Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

SISTEMA DE ASISTENCIA EN PROCESOS DE EMPRESA CON DISPOSITIVOS DE HARDWARE ABIERTO

Estudante: Álvaro Vázquez Sánchez
Dirección: Carlos Vázquez Regueiro
Juan Antonio Martín Pernas
Sandra Figueira Muñíz

A Coruña, February de 2022.

Dedicado a todo el mundo que contribuyó a que este proyecto fuese posible.

Agradecimientos

Quiero agradecer a mis familia, amigos, a todos los profesores de la Facultad de Informática de A Coruña, en especial a Carlos, el director de este proyecto. Además, agradecer la ayuda por parte de la empresa SREC y de Juan Martín, también director del proyecto.

Resumen

Este proyecto consiste en la elaboración de un prototipo sobre un dispositivo de hardware abierto que posibilite la comunicación con el servidor de Sinvad únicamente con la propia voz. Para ello se ha investigado acerca de las distintas posibilidades de hardware que permitan crear dicho prototipo. También se ha analizado que software, siempre software libre, puede desempeñar dichas funciones.

La aplicación final es capaz de funcionar únicamente con la voz, tanto la entrada como la salida. Está compuesta de cinco módulos que se encargan de, respectivamente, iniciar el proceso después de oir una palabra o frase clave (despertador), transcribir de lenguaje natural a texto de forma acertada (transcriptor), enviar esta transcripción al servidor SINVAD (comunicador) y sintetizar por voz la respuesta obtenida (sintetizador). Se ha implementado usando Raspberry OS, Java, Porcupine, Vosk y FreeTTS en una Raspberry Pi 4.

Abstract

This project consists of developing a prototype of an open hardware device that enables communication with the Sinvad server only with our's own voice. For this, the different hardware possibilities that allow the creation of this prototype will be investigated. It will also be analyzed what kind of software (GNU) can perform these functions. The application must be able to start working with just the voice, be able to transcribe natural language correctly, send this transcription and synthesize the response obtained. They are implemented using Raspberry OS, Java, Porcupine, Vosk and FreeTTS.

Palabras clave:

- Raspberry Pi
- Hardware Abierto
- Software Libre
- Java
- Sinvad
- Transcriptor
- Sintetizador

Keywords:

- Raspberry Pi
- Open Hardware
- Free Software
- Java
- Sinvad
- Transcriber
- Synthesizer

Índice general

1 Introducción **1**

1.1 Motivación . . . 1
1.2 Objetivos . . . 2
1.3 Estructura de la memoria . . . 2

2 Fundamentos Tecnológicos **3**

2.1 Hardware del proyecto . . . 3
2.1.1 Placa . . . 3
2.1.2 Micrófono y Altavoz . . . 3
2.2 Software del proyecto . . . 4
2.2.1 Raspberry OS . . . 4
2.2.2 Java . . . 4
2.2.3 Eclipse . . . 4
2.2.4 Maven . . . 4
2.2.5 Git . . . 4
2.3 Librerías del proyecto . . . 4
2.3.1 Voice2Json . . . 5
2.3.2 CMUSphinx . . . 5
2.3.3 Vosk . . . 5
2.3.4 FreeTTS . . . 5
2.3.5 Porcupine . . . 5

3 Análisis y Metodología **7**

3.1 Requisitos funcionales . . . 7
3.2 Requisitos no funcionales . . . 7
3.3 Actores . . . 8
3.4 Metodología de desarrollo . . . 8
3.4.1 Metodologías ágiles . . . 8

3.4.2 Definición de la metodología 8

4 Análisis del Hardware y Software **11**
4.1 Elección de la placa 11
4.1.1 Raspberry Pi 4B+ 11
4.1.2 Raspberry Pi 3B+ 11
4.1.3 Raspberry Pi Zero 12
4.1.4 Banana PI MPI-M4 12
4.1.5 Conclusiones 12
4.2 Elección de micrófono y altavoces 12
4.2.1 Micrófono 13
4.2.2 Altavoz 13
4.3 Librerías STT 14
4.3.1 Voice2Json 14
4.3.2 Porcupine 15
4.3.3 CMUSphinx 15
4.3.4 Vosk 16

5 Planificación y recursos **17**
5.1 Hoja de ruta y planificación temporal 17
5.2 Recursos del proyecto 18
5.2.1 Recursos humanos 18
5.2.2 Recursos materiales 18
5.2.3 Recursos software 19
5.3 Estimación de costes 19

