Universidad de La Laguna

Universidad de La Laguna

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

Agente inteligente con control por voz para redes IoT

Intelligent agent with voice control for IoT networks

Alumno:

Josué Gutiérrez Ledesma

Tutor:

Dr. Carina S. González González

C/ Padre Herrera s/n
38207 La Laguna
Santa Cruz de Tenerife. España

T: 900 43 25 26

ull.es

AGRADECIMIENTOS

Agradecer a todas las personas que me apoyaron cada día en estos cuatro años. Aquellas que me ayudaron siempre cuando lo necesitaba y me brindaron una sonrisa y la fuerza necesaria para continuar.

Resumen

Este TFG se ha enfocado en el ámbito del desarrollo de agentes inteligentes que gestionen redes en IoT (Internet of Things) adaptándose para cada usuario y utilizando medidas biométricas para identificar a cada uno de ellos. Aplicando tecnologías como el lenguaje C++ y Python, se crea un ecosistema que, a través de un servidor, sea capaz de satisfacer todas las necesidades del usuario en cualquier situación, por ejemplo, durante la conducción, en la propia vivienda o incluso en lugares públicos, siendo esta caracterizada por la comunicación simultánea con diferentes usuarios y ofreciendo respuestas personalizadas para cada uno. Además, este ecosistema se basa en uno de los desarrollos de mayor importancia hoy en día como es la tecnología de IoT, caracterizado por ser un entorno seguro y 100 % accesible desde cualquier punto de conexión a Internet, ya sea, desde nuestro teléfono personal, como cualquier dispositivo en nuestra vivienda o vehículo, lo que nos aporta la comunicación, modificación o consulta con nuestro servidor en cualquier instante. Con la llegada de IoT y el 5G se necesita un agente inteligente que sea capaz de entender quién se comunica con el y aprender a comunicarse con el resto de los usuarios dentro de una misma red, facilitando acciones y ofreciendo un gestor sencillo de todo nuestro ecosistema. La implementación de la identificación de voz ofrece la capacidad de gestionar determinadas acciones en función de quién se comunique y filtrar determinadas acciones a cada usuario, aportando, por tanto, un control personal e inaccesible que a su vez se traduce en un aumento de la seguridad del ecosistema.

Palabras clave: agente inteligente, reconocimiento de voz, IoT.

ABSTRACT

The TFG has focused on the developing of an intelligence agent implemented for IoT (Internet of Things). This agent is capable of adapting for each user using biometric measurement to identify each one. Using technology based on C++ and Python languages, it has been developed an ecosystem that through a server, it would be able to satisfy each necessity by the user in any situations as possible, for example, driving, in their home or even in public places. Also, it would communicate with different users and offer personalized responses for each one. In addition, this system has been implemented on IoT technology, making it able to connect our ecosystem in a secure environment and 100 % accessible from each point with Internet access, either from their personal phone, their home or their car. With IoT and 5G systems, it is necessary to implement an intelligence agent able to understand who is communicating with and learn about communicating with different users within the same network. The implementation of voice recognizer offers the capacity to determinate whether is able to make an action with the privileges giving to their and providing a personal control and inaccessible for each unknown person, raising the security of this ecosystem.

Keywords: intelligent agent, voice recognizer, IoT.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1. INTRODUCCIÓN, OBJETIVOS Y ESTRUCTURA

1.1 INTRODUCCIÓN

Este TFG tiene como objetivo crear un agente inteligente para gestionar las redes privadas en las que se encuentran todos nuestros dispositivos y formar un ecosistema inteligente y seguro. Esta herramienta está diseñada para que pueda acompañarnos de tal forma que mantengamos siempre la comunicación con nuestra red, es decir, nuestra vivienda, vehículo, o la propia nube con nuestros archivos personales. Toda esta tecnología se está expandiendo cada vez más, lo que la hace más complicada. De ahí, esa necesidad de un gestor que nos aporte un plus de facilidad y conocimiento extra para realizar determinadas acciones y que sean totalmente personalizables en función de quién lo utilice. Este agente inteligente ha sido creado integralmente, desde la primera comunicación hasta lograr la identificación de la persona, aportando al agente, la capacidad de saber con quién se está comunicando. Al igual que no nos comunicamos ni tenemos los mismos permisos de actuar en algunas situaciones, es posible que se den circunstancias en las que el propio agente no permita realizar determinadas acciones a los usuarios. Esto solo se lograría conociendo al usuario con el que se esta comunicando. Si no supiera con quién se comunica, este concepto sería imposible de implementar. El agente, por tanto, tiene:

- Capacidad de entender y escuchar.
- Capacidad de identificación de usuario.
- Capacidad de responder y actuar.

La identificación aumenta esa complejidad en la comunicación persona–máquina, pero genera la capacidad del sistema de permitir comunicación grupal, es decir, es capaz de poder mantener una comunicación con diferentes personas simultáneamente generando respuestas en función de cada uno de los usuarios, lo que nos evita el tener que estar configurándolo predeterminadamente para un solo usuario.

Todo el conjunto de comunicación, es decir Internet y nuestros dispositivos, formarían el ecosistema privado de cada usuario.

1.2 MOTIVACIÓN

La tecnología está introduciéndose cada vez más a nuestra vida personal, con la llegada del teléfono móvil, Internet y las redes sociales, la vida personal se vuelca en las mismas. IoT aportará algo más, incluirá poder comunicarse entre diferentes sistemas privados, y a partir de ahí, la tecnología explotará de manera exponencial. Lo que me ha llevado a realizar este TFG es el simple hecho de poder aportar algo de simplicidad a tanta complejidad. Crear un gestor que sea capaz de entender y de responder de manera natural y humana satisfaciendo estas necesidades. En situaciones tales como en el vehículo, se limita el uso de determinados servicios con el teléfono, sin embargo, un gestor controlado por la voz habilitaría funciones importantes y no solo para controlar un móvil personal sino para controlar todo lo que tengamos conectados en IoT, lo que se aprovecharía el tiempo gestionando llamadas, respondiendo mensajes, comprobando el estado de la casa a través de IoT, y todo gestionado con la simple voz.

A su vez, desde un punto de vista más técnico, el desarrollo interacción Humano – Máquina está cobrando mayor importancia a lo largo de los años. Cada vez son más los diferentes asistentes disponibles en el mercado con mayores capacidades de comprensión y respuesta. La motivación radica en cómo aportar a estos la identificación de usuarios con fines de seguridad y dar más información al agente inteligente para que la interacción aporte una mayor relación entre el humano y la máquina y no sea solo una relación de manera general, es decir, que la máquina se comporte de igual forma para cualquier humano. Esto significa, que la interacción pasaría de ser Humano – Máquina a Humano conocido – Máquina. Estas interacciones afectarían no solo al ámbito privado sino al sector industrial, el modo en el que un operario puede comunicarse con la maquinaria dentro de la fábrica, la seguridad dentro de esta, y la gestión de cambios o de funcionamiento a distancia de la empresa a través de las redes.

Uno de los dispositivos más usados dentro de un ecosistema privado es el móvil personal, sin embargo, en determinadas circunstancias es inaccesible, por ejemplo, en la conducción. Este problema es muy grave hoy en día, lamentablemente se sigue teniendo accidentes muy graves por el uso de este dispositivo, ya sea por necesidad de mensajería instantánea o incluso por atender al trabajo. El poder estar comunicado con todo este ecosistema a través de la voz, afecta también a estas situaciones. Si existiera la posibilidad de tener un 100 % de control del dispositivo móvil, y no solo este, sino todos los dispositivos dentro de un ecosistema, se podría gestionar solamente con la voz, lo que eliminaría la necesidad de usar el móvil en el coche y poder prestar mucha más atención a la conducción.

1.3 OBJETIVOS

Los objetivos del TFG son los siguientes:

- Diseñar y desarrollar un agente inteligente capaz de reconocer e identificar al usuario que se está comunicando y dar respuestas coherentes en función de este.

- Implementar el control del sistema de voz del agente en una misma red formando una red privada IoT.

- Diseñar diferentes modos de interacción con el agente incluyendo el modo multiusuario de forma simultánea.

- Implementar el agente en un servidor de una red privada, de forma accesible para todos los usuarios y dispositivos de la red.

El agente inteligente no se encuentra instalado dentro de cada uno de los dispositivos, sino se crea un servidor principal dentro de la red, disponible para todos estos dispositivos de tal forma que solo se tienen que comunicar entre ellos. Este procedimiento genera una serie de ventajas que son:

- Mayor eficiencia en cuanto a recursos consumidos por los dispositivos: realmente no consumen recursos solo envían y reciben datos.

- Rapidez en la respuesta: todos los procesos de procesamiento de texto e identificación se realizan en un servidor, por tanto, la potencia será igual a la que pueda producir el servidor.

- Sistema personalizable para cada usuario.

1.4 METODOLOGÍA

La estructura del TFG tendrá diferentes etapas, tanto de análisis, como de propuesta y resultados obtenidos.

En primer lugar, se realizará un estudio de la situación actual, tanto de las redes IoT como los asistentes inteligentes, y cómo se está desarrollando esta tecnología. También, se valorará la razón de este TFG dentro de esta situación y qué es lo que puede aportar.

En segundo lugar, se debe realizar un análisis de los problemas que han llevado a realizar este TFG, de forma que se expongan los requisitos que se deben cumplir para alcanzar los objetivos descritos en el punto anterior.

En tercer lugar, pasaremos a explicar en detalle la solución propuesta y cómo afecta a diferentes factores como la seguridad o la eficiencia esta solución. También, se tendrá en cuenta aspectos legales como la protección de datos y no solo aspectos técnicos, debido al uso de bases de datos que almacenan y utilizan datos personales de cada usuario.

En cuarto lugar, se explicará el diseño final de la solución y la implementación de este a niveles prácticos. Además, se incluirá un estudio de los resultados obtenidos y se comprobará que se hayan cumplido los objetivos propuestos al inicio de este TFG.

1.5 ESTRUCTURA DEL TFG

El TFG se estructurará en diferentes capítulos:

- Capítulo 1. Introducción al TFG: tratará de la introducción, la motivación que ha supuesto el TFG, los objetivos ha alcanzar, la metodología a seguir, y la estructura del TFG.
- Capítulo 2. Estado del arte: tratará de la situación actual de la tecnología y cuál es la razón del TFG.
- Capítulo 3. Análisis del problema: tratará de un análisis de requisitos para cumplir los objetivos, análisis de las diferentes situaciones a implementar, la solución que se ha propuesto finalmente, un análisis de los problemas de seguridad que pueden afectar a este sistema, las ventajas que aportan este sistema a nivel de implementación, los problemas con las bases de datos a tener en cuenta y las colaboraciones utilizadas.

- Capítulo 4. Diseño de la solución: tratará del análisis de las herramientas utilizadas, el esquema de implementación y la propia implementación que redactará con más detalle cómo se ha logrado cada aspecto de la solución propuesta.
- Capítulo 5. Conclusiones: tratará de describir los resultados obtenidos y valorarlos con los objetivos previstos.

CAPÍTULO 2. ESTADO DEL ARTE

2.1 ESTADO ACTUAL DE LA TECNOLOGÍA

Este TFG trata diversos temas de la tecnología actual como es IoT y los agentes inteligentes además de estar relacionados con el 5G y la industria 4.0. Todos estos términos son los más actuales y están en pleno desarrollo, pronosticando su llegada dentro de pocos años a las grandes empresas, a las ciudades e incluso a nuestra propia vivienda. La tecnología incrementa cada año aportando mayores recursos, con procesadores y comunicaciones más avanzados siendo cada vez más eficientes, lo que facilita la implementación de sistemas más inteligentes, autónomos y comunicados entre sí. Por ello, uno de los pasos más importantes será adaptar toda la vida cotidiana a esta nueva tecnología, de ahí a que se considere una nueva revolución industrial llamada Industria 4.0, y que los agentes inteligentes estén más presente y aparezca cada vez más la palabra "autónomo".

Internet de las cosas, o más conocido como IoT, es una infraestructura global de información entre dispositivos tanto físicos como virtuales permitiendo la comunicación y envío de información entre ellos. Es un concepto que nació en 1999 por Kevin Ashton. La idea principal es poder identificar todos los dispositivos posibles a través de Internet y poder conocer su estado y sus movimientos en todo momento. Se estima que se deberá identificar entre 50 a 100 000 millones de objetos, en los que en 2020 habrá aproximadamente 26 mil millones de dispositivos. Como se comentó anteriormente, la tecnología avanza y gracias a la mejora de los diferentes protocolos de Internet, como el protocolo IPv6, es posible hablar de identificación de objetos a tales cantidades, lo cual sería inviable con los protocolos anteriores tal como el IPv4.[24]

Esta idea tiene diferentes ámbitos donde ponerlos en práctica, pero comparten la misma idea de IoT, un mundo conectado. Podemos hablar tanto de aspectos privados, como públicos o empresariales. IoT es una tecnología que pretende unir todo el servicio público interconectado formando así las llamadas *Smart City,* concepto que poco a poco se está creando en diferentes partes del mundo, así como implementar ciudades que puedan comunicar el estado del servicio público de transporte en tiempo real, cámaras de seguridad interconectadas para poder rastrear o detectar y así obtener más

información como por ejemplo la búsqueda de personas en una gran ciudad. En ámbitos privados también tiene mucha importancia con la inclusión de los archivos e información personal en la nube. Poder tener un servicio en IoT implica controlar todos los dispositivos de una vivienda (domótica), además de mejorar la seguridad de las viviendas con sensores biométricos tanto por voz como las huellas dactilares.

En el ámbito industrial, IoT es la piedra fundamental de la industria 4.0. Las empresas cada vez se hacen más autónomas para poder ser mucho más productivas, la industria 4.0 busca interconectar toda la maquinaria entre sí y así ser más inteligente y productivo que sistemático y disminuir los problemas de gestión, retrasos o fallos. Sin embargo, la industria 4.0 esta construida para poder adaptarse a cualquier necesidad de la propia empresa, y ser modificada o poder consultar cualquier información en tiempo real. Esto es una de las mayores ventajas ya que produce mayor flexibilidad y adaptación al entorno y a la situación de mercado.[23]

Para poder aportar ese gestor que tome decisiones o interprete los datos y la información recibida, se utiliza los agentes inteligentes. Los agentes inteligentes se definen como la combinación de algoritmos con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano. Dentro de la industria, la vida privada o cualquier servicio, puede aplicarse los agentes inteligentes, tanto para la automatización de la maquinaria, como el control de los electrodomésticos de una vivienda, como la gestión y comunicación con el usuario. Algunas de los más destacados son los asistentes virtuales como Siri de Apple, Alexa de Amazon o Cortana de Microsoft.

También algunas técnicas como el reconocimiento por voz o reconocimiento facial forman parte de estos agentes inteligentes aportando mayor seguridad al sistema. En el entorno de Internet of Things, una de las funcionalidades más importantes es reconocer y entender la información que reciben para desarrollar una respuesta, es decir, simular un razonamiento o desarrollar una acción a partir de la información que transmite el usuario. Toda esta tecnología, la podemos ver aplicada en nuestros teléfonos móviles, tabletas, e incluso en unos años en todos nuestros dispositivos más cercanos.

Para finalizar este punto, introducimos el 5G como la responsable de que todas estas comunicaciones funcionen de manera correcta. Si conectamos todos los dispositivos a la red, la velocidad de transmisión de datos debe ser la adecuada sino la experiencia podría ser negativa y causar muchos problemas.

El 5G es una velocidad de transmisión de datos capaz de superar en 1000 veces la red 4G. Además, es capaz de soportar un mayor número de usuarios, con menor latencia,

mayor fiabilidad y mejor conectividad, lo necesario para poder conectar los casi 26 mil millones de dispositivos, incluidos edificios, vehículos, entre otros. Esto significa que la transmisión de datos es casi instantánea, lo que da lugar a la posibilidad de realizar operaciones complejas a distancia, como obtener la información en tiempo real de las ciudades para controlar el tráfico, el tiempo y hasta la delincuencia.[17]

2.2 CRÍTICA AL ESTADO ACTUAL

Como se ha mencionado anteriormente, el ser humano se encuentra en un momento importante a nivel tecnológico mundialmente. En unos pocos años, se implantarán estos 4 términos mencionados en el punto anterior, IoT, agentes inteligentes más avanzados, 5G e Industria 4.0, justificando la necesidad de estar preparados. Toda la tecnología sufrirá un pequeño cambio, se pasará de estar interconectados entre los usuarios a un nuevo nivel de interconexión entre los usuarios y los dispositivos que nos rodean. Cada año la tecnología aumenta en un porcentaje de eficiencia y productividad, es decir, menor consumo para mayor producción lo cual beneficia y es interesante para IoT ya que uno de los principales problemas que se encuentra es el control de la cantidad de información que se enviará por las redes.

A la par que avanza la tecnología, los dispositivos van siendo capaces de incorporar nuevas opciones de control a través de Internet, mayor capacidad de comunicación por vía 5G, etc. La idea principal es permitir desde nuestro propio Smartphone comunicarse con todos los dispositivos conectados a nuestra red privada, lo cual será sencillo, pero a la vez complejo a medida que se aumenta el número de dispositivos y la cantidad de acciones que se puedan realizar con cada uno de ellos. Lo cual se crea un hueco que existe hoy en día como el asistente personal, sin embargo, se deberá ampliar con mayores técnicas como, por ejemplo, comprensión del lenguaje natural, identificación del usuario, capacidad de tomar decisiones, y comunicación con el usuario de manera natural y clara, entre otras. Además, deberá ser compatible con redes IoT y poder gestionar las comunicaciones a través del 5G para poder realizar la acción dicha por el usuario.

2.3 PROPUESTA

Para formalizar esta idea, se debe diseñar un sistema que sea capaz de gestionar y controlar una red privada a través de un agente inteligente. Esta red privada se entenderá la que pueda aplicarse tanto en viviendas privadas como establecimientos públicos o en empresas privadas. La red estará formada por todos los dispositivos que se deseen conectar, con un identificador propio, y los usuarios registrados con la capacidad para solicitar información, realizar cambios desde cualquier punto de red que conecte con la red IoT.

No obstante, como se ha comentado en el punto anterior, esta gran cantidad de información deberá ser gestionada por un agente inteligente que cumpla con los siguientes puntos:

- Deberá ser capaz de identificar al usuario con el que se comunica para filtrar las acciones permitidas a este, o dar información personalizada para cada usuario.
- Deberá generar una respuesta a la solicitud de los diferentes usuarios.
- Deberá estar 100 % disponible a cualquier usuario, de tal manera, que pueda aplicarse simultáneamente.

Para cumplir con los objetivos, se partirá de dicha propuesta y se analizará los diferentes requisitos a cumplir en cada aspecto de esta y generar así diferentes soluciones para lograr un análisis global y conformar una solución final.

CAPÍTULO 3. ANÁLISIS DEL PROBLEMA

3.1 ANÁLISIS DE REQUISITOS

Para poder conseguir los objetivos propuestos en este TFG, se deberá cumplir una serie de requisitos básicos.

