Universitat Oberta de Catalunya

uoc.edu

OpenKNXController

Jorge Lozano Santamaría
Grado en Ingeniería informática – Ingeniería del Software
Desarrollo de Aplicaciones para Dispositivos Móviles (Android)

David Escuer Latorre
Carles Garrigues Olivella

Fecha Entrega: 30/05/2022

FICHA DEL TRABAJO FINAL

Título del trabajo:	*OpenKNXController*
Nombre del autor:	*Jorge Lozano Santamaria*
Nombre del consultor/a:	*David Escuer Latorre*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	05/2022
Titulación::	*Grado de ingeniería informática*
Área del Trabajo Final:	*Desarrollo de Aplicaciones para Dispositivos Móviles (Android)*
Idioma del trabajo:	***Español***
Palabras clave	*KNX, domótica, smartphone*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Se persigue con esta aplicación poder dotar de un modo ágil y simple a los usuarios de viviendas con sistemas domóticos KNX (estándar europeo) para poder gestionar sus viviendas.

La idea es suministrar un fichero de marcación con la configuración de nuestra vivienda, que la aplicación móvil pueda interpretar y generar, en base a la misma, el entorno gráfico necesario.

Para el desarrollo se ha utilizado una metodología ágil, generando pequeñas iteraciones en las que se van implementando funcionalidades, desde las de más bajo nivel hasta finalizar con el entorno gráfico.

El resultado cumple con los objetivos fijados, permitiendo la aplicación, supervisar y controlar los tipos más habituales de dispositivos KNX, así como, dotando de medios para definir equipos "a medida" mediante el fichero de configuración. La aplicación funciona con una velocidad adecuada y comunica con redes KNX/IP sin complicaciones, incluyendo simplemente la dirección IP del "gateway" KNX/IP.

Concluyendo, podemos decir que se han aplicado gran parte de los conocimientos adquiridos a lo largo del grado en cuanto a planificación y gestión de proyectos (aunque se haya tratado de un proyecto de un solo hombre), desde el uso del DCU, pasando por el diseño de la arquitectura y llegando a la fase de implementación y pruebas.

Durante todo el proceso han surgido nuevas ideas o funcionalidades al respecto de la aplicación que aunque no se han podido llevar a cabo por falta

de tiempo, pero han quedado “preparadas” dentro de la solución para que en subsiguientes versiones se puedan implementar.

Abstract (in English, 250 words or less):

The goal with this app is to provide an agile and simple way of managing automated houses that are based in the european standard KNX.

The main idea is supply a markup language file (xml) with the house set up in a way that the mobile app can parse and generate , based on the file, the proper UI.

The agile methodology is been selected, by making short iterations, coding functionalities from lower level functions until finishing with the graphical interface.

The final product complies with the main goals expressed in this document, allowing the app, supervising and controlling the more usual types of KNX devices, and also, providing the ways to define “ad-hoc” devices by means of the xml file. The app works with a reasonable speed and can communicate with KNX/IP networks without glitches just by defining the IP address of the KNX gateway.

To resume, we can say that a great deal of the knowledge acquired by the degree duration its been applied, like planning or project management (bearing in mind that this has been a one-man-project), from the DCU use going through the architectural design to end with the coding and testing phases.

Through all the process new ideas about functionalities had arised about the app, and even most of them couldn’t be coded due to a lack of time, everything has been prepared to suit them in future iterations of the program beyond the “TFG” deadline.

Índice

Índice de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En la actualidad tenemos varias opciones en lo que respecta a la automatización de un inmueble, por un lado, disponemos de soluciones inalámbricas (tipo Alexa o Google Home) que permiten un grado de automatización básico con equipos que no requieren de instalación física y que pueden ser controlados por el usuario desde su dispositivo móvil. Este tipo de automatización presenta varios problemas:

- Equipos con una lógica muy básica.
- No es posible cubrir todas las necesidades de control de un inmueble.
- La lógica está gestionada remotamente en "la nube" por sistemas externos que en caso de fallo en la comunicación no nos permiten gestionar el inmueble de manera local.

Por tanto para proveer de un sistema de control totalmente autónomo la respuesta es acudir a sistemas que habitualmente son cableados y disponen de gestión local. Entre estos sistemas el más utilizado a nivel europeo (siendo un estándar para automatización) es el sistema KNX[1], este surge de la unión de los dos grandes sistemas a principios del 2000, el sistema EIB[1] belga y el sistema KNX alemán. Se trata de un sistema con lógica distribuida a través de dispositivos conectados en un bus de datos de tipo serie. Cada equipo actúa como un nodo que tiene su propia lógica de funcionamiento y puede enviar y recibir mensajes de cualquier otro nodo en la red. En la práctica actúa como una red Peer-to-Peer, donde cada nodo hace caso sólo a los mensajes que le interesan (previamente configurados) del resto. La ventaja de este sistema es que no existe ningún proceso o CPU central que dirija todo el sistema, por lo que en caso de fallo de un dispositivo, el resto del sistema sigue funcionando perdiendo sólo la funcionalidad que realizaba el dispositivo en fallo. Este comportamiento es crucial en inmuebles críticos como Hospitales, estaciones de tren o aeropuertos donde debe evitarse un fallo general del sistema de control.

El sistema KNX está formado por más de 500 fabricantes a nivel mundial con más de 8000 dispositivos registrados, siendo como es un estándar, cualquier dispositivo KNX puede formar parte de una red con dispositivos de otros fabricantes que también sean KNX. Adicionalmente la herramienta de programación ETS[2] es la misma para todos los fabricantes.

Por otro lado, es común que todos los sistemas KNX incorporen una pasarela que traduce las tramas del formato serie al IP, siendo este último conocido como KNXnet/IP. Mediante esta conversión es posible supervisar la instalación mediante soluciones de software SCADA[3]. Cuando nos referimos a inmuebles de importancia como los comentados anteriormente, o bien en el caso de viviendas, es habitual la instalación de equipos que conectan la instalación con sistemas en la nube que mediante suscripción permiten gestionar el inmueble a través de "apps" de móvil. Este último caso tiene dos inconvenientes habitualmente:

- Suelen tratarse de sistemas de suscripción mediante pago.
- Aunque la vivienda es plenamente operativa localmente (mediante pulsadores en la pared por ejemplo), si perdemos la conexión a Internet o nuestro proveedor de servicios deja de prestarnos el servicio “app”, dejaremos de tener control a través de nuestro dispositivo móvil.

Es por ello que entendemos sería justificable la existencia de una aplicación libre para móvil capaz de conectarse a nuestra red KNX, de manera local o remota, y que mediante una descripción de la instalación permita su gestión y supervisión sin necesidad de utilizar sistemas de terceros o pago de suscripciones.

1.2 Objetivos del Trabajo

Los requerimientos a nivel funcional serían los siguientes:

- (OF1) Un usuario debe poder definir las URL locales y remotas de mi instalación.
- (OF2) Un usuario debe poder definir un código de acceso a la aplicación para evitar usos maliciosos por parte de terceros.
- (OF3) Un usuario debe poder acceder a las distintas zonas del inmueble organizadas por iconos con un nombre descriptivo.
- (OF4) Un usuario debe poder gestionar todos los dispositivos de control desde una única pantalla de resumen.
- (OF5) Un usuario debe poder seleccionar el dispositivo a supervisar por tipo (iluminación, sensor, clima, persiana...)
- (OF6) Un usuario debe disponer de una pantalla de inicio con acciones globales que afecten a dispositivos de distintas zonas.
- (OF7) Como requisito de datos debe disponerse de un fichero (tipo xml) para la definición de la instalación a supervisar/controlar.
  - Mediante definición de Secciones (Planta Baja, Planta Primera,…).
  - Mediante definición de zonas dentro de secciones (Comedor, habitación de matrimonio,…).
  - Definición de elementos dentro de zonas (Luz, persiana, sensor,…).
- (OF8) Como requisito de datos debe haber soporte para los siguiente tipos de elementos:
  - Accionamiento y supervisión de circuitos On/Off.
  - Accionamientos y supervisión de circuitos con regulación 0-100%.
  - Accionamiento y supervisión de regulación de luminarias de colores RGB.
  - Accionamiento y supervisión de dispositivos de climatización básicos.

Los requerimientos a nivel no funcional:

- Diseñar una aplicación móvil libre para supervisión y control de un sistema domótico de tamaño pequeño (vivienda) basado en el estándar KNX.
- Establecer un interfaz de usuario que facilite la usabilidad en usuarios poco versados en tecnología.
- Acceso directo a la instalación KNX a través de red Wifi o conexión remota.
- Dirigida a dispositivos móviles Android.

- La aplicación será de libre distribución y código libre.

1.3 Enfoque y método seguido

La estrategia elegida es la desarrollar una aplicación nueva para plataforma Android utilizando en SDK oficial Android Studio[4]. Sin embargo, para la parte de la aplicación correspondiente a las comunicaciones a bajo nivel en formato KNXnet/IP utilizaremos la librería “Calimero”[5], se trata de una librería para Java que implementa la pila de comunicaciones con KNXnet/IP.
De este modo, el trabajo de desarrollo se centrará en:

- Interpretación del fichero de configuración de la instalación.
- Desarrollo lógica de negocio.
- Desarrollo de la interfaz de usuario.

El diseño por nuestra parte de la pila de comunicaciones en KNXnet/IP podría ser objeto de un TFG por si mismo, por lo tanto, para centrar los recursos temporales disponibles en el diseño de la aplicación Android, considero que la mejor opción es utilizar una librería existente (ya probada en otros desarrollos). Para el interfaz de usuario buscaremos algo relativamente sencillo que sea muy directo en su manejo, por lo que utilizaremos los objetos gráficos estándar disponibles en Android Studio por defecto. Con todo ello en mente ahorraremos tiempo pues no se va a requerir el aprendizaje de ningún “framework” específico.

Utilizaremos una metodología ágil, creando varios prototipos que implementen distintos aspectos de la funcionalidad. Empezando por la interpretación del fichero de configuración junto con clases básicas, para después generar otro con las capacidades de conexión a la red KNX y resto de clases que representen los dispositivos a controlar. Y por último, generar un prototipo que implemente la parte gráfica de la aplicación o UI.

Es más fácil usar esta metodología en una aplicación de este tipo en la que hay comunicaciones con dispositivos de terceros de por medio, ya que, de hacerlo de manera clásica en cascada, en caso de encontrarnos con errores en esta parte tan crítica, podría alterarnos el resto de clases y objetos ya definidos, provocando modificaciones por toda la aplicación.

1.4 Planificación del Trabajo

Los recursos a nivel material necesarios para el realizar el proyecto serán los siguientes:

- Pc con Android Studio para desarrollo de la aplicación.
- Dispositivo Android o emulador para testeo y prototipado.
- Instalación KNX con pasarela IP para realizar las pruebas de comunicación y testeo de funcionamiento.

Los recursos a nivel humano estarán compuestos por una única persona (yo mismo).
La planificación propuesta es la siguiente:

		Nombre	Duracion	Inicio	Terminado
1		**Diseño y Arquitectura**	**20 days?**	**3/3/22 8:00**	**30/3/22 17:00**
2		Análisis [DCU]	7 days	3/3/22 8:00	9/3/22 17:00
3		Diseño [DCU]	10 days	10/3/22 8:00	19/3/22 17:00
4		Evaluación [DCU]	7 days	20/3/22 8:00	26/3/22 17:00
5		Definición de los casos de uso	2 days?	27/3/22 8:00	28/3/22 17:00
6		Diseño arquitectura aplicación	2 days?	29/3/22 8:00	30/3/22 17:00
7		**Implementación**	**42 days?**	**31/3/22 8:00**	**11/5/22 17:00**
8		Implementación Invariante	7 days?	31/3/22 8:00	6/4/22 17:00
9		Desarrollo parser fichero xml	2 days?	7/4/22 8:00	8/4/22 17:00
10		Implementación librería "Calimero"	5 days?	9/4/22 8:00	13/4/22 17:00
11		Implementación pantallas generales	10 days?	14/4/22 8:00	23/4/22 17:00
12		Implementación elementos de control 1/3	3 days?	24/4/22 8:00	26/4/22 17:00
13		Despliegue de primer prototipo	1 day?	27/4/22 8:00	27/4/22 17:00
14		Test de funcionamiento primer prototipo	2 days?	28/4/22 8:00	29/4/22 17:00
15		Implementación elementos de control 2/3	3 days?	30/4/22 8:00	2/5/22 17:00
16		Despliegue del segundo prototipo	1 day?	3/5/22 8:00	3/5/22 17:00
17		Test de funcionamiento segundo prototipo	2 days?	4/5/22 8:00	5/5/22 17:00
18		Implementación elementos de control 3/3	3 days?	6/5/22 8:00	8/5/22 17:00
19		Despliegue del tercer prototipo	1 day?	9/5/22 8:00	9/5/22 17:00
20		Test finales	2 days?	10/5/22 8:00	11/5/22 17:00
21		**Entrega Final**	**13 days?**	**12/5/22 8:00**	**30/5/22 17:00**
22		Despliegue aplicación	4 days?	12/5/22 8:00	15/5/22 17:00
23		Finalización Memoria	11 days?	16/5/22 8:00	26/5/22 17:00
24		Realización vídeo descriptivo	4 days?	27/5/22 8:00	30/5/22 17:00

Figura 1: Planificación Tabla

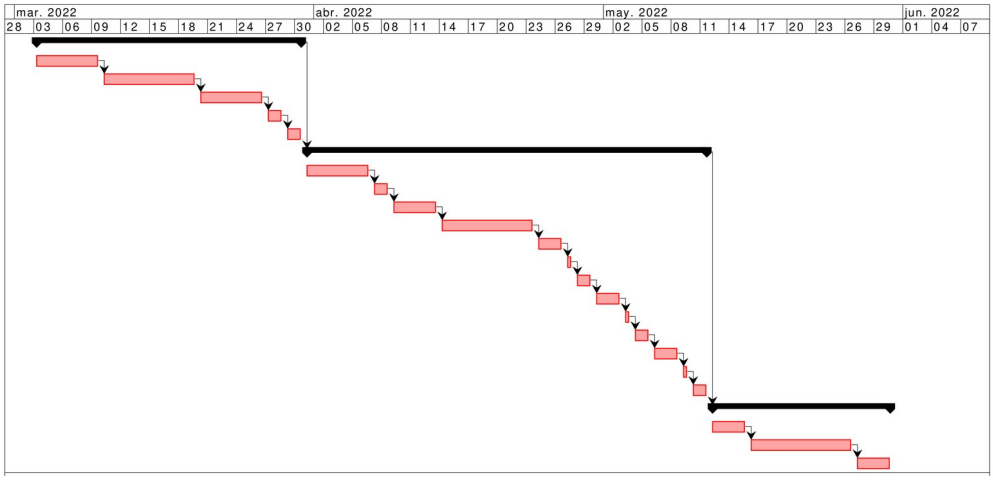


Figura 2: Planificación Gantt

La planificación por horas para cada tarea es la siguiente:

Tarea	**Horas en días laborables**	**Horas en festivos**
Análisis [DCU]	7h	6h
Diseño [DCU]	9h	9h
Evaluación [DCU]	5h	6h
Definición casos de uso	1,5h	3h
Arquitectura aplicación	3h	0h
Implementación	42h	36h
Despliegue aplicación	4h	6h
Finalización de memoria	11h	6h
Vídeo descriptivo	4h	6h

Figura 3: Planificación por horas

1.5 Breve sumario de productos obtenidos

Los entregables del proyecto serán los siguientes:
Fichero de instalación Android (fichero con extensión apk).

- [E1] Código fuente de la aplicación (proyecto de Android Studio).
- [E2] Memoria del proyecto.
- [E3] Video explicativo del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

- Capítulo 2.1. “DCU”.
- Capítulo 2.2 Definición detallada de casos de uso.
- Capítulo 2.3. Diseño de la aplicación, definición del invariante y arquitectura elegida.
- Capítulo 2.4. Estructura del fichero XML de configuración.
- Capítulo 2.5. Decisiones de implementación entorno gráfico.
- Capítulo 2.6. Pruebas.
- Capítulo 3. Conclusiones.

2. DCU

2.1 DCU Usuarios y contextos de uso.

2.1.1 Justificación del método de indagación empleado.

El método seleccionado para llevar a cabo la indagación ha sido el de “entrevistas en profundidad”.

Dado que estamos desarrollando un producto nuevo que no existe el método que mejor se adapta es el de realizar “entrevistas en profundidad” a distintas personas con perfiles diferenciados para conocer que características consideran que debería tener nuestra aplicación y cuales serían sus preferencias.

En este caso no podemos usar “Observación contextual” en el momento actual, ya que, este método de indagación se suele utilizar para mejorar o depurar aplicaciones a existentes, mediante un “test de usuarios”. Dado que buscamos obtener una orientación sobre que debemos implementar, dejaremos pues, este método para cuando debamos refinar nuestros prototipos.

