



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Alarma para el hogar basada en Raspberry Pi

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Valiente Lerma, Arturo José

Tutor: Acebrón Linuesa, Floreal

Curso 2018/2019

Resumen

El objetivo es desarrollar una alarma económica para el hogar usando una Raspberry PI, un sensor de presencia y un conjunto de elementos emisores de sonido y software libre. La Alarma podrá armarse al abandonar el hogar conectando por Wi-Fi a la Raspberry a través de una aplicación Android. Desde el interior de la casa podrá armarse y desarmarse simplemente pulsando un botón. Si se activa alguno de los sensores de presencia se pondrán en marcha los emisores de sonido, que deben emitir un sonido muy fuerte y estridente. Si algún miembro de la familia llega a casa cuando la alarma esta armada, puede conectarse a la Raspberry con su móvil y una vez identificado proceder a desarmarla. También se grabarán vídeos de los momentos en los cuales la alarma está activada y se guardarán en la memoria local de nuestra Raspberry Pi.

Palabras clave: Raspberry Pi, Alarma hogar, Android

Abstract

The objective is to develop an economic alarm for the home using one Raspberry PI, a presence sensor and a set of transmitting sound and free software components. Alarm will arm themselves to leave the home by connecting by Wi-Fi to the Raspberry through an Android app. From the inside of the house can be assembled or disassembled simply by pressing a button. If any of the presence sensors implemented sound emitters, which should make a very strong and strident sound. If any member of the family comes home when the alarm is armed, it can connect to the Raspberry with your phone and an identified time proceed to disarm it. Also, it will record videos of the moments in which the alarm is activated and it will be stored in the local storage of our Raspberry Pi

Keywords: Raspberry Pi, Home Alarm, Android



Tabla de contenidos

1. Introducción	9
1.1. Motivación	9
1.2. Objetivos	10
1.3. Estructura	10
2. Estado del arte	13
2.1. Alarmas comerciales con dispositivos dependientes.....	13
2.2. Alarmas comerciales con dispositivos independientes.....	14
3. Hardware utilizado	16
3.1. Raspberry Pi	16
3.2. Detector de movimiento	17
3.3. Zumbador de advertencia	18
3.4. Webcam Logitech C270.....	18
3.5. Pulsador de emergencia.....	19
3.6. Protoboard	20
4. Software utilizado	22
4.1. Sistema operativo	22
4.2. Firebase	23
4.2.1. Firebase Realtime Database	23
4.2.2. Firebase Authentication.....	23
4.2.3. Firebase Cloud Messaging	24
4.3. No-IP	24
4.4. Java Secure Channel.....	25
4.5. Motion	25
4.6. Python	26
4.7. Android Studio	26
5. Diseño y construcción del circuito	29
6. Desarrollo del programa controlador	34
7. Desarrollo de la aplicación Android.....	36
8. Pruebas	42
9. Posibles mejoras futuras	48
10. Conclusiones	50
11. Bibliografía.....	52
Apéndice A: Glosario.....	57



Índice de figuras

Figura 1. Dispositivos ofrecidos por la empresa Securitas Direct	13
Figura 2. Alarma Chuango G5	14
Figura 3. Raspberry Pi con sus pines de entrada y salida	16
Figura 4. Sensor PIR	17
Figura 5. Zumbador de advertencia	18
Figura 6. Cámara Logitech C270	19
Figura 7. Pulsador de emergencia	19
Figura 8. Estructura de una protoboard	20
Figura 9. Interfaz de Raspbian en nuestra Raspberry Pi	22
Figura 10. Reglas que protegen el acceso a los datos de la aplicación	23
Figura 11. Panel de administración de No-IP	25
Figura 12. Interfaz de Android Studio	26
Figura 13. Emulador de Android Studio	27
Figura 14. Foto pines entrada y salida de Raspberry Pi	29
Figura 15. Correlación pines de entrada y salida Raspberry Pi	29
Figura 16. Esquema eléctrico del pulsador	31
Figura 17. Parte interior de nuestra alarma	31
Figura 18. Parte exterior de nuestra alarma	32
Figura 19. Ventana de inicio de sesión	36
Figura 20. Ventana de recuperar contraseña	37
Figura 21. Ventana principal	37
Figura 22. Ventana ver vídeos	37
Figura 23. Añadir Firebase a proyecto Android Studio	38
Figura 24. Consola de administración de Firebase	39

Índice de tablas

Tabla 1. Prueba del sistema 1	42
Tabla 2. Prueba del sistema 2	43
Tabla 3. Prueba del sistema 3	43
Tabla 4. Prueba del sistema 4	44
Tabla 5. Prueba del sistema 5	44
Tabla 6. Prueba del sistema 6	45
Tabla 7. Prueba del sistema 7	45
Tabla 8. Prueba del sistema 8	46
Tabla 9. Prueba del sistema 9	46

1. Introducción

Con el paso del tiempo, el robo en viviendas es cada vez más común, y Valencia es la tercera ciudad en la que más casas se fuerzan, causando unas pérdidas aproximadas de 1.500€ por robo de media. Los meses de verano son los meses de mayor riesgo para los sufrir un robo, debido a que casi todo el mundo suele irse de vacaciones en estos meses.

Aunque no hay una fórmula mágica para evitar ser víctima de un robo, sí que se pueden tomar medidas para prevenir y disuadir estos hechos. Una medida para evitar ser robados podría ser no dejar las persianas completamente cerradas, para sí no anunciar que no vamos a estar en casa durante un tiempo. Otra medida, y tal vez la mejor para disuadir los efectos ante un posible robo sería la colocación de una alarma antirrobo, cuyas ventajas son la seguridad, tranquilidad y la protección. La seguridad de nuestros enseres es un factor muy importante a la hora de decidir si es conveniente instalar una alarma antirrobo, pues así conseguiremos que estén más protegidos debido a que, en el caso de que se produjese un robo en nuestro domicilio, habría más posibilidades de que los intrusos abandonen nuestro hogar cuando ésta comience a sonar y alertar a los vecinos, consiguiendo minimizar los daños ocasionados en nuestra vivienda. Gracias a esto, ganaremos en tranquilidad a la hora de abandonar nuestra casa sabiendo que hemos hecho lo posible para evitar un robo.

Pero el problema principal de instalar una alarma suele ser su elevado coste, pues la mayoría de las empresas especializadas en seguridad nos cobrarán una cuota mensual. Por ello, en este trabajo se van a detallar los pasos para fabricar una alarma antirrobo casera, con un bajo coste y con conocimientos básicos de desarrollo de aplicaciones Android para gestionar nuestra alarma casera y Python, para programar su comportamiento.

1.1. Motivación

El motivo principal para la realización de este TFG ha sido poder aplicar los conocimientos adquiridos durante la carrera de Ingeniería Informática a la hora de programar aplicaciones móviles utilizando el lenguaje Java, y también poder configurar sistemas basados en microprocesadores de bajo coste para que se comporten como nosotros deseamos, en este caso se ha hecho uso de una Raspberry Pi.

Además de ello, otro factor muy influyente para la elección de este trabajo ha sido el poder construir una alarma antirrobo capaz de ponerle más difícil las cosas a un ladrón, con una inversión mínima y con conocimientos de Android y Python.

También, familiarizarse con el lenguaje de programación Python que, aunque durante el grado no se haya hecho hincapié en él, partiendo de la base de otros lenguajes de programación es muy fácil e intuitivo aprender a programar en Python.

Por último, este trabajo va a aportar al autor experiencia a la hora de desarrollar aplicaciones Android utilizando el entorno de programación de Android Studio y a su vez poner en práctica los conocimientos adquiridos en Java en un proyecto real.

1.2. Objetivos

Los objetivos principales del proyecto son:

- No sobrepasar el presupuesto de 90€.
- Implementar una alarma casera de forma que cumpla todas las funcionalidades que se describen dentro del resumen de este trabajo.
- Integrar todos los dispositivos de entrada con nuestra Raspberry Pi.
- Integrar todos los dispositivos de salida con nuestra Raspberry Pi.
- Conectar nuestro teléfono móvil con nuestro controlador para conseguir administrar nuestra alarma
- Integrar un sistema de encendido y apagado manual alternativo al teléfono móvil.
- Conseguir que cualquier usuario de nuestra alarma se identifique previamente a cualquier acción que se quiera realizar sobre ella.
- Ser capaces de grabar imágenes cuando la alarma esté activada.
- Conseguir una solución mucho más económica frente a las soluciones comerciales que existen en la actualidad.

1.3. Estructura

Este documento está formado por 11 bloques:

1. Introducción: En este bloque se planteará el problema, además de explicar qué objetivos queremos lograr y qué nos ha motivado a la realización de este trabajo.
2. Estado del arte: Introducir el contexto actual de soluciones parecidas a la nuestra y sus principales ventajas y desventajas.
3. Hardware utilizado: Se realizará una explicación de los diferentes elementos de hardware que vamos a utilizar en nuestro sistema de alarma casera, explicando su funcionamiento y el por qué han sido elegidos estos y no otros para nuestra solución.
4. Software utilizado: Al igual que en el apartado anterior, introducir y explicar qué servicios y programas hemos utilizado para realizar nuestro TFG, con una explicación de por y para qué los hemos utilizado.
5. Diseño y construcción del circuito: Explicaremos el esquema de nuestro circuito para entender por qué ha sido diseñado de esta forma y las diferentes utilidades que tienen los elementos dentro del circuito
6. Desarrollo del programa controlador: La idea principal de este punto es comentar el funcionamiento esperado de nuestro controlador de la alarma, exponiendo qué es lo que hace para que cualquier persona, sin necesidad de entender el código, entienda qué hace nuestra alarma.
7. Desarrollo de la aplicación Android: En este punto explicaremos los distintos pasos que hemos seguido para desarrollar la aplicación móvil para dispositivos que tengan sistema operativo Android.
8. Pruebas: En este apartado mostraremos una serie de pruebas que hemos realizado para comprobar que nuestro sistema se comporta correctamente.
9. Posibles mejoras futuras: Introduciremos una serie de mejoras para mejorar la funcionalidad de nuestra alarma casera, que por motivos de tiempo o complejidad no hemos implementado dentro de nuestro trabajo.

