



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Control inteligente del hogar mediante asistente de voz

Smart home control by voice assistant

Alexandra Esplugas Sabina

La Laguna, 6 de julio de 2020

D. **Jezabel Miriam Molina Gil**, con N.I.F. 78.507.682-B profesora Ayudante Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

D. **José Iván Santos González**, con N.I.F. 78.637.989-T investigador predoctoral en el grupo de investigación en criptología de la Universidad de La Laguna, como cotutor.

C E R T I F I C A (N)

Que la presente memoria titulada:

“Control inteligente del hogar mediante asistente de voz”

ha sido realizada bajo su dirección por D. **Alexandra Esplugas Sabina**,
con N.I.F. 79.085.385-T.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de julio de 2020

Agradecimientos

Agradecer a mi familia todo el apoyo que he recibido durante estos años para poder sacar mis estudios a delante.

A mis dos tutores Jezabel Molina y José Iván Santos por haberme dado la oportunidad de participar en este proyecto; así como su apoyo, ayuda y consejos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial-CompartirIgual 4.0 Internacional

Resumen

Los asistentes virtuales son cada vez más comunes en nuestro día a día, haciendo más fácil y dinámica la realización de diferentes tareas, ya sean en el hogar, en el trabajo o en otros ámbitos. Este tipo de tecnologías y la interacción que tenemos con ellas puede llegar a convertirse en una relación más personal.

Las relaciones entre humanos y máquinas se llevan desarrollando y mejorando desde hace muchos años, y no es de extrañar que hoy en día interactuemos más con la tecnología que con personas de carne y hueso.

El uso de asistentes virtuales, como Cortana, Siri, Google Assistant, Alexa y demás, se han extendido y es una herramienta que podemos usar a través de ordenadores, smartphones, altavoces inteligentes y cualquier dispositivo conectado a internet.

El objetivo de este trabajo es el desarrollo de una aplicación Android haciendo uso de la tecnología Google Assistant y con ello desarrollar un asistente virtual que, mediante peticiones por voz, ayude a la automatización de diferentes tareas del hogar.

El uso de la tecnología Google Assistant es clave para el desarrollo de nuestra aplicación, ya que es un asistente personal inteligente que nos permitirá realizar cualquier tipo de búsqueda mediante voz. Esto hará que nuestra aplicación sea sencilla a la hora de utilizarla, ya que solamente tendríamos que hacer la petición mediante voz y nuestro asistente se encargará de obtener la información para luego obtener la respuesta también por voz, por lo que al final será una conversación bidireccional.

Palabras clave: Google Assistant, asistente virtual, automatización, tareas del hogar.

Abstract

Virtual assistants are increasingly more common in our day-to-day life, making it easier and more dynamic to carry out different tasks, whether at home, at work or in other areas. This type of technology and the interaction we have with them can become a more personal relationship.

Human-machine relationships have been developing and improving for many years, and it is no wonder that today we interact more with technology than with flesh and blood people.

The use of virtual assistants, such as Cortana, Siri, Google Assistant, Alexa and others, have spread and it is a tool that we can use through computers, smartphones, smart speakers and any device connected to the internet.

The objective of this work is to develop an Android application using Google Assistant technology and thereby develop a virtual assistant that, through voice requests, helps to automate different household tasks.

The use of Google Assistant technology is key to the development of our application, since it is an intelligent personal assistant that will allow us to carry out any type of search by voice. This will make our application simple when using it, since we would only have to make the request by voice and our assistant will take care of obtaining the information and then obtain the answer also by voice, so that in the end it will be a two-way conversation.

Keywords: Google Assistant, virtual assistant, automation, homework.

Índice general

Capítulo 1	Introducción.....	1
1.1	Estructura del documento	1
1.2	Motivación.....	2
1.3	Objetivos.....	2
1.3.1	Aplicación móvil.....	3
1.3.2	Action para Google Assistant	3
Capítulo 2	Estado del arte	4
2.1	Iluminación y climatización del hogar	4
2.2	La seguridad en el hogar	4
2.3	Electrodomésticos	5
2.3.1	Robots aspiradora.....	5
2.3.2	Frigoríficos y lavadoras inteligentes	6
2.4	Altavoces inteligentes	7
2.4.1	Google Home.....	7
2.4.2	Alexa.....	8
Capítulo 3	Preliminares.....	9
3.1	Definición.....	9
3.2	Conceptualización.....	10
Capítulo 4	Tecnologías utilizadas.....	12
4.1	Aplicación móvil.....	12
4.2	Action de Google Assistant	15
4.3	Backend y Base de datos.....	18
4.4	Dispositivo bluetooth beacon	18
4.5	Otros.....	19

Capítulo 5	Aplicación móvil	20
5.1	Diseño	20
5.2	Funcionalidades	21
5.3	Actividad principal	25
Capítulo 6	Action para Google Assistant	27
6.1	Intent de bienvenida	28
6.2	Intent de despedida	29
6.3	Sobre la temperatura	30
6.4	Sobre la humedad	31
6.5	Encender el aire acondicionado y la calefacción	32
6.6	Intent de respuesta a los errores	33
6.7	Desplegar el proyecto con Firebase CLI	33
6.8	Webhook	36
Capítulo 7	Presupuesto	38
Capítulo 8	Conclusiones y líneas futuras	39
Capítulo 9	Conclusions and future lines	40

Índice de figuras

Figura 1: Climatización del hogar	4
Figura 2: La seguridad siempre es primordial (1).....	5
Figura 3: Robot aspiradora Roomba de IRobot	6
Figura 4: Cocina inteligente	7
Figura 5: Altavoces inteligentes Google Home, Google Home mini y Google Home Max.....	8
Figura 6: Amazon Alexa.....	8
Figura 7 : Funcionamiento de las peticiones a Google Assistant.....	10
Figura 8 : Funcionamiento de la Action de Google Assistant	11
Figura 9: iPhone vs Android: cuota de mercado (6).....	13
Figura 10 : Interfaz de la herramienta Android Studio.....	14
Figura 11 : Código XML y la visualización de la interfaz de la aplicación	15
Figura 12 : Asistente de Google desde tu móvil	16
Figura 13 : Proyecto de Firebase haciendo uso de la herramienta Realtime Database	18
Figura 14 : Dispositivo bluetooth beacon	19
Figura 15 : Los tres estados de los botones.....	20
Figura 16 : Visualización de la aplicación al iniciar.....	21
Figura 17 : Activando el bluetooth del dispositivo	22
Figura 18 : Iniciando conexión con el sensor	23
Figura 19 : Proceso donde se insertan los datos en la base de datos.....	24
Figura 20 : Interrumpiendo la conexión con el dispositivo	25
Figura 21 : Consola de Android Studio al iniciar el escaneo de la aplicación	26

Figura 22 : Consola integrada de Dialogflow	28
Figura 23 : Consola para simular proyectos de Google Assistant	28
Figura 24 : Intent de bienvenida	29
Figura 25 : Intent de despedida	29
Figura 26 : Sección training phrases de la Action de nuestra aplicación	30
Figura 27 : Consulta de la temperatura	31
Figura 28 : Consulta de la humedad	32
Figura 29 : Encendido de los dispositivos electrónicos de la casa	32
Figura 30 : Intent de respuesta a los errores.....	33
Figura 31 : Fulfillment de la Action en JavaScript imagen nº1	34
Figura 32 : Fulfillment de la Action en JavaScript imagen nº2	35
Figura 33: Consola de Firebase CLI.....	36
Figura 34 : URL de nuestro Webhook	36
Figura 35 : Fulfillment Request del servicio Webhook en formato JSON.....	37
Figura 36 : Fulfillment Response del servicio Webhook en formato JSON.	37

Índice de tablas

Tabla 1: Presupuesto del proyecto	38
------------------------------------------------	-----------

Capítulo 1

Introducción

1.1 Estructura del documento

Este documento tiene una estructura ordenada por los siguientes capítulos:

- Capítulo 1: Introducción.
Se comenzará hablando de la motivación y el objetivo de este trabajo.
- Capítulo 2: Estado del arte.
Se explicará la situación actual de la tecnología y de la investigación de otros proyectos relacionados con el tema escogido.
- Capítulo 3: Preliminares.
Se tratarán todas las etapas preparatorias necesarias para poner en marcha el proyecto.
- Capítulo 4: Tecnologías utilizadas.
En este capítulo se hablará también sobre las distintas herramientas y tecnologías utilizadas tanto en la aplicación como en el desarrollo del asistente virtual. Se explicarán con detalle el desarrollo de cada etapa y los conceptos más importantes con ilustraciones.
- Capítulo 5: Aplicación móvil.
Se describen los pasos seguidos para el desarrollo e implementación de la aplicación móvil.
- Capítulo 6: Action para Google Assistant.
Se detallará el procedimiento que se ha seguido para el desarrollo del asistente virtual con ilustraciones.

- Capítulo 7: Presupuesto.

Se presenta una pequeña estimación en una tabla donde se especifica el tiempo empleado para cada etapa del proyecto.

- Capítulo 8: Conclusiones y líneas futuras.

Resumen de la experiencia adquirida gracias a la realización de este trabajo y la conclusión del mismo.

- Capítulo 9: Conclusions and future lines.

1.2 Motivación

Esta aplicación está más enfocada a la domótica, una tecnología aplicada al control y a la automatización inteligente del hogar. Es precisamente esto lo que me ha llevado a motivarme a realizar este proyecto. La domótica está pensada para una gestión más eficiente del uso de la energía, como en este caso, el uso de la calefacción y el uso del aire acondicionado. Además, facilita la comunicación entre el usuario y el sistema. Esta comunicación hace que este proyecto sea interesante, ya que hace que la interacción con las máquinas sea más natural y fluida, creando una relación más personal entre usuario y asistente. Por esta razón este tipo de tecnología es tan atractiva para los usuarios y cada vez es más popular en el mercado.

1.3 Objetivos

El objetivo de este proyecto es crear un asistente virtual, que se compone de una sencilla aplicación Android, y gracias a la tecnología Google Assistant nos permitirá hacer peticiones mediante voz y obtener esta respuesta también mediante voz, creando una conversación fluida entre el usuario y la aplicación. En este caso, el usuario preguntará sobre la temperatura y la humedad de una habitación.

Está más enfocado a la domótica, que nos ayuda a la automatización de diferentes tareas del hogar, controlando el encendido y el apagado tanto del aire acondicionado como de la calefacción según la temperatura y la humedad que haya en el ambiente de una zona de la casa, que es donde tendremos nuestro sensor de temperatura.

Este proyecto consta de dos partes: una Action para Google Assistant y una aplicación Android para móvil.

1.3.1 Aplicación móvil

Nuestra aplicación será capaz de escanear los datos que emite un sensor de temperatura que se encontrará alojado en una habitación. Dicha información se almacenará en una base de datos, en este caso usaremos Firebase. Este sensor de temperatura emite datos constantemente y nuestra aplicación Android, mediante un escáner que funciona vía Bluetooth, será capaz de detectar los dispositivos que encuentre en su cercanía y capturar los datos que emite. La aplicación se conectará vía Wi-Fi o 4G a la base de datos para almacenarlos.

