



ugr

Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

**Interacción móvil en sistemas de automatización
domótica**

Autor:
Javier Izquierdo Vera

Director:
Prof. Dr. Juan Antonio Holgado Terriza



Escuel Técnica Superior de Ingenierías informática y de
Telecomunicación

Granada, septiembre de 2019

Interacción móvil en sistemas de automatización domótica

Javier Izquierdo Vera (alumno)

Palabras clave: Domótica, Android, Sistemas de iluminación, Emociones, Estados de ánimo, Computación ubicua, Hogar, Experiencias de usuario.

Resumen

A medida que se van incorporando distintos tipos de dispositivos en el hogar digital conectado (e.g., bombillas inteligentes, termostatos inteligentes) junto con los dispositivos corporales (e.g., pulseras o relojes inteligentes) y los dispositivos móviles personales (e.g., Smartphone) crece la necesidad de proporcionar nuevas funcionalidades derivadas de la colaboración del conjunto de dichos dispositivos. Tareas como la regulación de la temperatura en el hogar, el consumo energético o el confort se pueden realizar de una forma proactiva facilitando la cooperación entre los diferentes dispositivos [43]. En este trabajo estamos interesados en explorar en las técnicas y acciones que se pueden realizar en el entorno del hogar para establecer ambientes en cuanto a iluminación y sonido. Este proyecto consiste en la elaboración de una aplicación para Android que permita controlar el sistema de iluminación del hogar de un modo intuitivo para el usuario, y que facilite recrear distintas ambientaciones basadas en la iluminación y el sonido por medio de recetas; con el objetivo de que sirva de aplicación base para un proyecto de investigación que busca tratar de predecir las emociones de los habitantes de la casa para adaptar el ambiente de un modo que favorezca las emociones positivas [39, 40, 37].

Yo, **Javier Izquierdo Vera**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 76589384M, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Javier Izquierdo Vera

Granada a 4 de agosto de 2019

Prof. Dr. Juan Antonio Holgado Terriza (tutor), Profesor del Área de Lenguajes y Sistemas Informáticos del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado **Interacción móvil en sistemas de automatización domótica**, ha sido realizado bajo su supervisión por **Javier Izquierdo Vera (alumno)**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 5 de agosto de 2019.

El director:

Juan Antonio Holgado Terriza

Índice

1. Introducción	6
1.1. Motivación	6
1.2. Objetivos del proyecto	6
1.3. Estructura del trabajo	7
2. La Ambientación y las emociones	9
2.1. Modelos emocionales	9
2.2. Color, sonido, y luz	9
2.3. Ambientación en el hogar	10
2.3.1. Sistemas de iluminación	11
2.3.2. Altavoces	12
3. Desarrollo de la plataforma Lights	13
3.1. Descripción del sistema	13
3.2. Público objetivo y Personas	13
3.3. Análisis de aplicaciones similares	19
3.4. Modelado de requisitos	22
3.4.1. Requisitos funcionales	22
3.4.2. Requisitos no funcionales	24
3.4.3. Requisitos de información	25
3.5. Actores y Casos de uso	26
3.5.1. Actores	26
3.5.2. Casos de uso	27
3.6. Interfaz de usuario	53
3.6.1. Referencias	54
3.6.2. Wireframes y navegabilidad	58
3.6.3. Diseño gráfico	61
3.7. Diagrama de flujo del sistema	62
3.8. Modelo conceptual	64
3.9. Arquitectura del sistema	66

3.10. Implementación	68
3.10.1. Tecnologías, herramientas y servicios utilizados	68
3.10.2. Proyectos de luz	73
4. Dirección y gestión del proyecto	79
4.1. M étodología de trabajo, planificación y herramientas	79
4.1.1. Conclusiones	81
4.2. Estimación de costos	82
4.2.1. COCOMO II: Input	83
4.2.2. COCOMO II: Resultados	84
4.2.3. COCOMO II: Análisis de los resultados y conclusiones	84
5. Publicación y mercado	85
5.1. Monetización	85
5.2. Versiones	86
5.2.1. Primera aproximación	86
5.2.2. Segunda aproximación	88
5.2.3. Roadmap	90
6. Manual de usuario	92
6.1. Configuración del sistema Philips Hue	92
6.2. Configuración del altavoz	93
6.3. Configuración de la aplicación	94
6.3.1. Actividades de error	98
6.3.2. Control de la iluminación de una habitación	99
6.3.3. Proyectos de luz	105
7. Conclusiones	108
7.1. Análisis de los objetivos planteados	110
7.2. Trabajos futuros	111
7.3. Experiencia personal	112
8. Anexo I: Herramientas y tecnologías utilizadas durante el proyecto	121

1. Introducción

1.1. Motivación

A día de hoy el mundo ha cambiado y casi toda la gente tiene un ordenador en su propio bolsillo, hay millones de aplicaciones que nos proporcionan servicios, entretenimiento, mejoras en nuestro estilo de vida o nos facilitan tareas como la comunicación, la organización, etc; la tecnología cada vez es más accesible y nos brinda más oportunidades. Cada vez es más frecuente encontrar en los hogares aparatos inteligentes que sirven para mejorar el estilo de vida; por ejemplo, los asistentes de voz como Google Home o Alexa, los sistemas de iluminación, enchufes inteligentes, aspiradoras, cámaras, etc.

Además del gran número dispositivos con los que se inunda el hogar, están apareciendo también aplicaciones que nos facilitan el control de dichos dispositivos, e incluso aplicaciones nuevas que integran el funcionamiento de varios dispositivos a la vez, en lo que se ha denominado el Hogar Conectado en el contexto del Internet de las Cosas. Dentro de este contexto, están apareciendo un sinfín de aparatos en nuestro día a día que estamos comenzando a integrar en diferentes pasarelas residenciales y en sistemas cloud que dejan muchas posibilidades a nuestro alcance para explotarlos y crear herramientas basadas en dispositivos móviles que mejoran su utilidad y la experiencia de los usuarios. El dispositivo móvil se está convirtiendo en el elemento fundamental a través del cual el usuario puede percibir lo que el hogar conectado puede proporcionarle (un mejor confort, una mayor seguridad, una mejor climatización) y puede controlar todo el ecosistema.

La incorporación de dispositivos capaces de leer el estado del entorno del hogar y de dispositivos actuadores capaces de modificar el estado del entorno del hogar proporcionan una nueva gama de aplicaciones que pueden personalizarse y adaptarse a los habitantes del hogar. Incluso puede realizar tareas de forma proactiva para garantizar que el habitante o habitantes se encuentran comodos y seguro en el ámbito del hogar.

Este proyecto nace con la idea aprovechar estos sistemas para tratar de mejorar el estado de ánimo de los usuarios, o de fomentar las emociones positivas, intentando exprimir aún más estos sistemas y hacer que funcionen en conjunto para ofrecer mayor valor a los usuarios.

1.2. Objetivos del proyecto

Con el objetivo de influir en el estado de ánimo de los usuarios se recrearán distintas ambientaciones en el hogar que puedan asociarse con sentimientos positivos o mejorar un estado de excitación. Para ello se hará uso de la iluminación y el sonido, tratando de brindar al usuario una herramienta no intrusiva con la que poder controlar la ambientación del hogar del modo que desee, sin perder el control de lo que está ocurriendo en ningún momento.

En este proyecto se pretende llevar a cabo el desarrollo de una herramienta para dispositivos móviles Android capaz de controlar el sistema de iluminación Philips Hue, y que permita la elaboración y ejecución de recetas que recreen una ambientación mediante luz y sonido más confortable para los usuarios, utilizando los altavoces que haya disponibles en el hogar.

Para llevar a cabo dicho proyecto se tendrán que cumplir asimismo una serie de objetivos específicos:

- Estudio de los factores que caracterizan el estado de ánimo de una persona.
- Analizar de qué modo se puede medir el estado de ánimo de una persona mediante un sistema de cómputo.
- Explorar los diferentes sistemas de iluminación del mercado, y realizar un análisis comparativo.
- Revisar las aplicaciones de control de iluminación ya publicadas para dispositivos Android, para averiguar qué características ofrecen a los usuarios y cómo las llevan a cabo.
- Averiguar y utilizar las herramientas y servicios de Android que mejor se adapten a nuestro propósito.
- Realizar un trabajo de extracción y análisis de requisitos, y de elaboración de casos de uso que nos permita especificar todo lo posible las funciones que tendrá esta versión de la aplicación.
- Intentar averiguar cómo proporcionar una experiencia de usuario satisfactoria mediante una navegabilidad y diseño de interfaz adecuado, buscando y analizando otras referencias y siguiendo las guías de estilo.
- Llevar a cabo una correcta gestión y organización del proyecto.
- Aplicar buenas prácticas de ingeniería en el proceso de desarrollo de la aplicación.
- Desarrollar un prototipo de la aplicación, y probarlo con una muestra de usuarios preseleccionada.

1.3. Estructura del trabajo

En este trabajo se tratan los siguientes capítulos:

- **La Ambientación y las emociones:**

- Análisis de los factores que caracterizan el estado de ánimo de una persona. Buscando un modelo emocional que permita medir esto mediante un sistema de cómputo.

- Análisis de cómo afectan el colores, el sonido y la iluminación a las personas y a su estado de ánimo.
- Comparativa de los sistemas de iluminación disponibles en el mercado y conclusión acerca del motivo de utilizar Philips Hue.

■ **Ingeniería de Software y diseño de la aplicación:**

- Descripción del sistema.
- Análisis de las aplicaciones similares disponibles en el mercado y conclusiones.
- Especificación del público objetivo de la aplicación.
- Extracción de requisitos y especificación de los casos de uso del sistema.
- Análisis de referencias en cuanto a la interfaz gráfica y experiencia de usuario se refiere, y elaboración de wireframes para especificar el diseño de la interfaz; además de especificaciones sobre el apartado gráfico.
- Detalles sobre el diseño y la implementación. Justificación de decisiones tomadas respecto al uso de servicios externos, Android, y SDK.

■ **Dirección y gestión del proyecto:**

- Comentarios acerca de cómo se ha llevado a cabo la organización del proyecto y las conclusiones obtenidas.
- Comentarios y conclusiones acerca de elaboración de un presupuesto, y de cómo se ha intentado aplicar el modelo COCOMO II.

■ **Publicación y mercado:**

- Búsqueda del sistema de monetización adecuado para la aplicación elaborada.
- Versiones del proyecto elaboradas, capturas de pantalla y publicación en Google Play.

■ **Manual de usuario:**

- Especificación sobre el despliegue del sistema.
- Documentación de las funciones de la aplicación.

■ **Conclusiones:** Repaso de los objetivos propuestos, los objetivos satisfechos y las conclusiones obtenidas.

2. La Ambientación y las emociones

2.1. Modelos emocionales

A lo largo del tiempo se han ido elaborando distintos modelos emocionales con los cuales se pretende poder medir las emociones y alcanzar un mayor entendimiento de ellas. Dentro del conjunto de modelos se han intentado establecer modelos emocionales de referencia para sistematizar el proceso [41].

El modelo seleccionado que se utilizará como referencia para la parte inteligente de la aplicación es “Circumplex Model of Emotion”, también conocido como “Circumplex Model of Affect” ideado en 1980 por James Russell. Este modelo entiende las emociones como sensaciones relacionadas unas con otras y ambiguas, y las representa como un punto en un espacio circular bidimensional, de modo que su proximidad a cada uno de los extremos denota el carácter de la emoción y la intensidad. Este modelo permite que la representación de las emociones sea computable (ejemplo: FEELTRACE tool [36]), además de ser fácil de entender para los humanos [41]. Por lo que para las siguientes etapas del desarrollo será útil para poder medir las emociones en base a este sistema.

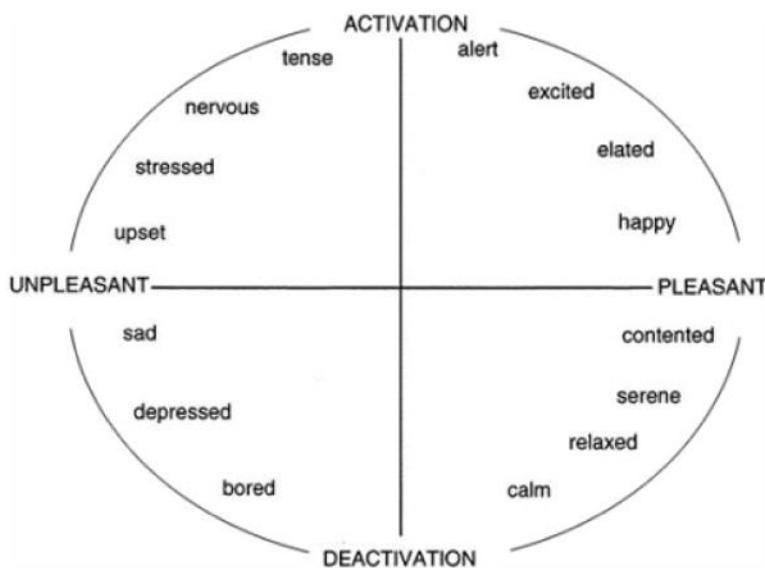


Figura 1: Circumplex Model of Emotion (Russel 1980) [38]

2.2. Color, sonido, y luz

Es bien conocido que los ambientes en el hogar mediante el uso de luz, color y sonido tienen una influencia importante en las emociones y estados de ánimo de las personas que habitan en dicho hogar. Por lo que cabe preguntarse cómo estos nos afectan realmente.

En la actualidad hay algunos estudios que proporcionan evidencias sobre efectos de colores concretos en las emociones (como el color azul, que favorece

la relajación [42]). Aunque podemos llegar a la conclusión de que, por ejemplo, en España consideramos el color rojo como algo intenso o equivocado, el verde como sensación de calma y de que algo está correcto, o la felicidad y la alegría con los colores vivos como el naranja o el amarillo. Esto es debido a que tendemos a asociar los colores con sensaciones en base a nuestras experiencias previas. Por lo cual, los efectos que causan los colores varían en función de la persona y sus vivencias. Sin embargo, sí que podríamos decir que tiene un factor cultural, ya que dentro de la misma cultura se crean patrones en base a los colores dotándolos de este modo un significado que se asimila de forma inconsciente.

Si tratamos las emociones de las personas utilizando colores, por lo tanto, debemos de tener en cuenta que los efectos variarán en función de la cultura y en función de la persona. Por lo que no podemos suponer que un color va a causar un efecto concreto, sino más bien deberíamos intentar aproximarnos y aprender de cada usuario los efectos que le causan.

Sin embargo, en el caso del sonido sí que podemos afirmar que tiene unas consecuencias directas sobre las personas. Aunque hay muchas preferencias y gustos en cuanto a ritmos y música, sí es posible inducir sentimientos como la calma con un ritmo pausado, o todo lo contrario con un ritmo rápido. Una vez más, el sonido también tiene un componente basado en nuestras experiencias previas y en las asociaciones que podemos hacer con lugares o acciones. Con lo cual, si queremos simular una ambientación de bosque (por ejemplo) será más fácil para el usuario relacionar el ambiente con ese lugar si el sonido que se reproduce tiene sentido en ese lugar y se puede asociar con ese lugar.

En cuanto a la iluminación sí que está demostrado que nos afecta de muchos modos diferentes, ya que afecta a los ritmos circadianos y a los relojes biológicos de nuestro cuerpo. Cuando hay menos luz, nuestro cerebro produce más melatonina, lo cual nos provoca sueño. Y por el contrario, una iluminación más intensa nos hace estar más activos y despiertos. La luz puede afectar a nuestra conducta y a nuestra salud [17]. Esto nos afecta hasta el punto de que algunas personas ciegas suelen tener dificultades para dormir [35]. Lo cual, nos lleva a la conclusión de que hay que tener en cuenta la iluminación en nuestro sistema, y de que ajustarla de forma correcta puede ayudar a mejorar el estilo de vida de los usuarios.

2.3. Ambientación en el hogar

Los dispositivos del hogar pueden ayudar a modificar la ambientación del hogar. En esta sección vamos a analizar los dispositivos que vamos a utilizar para modificar la ambientación de una casa.

2.3.1. Sistemas de iluminación

A día de hoy, existen muchos sistemas de iluminación diferentes en el mercado, y cada vez es más común poder encontrarlos en casa debido a que son más accesibles. Los precios se van abaratando a medida que el internet de las cosas está penetrando cada vez más en nuestros hogares y cada vez es más sencillo encontrarlos en cualquier tipo de tienda habitual, y no únicamente en tiendas especializadas.

Estos sistemas tienen en común la posibilidad de regular la iluminación de los dispositivos de luz. Algunos de ellos, ni siquiera necesitan un gateway, sino que es posible comunicarse directamente con el dispositivo. Los hay incluso que tienen batería propia y no es necesario conectarlos a la red eléctrica (como AVANTEK).

El dispositivo ideal depende de las necesidades de cada persona. Si se desea mayor portabilidad y comodidad para un uso concreto, una buena opción podría ser la comentada en el párrafo anterior, AVANTEK. Si se necesita lo mismo, pero no nos importa la opción de batería, LOHAS se encarga de ofrecérnoslo por la mitad de precio, incluyendo muchas más horas de vida y algunos lúmenes más que las opciones más costosas del mercado. Por el contrario, si nuestra intención es controlar toda la iluminación de una casa, o una sala con diversas luces, tal vez la mejor opción sea alguna de las que incluyen un gateway, ya que la probabilidad de fallo en la conectividad es menor y parece más fácil ordenarlas y agruparlas.

En nuestro caso hemos elegido las bombillas de Philips: Philips Hue. Esto es debido a que han sido de las primeras en aparecer en el mercado, y ofrecen una API muy completa que nos permite interactuar con ellas a nuestro gusto. Existe un mercado de aplicaciones bastante grande para este tipo de bombillas, con un gran número de usuarios, pero a pesar de ello ninguna parece innovar u ofrecer posibilidades que no se hayan ofrecido ya en otros formatos. Nuestro objetivo será centrarnos en este sistema de iluminación pero intentando generalizar, de forma que pueda ser adaptado a otros sistemas similares.

Se ha elaborado la siguiente tabla comparativa [12, 33, 31, 32, 25, 34, 5, 19], con algunas opciones del mercado:

	GE Link	Belkin WeMo LED Lighting	TCP Connected	Trådfri	Philips Hue	Xiaomi Yeelight	AVANTE K Bombillas de Wi-Fi	LOHAS LED
Precio	\$15 /\$25 (bombillas) + \$69 (hub)	80€ (2 bombillas + bridge) / 1 bombilla 30€	\$190 (3 bombillas + control remoto) / 1 bombilla \$12.49	79.99€ (pack completo con dos bombillas) / 1 bombilla → 9.99€	164.90€ (3 bombillas de colores + bridge)	28.99€ (una bombilla)	45,99€ (una bombilla)	22,99€ (una bombilla)
Conectividad	Wi-Fi - Z-Wave	Wi-Fi - ZigBee	Wi-Fi / Control remoto	ZigBee Light Link	Wi-Fi / ZigBee		Wi-Fi / Se puede usar manualmente	Wi-Fi
Bridge	Sí. Un hub	Sí. Usa un bridge.	Sí.	Sí. Un hub.	Sí, un bridge.	No.	No.	No.
Colores	No	No	No	No	Sí (opcional)	Sí	Sí.	Sí.
Lúmenes	800	800	600	1.000	800	600		810
Horas de vida		25.000	15.000	25.000	25000	9.000		30.000

Figura 2: Comparativa entre sistemas de iluminación del mercado

Philips Hue nos proporciona todo lo que necesitamos para elaborar este proyecto, ya que nos permite modificar la tonalidad del color y el brillo y nos proporciona un gateway (con las ventajas que comentábamos anteriormente); además, tiene un número de horas de vida y lúmenes aceptable y superior a otros sistemas que cuentan con las características deseadas.

2.3.2. Altavoces

Para emitir sonido utilizaremos sistemas como Google Home o Chromecast, los cuales también están cada vez más presentes en los hogares, siendo también dispositivos pensados para integrarse en el hogar. Además, en muchos casos nos proporcionará mejor calidad de audio que el altavoz del dispositivo Android. Así que, trataremos de utilizarlos en modo conjunto con el sistema de iluminación para lograr nuestro cometido y exprimir las posibilidades que nos ofrecen siempre que sea posible.

3. Desarrollo de la plataforma Lights

3.1. Descripción del sistema

Esta aplicación está dirigida a personas de todas las edades, de modo que esté a disposición de cualquier inquilino de la casa que disponga de este sistema de iluminación, por lo tanto esto refuerza la idea de que debe de ser una aplicación fácil de utilizar que ofrezca únicamente la información necesaria para el usuario. Podrá ser utilizada por niños o por personas menos acostumbradas a utilizar aparatos tecnológicos, a modo de mando a distancia para regular la iluminación.

La aplicación podría ser utilizada con bastante frecuencia, ya que permitirá al usuario regular de forma manual la iluminación de la casa. Por lo tanto, se espera que pueda ser utilizada para acciones tan sencillas y cotidianas como encender o apagar la luz de una habitación.

Además de lo anterior, la aplicación también busca ofrecer experiencias agradables a personas que quieran relajarse, ambientar el hogar de un modo concreto, entretenerse un rato, o simplemente experimentar con las posibilidades que ofrece este sistema. Por lo que podría llegar a usarse para ambientar la habitación de forma adecuada para la lectura, para crear un ambiente de relajación tras una jornada de trabajo, o para simular un amanecer a la hora de levantarse. La herramienta también está dirigida a personas que quieran ir más allá y estén dispuestas a elaborar sus propios proyectos de luz.

Un proyecto de luz es una receta que permite indicar al sistema el color, brillo y sonido que se reproducirá en una habitación, de modo que pueda variar en función del tiempo. Los proyectos de luz pretenden simular una ambientación concreta. Más adelante se explicará con más detalle.

3.2. Público objetivo y Personas

Para tener presente a posibles usuarios objetivo de la aplicación, se ha utilizado la técnica *Personas* para determinar las necesidades de los potenciales usuarios. Esto nos permite ser más conscientes de las necesidades específicas de los usuarios a los que pretendemos dirigirnos, y así tenerlos siempre en cuenta a la hora de diseñar el producto. Cada una de las personas creadas nos da la idea de un conjunto del público objetivo de la aplicación, brindándonos un punto de vista más realista y con el que valoremos características que podríamos pasar por alto. Esta técnica también nos permite empatizar con el público, y verificar y definir las ideas [26].

Las Personas se han elaborado intentando abarcar las siguientes características:

- Personas jóvenes, mayores y de mediana edad.

- Personas a las que les cuesta más usar aparatos tecnológicos.
- Personas a las que les gusta la tecnología y experimentar con ella.
- Personas que podrían vivir situaciones o experiencias para los cuales les podría venir bien adaptar la iluminación o la ambientación de la casa a diferentes situaciones, con el objetivo de: relajarse, divertirse, conciliar el sueño, facilitar la lectura, estudiar, etc.
- También se podría utilizar las ambientaciones para divertir o dormir a un bebé (por ejemplo).
- Personas de distintos países (España, Italia, Estados Unidos).



Figura 3: Persona 1 - Antonio Carbonero Brunel

Iolanda Venuti

Xtenso



"Cafeína y código"

Age: 25.

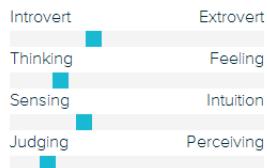
Work: Ingeniera de software.

Family: Padres.

Location: Florencia.

Archetype: Trabajo desde casa.

Personality



Goals

- Ser una gran programadora.
- Llegar muy lejos en su trabajo.
- Viajar por todo el mundo.

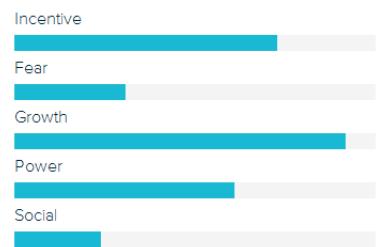
Frustrations

- No concentrarse en su trabajo.
- El dolor de cabeza tras una larga jornada.
- Miedo a no ser lo suficiente buena como desarrolladora.

Bio

Iolanda trabaja desde casa como desarrolladora en una empresa de software. A diario pasa casi todo su tiempo sentada frente al monitor del ordenador, enfocada en resolver diversos problemas. En ocasiones se frustra porque no logra concentrarse en su trabajo y rinde menos de lo que le gustaría. Tras algunas jornadas largas de trabajo acaba con dolor de cabeza y necesita despejarse un rato, pero a pesar de todo le encanta su trabajo y quiere crecer todo lo posible en su campo profesional.

Motivation



Brands & Influencers



Preferred Channels

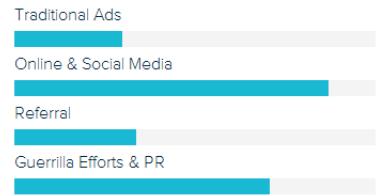


Figura 4: Persona 2 - Iolanda Venuti

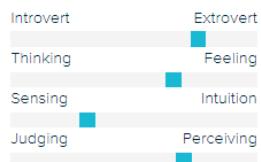
Izan Brown



"A quotation that captures this user's personality."

Age: 11
Work: Estudiante
Family: Padres y hermana
Location: Chicago, Illinois
Archetype: Niño

Personality



Goals

- Ser bombero algún día.
- Jugar y divertirse.

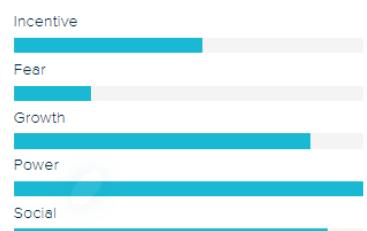
Frustrations

- Ir al colegio.
- Le frustra que le obliguen a tener responsabilidades.

Bio

Izan es un joven de Chicago que vive con sus padres en una pequeña casa y va al colegio, soñando algún día con convertirse en bombero. Es una persona muy inquieta y con mucha imaginación. Suele pasar el día jugando, a menudo con su hermana con quien comparte un gusto por los juegos de construcción. No le gusta demasiado estudiar, pero al final lo hace porque no quiere hacer enfadar a sus padres.

Motivation



Brands & Influencers



Preferred Channels

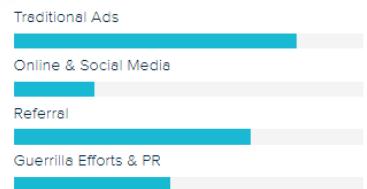


Figura 5: Persona 3 - Izan Brown

Lucía Esparta Serrano

Xtentio



"Quiero todo lo mejor para mi bebé"

Age: 42

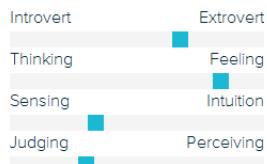
Work: Bibliotecaria.

Family: Marido e hijo.

Location: Madrid.

Archetype: Madre primeriza.

Personality



Goals

- Conseguir que su hijo tenga todo lo mejor y crezca sano.
- Tener una gran colección de libros en su propia casa.
- Aprender más sobre diseño de interiores.
- Tener una casa como la de los catálogos de revistas de decoración.

Frustrations

- No es muy amiga de la tecnología, prefiere usar lo que ya conoce.
- No lograr calmar el llanto de su hijo.

Bio

Lucía es una madre primeriza preocupada por su hijo recién nacido. Desea darle todo lo mejor y a menudo le preocupa no tener la capacidad de hacerlo. Pasa largas horas pendiente de su bebé y en alerta para consolar su llanto. Lucía ha pasado parte de su vida como bibliotecaria, siempre le han gustado mucho los libros, aunque admite que colecciona más libros de los que lee. Le encanta el diseño de interiores, sobretodo todo lo relacionado con tendencias minimalistas.

Motivation

Incentive

Fear

Growth

Power

Social

Brands & Influencers

 CÍRCULOS LECTORES La Redoute

Preferred Channels

Traditional Ads

Online & Social Media

Referral

Guerrilla Efforts & PR

Marco Ariza Maldonado

Xtensio



"Es duro cuidar de dos hijos"

Age: 34

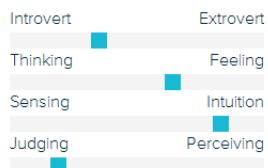
Work: No trabaja

Family: Mujer e hijos

Location: Barcelona

Archetype: Ahorrador, amigo de la tecnología, con problemas para dormir.

Personality



Goals

- Cuidar de su familia.
- Poder recuperar su vida laboral algún día.
- Jugar a muchos videojuegos.

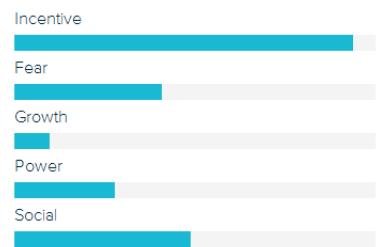
Frustrations

- Es muy nervioso y a veces le cuesta conciliar el sueño.
- Le preocupa no tener dinero suficiente para el futuro de sus hijos.

