

uc3m | Universidad **Carlos III** de Madrid

Diseño e implementación de un asistente personal inteligente en español basado en reconocimiento de voz empleando Raspberry Pi

Grado Ingeniería Telemática

2017-2018

*Trabajo Fin de Grado*

---

Carlos Díaz Fernández

Tutor:

Julio Villena Román

9 octubre 2018, Leganés



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**



# RESUMEN

El presente Trabajo Fin de Grado tiene como objetivo el diseño e implementación de un asistente inteligente personal en español para el hogar, con el fin de facilitar acciones cotidianas simplemente con el uso de la voz del propio usuario, como realizar compras, reproducir música, realizar cualquier acción sobre el hogar o cualquier otro tipo de consultas que el usuario le realice para obtener información.

La motivación de este proyecto surge con la idea de crear una versión reducida, una demo de un asistente inteligente casero para el hogar de bajo coste, pero con las mismas prestaciones que los asistentes inteligentes ya existentes en el mercado actual como pueden ser Siri o Google Home y con el propósito de que el dispositivo diseñado sea capaz de procesar la voz y reproducir las respuestas dirigidas al usuario en español.

Se implementará un sistema realizado sobre código Python, empleando para ello una Raspberry Pi. El asistente a diseñar, debe de cumplir los objetivos básicos establecidos para su diseño que son: diseño previo del asistente inteligente en un PC portátil, realizar el diseño e implementación del primer módulo denominado Snowboy, diseño e implementación del segundo módulo llamado conversión de voz a texto, diseño e implementación del tercer módulo donde se encuentra el procesado de la consulta, diseño e implementación del cuarto modulo llamado conversión de texto a voz y por último la activación y el acoplamiento del diseño a una Raspberry Pi.

Para conseguir la solución final hubo que pasar por ciertas dificultades como la configuración de Raspberry Pi, además se ha requerido un alto nivel en programación sobre lenguaje Python que ha requerido un aprendizaje externo. Por otro lado, ha habido que adaptarse a trabajar mediante la voz. Finalmente, el objetivo del proyecto ha sido completado y el trabajo realizado ha servido para aprender nuevas técnicas e incrementar las ya estudiadas anteriormente.



# ABSTRACT

The objective of this TFG is the design and implementation of an intelligent personal assistant in Spanish for home, in order to facilitate daily actions simply by using the user's own voice. This type of device, helps users to perform tasks such as shopping, playing music, performing any action on the home or any other type of query that the user may have to obtain information from the device, such as weather queries, or queries about current society information.

Virtual assistants have a great future projection, so much so that it is estimated that their value in the market can be doubled in the coming years, even tripled in a three-year period. Therefore, companies that are dedicated to implementing these Smart wizards strive to improve the quality of the devices and design them in such a way that they are attractive to their demand.

The artificial intelligence is a market that is booming and has a great prospect for the future. The artificial intelligence is and will be even more so in all environments of human life. Thanks to it, now is possible to find projects related to health, where apart from improving the quality of this sector, the waiting lists would be considerably reduced. Also can find projects within the technology sector such as robotization and automation of tasks or in education to help both teachers and students in learning to be able to get a more focused learning in each student, being more personal.

The motivation behind this Project came with the idea of creating a reduced version, a demo of a low-cost intelligent home wizard for the home, but with the same features as the intelligent wizards already existing in the current market, such as Siri or Google Home, and with the aim that the device designed is capable of processing the voice and reproducing the answers addressed to user in Spanish, since most of the systems launched to date only support English as a language and Spanish-speakers users can't have this technology within their reach.

The assistant to design must meet the basic objectives established for this designed, which are: previous designed of the intelligent assistant on a laptop PC, design and implementation of the first module called Snowboy, design and implementation of the second module called speech to text conversion, design and implementation of the third module where the processing of the query is located, design and implementation of the fourth module called text to speech conversion, and finally the activation and coupling of the design to a Raspberry Pi.

In addition to meeting the above objectives, the device must comply with current laws that protect users from attacks on their privacy and security. Users know the power and everything that a Smart assistant can offer to humans, like do jobs instead of people, perform tasks that a human can't do, or facilitate any task both inside and outside the home. But users also know that this type of assistant Works with the personal data of the users who live inside the home, and with their voice. This is of concern to users, as they don't approve that their private information and voices being stored and recorded on a server and who may be behind the loudspeaker. That is why is important that the device ensures privacy to the user so that they increasing rely on home assistant, so they must comply with the laws on data protection, both the voice that recorded and private personal data of users.

Using Raspberry Pi 2.0, common tasks such as Reading the news, Reading your latest emails or making any kind of query can be done with ease. This miniature computer offers adequate power for this Project at a low cost compared to the Smart home assistants currently available on the market. Raspberry Pi offers the possibility of connecting USB microphones to detect the users, more powerful speaker or connecting numerous hardware items using the GPIO ports offered by the Raspberry Pi board as leds or other items. On the other hand, it also offers an Internet connection with which the user can Access servers or web pages to make the necessary queries. In addition, can also make bluetooth connections to connect with other household items such as televisions, speaker or any other device with bluetooth connectivity. Therefore, it's an appropriate option to implement the design on a platform like this one as it has qualities like those of the current intelligent assistant, but at a lower cost.

The system will be implemented on Python code. This language offers a great power to the programmer, as it has the ability to import a large number of libraries that can provide different functions for the implementation of this device as libraries related to the detection or processing of voice. On the other hand, this language, in comparison with others, has at your disposal forums and web pages where you can find base code for the implementation of any type of project or program, or you can also find doubts from other users answered by other users of the forum where they discuss what is the best option or where you can offer possible advice to improve the problems proposed or to improve the skill of the programmer.

Therefore, the solution for the design of this prototype will be based on a Raspberry Pi 2 based on Python code divided into different phases or modules to achieve the main objective of the project and obtain the expected functionality.

This group of modules will start, as well as the intelligent wizards that can be found in the market, with the recognition of a keyword or phrase called hotword for its activation, made through the Snowboy software API that provide the necessary material to create keywords for this type of devices. To do this, the tool allows the user to create a personalized model where he can choose the name he will give to his activator and must record the voice with the activation phrase on three occasions. Once the model is obtained and after the activation, the device will start to listen to any phrase from the user, if the phrase is erroneous (noise) or has occurred some unforeseen failure, the device will notify, if not the case, will have obtained the phrase with which you can make a request or a query, will be when the device performs a conversion from audio to text (made through the speechrecognition tool and more specifically the service offered by Google due to its reliability) captured by the microphone to process the phrase and begin to work on it. The device will process the captured phrase and will obtain from it the necessary information to search the solution and offer the corresponding answer addressed to the user, whether a query to a server to provide data, a query to any Internet page to offer information, carry out tasks on the home itself such as regulating the temperature, lowering or raising blinds, etc. The processing of the sentence involves recognizing the different entities within it. This response must also meet the objective of being in Spanish so the change from text to voice (made through Python tts own tool, which transforms a series of characters into voice) to reproduce the response addressed to the user will be carried out for the Spanish language.

There are four queries in the test plan developed for this project, although the device can be easily improved by creating new queries and functions or by improving existing queries. These tasks include simple tasks such as reporting the current time of the area where the device is located, as well as providing weather information for any city in the world, reporting both weather and temperature at the time. The device also has the functionality to read the main headlines of the newspaper that the user indicates through the voice. Finally, this device has the

functionality to Access the user's personal mail to read his or her mail from the inbox and let the user know if he or she has any new or unread mail and who owned the message.

As for the improvements that can be implemented in the future, could find all those related about music playback, as it is one of the functions most used by users and are most appealing to them. A future device could play music from user accounts such as Spotify, Youtube or others, play saved lists, favorite songs or search for any available platform. Also, it could include the possibility of choosing where to play the music, either on the device itself, on a bluetooth speaker, on home televisions, etc. In the future, users can use the device to find alerts about traffic or travel times. The user may have planned a journey either by car or walking, or simply a trip that is about to make, before leaving home, the user could make a query through which the device could alert of the traffic that can be found and provide an alternative route. It would also alert to the time of that new route or the route you were going to make by default.

To achieve the final solution had to go through some difficulties such as the configuration of the Raspberry Pi, the installation of numerous libraries and add-ons, also has required a high level of programming on Python language that has required external learning. On the other hand, it has been necessary to adapt to work through voice, a work that hadn't been done before.

Finally, in spite of some difficulties, the objective of the project has been completed and the work done has served to learn new techniques and increase those already studied previously.



# Índice

RESUMEN .....	I
ABSTRACT .....	III
Índice .....	IX
Índice de figuras .....	XII
Índice de tablas .....	XV
Introducción.....	1
1.1- Motivación.....	1
1.2- Objetivos.....	2
1.3- Estructura del documento .....	3
1.4- Regulación .....	3
Estado del arte .....	6
2.1- Asistentes personales inteligentes .....	6
2.1.1- Situación actual de los asistentes inteligentes.....	7
2.1.2- Google Home Assistant .....	8
2.1.3- Google Home Assistant Mini.....	10
2.2- La inteligencia artificial.....	11
2.2.1- ¿Qué es la inteligencia artificial?.....	11
2.2.2- Historia de la inteligencia artificial.....	11
2.2.3- Tipos de inteligencia artificial .....	12
2.2.4- Aplicaciones de la inteligencia artificial.....	13
2.3- Raspberry Pi .....	15
2.3.1- Historia de la Raspberry Pi .....	17
2.3.2- Modelos de Raspberry Pi .....	17
2.3.3- Algunos usos de aplicaciones en Raspberry Pi.....	17
2.4- Python.....	19
2.4.1- Características del lenguaje.....	20
2.4.2- Ventajas frente a otros lenguajes .....	20

2.4.3- Razones para el uso de Python.....	21
Diseño.....	23
3.1- Función del dispositivo.....	24
3.1.1- Funcionalidad.....	25
3.2-Modulos.....	27
3.2.1- Snowboy .....	27
3.2.2- Escuchar consultas del usuario y Speech-to-text. ....	30
3.2.3- Wit.ai.....	30
3.2.4- Text-to-speech.....	32
Implementación .....	33
4.1- Introducción y visión general .....	33
4.2- Preparación del Entorno .....	34
4.3- Instalación de Raspberry Pi .....	34
4.3.1- Instalación del sistema operativo .....	34
4.4 Instalación Snowboy.....	37
4.5- Instalación PIP.....	38
4.6- Instalación SPEECH-TO-TEXT.....	38
4.6.1- Requisitos.....	38
4.6.2- Instalación .....	40
4.7- Instalación WIT.AI.....	41
4.8- Instalación TEXT-TO-SPEECH .....	44
4.9 Consultas implementadas .....	46
4.9.1- Consulta time .....	46
4.9.2- Consulta tiempo climático .....	46
4.9.3- Consultar periódicos .....	48
4.9.4- Consulta correo electrónico .....	49
Marco socioeconómico.....	56
5.1- Análisis del entorno socioeconómico .....	56
5.2 Evolución de los asistentes virtuales y tendencias futuras .....	59
5.3- Conclusión al análisis del entorno socioeconómico .....	60
	X

Planificación y presupuesto .....	61
6.1- Planificación .....	61
6.2- Presupuesto .....	64
Conclusiones y trabajos futuros.....	66
7.1- Conclusiones.....	66
7.2 Trabajos futuros .....	67
Bibliografía.....	69

# Índice de figuras

Figura 1-1 Reglamento General de Protección de Datos (RGPD) .....	5
Figura 2-1 Logo Cortana    Figura 2-2 Logo Siri .....	8
Figura 2-3 Altavoz Google Home .....	9
Figura 2-4 Altavoz Google HomeMini .....	10
Figura 2-5 Principales aplicaciones AI.....	15
Figura 2-6 Placa Raspberry Pi .....	16
Figura 2-7 Logo Python.....	19
Figura 3-1 Diseño asistente inteligente .....	24
Figura 3-2 Diagrama del dispositivo.....	24
Figura 3-3 Respuesta dispositivo a consulta time .....	25
Figura 3-4 Respuesta dispositivo a consulta clima .....	26
Figura 3-5 Respuesta dispositivo a error en consulta lectura de correos.....	26
Figura 3-6 Respuesta dispositivo a consulta lectura de correos .....	26
Figura 3-7 Respuesta dispositivo a consulta lectura de titulares .....	27
Figura 3-8 Logo Snowboy .....	27
Figura 3-9 Creación palabra clave.....	28
Figura 3-10 Grabación palabra clave.....	29
Figura 3-11 Comando ejecución del dispositivo.....	29
Figura 3-12 Logo Wit.ai .....	31
Figura 4-1 Logos sistemas operativos Raspberry Pi .....	35
Figura 4-2 Instaladores sistemas operativos Raspberry Pi .....	35
Figura 4-3 Escritorio Raspberry Pi.....	36
Figura 4.4 Comando instalación pyaudio.....	37
Figura 4.5: Comando instalación portaudio .....	37
Figura 4.6: Comando instalación pip.....	38

Figura 4.7: Comando instalación paquetes pip.....	38
Figura 4.8: Comando comprobación micrófonos activos .....	39
Figura 4.9: Comando activar micrófono USB.....	39
Figura 4.10: Comando instalación Speech Recognition.....	40
Figura 4.11: Código Speech Recognition.....	41
Figura 4-12 Registro Wit.ai .....	42
Figura 4-13 : Entrenamiento Wit.ai.....	42
Figura 4.14: Código Wit.ai.....	44
Figura 4.15: Instalación Google text to speech.....	45
Figura 4.16: Importar librería Google text to speech.....	45
Figura 4.17: Código Google text to speech.....	45
Figura 4.18: Importar librería time.....	46
Figura 4.19: Código función time.....	46
Figura 4-20 Consulta Google del clima.....	47
Figura 4.21: Código función tiempo climático.....	48
Figura 4.22: Código función lectura de titulares.....	49
Figura 4-23 Activación del Api de Gmail .....	50
Figura 4-24 : Archivo de configuración JSON.....	50
Figura 4.25: Comando instalación cliente Gmail.....	51
Figura 4.26: Importar librerías Gmail.....	51
Figura 4.27: Código función obtener credenciales.....	51
Figura 4.28: Código función obtener correo electrónico.....	53
Figura 4.29: Código función obtener correo electrónico MIME.....	53

Figura 4.30: Código función obtener etiquetas del mensaje.....	54
Figura 4.31: Código consulta leer correos electrónicos.....	55

# Índice de tablas

Tabla 2-1 Grafica lenguajes programación usados.....	21
Tabla 4-1 Lista de entidades.....	47
Tabla 4-2 Comandos accesos Gmail.....	52
Tabla 5-1 Respuesta encuesta riesgos asistentes .....	58
Tabla 5-2 Respuesta encuesta beneficios asistentes .....	58
Tabla 5-3 Volumen valor mercado por millón asistentes inteligentes (US Dolares) .....	59
Tabla 6.1: Diagrama de Grantt.....	64
Tabla 6-2 Presupuesto del proyecto.....	65



# Capítulo 1

## Introducción

### 1.1- Motivación

En los últimos años se ha popularizado la idea expandir la robotización a los hogares con el objetivo de la interacción entre usuario y robots o sistemas para así facilitar la vida de los humanos. También ha crecido el uso de asistentes virtuales inteligentes, no solo para la telefonía móvil sino también para asistentes del hogar.

Aunque, debido a su alto coste, los actuales asistentes no son accesibles para todos, surge la idea de crear un asistente casero de coste reducido y sencillo, pero con funcionalidades parecidas a las que se pueden encontrar en el mercado. Para ello toma un peso importante la implementación de Raspberry Pi, este ordenador en miniatura puede soportar las funciones necesarias para conseguir este prototipo y proporcionar consultas que hagan más sencillas algunas tareas del hogar.

Por otro lado, otro factor es que la mayoría de los asistentes virtuales para el hogar disponibles para los usuarios en el mercado actual, tan solo soportan el lenguaje del inglés, es decir, las consultas proporcionadas por el usuario y las respuestas de los

dispositivos tan solo están disponibles en un idioma, por lo tanto, es interesante poner a disposición y crear este prototipo accesible para usuarios y países de habla hispana.

