|  |
| --- |
| A description...  **UNIVERSIDAD EUROPEA DE MADRID**  A description...  ESCUELA ARQUITECTURA INGENIERÍA Y DISEÑO  CICLO FORMATIVO DE GRADO SUPERIOR  DESARROLLO DE APLICACIONES MULTIPLATAFORMA |
| PROYECTO FIN DE CICLO |
| **FUTH (Futurizing Homes)** |
|  |
|  |
| **CURSO 2017-18** |

**TÍTULO**: Futh

**AUTORES**: Iván Gabriel Pajón Rodríguez

Jozet Stiven Quipuscoa Ponte

**TUTOR DEL PROYECTO**: Ernesto Ramiro Córdoba

**FECHA DE LECTURA**: ... de Junio de 2018

**CALIFICACIÓN**:

Fdo: NOMBRE APELLIDO1 APELLIDO2

Ernesto Ramiro Córdoba

Tutor/a del Proyecto

# RESUMEN

En este trabajo se va a desarrollar una aplicación para dispositivos Android que sea capaz de monitorizar una serie de sensores (tales como de temperatura, humedad, gas…) y controlar objetos cotidianos como una bombilla o un enchufe. Cada usuario que se descargue la aplicación deberá iniciar sesión con Google+ para poder empezar a usarla, lo que asegura un mayor control de uso sobre los usuarios de nuestra app. Uno de los fundamentos de este trabajo es una aplicación útil, real y escalable, refiriéndonos a escalable en el sentido de que para la realización del mismo solo se desarrollará un dispositivo, pero la app estará diseñada para permitir añadir múltiples dispositivos (como si fuera un producto que podemos comprar). De igual manera, nos centraremos en desarrollar una interfaz simple, organizada y atractiva, permitiendo a usuarios sin mucho conocimiento de tecnología y/o domótica entender cómo funciona nuestro producto y permitiéndoles usar las herramientas que ponemos a su disposición sin dificultad alguna.

A su vez, se va a desarrollar desde cero un dispositivo basado en código abierto que sea capaz de monitorizar una serie de sensores y objetos cotidianos como los antes mencionados, incluyendo el hardware necesario para todo ello. Este dispositivo se conectará Firebase, al igual que la app, y de esa manera serán capaces de comunicarse entre ellos, dando la posibilidad de controlarlo tanto desde una red local como de internet, sin tener que depender de direcciones IP y/u otros factores que suelen restringir el perfil medio del usuario que consume el producto.

También se va a desarrollar un servicio de notificaciones “inteligente” que interprete los datos que recibe del dispositivo y en base a ellos genere el aviso más oportuno para el usuario. Dicho servicio estará implementado no sólo con dispositivos móviles Android, sino que también tendrá soporte para coches con Android Auto y relojes con Android Wear.

# ABSTRACT

In this work we will develop an application for Android devices capable of monitoring some sensors (such as temperature, humidity, gas ...) and control everyday objects such as a light bulb or a plug. Each user that downloads the application must log in with Google+ in order to start using it, which ensures greater usage control over the users of our app. One of the fundamentals of this work is a useful, real and scalable application, referring us to scalable in the sense that for the realization of the same one device will be developed, but the app will be designed to allow adding multiple devices (as if it were a product that we can buy). In the same way, we will focus on developing a simple, organized and attractive interface, allowing users without much knowledge of technology and/or domotics to understand how our product works and allowing them to use the tools that we put at their disposal without any difficulty.

At the same time, we will develope a device from scratch based on open source that is capable of monitoring some sensors and everyday objects such as the ones mentioned above, including the necessary hardware for all this. This device will connect Firebase, like the app, and in that way they will be able to communicate with each other, giving the possibility to control it both from a local network and the internet, without having to depend on IP addresses and/or other factors that they usually restrict the average profile of the user who consumes the product.

It will also develop an "intelligent" notification service that interprets the data it receives from the device and, based on them, generates the most opportune notification for the user. This service will be implemented not only with Android mobile devices, but will also have support for cars with Android Auto and watches with Android Wear.

# AGRADECIMIENTOS

Aquí nos ponemos sentimentales y expresamos el agradecimiento a quien pueda haber sido significativo en el desarrollo del proyecto (parejas, familia, profesores…)



Esta obra se distribuye bajo una licencia Creative Commons.

Se permite la copia, distribución, uso y comunicación de la obra si se respetan las siguientes condiciones:

* Se debe reconocer explícitamente la autoría de la obra incluyendo esta nota y su enlace.
* La copia será literal y completa.
* No se podrá hacer uso de los derechos permitidos con fines comerciales, salvo permiso expreso de los autores.

El texto precedente no es la licencia completa sino una nota orientativa de la licencia original completa (jurídicamente válida) que puede encontrarse en: http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es

# Índice

[RESUMEN 3](#_Toc515453982)

[ABSTRACT 4](#_Toc515453983)

[AGRADECIMIENTOS 5](#_Toc515453984)

[Índice 7](#_Toc515453985)

[Introducción 8](#_Toc515453986)

[Objetivos 9](#_Toc515453987)

[Motivación 10](#_Toc515453988)

[Antecedentes 11](#_Toc515453989)

[Desarrollo del proyecto 12](#_Toc515453990)

[Herramientas tecnológicas 13](#_Toc515453991)

[Java 13](#_Toc515453992)

[Android Studio 14](#_Toc515453993)

[Firebase 15](#_Toc515453994)

[Arduino 16](#_Toc515453995)

[C++ 17](#_Toc515453996)

[Planificación 18](#_Toc515453997)

[Descripción del trabajo realizado 19](#_Toc515453998)

[Documentación técnica 21](#_Toc515453999)

[Esquemas 24](#_Toc515454000)

[Bibliografía y Webgrafía 25](#_Toc515454001)

[Anexos 26](#_Toc515454002)

[1. Apps domótica 26](#_Toc515454003)

[A. Imperihome 26](#_Toc515454004)

[B. Houseinhand KXN 29](#_Toc515454005)

[C. TaHoma by Somfy 32](#_Toc515454006)

# Introducción

El propósito de este proyecto es desarrollar un dispositivo y una aplicación para móviles conjuntamente, con el ánimo de estudiar de qué manera se podría mejorar la calidad de vida de las personas mediante el uso de la tecnología. A su vez, a medida que se vaya desarrollando el proyecto se irán estudiando aplicaciones futuras y sectores distintos al personal (por ejemplo laboral, y cómo mejorar la productividad y el confort de los empleados).