En la pantalla hay dos botones “Scan” y “Stop”. Al pulsar el botón de Scan se inicia el escaneo y por tanto se actualizarán los datos de manera instantánea hasta que pulsemos el botón Stop, que interrumpe la conexión con el sensor. En la base de datos de Firebase se quedarán almacenados los últimos datos que se escanearon. De esta manera siempre se tendrá actualizada la temperatura y la humedad de la habitación.

1.3.2 Action para Google Assistant

El objetivo es poder informar al usuario sobre la temperatura y humedad de una habitación, haciendo preguntas y devolviendo el valor que se almacena en la base de datos.

Para poder hacer peticiones por voz necesitamos hacer una consulta a nuestra base de datos mediante un request y poder obtener una respuesta también mediante voz. Se trata de una herramienta de creación de chatbots, lo que significa que nos permite crear conversaciones y construir diálogos para poder interactuar con nuestra aplicación de manera fluida. También permite la integración con Google Assistant, que es nuestro objetivo. Para ello creamos una acción o “action” de Google Assistant, que luego podemos utilizar tanto en smartphones Android como en los altavoces inteligentes de Google.

Capítulo 2

Estado del arte

La domótica lleva años en nuestras casas y esta tecnología es cada vez más conocida y es más común su utilización. Esta conectividad nos permite que nuestras casas sean más inteligentes y mejorar nuestra calidad de vida dentro de la vivienda, permitiendo hacer un control del hogar sin estar en casa, seguridad, robots que realizan tareas automáticas y repetitivas que antes realizaban los humanos, ahorro de energía para ser más respetuosos con el medio ambiente, etc. (1)

2.1 Iluminación y climatización del hogar

Desde nuestro smartphone se nos permite tener un control y regulación de los horarios de funcionamiento, tanto de la seguridad como la climatización y la optimización energética de nuestro hogar. Tener una mayor eficiencia energética en casa ha sido uno de los avances que la domótica ha ofrecido y así poder reducir el impacto en el medio ambiente. En la Figura 1 se puede ver un ejemplo de una aplicación móvil donde se controla el aire acondicionado según la temperatura.



Figura 1: Climatización del hogar

2.2 La seguridad en el hogar

Otro de los avances en la domótica es mejorar la seguridad dentro de las viviendas. Es uno de

los aspectos más relevantes. Se han adaptado sistemas de videovigilancia, pero este sistema ha evolucionado a tal punto que ya las cámaras de última generación nos permiten su manejo a distancia desde nuestro smartphone.

Nuestras puertas y ventanas también son un elemento fundamental de la seguridad. Hoy en día las podemos controlar con tecnología Bluetooth, controlando también la gente que entra y sale de la vivienda, o si las puertas y ventanas están correctamente cerradas o no. La Figura 2 muestra un ejemplo.



Figura 2: La seguridad siempre es primordial (1)

2.3 Electrodomésticos

Hasta ahora se ha hablado de seguridad y ecologismo, pero lo cierto es que en el mundo de la domótica existen multitud de dispositivos capaces de mejorar nuestra vida y facilitarnos las tareas en casa, como la limpieza, cocina y entretenimiento (2).

2.3.1 Robots aspiradora

Estos robots nos ayudan a mantener la limpieza de la casa. A este tipo de aspiradoras se les puede especificar el territorio a limpiar. Suelen tener una autonomía bastante larga y se pueden manejar desde un smartphone o un mando a distancia. Este tipo de robots son perfectos ya que son capaces de ejecutar tareas repetitivas. El robot aspiradora más conocido en el mercado es la Roomba de IRobot, mostrado en la Figura 3.



Figura 3: Robot aspiradora Roomba de IRobot

2.3.2 Frigoríficos y lavadoras inteligentes

Electrodomésticos como las lavadoras inteligentes capaces de ser configuradas a través de un smartphone, controlando el lavado de la ropa a distancia y, como no, controlando la eficiencia energética.

En la cocina también contamos ya con hornos y frigoríficos inteligentes. Podemos ordenar desde nuestro móvil precalentar el horno de camino a casa para encontrarlo ya listo. Son capaces de avisarnos cuándo es necesaria una limpieza y qué sistema es el más adecuado. En cuanto a los frigoríficos, ya no solo enfrían, sino que también son capaces de avisarnos sobre los productos que faltan o sobre la caducidad de alguno. Muchos de ellos ya disponen de una pantalla colocada en la puerta para poder hacer la compra on-line. En la Figura 4 vemos un ejemplo de una cocina inteligente.

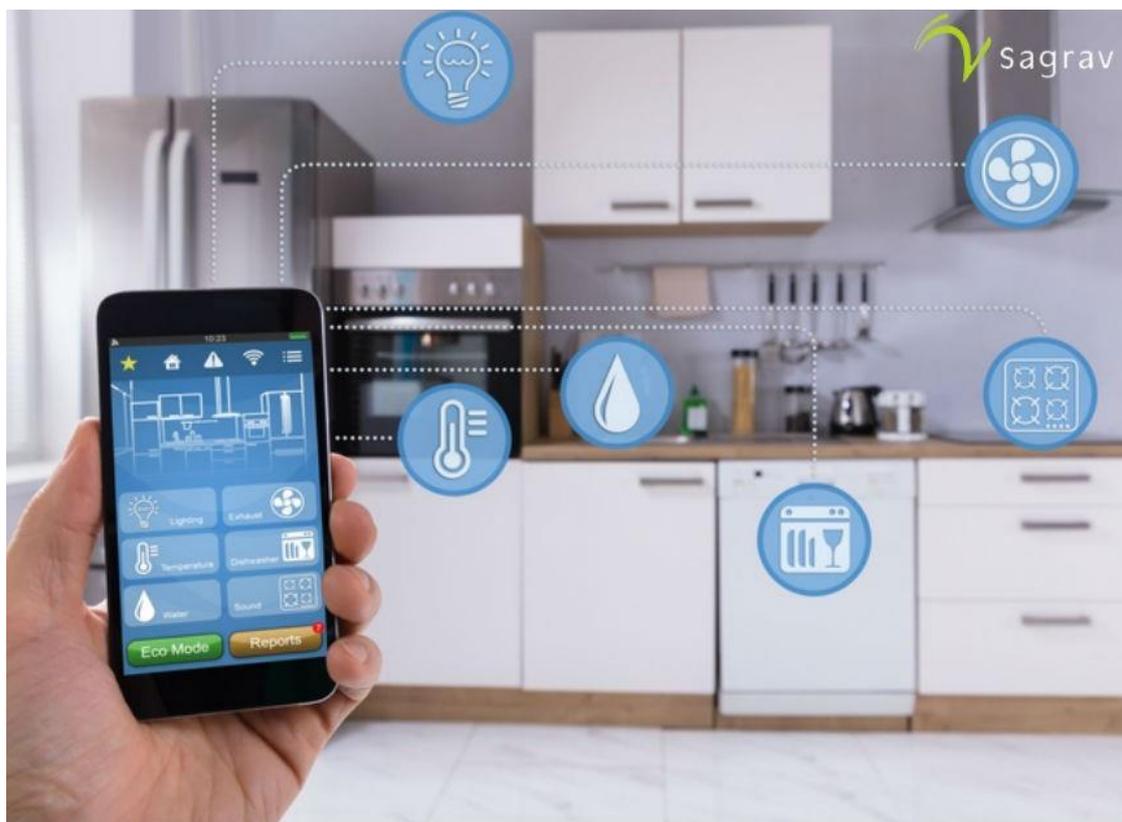


Figura 4: Cocina inteligente

2.4 Altavoces inteligentes

La inteligencia artificial ya ha llegado a nuestras casas en forma de altavoz. Este tipo de altavoces son agentes capaces de comunicarse con el usuario mediante voz y audio. Son capaces de mantener una conversación fluida y están diseñados para que respondan a las peticiones de los usuarios, como reproducir música, informar sobre el tiempo, informar sobre las noticias, etc. Pero también son capaces de controlar los dispositivos en el hogar, como apagar o encender la luz, apagar o encender el aire acondicionado o el calefactor (3).

Los altavoces más conocidos en el mercado son Google Home y Alexa (4).

2.4.1 Google Home

Google lanzó su gama de altavoces en castellano en el año 2018, pero previamente salió al mercado en Estados Unidos en el año 2016. Permite comunicarnos con su agente llamado Google Assistant y así hacerle peticiones mediante voz. Se pueden ver algunos de sus modelos en la Figura 5



Figura 5: Altavoces inteligentes Google Home, Google Home mini y Google Home Max

2.4.2 Alexa

Alexa, creado por Amazon, fue el primer altavoz inteligente en llegar al mercado. Actualmente está disponible en varios idiomas como inglés, alemán, japonés, francés, italiano y español. Es capaz de controlar dispositivos que sean compatibles. La diferencia que tiene con los altavoces de Google es su increíble calidad de sonido. Es por eso que en Estados Unidos sigue teniendo tanto éxito. Aparte de ser capaz de responder a las peticiones de los usuarios, es capaz de contar chistes, ayudar a dormir o hacer matemáticas.

También ofrece una amplia gama de diseño en sus altavoces como se muestra en la Figura 6.



Figura 6: Amazon Alexa

Capítulo 3

Preliminares

3.1 Definición

Como ya se ha explicado en el apartado anterior, donde se definen los objetivos de este proyecto, se ha desarrollado un agente que nos permite automatizar diferentes tareas del hogar. En este caso controlar la temperatura y humedad de una de las zonas de la casa.

Este proyecto está formado por una aplicación móvil desarrollada para Android, una Action para Google Assistant y Firebase como base de datos.

Se necesitó el uso de dos variables y dos dispositivos como:

- La temperatura
- La humedad
- El aire acondicionado
- La calefacción

Para el uso de esta aplicación no es necesario que el usuario se registre en ningún momento.

Las peticiones que puede realizar un usuario son:

- Por comando de voz o entrada de texto
 - Pedir la temperatura actual de la habitación donde se encuentre alojado el sensor.
 - Pedir la humedad actual de la habitación donde se encuentre alojado el sensor.
 - Pedir encender el aire acondicionado automáticamente.
 - Pedir encender la calefacción automáticamente.
- A través de la aplicación móvil
 - Permitir la conexión directa con el sensor que está alojado en una de las habitaciones de la casa.
 - Botón “Scan” para poder escanear y capturar los datos que este sensor emite constantemente y así poder almacenarlos en una base de datos para su posterior consulta.

- Botón “Stop” para poder parar la conexión y el escaneo con el sensor.

3.2 Conceptualización

En la Figura 7 se pueden visualizar las diferentes partes que componen el sistema, tal y como se explicó en el apartado anterior. Podemos distinguir: la aplicación móvil, siendo utilizada por el usuario; el agente de Google Assistant, que permite al usuario la interacción mediante comandos de voz; Dialogflow que procesa la petición y envía una petición o request; y un proyecto de Firebase cuya función es actuar como servidor y base de datos.