Por último, a todos los usuarios les gusta ahorrar tiempo en sus tareas del hogar o evitar realizar tareas rutinarias, por lo tanto, realizar este tipo de dispositivos pueden hacer la vida más llevadera y cómoda para los usuarios que tengan a su alcance este tipo de dispositivos.

## 1.2- Objetivos

El objetivo principal de este proyecto es la creación de un asistente personal inteligente para el hogar que tenga la capacidad de soportar el español como idioma principal y desplegado en un hardware reducido como es Raspberry Pi. Para ello la forma apropiada de trabajo es a través de hitos para ir completando su creación:

- **Hito 1:** Diseño del asistente mediante un PC portátil.
- **Hito 2:** Diseño e implementación del primer módulo, Snowboy, de activación mediante frase clave.
- **Hito 3:** Diseño e implementación del segundo módulo, conversión de voz a texto.
- **Hito 4:** Diseño e implementación del tercer módulo, procesado de la consulta.
- **Hito 5:** Diseño e implementación del cuarto módulo, conversión de texto a voz.
- **Hito 6:** Activación y acoplamiento del diseño a Raspberry Pi

Una vez completada la serie de hitos, se podrá disponer de un prototipo de asistente virtual para el hogar capaz de interactuar con los humanos y resolver una serie de consultas que el usuario realice a través de su propia voz.

## 1.3- Estructura del documento

**Capítulo 1: Introducción.** En este apartado del documento se dará un breve resumen del proyecto, donde se exponen las motivaciones y objetivos del proyecto en cuestión, la estructura del documento y las regulaciones a las que se somete.

**Capítulo 2: Estado del arte.** Capítulo donde se recogerá la información sobre las tecnologías usadas para el desarrollo del proyecto.

**Capítulo 3: Diseño.** Solución del sistema del cual trata este documento. Se describen las características de la solución para poder llegar a la solución y el plan de pruebas.

**Capítulo 4: Implementación.** Capítulo donde se expone la forma de implementación de las tecnologías usadas para poder desarrollar el proyecto de este documento.

**Capítulo 5: Marco socioeconómico.** Donde se explica el impacto económico que tiene este proyecto y el impacto sobre la sociedad actual y futura.

**Capítulo 6: Planificación y presupuesto.** Capítulo dedicado a la planificación en cuanto a tiempo y fechas de las distintas fases y tareas, así como el presupuesto del proyecto.

**Capítulo 7: Trabajos futuros y conclusiones.** Conclusiones a las que se ha llegado una vez finalizado el proyecto donde también se exponen posibles mejoras futuras.

## 1.4- Regulación

Los asistentes virtuales son mundialmente famosos por los usuarios y formas parte de la rutina diaria para mucho de ellos. Son conocidas sus funciones y el potencial

que pueden ofrecer a los humanos, para facilitar funciones rutinarias o realizar tareas por nosotros, sin embargo, no todos los usuarios tienen tan buena imagen de los asistentes para el hogar o no están del todo seguros de que su utilización sea del todo fiable, sobre todo, no se sienten seguros al saber que un dispositivo escucha continuamente y graba todo lo que se dice en su hogar y quien se encuentra detrás del dispositivo, es decir, les preocupa su privacidad.

Para garantizar esta privacidad, en primer lugar, el usuario debe aceptar una serie de términos y condiciones por las que el usuario da permisos y acepta las condiciones del dispositivo. El usuario puede permitir o no el uso de micrófonos, cámaras u otro tipo de entidades que pueden trabajar con datos del usuario.

Por otro lado, las empresas que desarrollan estos dispositivos se encargan de garantizar la privacidad del usuario, como a la hora de enviar información a un servidor para realizar una consulta en la que queda expuesta la ubicación del usuario, como calcular una ruta desde la ubicación actual al destino solicitado, la información sobre la ubicación usa identificadores anónimos y aleatorios para proteger al usuario y que su información o ubicación no queden expuestos y no se relacionen con el propio usuario.

También, los asistentes virtuales deben cumplir las leyes de protección de datos, estos dispositivos almacenan y cargan información sobre el usuario continuamente, información personal y privada, información necesaria para su correcto funcionamiento y para conocer al usuario lo mejor posible para así obtener una respuesta más eficaz y personal para el usuario, pero información por la que ningún otro sistema debe de poder acceder a ella.

Por lo tanto, todos los asistentes virtuales deben adaptarse a la ley de Reglamento General de Protección de Datos (RGPD), creada en 2016 y que entró en vigor el 25 de mayo de 2018. Esta ley nace para adaptarse a las nuevas tecnologías como Big Data o a los chatbots integrados en los actuales y futuros asistentes para el hogar para proteger los datos de identidad del usuario, sobre su privacidad, información administrativa o datos sensibles o estado del usuario. El incumplimiento de esta ley

puede suponer para las empresas que lo incumplan multas de hasta el 4% de su facturación anual [1] [2] [3].



**Figura 1-1** Reglamento General de Protección de Datos (RGPD)

# Capítulo 2

## Estado del arte

### 2.1- Asistentes personales inteligentes

Un asistente personal inteligente es un agente tipo software que puede realizar tareas y ofrecer servicios a un individuo. Estas tareas o servicios están basados en datos de entrada de usuario, reconocimiento de ubicaciones y la habilidad de acceder a información de una variedad de recursos en línea (como al clima o al tráfico, noticias, precios de acciones, horario del usuario, precios al por menor, etc). Algunos ejemplos de asistentes personales son Siri de Apple, Braina, Google Assistant, Google Now, Amazon Echo, Microsoft Cortana y S Voice de Samsung <sup>1</sup>. Uno de los aspectos clave de un asistente personal inteligente es su habilidad para organizar y mantener información. Esto incluye el manejo de correos electrónicos, eventos en el calendario, archivos, lista a seguir, etc.

---

<sup>1</sup> <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/siri-y-otros-asistentes-inteligentes-en-desarrollo-80552>

Ejemplos de tareas que pueden ser ejecutadas por inteligentes automatizados agentes personales, incluyen manejo de agenda como puede ser mandar una alerta a un contacto que llegará tarde a su cena por un problema de tráfico, o cambiar el horario de reserva de un restaurante.

También puede ser ejecutado como un manejador personal de salud para monitorizar la cantidad diaria de calorías, el ritmo cardíaco y el volumen de ejercicio para después hacer recomendaciones y dar opciones y opiniones para mejorar la salud.

La tecnología del asistente personal inteligente se da por la combinación de dispositivos e interfaces de programación de aplicaciones llamadas APIs.

### **2.1.1- Situación actual de los asistentes inteligentes**

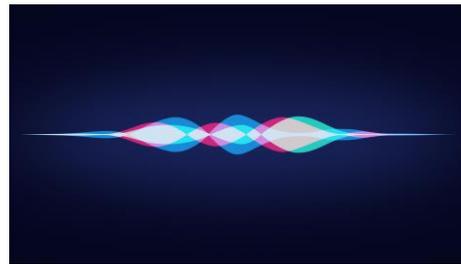
Ya que el dispositivo que se va a crear en este proyecto es un asistente inteligente para el hogar, se va a ilustrar como es la situación actual de los asistentes virtuales. En la actualidad ya existen numerosos asistentes como pueden ser Siri para Apple, Cortana para Windows o Google Assistant para Google, aunque estos están integrados en PC's, tablets o móviles. En cuanto a asistentes personales para el hogar podemos encontrar algunos como Amazon Echo de Amazon o también conocido como Alexa, Google Home, Jasper... aunque casi ninguno de ellos están preparados para responder y recibir órdenes en inglés<sup>2</sup>.

---

<sup>2</sup> Solo Google Home ha sido lanzado recientemente (mayo 2018) en español de forma comercial; Alexa se encuentra en periodo de pruebas.



**Figura 2-1** Logo de Cortana



**Figura 2-2** Logo de Siri

La funcionalidad básica de estos asistentes es la misma, todos ellos están compuestos por micrófonos integrados, hasta siete en el caso de Amazon Echo, inicialmente se encuentran en un estado de “reposo” y se activan con una palabra clave o “hotword” como puede ser “hey Siri” para Apple o “Alexa” para el caso del asistente de Amazon y es entonces cuando comienzan a escuchar la voz del usuario. Cuando el dispositivo se ha “despertado” está listo para recibir cualquier orden por parte del usuario para tratarla adecuadamente mediante el uso de la inteligencia artificial y tomar decisiones para cada una de las órdenes, como pueden ser tomar el control de la iluminación de la casa, reproducir música, incluso de otras plataformas como Spotify o Apple music, realizar compras en el propio Amazon, leer o responder tu correo...

### **2.1.2- Google Home Assistant**

La funcionalidad del Google Home Assistant, ya podía encontrarse en los smartphones, pero la empresa ha lanzado un altavoz inteligente cuya funcionalidad es muy similar al anterior, hasta la fecha, los asistentes para el hogar que podíamos encontrar en el mercado tan solo reconocían el inglés como idioma, sin embargo, Google se ha adelantado a sus competidores y lanzado su altavoz, que puede observarse en la figura 2.3, inteligente en español. Este altavoz de Google actualmente se encuentra a la venta en Estados Unidos, Canadá, Reino Unido, España, Francia y Alemania.



**Figura 2-3** Altavoz Google Home

Google Home Assistant no posee botonería, pero si una superficie táctil con leds para mostrar su estado de actividad. En este caso, la frase para activar el asistente es Ok Google, mediante el reconocimiento de voz puede realizar una gran variedad de tareas como:

-Responder preguntas generales del usuario: como obtener información de servicios como el tiempo, las últimas noticias, recetas de cocina, etc.

-Interactuar con el calendario del usuario: para realizar avisos de reuniones, fechas importantes como cumpleaños o citas del usuario.

-Reproducir contenidos en tu televisión: mediante Chromecast o un televisor compatible, es posible solicitar al altavoz que ponga una serie o película en un televisor de la casa. Ok Google podrá buscar el nombre de la serie en Netflix y comenzará su reproducción. También puede ejecutar ordenes consecuentes a esta como puede ser pausar la reproducción, avanzar o reproducir el siguiente capítulo.

-Reproducir música desde el propio altavoz: a través de plataformas como Youtube, spotify, entre otras, se puede dar la orden de reproducir una canción en propio altavoz o en otros dispositivos mediante bluetooth,etc.

-Controlar el hogar conectado: si se tienen dispositivos del hogar conectados al asistente Google Home, como lámparas, persianas o televisores, estos pueden ser controlados con facilidad a través del altavoz. También se pueden establecer grupos, de tal forma que puedas solicitar apagar todas las luces del salón o de la casa y hará lo propio con todas las bombillas del lugar solicitado [4].

### 2.1.3- Google Home Assistant Mini

Basándose en el anteriormente hablado, Google Home Assistant, la compañía Google ha lanzado al mercado un nuevo asistente para el hogar denominado, Google Home Assistant Mini (figura 2.4), es decir, un asistente reducido en tamaño, aunque no reducido en prestaciones. Se puede encontrar para los mismos idiomas que la versión Google Home Assistant y lleva incorporada una nueva función para el reconocimiento de voz llamado Voice Match, con el cual, el asistente puede reconocer distintas voces con el objetivo de diferencias a los distintos usuarios.



**Figura 2-4** Altavoz Google HomeMini

Al igual que su versión ampliada, posee 4 leds en su parte superior con la que podemos observar si el asistente esta encendido o procesando, entre otras. Este nuevo modelo, posee un tejido táctil con el que también se puede interaccionar con él de forma manual, como puede ser tomar el control en la reproducción de música. Pero la gran virtud de este asistente en su versión reducida es el anteriormente mencionado reconocimiento de voz, Voice Match, mediante la cual el asistente puede distinguir que usuario esta interactuando con él, y puede tomar una lista de contactos u otra, dependiendo del usuario que quiera realizar una llamada [5].

También podemos encontrar otros dispositivos de software libre como Jasper, creado a partir de una Raspberry Pi. Este asistente se encuentra disponible en su página oficial (<http://jasperproject.github.io/documentation/>), basta con descargar el software y conectar la Raspberry Pi. Mediante Jasper se puede acceder al calendario de Google donde el usuario puede consultar sus eventos, acceder a Twitter, ver películas acceder a Spotify, etc.

## **2.2- La inteligencia artificial**

### **2.2.1- ¿Qué es la inteligencia artificial?**

La inteligencia artificial (AI) está considerada como aquella rama de la tecnología que diseña programas o sistemas, mediante algoritmos, que son capaces de tomar decisiones por ellos mismos, es decir, que imitan los procesos humanos mediante aprendizaje [6].

### **2.2.2- Historia de la inteligencia artificial**

En cuanto al origen de este avance, surgió sobre la década de los 40, alrededor de algunos trabajos de Walter Pitts y Warren Mccolloch, donde se empezaron a plantear teoría e ideas sobre la inteligencia artificial, mediante el estudio del cerebro humano y las formas de sus neuronas, lo entendieron como un organismo computacional, por lo que sería posible la idea de construir computadoras que imitaran nuestro sistema. Pero

no fue hasta los años 50 cuando y gracias a Allan Turing, cuando esta idea empezó a tomar formas gracias a un experimento llamado Computing Machinery and Intelligence, donde intento demostrar si una computadora era capaz de pensar. Este experimento constaba de dos personas y un ordenador, una de las personas debía de introducir preguntas en la computadora e intentar adivinar si las respuestas eran otorgadas por otra persona o por la propia computadora. Ya en la década de los 80, Martin Fischles y Oscar Firschein, crearon los atributos que debe tener un agente inteligente, como tener actitudes mentales como tener creencias e intenciones, tener la capacidad de aprender, resolver problemas complejos descomponiéndolos en otros problemas más simples o entender, es decir, tener la capacidad de dar sentido a ideas contradictorias, entre otros atributos. En definitiva, un agente inteligente debe ser capaz de aprender, entender y razonar.

Estas ideas fueron avanzando y evolucionando hasta la tecnología que se tiene hoy en día, pasando previamente por programas como A.L.I.C.E (Artificial Linguistic Internet Computer Entity), que ganó un premio al Chatbot más humano [7].

### **2.2.3- Tipos de inteligencia artificial**

Según algunos expertos, dentro de la inteligencia artificial se pueden diferenciar cuatro categorías [8]:

- Maquinas reactivas: este tipo de máquinas son del estilo acción-reacción, es decir, toman una decisión a partir de un escenario, pero no tienen memoria ni pueden basarse en experiencias anteriores para la toma de una decisión. Como ejemplos se pueden encontrarlas maquinas AlphaGo, diseñado para el juego de mesa Go, o Deep Blue, diseñado por IBM para jugar al ajedrez, capaz de detectar las figuras dentro del tablero para hacer predicciones y realizar el movimiento más estratégico.

Este tipo de sistemas pueden clasificarse como capaces de pensar como humanos.

- Memoria limitada: este tipo de sistemas, a diferencia del anterior, si tienen la capacidad de basarse en experiencias pasadas para la toma de decisiones, aunque estas decisiones no se almacenan en memoria por un largo tiempo.

Se clasifican como máquinas que actúan como los seres humanos, como el caso de robots.

- Teoría de la mente: son sistemas que intentan actuar siguiendo el modelo humano lógico racional. Este tipo de máquinas aun no existen, pero se investiga en cómo lograr que las máquinas sean capaces de percibir, razonar y actuar en consecuencia, tal como lo haría una persona.
- Autoconocimiento: este tipo de sistemas son los que tienen conciencia y pueden interferir en los sentimientos, por lo que tratan de emular los procesos racionales humanos.

#### **2.2.4- Aplicaciones de la inteligencia artificial**

Algunos de los ejemplos que se pueden encontrar en la actualidad sobre inteligencia artificial la automatización, el aprendizaje automático, la visión de la máquina, procesamiento del lenguaje, el reconocimiento de patrones y la robótica.

- La automatización son los procesos mediante los cuales un sistema realiza procesos sin interacción humana, como programar tareas que sean repetitivas y así ahorrar tiempo a los humanos.
- En cuanto al aprendizaje automático, es el proceso mediante el cual una máquina se capaz de actuar sin programación, es decir, por sí sola. Dentro de este aprendizaje se pueden diferenciar tres subconjuntos que son, el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje de refuerzo.
- La visión de la máquina es la facultad de crear sistemas que vean, Esta visión analiza la información captada a través de la cámara, para ellos se requiere de una transformación de analógico al digital y el procesamiento de esa

información digital. Este tipo de funciones puede usarse para la comprobación de firmas, por ejemplo.