Para el desarrollo del trabajo se han analizado apps ya existentes tanto en busca de puntos fuertes y posibles mejoras como para buscar un patrón de diseño y uso en común sobre el que luego desarrollar la nuestra, son las siguientes:

* ImperiHome
* Houseinhand KXN
* TaHoma by Somfy

## Objetivos

Objetivos principales:

* Realizar una Aplicación para dispositivos con sistema operativo Android
* Implementar diferentes sensores conectados a Arduino
* Gestión del control absoluto de sensores desde la Aplicación Android
* Capacidad de planificar y realizar las diferentes tareas cada miembro del grupo

Nuestro principal objetivo consiste en la realización de una aplicación Android capaz de poder controlar los diferentes sensores añadidos previamente por el usuario.

Para ello debemos poner en práctica todos los conocimientos adquiridos durante el curso de Desarrollo de Aplicaciones Multiplataforma y si hiciese falta aprender nuevas tecnologías y/o lenguajes.

Otro de nuestros objetivos a tener en cuenta, es el poder llevar a cabo una implementación de las notificaciones de nuestra aplicación para que sean compatibles con Android Auto y Android Wear, y que los usuarios que la usen, puedan tener al alcance y de múltiples formas la información de los diferentes sensores instalados en sus casas. Esto es un gran avance tecnológico y significativo para la vida de nuestros futuros usuarios, ya que nos libera de tener que llevar nuestro dispositivo móvil a todos lados para poder recibir cualquier alerta desde nuestra aplicación.

## Motivación

La elección final de este proyecto fue debido a que creemos que la domótica está creciendo día a día, y el desafío de poder juntar dos de las grandes tecnologías actualmente, como es una aplicación Android, que son muy utilizadas en el día a día de muchas personas y el mundo de la domótica, que es una forma de simplificar más nuestras vidas.

## Antecedentes

Para el desarrollo de nuestra app hemos tenido en cuenta 3 que ya están en el Play Store, sirviendo de referentes a la hora de elegir paleta de colores, estilo de iconos, navegabilidad y organización de la nuestra.

Para el desarrollo de la parte de domótica hemos usado como referente desarrollos caseros que encontramos por internet, y trabajos realizados anteriormente por nosotros mismos. Cabe destacar que al diseñar nosotros mismos el hardware y los protocolos de comunicación de este proyecto hemos tenido que usar en gran parte nuestra imaginación y/o las herramientas que en el momento de la realización están a nuestro alcance.

A continuación exponemos el análisis de las 3 apps analizadas como antecedentes de nuestro proyecto:

**Imperihome** es una app con gran potencial, pero que lamentablemente su interfaz deja mucho que desear: mala organización, “look” de aplicación antigua y algún icono poco descriptivo. El mayor impedimento que hemos encontrado ha sido que para poder probarla nos hemos tenido que registrar, y una vez registrados, ya pudimos probar el modo demo, cosa que con las otras 2 apps no ocurría (las probamos sin registrarnos). Estamos seguros que esto penalizará bastante sobre el número de usuarios que se deciden por usar la app. ([Ver Anexo 1.A](#_Imperihome))

**Houseinhand KNX** tiene una interfaz muy limpia y organizada, con iconos descriptivos. La interfaz está construida en “modo noche”, con fondo negro y texto e iconos en blanco, lo cual cansa menos a la vista, pero quizá deberían dar la opción al usuario de elegir si desean usar la aplicación con dicho modo o no. ([Ver Anexo 1.B](#_Houseinhand_KXN))

En el caso de **Tahoma by Somfy** tenemos una app con funciones muy avanzadas y animaciones muy bien diseñadas para permitir al usuario entender qué está haciendo. ([Ver Anexo 1.C](#_TaHoma_by_Somfy))

# Desarrollo del proyecto

En el desarrollo de la aplicación para dispositivos Android, hemos empezado implementando la pantalla de Login, en la cual se podrá iniciar sesión mediante cuenta de Google+. Hemos pensado en esa única forma de poder iniciar sesión con nuestra aplicación debido a que creemos innecesarios los otros métodos de inicio como:

* Facebook
* Twiter
* GitHub

Al buscar e instalar nuestra aplicación mediante Play Store, damos por hecho que el usuario tiene cuenta de Google+, ya que sin ella no sería capaz de hacer descargas desde Play Store y por lo tanto no sería ningún inconveniente el inicio de sesión para nuestros usuarios.

Para la autenticación por Google+ hemos utilizado el servicio Authentication que nos proporciona Firebase, ya que es fácil de implementar y más seguro.

Al hacer click en el botón de Iniciar sesión, si se tiene una cuenta de Google+ asociada en nuestro dispositivo, nos abrirá un pop-up visualizando nuestra cuenta de Google+ y podrá seleccionarse directamente sin necesidad de escribir correo y contraseña de la cuenta.

Al iniciar sesión se mantendrá siempre la sesión iniciada aunque se cierre la aplicación, ya que es mucho más cómodo para el usuario, en lugar de tener que iniciar sesión cada vez que se abra nuestra aplicación.