Se han desestimado las “Dinámicas de grupo”, ya que, debido a la pandemia todavía existente no es posible reunir a un grupo de personas, además los problemas de agenda de los posibles asistentes hacen que debamos decantarnos por métodos asíncronos o bien que se puedan evaluar a individuos en distintos momentos en el tiempo.

Las encuestas también han sido desestimadas, básicamente porque el grupo de personas sobre las que podría haberse realizado ya han formado parte de los métodos expuestos al inicio del capítulo, por tanto no arrojarían resultados distintos a los ya obtenidos.

2.1.2 Planteamiento y Desarrollo.

Para el método de entrevistas nos basaremos en la guía de buenas prácticas de Nacho Madrid[6], Tomamos notas manuscritas de las respuestas de los distintos usuarios y usaremos una metodología estructurada, ya que, vamos a tratar un tema que hemos identificado y estudiado previamente (aplicación de gestión para domótica).

Para llevar a cabo las entrevistas generamos un guion que nos permita obtener información que nos ayudará a definir la aplicación que queremos desarrollar.

Plantearemos las siguientes preguntas:

- ¿Tienes alguna relación con sistemas domóticos o has utilizado alguno?
- En caso afirmativo, como interactúas habitualmente con el mismo.
- ¿Surge algún problema en el uso cotidiano?
- Entre los interfaces que utilizas para controlar tu casa ¿usas alguna aplicación móvil?
- De usar una, ¿Que es lo que más y lo que menos te gusta?

- ¿Cuáles son las tareas que realizas más a menudo con el sistema?
- ¿Consideras que puedes realizar todas las tareas que deseas con los interfaces de que dispones actualmente?

En base a los resultados del método de indagación podremos obtener resultados al respecto de que funciones de base deberá soportar nuestra aplicación.

2.1.3 Resultados.

A continuación veremos los resultados de varios perfiles tanto en la entrevista como en el test de usuario realizado.

Perfil usuario 1: 42 años, mujer casada, administrativa en correduría de seguros, uso de “smartphone” de manera habitual pero no de otras tecnologías.No suele comprar aplicaciones y utiliza las que vienen de serie con el dispositivo. No está interesada en la tecnología en general, mientras que su marido sí, por este último motivo vive en una vivienda que cuenta con un sistema de control automatizado.

Resultado de la entrevista:

¿Tienes alguna relación con sistemas domóticos o has utilizado alguno?
La usuaria 1, indica que dispone de una vivienda domótica que compró su marido, utiliza a diario las funciones básicas de la misma, prácticamente como si fuera un vivienda convencional. En la medida de lo posible, deja que sea su marido el que lleve a cabo las acciones más “complejas”.
En caso afirmativo, como interactúas habitualmente con el mismo.
La usuaria utiliza los pulsadores de la pared, que disponen de iconos identificativos en cada tecla. También dispone de una pantalla táctil, pero que sólo utilizar para realizar un apagado general de la casa cuando sale de la misma.
¿Surge algún problema en el uso cotidiano?
Normalmente no, ya que, hace un uso como si de una vivienda convencional se tratase. Pero si indica que en ocasiones cuando está fuera de la casa le gustaría asegurarse de que ha apagado la calefacción o ha recogido los toldos y le toca volver a casa para comprobarlo al no disponer de herramientas remotas.
Entre los interfaces que utilizas para controlar tu casa ¿usas alguna aplicación móvil?
Existe un acceso a un servidor web instalado en la vivienda, en realidad se trata de la pantalla táctil que dispone de este servicio, sin embargo, según comenta es fácil de usar si se hace desde un ordenador por el tamaño de la pantalla, pero si se utiliza desde el móvil, todo se visualiza extremadamente pequeño y no es fácil de manejar. Además, la página tiene muchos gráficos que representan los planos de la vivienda y le cuesta bastante de cargar. Por lo que indica, es muy poco útil si se desea realizar alguna acción o ver un valor rápidamente.
De usar una, ¿Que es lo que más y lo que menos te gusta?
Como se ha comentado en la pregunta anterior, no dispone de una aplicación móvil como tal, por lo que no tiene una opinión al respecto.
¿Cuáles son las tareas que realizas más a menudo con el sistema?
Las acciones más habituales son:

• Encender/Apagar luces. • Recoger/extender los toldos. • Encender/apagar/regular los radiadores. • Encender/apagar/regular las máquinas de aire acondicionado. • Cerrar el aporte general de agua a la vivienda. • Cerrar el aporte de gas a la vivienda.
¿Consideras que puedes realizar todas las tareas que deseas con los interfaces de que dispones actualmente?
Para el día a día, nos indica que es más que suficiente con los accionamientos existentes por medio de los pulsadores físicos de la vivienda y las escenas generales disponibles en la pantalla táctil. Sin embargo, no tiene la posibilidad de ejecutar ninguna acción de modo “rápido”, ya que, el servicio web del que dispone tarda demasiado en cargar y su uso es muy complejo desde el móvil al no estar diseñado para visualizarse en estos equipos.

De la primera usuaria, deducimos que desearía utilizar la aplicación fuera de casa para poder realizar actuaciones remotas de un modo sencillo y poder revisar el estado de algunos elementos también de manera rápida.

Las tareas que se derivan de esta usuaria son:

- Poder encender/apagar todos los elementos de su vivienda.
- Poder ejecutar escenas que realicen un conjunto de acciones (como un cierre general de la vivienda)

Entre las características que deberá tener la aplicación derivadas de esta usuaria, podemos indicar que la aplicación debe mostrar los valores rápidamente así como ejecutar las ordenes del mismo modo. Debe también tener un diseño simple y accesible, evitando usar un entorno gráfico recargado (evitar diseños de vivienda personalizados buscando algo más genérico y directo).

Perfil Usuario 2: 20 años, estudiante, el móvil es casi “una extensión” de su cuerpo, lo usa de manera continuada y dispone de aplicaciones de todo tipo, tanto de pago como gratuitas. Desarrolla casi toda su actividad social a través del mismo, así como, cualquier gestión que deba realizar en el día a día.

Resultado de la entrevista:

¿Tienes alguna relación con sistemas domóticos o has utilizado alguno?
No dispone de una vivienda con una instalación avanzada de domótica, pero si que ha ido adquiriendo dispositivos Wifi que son compatibles con el asistente virtual “Amazon Alexa”[7], como: bombillas (monocolor y RGB), enchufes y un purificador de aire.
En caso afirmativo, como interactúas habitualmente con el mismo.

El usuario utiliza los altavoces inteligentes que tiene distribuidos por la vivienda además de la aplicación para Android[8] de “Amazon Alexa”.
¿Surge algún problema en el uso cotidiano?
Normalmente, el mayor problema viene dado porque las luces que tienen bombillas “inteligentes” no pueden encenderse desde el interruptor de la pared y requiere siempre de usar comandos vocales para encenderlas o apagarlas.
Entre los interfaces que utilizas para controlar tu casa ¿usas alguna aplicación móvil?
Utiliza la aplicación de “Amazon Alexa”, que aunque no es una aplicación propiamente de gestión domótica, si dispone de funciones de este tipo entre sus caracterísitcas.
De usar una, ¿Que es lo que más y lo que menos te gusta?
Lo que menos le gusta es la velocidad de la misma, cada vez que la abre, tarda en ocasiones entre 10-15 segundos en acceder a la pantalla principal de la aplicación, después, al acceder a los dispositivos instalados, el proceso de obtener información (del purificador por ejemplo) puede tardar otros 20seg o a veces incluso más. También tiene el problema que en varias ocasiones a tenido cortes en el acceso a Internet, y no ha podido manejar ninguno de los dispositivos que tiene instalados hasta que ha vuelto la conexión. Un punto positivo es que se pueden agrupar los dispositivos por estancia, haciendo más fácil el uso de la aplicación. Lo que más le gusta de la aplicación es que no tiene demasiadas opciones y es muy directa de manejar, todos los elementos disponen de iconos identificativos que cambian de color/forma para representar los diferentes estados. La ejecución de escenas también es muy sencilla, siendo en este caso que el propio usuario puede hacer su configuración directamente.
¿Cuáles son las tareas que realizas más a menudo con el sistema?
Las tareas que realiza más habitualmente son: • Encender/Apagar/regular luces de la habitación. • Subir/bajar la persiana de su habitación. • Encender/apagar/regular el suelo radiante de su habitación. • Encender/apagar/regular la máquina de aire acondicionado.
¿Consideras que puedes realizar todas las tareas que deseas con los interfaces de que dispones actualmente?
Actualmente el usuario puede realizar todas las tareas que desea con los medios de los que dispone.

De este usuario deducimos que hará uso de la aplicación tanto dentro de su vivienda como fuera de ella, al ser un usuario que hace un uso intensivo de tecnología podemos prever que hará uso de la aplicación de manera continuada.

Las tareas que realiza este usuario serán del tipo:

- Poder encender/apagar/regular los circuitos de iluminación.
- Poder encender/apagar elementos de climatización

Para este usuario es importante que la aplicación funcione en sus diferentes dispositivos móviles, ya que, dada su afición por la tecnología dispone de varios. Por tanto, la aplicación deberá ajustarse correctamente a distintos tamaños de pantalla.

Perfil usuario 2: 37 años, hombre casado, diseñador gráfico 3D, acostumbrado a utilizar tecnología desde muy joven, tiene una gran experiencia tanto en el uso de ordenadores con software de productividad y diseño, así como un uso avanzado del “smartphone”, usándolo para todo tipo de tareas como videoconferencias, juegos, GPS, email, etc.

Resultado de la entrevista:

¿Tienes alguna relación con sistemas domóticos o has utilizado alguno?
La vivienda donde vive dispone de un sistema de domótica, por lo que está familiarizado con estos sistemas.
En caso afirmativo, como interactúas habitualmente con el mismo.
Utiliza los distintos controles repartidos por la vivienda, así como una tablet que hace las veces de pantalla táctil, conectada a un servidor web local y por último dispone de una aplicación móvil de pago que el fabricante de la pantalla pone a disposición de los clientes a través de sus servidores “cloud”.
¿Surge algún problema en el uso cotidiano?
El usuario ha crecido utilizando el sistema, por lo que su uso es algo innato para él.
Entre los interfaces que utilizas para controlar tu casa ¿usas alguna aplicación móvil?
Dispone de la aplicación de móvil del fabricante de la pantalla.
De usar una, ¿Que es lo que más y lo que menos te gusta?
La aplicación funciona de manera correcta y permite tener la casa organizada por habitaciones, pero al tener que conectar con un servidor remoto para que lo redirija a la vivienda (haga un túnel), el acceso no siempre es fluido. Además el fabricante ha tenido varias caídas de servicio recientemente imposibilitando el acceso remoto a la vivienda. Por último indicar que es de pago recurrente, que puede ser mensual o anual.
¿Cuáles son las tareas que realizas más a menudo con el sistema?
Las tareas que realiza más habitualmente son: • Encender/apagar/regular/cambiar color de luces de lámparas. • Encender/apagar enchufes de manera remota. • Encender/apagar el aparato purificador de aire.

¿Consideras que puedes realizar todas las tareas que deseas con los interfaces de que dispones actualmente?
Para llevar a cabo las funciones de los equipos que tiene instalados tanto desde los altavoces como desde la aplicación móvil puede llevar a cabo todas las acciones posibles. Sin embargo, como ya se ha comentado en las otras preguntas, el no poder usar los elementos directamente sin el uso del altavoz o la aplicación móvil, a veces hace farragoso el utilizarlos. Además la aplicación móvil necesita conectarse a los servidores de Amazon para acceder a nuestra vivienda, lo que hace que la velocidad se resienta.

De manera similar al usuario del perfil 2 deducimos que hará uso de la aplicación tanto dentro de su vivienda como fuera de ella, al ser un usuario que hace un uso intensivo de tecnología podemos prever que hará uso de la aplicación de manera continuada.

Las tareas que realiza este usuario serán del tipo:

- Poder encender/apagar/regular los circuitos de iluminación.
- Poder encender/apagar elementos de climatización

Es importante que al igual que ocurre con la aplicación de “Amazon Alexa” sea posible dividir la vivienda en secciones o áreas donde se ubiquen los diferentes dispositivos que podemos controlar, ya que, eso le facilita mucho a este usuario el poder localizarlos.

Se realiza la misma entrevista a otros dos usuarios en la franja entre 20-25 años, otros dos en la franja de 40-45 años. En ambos casos ninguno dispone de sistemas domóticos ni los ha utilizado, por lo que la respuesta a la primera pregunta del guion es “No” en todos los casos y no es posible continuar con el guion de la entrevista. Aunque en ambos casos manifiestan interés por conocer más del tema.

También se realiza la entrevista a tres usuarios en la franja de los 55-65 años, pero además de no disponer de sistemas de control domésticos manifiestan no tener ningún interés sobre los mismos.

Podemos concluir que la franja de edad objetivo será entre los 20-45 años, ya que, las franjas por encima no han mostrado interés y por debajo de los 20 años la capacidad adquisitiva es muy reducida.

2.2 DCU Diseño Conceptual.

2.2.1 Perfiles y escenarios de uso.

Dado el limitado espectro de usuarios que podrán tener acceso a la aplicación las entrevistas no han podido ser “masivas” siendo que sólo unos pocos

usuarios tenían relación con la materia a la que va dirigida la aplicación a desarrollar.

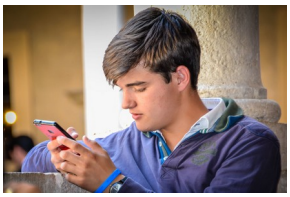
Por ello las personas de las entrevistas serán nuestras “Personas” que ubicaremos en diferentes escenarios, les dotaremos de un nombre ficticio, aunque, sus motivaciones e intereses serán directamente los que hemos obtenido de las entrevistas.

Perfil usuario 1, la llamaremos María. (mujer entre 40-45 años sin gran afinidad por la tecnología).

	“No quiero complicarme la vida para encender una luz en mi casa”

- Objetivos:
  - Apagar toda la casa con un solo “click”.
  - Conocer la temperatura de la casa estando fuera de la misma.
- Comportamientos:
  - Usuaria de móvil casual.
  - No está interesada en la tecnología pero sí en la comodidad.
- Necesidades:
  - No tener complicaciones a la hora de manejar los dispositivos su casa.
  - Estar informada de manera rápida si sucede cualquier evento en su domicilio.
- Escenarios:
  - “María ha salido de casa para el fin de semana, no recuerda si ha cerrado el gas, pero ya está lejos en el coche y no puede volver atrás, desearía poder hacerlo desde su móvil cuando pare a tomar un café en la próxima parada”.
  - “María está trabajando y sus hijos van a llegar a casa del colegio en una hora, le gustaría saber si la temperatura de la casa está bien porque su hijo menor está acatarrado y hoy hace mucho frio. Desde su móvil puede comprobarla y si es necesario encender la calefacción para mejorar el ambiente”.

Perfil usuario 2, lo llamaremos Toni. (persona entre 20-30 años, “nativo digital”).

	“No puedo vivir sin mi ‘Smartphone’, sin él no se que hacer”

- Objetivos:
  - Controlar la iluminación de su habitación, creando escenas de iluminación según la actividad que vaya a desarrollar.
  - Controlar la climatización de su habitación de manera independiente al resto de la vivienda.
- Comportamientos:
  - Nativo digital.
  - Se relaciona mayoritariamente con la gente a través de redes sociales.
  - Desea vincular todas sus actividades con el “Smartphone”, si no hay una aplicación disponible para una actividad la descarta.
- Necesidades:
  - Disponer de las opciones de control de su habitación en su móvil, de manera rápida.
  - Manejar toda la información posible desde una pantalla, no tiene tiempo de acceder a “mil menús”.
  - Poder acceder desde dentro de casa y desde fuera.
- Escenarios:
  - “Toni quiere jugar una partida al nuevo Halo de su Xbox series x, pero antes decide ejecutar la escena “gaming” en su móvil, que reduce la iluminación a un 15% y la colorea de rojo para generar un ambiente más acorde al juego”.
  - “Toni se dispone a ver una película con unos amigos en su habitación, antes de empezar ejecuta la escena “cine” en su móvil que apaga la iluminación encima del televisor y deja encendidas las luces junto a la puerta como si de un cine se tratase. Adicionalmente, la escena regula el aire acondicionado a una temperatura de 21ºC”

Perfil usuario 3, lo llamaremos Carlos. (Persona entre 30-40 años , “Millenial”).