- El procesamiento del lenguaje natural es el proceso por el cual una máquina o sistema es capaz de reconocer el lenguaje humano. Entre estos procesos se pueden encontrar el análisis de las expresiones para reconocer estados de ánimos, la traducción o comprensión de un texto o el reconocimiento por voz.
- Por último, en la robótica, los robots pueden emplearse para realizar tareas que los humanos no son capaces o tareas periódicas y recurrentes, como líneas de montaje de vehículos.

La inteligencia artificial posee un gran potencial con el que ofrecer numerosos usos con el objetivo de reemplazar procesos humanos, realizar tareas o emular comportamientos humanos, entre otras. Uno de los usos más conocidos es en el sector tecnológico como asistentes personales virtuales o aplicaciones móviles, pero su uso no se limita solo al área tecnológica, sino que abarca otros múltiples campos.

En sanidad, se trabaja para reducir costes y esperas por parte de los pacientes. Como ejemplos de los sistemas de inteligencia artificial dentro de este campo se pueden encontrar IBM Watson y chatbots. Los chatbots son programas en línea que tienen la función de crear una conversación con el paciente a través de preguntas y respuestas y son capaces de realizar un diagnóstico. Por otro lado, en cuanto al IBM Watson, es capaz de comprender el lenguaje humano para responder a preguntas formuladas por el usuario y es capaz de obtener un diagnóstico para el paciente que le está consultando a partir de sus respuestas y obteniendo datos de él.

En el sector de las finanzas, se pueden encontrar aplicaciones como Turbo Tax o Mint que tienen la capacidad de obtener datos personales y proporcionar apoyo financiero. También se puede emplear la inteligencia artificial para ayudar a entidades como los bancos a detectar fraudes o detectar patrones dentro del mercado.

En el área de la educación la inteligencia artificial puede realizar la tarea de calificación, para así ahorrar tiempo en la entrega de calificaciones y proporcionar a profesores más tiempo en otras tareas como tutorías, adaptarse a las necesidades de los

alumnos y proporcionar apoyo extra a los estudiantes. Por otro lado, también podría ser posible crear sistemas de inteligencia artificial que realicen este tipo de tareas de apoyo adicional cambiando así el modelo actual ya que se podría adaptar mejor a los estudiantes en cuanto al cómo y el dónde de su formación [9].

En la siguiente figura 2.5 se puede observar un resumen de las aplicaciones básicas que puede ofrecer la inteligencia artificial.



Figura 2-5 Principales aplicaciones AI

## 2.3- Raspberry Pi

Para la implementación de este proyecto se empleará una Raspberry Pi. La Raspberry Pi se puede definir como una placa computadora de bajo coste o un ordenador de magnitud reducida como la de la siguiente figura 2.6, de tal forma que elimina los componentes de un ordenador básico que no afectan a su rendimiento básico.

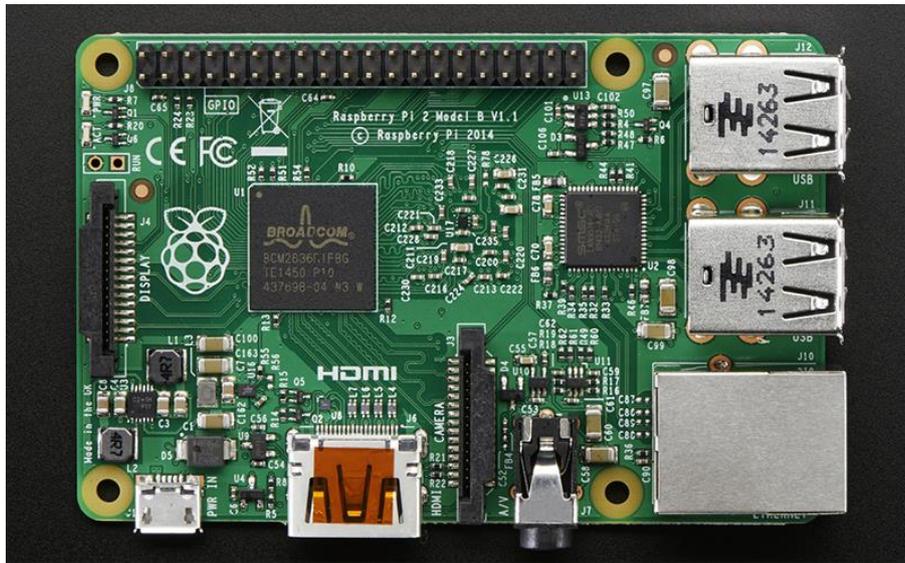


Figura 2-6 Placa Raspberry Pi

En cuanto al diseño para esta solución, Raspberry Pi incluye: una CPU, un procesador gráfico (GPU), memoria RAM con 512 MB de capacidad, un conector Ethernet para proporcionar acceso a Internet (aunque en Raspberry Pi ya es posible conectarse a través de WIFI), 2 puertos USB, una salida de audio Jack de 3.5 mm, salida digital de vídeo y audio por HDMI, 26 pines GPIO, un conector de alimentación microUSB y un lector de tarjetas SD, ya que Raspberry no posee un disco duro de almacenamiento, es donde se almacenará el sistema operativo seleccionado y nuestros archivos.

En cuanto a los sistemas operativos que soportan este tipo de dispositivos, encontramos los básicos de los PC's como son Linux, Windows u Os X. Aunque está diseñada para usar Linux, de hecho, el sistema operativo creado exclusivamente para Raspberry Pi es el denominado Raspbian que trabaja sobre Linux. Esto supone una ventaja ya que Linux en un sistema operativo de código abierto, es decir, se puede descargar y hacer cambios de código, a diferencia de Windows o Os X [10].

### **2.3.1- Historia de la Raspberry Pi**

Respecto a su historia, sus primeros esquemas y diseños son del año 2006. En 2009 se creó la fundación Raspberry Pi en Reino Unido, que tenía el objetivo de reunir a profesores informáticos, académicos o en definitiva todo aquel con conocimiento en informática para crear un ordenador reducido, sobre todo, el objetivo de introducirlo en los colegios y aulas para enseñar y para que los niños tuvieran una motivación en aprender sobre este campo.

Pero los primeros modelos no llegaron hasta el año 2011, donde se fabricó un modelo alfa donde ya era posible mostrar el escritorio de la aplicación en calidad vídeo Full HD, pero este primer modelo alfa era mucho más grande que las actuales, ya que aún no había sido posible depurarlo. Ya en 2012, la fundación anunció la puesta en venta del modelo B de Raspberry [11].

### **2.3.2- Modelos de Raspberry Pi**

A fecha actual, existen tres modelos con distintas variantes, Raspberry Pi 1 con sus modelos A (sin puerto Ethernet), B y B+, estos dos últimos doblando en capacidad de memoria RAM al modelo A, también se acopló un puerto USB extra y el puerto Ethernet fundamental para versiones posteriores. Raspberry Pi 2 modelo B, donde el procesador que era de un núcleo se sustituye por uno de cuatro y la memoria RAM vuelve a doblarse en capacidad obteniendo 1 GB. Y por último Raspberry Pi B, en la cual se incluye WIFI y B+, contando con un nuevo procesador y obteniendo una mejor conectividad. Aparte, la fundación también creó otro modelo mucho más pequeño y con mucha menos potencia, la denominada Raspberry Pi Zero y Raspberry Pi Zero W [12].

### **2.3.3- Algunos usos de aplicaciones en Raspberry Pi**

Como ejemplo de aplicaciones diseñadas por otros usuarios para Raspberry Pi, podemos encontrar aplicaciones sencillas con el tema de la electrónica digital como

controlar leds por medio de la voz o realizar juegos de luces, semáforos... Gracias a los GPIO que sistema nos proporciona podemos conectar una gran variedad de dispositivos como leds, displays...

- El uso más empleado para Raspberry es como un PC en miniatura, al tratarse de un miniordenador no podrá dar las características o prestaciones de un PC portátil o uno de sobre mesa, más potentes, pero la Raspberry Pi trae incorporado todo lo necesario para que con ella se puedan realizar tareas como leer el propio correo, navegar, editar documentos, etc.

-Raspberry Pi como Media Center. Uno de los usos más populares entre los usuarios de este tipo de dispositivos es el de emplear Raspberry Pi como un servidor o dispositivo multimedia con el que reproducir el contenido de los usuarios, tales como reproducir series, películas o sus canciones favoritas. Raspberry Pi proporciona a los usuarios una gran variedad de recursos para conseguir este Media Center y además de ello, potenciarlo mediante algún elemento hardware. Aparte, se pueden instalar algunos complementos como es el caso de Rock64, con el que el usuario puede reproducir contenido en 4K HDR.

-Convertir una televisión en un SmartTv. Una aplicación con mucho potencial es la de poder convertir una televisión antigua o de una versión anterior en un SmartTv. Gracias a Raspberry Pi, se puede incorporar las herramientas necesarias para dar este paso, es decir, aportar una conexión a Internet a esa televisión antigua, que es la principal virtud de las SmartTv's. Empleando para ello plataformas como la anteriormente mencionada con la que es posible obtener un centro multimedia o herramientas como Plex, que puede permitir añadir funciones de servidor de contenidos para poder disfrutar de los archivos multimedia que el usuario tiene almacenado en otros equipos, incluso enviar contenido desde el Smartphone a la Raspberry, donde se podrá reproducir en la televisión.

Por otro lado, una vez se ha decidido el modelo de la Raspberry a emplear y que aplicación realizar, se debe seleccionar el lenguaje de programación sobre el cual realizar la aplicación. Raspberry nos proporciona ya mediante su instalación el lenguaje de Python en sus versiones 2 y 3, y podemos acceder a él a través de la herramienta IDLE, un entorno de desarrollo para Python. Este lenguaje es potente en comparación

con otros lenguajes y muy visual, es decir, es limpio y fácil de leer y escribir. Mediante Python podemos controlar los GPIO de la placa y poder realizar las aplicaciones anteriormente mencionadas, con el uso de librerías, mediante las cuales podemos utilizar funciones ya implementadas para utilizarlas en las aplicaciones o programas que crean otros usuarios. Es posible importar librerías como RPi.GPIO para controlar los GPIO de la placa u otras librerías para el reconocimiento por voz.

Debido a la gran compatibilidad del sistema operativo Linux en Raspberry, se ha decidido trabajar con el sistema Raspbian para la realización de este proyecto, sobre un código basado en Python y empleando para ello una Raspberry Pi modelo B [13] [14].

## 2.4- Python

Python es un lenguaje de programación que se caracteriza por su sencillez y legibilidad ya que fue diseñado para ser leído con facilidad, además, Python, en comparación a otros lenguajes como pueden ser Java, C o C++, requiere menos líneas de código para elaborar tareas básicas o algoritmos complejos, pudiendo hablar de un código de cinco veces más corto para el caso de Java y entre cinco y diez veces para el caso de C++, dándole así la virtud de ser un lenguaje práctico y que ahorra al usuario líneas de código y tiempo.



**Figura 2-7** Logo Python

### 2.4.1- Características del lenguaje

Hablando del propio lenguaje, algunas de sus características son:

- Palabras por símbolos: donde otros lenguajes utilizan símbolos para definir variables o acciones, Python utiliza palabras, como puede ser el caso de los operadores lógicos, ya que en vez de emplear símbolos como `&&` o `||`, utiliza las palabras “and” o “or” respectivamente.
- Delimitación de bucles o funciones: El contenido de los bucles, condiciones o funciones para este lenguaje no se encuentra delimitado por llaves ( { } ) como para otros lenguajes, sino que utiliza para ello espacios o tabuladores, es decir, hace uso de la indentación, donde todo lo que pertenece a una misma altura, forma parte del mismo bloque.
- Comentarios: Normalmente cada lenguaje define su propia forma a la hora de definir comentarios en el código del programa, así como en el caso de Java o C se emplea el uso de barras laterales “/” para comentar una línea o los símbolos “/\* ” y “ \*/ ” para comentar un bloque de líneas, en el caso de Python se usan tres apostrofes ( ‘ ‘ ‘ ) anterior y posteriores a las líneas que se quieren comentar, aunque también se puede utilizar el carácter “ # ” para comentar una sola línea.
- Funciones: una de sus características principales, ya que nos permite agrupar expresiones o sentencias para realizar alguna ejecución al ser llamadas por el programa [15].

### 2.4.2- Ventajas frente a otros lenguajes

Las virtudes que principalmente destacan frente a otros lenguajes de programación son su facilidad de aprendizaje por parte de los usuarios y la amplia y actualizada comunidad de Python. En cuanto a la facilidad de aprendizaje depende de la capacidad y habilidad del usuario en cuestión que se dispone a aprender el lenguaje,

pero según la opinión de muchos usuarios, en comparación con otros lenguajes, Python es más sencillo que cualquier otro lenguaje. En cuanto a su comunidad, se pueden encontrar foros con dudas de los usuarios, donde otros usuarios de la comunidad resuelven y pueden ser consultadas por toda la comunidad, con el fin de compartir conocimiento sobre el lenguaje y ayudar a otros usuarios. También en estos foros podemos encontrar código o proyectos de otros usuarios relacionado con dudas o problemas con el fin de ser solucionados o que pueden servir como punto de partida para proyectos en mente de los usuarios de Python.

Por otro lado, según el índice de TIOBE (The Important Of Being Earnest), Python es el cuarto lenguaje de programación más utilizado y popular, además de encontrarse en el tercer puesto en la tabla de lenguajes de programación con más ingresos, como se puede ver en la tabla 1.1 [16].

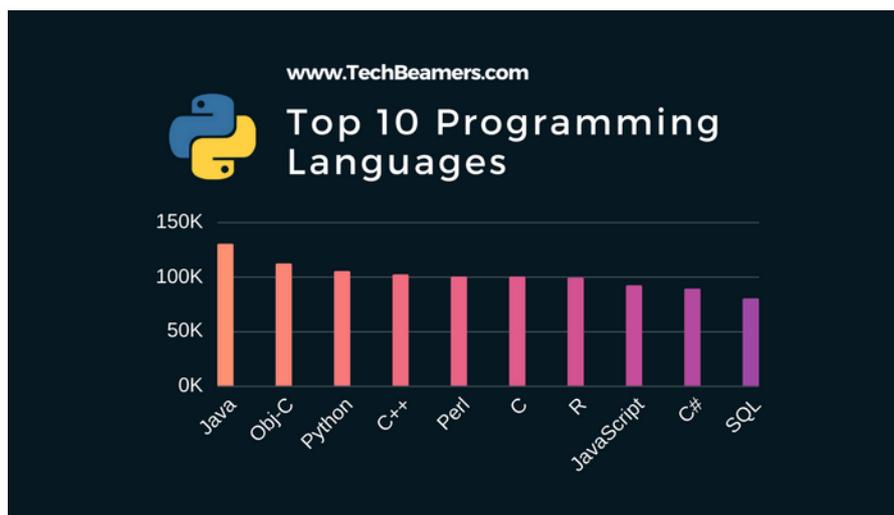


Tabla 2-1 Grafica de lenguajes programación más usados

### 2.4.3- Razones para el uso de Python

Python posee numerosas cualidades que le hacen un lenguaje atractivo para el usuario a la hora de elegir un lenguaje de programación para desarrollar cualquier tipo de proyecto. Una de estas virtudes es su capacidad multiplataforma, ya que puede ser empleado en multitud de ellas, como en Raspberry para realizar el dispositivo del cual

trata este documento, gracias a su lenguaje de programación interpretado y puede ser funcional en cualquier tipo de plataforma siempre y cuando lleve integrado su interpretador. Aparte del integrador, se pueden encontrar dialectos tales como Jython, que se puede usar para programar en Java.

Por otro lado, se encuentra la sintaxis propia de programación de este lenguaje, ya que una de sus características principales es el uso de bloques formados mediante indentación para la implementación de bucles o condiciones, obligando así al programador a indentar el código que está realizando, consiguiendo así un código legible y con facilidad de leer y comprender [17].

# Capítulo 3

## Diseño

El siguiente proyecto de asistente para el hogar en español está compuesto por un sistema que trabaja sobre una Raspberry Pi 2, mediante el cual, sea capaz de realizar tareas que un usuario le exija simplemente con el uso de su propia voz con el fin de que el dispositivo responda a esta petición y tome las decisiones pertinentes.