Una vez iniciada la sesión hemos implementado una pantalla principal, en la cual se mostrará que dispositivos tienes añadidos, un botón para añadir dispositivos, un panel lateral, que tendrá como cabecera la foto de perfil, el correo y el nombre asociada a la cuenta de Google+. En el cuerpo del panel tendrá también todos los dispositivos añadidos, para que en cualquier momento puedas seleccionar uno, sin necesidad de tener que pasar siempre por la pantalla principal.

Habrá también un botón de Logout, se encargará de cerrar sesión tanto en Firebase Authentication, como en nuestra aplicación.

Hemos decidido que al pulsar en el botón de añadir dispositivos que aparece en nuestra pantalla principal, se introduzca un ID que será único y que vendría junto con nuestros dispositivos. Una vez añadido, se creará automáticamente un botón nuevo en el panel lateral con el dispositivo nuevo añadido.

Trata sobre la realización del trabajo en sí. En esta parte se describe lo que se ha hecho, cómo se ha llevado a cabo, por qué se ha hecho así y no de otra manera, qué materiales o herramientas ha sido necesario utilizar, qué metodología de trabajo y validación se ha utilizado, etc.

## Herramientas tecnológicas

### Java

El lenguaje escogido para el desarrollo de la aplicación será Java. Es el lenguaje que más hemos usado durante el curso que hemos realizado y por el cual estamos más familiarizados.

Java es lenguaje de programación orientado a objetos, lo que facilitará bastantes la creación y utilización de clases objetos.

Para poder programar en Java, lo primero que necesitaremos será un JDK (Java Development Kit), que contiene la JVM (Java Virtual Machine) y diferentes librerías.



### Android Studio

La realización de la aplicación se hará en el entorno Android Studio.

Es un entorno de desarrollo de software libre destinado a programar aplicaciones para la plataforma Android.

Es una herramienta bastante útil, nos permite hacer el diseño de la aplicación y a su vez tener una vista previa de los cambios que vamos realizando en el diseño. Nos proporciona una lista de widgets y diseños que podemos arrastrar directamente en el editor. Otra de las facilidades es dar la posibilidad de poder mostrar la visualización de la aplicación de forma horizontal o vertical.

Otra de las características de agradecer por parte de este entorno, es el editor de código inteligente que integra, a diferencia de otros entornos como Eclipse, Android Studio nos permite ir escribiendo código y a su vez nos va mostrando diferentes opciones de código relacionadas a lo que escribimos. Soporta lenguajes de programación como Java, Kotlin y C / C++.

Tiene un emulador de dispositivo Android el cual nos brinda la posibilidad de ir ejecutando nuestra aplicación cada vez que hayamos realizado cambios y queramos comprobar su correcto funcionamiento. Esta opción es bastante cómoda, ya que no nos haría falta tener un dispositivo Android real conectado a nuestro portátil cada vez que queramos ejecutar nuestra Aplicación en la fase de desarrollo. Estas características harán que nuestro desarrollo sea más rápido y productivo.

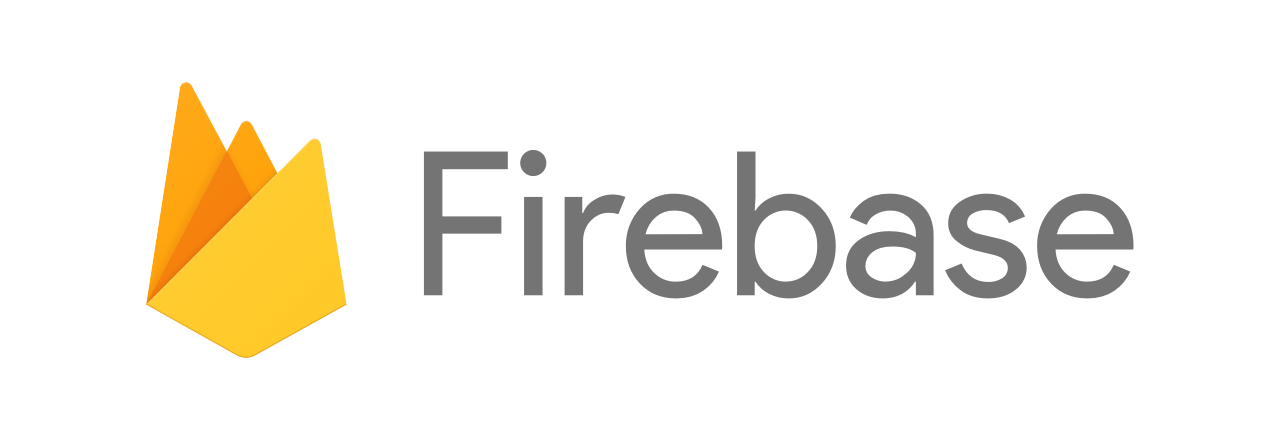


### Firebase

Firebase es una plataforma de desarrollo móvil en la nube, gratis. El gran atractivo de esta tecnología es que dispone de una gran variedad de productos de manera gratuita, y ofrece soporte para los más comunes IDEs, plataformas y sistemas operativos. En nuestro proyecto usaremos concretamente dos: Realtime Database y Authentication.

Uno de ellos para almacenar los datos de los sensores y el otro para gestionar los usuarios que acceden a la aplicación.

Otra ventaja es que es realmente escalable, por lo que no tendríamos problemas a la hora de llevar nuestro producto a más personas, o adaptarlo en caso de que exista una gran demanda. Y en cualquier momento podríamos implementar más servicios de Firebase en el proyecto sin necesitar rehacerlo por completo.



### Arduino

Arduino es una compañía Open Source y Open Hardware, así como un proyecto y comunidad internacional que diseña y manufactura multitud de dispositivos capaces de controlar objetos del mundo real mediante la programación.

Hemos elegido esta plataforma para desarrollar el dispositivo físico debido a la gran comunidad que tiene, y al gran número de placas con microcontroladores distintos que nos ofrece. Por ello, hemos elegido el ESP32, una placa que incluye un microprocesador con dos núcleos y capaz de realizar tareas un tanto pesadas como puede ser la conexión a Firebase mientras monitoriza los sensores.