Para poder convertir el ecosistema en una red IoT, primero se debe poder conectar todos los dispositivos en un punto común, en este caso, un servidor privado. Este servidor se encargará de gestionar todas las comunicaciones entre dispositivos, acceso externo de dispositivos fuera de la red privada, almacenar los dispositivos que pueden conectarse a la red y controlar todos los dispositivos conectados a ella. Para ello, utiliza un gestor en forma de agente inteligente que pueda interactuar con el usuario y con los propios dispositivos. Este agente inteligente estará incluido en el servidor, de tal manera, que cualquier dispositivo puede conectarse a él mismo de manera simultánea. Esta ventaja nos permite aumentar la flexibilidad y personalización con cada usuario, aumentar la velocidad de respuesta y de procesamiento de acciones en tiempo real, transmitir información a diferentes dispositivos y controlarlos en cualquier punto con acceso a Internet.

El agente inteligente debe constar con un sistema de comprensión de lenguaje natural y saber utilizar el lenguaje como medio de transmisión de datos entre el usuario y el agente. El sistema también debe ser capaz de identificar al usuario que está solicitando un determinado servicio, ya que una de las ventajas es que el agente es flexible con cualquier usuario, por lo que las acciones o la respuesta de dicha consulta será diferente para cada uno de ellos. Esto nos ofrece una de las mayores ventajas, posiblemente la más importante y en la que más se enfoca el TFG como un objetivo principal y es que el agente es capaz de identificar, aportando a cada usuario permisos y una serie de condiciones para realizar determinadas acciones, es decir, no es lo mismo que se solicite conocer el estado de la temperatura de la vivienda, que solicitar la apertura de una de las puertas de la vivienda, o modificar el estado on/off de cualquier toma de corriente en la vivienda y, por tanto, se puede atribuir este nivel de configuración a determinados usuarios.

El agente debe asegurar la total seguridad de los datos de cada usuario, y el no acceso de otro usuario diferente a la red privada utilizando las medidas de verificaciones

adecuadas. Estas deben ser lo más biométricas posibles, reduciendo la posibilidad de que alguien ajeno pueda acceder. El agente, en caso de que no pueda asegurar 100 % la verificación del usuario debe exigir el completo acierto de diferentes pruebas biométricas, como la voz o la huella dactilar, de tal forma, que pueda asegurar el acceso al usuario. En caso negativo, el agente no permitiría la acción y, por tanto, evitaría así la corrupción de los datos privados del sistema y el completo acceso a la red privada.

3.2 ANÁLISIS DE LAS SOLUCIONES

Para cumplir con los requisitos previamente dichos, debemos dividir el proyecto en diferentes puntos:

1. Comunicación con el servidor.
2. Agente inteligente.
3. Conectividad de los dispositivos.

3.2.1 COMUNICACIÓN CON EL SERVIDOR

En este apartado se estudiarán los diferentes métodos de envíos de datos. En la actualidad el medio más rápido de envío es a través de Internet. Con la llegada del 5G las velocidades serán superiores a las actuales por lo que serán más que suficiente. Para el envío de datos por internet, se estudia los diferentes protocolos a implementar en el servidor.

Según IBM, “un protocolo es un conjunto de normas para formatos de mensajes y procedimientos que permite a las máquinas intercambiar información”. Los sistemas dentro de la red deben seguir un mismo protocolo para el envío de los datos y poder realizar una comunicación entendible, completa y eficiente.

En primer lugar, tenemos el protocolo TCP (*Protocolo de Control de Transmisión*). Junto al protocolo IP, forma uno de los principales protocolos de la capa de transporte TCP/IP.[18] TCP es un protocolo orientado a la conexión, permite que los dos sistemas en conexión estén comunicados y controlen el estado de la transmisión. Por tanto, deben establecer conexión antes de enviar los datos.[5]

Las principales características de este protocolo es que puede comunicarse de forma segura, ya que verifica que el destinatario reciba la información de manera correcta. Además, implica que, en la capa de Internet, es decir, los routers, tienen que enviar datos

en forma de datagramas sin preocuparse de controlar los datos ya que esta función es responsabilidad del protocolo TCP.

La comunicación la realiza uno de los sistemas llamado cliente hasta otro sistema llamado servidor, funcionando en un entorno Cliente – Servidor en el que la información se envía de manera bidireccional.

En segundo lugar, tenemos el protocolo UDP (*User Datagram Protocol*). En este caso, es un protocolo no orientado a la conexión. A diferencia del TCP, en este protocolo no es necesario establecer conexión con el destinatario para enviar datos a través de la red. No tiene tampoco información de control de mensajes ni confirmación de entrega o recepción. Se suele utilizar para el envío de datos con mayor cantidad de información como, por ejemplo, archivos de audio o video que sea necesario enviarlos en tiempo real, ya que al aplicar el protocolo TCP se producirían retardos. [6]

Se caracteriza por ser un sistema no fiable, es decir, no tiene la responsabilidad de que el mensaje se haya entregado, o estén completos o se hayan duplicados, por lo que la calidad de emisión y recepción de datos es baja.

En este caso, la red IoT deberá ser capaz de emitir y recibir datos de todos los dispositivos conectados simultáneamente, por lo que a priori parece recomendable usar el protocolo UDP para obtener una mayor calidad de respuesta. Sin embargo, la llegada del 5G solucionaría este problema, proporcionándonos la capacidad de enviar mayor cantidad de información en un menor tiempo, y aplicar protocolos más eficientes y seguros como es el TCP.

Datos/Protocolos	TCP	UDP
Tipo	Orientado a la conexión	No orientado a la conexión
Necesita conexión con destinatario	Si	No
Control de mensajes	Si	No
Confirmación de entrega	Si	No
Calidad del sistema	Fiable	No fiable
Archivos	Textos	Audio, video

Tabla 1. Comparativa protocolos

3.2.2 AGENTES INTELIGENTES: ASISTENTES VIRTUALES

En la actualidad, existen diferentes asistentes virtuales o agentes inteligentes como *Siri* de Apple, *Cortana* de Windows o *Alexa* de Amazon. En este caso, se necesita un agente inteligente que, a parte de aportar la capacidad de entender y realizar algunas acciones, pueda comprender lenguaje natural y tener el conocimiento de gestionar los diferentes equipos conectados en IoT.

Existen aplicaciones diseñadas para acompañar a estos agentes inteligentes como es *Casa* de Apple.[7][8] Esta aplicación necesita unos accesorios llamados *Homekit* que permiten el envío de información a la aplicación permitiendo el control de los dispositivos. Algunos accesorios pueden ser luces, altavoces, enchufes, termostatos, ventanas, ventiladores, etc.

Sin embargo, esta solución esta muy limitada a algunas partes de una vivienda en el que Siri comprende el comando que le solicita el usuario y guarda sus preferencias. El problema de esto es que la opción multiusuario, es decir, la simultaneidad del uso de dos usuarios diferentes a través del mismo dispositivo, por ejemplo, un altavoz inteligente, hace que no pueda distinguir las preferencias de uno y de otro ya que se configura para una cuenta determinada.

Además, los dispositivos están limitados a donde coloquemos los *Homekit*, lo que significa que podamos controlar solamente accesorios dentro de la vivienda. Otro punto negativo, es que la única forma de conectarse a un accesorio desde fuera de casa es manteniendo conectado un dispositivo como un IPad o un Apple TV.

La implementación de un servidor dará la ventaja de tener la capacidad de controlar toda la red IoT dentro y fuera de la casa, lo que solucionaría la limitación de accesorios y la necesidad de mantener un dispositivo conectado para poder controlar diferentes aspectos dentro de la vivienda.

Otros agentes inteligentes se suelen encontrar instalados dentro de cada dispositivo, lo cual, sería como diseñar un mismo agente en cada dispositivo, haciendo independiente la acción de cada uno. Esto no podría ser compatible con el objetivo de este TFG, debido a que sería una pérdida de recursos estar diseñando el mismo asistente para cada uno de los dispositivos, y tampoco sería compatible debido a la falta de conexión entre todos ellos, lo que no podría ser gestionado de manera eficiente. La mejor solución es que cada dispositivo pueda conectarse al agente inteligente mediante el servidor, es decir, un único agente inteligente que controle todos los dispositivos y pueda ser consultado en

cada uno de ellos, de manera que sea capaz de gestionar todas las acciones requeridas tanto por los usuarios como por los dispositivos de una manera más rápida, eficiente y en tiempo real.

Como se ha comentado, todo será posible a la gran velocidad que nos proporcionarán las redes 5G ya que será necesario para enviar y recibir datos de tal cantidad.

La comunicación usuario – agente debe ser implementada a través de lenguaje natural, es decir, el usuario no debe instruirse en aprender comandos específicos para realizar determinadas acciones, sino el propio sistema debe tener la capacidad para entender una misma acción expresada en diferentes formas.

Un ejemplo claro de esto sería el uso del agente inteligente en el vehículo. Hoy en día, existen compañías que ya implementan sistemas inteligentes con mayor capacidad como es Apple o Google, sin embargo, estas opciones suelen ser implementadas en vehículos de gama alta o con mayor dificultad de accesibilidad o muy exclusivos. Si observamos las soluciones más normalizadas en el resto de los vehículos, existen diferentes alternativas, pero casi siempre con una característica en común, la comunicación se realiza a través de comandos, tales como "llamar a …", o "ver mensajes".

Este tipo de lenguaje no sería adecuado y se debe evitarlo, por lo que se tratará de crearlo fijándose más al sentido de *Siri o Alexa*, de tal forma, que las acciones anteriores puedan ser introducidas como "necesito que llames a…", "podrías llamar a …", "en cuanto puedas llama a …", "¿tengo mensajes disponibles?", entre otros. Como se observa en este ejemplo, la diferencia es notable, y la segunda opción es mucho más natural, no se debe suponer que el usuario tendrá que aprender ciertas palabras clave para que el sistema entienda al usuario, sino todo lo contrario, como se puede diseñar el sistema de tal forma que el usuario pueda ser alternativo y se obtenga la misma respuesta.

Tras analizar estos sistemas, se ha de diseñar un sistema muy parecido a lo que haría *DialogFlow* de Google.

DialogFlow se basa en un sistema de comprensión natural del lenguaje basada en la API AI de Google. Se basa en un procesamiento de *machine learning* en el que es capaz de entender la intención del usuario a través de palabras que introduzca el usuario. Esto permite que la conversación entre usuario y máquina sea lo más natural posible y se reduzcan los errores de comprensión.[9]

La solución que se busca debe tener la misma orientación, se ha de crear un sistema que pueda comprender de manera natural la intención del usuario independientemente de la manera en la que lo exprese. Por tanto, el agente inteligente trata de buscar palabras claves para obtener una orientación de la idea del usuario, pero, ¿por qué debe el agente a través de palabras clave intentar entender la intención del usuario?

Lo que tratará el sistema inteligente es comprender exactamente la intención del usuario, enseñándole previamente qué es lo que significa dichas palabras e intenciones, es decir, ampliaremos el sistema enseñando el uso de verbos, adjetivos, sustantivos, y poder relacionarlos entre sí.

Un ejemplo de esto sería el siguiente:

USUARIO: Necesito información para solicitar un puesto de trabajo.

Si solo entendiera palabras claves sería lo siguiente:

Información, puesto de trabajo → En una página de un portal de trabajos, se podría llegar a la conclusión de que el usuario pide información de puestos de trabajo, es decir, toda la información, puestos disponibles, solicitar puestos, puestos no disponibles, puestos más frecuentes, etc.

Si entendiera el significado de la intención del usuario:

Necesito: necesidad del usuario. No relevante.

Información: mostrar información, instrucciones, etc.

Para: preposición no relevante.

Solicitar: significa pedir, necesito un documento o cualquier información. El agente solicitaría la necesidad de un documento al usuario a no ser que se detecte uno.

Puesto de trabajo: documento o referencia del verbo solicitar

La acción final ***es mostrar en una página información de como solicitar un puesto de trabajo.***

Como se observa, la 2ª opción resulta más eficiente y mucho más realista que la 1ª opción mostrando toda la información relacionada con puestos de trabajo.

Otras de las características que deben definir al agente es la capacidad de identificación. Para ello, será necesario que el agente analice diferentes parámetros del audio de la persona para obtener las características de cada usuario. Para ello, se puede optar por dos métodos:

1. Análisis de frecuencia.
2. Análisis de frecuencia aplicando inteligencia artificial.

MÉTODO 1: ANÁLISIS DE FRECUENCIA

La identificación del usuario es un proceso en el que el sistema verifica que el usuario es conocido y puede acceder y comunicarse dentro de la red IoT. Las principales ventajas que aporta este sistema es tener la capacidad de personalizar determinadas acciones en función de quién se comunique y proteger la red para que solamente pueda acceder los usuarios registrados en la red IoT.

Para realizar esta identificación se partirá de un análisis de las ondas de sonido que se produce al comunicarnos. En concreto, se usará un análisis de Fourier para hallar la FFT (*Fast Fourier Transform)* del audio del usuario. La FFT nos mostrará las frecuencias que tienen un mayor peso o importancia en la amplitud de la señal y, por tanto, que caracterice a los diferentes usuarios.

En la siguiente figura se muestra un análisis de Fourier de un audio:

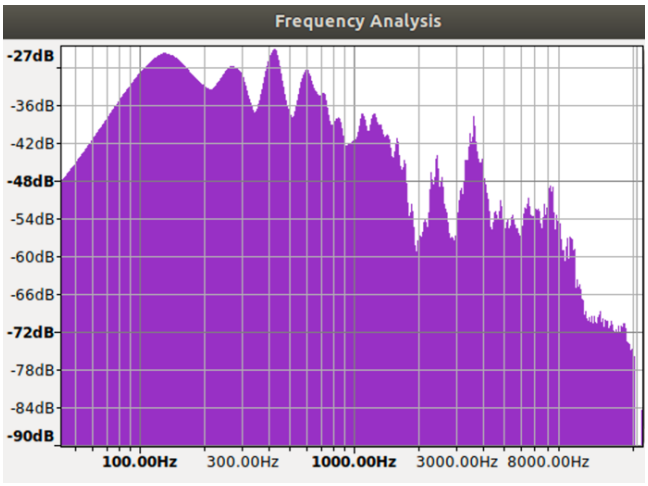


Figura 1. Análisis de frecuencia

Como podemos observar, los picos más altos son los que contribuyen a una mayor amplitud y nos ofrece más información. El agente inteligente se encargará de examinar qué frecuencias tienen más información del sonido y cómo caracterizarlas para poder diferenciar entre los usuarios.

A priori, podríamos indicar que las frecuencias con mayor amplitud serían suficiente, sin embargo, esto no es así. Esto solo sería el resultado obtenido en un solo audio de un usuario que puede depender tanto de su voz como otras características que podríamos tomar como pequeñas perturbaciones tales como la distancia al micrófono, el sonido externo, la pronunciación, la vocalización, la rapidez de la frase, entre otras. Por lo que no podemos asegurar que un solo audio nos aporte la información necesaria como para definir de manera universal las características de este usuario, ya que los audios pueden diferir siendo la misma frase y el mismo usuario.

Es por ello, que se tomarán 4 muestras diferentes del mismo usuario para comparar con el audio que se quiere identificar y así asegurar con un poco más de seguridad los resultados obtenidos.

MÉTODO 2: ANÁLISIS DE FRECUENCIAS CON INTELIGENCIA ARTIFICIAL

El concepto de inteligencia artificial se refiere a la capacidad del propio sistema de crear un algoritmo capaz de encontrar una relación entre las diferentes entradas y las salidas de forma que pueda predecir una solución para otra entrada diferente.

Para aplicar este método en el sistema de identificación, se utilizará las llamadas *redes neuronales artificiales.*

Una *red neuronal artificial* es un modelo simple del funcionamiento del sistema nervioso, formado por neuronas organizadas en capas.[10]

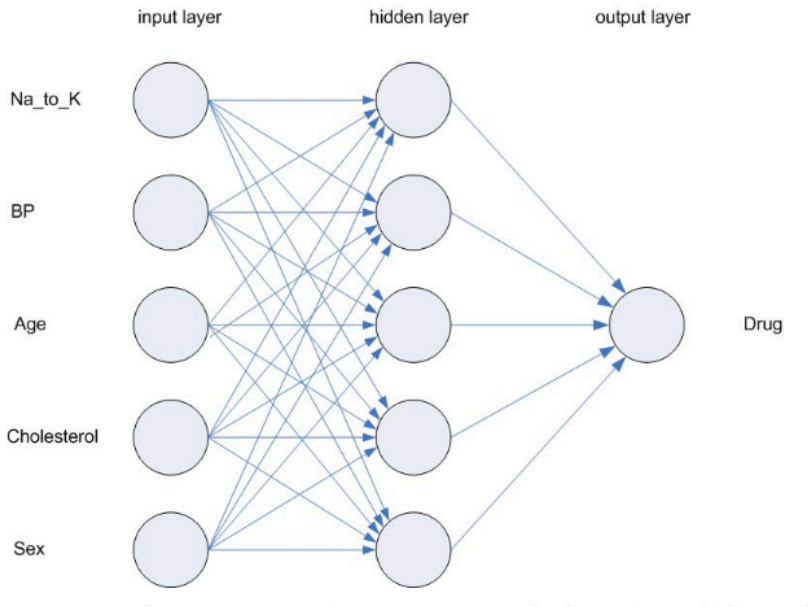


Figura 2. Esquema red neuronal

Principalmente, el objetivo de la red neuronal es tratar de buscar la codificación de una serie de parámetros de entrada para obtener la salida deseada o correcta.

Las unidades de procesamiento se organizan en tres capas:

1. Capa de entrada
2. Capas ocultas
3. Capa de salida

Estas capas están conectadas entre sí a través de conexiones llamadas *ponderaciones*. Estas ponderaciones se aplican a los parámetros de la capa de entrada que a su vez generan diferentes resultados que creando una serie de indefinidas conexiones en las capas ocultas generan una salida. La red crea una serie de predicciones en función de los parámetros de entrada establecidos para luego compararlo con el resultado real. Este proceso se realiza de forma iterativa, de modo que la red automáticamente reconstruya sus ponderaciones aplicadas a los parámetros de entrada y obteniendo resultados con un menor error al resultado real.

La red establece, al comienzo, ponderaciones aleatorias obteniendo resultados muy dispares con los reales. Aplicando un entrenamiento con diferentes iteraciones va obteniendo mejores ponderaciones. Para ello toma los resultados correctos y realiza las iteraciones hacia atrás cambiando las ponderaciones. La red se hace cada vez más precisa en los resultados para los que ha entrenado, si es cierto, que puede producirse errores o que no funcione en otras situaciones en la que los parámetros de entrada difieran con los de entrenamiento para un mismo problema.

En la identificación se usará para buscar patrones del mismo usuario. Se solicitarán 4 audios y se aplicará a través de una red neuronal que se encargue de buscar los patrones de dichos audios. La idea principal sería encontrar las ponderaciones que, aplicadas a la entrada, la salida sea semejante o se obtenga unos valores característicos de un usuario en concreto. Al igual que en el método 1, se aplicará este método con 4 ponderaciones diferentes que determinen el patrón de los audios y así ofrecer unos resultados con mayor probabilidad de acierto.

