



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Linares

## Trabajo Fin de Grado

# **CONTROL DE DOMÓTICO BASADO EN RASPBERRY PI CON INTERFAZ PARA KNX Y GESTIÓN A TRAVÉS DE TERMINAL MÓVIL**

Alumno/a: Juan de Dios Martínez Vallejo  
Tutor/a: Juan Carlos Cuevas Martínez  
Dpto.: Ingeniería de Telecomunicación

02, 2017

# 1 CONTENIDO

|       |  |           |
|-------|--|-----------|
| 1     | INTRODUCCIÓN .....                               | 2         |
| 2     | OBJETO .....                                     | 3         |
| 3     | ALCANCE .....                                    | 4         |
| 4     | ANTECEDENTES .....                               | 4         |
| 5     | JUSTIFICACIÓN DE LA TECNOLOGÍA.....              | 5         |
| 6     | ESTADO DEL ARTE.....                             | 6         |
| 6.1   | ESTÁNDAR DOMÓTICO .....                          | 6         |
| 6.2   | COMUNICACIÓN DE APLICACIÓN .....                 | 7         |
| 7     | DISEÑO DEL PROYECTO .....                        | 12        |
| 7.1   | <b>ANÁLISIS DE REQUISITOS .....</b>              | <b>12</b> |
| 7.1.1 | REQUISITOS FUNCIONALES.....                      | 12        |
| 7.1.2 | REQUISITOS NO FUNCIONALES.....                   | 14        |
| 7.1.3 | REQUISITOS DE DISEÑO .....                       | 14        |
| 7.1.4 | REQUISITOS DE INTERFAZ.....                      | 15        |
| 7.2   | <b>Recursos.....</b>                             | <b>15</b> |
| 7.2.1 | HARDWARE.....                                    | 15        |
| 7.2.2 | SOFTWARE .....                                   | 16        |
| 7.2.3 | RECURSOS HUMANOS .....                           | 16        |
| 7.3   | <b>ANÁLISIS DE SOLUCIONES .....</b>              | <b>17</b> |
| 7.4   | <b>DISEÑO .....</b>                              | <b>28</b> |
| 7.4.1 | RASPBERRY PI.....                                | 28        |
| 7.4.2 | DISPOSITIVO ANDROID .....                        | 30        |
| 7.4.3 | SERVIDOR.....                                    | 33        |
| 7.5   | <b>PROTOCOLO DE COMUNICACIONES.....</b>          | <b>34</b> |
| 7.5.1 | TIPOS DE MENSAJES.....                           | 34        |
| 7.5.2 | DISPOSITIVO ANDROID .....                        | 37        |
| 7.5.3 | RASPBERRY PI.....                                | 45        |
| 7.5.4 | SERVIDOR.....                                    | 49        |
| 7.6   | <b>MODELO DE CASOS DE USO .....</b>              | <b>55</b> |
| 7.6.1 | DESCRIPCIÓN CASO DE USO ACCESO AL SERVICIO ..... | 55        |
| 7.6.2 | DESCRIPCIÓN MODULO CONFORT .....                 | 61        |
| 7.6.3 | DESCRIPCIÓN MÓDULO CLIMATIZACIÓN .....           | 70        |
| 7.6.4 | DESCRIPCIÓN MÓDULO SEGURIDAD .....               | 72        |

|            |  |            |
|------------|--|------------|
| 7.6.5      | DESCRIPCIÓN MÓDULO MULTIMEDIA.....               | 75         |
| 7.6.6      | DESCRIPCIÓN MODULO TRADUCTOR DE VOZ A TEXTO..... | 85         |
| <b>7.7</b> | <b>IMPLEMENTACIÓN .....</b>                      | <b>86</b>  |
| 7.7.1      | DISEÑO.....                                      | 86         |
| 7.7.2      | DESCRIPCIÓN DETALLADA DE LA BASE DE DATOS.....   | 127        |
| <b>7.8</b> | <b>PRUEBAS.....</b>                              | <b>132</b> |
| <b>8</b>   | <b>CONCLUSIONES .....</b>                        | <b>135</b> |
| <b>8.1</b> | <b>OBJETIVOS CONSEGUIDOS .....</b>               | <b>135</b> |
| <b>8.2</b> | <b>PROBLEMÁTICAS ENCONTRADAS.....</b>            | <b>135</b> |
| <b>9</b>   | <b>LÍNEAS DE FUTURO.....</b>                     | <b>136</b> |
|            | <b>REFERENCIAS .....</b>                         | <b>137</b> |
|            | <b>DEFINICIONES Y ABREVIATURAS .....</b>         | <b>138</b> |
|            | ABREVIATURAS.....                                | 138        |

## Índice de Figuras

|            |  |    |
|------------|--|----|
| Figura 1.  | Servicio domótico.....                             | 3  |
| Figura 2.  | Esquema de comunicación .....                      | 9  |
| Figura 3.  | Bus KNX .....                                      | 10 |
| Figura 4.  | Topología Hogar.....                               | 11 |
| Figura 5.  | Formato telegrama.....                             | 21 |
| Figura 6.  | Asentimiento.....                                  | 22 |
| Figura 7.  | Esquema nueva conexión Raspberry Pi.....           | 57 |
| Figura 8.  | Esquema conexión Raspberry Pi .....                | 58 |
| Figura 9.  | Esquema nueva conexión Android.....                | 60 |
| Figura 10. | Esquema conexión Android.....                      | 61 |
| Figura 11. | Primer ACK encendido luces .....                   | 63 |
| Figura 12. | Primer ACK apagado luces .....                     | 63 |
| Figura 13. | Segundo ACK encendido luces .....                  | 63 |
| Figura 14. | Segundo ACK apagado luces .....                    | 64 |
| Figura 15. | Tercer ACK encendido luces, final .....            | 64 |
| Figura 16. | Tercer ACK apagado luces, final .....              | 64 |
| Figura 17. | Primer ACK encendido electrodomésticos.....        | 68 |
| Figura 18. | Primer ACK apagado electrodomésticos.....          | 68 |
| Figura 19. | Segundo ACK electrodomésticos .....                | 69 |
| Figura 20. | Tercer ACK encendido electrodomésticos, final..... | 69 |
| Figura 21. | Tercer ACK apagado electrodomésticos, final.....   | 70 |
| Figura 22. | Flujo de clases de conexión en servidor .....      | 86 |
| Figura 23. | Modelo de pila Raspberry Pi .....                  | 89 |
| Figura 24. | Funciones de pila Raspberry Pi .....               | 91 |

|   |     |
|---|-----|
| Figura 25. Flujo de datos Android .....           | 95  |
| Figura 26. Diagrama Activitys Android .....       | 96  |
| Figura 27. FragmentLogin .....                    | 97  |
| Figura 28. FragmentConnecting.....                | 98  |
| Figura 29. SignUpUser .....                       | 99  |
| Figura 30. SignUpIdHome .....                     | 99  |
| Figura 31. SignUpWait .....                       | 100 |
| Figura 32. ActivityHome .....                     | 101 |
| Figura 33. Estado conexión.....                   | 101 |
| Figura 34. Menu rápido.....                       | 102 |
| Figura 35. ActivityComfort.....                   | 103 |
| Figura 36. LightDevices.....                      | 104 |
| Figura 37. LightConfig .....                      | 104 |
| Figura 38. BlindDevices.....                      | 105 |
| Figura 39. BlindConfig .....                      | 106 |
| Figura 40. BlindRun.....                          | 106 |
| Figura 41. ApplianceDevices .....                 | 107 |
| Figura 42. ActivityScene.....                     | 108 |
| Figura 43. ActivityAir .....                      | 109 |
| Figura 44. ActivityHeating.....                   | 110 |
| Figura 45. ActivityHeating selección.....         | 110 |
| Figura 46. ActivityHeating activo.....            | 111 |
| Figura 47. AirConditioner .....                   | 112 |
| Figura 48. ActivitySecure .....                   | 113 |
| Figura 49. ActivityImages.....                    | 114 |
| Figura 50. ActivityImages selección cámara .....  | 114 |
| Figura 51. ActivityHolidays.....                  | 115 |
| Figura 52. ActivityMultimedia.....                | 116 |
| Figura 53. RemoteControl.....                     | 117 |
| Figura 54. RemoteControl agregar mando.....       | 117 |
| Figura 55. RemoteControlTV .....                  | 118 |
| Figura 56. RemoteControlAIR.....                  | 118 |
| Figura 57. RemoteControlMUSIC .....               | 119 |
| Figura 58. RemoteSaveInstruction.....             | 120 |
| Figura 59. RemoteSaveSteps .....                  | 120 |
| Figura 60. RemoteSaveFinal.....                   | 121 |
| Figura 61. Settings .....                         | 122 |
| Figura 62. SettingsSessions.....                  | 122 |
| Figura 63. SettingsUserHome .....                 | 123 |
| Figura 64. DrawerActivity menú lateral .....      | 124 |
| Figura 65. ActivityAlarm petición de usuario..... | 125 |
| Figura 66. ActivityAlarm alarma .....             | 125 |
| Figura 67. PinLock.....                           | 126 |
| Figura 68. Login Raspberry P .....                | 147 |
| Figura 69. Monitorización KNX .....               | 148 |
| Figura 70. Control por voz .....                  | 149 |
| Figura 71. Recepción de datos .....               | 150 |

|  |     |
|--|-----|
| Figura 72. Recepción de datos uno .....        | 151 |
| Figura 73. Recepción de datos dos.....         | 152 |
| Figura 74. Esquema conexión infrarrojo.....    | 153 |
| Figura 75. Login Android .....                 | 154 |
| Figura 76. Autenticación clientes.....         | 155 |
| Figura 77. Recepción datos servidor.....       | 156 |
| Figura 78. Recepción datos servidor uno.....   | 157 |
| Figura 79. Recepción datos servidor dos .....  | 158 |
| Figura 80. Recepción datos servidor tres ..... | 159 |
| Figura 81. Modelo base de datos.....           | 160 |

## 1. MEMORIA

## 1 INTRODUCCIÓN

El presente proyecto presenta un sistema de control domótico remoto que integra varias tecnologías de campos diferentes para conseguir una solución integral: KNX para el control domótico, la tecnología Raspberry Pi para la centralización de comunicaciones y servicios, y las aplicaciones para terminales móviles como interfaz de usuario. De esta manera, el sistema no solo permite el control de dispositivos domóticos KNX a través de su pasarela de red, sino la actuación sobre dispositivos convencionales operados con mandos a distancia, toma de imágenes, control por voz, gestión de usuarios y perfiles, y la facilidad de disponer de una interfaz de usuario para terminales móviles Android con todas las capacidades de actuación y programación en un entorno amigable y seguro.

La motivación de este proyecto radica en la importancia que representa para la telemática la aplicación de tecnología en el hogar. Esto abre un campo de posibilidades muy amplio, y la domótica es una de ellas, y que además de ser una disciplina técnica, tiene una importante faceta social. Como primera aproximación, la domótica puede definirse como la introducción de tecnología en los hogares para mejorar la calidad de vida de sus habitantes y ampliar sus posibilidades de comunicación, automatizando procesos domésticos o permitiendo el control de los dispositivos del hogar desde un mismo punto.

La domótica ha pasado por diferentes niveles de interés, pero hoy en día vuelve a estar en auge, y no es raro pues nos permite mejorar nuestra calidad de vida, tener un control preciso de lo consumido en cada instante, controlar dispositivos a través de la voz y estar más seguros en nuestros hogares con sofisticados sistemas de seguridad. Su crecimiento y adaptación a las pequeñas viviendas y grandes edificios resulta ser una curva exponencial pues en los últimos tiempos su precio es menor y sus características son mejoradas constantemente.

Dentro de tal termino, domótica, entra la gran necesidad que hoy día surge, tener un dispositivo móvil con conexión a Internet en todo momento. Las posibilidades que habilitan los estándares domóticos actuales por sus características técnicas recientes hacen que termine una época y comienza otra. Sus características permiten conectar redes domóticas de hogares o edificios a Internet y gracias a esto se posibilita un notable aumento de servicios prestados por terceros con la finalidad de hacer posible la gestión domótica remota desde cualquier dispositivo con conexión a Internet.

Éste proyecto abarca todas estas posibilidades anteriormente mencionadas, pero se hace especialmente hincapié en el hecho de poder conectar los hogares a través de Internet haciendo posible su gestión desde donde esté el usuario. Un esquema que simplifica en exceso lo realizado, pero ayuda a facilitar la comprensión de lo anteriormente expuesto y el fin de dicho proyecto es el siguiente.

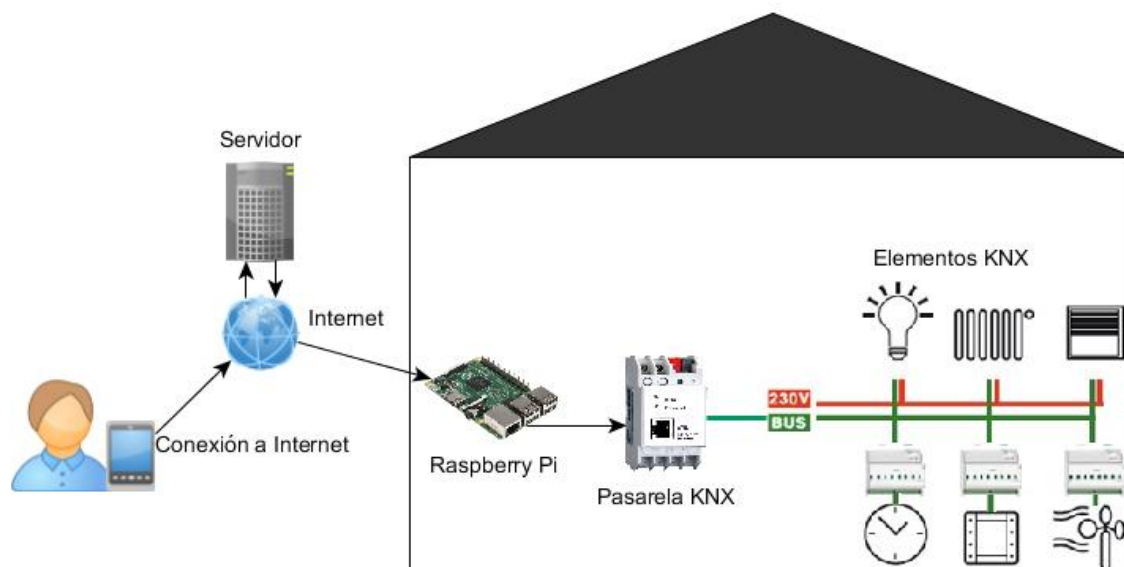


Figura 1. Servicio domótico

Como se observa en la Figura 1. Servicio domótico, un inquilino de un hogar digital podría acceder a todas las funciones posibles de su hogar y también a la monitorización de la totalidad de los dispositivos desde cualquier parte con el requisito de poseer de conexión a Internet, lo que, hoy día no resulta ser un problema para casi cualquiera de nosotros.

## 2 OBJETO

Realizar un software de gestión domótica para el hogar en un dispositivo Raspberry Pi que interactúe con elementos de control domóticos locales y con otros KNX a través de un módulo de gestión web del propio sistema.

Las funciones principales que debe tener el sistema son las siguientes:

- Definición, control y gestión de dispositivos, tales como sondas, actuadores, etc. directamente conectados al Raspberry Pi.
- Actuación a través del módulo de gestión de red KNX de los dispositivos compatibles con éste sistema en la casa.
- Gestión remota del servicio a través de una aplicación para terminal móvil.

Objetivo opcional:

- Control por voz de los recursos domóticos ofrecidos por el Raspberry Pi.

## **Requisitos de calidad.**

La aplicación móvil, así como la aplicación de gestión, deben cumplir suficientemente con los estándares de calidad de ambas plataformas y ser funcionales en diferentes tipos de dispositivos y resoluciones de pantalla.

## **3 ALCANCE**

El alcance de éste proyecto abarca el planteamiento, diseño, desarrollo software e instalaciones necesarias para el adecuado funcionamiento y control remoto de un sistema domótico visto como un servicio integral y final al usuario: programación de la funcionalidad asociada a cada uno de los componentes, control de los dispositivos domóticos a través de Internet, obtención del estado de los dispositivos domóticos del hogar, gestión de usuarios asociados a un hogar, control de elementos infrarrojos, y por último, control de los distintos elementos domóticos por voz. Dicho diseño será desarrollado minuciosamente en apartados consecutivos.

## **4 ANTECEDENTES**

En primer lugar, se tiene que, en julio de 2013 fue presentado el proyecto final de carrera en la Escuela Politécnica Superior de Linares con el título “*Proyecto domótico. Instalación de un sistema basado en el estándar KNX*” (Vargas Sandoval, 2013), realizado por José Alberto Vargas Sandoval.

El objeto de dicho proyecto era desarrollar un banco de entrenamiento utilizando el sistema domótico de bus de datos KNX. Dicho montaje se realizó para la utilización por parte de los alumnos y para el estudio y prácticas de éste sistema, buscar información acerca del sistema KNX, los elementos que pueden utilizarse en un montaje y estudiar y utilizar el software específico para éste determinado sistema de bus de datos.

En éste proyecto el elemento KNX que fue instalado, pero no utilizado fue el router KNX/IP. En el presente proyecto dicho módulo ha sido utilizado dado la necesidad existente hoy día acceso remoto. Así pues, el módulo router KNX/IP permite conectar la instalación KNX a una red de área local, y por extensión a Internet, abriendo así un amplio abanico de posibilidades para acceder a todas sus funciones gracias a la integración de tecnologías existentes.

La finalidad del proyecto es poder desarrollar y controlar un sistema que sirva como herramienta para el estudio y practica de los sistemas domóticos basados en el estándar KNX.

Gracias a al proyecto fin de carrera de José Alberto Vargas Sandobal se logra un correcto entendimiento de cómo funciona el protocolo KNX, desde aspectos básicos hasta entrar en gran detalle acerca de la comunicación entre dispositivos, instalación del hardware y software necesario compatible con el protocolo. Éste proyecto ha sido de gran ayuda para el desarrollo del presente trabajo, ya que, aunque no se ha realizado la instalación de los componentes KNX, si ha sido necesario conocer en profundidad el protocolo. Un tema de

gran importancia para éste proyecto es la comunicación que establecen los dispositivos KNX para lograr alcanzar el servicio domótico, el cual gracias al proyecto anteriormente mencionado ha sido comprendido adecuadamente y con claridad.

## 5 JUSTIFICACIÓN DE LA TECNOLOGÍA

Hace tiempo que las demandas del usuario en su hogar han cambiado, incrementando el nivel de exigencia y calidad en distintos ámbitos como son la seguridad, confort, eficiencia energética, automatización, gestión remota, entre otros. La actual situación hace que surja la necesidad de integrar estas nuevas tendencias con las anteriores necesidades existentes: distribución, conmutación y regulación de la energía eléctrica.

Para el control domótico resulta muy difícil esta integración utilizando la red eléctrica convencional, debido a diversos problemas entre los que encuentra el gran número de cables y dispositivos sin posibilidad de comunicación entre sí.

Una solución al problema anterior para conseguir dar servicio a las nuevas necesidades y posibilitar la integración anterior con la red eléctrica, es la realización de una instalación domótica en las viviendas.

Se ha elegido el protocolo domótico KNX, por cumplir con los requerimientos anteriormente expuestos, y por ser un estándar europeo e internacional con más de veinte años de antigüedad.

El objeto de éste proyecto es la gestión remota, a través de Internet, de una red domótica con el estándar KNX. Existen soluciones como las pasarelas de comunicaciones KNX que conectan la instalación domótica a la red local del hogar y por consecuente a Internet. En éste proyecto éste dispositivo es crucial ya que gracias a él tenemos acceso a los dispositivos KNX, esto será explicado más adelante.

Una de las limitaciones que presenta éste dispositivo es que con una configuración normal en la red de un hogar no es enrutable desde Internet. Para poder lograr esto, el router del hogar tiene que ser modificado por el usuario para poder llegar hasta los dispositivos KNX desde fuera del hogar.

Existen servicios que hacen uso de pasarelas de comunicaciones KNX. Estos tienen como limitación la configuración del router del hogar. Ya que el procedimiento de configuración de un router no es entendible ni realizable por todo el mundo. En este proyecto se solventa esta limitación, esto será explicado en los siguientes documentos.

El modulo hardware instalado en el hogar es el conocido Raspberry Pi. Como competidor se encuentra Arduino. La potencia de computación del Raspberry Pi es mucho mayor en casi todos los aspectos y contando como sistemas operativos disponibles para éste distribuciones Linux, hace de éste un hardware que permite realizar casi cualquier función. Gracias a que el Raspberry Pi está basado en Linux los posibles entornos de desarrollo (IDE) son más potentes que el de Arduino, ya que éste sólo acepta su IDE propietario. El precio entre estos dos dispositivos es semejante, siendo ambos bastante asequibles.

Por estas razones y por poseer conocimientos sobre Linux adquiridos durante mi paso por el Grado de Ingeniería Telemática se ha elegido como hardware inteligente en el hogar el Raspberry Pi.

## 6 ESTADO DEL ARTE

### 6.1 ESTÁNDAR DOMÓTICO

Existen diversas alternativas a la hora de realizar el diseño de una instalación domótica. Las diferencias entre los distintos protocolos se concentran en la arquitectura (centralizada, descentralizada y mixta), el tipo de sistema (propietario o libre), el tipo de infraestructura y finalmente, el precio. A continuación, se expondrán en el siguiente proyecto, los protocolos más relevantes en el mercado actual, con sus respectivas fortalezas y debilidades:

#### **X10 (X10, 1993): Arquitectura descentralizada.**

Ventajas:

- Precio de los equipos.
- Simplicidad.
- No requiere alta especialización.
- Control de hasta 256 dispositivos.
- Reutilización de la red eléctrica.

Desventajas:

- Sus dispositivos poseen una baja seguridad.
- El protocolo X-10, no incluye ningún mecanismo de análisis de errores.
- Delay de un segundo en el envío de señales.
- Otros dispositivos de la red eléctrica del hogar pueden causar interferencias.
- No permite formas de comunicación complejas, como, por ejemplo, seleccionar el canal de una televisión.

#### **LonWorks (LonWorks, 1999): Arquitectura descentralizada/centralizada/mixta.**

Ventajas:

- Reducción de la carga inteligente de los equipos.
- Extensa comunidad de desarrolladores.
- Modo de instalación “plug & play”.

Desventajas:

- Hardware específico, y requiere del chip “neuron” chip.
- Protocolo propietario.

### **EIB KONNEX (KNX, s.f.).**

Uno de los motivos principales de esta elección es que el estándar KNX lleva intrínseco una arquitectura distribuida en la que no existe dependencia alguna de un dispositivo central o maestro. Esta arquitectura es idónea para instalaciones que exigen cierta robustez, ya que la inteligencia está repartida por los distintos dispositivos. El uso de la arquitectura distribuida es aconsejable para instalaciones donde no resulte rentable invertir en un controlador maestro, como, por ejemplo, en pequeñas viviendas.

Ventajas KNX frente a otros sistemas:

- Incremento de la seguridad.
- Uso económico y responsable de la energía en la gestión de edificios.
- Aumento del confort.
- Instalaciones pensadas para el futuro.
- Fácil adaptación de la instalación eléctrica a las necesidades cambiantes del usuario final.
- Multiplicidad de productos disponibles debido a la gran diversidad de fabricantes.
- Gran número de profesionales cualificados.
- Inexistencia de la dependencia de un sólo fabricante.
- Existencia de pasarelas para otros sistemas o protocolos.
- Productos con precios competitivos y disponibilidad en stock.
- Productos con multiplicidad de funciones.
- Está aprobado como estándar Internacional ISO/IEC 14543-3, así como estándar europeo CENELEC EN 50090 y CEN EN 13321 -1 y estándar en China bajo la GB/Z 20965, por lo que KNX es prueba de futuro.
- Soporta diversos medios de transmisión como pueden ser cable bus, corrientes portadoras, radiofrecuencia, Ethernet entre otros.

## **6.2 COMUNICACIÓN DE APLICACIÓN**

La comunicación, la cual es uno de los objetos de éste proyecto, requiere de un análisis de la gestión remota para que sea alcanzable a través de Internet.

Como ha sido mencionado anteriormente, las soluciones existentes a día de hoy realizan una excelente gestión de los elementos KNX, siempre que el usuario se encuentre dentro de la misma red local.

Soluciones existentes:

- Houseinhand (Houseinhand, 2012).
- ayControl (ayControl, 2010).
- See-home (Electric, 2013).

La desventaja de estos servicios es la gestión domótica a través de Internet cuando el usuario no se encuentra conectado a la misma red local que la pasarela KNX, lo cual, es fundamental hoy día.

Para profundizar, y entender mejor estas soluciones anteriormente mencionadas, se describen a alto nivel cómo funcionan dichos servicios.

Para tener acceso al bus KNX y poder gestionar los dispositivos domóticos estos servicios realizan una conexión con la pasarela KNX del hogar para comenzar con la transmisión de datos. Las desventajas de brindar este servicio de tal forma son, entre otros, el limitado número de usuarios conectados y la configuración del router del hogar para poder tener acceso desde Internet.

La limitación del acceso desde Internet se encuentra presente debido a que la pasarela KNX está conectada a la red local del hogar. Su dirección IP de red es privada, y dada la estructura de Internet Protocol version 4 (IPv4), ésta no es accesible desde fuera de la red local. Para lograr esto, el usuario tiene que realizar algunas configuraciones en el router.

Las soluciones adoptadas por los servicios anteriormente mencionados para esta deficiencia son:

- Regla de redirección de puertos o más conocida por NAT (del inglés *Network Address Translation*).
- Establecimiento de una Red Privada Virtual o VPN (del inglés *Virtual Private Network* en el hogar.

Esto puede desembocar en un problema para el usuario ya que es muy corriente el reseteo del router o la actualización del software de éste. Tras estos procesos es posible que se pierda toda la configuración realizada en dicho dispositivo, y en consecuencia, realizar estas configuraciones reiteradamente puede llegar a ser algo muy incómodo para el usuario final.

También hay que tener en cuenta que la realización de una configuración de redirección de puertos no es comprensible por todos y mucho menos la creación de una VPN. Otro problema añadido a estos anteriores es que el usuario tendría que contratar una dirección IP pública estática o contratar algún servicio de resolución de nombres de dominio o DNS (*Domain Name System*).

El objeto de éste proyecto es brindar a los usuarios una gestión domótica remota eficiente, segura y cómoda. Tras los inconvenientes mencionados se va a proceder a la descripción de las soluciones adoptadas.

Para afrontar estos problemas se ha enfocado la gestión domótica de un modo distinto a los servicios anteriores. Han sido desarrollados distintos softwares, los cuales se ejecutan en distintos dispositivos: servidor público, Raspberry pi y Android.

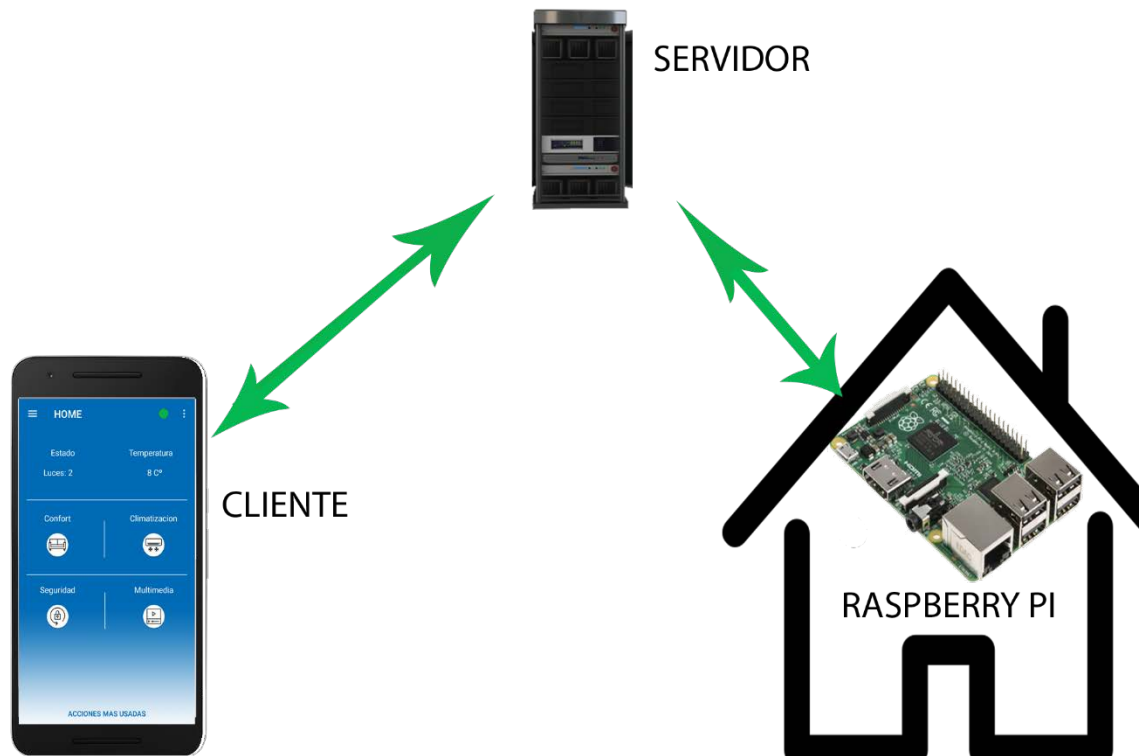
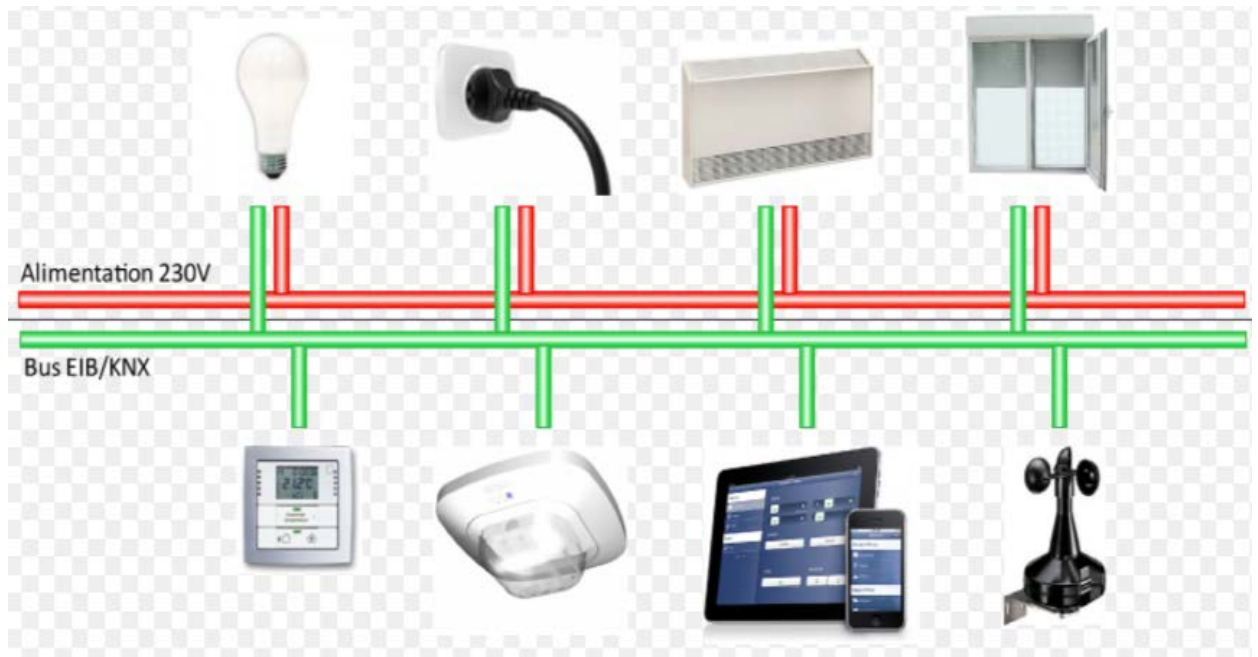


Figura 2. Esquema de comunicación

En la Figura 2. Esquema de comunicación se puede observar cómo se logra la comunicación entre el usuario y el hogar. También vemos que no aparece nada del estándar domótico, esto es debido a que el usuario no realiza ninguna comunicación con él, ya que, la comunicación con la domótica implantada es tarea del Raspberry pi (Figura 4. Topología Hogar). Con la estructura que se observa se procede a explicar cuál es la función de cada uno de estos para poder llegar a entender las soluciones.

- Servidor:
  - Es el software desarrollado que hace posible la comunicación entre los dispositivos Android y los Raspberry Pi.
  - Se encarga de enlazar una comunicación existente entre usuarios y pasarela Raspberry instalada en el hogar digital.
  - Gestiona los usuarios.
  - Gestiona sesiones de usuarios.
  - Éste también es el encargado de escuchar a los dispositivos Raspberry Pi ante cambios en los dispositivos KNX en los hogares y así comunicar estos a los usuarios y almacenarlos en la base de datos.
- Android:
  - Es el software de control por parte del usuario.
  - Realiza una conexión TCP con el servidor.
  - Realiza peticiones al servidor de los dispositivos instalados en el hogar consiguiendo un estado actualizado de los dispositivos en todo momento.
  - Es el encargado de recibir las órdenes de los usuarios por medio de su interfaz gráfica, transmitir las al servidor y con esto hasta el hogar.

- Bus KNX: KNX es un sistema bus descentralizado y controlado por eventos. Esto quiere decir que los sensores detectan eventos en los hogares procedentes de pulsadores o motivados por cambios en la luminosidad, la temperatura, la humedad o los movimientos entre otros. A continuación, envían telegramas al bus y finalmente estos telegramas llegan a los actuadores o pasarelas, los cuales llevan a cabo las órdenes. Su topología se puede observar en la Figura 3. Bus KNX.



*Figura 3. Bus KNX*

- Pasarela IP/KNX: se trata de un dispositivo modular que se puede integrar en las cajas de distribución. Utiliza la norma KNXnet/IP y actúa como interfaz entre las instalaciones KNX y las redes de datos que utilizan el protocolo Internet (IP). Permite a un PC u a otros aparatos de tratamiento de datos acceder al bus. La conexión al bus KNX se establece por medio de un terminal de conexión bus estándar. La conexión a la red informática (IP vía 10 BaseT) se realiza por medio de un conector RJ45. Este dispositivo es el encargado de la retransmisión de los telegramas KNX enviados a bus KNX a la red local del hogar. De igual forma también es el encargado de la retransmisión de los telegramas enviados por el Raspberry al bus KNX.
- Raspberry pi:
  - Establece una conexión con el servidor.
  - Realiza la función de pasarela entre usuario y dispositivos KNX.
  - Establece una conexión con la pasarela KNX. Con esta conexión se escucha las retransmisiones que dicho dispositivo realiza ante telegramas KNX enviados a través del bus.
  - Informa al servidor de los cambios realizados en los dispositivos domésticos del hogar.

- También se encarga de recibir órdenes por parte de los usuarios, ensamblar la orden en el formato de telegrama KNX, enviarla a la pasarela KNX y así al dispositivo final para la ejecución de la orden.
- También es el encargado de la gestión de control por voz de los recursos domóticos.
- Otra funcionalidad que posee es el control de elementos con receptores infrarrojos. Realiza el grabado de mandos a distancia infrarrojos para posteriormente hacer uso de estos.

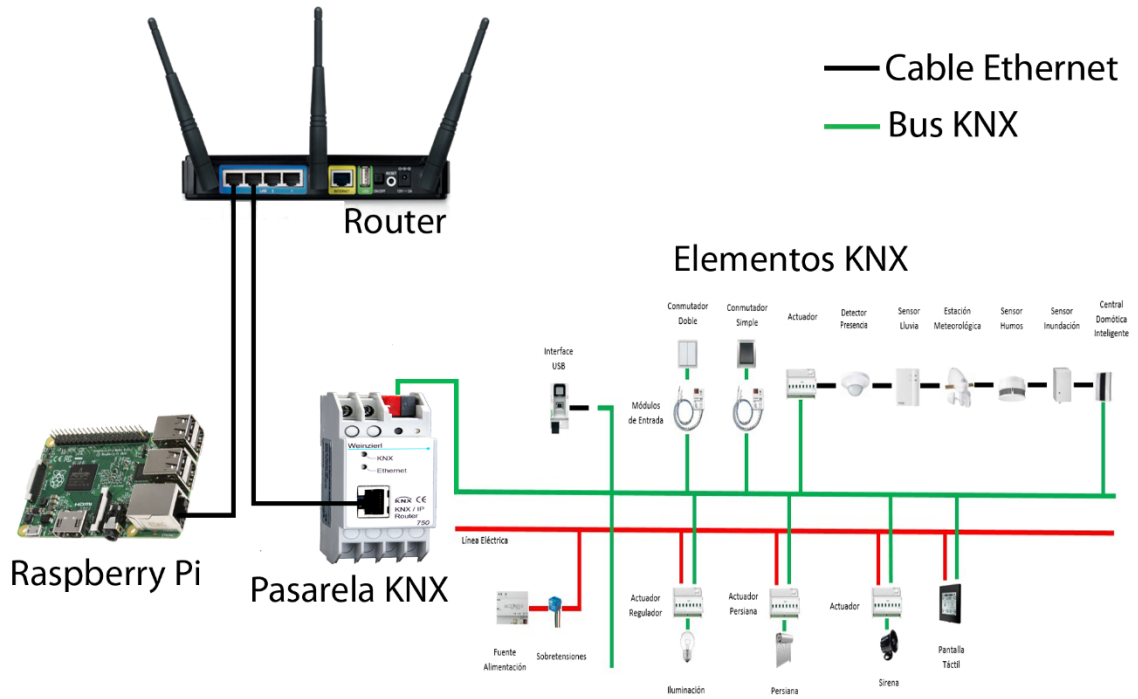


Figura 4. Topología Hogar

Tras esta breve explicación es más sencillo comprender las soluciones adoptadas.

El problema que limita a los usuarios conectados simultáneamente es solventado mediante este esquema ya que sólo se establece una conexión con la pasarela KNX por parte del Raspberry Pi. Así, éste es el encargado de toda la comunicación con los elementos KNX.

El gran problema que establece la configuración del router por parte del usuario se soluciona ya que el dispositivo encargado de la interacción con el usuario no establece una comunicación directa con la pasarela KNX, sino con el servidor público de aplicación. Con éste esquema (Figura 2. Esquema de comunicación) Cliente/Servidor y Pasarela/Servidor no es necesario ningún tipo de ajustes por parte del usuario. Toda la comunicación es transparente para él. El usuario final no tiene que conocer nada acerca de cómo se brinda el servicio, tampoco entrar en los ajustes.

## 7 DISEÑO DEL PROYECTO

### 7.1 ANÁLISIS DE REQUISITOS

Las diferentes acciones que el sistema ha de tratar han de ser objeto de análisis. Esto implica que una vez se ha entendido el problema, es necesario especificar sus requisitos. Hay que obtener todos los requisitos y poner de manifiesto las necesidades y los objetivos que se pretenden alcanzar.

El hecho de laborar toda esta información, permitirá valorar la calidad del software que se diseñará y si éste cumple nuestras expectativas.

#### 7.1.1 REQUISITOS FUNCIONALES

**SERVIDOR:** El servidor de aplicación debe estar en todo momento disponible para conexiones entrantes tanto de Raspberry Pi como de dispositivos Android. Permanentemente tendrá conocimiento de los canales TCP establecidos.

##### 7.1.1.1 USUARIOS Y SEGURIDAD

El servicio debe garantizar que únicamente los usuarios autorizados puedan acceder a la aplicación y de esta manera asegurar la confidencialidad de los datos almacenados en la aplicación.

##### 7.1.1.2 GESTIÓN DE USUARIOS

La aplicación deberá permitir el registro y eliminación de usuarios. También la visualización de usuarios vinculados a un hogar.

##### 7.1.1.3 GESTIÓN DE CONTRASEÑAS DE USUARIO

La aplicación deberá permitir la modificación de contraseña de usuario por parte de éste.

##### 7.1.1.4 MULTIUSUARIO

El servicio debe ser multiusuario, lo que significa que múltiples usuarios pueden acceder al servicio simultáneamente.

##### 7.1.1.5 GESTIÓN DE SESIONES

La aplicación deberá permitir el cierre de sesiones de usuario, para así, proporcionar una seguridad adecuada al servicio.

##### 7.1.1.6 DUPLICIDAD DE DATOS

La aplicación deberá verificar que no se produzcan duplicidad de datos y garantizar la correspondencia correcta entre los datos. A la hora de realizar inserciones y modificaciones de dispositivos KNX del hogar, usuarios y sesiones se deberá evitar que la aplicación permita añadir datos repetidos o con una correspondencia incorrecta.

#### 7.1.1.7 DISPOSITIVOS KNX

Los dispositivos deberán estar correctamente insertados en la base de datos para la correcta visualización y gestión de estos.

#### 7.1.1.8 ESTADO DE DISPOSITIVOS KNX

El servicio deberá tener actualizado en todo momento el estado de los dispositivos KNX del hogar para garantizar una precisa visualización por parte del usuario.

#### 7.1.1.9 SOFTWARE DE RASPBERRY PI

El software desarrollado y ejecutado en el Raspberry Pi debe de ser estable. El canal de comunicación siempre ha de estar activo. Éste debe ser capaz de encapsular una petición de usuario a telegrama KNX. De igual forma debe permitir también la desencapsulación de los telegramas KNX a un lenguaje de aplicación y con esto, de entendimiento para el usuario.

#### 7.1.1.10 ESTADO DE PASARELA RASPBERRY PI

La aplicación debe proporcionar al usuario el estado de la conexión entre el hogar y el servidor de aplicación. Esto engloba la conexión entre dispositivo Android/Servidor y Raspberry Pi/Servidor. Esta información es de gran ayuda para el usuario ya que así conoce el estado de la comunicación en todo momento.

#### 7.1.1.11 GESTIÓN DE LOS ELEMENTOS DOMÓTICOS A TRAVÉS DE VOZ

La aplicación desarrollada en el Raspberry Pi debe proporcionar al usuario el servicio de control de voz de los distintos elementos KNX del hogar.

#### 7.1.1.12 GESTIÓN DE LOS ELEMENTOS INFRARROJOS

La aplicación desarrollada en el Raspberry Pi debe proporcionar al usuario el servicio de control de los dispositivos que constan de un receptor infrarrojo para su gestión. También tiene que incluir la posibilidad de incorporar al servicio nuevos controles infrarrojos por parte del usuario.

#### 7.1.1.13 SEGURIDAD DEL CANAL DE COMUNICACIONES

El servicio debe garantizar una comunicación segura con las siguientes propiedades: confidencialidad, integridad de los datos y autenticación.

#### 7.1.1.14 COPIAS DE SEGURIDAD

La aplicación deberá tener un sistema de copias de seguridad que garantice la copia de los datos y que permita la restauración de estos de manera automática en caso de ser necesario. Las copias realizadas por la aplicación deberán almacenarse con fecha para poder tener un control sobre estas.

## 7.1.2 REQUISITOS NO FUNCIONALES

Estos requisitos afectan al diseño del propio sistema y tienen relación directa con el rendimiento de la solución desarrollada.

### 7.1.2.1 REQUISITOS DE RENDIMIENTO

Los requisitos de rendimiento son aquellas restricciones que afectan al rendimiento del sistema, como pueden ser las características y el comportamiento de la ejecución del programa.

#### 7.1.2.2 CONCURRENCIA

La aplicación podrá ser usada al mismo tiempo por diversos usuarios, sin que esto suponga ningún problema entre ellos.

#### 7.1.2.3 TIEMPO DE RESPUESTA

La velocidad de la aplicación deberá estar presente durante todo el desarrollo de la aplicación. Los datos transmitidos deberán ser reducidos en la medida de lo posible para garantizar la velocidad y el tiempo de respuesta de la aplicación.

## 7.1.3 REQUISITOS DE DISEÑO

Los requisitos de diseño son factores que restringen las opciones del diseñador y que están presentes en el entorno del propio cliente y permiten garantizar la seguridad y fiabilidad del sistema.

### 7.1.3.1 COMPATIBILIDAD

Se deberá garantizar que la aplicación funciona correctamente en su totalidad en dispositivos Android Ice Cream Sandwich 4.0 o superior. También en versiones Debian jessie 8.0 o superior.

#### 7.1.3.2 SERVIDOR DE APLICACIÓN

El servidor para la aplicación será privado, pero se ha de garantizar el poder acceder adecuadamente desde el exterior, manteniendo siempre la seguridad y la confidencialidad de los datos que se almacenan en la aplicación.

#### 7.1.3.3 FIABILIDAD Y CONSISTENCIA

Con tal de que la aplicación no presente inconsistencias, se deberá garantizar, siempre que sea posible, la validez de los datos introducidos por los usuarios, detectando y avisando de posibles errores.

#### 7.1.3.4 DUPLICIDAD Y RELACIÓN DE DATOS

La base de datos deberá garantizar la no duplicidad de datos en aquellos puntos necesarios para el buen funcionamiento de la aplicación. Aquellos datos que precisen de

relación entre ellos deberán garantizar no ser cruzados con otros que puedan provocar datos incorrectos.

#### 7.1.3.5 SEGURIDAD

Se ha de garantizar la protección de los datos almacenados en la base de datos con los mecanismos necesarios para evitar que ningún individuo ajeno no autorizado pueda acceder a los datos almacenados.

Debido al hecho de que la aplicación almacena datos de carácter personal, la seguridad de la base de datos debe garantizar que se respeta la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal.

La identificación de usuarios se deberá realizar de manera rigurosa cada vez que se use la aplicación.

#### 7.1.4 REQUISITOS DE INTERFAZ

Estos requisitos especifican las características de la interacción entre el sistema a desarrollar y los usuarios. Marca unas reglas en la manera de mostrar la información por parte de la aplicación hacia el usuario.

##### 7.1.4.1 SIMPLICIDAD Y CLARIDAD

La interfaz gráfica deberá ser sencilla y clara, mostrando la información en un tamaño adecuado, y sin hacer uso de efectos que retrasen la ejecución de la aplicación.

##### 7.1.4.2 INTERFAZ INTUITIVA

Todas las partes del software deberán mantener el mismo estilo y éste deberá tener unos tonos cálidos.

#### 7.2 RECURSOS

Para poder alcanzar el fin de desarrollar dicho proyecto y también conocer su viabilidad, hay que examinar el requisito de poseer distintos recursos. Estos pueden ser englobados en tres grupos: hardware, software y recursos humanos.

Por otra parte, para los usuarios no es requisito que estos dispongan de niveles altos de conocimientos informáticos dado que la aplicación será sencilla y eficaz. Se realizará un manual de usuario para facilitar la experiencia de uso y obtener el máximo provecho de la aplicación. Esta formación será realizada por el instalador del servicio en el hogar.

##### 7.2.1 HARDWARE

El hardware recomendado para el despliegue del sistema y pruebas son los siguientes:

###### 7.2.1.1 Requisitos hardware del servidor de aplicación

- Un equipo informático Intel Pentium IV.

- 4 GB de memoria RAM.
- Tarjeta de Red.
- Cable CAT-6.

#### 7.2.1.2 Requisitos hardware para los usuarios

- Dispositivo Android.
- Dispositivo Android.
- 512 Mb de memoria RAM.

#### 7.2.1.3 Requisitos hardware para hogares

- Raspberry Pi 2 Model B o posterior.
- Tarjeta de memoria micro sd 8GB.
- Cámara USB.
- Micrófono.
- Convertidor USB a Jack 3.5mm
- Cable CAT-6.

### 7.2.2 SOFTWARE

El software recomendado para el despliegue del sistema y pruebas son los siguientes:

#### 7.2.2.1 Requisitos software del servidor de aplicación

- Windows 7 Profesional o superior.
- Máquina virtual Java, versión 8.
- MySQL.
- Python 3.5.
- Django.

#### 7.2.2.2 Requisitos software en los equipos de los usuarios

- Android 4.0 o superior

#### 7.2.2.3 Requisitos software para hogares (Raspberry Pi)

- Debian jessie 8.0 o superior.
- Python 3.4 o superior.

### 7.2.3 RECURSOS HUMANOS

Los recursos humanos de éste proyecto engloban al autor y al tutor de dicho proyecto.

## 7.3 ANÁLISIS DE SOLUCIONES

A continuación, se mostrarán todas las soluciones adoptadas para la implementación total de dicho proyecto.

### 7.3.1.1 LENGUAJE DE PROGRAMACIÓN EN EL LADO DEL SERVIDOR

El servidor ha sido desarrollado con el lenguaje de programación Java. Las razones por las que ha sido implementado con este lenguaje son las siguientes:

- Independiente de la Plataforma.
- Gestión de Memoria.
- Librerías Estándar.
- Comunidad Open Source.
- Gestión de Errores.
- Multi hilo y Sincronización.

### 7.3.1.2 LENGUAJE DE PROGRAMACIÓN EN EL LADO DEL RASPBERRY PI

En el Raspberry Pi el software realizado se ha realizado con el lenguaje Python version 3.4.2. Las razones por la que ha sido implementado con este lenguaje son las siguientes:

- Es de código abierto, permite crear nuestras aplicaciones e involucrarnos en su desarrollo.
- Tipado dinámico, el dato se determinará en tiempo de ejecución, según el valor que se le asigne a una variable.
- Multiplataforma
- Gran cantidad de librerías.
- Buenas librerías para Raspberry Pi.

### 7.3.1.3 PROTOCOLO DE APLICACIÓN

Es necesario examinar el tipo de conexión que se realizará entre los distintos dispositivos. Esto es relevante debido al peso que tiene en la comunicación.

Ante el tipo de conexión se planteó el intercambio de datos mediante:

- Hypertext Transfer Protocol (HTTP).
- Socket.

Se ha escogido y desarrollado el intercambio de información mediante Socket utilizando como protocolo de transporte Transmission Control Protocol (TCP).

Para poder proporcionar un servicio libre de errores el protocolo de transporte utilizado es TCP el cual es un protocolo de transporte confiable. Es un protocolo orientado a conexión y asegura que los datos enviados sean recibidos por el receptor sin errores y en el mismo orden en el que fueron emitidos.

Se ha desarrollado la comunicación utilizando socket dado que:

- Permite una comunicación asíncrona.
- No hace uso de cabeceras, lo que significa un disminuido volumen de datos a transmitir por parte del usuario.
- Tanto el usuario como el servidor pueden enviar información sin necesidad de una petición por parte del usuario.

Por estos motivos toda la comunicación ha sido desarrollada mediante Socket TCP.

#### 7.3.1.4 INTERCAMBIO DE IMÁGENES

En el apartado anterior ha sido descrito el tipo de comunicación establecida entre todos los miembros del servicio.

No obstante, también ha sido utilizado el protocolo de aplicación HTTP. Dado que el servicio proporcionado es multiusuario y la carga de cómputo del servidor podría verse afectada ante un gran número de usuarios conectados simultáneamente, se ha querido rebajar la carga al servidor de aplicación.

Todas las imágenes intercambiadas en el servicio son transferidas a través del protocolo HTTP por un servidor externo, el cual ha sido desarrollado en Python mediante el Frameworks Django.

Se ha escogido éste Frameworks para el intercambio de imágenes por ser un Frameworks web de alto nivel que fomenta el desarrollo rápido.

#### 7.3.1.5 FORMATO DE INTERCAMBIO DE DATOS

En la actualidad son dos los grandes estándares existentes para el formato de intercambio de datos. Cada uno posee ventajas respecto al otro.

- eXtensible Markup Language (XML)
  - Ventajas:
    - Tiene un formato muy estructurado y fácil de comprender.
    - Puede ser validado fácilmente mediante Schemas(XSD).
    - Se pueden definir estructuras complejas y reutilizables.
  - Desventajas:
    - Es más complicado de entender.
    - El formato es sumamente estricto.
    - Conlleva más tiempo procesarlo.
    - Un error con los namespace puede hacer que todo el documento sea inválido.

- JavaScript Object Notation (JSON)
  - Ventajas:
    - Formato sumamente simple.
    - Velocidad de procesamiento alta.
    - Archivos de menor tamaño.
  - Desventajas:
    - Sintaxis simple, no todos los tipos de datos son compatibles.

Para éste proyecto se ha utilizado el formato de intercambio de datos JSON por ser un formato simple, ligero y rápido.

Una de las grandes ventajas de utilizar éste formato es el reducido tamaño de datos a transferir con una estructura bien definida. La poca carga de datos utilizando éste formato logra un intercambio de datos ligero para el usuario, y gracias a esto un reducido consumo de datos móviles, lo cual, hoy día es importante por las redes móviles.

#### 7.3.1.6 BASE DE DATOS

A la hora de realizar una aplicación, una parte importante para el desarrollo son las bases de datos. Tenemos distintas bases de datos como son Oracle, MySQL, PostgreSQL, NoSQL.

Como podemos ver existen multitud de bases de datos que se pueden utilizar. A continuación, enumero algunas razones por las que se utiliza MySQL.

- Multiplataforma. Para S.O. como Windows, Linux y Mac, esto hace que sea fácilmente trasladable.
- Fácil encontrar ayuda. Al ser una base de datos que se utiliza en multitud de aplicaciones web existen multitud de tutoriales o foros en la red en los que podemos encontrar la información que necesitamos.
- Fácil de aprender. Simplemente con conocer el estándar de SQL podemos manejar la base de datos MySQL sin ningún problema.
- MySQL es una base de datos ampliamente probada por distintos usuarios y empresas con alto éxito.
- Menos características. Menos mantenimientos. Realmente esto nos da la ventaja para que un programador cualquiera pueda aprender rápidamente como debe mantener la base de datos para sus aplicaciones. Sin necesidad de ser un experto Administrador en Base de Datos (DBA). Bases de datos como Oracle requieren de DBA para la gestión de su información debido a todas las características que existen para su administración. En cambio, MySQL para el funcionamiento habitual de una aplicación incluye unas características mínimas que nos sirven ampliamente para nuestras aplicaciones sin tener que recurrir a un DBA que administre la base de datos.
- El coste total de inicio es ampliamente inferior al de Oracle o Microsoft SQL Sever. Permite la opción de crear tantos servidores MySQL como sean necesarios sin incurrir en ningún coste.

- MySQL es escalable. Soporte para las transacciones. Una de las grandes ventajas de utilizar MySQL es la gestión de las transacciones. Las transacciones en MySQL son Atómicas, consistentes, aisladas (isolated) y durable; ACID.

Por estas razones anteriormente mencionadas se ha utilizado MySQL como sistema de gestión de bases de datos relacional.

### 7.3.1.7 COMUNICACIÓN CON DISPOSITIVOS KNX

La comunicación que debe haber entre el Raspberry Pi y los elementos KNX del hogar el fundamental para éste proyecto, debe ser estable, rápida y debe ser libre de errores.

Lo que éste apartado pretende explicar es la comunicación que se establece entre el Raspberry Pi y la pasarela KNX/IP para la monitorización del bus KNX y la interacción con éste. Para conseguir el fin de éste apartado que intenta describir por qué es necesario la comunicación con el bus, se detalla el formato de telegramas KNX.

Siempre que se detecte un evento se producirá el envío de un telegrama KNX. El telegrama que se transmite por el bus y que contiene la información específica sobre el evento que se ha producido, tiene siete campos, seis de control para conseguir una transmisión fiable y un campo de datos útiles con el comando a ejecutar.

- Control. Éste campo de 8 bits incluye la prioridad de dicho telegrama, es decir, el tipo de función (alarma, servicios del sistema o servicios habituales). El bit de repetición se pone a cero en caso de repetirse algún envío a causa del no reconocimiento de alguno de los destinatarios. De éste modo se evita que los mecanismos que ya han ejecutado la orden la vuelvan a repetir.
- Dirección de origen. El dispositivo que retransmite la trama envía su dirección física (4 bits con el área, 4 bits de identificador de línea y 8 bits de identificador de dispositivo), de modo que se conozca el emisor del telegrama en las tareas de mantenimiento.
- Dirección de destino. La dirección de destino puede ser de dos tipos en función del valor que tome el bit de mayor peso de éste campo (bit 17). Si tiene valor '0', se trata de una dirección física, y el telegrama se dirige únicamente a un dispositivo. Si tiene valor '1', se trata de una dirección de grupo, y el telegrama se dirige a todos los mecanismos que deben escucharlo (los que tengan esa dirección de grupo).