10. Conclusiones: Apartado en el que se expondrán las conclusiones obtenidas durante la realización de este proyecto, comprobando si hemos conseguido los objetivos mencionados en el apartado de introducción.
11. Bibliografía: En este apartado pondremos los enlaces de interés que se han utilizado para realizar este trabajo, mostrando el título de la página, su URL y cuando fue visitada por última vez.

Finalmente, habrá un glosario en el que definiremos aquellas palabras que aparecen a lo largo del trabajo y que puedan ser complicadas de entender para aquellas personas que no estén familiarizadas con el ámbito de nuestro proyecto.



2. Estado del arte

En este apartado se va a presentar el Estado del Arte de nuestro proyecto pues, en la actualidad, existen multitud de soluciones para ponerle las cosas más difíciles a un posible ladrón, entre las que podemos distinguir principalmente dos alarmas comerciales con dispositivos dependientes y alarmas comerciales con dispositivos independientes de una empresa de seguridad.

2.1. Alarmas comerciales con dispositivos dependientes

Este tipo de alarmas están conectadas las 24h del día a internet y avisan a una central receptora, propiedad de la empresa de seguridad, de que se ha producido una situación indeseada. Una vez en la central receptora, un empleado de la empresa comprueba que no haya sido una falsa alarma y, en caso de que no lo sea, procederá a llamar a las autoridades pertinentes, ya sea bomberos, policía o el contacto de emergencia que nosotros hayamos elegido. Además de contactar con la central receptora, estos dispositivos también pueden activar alarmas acústicas.

La principal ventaja de implantar esta solución es su fiabilidad, pues siempre habrá un empleado de la empresa de seguridad que, cuando se produzca la activación de la alarma, va a estar ahí para distinguir entre falsos positivos o no, además de comunicar la situación a las autoridades pertinentes.

No obstante, su principal desventaja es el precio, ya que tendremos que pagar los dispositivos que queramos implantar en nuestro hogar deberemos pagar su instalación, pues debe ser realizada por un técnico especializado de la empresa de seguridad. También debemos pagar una cuota mensual para poder utilizar sus servicios. Otra desventaja importante de este tipo de solución es que, cuando queramos dar por finalizada la suscripción con la empresa de seguridad, los dispositivos que compramos quedarán inservibles.

El precio aproximado de la instalación de los dispositivos ronda los 200€ y las cuotas mensuales los 30€, sin contar el precio de los dispositivos que queramos instalar.

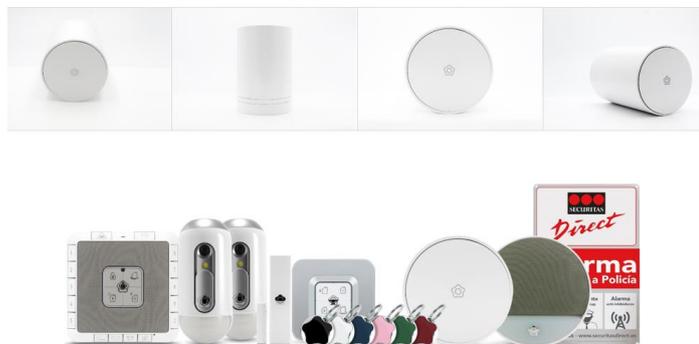


Figura 1. Dispositivos ofrecidos por la empresa Securitas Direct

Fuente: [zerovision_securitas-direct 1.png](#)

2.2. Alarmas comerciales con dispositivos independientes

Este tipo de alarma normalmente poseen un dispositivo que produce un aviso acústico y es ser muy efectivo ya que es capaz de actuar con rapidez y disminuir la posibilidad de sufrir un robo o, en caso de que se produzca, minimizar los daños ocasionados por él.

La principal ventaja de este tipo de alarmas es que tan solo tendremos que realizar la compra de ella para utilizarla, no deberemos pagar una cuota mensual, como si ocurría en las alarmas dependientes de la empresa de seguridad. Además, su instalación es muy sencilla y es muy intuitiva, por lo que no tendremos que contratar a un especialista para que realice la instalación, podremos hacerlo nosotros mismos.

Por otro lado, existe la posibilidad de que de falsos positivos y como no hay nadie vigilando que situaciones son reales o no, podrá dar algún falso positivo que alertará a nuestros vecinos cuando en realidad no está ocurriendo nada en casa.

Hay multitud de tipos de alarmas independientes de una central receptora, pero casi todas disponen de una unidad de control, un teclado de armado/desarmado, una sirena y sensor para detectar la intrusión en nuestro hogar.

El precio de este tipo de alarma comienza en 165€, como por ejemplo la alarma Chuango G5 o la alarma Samsung SmartThings, que ronda los 250€.



Figura 2. Alarma Chuango G5

Fuente: [chuango-g5.jpg](#)

3. Hardware utilizado

Para la implantación de nuestra alarma casera hemos hecho uso de los siguientes elementos hardware:

- Raspberry Pi
- Detector de movimiento
- Zumbador de advertencia
- Webcam Logitech C270
- Pulsador de emergencia

3.1. Raspberry Pi

Raspberry Pi es un microprocesador de bajo coste, con un tamaño aproximado al de una tarjeta de crédito y que comenzó a comercializarse en el Reino Unido por la fundación Raspberry Pi, perteneciente a la Universidad de Cambridge, en el año 2012 para ayudar a la enseñanza de la informática en las escuelas.

Para la realización del TFG ha sido de gran ayuda la utilización de los pines de entrada y salida que contiene esta placa, para poder conectar los dispositivos de nuestra alarma, además de utilizarla para lanzar el código desarrollado en Python para controlar la alarma. El precio de la compra fue de 37'99€.

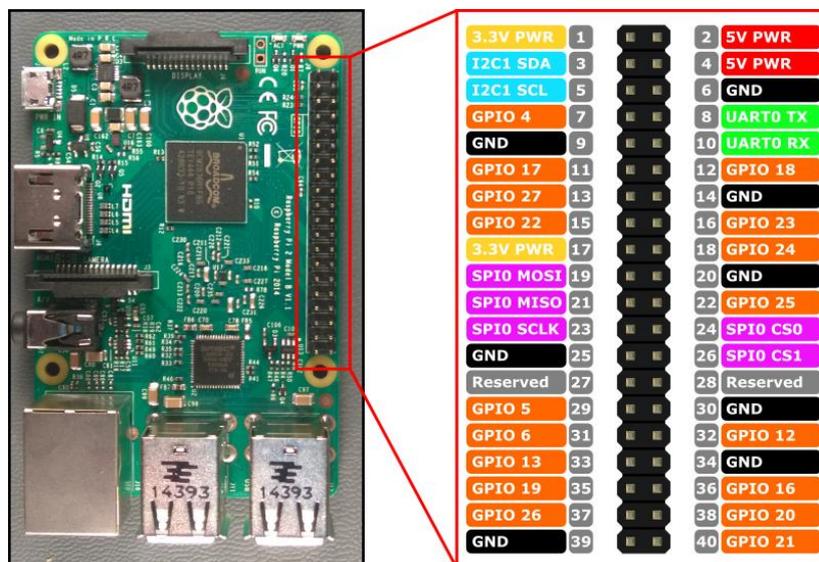


Figura 3. Raspberry Pi con sus pines de entrada y salida

Fuente: <https://www.prometec.net/usando-los-gpio-con-python/>

3.2. Detector de movimiento

Un detector de movimiento o de presencia es un dispositivo electrónico que responden al movimiento físico y pueden ser activos si inyectan luz, sonido o microondas en el ambiente y detectan si existen cambios en él, o pasivos si utilizan detección de ondas infrarrojas.

Un detector PIR o pasivo infrarrojo reacciona ante los cambios de energía o radiaciones infrarrojas de la zona de cobertura. Para poder detectar los cambios de las radiaciones infrarrojas se dividen en dos mitades, dando señal de que hay una variación cuando la radiación infrarroja en ambas partes no es igual. La principal ventaja de utilizar estos sensores es que son muy fiables y además son baratos, por lo que suele ser muy común la utilización de estos sensores. Se puede ajustar su sensibilidad (de 3 a 7m de distancia) y el tiempo de aviso (de 5 a 300s).

Para este proyecto, se ha hecho uso de un sensor de movimiento pasivo infrarrojo, además de por su reducido precio porque no emiten señal al entorno, además de ser muy fácil de configurar y por la cantidad de información que podemos encontrar acerca de ellos en internet. Su compra fue de 3'49€.



Figura 4. Sensor PIR

Fuente: <http://tdrobotica.co/sensor-de-movimiento-pir/155.html>

3.3. Zumbador de advertencia

Un zumbador es un transductor electroacústico, es decir, un dispositivo que transforma la electricidad en sonido o viceversa. En este caso, cuando se le aplica tensión, emite un sonido continuo o intermitente de un tono generalmente agudo y muy fuerte. Comúnmente es utilizado como mecanismo de aviso, como por ejemplo en los despertadores.

Existen dos tipos de zumbadores, los zumbadores activos y los pasivos. La diferencia entre ambos es que para hacer funcionar al zumbador activo solo necesitas suministrarle corriente, mientras que al zumbador pasivo además debemos suministrarle una onda de la frecuencia (para variar su tono y su intensidad).

Para esta alarma, no precisamos cambiar ni el tono ni la intensidad de nuestro zumbador, ya que con que sea agudo e intermitente nos es suficiente para alertar de que nuestro hogar está en peligro. Además, los zumbadores activos son más sencillos de conectar y controlar, pues no necesitamos generar la onda eléctrica que se convertirá en sonido. Por ello, hemos optado por un zumbador activo. Su precio fue de 3'99€.



Figura 5. Zumbador de advertencia

Fuente: https://images-na.ssl-images-amazon.com/images/I/51-OILfYGsL_SL1000.jpg

3.4. Webcam Logitech C270

Para la grabación de imágenes hemos optado por la webcam Logitech C270, que es capaz de grabar vídeos a 720p/30fps y tiene un campo visual de 60°, por lo que obtendremos una calidad de grabación suficiente para detectar qué ha pasado en nuestro domicilio. Cuando nuestro sensor de movimiento, mencionado anteriormente, detecta que hay movimiento en su campo de visión, lanza la orden a esta webcam de que debe grabar un vídeo y posteriormente guardarlo en nuestra Raspberry Pi, para así poder acceder a ellas cuando precisemos o cuando queramos acudir a las autoridades para interponer una denuncia. Su precio es de 24'99€, aunque no he precisado comprarla pues he reciclado una que tenía por casa.