En conclusión, si realizamos un análisis del mercado, observamos que la mayoría de los asistentes virtuales cada vez son más potentes y semejantes, lo que significa que se podría trabajar sobre cualquiera de ellos. Para ello, analizaremos el funcionamiento de *Siri* y *Google Assistant* para comprobar si ambos asistentes cumplen con lo siguiente:

- Sean capaces de procesar la misma información con diferentes frases y dar el mismo resultado.
- Sean capaces de identificar al usuario que esta hablando.

Los resultados son los siguientes:

Pruebas / Asistentes	Siri	Google Assistant
Diferentes frases	No	Si
Identificación	Si	No

Tabla 2. Análisis asistentes virtuales

Podemos comprobar la prueba de diferentes frases en los dos asistentes:

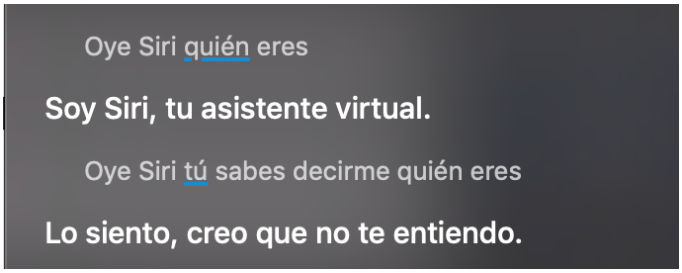


Figura 3. Respuestas de Siri

Figura 4. Respuestas de Google Assistant

Por tanto, *Siri* nos permitiría consultar de manera segura, pero con menor procesamiento que *Google Assistant*. Por lo que se tratará es de diseñar un sistema tanto de procesamiento como de identificación que puedan ser aplicados a cada uno de los asistentes virtuales como un módulo adicional.

3.2.3 CONECTIVIDAD DE LOS DISPOSITIVOS

Cada dispositivo conectado al servidor de la red IoT debe tener capacidad de conexión a Internet. Hoy en día, cada vez son más los dispositivos con capacidad de conexión, pero otros todavía no cuentan con esta tecnología. Para ello, una de las soluciones será crear una placa gestionado por un microcontrolador tal como *arduino* o *raspberry* que sea capaz de gestionar el dispositivo enviando y recibiendo datos desde el servidor y realizando la función de actuador en las distintas órdenes dichas por el propio agente inteligente.

En primer lugar, tenemos como opción Arduino que se trata de una plataforma de código abierto para la creación de proyectos. Se trata de un microcontrolador que nos permite crear códigos que puedan usar diferentes entradas tantas analógicas como digitales y controlar las diferentes salidas como luces, motores o actuadores.[11]

Este dispositivo es programable a través de su entorno de desarrollo y por USB lo que evita usar un programador. Su código se basa en una versión simplificada de C++. En función de las necesidades del usuario, existen alternativas al elegir la placa.

Figura 5. Marca Arduino

En este caso, es interesante que la placa tenga acceso a Internet, por lo que la opción más recomendable sería el Arduino MKR1010 que contiene la funcionalidad adicional Wi-Fi. Aunque, también se podría optar por una placa normal de Arduino y ampliarla con el módulo ESP8266 Wi-Fi.

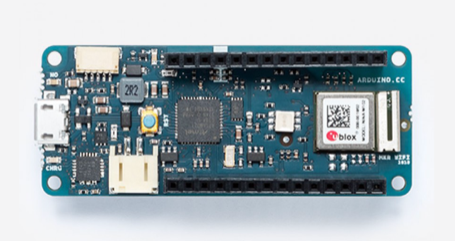


Figura 6. Arduino MKR 1010

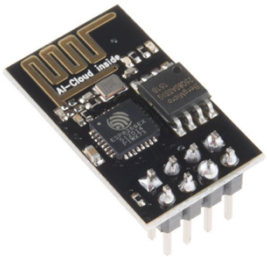


Figura 7. ESP8266 Módulo Wi-Fi

La segunda opción sería usar una Raspberry PI. Es una placa de bajo coste desarrollado por la Universidad de Cambridge en 2011 con el objetivo de la enseñanza de la informática.[12] A nivel de hardware, la Raspberry PI consta de:

- Un Chipset Broadcom conteniendo el procesador central (CPU).
- Un procesador gráfico
- Un módulo de 512 MB de memoria RAM
- Conector RJ45.
- 2 buses USB 2.0
- Salida analógica de audio estéreo por Jack.
- Salida digital + audio HDMI.
- Salida analógica de video RCA
- Pines de entrada
- Conector micro USB
- Lector de tarjetas SD.

Los modelos más actualizados como la Raspberry Pi 3 Model B+, consta de estos mismos elementos mucho más potentes y además con soporte de redes Wi-Fi sin necesidad de usar un adaptador como se hubiese necesitado en modelos anteriores.[13]

La Raspberry puede utilizar diferentes sistemas operativos siendo el más usado el Raspbian, una versión adaptada de Debian. Principalmente, se programa en lenguaje Python.

Las principales diferencias entre Arduino y Raspberry es su finalidad. Mientras Arduino se crea como un producto con mayor utilidad y versatilidad por las diferentes aplicaciones, Raspberry tiende más a ser un pequeño ordenador con potencia de cálculo. Arduino está más orientado a realizar proyectos más pequeños, con menor potencia y mayor rapidez de procesamiento, mientras que Raspberry tienen mayor potencia y se nos ofrece como una buena opción para proyectos con un nivel superior de cálculo. También Raspberry Pi viene incluido la capacidad de Wi-Fi y Ethernet en la misma placa, mini Jack, capacidad para conectar periféricos como un teclado o ratón, entre otros.[14]

En este proyecto, la función que debe realizar es la de enviar y recibir datos con el servidor y ejecutar diferentes acciones, por lo que, simplemente serviría utilizar un módulo Arduino que conste de conectividad inalámbrica Wi-Fi.

La opción de la Raspberry Pi se podría aplicar en el servidor, es decir, como controlador y gestor de todas las comunicaciones con los diferentes dispositivos por su gran potencia, aunque con un número elevado de dispositivos no se podría determinar con exactitud si sería suficiente potencia para manejar tantos dispositivos de manera simultánea.

3.3 SOLUCIÓN DEFINITIVA

Una vez hemos analizado los requisitos que debe ofrecer el TFG para lograr los objetivos y analizado las diferentes soluciones, se ha de plantear cuál va a ser la solución definitiva.

Como se ha hecho en el apartado anterior, se divide la solución en diferentes puntos:

- COMUNICACIONES DEL SERVIDOR Y DISPOSITIVOS
- AGENTE INTELIGENTE
- PLACA DE COMUNICACIÓN DE LOS DISPOSITIVOS
- DESARROLLO DE INTERFACES

3.3.1 COMUNICACIONES DEL SERVIDOR Y DISPOSITIVOS

En función de la tabla 1, se ha llegado a la conclusión de utilizar el protocolo TCP/IP debido a que es un protocolo más seguro para enviar y recibir datos.

Una de las características de las redes IoT debe ser su alta seguridad, por tanto, no podemos recortar aplicando protocolos más rápidos, pero menos seguros como el UDP. Además, las redes aumentarán con la llegada del 5G por lo que la transmisión de datos será más rápida y eficiente.

El servidor permanecerá fijo dentro de cada vivienda o local con una conexión a Internet creando un espacio personal para esta red privada. Dentro de ella se pueden comunicar los distintos dispositivos y usuarios entre sí, quedando registrado la actividad de estos y el historial de comandos efectuados. Se encargará de gestionar todas las comunicaciones y establecerlas de manera segura y simultáneamente lo que aporta una mayor rapidez y flexibilidad.

El servidor constará de un sistema de archivos que almacena toda la información de usuarios y dispositivos conectados a la red IoT. Este sistema de archivos es el único que puede permitir que un usuario o dispositivo pueda permanecer en la red y usarla para comunicarse dentro de esta. Cada dispositivo está caracterizado por un número identificativo único que le servirá al servidor para reconocer los diferentes dispositivos. Esto aportará una mayor seguridad al no permitir la entrada a un dispositivo que no se encuentre en los archivos a la red.

Lo mismo ocurre para los usuarios que se conecten a la red. Cada usuario puede comunicarse dentro de esta con cualquier dispositivo que se encuentre en el sistema de archivos ya sea un *Smartphone, Smart TV,* etc. Además, se incluirá, a través de la agente inteligente, una identificación del usuario de forma que existan 2 filtros para permitir la conexión:

1. Dispositivo en el sistema de archivos.
2. Identificación positiva de usuario.

Si un usuario ajeno a la red intenta comunicarse a través de un dispositivo, la red bloqueará la acción y no permitirá que acceda. Además, en el momento que la red crea oportuno, puede utilizar cualquier método adicional de verificación biométrica como la huella digital para verificar el usuario.

Todos los dispositivos están en constante comunicación con el servidor a través de sus módulos conectados a Internet intercambiando información y realizando acciones. Ninguno dispositivo toma decisiones, solo envían y ejecutan. El encargado de indicarle la acción es el propio servidor.

El servidor se comunicará con los dispositivos o usuarios cuando sea necesario. Esta comunicación puede ser tanto escrita como hablada. Por lo que se mandarán archivos de audio con la respuesta del propio servidor, lo que no será un problema para la red por la alta capacidad de transmisión.

Normalmente, la comunicación usuario – servidor debe ser en formato audio. A la vez que el usuario se puede comunicar con un lenguaje natural con el sistema, este debe responder del mismo modo para semejarlo a una comunicación natural entre dos personas.

El esquema de conexión sería el siguiente:

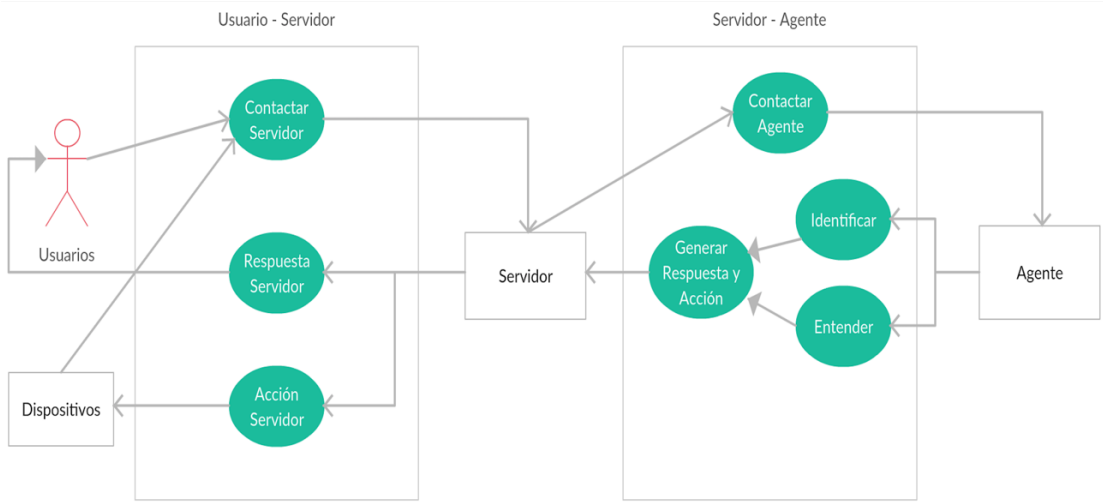


Figura 8. Diagrama de comunicaciones

Como se observa en el esquema anterior, todos los dispositivos y usuarios deben estar conectados a través del servidor, no permitiendo la conexión directa de un usuario con un dispositivo o viceversa.

Se puede apreciar con más claridad que la comunicación siempre es establecida a través del servidor. Todas estas comunicaciones se harían a través del protocolo TCP/IP nombrado anteriormente, y la respuesta generada por el servidor será un archivo de audio enviado al usuario como respuesta.

3.3.2 AGENTE INTELIGENTE

Además de diseñar un sistema que procese diferentes entradas y genere respuestas, se encontró, según lo comentado en el análisis de las soluciones, dos métodos para la identificación de usuario. Ambos métodos podrían formar solución al problema, sin embargo, el análisis con las redes neuronales artificiales es mucho más complejo a la hora de resolver los patrones. Es por ello, que se seleccionará el método 1 como el más prioritario, aunque se aplicará el método 2 para saber cuál ofrece mejores resultados.

3.3.3 PLACA DE COMUNICACIÓN DE LOS DISPOSITIVOS

En este punto se explicará como están constituidos los dispositivos para poder pertenecer a esta red IoT.

En primer lugar, al ser una comunicación a través de Internet tienen que tener la capacidad para acceder a este servicio. Principalmente, no todos los dispositivos están capacitados para conectarse a Internet, lo que sufrirá un cambio drástico en la industria al ir permitiendo esta capacidad. Hoy en día, las Smart TV son los dispositivos que han incorporado este acceso a Internet por las diferentes ventajas y servicio adicionales que se le pueden incorporar, una de ellas la accesibilidad a IoT. Al igual que estas televisiones, se debe seguir incorporando a diferentes dispositivos, como los electrodomésticos.

Para lograr esto, se debe ir implementado placas que permitan el acceso a Internet a través del protocolo Wi-Fi, que incluyan una pequeña pantalla de configuración del dispositivo o a través de una aplicación.

Como se comentó en el análisis de las soluciones, se puede optar por dos placas que tienen potencia suficiente para estas tareas: Arduino o Raspberry. Según el análisis, se optará por el uso de la placa Arduino debido a su facilidad de código, ya que no requiere un inicio de un sistema operativo, y que el código debe ser rápido al ser operaciones de

comunicación. Debe ser una placa con capacidad de acceso a Internet principalmente por Wi-Fi, para evitar uso de cables ethernet distribuidos por toda la vivienda o local.

Según el mercado de Arduino, tenemos dos opciones:

1. Placa con acceso a Internet incorporado.
2. Placa con potencia suficiente y adicionalmente otro componente extra que permita acceso a Internet.

Ambas opciones son totalmente válidas en este proyecto ya que la función principal de la placa es mantener una comunicación con el servidor y ejecutar las acciones dichas por el servidor.

Para simplificar el proyecto, se utilizará la 2ª opción, que estará formado por un Arduino UNO Rev3 y un módulo Wi-Fi, concretamente el ESP8266 como se ha comentado en el análisis de las soluciones.

3.3.4 DESARROLLO DE INTERFACES

Este punto está dedicado al diseño de las aplicaciones en determinados dispositivos para poder acceder al agente inteligente del servidor.

Este apartado abarca cómo deben ser las interfaces para estos dispositivos. Los dispositivos que se encuentran pueden variar desde ordenadores, consola del vehículo, teléfono móvil, electrodomésticos, etc.

Debe ser un entorno simple y fácil para el usuario desde el que se pueda configurar el dispositivo en la red IoT e interactuar con este a través del sistema de reconocimiento de voz. No por ser un sistema controlado por voz no se deba prestar atención al diseño de la interfaz, ya que también debe permitir el uso de configurar o comunicar cualquier aspecto a la red de manera escrita al agente inteligente.

Según el dispositivo, se diseña una interfaz diferente. No sería a niveles prácticos que la interfaz de los electrodomésticos y el teléfono móvil sean prácticamente iguales, ya que cada uno tiene un propósito en concreto. Por ejemplo, con el teléfono móvil interesa que se acceda a la red tanto por voz como escrito ya que pueden existir situaciones en la que se prefiera usar una u otra. Además, las operaciones que se puedan realizar sean más amplias, ya que es un dispositivo que suele estar en uso muchas horas lo que es

interesante tener un mayor control desde este. También sería aplicable este pensamiento hacia los ordenadores.

En el caso de los electrodomésticos es más práctico tener un sistema más simple de pequeñas acciones y un control de voz, ya que la interacción con estos será menos usual y, por tanto, se ahorra en recursos. El sistema del vehículo iría por el mismo camino que los electrodomésticos, interesa que su sistema de voz sea su principal método ya que tratará de acceder a la red mientras se conduce.

La conclusión es que los dispositivos que se conocen hoy en día sufrirán un cambio con la inclusión de accesibilidad a Internet y el control de estos a través de pantallas integradas con acceso a la red IoT.

3.4 ANÁLISIS DE SEGURIDAD

Las redes IoT permiten conectar diferentes dispositivos a través de la red de Internet. El problema de seguridad que se encuentra es en la propia definición de estas redes, es decir, el simple hecho de tener conexión a Internet provoca la necesidad de tener un mínimo de seguridad frente a las amenazas de Internet.

Aunque Internet se ha creado para muchos usos, se debe tener en cuenta que es un camino de comunicación compartido con más usuarios, por lo que, todos los datos viajarán a través de los demás y viceversa. Conectar a Internet implica tener en cuenta que existe la amenaza constantemente de corrupción de datos afectando directamente a nuestros datos personales.

Hoy en día, existen muchos casos de estas características en la que grandes empresas han sufridos múltiples ataques a sus servidores y con ello obtención de datos personales. Un ejemplo, puede ser el de Facebook, según *el país*, Facebook sufría un ataque que deja al descubierto datos de 50 millones de usuarios.[15] Esta noticia fue publicada el día 30 de septiembre de 2018, lo que significa que son problemas actuales que están sin resolver. En definitiva, ninguna empresa con gran capital puede defenderse de pequeños ataques contra sus usuarios.

Al diseñar un proyecto basado en Internet, se debe tener en cuenta, que los recursos actuales no aseguran al 100 % la invulnerabilidad de los datos de millones de usuarios que usarían redes IoT tales como pequeños y grandes locales, viviendas,

administraciones públicas, entre otros. Lo que significa que este proyecto está acompañado siempre de un pequeño problema de seguridad y privacidad.

Hay que tener en cuenta, que no se habla de datos como la dirección de la persona, su información privada y personal, números de cuenta bancarias, etc., sino se habla de problemas de control globales tal como la pérdida de acceso al control de nuestra red, como el control de nuestra vivienda, vehículo, cualquier dispositivo dentro de nuestra red, y esto a su vez, toda la información almacenada dentro de esta.

Por lo que este proyecto debe ofrecer siempre las medidas de seguridad diseñadas más actualizadas existente en el mercado, tales como verificación por controles biométricos, cifrado de datos, etc.

En caso de corrupción de datos, utilizar cualquier alternativa de manera que estos datos sean inaccesibles para los atacantes, como, por ejemplo, eliminación inmediata de todo los archivos y sistema de control.

3.5 VENTAJA ALGORÍTMICA

Una clave de este sistema de redes IoT es el agente inteligente. La opción más clara sería implementar el agente en cada dispositivo y que este se conecte a Internet si necesita algún dato que le este solicitando el usuario. El problema es cuando se solicite realizar alguna acción con el servidor simultáneamente a la vez que otro usuario, ¿cómo puede conocer el agente inteligente que dos usuarios solicitan acceder al servidor y gestionar las dos acciones de manera simultánea? No podría, en este caso, el agente aplicaría la acción de uno y posteriormente la del otro, sin tener en cuenta ninguna comunicación o dar algún error con algún dato ya que estaría siendo modificado por otro dispositivo o usuario.