	“La tecnología es una parte básica de mi vida, trabajo con ella y disfruto con ella”

- Objetivos:
  - Controlar la iluminación de la vivienda, pudiendo realizar encendidos aleatorios cuando no se encuentra en casa para simular presencia.
  - Controlar la climatización y la calidad del aire de la vivienda.
  - Poder disponer de escenas de comfort para ver la tv, jugar a la consola o hacer fiestas con los amigos.
- Comportamientos:
  - Gran amante de la tecnología, actual y “retro”.
  - Buen nivel adquisitivo que le permite tener lo “último” en tecnología.
  - Exigente con los dispositivos y sistemas de que dispone.
- Necesidades:
  - Tener su vivienda controlada tanto dentro de ella como fuera.
  - Disponer de información inmediata de cualquier evento que se pueda producir en su vivienda.
- Escenarios:
  - “Carlos ha salido de vacaciones por un par de semanas, ha activado el modo simulación en su vivienda, de manera que al anochecer se encienden las luces de manera aleatoria, se bajan parcialmente las persianas y al amanecer se enciende el riego del jardín”.
  - “Carlos va a acostar a su bebé después del baño, antes de bañarlo activa desde su móvil el purificador de aire de la habitación y activa el radiador para que la temperatura sea la adecuada para que el bebé pueda dormir”.
  - “Carlos y su mujer se han ido a cenar el sábado por la noche, cuando están llegando al restaurante Carlos no recuerda si ha apagado la plancha con la que se había planchado la camisa, coge su móvil y desconecta el enchufe de seguridad en el que se encuentra conectada la plancha”.

2.2.2 Flujos de interacción.

A raíz de toda la información obtenida en las entrevistas y el análisis de los perfiles de usuario, podemos definir una serie de flujos de interacción que nuestra aplicación móvil tendrá que implementar.

Flujo de interacción para “acceso a un área en modo local o remoto”.

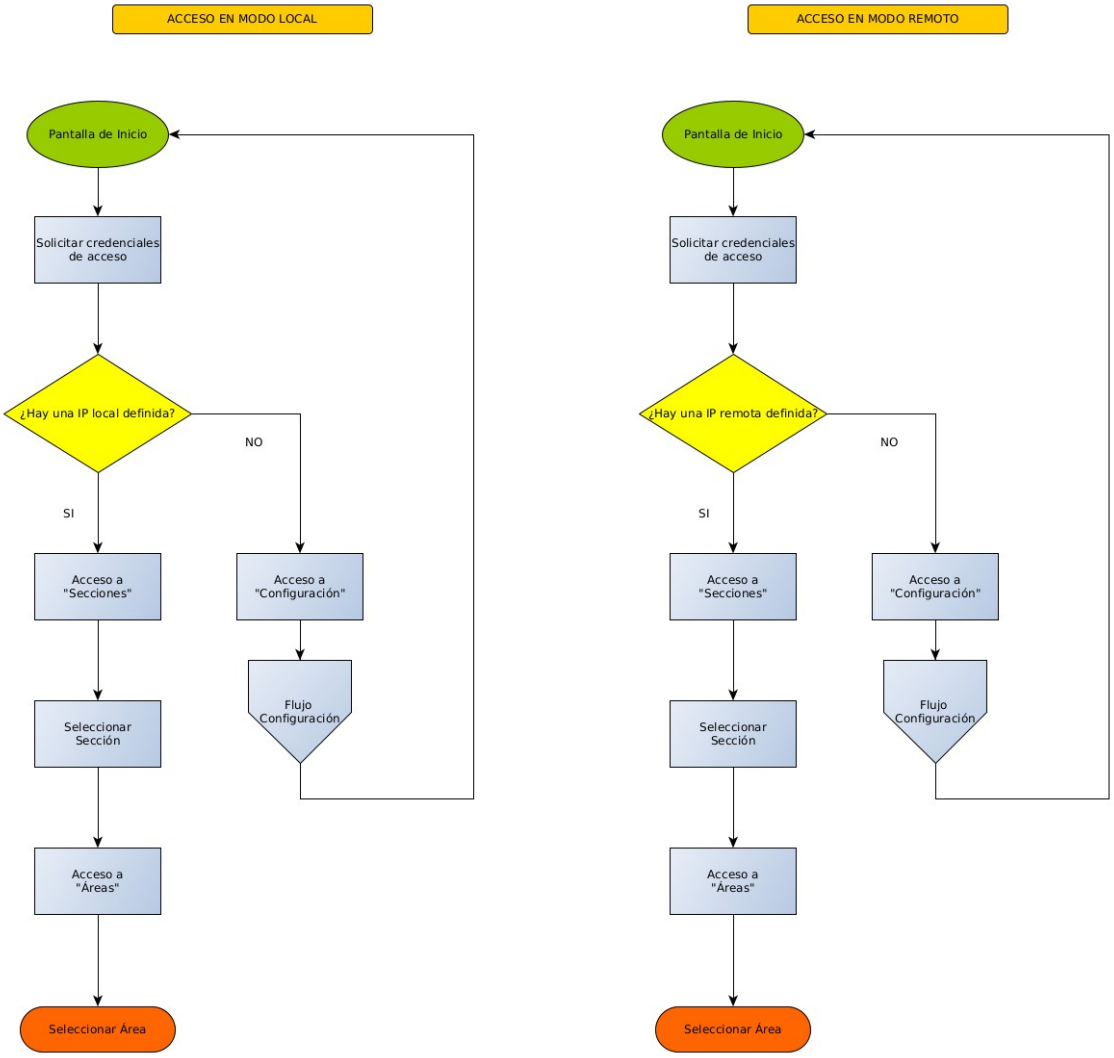


Figura 4: Flujo de interacción para “acceso a un área en modo local o remoto”.

Flujo de interacción para “Configuración del sistema”.

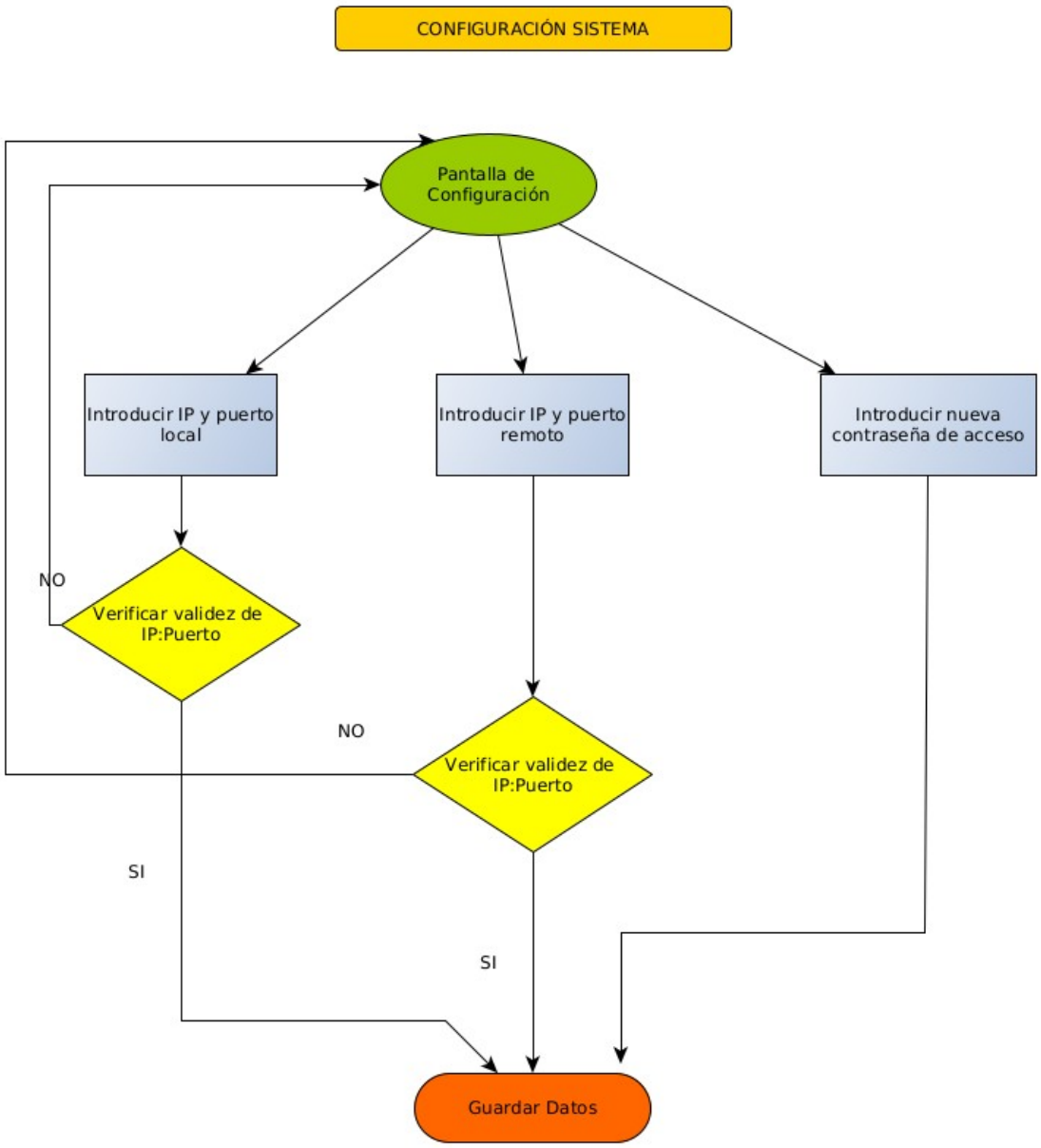


Figura 5: Flujo de interacción para “Configuración del sistema”.

Flujo de interacción para “Selección de elemento de control”.

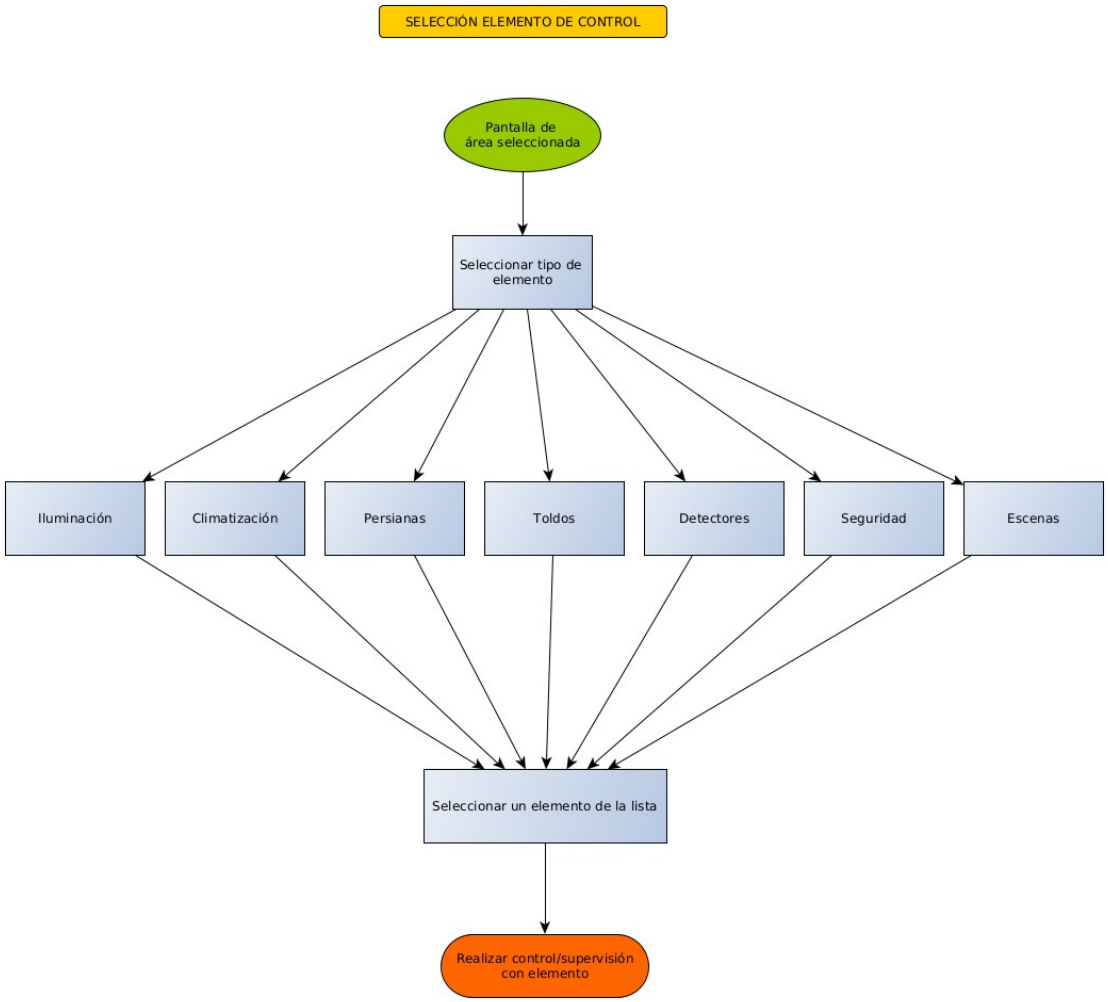


Figura 6: Flujo de interacción para “Selección de elemento de control”.

2.3 DCU Prototipado.

2.3.1 “Sketches”

Se adjuntan a continuación los bocetos a mano alzada para la primera versión del prototipo de la aplicación.

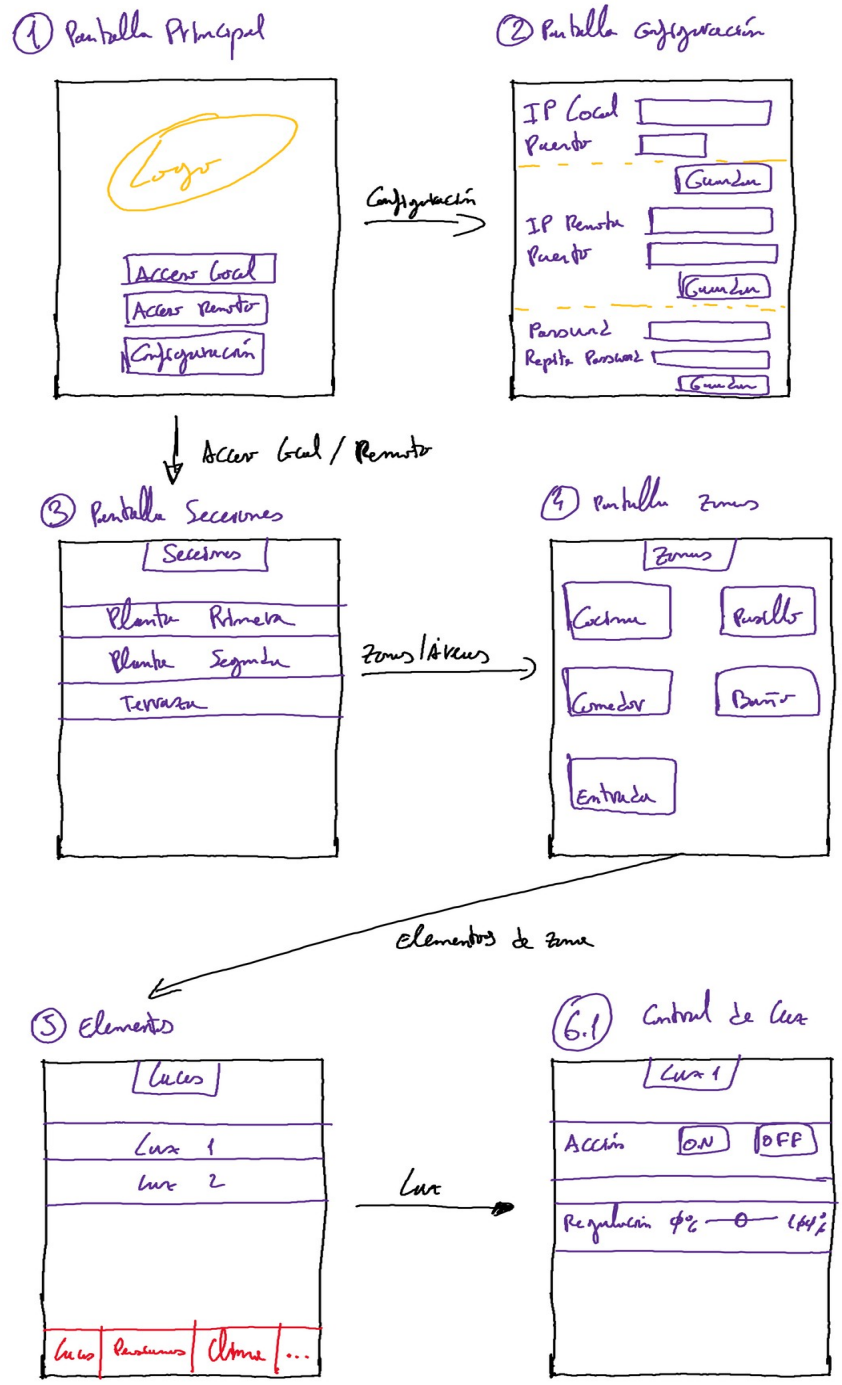


Figura 7: Prototipo Sketch 1/2

Figura 8: Prototipo Sketch 2/2

Partimos de una pantalla de inicio desde la que podemos acceder a la configuración del sistema o bien realizar un acceso a nuestra instalación de manera local o remota.