Figura 6. Cámara Logitech C270

Fuente: <https://www.logitech.com/es-es/product/hd-webcam-c270>

3.5. Pulsador de emergencia

Un pulsador es un dispositivo que cuando se pulsa permite el paso de la corriente eléctrica y, cuando se deja de oprimir, la interrumpe. Es muy común verlos en timbres o en ascensores, aunque también pueden utilizarse como medidas de seguridad para detener maquinaria en caso de emergencia, aunque en este caso su funcionamiento es justo el contrario, pues deja pasar la corriente a menos que apretemos el pulsador.

Toda alarma debe tener una forma alternativa de ser desconectada para evitar que suene cuando el primer método, ya que si no no habría forma de desconectarla. Por ello, para este trabajo hemos utilizado un pulsador que estará oculto en algún lugar que solo nosotros conozcamos para desarmar la alarma en caso de que no dispongamos de conexión a Internet en nuestro dispositivo móvil, no tengamos acceso a él, o en cualquier otra situación indeseada, para evitar que la alarma esté sonando cuando nosotros entremos a casa y no haya forma de desconectarla.

Su precio de compra fue de 3'99€.



Figura 7. Pulsador de emergencia

Fuente: [Pulsador emergencia](#)

3.6. Protoboard

Una Protoboard o placa de pruebas es un tablero con agujeros interconectados siguiendo un patrón y que se utiliza para armar circuitos electrónicos. Su compra fue de 4'99€

El Protoboard se divide en 3 regiones:

- Canal central: Localizada en el centro del Protoboard. Es utilizada para colocar circuitos integrados, en caso de ser necesario (A)
- Buses: En los extremos, simbolizados con color rojo y azul para conectar ahí los polos positivos y negativos de la fuente de alimentación, respectivamente, y conectar ahí los dispositivos que queramos utilizar dentro de nuestro circuito electrónico (B).
- Pistas: Localizadas en la parte central del Protoboard, internamente conectadas para según las líneas rosas mostradas en la siguiente figura (C).

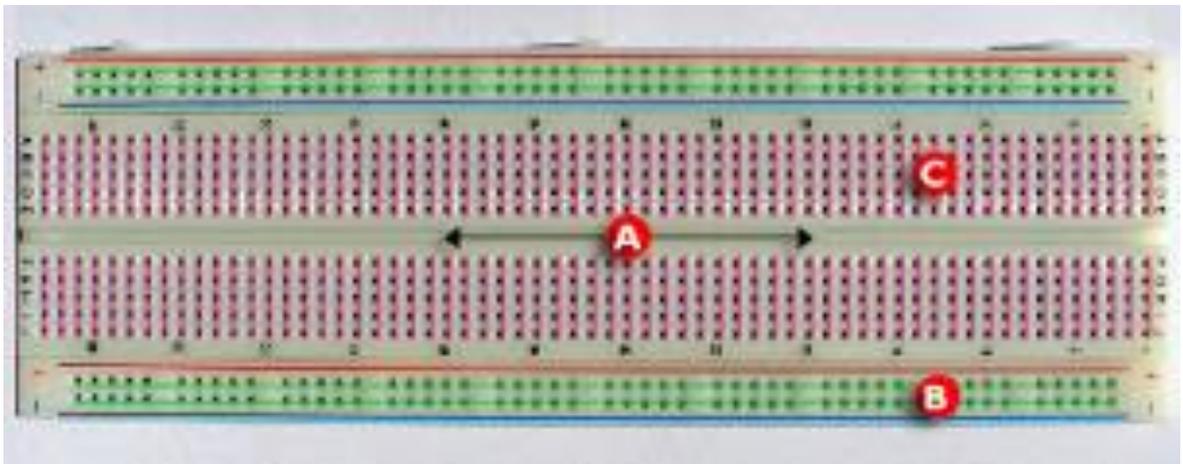


Figura 8. Estructura de una protoboard

Fuente: <http://www.circuitoselectronicos.org/2007/10/el-protoboard-tableta-de-experimentacin.html>

4. Software utilizado

Para la realización de este TFG hemos hecho uso de una serie de productos software, que nos han ayudado a hacer que nuestra aplicación móvil sea segura y robusta.

4.1. Sistema operativo

El software fundamental para hacer funcionar nuestra Raspberry Pi. Hay multitud de sistemas operativos para nuestra Raspberry Pi, pero hemos decidido utilizar Raspbian. Raspbian es una distribución del sistema operativo GNU/LINUX y que desde 2015, el fabricante de nuestra placa de bajo coste proporciona de forma oficial, por lo que considero que es un sistema operativo potente, además de estar respaldado por los propios fabricantes de nuestra placa, además de estar altamente optimizado para ella.

Muy intuitivo para utilizar si nunca has trabajado en este entorno, además de incluir un entorno de programación para Python (entre otros lenguajes de programación, como Java), que nos ayudará a la hora de escribir nuestro programa Python.

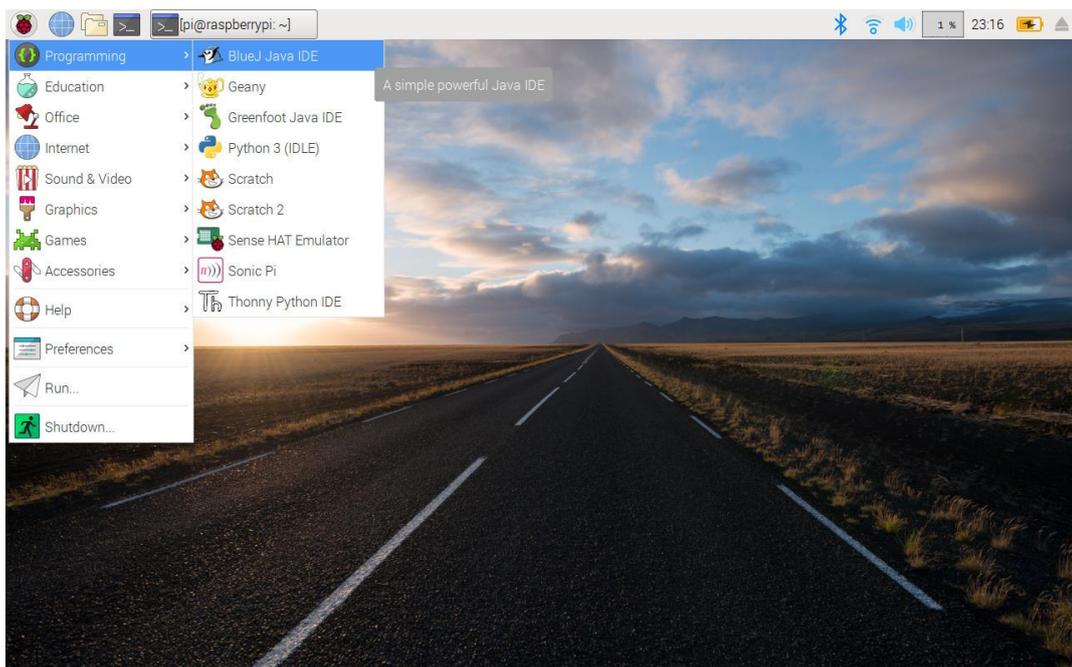


Figura 9. Interfaz de Raspbian en nuestra Raspberry Pi

4.2. Firebase

Firebase es la plataforma, perteneciente a Google, que ayuda a desarrollar aplicaciones de calidad con rapidez en entornos móviles o web. Para este trabajo, nos ha servido a la perfección porque hemos utilizado 3 productos de esta plataforma que, a pesar de ser diferentes, se integran a la perfección y se complementan los unos a los otros. Como medida de seguridad, todos los servicios de esta plataforma encriptan los datos en tránsito con HTTPS y registra el acceso de sus empleados a todos los sistemas donde se contienen datos personales, con el fin de evitar que puedan vender nuestros datos con fines comerciales.

4.2.1. Firebase Realtime Database

Firebase Realtime Database es una base de datos NoSQL alojada en la nube que almacena datos en formato JSON. Sus datos, cuando son cambiados, se reflejan en tiempo real en las aplicaciones cliente de los demás usuarios de esta base de datos y, en caso de no tener conexión a internet, se guardarán los cambios de forma local y se actualizarán para todos los usuarios de la aplicación una vez tengamos conexión a internet.

Este producto protege los datos utilizando reglas basadas en expresiones que se ejecutan antes de la lectura o la escritura de datos. Podemos utilizar diferentes reglas para diferentes usuarios, pero como en esta aplicación los datos son comunes para todos ellos, hemos puesto como reglas que solamente aquel usuario que se haya identificado previamente podrá leer o escribir sobre los datos de la base de datos.

