

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Control domótico low-cost mediante asistente de voz utilizando software Open Source



Grado en Ingeniería
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Joffre Alcívar Riera

Ander Gabilondo Areta

Pamplona, 19 de junio

AGRADECIMIENTOS

Este trabajo lo agradezco a mis padres, que sin el apoyo incondicional que me han dado a lo largo de los años esto no hubiera sido posible, su ayuda y su insistencia han hecho que pueda acabar estos estudios. Gracias a vosotros dos, el mejor padre y la mejor madre.

También quiero agradecer este trabajo a todos aquellos que os conocí en la universidad, formáis parte de mi vida universitaria y que sin vosotros esto se hubiera hecho muy duro, gracias amigos.

A mi tutor, Ander, por ayudarme y ser comprensivo en este semestre tan difícil.

A ti Javier, que sin ti esto no podría salir adelante, ya que preguntándome constantemente inspiras mis ideas.

A 尹兢, por darme tu tiempo, hacerme conocer nuevos mundos y abrir más mi mente, eres importante para mí.

RESUMEN

En este trabajo de fin de grado se realiza una prueba de concepto de una instalación domótica. Se integrarán varios dispositivos con diferentes protocolos domóticos.

El control de esta instalación se realizará por la voz, una página web, un teléfono móvil o incluso exponer dispositivos por IFTTT.

Se explican los servicios en la nube y los conceptos de la plataforma. Se muestra la integración multi-protocolo en una única solución.

PALABRAS CLAVE

Domótica KNX Open Source openHAB IFTTT Alexa Google Assistant Raspberry Pi Código Abierto Cloud Service Linux

ÍNDICE

AGRADECIMIENTOS	3
RESUMEN.....	4
PALABRAS CLAVE	4
1. OBJETIVOS.....	7
2. INTRODUCCIÓN Y ESTADO DEL ARTE.....	8
3. REQUISITOS Y NECESIDADES PLANTEADAS.....	11
4. HERRAMIENTAS UTILIZADAS	12
4.2 ORDENADOR DE BAJO COSTE.....	12
4.2 CONEXIÓN PERMANENTE A INTERNET	12
4.3 INTERACTIVIDAD CON EL USUARIO.....	13
4.4 CONTROL REMOTO CON SMARTPHONE, TABLET O PC	14
4.5 SERVICIOS EN LA NUBE PARA EL CONTROL DE VOZ	14
4.5.1 SERVICIO DE VOZ DE ALEXA AVS.....	15
4.6 SERVICIO EN LA NUBE PARA EL ACCESO REMOTO.....	15
4.7 INTERFACES PARA INTEGRACIÓN MULTIPROCOLO	15
4.8 DISPOSITIVO DE ELECTRÓNICA DE CONSUMO.....	15
5. DESARROLLO E IMPLEMENTACIÓN.....	16
5.1 QUÉ ES OPENHAB E INSTALACIÓN	16
5.1.1 ESTRUCTURA DE OPENHAB	16
5.1.2 INSTALACIÓN DE OPENHABIAN	17
5.2 CONCEPTOS DE LA HERRAMIENTA OPENHAB	21
5.2 CONFIGURACIÓN DE UN SITEMAP E ITEMS.....	21
5.3 BINDING DE KNX CON OPENHAB Y REGLAS CON ESCENAS.....	23
5.3.1 BINDING DE KNX.....	23
5.3.2 CONFIGURACIÓN DE UNA REGLA CON ESCENAS.....	25
5.4 ASISTENTE DE VOZ.....	26
5.4.1 CONFIGURACIÓN DE OH CLOUD	26
5.4.2 CONFIGURACIÓN DE ALEXA.....	27
5.6 CONFIGURACIÓN DE THINGS PARA EXPONER A ALEXA	28
5.7 POSIBLES USOS	28
6. MONTAJE DEL ENTORNO Y COSTE DE OTRAS SOLUCIONES.....	29

6.1 MONTAJE EN LA RASPBERRY PI	29
6.2 COSTE DE OTRAS SOLUCIONES	30
7. CONCLUSIONES Y LÍNEAS FUTURAS	31
8. BIBLIOGRAFÍA	32
8.1 FIGURAS.....	34
8.2 MARCOS	34
9. ANEXO	35
9.1 HOJA DE ESPECIFICACIONES DE LA RASPBERRY PI 3 MODELO B	35
9.2 SERVICIO DE VOZ DE AMAZON.....	36

1. OBJETIVOS

En este trabajo los objetivos que se llevarán a cabo son los siguientes:

1. Recorrido por los estándares actuales en domóticas, profesionales y de electrónica de consumo.
2. Descripción de plataformas de código abierto para instalaciones domóticas.
3. Comparación con otras soluciones en el mercado.
4. Controlar una instalación domótica de una manera más natural: Con un asistente de voz, una pantalla táctil, o desde cualquier parte con Internet.
5. Integración de varias soluciones existentes en el mercado en una plataforma única.
6. Reducción de los costes con relación a soluciones comerciales.
7. Implementación de una solución de instalación simple y al alcance de cualquier persona.

2. INTRODUCCIÓN Y ESTADO DEL ARTE

Para la instalación domótica profesional en una casa, se utilizan protocolos como KNX.

En la actualidad, según los datos aportados por un estudio de CEDOM [1], el número de instalaciones que cuentan con dispositivos propietarios son muy similares a las de protocolos abiertos.

El mercado para la eficiencia energética, debido a las directivas de la UE de uso de energías limpias para el cambio climático, se estima que experimentará una subida considerable en los próximos años. Se tiene como frontera reducir en al menos un 40% las emisiones de dióxido de carbono para 2030.

La economía después de los años de crisis está creciendo con ligeros repuntes en la construcción y esto se ve reflejado en la tendencia al alza del volumen de mercado de las compañías relacionadas con la domótica e inmótica, según los datos recabados en el estudio de CEDOM.

En las viviendas se está poniendo de moda tener soluciones domóticas de instalación simple y corta vida útil. Ejemplos de estos productos son sensores de humedad para plantas (Xiaomi Mijia), Amazon Echo, bombillas Hue de Philips, etcétera.



Figura 1. Xiaomi Mi Plant

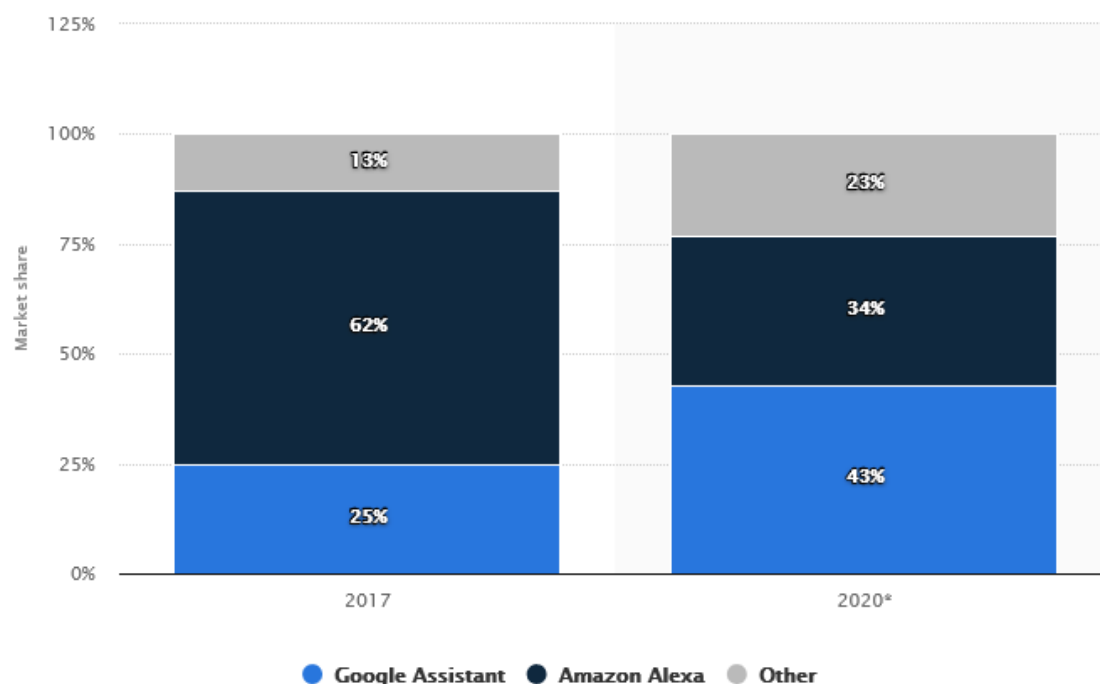


Figura 2. Cuota de mercado estimada para asistentes de voz en los E.U.A.

