremos utilizando a lo largo del desarrollo del proyecto y de su compatibilidad con todo tipo de plataformas de hardware libre, como lo es Arduino, donde estaremos aprovechando la comunicación UART como principal vía de comunicación serial con el Shield GSM/GPRS SIM900.

Con esta aplicación se pretende monitorear en tiempo real la velocidad y el tiempo del secado del grano de café, los cuales serán desplegados en una interfaz web y aplicación móvil, esto se logrará gracias a la implementación del servicio de nube proporcionada por Adafruit. Con Adafruit IO estaremos recibiendo la información en tiempo real a través del modelo de comunicación TCP/IP proporcionado por la shield, además debido a las características de Adafruit IO también tendremos acceso a información adicional como los llamados metadatos que nos indican cosas como fecha y hora de transmisión de datos, coordenadas GPS del emisor, etc. Esto con el fin de poder recolectar los datos enviados y poder desplegarlos finalmente en la página

web/aplicación móvil.

Con todas las características que se mencionaron anteriormente, se pretende dotar al robot móvil con herramientas para automatizar el proceso de secado al sol para lograr controlar con mayor precisión el tiempo adecuado para el secado del grano de café, incrementando la producción y la productividad en el cultivo.

Por último, se muestra el Bill of Materials completo de todos los componentes electrónicos que conforman el robot móvil. Este incluye el precio de cada componente y el precio final.

La utilización de robots móviles en la agricultura es algo que se ha estado estudiando por varias universidades a nivel internacional. Las universidades europeas son de las pioneras en esta área. Cada año se realiza en Alemania una competencia de robots móviles de agricultura, en dónde participan universidades o instituciones y es patrocinado por las empresas más grandes de tecnología y agricultura de Europa. Para conocer más de esta competencia puede visitar [\[1\]](#).

Por otro lado, Los sistemas de control y monitoreo remoto han sido implementados paulatinamente con características de conectividad, directamente con acceso a Internet y últimamente en las redes de telefonía celular, en particular mediante los servicios de transmisión de datos que sobre ella se ejecutan, debido a las ventajas que ofrece en características como alcance, velocidad y movilidad. Un ejemplo con resultados satisfactorios es el robot móvil Giraa 02 desarrollado en la Universidad de Antioquia, el cual describe un sistema que permite teleoperar un robot a través de teléfonos celulares. Se emplea la red de telefonía celular existente en el país, utilizando el Servicio General de Paquetes por Radiofrecuencia GPRS. De esta manera se permite a los usuarios del sistema un rango más amplio de movilidad. La implementación de dichas aplicaciones se hace mediante el lenguaje de programación JAVA [\[2\]](#).

El desarrollo del presente trabajo atiende una gran problemática que tienen actualmente muchos caficultores en el país que les genera una baja productividad por hectárea. El proceso de secado al sol del café no es eficiente; si se realiza un rápido secado en el grano del café, este no resulta ser de una buena calidad y si se realiza un secado cauteloso cuidando las propiedades inmersas en el grano, en consecuencia, el proceso resulta ser demasiado lento [3].

Según diversos estudios, el proceso de secado de café es más complicado que el de cualquier otro grano, ya que este después de ser lavado y escurrido contiene alrededor de 55 % de humedad y es en el proceso de secado que esta humedad debe ser reducida entre el 10 % y el 12 %, debido a que es la norma vigente de la comercialización del café pergamino seco. Se debe tener mucho cuidado al secar el café, puesto que durante este proceso puede ocurrir pérdidas aromáticas del grano, si se emplean altas temperaturas, además puede causar deformaciones en el aspecto físico, si no se tiene buenas condiciones de operación [3].

Con esta aplicación se pretende monitorear en tiempo real la velocidad y el tiempo del secado del grano de café, permitiendo que los caficultores puedan llevar un registro de las velocidades, sin que existan riesgos de daño del producto lo cual no ocasionara pérdidas económicas. Luego de la implementación se pretende controlar con mayor precisión el tiempo adecuado para el secado del grano de café, incrementando la producción y la productividad en el cultivo y además que se garantice la calidad del grano para su venta.

4.1. Objetivo general

Diseñar e implementar un control remoto vía RF y un sistema de recopilación y despliegue de datos vía GSM/GPRS.

4.2. Objetivos específicos

- Diseñar e implementar un sistema de recopilación de datos y monitoreo del robot de café utilizando tecnología GSM/GPRS.
- Diseñar e implementar una interfaz web y aplicación móvil que desplieguen los datos y estadísticas del robot.
- Diseñar y fabricar un sistema de control remoto para el robot.
- Realizar un listado de costos para la implementación del control remoto y el sistema de recopilación de datos.

El trabajo realizado en este proyecto consistió en el diseño e implantación, hardware y software de los sistemas de comunicación mencionados en los objetivos, además de sus respectivas pruebas en simulaciones físicas y realizar la fabricación de sus respectivos PCBs. Debido a la problemática relacionada con la pandemia COVID-19 las pruebas donde se incorporen todos los sistemas de comunicación, tanto los del control remoto como los del robot móvil, se realizarán en el campus de la universidad.

Debido al supuesto de la ausencia de internet inalámbrico en la finca se implementó la tecnología GPRS que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de paquetes, a la que se denomina hoy en día como “paquete de datos”. Dichos paquetes de datos serán procesados por Adafruit IO, donde estaremos recibiendo la información en tiempo real a través de la shield GSM/GPRS SIM900. Dicho moduló fue probado con el microcontrolador y los resultados se muestran en este documento.

Por otra parte, se desarrolló una segunda propuesta, tanto hardware y software como para el diseño CAD, del control remoto del robot móvil.

6.1. Estándar GSM (Groupe Special Mobile)

El sistema global para las comunicaciones móviles (GSM), es un sistema estándar libre de regalías, de telefonía móvil digital.

Un cliente GSM puede conectarse a través de su teléfono con su computador y enviar y recibir mensajes por correo electrónico, faxes, navegar por Internet, acceder con seguridad a la red informática de una compañía (red local/Intranet), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el servicio de mensajes cortos (SMS) o mensajes de texto. GSM se considera, por su velocidad de transmisión y otras características, un estándar de segunda generación (2G). Su extensión a 3G se denomina UMTS y difiere en su mayor velocidad de transmisión, el uso de una arquitectura de red ligeramente y sobre todo en el empleo de diferentes protocolos de radio (W-CDMA) [4].

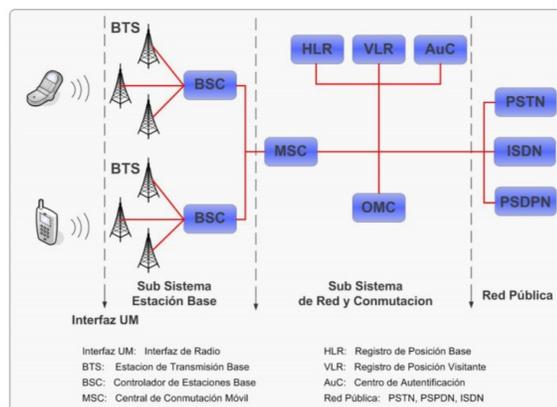


Figura 1: Arquitectura de red del sistema GSM.

6.2. Estándar GPRS (General Packet Radio Service)

Se fundamenta en la transmisión de paquetes y donde los canales de comunicación se comparten entre los distintos usuarios de forma dinámica. GPRS coexiste con GSM, compartiendo gran parte de la infraestructura desplegada en el mismo, ofreciendo un servicio portador más eficiente para las comunicaciones de datos, especialmente en el caso de los servicios de acceso a redes IP como Internet [4].

La velocidad teórica máxima que puede alcanzar GPRS es de 171,2 kbit/s, este tipo de transmisión permite una mayor capacidad y velocidad complementando los servicios existentes tales como conexiones de teléfonos celulares con conmutación de circuitos y el servicio de mensajes cortos (SMS). A medida que se ha ampliado la disponibilidad de GPRS, junto con otros servicios 2.5G y 3G, los usuarios móviles de redes privadas virtuales (VPNs) han podido acceder a la red privada de forma continua por medio de la red inalámbrica en lugar de una conexión de acceso telefónico. [4].

Esta tecnología permite conseguir en condiciones óptimas rendimientos de hasta 5 veces la velocidad máxima de GSM. Así es posible desarrollar múltiples servicios con un mayor y mejor contenido de imágenes, sonido y video. Además, posibilita en el futuro la creación de otras aplicaciones en base a esta plataforma. [4].

GPRS también complementa Bluetooth, un estándar para reemplazar conexiones cableadas entre dispositivos con conexiones de radio inalámbricas. Además del Protocolo de Internet (IP), GPRS soporta X.25, un protocolo basado en paquetes que se utiliza principalmente en Europa. GPRS es un paso evolutivo hacia el Entorno GSM Mejorado de Datos (EDGE) y el Servicio Telefónico Móvil Universal (UMTS) [4].

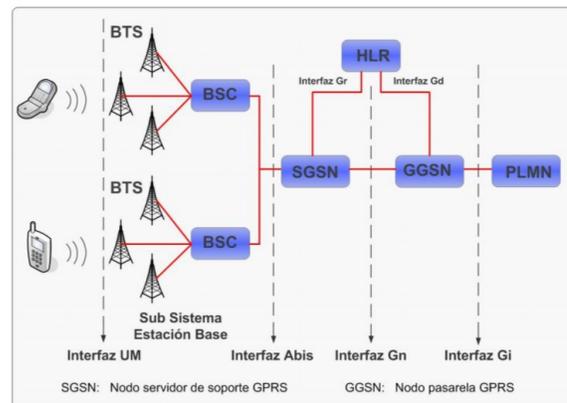


Figura 2: Arquitectura de red del sistema GPRS.

6.3. Módulo GSM/GPRS

En la Figura 3 se muestra un cuadro comparativo de diversos dispositivos con el fin de seleccionar el más adecuado para el desarrollo del proyecto, tomando en cuenta

que se requiere enviar datos hacia una página de monitoreo en Internet [5].

Modelo	Shield GSM/GPRS SIM900	Shield GSM M95	GSM Shield	Shield M2M	ICOMSAT
Fabricante	SIMCOM	QUECTEL	Arduino	MCI Electrònics	Itead Studio
Microprocesador	Sim900	FT232	Quectel M66	Quectel M10	Sim900
Banda	Quad-Band	Quad-Band	Quad-Band	Quad-Band	Quad-Band
Frecuencia	850, 900, 1800 y 1900 MHz	850, 900, 1800 y 1900 MHz			
Velocidad uplink/downlink	42.8/85.6 Kbps	42.8/85.6 Kbps	85.6 Kbps	85.6 Kbps	42.8/85.66 Kbps
Protocolos soportados	TCP/HTTP, FTP	TCP/UDP/PPP/FF TP/HTTP	TCP/UDP, HTTP	TCP/UDP/PPP/FTP/SMTP/HTTP/SSL/CMUX	TCP/HTTP, FTP
Funciones	SMS, voz y datos	SMS, voz, fax y datos	SMS, llamadas y datos	SMS, voz y datos	SMS, voz y datos
Alimentaci3n	4.8 – 5.2 VDC	5 VDC	5 VDC	5 VDC	4.5 – 5.5 VDC
Corriente	50 – 450 mA	1.2 – 1.3 mA	700 – 1000 mA	40 – 1800 mA	500 – 200 mA
Puerto de Antena	Si	Si	No	Si	Si
Leds indicadores	Status, Net y PWR	Status, Net y PWR	On, Status y Net	PWR Y Net	Status, Net y PWR
Puerto de Micr3fono y Altavoz	Si	Si	No	No	Si
Dimensiones	110 mm x 69 mm	55 mm x 54 mm	101.6 mm x 76.2 mm	68.6 mm x 53.3 mm	77.2 mm x 66 mm
Costo	\$ 45.00	\$ 55.00	\$ 30.00	\$20.00	\$ 50.00

Figura 3: Comparaci3n m3dulos GSM/GPRS.

6.4. Manual de normas t3cnicas para el dise1o ergon3mico de puestos con pantallas de visualizaci3n

El objetivo de este manual es dar a conocer los principales aspectos de las normas t3cnicas de 1mbito europeo e internacional referidas al dise1o ergon3mico de puestos que utilizan equipos con pantallas de visualizaci3n de datos (PVD). El manual se basa, principalmente, en el contenido de las normas t3cnicas desarrolladas en el seno del Comit3 Europeo de Normalizaci3n (CEN) en colaboraci3n con ISO, concretamente en las normas ISO-9241 y EN-ISO 9241 [6].

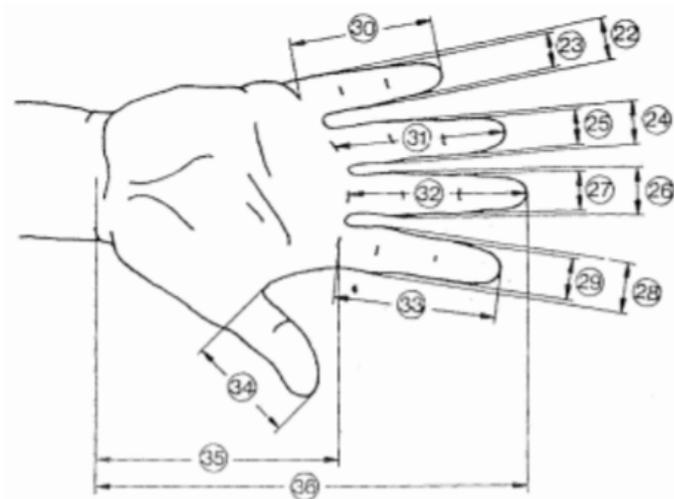
Estas normas establecen los requisitos ergon3micos para equipos de PVD's empleados en actividades de oficina, con objeto de asegurar que los usuarios puedan desarrollar sus actividades de manera segura, eficiente y confortable. Los destinatarios son los diversos agentes implicados en el dise1o, fabricaci3n, adquisici3n y uso de los equipos de PVD's, as3 como los responsables de dirigir y supervisar las actividades realizadas con ellos. Si bien una parte importante de su contenido est1 dedicada al

diseño de los equipos de PVD's, también se abordan los aspectos relativos al diseño físico del puesto, al medio ambiente físico y a la gestión y organización del trabajo con estos equipos [6].

6.5. Norma DIN 33 402

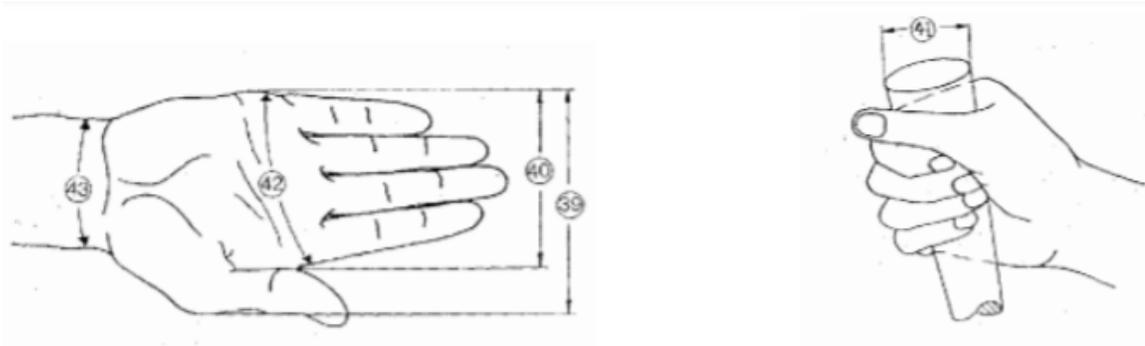
Las dimensiones de las manos tienen una gran importancia para el diseño de herramientas, utillaje y mandos. Ante tal necesidad se estableció la norma DIN 33 402 con el fin de estandarizar estas medidas y poder ser utilizadas en la industria [7].

A continuación se muestran las medidas para las manos que esta norma dicta.



Dimensiones En cm.		PERCENTIL					
		Hombres			Mujeres		
		5 %	50 %	95 %	5 %	50 %	95 %
22	Ancho del meñique en la palma de la mano	1,8	1,7	1,8	1,2	1,5	1,7
23	Ancho del meñique próximo de la yema	1,4	1,5	1,7	1,1	1,3	1,5
24	Ancho del dedo anular en la palma de la mano	1,8	2,0	2,1	1,5	1,6	1,8
25	Ancho del dedeo anular próximo a la yema	1,5	1,7	1,9	1,3	1,4	1,6
26	Ancho del dedo mayor en la palma de la mano	1,9	2,1	2,3	1,6	1,8	2,0
27	Ancho del dedo mayor próximo a la yema	1,7	1,8	2,0	1,4	1,5	1,7
28	Ancho del dedo índice en la palma de la mano	1,9	2,1	2,3	1,6	1,8	2,0
29	Ancho del dedo índice próximo a la yema	1,7	1,8	2,0	1,3	1,5	1,7
30	Largo del dedo meñique	5,6	6,2	7,0	5,2	5,8	6,6
31	Largo del dedo anular	7,0	7,7	8,6	6,5	7,3	8,0
32	Largo del dedo mayor	7,5	8,3	9,2	6,9	7,7	8,5
33	Largo del dedo índice	6,8	7,5	8,3	6,2	6,9	7,6
34	Largo del dedo pulgar	6,0	6,7	7,6	5,2	6,0	6,9
35	Largo de la palma de la mano	10,1	10,9	11,7	9,1	10,0	10,8
36	Largo total de la mano	17,0	18,6	20,1	15,9	17,4	19,0

Figura 4: Medidas de los dedos según Norma DIN 33402 (Parte 1) [7]



Dimensiones	PERCENTIL					
	Hombres			Mujeres		
	5 %	50 %	95 %	5 %	50 %	95 %
37 Ancho del dedo pulgar	2,0	2,3	2,5	1,6	1,9	2,1
38 Grosor de la mano	2,4	2,8	3,2	2,1	2,6	3,1

Dimensiones	PERCENTIL					
	Hombres			Mujeres		
	5 %	50 %	95 %	5 %	50 %	95 %
39 Ancho de la mano incluyendo dedo pulgar	9,8	10,7	11,6	8,2	9,2	10,1
40 Ancho de la mano excluyendo el dedo pulgar	7,8	8,5	9,3	7,2	8,0	8,5
41 Diámetro de agarre de la mano*	11,9	13,8	15,4	10,8	13,0	15,7
42 Perímetro de la mano	19,5	21,0	22,9	17,6	19,2	20,7
43 Perímetro de la articulación de la muñeca	16,1	17,6	18,9	14,6	16,0	17,7

* Las medidas corresponden al anillo descrito por los dedos pulgar e índice

Figura 5: Medidas de los dedos según Norma DIN 33402 (Parte 2) [7]

6.6. Tarjeta SIM900 GSM/GPRS

En función de las características de los módulos GSM/GPRS mencionados anteriormente, se ha escogido el Shield GSM/GPRS SIM900, en función de sus datos

técnicos, ya que da soporte a los principales protocolos de transmisión de datos, en la Figura 6 se muestra el modelo físico de este módulo.

