

IES AUGUSTÓBRIGA



---

# MÓDULO DE “PROYECTO”

---

Departamento de informática



9 DE JUNIO DE 2020

CARLOS SÁNCHEZ PRUDENCIO  
IES Augustóbriga, Navalmoral de la mata, Cáceres

Módulo de proyecto

# **AutoRoom -- Automatización de una habitación.**

**Navalmoral de la mata, Cáceres.**



**Peticionario:** departamento de informática del IES Augustóbriga

**Autor:** Carlos Sánchez Prudencio

IES Augustóbriga, Cáceres a 9 de Junio del 2020

## Índice

1. <a href="#">Introducción</a>	3
2. Análisis de requisitos	
a. Objetivos	
b. Tecnologías utilizadas	
i. Eclipse	
ii. Android Studio	
iii. Firebase	
iv. Raspberry Pi	
v. Raspbian	
vi. MariaDB	
vii. PI4J	
3. Diseño de la aplicación	
a. Diseño de la base de datos	
i. FirebaseDatabase (remota)	
ii. MariaDB (local)	
iii. Sistema de comunicación	
b. Diseño de la aplicación	
i. Desktop App	
ii. App	
4. Manuales	
5. Conclusiones	
6. Propuestas de ampliación o mejoras	
7. Bibliografía	
8. Anexos	

# 1. Introducción

AutoRoom pretende desarrollar un sistema autónomo que controle todos los aspectos de la vida cotidiana de una habitación haciendo la vida a las personas más sencilla y simple.

Estos aspectos abordan desde los controles más básicos como lo son la iluminación o la temperatura, hasta los más complejos como el control de la climatización o video vigilancia.

Al ser éste un proyecto muy concreto, en las próximas páginas se especificarán las necesidades, los inconvenientes y las medidas adoptadas para conseguir el objetivo final, la automatización de la habitación. Es por ello que el título del proyecto sea *AutoRoom*.

Este proyecto es desarrollado y puesto en marcha por el alumno autor de esta documentación. Claro que no es el único participante

## 2. Análisis de requisitos

### 2.1 Objetivos

Las metas para alcanzar son las siguientes:

- Lectura de temperatura, tanto interior como exterior, para historial
- Lectura de movimiento con historial
- Control de iluminación, automático, manual, eléctrico manual
- Control de climatización
- Tanto las lecturas como los controles deben poder hacerse en local y en remoto, desde cualquier parte del globo terráqueo.
- Alarmas de intruso y despertadores luminarios.
- Escenificación ambiental con leds
- Control de usuarios, remotos y locales.
- Video vigilancia con envío de correo si está la alarma activada
- Los usuarios remotos son aquellos que se registran únicamente a través de la web o de la app para móvil.
- Los usuarios locales son aquellos que se registran únicamente a través de la raspberry y solo con permiso del administrador.

## 2.2 Tecnologías utilizadas

### Eclipse

Es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.



Esta plataforma, típicamente ha sido usada para desarrollar IDE (Interface Development Environment) como el IDE de Java (JDK) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue relicenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL).

Su última versión, así como todos sus plugins y temas necesarios, puede ser descargada desde su página web <https://www.eclipse.org/downloads/>.

El proyecto creado será de tipo maven para poder utilizar las bibliotecas necesarias sin necesidad de descargar el .jar. Todas estarán en el archivo "pom.xml".

### Android Studio

Es el IDE oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.



Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014.<sup>2</sup>

Se puede descargar desde su página web <https://developer.android.com/studio>

Se creará un proyecto escrito en Java con la integración de AndroidX para el uso de todas las herramientas disponibles hasta la fecha.

## Firestore



Es una plataforma ubicada en la nube desarrollada por Google en 2014, integrada con Google Cloud Platform.

Los desarrolladores tendrán una serie de ventajas al usar esta plataforma:

- Sincronizar fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Usa herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Unity y C++.
- Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web.

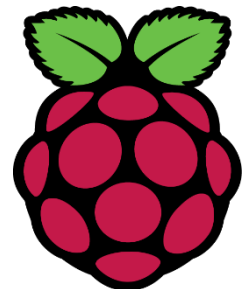
Firestore dota a sus usuarios de una gran documentación para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan activamente en plataformas como Github y StackOverflow, así como poseen un canal de Youtube explicando el funcionamiento de varias de sus herramientas. Su página oficial es <https://firebase.google.com/>.

La API se implementará en el archivo *pom.xml* de la aplicación de escritorio y a nivel de App en el archivo *build.gradle* de la aplicación para Android.

## Raspberry Pi

Es un ordenador de placa reducida (SBC) de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, con el objetivo de estimular la enseñanza de informática en las escuelas.

El software es de código abierto aunque permite usar otros sistemas operativos, incluido una versión de Windows 10. En todas sus versiones, incluye un procesador Broadcom, memoria RAM, GPU, puertos USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara.



La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian, RISC OS 5, Arch Linux ARM y Pidora y promueve principalmente el aprendizaje del lenguaje de programación Python. Otros lenguajes también soportados son Tiny BASIC, C, Perl y Ruby.

## Raspbian

Raspbian es una distribución del sistema operativo GNU/Linux basado en Debian y por lo tanto libre.

Desde 2015, la Raspberry Pi Foundation lo ha proporcionado de forma oficial como el sistema operativo primario para la familia de placas SBC de Raspberry Pi.



Es el sistema operativo (OS) que utilizaremos para usar la raspberry. Su última versión puede ser descargada desde su página web <https://www.raspberrypi.org/downloads/>

## MariaDB

Es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael Widenius fundador de MySQL, la fundación MariaDB y la comunidad de desarrolladores de software libre.



# MariaDB

Introduce dos motores de almacenamiento nuevos, uno llamado Aria que reemplaza a MyISAM y otro llamado XtraDB en sustitución de InnoDB. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Este gestor surge a raíz de la compra de Sun Microsystems compañía que había comprado previamente MySQL por parte de Oracle. MariaDB es una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con licencia GPL.

Su página oficial para la descarga y documentación es <https://mariadb.org/>.

## PI4J

Es una API desarrollada por Robert Savage y Daniel Sendula para el control de las entradas y salidas de la Raspberry.

Proporciona control sobre todos los GPIO (General Purpose Input/Output), brinda modo de salida de PWM (Power Wave Modulation) para el control del brillo de los leds, control de velocidad de motores, etc.

Los pines de la Raspberry están numerados y siguen una numeración en concreto, cada desarrollador sigue la suya.

La numeración que sigue PI4J para las entradas y salidas corresponden a los números en negro de la ilustración.

El número 7 es especial puesto que es el único canal de comunicación que soporta W1 (One Wire).

Su página oficial es <https://pi4j.com/1.2/>.

Utilizaremos esta biblioteca para tener el control sobre las entradas y salidas (GPIO) de la raspberry.

GPIO#	NAME		NAME	GPIO#
	3.3 VDC Power	1	5.0 VDC Power	2
8	GPIO 8 SDA1 (I2C)	3	5.0 VDC Power	4
9	GPIO 9 SCL1 (I2C)	5	Ground	6
7	GPIO 7 GPCLK0	7	GPIO 15 Tx0 (UART)	15
	Ground	9	GPIO 16 Rx0 (UART)	16
0	GPIO 0	11	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13	Ground	14
3	GPIO 3	15	GPIO 4	4
	3.3 VDC Power	17	GPIO 5	5
12	GPIO 12 MOSI (SPI)	19	Ground	20
13	GPIO 13 MISO (SPI)	21	GPIO 6	6
14	GPIO 14 SCLK (SPI)	23	GPIO 10 CE0 (SPI)	10
	Ground	25	GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27	SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29	Ground	30
22	GPIO 22 GPCLK2	31	GPIO 26 PWM0	26
23	GPIO 23 PWM1	33	Ground	34
24	GPIO 24 PCM_FS/PWM1	35	GPIO 27	27
25	GPIO 25	37	GPIO 28 PCM_DIN	28
	Ground	39	GPIO 29 PCM_DOUT	29

**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>



## 3. Diseño de la aplicación

### 3.1. Diseño de la base de datos

Se crearán dos tipos de bases de datos, una local y otra remota.

Para la base de datos local, se utilizará el sistema gestor **MariaDB**, basada en MySQL. Esta base de datos sólo se dedicará a la recogida de datos de los usuarios registrados en la Raspberry, es decir, los usuarios locales.

Tendrá una tabla cuyos campos, entre otros, será la contraseña, estará cifrada en md5. Dicha contraseña será utilizada para el inicio de sesión junto con el nombre de usuario.

Estos datos nunca serán guardados en Firebase y sólo podrán ser modificados, creados o borrados por el usuario administrador del sistema. El administrador es el único que puede registrar nuevos usuarios, así como hacerlos administradores también.

Para la base de datos remota, se utilizará **FirestoreDatabase**. Es una base de datos NoSql es decir que posee esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas.

Esta base de datos de Firebase se basa en el esquema *clave-valor*, posee jerarquía *padre-hijo*. Es por ello que para acceder a un valor, primero habrá que acceder a las claves de sus padres.

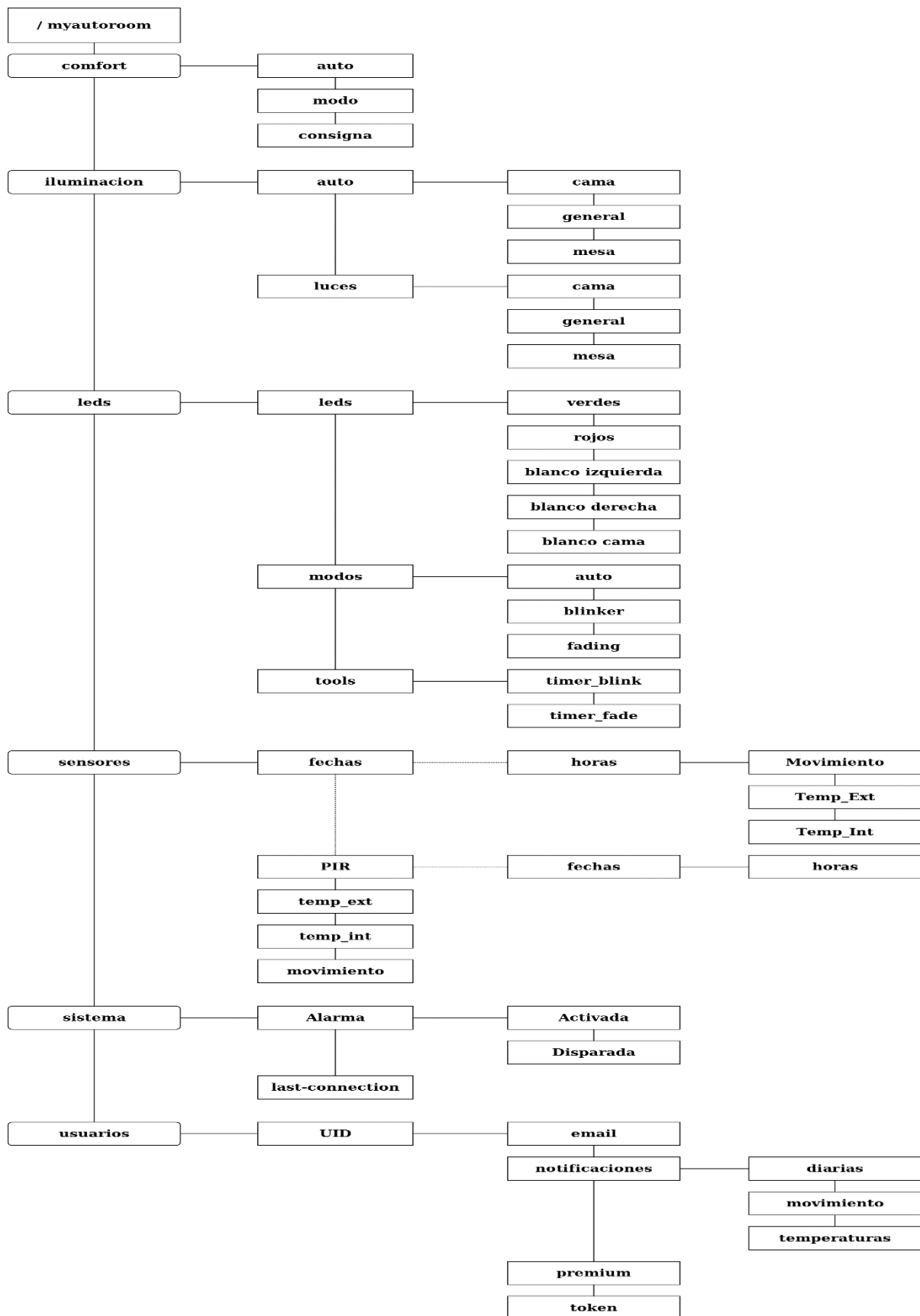
Al ser una base de datos en tiempo real no hace falta hacer ningún commit para guardar los cambios realizados. Esto presenta el problema de no poder volver atrás, es decir un RollBack, en el caso de error. El único error que puede aparecer a la hora de guardar datos, es el fallo de conexión con la base de datos.