El agente de Google Assistant se comunica con Firebase a través de una petición o “request” que éste realiza, se hace una consulta a la base de datos y devuelve lo que se le ha pedido para completar la acción. De esta manera el usuario podrá hacer peticiones y hacer consultas a la base de datos para luego obtener la respuesta inmediatamente.

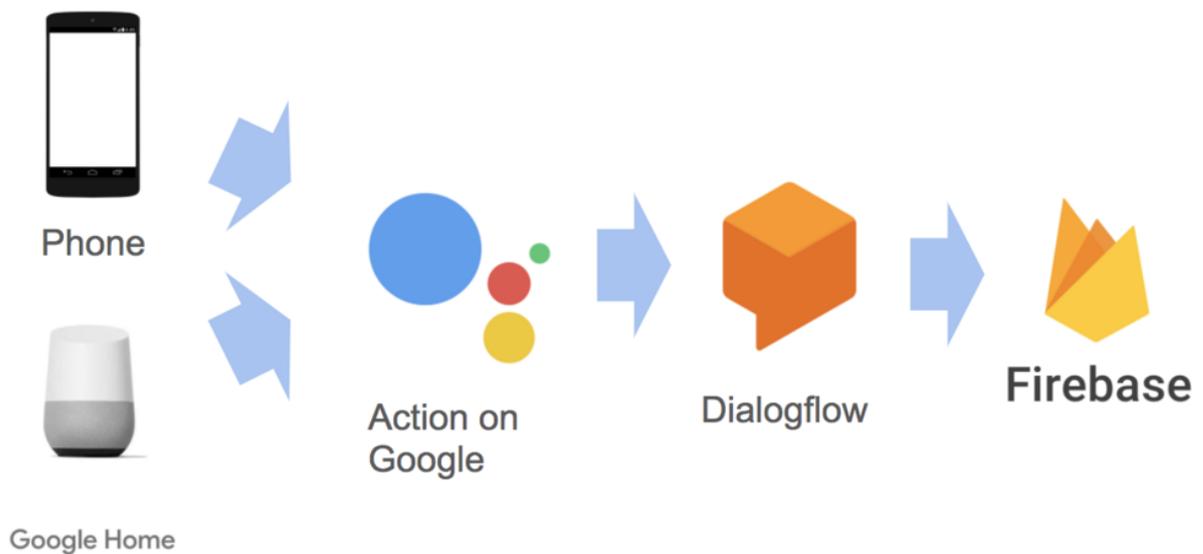


Figura 7 : Funcionamiento de las peticiones a Google Assistant

Cuando el usuario pronuncia las palabras adecuadas para invocar una acción (“Ok Google!, Talk to...”) ocurren varias cosas:

1. Google Assistant se encarga de transcribir la voz a texto. Ahora el agente está preparado para recibir las peticiones del usuario.
2. Seguidamente, Google Assistant invoca la acción solicitada.
3. Dialogflow lanza el mensaje de bienvenida, que hemos personalizado para que pronuncie el nombre del usuario y se reproduzca en nuestro móvil por voz (“Hello Alex! What can I

do for you today?”).

4. La acción se está ejecutando y está a la espera de una tarea a realizar.
5. El usuario realiza la petición mediante voz (“What is the temperatura in the room?” o “What is the humidity in the room?”) y es ahora Dialogflow quien identifica y procesa la petición.
6. Webhook, que es quien contiene el código desplegado como una Cloud Function de Firebase, realiza el fulfillment. Es decir, hace una consulta a la base de datos y completa la acción. Estos conceptos serán explicados más detalladamente más adelante en el Capítulo 4: Tecnologías utilizadas.
7. Cuando esta consulta es realizada, Firebase devuelve una respuesta, que son los valores de la temperatura y la humedad de la habitación. Google Assistant recibe dichos valores y nos responde con una frase (“The room temperature is ... degrees” o “The humidity in the room is ... percent”) que nuestro móvil reproduce por voz.
8. El código está realizado de tal forma que cuando la temperatura alcance un valor máximo indicado, se encienda automáticamente el aire acondicionado, sin necesidad de que el usuario tenga que hacerlo. Y, por otra parte, si la temperatura alcanza un valor mínimo o la humedad alcanza un porcentaje indicado, la calefacción se encenderá automáticamente.

En la Figura 8 se pueden ver los pasos explicados anteriormente de forma visual.

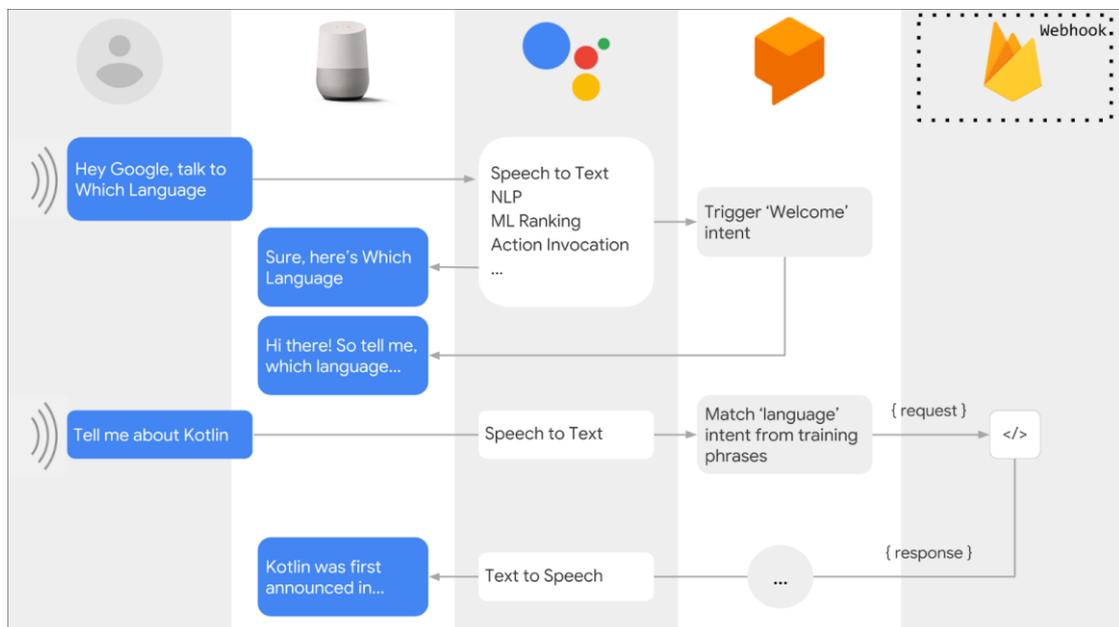


Figura 8 : Funcionamiento de la Action de Google Assistant

Capítulo 4

Tecnologías utilizadas

En este capítulo se describirán y explicarán las diferentes tecnologías utilizadas para llevar a cabo este proyecto: Se han agrupado de la siguiente manera:

1. Aplicación móvil.
2. Action de Google Assistant.
3. Backend y Base de datos.
4. Beacon
5. Otros

4.1 Aplicación móvil

- **Android**

Android es un sistema operativo desarrollado por Google.

“Cuando un dispositivo, además de para trabajar, sirve para hacerte la vida más fácil, es porque Android lo hace posible. Es por eso que tu GPS evita el tráfico, tu reloj puede escribir mensajes y el Asistente puede responder a tus preguntas. Es el sistema operativo que se utiliza en 2500 millones de dispositivos activos. Android siempre está presente en todos ellos, desde dispositivos 5G a tablets” (5).

Android nos ofrece una gran variedad de dispositivos, más formas de crear nuestras propias aplicaciones e innovación gracias a Google. Android también nos ofrece la mejor protección de datos para nuestro sistema operativo.

Los dos sistemas operativos móviles que dominan el mercado actualmente son iOS de Apple y Android de Google. Android lanzó su primera versión beta en el año 2007, mientras que Apple lanzó su primera versión beta del iPhone en el año 2008.

A continuación, se muestra una gráfica de las estadísticas mundiales de ventas entre el sistema operativo Android y el sistema operativo iOS (Figura 9).

Según un artículo de pcworld.es publicado en febrero del 2019, durante el período comprendido entre enero de 2018 y enero de 2019, Android representó el 74% del total de ventas en el mercado. En comparación, iOS domina apenas un 22% del mercado, que básicamente deja un porcentaje extremadamente bajo a otros sistemas operativos. El 2% restante representa el resto de los sistemas operativos.

Como se puede observar Android tiene una clara ventaja frente al sistema operativo iOS.

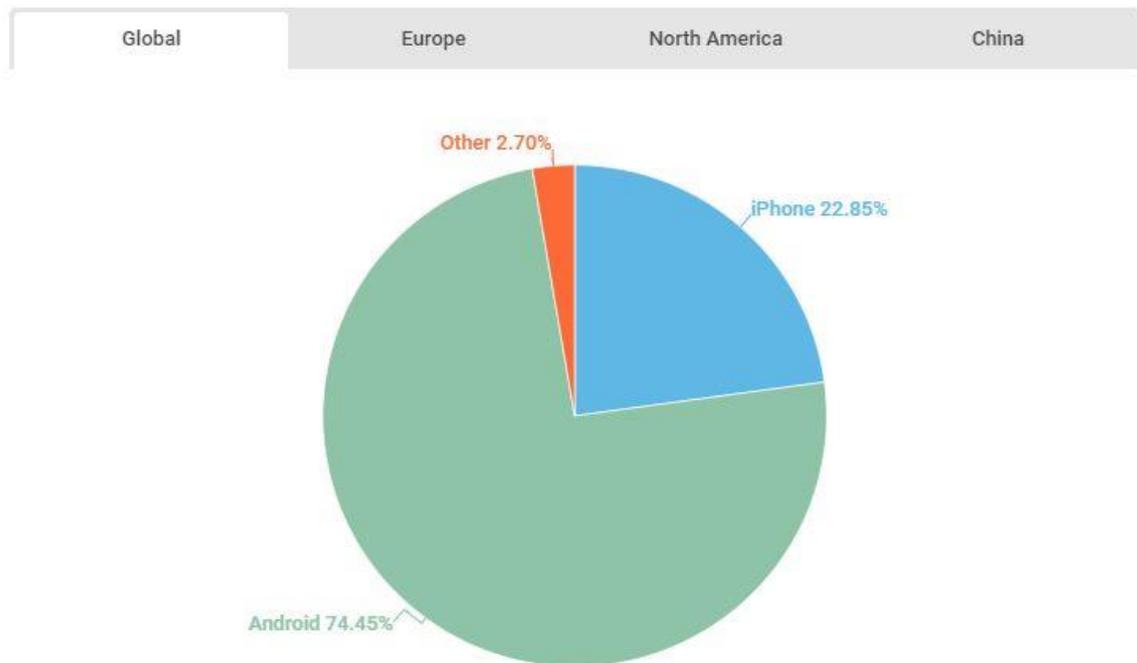


Figura 9: iPhone vs Android: cuota de mercado (6).

- **Android Studio**

Android nos ofrece un IDE (entorno de desarrollo integrado) para poder desarrollar nuestras propias aplicaciones Android. Cuenta con una gran variedad de herramientas para hacernos el entorno mucho más sencillo a la hora de escribir código, como su herramienta de refactorización de código Java.