Existen varios asistentes de voz que hoy en día se encuentran compitiendo entre ellos, sin un claro ganador en el corto plazo [2].

Entre ellos se encuentran Siri, Google Assistant o Alexa, además de motores de IA para poder integrar servicios web multi-propósito (IBM Cloud). Habiendo tantas opciones y sin un estándar de facto, en la actualidad se está construyendo lo que en el futuro puede ser el estándar domótico que los conecta a todos, o que simplemente todos estos dispositivos se integren en una suite futura IoT, algo similar a lo que ocurre en la actualidad con la suite de protocolos TCP/IP [3].

En los dispositivos y servicios propietarios mencionados anteriormente, no se llega a un estándar abierto. Para el usuario es necesario la interactuar de manera transparente y sin apenas configuración por parte de un técnico. No está extendido el UPnP como en las redes.

Desde el punto de vista de un desarrollador de aplicaciones existen frameworks como Eclipse SmartHome (sistema modular) [4].

OpenHAB en su versión 2.x se basa en Eclipse SmartHome. Existen entornos de desarrollo propietarios como Apple HomeKit o soluciones abiertas que lo simulan como HomeBridge [5]. Este software Open Source se ha creado debido a la ubicuidad de los productos Apple en algunos países, y por su gran comunidad de código abierto [6].

Control domótico low-cost mediante asistente de voz utilizando software Open Source

Por último, hay productos de empresas como Nest. Son soluciones integrales para que una instalación antigua de calefacción se modernice. Como resultado se tiene comodidades propias de despliegues de alto coste a un precio ajustado.

3. REQUISITOS Y NECESIDADES PLANTEADAS

En las instalaciones basadas en KNX los costes asociados son bastante altos, por ende, uno de los objetivos de este trabajo es reducir estos costes.

Asimismo, como se ha comentado en apartados anteriores, surge una necesidad de tener algún dispositivo o servicio capaz de integrar en sí mismo múltiples protocolos. Esto se pretende realizar de manera centralizada y conectada a la nube. En un mismo hogar pueden existir dispositivos conectados por wifi, z-wave, bluetooth, etcétera. Por todo esto, es imperante que los aparatos sean capaces de interactuar entre ellos de manera transparente.

Por otra parte, el usuario debe tener las mayores facilidades para poder tener acceso a su instalación domótica. Es necesario interactuar por distintos medios como la voz, una aplicación una página web. Hay métodos más novedosos: la interacción mediante la voz.

Para el desarrollo de este trabajo se deben tener en cuenta los siguientes requisitos técnicos:

1. Ordenador de bajo coste.
2. Conexión a internet de manera permanente.
3. Interacción con el usuario mediante voz y audio.
4. Un smartphone para poder controlar nuestra instalación.
5. Servicios en la nube para un asistente de voz.
6. Servicios en la nube para acceso remoto.
7. Interfaces que permitan integrar varios protocolos con nuestro dispositivo central.
8. Dispositivo de electrónica de consumo que integre domótica en alguna forma para la prueba de concepto.

Al comienzo de este trabajo se eligió para el interfaz KNX la solución del fabricante italiano Eelectron [7], pero debido al alcance de este proyecto se optó por otra solución. La selección es el interfaz USB KNX del fabricante Jung.

4. HERRAMIENTAS UTILIZADAS

Para completar este proyecto ha sido necesario barajar varias opciones. Ahora se procederá a una explicación breve de varias alternativas. Los productos o entornos seleccionados serán explicados de una manera más extensa.

4.2 ORDENADOR DE BAJO COSTE

Para la elección del ordenador de bajo coste se debe tener en cuenta varias características, las más importantes son:

- Bajo coste
- Bajo consumo eléctrico
- Alta disponibilidad y fiabilidad
- Múltiples interfaces: varios puertos, alta conectividad
- Documentación extensa
- Soporte de alguna distribución de Linux de manera oficial
- Soporte por parte de la comunidad, es decir, que existan proyectos de los que forme parte este microordenador

Debido a que se elige el software openHAB, se selecciona un ordenador que tenga soporte oficial.

Se desarrollará todo en una Raspberry Pi en su versión 3 y modelo B, por su amplio soporte de la comunidad de software libre (Raspbian) [8]. OpenHAB ofrece soporte oficial con openHABian, esta es una distribución Raspbian modificada.

En el primer arranque de Raspbian se autoejecuta un script para la instalación de los paquetes de la distribución de openHAB. Las especificaciones técnicas son más que suficientes para poder ejecutar sobre el mismo dispositivo un servidor web, java, Alexa y varios interfaces para otros protocolos. Las especificaciones de la Raspberry se adjuntan en la página 35 del anexo.

4.2 CONEXIÓN PERMANENTE A INTERNET

En este caso, para la demostración se opta por conectar la Raspberry Pi mediante un cable ethernet a un ordenador con internet. Se puede apreciar el montaje de esta instalación en la figura [FIGURA 3](#).

También sería factible conectar el microordenador directamente a un router de casa, por WiFi o por cable. Una forma con mayor independencia respecto la instalación del hogar es conectar la Raspberry a un módulo LTE. Así se tiene redundancia en el caso de que la conexión fuese crítica para el control de las instalaciones [9].



Figura 3. Conexión Raspberry - PC

4.3 INTERACTIVIDAD CON EL USUARIO

Se han considerado varias maneras para interactuar con el usuario. Las más naturales son el uso de la voz y una pantalla táctil, estas son las opciones que permite la tecnología actual.

Si se optase por integrar una pantalla táctil, resultaría caro, todavía más si el módulo usado fuese el oficial. En este caso se ha decidido como alternativa controlar la instalación mediante un móvil.



Figura 4. Raspberry con pantalla táctil

Los altavoces para la prueba de concepto son autoalimentados por USB. Se conecta al mini-jack para el audio y a uno de los USBs libres, para la alimentación.

La Raspberry no dispone de una entrada de audio. Para suplir esta carencia se ha elegido un micrófono USB, pero también valdría uno que se conectase a los GPIOs.

4.4 CONTROL REMOTO CON SMARTPHONE, TABLET O PC

Para este apartado es necesario un dispositivo con conexión a internet y un navegador actualizado. Se conectará a openHAB Cloud para controlar la interfaz web de openHAB.

4.5 SERVICIOS EN LA NUBE PARA EL CONTROL DE VOZ

El control por voz elegido es Alexa, de Amazon. Se ejecuta un servicio al arranque de Raspbian encargado de interpretar los comandos de voz y actuar acordeamente.

Se selecciona Alexa por el soporte dado por Amazon, por su fácil instalación con AlexaPi y porque permite utilizar la Raspberry Pi como un cliente del servicio de voz de Alexa.