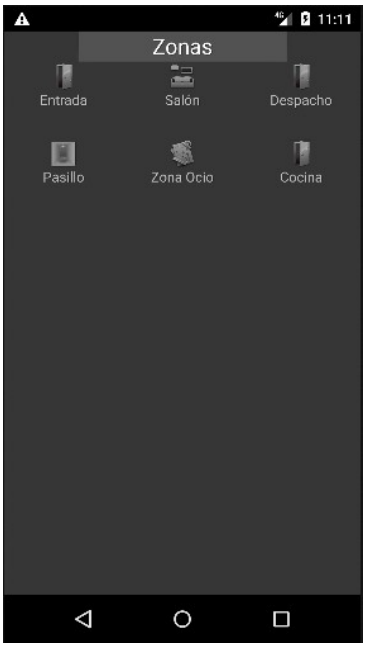
Una vez accedido, deberemos seleccionar una sección de la vivienda y dentro de esta un área, a continuación se mostrarán todos los elementos gestionables en la misma. Pulsando sobre cada elemento de la lista accederemos a su pantalla de control/supervisión.

2.3.2 Prototipo de alta fidelidad.

Se ha desarrollado en el propio Android Studio una serie de pantallas (sin lógica de negocio) para representar los conceptos indicados en el “sketch” manual, adjuntamos las capturas a continuación.

1 Pantalla Principal	2. Pantalla Configuración	3. Pantalla Secciones

Figura 9: Prototipo de alta fidelidad 1/3



4. Pantalla Zonas

5. Elementos (Luces)

6.1 Control de luz

Figura 10: Prototipo de alta fidelidad 2/3

6.2 RGB

6.3 Persianas

6.4 Clima

Figura 11: Prototipo de alta fidelidad 3/3

2.4 DCU Tests de usuarios.

2.4.1 Definición del alcance

Las pruebas se llevarán a cabo sobre cada iteración de la aplicación prototipo que hemos desarrollado.

Durante la sesión de pruebas, que se realizará en persona, el moderador solicitará a los usuarios que lleven a cabo una serie de tareas que deseamos confirmar el funcionamiento para varias de las acciones que la aplicación puesta a prueba debe poder desarrollar con facilidad.

Al acabar la sesión, se les preguntará sobre como ha sido su experiencia con la aplicación, si han encontrado que les ha resultado fácil llevar a cabo las tareas, que mejorarían o que función no han encontrado y les hubiera gustado que estuviese presente.

Para la recolección de la información se tomarán notas mediante grabación en vídeo, audio o el uso del clásico papel y lápiz.

El objetivo que se pretende conseguir en las sucesivas sesiones de tests es conocer la complejidad que experimenta el usuario para realizar tareas habituales como son las siguientes:

- Configurar el acceso a la vivienda en modo local.
- Definir una palabra (contraseña) de acceso a la aplicación.
- Encender la luz encima del sofá que se encuentra en el salón.
- Bajar todas las persianas del salón.
- Encender la climatización de la casa y poner la consigna en 23ºC

2.4.2 Descripción del perfil de los participantes

Una de las tareas a llevar a cabo será la de seleccionar los usuarios para agruparlos en perfiles que realicen los test. Esta tarea es similar a la llevada a cabo en el análisis de DCU, de hecho, podemos utilizar el mismo conjunto de perfiles que usamos en su momento para especificar las necesidades.

Una vez establecidos dichos perfiles, llevaremos a cabo los tests, normalmente de manera anónima por lo que no suele ser necesario rellenar un documento de conformidad para la participación.

2.4.3 Definición del guión de las sesiones.

Para la consecución de los objetivos vamos a plantear una serie de preguntas relacionadas con los mismos, observaremos para cada pregunta el desempeño del usuario y sus reacciones y/o comentarios.

1. **Configurar el acceso a la vivienda en modo local.**

   La vivienda que vamos a manejar dispone de un punto de acceso con la IP: 192.168.0.10 ¿serías capaz de configurar la aplicación móvil para poder acceder a la misma?

2. **Definir una contraseña de acceso a la aplicación.**

   Queremos proteger el acceso a la aplicación para que no acceda cualquier persona ¿sabrías como configurar una contraseña, por ejemplo “4312”?

3. **Encender la luz encima del sofá que se encuentra en el salón.**

   Vamos a tener visita y me gustaría recibirlos en el salón ¿sabrías encender la luz del sofá donde nos vamos a sentar?

4. **Bajar todas las persianas del salón.**

   Parece que se ha puesto a llover intensamente y la lluvia está golpeando en las ventanas, sería conveniente bajar todas las persianas en el salón ¿Sabrías hacerlo?

5. **Encender la climatización de la casa y poner la consigna en 23ºC**

   Se ha vuelto un día bastante frío, ¿sabrías encender la calefacción de la casa y ponerla en 23ºC?

2.4.4 Descripción breve de la realización de las sesiones

Se han planteado las tareas siguiendo el orden indicado en el apartado anterior, cada usuario reaccionará a las mismas y se tomará nota del tiempo que ha tardado en resolverlas (en cuanto a poco tiempo, mucho tiempo o no conseguido), del mismo modo, se tomará nota de la gestualidad de los mismos para identificar si les ha parecido fácil o difícil la tarea. Por último se tomará nota de los comentarios y sugerencias realizadas por los mismos.

2.4.5 Análisis de los datos cuantitativos

Una vez obtenidos los resultados de los diferentes tests se llevará a cabo un análisis de los resultados por cada tipo de tarea llevada a cabo, estos resultados se podrán graficar para tener una mejor interpretación de los mismos.

Ratio de éxito por tarea:

Tarea	Ratio de éxito
1	100%
2	0%
3	100%
4	80%
5	100%

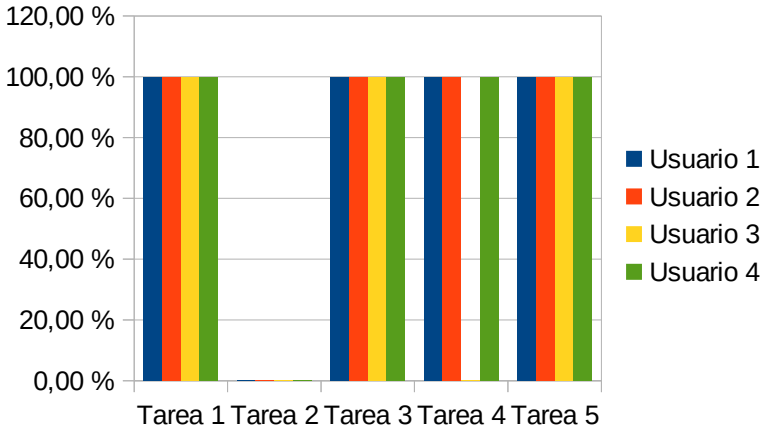


Figura 12: Ejemplo gráfica datos cuantitativos 1/2

Tiempo medio por tarea.

Tarea	Tiempo en minutos
1	5,75
2	16,25
3	1,75
4	5,75
5	0,625

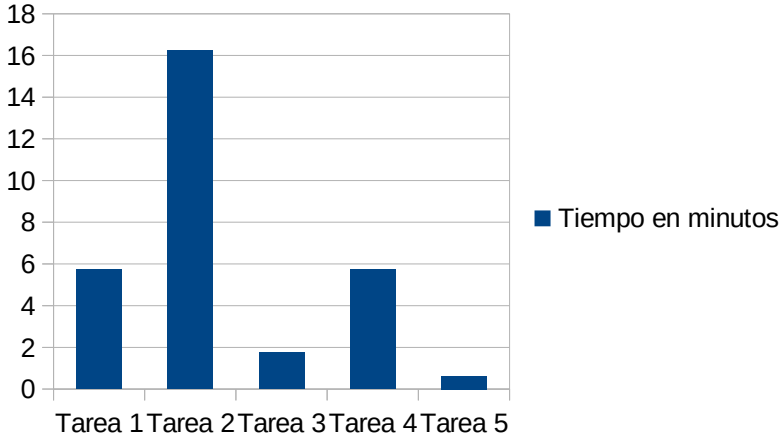


Figura 13: Ejemplo gráfica datos cuantitativos 2/2

2.4.6 Análisis de los datos cualitativos

En este punto detallaremos si los usuarios se han mostrado seguros o dubitativos al manejar la aplicación. Veremos también aspectos como si algún usuario se ha sentido frustrado o por el contrario a disfrutado del uso. En general, comprobaremos para cada tarea cuales han sido las “sensaciones” del usuario anotando los puntos positivos y negativos de la experiencia, anotándolos en cada caso, con el fin de refinar la aplicación final.

3. Diseño técnico de la aplicación

3.1 Definición de los casos de uso.

3.1.1 Diagrama UML.

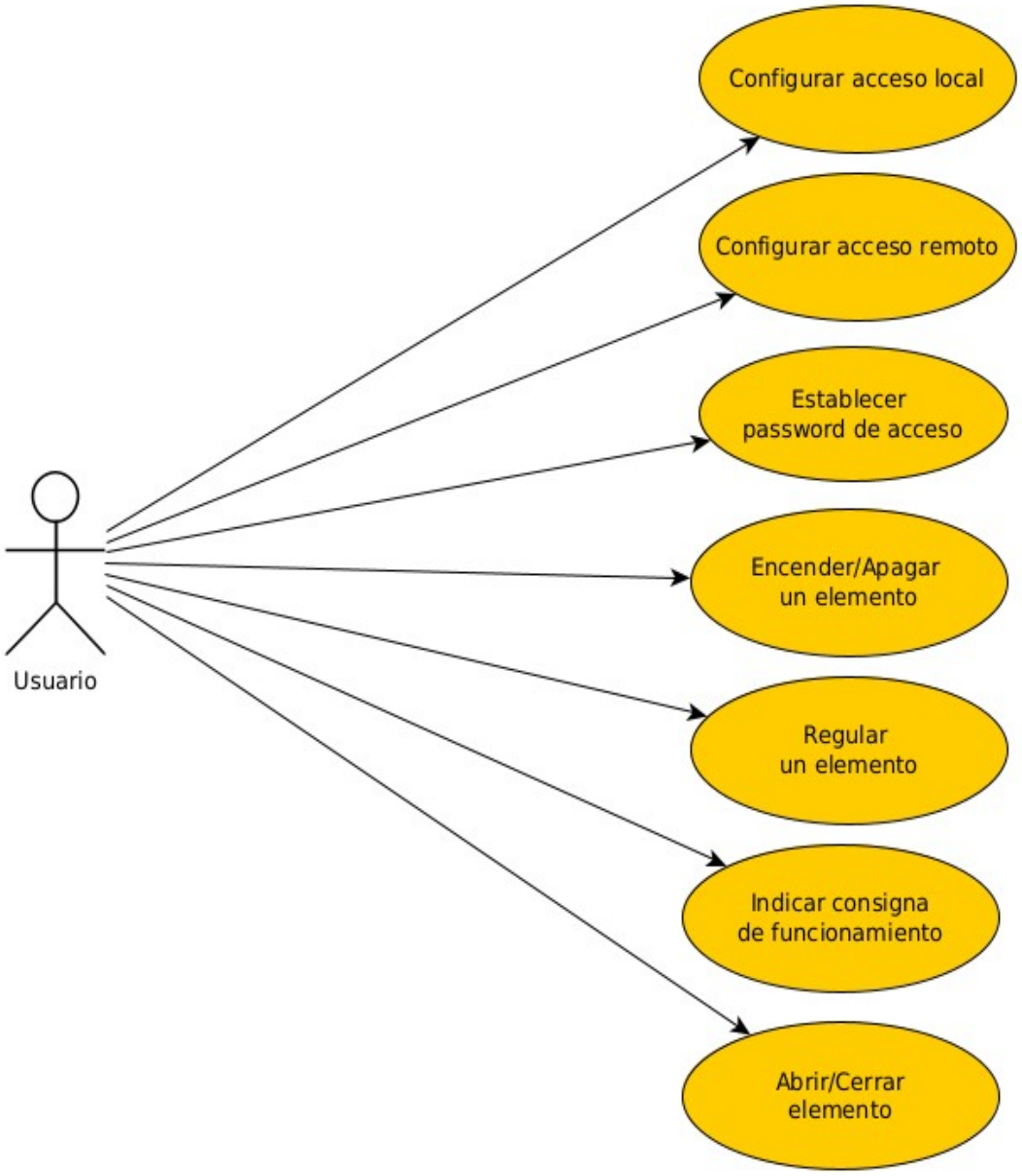


Figura 14: Casos de uso, diagrama UML

3.1.2 Casos de uso.

Definimos los diferentes casos de uso descritos en el UML anterior.

Caso de uso:	Configurar acceso local.
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere poder acceder al control local (wifi) de su vivienda.
Precondición:	El móvil del usuario debe estar conectado a la red wifi de la vivienda donde está conectado el router KNX/IP.
Garantías mínimas:	Se guardará la dirección IP en el registro de la aplicación.
Garantías en caso de éxito:	Se guardará la dirección IP y la conexión será validada.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema valida la contraseña de usuario 3. El usuario accede al menú “configuración”. 4. El usuario introduce la IP local del router KNX/IP. 5. El sistema registra y valida la IP.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 4A. El usuario introduce la IP en un formato incorrecto. 4A1. El sistema rechaza la IP y solicita introducirla de nuevo. 5A. El sistema no puede validar la conexión con el router KNX/IP. 5A1. El sistema registra el valor de la IP de todas maneras.	

Caso de uso:	Valida la contraseña de usuario
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere validar su acceso a la aplicación.
Precondición:	Debe haberse establecido una contraseña previamente.
Garantías mínimas:	Se obtendrá el resultado de la validación.
Garantías en caso de éxito:	Se permitirá el acceso al resto de opciones de la aplicación.
Escenario principal de éxito:	
1. El sistema solicita la contraseña de acceso 2. El usuario introduce la contraseña. 3. El sistema verifica que la contraseña es correcta. 4. El sistema permite el acceso a la aplicación.	
Extensiones:	
3A. El usuario introduce una contraseña incorrecta 3A1. El sistema rechaza la contraseña y solicita introducirla de nuevo.	

Caso de uso:	Configurar acceso remoto
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere poder acceder desde fuera de su vivienda a las acciones de control sobre la misma.
Precondición:	El usuario debe poder acceder a Internet y el router KNX/IP debe disponer de un puerto público accesible en el router de la vivienda.
Garantías mínimas:	Se guardará la dirección IP y puerto remoto en el registro de la aplicación.
Garantías en caso de éxito:	Se guardará la dirección IP y puerto remoto y la conexión será validada.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema valida la contraseña de usuario 3. El usuario accede al menú “configuración”. 4. El usuario introduce la IP remota y el puerto de acceso del router KNX/IP. 5. El sistema registra y valida la IP y el puerto.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 4A. El usuario introduce la IP en un formato incorrecto. 4A1. El sistema rechaza la IP y solicita introducirla de nuevo. 5A. El sistema no puede validar la conexión con el router KNX/IP. 5A1. El sistema registra el valor de la IP y el puerto de todas maneras.	

Caso de uso:	Establecer contraseña de acceso.
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere disponer de una medida de seguridad para que un tercero no pueda acceder a realizar acciones en su vivienda si el móvil del usuario cae en manos de un tercero no autorizado.
Precondición:	Si existe una contraseña previa establecida, debe conocerse previamente. Garantías mínimas: No se realizará ningún cambio en caso de error.
Garantías mínimas:	No se realizará ningún cambio en caso de error.
Garantías en caso de éxito:	Se guardará la nueva contraseña.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema **valida la contraseña de usuario.** 3. El usuario accede al menú “configuración”. 4. El usuario introduce la nueva contraseña por duplicado. 5. El sistema la guarda como nuevo contraseña del sistema.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 4A. El usuario introduce dos contraseñas que no coinciden. 4A1. El sistema rechaza la nueva contraseña y solicita volver a introducirla.	

Caso de uso:	Encender/Apagar un elemento
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere poder encender y/o apagar un dispositivo que soporte esta función como: un circuito de iluminación, un enchufe, una máquina de aire acondicionado, etc.
Precondición:	la aplicación debe disponer de conexión local o remota a la instalación.
Garantías mínimas:	Se accederá hasta el elemento a controlar y el estado del elemento no cambiará.
Garantías en caso de éxito:	Se accederá hasta el elemento a controlar y se modificará su estado actual.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema **valida la contraseña de usuario** 3. El usuario accede al menú “Acceso Local”. 4. El usuario accede a la sección y zona deseada de la vivienda. 5. El usuario selecciona el elemento que desea encender o apagar. 6. El usuario presiona “ON” para encender el elemento. 7. El sistema refresca el valor de estado del elemento después de realizar la acción.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 3A. El usuario accede a “Acceso remoto” si se encuentra fuera del alcance de la red local (Wifi) 3A1. El sistema conecta mediante acceso remoto. 6A. El usuario presiona “OFF” para apagar el elemento.	