Explicar que esta base de datos se utilizará para dos cosas importantes:

- Guardar datos: se guardarán datos relativos a los sensores de temperatura, de movimiento y de luz. Los valores de los sensores de serán guardados cada media hora de cada día. También se registrará las veces que se detecta movimiento en la habitación. También se guardarán datos relativos a los usuarios remotos.
- Hacer cambios en la Raspberry: estos cambios son los que hacen posible que desde cualquier punto del mundo se pueda encender la luz, por ejemplo. Así, habrá una ruta para cada cambio que se necesite.

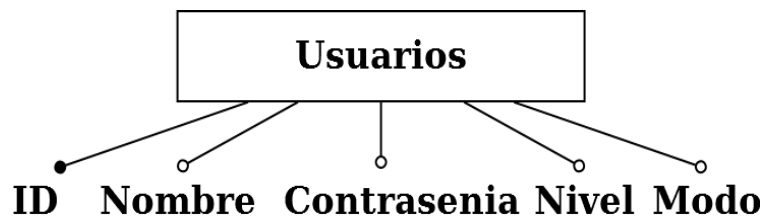
Los usuarios remotos son aquellos que se registran vía web o vía móvil. No tendrán acceso completo a los cambios necesarios, pero se guardarán datos como el email con el que se registran o el token que se genera en el móvil. Este último se guarda para poder mandar notificaciones push.

## FirestoreDataBase (remota)



## MariaDB (local)

En esta tabla se guardarán todos los usuarios que accedan a la raspberry.



ID → Es un identificador único para enumerar a los usuarios

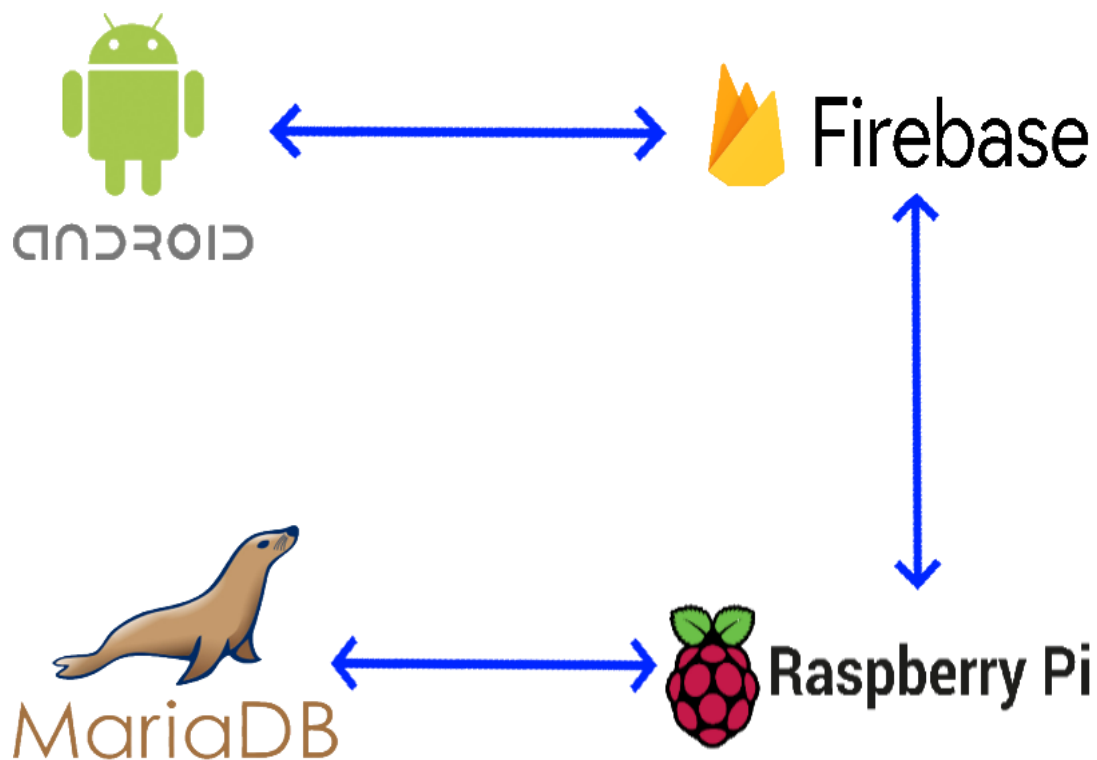
Nombre → Nombre del usuario con el que se logueará

Contraseña → Contraseña del usuario con la que iniciará sesión

Nivel → 1,2

Modo → Admin, Usuario (deprecated)

## Comunicación entre los cuatro sistemas principales

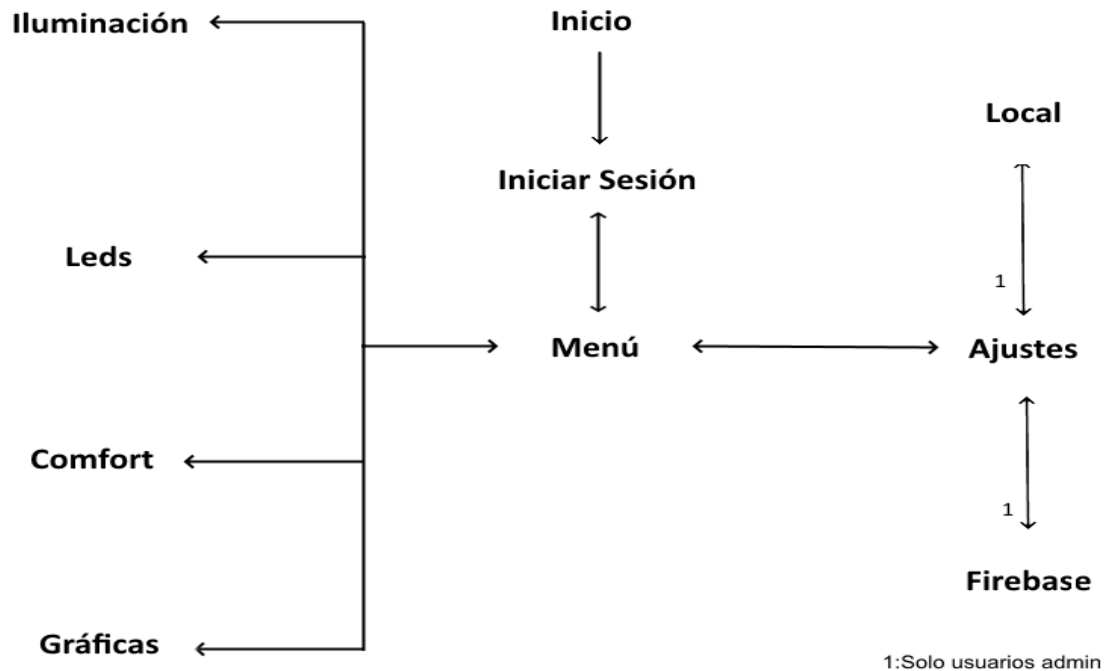


Detalle de comunicaciones

### 3.2. Diseño de la aplicación

#### Aplicación para Raspberry (Desktop App)

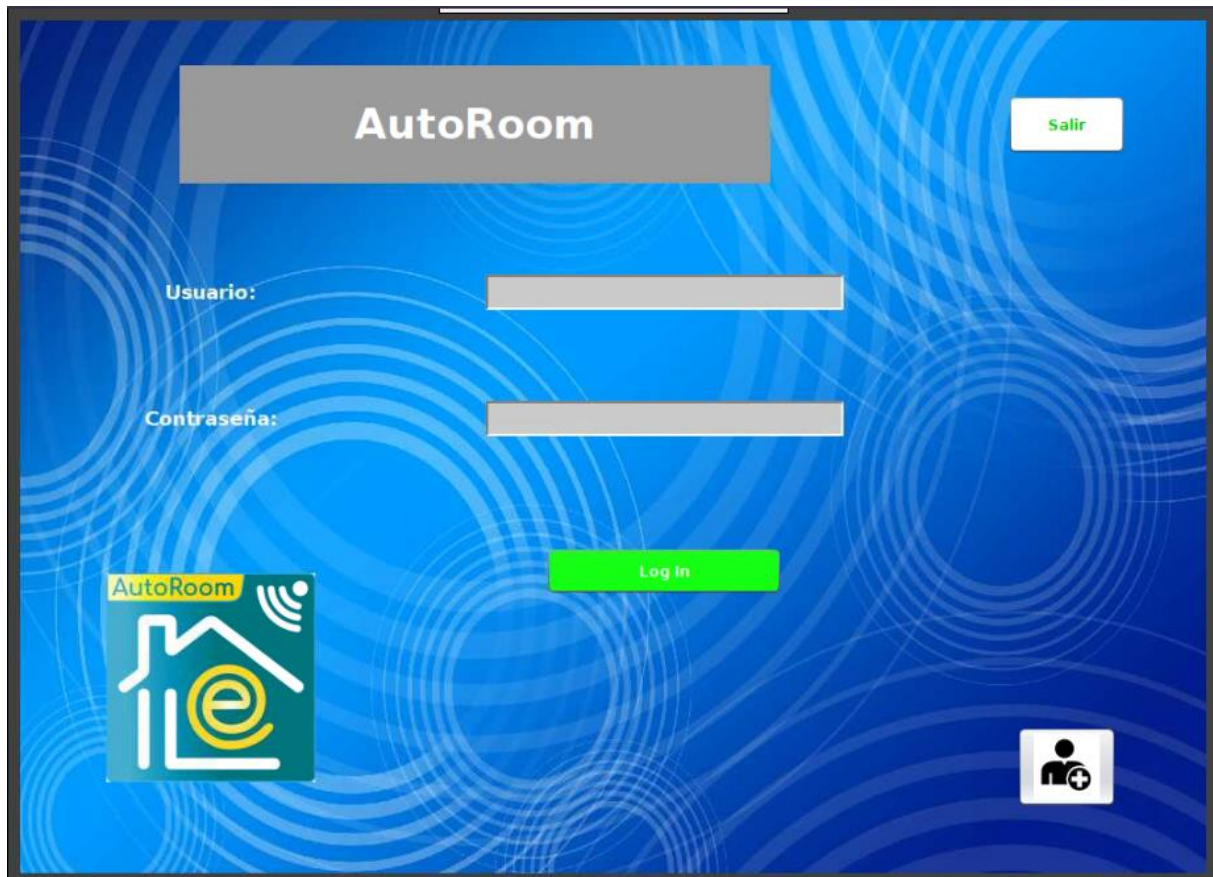
Navegación entre las pantallas.



Los usuarios administradores son los únicos que podrán acceder a los ajustes de usuarios locales y remotos puesto que en estas pantallas se pueden hacer modificaciones sobre los dos tipos de usuarios.

## Inicio de sesión

Esta pantalla es la primera que aparece al inicio de la ejecución de la aplicación. Si se cierra, se dejará de tener comunicación con la base de datos remota. Por eso, nada más ejecutarse, se conectará directamente a Firebase.



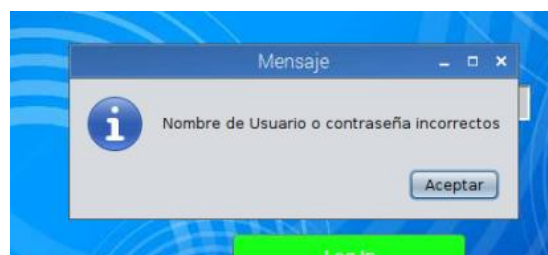
En los campos correspondientes se podrá iniciar sesión como usuario registrado.

Si no eres un usuario registrado y quieres tener acceso en la raspberry, contacta con el administrador correspondiente.