Bio

Marco es un hombre casado. Pasa la mayor parte del tiempo en casa cuidando de sus dos hijos, ya que su pareja trabaja. Le preocupa mucho la economía de su familia y siempre trata de ahorrar todo lo posible. A Marco no le asusta la tecnología, cuando tiene un respiro le gusta pasar el tiempo con su ordenador jugando a videojuegos. Es una persona muy nerviosa, a veces sus hijos le dan mucho trabajo y le cuesta conciliar el sueño.

Motivation



Brands & Influencers

CALL-DUTY MERCADONA

Preferred Channels

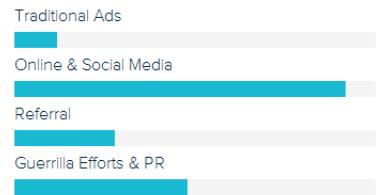


Figura 7: Persona 5 - Marco Ariza Maldonado



"Me encantan los animales marinos y la lectura."

Age: 19

Work: Estudiante de biología

Family: Padres.

Location: Valencia

Archetype: Estudiante.

Goals

- Ser bióloga marina.
- Vivir en un lugar con mar, pero fuera de España.
- Mejorar su nivel de inglés.

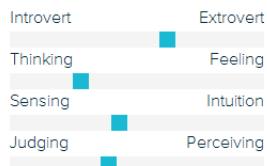
Frustrations

- La contaminación y el cambio climático.
- No tener tiempo suficiente para todo lo que quiere hacer.
- El machismo y ciertas corrientes políticas.

Bio

Marina es una estudiante muy aplicada que dedica gran parte de su tiempo libre a la lectura. Comparte piso con otros estudiantes, aunque suele pasar bastante tiempo fuera de casa. A menudo suele llegar tarde a casa y algo estresada, está acostumbrada a acostarse tarde y a madrugar. Suele quedarse dormida mientras lee o mientras comparte sus últimos pensamientos del día a través de Twitter usando su móvil. Cuando era más joven era muy tímida, pero desde que empezó la universidad Marina se ha vuelto una persona más sociable y está acostumbrada a dar fiestas y recibir amigos en su casa.

Personality



Motivation

Incentive

Fear

Growth

Power

Social

Brands & Influencers

GREENPEACE



Preferred Channels

Traditional Ads

Online & Social Media

Referral

Guerrilla Efforts & PR

Figura 8: Persona 6 - Marina Páramos Díaz

Para realizar estas Personas se ha utilizado la herramienta <https://xtensio.com/>.

3.3. Análisis de aplicaciones similares

En la fase de diseño, se realizó un análisis de otras aplicaciones para el sistema Philips Hue disponibles en el mercado, a continuación comentaremos las conclusiones obtenidas.

El mercado de aplicaciones de Philips Hue se basa principalmente en dos tipos de aplicaciones:

- Aplicaciones para simular efectos concretos: Como Hue Thunder (simular

una tormenta), Hue Body Soul (escenas de meditación), Hue Halloween, Fireworks for Philips Hue (chimenea), etc.

- Aplicaciones para controlar el sistema de iluminación: Todas ellas permiten apagar y encender los dispositivos. La mayoría permite cambiar la tonalidad de la iluminación en base a la temperatura o el color. También suelen permitir crear habitaciones. Algunas de ellas incluyen funcionalidades como:
 - Escenas, presets, etc: Se trata de estados estáticos guardados.
 - Timers, alarms: Permite encender o apagar luces concretas en un tiempo determinado.
 - También se ha visto en la aplicación Hue Pro y all 4 hue la posibilidad de crear eventos basándose en condiciones definidas por el usuario.

Las aplicaciones analizadas en profundidad han sido: Hue Essentials, Hue Thunder, Hue Pro, all 4 hue, Hue Switcher y Lumio. A continuación, se muestra una tabla de comparación:

Nombre	Comentarios	Precio	Descargas	Puntuación en la tienda
Hue Essentials	No permite cambiar el brillo ni color. Permite aplicar escenas Aspecto ordenado.	Gratis		
Hue Thunder	Simula el efectos de tormentas.	2.99€	>1.000	4.3
Hue Pro	Gran nivel de personalización. Poco intuitiva. Permite el control completo.	2.29€	> 100.000	4.2
all 4 hue	Interfaz muy poco cuidada y poco intuitiva. Permite el control completo.	Gratis	> 100.000	4.5
Hue Switcher	Interfaz es demasiado oscura y poco intuitiva.. Permite el control completo.	Gratis	>50.000	4.3
Lumio	Es la única de la tabla con estilo Material Design.. Permite el control completo.	Gratis	>5.000	3.9

Figura 9: Tabla comparativa de algunas aplicaciones para controlar el sistema de iluminación Philips Hue.

Estas aplicaciones tienen decenas de miles de descargas, algunas cientos de miles, y muchas de ellas suelen tener un precio superior a 2 euros (aunque en esta tabla solo se hayan analizado dos de pago). Lo cual nos permite obtener la conclusión de que, a pesar de ser un sistema muy concreto y que hay que comprar, hay bastantes usuarios que demandan este tipo de aplicaciones.

Un problema observado es que muchas de estas aplicaciones cuentan con una interfaz demasiado oscura, sin seguir aparentemente las guías de estilo, y llegando a ser poco intuitivas. Algunas parecen querer abarcar más funciones u ofrecer más información de la que realmente es útil al usuario. En otras, en cambio, se echan de menos algunas funciones básicas. Creo que algunas de las aplicaciones deberían ser más intuitivas y sencillas.

En la aplicación desarrollada se ha querido hacer hincapié en estos puntos como elemento diferenciador. Además de que la aplicación pretende ofrecer en el futuro una funcionalidad totalmente distinta a las encontradas en el mercado, el modo inteligente. Tampoco se han encontrado otras aplicaciones que permitan

crear escenas dinámicas del modo que se lleva a cabo en la aplicación desarrollada.

3.4. Modelado de requisitos

Para desglosar todo lo que debe contener y realizar la aplicación, se ha llevado a cabo una fase de extracción de requisitos. Estos requisitos son el resultado de discutir y plantear todo lo que queremos que haga la aplicación, además de intentar satisfacer todas las necesidades posibles del público que comentábamos en el apartado anterior. Durante el desarrollo, esta lista ha sufrido diversas modificaciones en base a las pruebas y prototipos que se han ido realizando. A continuación se muestra el resultado final después de las diversas iteraciones realizadas.

3.4.1. Requisitos funcionales

Configuración y red:

- **RF_0.** Detectar si ya ha sido configurado algún gateway.
- **RF_1.** Buscar gateways disponibles en la red local.
- **RF_2.** Establecer conexión con un gateway.
- **RF_3.** Detectar si el usuario se encuentra conectado a una red Wi-Fi, para mostrar un error en caso de que no.
- **RF_4.** Permitir reintentar la búsqueda del gateway en caso de que no se encuentre ninguno disponible.
- **RF_5.** Iniciar sesión con la cuenta de Google.
- **RF_6.** Detectar si el usuario ya ha iniciado sesión previamente.
- **RF_7.** Permitir buscar un nuevo gateway en la red.

Habitaciones y dispositivos de luz:

Los siguientes requisitos recogen todas las necesidades a la hora de controlar el sistema de iluminación, además de la personalización y estructuración en habitaciones. Permitiendo al usuario crear habitaciones y controlar la iluminación de ellas.

- **RF_8.** Detectar dispositivos de luz válidos y conectarlos automáticamente.
- **RF_9.** Listar habitaciones disponibles.
- **RF_10.** Mostrar las habitaciones que están encendidas.

- **RF_11.** Añadir nuevas habitaciones indicando el nombre, el tipo, y los dispositivos de luz que formarán parte de ella.
- **RF_12.** Modificar una habitación ya existente.
- **RF_13.** Encender o apagar luces de una habitación.
- **RF_14.** Permitir hacer parpadear un dispositivo de luz para que el usuario pueda ubicarlo.
- **RF_15.** Modificar el valor de la iluminación de una habitación.
- **RF_16.** Modificar el valor de la iluminación de un dispositivo de luz en concreto.
- **RF_17.** Cambiar el color de la iluminación de una habitación.
- **RF_18.** Cambiar el color de la iluminación de un dispositivo de luz.
- **RF_19.** Listar colores disponibles.
- **RF_20.** Añadir nuevos colores.
- **RF_21.** Borrar colores existentes.
- **RF_22.** Restaurar colores por defecto.
- **RF_23.** Modificar el nombre de un dispositivo de luz seleccionado.
- **RF_24.** Eliminar habitaciones.
- **RF_25.** Sincronizar configuración de las habitaciones entre distintas instancias de la aplicación para la misma cuenta de usuario.
- **RF_26.** Refrescar lista de habitaciones.
- **RF_27.** Aplicar escenas en habitaciones y dispositivos de luz.
- **RF_28.** Listar dispositivos de luz disponibles en una habitación concreta.
- **RF_29.** Indicar los dispositivos de luz que están encendidos para una habitación.

Proyectos de luz:

Los siguientes requisitos detallan todas las necesidades que tenemos que satisfacer para poder elaborar el sistema de recetas que nos permite especificar ambientaciones.

- **RF_30.** Leer proyectos de luz en formato JSON.
- **RF_31.** Interpretar proyectos de luz. Permitiendo indicar el color de cada dispositivo de luz en cada momento, si un dispositivo parpadea, si debe reproducirse en bucle, el sonido a ejecutar, etc.

- **RF_32.** Reproducir sonido asociado a un proyecto de luz en el altavoz del dispositivo.
- **RF_33.** Hacer streaming del sonido a algún dispositivo que lo permita de la red (Google Home, Google Chromecast, etc).
- **RF_34.** Pausar sonido.
- **RF_35.** Continuar con la reproducción del sonido.
- **RF_36.** Leer proyectos de luz alojados en un directorio del dispositivo.
- **RF_37.** Leer proyectos de luz compilados con la aplicación.
- **RF_38.** Recargar proyectos de luz.
- **RF_39.** Listar proyectos de luz en la aplicación.
- **RF_40.** Personalizar la imagen asociada con un proyecto de luz.
- **RF_41.** El recurso de sonido puede estar alojado en un servidor o en el dispositivo, para el primer caso la aplicación lo reproducirá en streaming.
- **RF_42.** Detener proyecto de luz.

3.4.2. Requisitos no funcionales

En la lista de requisitos no funcionales, haremos referencia a todas las restricciones y elementos que debemos tener en cuenta en la aplicación para lograr los objetivos propuestos.

- **RNF_0.** El usuario no puede perder el control de lo que está pasando en ningún momento.
- **RNF_1.** Los proyectos de luz se ejecutan de forma asíncrona y permiten al usuario seguir navegando por la aplicación, permitiendo interrumpir el proyecto en cualquier momento.
- **RNF_2.** La aplicación sigue las guías de estilo Material Design.
- **RNF_3.** La aplicación tiene una interfaz coherente, clara, bien estructurada y atractiva.
- **RNF_4.** La aplicación es localizable, e inicialmente estará disponible en inglés y español.
- **RNF_5.** Todas las imágenes tienen descripción para procurar la accesibilidad.
- **RNF_6.** El usuario puede identificar el tipo de habitación.
- **RNF_7.** La aplicación muestra únicamente la información relevante para el usuario.

- **RNF_8.** La aplicación no puede crashear en ningún momento. Todos los errores tienen que estar controlados y previstos.
- **RNF_9.** La aplicación es capaz de inicializar el sistema tras un hard reset del gateway.
- **RNF_10.** Si se cae la red, el gateway, o un dispositivo; la aplicación puede continuar funcionando indicando los problemas detectados.
- **RNF_11.** Se debe hacer un uso correcto de los assets de Android, de modo que la aplicación sea fácilmente escalable.
- **RNF_12.** Las imágenes tienen que ser visualmente descriptivas.
- **RNF_13.** La aplicación puede detectar cambios en las habitaciones o dispositivos de luz en realtime. De modo que se mantenga actualizada incluso si se están realizando cambios desde otro dispositivo.
- **RNF_14.** El número de bombillas disponibles debe de ser indiferente a la hora de ejecutar escenas o proyectos de luz.
- **RNF_15.** Si falta algún atributo poco relevante en el proyecto de luz, esto no debe de ser motivo para que el proyecto de luz no se pueda interpretar.

3.4.3. Requisitos de información

La aplicación deberá almacenar la información necesaria para sincronizarse si el usuario inicia sesión en dos instancias diferentes, la información básica del usuario, información sobre la personalización, y los proyectos de luz.

- **RI_0.** Información básica del usuario: nombre completo, email, id, foto de perfil.
- **RI_1.** Habitaciones creadas: tipo de habitación, dispositivos que forman parte de ella y nombre.
- **RI_2.** Lista de colores modificada por el usuario.
- **RI_3.** Lista de escenas disponibles para cada habitación.
- **RI_4.** Proyectos de luz por defecto y personalizados por el usuario. Los proyectos de luz se componen de la siguiente información:
 - - Nombre (cadena de texto).
 - - Autor (cadena de texto).
 - - Descripción (cadena de texto).
 - - Fecha de creación (cadena de texto).

- - Emociones que trata (array de cadenas de texto).
- - Bucle (booleano).
- - Imagen (cadena de texto, ruta relativa al archivo de imagen).
- - Sonido (cadena de texto, ruta relativa al archivo de sonido o url).
- - Activado (booleano).
- - Lista de acciones. Cada acción se compone por:
 - - Duración de la acción (entero, se mide en segundos).
 - - Lista de acciones de luz. Cada acción de luz se compone por:
 - - Brillo (entero, valor entre 0 y 255).
 - - Color (se puede indicar como el valor entero del color o valor hexadecimal).
 - - Parpadeo (booleano).

3.5. Actores y Casos de uso

En esta sección se identifican los principales actores y casos de uso que serán desarrollados en la aplicación.

3.5.1. Actores

Actor	Usuario			A0
Descripción	Usuario que utiliza la aplicación. Podría ser cualquier usuario descrito en la sección de “Público objetivo y Personas”.			
Características				
Relaciones	Este actor interactúa con Google Home y Philips Hue a través de las opciones disponibles en la aplicación. Por ejemplo, el usuario puede pedir a Philips Hue que encienda un dispositivo de luz, o que ejecute un proyecto de luz.			
Referencias				
Autor	Javier	Fecha	05/04/2019	Versión 1
Atributos				
Nombre	Descripción			Tipo
Datos personales	Nombre completo, email, id, foto de perfil.			Tabla de atributos de texto.
Preferencias de personalización	Colores personalizados, proyectos de luz añadidos, habitaciones asociadas.			

Figura 10: A0 - Usuario

3.5.2. Casos de uso

Lista de casos de uso:

- **CU0.** Conectar al gateway del sistema de iluminación
- **CU1.** Crear habitación
- **CU2.** Modificar habitación
- **CU3.** Eliminar habitación
- **CU4.** Visualizar habitación
- **CU5.** Escanear dispositivos de luz
- **CU6.** Apagar luces de la habitación
- **CU7.** Encender luces de la habitación
- **CU8.** Cambiar intensidad del brillo de un dispositivo de luz
- **CU9.** Silenciar proyecto de luz
- **CU10.** Cambiar color de las luces de la habitación
- **CU11.** Apagar un dispositivo de luz
- **CU12.** Encender un dispositivo de luz
- **CU13.** Cambiar intensidad del brillo de un dispositivo de luz
- **CU14.** Iniciar proyecto de luz
- **CU15.** Detener proyecto de luz
- **CU16.** Reproducir sonido
- **CU17.** Arrancar el sistema
- **CU18.** Escanear habitaciones
- **CU19.** Hacer que un dispositivo de luz parpadee
- **CU20.** Aplicar una escena a una habitación
- **CU21.** Cargar proyectos de luz
- **CU22.** Añadir un nuevo color a la paleta
- **CU23.** Eliminar colores de la paleta

Diagrama de casos de uso:

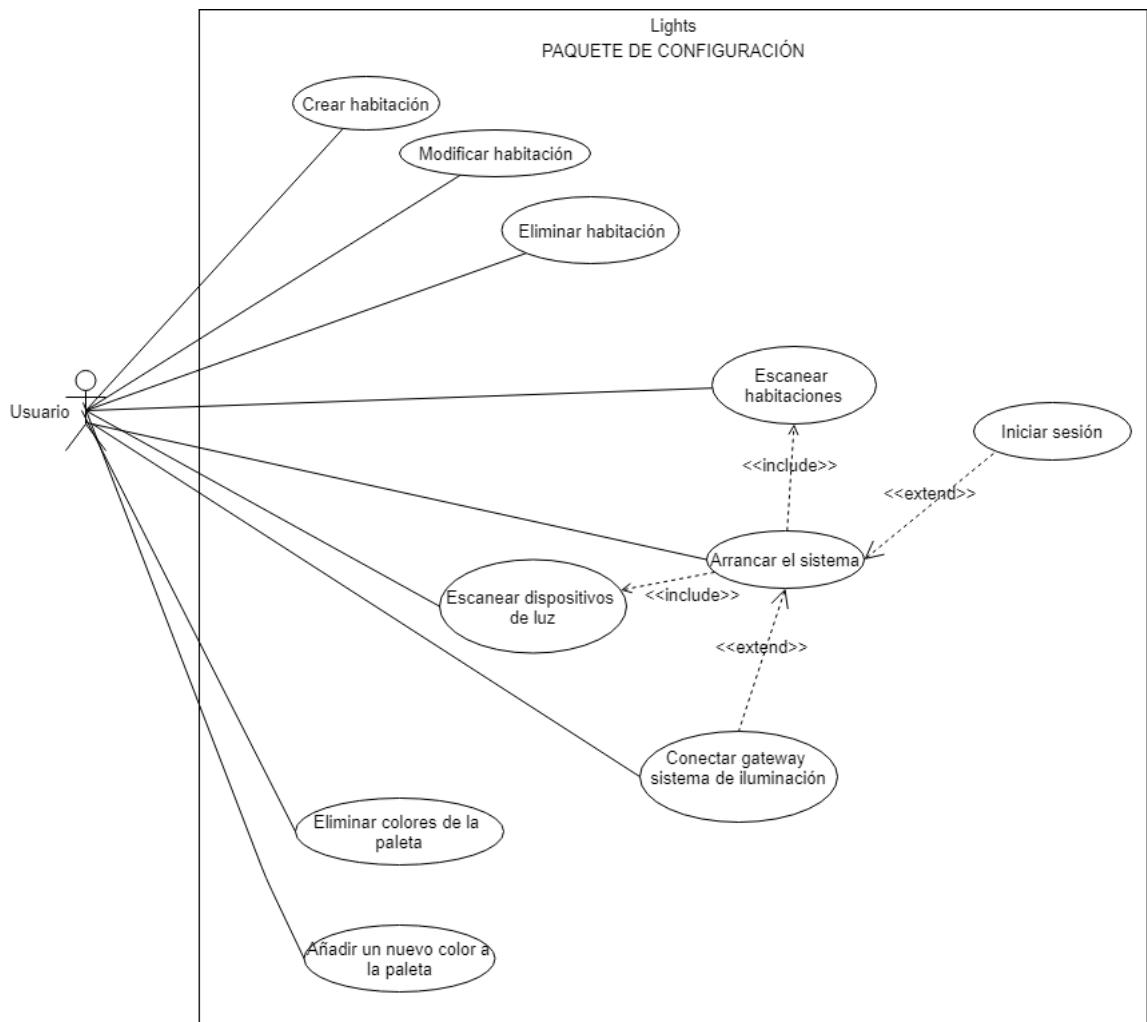


Figura 11: Diagrama de casos de uso: Paquete de configuración

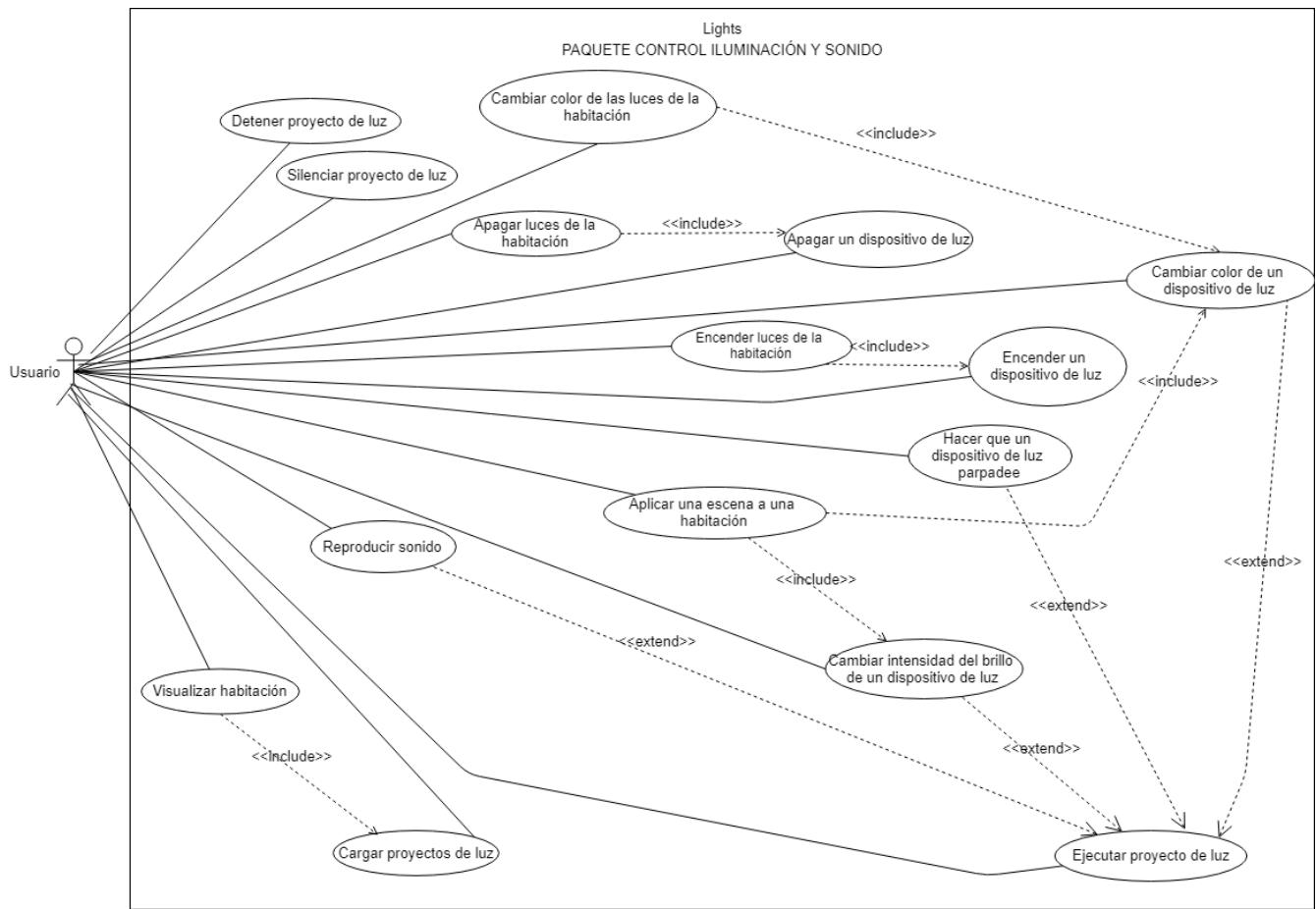


Figura 12: Diagrama de casos de uso: Paquete de iluminación y sonido

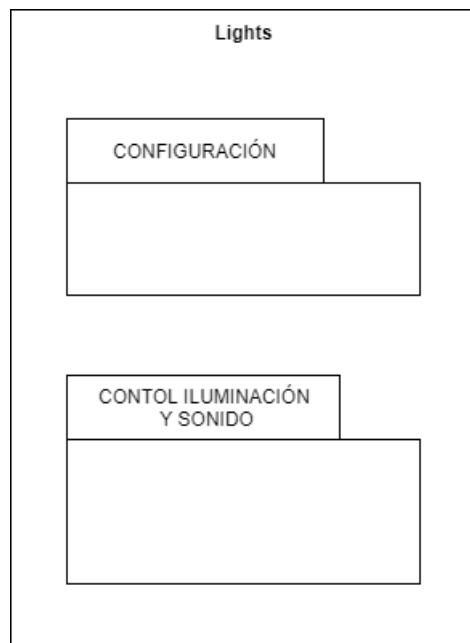


Figura 13: Diagrama de paquetes funcional

Especificación casos de uso:

Casos de uso	Conectar al gateway del sistema de iluminación				CU0		
Actores	Usuario						
Tipo	Configuración						
Referencias							
Precondición							
Postcondición	La aplicación quedó conectada al gateway del sistema de iluminación.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	Establecer conexión del software con el gateway para poder comenzar a utilizar el sistema de iluminación.						
Resumen	La primera vez que el usuario inicia la aplicación, o cuando el usuario lo solicite, se abrirá un asistente que le indicará paso por paso cómo vincular la aplicación y el bridge para así establecer la conexión entre el sistema de iluminación y la aplicación.						
Curso normal (básico)							
1	El usuario inicia la conexión.		2	El asistente busca en la red local los gateways que haya disponibles.			
			3	Pedir al usuario que pulse el botón para aceptar la conexión en el gateway.			
4	El usuario pulsa el botón superior del gateway al que desea conectar la aplicación, para permitir a la app conectarse.		5	Se establece la conexión y se guarda como punto de acceso al sistema de iluminación.			
Cursos alternos							
2a	El usuario no se encuentra conectado a una red WiFi. Se muestra un mensaje con el error y se muestra la opción de reintentar, volviendo a 2.						

Figura 14: CU0 - Conectar al gateway del sistema de iluminación

Casos de uso	Crear habitación			CU1
Actores	Usuario			
Tipo	Configuración			
Referencias				
Precondición	Hay dispositivos de luz registrados.			
Postcondición	La aplicación ha registrado una nueva habitación y la ha añadido a la lista de habitaciones disponibles en el sistema.			
Autor	Javier	Fecha	05/04/2019	Versión 1
Propósito				
Registrar una nueva habitación. Permitiendo que el usuario seleccione qué dispositivos la conforman y qué tipo de habitación es.				
Resumen				
El usuario puede seleccionar todos los dispositivos de luz que conformen la habitación, de modo que se registre en el sistema una nueva habitación disponible indicando, además, el tipo de habitación que es. Los dispositivos de luz seleccionables son los que el propio gateway de Philips Hue nos indica.				
Curso normal (básico)				
1	El usuario inicia el proceso de creación.	2	Se indica al usuario que seleccione de una lista qué tipo de habitación es. Las opciones son: Dormitorio, comedor, salón, cuarto de baño, estudio, cocina, otro. Y que indique un nombre para la habitación. Además, se solicita un nombre para la habitación y seleccionar la lista de dispositivos de luz que la conformarán.	
3	El usuario selecciona uno de los tipos de habitaciones disponibles en la lista, además de especificar un nombre y seleccionar los dispositivos de luz deseados.	4	La aplicación guarda la configuración creando en el sistema una nueva habitación disponible.	

Figura 15: CU1 - Crear habitación

Casos de uso	Modificar habitación			CU2
Actores	Usuario			
Tipo	Configuración			
Referencias				
Precondición	Hay habitaciones existentes. La habitación que se desea modificar se está visualizando.			
Postcondición	Los cambios realizados en la habitación se han registrado en el sistema.			
Autor	Javier	Fecha	05/04/2019	Versión 1
Propósito				
Permitir modificar habitaciones ya existentes en el sistema. Permitiendo así adaptar las habitaciones existentes a las nuevas necesidades del usuario o corregir errores.				
Resumen				
El usuario puede modificar la configuración de una habitación, modificando los datos existentes y registrando los cambios en el sistema.				
Curso normal (básico)				
1	El usuario inicia el proceso de edición seleccionando la opción de editar en la habitación seleccionada.	2	Se muestran los datos editables de la habitación actual seleccionada: - Nombre. - Tipo. - Lista de dispositivos de luz asociados.	
3	El usuario puede modificar el nombre, o cambiar el tipo seleccionando otro de una lista desplegable, o deseleccionar sistemas de luz asociados y seleccionar otros.			
4	El usuario indica que desea guardar los cambios.	5	Los cambios se registran y se actualizan la habitación.	

Figura 16: CU2 - Modificar habitación

Casos de uso	Eliminar habitación			CU3			
Actores	Usuario						
Tipo	Configuración						
Referencias							
Precondición	Hay habitaciones existentes. La habitación que se desea eliminar se está visualizando.						
Postcondición	Los cambios indicados se han registrado en el sistema.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	Permitir al usuario eliminar habitaciones existentes en el sistema.						
Resumen							
Curso normal (básico)							
1	El usuario indica que desea eliminar la habitación.		2	El sistema pide al usuario confirmación, ya que la habitación eliminada no se podrá restaurar en el futuro.			
3	El usuario confirma la operación.		4	La habitación es eliminada.			
Cursos alternos							
3a	El usuario cancela la operación y por lo tanto no se efectúa ningún cambio.						