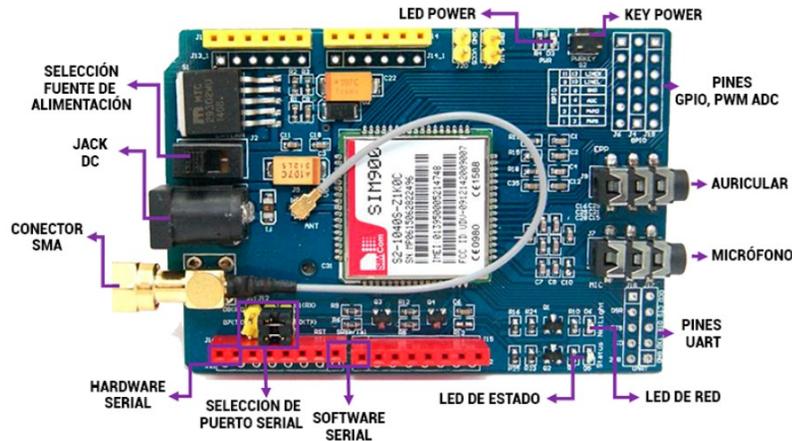


Figura 6: Descripción física de la tarjeta SIM900 GSM/GPRS.

Esta tarjeta se conecta a la red de telefonía celular para la transmisión/recepción de voz y datos, es compatible con todas las plataformas de hardware libre, especialmente con tarjetas Arduino con las cuales comparte el mismo estándar de fabricación. La configuración y el control del módulo se lo realiza mediante comandos AT a través de UART (Universal Asynchronous Receiver-Transmitter) que es un dispositivo integrado en la placa para interactuar en procesos vía puerto serial [8].

6.7. Comandos AT Tarjeta GSM/GPRS

Los comandos AT son nada más que órdenes que un microcontrolador envía en este caso a la Tarjeta GSM/GPRS para que lleve a cabo las funciones que se desea realizar, cada comando ejecuta una acción para lo cual al desarrollar el programa se debe incluir tiempos para que se ejecute cada instrucción en el Shield, y así de forma inversa recibir un mensaje de confirmación, los cuales se divisan en cuestión de diseño a través de la interfaz del puerto serial del IDE Arduino.

El control mediante comandos AT, inicia en el instante en que la tarjeta Arduino esté conectada a la Tarjeta GSM/GPRS, generalmente por puertos seriales debidamente configurados. En la Figura 6 se muestran los comandos AT más utilizados para el control de la Tarjeta GSM/GPRS, que se dividen en tres grupos [5].

COMANDOS AT PARA CONEXIÓN A INTERNET	
COMANDO AT	DESCRIPCIÓN
AT	Para comprobar si el dispositivo acepta comandos AT
AT+CPIN?	Estado del PIN de la tarjeta SIM, en estado Ready está listo para su uso.
AT+CCALR?	Consulta si está activo el servicio de llamadas, debe estar en 1.
AT+CIPSHUT	Cierra la sesión GPRS, Shut OK significa que se ha desconectado
AT+CGATT=0	Desconexión GPRS para liberar recursos utilizados.
AT+CGATT=1	Conectarse a la red GPRS.
AT+CSQ	Para conocer la calidad de señal, si es 13 la calidad es de -87 dBm, que es un nivel correcto.
AT+CREG?	Estado de registro y acceso a la red GPRS cuando el valor es 1, si es 0 se ha conectado a red HSUPA.
AT+CSST?	Consulta de APN de la operadora en uso, si sale CMNET indica que no se ha registrado una APN.
AT+SAPBR=3,1,"CONTYPE",GPRS	Configuración GPRS con la APN.
AT+SAPBR=3,1,"APN", <APN>	Configuración GPRS con la APN.
AT+SAPBR=3,1,"USER",""	Consulta del nombre de usuario.
AT+SAPBR=3,1,"PWD",""	Consulta de contraseña.
AT+SAPBR=1,1	Indica si se ha enlazado a la red GPRS, debe mostrar OK.
AT+CSST="<APN>","",""	Confirmación de aceptación de los datos.
AT+CIPSRIP=1	Habilita al dispositivo para tener dirección IP así como un puerto de enlace de datos.
AT+CIICR	Registro de red.
AT+CIFSR	Muestra la dirección IP local.
COMANDO PARA INICIAR UNA SESIÓN TCP	
AT+CIPSTART="TCP", "dirección IP", "PORT"	Estableciendo la conexión con el sitio remoto.
COMANDOS PARA SUBIR DATOS A INTERNET POR SESIÓN TCP	
AT+CIPSEND= "Texto"	Envía datos a través de una conexión TCP.El modulo nos contesta SEND OK al realizar la operación
AT+CIPCLOSE	Cierra la transmisión.

Cuadro 1: Comandos AT para GSM/GPRS.

6.8. Adafruit IO

Adafruit IO (AIO) es un servicio en la nube desarrollado por la empresa electrónica Adafruit. Un servicio en nube es una plataforma que se pone a disposición del usuario a través de Internet con la finalidad de facilitar el flujo de datos a través de este. Principalmente se utiliza para el almacenamiento y la recuperación de información, pero también tiene otras funcionalidades, como pueden ser:

- Monitorización de datos en tiempo real.
- Control de elementos en tiempo real a través de Internet (controlar luces, motores, leer sensores, etc.)
- Conectar proyectos a otros servicios web (meteorología, redes sociales como Twitter, etc.)
- Enlazar proyectos con otros dispositivos con acceso a Internet.

9

En esta plataforma existen dos elementos fundamentales: feeds y dashboards.

6.8.1. Feeds

Estos elementos son la pieza principal de la plataforma. Contienen tanto los datos recibidos como los metadatos acerca de estos, como la fecha y hora a la que llegaron los datos o las coordenadas GPS de donde vinieron. Los feeds pueden ser públicos o privados, de manera que es posible compartir la información con diferentes usuarios

9.

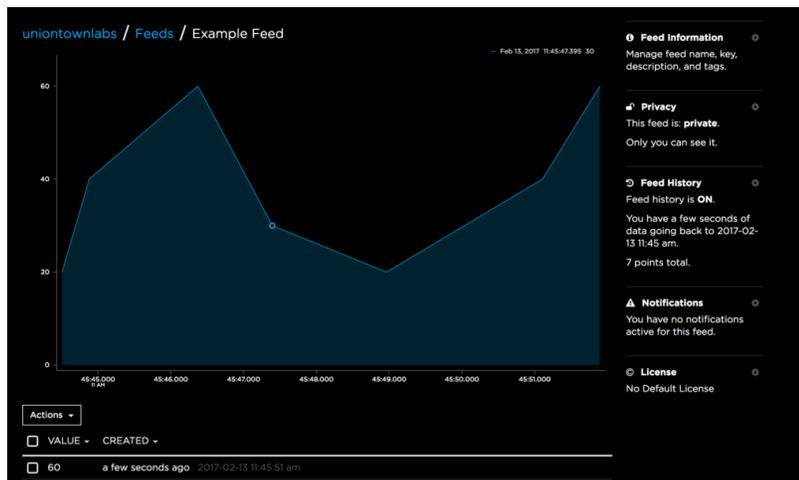


Figura 7: Ejemplo de feed

6.8.2. Dashboards

Actúan de interfaz, permitiendo visualizar los datos recibidos en los feeds y controlar el dispositivo conectado a la plataforma. Disponen de funcionalidades como representaciones gráficas, interruptores, botones, barras de deslizamiento, etc. En esta plataforma, el protocolo de transferencia de datos utilizado es MQTT y HTTP [9].



Figura 8: Ejemplo de Dashboards

6.9. Protocolo MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de transporte de datos Cliente/Servidor. Es considerado como un protocolo ligero y eficiente en cuanto al ancho de banda debido a la estructura de sus paquetes. El mensaje, una vez codificado, tiene la siguiente estructura:

- **Encabezado de paquetes (header):** Se trata de un conjunto de bytes que contienen información acerca del mensaje (origen, destino, tipo de instrucción, etc.). Cada mensaje tiene 2 bytes fijos, pero puede tener un segundo encabezado variable con n bytes.
- **Carga útil (payload):** Se trata del conjunto de bytes en los que se encuentra el mensaje almacenado, excluyendo cualquier tipo de metadato presente en el encabezado. En este protocolo, la carga útil está limitada a 256 bytes de información.

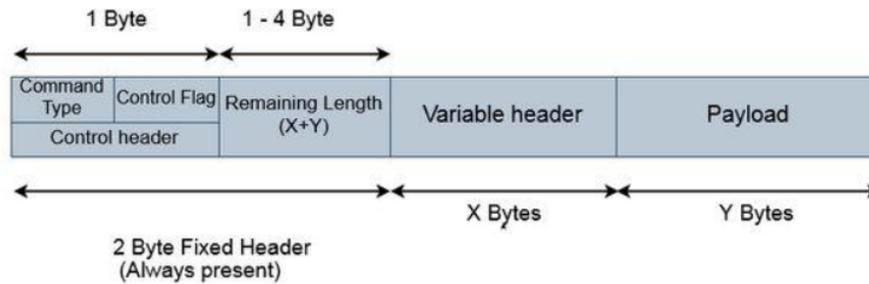


Figura 9: Codificación de datos en MQTT

6.9.1. Cliente

Un cliente MQTT es cualquier dispositivo (desde un microcontrolador hasta un servidor) que ejecuta una librería MQTT y se conecta a un servidor o bróker de este tipo a través de una red. Básicamente, cualquier dispositivo que pueda utilizar una conexión TCP/IP es considerado un cliente [10].

De entre todos los clientes, hemos de diferenciarlos en dos tipos: los publishers y los subscribers. Los primeros son los clientes encargados de enviar el mensaje, mientras que los segundos son los que reciben el mensaje. En nuestro caso, el Arduino sería el publisher, mientras que el ordenador o el móvil en el que vemos los datos recibidos sería el subscriber. La conexión entre ellos es manejada por el bróker, que es el encargado de filtrar los mensajes recibidos y distribuirlos correctamente al segundo tipo de clientes. Tanto el servidor como el tipo de conexión se verán a continuación con más detalle [10].

6.9.2. Servidor

Por otro lado, el servidor o bróker es la contraparte del cliente. Si el cliente es el encargado de enviar los mensajes o recibirlos, el servidor es el responsable de recibir y filtrar todos estos mensajes, determinar qué cliente está suscrito a cada mensaje y mandar los mensajes a dichos clientes. Dependiendo de la implementación, un bróker puede llegar a encargarse de una cantidad significativa de clientes [10].

El bróker es por lo tanto un eje central del protocolo que además ha de encargarse de la identificación y la autorización de los clientes.

6.9.3. Conexión

Como se ha mencionado, MQTT basa su conexión en el modelo TCP/IP, tanto el cliente como el servidor necesitan soportar este protocolo de red.

El modelo TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet) es un protocolo encargado del enlace de datos y que se utiliza en Internet. Posee cuatro capas, en las que están divididas las tareas de comunicación. De forma resumida, estas son las funciones de cada capa:

- **Capa de aplicación:** Es la capa que está más cerca del usuario y es la encargada de interactuar con el software para implementar una comunicación.
- **Capa de transporte:** Determina cuántos datos se deben enviar, dónde y a qué velocidad. Ayuda a garantizar la entrega de datos sin errores.
- **Capa de Internet:** También conocida como capa de red, es la responsable de la transmisión de datos a través de Internet, como su nombre indica, utilizando la ruta óptima.
- **Capa de acceso a la red:** Responsable de definir los detalles de cómo enviar los paquetes a través de la red. Se encarga de la asignación de las direcciones IP, que son una combinación de números que identifican la interfaz de red de un dispositivo que utilice Internet.

10

La conexión MQTT siempre se realiza entre el bróker y un cliente, de manera que no hay interacción directa entre clientes. Para iniciar la conexión, el cliente le envía un mensaje CONNECT (solicitud de conexión al servidor) al bróker, y este responde con un mensaje CONNACK (Connection Acknowledged, mensaje recibido) 10.

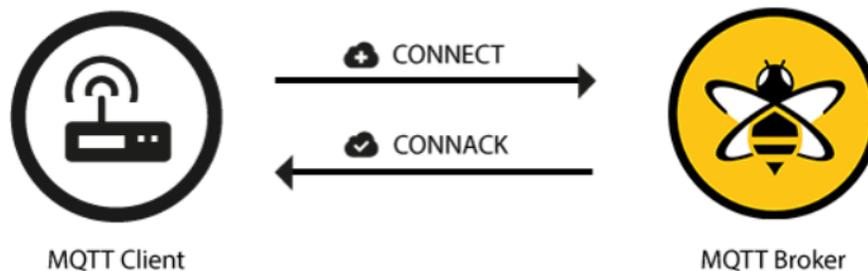


Figura 10: Inicio de la conexión MQTT entre cliente y servidor.

El contenido del mensaje CONNECT puede variar según el servidor, pero debe de aportar al servidor la información para la autorización del cliente (usuario y contraseña) y del tipo y calidad de conexión a establecer, que se verá más adelante en la “Calidad del servicio”. Si el mensaje no está correctamente estructurado o pasa mucho tiempo desde que se abre la conexión y se envía este mensaje, el bróker cierra la conexión para evitar clientes “malintencionados” 10.

El mensaje CONNACK, por otro lado, contiene dos parámetros: un primer elemento booleano, que indica si ya existía o no previamente una conexión entre dicho cliente y el bróker, y un segundo parámetro con un valor que indica el estado de la conexión. En esta última variable, el bróker devuelve un 0 si la conexión es aceptada, mientras que devuelve un valor entre 1 y 5 si la conexión se rechaza [10].

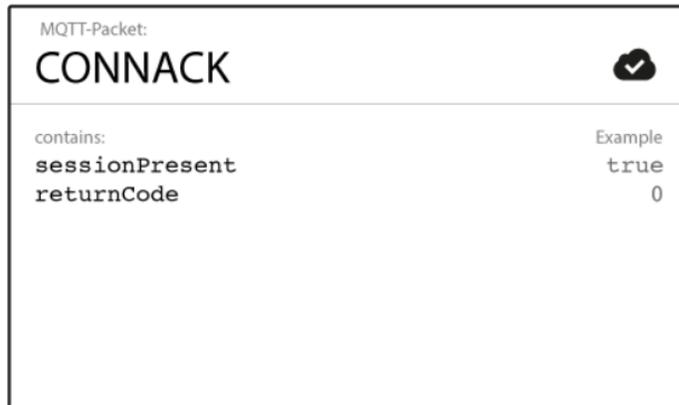


Figura 11: Contenido del mensaje CONNACK.

Return Code	Return Code Response
0	Connection accepted
1	Connection refused, unacceptable protocol version
2	Connection refused, identifier rejected
3	Connection refused, server unavailable
4	Connection refused, bad user name or password
5	Connection refused, not authorized

Figura 12: Código del estado de conexión en el mensaje CONNACK.

6.10. Protocolo HTTP/1.1/

HTTP/1.1 es un protocolo de tipo cliente-servidor que opera con mensajes pedido/respuesta (request /reply). El cliente es el programa que establece conexiones con el objetivo de enviar pedidos (requests), se lo denomina agente de administrador y puede ser un navegador [11].

El servidor es un programa que acepta conexiones entrantes para responder a los

pedidos (requests), con el envío de respuestas (replies). El servidor debe estar activo previo al envío de pedidos del cliente. Un mismo software, o equipo de la red, puede actuar simultáneamente como cliente y servidor [11].

HTTP/1.1 provee encabezados (headers) para enviar el pedido, con métodos para indicar el tipo de pedido y define a la ubicación del recurso referido a partir de su URI (Uniform Resource Identifier) o su URN (Uniform Resource Name). Los mensajes son transmitidos en formato tipo MIME (Multipurpose Internet Mail Extensions) [11].

La Figura 13 muestra el diagrama de secuencia de una interacción entre un cliente y un servidor, comunicados de forma directa, donde el cliente envía un pedido y el servidor envía su respuesta, luego de procesar el mismo [11].

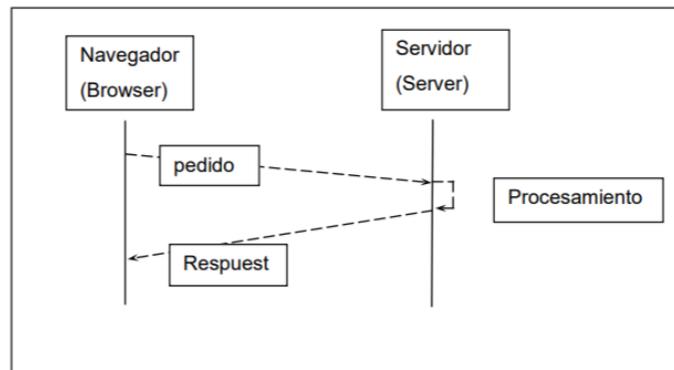


Figura 13: Diagrama de secuencia de una interacción simple entre un cliente y un servidor.

Comandos de HTTP

El protocolo HTTP/1.1 consta de los siguientes comandos:

- GET: sirve para recoger cualquier tipo de información del servidor. Se utiliza siempre que se pulsa sobre un enlace o se teclea directamente a una URL como resultado, el servidor HTTP envía el documento ubicado en la dirección específica por dicha URL.
- HEAD: es un comando similar a GET pero que pide solamente la cabecera del objeto. Lo utilizan principalmente los gestores de caches de páginas o los servidores proxy para conocer cuando es necesario actualizar la copia que se mantiene de un fichero.
- POST: este comando envía datos de información al servidor, normalmente procedentes de un formulario web, para que el servidor los administre o los añada a una base de datos.
- PUT: almacena un objeto en la URL especificada. Si la dirección de destino ya contenía un objeto, se considera que se está enviando una versión actualizada

del mismo.

- DELETE: elimina el objeto especificado. Este comando es muy poco utilizado.
- TRACE: realiza un eco de la solicitud recibida para que el cliente pueda conocer que servidores intermedios están añadiendo información o modificando la petición.
- OPTIONS: devuelve los métodos HTTP que soporta el cliente. Se suele utilizar para comprobar la funcionalidad de un servidor web.
- CONNECT: se utiliza en los servidores proxy que puedan establecer un túnel dinámicamente.

[11]

Ante cada transacción con el servidor HTTP, este devuelve un código numérico en la primera línea del mensaje de respuesta que informa sobre el resultado de la operación. Estos códigos aparecen en algunos casos en la pantalla del cliente, cuando se produce un error [11].

