

Sistema de vigilancia basado en componentes de bajo coste

Ridouan El bachiri

Resumen—Este proyecto consiste en la implementación de un sistema de vigilancia para mejorar la seguridad del hogar, utilizando componentes de bajo coste (placas programables y sensores) con el fin de captar la información de las acciones del hogar y enviarla a una base de datos en tiempo real (Firebase), que posteriormente será visualizada a través de una aplicación móvil propia en Android. El sistema se basa en controlar y minimizar los factores de riesgo internos y externos.

Palabras clave—Seguridad, microcontrolador, sensores, Android, hogar, aplicación, domótica, placa programable, Arduino Uno, ESP32 WROM, ESP32CAM, sistema, digital, analógico, API, circuito integrado, prototipo, java, c++

Abstract—This project consists of the implementation of a surveillance system to improve home security, using low-cost components (programmable boards and sensors) in order to capture the information on the actions of the home and send it to a database in real time (Firebase), which will later be viewed through its own mobile application on Android. The system is based on controlling and minimizing internal and external risk factors.

Index Terms—Security, micro-controller, sensors, Android, home, application, home Automation, programable board, Arduino Uno, ESP32 WROM, ESP32 CAM, System, digital, analogical, API, integrated circuit, prototype, java, c++

◆

1 INTRODUCCIÓN

El hecho de tener un sistema de seguridad en el hogar aumenta la sensación de tranquilidad y confianza. Por lo contrario, siempre está la posibilidad de que suceda algo y eso puede causar problemas, que no solo suelen ser pérdidas de materiales, sino pueden tener efectos personales, familiares y pérdida de confianza en el hogar. Ya que el hogar es aquel lugar donde se fraguan los vínculos más íntimos y personales, contiene a quien más se ama y lo que más se desea proteger [1].

Para dar una solución y mejorar algunos problemas relacionados con la seguridad del hogar, surge este proyecto, que se basa en un prototipo de sistema de vigilancia exterior e interior.

Con tal de prevenir riesgos exteriores, se desarrolla un módulo de control mediante sensores de presencia [12] para supervisar el estado de puertas y ventanas. Y, en cuanto a riesgos interiores, se desarrollan los módulos que controlan el estado de gas [13], agua [13], movimiento [13], iluminación [14], temperatura [15] y humedad [15], mediante los diferentes sensores indicados en el apartado de componentes 6.1.

Para agregar más seguridad, se desarrolla un módulo que consiste en una cámara de vigilancia capaz de realizar reconocimiento facial [16] y capturar imágenes cuando hay un acceso al

hogar.

En este documento se explica el proceso de desarrollo del proyecto que consiste en la especificación de objetivos, establecimiento de la metodología, selección de material y recursos, planificación, diseño e implementación del proyecto y, finalmente, la experimentación de los resultados.

2 ESTADO DEL ARTE

Frecuentemente, cuando se habla de la seguridad del hogar, se destacan los sistemas anti-intrusión en mecanismos que previenen actos vandálicos, robos y otros percances procedentes principalmente del exterior, algo absolutamente comprensible pero que responde a una visión incompleta y sesgada si no se amplía la perspectiva [1]. Es decir, se deben tratar otros asuntos y considerar amenazas que pueden suceder en el interior y que puedan poner en riesgo tanto a personas como a infraestructuras o bienes.

Actualmente, existen muchas alternativas en el mercado que ofrecen servicios de seguridad mediante la tecnología, por ejemplo, el “sistema de alarmas de securitasDirect” [2] o las “alarmas para casa en Amazon” [3]. Estas alternativas se basan en la implantación de sensores de movimiento en puertas o ventanas, con el fin de activar la alarma en el caso de detectar movimiento.

La mayoría de estas alternativas, que no están al alcance de todos debido a su elevado coste, se basan en videovigilancia, alarmas o combinación de ambas empleando un software de

-
- *E-mail de contacto:* ridouan.elbachiri@e-campus.uab.cat
 - *Mención realizada:* Ingeniería de computadores.
 - *Trabajo tutorizado por:* Diego Javier Mostaccio Mancini
 - *Curso* 2019/20

control propietario. Además, éstas solo se enfocan en controlar un aspecto en concreto como puede ser el movimiento [2] [3]. En cambio, en este proyecto se pretende controlar más aspectos internos como el estado del gas, agua, temperatura y humedad con un mismo sistema de control.

3 OBJETIVOS

En este apartado se define tanto el objetivo principal como los objetivos específicos que se desean lograr con el desarrollo del proyecto.

3.1 Objetivo principal

El objetivo principal consiste en desarrollar un **sistema de vigilancia basado en componentes de bajo coste** para mejorar la seguridad del hogar.

3.2 Objetivos específicos

Los objetivos específicos se basan en los pasos que se llevarán a cabo para cumplir el objetivo principal.

- Planificación de las tareas necesarias para cada etapa del desarrollo del proyecto y a su vez establecer la metodología.
- Diseñar un sistema de control de vigilancia del hogar.
- Diseño de los módulos que componen el sistema y a su vez la conexión de los componentes con la unidad central.
- Diseño de la aplicación móvil y hardware del proyecto.
- Selección de componentes electrónicos adecuados para el proyecto, microcontroladores y distintos sensores.
- Estudiar y familiarizarse con los componentes electrónicos seleccionados. Así mismo, la capacidad de decidir los componentes según el propósito del proyecto.
- Diseño e implementación de los controladores de cámara y sensores.
- Establecer la conectividad del sistema central con Internet mediante conexión wifi.
- Comunicación del sistema central con la base de datos, y a su vez, enviar y recibir información en tiempo real.
- Implementación de la cámara de vigilancia con la posibilidad de aplicar el reconocimiento facial.
- Realizar las comprobaciones necesarias del funcionamiento del sistema.

4 METODOLOGÍA Y PLANIFICACIÓN

Para llevar a cabo el proyecto, se ha optado por utilizar la metodología V [4] debido a la facilidad de desarrollo de cada módulo de forma independiente y su posterior integración en el sistema una vez comprobado.

La metodología V tiene como objetivo principal describir las actividades y resultados que se producen durante el desarrollo del proyecto, y a la vez, proporciona una guía para su planificación y realización.

La finalidad de la metodología V es:

- Minimizar los riesgos del proyecto.
- Mejorar y garantizar la calidad.
- Reducción de gastos totales del proyecto.

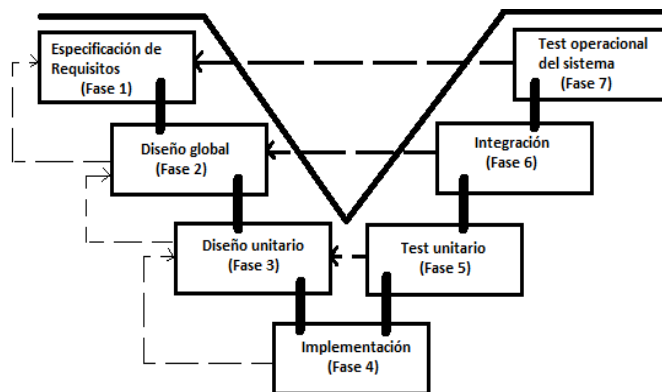


Fig 1. Ciclo de vida del proyecto (metodología V).