Figura 17: CU3 - Eliminar habitación

Casos de uso	Visualizar habitación			CU4	
Actores	Usuario				
Tipo	Básico				
Referencias					
Precondición	El usuario tiene una o más habitaciones registradas en el sistema.				
Postcondición	Todas las opciones disponibles para esa habitación se muestran al usuario.				
Autor	Javier	Fecha	05/04/2019	Versión	1
Propósito					
El usuario puede visualizar una habitación, para así modificar los dispositivos de luz que se encuentran en esa habitación, cambiar el color, modificar la escena, acceder a las opciones de edición, etc.					
Resumen					
El usuario toca la habitación que desea visualizar.					
Curso normal (básico)					
		1	El sistema muestra al usuario la lista de todas las habitaciones disponibles en el sistema.		
2	El usuario pulsa sobre la habitación que desea visualizar.				

Figura 18: CU4 - Visualizar habitación

Casos de uso	Escanear dispositivos de luz			CU5				
Actores								
Tipo	Configuración							
Referencias								
Precondición								
Postcondición	Se ha actualizado la configuración en caso de que el estado del sistema haya cambiado (se hayan eliminado o añadido nuevos dispositivos).							
Autor	Javier	Fecha	05/04/2019	Versión	1			
Propósito								
Reaccionar ante modificaciones o problemas del conjunto de dispositivos de luz conectados al sistema.								
Resumen								
El sistema pregunta al gateway por los dispositivos conectados y actualiza los datos que dispone.								
Curso normal (básico)								
		1	El sistema comienza a escanear los dispositivos.					
		2	El sistema pide al gateway la lista de dispositivos conectados.					
		3	Se actualiza la lista de dispositivos de luz.					
Cursos alternos								
3a	En caso de detectar un dispositivo de luz que no está registrado en el gateway, lo registra para que esté disponible en el sistema.							

Figura 19: CU5 - Escanear dispositivos de luz

Casos de uso	Apagar luces de la habitación			CU6			
Actores	Usuario						
Tipo	Control de iluminación						
Referencias	CU15 Detener proyecto de luz						
Precondición	Las luces de la habitación tienen que estar encendidas..						
Postcondición	Las luces de la habitación se han apagado. Han dejado de aplicarse acciones de luz en la habitación.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	El usuario puede apagar las luces de la habitación cuando lo desee.						
Resumen	El usuario le indica a la aplicación que apague todas las luces de la habitación.						
Curso normal (básico)							
1	El usuario pulsa sobre el botón que le permite apagar la luz de la habitación actual.	2	Se apagan todos los dispositivos de luz presentes en la habitación actual. En caso de haber algún proyecto de luz en ejecución, se detiene.				

Figura 20: CU6 - Apagar luces de la habitación

Casos de uso	Encender luces de la habitación			CU7			
Actores	Usuario						
Tipo	Control de iluminación						
Referencias							
Precondición	Tiene que haber luces apagadas en la habitación para que tenga efecto.						
Postcondición	Las luces de la habitación han quedado encendidas.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito							
El usuario puede encender las luces de la habitación cuando lo desee.							
Resumen							
El usuario le indica a la aplicación que encienda todas las luces de la habitación.							
Curso normal (básico)							
1	El usuario pulsa sobre el botón que le permite encender la luz de la habitación actual.		2	Se encienden todos los dispositivos de luz presentes en la habitación actual.			

Figura 21: CU7 - Escanear luces de la habitación

Casos de uso	Cambiar intensidad del brillo de las luces de la habitación				CU8
Actores	Usuario				
Tipo	Control de iluminación				
Referencias					
Precondición	Tiene que haber dispositivos de luz encendidos en la habitación.				
Postcondición	El brillo de los dispositivos de luz ha quedado modificado.				
Autor	Javier	Fecha	05/04/2019	Versión	1
Propósito	El usuario puede cambiar el brillo de los dispositivos de luz de la habitación a su antojo.				
Resumen	El usuario indica el brillo deseado para la habitación actual. Este brillo se aplica a todos los dispositivos de luz presentes en la habitación.				
Curso normal (básico)					
1	El usuario especifica el brillo deseado para los dispositivos de luz de la habitación actual, partiendo del brillo actual.	2	El sistema actualiza el brillo para todos los dispositivos de la habitación actual.		

Figura 22: CU8 - Cambiar intensidad el brillo de las luces de la habitación

Casos de uso	Silenciar proyecto de luz			CU9			
Actores	Usuario						
Tipo	Proyectos de luz						
Referencias							
Precondición	Hay un proyecto de luz con sonido en ejecución.						
Postcondición	El sonido se ha pausado.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito							
Permitir al usuario detener el sonido si lo desea.							
Resumen							
El sonido se detiene cumpliendo con las órdenes del usuario.							
Curso normal (básico)							
1	El usuario indica que desea silenciar el sonido que se está reproduciendo.		2	El sonido se pausa.			

Figura 23: CU9 - Silenciar proyecto de luz

Casos de uso	Cambiar color de las luces de la habitación			CU10			
Actores	Usuario						
Tipo	Control de iluminación						
Referencias	CU15 Detener proyecto de luz						
Precondición	Hay dispositivos de luz encendidos en la habitación.						
Postcondición	El color de los dispositivos de luz de la habitación ha cambiado.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	El usuario puede seleccionar el color de las luces de la habitación cuando lo desee.						
Resumen	El usuario seleccionará en una tabla de colores el color que deseé establecer en la habitación. El color de todas las luces de la habitación pasará a ser el seleccionado por el usuario.						
Curso normal (básico)							
1	El usuario indica que desea cambiar el color de las luces de la habitación.		2	Se muestra al usuario una tabla de colores disponibles.			
3	El usuario selecciona el color que deseé.		4	Se establece el color seleccionado por el usuario como color de todos los dispositivos de luz de la habitación. En caso de haber algún proyecto de luz en ejecución, se detiene.			

Figura 24: CU10 - Cambiar color de las luces de la habitación

Casos de uso	Apagar un dispositivo de luz			CU11			
Actores	Usuario						
Tipo	Control de iluminación						
Referencias							
Precondición	El dispositivo de luz tiene que formar parte de la habitación que se está visualizando. El dispositivo de luz tiene que estar encendido.						
Postcondición	Las luz indicada queda apagada.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	El usuario puede apagar alguna luz en concreto cuando lo desee.						
Resumen	El usuario indica a un dispositivo de luz concreto de la habitación actual que se apague.						
Curso normal (básico)							
			1	El sistema lista todos los dispositivos de luz de una habitación.			
2	El usuario pulsa sobre la opción de apagar un dispositivo encendido.		3	El sistema apaga el dispositivo indicado por el usuario.			

Figura 25: CU11 - Apagar un dispositivo de luz

Casos de uso	Encender un dispositivo de luz			CU12			
Actores	Usuario, Philips Hue						
Tipo	Control de iluminación						
Referencias							
Precondición	El dispositivo de luz tiene que formar parte de la habitación que se está visualizando. El dispositivo de luz tiene que estar apagado.						
Postcondición	Las luz indicada queda encendida.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito							
El usuario puede encender alguna luz en concreto de una habitación cuando lo desee.							
Resumen							
El usuario indica a un dispositivo de luz concreto de la habitación actual que se encienda.							
Curso normal (básico)							
			1	El sistema lista todos los dispositivos de luz de una habitación.			
2	El usuario pulsa sobre la opción de encender un dispositivo apagado.		3	El sistema enciende el dispositivo indicado por el usuario.			

Figura 26: CU12 - Encender un dispositivo de luz

Casos de uso	Cambiar intensidad del brillo de un dispositivo de luz			CU13			
Actores	Usuario						
Tipo	Control de iluminación						
Referencias							
Precondición	El dispositivo de luz tiene que formar parte de la habitación que se está visualizando. El dispositivo de luz tiene que estar encendido.						
Postcondición	El brillo del dispositivo de luz ha quedado modificado.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito	El usuario puede modificar el brillo de un dispositivo de luz concreto cuando lo desee.						
Resumen	El usuario modifica el brillo actual de un dispositivo de luz concreto de la habitación.						
Curso normal (básico)							
				1	El sistema lista todos los dispositivos de luz de una habitación.		
2	El usuario modifica el valor el brillo actual del dispositivo, pudiendo bajarlo o subirlo.			3	El sistema aplica el brillo indicado por el usuario en el dispositivo indicado.		

Figura 27: CU13 - Cambiar intensidad del brillo de un dispositivo de luz

Casos de uso	Iniciar proyecto de luz			CU14	
Actores	Usuario				
Tipo	Proyectos de luz				
Referencias					
Precondición	Se está visualizando una habitación.				
Postcondición	El proyecto de luz ha comenzado a ejecutarse sobre todas los dispositivos de luz de la habitación seleccionada. Si había algún proyecto de luz activo, ha quedado interrumpido. El usuario no podrá realizar otras operaciones hasta que el proyecto de luz sea interrumpido.				
Autor	Javier	Fecha	05/04/2019	Versión	1
Propósito					
El usuario puede aplicar un proyecto de luz cuando lo desee.					
Resumen					
El usuario puede aplicar a una habitación un proyecto de luz seleccionado de una lista de proyectos de luz disponibles. El proyecto puede tener efecto en todos los dispositivos de luz y de sonido de la habitación.					
Curso normal (básico)					
1	El usuario indica que desea aplicar un proyecto de luz manualmente a la habitación.		2	El sistema lista los proyectos de luz disponibles.	
3	El usuario selecciona el proyecto de luz deseado.		4	El sistema aplica el proyecto de luz en la habitación.	

Figura 28: CU14 - Iniciar proyecto de luz

Casos de uso	Detener proyecto de luz			CU15			
Actores	Usuario						
Tipo	Proyectos de luz						
Referencias							
Precondición	Hay un proyecto de luz en ejecución						
Postcondición	Se ha detenido la reproducción del proyecto de luz, los dispositivos de luz se han quedado en el estado que se encontraban en el momento de la interrupción.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito							
El usuario puede detener un proyecto de luz activo cuando lo desee.							
Resumen							
El usuario indica al sistema que quiere detener el proyecto de luz en ejecución, este es detenido obedeciendo la orden.							
Curso normal (básico)							
1	El usuario indica que desea detener el proyecto de luz en ejecución.	2	El proyecto de luz es interrumpido.				

Figura 29: CU15 - Detener proyecto de luz

Casos de uso	Reproducir sonido			CU16
Actores				
Tipo	Proyectos de luz			
Referencias				
Precondición	Se está ejecutando un proyecto de luz con un sonido válido asociado.			
Postcondición	Un sonido concreto ha comenzado a reproducirse a través del dispositivo.			
Autor	Javier	Fecha	11/04/2019	
Propósito				
Reproducir un sonido que acompañe las acciones.				
Resumen				
El sistema comenzará a reproducir un sonido en concreto a través del dispositivo Android.				
Curso normal (básico)				
		1	El sistema detecta un sonido asociado con el proyecto de luz actual en ejecución.	
		2	El sonido comienza a reproducirse utilizando el altavoz del dispositivo.	
		3	El sonido comienza a reproducirse utilizando el altavoz encontrado.	

Figura 30: CU16 - Reproducir sonido

Casos de uso	Arrancar el sistema			CU17
Actores	Usuario			
Tipo	Configuración			
Referencias	CU0 - Conectar al gateway del sistema de iluminación CU5 - Escanear dispositivos			
Precondición				
Postcondición	El sistema se ha iniciado, habiendo quedado listo para poder utilizarse.			
Autor	Javier	Fecha	11/04/2019	Versión 1
Propósito				
Iniciar el sistema de modo que quede listo para poder utilizarse.				
Resumen				
El sistema se inicia, registrado para ello al usuario en caso de ser necesario, configurando el gateway, creando las primeras habitaciones, etc.				
Curso normal (básico)				
1	El usuario inicia la aplicación.	2	Se comprueba que el usuario esté conectado a una red Wi-Fi.	
		3	Se comprueba si el usuario está identificado.	
		4	En caso de no existir la configuración de usuario, se le indica al usuario que inicie sesión a través de Google.	
5	El usuario inicia sesión utilizando su cuenta de Google.	6	El sistema crea las bases de datos y almacena los datos del usuario.	
		7	En caso de no detectarse gateway, se ejecuta el caso de uso: CU0 - Conectar al gateway del sistema de iluminación	
		8	Se lanza el caso de uso: CU5 - Escanear dispositivos.	
		9	Se lanza la actividad principal de la aplicación.	
Cursos alternos				
3a	En caso de que el dispositivo no esté conectado a una red Wi-Fi, se pide al usuario que lo conecte y se <u>reintenta</u> el paso 2.			
4b	Si el usuario ya estaba identificado en la app, se pasa al paso 7.			
7b	Si el gateway está configurado, saltamos al paso 8.			

Figura 31: CU17 - Arrancar el sistema

Casos de uso	Escanear habitaciones			CU18			
Actores							
Tipo	Configuración						
Referencias							
Precondición							
Postcondición	Se ha actualizado la lista de habitaciones en caso de que se hayan detectado nuevas habitaciones, habitaciones eliminadas, o modificaciones.						
Autor	Javier	Fecha	05/04/2019	Versión	1		
Propósito							
Reaccionar ante modificaciones de las habitaciones desde otros dispositivos identificados con la misma cuenta.							
Resumen							
El sistema pregunta a un servidor por la lista de habitaciones para ese usuario en esa casa, para a continuación actualizar la lista de habitaciones disponibles.							
Curso normal (básico)							
		1	El sistema pregunta a un servidor por la lista de habitaciones para ese usuario en esa casa.				
		2	El sistema obtiene la respuesta del servidor.				
		3	Se actualiza la lista de habitaciones disponible.				

Figura 32: CU18 - Escanear habitaciones

Casos de uso	Hacer que un dispositivo de luz parpadee			CU19
Actores	Usuario			
Tipo	Control de iluminación			
Referencias				
Precondición				
Postcondición	El dispositivo de luz ha mantenido el mismo estado que tenía antes de realizar esta acción.			
Autor	Javier	Fecha	05/04/2019	Versión 1
Propósito	Hacer que un dispositivo de luz parpadee, para que el usuario lo pueda identificar.			
Resumen	El usuario indica que desea hacer parpadear a un dispositivo de luz, este parpadea.			
Curso normal (básico)				
1	El usuario indica que desea que un dispositivo concreto parpadee.	2	El sistema hace parpadear ese dispositivo de luz concreto.	

Figura 33: CU19 - Hacer que un dispositivo de luz parpadee

Casos de uso	Aplicar una escena a una habitación			CU20
Actores	Usuario			
Tipo	Control de iluminación			
Referencias				
Precondición	Se está visualizando una habitación.			
Postcondición	El color y brillo de todos los dispositivos de luz de una habitación han quedado modificados a los valores establecidos en la escena seleccionada.			
Autor	Javier	Fecha	05/04/2019	
Propósito				
Permitir al usuario aplicar escenas guardadas.				
Resumen				
El usuario indica al sistema que desea aplicar una escena, esa escena se aplica afectando a todos los dispositivos de luz presentes en esa habitación.				
Curso normal (básico)				
1	El usuario indica que desea que aplicar una escena.	2	El sistema aplica para todos los dispositivos de la habitación los colores y niveles de brillo especificados en la escena.	

Figura 34: CU20 - Aplicar una escena a una habitación

Casos de uso	Cargar proyectos de luz			CU21
Actores				
Tipo	Proyectos de luz			
Referencias				
Precondición				
Postcondición	Los proyectos de luz detectados han quedado listos para ejecutarse.			
Autor	Javier	Fecha	05/04/2019	
Propósito				
Permitir leer proyectos de luz desde el sistema de archivos.				
Resumen				
El sistema detecta todos los proyectos de luz del sistema de archivos y los carga para que puedan ser lanzados en cualquier momento por el usuario.				
Curso normal (básico)				
		1	El sistema busca los proyectos de luz compilados con la aplicación o ubicados en un directorio externo, los cuales se encuentran en formato JSON.	
		2	El sistema interpreta el archivo JSON y carga todos los datos.	
Cursos alternos				
		2a	El archivo JSON está estructurado de forma que impide su lectura, por lo tanto se ignora.	

Figura 35: CU21 - Cargar proyectos de luz

Casos de uso	Añadir un nuevo color a la paleta			CU22			
Actores	Usuario						
Tipo	Configuración						
Referencias							
Precondición							
Postcondición	Un nuevo color definido por el usuario se ha registrado en la lista de colores disponibles						
Autor	Javier	Fecha	05/04/2019				
Propósito							
Permitir al usuario añadir los colores que desee. Pudiendo de este modo crear una paleta personalizada.							
Resumen							
El usuario indica de forma visual el color que desea añadir, este se añade a la lista de colores disponibles.							
Curso normal (básico)							
1	El usuario indica que desea añadir un nuevo color.		2	El sistema permite seleccionar de forma visual el color exacto que desea añadir.			
3	El usuario selecciona el color.		4	El color se registra y se añade a la paleta de colores disponibles.			

Figura 36: CU22 - Añadir un nuevo color a la paleta

Casos de uso	Eliminar colores de la paleta			CU23
Actores	Usuario			
Tipo	Configuración			
Referencias				
Precondición				
Postcondición	Una lista de colores han sido eliminados de la paleta de colores disponibles.			
Autor	Javier	Fecha	05/04/2019	
Propósito				
Permitir al usuario eliminar los colores que deseé. Pudiendo de este modo crear una paleta personalizada.				
Resumen				
El usuario selecciona los colores que desea eliminar de la paleta, y estos son eliminados.				
Curso normal (básico)				
1	El usuario indica que desea eliminar colores.			
2	El usuario selecciona todos los colores que desea eliminar.	3	El sistema elimina los colores.	

Figura 37: CU23 - Eliminar colores de la paleta

3.6. Interfaz de usuario

Es necesario elaborar una interfaz de usuario que permita crear la experiencia de usuario que deseamos. El primer paso que llevaremos a cabo es preguntarnos cuáles son las necesidades que debe de satisfacer esta interfaz:

- Tiene que ser intuitiva y estar poco cargada de información. El usuario tiene que poder cumplir su objetivo rápidamente.
- El usuario no se puede perder en la navegación. Buscaremos crear una sensación de continuidad entre las diferentes actividades que presentemos de modo que el usuario pueda visualizar en qué lugar de la aplicación se encuentra en todo momento.
- Las acciones más usadas, como encender o apagar las luces, deben de estar en un lugar privilegiado. De modo que siempre puedan encontrarse con un vistazo rápido a la pantalla.
- La aplicación deberá cumplir con las guías de estilo de Google Material Design. De esta forma la aplicación se integrará correctamente dentro de

la estética del dispositivo, y utilizando la simbología, efectos, y estructura a la que el usuario ya está acostumbrado. Con esto al usuario le resultará más sencilla de utilizar y crearemos una sensación de continuidad con el resto de aplicaciones del dispositivo que también cumplan con estas guías de estilo y el entorno [20].

- La aplicación debe de ser amigable y atractiva. Al ser una aplicación que permitirá controlar el sistema de iluminación y recrear ambientaciones, haremos uso de una gama amplia de colores que hagan referencia a la iluminación que presentan los dispositivos de luz.
- El usuario no puede perder en ningún momento el control de lo que está ocurriendo, por lo que siempre tiene que poder encontrar la opción adecuada para detener un proyecto de luz, ver qué se está ejecutando, o averiguar qué luz está encendida o apagada.
- Todas las opciones posibles deben de entenderse, sin dar por hecho que el usuario tiene conocimientos previos sobre el funcionamiento.
- Los elementos tienen que seguir una jerarquía visual que permita al usuario visualizar qué elemento forma parte de otro y entender como se estructura.
- Los elementos deberán ser coherentes, de modo que no confundamos al usuario.

3.6.1. Referencias

Además de basarnos en las recomendaciones y pautas que marcan las guías de estilo de Material Design, se han tenido en cuenta otras aplicaciones Android como referencia, incluyendo el hecho de contar con la propia experiencia de usuario utilizando otras aplicaciones diariamente. En primer lugar se hizo un recorrido por las principales herramientas del mercado para controlar el sistema de iluminación de Philips Hue (lo cual puede verse con más detalle en el apartado de Aplicaciones similares). También se han tenido en cuenta las interfaces de aplicaciones Android que aplican de forma correcta el estilo Material Design, como lo son las siguientes que se comentan en esta sección.

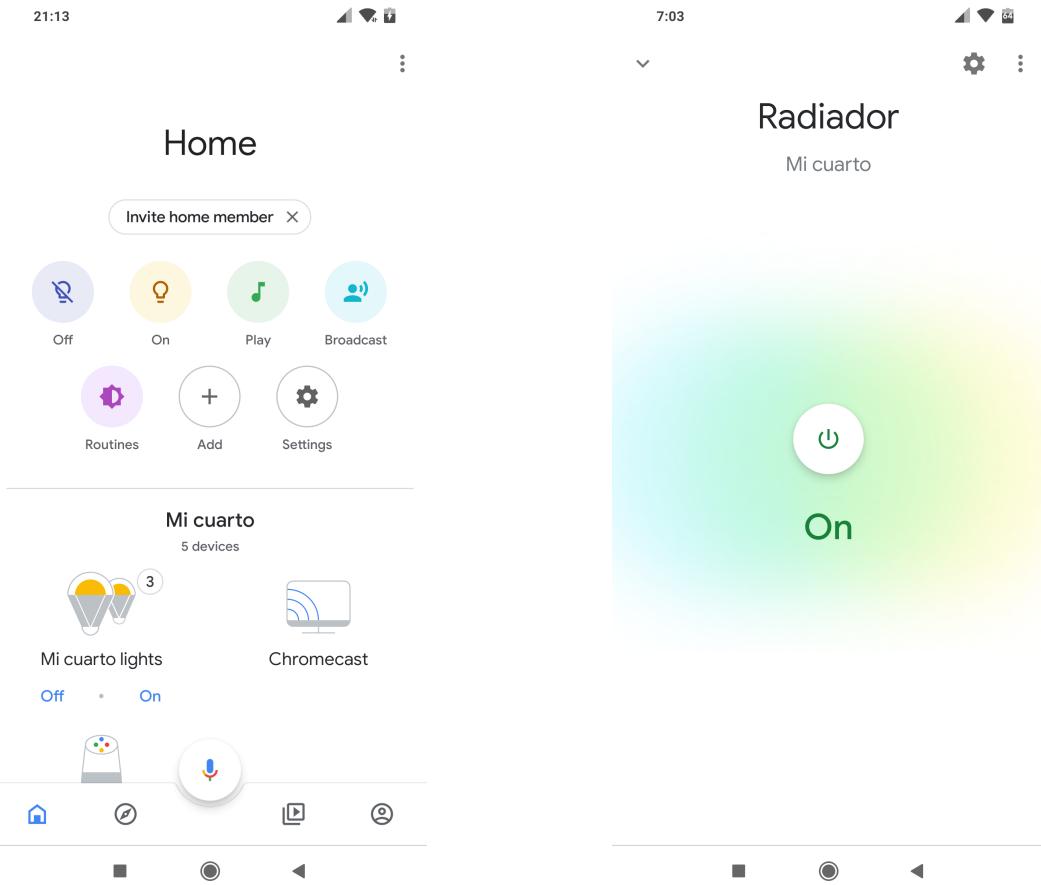


Figura 38: Referencia interfaz gráfica - Google Home [15]

Figura 39: Referencia interfaz gráfica - Google Home [15]

De las dos figuras anteriores, tomaremos como referencia:

- La lista de dispositivos que permite desplazar toda la interfaz para mostrar los elementos que no caben en la pantalla. Al no desplazarse únicamente los elementos de la lista, estos nunca se mostrarán cortados por una cabecera que no siempre es necesaria.
- La interfaz blanca y clara, que transmite sencillez y es atractiva.
- El menú de navegación en la parte de abajo, lo que permite a los usuarios navegar sin tener que desplazar demasiado el dedo [6].
- El uso de la fuente ‘sans-serif’.
- La actividad que permite centrar la atención en el control del dispositivo que se está ejecutando, colocando en el lugar más destacado de la pantalla la acción más útil e importante: detener la acción.
- El uso de colores llamativos que dan a entender que el dispositivo está activo.

- Las animaciones con las que se abren y se cierran las actividades. Transmite la sensación de que la pantalla de ejecución del dispositivo se despliega sobre la actividad principal, volviendo a minimizarse al cerrarla. La flecha hacia abajo como botón de cerrar refuerza aún más esa sensación.
- Menú de opciones situado en la esquina superior izquierda, de forma discreta pero fácilmente localizable. Ocupando un lugar coherente [4].

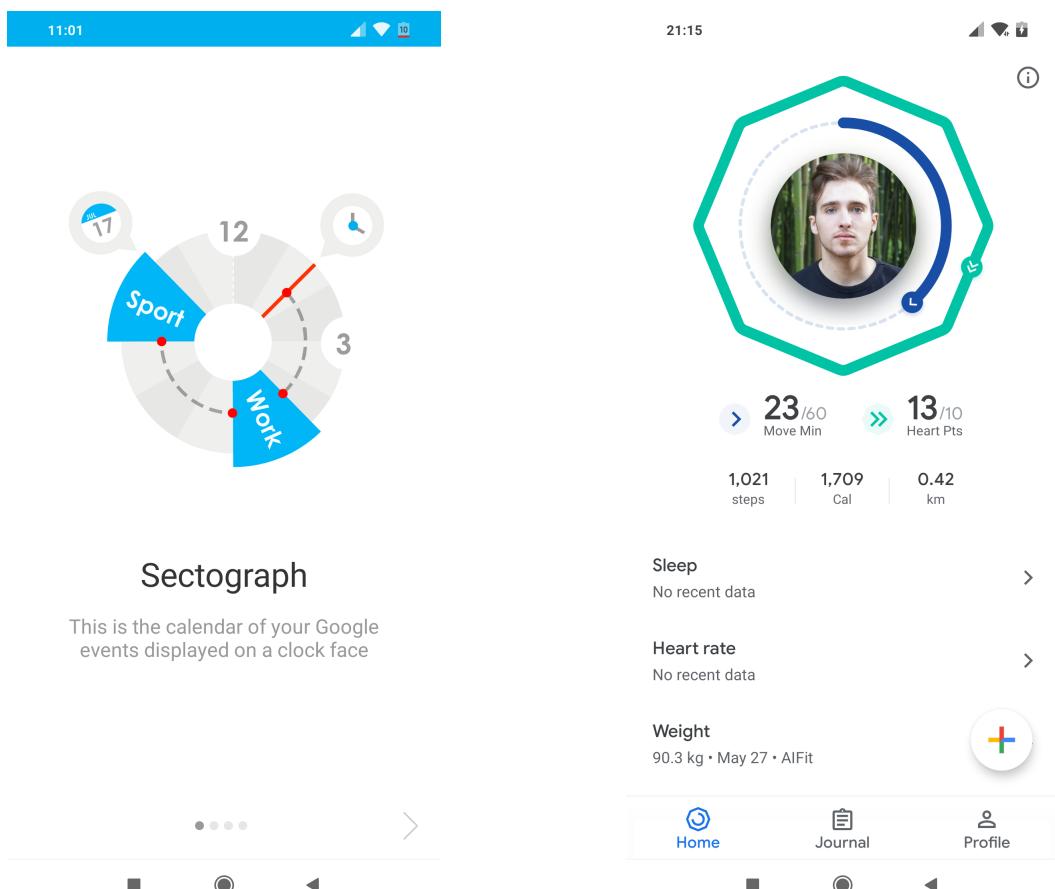


Figura 40: Referencia interfaz gráfica - Sectograph [27]

Figura 41: Referencia interfaz gráfica - Google Fit [14]

En las dos figuras anteriores, podemos observar lo siguiente:

- Una vez más, las interfaces son claras con colores saturados, y de nuevo utilizando la fuente 'sans-serif' (en lugar de Roboto, que suele ser la fuente por defecto). Además de situar el menú en la parte inferior de la pantalla.
- La interfaz centrando en la pantalla lo más importante, y mostrando únicamente los elementos necesarios.
- La navegación con puntos durante la configuración que señalan en qué lugar del proceso se encuentra el usuario. Manteniéndole así informado de dónde está, pero de forma discreta y elegante.

- El botón flotante que representa la acción principal que se puede llevar a cabo en la actividad [7].
- Las flechas que apuntan hacia la derecha en las opciones, son acordes a la animación, de modo que el usuario tiene la sensación de situarse a la derecha cuando abre esa actividad. Al cerrarla volverá hacia la izquierda pulsando una flecha en esa dirección. De este modo se consigue continuidad y coherencia en la navegación.