Otra de las razones por las que elegimos esta plataforma es porque pretendemos que nuestro producto sea lo más personalizable por el usuario posible, consiguiéndolo por completo gracias a este tipo de microcontroladores.



### C++

Cabe destacar que el entorno de desarrollo que utilizaremos con el ESP32 será el de Arduino, que está basado en el lenguaje de programación C++.

A pesar de ser un lenguaje diseñado en el año 1979, permite una programación orientada a objetos, lo cual es realmente útil y cómodo a la hora de trabajar con Arduino. Tenemos que remarcar que incluimos este lenguaje como tecnología usada, pero en realidad Arduino usa una versión modificada del mismo, haciéndolo más simple y comprensible por los usuarios que lo utilizan.

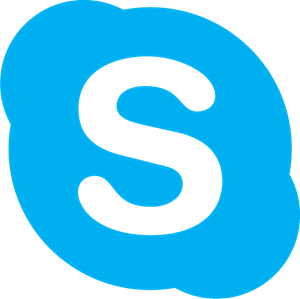


## Planificación

El principal canal de comunicación que usaremos será Whatsapp, por el cual indicaremos que tareas tenemos pendientes por realizar y cuales están ya realizadas.

Otro canal de comunicaciones que usaremos con frecuencia será Skype, en el cual haremos pequeñas sesiones de llamadas para detallar lo que hemos hecho hasta el momento, errores encontrados y posibles mejoras.

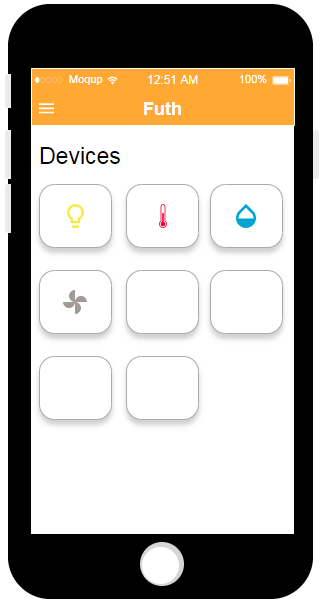
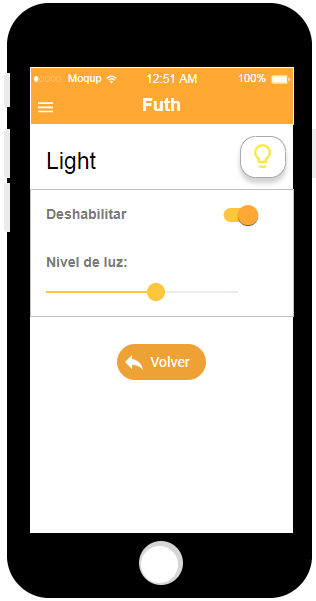
La comunicación con nuestro tutor de proyecto será mediante Slack, en la cual tendremos un canal privado que estará compuesto por los integrantes del proyecto y el tutor. Será por este canal por el que nos mantendremos informados de las entregas y reuniones por videoconferencia para informar de los avances del proyecto.

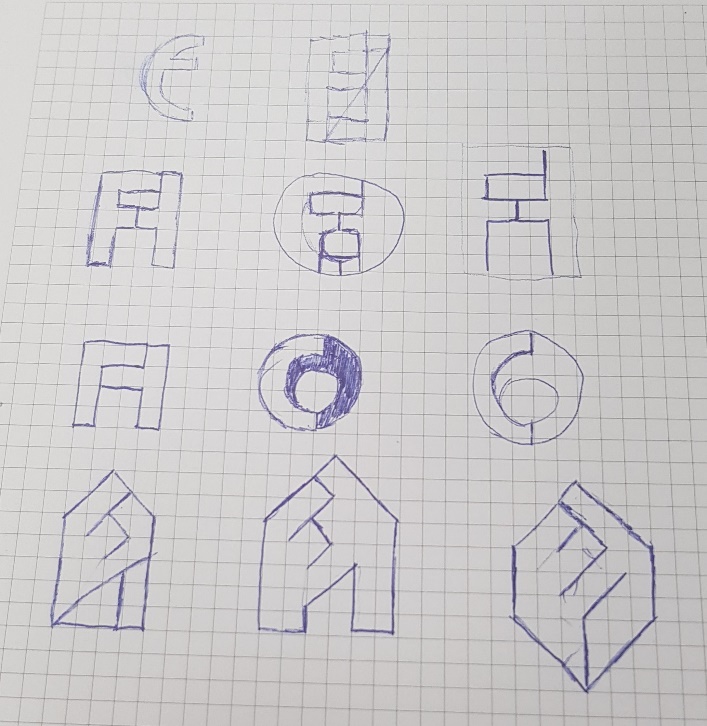


## Descripción del trabajo realizado

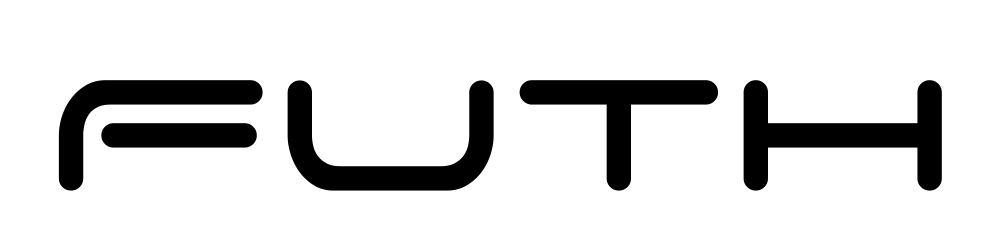
El primer paso de nuestro proyecto fue desarrollar un mockup, para tener una visión global de lo que pretendíamos conseguir y una base sobre la que empezar a trabajar.