Una vez que el cliente de Alexa funciona se tiene las *Alexa Skills*. Con su uso se es capaz de controlar dispositivos compatibles con el servicio.

Se puede controlar dispositivos no soportados por Alexa, a través de un intermediario. En este trabajo los intermediarios son openHAB o IFTTT.

En resumen, para este apartado es necesario exponer los dispositivos al servicio de Alexa y éste se encargará de interactuar con la voz del usuario.

4.5.1 SERVICIO DE VOZ DE ALEXA AVS

El funcionamiento del cliente de voz de Amazon Voice Service, que se ha tomado del AVS SDK [10], se puede encontrar en la página 36 del anexo.

4.6 SERVICIO EN LA NUBE PARA EL ACCESO REMOTO

Debido a que la Raspberry Pi está detrás de un firewall y no está conectada directamente a Internet (utiliza NAT), es necesario tener algún servidor externo. La opción de conectarla directamente a internet se descarta porque supone un problema de seguridad.

Se ha decidido utilizar openHAB Cloud (OH Cloud) porque así no se cambia la topología y el servicio de alta disponibilidad es ofrecido por un tercero. En este caso la instancia de openHAB Cloud está alojada en un servidor de la misma organización.

4.7 INTERFACES PARA INTEGRACIÓN MULTIPROCOLO

Este trabajo está enfocado hacia la implementación de openHAB con KNX, para ello se conecta un interfaz USB KNX a través del servicio de knxd (ejecutando un daemon en segundo plano).

Una vez se tiene habilitado el servicio, es posible exponer los dispositivos KNX a openHAB y a su vez a internet mediante OH Cloud.

También se baraja la posibilidad de integrar aparatos compatibles con IFTTT a través del servicio de voz de Alexa y OH Cloud, e incluso con otros asistentes de voz.

4.8 DISPOSITIVO DE ELECTRÓNICA DE CONSUMO

En este apartado se ha elegido la lámpara de Yeelight por su conexión WiFi integrada. Tiene además una aplicación propia que puede utilizarse con los servicios en la nube de OH Cloud, Alexa e IFTTT.

5. DESARROLLO E IMPLEMENTACIÓN

En este capítulo se procederá a la explicación en detalle de cada uno de los conceptos que se manejan en la herramienta de openHAB, asimismo, las tecnologías que van detrás. Se mostrará el proceso de instalación detalladamente además de una guía paso a paso.

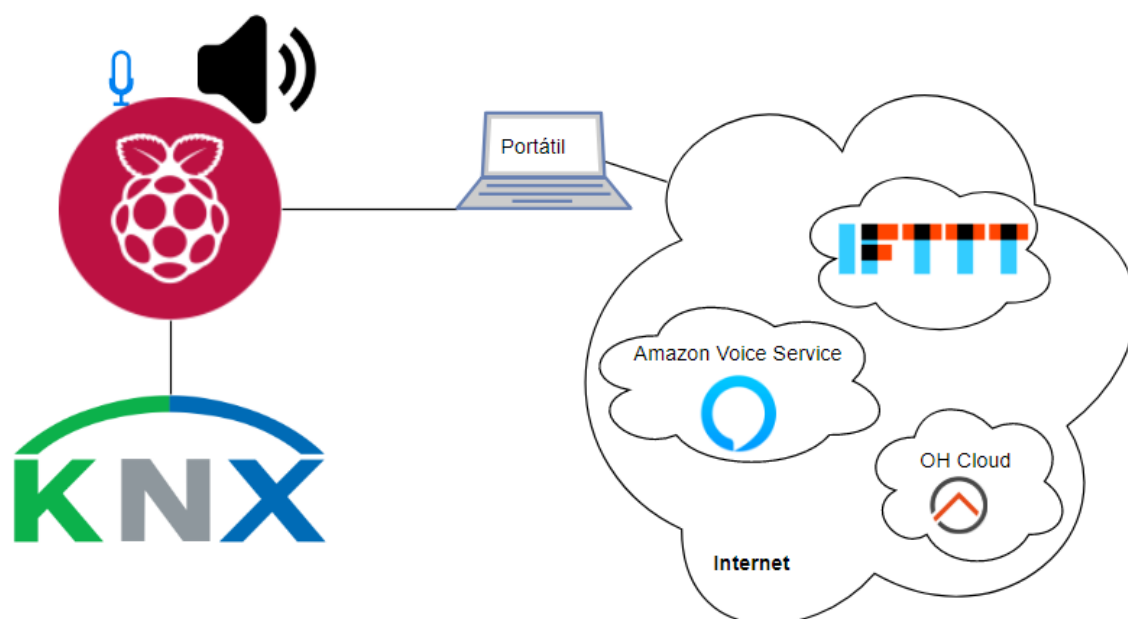


Figura 5. Esquema Lógico de la instalación

5.1 QUÉ ES OPENHAB E INSTALACIÓN

OpenHAB (**open Home Automation Bus**) es una solución de software integral centralizada desarrollada en Java [11]. A través de distintos interfaces se convierte en una arquitectura totalmente modular. Es capaz de expandirse a múltiples protocolos y comunicarse con varios productos y software. Amazon Dash, Kodi, un coche (Tesla) son algunos ejemplos a los que se puede vincular openHAB. También existe la capacidad de integrar un asistente por chat, en definitiva, con la configuración adecuada es totalmente multipropósito.

5.1.1 ESTRUCTURA DE OPENHAB

La arquitectura de openHAB, como aparece en la FIGURA 6, consiste en un desarrollo en Java basado, en el framework Eclipse SmartHome [12]. Es un sistema modular debido a cómo está implementado.

El sistema consiste en un contenedor de Apache Karaf (con arquitectura OSGi) que es capaz de integrar Servlets¹.

Sobre esta base se ejecuta la librería de programación de Eclipse Equinox. Todo ello brinda la modularidad propia de los sistemas OSGi [13]:

1. Empaquetar programas listos para el despliegue en el servidor
2. Añadir más paquetes en caliente sin necesidad de reiniciar el servidor
3. Entornos aislados del resto de programas

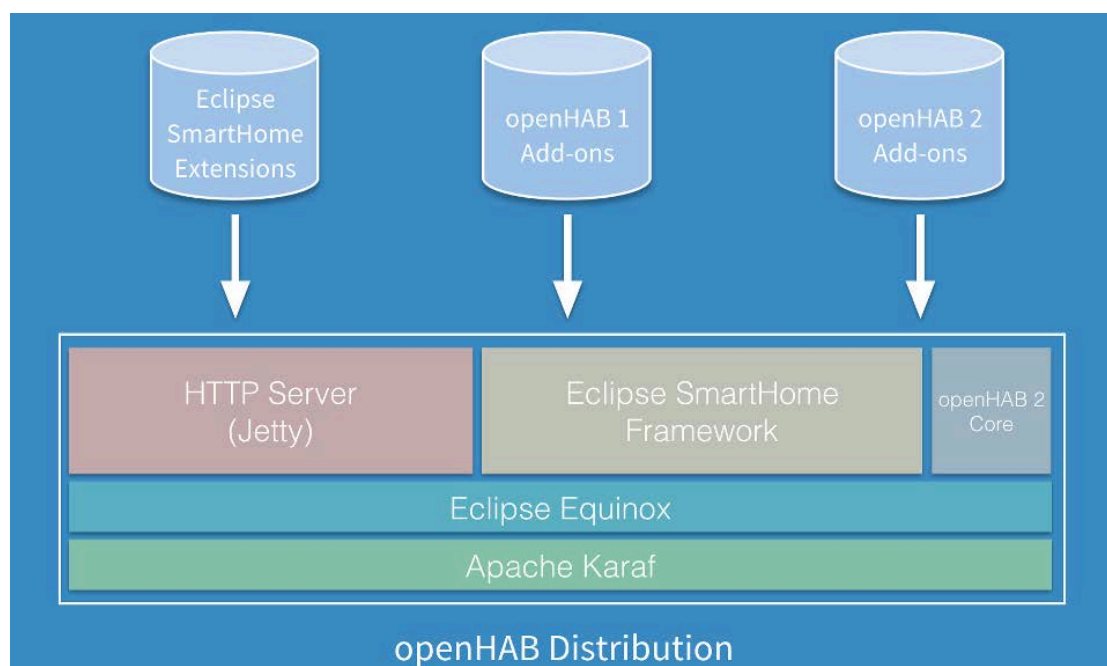


Figura 6. Arquitectura de openHAB

5.1.2 INSTALACIÓN DE OPENHABIAN

Como se comentaba en el capítulo anterior, openHABian es una distribución de software libre que permite la instalación de openHAB en una Raspberry.