Figura 42: Referencia interfaz gráfica - Google Trips [16]

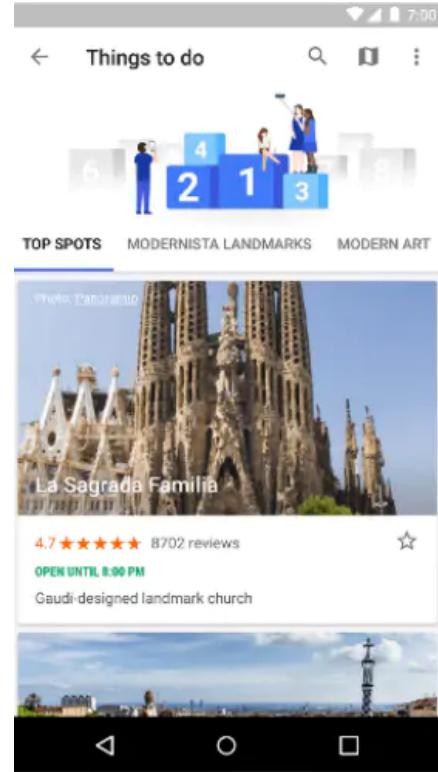


Figura 43: Referencia interfaz gráfica - Google Trips [16]

Además de todo lo comentado anteriormente, en estas dos figuras procedentes de la aplicación Google Trips podemos observar el uso de pestañas dentro de una misma actividad para representar las distintas partes en las que se divide. Manteniendo de este modo al usuario en el mismo lugar, pero estructurando la información y acciones en vistas diferentes [29].

3.6.2. Wireframes y navegabilidad

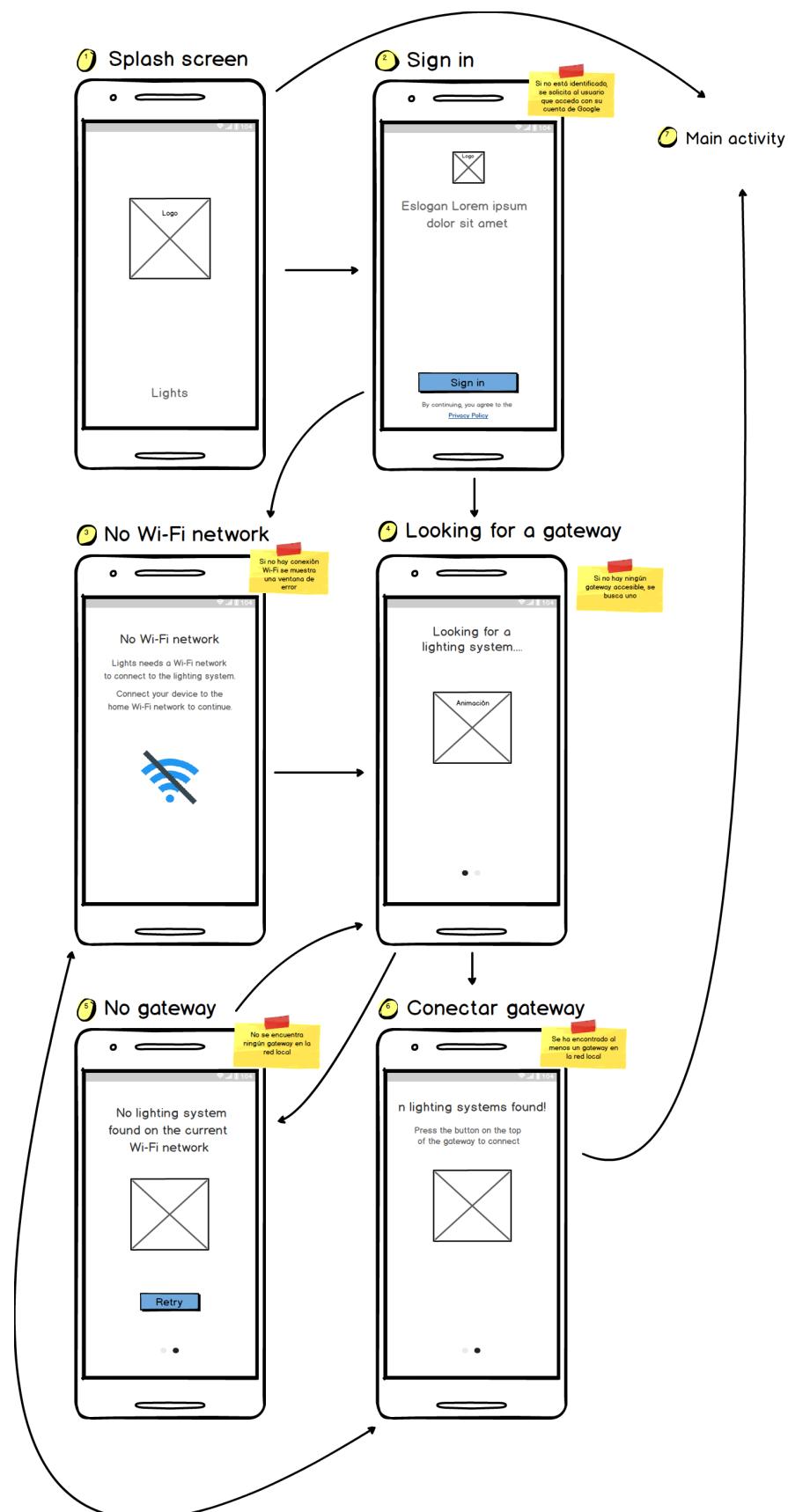


Figura 44: Wireframe 1 - Configuración

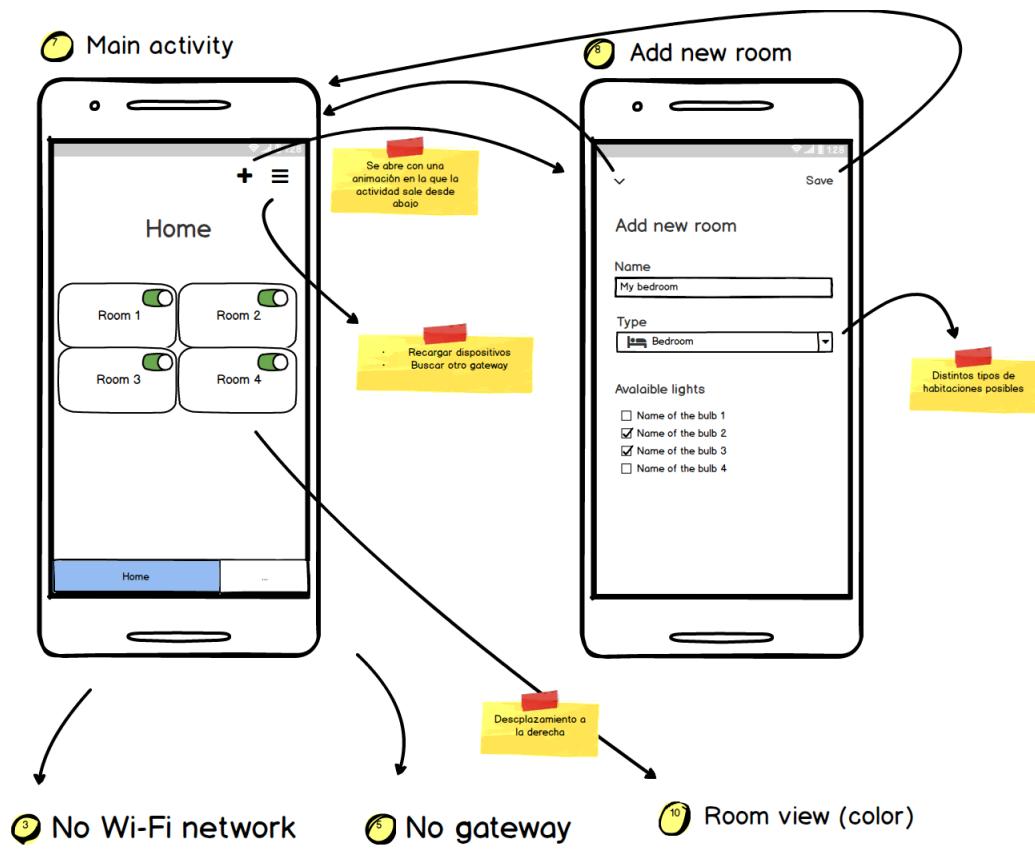


Figura 45: Wireframe 2 - Actividad principal

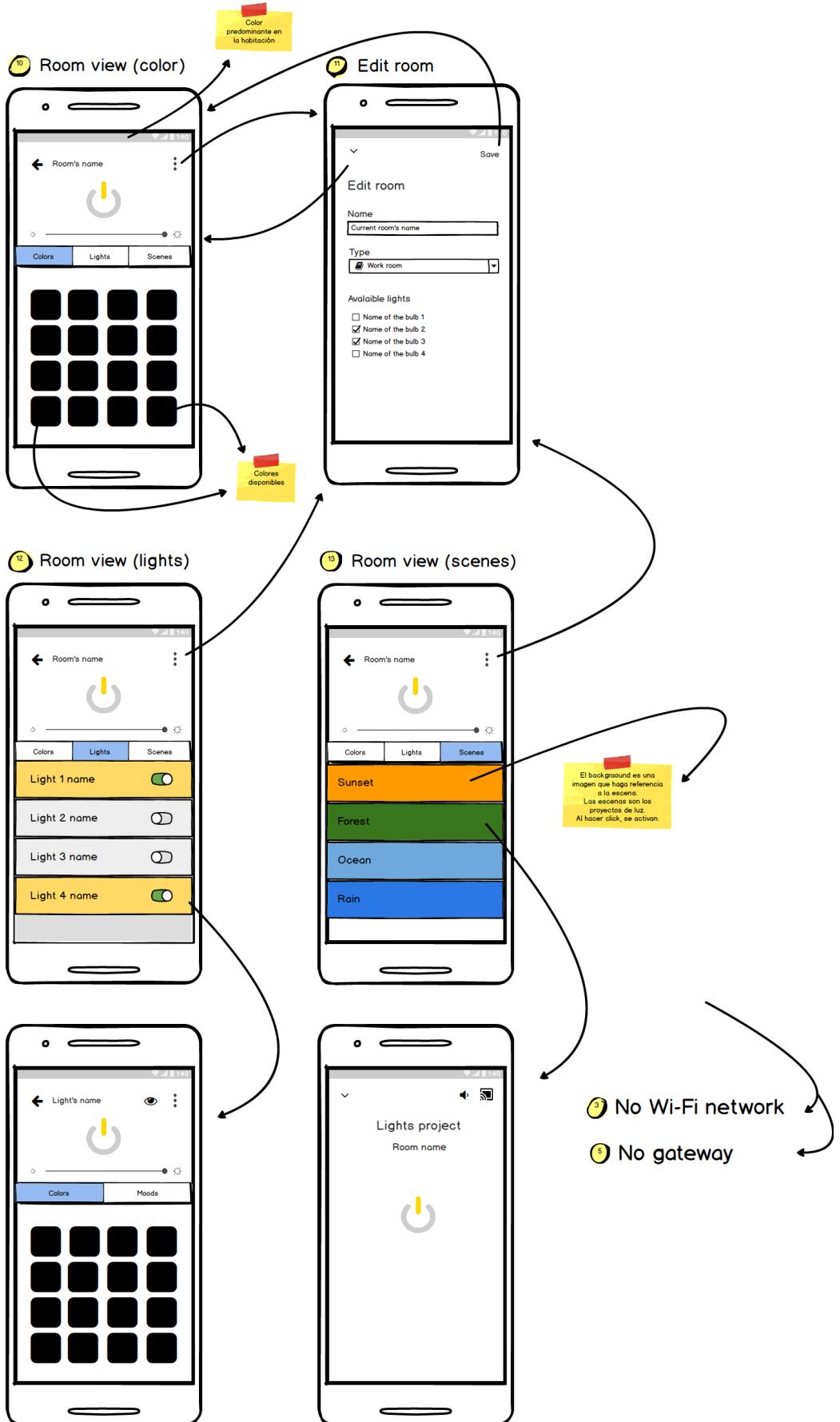


Figura 46: Wireframe 3 - Vistas de habitación, de dispositivo de luz, y de proyecto de luz.

Los wireframes se han realizado teniendo en cuenta todos los puntos comentados anteriormente, y cumpliendo con todos los casos de uso propuestos. Ha sido necesario borrar y rehacer algunas actividades para satisfacer todo lo propuesto durante el diseño y en base a la experiencia que nos ha brindado el prototipo. Esto nos servirá de guía posteriormente para implementar la interfaz en Android y también para visualizar la aplicación que vamos a desarrollar. Han sido hechos con Balsamiq <https://balsamiq.com>.

3.6.3. Diseño gráfico

Un problema recurrente en una aplicación móvil es el apartado gráfico, ya que las aplicaciones móviles necesitan impactar desde que se descarga por primera vez para que tenga una apariencia realmente profesional. Esto es algo que no se puede pasar por alto, y es tan importante como la funcionalidad de la aplicación. Un mal diseño gráfico nos llevaría a que los usuarios ni siquiera abran la aplicación entre los miles y millones de aplicaciones que hay en los mercados de los ecosistemas móviles. Las referencias comentadas en los apartados anteriores también destacan por tener bonitas imágenes flat que concuerdan con el estilo de la aplicación. Otra dificultad añadida es que es necesario disponer de las imágenes vectoriales para poder adaptarlas sin problema a diferentes resoluciones. Por ello, para este proyecto he optado por comprar casi todas las imágenes necesarias para el apartado gráfico. Las licencias de las imágenes compradas me permiten modificarlas y usarlas en productos comerciales.

Estas son algunas de las imágenes usadas:



Figura 47: Logo de la aplicación.



Figura 48: Baño



Figura 49: Dormitorio



Figura 50: Comedor



Figura 51: Estudio



Figura 52: Cocina



Figura 53: Otro

Los colores elegidos como colores principales de la aplicación, son:



Figura 54: Colores principales de la aplicación

3.7. Diagrama de flujo del sistema

Para entender mejor el proceso de inicio de la aplicación, se ha llevado a cabo un diagrama de flujo:

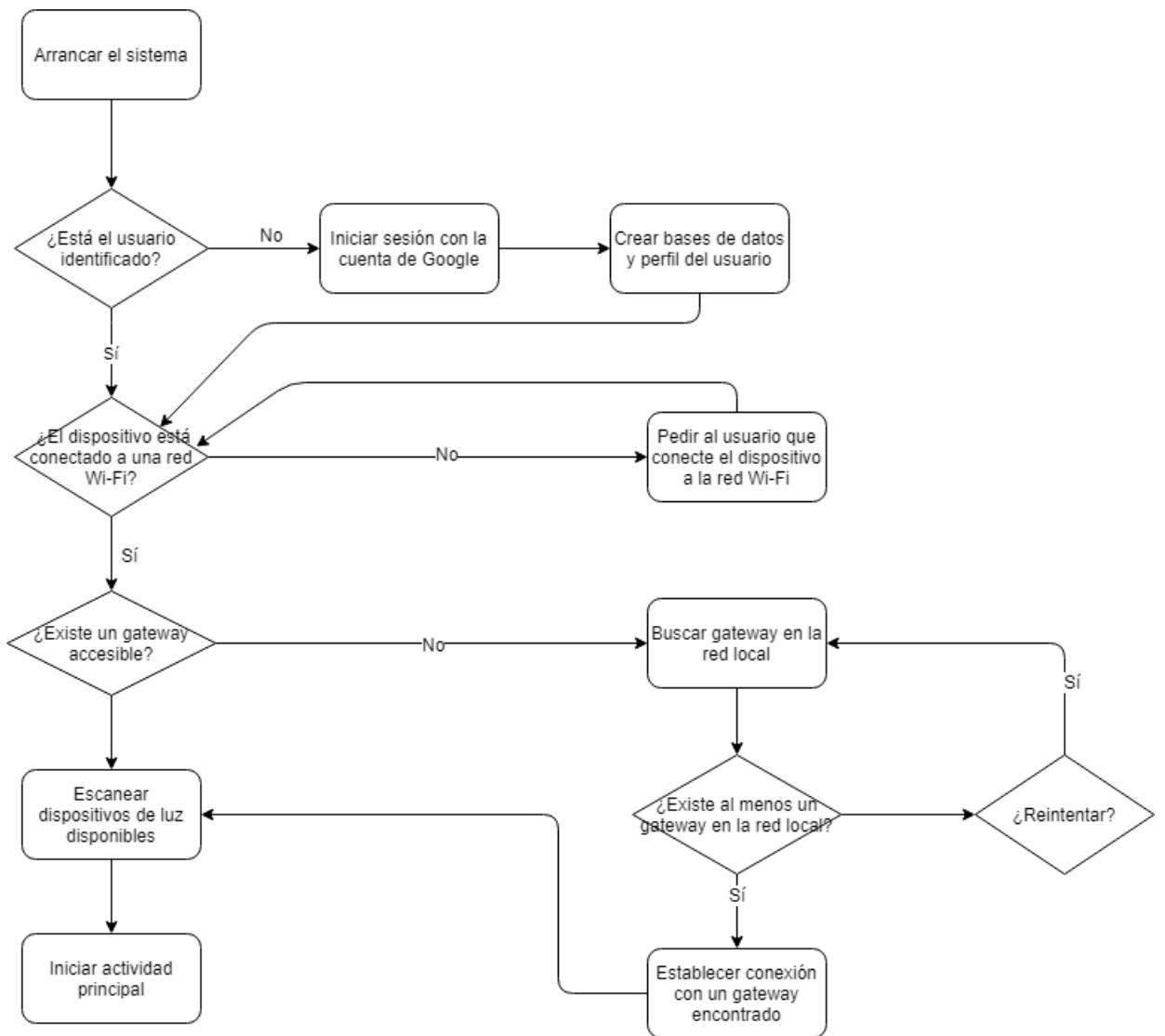


Figura 55: Diagrama de flujo arranque del sistema

Al arrancar el sistema debemos de tener en cuenta los puntos:

- El usuario tiene que iniciar sesión, aunque puede ser que ya lo haya hecho previamente.
- Es necesario que el sistema esté conectado a la red Wi-Fi.
- Hay que crear las bases de datos y el perfil del usuario si es la primera vez que usa la aplicación.
- Hay que comprobar si ya había un gateway conectado, y en caso de no haberlo hay que buscar uno disponible en la red.
- Puede ser que no se logre encontrar un gateway al que conectar la el sistema.

Todos los puntos señalados se han tenido en cuenta en el diagrama de flujo anterior, permitiendo de esta forma visualizar de una forma más clara el orden de ejecución y los hilos posibles.

3.8. Modelo conceptual

El siguiente diagrama muestra los elementos que compondrán el modelo del sistema. Lo cual nos ayuda a entender mejor el sistema y a verificar los requisitos.

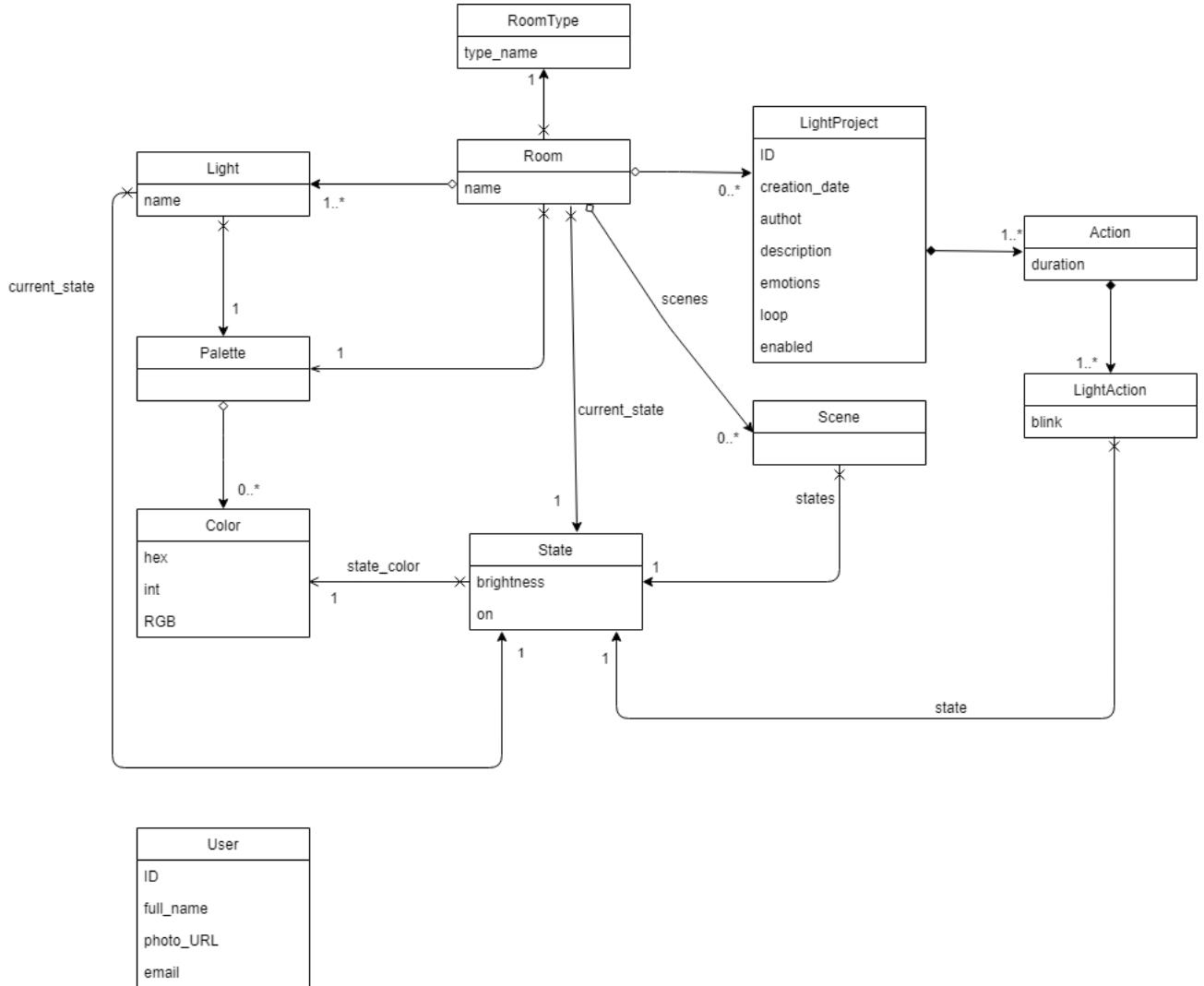


Figura 56: Modelo conceptual

De la figura anterior, podemos determinar lo siguiente:

- Una habitación está compuesta por:
 - Un conjunto de luces, y mínimo debe de tener una.

- Nombre.
 - Tipo de habitación.
 - Estado actual de la habitación (brillo, color, y encendido/apagado).
 - Una paleta de colores.
 - Una conjunto de proyectos de luz.
 - Una conjunto de escenas.
- Un dispositivo de luz, está compuesto por:
- Un estado actual (brillo, color, y encendido/apagado).
 - Una paleta de colores.
- Un estado se compone por:
- Encendio o apagado.
 - Nivel de brillo.
 - Color asociado.
- Un color se compone por los valores en hexadecimal, entero y RGB que lo definen.
- Una escena se compone por un estado concreto (brillo, color, y encendido/apagado).
- Una paleta de colores se compone de un conjunto de colores.
- Un usuario se compone por:
- Identificador.
 - Nombre completo
 - URL de la foto de perfil.
 - Email.
- Un proyecto de luz se compone por:
- Identificador.
 - Una fecha de creación.
 - El autor.
 - Descripción.
 - Lista de emociones.

- Bucle o fijo.
 - Activado o desactivado.
 - Una lista de acciones.
- Las acciones se componen de:
- Un tiempo de duración.
 - Un conjunto de estados para los dispositivos de luz, a los que hemos llamado ‘LightAction’.
- LightAction, se compone de:
- Parpadeo o fijo.
 - Un conjunto de estados.

En la implementación, se han producido los siguientes cambios:

- El concepto ‘paleta’ ha sido omitido, debido a que una paleta se puede representar simplemente por una lista de colores.
- Los proyectos de luz no están asociados a habitaciones concretas. Es posible lanzar el mismo proyecto de luz en cualquier habitación.
- Las escenas y los LightAction se componen de varios estados, debido a que es posible aplicar para una misma escena varios estados diferentes a los dispositivos de luz que haya presentes en la habitación.

3.9. Arquitectura del sistema

Se utilizará el conocido patrón arquitectónico Modelo-vista-controlador (MVC) como base para definir la arquitectura del sistema. De este modo se separan las elementos de la vista de Android, el modelo de datos ideado, y los controladores que manejan los anteriores datos, intentando reducir las dependencias y facilitando la escalabilidad y el mantenimiento.

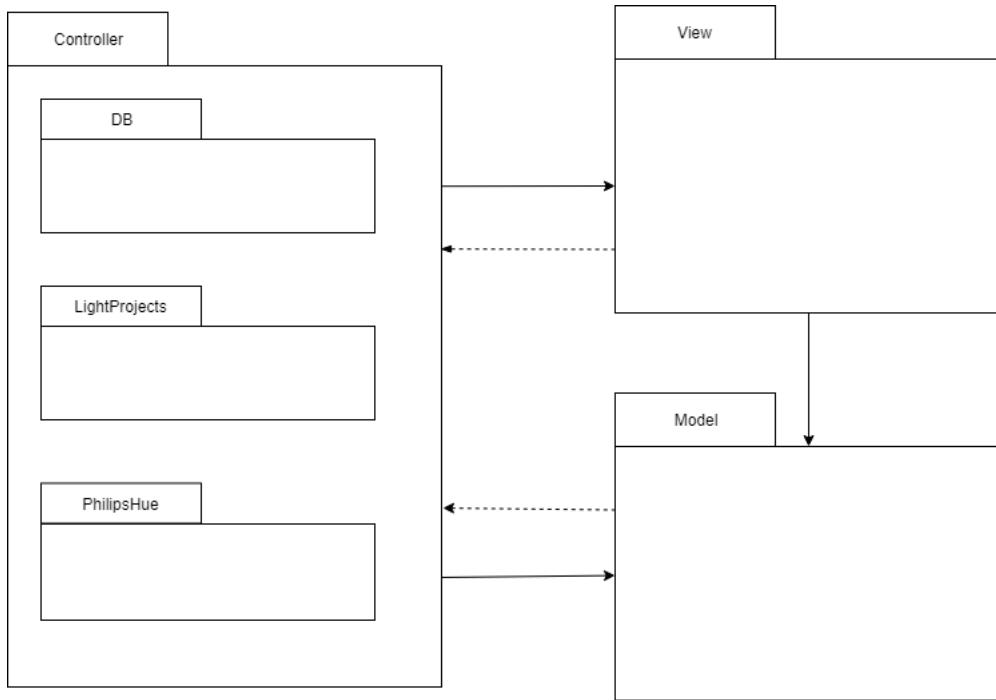


Figura 57: Arquitectura del sistema Lights en base a un diagrama de paquetes

A su vez, todos los assets (imágenes, textos, temas, archivos xml, etc), también están abstractos del código. Lo cual nos facilita la adaptación a otras resoluciones, el mantenimiento, y la localización.