El proyecto se ha desarrollado siguiendo las fases que se observan en la *Figura 1*. Estas fases están definidas por la metodología en [4]:

- **DEFINICIÓN DE ESPECIFICACIONES (Fase 1):** Se deben definir y documentar los diferentes requisitos del sistema a desarrollar, identificando los valores numéricos más concretos posibles. Entre ellos debe estar la especificación del nivel de integridad.
- **DISEÑO GLOBAL (Fase 2):** También llamado diseño de alto nivel. Su objetivo es obtener un diseño y visión general del sistema. Un diseño que engloba todo el funcionamiento del sistema.
- **DISEÑO EN DETALLADO (Fase 3):** Consiste en detallar cada bloque de la fase anterior. para cada función del sistema que se desarrolla.
- **IMPLEMENTACIÓN (Fase 4):** Es la fase en la que se materializa el diseño en detalle.
- **TEST UNITARIO DE CADA BLOQUE (Fase 5):** En esta fase se verifica cada módulo HW y SW de forma unitaria, comprobando su funcionamiento adecuado.
- **INTEGRACIÓN DE CADA BLOQUE (Fase 6):** En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse que ante la presencia de un fallo persiste el funcionamiento correcto.
- **TEST OPERACIONAL DEL SISTEMA (Fase 7):** Se realizan las últimas pruebas pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos.

Las fases 3, 4, 5 y 6 se realizan para cada módulo y la 1, 2 y 7 se aplican solo una vez. De esta forma, se obtiene un desarrollo por iteraciones como indica la *Figura 2*, donde cada iteración desarrolla un módulo que, a su vez, consta de una fase de comprobación que puede provocar una iteración en caso de que éste no cumpla los requisitos.

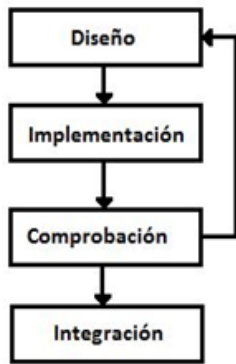


Fig 2. Diagrama de iteración de cada módulo.

Se ha realizado una planificación inicial mediante el diagrama de actividades donde se han listado las fases del proyecto y las tareas involucradas en cada fase. En el apéndice 1, se observa la planificación en el tiempo mediante el diagrama de Gantt [A1] para poder obtener una estimación de la evolución del proyecto y lograr finalizarlo en la fecha indicada. Por otro lado, en el apéndice 2 se observan las tareas gestionadas mediante la herramienta Trello [A2].

Dicha planificación consta de 7 fases:

1. Estudio y análisis.
2. Diseño del sistema.
3. Aprendizaje y selección del material y recursos.
4. Implementación.
5. Comprobación.
6. Creación de informes.
7. Reuniones y entregas.

5 DISEÑO DEL SISTEMA

5.1 Diagrama de bloques

La figura 3 muestra la representación del sistema mediante diagrama de bloques que se visualiza a continuación.

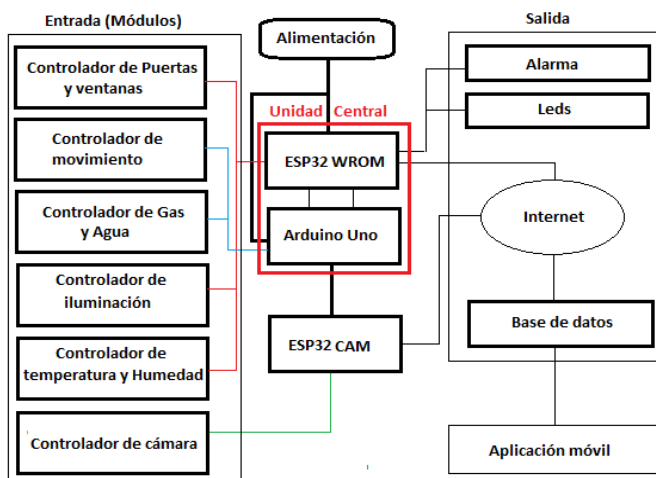


Fig 3. Diagrama de bloques.

Los módulos se conectan con la unidad central como entradas, donde se envía la información a los microcontroladores que forman la unidad central y se procesa siguiendo la lógica que se explica en el apartado 5.2. Los módulos de movimiento,

gas y agua, se conectan con Arduino Uno a través de las entradas analógicas y, el módulo controlador de puertas/ventanas, temperatura e iluminación se conectan con ESP32WROOM a través de entradas digitales. El módulo de la cámara utiliza la placa ESP32CAM que ya integra una unidad de procesamiento y con conectividad a Internet. Para la comunicación entre Arduino Uno y ESP32WROOM se utiliza el bus I2C como se explica en el apartado 5.3.

La unidad central envía la información a la base de datos, donde comunica dicha información con la aplicación móvil.

5.2 Lógica del sistema

La lógica aplicada se basa en dar respuesta a los eventos captados por los sensores y procesados por los microcontroladores, donde se aplica la lógica del diagrama de flujos de la Figura 4 y Figura 5.

El sistema tiene un modo de seguridad que puede activar y desactivar el usuario. Cuando está activado, éste expresa que el usuario está inseguro o bien no se encuentra en el hogar, de manera que siempre que el sistema detecte alguna indicación de riesgo, la notificará al usuario. En cambio, con el modo desactivado, éste sólo enviará alertas que se consideran importantes (alertas de gas, agua y temperatura).

La representación de la lógica se ha diseñado mediante diagramas de flujos y se ha dividido en sub-diseños para una mejor interpretación. La Figura 4 representa la iteración del sistema, donde se obtiene los datos de los módulos y después se ejecuta la función para comprobar las alertas y el estado de seguridad [Figura 5].

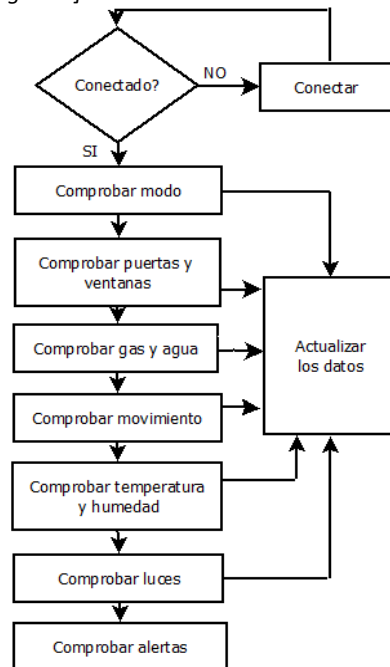


Fig 4. Diagrama de flujos de la iteración del sistema.

En la figura 4 se observa la iteración donde se comprueba el modo de seguridad con la base de datos. Seguidamente, se comprueba el estado de puertas/ventanas, gas, agua, movimiento, temperatura y humedad y se actualiza la información en

la base de datos.

A continuación se comprueban el estado de las luces con la base de datos para encenderlas o apagarlas y por último, se comprueban las alertas [Figura 5].