También puedes registrarte como usuario remoto en el **botón de registro**

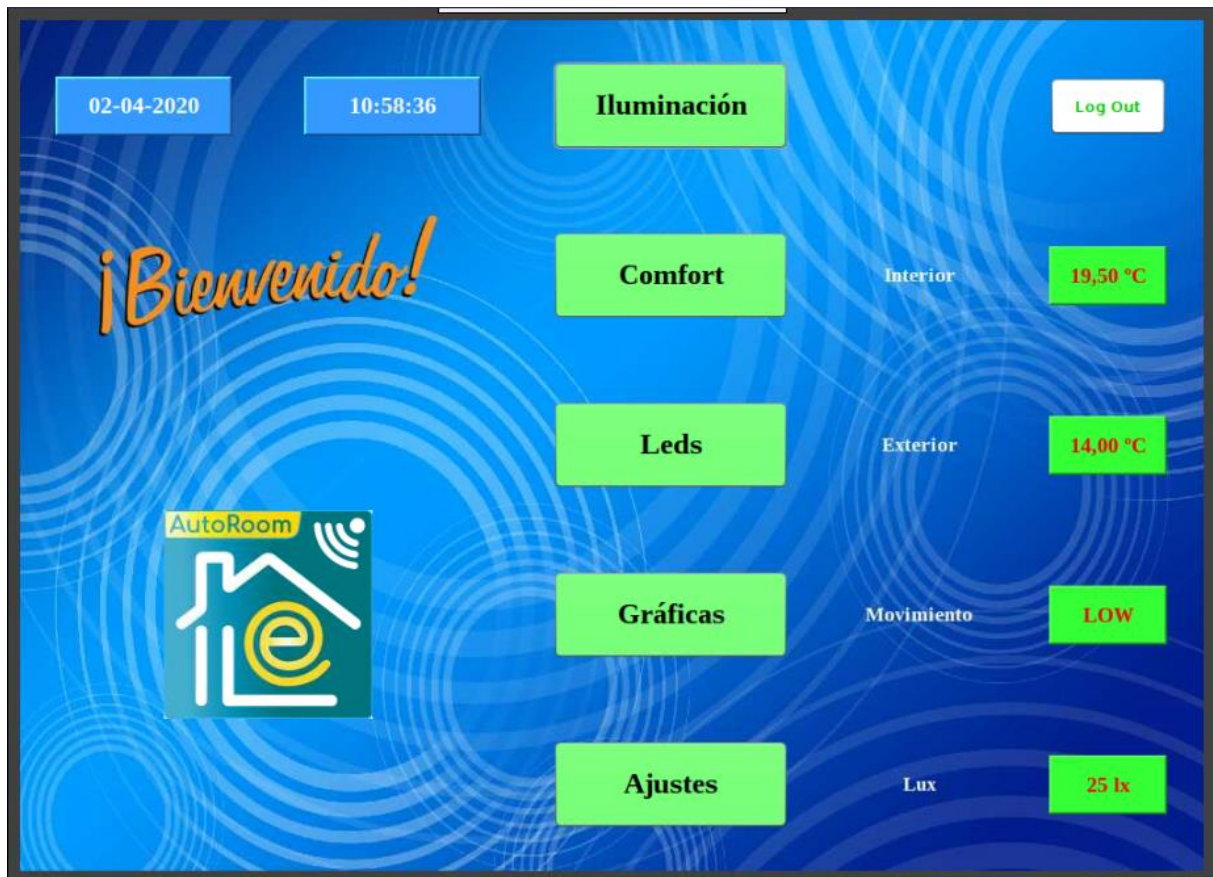


Esto abrirá la web (<https://myautoroom.web.app>) y podrás acceder al registro como usuario remoto, es decir, te registrarás en Firebase. Así podrás iniciar sesión en la App para android o en la propia web.



## Menú Principal

Esta pantalla sirve para navegar entre las distintas opciones de la aplicación.



Se puede observar la fecha y la hora actuales. También los valores de los sensores en tiempo real.

También es posible cerrar sesión.



## Iluminación

Esta pantalla será la encargada de controlar el estado de las tres lámparas. También activa/desactiva el modo auto.



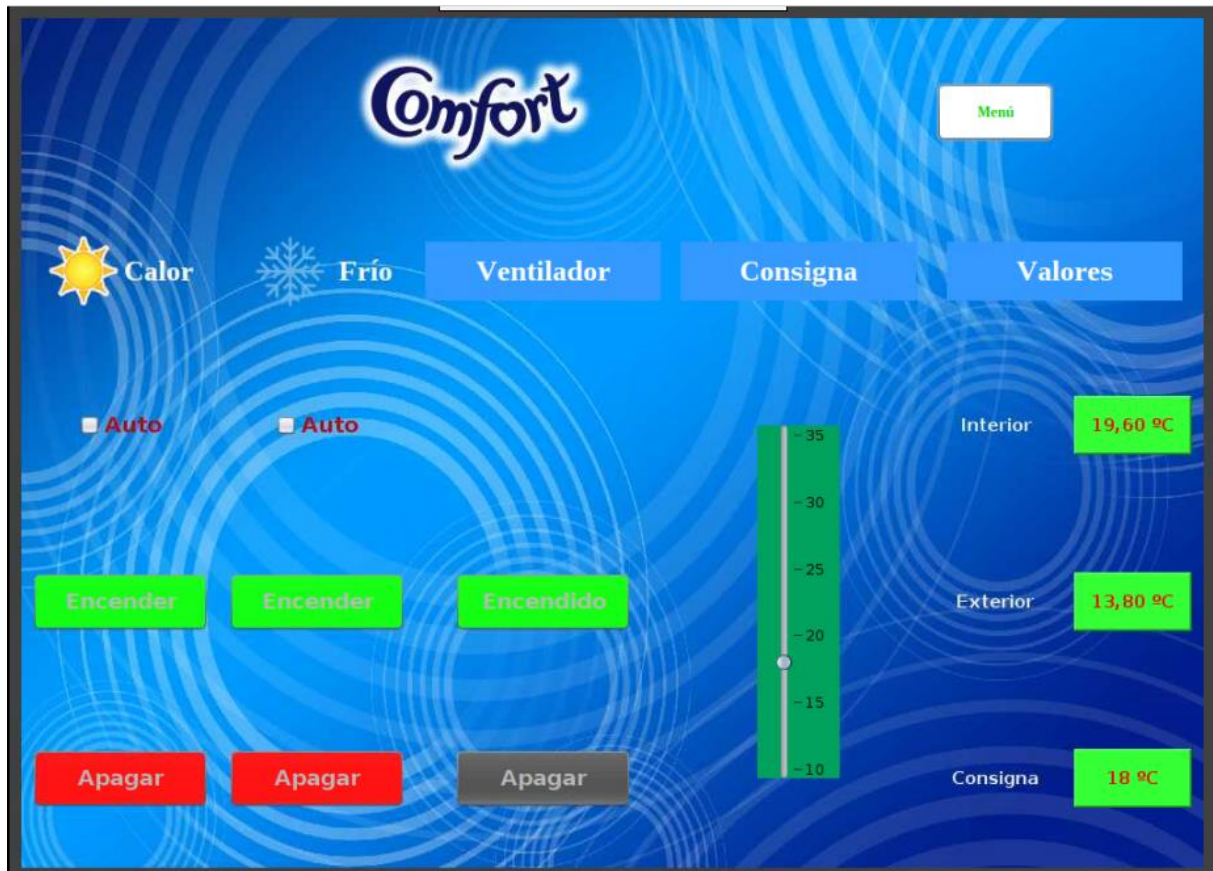
El modo “Auto” es el encargado de enlazar el estado de la lámpara con el sensor Pir, así, cada vez que se detecte movimiento, se encenderá la luminaria que tenga activado el modo automático.

Al activarse este modo, los botones de encendido y apagado, se ocultarán para evitar el activado/desactivado de la lámpara accidental o deliberadamente.

En cada cambio de estado se actualiza la base de datos

## Comfort

Esta pantalla es para el control de la climatización.



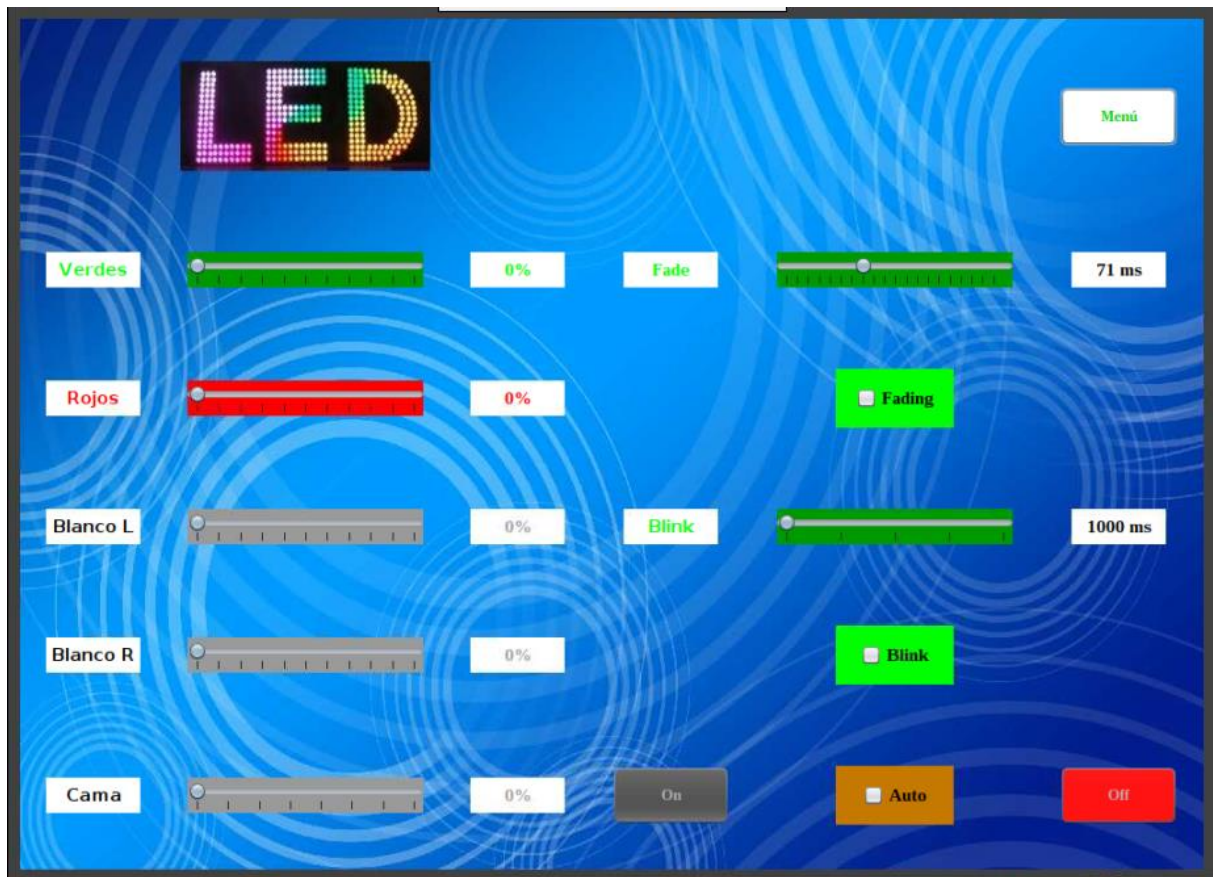
En la habitación hay instalada una bomba frío/calor de la cual sólo se controla su modo, es decir, frío o calor.

Se puede controlar, con el modo "Auto", la temperatura a la que queremos que se encuentre la habitación.



## Leds

En esta pantalla se controlan las escenas ambientales automáticas o manuales.



El modo fading, es el modo “respiración”, los leds se encienden y se apagan poco a poco. El tiempo se puede controlar con el slider “Fade”. Este modo se activa/desactiva con la casilla “Fading”.

El modo blink, es el modo “parpadeo”, los leds se encienden y se apagan alternativamente. El tiempo se puede controlar con el slider “Blink”. Este modo se activa/desactiva con la casilla “Blink”.

El modo auto, este modo depende del sensor de movimiento, cuando lo detecta, se encienden poco a poco y cuando ya no hay movimiento, se apagan poco a poco. Este modo se activa/desactiva con la casilla “Auto”.