Las herramientas de diseño de interfaces de usuario son sencillas de usar y también está disponible la documentación para desarrolladores, que es bastante amplia.

Una de las principales ventajas, son sus emuladores, como se muestra en la Figura 10, que nos da la posibilidad de probar lo que se está haciendo. Nos proporciona una gran variedad de versiones del sistema operativo y de tamaños de pantalla.

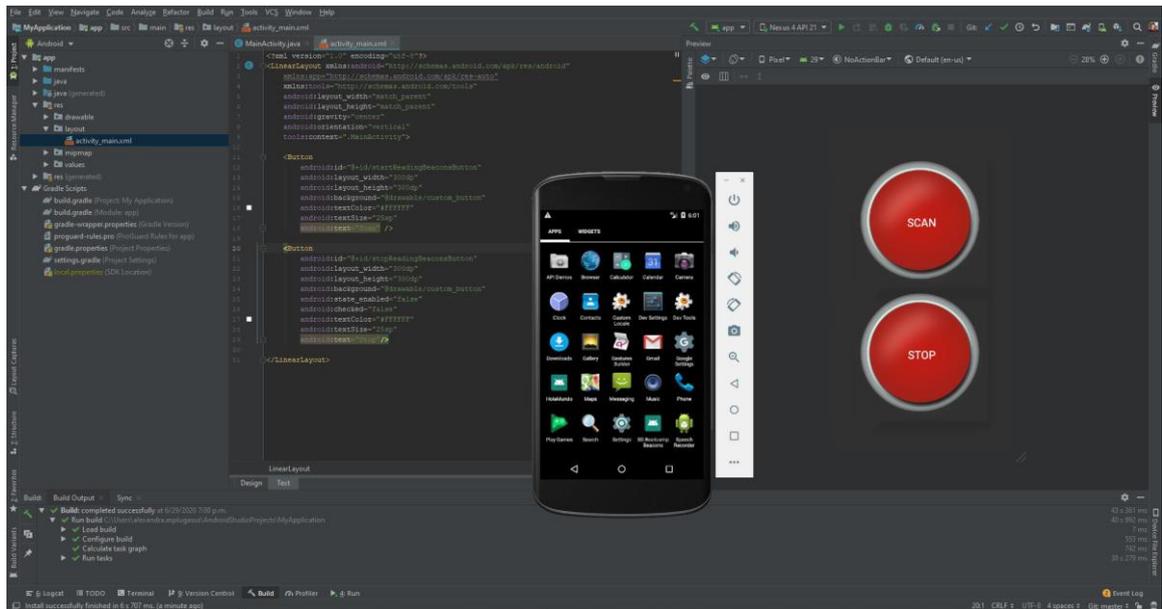


Figura 10 : Interfaz de la herramienta Android Studio

- **Java**

Android Studio utiliza el lenguaje Java para desarrollar sus aplicaciones. Java es un lenguaje de programación orientado a objetos. Es rápido, seguro y fiable. Java se convierte a partir del año 2012 en uno de los lenguajes de programación más usados y es por eso que es tan popular.

- **XML**

Para el diseño de interfaces se utiliza el lenguaje XML, un ejemplo de código en la Figura 11. Aunque Android Studio nos ofrece la posibilidad de diseñar la interfaz de usuario de manera rápida y sencilla mediante plantillas o “layouts”, también se tiene la opción de escribir directamente el código.

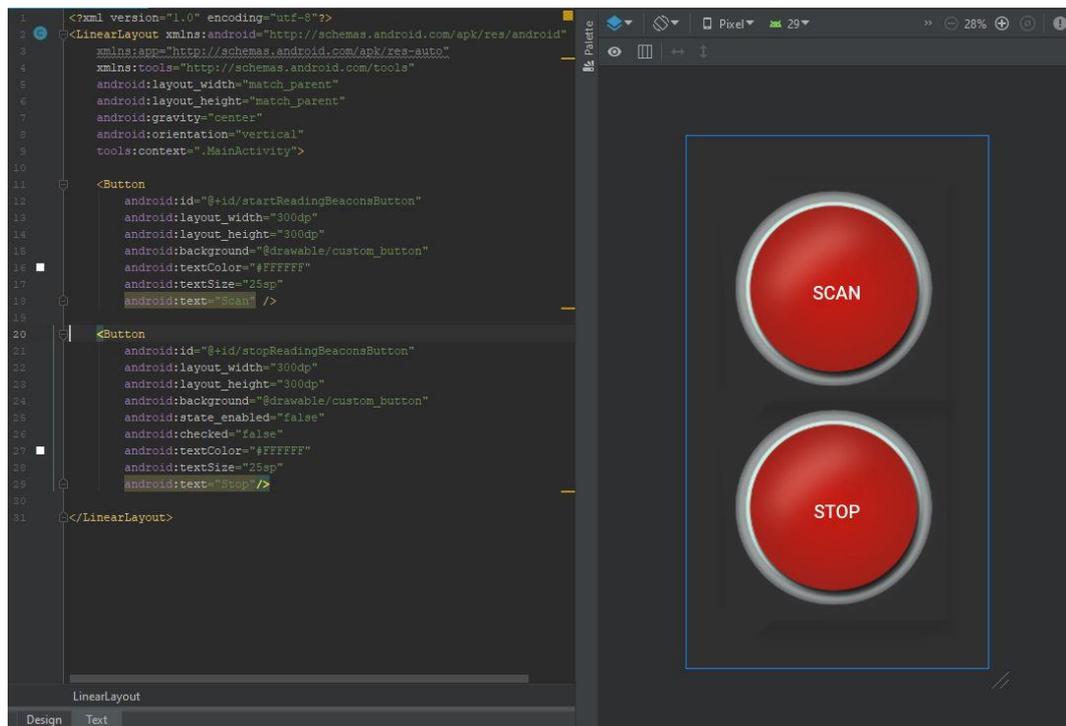


Figura 11 : Código XML y la visualización de la interfaz de la aplicación

4.2 Action de Google Assistant

- **Google Assistant**

Google Assistant es un asistente virtual desarrollado por Google. Se muestra la aplicación móvil en la figura 12. Está desarrollado con inteligencia artificial y es capaz de participar en conversaciones bidireccionales, lo que garantiza una interacción mucho más efectiva y real. Es por esto que es tan sencillo de utilizar. Este asistente está disponible para utilizarlo en una gran variedad de dispositivos como móviles, tablets, altavoces inteligentes, relojes, televisores o cualquier dispositivo que se pueda conectar vía bluetooth.

Es capaz de hacer búsquedas por voz en Google y ayudar al usuario a encontrar información en internet de manera rápida y sencilla. Tan solo tienes que activarlo diciendo (Ok Google! ...), realizar la pregunta y esperar a que este ejecute la acción.

Según un artículo publicado en rockcontent.com en febrero de 2019, la compañía de analíticas Loup Ventures lo catalogó como el asistente virtual más fiable de 2018 al responder de forma correcta el 88% de las peticiones de los usuarios en las búsquedas de voz a través de Google Home (7).

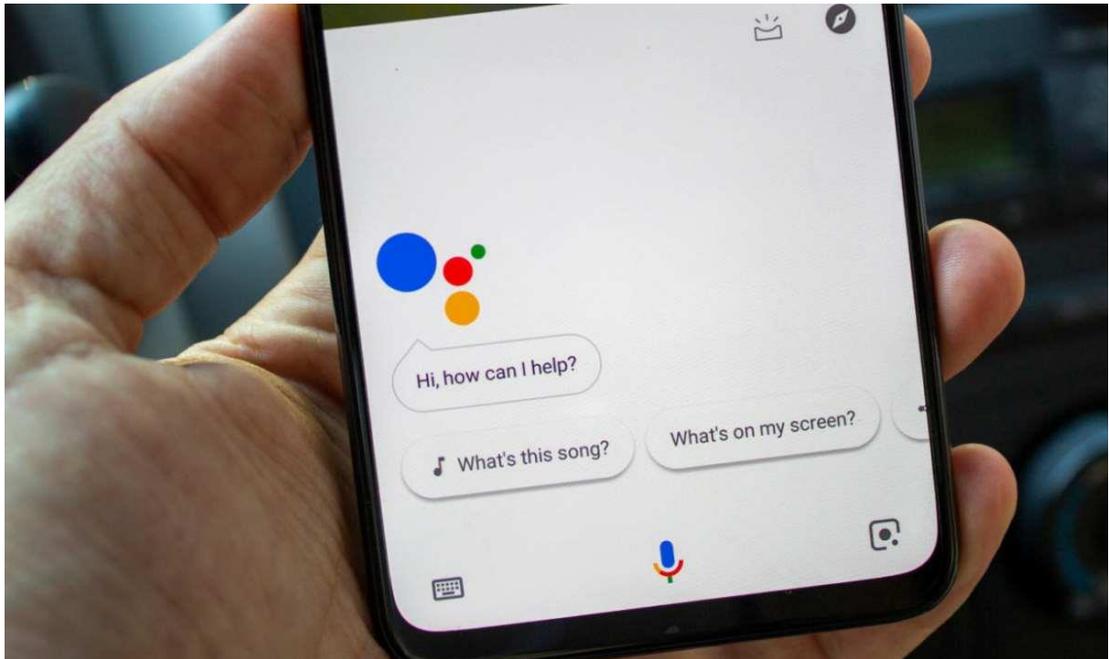


Figura 12 : Asistente de Google desde tu móvil

- **Actions on Google**

Es una plataforma que nos permite desarrollar aplicaciones haciendo uso de las funcionalidades de Google Assistant. Las Actions son aplicaciones encargadas de procesar las peticiones de los usuarios y de ejecutar la acción que se solicita. Esa acción puede ser realizar una búsqueda en la web, abrir una aplicación o hacer una consulta a alguna base de datos.

Nos permite crear Actions para diferentes entornos como extensiones para aplicaciones móviles o para altavoces inteligentes de Google.

Una Action de Google Assistant es un punto de entrada que creamos para interactuar con el asistente. Los usuarios pueden usar tanto la entrada de texto como de voz para invocar nuestra Action y poder iniciar una conversación. Para ello solo tienen que decir “Ok Google! Talk to ...” y a continuación el nombre de nuestra Action.

- **Dialogflow**

Google también nos ofrece una herramienta llamada Dialogflow, que nos permite crear y diseñar una interfaz para el usuario con inteligencia artificial, donde puede participar en conversaciones de lenguaje natural. Estas conversaciones se llaman “intents”.

Los intents clasifican un turno de conversación. Para cada agente, se definen muchos intents, que combinados crean una conversación completa.

- **Webhook**

La finalidad de este agente es hacer una consulta a nuestra base de datos. Para realizar dicha consulta y completar la acción necesitamos hacer uso de Webhook, que no es más que un servicio que nos proporciona Dialogflow. Este servicio hará de intermediario entre Dialogflow y la base de datos. Webhook acepta solicitudes en formato JSON y muestra respuestas en formato JSON. Lo interesante de este servicio es que nos permite crear una autenticación para que los usuarios autorizados puedan realizar solicitudes. Aunque para la realización de nuestro agente no fue necesario hacer uso de la autenticación.