OpenHABian está basado en la famosa distribución Raspbian. Este sistema operativo incluye de serie los archivos de configuración y repositorios de la openHAB Foundation.

Los pasos que seguir para la instalación son los siguientes:

1. Se descarga la última imagen desde la página de openHAB, en el apartado de instalación [14].

¹ Pequeño programa de Java capaz de responder a peticiones web.

2. Grabar la imagen en una microSD con programas como Rufus [15].
3. Para tener el acceso por SSH activado es necesario poner un archivo de texto vacío sin ninguna extensión que se llame "ssh".
4. Se conecta la Raspberry a un router que tenga habilitado el servicio de DHCP para tener una dirección IP asignada. Este paso se requiere para la conexión a internet, necesaria para descargar los paquetes de openHAB.
5. Esperar alrededor de unos 30 minutos para que acabe la instalación. Este proceso es posible monitorizarlo si se conecta la Raspberry por HDMI, así se autoiniciará una sesión con una shell.
6. OpenHAB anuncia a través de la resolución de nombre local (DNS local) "openhabianpi". Esto no es necesario si es posible ver la IP autoasignada en el servidor de DHCP del router.
7. Se conectará al servicio IP:8080. Usuario y contraseña son "openhabian", tanto para el servicio web como para el usuario de Linux. Se proceden a cambiarlos por seguridad, ya que el dispositivo estará expuesto a Internet.
8. Para mayor versatilidad para conectarnos al servicio de SSH se especifica los comandos siguientes como aparecen en el [MARCO 1](#)

```
#####
### openHABianPi #####
#####
##      Ip = 192.168.137.2
##      Release = Raspbian GNU/Linux 9 (stretch)
##      Kernel = Linux 4.14.34-v7+
##      Platform = Raspberry Pi 3 Model B Rev 1.2
##      Uptime = 0 day(s). 11:51:5
##      CPU Usage = 1.01 % avg over 4 cpu(s) (4 core(s) x 1 socket(s))
##      CPU Load = 1m: 0.32, 5m: 0.34, 15m: 0.30
##      Memory = Free: 0.23GB (26%), Used: 0.67GB (74%), Total: 0.91GB
##      Swap = Free: 0.09GB (100%), Used: 0.00GB (0%), Total: 0.09GB
##      Root = Free: 24.58GB (90%), Used: 2.62GB (10%), Total: 28.40GB
##      Updates = 4 apt updates available.
##      Sessions = 1 sessions
##      Processes = 102 running processes of 32768 maximum processes
#####

Welcome to
openHABian
openHAB 2.3.0-1 (Release Build)
```

Figura 7: Prompt de openHABian

Con los pasos anteriores se tiene una Raspberry con openHAB y acceso ssh para cualquier gestión del sistema operativo.

```
$ ssh-keygen
# Se da un nombre a la clave generada
$ ssh-copy-id -i $ruta_clave_privada openhabian@IP
# Se crea un alias para la conexión a la
# Raspberry
# Se crea una entrada en el archivo ~/.ssh/config
# con la siguiente sintaxis:
##### Comienzo del archivo #####
# host $nombre_de_alias
#     hostname $IP_openhabian
#     user $usuario_openhabian
#     IdentityFile $ruta_clave_privada
##### Fin del archivo #####
# Cumpliendo estos requisitos se puede conectar a la
# Raspberry directamente
$ ssh $nombre_de_alias
```

Marco 1. Instrucciones para conexión por SSH

Es necesario utilizar una IP estática por la facilidad de acceder al dispositivo con nuestro ordenador de trabajo. Como la solución es con un ordenador con Windows 10, el direccionamiento de compartir internet seguirá el que aparece en el [MARCO 2](#). Con estas instrucciones se tendrá una IP estática con systemd.

```
# Crear un archivo en el directorio /etc/systemd/network/  
# en este caso se opta por configuracion.knx  
# En el archivo se propone esta configuración,  
# notar que la subred utilizada es la predeterminada del  
# direccionamiento para compartir internet de Windows 10,  
# así que es importante conservarlo  
  
##### Aquí el archivo de configuración #####  
[Match]  
Name=eth0 # Nombre del interfaz utilizado  
[Network]  
Address=192.168.137.2/24 # IP/Máscara  
Gateway=192.168.137.1 # IP Windows 10  
DNS=192.168.137.1 9.9.9.9  
# En este apartado se ha incluido el DNS 9.9.9.9 como  
# redundancia  
##### Fin del archivo de configuración #####  
  
# Posteriormente se procede a deshabilitar los servicios  
# de DHCP que pudieran estar habilitados  
# Se hace una copia por si no funciona el proceso  
$ mv /etc/network/interfaces /etc/network/interfaces.bak  
$ systemctl enable systemd-resolved.service  
$ systemctl enable systemd-networkd.service  
$ systemctl start systemd-resolved.service  
# Deshabilitar DHCP  
$ systemctl disable dhcpcd.service  
# Se habilita el servicio que espera a que se haya  
# autoconfigurado la ip estática  
$ systemctl enable systemd-networkd-wait-online.service
```

Marco 2. Configuración de la IP

5.2 CONCEPTOS DE LA HERRAMIENTA OPENHAB

En este apartado se procederá a la explicación de los conceptos para entender la estructura de openHAB.

Things son las entidades físicas en una instalación, cualquier servicio o dispositivo que provea de alguna funcionalidad a openHAB.

Bindings corresponde al driver que se encarga de asociar un *Thing* con openHAB. Esto es similar a la capa de abstracción en las topologías TCP/IP, al nivel dos (nivel de enlace).

Items son las propiedades que tiene un *Thing* interpretados por la aplicación de openHAB.

Links la unión entre un *Thing* y un *Item*.

Channel son las características de cada *Thing*.

Para una mayor claridad se pondrá de ejemplo la lámpara Xiaomi Yeelight [16] junto con la Raspberry y el micrófono. En este ejemplo la lámpara es una *Thing*, el tono y la intensidad de la luz son los *Channel* y el *Binding* sería la unión entre IFTTT/Alexa con openHAB.

Un *Item* sería la cantidad de luz que tiene la lámpara visto desde openHAB.

5.2 CONFIGURACIÓN DE UN SITEMAP E ITEMS

En este apartado se muestra un ejemplo de cómo se crea un sitemap² y la definición de items integrables en Alexa con KNX.

De acuerdo con la documentación [17] se explica en el [MARCO 3](#) cómo configurar un *Item* junto a un *Channel* en la *Basic UI*³.

