Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología - ESIT

Máster Universitario en Industria 4.0

Implementación de un Sistema de Gestión y Monitoreo de medicamentos basado en IoT

Trabajo fin de estudio presentado por:	Ximena Elizabeth Sigcha Guachamin
Tipo de trabajo:	Tipo 2. Planteamiento de un proyecto de Industria 4.0
Director/a:	Manuel Pedro Blazquez Merino
Fecha:	20/07/2022

Resumen

Este proyecto está orientado a solucionar el problema que se genera a partir de la propensión a olvidar tomar los medicamentos por parte de las personas medicadas, situación que puede repercutir negativamente en la salud de los pacientes e interfiriendo en el tratamiento, y en muchos casos perjudicando la salud de las personas. Por esta razón, en este trabajo se ha propuesto el desarrollo de un prototipo de caja de medicamentos inteligente. El prototipo está equipado con sensores que permiten saber si los medicamentos han sido recogidos de la caja en su respectivo horario. Además, este prototipo fue equipado con alertas de alarma visual y sonora, implementadas mediante leds y un zumbador, dichas alarmas pueden ser programadas por los médicos a través de una página web, que permite gestionar, y monitorizar el tratamiento farmacéutico de cada paciente en tiempo real con el objetivo de mejorar el seguimiento remoto y automatizado.

Esta misma información puede ser visualizada por el usuario desde una aplicación Android.

Toda esta información puede ser leída y compartida debido a que se guarda en una base de datos alojada en la nube de Firebase, creando un proyecto orientado al entorno IoT.

Palabras clave: Firebase, Sistema de gestión, IoT, Base de datos, sistema de monitorización

Abstract

This project is aimed at solving a daily problem in people's health, the tendency to forget to take medications, a situation that can have a negative impact on the health of patients and interfere with treatment, and in many cases harming the health of the people. For this reason, in this work the development of a prototype of a smart medicine box has been proposed. The prototype is equipped with sensors that allow knowing if the medicines have been collected from the box at their respective times. In addition, this prototype was equipped with visual and sound alarm alerts, implemented by means of LEDs and a buzzer, these alarms can be programmed by doctors through a web page, which allows them to manage and monitor the pharmaceutical treatment of each patient in time. real with the aim of improving remote and automated monitoring.

This same information can be viewed by the user from an Android application.

All this information can be read and shared because it is stored in a database hosted in the Firebase cloud, creating a project oriented to the IoT environment

Keywords: Firebase, Management system, IoT, Database, monitoring

Índice de contenidos

Índice de figuras

Índice de tablas

Tabla de acrónimos

Sigla	Significado
UID	Número de identificación en Firebase authentication
IDE	Entorno de desarrollo integrado
HTTP	Protocolo de transferencia de hipertextos
HTML	Lenguaje de Marcas de Hipertexto
OMS	Organización Mundial de la Salud
RFID	Identificación por radiofrecuencia
LED	Diodo emisor de luz
MQTT	*Message Queuing Telemetry Transport*
IOT	Internet de las cosas
IR	Sensor Infrarrojo
NTP	*Network Time Protocol*

1. Introducción

El desarrollo de la tecnología, las redes y las comunicaciones, ha permitido que el ser humano pueda estar conectado al internet ininterrumpidamente, es decir el internet se ha hecho parte de la vida cotidiana de las personas, tanto a nivel personal como en la industria.

El internet de las cosas (IoT) es la conexión de dispositivos y sensores con el internet, en el campo de la medicina tiene el propósito solucionar y facilitar los servicios de atención, seguimiento, diagnósticos en los pacientes, incluso al añadir manejo de datos e inteligencia artificial se puede llegar a predecir enfermedades en las personas.

Entre las aplicaciones del IoT en la medicina están: La telemedicina, la emergencia inteligente, la medicación inteligente, las redes sociales de salud, la monitorización, la medición de signos vitales y el seguimiento de pacientes.

En el presente trabajo se desarrollará un dispositivo IoT aplicado a la medicina para solucionar el problema que tienen las personas al olvidar tomarse los medicamentos prescritos por un médico, interfiriendo en el tratamiento, y en muchos casos perjudicando el tratamiento y la salud de las personas.

1.1. Motivación

Para la mayoría de los tratamientos médicos, se prescriben medicamentos, con la finalidad de que el paciente mejore su estado de salud. Sin embargo, fuera del consultorio médico no existe un seguimiento del tratamiento farmacéutico, es decir cada persona es responsable de tomar su medicamento sin ningún tipo de supervisión.

De hecho, hay personas que tienen la tendencia a olvidar tomar sus medicamentos en la hora o de la forma correcta y esto afectando su salud, entre las causas comunes están los problemas de memoria o de visión (Rushikesh J., 2019). Otro grupo de personas no terminan el tratamiento completo debido a que dejan de sentir dolores e incomodidades relacionadas con su salud y suspenden su tratamiento sin terminarlo.

Por lo general, las personas que consumen medicamentos con mayor frecuencia son las personas son ancianas y frecuentemente estos sujetos necesitan ayuda de familiares o amigos debido al deterioro cognitivo producido por el envejecimiento. Por lo tanto, la propuesta de este trabajo es crear un sistema de monitoreo, que permita vigilar de manera automática si el usuario, toma sus medicamentos, mediante dispositivos y sensores basados en IoT. Además, se propone la creación de un sistema de gestión que permite generar recordatorios y alarmas para que el usuario tome correctamente sus medicamentos con el fin de ayudar a mejorar el régimen de tratamientos mediante el consumo adecuado la medicación.

1.2. Planteamiento del trabajo

Según la Organización Mundial de la Salud (OMS), el uso racional de los medicamentos implica que los pacientes reciban el medicamento correcto, en tiempo y dosis adecuadas, y con el menor costo posible.

Si no se hace un uso racional de los medicamentos se puede producir consecuencias negativas como efectos secundarios, perdida de eficacia, desgaste de la salud e incluso la muerte del paciente. (Valdés, 2021).

Las constantes distracciones de la vida cotidiana, la cantidad de medicamentos, pérdida de memoria o algún tipo de discapacidad son factores para que las personas olviden tomar sus medicamentos a tiempo y de forma incorrecta, especialmente en la población anciana.

Por lo tanto, la propuesta de este trabajo es implementar un servicio de medicación inteligente, el cual tiene dos bloques. El primer bloque es el de la gestión de información recolectada, con la finalidad de monitorear la evolución del tratamiento médico, evitando que las personas abandonen y olviden el tratamiento. Y el segundo bloque consiste en la implementación de una caja de medicamentos inteligente, que permitirá realizar el monitoreo del paciente y la toma adecuada de sus medicamentos mediante sensores y dispositivos IoT.

1.3. Estructura de capítulos

En el presente proyecto se ha dividido el trabajo en 5 capítulos:

- En el primer capítulo se describe una breve introducción, la motivación del proyecto y el planteamiento de una posible solución al problema.
- El segundo capítulo hace referencia al proceso de investigación del proyecto, trabajos similares, para empaparnos y entrar en el contexto actual del proyecto.
- En el capítulo 3, se describe una propuesta para solucionar el problema planteado en este proyecto, partiendo de los objetivos, alcance y limitaciones, además se plantea el itinerario del proyecto, la metodología, y la arquitectura de este.
- En el capítulo 4, se describe el desarrollo de la solución y pruebas del proyecto siguiendo el cronograma y la metodología de trabajo propuesto.
- En el capítulo 5, se realiza un análisis de todo el trabajo y se presentan las conclusiones a las que se ha llegado al término de este el trabajo.

2. Contexto y estado del arte

En este capítulo se describirá proyectos y tecnologías aplicados con anterioridad para solucionar el problema que tienen las personas al olvidar tomar sus medicamentos.

2.1. Descripción general del contexto del proyecto

El proyecto se encuentra inmerso en los sistemas IoT orientados al sector de la salud, estos proyectos por lo general buscan hacer un seguimiento de la salud del paciente sin la necesidad de contar con el personal médico de forma presencial, esto funciona a través del uso de sensores y actuadores que permiten monitorizar a los pacientes. Estas tecnologías los conectan con el personal médico a través de las redes, manteniendo estrecha comunicación entre un médico y paciente.

Por esta razón, se han desarrollado varios proyectos y plataformas para el cuidado de la salud, por ejemplo, en el artículo de Singh, B (2017) menciona una Plataforma *health IoT basada en intelligence system*, es una plataforma completa para la salud inteligente en el hogar, está dividido en tres componentes que permiten el monitoreo del paciente con sensores inteligentes, los componentes son *iMedPAck, iMedBox y BioPatch*.

iMedBox es una caja de medicamentos en donde existe la conexión con imedPack que son blíster de medicina con sensores RFID y con *bioPacth* que son sensores para el seguimiento de signos vitales del paciente. Tomando en cuenta esta plataforma se han ido originando varios proyectos que se asemejan, o se han mejorado para satisfacer las nuevas necesidades y requerimientos de los pacientes.

2.2. Proyectos relacionados con el tema del TFE

Como se mencionó en el apartado anterior se han ido desarrollando diferentes proyectos con esta temática, entre los que se puede mencionar los siguientes:

- En *A Smart Pill Box to Remind of Consumption using IoT* (2018), se realizó un dispensador de medicamentos que emite una alarma a la hora de tomar el medicamento, esta alerta termina cuando el usuario presiona un botón para detener la alarma. Además, está diseñado para realizar un conteo de las píldoras, por ejemplo, si el sistema detecta que el usuario no ha tomado, la alarma sonara indefinidamente. Para este sistema se ha usado el microcontrolador Arduino wemos1, que es un microcontrolador de la familia de Arduino, pero con un ESP2866 implementada. Las alarmas, son programadas por medio de una página web, en donde el administrador tiene que registrar al paciente y sus datos. Posteriormente un tutor o el paciente pueden agregar sus medicamentos en los horarios establecidos.

- Asad M (2018), ha propuesto un dispositivo equipado con un microcontrolador Arduino que controlan sensores táctiles que detectan el contacto con la caja de medicamentos y de esta manera se reconoce si el medicamento ha sido tomado o no tomado, además consta de un sensor de glucosa para monitorear el estado de salud del usuario de la caja de medicamentos

- En el trabajo denominado *Using IoT technologies to develop a low-cost* smart medicine (2019), utilizan una interfaz web por donde configuran las horas en que el usuario debe tomar su medicamento. Además, esta web le permite visualizar al médico la configuración de las alarmas y estado de la caja, mientras que la caja de medicinas emite alarmas en la hora programada. Dicha caja esta divida en compartimientos, equipados con leds y un zumbador para alertar que el usuario debe tomar el medicamento del compartimiento en donde el led se enciende. El sistema comparte la información con la web mediante el protocolo MQTT, implementado a través de un microcontrolador ESP2866.

  Una limitación mencionada por el autor del proyecto es que se ha usado un sensor de inclinación que solo detecta, si la caja de medicamentos ha sido abierta o cerrada, y

no detecta si la persona ha tomado sus medicamentos, por lo que recomienda añadir otro tipo de sensores para mejorar el trabajo.

- En el trabajo *Internet of things (IoT) based smart health care medical box for elderly people (2020),* ha implementado una caja de medicamentos con capacidad de conectarse a un servidor por medio de internet y comunicación bluetooth. En el servidor se aloja la información sobre el tratamiento farmacéutico. La caja consta de tres cajones de medicamentos, cada cajón está dotado con un mecanismo con servomotores que bloquean o desbloquean automáticamente la apertura de cada cajón. Las alertas de alarma están dadas por el sonido de un zumbador, un desbloqueo del cajón del medicamento y un mensaje de correo electrónico al usuario, el correo electrónico contiene la información de su tratamiento farmacéutico. Con un sensor de efecto hall se detecta cuando algún cajón sea abierto, y esta información es enviada al servidor de Firebase mediante un microcontrolador ESP32. Además, desde el servidor el médico puede monitorear y realizar el seguimiento del paciente.

- Karagiannis, D. (2020), propone un pastillero con capacidad de 48 pastillas programadas de forma individual, es un prototipo realizado mediante impresión 3D, cuyas dimensiones caben en una mano. Además, mediante la utilización de IoT se consigue monitorizar el entorno y condiciones del paciente. El sistema consta de una interfaz web, para insertar el horario de medicación del paciente, un microcontrolador que capta las mediciones de los sensores de temperatura y humedad. El microcontrolador también realiza solicitudes HTTPP a un servidor para obtener la información de las alarmas, lo que permite notificar al usuario mediante el sonido de un zumbador, la información del sistema es alojada en un servidor que almacena con una base de datos MariaDB.

- Ahmad, S. (2021) implemento una aplicación en Android donde el usuario puede configurar el momento en que la medicina debe ser tomada, la aplicación tiene una

función para conocer detalles sobre el medicamento. El proyecto usa sensores infrarrojos para monitorizar los medicamentos en una botella, los sensores están programados para detectar si la tapa del medicamento ha sido abierta o cerrada, mientras que, las alarmas de toma de medicamento se realizan por medio de notificaciones enviadas desde la aplicación Android. La información de este proyecto es alojada en la nube de AWS y es leída por el microcontrolador Lin kit Smart 7688 Duo.

2.3. Tecnologías relacionadas con el tema del TFE

Las tecnologías que se han empleado en trabajos previos y que se emplearon durante el desarrollo de este trabajo incluyen los siguientes habilitadores de la Industria 4.0:

2.3.1. IoT

Se utilizará dispositivos y sensores que serán conectados al internet mediante una plataforma en la nube, en donde la información será almacenada. Además, con la ayuda de aplicaciones desarrolladas para el usuario se pueda visualizar y programar dicha información cuando el usuario lo requiera.

2.3.2. Servicios Cloud

Los servicios en la nube son servicios remotos conectados a internet que permiten almacenar, administrar y procesar datos. Para este proyecto se necesita un servicio que permita al usuario la conexión de la nube a través de aplicaciones web o móviles.

2.3.3. Ciberseguridad

Al trabajar con datos que se encuentran en la red, el sistema es propenso a sufrir ciberataques con la finalidad de robo de información o actuar de forma maliciosa. Por lo tanto, es importante tener un programa de seguridad, que pueda evitar que personas ajenas al proyecto puedan ingresar.

2.4.Conclusiones sobre el estado del arte

Después de realizar una revisión del estado del arte y los proyectos relacionados a la propuesta de este trabajo, se ha llegado a la conclusión que el IoT es una herramienta potente que puede ser utilizada en una multitud de campos y aplicaciones para la vida cotidiana. Para este caso en específico, se puede observar que existe gran variedad de dispositivos y sensores al servicio de la salud, los que han sido aplicados en diferentes contextos, cada uno con características y procesos diferentes. Por esta razón, se identifica que es indispensable tener claro los requerimientos y límites de la solución propuesta, con la finalidad de enfocar la investigación a dar respuesta a un problema en concreto y evitar desviaciones a otros temas semejantes que podrían dificultar el desarrollo del proyecto.

Por este motivo, para este proyecto se plantea la realización de una caja de medicamentos que conste de varios compartimentos, y que cada compartimento tenga alertas de alarma, y sensores individuales (en cada compartimiento). La caja de medicamentos necesita estar conectada a una base de datos o servicio en la nube, porque unirá todas las partes del proyecto, centralizara la información y es el lugar donde se recupera la información para ser mostrada en las aplicaciones destinadas al uso del usuario

Con respecto a implementación de alertas de alarma se puede observar que se utilizan mensajes de texto, notificaciones en una aplicación Android, mensajes a correo electrónico, leds y un zumbador. Razón por la cual, en este proyecto se plantea reutilizar un sistema de alerta mediante leds y zumbador, debido a su bajo costo, y es amigable con el usuario; y por tanto, es el usuario el que mediante sus sentidos puede saber que medicamentos debe tomar de su caja de medicamentos. Se propone esta alternativa, al contrario de usar notificaciones y correos electrónicos, que, si bien aportan información correcta, el usuario tipo podría tener dificultades con la utilización esta tecnología al tratarse de usuarios con edad avanzada.

En cuanto a la comunicación de la caja de medicamentos con la nube, los proyectos anteriormente mencionados han utilizado los protocolos WiFi o Bluetooth. Ambos protocolos facilitan la comunicación inalámbrica, sin embargo, se puede establecer las siguientes diferencias: La tecnología WiFi es compleja de implementar puesto que tiene que ser

configurada previamente, mientras que con Bluetooth el emparejamiento entre dispositivos es fácil. No obstante, la distancia de comunicación en Bluetooth es de 10 metros, mientras que distancia de WiFi es hasta 100 metros, además, el consumo de energía de Bluetooth es alto en comparación con WiFi, y la velocidad en transmisión la transmisión de datos en Bluetooth es lenta comparada con WiFi. (García, É. 2021)

Con respecto a los sensores se observa una gran variedad de opciones para sensores y todos han funcionado adecuadamente en distintos contextos. En este aspecto se concluye que, para realizar una adecuada elección de los sensores, se debería tener claro los requerimientos y objetivos del proyecto. Además del microcontrolador a usar, puesto que tanto el microcontrolador como sensores están directamente relacionados, como por ejemplo en su modo programación, es decir si ambos son compatibles o tienen recursos como bibliotecas para dichos sensores.

En cuanto al almacenamiento de información en una base de datos se identifica como potenciales sistemas de gestión de bases de datos a las bases de datos MariaDB, y Firebase, debido a que los datos enviarlos por protocolo HTTP, facilitando la implementación de aplicaciones.

En cuanto a los servicios en la nube tanto Firebase con AWS, son buenas opciones, sin embargo, AWS es un servicio que ofrece una infraestructura completa (almacenamiento, redes, servidores, entre otros recursos informáticos) y robusta que puede satisfacer para las necesidades del proyecto. Por otro lado, Firebase según Moreno I. (2022) puede proveer el servicio *backend* (desarrollo lógico de la página web), situación que puede facilitar el desarrollo de las aplicaciones web o móviles.

3. Descripción general de la contribución del TFE

Para el desarrollo de este trabajo y con la finalidad de ofrecer una solución tecnológica a los problemas generados a partir del olvido en la toma de medicamentos se ha establecido los siguientes objetivos y metodología de trabajo:

3.1. Objetivos

3.1.1. Objetivo General

Implementar un sistema de gestión y monitoreo de toma de medicamentos basado en sensores y dispositivos IoT.

3.1.2. Objetivos Específicos

- Diseñar y construir una caja de medicinas inteligente.
- Diseñar un sistema de gestión y monitoreo de información relevante a la toma del tratamiento farmacéutico.
- Realizar pruebas para comprobar el funcionamiento del sistema.

3.2. Metodología del trabajo

En este apartado del proyecto se describirá los componentes del sistema propuesto, sus tecnologías, usuarios y planificación del desarrollo.

3.3. Descripción general de las partes o componentes de la propuesta

3.3.1. Descripción general

Con el propósito de resolver la problemática que tienen las personas al olvidar tomar sus respectivas medicinas en el horario y prescripción correcta, se ha propuesto un sistema que permite gestión y monitorización de medicamentos. En la Figura 1, se muestra la integración del sistema.

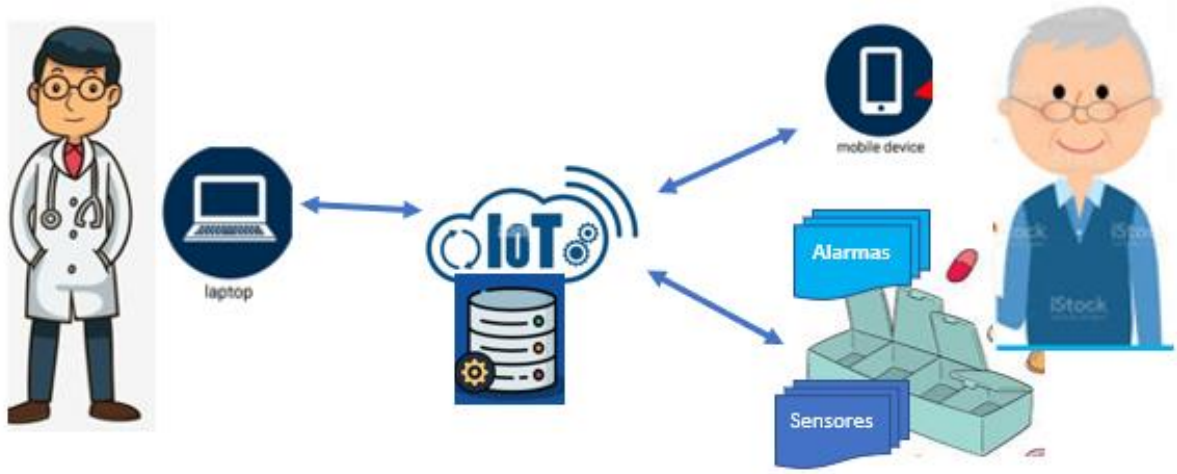
El sistema consta de una caja de medicación inteligente, la cual estará dividida en comportamientos en donde se introducirán los medicamentos, cada compartimento estará equipado con sensores que monitorean si existe movimiento o cambio en el compartimento, además actuadores que permitirán avisar a la hora en que este programado las alarmas.

La información recopilada por los sensores, son analizadas y procesadas, la información resultante es enviada a la base de datos alojada en la nube para ser mostrada en las diferentes aplicaciones.

Las alarmas e información sobre la medicación serán programadas por un médico a través de un sistema de gestión alojado en la web. Desde este sistema estos datos serán enviados a la nube.

Mientras que, los usuarios de la caja pueden revisar información relevante al tratamiento farmacéutico mediante una aplicación móvil.

Figura 1.Diagrama de funcionamiento del sistema



Fuente: Elaboración propia

3.3.2. Alcance y limitaciones

Se pretende diseñar y construir un prototipo de caja de medicamentos inteligente, el cual está en constantemente registrando los cambios en sensores, de esta manera se monitoriza la ingesta de medicamentos.

En este proyecto se propone el desarrollo de una caja que puede ser utilizada por una persona adulta promedio que no requiera de una cantidad elevada de medicamento (cuatro tipos distintos de medicamento).

Por lo tanto, la caja constará de 4 compartimentos, cada compartimento tendrá capacidad para tener entre uno y tres alarmas al día, por ejemplo, puede ser usada en las comidas (desayuno, almuerzo y cena), o solo una vez al día. Cada compartimento tendrá capacidad para almacenar un blíster de medicamentos, eso implica que no se almacenará por unidades de pastillas o por cajas.

Por un lado, las pruebas del dispositivo están orientadas a un paciente que está en tratamiento médico, además, debe ser capaz de usar el dispositivo en su domicilio y tener acceso a una red WiFi. Mientras que el sistema de gestión y monitoreo será utilizado únicamente por los médicos, este sistema consta de una aplicación web en donde se podrán crear pacientes y usuarios de la caja. Además, el médico puede gestionar medicamentos, programar alarmas y monitorear el estado de cada paciente.

Cabe resaltar que os experimentos de evaluación del sistema se realizaron como una prueba de concepto sin la participación de usuarios tipo, por lo que, la implantación de la solución en contextos reales/vida libre puede generar resultados diferentes y nuevos desafíos que solventar.

3.3.3. Listado de participantes

En este proyecto existe participación de los siguientes usuarios:

- Médico tratante, que es la persona que prescribe el tratamiento farmacológico y puede acceder al sistema de gestión para la monitorización del paciente y su tratamiento.

- El usuario de la caja, que es la persona que ingiere los medicamentos y usuario de la caja.
- Usuario de mantenimiento: La persona que puede acceder al sistema, y está pendiente en caso de incidencias, ya sea con la caja de medicamentos o con el sistema de gestión, además es el proveedor de las cajas de medicamento

3.3.4. Tecnologías implicadas

El proyecto consta de dos partes según las tecnologías a usar, el dispositivo y los servicios en la nube (cloud) (*Figura 2*).

El dispositivo fue implementado utilizando dos microcontroladores ESP2866. El primero que se encargara de recopilar la información de los sensores flex provenientes de la caja de medicamentos y el segundo microcontrolador encargado del sistema de alarmas, controla los leds y un zumbador.

Por otro lado, los servicios en la nube están compuestos principalmente por una base de datos en donde se alojará la información personal del usuario del dispositivo, información del tratamiento farmacéutico y alarmas para la caja de medicamentos.

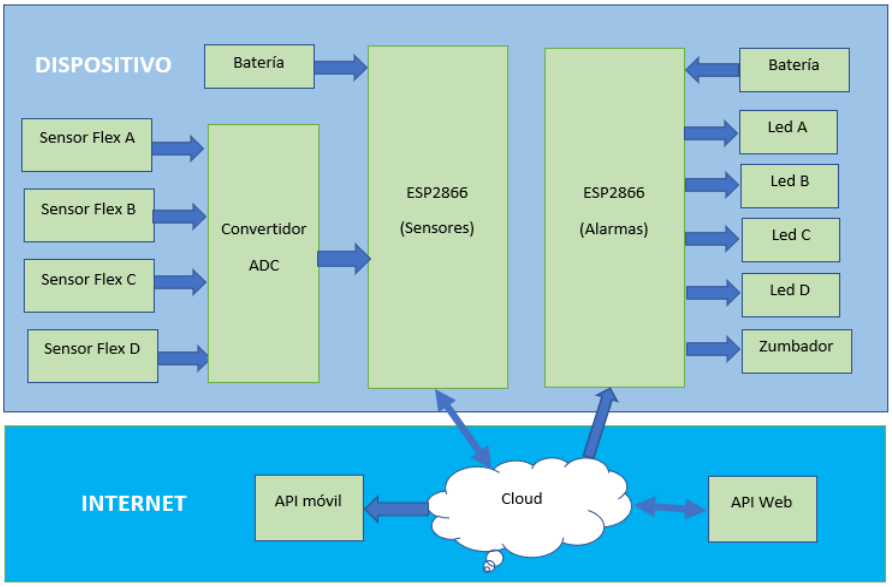
La caja de medicamentos envía información sobre la ingesta (o no) de medicamentos hacia la base de datos, y esta a su vez recibe las horas establecidas para las alarmas.

La información del sistema se puede monitorear desde una página web para médicos o mediante una aplicación móvil para el usuario de la caja.

El dispositivo (caja de medicamentos) y el servicio en la nube están en constante comunicación mediante Wifi, de hecho, el sistema se actualiza en tiempo real, lo que implica que se está trabajando en un entorno IoT.

Adicionalmente, el sistema de monitorización y gestión está protegido con reglas de seguridad para acceso a lectura y escritura a la base de datos, y también mediante la autenticación usuarios autorizados para el ingreso al proyecto.

Figura 2. Tecnologías



Fuente: Elaboración propia

3.3.5. Arquitectura, componentes e integración de tecnologías

El sistema está basado en arquitectura IoT por lo tanto tiene las siguientes capas:

- La capa de dispositivos que contiene la caja de medicamentos equipada con controladores, sensores y actuadores.
- La capa de red que envía los datos vía internet hacia la capa de Cloud, donde se almacena los datos del sistema.
- Una capa de seguridad para evitar que ingresen intrusos o exista robo de información.
- Y una capa de gestión, filtro y procesamiento de información, es decir los datos de los sensores no son mostrados, sino información concreta sobre las alertas de alarma del medicamento. El procesamiento será de tipo *Edge Computing* (computación en el borde), es decir el procesamiento y filtrado de los sensores será en el microcontrolador.
- Capa de Aplicación: Esta compuesta por aplicaciones diseñadas para que el usuario pueda interactuar y visualizar, la información del sistema, alojado en la base de datos. El usuario

medico tendrá acceso a una aplicación web y mientras que el usuario de la caja de medicamentes o su familiar tienen acceso a aplicación móvil.

En la Figura 3 se puede observar el diagrama de la arquitectura del sistema

Figura 3. Arquitectura del sistema

Fuente: Elaboración propia

3.3.6. Resultados Esperados

Se espera que el sistema de gestión pueda conectar a los pacientes con los médicos de forma remota, que los datos e información se guarde en la nube para ser vista en cualquier momento, que el sistema sea amigable y fácil de usar para los usuarios, además sea seguro para que no existan fugas de información que pueden ser usadas de forma maliciosa.

3.3.7. Presupuesto y retorno de la inversión

3.3.7.1. Presupuesto

En este apartado se presenta una estimación de los costes asociados al desarrollo del proyecto, como, por ejemplo, el diseño y construcción del dispositivo (caja de medicamentos),

diseño e implementación del sistema de gestión, recursos humanos que incluyen las horas de trabajo dedicadas al proyecto.

En la Tabla 1 se muestra los materiales y costes de construcción de la caja de medicamentos (Hardware). En esta tabla se incluye los costos derivados de la utilización de las plataformas de desarrollo (Arduino IDE y Firebase).

Tabla 1. *Costo caja de medicamentos*

	Cantidad	Costo Unitario	Costo total
	Software		
Arduino IDE	1	0 €	0 €
Firebase	1	0 €	0 €
	Hardware		
Caja madera	1	10 €	10 €
Cartón	1	5 €	5 €
Material Velostat	1	23,35 €	23,35 €
Pack Cables	1	5 €	5 €
Led	5	0,10 €	0,50 €
Resistencia	10	0,10 €	1,00 €
ESP2866	2	10 €	20 €
Zumbador	1	0,10 €	0,10 €
Interruptor	1	0,10 €	0,10 €
Alimentación	1	10 €	10 €
Total			75,05 €

Fuente: Elaboración propia

Para el sistema de gestión se utilizaron las siguientes plataformas de desarrollo (*Tabla 2*), todos estos programas tienen acceso a sus funciones de forma gratuita.