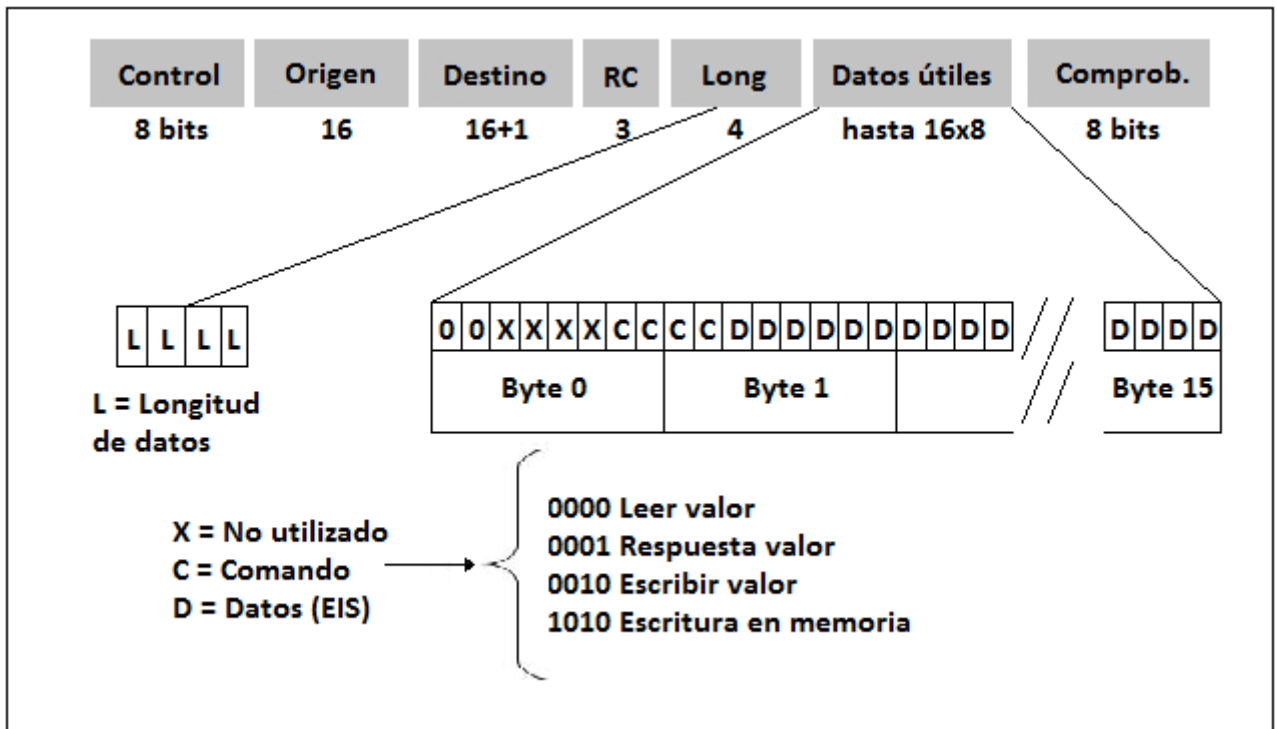


Figura 5. Formato telegrama

- Longitud e información útil. Contiene los datos necesarios para la ejecución de órdenes y la transmisión de valores. En los cuatro bits de longitud se indica cuántos bytes contiene el campo de datos (0 = 1 byte, 15 = 16 bytes). El campo de datos útiles contiene el tipo de comando y los datos, de acuerdo con el EIB Interworking Standard (EIS).

El EIS contiene los datos útiles para cada función asignada a los objetos de comunicación. Según éste estándar existen siete tipos diferentes, cada uno asignado a un tipo de acción de control (conmutación, regulación de luz, envío de valor absoluto, envío de valor en punto flotante, etc). De éste modo se garantiza la compatibilidad entre dispositivos del mismo tipo de diferentes fabricantes.

Los objetos de comunicación son instancias de clases definidas en el estándar, y son los programas almacenados en la memoria de los dispositivos para realizar una determinada acción.

- Campo de comprobación. Consiste en un byte que se obtiene del cálculo de la paridad longitudinal par (LRC2) de todos los bytes anteriores incluidos en el telegrama. Cuando un dispositivo recibe el telegrama, comprueba si éste es correcto a partir del byte de comprobación. Si dicha recepción es correcta, se envía un reconocimiento. De lo contrario, se envía un no reconocimiento (NAK) para que el emisor repita el envío. Si el dispositivo está ocupado envía un código Busy para que el emisor reintente la transmisión tras un pequeño retardo.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |                                   |
|----|----|----|----|----|----|----|----|-----------------------------------|
| N  | N  | 0  | 0  | B  | B  | 0  | 0  |                                   |
| 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | <b>BUSY (ocupado)</b>             |
| 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | <b>NAK (Recepción incorrecta)</b> |
| 1  | 1  | 0  | 0  | 1  | 1  | 0  | 0  | <b>ACK (recepción correcta)</b>   |

Figura 6. Asentimiento

Como puede comprobarse esta comunicación entre dispositivos es un estándar impuesto por KNX. Para monitorizar los dispositivos y realizar funciones tales como encender, apagar o regular en éste proyecto deben enviarse al bus tramas con dicho formato.

El desarrollo de un software que realice esta transformación de mensajes es una tarea larga. Existen proyectos y librerías open source en las cuales este proyecto se apoya para realizar la transmisión de datos con KNX.

Tras esta explicación, es necesario saber que la comunicación entre el Raspberry Pi y los dispositivos KNX es de gran importancia ya que si no es óptima la monitorización del bus KNX podría desembocar en una lectura de los estados de los dispositivos errónea y, en consecuencia, la interacción entre Raspberry Pi y KNX podría verse afectada.

Como se ha mencionado anteriormente existen dos grandes librerías para establecer comunicación con KNX a través de su pasarela. Estas son:

- El Proyecto Calimero (Calimero, 2005).  
Proporciona bibliotecas de Java de la pila de protocolo KNX para clientes, servidores KNXnet/IP y dispositivos KNX.  
Calimero está diseñado para funcionar en sistemas embebidos. Los requisitos mínimos son JRE ya sea Java 2 Micro Edition CDC con el Perfil de Fundación (J2ME CDC FP), o Java SE 8 con el perfil Compact1. Las bibliotecas son compatibles con entornos comúnmente desplegados en los ordenadores personales Java Standard Edition (J2SE).
- Librería EIBD (EIBD, 2005).  
El demonio EIBD es un componente central de la SDK BCU. Proporciona una interfaz al bus EIB/KNX desarrollada en C.  
EIBD proporciona más de una red TCP/IP y/o UNIX sockets de dominio de acceso al bus EIB usando un protocolo simple. Proporciona acceso en la capa 4, así como a la función de gerencia de alto nivel. Múltiples usuarios concurrentes son compatibles. Establece un monitor de bus KNX llamado Busmonitor, que entrega todo el tráfico y no perturba ninguna actividad KNX.  
Además, EIBD actúa como un túnel EIBnet/IP y/o servidor de enrutamiento. La limitación es que sólo una dirección KNX puede ser utilizada como dirección de origen de los clientes EIBnet/IP. Esta dirección es compartida entre varias posibles

conexiones de enrutamiento o de túnel. Para el cliente EIBnet/IP, utiliza la dirección 0.0.0.

Ante estas dos posibilidades para el acceso al bus KNX, las cuales poseen unas características semejantes en éste proyecto, se han utilizado las librerías EIBD.

La fundamental funcionalidad por la que se utiliza esta librería y no en Calimero es que con Calimero he podido comprobar que sólo permite una conexión con la pasarela KNX, esto deja al Raspberry Pi con sólo una opción, la monitorización o la interacción con los dispositivos KNX. No se puede abrir una conexión para escuchar el bus y otra para mandar telegramas.

Esta no es una limitación para EIBD pues realiza estas dos funciones sin problemas. Así se consigue tener siempre un proceso activo en el Raspberry Pi monitorizando el bus.

### 7.3.1.8 CONTROL DOMÓTICO POR VOZ

Es requisito para dicho servicio la inclusión del control domótico a través de voz. Para tal fin han sido estudiadas distintas estrategias, las cuales tienen algunas ventajas frente a otras.

La diferencia de las estrategias para el reconocimiento de voz es hacer uso de Internet para la transformación de voz a texto.

Una gran empresa que brinda el servicio de convertir voz a texto mediante software es Google. Con la API Google's Speech Recognition permite a los desarrolladores consumir éste servicio. Para poder hacer uso de éste servicio el dispositivo que implementa el software de la API tiene que estar conectado a Internet ya que son los servidores de google los que realizan la transformación de voz a texto. Éste no es su aspecto más atractivo ya que limita el servicio a esta necesidad.

Por otro lado, también hay tratamientos digitales para convertir de voz a texto por medio de diccionarios. Una librería de código abierto bastante conocida y recomendada es CMU Sphinx, que realiza las transformaciones por medio de diccionarios almacenados localmente, sin necesidad de estar conectado a Internet y por consecuencia posibilita a dicho servicio estar siempre activo.

Se realiza un resumen de las ventajas y desventajas de cada uno:

- API Google Speech Recognition:
  - Ventajas
    - Rapidez en la transformación de voz a texto.
    - Transformaciones de voz a texto muy fiables.
    - Se pueden realizar todas las transformaciones requeridas.
    - Servicio gratuito.
  - Desventajas
    - Necesidad de conexión a Internet.

- CMU Sphinx (Carnegie Mellon, 2002):
  - Ventajas
    - No hace uso de conectividad a internet.
    - Posibilita al usuario a crear su propio diccionario con las palabras requeridas.
    - Open Source.
    - Se pueden realizar todas las transformaciones requeridas.
  - Desventajas
    - Un poco más lento en la transformación que API de Google.
    - El usuario tiene que crear sus propios diccionarios.
    - La transformación de voz a texto puede contener errores.

Para éste proyecto se ha optado por utilizar la API de Google Speech Recognition por el brillante reconocimiento de voz que realiza a una gran velocidad. El inconveniente que presenta la necesaria conectividad a Internet para utilizar éste servicio no es una traba ya que para poder gestionar el hogar digital es necesario en todo momento conectividad a Internet, esto no es problema a día de hoy.

#### 7.3.1.9 CONTROL DE ELEMENTOS INFRARROJOS

Dado el número de dispositivos que existen en los hogares que cuentan con elementos infrarrojos para su control como televisiones, aire acondicionados o minicadenas. En éste proyecto se abarca la simulación de todos estos dispositivos infrarrojos.

El hardware necesario para esto es un receptor infrarrojo, un emisor, un transistor y una resistencia.

El software desarrollado en el Raspberry Pi cuenta con los métodos necesarios para que un usuario de un hogar pueda seleccionar desde la aplicación Android la opción de un grabado de mando a distancia que envía al usuario a un pequeño tutorial de cómo hacerlo. Se tarda en realizar el grabado aproximadamente 2 minutos. Posteriormente el usuario ya tendrá agregado el nuevo mando a distancia que podrán utilizar todos los integrantes del hogar.

Esto se consigue gracias al software Linux Infrared Remote Control (LIRC, 1999) y al software desarrollado en éste proyecto, que interactuando con LIRC graba los pulsos infrarrojos que emite el mando a distancia, generando un fichero con los pulsos altos y bajos para ser posteriormente transmitidos en el momento en el que el usuario lo desee.

#### 7.3.1.10 DATOS EN TIEMPO REAL

Los presentes servicios existentes no permiten mostrar los cambios de estados de los dispositivos del hogar en tiempo real. Con éste servicio se ha solventado esta limitación. El usuario puede comprobar en directo los cambios producidos en el hogar, sin necesidad de recarga. También se informa al usuario de quién ha sido el autor del cambio. Esta información es de gran ayuda para los clientes ya que así tienen un control y conocimiento total.

La conexión establecida entre servidor y cliente es la que hace esto posible, como anteriormente ha sido mencionado es una conexión TPC que permite el envío de datos asíncronos.

#### 7.3.1.11 MÚLTIPLES SESIONES DE USUARIO

A fin de mejorar el servicio proporcionado para el usuario se incluye la metodología necesaria en el lado del servidor y del cliente para que un usuario tenga la posibilidad de tener registrada su cuenta en distintos dispositivos para que así la entrada a la aplicación sea mucho más fácil.

Para poder proporcionar éste servicio al usuario es necesario conocer las características del Smartphone del cliente.

Todos los Smartphone tienen un id único de dispositivo. Gracias a esto es posible que se pueda asociar la cuenta de un cliente al id del teléfono inteligente y almacenarlo en una base de datos.

La asociación anterior permite al servidor, gracias a unos métodos previamente desarrollados, enviar los datos de sesiones al usuario para que éste pueda identificar las sesiones que tiene activas y decidir si quiere cerrar alguna de ellas o no.

#### 7.3.1.12 SEGURIDAD DE USUARIO

La seguridad de la comunicación y métodos de seguridad para los usuarios es un apartado imprescindible para éste proyecto.

La comunicación prestada por los servicios anteriormente mencionados no hace uso de herramientas de seguridad. En éste proyecto se consiguen los siguientes métodos.

#### *SEGURIDAD DEL HOGAR*

Un usuario que intenta registrarse en el servicio para poder empezar a utilizar la gestión remota de un hogar debe conocer el identificador del hogar que es único para cada uno y un código de seguridad que es establecido y comunicado al propietario del hogar por la empresa instaladora.

Sin éste código el usuario no puede registrarse en el servicio. Esto hace que sólo los clientes confiables por el propietario del hogar puedan acceder al servicio.

#### *SEGURIDAD DE CLIENTE*

Para poder tener acceso al servicio tanto el usuario como el Raspberry Pi tienen que validarse con el servidor mediante usuario y contraseña, que es configurable por el usuario. Esto incluye más seguridad que los servicios existentes, ya que, con dichos servicios sólo es necesario establecer una conexión con la pasarela KNX del hogar. Conociendo los datos necesarios para el reenvío de puertos esto es posible y es una brecha de seguridad bastante cuestionable.

## *USUARIO MAESTRO*

Se ha querido dar un plus de seguridad para la gestión de usuarios por parte del propietario del hogar que se explica a continuación.

En el momento de instalación del servicio se establece un usuario root del hogar que pretende ser el titular de la propiedad. Puede realizar las siguientes funciones:

1. Cuando un usuario realiza un registro en el hogar se tendrá que esperar para tener acceso al servicio hasta que el usuario root del hogar lo permita. Cuando se realiza el registro se envía al usuario root un email y una notificación de aplicación de tipo nuevo registro en el hogar con los datos del nuevo usuario. El usuario root puede aceptar al usuario para poder empezar a utilizar el servicio o cancelar su suscripción.
2. Dado que se ha creado un módulo de seguridad en la aplicación Android que incluye la vista de los distintos usuarios que tienen acceso al hogar, el usuario root puede echar desde aquí al usuario que él considere oportuno.

Con esto se consigue un total conocimiento y seguridad de acceso de usuarios para el hogar.

## *PIN DE APLICACIÓN*

Un usuario puede establecer un pin de aplicación para acceder a esta. Con esto se consigue que, ante una pérdida del Smartphone o sustracciones de éste, no sea accesible la entrada a la aplicación y con ello al control domótico.

### *7.3.1.13 SEGURIDAD DE LA COMUNICACIÓN*

Con éste servicio tenemos información sensible, ya que se realizan transmisiones de datos como usuario y contraseña, acciones a realizar en los hogares e imágenes en directo del interior del hogar.

El servicio realiza una comunicación segura entre todos los dispositivos que son desarrollados. Se han implementado los siguientes métodos de seguridad:

## *INTERCAMBIO DE CERTIFICADOS DIGITALES*

Para la seguridad por parte de usuario ha sido realizado un certificado Secure Sockets Layer (SSL) para garantizar la identidad del servidor. Éste es enviado por el servidor y comprobado por la aplicación Android y el Raspberry pi en cada conexión realizada. La conocida empresa certificadora StartCom (StartCom, 2005), es la entidad de confianza para la totalidad del servicio. Si un dispositivo Android o el Raspberry pi consideran errónea la autenticidad del certificado recibido, no se completará la conexión.

Con esto se consigue autenticar al servidor y verificar la identidad de éste.

## *CIFRADO DE LA COMUNICACIÓN*

Como se ha mencionado anteriormente, en el servicio que se proporciona se intercambia información sensible. Se han tomado las medidas necesarias para mantener segura la comunicación.

Todo el intercambio de información es cifrado mediante un estándar de criptografía simétrica que se conoce como Advanced Encryption Standard (AES).

AES está clasificado por la Agencia de Seguridad Nacional, National Security Agency (NSA), de los Estados Unidos para la seguridad más alta de información secreta.

Las claves de cifrado para éste cifrador pueden ser de 128, 192 y 256 bits. Se han implementado las claves de cifrado de 256 bits para una mayor seguridad.

AES realiza un cifrado por bloques y como tal se puede implementar con distintos modos de operación. Para dicho proyecto se ha utilizado en modo de operación Cipher-block chaining (CBC).

Éste modo de operación es fuerte frente a ataques de sustitución y repetición. La desventaja de éste es la necesidad de realizar el cifrado de forma secuencial (no puede ser paralelizado), lo que significa que el descifrado se realiza bloque a bloque, lo cual no es un problema. Gracias al poder computacional de hoy día no es apreciable el retardo.

### *MÉTODO SEGURO DE INTERCAMBIO DE CLAVES*

En el anterior apartado se ha descrito el modo de cifrado simétrico del servicio. Para que sea totalmente seguro, el intercambio de claves del cifrado simétrico debe serlo también. Si no es así y las claves son enviadas como texto claro podrían ser interceptadas por algún atacante.

Para solventar éste problema y conseguir un servicio seguro, el intercambio de claves se realiza mediante Rivest, Shamir y Adleman (RSA) en el momento de la conexión.

RSA es un sistema criptográfico de clave pública desarrollado en 1977. Es el primer y más utilizado algoritmo de éste tipo y es válido tanto para cifrar como para firmar digitalmente. RSA es uno de los sistemas de cifrado asimétricos más exitosos en la actualidad.

En cuanto a las longitudes de claves, el sistema RSA permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits. Para éste servicio la longitud de clave utilizada es de 2048 bits, lo cual dado el poder de computación actual es una longitud de clave computacionalmente segura.

## 7.4 DISEÑO

A continuación, serán descritos los distintos módulos que han sido implementados en los tres dispositivos que forman la totalidad del servicio, los cuales son: Raspberry Pi, dispositivo Android y servidor.

| Raspberry Pi  | Dispositivo Android  | Servidor  |
|---|--|---|
| <ul style="list-style-type: none"><li>• Login</li><li>• Monitorización bus KNX</li><li>• Contro por voz</li><li>• Recepción de datos</li><li>• Control de elementos infrarrojos</li></ul> | <ul style="list-style-type: none"><li>• Login</li><li>• Registro</li><li>• Inicio</li><li>• Confort</li><li>• Climatización</li><li>• Seguridad</li><li>• Multimedia</li><li>• Configuración</li></ul> | <ul style="list-style-type: none"><li>• Gestión de conexiones</li><li>• Base de datos</li><li>• Comunicación segura</li></ul> |

El orden en el que serán descritos es el siguiente.

- Raspberry Pi.
- Dispositivos Android.
- Servidor.

### 7.4.1 RASPBERRY PI

En el lado del Raspberry Pi el software desarrollado para alcanzar el fin de éste proyecto ha sido implementado en el lenguaje de programación Python versión 3.4.

#### 7.4.1.1 LOGIN

Éste realiza conexiones con el servidor para poder enviar eventos producidos en el hogar y esperar a entrada de datos o peticiones de usuario para poder ejecutarlas. Se ha desarrollado un Login seguro con el servidor para alcanzar la seguridad que en los requisitos es descrita. El diagrama de flujo de la autenticación desde el lado del Raspberry Pi puede verse en los planos Figura 68. Login Raspberry Pi.

En éste diagrama de flujo anterior se visualiza claramente la traza del programa cuando éste es ejecutado. También se observa que tras el Login éste comienza con todos los procesos necesarios implementados para alcanzar el comportamiento final esperado.

#### 7.4.1.2 MONITORIZACIÓN BUS KNX

Como se observa en el diagrama de flujo de la Figura 1 es el encargado de la monitorización del bus KNX para tener un informe actualizado de los dispositivos del hogar. Como se ha expresado anteriormente la monitorización se realiza gracias a las librerías EIBD que transforman los telegramas KNX a mensajes legibles. Esta información permite conocer todos los estados de los dispositivos. El diagrama de flujo de la monitorización desde el lado del Raspberry Pi puede verse en los planos Figura 69. Monitorización KNX.

#### 7.4.1.3 CONTROL POR VOZ

Como también es observado en el diagrama de flujo de la Figura 68. Login Raspberry Pi se ha desarrollado la metodología necesaria para poder realizar cambios en el hogar a través de la voz por medio de micrófonos. Si las palabras clave son mencionadas el Raspberry Pi realizara la acción requerida en el dispositivo mencionado. El diagrama de flujo de este puede verse en la Figura 70. Control por voz. Las claves para poder accionar un elemento por medio de micrófonos son los siguientes.

El usuario para accionar el modulo tiene que decir la palabra “casa”. Tras esta tendrá que seguir como se describe a continuación.

- Para activar un dispositivo de tipo luz, electrodomésticos y climatización el usuario tiene que decir las siguientes palabras.
  - Encendido.

Para el encendido las palabras clave son: encender, encendido, enciende. Tras esto el usuario debe decir el nombre del dispositivo que desea encender. Si tras decir el nombre se comprueba que el dispositivo es de tipo regulado se enviara un telegrama EIS6 y el usuario deberá decir un número entre 0 y 100 con el que desea que el dispositivo sea activado. Si no se encuentra ningún número en la cadena de texto se activará el dispositivo con el mayor valor posible, en éste caso será 100 para luces. El valor para climatización serán 22 grados. Si el dispositivo es de tipo EIS1 enviara un telegrama de éste tipo con valor 1.

- Apagado.

Para el apagado las palabras clave son: apaga, apagar y apagado. Cuando esta clave es recibida el programa comprobara que tipo de dispositivo. Si es de tipo regulado enviara un telegrama EIS6 con el valor 0. Si el dispositivo es de tipo EIS1 enviara un telegrama de éste tipo con valor 0.

- Para subir, bajar o parar un dispositivo de tipo persianas el usuario tiene que decir las siguientes palabras.
  - Subir persiana.

Para subir persianas las palabras clave son: sube y subir.

- Bajar persiana.

Para bajar persianas las palabras clave son: baja y bajar.

Con todo lo anteriormente mencionado el formato para accionar dispositivos por medio de voz es el siguiente.

|   |
|---|
| casa [acción] [nombre del dispositivo] [número] |
|---|

#### 7.4.1.4 RECEPCIÓN DE DATOS

Como también es observado en el diagrama de flujo de la Figura 68. Login Raspberry Pi el programa también queda a la espera de recibir datos del servidor. Estos datos pueden provenir del servidor de aplicación o desde algún cliente para realizar un cambio en el hogar. En el diagrama de flujo Figura 71. Recepción de datos se expresa la traza que sigue el programa cuando el Raspberry Pi recibe datos. Para mejorar la comprensión y conseguir una mejor visualización del diagrama se ha dividido en 3 partes. Los cuales pueden verse en los planos y son referenciados como: Figura 71. Recepción de datos, Figura 72. Recepción de datos uno y Figura 73. Recepción de datos dos.

El diagrama el flujo Figura 71. Recepción de datos se visualiza la traza que realiza el programa ante entrada de datos. En éste se visualiza la mayoría de la traza que sigue cuando son recibidos datos enviados por un cliente. En éste queda por describir el punto uno y dos los cuales son expuestos en la Figura 72. Recepción de datos uno y Figura 73. Recepción de datos dos.

En el diagrama de flujo de la Figura 73. Recepción de datos dos se visualiza la parte que faltaba de entrada de datos enviados por el usuario. Éste hace referencia a las opciones de los elementos infrarrojos del hogar.

#### 7.4.1.5 CONTROL DE ELEMENTOS INFRARROJOS

Para el control de los elementos infrarrojos del hogar hago uso de receptores y emisores infrarrojos. Éste control es posible gracias a los elementos infrarrojos instalados en el Raspberry Pi. El esquema de conexiones en el Raspberry Pi puede verse en los planos, Figura 74. Esquema conexión infrarrojo.

Los modelos empleados son:

- Receptor TSOP38238.
- Emisor IR333C.

Gracias al receptor infrarrojo se habilita la posibilidad de realizar lecturas de nuevos mandos a distancia por parte de los usuarios.

A través del emisor infrarrojo se replican los pulsos previamente leídos de los dispositivos remotos deseados en la aplicación.

#### 7.4.2 DISPOSITIVO ANDROID

En el lado del cliente ha sido desarrollada la aplicación Android para alcanzar el fin de éste proyecto, el cual es poder monitorizar el estado del hogar y realizar las acciones deseadas por los usuarios. En la estructura construida en dicho proyecto éste es el dispositivo que permite interactuar con el hogar y realizar todas las opciones disponibles.

#### 7.4.2.1 LOGIN

Éste realiza conexiones con el servidor para poder enviar eventos producidos por la interacción del usuario con la aplicación Android. Se ha desarrollado un Login seguro con el servidor para alcanzar la seguridad que en los requisitos es descrita. El diagrama de flujo de la autenticación desde el lado del dispositivo Android puede verse en los planos, Figura 75. Login Android.

#### 7.4.2.2 REGISTRO

En éste apartado un usuario no registrado en el servicio proporcionado podrá registrarse en éste para la gestión domótica remota de su hogar.

#### 7.4.2.3 INICIO

En la pantalla de inicio de la aplicación se visualizarán de manera clara los siguientes módulos:

- Confort.
- Climatización.
- Seguridad.
- Multimedia.
- Estado de la conexión con el servidor.

#### 7.4.2.4 CONFORT

En esta pantalla están disponibles las opciones referentes al confort domótico:

- Luces: permite el control remoto de las luces del hogar y a su vez, la visualización del estado de estas.
- Persianas: permite el control remoto de las persianas.
- Electrodomésticos: permite el control remoto de los electrodomésticos del hogar y a su vez, la visualización del estado de estos.
- Escenas: permite la programación de escenas KNX en los dispositivos del hogar.

#### 7.4.2.5 CLIMATIZACIÓN

En esta pantalla están disponibles las opciones referentes a los dispositivos KNX de climatización:

- Calefacción: permite el control remoto de la calefacción del hogar y el estado de ésta.
- Aire acondicionado: permite el control remoto del aire acondicionado del hogar y el estado de éste.

#### 7.4.2.6 SEGURIDAD

En éste apartado de la aplicación se visualizan las opciones de seguridad del hogar domótico.

- Imágenes: a través de éste apartado se pueden realizar descargas de imágenes de las distintas cámaras conectadas al Raspberry Pi por petición del usuario.
- Modo Vacaciones: en éste apartado se contempla la posibilidad de activar el modo vacaciones, el cual si está activado se encarga del encendido y apagado de las distintas luces del hogar para simular que hay personas en casa.

#### 7.4.2.7 MULTIMEDIA

En éste apartado se contemplan las opciones referentes a la tecnología multimedia del hogar.

- Mando a distancia: en esta pantalla se pueden simular los elementos infrarrojos emisores de los siguientes dispositivos: Televisión, Aire Acondicionado y dispositivos musicales como minicadenas. La inclusión de nuevos controles remotos será clara y accesible para el usuario.

#### 7.4.2.8 CONFIGURACIÓN

En éste apartado de la aplicación se pueden distinguir las opciones de configuración de usuario.

- Configuración de usuario: Contempla la visualización del nombre de usuario y el correo registrado para el servicio.
- Seguridad de usuario: en éste apartado se pueden distinguir varios subapartados:
  - Contraseña de usuario: en éste apartado se podrá cambiar la contraseña vinculada al usuario.
  - Pin de seguridad: la aplicación requerirá de un pin de seguridad, establecido por el usuario, cada vez que se inicia la aplicación para poder acceder a la gestión y control remoto doméstico.
  - Sesiones activas: dada la opción de éste servicio de multisesión, un usuario puede tener activas varias sesiones en distintos dispositivos. Desde éste apartado se pueden visualizar y eliminar las distintas sesiones.
  - Usuarios en casa: en dicho apartado se visualiza la totalidad de los usuarios que hay registrados en el hogar.
- Notificaciones: éste apartado dota a la aplicación de una configuración para las siguientes opciones ante notificaciones de ésta:
  - Activar/Desactivar led de notificaciones.
  - Color de led.
  - Activar/Desactivar sonido de aplicación.
  - Activar/Desactivar vibración de aplicación.
- Cambio de imagen de perfil: los usuarios pueden establecer una imagen de perfil. Desde aquí se podrá cambiar la imagen de usuario.

### 7.4.3 SERVIDOR

Para poder servir todas las funcionales que el servicio pretende prestar es necesario de un dispositivo público que permita recibir conexiones de otros dispositivos en todo momento. Éste elemento de la topología descrita en la Figura 2. Esquema de comunicación es el servidor desarrollado. Está escrito en Java y haciendo uso de base de datos es el encargado de gestionar todas las comunicaciones.

- Acepta conexiones de los Raspberry Pi.
- Acepta conexiones de los dispositivos Android.
- Éste se autentica frente a los clientes con un certificado SSL firmado por la Autoridad certificadora StartSSL.
- Autentica usuarios por medio de usuario/email y contraseña.
- Autentica Raspberry Pi a través de id y contraseña.
- Realiza los reenvíos de mensajes del protocolo entre clientes/Raspberry.
- Almacena los estados de los dispositivos de las viviendas.
- Cifra toda la comunicación.
- Gestiona usuarios pertenecientes a los hogares.
- Gestiona sesiones de usuarios.
- Hace uso de la API de Firebase para el envío de mensajes importantes para los usuarios.
- Realiza envíos de email de confirmación.

La autenticación contra el servidor es el primer paso que los demás dispositivos de la estructura realizada deben hacer para optar a los beneficios que presta éste proyecto. El diagrama de flujo de la autenticación de Raspberry Pi y de usuarios puede verse en los planos, Figura 76. Autenticación clientes.

Como se observa en la Figura 76. Autenticación clientes se realiza un Login seguro para las dos partes del servicio. Se observa también como se produce el intercambio de claves para el cifrado simétrico.

Las claves de cifrado simétrico se intercambian en la primera autenticación realizada por los clientes. Para los posteriores inicios se observa que los pasos de intercambio de claves no son realizados, éste proceso se realiza de tal forma ya que hace que los inicios sean mucho más rápidos.

En la Figura 77. Recepción datos servidor de los planos se muestra el diagrama de flujo del servidor cuando el usuario ya está conectado y autenticado. Éste se realiza por bloques ya que exponer el diagrama completo dificultaría su comprensión. Tampoco se ha incluido el bloque de cifrado de mensajes cuando el servidor envía datos para la reducción de éste, todos los datos enviados son cifrados con la clave simétrica del cliente.

En la Figura 77. Recepción datos servidor se muestra todos los tipos de mensajes que el servidor puede recibir y el modelo de descifrado que realiza. Su procesamiento es descrito en los siguientes diagramas.

En el diagrama de flujo de la Figura 78. Recepción datos servidor uno se observa el procesamiento de mensajes que se realiza ante mensajes de tipo ACK, power, dimmer y scene.

El punto uno del diagrama de la Figura 79. Recepción datos servidor dos indica la continuidad de la Figura 78. Recepción datos servidor uno. En el diagrama de la Figura 79. Recepción datos servidor dos de flujo se observa el procesamiento de mensajes que realiza el servidor ante mensajes de tipo change, remote, ping y requestdevices.

El punto dos del diagrama de flujo de la Figura 80. Recepción datos servidor tres indica la continuidad de la Figura 79. Recepción datos servidor dos. En el diagrama de flujo de la Figura 80. Recepción datos servidor tres se observa el procesamiento de mensajes que realiza el servidor ante mensajes de tipo camera, config, secure y shutdown.

Como se comprueba el servidor es el encargado de toda la comunicación del servicio.

## 7.5 PROTOCOLO DE COMUNICACIONES

### 7.5.1 TIPOS DE MENSAJES

Todos los mensajes constan de un campo type, value y idusersend. El campo type es el que hace posible la identificación de los mensajes apoyado en el campo value y idusersend. Los tipos de mensajes implementados para la realización de la totalidad del servicio son los siguientes.

- newlogin. Mensaje de nueva autenticación.
- login. Mensaje de autenticación.
- shutdown.
- closesession.
- refresh. Mensaje de refresco periódico de datos.
- requestdevices. Mensaje de petición de lista de dispositivos del hogar.
- power. Mensaje de encendido/apagado.
- dimmer. Mensaje de regulación.
- scene. Mensaje de escenas.
- secure. Mensaje de seguridad.
- alarm. Mensaje de seguridad.
- ack.
- change. Mensaje de cambio producido en el hogar.
- camera. Mensaje de cámaras.
- config. Mensaje de configuración.
- ping. Mensaje de sondeo de conexión.
- deviceshome. Mensaje de envió de dispositivos.
- updateconnection. Mensaje de información de conexión.
- remote. Mensaje de controles remotos.

En todos los tipos de mensaje existen campos comunes. Son los siguientes.

- typeuser: Campo en el que se inserta el tipo de usuario que envía el mensaje. Tiene 3 posibles valores: server, user, raspi.
  - user: identifica a un usuario.
  - server: identifica al servidor.
  - raspi: identifica al Raspberry Pi.
- iduser. Campo que identifica al dispositivo que envía los datos. Éste campo posee el valor 0 si el mensaje es enviado por el servidor. En caso de ser enviado por el Raspberry Pi contendrá el id del Raspberry Pi. Para los usuarios contiene una cadena la cual está compuesta por los siguientes campos:

|                                  |
|----------------------------------|
| id_de_usuario-phone-id_de_sesión |
|----------------------------------|

- idusersend. Campo que identifica el receptor del mensaje. En caso de enviar
- time. Marca de tiempo en formato Día/Mes/Año-Hora:Minuto:Segundo.
- idmsg. Id del mensaje. Dato de tipo numérico.
- type: identifica el tipo de mensaje. Los posibles valores de éste campo son los mencionados anteriormente.

Para exponer el protocolo de intercambio de datos por claridad se obvian los campos anteriormente mencionados. Todos los mensajes excepto el mensaje de tipo Login o Newlogin que son cifrados por medio de RSA son enviados cifrados con el cifrador de bloque AES con un tamaño de clave de 256 bits. Tras cifrar los datos son codificados en base64 y enviados.

Antes de empezar para una mejor comprensión se muestra una tabla con el valor que se ha establecido a cada tipo de dispositivo para su identificación.

### **Tipos de dispositivos.**

- Luces = 1. bits: 0001.
- Persianas = 2. bits: 0010.
- Electrodomésticos = 3. bits: 0011.
- Televisión = 4. bits: 0100.
- Aire acondicionado = 5. bits: 0101.
- Calefacción = 6. bits: 0110.
- Sensor presencia = 7. bits: 0111.
- Sensor humo = 8. bits: 1000.
- Sensor humedad = 9. bits: 1001.
- Cámaras = 10. bits: 1010.
- Control Remoto = 11. bits: 1011.
- Música = 12. bits: 1100.
- Sensor magnético = 14. bits: 1110.
- Sensor Aire = 16. bits: 00010000.
- Todos = 13. bits: 1101.

### **Tipos de tramas/acciones.**

- EIS1: Interruptor (KNX). 1 bit.
- EIS6: Valor relativo, regulación (KNX). 1 byte. Valores entre 0 y 255.
- IRAIR.
- IRMUSIC.
- IRTV.
- ALARM

## 7.5.2 DISPOSITIVO ANDROID

### 7.5.2.1 NEWLOGIN

Éste mensaje indica al servidor que es el primer inicio de sesión, por tanto, los dispositivos tendrán que realizar el intercambio de clave simétrica. Éste mensaje es cifrado mediante RSA. El formato del mensaje es el siguiente.

```
{ "typeuser": "user", "idusersend": "", "nameuser": "vallemar", "pass": "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92", "time": "", "type": "newlogin" }
```

Valores de los campos:

- nameuser: éste campo contiene el nombre de usuario o el email que el usuario ha introducido en el formulario de inicio.
- pass: campo contiene la contraseña introducida por el usuario. La contraseña es un hash SHA-256 formateado como hexadecimal con relleno a la izquierda.

Si la autenticación del cliente es correcta el usuario enviara su clave pública RSA para que el servidor cifre la clave simétrica con su clave pública. El formato de envío es el siguiente.

```
{ "typeuser": "", "idusersend": "", "time": "", "type": "login", "value": "newlogin", "modulus": "ALiGzCSjtpTqH5AJk6+t0Hp+BAQffxwUbJcekLnIq9mXE2FX4VoWD3v+ObF5Sex9jvN08KGdCbMEwEH6u6wqKf1qguJJ2DG4LTwQ6TPGpw1If7td1+ul2tzqN99F5GYc08wxGryCMrmmCZKzbnqWzJKsDIBn4DGhHtzNerJxBohoRLUp9BzVMf+edXYnKmmJh1m4mw/vz1yulCRX4dzOR7LbTaUzHnzRIQOVL8I5Kfi4CKHAUIqUkC95VDQ+rZ/kEdirVRNwNSTWwEas/bcU+0uGkqykDx7+9IsLevfM0ItkXI/8+8TEOxdVxczcSUKIQopFJiNK7S4qP645sHM/0U=", "exponent": "AQAB" }
```

### 7.5.2.2 LOGIN

Éste mensaje indica al servidor que el usuario ya tiene una clave simétrica, por tanto, se evita el mecanismo de intercambio de claves, es cifrado mediante RSA. El formato del mensaje es el siguiente.

```
{ "typeuser": "user", "nameuser": "vallemar", "pass": "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92", "time": "", "type": "login" }
```

Valores de los campos:

- nameuser: éste campo contiene el nombre de usuario o el email que el usuario ha introducido en el formulario de inicio.
- pass: campo contiene la contraseña introducida por el usuario. La contraseña es un hash SHA-256 formateado como hexadecimal con relleno a la izquierda.

Si la autenticación del cliente es correcta el usuario enviara sus datos de sesión con el siguiente formato.

```
{ "typeuser": "", "time": "", "type": "login", "idfirebase": "clPu2Biz7HA:APA91bFyK-
WHDlPDOtEF88YrEU-
_LodoPv5d4vWGzf6DaQcvAQcfRusinJsMo3VLC0lBJEymAd_oZ0ByCAVAi2ydN7c
PYn7E2m0RVXlJa35wz6S-B3uWa8vJ8kTtfyc-
acwESmOLcX3T", "modeldevice": "Android SDK Built For
X86", "iddevice": "6a96b85ef2c1d49d", "os": "7.0", "typedevice": "phone", "idmsg": }
```

### 7.5.2.3 SHUTDOWN

Éste mensaje indica al servidor que el usuario desea hacer una de las dos siguientes opciones, desconectarse o cerrar la sesión actual. Éste es identificado por el campo value. El cual puede tener los siguientes valores.

Posibles opciones del campo value:

- Value = disconnect. El dispositivo Android indica el cierre de la comunicación. Causas: borrado de memoria en el dispositivo o salir de la app durante más de un minuto.
- Value = closesession. Indica que el usuario ha cerrado la sesión del dispositivo.

### MENSAJE DE DESCONEXIÓN

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "shutdown", "value": "disconn
ect", "idmsg": }
```

### MENSAJE DE CIERRE DE SESIÓN

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "",
"type": "shutdown", "value": "closesession", "idmsg": }
```

### 7.5.2.4 REQUEST DEVICES

Éste mensaje es enviado por el dispositivo Android al servidor o al Raspberry Pi indicando que desea la descarga de un tipo de dispositivo o la totalidad de ellos. El formato del mensaje es el siguiente.

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "requestdevices", "value": "", "
idmsg": }
```

Posibles valores:

- Posibles opciones del campo value:
  - value = 1. Indica que el cliente está pidiendo al servidor los dispositivos de tipo luces del hogar.
  - value = 2. Indica que el cliente está pidiendo al servidor los dispositivos de tipo persianas del hogar.
  - value = 3. Indica que el cliente está pidiendo al servidor los dispositivos de tipo electrodomésticos del hogar.
  - value = 5. Indica que el cliente está pidiendo al servidor el aire acondicionado de tipo KNX.

- value = 6. Indica que el cliente está pidiendo al servidor la calefacción de tipo KNX.
- value = 10. Indica que el cliente está pidiendo al Raspberry Pi las cámaras conectadas. El servidor identifica que el mensaje no va para el por el campo idusersend, el cual lleva el identificador del Raspberry Pi.
- value = 11. Indica que el cliente está pidiendo al servidor todos los mandos a distancia disponibles.
- value = 13. Indica que el cliente está pidiendo al servidor todos los dispositivos del hogar disponibles.
- Posibles opciones del campo idusersend.
  - Idusersend = 0. Mensaje enviado cuando el destinatario es el servidor. Éste es quien debe contestar.
  - Idusersend = id de Raspberry Pi. Mensaje enviado cuando el destinatario es el Raspberry Pi. Éste es quien debe contestar.

#### 7.5.2.5 POWER

Éste mensaje es enviado por el dispositivo Android al servidor y reenviando por el servidor al Raspberry Pi indicando que desea accionar alguno de los elementos KNX del hogar. Éste tipo de mensajes es empleado para dispositivos que escuchan telegramas EIS1 y EIS2. El formato del mensaje es el siguiente.

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"power","value":"","iddevice":"","typedevice":"","address":"","idmsg:"}
```

Posibles valores de los campos:

- Campo value:
  - Value = 0. Indica apagar un elemento KNX. Para persianas, subir persiana.
  - Value = 1. Indica encender un elemento KNX. Para persianas, bajar persiana.
- Campo iddevice:
  - Éste campo identifica el identificador del dispositivo.
- Campo typedevice:
  - Identifica el tipo de dispositivo. Sus valores son iguales a los explicados en el tipo de mensaje requestdevice.
- Campo address:
  - Dirección KNX del dispositivo que se desea encender/apagar.

### 7.5.2.6 DIMMER

Éste mensaje es enviado por el dispositivo Android al servidor y reenviando por el servidor al Raspberry Pi indicando que desea regular alguno de los elementos KNX del hogar. Éste tipo de mensajes es empleado para dispositivos que escuchan telegramas EIS6. El formato del mensaje es el siguiente.

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"dimmer","value":"","iddevice":"","address":"","typedevice":"","idmsg":}
```

Posibles valores de los campos:

- Campo value:
  - Éste soporta un rango alto de valores, entre 0 y 100. Los dispositivos EIS6 soportan un rango de valores entre 0 y 255. Para realizar el cambio se normaliza y se convierte en hexadecimal.
- Campo iddevice:
  - Éste campo identifica el identificador del dispositivo.
- Campo typedevice:
  - Identifica el tipo de dispositivo. Sus valores son iguales a los explicados en el tipo de mensaje requestdevice.
- Campo address:
  - Dirección KNX del dispositivo que se desea encender/apagar.

### 7.5.2.7 SCENE

Éste mensaje es enviado por el dispositivo Android al servidor indicando que desea alguna de las opciones disponibles de escenas. Éste tipo de mensajes tiene distinto formato y son identificados en el servidor por el campo value. Los mensajes que pueden ser enviados son los siguientes.

- Campo value:
  - Value = 1. Indica que es una escena de encendido. Mensaje de creación de escena.
  - Value = 0. Indica que es una escena de apagado. Mensaje de creación de escena.
  - Value = history. Pide al servidor las escenas creadas.
  - Value = change. Cambia estado de una escena (activa/no activa).
  - Value = actívale. Activar escena en el momento.
  - Value = delete. Elimina una escena.

### *MENSAJE DE CREACIÓN DE ESCENA*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "scene", "value": "", "typescene": "", "hour": "", "namescene": "", "devices": [{"iddevice": "", "action": "", "typedevice": ""}], "days": [{"day": ""}], "idmsg": }
```

Posibles valores de los campos:

- Typescene: campo identificador del tipo de dispositivos que entran a formar parte de la nueva escena. Valores: 1 luces, 2 persianas, 3 electrodomésticos y 13 todo tipo de dispositivos.
- Hour: indica la hora a la que el usuario pide que se encienda la escena.
- Namescene: cadena de texto identificativa impuesta por el usuario.
- Days: Array JSON en el que se encuentran los días de la semana que la escena estará activa. Los valores son de 1 a 7 donde 1 es lunes y 7 domingo.
- Devices: Array JSON en el que se encuentran los siguientes campos.
  - Iddevice: id de dispositivo. Identifica los dispositivos que entran en el conjunto de dispositivos de la escena.
  - Action: acción KNX a realizar en el dispositivo.
  - Typedevice: tipo de dispositivo.

### *MENSAJE DE PETICIÓN DE ESCENAS*

```
{ "typeuser": "", "iduser": "", "idusersend": "0", "time": "", "type": "scene", "value": "history", "typedevice": "", "idmsg": }
```

- Typedevice: indica el tipo de historial de escena que el usuario pide al servidor. Se identifica mediante el tipo de dispositivo. Valores: 1 luces, 2 persianas, 3 electrodomésticos y 13 todo tipo de dispositivos.

### *MENSAJE DE CAMBIO DE ESTADO DE ESCENA (ACTIVA/NO ACTIVA)*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "scene", "value": "change", "idscene": "", "state": "", "idmsg": }
```

- idscene: identifica el id de escena a realizar el cambio.

### *MENSAJE DE ACTIVAR ESCENA*

```
{ "typeuser": "user", "iduser": "1000-phone-246", "idusersend": "0", "time": "04/02/2017-23:23:17", "type": "scene", "value": "activate", "idscene": "56", "idmsg": "855234" }
```

- idscene: identifica el id de escena a activar.

### *MENSAJE DE ELIMINACIÓN DE ESCENA*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "scene", "value": "delete", "idscene": "", "idmsg": }
```

- idscene: identifica el id de escena a realizar el cambio.

#### 7.5.2.8 SECURE

Éste mensaje es enviado por el dispositivo Android al servidor indicando que desea alguna de las opciones disponibles de seguridad. Éste tipo de mensajes tiene distinto formato y son identificados en el servidor por el campo value. Los mensajes que pueden ser enviados son los siguientes.

- Campo value:
  - Value = mode. Indica el cambio de modo de seguridad del hogar.
  - Value = history. Indica un mensaje de petición de historial de alarmas producidas en el hogar.
  - Value = holidays. Indica el cambio de estado del modo vacaciones.

#### *MENSAJE DE CAMBIO DEL MODO SEGURIDAD*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"secure","value":"mode","state":"","idmsg:"}
```

Posibles valores de los campos:

- state: Indica en el estado en el que se desea poner el hogar. Valores: 0 apagado modo seguro, 1 encendido modo seguro.

#### *MENSAJE DE PETICIÓN DE HISTORIAL DE ALARMAS*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"secure","value":"history","idmsg:"}
```

#### *MENSAJE DE MODO VACACIONES*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"secure","value":"holidays","state":"","idmsg:"}
```

Posibles valores de los campos:

- state: Indica en el estado en el que se desea poner el hogar. Valores: 0 apagado modo holidays, 1 encendido modo holidays, “state” es un mensaje de petición para saber si el modo vacaciones está activado.

Este mensaje es enviado al Raspberry Pi.

#### 7.5.2.9 CAMERA

Éste mensaje es enviado por el dispositivo Android al servidor y reenviando por el servidor al Raspberry Pi indicando que desea realizar la captura de una imagen en el hogar. El formato del mensaje es el siguiente.

```
{"iduser":"","idusersend":"","time":"","typeuser":"","type":"camera","value":"capture","address":"","idmsg:"}
```

Posibles valores de los campos:

- Campo value:
  - Value = capture. Indica que el usuario desea realizar la captura de una imagen de una de las cámaras conectadas.
- Campo address:
  - Dirección de la cámara, esta había sido recibida tras enviar un mensaje requestdevice con valor 10.

#### 7.5.2.10 CONFIG

Éste mensaje es enviado por el dispositivo Android al servidor indicando que desea realizar alguna de las opciones disponibles en el módulo de configuración. Éste tipo de mensajes tiene distinto formato y son identificados en el servidor por el campo typeconfig y configuser. Los mensajes que pueden ser enviados son los siguientes.

- Campo typeconfig:
  - typeconfig = configuser. Indica que el mensaje es destinado a cambiar la configuración del usuario.
  - typeconfig = confighome. Indica que el mensaje es destinado a cambiar la configuración del hogar. Éste mensaje no es utilizado, ha sido diseñado para posibles ampliaciones.
- Campo configuser:
  - configuser = password. Indica que es un mensaje de cambio de contraseña de usuario.
  - configuser = sessions. Indica que es un mensaje de petición de historial de sesiones.
  - configuser = connectedhome. Indica que es un mensaje de petición. Con éste se indica que se desea la lista de los usuarios conectados al hogar.
  - configuser = deleteuser. Indica que el usuario maestro o root del hogar desea eliminar a un integrante del hogar.
  - configuser = updatefirebase. Mensaje para actualizar el token de firebase.

#### *MENSAJE DE CAMBIO DE CONTRASEÑA*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "config", "typeconfig": "configuser", "configuser": "password", "value": "", "idmsg": }
```

- Value: contiene el hash SHA-256 de la nueva contraseña introducida.

#### *MENSAJE DE PETICIÓN DE SESIONES*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "config", "typeconfig": "configuser", "configuser": "sessions", "value": "", "idmsg": }
```

- Value: éste indica el tipo de sesiones que quiere ver el usuario. Posibles valores son:

- Current. Indica sesiones activas.
- All. Indica sesiones activas y antiguas.

#### *MENSAJE DE PETICIÓN DE USUARIOS CONECTADOS AL HOGAR*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"config","typeconfig":"configuser","configuser":"connectedhome","value":"","idmsg":}
```

- Value: éste indica el tipo de usuario conectados al hogar que desea obtener el cliente. Posibles valores son:
  - Current. Indica usuario activo.
  - All. Indica todos los usuarios.

#### *MENSAJE DE PETICIÓN DE BORRADO DE UN USUARIO DEL HOGAR*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"config","typeconfig":"configuser","configuser":"deleteuser","value":"4536","idmsg":}
```

- Value: contiene id de usuario que desea borrar el propietario del hogar.

#### *MENSAJE DE ACTUALIZACIÓN DE TOKEN FIREBASE*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"config","typeconfig":"configuser","configuser":"updatefirebase","iddevice":"","value":"","idmsg":}
```

- Value: contiene el nuevo token de firebase.
- iddevice: campo que contiene el identificador único de dispositivo.

#### *MENSAJE DE PETICIÓN DE CIERRE DE SESIÓN EN OTRO DISPOSITIVO*

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"config","typeconfig":"configuser","configuser":"closeessions","value":"763e50965d4e6a25","idmsg":}
```

##### *7.5.2.11 ACK*

Éste mensaje es enviado por el dispositivo Android al servidor indicando que ha sido recibido el mensaje enviado por el servidor. El campo value en éste indica el número de mensaje recibido.

```
{"typeuser":"","iduser":"","idusersend":"","time":"","type":"ack","value":"199","idmsg":}
```

##### *7.5.2.12 REMOTE*

Éste mensaje controla las acciones relacionadas con los elementos infrarrojos del hogar. Por su complejidad será explicado en los casos de uso de la aplicación.

## 7.5.3 RASPBERRY PI

### 7.5.3.1 NEWLOGIN

Éste mensaje indica al servidor que es el primer inicio de sesión, por tanto, los dispositivos tendrán que realizar el intercambio de clave simétrica. Éste mensaje es cifrado mediante RSA. El formato del mensaje es el siguiente.

```
{'time':, 'iduser':, 'idusersend': ", 'pass':  
'd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35', 'type':  
'newlogin', 'typeuser': 'raspi'}
```

Valores de los campos:

- iduser: contiene el id del Raspberry Pi.
- pass: contiene la contraseña del Raspberry Pi. La contraseña es un hash SHA-256 formateado como hexadecimal con relleno a la izquierda.

Si la autenticación del Raspberry Pi es correcta enviara su clave pública RSA para que el servidor cifre la clave simétrica con su clave pública. El formato de envío es el siguiente.

```
{"typeuser": "raspi", "modulus":  
"MjI2MDQ2ODE3MTY4OTkxNTUwNTczMzY3NDg1NjIyMjU3Mjg1OTIxNzc2MTg  
5NjA1OTQzMzcwOTI1NDM5NDIyNzIzNTQ2ODE1NTcxOTIwOTg1NjIwMjU5NDI5  
ODQyOTYzODE5ODczODkxNTM0NTk1NjI1MzkwMDk1MzM3OTE4Nzg1MjU1O  
DU4MjYyMzc3Njg0MDM0Mzc3NzU5MDU4ODU1OTUzODAxMTE4NjQxMTAyM  
zA4ODIzMTM3NDQ3OTcxMzEzMjc1NzYzODQyMzg0NjI3OTc3NDg4ODAyNTA4  
NTY2NzE4NTg0MjQxMDE4OTExMzQyMTQxODQzNTA2NTMzNDk2NjM3MDQ0  
OTM4OTQ3ODU3MTAzNDg1NjI3MTM5MTQzMtUxNzI2MDQ2OTgxNjMxNTQ2  
Mzc2NDc4MTg0NjkwMzI0NjU2NTM1NTg1Mjc2MjI3NjgwMjU1NjAyNzY2NjkwN  
Dc5MDgwOTQ2ODQ5MDc0MjczMTg0NDgxNTc2ODQ0NTAyMzgzNTQ5NzQ5ND  
c5MDc3MTk0ODE1NTE5MDU5MjQ3NTg4MjIyODcyOTI2MjE2MjM1MjE4MDA3  
Mzg2MjI4NzIwNzI5MDE2NjQ1NTk0NDM5MDI0NDQxMTY5ODcxNjEyODQwNj  
Y1MTIwNTU0NzI2OTA0Mjk0NTcyODI2NzY2MjQ1NTIxNzY4MzU1OTgzMDgxNj  
kzMTU5NDY2NDg1ODkzMjUzODE3MTIxNTA5ODcxODYwNzc2OTg1MDQ3ODU  
4NjA4MjgzNTE5MzkxMjA5NTE5NTM0MjI4NTc2NTE5NjEzMDAwNTUyODQ3M  
Tk1NzcyNzc=", "exponent": "NjU1Mzc=", "time": "", "iduser": "", "type": "newlogin",  
"idusersend": ""}
```

### 7.5.3.2 LOGIN

Éste mensaje indica al servidor que el Raspberry Pi ya tiene una clave simétrica, por tanto, se evita el mecanismo de intercambio de claves, es cifrado mediante RSA. El formato del mensaje es el siguiente.

```
{"idusersend": "0", "iduser": "254630", "typeuser": "raspi", "type": "login", "pass": "d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35", "time": "26/01/2017-23:43:43", "idmsg": 698}
```

Valores de los campos:

- iduser: contiene el id del Raspberry Pi.
- pass: contiene la contraseña del Raspberry Pi. La contraseña es un hash SHA-256 formateado como hexadecimal con relleno a la izquierda.

### 7.5.3.3 REQUESTDEVICES

Éste mensaje es enviado al cliente tras recibir un mensaje de requestdevices con valor 10, lo que indica que el usuario pide las cámaras conectadas al Raspberry Pi y éste contesta con un mensaje por cada cámara conectada. Formato del mensaje:

```
{'typeuser':, 'iduser':, 'typedevice': '10', 'value': 'number', 'time':, 'idusersend': '1000-phone-216', 'idmsg':, 'address': 'video0', 'type': 'requestdevices'}
```

Valores de los campos:

- typedevice= 10. Indica que está enviando los dispositivos de cámara.
- Value = number. Indica que el mensaje es el número de cámaras.
- Address = el id del dispositivo USB conectado al Raspberry Pi.

### 7.5.3.4 SECURE

Mensaje de tipo seguridad, este puede ser enviado por dos causas: por una alerta de seguridad o por interacción del usuario en el modo vacaciones.

#### *MENSAJE DE ALERTA DE ALARMA*

```
{'time':, 'type': 'secure', 'typeuser':, 'idusersend':, 'address':, 'iddevice':, 'iduser':, 'idmsg': 244, 'state': '1', 'value': 'alarm'}
```

Valores de los campos:

- Value = alarm. Indica que se ha detectado un telegrama de un dispositivo de seguridad.
- Address: indica la dirección KNX del dispositivo que ha enviado el telegrama.
- Iddevice: indica el id del dispositivo que hace saltar la alarma.

### *MENSAJE DE MODO VACACIONES*

```
{ "typeuser": "", "iduser": "", "idusersend": "", "time": "", "type": "secure", "value": "holidays", "state": "", "idmsg": }
```

Posibles valores del campo state:

- Ante un mensaje de pregunta de estado por un usuario, valores: 1 activo, 0 no activo.
- Ante un mensaje de cambio de estado del modo vacaciones los posibles valores son: “activate” ha sido activado el modo vacaciones, “stop” ha sido desactivado el modo vacaciones.

#### *7.5.3.5 ACK*

Los mensajes de tipo ack se envían tras recibir un mensaje en el Raspberry Pi. Hay diversos tipos de ack. Éste es identificado por el cliente por el campo value. El cual puede tener las siguientes cadenas.

- Posibles opciones del campo value:
  - value = received. Indica que el Raspberry Pi ha recibido una orden por parte del usuario y va a procesar la petición.
  - Value = terminate. Indica que el Raspberry Pi ha procesado la orden y la ha completado.
  - Value = id del mensaje recibido, éste mensaje es para mensajes entre Raspberry Pi y servidor.
  - Value = capture. Indica que el Raspberry Pi ante una petición de captura de imagen ha terminado la subida de la imagen al servidor y está lista para descargar.

### *MENSAJE DE ORDEN RECIBIDA*

```
{ 'iduser':, 'time':, 'value': 'received', 'state':, 'iddevice':, 'type': 'ack', 'address':, 'typeuser': 'raspi', 'idmsg':, 'idusersend': }
```

Posibles valores de los campos:

- state: contiene el valor de cambio que ha enviado el usuario. Éste puede ser: 0, 1 y rango entre 0 y 100.
- Iddevice: contiene el id del dispositivo del cambio que el usuario ha enviado.
- Address: contiene la dirección KNX del dispositivo del cambio que el usuario ha enviado.

### *MENSAJE DE ORDEN TERMINADA*

```
{ 'iduser':, 'time':, 'value': 'terminate', 'state': '1', 'iddevice': '3', 'type': 'ack', 'address': '1/1/2', 'typeuser': 'raspi', 'idmsg':, 'idusersend': }
```

Posibles valores de los campos:

- state: contiene el valor de cambio que ha enviado el usuario. Éste puede ser: 0, 1 y rango entre 0 y 100.
- Iddevice: contiene el id del dispositivo del cambio que el usuario ha enviado.
- Address: contiene la dirección KNX del dispositivo del cambio que el usuario ha enviado.

#### *MENSAJE DE MENSAJE RECIBIDO*

```
{'typeuser': 'raspi', 'time': '27/01/2017-00:06:14', 'type': 'ack', 'value': '375',
'idusersend': '0', 'idmsg:', 'iduser': '254630'}
```

#### *MENSAJE ORDEN DE CAPTURA DE IMAGEN TERMINADA*

```
{'address': 'ziaRb3LbDdQaL+Me4pRYtiHq6qDe1QCVBvdgs6/GGcI=', 'value':
'capture', 'idmsg:', 'idusersend:', 'typeuser:', 'state': 'terminate', 'type': 'ack', 'time:', 'iduser':
}
```

Posibles valores de los campos:

- state: terminate. Indica que ya está disponible la imagen en el servidor de imágenes.
- Address: hash del nombre de la imagen. Éste parámetro es enviado como variable de datos en la petición GET para poder descargar la imagen.

#### *7.5.3.6 CHANGE*

Éste mensaje es enviado al servidor cuando el monitor del bus KNX recibe un telegrama. Informa de un cambio realizado. El formato del mensaje es el siguiente.