² Archivo que define la vista en la vista de openHAB Basic UI

³ Vista básica de los *Items* en openHAB

```
# Se accede al directorio /etc/openhab2/ donde se
encuentran los archivos de configuración.
$ cd /etc/openhab2
# Se crean los siguientes archivos
$ > ./sitemap/$nombre_de_archivo.sitemap
$ > ./items/$nombre_de_archivo.items
# En items se especifican los items con la siguiente
# sintaxis
#
# ItemType item=ItemName label="Etiqueta que se ve en la
# Basic UI" <icono> {channel="canal que se elige en la
Paper UI"}
#
# Posteriormente se crea el sitemap
#
# sitemap $nombre_de_archivo
# { ItemType item=ItemName label="Etiqueta que se ve en
la Basic UI" }
```

Marco 3. Configuración del sitemap

Una vez configurado esto se debería tener en la *Basic UI* algo similar a lo que aparece en [FIGURA 8](#). En el siguiente apartado estos dos Items van a utilizarse para la creación de reglas.

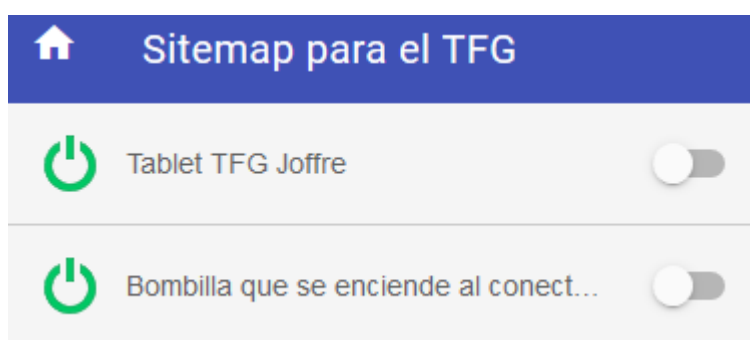


Figura 8. Sitemap de ejemplo

5.3 BINDING DE KNX CON OPENHAB Y REGLAS CON ESCENAS

En este capítulo se realizará con *Bindings* de dispositivos de KNX en openHAB. Una vez configurados los aparatos, se procederá a definir reglas. El desarrollo consistirá en automatizar un dispositivo que tenga dos estados (encendido/apagado). Se activará en el caso de que detecte una tableta en la red de área local.

5.3.1 BINDING DE KNX

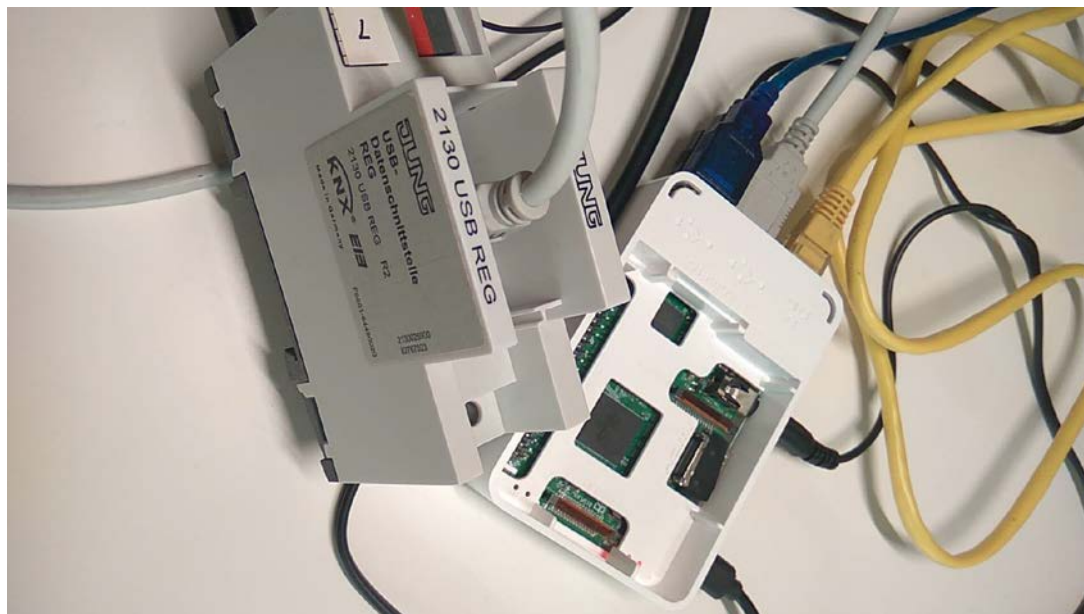


Figura 9. Interfaz USB KNX Jung con la Raspberry Pi

Para este apartado es necesario una conexión a Internet y una instalación funcional de openHAB en una Raspberry Pi.

Se utiliza el manual de la web del proyecto de knxd [18]. En este trabajo, se parte a partir del punto en el que ya se tienen instalados los paquetes .deb.

Por la necesidad de configurar el servicio con los parámetros deseados, se crea un archivo que define el servicio (en este caso se está trabajando con systemd). A continuación, se muestra un proceso de instalación simplificado en el [MARCO 4](#).

```
# Crear y asociar el usuario con el grupo knxd
# esto es necesario porque la daemon se ejecutará bajo
# este grupo.
$ usermod -a -G knxd $usuario_a_añadir
# Posteriormente se crea una regla con el nombre de
# dispositivo KNX USB para que sea posible conectarse a
# través de knxd
# en el post [19] se explica este paso, con los ids
# asociados a cada producto de KNX.

# Para probar el funcionamiento de knxd se ejecuta knxd
# con las siguientes opciones
##### A incluir en el archivo #####
KNXD_OPTIONS="--eibaddr=$DIRECCION_KNXD --client-
  addr=$KNX_DIRECCION_CLIENTES_INICIO:$KNX_DIRECCION_CLIEN
  TES_FINAL -d -D -T -R -S -i --listen-local=/tmp/knx -b
  usb:"
##### Fin del archivo #####
$ knxd $KNXD_OPTIONS
# En este caso las opciones corresponden a la ejecución
# en una daemon de un servicio capaz de comportarse como
# un KNX Net/IP Gateway o router, además de escuchar por
# el interfaz KNX USB los telegramas en el bus de KNX
```

Marco 4. Configuración de knxd

Para más detalle en la configuración del servicio se encuentra en la página del proyecto de knxd [19].

Una vez el servicio knxd se ejecuta correctamente en el *systemd*, se procede a la *Paper UI*. Es inmediato el unir estos dispositivos KNX a openHAB mediante un *Binding* [20], sin embargo, es necesaria la configuración de *Things* para la creación de reglas, sitemaps, etc.

5.3.2 CONFIGURACIÓN DE UNA REGLA CON ESCENAS

Las reglas son condiciones para que se efectúe una acción, por ejemplo, una bombilla se enciende cuando un móvil se conecta a la red WiFi del hogar a determinada hora. En resumen, se detecta la presencia de una persona cuando llega del trabajo a una determinada hora.

En openHAB es posible integrar reglas. El propio sistema incluye un método para configurarlas. Se especifica a continuación en el [MARCO 5](#) la configuración necesaria para la regla explicada en el párrafo anterior.

```
# En el directorio de configuración de openHAB en
# conf/rules/$nombre_regla.rules
# se escribe en el archivo lo siguiente
### Inicio ###
rule "Bombilla_encendida_usuario_casa"
when
Item Tablet_Online_Joffre changed from OFFLINE to ONLINE
and Time is noon
then
Bombilla_KNX_Joffre.sendCommand(ON)
end
### Fin del archivo ###
```

Marco 5. Regla de ejemplo

En esta regla primero se define su nombre, después los objetos con su respectivo cambio de estado. Finalmente se ejecuta una acción, en este caso se cambia el estado de un objeto. Esto se traduce en una bombilla que se enciende ante un suceso.