La siguiente figura muestra el diagrama de despliegue del sistema:

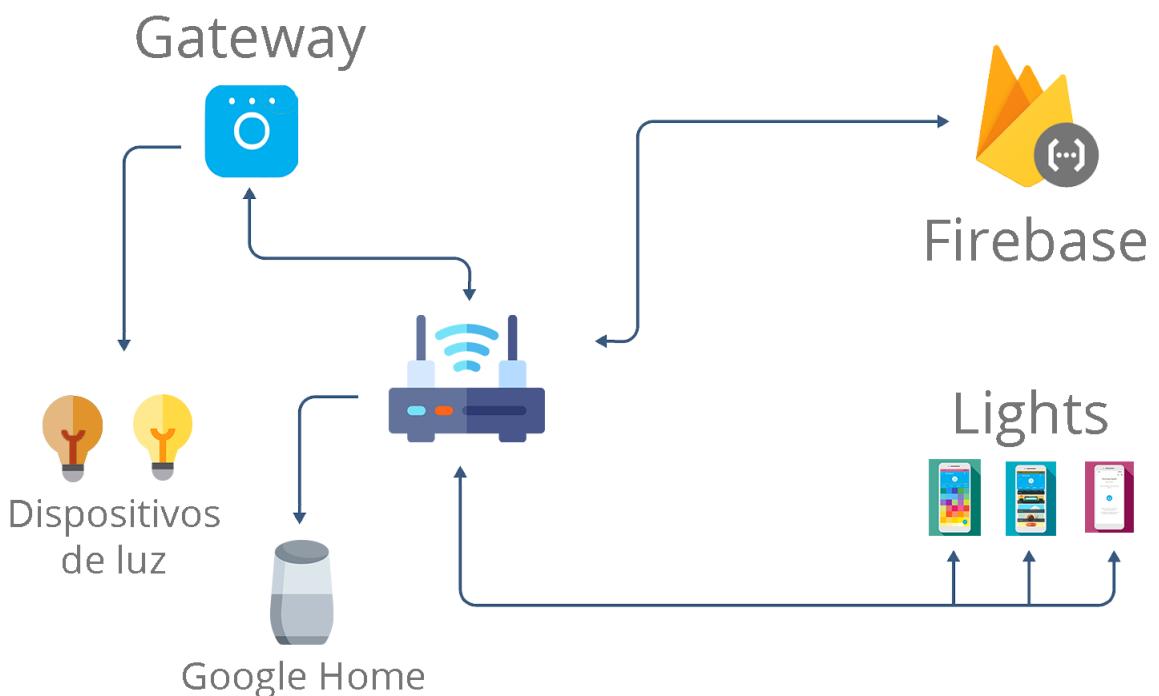


Figura 58: Arquitectura del sistema diagrama de despliegue

En el diagrama de despliegue se pueden ver las siguientes conexiones en el sistema:

- La aplicación Lights manda órdenes a través de la red Wi-Fi para controlar el sistema de iluminación y el altavoz. A su vez, recibe datos a través de la red Wi-Fi desde Firebase (esto se explica con más detalle en el apartado de Firebase), y del sistema de iluminación, el cual le indica los estados de los dispositivos de luz.
- Google Home actúa como altavoz, recibiendo a través de la red el sonido que debe reproducir.
- El gateway envía órdenes a los dispositivos de iluminación para cambiar sus estados, en función de lo que le indique Lights. La comunicación entre el gateway y los dispositivos de iluminación se realiza utilizando Zigbee.

3.10. Implementación

3.10.1. Tecnologías, herramientas y servicios utilizados

El proyecto se ha podido llevar a cabo utilizando las siguientes herramientas y servicios:

Philips Hue

Como ya se ha comentado, tras un análisis de las opciones del mercado finalmente se decidió utilizar Philips Hue como herramienta para el control de la iluminación del hogar. Para poder utilizar este sistema ha sido necesario familiarizarnos con la API, los SDKs disponibles, y elaborar nuestro propio wrapper que abstraiga el resto del código del uso directo de las opciones de esta herramienta, de modo que facilite la escalabilidad, y en el futuro puedan ser admitidos otros sistemas de iluminación del mercado.

Philips Hue proporciona una API capaz de mandar órdenes al gateway, permitiendo realizar todo el manejo y obtener todos los datos disponibles [13].

Un ejemplo del uso de la API sería el siguiente:

The screenshot shows the CLIP API Debugger interface. The URL field contains '/api/'. Below it are four buttons: GET, PUT, POST, and DELETE. The message body field contains the JSON string: {"devicetype": "test#javi"}. The command response field displays the following JSON output:

```
[{"success": {"username": "98IMRuJ1Yg7Gg6eHTdhBPxAq5JfJvJ9Zo6nB2cB8"}, "error": null}
```

The screenshot shows the CLIP API Debugger interface. The URL field contains '/api/98IMRuJ1Yg7Gg6eHTdhBPxAq5JfJvJ9Zo6nB2cB8/groups'. Below it are four buttons: GET, PUT, POST, and DELETE. The message body field is empty. The command response field displays the following JSON output:

```
{ "1": { "name": "My bedroom", "lights": [ "1", "2", "3" ], "sensors": [], "type": "LightGroup", "state": { "all_on": true, "any_on": true }, "recycle": false, "action": { "on": true, "bri": 100, "hue": 34205, "sat": 15, "effect": "none", "xy": [ 0.3676, 0.3805 ] } }}
```

Figura 59: Muestra de la API de Philips Hue - Registrando usuario

Figura 60: Muestra de la API de Philips Hue - Obteniendo lista de habitaciones

A partir de esta API se han elaborado multitud de SDKs para diferentes plataformas. Para Android existe un SDK oficial, pero además, en la página oficial también se ofrecen una lista de SDKs elaborados por otros desarrolladores.

Al principio se valoró la posibilidad de valernos de la API, construyendo nuestro propio SDK, pero pronto se descartó esta idea debido a que considero que llevaría un tiempo y un esfuerzo extra innecesario, habiendo otras opciones disponibles que ya satisfacen esta necesidad y están a nuestro alcance, y que por

lo tanto es preferible hacer uso de ellas. Finalmente se decidió utilizar el SDK oficial de Philips Hue, debido a que de este modo podemos echar mano de la documentación y los foros oficiales.

Tools and SDKs			
Deprecated – Hue SDK			Below is a list of 3rd party libraries, tools and software
 If you want to be added to this list, please contact us			
Hue SDK Guide for Java	Universal SDK	Java	Python
Hue SDK Guide for iOS and OS-X	CloudRail by Licode GmbH	Official Hue SDK for Android by Philips	beautifulhue by Allan Bunch
		Jue by Q42	hue-rgb_converter by benknight
	Android	HueControl by tachoknight	phue by Nathanael Lécaudé
	Weaver Lights by Weaver	Yet Another Hue API by Ville Saalo	pyhue by Alex Rodas
			Qhue by Quentin Stafford-Fraser
Arduino	JavaScript	huedump by JungleManSweep	
ArduinoHue by Ben Salinas	Anugular Hue Service by Christian Jellinghaus	home-assistant by Paulus Schouten	
Bash script	Hue Console by Steven Victor	Racket	
danrandom by Dan Radom	Hue Hacking by Bryan Johnson	Simple Hue Control by Bruce Steinberg	
lampe by André Klausnitzer	huepi by Arnd Brugman		
	jsHue by John Peloquin	Ruby	
Batch file	NodeJS	Hue by Sam Soffes	
hue-batch by I anielin		hue-lib by Rirkir A. Barkarson	

Figura 61: SDKs de Philips Hue [24]

Android

Se ha elegido Android como plataforma debido a que es uno de los sistemas móviles más usados y su programa de desarrollo es más económico y accesible. A día de hoy, por 25€ es posible tener una cuenta de desarrollador con la que poder publicar en Google Play de por vida; sin embargo, en su competidor más directo (iOS), es necesario pagar casi 100€ anuales y disponer de un Mac con una versión de iOS más o menos reciente.

Como entorno de desarrollo se ha utilizado Android Studio, debido a que es el entorno de desarrollo oficial y nos ofrece cada vez más ventajas, soporte y utilidades para el desarrollo Android.

A la hora de elegir el lenguaje de programación me he decantado por utilizar Java durante todo el proceso de desarrollo, debido a que ya tengo experiencia programando en este lenguaje tanto en Android como en otro tipo de plataformas; y porque la otra opción, Kotlin, ha comenzado a ser utilizada posteriormente al inicio de este proyecto. Aunque me he sentido tentado a utilizar este nuevo lenguaje para llevar a cabo algunas partes de esta segunda versión de la app (es posible combinar Java y Kotlin), pero finalmente todo el desarrollo hasta la fecha se ha llevado a cabo en Java por motivos de tiempo.

Me gustaría señalar que la segunda versión del proyecto, ha sido llevada a cabo utilizando AndoridX, lo cual es una versión mejorada de la clásica biblioteca Android Support Library. Buscando mejorar y reordenar el código de la biblioteca de Android, y mejorando la compatibilidad entre diferentes versiones de la API de Android, lo cual es algo que suele dar problemas a los desarrolladores [3]. Para ello se migró el proyecto anterior a esta nueva biblioteca.

Firebase

Un error que se cometió cuando se liberó la primera versión de la aplicación fue realizar todas las pruebas en un dispositivo con aspect ratio 16:9, sin tener en cuenta que ha aparecido otro estándar que ha cobrado bastante auge en el mercado basado en 18:9, además de la infinidad de resoluciones diferentes que podemos encontrar en la amplia variedad de dispositivos que utilizan Android. A pesar de haber seguido las recomendaciones para que la aplicación sea adaptable a otras resoluciones, algunas vistas no se veían exactamente como se había planificado. Viéndose, por ejemplo, el logo de la splash screen estirado, o la paleta de colores descuadrada. A partir de estas pruebas se replanificaron nuevas tareas para considerar dispositivos con resoluciones diferentes, centrándonos sobre todo en dispositivos con aspect ratio 16:9 y 18:9, ya que son los más comunes actualmente en los smartphones.

Al realizar las primera pruebas de la aplicación en distintos dispositivos al mismo tiempo, fue frustante comprobar la falta de coherencia encontrada incluso identificándose en la aplicación con la cuenta de Google en la que algunos aspectos del interfaz que configuraba en un dispositivo (como el tipo de habitación) no estaban en la otra configuración a pesar de que la habitación existiera. Y con esto me refiero al punto de vista como usuario de la aplicación, un usuario que dispone de dos dispositivos diferentes en su casa, y se ve obligado a realizar la misma configuración dos veces (o que formatea el dispositivo, y pierde algunos datos de la casa pero otros no). Esto es debido a que únicamente se registra la habitación en el gateway, y por lo tanto los demás dispositivos al obtener la lista de habitaciones conocen la existencia de esta, pero no el tipo de habitación ya que es un valor que hemos añadido nosotros. Por lo tanto es necesario, si deseamos conservar configuraciones como esta, utilizar una base de datos en la nube para almacenar más información descriptiva de la habitación.

Además, el motivo más importante por el que es necesario utilizar una base de datos externa tiene que ver con el objetivo de que la aplicación en el futuro sea capaz de recopilar datos de los usuarios. Datos de las emociones detectadas y del tipo de usuario, con el objetivo de que puedan servir para tomar muestras en este área de estudio.

Firebase es una base de datos cloud NoSQL que nos soluciona estos problemas, y además nos ofrece ventajas importantes, como el hecho de que tiene una funcionalidad realtime que permite notificar a los usuarios en caso de que se produzcan cambios en los datos [11]. Con lo cual, podemos llevar a cabo una actualización de la interfaz de la aplicación en tiempo real, permitiéndonos ver en todos los dispositivos a la vez los cambios que se produzcan en uno de ellos. Proporcionándonos de este modo coherencia en los datos disponibles.

The screenshot shows the Firebase Firestore interface. On the left, there's a sidebar with a 'lights-1562593193891' project. Under 'users', a document named 'lfjCpvV5JslrgktxvC0' is selected. This document contains the following fields:

- display_name: "Javier Izquierdo Vera"
- email: "javierizquierdovera@gmail.com"
- id: "TVx5lseZrkcvZZ7INmH2LEc7JX2"
- photo_URL: "https://lh5.googleusercontent.com-/ABD00fHseZA/AAAAAAAAl/AAAAAAA_4U/hxlzfa1tWng/s96-c/photo.jpg"

Figura 62: Uso de Firebase en la aplicación

Como conclusión podemos indicar que Firebase a pesar de llevar mucho tiempo en el mercado, sorprende por su sencillez y las posibilidades que ofrece. Es, por tanto, interesante utilizarlo para el desarrollo de aplicaciones móviles.

Cast Application Framework y altavoces

Al iniciar el proyecto se pensó en utilizar libremente Google Home para enlazarlo a la aplicación y servirme de este dispositivo como altavoz, pero no ha sido así. El SDK de Google Home a día de hoy no puede utilizarse libremente con fines comerciales sin solicitarlo previamente. Y aunque no fuera con fines comerciales, no proporciona las utilidades que se necesitaban en su kit de desarrollo. Sin embargo, existe otra herramienta de Google que sí que se adapta a las necesidades de utilizar Google Home como altavoz, y que además nos ofrece la ventaja de que no es exclusiva de este dispositivo, sino de cualquier dispositivo de Google capaz de realizar streaming (incluyendo así, por ejemplo, Google Chromecast). Esta herramienta es Cast Application Framework [8].

Una vez configurado este SDK e incluido dentro de nuestra aplicación, nos ofrece todo lo necesario para detectar dispositivos que cumplan con la tecnología Google Cast en la red local, y así enviarles el contenido que queremos que reproduzcan.

Para utilizar Cast Application Framework hemos tenido que registrar una cuenta como desarrollador de este servicio, para lo cual se ha tenido que pagar 5€. Es una cuenta vitalicia por lo que es posible utilizar dicha tecnología de forma ilimitada. Una vez registrada la cuenta, se tiene que registrar la aplicación en el servicio indicando una serie de detalles sobre la aplicación, incluyendo el uso

que se le dará. Al registrar la aplicación se obtiene un ID con la que se puede configurar el SDK para usarlo en esta aplicación.

Cabe comentar que la aplicación no dependerá de tener aparatos con tecnología Google Cast en el hogar, ya que el propio dispositivo Android tiene un altavoz que puede usarse, y esto es algo de lo que sí que podrán disponer la mayoría de los usuarios. Así que la reproducción de sonido en streaming es opcional en la aplicación, y está implementada de forma que el sistema es extensible a otra tecnología diferente. Utilizándose por defecto el altavoz del dispositivo Android. Aunque, siempre que se disponga, es preferible utilizar estos altavoces externos para lograr mejor calidad.

AmbilWarna

Una de las necesidades de la paleta de colores personalizada, es permitir a los usuarios añadir los colores que deseen (además de borrar los que no). Para ello ha sido necesario elaborar un color picker que puedan utilizar los usuarios.

En este caso se ha decidido hacer uso de una biblioteca ya existente, con el objetivo de no perder tiempo reinventando algo que ya ha sido elaborado y que puedo utilizar sin problema, esta biblioteca es AmbilWarna [2].

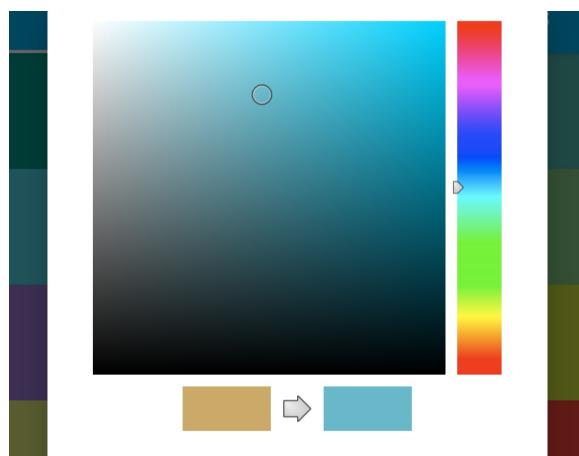


Figura 63: Uso de la biblioteca AmbilWarna como color picker

3.10.2. Proyectos de luz

Un proyecto de luz es una receta que le indica a los dispositivos de luz qué color y brillo deben de tener en cada momento. Permitiendo además incluir sonido, y otros elementos como imagen, nombre, y descripción. El objetivo es que el usuario pueda definir sus propias recetas si lo deseas, y lanzarlas cuando le resulte conveniente, con la finalidad de ambientar la habitación del modo definido.

Para especificar los proyectos de luz se ha usado el formato JSON. De modo que cada proyecto de luz es un archivo independiente, que la aplicación es capaz de leer e interpretar.

Estos son los atributos que pueden definirse en el archivo:

- ‘**name**’: Nombre del proyecto de luz.
- ‘**author**’: Autor del proyecto de luz.
- ‘**description**’: Descripción del proyecto.
- ‘**date**’: Fecha de creación. Este parámetro puede omitirse.
- ‘**emotions**’: Lista de emociones que pretende transmitir esta ambientación del hogar.
- ‘**loop**’: Si está en ‘true’, el proyecto de luz se ejecutará en bucle. Este parámetro puede omitirse, por defecto es ‘false’.
- ‘**enabled**’: Si está en ‘true’, el proyecto de luz se leerá y se sumará a la lista de proyectos disponibles, en el caso opuesto se ignorará. Este parámetro puede omitirse, por defecto es ‘true’.
- ‘**image**’: Imagen que representa el proyecto de luz. Se recomienda que tenga una resolución de: 626x207px. La imagen se indica mediante una ruta relativa al directorio de proyectos de luz. Este parámetro puede omitirse, se asignará una imagen por defecto en caso de no encontrarse la imagen.
- ‘**sound**’: Sonido que se reproducirá durante la ejecución del proyecto de luz. La ruta del sonido debe de ser relativa al directorio de proyectos de luz, aunque también puede ser una URL. Para que el sonido sea reproducible en streaming, es necesario que esté alojado en algún servidor. Este parámetro puede omitirse, por defecto no hay sonido.
- ‘**actions**’: Lista de acciones que se llevarán a cabo. Las acciones se componen de dos parámetros: ‘**duration**’ y ‘**light_actions**’.
 - ‘**duration**’: Es el tiempo que durará la acción que se está ejecutando.
 - ‘**light_actions**’: Lista de estados que adoptarán las luces durante la acción que se está ejecutando. Se compone de los parámetros: ‘**brightness**’, ‘**color**’, ‘**blink**’.
 - ‘**brightness**’: Brillo que tendrá el dispositivo de luz.
 - ‘**color**’: Color que tendrá el dispositivo de luz.
 - ‘**blink**’: Si es ‘true’, la luz del dispositivo parpadeará. En caso de ser ‘false’ la luz será fija. Este parámetro se puede omitir, por defecto tiene el valor ‘false’.

Este es un ejemplo de un posible proyecto de luz.

```
{
  "name": "Hearth",
  "author": "Javier Izquierdo Vera",
  "description": "The crackling of the flames at home.",
```

```

    "date": "01/09/2019",
    "emotions": ["happy", "calm"],
    "loop": true,
    "enabled": true,
    "image": "hearth.png",
    "sound": "http://javierizquierdovera.com/lights-app/resources/
              lights_projects/hearth.mp3",
    "actions": [
        {
            "duration": 2,
            "light_actions": [
                {
                    "brightness": 255,
                    "color": "#8F1700"
                },
                {
                    "brightness": 50,
                    "color": "#e64602",
                    "blink": true
                },
                {
                    "brightness": 100,
                    "color": "#ffc800"
                }
            ]
        },
        {
            "duration": 2,
            "light_actions": [
                {
                    "brightness": 255,
                    "color": "#8F1700"
                },
                {
                    "brightness": 100,
                    "color": "#e64602"
                },
                {
                    "brightness": 50,
                    "color": "#ffc800",
                    "blink": true
                }
            ]
        }
    ]
}

```

No importa si el número de dispositivos de luz no concuerda con el número de

estados para los dispositivos definidos, ya que en caso de ser menos dispositivos solo se ejecutarán los estados que sean posibles, y en caso de ser un número superior se repetirán estados. Si en una habitación no se desea que el proyecto de luz tenga efecto en un dispositivo en concreto, bastará con apagarlo.

Los proyectos de luz pueden ubicarse en la carpeta de assets (por lo que serán incluidos en el paquete de la aplicación al compilarla), o en un directorio del dispositivo. Por defecto ese directorio es '/storage/emulated/0/Lights/light_projects', lo cual equivale al directorio raíz de la memoria interna del dispositivo:



Figura 64: Directorio externo de proyectos de luz

Ese directorio está preparado para poder configurarse, ya que se almacena en la base de datos junto a las preferencias del usuario, pero aún no es posible configurarlo desde la interfaz de la aplicación.

En la figura anterior también puede observarse como en el directorio que la aplicación ha generado en la ruta especificada para almacenar los proyectos de luz, hay un archivo ya creado. Ese proyecto de luz es un proyecto de muestra que la aplicación genera en caso de no existir ningún otro proyecto, por defecto está desactivado.

¿Cómo se ejecutan los proyectos de luz?

En la primera aproximación que se llevó a cabo de la implementación de los proyectos de luz, estos se ejecutaban en una hebra. De modo que se creaba una tarea asíncrona y se lanzaba. El problema surgió cuando se constató que para que esta hebra lanzada se mantenga en ejecución, la aplicación debe de mantenerse constantemente en primer plano, lo cual obligaba al usuario a no poder salir de la aplicación en caso de querer mantener la ejecución del proyecto. Esta

implementación no parecía ser suficientemente elegante por lo que se buscó si Android disponía de algún servicio que facilitase esta labor de ejecutar acciones con una determinada frecuencia, como el sistema de alarmas (AlarmManager) [1].

Tras consultar la documentación de Android acerca de esta utilidad, se preparó un proyecto para poder realizar pruebas, pero no se logró tener el éxito esperado. En la conocida web de preguntas y respuestas de desarrollo ‘Stack Overflow’ (<https://stackoverflow.com/questions/57533376/how-to-call-a-specific-class-method-using-alarmmanager>) se pudo encontrar la respuesta. En una respuesta se indicaba que no debía usarse AlarmManager para este cometido, sino WorkManager. Después de comprobar el funcionamiento de WorkManager se constató que tampoco solucionaba el problema, ya su objetivo es asegurar la ejecución de una acción, incluso si la aplicación se cierra completamente. La utilidad de WorkManager está más cercana a tareas de bajo nivel o que se ejecutan sin hacer consciente al usuario de ello.

Finalmente, en el blog oficial de desarrollo de Android, en el que se hace un repaso de las diferentes herramientas con las que cuenta Andoid para ejecutar una tarea en background [21], encontramos un artículo que commenta las distintas alternativas. Resumiendo en la siguiente tabla tenemos:

Use Case	Examples	Solution
Guaranteed execution of deferrable work	<ul style="list-style-type: none"> Upload logs to your server Encrypt/Decrypt content to upload/download 	WorkManager
A task initiated in response to an external event	<ul style="list-style-type: none"> Syncing new online content like email 	FCM + WorkManager
Continue user-initiated work that needs to run immediately even if the user leaves the app	<ul style="list-style-type: none"> Music player Tracking activity Transit navigation 	Foreground Service
Trigger actions that involve user interactions, like notifications at an exact time.	<ul style="list-style-type: none"> Alarm clock Medicine reminder Notification about a TV show that is about to start 	AlarmManager

Figura 65: Tabla resumen ejecución de tareas en background en Android [21]

I need to run a task in background, how should I do it?

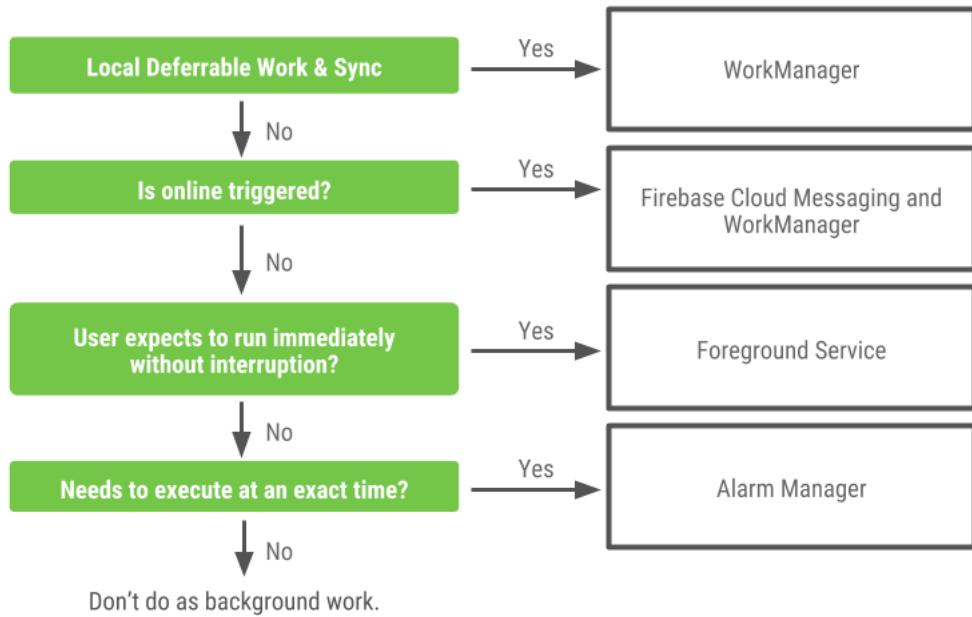


Figura 66: 'I need to run a task in background, how should I do it' [21]

De modo que, la conclusión para nuestro caso de uso es utilizar un Foreground Service. Además, este tipo de servicio se mantiene en ejecución mostrando siempre una notificación al usuario, de tal forma que el usuario es consciente en todo momento de lo que se está ejecutando, y además puede recuperar el control [28]. Se llama foreground, por el hecho de que se mantiene visible para el usuario, y es justo lo que necesitamos para este caso.

Un nuevo problema surge de esta conclusión, y es el hecho de que si queremos que las acciones se ejecuten en una frecuencia de tiempo concreta, tenemos dos opciones: ser notificados cuando pase ese tiempo concreto, o poner en pausa la ejecución durante ese tiempo. Un servicio de Android no se puede poner en pausa, ya que es un turno de ejecución que se comparte con otras aplicaciones. Por lo tanto, si lo ponemos en pausa, provocaría una situación de bloqueo. De modo que para solucionar esto, se optó por una solución mixta, utilizando la hebra comentada anteriormente, pero en este caso a través del Foreground Service, quien se encarga de lanzarla e intervenir en ella, de modo que ya sí se mantiene viva a pesar de poner la aplicación en segundo plano, y también se puede controlar mediante la notificación que comentábamos en el párrafo anterior.

4. Dirección y gestión del proyecto

4.1. Métodología de trabajo, planificación y herramientas

La organización y planificación del proyecto es una parte fundamental para que un proyecto software pueda llevarse a cabo con éxito. El informe Chaos nos aporta motivos más que suficientes para tener conciencia de esto [9]. Por lo tanto, este proyecto se ha tratado de llevar a cabo siguiendo una metodología ágil basada en Scrum. Mediante un proceso de desarrollo iterativo basado en el prototipado y las revisiones constantes.

Durante el proceso de diseño, hemos realizado sprints de dos semanas llevando a cabo una reunión al final de cada sprint. Durante esos sprints se ha ido desarrollando a la vez un primer prototipo capaz de interactuar con el sistema de iluminación Philips Hue. Seguido de las primeras pruebas para ejecutar los proyectos de luz. En las reuniones primero llevábamos a cabo una revisión del trabajo realizado, permitiendo así evaluar los fallos y descubrir problemas y nuevas necesidades, para luego fijar las nuevas tareas a realizar durante el siguiente sprint. Otra parte fundamental de todo esto ha sido tratar de estimar el tiempo que me va a llevar hacer cada tarea y luego compararlo con el tiempo que realmente me ha llevado. Esto nos ayuda a tomar conciencia de en qué estamos invirtiendo el tiempo (lo cual nos permite detectar problemas en las tareas) además de aprender a estimar mejor los tiempos y por lo tanto ser más realistas a la hora de planificar el trabajo.

Para llevar a cabo la tarea de gestión y documentación de las reuniones y tareas fijadas, se ha utilizado la plataforma de gestión de proyectos Redmine.

TFG Javier Izquierdo - Interaccion movil

+ Vistazo Actividad Peticiones Tiempo dedicado Gantt Calendario Noticias Documentos Wiki Archivos Configurac

Peticiones

Filtros Estado abierta Añadir el filtro

Aceptar Anular Guardar

#	Tipo	Estado	Prioridad	Asunto	Asignado a
1688	Tareas	Nueva	Normal	Elaborar el diagrama conceptual	Javier Izquierdo Vera
1687	Tareas	Nueva	Normal	Completar la tabla de preguntas	Javier Izquierdo Vera
1686	Tareas	Nueva	Normal	Completar la lista de casos de uso	Javier Izquierdo Vera
1685	Reuniones	Resuelta	Normal	Reunión	
1590	Tareas	Resuelta	Normal	Definición técnica de las acciones	Javier Izquierdo Vera
1589	Reuniones	Resuelta	Normal	Reunión	
1535	Tareas	Resuelta	Normal	Buscar acciones para las emociones seleccionadas	
1534	Reuniones	Resuelta	Normal	Reunión	
1504	Tareas	Nueva	Normal	Possibilidad de incluir un sensor en las habitaciones	
1503	Tareas	Nueva	Normal	Permitir detectar bombillas si se conectan nuevas	Javier Izquierdo Vera
1502	Tareas	Resuelta	Normal	Perfeccionar el JSON para crear proyectos de luz	
1501	Reuniones	Resuelta	Normal	Reunión	
1482	Tareas	Resuelta	Normal	Revisar el estado de la ingeniería del software del proyecto	
1481	Tareas	Resuelta	Normal	Permitir especificar las acciones a realizar con las emociones elegidas en un JSON	
1480	Tareas	Nueva	Normal	Estudiar sistemas de formalización de emociones e identificar las emociones con las que vamos a trabajar	
1479	Tareas	En curso	Normal	Desarrollar app "básica"	Javier Izquierdo Vera
1478	Reuniones	Resuelta	Normal	Reunión	
1456	Tareas	Resuelta	Normal	Listado de casos de uso para el sistema de iluminación	Javier Izquierdo Vera

Figura 67: Redmine - Plataforma de gestión del proyecto

Estado: Nueva
 Prioridad: Normal
 Asignado a: Javier Izquierdo Vera
 % Realizado: 100%
 Tiempo estimado: 6.00 h
 Tiempo dedicado: 6.00 h

Descripción Citar
 Realizar un documento que resuma y analice una serie de soluciones existentes en el mercado acerca del control de Philips Hue.