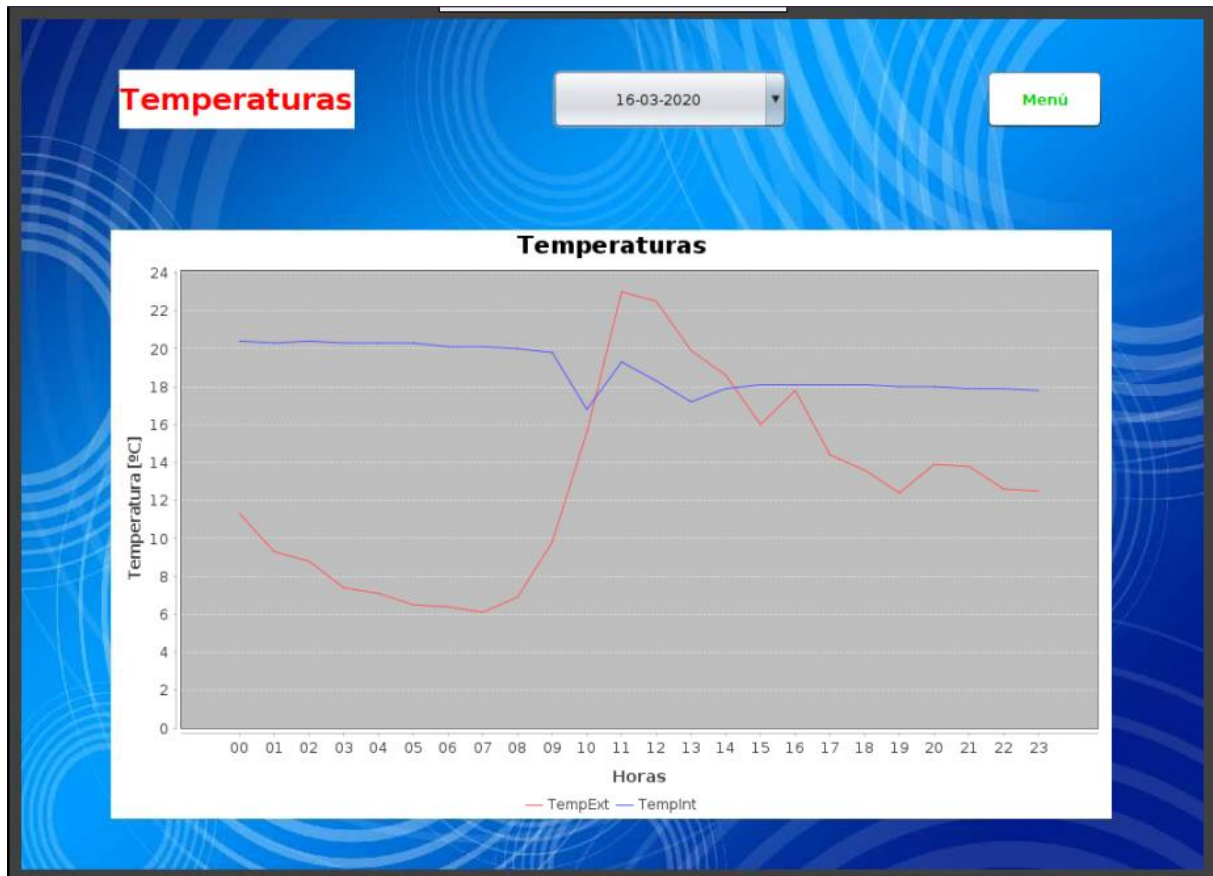
Cada led puede encenderse y apagarse individualmente usando los slider correspondientes.

Hay dos botones de encendido y apagado que enciende o apaga todos los leds a la vez.

## Gráficos de temperaturas.

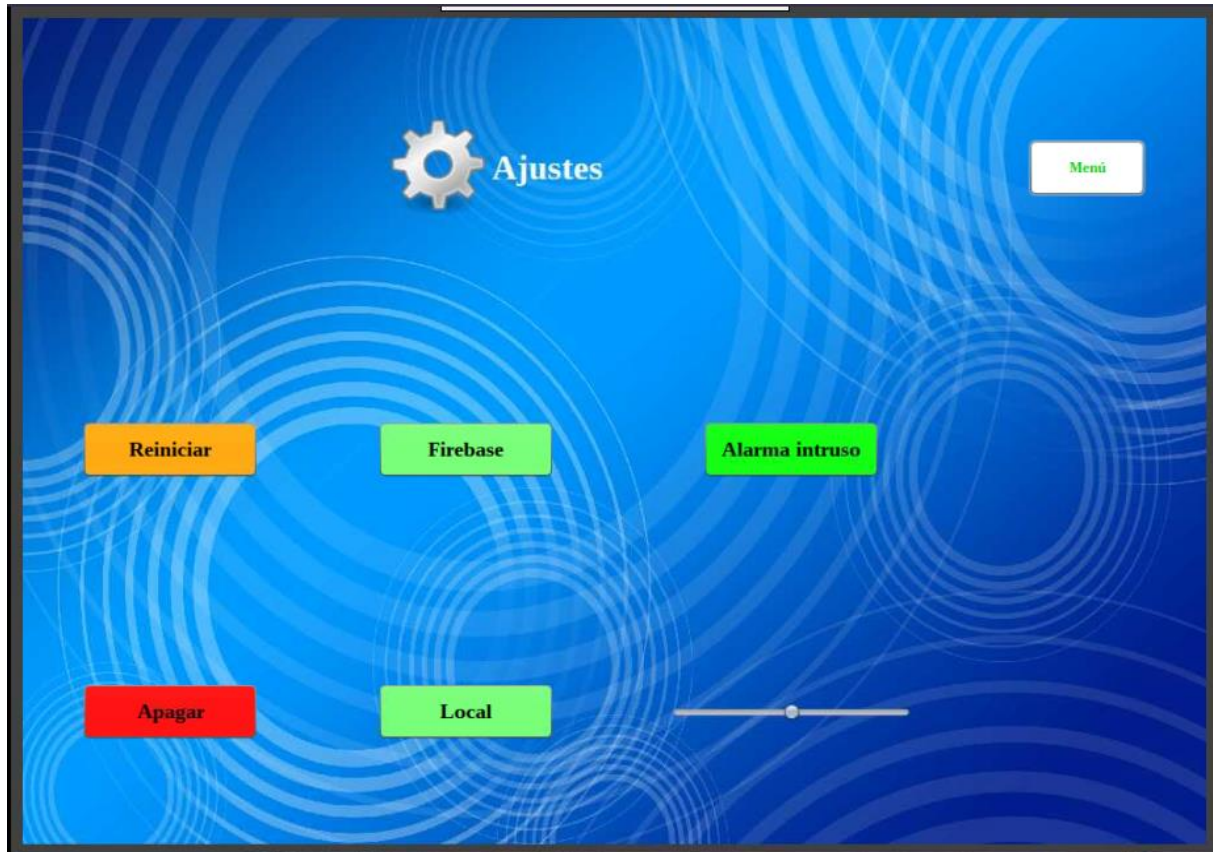
En esta pantalla se puede visualizar la evolución de la temperatura tanto exterior (Rojo), como la interior (Azul), a lo largo del día.

En el desplegable, se puede elegir el día que queremos visualizar



## Ajustes.

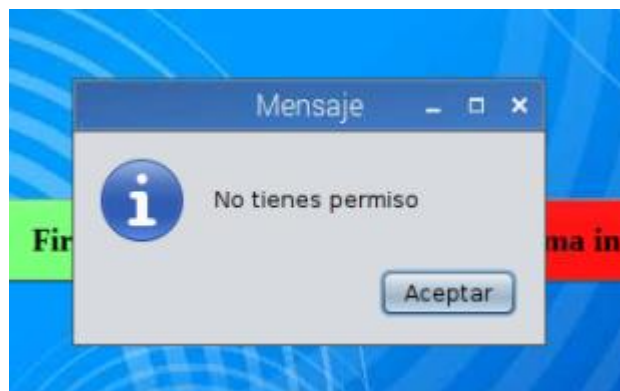
(**TODO LIST**). En esta pantalla se puede reiniciar/apagar el sistema, es decir la propia Raspberry.



El botón Firebase, lleva a una pantalla para la gestión de usuarios remotos

El botón de Local, lleva a una pantalla para la gestión de usuarios locales.

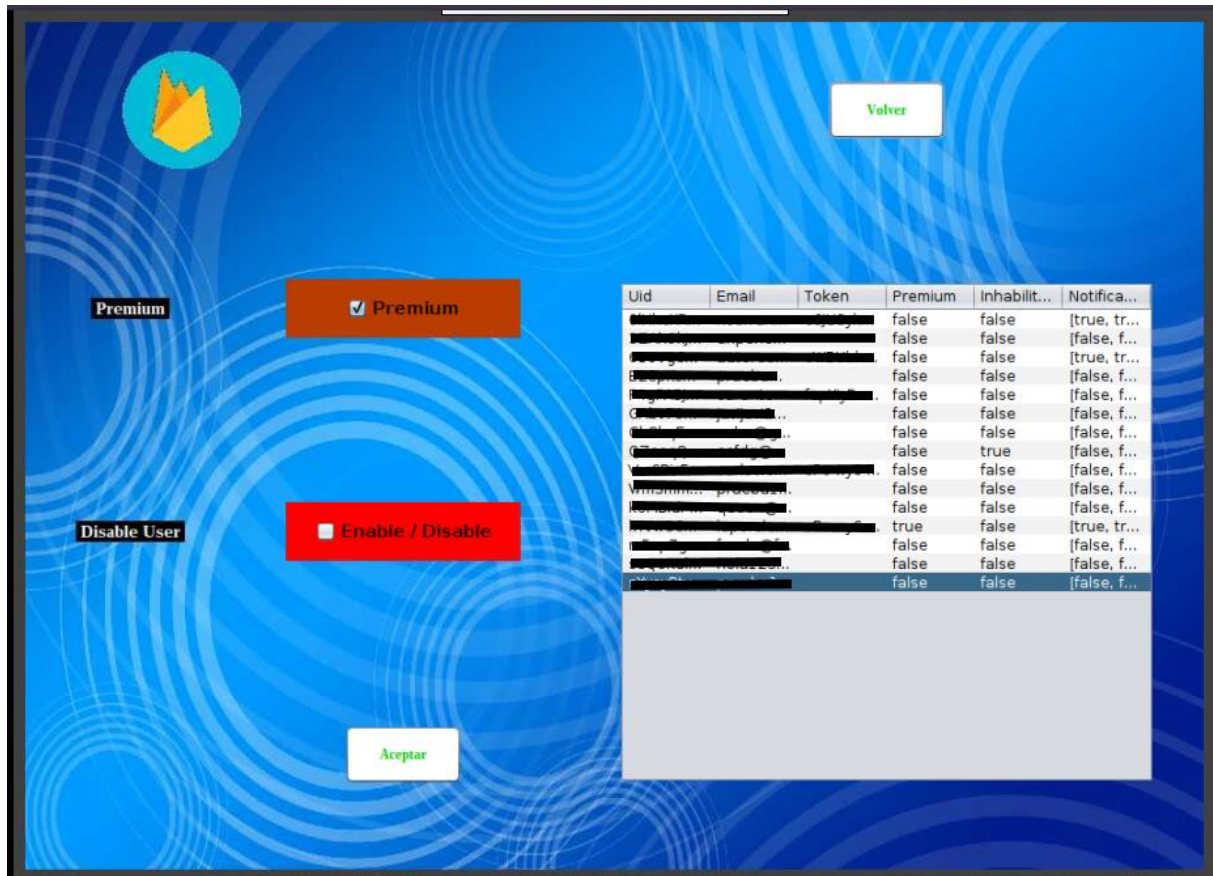
El Slider es para mover el servo sobre el que va montada la cámara de seguridad





### Pantalla de usuarios remotos.

Aquí se visualizan todos los usuarios que se han registrado a través de la web o a través de la App para Android.



Se pueden deshabilitar/ habilitar cada usuario en la casilla "Enable/Disable".