5.4 ASISTENTE DE VOZ

Para el uso de la voz se ha utilizado el asistente de voz gratuito que ofrece Amazon, Alexa. La pila de protocolos y servicios que conectan openHAB con Alexa e IFTTT se muestran en la [FIGURA 10](#).

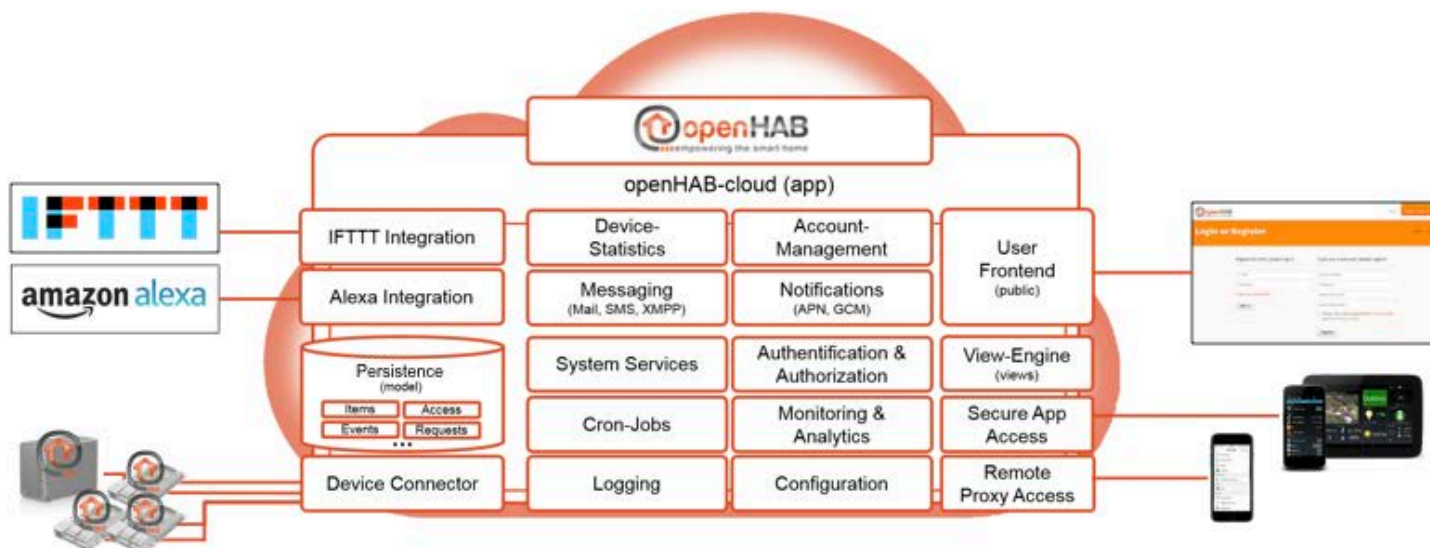


Figura 10. Esquema de funcionamiento de OH

5.4.1 CONFIGURACIÓN DE OH CLOUD

La instancia de openHAB de la Raspberry se conecta mediante una conexión persistente a OH Cloud. Se expone a internet de una manera segura los *Things* vinculados mediante un *Binding* a openHABian.

Para conectarse a OpenHAB Cloud (OH Cloud) es necesario un UUID (Identificador Universal Único) y un secreto [21]. En el [MARCO 6](#) hay más detalles.

```
# El UUID estará ubicado en /var/lib/openhab2/uuid
# El secreto en /var/lib/openhab2/openhabcloud/secret
# Una vez se tienen localizados esos archivos se
# introducen estos identificadores en myopenhab.org
# en la cuenta que se haya creado el usuario.
```

Marco 6. Configuración de OH Cloud

5.4.2 CONFIGURACIÓN DE ALEXA

Primero se instalará el micrófono y para que funcione en la Raspberry, se ha configurado con los siguientes parámetros, como se especifica en el [MARCO 7](#).

```
# Comprobar si el micrófono está conectado
$ arecord -l # lista todos los micrófonos conectados
# en la salida del comando aparecerá el micrófono
# a partir de los parámetros dados por la salida de dicho
# comando se guarda en ~/.asoundrc
# este será el micrófono predeterminado. En este caso en
# capture.pcm se pone el número de tarjeta conectada
# y en playback.pcm se conecta al 0 donde aparece
# "USB audio"
### Archivo a guardar para la configuración del micrófono
pcm.!default {
    type asym
    playback.pcm "plughw:0"
    capture.pcm  "plughw:1"
}

ctl.!default {
    type hw
    card 1
}
#### Fin del archivo #####
```

Marco 7. Configuración del micrófono en la Raspberry Pi

Para la instalación del cliente Alexa, se utiliza el servicio de AlexaPi disponible en GitHub [22]. Visto esto, la instalación es inmediata.

Con el servicio ejecutándose en el arranque, ahora la Raspberry responde a comandos de voz. Por ejemplo, ahora se puede decir: "Alexa, what's the weather in New York City?", y responderá con la actualidad del tiempo en la ubicación dada. Soporta cualquiera de las órdenes compatibles con el Amazon Echo, como programar una alarma, leer el calendario, etcétera. Se utiliza el asistente de voz para controlar la lámpara de Yeelight mediante IFTTT.

5.6 CONFIGURACIÓN DE THINGS PARA EXPONER A ALEXA

La configuración para exponer un *Item* de openHAB a Alexa sigue las pautas del etiquetado en HomeKit [23]. En el [MARCO 8](#) se explica la sintaxis que debe seguir.

Con los dispositivos expuestos a Alexa, es posible conectarlos a IFTTT para crear *Applets* [24]. Esto es debido a que los *Items* en openHAB utilizarán la API de Alexa.

```
# Teniendo un objeto definido por
Switch Bombilla_KNX_Joffre "Bombilla que se enciende al
conectar la tablet" <lightbulb> {}
# Se modifica de la siguiente manera
Switch Bombilla_KNX "Light KNX" <light> (gTest)
["Lighting"] {channel="..."}
# Respecto a la definición de un Item se añade nuevas
# etiquetas distintas a las de openHAB.
gTest, corresponde al grupo (habitación)
Lighting son el tipo de canal que reconoce Alexa
```

Marco 8. Things expuestos a Alexa

5.7 POSIBLES USOS

A partir de la solución explicada en este trabajo, se podría controlar la temperatura del hogar con un simple comando de voz.

Se pueden crear reglas que pudiesen automatizar funciones, es decir, escenas programables por el usuario.

6. MONTAJE DEL ENTORNO Y COSTE DE OTRAS SOLUCIONES

En este apartado se muestra la topología física del montaje de la instalación.