```
{'iduser': '254630', 'time': '26/01/2017-23:52:43', 'value': 'device', 'state': '1', 'type':
'change', 'address': '1/1/2', 'typeuser': 'raspi', 'idmsg:', 'userchange': '1000', 'idusersend':
'0'}
```

Valores de los campos:

- Value: device. Indica al servidor que se ha detectado un cambio en un dispositivo.
- state. Indica el estado del cambio del dispositivo. Posibles valores: 0 activo, 1 no activo y rango entre 0 y 100. Esta información es variable. Depende de los valores programados en los dispositivos KNX.
- address. Contiene la dirección KNX del dispositivo que ha cambiado de estado.
- userchange. Indica quien ha sido el responsable del cambio. Posibles valores: id de sesión de usuario, micrófono o Cambio desde su hogar.

#### *7.5.3.7 PING*

Mensaje enviado al servidor cada 50 segundos para garantizar la conectividad entre el Raspberry Pi y el servidor, como segunda misión tiene como misión mantener el estado de temperatura en el servidor actualizada.

```
{'idmsg:', 'type': 'ping', 'typeuser:', 'idusersend': '0', 'time:', 'temperature': '19.0', 'iduser': }
```

Valores de los campos:

- Temperatura: Indica la temperatura del hogar.

## 7.5.4 SERVIDOR

### 7.5.4.1 LOGIN

Mensajes de respuesta del servidor ante un Login o nuevo Login de cliente o Raspberry Pi.

#### *MENSAJE DE RESPUESTA TRAS UN NUEVO LOGIN CORRECTO*

```
{"time":"","type":"login", "idmsg":, "value":"ok",  
"key":"qXNG7xPSHsy0FsSRzoDMmKm4vKAAbpf91pXv13uXFFA="}
```

En éste mensaje se produce el intercambio de clave simétrica. La clave para el cifrado simétrico es enviada codificada en Base64.

#### *MENSAJE DE RESPUESTA TRAS UN LOGIN/NUEVO LOGIN INCORRECTO*

```
{"typeuser":"","time":"","type":"login","value":"err"}
```

#### *MENSAJE DE RESPUESTA TRAS UN LOGIN CORRECTO*

```
{"idmsg":568,"iduser":"0","idusersend":"1000","typeuser":"server","time":"27/01/  
2017-03:58:13","type":"login","value":"ok"}
```

#### *MENSAJE DE INTERCAMBIO DE INFORMACIÓN CON EL CLIENTE*

```
{"idmsg":625,"iduser":"0","nameuser":"vallemar","idhome":"254630","session":  
"218","idusersend":"1000","typeuser":"server","time":"27/01/2017-03:49:29",  
"type":"login","secure":"0","value":"ok","email":"JuandeDiosMartínezVallejo@gmail.c  
om" "root":"1"}
```

Valores de los campos:

- nameuser: éste campo contiene el nombre de usuario.
- idhome: contiene el identificador del hogar asociado al cliente.
- Sesión: número de sesión establecido por el servidor para la identificación del terminal.
- Secure: indica el estado de la seguridad del hogar. 0 no activa, 1 activa.
- Email: email con el que el usuario está registrado en el servicio.
- root: indica si el usuario es el usuario root del hogar. 0 no es root, 1 es root.

### 7.5.4.2 SHUTDOWN

Mensaje de respuesta cuando un usuario desea terminar la comunicación.

## MENSAJE DE RESPUESTA TRAS UN MENSAJE DE CIERRE DE CONEXIÓN

```
{ "idmsg": "", "iduser": "", "idusersend": "", "typeuser": "", "time": "", "type": "shutdown", "value": "bye" }
```

Valores:

- Value: bye. Indica que el servidor ha recibido correctamente una petición de terminar con la conexión.

### 7.5.4.3 CLOSESESSION

Mensaje enviado por el servidor ante la petición de usuario de cierre de sesión. Mensaje enviado cuando una sesión que un usuario ha solicitado cerrar y el mensaje enviado por medio de firebase ha fallado por algún motivo. Cuando la sesión desea realizar la conexión es iniciada el siguiente mensaje.

```
{ "idmsg": "", "iduser": "", "idusersend": "", "typeuser": "server", "time": "", "type": "closeession", "value": "terminated" }
```

Valores de los campos:

- terminated: indica que la sesión ha de ser terminada en el dispositivo.

### 7.5.4.4 REFRESH

Éste mensaje es enviado al usuario cuando se ha producido un cambio en el hogar o ante un mensaje de ping por parte del Raspberry Pi. Tiene la información que se muestra en la pantalla principal de la aplicación.

```
{ "idmsg": "", "iduser": "", "idusersend": "", "temperature": "18.0", "countlight": "3", "typeuser": "", "time": "", "type": "refresh" }
```

Valores de los campos:

- Type: refresh.
- Temperature: indica la temperatura del hogar.
- Countlight: indica las luces encendidas en el hogar.

### 7.5.4.5 REQUESTDEVICES

Mensaje de respuesta del servidor ante una petición de lista de dispositivos por el usuario. El formato de respuesta es el siguiente.

```
{ "idmsg": "", "iduser": "", "typedevice": "", "idusersend": "", "typeuser": "", "time": "", "type": "requestdevices", "value": [{"description": "Luz Salón", "state": "0", "id": "1", "type": "1", "actions": [{"address": "1/1/1", "action": "EIS1"}]} ] }
```

Valores de los campos:

- Type: requestdevices
- Typedevice: indica el tipo de dispositivo que esta enviado.
- Value: Array JSON el cual envía toda la información de los dispositivos. Campo que existen en el:
  - description: descripción del dispositivo.

- State: indica el estado. 0 apagado. 1 encendido. Otro valor pueden producirlo los dispositivos de tipo regulación el cual tienen valores como las luces entre 0 y 100.
- Id: id del dispositivo para el servidor.
- Type: indica el tipo de dispositivo. Éste valor también está en éste campo ya que para mensajes en los que se manda varios tipos de dispositivos el campo que anterior typedevice es omitido y los dispositivos se identifican mediante éste campo.
- actions: Array JSON. Contiene las distintas direcciones asociadas a su acción. Un dispositivo puede tener varias direcciones como ocurre con las persianas. Con éste campo se identifican todas sus acciones.
  - action: tipo de dato KNX que produce su acción.
  - address: dirección KNX.

Tipos de datos KNX que se envían en la totalidad de la aplicación son los siguientes.

- EIS1 Conmutar. Enciende/apaga, subir/bajar persianas.
- EIS2 Parar persiana.
- EIS6 Valor relativo. Regular luces.

#### 7.5.4.6 POWER

Éste tipo de mensajes sólo será reenviado por el servidor desde el usuario al Raspberry Pi.

#### 7.5.4.7 DIMMER

Éste tipo de mensajes sólo será reenviado por el servidor desde el usuario al Raspberry Pi.

#### 7.5.4.8 SCENE

Mensaje de respuesta ante un mensaje de tipo scene y valor history. Se envían las escenas creadas.

```
{ "idmsg": "", "iduser": "", "idusersend": "", "time": "", "type": "scene", "value": "request", "scene": [
  { "hour": "08:00", "name": "Encendido
  mañanas", "activate": "0", "action": "1", "id": "53", "type": "1", "autor": "vallemar" } ] }
```

Valores de los campos:

- Value: request. Indica al usuario que está contestando a una petición de historial de escenas.
- scene: Array JSON el cual contiene la totalidad de las escenas y sus propiedades. Dentro de éste se encuentran los siguientes campos.
  - Hour: indica la hora a la que la escena será activada.
  - Name: nombre identificador de la escena.

- Actíivate: indica si la escena esta activa o no donde los valores son: 0 no activa, 1 activa.
- Action: Indica la acción de la escena. Los posibles valores son: 0 apagar dispositivos, 1 encender.
- Id: id de la escena.
- Type: indica los tipos de dispositivos implicados en dicha escena. Ver tabla de dispositivos.
- Autor: nombre de usuario que creo la escena.

#### 7.5.4.9 SECURE

Mensaje de respuesta ante un mensaje de tipo secure y valor history.

```
{"idmsg":300993,"iduser":"0","record":[{"date":"","type":"","body":""}],
  "idusersend":"254630","typeuser":"server","time":"05/02/2017-
  21:58:21","type":"secure","value":"history"}
```

Valores de los campos:

- Value = history. Indica que es un mensaje de tipo respuesta ante un mensaje de tipo history del usuario.
- record: Array JSON el cual contiene la totalidad de las alarmas detectadas en el hogar. Dentro de éste se encuentran los siguientes campos.
  - Date: Indica la fecha en la que la alarma fue detectada.
  - Type: indica el tipo de dispositivo que realizo la detección de la alarma. Estos pueden ser:
    - Sensor presencia.
    - Sensor humo.
    - Sensor humedad.
    - Sensor magnético.
    - Sensor de viento.
  - Body: Indica la cadena de texto que fue enviada al usuario.

#### 7.5.4.10 ALARM

Mensaje enviado al usuario ante la llegada de un mensaje por el Raspberry Pi de tipo “secure” y value “alarm”. Éste mensaje es enviado a través de Firebase.

```
{'idalarm'= , 'body'=, 'type'='alarm'}
```

Valores de los campos:

- type= alarm. Indica que es un mensaje de tipo alarma detectada.
- body: cadena enviada indicando cual es el dispositivo que ha realizado la alerta de seguridad.
- Idalarm: indica el id de la alarma.

#### 7.5.4.11 ACK

Mensaje de confirmación de datos recibidos en el servidor. Éste mensaje es diferenciado por el campo value. Formato del mensaje.

```
{"idmsg":,"iduser":"0","idusersend":"","typeuser":"server","time":"","type":"ack","value":""}
```

Campo value:

- Value = ping. Indica que está asintiendo la llegada de un mensaje de tipo ping del Raspberry Pi.
- Value = serverreceived. Indica que está respondiendo al usuario ante los mensajes de tipo:
  - Scene.
  - Remote.
  - Requestdevices.
  - Alarm.
  - camera.
- Value = serverreceived\_change. Mensaje de confirmación de recepción ante los mensajes enviados por el usuario de tipo:
  - Power.
  - Dimmer.
- Value = id del mensaje recibido. Para todos los demás mensajes el servidor responde asintiendo el id del mensaje recibido.

#### 7.5.4.12 CHANGE

Mensaje enviado por el servidor cuando es recibido un mensaje por el usuario de cambio de estado en la seguridad del hogar o mensaje reenviado desde el Raspberry Pi al usuario cuando se produce un cambio en el hogar. Formato del mensaje.

```
{"idmsg":,"iduser":"","idusersend":"","typeuser":"server","time":"","state":"1","type":"change","value":"secure","autor":"vallemar"}
```

Valores de los campos:

- Value = secure. Indica al usuario que ha sido realizado un cambio en el estado de la seguridad del hogar.
- Autor = indica el nombre de usuario que ha realizado el cambio.

El mensaje reenviado de valor estatedevice es obviado ya que es enviado por el Raspberry Pi y ya está explicado.

### 7.5.4.13 CONFIG

Mensaje de respuesta enviado al usuario cuando es recibido un mensaje de tipo config por éste. Los mensajes que pueden ser enviados son los siguientes. Éste es diferenciado por su campo typeconfig y configuser. Posibles valores:

- Campo typeconfig:
  - typeconfig = configuser. Indica que es un mensaje de respuesta ante un mensaje configuser.
  - typeconfig = confighome. Indica que es un mensaje de respuesta ante un mensaje confighome. Éste mensaje no es utilizado, ha sido diseñado para posibles ampliaciones.
- Campo configuser:
  - configuser = sessions. Indica que es un mensaje de respuesta que contiene las sesiones de usuario.
  - configuser = connectedhome. Indica que es un mensaje de respuesta que contiene los usuarios conectados al hogar.

### MENSAJE DE SESIONES

```
{ "idmsg": "", "iduser": "", "configuser": "sessions", "typeconfig": "configuser", "iduserse  
nd": "", "typeuser": "server", "time": "", "type": "config", "value": [{"date": "26/01/2017", "os":  
"6.0.1", "iddevice": "763e50965d4e6a25", "modeldevice": "BQ Aquaris  
X5", "typedevice": "phone", "idsession": "763e50965d4e6a25"}]}
```

- Value: array JSON conteniendo los datos de sesiones.
  - Date: fecha del primer inicio de sesión del dispositivo.
  - Os: versión del sistema operativo.
  - Iddevice: indica el id del dispositivo.
  - Modeldevice: indica el modelo del dispositivo.
  - Typedevice: phone. Éste campo ha sido agregado por posibles ampliaciones en el servicio. Soporta tipos phone, pc y browser.

### MENSAJE DE USUARIO CONECTADOS AL HOGAR

```
{ "idmsg": "", "iduser": "", "configuser": "connectedhome", "typeconfig": "configuser", "i  
dusersend": "", "typeuser": "", "time": "", "type": "config", "value": [{"nameuser": "vallemar", "  
iduser": "1000", "emailuser": "JuandeDiosMartínezVallejo@gmail.com", "dateup": "17/10/  
2016", "isconnect": "1"}]}
```

- Value: array JSON conteniendo los datos de usuarios.
  - nameuser: nombre de usuario.
  - iduser: id del usuario.
  - emailuser: indica el email del usuario.
  - dateup: indica la fecha de registro en el hogar.
  - isconnect: indica el estado de conexión actual del usuario. Los posibles valores son 0 no conectado, 1 conectado.

#### 7.5.4.14 DEVICESHOME

Mensaje enviado por el servidor al Raspberry Pi indicando la totalidad de dispositivos existentes en el hogar y sus características.

```
{"idmsg":,"iduser":"0","devices":[{"iddevice":"1","description":"Luz Sal\u00f3n","device":[{"address":"1/1/1","type":"EIS1"}]},{"idusersend":"254630","typeuser":"server","time":"27/01/2017-13:00:14","type":"deviceshome","value":"update"}
```

Valores de los campos:

- Value = update. Indica que el Raspberry Pi debe de actualizar su tabla de dispositivos.
- Devices. Array JSON conteniendo los valores de los dispositivos.
  - Iddevice: indica el id del dispositivo.
  - Description: nombre del dispositivo.
  - Device: array JSON que contiene las propiedades de accionamiento de los dispositivos.
    - Address: direcci\u00f3n asociada a un dispositivo.
    - type: tipo de acci\u00f3n a realizar asociada a la direcci\u00f3n anterior.

#### 7.5.4.15 UPDATECONNECTION

Mensaje enviado por el servidor al usuario para informar de que ha habido alg\u00fan cambio en la conexi\u00f3n con el hogar (Raspberry Pi).

```
{"idmsg":,"iduser":"","idusersend":"","typeuser":"","time":"","type":"updateconnection","value":""}
```

Valor del campo value:

- Value = lost. Indica que ha habido un error en la comunicaci\u00f3n con el Raspberry Pi y ha sido cerrada.
- Value= connected. Indica al usuario que ha sido recuperada la comunicaci\u00f3n con el hogar (Raspberry Pi).

## 7.6 MODELO DE CASOS DE USO

### 7.6.1 DESCRIPCI\u00d3N CASO DE USO ACCESO AL SERVICIO

#### 7.6.1.1 RASPBERRY PI – SERVIDOR

El dispositivo Raspberry Pi tras iniciar el sistema operativo, arrancara el software desarrollado. Tras la ejecuci\u00f3n del software se ejecutan las siguientes instrucciones:

1. Se confirma que el dispositivo tiene conexi\u00f3n a Internet. Si no tiene espera hasta tenerla para seguir con el inicio de sesi\u00f3n.
2. Se comprueba que han sido generadas las claves RSA de 2048 bits. Si no existen, se generan y son almacenadas en archivos con extensi\u00f3n Privacy-enhanced Electronic

Mail (PEM) para que así los posteriores inicios sean más rápidos ya que generar las claves RSA es un proceso muy costoso.

3. Se realiza una conexión TCP con el servidor al puerto 443.
4. Cuando es aceptada la conexión el servidor envía su certificado digital SSL que contiene su clave pública de RSA.

### Primer inicio de sesión de Raspberry Pi.

1. El dispositivo Raspberry Pi comprueba que el certificado digital sea válido. Si éste lo es son cifrados las credenciales del Raspberry con la clave pública del servidor y se envían los datos. Éste mensaje también contiene un campo que indica que es un nuevo Login. Formato del mensaje:

```
{'idusersend': '0', 'typeuser': 'raspi', 'time': '27/01/2017-12:28:55', 'iduser': '254630', 'type': 'newlogin', 'pass': 'd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35', 'idmsgsend': '0'}
```

2. Raspberry pi envía un mensaje con su clave pública RSA al servidor. Formato del mensaje:

```
{'idusersend': '0', 'typeuser': 'raspi', 'time': '27/01/2017-12:28:55', 'iduser': '254630', 'exponent': 'NjU1Mzc=', 'modulus': 'MTg0NTIzMDcwNTcwMzA4MjM0NDk1MTU2NzIwMDg1OTI1NDkxMzcxNzQ0NTUzMzY0MDc5NTAwNDZlOTU5MDk5NTIwMzY4MzkwNDUzNzk3MDU4NDQzMTE1MzcxNDE0MTQzNzc5NzYxNzY0MjE1MjY0MTc4MzY3Mzc0ODAwNDc5MzYxOTU5NDMwNDM1MDU3NDI2NDQ3Mjc5NTEzOTU4MTI1ODMwMTE2NzYxMTg2MzE5OTg0NTQ2MDc0MDUwNTcyNDE3MTg0ODY3MDYyNTQ5NDIxNjIxNjk2NTc1MjQ4NjkzOTYxOTgwNzY0MzYzNzY3MzY1ODU1ODc1MjI0NzI0NDg1NDk3MDIyNzAyMjI0OTAyNzI4NzY3MTY0NjI3NjExNzkyNDk2MTE3OTU4Mjg0MjcyOTA2NjA3MTM0Mjk1OTAzMzkyNTA5NTMzNTYyODEyNTg1NTE4NjQxNDE3NDkzOTE2MzYxNTM2Mzg4OTk4MDE1Nzk0MTI4NTIxNjY5NzI0MTUwNzY1MTkxMDQ4NTc5ODIyODg4MTMwNjMwODk3NDAwMDIzNzY0MDg0OTQ4NDU3MjYyNTE4MzY0MzY3MDE4NjM4MzI1NzY2NzY3NzY1NTg2NTkxMTcxNjcyNDI5OTE2NzYxNjY0MjI3MzY0OTMxODI0NDYyMTYzMTg0MTc2NTYyNjE5ODYyMjgzMDE5MjkwNzU3MjQ3MjQ1MTI3ODI2MDIzNDQxNjA2NDYyMjE3NjQxNDM4MTkxOTc4NDE5MzY3OTM2ODc0NDc1NTQ4ODY2NjUyNTU3NjA4Mjg3NzY1', 'type': 'newlogin'}
```

3. El servidor descifra el mensaje con su clave privada RSA y comprueba las credenciales del Raspberry Pi almacenadas en la base de datos. Si son correctos el servidor genera una clave simétrica aleatoria AES 256 bits codificada en base64. Si no lo son responde con un mensaje de autenticación incorrecta y cierra la conexión.

```
{"typeuser": "server", "time": "27/01/2017-12:42:15", "type": "login", "value": "err"}
```

4. El servidor cifra la clave simétrica con la clave pública del Raspberry Pi y se envía a éste. En el mensaje posterior el servidor envía el primer número de mensaje, lo que indica al Raspberry Pi que tiene que seguir esta secuencia.

```
{"idmsg":196,"time":"27/01/2017-12:29:18","type":"login","value":"ok",  
"key":"WJ7Ajm7k5wRIYdPD+WNY6qcOFIyySIX6/vbS7P/w22U="}
```

5. El Raspberry Pi descifra los datos con su clave privada y obtiene la clase simétrica de AES. Esta es almacenada en la base de datos para ser utilizada en los siguientes inicios de sesión.
  - La autenticación ha sido completada.
  - El Raspberry Pi pasa a la espera de eventos.

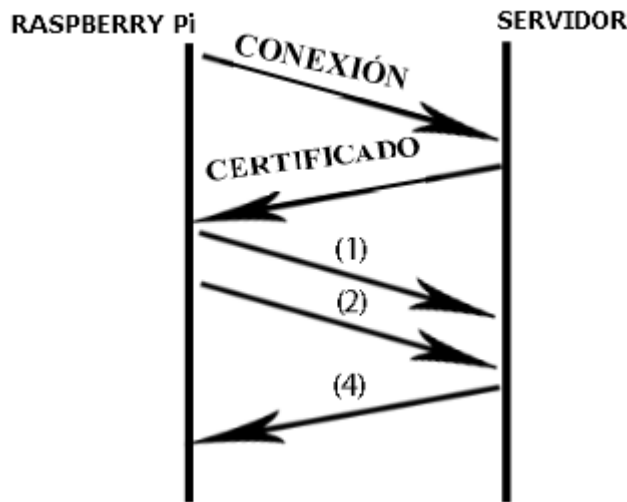


Figura 7. Esquema nueva conexión Raspberry Pi

### Inicio de sesión posteriores.

Si anteriormente ya se había iniciado una sesión con el usuario de Raspberry Pi y dispositivo se ejecutan las siguientes instrucciones:

1. El dispositivo Raspberry Pi comprueba que el certificado digital sea válido. Si éste lo es son cifrados las credenciales del Raspberry con la clave pública del servidor y se envían los datos, éste mensaje también contiene un campo que indica que es un Login ya existente. La contraseña es enviada realizando a éste el hash SHA-256 formateado a hexadecimal con relleno a la izquierda.

```
{"iduser": "254630", "type": "login", "time": "27/01/2017-12:43:23", "idusersend": "0",  
"typeuser": "raspi", "pass":  
"d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35"}
```

2. El servidor descifra el mensaje con su clave privada RSA y comprueba las credenciales del Raspberry Pi almacenadas en la base de datos. Si las credenciales son correctas el servidor responde con un mensaje de autenticación correcta.

```
{"idmsg":105,"iduser":"0","idusersend":"254630","typeuser":"server","time":"27/  
01/2017-13:43:46","type":"login","value":"ok"}
```

En el caso de que las credenciales sean incorrectas el servidor responde con el siguiente mensaje y éste cierra la conexión:

```
{"typeuser":"server","time":"27/01/2017-13:42:15","type":"login","value":"err"}
```

- La autenticación ha sido completada.
- El Raspberry Pi pasa a la espera de eventos.

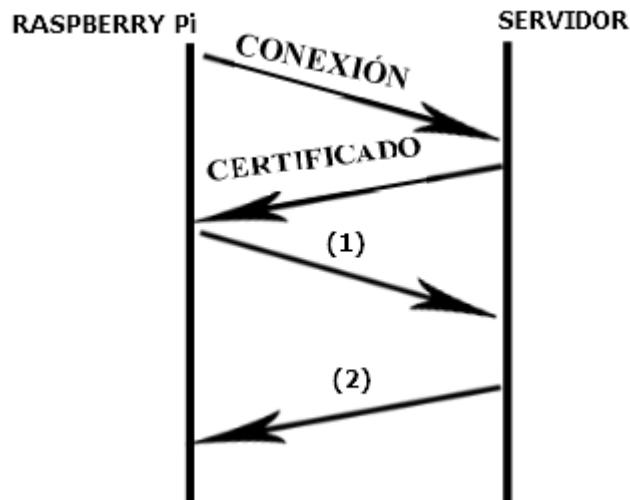


Figura 8. Esquema conexión Raspberry Pi

#### 7.6.1.2 USUARIO ANDROID – SERVIDOR

Para el acceso al servicio por parte del usuario éste debe de tener instalada la aplicación Android y estar registrado en la aplicación.

El acceso al servicio de un dispositivo Android es parecido al del Raspberry Pi, pero con algunas modificaciones. En lo referente al envío de datos éste incluye algunos campos para mejorar la experiencia de uso por parte del usuario.

Dicha aplicación está diseñada para alcanzar un máximo rendimiento. La primera instancia al ser ejecutada la aplicación comprueba si existe una sesión iniciada. Si esta no existe se ejecutan las siguientes instrucciones

##### **Primer inicio de sesión de un usuario.**

1. La aplicación lanza la pantalla de Login, ésta requiere las credenciales de usuario para continuar. También es posible un registro de usuario en el servicio, esto es visto en detalle en el apartado de registro.
2. Una vez el usuario introduce las credenciales: se comprueba que han sido generadas las claves RSA de 2048 bits para un Login seguro. Si no existen se generan y son almacenadas en Base64 en las preferencias de usuario de Android.
3. Se realiza una conexión TCP con el servidor al puerto 443.
4. Cuando es aceptada la conexión el servidor envía su certificado digital SSL que contiene su clave pública de RSA.
5. Se confirma que el certificado digital sea válido. Si éste lo es son cifrados las credenciales de usuario con la clave pública del servidor y se envían los datos. Éste mensaje también contiene un campo que indica que es un nuevo Login. Formato del mensaje:

```
{"typeuser":"user","nameuser":"vallemar","pass":"8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92","idusersend":"0","time":"27/01/2017-12:56:24","type":"newlogin"}
```

6. La aplicación envía un mensaje con su clave pública RSA al servidor.

```
{"typeuser":"user","idusersend":"0","time":"27/01/2017-12:58:37","type":"login","value":"newlogin","modulus":"AMgqDPBRgcOkKK4fbobLZg8nAPZelpP1oN1rByQc1K6co04p5uCsR085yTWXOhu72DflEHZYZ+p4X128fBFRzE91Do0gERUignZRrhv4Ly8uQcPPRLZcKeKbHuXDqo2Pn5GoxDMV/Z72zqmpA01xGGyfNFI+4wU6zuzJ6TH9muhg8sOokCptb9iK70ZqKoGH9etlZnADt10LUT\UGXKO DC7TvTYH9DrvyZ5\BbJIP9cFhxpQvmUsiXjtKs7Npu0gBCpsNz42ki4D9KSvo5fKFt w2O45US48j2YwR5xgrr8lSGRyPwADGScjkyKWTlvGBjvh8lFbdyyv\IMbUIg+GRd8 =","exponent":"AQAB"}
```

7. El servidor descifra el mensaje con su clave privada RSA y comprueba las credenciales del usuario almacenadas en la base de datos. Si son correctos el servidor genera una clave simétrica aleatoria AES 256 bits codificada en base64. Si no lo son responde con un mensaje de autenticación incorrecta y cierra la conexión.

```
{"typeuser":"server","time":"27/01/2017-13:42:15","type":"login","value":"err"}
```

8. El servidor cifra la clave simétrica con la clave pública del usuario y se envía a éste. En éste mensaje se inicia la numeración de mensajes impuesta por el servidor.

```
{"idmsg":589,"time":"27/01/2017-13:59:06","type":"login","value":"ok","key":"qXNG7xPSHsy0FsSRzoDMmKm4vKAA bpf91pXv13uXFFA="}
```

9. La aplicación descifra los datos con su clave privada y obtiene la clave simétrica de AES. Esta es almacenada en la base de datos para ser utilizada en los siguientes inicios de sesión y comunicación. Si esto ocurre significa que las credenciales son correctas, la aplicación guarda los datos de usuario para no tener que ser introducidos en cada inicio de sesión.
10. Para mejorar la experiencia de uso por parte del usuario y el manejo de sesiones que el usuario puede tener activas éste paso es incluido en el inicio de sesión. El usuario envía al servidor los siguientes datos cifrados con la clave simétrica anteriormente establecida: Modelo de móvil, id de dispositivo, tipo de dispositivo e id Token Firebase.

```
{"typeuser":"user","nameuser":"vallemar","idusersend":"0","time":"27/01/2017-12:58:37","type":"login","idfirebase":"clPu2Biz7HA:APA91bFyK-WHdIPDOtEF88YrEU-_LodoPv5d4vWGzf6DaQcvAQcfRusinJsMo3VLC0lBJEymAd_oZ0ByCAVAi2ydN7cPYn7E2m0RVXlJa35wz6S-B3uWa8vJ8kTtfyc-acwESmOLcX3T","modeldevice":"Android SDK Built For X86","iddevice":"6a96b85ef2c1d49d","os":"7.0","typedevice":"phone","idmsg":}
```

- La autenticación ha sido completada.

- El servidor responde tanto con los datos de sesión del usuario como con los datos del hogar asociado:

```
{"idmsg":591,"iduser":"0","nameuser":"vallemar","idhome":"254630","session":"224","idusersend":"1000","typeuser":"server","time":"27/01/2017-13:59:06","type":"login","secure":"1","value":"session","email":"JuandeDiosMartínezVallejo@gmail.com","root":"1"}
```

- El usuario puede comenzar con el control remoto domótico. Todos los datos transmitidos serán cifrados con AES-CBC 256 bits y MAC.

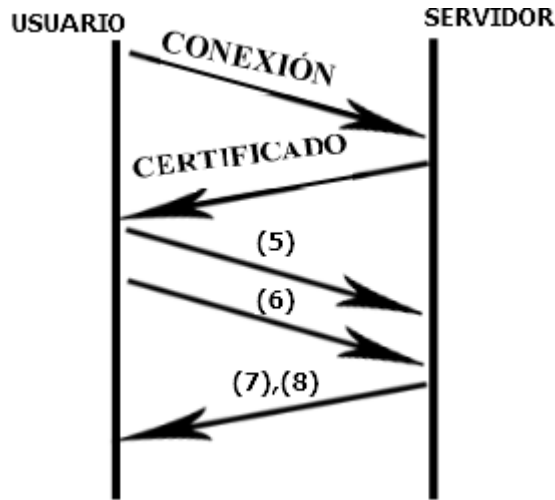


Figura 9. Esquema nueva conexión Android

**Inicio de sesión posteriores.**

Si anteriormente ya se había sido iniciada una sesión por el usuario y dispositivo Android no será necesario introducir las credenciales por parte de usuario ya que estas han sido almacenadas en la base de datos. Se ejecutan las siguientes instrucciones.

1. Se realiza una conexión TCP con el servidor al puerto 443.
2. Cuando es aceptada la conexión el servidor envía su certificado digital SSL que contiene su clave pública.
3. Se confirma que el certificado digital sea válido. Si éste lo es son cifrados las credenciales de usuario con la clave pública del servidor y se envían los datos, éste mensaje también contiene un campo que indica que es un login de sesión existente.

Formato del mensaje:

```
{"typeuser":"user","nameuser":"vallemar","pass":"8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92","time":"27\01\2017-13:14:04","type":"login"}
```

4. El servidor descifra el mensaje con su clave privada RSA y comprueba las credenciales del usuario almacenadas en la base de datos. Si las credenciales son correctas el servidor responde con un mensaje de autenticación correcta. En éste mensaje se inicia la numeración de mensajes.

```
{"idmsg":332175,"iduser":"0","idusersend":"1000","typeuser":"server","time":"27/01/2017-14:14:33","type":"login","value":"ok"}
```

5. El usuario envía al servidor los siguientes datos con la clave simétrica anteriormente establecida. Formato del mensaje:

```
{ "typeuser": "user", "nameuser": "vallemar", "pass": "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92", "time": "27/01/2017-13:14:04", "type": "login", "idfirebase": "clPu2Biz7HA:APA91bFyK-WHdIPDOtEF88YrEU-_LodoPv5d4vWGzf6DaQcvAQcfRusinJsMo3VLC0lBJEymAd_oZ0ByCAVAi2ydN7cPYn7E2m0RVXlJa35wz6S-B3uWa8vJ8kTtfyc-acwESmOLcX3T", "modeldevice": "Android SDK Built For X86", "iddevice": "6a96b85ef2c1d49d", "os": "7.0", "typedevice": "phone", "idmsg": 332176 }
```

6. El servidor responde tanto con los datos de sesión del usuario como con los datos del hogar asociado:

```
{ "idmsg": 332177, "iduser": "0", "nameuser": "vallemar", "idhome": "254630", "session": "224", "idusersend": "1000", "typeuser": "server", "time": "27/01/2017-14:14:33", "type": "login", "secure": "1", "value": "session", "email": "JuandeDiosMartínezVallejo@gmail.com", "root": "1" }
```

- La autenticación ha sido completada.
- El usuario puede comenzar con el control remoto domótico. Todos los datos transmitidos serán cifrados con AES-CBC 256 bits y MAC.

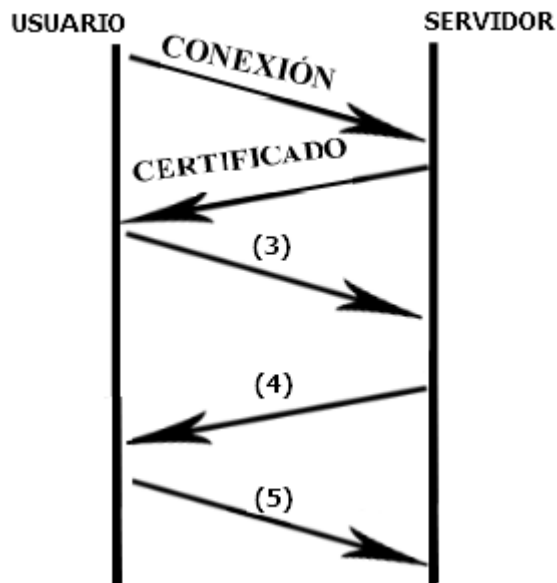


Figura 10. Esquema conexión Android

## 7.6.2 DESCRIPCIÓN MODULO CONFORT

El módulo de confort de la aplicación dota al usuario de los beneficios del control remoto del hogar de los dispositivos que son descritos. A continuación, se desglosa como actúa la aplicación ante la interacción de los usuarios con la aplicación.

### 7.6.2.1 DESCRIPCIÓN CASO DE USO GESTIÓN DE LUCES

Desde la pantalla de luces puede monitorizarse el estado de las distintas luces del hogar y la gestión de estas. Cuando el usuario entra a éste módulo se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste de tipo “requestdevices” y valor “1” (Véase tipos de dispositivos). Formato del mensaje:

```
{ "typeuser": "user", "iduser": "1000-phone-224", "idusersend": "0", "time": "27\01\2017-13:20:04", "type": "requestdevices", "value": "1", "idmsg": "332198" }
```

2. El servidor realiza una consulta a la base de datos y obtiene los dispositivos de tipo “1” asociados al hogar, también su estado. Estos datos son enviados al usuario. Formato del mensaje:

```
{ "idmsg": "332200", "iduser": "0", "typedevice": "1", "idusersend": "1000-phone-224", "typeuser": "server", "time": "27/01/2017-14:20:33", "type": "requestdevices", "value": [{"description": "Luz Sal3n", "state": "0", "id": "1", "type": "1", "actions": [{"address": "1/1/1", "action": "EIS1"}]}, {"description": "Luz Piscina", "state": "0", "id": "13", "type": "1", "actions": [{"address": "1/1/16", "action": "EIS1"}]} ] }
```

En el caso de que no haber luces compatibles con KNX instalado en el hogar el servidor responde con el siguiente mensaje:

```
{ "idmsg": "555", "iduser": "0", "typedevice": "1", "idusersend": "1000-phone-224", "typeuser": "server", "time": "29/12/2016-17:52:06", "type": "requestdevices", "value": [] }
```

3. La aplicación tras recibir los datos los parsea y se muestra de forma clara al usuario informando el estado de la totalidad de las luces.
4. Si el usuario quiere modificar el estado de una de las luces puede hacerlo pulsando el botón asociado al dispositivo. Si esto sucede la aplicación envía al servidor un mensaje que puede ser de varios tipos. Esto es así ya que un dispositivo de luz puede tener varios tipos de opciones de modificación: Power o Dimmer.
  - a. Power.

```
{ "typeuser": "user", "iduser": "1000-phone-224", "idusersend": "254630", "time": "27\01\2017-13:23:24", "type": "power", "value": "1", "iddevice": "1", "typedevice": "1", "address": "1/1/1", "idmsg": "332213" }
```

- b. Dimmer:

```
{ "typeuser": "user", "iduser": "1000-phone-224", "idusersend": "254630", "time": "27\01\2017-13:26:35", "type": "dimmer", "value": "46", "iddevice": "2", "address": "1/1/5", "typedevice": "1" }
```

5. Tras ser recibido el mensaje de cambio de estado en el servidor éste responde con un mensaje de tipo ACK para indicar que se ha recibido. Cuando el mensaje de ACK es

recibido por la aplicación muestra al usuario un progreso en el envío de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:

- a. Encendido: La bombilla con relleno en color negro indica que dicha luz está apagada, esta cambia de color de relleno a transparente.



Figura 11. Primer ACK encendido luces

- b. Apagado: La bombilla con relleno en color amarillo indica que dicha luz está encendida, esta pasa a un relleno del 50% amarillo 50% transparente.



Figura 12. Primer ACK apagado luces

- c. Formato del mensaje de ACK:

```
{"idmsg":332247,"iduser":"0","iddevice":"2","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-14:27:03","state":"46","type":"ack","value":"serverreceived_change"}
```

6. El servidor envía la orden a el Raspberry Pi asociada al usuario.
7. El Raspberry Pi recibe la orden y envía un mensaje al usuario de tipo ACK de tipo recibido indicando que se ha recibido la orden y se va a proceder a realizar el cambio.  
Formato del mensaje:

```
{"idmsg":332248,"iduser":"254630","address":"1/1/5","iddevice":"2","idusersend":"1000-phone-224","typeuser":"raspi","state":"46","time":"27/01/2017-13:26:41","type":"ack","value":"received"}
```

8. Cuando el mensaje de ACK de tipo recibió enviado por el Raspberry Pi es recibido por la aplicación muestra al usuario un progreso en el envío de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:
  - a. Encendido: La bombilla con relleno de color transparente indica que la orden había sido recibida por el servidor, esta cambia de color de relleno 50% amarillo 50% transparente.



Figura 13. Segundo ACK encendido luces

- b. Apagado: La bombilla con relleno 50% amarillo 50% transparente indica que la orden había sido recibida por el servidor, esta cambia de color de relleno a transparente.



Figura 14. Segundo ACK apagado luces

- 9. Tras ser enviado el telegrama al bus KNX el Raspberry envía un mensaje de tipo ACK con valor terminado. Esto indica que se ha realizado la petición del usuario. Formato del mensaje:

```
{"idmsg":332251,"iduser":"254630","address":"1/1/5","iddevice":"2","idusersend":"1000-phone-224","typeuser":"raspi","state":"46","time":"27/01/2017-13:26:41","type":"ack","value":"terminate"}
```

- 10. Cuando el mensaje de ACK de tipo terminado enviado por el Raspberry Pi es recibido por la aplicación muestra al usuario un progreso en el envío de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:

- a. Encendido: La bombilla con relleno 50% amarillo 50% transparente indica que la orden había sido recibida por el Raspberry, esta cambia de color de relleno 100% amarillo indicando que la orden ha sido completada exitosamente.



Figura 15. Tercer ACK encendido luces, final

- b. Apagado: La bombilla con relleno transparente indica que la orden había sido recibida por el Raspberry, esta cambia de color de relleno a negro indicando que la orden ha sido completada exitosamente.



Figura 16. Tercer ACK apagado luces, final

- 11. El Raspberry Pi envía un mensaje de tipo “change” indicando que se ha registrado un cambio en un dispositivo KNX del hogar. Éste envía al servidor la dirección KNX que ha sido modificada, el nuevo valor y el usuario que ha realizado el cambio. Formato del mensaje:

```
{'type': 'change', 'typeuser': 'raspi', 'state': '0', 'idmsg': 695, 'iduser': '254630', 'userchange': '1000', 'idusersend': '0', 'address': '1/1/1', 'value': 'device', 'time': '27/01/2017-13:31:22'}
```

12. El servidor de aplicación tras recibir el anterior mensaje realiza una actualización en la base de datos actualizando el valor del estado del dispositivo, tras esto envía a todos los clientes conectados excepto al usuario que ha realizado la acción si lo hubiera un mensaje de tipo change indicando que ha sido realizado un cambio en el hogar. Esto hace que un cambio sea recibido por todos los usuarios sin tener que enviar mensajes al servidor de actualización de valores. Formato del mensaje:

```
{"idmsg":332257,"iduser":"0","address":"1/1/5","iddevice":"2","idusersend":"1000-phone-224","userchange":"vallemar","typeuser":"server","time":"27/01/2017-14:27:04","state":"46","type":"change","value":"estatedevice"}
```

13. Cuando el mensaje de tipo change es recibido por la aplicación si es usuario está en la pantalla de luces se modifica el valor del dispositivo a encendido o apagado. Esto es de gran utilidad para el usuario ya que así se logra una monitorización actualizada en todo momento.

#### 7.6.2.2 DESCRIPCIÓN CASO DE USO GESTIÓN DE PERSIANAS

Desde la pantalla de persianas puede se pueden gestionar las distintas persianas del hogar. Cuando el usuario entra a éste apartado se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste es de tipo “requestdevices” y valor “2” (Véase tipos de dispositivos). Formato del mensaje:

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"0","time":"27/01/2017-13:32:40","type":"requestdevices","value":"2","idmsg":332319}
```

2. El servidor realiza una consulta a la base de datos y obtiene los dispositivos de tipo “2” asociados al hogar. Estos datos son enviados al usuario.

```
{"idmsg":332321,"iduser":"0","typedevice":"2","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-14:33:09","type":"requestdevices","value":[{"description":"Persiana Sal3n","state":"0","id":"18","type":"2","actions":[{"address":"1/2/1","action":"EIS1"}, {"address":"1/2/2","action":"EIS2"}]}
```

Se puede observar en el mensaje anterior los dos tipos de acciones asociados a la persiana. La acción EIS1 indica la direcci3n de subir/bajar persiana. La acci3n EIS2 indica la direcci3n KNX de parar persiana.

En el caso de que no haber persianas compatibles con KNX instalado en el hogar el servidor responde con el siguiente mensaje:

```
 {"idmsg":555,"iduser":"0","typedevice":"2","idusersend":"1000-phone-224","typeuser":"server","time":"29/12/2016-17:52:06","type":"requestdevices","value":[]}
```

3. La aplicaci3n tras recibir los datos los parsea y se muestra de forma clara al usuario la totalidad de las persianas del hogar.
4. Si una de las persianas es seleccionada la aplicaci3n muestra una pantalla para realizar una de las dos opciones disponibles: Subir/bajar persiana.

5. Si el usuario quiere modificar el estado de una de las persianas puede hacerlo pulsando sobre el ítem de la lista asociado al dispositivo. Si esto sucede se mostrará en pantalla los botones asociados a las acciones permitidas: Subir/Bajar. Cuando es pulsado uno de estos botones la aplicación envía al servidor un mensaje de tipo power el cual puede variar según el botón pulsado. Formato del mensaje:

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"254630","time":"27/01/2017-13:38:05","type":"power","value":"0","iddevice":"18","typedevice":"2","address":"1/2/1","idmsg":332348}
```

6. Tras ser recibido el mensaje de cambio de estado en el servidor éste responde con un mensaje de tipo ACK para indicar que se ha recibido.

```
{"idmsg":332349,"iduser":"0","iddevice":"18","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-14:38:34","state":"0","type":"ack","value":"serverreceived_change"}
```

7. El servidor reenvía el mensaje de cambio al Raspberry asociado al usuario.
8. Cuando es recibida la orden en el Raspberry Pi éste responde con un mensaje de ACK indicando que ha sido recibida.

```
{"idmsg":332351,"iduser":"254630","address":"1/2/1","iddevice":"18","idusersend":"1000-phone-224","typeuser":"raspi","state":"0","time":"27/01/2017-13:38:11","type":"ack","value":"received"}
```

9. Dicho mensaje de power cuando es recibido por el Raspberry pasa a ser procesado y enviado al bus KNX como telegrama. Tras realizar esta acción es envía un mensaje de tipo ACK con valor recibido y terminado. Esto indica que la petición del usuario se está realizando.

```
{"idmsg":332357,"iduser":"254630","address":"1/2/1","iddevice":"18","idusersend":"1000-phone-224","typeuser":"raspi","state":"0","time":"27/01/2017-13:38:11","type":"ack","value":"terminate"}
```

10. Cuando el mensaje de ACK de valor terminado es recibido por la aplicación, esta muestra al usuario un progreso en el envío de la orden. Éste progreso se visualizará por el dispositivo con una animación y es realizado de forma estática, la animación terminara tras 15 segundos. Hay 2 posibles casos:
  - a. Subir persiana: en caso de ser pulsado el botón de subida la pantalla del dispositivo pasará a color negro, también se mostrará una barra de progreso la cual está calculada para completarse en 15 segundo. El proceso de dicha animación es pasar del fondo negro a claro, de forma síncrona también se visualizará progresivamente una imagen simulando el sol.
  - b. Bajar persiana: en caso de ser pulsado el botón de bajar persiana la pantalla del dispositivo pasará a color blanco y se visualizará una imagen simulando el sol, también se mostrará una barra de progreso la cual está calculada para completarse en 15 segundos. El proceso de dicha animación es pasar del

fondo blanco a negro, de forma síncrona la imagen del sol desaparecerá progresivamente simulando la puesta de sol.

11. El Raspberry Pi envía un mensaje de tipo “change” indicando que se ha registrado un cambio en un dispositivo KNX del hogar. Éste envía al servidor la dirección KNX que ha sido modificada, el nuevo valor y el usuario que ha realizado el cambio. Formato del mensaje:

```
{'type': 'change', 'typeuser': 'raspi', 'state': '0', 'idmsg': 762, 'iduser': '254630', 'userchange': '1000', 'idusersend': '0', 'address': '1/2/1', 'value': 'device', 'time': '27/01/2017-13:53:25'}
```

12. El servidor de aplicación tras recibir el anterior mensaje realiza una actualización en la base de datos actualizando el valor del estado del dispositivo, tras esto envía a todos los clientes conectados excepto al usuario que ha realizado la acción si lo hubiera un mensaje de tipo change indicando que ha sido realizado un cambio en el hogar. Esto hace que un cambio sea recibido por todos los usuarios sin tener que enviar mensajes al servidor de actualización de valores. Formato del mensaje:

```
{"idmsg":332462,"iduser":"0","address":"1/2/1","iddevice":"18","idusersend":"1000-phone-224","userchange":"vallemar","typeuser":"server","time":"27/01/2017-14:53:48","state":"0","type":"change","value":"estatedevice"}
```

13. Cuando el mensaje de tipo change es recibido por la aplicación si es usuario está en la pantalla de persianas la animación de subida o bajada de persianas es iniciada. Esto es de gran utilidad para el usuario ya que así se logra una monitorización actualizada en todo momento.
14. Cuando el usuario se encuentra observando la animación durante la subida/bajada de la persiana aparece un botón en la pantalla que permite parar el recorrido. Si éste es pulsado se envía el siguiente formato.

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"254630","time":"27/01/2017-14:16:33","type":"power","value":"0","iddevice":"18","typedevice":"2","address":"1/2/2","idmsg":332634}
```

Tras éste mensaje se envían nuevamente los mensajes de ACK y change. Se consigue una total interacción en la subida o bajada del dispositivo entre todos los usuarios del hogar.

### 7.6.2.3 DESCRIPCIÓN CASO DE USO GESTIÓN DE ELECTRODOMÉSTICOS

Desde la pantalla de electrodomésticos puede monitorizarse el estado de los distintos electrodomésticos del hogar y la gestión de estos. Cuando el usuario entra a éste apartado se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste de tipo “requestdevices” y valor “3” (Véase tipos de dispositivos). Formato del mensaje:

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"0","time":"27/01/2017-13:56:18","type":"requestdevices","value":"3","idmsg":332481}
```

2. El servidor realiza una consulta a la base de datos y obtiene los dispositivos de tipo “3” asociados al hogar, también su estado. Estos datos son enviados al usuario.  
Formato del mensaje:

```
{ "idmsg":332483,"iduser":"0","typedevice":"3","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-14:56:47","type":"requestdevices","value":[{"description":"Horno","state":"1","id":"14","type":"3","actions":[{"address":"2/2/2","action":"EIS1"}]}]}
```

En el caso de que no haber electrodomésticos compatibles con KNX instalado en el hogar el servidor responde con el siguiente mensaje:

```
{ "idmsg":555,"iduser":"0","typedevice":"3","idusersend":"1000-phone-90","typeuser":"server","time":"29/12/2016-17:52:06","type":"requestdevices","value":[]}
```

3. La aplicación tras recibir los datos los parsea y se muestra de forma clara al usuario informando el estado de la totalidad de los electrodomésticos.
4. Si el usuario quiere modificar el estado de una de los electrodomésticos puede hacerlo pulsando el botón asociado al dispositivo. Si esto sucede la aplicación envía al servidor un mensaje de tipo Power, el cual es el único tipo de mensaje para los electrodomésticos.

```
{ "typeuser":"user","iduser":"1000-phone-224","idusersend":"254630","time":"27/01/2017-13:59:14","type":"power","value":"1","iddevice":"15","typedevice":"3","address":"2/2/3","idmsg":332502}
```

5. Tras ser recibido el mensaje de cambio de estado en el servidor éste responde con un mensaje de tipo ACK para indicar que se ha recibido. Cuando el mensaje de AKC es recibido por la aplicación muestra al usuario un progreso en él envió de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:
  - a. Encendido: El icono que indica el apagado del electrodoméstico pasa al icono que indica que ha sido recibido por el servidor.



Figura 17. Primer ACK encendido electrodomésticos

- b. Apagado: El icono que indica el encendido del electrodoméstico pasa al icono que indica que ha sido recibido por el servidor.



Figura 18. Primer ACK apagado electrodomésticos

- c. Formato del mensaje de ACK:

```
{"idmsg":332503,"iduser":"0","iddevice":"15","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-14:59:43","state":"1","type":"ack","value":"serverreceived_change"}
```

6. El servidor envía la orden a el Raspberry Pi asociada al usuario.
7. El Raspberry Pi recibe la orden y envía un mensaje al usuario de tipo ACK de tipo recibido indicando que se ha recibido la orden y se va a proceder a realizar el cambio.  
Formato del mensaje:

```
{"idmsg":332504,"iduser":"254630","address":"2/2/3","iddevice":"15","idusersend":"1000-phone-224","typeuser":"raspi","state":"1","time":"27/01/2017-13:59:20","type":"ack","value":"received"}
```

8. Cuando el mensaje de ACK de tipo recibió enviado por el Raspberry Pi es recibido por la aplicación muestra al usuario un progreso en él envió de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:
  - a. Encendido/Apagado: el icono que indica que había sido recibido por el servidor cambia indicando que ha sido recibido por el hogar y que la petición se está procesando.



Figura 19. Segundo ACK electrodomésticos

9. Tras ser enviado el telegrama al bus KNX el Raspberry envía un mensaje de tipo ACK con valor terminado. Esto indica que se ha realizado la petición del usuario.  
Formato del mensaje:

```
{'type': 'ack', 'typeuser': 'raspi', 'state': '1', 'idmsg': 779, 'iduser': '254630', 'idusersend': '1000-phone-224', 'address': '2/2/3', 'iddevice': '15', 'value': 'terminate', 'time': '27/01/2017-13:59:20'}
```

10. Cuando el mensaje de ACK de tipo terminado enviado por el Raspberry Pi es recibido por la aplicación muestra al usuario un progreso en él envió de la orden. La bombilla asociada en la pantalla al dispositivo cambiara de estado. Hay 2 posibles casos, encendido o apagado:
  - a. Encendido: El icono que indica el proceso de petición en el hogar cambia al icono de encendido indicando que la orden ha sido completada exitosamente.



Figura 20. Tercer ACK encendido electrodomésticos, final

- b. Apagado: El icono que indica el proceso de petición en el hogar cambia al icono de apagado indicando que la orden ha sido completada exitosamente.



Figura 21. Tercer ACK apagado electrodomésticos, final

11. El Raspberry Pi envía un mensaje de tipo “change” indicando que se ha registrado un cambio en un dispositivo KNX del hogar. Éste envía al servidor la dirección KNX que ha sido modificada, el nuevo valor y el usuario que ha realizado el cambio. Formato del mensaje:

```
{'type': 'change', 'typeuser': 'raspi', 'state': '1', 'idmsg': 779, 'iduser': '254630', 'userchange': '1000', 'idusersend': '0', 'address': '2/2/3', 'value': 'device', 'time': '27/01/2017-13:59:20'}
```

12. El servidor de aplicación tras recibir el anterior mensaje realiza una actualización en la base de datos actualizando el valor del estado del dispositivo, tras esto envía a todos los clientes conectados excepto al usuario que ha realizado la acción si lo hubiera un mensaje de tipo change indicando que ha sido realizado un cambio en el hogar. Esto hace que un cambio sea recibido por todos los usuarios sin tener que enviar mensajes al servidor de actualización de valores. Formato del mensaje:

```
{"idmsg":332510,"iduser":"0","address":"2/2/3","iddevice":"15","idusersend":"1000-phone-224","userchange":"vallemar","typeuser":"server","time":"27/01/2017-14:59:43","state":"1","type":"change","value":"estatedevice"}
```

13. Cuando el mensaje de tipo change es recibido por la aplicación si es usuario está en la pantalla de electrodomésticos realizara el cambio del dispositivo e indicara quien ha sido el responsable de dicho cambio. Esto es de gran utilidad para el usuario ya que así se logra una monitorización actualizada en todo momento.

### 7.6.3 DESCRIPCIÓN MÓDULO CLIMATIZACIÓN

El módulo de climatización de la aplicación dota al usuario de los beneficios del control remoto del hogar de los dispositivos que son descritos. A continuación, se desglosa como actúa la aplicación ante la interacción de los usuarios con la aplicación.

#### 7.6.3.1 DESCRIPCIÓN CASO DE USO GESTIÓN DE CALEFACCIÓN

Desde la pantalla de Calefacción puede monitorizarse el estado de esta y es posible la gestión por medio de KNX. Esto significa que la calefacción a controlar desde éste apartado necesita ser compatible con el estándar KNX, si el usuario no dota a su hogar de éste servicio desde éste apartado no puede ser gestionada la calefacción. Cuando el usuario entra a éste apartado se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste de tipo “requestdevices” y valor “6” (Véase tipos de dispositivos). Formato del mensaje:

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"0","time":"27/01/2017-14:26:01","type":"requestdevices","value":"6","idmsg":646369}
```

2. El servidor realiza una consulta a la base de datos y obtiene los dispositivos de tipo "6" asociados al hogar, también su estado. Estos datos son enviados al usuario. Formato del mensaje:

```
{ "idmsg":789519,"iduser":"0","typedevice":"6","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-15:51:17","type":"requestdevices","value":[{"description":"Calefacción","state":"0","id":"25","type":"6","actions":[{"address":"4/0/2","action":"EIS6"}]}]}
```

En el caso de que no haber calefacción compatible con KNX instalado en el hogar el servidor responde con el siguiente mensaje:

```
{ "idmsg":555,"iduser":"0","typedevice":"6","idusersend":"1000-phone-90","typeuser":"server","time":"29/12/2016-17:52:06","type":"requestdevices","value":[]}
```

3. La aplicación tras recibir los datos los parsea y se muestra de forma clara al usuario informando el estado.
4. Si el usuario quiere modificar el estado de una de la calefacción puede hacerlo pulsando el botón de encendido/apagado. Si el usuario desea encender la calefacción la aplicación pedirá la temperatura deseada y envía al servidor un mensaje de tipo Dimmer. Formato del mensaje.

```
{ "typeuser":"user","iduser":"1000-phone-224","idusersend":"254630","time":"27/01/2017-14:51:34","type":"dimmer","value":"28","iddevice":"25","address":"4/0/2","typedevice":"6" }
```

5. Tanto si el usuario manda un mensaje de encendido o apagado la traza del programa será idéntica, sólo cambia los valores de mensajes value y state. Tras ser recibido el mensaje de cambio de estado en el servidor éste responde con un mensaje de tipo ACK para indicar que se ha recibido. Cuando el mensaje de ACK es recibido por la aplicación en esta ocasión no muestra nada en la aplicación, el fin de éste mensaje es para la gestión del canal establecido entre el servidor y el cliente. Formato del mensaje de ACK:

```
{ "idmsg":789527,"iduser":"0","iddevice":"25","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-15:52:03","state":"28","type":"ack","value":"serverreceived_change" }
```

6. El servidor envía la orden a el Raspberry Pi asociada al usuario.
7. El Raspberry Pi recibe la orden y envía un mensaje al usuario de tipo ACK de tipo recibido indicando que se ha recibido la orden y se va a proceder a realizar el cambio. Formato del mensaje:

```
{ "idmsg":789529,"iduser":"254630","address":"4/0/2","iddevice":"25","idusersend":"1000-phone-224","typeuser":"raspi","state":"28","time":"27/01/2017-14:51:40","type":"ack","value":"received" }
```

8. Cuando el mensaje de ACK de tipo recibió enviado por el Raspberry Pi es recibido por la aplicación tampoco muestra nada al usuario, éste mensaje sigue siendo para la gestión del canal.
9. Tras ser enviado el telegrama al bus KNX el Raspberry envía un mensaje de tipo ACK con valor terminado. Esto indica que se ha realizado la petición del usuario. Formato del mensaje:

```
{"idmsg":789533,"iduser":"254630","address":"4/0/2","iddevice":"25","idusersend":"1000-phone-224","typeuser":"raspi","state":"28","time":"27/01/2017-14:51:40","type":"ack","value":"terminate"}
```

10. Cuando el mensaje de ACK de tipo terminado enviado por el Raspberry Pi es recibido por la aplicación muestra al usuario un progreso en él envió de la orden.
11. El Raspberry Pi envía un mensaje de tipo “change” indicando que se ha registrado un cambio en un dispositivo KNX del hogar. Éste envía al servidor la dirección KNX que ha sido modificada y el nuevo valor. Formato del mensaje:

```
'typeuser': 'raspi', 'iduser': '254630', 'address': '4/0/2', 'state': '28', 'time': '27/01/2017-14:51:40', 'idmsg': 945, 'userchange': '1000', 'type': 'change', 'idusersend': '0', 'value': 'device'}
```

12. El servidor de aplicación tras recibir el anterior mensaje realiza una actualización en la base de datos actualizando el valor del estado del dispositivo, tras esto envía a todos los clientes conectados excepto al usuario que ha realizado la acción si lo hubiera un mensaje de tipo change indicando que ha sido realizado un cambio en el hogar. Esto hace que un cambio sea recibido por todos los usuarios sin tener que enviar mensajes al servidor de actualización de valores. Formato del mensaje:

```
{"idmsg":789531,"iduser":"0","address":"4/0/2","iddevice":"25","idusersend":"1000-phone-224","userchange":"1000","typeuser":"server","time":"27/01/2017-15:52:03","state":"28","type":"change","value":"estatedevice"}
```

13. Cuando el mensaje de tipo change es recibido por la aplicación si es usuario está en la pantalla de calefacción se modifica el valor del dispositivo a encendido o apagado y el usuario que ha realizado el cambio. Esto es de gran utilidad para el usuario ya que así se logra una monitorización actualizada en todo momento.