El cliente normalmente crea una comunicación con el servidor a través de un puerto predeterminado (puerto 80) y le envía los mensajes con los datos de su solicitud. El servidor le responde con un mensaje indicando el estado de la operación de consulta y su posible resultado, seguido de la información que haya solicitado como tal [11].

Los resultados normalmente se conocen como objetos o recursos y suelen ser páginas HTML, imágenes o archivos binarios. La comunicación completa entre clientes y servidores se realiza a partir de caracteres de 8 bits. Así, se puede transmitir cualquier tipo de documento: texto, binario, etc., sin cambiar su formato original [11].

Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Esto quiere decir que para cada objeto, archivo o imagen que pueda contener la solicitud del cliente, se realizará una conexión para transmitirlo. Una vez transmitido, el protocolo TCP se encargará de cerrar la conexión, para establecer una nueva de ser necesario [11].

El protocolo no mantiene el estado, esto quiere decir que cada petición de un cliente a un servidor no depende, ni es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto de operaciones que le preceden o le anteceden [11].

La comunicación con los servidores HTTP, normalmente se establece a través de mensajes de texto, que contienen los diferentes comandos y opciones del protocolo [11].

6.11. Interfaz de usuario

6.11.1. React.js

React es una biblioteca de JavaScript creada para construir interfaces de usuario rápidas e interactivas para aplicaciones web y móviles. Es una biblioteca de código abierto, basada en componentes, responsable únicamente de la capa de vista de la aplicación. En la arquitectura Modelo-Vista-Controlador (MVC), la capa de vista es responsable del aspecto y la sensación de la aplicación. React fue creado por Jordan Walke, un ingeniero de software de Facebook [12].

6.11.2. React Native

React Native es un framework para la creación de aplicaciones móviles de código abierto creado por Facebook. Se utiliza para desarrollar aplicaciones para Android, iOS, Web y UWP (incluso existe un puerto incompleto para Qt) al permitir a los desarrolladores integrar los componentes de React junto a las mejores capacidades de una plataforma nativa [13].

6.11.3. Node.js

Node.js es una plataforma de software que está construida sobre el tiempo de ejecución de JavaScript V8 de Chrome para construir aplicaciones de red escalables sin esfuerzo. Node.js utiliza un sistema de modelo de I/O basado en eventos y sin bloqueos que lo hace ligero y eficiente, perfecto para aplicaciones en tiempo real con gran cantidad de datos de datos en tiempo real que se ejecutan en dispositivos distribuidos. Node.js, desarrollado inicialmente por Ryan Dahl, también proporciona un entorno REPL (Read-Eval-Print-Loop) para realizar pruebas interactivas [14].

6.11.4. Firebase

Para la realización del back-end del proyecto se utilizó Firebase como servidor. Firebase ofrece un conjunto de herramientas divididas en diferentes módulos: desarrollo, crecimiento y monetización (análisis). Para la realización del proyecto nos centraremos en el grupo de desarrollo, también llamado Develop en Firebase [15].

Dentro del grupo de desarrollo, Firebase proporciona una herramienta para gestionar una base de datos en tiempo real alojada en la nube. La base de datos es NoSQL y almacena los datos en formato JSON y funciona de tal manera que se sincroniza en tiempo real con cada cliente conectado, por lo que, si un usuario realiza una modificación, se notifica a la base de datos y simultáneamente es actualizada la información para el resto de los usuarios [15].

Cabe destacar que, si un usuario realiza cambios y no dispone de conexión a Internet, la plataforma usa una caché local en el dispositivo donde guarda estos cambios. Una vez que vuelve a tener conexión, automáticamente se sincronizan los datos locales [15].

Aparte del servicio de Real-Time Database, en el proyecto se utilizará el servicio de autenticación proporcionado por el mismo Firebase. Permite tanto el registro propiamente dicho (mediante email y contraseña) como el acceso a través de otros proveedores utilizando los perfiles correspondientes (por ejemplo, de Facebook o Google). Al aportar este servicio, Firebase nos facilita la creación de sistemas de autenticación, sin necesidad de que nos preocupemos en desarrollarlos, asegurándonos el acceso rápido y seguro [15].

6.12. Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

El Arduino DUE es una placa que incluye un microcontrolador - ver Figura 14. El microcontrolador que utiliza es un CPU Atmel SAM3X8E ARM Cortex-M3. Es un microcontrolador de alto procesamiento de 32 bit y una velocidad máxima de 84 MHz. Incluye 5 UARTs, 4 canales de SPI, 3 timers de propósito general de 32 bits, 12 bit ADC y 12 bits DAC entre muchas otras cosas. La placa Arduino DUE provee 54 pines de entrada y salida, de los que 12 se pueden usar como PWM, 4 UART, 12 analógicos, 2 DAC, entre otros. Cuenta con 512K bytes de memoria Flash y 96K bytes de memoria SRAM [16].

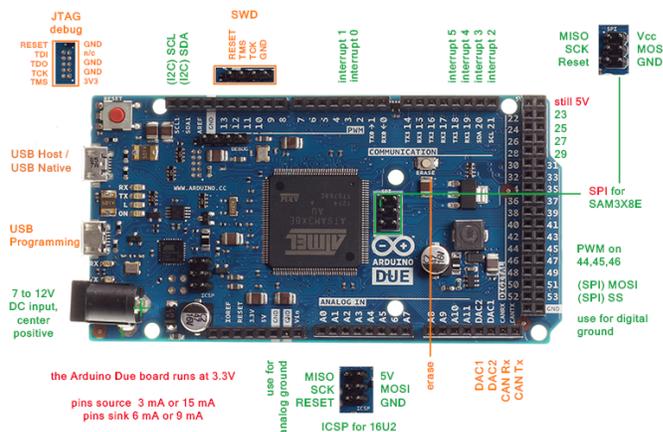


Figura 14: Arduino DUE.

6.13. Comunicación de radiofrecuencia

La comunicación por radiofrecuencia es una comunicación inalámbrica entre dos dispositivos electrónicos. Utiliza ondas del espectro electromagnético que utiliza ondas desde los 3 Hz hasta los 300 GHz. El módulo electrónico NRF24L01 utiliza este tipo de ondas para su comunicación inalámbrica. Este se usará en el trabajo para la comunicación del control remoto con el robot móvil. Se utilizan dos módulos y ambos funcionan como emisor y receptor, es decir la comunicación es bi-direccional. Es capaz de transmitir 32 bytes. Se utiliza una antena para aumentar la distancia máxima de comunicación (aprox. 1Km) [17]. En la Figura 15 se observa el módulo con la antena.

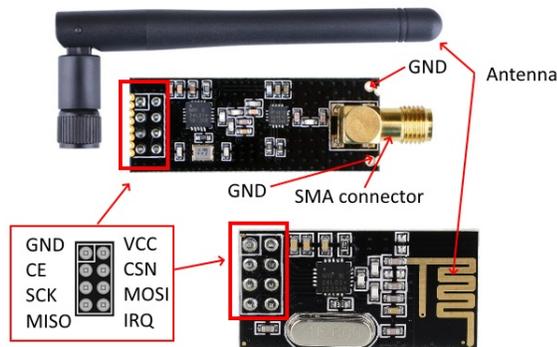


Figura 15: NRF24L01.

6.14. Cálculo de batería para el control remoto

La no linealidad de la batería se toma en cuenta mediante la fórmula de Peukert:

$$T = \frac{C}{I^n} \quad (1)$$

Donde T es el tiempo de vida de la batería, C es la capacidad teórica medida en Amperios-Hora que se encuentra en la hoja de datos de la batería y por último, I es la corriente que entrega constantemente la batería y n es el numero de Peukert, que es el coeficiente de no linealidad de la descarga de baterías. Las baterías tienen un número de Peukert de $n = 1$ a $n = 1.5$ [18].

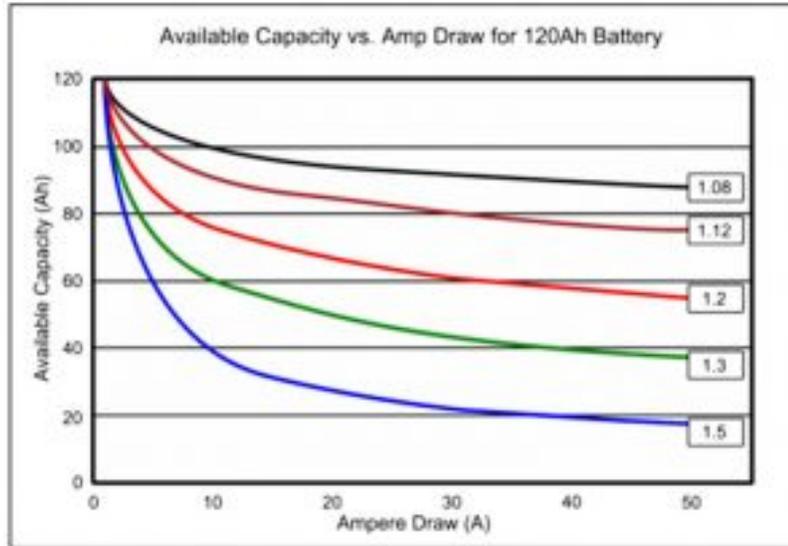


Figura 16: Constante de peukert para una batería de litio [19]

7.1. Implementación

7.1.1. Hardware

Comunicación UART



Figura 17: Serial selected

Para esta aplicación en específico se decidió utilizar el software serial selected.

Powering Up/Down SIM900 chip

Además de encender el shield, se debe de encender el chip SIM900 para que funcione el módulo correctamente. Para esta aplicación en específico se necesita que el encendido sea totalmente manipulado por software, según la hoja de datos se puede activar el software trigger al soldar el jumper R13 como se indica a continuación.

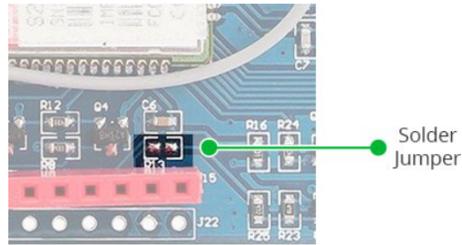


Figura 18: Software trigger

Alimentación del Shield SIM900

Para esta aplicación en específico se utilizó una fuente de alimentación externa debido a que la fuente de alimentación debe ser capaz de generar un mínimo de 2 A de corriente, de lo contrario, corre el riesgo de apagarse. Como se fundamenta a continuación.

Modes	Frequency	Current Consumption
Power down		60 uA
Sleep mode		1 mA
Stand by		18 mA
Call	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453 mA
Transmission burst		2 A

Figura 19: Consumo del shield SIM900 [20]

7.1.2. Software

Para conseguir la comunicación entre el sistema electrónico del robot móvil y Adafruit IO se realizaron las siguientes configuraciones:

Comunicación UART

El shield SIM900 GSM / GPRS utiliza el protocolo UART para comunicarse con el Arduino Due. El chip admite una velocidad en baudios de 1200 bps a 115200 bps

con detección automática de baudios. Se usaron los pines D16(TX2) y D17(RX2) en el Arduino Due, además se usaron los pines D7(TX) y D9(RX) en el shield SIM900. Como se puede observar en la Figura 20.

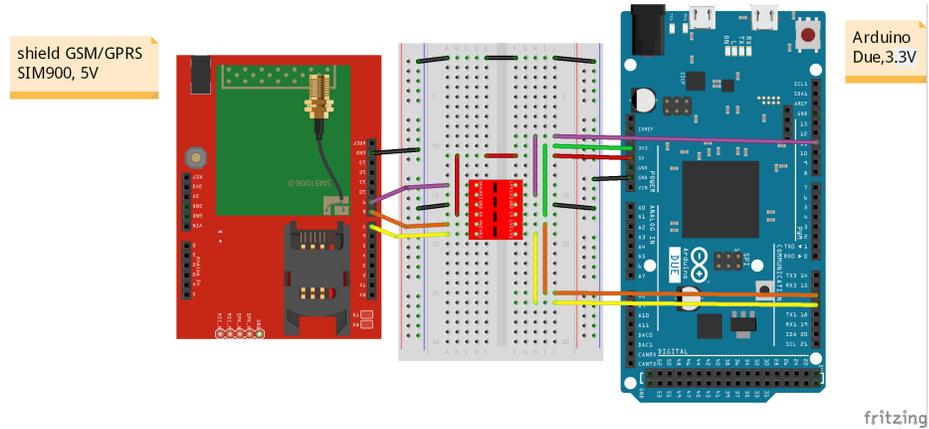


Figura 20: Esquemático SIM900 Y Arduino Due

Modelo TCP/IP

La API HTTP de Adafruit IO proporciona acceso a sus datos de Adafruit IO desde cualquier lenguaje de programación o entorno de hardware que pueda hablar HTTP, el cual debe seguir la estructura descrita en la Figura 21. Por esa razón, la capa de aplicación es dicho protocolo el cual permitirá la comunicación a través de la red una vez que se establezca la conexión.

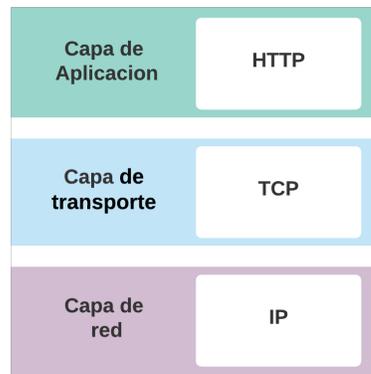


Figura 21: Solicitud HTTP

Las características de TCP / IP proporcionadas por los comandos AT para el SIM900 están destinadas a proporcionar un subconjunto de las características que normalmente están disponibles en el nivel de transporte cuando se utiliza un modelo TCP / IP convencional, con algunas simplificaciones y personalizaciones con base en

las características específicas de la SIM900.

Los comandos permiten al escritor de la aplicación crear sockets TCP, para controlar contextos PDP GPRS ya configurados, para transferir datos hacia y desde el módulo, y para interrogar la información del estado de IP sobre el enlace activo. La interfaz de socket es proporcionada por una serie de comandos AT para la conexión TCP/IP que se describen en la Tabla 11.

La Data que se observara en la página web estará contenida en un servidor que posee una base de datos la cual será proporcionada por Adafruit IO que almacenara las variables a supervisar (velocidad, tiempo de batería, etc..) enviadas por el Shield SIM900 a través de una Petición HTTP 1.1.

La configuración quedaría de la siguiente manera:

Detalle	Contenido
PROTOCOLO	TCP
DIRECCIÓN IP	io.adafruit.com
PORT	80

Cuadro 2: Comandos TCP

Punto de acceso a la red

Para poder enviar datos hacia Internet por medio del shield GSM/GPRS se necesita tener a disposición un **APN (Access Point Name)**: (Access Point Name) que es el nombre del punto de acceso que la operadora de telefonía móvil posee para que sus dispositivos abonados se conecten a internet. El APN es un camino virtual que permite que los dispositivos móviles naveguen por la red.

Se ha decidido trabajar con la operadora Claro y la configuración quedaría de la siguiente manera:

Detalle	Claro
APN	internet.ideasclaro
USUARIO	claro
CONTRASENA	claro

Cuadro 3: Comandos APN

Solicitud HTTP 1.1

Para completar los apartados solicitados en la petición HTTP 1.1 fue necesario consultar los requerimientos de Adafruit IO, los cuales se pueden observar en la Figura 22.

HTTP Request

POST /api/v2/{username}/feeds/{feed_key}/data

New data

Path Parameters

Parameter	Type	Required	Description
username	string	true	a valid username string
feed_key	string	true	a valid feed key

Body Parameters

Parameter	Type	Required	Description
datum	object	true	Data record including a <code>value</code> field (required) and optionally including: <code>lat</code> , <code>lon</code> , <code>e1e</code> (latitude, longitude, and elevation values), and <code>created_at</code> (a date/time string).

Figura 22: Create Data Adafruit IO

Posteriormente la petición HTTP 1.1 que se realiza al servidor debe ser configurada, ya que esta se realiza mediante el método POST se define en la programación de Arduino como se pueden observar en la Figura 23.

```

static String PostData ="value="+String(BATERIA);

static String post1 = "POST /api/v2/Lop17065/feeds/nivel-de-bateria/data HTTP/1.1\r\n";
  post1 += "Host: io.adafruit.com\r\n";
  post1 += "X-AIO-KEY: aio_venk51jGs1oLM5r9ldgZOTAzH9LR\r\n";
  post1 += "Content-Type: application/x-www-form-urlencoded\r\n";
  post1 += "Content-Length: ";
  post1 +=String(PostData.length())+"\r\n";
  post1 += "Connection: keep-alive\r\n";
  post1 += "\r\n";
  post1 += PostData; } Request Message Body
} Request Headers
} Request Line
Request Message Headers

```

Figura 23: Petición HTTP

Algoritmo para el envío de datos

En la Figura 24 se detalla el proceso para enviar los datos al servidor web.