- **Cloud Functions for Firebase**

Para terminar de configurar nuestra Action, necesitamos configurar el entorno para implementar en “Cloud Functions for Firebase”. Es un framework que nos permite ejecutar de forma automática el código en respuesta a las solicitudes. Para su configuración necesitamos las herramientas NodeJs, JavaScript y Firebase CLI que se describirán a continuación.

- **NodeJs y JavaScript**

Necesitamos hacer uso de NodeJs, que es un entorno que nos ayuda a ejecutar código de JavaScript en tiempo real para la capa del servidor. JavaScript es un lenguaje de programación orientado a objetos que se ejecuta en el lado del cliente.

- **Firebase CLI**

Para poder administrar, implementar y visualizar proyectos de Firebase, que es la base de datos que usamos para nuestro proyecto y que explicaremos más detalladamente en el siguiente punto, hacemos uso de la herramienta Firebase CLI. De esta manera, podemos crear un Fulfillment, que conectará nuestra Action de Dialogflow con nuestro proyecto en Firebase.

4.3 Backend y Base de datos

- **Realtime Database**

Otro de los objetivos de esta aplicación es almacenar en una base de datos dos parámetros que serán en este caso la temperatura y la humedad.

Como se comentó anteriormente, se hace uso de Firebase. Esta herramienta es perfecta para la finalidad de este proyecto, ya que, entre otras funcionalidades que ofrece, también nos proporciona una base de datos llamada Realtime Database y Backend, lo que permite que la información de la aplicación sea almacenada en la nube de Firebase de manera sincronizada e instantánea.

En la Figura 13 se muestra la base de datos de nuestro proyecto, donde nuestro agente irá a consultar nuestras peticiones.

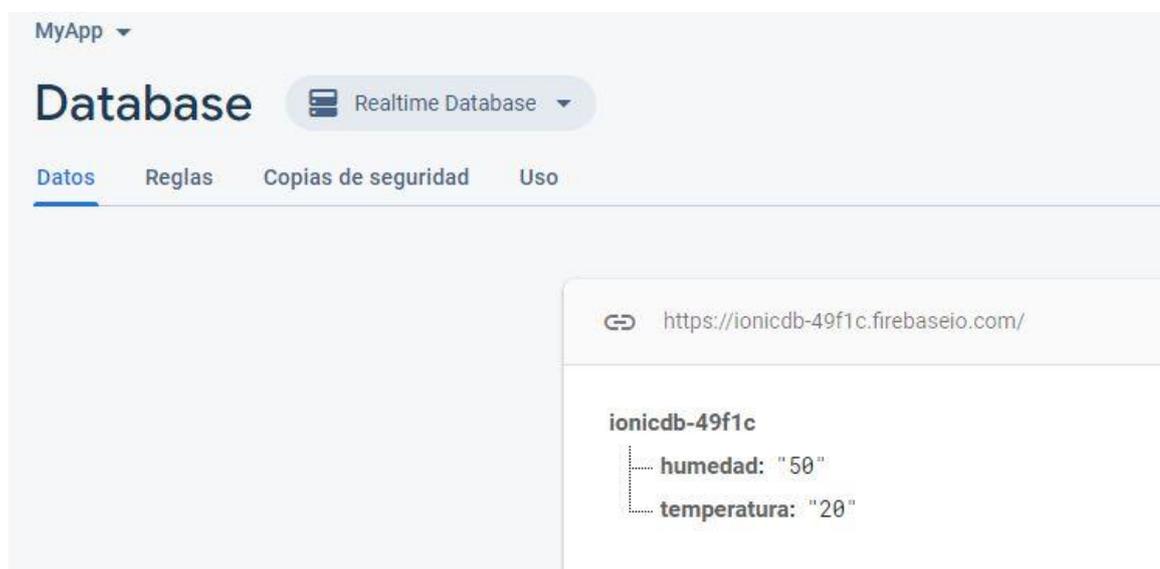


Figura 13 : Proyecto de Firebase haciendo uso de la herramienta Realtime Database

4.4 Dispositivo bluetooth beacon

- **Tarjeta beacon**

Como ya se explicó anteriormente, esta aplicación se encarga de leer los datos que emite un sensor de temperatura alojado en una de las habitaciones de la vivienda. Al no poder conseguir tal sensor, se reemplazó por una tarjeta beacon que hace la misma funcionalidad, con la diferencia de que los datos que este emite los podemos modificar manualmente y así poder hacer pruebas con datos concretos.

Una tarjeta beacon no es más que un dispositivo que alerta a otros dispositivos, como móviles o tablets, de su presencia a través de Bluetooth Low Energy (BLE). Se encarga

de emitir unos datos cada cierto tiempo. Si nuestro dispositivo móvil está lo suficientemente cerca, es capaz de capturar estos datos y almacenarlos, en este caso en la base de datos. En la Figura 14 se muestra una imagen del dispositivo utilizado para este proyecto.



Figura 14 : Dispositivo bluetooth beacon

4.5 Otros

- **GitHub**

Es un sistema de control de versiones que hemos elegido para almacenar nuestro proyecto en la nube, registrar los cambios realizados tanto en la aplicación Android como en la Action desarrollada. Es también posible integrarlo con Android Studio.

Capítulo 5

Aplicación móvil

En este apartado se explicarán detalladamente todos los puntos relacionados con el desarrollo y el diseño de la aplicación móvil.

Para que esta aplicación fuera compatible con el mayor número de dispositivos Android posibles, se utilizó la versión Android 4.4 (KitKat). De este modo, todos los sistemas operativos cuya versión sea igual o superior a esta versión serán compatibles.

Esta aplicación se desarrolló con diferentes versiones de Android Studio, ya que durante su desarrollo se han publicado varias versiones nuevas. Finalmente, la aplicación fue desplegada en junio de 2020 con la versión Android Studio 3.5.2.

5.1 Diseño

Para el diseño de esta aplicación se eligió el tema o “theme” llamado Darcula, proporcionando un diseño oscuro. Todos estos valores están definidos en el fichero de estilos style.xml y los colores utilizados en el fichero colors.xml.

El diseño de esta aplicación no es muy compleja. Se compone de una sola pantalla con dos botones.

Como se puede observar en la Figura 15, se han elegido tres imágenes del mismo botón y cada una de ellas representa un estado diferente.

La imagen de la izquierda representa el botón por defecto, la imagen del centro representa al botón desactivado y por último la imagen de la derecha representa el botón pulsado. Estos son los tres estados que puede adoptar un botón.



Figura 15 : Los tres estados de los botones

Al iniciar nuestra aplicación, se visualizarán dos botones. Uno con el estado por defecto llamado “Scan” y otro con el estado también por defecto llamado “Stop” sobre un fondo oscuro como se puede ver en la Figura 16. A medida que se vayan pulsando los botones, estos van cambiando de estado, y mientras uno está disponible el otro permanece desactivado y así sucesivamente.

Como se puede observar también en la Figura 16, no hizo falta hacer uso de una barra de inicio o “tool bar”. Se utilizó una etiqueta en el fichero de los estilos para esconder esta barra y así tener más espacio en la pantalla y tener una imagen más limpia de la aplicación.

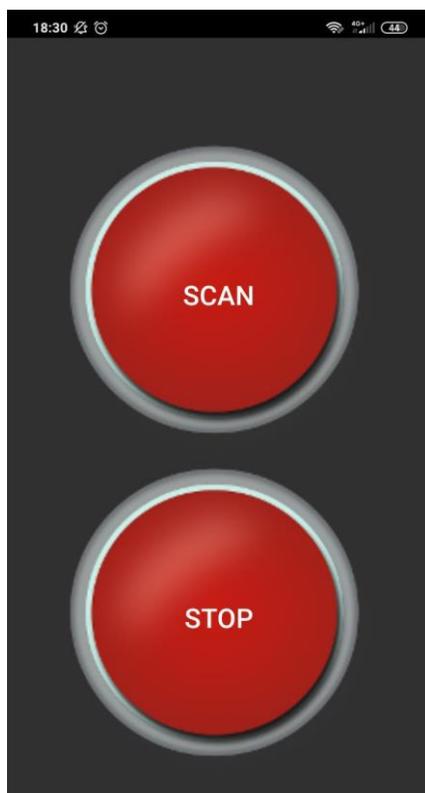


Figura 16 : Visualización de la aplicación al iniciar

5.2 Funcionalidades

A continuación, se explicarán las diferentes funcionalidades de la aplicación.

- **Acceso a la aplicación**

Al acceder lo primero que veremos serán dos botones en su estado por defecto sobre un fondo oscuro, uno para escanear y otro para parar.

- **Escaneo de datos**

Como ya se ha explicado anteriormente, la aplicación es capaz de escanear datos de otro dispositivo que está en sus proximidades, por lo que también necesitará el uso del Bluetooth. Para comenzar lo único que tenemos que hacer es pulsar el botón “Scan” y

nos aparecerá en pantalla un mensaje diciendo que nuestra aplicación quiere hacer uso del bluetooth y el botón “Allow” para activarlo, como se muestra en la Figura 17.



Figura 17 : Activando el bluetooth del dispositivo

A continuación, se inicia la conexión con la tarjeta beacon y nos informa con un mensaje en pantalla. Como también se puede observar en la Figura 18, el botón previamente pulsado pasa de su estado por defecto a un estado desactivado. De este modo solo podemos pulsar el otro botón para interrumpir la conexión. Es ahora cuando nuestra aplicación empieza a buscar otro dispositivo en sus cercanías para así poder capturar los datos que se emiten.

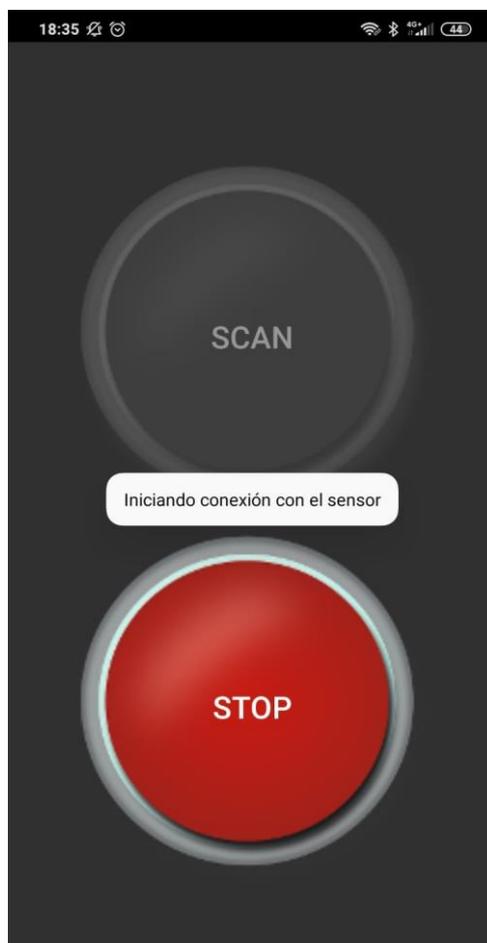


Figura 18 : Iniciando conexión con el sensor

- **Actualización de los datos**

La aplicación también nos avisará en el caso de que encuentre un dispositivo con otro mensaje por pantalla, en nuestro caso la tarjeta beacon, que ya hemos dicho anteriormente que sustituirá a nuestro sensor de temperatura. Es en este momento cuando se empiezan a capturar los datos emitidos por la tarjeta beacon para insertarlos en nuestra base de datos de Firebase. Como podemos ver en la siguiente Figura 19, la aplicación nos avisa de la actualización de dichos datos. La tarjeta beacon contiene un array de datos, la primera posición corresponde al valor de la temperatura y la segunda posición corresponde al valor de la humedad.