Tabla 2. *Costo Sistema de gestión y monitorización*

	Cantidad	Costo Unitario	Costo total
	Software		
Android Studio	1	0 €	0 €
Firebase	1	0 €	0 €
Visual estudio Code	1	0 €	0 €

Fuente: Elaboración propia

Hay que tener en cuenta que el costo de Firebase el en momento de desarrollo de este proyecto fue gratuito, no obstante, cuando la aplicación comience a escalar en el número de usuarios y llegue a generar ganancias, la plataforma Firebase empezará a cobrar por cada uno de los servicios prestados.

En las tablas anteriores solo se ha mencionado los recursos materiales, usados, sin considerar las personas. Por lo cual, se ha realizado una estimación de los costeas asociados a los recursos humanos.

Para el desarrollo del proyecto se ha trabajado aproximadamente en la construcción e implantación un periodo de seis meses, contado ocho horas cada día durante seis días a la semana. Con estos datos se ha calculado que se ha trabajado un total de 1152 horas en los últimos 6 meses. Este es un cálculo aproximado puesto que hay días en que se ha trabajado más horas a la semana, de las planteadas.

Según Talent.com (2022), el salario medio de un ingeniero en proyectos en España es de €15,38 por hora, por lo tanto, el coste del proyecto en recursos humanos es de € 17.717,76

En conclusión, el costo del proyecto ha sido de €17.792,76, resultado de la suma de todos los costos antes mencionados.

3.3.7.2. Retorno de la inversión

El retorno de la inversión (ROI), es una métrica para medir el éxito de la inversión en este proyecto. Se calcula mediante la fórmula:

$$ROI = \frac{Ganancia - \ Coste\ de\ la\ inversion}{Coste\ de\ la\ inversion}\ x\ 100$$

En este proyecto se espera tener una ganancia de €40000 que es el redondeo del doble de lo invertido. El coste de la inversión estará dado por el coste total del proyecto que es €17.792,76. Entonces con los datos anteriores calculamos el retorno de la inversión:

$$ROI = \frac{Ganancia - \ Coste\ de\ la\ inversion}{Coste\ de\ la\ inversion}\ x100$$

$$ROI = \frac{40000 - \ 17.792{,}76}{17.792{,}76} = 124\%$$

El ROI se expresa en porcentaje y se interpreta dependiendo si el porcentaje es mayor o menor al 100%. Cuando el ROI es superior al 100%, implica que la inversión es rentable y para este caso, se está ganando el 124% de la inversión, lo que implica, que, por cada dólar invertido, se estaría ganando 1.24 dólares.

3.3.8. Planificación general

Para la elaboración del proyecto se han propuesto la siguiente distribución de tiempo y actividades mostrados en la *Figura 4*, mediante un diagrama de Gantt. Se puede observar que los dos primeros meses febrero y marzo fueron destinados a la investigación búsqueda de opciones para el desarrollo de trabajo, el mes de abril se ha empleado en el desarrollo de la página web (sistema de gestión) y la aplicación móvil. En el mes de mayo se destinó a la construcción de la caja de medicamentos inteligente. En los meses de mayo y julio se emplearon para la redacción de la memoria del TFM, en conjunto con las actividades como el desarrollo de la metodología y sistema de gestión, así como su evaluación.

Figura 4. *Diagrama de Gantt planificación del proyecto*

	Febrero				Marzo					Abril					Mayo					Junio					Julio				
Semana	1	2	3	4	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Desarrollo formulario TFM																													
Introducion TFM																													
Estado del Arte TFM																													
Objetivos TFM																													
Metodologia y Descripcion del trabajo																													
Desarrollo Requerimentos TFM																													
Desarrollo sistema de gestión																													
Desarrollo caja de medicamentos																													
Dearrollo metodologia TFM																													
Conexión caja + sistema de informacion																													
pruebas																													
Memoria TFM																													

Fuente: Elaboración propia

4. Desarrollo específico de la contribución

4.1.Requerimientos

Para definir los requerimientos se tomará en cuenta las características de una arquitectura de referencia IoT:

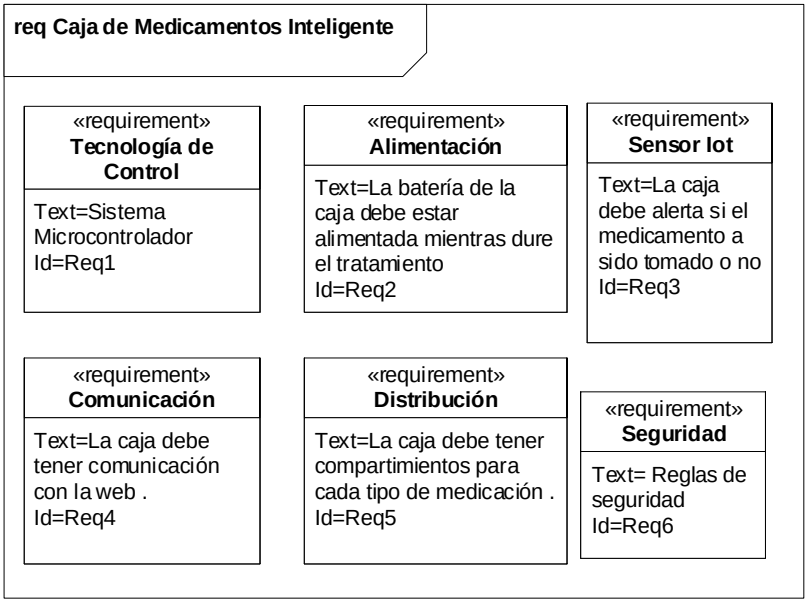
- Conectividad y comunicaciones
- Gestión de dispositivos
- Escalabilidad e integración
- Seguridad y privacidad

4.1.1. Requerimientos sistema de monitoreo

En la Figura 5, se muestra los requerimientos necesarios para el desarrollo y funcionamiento correcto de la caja de medicamentos.

Primero se necesita una caja dotada de compartimientos en donde alojaran los medicamentos, además de tener canales por donde va a pasar el cableado del dispositivo. La caja debe estar dotada de sensores que monitoreen los medicamentos y sus cambios. Estos sensores deben ser gestionados y controlados por un controlador, el cual debe tener una alimentación que dure durante todo el proceso. El controlador debe estar programado para recopilar datos, gestionarlos. Además, el sistema debe estar conectado a la red para la comunicarse con el sistema de gestión y visualización.

Figura 5. *Requerimientos caja inteligente*



Fuente: Elaboración propia

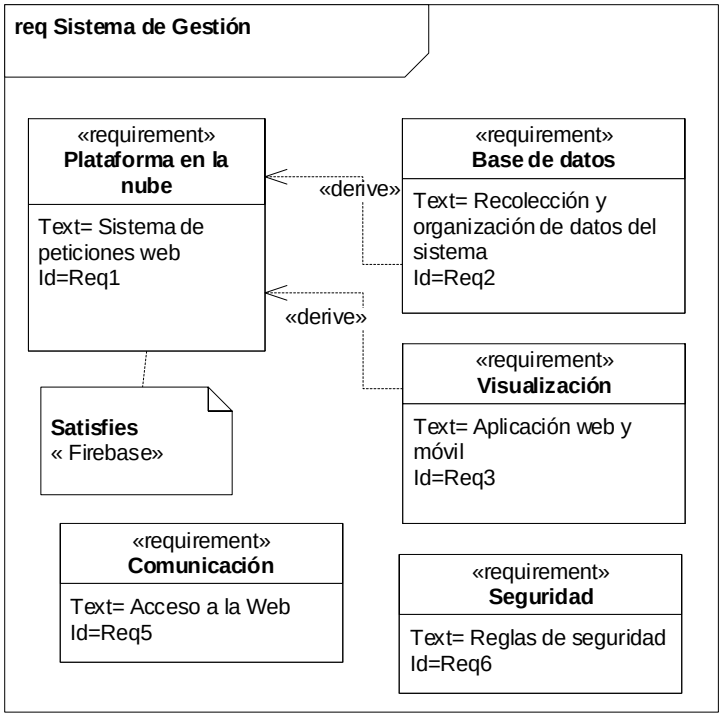
4.1.2. Requerimientos del sistema de gestión e información

Los requerimientos para el sistema de gestión:

- La información procesada de los sensores debe ser enviada a una base de datos en donde se aloja los datos del tratamiento farmacéutico, usuarios, alarmas.
- El sistema de información debe tener seguridad para que no hay robo de datos.
- El sistema de seguridad de dar acceso a la información a médicos.
- La información de la base de datos debe ser mostrada a través de una aplicación web y una aplicación móvil.

En la Figura 6, se muestra los requerimientos necesarios para el funcionamiento del sistema de Gestión de información.

Figura 6. *Requerimientos sistema de gestión*



Fuente: Elaboración propia

4.2. Diseño general del sistema

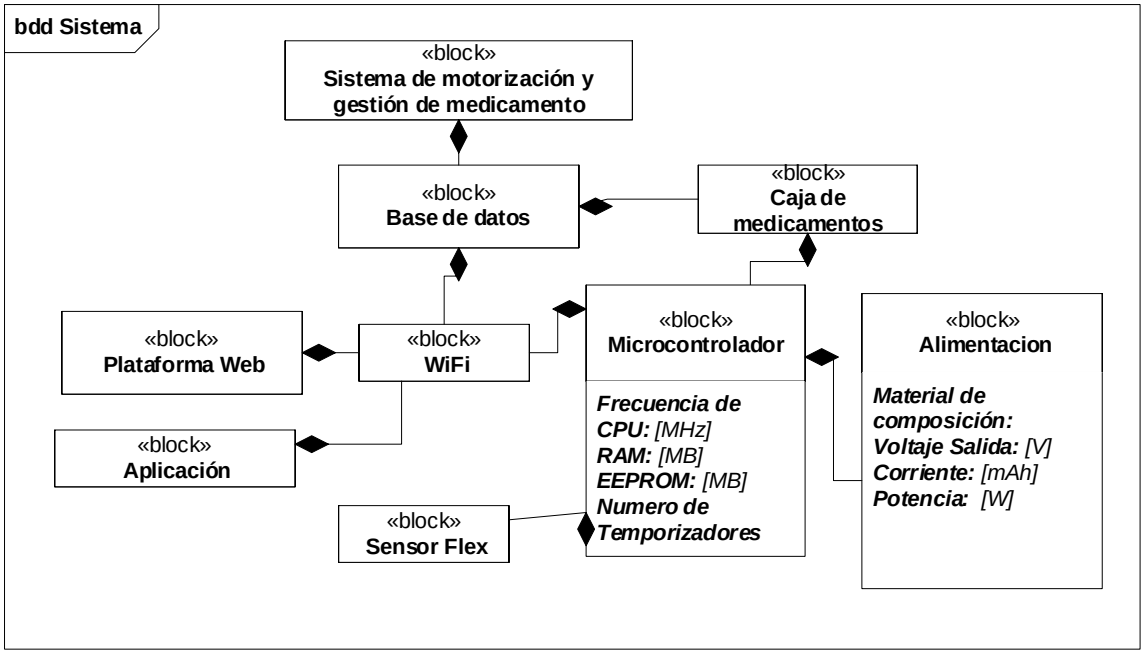
En la Figura 7, se muestra el diagrama de definición de bloque (bdd), este diagrama representa la estructura del sistema dividiéndola en bloques estructurales genéricos y sus relaciones.

El sistema consta de dos subsistemas:

- Sistema de Monitorización: Consta de una caja de medicamentos con compartimientos, controlada por un microcontrolador que procesa datos provenientes de los sensores flex, un módulo que permita acceso a la red WiFi. Además, está equipado con leds y un zumbador, para generar alertas de alarma, cuanto sea el momento de tomar medicamentos. Todo este sistema debe estar siempre alimentado mientras el tratamiento médico del usuario termine su tratamiento.
- Sistema de Gestión: El sistema debe estar conectado a la red WiFi para conectarse al controlador, y recibir los datos del sistema.

- La información del sistema se aloja en una la nube. Para que los usuarios puedan observar dicha información el sistema cuenta con una plataforma en la web y una aplicación móvil.

Figura 7. *Diagrama general funcionamiento del sistema*



Fuente: Elaboración propia

4.3. Servicio Cloud

Firebase es un servicio en la nube de tipo BaaS (*Backend* como servicio), lo que significa que es un servicio en la nube que proporciona *backend* (lógica de la página web), mediante herramientas de desarrollo para diferentes lenguajes de programación, además provee servicios como almacenamiento, análisis de datos, entre otros.

Firebase es una plataforma móvil de Google, que facilita el desarrollo y la creación de aplicaciones de forma rápida, mediante la implementación de distintos servicios y funciones.

Firebase incluye servicios utilizados para la creación de aplicaciones móviles o web, los servicios son: Base de datos en tiempo real, Autenticación de usuarios, almacenamiento en la nube, mensajes *cloud, hosting, test lab*, configuración remota, entre otras

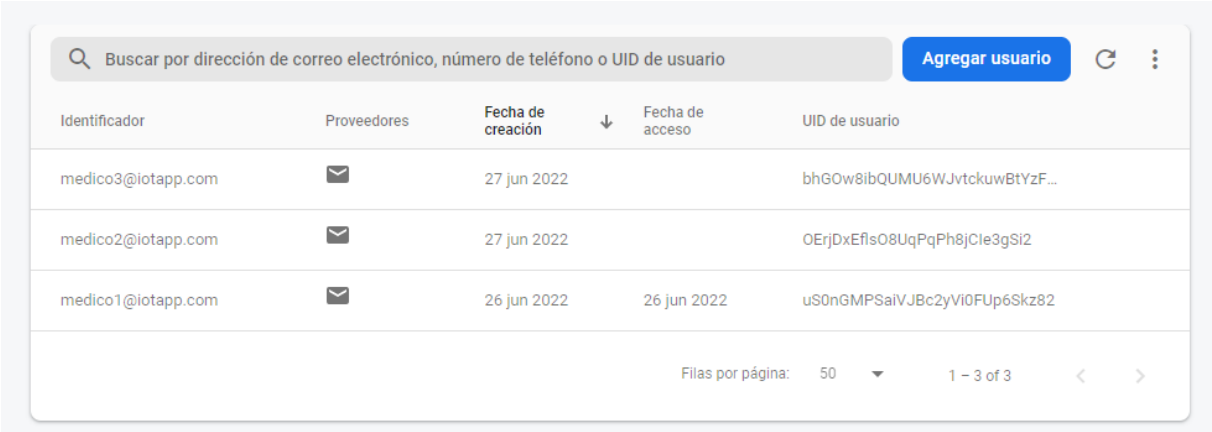
Para este proyecto se utilizará los servicios de base de datos en tiempo real y autenticación.

4.3.1. Firebase Autentication

Firebase Autentication permite conocer la identidad de los usuarios y guarden de forma segura sus datos, de esta manera el sistema gestiona quien puede ver o no la información proporcionada por cualquiera de los servicios.

En la Figura 8, se muestra varios usuarios agregados a este servicio mediante correo electrónico y una clave, además se puede observar que el servicio proporciona un UID por usuario, este UID es un identificador único dado por Firebase para identificar a cada usuario añadido. Para este caso, se observa los médicos agregados al servicio de autenticación que tienen acceso a la información de la base de datos.

Figura 8. *Firebase Autenticathion*



Fuente: Elaboración propia

4.3.2. Firebase Real time Data Base

Es una base de datos NoSQL, que almacena y sincroniza los datos alojados en la nube, los datos se sincronizan con los usuarios de las aplicaciones en tiempo real. Para este proyecto la base de datos permite la interacción entre el médico, el usuario de la caja y los sensores de la caja.

En la Figura 9 se observa dos nodos principales de la base de datos. El primer nodo “caja” que contiene 5 nodos hijos que son la letra de cada compartimento en la caja (A, B, C, D) y un nodo

hijo que verifica si la caja esta encendida o apagada (nodo Caja_state). La información de Caja_state es información tomada por el ESP2866 conectada a un interruptor, cuando la caja esta encendida las alarmas establecidas funcionarán, sin embargo, cuando la caja esta apagada, no existirán alertas de alarma.

El segundo nodo denominado “pacientes”, es donde se van guardando los pacientes y sus datos (información personal), Estos pacientes son agregados a través de la página web.

Figura 9. *Nodos Base de datos*

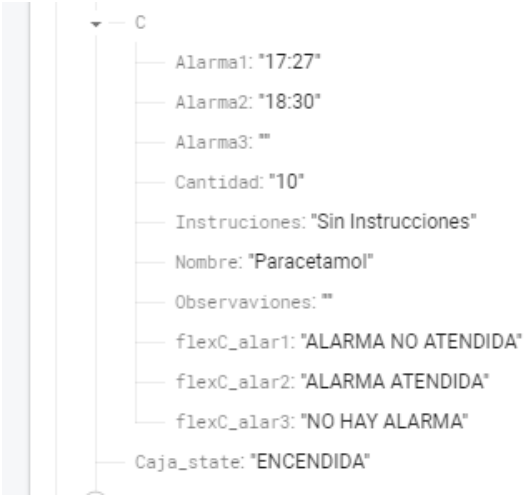
Fuente: Elaboración propia

En la Figura 10, se observa el contenido de los nodos hijos que pertenecen a cada compartimento de la caja de medicamentos, aquí podemos observar los siguientes nodos hijos: Nombre, Alarma1, Alarma2, Alarma3, Cantidad, Instrucciones y Observaciones, nodos son datos e información con respecto al tratamiento farmacéutico, y son tomadas de la página web con la que interactúan los médicos.

Los nodos flexA_alar1, flexA_alar2, flexA_alar3 son nodos que provienen de la ESP2866, los datos recopilados por los sensores son analizados, procesados y catalogados en tres grupos,

ALARMA NO ATENDIDA, ALARMA ATENDIDA y NO EXISTE ALARMA

Figura 10. *Nodo hijo base de datos*



Fuente: Elaboración propia

En el nodo pacientes, se observa los pacientes y su información añadidos por el medico a través de la plataforma web. (Figura 11)

Figura 11. *Nodo Pacientes*

Fuente: Elaboración propia

4.4.Caja de Medicamento inteligente

En este apartado se describirá la elección del controlador, sensores, actuadores. Además de las partes principales de la codificación para el funcionamiento de la caja inteligente.

4.4.1. Sistema Electrónico

4.4.1.1. Controlador

En base a los requerimientos se necesita una tarjeta, que permita acceso a internet, debido a que el sistema necesita conectarse a la nube. Como el proyecto no se desarrolla a nivel industrial entonces los controladores elegidos pueden ser microcontroladores o procesadores utilizados para prototipos.

Entre los controladores más conocidos están Arduino, Node MCU ESP8266 y Rasberry PI. Por lo que, en el caso de Arduino y Rasberry se eligió solo las placas que cuenten con WIFI.

En la Tabla 3, se observa una comparativa de los controladores. Primero se observa que la Raspberry Pi 3b, tiene mayor procesamiento, sin embargo, no tienen ninguna entrada analógica para los sensores Flex. La segunda opción es la ESP2866 que tiene un procesamiento medio y una sola entrada analógica, tanto para la ESP2866 como para la Raspberry existe solución para la falta de entradas analógicas y es implementar extensiones de entradas analógicas, y finalmente el Arduino MKR1000, es un Arduino con modulo WiFi incluido, sin embargo, hay varios dispositivos que no son compatibles.

Para el proyecto se podía utilizar una ESP8266 o una Raspberry de manera aceptable, sin embargo, la Rasberry PI 3b se encuentrava agotada en el mercado. Por lo tanto, se ha decidido utilizar Node MCU ESP8266, que además cuenta con librerías incorporadas para la conexión con internet y la nube.

Tabla 3. *Comparación microcontroladores*

	Rasberry PI	Node MCU	Arduino
Placa	3b+	ESP8266	MKR1000
WiFi	Si	Si	Si
I/O Digitales	40	11	8
Entradas Analógicas	No	1	7
RAM	1G	64Kb	32 KB

Fuente: Elaboración propia

4.4.1.2. Sensores

Basada en la investigación en el marco teórico sobre los sensores que se pueden utilizar en para detectar la toma de medicamentos, se eligieron dos opciones de sensores para ser acoplados al proyecto.

- RFID:

  La idea de usar tarjetas RFI, era identificar el medicamento que se está tomando mediante la letra de su compartimento,

  Por lo tanto, se iba a usar 4 tarjetas RFID que se encontraban en los compartimentos junto con la respectiva medicación. (Figura 12)

Figura 12. *Prueba sensores RFID*

Fuente: Elaboración propia

La tarjeta RFI, necesita un lector de tarjetas RFID, que toma el serial de cada tarjeta y compara con seriales guardados en el programa cargado en el microcontrolador para reconocer a que compartimento pertenece cada tarjeta. En la Figura 13 se observa una parte del código para el microcontrolador, se observa la comparación de las tarjetas y si las tarjetas coinciden con los seriales antes guardados, se imprime a el compartimento que pertenece, si no cumple esto el sistema imprime que el medicamento no existe.

Figura 13. *Programación RFID*

```
if(comparaUID(LecturaUID, MedA))
  Serial.println("compartimientoA");
else if(comparaUID(LecturaUID, MedB))
  Serial.println("compartimientoB");
 else if(comparaUID(LecturaUID, MedB))
  Serial.println("compartimientoC");
  else if(comparaUID(LecturaUID, MedB))
  Serial.println("compartimientoD");
 else
  Serial.println("No existe medicamento");
```

Fuente: Elaboración propia

El uso de tarjetas RFID son una buena opción para este proyecto, sin embargo, al realizar el análisis y pruebas con este sensor, las tarjetas pueden confundirse, o el usuario puede olvidar de pasar la tarjeta por el lector después de ingerir sus medicamentos, además que, al intentar enviar esta información a la base de datos, se interrumpía la conexión. Por esta razón, no se usó tecnología RFID en la implementación del sistema propuesto.

- Sensor Flex:

  Los sensores Flex pueden detectar la flexión de un material, es decir cambian el valor de la resistencia cuando este es flexionado, presionado o doblado.

  Entonces como primera idea para usar este sensor era pegar el sensor en cada medicamento, sin embargo, en la Figura 14**,** se observa que el sensor siempre tenía

que estar conectado a los cables, y había momentos en que los cables resultaban molestos para sacar el medicamento y en el peor de los casos los cables se desconectaban. Adicionalmente a esto los sensores Flex en el mercado son costosos, (19 euros cada uno) y se necesitaban cuatro sensores.

Figura 14. *Prueba Sensor Flex comparado*

Fuente: Elaboración propia

En la búsqueda de información sobre este sensor, se encontró que se podían hacer estos sensores me forma casera, la idea de crear sensores caseros es que se acople a los medicamentos, pero sin la necesidad de tener esos cables molestos y bajar el costo de los sensores.

La forma de realizar sensores Flex caseros es mediante el uso de material Velostat que es un material conductivo sensible a la presión, al apretar el material este reducirá su resistencia. Entre los usos del material Velostat está la creación de sensores, flexibles, sensores de impacto, táctiles entre otros.

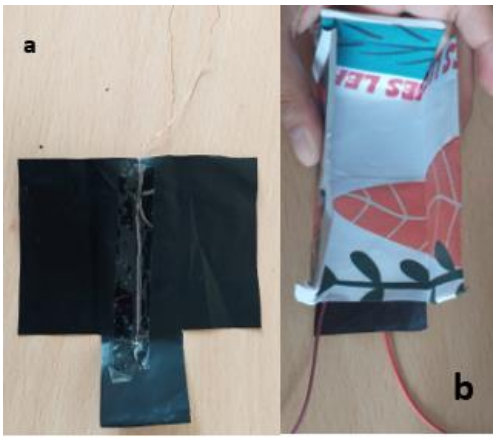
En este punto de la investigación se identificó que se puede usar Velostat como sensor táctil y sensor flexible. Aplicado al proyecto se puede monitorear el comportamiento de cada compartimento, cuando se produce una reducción en la resistencia de cada compartimento recubierto con este material, se asume que el medicamento ha sido ingerido, debido a que al introducir o extraer las manos para sacar los medicamentos, la

resistencia de dicho compartimento cambia por el simple contacto del material velostat con las manos.

- Construcción de sensores:

Para construir un sensor Flex, se necesita dos cables y el material Veslostat, entonces se pega cada cable en ambas caras del Velostat. En la Figura 15a, se observa un sensor Flex construido de forma casera, mientras que la Figura 15b, se observa el sensor recubierto por papel para mantener la integridad del sensor y mantener la forma del compartimento.

Figura 15. *Sensor Flex Velostat*



Fuente: Elaboración propia

4.4.1.3. Alerta de alarmas

Para que la caja de medicamentos sea capaz de enviar avisos al usuario de alarma, se empleó, cuatro diodos emisores de luz (leds), ubicados en cada uno de los compartimentos y un zumbador, los cuales se activarán cada vez que exista una alarma en el respectivo compartimento. (Figura 16).

Figura 16. *Alarma caja de medicamentos*

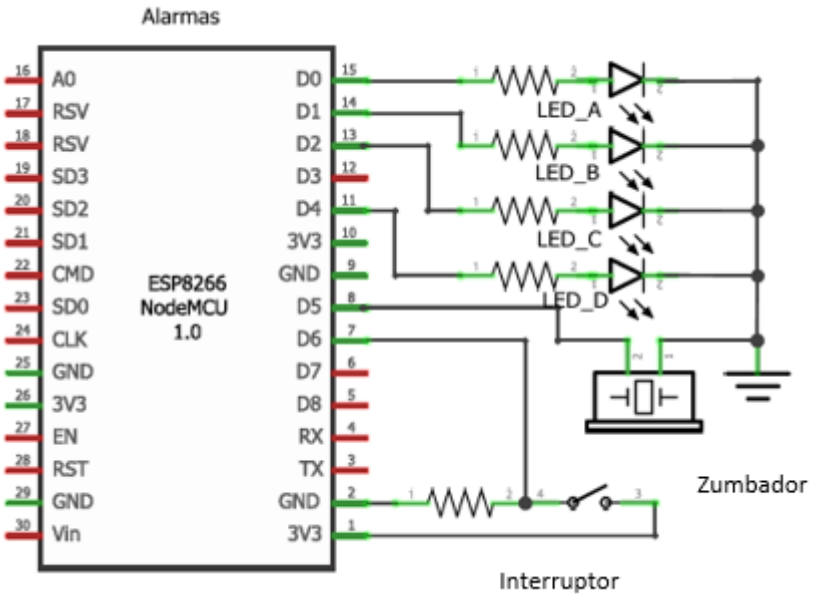
Fuente: Elaboración propia

4.4.1.4. Conexión electrónica caja medicamentos

Para el funcionamiento de la caja de medicamentos se ha decidido implementar dos circuitos electrónicos y realizar dos programas, unidos por la base de datos. Entonces se utilizará dos placas ESP8266.

La primera placa ESP2866, está configurada para que realice el recordatorio de alarmas, por lo tanto, cuenta con 4 leds ubicados en cada compartimento, un zumbador, y un interruptor para prender o apagar la caja. En la Figura 17, se observa el diagrama electrónico y las conexiones, se puede observar que cada led es conectado a una resistencia de 330 Kohm, y se conectan a los pines de la ESP2866, el led A esta conectado a el pin D0, el led B conectado al pin D1, el led C conectado al pin D2 y el led D conectado a el pin D4, el interruptor está conectado al pin D6 y el zumbador conectado al pin D5.

Figura 17. *Conexión circuito Alarmas*



Fuente: Elaboración propia

La segunda ESP2866 placa realiza la toma de datos de los sensores alojados en los compartimentos. Los sensores Flex son sensores que se leen de forma analógica, la placa solo tiene un pin analógico por lo tanto se implementó un convertidor analógico digital.