La realización sobre Raspberry Pi proporciona una gran potencia al dispositivo, ya que se trabaja sobre un pequeño ordenador. La gran comunidad existente y su bajo coste, la hacen un modelo atractivo para trabajar sobre ella, ya que se emplean para casi todo, se pueden encontrar numerosos proyectos.

Enfocado en este dispositivo, Raspberry Pi proporciona al usuario la capacidad de conectar numerosos elementos hardware como micrófonos, altavoces, leds, etc. con lo que implementar este tipo de dispositivos. Por otro lado, Raspberry Pi, puede trabajar sobre Python, lo que genera numerosos beneficios.



Figura 3-1 Diseño asistente inteligente

### 3.1- Función del dispositivo

El asistente captará la voz con la orden o consulta por parte del usuario y realizará las acciones apropiadas para después dar la respuesta de la acción de forma hablada a través del altavoz, de tal forma que se produzca una conversación entre el asistente y el usuario. Para lograr esta funcionalidad la aplicación cuenta con varios módulos, cuyo diagrama se presenta en la figura 3.2, diferenciando entre la detección de la palabra clave, el paso de voz a texto, el proceso mediante el cual se toma la decisión sobre la petición y por último pasar a voz la respuesta necesaria.

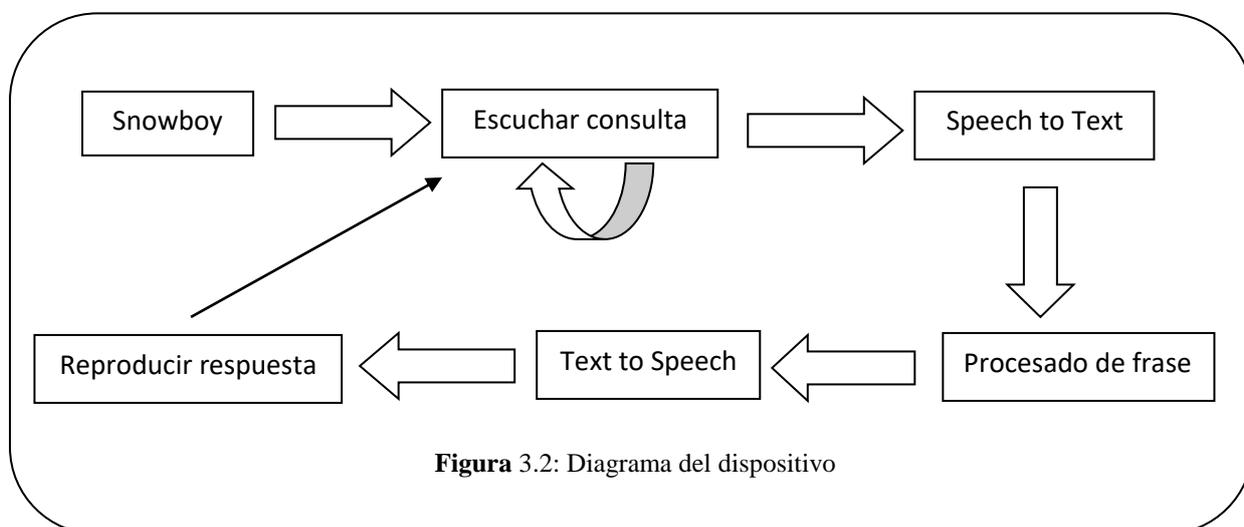


Figura 3.2: Diagrama del dispositivo

- Snowboy: mediante el cual el dispositivo cambiará entre sus estados “dormido” y “despierto” gracias a una palabra clave creada por el propio usuario.
- Speech-to-text: en él se realizará el cambio de la orden o consulta generada de voz a texto para poder tratar la cadena y tomar las decisiones pertinentes.
- Wit.ai: recibiendo una cadena de texto, este módulo identificará las palabras relevantes, devolviéndolas como parámetro y realizar las acciones oportunas que el usuario había requerido.
- Text-to-speech: en este último modulo, se llevará a cabo el cambio de texto a voz con el resultado final de la consulta, con el objetivo realizar una conversación.

### 3.1.1- Funcionalidad

Para la implementación de este dispositivo se han decidido realizar cuatro pruebas, con el objetivo de realizar una demo de los que puede llegar a ser un dispositivo real, como los mencionados anteriormente y que están o pueden llegar a estar en el mercado y así conseguir una versión reducida de estos dispositivos. Estas cuatro funciones pueden ser ampliadas y ser añadidas otras muchas más como las mencionadas en el capítulo 6.

**Caso de prueba 1:** Para comenzar, una de las primeras implementaciones que se pueden encontrar para este prototipo es la capacidad de proporcionar la hora actual, para ello el dispositivo realiza una consulta al propio dispositivo que accede a la hora local para así proporcionársela al usuario.

El prototipo responderá al usuario con la siguiente frase (figura 3.3):

Son las +” hora del sistema”.

**Figura 3.3:** Respuesta dispositivo a consulta time

**Caso de prueba 2:** el usuario puede consultar el tiempo climático y temperatura de cualquier ciudad del mundo. Para ello el usuario pregunta por el tiempo en cualquier ciudad y el dispositivo accede a los servicios de Google para así obtener el clima de esa ciudad requerida.

La respuesta del dispositivo será de la siguiente forma (figura 3.4):

El tiempo en +”ciudad” + es + “clima proporcionado por Google”+con una temperatura de “temperatura proporcionada por Google”.

**Figura 3.4:** Respuesta dispositivo a consulta clima

**Caso de prueba 3:** el dispositivo es capaz de acceder al correo personal del usuario, previamente este debe de haber proporcionado sus credenciales y autorización para su acceso, de tal forma que el usuario podrá comprobar si tiene correos nuevos, es decir, mediante una simple consulta por voz, el dispositivo proporcionará una lista de los nuevos correos en la bandeja de entrada del email. Si no existen correos sin leer, el dispositivo responderá con la siguiente frase (figura 3.5):

No existen nuevos correos.

**Figura 3.5:** Respuesta dispositivo a error en consulta lectura de correos

Si existen correos sin leer (figura 3.6):

Tienes +”número de nuevos correos”+ correos nuevos de + “nombre de los usuarios que enviaron dichos correos”.

**Figura 3.6:** Respuesta dispositivo a consulta lectura de correos

**Caso de prueba 4:** La última funcionalidad para este dispositivo es la posibilidad de leer los titulares de las noticias más destacadas. Entre estos periódicos se pueden encontrar: el mundo, diario As, Marca y El País. Para esta funcionalidad, el dispositivo hará consultas a las páginas RSS correspondientes del periódico en cuestión. Mediante esta consulta, recorrerá la sección de los titulares y los almacenará para

después devolver la consulta al usuario que realizó la petición. Para no exceder en tiempo, esta función ha sido implementada de tal forma que tan solo se reproduzcan los primeros cinco titulares para cada consulta, ya que serán los de más relevancia y los titulares más recientes.

Para este último caso, el dispositivo devolverá una frase del siguiente estilo (figura 3.7):

Los titulares de +” periódico”+ son + “listado con los cinco primeros titulares encontrados”.

**Figura 3.7:** Respuesta dispositivo a consulta lectura de titulares

## 3.2-Modulos

### 3.2.1- Snowboy

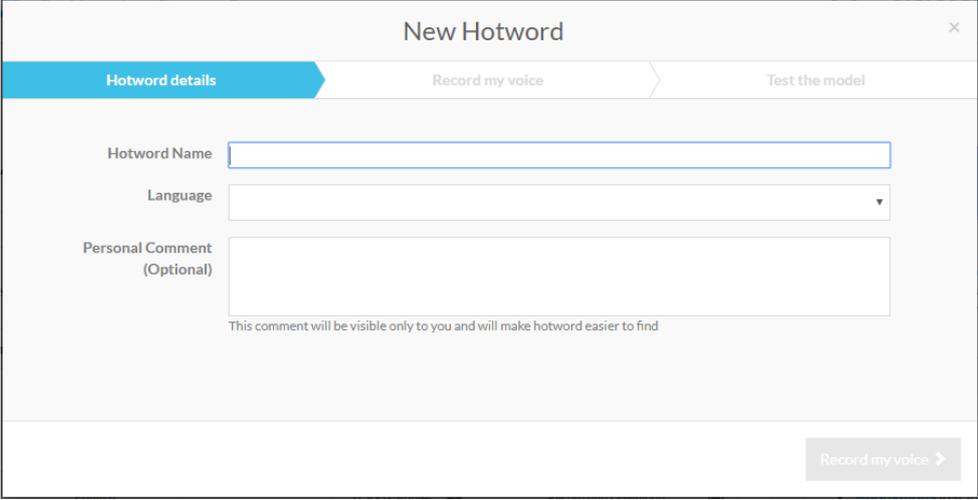
El primero de los módulos que se encuentran en el dispositivo es la detección de la palabra o frase clave para su activación y así conseguir la interacción entre el usuario y el dispositivo.



**Figura 3-8** Logo de Snowboy

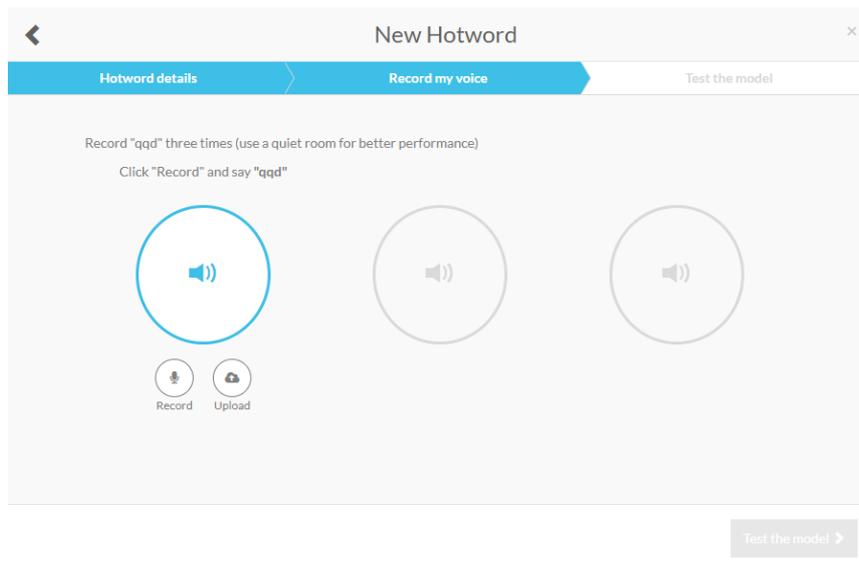
Para este proyecto el método implementado que se utilizará es a través de la API de Snowboy. Snowboy es un motor de detección de palabras claves que se caracteriza por ser personalizable, ya que un usuario puede crear y entrenar un modelo propio, con el cual tener una alta precisión y ser compatible en múltiples plataformas [18].

Para entrenar un modelo mediante Snowboy son necesarios tres pasos. En primer lugar, es preciso pensar y seleccionar una palabra clave con la cual activar el dispositivo y activar el micrófono con el que se va a escuchar y grabar al usuario, como pueden ser “hey Siri” o “Alexa” para asistentes más populares por los usuarios, algo único que caracterice y diferencie a la aplicación frente a las demás.



**Figura 3-9** Creación de la palabra clave

Una vez elegida la palabra clave que acompañará al dispositivo, es necesario grabarla tres veces para entrenar correctamente el modelo y así evitar errores en la detección o confusiones con otras palabras parecidas y, en definitiva, conseguir una mayor fiabilidad. Para ello, la página oficial de Snowboy proporciona la herramienta necesaria para lograr un modelo personalizado. Para crear este modelo, se elegirá un nombre y el lenguaje para el cual será utilizado, para más adelante empezar a grabar la voz en tres ocasiones.



**Figura 3-10** Grabación de la palabra clave

Por último, se envían los archivos mediante RESTful API, la cual formará y generará el modelo adecuadamente, con el nombre que el usuario había seleccionado anteriormente y con el formato propio para este tipo de ficheros “.pmdl”, para que el usuario ya lo pueda emplear en su aplicación. Además, para poder utilizar la herramienta Snowboy en la plataforma seleccionada es necesario instalar las librerías de Portaudio y Pyaudio para Python, que se explicarán en el punto 4 del documento. Estas dos librerías son necesarias para la entrada y la salida de audio.

Una vez ya se tiene nuestro modelo entrenado y descargado, es necesario incluirlo en el código de la aplicación para su funcionamiento. Para ello, mediante el comando de la figura 3.11 se informa a la aplicación, a través del terminal, de los ficheros que necesita para su ejecución, donde se le incluirá el modelo que ha sido preparado para el dispositivo y el ejecutable que lanzará del programa que lanzará la aplicación.

```
python app_v1.py Modelo.pmdl
```

**Figura 3.11:** Comando ejecución del dispositivo

Si la palabra detectada por voz coincide con la palabra clave, anteriormente entrenada, el dispositivo emitirá un pequeño sonido de verificación para así hacer ver al usuario que su voz ha sido captada correctamente y que el dispositivo ya se encuentra disponible para realizar cualquier consulta, es decir, se encuentra en modo despierto.

Por otro lado, si la palabra detectada es incorrecta o ha existido algún tipo de fallo, como ruido o interferencias, la aplicación emitirá un sonido de error y volverá al estado anterior, de tal forma que el usuario pueda volver a intentar activarlo mediante la palabra clave.

### **3.2.2- Escuchar consultas del usuario y Speech-to-text.**

Una vez el dispositivo se encuentre en modo “despierto” está listo para interactuar con el usuario. El asistente ya está grabando cualquier frase por parte del usuario con la consulta proporcionada, y una vez captada se necesita una conversión de audio a texto para que la orden pueda ser tratada. En este caso, la herramienta seleccionada será Speech Recognition y más concretamente el servicio que ofrece Google Speech, debido a su fiabilidad y su capacidad para reconocer numerosos idiomas, entre ellos el español, uno de los factores más determinantes para este proyecto. Mediante esta herramienta, nuestro archivo de voz será enviado a los servidores de Google, el cual nos devolverá una cadena con la frase que el usuario había proporcionado con anterioridad a través del micrófono, para poder empezar a trabajar con ella. Otras opciones pueden ser la API Google Cloud o pocketsphinx, entre otros [19].

### **3.2.3- Wit.ai**

Cuando ya se tiene la voz convertida a texto, se puede empezar a tratar esta cadena de caracteres con el objetivo de realizar unas acciones u otras dependiendo del resultado que se quiera esperar. Para este módulo, se va a implementar el Api de Wit.ai,

ya que a través de esta herramienta se puede procesar el texto para convertirlo en acciones [20].



**Figura 3-12** Logo Wit.ai

Wit.ai es una herramienta de fácil utilización que permitirá cumplir con el objetivo.

Wit.ai permite diferenciar las diferentes entidades dentro de una frase, estas entidades tomarán un tipo, como puede ser int o float para otros lenguajes de programación, y un valor, el tipo será el que el usuario le adjudique en el entrenamiento, como puede ser una ciudad, un número o una fecha, entre otras, y el valor será el propio campo de esa palabra detectada. Con un correcto entrenamiento de la herramienta, por cada frase introducida por el usuario se conseguirá un conjunto de entidades con sus correspondientes valores.

Posteriormente, se tomarán acciones dependiendo de recibir unas entidades u otras. Estas acciones pasan tanto desde consultar a servidores como hacer preguntas al propio sistema, como en el caso de que el usuario necesite saber la hora actual, hasta consultar páginas html con el objetivo de obtener información de ellas como pueden ser temperaturas o conseguir enlaces de noticias que pueden ser reproducidos más adelante.

Para este asistente personal se implementarán cuatro acciones a forma de demo de lo que puede llegar a ser una aplicación totalmente real y ser competente en el mercado, las cuales se desarrollarán en el apartado 5 de este documento. Estas acciones serán:

- Consulta de la hora actual
- Consulta del tiempo climático en cualquier población del mundo

- Leer las últimas noticias de varios periódicos
- Consulta de nuevos correos electrónicos.
- También será posible cerrar la aplicación por voz

### **3.2.4- Text-to-speech**

Como último módulo del asistente se encuentra la conversión de texto a audio, el análogo del paso de audio a texto. En esta ocasión, será necesario que se consulte a la API de “text to speech” de Python, que transforma una cadena de caracteres a voz. El estilo de la voz, la velocidad incluso el idioma pueden ser modificados con el fin de construir una aplicación y un asistente personalizado diferente a otros ya existente.