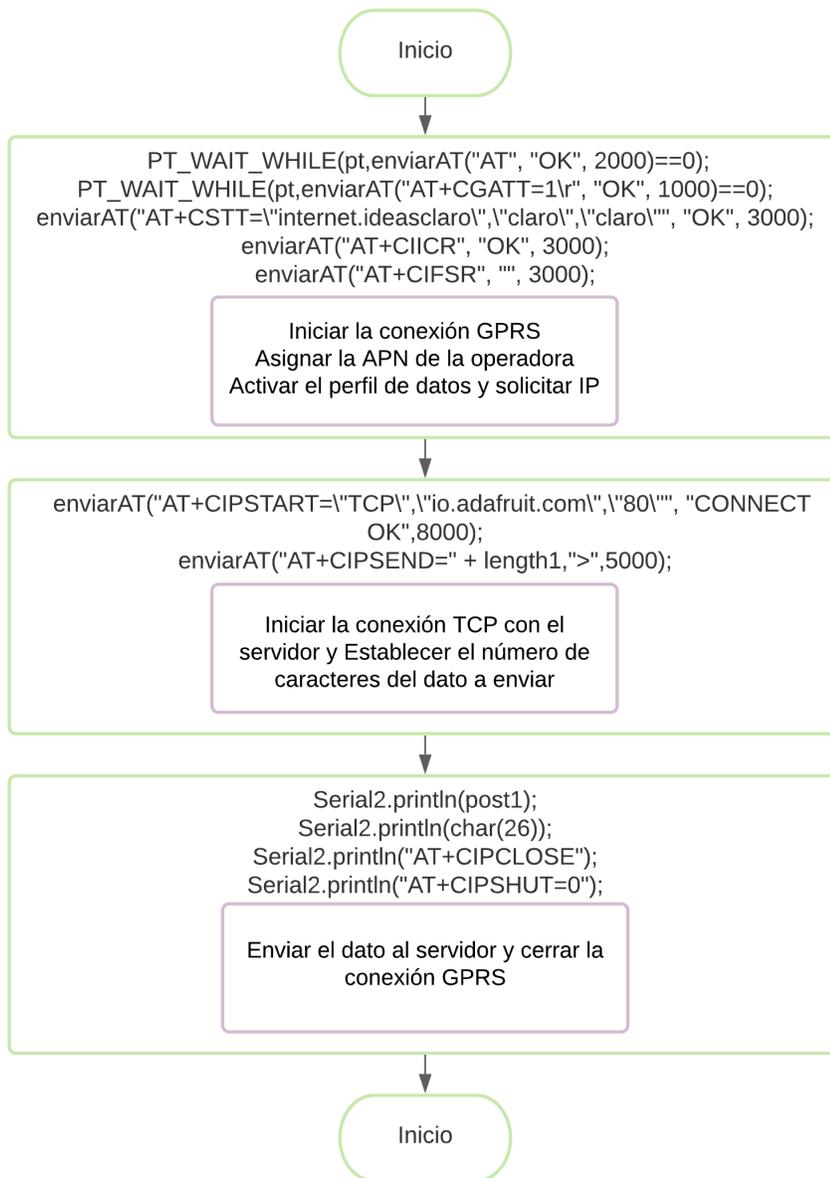


Figura 24: Flujograma GPRS

7.2. Pruebas y resultados

7.2.1. Comunicación Arduino Due -> Shield SIM900 -> Adafruit IO

Como se observa en la Figura 25 la conexión GPRS y TCP fueron satisfactorias debido a que se obtuvieron las respuestas previamente mencionadas en la Tabla 1

```

Iniciando...
AT
OK
Conectado a la red.
AT+CGATT=1
OK
AT+CSTT="internet.ideasclaro","claro","claro"
OK
AT+CIICR
OK
A
T+CIFSR
10.72.67.2
AT+CIPSTART="TCP","io.adafruit.com","80"
OK
CONNECT OK
AT+CIPSEND=230

```

Iniciar la conexión GPRS

Asignar la APN de la operadora Activar el perfil de datos

solicitar IP, Iniciar la conexión TCP con el servidor y Establecer el número de caracteres del dato a enviar

Figura 25: Prueba de conexión TCP

Para la petición HTTP 1.1 que se observa en la Figura 26 la respuesta de Adafruit IO fue satisfactoria, debido a que presenta la información de los detalles de la data que fue creada en el feed nivel de batería.

```

>
POST /api/v2/Lop17065/feeds/nivel-de-bateria/data HTTP/1.1
Host: io.adafruit.com
X-AIO-KEY: aio_venk51jGsl0LM5r9IdgZ0TAzH9LR.com
X-AIO-KEY: aio_venk51jGsl0LM5r9IdgZ0TAzH9LR
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Connecti2021 17:14:06 GMT
content-type: application/json; charset=utf-8
transfer-encoding: chunked
x-frame-options: SAMEORIGIN
x-xs, DELETE, GET, OPTIONS, PATCH
access-control-allow-headers: DNT, Origin, X-Requested-With, X-AIO-Key, Content-Type, Accept, Auxy: 1

C3
{"id":"0EV8PFTV2RGET73RV1E92GCWB2","value":"27","feed_id":1689715,"feed_key":"nivel-de-bateria","created_at":"2021AT+CIPSEND=218

```

Request Line

Request Headers

Respuesta de Adafruit

Figura 26: Prueba de petición HTTP 1.1

Como se observa en la Figura 27 los datos fueron enviados correctamente y la respuesta de la petición HTTP 1.1 fue aceptada por el servidor.

```
SEND OK } Envio de datos por conexión TCP
HTTP/1.1 200 OK } Response
server: nginx
date: Thu, 30 Sep 2021 17:46:33 GMT
content-type: application/jstestrunner
access-control-request-method: *
access-control-allow-headers: *
access-control-allow-methods: POST, PUT, DELETE, GET, OPTIONS, private, must-revalidate
x-request-id: cde91cb6-dfd1-4e05-b3fe-9719fed524b0
x-runtime: 0.046446
x-aio-proxy: 1
```

Figura 27: Respuesta de la petición HTTP 1.1

Como se observa en la Figura 28 los datos fueron recibidos correctamente por Adafruit IO, confirmando positivamente los resultados anteriormente descritos.



Figura 28: Dashboard de Adafruit IO

8.1. Firebase

Firestore se utilizó como hosting tanto de la aplicación web como de la aplicación móvil, simultáneamente se utilizó para la autenticación de los usuarios en el sistema.

8.1.1. Autenticación de los usuarios

Se implementó Firebase Authentication para gestionar las cuentas de todos los usuarios que usan la aplicación. Debido a que dispone de un sistema de autenticación que emplea únicamente código en el lado del cliente, siendo asimismo posible personalizar las reglas de los criterios de acceso de los usuarios. Permite a su vez el uso de una gran cantidad de opciones para poder acceder como usuario a la aplicación a desarrollar, ya sea bien usando un correo y contraseña, una cuenta de Google, una cuenta de Facebook, etc.

Para este proyecto se ha decidido acceder por medio de correo y contraseña y por medio de una cuenta de Google, como se observa en la Figura 29. Además, dentro de las aplicaciones se tendrá a disposición del usuario la recuperación de contraseña, mediante el envío de un correo electrónico desde Firebase al usuario para el ingreso de la nueva contraseña. La autenticación de las credenciales será realizada por el servicio de Firebase Authentication.

8.1.2. Hosting

Las pruebas de funcionalidad se realizaron con ayuda de la herramienta Firebase Hosting, un servicio de hosting de contenido web con un nivel de producción orientado a programadores. Con Hosting, se puede implementar apps web y contenido estático en una red de distribución de contenido (CDN) global con un solo comando, de forma rápida y sencilla. De esta manera se probará la aplicación en diferentes navegadores y ver su funcionalidad en práctica.

Para concluir este apartado, cabe destacar que el uso de Firebase es gratuito, siempre y cuando no se sobrepasen los límites establecidos en su plan de pago. El plan escogido para el proyecto es el denominado plan Spark, plan gratuito si no se exceden los 10000 usuarios autenticados al mes, 1 GB de datos almacenados en Cloud Firestore y 5 GB almacenados en Storage.

8.2. Servidor MQTT

El servidor utilizado en el proyecto fue HIVEMQ, debido a que ofrece un amplio abanico de soluciones a empresas que requieran el uso de un servidor privado, pero en el desarrollo del proyecto se ha utilizado un bróker público que tienen habilitado para uso libre, ubicado en la dirección `broker.hivemq.com`. Además, facilitan un dashboard en su portal web con información acerca del uso que se hace en tiempo real del bróker, como se puede apreciar en la Figura 31.

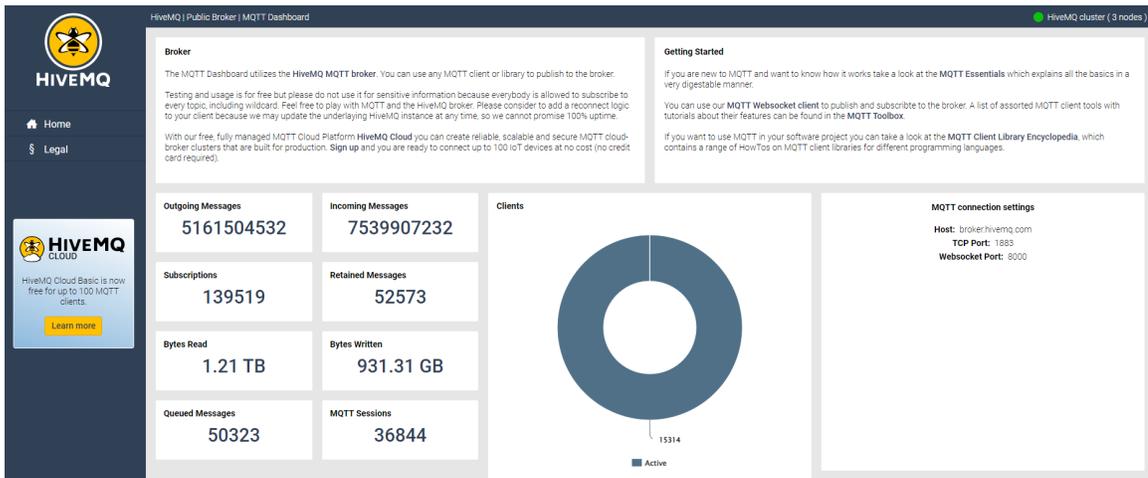


Figura 31: Interfaz de HiveMQ MQTT Dashboard

La propuesta del servidor tiene como objetivo principal ahorrar tráfico al servidor, de manera que este tan solo se encargue de atender peticiones que impliquen a la base de datos de Adafruit IO, a la base de datos en MYSQL y a la aplicación móvil. Por lo tanto, la arquitectura del servidor se detallar en la Figura 32, en la que se observan

las relaciones entre clientes referenciadas por color.

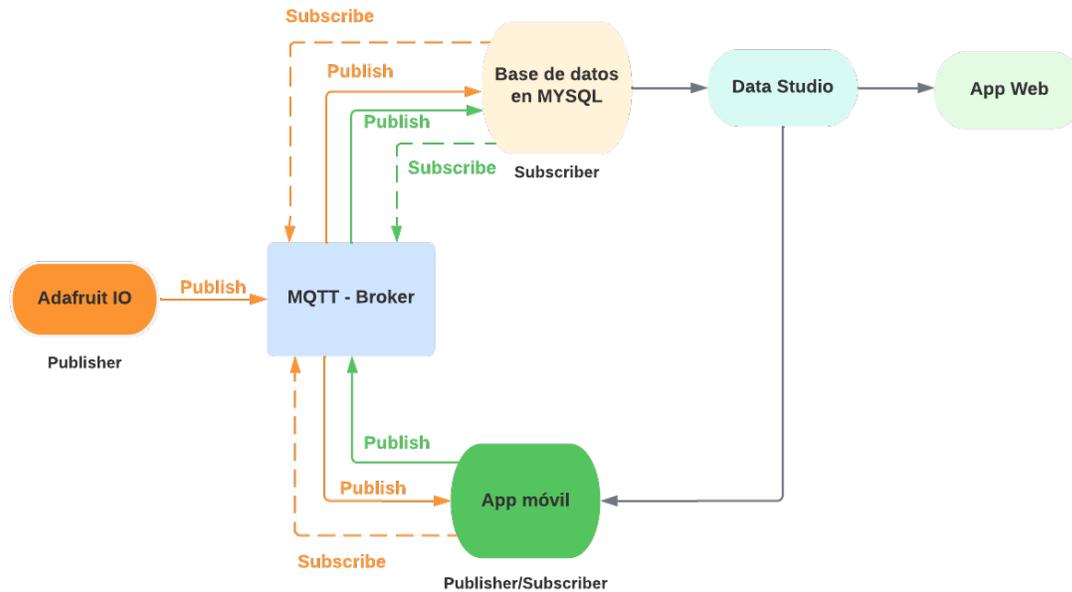


Figura 32: Arquitectura del servidor mqtt

8.2.1. Cliente Adafruit IO

Como se ha mencionado anteriormente, la idea principal de utilizar Adafruit IO es que el robot móvil publique los datos correspondientes cada uno de los feeds que se muestran en la Figura 33, los cuales se crearon en la plataforma de Adafruit IO. De esta manera se estará generando un primer filtro de las variables que se consideran relevantes del robot móvil.

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> Ciclos completados	ciclos-completados		7 months ago
<input type="checkbox"/> duracion de bateria	duracion-de-bateria		1 minute ago
<input type="checkbox"/> Nivel de bateria	nivel-de-bateria	180	4 months ago
<input type="checkbox"/> registro de operaciones	registro-de-operaciones		3 minutes ago
<input type="checkbox"/> Tiempo de secado	tiempo-de-secado		3 minutes ago
<input type="checkbox"/> Velocidad	velocidad	25	about 3 hours ago

Figura 33: Feeds del sistema

Posteriormente se integró Adafruit IO como cliente del servidor mqtt que se desarrolló para este proyecto, debido a que el protocolo mqtt facilitará que el cliente MYSQL y el cliente aplicación móvil se suscriban a los feeds deseados a través del

host HIVEMQ que actúa como servidor de la red. Los feeds asignados a cada cliente se observan en la Figura 34.

Cliente MYSQL	Cliente aplicación móvil
Ciclos completados	Ciclos completados
Duración de la batería	Duración de la batería
Registro de operaciones	Registro de operaciones
Tiempo de secado	Tiempo de secado
Velocidad	Velocidad
	Nivel de batería

Figura 34: Feeds asignados de cada cliente

8.2.2. Cliente base de datos en MYSQL

Como se ha mencionado anteriormente, existe la necesidad de almacenar los datos tanto del robot móvil como los proporcionados por la aplicación móvil, por ello se ha escogido MySQL como sistema de gestión de base de datos principal. Por otra parte, se ha decidió usar phpMyAdmin proporcionado por db4free.net para la gestión de las bases de datos MySQL. Esta herramienta escrita en PHP facilita la administración ofreciendo una interfaz gráfica a través de un navegador web. Además, proporciona gratuitamente 200 MB de almacenamiento, los cuales serán suficiente para el proyecto en cuestión.

Por último, se creó dentro de la base de datos llamada “baseinjerto” la tabla llamada “Robot Movil” que se observa en la Figura 35, en la que se guarda la información recibida desde el cliente Adafruit io y el cliente aplicación móvil mediante el protocolo MQTT. Para posteriormente ser visualizada tanto en el dashboard de la aplicación web como en el de la aplicación móvil.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 fecha	datetime			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2 procesos_realizados	int			Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	3 t_secado	int			Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	4 recargas_baterias	int			Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	5 velocidad_promedio	int			Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	6 registro_operaciones	text	utf8mb4_0900_ai_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	7 duracion_bateria	int			Si	NULL			Cambiar Eliminar Más

Figura 35: Estructura de la base de datos MYSQL

8.2.3. Cliente aplicación móvil

Como se menciona anteriormente, la idea principal de utilizar una aplicación móvil es el poder visualizar en tiempo real el proceso de secado de café que el robot móvil este desarrollando. Por esa razón, el cliente aplicación móvil estará suscrito a los feeds que se observan en la Figura 34, pero al mismo tiempo se encargara de publicar cuando el usuario realice una recarga de batería. La acción que se menciona anteriormente se verá reflejada en el contador de recargas de batería del dashboard tanto de la aplicación web como en el de la aplicación móvil.

8.3. Pruebas y resultados

8.3.1. Firebase

Se realizaron ambos registros y las credenciales fueron asignadas correctamente, como se observa en la Figura 36. Por lo tanto, las autenticaciones que realizó el servicio de Firebase Authentication fueron satisfactorias.

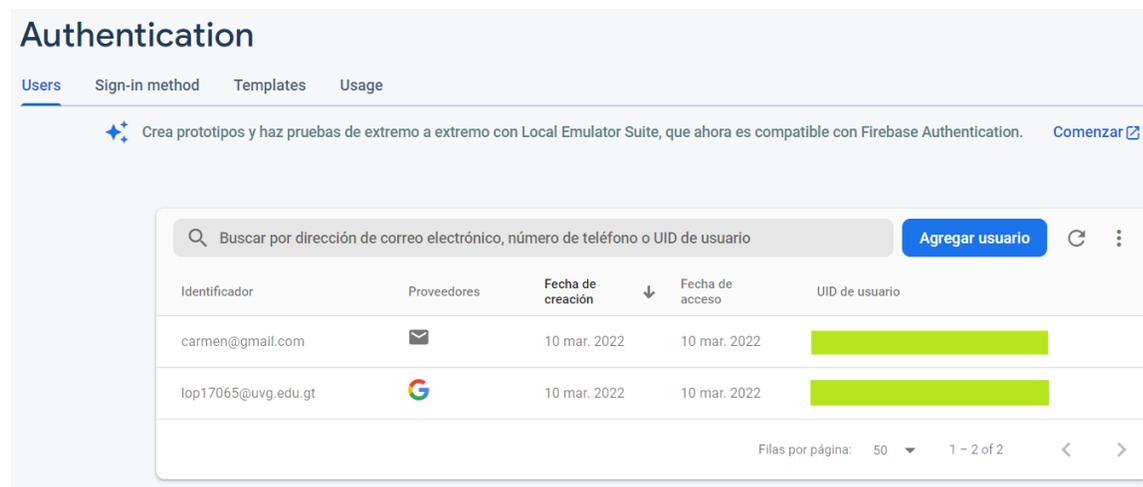


Figura 36: Prueba de los métodos de autenticación

8.3.2. Servidor MQTT

Para comprobar el funcionamiento del servidor mqtt se ingresa un valor en el feed de velocidad como se observa en la Figura 37.

Created at	Value	Location
2022/03/31 12:22:18AM	26	

Figura 37: Prueba del cliente adafruit io

Posteriormente se comprueba que el cliente Adafruit io realizará la publicación del valor mencionado anteriormente de manera correcta y que además el cliente MYSQL que está suscrito al feed de velocidad recibiera el dato, ambas operaciones se pueden comprobar con los resultados observados en la Figura 38.

```

PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node broker.js
Connected!
PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node broker.js
Connected!
PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node broker.js
Connected!
PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node cliente_adafruit.js
La conexión al servido de Adafrii io es exitos
a
La conexión al servidor HIVEMQ es exitosa
Tema actual: Lop17065/feeds/velocidad
valor: 26
velocidad 26
PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node cliente_mysql.js
La conexión al servidor HIVEMQ es exitosa
Database ok!
PS C:\Users\debor\OneDrive\Desktop\application-el-injerto\backend> node cliente_mysql.js
La conexión al servidor HIVEMQ es exitosa
Database ok!
Data Saved

```

Figura 38: Prueba del servidor MQTT/clientes

Luego, se comprobó que el dato recibido por el cliente MYSQL fue el correcto debido a que el valor observado en la Figura 39 coincide con el valor publicado por el cliente Adafruit io.

fecha	procesos_realizados	t_segado	recargas_baterias	velocidad_promedio	registro_operaciones	duracion_bateria
2022-03-30 23:11:39	1	25	1	20	AVANZAR	3
2022-03-30 23:14:18	1	25	1	20	AVANZAR	3
2022-03-31 03:36:14	1	25	1	20	AVANZAR	3
2022-03-31 03:37:00	1	25	1	21	AVANZAR	3
2022-03-30 22:07:37	1	36	1	22	RETROCEDER	8
2022-03-30 22:06:10	1	25	1	24	AVANZAR	3
2022-03-30 22:21:34	1	25	1	22	AVANZAR	3
2022-03-31 05:43:51	1	25	1	20	AVANZAR	3
2022-03-31 00:16:40	1	25	1	24	AVANZAR	3
2022-03-31 00:18:26	1	25	1	23	AVANZAR	3
2022-03-31 00:21:54	1	25	1	26	AVANZAR	3

Figura 39: Prueba del cliente MYSQL

Por último, se comprobó que el dato recibido por el cliente aplicacion móvil fue el correcto debido a que el valor observado en la Figura 40 coincide con el valor publicado por el cliente Adafruit io.