La solución eficiente de este código es que la implementación del agente inteligente se recoge dentro del servidor, por lo que, tiene mayor facilidad de acceso a los diferentes datos y poder gestionar todas las llamadas hacia el servidor de los diferentes dispositivos y usuarios. Además, tendría la capacidad de realizar simultáneamente la conexión con diferentes usuarios y dispositivos, teniendo el control en tiempo real de quién está conectado, qué acción están solicitando y tomar las decisiones adecuadas.

Además de ofrecer mejoras en gestión y organización, tenemos ventajas de recursos consumidos. Implementar el código del agente inteligente en cada usuario implica gasto de memoria, gasto de recursos al procesar datos y, por tanto, presentar ciertos retrasos en la experiencia de usuario. En cambio, si las únicas funciones que realizan los dispositivos es comunicarse con el servidor, se usaría esos recursos para enviar y recibir datos, que son operaciones con menor procesamiento y, por tanto, necesitan menos recursos para ofrecer una experiencia de usuario mucho mayor, rápida y eficiente.

Otra ventaja que ofrece es que, a nivel de hardware en determinados dispositivos como el teléfono móvil, el vehículo o los propios electrodomésticos no es una cuestión crítica para el correcto funcionamiento. Un usuario podría gestionar a través de su electrodoméstico un parámetro del servidor, lo que significa que no es una operación que consuma recursos. Esto libera una gran cantidad de procesamiento que se puede utilizar para mejorar la sensación del usuario en la interfaz de cada dispositivo.

Para ilustrar la idea, se usa la siguiente figura:

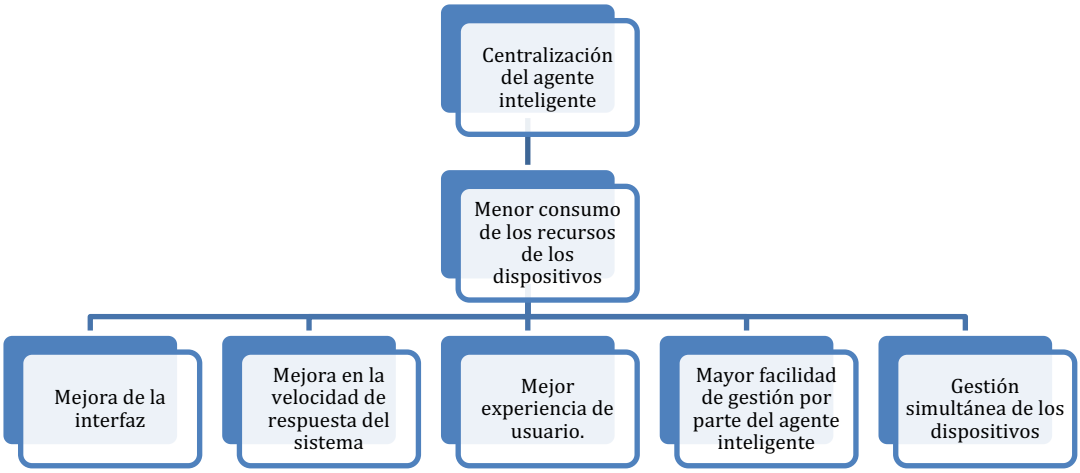


Figura 9. Ventajas de la centralización

3.6 ANÁLISIS DE LA PROTECCIÓN DE DATOS

Al tratar con datos personales y privados se debe respetar el derecho de los usuarios a su intimidad. Ya se trate de una empresa o una vivienda, la información que maneja las redes debe ser potencialmente segura, es decir, no puede existir de ninguna forma que un usuario o dispositivo ajenos a la red pueda acceder a dicha información.

Esta información puede estar compuesta por datos privados, tales como: documentación del local o vivienda, número de dispositivos conocidos, la información con la que trabaja la red, la información que se esta comunicando en tiempo real por la red, archivos de datos personalizados de cada usuario, entre otros.

Esta red IoT debe ofrecer la encriptación de toda esta información de manera segura, inaccesible al exterior y se aplique constantemente, es decir, el sistema envía la información de un punto a otro a través de Internet con un cifrado seguro. Solamente se podrá acceder a dicha información una vez el agente inteligente, a través de sensores biométricos comentado en puntos anteriores, verifique o autorice el acceso. En caso contrario, no se permitirá acceder y se bloqueará el acceso a dicho dispositivo hasta que se pueda verificar que el usuario tiene autorización. Si el problema fuera superior, y el acceso sea inevitable, el propio sistema provocará la eliminación de dichos archivos por no poder asegurar la seguridad de estos datos. El método más sencillo de no tener ningún problema es realizar copias de seguridad cada cierto tiempo por si el sistema detecta alguna situación similar.

3.7 COLABORACIÓN

Para la realización de las siguientes partes del proyecto, se ha usado el trabajo realizado por la compañía *Google* creando la API Speech:

1. Convertir texto a voz.
2. Convertir voz a texto.

Para la implementación de las siguientes partes del código realizado en Python, se han utilizado las siguientes librerías:

1. Speech_recognition
2. Numpy
3. Scipy.io: wavfile
4. Gtts: Google Text to Speech
5. Playsound

Speech Recognition

Esta librería se encarga de grabar un sonido a través del micrófono y transcribirlo en texto gracias a la API de Google.

Numpy

Esta librería aporta a Python herramientas para manejar vectores y matrices.

Scipy.io: wavfile

Esta librería se usa para leer los valores de los audios creados. En este caso, las frecuencias y los valores obtenidos de amplitud.

GTTS

Esta librería se usa para poder crear el audio de una oración creada por el propio agente inteligente.

Playsound

Esta librería se usa para reproducir audios. En este caso, se usará para reproducir los audios generados por el agente inteligente.

CAPÍTULO 4. DISEÑO DE LA SOLUCIÓN

En este apartado se va a tratar las herramientas utilizadas para cumplir con la solución definitiva, tanto de los programas utilizados, como las APIs o los esquemas de las funciones implementadas.

4.1 ANÁLISIS DE LAS HERRAMIENTAS

En este proyecto se va a desarrollar las acciones de comunicación, el agente inteligente, la identificación de usuario y el desarrollo de interfaces. Para realizar estas tareas, se utilizan diferentes herramientas.

La parte de comunicación se realiza a través del lenguaje de programación de Python. Se ha utilizado este lenguaje por su amplia variedad de librerías que contiene, ofreciendo al usuario realizar indefinidas acciones. En nuestro caso, la parte de comunicación consta de conectar al usuario y los dispositivos con el servidor, permitiendo recibir el audio o mensaje al servidor para su posterior procesamiento. También se realiza el envío de la respuesta del propio servidor a los diferentes dispositivos o usuarios para indicarle la acción o dar información, según la situación.

El agente inteligente se ha desarrollado en C++, ya que es un lenguaje que nos permite diseñar desde la parte más simple del mensaje. Tanto las funciones de Python como las de C++ se comunican a través de ficheros de datos.

La identificación de usuario se ha procesado con Python por la facilidad que ofrece al control de funciones y sus características, concretamente, series de Fourier y transformadas de Fourier.

Las interfaces de los dispositivos que no cuentan con un sistema operativo se han realizado a través de Arduino. Todos los electrodomésticos y dispositivos, que no suelen contener un sistema operativo como lo haría un teléfono móvil con Android o IOS, se crea una simple interfaz con el software de Arduino ya que no requiere determinadas características de procesamiento y es suficiente con la potencia de Arduino.

Las interfaces de dispositivos con sistema operativo Android, se ha utilizado el programa Android Studio. Este programa consta de los siguientes lenguajes:

1. Código: Java
2. Gráfico: XML

El servidor está formado por el sistema operativo Linux que contiene el programa en Python para las comunicaciones y C++ para realizar todo el procesamiento de los datos.

Función	Herramienta
Comunicaciones	Python
Identificación de usuario	Python
Procesamiento de texto	C++
Interfaces en dispositivos con Android	Android Studio: Java y XML
Placa de dispositivos	Arduino: C++

Tabla 3. Herramientas usadas para cada función del sistema

4.2 ARQUITECTURA DE LA IMPLEMENTACIÓN

La arquitectura de este TFG seguirá el siguiente esquema:

1. Comunicaciones:
    a. Comunicación Sockets vía Internet al servidor.

Las comunicaciones se realizarán a través de una red privada (VPN) para mejorar la seguridad de este. Esta red VPN conecta directamente al servidor principal que es el gestor de todas las comunicaciones dentro de la misma red, es decir, a este servidor se conectan todos los dispositivos que formen parte de este ecosistema IoT y se encargará de direccionar las comunicaciones, gestionar los recursos, permitir acciones, bloquearlas y obtener información de los diferentes dispositivos.

2. Diseño del agente inteligente
    a. Base de datos.
    b. Sistema de procesamiento.
    c. Sistema de generación de respuestas.

Para que el servidor funcione con éxito debemos acoplarlo con el agente inteligente capaz de entender e identificar qué usuario está intentando conectarse al servidor. El agente inteligente es el puente que une a los usuarios con el servidor.

3. Reconocimiento de voz:
    a. Transcribir audios.
    b. Análisis de la voz para su identificación.
    c. Generación de audio a partir de texto.

Se tratará de usar herramientas para transformar el audio en texto y ser comprensible por el agente inteligente, además de analizar los diferentes audios para poder diferenciar a los usuarios. A su vez, el agente estaría capacitado para transformar en la dirección inversa, es decir, pasar de texto a voz para comunicarse con el usuario.

4. Implementación en Android:
    a. Configurar comunicación de Android.
    b. Realizar aplicación.

Aplicaremos la comunicación de un dispositivo Android al servidor, diseñándola a través de la herramienta de Android Studio.

5. Actuadores en Arduino:
    a. Configurar control de los dispositivos.

Arduino será el que accione todas las instrucciones enviadas por el servidor para controlar algunos dispositivos físicos, como por ejemplo en una vivienda, control de la iluminación, electrodomésticos, entradas y salidas de la vivienda, etc.

4.3 ESQUEMA DE LA IMPLEMENTACIÓN

Como se ha comentado, el esquema general de implementación será el siguiente:

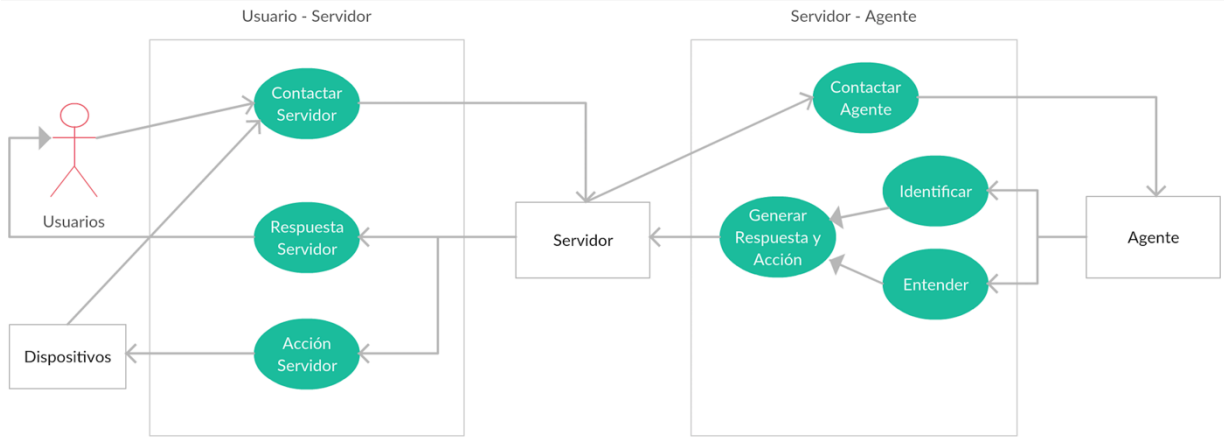


Figura 10. Esquema general de implementación

El usuario y los dispositivos deben contactar con el servidor para realizar cualquier acción. Este contactará con el agente inteligente para procesar la información y generar la respuesta deseada.

Ahora se procederá a representar en detalle cada una de las funciones realizadas por el agente inteligente.

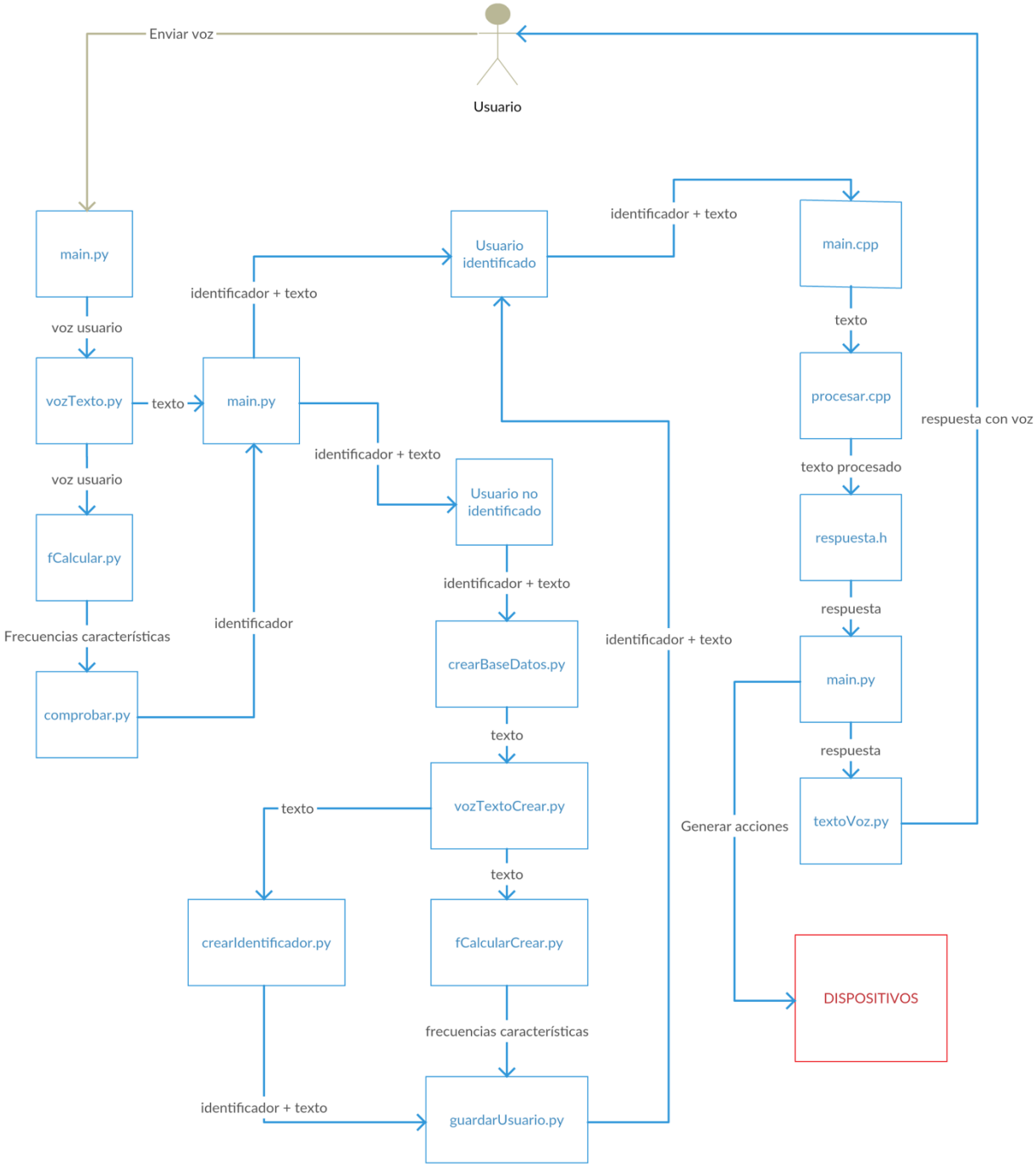


Figura 11. Esquema de implementación detallado

La parte de la izquierda del diagrama corresponde a la parte de identificación.

El programa comienza con la función *main.py* que simplemente activa el funcionamiento del servidor y conecta con la parte del procesamiento para el posterior envío de datos. Cuando el usuario se comunica, se utiliza la función *voz_texto.py* para convertir el audio en texto y que pueda ser procesado por el agente inteligente. A su vez, se obtienen, del audio del usuario, los valores de la señal en función de la frecuencia que se envía a la función *fCalcular.py*. Esta función se encarga de calcular los valores de la frecuencia características del audio del usuario. Se utiliza la función *comprobar.py* para verificar si esas frecuencias características coinciden con algún usuario en la base de datos del sistema. Si la respuesta es positiva, se obtiene el identificador de quién está comunicándose con el servidor y cuál es el mensaje. En cambio, si la respuesta es negativa, un usuario autorizado deberá permitir al sistema agregar al usuario nuevo.

En el bloque de usuario no identificado, observaremos los pasos a seguir para agregar a un usuario al servidor. En primer lugar, se llama a la función *crearBaseDatos.py* para inicializar el proceso de usuario nuevo. Esta función envía el texto del usuario a *vozTextoCrear.py* que solicitará al usuario diferentes audios para calcular las frecuencias características a través de la función *fCalcularCrear.py*, y su identificador con la función *crearIdentificador.py*. Una vez obtenidas, se enviará a *guardarUsuario.py* que grabará en la base de datos el usuario creado con su identificador. Este identificador y el texto dicho por el usuario se enviará a usuario identificado para su procesamiento.

En el bloque de usuario identificado, se enviará hacia la función *main.cpp* para el procesamiento del texto.

En la función *main.cpp* se utilizará el identificador para acceder a la base de datos del usuario y poder procesar el texto y crear la respuesta. El texto se envía a la función *procesar.cpp* para obtener el texto procesado y así enviarlo a la función *respuesta.h*.

Una vez se obtiene la respuesta, se envía de nuevo al *main.py* para convertir la respuesta en voz con *textoVoz.py* y enviársela al usuario, además de generar la acción que se envía al dispositivo si fuera necesario.

Programa	Función
Main.py	Se encarga de las comunicaciones
vozTexto.py	Convertir a texto el audio del usuario y devuelve el identificador de este
fCalcular.py	Calcula las frecuencias características del audio recibido
Comprobar.py	Comprueba en la base de datos para hallar el identificador
crearBaseDatos.py	Crea un usuario nuevo en la base de datos
vozTextoCrear.py	Inicializa el proceso de crear usuario grabando los audios necesarios
fCalcularCrear.py	Calcula las frecuencias características de los audios para agregar al usuario
crearIdentificador.py	Crea el nuevo identificador para el usuario
guardarUsuario.py	Guarda al usuario en la base de datos
Main.cpp	Recibe la información de main.py y envía los datos al agente para procesar el texto
Procesar.cpp	Procesa el texto
Respuesta.h	Genera la respuesta
textoVoz.py	Convierte el texto en audio

Tabla 4. Descripción de funciones del sistema

CAPÍTULO 5. IMPLEMENTACIÓN

En este punto se explicará en profundidad las ideas generadas para desarrollar las diferentes etapas del proyecto.