Una vez finalizado este último modulo, el asistente quedara a la espera de recibir una nueva instrucción y ponerse a trabajar en ella de nuevo [21].

# Capítulo 4

## Implementación

### 4.1- Introducción y visión general

El siguiente punto se centrará en la implementación del asistente virtual inteligente.

Para ello, en primer lugar, será necesaria la instalación de la Raspberry Pi 2 para la implementación del dispositivo, para después instalar las librerías correspondientes para la elaboración del asistente, diferenciando los cuatro módulos.

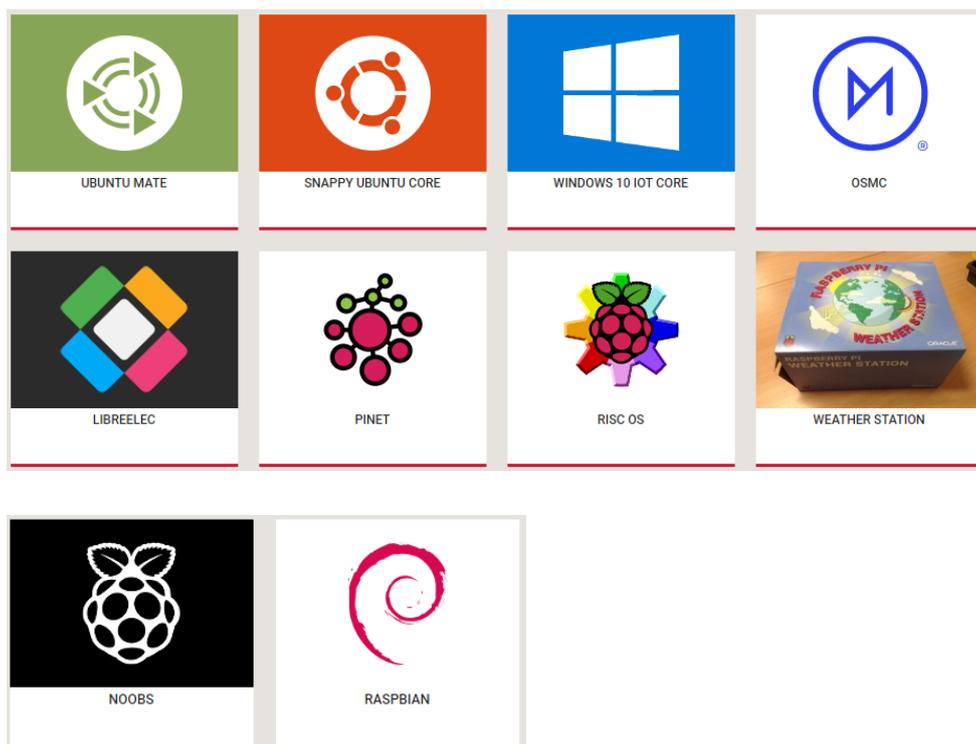
## 4.2- Preparación del Entorno

La implementación de esta aplicación podría haberse realizado sobre múltiples plataformas como pueden ser una aplicación móvil, una aplicación de escritorio... La decisión de realizarlo sobre una Raspberry Pi, se debe al bajo coste, a la facilidad de interacción, ya que puede ser colocado en cualquier lugar del hogar y con un micrófono adecuado, puede trabajar sin ningún problema y no es necesario desbloquear una pantalla de móvil ni encender ningún ordenador, además, ya que Raspberry Pi trabaja sobre lenguaje Python, se encuentra la facilidad de implementación de librerías.

## 4.3- Instalación de Raspberry Pi

### 4.3.1- Instalación del sistema operativo

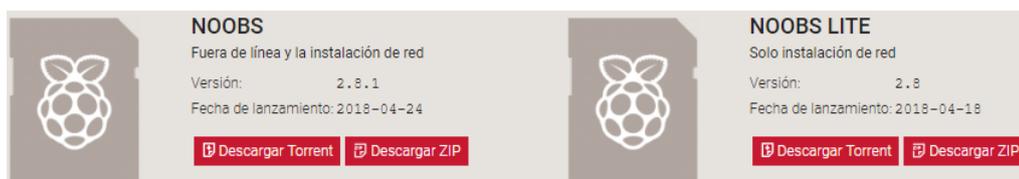
Antes de realizar cualquier tipo de instalación, es necesario elegir el sistema operativo con el que se va a realizar la implementación. La página oficial de Raspberry <https://www.raspberrypi.org/downloads/> nos ofrece una gran variedad de distribuciones basados en sistemas operativos compatibles con Raspberry, sobre todo sobre Linux, como pueden ser Ubuntu Mate o Snappy Ubuntu Core entre otros, aunque también podemos encontrar otros como Windows 10 IOT y el sistema oficial para Raspberry, Raspbian.



**Figura 4-1** Logos de los sistemas operativos Raspberry Pi

El sistema más indicado y que se utilizará para esta instalación es Noobs, un instalador sencillo de Raspbian. También existe la opción de realizar la instalación mediante Raspbian Lite, aunque esta versión solo puede instalarse mediante red y no contiene el sistema Raspbian precargado.

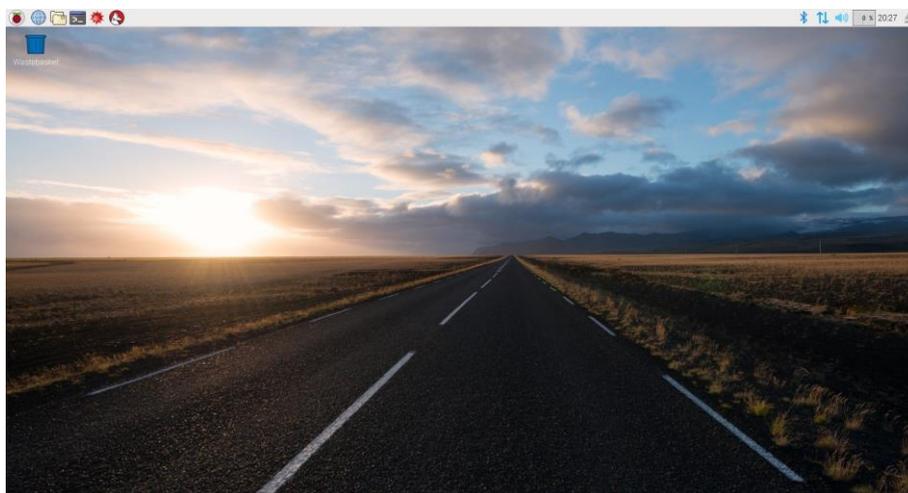
Para su instalación, basta con descargar el archivo que nos ofrece la página oficial, descomprimirlo y trasladar los archivos descomprimidos en una tarjeta SD con al menos 8GB de memoria.



**Figura 4-2** Instaladores sistemas operativos Raspberry Pi

Una vez descargado el archivo en la tarjeta SD, de Raspberry Pi. Para comenzar con la instalación, se necesita conectar la Raspberry Pi a un monitor mediante un cable HDMI, se deberá insertar en el puerto correspondiente la tarjeta SD anteriormente mencionada y conectar un cable Ethernet por el puerto Ethernet (RJ45) incorporado en Raspberry Pi. Alternativamente y para completar procesos de instalación o posteriores usos con el dispositivo, serán necesarios un teclado y un ratón conectados a los dos puertos USB disponibles. Por último, mediante un conector micro USB proporcionaremos alimentación a la Raspberry que se encenderá inmediatamente sin pulsar ningún tipo de botón.

Una vez se visualice la interfaz en el monitor, se mostrará una lista con los sistemas operativos disponibles, en este caso, Raspbian que es el sistema precargado recomendado, se seleccionará esta opción y se procederá con la instalación. Este proceso puede tardar varios minutos, incluso una hora. Una vez finalizada la instalación, el dispositivo se reiniciará y una vez reanudado se mostrará el escritorio de Raspberry Raspbian (figura 4.3).



**Figura 4-3** Escritorio Raspberry Pi

## 4.4 Instalación Snowboy

Snowboy es un motor de detección de palabras claves que se caracteriza por ser personalizable, ya que se puede crear y entrenar un modelo propio, con el cual tener una alta precisión y ser compatible en múltiples plataformas, como:

- Todas las versiones de Raspberry Pi (con Raspbian)
- Mac Os X 64 bits
- Ubuntu 64 bits
- IOS
- Android

Para poder utilizar la herramienta Snowboy en la plataforma seleccionada es necesario instalar las librerías de Portaudio (figura 4.5) y Pyaudio (figura 4.4) para Python. Estas dos librerías son necesarias para la entrada y la salida de audio.

```
sudo apt -get install python -pyaudio python3 -pyaudio sox
```

**Figura 4.4:** Comando instalación pyaudio

```
sudo pip install pyaudio
```

**Figura 4.5:** Comando instalación portaudio

Por otro lado, la página de Snowboy nos proporciona una serie de demos donde probar el correcto funcionamiento del modelo creado y una base con la que empezar a trabajar sobre la aplicación del asistente personal. Estas demos son sobre código Python y se encuentran en varias plataformas como:

- Ubuntu 64 bit
- Mac OS X
- Raspberry PI (1,2,3,zero)

## 4.5- Instalación PIP

Pip es un sistema de gestión de paquetes de Python que se utilizará para la instalación de paquetes y librerías sobre código Python (figura 4.6). Pip ya está instalado si emplea Python2 ( $\geq 2.7.9$ ) o Python3 ( $\geq 3.4$ ), por lo que solo será necesario actualizarlo en caso de ya estar preinstalado.

```
sudo pip install -U pip
```

**Figura 4.6:** Comando instalación pip

Para instalar cualquier paquete necesario a través de pip bastará con lanzar el comando de la figura 4.7 añadiendo el nombre del paquete correspondiente.

```
sudo install <nombre_paquete>
```

**Figura 4.7:** Comando instalación paquetes pip

## 4.6- Instalación SPEECH-TO-TEXT

### 4.6.1- Requisitos

Esta biblioteca permite el reconocimiento por voz con soporte para numerosos motores como y APIs como:

- Reconocimiento de voz de Google (el cual se implementará en este proyecto)
- API de Google Cloud Speech

- Reconocimiento de voz de Microsoft Bing

Para la instalación de esta biblioteca son necesarios unos requisitos mínimos en el sistema:

- Python 2.6 o superior
- Pyaudio 0.2.11, ya que el dispositivo empleará un micrófono como salida de audio, en caso contrario, se deberá tener PocketSphinx.
- Encoder FLAC (Si el sistema no trabaja sobre Windows, Linux o Os X)

Puede que el micrófono no detecte la voz adecuadamente y se pueda obtener un error de que no se está captando la voz. Este error suele darse en las Raspberry Pi que no tienen un puerto de entrada de audio. Para solucionar este error es necesario actualizar el valor de `Microphone()` `Index`, para ello es necesario averiguar el índice mediante el siguiente comando (figura 4.8):

```
print("microfono con nombre with name \ “{1} \” encontrado para  
microfono(device_index={0}) “.format(index,name))
```

**Figura 4.8:** Comando comprobación micrófonos activos

Este comando devolverá al usuario una lista con los micrófonos encontrados, bastará con dar el índice del valor encontrado a la variable anterior con el comando de la figura 4.9.

```
Microphone(device_index = <valor del micrófono USB>)
```

**Figura 4.9:** Comando activar micrófono USB

### 4.6.2- Instalación

Para realizar la instalación de la biblioteca, a continuación, se muestra el comando, en la figura 4.10 con el cual será posible realizar la instalación del módulo que permita al dispositivo ejecutar el paso de voz a texto para así poder empezar a tratar la consulta del usuario.

```
sudo pip install SpeechRecognition
```

**Figura 4.10:** Comando instalación Speech Recognition

Mediante la función de la figura 4.11, se entra en bucle try por el cual se ajusta el ruido para una mayor precisión en la captura del audio. Realizando la llamada `audio=r.listen(source)`, el micrófono capta la voz del usuario, la guarda en una variable con la cual hace la llamada a `recognize_google` para realizar el paso a texto y la imprime por consola para realizar una comprobación de que la captura ha sido correcta. Si la voz que se ha grabado no fuera correcta, es decir, que tan solo se haya captado ruido a través del micrófono o los servidores de Google no fueran capaces de proporcionar la cadena de regreso, o bien, que el prototipo no fuera capaz de tratar la consulta que le ha realizado el usuario, el dispositivo haría una notificación al usuario para que vuelva a realizar la consulta anterior y poder proporcionarle una respuesta.

```
r = sr.Recognizer()
m= sr.Microphone ()
try:
    with m as source: r.adjust_for_ambient_noise(source)
        with m as source:
            audio=r.listen(source)
            try:
                value = r.recognize_google(audio)
                if str is bytes:
                    printf (u"Has dicho: {}".format(value).encode("utf-8"))

                    entidad= wit_response(value)
                    entity_recognize(entidad)
            except sr.UnknownValueError:

                play_audio_file(ERROR_NOTIF)
                printf("Fallo en la grabación... Repita")

        except sr.RequestError as e:

            printf ("No se pueden recoger los resultados del servicio
            Google Speech Recognition; {}".format(e))

except KeyboardInterrupt:
    pass
```

Figura 4.11: Código Speech Recognition

## 4.7- Instalación WIT.AI

Para usar esta herramienta, en primer lugar, es necesario registrarse (figura 4.12), existen dos posibilidades para ello, tanto por una cuenta propia de Facebook o por una de GitHub.

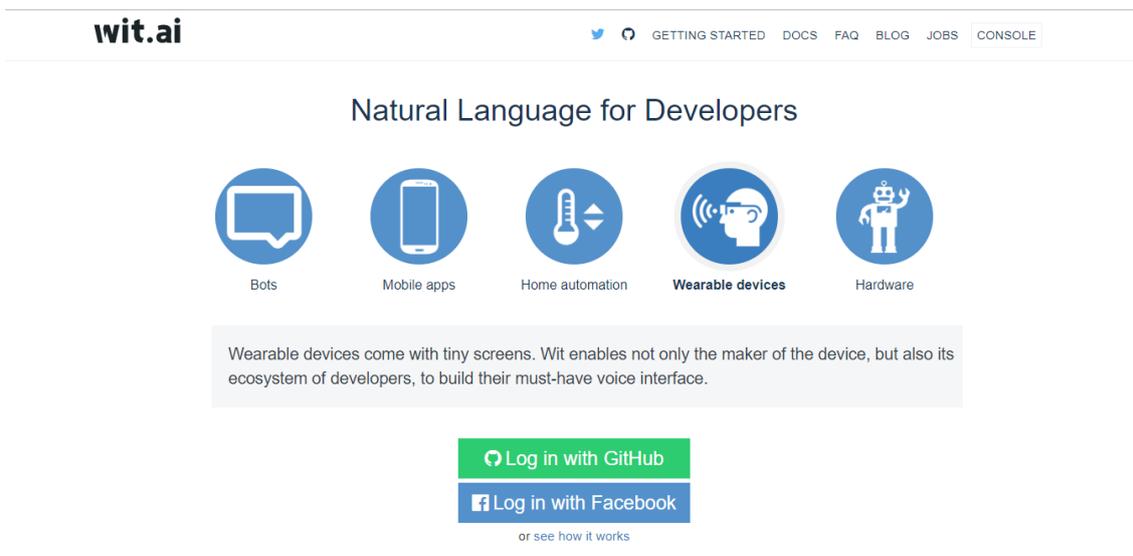


Figura 4-12 Registro Wit.ai

Una vez registrados y dentro de la herramienta, esta permitirá al usuario crear su primera aplicación, a partir de la cual se podrá empezar a trabajar.

En primer lugar, es necesario realizar un entrenamiento, para ello se introducen frases en la interfaz de Wit y se seleccionan las palabras que van a ser importantes para trabajar con ellas, estas palabras son las denominadas entidades o entities, tal y como se muestra en la figura 4.13.