Nueva Recarga

26

Bateria Restante

95%



Figura 40: Prueba del cliente aplicacion móvil

9.1. Aplicación móvil

Para la aplicación móvil se usó React Native. React native provee un framework basado en JS (JavaScript) y que puede lograr una solución para ambas plataformas como Android y IOS sin necesidad de programar nativamente. La aplicación se encarga del análisis de los datos enviados por el robot móvil. Los datos están constituidos por: recargas de batería, duración del proceso de secado, cantidad de procesos realizados y el registro de operaciones. Además, se encarga de visualizar los datos del proceso de secado que el robot móvil este desarrollando en tiempo real. Los datos visualizados son: la velocidad y el nivel de batería del robot móvil. Por último, se tiene un botón que incrementa el contador de recargas de batería realizadas.

9.1.1. Prototipo final

En la Figura [41](#) se observa la pantalla de autenticación, en dicha pantalla se cuenta con dos campos de texto, uno para ingresar el correo electrónico y otro para la contraseña. Contiene también un botón de Inicio de Sesión el cual llamará a la función de autenticación de Firebase para validar las credenciales del usuario. Si esa validación es satisfactoria pasará a la siguiente pantalla. Además, cuenta con un botón de ingresar con una cuenta de Google el cual realizará la misma función descrita anteriormente, con la única salvedad de que utilizará los datos proporcionados por la cuenta de Google del usuario que la cree. Por último, se observa otro botón de registrarse el cual iniciará el flujo de registro de un nuevo usuario.

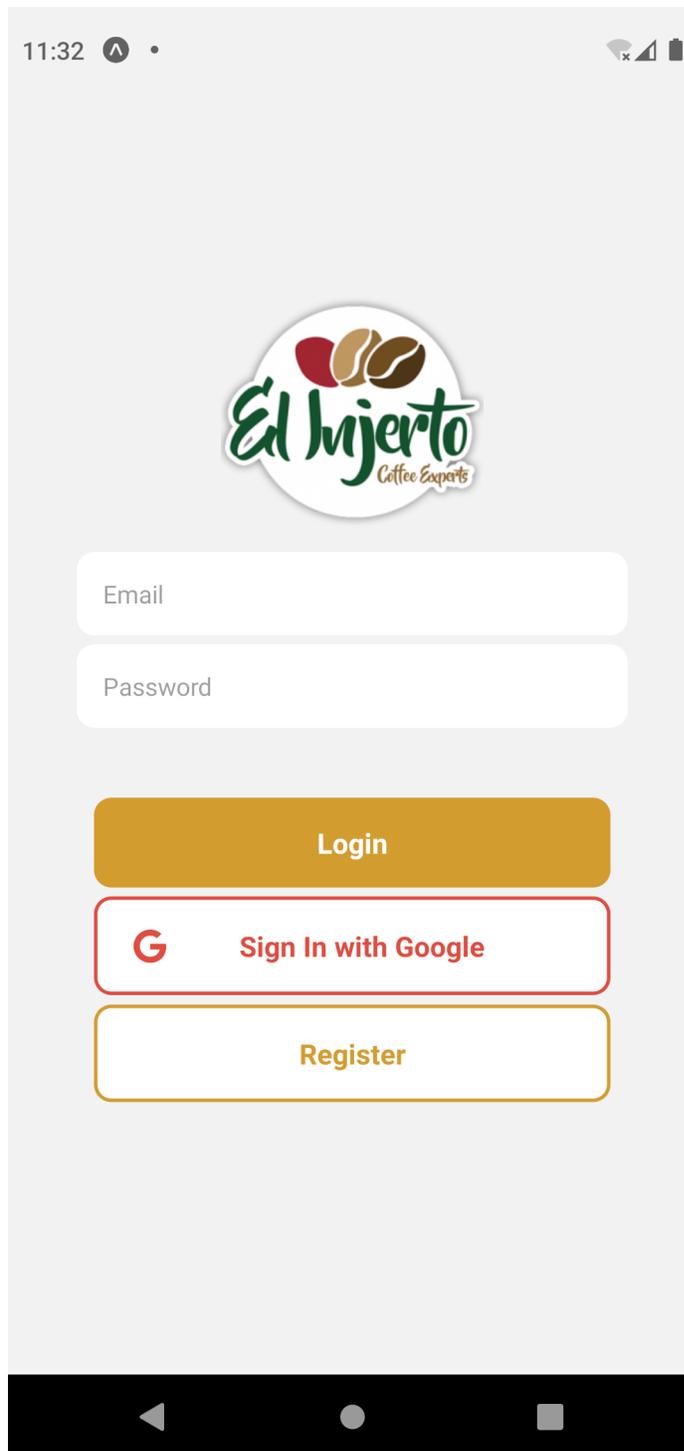


Figura 41: Autenticación de la aplicación móvil

En la Figura 42 se observa la pantalla de registro. Esta pantalla cuenta con dos campos de texto: uno para ingresar el correo electrónico y otro para la contraseña. Contiene también un botón de Registrar el cual permitirá que Firebase asigne un UID (User ID o ID de usuario) al nuevo usuario. De esa manera se estará completando el

flujo de Registro de un nuevo usuario. Por último, se observa otro botón de Login, el cual retornará a la pantalla de autenticación de usuarios previamente descrita.

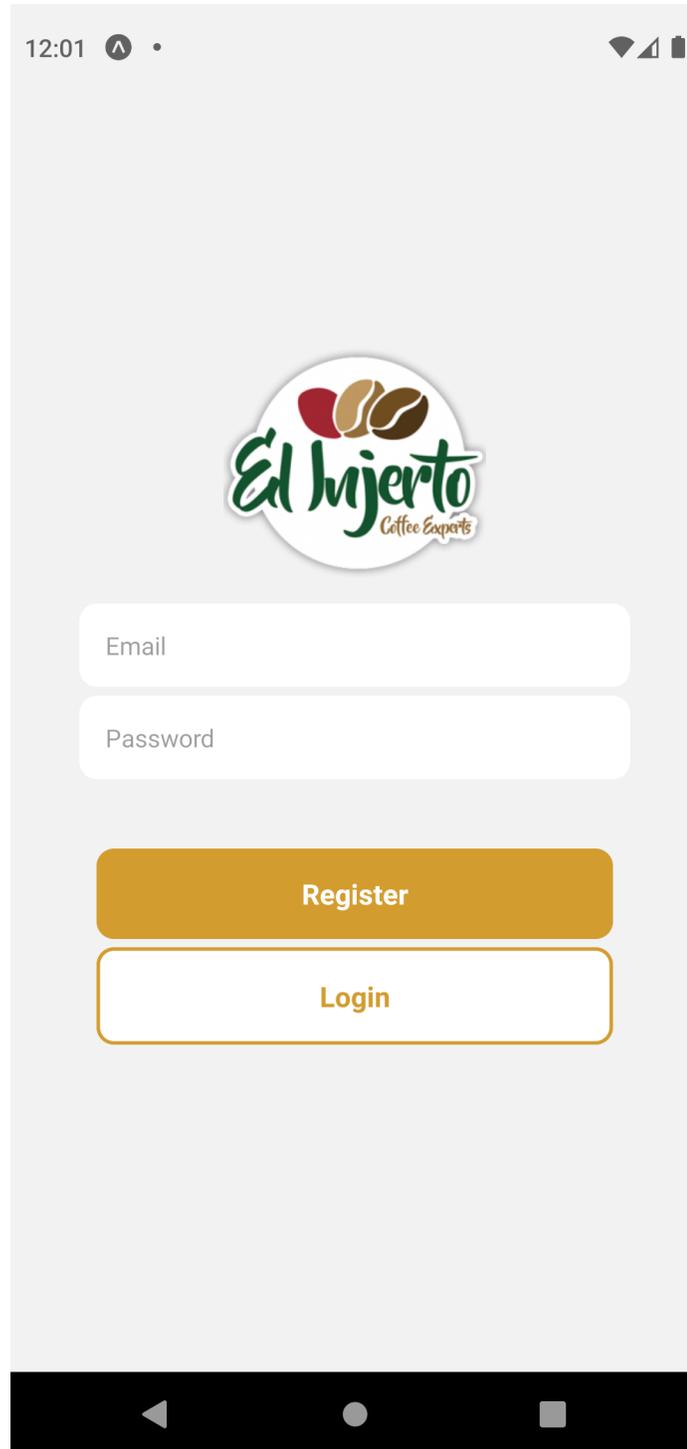


Figura 42: Registro de la aplicación móvil

En la Figura 43 se observa la pantalla del menú al cual el usuario accederá si la

validación resultó satisfactoria, en el ejemplo se accedió por medio de una cuenta de Google.

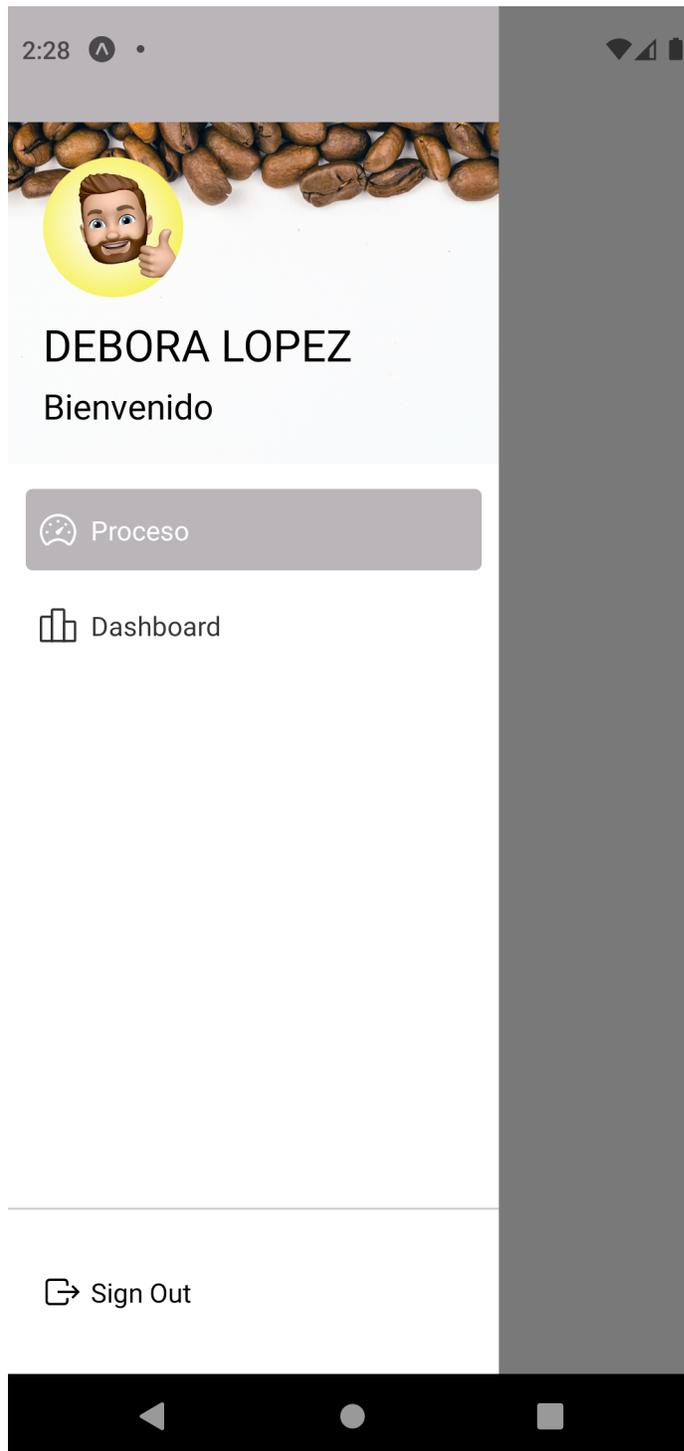


Figura 43: Menú de la aplicación móvil

En la Figura 44 se observa la pantalla del Dashboard en donde se mostrarán los

siguientes datos: recargas de batería, duración del proceso de secado, cantidad de procesos realizados y el registro de operaciones. Además, el usuario podrá filtrar los datos según el rango de tiempo que desee visualizar.

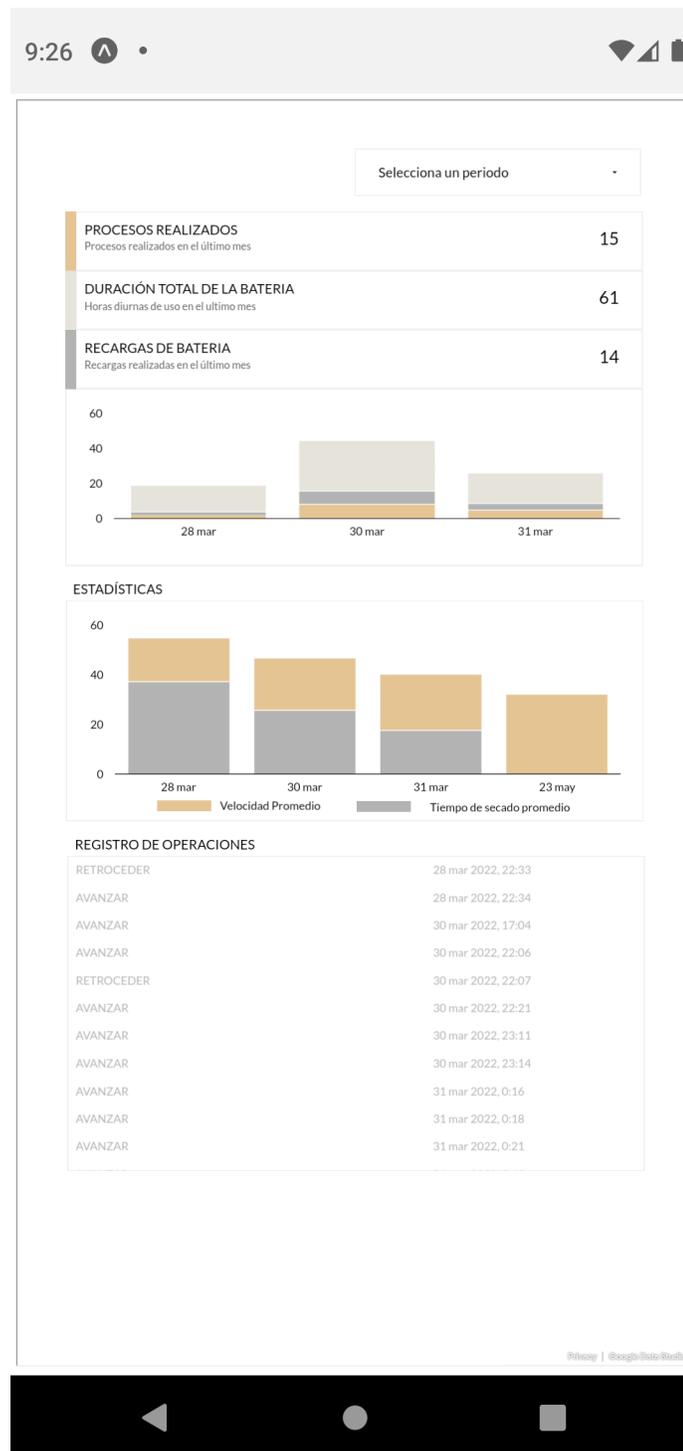


Figura 44: Dashboard de la aplicación móvil

En la Figura 45 Se observa la pantalla del WorkScreen en donde se mostrarán los datos del proceso de secado que el robot este realizando en tiempo real, además se observa la alerta que se mostrará al momento de que el usuario presione el botón de nueva recarga.

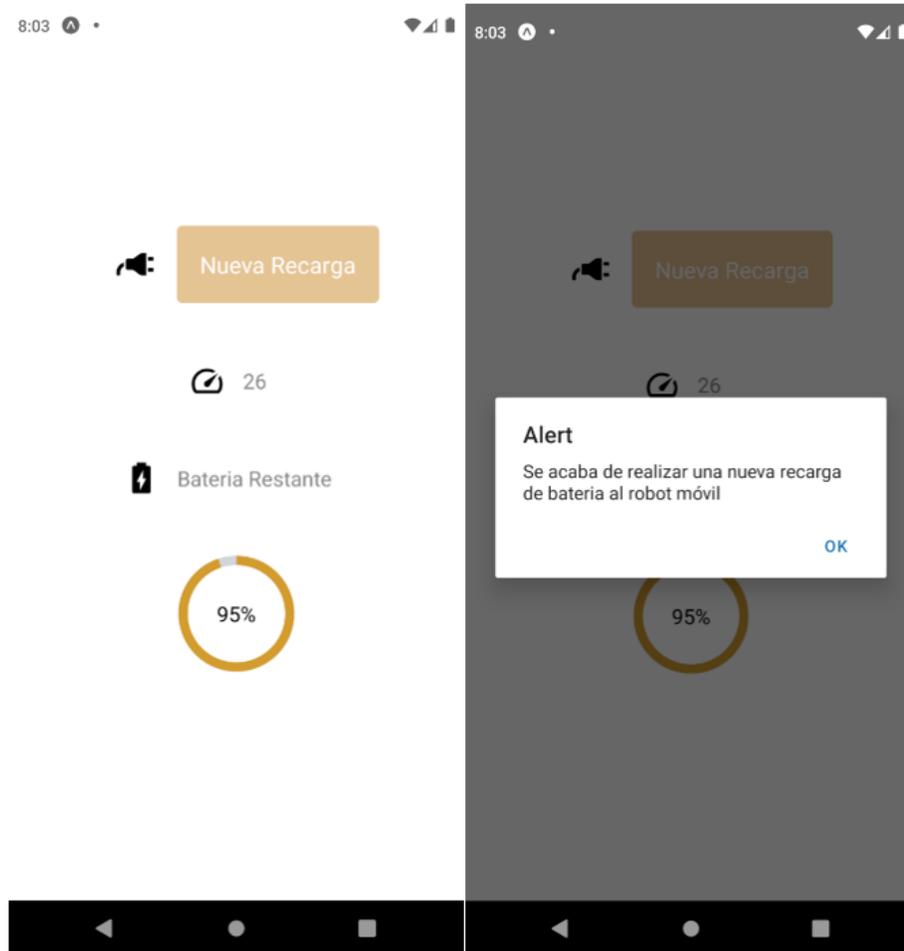


Figura 45: WorkScreen de la aplicación móvil

9.2. Dashboard/aplicación web

El dashboard fue realizado con React.js. La razón de escogerlo es la similitud a nivel de lógica y de sintaxis con el framework usado para la aplicación: React Native. Esta característica permite amalgamar de mejor manera la lógica usada en un proyecto y adaptarla al otro sin mayores reparos. Debido a eso, la curva de implementación se disminuye y es más simple prototipar más rápido y obtener resultados similares. La aplicación se encarga del análisis de los datos enviados por el robot móvil. Los datos están constituidos por: recargas de batería, duración del proceso de secado, cantidad de procesos realizados y el registro de operaciones.