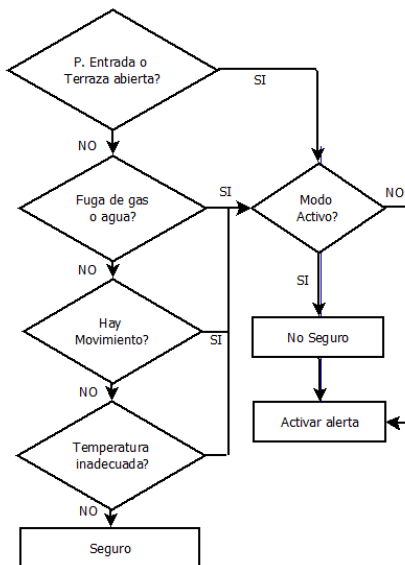


Fig 5. Diagrama de flujos para la comprobación de alertas.

La figura 5 representa la función que comprueba los datos obtenidos de cada módulo para conocer el estado de seguridad y así notificar las alertas.

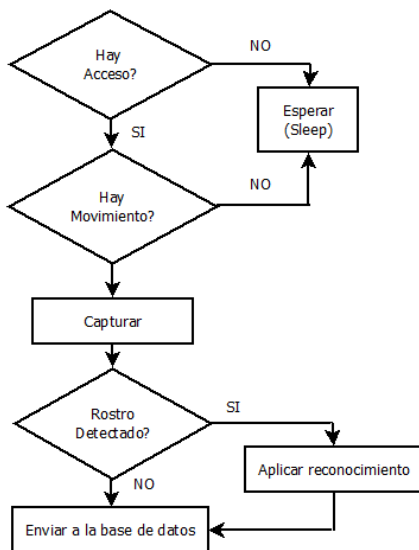


Fig 6. Diagrama de flujos del módulo de la cámara.

Se captura la imagen cuando hay acceso o movimiento en el hogar y ésta es enviada a la base de datos. En caso de detectar el rostro de la persona, se realiza un reconocimiento facial con el fin de identificar un acceso autorizado [Figura 6].

5.3 Comunicación

La comunicación entre Arduino Uno y ESP32WROOM se mantiene utilizando un bus I2C. Esta comunicación permite conectar un microcontrolador principal (master) y otro secundario (worker) para poder repartir la carga.

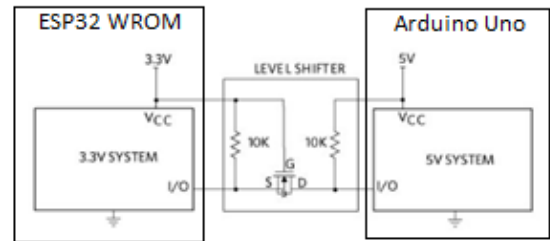


Fig 7. Conexión I2C entre Arduino Uno y ESP21 WROOM utilizando un convertidor a nivel.

El nivel de voltaje con el que funciona cada microcontrolador es diferente, para ello se ha utilizado un convertidor a nivel [Figura 7] para la comunicación entre Arduino Uno y ESP32WROOM [6].

Para la comunicación de los módulos con la aplicación, se implementan funciones para interactuar con la base de datos Firebase.

5.4 Aplicación móvil

Inicialmente se han diseñado las diferentes interfaces que componen la aplicación móvil [A4] y el "Diagrama de clases" [A3]. La aplicación se ha desarrollado en Android mediante el entorno de desarrollo Android Studio.



Fig 8. Interfaz principal.

En la figura 8 se muestra la Interfaz principal, donde se visualiza el estado de seguridad, configuración del modo de seguridad y activación o desactivación de la alarma.

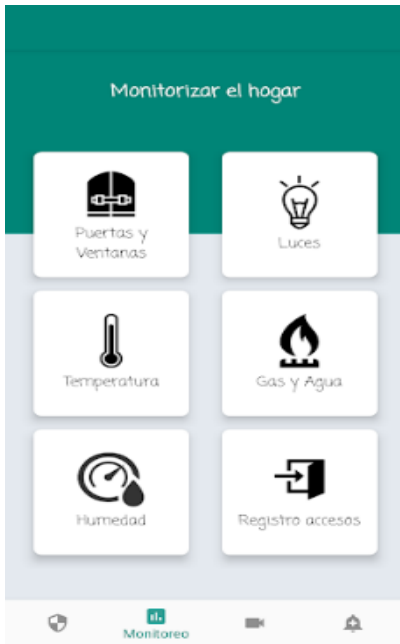


Fig 9. Interfaz para monitorizar

La Interfaz de la figura 9 muestra las diferentes opciones para monitorizar los registros relacionados con el estado de las puertas, iluminación, gas, agua, temperatura y humedad.



Fig 10. Interfaz de alertas

En la figura 10 se visualiza un listado cronológico de alertas del hogar.

5.5 Maqueta

Se ha construido una maqueta de madera representando una casa con microcontroladores y sensores instalados para captar la información.



Fig 11. Foto de la maqueta

En el apartado del apéndice A5 se pueden visualizar más fotos de la maqueta y desde diferentes perspectivas.

6 SELECCIÓN DE MATERIAL Y RECURSOS

En este apartado se especifican los materiales y recursos necesarios para el desarrollo el proyecto con su respectivo presupuesto.

6.1 Componentes utilizados

Para la unidad central de control se ha utilizado microcontroladores Arduino Uno y ESP32WROM por su bajo coste.



ESP32WROOM



Arduino Uno

Para el módulo de la cámara se ha utilizado ESP32CAM que integra conectividad wifi, FTDI para su programación y Servomotor para moverla.



ESP32CAM



Programador FTDI



Servomotor

En la implementación del módulo controlador de puertas y ventanas se ha utilizado el sensor de presencia, y para el de gas y agua, el sensor MQ2 y el de agua.



Sensor de presencia



Sensor MQ2

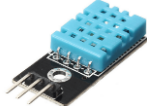


Sensor de agua

Se ha utilizado el sensor PIR para el módulo de movimiento y el sensor DHT11 para el de temperatura y humedad. Estos son los más utilizados por sus librerías ya implementadas.



Sensor PIR



Sensor DHT11

En cuanto al módulo controlador de iluminación se ha utilizado leds y buzzer para la implementación de la alarma.

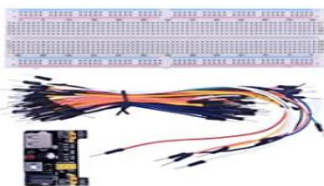


Leds

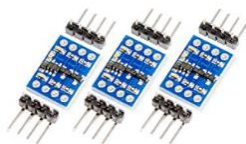


Buzzer

Finalmente, para poder programar los componentes se ha utilizado una Brotboard con cables para su conexión y el conversor a nivel para la comunicación I2C.



Brotboard y cables de conexión



Conversor a nivel

6.2 Presupuesto

Listado de costes de los componentes utilizados en el proyecto y su respectivo presupuesto final.