Caso de uso:	Regular un elemento
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere poder regular el valor de funcionamiento de un dispositivo que soporte esta función como: un circuito de iluminación, una persiana o toldo, un ventilador, etc.
Precondición:	la aplicación debe disponer de conexión local o remota a la instalación.
Garantías mínimas:	Se accederá hasta el elemento a controlar y el estado del elemento no cambiará.
Garantías en caso de éxito:	Se accederá hasta el elemento a controlar y se modificará su estado actual en función de la regulación seleccionada.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema **valida la contraseña de usuario** 3. El usuario accede al menú “Acceso Local”. 4. El usuario accede a la sección y zona deseada de la vivienda. 5. El usuario selecciona el elemento que desea regular. 6. El usuario desplaza el selector de la barra de porcentaje hasta llevarlo al valor deseado. 7. El sistema refresca el valor de estado del elemento después de realizar la acción.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 3A. El usuario accede a “Acceso remoto” si se encuentra fuera del alcance de la red local (Wifi) 3A1. El sistema conecta mediante acceso remoto.	

Caso de uso:	Indicar consigna de funcionamiento
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere fijar el valor de consigna de funcionamiento de un dispositivo que soporte esta función como: una máquina de aire acondicionado, control de radiadores o una caldera.
Precondición:	la aplicación debe disponer de conexión local o remota a la instalación.
Garantías mínimas:	Se accederá hasta el elemento a controlar y el estado del elemento no cambiará.
Garantías en caso de éxito:	Se accederá hasta el elemento a controlar y se modificará el valor de consigna de funcionamiento.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema **valida la contraseña de usuario** 3. El usuario accede al menú “Acceso Local”. 4. El usuario accede a la sección y zona deseada de la vivienda. 5. El usuario selecciona el elemento en el que desea modificar su consigna de funcionamiento. 6. El usuario introducirá el nuevo valor de consigna. 7. El sistema refresca el valor de consigna del elemento después de realizar la acción.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 3A. El usuario accede a “Acceso remoto” si se encuentra fuera del alcance de la red local (Wifi) 3A1. El sistema conecta mediante acceso remoto. 5A. El valor de consigna introducido por el usuario está fuera del rango permitido. 5A1. El sistema solicita introducir un nuevo valor.	

Caso de uso:	Abrir/Cerrar un elemento
Actor principal:	usuario.
Ámbito:	Aplicación móvil.
Stakeholder e intereses:	El usuario quiere poder abrir y/o cerrar un dispositivo que soporte esta función como: una persiana, un toldo, una cortina, etc.
Precondición:	la aplicación debe disponer de conexión local o remota a la instalación.
Garantías mínimas:	Se accederá hasta el elemento a controlar y el estado del elemento no cambiará.
Garantías en caso de éxito:	Se accederá hasta el elemento a controlar y se modificará su estado actual según la última orden indicada.
Escenario principal de éxito:	
1. El usuario accede a la aplicación móvil. 2. El sistema v**alida la contraseña de usuario** 3. El usuario accede al menú “Acceso Local”. 4. El usuario accede a la sección y zona deseada de la vivienda. 5. El usuario selecciona el elemento que desea encender o apagar. 6. El usuario presiona “Abrir” para accionar el elemento. 7. El sistema refresca el valor de estado del elemento después de realizar la acción.	
Extensiones:	
2A. No hay contraseña establecida y se puede acceder libremente a la aplicación. 3A. El usuario accede a “Acceso remoto” si se encuentra fuera del alcance de la red local (Wifi) 3A1. El sistema conecta mediante acceso remoto. 6A. El usuario presiona “Cerrar” para accionar el elemento.	

3.2 Diseño de la arquitectura.

3.2.1 Diagrama UML correspondiente a base de datos.

La aplicación conforme se ha planteado no dispone de acceso a base de datos, en su lugar se utilizará un objeto «SharedPreferences» que guardará en un fichero local de la aplicación los valores de las direcciones IP, puertos y contraseña de acceso, así como, cualquier otro parámetro que pueda surgir.

3.2.2 Diagrama UML correspondiente al diseño de entidades y clases.

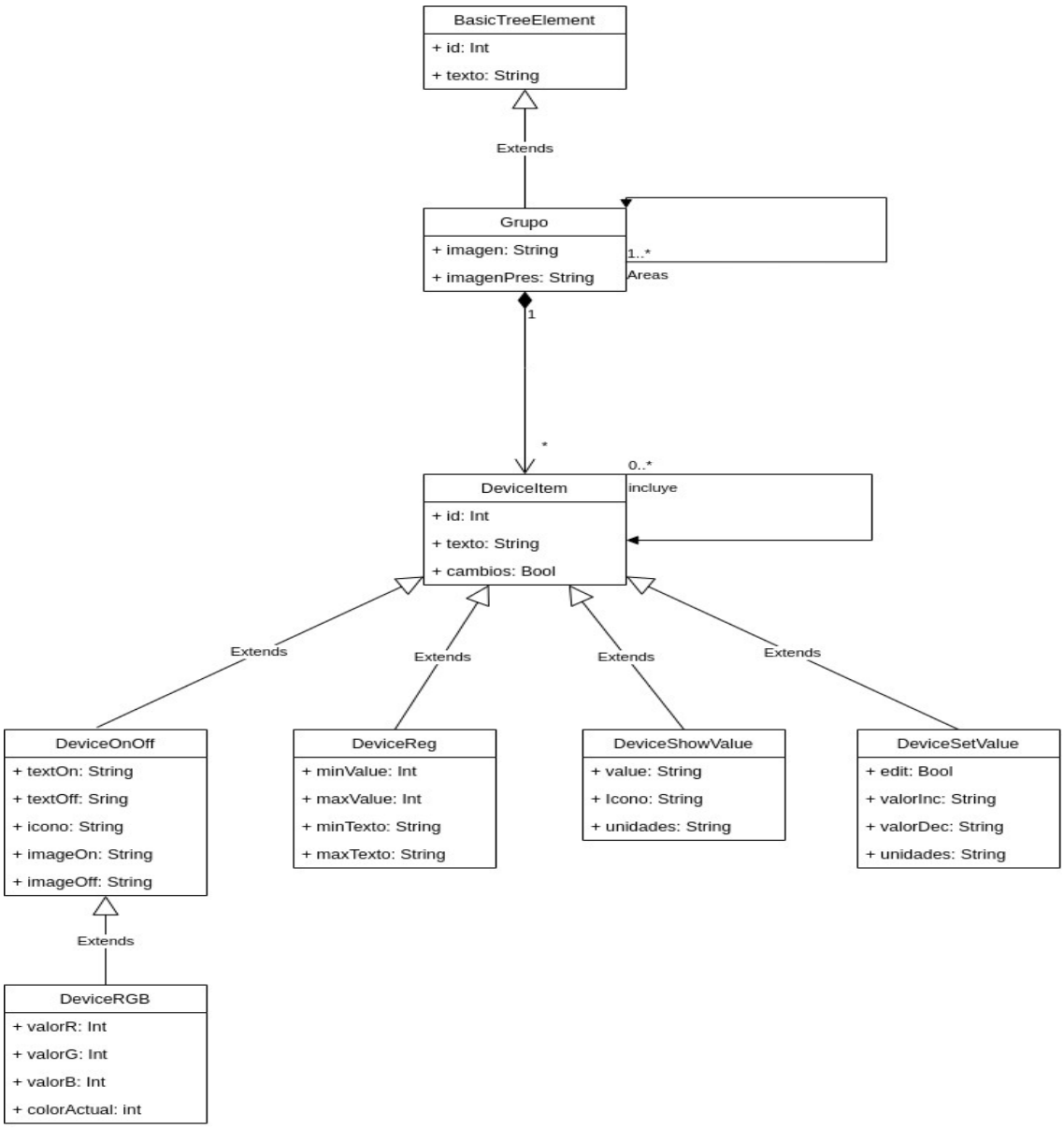


Figura 15: Diagrama UML correspondiente al diseño de entidades y clases inicial.

Las secciones/Areas del sistema las modelaremos en base a la clase “Grupo” que derivan de la base “BasicTreeElement”, se ha llevado a cabo esta

generalización por si durante el transcurso del desarrollo hubiera que añadir un nuevo tipo de agrupación distinto de lo expuesto en “Grupo”.

Por otro lado, se define una clase base con los campos que cualquier dispositivo debe implementar “DeviceItem” y luego tenemos una serie de clases que extienden a la primera y nos permiten modelar los diferentes tipos de objeto que vamos a necesitar, por ejemplo “DeviceReg” además de las propiedades de su clase base, define los valores mínimo y máximo sobre los que se puede regular, así como el texto que deberá aparecer a cada lado del elemento “slider”. En el caso de “DeviceOnOff” nos permitirá modelar dispositivos que cambien entre dos estados, podemos definir un texto para la acción de activación “ON” y otro para desactivación “OFF” (podríamos indicar Subir/Bajar también , por ejemplo), así como un icono representativo del objeto y dos imágenes que representen cuando se encuentra en estado “OFF” y estado “ON”. “DeviceRGB” permite modificar un objeto que pueda representar colores, el valor de cada componente del color queda reflejado en las propiedades del objeto. Por último “DeviceShowValue” permite visualizar un valor concreto (por ejemplo una temperatura o un estado) pudiendo marcar unidades de medida a aplicar y “DeviceSetValue” nos permite modificar un valor concreto aplicando pulsos incrementales o decrementales con una magnitud expresada en las propiedades del objeto.

3.2.3 Diagrama explicativo de la arquitectura del sistema.

La aplicación no será distribuida, por lo que el almacenamiento del sistema, la lógica de negocio y la presentación serán gestionadas todas ellas en una única aplicación. Esta aplicación utilizará el paradigma MVC (modelo-vista-controlador) para su desarrollo. Dispondremos de un componente en forma de librería (Calimero Project) al cual se le solicitarán servicios de comunicación con la red de domótica instalada en la vivienda. Podemos ver un esquema de los componentes a continuación.

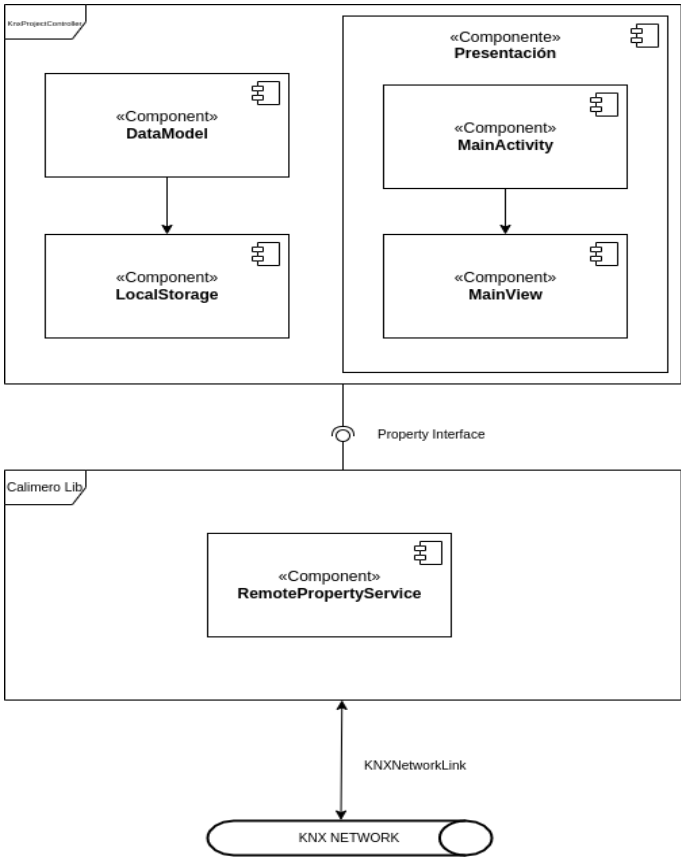


Figura 16: Diagrama arquitectura del sistema

4. Implementación y modificaciones sobre el diseño original.

4.1 Definición del fichero de configuración XML

Para llevar a cabo el diseño del fichero, tenemos que tener en cuenta como funciona una red KNX, conociendo el concepto de "dirección de grupo" así como los diferentes tipos de datos que se pueden aplicar a las mismas. Por ese motivo indicaremos primeramente dicha información.

4.1.1 Direcciones de grupo KNX.

El diseño de instalaciones de control KNX debe ser realizado por un profesional certificado mediante el software de desarrollo ETS. Una vez finalizado el proyecto, el desarrollador habrá asignado a cada dispositivo una funcionalidad y unas serie de direcciones de grupo. Las direcciones de grupo son una serie de puntos de intercambio de datos entre los dispositivos de una red KNX y están denotados por tres valores numéricos separados por el símbolo "/", dichos valores se configuran en la aplicación ETS por parte del desarrollador y suelen denotar una función del dispositivo en cuestión, como puede verse en la imagen siguiente:

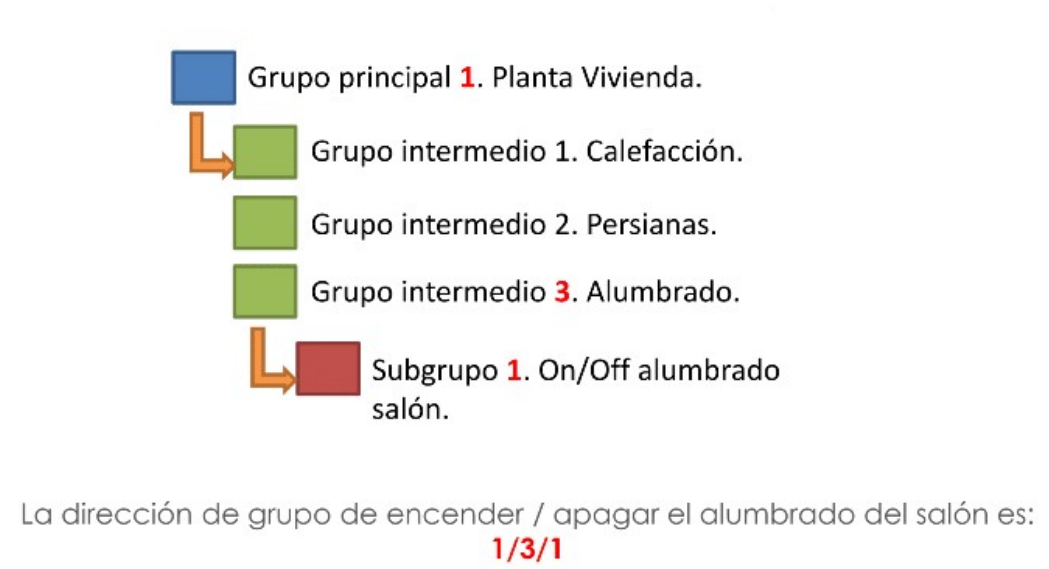


Figura 17: Direcciones de grupo KNX

Normalmente se definen más de una dirección de grupo para una función, por ejemplo, en el caso de la imagen superior la dirección de grupo "1/3/1" es un dirección de escritura, si escribimos un 1 en ella se encenderá la luz y si escribimos un 0 se apagará. Sin embargo, necesitaremos una segunda dirección en este caso de lectura para disponer del estado del alumbrado de salón. Por ejemplo podríamos definir "1/3/2" como dirección receptora del estado, de este modo, si algún otro dispositivo enciende la luz del salón podremos enterarnos mediante su dirección de lectura.

En función del tipo de dispositivo podremos tener la necesidad de disponer de varias direcciones de grupo de escritura y de lectura. Así por ejemplo, una máquina de aire acondicionado tendrá habitualmente tres direcciones de grupo de escritura (encendido/apagado, consigna de temperatura y velocidad de ventilador) y tendremos otras tantas de lectura con los estados correspondientes e incluso alguna más como puede ser el código de error de la máquina en caso de fallo técnico.

4.1.2 Tipos de datos KNX.

En la definición de dispositivos hemos de referenciar que direcciones de grupo se utilizarán en cada uno de ellos. Al igual que con las variables tradicionales, las direcciones de grupo de KNX disponen de tipo, según el valor que representan, sólo que en este caso, se trato de un valor numérico y no un texto de como “int” o “float”. A continuación indicamos los tipos más habituales que podemos definir y que deberemos definir en nuestro fichero de descripción de la instalación.

TIPOS BINARIOS	TIPOS ESCALARES	TIPOS CONTROL	TIPOS FLOTANTES
SWITCH="1.001" BOOL = "1.002" ENABLE ="1.003" RAMP = "1.004" ALARM = "1.005" BINARYVALUE ="1.006" STEP = "1.007" UPDOWN = "1.008" OPENCLOSE ="1.009" START = "1.010" STATE = "1.011" INVERT ="1.012" DIMSENDSTYLE ="1.013" INPUTSOURCE ="1.014" RESET = "1.015" ACK ="1.016" TRIGGER ="1.017" OCCUPANCY ="1.018" WINDOW_DOOR ="1.019" LOGICAL_FUNCTION ="1.021" SCENE_AB ="1.022" SHUTTER_BLINDS_MODE ="1.023"	SCALING = "5.001" ANGLE = "5.003" PERCENT_U8 ="5.004" DECIMALFACTOR ="5.005" VALUE_1_UCOUNT ="5.010"	CONTROL_DIMMING ="3.007" CONTROL_BLINDS ="3.008"	TEMPERATURE ="9.001" TEMPERATURE_DIFFERENCE ="9.002" TEMPERATURE_GRADIENT ="9.003" INTENSITY_OF_LIGHT ="9.004" WIND_SPEED ="9.005" AIR_PRESSURE ="9.006" HUMIDITY ="9.007" AIRQUALITY ="9.008" TIME_DIFFERENCE1 ="9.010" TIME_DIFFERENCE2 ="9.011" VOLTAGE ="9.020" ELECTRICAL_CURRENT ="9.021" POWERDENSITY ="9.022" KELVIN_PER_PERCENT ="9.023" POWER ="9.024"

Figura 18: Tipos KNX implementados

4.1.3 Definición de estructura de nodos en el fichero.

Por lo tanto, nuestro fichero de definición de la instalación tendrá en cuenta los datos sobre KNX aportados en los apartados anteriores.