El motivo principal para utilizar este servicio ha sido el de poder comunicar a la Raspberry Pi y a nuestra aplicación Android de forma óptima, haciendo que los cambios que detecte el sensor de movimiento se notifiquen a nuestra base de datos y, casi al instante, nos avise de que se está produciendo una situación anómala. En la comunicación en el otro sentido, podemos indicar que queremos armar/desarmar la alarma de manera rápida y sencilla.

```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

Figura 10. Reglas que protegen el acceso a los datos de la aplicación

4.2.2. Firebase Authentication

Para asegurar una aplicación, a menudo es necesario implementar algún mecanismo para identificar a los usuarios, para evitar que alguien no deseado pueda corromper nuestros datos.

Este producto nos ofrece autenticación mediante correo electrónico y contraseña, número de teléfono, redes sociales, etc. Para el trabajo desarrollado, nos hemos centrado en la autenticación por correo electrónico y contraseña pues resulta muy fácil añadir nuevos usuarios a través de la

consola de Firebase, y también nos proporciona mecanismos para la recuperación de la contraseña en caso de que la hayamos olvidado.

Como medidas de seguridad, Firebase Authentication usa los datos para habilitar la autenticación y facilitar su administración, además de guardar direcciones IP para prevenir usurpaciones de identidad.

4.2.3. Firebase Cloud Messaging

Firebase Cloud Messaging es un producto útil para enviar mensajes de forma fiable y segura. Se pueden enviar mensajes de notificación o mensajes de datos, aunque para este trabajo hemos utilizado solamente los mensajes de notificaciones, para advertir a los usuarios que hayan iniciado sesión que la alarma está sonado o que ha surgido alguna situación anómala.

Existen 3 formas de enviar mensajes: a dispositivos individuales, a grupos de dispositivos o a temas. Para nuestra aplicación, tan solo hemos utilizado las notificaciones de dispositivos suscritos a temas, en la que los usuarios de la aplicación se suscribían cuando iniciaban sesión en la aplicación y se daban de baja cuando finalizaban la sesión, siendo así notificados únicamente cuando poseen una sesión abierta, para que así no se envíen notificaciones a los clientes cuando no lo deseen.

4.3. No-IP

No-IP es un servicio de DNS dinámico que actualiza en tiempo real los nombres de dominio dentro de un servidor de nombres. Su uso más común es la asignación de un nombre de dominio a un dispositivo con IP dinámica, como es el caso de nuestra Raspberry Pi, por ello hemos utilizado este servicio para poder conectarnos con ella independientemente de que su dirección IP cambie, pues ahora posee un nombre de dominio que nunca va a cambiar. De esta forma podremos ver y descargar los vídeos grabados por nuestra webcam sin necesidad de saber la dirección IP que posea nuestro dispositivo a cada instante.

Para que este nombre de dominio sepa a qué dispositivo debe hacer referencia, se debe descargar un archivo que es muy fácil de configurar y que debe estar siempre corriendo en nuestro dispositivo, así cuando alguien haga uso de nuestro nombre de dominio se comunicará con la Raspberry Pi.

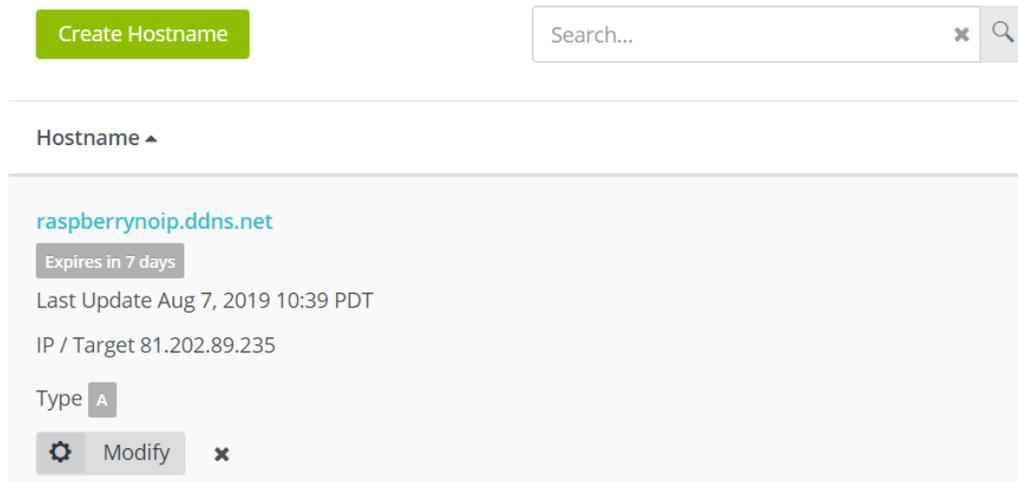


Figura 11. Panel de administración de No-IP

4.4. Java Secure Channel

Java Secure Channel (JSCh) es una implementación de Java del protocolo SSH2, cuyo propósito es acceder remotamente a un servidor a través de un canal seguro, ya que toda la información que se transmite está cifrada. Esta segunda versión, posee una serie de mejoras de seguridad respecto a su predecesora, ya que SSH2 utiliza un mecanismo de cifrado más robusto que SSH1.

Esta biblioteca JSCh ha sido utilizada para nuestra aplicación Android, desarrollada en Java, para poder acceder de forma remota a los archivos contenidos dentro de nuestro servidor FTP de forma segura y eficaz.

4.5. Motion

Es un programa que monitoriza las señales de vídeo de las cámaras del dispositivo sobre el cual sea lanzado, desde el que podemos grabar vídeos y guardarlos en la memoria internas, además de instantáneas. Esta aplicación está optimizada de forma que solo guardará las partes del vídeo en la que note diferencias pues, si entre las imágenes consecutivas no se observan variaciones, se desecharán hasta que la siguiente varíe. Esta funcionalidad es muy importante para no utilizar más memoria que la necesaria, pues los fotogramas iguales no serán útiles para identificar la persona que ha allanado nuestro hogar.

Es muy fácil e intuitivo configurar el archivo de configuración de este programa, en el que podremos elegir desde dónde queremos guardar los vídeos grabados, a qué resolución queremos que grabe la cámara, la longitud máxima del vídeo o el nombre con el que queremos guardar los vídeos grabados, entre otras.

Para lanzar el programa tan solo tendremos que lanzar el comando desde nuestro programa escrito en Python cuando el sensor de movimiento ha detectado actividad dentro de su rango y finalizándolo cuando la alarma haya dejado de sonar, ya sea por desactivación manual o por

tiempo, ya que el tiempo máximo de aviso de las alarmas es de 3 minutos en algunas ciudades como Madrid, Barcelona, Sevilla, Zaragoza o Bilbao.

4.6. Python

Con Python, conectaremos nuestra alarma con nuestra base en Firebase para recibir periódicamente el estado de ella. También se encargará de controlar los diferentes pines de entrada y salida para conseguir que suene cuando nuestro sensor de movimiento ha detectado una presencia.

Es un lenguaje de programación que tiene por objetivo conseguir un código legible y es multiplataforma, es decir, que el código que hemos desarrollado sobre un sistema operativo es perfectamente funcional en otro totalmente diferente.

4.7. Android Studio

Es el entorno de desarrollo de aplicación oficial para el entorno Android. Fue anunciado el 16 de mayo de 2013 y sustituyó a Eclipse como entorno de desarrollo oficial para aplicaciones Android. Está disponible para Windows, MacOS y Linux.

Desde este entorno, podemos tanto programar el código de nuestra aplicación Android como diseñar su interfaz, y nos proporcionará multitud de herramientas para depurar nuestro código y poder encontrar fácilmente los errores de programación que hayamos podido tener. También existe la opción de descargar un emulador de Android para lanzar sobre él las aplicaciones para probar su funcionalidad, etc.

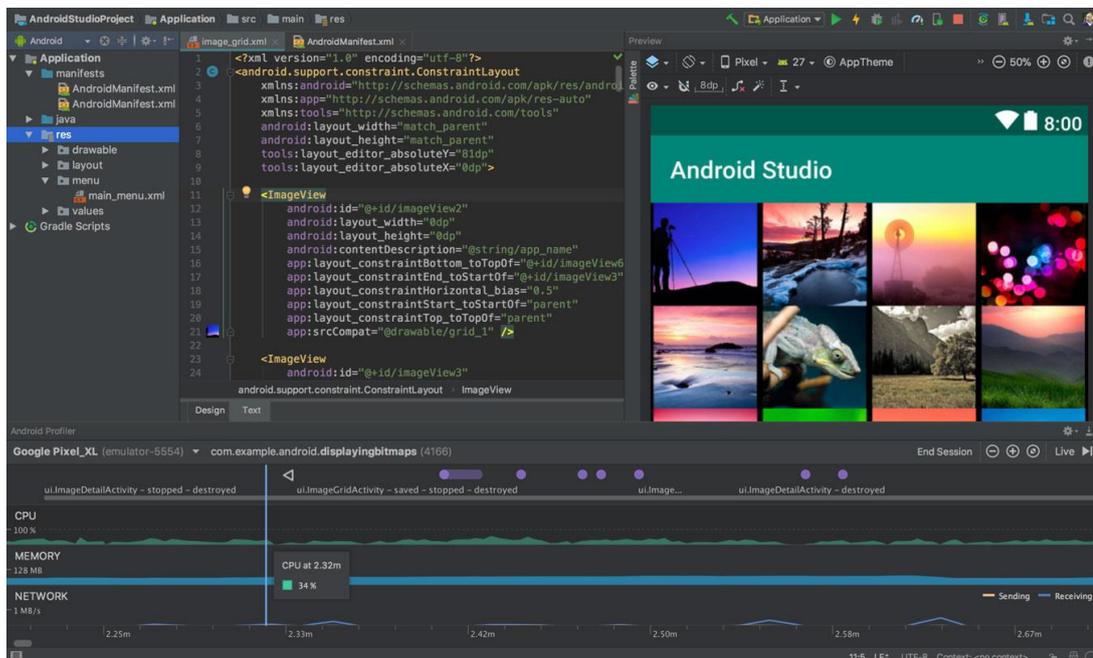


Figura 12. Interfaz de Android Studio

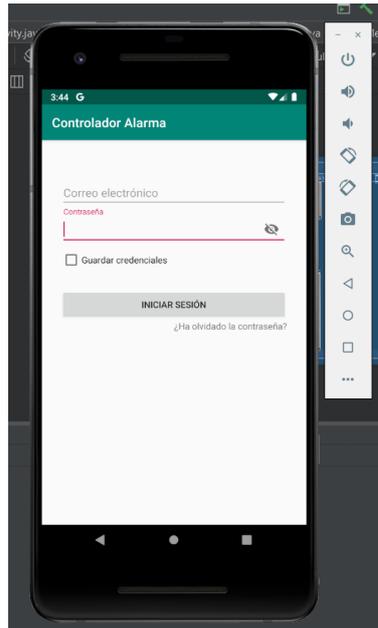


Figura 13. Emulador de Android Studio

5. Diseño y construcción del circuito

Para diseñar el circuito de nuestra alarma hemos hecho uso de una Protoboard, una placa que posee una serie de agujeros conectados entre sí y que sirve para conectar componentes electrónicos y cables para el armado de circuitos eléctricos además de una Raspberry Pi, con los pines que posee. Estos 40 pines, se distribuyen de la siguiente manera:

- 2 pines de 5V (pines 2 y 4)
- 2 pines de 3.3V (pines 1 y 17)
- 8 pines de toma de tierra (GND)
- 26 pines para dispositivos de entrada y salida
- 2 pines reservados para uso avanzado



Figura 14. Foto pines entrada y salida de Raspberry Pi

Fuente: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>

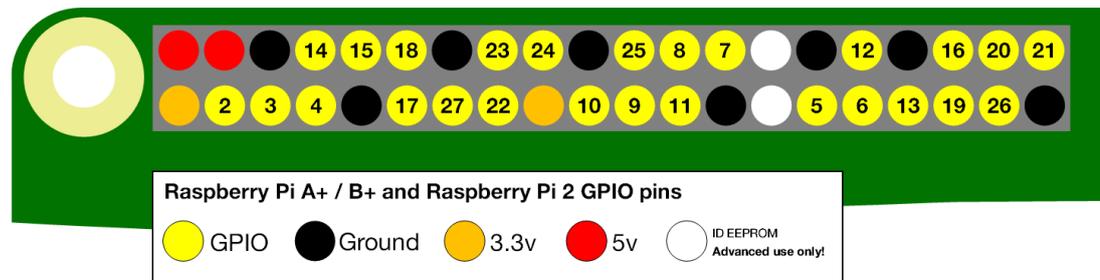


Figura 15. Correlación pines de entrada y salida Raspberry Pi

Fuente: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>

Para diseñar este circuito, primero debemos saber qué vamos a necesitar. En primer lugar, necesitamos un dispositivo de entrada que nos informe de que se ha producido una intrusión en nuestro hogar y, tras investigar sobre los diferentes dispositivos que nos ofrecían esta funcionalidad, hemos elegido un sensor de movimiento pasivo infrarrojo y posee 3 pines: el pin de 5V, el pin de toma de tierra y el pin de salida que será 1 cuando detecte movimiento y 0 cuando no lo haga. Por ello, debemos conectar este dispositivo con 5V, 0V y a cualquiera de los pines de nuestra Raspberry Pi destinados a la entrada y salida, en este caso el número 17 en la figura anterior. Este pin lo configuraremos como pin de entrada, pues en él consultaremos mientras la alarma está activada si se ha producido movimiento o no.