El ADS1015 (Figura 18), es un convertidor analógico digital externo que permite obtener mayor resolución en la lectura de pines y ampliar el número de pines analógicos

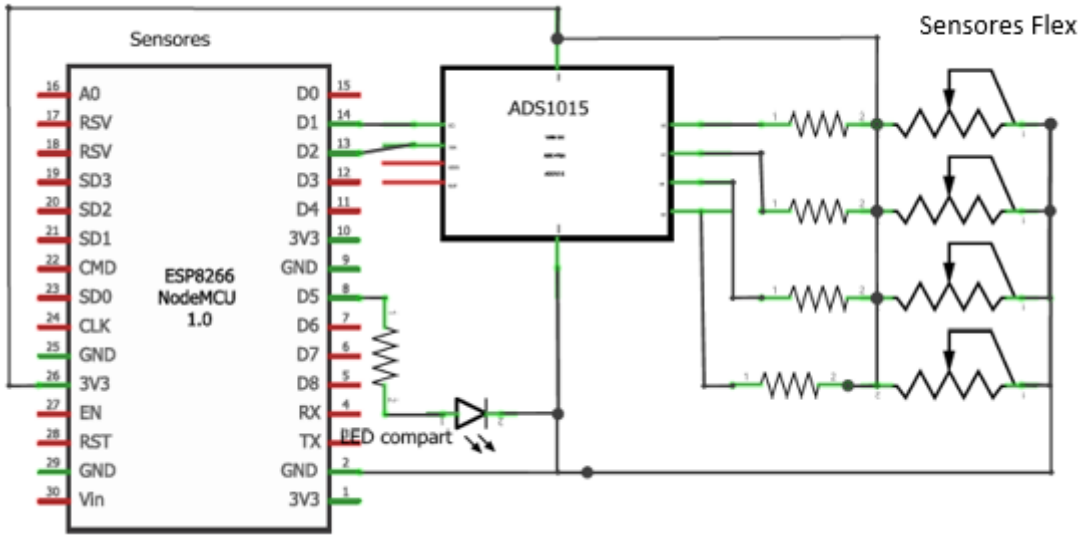
Figura 18. *Convertidor ADS1015*

Fuente: ADS1015 12-Bit ADC board, Adafruit

En la Figura 19**,** se observa la conexión de la segunda placa ESP2866, la conexión consta de un led con su respectiva resistencia de 330kOhms conectada al pin D5 del microcontrolador, el convertidor analógico ADS1025 está conectada al ESP2866 mediante

protocolo de comunicación I2C, entonces los pines SCL y SDA ADS1025 están conectados a los pines D1 y D2 de la ESP2866 respectivamente. Cada sensor Flex está conectado a una resistencia de 1 Kohm y estos conectados a los pines analógicos de la ADS A0, A1, A2, A3

Figura 19. *Conexión sensores Flex*



Fuente: Elaboración propia

4.4.1.5. Programación, análisis y procesamiento de datos

La programación de cada ESP2866 se ha realizado en IDE de Arduino (editor de texto).

Como se mencionó en el apartado anterior existen dos circuitos electrónicos por separado, por lo tanto, existen dos códigos y programas para su funcionamiento. Sin embargo, ambos programas comparten librerías comunes. Las librerías son:

- lib_red.h: Es una librería, que permite a la ESP2866 establecer conexión con WiFi.
- fechahora_actual.h: Es una librería que usa protocolo NTP (*Network Time Protocol*), este protocolo es encargado de sincronizar el reloj de la ESP2866, y retorna la información de hora y fecha actual.
- funcionreturnhora_fecha.h: Es una librería que retorna hora y fecha de la base de datos modificada su formato.

En la Tabla 4. Formato fecha y hora se observa los formatos de hora y fecha dados por la ESP2866 (columna 2 de la tabla) y los de la base de datos Firebase (columna 3). Además, se observa que la base de datos retorna hora y fecha, añadiendo un cero cuando no existe dígitos solos, sin embargo, las horas y fechas generadas por la librería funcionreturnhora_fecha.h, omiten el cero.

Tabla 4. *Formato fecha y hora*

	ESP2866	**DB**
Hora	18:2	18:02
Fecha	13-6-2022	13-06-2022

Fuente: Elaboración propia

Entonces al momento de hacer comparaciones entre fechas y horas, la respuesta siempre mostraría que los datos son diferentes, y no se podría realizar alertas de alarma. Por tanto, la librería funcionreturnhora_fecha.h, toma los datos de la hora y fecha alojados en la base de datos de Firebase, y los procesa para obtener hora y fecha en formato igual a la hora y fecha del protocolo de la ESP2866.

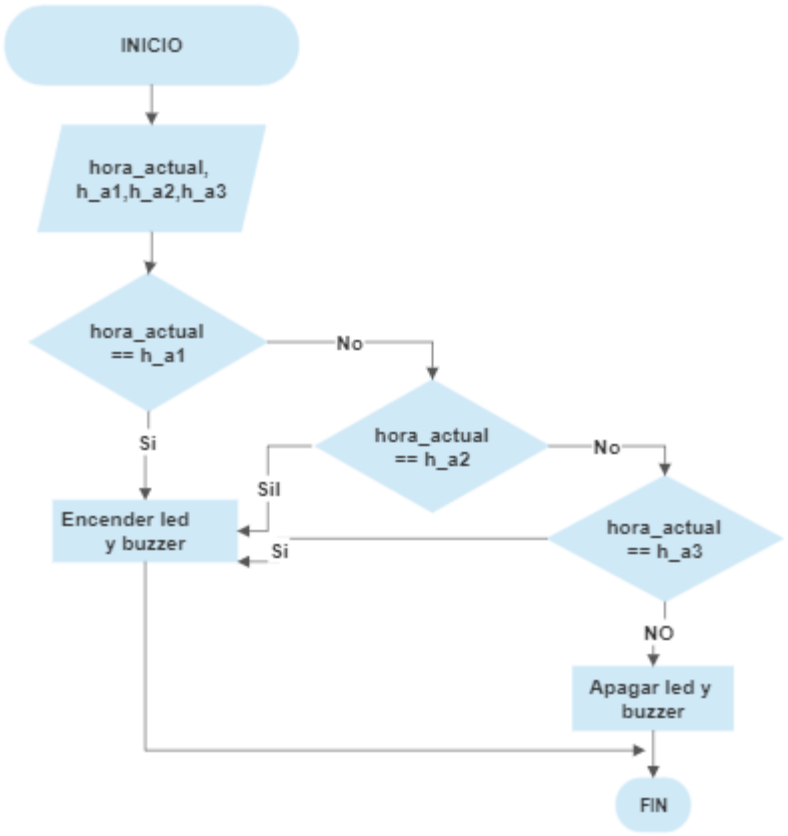
Programa esp2866 alarmas:

El propósito de este algoritmo es establecer una alerta de alarma mediante leds y un zumbador, estableciendo una comparación con la hora actual y las horas de las alarmas propuestas por el médico, estas alarmas provienen de la base de datos de Firebase.

En la Figura 20, se observa la lógica de la función principal del programa, comienza tomando la información de hora actual, y tres alarmas provenientes de cada compartimento. Entonces el programa procede a comparar una a una las alarma (en formato horas) con la hora actual, la comparación comienza con la alarma1, si la comparación resulta ser verdad, entonces se prende el led correspondiente a este compartimento y se enciende el zumbador o buzzer. Si la comparación resulta ser falsa

entonces pasa a la segunda comparación, y repite el proceso para las alarmas dos y tres. Finalmente, si no se cumple ninguna de estas condiciones el programa apaga los leds y el zumbador. Esta función se repite para cada compartimento por lo que hay 4 funciones iguales con la diferencia que se usa diferentes leds y diferentes alarmas por cada compartimento.

Figura 20. *Función Alarmas*



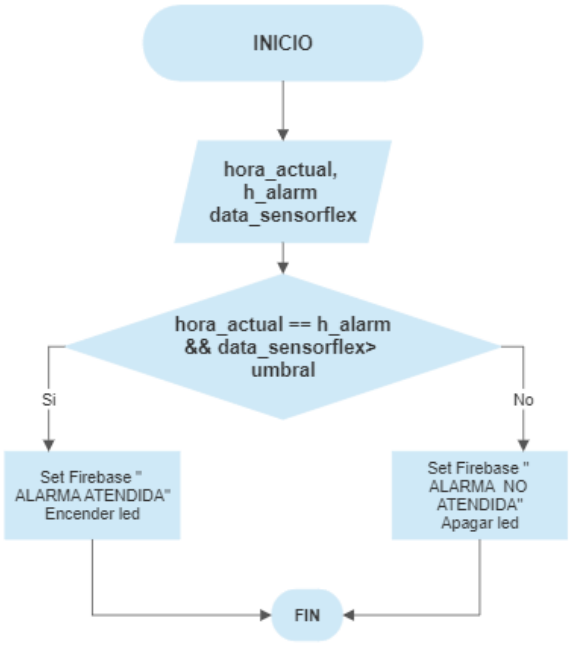
Fuente: Elaboración propia

Programa ESP2866 Sensores:

Lo que hace este programa es recoger los datos de sensores Flex y comparar si estos superan un umbral, este umbral indica si el sensor fue tocado, flexionado, presionado o doblado. Además de esto, también se realiza una comparación entre hora actual y hora de las alarmas.

En la Figura 21, se observa un condicional compuesto por el umbral del sensor Flex y la comparación con la hora de las alarmas. Esta doble comparación se realiza para que los sensores tomen datos solo cuando haya una alarma y evitar falsos positivos. Cuando la condición doble se cumpla, se envía a la base de datos "ALARMA ATENDIDA", lo que implica que hubo una alarma y se introdujo la mano del usuario en la caja, sin embargo, si la condición no se cumple, implica que la alarma sonó, pero el usuario no tomo los medicamentos del compartimento, y por lo tanto se envía a la base de datos "ALARMA NO ATENDIDA".

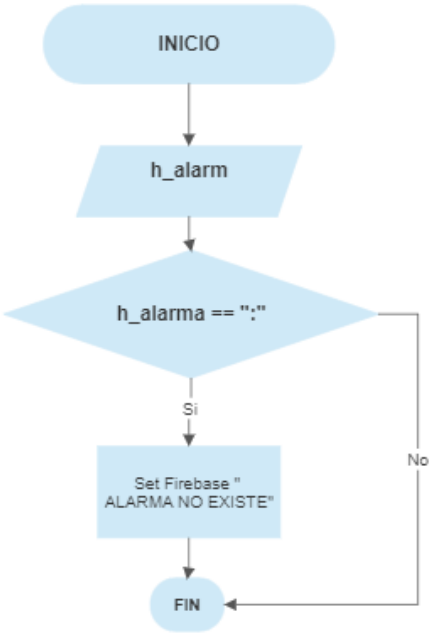
Figura 21. *Función alarma/sensor Flex*



Fuente: Elaboración propia

Otra función del programa incluye la verificación de la existencia de las alarmas, entonces compara todas las alarmas con ":", indica que no hay hora de alarma y no existe alarma, por lo tanto, al cumplir la condición el microcontrolador escribe en la base de datos, "NO EXISTE ALARMA" (Figura 22)

Figura 22. *Función no alarma*



Fuente: Elaboración propia

4.5.Sistema de Gestión de información

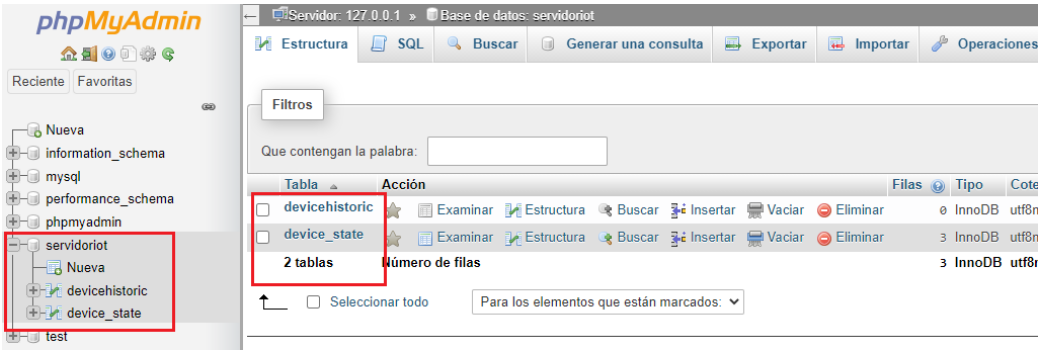
Según los requerimientos para el sistema de información se necesita tener un lugar en donde almacenar datos de los pacientes y sus medicamentos, es decir una base de datos, que pueda conectarse a un sitio web y una aplicación Android desde donde se puede visualizar dicha información. Entonces se propuso dos soluciones de base de datos que pueden ser conectadas a un *Node MCU* mediante librerías de Arduino:

1. MySQL + PHP: Se utilizó XAMP que es un servidor web para HTTP, que incluye la base de datos phpMyAdmin en donde se alojaban los datos enviados por los sensores de la caja de medicamentos mediante protocolo http. Los datos eran recopilados por la NodeMCU con una librería que facilitaba la comunicación entre el microcontrolador y un programa realizado en html + php, los mismos que permitía visualizar los datos del paciente, de principio se ha desarrollado por este método y se ha concluido que, al trabajar con http, se presentaban fallos en la seguridad y tenía una navegación no

segura, además se necesitaba contratar un dominio web para montar la página aumentando costos. Por otro lado, al hacer una prueba gratuita del dominio web se restringían varias funcionalidades, he incluso se perdía información.

En la Figura 23 se muestra el ingreso de datos provenientes del microcontrolador NodeMCU y alojados en la base de datos phpMyAdmin

Figura 23.*Ingreso de datos con PhpMyAdmin*



Fuente: Elaboración propia

2. Firebase: Como se mencionó en el apartado 4.3, el proyecto utiliza servicio en la nube Firebase, con sus dos servicios *Firebase Autentication y Firebase Realtime*. El microcontrolador *Node MCU* (ESP2866) tiene librerías para trabajar con Firebase permitiendo escribir información de los compartimentos para cada medicamento en los nodos de la base de datos.

   Otra cuestión importante de Firebase es que brinda seguridad de los datos. Al usar *Firebase Autentication y Firebase Realtime* se puede controlar el acceso a lectura y escritura en la base de datos (Figura 24). En la figura se observa las reglas establecidas para la base de datos, que significa que la base de datos puede ser leída por cualquier persona con acceso a las aplicaciones, pero solo puede escribir en la base de datos la persona que este autenticada.

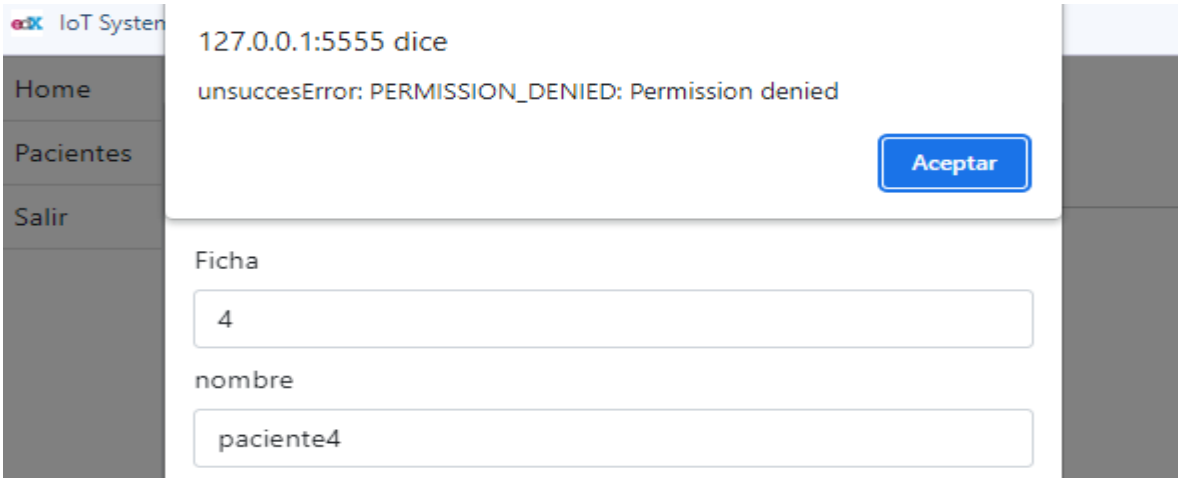
Figura 24. *Reglas Firebase database*

```
{
  "rules": {
    "users": {
      "$uid": {
        ".read": true,
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

Fuente: Elaboración propia

En la Figura 25, se observa un extracto de la página web del médico, en este apartado se intenta crear un nuevo paciente, pero al intentar escribir en la base de datos sin ser autenticado Firebase nos envía un mensaje de alerta, negándonos los permisos.

Figura 25.*Permiso denegado*



Fuente: Elaboración propia

4.5.1. Desarrollo aplicación web

La página web, esta desarrollada para que utilicen los médicos tratantes del usuario de la caja, en esta página web el médico podrá crear nuevos usuarios de pacientes y podrá monitorizar el estado la información de paciente y el estado de la caja y sus compartimentos.

Para el desarrollo de la página web se utilizó las herramientas Javascript, Html, Css y Bootstrap.

HTLM, CSS, JavaScript son tecnologías más usadas para creación de sitios web, HTML *(HyperTExT Markup Languaje*), aporta a la página web, la estructura de la página mediante etiquetas que indican tipos de elementos como párrafos, encabezados, tablas, imágenes y videos.

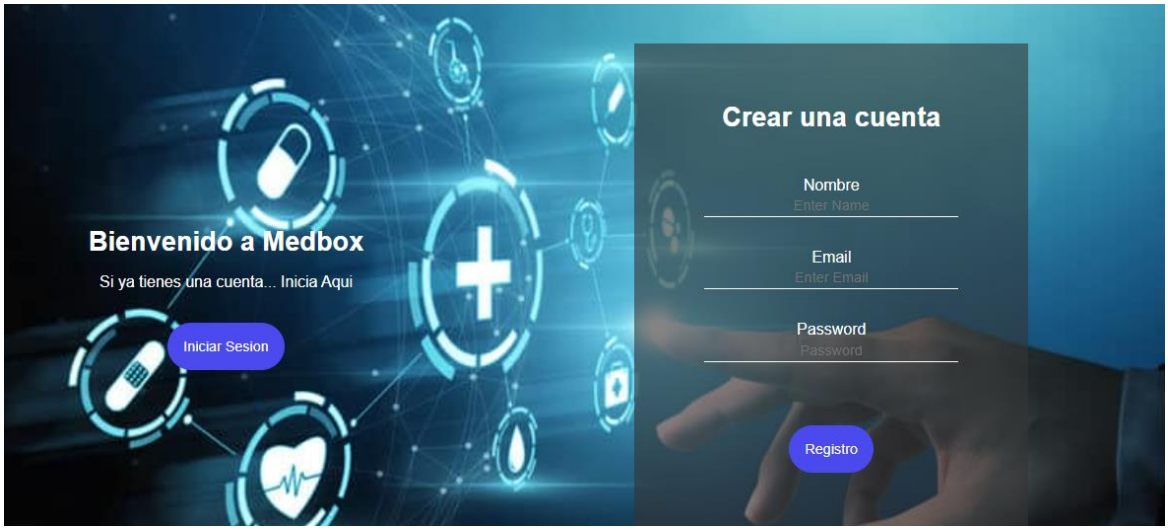
Por un lado, CSS (*Cascading Style Sheet*) permite dar estilo y formato al contenido HTML, como colores de fondo tipos de letras, márgenes, etc. Mientras que, JavaScript es un lenguaje de programación que permite que los elementos de la página web sean dinámicos, además que permite el procesamiento de los datos enviados y recibidos.

Adicional a estos lenguajes se utilizó el Framework Bootstrap, es decir un esquema o marco de trabajo, que permite hacer aplicaciones web adaptable a diferentes tamaños de dispositivos, además, facilita el desarrollo web haciéndolo más rápido.

4.5.1.1. Ingreso y Registro a la página web

Antes de acceder al sistema de gestión, los usuarios tienen que ingresar con un usuario y contraseña. Si el usuario se encuentra en el servicio de Firebase *Autentication,* y si no se encuentra en el sistema, el médico puede registrarse, En Figura 26, se muestra la interfaz de la página web para el ingreso al sitio.

Figura 26. *Ingreso y Registro Médicos*



Fuente: Elaboración propia

Para los nuevos usuarios lo que el sistema hace es que recopila los datos de correo electrónico y contraseña, y los guarda en Firebase. Para esto, Firebase *Autentication* agrega nuevos usuarios a través de la función de crear nuevos usuarios, como se muestra en Figura 27, esta función necesita un *auth*, que es un llamado al módulo de autenticación de Firebase, un correo electrónico que se guarda en la variable email y una contraseña guardada en la variable *password.*

Figura 27. *Función nuevo usuario medico*

```
createUserWithEmailAndPassword(auth, email, password)
.then((userCredential) => {
  // Signed in
  const user = userCredential.user;
  alert("usuario creado correctamente");
  console.log(email,password,usuaromed);
  window.location="home.html";
})
.catch((error) => {
  const errorCode = error.code;
  const errorMessage = error.message;
  alert(errorMessage + errorCode );
  // ..
});
```

Fuente: Elaboración propia

Para el ingreso de usuarios ya creados, se utiliza la función ingreso igual que la función de creación de usuario, se necesita un correo electrónico y una contraseña, pero compara con datos de Firebase *autentication*, con los datos ingresado, si coincide con algún usuario el sistema lo redirige a la página principal, sino el acceso es denegado y el sistema envía un cuadro de dialogo con una alerta de error. La función se muestra en la Figura 28.

Figura 28. *Función ingreso usuarios*

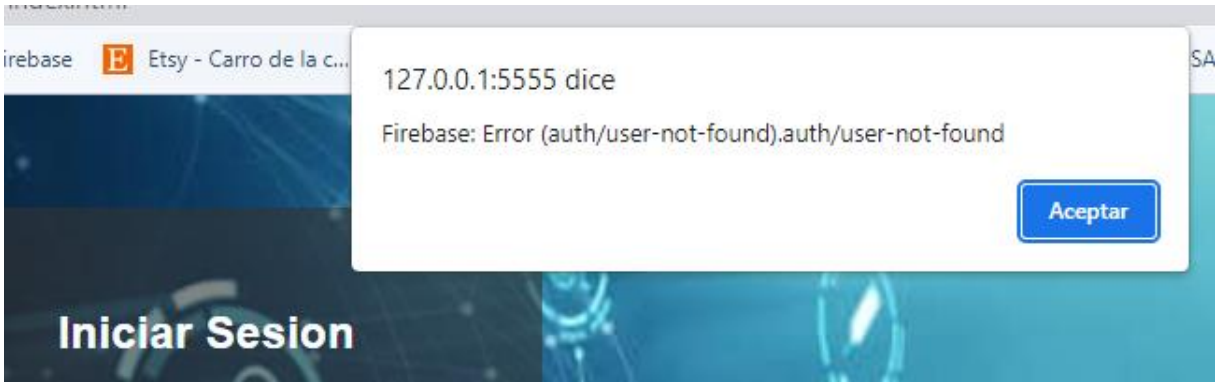
```
signInWithEmailAndPassword(auth, email, password)
.then((userCredential) => {
  // Signed in
  const user = userCredential.user;
  alert("Bienvenido");
  console.log(email,password);
  window.location="home.html";

})
.catch((error) => {
  const errorCode = error.code;
  const errorMessage = error.message;
  alert(errorMessage + errorCode );
});
```

Fuente: Elaboración propia

En la Figura 29, se observa un error cuando el usuario no está autenticado en la Firebase, se envía una alerta de error, diciendo que el usuario no fue encontrado, y esto provoca que el usuario no pueda ingresar a la página web.

Figura 29. *Ingreso Fallido al sistema*



Fuente: Elaboración propia

4.5.1.2. Acceso Autorizado a la página web

Luego de que el medico ingresa correctamente a la página web es direccionado a la página principal mostrada. En la Figura 30, se muestra un panel de navegación compuesta por un menú "*Home*" que se refiere a la página principal, además de una pestaña "Pacientes", donde están alojados los pacientes y sus datos, y un botón para salir de la página web.

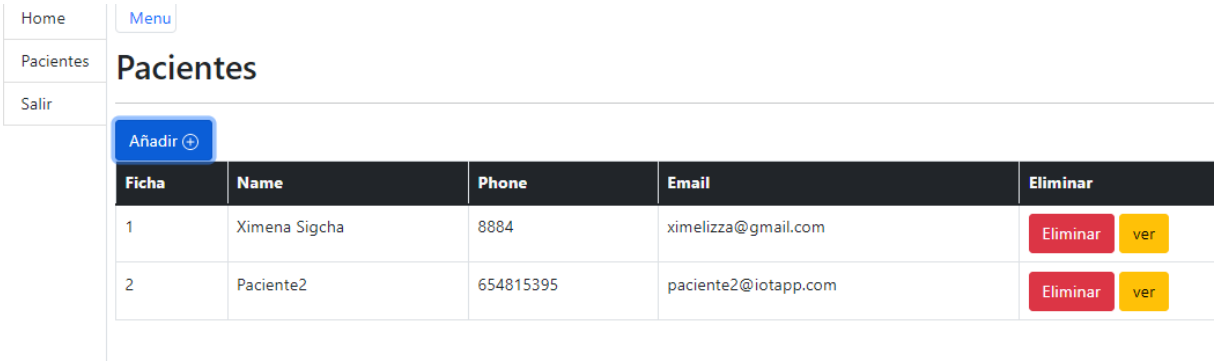
Figura 30 . *Página Principal de la web*

Fuente: Elaboración propia

En la Figura 31, se muestra la pestaña paciente, aquí el médico tratante agrega a nuevos pacientes en una tabla que muestra todos los usuarios actualmente guardados.

Cada paciente cuenta con un botón ver, y otro eliminar, el botón ver nos redirige a la página, en donde se puede visualizar datos del paciente. Si el paciente tiene una caja de medicamentos, se muestra la información para realizar el seguimiento de la caja de medicamentos.

Figura 31. *Pestaña Pacientes*



Fuente: Elaboración propia

Al presionar en el botón Añadir se abre una ventana emergente (Figura 32) en donde se puede añadir información para crear un nuevo paciente.

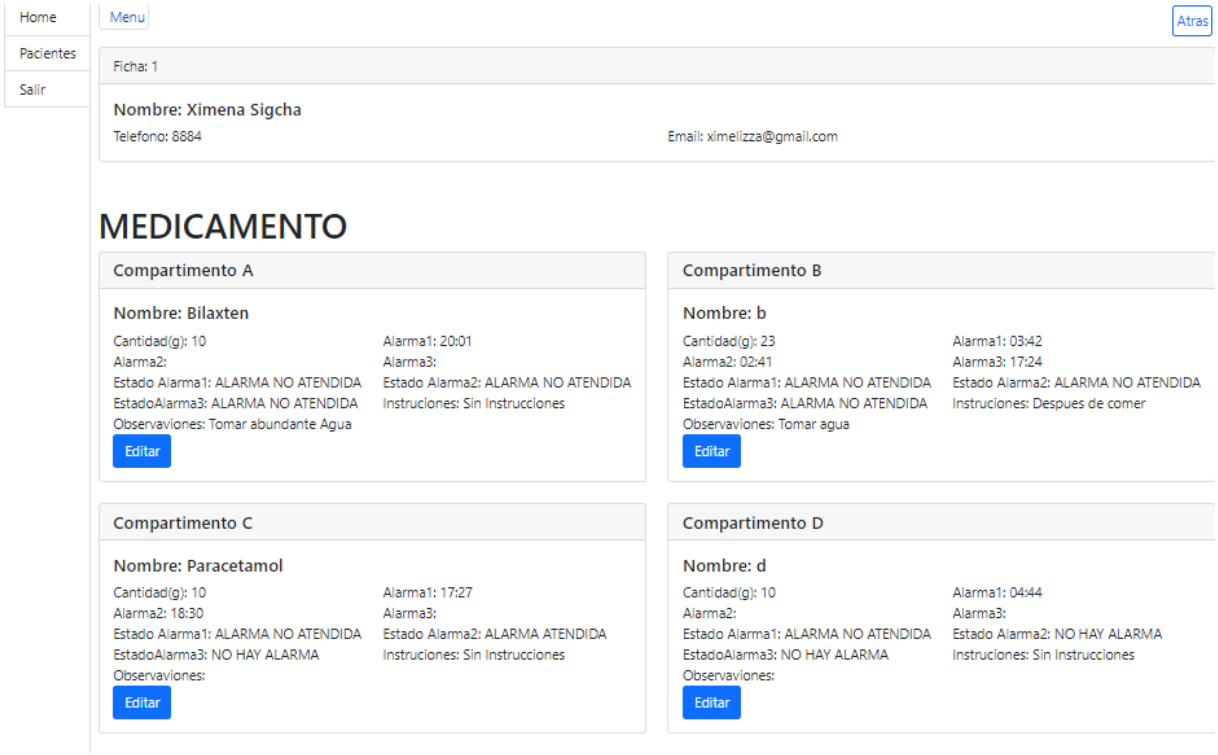
Figura 32. *Añadir nuevo paciente*

Fuente: Elaboración propia

4.5.1.3. Página Información pacientes

Al presionar el botón Ver en cualquier paciente, se abre una página con la información del paciente. En la Figura 33, se observa al paciente que tiene en su poder una caja de medicamentos, primero se observa la información básica del paciente, seguido de 4 áreas que son los compartimentos A, B, C y D, y se puede observar el estado de las alarmas.

Figura 33. *Interfaz paciente*



Fuente: Elaboración propia

4.5.2. Desarrollo aplicación móvil

La aplicación móvil está diseñada para que el usuario de la caja o sus familiares puedan visualizar, la información del tratamiento farmacéutico y el estado de las alarmas.

Para el desarrollo de la aplicación móvil se ha utilizado Android Studio que es un entorno de desarrollo integrado (IDE), para este proyecto la aplicación fue programada con el lenguaje

Java. La aplicación cuenta con una página principal y 4 páginas correspondientes a cada compartimento.

4.5.2.1. Página Principal

En la Figura 34, se puede visualizar la estructura de la página principal, en esta página se observa el estado de la caja y 4 botones, que redirigen a la página de cada compartimento.

Figura 34. *Página principal app Android*

Fuente: Elaboración propia

El estado de la caja puede tener dos opciones encendido o apagado, el propósito del estado de la caja, es que cuando la caja este encendida, los leds y el zumbador pueden funcionar normalmente. Sin embargo, si la caja esta apagada las alarmas no dan ningún aviso de alarma porque los leds y el zumbador permanecen apagadas todo el tiempo, pero la página web, la aplicación Android y la base de datos siguen funcionando.

En la Figura 35, se observa las librerías empleadas para conectar la aplicación a Android con Firebase, estas librerías estarán en la cabecera de todos los programas Android.

Figura 35. *Bibliotecas Android/Firebase*

```
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

Fuente: Elaboración propia

En la figura se observa que la aplicación se conecta Firebase mediante la creación una instancia Firebase (primera línea Figura 36) y la referencia Firebase se refiere al lugar de donde se obtendrá la información para mostrar en la aplicación (segunda fila Figura 36).