Figura 19 : Proceso donde se insertan los datos en la base de datos

- **Finalización de la conexión**

Este dispositivo seguirá funcionando hasta que nosotros interrumamos su conexión con el dispositivo pulsando el botón “Stop”. Como vemos en la siguiente Figura 20, al pulsar dicho botón su estado vuelve a cambiar y a quedar deshabilitado para dejar el botón “Scan” habilitado para un nuevo escaneo.

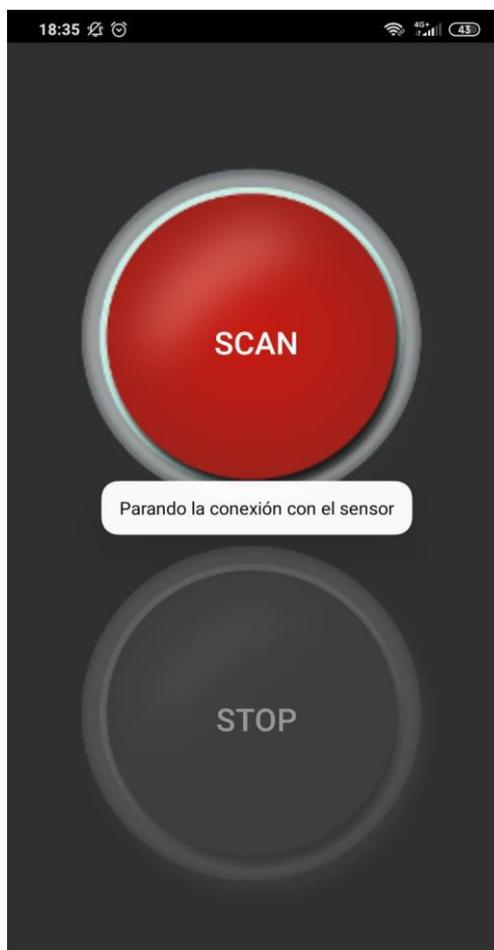


Figura 20 : Interrumpiendo la conexión con el dispositivo

5.3 Actividad principal

El objetivo de esta aplicación es detectar un dispositivo para luego capturar los datos emitidos por este. La tarjeta beacon emitirá un array con dos valores que luego esta aplicación insertará en la base de datos. La primera posición será el valor de la temperatura y la segunda posición será el valor de la humedad. Como se puede ver en la Figura 21, se ha hecho un “debug” del código para ver exactamente que es lo que está haciendo la aplicación paso a paso, imprimiendo los mensajes por consola. En la primera línea se ve como detecta el dispositivo llamado iBKS Card Beacon. La segunda línea muestra la distancia a la que se encuentra el dispositivo. Luego se visualizan los

valores en hexadecimal de la temperatura y la humedad.

Esta aplicación también se encarga de parsear estos valores de hexadecimal a decimal para luego poder insertarlos en la base de datos de Firebase y que sean legibles por una persona.

```
I/System.out: Beacon Card name: iBKS Card Beacon
I/System.out: Beacon Distance: 0.024339966583586465 meters
  Temperatura: 0x0000000000020
I/System.out: Beacon Card name: iBKS Card Beacon
  Beacon Distance: 0.0055451208439939205 meters
  Humedad: 0x00000000000050
```

Figura 21 : Consola de Android Studio al iniciar el escaneo de la aplicación

Capítulo 6

Action para Google Assistant

En este apartado se explicará la implementación de la Action que hemos desarrollado para Google Assistant. Se ha creado precisamente para que los usuarios puedan comunicarse con el sistema mediante voz.

Para poder hacer uso de esta Action, solo se necesita invocarla desde cualquier dispositivo que tenga instalado Google Assistant, como un smartphone con la cuenta de Google con la que fue creada dicha Action. Para que funcione en otros dispositivos sin tener sincronizada esta cuenta de Google, es necesario tenerla publicada en la tienda de Google. En este caso, lo que se hizo fue lanzar una versión Alpha, que es suficiente para tenerla disponible para veinte usuarios de Google Assistant.

El nombre de la Action es “Alexa Home Controller” y se invoca diciendo “Ok Google! Talk to Alexa Home Controller”.

Crear esta Action con Dialogflow fue relativamente sencilla, ya que cuenta con una consola de Google Assistant para poder hacer pruebas mientras se desarrolla su comportamiento y antes de lanzar su versión definitiva. En la siguiente Figura 22 se puede ver la consola integrada de Dialogflow y en la Figura 23 la consola de Google Assistant para poder hacer pruebas con nuestro Intent.

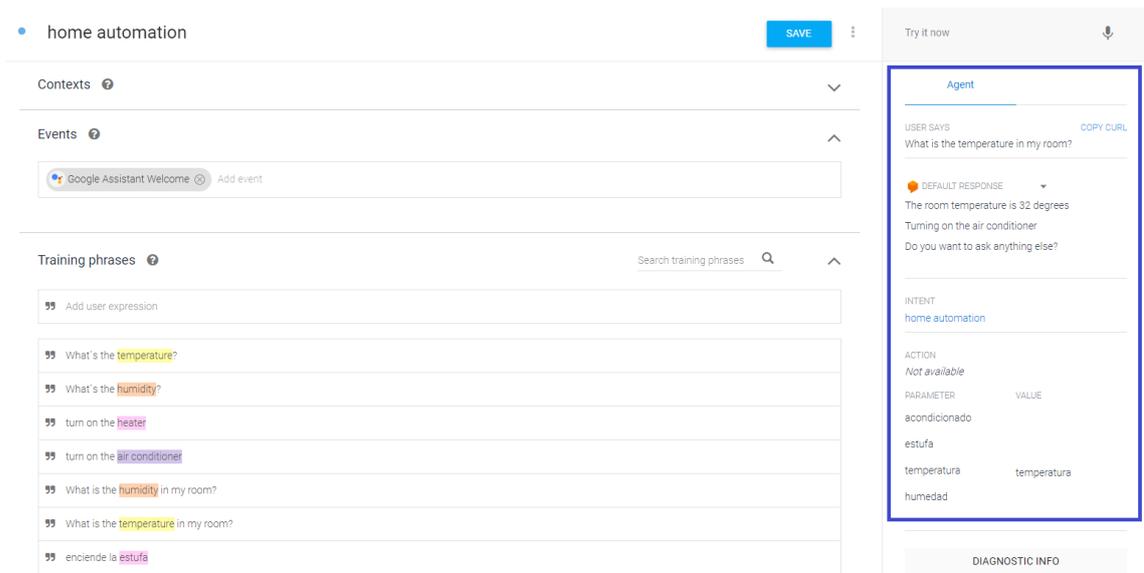


Figura 22 : Consola integrada de Dialogflow

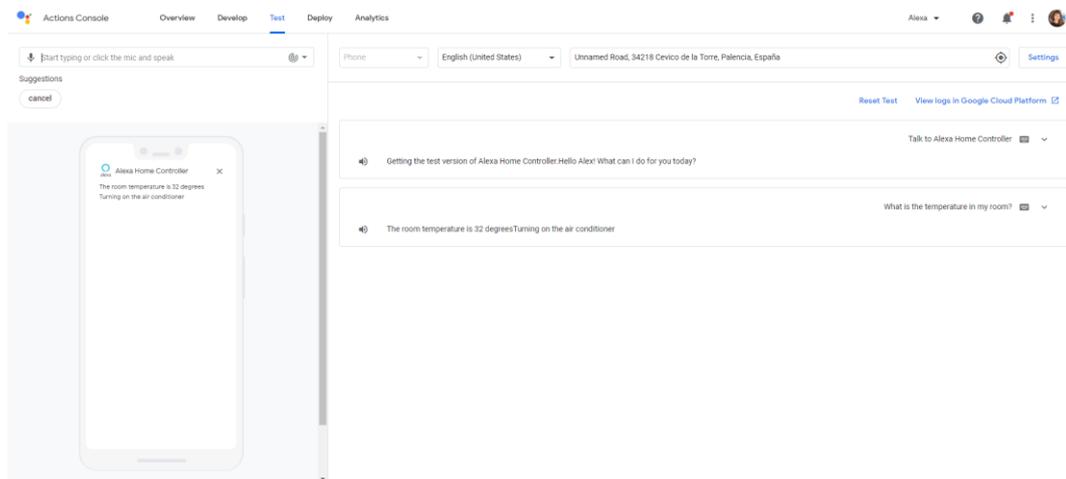


Figura 23 : Consola para simular proyectos de Google Assistant

6.1 Intent de bienvenida

Hay varias formas de crear un Intent, con webhook o con eventos del propio Google Assistant. Para el Intent de bienvenida se decidió usar la llamada a Webhook para buscar la función de bienvenida en nuestro código del fulfillment, como se muestra más adelante en la Figura 31.

Para personalizar nuestro agente, se ha creado un Intent llamado “Default Welcome Intent” que se lanza al inicio de la conversación. De este modo el usuario se asegura que el agente está escuchando y listo para ejecutar la acción. Se puede ver el ejemplo en la Figura 24.

Este Intent está configurado con varias frases que son los saludos comunes en el apartado “training phrases”.

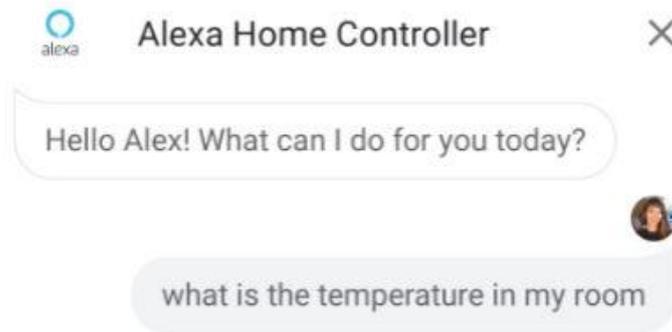


Figura 24 : Intent de bienvenida

6.2 Intent de despedida

Para finalizar la conversación se ha creado otro Intent llamado “Default Goodbye Intent”.

Para este Intent, se utilizó un evento del propio Google Assistant. De esta manera se utilizaron las dos formas de crear Intents en Dialogflow.

En este caso no hizo falta hacer uso de Webhook. Bastó con escribir la frase que queremos que nos devuelva el agente antes de finalizar la conversación en la sección “Responses” y habilitando el botón “Set this intent as end of conversation”. En la sección de “training phrases” se escriben las palabras claves que son las que se usarán para avisar al agente que queremos dar por finalizada la conversación.