Resultado:
 Un documento con una lista de aplicaciones analizadas, con su respectivo análisis, y una tabla comparativa que ofrezca un vistazo rápido y útil para tomar conclusiones.

Subtareas Añadir

Figura 68: Redmine - Ejemplo de tarea en Redmine

Cabe mencionar también, que para trabajar he utilizado la llamada Técnica Pomodoro. Esta técnica consiste en trabajar completamente centrado en una tarea durante 25 minutos, seguido de una pausa de 5 minutos. Esto me ha ayudado a concentrarme en realizar la tarea que me proponía teniendo mayor conciencia del tiempo que pasaba, y sin dejar de tener en cuenta que durante esos 25 minutos no podía distraerme en otras cosas.

Además de Redmine, como medio de comunicación hemos utilizado Skype, lo cual me ha permitido aclarar dudas cuando ha sido necesario, o tener reuniones cuando no ha sido posible vernos en persona.

Hemos utilizado un software controlador de versiones para compartir los archivos y llevar un control de los cambios realizados, teniendo de este modo una

copia de seguridad de todas las versiones. Este software es Subversion. Aunque por mi cuenta también he utilizado Git y GitHub para realizar el control de versiones de la implementación del proyecto.

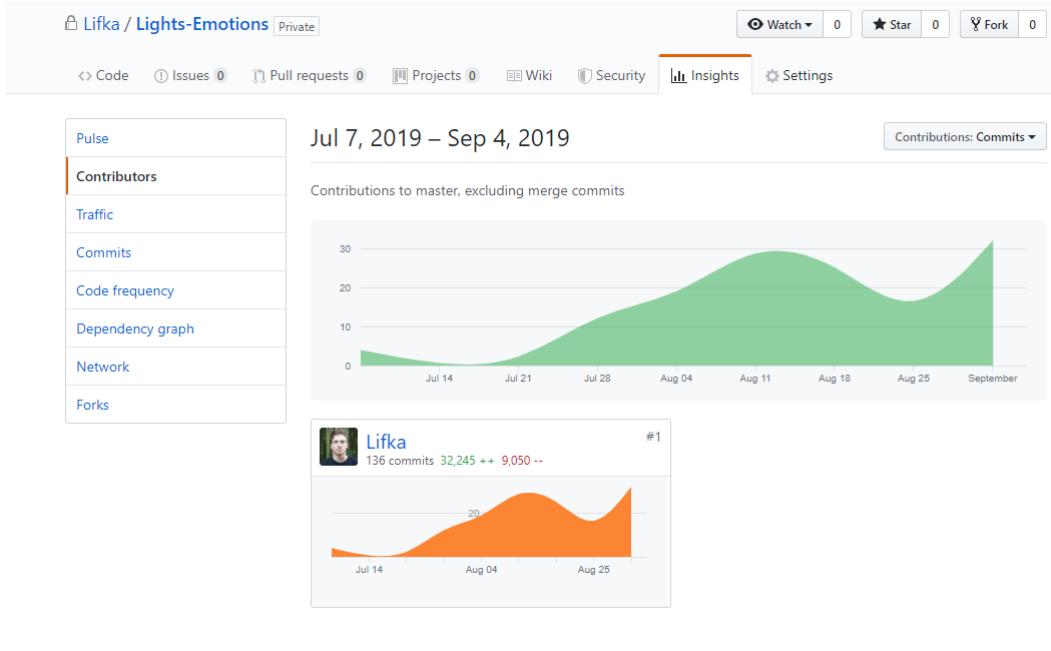


Figura 69: Contribuciones a la segunda versión de la aplicación en el repositorio de GitHub.

Los principales milestone en el proyecto han sido los siguientes:

- Realizar una aplicación simple capaz de conectarse al sistema Philips Hue.
- Crear un sistema utilizando el SDK de Philips Hue que nos permita llevar a cabo cualquier acción con el sistema de iluminación.
- Primera versión del modo manual.
- Implementación de los proyectos de luz.
- Segunda versión del modo manual, incluyendo los proyectos de luz y solucionando todos los fallos de la primera versión.

4.1.1. Conclusiones

A pesar de todo lo anterior, el proyecto se ha llevado a cabo de forma interrumpida debido a que se ha tenido que compatibilizar el desarrollo del proyecto con un trabajo, por lo que se ha extendido en el tiempo más de lo esperado y era algo con lo que yo no contaba. No obstante, ha permitido perfeccionar la labor de estimar los tiempos y aprender a gestionar mejor el tiempo y la planificación de las tareas. Se ha aprendido a establecer objetivos, a dividirlos, y a organizar esas

tareas con el tiempo que disponga, centrándome en cumplir objetivos realistas y a corto plazo.

Dividir las tareas y planificar objetivos en el desarrollo ha sido muy positivo, ya que facilita conocer el producto que se está desarrollando, el estado del desarrollo en el que se encuentra, y la replanificación de tareas en los siguientes sprints para corregir cualquier discrepancia.

4.2. Estimación de costos

Otro punto importante en la planificación de un proyecto para que sea viable, es ser capaces de establecer un presupuesto realista y acorde con los objetivos y el tiempo de desarrollo. De este modo podemos prever los gastos y ser conscientes de cómo de rentable es el proyecto, y en qué tiempos tenemos que finalizarlo para no superar el presupuesto previsto.

Para este caso, vamos a utilizar el modelo COCOMO II, basándonos en la experiencia de desarrollo de esta primera parte del proyecto que hemos llevado a cabo.

Este modelo se puede utilizar basándonos en las líneas de código del proyecto. Para obtener las líneas de código en Android Studio he utilizado el plugin Statistic [30] y el resultado de las líneas escritas en Java (restando líneas en blanco, comentarios, y lo que yo no he escrito) es de: 38.563 líneas de código.

Vamos a suponer que la parte desarrollada hasta ahora es el 75 % del total deseado. Falta llevar a cabo el modo inteligente, y algunas de las funciones que se incluyen en el Roadmap. Si hasta ahora hemos escrito el 75 %, faltan 12854 líneas de código.

4.2.1. COCOMO II: Input

Software Size Sizing Method **Source Lines of Code ▾**

SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New <input type="text" value="12854"/>						
Reused <input type="text" value="2000"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="20"/>	<input type="text" value="4"/>		
Modified <input type="text" value="1000"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="3"/>	<input type="text" value="10"/>	<input type="text" value="0.1"/>

Software Scale Drivers

Precededness	<input type="text" value="Nominal ▾"/>	Architecture / Risk Resolution	<input type="text" value="Nominal ▾"/>	Process Maturity	<input type="text" value="Nominal ▾"/>
Development Flexibility	<input type="text" value="Nominal ▾"/>	Team Cohesion	<input type="text" value="Nominal ▾"/>		

Software Cost Drivers

Product	Personnel	Platform	
Required Software Reliability	<input type="text" value="Nominal ▾"/> Analyst Capability	<input type="text" value="Nominal ▾"/> Time Constraint	<input type="text" value="Nominal ▾"/>
Data Base Size	<input type="text" value="Nominal ▾"/> Programmer Capability	<input type="text" value="Nominal ▾"/> Storage Constraint	<input type="text" value="Nominal ▾"/>
Product Complexity	<input type="text" value="Nominal ▾"/> Personnel Continuity	<input type="text" value="Nominal ▾"/> Platform Volatility	<input type="text" value="Nominal ▾"/>
Developed for Reusability	<input type="text" value="Nominal ▾"/> Application Experience	<input type="text" value="Nominal ▾"/>	
Documentation Match to Lifecycle Needs	<input type="text" value="Nominal ▾"/> Platform Experience	<input type="text" value="Nominal ▾"/>	
	<input type="text" value="Language and Toolset Experience"/>	<input type="text" value="Nominal ▾"/>	

Project	
Use of Software Tools	<input type="text" value="Nominal ▾"/>
Multisite Development	<input type="text" value="Nominal ▾"/>
Required Development Schedule	<input type="text" value="Nominal ▾"/>

Maintenance

Software Labor Rates
Cost per Person-Month (Dollars)

Figura 70: COCOMO II Input

4.2.2. COCOMO II: Resultados

Results

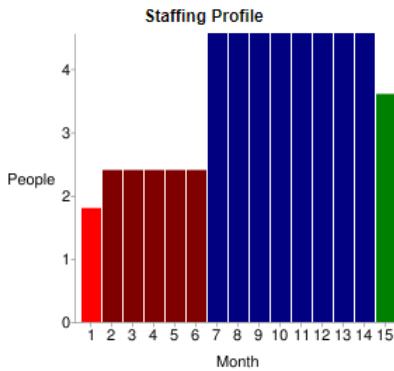
Software Development (Elaboration and Construction)

Effort = 50.1 Person-months
 Schedule = 13.4 Months
 Cost = \$75201

Total Equivalent Size = 13186 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	3.0	1.7	1.8	\$4512
Elaboration	12.0	5.0	2.4	\$18048
Construction	38.1	8.3	4.6	\$57153
Transition	6.0	1.7	3.6	\$9024



Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.4	1.4	3.8	0.8
Environment/CM	0.3	1.0	1.9	0.3
Requirements	1.1	2.2	3.0	0.2
Design	0.6	4.3	6.1	0.2
Implementation	0.2	1.6	13.0	1.1
Assessment	0.2	1.2	9.1	1.4
Deployment	0.1	0.4	1.1	1.8

Your output file is http://csse.usc.edu/tools/data/COCOMO_September_4_2019_07_41_48_38373.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Figura 71: COCOMO II Resultados

4.2.3. COCOMO II: Análisis de los resultados y conclusiones

Los resultados obtenidos son bastante desorbitados. Se estima más de un año de desarrollo para un equipo de muchas personas. Sin embargo, ya ha sido realizada una parte mayor del proyecto y el tiempo y el esfuerzo requerido ha sido muy distinto al resultado del modelo COCOMO II. Esto puede deberse a que las variables que utiliza COCOMO II para calibrarse se toman en base a proyectos anteriores, y como este es un proyecto sin precedentes, los datos introducidos son una suposición. Además de eso no deja de ser una estimación basada en unas métricas.

Mi conclusión es que para elaborar un presupuesto más coherente deberíamos basarnos en lo aprendido a la hora de estimar el tiempo de las tareas. Dividir todo el proyecto en objetivos, y desglosar y realizar las estimaciones temporales de las tareas en base a nuestra experiencia. Teniendo en cuenta la dificultad de las tareas, que pueden surgir imprevistos, y el nivel de conocimiento previo con el que contemos para realizar las tareas. Además de incluir el precio de las licencias necesarias, equipo, y cualquier otro gasto que se pueda producir.

5. Publicación y mercado

5.1. Monetización

Existen muchas formas posibles para obtener una remuneración con las aplicaciones publicadas en Android, tantas como podamos idear. Aún así, existen unos modelos definidos que podemos adoptar en función de las características del proyecto y del público al que nos estemos dirigiendo [22]. Estos modelos son:

- Pago por descarga: El usuario realiza un pago único para disfrutar de la aplicación. Por lo que en la mayoría de los casos el usuario ya espera disponer del producto al completo.
- Publicidad: Incluir banners o vídeos publicitarios de otros productos. De forma que obtengamos algún tipo de remuneración por ello. Google ofrece el servicio Google AdMob el cual nos permite incluir publicidad de forma casi automática en función de las características de nuestra aplicación, los gustos de los usuarios, y las empresas que estén dispuestas a pagar por la inclusión de su anuncio [18].
- Compras en la aplicación: Ofrecer algún tipo de contenido al usuario por el que pueda estar dispuesto a realizar un desembolso para mejorar los beneficios que le ofrece la aplicación u obtener algún tipo de recompensa. Este modelo es de los más populares y efectivos en Android [10].
- Suscripción: Se establece una tarifa que el usuario debe pagar con cierta regularidad para disfrutar de la aplicación y el servicio.
- Comercio electrónico: Utilizar la aplicación a modo de tienda para vender servicios y productos físicos.

Con las funciones actuales de la aplicación, se publicará en Google Play utilizando el modelo de pago por descarga, bajo un precio reducido. Este modelo de negocio no es el más adecuado en Android, porque a priori los usuarios de esta plataforma no suelen estar dispuestos a pagar por las aplicaciones [10]. En este caso se presupone que la persona que busca este tipo de herramienta sí tiene esta predisposición a realizar una inversión para disfrutar de este producto y las posibilidades que le puede ofrecer, ya que ha tenido que hacer un desembolso previo mucho mayor para disponer del sistema de iluminación Philips Hue.

Descartamos otros modelos como la publicidad porque no queremos interrumpir de ningún modo la experiencia de usuario, pudiendo llegar así a causar algún efecto no deseado. Además, uno de los objetivos del proyecto es que la aplicación sea lo más limpia y sencilla posible, por lo que excluimos la opción de incluir banners. Tampoco consideramos adecuado el modelo freemium en este caso, ya que el objetivo es que el usuario pueda aprovechar todas las funciones de la aplicación y brindarle la opción de que las pueda exprimir al máximo para mejorar la experiencia que ofrecen estos sistemas. Impedirle realizar algún tipo

de acción sobre su hogar una vez ha instalado y configurado la aplicación podría ser contraproducente.

El valor a pagar por la descarga de la aplicación será de 0.95€, realizándose una equivalencia exacta al valor de la moneda en otros países que no utilicen el euro. Google Play permite hacer esta conversión de forma automática, aunque también permite establecer un precio distinto en función del país de descarga. En USD el precio equivaldría a 0.99. Este precio establecido es notablemente inferior al precio de cualquier aplicación similar para Philips Hue publicada en Google Play, debido a que la aplicación aún no dispone de las funciones que se desean, y porque se pretende poder llegar al mayor público posible. Este precio incluye las tasas a pagar por cada venta, lo cual también depende del país desde el que se realice la compra. El dinero obtenido con las descargas se empleará para continuar con el desarrollo y la investigación.

5.2. Versiones

Durante el proceso de desarrollo se han llevado a cabo dos versiones de la aplicación, de las cuales hablaremos a continuación.

5.2.1. Primera aproximación

En esta primera versión se desarrolló el modo manual. El cual quedó funcionando de forma básica y la aplicación era capaz de controlar el sistema de iluminación y de agrupar los dispositivos en habitaciones. Aunque a pesar de ello, la aplicación tenía varios bugs que impedían el correcto funcionamiento a veces, y había situaciones concretas que no estaban controladas, por lo que la aplicación podía llegar, incluso, a crashear, o quedarse bloqueada en la splash screen.

Tampoco se habían tenido en cuenta situaciones en las que se añadiesen nuevos dispositivos de luz al sistema, o la sincronización de datos entre varios dispositivos (ya que no se iniciaba sesión). Además, a la interfaz aún le faltaba algo para tener el aspecto profesional deseado.

Estas son algunas capturas de pantalla de esta primera versión:

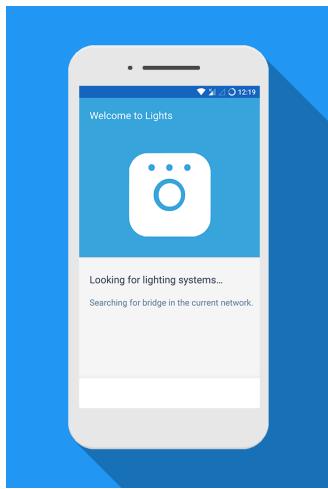


Figura 72: Primera aproximación - Captura 1

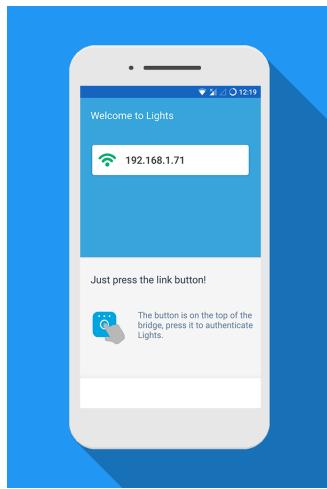


Figura 73: Primera aproximación - Captura 2

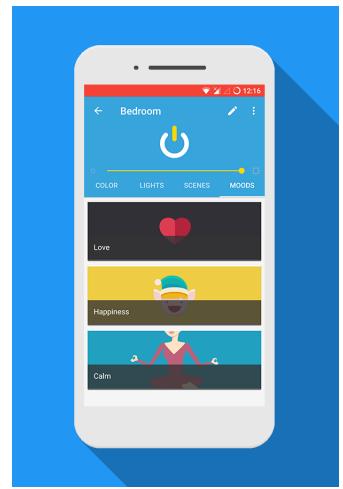


Figura 74: Primera aproximación - Captura 3

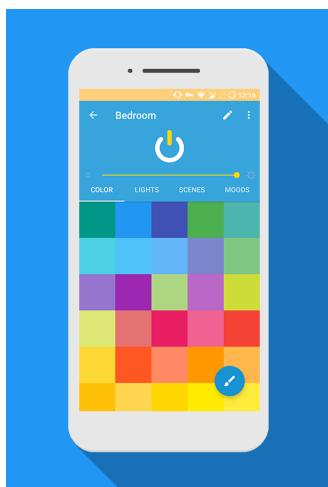


Figura 75: Primera aproximación - Captura 4

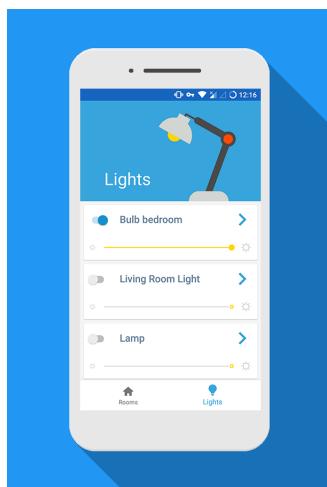


Figura 76: Primera aproximación - Captura 5

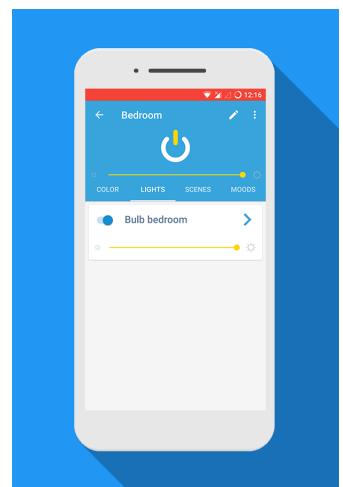


Figura 77: Primera aproximación - Captura 6

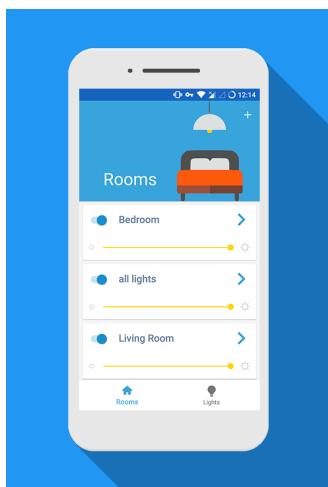


Figura 78: Primera aproximación - Captura 7

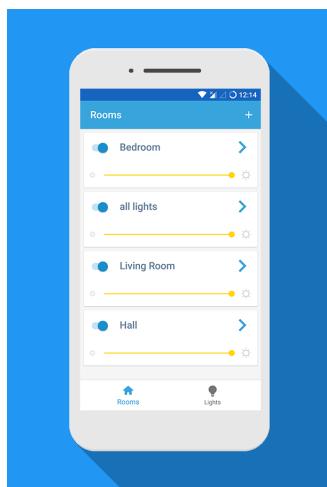


Figura 79: Primera aproximación - Captura 8

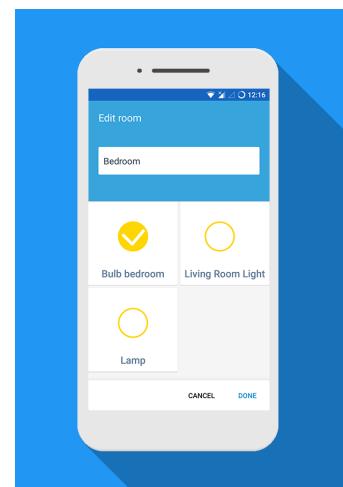


Figura 80: Primera aproximación - Captura 9

5.2.2. Segunda aproximación

Esta es la versión que se entrega junto a este documento, es la versión en la que se satisfacen todos los casos de uso, requisitos y aspectos comentados; además de solucionar todos los bugs y casos de uso que no se habían tenido en cuenta en la anterior versión.

Esta aplicación se encontrará disponible en Google Play en los próximos días, fue enviada el día 2 de septiembre pero, en la fecha de entrega de este documento, aún se encuentra en proceso de revisión. La aplicación se adjuntará con este documento en formato apk.



Figura 81: Aplicación enviada a Google Play esperando ser publicada.

Estas son algunas capturas de pantalla de esta versión:

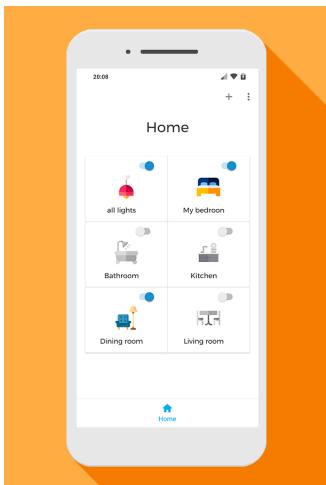


Figura 82: Segunda aproximación - Captura 1

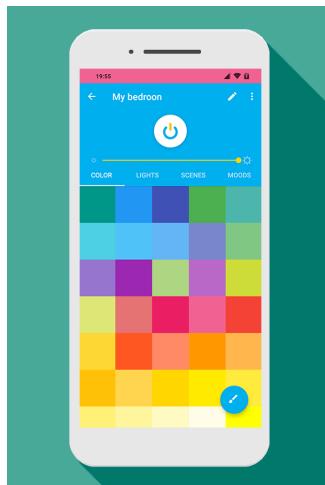


Figura 83: Segunda aproximación - Captura 2

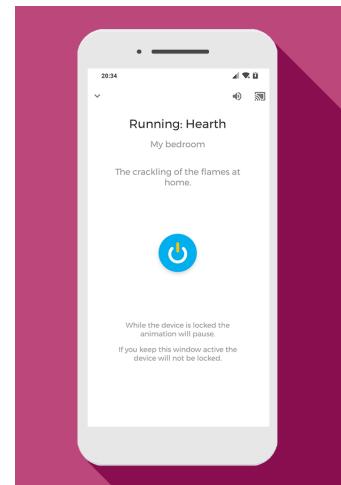


Figura 84: Segunda aproximación - Captura 3

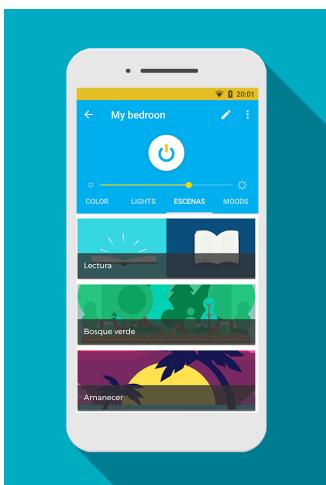


Figura 85: Segunda aproximación - Captura 4

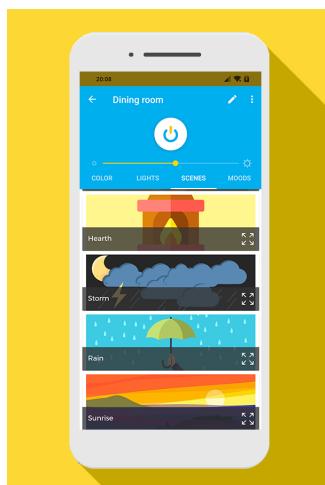


Figura 86: Segunda aproximación - Captura 5

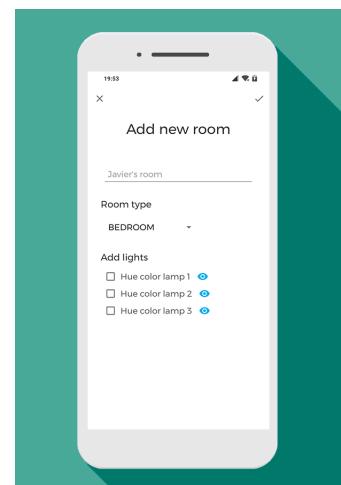


Figura 87: Segunda aproximación - Captura 6

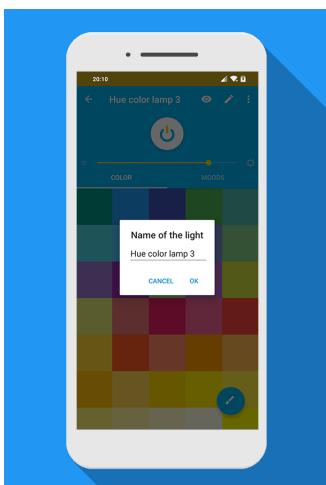


Figura 88: Segunda aproximación - Captura 7

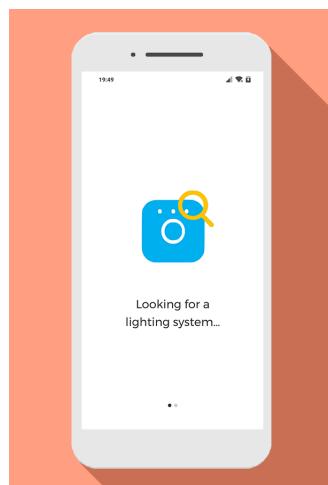


Figura 89: Segunda aproximación - Captura 8

5.2.3. Roadmap

Como ya se ha comentado en apartados anteriores, esta aplicación será continuada como proyecto de una beca de iniciación a la investigación. Se elaborará una nueva función dentro de la aplicación que buscará ser un modo inteligente del control de ambientaciones del hogar. Este modo tratará de averiguar el estado de ánimo de los usuarios mediante preguntas no invasivas, e irá elaborando un perfil emocional del usuario, con la intención de poder lanzar proyectos de luz que puedan provocar mejoras en el estado de ánimo detectado en cada momento.

Además de eso, otro objetivo es continuar con el desarrollo de otras funcionalidades, aumentando las posibilidades de esta aplicación, y buscando el modo de que la aplicación quede lo más completa posible. Además de solucionar todos los fallos que puedan llegar a encontrarse con esta nueva versión.

Por lo tanto, podríamos dividir el roadmap en dos partes.

Proyecto de iniciación a la investigación:

- Modo inteligente:

- Seguir diseñando y perfeccionando la idea. Por medio de prototipos e investigación al respecto, tratando de averiguar las posibilidades y funcionalidades que podemos desarrollar para exprimirlo lo máximo posible.
- Continuar con la elaboración y perfeccionamiento de la tabla de preguntas. Esta tabla se compone de una lista de emociones asociadas con acciones, propósito de cada acción, procedimiento y preguntas para confirmar las acciones. Además de una puntuación y un multiplicador de la puntuación en base al perfil emocional del usuario. Esta tabla buscar elaborar una serie de procedimientos que se puedan aplicar cuando se detecte una emoción concreta en el usuario. A continuación esa acción se validará por medio de una pregunta indirecta y opcional que permita ir elaborando un perfil del usuario. Buscando conocer si esa acción ha tenido un efecto positivo o no.
- Continuar con la elaboración y perfeccionamiento de la tabla de emociones. Esta tabla asocia a cada emoción con una serie de preguntas y respuestas posibles a esa pregunta, con la intención de poder averiguar el estado de ánimo de la persona, y poder actuar en consecuencia con las acciones. Además de recopilar los datos obtenidos. Nos basaremos en los modelos emocionales ya comentados en este documento, para tratar de averiguar la intensidad y tipo de emoción asignándole una puntuación de dos dimensiones a cada respuesta.
- Preparar el sistema de preguntas al usuario que se lleve a cabo de forma no invasiva y que puedan ser disparadas en momentos concretos.

- Lanzar proyectos de luz en momentos concretos en función de los datos obtenidos del usuario.
- Ampliar la aplicación para poder ofrecer otro tipo de respuestas, como consejos sobre una acción que podría realizar o algún contenido que podría visualizar si está, por ejemplo, aburrido.
- Perfeccionar los datos recopilados y el modo en que se obtienen, con el objetivo de que tengan valor para un posible estudio.
- Elaborar una vista del perfil de usuario en la que puedan mostrarse algunas métricas en base a los datos recopilados.

Mejoras de la funcionalidad:

■ Proyectos de luz:

- Seguir añadiendo nuevos proyectos de luz. Proyectos que puedan ser útiles o divertidos para los usuarios.
- Añadir más opciones posibles a los proyectos de luz, con el objetivo de poder ampliar las posibilidades que ofrece este sistema.
- Permitir al usuario desactivar proyectos de luz desde la aplicación.
- Elaborar un sistema que permita al usuario crear directamente desde la aplicación un proyecto de luz, generando un archivo JSON.
- Permitir a los usuarios añadir seleccionar la ubicación de los proyectos de luz en el sistema.