Figura 36. *Instancias Firebase*

```
FirebaseDatabase database= FirebaseDatabase.getInstance();
DatabaseReference cajastateRef = database.getReference( path: "caja/Caja_state");
```

Fuente: Elaboración propia

Posteriormente se hace un llamado a la función *OnDataChange* de Firebase, esta función permite obtener la información del nodo seleccionado. En la Figura 37, se observa la función para recuperar datos desde la base de datos, la información se obtiene del nodo caja/Caja_state, esta información es guardada en una variable *string* llamada boxstatevlaue y posteriormente mostrada en la aplicación a través de una etiqueta de texto.

Figura 37. *Insertar información en la base de datos*

```
    DatabaseReference cajastateRef = database.getReference( path: "caja/Caja_state");
    cajastateRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            String boxstatevlaue= dataSnapshot.getValue(String.class);
            estatecaja.setText("Caja " + boxstatevlaue);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
```

Fuente: Elaboración propia

En la Figura 38 se muestra la función utilizada en los botones con la finalidad de navegar entre compartimentos.

Figura 38. *Función cambio de página app Android*

```
public void irAcompartA(View view){
    //this esta ventana y a la que queremos ir
    Intent i=new Intent( packageContext: this,compartA.class);
    startActivity(i);
}
```

Fuente: Elaboración propia

4.5.2.2. Página Compartimento

Al presionar un botón de compartimento en la página principal esta muestra la página con la información alojada en la base de datos de este compartimento, las pestañas de compartimentos son solo lectura.

En la Figura 39, se muestra la información del compartimento C

Figura 39. *Página compartimento*

Fuente: Elaboración propia

Para recuperar la información de la base de datos se utiliza la misma función que se utilizó para leer el valor de Caja_state, pero cambiando el nodo del que se lee (A, B, C, D) (Figura 40)

Figura 40. *Referencia a nodo hijo*

```
cajARef.child("caja/A").addValueEventListener(new ValueEventListener() {
```

Fuente: Elaboración propia

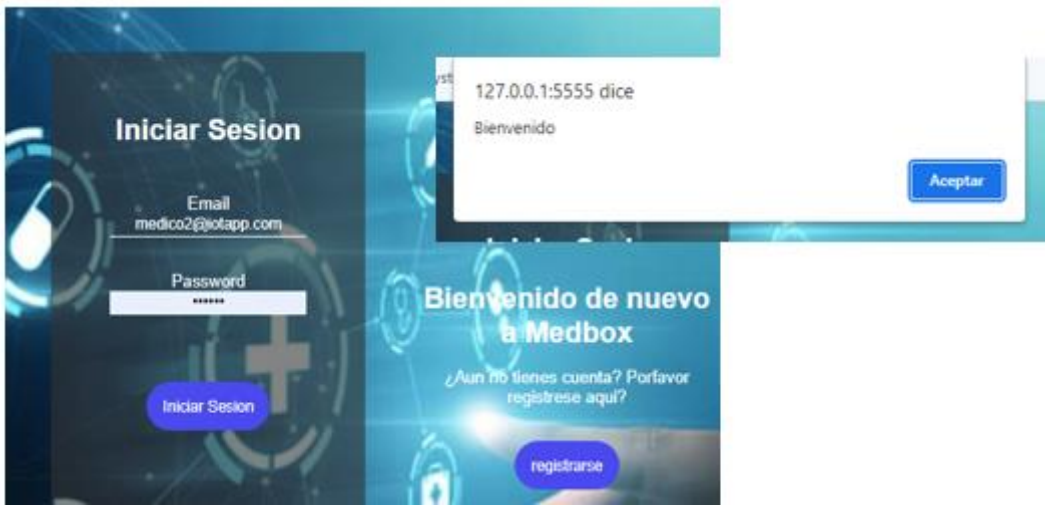
4.6. Integración del sistema y Pruebas

En este apartado se explicara el funcionamento del sistema mediante pruebas.

4.6.1. Prueba funcionamiento página web

En la Figura 41, se muestra el ingreso de un usuario médico a la página web, al comprobar sus credenciales recibe el mensaje de bienvenida y posteriormente es redirigido a la página principal.

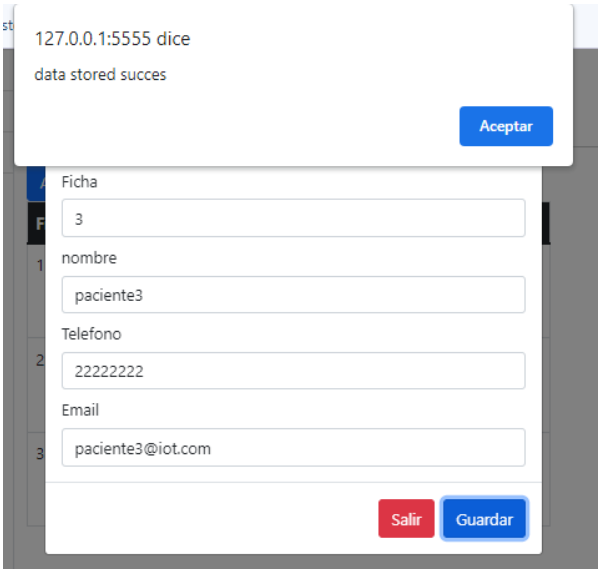
Figura 41. *Ingreso a la plataforma*



Fuente: Elaboración propia

Para que el medico pueda tener interacciones con los pacientes, tiene que dirigirse a la pestaña pacientes, en donde podrá ver un listado de pacientes y la posibilidad de añadir pacientes a la base de datos. En la Figura 42,se muestra el formulario que el médico tiene que llenar para crear un paciente.

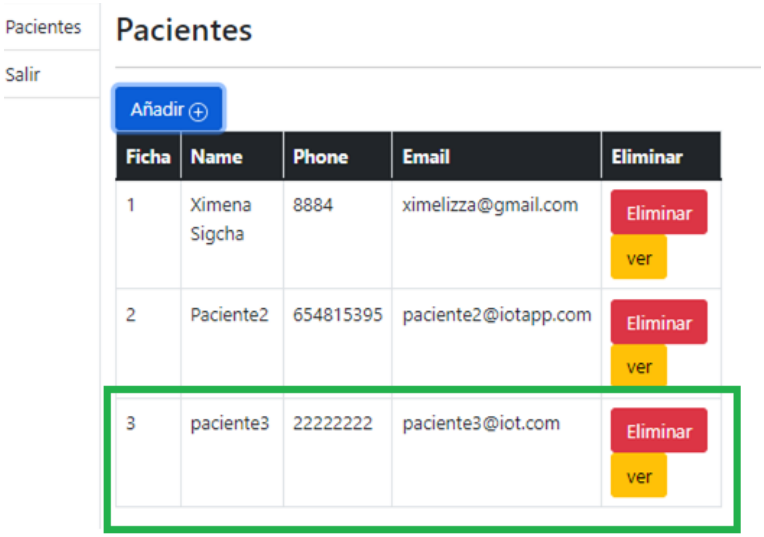
Figura 42. *Paciente Agregado*



Fuente: Elaboración propia

Al presionar el boton guardar, se puede observar el listado de los pacientes incluido el nuevo paciente (Figura 43).

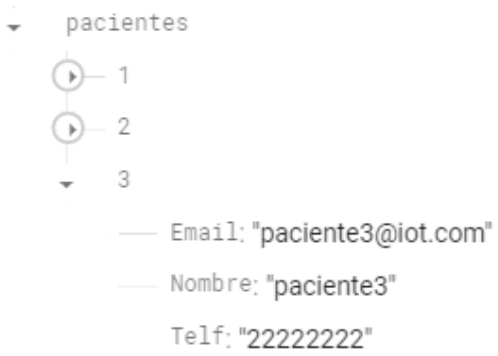
Figura 43. *Lista de pacientes*



Fuente: Elaboración propia

Mediante este paso, se añade el paciente a la base de datos de firebase en el nodo llamado pacientes.(Figura 44).

Figura 44. *Nodos pacientes*

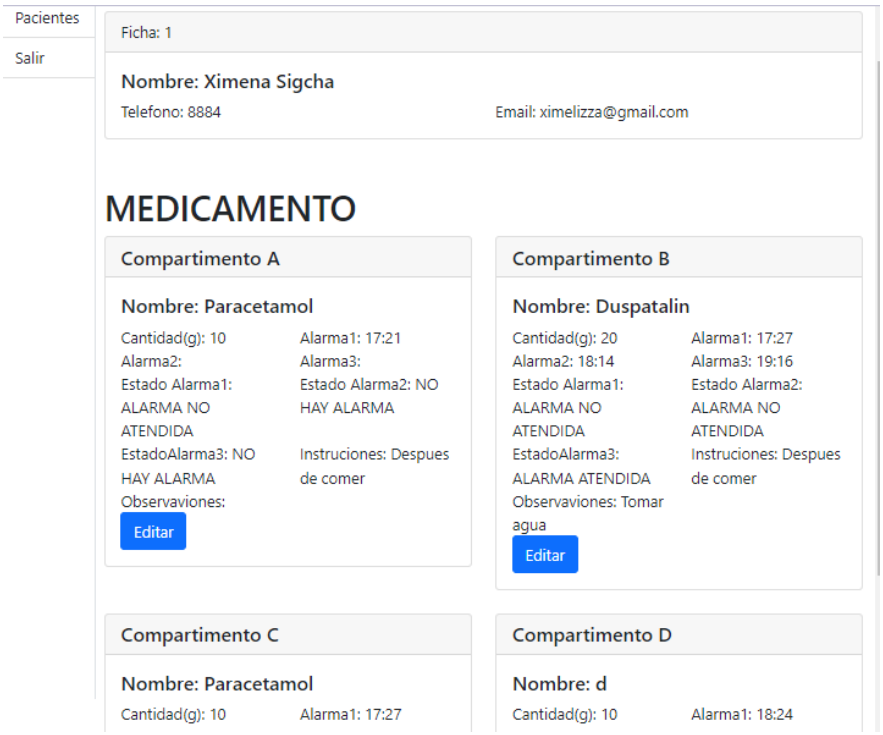


Fuente: Elaboración propia

El sistema por el momento, solo con una persona cuenta con acceso a la función de seguimiento de medicamentos, es el usuario de la caja.

Este paciente es el paciente número 1 o Ximena Sigcha en la Figura se observa la página de datos del usuario caja (Figura 45).

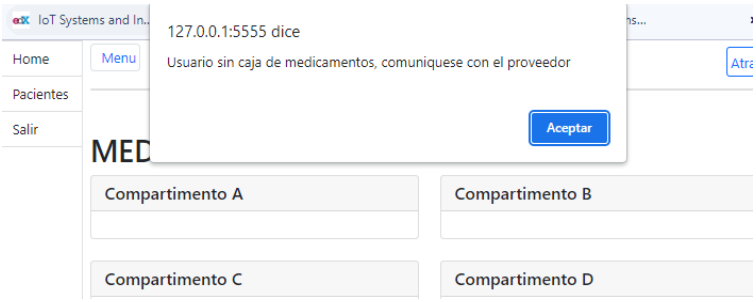
Figura 45. *Interfaz usuario caja medicamentos*



Fuente: Elaboración propia

Para los usuarios que no tienen una caja de medicamentos el sistema pide al medico que utilize el sistema de gestión, comunicarse con el provedor de las cajas de medicamentos. (Figura46). Y evidentemente no muestra información de ningún compartimento debido a sus inexistencia.

Figura 46. *Usuario sin caja de medicamentos*

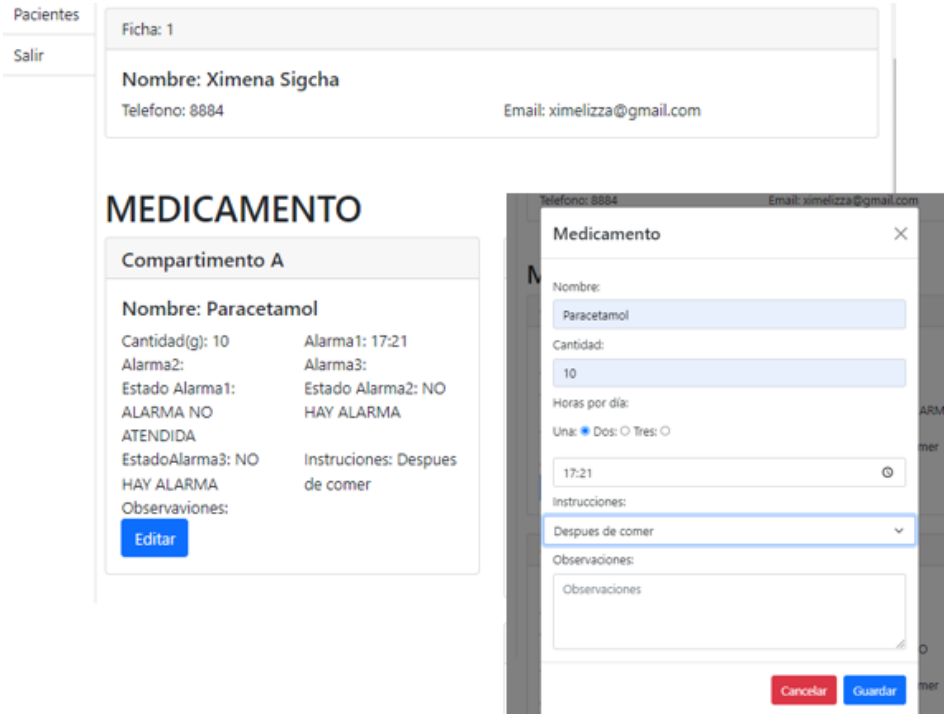


Fuente: Elaboración propia

4.6.2. Prueba programacion de alarmas en compartimentos

En la página de datos del ususario de la caja, el medico puede configurar informacion del tratamiento farmaceutico, mediante el formulario Medicamento, que se abre al presionar el boton de editar de cada compartimento. En la Figura 47, se observa la configuracion de una alarma e informacion del medicamento en el compartimento A.

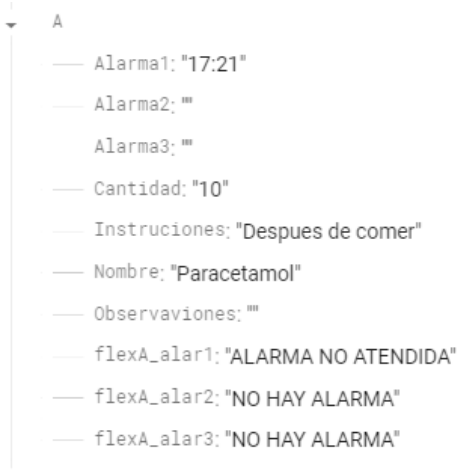
Figura 47. *Formulario editar medicamentos*



Fuente: Elaboración propia

Una vez presionado el boton guardar, se actualiza el medicamento y sus instrucciones en la página web y los guarda en la base de datos (Figura 48), en el nodo caja, su nodo hijo A (caja/A).

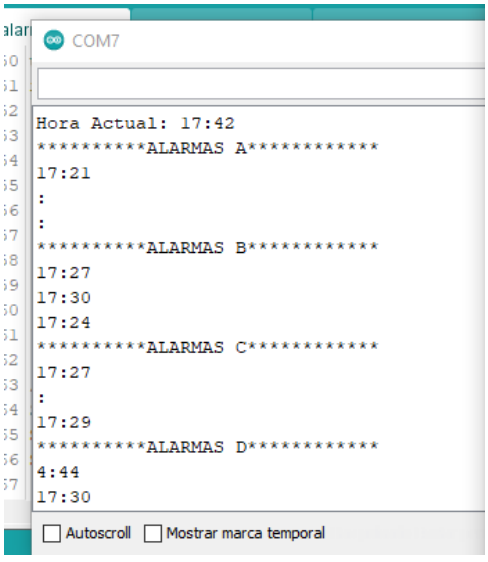
Figura 48. *Base de datos actualizada luego de ingresar datos a el formulario medicamento*



Fuente: Elaboración propia

Al mismo tiempo la información de alarmas de todos los compartimentos es enviada al IDE de Arduino, y procesada por el microcontrolador ESP2866, se puede observar que cuando no existe alarma, nos imprime ":". (Figura 49)

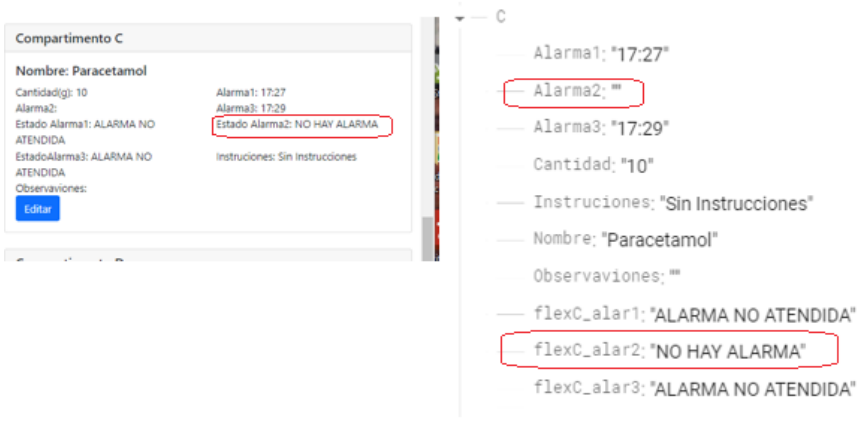
Figura 49. *Alarmas recuperadas de Firebase en Arduino ID*



Fuente: Elaboración propia

Un ejemplo en particular es el compartimento C, en donde se puede observar todas las opciones de alarmas, pueden existir 3 alarmas para el medicamento, pero también pueden existir campos vacíos de medicamentos, en ese caso, la caja escribe en la base de datos que no existe alarma (Figura 50)

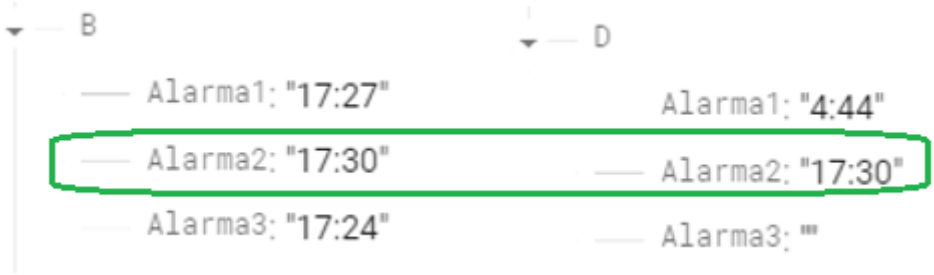
Figura 50. *Compartimento C*



Fuente: Elaboración propia

Una vez, que las alarma esten correctamente configuradas ,se puede hacer pruebas en la caja. Se ha propuesto dos alarmas, a la misma hora para compartimentos diferentes, es decir se puede tomar dos diferentes medicamentos al mismo tiempo. Entonces partimos de la base de datos en la figura se observa los nodos B y D tienen una alarma a las 17:30 de la tarde.(Figura 51).

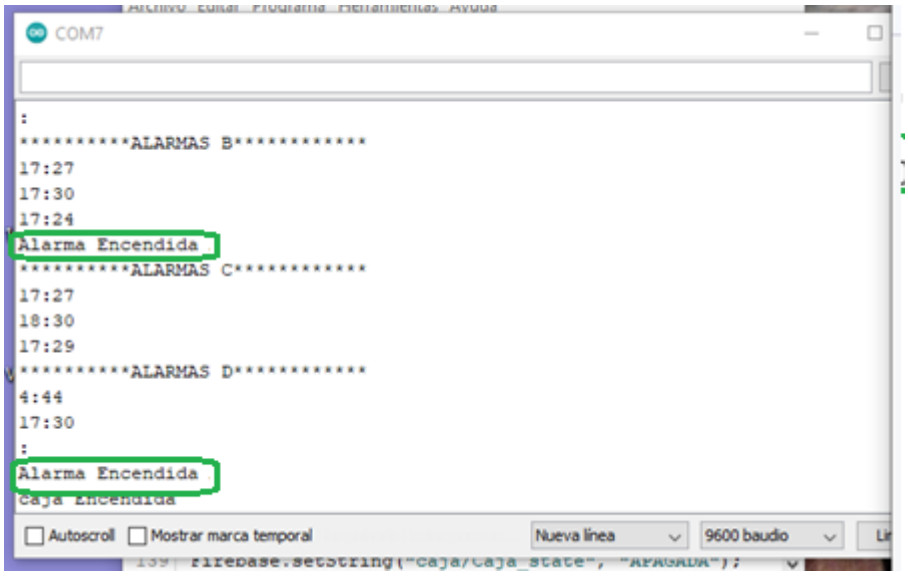
Figura 51. *Alarma B y C programadas en la base de datos*



Fuente: Elaboración propia

En la Figura 52, se observa las alarmas leidas por el ESP2866, mientra que el microcontrolador ejecuta la funcion que crea alerta de alarmas. Como se puede ver, informa que existe alarmas encendidas en dichos nodos.

Figura 52. *Alarmas Encendidas*



Fuente: Elaboración propia

Físicamente lo que ocurre en la caja de medicamentos es que los leds de esos compartimentos se encienden a la hora de la alarma, acompañados del sonido de en el zumbador (Figura 53).

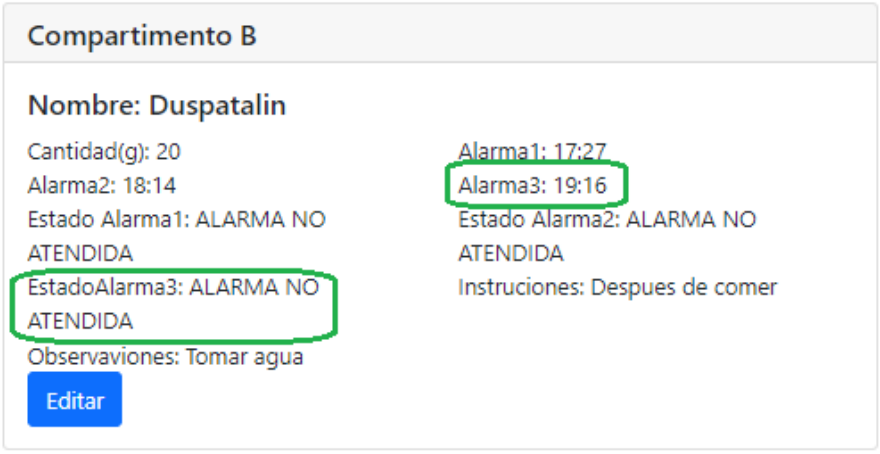
Figura 53. *Caja de medicamentos alarma encendida*

Fuente: Elaboración propia

4.6.3. Pruebas con los sensores flex (Integracion total del sistema)

Según la programacion antes descrita los sensores flex, activan solo si existe una alarma, por lo tanto esta prueba es la integracion de todo el sistema. Para esta prueba se ha programado una alarma en el compartimento B, a las 19:16 de la tarde. En la Figura 54, se observa la programacion de la alarma en el sistema de gestión.

Figura 54. *Compartimento B programada alarma*



Fuente: Elaboración propia

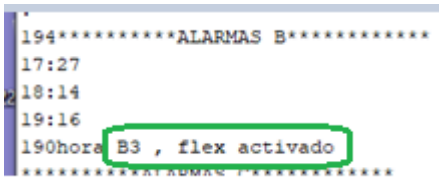
En la Figura 55, se observa que la alarma 3, ha sido cargada a la base de datos.

Figura 55. *Alarma programada base de datos.*

Fuente: Elaboración propia

En la Figura 56, se observa que el ESP2866 ha recibido la alarma programada y activa la recolección analógica del sensor flex de compartimento B.

Figura 56. *Sensor Flex del compartimento B activado*



Fuente: Elaboración propia

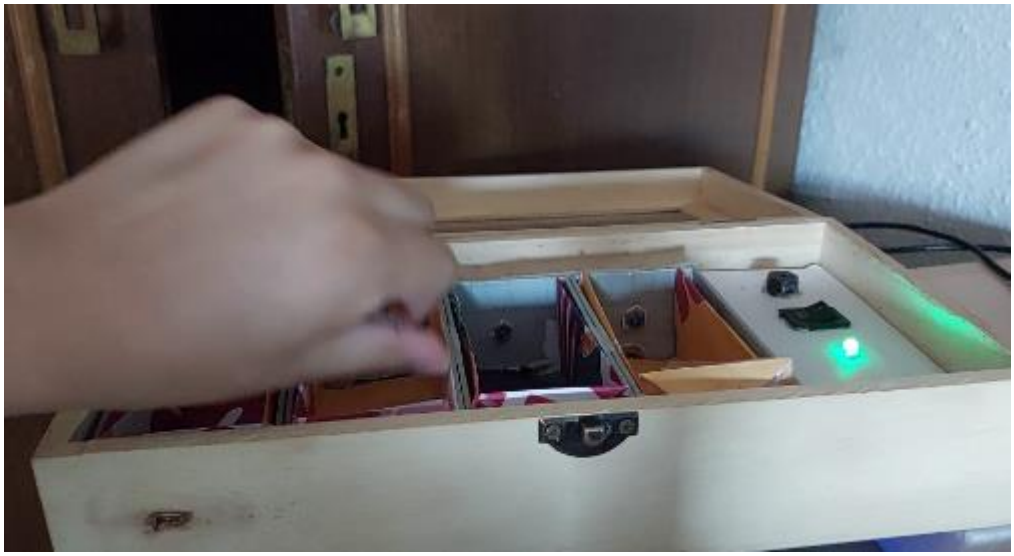
En la Figura 57, se observa la caja de medicamentos y su respectiva alerta de alarma a las 19:16 de la tarde.

Figura 57. *Alarma compartimento B*

Fuente: Elaboración propia

En el momento en que la alarma suena, al introducir la mano, los valores del sensor Flex disminuyen, y encienden el led verde, este led indica que el compartimento B ha sido tocado o flexionado, y el programa procede a escribir en la base de datos “ALARMA ATENDIDA” (Figura 58).

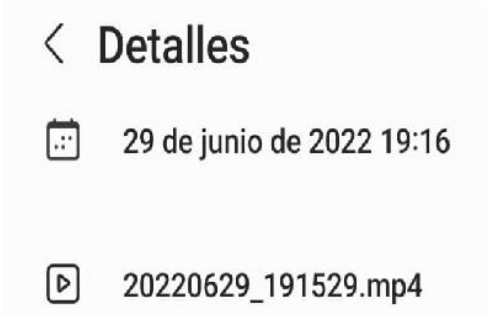
Figura 58. *Alarma atendida*



Fuente: Elaboración propia

El proceso del funcionamiento de la caja ha sido grabado, por lo tanto, se ha podido obtener la hora en que se está probando la alarma (Figura 59).

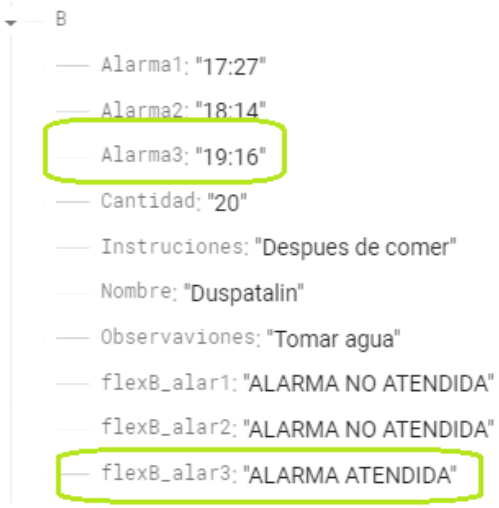
Figura 59. *Hora caja de medicamentos*



Fuente: Elaboración propia

La base de datos se sobre escribe el nuevo estado de la alarma a “ALARMA ATENDIDA”, en la Figura 60, se observa la base de datos actualizada después de tomar los medicamentos.

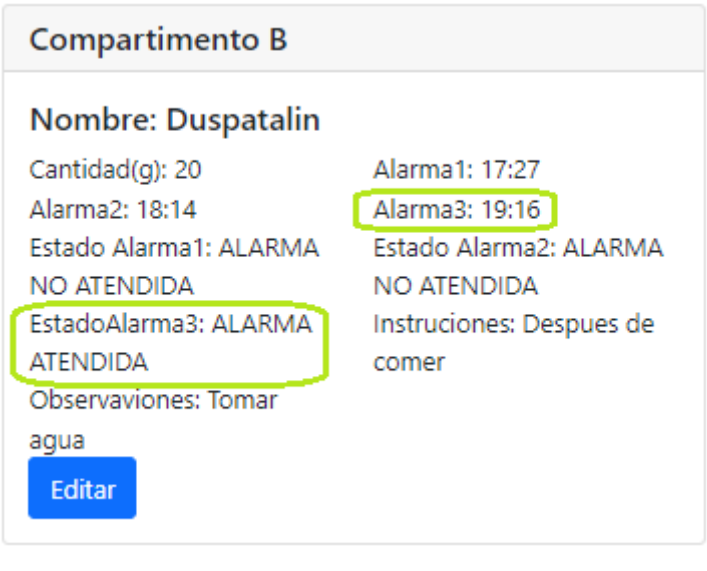
Figura 60. *Base de datos actualizada por lo sensores*



Fuente: Elaboración propia

Finalmente, los usuarios (médico y paciente usuario de la caja) pueden observar el estado de las alarmas y el tratamiento farmacéutico. En la Figura 61, se observa la interfaz del médico a través de la página web.