Para los gráficos del dashboard se utilizó el software online Google Data Studio. Una de las alternativas fue el framework de pago Tableau Studio o Power BI de Microsoft, que pueden formar gráficos de manera dinámica y son sumamente potentes en ámbitos empresariales y comerciales. Sin embargo, se eligió Google por sobre el resto de alternativas debido a su dinamismo, accesibilidad, potencia y facilidad de uso.

9.2.1. Implementación del módulo de autenticación

La interfaz de la pantalla de autenticación se observa en la Figura 46. La misma es parte del módulo de ingreso al sistema que van a usar los usuarios previamente registrados.

La pantalla cuenta con dos campos de texto, uno para ingresar el correo electrónico y otro para la contraseña. Contiene también un botón de Inicio de Sesión el cual llamará a la función de autenticación de Firebase para validar las credenciales del usuario. Si esa validación es satisfactoria pasará a la siguiente pantalla. Además, cuenta con un botón de ingresar con una cuenta de Google el cual realizará la misma función descrita anteriormente, con la única salvedad de que utilizará los datos proporcionados por la cuenta de Google del usuario que la cree. Por último, se observa otro botón de registrarse el cual iniciará el flujo de registro de un nuevo usuario.

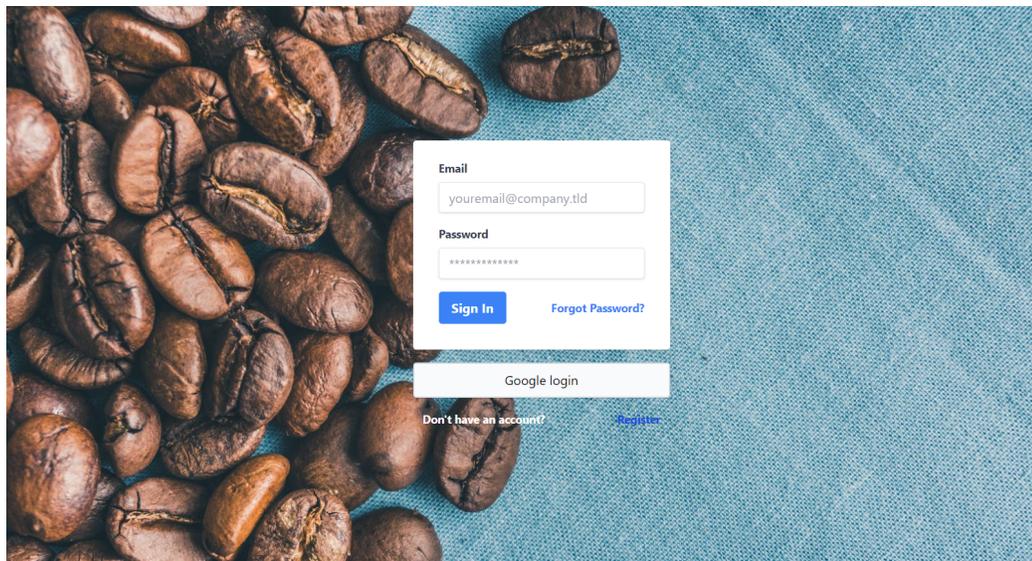


Figura 46: Inicio de sesión

La interfaz de la pantalla de registro se observa en la Figura 47. Esta pantalla cuenta con dos campos de texto: uno para ingresar el correo electrónico y otro para la contraseña. Contiene también un botón de Registrar el cual permitirá que Firebase asigne un UID (User ID o ID de usuario) al nuevo usuario. De esa manera se estará completando el flujo de Registro de un nuevo usuario. Por último, se observa otro bo-

tón de Login, el cual retornará a la pantalla de autenticación de usuarios previamente descrita.

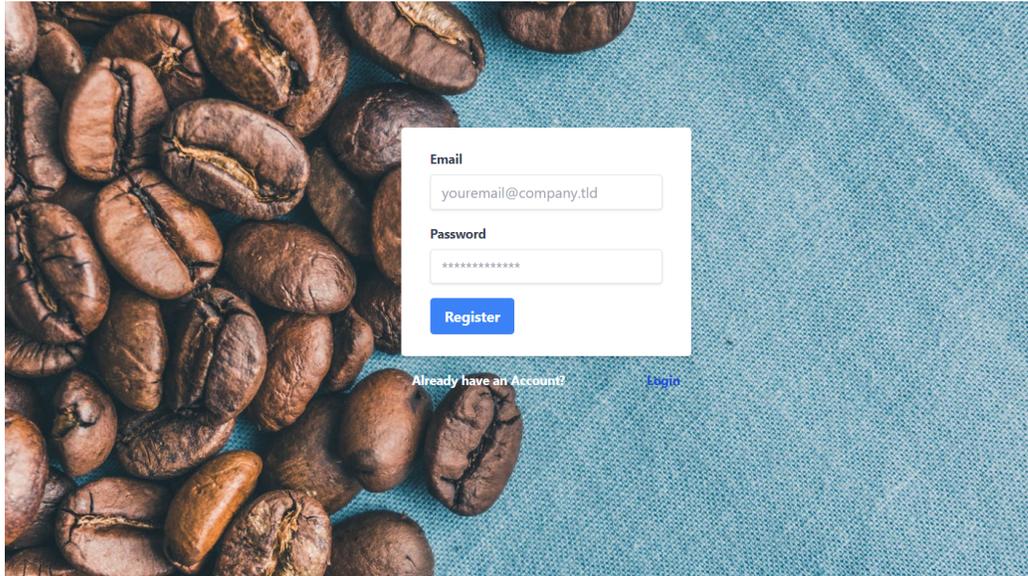


Figura 47: Flujo de registro de un nuevo usuario

9.2.2. Prototipo final de la pantalla principal

En la Figura 48 se observa la pantalla principal a la cual el usuario accederá si la validación resulto satisfactoria, en el ejemplo se accedió por medio de una cuenta de Google.

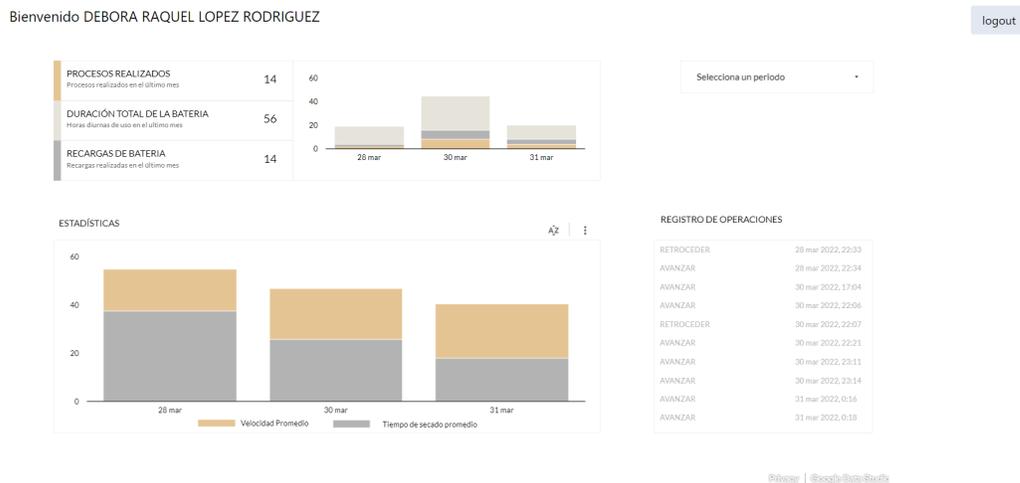


Figura 48: Prototipo de Dashboard

10.1. Funcionamiento

El robot móvil fue diseñado para ser autónomo en su funcionamiento. Sin embargo, es importante que el usuario encargado de la manipulación del robot pueda controlar el sistema de forma manual, de esta manera se tendrá un mayor control del proceso de secado. El control remoto posee la capacidad de controlar el avance, retroceso, los giros, la velocidad, las aspas y la activación del modo autónomo o manual. Además, se tiene como propósito que el usuario pueda tener a su disposición información valiosa acerca del robot en tiempo real, como podría ser el nivel de batería, tiempo total del proceso y modos de operación. El diseño del control remoto se desarrollará en los siguientes apartados.

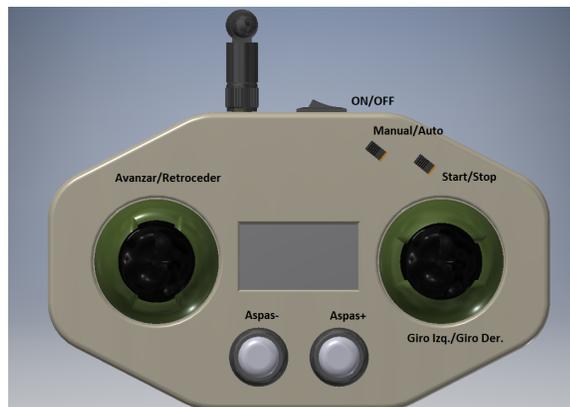


Figura 49: CAD del control remoto

10.2. Implementación

10.2.1. Hardware

En la Figura 50 se muestra el diagrama de bloques de los componentes electrónicos que forman al control remoto. De color gris se muestran las partes correspondientes a la alimentación energética del dispositivo. En verde se incluyen todos los pulsadores y switches, en azul se representa al módulo de radiofrecuencia que se encarga de comunicarse directamente con el robot móvil, en violeta la pantalla OLED y en naranja el microcontrolador que lee los estados de los pulsadores, switches y del módulo sensor de voltaje que se encarga de llevar el control del porcentaje de batería restante. Además, controla los datos de envío y recepción del módulo de radiofrecuencia y dicta lo que debe desplegarse en la pantalla OLED.

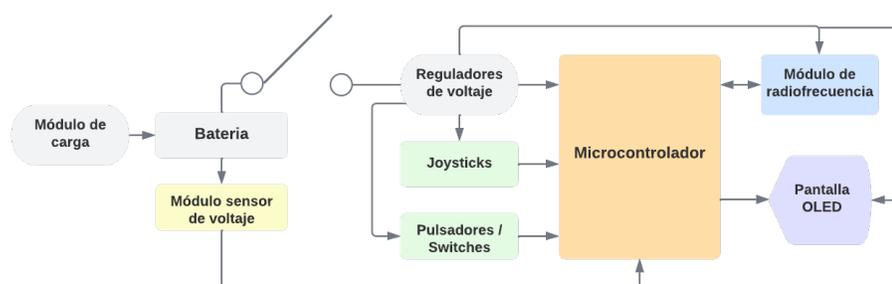


Figura 50: Diagrama de bloques control remoto

Como fuente de alimentación se utilizaron dos baterías 18650 de 3.7V. y 2550mAh. Se pusieron en serie para aumentar el voltaje a 7.4V. Esto se logra con un porta baterías especial para dichas baterías. El consumo energético es muy bajo; mucho menor a 100mA. Con la capacidad energética de 2550mAh y asumiendo un consumo de 100mA, la batería tiene una duración de aprox. 17.68 horas según la Ecuación de Peukert 2, que son un poco más de 2 días de trabajo de 8 horas. Además, se contempló una alarma para indicar cuando la batería se encuentre en el 10 % de su capacidad.

$$T = \frac{C}{I^n} = \frac{2550mAh}{100mA^{1.08}} = 17.64h \quad (2)$$

Sin embargo, para que el sistema fuera más eficiente se integró un módulo de carga en el circuito final. Luego en serie con la batería hay un interruptor que es el que enciende y apaga el control remoto. Luego le sigue el regulador de voltaje AMS1117-5.0V para convertir los 7.4V en 5V y posteriormente al regulador de voltaje AMS1117-3.3V para convertir los 5v en 3.3v, debido a que todos los componentes electrónicos utilizados se alimentan con 5V a excepción del módulo de radiofrecuencia.

10.2.2. Módulo NRF24L01

En el diseño original la comunicación entre el robot móvil y el control remoto se logra a través de los módulos NRF24L01. La razón de continuar con estos módulos es por la capacidad de la comunicación bidireccional, por el rango máximo de funcionamiento (1Km), por la capacidad de envío y recepción (32 bytes). Se utilizó un módulo tanto en el robot móvil como en el control remoto.

El NRF24L01 del control remoto le envía al robot la información de los estados de los joysticks y pulsadores. La información de cada uno de ellos se envía en forma de un bit, para un total de ocho bits comprimidos en un byte. Cuando se establece. También se envían 2 bytes dedicados a la verificación de buena comunicación con el control remoto. El byte de verificación inicial tiene un valor predeterminado en cada bit, dicho valor corresponde al carácter ASCII '<' y el byte de verificación Final tiene un valor predeterminado en cada bit, dicho valor corresponde al carácter ASCII '>'. Si el control remoto detecta ambos bytes en su respectivo orden, la comunicación entre ambos lados se considera exitosa y el robot está en disposición de operación. En la Tabla 4 se muestra que información contiene cada bit de los bytes mencionados.

	Descripción	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	Verificación inicial	0	0	1	1	1	1	0	0
2	Información joysticks y pulsadores	Modo Auto/CR	Start /Stop	X	X	Avanzar	Retroceder	Giro Izq.	Giro Der.
3	Aspas	X	Porcentaje de desplazamiento(%) [0...6]						
4	Velocidad	Velocidad variada(0-255) [0...7]							
5	Verificación final	0	0	1	1	1	1	1	0

Cuadro 4: Bytes enviados por el control remoto

En cambio, el robot móvil envía al control remoto la información respecto al nivel de batería, tiempo de operación en horas y minutos, estado de la comunicación remota y los estados actuales del robot. Al referirnos del modo de operación actual, se debe de considerar el movimiento en tiempo real como podría ser el avance, cambio de dirección, los diferentes giros, el movimiento de las aspas tanto si están arriba o abajo y la velocidad en tiempo real del robot.

Para el robot móvil también se envían 2 bytes dedicados a la verificación de buena comunicación con el control remoto. El byte de verificación inicial tiene un valor predeterminado en cada bit, dicho valor corresponde al carácter ASCII '<' y el byte de verificación Final tiene un valor predeterminado en cada bit, dicho valor corresponde al carácter ASCII '>'. El segundo byte se encarga de transportar la información del nivel de batería del robot, dicho porcentaje se traduce de decimal a binario y el bina-

rio es el que se envía en dicho byte. Como el valor máximo es 100 % el cual se traduce a 1100100, solo se requieren 7 bits para representar este número. El tercer y cuarto byte transportan la información del tiempo actual de operación del robot móvil, dónde el tercero contiene las horas y el cuarto los minutos. No obstante, se mantuvo el supuesto de que un día de trabajo en la finca no excederá 10 horas por lo que se habilitan 5 bits para un máximo de 32 horas. Una hora tiene sesenta minutos por lo que se requieren de 7 bits para representar este número. El quinto byte representa la información del tipo de movimiento en tiempo real que está realizando el robot móvil y el modo en el que se encuentra. El sexto byte corresponde al porcentaje de desplazamiento de las aspas, el cual sigue la misma lógica que el segundo byte. El séptimo byte correspondo a la velocidad en tiempo real del robot móvil. En la Tabla 5 se muestra con mayor detalle la información de cada bit de los bytes mencionados.

	Descripción	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	Verificación inicial	0	0	1	1	1	1	0	0
2	Batería	X	Nivel de batería(%)[0...6]						
3	Tiempo de operación	X	X	X	Horas [0...4]				
4	Tiempo de operación	X	Minutos [0...6]						
5	Información actual	Avanzar	Retroceder	Giro Izq	Giro Der.	X	X	X	Modo Au-to/CR
6	Aspas	X	Porcentaje de desplazamiento(%)[0...6]						
7	Velocidad	Velocidad variada(0-255)[0...7]							
8	Verificación final	0	0	1	1	1	1	1	0

Cuadro 5: Bytes enviados por el robot móvil

10.3. Diseño asistido por computadora para el control remoto físico

10.3.1. Estudio ergonómico

Asignación de funciones

Se ha escogido el uso de dos joysticks para las funciones principales del control, estos serán usados para todas las funciones de movimientos como lo son el avance, retroceso, cruzar a la derecha, cruzar a la izquierda y el movimiento de las aspas. El movimiento de las aspas en particular se colocará en función de los dos botones posicionados debajo de la pantalla OLED, derecho para elevarlas y el izquierdo para bajarlas dependiendo de la necesidad del usuario.

Se ha decidido colocar todas estas funciones en los joysticks para evitar que el control tenga una cantidad innecesaria de botones, esto con la finalidad de que el usuario pueda aprender a operar el robot en poco tiempo y que esta operación sea tan intuitiva como sea posible. Las diferentes direcciones de los joysticks estarán confirmadas por: joystick izquierdo será para el avance hacia arriba, retroceso abajo, joystick izquierdo será para girar a la izquierda y derecha con sus respectivas direcciones en el joystick.

Para escoger el modo automático o manual del robot se estará utilizando un switch de dos posiciones, siendo el modo de operación principal como la opción superior o a la derecha dependiendo de la orientación final de este componente. El modo principal que se colocará en esta posición será el automático, siendo el manual el que este abajo o del lado izquierdo, de nuevo, dependiendo de la orientación del switch.

Para el switch de start y stop, start será la opción superior y stop la inferior, este controlara si el robot se está moviendo o no en el modo automático y de forma similar, en una posición muy cercana a este switch tendremos el switch de apagado y encendido para el control, con encendido en la opción superior.

La posición de los botones fue escogida cuidadosamente para que el usuario, mientras tenga el control manual del robot habilitado pueda realizar la mayoría, si no es que todas las acciones sin tener que levantar los dedos pulgares de los joysticks y por esto mismo serán los controles más cercanos a la mano, más en el centro y arriba del control estarán los switches que serán la forma principal de escoger los diferentes modos de operación del robot, de izquierda a derecha tendremos el apagado y encendido del control, luego el modo del robot (automático o manual), luego tendremos el control de velocidad.

Dimensionamiento

Se realiza un estudio antropométrico y ergonómico de las partes relacionadas con la carcasa del control remoto y más concretamente de las manos, que estarán en permanente contacto con el control remoto. Con esto se pretende conseguir la máxima comodidad, seguridad y eficiencia a la hora de su sujeción y manejo de los diversos botones, joysticks y switches que lo constituyen.

Para el estudio, se contemplan datos antropométricos para la disposición de los componentes en la carcasa según la norma DIN 33 402.