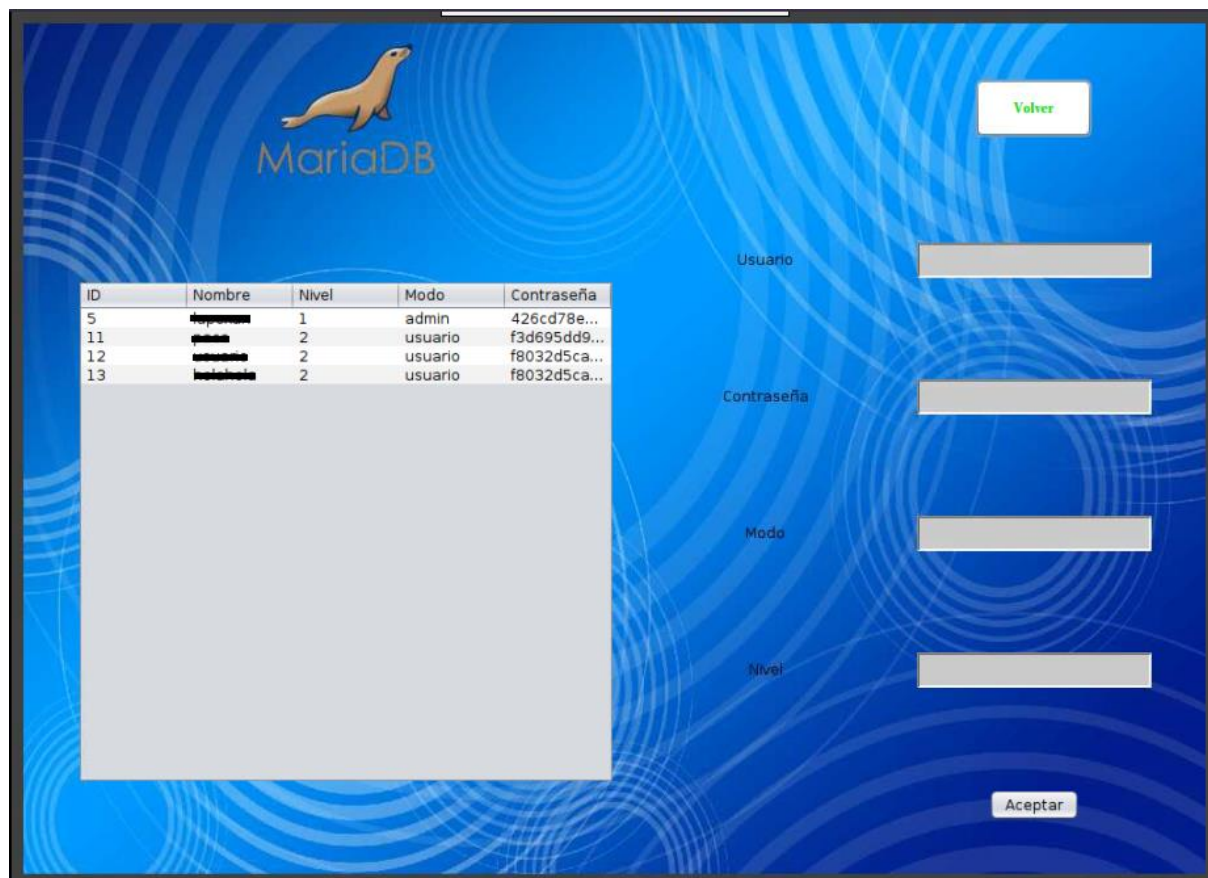
Se pueden hacer Premium a los usuarios remotos para tener más acceso en la App.

Primero, elige el usuario que quieras modificar, cambia los parámetros que desees y pulsa sobre el botón aceptar.

Para volver atrás, pulsa "volver".

### Pantalla de usuarios locales

En esta pantalla podremos modificar a los usuarios que tienen acceso a la Raspberry



ID	Nombre	Nivel	Modo	Contraseña
5	root	1	admin	426cd78e...
11	root	2	usuario	f3d695dd9...
12	root	2	usuario	f8032d5ca...
13	root	2	usuario	f8032d5ca...

Usuario

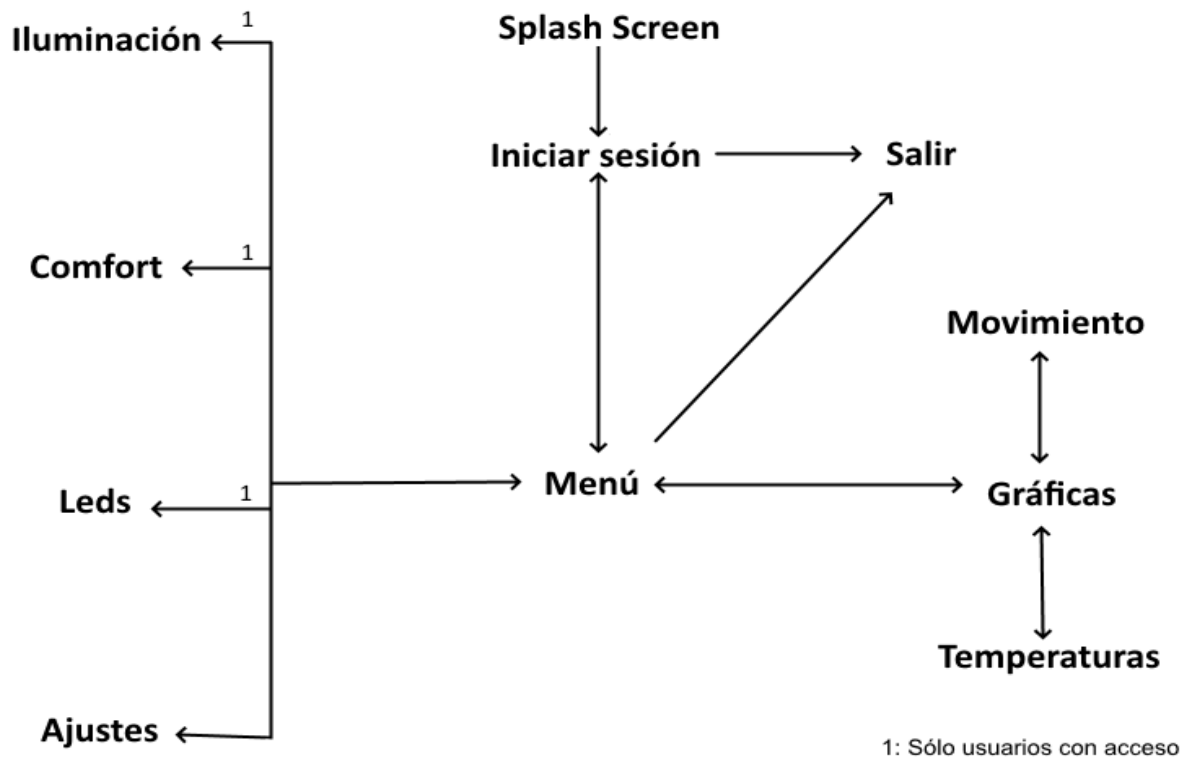
Contraseña

Modo

Nivel

## Aplicación para Android (Mobile App)

Navegación entre las pantallas



## Splash Screen

Es la primera pantalla que aparece nada más abrir la aplicación a modo de bienvenida.

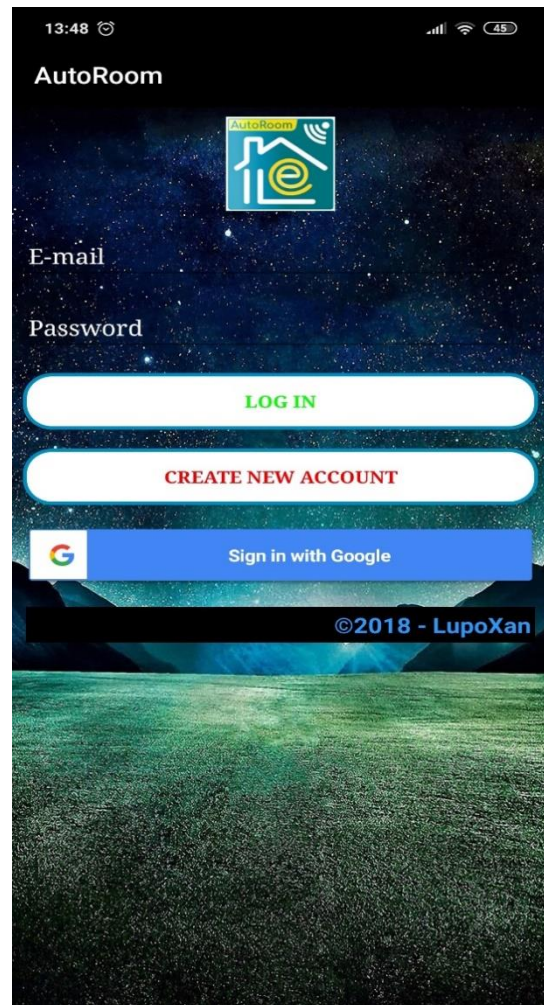


En esta pantalla a nivel de programación se comprueba únicamente si la sesión está iniciada o no. En el caso de no estar iniciada, se pasará a la pantalla de registro e inicio de sesión. En caso contrario, se iniciará la mainActivity.

## Login Screen

En esta pantalla es posible iniciar sesión (si no está iniciada) o registrarse. Hay dos formas diferentes para hacerlo:

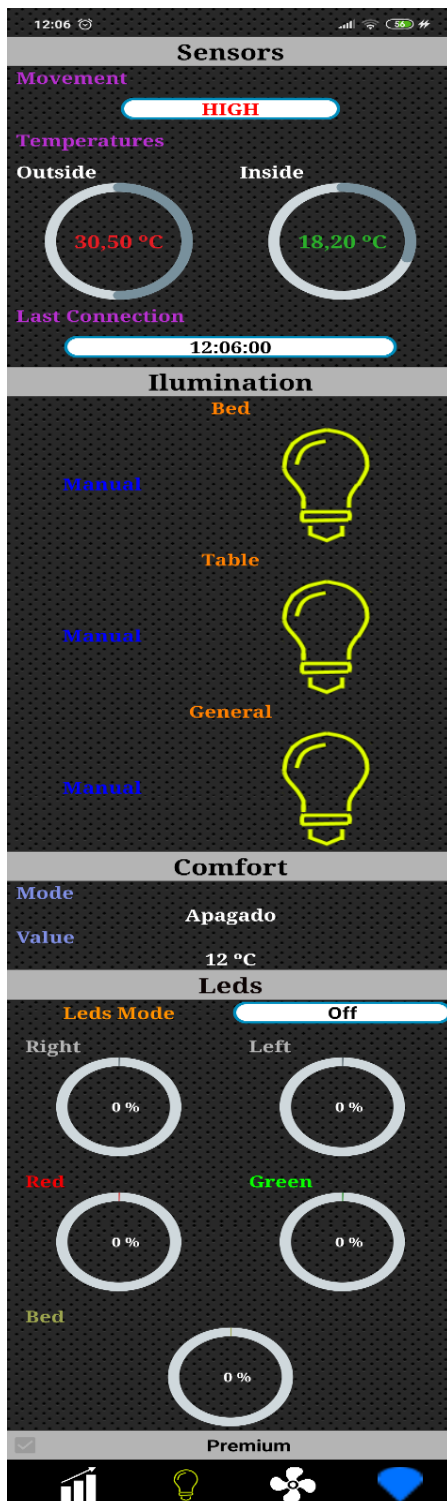
1. Por cualquier correo electrónico: si se usa esta opción, primero, habrá que registrarse para iniciar sesión. Para ello introducir el email y la contraseña en los campos correspondientes y clicar en el botón de crear nueva cuenta. Una vez que nos registremos, podremos clicar en el botón de iniciar sesión.
2. Por nuestra cuenta de google: simplemente clicamos en el botón con el símbolo de Google y nos aparecerá las cuentas asociadas en nuestro teléfono. Cuando elegimos la cuenta, se registra e inicia sesión automáticamente.





## Main Activity

Esta pantalla es la que se carga una vez iniciada la sesión y se visualizan todos los estados de cada sector. Los sectores son:

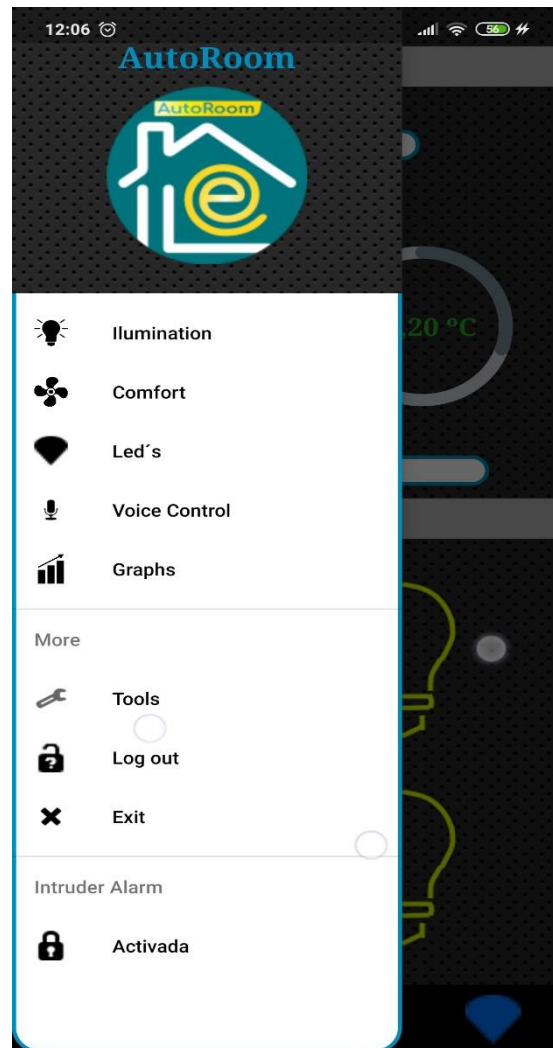


1. **Sensores:** Los valores de los sensores son a tiempo real.
2. **Iluminación:** Los estados de las lámparas se iluminación aquí.
3. **Comfort:** El estado de la bomba de calor y el ventilador / radiador.
4. **Leds:** El estado de los leds.