Figura 61. *Información en página web actualizada*



Fuente: Elaboración propia

En la Figura 62, se muestra la misma información para el paciente o un tutor de este mediante la aplicación Android.

Figura 62. *Aplicación Android actualizada*

Fuente: Elaboración propia

5. Conclusiones y trabajo a futuro

5.1.Conclusiones

El presente trabajo ha sido motivado por la necesidad de solucionar un problema que puede aparecer cuando una persona está siendo medicada, la propensión a olvidar los horarios de la ingesta medicamentos, lo que produce que estos se consuman de una forma inadecuada, provocando interferencias en el tratamiento médico, e incluso poner en riesgo la salud de los pacientes.

Este proyecto y la solución presentada son funcionales para cualquier persona, sin embargo, se ha hecho énfasis en las necesidades de los adultos mayores, puesto que son personas que necesitan mayor atención y ayuda. Por esta razón este proyecto, puede ser clave para mitigar equivocaciones en la ingesta de un tratamiento farmacéutico, ya que, por un lado, el sistema propuesto consta con un sistema de alarmas para los medicamentos, que son una guía visual y auditiva para la ingesta de dichos medicamentos, pero también permite el monitoreo de dicho tratamiento a través de las aplicaciones web y móvil, por parte del personal de salud. De esta manera se puede mejorar la adherencia a los tratamientos a base de medicamentos y tener un mayor control del paciente.

Tras concluir con el desarrollo de este trabajo de fin de máster, se han obtenido las siguientes aportaciones:

1. Se ha construido una caja de medicamentos totalmente funcional, y cómoda para usuario, es decir se ha evitado tener cables expuestos que pueden ser molestos para el usuario o para éxito del proyecto.

2. Se ha creado un sistema de gestión y monitoreo para médicos, que se conecta de forma exitosa con la caja de medicamentos IoT y la base de datos.

3. Se ha implementado una aplicación Android que se conecta al sistema de monitoreo, que permite, visualización de información del tratamiento farmacéutico para el usuario de la caja, esto le permite llevar un control de su medicamento y sus respectivas instrucciones.

4. Se ha utilizado Edge Computing para el procesamiento de la información proveniente de los sensores, transformando dicha información en información entendible para el usuario final (médicos y usuarios de la caja).

5. Mediante el uso de la Base de Datos de Firebase se ha logrado centralizar la información de todo el proyecto, permitiendo que la información alojada en la nube se pueda leer y escribir en todo momento que sea requerida dicha información.

Considerando estos resultados obtenidos en el desarrollo de este proyecto se ha concluido que se han cumplido los objetivos específicos planeados en este TFE, ya que:

Se ha diseñado y construido una caja de medicamentos inteligente, que puede ser utilizada por cualquier persona que tenga acceso y autorización para el uso de esta, sin embargo, tiene algunas funcionalidades que fueron pensadas principalmente para adultos mayores. Entre estas funcionalidades se destacan: La caja de medicamento tiene cuatro compartimentos (A, B, C, D), cada compartimento tiene implementado un sensor Flex casero, hecho con material Velostat (con la forma de cada compartimento).

El material Velostat, disminuye su resistencia cuando es flexionado, presionado o doblado, lo que implica que cada compartimento está en reposo con una resistencia inicial de reposo hasta que se introduzca o extrae las manos del usuario para tomar su medicamento. La utilización de este material puede ser considerada como una solución alternativa y de bajo coste para la implementación de sensores específicos para esta aplicación, situación que puede facilitar la implementación y producción de este sistema en aplicaciones reales. Además, el uso de material Velostat ha permitido construir una caja de medicamentos totalmente cómoda para usuario, evitado tener cables expuestos que pueden ser molestos para el usuario, de este modo mejorando su usabilidad.

Otra funcionalidad importante está relacionada con la utilización de alertas auditivas y visuales. En el sistema propuesto, cada compartimento de la caja de medicamentos dispone de un led que se enciende cuando existe una alerta de alarma en dicho compartimento, además de un zumbador que funciona a la par de los leds. Esta funcionalidad visual y auditiva muestra la capacidad de interacción de los dispositivos con el usuario final en este caso,

evitando que haya confusiones en la toma de medicamentos. Incluso el usuario no necesita recurrir a la aplicación móvil para saber cuál medicamento tomar o recurrir a correos electrónicos, mensajes de texto, etc. De este modo reduciendo la carga de utilización al usuario tipo (adultos mayores) quienes en su mayoría tiene capacidades reducidas en el manejo de tecnologías y dispositivos como los teléfonos móviles y computadores.

También se ha diseñado e implementado un sistema de gestión y monitoreo, que permite interactuar al paciente (usuario de la caja) y el médico, a través de una página web que se actualiza en tiempo real, es decir el medico tiene acceso a la información personal y seguimiento del tratamiento farmacéutico, situación que puede ayudar a mejorar la gestión de tratamientos médicos de manera remota. Además, el paciente tiene acceso a la información e indicaciones dadas por el medico a través de una aplicación móvil amigable e intuitiva que en un futuro podría ayudar a mejorar la adherencia a los tratamientos.

Por otro lado, para el diseño del sistema de gestión y monitorización se ha tenido las siguientes consideraciones:

En vista de que existe la posibilidad en que un adulto mayor no sepa usar o no tenga acceso a un computador o el internet, en este estudio se optó por la adición de una aplicación móvil, con la finalidad de proveer un mecanismo que resulte más manejable y accesible que un computador. No obstante, se puso énfasis en desarrollar una aplicación sencilla en su manejo, que no requiera, la utilización de formularios y que únicamente permita la lectura, y obtención de información del estado de la caja (encendida o apagada) como referencia al usuario.

Por otro lado, la aplicación web fue diseñada pensando el médico, que por lo general trabaja junto a un computador, entonces para el acceso a esta plataforma, provee de funcionalidad avanzadas como de agregar y eliminar pacientes, revisar el perfil de cada paciente (monitoreo), y programar hasta tres alarmas en el día del medicamento. En un futuro esta esté tipo de gestión automatizada podría a ayudar a mejora el seguimiento continuo y personalizado de los tratamientos farmacéutico (gestión) e incluso ayudar a verificar la eficacia de los mismo.

Por consiguiente, se puede concluir que se ha logrado implementar un sistema de gestión y monitoreo de toma de medicamentos basado en sensores y dispositivos IoT, dado que, se ha diseñado un sistema de gestión y monitoreo que trabaja en conjunto con una caja de medicamentos inteligente. La caja de medicamentos se ha convertido en un dispositivo IoT debido a que integra una serie sensores y cuenta con la capacidad para conectarse a internet de forma bidireccional (enviar y recibir información) para generar información útil para el monitoreo y gestión de la ingesta de medicamentos.

Con respecto a la parte técnica del desarrollo de la propuesta, se ha logrado conectar la caja de medicamentos y el sistema de gestión y monitorización mediante servicios en la nube. Para ser específicos el tipo de servicio en la nube utilizado fue el servicio Baas, implementado mediante Firebase y sus servicios "Real time database" y "*Firebase Autentication* ". Con respecto a este último punto, se concluye que con la utilización de la Base de Datos de Firebase es posible centralizar la información de todo el proyecto, permitiendo que la información alojada en la nube se pueda leer y escribir en todo momento que sea requerida dicha información y esta información llega al usuario a través de aplicaciones web y móviles. Creando un sistema con capacidad de conectarse a la red, e intercambiar datos entre objetos y dispositivos en tiempo real, además con la capacidad de para generar grandes cantidades de información, que brindan nuevas oportunidades tanto en el área médica, así como para el área del análisis de datos para mejorar la de gestión médica de manera remota y mejorar los tratamientos.

5.2. Trabajo a futuro

A través de la realización de este trabajo se han identificado diversas líneas de investigación y trabajo a futuro, como, por ejemplo:

Se podría implementar en la caja de medicamentos, sensores que permitan controlar el estado de los signos vitales del paciente, como glucómetros, sensores de ritmo cardiaco. Además de crear un el ambiente controlado de temperatura de la caja y sus medicamentos, mediante sensores de temperatura, humedad y módulo peltier (sistema de refrigeración). La

información de los signos vitales daría apertura a crear estadísticas y gráficos del estado del paciente, y ser un apartado más en el sistema de gestión web del médico.

El sistema propuesto puede ser evaluado en un contexto real mediante el reclutamiento de un grupo de pacientes y médicos con la finalidad evaluar la capacidad del sistema para mejorar la gestión del medicamente. Además, mediante un experimento de este tipo se podría evaluar de manera formal usabilidad de la propuesta y su impacto la labor clínica.

Para que el sistema sea más robusto se podría añadir, cámaras de video, mediante inteligencia artificial, y visión por computador, se podría hacer modelos de reconocimiento del evento usuario hace uso de la caja de medicamentos y toma de medicamentos.

Además, se podría mejorar el prototipo, mediante impresión en 3D. Se podría rediseñar el prototipo de la caja para que se adapte a las necesidades de trabajos futuros, como, por ejemplo, desarrollando una caja con un mayor número de compartimientos para medicinas o elaborar sistemas con un número diferente de sensores, con el objetivo de mejorar la eficiencia y autonomía energética.

Respecto al sistema de gestión, se podría añadir funciones adicionales de análisis como, por ejemplo, un apartado de estadísticas y gráficas en tiempo real de los sensores, que resulte fácil de utilizar usuarios tipo (médicos) y que les permita extraer información relevante para su práctica médica rutinaria. Además, se podría utilizar de una manera más específica los servicios de computación en la nube para realizar un análisis de datos (Big data) con el fin de identificar y extraer información útil que pueda ser empleada en la gestión médica.

Con respecto a la aplicación Android, se podría mejora su funcionalidad mediante la adición de un módulo de gestión de alarmas, que permita interactuar con la base de datos directamente. Además, en trabajos futuros se puede evaluar el grado de funcionalidad y aceptación de esta ampliación en conjunto con el prototipo de caja desarrollado en este estudio.

Referencias bibliográficas

- Ahmad, S., Hasan, M., Shahabuddin, M., Tabassum, T., & Allvi, M. W. (2020). IoT based pill reminder and monitoring system. International Journal of Computer Science and Network Security, 20(7), 152-158.
- Agrega Firebase al proyecto de JavaScript | Firebase Documentation. (2014, 15 abril). Firebase. Recuperado 21 de abril de 2022, de https://firebase.google.com/docs/web/setup?authuser=0&%3Bhl=es&hl=es
- Aldeer, M., Javanmard, M., & Martin, R. P. (2018). A review of medication adherence monitoring technologies. Applied System Innovation, 1(2), 14.
- Al-Mahmud, O., Khan, K., Roy, R., & Alamgir, F. M. (2020, junio). Internet of things (IoT) based smart health care medical box for elderly people. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.
- Asad M. (2018, febrero). Smart Medicine Box Using IOT. In 2018 International Journal of Scientific & Engineering Research
- Ballerini, R. (2021, 23 julio). HTML, CSS y Javascript, ¿cuáles son las diferencias? ALURA LATAM. Recuperado 19 de mayo de 2020, de https://www.aluracursos.com/blog/html-css-javascript-cuales-son-las-diferencias
- Comparando Arduino y el módulo WiFi ESP8266. (2016). Recuperado 1 de julio de 2021, en: https://polaridad.es/compara-arduino-esp8266/
- Da Silva, D. V., Gonçalves, T. G., & Pires, P. F. (2019, octubre). Using IoT technologies to develop a low-cost smart medicine box. In Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web (pp. 97-101). SBC.
- Definición de IaaS, PaaS y SaaS ¿En qué se diferencian? (2020, 9 enero). Ambit-bst. Recuperado 10 de julio de 2022, de https://www.ambit-bst.com/blog/definici%C3%B3n-de-iaas-paas-y-saas-en-qu%C3%A9-se-diferencian

- Deshpande, S., Choudhari, M., Charles, D., & Shaikh, S. (2018). A smart pill box to remind of consumption using IoT. MES College of Engineering, Pune.
- Flex Sensor. (s. f.). Components101. Recuperado 6 de julio de 2022, de https://components101.com/sensors/flex-sensor-working-circuit-datasheet
- Firebase: qué es, para qué sirve, funcionalidades y ventajas. (2020, 17 mayo). Digital 55. Recuperado 21 de abril de 2022, de https://digital55.com/que-es-firebase-funcionalidades-ventajas-conclusiones/
- García, É. (2021, 11 marzo). Wi-Fi vs. Bluetooth: diferencias, ventajas y desventajas. ADSLZone. Recuperado 10 de julio de 2022, de https://www.adslzone.net/reportajes/wifi/wifi-vs-bluetooth/
- Get started with Bootstrap. (s. f.). Get started with Bootstrap. Recuperado 19 de mayo de 2022, de https://getbootstrap.com/docs/5.0/getting-started/introduction/
- Giraldo, V. (2021, 12 febrero). ¿Ya conoces Firebase? La herramienta de desarrollo y análisis de aplicaciones mobile. Rock Content - ES. Recuperado 21 de abril de 2022, de https://rockcontent.com/es/blog/que-es-firebase/
- Haverbeke, M. (2018). Eloquent javascript: A modern introduction to programming. No Starch Press.
- IBM Cloud Education. (2022, 18 marzo). IaaS frente a PaaS frente a SaaS. IBM. Recuperado 10 de julio de 2022, de https://www.ibm.com/es-es/cloud/learn/iaas-paas-saas
- Industries, A. (s. f.). ADS1015 12-Bit ADC - 4 Channel with Programmable Gain Amplifier. Adafruit. Recuperado 6 de julio de 2022, de https://www.adafruit.com/product/1083
- Industries, A. (s. f.-b). Pressure-Sensitive Conductive Sheet (Velostat/Linqstat). Adafruit. Recuperado 6 de julio de 2022, de https://www.adafruit.com/product/1361
- IR Proximity sensor. (s. f.). Silicon TechnoLabs. Recuperado 6 de julio de 2022, de https://www.silicontechnolabs.in/ir-proximity-sensor-silicon-technolabs?search=ir%20pro

- Karagiannis, D., & Nikita, K. S. (2020, agosto). Design and development of a 3D Printed IoT portable Pillbox for continuous medication adherence. In 2020 IEEE International Conference on Smart Internet of Things (SmartIoT) (pp. 352-353). IEEE.
- Moreno, I. (2022, 10 julio). ¿Cuál es el mejor servidor en la nube: Amazon AWS o Firebase? [Comentario sobre el artículo "Amazon AWS o Firebase"]. Quora. https://es.quora.com/Cu%C3%A1l-es-el-mejor-servidor-en-la-nube-Amazon-AWS-o-Firebase
- PDP Banner Version 2 1083 - ADS1015 12-Bit ADC board, Adafruit. (s. f.). [Imagen]. ADS1015 Adafruit. https://www.distrelec.de/en/ads1015-12-bit-adc-board-adafruit-1083/p/30091130
- (PDF) MFRC522 Datasheet - Standard performance MIFARE and NTAG frontend. (s. f.). Datasheet Recuperado 6 de julio de 2022, de http://www.datasheet.es/PDF/659135/MFRC522-pdf.html
- Poiala, C. (2015, 2 marzo). Cloud offering: Comparison between IaaS, PaaS, SaaS, BaaS. Assist-software. Recuperado 30 de junio de 2022, de https://assist-software.net/blog/cloud-offering-comparison-between-iaas-paas-saas-baas
- ¿Qué es la arquitectura de edge computing? (2021, 14 mayo). RedHat. Recuperado 11 de julio de 2022, de https://www.redhat.com/es/topics/edge-computing/what-is-edge-architecture?sc_cid=7013a000002pu3wAAA&gclid=Cj0KCQjwlK-WBhDjARIsAO2sErTm3apRUQvpyctuvj3tsv_xRXzai028jKBTAmB7UP5C82UywVBNL7kaApmPEALw_wcB&gclsrc=aw.ds
- Rebato, C. (2022, 14 abril). Qué es el Edge Computing, explicado de manera sencilla. Think Big. Recuperado 11 de julio de 2022, de https://empresas.blogthinkbig.com/edge-computing-que-es/
- Salario para Ingeniero De Proyectos en España - Salario Medio. (s. f.). Talent.com. Recuperado 12 de julio de 2022, de https://es.talent.com/salary?job=ingeniero+de+proyectos#:%7E:text=El%20salario%20ingeniero%20de%20proyectos,hasta%20%E2%82%AC%2037.300%20al%20a%C3%B1o.

- Singh, B., Bhattacharya, S., Chowdhary, C. L., & Jat, D. S. (2017). Recuperado en internet of things and its applications in healthcare. Journal of Chemical and Pharmaceutical Sciences, 10(1), 447-452.
- Ruas, Octavio O. (2020). Internet de las cosas en salud IoMT-octubre 2020. 10.13140/RG.2.2.23348.27520.
- Rushikesh J., Gajanan B., Jyotsna M., &Yogita P. (2019, diciembre). Intelligent Pillbox for Monitoring the Healt using Iot Concepts.
- Valdes, A. (2021). OPS/OMS | Uso racional de medicamentos y otras tecnologías sanitarias. Retrieved 27 May 2021, De: https://www3.paho.org/hq/index.php?option=com_content&view=article&id=1417:2009-uso-racional-medicamentos-otras-tecnologias-salud&Itemid=1180&lang=es
- 5 requerimientos de una arquitectura IoT - Chakray. (2021). Recuperado 1 de julio de 2021, en: https://www.chakray.com/es/5-requesitos-de-una-arquitectura-iot/

Anexo A. Código Arduino

Librería lib_red.h

```
#include <ESP8266WiFi.h>
#include <strings_en.h>
#include <WiFiManager.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h> //levantar web nombre y server
void conectaWifi(){
  Serial.begin(9600);
//objeto de la clase wifimanager
WiFiManager;
//descomentar para reset configuracion
//wifiManager.resetSettings();
//creamos AP y portal cautivo
wifiManager.autoConnect("ESP8266wifi"); //nombre de la red que levanta el micro
Serial.println("Ya estas conectado");
}
```

Librería funcionreturnhota_fecha.h

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>

#define FIREBASE_HOST  "iotapp-a1c0a-default-rtdb.firebaseio.com"  ///copiar de la base de datos
#define FIREBASE_AUTH  "fWN8AmkKEeEstzVpn5mXYxOOwPlRo2O5SFoH5i9K"

String unsolodigitofecha( String fecha);
String unsolodigitohora( String hora);
```

```
String fechaIniA();
String fechaFinA();

String fechaIniB();
String fechaFinB();

String fechaIniC();
String fechaFinC();

String fechaIniD();
String fechaFinD();


String Alarma1A();
String Alarma2A();
String Alarma3A();

String Alarma1B();
String Alarma2B();
String Alarma3B();

String Alarma1B();
String Alarma2B();
String Alarma3B();

String Alarma1B();
String Alarma2B();
String Alarma3B();
```

```
////////FUNCIONES
////cambio fecha
String unsolodigitofecha( String fecha){
 if(fecha=="01"){
 fecha="1";

 }
 else if(fecha == "02"){
 fecha="2";

 }
 else if(fecha == "03"){
 fecha="3";

 }
 else if(fecha == "04"){
 fecha="4";

 }
 else if(fecha == "05"){
 fecha="5";

 }
 else if(fecha == "06"){
 fecha="6";

 }
 else if(fecha == "07"){
 fecha="7";
```

```
  //Serial.println(fecha);

  }
  else if(fecha == "08"){
  fecha="8";

  }
  else if(fecha == "09"){
  fecha="9";

  }
  else{
    fecha=fecha;
  }

  return fecha;
  }

///cambio hora
String unsolodigitohora( String hora){
  if(hora == "00"){
  hora="0";

  }
  else if(hora == "01"){
  hora="1";

  }
  else if(hora == "02"){
  hora="2";
```

```
}
else if(hora == "03"){
hora="3";

}
else if(hora == "04"){
hora="4";

}
else if(hora == "05"){
hora="5";

}
else if(hora == "06"){
hora="6";

}
else if(hora == "07"){
hora="7";
}
else if(hora == "08"){
hora="8";

}
else if(hora == "09"){
hora="9";

}
else{
```

```
   hora=hora;
  }

  return hora;
  }

//********** Fechas A
String fechaIniA(){
  ///fecha inicio
String fechainiA= Firebase.getString("caja/A/Finicio");
//Serial.println(fechainiA);
//Serial.println(fechainiA.length());
String diainiA= fechainiA.substring(8,10);
String mesiniA= fechainiA.substring(5,7);
String anioiniA= fechainiA.substring(0,4);
///funcion un solo digito fecha o mes
String diainiA_dig= unsolodigitofecha(diainiA);
String mesiniA_dig=unsolodigitofecha(mesiniA);
String fechainiA_final= anioiniA+ "-" + mesiniA_dig + "-" +diainiA_dig;
//Serial.println(fechainiA_final);
return fechainiA_final;
  }
String fechaFinA(){
String fechafinA= Firebase.getString("caja/A/Ffinal");
//Serial.println(fechafinA);
//Serial.println(fechafinA.length());
String diafinA= fechafinA.substring(8,10);
String mesfinA= fechafinA.substring(5,7);
String aniofinA= fechafinA.substring(0,4);
///funcion un solo digito fecha o mes
```

```
String diafinA_dig= unsolodigitofecha(diafinA);
String mesfinA_dig=unsolodigitofecha(mesfinA);
String fechafinA_final= aniofinA+ "-" + mesfinA_dig + "-" +diafinA_dig;

return fechafinA_final;
}

//********** Fechas B
String fechaIniB(){
String fechainiB= Firebase.getString("caja/B/Finicio");
//Serial.println(fechainiB);
//Serial.println(fechainiB.length());
String diainiB= fechainiB.substring(8,10);
String mesiniB= fechainiB.substring(5,7);
String anioiniB= fechainiB.substring(0,4);
///funcion un solo digito fecha o mes
String diainiB_dig= unsolodigitofecha(diainiB);
String mesiniB_dig=unsolodigitofecha(mesiniB);
String fechainiB_final= anioiniB + "-" + mesiniB_dig + "-" +diainiB_dig;
return fechainiB_final;
 }

String fechaFinB(){
String fechafinB= Firebase.getString("caja/B/Ffinal");
//Serial.println(fechafinB);
//Serial.println(fechafinA.length());
String diafinB= fechafinB.substring(8,10);
String mesfinB= fechafinB.substring(5,7);
String aniofinB= fechafinB.substring(0,4);
```

```
///funcion un solo digito fecha o mes
String diafinB_dig= unsolodigitofecha(diafinB);
String mesfinB_dig=unsolodigitofecha(mesfinB);
String fechafinB_final= aniofinB+ "-" + mesfinB_dig + "-" +diafinB_dig;
return fechafinB_final;
 }


//********** Fechas C
String fechaIniC(){
String fechainiC= Firebase.getString("caja/C/Finicio");
//Serial.println(fechainiC);
//Serial.println(fechainiA.length());
String diainiC= fechainiC.substring(8,10);
String mesiniC= fechainiC.substring(5,7);
String anioiniC= fechainiC.substring(0,4);
///funcion un solo digito fecha o mes
String diainiC_dig= unsolodigitofecha(diainiC);
String mesiniC_dig=unsolodigitofecha(mesiniC);
String fechainiC_final= anioiniC+ "-" + mesiniC_dig + "-" + diainiC_dig;
return fechainiC_final;
 }
String fechaFinC(){
String fechafinC= Firebase.getString("caja/C/Ffinal");
//Serial.println(fechafinC);
//Serial.println(fechafinC.length());
String diafinC= fechafinC.substring(8,10);
String mesfinC= fechafinC.substring(5,7);
String aniofinC= fechafinC.substring(0,4);
///funcion un solo digito fecha o mes
```

```
String diafinC_dig= unsolodigitofecha(diafinC);
String mesfinC_dig=unsolodigitofecha(mesfinC);
String fechafinC_final= aniofinC+ "-" + mesfinC_dig + "-" +diafinC_dig;
return fechafinC_final;
 }

//********** Fechas D
String fechaIniD(){
String fechainiD= Firebase.getString("caja/D/Finicio");
//Serial.println(fechainiD);
//Serial.println(fechainiD.length());
String diainiD= fechainiD.substring(8,10);
String mesiniD= fechainiD.substring(5,7);
String anioiniD= fechainiD.substring(0,4);
///funcion un solo digito fecha o mes
String diainiD_dig= unsolodigitofecha(diainiD);
String mesiniD_dig=unsolodigitofecha(mesiniD);
String fechainiD_final= anioiniD+ "-" + mesiniD_dig + "-" +diainiD_dig;
return fechainiD_final;
 }
String fechaFinD(){
String fechafinD= Firebase.getString("caja/D/Ffinal");
//Serial.println(fechafinD);
//Serial.println(fechafinA.length());
String diafinD= fechafinD.substring(8,10);
String mesfinD= fechafinD.substring(5,7);
String aniofinD= fechafinD.substring(0,4);
///funcion un solo digito fecha o mes
String diafinD_dig= unsolodigitofecha(diafinD);
String mesfinD_dig=unsolodigitofecha(mesfinD);
```

```
String fechafinD_final= aniofinD+ "-" + mesfinD_dig + "-" +diafinD_dig;
//Serial.println(fechafinD_final);
return fechafinD_final;
 }

//////////////HORAS

//////////**********ALARMAS A
String Alarma1A(){
String Alarma1A= Firebase.getString("caja/A/Alarma1");
//Serial.println(Alarma1A);
//Serial.println(Alarma1A.length());
String horaAlarma1A= Alarma1A.substring(0,2);
String minutoAlarma1A= Alarma1A.substring(3,5);

String horaAlarma1A_dig= unsolodigitohora(horaAlarma1A);
String minutoAlarma1A_dig=unsolodigitohora(minutoAlarma1A);
String horaAlarma1A_final= horaAlarma1A_dig + ":" + minutoAlarma1A_dig;
//Serial.println(horaAlarma1A_final);
return horaAlarma1A_final;
 }

String Alarma2A(){
String Alarma2A= Firebase.getString("caja/A/Alarma2");
//Serial.println(Alarma2A);
//Serial.println(Alarma2A.length());
String horaAlarma2A= Alarma2A.substring(0,2);
String minutoAlarma2A= Alarma2A.substring(3,5);

String horaAlarma2A_dig= unsolodigitohora(horaAlarma2A);
```

```
String minutoAlarma2A_dig=unsolodigitohora(minutoAlarma2A);
String horaAlarma2A_final= horaAlarma2A_dig + ":" + minutoAlarma2A_dig;
//Serial.println(horaAlarma2A_final);
return horaAlarma2A_final;
 }

 String Alarma3A(){
String Alarma3A= Firebase.getString("caja/A/Alarma3");
//Serial.println(Alarma3A);
//Serial.println(Alarma3A.length());
String horaAlarma3A= Alarma3A.substring(0,2);
String minutoAlarma3A= Alarma3A.substring(3,5);
;
String horaAlarma3A_dig= unsolodigitohora(horaAlarma3A);
String minutoAlarma3A_dig=unsolodigitohora(minutoAlarma3A);
String horaAlarma3A_final= horaAlarma3A_dig + ":" + minutoAlarma3A_dig;
//Serial.println(horaAlarma3A_final);
return horaAlarma3A_final;
 }

String Alarma1B(){
String Alarma1B= Firebase.getString("caja/B/Alarma1");
//Serial.println(Alarma1B);
//Serial.println(Alarma1B.length());
String horaAlarma1B= Alarma1B.substring(0,2);
String minutoAlarma1B= Alarma1B.substring(3,5);

String horaAlarma1B_dig= unsolodigitohora(horaAlarma1B);
String minutoAlarma1B_dig=unsolodigitohora(minutoAlarma1B);
String horaAlarma1B_final= horaAlarma1B_dig + ":" + minutoAlarma1B_dig;
```

```
//Serial.println(horaAlarma1B_final);
return horaAlarma1B_final;
 }

String Alarma2B(){

String Alarma2B= Firebase.getString("caja/B/Alarma2");
//Serial.println(Alarma2B);
//Serial.println(Alarma2B.length());
String horaAlarma2B= Alarma2B.substring(0,2);
String minutoAlarma2B= Alarma2B.substring(3,5);
String horaAlarma2B_dig= unsolodigitohora(horaAlarma2B);
String minutoAlarma2B_dig=unsolodigitohora(minutoAlarma2B);

String horaAlarma2B_final= horaAlarma2B_dig + ":" + minutoAlarma2B_dig;
//Serial.println(horaAlarma2B_final);
return horaAlarma2B_final;
 }