Con este Intent los usuarios pueden finalizar rápidamente la conversación y evitar que no termine de forma inesperada. La aplicación se cierra cuando el usuario pronuncia las palabras que se encuentran en la sección “training phrases” de su Intent , como “Goodbye” o “Have a nice day”. Como se puede ver en la Figura 25, la Action abandona la conversación tras devolver la respuesta al usuario.

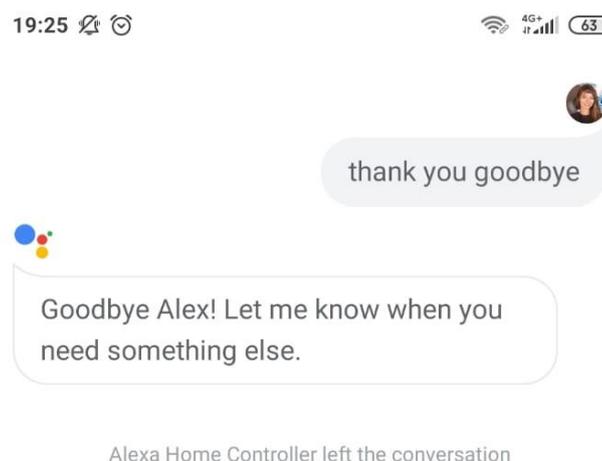


Figura 25 : Intent de despedida

6.3 Sobre la temperatura

Se ha creado otro Intent llamado “home automation” para cuando el usuario quiera hacer consultas a la base de datos. Este Intent ha sido más complejo, ya que para este caso ha sido necesario crear un fulfillment, que se explicará más detalladamente en la sección 6.7 “Despliegue del proyecto con Firebase CLI”, para poder buscar palabras claves como “temperatura”, “humedad”, “aire acondicionado” y “calefacción”, poder especificarle la ruta de la base de datos a consultar y personalizar la respuesta para cada una de las consultas.

Para las palabras claves se han usado unos parámetros que recibirá dicho código. De esta forma, cuando el usuario pronuncie una de estas palabras clave, el código recibirá dicho parámetro que necesitará para hacer la consulta adecuada en la base de datos.

También se hace uso de la sección “training phrases”, como se visualiza en la Figura 26, que activa el intent pronunciando la siguiente frase: “What is the temperatura in my room?”. En la Figura 26 también se ven las palabras claves marcadas, que son los parámetros nombrados anteriormente.



Figura 26 : Sección training phrases de la Action de nuestra aplicación

Para el caso de la temperatura, se ha personalizado de modo que cuando el valor recogido en la base de datos supere los 27°, se encienda el aire acondicionado automáticamente, haciéndoselo saber al usuario con una frase “Turning on the air conditioner”. Y cuando el valor recogido sea inferior a los 20°, se enciende la calefacción automáticamente haciéndoselo saber al usuario con la frase “Turning on the heater”. Se puede ver un ejemplo de la conversación en la Figura 27.



Figura 27 : Consulta de la temperatura

6.4 Sobre la humedad

Para el caso de la humedad, se ha personalizado de modo que cuando el valor recogido en la base de datos supere el 70%, se encienda la calefacción automáticamente, haciéndoselo saber al usuario con una frase “Turning on the heater”. También se puede ver un ejemplo de la conversación en la Figura 28.

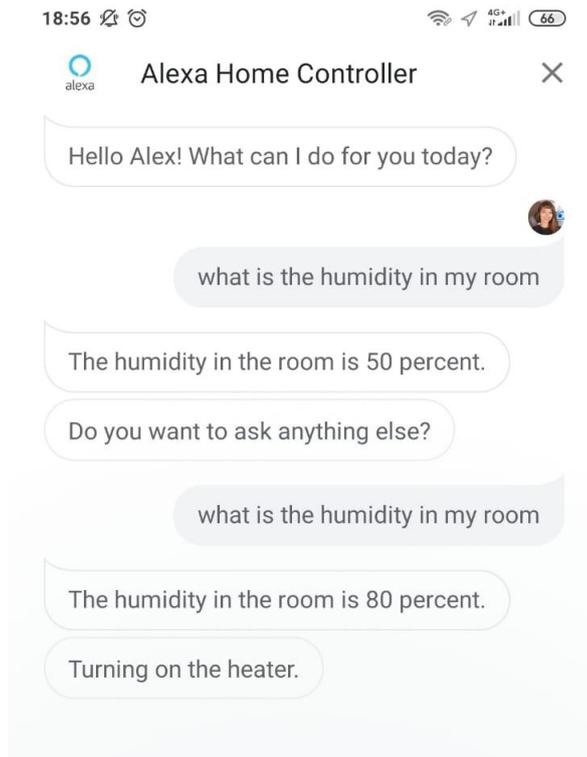


Figura 28 : Consulta de la humedad

6.5 Encender el aire acondicionado y la calefacción

También el código se ha diseñado para que el usuario pueda encender los dispositivos electrónicos, como la calefacción o el aire acondicionado de la casa, sin tener que preguntar primero la temperatura y la humedad. De esta manera también se tienen controlados a distancia. A continuación, en la Figura 29 se puede ver un ejemplo.

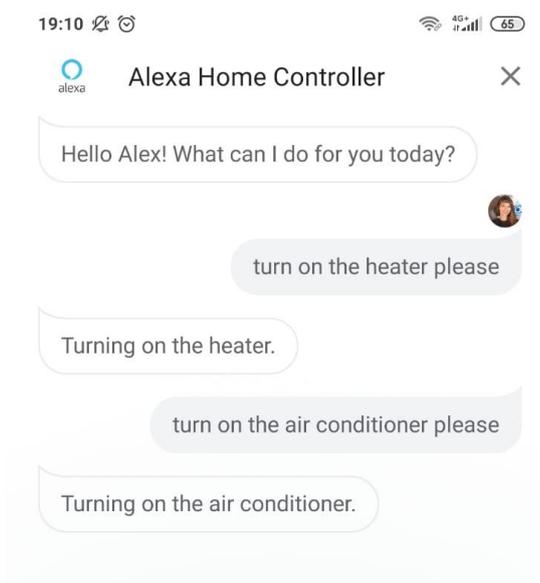


Figura 29 : Encendido de los dispositivos electrónicos de la casa

6.6 Intent de respuesta a los errores

Es muy usual en una conversación con el agente que no se entienda la frase o que el usuario no pronuncie la frase de manera adecuada. De modo que se ha creado un Intent personalizado para este tipo de errores. Cuando el agente no entienda alguna petición, éste responderá con la frase “Sorry Alex! I didn’t understand that. Can you try again?” y el usuario podrá realizar la petición de nuevo sin que se interrumpa la conversación. Se puede ver un ejemplo en la Figura 30.

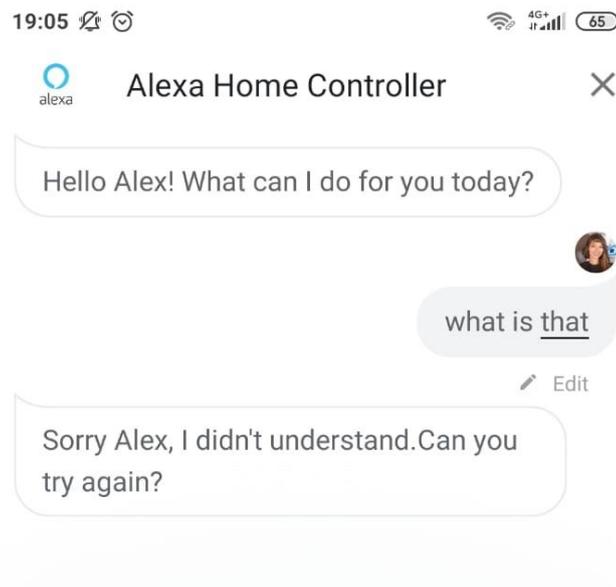


Figura 30 : Intent de respuesta a los errores

6.7 Desplegar el proyecto con Firebase CLI

Para que la herramienta Webhook, nombrada anteriormente en el apartado 6.3 “Sobre la temperatura”, termine de realizar su función, es necesario desplegar el proyecto con la herramienta Firebase CLI, ya descrita en el capítulo 4 “Herramientas utilizadas”. Solo se tendrá que escribir el código JavaScript, mostrado en las Figuras 31 y 32, con las peticiones, desplegar el proyecto para Firebase con el comando *firebase deploy* y gracias a NodeJs se ejecutará en la nube.

```

'use strict';

const {dialogflow} = require('actions-on-google');
const functions = require('firebase-functions');
const admin = require('firebase-admin');
const {WebhookClient} = require('dialogflow-fulfillment');

const app = dialogflow({debug: true});
admin.initializeApp(functions.config().firebase);

let db = admin.database();
let tempRef = db.ref('temperatura'); //Ruta a la base de datos para la temperatura
let humRef = db.ref('humedad'); //Ruta a la base de datos para la humedad

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  const agent = new WebhookClient({ request, response });

  function welcome(agent) {
    agent.add(`Hello Alex! What can I do for you today?`);
  }

  function fallback(agent) {
    agent.add(`Sorry Alex, I didn't understand.Can you try again?`);
  }

  function readFromDb (agent) {
    if(agent.parameters.temperatura) {
      return tempRef.once('value') //Esta es la consulta para la temperatura
      .then(snapshot => {
        const temp = snapshot.val();
        agent.add(`The room temperature is ${temp} degrees.`);
        //Condiciones para encender los dispositivos electrónicos de la casa
        if(temp >= "27") {
          agent.add(`Turning on the air conditioner.`);
        }

        if(temp <= "20") {
          agent.add(`Turning on the heater.`);
        }
        agent.add("Do you want to ask anything else?");
      })
      .catch(err => {
        console.log('Error', err);
      });
    }
  }
}

```

Figura 31 : Fulfillment de la Action en JavaScript imagen nº1

```
if(agent.parameters.humedad) {
  return humRef.once('value') //Esta es la consulta para la humedad
  .then(snapshot => {
    const hum = snapshot.val();
    agent.add(`The humidity in the room is ${hum} percent.`);

    if(hum >= "70") {
      agent.add(`Turning on the heater.`);
    }
    agent.add(`Do you want to ask anything else?`);
  })
  .catch(err => {
    console.log('Error', err);
  });
}

if(agent.parameters.acondicionado) {
  agent.add(`Turning on the air conditioner.`);
}

if(agent.parameters.estufa) {
  agent.add(`Turning on the heater.`);
}
}

let intentMap = new Map(); //Nombre del intent
intentMap.set('Default Welcome Intent', welcome);
intentMap.set('Default Fallback Intent', fallback);
intentMap.set('home automation', readFromDb);
agent.handleRequest(intentMap);
```

Figura 32 : Fulfillment de la Action en JavaScript imagen n°2

Al instalar el Firebase CLI y la dependencia actions-on-google, se crea una carpeta donde algunos de los ficheros son *firebase.json* y otra carpeta llamada *function* que es donde se encuentran los ficheros *index.js* y el fichero *package.json*.