Se dividirá en los siguientes puntos:

1. Comunicaciones.
2. Identificación de usuario.
3. Agente inteligente.
4. Arduino.

5.1 COMUNICACIONES

Las comunicaciones que se establecen en el proyecto se pueden dividir en externas e internas. Las comunicaciones externas son las que producen entre los usuarios y el propio servidor, mientras que las comunicaciones internas son entre los propios programas entre sí.

Se ha analizado en las comunicaciones externas aplicar el protocolo TCP/IP. Estas comunicaciones se realizarán y serán gestionadas a través del programa *main.py* con el uso de sockets.

Las comunicaciones internas son necesarias para que los programas desarrollados en Python y en C++ se comuniquen entre sí para que cada uno realice su función como se vio en la figura 11. Si no existiera esta comunicación, no sería posible determinar cuando el agente ha recibido un mensaje procedente de algún dispositivo o usuario. Para llevar a cabo esta comunicación se ha implementado una comunicación a través de sockets en modo local y con el protocolo TCP/IP. En este se implementa los sockets tanto para enviar como recibir datos en los dos programas principales, *main.py* y *main.cpp*. Para conectar los programas en *localhost*, se utiliza la dirección IP *127.0.0.1* que apunta directamente al propio ordenador que ejecuta la conexión.

5.2 IDENTIFICACIÓN DE USUARIOS

Para la identificación de usuario se aplicará el método 1 de análisis de frecuencias. Este sistema se encargará de generar la transformada rápida de Fourier (*Fast Fourier Transform)* para hallar las frecuencias con mayor aportación de información.

Se comienza a analizar un ejemplo de audio al que se le ha aplicado la transformada rápida de Fourier del mismo usuario, pero con frases distintas:

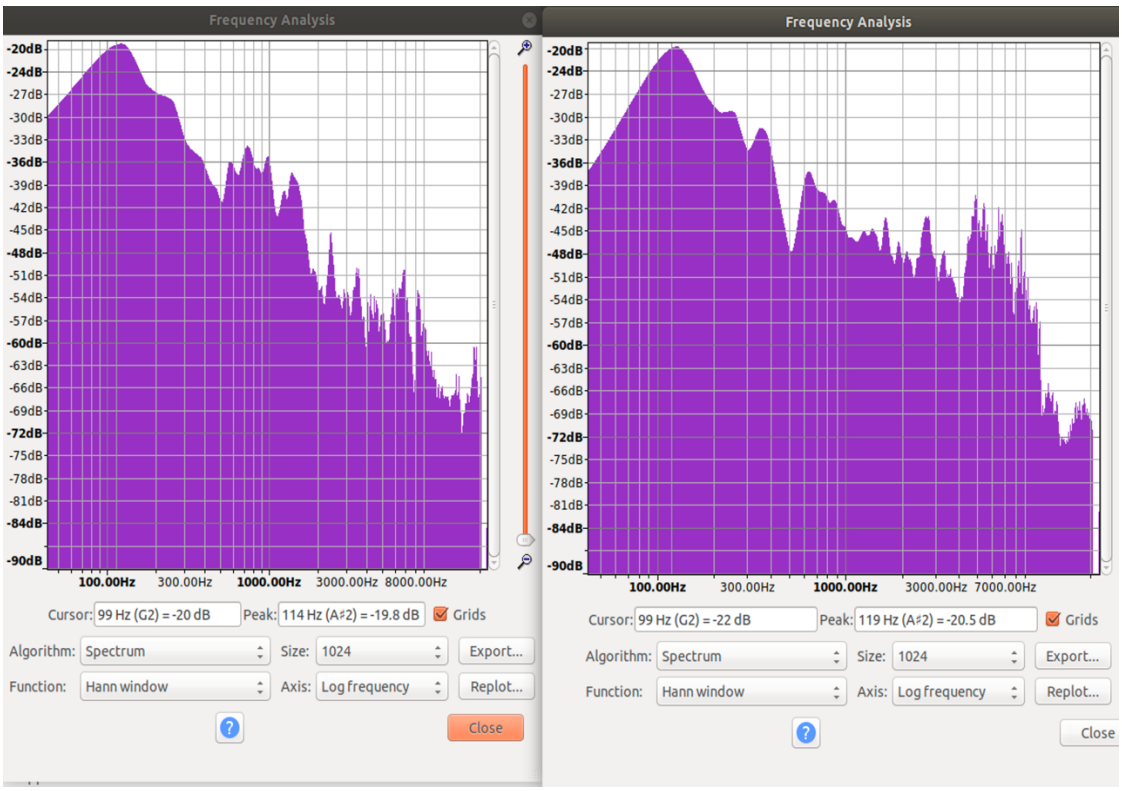


Figura 12. Análisis de frecuencia usuario.

El análisis se realiza en torno a la frecuencia de 80 Hz hasta los 1000 Hz. Como podemos observar, en este rango de frecuencias se aprecia la similitud entre uno y otro, ya que se trata del mismo usuario. En la parte inferior indica una medida para ambos a 99 Hz obteniendo un resultado de -20 db y -22 db respectivamente, lo que nos confirma la proximidad de ambas medidas. Además, los picos más altos se dan en la frecuencia 114 Hz y 119 Hz respectivamente, la cuál formarán parte de una de las frecuencias características de este usuario.

Ahora comparamos el audio de otra persona con el usuario anterior para obtener las diferencias:

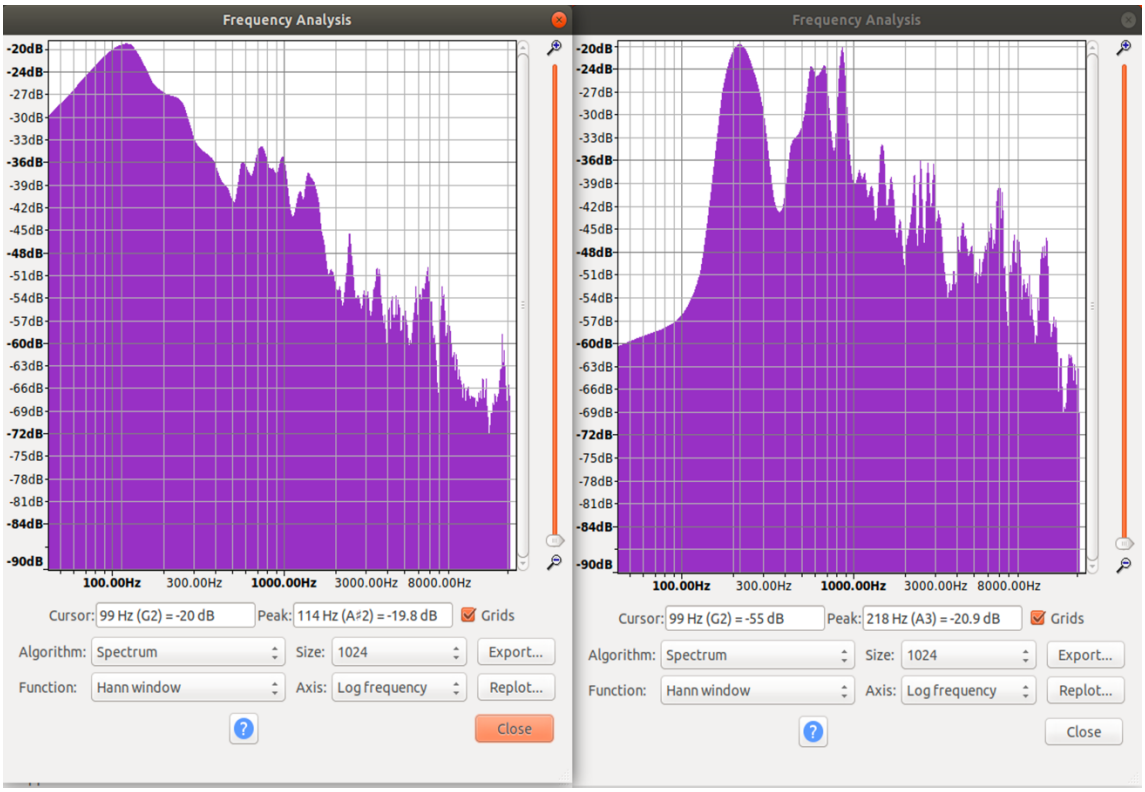


Figura 13. Análisis de frecuencia entre usuarios distintos.

En este caso, la diferencia es notable a simple vista. Entre el rango de frecuencias de 80 Hz y 1000 Hz podemos concluir que las frecuencias características serán bastante distintas. Si observamos la parte inferior, con un valor de frecuencia de 99 Hz obtenemos los valores de -20 db y -55 db respectivamente, lo que confirma esta diferencia. Además, la frecuencia que tiene el valor mayor es de 114 Hz y 218 Hz respectivamente, que formarán parte de las frecuencias características al igual que el caso anterior.

En este caso se han analizado dos usuarios diferentes, uno masculino y otro femenino, de ahí la diferencia tan notable entre ellos. Por lo que procederemos a ver un ejemplo de usuarios del mismo género y analizar si se sigue cumpliendo la diferencia.

En este caso, se compara dos usuarios masculinos:

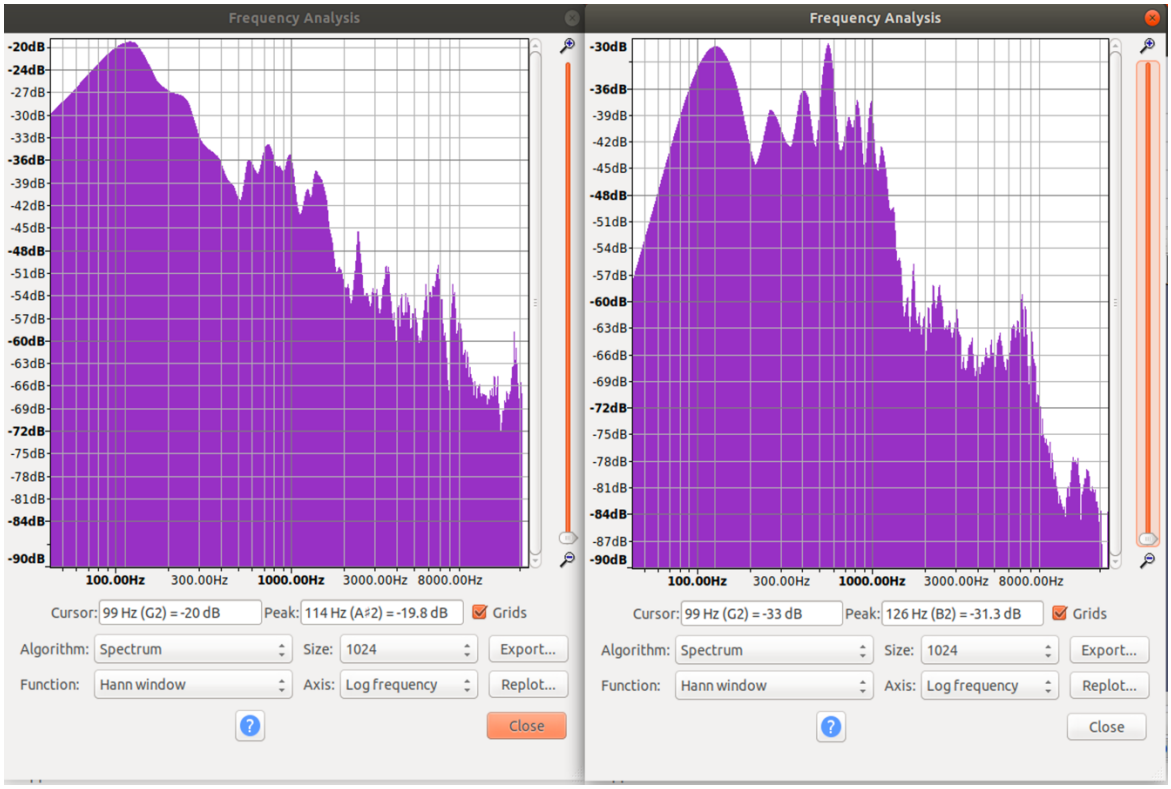


Figura 14. Análisis de frecuencias entre usuarios masculinos.

En este caso, la diferencia no es tan considerable. Al ser registros masculinos, se suelen obtener registros muy parecidos. Si observamos entre el rango de 80 Hz a 1000 Hz, observamos que las frecuencias no coinciden tampoco, por lo que las frecuencias características de ambos serán diferentes. En la parte inferior se observa que para una frecuencia de 99 Hz obtenemos los valores de -20 db y -33 db respectivamente. La diferencia con el caso anterior es bastante menor, sin embargo, no anula la capacidad al sistema para poder diferenciar a ambos usuarios.

En este caso, analizaremos dos audios de usuarios con un género femenino:

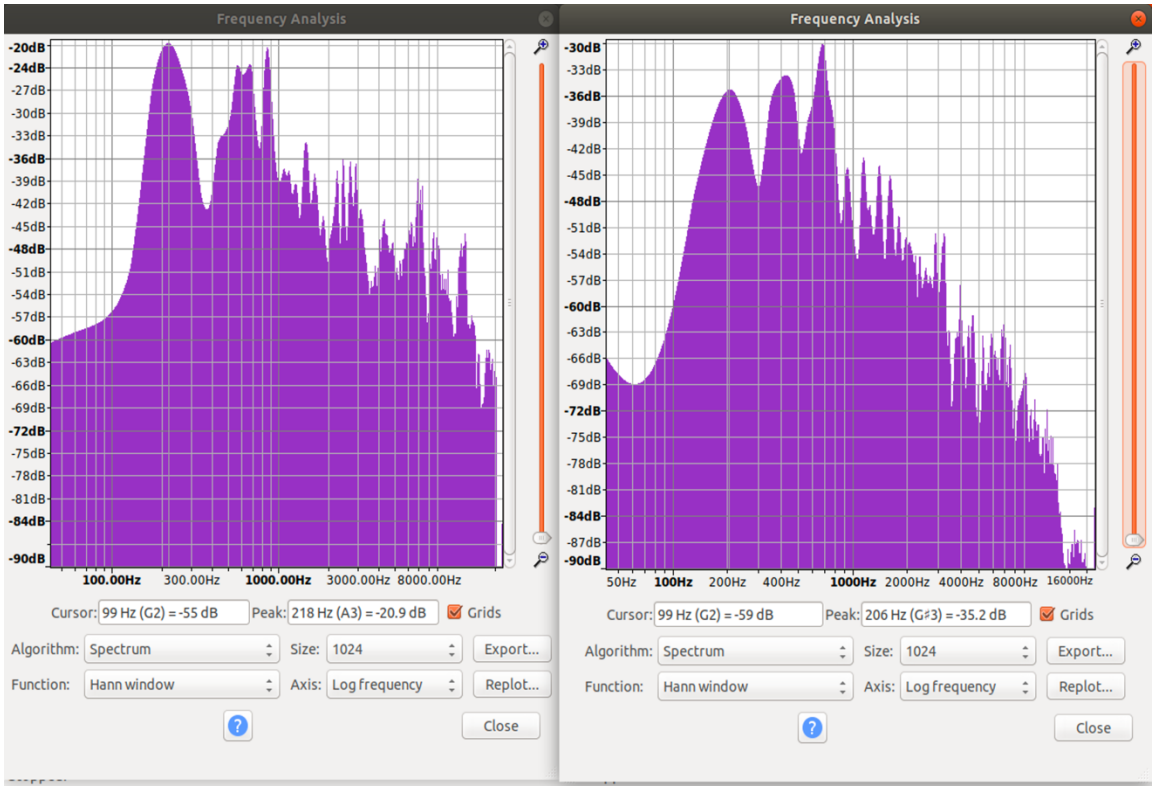


Figura 15. Análisis de frecuencias ente usuarios femeninos.

Al igual que en el caso anterior, estos dos usuarios tienen parecidos notables, sin embargo, existen diferencia entre ellos. Entre el rango de frecuencias de 80 Hz y 1000 Hz se puede notar la diferencia entre ambos. En la parte inferior, observamos para un valor de 99 Hz los valores de -55 db y -59 db respectivamente y los valores picos son de 218 Hz y 206 Hz respectivamente. Estos datos confirman que, a pesar de la similitud, el agente sería capaz de diferenciar entre uno y otro.

En conclusión, la distinción entre usuarios es viable y aplicable. En casos de comparación entre usuarios del mismo género, el sistema deberá tener información suficiente para notar las diferencias entre ellos que, en definitiva, son las posibles situaciones más complejas que encontrará el sistema.

Para la implementación de este sistema se procederá de la siguiente manera:

Se calculará el valor más alto y se hallará el valor de la frecuencia correspondiente. Esta frecuencia se denotará como la frecuencia característica. El usuario debe coincidir con la frecuencia característica con un margen de error definido para que se pueda analizar las siguientes frecuencias. En caso negativo, se descarta dicho usuario.

Para el análisis de las otras frecuencias, se utilizan cuatro ficheros que contendrán las frecuencias características de cada archivo de audio. El servidor le pedirá al usuario que grabe 4 archivos de audio. Para cada archivo de audio se calculará la frecuencia máximo en diferentes rangos de frecuencia. Para ello, usando el valor máximo de la señal, se determina un valor límite para poder considerar un máximo dentro de cada rango. Por tanto, para cada rango se hallará un máximo con el que se obtendrá la frecuencia máxima de ese rango y será esta la que se guarde en el fichero. Si no existe ningún valor superior al valor límite, no se añadirá ninguna frecuencia y se pasará al siguiente rango.