TABLA 1
PRESUPUESTO DEL MATERIAL UTILIZADO

COMPONENTE	CANTIDAD	PRECIO (€)
ESP32 WROM	1	4
ESP32 CAM	1	8
Arduino Uno	1	10
Sensor MQ2	1	6
Sensor DHT11	1	3.95
Leds	10	4
Sensores de presencia	8	4
Sensor de agua	1	1.70
Servo motor	1	4
Brotboard + Cables de conexión	1	8
Programador FTDI	1	7
Buzzer	1	2
Conversor a nivel	1	2.25
Soldador	1	15
TOTAL		85,80

Aparte del coste de los componentes, existe un coste de espacio de trabajo. Investigando sobre los “coworkings en Barcelona” [18], se ha encontrado una oferta de 70€/mes que no ha sido necesaria contratar ya que se ha realizado en espacio propio y ha supuesto un ahorro de aproximadamente 280€ (70€ x 4 meses).

6.3 Entornos de desarrollo

Se han utilizado los siguientes entornos de desarrollo:

- **Firestore [8]:** Base de datos en tiempo real.
- **Arduino IDE [25]:** Programación de los microcontroladores y sensores.
- **Android studio [26]:** Diseño y creación de la aplicación móvil.
- **Tinkercad [27]:** Diseño de las conexiones de sensores con microcontroladores.

6.3 Librerías utilizadas

Por otra parte, se han utilizado algunas librerías ya implementadas que ofrecen los fabricantes de los componentes utilizados. Estas librerías se describen a continuación:

- **Wifi.h [20]:** Implementa las funciones necesarias para conectar los microcontroladores mediante conexión wifi.
- **Firestore.h [19]:** Implementa las funciones para la comunicación e interacción con la base de datos Firestore.
- **Dht11.h [15]:** Implementa el sensor de temperatura y humedad (Dht11). Requiere de una inicialización con el pin del sensor.
- **FirestoreJson.h [19]:** Encapsula la información en objetos JSON y enviarla a la base de datos.
- **Servo.h [22]:** Implementa las funcionalidades de los servo motores.
- **NTPClient.h [21]:** Obtiene la fecha y hora de un servidor externo.
- **base64.h [10]:** Implementa las funciones para codificar y decodificar imágenes en base64.
- **esp_camera.h [23]:** Inicializa los registros de la cámara y mapeo de los pines.

6.4 Lenguajes utilizados

Para programar los módulos y la aplicación móvil se ha utilizado los siguientes lenguajes:

- **Java:** Desarrollo de la aplicación.
- **C:** Librerías ya implementadas.
- **C++:** Implementación de módulos y librerías.
- **XML:** Diseño de las interfaces de la aplicación.

7 IMPLEMENTACIÓN Y RESULTADOS

En este apartado se describen los módulos implementados y la forma en la que se conectan los componentes con la unidad central.

7.1 Módulo controlador de puertas y ventanas

Este módulo se basa en controlar el estado, abierto o cerrado, de puertas y ventanas mediante sensores de presencia que indican el estado “on/off”.

“Documentación de la implementación del controlador de puertas y ventanas” [12].

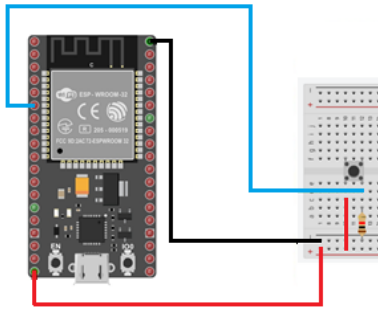


Fig 12. Conexión del sensor de presencia con ESP32 WROOM.

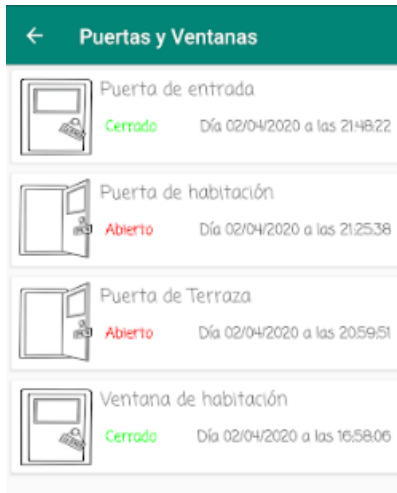


Fig 13. Listado de puertas y ventanas con sus estados.

En la interfaz de la figura 13 se visualiza el estado de puertas y ventanas en tiempo real junto a la fecha y hora del último cambio.

7.2 Módulo controlador de iluminación

La finalidad de este módulo es controlar la iluminación utilizando leds, con opciones de encendido/apagado a través de la aplicación.

“Documentación de la implementación” [14].

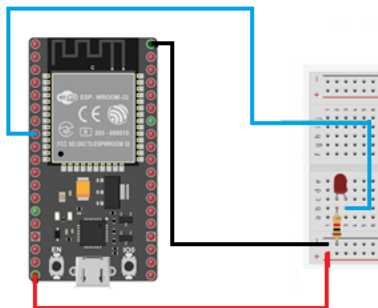


Fig 14. Conexión del led con ESP32WROOM.



Fig 15. Listado de luces y estados.

En la figura 15 se visualiza la interfaz que muestra el listado de luces y la posibilidad de encendido/apagado.

7.3 Módulo controlador de gas y agua

Para la implementación del módulo de gas y agua, se utiliza el sensor MQ2 para gas y el de agua indicados en el apartado 6.1.

Para la conexión se utiliza la placa Arduino Uno, que comunica los valores detectados al módulo ESP32WROOM mediante el bus I2C [apartado 5.3].

“Documentación de la implementación” [13].

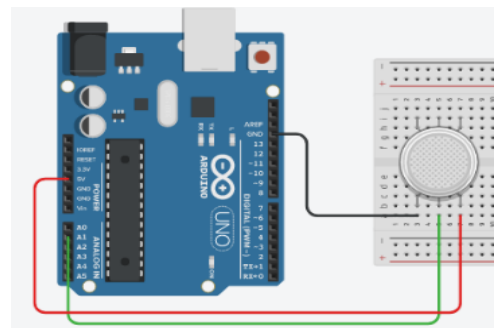


Fig 16. Conexión del sensor de gas con Arduino Uno.

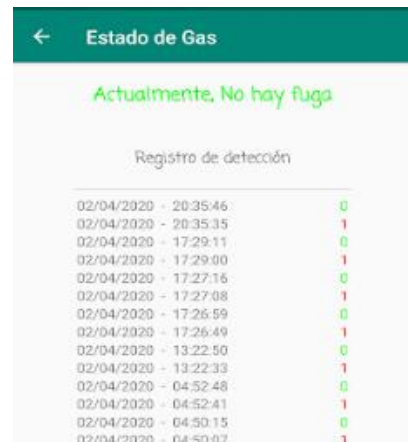


Fig 17. Estado y registros de gas

La interfaz de la figura 17 muestra el estado de gas en tiempo real y la lista de registros; 1 indica la existencia de fuga y 0 el paso de desaparición de ésta.

7.4 Módulo controlador de temperatura y humedad

Para la implementación de este módulo se utiliza una librería externa ya implementada, "DHT11.h" [15], que implementa las funciones de lectura de la temperatura y humedad.