### 7.6.3.2 DESCRIPCIÓN CASO DE USO GESTIÓN DE AIRE ACONDICIONADO

Éste módulo ha sido realizado idéntico al de calefacción, lo único que cambiaría es la dirección de grupo del dispositivo.

### 7.6.4 DESCRIPCIÓN MÓDULO SEGURIDAD

El módulo de Seguridad de la aplicación dota al usuario de los beneficios del control remoto del hogar de los dispositivos que son descritos. A continuación, se desglosa como actúa la aplicación ante la interacción de los usuarios con la aplicación.

#### 7.6.4.1 DESCRIPCIÓN CASO DE USO IMÁGENES

Desde la pantalla de imágenes un usuario puede acceder a su hogar para realizar una imagen capturada por las cámaras disponibles en esta. Para esto es necesario que en al Raspberry pi del hogar estén conectadas 1 o más cámaras al bus universal serial bus (USB), si no hay ninguna cámara USB éste apartado del servicio no podrá realizar ninguna acción. Cuando el usuario entra a éste apartado se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste de tipo “requestdevices” y valor “10” indicando que es de tipo: cámara (Véase tipos de dispositivos). Formato del mensaje:

```
{ "typeuser": "user", "iduser": "1000-phone-224", "idusersend": "254630", "time": "27/01/2017-15:24:34", "type": "requestdevices", "value": "10", "idmsg": "789856" }
```

2. Cuando éste mensaje es recibido en el servidor éste es reenviado al Raspberry Pi.
3. Tras recibir el mensaje el Raspberry pi procesa la solicitud, con esta se detecta el número de cámaras conectadas a los puertos USB y responde al usuario con un mensaje indicando el número de cámaras conectadas. Formato del mensaje:

```
{ "idmsg": "789859", "iduser": "254630", "typedevice": "10", "address": "video0", "idusersend": "1000-phone-224", "typeuser": "raspi", "time": "27/01/2017-15:24:40", "type": "requestdevices", "value": "number" }
```

Como se puede observar en el mensaje anterior se envía una respuesta de tipo 'requestdevices' y valor número, éste indicando que el propósito de éste mensaje es indicar el número de cámaras que están actualmente conectadas.

También se observa el campo address el cual indica el nombre de una de las cámaras conectadas. Éste mensaje es enviado por cada una de las cámaras que están conectadas al Raspberry pi.

4. Cuando el dispositivo Android recibe estos mensajes crea una lista con todas las cámaras disponibles. Si el usuario interactúa con la aplicación pulsado en el botón relacionado para tomar una imagen nueva esta muestra el número de cámaras disponibles para escoger una y realizar la captura. Cuando esto ocurre se envía el siguiente mensaje: Formato del mensaje:

```
{ "iduser": "1000-phone-224", "idusersend": "254630", "time": "27/01/2017-15:26:21", "typeuser": "user", "type": "camera", "value": "capture", "address": "video0", "idmsg": "789865" }
```

5. Cuando éste mensaje es recibido en el Raspberry pi éste procesa la solicitud y es realiza la captura.
6. Tras realizar la imagen se utiliza el servidor de imágenes desarrollado en Python mediante el Frameworks Django como anteriormente había sido mencionado. El Raspberry pi realiza una solicitud HTTP al servidor de imágenes enviado su usuario, contraseña, nombre de la imagen e imagen. Éste nombre de la imagen se establece por la hora en la que la captura ha sido realizada, un ejemplo es: 02012017190339.

El formato de éste nombre se establece por: Día-Mes-Año-hora-minutos-segundos. Tras establecer éste nombre se envía al servidor, el nombre de la imagen es cifrado con la contraseña aes del Raspberry Pi y se envía. La petición realizada es la siguiente:

```
POST http://vallemarjaen.ddns.net:80/upload_photo_raspberry/
Authorization:
2:19Bog9UPLiF8AmDFo61xhDdi8IpMQLrbQgrlq9Vo73JUee9ZUGinO1KIr8BzyV5Cr
EdO+17zo00kLNfvGNXnIQ==
Content-Type: multipart/form-data; boundary=54824fb080b14e3aa5ade0afa5d91520
Content-Length: 154
Content-Disposition: form-data; name="img_home"; filename="
WvuTPGoSRE2rFbe0DB0Ig1kapn7pVIGEwQpt+CJBeISlxEMqbrdxQiHdBMPb5JME
kdCwrZjSEhDAPGH+z5QVLA=="
```

7. Tras llegar esta petición al servidor éste almacena la imagen con el nombre original, es decir, descifrando el nombre de la imagen. También almacena en la base de datos una referencia, esta referencia la imagen. Cuando éste guarda la imagen en el servidor realiza un hash el nombre de la imagen y asigna el nombre de la imagen con dicho hash en la base de datos. Esto se realiza para que cuando el dispositivo Android realice la petición lo haga con el hash del nombre de la imagen para mayor seguridad.
8. Cuando todo el proceso anterior ha sido completado el Raspberry Pi responde al cliente para indicar que la imagen está disponible en el servidor de imágenes, el formato del mensaje es el siguiente:

```
{"idmsg":789870,"iduser":"254630","address":"DevdsMvqfO0l3mxLjU7DZ7n5mSs7h3
xEJjKvp+rZU+M=","idusersend":"1000-phone-
224","typeuser":"raspi","state":"terminate","time":"27/01/2017-
15:26:29","type":"ack","value":"capture"}
```

9. Tras recibir el mensaje anterior el dispositivo Android realizara una petición HTTP Get al servidor de imagen con el siguiente formato.

```
GET
/download_photo/?namephoto="
DevdsMvqfO0l3mxLjU7DZ7n5mSs7h3xEJjKvp+rZU+M="
Content-Type: text/plain
Host: vallemarjaen.ddns.net:80
Authorization:
1000:B3KqiOS1Cgy6YcENsUOcBNbciXOEpsZ9xrywAzO/1RgPl8SEe9+JTWtXgdkG
czpAAItj/jCIjtv6qzh4Zdq1lw==
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; Aquaris X5 Build/MMB29M)
Accept-Encoding: gzip
Connection: Keep-Alive
```

Si todo el proceso anterior ha concluido correctamente, la nueva imagen será descargada, almacenada y mostrada en el dispositivo del usuario. Si la imagen no está disponible éste responderá con un código de estado= 400 y el usuario será informado del error.

## 7.6.5 DESCRIPCIÓN MÓDULO MULTIMEDIA

El módulo de multimedia de la aplicación dota al usuario de los beneficios del control remoto del hogar de los dispositivos que son descritos. A continuación, se desglosa como actúa la aplicación ante la interacción de los usuarios con la aplicación.

### 7.6.5.1 DESCRIPCIÓN CASO DE USO MANDO A DISTANCIA

Desde la pantalla de Mando a distancia el usuario podrá ver los mandos a distancia que tiene agregados al hogar y realizar la acción de desee con ellos. Estas acciones están asociadas a los elementos infrarrojos instalados en el Raspberry Pi.

Cuando el usuario entra a éste apartado se ejecutan las siguientes instrucciones:

1. La aplicación envía un mensaje al servidor. Éste de tipo “requestdevices” con valor “11” (Véase tipos de dispositivos). Indicando que es pide los dispositivos de mandando a distancia. El formato de mensaje es el siguiente:

```
{"typeuser":"user","iduser":"1000-phone-224","idusersend":"0","time":"27\01\2017-15:30:23","type":"requestdevices","value":"11","idmsg":789882}
```

2. El servidor realiza una consulta a la base de datos y obtiene los dispositivos de tipo “11” asociados al hogar. Estos datos son enviados al usuario. Formato del mensaje:

```
{"idmsg":789884,"iduser":"0","typedevice":"11","idusersend":"1000-phone-224","typeuser":"server","time":"27/01/2017-16:30:52","type":"requestdevices","value":[{"description":"Aire Salita","state":"0","id":"48","type":"11","actions":[{"address":"IRAIR1","action":"IRAIR"}]},{"description":"Musica Salón","state":"0","id":"53","type":"11","actions":[{"address":"IRMUSIC1","action":"IRMUSIC"}]},{"description":"Television Salón","state":"0","id":"54","type":"11","actions":[{"address":"IRTV1","action":"IRTV"}]}}
```

Los dispositivos infrarrojos disponibles están limitadas a 3 tipos. Esta limitación es sólo estética ya que es producida a que son los tipos de interfaces que se han desarrollado en la aplicación Android. Estos están bien definidos en la aplicación Android y en el Raspberry Pi, los tipos son los siguientes:

- Dispositivo de Televisión.
- Dispositivo de Aire Acondicionado.
- Dispositivo Musical.
  - Type. Valor: requestdevices. Indica que responde a un mensaje de requestdevices previamente enviado por el usuario.
  - Typedevice. Valor: 11. Indica que los dispositivos que envía son de tipo 11, mando a distancia.
  - Value. Valor: Array en formato JSON que describe los distintos dispositivos que existen en el hogar, éste tiene los siguientes campos:
    - Action. Posibles valores: IRTV, IRAIR, IRMUSIC.
      - IRTV: indica que es un control remoto de televisión.
      - IRAIR: indica que es un control remoto de aire acondicionado.

- IRMUSIC: indica que es un control remoto de música.
- address. Éste campo es el id del mando a distancia en el Raspberry Pi. Apunta a un archivo en el cual están almacenados los pulsos asociados cada botón y cada mando a distancia. La ruta de éste archivo es: /etc/lirc/lircd.conf. Éste archivo es necesario para la simulación de los pulsos enviados por el mando a distancia. El formato en el que se autogenera éste id es:
  - Tipo de dispositivo y numero identificador. Un ejemplo: IRTV3. Éste es el identificar en el archivo anteriormente mencionado.

3. Tras recibir el mensaje anterior, se mostrará al usuario en forma de lista todos los mandos a distancia disponibles. Éste pinchando en alguno de éste se mostrará una pantalla donde aparecen los botones disponibles para dicho mando a distancia.

En éste apartado serán expuestos los distintos mensajes enviados por la aplicación ante el evento que se produce tras pulsar un botón por el usuario. Serán expuestos los distintos id de botones para los distintos botones disponibles. Las opciones disponibles son los siguientes.

### *MANDO A DISTANCIA DE TELEVISIÓN*

1. Botón: Encender/Apagar.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:41:47","type":"remote","value":"click","address":"IRTV1","button":"KEY_POWER","idmsg":274}
```

- Type. Valor: remote. Indica que es un mensaje de tipo control remoto.
- Vale. Valor: click. Indica que es un mensaje de toque de botón.
- Address. Valor: id de mando a distancia. Éste valor diferencia cada mando a distancia.
- Button. Valor: id del botón. Éste valor indica el botón pulsado por el usuario.

2. Botón: Mute.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:42:09","type":"remote","value":"click","address":"IRTV1","button":"KEY_MUTE","idmsg":279}
```

3. Botón: 1.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:42:25","type":"remote","value":"click","address":"IRTV1","button":"KEY_1","idmsg":283}
```

4. Botón: 2.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:42:37","type":"remote","value":"click","address":"IRTV1","button":"KEY_2","idmsg":287}
```

5. Botón: 3.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:42:50","type":"remote","value":"click","address":"IRTV1","button":"KEY_3","idmsg":292}
```

6. Botón: 4.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:43:04","type":"remote","value":"click","address":"IRTV1","button":"KEY_4","idmsg":296}
```

7. Botón: 5.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:43:35","type":"remote","value":"click","address":"IRTV1","button":"KEY_5","idmsg":304}
```

8. Botón: 6.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:43:38","type":"remote","value":"click","address":"IRTV1","button":"KEY_6","idmsg":308}
```

9. Botón: 7.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:44:18","type":"remote","value":"click","address":"IRTV1","button":"KEY_7","idmsg":313}
```

10. Botón: 8.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:44:18","type":"remote","value":"click","address":"IRTV1","button":"KEY_8","idmsg":316}
```

11. Botón: 9.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:45:47","type":"remote","value":"click","address":"IRTV1","button":"KEY_9","idmsg":323}
```

12. Botón: 0.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:45:47","type":"remote","value":"click","address":"IRTV1","button":"KEY_0","idmsg":327}
```

13. Botón: Canal arriba.

```
{"typeuser":"user","iduser":"1000-phone-151","idusersend":"2","time":"14\01\2017-00:46:13","type":"remote","value":"click","address":"IRTV1","button":"KEY_UP","idmsg":332}
```

14. Botón: Canal abajo.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:46:14", "type": "remote", "value": "click", "address": "IRTV1", "button": "KEY_DOWN", "idmsg": 336 }
```

15. Botón: Volumen arriba.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:46:41", "type": "remote", "value": "click", "address": "IRTV1", "button": "KEY_VOLUMEUP", "idmsg": 340 }
```

16. Botón: Volumen abajo.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:46:42", "type": "remote", "value": "click", "address": "IRTV1", "button": "KEY_VOLUMEDOWN", "idmsg": 344 }
```

17. Botón: Ok.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:47:04", "type": "remote", "value": "click", "address": "IRTV1", "button": "KEY_OK", "idmsg": 349 }
```

#### *MANDO A DISTANCIA DE AIRE ACONDICIONADO*

1. Botón: Encender.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:49:52", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_POWER", "idmsg": 356 }
```

2. Botón: Apagar.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:51:19", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_CLOSE", "idmsg": 362 }
```

3. Botón: Grado Arriba.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:51:40", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_UP", "idmsg": 366 }
```

4. Botón: Grado Abajo.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:51:42", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_DOWN", "idmsg": 370 }
```

5. Botón: Activar modo Swing.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:52:07", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_0", "idmsg": 375 }
```

6. Botón: Mover aspas.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:52:12", "type": "remote", "value": "click", "address": "IRAIR2", "button": "KEY_1", "idmsg": 379 }
```

### *MANDO A DISTANCIA DE MÚSICA*

1. Botón: Encender/Apagar.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:53:31", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_POWER", "idmsg": 384 }
```

2. Botón: Mute.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:54:54", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_MUTE", "idmsg": 390 }
```

3. Botón: Play.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:55:13", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_PLAY", "idmsg": 394 }
```

4. Botón: Siguiente canción.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:55:14", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_UP", "idmsg": 398 }
```

5. Botón: Anterior canción.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:55:33", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_DOWN", "idmsg": 403 }
```

6. Botón: Arriba volumen.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:55:42", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_UP", "idmsg": 407 }
```

7. Botón: Abajo volumen.

```
{ "typeuser": "user", "iduser": "1000-phone-151", "idusersend": "2", "time": "14\01\2017-00:55:53", "type": "remote", "value": "click", "address": "IRMUSIC1", "button": "KEY_VOLUMEDOWN", "idmsg": 411 }
```

Se puede distinguir que en las acciones asociadas a los mandos a distancia de televisiones y de música el botón encender y apagar son los mismos. Esto no sucede con los mandos a distancia del aire acondicionado. En estos la secuencia de pulsos que se emiten son distintos.

También se ha tenido la problemática de que al intentar interactuar con el programa Lirc para el caso de los mandos de aire acondicionado éste no funcionaba.

El problema producido era porque los bits enviados ante el evento de pulsar un botón de los mandos a distancia de aire acondicionados modernos son mayores que los mandos a distancia como por ejemplo de televisión, exactamente manda 48 bits.

Otra opción que puede ser realizada desde éste módulo de aplicación es la incorporación de un nuevo mando a distancia asociado al hogar. Para esto primero hay que elegir el tipo de mando que el usuario quiere registrar, las 3 opciones son las descritas anteriormente: Aire acondicionado, televisión y música.

Si el usuario elige esta opción le llevara a una pantalla indicando unas instrucciones previamente, lo cual es poner unas pilas cargadas al mando al cual desea realizar el grabado y posicionarse justo delante del receptor infrarrojo. Esto es para asegurar un correcto grabado del mando a distancia, ya que si no podría grabarse alguno de los botones erróneamente. En esta pantalla también es pedido un nombre identificador para asignar al mando a distancia.

Tras esta pantalla se envía al Raspberry Pi un mensaje indicando que se quiere grabar un mando a distancia. El formato de mensaje es el siguiente:

```
{"typeuser":"user","iduser":"1000-phone-154","idusersend":"2","time":"14/01/2017-13:04:51","type":"remote","value":"save","typedevice":"4","nameremote":"Nombre mando televisión","idmsg":457}
```

- Value. Valor: save. Indica que la acción del mensaje es para realizar el grabado de un mando a distancia.
- Typedevice. Valor. Éste identifica el tipo de mando a distancia. Puede tener múltiples valores numéricos:
  - 4: Indica Televisión.
  - 5: indica Aire acondicionado.
  - 12: indica Musical.

Tras recibir éste mensaje en el Raspberry Pi éste empezara ejecutara los métodos desarrollados para el fin de realizar el grabado. Se inicia e interactúa con el programa Lirc en modo grabación.

Éste pedirá al usuario que toque las teclas del control remoto una a una con un sencillo tutorial. Tras finalizar el grabado se enviará un mensaje al usuario indicando que ha finalizado el proceso de grabación. El formato de mensajes es el siguiente.

### *GRABADO DE MANDO DE AIRE ACONDICIONADO*

1. Indica en la pantalla al usuario que pulse el botón de encendido

```
{"idmsg":949,"button":"Pulsa Encender","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:05","type":"remote","value":"save"}
```

2. Indica en la pantalla al usuario que pulse el botón de apagado

```
{"idmsg":950,"button":"Pulsa Apagar","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:06","type":"remote","value":"save"}
```

3. Indica en la pantalla al usuario que pulse el botón grado arriba

```
{"idmsg":951,"button":"Pulsa Grado Arriba","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:08","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que pulse el botón grado abajo

```
{"idmsg":952,"button":"Pulsa Grado Abajo","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:10","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que pulse el botón swing

```
{"idmsg":953,"button":"Pulsa Swing","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:12","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que pulse el botón regular aspas de aire

```
{"idmsg":954,"button":"Pulsa Regular Aspas","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:14","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que pulse cualquier botón para terminar botón

```
{"idmsg":955,"button":"¡Pulsa Cualquier Boton!","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:17","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que ya ha terminado y puede empezar a disfrutar del nuevo mando a distancia.

```
{"idmsg":956,"button":"¡Hemos terminado!\nSe reiniciara el Raspberry para aplicar los cambios. Espere unos segundos.","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:13:20","type":"remote","value":"save"}
```

- Tras todo el proceso anterior el Raspberry Pi envía al servidor un mensaje indicando que ha terminado un proceso de grabación, éste con los datos proporcionados por el Raspberry Pi guarda dicho mando en su base de datos para poder ser consumido posteriormente por todos los integrantes del hogar. El formato del mensaje enviado es:

```
{'typedevice': '5', 'idmsg': 958, 'iduser': '2', 'typeuser': 'raspi', 'nameremote': 'Nombre mando Aire Acondicionado', 'address': 'IRAIR3', 'type': 'remote', 'value': 'save', 'idusersend': '0', 'time': '14/01/2017-16:13:20'}
```

### GRABADO DE MANDO DE TELEVISIÓN

- Indica en la pantalla al usuario que pulse el botón de encender/apagar.

```
{"idmsg":995,"button":"Pulse Encender/Apagar","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:24:17","type":"remote","value":"save"}
```

- Indica en la pantalla al usuario que pulse el botón de Canal Arriba.

```
{ "idmsg":996,"button":"Pulse Canal Arriba","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:18","type":"remote","value":"save" }
```

3. Indica en la pantalla al usuario que pulse el botón Canal Abajo.

```
{ "idmsg":997,"button":"Pulse Canal Abajo","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:22","type":"remote","value":"save" }
```

4. Indica en la pantalla al usuario que pulse el botón Mute.

```
{ "idmsg":998,"button":"Pulse Mute","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:24:24","type":"remote","value":"save" }
```

5. Indica en la pantalla al usuario que pulse el botón OK.

```
{ "idmsg":999,"button":"Pulse OK","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-  
16:24:27","type":"remote","value":"save" }
```

6. Indica en la pantalla al usuario que pulse el botón Volumen Arriba.

```
{ "idmsg":1000,"button":"Pulse Volumen Arriba","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:30","type":"remote","value":"save" }
```

7. Indica en la pantalla al usuario que pulse el botón Volumen Abajo.

```
{ "idmsg":1001,"button":"Pulse Volumen Abajo","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:33","type":"remote","value":"save" }
```

8. Indica en la pantalla al usuario que pulse el botón 0.

```
{ "idmsg":1002,"button":"Pulse 0","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:24:35","type":"remote","value":"save" }
```

9. Indica en la pantalla al usuario que pulse el botón 1.

```
{ "idmsg":1003,"button":"Pulse 1","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:24:38","type":"remote","value":"save" }
```

10. Indica en la pantalla al usuario que pulse el botón 2.

```
{ "idmsg":1004,"button":"Pulse 2","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:24:41","type":"remote","value":"save" }
```

11. Indica en la pantalla al usuario que pulse el botón 3.

```
{ "idmsg":1005,"button":"Pulse 3 ","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-  
16:24:47","type":"remote","value":"save" }
```

12. Indica en la pantalla al usuario que pulse el botón 4.

```
{"idmsg":1006,"button":"Pulse 4 ","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:24:50","type":"remote","value":"save"}
```

13. Indica en la pantalla al usuario que pulse el botón 5.

```
{"idmsg":1007,"button":"Pulse 5 ","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:24:52","type":"remote","value":"save"}
```

14. Indica en la pantalla al usuario que pulse el botón 6.

```
{"idmsg":1008,"button":"Pulse 6 ","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:24:55","type":"remote","value":"save"}
```

15. Indica en la pantalla al usuario que pulse el botón 7.

```
{"idmsg":1009,"button":"Pulse 7","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:24:58","type":"remote","value":"save"}
```

16. Indica en la pantalla al usuario que pulse el botón 8.

```
{"idmsg":1011,"button":"Pulse 8","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:25:00","type":"remote","value":"save"}
```

17. Indica en la pantalla al usuario que pulse el botón 9.

```
{"idmsg":1012,"button":"Pulse 9 ","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:25:02","type":"remote","value":"save"}
```

18. Indica en la pantalla al usuario que pulse cualquier botón para terminar. Éste paso esta implementado para comprobar que el programa ha salido correctamente del bucle de grabado.

```
{"idmsg":1013,"button":"Pulse cualquier botón hasta que se avise","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:25:04","type":"remote","value":"save"}
```

19. Indica en la pantalla al usuario que ya ha terminado y puede empezar a disfrutar del nuevo mando a distancia.

```
{"idmsg":1014,"button":"Hemos terminado!\n\nPuede empezar a disfrutar de su nuevo mando a distancia","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:25:15","type":"remote","value":"save"}
```

20. Tras esto el Raspberry Pi envía al servidor un mensaje indicando que ha terminado un proceso de grabación, éste con los datos proporcionados por el Raspberry Pi guarda dicho mando en su base de datos para poder ser consumido posteriormente por todos los integrantes del hogar. El formato del mensaje enviado es

```
{'typedevice': '4', 'idmsg': 1015, 'iduser': '2', 'typeuser': 'raspi', 'nameremote': 'Nombre mando televisión', 'address': 'IRTV4', 'type': 'remote', 'value': 'save', 'idusersend': '0', 'time': '14/01/2017-16:25:15 '}
```

## GRABADO DE MANDO DE MUSICA

1. Indica en la pantalla al usuario que pulse el botón de encender/apagar.

```
{"idmsg":995,"button":"Pulse Encender/Apagar","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:17","type":"remote","value":"save"}
```

2. Indica en la pantalla al usuario que pulse el botón de Canción Arriba.

```
{"idmsg":996,"button":" Pulse Canción Arriba","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:18","type":"remote","value":"save"}
```

3. Indica en la pantalla al usuario que pulse el botón Canción Abajo.

```
{"idmsg":997,"button":" Pulse Canción Abajo","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:22","type":"remote","value":"save"}
```

4. Indica en la pantalla al usuario que pulse el botón Mute.

```
{"idmsg":998,"button":"Pulse Mute","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:24:24","type":"remote","value":"save"}
```

5. Indica en la pantalla al usuario que pulse el botón Play/Stop.

```
{"idmsg":999,"button":" Pulse Play/Stop ","iduser":"2","idusersend":"1000-  
phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:27","type":"remote","value":"save"}
```

6. Indica en la pantalla al usuario que pulse el botón Volumen Arriba.

```
{"idmsg":1000,"button":" Pulse Volumen Arriba  
","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:30","type":"remote","value":"save"}
```

7. Indica en la pantalla al usuario que pulse el botón Volumen Abajo.

```
{"idmsg":1001,"button":" Pulse Volumen Abajo  
","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-  
16:24:33","type":"remote","value":"save"}
```

8. Indica en la pantalla al usuario que pulse cualquier botón para terminar. Éste paso esta implementado para comprobar que el programa ha salido correctamente del bucle de grabado.

```
{"idmsg":1013,"button":"Pulse cualquier botón hasta que se  
avise","iduser":"2","idusersend":"1000-phone-  
156","typeuser":"raspi","time":"14/01/2017-16:25:04","type":"remote","value":"save"}
```

9. Indica en la pantalla al usuario que ya ha terminado y puede empezar a disfrutar del nuevo mando a distancia.

```
{"idmsg":1014,"button":"Hemos terminado!\n\nPuede empezar a disfrutar de su nuevo mando a distancia","iduser":"2","idusersend":"1000-phone-156","typeuser":"raspi","time":"14/01/2017-16:25:15","type":"remote","value":"save"}
```

10. Tras esto el Raspberry Pi envía al servidor un mensaje indicando que ha terminado un proceso de grabación, éste con los datos proporcionados por el Raspberry Pi guarda dicho mando en su base de datos para poder ser consumido posteriormente por todos los integrantes del hogar. El formato del mensaje enviado es

```
{'typedevice': '4', 'idmsg': 1015, 'iduser': '2', 'typeuser': 'raspi', 'nameremote': 'Nombre mando música', 'address': 'IRMUSIC2', 'type': 'remote', 'value': 'save', 'idusersend': '0', 'time': '14/01/2017-16:25:15 '}
```

### 7.6.6 DESCRIPCIÓN MODULO TRADUCTOR DE VOZ A TEXTO

Éste módulo no está implementado en la aplicación, está incluido en el lado del Raspberry Pi. Para ser utilizado no hay que hacerlo desde la aplicación, si no desde el propio hogar.

Para poder utilizar éste módulo por los usuarios del hogar hay que insertar un micrófono y un convertidor USB a Jack 3.5mm.

La limitación de éste módulo es que el usuario sólo puede modificar los dispositivos como luces, electrodomésticos, persianas, aire acondicionado y calefacción.

Para utilizarlo el usuario tiene que decir una palabra clave, la cual se ha establecido como la palabra casa. Tras esta clave el usuario debe nombrar el nombre del dispositivo que desea encender o pagar con las mismas palabras que aparece en la aplicación.

Cuando esto sucede el Raspberry Pi enciende o apaga el dispositivo y envía al servidor un mensaje de tipo “change”, el formato del mensaje es el siguiente:

```
{'time': '17/01/2017-10:49:37', 'typeuser': 'raspi', 'address': '1/1/2', 'idusersend': '0', 'state': '1', 'type': 'change', 'userchange': 'Micrófono', 'value': 'device', 'iduser': '2'}
```

- Type: indica que es un mensaje de tipo change, lo que indica que ha habido un cambio en el hogar.
- Address: es la dirección KNX que ha cambiado de estado.
- State: nuevo valor del dispositivo.
- Value: indica que el cambio producido ha sido en un dispositivo.
- Userchange: indica que se ha cambiado el estado del dispositivo a través del micrófono.

Tras esto el servidor enviara a los dispositivos conectados un mensaje de change para mostrar en pantalla el cambio realizado.

## 7.7 IMPLENTACIÓN

### 7.7.1 DISEÑO

#### 7.7.1.1 SERVIDOR

Éste es el dispositivo que hace posible toda la comunicación entre clientes y hogares (Raspberry Pi). Sus clases están bien definidas y realizadas para efectuar un rápido reenvío de mensajes entre los participantes de la comunicación.

Consta de distintas clases implementadas, las cuales tienen una finalidad definida. Un diagrama de flujo que puede aclarar cómo responde el servidor ante una nueva conexión es el siguiente.

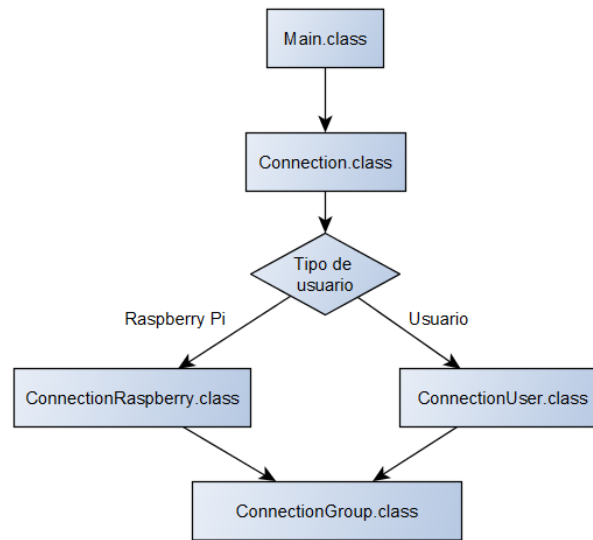


Figura 22. Flujo de clases de conexión en servidor

Para definir la finalidad de cada una se especifica cada una de estas.

|             |   |
|-------------|---|
| CLASE       | <b>Main.</b>  |
| DESCRIPCIÓN | Esta es la clase de inicio del servidor. En ella se establece el proceso de escucha de nuevas conexiones. Para garantizar una rápida disponibilidad ante recibir conexiones simultaneas ha sido desarrollada de tal forma que tras recibir una nueva conexión esta pasa a la siguiente clase en una hebra distinta y así volviendo a la escucha de nuevas conexiones. Su única función es la de recibir conexiones. |

|             |  |
|-------------|--|
| CLASE       | <b>Connection.</b>   |
| DESCRIPCIÓN | Esta clase es la segunda en el árbol anteriormente expuesto. Se encarga de la autenticación del servidor con él envió del certificado digital y a la vez gestionar el tipo de usuario que está accediendo al servicio para poder realizar los métodos convenientes en función del tipo de conexión. Esta es la encargada de recibir el mensaje de autenticación del usuario y descifrarlo. |

|             |   |
|-------------|---|
| CLASE       | <b>ConnectionRaspberry.</b>   |
| DESCRIPCIÓN | Esta clase se encarga de comprobar los datos enviados por el Raspberry Pi en el momento de la autenticación. A su vez es quien decide qué tipo de Login está realizando el Raspberry Pi para proceder al intercambio de claves de cifrado simétrico o sólo autenticar al dispositivo. En ella se produce el envío de los dispositivos KNX del hogar para que el dispositivo posea una lista actualizada de estos. |

|             |  |
|-------------|--|
| CLASE       | <b>ConnectionUser.</b>   |
| DESCRIPCIÓN | Esta clase se encarga de comprobar los datos enviados por el usuario en el momento de la autenticación. A su vez es quien decide qué tipo de Login está realizando el usuario para proceder al intercambio de claves de cifrado simétrico o sólo autenticar al dispositivo. Es la encargada de gestionar las sesiones de usuario, comprobando si el dispositivo asociado al usuario posee de una sesión activa o por lo contrario crearla. |

|             |   |
|-------------|---|
| CLASE       | <b>ConnectionGroup.</b>   |
| DESCRIPCIÓN | Esta es la última clase del diagrama de flujo del diagrama de flujo de la Figura 76. Autenticación clientes. Cuando la conexión de usuario llega a esta significa que el usuario o Raspberry ha sido autenticado con éxito y por tanto la transmisión de datos entre usuario/hogar puede comenzar. Es la encargada de realizar el filtrado de mensajes para realizar un comportamiento u otro en función de su tipo. En esta es donde se realiza el reenvío de datos. Es quien actualiza el temporizador del Raspberry Pi ante datos entrantes, si no, es quien cierra la conexión y envía los avisos a los usuarios. |

También existen otras clases las cuales no se incluyen en la traza del programa ya que son clases de apoyo o estructuras definidas para lograr el fin de dicho dispositivo. Otras clases que no aparecen en la Figura 22. Flujo de clases de conexión en servidor, pero no por ello no son importantes son las siguientes.

|             |   |
|-------------|---|
| CLASE       | <b>User.</b>  |
| DESCRIPCIÓN | Esta clase es la que mantiene la estructura de datos referente a los usuarios conectados. |

|             |   |
|-------------|---|
| CLASE       | <b>Raspberry.</b>   |
| DESCRIPCIÓN | Esta clase es la que mantiene la estructura de datos referente a los Raspberry Pi conectados. |

|             |  |
|-------------|--|
| CLASE       | <b>Certificate.</b>  |
| DESCRIPCIÓN | La clase certificate es la encargada de tener disponible en todo momento el certificado digital SSL. |

|             |  |
|-------------|--|
| CLASE       | <b>RSA.</b>  |
| DESCRIPCIÓN | Clase encargada de la obtención de las claves publica/privada del servidor. En ella es donde se produce el descifrado de la autenticación de usuarios y Raspberry Pi. Esta también es la encargada de cifrar con la correspondiente clave pública de los usuarios el mensaje que contiene la clave de cifrado simétrico. |

|             |   |
|-------------|---|
| CLASE       | <b>Aes.</b>   |
| DESCRIPCIÓN | Clase encargada del cifrado simétrico de las comunicaciones. En ella se produce el cifrado y descifrado de los datos. Esta también es la encargada de realizar el cálculo de MAC de cada mensaje. En ella es donde se genera las claves de cifrado simétrico enviadas a los usuarios. |

|             |  |
|-------------|--|
| CLASE       | <b>FirestoreSend.</b>  |
| DESCRIPCIÓN | Su fin es realizar el envío de los mensajes enviados a través de la API de Firebase Cloud Messaging. |

|             |   |
|-------------|---|
| CLASE       | <b>BDOperations.</b>  |
| DESCRIPCIÓN | El fin de esta clase es contener todas las operaciones a utilizar con la base de datos. |

|             |  |
|-------------|--|
| CLASE       | <b>Ssocket.</b>  |
| DESCRIPCIÓN | Esta clase contiene toda la información que hace referencia a la conexión existente. En ella se produce el envío y recepción de datos. Posee instancias de objetos Socket, User, Raspberry, Sessions, Aes. |

|             |  |
|-------------|--|
| CLASE       | <b>Device.</b>   |
| DESCRIPCIÓN | Contiene la estructura de datos referente a los dispositivos de los hogares. |

|             |  |
|-------------|--|
| CLASE       | <b>Scene.</b>  |
| DESCRIPCIÓN | Contiene la estructura de datos referente a las sesiones de usuario. |

|             |   |
|-------------|---|
| CLASE       | <b>SignUpRequest.</b>   |
| DESCRIPCIÓN | Clase con todos los métodos necesarios ante un registro de nuevo usuario. |

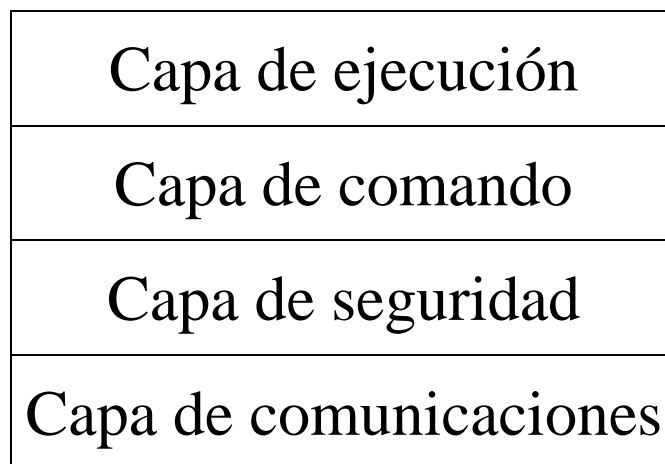
|             |   |
|-------------|---|
| CLASE       | <b>MapSocket.</b>   |
| DESCRIPCIÓN | Para realizar un rápido reenvío de mensajes han sido creados mapas concurrentes lo cuales contienen mapas de instancias de objetos Ssocket, esta clase es el contenedor de todos los mapas. Esto permite dentro de un mapa general agrupar mapas de la totalidad de hogares existentes. Para realizar el reenvío de datos de un usuario a su hogar se busca en el mapa general el mapa con clave el id del hogar obteniendo el mapa con la totalidad de los usuarios conectados asociados a dicho hogar y así dentro de éste mapa el socket en cuestión a enviar siendo posible dos distinciones: clave para la posición del mapa del socket del Raspberry Pi su id, clave para la posición del mapa de los usuario id_de_usuario-phone-id_de_sesión. |

**Librerías utilizadas:**

- commons-codec.jar. El paquete Apache Commons Codec contiene codificador simple y decodificadores para diversos formatos como base 64. Descargado en <https://commons.apache.org/proper/commons-codec/>.
- java-json.jar. Paquete para codificación y descodificación en formato JSON. Descargado en <https://mvnrepository.com/artifact/org.json/json/1.5-20090211>.
- javax.mail.jar. Paquete para el envío de correos electrónicos con Java. Descargado en <http://www.oracle.com/technetwork/java/javamail/index-138643.html>.
- mysql-connector-java-5.1.20.jar. Paquete conector con base de datos MySQL. Descargado en <https://dev.mysql.com/downloads/connector/j/>.

**7.7.1.2 RASPBERRY PI**

El software desarrollado en el Raspberry Pi ha sido creado por capas. Un esquema que puede verse su estructura es el siguiente.



*Figura 23. Modelo de pila Raspberry Pi*

Cada una de estas capas tiene una finalidad definida. Ha sido desarrollado mediante éste esquema para asegurar un comportamiento aislado entre cada una de ellas. Estas se comunican con sus capas inmediatamente superior o inferior consiguiendo, una estructura secuencial. Si se desea realizar cualquier modificación el desarrollador sólo tiene que modificar la capa que afecte a dicho cambio.

Éste software es lanzado con cada inicio del sistema operativo del Raspberry Pi mediante un demonio<sup>1</sup>, ha sido realizado para que ante algún fallo no controlado (siendo difícil que esto ocurra) el programa sea reiniciado. Si el software falla sólo necesitara de unos 3 segundos aproximadamente para reiniciarse.

Como ha sido mencionado en el apartado de análisis de soluciones se utilizan diversos softwares para alcanzar los objetivos.

- SpeechRecognition 3.6.0 (Zhang, 2017). Hace uso de la API Google Speech Recognition. Está escrito en Python y apoya compatibilidad para versiones de Python como son 2.6, 2.7 o 3.3 y superiores.
- Librerías EIBD (EIBD, 2005). Desarrolladas en C, proporcionan una conexión crucial para el funcionamiento del fin de éste proyecto.
- LIRC (LIRC, 1999). Es un paquete que permite decodificar y enviar señales de infrarrojos de muchos (pero no todos) los mandos a distancia de uso común.

El software SpeechRecognition ha sido integrado dentro de las propias clases del proyecto, dado su compatibilidad con el software desarrollado que, también está escrito en Python. El uso del demonio EIBD y LIRC se completa mediante comunicación por la consola de Linux.

Los softwares SpeechRecognition y EIBD son ejecutados en procesos distintos al programa, que comunicándose por medio de colas (*Queue*) permite una ejecución totalmente asíncrona consiguiendo una rápida ejecución de las ordenes de los usuarios.

Los procesos que corren por separado dentro del programa son los siguientes.

- Programa principal. Hebras que corren en el:
  1. Escucha del socket (Hebra principal del programa).
  2. Escucha el proceso de voz.
  3. Espera órdenes del proceso de escucha del bus KNX.
  4. Espera órdenes del proceso de ping.
- Escucha del bus KNX.
- Escucha de micrófonos.
- Proceso ping.

---

<sup>1</sup> Démonio: programa ejecutado en Linux cuando el sistema operativo es iniciado. El demonio creado ha sido llamado domoticKNX.service con el alias domotic service.

Han sido separados en procesos por la peculiaridad del lenguaje Python y el uso del Global interpreter Lock (GIL)<sup>2</sup>.

Por esta razón se ha utilizado el módulo de Python multiprocessing para la ejecución de estos. Todos, ante una muerte del proceso padre, el cual es el programa principal serán terminados y se iniciara todo nuevamente.

La misión de las capas del sistema (Figura 23. Modelo de pila Raspberry Pi) es la siguiente.

|                        |   |
|------------------------|---|
| Capa de ejecución      | Ejecuta todos los metodos,<br>Escucha de microfonos,<br>Monitor de Bus KNX.     |
| Capa de comando        | Organiza la información entre capas   |
| Capa de seguridad      | Comprobar certificadó SSL,<br>Cifrado asimétrito RSA,<br>Cifrado simétrico AES. |
| Capa de comunicaciones | Gestión de conexión,<br>Ping.   |

Figura 24. Funciones de pila Raspberry Pi

#### COMMUNICATIONS LAYER

Esta capa se encarga de realizar la conexión con el servidor y de mantener esta estable. En esta capa es donde corre el hilo principal, si éste muere, todo será cerrado y reiniciado. Consta de un proceso que realiza el envío un mensaje de “ping” con el servidor para asegurar la conexión. Se ha comprobado que ante caídas de conectividad en el Raspberry Pi por desconexión del cable ethernet o pérdidas de conexión por parte del router del hogar tanto el servidor como el Raspberry Pi no detectan esta desconexión. Éste problema es solventado realizando éste mensaje de sonda. Si el socket existente en el Raspberry Pi no recibe datos en un minuto la conexión será cerrará y comenzará con la reconexión.

Ante la recepción de datos en esta capa se lanzará una hebra ejecutando los métodos de la capa inmediatamente superior, subiendo capa a capa hasta llegar a la capa Run Layer la cual, realizará un comportamiento u otro en función del tipo de mensaje recibido.

---

<sup>2</sup> El GIL es el mecanismo utilizado en Python para impedir que múltiples hebras modifiquen los objetos a la vez en una aplicación multi hilo. Al igual que en otros lenguajes, si queremos que sólo un hilo de ejecución haga cambios en los datos de la sección crítica, debemos hacer uso de la clase Lock que nos permite adquirir una sección crítica. El GIL es un bloqueo a nivel de intérprete. Este bloqueo previene la ejecución de múltiples hilos a la vez en un mismo intérprete de Python. Cada hilo debe esperar a que el GIL sea liberado por otro hilo.

Las clases desarrolladas para esta capa son las siguientes:

|             |   |
|-------------|---|
| CLASE       | <b>Ssockeckt.</b>   |
| DESCRIPCIÓN | Encargada de realizar la conexión con el servidor, realizar la autenticación del Raspberry Pi y gestionar dicha conexión ante posibles errores. Realiza mensajes de ping cada 50 segundos al servidor para garantizar el canal. |

### *CRYPTO LAYER*

El fin de esta es poder asegurar la privacidad, integridad de los datos y autenticación. Sus funciones son las siguientes:

- Autenticar al servidor.
- Realizar el cifrado asimétrico para la autenticación de los usuarios, Raspberry Pi y el intercambio de claves de cifrado simétrico.
- Realizar el cifrado simétrico de las comunicaciones durante el intercambio de datos.
- Comprobar la integridad de los datos.

Las funciones de esta capa anteriormente mencionadas se realizan mediante:

- Autenticación del servidor mediante un certificado digital. Éste enviado en el momento de la conexión.
- Autenticación de usuario e intercambio de clave simétrica mediante RSA con una clave de 2048 bits.
- Cifrado de las comunicaciones realizado mediante AES con un tamaño de clave de 256 bits y cifrador de bloque en modo CBC.
- Comprobar integridad de los datos realizado mediante el cálculo de MAC del mensaje.

Las clases desarrolladas para esta capa son las siguientes:

|             |   |
|-------------|---|
| CLASE       | <b>CryptLayer.</b>  |
| DESCRIPCIÓN | Esta clase es que conecta la capa Communications Layer y Command Layer. Los mensajes entrantes y salientes son enviados a esta clase y esta se ocupa de realizar el cifrado y descifrado. |

|             |  |
|-------------|--|
| CLASE       | <b>AESCypher.</b>  |
| DESCRIPCIÓN | Se encarga de realizar el cifrado y descifrado de los datos. Utiliza una clave de 256 bits y el modo de operación utilizado es CBC con lo que está también debe incluir en el mensaje el vector de inicialización. También inserta el MAC. |

|             |  |
|-------------|--|
| CLASE       | <b>RSACypher.</b>  |
| DESCRIPCIÓN | Es la encargada de generar la clave RSA de 2048 bits del Raspberry Pi en la primera conexión con el servidor y guardarla en un fichero (.pem). También se encarga de cifrar y descifrar los datos pertenecientes al inicio de sesión con la clave pública del servidor. Valida el certificado digital enviado por el servidor. |

### COMMAND LAYER

Esta capa es la encargada de preparar la información que se recibe de la capa inferior para pasarla a capa superior de ejecución. Esta capa ha sido realizada para organizar los datos entrantes al Raspberry Pi. Examina quien ha enviado el mensaje y tipo de mensaje. Las comprobaciones anteriores son realizadas para clasificar los datos entrantes y ejecutar el método adecuado de la capa superior.

Las clases desarrolladas para esta capa son las siguientes:

|             |  |
|-------------|--|
| CLASE       | <b>Command.</b>  |
| DESCRIPCIÓN | Clase que conecta la capa de cifrado con su capa superior (RunLayer). Se encarga de analizar el tipo de mensaje entrante y ejecutar el método oportuno de la clase Method. Para su capa superior es un conector para poder enviar los datos. |

### RUN LAYER

Esta es la capa que interpreta todos los datos entrantes y ejecuta los métodos necesarios para funcionamiento del sistema en el lado del Raspberry Pi. Gracias a esta el programa es fácilmente modificable, ya que sólo tendremos que cambiar el código en esta capa. Sus funciones desarrolladas son las siguientes.

- Escucha el proceso de monitorización del Bus KNX. Si se recibe un evento en el bus, esta capa interactuara con la capa inferior para enviar al servidor los cambios producidos.
- Escucha el proceso de voz. El proceso de voz sólo escucha y filtra los resultados de la transformación de voz a texto en busca de una secuencia de palabras para la ejecución de un evento. Cuando esto ocurre éste proceso interactúa con la capa run para indicarlo, por consecuencia, enviará al bus el telegrama pertinente y el mensaje al servidor indicando el cambio.
- Tras recibir un cambio por parte del usuario se enviará al bus el telegrama pertinente y responderá con la finalización de ejecución de éste al usuario.

Las clases desarrolladas para esta capa son las siguientes:

|             |   |
|-------------|---|
| CLASE       | <b>Method.</b>  |
| DESCRIPCIÓN | Clase en la que desarrollan las acciones según el tipo de mensaje recibido o realizar acciones por eventos de voz o por entrada de datos del bus KNX. |

|              |   |
|--------------|---|
| CLASE        | <b>ElementRaspi.</b>  |
| DESCRIPCIÓN  | Realiza las acciones asociadas a los elementos ajenos a KNX que están conectados al Raspberry Pi como los elementos infrarrojos, cámaras y obtener la temperatura |
| DEPENDENCIAS | <b>Clase que hace uso de la biblioteca LIRC.</b>  |

|             |   |
|-------------|---|
| CLASE       | <b>Microphone. Clase que hace uso de la biblioteca SpeechRecognition.</b>   |
| DESCRIPCIÓN | Esta clase es la encargada de todo lo relacionado con los micrófonos, como buscar el número de micrófonos conectados y escuchar las peticiones de usuario. Si hay una coincidencia entre las peticiones de usuario esta informara a la clase Method y se realizara la acción. |

|             |   |
|-------------|---|
| CLASE       | <b>MonitoringBus. Clase que hace uso de la biblioteca EIBD.</b>   |
| DESCRIPCIÓN | Su función es escuchar en todo momento el bus KNX y ante eventos informar a la clase Method. También buscar la dirección IP del router KNX para realizar la conexión. |

|             |   |
|-------------|---|
| CLASE       | <b>SendTelegram. Clase que hace uso de la biblioteca EIBD.</b>  |
| DESCRIPCIÓN | Esta clase tiene como fin enviar al bus KNX los distintos telegramas cuando se reciba en la clase Method una orden. |

Otras clases no especificadas anteriormente son las siguientes.

|             |   |
|-------------|---|
| CLASE       | <b>Start.</b>   |
| DESCRIPCIÓN | Clase main del proyecto. Esta es la clase que es ejecutada por el demonio Linux, se encarga de iniciar el programa e inicializar todos los objetos. |

|             |   |
|-------------|---|
| CLASE       | <b>InputMessage.</b>  |
| DESCRIPCIÓN | Clase encargada de generar los métodos getter para extraer los distintos campos de los datos recibidos en formato JSON. |

|             |   |
|-------------|---|
| CLASE       | <b>OutputMessage.</b>   |
| DESCRIPCIÓN | Clase encargada de encapsular los mensajes salientes en formato JSON. |

|             |   |
|-------------|---|
| CLASE       | <b>ReadFileConfig.</b>  |
| DESCRIPCIÓN | Clase encargada de extraer los datos del fichero de configuración en el que son almacenados los datos de Raspberry Pi y del servidor. |

|             |  |
|-------------|--|
| CLASE       | <b>Connection.</b>                               |
| DESCRIPCIÓN | Realiza una conexión con la base de datos local. |

|             |  |
|-------------|--|
| CLASE       | <b>BD.</b>   |
| DESCRIPCIÓN | Clase en el que almacena las operaciones para la extracción de datos, insertar o actualizar. |

### 7.7.1.3 DISPOSITIVO ANDROID

Ha sido desarrollada una aplicación Android la cual contiene los distintos modelos necesarios, lógica y vistas. Un esquema que define dicha aplicación es el siguiente.

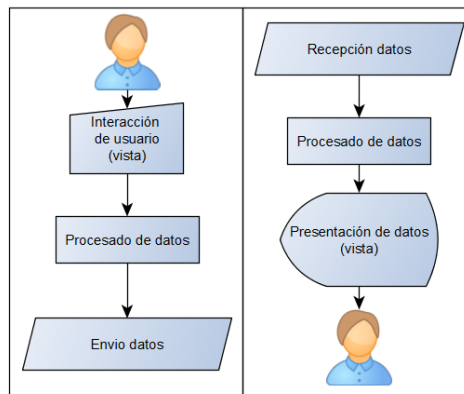


Figura 25. Flujo de datos Android

En la Figura 25. Flujo de datos Android se observa la estructura realizada para completar con éxito el fin de dicha aplicación.

Ante la entrada de datos, son procesados para administrar las vistas y estas informar de forma clara al usuario. Ante la interacción del usuario con la aplicación el proceso es inverso, la interacción es procesada y se envían los datos convenientes.

El diagrama de flujo de la Figura 26. Diagrama Activitiys Android puede verse el ciclo de Activitiys por los que se puede navegar en la aplicación.

Para conseguir el objetivo de dicha aplicación se exponen algunas de las clases principales desarrolladas.

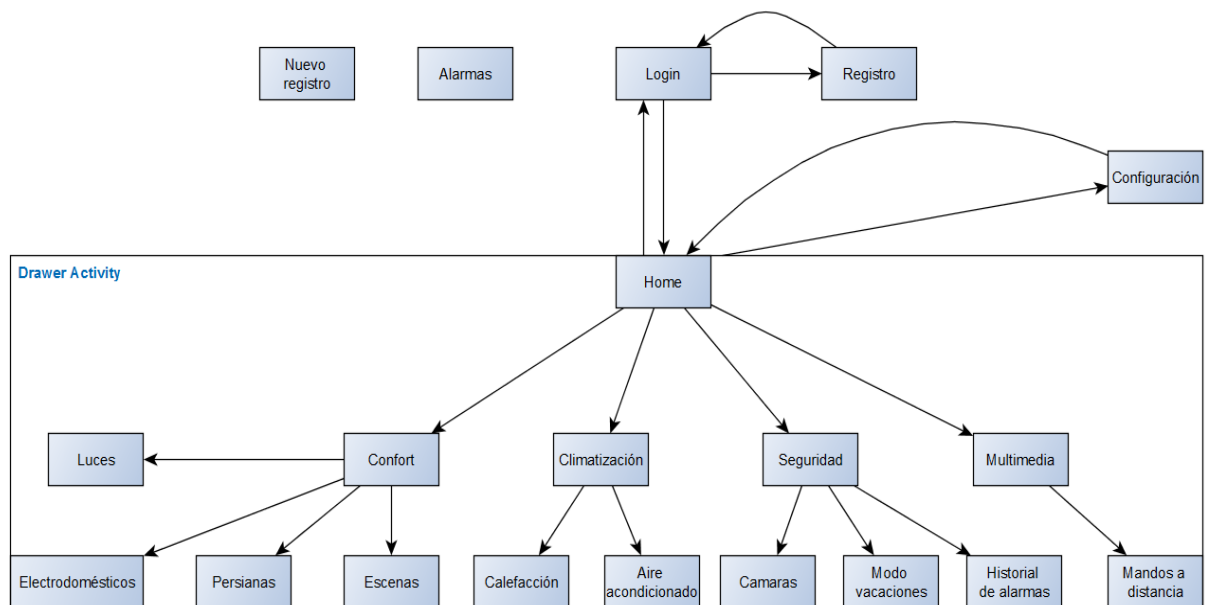


Figura 26. Diagrama Activitys Android

### MÓDULO DE INICIO EN LA APLICACIÓN.

|             |   |
|-------------|---|
| CLASE       | <b>ActivityLogin.</b>   |
| DESCRIPCIÓN | Activity encargado del control de acceso de la aplicación. En el se encuentra el formulario de Login y el acceso para el registro en el servicio.           |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"> <li>• FragmentLogin.</li> <li>• FragmentConnecting.</li> </ul> |

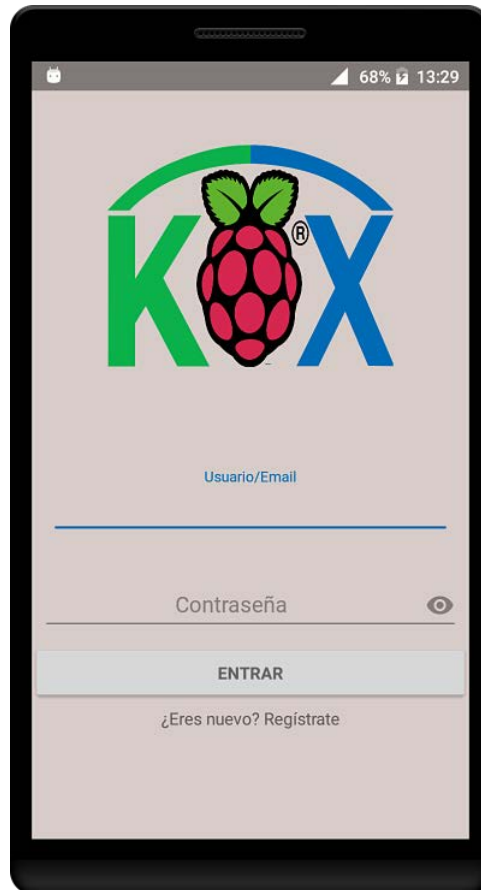


Figura 27. FragmentLogin



Figura 28. FragmentConnecting

### MÓDULO DE REGISTRO EN EL SERVICIO

|             |  |
|-------------|--|
| CLASE       | <b>ActivitySingUp.</b>   |
| DESCRIPCIÓN | Actividad encargada del registro en el servicio.   |
| FRAGMENTOS  | <p>Este activity está formado por los siguientes fragmentos:</p> <ul style="list-style-type: none"> <li>• SignUpUser. Fragmento encargado de introducir los nuevos datos de usuario.</li> <li>• SignUpIdHome. Fragmento que muestra el formulario de datos del hogar.</li> <li>• SignUpWait. Fragmento que muestra al usuario información tras el registro.</li> </ul> |

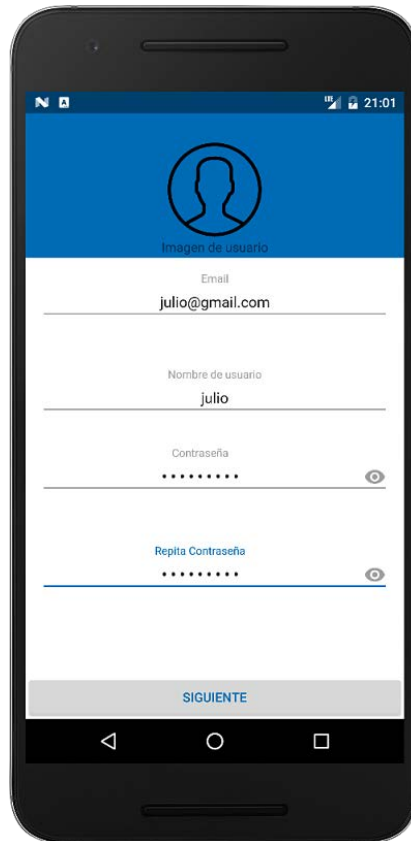


Figura 29. SignUpUser

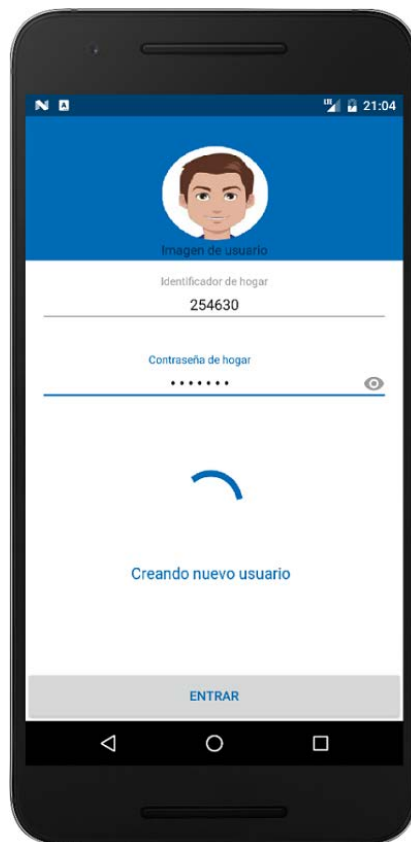


Figura 30. SignUpIdHome

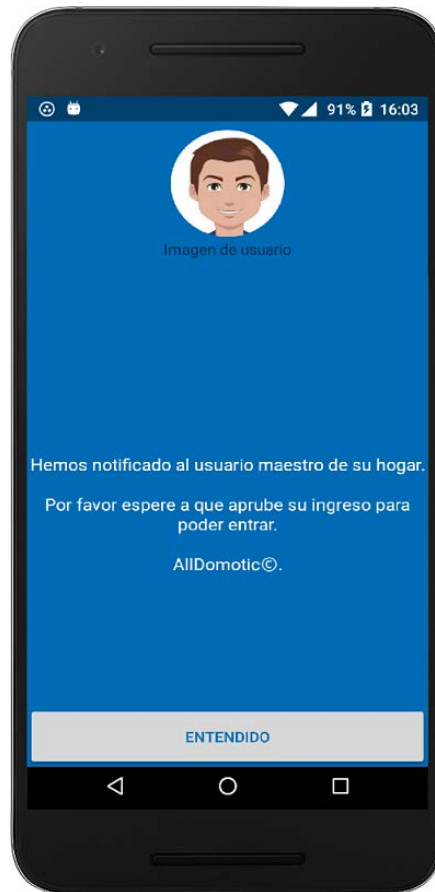


Figura 31. SignUpWait

### MÓDULO PRINCIPAL DE LA APLICACIÓN

|             |  |
|-------------|--|
| CLASE       | <b>ActivityHome.</b>   |
| DESCRIPCIÓN | Como su propio nombre indica éste es el activity home de la aplicación. En ella están disponible la entra a los distintos módulos y una vista reducida del estado del hogar. |

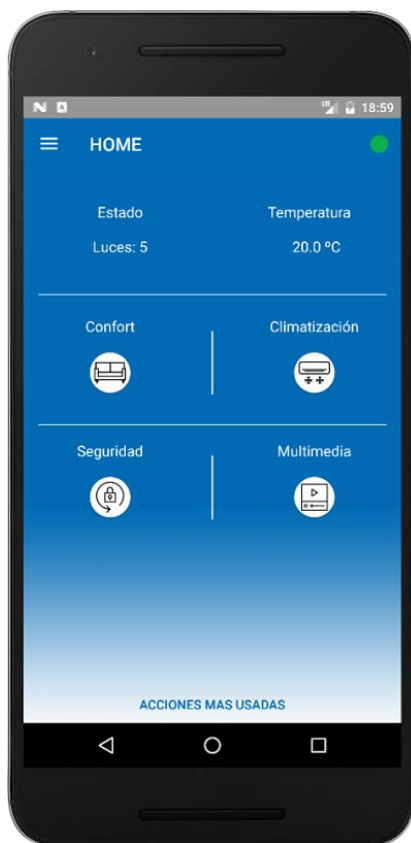


Figura 32. ActivityHome

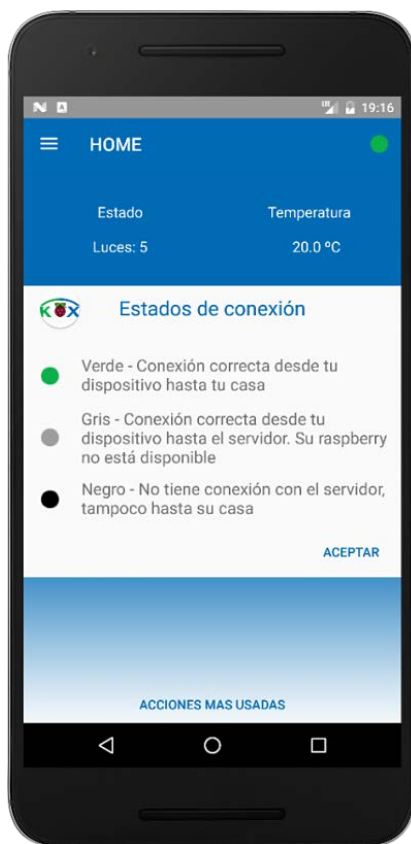


Figura 33. Estado conexión

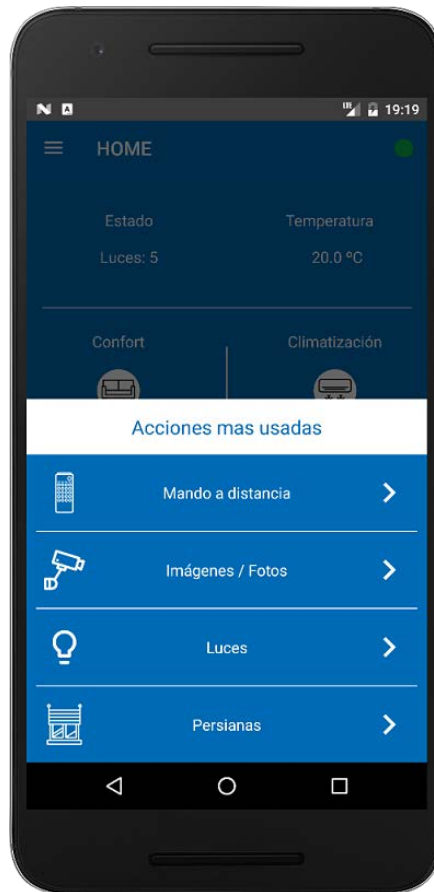


Figura 34. Menu rápido

### MÓDULO CONFORT

|             |   |
|-------------|---|
| CLASE       | <b>ActivityComfort.</b>   |
| DESCRIPCIÓN | Actividad en el que se muestran las distintas opciones del módulo de confort. |



Figura 35. ActivityComfort

|             |   |
|-------------|---|
| CLASE       | <b>ActivityLight.</b>   |
| DESCRIPCIÓN | Actividad en el que se encuentra las opciones disponibles para los dispositivos de tipo luces. Esta consta de un ViewPager, tablayout y dos recyclerview. |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"><li>• LightDevices.</li><li>• LightConfig.</li></ul>          |

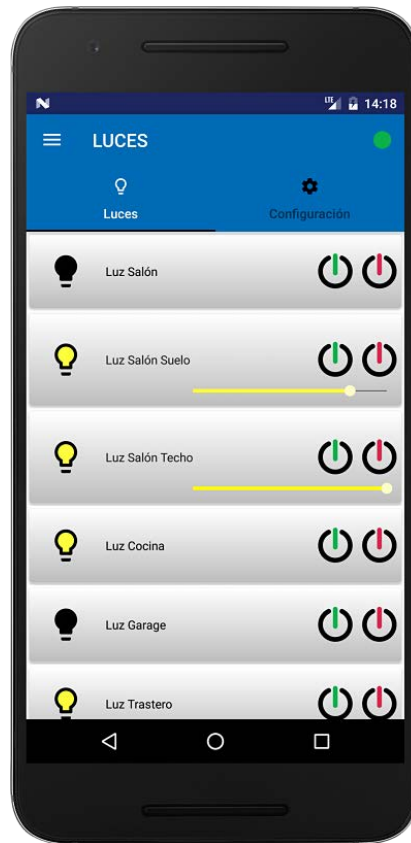


Figura 36. LightDevices

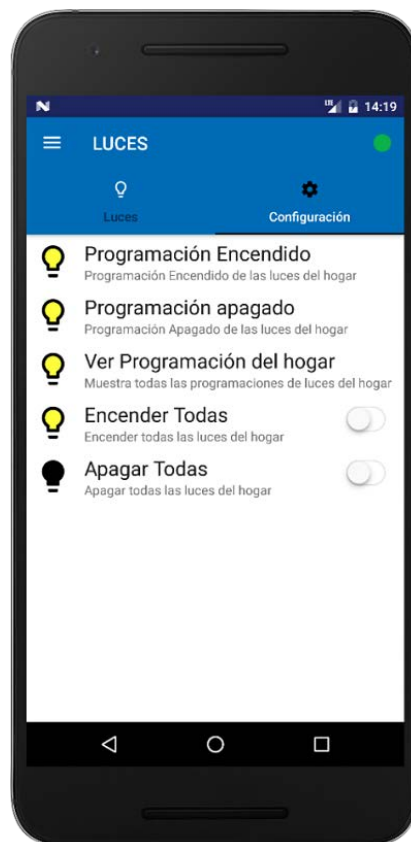


Figura 37. LightConfig

|             |  |
|-------------|--|
| CLASE       | <b>ActivityBlind.</b>  |
| DESCRIPCIÓN | Actividad en el que se encuentra las opciones disponibles para los dispositivos de tipo persiana. Esta consta de un Viewpager, tablayout y dos recyclerview.         |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"><li>• BlindDevices.</li><li>• BlindConfig.</li><li>• BlindRun.</li></ul> |

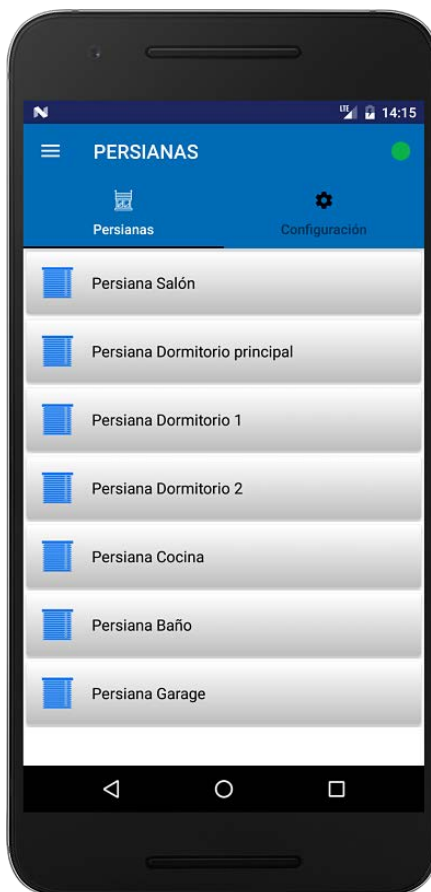


Figura 38. BlindDevices

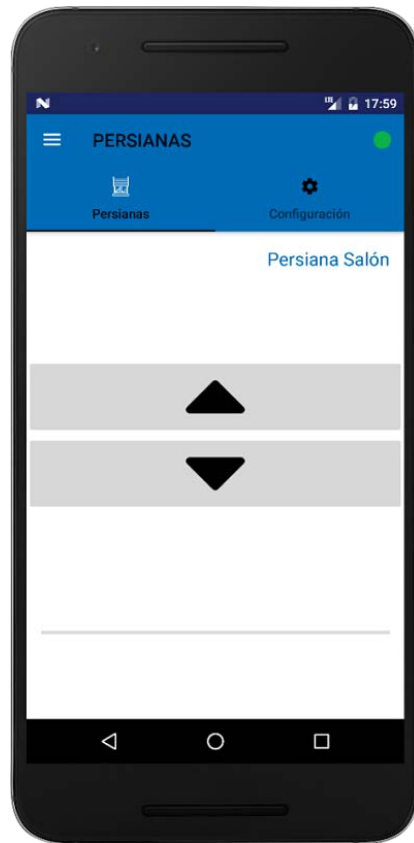


Figura 39. BlindConfig

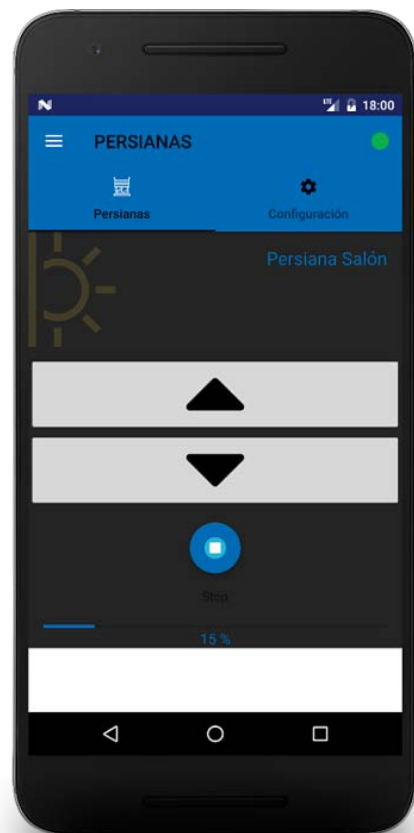


Figura 40. BlindRun

|             |  |
|-------------|--|
| CLASE       | <b>ActivityAppliance.</b>  |
| DESCRIPCIÓN | Actividad en el que se encuentra las opciones disponibles para los dispositivos de tipo electrodoméstico. Esta consta de un ViewPager, tablayout y dos recyclerview. |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"><li>• ApplianceDevices.</li><li>• ApplianceConfig.</li></ul>             |

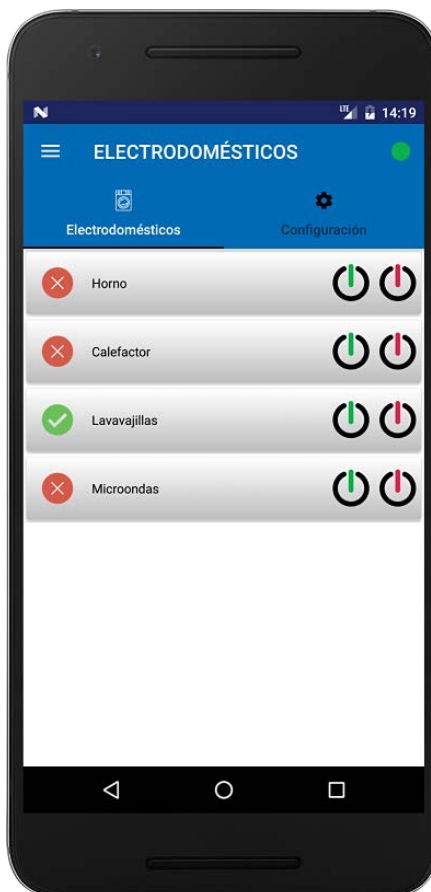


Figura 41. ApplianceDevices

|             |  |
|-------------|--|
| CLASE       | <b>ActivityScene.</b>  |
| DESCRIPCIÓN | Actividad en el que se encuentran las opciones de programación de escenas del total de los dispositivos del hogar. |

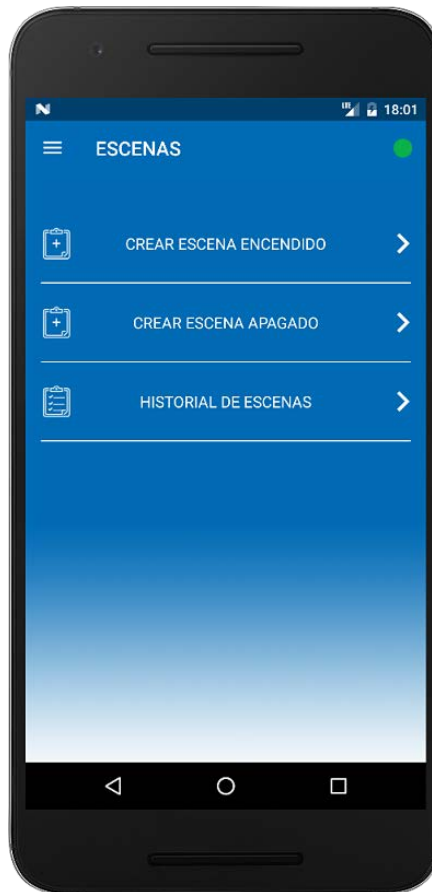


Figura 42. ActivityScene

### MÓDULO CLIMATIZACIÓN

|             |   |
|-------------|---|
| CLASE       | <b>ActivityAir.</b>   |
| DESCRIPCIÓN | Actividad en el que se muestran las distintas opciones del módulo de climatización. |



Figura 43. ActivityAir

|             |  |
|-------------|--|
| CLASE       | <b>ActivityHeating.</b>                            |
| DESCRIPCIÓN | Actividad de gestión de calefacción KNX del hogar. |

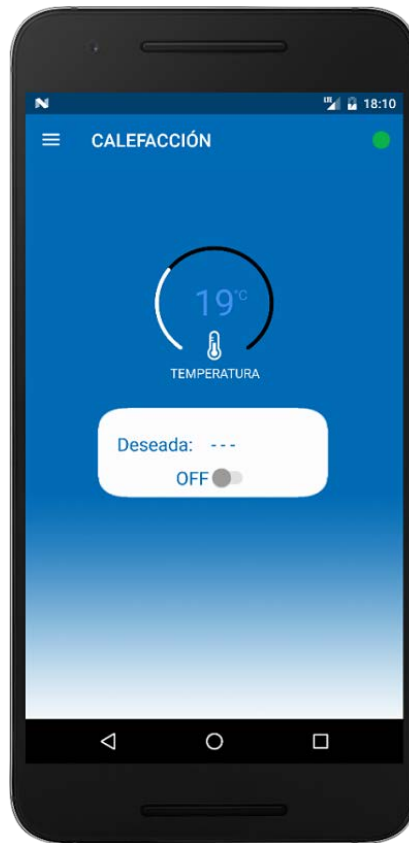


Figura 44. ActivityHeating



Figura 45. ActivityHeating selección



Figura 46. ActivityHeating activo

|             |   |
|-------------|---|
| CLASE       | <b>AirConditioner.</b>                                    |
| DESCRIPCIÓN | Actividad de gestión de aire acondicionado KNX del hogar. |

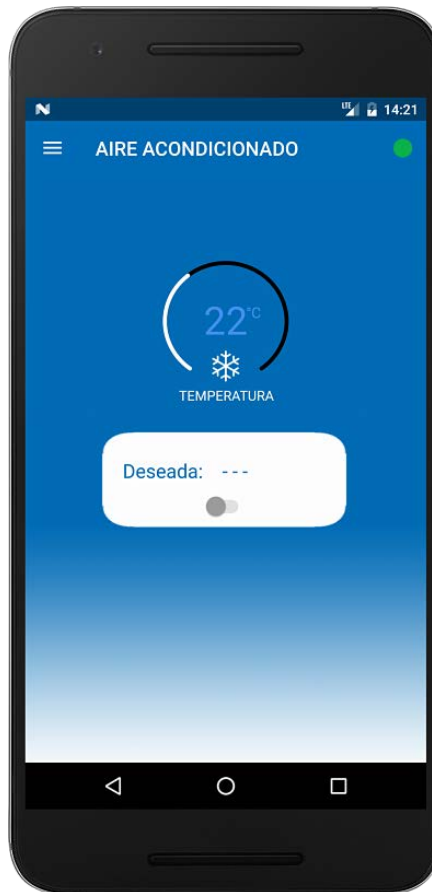


Figura 47. AirConditioner

### MÓDULO SEGURIDAD

|             |   |
|-------------|---|
| CLASE       | <b>ActivitySecure.</b>  |
| DESCRIPCIÓN | Actividad en el que se muestran las distintas opciones del módulo de seguridad. |



Figura 48. ActivitySecure

|             |  |
|-------------|--|
| CLASE       | <b>ActivityImages.</b>   |
| DESCRIPCIÓN | Actividad en el que se encuentran las opciones de cámara y se visualizan las imágenes previamente descargadas. |

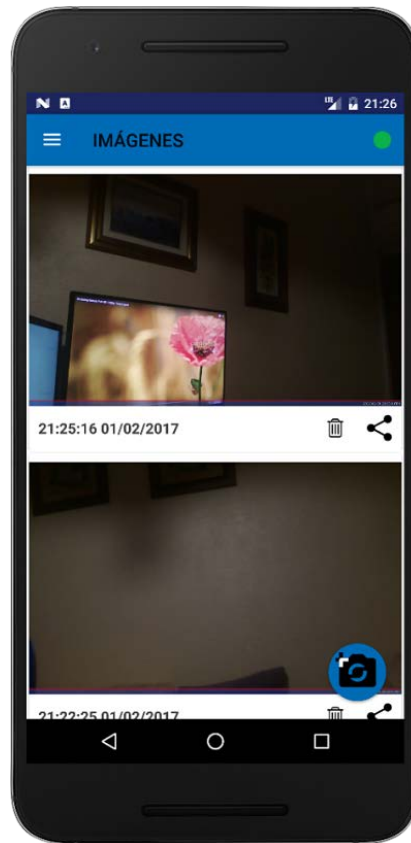


Figura 49. ActivityImages

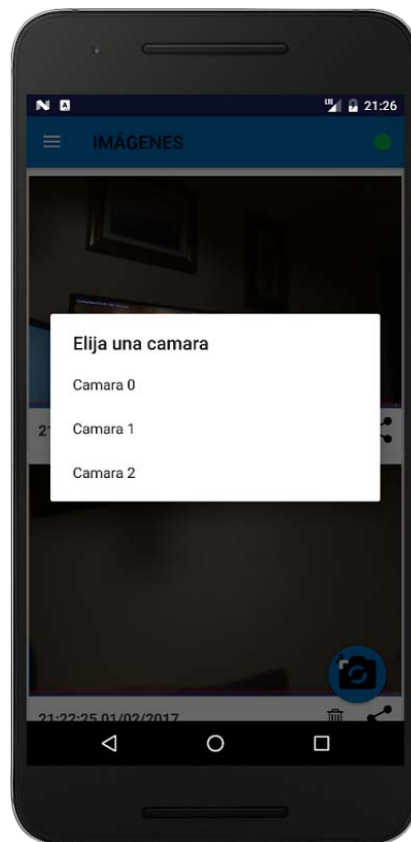


Figura 50. ActivityImages selección cámara

|             |  |
|-------------|--|
| DESCRIPCIÓN | Actividad en el que se encuentran las opciones de cámara y se visualizan las imágenes previamente descargadas. |
|-------------|--|



Figura 51. ActivityHolidays

|             |  |
|-------------|--|
| CLASE       | <b>HistoryAlarm.</b>   |
| DESCRIPCIÓN | Actividad en el que se encuentran las opciones de cámara y se visualizan las imágenes previamente descargadas. |

### MÓDULO MULTIMEDIA

|             |   |
|-------------|---|
| CLASE       | <b>ActivityMultimedia.</b>  |
| DESCRIPCIÓN | Actividad en el que se muestran la opcion del módulo de multimedia. |

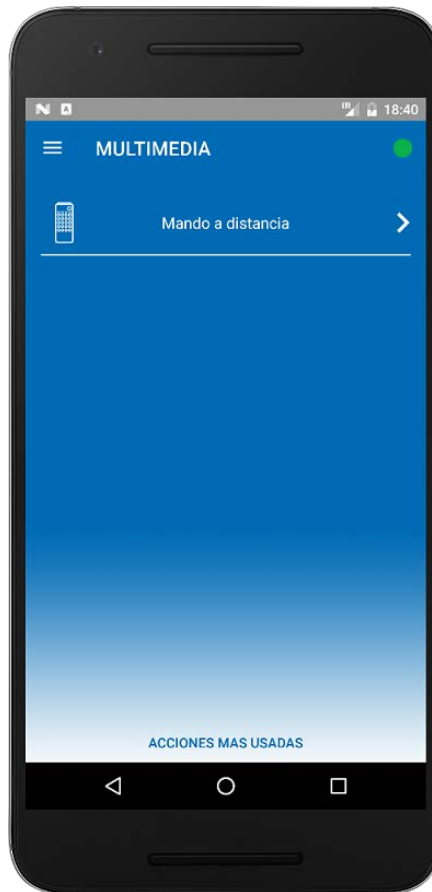


Figura 52. ActivityMultimedia

|             |   |
|-------------|---|
| CLASE       | <b>ActivityIR.</b>  |
| DESCRIPCIÓN | Actividad en el que están presentes las opciones de mando a distancia. Los  |
| FRAGMENTOS  | <p>Éste activity está formado por los siguientes fragmentos:</p> <ul style="list-style-type: none"> <li>• RemoteControl.</li> <li>• RemoteControlTV.</li> <li>• RemoteControlAIR.</li> <li>• RemoteControlMUSIC.</li> </ul> |