Una vez realizamos el boceto de la aplicación, nos pusimos a intentar diseñar el logo de la misma. A continuación dejamos unos bocetos iniciales.

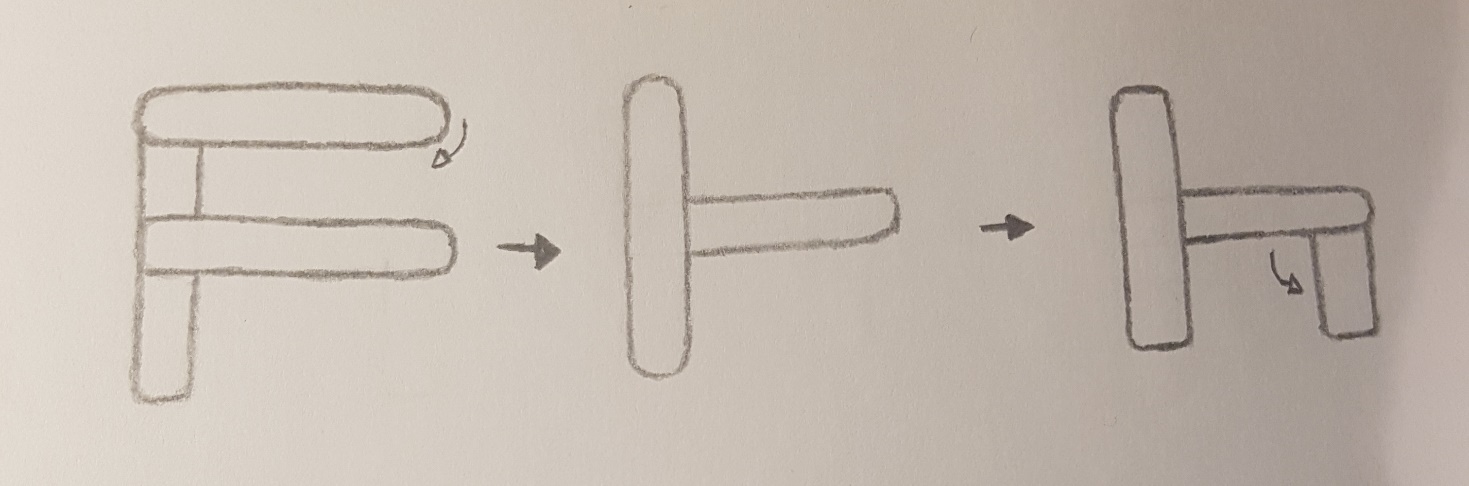


Finalmente derivamos la decisión en elegir un logo menos complejo, acorde con el estilo “material” de la app. Obtuvimos un logo estilo icono y otro de texto para aplicar en títulos y demás.





De igual manera, desarrollamos un boceto de cómo debía transcurrir la animación que posteriormente haríamos con el logo.

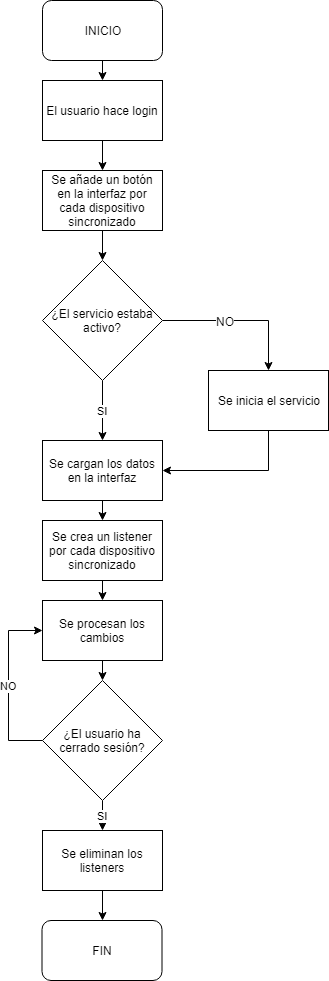


### Documentación técnica

En este apartado expondremos el trabajo que hemos realizado sobre el modelado de datos que aplicaremos sobre nuestro proyecto.

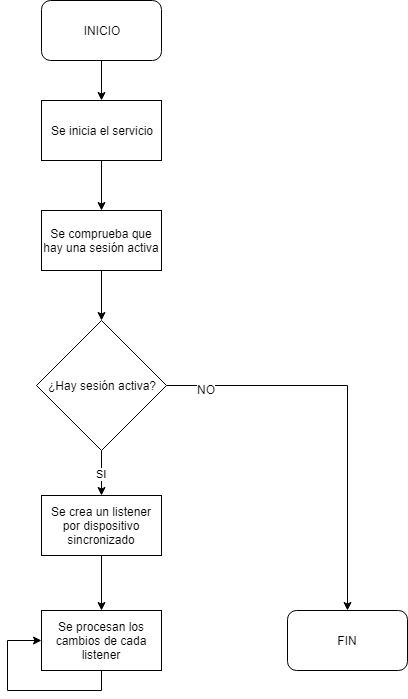
#### Diagramas de flujo

##### Flujo aplicación Android

Establecimos con el siguiente diagrama cuál debía ser el flujo de funcionamiento de la app en general.

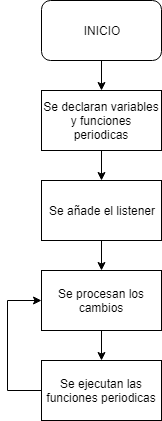
##### Flujo servicio Android

A continuación se expone el flujo con el que debería funcionar el servicio en segundo plano que tendrá la app para poder mostrarle notificaciones al usuario.