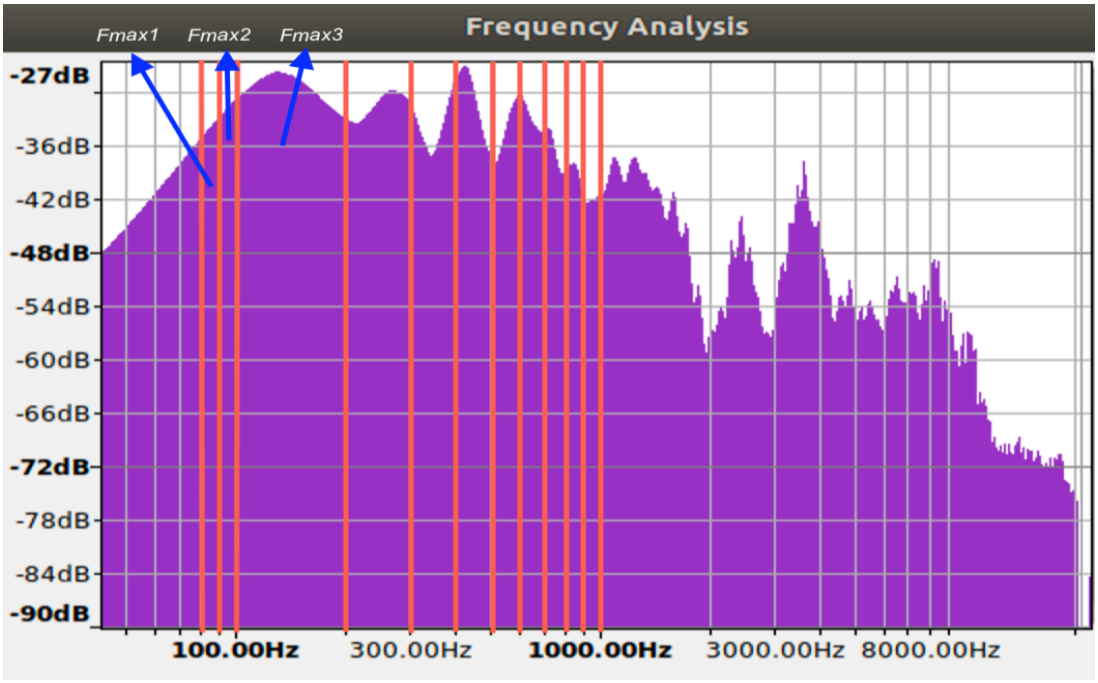
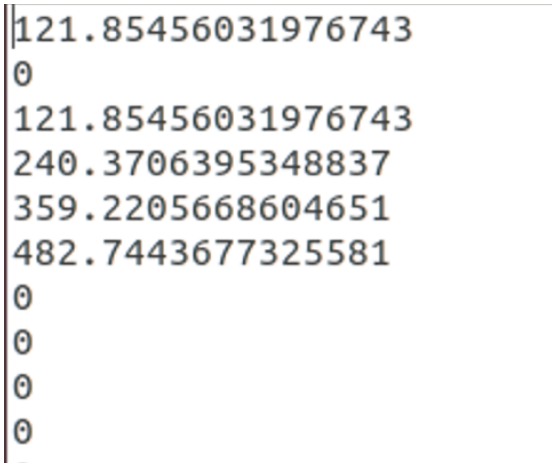


Figura 16. Análisis de frecuencia por rangos



Como se observa en la figura 16, se aplican los diferentes rangos de frecuencias limitados por las franjas en color rojo y se obtienen las frecuencias máximas de cada zona. Esas frecuencias se guardan en un fichero tal y como se muestra en la figura 17, en el que en primer lugar se encuentra la frecuencia máxima absoluta que caracteriza principalmente al usuario, y luego las frecuencias restantes de cada zona.

Figura 17. Frecuencias características

Para la implementación del método 2 (análisis de frecuencias con inteligencia artificial), se utilizarán las transformadas de Fourier, pero la idea principal es diferente. En este caos, se utilizarán los valores para determinar el patrón de cada usuario.

Como se ha comentado, las redes neuronales artificiales se encargarán de calcular las ponderaciones para determinar los valores de salida. La idea será la siguiente:

Se grabarán cuatro audios al igual que el método anterior. Antes de aplicar las redes, debemos realizar un ajuste en los valores:

- Se debe normalizar los valores entre 0 y 1 dividiendo los valores entre el valor máximo.
- Se deben centrar las cuatro muestras, es decir, los valores máximos deben estar en la misma posición.
- Se deben ajustar para tener las mismas dimensiones.

Una vez se ha realizado el ajuste se aplicará la red para hallar las ponderaciones. Suponemos que los cuatro archivos son A, B, C, D. Estos archivos se aplicarán de la siguiente manera:

- Ponderaciones 1:
  - Se aplicarán como entradas: B, C, D
  - Se utilizará como salida: A

- Ponderaciones 2:
    - Se aplicarán como entradas: A, C, D
    - Se utilizará como salida: B

- Ponderaciones 3:
    - Se aplicarán como entradas: A, B, D
    - Se utilizará como salida: C

- Ponderaciones 4:
    - Se aplicarán como entradas: A, B, C
    - Se utilizará como salida: D

Estas cuatro ponderaciones se deberán configurar, reduciendo o aumentando el número de iteraciones, el ratio de aprendizaje, y el número de capas internas.

Una vez obtenidas las ponderaciones, solo debemos aplicarlas a cada entrada y comprobar que coincide con cada salida respectivamente.

Un ejemplo para representarlo sería el siguiente:

Archivos guardados: A, B, C, D

Ponderaciones: Pon1, Pon2, Pon3, Pon4

Entrada: Audio

Las redes neuronales calcularán:

Salida_A = Audio · Pon1

Salida_B= Audio · Pon2

Salida_C = Audio · Pon3

Salida_D = Audio · Pon4

Una vez calculadas las salidas, se debe comprobar:

Si Salida_A ≈ A ⇒Porcentaje de acierto

Si Salida_B ≈ B ⇒Porcentaje de acierto

Si Salida_C ≈ C ⇒Porcentaje de acierto

Si Salida_D ≈ D ⇒Porcentaje de acierto

Los porcentajes indicarán cuánto es la posibilidad de que sea ese usuario. Si aplicamos este procedimiento con los usuarios registrados en la base de datos, el de mayor porcentaje sería el usuario que le corresponde el audio de entrada. Si el porcentaje es muy próximo a 0, significa que ese usuario no corresponde con la entrada y se descartaría.

```
('Usuario a analizar: ', 'Josue')


('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.00351079, 0.00343977,
       0.00647715]))
('Valores salida teorico: ', array([1, 1, 1, ..., 1, 1, 1]))
('Valores calculados: ', array([0.02549474, 0.04466951, 0.0133371 , ..., 0.00689445, 0.00792257,
       0.00750842]))
Porcentaje de acierto: 0 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.00351079, 0.00343977,
       0.00647715]))
('Valores salida teorico: ', array([1, 1, 1, ..., 1, 1, 1]))
('Valores calculados: ', array([0.01898577, 0.01569282, 0.02438983, ..., 0.00478382, 0.01277911,
       0.02035421]))
Porcentaje de acierto: 0 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.00351079, 0.00343977,
       0.00647715]))
('Valores salida teorico: ', array([1, 1, 1, ..., 1, 1, 1]))
('Valores calculados: ', array([0.00919174, 0.00918497, 0.01787305, ..., 0.00547311, 0.00672891,
       0.00732548]))
Porcentaje de acierto: 3 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.00351079, 0.00343977,
       0.00647715]))
('Valores salida teorico: ', array([1, 1, 1, ..., 1, 1, 1]))
('Valores calculados: ', array([0.02828663, 0.02828343, 0.04295643, ..., 0.0030912 , 0.00398486,
       0.00102567]))
Porcentaje de acierto: 0 %
```

Figura 18. Resultados redes neuronales. 1 de 2

```
('Usuario a analizar: ', 'Montse')


('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.01097134, 0.01459652,
       0.01724988]))
('Valores salida teorico: ', array([0, 0, 0, ..., 0, 0, 1]))
('Valores calculados: ', array([-0.01001224,  0.01843352,  0.00368991, ...,  0.04059436,
        0.24824728,  0.07487651]))
Porcentaje de acierto: 3 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.01097134, 0.01459652,
       0.01724988]))
('Valores salida teorico: ', array([0, 0, 0, ..., 0, 0, 1]))
('Valores calculados: ', array([-0.01001315,  0.01843595,  0.00369042, ...,  0.04059862,
        0.24827504,  0.0748851 ]))
Porcentaje de acierto: 3 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.01097134, 0.01459652,
       0.01724988]))
('Valores salida teorico: ', array([0, 0, 0, ..., 0, 0, 1]))
('Valores calculados: ', array([-0.01001419,  0.01843873,  0.00369115, ...,  0.04060322,
        0.24830413,  0.07489388]))
Porcentaje de acierto: 3 %



('Valores entrada: ', array([0.02112641, 0.02112641, 0.00554082, ..., 0.01097134, 0.01459652,
       0.01724988]))
('Valores salida teorico: ', array([0, 0, 0, ..., 0, 0, 1]))
('Valores calculados: ', array([-0.01001373,  0.01843742,  0.00369047, ...,  0.04060171,
        0.24829655,  0.07489216]))
Porcentaje de acierto: 3 %
```

Figura 19. Resultados redes neuronales. 2 de 2

Las dos figuras anteriores serían un ejemplo real en el que se aprecia los resultados obtenidos para dos usuarios diferentes.

5.3 AGENTE INTELIGENTE

Las funciones del agente inteligente se pueden dividir en:

1. Comprender lenguaje natural.
2. Respuesta del agente.

COMPRENDER LENGUAJE NATURAL

Para que el sistema sea efectivo, debe ofrecer una alta capacidad de comprensión, no puede permitirse funcionar a través de comandos o que el usuario diga una serie de palabras clave para hacer determinadas acciones. No podemos permitir esto debido a que cada usuario se expresa de manera diferente lo que no podemos considerar que el usuario deba aprender a *"hablar"* de una forma determinada.

Es por ello, que la comprensión del lenguaje natural debe ser algo esencial en este proyecto. Como se ha comentado en el análisis de soluciones, se utilizará un sistema basado en *DialogFlow* de Google. En primer lugar, el usuario mandará un audio a través del servidor que se encargará de transcribirlo con el uso de la API Speech de Google. Esta API se encargará de convertir el texto en voz o viceversa, por lo que nos interesará aplicarla cuando el usuario contacta con el servidor o cuando el servidor genera una respuesta para el usuario.

Una vez se obtiene el texto que ha dicho el usuario, se debe comprenderlo de tal forma que el sistema origine una respuesta eficiente y clara. En este caso, se constará de un sistema de archivos de entrada en el que se queda registrado las palabras clave que se pueden encontrar y otro sistema de archivos de salida con las respuestas que puede generar en función de la entrada recibida. Una de las características del *DialogFlow* es utilizar listas de palabras de entrada que puedan obtener una lista de palabras de salida, eligiendo estas de manera aleatoria para crear ese efecto de naturalidad en el sistema y que no sea siempre la misma respuesta generada. Sin embargo, puede darse el caso en el que ciertas palabras tengan un significado diferente en determinadas situaciones como la siguiente:

Situación 1:

USUARIO: ¿quién soy?

AI: Eres…

Situación 2:

USUARIO: soy…

AI: Encantado.

En estos dos casos, se utiliza la palabra "soy" del usuario con significados diferentes, una para expresar una pregunta y otra para dar información al servidor. Estos dos casos, deben ser diferenciados por el propio agente, lo que el sistema debe tener en cuenta no solo las palabras usadas sino su intención.

Para solucionar este problema, se ha de establecer una normativa que ha de seguir el sistema de forma que se comprenda la idea del usuario. Para ello se definen 3 aspectos esenciales que debe hallar el agente inteligente:

1. *Acción:* se define la acción que esta requiriendo el usuario. Por ejemplo: consultar, llamar, ver, notificaciones, etc. Se suele detectar los verbos de la oración que son los que definen lo que necesita el usuario.
2. *Dirección:* se define la intención de la acción, es decir, si la acción va referida al propio usuario o al propio sistema. Por ejemplo: ¿Quién eres? -> acción referida hacia el agente, mientras que ¿Quién soy?, es referida hacia el usuario.
3. *Estructura:* se diferencia si en la frase existe pregunta o si es una acción directa.

Existen dos datos más que se denomina Grupo Especial (GE) y datos (D). El grupo especial (GE) hace referencia a alguna palabra en especial como saludos y despedidas. Datos (D) es la parte que acompaña a la acción y que le sirve al sistema como información, por ejemplo, el nombre de una persona, el número de teléfono, el nombre del archivo, etc.

Si unimos esta normativa nos queda la siguiente forma: GEADED. Este protocolo GEADED son las siglas de las diferentes partes que se calculan al comunicarse el usuario.

Para entender esto mejor, pongamos algunos ejemplos:

EJEMPLO 1:

USUARIO: Buenos días, ¿quién soy?

En este ejemplo los parámetros calculados son los siguientes:

Grupo especial: Buenos días.

Acción: soy

Dirección: usuario

Estructura: pregunta

Datos: ninguno

El sistema, a partir de estos datos, generará la respuesta correspondiente:

AI: Buenos días, eres …

EJEMPLO 2:

USUARIO: hola, ¿quién eres?

En este ejemplo los parámetros calculados son los siguientes:

Grupo especial: hola

Acción: eres

Dirección: sistema

Estructura: pregunta

Datos: ninguno

El sistema, a partir de estos datos, generará la respuesta correspondiente:

AI: hola, soy un agente inteligente.

EJEMPLO 3:

USUARIO: me gustaría llamar a Josué.

En este ejemplo los parámetros calculados son los siguientes:

Grupo especial: ninguno

Acción: llamar

Dirección: infinitivo

Estructura: afirmativo

Datos: Josué

El sistema, a partir de estos datos, generará la respuesta correspondiente:

AI: Llamando a Josué.

La generación de la respuesta se explicará en el siguiente apartado.

El cálculo de la dirección sigue también un proceso adicional denominado Protocolo 1234. Cada uno de estos números significan los 4 diferentes estados que pueden darse en la frase del usuario.

Los distintos estados de la dirección son:

1. Infinitivo.
2. Dirección usuario.
3. Dirección sistema inteligente.
4. Imperativo.

Por ejemplo, el verbo ser, cumpliría la siguiente forma:

1. Ser
2. Soy
3. Eres
4. Eres

Algunos verbos no utilizan estas 4 direcciones por lo que se copia la anterior.

Por tanto, la conclusión es bastante clara que solamente buscando la palabra se pierde información de la intención del usuario. Aplicando el protocolo GEADED y el protocolo 1234 en el apartado de la dirección, se aporta esta información para la respuesta del agente inteligente.

Una vez hallamos la forma de detectar la información, debemos almacenar en un archivo las diferentes palabras o combinaciones según los protocolos. Para ello, la siguiente figura se puede observar un fichero de las posibles entradas del usuario:

```
ser_1
soy_1
eres_1
es_1
llamar_2
llama_2
llama_2
llama_2
notificacion_3
notificaciones_3
notificaciones_3
notificaciones_3
```

Figura 20. Ejemplo fichero de entrada

En este ejemplo, encontramos 3 acciones claras: ser, llamar y notificación. El número que se encuentra a la derecha indica al sistema que acción es, si es la número 1, número 2 o número 3. Si observamos detenidamente, el orden de estas palabras sigue el protocolo 1234:

1. Ser
2. Soy
3. Eres
4. Es

Para el apartado de grupo especial, se diferencia en 3 subgrupos:

1. Saludos.
2. Saludos especiales.
3. Despedida.

Por tanto, obtendríamos un archivo de esta forma:

```
1_saludos
hola
2_especiales_depende_hora
buenos_dias
buenas_tardes
buenas_noches
3_despedirse
adios
hasta_luego
hasta_pronto
hasta_la_vista
hasta_la_proxima
```

Figura 21. Fichero grupo especial

Estos tres grupos formarían parte de grupos especiales, de forma que, si se dan en la frase del usuario se detecte de manera independiente. Esto es útil para el sistema de respuesta que se explicará en el apartado siguiente. La única diferencia del grupo 2 es que depende de la hora, es decir, la respuesta va a estar definida en función de la hora que sea, ya que no tiene sentido que a las 21:00h el sistema responda buenos días.

El apartado de estructura vendrá definido por aquellas palabras que puedan entonar una pregunta, como, por ejemplo, *quién, cómo, cuándo, dónde*, etc.

En este caso, solo está implementado *quién*:

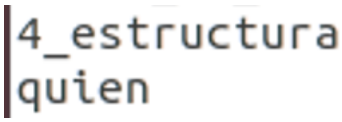


Figura 22. Fichero estructura

El apartado de datos se encargará de buscar un sustantivo importante en función de la acción o un nombre propio.

RESPUESTA DEL AGENTE INTELIGENTE

Para la respuesta del agente, partimos de los datos obtenidos en el punto anterior del protocolo GEADED. El sistema de respuesta recoge esta información para generar una respuesta en función de lo que solicita el usuario.

Al igual que ocurría para detectar la información, se usará un fichero de respuestas vinculado con el fichero de entradas.

Las respuestas se generan en cuatro partes:

1. Grupo Especial
2. Bloque 1.
3. Verbo
4. Bloque 2.

La parte del grupo especial (GE) hace referencia algunas partes de la conversación específicas como los saludos o despedida, si el usuario saluda o se despide, el agente inteligente hará lo mismo. Las respuestas de este grupo se realizan de manera aleatoria al despedirse y saludar. Como comentamos en el punto anterior, existe también un grupo especial que dependía de la hora, este, por tanto, no aplica la aleatoriedad.

La parte del verbo se utilizará para poder conjugar la frase del agente inteligente dependiendo de los parámetros obtenidos a través del GEADED. Si la dirección es usuario o agente, se usará la dirección contraria, es decir, si la frase del usuario hace

referencia a una cuestión del agente, este deberá responder con una forma verbal que haga referencia así misma o lo que es lo mismo en primera persona.

Los bloques 1 y 2 es información adicional que pueda usar el agente en determinados casos tanto del propio agente como el usuario que esta hablando.

Para comprender mejor las direcciones, pondremos el mismo ejemplo que en el punto anterior:

EJEMPLO 2:

USUARIO: hola, ¿quién eres?

En este ejemplo los parámetros calculados son los siguientes:

Grupo especial: hola

Acción: eres

Dirección: agente inteligente

Estructura: pregunta

Datos: ninguno

El sistema, a partir de estos datos, generará la respuesta correspondiente:

AI: hola, soy un agente inteligente.

Podemos observar que la respuesta se divide en 3 partes:

La primera parte correspondería a *hola*, ya que el usuario saludó, por lo que el agente también saluda.

El verbo, en este caso, será en primera persona *soy*, ya que el usuario dijo *¿quién eres?,* haciendo referencia al agente, por lo que la dirección es el propio sistema.

El bloque 1 y 2 correspondería a la parte de datos adicionales que completa el propio agente. Para ello, el agente, a parte de almacenar las diferentes palabras o verbos, debe conocer el significado para poder dar la información que requiere el usuario. En este caso, el verbo ser le permite al agente definir qué es, a lo que responde, *agente inteligente.*

Para poder realizar todo este proceso, se utilizan los parámetros del GEADED. Como observamos, cada palabra o verbo estaba seguida de un número, el cual, hace referencia a la respuesta que debe ejecutar el sistema.

Para entenderlo mejor, primero explicaremos cuál es la normativa de los ficheros de base de datos de salida:

El sistema funciona de la siguiente manera:

```
4_ser
ser_2_3_un_agente_inteligente
ser_2_3_trobox
ser_2_2_Josue
ser_3_2_eres
ser_3_3_soy
ser_0_2_Encantado
```

Figura 23. Fichero de salida ejemplo verbo ser

Primero el 4 hace referencia al primero de los verbos ya que el 1 2 y 3 queda reservado para el grupo especial (GE). Por lo que, si la acción es ser, el número asociado es 1 en el fichero de datos de entrada que corresponde a 4 en el fichero de salida según, por lo que sabe que la acción es el verbo ser.

El esquema que sigue es el siguiente:

Verbo_x_y_[]

Verbo: acción, en este caso, *ser*.