String Alarma3B(){
String Alarma3B= Firebase.getString("caja/B/Alarma3");
//Serial.println(Alarma3B);
//Serial.println(Alarma3B.length());
String horaAlarma3B= Alarma3B.substring(0,2);
String minutoAlarma3B= Alarma3B.substring(3,5);

String horaAlarma3B_dig= unsolodigitohora(horaAlarma3B);
String minutoAlarma3B_dig=unsolodigitohora(minutoAlarma3B);
String horaAlarma3B_final= horaAlarma3B_dig + ":" + minutoAlarma3B_dig;
//Serial.println(horaAlarma3B_final);
```

```
return horaAlarma3B_final;
 }

String Alarma1C(){
String Alarma1C= Firebase.getString("caja/C/Alarma1");
//Serial.println(Alarma1C);
//Serial.println(Alarma1C.length());
String horaAlarma1C= Alarma1C.substring(0,2);
String minutoAlarma1C= Alarma1C.substring(3,5);

String horaAlarma1C_dig= unsolodigitohora(horaAlarma1C);
String minutoAlarma1C_dig=unsolodigitohora(minutoAlarma1C);
String horaAlarma1C_final= horaAlarma1C_dig + ":" + minutoAlarma1C_dig;
//Serial.println(horaAlarma1C_final);
return horaAlarma1C_final;
 }

String Alarma2C(){
String Alarma2C= Firebase.getString("caja/C/Alarma2");
//Serial.println(Alarma2C);
//Serial.println(Alarma2C.length());
String horaAlarma2C= Alarma2C.substring(0,2);
String minutoAlarma2C= Alarma2C.substring(3,5);

String horaAlarma2C_dig= unsolodigitohora(horaAlarma2C);
String minutoAlarma2C_dig=unsolodigitohora(minutoAlarma2C);
String horaAlarma2C_final= horaAlarma2C_dig + ":" + minutoAlarma2C_dig;
//Serial.println(horaAlarma2C_final);
return horaAlarma2C_final;
 }
```

```
String Alarma3C(){
String Alarma3C= Firebase.getString("caja/C/Alarma3");
//Serial.println(Alarma3C);
//Serial.println(Alarma3C.length());
String horaAlarma3C= Alarma3C.substring(0,2);
String minutoAlarma3C= Alarma3C.substring(3,5);

String horaAlarma3C_dig= unsolodigitohora(horaAlarma3C);
String minutoAlarma3C_dig=unsolodigitohora(minutoAlarma3C);
String horaAlarma3C_final= horaAlarma3C_dig + ":" + minutoAlarma3C_dig;
//Serial.println(horaAlarma3C_final);
return horaAlarma3C_final;
 }

String Alarma1D(){
String Alarma1D= Firebase.getString("caja/D/Alarma1");
//Serial.println(Alarma1D);
//Serial.println(Alarma1D.length());
String horaAlarma1D= Alarma1D.substring(0,2);
String minutoAlarma1D= Alarma1D.substring(3,5);

String horaAlarma1D_dig= unsolodigitohora(horaAlarma1D);
String minutoAlarma1D_dig=unsolodigitohora(minutoAlarma1D);
String horaAlarma1D_final= horaAlarma1D_dig + ":" + minutoAlarma1D_dig;
//Serial.println(horaAlarma1D_final);
return horaAlarma1D_final;
 }

String Alarma2D(){
```

```
String Alarma2D= Firebase.getString("caja/D/Alarma2");
//Serial.println(Alarma2D);
//Serial.println(Alarma2A.length());
String horaAlarma2D= Alarma2D.substring(0,2);
String minutoAlarma2D= Alarma2D.substring(3,5);

String horaAlarma2D_dig= unsolodigitohora(horaAlarma2D);
String minutoAlarma2D_dig=unsolodigitohora(minutoAlarma2D);
String horaAlarma2D_final= horaAlarma2D_dig + ":" + minutoAlarma2D_dig;
//Serial.println(horaAlarma2D_final);
return horaAlarma2D_final;
 }

 String Alarma3D(){
String Alarma3D= Firebase.getString("caja/D/Alarma3");
//Serial.println(Alarma3D);
//Serial.println(Alarma3D.length());
String horaAlarma3D= Alarma3D.substring(0,2);
String minutoAlarma3D= Alarma3D.substring(3,5);

String horaAlarma3D_dig= unsolodigitohora(horaAlarma3D);
String minutoAlarma3D_dig=unsolodigitohora(minutoAlarma3D);
String horaAlarma3D_final= horaAlarma3D_dig + ":" + minutoAlarma3D_dig;
//Serial.println(horaAlarma3D_final);
return horaAlarma3D_final;
 }
////////////////////TODO CON PRINT
void getfirebasedat(){

//*******************compartimiento A
```

```
Serial.println("*****compartimiento A*****");
///fecha inicio
String fechainiA= Firebase.getString("caja/A/Finicio");
Serial.println(fechainiA);
//Serial.println(fechainiA.length());
String diainiA= fechainiA.substring(8,10);
String mesiniA= fechainiA.substring(5,7);
String anioiniA= fechainiA.substring(0,4);
///funcion un solo digito fecha o mes
String diainiA_dig= unsolodigitofecha(diainiA);
String mesiniA_dig=unsolodigitofecha(mesiniA);
String fechainiA_final= anioiniA+ "-" + mesiniA_dig + "-" +diainiA_dig;
Serial.println(fechainiA_final);
//Serial.println(diainiA_dig);
//Serial.println(mesiniA_dig);
//Serial.println(anioiniA);


////////////////**********ALARMA**********////////////////////

Serial.println("**********Alarma A**********");
String Alarma1A= Firebase.getString("caja/A/Alarma1");
Serial.println(Alarma1A);
//Serial.println(Alarma1A.length());
String horaAlarma1A= Alarma1A.substring(0,2);
String minutoAlarma1A= Alarma1A.substring(3,5);
Serial.println(horaAlarma1A);
Serial.println(minutoAlarma1A);
String horaAlarma1A_dig= unsolodigitohora(horaAlarma1A);
String minutoAlarma1A_dig=unsolodigitohora(minutoAlarma1A);
```

```
String horaAlarma1A_final= horaAlarma1A_dig + ":" + minutoAlarma1A_dig;
Serial.println(horaAlarma1A_final);
String Alarma2A= Firebase.getString("caja/A/Alarma2");
Serial.println(Alarma2A);
//Serial.println(Alarma2A.length());
String horaAlarma2A= Alarma2A.substring(0,2);
String minutoAlarma2A= Alarma2A.substring(3,5);
Serial.println(horaAlarma2A);
Serial.println(minutoAlarma2A);
String horaAlarma2A_dig= unsolodigitohora(horaAlarma2A);
String minutoAlarma2A_dig=unsolodigitohora(minutoAlarma2A);
String horaAlarma2A_final= horaAlarma2A_dig + ":" + minutoAlarma2A_dig;
Serial.println(horaAlarma2A_final);



String Alarma3A= Firebase.getString("caja/A/Alarma3");
Serial.println(Alarma3A);
//Serial.println(Alarma3A.length());
String horaAlarma3A= Alarma3A.substring(0,2);
String minutoAlarma3A= Alarma3A.substring(3,5);
Serial.println(horaAlarma3A);
Serial.println(minutoAlarma3A);
String horaAlarma3A_dig= unsolodigitohora(horaAlarma3A);
String minutoAlarma3A_dig=unsolodigitohora(minutoAlarma3A);
String horaAlarma3A_final= horaAlarma3A_dig + ":" + minutoAlarma3A_dig;
Serial.println(horaAlarma3A_final);



Serial.println("**********Alarma B**********");
```

```
String Alarma1B= Firebase.getString("caja/B/Alarma1");
Serial.println(Alarma1B);
//Serial.println(Alarma1B.length());
String horaAlarma1B= Alarma1B.substring(0,2);
String minutoAlarma1B= Alarma1B.substring(3,5);
Serial.println(horaAlarma1B);
Serial.println(minutoAlarma1B);
String horaAlarma1B_dig= unsolodigitohora(horaAlarma1B);
String minutoAlarma1B_dig=unsolodigitohora(minutoAlarma1B);
String horaAlarma1B_final= horaAlarma1B_dig + ":" + minutoAlarma1B_dig;
Serial.println(horaAlarma1B_final);

String Alarma2B= Firebase.getString("caja/B/Alarma2");
Serial.println(Alarma2B);
//Serial.println(Alarma2B.length());
String horaAlarma2B= Alarma2B.substring(0,2);
String minutoAlarma2B= Alarma2B.substring(3,5);
Serial.println(horaAlarma2B);
Serial.println(minutoAlarma2B);
String horaAlarma2B_dig= unsolodigitohora(horaAlarma2B);
String minutoAlarma2B_dig=unsolodigitohora(minutoAlarma2B);
String horaAlarma2B_final= horaAlarma2B_dig + ":" + minutoAlarma2B_dig;
Serial.println(horaAlarma2B_final);

String Alarma3B= Firebase.getString("caja/B/Alarma3");
Serial.println(Alarma3B);
//Serial.println(Alarma3B.length());
String horaAlarma3B= Alarma3B.substring(0,2);
String minutoAlarma3B= Alarma3B.substring(3,5);
Serial.println(horaAlarma3B);
```

```
Serial.println(minutoAlarma3B);
String horaAlarma3B_dig= unsolodigitohora(horaAlarma3B);
String minutoAlarma3B_dig=unsolodigitohora(minutoAlarma3B);
String horaAlarma3B_final= horaAlarma3B_dig + ":" + minutoAlarma3B_dig;
Serial.println(horaAlarma3B_final);


Serial.println("*****************Alarma C***************");
String Alarma1C= Firebase.getString("caja/C/Alarma1");
Serial.println(Alarma1C);
//Serial.println(Alarma1C.length());
String horaAlarma1C= Alarma1C.substring(0,2);
String minutoAlarma1C= Alarma1C.substring(3,5);
Serial.println(horaAlarma1C);
Serial.println(minutoAlarma1C);
String horaAlarma1C_dig= unsolodigitohora(horaAlarma1C);
String minutoAlarma1C_dig=unsolodigitohora(minutoAlarma1C);
String horaAlarma1C_final= horaAlarma1C_dig + ":" + minutoAlarma1C_dig;
Serial.println(horaAlarma1C_final);

String Alarma2C= Firebase.getString("caja/C/Alarma2");
Serial.println(Alarma2C);
//Serial.println(Alarma2C.length());
String horaAlarma2C= Alarma2C.substring(0,2);
String minutoAlarma2C= Alarma2C.substring(3,5);
Serial.println(horaAlarma2C);
Serial.println(minutoAlarma2C);
String horaAlarma2C_dig= unsolodigitohora(horaAlarma2C);
String minutoAlarma2C_dig=unsolodigitohora(minutoAlarma2C);
String horaAlarma2C_final= horaAlarma2C_dig + ":" + minutoAlarma2C_dig;
```

```
Serial.println(horaAlarma2C_final);

String Alarma3C= Firebase.getString("caja/C/Alarma3");
Serial.println(Alarma3C);
//Serial.println(Alarma3C.length());
String horaAlarma3C= Alarma3C.substring(0,2);
String minutoAlarma3C= Alarma3C.substring(3,5);
Serial.println(horaAlarma3C);
Serial.println(minutoAlarma3C);
String horaAlarma3C_dig= unsolodigitohora(horaAlarma3C);
String minutoAlarma3C_dig=unsolodigitohora(minutoAlarma3C);
String horaAlarma3C_final= horaAlarma3C_dig + ":" + minutoAlarma3C_dig;
Serial.println(horaAlarma3C_final);

Serial.println("**********Alarma D**********");
String Alarma1D= Firebase.getString("caja/D/Alarma1");
Serial.println(Alarma1D);
//Serial.println(Alarma1D.length());
String horaAlarma1D= Alarma1D.substring(0,2);
String minutoAlarma1D= Alarma1D.substring(3,5);
Serial.println(horaAlarma1D);
Serial.println(minutoAlarma1D);
String horaAlarma1D_dig= unsolodigitohora(horaAlarma1D);
String minutoAlarma1D_dig=unsolodigitohora(minutoAlarma1D);
String horaAlarma1D_final= horaAlarma1D_dig + ":" + minutoAlarma1D_dig;
Serial.println(horaAlarma1D_final);


String Alarma2D= Firebase.getString("caja/D/Alarma2");
Serial.println(Alarma2D);
```

```
//Serial.println(Alarma2A.length());
String horaAlarma2D= Alarma2D.substring(0,2);
String minutoAlarma2D= Alarma2D.substring(3,5);
Serial.println(horaAlarma2D);
Serial.println(minutoAlarma2D);
String horaAlarma2D_dig= unsolodigitohora(horaAlarma2D);
String minutoAlarma2D_dig=unsolodigitohora(minutoAlarma2D);
String horaAlarma2D_final= horaAlarma2D_dig + ":" + minutoAlarma2D_dig;
Serial.println(horaAlarma2D_final);

String Alarma3D= Firebase.getString("caja/D/Alarma3");
Serial.println(Alarma3D);
//Serial.println(Alarma3D.length());
String horaAlarma3D= Alarma3D.substring(0,2);
String minutoAlarma3D= Alarma3D.substring(3,5);
Serial.println(horaAlarma3D);
Serial.println(minutoAlarma3D);
String horaAlarma3D_dig= unsolodigitohora(horaAlarma3D);
String minutoAlarma3D_dig=unsolodigitohora(minutoAlarma3D);
String horaAlarma3D_final= horaAlarma3D_dig + ":" + minutoAlarma3D_dig;
Serial.println(horaAlarma3D_final);

}
```

Librería Fechahora_actual.h

```
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
```

```
// Set offset time in seconds to adjust for your timezone, for example:
  // GMT +1 = 3600
  // GMT +8 = 28800
  // GMT -1 = -3600
  // GMT 0 = 0
const long utcOffsetInSeconds = 3600;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "europe.pool.ntp.org", utcOffsetInSeconds);

String horaActual();
String fechaActual();

String horaActual(){
timeClient.update();
  ///sumo 1 por el horario de verano
  int currentHour = timeClient.getHours()+1;
//Serial.print("Hour: ");
//Serial.println(currentHour);

int currentMinute = timeClient.getMinutes();
//Serial.print("Minutes: ");
//Serial.println(currentMinute);

String hora_min= String(currentHour) + ":" + String(currentMinute);
//Serial.print("hora_min:");
//Serial.println(hora_min);
return hora_min;
  }
```

```
String fechaActual(){
  timeClient.update();
    time_t epochTime = timeClient.getEpochTime();
struct tm *ptm = gmtime ((time_t *)&epochTime);

  int monthDay = ptm->tm_mday;
  //Serial.print("Month day: ");
 // Serial.println(monthDay);

  int currentMonth = ptm->tm_mon+1;
 // Serial.print("Month: ");
  //Serial.println(currentMonth);

 int currentYear = ptm->tm_year+1900;
 // Serial.print("Year: ");
 // Serial.println(currentYear);

  //Print complete date:
  String FechaActual = String(currentYear) + "-" + String(currentMonth) + "-" + String(monthDay);
  //Serial.print("Fecha: ");
 //Serial.println(FechaActual);
 return FechaActual;
   }
```

Programa Alarmas

```
#include "lib_red.h"

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
```

```
#include <ArduinoJson.h>

#include "fechahora_actual.h"
#include "funcionreturnhora_fecha.h"

//credenciales firebase

#define FIREBASE_HOST  "iotapp-a1c0a-default-rtdb.firebaseio.com"  ///copiar de la base de datos
#define FIREBASE_AUTH  "fWN8AmkKEeEstzVpn5mXYxOOwPlRo2O5SFoH5i9K"

int ledA= 16; //D0
int ledB= 5;  //D1
int ledC= 4;  //D2
int ledD= 2;   //D4
int buzzer= 14;  //D5
int interruptor=12; //D6
//int ledPincompartment = 16;//D0

unsigned long tf=0;
void setup() {
 Serial.begin(9600);

 conectaWifi();

 pinMode(ledA,OUTPUT);
 pinMode(ledB,OUTPUT);
 pinMode(ledC,OUTPUT);
 pinMode(ledD,OUTPUT);
 pinMode(buzzer,OUTPUT);
```

```
///inicializacion firebase
Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);

// Initialize a NTPClient to get time
  timeClient.begin();

}

void loop() {

unsigned long ti=millis();
if(ti>tf){
  tf=ti+10000;

  if(Firebase.success()){
  Serial.println("conexion exitosa");
  }
  else{
    Serial.println("conexion fallida");
    }


//////////////////tiempo Actual//
String hora_actual=horaActual();
Serial.print("Hora Actual: ");
Serial.println(hora_actual);

///////////////ALARMAS

Serial.println("**********ALARMAS A**********");
```

```
//Serial.print("Alarma 1A");
String alarma1A=Alarma1A();
Serial.println(alarma1A);

//Serial.print("Alarma 2A");
String alarma2A=Alarma2A();
Serial.println(alarma2A);

//Serial.print("Alarma 3 A");
String alarma3A=Alarma3A();
Serial.println(alarma3A);

enciendeledA(hora_actual,alarma1A,alarma2A,alarma3A);

Serial.println("**********ALARMAS B**********");
String alarma1B=Alarma1B();
Serial.println(alarma1B);

String alarma2B=Alarma2B();
Serial.println(alarma2B);

String alarma3B=Alarma3B();
Serial.println(alarma3B);

enciendeledB(hora_actual,alarma1B,alarma2B,alarma3B);

Serial.println("**********ALARMAS C**********");
String alarma1C=Alarma1C();
Serial.println(alarma1C);
```

```
String alarma2C=Alarma2C();
Serial.println(alarma2C);

String alarma3C=Alarma3C();
Serial.println(alarma3C);

enciendeledC(hora_actual,alarma1C,alarma2C,alarma3C);



Serial.println("**********ALARMAS D**********");
String alarma1D=Alarma1D();
Serial.println(alarma1D);

String alarma2D=Alarma2D();
Serial.println(alarma2D);

String alarma3D=Alarma3D();
Serial.println(alarma3D);

enciendeledD(hora_actual,alarma1D,alarma2D,alarma3D);

}
}

///LED A
void enciendeledA(String horActual,String alarma1, String alarma2, String alarma3){

  if(horActual == alarma1){
    Serial.println("Alarma Encendida A1");
    digitalWrite(ledA,HIGH);
```

```
    digitalWrite(buzzer,HIGH);

    }
else if(horActual == alarma2){
    Serial.println("Alarma Encendida A2");
    digitalWrite(ledA,HIGH);

    digitalWrite(buzzer,HIGH);

    }

else if(horActual == alarma3){
   Serial.println("Alarma Encendida A3");
   digitalWrite(ledA,HIGH);

   digitalWrite(buzzer,HIGH);
   //delay(500);
   //digitalWrite(buzzer,LOW);
   }
   else{
  digitalWrite(ledA,LOW);
  digitalWrite(buzzer,LOW);
  }
  }

/////LED B
void enciendeledB(String horActual,String alarma1, String alarma2, String alarma3){

  if(horActual == alarma1){
```

```
    Serial.println("Alarma Encendida A1");
    digitalWrite(ledB,HIGH);

    digitalWrite(buzzer,HIGH);

    }
else if(horActual == alarma2){
    Serial.println("Alarma Encendida A2");
    digitalWrite(ledB,HIGH);

    digitalWrite(buzzer,HIGH);

    }

else if(horActual == alarma3){
   Serial.println("Alarma Encendida A3");
   digitalWrite(ledB,HIGH);

   digitalWrite(buzzer,HIGH);

   }
   else{
  digitalWrite(ledB,LOW);
  digitalWrite(buzzer,LOW);
  }
  }

////LED C
 void enciendeledC(String horActual,String alarma1, String alarma2, String alarma3){
```

```
  if(horActual == alarma1){
    Serial.println("Alarma Encendida A1");
    digitalWrite(ledC,HIGH);

    digitalWrite(buzzer,HIGH);

    }
else if(horActual == alarma2){
    Serial.println("Alarma Encendida A2");
    digitalWrite(ledC,HIGH);

    digitalWrite(buzzer,HIGH);

    }

else if(horActual == alarma3){
   Serial.println("Alarma Encendida A3");
   digitalWrite(ledC,HIGH);

   digitalWrite(buzzer,HIGH);

   }
   else{
  digitalWrite(ledC,LOW);
  digitalWrite(buzzer,LOW);
  }
  }

//D
 void enciendeledD(String horActual,String alarma1, String alarma2, String alarma3){
```

```
 if(horActual == alarma1){
   Serial.println("Alarma Encendida A1");
   digitalWrite(ledD,HIGH);

   digitalWrite(buzzer,HIGH);

   }
else if(horActual == alarma2){
   Serial.println("Alarma Encendida A2");
   digitalWrite(ledD,HIGH);

   digitalWrite(buzzer,HIGH);

   }

else if(horActual == alarma3){
   Serial.println("Alarma Encendida A3");
   digitalWrite(ledD,HIGH);

   digitalWrite(buzzer,HIGH);


   }
   else{
  digitalWrite(ledD,LOW);
  digitalWrite(buzzer,LOW);
  }
  }
```

Programa Sensores

```
#include "lib_red.h"

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>



#include "fechahora_actual.h"
#include "funcionreturnhora_fecha.h"

#include <Wire.h>
#include <Adafruit_ADS1X15.h>

//credenciales firebase

#define FIREBASE_HOST  "iotapp-a1c0a-default-rtdb.firebaseio.com"  ///copiar de la base de datos
#define FIREBASE_AUTH  "fWN8AmkKEeEstzVpn5mXYxOOwPlRo2O5SFoH5i9K"

int ledPincompartment = 14;//D0
Adafruit_ADS1015 ads;

unsigned long tf=0;
void setup() {
 Serial.begin(9600);

 conectaWifi();

 pinMode(ledPincompartment,OUTPUT);
```

```
///inicializacion firebase
Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);

// Initialize a NTPClient to get time
 timeClient.begin();

 ads.setGain(GAIN_ONE);       // 1x gain   +/- 4.096V  1 bit = 2mV      0.125mV


 if (!ads.begin()) {
  Serial.println("Failed to initialize ADS.");
  while (1);
 }


}

void loop() {

unsigned long ti=millis();
if(ti>tf){
 tf=ti+10000;

 if(Firebase.success()){
 Serial.println("conexion exitosa");
 }
 else{
  Serial.println("conexion fallida");
```

```
  }


//////////////////tiempo Actual//
String hora_actual=horaActual();
Serial.print("Hora Actual: ");
Serial.println(hora_actual);



///////////////ALARMAS

Serial.println("**********ALARMAS A**********");
//Serial.print("Alarma 1A");
String alarma1A=Alarma1A();
Serial.println(alarma1A);

//Serial.print("Alarma 2A");
String alarma2A=Alarma2A();
Serial.println(alarma2A);

//Serial.print("Alarma 3 A");
String alarma3A=Alarma3A();
Serial.println(alarma3A);

alarmaFlexA(hora_actual,alarma1A,alarma2A,alarma3A);
noAlarmaA(alarma1A,alarma2A,alarma3A);

Serial.println("**********ALARMAS B**********");
String alarma1B=Alarma1B();
```

```
Serial.println(alarma1B);

String alarma2B=Alarma2B();
Serial.println(alarma2B);

String alarma3B=Alarma3B();
Serial.println(alarma3B);

alarmaFlexB(hora_actual,alarma1B,alarma2B,alarma3B);
noAlarmaB(alarma1B,alarma2B,alarma3B);



Serial.println("**********ALARMAS C**********");
String alarma1C=Alarma1C();
Serial.println(alarma1C);

String alarma2C=Alarma2C();
Serial.println(alarma2C);
String alarma3C=Alarma3C();
Serial.println(alarma3C);

alarmaFlexC(hora_actual,alarma1C,alarma2C,alarma3C);
noAlarmaC(alarma1C,alarma2C,alarma3C);



Serial.println("**********ALARMAS D**********");
String alarma1D=Alarma1D();
Serial.println(alarma1D);

String alarma2D=Alarma2D();
```

```
Serial.println(alarma2D);

String alarma3D=Alarma3D();
Serial.println(alarma3D);

alarmaFlexD(hora_actual,alarma1D,alarma2D,alarma3D);
noAlarmaD(alarma1D,alarma2D,alarma3D);

}
}

//////////////////////////funciones
///LED A
void alarmaFlexA(String horActual,String alarma1, String alarma2, String alarma3){

int16_t flexA;
flexA = ads.readADC_SingleEnded(0);
Serial.print(flexA);
int flexcomparcionA=120;
  if(horActual == alarma1 && flexA>flexcomparcionA){
    Serial.println("hora a1 , flex activado");
    Firebase.setString("caja/A/flexA_alar1", "ALARMA ATENDIDA");
    digitalWrite(ledPincompartment,HIGH);
     }
    else{
     Firebase.setString("caja/A/flexA_alar1", "ALARMA NO ATENDIDA");
     digitalWrite(ledPincompartment,LOW);
     }

    if(horActual == alarma2 && flexA>flexcomparcionA){
```

```
   Serial.println("hora a2 , flex activado");
   Firebase.setString("caja/A/flexA_alar2", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/A/flexA_alar2", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }

    if(horActual == alarma3 && flexA>flexcomparcionA){
   Serial.println("hora a3 , flex activado");
   Firebase.setString("caja/A/flexA_alar3", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/A/flexA_alar3", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }

 }

///LED B
void alarmaFlexB(String horActual,String alarma1, String alarma2, String alarma3){

int16_t flexB;
flexB = ads.readADC_SingleEnded(1);
Serial.print(flexB);
int flexcomparcionB=175;
//alarma1
  if(horActual == alarma1 && flexB>flexcomparcionB){
```

```
  Serial.println("hora B1 , flex activado");
  Firebase.setString("caja/B/flexB_alar1", "ALARMA ATENDIDA");
  digitalWrite(ledPincompartment,HIGH);
   }
  else{
   Firebase.setString("caja/B/flexB_alar1", "ALARMA NO ATENDIDA");
   digitalWrite(ledPincompartment,LOW);
   }
   ///alarm2

  if(horActual == alarma2 && flexB>flexcomparcionB){
  Serial.println("hora B2 , flex activado");
  Firebase.setString("caja/B/flexB_alar2", "ALARMA ATENDIDA");
  digitalWrite(ledPincompartment,HIGH);
   }
  else{
   Firebase.setString("caja/B/flexB_alar2", "ALARMA NO ATENDIDA");
   digitalWrite(ledPincompartment,LOW);
   }
///alarm3
    if(horActual == alarma3 && flexB>flexcomparcionB){
  Serial.println("hora B3 , flex activado");
  Firebase.setString("caja/B/flexB_alar3", "ALARMA ATENDIDA");
  digitalWrite(ledPincompartment,HIGH);
   }
  else{
   Firebase.setString("caja/B/flexB_alar3", "ALARMA NO ATENDIDA");
   digitalWrite(ledPincompartment,LOW);
   }
```

```
 }

///LED C
void alarmaFlexC(String horActual,String alarma1, String alarma2, String alarma3){

int16_t flexC;
flexC = ads.readADC_SingleEnded(2);
Serial.print(flexC);
int flexcomparcionC=108;
 if(horActual == alarma1 && flexC>flexcomparcionC){
   Serial.println("hora c1 , flex activado");
   Firebase.setString("caja/C/flexC_alar1", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/C/flexC_alar1", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }

   if(horActual == alarma2 && flexC>flexcomparcionC){
   Serial.println("hora c2 , flex activado");
   Firebase.setString("caja/C/flexC_alar2", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/C/flexC_alar2", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }

    if(horActual == alarma3 && flexC>flexcomparcionC){
```

```
   Serial.println("hora C3 , flex activado");
   Firebase.setString("caja/C/flexC_alar3", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/C/flexC_alar3", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }

 }

///LED D
void alarmaFlexD(String horActual,String alarma1, String alarma2, String alarma3){

int16_t flexD;
flexD = ads.readADC_SingleEnded(3);
Serial.print(flexD);
int flexcomparcionD=100;
//alarma1
  if(horActual == alarma1 && flexD>flexcomparcionD){
   Serial.println("hora al1 , flex activado");
   Firebase.setString("caja/D/flexD_alar1", "ALARMA ATENDIDA");
   digitalWrite(ledPincompartment,HIGH);
    }
   else{
    Firebase.setString("caja/D/flexD_alar1", "ALARMA NO ATENDIDA");
    digitalWrite(ledPincompartment,LOW);
    }
    ///alarm2
```

```
  if(horActual == alarma2 && flexD>flexcomparcionD){
  Serial.println("hora alar2 , flex activado");
  Firebase.setString("caja/D/flexD_alar2", "ALARMA ATENDIDA");
  digitalWrite(ledPincompartment,HIGH);
   }
  else{
   Firebase.setString("caja/D/flexD_alar2", "ALARMA NO ATENDIDA");
   digitalWrite(ledPincompartment,LOW);
   }
 ///alarm3
    if(horActual == alarma3 && flexD>flexcomparcionD){
  Serial.println("hora D3 , flex activado");
  Firebase.setString("caja/D/flexD_alar3", "ALARMA ATENDIDA");
  digitalWrite(ledPincompartment,HIGH);
   }
  else{
   Firebase.setString("caja/D/flexD_alar3", "ALARMA NO ATENDIDA");
   digitalWrite(ledPincompartment,LOW);
   }