Los diferentes componentes del control están separados en tres grupos, los joysticks y la pantalla serán el primer grupo ya que estarán aproximadamente a la misma altura, este grupo son los componentes principales y los cuales estarán lo más cerca posible de las manos y del usuario en general. El segundo grupo será los switches que manipularán los diferentes modos de operación tanto del control como del robot, estos estarán a la misma altura y estarán más lejos del usuario ya que no se debe interactuar tanto con ellos como con los Joysticks. Por último, el tercer grupo será

los botones que se encargaran de manipular las aspas del robot móvil, estos estarán a la misma altura y al igual que el segundo grupo se encontraran más lejos del usuario.

Siguiendo los percentiles de la Figura 5 la separación entre ambas manos no debería ser menor a aprox. 15cm, este valor, aunque arbitrario, fue seleccionado debido a que una de las razones de incomodidad de ciertos controles proviene de que estos sean muy pequeños, una distancia mayor ayudara al agarre del usuario y brindara espacio adicional para colocar los componentes cómodamente.

En cuanto a la distancia entre componentes, se reducirá en lo más posible la distancia entre los switches mientras estos no tengan problemas para ser accionados debido a la posición de la pantalla o joysticks, debido a que, al colocarlos más alejados del usuario, se está obligando al usuario a estirar los pulgares por encima de su capacidad según los percentiles de la Figura 4, para operar los switches. Por lo tanto, los espacios entre los grupos deben de seguir las recomendaciones antes mencionadas, para garantizar una buena interacción usuario/control remoto.

10.4. Pruebas y resultados

10.4.1. Hardware

Como se observa en la Figura 51 los resultados de la implementación de la parte electrónica del control remoto fueron exitosos. El sistema fue capaz de mover el prototipo a escala del robot móvil con la comunicación inalámbrica. Además, los comandos que rigen el movimiento del robot fueron recibidos correctamente por el prototipo a escala robot móvil. Por último, la información de la pantalla Oled fue desplegada y actualizada al momento de recibir cambios en los movimientos de los joysticks, switches y pulsadores. Por lo tanto, se considera el circuito electrónico del control remoto apto para controlar el robot móvil.



Figura 51: Prueba de funcionamiento de la pantalla Oled.

10.4.2. Diseño asistido por computadora para el control remoto físico

Para cada uno de los prototipos de los modelos 3D de la carcasa y los rieles del control remoto, se utilizó el programa de computación AUTODESK INVENTOR el cual permite diseñar, modelar e integrar componentes de manera fácil. Además, para esta aplicación en concreto permite exportar los diseños para posteriormente ser impreso en 3D. Dichas impresiones fueron fabricadas a base de PLA (ácido poliláctico).

A continuación, se muestran las distintas propuestas estudiadas, así como las ventajas e inconvenientes de cada una y la solución seleccionada teniendo en cuenta los procesos empleados y la normativa aplicable.

Prototipo No.1

Carcasa del control remoto:



Figura 52: Vista frontal para la carcasa



Figura 53: Vista trasera para la carcasa

Una vez analizada la carcasa, se concluye que la falta de simetría en el diseño de

la carcasa presento desventajas con respecto al agarre de la pieza.

Riel para joysticks:

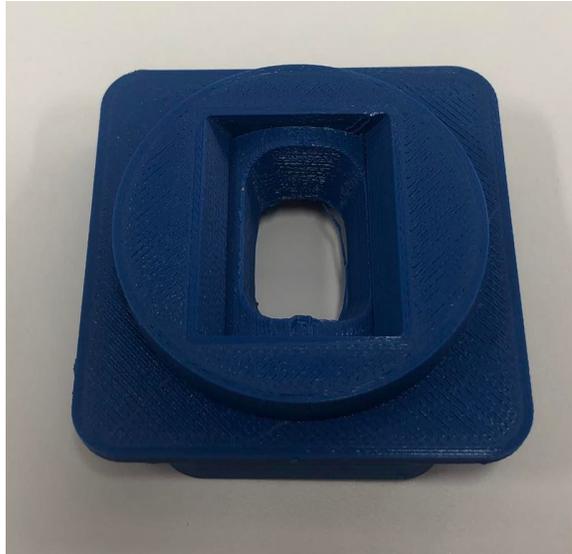


Figura 54: Vista frontal de riel para joystick

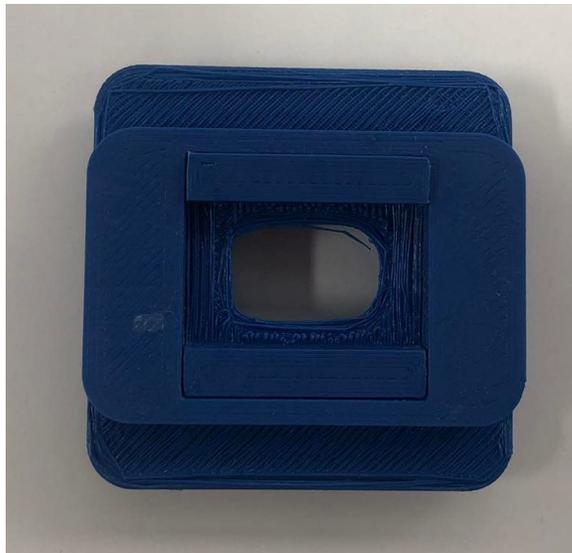


Figura 55: Vista trasera de riel para joystick

Una vez analizado el riel el cual está encargado de limitar el movimiento del joystick, se concluye que el mecanismo basado en dos piezas ocupa demasiado espacio tanto en el interior como en el exterior del control remoto, por lo que resulta obsoleto para esta aplicación.

Prototipo No.2

Carcasa del control remoto:



Figura 56: Vista frontal para la carcasa



Figura 57: Vista trasera para la carcasa

Una vez corregida la asimetría del prototipo No.1, se confirmó una mejora con respecto al agarre de la pieza. Sin embargo, se presentó una desventaja con respecto a la localización de los botones, debido a que se sobrepasaron los percentiles de la Figura 4 causando incomodidad.

Riel para joysticks:



Figura 58: Vista frontal de riel para joystick 1



Figura 59: Vista frontal de riel para joystick 2

Para el riel del segundo prototipo se cambió drásticamente la geometría del diseño. Sin embargo, se sigue presentando la misma desventaja del prototipo No.1 debido a que la pieza aun ocupa un espacio significativo en el interior del control remoto, por lo que aún resulta obsoleto para esta aplicación.

Prototipo No.3

Carcasa del control remoto:

A través de las pruebas físicas se observaron las ventajas de la creación de un mando con dos joysticks, como la mayoría que se fabrican actualmente, ya que con esto se aprovechan las grandes posibilidades de movimiento y precisión que tiene el pulgar en las manos del ser humano.



Figura 60: Vista frontal para la carcasa

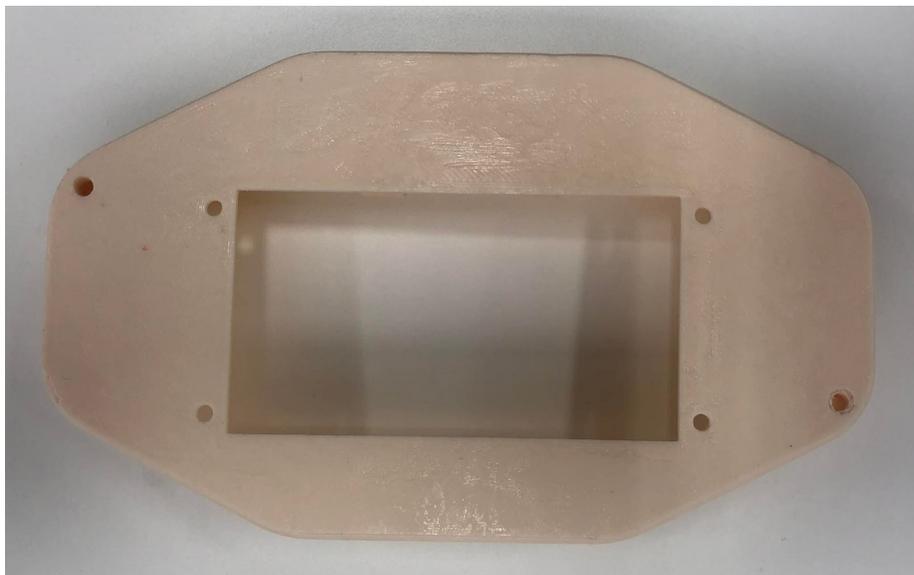


Figura 61: Vista trasera para la carcasa 1

Además de esto se trató de que las curvaturas, localización de los botones y demás elementos no obliguen a ejercer una postura forzada de las manos que pueda crear incomodidades al usuario.



Figura 62: Vista trasera para la carcasa 2



Figura 63: Vista trasera para la carcasa 3

Todo lo antes mencionado contribuyó a obtener resultados satisfactorios en el estudio ergonómico, así como de los requisitos de diseño analizados en los prototipos anteriores. Por lo tanto, se propone como diseño definitivo de la carcasa el correspondiente al prototipo No.3.

Riel para joysticks:



Figura 64: Vista frontal de riel para joystick



Figura 65: Riel ensamblado en el control remoto

Tomando los resultados de los prototipos anteriores, se tomó la decisión de diseñar un riel que se adaptara a la forma del joystick, lo cual dio como resultado una pieza que no ocupa espacio en el interior del control remoto. Además, el estudio ergonómico no reflejó incomodidad al usuario al momento de operar el robot móvil.

10.4.3. Módulo NRF24L01

Como se observa en la Figura 66 la prueba de comunicación entre el módulo NRF24L01 en el ATMEGA328P y el módulo NRF24L01 en el Arduino Due fue satisfactoria.

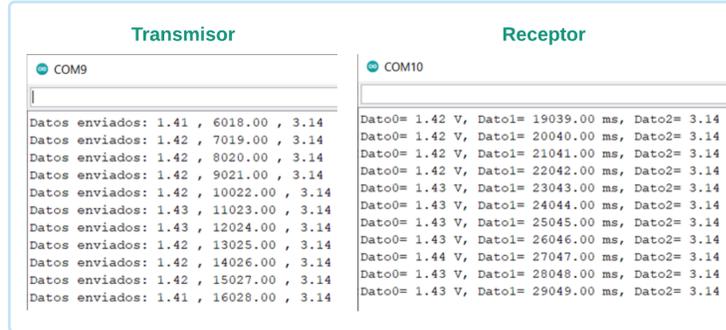


Figura 66: Prueba de transmisión y recepción

Listado de materiales electrónicos completo y servicios

En la Tabla 6 se muestra el listado completo de todos los componentes electrónicos utilizados en el control remoto y en el sistema de recopilación de datos GSM/GPRS. Se muestra el lugar de adquisición, la cantidad de piezas utilizadas, el precio unitario y el total.

La Tabla 7 muestra los precios de los servicios de Firebase dependiendo del plan que se contrate, sin embargo, el plan Spark es suficiente para cubrir las necesidades del proyecto.

Por lo tanto, el monto total de la compra de los componentes y de los servicios de Firebase, tendría un costo de (sin incluir costes de envío): Total = **Q949.04**.

Componente	Lugar de compra	Cantidad	Precio	Sub-total
Arduino DUE ARM Cortex-M3 + Cable USB	Electrónica DIY	1	Q252.00	Q252.00
Tranceptor RF 2.4GHZ NRF24L01 Antena Externa	Electrónica DIY	2	Q51.00	Q102.00
Joystick 5 pines	Electrónica DIY	2	Q21.00	Q42.00
Integrado Atmega328-PU 28 Pins	Electrónica DIY	1	Q34.00	Q34.00
Base para circuito integrado ancho 28 Pines	Electrónica DIY	1	Q2.50	Q2.50

Componente	Lugar de compra	Cantidad	Precio	Sub-total
Osciladores de cristal 16MHZ	Electrónica DIY	1	Q4.00	Q4.00
Capacitores cerámicos picofaradios 22pf	Electrónica DIY	4	Q0.50	Q2.00
Capacitores electrolíticos 50V 0.1uF	Electrónica DIY	2	Q0.90	Q1.80
Porta batería 2x 18650 7.4V	Electrónica DIY	1	Q12.00	Q12.00
Batería EVE 26V 18650 2550mAh 7.5A	Electrónica DIY	2	Q43.00	Q86.00
Cargador de baterías de litio con elevador de voltaje ajustable	Electrónica DIY	1	Q25.00	Q25.00
Shield GSM GPRS SIM900	Electrónica DIY	1	Q264.00	Q264.00
REGULADOR DE VOLTAJE DE 5.0V AMS1117-5.0	La electrónica GT	1	Q3.00	Q3.00
REGULADOR DE VOLTAJE DE 3.3V AMS1117-3.3	La electrónica GT	1	Q3.00	Q3.00
Pantalla Oled LCD 128x64 1.3"SH1106 I2C para Arduino	OXDEA GT	1	Q87.30	Q87.30
Capacitor electrolitico 0.33uf 50V	OXDEA GT	1	Q1.94	Q1.94
Mini switch negro 2 posiciones 250V 3A-125V 6A	OXDEA GT	1	Q3.50	Q3.50
Micro switch deslizable de 1 polo, 2 tiros, 2 posiciones	OXDEA GT	1	Q3.00	Q3.00
Switch push botón N/A grande color rojo	OXDEA GT	2	Q10.00	Q20.00
Total componentes electrónicos				Q949.04

Cuadro 6: BOM completo del control remoto y del sistema de recopilación de datos GSM/GPRS

Producto	Sin costo - Plan Spark	Prepago - Plan Blaze
Authentication	Sin costo	Sin costo
Hosting		
Almacenamiento	10 GB	\$0.026 por GB

Producto	Sin costo - Plan Spark	Prepago - Plan Blaze
Transferencia de datos	360 MB por día	\$0.15 por GB
Dominio personaliza- do y SSL	sin costo	sin costo
Varios sitios por pro- yecto	sin costo	sin costo
Plan Spark		Q0.00

Cuadro 7: Servicio de Authentication y Hosting de Firebase

1. Al implementar la propuesta a la solución para la transmisión inalámbrica, la comunicación móvil usando paquete de datos (GPRS) ofreció una buena respuesta a la hora de operar el sistema de monitoreo, ya que al enviar la data desde el recinto donde se está controlando el equipo, al servidor externo (Adafruit IO), la información es recibida en intervalos menores a 1 minuto. Además, al no tener retardos significativos en la lectura remota de las variables, se garantiza una pronta detección de una posible emergencia.
2. Se diseñó un control remoto capaz de comunicarse con éxito al prototipo a escala del robot móvil.
3. La parte electrónica del control remoto, del sistema de recopilación de datos GSM/GPRS y de los servicios de Firebase tiene un costo aproximado de Q949.04
4. Se implementó un servidor mqtt, capaz de gestionar las publicaciones y suscripciones de cada uno de los clientes, permitiendo que el sistema sea capaz de analizar los datos enviados por el robot móvil.
5. Se desarrolló un sistema compuesto por dos aplicaciones web y una móvil, capaz de visualizar y retroalimentar los datos enviados por el robot móvil a través del servidor mqtt. Cabe destacar que gracias al uso de servidores externos se obtienen muchas facilidades a la hora de almacenar data y generar una interfaz visual que sea interactiva con el usuario u operador del sistema.

1. Se recomienda utilizar los servicios de telefonía celular de la compañía que tenga mayor cobertura en la localidad de la Finca El Injerto.
2. Se recomienda soldar un capacitor electrolítico de 47uF en el VCC-GND del NRF24L01 para evitar que los picos de corriente interfieran que podrían surgir la comunicación por radiofrecuencia.
3. Se recomienda utilizar un módulo joystick que físicamente abarque todo el rango del potenciómetro para que el aumento gradual de velocidad del robot móvil no sea tan brusco.
4. MQTT es un protocolo sumamente versátil, que presenta una muy buena opción a la hora de realizar comunicación entre dispositivos enviando mensajes sumamente pequeños, pero en un ambiente profesional es necesario implementar la autenticación con HiveMQ MQTT Broker, ya que cualquier usuario que conozca la IP del host puede suscribirse a un tópico y recibir mensajes.
5. Se sugiere seguir el siguiente paradigma, tanto de programación como de diseño para la aplicación web y móvil: Diseñar un mockup como guía visual para el programador, dejando en claro la solución. Luego se debe programar el fronted con la tecnología React.js y React native. Por último se debe de enlazar el backend diseñado den Node.js con el frontend.
6. Se sugiere agregar un botón en la pantalla de dashboard en donde se descargue la información filtrada por data studio siguiendo los siguientes pasos: se debe de instalar la librería react-csv, luego en el componente `onClick=` del botón, se debe de añadir un enlace de descarga CSV utilizando el componente `CSVLink` de la librería react-csv. Luego, se debe de generar un array anidado como valor de la data prop para que react-csv pueda generar el CSV a partir de él. Por último, se debe de crear el array anidado llamando a `useMemo()` con un callback que devuelve el array anidado que se creó llamando a `csvData.map()`

con un callback que devuelve los valores en un array de cada entrada de csvData con Object.values(). Dentro del csvData definiremos los datos que deseamos exportar.

7. Se recomienda eliminar al cliente Adafruit IO, por lo que el Shield SIM900 GSM/GPRS realizar la solicitud POST directamente a la base de datos Mysql. Por lo tanto, el cliente Mysql se encargará de realizar las publicaciones al Servido MQTT y su único suscriptor será la aplicación móvil. En donde solo se estarán publicando los valores de velocidad y batería restante del robot móvil. Sin embargo, el cliente Mysql seguirá siendo suscriptor de la aplicación móvil para continuar recibiendo los ingresos de una nueva recarga de batería.
8. Se recomienda evaluar la posibilidad de cambiar una base de datos Mysql por la Cloud Firestore que ofrece firebase, tomando en cuenta que para este proyecto se utilizo phpMyAdmin proporcionado por db4free.net para la gestión de la base de datos MySQL que proporciona gratuitamente 200 MB de almacenamiento. Sin embargo, es una plataforma gratuita para estudiantes y no es recomendada para usos en la industria. Entonces se debería de adquirir un servicio de nube para poner administrar la base de datos Mysql. Por lo tanto, los aspectos a tomar en cuenta son: La diferencia entre Firebase y MySQL se reduce a cómo una empresa maneja y administra sus datos. Firebase maneja de manera efectiva grandes conjuntos de datos. Utiliza columnas anchas, valores clave, almacenes de documentos o gráficos y tiene esquemas dinámicos para facilitar los datos no estructurados. Por otro lado, MySQL está basado en tablas y tiene esquemas predefinidos. Es una opción preferida para manejar datos complejos. Además, comparado con Firebase, MySQL es mejor para transacciones de varias filas. Por otro lado, Firebase es una opción satisfactoria cuando se trata de administrar grandes conjuntos de datos porque NoSQL escala los datos horizontalmente y es mucho más rápido que MySQL que escala verticalmente. Lo cual sería beneficioso si se requiriera aumentar los datos recopilados por el sistema de monitoreo. Por último, el precio por GB almacenado en Cloud Firestore es de \$0.108 por GB [21] y si lo comparamos con Google Cloud que administra bases de datos Mysql es de \$0.170 por GB [22].