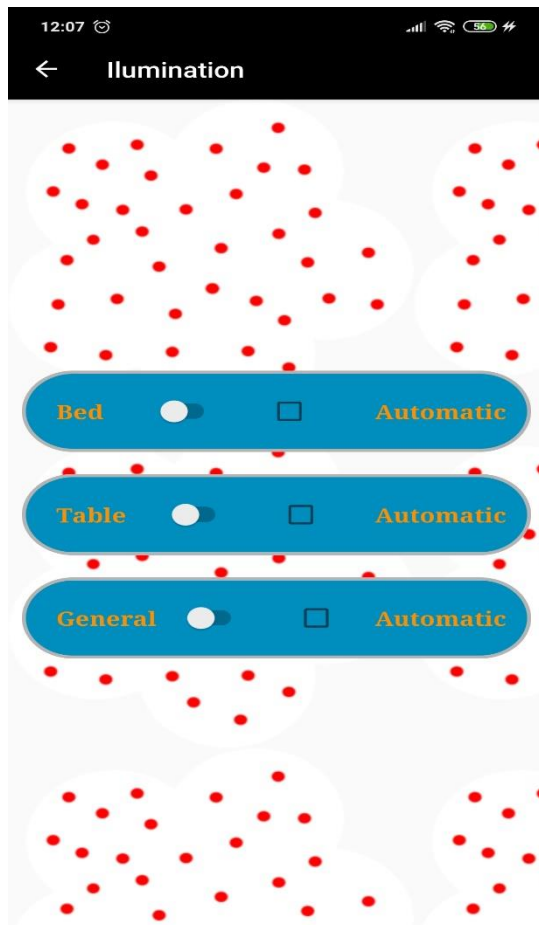
En la barra de abajo se puede acceder a las distintas pantallas:

## Navigation View

Esta es una opción de la pantalla principal. Se visualiza deslizando hacia la derecha.

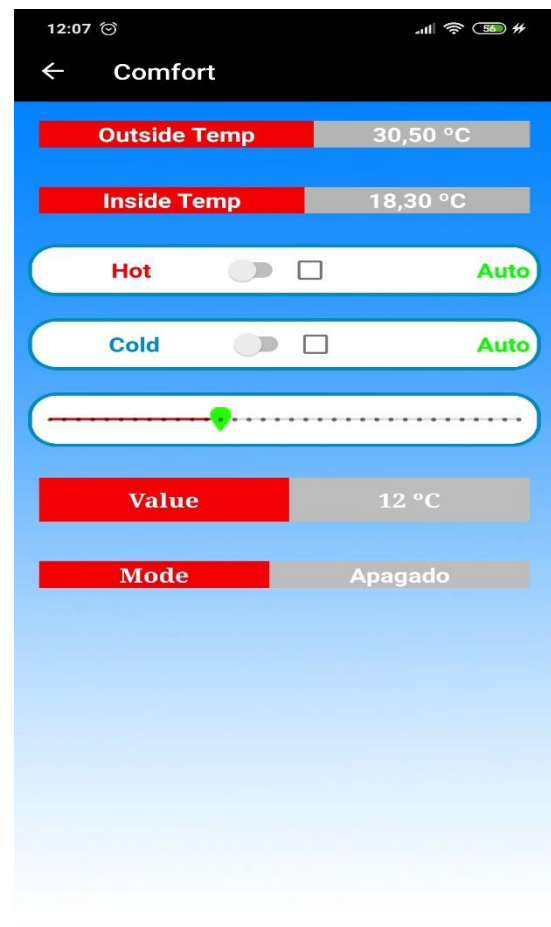


## Lights Screen

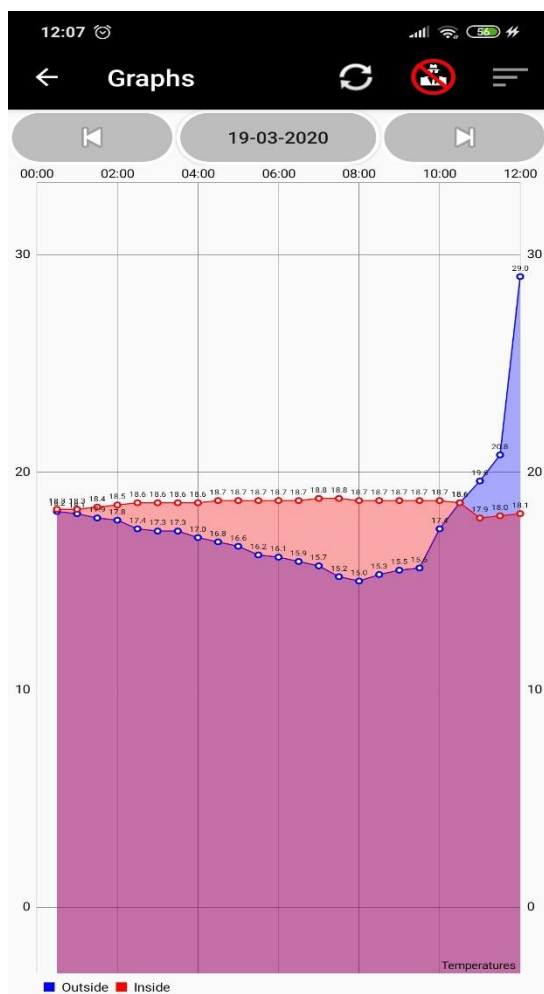


## Comfort Screen

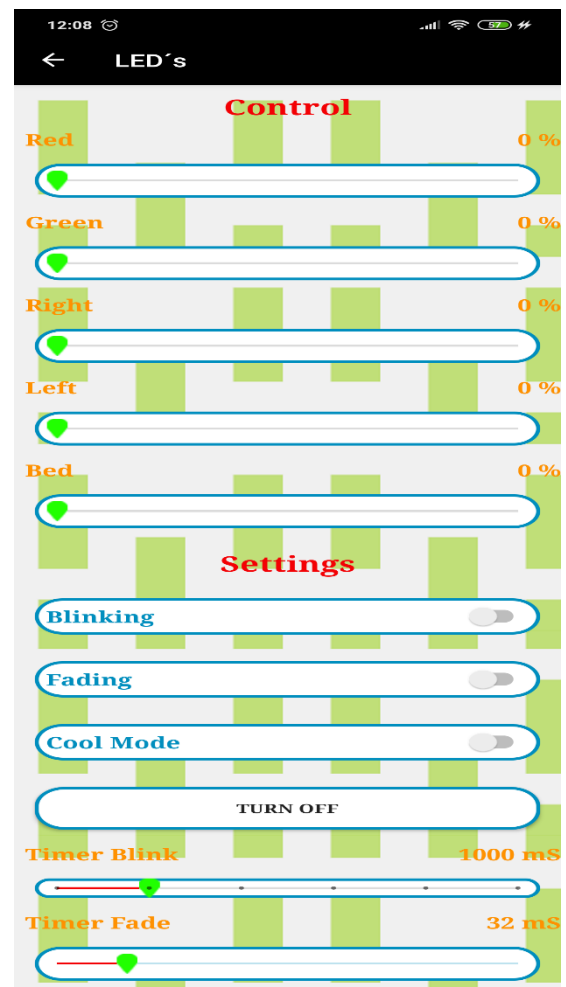
En esta pantalla se puede controlar la climatización (comfort) de la habitación, cambiando el valor de la consigna o del modo (frío / calor).



## Graphs Screen

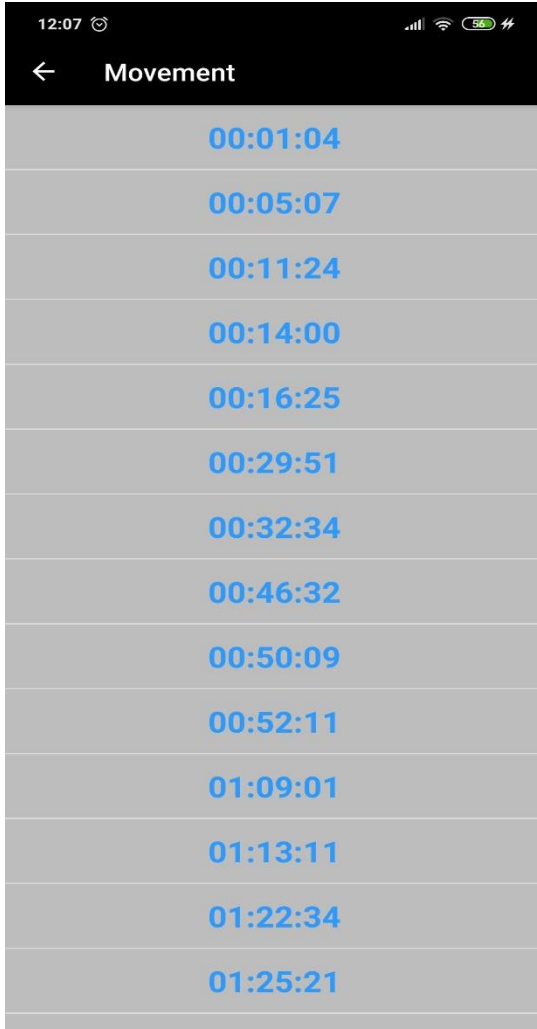


## Leds Screen



### Movement Values Screen

Esta pantalla es la que muestra el historial de las veces que se ha detectado movimiento a lo largo del día. Se muestra solo la hora en la que el detector ha dado HIGH.



The screenshot shows a mobile application interface with a black header bar. On the left of the header is a back arrow icon, and on the right is the title 'Movement'. Below the header is a list of 14 movement detection times, each displayed in blue text on a light gray background. The times are: 00:01:04, 00:05:07, 00:11:24, 00:14:00, 00:16:25, 00:29:51, 00:32:34, 00:46:32, 00:50:09, 00:52:11, 01:09:01, 01:13:11, 01:22:34, and 01:25:21. The status bar at the top of the phone shows the time 12:07, signal strength, Wi-Fi, and 50% battery.

00:01:04
00:05:07
00:11:24
00:14:00
00:16:25
00:29:51
00:32:34
00:46:32
00:50:09
00:52:11
01:09:01
01:13:11
01:22:34
01:25:21

Para ver los valores de otros días tan solo has de elegir el día que quieras en el calendario de la pantalla de los gráficos.

## Values Screen

En esta pantalla se puede observar el historial de los valores de los sensores del día. Al deslizar hacia abajo se observarán todos los valores registrados hasta la hora actual.





Valor de la hora en la que se han registrado los valores, normalmente es cada 30 minutos.


00:00:00

Valores de los sensores. De izquierda a derecha son, movimiento\*, exterior, interior.

LOW 18,20 °C 18,30 °C

Para ver los valores de otros días tan solo has de elegir el día que quieras en el calendario de la pantalla de los gráficos. El título de la pantalla cambiará con el día seleccionado.