Después de ejecutar el comando `firebase deploy`, por consola, como se muestra en la Figura 33, se puede ver que se crea una URL llamada “Function URL”, que es la que se usará para insertarla en la sección fulfillment de Dialogflow, como se observa en la Figura 34.

```
> firebase deploy
=== Deploying to 'ionicdb-49f1c'...

i deploying functions
i functions: ensuring required API cloudfunctions.googleapis.com is enabled...
+ functions: required API cloudfunctions.googleapis.com is enabled
i functions: preparing functions directory for uploading...
i functions: packaged functions (32.96 KB) for uploading
+ functions: functions folder uploaded successfully
i functions: creating Node.js 8 function receiveAssistantRequests(us-central1)...
+ functions[receiveAssistantRequests(us-central1)]: Successful create operation.
Function URL (receiveAssistantRequests): https://us-central1-ionicdb-49f1c.cloudfunctions.net/receiveAssistantRequests

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/ionicdb-49f1c/overview
>
```

Figura 33: Consola de Firebase CLI

⚡ Fulfillment

Webhook ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

URL*

BASIC AUTH

HEADERS

[+ Add header](#)

Figura 34 : URL de nuestro Webhook

6.8 Webhook

Como ya se ha explicado anteriormente, se hizo uso de la herramienta Webhook. Es quien realiza la consulta a la base de datos y completa la acción. Webhook acepta solicitudes en formato JSON y muestra respuestas en formato JSON.

A continuación, en la Figura 35 se puede observar un ejemplo del envío del objeto JSON al realizar la pregunta o request “What is the temperatura in my room?”. Se visualiza también el parámetro, que es la temperatura, que utiliza para realizar la consulta y el texto de bienvenida.

Raw API response **Fulfillment request** Fulfillment response Fulfillment status

```

1 {
2   "responseId": "a13d6eb8-c2d9-4e3b-b020-9724ebee7f7d-e13762d2",
3   "queryResult": {
4     "queryText": "What is the temperature in my room?",
5     "parameters": {
6       "temperatura": "temperatura",
7       "humedad": "",
8       "estufa": "",
9       "acondicionado": ""
10    },
11    "allRequiredParamsPresent": true,
12    "fulfillmentMessages": [
13      {
14        "platform": "ACTIONS_ON_GOOGLE",
15        "simpleResponses": {
16          "simpleResponses": [
17            {
18              "textToSpeech": "Hello Alex! What can I do for you today?"
19            }
20          ]
21        }
22      }
23    ]
24  }
25 }

```

Figura 35 : Fulfillment Request del servicio Webhook en formato JSON

Seguidamente, en la Figura 36 se puede ver la respuesta o “response” de Webhook, señalando el texto de respuesta, también en formato JSON.

Raw API response Fulfillment request **Fulfillment response** Fulfillment status

```

1 {
2   "fulfillmentMessages": [
3     {
4       "text": {
5         "text": [
6           "The room temperature is 20 degrees"
7         ]
8       }
9     },
10    {
11      "text": {
12        "text": [
13          "Turning on the heater"
14        ]
15      }
16    },
17    {
18      "text": {
19        "text": [
20          "Do you want to ask anything else?"
21        ]
22      }
23    }
24  ]
25 }

```

Figura 36 : Fulfillment Response del servicio Webhook en formato JSON.

Capítulo 7

Presupuesto

En esta sección se hace un breve desglose del presupuesto del proyecto desarrollado.

Descripción	Tiempo empleado	Precio/Hora (€/h)	Total (€)
Planificación del proyecto	14 días x 8 horas = 112	6	672 €
Lectura de bibliografía y documentación	6 días x 8 horas = 48 horas	6	288 €
Curso Android Studio	7 días x 8 horas = 56 horas	-	13 €
Desarrollo aplicación Android	150 días x 8 horas = 1200 horas	15	18000 €
Desarrollo Agente Google Assistant	60 días x 8 horas = 480 horas	15	7200 €
Preparación de la memoria	6 días x 8 horas = 48 horas	10	576 €
Ordenador portátil	-	-	900 €
Smartphone Xiaomi Mi 10	-	-	799 €
Dispositivo iBKS Card beacon	-	-	25 €
Total	1944 horas	-	28473 €

Tabla 1: Presupuesto del proyecto

Capítulo 8

Conclusiones y líneas futuras

La aplicación desarrollada para este trabajo de fin de grado, que se compone de un agente personalizado y una aplicación móvil desarrollada para Android cumple con los objetivos que se han propuesto inicialmente. Se ha intentado hacer una pequeña introducción al mundo de la domótica y a su utilidad en el día a día en el hogar. Es bastante sencilla de usar e incluso puede llegar a ser bastante útil para las personas con dificultad de movimiento. Cuando nuestra casa es capaz de adaptarse por sí misma a nuestras necesidades, nuestra calidad de vida mejora. Es capaz de ejecutar las tareas que son repetitivas y rutinarias, y se pueden realizar de forma automática.

En cuanto a mi experiencia personal, el desarrollo de esta aplicación me ha motivado mucho a la hora de seguir investigando sobre el mundo de la domótica y de seguir experimentando y mejorando la aplicación. He aprendido a manejar estas diferentes herramientas y he mejorado mis habilidades, ya que es la primera vez que desarrollo una aplicación así, contando con la ayuda de mi tutora y mi cotutor. Pienso que ha sido bastante útil, ya que estas herramientas son muy innovadoras y se usan bastante en el desarrollo de aplicaciones. Dialogflow y Google Assistant se usan cada vez más y gracias a ellas podemos crear agentes capaces de ayudarnos en nuestra vida diaria y a mejorar nuestra calidad de vida. Y si se tienen los suficientes conocimientos, el agente se puede personalizar a tal punto que se puede adaptar perfectamente a ti mismo y a tu rutina diaria. También he conseguido mejorar mis habilidades con la herramienta Android Studio.

Este proyecto desarrollado es un prototipo, por lo que todavía es posible añadir funcionalidades nuevas y seguir mejorándolo. Como, por ejemplo:

- Con ayuda de una placa Arduino, intentar encender y apagar una bombilla cuando el usuario lo indique por comando de voz, para poco a poco adaptarlo y conseguir controlar las luces del salón a distancia.
- Añadir avisos o alarmas cuando los dispositivos electrónicos, tales como el aire acondicionado o la calefacción, lleven un determinado tiempo encendido y avisar al usuario si desea apagarlos. Ya que una de las finalidades de la domótica es el uso eficiente de la energía en el hogar.
- Preguntar si los dispositivos están encendidos o apagados cuando el usuario se encuentre fuera de la vivienda.

Capítulo 9

Conclusions and future lines

The application developed for this project, which is made up of a personalized agent and a mobile application developed for Android, meets the objectives that were initially proposed. An attempt has been made to make a short introduction to the world of home automation and its usefulness in daily at home tasks. It is quite easy to use and can even be quite useful for people with movement difficulties. When our house is able to adapt itself to our needs, our quality of life improves. It can automatically perform repetitive and routine tasks.

Regarding my personal experience, the development of this application has motivated me a lot to continue investigating the world of home automation and to continue experimenting and improving the application. I was able to learn how to use these different tools and improve my skills in the world of application development with the help of my two tutors. I think it has been quite useful, since these tools are very innovative and are used a lot in the development of applications. Dialogflow and Google Assistant are used more and more and thanks to them we can create agents capable of helping us in our daily life and improving our quality of life. And if you have enough knowledge, the agent can be customized to the point that you can perfectly adapt to yourself and your daily routine. I have also managed to improve my skills with the Android Studio tool.

This developed project is a prototype, so it is still possible to add new functionalities and continue improving it. For example:

- With the help of an Arduino board, try to turn on and off a light bulb when the user indicates it by voice command, to gradually adapt it and manage to control the lights in the room remotely.
- Add warnings or alarms when electronic devices, such as air conditioning or heating, have been on for a certain time and notify the user if they want to turn them off. Since one of the purposes of home automation is the efficient use of energy in the home.
- Ask if the devices are on or off when the user is outside.

Bibliografía

1. Lo mejor en domótica para tu Smart home: seguridad y eficiencia energética. [Online].; 2017 [cited 2020 junio 23]. Available from: HYPERLINK "https://blogthinkbig.com/lo-mejor-en-domotica-para-tu-smart-home" <https://blogthinkbig.com/lo-mejor-en-domotica-para-tu-smart-home> .
2. Qué es Android. [Online].; 2017 [cited 2020 junio 24. Available from: HYPERLINK "https://www.android.com/intl/es_es/what-is-android/" https://www.android.com/intl/es_es/what-is-android/ .
3. iPhone vs Android: cuota de mercado. [Online].; 2019 [cited 2020 junio 24. Available from: HYPERLINK "https://www.pcworld.es/articulos/smartphones/iphone-vs-android-cuota-de-mercado-3692825/" <https://www.pcworld.es/articulos/smartphones/iphone-vs-android-cuota-de-mercado-3692825/> .
4. Asistente de Google: ¡Conoce las funciones del dispositivo y cómo utilizarlo! [Online].; 2019 [cited 2020 junio 25. Available from: HYPERLINK "https://rockcontent.com/es/blog/funciones-del-asistente-de-google/" <https://rockcontent.com/es/blog/funciones-del-asistente-de-google/> .
5. 6 electrodomésticos domóticos. [Online].; 2016 [cited 2020 junio 26. Available from: HYPERLINK "https://www.casasdigitales.com/7-electrodomesticos-domoticos/" <https://www.casasdigitales.com/7-electrodomesticos-domoticos/> .
6. Qué es domótica. [Online].; 2017 [cited 2020 junio 26. Available from: HYPERLINK "http://www.cedom.es/sobre-domotica/que-es-domotica" <http://www.cedom.es/sobre-domotica/que-es-domotica> .
7. Evolución de los asistentes de voz inteligentes. [Online].; 2018 [cited 2020 junio 26. Available from: HYPERLINK "https://virtuales.top/asistentes-de-voz-inteligentes/" <https://virtuales.top/asistentes-de-voz-inteligentes/> .
8. La evolución de los Asistentes en 8 años y para qué los utilizan los usuarios. [Online].; 2019 [cited 2020 junio 26. Available from: HYPERLINK "https://blogthinkbig.com/asistentes-virtuales-usuarios" <https://blogthinkbig.com/asistentes-virtuales-usuarios> .
9. Documentación de Firebase [11/04/2020].
<https://firebase.google.com/docs/hosting/quickstart?hl=es>
- 10.Stack overflow para resolver errores durante el desarrollo de la aplicación Android [02/05/2020].
<https://stackoverflow.com/questions/57528183/how-to-fix-error-cannot-find-module-core-js-es6>
- 11.Documentación de Android para desarrolladores [03/05/2020].
<https://developer.android.com/training/basics/firstapp/running-app?hl=es-419>
- 12.Documentación de Google Assistant para desarrolladores [20/05/2020].
<https://developers.google.com/assistant/conversational/df-asdk/deploy-fulfillment>