Adicionalmente, y después de revisar la estructura tipo de la mayoría de instalaciones hemos llegado a la conclusión de que la estructura óptima a nivel organizativa sería la siguiente:

Dispondremos de un conjunto de “secciones” que representarán zonas de gran tamaño de nuestra instalación, habitualmente, serán plantas de un edificio o

vivienda o separación de áreas físicas por funcionalidad, por ejemplo, en una nave industrial podríamos tener dos secciones, oficinas y almacén.
Dentro de cada sección dispondremos de “zonas” para afinar más la ubicación de los elementos, así pues, en una vivienda podríamos decir que nuestra sección “Planta Baja” dispone de las zonas “Entrada” y “Garaje”.
Por último, dentro de una zona distribuiremos los diferentes tipos de dispositivo en “Grupos” que serán habitualmente por funcionalidad, por tanto, podemos tener un grupo que sea para “Luces” otro grupo para “Sensores” otro para “Persianas” y así, para cualquier funcionalidad que pensemos. Dentro de estos grupos es donde definiremos los tipos de dispositivo que vamos a gestionar con sus correspondientes direcciones de grupo.

A continuación, podemos ver un diagrama explicativo de ejemplo de como queda la distribución de elementos.

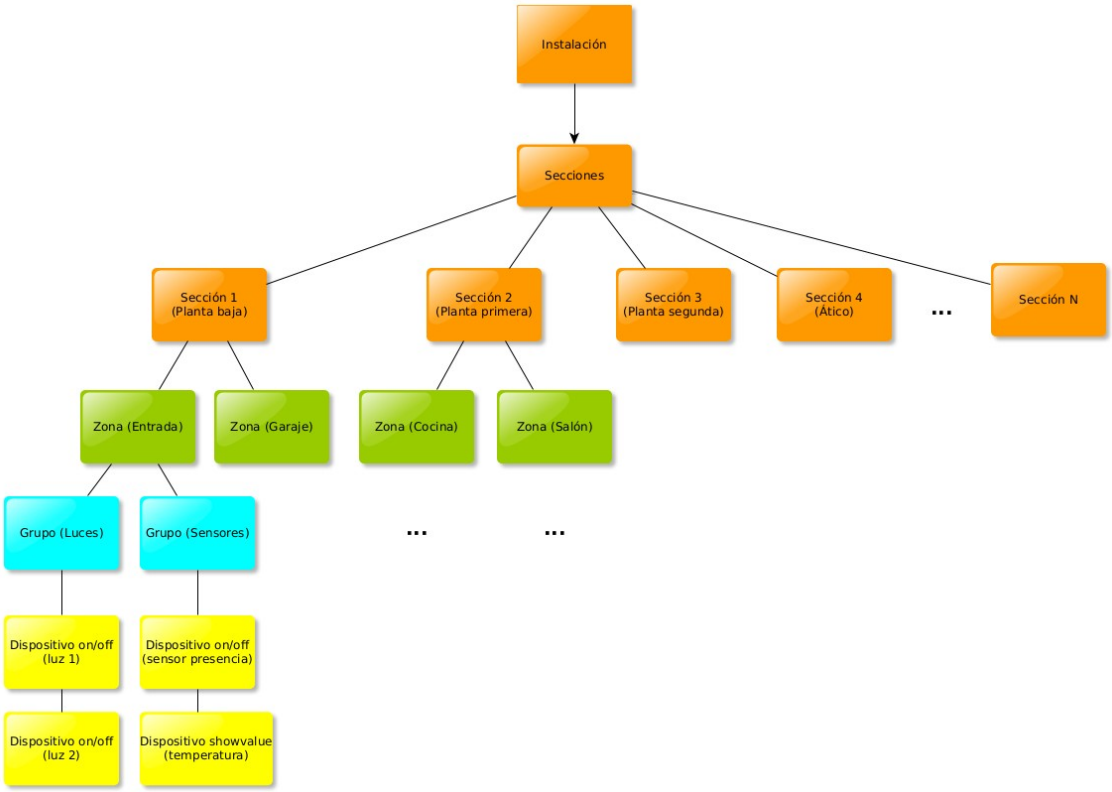


Figura 19: Diagrama explicativo distribución de elementos

Por tanto, nuestro fichero deberá definir todos los elementos de estructura de la instalación (secciones, zonas y grupos) y adicionalmente los distintos tipos de dispositivo que podemos controlar.
Para definir estos dispositivos se ha tenido en cuenta que un mismo tipo de dispositivo puede servir para realizar múltiples acciones en el mundo real. Por ejemplo, una luz requiere una orden de encendido (On/Off), pero una lavadora también, una caldera o un radiador, por tanto, con un mismo objeto “device_onoff” podremos aplicarlo a diferentes tipos de dispositivo en el mundo real.

Como elementos comunes a todos los objetos tendremos que todos ellos disponen de un identificador numérico (id), del mismo modo un campo (texto) para mostrar el nombre del objeto en pantalla. A continuación veremos cada uno de los tipos que podemos definir y los campos que se han previsto para cada uno de ellos.

4.1.4 Tipos de nodos estructurales

- “secciones”, se trata del nodo inicial donde tendremos a su vez tres propiedades:
  - “fondo”, nombre del fichero gráfico para utilizar como fondo de pantalla, en caso de dejar en blanco se aplica uno por defecto.
  - “showlogo”, en caso de valer “1” se muestra el logo de la aplicación en la pantalla principal, nada en caso de valer “0”.
  - “showcredits”,en caso de valer “1” se muestran los créditos de desarrollador en la pantalla principal, nada en caso de valer “0”.
  - Ejemplo:
    - <secciones fondo="" showlogo="1" showcredits="1"> </secciones>
- “seccion”, este nodo forma parte del nodo “secciones” y define un área de control.
  - “id”, valor numérico único para identificar el elemento.
  - “tipo”, campo que en un futuro nos permitirá diferencias entre tipos de secciones (por ejemplo si quisiéramos hacer pantallas para ver cámaras IP, podría ser el tipo 2). De momento, este campo no aporta funcionalidad.
  - “texto”, es el nombre de la sección.
  - Ejemplo:
    - <seccion id="1" tipo="1" texto="Mi casa"> </seccion>
- “zona”, este nodo hace referencia a un área más reducida dentro de una sección, sus propiedades son:
  - “id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre de la zona.
  - “imagen”, nombre del fichero gráfico que ilustrará el icono de la zona en pantalla.
  - Ejemplo:
    - <zona id="6" texto="Entrada" imagen="puerta.png"> </zona>
- “grupo”, este nodo contiene a otros nodos de tipo dispositivo, normalmente, los asignaremos por su función, pero podríamos hacerlo según el criterio que se prefiera.
  - “id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre del grupo.
  - “imagen”, nombre del fichero gráfico que ilustrará al grupo en pantalla.
  - “tipocom”, este campo es para futuras ampliaciones, actualmente se ha definido tipocom como “1” (Knx), pero es posible ampliar la aplicación en un futuro y que soporte otros protocolos como Bacnet IP o Modbus TCP, es ese caso podríamos utilizar otros valores para indicar distintos tipos de protocolo que conviven en una misma instalación.

- Ejemplo:
  - <grupo id="19" texto="Luces" imagen="luzmenu" tipocom="1"></grupo>

4.1.5 Tipos de nodos para dispositivos

A continuación, definiremos los diferentes tipos de dispositivo genéricos que se han tenido en cuenta para poder gestionar la mayor cantidad de dispositivos reales que pueden existir.

Para definir las direcciones de grupo dentro de otros objetos usaremos los siguientes nodos xml:

- “direccion_escritura”, indica una dirección de grupo sobre la que escribiremos valores (ordenes).
  - “texto”, es una propiedad que vincula la dirección a una acción del dispositivo contenedor (por ejemplo, texto=”accion_onoff”, indicaría que esta dirección se use para dar la orden de encendido/apagado en un objeto “device_onoff”).
  - “direccion”, el valor de la dirección de grupo sobre la que escribiremos el valor (por ejemplo 1/1/91).
  - “tipo”, el tipo de datos a escribir en formato KNX (por ejemplo, si se trata de un booleano sería 1.002).
  - Ejemplo:
    - <direccion_escritura texto="accion_onoff" direccion="1/1/91" tipo="1.001"/>
- “direccion_lectura”, indica una dirección de grupo sobre la que leeremos valores (estados).
  - “texto”, es una propiedad que vincula la dirección a una acción del dispositivo contenedor (por ejemplo, texto=”estado_onoff”, indicaría que esta dirección se use para recoger el estado de encendido/apagado en un objeto “device_onoff”).
  - “direccion”, el valor de la dirección de grupo sobre la que leeremos el valor (por ejemplo 1/1/92).
  - “tipo”, el tipo de datos a leer en formato KNX (por ejemplo, si se trata de un booleano sería 1.002).
  - Ejemplo:
    - <direccion_lectura texto="estado_onoff" direccion="1/1/4" tipo="1.001"/>

A continuación podemos definir los distintos tipos de objeto con los que trabajaremos:

- “device_onoff”, dispositivo para poder encender/apagar un elemento mediante una orden simple, dispone de las siguientes propiedades:
  - “id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre del dispositivo.
  - “icono”, nombre del fichero gráfico que representa al dispositivo.
  - “imagen_on”, nombre del fichero gráfico que representa al dispositivo cuando está encendido.

- “imagen_off”, nombre del fichero gráfico que representa al dispositivo cuando está apagado.
- “texto_on”, texto a representar cuando el objeto está encendido.
- “texto_off”, texto a representar cuando el objeto está apagado.
- “tipocom”, protocolo de comunicaciones a usar, para el presente proyecto será siempre 1(KNX) ya que, la implementación de otros protocolos está fuera del alcance del presente proyecto.
- Dentro del dispositivo se incluyen dos objetos de definición de direcciones de grupo, unos para escritura y otro para lectura.
- Ejemplo:

```
<device_onoff id="59" texto="Luz entrada vivienda" icono="light_on" imagen_on="light_on.png" imagen_off="light_off.png" texto_on="Estado: ON" texto_off="Estado: OFF" tipocom="1">
        <direccion_escritura texto="accion_onoff" direccion="1/1/91" tipo="1.001"/>
        <direccion_lectura texto="estado_onoff" direccion="1/1/4" tipo="1.001"/>
</device_onoff>
```

- “device_setvalue”, este objeto nos permite mandar un valor fijo, siempre el mismo, a una dirección de grupo (por ejemplo ejecutar una escena mediante un valor binario como encender todas las luces de una estancia, siempre conllevará enviar un 1 a una dirección de grupo).
  - “id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre del dispositivo.
  - “icono”, nombre del fichero gráfico que representa al dispositivo.
  - “valor”, valor numérico que se enviará cada vez que se pulse sobre el elemento.
  - “tipocom”, protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
  - Dentro del dispositivo se incluye un objeto de definición de dirección de grupo de escritura.
  - Ejemplo:

```
<device_setvalue id="73" texto="Luces salón ON" icono="light_on.png" valor="1" tipocom="1">
        <direccion_escritura texto="accion_valor" direccion="10/1/5" tipo="1.001"/>
</device_setvalue>
```

- "device_showvalue”, objeto diseñado para mostrar un valor, por ejemplo para mostrar el estado de un detector, un sensor de temperatura o humedad o cualquier elemento que reporte información de tipo numérico. El objeto permite un modo de edición con el cual podemos modificar un valor de consigna, para conseguir el comportamiento similar al de un termostato.
  - “id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre del dispositivo.
  - “edit”, indicando valor “1” validamos el objeto para funcionar en modo “termostato” con botones de incrementar valor y decrementar valor. Con valor “0” sólo mostramos el valor de lectura, pero sin posibilidad de interactuar.
  - “incremento”, el valor numérico que aumentaremos la consigna en caso de que el modo edición esté activo.
  - “decremento”, el valor numérico que reduciremos la consigna en caso de que el modo edición esté activo.

- “unidad”, tipo de magnitud que estamos mostrando, (“ºC” si mostramos temperatura, por ejemplo)
- “tipocom”, protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
- Dentro del dispositivo podemos incluir hasta dos objetos de definición de direcciones de grupo, unos para escritura y otro para lectura.
- Ejemplo:

```
<device_showvalue id="305" texto="Temp.consigna" tipocom="1" edit="1" incremento="1" decremento="1" unidad=" ºC" >
    <direccion_lectura texto="estado" direccion="2/2/12" tipo="9.001"/>
    <direccion_escritura texto="accion" direccion="2/2/12" tipo="9.001"/>
</device_showvalue>
```

- “device_reg”, el dispositivo de regulación es similar al On/Off en la parte en que tiene que encender un equipo para luego poder regularlo, y es en este punto donde difiere del anterior, ya que, proporciona el mecanismo para dar proporcionar un valor discreto dentro de un conjunto de posibles valores de regulación establecidos por un mínimo y un máximo.
  - id”, valor numérico único para identificar el elemento.
  - “texto”, es el nombre del dispositivo.
  - “icono”, nombre del fichero gráfico que representa al dispositivo.
  - “imagen_on”, nombre del fichero gráfico que representa al dispositivo cuando está encendido.
  - “imagen_off”, nombre del fichero gráfico que representa al dispositivo cuando está apagado.
  - “texto_on”, texto a representar cuando el objeto está encendido.
  - “texto_off”, texto a representar cuando el objeto está apagado.
  - “min”, valor numérico que representa al limite inferior de los valores que podemos adoptar.
  - “mintexto”, valor a representar en la barra de regulación junto al valor mínimo.
  - “max”, valor numérico que representa al limite superior de los valores que podemos adoptar.
  - ““maxtexto”, valor a representar en la barra de regulación junto al valor máximo.
  - “tipocom”, protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
  - Dentro del dispositivo se incluyen cuatro objetos de definición de direcciones de grupo, dos de escritura (orden de on/off y orden de regulación) y dos de lectura (estado on/off y estado de regulación).
  - Ejemplo:

```
<device_reg id="67" texto="Luz regulada" icono="dimmer.png" imagen_on="light_on.png" texto_on="ON" imagen_off="light_off.png" texto_off="OFF" tipocom="1" min="0" mintexto="Apagado" max="255" maxtexto="Máximo" >
        <direccion_escritura texto="accion_onoff" direccion="1/2/1" tipo="1.001"/>
        <direccion_lectura texto="estado_onoff" direccion="1/2/4" tipo="1.001"/>
        <direccion_escritura texto="accion_setvalor" direccion="1/2/3" tipo="5.001"/>
        <direccion_lectura texto="estado_valor" direccion="1/2/5" tipo="5.001"/>
</device_reg>
```

- “device_rgb”, dispositivo creado para gestionar sistemas de iluminación basados en cambio de color por componentes RGB.

- id", valor numérico único para identificar el elemento.
- "texto", es el nombre del dispositivo.
- "icono", nombre del fichero gráfico que representa al dispositivo.
- "imagen_on", nombre del fichero gráfico que representa al dispositivo cuando está encendido.
- "imagen_off", nombre del fichero gráfico que representa al dispositivo cuando está apagado.
- "texto_on", texto a representar cuando el objeto está encendido.
- "texto_off", texto a representar cuando el objeto está apagado.
- "tipocom", protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
- Dentro del dispositivo se incluyen siete objetos de definición de direcciones de grupo, cuatro de escritura (orden de on/off y orden de regulación para cada componente, R,G y B) y tres de lectura (estado de regulación de los componentes R,G y B).
- Ejemplo:
  - <device_rgb id="35" texto="RGB TV" icono="rgb.png" imagen_on="rgb.png" imagen_off="rgb.png" texto_on="ON" texto_off="OFF" tipocom="1">
    <direccion_escritura texto="accion_onoff" direccion="15/7/10" tipo="1.001"/>
    <direccion_escritura texto="accion_setvalor_r" direccion="15/1/3" tipo="5.001"/>
    <direccion_escritura texto="accion_setvalor_g" direccion="15/2/3" tipo="5.001"/>
    <direccion_escritura texto="accion_setvalor_b" direccion="15/3/3" tipo="5.001"/>
    <direccion_lectura texto="estado_valor_r" direccion="15/1/5" tipo="5.001"/>
    <direccion_lectura texto="estado_valor_g" direccion="15/2/5" tipo="5.001"/>
    <direccion_lectura texto="estado_valor_b" direccion="15/3/5" tipo="5.001"/>
    </device_rgb>

- "device_slide", este dispositivo surge de la necesidad de representar distintos estados de un elemento mediante un icono y a la vez permitir transicionar entre los distintos estados (slides), un ejemplo sería cambiar el modo de funcionamiento de un aire acondicionado, dicho modo podría ser (simplificando) frío o calor y al igual que hacemos con el mando a distancia pulsando sobre el elemento queremos cambiar entre uno u otro modo.
  - id", valor numérico único para identificar el elemento.
  - "texto", es el nombre del dispositivo.
  - "tipocom", protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
  - Ejemplo:
    - <device_slide id="302" texto="Slide Modo" tipocom="1">
      <slide id="1" texto="Modo Calor" tipocom="1" condicion="==" valor="1" icono="heat.png" valor_escritura="4">
      <direccion_lectura texto="estado" direccion="2/2/19" tipo="5.001"/>
      <direccion_escritura texto="accion" direccion="2/2/18" tipo="5.001"/>
      </slide>
      <slide id="2" texto="Modo Frio" tipocom="1" condicion="==" valor="4" icono="cold.png" valor_escritura="1">
      <direccion_lectura texto="estado" direccion="2/2/19" tipo="5.001"/>
      <direccion_escritura texto="accion" direccion="2/2/18" tipo="5.001"/>
      </slide>
      </device_slide>

- "slide", este objeto es una componente del anterior que representa uno de los estados sobre los que se puede transicionar.