Tras haber conectado este dispositivo, debemos añadir un dispositivo que alerte de que se está produciendo un robo dentro de nuestro hogar, por lo que hemos decidido comprar e instalar un zumbador, el cual produce un sonido alto y estridente. Su conexión no tiene ninguna complicación, tan solo conectaremos su polo positivo a un pin de la Raspberry que no esté siendo usado y el polo negativo a un pin de toma de tierra. En este caso hemos utilizado el pin 18. Este pin será configurado como pin de salida pues, dependiendo de lo que precisemos, le indicaremos que el zumbador debe o no sonar.

Para poder grabar imágenes de lo que está ocurriendo en nuestro hogar vamos a necesitar una cámara y, tras investigar sobre el módulo de cámara de Raspberry Pi cuyo precio ronda los 10€, nos hemos decidido por una webcam que tenía por casa, para así ahorrar costes obteniendo los mismos resultados. La conexión de esta webcam es vía USB, por lo que es muy fácil añadirla a nuestro circuito. Cuando queramos iniciar a grabar imágenes, debemos arrancar el programa Motion por línea de comandos, al igual que para parar este programa.

Por último, debemos incorporar un dispositivo que sirva para desactivar la alarma de forma manual, y hemos decidido utilizar un pulsador muy pequeño, para poder ocultarlo y que solo las personas que nosotros queramos conocerán su ubicación y puedan desactivar la alarma de forma manual. Para este trabajo se ha decidido hacer uso de un pulsador porque no mantiene su estado, por lo que si desconectamos la alarma de forma manual podremos activarla posteriormente desde nuestro teléfono móvil. El pulsador está formado por 4 patillas conectadas entre sí 2 a 2 y mientras se está pulsando el pulsador, conecta estas 2 partes dejando pasar la corriente.

Tras conectar el pulsador nos dimos cuenta de que necesitábamos una resistencia para que, cuando el pulsador estuviese abierto, no nos diese un valor indeterminado, que se traducía como 0 o 1 de forma aleatoria. Por ello, se ha conectado cada una de las zonas de nuestro pulsador a la toma de 5V y a un lado de la resistencia, y al otro lado de la resistencia la conexión de toma de tierra (0V). La salida de este circuito irá conectado al pin 27 de nuestra Raspberry Pi y cuando en este pin se detecte valor 1, se desarmará la alarma en caso de que esté previamente activada, si no se realizará ninguna acción. A continuación, se muestra una imagen que muestra el esquema elegido para conectar nuestro pulsador.

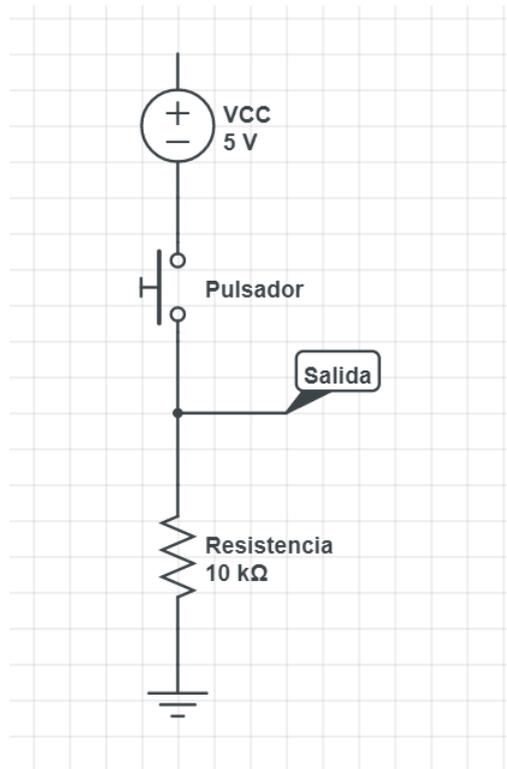


Figura 16. Esquema eléctrico del pulsador

Una vez hemos diseñado el circuito debemos construirlo, para ello necesitaremos cableado y una placa protoboard, utilizada para montar circuitos. Tras conectar todos los dispositivos con nuestra Raspberry Pi, decidimos fabricar una caja para introducir todo el circuito dentro y poder ponerlo en el sitio que deseamos. Tras haber montado la caja y metido todo dentro de ella, nuestra alarma casera quedó lista para instalar en el lugar donde nosotros queramos.



Figura 17. Parte interior de nuestra alarma



Figura 18. Parte exterior de nuestra alarma

En la primera imagen podemos observar cómo será nuestra alarma cuando la tapa de nuestra caja está abierta. Se puede ver a la izquierda de la imagen nuestra Raspberry Pi con sus pines de entrada y salida. También podemos observar la protoboard para ayudarnos a conectar los diferentes dispositivos con la Raspberry, como el zumbador y el sensor de movimiento.

Por otro lado, en la segunda foto se puede observar cómo será nuestra alarma vista desde la parte exterior. En la tapa vemos la parte de plástico blanco que pertenece al sensor de movimiento y que, debajo de este, hay un agujero para que el sensor de movimiento vea qué está sucediendo en la zona de acción. Además, podemos observar la webcam que se ha utilizado para la grabación de imágenes en el momento de la activación de alarma. Por último, en la parte superior de la derecha podemos encontrar el pulsador, que desactivará y activará la alarma en lugar de hacerlo desde nuestro dispositivo móvil. El pulsador está localizado en la parte alta porque la idea es colocar esta caja en la parte superior de una puerta y, de esta forma, el pulsador no será visto, y solo sabrán su ubicación aquellas personas a las que previamente se les haya informado sobre su ubicación.

6. Desarrollo del programa controlador

Para desarrollar el programa que controla el comportamiento de nuestra alarma hemos hecho uso del lenguaje de programación Python.

La idea de este programa es que compruebe continuamente el estado de nuestro dispositivo de entrada, el sensor de movimiento pasivo, y en caso de que haya detectado movimiento, comprobar que la alarma está armada para que les comunique a los dispositivos de salida de nuestra aplicación que se accionen, como por ejemplo la cámara o el zumbador. Para que el usuario tenga constancia de que la alarma ha sonado, debemos implementar algún mecanismo para poder comunicar a su teléfono móvil que se ha producido una activación de ella, por ello hemos utilizado el servicio ofrecido por Google para mensajería, llamado Firebase Cloud Messaging, descrito en el punto de software utilizado.

Además, mientras la alarma está sonando, debemos proporcionarle un mecanismo oculto de desactivación manual, que evite que la alarma esté sonando cuando no debe de forma que, al pulsar un pulsador de emergencia, se desactive la alarma y la desarme hasta que no sea armada de nuevo desde el dispositivo móvil.

Para que este programa funcione correctamente, hemos hecho uso de las API de Firebase para peticiones HTTP, utilizando el módulo *requests* de Python. De Firebase Realtime Database hemos utilizado las funciones para consultar y modificar los datos de nuestra base de datos y de Firebase Cloud Messaging la petición que permite enviar una notificación a los usuarios que tengan la aplicación instalada en su dispositivo móvil. Para poder realizar las peticiones debemos identificarnos, pues sería un grave peligro de seguridad que cualquier persona realice estas peticiones sin necesidad de identificarse, por ello hemos utilizado Firebase Authentication, introduciendo los datos sobre nuestro producto y sobre una cuenta que tiene permiso para realizar peticiones hacia nuestro proyecto.

Además, se ha hecho uso de los módulos *os*, *RPi.GPIO* y *time*. El primer módulo lo hemos utilizado para lanzar comandos a nuestra terminal para arrancar y parar el programa Motion, que realiza la grabación de imágenes de nuestra webcam, mientras que el segundo módulo nos ha servido para indicar qué pines serán de entrada y salida, consultar los valores de los pines de entrada y modificar los valores de los pines de salida. Por último, el tercer módulo se ha utilizado para esperar entre una consulta y otra sobre el estado del sensor de movimiento y para conseguir que el zumbador produzca sonido de forma intermitente, pues cada segundo se cambia el estado de él mientras nuestra alarma está sonando.

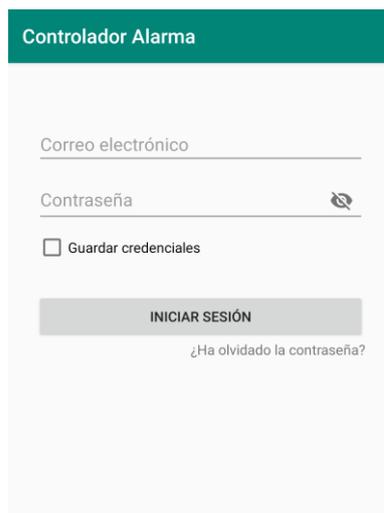
Todos estos módulos deben ser descargados a través de la línea de comandos de nuestra Raspberry Pi.

7. Desarrollo de la aplicación Android

Para completar la funcionalidad ofrecida por nuestra alarma casera, había que desarrollar una aplicación capaz de comunicarse con nuestra Raspberry Pi, y debido a que en la actualidad siempre llevamos el teléfono móvil con nosotros y la experiencia previa con el desarrollo de aplicaciones Android decidimos hacer una aplicación para dispositivos que dispongan de este sistema operativo.

Antes de desarrollar nada, pensamos en que necesitábamos que nuestra aplicación fuera capaz de hacer. Primero, nos dimos cuenta de que deberíamos tener la opción de activar y desactivar la alarma desde nuestro teléfono móvil desde cualquier lugar, para poder activarla desde cualquier lugar en caso de que se nos haya olvidado a la hora de salir de casa o para poder desactivarla mientras estamos en la puerta. Además de esto, también deberíamos poder ver todos los vídeos que nuestra webcam ha grabado cada vez que la alarma se ha puesto a sonar porque ha detectado movimiento. Además de estos requisitos funcionales, también debemos conseguir que nuestra aplicación sea segura, porque si no cualquier usuario que tenga la aplicación podrá realizar estas acciones, en caso de que nos quiten el teléfono móvil, entren a la aplicación y consigan desactivar la alarma. Por ello, hemos pensado que sería una buena solución implantar un sistema de autenticación que solo deje hacer uso de la funcionalidad de la aplicación una vez el usuario se ha identificado de forma satisfactoria. Por último, también necesitamos algún mecanismo para conseguir que el usuario de la aplicación tenga constancia de que nuestra alarma ha empezado a sonar, para que pueda realizar las acciones oportunas.