■ Nuevas funcionalidades:

- Posibilidad de crear un widget que permita apagar o encender la luz de una habitación sin tener que abrir la aplicación.
- Permitir configurar un sensor de movimiento con la aplicación. De modo que pueda configurarse para que la luz de una habitación concreta se encienda automáticamente al detectar la presencia de un usuario.
- Permitir la opción de programar efectos de luz para que se activen a una hora y días determinados. Por ejemplo: De lunes a viernes, a las 8 de la mañana, que se active el proyecto de luz ‘Amanecer’ en la habitación ‘Dormitorio’. Esto ya comenzó a desarrollarse, pero no está disponible aún por falta de tiempo.

■ Mejoras al sistema de iluminación:

- Permitir al usuario poder guardar el estado de la habitación actual como una escena. De este modo el usuario podrá poner los dispositivos de luz en un estado que desee, y a continuación salvar ese estado de forma sencilla para poder recuperarlo en el futuro.

- Permitir al usuario borrar escenas de las habitaciones.
- Permitir añadir sistemas de iluminación de otros fabricantes. Este punto me parece muy interesante, ya que así será posible llegar a un público mucho más amplio, sobre todo, teniendo en cuenta que cada vez hay más opciones en el mercado y algunas de ellas muy accesibles (debido al precio). En la actualidad ya he adquirido dispositivos de iluminación de Xiaomi, con el objetivo de poder probar a extender el sistema.

6. Manual de usuario

6.1. Configuración del sistema Philips Hue

Philips Hue se compone de dos piezas fundamentales para funcionar:

- Gateway: El gateway se conecta con los dispositivos de luz y es capaz de mandarles órdenes mediante el protocolo Zigbee [23]. Además, cuenta con una memoria interna que almacena información útil sobre el estado del sistema y la configuración.
- Dispositivos de luz: Dispositivo capaz de ser controlado por el gateway.

Para utilizar este sistema, es necesario tener una red Wi-Fi que nos permita comunicarnos con el gateway.



Figura 90: Gateway Philips Hue



Figura 91: Dispositivo de iluminación

Los pasos al seguir para configurar este sistema son los siguientes:

1.- Mediante un cable de ethernet se conecta el gateway al router, el cual debe encontrarse conectado y configurado creando una red Wi-Fi accesible. También es necesario conectar el dispositivo a la red eléctrica.



Figura 92: Configuración de Philips Hue: Conectar gateway.

2.- Los dispositivos de iluminación deben ser conectadas a la red eléctrica en función del tipo de dispositivo. El interruptor ha de dejarse en encendido.



Figura 93: Configuración de Philips Hue: Conectar dispositivos de luz.

3.- Utilizar la aplicación para registrar y configurar los dispositivos de luz (se detalla en los siguientes apartados).

6.2. Configuración del altavoz

La configuración del altavoz varía en función del dispositivo que decidamos usar, como ya se ha mencionado, sirve cualquier dispositivo compatible con la tecnología Google Cast (como Google Home o Google Chromecast). Por defecto, se utilizará el altavoz del dispositivo móvil.

Para poder utilizar un altavoz es necesario que, además de contar con lo anteriormente mencionado, esté conectado y configurado en la misma red Wi-Fi que el dispositivo Android que ejecuta la aplicación.

6.3. Configuración de la aplicación

Si la aplicación no ha sido configurada aún, se lanzará el proceso de configuración. Este proceso es el siguiente:

- 1.- Iniciar sesión la cuenta de Google.

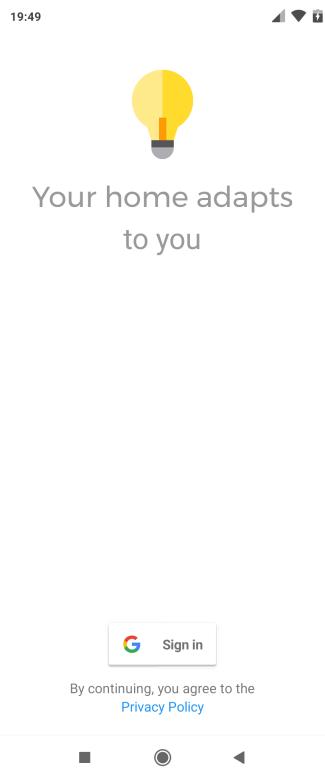


Figura 94: Configuración de Lights: Iniciar sesión

Para ello hay que pulsar el botón de iniciar sesión, y a continuación seleccionar la cuenta de Google con la que se desea acceder.

- 2.- Conectar sistema de iluminación.

La aplicación comenzará a buscar los dispositivos de luz disponibles en el sistema. Una vez encontrados, se solicitará al usuario que presione el botón del gateway para conectarlo.

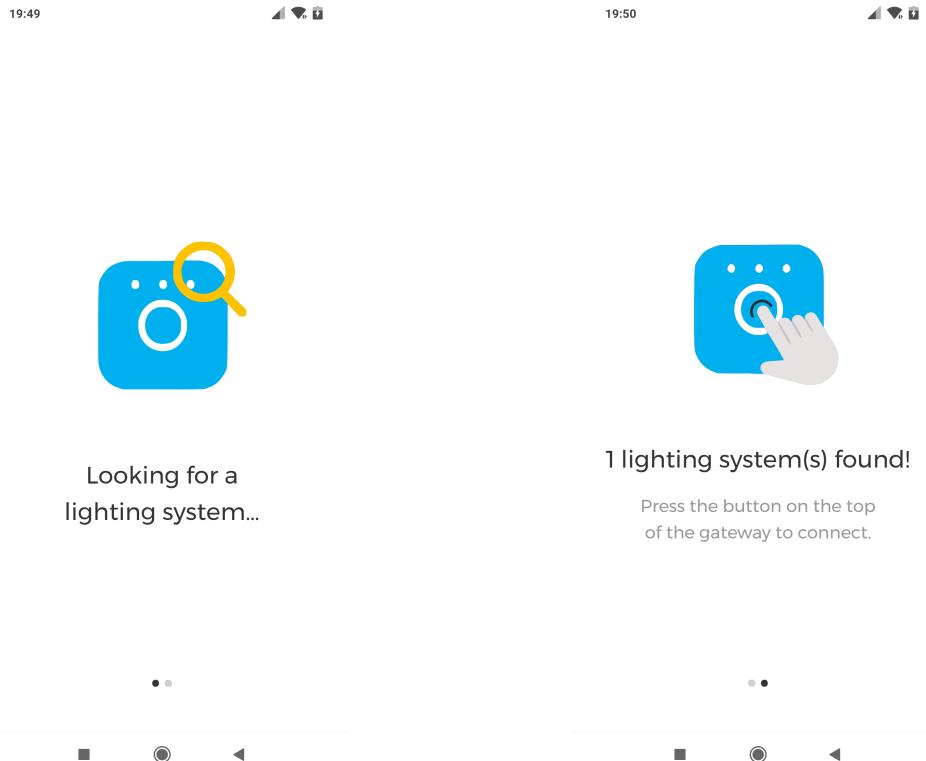


Figura 95: Configuración de Lights: Buscando sistemas de iluminación

Figura 96: Configuración de Lights: Conectar con el sistema de iluminación

3.- Buscar y registrar dispositivos de luz

En caso de no tener registrado el gateway ningún dispositivo de iluminación, la aplicación mostrará una animación de espera mientras se encuentra y registra al menos algún dispositivo. Cuando esto ocurra, la aplicación ya podrá utilizarse, aunque puede ser que encuentren más dispositivos de luz y se añadan automáticamente mientras el usuario ya está utilizando la aplicación.

Para que los dispositivos de luz puedan encontrarse, deben de estar conectados en la misma red que el gateway y la aplicación, y estar encendidos.

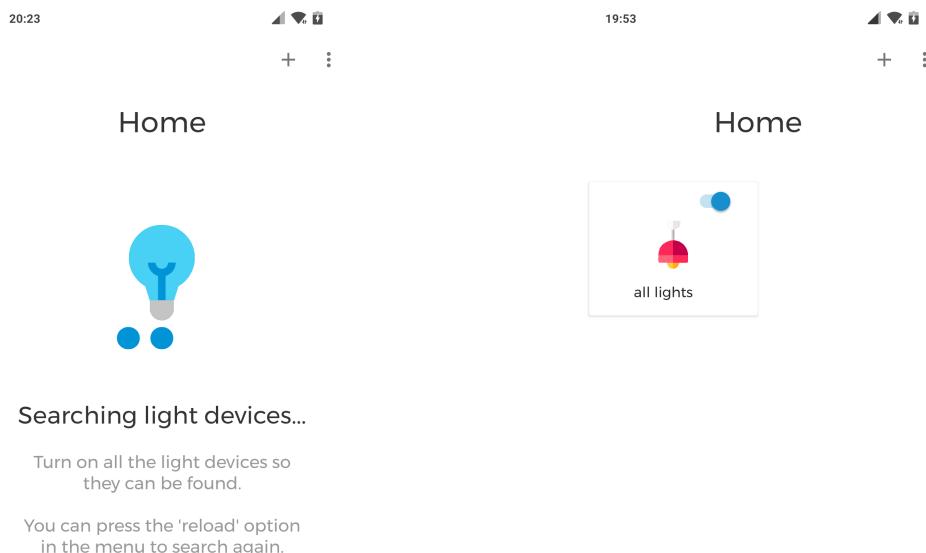


Figura 97: Configuración de Lights: Buscando dispositivos de iluminación

Figura 98: Lights: Aplicación recién configurada

Por defecto, el gateway devuelve una habitación en la que se engloban todos los dispositivos de luz disponibles en la casa.

A continuación, es posible crear nuevas habitaciones. Para crear nuevas habitaciones hay que pulsar sobre la opción añadir ('+') situada en la esquina superior derecha del menú.

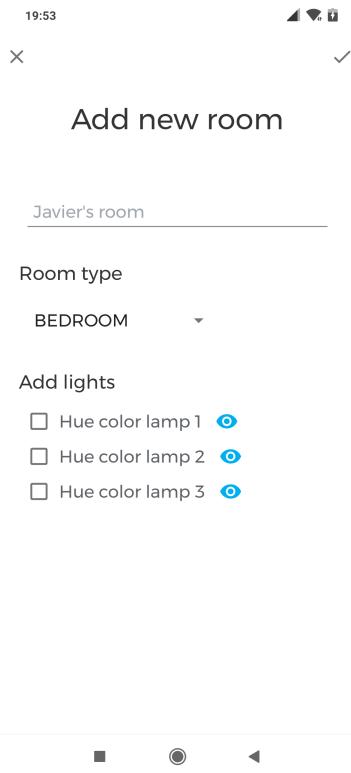


Figura 99: Lights: Crear nueva habitación

En esta actividad es necesario introducir el nombre de la habitación (por defecto aparece con el nombre de la cuenta de Google), el tipo de habitación, y los dispositivos que se deseen añadir. La lista de dispositivos se conforma por todos los dispositivos detectados y registrados en el sistema.

El icono con forma de ojo que aparece a la derecha de cada dispositivo de luz, sirve para hacerlo parpadear y así poder ubicarlo en el espacio físico.

Una vez configurada la habitación, para guardarla y añadirla a la lista de habitaciones disponibles del sistema, hay que pulsar el borón de guardar que aparece en la esquina superior derecha. En caso de querer volver atrás sin guardar, el botón de la flecha hacia la izquierda que aparece en la esquina superior izquierda.

Las habitación que se vayan creando, irán apareciendo en la lista de habitaciones de la casa:

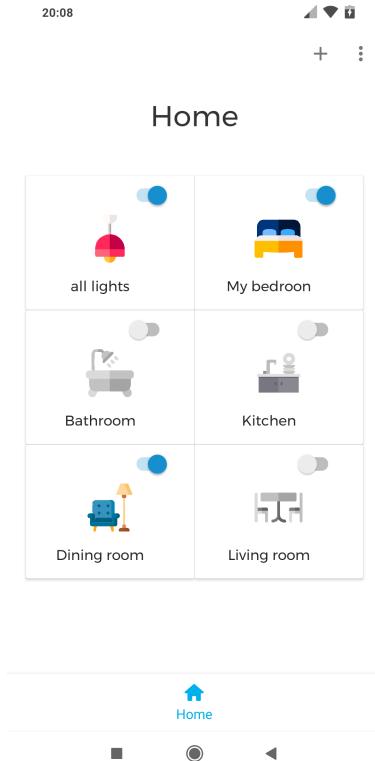


Figura 100: Lights: Lista de habitaciones disponibles

En el menú contextual que se despliega al pulsar el ícono de los tres puntos que se ubica en la esquina superior derecha, se encuentran las siguientes opciones:

- Buscar un nuevo gateway: Opción para configurar de nuevo el sistema de iluminación. Esta opción nos llevaría al segundo paso de la configuración inicial.
- Refrescar: Esta opción sirve para forzar que se refresque la lista de habitaciones (sincronizando las habitaciones con la base de datos de la nube y del gateway) y buscando nuevos dispositivos de luz disponibles en la red y registrándolos en caso de ser necesario. Esta opción ya la realiza la aplicación automáticamente, pero es posible ser forzada por el usuario.

6.3.1. Actividades de error

Hay dos posibles actividades de estados de error a las que podemos llegar:

- Se ha detectado la pérdida de conexión con el gateway: Se soluciona buscando de nuevo el gateway en la red.
- Se ha perdido la conexión Wi-Fi: Se soluciona conectando el dispositivo a la red Wi-Fi. La aplicación volverá automáticamente al punto anterior al detectar que se ha vuelto a conectar a la red.

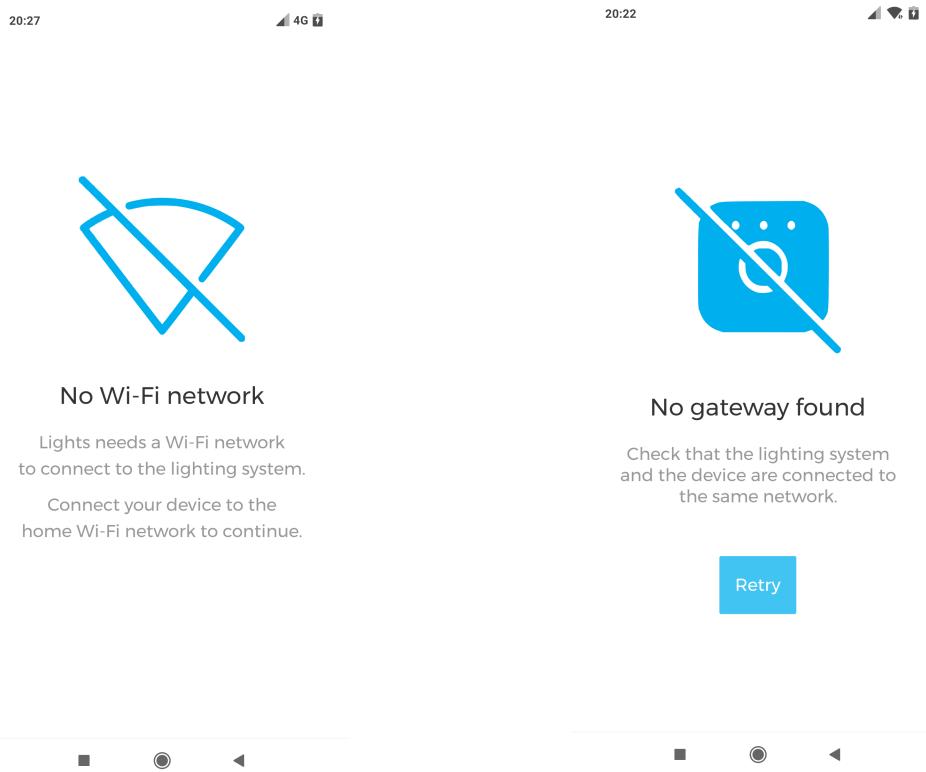


Figura 101: Lights: Pérdida de la conexión Wi-Fi

Figura 102: Lights: Pérdida de la conexión con el gateway

6.3.2. Control de la iluminación de una habitación

En la lista de habitaciones es posible encender o apagar directamente una habitación pulsando en el botón que incluye cada una de las habitaciones.

Para acceder a la vista con todas las opciones disponibles de una habitación, es necesario pulsar sobre la habitación deseada de la lista.



Figura 103: Lights: Vista de una habitación encendida. Dispositivo: Pocophone F1.

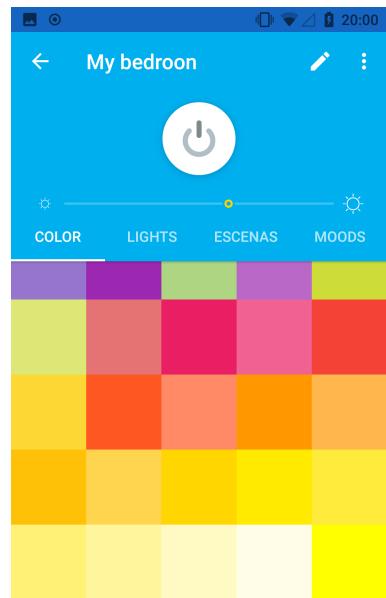


Figura 104: Lights: Vista de la habitación apagada. Dispositivo: Xiaomi MI5.

El botón superior que se sitúa alineado en el eje vertical, permite apagar o encender la habitación completa al pulsarlo. Cuando el ícono y la barra superior tienen color, significa que la habitación está encendida. En las figuras anteriores se puede ver como en el caso de la izquierda la habitación está encendida, y en el de la derecha apagada.

La barra que aparece justo debajo del botón de encendido/apagado, indica el nivel de brillo de la habitación. El brillo es el valor medio obtenido de todos los dispositivos de luz que se encuentren activos en la habitación. Al desplazar el indicador a la izquierda bajaremos el brillo, y al desplazarlo hacia la derecha aumentaremos el valor del brillo, cambiando de este modo el brillo de todos los dispositivos de la habitación al mismo tiempo. En la figura anterior se puede ver como en el caso de la izquierda el brillo de la habitación está al máximo, y en el de la derecha se encuentra con un valor ligeramente superior a la mitad.

En el menú de opciones que se encuentra en la esquina superior derecha, se encuentran las siguientes opciones:

- Recargar escenas: Vuelve a leer los proyectos de luz del sistema de archivos y los incorpora al sistema. Incluyendo los cambios producidos en los proyectos, y los nuevos proyectos que se hayan guardado.
- Borrar habitación: Elimina la habitación. Requiere confirmación.

Otra opción disponible en el menú, es la que se denota con el icono de un lápiz. Esta opción desplegará la actividad que nos permite modificar los mismos parámetros que ya habíamos configurado previamente al crear la habitación.

La vista de la habitación cuenta con una serie de pestañas que nos permitirán ir desplegando las opciones disponibles. Estas pestañas son: Colores, Luces, Escenas y Moods.

La vista de la pestaña ‘Colores’ es la que podemos ver en las figuras anteriores. En esta pestaña vemos una vista de la paleta de colores disponible para esta habitación. La paleta de la habitación es personalizable, pudiéndose añadir nuevos colores o borrando los ya existentes.

Para añadir nuevos colores a la paleta, es necesario pulsar el botón flotante que podemos ver en la figura izquierda anterior. En la figura de la derecha no podemos ver esta opción porque el botón desaparece al desplazar abajo del todo para no tapar los últimos colores disponibles en la paleta.

Para borrar colores, hay que mantener presionado uno de los colores para entrar al modo edición. Una vez en el modo edición es posible seleccionar tantos colores como se desee y borrarlos pulsando en el ícono de la papelera. Si se vuelve atrás, o se cambia de pestaña, se cancelará la acción.

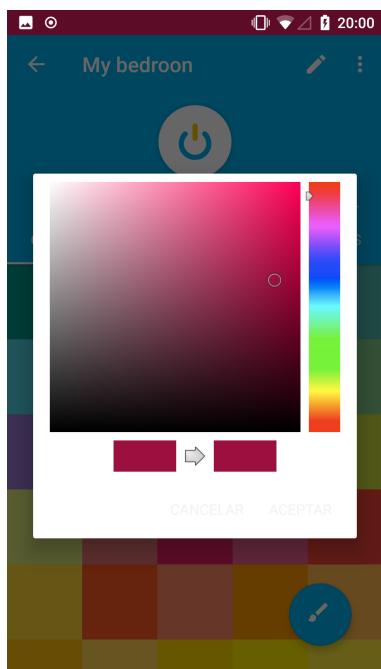


Figura 105: Lights: Añadir colores a la paleta de colores de una habitación.

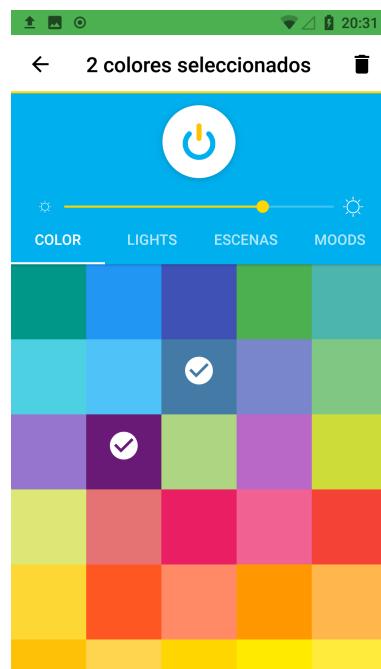


Figura 106: Lights: Eliminar colores de la paleta de colores de una habitación.

En el caso de esta pestaña, contamos con una opción extra en el menú de opciones. Esta opción es la de restaurar los colores por defecto.

La siguiente pestaña es la de luces. En ella se listan todos los dispositivos de luz disponibles en la habitación:

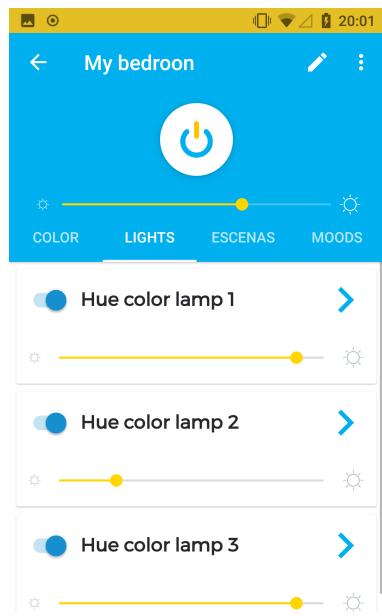


Figura 107: Lights: Pestaña de luces de la habitación

Para cada dispositivo de luz, es posible modificar el brillo, encenderlo y apagarlo, o entrar en la vista de visualización del dispositivo si se pulsa. La vista de visualización del dispositivo de luz es idéntica a la de la habitación, pero cuenta con menos pestañas.

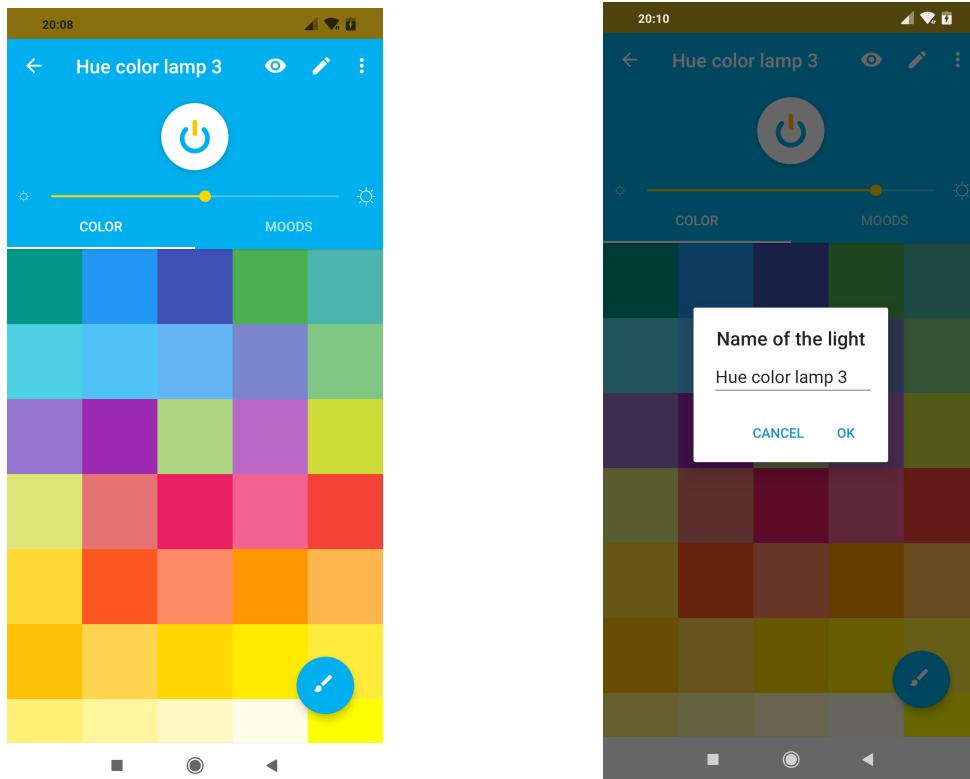


Figura 108: Lights: Vista de un dispositivo de luz.

Figura 109: Lights: Cambiar nombre de un dispositivo de luz.

La opción de editar, en este caso nos permite modificar el nombre del dispositivo de luz.

La siguiente pestaña es la de escenas. Esta pestaña muestra la lista de todas las escenas y proyectos de luz disponibles en el sistema.

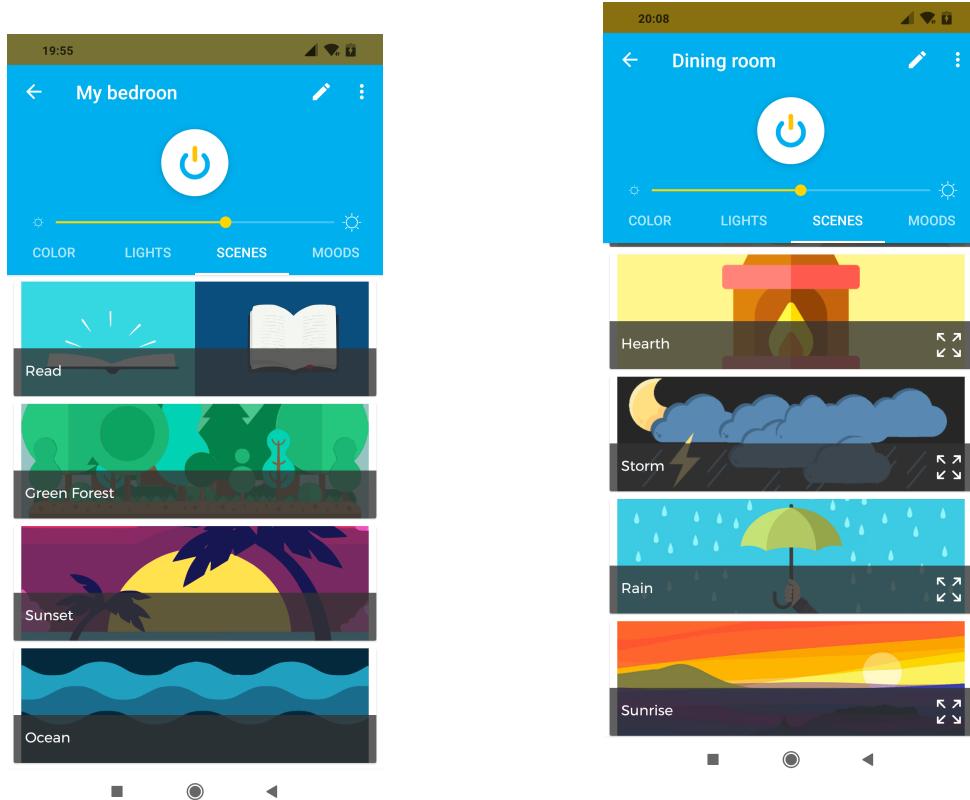


Figura 110: Lights: Vista de escenas.

Figura 111: Lights: Vista de proyectos de luz.

Una escena es un estado guardado que afecta a todos los dispositivos de luz de la habitación de forma estática. Este estado incluye brillo y color. Un proyecto de luz también afecta del mismo modo al estado de los dispositivos, pero por el contrario, es dinámico e incluye cambios en función del tiempo. Estos últimos se pueden visualizar en pantalla completa, por lo que visualmente se diferencian en la lista por un ícono que indica que se abren expandiéndose en la pantalla. Tanto los proyectos de luz como las escenas pretenden simular ambientes de acciones o paisajes por medio de la iluminación de la habitación.

La pestaña restante es la de moods, lo cual es muy parecido a las escenas, pero con un objetivo diverso. Los moods pretenden adecuar el ambiente de la habitación a un estado de ánimo concreto.