- id”, valor numérico único para identificar el elemento.
- “texto”, es el nombre del dispositivo.
- “icono”, nombre de la imagen a mostrar cuando el “slide” está activo.
- “condicion”, representa la comparación que se hará con el atributo valor para determinar si este es el “slide” activo dentro de un device_slide. Las condiciones a aplicar pueden ser ">" , "<" , "==" , "!=" , ">=" , "<=".
- “valor”, es el valor que se usará junto con la condición para validar el estado de activo del “slide”.
- “valor_escritura”, cuando pulsamos sobre el “slide” lanzaremos el valor numérico indicado en este campo como valor escritura al dispositivo.
- “tipocom”, protocolo de comunicaciones a usar (en el presente proyecto será siempre 1 representando KNX).
- Dentro del dispositivo podemos incluir hasta dos objetos de definición de direcciones de grupo, unos para escritura de orden y otro para lectura del estado actual.
- Ejemplo:
  - <device_slide id="302" texto="Slide Modo" tipocom="1">
    <slide id="1" texto="Modo Calor" tipocom="1" condicion="==" valor="1" icono="heat.png" valor_escritura="4">
    <direccion_lectura texto="estado" direccion="2/2/19" tipo="5.001"/>
    <direccion_escritura texto="accion" direccion="2/2/18" tipo="5.001"/>
    </slide>
    <slide id="2" texto="Modo Frio" tipocom="1" condicion="==" valor="4" icono="cold.png" valor_escritura="1">
    <direccion_lectura texto="estado" direccion="2/2/19" tipo="5.001"/>
    <direccion_escritura texto="accion" direccion="2/2/18" tipo="5.001"/>
    </slide>
    </device_slide>

Con todo este conjunto de objetos podemos representar una amplia gama de dispositivos en el mundo real, ya sea como elementos simples o mediante la combinación de varios de ellos.

4.2 Definición del entorno gráfico.

El diseño del entorno gráfico se ha llevado a cabo respetando el prototipo existente, pero mejorando la integración de los distintos elementos de pantalla en los ficheros xml de definición de las mismas. De igual manera, se ha diseñado un conjunto de iconos para utilizar en la aplicación.

Conjunto 1/2

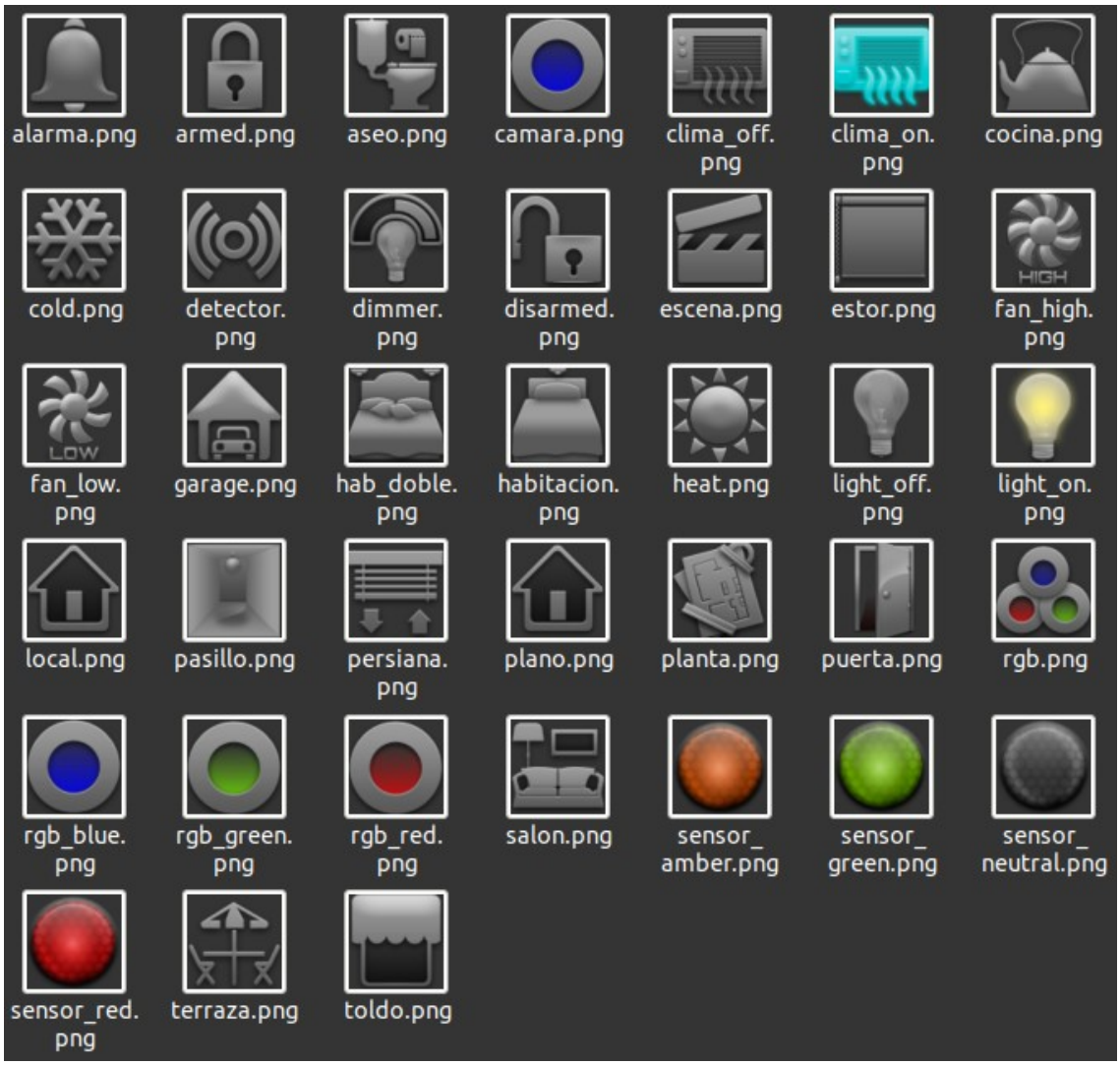


Figura 20: Conjunto elementos gráficos 1/2

conjunto 2/2

Figura 21: Conjunto elementos gráficos 2/2

Para el presente proyecto los iconos estarán en la carpeta “assets” dentro del empaquetado y no serán sustituibles, como una de las ampliaciones a futuro quedaría la posibilidad de crear un sistema de carga de conjuntos de iconos que permita modificar todo el aspecto gráfico de la aplicación. Para el presente proyecto, los elementos gráficos no serán modificables.

4.3 Selección de plataforma de desarrollo.

La plataforma de desarrollo seleccionada ha sido “Android Studio” de Google bajo plataforma Linux. Esto ha sido así por:

- Es el entorno de desarrollo oficial para Android ofreciendo mejor integración con el emulador y el SDK de desarrollo.
- Dispone de un editor gráfico para los ficheros XML de representación gráfica muy avanzado.
- Su funcionamiento bajo Linux que es la plataforma Host que utilizo habitualmente ofrece un buen rendimiento.

Como lenguaje de desarrollo se ha optado por Java en lugar de Kotlin, si bien el segundo es más moderno en su concepción, dispongo de mayor experiencia con Java y eso ha facilitado el desarrollo de la aplicación, aunque es posible migrar la aplicación a Kotlin una vez finalizada gracias a un asistente de conversión (pero requeriría de algún que otro ajuste posterior).

Como plataforma mínima para el compilado, hemos seleccionado el SDK de nivel 24 que se corresponde con Android 7.0. Esto es así, porque esa versión todavía incluye un porcentaje muy elevado de dispositivos aún ha día de hoy y además incorpora todos los elementos del sistema que podemos necesitar para el desarrollo de la aplicación, siendo igualmente compatible con las versiones posteriores de Android, por lo que abarcamos un campo mayor de dispositivos compatibles.

En cuanto a API de terceros, utilizaremos Calimero[5] para el acceso a bajo nivel a las comunicaciones de la red KNX. Se incorpora en forma de fichero jar a la solución.

4.4 Modificaciones sobre el diseño de origen.

Como hemos visto en la sección del fichero de definición de la solución, se han añadido algunos nuevos tipos de dispositivo respecto de los que partíamos en el diseño original, además se han adaptado y creado clases necesarias para la correcta implementación en entorno Android.

El nuevo diagrama de clases queda como puede verse a continuación.

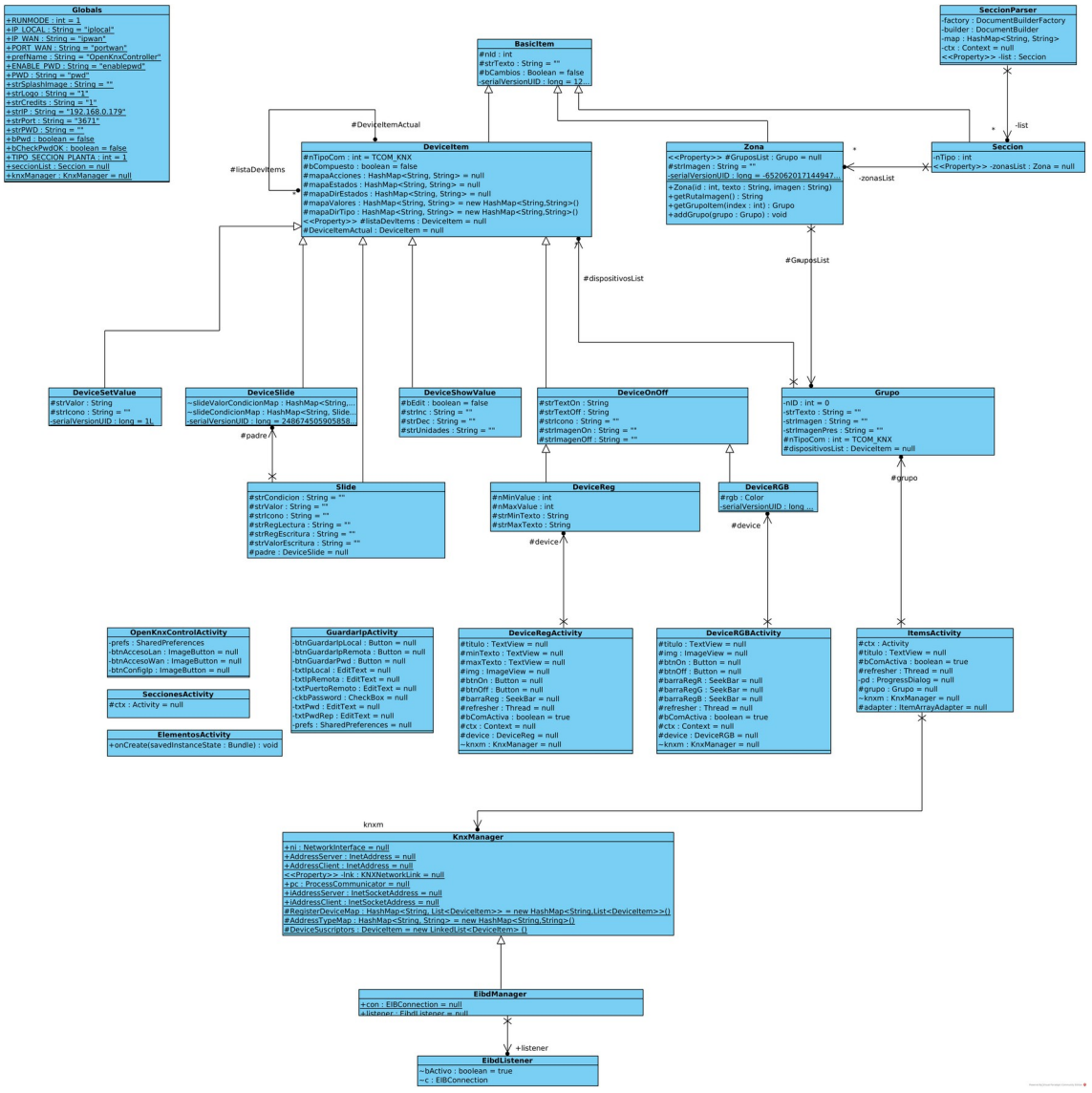


Figura 22: Diagrama de clases UML final

En esta arquitectura podemos ver que disponemos de un objeto “BasicItem” del que derivan prácticamente todas nuestras clases, por un lado aquellas que dan forma a la estructura de ordenación de dispositivos como son “Seccion” y “Zona” y por otro lado tenemos “DeviceItem” que es la clase de la que derivarán todo el resto de clases de dispositivos que hemos visto en la sección 4.1.

Como clases a tener en cuenta además de las comentadas, tenemos “SeccionParser”, esta clase que actua a modo de “Singleton” es la encargada de cargar nuestro esquema de información del fichero xml, generando tanto secciones, zonas, grupos como sus dispositivos contenidos. En resumen, genera toda la estructura de objetos que utilizaremos a partir del fichero xml.

La clase estática “Globals”, contiene los parámetros generales de funcionamiento de la aplicación así como la referencia al conjunto de secciones procesada por la clase “SeccionParser” y la referencia al KnxManager para gestionar las comunicaciones con la red KNX. Esta clase puede ser consultada desde cualquier objeto presente en la aplicación.

Disponemos de una serie de clases que derivan de “Activity” y que junto con sus respectivos ficheros de presentación xml, permiten visualizar los diferentes tipos de pantalla que tendremos disponibles, las resumimos a continuación:

- “OpenKnxControlActivity”, Actividad de arranque de la aplicación que incluye los botones del menú general.
- “GuardarActivity”, Actividad correspondiente con la pantalla de configuración donde podemos establecer las distintas Ips de acceso , así como las credenciales de seguridad.
- “SeccionesActivity”, Actividad para representar/gestionar las diferentes secciones definidas en nuestro fichero xml.
- “ZonasActivity”, Actividad para representar/gestionar las zonas correspondientes a la sección que se haya seleccionado.
- “ElementosActivity”, Actividad para representar/gestionar los dispositivos presentes en una zona, organizados en grupos.
- “ItemsActivity”, Actividad para representar/gestionar un dispositivo concreto seleccionado de la actividad de elementos.
- “DeviceRegActivity”, Actividad para representar/gestionar un dispositivo de regulación.
- “DeviceRGB”, Actividad para representar/gestionar un dispositivo de control de iluminación RGB.

Adicionalmente, para las comunicaciones, utilizaremos la clase “EibdManager” que deriva de “KnxManager” que permite establecer la comunicación IP con una red KNX. “EibdManager” incorpora una referencia a la clase “EibdListener” que nos permitirá refrescar los estados que se vayan actualizando dentro de nuestra red de control, notificándolos a la actividad “ItemsActivity” que actualizará los valores en pantalla para cada elemento que se esté visualizando.

5. Pruebas.

5.1 Definición de las pruebas.

Las pruebas se realizarán tanto en simulador como en dispositivo físico, mediante las primeras se verificará la correcta comunicación con la red de control KNX y se llevarán a cabo las pruebas unitarias necesarias. Estas son las pruebas esenciales del sistema, ya que, la mayor parte de la funcionalidad se basa en el correcto envío de valores a la red de control, la lectura del "feedback" y una correcta interpretación de dichos valores.

Mediante el dispositivo físico se realizarán las pruebas "empíricas" , en donde comprobaremos que las acciones en nuestra aplicación tienen resultado en los dispositivos en el mundo real.

Dada la simplicidad del entorno gráfico (repetimos en pantalla listas que contienen los mismo tipos de objeto continuamente) no se ha previsto ninguna prueba automatizada del "user Interface" más allá del chequeo manual del mismo.

5.1.1 Pruebas unitarias

Llevaremos a cabo pruebas unitarias de funcionamiento de:

- Las acciones de chequeo y verificación de valores de la pantalla de configuración.
- Las acciones que pueden realizar los distintos dispositivos soportados en el sistema.