X = 0 → Bloque 1

X = 2 → Bloque 2

X = 3 → conjugaciones de los verbos

Y = 1 → dirección infinitivo.

Y = 2 → dirección usuario

Y = 3 → dirección sistema inteligente

Y = 4 → dirección imperativo.

Como se ha comentado anteriormente, la respuesta se divide en 3 partes, bloque 1, verbo y bloque 2. Cuando se conoce el verbo que esta usando el usuario, sabemos en qué bloque de respuesta tenemos que buscar la información.

La x nos indica si la información que se compone es del Bloque 1, Bloque 2 o una conjugación.

La y nos indica que información usar en función de la dirección que tenga.

Teniendo esto en claro, procedemos a examinar la respuesta del ejemplo anterior:

USUARIO: *hola, ¿quién eres?*

Grupo especial: hola

Acción: eres

Dirección: sistema inteligente

Estructura: pregunta

Datos: ninguno

La respuesta se formará de la siguiente manera:

En primer lugar, el agente no crea los valores de *grupo especial, acción, dirección, estructura y datos* escribiendo los valores reales, sino utiliza los valores de referencia. Usando la información del fichero de base de datos de entrada, procedemos a escribir realmente el protocolo GEADED:

```
1_saludos
hola
2_especiales_depende_hora
buenos_dias
buenas_tardes
buenas_noches
3_despedirse
adios
hasta_luego
hasta_pronto
hasta_la_vista
hasta_la_proxima
```

Figura 24. Fichero grupo especial

El valor de *hola* corresponde al primer grupo de bloque de grupo especial, lo que se colocará un 1 en grupo especial.

```
4_estructura
quien
```

Figura 25. Fichero estructura

Cuando se detecta la palabra *quién* se pone a 1 la estructura.

Cuando se detecta la palabra *eres* vemos que corresponde a la posición 3 del verbo ser. La posición en la que se encuentre dará el valor de la dirección, y el número que tiene asociado indica al sistema que verbo se trata, por lo que en acción colocaremos un 1.

```
ser_1
soy_1
eres_1
es_1
llamar_2
llama_2
llama_2
llama_2
notificacion_3
notificaciones_3
notificaciones_3
notificaciones_3
```

Figura 26. Acción y dirección

Por tanto, según esta información se calcula el protocolo GEADED:

Grupo especial: 1

Acción: 1

Dirección: 3

Estructura: 1

Datos: ninguno

Una vez obtenido el sistema real de GEADED, procedemos a realizar la respuesta del sistema:

Primero se debe añadir un saludo inicial ya que el usuario está saludando, por lo que se pondrá expresiones como *hola, buenas*, etc. Como el valor de grupo especial es 1, se colocará un valor de respuesta aleatorio del grupo 1.

El fichero de base de datos de salida está formado por las respuestas de los tres grupos especiales y las diferentes acciones. En nuestro caso, debemos buscar la referencia 4 que

corresponderá a la primera acción almacenada ya que el valor de la acción según GEADED es 1.

```
4_ser
ser_2_3_un_agente_inteligente
ser_2_3_trobox
ser_2_2_Josue
ser_3_2_eres
ser_3_3_soy
ser_0_2_Encantado
```

Figura 27. Base de datos salida referencia 4

Como explicamos anteriormente, se utilizará la dirección para completar el resto de la respuesta. En este caso, el valor de la dirección es 3, por lo que el bloque 2 se usará la y = 3, que corresponde con el primer número de la lista. El bloque 1, vendrá dado por un 0 que se utilizará para dar información extra en algunas situaciones.

Si observamos la figura 14, la conjugación a utilizar será x = 3, que corresponde a *eres y soy*, como el valor de la dirección es 3, es decir, y = 3, se utilizará *soy.*

Para el bloque 2, se utilizará x = 2 como se comentó anteriormente, y la dirección 3 que corresponde a *un agente inteligente y trobox*. En este caso, al tener dos posibles respuestas el sistema generará una de ellas de manera aleatoria provocando el efecto de naturalidad en el sistema de respuesta.

Si juntamos toda la información, obtenemos la siguiente respuesta:

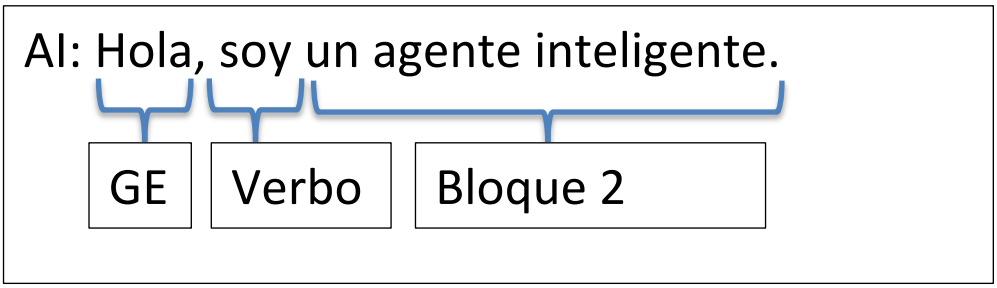


Figura 28. Respuesta del sistema

Para que el agente inteligente comprenda más palabras, se deben añadir en los respectivos ficheros de base de datos de entrada y salida las palabras con su dirección como se comentó anteriormente.

Una vez se ha comprendido los puntos anteriores, se explicará cómo está organizado. En primer lugar, los programas de Python y C++ se comunican a través de sockets. A esto se le añade el uso de ficheros para enviar datos entre los diferentes programas.

En la parte de programación de Python se encuentra un fichero llamado *listaNombres.txt* que contiene los usuarios registrados con su número de identificación. Además, por cada usuario, se encuentran cuatro ficheros formados por el nombre del usuario seguido de "Frecuencias" de la siguiente forma:

- NombreUsuario_Frecuencias.txt
- NombreUsuario_Frecuencias1.txt
- NombreUsuario_Frecuencias2.txt
- NombreUsuario_Frecuencias3.txt

Estos ficheros contienen las frecuencias características del usuario para cada uno de los audios. Serán los que acuda el código para comprobar las frecuencias obtenidas con las registradas y detectar al usuario. También, se encuentra archivos temporales en formato *mp3* que almacenan los audios recogidos por los usuarios con los que se esté comunicando el servidor.

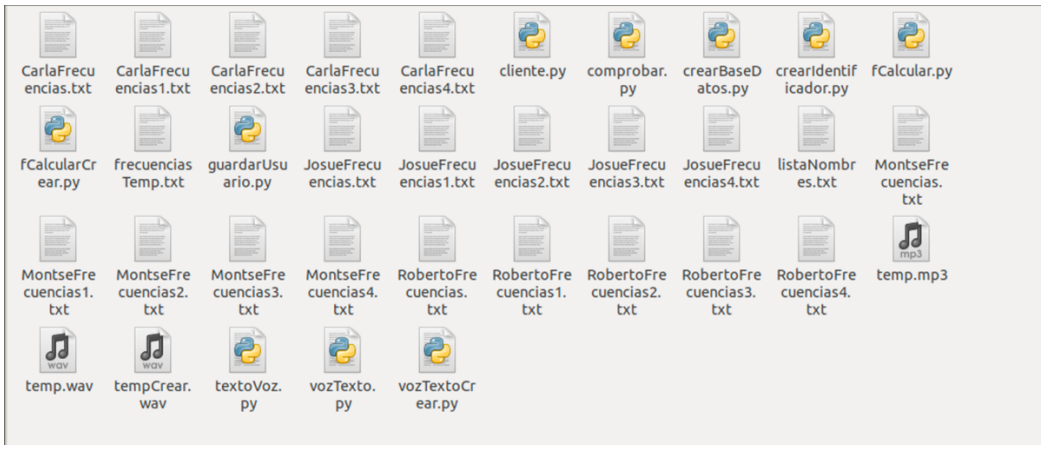


Figura 29. Archivos Python

En la parte de programación de C++, se encuentran diferentes ficheros correspondientes a los usuarios y al propio funcionamiento del sistema. El primer archivo que encontramos se denomina *baseDatosEntrada.txt* que almacena las posibles palabras claves que usará el sistema para entender la intención del usuario. Este fichero está redactado según la normativa explicada en el punto *Comprender Lenguaje Natural* como podemos ver en el ejemplo de la figura 20. El siguiente archivo denominado *codificacionGEADED.txt* es un fichero que se encargará de almacenar las características extraídas de la frase del usuario. Se recurrirá a este fichero, por parte del sistema, para generar la respuesta al usuario. Para finalizar, los últimos ficheros son los pertenecientes a las características de cada usuario y el propio sistema inteligente. Cuando el sistema genera la respuesta necesita acudir a un fichero en específico para cada usuario de tal forma que la respuesta sea siempre determinada por el usuario con el que se comunica. En primer lugar, tenemos un fichero denominado *listaNombres.txt* que se encargará de almacenar los nombres de cada usuario identificados dentro del servidor con un su número identificador. Esta lista servirá para que el sistema, a través del numero identificativo de cada usuario, acceda al nombre de ese usuario y le permita encontrar el fichero de ese usuario para generar la respuesta, por lo que, los ficheros restantes que se encuentran son los definidos para cada usuario con la siguiente denominación *nombre_usuario.txt*. Además, se tendrá un fichero adicional que corresponde al propio sistema denominado *sistema.txt*, en que se encontrará información relacionada con el agente inteligente de forma que pueda ser consultada por el mismo.

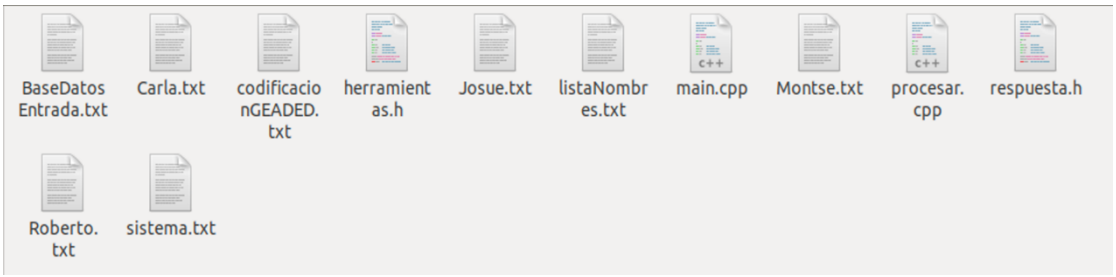


Figura 30. Archivos C++

5.4 ARDUINO

Para configurar correctamente las placas de comunicación de cada usuario se usarán las librerías predefinidas en el sistema de Arduino. En nuestro caso, se ha utilizado un módulo adicional para tener conectividad Wi-Fi en el sistema. Este módulo se conecta simplemente por los pines Tx y Rx de comunicación del arduino. El pin Tx sirve para transmitir información hacia el módulo, mientras que el pin Rx sirve para recibir información del módulo. Esta información se controla a través del comando *serial* el cuál enviará y recibirá las acciones desde el servidor. Arduino simplemente ejecutará un código de accionamiento para activar y desactivar funciones y comunicarle al servidor el estado de dicho dispositivo.

El esquema de conexión del arduino es el siguiente:

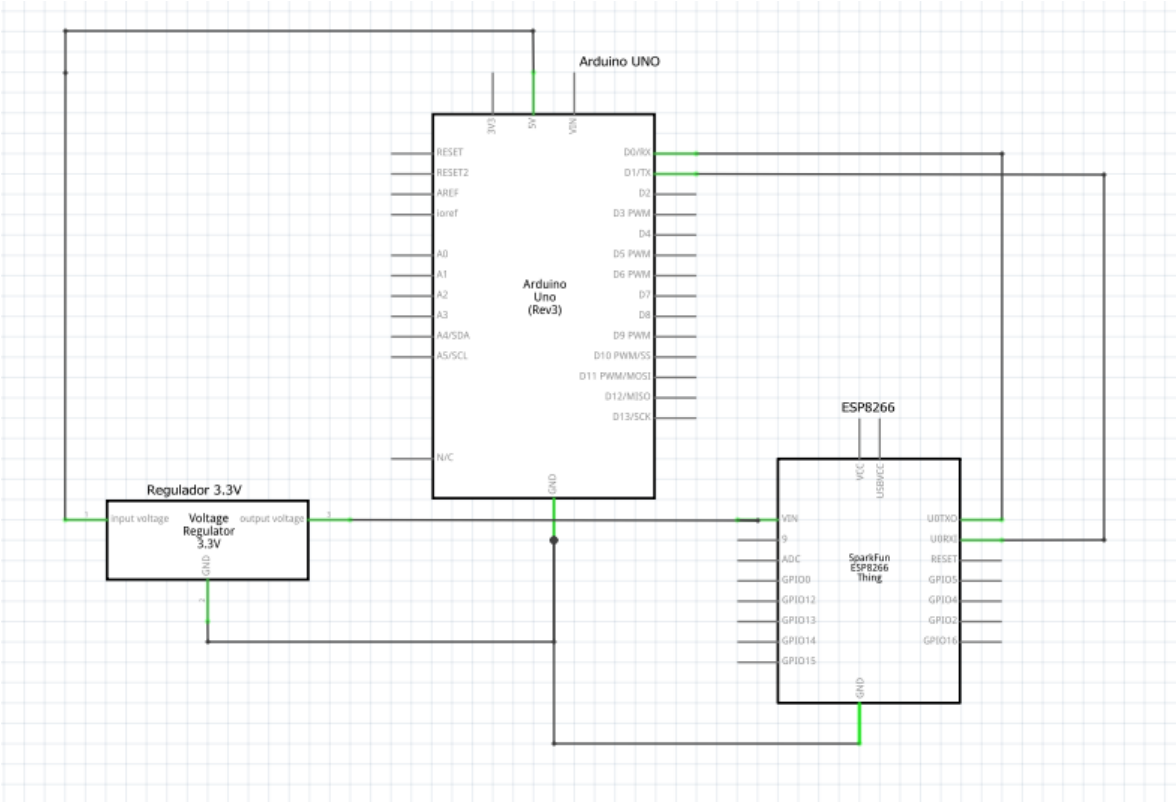


Figura 31. Esquema conexión placa de arduino

En la figura 31, se observan las conexiones detalladas anteriormente. También se usa un regulador de 3.3 V para la entrada del ESP8266, ya que utilizar la propia entrada del arduino de 3.3 V puede producir errores.

CAPÍTULO 6. RESULTADOS

Los resultado a comprobar, se harán de tres aspectos:

- Identificación de usuario.
- Procesamiento de texto.
- Acciones de la placa de arduino.

Para poder comprobar la efectividad de la propuesta elegida, se procede a realizar un estudio en el Anexo I. En este estudio, se evalúa la efectividad del código de identificación y procesamiento de mensaje por parte del sistema. Para ello, se realizarán diferentes pruebas con diferentes usuarios con características distintas. Las pruebas para la identificación de usuario serán las siguientes:

- *Prueba 1: Creación de usuario.*
  En esa prueba, se evalúa la capacidad del sistema en reconocer el usuario que está hablando, comprobar su inexistencia en la base de datos del sistema y añadirlo de forma correcta calculando sus frecuencias características.

- *Prueba 2: Mensaje predeterminado*
  Se comprueba que el usuario es identificado correctamente comunicándose con el servidor con un mensaje planificado. Esta prueba se realizará con tres mensajes diferentes siendo estos:
  - *Hola soy [nombre_Persona]*
  - *Hola soy una persona*
  - *¿Quién soy?*

- *Prueba 3: Mensaje incorrecto*
  En esta prueba, se evalúa el nivel de eficacia del sistema. El usuario dirá un mensaje con un nombre diferente al suyo para comprobar que el sistema es capaz de identificarlo.

- *Prueba 4: Mensaje aleatorio*
  El usuario se comunica con el servidor de manera aleatoria sin ningún mensaje modelo. El sistema debe ser capaz de identificar al usuario y el mensaje.

Según los resultados obtenidos de las pruebas del Anexo I, se comprueba que la eficacia del sistema es de un 83 %. Esto significa que el sistema no será capaz de identificar un usuario con un porcentaje del 100 %.

Podemos localizar en la *persona 3* que en ciertas ocasiones la identificación no es del todo correcta ya que la información recibida al sistema es insuficiente. Esto es debido a que el sonido está caracterizado por cuatro factores principales: intensidad, tono, timbre y duración.[16] Este sistema trata de identificar a una persona a través del tono y timbre debido a que se analiza la frecuencia fundamental y sus armónicos, luego, la intensidad y la duración son aspectos que no se están teniendo en cuenta y pueden afectar al resultado. Sin embargo, estos casos de error son mínimos en el que la diferencia de usuarios es pequeña y para una misma interacción del usuario con el mismo mensaje obtenemos un resultado u otro con diferencias mínimas, es decir, que podría darse el caso en que la persona 2 y la persona 3 con el mismo mensaje se detectara uno u otro. Esto significa, que existen pequeñas perturbaciones al capturar el sonido del propio usuario modificando el resultado final y que el sistema no pueda diferenciarlo con claridad.

En el ámbito del procesamiento de texto, el agente inteligente entiende al 100 % la intención del usuario. Se puede determinar que el protocolo GEADED funciona con gran eficacia ya que la respuesta generada es siempre acorde al usuario, tanto en dirección como en lo que solicita el usuario.