6.1 MONTAJE EN LA RASPBERRY PI

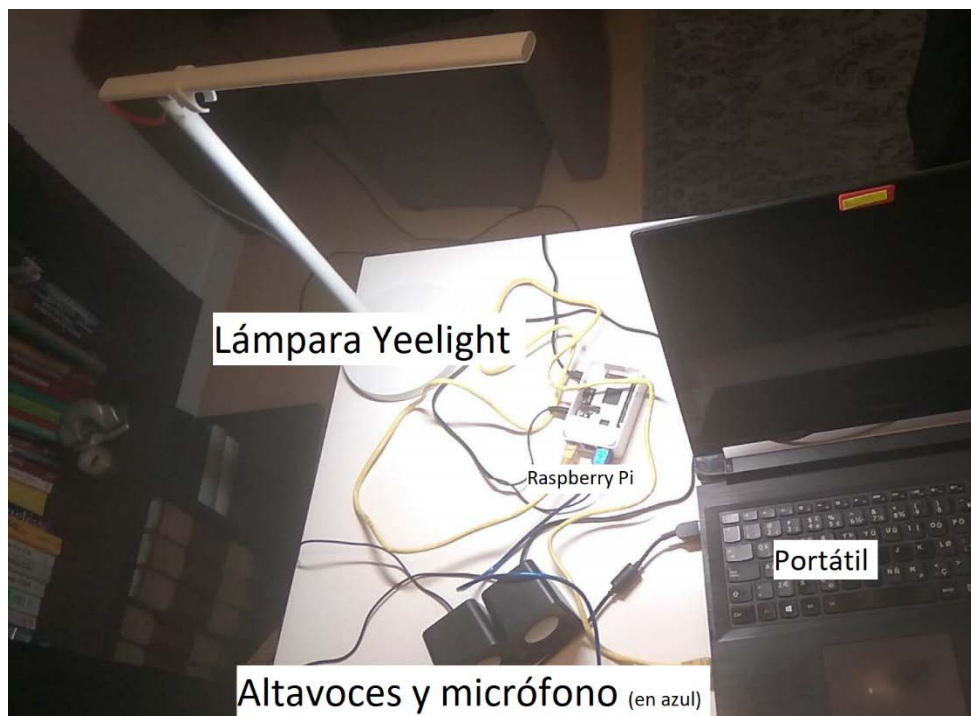


Figura 11. Montaje de la instalación

Como se puede observar en la FIGURA 11 el montaje de la instalación es inmediato.

El coste total de toda esta instalación es de 80€, teniendo en cuenta la microSD de la Raspberry, la Lámpara (~25€), los altavoces y micrófono (~10€) y la Raspberry Pi (~40€), si se suma el coste del USB KNX de JUNG el precio se incrementa en unos 200€. Sin embargo, si se optase por un adaptador de Busware conectando el KNX TPUART a la Raspberry, la subida del precio sería de unos 80€ [25].

6.2 COSTE DE OTRAS SOLUCIONES

En el mercado existen soluciones similares como las de BAB TECHNOLOGIE GmbH. Incluye un router IP-KNX y fácil integración con distintos asistentes de voz.

Para utilizar servicios de IoT hay que comprar aplicaciones en su propia tienda, por tanto el coste aumenta [26].



Figura 12. Solución de BAB-TEC

Otra opción es la de Voxior. Este fabricante vende el *Voxior Link* para conectarse al bus KNX y tiene un coste de 150€. Para los servicios en la nube y poder controlar la instalación remotamente, tienen un sistema de suscripciones con un coste de 75€/año, de nuevo, el coste se eleva [27].

7. CONCLUSIONES Y LÍNEAS FUTURAS

En la solución dada a lo largo de este trabajo se integra una lámpara con dos asistentes de voz a través de IFTTT, siendo todo este proceso transparente para el usuario una vez terminada la configuración.

El uso de software de código abierto en este proyecto, brinda versatilidad a la hora de integrar distintos dispositivos que en principio no pueden interactuar entre ellos. Se conecta servicios propietarios de voz con KNX a través de openHAB.

El acceso a servicios en la nube para integrarlo en domótica es fácilmente accesible y a un precio ajustado.

Con el abaratamiento de la electrónica se es capaz de integrar un sistema embebido en una lámpara, como la utilizada en este proyecto. Teniendo capacidad para conectar a Internet cualquier dispositivo. Se demuestra así que en el futuro la presencia de dispositivos inteligentes será la norma.

8. BIBLIOGRAFÍA

- [1] CEDOM, «Estudio de mercado 2016,» [En línea]. Available: <http://www.cedom.es/sala-de-prensa/noticias-sector-cedom/cedom-publica-su-estudio-de-mercado-2016>.
- [2] Statista, «WorldWide Digital Assistant Market Share,» [En línea]. Available: <https://www.statista.com/statistics/789633/worldwide-digital-assistant-market-share/>.
- [3] IBM, «IBM Cloud,» [En línea]. Available: <https://console.bluemix.net/docs/overview/ibm-cloud.html#overview>.
- [4] Eclipse Foundation, «Eclipse SmartHome Project,» [En línea]. Available: <https://www.eclipse.org/smarthome/>.
- [5] NFARINA, «Home Bridge Project,» [En línea]. Available: <https://github.com/nfarina/homebridge>.
- [6] Apple, «Iphone sales in the US Q4 2017,» [En línea]. Available: <https://9to5mac.com/2018/02/01/iphone-sales-usa-q4-2017/>.
- [7] Eelectron, «Raspberry Pi KNX interface,» [En línea]. Available: <http://www.eelectron.com/prodotto/raspberry-pi-knx-interface/?lang=en>.
- [8] Beta News, «Eben Upton Q&A,» [En línea]. Available: <https://betanews.com/2017/07/19/raspberry-pi-eben-upton-qa/>.
- [9] Sixfab, «Raspberry Pi 3g/4g/LTE shield v2,» [En línea]. Available: <https://sixfab.com/product/raspberry-pi-3g-4glte-base-shield-v2/>.
- [10] Amazon, «Alexa - AVS SDK,» [En línea]. Available: <https://alexa.github.io/avs-device-sdk/>.
- [11] OpenHAB Community, «OpenHAB Documentation,» [En línea]. Available: <https://www.openhab.org/docs/>.

- [12] OpenHAB community, "Arquitectura de openHAB," 2018. [Online]. Available: <https://www.openhab.org/docs/#architecture-overview>.
- [13] OpenHAB Foundation, «Architecture overview of openHAB,» [En línea]. Available: <https://www.openhab.org/docs/#architecture-overview>.
- [14] OpenHAB Community, «OpenHABian project,» [En línea]. Available: <https://github.com/openhab/openhabian/releases>.
- [15] Akeo, «Rufus,» [En línea]. Available: <https://rufus.akeo.ie/>.
- [16] Yeelight, «Mijia Lamp Product Overview,» [En línea]. Available: https://www.yeelight.com/en_US/product/mijia-lamp.
- [17] OpenHAB Community, «Sitemaps Documentation,» [En línea]. Available: <https://www.openhab.org/docs/tutorial/sitemap.html> .
- [18] KNXD, «knxd Project,» [En línea]. Available: <https://github.com/knxd/knxd>.
- [19] MichlsTech, «Raspberry Pi KNXD IP Gateway,» [En línea]. Available: <https://michlstechblog.info/blog/raspberry-pi-eibknx-ip-gateway-and-router-with-knxd/>.
- [20] OpenHAB Community, «KNX Bindings, Supported Things,» [En línea]. Available: <https://www.openhab.org/addons/bindings/knx/#supported-things>.
- [21] OpenHAB Community, «OH Cloud, Features,» [En línea]. Available: <https://docs.openhab.org/addons/ios/openhabcloud/readme.html#features>.
- [22] AlexaPi Project, «AlexaPi Installation,» [En línea]. Available: <https://github.com/alexa-pi/AlexaPi/wiki/Installation>.
- [23] OpenHAB Community, «OpenHAB HomeKit integration,» [En línea]. Available: <https://www.openhab.org/addons/integrations/homekit>.
- [24] IFTTT, «Amazon Alexa IFTTT Configuration,» [En línea]. Available: https://ifttt.com/amazon_alexa.

- [25] Busware, «Busware TPUART for Raspberry Pi,» [En línea]. Available: http://shop.busware.de/product_info.php/products_id/106 .
- [26] BAB-TEC, «BAB-TEC AppModule,» [En línea]. Available: <https://www.bab-tec.de/index.php/app-module-488.html>.
- [27] Voxior, «Voxior Product Overview,» [En línea]. Available: <https://www.voxior.com/shop> .