Hay que tener en cuenta que para el segundo caso, será necesario estar conectados a una red KNX o una simulación de la misma, para poder verificar que los valores se escriben y leen correctamente. En el presente caso se ha utilizado tanto una red real de dispositivos (correspondiente a las oficinas donde trabajo), así como, una simulación corriendo en Linux usando el driver EIBD de la universidad técnica de Viéna [9]. Dicho driver permite simular una red y recoger las tramas que se escriben o leen de la misma. En todos los casos la pasarela KNX estaba configurada en la IP local: 192.168.0.12.

A continuación se indica cada test unitario realizado y su resultado.

GuardarIpActivityTest
guardarActivity_CorrectPassword_ReturnsTrue() Comprobamos la validez de una contraseña que se ha introducido correctamente dos veces.
guardarActivity_WrongPassword_ReturnsFalse() Comprobamos la no validez de una contraseña que se ha introducido incorrectamente en una de las ocasiones.
guardarActivity_ValidIps_ReturnsTrue() Verificamos para un conjunto de Ips correctamente formadas que son todas ellas validadas como correctas
guardarActivity_ValidIps_ReturnsFalse() Verificamos para un conjunto de Ips mal formadas que ninguna es dada por válida.

Figura 23: Test unitario GuardarIpActivityTest

DeviceOnOffTest
CheckAccionEstado() Este test escribe primero en el bus un valor “1”, después verifica que el valor es correcto. Después escribe un “0”, simulando que se ha pulsado sobre el mismo elemento (forzando un comportamiento de “toggle”) y verifica que dicho valor es 0
Valores mostrados por consola de Bus


Figura 24: Test unitario DeviceOnOffTest

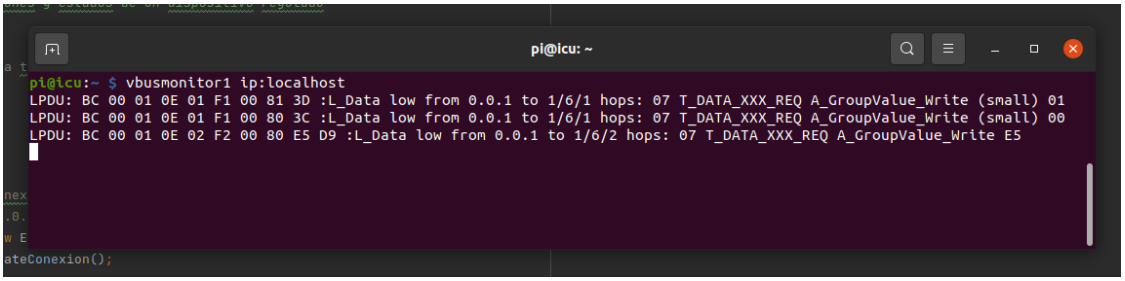
DeviceRegTest
CheckAccionEstado() Este test escribe primero en el bus un valor “1”, después verifica que el valor es correcto. Después escribe un “0”, simulando que se ha pulsado sobre el mismo elemento (forzando un comportamiento de “toggle”) y verifica que dicho valor es 0 Acto seguido mandamos un valor de regulación al bus aleatorio (como si el usuario regulase usando un “slider” y verificamos que dicho valor es correcto.
Valores mostrados por consola de Bus


Figura 25: Test unitario DeviceRegTest

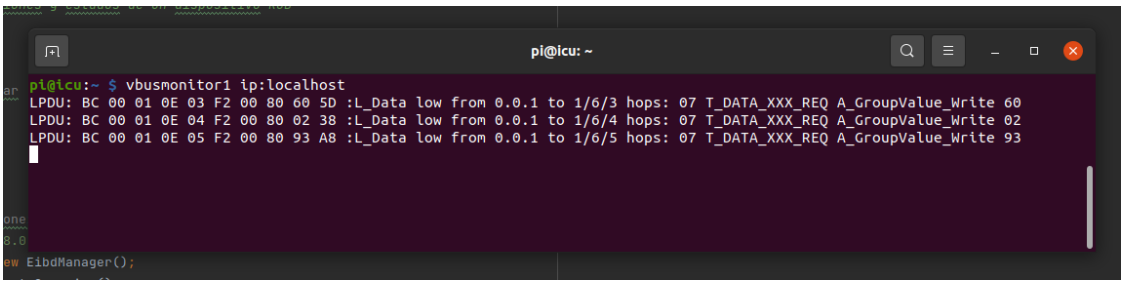
DeviceRGBTest
CheckAccionEstado() Este test escribe un valor aleatorio en el bus (simulando la selección del usuario en pantalla) y después comprueba que el valor lanzado al bus se corresponde con la orden, para los tres componentes (RGB).
Valores mostrados por consola de Bus


Figura 26: Test unitario DeviceRGBTest

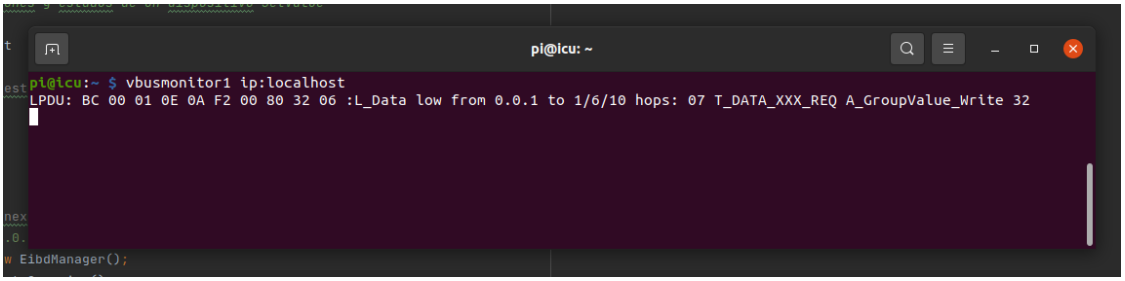
DeviceSetValueTest
CheckAccionEstado() Este test escribe el valor “50” en el bus como un valor fijo que se utiliza por parte de este objeto.
Valores mostrados por consola de Bus


Figura 27: Test Unitario DeviceSetValueTest

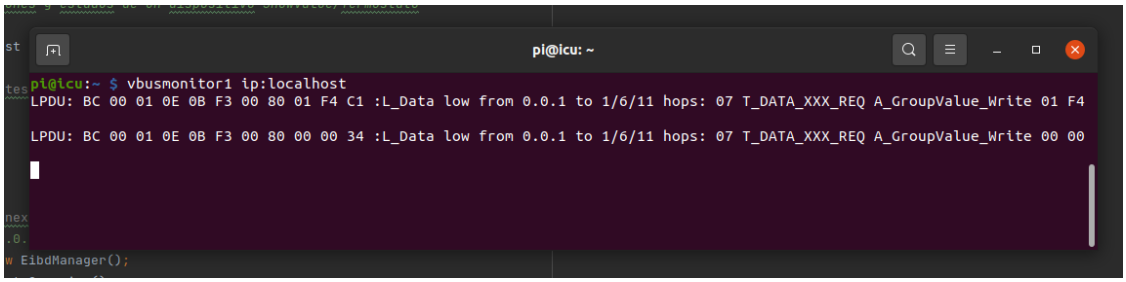
DeviceShowValueTest
CheckAccionEstado() Este test verifica el funcionamiento de tipo “termostato”, incrementando un valor un valor especificado en el objeto (5) y posteriormente decrementarlo. En ambos casos se verifica que el valor resultante en el bus coincide con los valores enviados.
Valores mostrados por consola de Bus


Figura 28: Test unitario DeviceShowValueTest

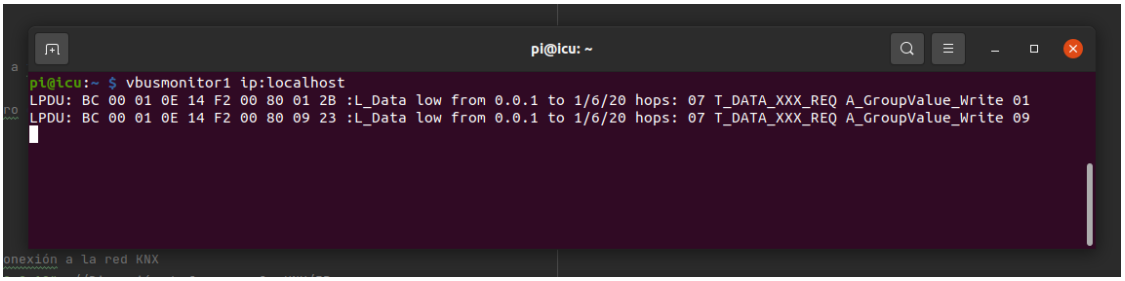
DeviceSlideTest
CheckAccionEstado() Este test crea un objeto DeviceSlide con dos Slides para cambiar entre un supuesto “modo frío” y un “modo calor” (de una máquina de clima por ejemplo). Primero se configura la máquina en frío (y se verifica que la pestaña que se visualiza es la adecuada) para posteriormente pasar a modo calor (y volver a verificar que es el estado actual)
Valores mostrados por consola de Bus


Figura 29: Test unitario DeviceSlideTest

Para ejecutar estos tests podemos utilizar en comando en la carpeta de proyecto:
./gradlew test

Esto ejecutará todos los tests definidos en el proyecto y almacenará los resultados en formato web en la subcarpeta “OpenKnxController/app/build/reports/tests/testDebugUnitTest/index.html”.

5.1.2 Pruebas empíricas

Estas pruebas son las que nos permiten comprobar que las acciones que llevamos a cabo en la aplicación tienen un resultado en “el mundo real”. Dada la idiosincrasia de estas pruebas, se mostrarán en el vídeo demostrativo del funcionamiento a entregar en la última entrega de documentación de este proyecto.

6. Conclusiones.

6.1 Revisión de la planificación inicial.

La fase de “Diseño y Arquitectura” se llevó a cabo según los plazos previstos en la planificación.

Es en la fase de “Implementación” donde se han producido variaciones sobre la planificación, en primer lugar, el desarrollo del “parser” conllevó 5 días en lugar de 2, así como, la implementación de la comunicación con la librería externa de comunicaciones a KNX llevó 8 días en lugar de los 5 previstos. Todo ello obligó a reducir de 3 prototipos previstos a 2 prototipos que fueron los que finalmente se realizaron, el primero incorporando todas las pantallas “generales” o comunes y el segundo implementando todos los dispositivos de control, tanto a nivel de lógica de negocio como en su capa de presentación. La fase de pruebas fue tal y como estaba previsto.

Por último para la entrega final se han mantenido los plazos indicados, con algún retraso en la finalización de la memoria (debido a obligaciones laborales) pero que finalmente se ha recuperado trabajando “extra” los fines de semana.

6.2 Lecciones aprendidas.

La primera lección y tal vez más importante, es el hecho de que previo a empezar a programar debemos tener claros los requisitos y necesidades que plantean nuestros potenciales usuarios, debemos diseñar de la manera más concienzuda posible tanto nuestro interfaz como la arquitectura a nivel de componentes, así como, el invariante para nuestra solución. Lanzarnos a programar y avanzar sobre la marcha sólo conducirá a un entregable fallido y a una solución que requerirá de continuos “parches2 para hacerla funcional.

Otra punto que hay que remarcar es que es importante la elección de la herramienta de desarrollo. Para el caso actual se ha utilizado Android Studio, que es una herramienta que no había utilizado previamente, lo cual ha conllevado una curva de aprendizaje sobre la misma que ha ocasionado un retraso sobre la planificación de las primeras etapas de la implementación. Por tanto, en ocasiones es mejor utilizar herramientas conocidas si contamos con plazos reducidos.

6.3 Objetivos cumplidos.

A nivel funcional hemos cubierto todos los requisitos expresados en el punto 1.2, gracias a todas las opciones semánticas de las que hemos dotado al fichero de configuración, es posible conseguir llevar a cabo todos ellos.

En el caso de los no funcionales, también los hemos cumplido, tal vez, el referido a “Establecer un interfaz de usuario que facilite la usabilidad en usuarios poco versados en tecnología” al tratarse de un objetivo de carácter más subjetivo (habrá usuarios que consideren el interfaz simple pero otros tal

vez no), requeriría de un test con usuarios para obtener sus impresiones y poder llegar a una conclusión a este respecto.

6.4 Líneas de trabajo futuro.

Hay una serie de puntos que se han aflorado durante el desarrollo de este proyecto que pese a no pertenecer a la planificación original se cree conveniente seguir desarrollando una vez realizada la entrega de este proyecto.

El primero de ellos sería al que hacíamos referencia en el punto anterior. Poder realizar Test de usuarios sobre el último entregable (que se corresponde con el segundo prototipo) para poder depurar el interfaz y la funcionalidad hacia los gustos predominantes indicados por dichos tests.

Otro aspecto a desarrollar estaría relacionado con la generación del fichero xml de configuración de la instalación. Dicho fichero puede llegar a ser bastante grande y complejo en cuanto a estructura si nuestra instalación es de gran tamaño. Sería muy interesante poder desarrollar una aplicación de escritorio que ayudase a generar dicho fichero de manera gráfica, esto lo haría más mantenible en el tiempo y sería mas fácil de usar por parte del público en general, así como, eliminaría la introducción de errores en el fichero que pueden darse al realizarlo de manera manual (comprobado de primera mano al realizar el fichero para las pruebas).

En cuanto a otras mejoras a añadir estaría la opción de incorporar tanto el fichero de configuración como un “set” de gráficos mediante carga desde la propia aplicación (por ejemplo en un empaquetado “zip”), en lugar de añadirlo como “assets” dentro del empaquetado “apk” como está realizado actualmente.

Una opción que sería muy interesante poder añadir las opciones necesarias en la interfaz gráfica para que una vez cargado el fichero de configuración (que ya establece una jerarquía de estancias y dispositivos), el usuario pudiera hacer un «remodelación» desde el propio interfaz de la aplicación, pudiendo añadir nuevas secciones y áreas y poblarlas con los dispositivos existentes según si criterio. A este respecto, en la versión actual ha dado tiempo a incluir la función de pulsación larga sobre un elemento para cambiar el nombre que viene por defecto definido en el fichero de configuración, quedando registrado el que quiera proporcionar el usuario.

Por último, el entorno ha quedado preparado para poder añadir otros protocolos a futuro que pudieran existir en el inmueble como “Modbus”[10] o “Bacnet”[11], dotando de este modo de un espectro aún más amplio de posibilidades de comunicación.

7. Glosario

E

- *ETS: "Engineering Tool Software", software para programación de todos los productos que han sido certificados como KNX.*

F

- *Framework*: es una plataforma conceptual donde código con funcionalidad genérica puede ser utilizado o reescrito por desarrolladores para conseguir un producto finalizado en menor tiempo.

K

- *KNX: Konnex, es un estándar abierto (EN 50090, ISO/IEC 14543) para automatización de edificios, evolucionó de tres sistemas estándar anteriores como EHS (European Home Systems Protocol), BatiBus y EIB (European Installation Bus). Puede utilizar cable de bus de par trenzado, red eléctrica "powerline", RF o enlaces IP.*

R

- RGB: acrónimo de "Red, Green and Blue" indica un método de definición de un color mediante la especificación de cada uno de los componentes indicados mediante un valor de 0 a 255. Aplicado a los sistemas de iluminación, decimos que un sistema es RGB si nos permite modificar el color con el que es capaz de iluminar dicho sistema.

S

- *SCADA: acrónimo de "Supervisory Control And Data Acquisition"* es un concepto que se emplea para realizar un software para ordenadores que permite controlar y supervisar procesos industriales a distancia.
- *SDK: acrónimo de "Software Development Kit", proporciona una serie de herramientas para la creación de aplicaciones en una determinada plataforma.*

U

- *URL: acrónimo de "Uniforme Resource Location", es un identificador de recursos diseñado para internet, en donde, la ubicación de los recursos puede variar con el tiempo.*

W

- *WIFI: es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con wifi (como ordenadores personales, teléfonos, televisores, videoconsolas, reproductores multimedia, etcétera) pueden conectarse entre sí o a Internet a través de un punto de acceso de red inalámbrica.*

8. Bibliografía

- **[1]Web:** https://www.knx.org/ 24/02/2022
- **[2]Web:**https://www.knx.org/knx-es/para-su-vivienda/como-empezar/ets-home-edition/ 24/02/2022
- **[3]Web:** https://es.wikipedia.org/wiki/SCADA 24/02/2022
- **[4]Web:** https://developer.android.com/studio 26/02/2022
- **[5]Web:** https://calimero-project.github.io/ 26/02/2022
- **[6]Web:** https://www.nachomadrid.com/2020/02/guia-entrevistas-usuarios/ 25/03/2022
- **[7]Web:** https://developer.amazon.com/es-ES/alexa 25/03/2022
- **[8]Web:** https://www.android.com/intl/es_es/ 25/03/2022
- **[9]Web:** https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd 29/03/2022
- **[10] Web:** https://modbus.org/ 25/05/2022
- **[11] Web:** http://www.bacnet.org/ 25/05/2022