12:07    



19-03-2020

	00:00:00	
LOW	18,20 °C	18,30 °C
	00:30:00	
HIGH	18,10 °C	18,30 °C
	01:00:00	
LOW	17,90 °C	18,40 °C
	01:30:00	
LOW	17,80 °C	18,50 °C
	02:00:00	
HIGH	17,40 °C	18,60 °C
	02:30:00	
LOW	17,30 °C	18,60 °C
	03:00:00	
LOW	17,30 °C	18,60 °C
	03:30:00	
LOW	17,00 °C	18,60 °C
	04:00:00	
LOW	16,80 °C	18,70 °C
	04:30:00	
LOW	16,60 °C	18,70 °C
	05:00:00	
LOW	16,20 °C	18,70 °C
	05:30:00	
LOW	16,10 °C	18,70 °C
	06:00:00	
LOW	15,90 °C	18,70 °C
	06:30:00	
LOW	15,70 °C	18,80 °C
	07:00:00	
LOW	15,90 °C	18,80 °C

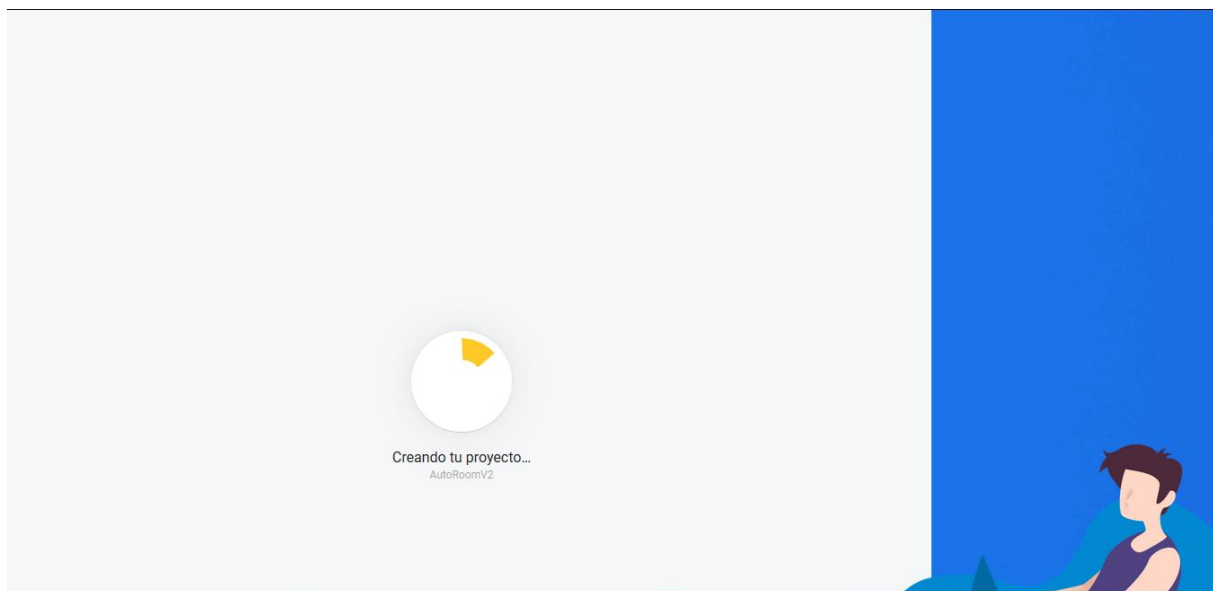
\* HIGH → Se ha detectado movimiento | LOW → No se ha detectado movimiento