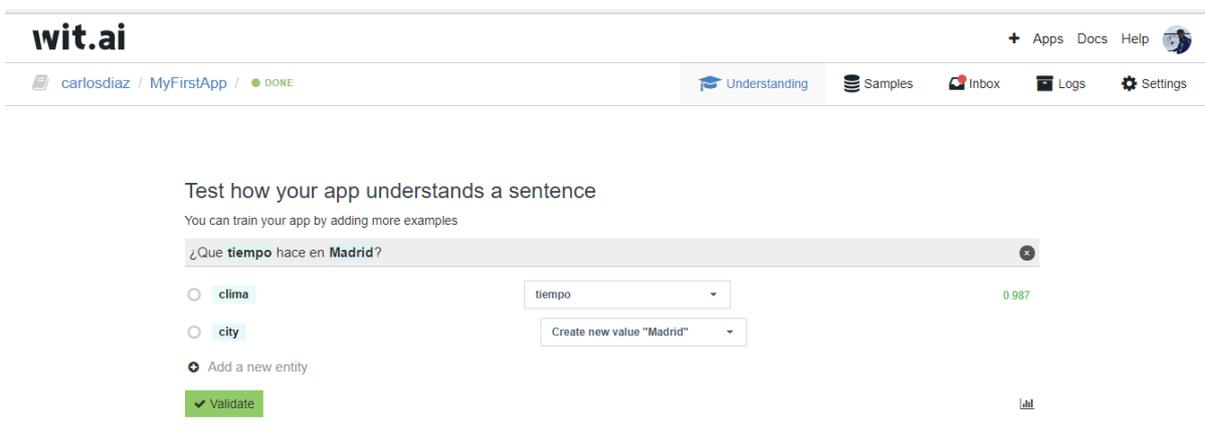


Figura 4-13 Entrenamiento Wit.ai

A modo de ejemplo, la frase ¿Qué tiempo hace en Madrid?, la herramienta obtendrá dos entidades, la palabra “tiempo” la detectará con el tipo de clima, weather u otro que haya creado el usuario y otra entidad denominada ciudad o city, con el campo en su interior del nombre de la ciudad, en este caso Madrid. Para obtener estas entidades como código y poder trabajar con las respuestas que nos proporciona Wit.ai, es necesario llamar al método get\_entities() y get\_value() de la propia API de Wit, que devolverán al usuario las entidades y su valor, si se requiere para trabajar con él, del ejemplo anteriormente mencionado o de cualquier otra frase entrenada previamente.

La lista de entidades para este dispositivo es la siguiente (tabla 4.1):

Entidad	Función
Hora	Para solicitar la hora actual como consulta Ej: ¿Qué <u>hora</u> es?
Clima	Para solicitar el clima de una determinada ciudad Ej: ¿Qué <u>tiempo</u> hace en Madrid?
Ciudad	Acompaña a la consulta anterior y su valor determina la ciudad donde se consulta. Ej: ¿Qué tiempo hace en <u>Madrid</u> ?
Noticias	Para solicitar la lectura de un determinado periódico. Ej: Leer las últimas <u>noticias</u> de El País
Periódico	Acompaña a la consulta anterior y su valor determina el periódico que se debe consulta. Ej: Leer las últimas noticias de <u>El País</u>
Correo	Para solicitar la lectura de los últimos correos de la bandeja de entrada. Ej: ¿Tengo <u>correo</u> nuevo?

**Tabla 4.1** Lista de entidades

Posteriormente, y hablando del código, la aplicación debe tratar las entidades dentro de una frase, como se ha comentado anteriormente, para ello, mediante el bucle if de la figura 4.14 el propio dispositivo es capaz de diferenciar tipo de consulta se le está realizando, a modo de inteligencia artificial. Si dentro del conjunto de entidades encontramos una variable de tipo “clima”, quiere decir que el usuario ha solicitado saber qué tiempo hace en una determinada ciudad, o bien si dentro de este conjunto de entidades se encuentra una variable de tipo “time”, el usuario habría solicitado saber la hora actual y se llamará a la función correspondiente para realizar la consulta adecuada.

```
def entity_recognize(conj_entidades):  
    if 'clima' in conj_entidades:  
        clima_recognize(conj_entidades)  
  
    elif 'mail' in conj_entidades:  
        getSubject()  
  
    if 'time' in conj_entidades:  
        time_recognize(conj_entidades)  
  
    if 'news' in conj_entidades:  
        news_recognize(conj_entidades)  
  
    if 'close' in conj_entidades:  
        close_recognize(conj_entidades)
```

Figura 4.14: Código Wit.ai

## 4.8- Instalación TEXT-TO-SPEECH

Por último, para poder realizar los cambios de texto a voz para permitir hablar al asistente con el usuario y así ofrecer la respuesta de las consultas a las que ha sido sometido, en primer lugar, hay que instalar la biblioteca necesaria, se pueden encontrar varias opciones para realizar esta operación. Se puede optar por:

- La librería propia de Python tts
- Pytts, un contenedor multiplataforma de texto a voz
- API de texto a voz que ofrece Google

Para esta implementación se ha decidido utilizar la opción de Google debido a su mejor calidad. Para realizar su instalación será necesaria la siguiente instrucción por línea de comando (figura 4.15).

```
Sudo pip install gTTS
```

**Figura 4.15:** Instalación Google text to speech

En segundo lugar, es necesario importar la biblioteca gTTS (figura 4.16), es decir, la biblioteca text to speech que ofrece Google para Python.

```
from gtts import gTTS
```

**Figura 4.16:** Importar librería Google text to speech

Y se creará una función con la que pasando un texto sea capaz de traducirla a voz para reproducírsela al usuario que se encuentra esperando una respuesta final. Para ello también debe indicarse el idioma en el que se debe dar la respuesta por voz (figura 4.17).

```
lang='es'  
  
def tts(text,lang):  
    tts = gTTS(text=text, lang=lang)  
    tts.save("respuesta.mp3")  
    os.system('mpg321 respuesta.mp3')
```

**Figura 4.17:** Código Google text to speech

## 4.9 Consultas implementadas

### 4.9.1- Consulta time

En primer lugar, para la consulta de la hora actual, se necesitará una llamada `time()` al propio sistema, el cual nos devolverá la hora solicitada, ya que ese es el objetivo requerido. Para ello se debe importar la librería `time` (figura 4.18) para poder acceder a la información actual del sistema.

```
Import time
```

Figura 4.18: Importar librería `time`

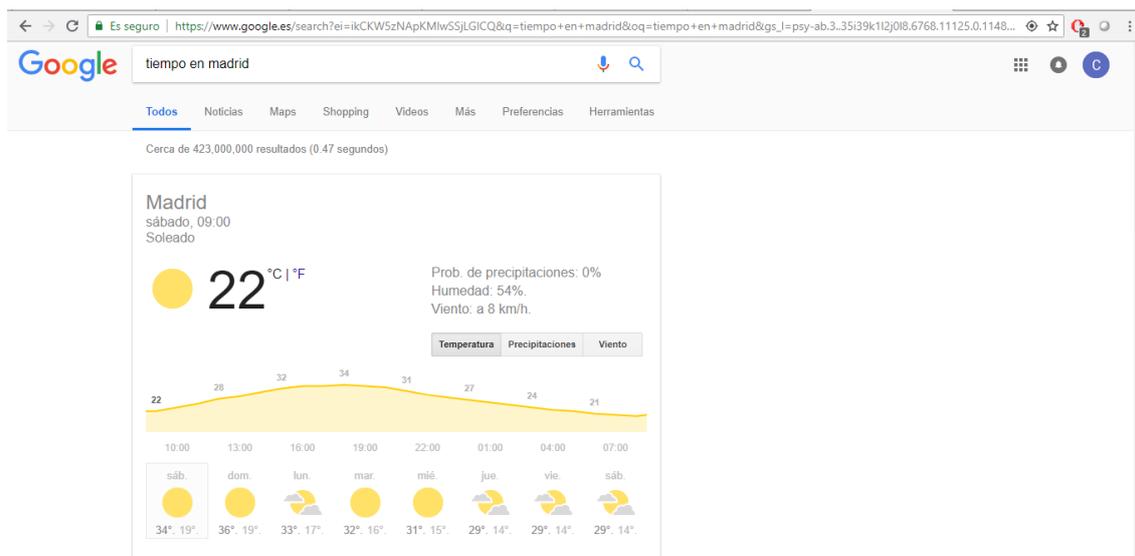
Y bastará con introducir en el código el formato en el que se desea recibir, para después realizar la llamada al convertidor de texto a voz como se observa en la figura 4.19.

```
lang='es'  
def time_recognize(conj_entidades):  
    hora='Son las: '+time.strftime("%H:%M:%S")  
    tts(hora,lang)  
    print(hora)
```

Figura 4.19: Código función `time`

### 4.9.2- Consulta tiempo climático

Para la consulta del clima en cualquier ciudad, se obtiene la URL de una consulta a Google. En esa URL se busca el clima y la temperatura de la ciudad sugerida por el usuario. Esa dirección web se forma a partir de la variable “`city`”, dentro del conjunto de entidades que han sido recibidas y se introduce dentro de la url, donde se accederá a una página web del estilo de la figura 4.20.



**Figura 4-20** Consulta Google del clima

Dentro de esta página, son necesarios dos campos para devolver al usuario, la temperatura y el estado climático. Para obtener dichos campos se hace uso de la librería de Python, BeautifulSoup. Esta librería se utiliza para analizar documentos HTML, creando un árbol con los elementos que este tipo de documentos contiene para así poder extraer la información necesaria, es decir, tiene la función de web scaping. Por lo tanto, mediante esta librería se da posibilidad al dispositivo de obtener los dos valores para devolverlos al usuario.

En caso de que el usuario no de una ciudad en la consulta se accederá a la información de Leganés, por defecto, ya que es el lugar donde será presentado el proyecto (figura 4.21).

```

def clima_recognize(con_entidades):

    if 'city' in con_entidades:

        city = entity_value(con_entidades,'city')
    else:

        city = 'Leganés'

        url =
'https://www.google.es/search?q='+city+'+weather&oq='+city+'+weather&aqs=chrom
e..69i57j69i60j013.3984j1j7&sourceid=chrome&ie=UTF-8'
        response = requests.get(url)
        soup = BeautifulSoup(response.text, 'lxml')
        titulo= 'El '+soup.h3.text+' es: '
        print (titulo)
        g=soup.find('div',class_="g")
        y=soup.find('span',class_='wob_t')
        grados= y.text
        print (grados)
        clima=soup.find('img').get('title')
        total=titulo+clima+'con una temperatura de '+grados
        tts(total,lang)
        print (clima)

```

Figura 4.21: Código función tiempo climático

### 4.9.3- Consultar periódicos

En el caso de que el usuario quiera leer las últimas noticias, deberá indicar que periódico desea consultar cuando realiza la consulta, con el fin de crear una aplicación más sencilla, manejable, veloz y evitar interaccionar en demasiadas ocasiones con el dispositivo y no convertir la conversación en una serie de preguntas y respuestas haciéndolo más pesado para el usuario. Para ello, el asistente consulta paginas RSS del periódico indicado, crea una url que le da acceso solo al periódico o diario indicado por el usuario, recorre la página web obtenida y lee los cinco primeros titulares para ofrecérselos al usuario, ya que pueden ser los de mayor actualidad o los más destacables para esa entidad (figura 4.22).

```
def news_recognize(con_entidades):
    periodico = entity_value(con_entidades,'newspaper')
    if periodico=='pais':
        i=0
        text='Los titulares del pais son: '
        tts(text,lang)
        url = 'http://ep00.epimg.net/rss/elpais/portada.xml'
        rss = feedparser.parse(url)

    elif periodico=='marca':
        i=0
        text='Los titulares del marca son: '
        tts(text,lang)
        url = 'http://estaticos.marca.com/rss/futbol/primera-division.xml'
        rss = feedparser.parse(url)

    for post in rss.entries:
        if i <5:
            text=post.title
            tts(text,lang)
```

Figura 4.22: Código función lectura de titulares

#### 4.9.4- Consulta correo electrónico

Como última acción, encontramos la consulta sobre el correo electrónico del usuario, se ha decido consultar la aplicación Gmail, ya que puede ser la más utilizada por el usuario. Para que el dispositivo pueda acceder a la cuenta personal del usuario son necesarios unos pasos previos.

Como requisitos son necesarios:

- Versión de Python 2.6 o superior.
- La herramienta PIP.
- Una cuenta personal de email, Gmail para este documento.

### Paso 1: activar API de Gmail.

Para ello se creará un nuevo proyecto dentro de la herramienta que ofrece Gmail. Se crea un nuevo proyecto y se introduce un nombre (figura 4.23).

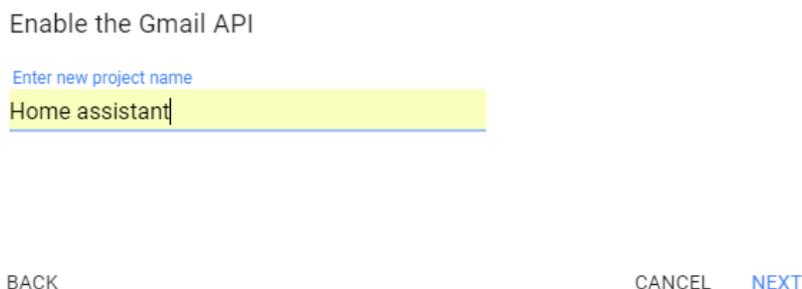


Figura 4-23 Activación de la API de Gmail

Después se descargará el archivo de configuración JSON (figura 4.24), que debe ser movido al directorio de trabajo con el nombre: credentials.json.

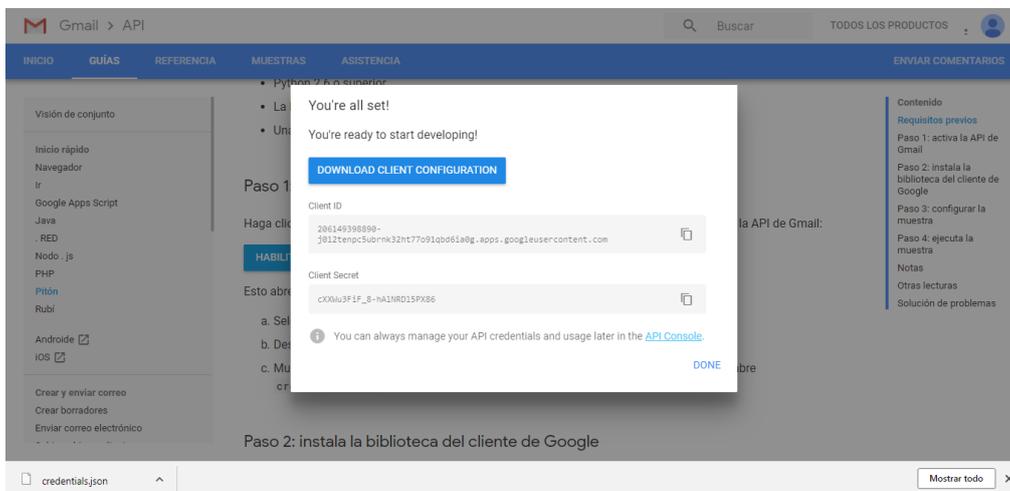


Figura 4-24 Archivo de configuración JSON

### Paso 2: instalar la biblioteca del cliente de Google

Es necesario instalar la biblioteca de Gmail para poder implementar el código del dispositivo posteriormente, mediante el siguiente comando (figura 4.25):

```
Sudo pip install --upgrade google-api-python-client oauth2client
```

**Figura 4.25:** Comando instalación del cliente Gmail

Por otro lado, para poder implementar el prototipo, se necesitan importar las librerías necesarias en el código para poder conseguir el resultado (figura 4.26).

```
from apiclient import discovery
from oauth2client import client
from oauth2client import tools
from oauth2client.file import Storage
from apiclient import errors
```

**Figura 4.26:** Importar librerías Gmail

En las siguientes capturas de código (figura 4.27, 4.28, 4.29, 4.30 y 4.31) se pueden encontrar las funciones que hacen posible acceder al correo personal del usuario y obtener de él, posible nuevos correos electrónicos y su procedencia.

```
SCOPES = 'https://www.googleapis.com/auth/gmail.readonly'
CLIENT_SECRET_FILE = 'client_secret.json'
APPLICATION_NAME = 'Gmail API Python Quickstart'

def get_credentials():

    home_dir = os.path.expanduser('~')
    credential_dir = os.path.join(home_dir, '.credentials')
    if not os.path.exists(credential_dir):
        os.makedirs(credential_dir)
    credential_path = os.path.join(credential_dir,
                                   'gmail-python-quickstart.json')

    store = Storage(credential_path)
    credentials = store.get()
    if not credentials or credentials.invalid:
        flow = client.flow_from_clientsecrets(CLIENT_SECRET_FILE, SCOPES)
        flow.user_agent = APPLICATION_NAME
        if flags:
            credentials = tools.run_flow(flow, store, flags)
        else: # Needed only for compatibility with Python 2.6
            credentials = tools.run(flow, store)
        print('Storing credentials to ' + credential_path)
    return credentials
```

**Figura 4.27:** Código función obtener credenciales

En la figura 4.27 se puede apreciar el código necesario para acceder al correo personal del usuario. Para ello se hace uso del fichero JSON con el que se habilita la API de Gmail para el proyecto creado, y se le denomina un alcance o “scope”. Para esta aplicación tan solo se van a leer correos, no será necesario crear, enviar o eliminar mensajes o hilos, por lo tanto, será suficiente con mantener un alcance de “readonly”. En caso de necesitar alguna otra opción o implementación se debe consultar la tabla 4.2 a continuación.

readonly	Lee todos los recursos y sus metadatos, sin operaciones de escritura.
Compose	Crear, leer, actualizar y eliminar borradores. Enviar mensajes y borradores.
Send	Enviar solo mensajes. No leer o modificar privilegios en el buzón.
Insert	Inserta e importa solo mensajes.
Labels	Crear, leer, actualizar y eliminar etiquetas solo.
Modify	Todas las operaciones de lectura / escritura excepto la eliminación inmediata y permanente de conversaciones y mensajes, sin pasar por la Papelera.
Settings. Basic	Administrar la configuración básica de correo
Settings. Sharing	Administrar las configuraciones de correo sensibles, incluidas las reglas y los alias de reenvío.
	Acceso completo a la cuenta, incluida la eliminación permanente de hilos y mensajes. Este alcance solo debe solicitarse si su aplicación necesita eliminar hilos y mensajes de forma inmediata y permanente, sin pasar por la Papelera.