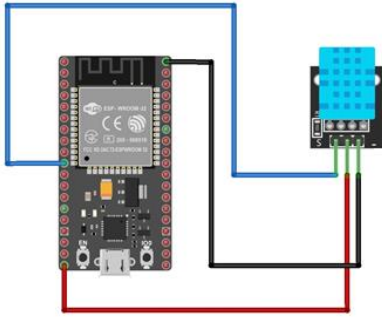


Fig 18. Conexión del sensor DHT11 con ESP32WROOM.



Fig 19. Registros de temperatura en tiempo real.

En la figura 19 se muestra la interfaz donde se visualiza el valor de la temperatura y registros de la misma con fecha y hora.



Fig 20. Registros de la humedad.

En la interfaz de la figura 20 se visualiza el valor de humedad y registros de la misma con fecha y hora.

7.5 Módulo controlador de movimiento

Este módulo capaz de detectar actividad a través de un sensor de movimiento PIR [6.1], envía una alerta a la base de datos y activa la cámara cuando éste detecta cualquier tipo de acción.

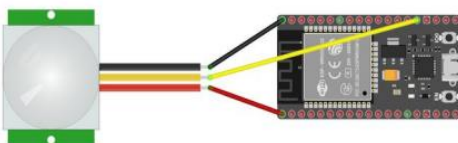


Fig 21. Conexión del sensor de movimiento con ESP32 CAM.

7.6 Módulo controlador de cámara

El módulo de la cámara se implementa utilizando la placa ESP32CAM que integra cámara y conexión Wifi. Este módulo captura imágenes cuando hay acceso o movimiento al hogar y realiza reconocimiento facial cuando detecta un rostro [28]. Documentación relacionada con la implementación del módulo de la cámara [16].

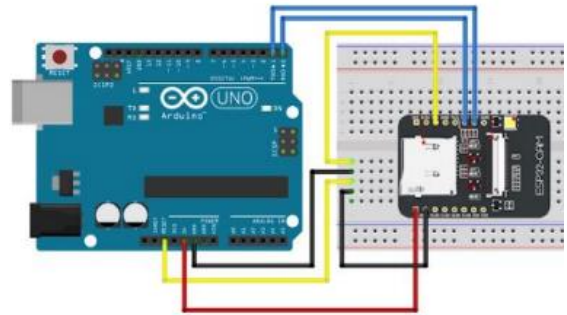


Fig 22. Conexión UART entre ESP32 CAM y Arduino UNO.

La figura 22 muestra la conexión y alimentación de la placa ESP32CAM con Arduino Uno mediante comunicación serial a través del bus UART. Para programar la placa ESP32CAM se utiliza un programador FTDI [6.1].



Fig 23. Interfaz de la cámara.

La Interfaz para cambiar la posición de la cámara y visualizar la última captura con fecha, hora y tipo de acceso como se observa en la figura 23. Además se puede visualizar los registros de capturas anteriores.

8 CONCLUSIONES

En este apartado se exponen las conclusiones obtenidas después del desarrollo, tanto las generales como las personales. La idea del proyecto se basa en mejorar la seguridad del hogar implementando un sistema de vigilancia de bajo coste, que cumple los objetivos indicados en el apartado 2.

Una de las conclusiones finales ha sido el acierto en la planificación y metodología elegida para llevar a cabo el proyecto. En cuanto a la planificación, no ha habido grandes problemas aunque es cierto que ha habido componentes que han tardado en llegar, especialmente el módulo ESP32CAM. Sin embargo, se ha ido implementando de forma paralela la parte de aplicación móvil con la construcción de la maqueta y así recompensar el retraso.

Una de las principales ventajas a nivel de desarrollo, ha sido tener conocimientos previos en la creación de aplicaciones móviles en Android con base de datos Firebase. Esto ha facilitado de forma significativa el desarrollo del proyecto.

Antes de comenzar la fase de implementación se ha investigado y familiarizado previamente con los entornos y componentes involucrados en el proyecto, de esta manera la fase de desarrollo se ha realizado de forma correcta.

Durante el desarrollo, uno de los módulos que más dificultades ha supuesto es el módulo de la cámara, tanto para enviar imágenes a la base de datos como para la lectura de la tarjeta SD. Se ha solucionado codificando las imágenes en base64 [10] y enviándolas a la base de datos. Ésto ha requerido de más tiempo para familiarizarse con el funcionamiento del módulo y encontrar soluciones tanto para enviar las imágenes como la forma en la que se procesan para el reconocimiento facial [11].

A nivel personal, este proyecto me ha ofrecido la oportunidad de desempeñar los conocimientos obtenidos durante mi formación además de obtener nuevos conocimientos y experiencias sobre microcontroladores, sensores y formas de interconectarlos mediante una aplicación.

8.1 Líneas de futuro

Inicialmente, el proyecto se basa en un prototipo para un sistema de vigilancia que cubre algunos aspectos relacionados con la seguridad del hogar, pero también se puede adaptar a otros establecimientos o entorno cuyas características sean similares.

El módulo de la cámara, además de captar imágenes, realiza un reconocimiento facial para posibilitar el control de acceso al hogar, entornos con accesos autorizados o espacios de Trabajo para fichar.

En cuanto a las imágenes capturadas, estudiar la posibilidad de mejorar la resolución de las mismas y la opción de utilizar la cámara como cámara IP y (para) acceder desde la aplicación.

El sistema está controlado por una aplicación móvil propia que permite actuar sobre las luces en la que, cara al futuro, podría implementarse una mejora de apertura y cierre a distancia de las puertas.

AGRADECIMIENTOS

En primer lugar, agradezco los consejos y soporte de mi tutor durante el desarrollo del proyecto, cumpliendo con las sesio-

nes de tutoría, en este caso virtuales, debido a la situación que hemos y estamos pasando. También agradezco la formación ofrecida por parte de los profesores que he tenido durante mi formación en el grado.

Por último, agradezco a mi familia el apoyo y opiniones que me han brindado sobre el proyecto, que me ha ayudado para reflexionar y tomar decisiones.