8.1 FIGURAS

Figura 1. Xiaomi Mi Plant.....	8
Figura 2. Cuota de mercado estimada para asistentes de voz en los E.U.A.....	9
Figura 3. Conexión Raspberry - PC.....	13
Figura 4. Raspberry con pantalla táctil.....	14
Figura 5. Esquema Lógico de la instalación	16
Figura 6. Arquitectura de openHAB	17
Figura 7: Prompt de openHABian.....	18
Figura 8. Sitemap de ejemplo	22
Figura 9. Interfaz USB KNX Jung con la Raspberry Pi.....	23
Figura 10. Esquema de funcionamiento de OH.....	26
Figura 11. Montaje de la instalación.....	29
Figura 12. Solución de BAB-TEC	30
Figura 13. Diagrama de AVS	36

8.2 MARCOS

Marco 1. Instrucciones para conexión por SSH	19
Marco 2. Configuración de la IP	20
Marco 3. Configuración del sitemap	22
Marco 4. Configuración de knxd	24
Marco 5. Regla de ejemplo	25
Marco 6. Configuración de OH Cloud	26
Marco 7. Configuración del micrófono en la Raspberry Pi.....	27
Marco 8. Things expuestos a Alexa	28

9. ANEXO

9.1 HOJA DE ESPECIFICACIONES DE LA RASPBERRY PI 3 MODELO B

Processor	Broadcom BCM2837
CPU Core	Quadcore ARM Cortex-A53, 64Bit
Clock Speed	1.2GHz (Roughly 50% faster than Pi2)
RAM	1 GB
GPU	400 MHz VideoCore IV®
Network Connectivity	1 x 10 / 100 Ethernet (RJ45 Port)
Wireless Connectivity	802.11n wireless LAN (WiFi) and Bluetooth 4.1
USB Ports	4 x USB 2.0
GPIOs	2 x 20 Pin Header
Camera Interface	15-pin MIPI
Display Interface	DSI 15 Pin / HDMI Out / Composite RCA
Power Supply (Current Capacity)	2.5 A

9.2 SERVICIO DE VOZ DE AMAZON

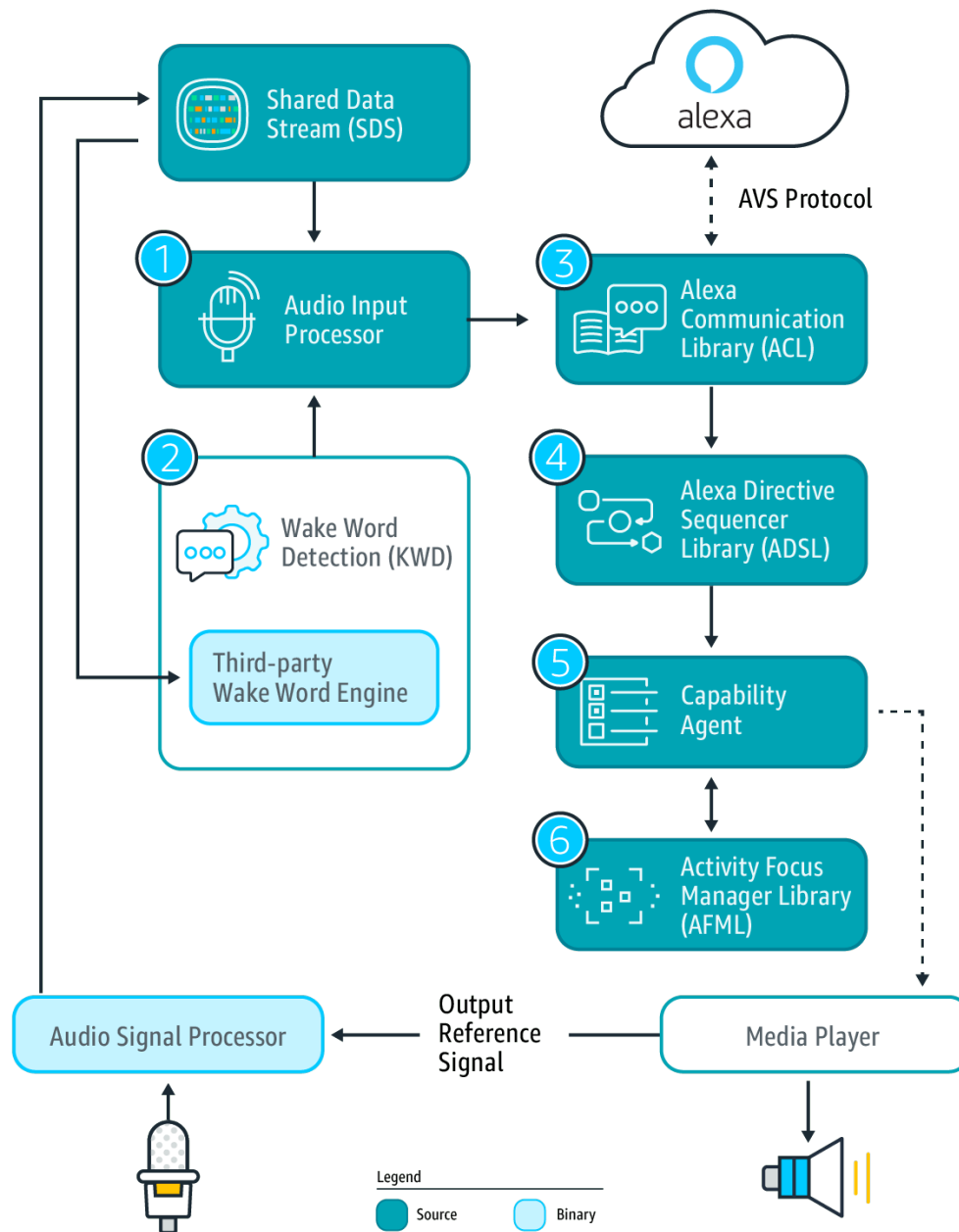


Figura 13. Diagrama de AVS

1. En primer lugar, la orden dada por el usuario es procesada por el ASP, que se encarga de filtrar y cancelar ecos.
2. El micrófono está permanentemente en escucha y el servicio solamente funcionará en el caso de que detecte la *Wake Word*. Evitando así crear tráfico innecesario con el servidor que da este servicio.
3. El ACL se encarga de mantener una conexión permanente al servidor a los servicios de Amazon, esto permite tener un servicio

en la nube detrás de un firewall, poder recibir notificaciones en cualquier momento, entre otros. En el caso de que la *Wake Word* sea detectada este servicio se encargará de comunicárselo al ADSL.

4. Teniendo varios canales posibles en el AVS (*Amazon Voice Service*), siendo estos diálogo, alertas y contenido, en el caso de que se detecte la palabra para despertar el servicio, el ADSL se encargará de tener una cola con las órdenes.
5. En el caso del CA, éste se encargará de manejar qué interfaces se activan o no, de todas las expuestas. En este trabajo eso corresponde a cualquier *Thing* conectada con openHAB y expuesta a AVS con su respectivo *Binding*.
6. Y por último el AFML se encarga de elegir que canal se encuentra en primer plano o no. Es decir, es capaz de reproducir música, pero en el caso de que detecte la *Wake Word*, se pasa al canal de diálogo, estando así a la escucha de la siguiente orden del usuario.

Hay que tener en cuenta que este servicio también es utilizado en los dispositivos de Alexa con pantalla, en estos casos la pantalla se toma como otro canal más, el CA será el que tiene la lista de canales disponibles, sea un panel táctil, una pantalla para stream de vídeo, una videocámara para videollamadas, etc.