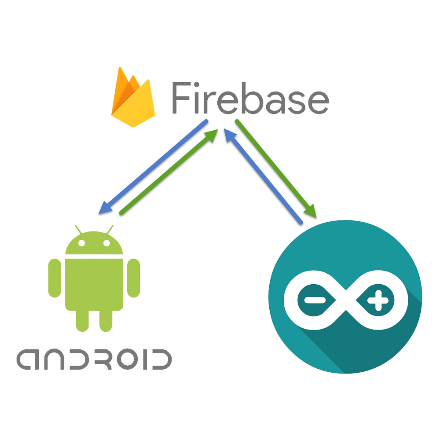
##### Flujo Arduino

Igualmente, de la siguiente manera deberá funcionar nuestro dispositivo. Cabe destacar que no se indica finalización debido a que el trabajo del dispositivo ha de ser constante e ininterrumpido.



### Esquemas

En el siguiente gráfico hemos detallado la manera en la que nuestro dispositivo y la app se deberán comunicar entre sí, dando como resultado una comunicación p2p. Hay que destacar que Firebase tiene una forma peculiar de trabajar, muy similar (por no decir idéntica) al protocolo MQTT, por lo que nosotros seguiremos las directrices de buenas prácticas de dicho protocolo.



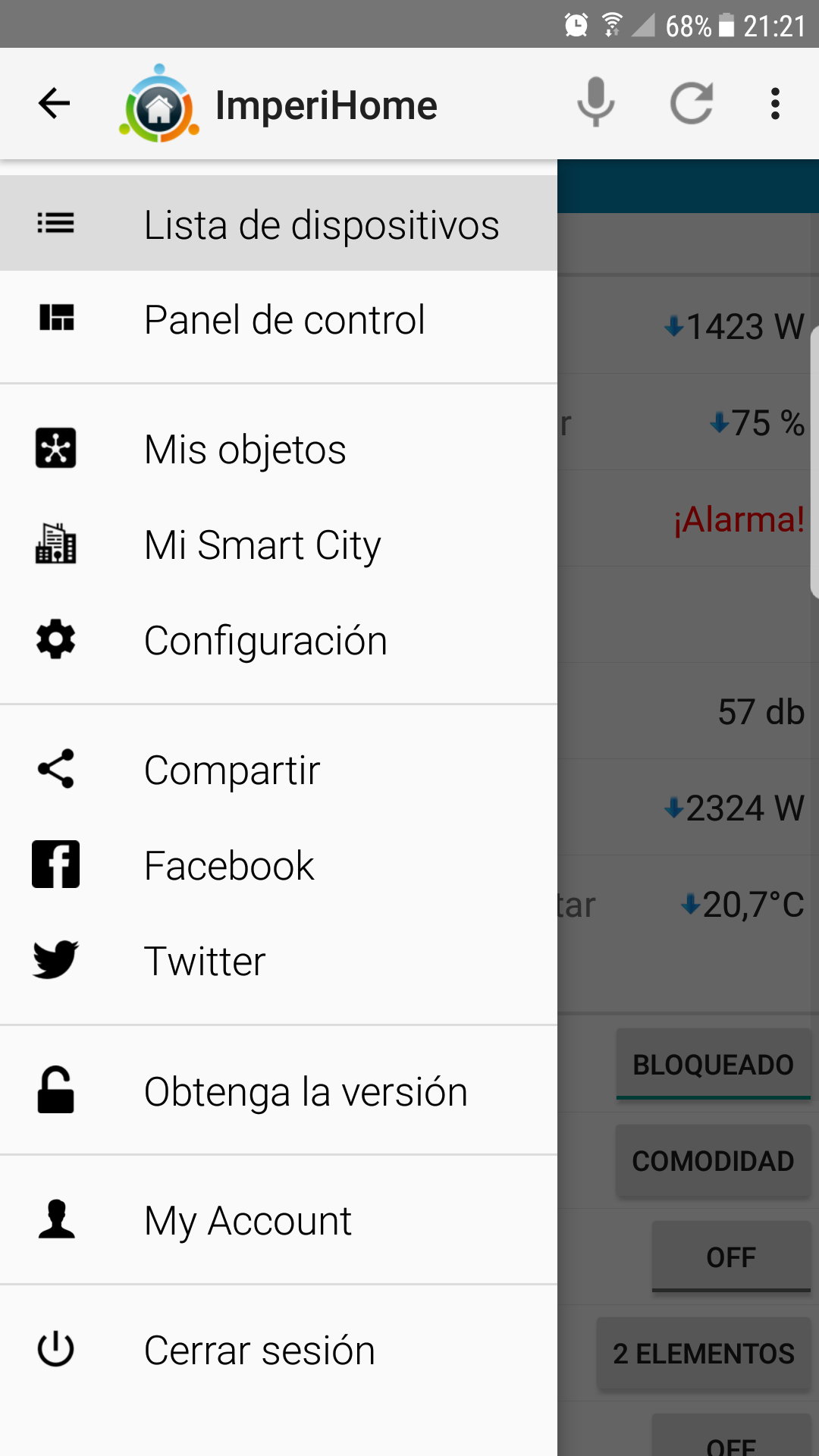
# Bibliografía y Webgrafía

* [Arduino](https://www.arduino.cc/)
* [¿Qué es Firebase?](https://elandroidelibre.elespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html)
* [Productos Firebase](https://firebase.google.com/products/?hl=es-419&gclid=Cj0KCQjw0a7YBRDnARIsAJgsF3OLHnSP6gsxEHp0JKOdHHQUjMK_uVP-APz4Nzvux9Xr1eBoHig1VIAaAi10EALw_wcB)
* [Firebase](https://firebase.google.com/)
* [¿Qué es MQTT?](https://geekytheory.com/que-es-mqtt)
* [MQTT](http://mqtt.org/)
* [C++ Wikipedia](https://es.wikipedia.org/wiki/C%2B%2B)
* [Arduino Wikipedia](https://es.wikipedia.org/wiki/Arduino)