BIBLIOGRAFIA

- [1] "Problemas con sistemas de alarmas actuales en el hogar", publicado el 12 de diciembre del 2018. Artículo disponible en <https://www.ingenio2010.es/problemas-con-los-sistemas-de-alarmas-actuales-en-el-hogar/>
- [2] "Sistemas de alarmas para el hogar". Sistemas de seguridad basados en alarmas. Servicios de Securitas Direct. <https://www.securitasdirect.es/alarmas-hogar>
- [3] "Alarmas para casa". Sensor de movimiento para puertas y ventanas. Productos de Amazon. <https://www.amazon.es/>
- [4] "Método-V". Artículo de Wikipedia sobre la metodología de desarrollo V. Disponible en https://es.wikipedia.org/wiki/M%C3%A9todo_en_V
- [5] "Datashet ESP32 Series", versión 3.4, actualizado por Expressif System en 2020. Disponible en. https://www.espressif.com/sites/default/files/documentation/esp32_datashet_en.pdf
- [6] "Comunicación serial entre Arduino Uno y Node MCU", publicado el 12 de diciembre del 2018. Foro de Arduino. <https://forum.arduino.cc/index.php?topic=578936.0>
- [7] "HOW TO LEVEL SHIFT 1-WIRE SYSTEMS", Artículo sobre conversión a nivel, publicado por Irvin E. Salais el 7 de noviembre del 2019. Disponible en. <https://www.maximintegrated.com/en/design/technical-documents/app-notes/7/7106.html>
- [8] "Firebase". Artículo de Wikipedia sobre la base de datos en tiempo real. Disponible en <https://en.wikipedia.org/wiki/Firebase>
- [9] "Manual de usuario sensor MQ2" datashet del sensor de gas MQ2, Artículo publicado por jiankai.li, disponible en <https://docs.particle.io/assets/datasheets/electronsensorkit/MQ-2.pdf>
- [10] "base64.h". Librería para codificar y decodificar en base64 en C. Publicado por "tomykair" en Github. <https://gist.github.com/tomykaira/f0fd86b6c73063283afe550bc5d77594>
- [11] "base64 decode snippet in c++". Problemas con caracteres, en la decodificar en base64, publicado hace 8 meses en el foro stackoverflow. Disponible en <http://stackoverflow.com/questions/180947/base64-decode-snippet-in-cc>
- [12] "Implementación del módulo controlador de las puertas y ventanas". Documentación del proyecto, disponible en Drive. <https://drive.google.com/drive/folders/1ROg0W2u0Sg-wgGqDjZuFeFdHQUG2CB1h?usp=sharing>
- [13] "Implementación del módulo controlador de Gas", compatible con el sensor de agua y movimiento. Documentación del proyecto, disponible en Drive. <https://drive.google.com/drive/folders/1IsToWMPjmZiSPBUj2Uhs7eAuq0MYeC8C?usp=sharing>
- [14] "Implementación del módulo controlador de la iluminación". Documentación del proyecto, disponible en Drive.

- https://drive.google.com/drive/folders/1RITa_M2UiAj3EIXQWlYxzWpH_ZAH0B25?usp=sharing
- [15] "DHT-Sensor-Library". Librería de la implementación del sensor de temperatura y humedad. Publicado por "adafruit2" en Github. <https://github.com/adafruit/DHT-sensor-library>
- [16] "Implementación del módulo de la cámara". Documentación del proyecto, disponible en Drive. <https://docs.google.com/document/d/1Xzl-Hx6GyaYc9U3UzY7AXmcl-VGz8UonCaOKRe-yzz8/edit?usp=sharin>
- [17] "Arduino Uno". Datasheet de Arduino Uno. Disponible en <https://www.farnell.com/datasheets/1682209.pdf>
- [18] "Coworking" Espacio de Trabajo para el desarrollo del proyecto. Presupuesto de https://onework.com/en/plan_categories/5-weekly-pass
- [19] Firebase-ESP32 "Firebase.h y FirebaseJson.h". Librería implementada para interactuar con base de datos Firebase. Disponible en Github. <https://github.com/mobizt/Firebase-ESP32/tree/master/src>
- [20] "Wifi.h". Librería para conectividad wifi en ESP32. Disponible en Github. <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h>
- [21] "NTPClient.h". Implementa funciones para obtener fecha y hora actual. Disponible en Github. <https://github.com/arduino-libraries/NTPClient/blob/master/NTPClient.h>
- [22] "Servo.h". Implementa funciones del servomotor. Disponible en Arduino. <https://www.arduino.cc/en/reference/servo>
- [23] "esp_camera.h". Librería de la cámara ESP32CAM. Disponible en Github. https://github.com/espressif/esp32-camera/blob/master/driver/include/esp_camera.h
- [24] "Trello". Plataforma para gestionar las tareas de proyectos. Disponible en línea. <https://trello.com/>
- [25] "Arduino IDE". Entorno de desarrollo para programación de microcontroladores. Disponible de forma gratuita en <https://www.arduino.cc/en/main/software>
- [26] "Android Studio". Entorno para programar en Android. Disponible de forma gratuita en <https://developer.android.com/>
- [27] "Tinkercad". Entorno en línea para diseños en 3 dimensiones. Disponible en <https://www.tinkercad.com/>
- [28] "Face recognition_ESP32CAM". Parte de código para el reconocimiento facial con ESP32CAM. Disponible en Github. https://github.com/fustyles/Arduino/tree/master/ESP32-CAM_FaceRecognition_SD_EnrollFace

APÉNDICE

A1. Diagrama de Gantt

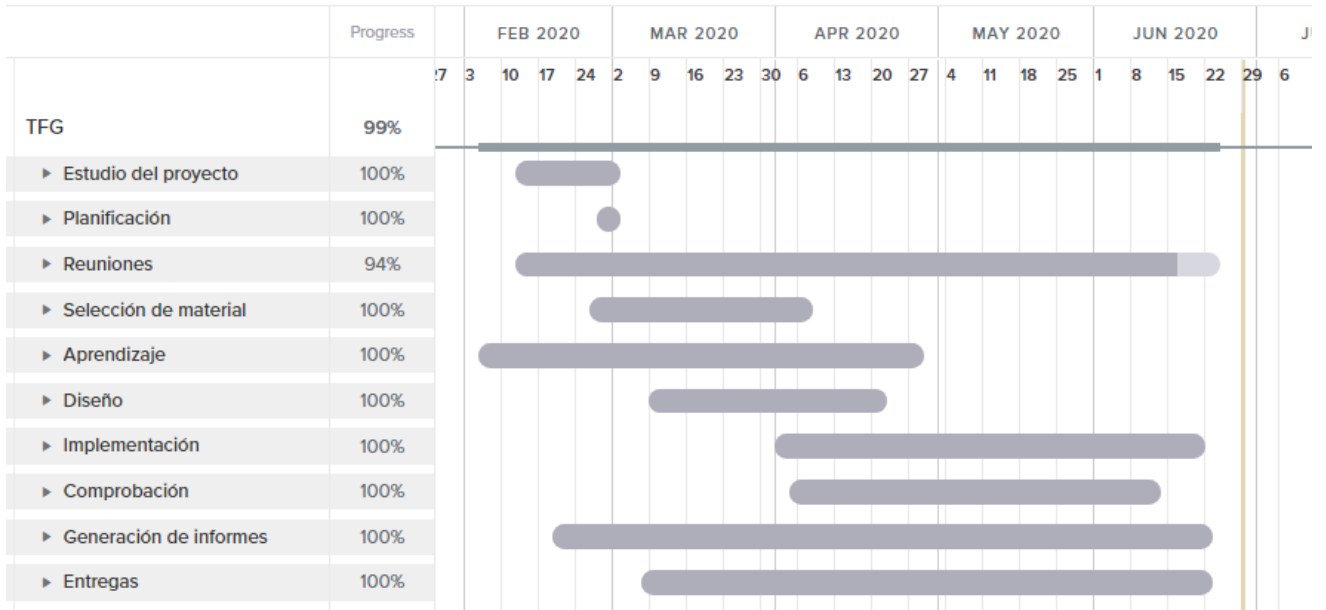


Fig 24. Diagrama de gantt. Se planifican las tareas asignadas para el proyecto las cuales se agrupan por fases dependiendo de la forma que indica la figura. También se tienen en cuenta las reuniones, creación y entregas de informes.