**Tabla 4.2:** Comandos accesos Gmail

A continuación, en la figura 4.28 se implementa la función `GetMessage`, con la que se obtiene un mensaje de la bandeja de entrada a partir del identificador del usuario y el identificador del mensaje que se está buscando. Si algún error fuese encontrado, esta función devolvería un error al no encontrar un mensaje con dicho identificador.

```
def GetMessage(service, user_id, msg_id):  
  
    try:  
        message = service.users().messages().get(userId=user_id, id=msg_id).execute()  
  
        print ('Message snippet: %s' % message['snippet'])  
  
        return message  
    except errors.HttpError, error:  
        print ('An error occurred: %s' % error)
```

**Figura 4.28:** Código función obtener mensaje

En esta otra función, denominada `GetMimeMessage`, (figura 4.29) se obtiene el mensaje en su versión MIME, es decir, se pueden comprobar sus cabeceras y no solo el cuerpo del mensaje que pide como parámetro mediante su identificador. En caso de no encontrar un mensaje con dicho identificador, devolverá un error para notificárselo al usuario.

```
def GetMimeMessage(service, user_id, msg_id):  
  
    try:  
        message = service.users().messages().get(userId=user_id,  
id=msg_id,format='raw').execute()  
  
        print ('Message snippet: %s' % message['snippet'])  
  
        msg_str = base64.urlsafe_b64decode(message['raw'].encode('ASCII'))  
  
        mime_msg = email.message_from_string(msg_str)  
  
        return mime_msg  
    except errors.HttpError, error:  
        print ('An error occurred: %s' % error)
```

**Figura 4.29:** Código función obtener mensaje MIME

Otra de las funciones necesarias para esta implementación es `ListMessagesWithLabels` (figura 4.30). Mediante esta función se obtienen los campos dentro del correo como destinatario, procedencia, dominio del correo, etc. Gracias a esta función se puede identificar quien envió el correo para poder notificárselo al usuario o si el correo ya ha sido leído anteriormente, con lo que poder notificar al usuario que realizó la petición que correos están aún sin abrir y su procedencia.

```
def ListMessagesWithLabels(service, user_id, label_ids=[]):  
  
    try:  
        response = service.users().messages().list(userId=user_id,labelIds=label_ids).execute()  
        messages = []  
        if 'messages' in response:  
            messages.extend(response['messages'])  
            while 'nextPageToken' in response:  
                page_token = response['nextPageToken']  
                response = service.users().messages().list(userId=user_id,labelIds=label_ids,pageToken=page_token).execute()  
                messages.extend(response['messages'])  
  
        return messages  
    except errors.HttpError, error:  
        print ('An error occurred: %s' % error)
```

**Figura 4.30:** Código función obtener etiquetas del mensaje

Por último, se puede observar la función `getSubject` (figura 4.31). Este se trata de la función principal para esta consulta. Apoyándose en las funciones anteriormente descritas, `getSubject()` accede a las credenciales del usuario para poder tener acceso al correo personal del usuario y poder realizar la búsqueda de los correos necesarios. Para ello recorre y busca las etiquetas de cada mensaje de la bandeja de entrada, es decir, filtra por dos etiquetas, la etiqueta 'important' que accede a la bandeja principal, y la etiqueta 'unread' que informa de que ese mensaje aun no ha sido abierto ni leído por el usuario.

Una vez los mensajes han sido filtrados se almacenan en una variable para poder comunicar al usuario el número exacto de correos sin leer y se informa además del contacto que envió dicho mensaje, obteniéndolo a partir de su etiqueta 'from'.

```
def getSubject():
    credentials = get_credentials()
    http = credentials.authorize(httplib2.Http())
    service = discovery.build('gmail', 'v1', http=http)
    results = service.users().labels().list(userId='me').execute()
    labels = results.get('labels', [])

    if not labels:
        print('No labels found.')
    else:
        print('Labels:')
        for label in labels:
            print(label['name'])
    message=GetMessage(service, 'me', '161be143f323ef28')
    print ('Los siguientes son los no leidos')
    noLeidos=ListMessagesWithLabels(service, 'me', label_ids=['IMPORTANT','UNREAD'])

    print ('numero de no leidos: '+ str(len(noLeidos)))
    tts("Tienes '+str(len(noLeidos))+' mensajes nuevos de','es')
    i=0
    for i in range(len(noLeidos)):

        mime=GetMimeMessage(service, 'me', str(noLeidos[i]['id']))
        print ('Enviado por: ' + mime['from'])
        text=mime['from']
        tts(text,'es')
```

**Figura 4.31:** Código consulta leer correos electrónicos

# Capítulo 5

## Marco socioeconómico

### 5.1- Análisis del entorno socioeconómico

A continuación, se procederá a describir el marco socioeconómico de este proyecto. Se analizará desde el punto de vista de los hogares y de cómo los usuarios perciben este tipo de sistemas en sus hogares. También se observará el crecimiento de los asistentes para el hogar en los años futuros y su impacto en la sociedad.

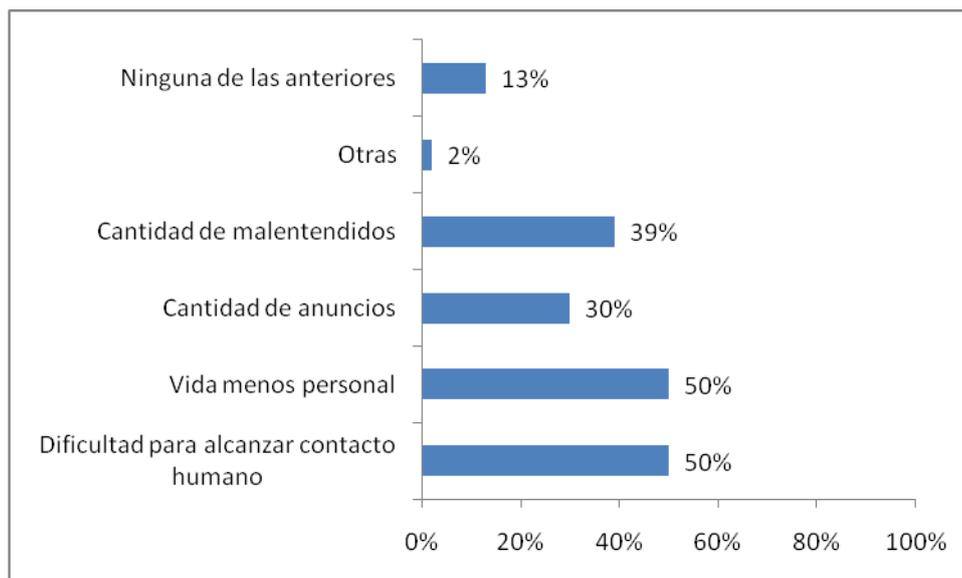
Los asistentes virtuales para el hogar han sufrido un fuerte crecimiento en los últimos años y pasar de ser una simple idea de robotizar un hogar a una realidad al alcance de cualquier usuario que se lo proponga. Sin embargo, este crecimiento puede y podría ser mucho mayor si se eliminaran algunas barreras, como eliminar las dudas de los clientes y usuarios que pretender involucrarse en este tipo de productos. Los usuarios conocen la potencia y todas las funciones que pueden realizar los asistentes para hacerles más fácil su rutina y su vida en el hogar, pero siguen percibiendo amenazas con estos productos. Los principales motivos son los siguientes [22]:

- Privacidad: es sin duda uno de los factores que más preocupa a la sociedad, cómo de seguro es tener un sistema que esté continuamente escuchando lo que ocurre en su entorno, y cómo de seguro es que la voz del usuario se envíe a un

servidor y quede grabada. Para evitar este tipo de problemas existen leyes que protegen al usuario frente a este tipo de amenazas.

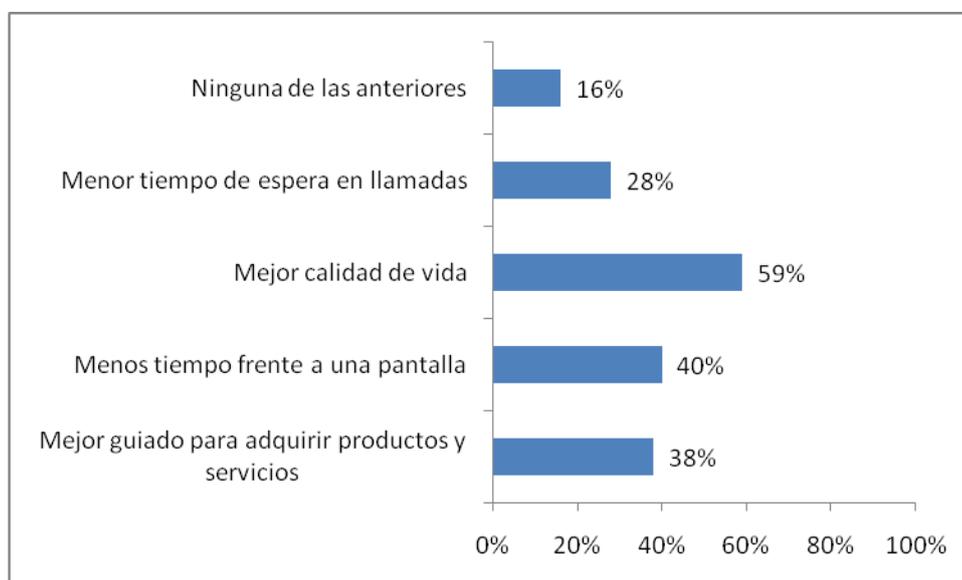
- El contacto humano: los asistentes virtuales son vendidos como sistemas capaces de interactuar con los humanos, capaces de dar respuestas y de mantener conversaciones con nosotros. Pero muchos de los usuarios no ven agradable el entablar una conversación con una máquina y no percibir esa calidad humana, creen que aún queda muy lejos que una máquina pueda sustituir a un humano o comportarse como él, por lo tanto, esto no sustituye el contacto humano.
- Malentendidos: En los últimos años se han conseguido numerosos avances en la mejora de los asistentes y técnicas como de machine learning, cada vez los asistentes domésticos son más inteligentes y tiene una mayor utilidad, pero los usuarios creen que no son del todo inteligentes y puede llevar a malentendidos, como puede ser a la hora de realizar una compra. Un ejemplo es la confusión que se produjo en una cadena de televisión, se estaba dando la noticia de cómo una niña pequeña había comprado sin ningún consentimiento una casa de muñecas a través de Amazon Echo, mientras se daba la noticia se reprodujo la consulta de la niña: “Alexa, quiero una casa de muñecas”, esto provocó que los asistentes de los hogares donde se estaba dando la noticia reconocieron la voz como una petición y también realizaron la compra. Sin embargo, este tipo de malentendidos se pueden solucionar con las técnicas actuales de Voice Match con la que el asistente solo reconoce las voces de los usuarios que tiene guardados, es decir, tan solo de los miembros del hogar.

Según un estudio realizado en Estados Unidos sobre cuáles son los mayores riesgos que perciben los usuarios sobre los asistentes virtuales (tabla 5.1), se encuentran reflejados en el siguiente gráfico. Como mayor riesgo se ven la dificultad del contacto humano junto a la pérdida de una vida más personal, en segundo lugar, los posibles malentendidos, y en tercer lugar el incremento de publicidad.



**Tabla 5-1** Respuesta encuesta riesgos asistentes

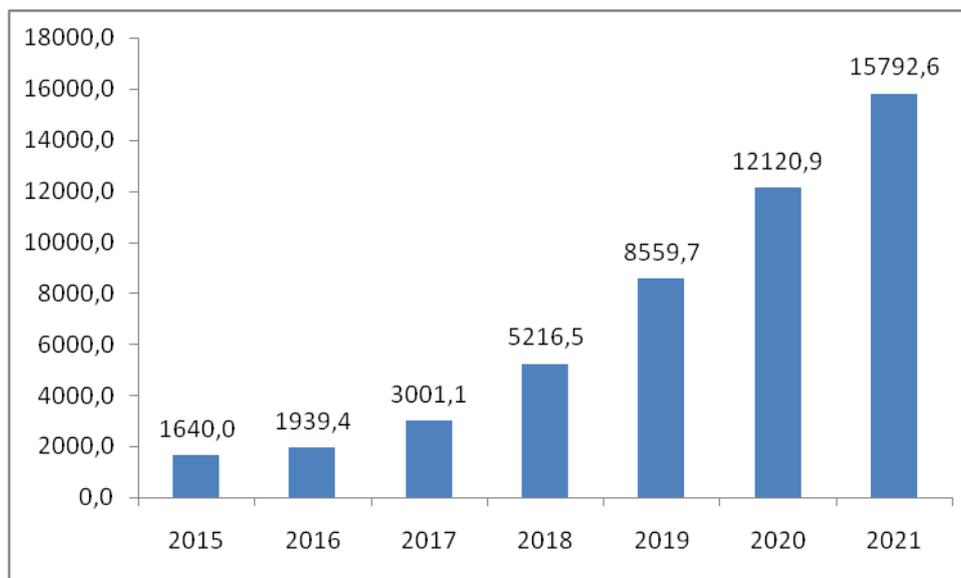
Por consiguiente, los beneficios que encuentran los usuarios según el mismo estudio (tabla 5.2) son, en primer lugar, una mayor calidad de vida, menos tiempo delante de una pantalla y la orientación al realizar compras.



**Tabla 5-2** Respuesta encuesta beneficios asistentes

## 5.2 Evolución de los asistentes virtuales y tendencias futuras

Cada vez son más demandados y conocidos estos tipos de asistentes para el hogar, de hecho, se espera que su tendencia aumente exponencialmente en los próximos años, con lo que aumentará su valor en el mercado, como se puede observar en la tabla 1.5



**Tabla 5-3** Volumen valor mercado por millón asistentes inteligentes (US Dólares)

Se puede estar hablando de doblar su volumen en dos años, es decir, crecer desde 8.559 millones de dólares a 12.120 millones, incluso de triplicar su valor en tres años, en la franja desde el año 2018 al 2021.