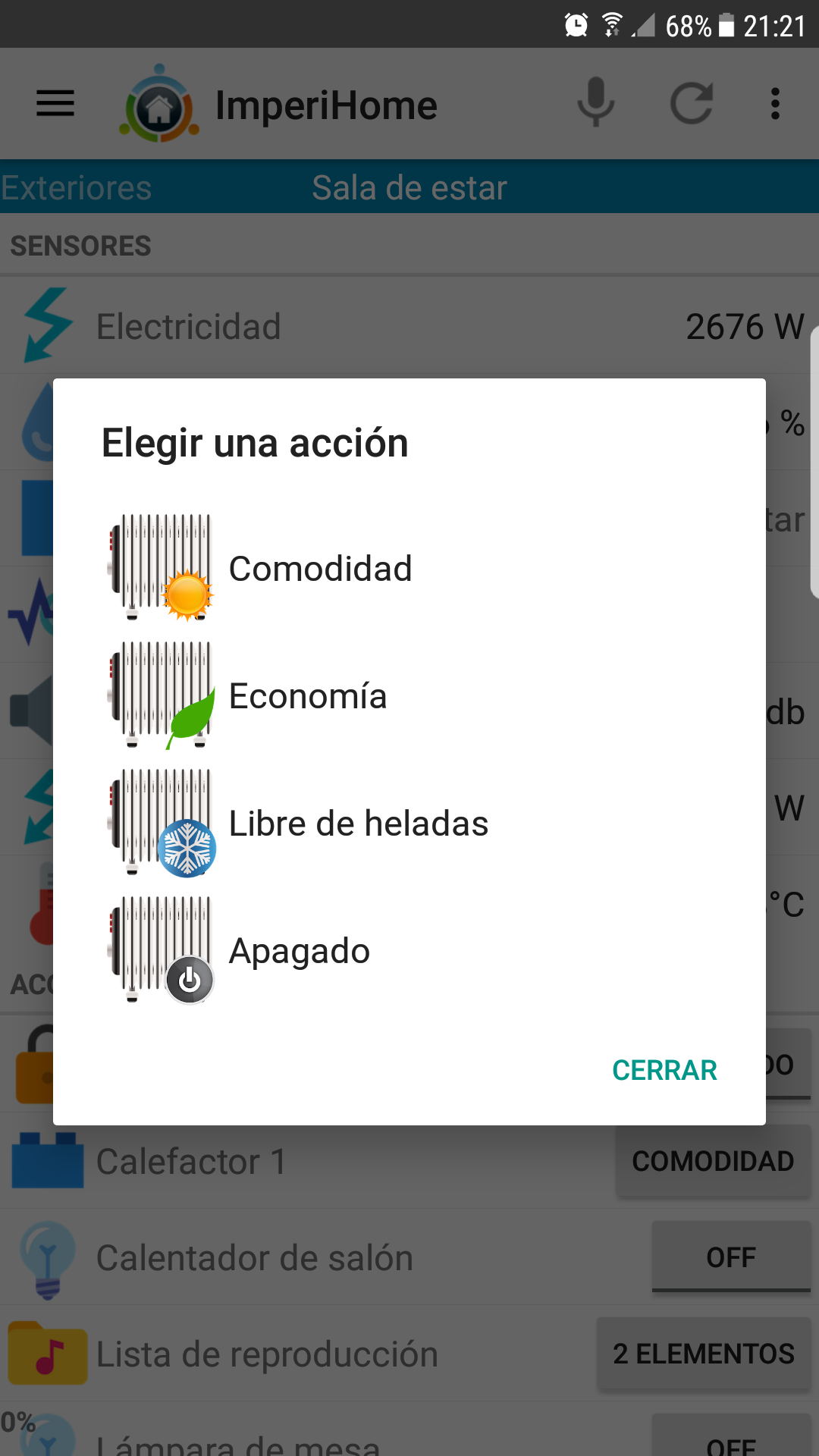
# Anexos

## Apps domótica

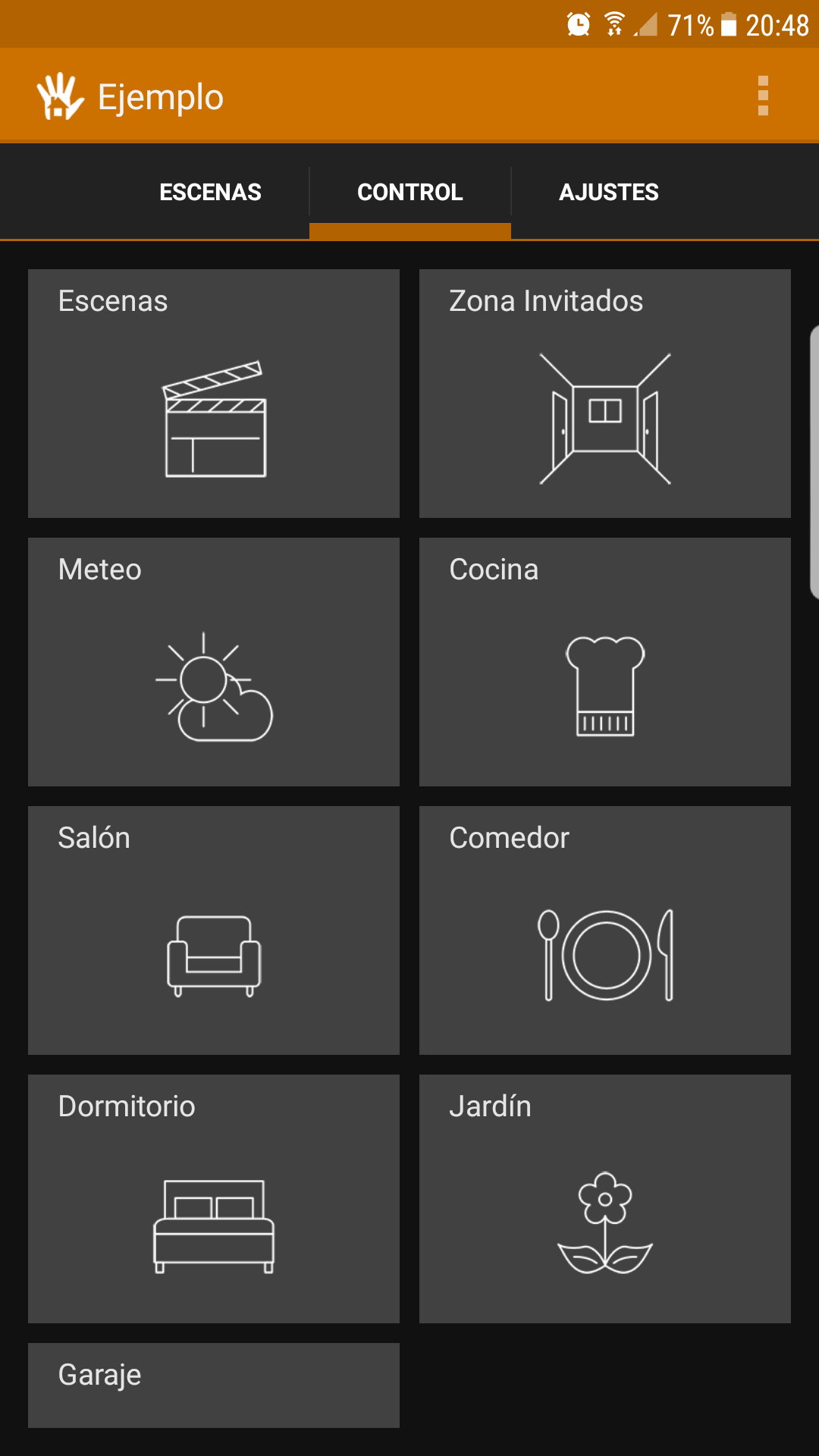