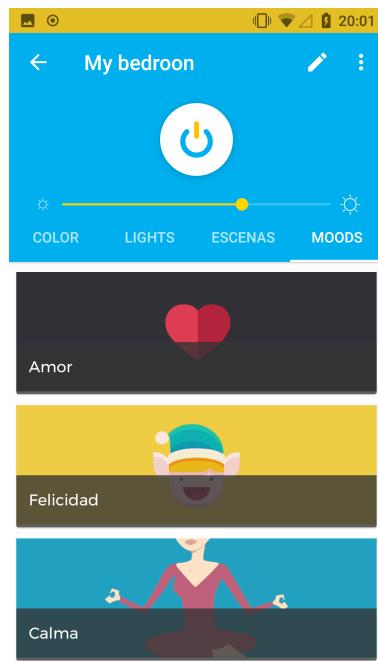


Figura 112: Lights: Pestaña de moods de la habitación

6.3.3. Proyectos de luz

Como ya se ha comentado en las secciones anteriores, los proyectos de luz son escenas dinámicas que pretenden simular un ambiente por medio de la luz y el sonido. Los proyectos de luz se pueden visualizar en pantalla completa:

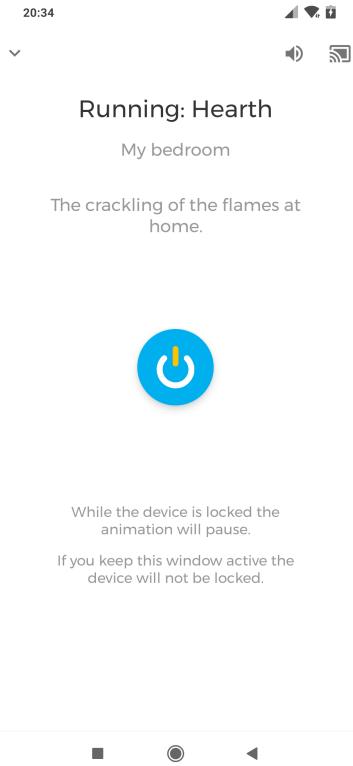


Figura 113: Lights: Vista de un proyecto de luz

En la parte superior podemos ver el nombre del proyecto de luz que se está ejecutando, en el caso de la figura anterior se trata de 'Hearth'. Justo debajo también se muestra la descripción del mismo. Esta vista nos ofrece las siguientes opciones:

- Detener el proyecto de luz: Es el botón que se encuentra justo en el centro del a pantalla.
- Minimizar el proyecto de luz: Es el botón con la flecha hacia abajo que se encuentra a la izquierda del menú superior. El proyecto de luz continuará ejecutándose en segundo plano, y se podrá volver a él pulsando sobre el proyecto de luz en la lista de escenas, o sobre la notificación que aparecerá en la barra de notificaciones indicando que este proyecto está activo.
- Silenciar proyecto de luz: En caso de que el proyecto de luz disponga de sonido, este comenzará a ejecutarse en los altavoces disponibles en el dispositivo Android. El sonido siempre podrá silenciarse o volver a sonar, pulsando sobre la opción del altavoz situada en la parte superior derecha del menú.
- Cast: El sonido podrá enviarse en streaming a cualquier dispositivo de la red actual que disponga de la tecnología Google Cast.

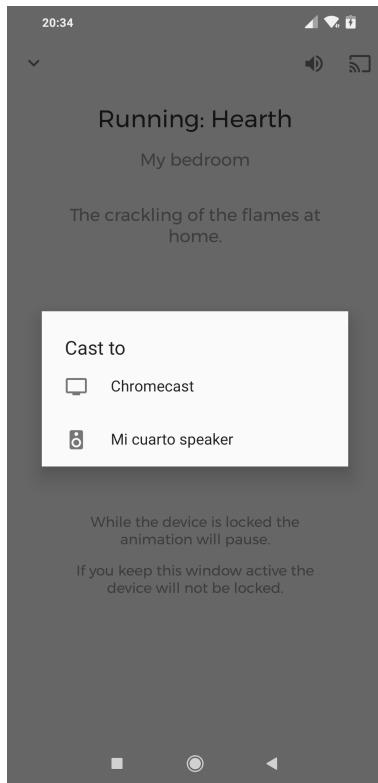


Figura 114: Lights: Lista de dispositivos disponibles para reproducir el sonido en streaming.

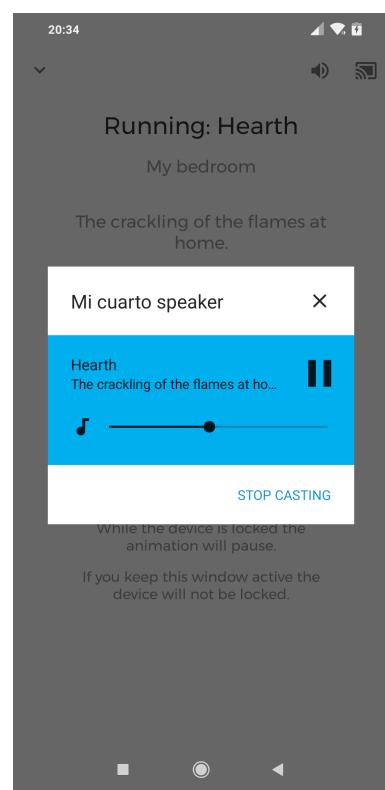


Figura 115: Lights: El sonido del proyecto de luz se está reproduciendo en un altavoz de la red.

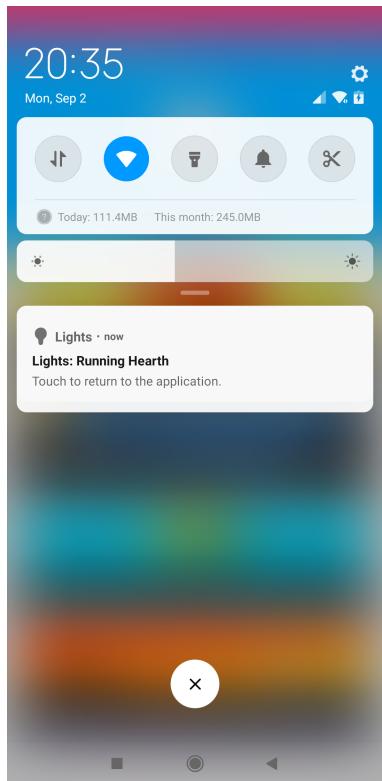


Figura 116: Lights: Notificación de proyecto de luz en ejecución.

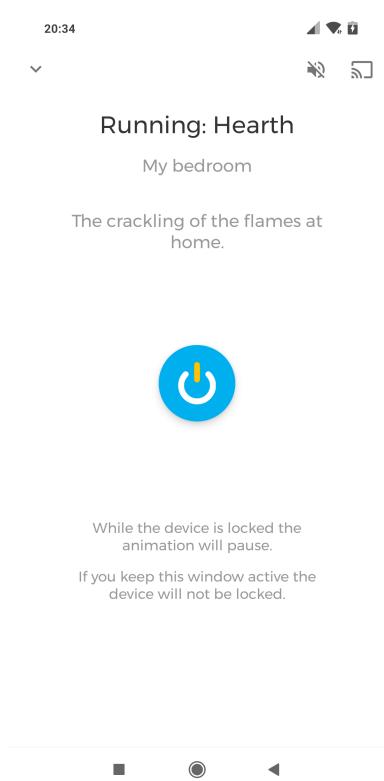


Figura 117: Lights: Proyecto de luz silenciado.

7. Conclusiones

Finalmente se han logrado satisfacer casi todos los objetivos y requisitos marcados para esta primera parte del proyecto, obteniendo la aplicación que nos permite controlar el sistema de iluminación y reproducir ambientaciones. De esta manera, se da por cerrada esta primera parte.

El proceso de desarrollo software es muy complejo, y aunque a priori el proyecto parezca más sencillo, siempre surgen imprevistos y problemas que no se habían tenido en cuenta, por lo que es fácil que se extienda en el tiempo más de lo previsto. Por ello es necesario una buena fase de especificación y diseño, a la que se le den todas las vueltas necesarias hasta tener claro qué vamos a desarrollar exactamente y cómo. De lo contrario se pierde mucho tiempo teniendo que volver atrás a rehacer cosas, o detiendo el desarrollo por no tener claras algunas partes. También se ha demostrado la gran utilidad de llevar a cabo prototipos de utilidades concretas del producto, ya que así podemos verificar nuestras ideas antes de embarcarnos a desarrollar algo de lo que no estamos seguros del todo.

También ha sido importante llevar a cabo una buena gestión del proyecto y una buena planificación. Esta es la mejor forma de ser eficientes y aprovechar el tiempo, además de poder prever así el tiempo que nos va a llevar el desarrollo, y cómo de bien vamos respecto a lo previsto. La experiencia nos proporcionará la

capacidad de prever mejor el tiempo que requieren las cosas y lo que nosotros mismos necesitamos para llevar a cabo cada tarea prevista.

Al principio, tanto la planificación como especificación y diseño pueden parecer un proceso tedioso cuando aún no tenemos la experiencia suficiente. Es fácil querer lanzarnos con la implementación lo antes posible para intentar ahorrar tiempo. Pero al final el resultado no suele ser el esperado.

7.1. Análisis de los objetivos planteados

Objetivo específico	Nivel de logro	Observaciones
Estudio de los factores que caracterizan el estado de ánimo de una persona.	50%	Se ha comenzado a realizar este estudio pero aún no ha habido tiempo suficiente de completarlo. Esto se espera lograr en la siguiente fase del desarrollo
Analizar de qué modo se puede medir el estado de ánimo de una persona mediante un sistema de cómputo.	50%	Se han estudiado modelos emocionales que nos permiten establecer un sistema de cómputo, pero aún no se ha llevado a cabo ni se han hecho pruebas sistemáticas con usuarios.
Explorar los diferentes sistemas de iluminación del mercado, y realizar un análisis comparativo.	70%	Se han buscado y analizado sistemas de iluminación del mercado, pero solo se ha puesto en práctica el uso de Philips Hue.
Revisar las aplicaciones de control de iluminación ya publicadas para dispositivos Android, para averiguar qué características ofrecen a los usuarios y cómo las llevan a cabo.	100%	Se ha probado y examinado una amplia gama de aplicaciones de este tipo, lo cual nos ha permitido tener una visión más amplia de las posibilidades y del mercado.
Averiguar y utilizar las herramientas y servicios de Android que mejor se adapten a nuestro propósito.	90%	Se ha tenido muy en cuenta la documentación de Android y los foros de desarrollo. Realizándose una investigación sobre qué era mejor utilizar en cada caso y utilizando servicios que, personalmente, nunca había utilizado
Realizar un trabajo de extracción y análisis de requisitos, y de elaboración de casos de uso que nos permita especificar todo lo posible las funciones que tendrá esta versión de la aplicación.	100%	Se ha prestado bastante atención a la extracción de requisitos y a la elaboración de los casos de uso, realizando varias iteraciones sobre ellos y función de los resultados obtenidos con los prototipos.

Figura 118: Tabla de análisis de los objetivos planteados I

Objetivo específico	Nivel de logro	Observaciones
Intentar averiguar cómo proporcionar una experiencia de usuario satisfactoria mediante una navegabilidad y diseño de interfaz adecuado, buscando y analizando otras referencias y siguiendo las guías de estilo.	100%	Se ha hecho bastante hincapié en este aspecto y se ha tenido totalmente en cuenta a la hora de diseñar y llevar a cabo la aplicación
Llevar a cabo una correcta gestión y organización del proyecto.	50%	Debido a que han ido surgiendo inconvenientes durante el desarrollo (compaginarlo con otro trabajo) ha habido épocas en las que no se ha cumplido correctamente con la planificación ni la metodología. Pero me ha servido para tomar mayor conciencia de la importancia de esto.
Aplicar buenas prácticas de ingeniería en el proceso de desarrollo de la aplicación.	70%	Me hubiera gustado poder acompañar el proyecto de una variedad amplia de diagramas y detalles sobre el procedimiento y el diseño, pero no he tenido tiempo material para mantener actualizados algunos diagramas con los cambios que han ido surgiendo durante el desarrollo. Aunque sí se han llegado a realizar en las fases más tempranas del proyecto. Se han aplicado procedimientos como la planificación del trabajo, el control de versiones, el análisis de las posibilidades de dispositivos, el prototipado, el uso de patrones, etc.
Desarrollar un prototipo de la aplicación, y probarlo con una muestra de usuarios preseleccionada.	50%	Se ha desarrollado el prototipo, pero no ha habido tiempo material de realizar las pruebas sistemáticas.

Figura 119: Tabla de análisis de los objetivos planteados II

7.2. Trabajos futuros

Esta aplicación es parte de un proyecto más grande, y como ya se ha comentado, se continuará su desarrollo en un proyecto de iniciación a la investigación

que tiene como objetivo tratar de averiguar de un modo no intrusivo el estado de ánimo de los usuarios y actuar en consecuencia. Aprendiendo de los inputs obtenidos para poder automatizar de un modo inteligente el ambiente que se recrea en el hogar, además de recopilar posibles datos de interés científico. Además, también se desean llevar a cabo otras mejoras como las que se detalla en la sección Roadmap.

7.3. Experiencia personal

Uno de mis objetivos personales con este proyecto, era seguir aprendiendo y mejorando mis conocimientos de Android y de ingeniería de software. Aprendiendo de la documentación de Android cuáles son los mejores caminos para resolver los problemas que surjan durante la elaboración de la aplicación, e intentando que la aplicación tenga un aspecto profesional y limpio (lo cual es algo que me cuesta conseguir). Construyendo por primera vez una aplicación que hiciera uso de los dispositivos del hogar para lograr un objetivo; y de distintas herramientas como las usadas: Firebase, Cast Application Framework, etc. Lo cual creo que se ha podido llevar a cabo, y me ha servido enormemente para ampliar mis conocimientos.

Referencias

- [1] Alarmmanager <https://developer.android.com/reference/android/app/AlarmManager>.
- [2] Ambilwarna <https://github.com/yukuku/ambilwarna>.
- [3] Androidx <https://developer.android.com/jetpack/androidx>.
- [4] App bars: top <https://material.io/components/app-bars-top/>.
- [5] Avantek bombillas de wi-fi <https://www.amazon.es/AVANTEK-Inal%C3%A1mbrico-Inteligente-Inal%C3%A1mbrica-Energ%C3%A9ticaamente/dp/B017QYHG7C>.
- [6] Bottom navigation <https://material.io/components/bottom-navigation/#>.
- [7] Buttons: floating action button <https://material.io/components/buttons-floating-action-button/>.
- [8] Cast application framework (caf) <https://developers.google.com/cast/docs/developers>.
- [9] Chaos report 2015 https://standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.
- [10] El negocio para los desarrolladores android no está en las apps de pago sino en las compras in-app <https://www.xatakamovil.com/aplicaciones/el-negocio-para-los-desarrolladores-android-no-esta-en-las-apps-de-pago-sino-en-las-compras-in-app>.
- [11] Firebase <https://firebase.google.com/docs/database?hl=es-419>.
- [12] Ge link <https://www.wink.com/products/ge-link-connected-led-bulbs/>.
- [13] Get started philips hue <https://developers.meethue.com/develop/get-started-2/>.
- [14] Google fit <https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=es>.
- [15] Google home <https://play.google.com/store/apps/details?id=com.google.android.apps.chromecast.app&hl=es>.
- [16] Google trips <https://get.google.com/trips/>.
- [17] Hoja informativa sobre los ritmos circadianos <https://www.nigms.nih.gov/education/pages/los-ritmos-circadianos.aspx>.
- [18] How admob works <https://support.google.com/admob/answer/7356092?hl=en>.
- [19] Lohas led <http://www.lohas-led.com/>.

- [20] Material design <https://material.io/>.
- [21] Modern background execution in android <https://android-developers.googleblog.com/2018/10/modern-background-execution-in-android.html>.
- [22] Obtén más ingresos con las opciones de monetización adecuadas <https://developer.android.com/distribute/best-practices/earn/monetization-options?hl=ES>.
- [23] Philips hue: Supported lights and devices (hue compatible / hue kompatibel) <https://iconnecthue.com/supported-devices/>.
- [24] Philips hue tools and sdks <https://developers.meethue.com/develop/tools-and-sdks/>.
- [25] Philips hue <http://www2.meethue.com/es-es>.
- [26] Putting personas to work in ux design: What they are and why they're important <https://theblog.adobe.com/putting-personas-to-work-in-ux-design-what-they-are-and-why-theyre-important/>.
- [27] Sectograph <https://play.google.com/store/apps/details?id=prox.lab.calclock&hl=es>.
- [28] Services overview <https://developer.android.com/guide/components/services>.
- [29] Tabs <https://material.io/components/tabs/#usage>.
- [30] Tabs <https://plugins.jetbrains.com/plugin/4509-statistic/versions>.
- [31] Tcp connected <https://www.cnet.com/es/analisis/connected-by-tcp-led-lighting-control-system/resena/>.
- [32] Trådfri <http://www.ikea.com/es/es/catalog/products/70338932>.
- [33] Wink hub <https://www.wink.com/products/wink-hub/>.
- [34] Xiaomi yeelight <https://www.amazon.es/Ollivan-Bombilla-Inteligent-Ajustable-Multi-dp/B01HXQF3PA>.
- [35] ¿qué son los ritmos circadianos? <https://www1.nichd.nih.gov/espanol/salud/temas/sleep/informacion/Pages/circadianos.aspx>.
- [36] Feeltrace: An instrument for recording perceived emotion in real time https://www.isca-speech.org/archive_open/speech_emotion/spem_019.html, Online; visto el 3 de agosto de 2019.
- [37] Erik Cambria. *Affective Computing and Sentiment Analysis*. IEEE Intelligent Systems Vol 31 (2), 102 – 107, 2016.

- [38] E Joosten, Giel Lankveld, and Pieter Spronck. Colors and emotions in video games. *11th International Conference on Intelligent Games and Simulation, GAME-ON 2010*, 01 2010.
- [39] Ragavendra Lingamaneni Jürgen Scheible, Lisa Schuhmacher, Anne Pagenkopf. *Emotion Enhancement through Ubiquitous Media Technology in a Smart Kitchen Environment*. MUM 2018 Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia, pp 317-325, 2018.
- [40] Vega K. Luo E. *Scentry: a calming multisensory environment by mixing virtual reality, sound, and scent*. MobileHCI 39;18 Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, pp 158-165. ACM, 2018.
- [41] Michael McTear, Zoraida Callejas, and David Griol. *Emotion, Affect, and Personality*. 05 2016.
- [42] Jesus Minguillon, Miguel Angel Lopez-Gordo, Diego A. Renedo-Criado, Maria Jose Sanchez-Carrion, and Francisco Pelayo. Blue lighting accelerates post-stress relaxation: Results of a preliminary study. *PLOS ONE*, 12(10):1–16, 10 2017.
- [43] JM Gutiérrez-Guerrero JL Muros-Cobos S. Rodríguez-Valenzuela, JA Holgado-Terriza. *Distributed Service-Based Approach for Sensor Data Fusion in IoT environments*. *Sensors*. 14: 19200-19228. MDPI, 2014.

Índice de figuras

1.	Circumplex Model of Emotion (Russel 1980) [38]	9
2.	Comparativa entre sistemas de iluminación del mercado	12
3.	Persona 1 - Antonio Carbonero Brunel	14
4.	Persona 2 - Iolanda Venuti	15
5.	Persona 3 - Izan Brown	16
6.	Persona 4 - Lucía Esparta Serrano	17
7.	Persona 5 - Marco Ariza Maldonado	18
8.	Persona 6 - Marina Páramos Díaz	19
9.	Tabla comparativa de algunas aplicaciones para controlar el sistema de iluminación Philips Hue.	21
10.	A0 - Usuario	26
11.	Diagrama de casos de uso: Paquete de configuración	28
12.	Diagrama de casos de uso: Paquete de iluminación y sonido	29
13.	Diagrama de paquetes funcional	29
14.	CU0 - Conectar al gateway del sistema de iluminación	30
15.	CU1 - Crear habitación	31
16.	CU2 - Modificar habitación	32
17.	CU3 - Eliminar habitación	33
18.	CU4 - Visualizar habitación	34
19.	CU5 - Escanear dispositivos de luz	35
20.	CU6 - Apagar luces de la habitación	36
21.	CU7 - Escanear luces de la habitación	37
22.	CU8 - Cambiar intensidad el brillo de las luces de la habitación	38
23.	CU9 - Silenciar proyecto de luz	39
24.	CU10 - Cambiar color de las luces de la habitación	40
25.	CU11 - Apagar un dispositivo de luz	41
26.	CU12 - Encender un dispositivo de luz	42
27.	CU13 - Cambiar intensidad del brillo de un dispositivo de luz	43
28.	CU14 - Iniciar proyecto de luz	44

29.	CU15 - Detener proyecto de luz	45
30.	CU16 - Reproducir sonido	46
31.	CU17 - Arrancar el sistema	47
32.	CU18 - Escanear habitaciones	48
33.	CU19 - Hacer que un dispositivo de luz parpadee	49
34.	CU20 - Aplicar una escena a una habitación	50
35.	CU21 - Cargar proyectos de luz	51
36.	CU22 - Añadir un nuevo color a la paleta	52
37.	CU23 - Eliminar colores de la paleta	53
38.	Referencia interfaz gráfica - Google Home [15]	55
39.	Referencia interfaz gráfica - Google Home [15]	55
40.	Referencia interfaz gráfica - Sectograph [27]	56
41.	Referencia interfaz gráfica - Google Fit [14]	56
42.	Referencia interfaz gráfica - Google Trips [16]	57
43.	Referencia interfaz gráfica - Google Trips [16]	57
44.	Wireframe 1 - Configuración	58
45.	Wireframe 2 - Actividad principal	59
46.	Wireframe 3 - Vistas de habitación, de dispositivo de luz, y de proyecto de luz.	60
47.	Logo de la aplicación.	61
48.	Baño	61
49.	Dormitorio	61
50.	Comedor	61
51.	Estudio	61
52.	Cocina	61
53.	Otro	61
54.	Colores principales de la aplicación	62
55.	Diagrama de flujo arranque del sistema	63
56.	Modelo conceptual	64
57.	Arquitectura del sistema Lights en base a un diagrama de paquetes	67
58.	Arquitectura del sistema diagrama de despliegue	68

59.	Muestra de la API de Philips Hue - Registrando usuario	69
60.	Muestra de la API de Philips Hue - Obteniendo lista de habitaciones	69
61.	SDKs de Philips Hue [24]	70
62.	Uso de Firebase en la aplicación	72
63.	Uso de la biblioteca AmbilWarna como color picker	73
64.	Directorio externo de proyectos de luz	76
65.	Tabla resumen ejecución de tareas en background en Android [21]	77
66.	'I need to run a task in background, how should I do it' [21] . . .	78
67.	Redmine - Plataforma de gestión del proyecto	80
68.	Redmine - Ejemplo de tarea en Redmine	80
69.	Contribuciones a la segunda versión de la aplicación en el repositorio de GitHub.	81
70.	COCOMO II Input	83
71.	COCOMO II Resultados	84
72.	Primera aproximación - Captura 1	87
73.	Primera aproximación - Captura 2	87
74.	Primera aproximación - Captura 3	87
75.	Primera aproximación - Captura 4	87
76.	Primera aproximación - Captura 5	87
77.	Primera aproximación - Captura 6	87
78.	Primera aproximación - Captura 7	87
79.	Primera aproximación - Captura 8	87
80.	Primera aproximación - Captura 9	87
81.	Aplicación enviada a Google Play esperando ser publicada.	88
82.	Segunda aproximación - Captura 1	89
83.	Segunda aproximación - Captura 2	89
84.	Segunda aproximación - Captura 3	89
85.	Segunda aproximación - Captura 4	89
86.	Segunda aproximación - Captura 5	89
87.	Segunda aproximación - Captura 6	89
88.	Segunda aproximación - Captura 7	89

89.	Segunda aproximación - Captura 8	89
90.	Gateway Philips Hue	92
91.	Dispositivo de iluminación	92
92.	Configuración de Philips Hue: Conectar gateway.	93
93.	Configuración de Philips Hue: Conectar dispositivos de luz.	93
94.	Configuración de Lights: Iniciar sesión	94
95.	Configuración de Lights: Buscando sistemas de iluminación	95
96.	Configuración de Lights: Conectar con el sistema de iluminación .	95
97.	Configuración de Lights: Buscando dispositivos de iluminación .	96
98.	Lights: Aplicación recién configurada	96
99.	Lights: Crear nueva habitación	97
100.	Lights: Lista de habitaciones disponibles	98
101.	Lights: Pérdida de la conexión Wi-Fi	99
102.	Lights: Pérdida de la conexión con el gateway	99
103.	Lights: Vista de una habitación encendida. Dispositivo: Pocophone F1.	100
104.	Lights: Vista de la habitación apagada. Dispositivo: Xiaomi MI5. .	100
105.	Lights: Añadir colores a la paleta de colores de una habitación. .	101
106.	Lights: Eliminar colores de la paleta de colores de una habitación.	101
107.	Lights: Pestaña de luces de la habitación	102
108.	Lights: Vista de un dispositivo de luz.	103
109.	Lights: Cambiar nombre de un dispositivo de luz.	103
110.	Lights: Vista de escenas.	104
111.	Lights: Vista de proyectos de luz.	104
112.	Lights: Pestaña de moods de la habitación	105
113.	Lights: Vista de un proyecto de luz	106
114.	Lights: Lista de dispositivos disponibles para reproducir el sonido en streaming.	107
115.	Lights: El sonido del proyecto de luz se está reproduciendo en un altavoz de la red.	107
116.	Lights: Notificación de proyecto de luz en ejecución.	108
117.	Lights: Proyecto de luz silenciado.	108

118. Tabla de analisis de los objetivos planteados I	110
119. Tabla de analisis de los objetivos planteados II	111
120. Anexo I: Herramientas y tecnologias utilizadas durante el proyecto I	121
121. Anexo I: Herramientas y tecnologias utilizadas durante el proyecto II	122

8. Anexo I: Herramientas y tecnologías utilizadas durante el proyecto

Herramienta o tecnología	Web del fabricante	Observaciones
Android	https://www.android.com/intl/es_es/	Plataforma para la que se ha llevado a cabo el sistema
Android Studio	https://developer.android.com/studio	Entorno de desarrollo de Android
Photoshop	https://www.adobe.com/es/products/photoshop.html	Utilizado para editar las imágenes
Vectorizer	https://www.vectorizer.io/	Utilizado para vectorizar imágenes
Flaticon	https://www.flaticon.com/home	Banco de imágenes
Visual Studio Code	https://code.visualstudio.com/	Utilizado para editar los archivos JSON
Firebase	https://firebase.google.com/	Base de datos cloud utilizada
Philips Hue	https://www2.meethue.com	Sistema de iluminación utilizado
SDK de Philips Hue	https://github.com/philipshue/philipshuesdkandroid	SDK utilizado para manejar Philips Hue
API de Philips Hue	https://developers.meethue.com/	Utilizado para hacer debug del sistema
Google Home	https://store.google.com/es/product/google_home_mini	Altavoz
Google Chromecast	https://store.google.com/es/product/chromecast	Altavoz
Google Cast	https://developers.google.com/cast/	Tecnología que nos permite hacer el streaming del audio
Overleaf	https://www.overleaf.com	Herramienta utilizada para escribir la memoria en Latex
Google Docs	https://docs.google.com/	Utilizado para elaborar algunos documentos
Git	https://git-scm.com/	Utilizado para el control de versiones

Figura 120: Anexo I: Herramientas y tecnologías utilizadas durante el proyecto I

GitHub	https://github.com/	Plataforma para alojar el proyecto controlado con Git
Google Play	https://play.google.com/	Mercado utilizado
Google Developer Console	https://developer.android.com/distribute/console?hl=es	Herramienta que nos ha permitido publicar la aplicación, y utilizar algunas funciones como el inicio de sesión de Google
Subversion	https://subversion.apache.org/	Controlador de versiones usado para los documentos
Google Drive	https://www.google.com/intl/es_ALL/drive/	Utilizado para compartir archivos
Balsamiq	https://balsamiq.com/	Utilizado para elaborar wireframes
Draw	https://www.draw.io/	Utilizado para elaborar diagramas
Skype	https://www.skype.com/es/	Utilizado para comunicarnos
AndroidX	https://developer.android.com/jetpack/androidx	Biblioteca Android
AmbilWarna	https://github.com/yukuku/ambilwarna	Color Picker
Toggl	https://toggl.com/	Herramienta para contabilizar el tiempo de cada tarea
Redmine	https://www.redmine.org/	Plataforma para gestionar el proyecto

Figura 121: Anexo I: Herramientas y tecnologías utilizadas durante el proyecto II