- [1] M. H. Floto y D. H. W. Griepentrog. (2019). “Field Robot Event.” Accedido en mayo de 2021, dirección: <https://www.fieldrobot.com/event/>.
- [2] N. Muñoz, N. Opina, H. Giraldo y J. Eusse. (2006). “Aplicación de gprs para arquitecturas de control de robots.” Accedido en mayo de 2021, dirección: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/6463>.
- [3] F. Juárez. (2018). “El café guatemalteco:Un enfoque en el mercado mundial y su productividad.” Accedido en mayo de 2021, dirección: <http://camcig.org/userfiles/2019/01/2018.-DICE-El-caf%C3%A9-guatemalteco-un-enfoque-en-el-mercado-mundial-y-su-productividad-1.pdf>.
- [4] K. Neyra, E. Ordinola y K. Siancas, “Prototipo de sistema de monitoreo y control de consumo de agua potable aplicando tecnología GSM,” Tesis de mtría., Universidad nacional de Piura, 2019.
- [5] P. Badillo y L. Gonzalo, “Diseño e implementación de un sistema electrónico para conducción segura, basado en monitorización de señales biométricas, utilizando tecnologías GSM, GPRS.” Tesis de mtría., Escuela Superior Politécnica de Chimborazo, 2017.
- [6] INSHT. (2018). “Manual de normas técnicas para el diseño ergonómico de puestos con pantallas de visualización (2ª Edición).” Accedido en mayo de 2021, dirección: <https://www.insst.es/documents/94886/96076/Manual+de+normas+t%C3%A9cnicas+para+el+dise%C3%B1o+ergon%C3%B3mico+de+puestos+con+pantallas+de+visualizaci%C3%B3n.pdf/b4818262-f8ba-4ddd-9c49-9e7d6ea4ce62?t=1543110942922>.
- [7] J. Melo. (2013). “Antropometría.” Accedido en mayo de 2021, dirección: https://sistemamid.com/panel/uploads/biblioteca/2014-09-09_10-10-29109608.pdf.
- [8] SIMCom. (2010). “SIM900 Hardware Desing.” Accedido en mayo de 2021, dirección: https://files.seeedstudio.com/wiki/GPRS_Shield_V3.0/res/SIM900_HD_V1.05.pdf.

- [9] B. Rubell. (2021). “Getting Started with Adafruit IO.” Accedido en mayo de 2021, dirección: <https://learn.adafruit.com/welcome-to-adafruit-io/getting-started-with-adafruit-io>.
- [10] T. H. Team. (2015). “Introducing the MQTT Protocol - MQTT Essentials.” Accedido en mayo de 2021, dirección: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>.
- [11] J. A. L. Rufián, “Esteganografía sobre protocolos de red/internet,” Tesis de mtría., Universidad de Jaén, 2017.
- [12] T. Sufiyan. (2021). “What is ReactJS: Introduction To React and Its Features.” Accedido en mayo de 2021, dirección: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>.
- [13] S. INDIA. (2019). “React Native vs. Ionic vs. Flutter: Comparison of Top Cross-Platform App Development Tools.” Accedido en mayo de 2021, dirección: <https://codeburst.io/react-native-vs-ionic-vs-flutter-comparison-of-top-cross-platform-app-development-tools-71c8011309ac>.
- [14] Node.js. (2021). “Introduction to Node.js.” Accedido en mayo de 2021, dirección: <https://nodejs.dev/learn>.
- [15] Google. (2021). “Firebase.” Accedido en mayo de 2021, dirección: <https://firebase.google.com/docs?hl=es-419>.
- [16] Arduino. (2021). “Arduini DUE-Overview.” Accedido en mayo de 2021, dirección: <https://store.arduino.cc/arduino-due>.
- [17] components101. (2018). “nRF24L01 Wireless RF Module.” Accedido en mayo de 2021, dirección: <https://components101.com/wireless/nrf24l01-pinout-features-datasheet>.
- [18] D. Small. (2014). “Tiempo de descarga de batería dependiendo de la carga.” Accedido en mayo de 2021, dirección: <https://es.planetcalc.com/2283/>.
- [19] (10 de abr. de 2020). “Calcular el tiempo de uso de la batería,” Tritec Intervento, dirección: <https://tritec-intervento.cl/como-calcular-el-tiempo-de-uso-de-la-bateria/> (visitado 26-05-2022).
- [20] LastMinuteEngineers. (2020). “Send Receive SMS and Call with SIM900 GSM Shield and Arduino.” Accedido en mayo de 2021, dirección: <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/>.
- [21] firebase. (2022). “Planes de precios.” Accedido en junio de 2022, dirección: <https://firebase.google.com/pricing>.
- [22] Google. (2022). “Precios de MySQL y PostgreSQL.” Accedido en junio de 2022, dirección: <https://cloud.google.com/sql/pricing#mysql-pg-pricing>.

15.1. Esquemáticos de los circuitos

15.1.1. Circuito del control remoto

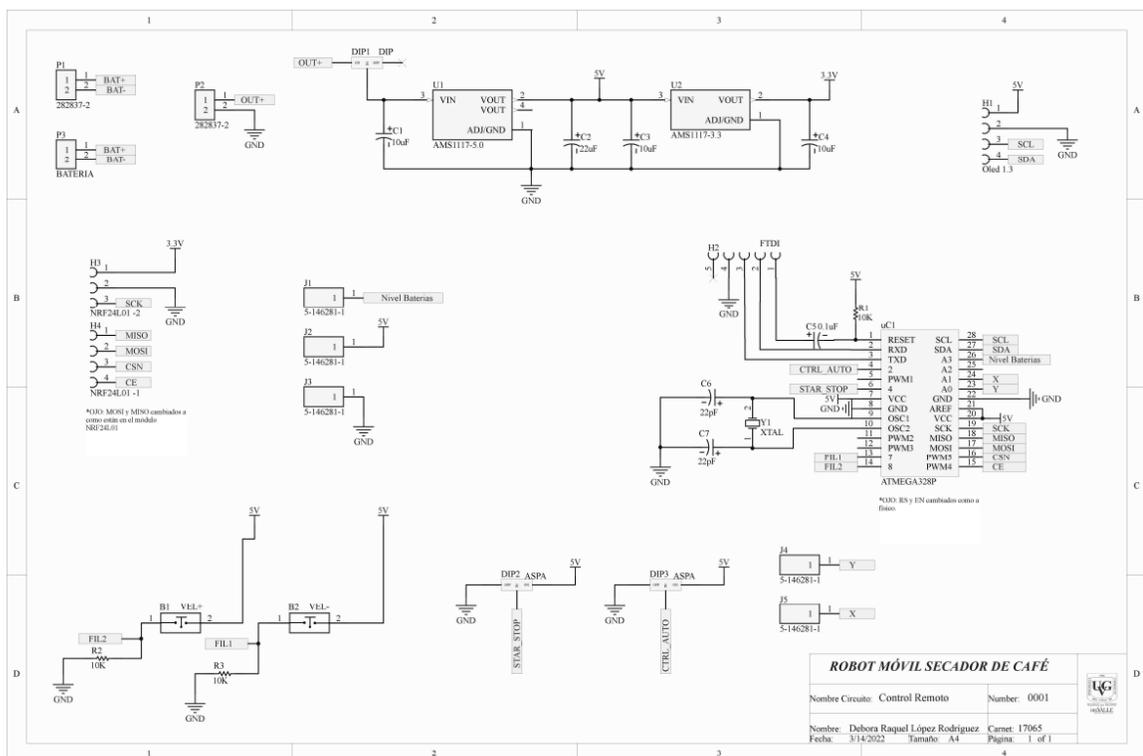


Figura 67: Esquemático circuito control remoto

15.2. PCBs

15.2.1. PCB control remoto

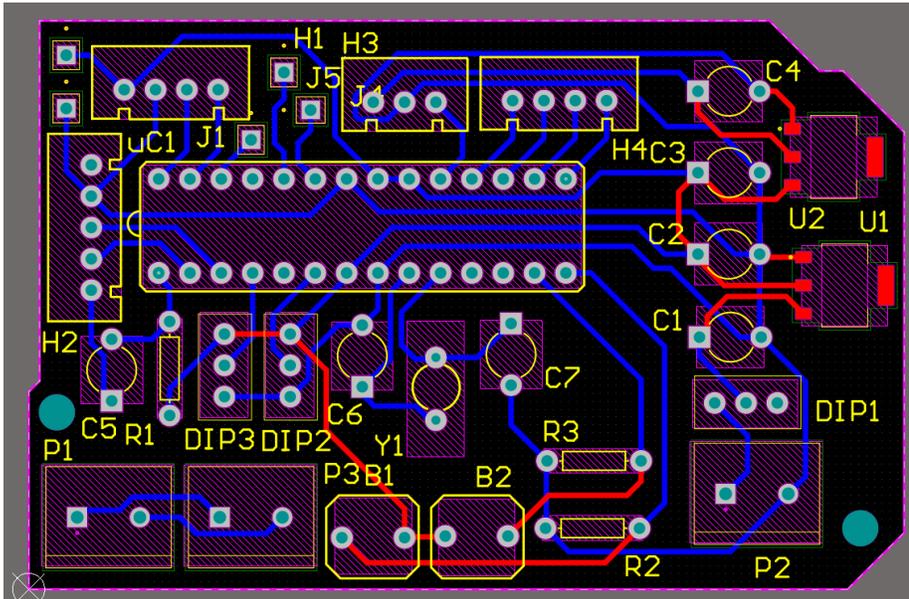


Figura 68: PCB circuito control remoto

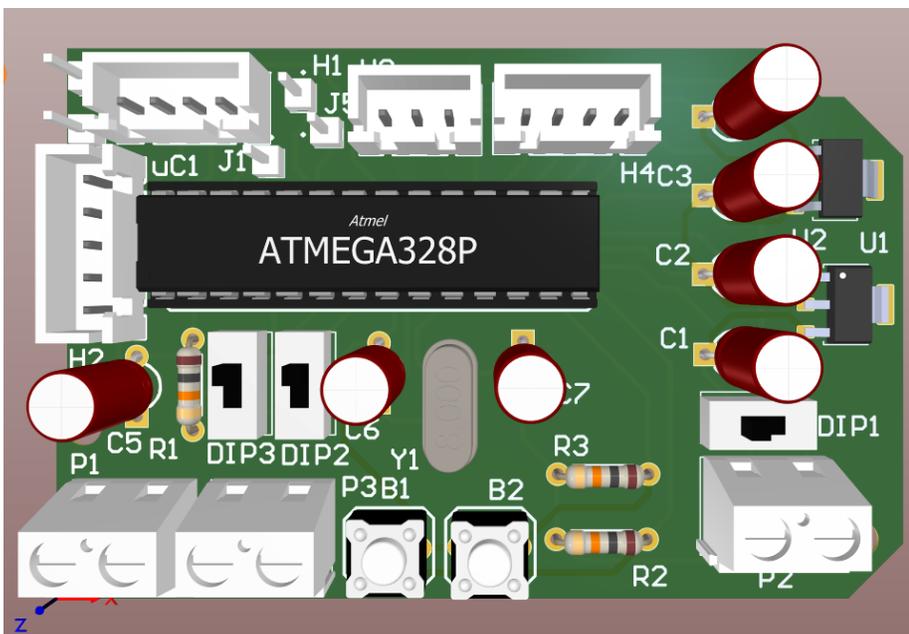


Figura 69: Representación 3D del circuito control remoto

15.3. Mockups

15.3.1. Aplicación móvil

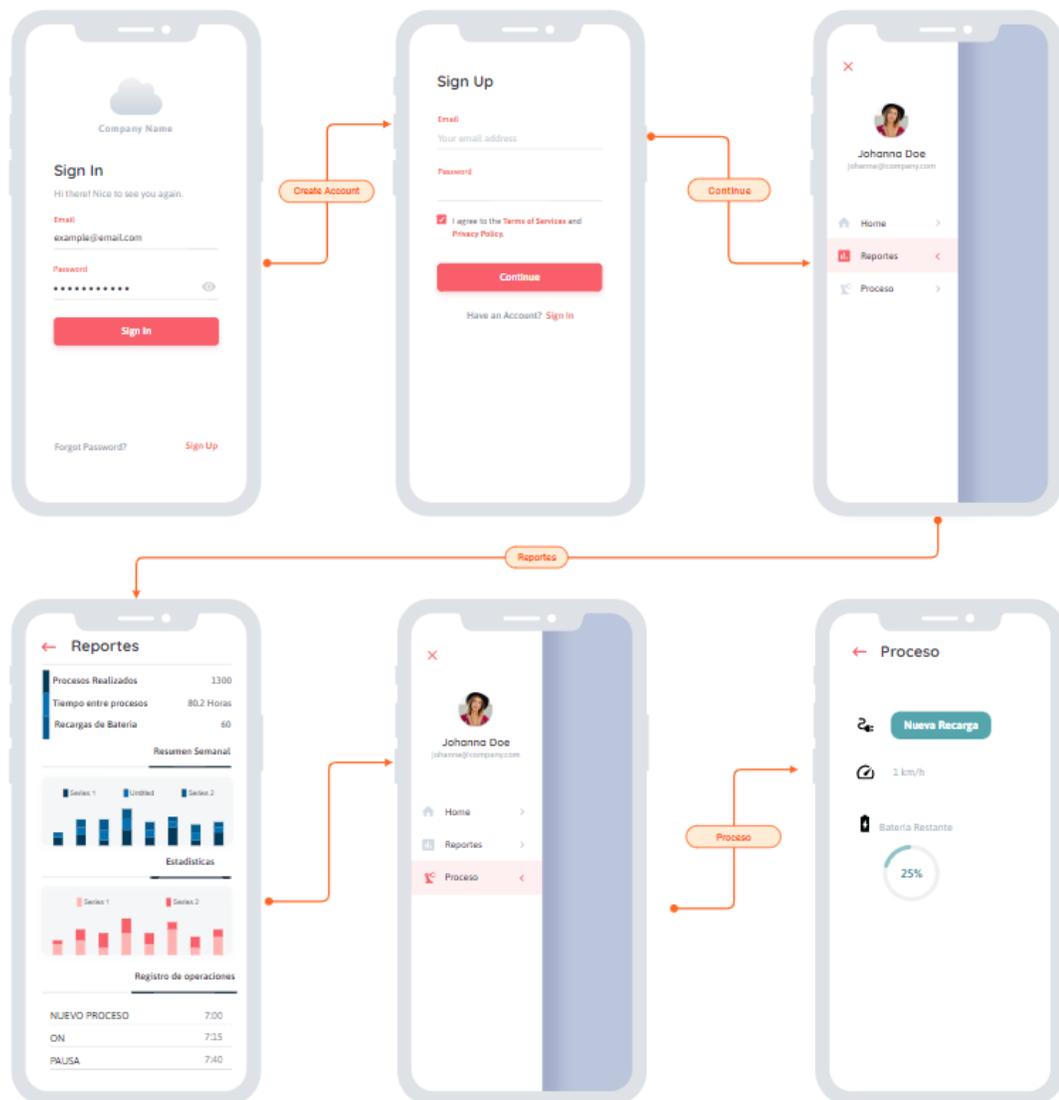


Figura 70: Mockup de aplicación móvil

15.3.2. Aplicación web

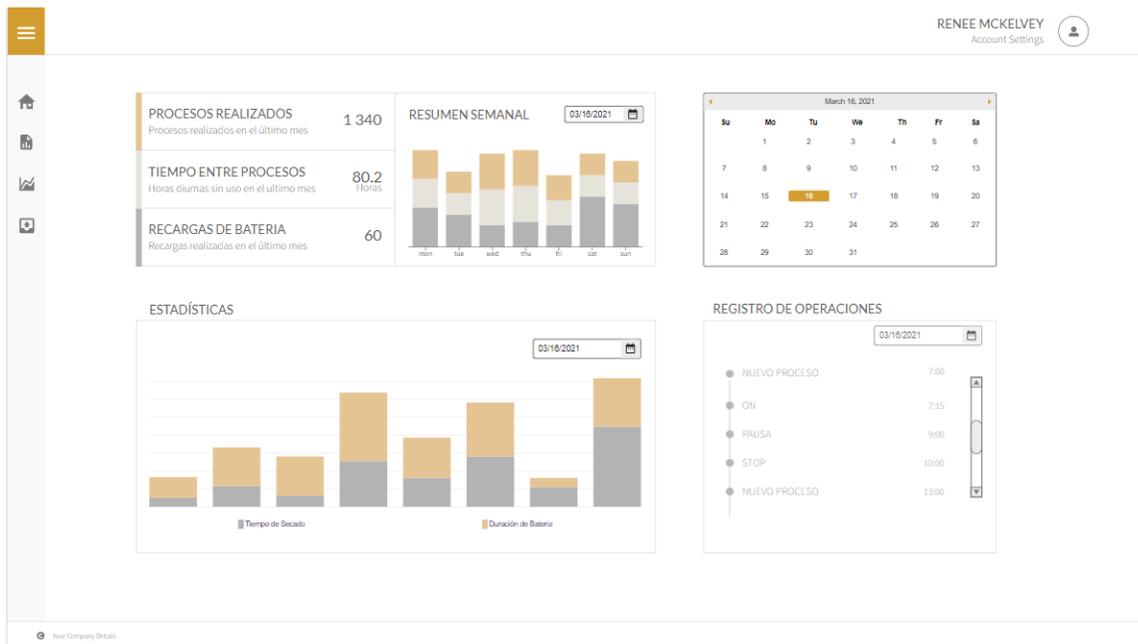


Figura 71: Mockup de la aplicación web

APN (Access Point Name): Es el nombre de punto de acceso para GPRS o estándares posteriores (cómo 3G y 4G) que debe configurarse en el dispositivo celular (bien sea un teléfono móvil, G2K8, o un módem USB), para que pueda acceder a redes computacionales (entre las que se puede incluir Internet).

Un punto de acceso es:

- Una dirección IP a la cual un móvil se puede conectar.
- Un punto de configuración que es usado para esa conexión.
- Una opción particular que se configura en un teléfono móvil.

. 32

SDK: Es el acrónimo de “Software Development Kit” (Kit de desarrollo de software). Es un kit de desarrollo de software que reúne un grupo de herramientas que permiten la programación de sistemas, aplicaciones tanto de escritorio, web, y móviles. . 38