### Imperihome

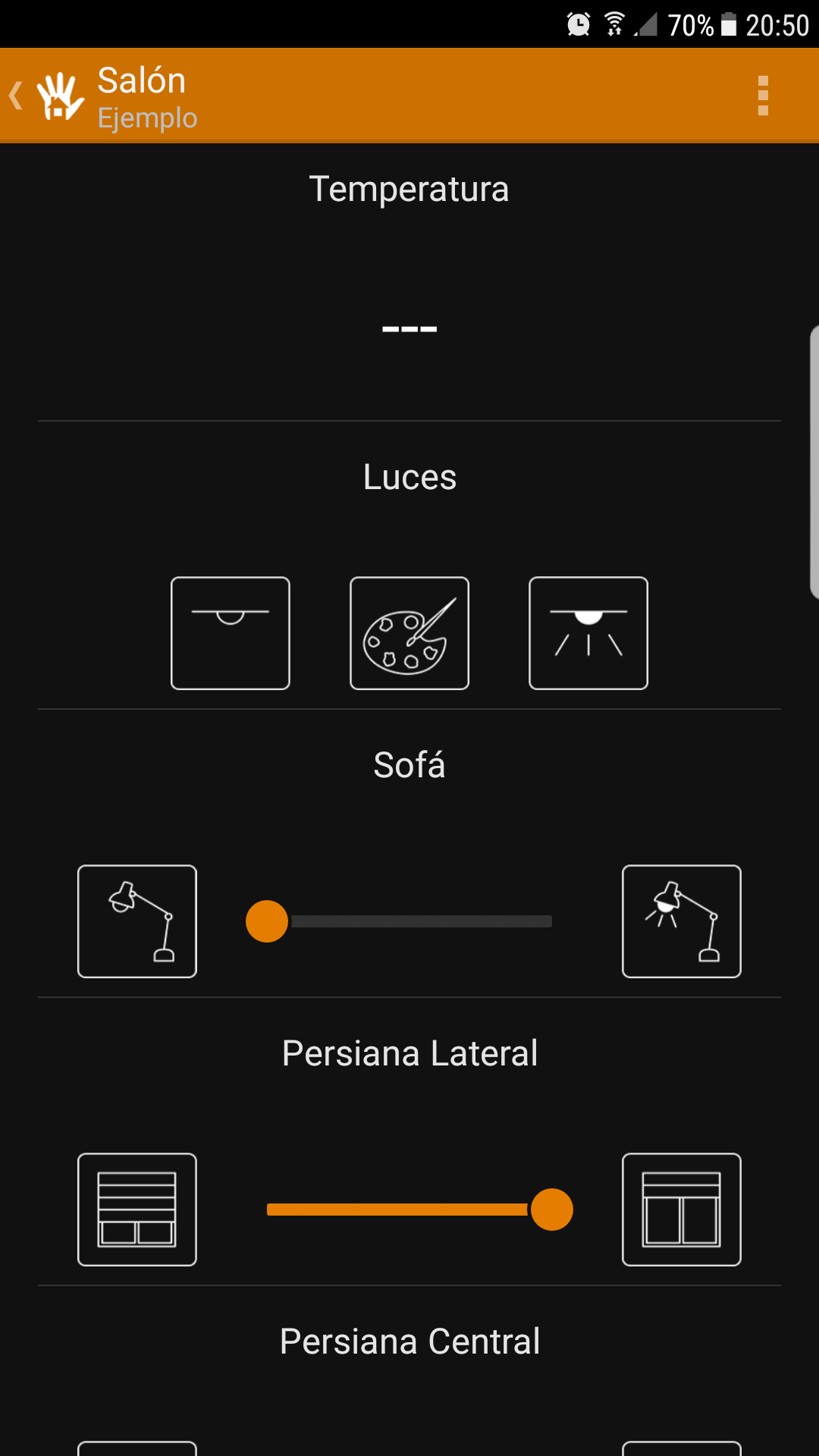






### Houseinhand KXN





### TaHoma by Somfy