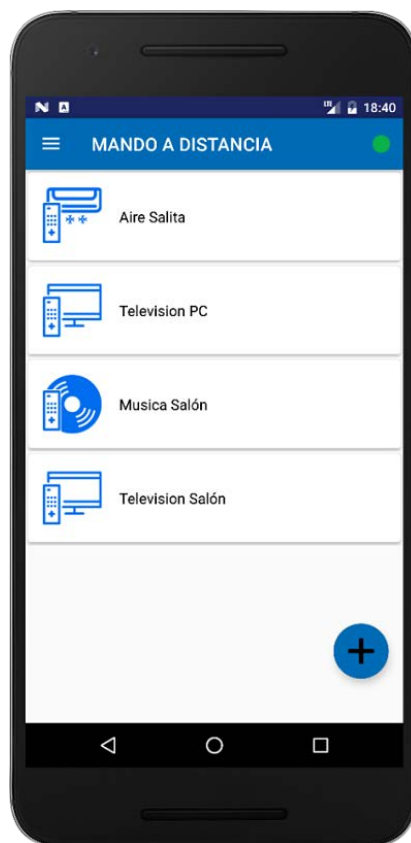


Figura 53. RemoteControl

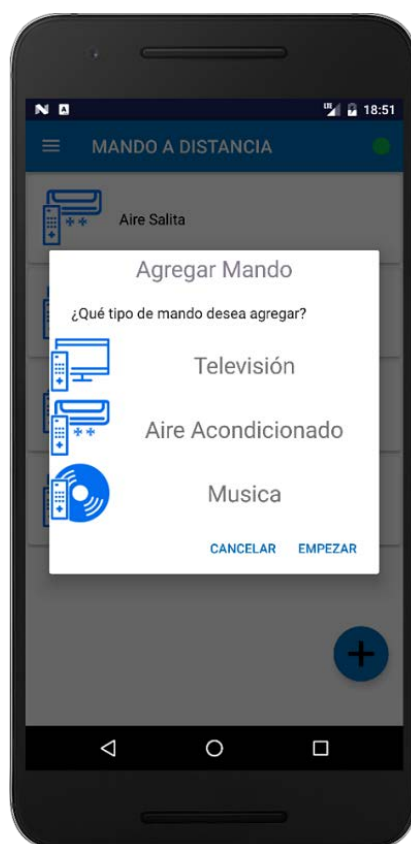


Figura 54. RemoteControl agregar mando



Figura 55. RemoteControlTV



Figura 56. RemoteControlAIR

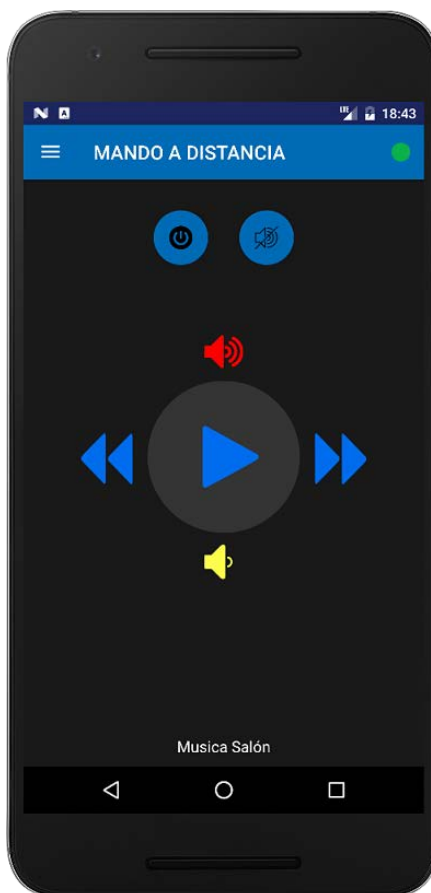


Figura 57. RemoteControlMUSIC

|             |  |
|-------------|--|
| CLASE       | <b>ActivitySaveRemote.</b>   |
| DESCRIPCIÓN | Actividad en el que se presenta la opción del grabado de un nuevo mando a distancia.   |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"><li>• RemoteSaveInstruction.</li><li>• RemoteSaveSteps.</li><li>• RemoteSaveFinal.</li></ul> |

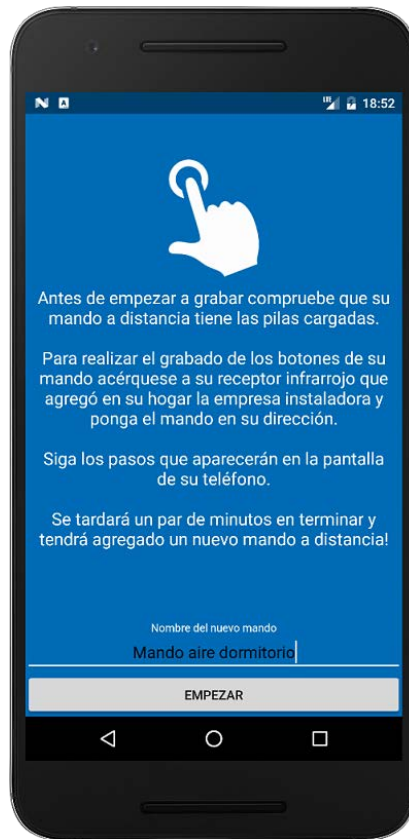


Figura 58. RemoteSaveInstruction

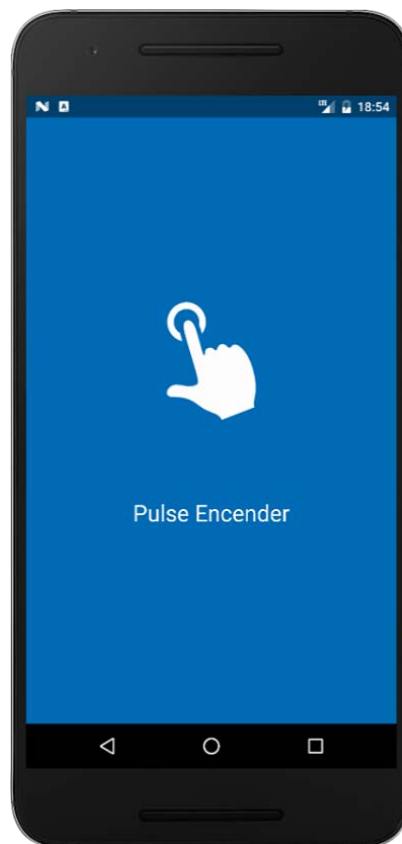


Figura 59. RemoteSaveSteps

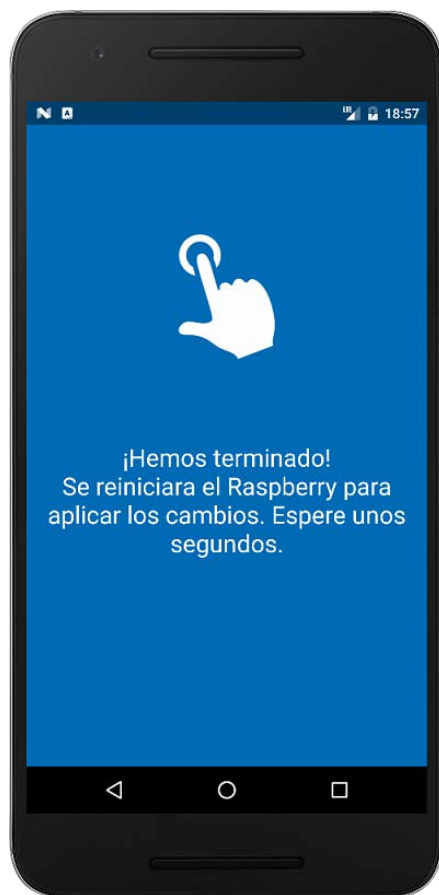


Figura 60. RemoteSaveFinal

### MÓDULO CONFIGURACION DE USUARIO

|             |  |
|-------------|--|
| CLASE       | <b>ActivitySettings.</b>   |
| DESCRIPCIÓN | Actividad en el que se presenta las opciones de configuracion de usuario.  |
| FRAGMENTOS  | Éste activity está formado por los siguientes fragmentos: <ul style="list-style-type: none"><li>• Settings. Éste fragmento extiende de PreferenceFragmentCompat.</li><li>• SettingsSessions.</li><li>• SettingsUserHome.</li></ul> |

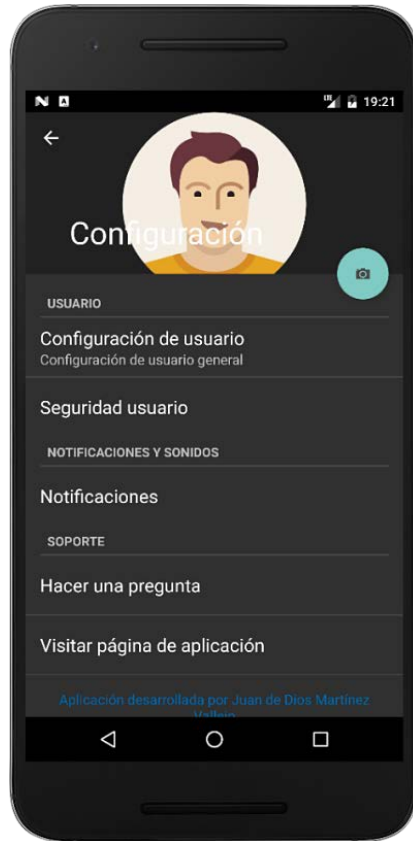


Figura 61. Settings

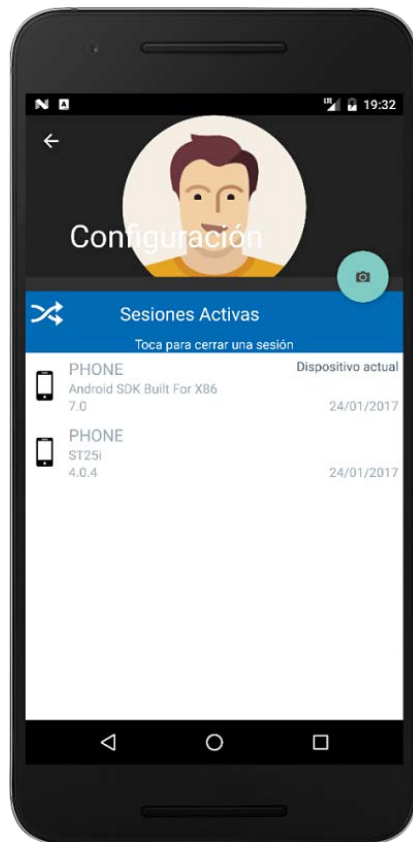


Figura 62. SettingsSessions

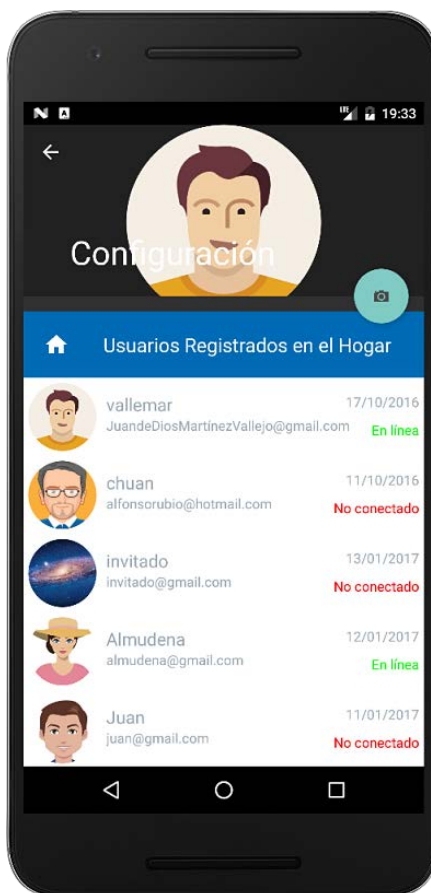


Figura 63. SettingsUserHome

### OTROS ACTIVITIES FUERA DE LOS MÓDULOS

| CLASE       | <b>DrawerActivity.</b>   |
|-------------|--|
| DESCRIPCIÓN | Activity encargado de la barra lateral de la aplicación. Éste se ha diseñado para realizar de tal forma que el mismo drawer es el mismo para toda la aplicación. Todos los Activities de la aplicación heredan de éste excepto el de Login y registro. Desde él se puede ir a cualquier módulo de la aplicación. |

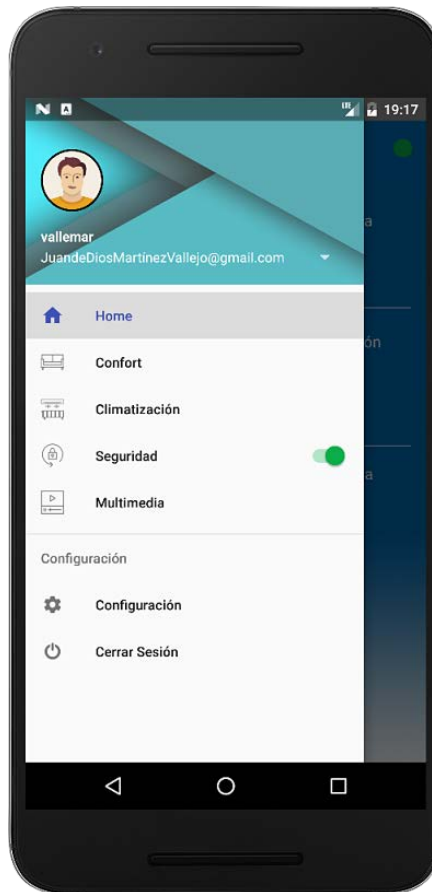


Figura 64. DrawerActivity menú lateral

|             |   |
|-------------|---|
| CLASE       | <b>ActivityAlarm.</b>   |
| DESCRIPCIÓN | Actividad que es mostrada ante una alarma en el hogar o ante un nuevo registro en el hogar (sólo para el usuario root del hogar). |

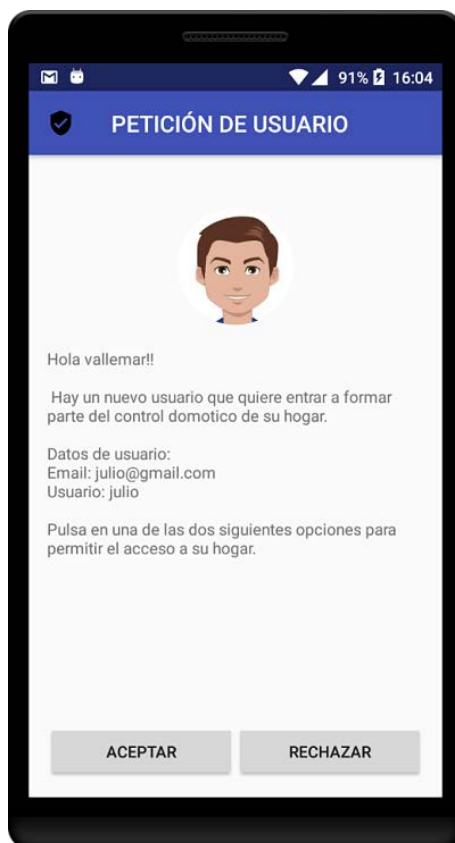


Figura 65. ActivityAlarm petición de usuario

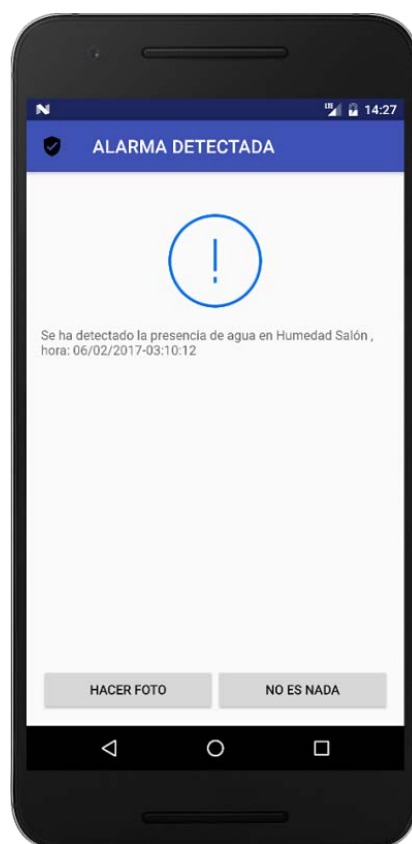


Figura 66. ActivityAlarm alarma

|             |  |
|-------------|--|
| CLASE       | <b>PinLock.</b>  |
| DESCRIPCIÓN | Actividad que, aunque también se encarga de la seguridad de usuario sólo es mostrada si el usuario solicita activar el pin de aplicación. Esta es el activity donde aparece el pin para desbloquear la aplicación. |

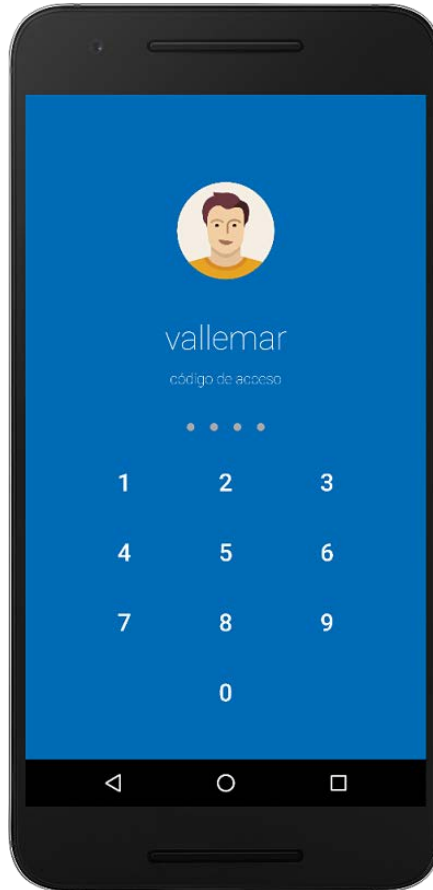


Figura 67. PinLock

|             |  |
|-------------|--|
| CLASE       | <b>SplashActivity.</b>   |
| DESCRIPCIÓN | Es la actividad principal, es decir, es la actividad lanzada al inicio de la aplicación. Esta es lanzada durante la carga de la aplicación. Su duración es de un segundo. Tras éste tiempo se lanzará el ActivityHome. |

### SERVICIOS DE APLICACIÓN

|             |  |
|-------------|--|
| CLASE       | <b>ServiceListenSocket.</b>  |
| DESCRIPCIÓN | Éste servicio es la toma de entrada de datos de toda la aplicación. Éste se mantiene a la escucha de la entrada de datos durante toda la conexión. Envía BroadcastReceiver tras la entrada de datos. |

|             |   |
|-------------|---|
| CLASE       | <b>FirebaseTokenRefresh.</b>  |
| DESCRIPCIÓN | Servicio que es ejecutado cuando el token de firebase del cliente es renovado. Tras ejecutarse se envía el nuevo token al servidor para actualizarlo. |

|             |   |
|-------------|---|
| CLASE       | <b>FirestoreMessaging.</b>  |
| DESCRIPCIÓN | Servicio al que llegan los datos del servidor al enviarlo a través de firebase. |

### *OTRAS CLASES DE APLICACIÓN*

|             |  |
|-------------|--|
| CLASE       | <b>DataUser.</b>   |
| DESCRIPCIÓN | Clase que mantiene el modelo de datos del usuario registrado en la aplicación. |

|             |   |
|-------------|---|
| CLASE       | <b>Connected.</b>   |
| DESCRIPCIÓN | Clase en la que se realiza la conexión y desconexión con el servidor. |

|             |  |
|-------------|--|
| CLASE       | <b>SocketTcp.</b>  |
| DESCRIPCIÓN | Objeto singleton. Clase contenedora del socket que es establecido como conexión con el servidor. |

|             |   |
|-------------|---|
| CLASE       | <b>GenerateMoreUsedActions.</b>                     |
| DESCRIPCIÓN | Clase generadora del layout de acciones más usadas. |

## 7.7.2 DESCRIPCIÓN DETALLADA DE LA BASE DE DATOS

En éste apartado se presenta una descripción detallada de las tablas de la base de datos, mostrando todos sus campos y sus tipos de datos.

### 7.7.2.1 DISEÑO DE BASE DE DATOS

La generación de la base de datos se realizará mediante el modelo relacional. La base de datos de la aplicación esta implementada mediante el SGBD MySQL. El diseño de las tablas con los datos se muestra a continuación. La relación entre tablas se realiza mediante claves foráneas. El modelo de la entidad relación puede verse en los planos, Figura 81. Modelo base de datos.

#### *TABLA HOME*

Esta tabla almacena datos y hace referencia a los hogares digitales registrados en el servicio. Guarda información de estos para una correcta identificación.

- id\_home: Almacena el id del hogar. Éste es un id único para cada hogar. Es la Clave primaria de la tabla.
- id\_raspi: hace referencia a cada Raspberry Pi. Actualmente éste campo es igual que el campo id\_home.
- Pass: Almacena la contraseña del Raspberry Pi.

- `pass_login_user`: en éste campo se almacena la contraseña que un usuario debe introducir para el registro en el hogar.
- `key_aes`: almacena su clave para el cifrado AES.

### *TABLA USER*

Esta tabla almacena los datos referentes a los usuarios o clientes, ofrece gran cantidad de campos para tener información detallada de cada cliente.

- `id_user`: Cada usuario tiene un identificador único, el cual, es la clave primaria de la tabla.
- `name_user`: Éste campo almacena el nombre de usuario de usuario.
- `pass`: almacena la contraseña del usuario.
- `email`: almacena el email del usuario.
- `id_home`: clave foránea de la tabla, identifica cada usuario con el id de su hogar.
- `root`: identifica si el usuario es el usuario root del hogar.
- `state_user`: campo que puede valer 0 si aún no ha sido aprobado por el usuario root o 1 si ha sido validado.
- `registration_day`: se almacena la fecha en la que se dio de alta en el sistema.
- `key_aes`: almacena su clave para el cifrado AES.
- `token_user`: en éste campo se almacena un token para validar el usuario root en el momento del registro en el hogar. Éste es el token que envía el usuario root si desea validar al usuario.
- `token_user_bad`: en éste campo se almacena un token para no validar el usuario root en el momento del registro en el hogar. Éste es el token que envía el usuario root si no desea validar al usuario.

### *TABLA SESSIONS*

Esta tabla almacena los datos asociados a las distintas sesiones que los usuarios pueden tener activas.

- `id_sessions`: identificador de la sesión. Éste campo es la clave primaria de la tabla.
- `id_device`: en éste campo se almacena el identificador del dispositivo que tiene asociada la sesión. Éste es un valor único.
- `model_device`: en éste campo se almacena el modelo del dispositivo que tiene activa la sesión.
- `type_device`: identifica el tipo de dispositivo. Éste campo actualmente sólo tiene el valor `phone`. Ha sido introducido para posibles versiones posteriores del servicio.
- `os_device`: almacena el sistema operativo del dispositivo.
- `date_device`: identifica el día que el usuario inicio la sesión.
- `id_user`: identifica al titular de la sesión.
- `session_state`: tiene 2 posibles valores: 1, la sesión está activa. 0, ante un campo `session_exit= 1` y `session_state = 1` significa que el usuario ha pedido que se

cierre la sesión, pero no ha sido confirmado la expulsión del dispositivo. Si éste valor es 0, el dispositivo ha confirmado la salida de la sesión.

- `session_exit`: tiene 2 posibles valores: 0, el usuario no ha pedido el cierre de sesión del dispositivo. 1, el usuario desde otro terminal ha solicitado el cierre de sesión.

#### *TABLA TID\_DEVICE\_ID\_FIREBASE*

Esta tabla identifica un dispositivo de una sesión con token del servicio firebase para mensajería en cualquier momento.

Esta ha sido realizada a parte de la tabla de sesiones por la posible ampliación del servicio proporcionado por éste proyecto para proporcionar un servicio web o aplicación de escritorio.

- `id_user`: asocia al usuario.
- `id_device`: asocia el token firebase con el id del dispositivo.
- `id_firebase`: almacena el token proporcionado por firebase.
- `id_session`: asocia el token firebase con una sesión.

#### *TABLA TYPE\_HW*

Esta tabla asocia cada componente hardware con un id número para servir un fácil manejo del servidor y la aplicación.

- `type_hw`: identificador único de un componente hardware.
- `description`: descripción del componente.

#### *7.7.2.2 TABLA ACTION\_HW*

Tabla que asocia una acción de un dispositivo con una cadena legible para el instalador.

- `id_action`: campo identificador del tipo de acción.
- `action_hw`: pequeña descripción del tipo de acción.
- `descripción`: descripción más extendida del tipo de acción.

#### *7.7.2.3 TABLA HOME\_HW*

Esta tabla identifica cada dispositivo asociado a un hogar. También establece como clave foránea para un identificar el tipo de dispositivo y acción que puede producir.

- `id_hw`: es el identificador del dispositivo. Forma parte de la clave primaria compuesta.
- `id_home`: idéntica el hogar al que está asociado el dispositivo. Es el segundo componente de la clave primaria.
- `type_hw`: indica el tipo de dispositivo. Es la clave foránea de esta tabla. Hace referencia a la tabla `type_hw`.

- hw\_is\_test: indica si el dispositivo está en pruebas o por lo contrario listo para usar por el usuario.
- description\_hw: establece una cadena de caracteres impuesta por el cliente para el reconocimiento del dispositivo.

#### 7.7.2.4 TABLA HOME\_HW\_ACTION

- id\_hw: es el identificador del dispositivo. Forma parte de la clave primaria compuesta.
- id\_home: idéntica el hogar al que está asociado el dispositivo. Es el segundo componente de la clave primaria.
- address\_hw: si se trata de un dispositivo KNX establece su dirección. Si es un dispositivo asociado a Raspberry Pi establece su pin del GPIO. Si se trata de un mando a distancia establece el id del mando establecido en el Raspberry Pi.
- action\_hw: indica el tipo de acción que puede realizarse con dicho dispositivo.

#### 7.7.2.5 TABLA HOME\_HW\_ROLE

Esta tabla relaciona dispositivos que engloban a otros.

- id\_home: idéntica el hogar al que está asociado el dispositivo. Es el segundo componente de la clave primaria.
- id\_father: id de un dispositivo que tiene engloba a otros.
- id\_children\_hw: id de un dispositivo que está dentro de un grupo mayor.

#### 7.7.2.6 TABLA STATE\_HW

Esta tabla almacena el estado de cada dispositivo del hogar. Esta tabla siempre ha de estar actualizada para que el usuario tenga un actualizado conocimiento de los estados del hogar.

- id\_hw: identificador del dispositivo.
- id\_home: identificador del hogar.
- state: Estado del dispositivo.

#### 7.7.2.7 TABLA SCENE

Tabla de la base de datos en la que se almacenan las escenas de cada hogar digital.

- id\_scene: clave primaria de la tabla, identifica cada escena.
- id\_home: identifica al hogar asociado a la escena. Es la clave foránea de la tabla.
- name\_scene: nombre de la escena impuesto por el usuario en el momento de la creación de esta.
- action\_disp: indica si es una escena de encendido o de apagado. 0 indica apagado, 1 indica encendido.
- type\_disp: indica el tipo de dispositivo que acciona la escena.
- hour: las escenas pueden estar programadas para una hora, en éste campo se almacena esto.

- `id_weekly`: campo que indica si la escena hay que repetirla cada semana. 0 indica no repetir, 1 indica si repetir.
- `is_activate`: indica si la escena esta activa o por acción del usuario esta desactivada. 0 indica desactivada, 1 indica activada.
- `autor`: campo en el que se almacena el autor que genero la escena.

#### 7.7.2.8 TABLA SCENE\_DISP

Tabla que relaciona cada escena con todos los dispositivos que se han de cambiar de estado.

- `Id_scene`: éste es el identificador de cada escena, es clave foránea, identifica el campo `id` de la tabla `scene`.
- `Id_disp`: identifica cada uno de los dispositivos asociados a una escena, es clave foránea del campo `id_hw` de la tabla `home_hw`.

#### 7.7.2.9 TABLA SCENE\_DAYS

Tabla que identifica cada uno de los días de la semana que hay que repetir las escenas.

- `Id_scene`: éste identificador de la escena asociada, es clave foránea, identifica el campo `id` de la tabla `scene`.
- `Day`: identifica los días de la semana en los que la escena hay que activar. Posibles valores de 1 a 7.

#### 7.7.2.10 TABLA ALARM

Tabla en la que son almacenadas las alarmas detectadas.

- `id_alarm`: id de alarma.
- `id_home`: id del hogar al que pertenece la alarma.
- `type_alarm`: tipo de alarma. Estas pueden ser el equivalente a los tipos de sensores.
- `date`: fecha en la que se produjo la alarma.
- `body`: mensaje que fue enviado al usuario.
- `is_read_all`: indica si la alarma ha sido leída por los integrantes del hogar.

#### 7.7.2.11 TABLAS IMAGES

Tabla en la que se relacionan las imágenes almacenadas con el parámetro que el usuario tendrá que enviar para poder acceder a ella.

- `id_image`: parámetro `get` que el usuario tendrá que enviar para acceder a ella.
- `name_img`: nombre de la imagen.
- `id_home`: id del hogar al que la imagen está asociada.

## 7.8 PRUEBAS

### INTRODUCCIÓN

En esta fase se realizará una evaluación del sistema para verificar que se satisfacen los requisitos especificados o para identificar las diferencias entre los resultados obtenidos y los esperados. Una prueba con éxito es aquella en la que se obtienen muchos defectos, y no todo lo contrario.

### PRUEBAS REALIZADAS

Estas son algunas de las pruebas realizadas a la aplicación con el objetivo de comprobar su correcto funcionamiento.

|                    |   |
|--------------------|---|
| <b>PRUEBA 1</b>    | <b>Conexión del Raspberry Pi.</b>   |
| DESCRIPCIÓN        | Comprobar durante 15 días la conexión del Raspberry Pi con el servidor.   |
| RESULTADO ESPERADO | Se espera que la conexión entre el Raspberry Pi y el servidor sea estable ante pérdidas de conexión a Internet o problemas de software. El Raspberry Pi debe mantener la conexión o realizar una nueva conexión en el menor tiempo posible. |
| RESULTADO OBTENIDO | Se cumple el resultado esperado.  |

|                    |  |
|--------------------|--|
| <b>PRUEBA 2</b>    | <b>Envío de mensajes.</b>  |
| DESCRIPCIÓN        | Comprobar mediante el envío de 1.000 mensajes que no hay errores en la comunicación.   |
| RESULTADO ESPERADO | Se espera que ante el envío al hogar de 10.000 mensajes de cambios de estado en los dispositivos KNX no se presente ningún error o en todo caso informar al usuario. |
| RESULTADO OBTENIDO | Se cumple el resultado esperado.   |

|                    |   |
|--------------------|---|
| <b>PRUEBA 2</b>    | <b>Conexión en Android.</b>   |
| DESCRIPCIÓN        | Comprobar la conexión del dispositivo Android con el servidor.  |
| RESULTADO ESPERADO | Se espera que ante conmutaciones en los dispositivos Android entre Wi-Fi (802.11) y LTE, así como problemas en la red la aplicación sea capaz de mantener la conexión o realizar una nueva conexión en el menor tiempo posible. |
| RESULTADO OBTENIDO | Se cumple el resultado esperado.  |

|                    |  |
|--------------------|--|
| <b>PRUEBA 3</b>    | <b>Informe de errores en la aplicación.</b>  |
| DESCRIPCIÓN        | Ante errores en el envío de mensajes al servidor por problemas de red.   |
| RESULTADO ESPERADO | Se espera que ante errores en la transmisión de paquetes con el servidor la aplicación pueda avisar al usuario del que ha habido un error. |
| RESULTADO OBTENIDO | Se cumple el resultado esperado.   |

|                    |   |
|--------------------|---|
| <b>PRUEBA 4</b>    | <b>Versiones de Android.</b>  |
| DESCRIPCIÓN        | Se comprueba que ante distintas versiones de Android la aplicación realice sin errores todos los módulos de esta.   |
| RESULTADO ESPERADO | Se espera que ante distintas versiones de Android la aplicación funcione sin errores y cumpliendo todos los módulos desarrollados.  |
| RESULTADO OBTENIDO | <p>La aplicación cumple con todos los módulos. El estado de conexión a través del socket se ha podido comprobar que versiones de Android más altas gestionan mejor las conexiones e informan al desarrollador de los problemas en la red. En la versión 6 Marshmallow el informe de errores es excelente.</p> <p>Esto no es un del todo verídico ya que implementar versiones de Android bajas indican que el dispositivo cuenta con prestaciones menores y por tanto el grado de computación es más bajo y hace que la aplicación se ejecute más lentamente.</p> |

|                    |   |
|--------------------|---|
| <b>PRUEBA 5</b>    | <b>Login de clientes</b>  |
| DESCRIPCIÓN        | Prueba en Login de usuarios.  |
| RESULTADO ESPERADO | Se espera que ante problemas de Login de usuario por datos erróneos o problemas de red la aplicación pueda informar al usuario. |
| RESULTADO OBTENIDO | Se consigue el resultado esperado.  |

|                    |  |
|--------------------|--|
| <b>PRUEBA 6</b>    | <b>Control de voz</b>  |
| DESCRIPCIÓN        | Comprobar el proceso de escucha en el Raspberry Pi.  |
| RESULTADO ESPERADO | Se espera que el proceso del Raspberry Pi funcione sin errores y correctamente durante 15 días.  |
| RESULTADO OBTENIDO | Se consigue el resultado esperado. Existe un retardo de traducción cuando hay un cambio en el ruido del ambiente. Cuando pasa de ambientes calmados a un ambiente ruidoso. |

|                    |   |
|--------------------|---|
| <b>PRUEBA 7</b>    | <b>Control de elementos infrarrojos</b>   |
| DESCRIPCIÓN        | En esta prueba se comprobará el control de elementos infrarrojos y el grabado de nuevos controles remotos.                            |
| RESULTADO ESPERADO | Se espera que el Raspberry Pi emita correctamente los pulsos infrarrojos y realice un grabado de un nuevo control remoto sin errores. |
| RESULTADO OBTENIDO | Se consigue el resultado esperado.  |

|                    |  |
|--------------------|--|
| <b>PRUEBA 8</b>    | <b>Múltiples Raspberry Pi conectadas</b>   |
| DESCRIPCIÓN        | En esta prueba se realizará la conexión de diversas Raspberry Pi simulando varios hogares.                               |
| RESULTADO ESPERADO | Se espera que el servidor realice las modificaciones en base de datos esperadas y gestione las conexiones correctamente. |
| RESULTADO OBTENIDO | Se consigue el resultado esperado.   |

|                    |  |
|--------------------|--|
| <b>PRUEBA 9</b>    | <b>Monitorización de dispositivos del hogar</b>  |
| DESCRIPCIÓN        | En esta prueba se pretende comprobar la monitorización por parte de usuario del estado de los distintos elementos que componen el hogar digital. |
| RESULTADO ESPERADO | Se espera que el usuario siempre disponga de la visualización del estado de los dispositivos del hogar actualizado.                              |
| RESULTADO OBTENIDO | Se consiguen los resultados esperados.   |

El número de repeticiones realizadas en cada prueba ha sido de treinta veces.

## 8 CONCLUSIONES

Todos los objetivos de dicho proyecto han sido conseguidos.

Gracias a la realización de éste proyecto, se han consolidado y aumentado los conocimientos y competencias adquiridos a lo largo del Grado de Ingeniería Telemática acerca del desarrollo Java, Android y Python, éste último aprendido en el desarrollo del proyecto. Para realizar un servicio de tal envergadura ha sido necesario la optimización del código desarrollado y el aprendizaje de muchas de las funcionalidades de dichos lenguajes.

También ha sido necesario realizar un esquema muy detallado de lo que quería que el servicio realizase en cada módulo implementado. Estos tienen multitud de parámetros a considerar para un funcionamiento correcto.

Los conocimientos acerca de estándares domóticos también se han mejorado cuantiosamente dado la integración de la comunicación con éstos.

### 8.1 OBJETIVOS CONSEGUIDOS

Los objetivos establecidos inicialmente en éste proyecto eran el control domótico de una vivienda a través de un dispositivo móvil. Éste ha sido logrado satisfactoriamente.

El objetivo opcional el cual es el control por voz de los recursos domóticos del hogar también ha sido alcanzado satisfactoriamente, pues el desarrollo implementado permite controlar todo tipo de dispositivos domóticos por voz.

También se ha incluido el módulo de gestión de sesiones de usuarios, el cual se pensó que es una necesidad para la seguridad de los usuarios.

### 8.2 PROBLEMÁTICAS ENCONTRADAS

Se ha podido comprobar la dificultad de brindar un servicio estable y sin errores, ya que por problemas hardware, software o de red existen multitud de problemas a reducir y controlar.

Uno de los grandes problemas encontrados ha sido en el lado del dispositivo con Android. Como ha sido mencionado, la comunicación por medio de socket con el servidor para la versión 4 de Android deja mucho que desear, pues éste, ante pérdidas de conexión con el servidor en ocasiones no lo notifica a la aplicación, haciendo saltar las debidas excepciones.

Otro de los problemas encontrados en el desarrollo ha sido la conmutación entre Wi-Fi y LTE en dispositivos Android, se ha podido comprobar que los dispositivos cuando cambian una conexión de datos LTE a Wi-Fi el dispositivo no conmuta la conexión TCP establecida entre LTE a Wi-Fi si no que esta sigue fijada por la red móvil. De todos modos, éste problema ha sido solventado realizando una reconexión casi instantánea.

## 9 LÍNEAS DE FUTURO

Las posibles líneas de futuro para tal proyecto son variadas ya que todas las tecnologías vistas en están en continuo desarrollo y esto favorece numerosas funcionalidades a añadir.

En primer lugar, estaría la adición de una interfaz web de control. Esto mejoraría el servicio ya que aumenta la accesibilidad al servicio por parte del usuario.

Además, se podría realizar una aplicación para los teléfonos de la plataforma de Apple. Esto mejoraría el servicio también en gran medida ya que aumenta la accesibilidad por parte del usuario.

Aumentar la seguridad es otro punto a tener en cuenta. Se podría realizar una mejora a las medidas de seguridad adoptadas en este proyecto con protocolos de seguridad más robustos.

Aumentar los dispositivos KNX del banco de entrenamiento. Un gran aporte sería el disponer de más dispositivos KNX en el banco de entrenamiento ya que, en este proyecto, la implementación de dispositivos ha sido muy variada incorporando por ejemplo aire acondicionado KNX. No se ha podido probar dado que no se dispone de este dispositivo.

## REFERENCIAS

- Arnold, K., Gosling, J., & Holmes, D. (2001). *El lenguaje de programación Java TM*. ADDISON-WESLEY.
- ayControl. (01 de 2010). *ayControl*. Obtenido de <http://aycontrol.com/>
- Calimero. (2005). *The Calimero Project*. Obtenido de <http://calimero-project.github.io/>
- Carnegie Mellon, U. (2002). *CMUSphinx*. Obtenido de <http://cmusphinx.sourceforge.net/>
- EIBD. (22 de 06 de 2005). *auto.tuwien*. Obtenido de <https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd>
- Electric, S. (05 de 2013). *Schneider Electric*. Obtenido de <http://www.schneider-electric.com/press/es/es/schneider-electric-presenta-see-home-20-la-nueva-version-de-la-aplicacion-para-la-vivienda-knx/>
- Fernández Valdivielso, C., & Matías Maestro, I. R. (2004). *El proyecto domótico: metodología para la elaboración de proyectos y aplicaciones domóticas*. Colegio Oficial Ingenieros de Telecomunicación, 2004.
- Houseinhand. (01 de 2012). *Houseinhand*. Obtenido de <http://www.houseinhand.com/es/>
- KNX. (s.f.). *KNX*. Obtenido de <https://www.knx.org/knx-es/>
- LIRC. (1999). *lirc*. Obtenido de <http://www.lirc.org/>
- LonWorks. (1999). *LonWorks*. Obtenido de <http://www.lonworks.es/>
- Lutz, M. (2013). *Learning Python, 5th Edition*. O'Reilly Media.
- Phillips, B., & Stewart, C. (2016). *Programación con Android, edición 2016*. ANAYA MULTIMEDIA.
- StartCom. (2005). *StartCom*. Obtenido de <https://www.startssl.com/>
- Vargas Sandoval, J. A. (07 de 2013). Proyecto domótico. Instalación de un sistema basado en el estándar KNX. Linares, Ján, España.
- X10. (1993). *X10*. Obtenido de <http://www.x10.es/pagina-inicial.htm>
- Zhang, A. (2017). *SpeechRecognition 3.6.0 [Software]*. Obtenido de [https://github.com/Uberi/speech\\_recognition#readme](https://github.com/Uberi/speech_recognition#readme)

## DEFINICIONES Y ABREVIATURAS

### ABREVIATURAS

- KNX: (*EIB KONNEX*).
- ETS: (*Engineering Tool Software*).
- IPv4: (*Internet Protocol version 4*): Protocolo de internet versión 4.
- NAT: (*Network Address Translation*): Traducción de direcciones de red.
- VNP: (*Virtual Private Network*): Red privada Virtual.
- IP: (*Internet Protocol*): Protocolo de internet.
- DNS: (*Domain Name System*): Sistema de nombre de dominio.
- HTTP: (*Hypertext Transfer Protocol*): Protocolo de transferencia de hipertexto.
- TCP: (*Transmission Control Protocol*): Protocolo de control de transmisión.
- XML: (*eXtensible Markup Language*): Lenguaje de marcado extensible.
- JSON: (*JavaScript Object Notation*): Notación de objetos de JavaScript.
- DBA: (*Database Administrator*): Administrador de bases de datos.
- API: (*Application Programming Interface*): Interfaz de programación de aplicaciones.
- LIRC: (*Linux Infrared Remote Control*): Control remoto infrarrojo de Linux.
- SSL: (*Secure Sockets Layer*): Capa de sockets seguros.
- AES: (*Advanced Encryption Standard*): Estándar avanzado del cifrado.
- CBC: (*Cipher Block Chaining*): Encadenamiento de bloques cifrados.
- RSA: (*Rivest, Shamir y Adleman*).
- PEM: (*Privacy-enhanced Electronic Mail*): Correo electrónico reforzado con privacidad.
- USB: (*Universal Serial Bus*): Bus universal en serie.
- GIL (*Global Interpreter Lock*): Bloqueo global de intérprete.

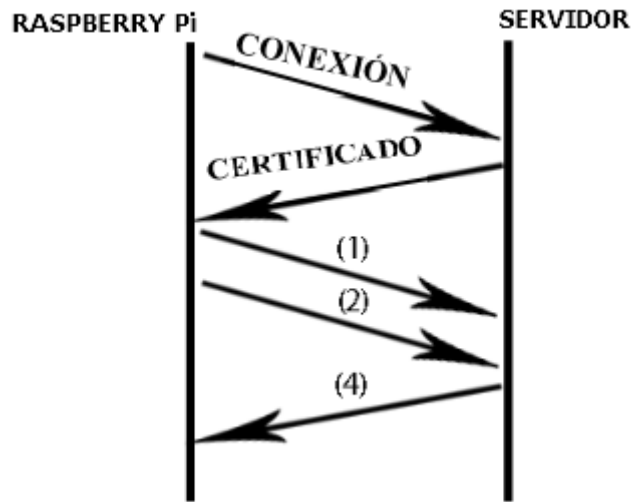
## 2. PLANOS

## ÍNDICE DE PLANOS

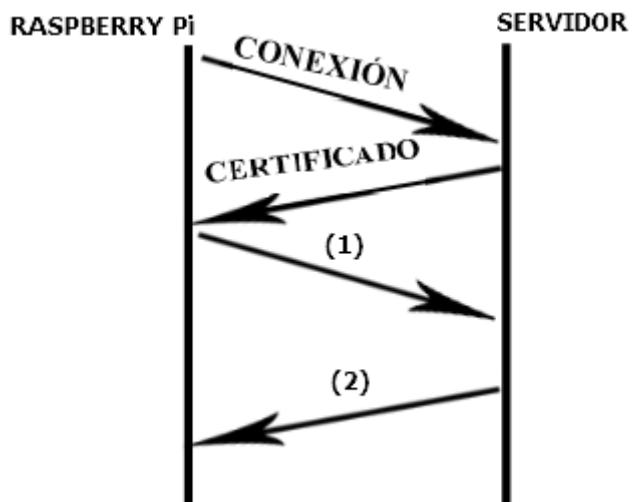
|    |   |     |
|----|---|-----|
| 1  | FIGURA 7. ESQUEMA NUEVA CONEXIÓN RASPBERRY PI .....     | 142 |
| 2  | FIGURA 8. ESQUEMA CONEXIÓN RASPBERRY PI .....           | 142 |
| 3  | FIGURA 9. ESQUEMA NUEVA CONEXIÓN ANDROID .....          | 143 |
| 4  | FIGURA 10. ESQUEMA CONEXIÓN ANDROID .....               | 143 |
| 5  | FIGURA 22. FLUJO DE CLASES DE CONEXIÓN EN SERVIDOR..... | 144 |
| 6  | FIGURA 23. MODELO DE PILA RASPBERRY PI .....            | 144 |
| 7  | FIGURA 26. DIAGRAMA ACTIVITYS ANDROID.....              | 146 |
| 8  | FIGURA 68. LOGIN RASPBERRY PI .....                     | 147 |
| 9  | FIGURA 69. MONITORIZACIÓN KNX.....                      | 148 |
| 10 | FIGURA 70. CONTROL POR VOZ.....                         | 149 |
| 11 | FIGURA 71. RECEPCIÓN DE DATOS .....                     | 150 |
| 12 | FIGURA 72. RECEPCIÓN DE DATOS UNO.....                  | 151 |
| 13 | FIGURA 73. RECEPCIÓN DE DATOS DOS.....                  | 152 |
| 14 | FIGURA 74. ESQUEMA CONEXIÓN INFRARROJO .....            | 153 |
| 15 | FIGURA 75. LOGIN ANDROID .....                          | 154 |
| 16 | FIGURA 76. AUTENTICACIÓN CLIENTES .....                 | 155 |
| 17 | FIGURA 77. RECEPCION DE DATOS SERVIDOR.....             | 156 |