 }

////////**********poner en la base de datos la alarma no existe

///A
 void noAlarmaA(String alarma1, String alarma2, String alarma3){

 if( alarma1 == ":"){

  Firebase.setString("caja/A/flexA_alar1", "NO HAY ALARMA");
```

```
  }

if(alarma2== ":"){
   Firebase.setString("caja/A/flexA_alar2", "NO HAY ALARMA");

  }

if(alarma3== ":"){
  Firebase.setString("caja/A/flexA_alar3", "NO HAY ALARMA");
  }

 }
 ///B
 void noAlarmaB(String alarma1, String alarma2, String alarma3){

 if( alarma1 == ":"){

   Firebase.setString("caja/B/flexB_alar1", "NO HAY ALARMA");
  }

 if(alarma2== ":"){
   Firebase.setString("caja/B/flexB_alar2", "NO HAY ALARMA");

  }

if(alarma3== ":"){
  Firebase.setString("caja/B/flexB_alar3", "NO HAY ALARMA");
  }

 }
```

```
 ///C
void noAlarmaC(String alarma1, String alarma2, String alarma3){

 if( alarma1 == ":"){

   Firebase.setString("caja/C/flexC_alar1", "NO HAY ALARMA");
  }

if(alarma2== ":"){
   Firebase.setString("caja/C/flexC_alar2", "NO HAY ALARMA");

   }

if(alarma3== ":"){
  Firebase.setString("caja/C/flexC_alar3", "NO HAY ALARMA");
  }

}

 ///D
void noAlarmaD(String alarma1, String alarma2, String alarma3){

 if( alarma1 == ":"){

   Firebase.setString("caja/D/flexD_alar1", "NO HAY ALARMA");
   }

 if(alarma2== ":"){
   Firebase.setString("caja/D/flexD_alar2", "NO HAY ALARMA");
```

```
  }

if(alarma3== ":"){
  Firebase.setString("caja/D/flexD_alar3", "NO HAY ALARMA");
  }

 }
```

Anexo B. Código JavaScript y HTML

HTML HOME

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" href="style2.css" />
    <link
href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
rel="stylesheet" />
    <link rel="stylesheet" href="path/to/font-awesome/css/font-
awesome.min.css">

    <title>Medbox</title>
  </head>

  <body>

    <div class="container-fluid">
```

```
        <div class="row flex-nowrap">
            <div class="col-auto px-0">
                <div id="sidebar" class="collapse collapse-horizontal show
border-end">
                    <div id="sidebar-nav" class="list-group border-0 rounded-0
text-sm-start min-vh-100">

                        <a href="../home/home.html" class="list-group-item
border-end-0 d-inline-block text-truncate" data-bs-parent="#sidebar"><i
class="bi bi-bootstrap"></i> <span>Home</span> </a>
                        <!--
                        <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
film"></i> <span>Perfil</span></a>
                        <a href="" onclick="return Salir();" class="list-
group-item border-end-0 d-inline-block text-truncate" data-bs-
parent="#sidebar"> <span>Salir</span></a>
                         -->
                        <a href="../pacientes/pacientes.html" class="list-
group-item border-end-0 d-inline-block text-truncate" data-bs-
parent="#sidebar"><i class="bi bi-heart"></i> <span>Pacientes</span></a>
                        <a href="" onclick="return Salir();" class="list-
group-item border-end-0 d-inline-block text-truncate" data-bs-
parent="#sidebar"> <span>Salir</span></a>

                    </div>
                </div>
            </div>
            <main class="col ps-md-2 pt-2">
                <a href="#" data-bs-target="#sidebar" data-bs-
toggle="collapse" class="border rounded-3 p-1 text-decoration-none"><i
class="bi bi-list bi-lg py-2 p-1"></i> Menu</a>
                <div class="page-header pt-3">
                    <h2>Imedbox</h2>
                </div>
                <p class="lead">A offcanvas "push" vertical nav menu
example.</p>
                <hr>
                <div class="row">
                    <div class="col-12">

                    </div>
                </div>
            </main>
        </div>
```

```
    </div>


    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js
" integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>

    <script type="module" src="../home/home.js"></script>
  </body>
</html>
```

HTML índice

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <link rel="stylesheet" href="style.css">


    <title>IotMedbox</title>
</head>
<body>

    <div class="container-form sign-up">
        <div class="welcome-back">

            <div class="message">
                <h2>Bienvenido a Medbox</h2>
                <p>Si ya tienes una cuenta... Inicia Aqui</p>
                <button class="sign-up-btn">Iniciar Sesion</button>
            </div>
        </div>
```

```
    <form class="formulario">

        <h2 class="create-account">Crear una cuenta</h2>

        <label>Nombre</label>
        <input type="text" placeholder="Enter Name"  id="nombreSingUp" />

        <label >Email</label>
        <input type="email" placeholder="Enter Email"  id="emailSingUp"/>

        <label>Password</label>
        <input type="password" placeholder="Password"  id="pasSingUp"/>

        <button  class="registerbtn"  id="signUp" >Registro</button>

    </form>
</div>

<div class="container-form sign-in">

    <form class="formulario">

        <h2 class="create-account">Iniciar Sesion</h2>


        <label >Email</label>
        <input type="text" placeholder="Enter Email" id="emailSingIn"/>

        <label>Password</label>
        <input type="password" placeholder="Password"  id="pasSingIn"/>

        <button id="signIn" class="registerbtn">Iniciar Sesion</button>

    </form>

    <div class="welcome-back">

        <div class="message">

            <h2>Bienvenido de nuevo a Medbox</h2>
            <p>¿Aun no tienes cuenta? Porfavor registrese aqui?</p>
            <button class="sign-in-btn" > registrarse</button>
        </div>
    </div>
```

```
    </div>



</body>
<script  src="movimientoindex.js"></script>
<script type="module" src="app.js"></script>
</html>
```

Js Índice

```
import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.11/firebase-app.js"
import {
  getAuth,
  createUserWithEmailAndPassword,
  signInWithEmailAndPassword,
  signOut,
} from "https://www.gstatic.com/firebasejs/9.6.11/firebase-auth.js";
import {
  getDatabase,
  set,
  ref,
  update,
} from "https://www.gstatic.com/firebasejs/9.6.11/firebase-database.js";


const firebaseConfig = {
  apiKey: "AIzaSyDJFJSBdTruJXjNVnZv1kC9UPZsPN_47zY",
  authDomain: "iotapp-a1c0a.firebaseapp.com",
  databaseURL: "https://iotapp-a1c0a-default-rtdb.firebaseio.com",
  projectId: "iotapp-a1c0a",
  storageBucket: "iotapp-a1c0a.appspot.com",
  messagingSenderId: "592743603808",
  appId: "1:592743603808:web:d98581ec24b58066cbad05"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getDatabase(app);

////Registro
document.getElementById("signUp").addEventListener('click', function(event){
```

```
  event.preventDefault();
  const email=document.getElementById("emailSingUp").value;
  const password=document.getElementById("pasSingUp").value;
  const usuaromed=document.getElementById("nombreSingUp").value;

  createUserWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {
    // Signed in
    const user = userCredential.user;
    alert("usuario creado correctamente");
    console.log(email,password,usuaromed);
    window.location="./dashboardfinal/home/home.html";
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    alert(errorMessage + errorCode );
    // ..
  });

})

//// acceso inicio sesion login
document.getElementById("signIn").addEventListener('click', function(event){

    event.preventDefault();
  const email=document.getElementById("emailSingIn").value;
  const password=document.getElementById("pasSingIn").value;

  signInWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {
    // Signed in
    const user = userCredential.user;
    alert("Bienvenido");
    console.log(email,password);
    window.location="./dashboardfinal/home/home.html";



  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    alert(errorMessage + errorCode );
  });
})
```

```
function Salir(){
  signOut(auth).then(() => {
    alert("adios");
  }).catch((error) => {
    alert(error);
  });
}
```

HTML pacientes

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

    <link
href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
rel="stylesheet" />
    <!--
     <link rel="stylesheet" href="path/to/font-awesome/css/font-
awesome.min.css">
    -->

    <title>Listado Pacientes</title>
  </head>

  <body>

    <div class="container-fluid">
        <div class="row flex-nowrap">
            <div class="col-auto px-0">
                <div id="sidebar" class="collapse collapse-horizontal show
border-end">
```

```
                    <div id="sidebar-nav" class="list-group border-0 rounded-0
text-sm-start min-vh-100">
                        <a href="../home/home.html" class="list-group-item
border-end-0 d-inline-block text-truncate" data-bs-parent="#sidebar"><i
class="bi bi-bootstrap"></i> <span>Home</span> </a>
                        <!--
                        <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
film"></i> <span>Perfil</span></a>
                       <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
bricks"></i> <span>Salir</span></a>
                    -->
                        <a href="../pacientes/pacientes.html" class="list-
group-item border-end-0 d-inline-block text-truncate" data-bs-
parent="#sidebar"><i class="bi bi-heart"></i> <span>Pacientes</span></a>
                        <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
bricks"></i> <span>Salir</span></a>

                    </div>
                </div>
            </div>
            <main class="col ps-md-2 pt-2">
                <a href="#" data-bs-target="#sidebar" data-bs-
toggle="collapse" class="border rounded-3 p-1 text-decoration-none"><i
class="bi bi-list bi-lg py-2 p-1"></i> Menu</a>
                <div class="page-header pt-3">
                    <h2>Pacientes</h2>
                </div>

                <hr>

                <div class="row" >
                    <div class="col-10">
                        <!--  data="--data-toggle" cuando queremos que al
pasar el mouse nos diga que es o para que sirve el boton-->
              <button id="btnNuevo" class="btn btn-primary" data="--data-
toggle"  title="Añadir paciente" data-bs-toggle="modal" data-bs-
target="#modaledicion"> Añadir
                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-plus-circle" viewBox="0 0 16 16">
                    <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0
8 0a8 8 0 0 0 0 16z"/>
```

```
                    <path d="M8 4a.5.5 0 0 1 .5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5
0 0 1-1 0v-3h-3a.5.5 0 0 1 0-1h3v-3A.5.5 0 0 1 8 4z"/>
                  </svg>
              </button>

              <!-- The Modal -->
              <div class="modal fade" id="modaledicion">
                <div class="modal-dialog">
                  <div class="modal-content">
                    <!-- Modal Header -->
                    <div class="modal-header">
                     <h4 class="modal-title">Nuevo Paciente</h4>
                     <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
                    </div>

                    <!-- Modal body -->
                    <div class="modal-body">
                      <form>
                        <div class="form-group">
                          <label for="recipient-name" class="col-form-
label">Ficha</label>
                          <input id="ficha" type="text" class="form-control" >
                        </div>

                        <div class="form-group">
                          <label for="recipient-name" class="col-form-
label">nombre</label>
                          <input id="nombre" type="text" class="form-control"
>
                        </div>

                        <div class="form-group">
                          <label for="recipient-name" class="col-form-
label">Telefono</label>
                          <input id="telf" type="text" class="form-control" >
                        </div>

                        <div class="form-group">
                          <label for="recipient-name" class="col-form-
label">Email</label>
                          <input id="email" type="text" class="form-control" >
                        </div>
                      </form>
                    </div>
```

```
                    <!-- Modal footer -->
                    <div class="modal-footer">
                      <button type="button" class="btn btn-danger" data-bs-
dismiss="modal">Salir</button>
                       <button  id="btnGuardar" class="btn btn-
primary">Guardar</button>
                    </div>
                  </div>
              </div>
            </div>

            <table id="crud" class="table table-bordered">

              <thead>
                <tr class="bg-dark text-light">
                  <th scope="col">Ficha</th>
                  <th scope="col">Name</th>
                  <th scope="col">Phone</th>
                  <th scope="col">Email</th>
                  <th scope="col">Eliminar</th>
                </tr>
              </thead>


              <tbody >

              </tbody>

            </table>


             </div>
             </div>



            </main>
          </div>
      </div>


      <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js
" integrity="sha384-
```

```
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
    <script type="module" src="pacientes.js"></script>


  </body>
</html>
```

Js pacientes

```
import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.9/firebase-app.js";
import { getDatabase, ref,push,onValue,get, set, child, update, remove } from
"https://www.gstatic.com/firebasejs/9.6.9/firebase-database.js";

const firebaseConfig = {
  apiKey: "AIzaSyDJFJSBdTruJXjNVnZv1kC9UPZsPN_47zY",
  authDomain: "iotapp-a1c0a.firebaseapp.com",
  databaseURL: "https://iotapp-a1c0a-default-rtdb.firebaseio.com",
  projectId: "iotapp-a1c0a",
  storageBucket: "iotapp-a1c0a.appspot.com",
  messagingSenderId: "592743603808",
  appId: "1:592743603808:web:d98581ec24b58066cbad05"
};


    //let tableBody=document.querySelector("tbody");
   // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const db=getDatabase(app);



    ///datos
   const userId=document.getElementById("ficha");
   const nombre=document.getElementById("nombre");
   const telf=document.getElementById("telf");
   const email=document.getElementById("email");
```

```
   const btnGuardar=document.getElementById("btnGuardar");

  function limpiar(){
    document.getElementById("nombre").value="";
    document.getElementById("telf").value="";
    document.getElementById("email").value="";
    document.getElementById("ficha").value="";
 }
 //window.limpiar=limpiar;




    /////----------------------------escribir en la base de data---------------
---------

    function InsertData(){
      ///const pruebaId=push(refprueba);
      set(ref(db,"pacientes/"+ userId.value),{
          Nombre: nombre.value,
          Telf: telf.value,
          Email: email.value
      })
      .then(()=>{
          alert("data stored succes");
          limpiar();

      })
      .catch((error)=>{
          alert("unsucces"+ error);

      });
  }

  btnGuardar.addEventListener('click',InsertData);

  ////onvalue
let tableBody=document.querySelector("tbody");


const patientRef = ref(db, 'pacientes/');
onValue(patientRef, (snapshot) => {

    const patients = snapshot.val();
```

```
    tableBody.innerHTML = " ";
    for(var user in patients)
    {

     let tr=`
    <tr>
    <td>${user}</td>
     <td>${patients[user].Nombre}</td>
     <td> ${patients[user].Telf}</td>
     <td> ${patients[user].Email}</td>


     <td>

      <button  onclick="DeleteData(${user})" class="btn btn-
danger">Eliminar</button>

      <a
href="../pacientes/datospacientes.html?idUsuario=${user}"><button  onclick="Ve
r(${user})" class="btn btn-warning">ver</button></a>

     </td>
   </tr>
    `
   tableBody.innerHTML += tr;
    }
 window.user=user;
 });

 //--------eliminar datos
function DeleteData(user){
  remove(ref(db,'pacientes/' + user),{

  })
  .then(()=>{
      alert("data remove succes");

  })
  .catch((error)=>{
      alert("unsucces"+ error);

  });
}
```

```
window.DeleteData=DeleteData;
```

HTML datospacientes

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" href="/assets/css/style.css" />
    <link
href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
rel="stylesheet" />

   <!-- <link rel="stylesheet" href="path/to/font-awesome/css/font-
awesome.min.css">-->


    <title>Datos paciente</title>
  </head>

  <body>

    <div class="container-fluid" >
        <div class="row flex-nowrap">
            <div class="col-auto px-0">
                <div id="sidebar" class="collapse collapse-horizontal show
border-end">
                    <div id="sidebar-nav" class="list-group border-0 rounded-0
text-sm-start min-vh-100">
                        <a href="../home/home.html" class="list-group-item
border-end-0 d-inline-block text-truncate" data-bs-parent="#sidebar"><i
class="bi bi-bootstrap"></i> <span>Home</span> </a>
                        <!--
```

```
                        <a href="" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
film"></i> <span>Perfil</span></a>
                        <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
bricks"></i> <span>Salir</span></a>
                    -->
                        <a href="../pacientes/pacientes.html" class="list-
group-item border-end-0 d-inline-block text-truncate" data-bs-
parent="#sidebar"><i class="bi bi-heart"></i> <span>Pacientes</span></a>
                        <a href="#" class="list-group-item border-end-0 d-
inline-block text-truncate" data-bs-parent="#sidebar"><i class="bi bi-
bricks"></i> <span>Salir</span></a>

                    </div>
                </div>
            </div>
            <main class="col ps-md-2 pt-2" >
                <a href="#" data-bs-target="#sidebar" data-bs-
toggle="collapse" class="border rounded-3 p-1 text-decoration-none"><i
class="bi bi-list bi-lg py-2 p-1"></i> Menu</a>
                <a href="../pacientes/pacientes.html" ><button type="button"
class="btn btn-outline-primary p-1 " style="position: absolute; right:0;
margin-right: 1rem;">Atras</button></a>
                <br><br>


                <!--tarjeta datos Paciente-->

                <div class="card" id="nombrePaciente">

                </div>
                <br><br>

                 <h1>MEDICAMENTO</h1>
                <div class="row row-cols-1 row-cols-md-2 g-4">

                  <div class="col">

                    <div  class="card">

                      <h5 class="card-header">Compartimento A </h5>

                      <div  id="compartA" class="card-body">
                        <!--
```

```
                        <button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#modal_A">Editar</button>
                        -->
                      </div>
                    </div>

                    <!-- The Modal A -->
                    <div class="modal fade" id="modal_A">
    <div class="modal-dialog">
      <div class="modal-content">

        <!-- Modal Header -->
        <div class="modal-header">
          <h4 class="modal-title">Medicamento</h4>
          <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
        </div>

        <!-- Modal body A-->
        <div class="modal-body">
          <form>
            <div class="form-group">
                <label for="recipient-name" class="col-form-
label">Nombre:</label>
                <input id="nombre_a" type="text" class="form-control" >
           </div>

           <div class="form-group">
               <label for="recipient-name" class="col-form-
label">Cantidad:</label>
               <input id="cantidad_a" type="text" class="form-control" >
           </div>

           <!--
           <div class="form-group">
               <div class="row">
                <div class="col">
                    <label for="recipient-name" class="col-form-label">Fecha
Inicio:</label>
                <input id="dateini_a" type="date"  class="form-control" >
                  </div>

                  <div class="col">
                    <label for="recipient-name" class="col-form-label">Fecha
Finalizacion:</label>
```

```
                    <input id="datefin_a" type="date" class="form-control" >
                  </div>
                </div>
            </div>
               -->

            <div class="form-group">
                <div class="row">
                 <label for="recipient-name" class="col-form-label"> Horas por
día:</label>

                  <p>

                   <label class="col-form-label" >Una:</label>
                   <input  type="radio" name="rad"  onclick="mostrarUnoA()" />

                   <label class="col-form-label" >Dos:</label>
                   <input  type="radio" name="rad" onclick="mostrarDosA()" />

                   <label class="col-form-label">Tres:</label>
                   <input  type="radio" name="rad" checked="checked"
onclick="mostrarTresA()" />

                  </p>

                  <div id="alarma1_a" class="col">
                    <input id="halarm1_a" type="time" class="form-control" >
                  </div>

                  <div id="alarma2_a"  class="col">
                      <input id="halarm2_a" type="time" class="form-control" >
                  </div>

                  <div id="alarma3_a"   class="col">
                   <input id="halarm3_a" type="time" class="form-control" >
                  </div>
                </div>

            </div>

            <div class="form-group">
              <div class="row">
                  <label for="recipient-name" class="col-form-label">
Instrucciones:</label>
```

```
                <select  class="form-select" aria-label="Default select
example" id="instruc_a">
                  <option value="Sin Instrucciones">Sin
Instrucciones</option>
                  <option value="Antes de comer">Antes de comer</option>
                  <option value="Despues de comer">Despues de comer
</option>
                  <option value="Mientras come">Mientras come</option>
                </select>
            </div>

            <div class="form-group">
               <label for="recipient-name" class="col-form-
label">Observaciones:</label>
                <textarea id="observ_a" class="form-control"
placeholder="Observaciones" style="height: 100px"></textarea>
            </div>

        </div>
        </form>
        </div>

        <!-- Modal footer -->
        <div class="modal-footer">
          <button type="button" class="btn btn-danger" data-bs-
dismiss="modal">Cancelar</button>
          <button type="submit" id="btnsave_a" class="btn btn-
primary">Guardar</button>
        </div>

      </div>
    </div>
                  </div>

                </div>

                <div class="col">
                  <div class="card">
                    <h5 class="card-header">Compartimento B </h5>

                    <div id="compartB" class="card-body">
                      <!--
                        <button type="button" class="btn btn-primary" data-
bs-toggle="modal" data-bs-target="#modal_B">Editar</button>
                      -->
```

```
                    </div>
                  </div>


                  <!-- The Modal B -->
                  <div class="modal fade" id="modal_B">
                    <div class="modal-dialog">
                      <div class="modal-content">

                        <!-- Modal Header -->
                        <div class="modal-header">
                          <h4 class="modal-title">Medicicamento</h4>
                          <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
                        </div>

                        <!-- Modal body b -->
                        <div class="modal-body">
                          <form>
                            <div class="form-group">
                                <label for="recipient-name" class="col-form-
label">Nombre:</label>
                                <input id="nombre_b" type="text"
class="form-control" >
                            </div>

                            <div class="form-group">
                                <label for="recipient-name" class="col-form-
label">Cantidad:</label>
                                <input id="cantidad_b" type="text"
class="form-control" >
                            </div>
                             <!--
                            <div class="form-group">
                                <div class="row">
                                 <div class="col">
                                     <label for="recipient-name" class="col-
form-label">Fecha Inicio:</label>
                                  <input id="dateini_b"
type="date"  class="form-control" >
                                    </div>

                                    <div class="col">
```

```
                                    <label for="recipient-name" class="col-
form-label">Fecha Finalizacion:</label>
                                    <input id="datefin_b" type="date"
class="form-control" >
                                  </div>
                                </div>
                              </div>
                             -->

                              <div class="form-group">
                                  <div class="row">
                                   <label for="recipient-name" class="col-form-
label"> Horas por día:</label>

                                    <p>

                                     <label class="col-form-label"
>Una:</label>
                                     <input  type="radio"
name="rad"  onclick="mostrarUnoB()" />

                                     <label class="col-form-label"
>Dos:</label>
                                     <input  type="radio" name="rad"
onclick="mostrarDosB()" />

                                     <label class="col-form-
label">Tres:</label>
                                     <input  type="radio" name="rad"
checked="checked" onclick="mostrarTresB()" />

                                    </p>

                                    <div id="alarma1_b" class="col">
                                      <input id="halarm1_b" type="time"
class="form-control" >
                                    </div>

                                    <div id="alarma2_b"  class="col">
                                        <input id="halarm2_b" type="time"
class="form-control" >
                                    </div>

                                    <div id="alarma3_b"   class="col">
```

```
                                    <input id="halarm3_b" type="time"
class="form-control" >
                                </div>
                            </div>

                        </div>

                        <div class="form-group">
                          <div class="row">
                              <label for="recipient-name" class="col-
form-label"> Instrucciones:</label>

                                <select  class="form-select" aria-
label="Default select example" id="instruc_b">
                                  <option value="Sin Instrucciones">Sin
Instrucciones</option>
                                  <option value="Antes de comer">Antes de
comer</option>
                                  <option value="Despues de comer">Despues
de comer </option>
                                  <option value="Mientras come">Mientras
come</option>
                                </select>
                            </div>

                            <div class="form-group">
                               <label for="recipient-name" class="col-form-
label">Observaciones:</label>
                                <textarea id="observ_b" class="form-control"
placeholder="Observaciones" style="height: 100px"></textarea>
                              </div>
                        </div>

                        </form>
                        </div>

                        <!-- Modal footer -->
                        <div class="modal-footer">
                          <button type="button" class="btn btn-danger" data-
bs-dismiss="modal">Cancelar</button>
                          <button type="submit" id="btnsave_b" class="btn
btn-primary">Guardar</button>
                        </div>

                    </div>
```

```
                    </div>
                                    </div>
                </div>

                <div class="col">
                  <div class="card">
                    <h5 class="card-header">Compartimento C </h5>
                    <div id="compartC" class="card-body">
                      <!--
                         <button type="button" class="btn btn-primary" data-
bs-toggle="modal" data-bs-target="#modal_C">Editar</button>
                      -->
                    </div>
                  </div>
                  <!-- The Modal C -->
                  <div class="modal fade" id="modal_C">
                    <div class="modal-dialog">
                      <div class="modal-content">

                        <!-- Modal Header -->
                        <div class="modal-header">
                          <h4 class="modal-title">Medicamento</h4>
                          <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
                        </div>

                        <!-- Modal body C -->
                        <div class="modal-body">
                          <form>
                            <div class="form-group">
                                <label for="recipient-name" class="col-form-
label">Nombre:</label>

                                <input id="nombre_c" type="text"
class="form-control" >
                            </div>

                            <div class="form-group">
                                <label for="recipient-name" class="col-form-
label">Cantidad:</label>

                                <input id="cantidad_c" type="text"
class="form-control" >
                            </div>
                            <!--
                            <div class="form-group">
                                <div class="row">
```

```
                                <div class="col">
                                    <label for="recipient-name" class="col-
form-label">Fecha Inicio:</label>
                                 <input id="dateini_c"
type="date"  class="form-control" >
                                   </div>

                                   <div class="col">
                                     <label for="recipient-name" class="col-
form-label">Fecha Finalizacion:</label>
                                      <input id="datefin_c" type="date"
class="form-control" >
                                   </div>
                                </div>
                            </div>
                            -->

                            <div class="form-group">
                                <div class="row">
                                 <label for="recipient-name" class="col-form-
label"> Horas por día:</label>

                                  <p>

                                   <label class="col-form-label"
>Una:</label>
                                   <input  type="radio"
name="rad"  onclick="mostrarUnoC()" />

                                   <label class="col-form-label"
>Dos:</label>
                                   <input  type="radio" name="rad"
onclick="mostrarDosC()" />

                                   <label class="col-form-
label">Tres:</label>
                                   <input  type="radio" name="rad"
checked="checked" onclick="mostrarTresC()" />

                                  </p>

                                  <div id="alarma1_c" class="col">
                                    <input id="halarm1_c" type="time"
class="form-control" >
                                  </div>
```

```
                                    <div id="alarma2_c"  class="col">
                                        <input id="halarm2_c" type="time"
class="form-control" >
                                    </div>

                                    <div id="alarma3_c"   class="col">
                                     <input id="halarm3_c" type="time"
class="form-control" >
                                    </div>
                                  </div>

                              </div>

                              <div class="form-group">
                                <div class="row">
                                    <label for="recipient-name" class="col-
form-label"> Instrucciones:</label>

                                     <select  class="form-select" aria-
label="Default select example" id="instruc_c">
                                       <option value="Sin Instrucciones">Sin
Instrucciones</option>
                                       <option value="Antes de comer">Antes de
comer</option>
                                       <option value="Despues de comer">Despues
de comer </option>
                                       <option value="Mientras come">Mientras
come</option>
                                     </select>
                                </div>

                                <div class="form-group">
                                   <label for="recipient-name" class="col-form-
label">Observaciones:</label>
                                   <textarea id="observ_c" class="form-control"
placeholder="Observaciones" style="height: 100px"></textarea>
                                </div>

                           </div>
                        </form>
                           </div>

                           <!-- Modal footer -->
                           <div class="modal-footer">
```

```
                            <button type="button" class="btn btn-danger" data-
bs-dismiss="modal">Cancelar</button>
                            <button type="submit" id="btnsave_c" class="btn
btn-primary">Guardar</button>
                          </div>

                        </div>
                      </div>
                                    </div>
                  </div>

                  <div class="col">
                    <div class="card">
                      <h5 class="card-header">Compartimento D </h5>
                      <div  id="compartD" class="card-body">
                        <!--
                          <button type="button" class="btn btn-primary" data-
bs-toggle="modal" data-bs-target="#modal_D">Editar</button>
                        -->
                      </div>
                    </div>

                    <!-- The Modal D -->
                    <div class="modal fade" id="modal_D">
                      <div class="modal-dialog">
                        <div class="modal-content">

                          <!-- Modal Header -->
                          <div class="modal-header">
                            <h4 class="modal-title">Nuevo Paciente</h4>
                            <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
                          </div>

                          <!-- Modal body D-->
                          <div class="modal-body">
                            <form>
                              <div class="form-group">
                                  <label for="recipient-name" class="col-form-
label">Nombre:</label>

                                  <input id="nombre_d" type="text"
class="form-control" >
                             </div>

                             <div class="form-group">
```

```
                                <label for="recipient-name" class="col-form-
label">Cantidad:</label>
                                <input id="cantidad_d" type="text"
class="form-control" >
                            </div>

                            <!--
                            <div class="form-group">
                                <div class="row">
                                 <div class="col">
                                     <label for="recipient-name" class="col-
form-label">Fecha Inicio:</label>
                                  <input id="dateini_d"
type="date"  class="form-control" >
                                    </div>

                                    <div class="col">
                                      <label for="recipient-name" class="col-
form-label">Fecha Finalizacion:</label>
                                       <input id="datefin_d" type="date"
class="form-control" >
                                    </div>
                                 </div>
                            </div>
                            -->

                            <div class="form-group">
                                <div class="row">
                                 <label for="recipient-name" class="col-form-
label"> Horas por día:</label>

                                  <p>

                                   <label class="col-form-label"
>Una:</label>
                                   <input  type="radio"
name="rad"  onclick="mostrarUnoD()" />

                                   <label class="col-form-label"
>Dos:</label>
                                   <input  type="radio" name="rad"
onclick="mostrarDosD()" />