Tras especificar los requisitos que nuestra aplicación debía poseer, realizamos el diseño de esta, lo cual intentamos que fuese lo más sencilla posible ya que tan solo será una aplicación que utilicemos nosotros y aquellas personas que nosotros quisiéramos. Como la interfaz de usuario iba a ser muy sencilla, no hicimos prototipos de esta y pasamos directamente a su diseño, cuyos resultados fueron los siguientes.



Controlador Alarma

Correo electrónico

Contraseña

Guardar credenciales

INICIAR SESIÓN

¿Ha olvidado la contraseña?

Figura 19. Ventana de inicio de sesión



Figura 20. Ventana de recuperar contraseña

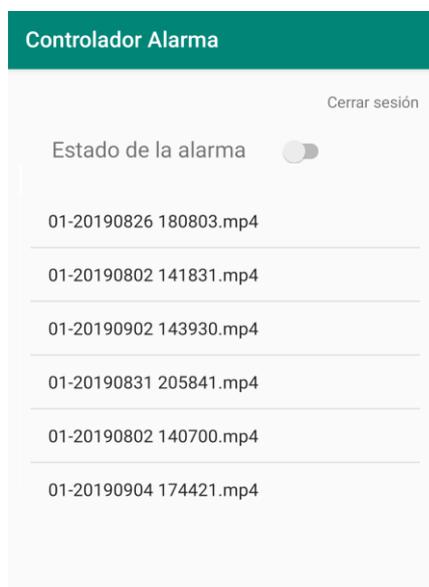


Figura 21. Ventana principal

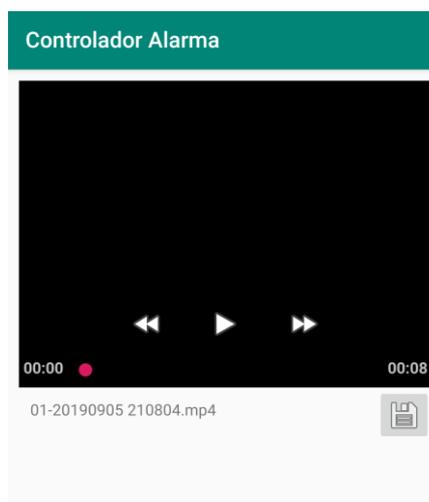


Figura 22. Ventana ver videos

Para llevar a cabo la comunicación del estado de la alarma entre nuestro dispositivo móvil nos decantamos por utilizar un servicio de Google llamado Firebase Realtime Database que es una base de datos en la nube en tiempo real, es decir, que en caso de que un usuario cambie el valor de una variable de la base de datos, inmediatamente se verá reflejado el cambio para el resto de los usuarios. Por ello, vinculamos el valor de un switch con el valor de una variable en la base de datos, de modo que siempre que se cambie el valor del switch se cambiará el valor en la base de datos o viceversa. Tras crear nuestro proyecto Firebase, debemos conectar nuestra aplicación con este proyecto siguiendo los pasos que nos indican.

Figura 23. Añadir Firebase a proyecto Android Studio

Una vez hemos conseguido conectar nuestros proyectos en Firebase y en Android Studio, tan solo tendremos que hacer uso de los métodos que este SDK nos proporciona.

El siguiente requisito es conseguir que nuestra aplicación pueda acceder a los vídeos que se han grabado desde la webcam y poder reproducirlos para ver su contenido, por ellos decidimos que nuestra Raspberry se comportaría como un servidor FTP, para poder acceder a ella desde nuestro dispositivo y acceder a la carpeta donde están guardados los vídeos y poder consultarlos y descargarlos. Para conseguir esta funcionalidad, necesitábamos una forma de abrir una conexión FTP desde nuestro teléfono móvil con la Raspberry Pi, y tras buscar posibles soluciones, nos quedamos con la librería JSCh, mencionada en el capítulo de software utilizado, que nos ayudó a abrir una conexión FTP con ella. Para conseguir esto, debemos indicarle una dirección para realizar la conexión, y como nuestra Raspberry poseía una IP dinámica, es decir, que cambiaba con el tiempo, buscamos una solución para que siempre tuviese la misma y así poder conectar con ella desde nuestra aplicación Android. Para solventar esto, optamos por utilizar el servicio No-IP, que asigna una dirección IP fija a un dispositivo que nosotros elijamos. Para que esta IP se asocie a nuestro dispositivo, debemos instalar un programa llamada *Dynamic DNS Update Client* (DUC) que sirve para detectar cuando nuestro dispositivo cambie su IP asociada y actualizar la dirección a que se hará referencia cuando introduzcamos la IP estática de este servicio. También deberemos abrir los puertos del router al que conectemos nuestra Raspberry que vayamos a redireccionar por

medio de este servicio, ya que si están cerrados surgirán problemas y la conexión FTP no se llevará a cabo.

Una vez hemos solventado el problema de la IP dinámica de nuestra Raspberry, debemos conectarla haciendo uso de la librería mencionada antes, y posteriormente conectar con la carpeta donde estaba ubicados los vídeos grabados por nuestra alarma, además de poder descargar aquellos que el usuario desee. Para llevar a cabo esta funcionalidad, hubo que desarrollar 2 nuevas clases que se ejecuten como una tarea asíncrona, ya que debe hacerse cada vez que exista la posibilidad de que esta tarea bloquee el hilo principal del interfaz de usuario, ya sea por la realización de cálculos complejos o por accesos a la red.

Por otro lado, necesitamos que el usuario tenga constancia de que se ha producido una activación de la alarma, y observando los diferentes servicios que ofrece la plataforma de Firebase vimos que existía la opción de enviar notificaciones a los dispositivos de las personas que tienen nuestra aplicación y están identificados en ella. Para ello, hicimos uso de la API HTTP de Firebase Cloud Messaging desde nuestra Raspberry, y que mandaba una notificación de que la alarma estaba sonando a todos los usuarios que tenían la aplicación excepto a aquellos que tenían la aplicación cerrada y no habían marcado la opción de recordar credenciales a la hora de identificarse. Para lanzar esta notificación, hay que indicarle quien queremos que sea el destinatario de esta, además de un título y un texto de notificación.

Por último, y por motivos de seguridad, necesitamos implantar un mecanismo de autenticación que no permitiese que cualquier persona que utilizase la aplicación pudiera acceder a la administración de nuestra alarma, y tras mirar qué soluciones podrían venirnos bien para conseguir este fin, nos dimos cuenta de que Firebase también posee un servicio que sirve justo para esto, por lo que decidimos hacer uso de él para nuestro proyecto, pues una vez hemos añadido Firebase a nuestra aplicación Android, podremos hacer uso de todos sus servicios. Esta API provee mecanismos para autenticar a los usuarios mediante usuario y contraseña, teléfono móvil, redes sociales, etc. pudiendo habilitar o deshabilitar aquellos que queramos. Para este proyecto hemos utilizado la autenticación mediante usuario y contraseña, pudiendo añadir y eliminar usuarios a nuestro gusto desde la consola de administración de Firebase.

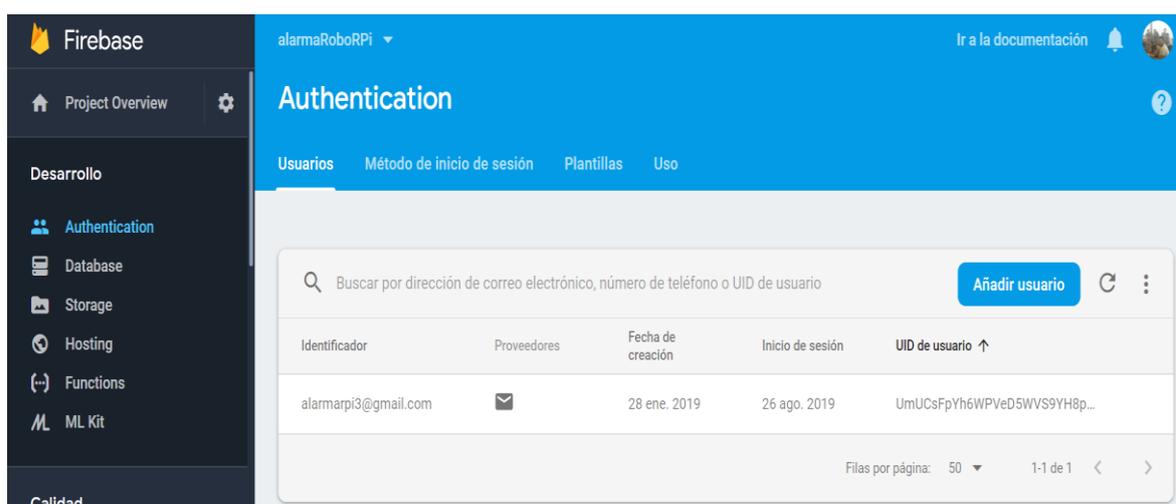


Figura 24. Consola de administración de Firebase

Aparte de la autenticación, también implementamos un mecanismo de recuperación de contraseña por si algún usuario no recordaba la suya, en la que introducía su correo electrónico y se le enviaba un correo de restablecimiento de contraseña.

8. Pruebas

Para confirmar la fiabilidad de nuestra alarma, hemos realizado una serie de pruebas para tratar de demostrar que la alarma funciona según lo esperado. Una prueba es una investigación que tiene como objetivo proporcionar información objetiva sobre la calidad del producto y que es una parte muy importante a la hora de desarrollar un producto, incluido dentro del proceso de control de calidad.

Las pruebas son simplemente una serie de actividades que intentan demostrar que el producto funciona correctamente y se pueden realizar a lo largo de todo el ciclo de vida del producto. Hay 4 tipos de prueba:

- Pruebas de unidad: Se prueba cada módulo individualmente
- Pruebas de integración: Se prueba todo el software como un conjunto para comprobar que cumple con los requisitos funcionales y de rendimiento
- Pruebas del sistema: Integrado con el resto de los elementos mecánicos para probar su funcionamiento en conjunto
- Pruebas de aceptación: El producto final es comprobado por el usuario final para determinar si es aceptado o debe ser modificado.

En este desarrollo solo vamos a realizar pruebas del sistema, pues son las pruebas que sirven para determinar que nuestra aplicación se comporta según lo esperado comunicándose con el resto de los elementos hardware, como la Raspberry Pi. Marcaremos como pasada aquellas pruebas cuyo resultado obtenido y esperado sean similares, de lo contrario dicha prueba será fallida.

Intentar iniciar sesión con credenciales incorrectas		PS-1
Descripción	En la pantalla principal de la aplicación introduciremos un correo electrónico y una contraseña que no sean correctos	
Prerrequisitos	Estar en la pantalla de inicio de nuestra aplicación Android	
Pasos	<ol style="list-style-type: none"> 1. Introducir un correo válido 2. Introducir una contraseña no válida 	