| 18 | FIGURA 78. RECEPCION DE DATOS SERVIDOR UNO .....        | 157 |
| 19 | FIGURA 79. RECEPCION DE DATOS SERVIDOR DOS.....         | 158 |

|    |  |     |
|----|--|-----|
| 20 | FIGURA 80. RECEPCION DE DATOS SERVIDOR TRES..... | 159 |
| 21 | FIGURA 81. MODELO BASE DE DATOS .....            | 160 |

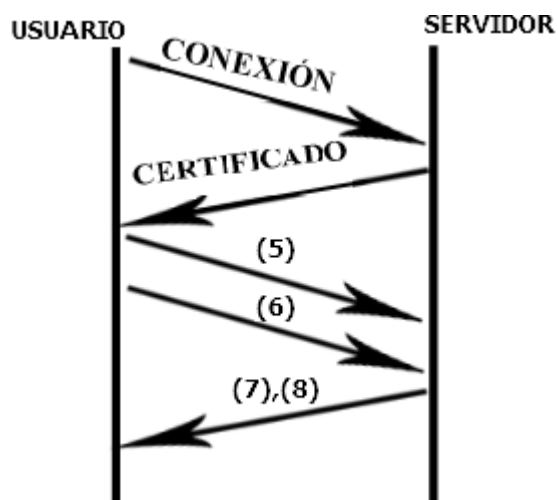
1 FIGURA 7. ESQUEMA NUEVA CONEXIÓN RASPBERRY PI



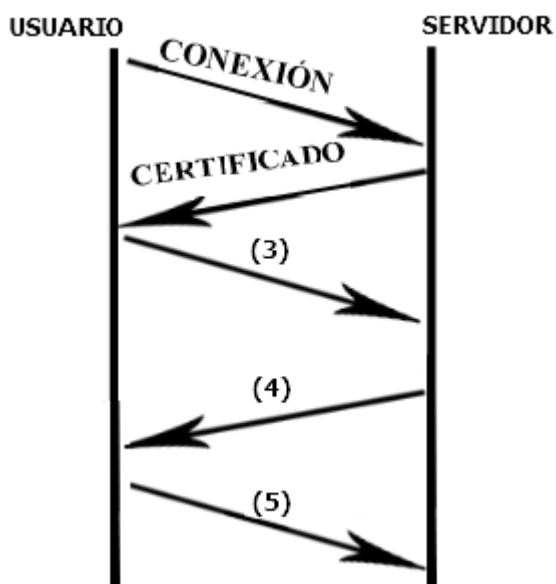
2 FIGURA 8. ESQUEMA CONEXIÓN RASPBERRY PI



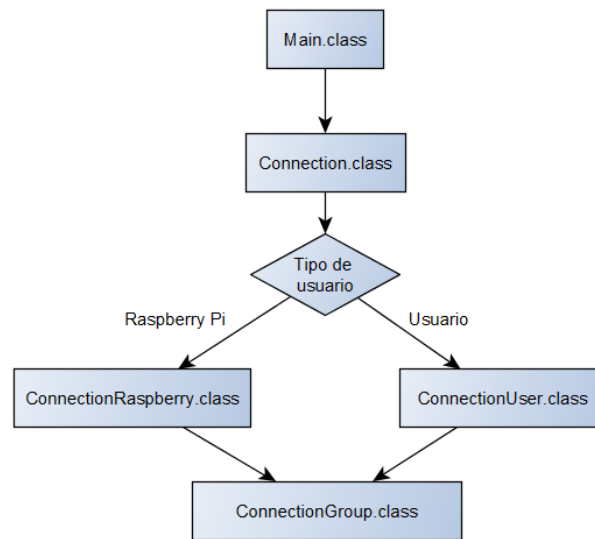
3 FIGURA 9. ESQUEMA NUEVA CONEXIÓN ANDROID



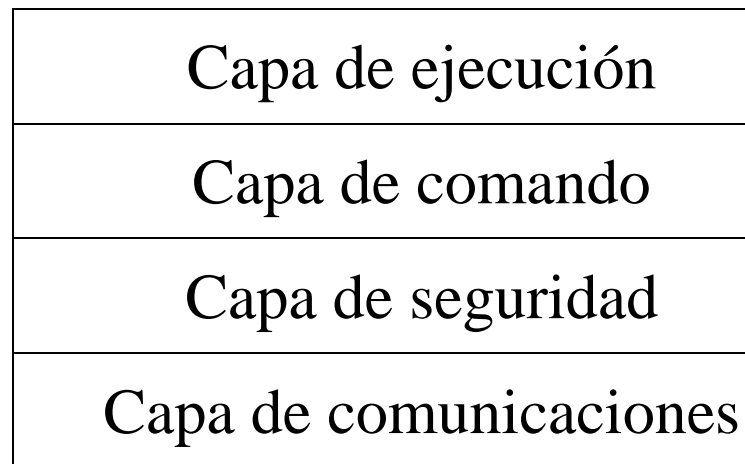
4 FIGURA 10. ESQUEMA CONEXIÓN ANDROID



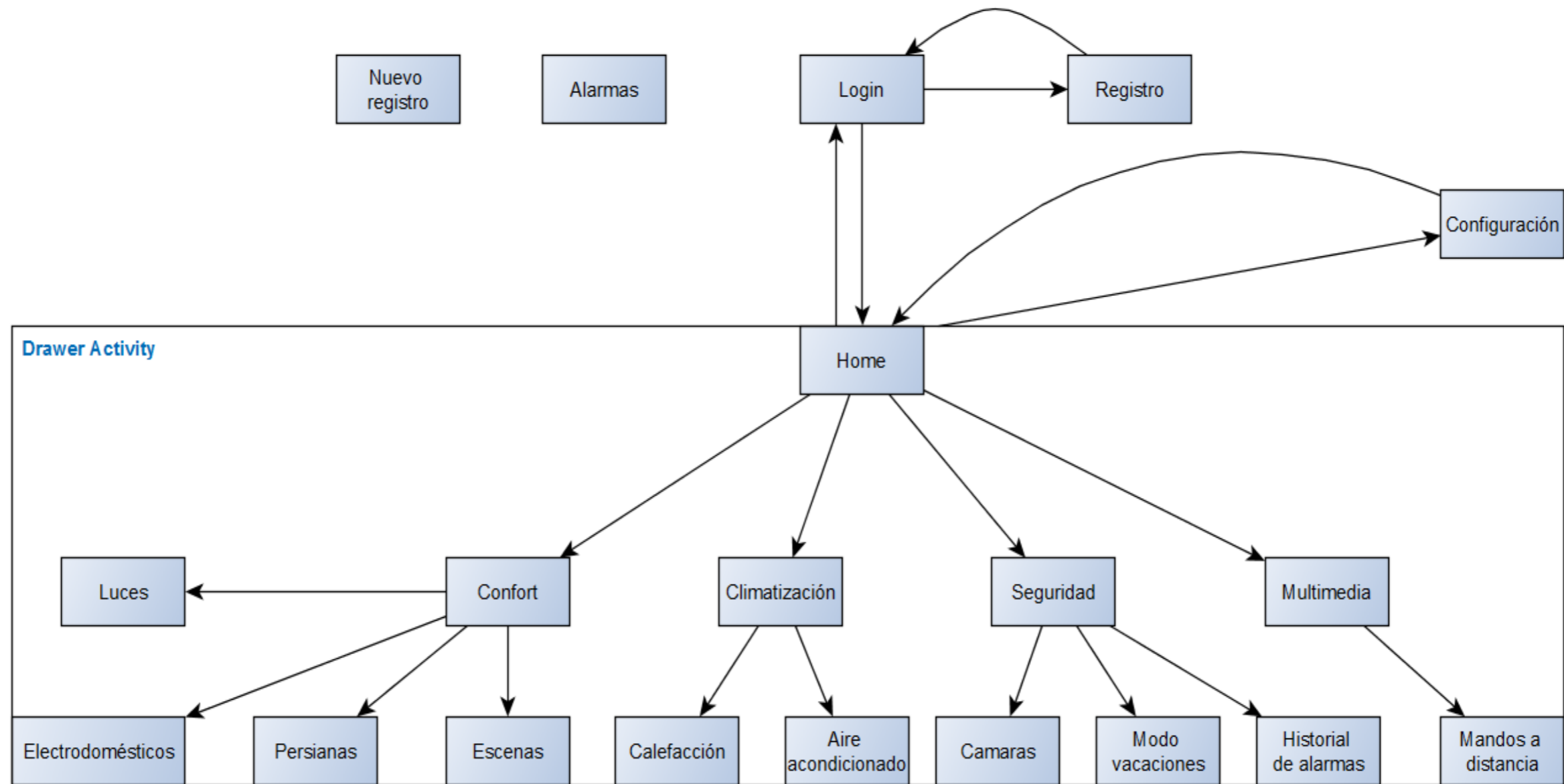
5 FIGURA 22. FLUJO DE CLASES DE CONEXIÓN EN SERVIDOR



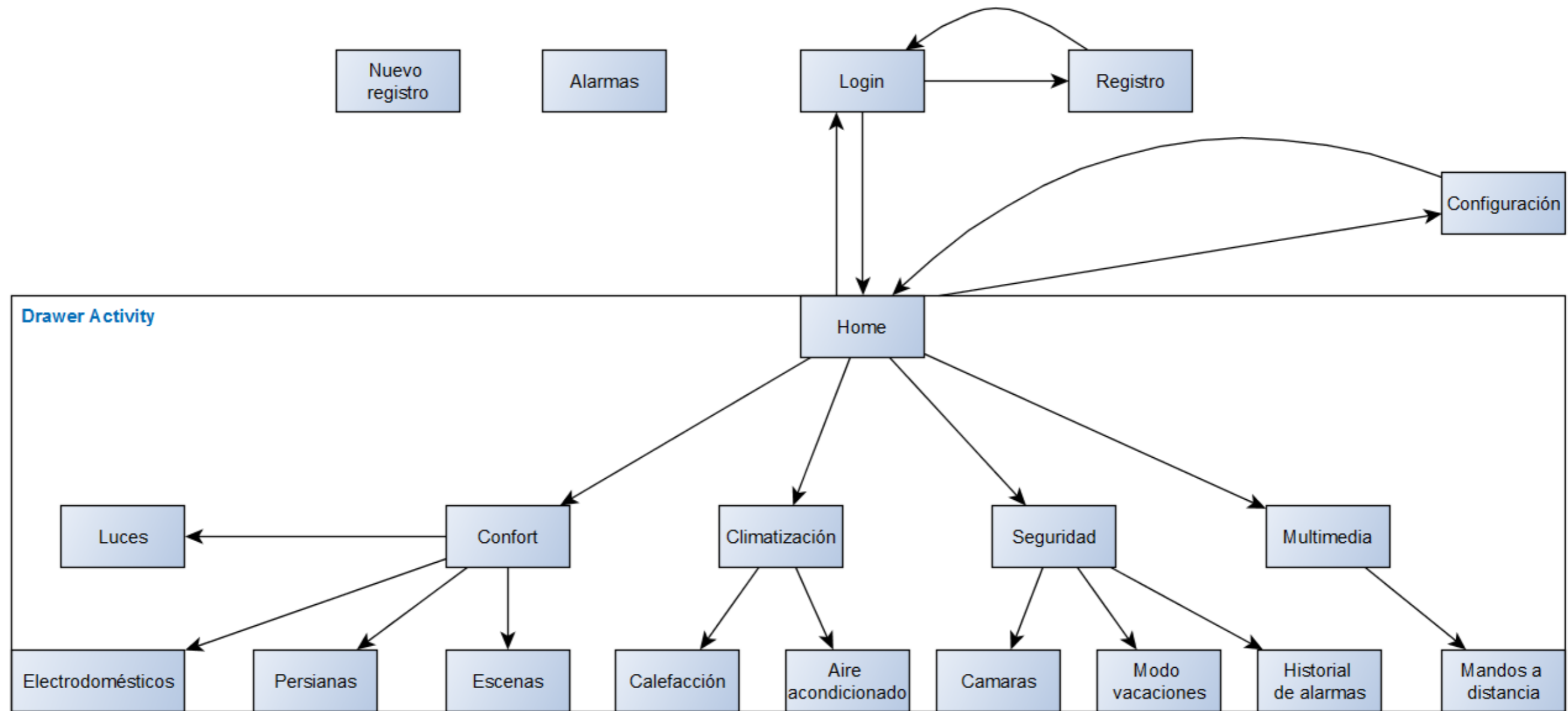
6 FIGURA 23. MODELO DE PILA RASPERRY PI



7 FIGURA 26. DIAGRAMA ACTIVITYS ANDROID



8 FIGURA 26. DIAGRAMA ACTIVITYS ANDROID



9 FIGURA 68. LOGIN RASPBERRY PI

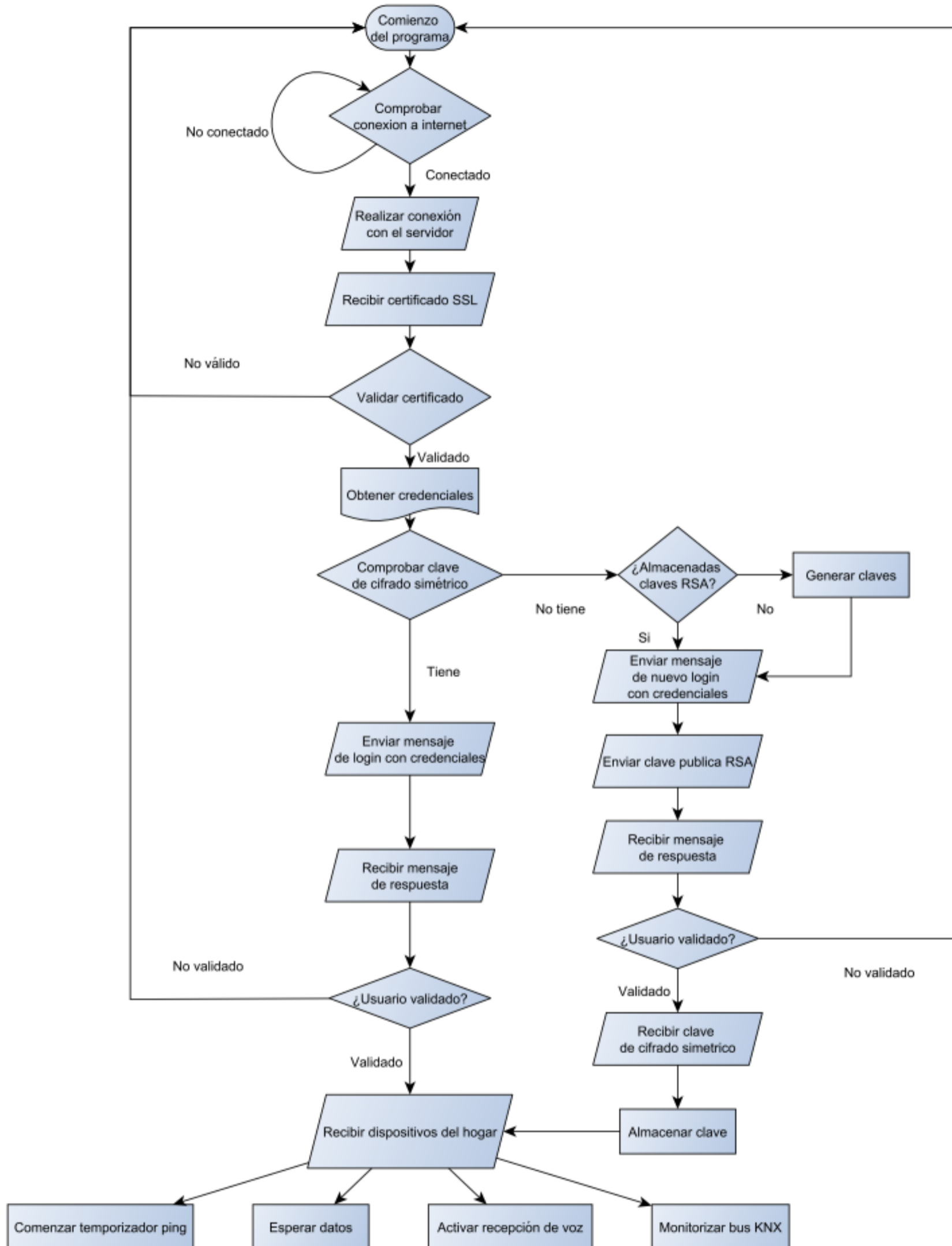


Figura 68. Login Raspberry P

10 FIGURA 69. MONITORIZACIÓN KNX

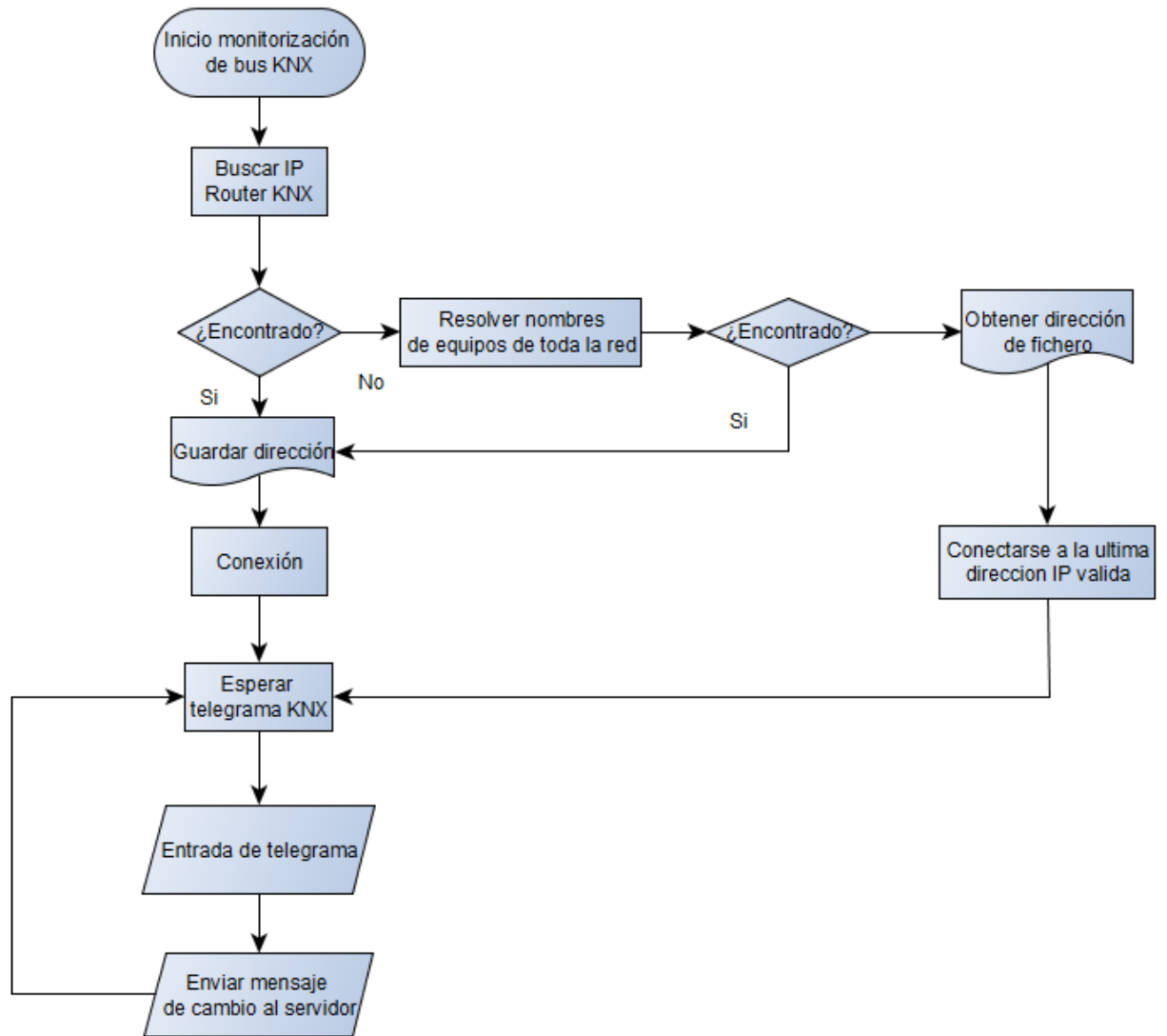


Figura 69. Monitorización KNX

11 FIGURA 70. CONTROL POR VOZ

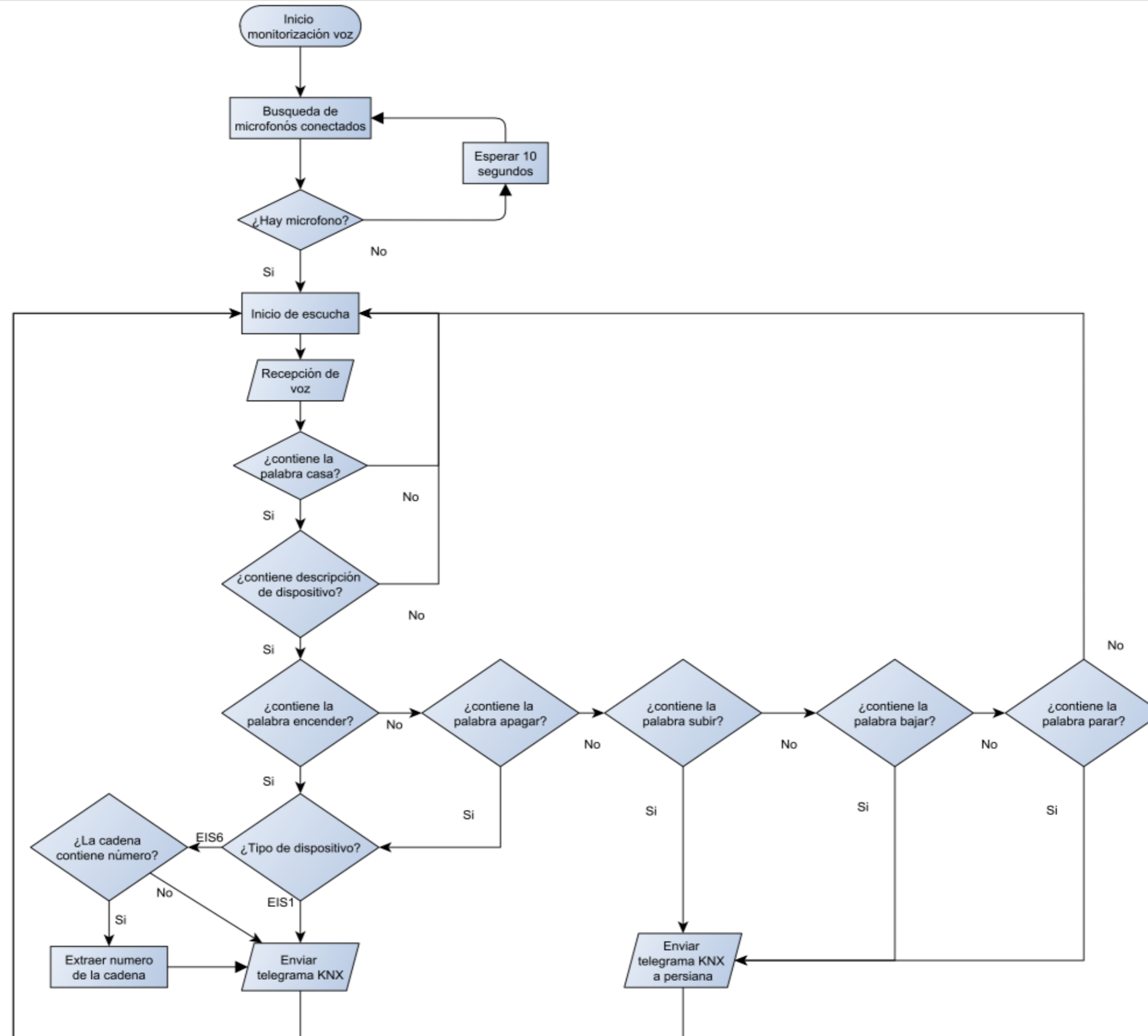


Figura 70. Control por voz

12 FIGURA 71. RECEPCIÓN DE DATOS

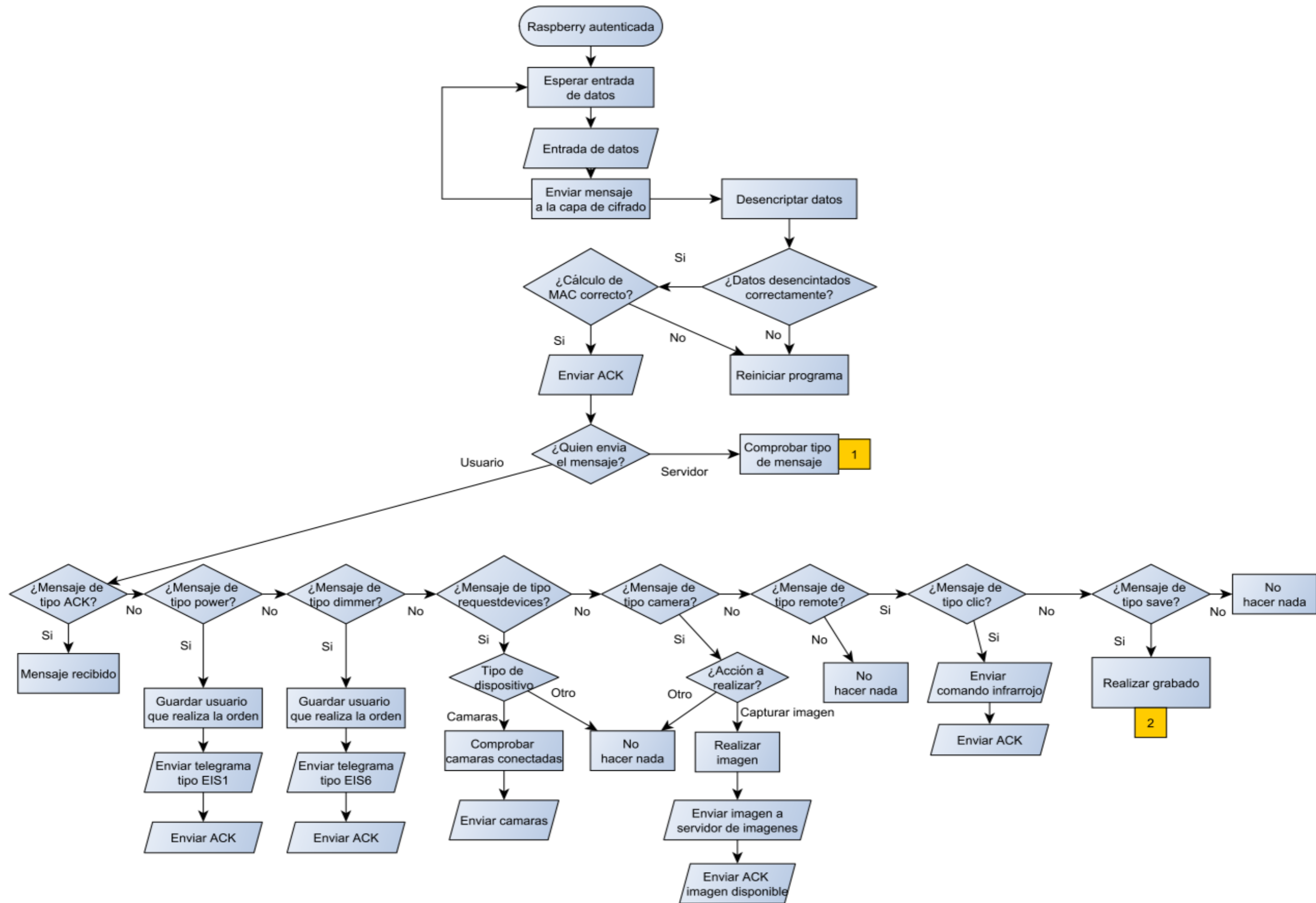


Figura 71. Recepción de datos

13 FIGURA 72. RECEPCIÓN DE DATOS UNO

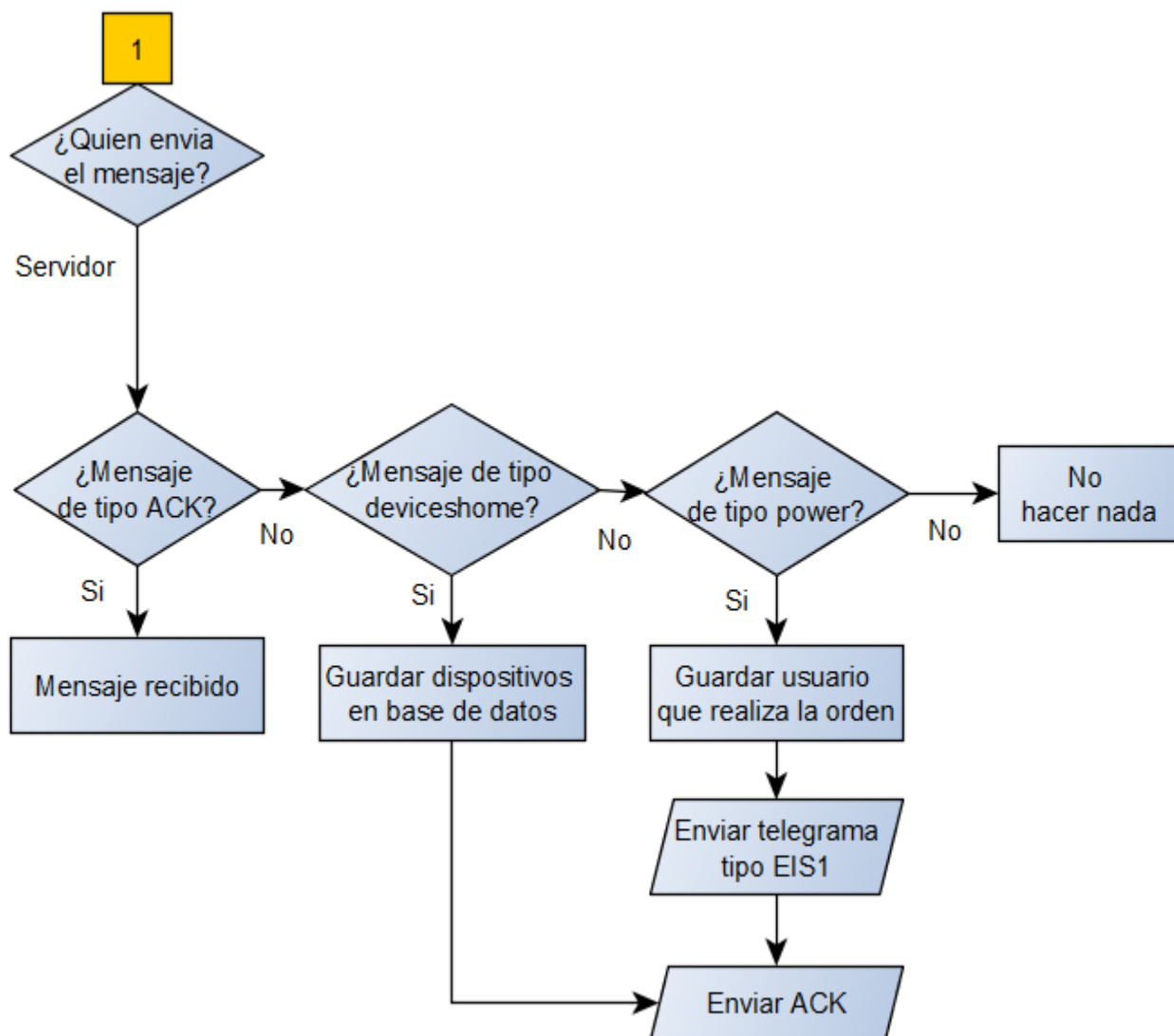


Figura 72. Recepción de datos uno

14 FIGURA 73. RECEPCIÓN DE DATOS DOS

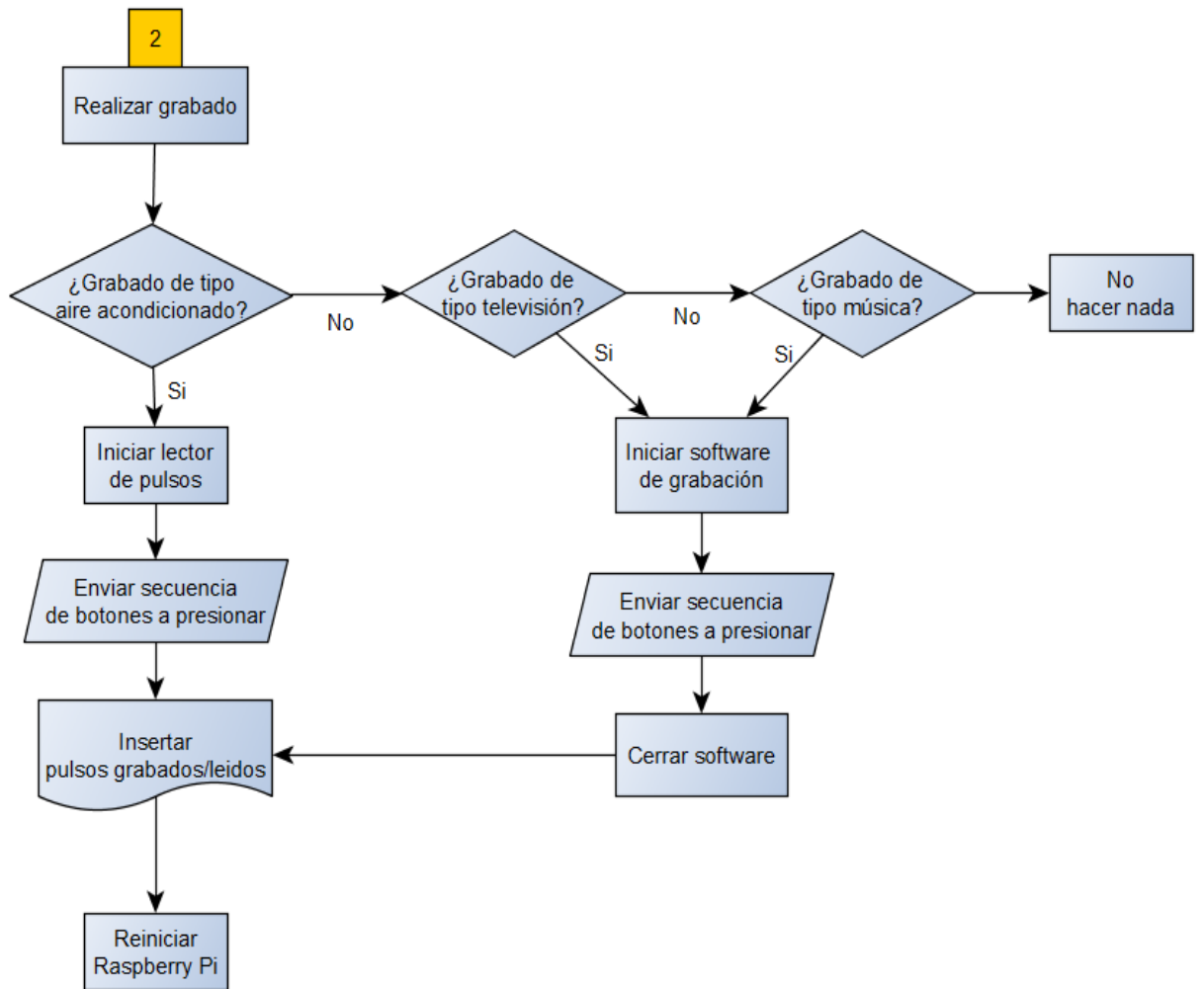


Figura 73. Recepción de datos dos

15 FIGURA 74. ESQUEMA CONEXIÓN INFRARROJO

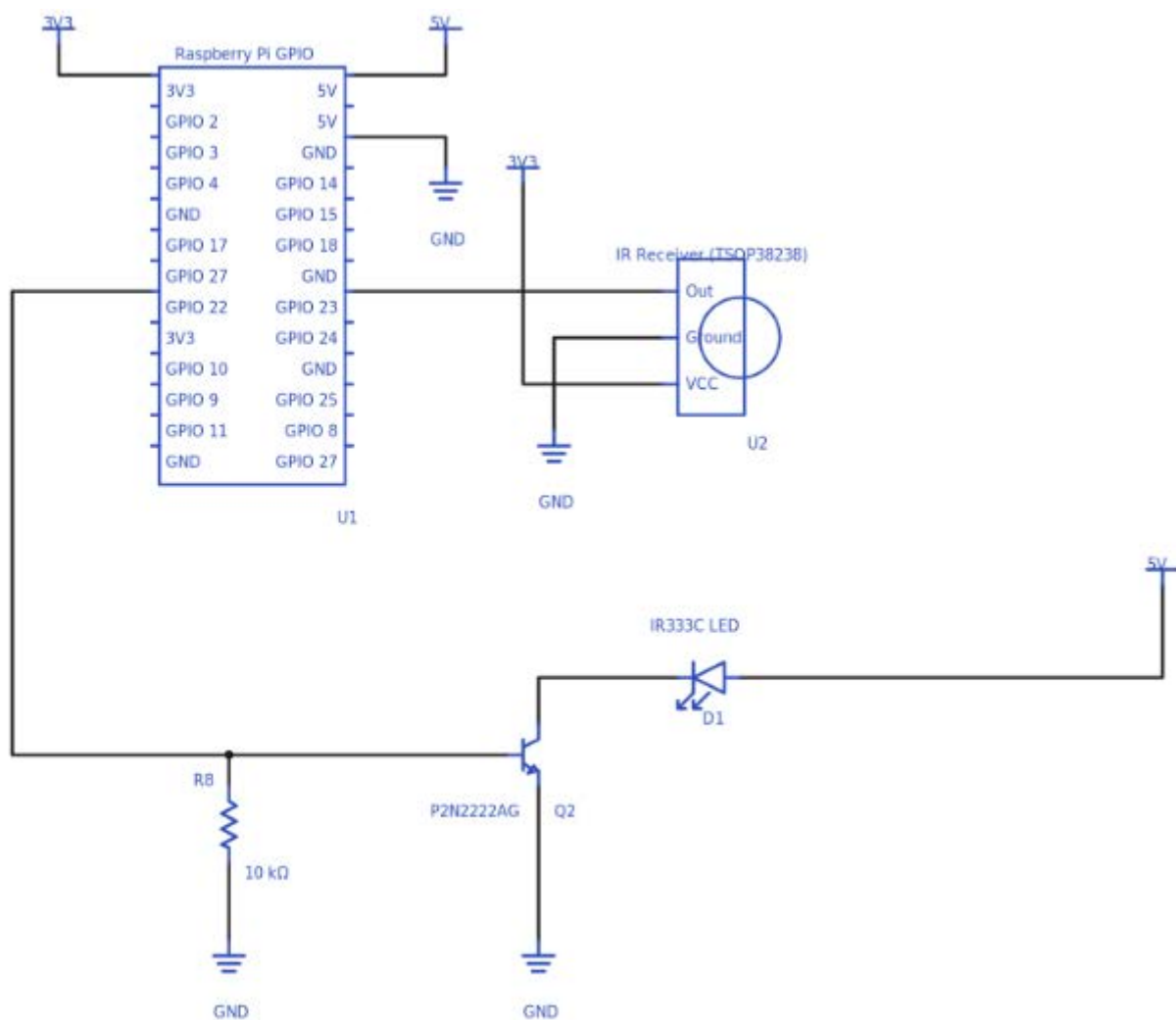


Figura 74. Esquema conexión infrarrojo

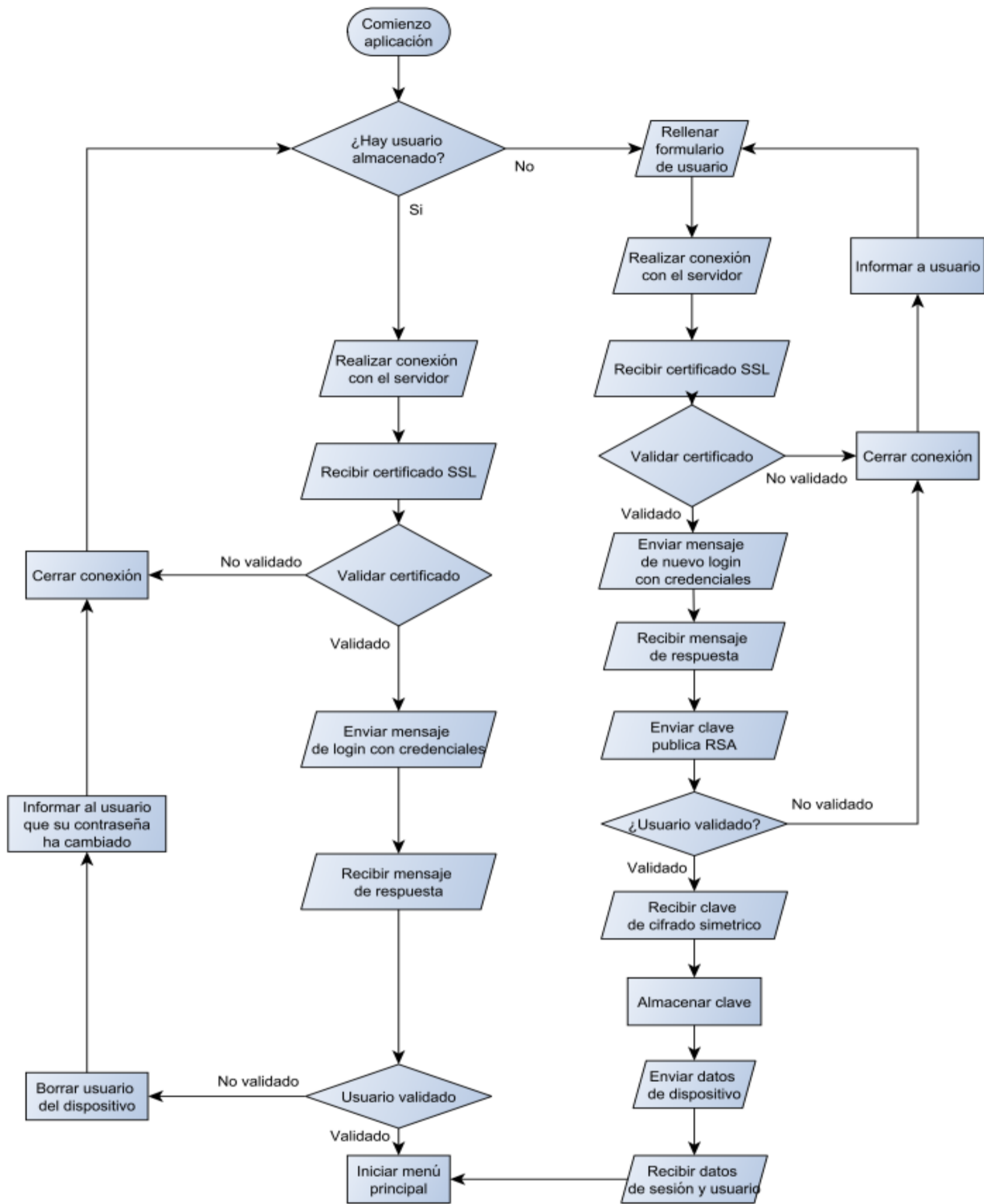


Figura 75. Login Android

17 FIGURA 76. AUTENTICACIÓN CLIENTES

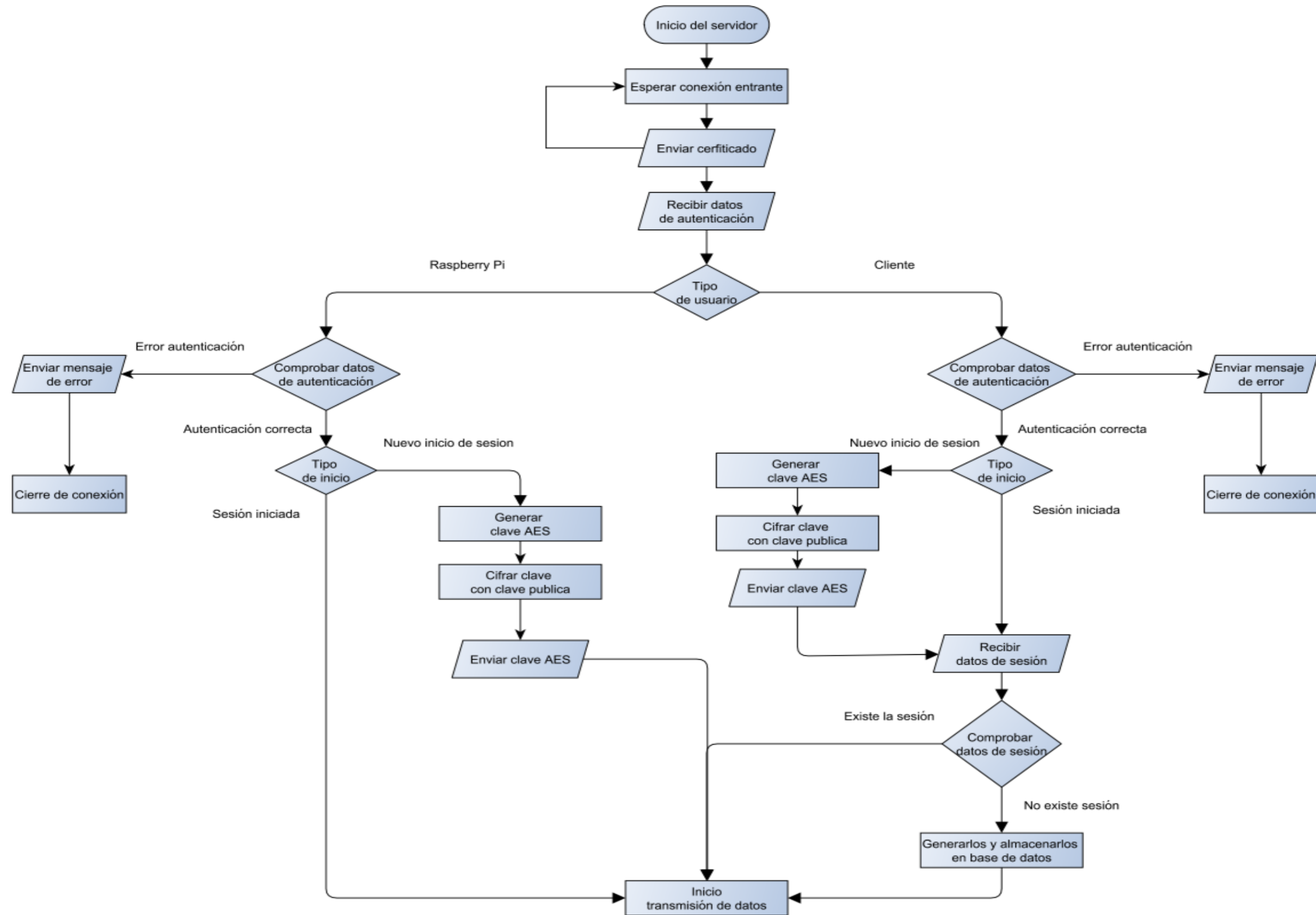


Figura 76. Autenticación clientes

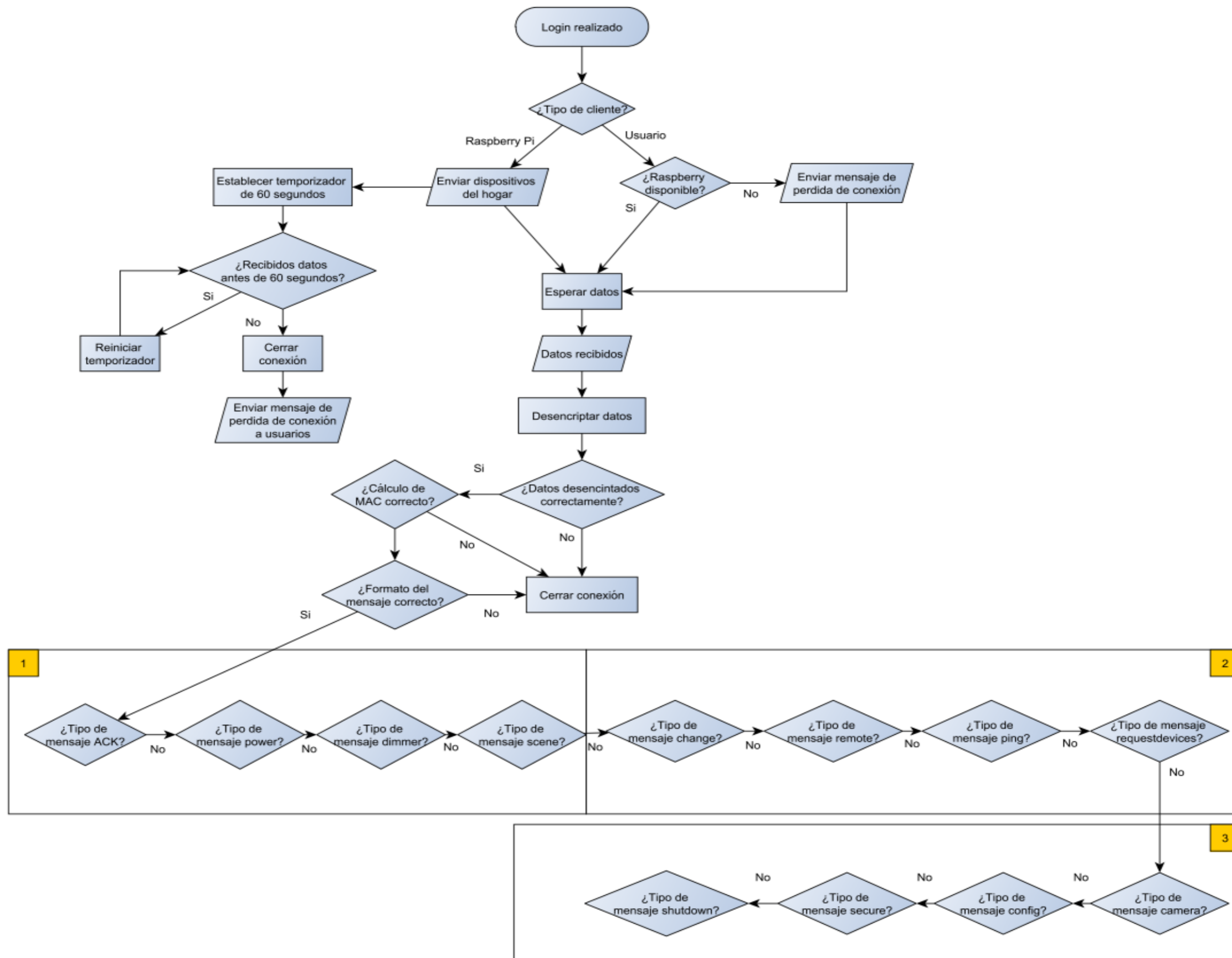


Figura 77. Recepción datos servidor

19 FIGURA 78. RECEPCION DE DATOS SERVIDOR UNO

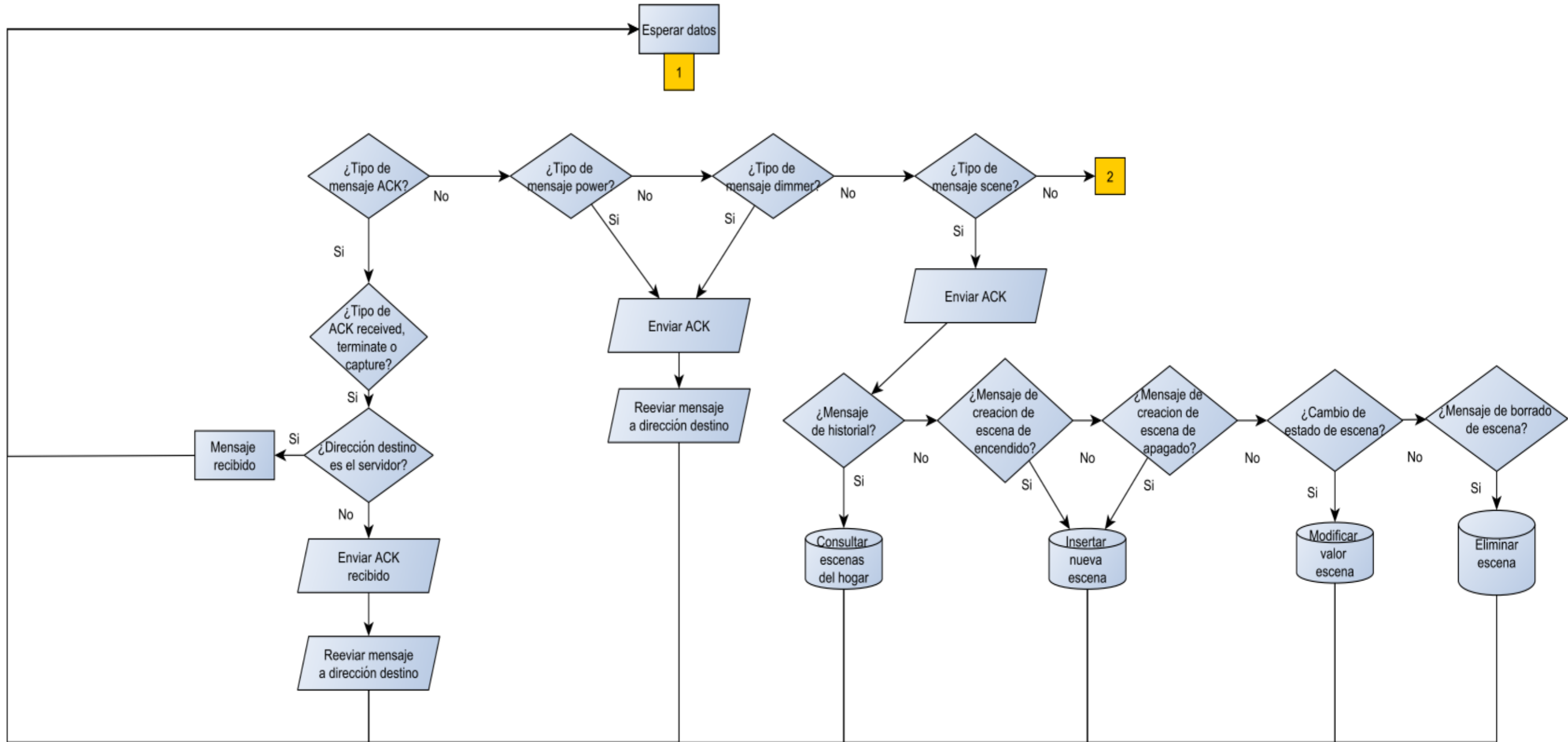


Figura 78. Recepción datos servidor uno

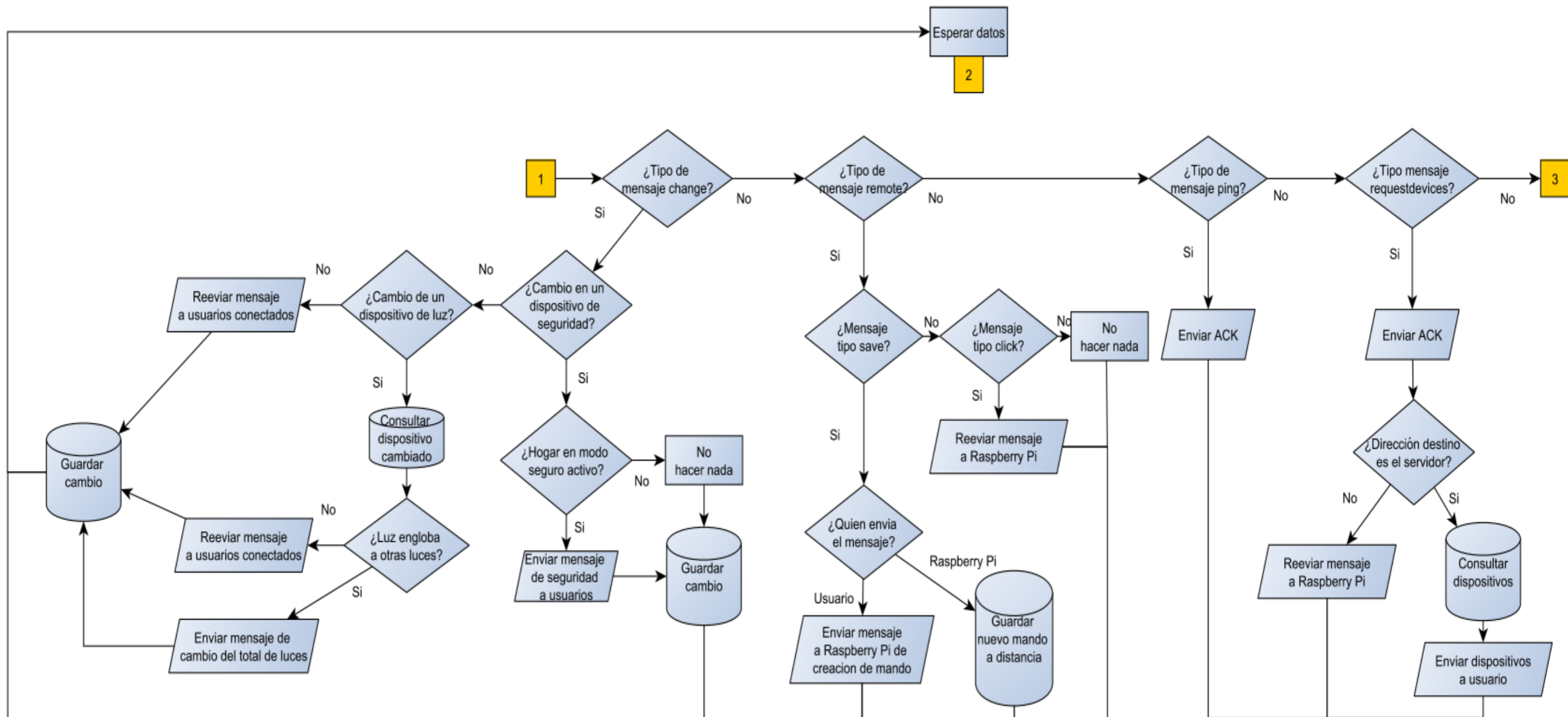


Figura 79. Recepción datos servidor dos

21 FIGURA 80. RECEPCION DE DATOS SERVIDOR TRES

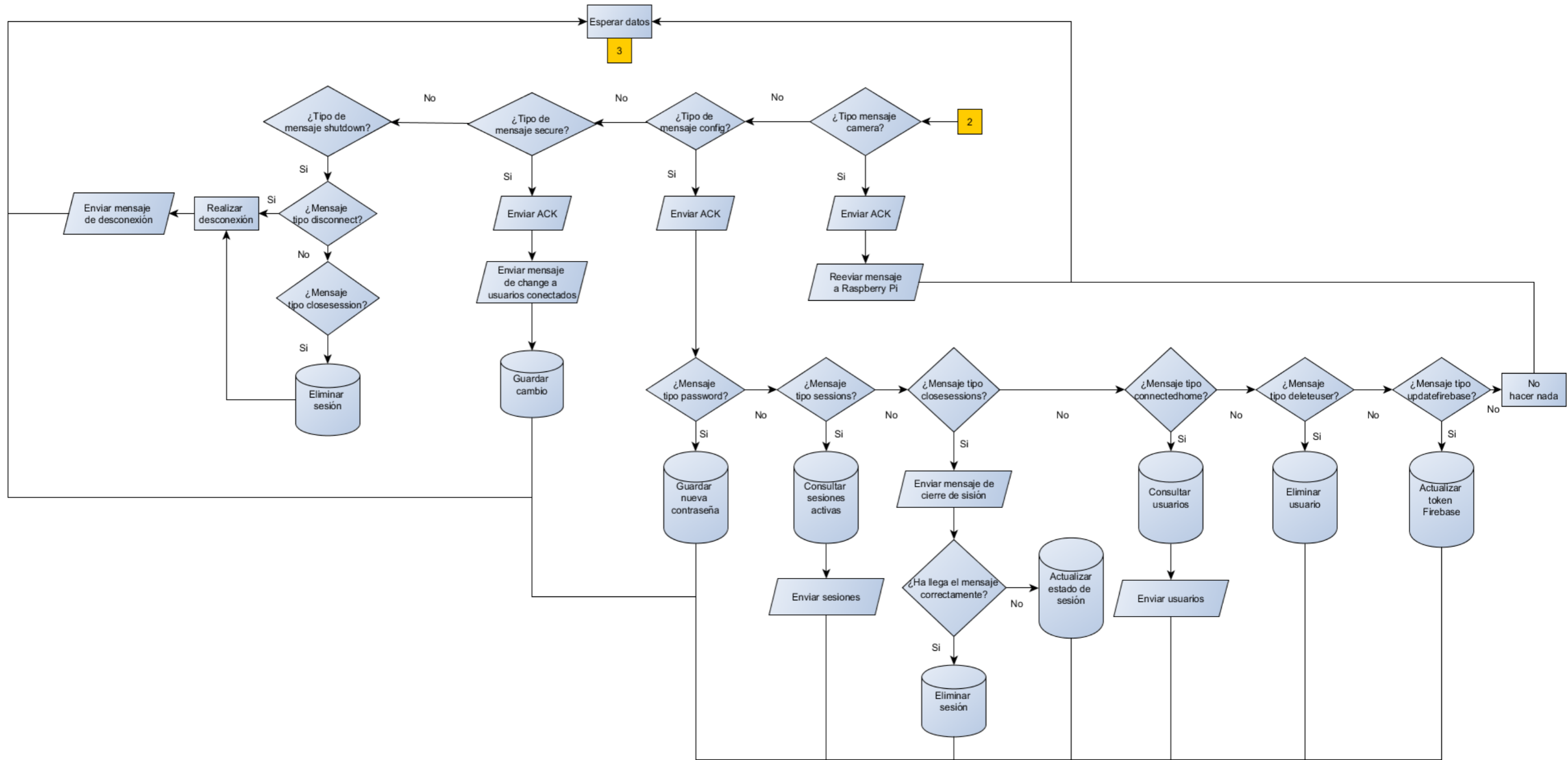


Figura 80. Recepción datos servidor tres

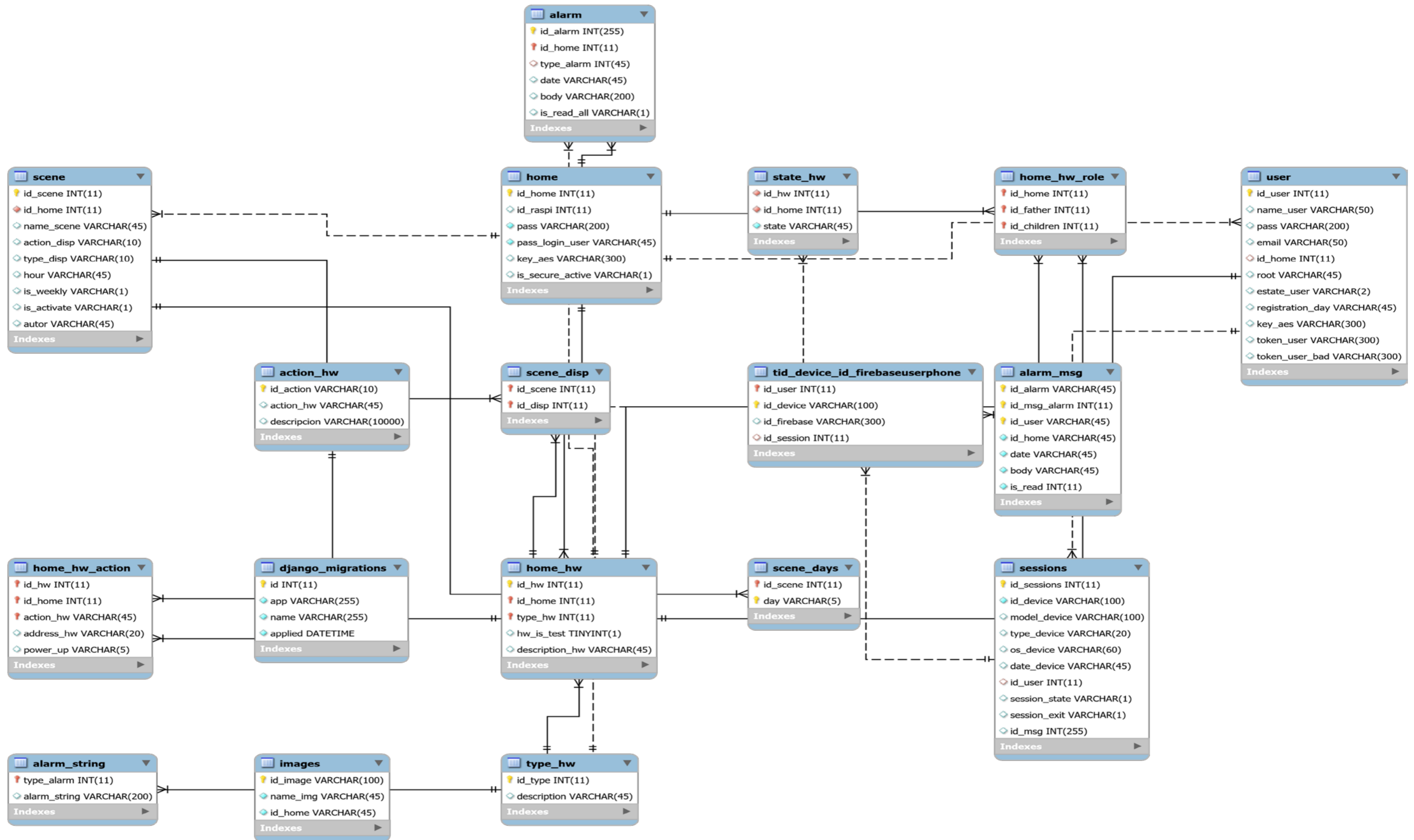


Figura 81. Modelo base de datos

### 3. PLIEGO DE CONDICIONES

## INDICE DE PLIEGO DE CONDICIONES

|  |     |
|--|-----|
| 1 CONDICIONES GENERALES.....                         | 162 |
| 1.1.1 INSTALACIÓN .....                              | 162 |
| 1.1.2 DISPOSICIONES LEGALES Y NORMAS APLICADAS ..... | 162 |
| 1.1.3 ESPECIFICACIÓN POR FUNCIONES DOMÓTICAS .....   | 162 |
| 1.1.4 CONDICIONES PARTICULARES.....                  | 164 |

### 1 CONDICIONES GENERALES

#### 1.1.1 INSTALACIÓN

La instalación y programación del sistema domótico se llevará a cabo por personal cualificado especializado en éste campo.

Todo se hará respetando las normativas referentes a instalaciones domóticas, REBT (842/2002), e ITC-BT-51, e instrucciones del fabricante de los diferentes dispositivos.

El cableado de la instalación domótica discurrirá en tubos independientes a los circuitos de fuerza y de la misma manera se hará con las derivaciones en cajas separadas, siguiendo la ITC-BT-21, pudiendo compartir la misma bandeja siempre y cuando se respeten las distancias precauciones previstas en la normativa, en especial el REBT.

#### 1.1.2 DISPOSICIONES LEGALES Y NORMAS APLICADAS

Los reglamentos y normas que se han tenido en consideración en el siguiente proyecto son los siguientes:

- Instrucción ITC-BT-51 del R.E.B.T. Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.
- Instrucción ITC-BT-04 del R.E.B.T. Documentación y puesta en servicio de las instalaciones.
- Instrucción ITC-BT-05 del R.E.B.T. Verificaciones e inspecciones.
- Norma UNE 157001. Criterios generales para la elaboración de proyectos.

#### 1.1.3 ESPECIFICACIÓN POR FUNCIONES DOMÓTICAS

Para que el diseño del software domótico resulte sencillo, lo más conveniente es agrupar las distintas aplicaciones o funcionalidades según el área de la domótica de que se trate. Recordamos que las funciones básicas de la domótica utilizadas: confort, gestión de la seguridad. También en el diseño de dicho software se ha incluido un módulo de multimedia.

Así pues, comenzamos definiendo las funciones que se desean implementar en la vivienda:

### 1.1.3.1 CONFORT

#### *CONTROL DE LA ILUMINACIÓN*

Se permitirá control de puntos de luz en la vivienda remotamente, dada la comodidad que supone para el usuario. Dicho control implica encendido/apagado y regulación.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

Una característica adicional será la posibilidad que brindará dicha aplicación para la programación de la iluminación de la totalidad del hogar.

#### *CONTROL DE PERSIANAS*

Será posible el control de las persianas de la vivienda, al igual que en el caso anterior de forma remota.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

Una característica adicional será la posibilidad que brindará dicha aplicación para la programación de las persianas de la totalidad del hogar.

#### *CONTROL DE ELECTRODOMÉSTICOS*

Será posible el control de los electrodomésticos de la vivienda, al igual que en el caso anterior de forma remota.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

Una característica adicional será la posibilidad que brindará dicha aplicación para la programación de los electrodomésticos de la totalidad del hogar.

#### *CONTROL DE ESCENAS*

Será posible el control, creación y eliminación de escenas de la vivienda, al igual que en el caso anterior de forma remota.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

#### *CONTROL DE CALEFACCIÓN*

Será posible el control de calefacción de la vivienda, al igual que en el caso anterior de forma remota.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

## *CONTROL DE AIRE ACONDICIONADO*

Será posible el control de aire acondicionado de la vivienda, al igual que en el caso anterior de forma remota.

El control remoto se realizará a través de una app Android, podrá hacerse a través de una conexión a Internet.

### *1.1.3.2 GESTION DE SEGURIDAD*

#### *SEGURIDAD*

Los usuarios a través de la aplicación Android podrán disponer de imágenes de su hogar en todo momento en función del número de cámaras conectadas.

Los usuarios a través de la aplicación Android ante la detección en el hogar de alertas de seguridad serán notificados indicándole el incidente ocurrido. Las alertas ante las que los usuarios serán notificados son las siguientes:

- Fugas de agua.
- Detección de incendios.
- Apertura magnética.
- Riesgo de velocidad de viento.
- Detección de fugas de agua.

### *1.1.4 CONDICIONES PARTICULARES*

#### *1.1.4.1 REQUISITOS HARDWARE*

El hardware mínimo recomendado para el funcionamiento del servicio es el siguiente.

#### *SERVIDOR*

Para la implementación del servidor del servicio el hardware mínimo es el siguiente:

- Procesador Intel Core i3.
- 4 GB de memoria RAM.
- Tarjeta de red Gigabit Ethernet.

#### *HOGAR*

Para la implementación del servicio el hardware mínimo en el hogar es el siguiente:

- Raspberry Pi 2 Model B.
- Tarjeta micro SD 8 GB.
- Micrófono Jack stereo 3.5mm o Micrófono USB.

## CLIENTES

Para la implementación del servicio el hardware mínimo en el lado del cliente es cualquier dispositivo Android.

### 1.1.4.2 REQUISITOS SOFTWARE

El software mínimo recomendado para el funcionamiento del servicio es el siguiente.

## SERVIDOR

Para la implementación del servidor del servicio el software mínimo es el siguiente:

- Windows 7 Profesional.
- IDE de desarrollo Java.
- Máquina virtual de Java, version 8.
- MySQL Workbench 6.3.

## HOGAR

Para la implementación del servicio el hardware mínimo en el hogar es el siguiente:

- Debian jessie 8.0 o superior.
- Python 3.4 o superior.
- MySQL-python 1.2.5.
- Python SpeechRecognition 3.6.0.
- LIRC 0.9.0
- EIBD.

## CLIENTES

Para la implementación del servicio el software mínimo en el lado del cliente es disponer de la version Android 4 *Ice Cream Sandwich*.

### 1.1.4.3 CONCEPTOS COMPRENDIDOS

Es competencia exclusiva del Instalador y, por lo tanto, queda totalmente incluido en el precio ofertado, el suministro de todos los elementos y materiales, mano de obra, medios auxiliares y, en general, todos aquellos elementos y/o conceptos que sean necesarios para el perfecto acabado y puesta a punto de las instalaciones, según se describen en la memoria, son representadas en los planos, quedan relacionadas de forma básica en el presupuesto y cuya calidad y características de montaje se indican en el Pliego de Condiciones Técnicas.

Queda entendido que los cuatro Documentos de Proyecto, es decir, Memoria, Presupuesto, Planos y Pliego de Condiciones Técnicas forman todo un conjunto. Si fuese advertida o existiese alguna discrepancia entre estos cuatro documentos, su interpretación será la que determine la Dirección de Obra. Salvo indicación contraria en su oferta, lo que debe quedar explícitamente indicado en contrato, queda entendido que el instalador acepta

éste criterio y no podrá formular reclamación alguna por motivo de omisiones y/o discrepancias entre cualquiera de los cuatro documentos que integran el proyecto.

Cualquier exclusión, incluida implícita o explícitamente por el instalador en su oferta y que difiera de los conceptos expuestos en los párrafos anteriores, no tendrá ninguna validez, salvo que, en el contrato de una forma particular y explícita, se manifieste la correspondiente exclusión.

Es responsabilidad del Instalador el cumplimiento de toda la normativa oficial vigente aplicable al proyecto. Durante la realización de éste proyecto se ha puesto el máximo empeño en cumplir toda la normativa oficial vigente al respecto. No obstante, si en el mismo existiesen conceptos que se desviasen o no cumplieren con las mismas, es obligación del instalador comunicarlo en su oferta y en la forma que se describirá más adelante.

Queda, por tanto, obligado el instalador a efectuar una revisión del proyecto, previo a la presentación de su oferta, debiendo indicar, expresamente, en la misma, cualquier deficiencia a éste respecto o, en caso contrario, su conformidad con el proyecto en materia de cumplimiento de toda la normativa oficial vigente aplicable al mismo.

El instalador efectuará a su cargo el plan de seguridad y el seguimiento correspondiente a sus trabajos, debiendo disponer de todos los elementos de seguridad, auxiliares y de control exigidos por la legislación vigente, todo ello con la debida coordinación en relación al resto de la obra, por lo que será preceptiva la compatibilidad y aceptación de éste trabajo con el plan de seguridad general de la obra y, en cualquier caso, deberá contar con la conformidad de la Dirección Técnica y el contratista general.

Quedan incluidos también, como parte de los trabajos del instalador, la preparación de todos los planos de obra, así como la gestión y preparación de toda la documentación técnica necesaria, incluido visado y legalizado de proyectos y certificados de obra, así como su tramitación ante los diferentes organismos oficiales, al objeto de obtener todos los permisos requeridos de acuerdo a la legislación.

También queda incluidas la realización de todas las pruebas de puesta en marcha de las instalaciones, realizadas según las indicaciones de la dirección de obra. No se procederá a efectuar la recepción provisional si todo lo anterior no estuviese debidamente cumplimentado a satisfacción de la dirección de obra.

Asimismo, quedan incluidos todos los trabajos correspondientes a la definición, coordinación e instalación de todas las acometidas de servicios, tales como electricidad, agua, gas, saneamiento y otros que pudieran requerirse, ya sean de forma provisional para efectuar los montajes en obra o de forma definitiva para satisfacer las necesidades del proyecto.

Se entiende, por tanto, que estos trabajos quedan plenamente incluidos en la oferta del instalador, salvo que se indique expresamente lo contrario.

Queda, por tanto, el Instalador enterado por éste pliego de condiciones que es responsabilidad suya la realización de las comprobaciones indicadas, previo a la presentación de la oferta, así como la presentación en tiempo, modo y forma de toda la documentación mencionada y la consecución de los correspondientes permisos.

El instalador, en caso de subcontratación, o la empresa responsable de su contratación, no podrán formular reclamación alguna con respecto a éste concepto, ya sea por omisión, desconocimiento o cualquier otra causa.

#### 1.1.4.4 CONCEPTOS NO COMPRENDIDOS

En general, están excluidos de realización por parte del instalador los conceptos que responden a actividades de albañilería. En general, cualquier tipo de albañilería necesaria para el montaje de las instalaciones, aunque a priori no se vayan a realizar ninguna.

#### 1.1.4.5 CALIDADES

Cualquier elemento, máquina, material y, en general, cualquier concepto en el que pueda ser definible una calidad, ésta será la indicada en el proyecto, bien determinada por una marca comercial o por una especificación concreta. Si no estuviese definida una calidad, la dirección de obra podrá elegir la que corresponda en el mercado a niveles considerados similares a los del resto de los materiales especificados en proyecto. En éste caso, el instalador queda obligado, por éste pliego de condiciones técnicas, a aceptar el material que le indique la Dirección de Obra.

Si el instalador propusiese una calidad similar a la especificada en proyecto, corresponde exclusivamente a la dirección de obra definir si ésta es o no similar.

Por tanto, toda marca o calidad que no sea la específicamente indicada en el documento de medición y presupuesto o en cualquier otro documento del proyecto deberá haber sido aprobada por escrito por la dirección de obra previamente a su instalación, pudiendo ser rechazada, por tanto, sin perjuicio de ningún tipo para la propiedad, si no fuese cumplido éste requisito.

Todos los materiales y equipos deberán ser productos normalizados de catálogo de fabricantes dedicados con regularidad a la fabricación de tales materiales o equipos y deberán ser de primera calidad y del más reciente diseño del fabricante que cumpla con los requisitos de estas especificaciones y la normativa vigente. Salvo indicación expresa escrita en contrario por la dirección de obra, no se aceptará ningún material y/o equipo cuya fecha de fabricación sea anterior, en 9 meses o más, a la fecha de contrato del instalador.

Todos los componentes principales de equipos deberán llevar el nombre, la dirección del fabricante y el modelo y número de serie en una placa fijada con seguridad en un sitio visible. No se aceptará la placa del agente distribuidor. En aquellos equipos en los que se requiera placa o timbre autorizados y/o colocados por la delegación de industria o cualquier otro organismo oficial, será competencia exclusiva del instalador procurar la correspondiente placa y abonar cualquier derecho o tasa exigible al respecto.

Durante la obra, el instalador queda obligado a presentar a la dirección de obra cuantos materiales o muestras de los mismos le sean solicitados. En el caso de materiales voluminosos, se admitirán catálogos que reflejen perfectamente las características, terminado y composición de los materiales de que se trate.

#### 1.1.4.6 REGLAMENTACIÓN DE OBLIGADO CUMPLIMIENTO

Con total independencia de las prescripciones indicadas en los documentos del proyecto, es prioritario para el instalador el cumplimiento de cualquier reglamentación de obligado cumplimiento que afecte, directa o indirectamente, a su instalación, bien sea de índole nacional, autonómico, municipal, de compañías o, en general, de cualquier ente que pueda afectar a la puesta en marcha legal y necesaria para la consecución de las funciones previstas en la vivienda. El concepto de cumplimiento de normativa se refiere no sólo al cumplimiento de toda normativa del propio equipo o instalación, sino también al cumplimiento de cualquier normativa exigible durante el montaje, funcionamiento y/o rendimiento del equipo y/o sistema.

Es, por tanto, competencia, obligación y responsabilidad del instalador la previa revisión del proyecto antes de la presentación de su oferta y, una vez adjudicado el contrato, antes de que realice ningún pedido, ni que ejecute ningún montaje.

Esta segunda revisión del proyecto, a efectos de cumplimiento de normativa, se requiere tanto por si hubiera habido una modificación en la normativa aplicable después de la presentación de la oferta, como si, con motivo de alguna modificación relevante sobre el proyecto original, ésta pudiera contravenir cualquier normativa aplicable. Si esto ocurriera, queda obligado el instalador a exponerlo ante la dirección técnica y la propiedad. Esta comunicación deberá ser realizada por escrito y entregada en mano a la dirección técnica de obra.

Una vez iniciados los trabajos o pedidos los materiales relativos a la instalación contratada, cualquier modificación que fuera necesaria realizar para cumplimiento de normativa, ya sea por olvido, negligencia o por modificación de la misma, será realizada con cargo total al instalador y sin ningún coste para la propiedad u otros oficios o contratistas, reservándose ésta los derechos por reclamación de daños y perjuicios en la forma que se considere afectada.

Queda, por tanto, el instalador enterado por éste pliego de condiciones que no podrá justificar incumplimiento de normativa por identificación de proyecto, ya sea antes o después de la adjudicación de su contrato o por instrucciones directas de la dirección de obra y/o propiedad.

#### 1.1.4.7 CONDICIONES ECONÓMICAS

#### 1.1.4.8 ABONO DE LA OBRA

En el contrato se deberá fijar detalladamente la forma y plazos que se abonarán las obras.

Las liquidaciones parciales que puedan establecerse tendrán carácter de documentos provisionales a buena cuenta, sujetos a las certificaciones que resulten de la liquidación final. No suponiendo, dichas liquidaciones, aprobación ni recepción de las obras que comprenden.

Terminadas las obras se procederá a la liquidación final que se efectuará de acuerdo con los criterios establecidos en el contrato.

#### 1.1.4.9 PRECIOS

El contratista presentará, al formalizarse el contrato, relación de los precios de las unidades de obra que integran el proyecto, los cuales de ser aceptados tendrán valor contractual y se aplicarán a las posibles variaciones que pueda haber.

Estos precios unitarios, se entiende que comprenden la ejecución total de la unidad de obra, incluyendo todos los trabajos aún los complementarios y los materiales, así como la parte proporcional de imposición fiscal, las cargas laborales y otros gastos repercutibles.

En caso de tener que realizarse unidades de obra no previstas en el proyecto, se fijará su precio entre el Técnico Director y el Contratista antes de iniciar la obra y se presentará a la propiedad para su aceptación o no.

## 4. PRESUPUESTO

## INDICE DE PRESUPUESTO

|   |                               |     |
|---|-------------------------------|-----|
| 1 | MEDICIONES.....               | 171 |
| 2 | DESCOMPUESTOS.....            | 171 |
| 3 | RESUMEN DEL PRESUPUESTO ..... | 173 |

### 1 MEDICIONES

| CÓDIGO                            | RESUMEN    | CANTIDAD | PRECIO   | IMPORTE         |
|-----------------------------------|------------|----------|----------|-----------------|
| <b>CAPÍTULO 01 Hardware</b>       |            |          |          |                 |
| 01.1                              | u Servidor | 1,00     | 1.004,00 | 1.004,00        |
| 01.2                              | u Hogar    | 1,00     | 93,97    | 93,97           |
| <b>TOTAL CAPÍTULO 01 Hardware</b> |            |          |          | <b>1.097,97</b> |

|                                    |                         |      |          |                  |
|------------------------------------|-------------------------|------|----------|------------------|
| <b>CAPÍTULO 02 Diseño proyecto</b> |                         |      |          |                  |
| 02.1                               | h Servidor              | 1,00 | 3.560,00 | 3.560,00         |
| 02.2                               | h Hogar                 | 1,00 | 4.600,00 | 4.600,00         |
| 02.3                               | h Aplicación<br>Android | 1,00 | 3.840,00 | 3.840,00         |
| <b>TOTAL CAPÍTULO 01 Hardware</b>  |                         |      |          | <b>12.000,00</b> |

|              |  |  |  |                  |
|--------------|--|--|--|------------------|
| <b>TOTAL</b> |  |  |  | <b>13.097,97</b> |
|--------------|--|--|--|------------------|

### 2 DESCOMPUESTOS

| CÓDIGO  | CANTIDAD.UD | RESUMEN                     | PRECIO | SUBTOTAL | IMPORTE         |
|---|-------------|-----------------------------|--------|----------|-----------------|
| <b>CAPÍTULO 01 Hardware</b>   |             |                             |        |          |                 |
| 01.1  | u           | Servidor                    |        |          |                 |
| P34   | 1,000       | u Servidor de<br>aplicación | 999,00 | 999,00   | 999,00          |
| P35   | 5,000       | m Cable CAT-6 1,00          | 5,00   |          |                 |
| <b>TOTAL PARTIDA</b>  |             |                             |        |          | <b>1.004,00</b> |
| Asciende el precio total de la partida a la mencionada cantidad de MIL CUATRO EUROS |             |                             |        |          |                 |

|      |       |                                       |       |       |       |
|------|-------|---------------------------------------|-------|-------|-------|
| 01.2 | u     | Hogar                                 |       |       |       |
| P36  | 1,000 | u Raspberry Pi 3 ModeloB              | 42,99 | 42,99 | 42,99 |
| P37  | 5,000 | m Cable CAT-6 2,00                    | 10,00 |       |       |
| P38  | 1,000 | u Tarjeta de memoria<br>micro sd 16GB | 9,99  | 9,99  | 9,99  |

|     |       |   |       |       |
|-----|-------|---|-------|-------|
| P39 | 1,000 | u Cámara USB                              | 12,99 | 12,99 |
| P40 | 1,000 | u Micrófono                               | 1,00  | 1,00  |
| P41 | 1,000 | u Convertidor USB a Jack 3.5mm            | 2,00  | 2,00  |
| P42 | 1,000 | u Caja estanca                            | 10,00 | 10,00 |
| P43 | 1,000 | u Modulo sensor de temperatura DHT11      | 2,00  | 2,00  |
| P44 | 1,000 | u Diodo led emisor infrarrojo IR333C 1,00 | 1,00  |       |
| P45 | 1,000 | u Sensor receptor infrarrojo TSOP38238    | 2,00  | 2,00  |

**TOTAL PARTIDA** **93,97**

Asciende el precio total de la partida a la mencionada cantidad de NOVENTA Y TRES EUROS con NOVENTA Y SIETE CÉNTIMOS

### CAPÍTULO 02 Diseño proyecto

| <b>02.1</b> | <b>h</b> | <b>Servidor</b>                |       |          |
|-------------|----------|--------------------------------|-------|----------|
| P46         | 76,000   | h Desarrollo servidor          | 40,00 | 3.040,00 |
| P47         | 10,000   | h Diseño base de datos         | 40,00 | 400,00   |
| P48         | 3,000    | h Implementación base de datos | 40,00 | 120,00   |

**TOTAL PARTIDA** **3.560,00**

Asciende el precio total de la partida a la mencionada cantidad de TRES MIL QUINIENTOS SESENTA EUROS

| <b>02.2</b> | <b>h</b> | <b>Hogar</b>                               |        |          |
|-------------|----------|--|--------|----------|
| P49         | 40,000   | h Desarrollo comunicaciones Raspberry Pi   | 40,00  | 1.600,00 |
| P50         | 21,000   | h Desarrollo traducción voz a texto        | 40,00  | 840,00   |
| P51         | 18,000   | h Desarrollo comunicación con KNX 40,00    | 720,00 |          |
| P52         | 21,000   | h Desarrollo control elementos infrarrojos | 40,00  | 840,00   |
| P53         | 15,000   | h Unión de módulos diseñados               | 40,00  | 600,00   |

**TOTAL PARTIDA** **4.600,00**

Asciende el precio total de la partida a la mencionada cantidad de CUATRO MIL SEISCIENTOS EUROS

| <b>02.3</b> | <b>h</b> | <b>Aplicación Android</b> |       |          |
|-------------|----------|---------------------------|-------|----------|
| P54         | 96,000   | h Desarrollo aplicación   | 40,00 | 3.840,00 |

**TOTAL PARTIDA** **3.840,00**

Asciende el precio total de la partida a la mencionada cantidad de TRES MIL OCHOCIENTOS CUARENTA EUROS

### 3 RESUMEN DEL PRESUPUESTO

| <b>CAPITULO</b>   | <b>RESUMEN</b>  | <b>EUROS</b>     | <b>%</b> |
|---|-----------------|------------------|----------|
| 01  | Hardware        | 1.097,97         | 91,62    |
| 02  | Diseño proyecto | 12.000,00        | 91,62    |
| <b>TOTAL EJECUCIÓN MATERIAL</b>   |                 | <b>13.097,97</b> |          |
| 13,00 % Gastos generales  |                 | 1.702,74         |          |
| 6,00 % Beneficio industrial   |                 | 785,88           |          |
| SUMA DE G.G. y B.I.   |                 | 2.488,62         |          |
| 21,00 % I.V.A.  |                 | 3.273,18         |          |
| <b>TOTAL PRESUPUESTO GENERAL</b>  |                 | <b>18.859,77</b> |          |
| Asciende el presupuesto general a la expresada cantidad de DIECIOCHO MIL OCHOCIENTOS CINCUENTA Y NUEVE EUROS con SETENTA Y SIETE CÉNTIMOS, a 01 de febrero de 2017. |                 |                  |          |

## ANEXOS

# MANUAL DE INSTALACIÓN

## SERVIDOR

El servidor es un proyecto Java. Para ejecutarlo hay que abrir el archivo (.jar) ejecutable o importarlo en algún IDE y ejecutarlo.

### PAQUETES

- bdd. Paquete que contiene la conexión con la base de datos y las operaciones necesarias.
- security. En el paquete están las clases referidas a la seguridad de comunicación.
- server. Paquete en el que se encuentran las clases más importantes.
- typemessage. Contiene las clases con los modelos de los tipos de mensajes.

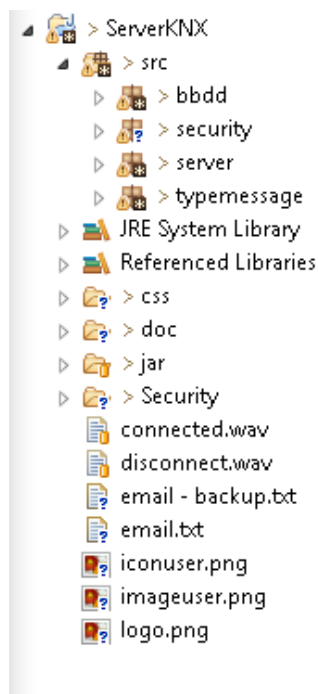


Figura 1

En el paquete server existe una clase llamada “Constant”, en esta están contenidas las constantes para la ejecución del programa. El servidor escucha conexiones en el puerto 443 esto es modificable en la clase anteriormente mencionada. Para que funcione correctamente habría que redirigir el puerto del router. La aplicación Android y el Raspberry Pi se conectan al puerto. Si esto no es así sólo funcionaría en la misma red.

### CARPETA SECURITY

Dentro del proyecto hay una carpeta llamada Security, se encuentra el certificado digital que envía el servidor y la clave privada de RSA que utiliza para descifrar.

### CARPETA JAR

También hay una carpeta llamada “jar” en la cual se encuentran todos los archivos (.jar) de los cuales hace uso el servidor.

## BASE DE DATOS

Para poder acceder a la base de datos desde Workbench los credenciales no han sido modificados, estos siguen siendo los que trae por defecto:

- Usuario: root.
- Contraseña: 123456.

## RASPBERRY PI

Para realizar una conexión a través de SSH, SFTP o cualquier otro las credenciales del Raspberry Pi son:

- Usuario: pi / root.
- Contraseña: 1003

La estructura del proyecto en el dispositivo Raspberry Pi puede verse con la siguiente figura.

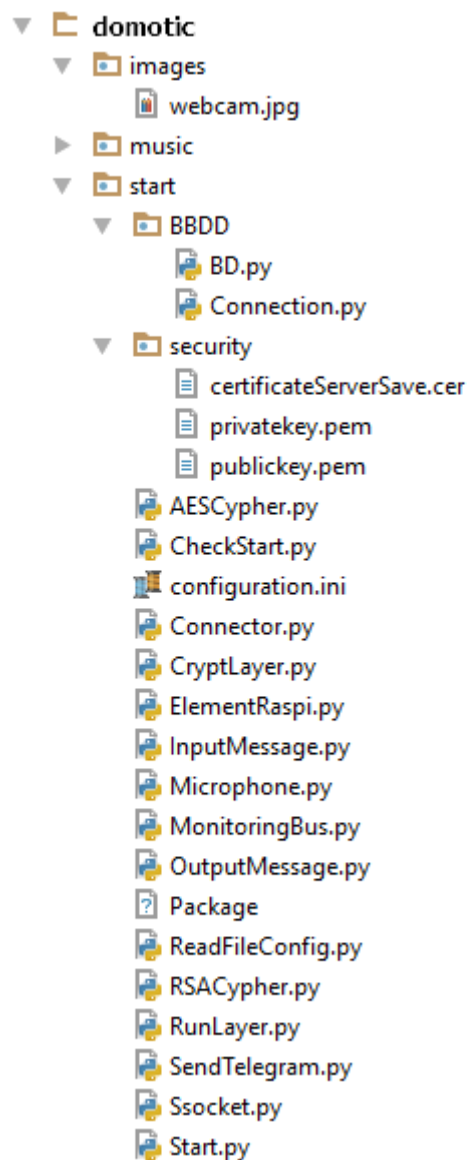


Figura 2

Se puede observar en la figura 2 el fichero configuration.ini. Es un fichero de configuración que almacena los siguientes campos: id del Raspberry Pi, contraseña, dirección del servidor, puerto al que ha de conectarse, dirección del servidor de imágenes y puerto.

En la carpeta images son guardadas las imágenes que el usuario a través de la aplicación Android solicita.

### CARPETA BD

En esta se encuentran las clases que realizan operaciones que existe en el Raspberry Pi.

Para poder acceder a la base de datos por consola:

```
mysql -u homedomotic -p
contraseña: 123456
use domotic;
select * from user_raspi;
select * from devices;
```

Con los anteriores comandos puede verse la estructura de la base de datos, el cual es “domotic” y usuario “homedomotic”.

### CARPETA SECURITY

En esta se almacena lo referente a la seguridad de la comunicación. En ella se encuentra almacenado.

- Certificado del servidor en un archivo (.cer).
- Clave privada RSA 2048 generada. Esta almacenada en un archivo (.pem).
- Clave pública RSA 2048 generada. Esta almacenada en un archivo (.pem).

### CARPETA LIRC

Esta carpeta no se encuentra en la estructura anterior. En esta carpeta es almacenado un archivo de configuración el cual contiene los pulsos de los elementos infrarrojos guardados por el usuario.

- Ruta del archivo: /etc/lirc/
- Nombre del archivo: lircd.conf

Decir que el archivo tiene una estructura bien definida. Si esta se altera no se ejecutará las ordenes de envío de pulsos infrarrojos.

### DEMONIO

El demonio creado y ejecutado en el inicio del sistema operativo ha sido llamado “domoKNX.service”, su alias identificador es “domotic daemon”.

Ruta: /etc/systemd/System/

Algunos comandos útiles:

- Inicialo: `sudo systemctl start domoknx.service`
- Pararlo: `sudo systemctl stop domoknx.service`
- Reinicialo: `sudo systemctl restart domoknx.service`
- Saber su estado: `sudo systemctl status domoknx.service -l`

# GUÍA DE USUARIO



# ÍNDICE

---

1. REGISTRO
2. LOGIN
3. CONFIGURACIÓN
4. CONFORT
  - 4.1. LUCES
  - 4.2. PERSINANAS
  - 4.3. ELECTRODOMÉSTICOS
  - 4.4. ESCENAS
5. CLIMATIZACIÓN
  - 5.1. CALEFACCIÓN
  - 5.2. AIRE ACONDICIONADO
6. SEGURIDAD
  - 6.1. IMÁGENES/FOTOS
  - 6.2. MODO VACACIONES
  - 6.3. HISTORIAL DE ALARMAS
7. MULTIMEDIA

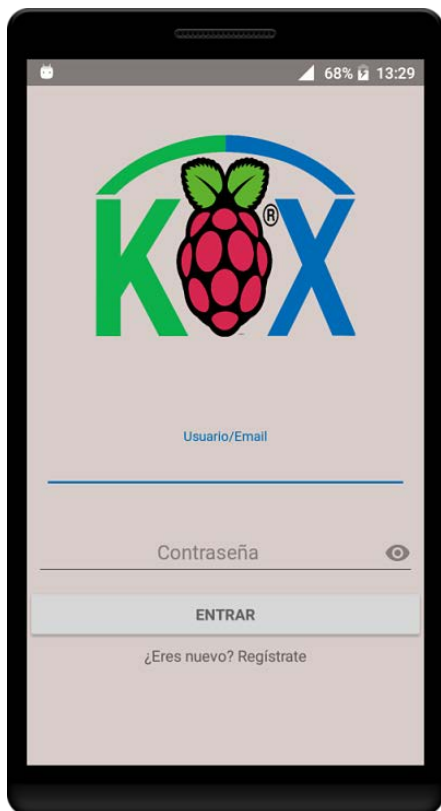


Ilustración 1

Cuando el usuario abre la aplicación por primera vez, observará la siguiente pantalla, tal y como refleja la Ilustración 1. El usuario tendrá que dirigirse al botón “¿Eres nuevo? Regístrate” para poder comenzar a utilizar la aplicación.



Ilustración 2

Este botón nos dirigirá a la pantalla de registro tal y como se puede ver en la Ilustración 2. El nuevo usuario tendrá que proporcionar una serie de datos tales como el correo, un nombre de usuario y una contraseña que tendrá que repetir para verificar que esta última es correcta.

## REGISTRO

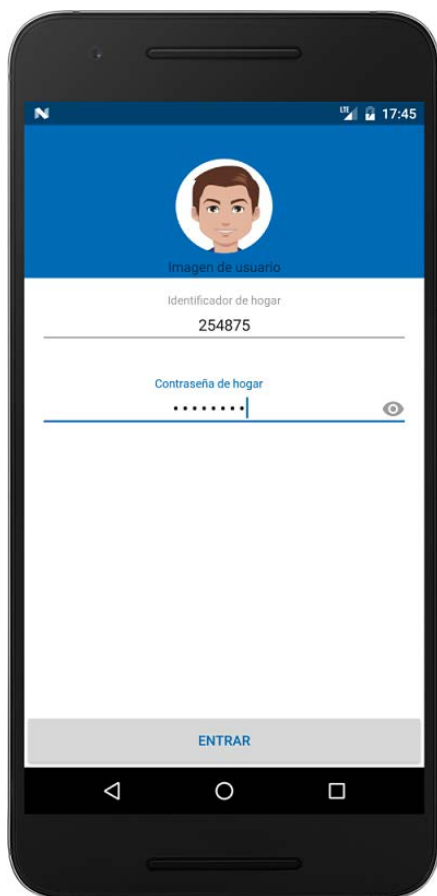


Ilustración 3

Una vez que el usuario se ha registrado con sus datos personales, la aplicación le dirigirá al siguiente formulario tal y como vemos en la Ilustración 3. Este formulario le requerirá un identificador de hogar con su respectiva contraseña. El identificador será proporcionado por el instalador del servicio.

También se observa que puede establecerse una imagen de usuario pulsando sobre la imagen.

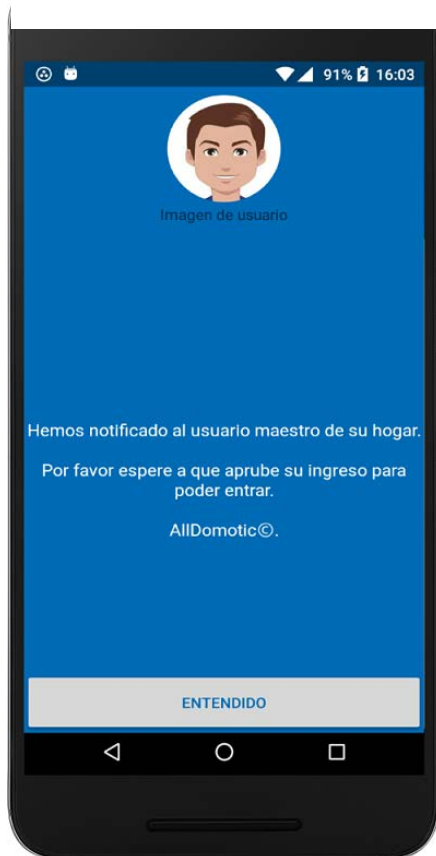


Ilustración 4

Una vez que el usuario ha proporcionado los datos requeridos se dirigirá a la siguiente pantalla (Ilustración 4) que indica que se ha notificado al propietario del hogar y que debe esperar a que su ingreso sea aprobado para poder comenzar a hacer uso de la aplicación.

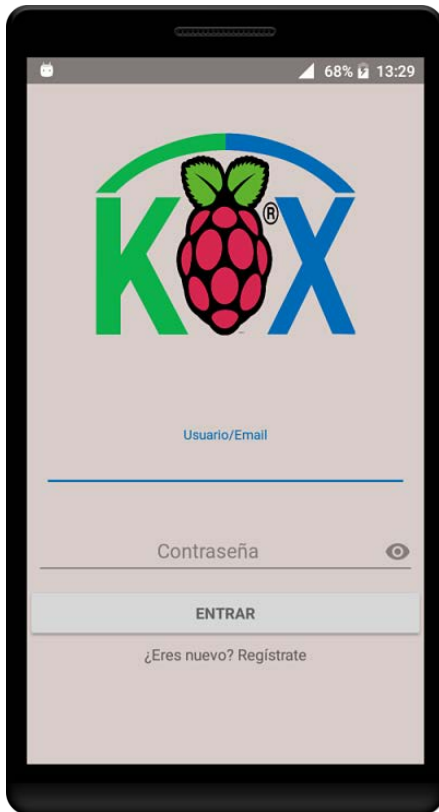


Ilustración 5

Una vez que el usuario ha realizado el proceso de registro, será devuelto a la pantalla inicial de la aplicación, teniendo que introducir el usuario/email junto con la contraseña y pulsar sobre el botón “ENTRAR”. Véase Ilustración 5.



Ilustración 6

Una vez que se ha pulsado el botón “ENTRAR” se mostrará el progreso de la conexión con su hogar como se ve en la Ilustración 6.



Ilustración 7

Realizado todo el proceso de registro anterior, cuando el usuario acceda finalmente a la aplicación podrá ver esta pantalla (Ilustración 7). En ella se pueden apreciar los distintos módulos con los que la aplicación nos permite trabajar (confort, climatización, seguridad y multimedia). A la vez que un estado resumido de la situación de las luces de la casa y la temperatura.



Esta pantalla presenta un indicador de la conexión actual de su hogar y su dispositivo móvil en la parte superior derecha, véase Ilustración 8. Si se pulsa el usuario podrá acceder a la leyenda.

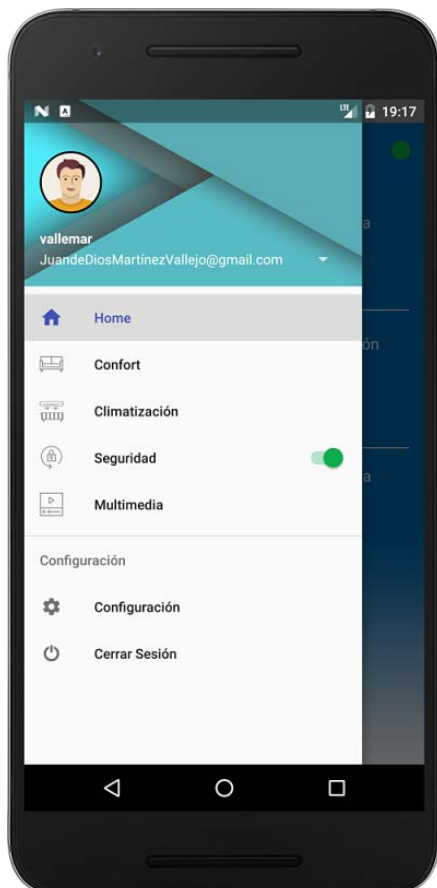


Ilustración 9

Se puede desplegar el menú lateral que estará presente en toda la aplicación desde la parte superior izquierda de la pantalla. El acceso a la configuración y la opción de Cerrar Sesión solo estarán activos si se entra a este menú desde la pantalla de inicio. En cambio se podrá acceder a los distintos módulos desde cualquier pantalla de la aplicación. Véase Ilustración 9.

Si se observa el menú lateral, en el apartado de seguridad se puede apreciar un botón de color verde. Desde él se activa y desactiva el modo de seguridad de la casa. Ante un incidente en el hogar los usuarios serán notificados si éste se encuentra activo.



Ilustración 10

En la parte inferior de la pantalla de inicio se encuentra el botón de “ACCIONES MAS USADAS”. Este botón es un panel de acceso rápido. Si el usuario lo pulsa durante el primer uso de la aplicación se puede observar como en la Ilustración 10 que aún no se ha realizado ninguna acción.



En esta pantalla se puede apreciar que el panel de inicio rápido se ha generado con las acciones más usadas por el usuario. Ya que este ya ha hecho uso de la aplicación en más ocasiones. Véase Ilustración 11.

Ilustración 11

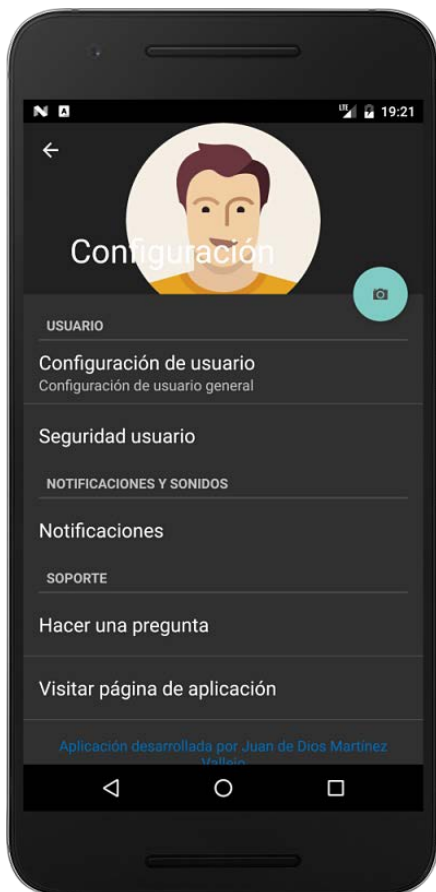


Ilustración 12

Cuando el usuario pulsa sobre la pestaña de configuración se despliega un menú donde es posible realizar cambios en los datos registrados. También permite el acceso a la Seguridad del usuario y las notificaciones tal y como se aprecia en la Ilustración 12.

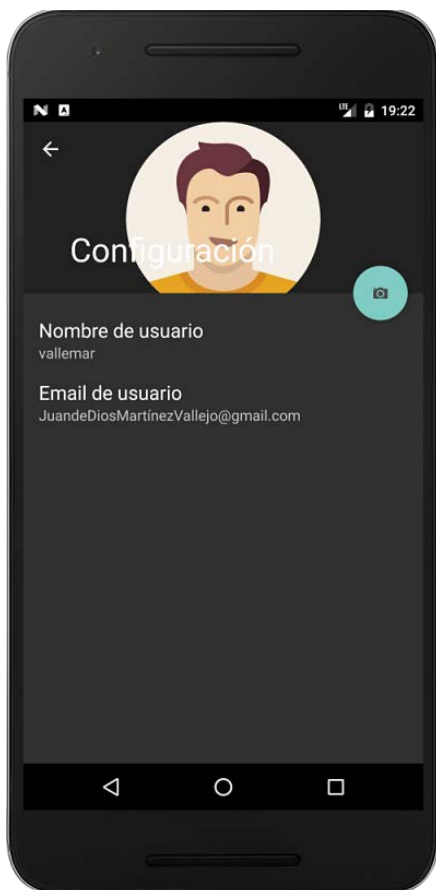


Ilustración 13

Si se pulsa sobre la pestaña de Configuración de usuario, el cliente tiene la posibilidad de visualizar su nombre de usuario y el email proporcionado en el momento del registro. Véase Ilustración 13.

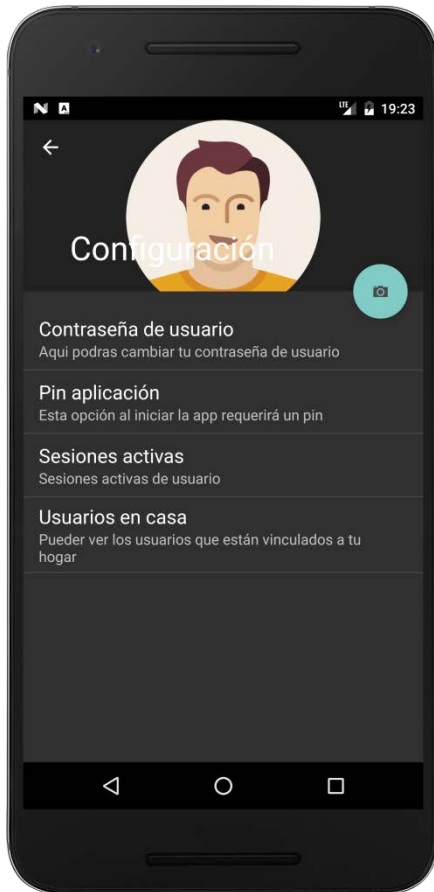


Ilustración 14

Si se pulsa la pestaña de Seguridad usuario, en esta pantalla aparecen las opciones referentes a la seguridad de la aplicación donde se permite al usuario realizar una serie de cambios y tal y como se ve en la Ilustración 14.

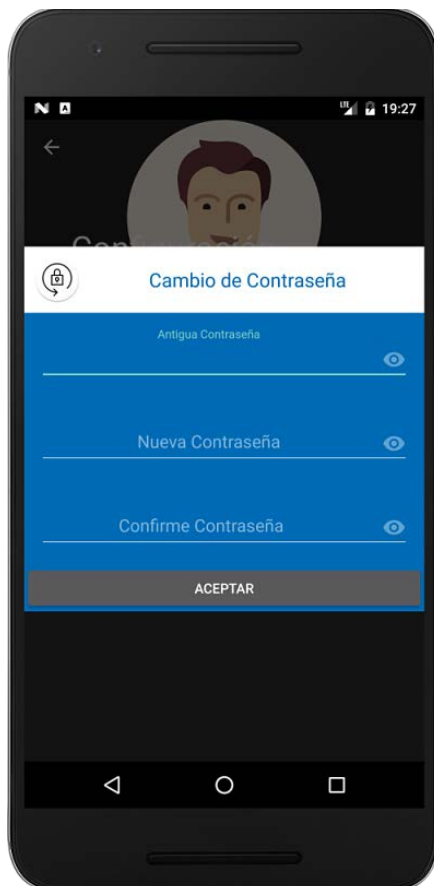


Ilustración 15

Esta pantalla permite la acción de cambio de contraseña por parte del usuario. Tal y como se ve en la Ilustración 15 se requiere la contraseña antigua y una nueva que se pedirá confirmar para verificar que esta es correcta.

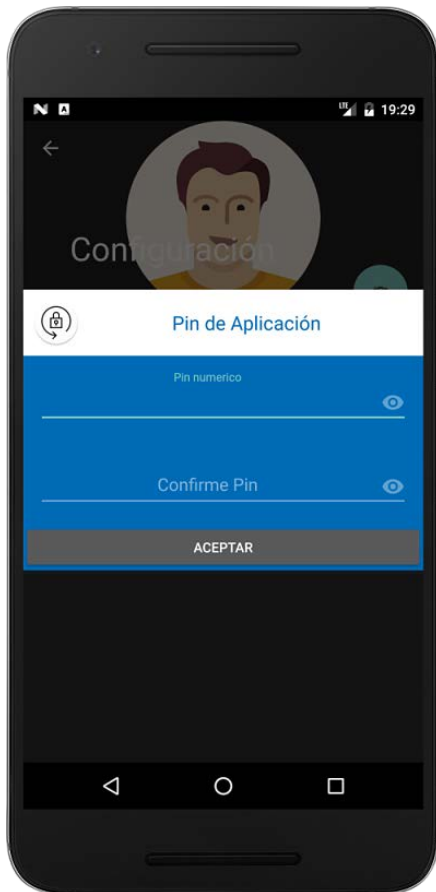


Ilustración 16

En esta pantalla se muestra la opción de introducir un pin de bloqueo cada vez que se acceda a la aplicación, tal y como está representado en la Ilustración 16.

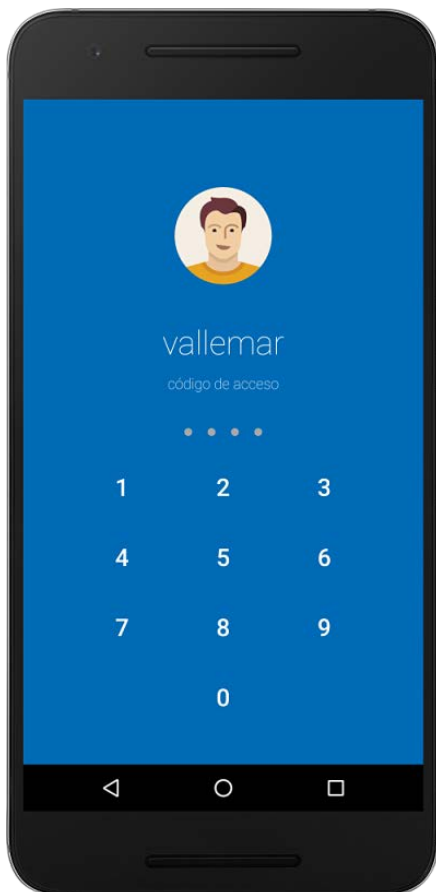


Ilustración 17

En la Ilustración 17 se puede apreciar la pantalla para introducir el código de acceso que la aplicación requerirá si se activa esta opción.

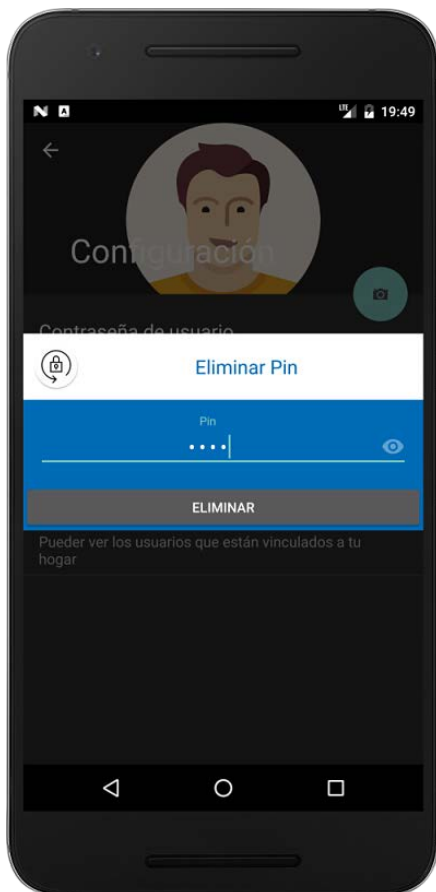


Ilustración 18

Esta pantalla ofrece al usuario la opción de eliminar el código de acceso a la aplicación en el momento en el que así lo desee, tal y como se aprecia en la Ilustración 18.

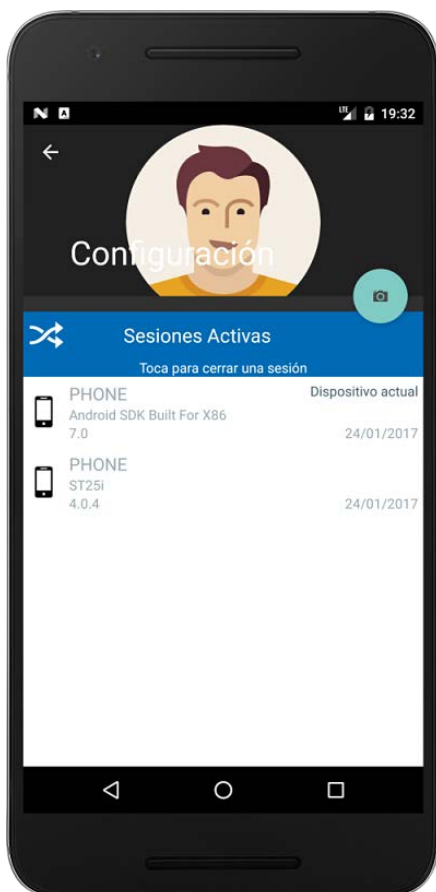


Ilustración 19

Esta pestaña permite visualizar las sesiones que el usuario mantiene activas desde distintos dispositivos. Esta misma pantalla ofrece la opción de cerrar la sesión que se señale tal y como muestra la Ilustración 19.

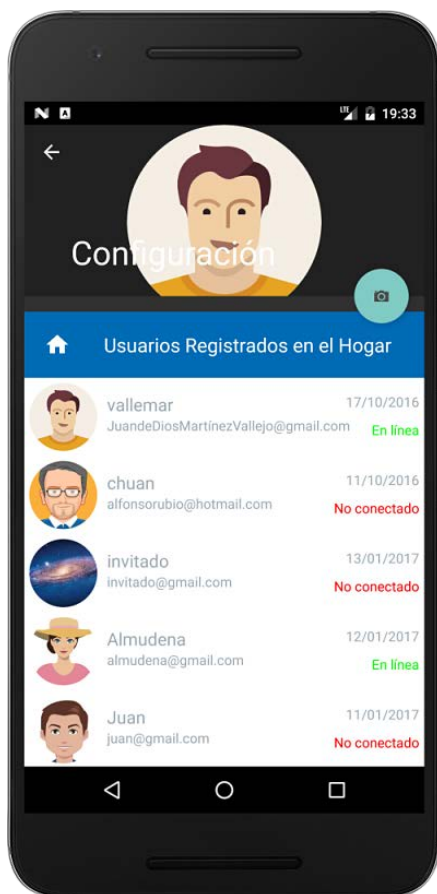


Ilustración 20

Esta pantalla ofrece la posibilidad de ver todos los usuarios que hay vinculados a un mismo hogar y si estos se encuentran en línea o no conectados en un momento determinado. Véase Ilustración 20.

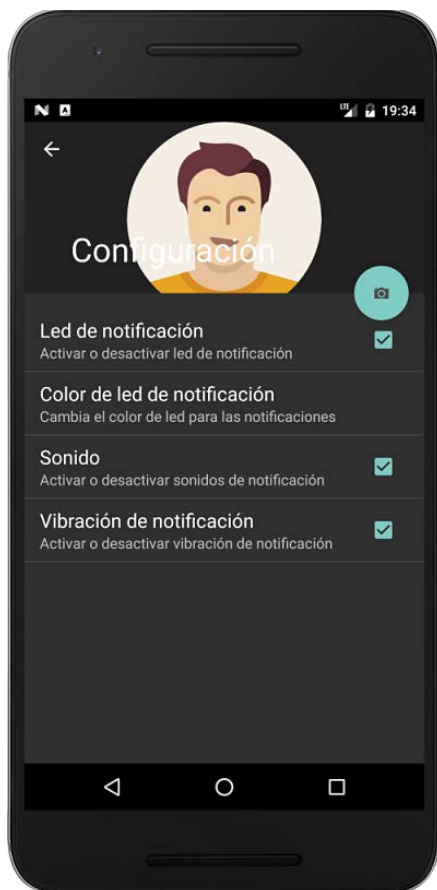


Ilustración 21

Con esta última pantalla se permite al usuario activar una serie de opciones tales como el Led de notificación, el sonido o la vibración cuando se produce una notificación de la aplicación en el dispositivo. También permite la opción de modificar el color de led de notificación tal y como se ve en la Ilustración 21.



Ilustración 22

Si el usuario accede al módulo de Confort se dirigirá a la siguiente pantalla donde se pueden observar las diferentes pestañas que éste ofrece. Pulsando estas pestañas, el usuario podrá configurar cada una de las opciones como lo desee. Véase Ilustración 22.

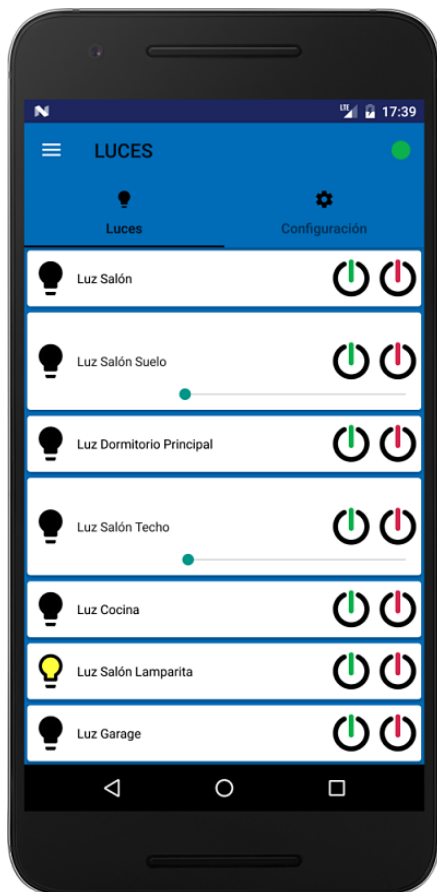


Ilustración 23

Si el usuario accede al módulo de Luces se dirigirá a la siguiente pantalla donde se puede observar el estado de las diferentes luces del hogar. Desde esta pantalla se ofrece al cliente la posibilidad de encender o apagar cada una de las luces de forma individualizada. Véase Ilustración 23.

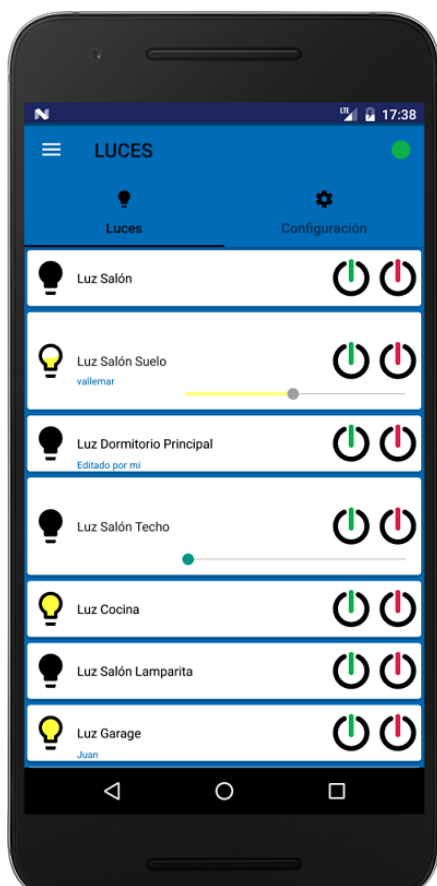


Ilustración 24

Si se produce algún cambio en el encendido o apagado de las luces, éste será notificado en la propia pantalla. A la vez que se informará del autor de dicho cambio que puede ser cualquiera de los usuarios del hogar.

Estos cambios pueden llevarse a cabo a través del micrófono, desde la propia casa o por alguna programación de escena. Véase Ilustración 24.

## CONFORT: LUCES



Ilustración 25

En esta pantalla se puede apreciar cómo al acceder a la pestaña de Configuración, el usuario se encontrará con las opciones de programación de las luces del hogar. Asimismo tendrá la opción de encender o apagar todas las luces a la vez tal y como se observa en la Ilustración 25.

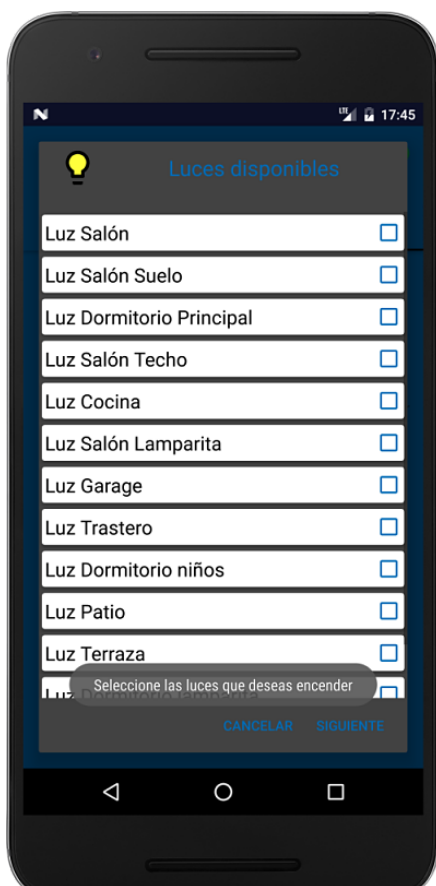


Ilustración 26

Al pulsar sobre las pestañas de Programación encendido/apagado, el usuario accederá a la siguiente pantalla donde podrá ver un resumen de todas las luces disponibles del hogar, pudiendo seleccionar cada una de ellas de forma individualizada. Véase Ilustración 26.



Ilustración 27

Una de vez que han sido seleccionadas las luces, el usuario se dirigirá a la siguiente pantalla donde podrá seleccionar a su vez qué días de la semana desea que sea activada la escena de luces que está siendo creada tal y como se ve en la Ilustración 27.

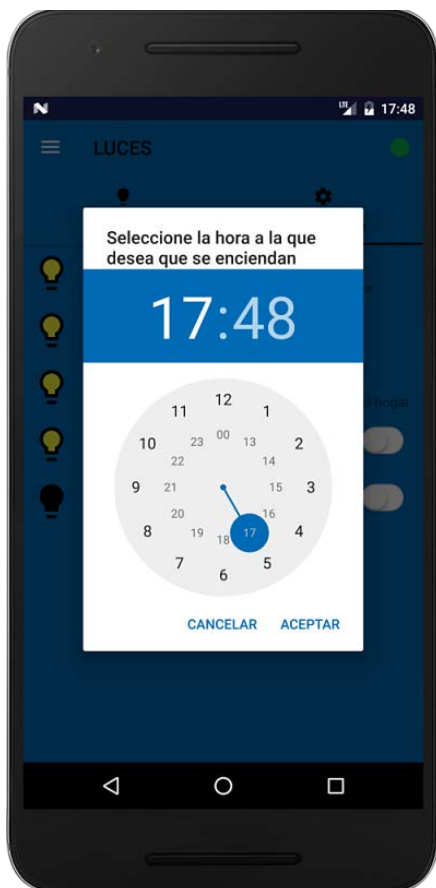


Ilustración 28

Una vez que han sido seleccionados los días de la semana, el usuario accederá a la siguiente pantalla. En esta se establece la hora a la que el usuario desee la activación de dicha escena de luces. Véase Ilustración 28.

En último lugar, el usuario será redirigido a esta pantalla donde podrá introducir un nombre a la escena creada para su posterior identificación tal como se aprecia en la Ilustración 29.

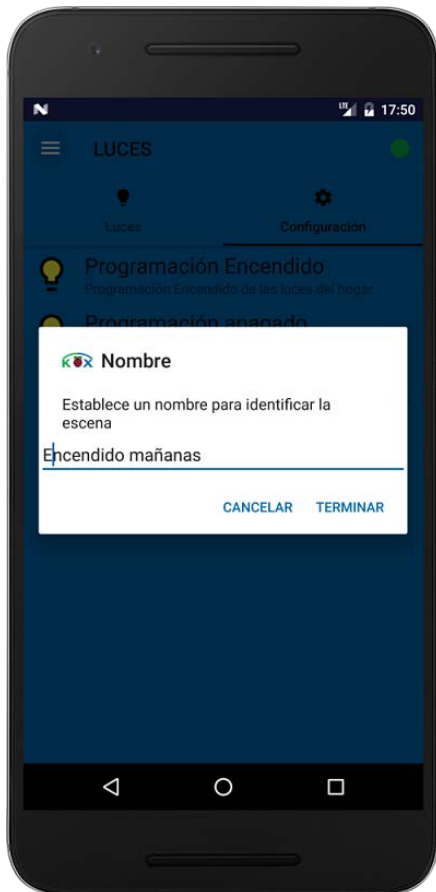


Ilustración 29

En esta pantalla el usuario podrá ver un resumen de todas las escenas de luces creadas. El primer botón ofrece la posibilidad de activarlas o desactivarlas en los horarios prefijados en el momento en que las escenas fueron creadas. Así como activarlas en cualquier momento a pesar de no ser la hora que se había establecido, en el segundo botón. Véase Ilustración 30.



Ilustración 30

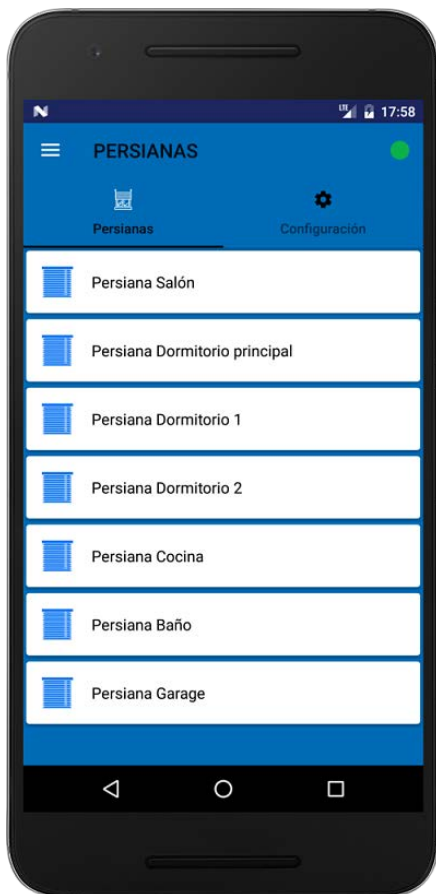


Ilustración 31

Si el usuario accede al módulo de Persianas se dirigirá a la siguiente pantalla donde se puede observar el estado de las diferentes persianas del hogar. Desde esta pantalla se ofrece al cliente la posibilidad de subir o bajar cada una de las persianas de forma individualizada. Véase Ilustración 31.



Ilustración 32

Si se pulsa sobre cualquiera de las persianas del hogar, el usuario se dirigirá a esta pantalla donde podrá subir o bajar cada una tanto como lo desee. Véase Ilustración 32.

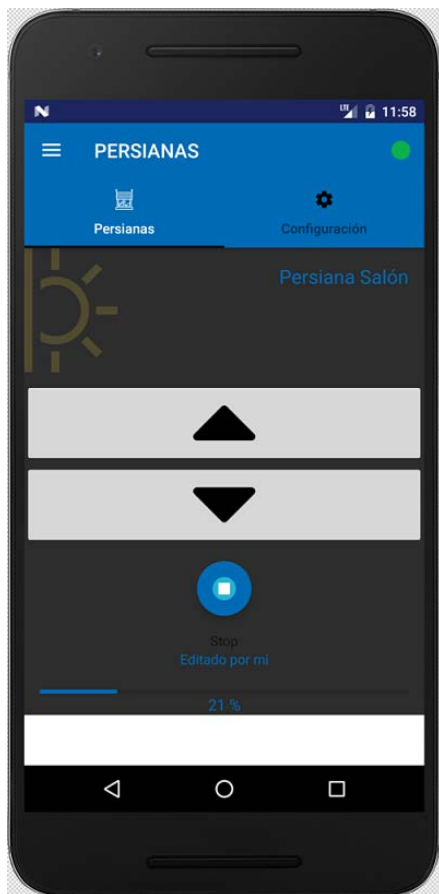


Ilustración 33

Cuando el usuario pulse sobre los botones de subida o bajada, aparecerá en la pantalla el progreso de esta acción. Cuando dicha acción finalice el cliente podrá observar como ésta se ha completado. Véase Ilustración 33.



Ilustración 34

En esta pantalla se puede apreciar cómo al acceder a la pestaña de Configuración, el usuario se encontrará con las opciones de programación de las persianas del hogar. Asimismo, tendrá la opción de subir o bajar todas las persianas a la vez tal y como se observa en la Ilustración 34. Las opciones de configuración son las mismas que las que el usuario encuentra en el módulo de luces.

## CONFORT: ELECTRODOMÉSTICOS

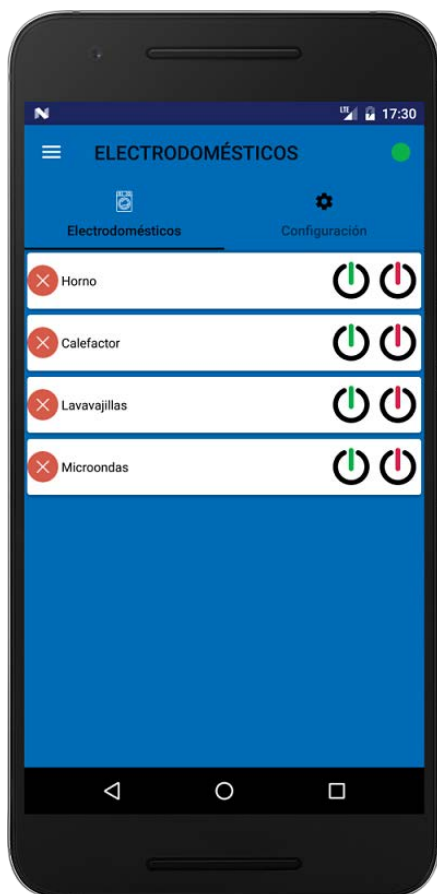


Ilustración 35

Si el usuario accede al módulo de Electrodomésticos se dirigirá a la siguiente pantalla donde se puede observar el estado los diferentes electrodomésticos del hogar. Desde esta pantalla se ofrece al cliente la posibilidad de encender o apagar cada uno de los electrodomésticos de forma individualizada. Véase Ilustración 35.

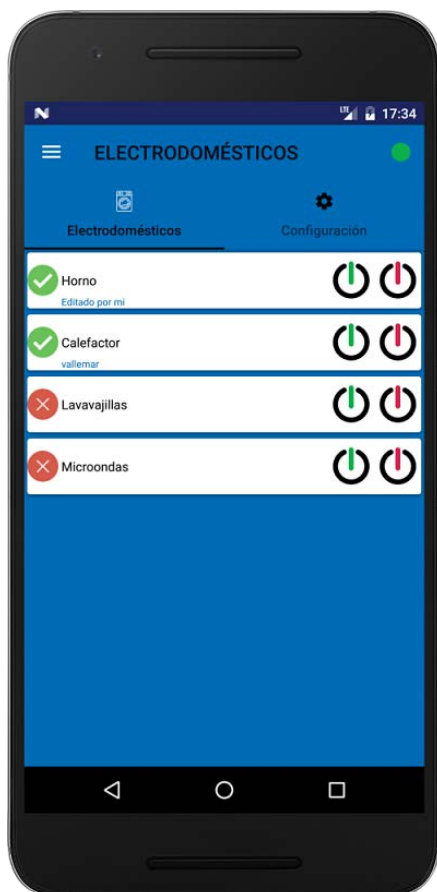


Ilustración 36

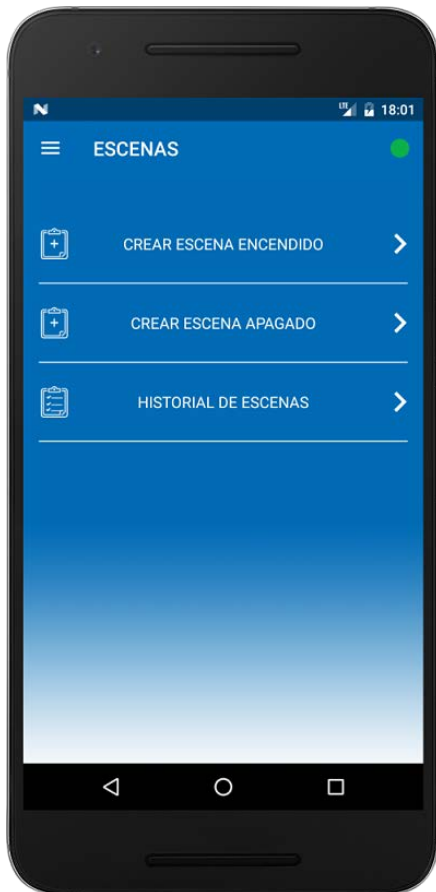
Si se produce algún cambio en el encendido o apagado de los electrodomésticos, éste será notificado en la propia pantalla. A la vez que se informará del autor de dicho cambio que puede ser cualquiera de los usuarios del hogar.

Estos cambios pueden llevarse a cabo a través del micrófono, desde la propia casa o por alguna programación de escena. Véase Ilustración 36.



Ilustración 37

En esta pantalla se puede apreciar cómo al acceder a la pestaña de Configuración, el usuario se encontrará con las opciones de programación de los electrodomésticos del hogar. Asimismo tendrá la opción de encender o apagar todos los electrodomésticos a la vez tal y como se observa en la Ilustración 37. Las opciones de configuración son las mismas que las que el usuario encuentra en los módulos de luces y persianas.



*Ilustración 38*

Si el usuario accede al módulo de Escenas se dirigirá a la siguiente pantalla donde se ofrece la posibilidad de crear escenas de la totalidad de dispositivos del hogar. De esta forma no habrá que crear escenas de forma individualizada.

También ofrece la posibilidad de acceder a un historial de todas las escenas creadas por los distintos usuarios del hogar. Véase Ilustración 38.



Ilustración 39

Si el usuario accede al módulo de Climatización se dirigirá a la siguiente pantalla donde se pueden observar las diferentes pestañas que éste ofrece. Pulsando sobre estas pestañas, el usuario podrá configurar cada una de las opciones como lo desee. Véase Ilustración 39.

## CLIMATIZACIÓN: CALEFACCIÓN/AIRE ACONDICIONADO

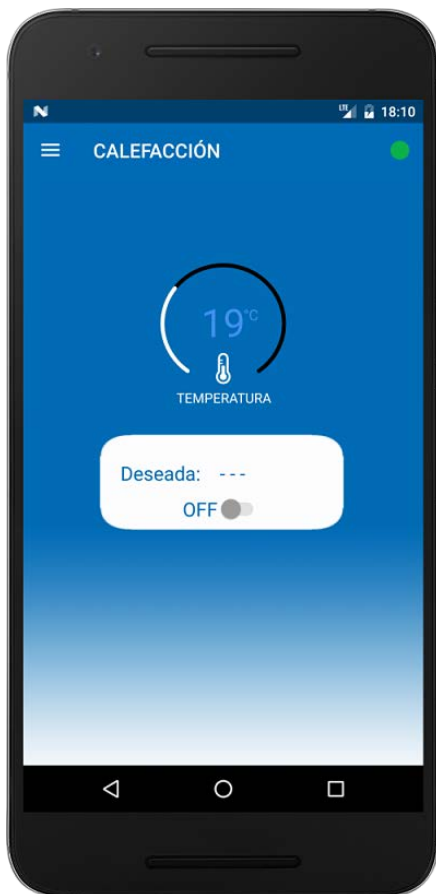


Ilustración 40

Si el usuario accede al módulo de Calefacción se dirigirá a la siguiente pantalla donde se puede observar la temperatura actual del hogar y un panel en el que podrá activar la calefacción a la temperatura deseada. Véase Ilustración 40.

Para esto el usuario tendrá que pulsar el botón de encendido/apagado.

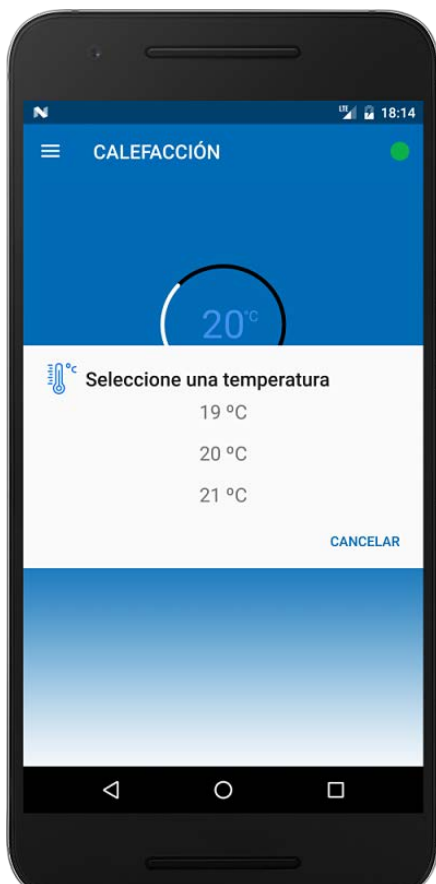


Ilustración 41

Al pulsar ese botón se desplegará un menú en el cual el usuario podrá elegir la temperatura que desee para su hogar. Simplemente deberá seleccionar el valor que desee. Véase Ilustración 41.

## CLIMATIZACIÓN: CALEFACCIÓN/AIRE ACONDICIONADO



Ilustración 42

Una vez que el usuario ha activado y seleccionado la temperatura deseada, aparecerá la Ilustración 42 en la pantalla con los nuevos valores seleccionados.

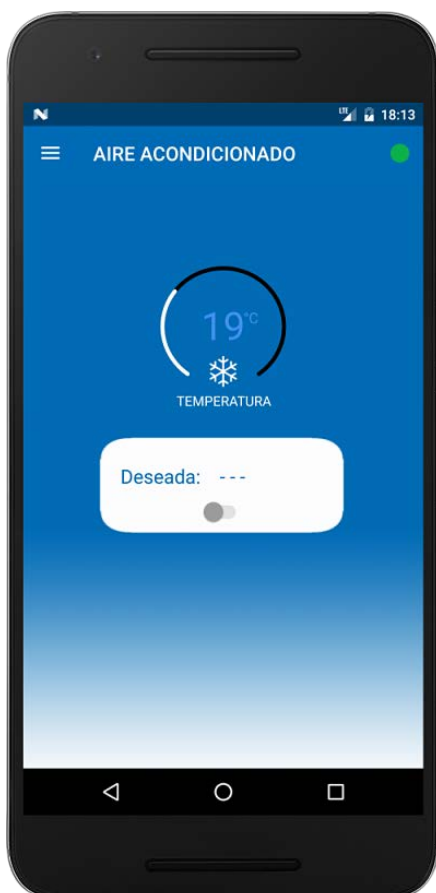


Ilustración 43

Si el usuario accede al módulo de Aire Acondicionado se dirigirá a la siguiente pantalla donde se puede observar la temperatura actual del hogar y un panel en el que podrá activar el aire acondicionado a la temperatura deseada. Véase Ilustración 43.

Para esto el usuario tendrá que pulsar el botón de encendido/apagado. Al pulsar ese botón se desplegará un menú en el cual el usuario podrá elegir la temperatura que desee para su hogar. Simplemente deberá seleccionar el valor deseado tal y como se ha explicado en el módulo de calefacción.



Ilustración 44

Si el usuario accede al módulo de Seguridad se dirigirá a la siguiente pantalla donde se pueden observar las diferentes pestañas que éste ofrece. Pulsando sobre las pestañas, el usuario podrá configurar cada una de las opciones como o desee. Véase Ilustración 44.

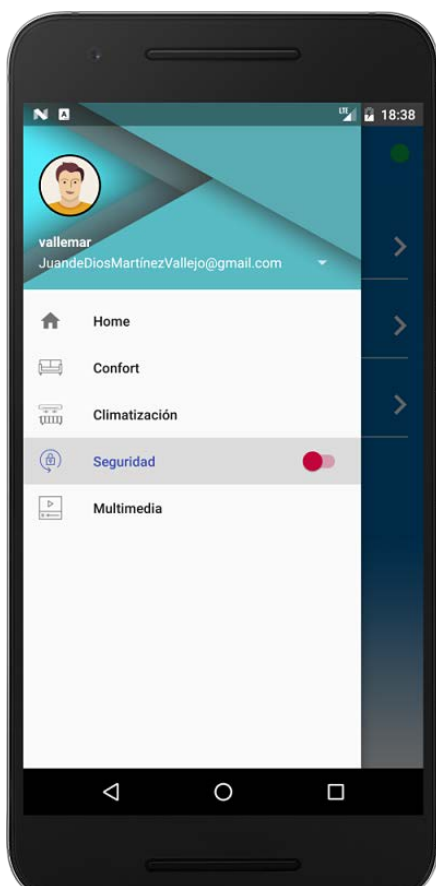


Ilustración 45

Si se observa el menú lateral, en el apartado de seguridad se puede apreciar un botón de color morado. Desde él se activa y desactiva el modo de seguridad de la casa. Ante un incidente en el hogar los usuarios serán notificados si éste se encuentra activo. Véase Ilustración 45.

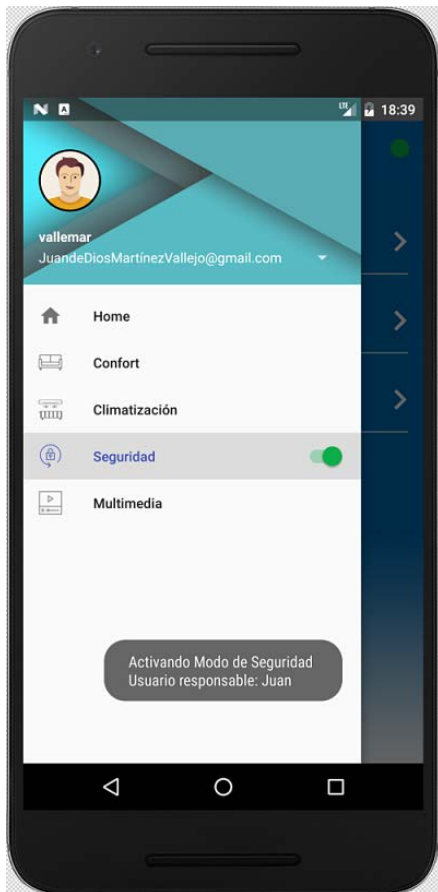


Ilustración 46

En esta pantalla se observa como el botón de seguridad se ha activado y pasa a ser verde. Una vez activado los usuarios siempre serán notificados ante cualquier incidente doméstico. Véase Ilustración 46.

## SEGURIDAD: IMÁGENES/FOTOS

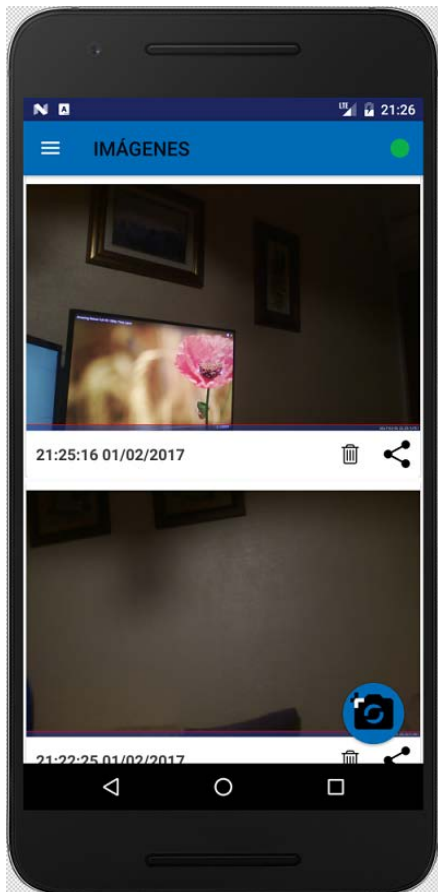


Ilustración 47

Si el usuario accede al Módulo de Imágenes/Fotos se dirigirá a la siguiente pantalla donde se ofrece la posibilidad de hacer una foto desde distintos ángulos de la casa y en cualquier momento que se desee. De esta forma los usuarios del hogar tendrán la tranquilidad de que no ha habido ningún incidente durante su ausencia, al mismo que tiempo que hacer un seguimiento del estado de ésta. Véase Ilustración 47.



Ilustración 48

Una vez que el usuario ha seleccionado la opción de hacer una foto, se desplegará el siguiente menú desde donde podrá elegir la cámara desde la que desea hacer la foto. Las diferentes cámaras se encuentran en distintos ángulos de la casa. Véase Ilustración 48.

## SEGURIDAD: MODO VACACIONES



Ilustración 49

Si el usuario accede al Módulo de Modo Vacaciones se dirigirá a la siguiente pantalla como se observa en la Ilustración 49. Esta opción permite al usuario programar las luces de su casa para simular que hay alguien en ella durante un periodo de ausencia.

## SEGURIDAD: HISTORIAL DE ALARMAS



Ilustración 51

Si el usuario accede al Historial de alarmas podrá visualizar la totalidad de las alarmas detectadas en el hogar desde el momento de la implantación del sistema domótico. Véase Ilustración 51.

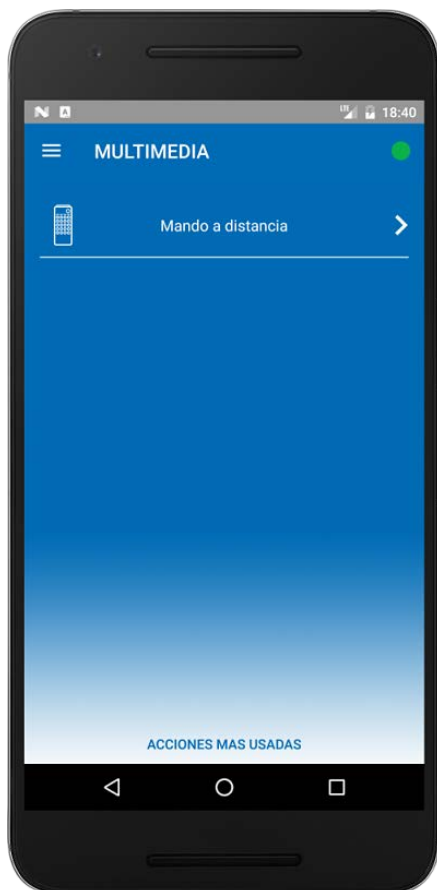


Ilustración 52

Si el usuario accede al módulo Multimedia se dirigirá a la siguiente pantalla donde se puede observar la opción que este módulo ofrece. En este caso es la de configurar los mandos a distancia de los diferentes dispositivos que cuenten con elementos infrarrojos tal y como se observa en la Ilustración 52.

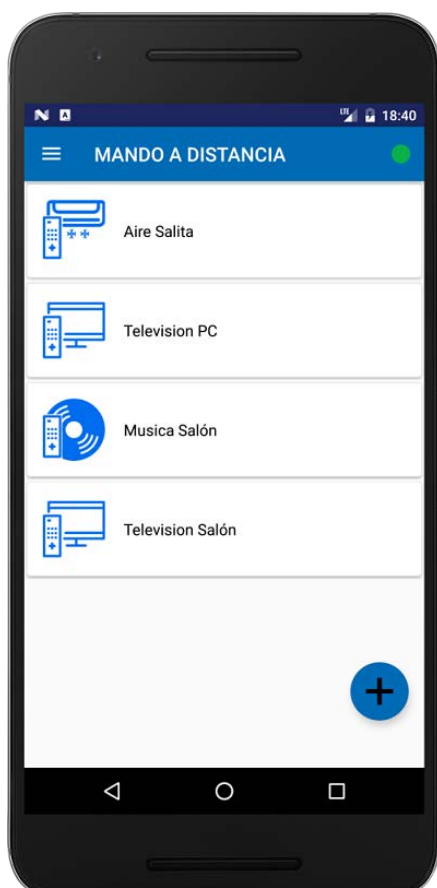


Ilustración 53

Una vez que se ha accedido a la pestaña de Mando a distancia, el usuario pasará a ver la siguiente pantalla. Desde aquí podrá seleccionar uno de los mandos que tenga guardados para su futura utilización.

Asimismo, mismo también se le presenta la opción de añadir otro tipo de mando a distancia pulsando el botón en la parte inferior derecha tal y como se percibe en la Ilustración 53.

.Si el usuario pulsa sobre uno de los mandos a distancia tipo de televisión previamente guardados, se visualizará esta pantalla para el control remoto de dicho mando como muestra la Ilustración 54.



Ilustración 54

Si el usuario pulsa sobre uno de los mandos a distancia tipo aire acondicionado previamente guardados, se visualizará esta pantalla para el control remoto de dicho mando como muestra la Ilustración 55.

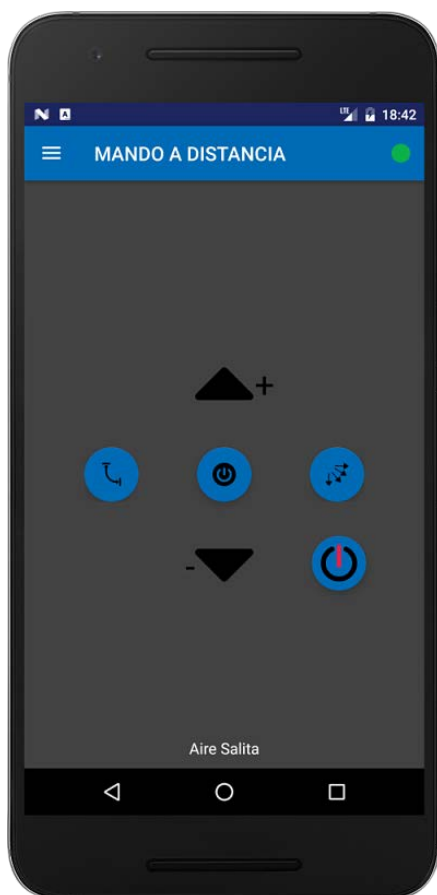


Ilustración 55

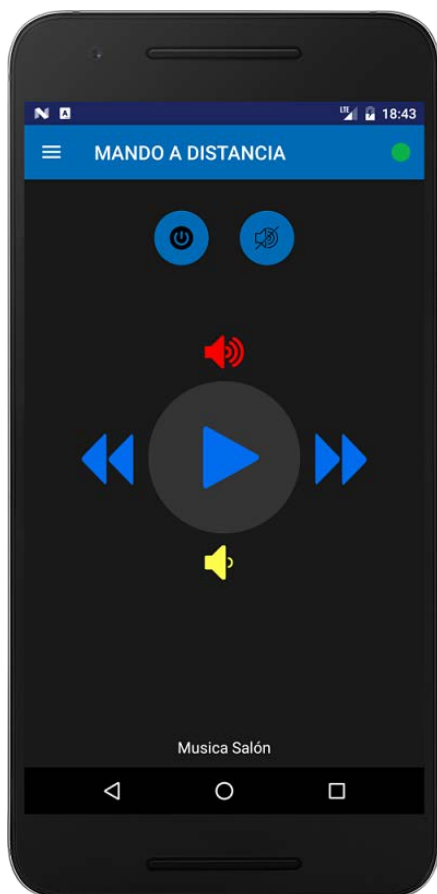


Ilustración 56

Si el usuario pulsa sobre uno de los mandos a distancia tipo reproductor de música, se visualizará esta pantalla para el control remoto de dicho mando como muestra la Ilustración 56.

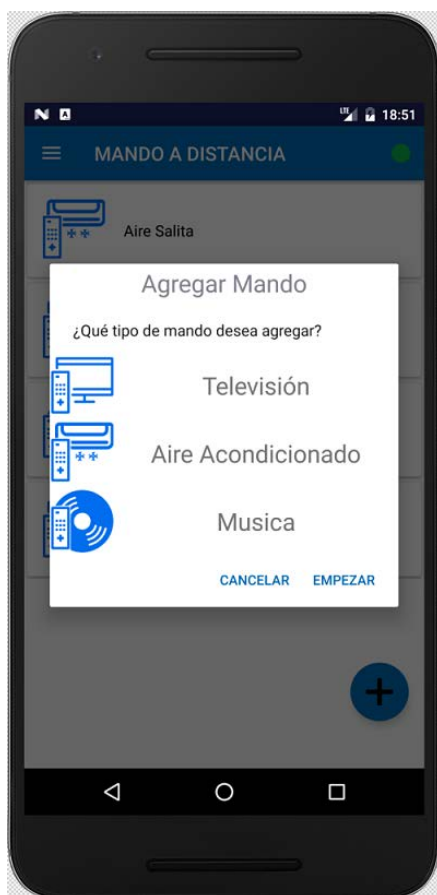


Ilustración 57

Al usuario se le ofrece la opción de añadir otro tipo de mando pinchando en el botón de la parte inferior derecha. Cuando lo pulse se desplegará un menú donde deberá seleccionar el tipo de mando que desea agregar al listado de los ya guardados previamente. Véase Ilustración 57.

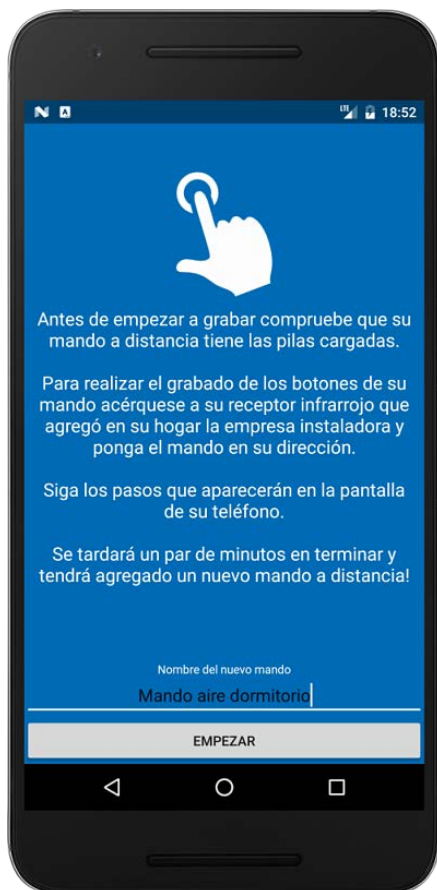


Ilustración 58

Para agregar un mando a distancia nuevo el usuario tendrá que seguir una serie de sencillos pasos que la propia aplicación le irá indicando.

Lo primero que aparecerá será una pantalla donde deberá elegir un nombre para el mando nuevo como se ve en la Ilustración 58.

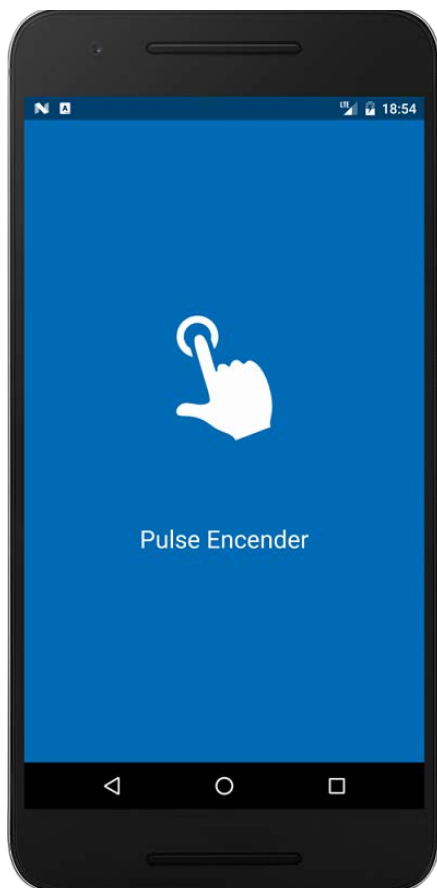


Ilustración 59

Una vez que el usuario ha elegido el nombre para su mando nuevo, será dirigido a una pantalla donde aparecerá el primer paso a seguir para la configuración del nuevo terminal. Véase Ilustración 59.

Tras esta primera pantalla la aplicación lo irá dirigiendo a otras con los diferentes pasos a seguir en los que simplemente tendrá que ir pulsando los botones indicados.



*Ilustración 60*

Cuando el usuario haya terminado todos los pasos a seguir se le dirigirá a una pantalla donde se le informará de que el proceso de configuración del nuevo mando ha finalizado y que tras unos segundos podrá volver a hacer uso del control domótico de su hogar. Véase Ilustración 60.