A2. Gestión de tareas en Trello



Fig 25. Gestión de las tareas planificadas. Dichas tareas pueden clasificarse por 4 estados, no iniciadas, en proceso, realizadas o descartadas. Durante el desarrollo todas las tareas empiezan en el estado no iniciadas y van cambiando de estado según su evolución.

A3. DIAGRAMA DE CLASES

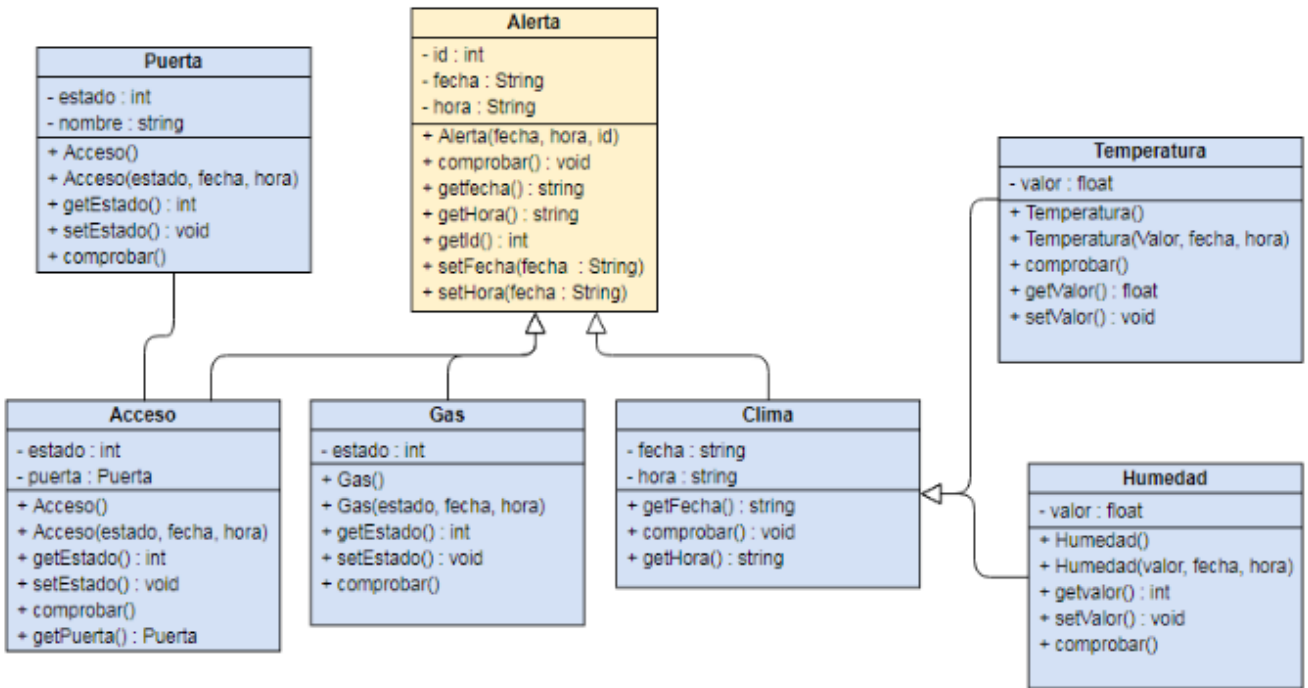


Fig 26. Diagrama de clases que indica la relación entre los objetos que forman parte de la lógica de la aplicación móvil. Inicialmente para cada sensor se requiere de un clase (objeto) para tratar los datos que se obtienen y se visualizan en las interfaces.

A4. INTERFACES DE LA APLICACIÓN MÓVIL

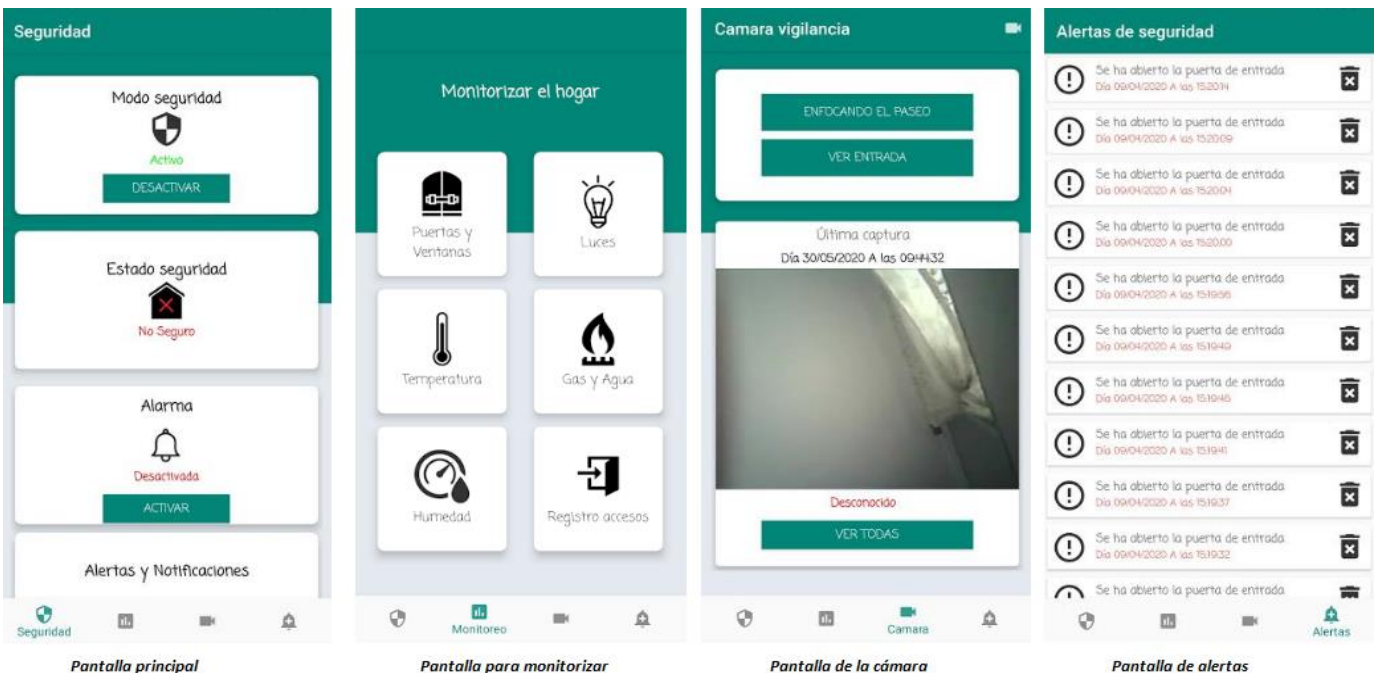
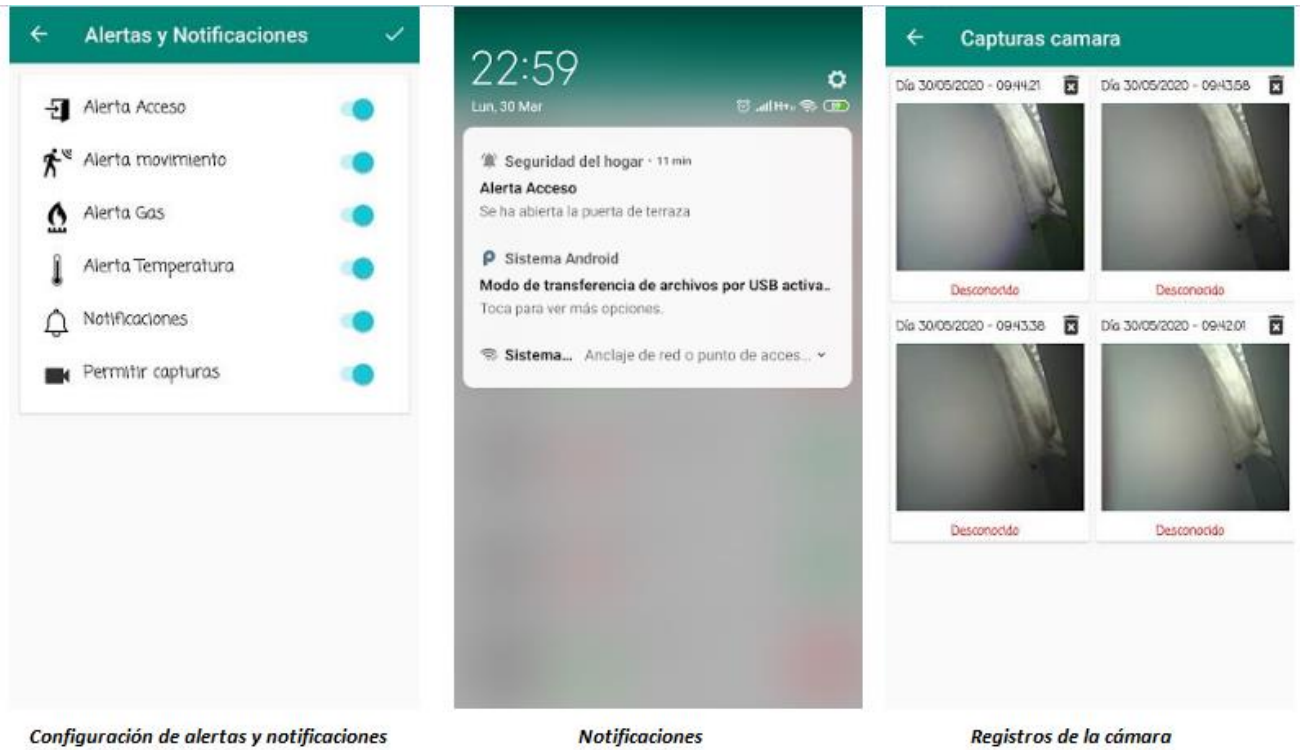