En las siguiente figuras, se visualizará diferentes ejemplos con respuestas por parte del sistema:

En la siguiente figura se visualizará que el programa entiende diferentes textos:

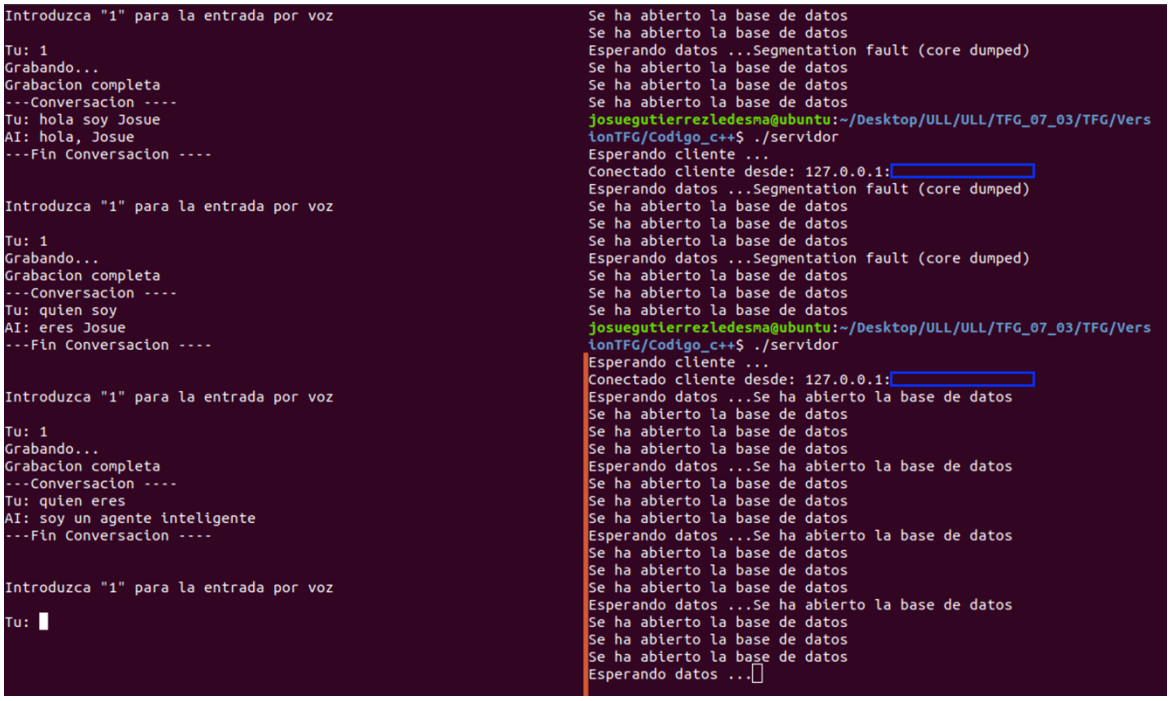


Figura 32. Resultado mismo usuario diferentes entradas

En la siguiente figura, se visualizará un mismo usuario pidiendo la misma información de manera distinta y obteniendo el mismo resultado:

Figura 33. Resultado mismo usuario mismas entradas

En la siguiente figura, se visualizará la acción de dos personas diferentes:

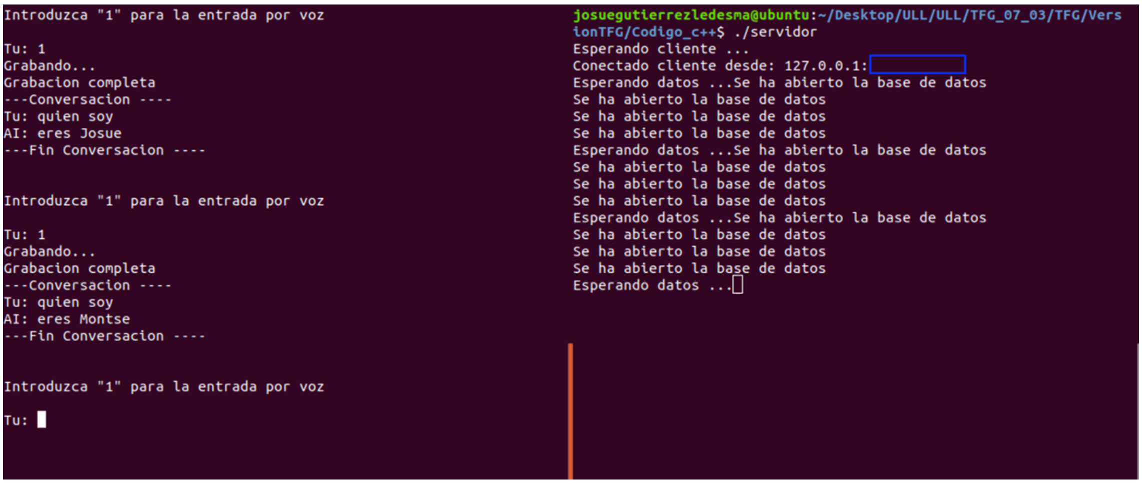


Figura 34. Resultado diferentes usuarios

En la siguiente figura se visualizará la acción de tres personas diferentes:

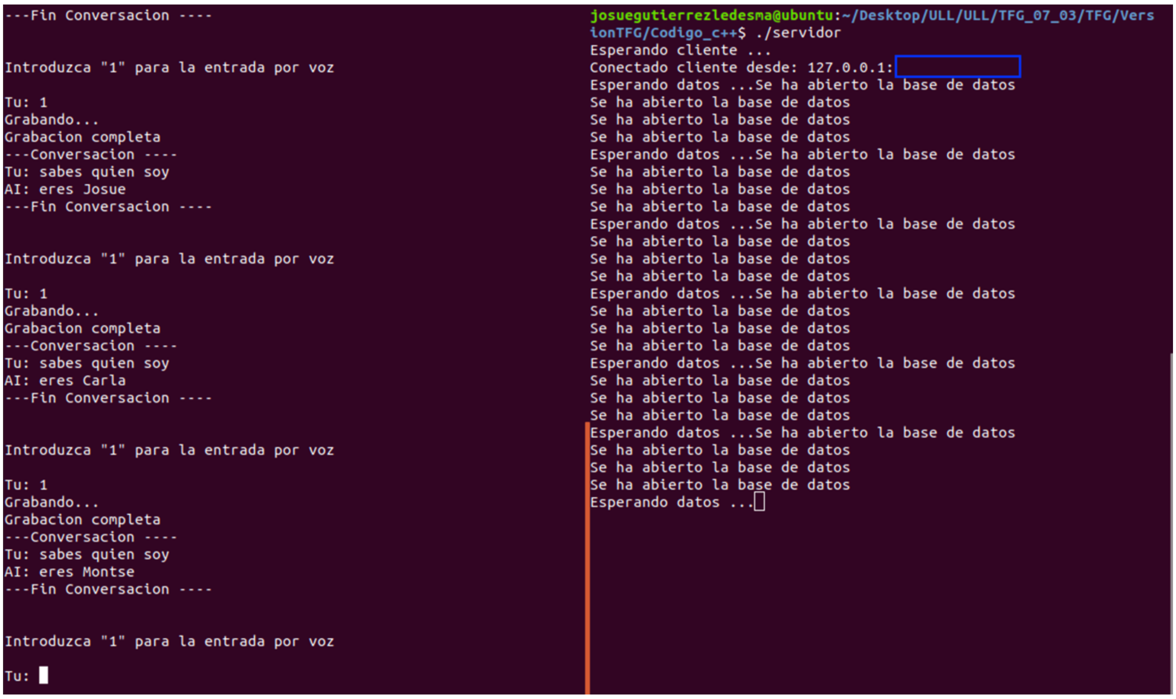


Figura 35. Resultado diferentes usuarios

En definitiva, el sistema es capaz de generar respuestas aportando aleatoriedad y, por tanto, creando un efecto de naturalidad en las respuestas. Además, es capaz de ofrecer los mismos resultados para distintas intervenciones del usuario formuladas de forma diferente y dar respuestas en función del usuario aportando la personalización para cada usuario.

En el caso de la placa de arduino, una vez el agente inteligente identifica al usuario y la acción, la placa lo ejecuta, por lo que los resultado son más que correctos.

CAPÍTULO 7. CONCLUSIONES

El objetivo principal de este TFG es implantar un agente inteligente que sea capaz de identificar y generar respuesta a las solicitudes de los usuarios. Según los resultados obtenidos, se comprueba si se ha cumplido con los objetivos previstos:

- Diseñar y desarrollar un agente inteligente capaz de reconocer e identificar al usuario que se está comunicando y dar respuestas coherentes en función de este.

- Implementar el control del sistema de voz del agente en una misma red formando una red privada IoT.

- Diseñar diferentes modos de interacción con el agente incluyendo el modo multiusuario de forma simultánea.

- Implementar el agente en un servidor de una red privada, de forma accesible para todos los usuarios y dispositivos de la red.

En este caso, los resultados indican un 83 % de acierto para la identificación y un 100 % para el procesamiento de texto y generación de respuesta por lo que se puede concluir de que el objetivo principal de este TFG se cumple con éxito.

Se observa en los objetivos que los puntos 1 y 3 se cumplen ya que identifica, reconoce, entiende y genera respuesta, tanto para un usuario como para otro totalmente distinto, en una misma comunicación, es decir, sin cerrar y volver a abrir el programa, por lo que se puede concluir que cumple con la parte de simultaneidad.

Sin embargo, los puntos 2 y 4 que hablan de una red IoT no se han cumplido del todo. Se intentó realizar pruebas de comunicación externa a través de sockets, pero las pruebas no pudieron concluir con éxito. También se intentó una alternativa denominada servidor xampp, pero por problemas de seguridad no se llegó a implementar. Se buscó alternativas a la comunicación con el servidor como bluetooth pero no se pudo implementar. Por lo que un dispositivo externo al servidor no pudo ser conectado.

Sin embargo, no afecta al éxito del TFG ya que el objetivo principal es conseguir las funciones del agente inteligente. Una vez se consigue las funciones, simplemente es un problema de comunicación de varios sistemas a través de la red.

Al igual que se comenta, la placa de arduino no pudo conectarse por socket al servidor, sin embargo, si pudo hacerse por puerto Serial por lo que se comprobó el funcionamiento de la implementación de la placa con resultados muy positivos. Al filtrar las acciones con la identificación del usuario, el sistema era capaz de decidir si realizar la acción dependiendo de quién se lo comunicara además de entender la acción que le esta indicando el usuario.

En definitiva, el sistema responde con una buena tasa de acierto y cumple con los objetivos principales. Se puede concluir que la investigación de este trabajo cumple con los mínimos exigidos para encontrar una respuesta al problema planteado.

7.1 RELACIÓN DEL TRABAJO CON LOS ESTUDIOS CURSADOS

Al ser la mayor parte del trabajo de código de programación, la asignatura de informática ha sido imprescindible. Los conocimientos desarrollados en las asignaturas de informática han hecho posible el desarrollo del agente inteligente. Además, se ha podido ampliar estos conocimientos gracias al trabajo de investigación lo que sería muy difícil de realizar si fuera un campo desconocido.

Otros de los puntos a los que hace referencia el TFG es la interacción humano – máquina, como se ha visto en el curso en asignaturas tales como informática industrial o incluso en automatización industrial avanzada. A nivel industrial, la comunicación entre una máquina y el humano es de vital importancia para evitar que se produzcan fallos en maquinaria, retrasos en la producción o situaciones de riesgo dentro de la nave industrial. Es por ello, que siempre se intenta simplificar los mecanismos para poder controlar una maquinaria, con paneles de control, luminarias identificativas, los colores asignados a cada una de las acciones, entre otros. El agente inteligente sería una interacción adicional, realmente, es un medio de comunicación entre las acciones que quiere realizar la persona frente a la maquinaria. De esta manera, diversas acciones no se tendría duda puesto que el agente se encargaría de realizarlo correctamente, o simplemente el hecho de realizar acciones con el uso de la voz facilitaría el aprendizaje del operario. Esta interacción, frente a la tradicional, trae ventajas como:

- Mayor rapidez en la toma de decisiones.
- Gestión por agente inteligente, mostrando resultados relevantes.

Como se ha visto en las asignaturas, al añadir al agente inteligente como interacción humana, se aumenta la seguridad dentro de la industria lo cual es un punto de vital importancia. El agente inteligente podría detectar diferentes zonas de riesgo y parar inmediatamente la producción o bloquear a una usuario tomar una decisión que suponga un riesgo en una zona en concreto, por lo que ya no se trata de una interacción simplemente física de un botón, sino lleva un análisis por parte de un sistema que aceptará esa acción o la desechará, aumentando así la seguridad para los trabajadores.

En otras asignaturas, se ha visto la utilización, por parte del profesorado, aplicaciones como Python o Arduino. Arduino está basado en C++ lo que ha sido bastante sencillo aprender este lenguaje de programación. También, con Python, con un simple conocimiento de programación en un lenguaje, la adaptación es mucho más rápida y sencilla, lo que no supone tampoco un problema para tener en cuenta.

7.2 TRABAJOS FUTUROS

Este TFG propone una solución en la que dentro de unos años estaremos sumergidos. IoT es una tecnología que ya se está aplicando dentro de empresas y poco a poco estará entrando dentro de otros locales e incluso en las propias viviendas.

El apartado de seguridad deberá ser uno de los más importante y prioritarios a seguir investigando y desarrollando, tanto como la identificación biométrica como la seguridad en la comunicación al enviar y recibir datos privados. Cuánto más conectada estén las cosas, más peligroso y vulnerable puede ser un sistema, lo que lo convierte a la seguridad y su privacidad en los requisitos más importantes y exigidos en cualquier red IoT.

En líneas de este trabajo de investigación, se debería tratar de investigar más características del propio sonido que pueda identificar al usuario con un mayor porcentaje de acierto a través de las redes neuronales artificiales que aportarán mayor rapidez y mejores resultados. En combinación con lo descrito en el párrafo anterior, la seguridad y la biométrica deben ir en conjunto para aportar medidas más seguras y privadas para cualquier usuario.

CAPÍTULO 8. PRESUPUESTOS

En este capítulo se desarrollará los presupuestos. Los presupuestos se deben tanto a material como al coste de las horas de trabajo. Por tanto, dividimos los costes en:

COSTES POR MATERIALES

Material	Coste por unidad	Unidades	Coste
Arduino UNO Rev 3	20,00 €	1	20,00 €
Módulo Wi-Fi Esp8266	4,20 €	1,00 €	4,20 €

COSTES POR TRABAJO:

Función	Coste por hora	Horas	Coste
Montaje del sistema	10,00 €	10	100,00 €
Conexión de la red	10,00 €	20	200,00 €
Programación	10,00 €	200	2.000,00 €
Pruebas de conexión	10,00 €	100	1.000,00 €
Configuración inicial	10,00 €	10	100,00 €

Los costes totales han sido de ***3424,30 euros.***

BIBLIOGRAFÍA

[1] «Redes Neuronales Python,» [En línea]. Available: http://www.aprendemachinelearning.com/crear-una-red-neuronal-en-python-desde-cero/.

[2] «Redes Neuronales Python,» [En línea]. Available: http://www.clubdetecnologia.net/blog/2017/python-como-construir-una-red-neuronal-simple/.

[3] Á. López González, Protocolos de Internet : diseño e implementación en sistemas Unix, Madrid: Ra-ma, 1999.

[4] A. Moreno Muñoz, Arduino: curso práctico, Madrid: Rama, 2018.

[5] «Protocolo TCP,» [En línea]. Available: https://es.ccm.net/contents/281-protocolo-tcp.

[6] «Protocolo UDP,» [En línea]. Available: https://www.ecured.cu/UDP.

[7] «Aplicacion Casa, Apple,» [En línea]. Available: https://www.apple.com/es/ios/home/.

[8] «Aplicación Casa, Apple,» [En línea]. Available: https://www.k-tuin.com/blog/casa-app-control-domestico-ios/.

[9] «DialogFlow,» [En línea]. Available: https://dialogflow.com.

[10] «Redes Neuronales IBM,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html.

[11] «Arduino,» [En línea]. Available: https://descubrearduino.com .

[12] «Raspberry Pi,» [En línea]. Available: https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/.

[13] «Raspberry Pi 3 model b plus,» [En línea]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.

[14] «Raspberry Pi y Arduino diferencias,» [En línea]. Available: https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias.

[15] «Caso Facebook,» [En línea]. Available: https://elpais.com/tecnologia/2018/09/28/actualidad/1538153776_573711.html.

[16] «Propiedades Sonido,» [En línea]. Available: http://www.ehu.eus/acustica/bachillerato/casoes/casoes.html.

[17] «Redes 5G,» [En línea]. Available: https://www.muyinteresante.es/tecnologia/articulo/que-podremos-hacer-con-la-tecnologia-5g-761487761901.

[18] «Protocolo TCP IBM,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/com.ibm.aix.networkcomm/tcpip_protocols.htm.

[19] «Diseño Diagrama UML,» [En línea]. Available: https://creately.com.

[20] «Grabar Audio Python,» [En línea]. Available: https://gist.github.com/mabdrabo/8678538.

[21] «Sockets Python,» [En línea]. Available: https://gist.github.com/skyrocknroll/8718086.

[22] «Audacity: análisis de frecuencia,» [En línea]. Available: https://audacity.es.

[23] «Industria 4.0,» [En línea]. Available: https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html.

[24] «Internet de las cosas,» [En línea]. Available: https://es.wikipedia.org/wiki/Internet_de_las_cosas.

ANEXOS

Para realizar el estudio, se procederá a realizar las cuatro pruebas siguientes:

- *Prueba 1: Creación de usuario.*
  En esa prueba, se evalúa la capacidad del sistema en reconocer el usuario que esta hablando, comprobar su inexistencia en la base de datos del sistema y añadirlo de forma correcta calculando sus frecuencias características.

- *Prueba 2: Mensaje predeterminado*
  Se comprueba que el usuario es identificado correctamente comunicándose con el servidor con un mensaje planificado. Esta prueba se realizará con tres mensajes diferentes siendo estos:
  - *Hola soy [nombre_Persona]*
  - *Hola soy una persona*
  - *¿Quién soy?*

- *Prueba 3: Mensaje incorrecto*
  En esta prueba, se evalúa el nivel de eficacia del sistema. El usuario dirá un mensaje con un nombre diferente al suyo para comprobar que el sistema es capaz de identificarlo.

- *Prueba 4: Mensaje aleatorio*
  El usuario se comunica con el servidor de manera aleatoria sin ningún mensaje modelo. El sistema debe ser capaz de identificar al usuario y el mensaje.

En este caso, constará de cuatro usuarios con perfiles diferentes para comprobar la flexibilidad del sistema. Se diferenciarán por edades y género:

- Persona 1: 15 años, femenino.
- Persona 2: 22 años, masculino.
- Persona 3: 49 años, masculino.
- Persona 4: 46 años, femenino.

Los resultados obtenidos se muestran en la siguiente figura:

ESTUDIO: IDENTIFICACIÓN DE VOZ				
PERSONA	PERSONA 1	PERSONA 2	PERSONA 3	PERSONA 4
EDAD	15	22	49	46
GENERO	FEMENINO	MASCULINO	MASCULINO	FEMENINO
PRUEBA 1	CORRECTO	CORRECTO	INCORRECTO	CORRECTO
PRUEBA 2.1	CORRECTO	CORRECTO	CORRECTO	CORRECTO
PRUEBA 2.2	CORRECTO	CORRECTO	INCORRECTO	CORRECTO
PRUEBA 2.3	CORRECTO	CORRECTO	INCORRECTO	CORRECTO
PRUEBA 3	CORRECTO	CORRECTO	INCORRECTO	CORRECTO
PRUEBA 4	CORRECTO	CORRECTO	CORRECTO	CORRECTO
PORCENTAJE ACIERTO	100%	100%	33%	100%
PORCENTAJE TOTAL	83%			

Tabla 5. Resultado pruebas de identificación

Se ha de tener en cuenta que:

- Las pruebas de la persona 1 se han realizado con la base de datos de la persona 2 para poder tener una comparación. Una vez terminadas, se elimina la base de datos de la persona 2 para realizar sus pruebas en comparación con la base de datos de la persona 1.

Las conclusiones de estos resultados se muestran a continuación:

- En la mayoría de los casos las pruebas han tenido un 100 % de éxito. El acierto global es de un 83 % lo que demuestra que es un método efectivo.
- El único caso incorrecto, ha sido la persona 3 por su gran parecido a la persona 2. Las diferencias entre ambos son mínimas, y los resultados obtenidos no representan un margen fiable, lo que da lugar al no poder diferenciar con exactitud entre estas dos personas.
- La persona 4 ha obtenido un 100 % a pesar de tener el mismo género que la persona 1, por lo que el sistema es capaz de diferenciar entre su mismo género.