Resultado esperado	La aplicación deberá mostrar un aviso de que las credenciales son incorrectas y no iniciaremos sesión	
Resultado obtenido	Se muestra un aviso de que las credenciales no son correctas. No nos deja iniciar sesión.	
Resultado prueba	Superada	

Tabla 1. Prueba del sistema 1

Intentar iniciar sesión con credenciales correctas		PS-2
Descripción	En la pantalla principal de la aplicación introduciremos un correo electrónico y la contraseña que se corresponde con este correo electrónico	
Prerrequisitos	Estar en la pantalla de inicio de nuestra aplicación Android	
Pasos	<ol style="list-style-type: none"> 1. Introducir un correo válido 2. Introducir la contraseña para el correo introducido anteriormente 	
Resultado esperado	Iniciaremos sesión e iremos a la pantalla principal de gestión de nuestra alarma	
Resultado obtenido	Entramos a la pantalla de gestión de nuestra alarma	
Resultado prueba	Superada	

Tabla 2. Prueba del sistema 2

Armar/desarmar la alarma		PS-3
Descripción	Cambiar el estado de la alarma, si está armada desarmarla o viceversa	
Prerrequisitos	<ul style="list-style-type: none"> - Haberse identificado con un correo y contraseña correctos - Estar en la pantalla principal para administrar la alarma casera 	
Pasos	<ol style="list-style-type: none"> 1. Hacer clic en el botón de cambio de estado de la alarma 	
Resultado esperado	Se deberá cambiar el estado del botón y el estado de la alarma cambiar	
Resultado obtenido	El botón cambia al estado desarmada, tras ello hemos realizado movimiento frente al sensor y no se ha producido ningún resultado	
Resultado prueba	Superada	

Tabla 3. Prueba del sistema 3



Moverse enfrente del sensor estando la alarma armada		PS-4
Descripción	La intención de esta prueba es ver que cuando armamos la alarma y nos movemos cerca de esta, comienza a sonar y la webcam graba el vídeo pertinente	
Prerrequisitos	- Estar identificado en la aplicación	
Pasos	<ol style="list-style-type: none"> 1. Armar la alarma 2. Movernos cerca del sensor de movimiento 	
Resultado esperado	Debemos recibir una notificación informando de que la alarma está sonando y además el zumbador comenzará a sonar intermitentemente, además la webcam debe grabar un vídeo que podremos consultar posteriormente desde la aplicación.	
Resultado obtenido	El teléfono móvil ha recibido una notificación advirtiendo que la alarma está sonando. Además, el zumbador ha comenzado a sonar de forma intermitente. Tras la desactivación de la alarma hemos entrado a la aplicación y hemos visto que había un nuevo vídeo en la lista de vídeos grabados	
Resultado prueba	Superada	

Tabla 4. Prueba del sistema 4

Moverse enfrente del sensor estando la alarma desarmada		PS-5
Descripción	Cuando la alarma está desarmada no se debe realizar ninguna acción, para evitar que la alarma comience a sonar cuando no debe	
Prerrequisitos	- Estar identificado en la aplicación	
Pasos	<ol style="list-style-type: none"> 1. Desarmar la alarma 2. Movernos cerca del sensor de movimiento 	
Resultado esperado	No deberá ocurrir nada, pues la alarma está desarmada	
Resultado obtenido	No se ha producido ningún cambio	
Resultado prueba	Superada	

Tabla 5. Prueba del sistema 5

Desarmar alarma desde pulsador		PS-6
Descripción	Queremos comprobar que mientras que la alarma está sonando, si pulsamos el pulsador de emergencia (debidamente escondido para evitar que sea fácil de localizar por cualquier otra persona) la alarma dejará de sonar de inmediato.	
Prerrequisitos	- La alarma debe estar sonando	
Pasos	1. Pulsar el pulsador	
Resultado esperado	Al pulsar el pulsador, el sonido de la alarma debe de cesar, al igual que el estado de la alarma debe pasar a desarmada, por lo que no se volverá a conectar hasta que no se quiera volver a armar.	
Resultado obtenido	La alarma deja de sonar cuando se pulsa el pulsador y no se han producido sonidos mientras se realizaban movimientos frente al sensor de movimiento	
Resultado prueba	Superada	

Tabla 6. Prueba del sistema 6

La alarma se desactiva tras pasar su tiempo máximo (120s)		PS-7
Descripción	La realización de esta prueba tiene como finalidad demostrar que la alarma se desconecta sin superar el tiempo máximo de funcionamiento indicado en el diseño de nuestro programa Python, en este caso 120s	
Prerrequisitos	- La alarma debe estar sonando	
Pasos	1. Cronometrar el tiempo de sonido de nuestra alarma	
Resultado esperado	Una vez la alarma haya dejado de sonar, nuestro cronómetro nos debe indicar que el tiempo de funcionamiento ha sido cercano a 120s, pero siempre sin pasarse de ese tiempo máximo	
Resultado obtenido	Una vez ha dejado de sonar la alarma, nuestro cronómetro marcaba 116s	
Resultado prueba	Superada	

Tabla 7. Prueba del sistema 7

Ver vídeo grabado por webcam		PS-8
Descripción	Esta prueba tiene como propósito comprobar que tras grabar las imágenes y estar almacenadas en nuestra Raspberry, podemos ver el contenido de esos vídeos desde nuestro dispositivo móvil, a través de la aplicación Android desarrollada	
Prerrequisitos	<ul style="list-style-type: none"> - La webcam debe haber grabado al menos un vídeo - Estar identificado en la aplicación 	
Pasos	1. En la lista de vídeos, hacer clic a uno de ellos	
Resultado esperado	Debe aparecer una nueva ventana donde veamos el contenido del vídeo deseado	
Resultado obtenido	Podemos observar el vídeo grabado por nuestra webcam	
Resultado prueba	Superada	

Tabla 8. Prueba del sistema 8

Guardar vídeo de nuestra Raspberry en nuestro teléfono		PS-9
Descripción	Esta prueba tiene el objetivo de demostrar que podemos guardar los vídeos de nuestra Raspberry en el almacenamiento interno de nuestro teléfono móvil, para poder tenerlo en caso de necesitarlo y no tener conexión a internet	
Prerrequisitos	<ul style="list-style-type: none"> - Estar en la pantalla de visualización de un vídeo grabado 	
Pasos	<ol style="list-style-type: none"> 1. Hacer clic en el botón con el icono de guardar de nuestra aplicación 2. Ir a nuestro almacenamiento interno y comprobar que el vídeo está guardado y puede verse desde el reproductor de vídeo de nuestro dispositivo 	
Resultado esperado	Aparecerá un mensaje de información que nos indicará que el vídeo se ha guardado correctamente. Al ir a la carpeta que nos indica, deberá aparecer el vídeo y podremos reproducir su contenido	
Resultado obtenido	Aparece un mensaje diciendo que el vídeo está en el directorio raíz de nuestro dispositivo, en una carpeta llamada <i>videosAlarma</i> . Tras acceder a esa carpeta, podemos ver el vídeo que hemos guardado y podemos reproducirlo	
Resultado prueba	Superada	

Tabla 9. Prueba del sistema 9

9. Posibles mejoras futuras

Para incrementar la funcionalidad de nuestra alarma, podemos introducir una serie de mejoras que podrían implementarse en nuestro proyecto para implementar su funcionalidad y poder y hacer más robusta frente a un posible robo. Durante las distintas fases del proyecto se han ido encontrando posibles mejoras que podían incluirse en nuestro proyecto, pero debido a distintos motivos, como la falta de tiempo o el aumento de los costes no se han llevado a cabo.

Las principales mejoras que se podrían implementar dentro de este proyecto pueden ser:

- Añadir una batería capaz de alimentar nuestra Raspberry Pi y todos los dispositivos de nuestra alarma en caso de que se vaya la luz.
- Conseguir ver qué ocurre en tiempo real mientras la cámara de nuestra Raspberry Pi detecta movimiento y no solo una vez la cámara está activada que, por motivos de tiempo no hemos podido conseguir que nuestra cámara, además de grabar imágenes, las comparta vía streaming para poder verlas en tiempo real.
- Agregar módulo GSM para avisar por SMS al número o números que deseemos en caso de que la alarma sea disparada.
- Agregar teclado numérico para armar y desarmar nuestra alarma con un código de 4 cifras, para poner más difícil la administración de nuestra alarma y hacerla más segura.
- Replicar el sensor de movimiento y el zumbador para que en caso de que uno no funcione, poder hacer uso del segundo dispositivo.
- Desarrollar una aplicación para dispositivos móviles que tengan sistema operativo iOS.

10. Conclusiones