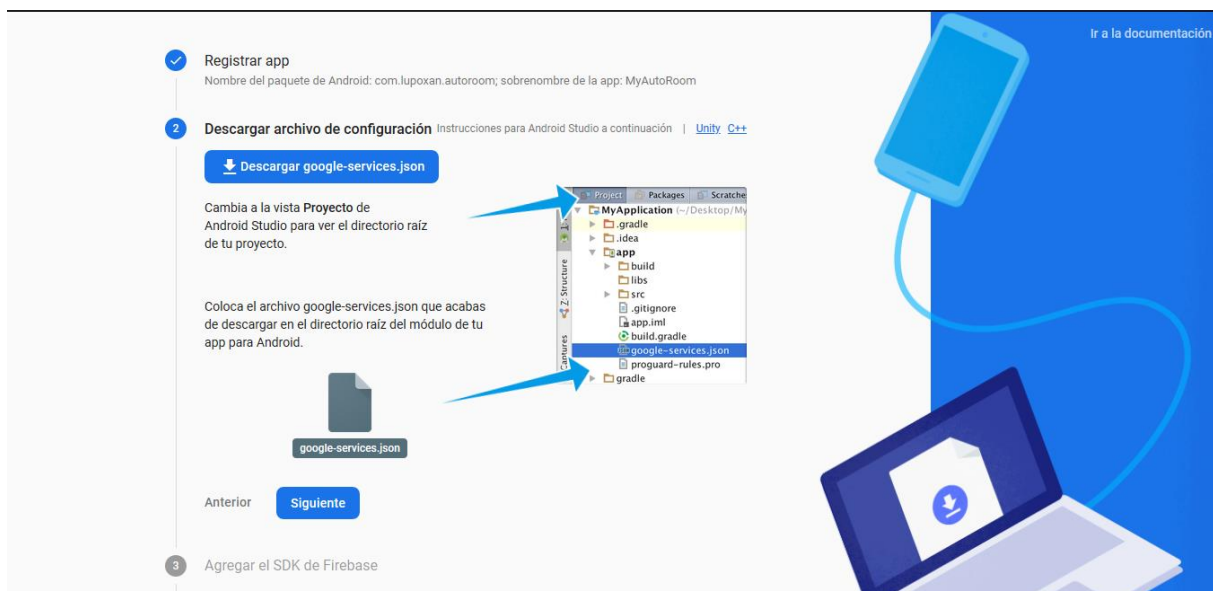


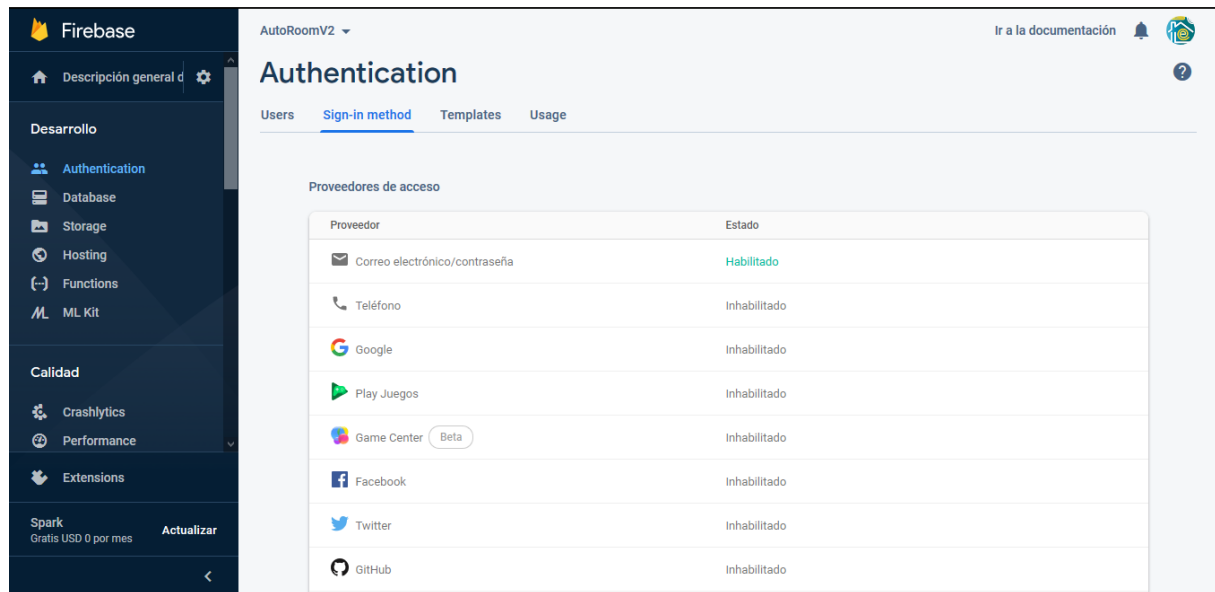
## 4. Manuales

### Firestore Google Inc.

<https://console.firebase.google.com/>



































## 5. Conclusiones

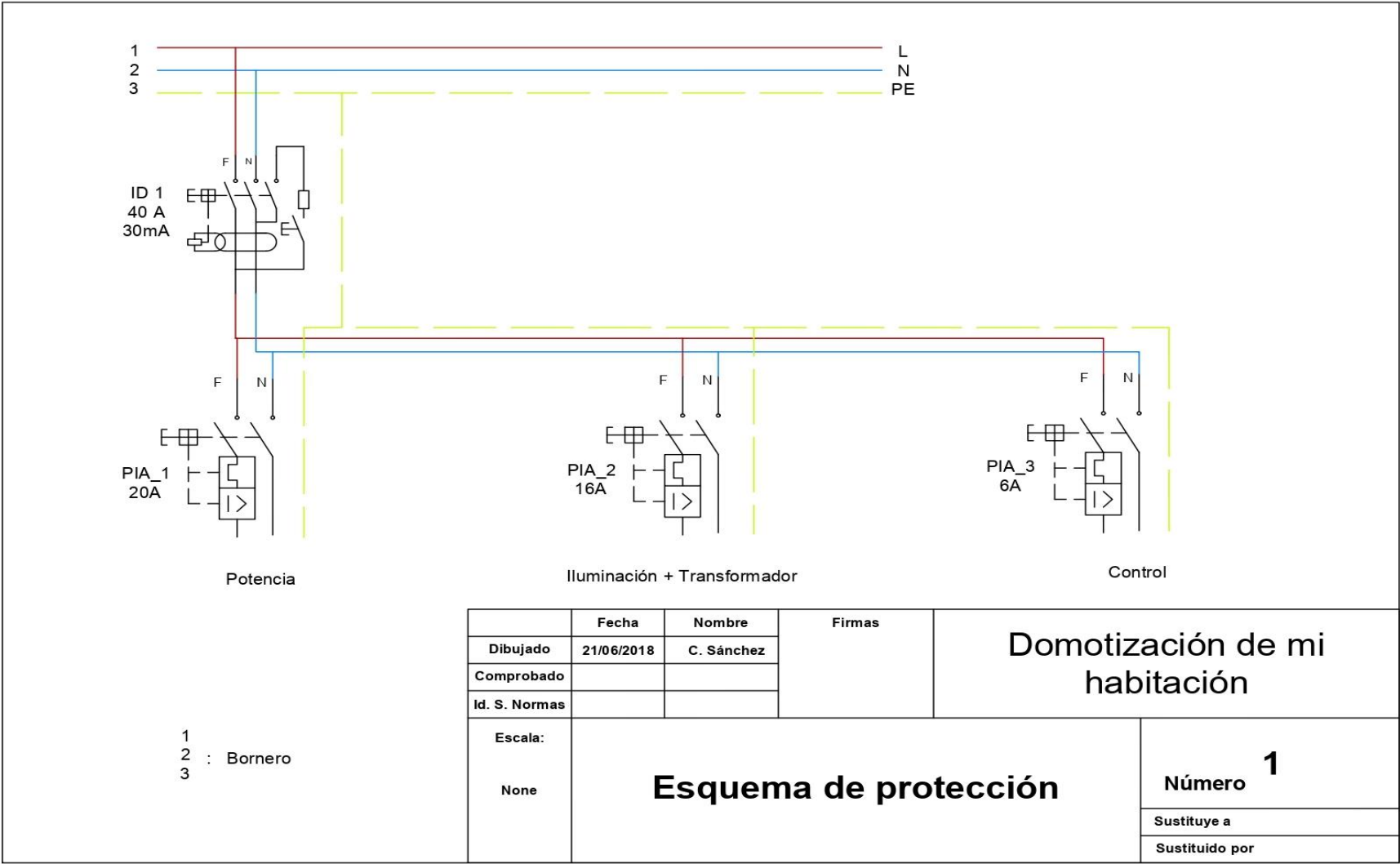
## 6. Propuestas de mejoras

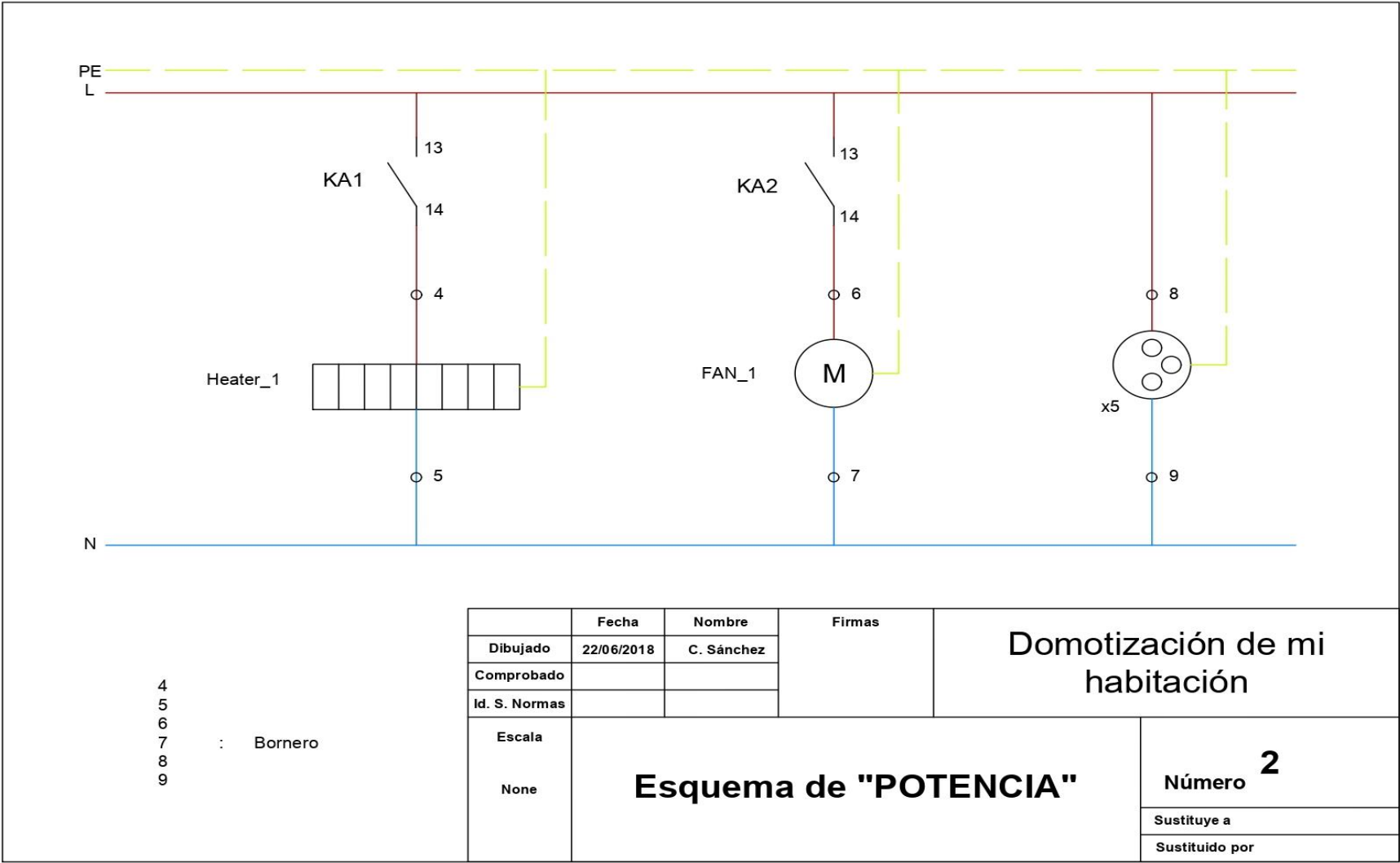
## 7. Bibliografía

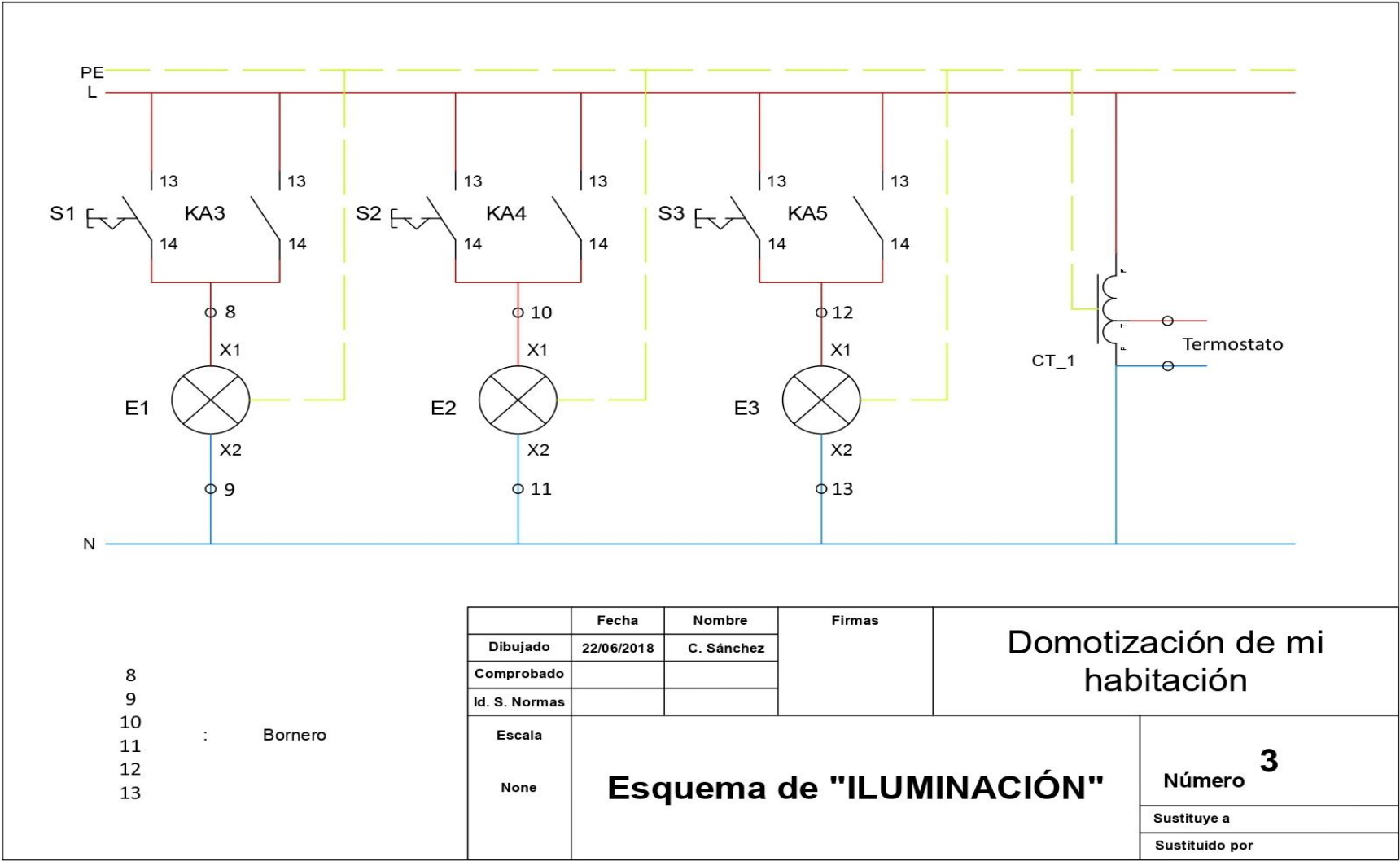
## 8. Anexos

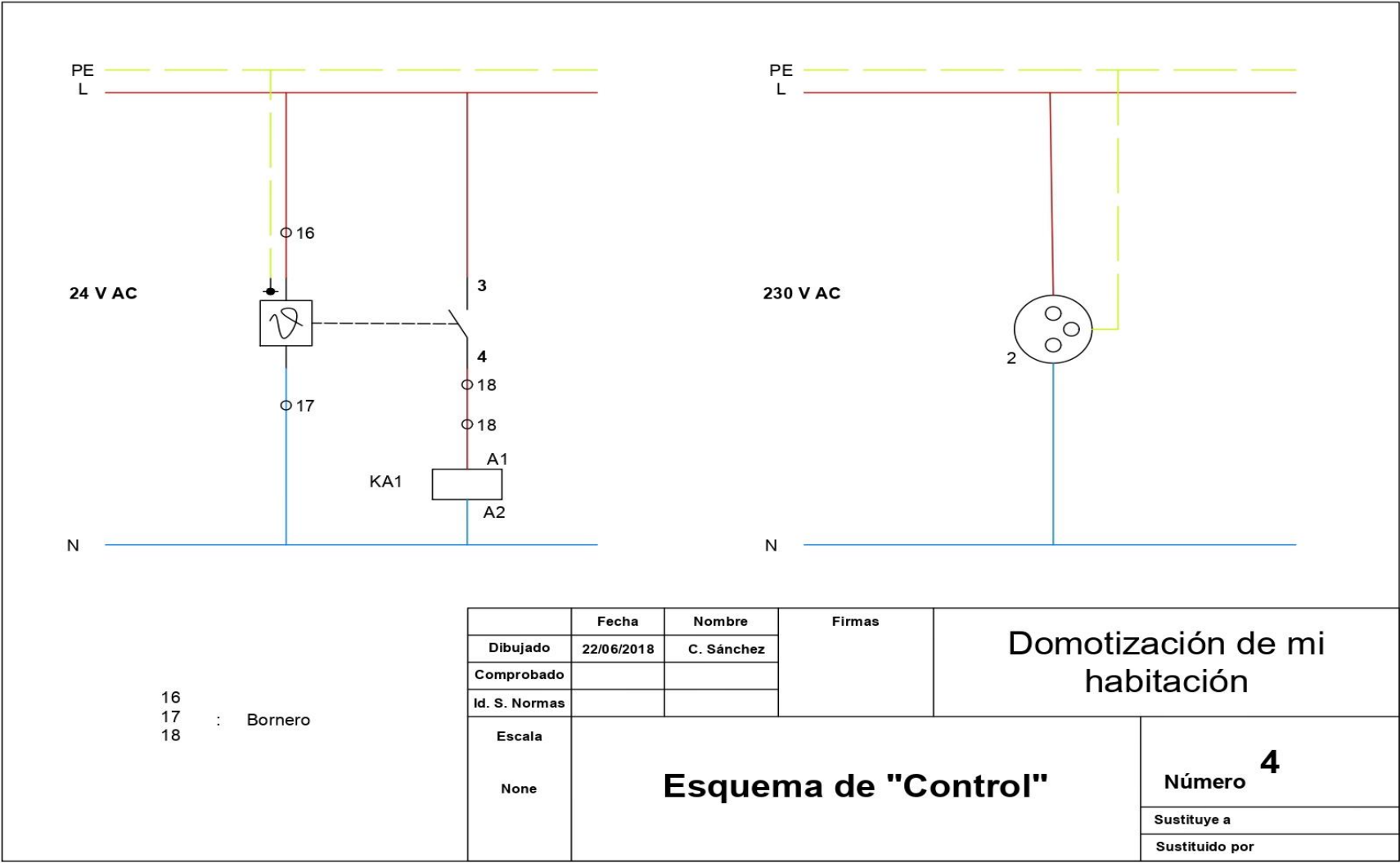
													
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	
													
<b>L</b>	<b>N</b>	<b>PE</b>	<b>L</b>	<b>N</b>	<b>L</b>	<b>N</b>	<b>L</b>	<b>N</b>	<b>L</b>	<b>N</b>	<b>L</b>	<b>N</b>	
Alimentación			Heater_1		Regletero		E1		E2		E3		

	Fecha	Nombre	Firmas	Domotización de mi habitación
Dibujado	22/06/2018	C. Sánchez		
Comprobado				
Id. S. Normas				
Escala	Bornero			Número 0
None				Sustituye a
				Sustituido por











**RaspberryPi 3B**

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39

1: 3v3	8: GPIO14	15: GPIO22	22: GPIO25	29: GPIO5	36: GPIO16
2: 5v	9: Gnd	16: GPIO23	23: General	30: Gnd	37: Blanco R
3: Cama	10: Blanco L	17: 3v3	24: GPIO8	31: GPIO6	38: GPIO20
4: 5v	11: Blanco C	18: Rojos	25: Gnd	32: GPIO12	39: Gnd Leds
5: Mesa	12: Servo	19: GPIO10	26: GPIO7	33: GPIO13	40: Verdes
6: Gnd	13: GPIO27	20: Gnd	27: ID_SD	34: Gnd	
7: W-1	14: Gnd	21: Comfort	28: ID_SC	35: PIR	

**Relé Shield**

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

1: Gnd	6: In 5
2: In 1	7: In 6
3: In 2	8: In 7
4: In 3	9: In 8
5: In 4	10: Vcc

	Fecha	Nombre	Firmas	Domotización de mi habitación
Dibujado	22/06/2018	C. Sánchez		
Comprobado				
Id. S. Normas				
Escala	<b>Esquema conexión Raspberry Pi-3B + Relé shield</b>			Número <b>5</b>
None				
				Sustituye a
				Sustituido por