                                   <label class="col-form-
label">Tres:</label>
```

```
                                    <input  type="radio" name="rad"
checked="checked" onclick="mostrarTresD()" />

                                </p>

                                <div id="alarma1_d" class="col">
                                  <input id="halarm1_d" type="time"
class="form-control" >
                                </div>

                                <div id="alarma2_d"  class="col">
                                    <input id="halarm2_d" type="time"
class="form-control" >
                                </div>

                                <div id="alarma3_d"   class="col">
                                 <input id="halarm3_d" type="time"
class="form-control" >
                                </div>
                              </div>

                            </div>

                            <div class="form-group">
                              <div class="row">
                                  <label for="recipient-name" class="col-
form-label"> Instrucciones:</label>

                                   <select  class="form-select" aria-
label="Default select example" id="instruc_d">
                                     <option value="Sin Instrucciones">Sin
Instrucciones</option>
                                     <option value="Antes de comer">Antes de
comer</option>
                                     <option value="Despues de comer">Despues
de comer </option>
                                     <option value="Mientras come">Mientras
come</option>
                                   </select>
                                </div>

                                <div class="form-group">
                                   <label for="recipient-name" class="col-form-
label">Observaciones:</label>
```

```
                                <textarea id="observ_d" class="form-control"
placeholder="Observaciones" style="height: 100px"></textarea>
                                </div>
                            </div>
                        </form>
                            </div>

                            <!-- Modal footer -->
                            <div class="modal-footer">
                              <button type="button" class="btn btn-danger" data-
bs-dismiss="modal">Cancelar</button>
                              <button type="submit" id="btnsave_d" class="btn
btn-primary" >Guardar</button>
                            </div>

                          </div>
                        </div>
                        </div>
                    </div>



                </div>

                </div>



            </main>
        </div>
    </div>


    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js
" integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
    <script type="text/javascript" src="radio.js"></script>
    <script type="module" src="datospacientes.js"></script>
```

```
  </body>
</html>
```

Js datos pacientes

```
import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.11/firebase-app.js"
import {
  getDatabase,
  set,
  ref,
  get,
  child,
  push,
  onValue,
  update,
  remove
} from "https://www.gstatic.com/firebasejs/9.6.11/firebase-database.js";


const firebaseConfig = {
  apiKey: "AIzaSyDJFJSBdTruJXjNVnZv1kC9UPZsPN_47zY",
  authDomain: "iotapp-a1c0a.firebaseapp.com",
  databaseURL: "https://iotapp-a1c0a-default-rtdb.firebaseio.com",
  projectId: "iotapp-a1c0a",
  storageBucket: "iotapp-a1c0a.appspot.com",
  messagingSenderId: "592743603808",
  appId: "1:592743603808:web:d98581ec24b58066cbad05"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const db = getDatabase(app);


//////////coge el dato de la URL
var queryString = window.location.search;
var urlParams = new URLSearchParams(queryString);
var idUsuariodata = urlParams.get('idUsuario');
//console.log(idUsuariodata);

//////////////////////////espacios para llenar
const nombrePaciente=document.getElementById("nombrePaciente");
```

```
////*******Datos USUARIO

const dbref = ref(db);

get(child(dbref,'pacientes/' + idUsuariodata)).then((snapshot)=> {
        if (snapshot.exists()){

            const Nombrep=snapshot.val().Nombre;
            const Telfp=snapshot.val().Telf
            const Emailp=snapshot.val().Email

            nombrePaciente.innerHTML +=`

        <div class="card-header">Ficha: ${idUsuariodata}</div>

        <div class="card-body">
            <h5 class="card-title">Nombre: ${Nombrep}</h5>
                <div class="row">
                    <div class="col-6">
                            <p class="card-text">Telefono: ${Telfp}</p>

                    </div>
                    <div class="col-6">
                        <p class="card-text">Email: ${Emailp}</p>

                    </div>
            </div>

        </div>

         `
        }
        else{
            alert("No data found");
        }
})
.catch((error)=>{
        alert("unsucces"+ error);

});

//////////////////editar datos paciente
////editar datos
```

```
/*
const nombredit=document.getElementById("nombreditpaci");
const telfedit=document.getElementById("telfeditpaci");
//const emailedit=document.getElementById("emailedit");

const btnEditar=document.getElementById("btnEditarpaci");

function UpdateData(idUsuariodata){
update(ref(db,'pacientes/' + idUsuariodata),{
  Nombre: nombredit.value,
  Telf: telfedit.value,
  //Email: emailedit.value
})
.then(()=>{
   alert("data update succes");
})
.catch((error)=>{
   alert("unsucces"+ error);

});
}

window.UpdateData=UpdateData;

btnEditar.addEventListener('click',UpdateData);
*/

//*******MEDICINAAAAAAAAAA********

///////condicional mostrar datos

if(idUsuariodata== 1){
function clearmed(){
   // document.getAnimations("nombre_a").value="";
    //document.getElementById("cantidad_a").value="";
   // document.getElementById("des").value="";
}

//***************compartimento A*********************
const medNameA=document.getElementById("nombre_a");
const medcantA=document.getElementById("cantidad_a");
//fecha
//const dateiniA=document.getElementById("dateini_a");
//const datefinA=document.getElementById("datefin_a");
```

```
//hora
const hora1A=document.getElementById("halarm1_a");
const hora2A=document.getElementById("halarm2_a");
const hora3A=document.getElementById("halarm3_a");

//instruciones
const instrucA=document.getElementById("instruc_a");

//observaciones
const observA=document.getElementById("observ_a");
const btnguardarA= document.getElementById("btnsave_a");

///editar datos
    btnguardarA.addEventListener('click', function(e){
        e.preventDefault();

        update(ref(db,'caja/A'),{
            Nombre: medNameA.value,
            Cantidad: medcantA.value,
            Alarma1:hora1A.value,
            Alarma2:hora2A.value,
            Alarma3:hora3A.value,
            Instruciones:instrucA.value,
            Observaviones:observA.value
        })
        .then(()=>{
            alert("datos guardados");

        })
        .catch((error)=>{
            alert("unsucces"+ error);

        });
    })


///visualizar datos A
const cardcompartA=document.getElementById("compartA");


get(child(dbref,'caja/A')).then((snapshot)=> {
    if (snapshot.exists()){

        const nomrecp_a=snapshot.val().Nombre;
```

```
        const cantrecp_a=snapshot.val().Cantidad;
        const alarm1recp_a=snapshot.val().Alarma1;
        const alarm2recp_a=snapshot.val().Alarma2;
        const alarm3recp_a=snapshot.val().Alarma3;
        const instrecp_a=snapshot.val().Instruciones;
        const observrecp_a=snapshot.val().Observaviones;
        const flexA_alarm1=snapshot.val().flexA_alar1;
        const flexA_alarm2=snapshot.val().flexA_alar2;
        const flexA_alarm3=snapshot.val().flexA_alar3;


        cardcompartA.innerHTML +=`


        <h5 class="card-title">Nombre: ${nomrecp_a}</h5>
           <div class="row">
                <div class="col-6">
                    <p class="card-text">Cantidad(g): ${cantrecp_a}</p>
                </div>


                <div class="col-6">
                  <p class="card-text">Alarma1: ${alarm1recp_a}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma2: ${alarm2recp_a}</p>
                </div>
                <div class="col-6">
                  <p class="card-text">Alarma3: ${alarm3recp_a}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma1: ${flexA_alarm1}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma2: ${flexA_alarm2}</p>
                </div>
                <div class="col-6">
                  <p class="card-text"> EstadoAlarma3: ${flexA_alarm3}</p>
                </div>


                <div class="col-6">
```

```
                  <p class="card-text">Instruciones: ${instrecp_a}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Observaviones: ${observrecp_a}</p>
                </div>
            </div>
            <button type="button" class="btn btn-primary"  data-bs-
toggle="modal" data-bs-target="#modal_A">Editar</button>

        </div>

    `
    }
    else{
        alert("No data found");
    }
})
.catch((error)=>{
    alert("unsucces"+ error);

});


//***************compartimento B*********************
const medNameB=document.getElementById("nombre_b");
const medcantB=document.getElementById("cantidad_b");
//fecha
//const dateiniB=document.getElementById("dateini_b");
//const datefinB=document.getElementById("datefin_b");

//hora
const hora1B=document.getElementById("halarm1_b");
const hora2B=document.getElementById("halarm2_b");
const hora3B=document.getElementById("halarm3_b");

//instruciones
const instrucB=document.getElementById("instruc_b");

//observaciones
const observB=document.getElementById("observ_b");
const btnguardarB= document.getElementById("btnsave_b");

    btnguardarB.addEventListener('click', function(e){
        e.preventDefault();
```

```
        update(ref(db,'caja/B'),{
            Nombre: medNameB.value,
            Cantidad: medcantB.value,
            //Finicio:dateiniB.value,
            //Ffinal:datefinB.value,
            Alarma1:hora1B.value,
            Alarma2:hora2B.value,
            Alarma3:hora3B.value,
            Instruciones:instrucB.value,
            Observaviones:observB.value
        })
        .then(()=>{
            alert("datos editados");

        })
        .catch((error)=>{
            alert("unsucces"+ error);

        });
    })

///visualizar datos B
const cardcompartB=document.getElementById("compartB");


get(child(dbref,'caja/B')).then((snapshot)=> {
    if (snapshot.exists()){

        const nomrecp_b=snapshot.val().Nombre;
        const cantrecp_b=snapshot.val().Cantidad;
        //const finirecp_b=snapshot.val().Finicio;
        //const ffinrecp_b=snapshot.val().Ffinal;
        const alarm1recp_b=snapshot.val().Alarma1;
        const alarm2recp_b=snapshot.val().Alarma2;
        const alarm3recp_b=snapshot.val().Alarma3;
        const instrecp_b=snapshot.val().Instruciones;
        const observrecp_b=snapshot.val().Observaviones;
        const flexB_alarm1=snapshot.val().flexB_alar1;
        const flexB_alarm2=snapshot.val().flexB_alar2;
        const flexB_alarm3=snapshot.val().flexB_alar3;


        cardcompartB.innerHTML +=`
```

```
        <h5 class="card-title">Nombre: ${nomrecp_b}</h5>
           <div class="row">
                <div class="col-6">
                    <p class="card-text">Cantidad(g): ${cantrecp_b}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma1: ${alarm1recp_b}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma2: ${alarm2recp_b}</p>
                </div>
                <div class="col-6">
                  <p class="card-text">Alarma3: ${alarm3recp_b}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma1: ${flexB_alarm1}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma2: ${flexB_alarm2}</p>
                </div>
                <div class="col-6">
                  <p class="card-text"> EstadoAlarma3: ${flexB_alarm3}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Instruciones: ${instrecp_b}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Observaviones: ${observrecp_b}</p>
                </div>
           </div>
           <button type="button" class="btn btn-primary"  data-bs-
toggle="modal" data-bs-target="#modal_B">Editar</button>

        </div>

    `

    }
    else{
```

```
        alert("No data found");
    }
})
.catch((error)=>{
    alert("unsucces"+ error);

});


//***************compartimento C*********************
const medNameC=document.getElementById("nombre_c");
const medcantC=document.getElementById("cantidad_c");
//fecha
//const dateiniC=document.getElementById("dateini_c");
//const datefinC=document.getElementById("datefin_c");

//hora
const hora1C=document.getElementById("halarm1_c");
const hora2C=document.getElementById("halarm2_c");
const hora3C=document.getElementById("halarm3_c");

//instruciones
const instrucC=document.getElementById("instruc_c");

//observaciones
const observC=document.getElementById("observ_c");
const btnguardarC= document.getElementById("btnsave_c");

    btnguardarC.addEventListener('click', function(e){
        e.preventDefault();

        update(ref(db,'caja/C'),{
            Nombre: medNameC.value,
            Cantidad: medcantC.value,
            //Finicio:dateiniC.value,
            //Ffinal:datefinC.value,
            Alarma1:hora1C.value,
            Alarma2:hora2C.value,
            Alarma3:hora3C.value,
            Instruciones:instrucC.value,
            Observaviones:observC.value
        })
        .then(()=>{
            alert("datos editados");
```

```
        })
        .catch((error)=>{
            alert("unsucces"+ error);

        });
    })

///visualizar datos C
const cardcompartC=document.getElementById("compartC");


get(child(dbref,'caja/C')).then((snapshot)=> {
    if (snapshot.exists()){

        const nomrecp_c=snapshot.val().Nombre;
        const cantrecp_c=snapshot.val().Cantidad;

        const alarm1recp_c=snapshot.val().Alarma1;
        const alarm2recp_c=snapshot.val().Alarma2;
        const alarm3recp_c=snapshot.val().Alarma3;
        const instrecp_c=snapshot.val().Instruciones;
        const observrecp_c=snapshot.val().Observaviones;
        const flexC_alarm1=snapshot.val().flexC_alar1;
        const flexC_alarm2=snapshot.val().flexC_alar2;
        const flexC_alarm3=snapshot.val().flexC_alar3;

        cardcompartC.innerHTML +=`


        <h5 class="card-title">Nombre: ${nomrecp_c}</h5>
           <div class="row">
                <div class="col-6">
                    <p class="card-text">Cantidad(g): ${cantrecp_c}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma1: ${alarm1recp_c}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma2: ${alarm2recp_c}</p>
                </div>
                <div class="col-6">
                  <p class="card-text">Alarma3: ${alarm3recp_c}</p>
                </div>
```

```
                <div class="col-6">
                  <p class="card-text">Estado Alarma1: ${flexC_alarm1}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma2: ${flexC_alarm2}</p>
                </div>
                <div class="col-6">
                  <p class="card-text"> EstadoAlarma3: ${flexC_alarm3}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Instruciones: ${instrecp_c}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Observaviones: ${observrecp_c}</p>
                </div>
            </div>
            <button type="button" class="btn btn-primary"  data-bs-
toggle="modal" data-bs-target="#modal_C">Editar</button>

        </div>

     `
    }
    else{
        alert("No data found");
    }
})
.catch((error)=>{
    alert("unsucces"+ error);

});



 //**************compartimento D*******************
const medNameD=document.getElementById("nombre_d");
const medcantD=document.getElementById("cantidad_d");
//fecha
///const dateiniD=document.getElementById("dateini_d");
//const datefinD=document.getElementById("datefin_d");
```

```
//hora
const hora1D=document.getElementById("halarm1_d");
const hora2D=document.getElementById("halarm2_d");
const hora3D=document.getElementById("halarm3_d");

//instruciones
const instrucD=document.getElementById("instruc_d");

//observaciones
const observD=document.getElementById("observ_d");
const btnguardarD= document.getElementById("btnsave_d");

    btnguardarD.addEventListener('click', function(e){
        e.preventDefault();

        update(ref(db,'caja/D'),{
            Nombre: medNameD.value,
            Cantidad: medcantD.value,
            //Finicio:dateiniD.value,
           // Ffinal:datefinD.value,
            Alarma1:hora1D.value,
            Alarma2:hora2D.value,
            Alarma3:hora3D.value,
            Instruciones:instrucD.value,
            Observaviones:observD.value
        })
        .then(()=>{
            alert("datos editados");

        })
        .catch((error)=>{
            alert("unsucces"+ error);

        });
    })

    ///visualizar datos D
const cardcompartD=document.getElementById("compartD");


get(child(dbref,'caja/D')).then((snapshot)=> {
    if (snapshot.exists()){

        const nomrecp_d=snapshot.val().Nombre;
        const cantrecp_d=snapshot.val().Cantidad;
```

```
        //const finirecp_d=snapshot.val().Finicio;
        //const ffinrecp_d=snapshot.val().Ffinal;
        const alarm1recp_d=snapshot.val().Alarma1;
        const alarm2recp_d=snapshot.val().Alarma2;
        const alarm3recp_d=snapshot.val().Alarma3;
        const instrecp_d=snapshot.val().Instruciones;
        const observrecp_d=snapshot.val().Observaviones;
        const flexD_alarm1=snapshot.val().flexD_alar1;
        const flexD_alarm2=snapshot.val().flexD_alar2;
        const flexD_alarm3=snapshot.val().flexD_alar3;

        cardcompartD.innerHTML +=`


        <h5 class="card-title">Nombre: ${nomrecp_d}</h5>
           <div class="row">
                <div class="col-6">
                    <p class="card-text">Cantidad(g): ${cantrecp_d}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma1: ${alarm1recp_d}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Alarma2: ${alarm2recp_d}</p>
                </div>
                <div class="col-6">
                  <p class="card-text">Alarma3: ${alarm3recp_d}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma1: ${flexD_alarm1}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Estado Alarma2: ${flexD_alarm2}</p>
                </div>
                <div class="col-6">
                  <p class="card-text"> EstadoAlarma3: ${flexD_alarm3}</p>
                </div>

                <div class="col-6">
                  <p class="card-text">Instruciones: ${instrecp_d}</p>
                </div>
```

```
                <div class="col-6">
                  <p class="card-text">Observaviones: ${observrecp_d}</p>
                </div>
            </div>
            <button type="button" class="btn btn-primary"  data-bs-
toggle="modal" data-bs-target="#modal_D">Editar</button>

        </div>

     `
    }
    else{
        alert("No data found");
    }
})
.catch((error)=>{
    alert("unsucces"+ error);

});

}
else{

  alert("Usuario sin caja de medicamentos, comuniquese con el proveedor")
}
```

Anexo C. Código Android Studio

Página Principal

```
package iotmed.ximiot.cajamediot;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

```
public class MainActivity extends AppCompatActivity {

    TextView estatecaja;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        estatecaja=findViewById(R.id.txtcajastate);
        FirebaseDatabase database= FirebaseDatabase.getInstance();
        DatabaseReference cajastateRef =
database.getReference("caja/Caja_state");
        cajastateRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                String boxstatevlaue= dataSnapshot.getValue(String.class);
                estatecaja.setText("Caja " + boxstatevlaue);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }



    public void irAcompartA(View view){
        //this esta ventana y a la que queremos ir
        Intent i=new Intent(this,compartA.class);
        startActivity(i);
    }

    public void irAcompartB(View view){
        //this esta ventana y a la que queremos ir
        Intent i=new Intent(this,compartB.class);
        startActivity(i);
    }

    public void irAcompartC(View view){
        //this esta ventana y a la que queremos ir
        Intent i=new Intent(this,compartC.class);
        startActivity(i);
    }

    public void irAcompartD(View view){
        //this esta ventana y a la que queremos ir
        Intent i=new Intent(this,compartD.class);
        startActivity(i);
    }
}
```

Compartimento A

```
package iotmed.ximiot.cajamediot;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class compartA extends AppCompatActivity {

    TextView namedata_A;
    TextView cantdata_A;
    TextView instdata_A;
    TextView obserdata_A;
    TextView alarm1data_A;
    TextView alarm2data_A;
    TextView alarm3data_A;
    TextView flexA_alarma1;
    TextView flexA_alarma2;
    TextView flexA_alarma3;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comparta);

        namedata_A=findViewById(R.id.txtnamedataA);
        cantdata_A=findViewById(R.id.txtcantdataA);
        instdata_A=findViewById(R.id.txtinstrucdataA);
        obserdata_A=findViewById(R.id.txtobservadataA);
        alarm1data_A=findViewById(R.id.alarm1dataA);
        alarm2data_A=findViewById(R.id.alarm2dataA);
        alarm3data_A=findViewById(R.id.alarm3dataA);
        flexA_alarma1=findViewById(R.id.flexA_alarm1) ;
        flexA_alarma2=findViewById(R.id.flexA_alarm2) ;
        flexA_alarma3=findViewById(R.id.flexA_alarm3) ;



        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference cajARef = database.getReference();


        cajARef.child("caja/A").addValueEventListener(new
ValueEventListener() {
```

```
            @Override
            public void onDataChange(@NonNull DataSnapshot datasnapshot) {
               if(datasnapshot.exists()){


                   String
nombreA=datasnapshot.child("Nombre").getValue().toString();

                   namedata_A.setText(nombreA);

                   String
cantidadA=datasnapshot.child("Cantidad").getValue().toString();
                   cantdata_A.setText(cantidadA + " " + "g");

                   String
istrucA=datasnapshot.child("Instruciones").getValue().toString();
                   instdata_A.setText(istrucA);

                   String
observA=datasnapshot.child("Observaviones").getValue().toString();
                   obserdata_A.setText(observA);

                   String
alarma1A=datasnapshot.child("Alarma1").getValue().toString();
                   alarm1data_A.setText(alarma1A);

                   String
alarma2A=datasnapshot.child("Alarma2").getValue().toString();
                   alarm2data_A.setText(alarma2A);

                   String
alarma3A=datasnapshot.child("Alarma3").getValue().toString();
                   alarm3data_A.setText(alarma3A);

                   String flexa_alr1=
datasnapshot.child("flexA_alar1").getValue().toString();
                   flexA_alarma1.setText(flexa_alr1);

                   String flexa_alr2=
datasnapshot.child("flexA_alar2").getValue().toString();
                   flexA_alarma2.setText(flexa_alr2);

                   String flexa_alr3=
datasnapshot.child("flexA_alar3").getValue().toString();
                   flexA_alarma3.setText(flexa_alr3);
               }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
```

```
    }
}
```

Compartimento B

```
package iotmed.ximiot.cajamediot;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class compartB extends AppCompatActivity {


    TextView namedata_B;
    TextView cantdata_B;
    TextView instdata_B;
    TextView obserdata_B;
    TextView alarm1data_B;
    TextView alarm2data_B;
    TextView alarm3data_B;
    TextView flexB_alarma1;
    TextView flexB_alarma2;
    TextView flexB_alarma3;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compart_b);

        namedata_B=findViewById(R.id.txtnamedataB);
        cantdata_B=findViewById(R.id.txtcantdataB);
        instdata_B=findViewById(R.id.txtinstrucdataB);
        obserdata_B=findViewById(R.id.txtobservadataB);
        alarm1data_B=findViewById(R.id.alarm1dataB);
        alarm2data_B=findViewById(R.id.alarm2dataB);
        alarm3data_B=findViewById(R.id.alarm3dataB);
        flexB_alarma1=findViewById(R.id.flexB_alarm1) ;
        flexB_alarma2=findViewById(R.id.flexB_alarm2) ;
        flexB_alarma3=findViewById(R.id.flexB_alarm3) ;
```

```
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference cajBRef = database.getReference();



        cajBRef.child("caja/B").addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot datasnapshot) {
                if(datasnapshot.exists()){


                    String
nombreB=datasnapshot.child("Nombre").getValue().toString();

                    namedata_B.setText(nombreB);

                    String
cantidadB=datasnapshot.child("Cantidad").getValue().toString();
                    cantdata_B.setText(cantidadB + " " + "g");

                    String
istrucB=datasnapshot.child("Instruciones").getValue().toString();
                    instdata_B.setText(istrucB);

                    String
observB=datasnapshot.child("Observaviones").getValue().toString();
                    obserdata_B.setText(observB);

                    String
alarma1B=datasnapshot.child("Alarma1").getValue().toString();
                    alarm1data_B.setText(alarma1B);

                    String
alarma2B=datasnapshot.child("Alarma2").getValue().toString();
                    alarm2data_B.setText(alarma2B);

                    String
alarma3B=datasnapshot.child("Alarma3").getValue().toString();
                    alarm3data_B.setText(alarma3B);

                    String flexB_alr1=
datasnapshot.child("flexB_alar1").getValue().toString();
                    flexB_alarma1.setText(flexB_alr1);

                    String flexB_alr2=
datasnapshot.child("flexB_alar2").getValue().toString();
                    flexB_alarma2.setText(flexB_alr2);

                    String flexB_alr3=
datasnapshot.child("flexB_alar3").getValue().toString();
                    flexB_alarma3.setText(flexB_alr3);
                }
            }
```

```
            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });

    }

}
```

Compartimento C

```
package iotmed.ximiot.cajamediot;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class compartC extends AppCompatActivity {

    TextView namedata_C;
    TextView cantdata_C;
    TextView instdata_C;
    TextView obserdata_C;
    TextView alarm1data_C;
    TextView alarm2data_C;
    TextView alarm3data_C;
    TextView flexC_alarma1;
    TextView flexC_alarma2;
    TextView flexC_alarma3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compart_c);

        namedata_C=findViewById(R.id.txtnamedataC);
        cantdata_C=findViewById(R.id.txtcantdataC);
        instdata_C=findViewById(R.id.txtinstrucdataC);
        obserdata_C=findViewById(R.id.txtobservadataC);
        alarm1data_C=findViewById(R.id.alarm1dataC);
        alarm2data_C=findViewById(R.id.alarm2dataC);
```

```
        alarm3data_C=findViewById(R.id.alarm3dataC);
        flexC_alarma1=findViewById(R.id.flexC_alarm1) ;
        flexC_alarma2=findViewById(R.id.flexC_alarm2) ;
        flexC_alarma3=findViewById(R.id.flexC_alarm3) ;




        FirebaseDatabase databaseC = FirebaseDatabase.getInstance();
        DatabaseReference cajCRef = databaseC.getReference();


        cajCRef.child("caja/C").addValueEventListener(new
ValueEventListener()  {
            @Override
            public void onDataChange(@NonNull DataSnapshot datasnapshot) {
                if(datasnapshot.exists()){


                    String
nombreC=datasnapshot.child("Nombre").getValue().toString();

                    namedata_C.setText(nombreC);

                    String
cantidadC=datasnapshot.child("Cantidad").getValue().toString();
                    cantdata_C.setText(cantidadC + " " + "g");

                    String
istrucC=datasnapshot.child("Instruciones").getValue().toString();
                    instdata_C.setText(istrucC);

                    String
observC=datasnapshot.child("Observaviones").getValue().toString();
                    obserdata_C.setText(observC);

                    String
alarma1C=datasnapshot.child("Alarma1").getValue().toString();
                    alarm1data_C.setText(alarma1C);

                    String
alarma2C=datasnapshot.child("Alarma2").getValue().toString();
                    alarm2data_C.setText(alarma2C);

                    String
alarma3C=datasnapshot.child("Alarma3").getValue().toString();
                    alarm3data_C.setText(alarma3C);

                    String flexC_alr1=
datasnapshot.child("flexB_alar1").getValue().toString();
                    flexC_alarma1.setText(flexC_alr1);

                    String flexC_alr2=
datasnapshot.child("flexB_alar2").getValue().toString();
                    flexC_alarma2.setText(flexC_alr2);
```

```
                    String flexC_alr3=
datasnapshot.child("flexB_alar3").getValue().toString();
                    flexC_alarma3.setText(flexC_alr3);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });

    }
}
```

Compartimento D

```
package iotmed.ximiot.cajamediot;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class compartD extends AppCompatActivity {
    TextView namedata_D;
    TextView cantdata_D;
    TextView instdata_D;
    TextView obserdata_D;
    TextView alarm1data_D;
    TextView alarm2data_D;
    TextView alarm3data_D;
    TextView flexD_alarma1;
    TextView flexD_alarma2;
    TextView flexD_alarma3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compart_d);

        namedata_D=findViewById(R.id.txtnamedataD);
```

```
        cantdata_D=findViewById(R.id.txtcantdataD);
        instdata_D=findViewById(R.id.txtinstrucdataD);
        obserdata_D=findViewById(R.id.txtobservadataD);
        alarm1data_D=findViewById(R.id.alarm1dataD);
        alarm2data_D=findViewById(R.id.alarm2dataD);
        alarm3data_D=findViewById(R.id.alarm3dataD);
        flexD_alarma1=findViewById(R.id.flexD_alarm1) ;
        flexD_alarma2=findViewById(R.id.flexD_alarm2) ;
        flexD_alarma3=findViewById(R.id.flexD_alarm3) ;



        FirebaseDatabase databaseD = FirebaseDatabase.getInstance();
        DatabaseReference cajDRef = databaseD.getReference();


        cajDRef.child("caja/D").addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot datasnapshot) {
                if(datasnapshot.exists()){


                    String
nombreD=datasnapshot.child("Nombre").getValue().toString();

                    namedata_D.setText(nombreD);

                    String
cantidadD=datasnapshot.child("Cantidad").getValue().toString();
                    cantdata_D.setText(cantidadD + " " + "g");

                    String
istrucD=datasnapshot.child("Instruciones").getValue().toString();
                    instdata_D.setText(istrucD);

                    String
observD=datasnapshot.child("Observaviones").getValue().toString();
                    obserdata_D.setText(observD);

                    String
alarma1D=datasnapshot.child("Alarma1").getValue().toString();
                    alarm1data_D.setText(alarma1D);

                    String
alarma2D=datasnapshot.child("Alarma2").getValue().toString();
                    alarm2data_D.setText(alarma2D);

                    String
alarma3D=datasnapshot.child("Alarma3").getValue().toString();
                    alarm3data_D.setText(alarma3D);

                    String flexD_alr1=
datasnapshot.child("flexB_alar1").getValue().toString();
                    flexD_alarma1.setText(flexD_alr1);
```

```
                    String flexD_alr2=
datasnapshot.child("flexB_alar2").getValue().toString();
                    flexD_alarma2.setText(flexD_alr2);

                    String flexD_alr3=
datasnapshot.child("flexB_alar3").getValue().toString();
                    flexD_alarma3.setText(flexD_alr3);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
}
```