Según Adam Cheyer, cofundador de Siri, en una entrevista para Accenture Digital Conference (ADC), “Dentro de poco, los asistentes de voz serán mas importantes que los smartphones o que buscadores web como Google” y se desconoce aún la capacidad y potencia que tendrán estos sistemas dentro de dos o tres años. También pronosticó que en los próximos años los usuarios tendrán a su disposición asistentes que realicen funciones más complicadas, no solo las sencillas como regular la temperatura o encender las luces de una habitación, si no que se tendrá un asistente que te conozca y te ayude a resolver problemas. Conocerá al usuario, es decir, sus gustos, marcas u otras preferencias a la hora de, por ejemplo, realizar compras.

Como ejemplo de estos futuros nuevos asistentes, ya se pueden ver algunos cambios importantes, como para la futura “Siri” de Google, que será capaz de reservar hora en una peluquería manteniendo una conversación con la recepción del local sin necesidad de interacción humana, accediendo a nuestra agenda sabrá que espacios tenemos libres en nuestro día a día y será capaz de dar respuesta a las preguntas de recepción [23] [24].

### **5.3- Conclusión al análisis del entorno socioeconómico**

La utilización de los asistentes virtuales supone un gran avance y mejora en el día cotidiano de los usuarios, les ayuda a la realización de sus tareas. Pero todo su potencial no queda ahí, la evolución de los asistentes para el hogar supone una revolución en la vida de los humanos y las empresas que trabajan en el desarrollo de estos dispositivos. Bien es cierto que no todos los usuarios comparten la idea de tener un asistente para el hogar, pero en ellos se encuentran un gran número de beneficios, no solo ahora si no en un futuro, ya que los asistentes virtuales conocerán al usuario personalmente, sus gustos y preferencias, podrán realizar llamadas o realizar compras sin ningún tipo de interacción humana.

# Capítulo 6

## Planificación y presupuesto

### 6.1- Planificación

El método de desarrollo llevado a cabo para este trabajo fin de grado ha sido el de división por fases del proyecto en la que cada una se divide en tareas.

- **Fase 1: Planificación**

Antes del comienzo del proyecto, es necesario establecer duraciones y divisiones en posibles fases del proyecto.

- Alcance: Para determinar las funcionalidades que va a tener el sistema y sus características.
- Planteamiento: Donde se establecen las distintas fases del proyecto, sus duraciones y las posibles herramientas y recursos necesarios para su elaboración.

- **Fase 2: Investigación**

Esta fase está basada en la búsqueda de aplicaciones similares como la propuesta y tecnologías necesarias con su investigación particular para elaborar su propio estudio.

- Búsqueda de sistemas similares: Para comparar funcionalidades y como se distribuyen en el mercado dichos sistemas.
- Estudio de las tecnologías: Un estudio de las tecnologías necesarias para la elaboración del proyecto y posibles alternativas comparando unas con otras para encontrar la mejor solución.
- Redacción del estudio final de las tecnologías y recursos a desarrollar.

- **Fase 3: Desarrollo y pruebas**

Fase de implementación del sistema con las correspondientes pruebas establecidas en su planificación.

- Desarrollo e implementación del sistema: Desarrollo de las tecnologías seleccionadas para implementar un sistema como el propuesto
- Elaboración plan de pruebas: Necesario para testear el sistema implementado. Se elaboran una serie de pruebas con el cual probar su funcionalidad.

- **Fase 4: Documentación**

Como última fase, se lleva cabo la documentación del proyecto, donde se elabora una memoria del sistema realizado.

- Elaboración de la estructura de la memoria: Para fijar una organización con los diferentes capítulos dentro de la memoria con su correspondiente fecha de finalización.

- Redacción de la memoria: Una vez finalizadas las pruebas y comprobada su correcta funcionalidad, se lleva a cabo la redacción de la memoria final junto con el estudio de investigación anterior.
- Elaboración de la presentación: Presentación para describir y defender el trabajo fin de grado.

• **Diagrama de Gantt**

Una vez definidas las fases y tareas es necesario planificarlas y a lo largo del periodo determinado, para ello se lleva a cabo la realización del diagrama de Gantt de la tabla 1.1. El periodo de tiempo consta desde febrero de 2018 hasta 25 de septiembre de 2018.

	ENERO				FEBRERO				ABRIL				MAYO			
Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<b>FASE 1: PLANIFICACIÓN</b>																
Alcance																
Planificación																
<b>FASE 2: INVESTIGACIÓN</b>																
Busqueda sistemas																
Estudio tecnologías																
Redacción estudio																
<b>FASE 3: DESARROLLO</b>																
Módulo Snowboy																
Módulo tratamiento voz																
Módulo stt																
Módulo Wit.ai																
Módulo tts																
Plan de pruebas																
<b>FASE 4: DOCUMENTACIÓN</b>																
Estructura memoria																
Redacción memoria																
Presentación																

	JUNIO				JULIO				AGOSTO				SEPTIEMBRE			
Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<b>FASE 1: PLANIFICACIÓN</b>																
Alcance																
Planificación																
<b>FASE 2: INVESTIGACIÓN</b>																
Busqueda sistemas																
Estudio tecnologías																
Redacción estudio																
<b>FASE 3: DESARROLLO</b>																
Módulo Snowboy																
Módulo tratamiento voz																
Módulo stt																
Módulo Wit.ai																
Módulo tts																
Plan de pruebas																
<b>FASE 4: DOCUMENTACIÓN</b>																
Estructura memoria																
Redacción memoria																
Presentación																

Tabla 6.1: Diagrama de Grantt

## 6.2- Presupuesto

En este apartado se describen los costes del proyecto tanto materiales como de personal.

### Costes materiales

- Portatil: Lenovo ideapad 300.
- Raspberry Pi 2.
- Micrófono.
- Conector HDMI
- Material de oficina.
- Alquiler de oficina.
- Gastos de oficina (electricidad e Internet).

## Costes personales

- Becario formado en Ingeniería Telemática.

Nombre	Cantidad	Tiempo(meses)	Coste (€)	Total (€)
<b>Costes materiales</b>				
Portatil	1	8	700	700
Raspberry Pi 2	1	8	35	35
Micrófono	1	8	15	15
Material de oficina	Varios	8	30	30
Conector HDMI	1	8	6	6
Alquiler oficina	1	8	300	2400
Gatos oficina	1	8	80	640
				3826
<b>Costes de personal</b>				
Becario	1	8	400	3200
IVA (21%)				1475,46
Coste total				8501,46

**Tabla 6-2** Presupuesto del proyecto

# Capítulo 7

## Conclusiones y trabajos futuros

### 7.1- Conclusiones

En este trabajo de fin de grado se propone el objetivo de diseñar un asistente inteligente para el hogar que sea capaz de funcionar y procesar la información en español.

Después de realizar el estudio del mercado actual de los asistentes inteligentes para el hogar, realizado el diseño del prototipo y la implementación que se ha llevado a cabo empleando para ello una Raspberry Pi basándose en un código Python, se han descrito las funciones que se han implementado para el sistema y por último se ha realizado un análisis tanto del entorno socio económico como del entorno legislativo, se puede llegar a la conclusión de que se ha obtenido el objetivo que se propone para este trabajo fin de grado.

Se ha conseguido desarrollar un sistema gracias a Raspberry Pi y basándose en un código Python, capaz de comunicarse con el usuario y dividida en diferentes módulos, comenzando con su activación mediante la voz del propio usuario para después esperar una orden, procesarla y dar una respuesta a esa consulta en español.

La solución es capaz de dar respuesta a distintas consultas informando al usuario en caso de no haber escuchado bien la voz o la consulta o en caso de no encontrar una solución a dicha orden, bien por problemas en servidores o por cualquier otro factor.

El dispositivo puede acceder paginas o plataformas mediante las credenciales que el usuario le proporciona, manteniendo su privacidad y ajustándose a las normativas vigentes.

Por otro lado, la solución puede ser mejorada en un futuro incluyendo mejoras o nuevas funciones para así conseguir un asistente completamente funcional.

Todo esto llevado a cabo en español, ya que es uno de los requisitos fundamentales para el desarrollo de este proyecto.

La solución no deja de ser una versión reducida o una demo de los asistentes inteligentes lanzados hasta la fecha, pero la solución puede llegar a tener una funcionalidad en el hogar igual que los asistentes inteligentes actuales con el objetivo de ayudar a las personas en la realización de sus tareas rutinarias y hacer tareas que ellos no pueden.

Para conseguir la solución final hubo que pasar por algunas dificultades como la configuración de Raspberry Pi, la instalación de numerosas librerías y complementos, además se ha requerido un alto nivel en programación sobre lenguaje Python que ha requerido un aprendizaje externo. Por otro lado, ha habido que adaptarse a trabajar mediante la voz, un trabajo que no se había realizado anteriormente.

Finalmente, a pesar de pasar por algunas dificultades, el objetivo del proyecto ha sido completado y el trabajo realizado ha servido para aprender nuevas técnicas e incrementar las ya estudiadas anteriormente.

## **7.2 Trabajos futuros**

Como se ha mencionado anteriormente, el proyecto descrito trata de una demo o versión reducida de un asistente inteligente por voz para el hogar, es decir, este prototipo puede ser ampliado con numerosas funciones o mejorar las actuales con el objetivo de ampliar su funcionalidad y ser más atractivo para el usuario. Aparte se

puede trabajar en mejorar el hardware del prototipo como lo altavoces o con un mejor micrófono para poder detectar la voz sin ningún problema, aunque haya ruido en la sala, como de televisores, o si el usuario está en otra habitación.

Entre estas funciones se pueden encontrar todas aquellas relacionadas con reproducción de música, ya que es una de las funciones más utilizadas por los usuarios y que son más llamativas para ellos. Un futuro dispositivo podría reproducir música de las cuentas del usuario como Spotify, Youtube u otras, reproducir las listas guardadas, canciones favoritas o buscar cualquiera que la plataforma tenga disponible. También, se podría incluir la posibilidad de elegir dónde reproducir la música, ya sea en el propio dispositivo, en algún altavoz bluetooth, en televisores del hogar... Con esto se conseguiría que el usuario no tenga que estar buscando de forma manual lo que él quiere reproducir si no que mientras el usuario está realizando cualquier otra tarea del hogar como limpiar o cocinar, puede comunicarse fácilmente con el dispositivo por voz y así evitar interferir en las acciones que el usuario está realizando y hacerla menos monótona.

Por otro lado, y como otra funcionalidad añadida, se podría utilizar el dispositivo para encontrar alertas sobre tráfico o duraciones de trayectos. El usuario puede tener planeado un trayecto ya sea en coche o andando, o simplemente un viaje que está a punto de realizar, antes de salir de su hogar, el usuario podría realizar una consulta mediante la cual, el dispositivo podría alertarle del tráfico que puede encontrarse y proporcionarle una ruta alternativa. También le alertaría del tiempo de esa nueva ruta o de la ruta que iba a realizar por defecto. De tal forma que, a modo de ejemplo, si el usuario vive en Leganés y va a realizar un trayecto a la estación de atocha, antes de salir del hogar el usuario realizaría la consulta, el dispositivo le respondería que la ruta más rápida es por la carretera A-42 con un tiempo estimado de 22 minutos. Sin embargo, puede que para ese día se haya producido algún tipo de problema en la carretera, por lo tanto, la ruta más rápida sería otra, el dispositivo alertaría de ese problema y recomendaría ir por la carretera M-40 con un tiempo estimado de 24 minutos. Con esta implementación se conseguiría algo parecido a un GPS, con el cual recibir alertas y recomendar rutas al usuario.

# Bibliografía

- [1] PlanetaChatbots, «La Ley de Protección de Datos en la era de los chatbots,» [En línea]. Available: <https://planetachatbot.com/ley-proteccion-de-datos-en-chatbots-e0ce40dbb35a>.
- [2] TicBeat, «¿Son seguros para tu privacidad Siri, Alexa, Cortana o Google Assistant?,» [En línea]. Available: <http://www.ticbeat.com/seguridad/estudio-trend-micro-seguridad-asistentes-virtuales/>.
- [3] SoftZone, «Los asistentes virtuales y la privacidad,» [En línea]. Available: <https://www.softzone.es/2016/06/01/los-asistentes-virtuales-y-la-privacidad/>.
- [4] Hipertextual, «Google Home: análisis, opinión y características,» [En línea]. Available: <https://hipertextual.com/analisis/google-home>.
- [5] Xataka, «Google Home Mini: más pequeño, pero con las mismas prestaciones que su hermano mayor,» [En línea]. Available: <https://www.xataka.com/accesorios/google-home-mini-mas-pequeno-pero-con-las-mismas-prestaciones-que-su-hermano-mayor>.
- [6] TechTarget, «¿Que es la inteligencia artificial o AI?,» [En línea]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Inteligencia-artificial-o-AI>.
- [7] Wikipedia, «Historia de la inteligencia artificial,» [En línea]. Available: [https://es.wikipedia.org/wiki/Historia\\_de\\_la\\_inteligencia\\_artificial](https://es.wikipedia.org/wiki/Historia_de_la_inteligencia_artificial).
- [8] Iberdrola, «¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?,» [En línea]. Available: <https://www.iberdrola.com/te-interesa/tecnologia/que-es-inteligencia-artificial>.
- [9] Wikipedia, «Inteligencia artificial,» [En línea]. Available: [https://es.wikipedia.org/wiki/Inteligencia\\_artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial).
- [10] XatakaHome, «Probamos la nueva Raspberry Pi 2: A fondo,» [En línea]. Available: <https://www.xatakahome.com/trucos-y-bricolaje-smart/probamos-la-nueva-raspberry-pi-2-a-fondo>.
- [11] Wikipedia, «Historia de la RaspBerry Pi,» [En línea]. Available: [https://es.wikipedia.org/wiki/Raspberry\\_Pi#Historia](https://es.wikipedia.org/wiki/Raspberry_Pi#Historia).
- [12] HistInf, «Raspberry Pi, Historia de la informática,» [En línea]. Available: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>.
- [13] Computer Hoy, «15 usos de la Raspberry Pi que no sabías que podías darle,» [En línea]. Available: <https://computerhoy.com/noticias/hardware/15-usos-raspberry->

pi-que-no-sabias-que-podias-darle-74905.

- [14] Xataka, «Las 13 mejores ideas que hemos encontrado hechas con Raspberry Pi,» [En línea]. Available: <https://www.xataka.com/accesorios/las-13-mejores-ideas-que-hemos-encontrado-hechas-con-raspberry-pi>.
- [15] Wikipedia, «Python- Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Python>.
- [16] Nobbot, «Python, un lenguaje simple para comprender la complejidad del mundo,» [En línea]. Available: <https://www.nobbot.com/general/python-lenguaje-programacion/>.
- [17] BeJob, «7 razones para programar en Python,» [En línea]. Available: <https://www.bejob.com/7-razones-para-programar-en-python/>.
- [17] Escuela de python, «Grandes proyectos hechos con Python,» [En línea]. Available: <http://www.escuelapython.com/grandes-proyectos-hechos-python/>.
- [18] Snowboy, «Snowboy Hotword Detection,» [En línea]. Available: <https://snowboy.kitt.ai/>.
- [19] pypi, «SpeechRecognition Pypi,» [En línea]. Available: <https://pypi.org/project/SpeechRecognition/>.
- [20] Wit, «Wit.ai,» [En línea]. Available: <https://wit.ai/>.
- [21] python programming language, «Text to speech|Python,» [En línea]. Available: <https://pythonprogramminglanguage.com/text-to-speech/>.
- [22] Google, «Youtube,» [En línea]. Available: <https://www.youtube.com/watch?v=dcG9bLhLYhU>.
- [23] Retina, «Inteligencia artificial Adam Cheyer: “Dentro de poco, los asistentes de voz serán más importantes que los ‘smartphones”,» [En línea]. Available: [https://retina.elpais.com/retina/2017/11/27/tendencias/1511800893\\_865293.html](https://retina.elpais.com/retina/2017/11/27/tendencias/1511800893_865293.html).
- [24] Nae, «Tendencias globales para el mercado de los asistentes virtuales,» [En línea]. Available: <https://nae.global/tendencias-globales-para-el-mercado-de-los-asistentes-virtuales/>.