6 Arquitectura e implementación **21**
6.1 Arquitectura general 21
6.1.1 Módulo despertador 22
6.1.2 Módulo transcriptor 23
6.1.3 Módulo coordinador 23
6.1.4 Módulo de comunicaciones 23
6.1.5 Módulo sintetizador 24
6.2 Implementación 24
6.2.1 Idea general 24
6.2.2 Paquete Login 24
6.2.3 Paquete utils 25
6.2.4 Paquete WakeUp 27

6.2.5 Paquete Transcriber . . . 28
6.2.6 Paquete Bumper . . . 29
6.2.7 Paquete apiClient . . . 30
6.2.8 Paquete Main . . . 32
6.2.9 Paquete Synthesizer . . . 33
6.2.10 Paquete Exceptions . . . 34

7 Pruebas **37**
7.1 Pruebas de micrófono . . . 37
7.1.1 Definición de la prueba . . . 37
7.1.2 Resultados y conclusiones . . . 39
7.2 Pruebas de la librería despertador . . . 39
7.2.1 Definición y ejecución de la prueba . . . 39
7.2.2 Resultados y Conclusiones . . . 40
7.3 Pruebas de la librería transcriptora . . . 40
7.3.1 Definición de la prueba . . . 40
7.3.2 Resultados y conclusiones . . . 40
7.3.3 Pruebas de la librería transcriptora CMUSphinx . . . 40
7.4 Prueba de la aplicación completa . . . 41
7.4.1 Definición de la prueba . . . 48
7.5 Resultados y conclusiones . . . 48

8 Conclusiones y líneas futuras **49**
8.1 Conclusiones . . . 49
8.2 Errores conocidos . . . 50
8.3 Líneas futuras . . . 50

A DOCUMENTACIÓN SINVADRASP **53**
A.1 PRERREQUISITOS . . . 53
A.2 CONFIGURACIONES ANTERIORES . . . 53
A.3 CONFIGURACIÓN DE SINVADRASP . . . 54
A.3.1 CONFIGURACION DE LOGIN . . . 54
A.3.2 MÓDULO TRANSCRIPTOR (CMUSPHINX) . . . 54
A.3.3 MÓDULO TRANSCRIPTOR (VOSK) . . . 54
A.3.4 MÓDULO WAKE-UP(porcupine) . . . 54
A.3.5 MÓDULO SINTETIZADOR(FREE-TTS) . . . 54
A.3.6 SONIDOS DE INICIO Y FIN DE ESCUCHA . . . 55
A.3.7 MICRÓFONOS DISPONIBLES . . . 55

A.3.8 CAMBIO ENTRE TRANSCRIPTORES 55
A.3.9 CONFIGURATIONPARAMETERSMANAGER 55

B Resultado prueba aplicación **57**

Lista de acrónimos **61**

Glosario **63**

Bibliografía **65**

Índice de figuras

4.1 Micrófono USB escogido para pruebas en entorno real 13

6.1 Representación de la arquitectura de la app. 22
6.2 Clases del paquete login 25
6.3 Clases del paquete utils 26
6.4 Diagrama de clases del paquete WakeUp 27
6.5 Diagrama de clases del paquete Transcriber 28
6.6 Diagrama de clases del paquete Bumper 30
6.7 Diagrama de estados de la clase Bumper 31
6.8 Diagrama de clases del paquete ApiClient 31
6.9 Diagrama de clases del paquete Main 32
6.10 Diagrama de clases del paquete Synthesizer 34
6.11 Diagrama de clases del paquete Exceptions 35

7.1 Resultados de las pruebas con la librería CMUSphinx en un entorno muy ruidoso. 42
7.2 Resultados de las pruebas con la librería CMUSphinx en un entorno muy ruidoso (cont). 43
7.3 Resultados de las pruebas con la librería CMUSphinx en un entorno de ruido medio. 44
7.4 Resultados de las pruebas con la librería CMUSphinx en un entorno de ruido medio (cont). 45
7.5 Resultados de las pruebas con la librería CMUSphinx en condiciones de poco ruido. 46
7.6 Resultados de las pruebas con la librería CMUSphinx en condiciones de poco ruido(cont). 47

Índice de tablas

5.1 Distribución temporal del proyecto en función de las tareas. 18
5.2 Costes de los recursos humanos en el proyecto. 19

7.1 Resultados de las pruebas de micrófono. 38
7.2 Pruebas de despertado con sensibilidad 1. 39
7.3 Pruebas de NO despertado con sensibilidad 1. 39
7.4 Pruebas de despertado con sensibilidad 0,5. 39
7.5 Pruebas de NO despertado con sensibilidad 0.5. 40
7.6 Pruebas de transcripción con librería Vosk. 41

Capítulo 1

Introducción

En este capítulo se repasa la motivación del proyecto y sus objetivos principales. También se comentan la situación en general de los altavoces inteligentes y dispositivos similares. Se finaliza el capítulo describiendo la estructura de la memoria.