Fig 27. Interfaces principales de la aplicación las cuales se agrupan en “bottom navigation”. En la primera interfaz se visualiza el estado y modo de seguridad con la opción de configurar la alarma y las notificaciones. En la segunda interfaz se visualizan los diferentes elementos que se pueden monitorizar en tiempo real. La tercera interfaz se relaciona con la cámara, última captura y configuración de la cámara. En la última interfaz se muestra un listado de alertas.



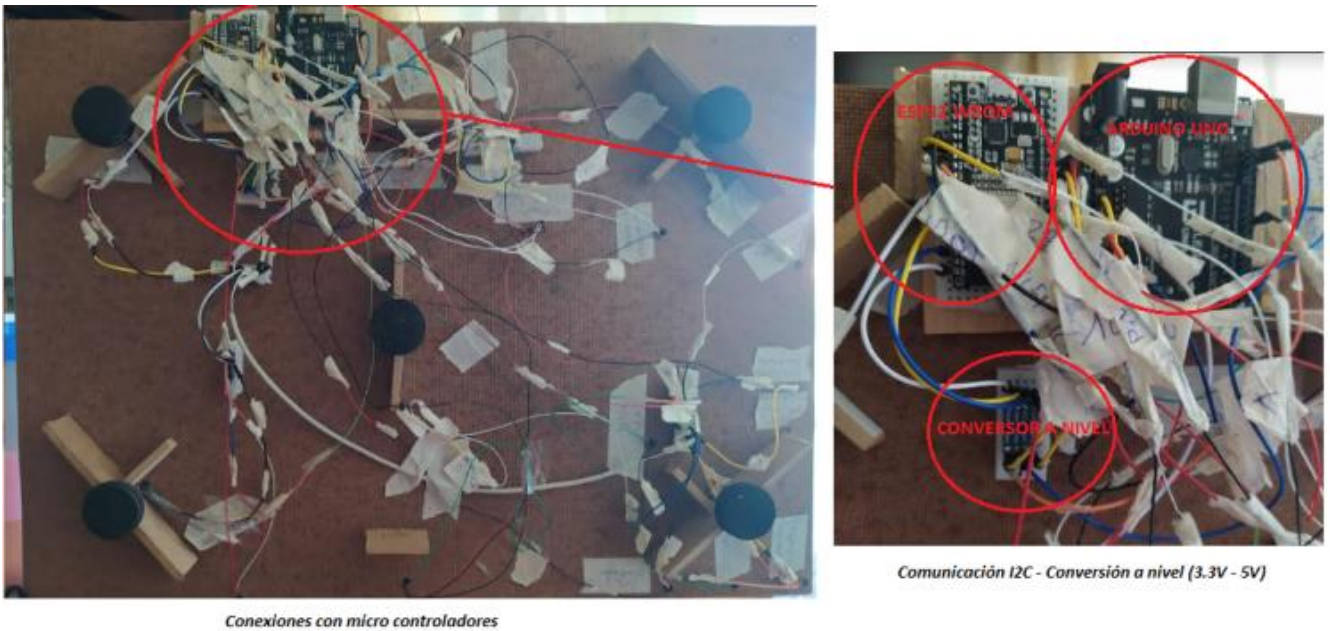
Configuración de alertas y notificaciones

Notificaciones

Registros de la cámara

Fig 28. Interfaces secundarias para la configuración, visualización de notificaciones y registros de la cámara. Existen otras interfaces ya mencionadas anteriormente en el apartado de implementación y resultados.

A5. FOTOS DE LA MAQUETA



Conexiones con micro controladores

Comunicación I2C - Conversión a nivel (3.3V - 5V)

Fig 29. Fotos de la conexión de los microcontroladores en la maqueta y comunicación entre ambos utilizando conversor a nivel.



Fig 30. Fotos de la maqueta donde se indica cada elemento (Sensores, leds y cámara) instalados. Se ha diseñado la maqueta para representar el sistema implementado de forma visual.



Fig 31. Fotos de la maqueta de diferentes ángulos.