Tras llevar a cabo nuestro proyecto de fabricar una alarma para el hogar, hemos sumado el valor de todos los objetos utilizados, un total de 79'44€, por lo que hemos cumplido con el objetivo de no superar los 90€ de inversión que hemos estipulado en el punto de objetivos de nuestro trabajo.

También hemos sido capaces de dotar a nuestra alarma casera de todas las funcionalidades que se han descrito a lo largo de todo el trabajo y el resumen de este, afrontando los problemas y complicaciones intrínsecos que se pueden dar en todo tipo de desarrollo de hardware y software, como por ejemplo la configuración de Motion en nuestra Raspberry para que guardase los vídeos en la carpeta seleccionada. Este problema nos llevó mucho tiempo resolverlo debido a que no había mucha información sobre problemas con este programa.

Por otro lado, hemos conseguido integrar todos los dispositivos de entrada y salida que se han propuesto en la parte de hardware utilizado, de modo que todos funcionan bien y se comportan según lo previsto, debido a que se han realizado pruebas sobre ellos y estas han sido superadas.

Además, también hemos conseguido implantar un sistema de seguridad para evitar que toda persona ajena pueda controlar nuestra alarma y consultar los vídeos que se han grabado desde ella, haciendo así nuestra aplicación mucho más robusta.

Hemos conseguido proteger nuestro hogar de una forma mucho más económica que si hubiéramos contratado los servicios de una empresa de seguridad, evitando la cuota mensual y el precio por la instalación de los aparatos que quedarán inservibles cuando demos de baja los servicios de esta.

Este trabajo nos ha sido altamente beneficioso por la adquisición de las competencias necesarias para el desarrollo íntegro de proyectos de hardware y software. Además, hemos obtenido nuevos conocimientos sobre microprocesadores, desarrollo de aplicaciones Android y Python, para poder familiarizarnos con estas tecnologías para futuros desarrollos.

11. Bibliografía

[1] Artículo del periódico Levante EMV sobre el robo en viviendas en Valencia.

URL: <https://www.levante-emv.com/comunitat-valenciana/2018/08/12/comunitat-valenciana-registra-robo-viviendas/1755424.html>

Consultada el 14 de marzo de 2019

[2] Página oficial de Securitas Direct

URL: <https://www.securitasdirect.es/es/servicios>

Consultada el 14 de marzo de 2019

[3] Información sobre los diferentes tipos de alarmas

URL: <https://alarmasparacasas.com/hogar/>

Consultada el 16 de marzo de 2019

[4] Información sobre las alarmas independientes de una central receptora

URL: <https://alarmasparacasas.com/baratas-sin-cuotas/>

Consultada el 16 de marzo de 2019

[5] Blog de la UPV que habla sobre la Raspberry Pi

URL: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

Consultada 25 de marzo de 2019

[6] Información sobre sensores de movimiento

URL: <https://www.prometec.net/sensor-pir/>

Consultada 25 de marzo de 2019

[7] Artículo de Wikipedia sobre los zumbadores

URL: <https://es.wikipedia.org/wiki/Zumbador>

Consultada 25 de marzo de 2019

[8] Página oficial de Logitech sobre la webcam C270

URL: <https://www.logitech.com/es-es/product/hd-webcam-c270>

Consultada 25 de marzo de 2019

[9] Tecnología de los pulsadores e interruptores

URL: <https://www.abc.com.py/edicion-impres/suplementos/escolar/tecnologia-de-los-pulsadores-e-interruptores-904222.html>

Consultada 27 de marzo de 2019

[10] Información acerca de la placa de pruebas o Protoboard

URL: https://es.wikipedia.org/wiki/Placa_de_pruebas

Consultada 27 de marzo de 2019

[11] Página oficial de Google sobre la plataforma Firebase

URL: <https://firebase.google.com/support/privacy/?hl=es-419>

Consultada 2 de abril de 2019

[12] Página oficial de Google sobre el producto Firebase Realtime Database

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

Consultada 2 de abril de 2019

[13] Página oficial de Google sobre el producto Firebase Authentication

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

Consultada 2 de abril de 2019



[14] Página oficial de Google sobre el producto Firebase Cloud Messaging

URL: <https://firebase.google.com/docs/cloud-messaging>

Consultada 2 de abril de 2019

[15] DNS Dinámico (DDNS) en Wikipedia

URL: https://es.wikipedia.org/wiki/DNS_din%C3%A1mico

Consultada 5 de abril de 2019

[16] Página oficial de JCraft sobre la librería Java Secure Channel (JSch)

URL: <http://www.jcraft.com/jsch/>

Consultada 5 de abril de 2019

[17] Artículo de EL PAÍS sobre el tiempo máximo de sonido de una alarma

URL: https://elpais.com/diario/1991/03/24/madrid/669817454_850215.html

Consultada 12 de abril de 2019

[18] Página oficial del servicio Motion para configurar su servicio

URL: https://motion-project.github.io/motion_config.html

Consultada 12 de abril de 2019

[19] Proyecto de GitHub del servicio Motion

URL: <https://motion-project.github.io/>

Consultada 12 de abril de 2019

[20] Página oficial de Python

URL: <https://www.python.org/about/>

Consultada 23 de abril de 2019

[21] Página oficial Android Studio

URL: <https://developer.android.com/studio>

Consultada 23 de abril de 2019

[22] Documentación sobre los pines de entrada y salida de la Raspberry Pi

URL: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>

Consultada 15 de marzo de 2019

[23] Material sobre pruebas ofrecido en la asignatura de la carrera Mantenimiento y Evolución del Software

URL: https://poliformat.upv.es/access/content/group/GRA_11569_2018/Teor%C3%ADa%20y%20seminarios/7%20-%20Pruebas.pdf

Consultada 15 de junio de 2019

[24] ¿Qué es un Protoboard?

URL: <http://www.circuitoselectronicos.org/2007/10/el-protoboard-tableta-de-experimentacin.html>

Consultada 15 junio de 2019

[25] Raspbian en Wikipedia

URL: <https://es.wikipedia.org/wiki/Raspbian>

Consultada 15 de agosto de 2019

[26] Editor online de circuitos electrónicos

URL: <https://www.circuitlab.com/editor/>

Visitada 15 de mayo de 2019

[27] Ejecutar una tarea en un nuevo hilo con AsyncTask

URL: <ejecutar-una-tarea-en-un-nuevo-hilo-con-async-task>

Visitada 3 de mayo de 2019

Apéndice A:

Glosario

- API: Interfaz de programación de aplicaciones. Es el conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software dentro de una aplicación.
- Buses: Es un sistema digital que transfiere datos entre los componentes de un ordenador o sistema. También llamado canal.
- DDNS: Sistema de nombres de dominio dinámico. Es un servicio que permite la actualización en tiempo real de la información sobre nombres de dominio situados en un servidor de nombres.
- Dirección IP: Número que identifica a una interfaz de red de un dispositivo que esta conectado a Internet.
- FPS: Acrónimo de fotogramas por segundo. Unidad de medida utilizada para medir la velocidad en que se muestran los fotogramas de un vídeo.
- GNU/LINUX: Sistema operativo libre multiplataforma, multiusuario y multitarea. Todo su código puede ser utilizado, modificado y redistribuido libremente por cualquiera.
- GSM: Sistema global para las comunicaciones móviles. Es un sistema estándar de telefonía móvil digital.
- HTTPS: Protocolo seguro de transferencias de hipertexto. Es la versión segura del protocolo HTTP, que permite las transferencias de información en la web.
- JSON: Notación de objeto de JavaScript. Es un formato de texto sencillo para el intercambio de datos, basado en pares nombre/valor.
- Línea de comandos: Es un método que permite a los usuarios dar instrucción a algún programa informático por medio de una línea de texto simple, llamada comando.
- Microprocesador: Es el circuito integrado central más complejo de un sistema informático y es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de usuario.
- NoSQL: Es una clase de gestión de bases de datos que difieren del modelo clásico, el más destacado es que no utilizan SQL como lenguaje principal de consultas, de ahí su nombre.



- **SDK:** Kit de desarrollo de software. Es un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto.
- **SSH2:** Es un protocolo de red para operar servicios de red seguros a través de una red no segura. Su uso más común es para línea de comandos remota y ejecución de comandos remotos.
- **Streaming:** Retransmisión en directo. Consiste en obtener un contenido en el momento en el que se está produciendo.
- **URL:** Localizador de recursos uniforme. Es un identificador de recursos que pueden cambiar en el tiempo, aunque su dirección se mantenga invariable.