1.1 Motivación

El personal de cualquier empresa pierde una parte importante de su tiempo para ser capaz de acceder a la información que necesita para llevar a cabo su trabajo o en registrar nuevos datos en un sistema de información de la empresa (hojas de cálculo, CRM, ERP, formularios web, etc.)

SINVAD es un sistema de Comprensión de Lenguaje Natural desarrollado por la empresa SREC Solutions. Su objetivo es asistir al usuario para consultar información, registrar datos y supervisar actividades desde un dispositivo iOS, Android o un navegador web con sus propias palabras, en múltiples idiomas y adaptado a los procesos de la empresa.

Una limitación del sistema es que, al ejecutarse sobre Android, iOs o navegadores web, no es posible mantener una escucha permanente por lo que es el usuario el que debe activar el asistente (pulsando en la pantalla o con el ratón).

En este proyecto, se propone la realización de un prototipo de dispositivo conectado sobre hardware abierto (p.e. Raspberry Pi) con micrófono, altavoz y conectividad, capaz de realizar llamadas a la API REST de SINVAD y gestionar los periféricos de tal manera que un usuario pueda simplemente hablar sin tener que realizar ninguna acción con las manos.

Esto facilitaría las tareas de gestión y consultas, como por ejemplo:

- ¿Cuántas unidades del producto X se han hecho hoy?
- ¿Cuántas unidades sin fallos del producto X se han hecho hoy?
- ¿Hay stock suficiente para producir X unidades del producto Y hoy?

1.2 Objetivos

El objetivo del proyecto es diseñar un sistema de asistencia automática basado en voz para facilitar tareas de consulta de información.

- Se realizará un prototipo de dispositivo sobre hardware abierto (Raspberry Pi) con micrófono, altavoz y conectividad.
- Se desarrollará una aplicación para que el usuario puede interaccionar con dicho dispositivo sin tener que usar las manos.
- Se integrará con el sistema de Comprensión de Lenguaje Natural SINVAD a través de llamadas tipo REST para crear un asistente.
- Se testeará el sistema final en condiciones reales de operación: p.e. el taller de montaje de una empresa de fabricación de mobiliario urbano.

1.3 Estructura de la memoria

A continuación, exponemos la estructura de la memoria:

1. Introducción: Se perfila en qué consiste el proyecto, por qué surge y cuál es su objetivo.
2. Fundamentos Tecnológicos: Se explican las tecnologías ya existentes de las que se hace uso para llevar a cabo el proyecto.
3. Análisis y Metodología: Se realiza un análisis de los ítems que debería cumplir la aplicación y de las metodologías que se podrían usar para desarrollarla.
4. Análisis de alternativas hardware y software: Se detallan las distintas posibilidades de harware y software entre las que se decidió para implementar el proyecto y las razones de estas.
5. Planificación y recursos: Se detalla cuál será el orden y tiempo dedicado a cada tarea y se explicaran los recursos de los que se dispone.
6. Arquitectura e implementación: Se explica conceptualmente como es el sistema además de como se la desarrollado en la práctica
7. Pruebas: Se explican la necesidad de las pruebas realizadas, los resultados y los efectos que han tenido en el proyecto.
8. Conclusiones y líneas futuras: Se hace una vista general para extraer una reflexión final del trabajo. Se comentan los errores conocidos y se analizan posibles lineas futuras.

Capítulo 2

Fundamentos Tecnológicos

En este capítulo analizaremos acerca de las tecnologías disponibles para llevar a cabo nuestro proyecto y las que finalmente hemos usado.

2.1 Hardware del proyecto

Para la realización del proyecto necesitaremos una placa, con su correspondiente alimentación, un micrófono y un altavoz.

2.1.1 Placa

Para la placa, definimos las siguientes alternativas:[1, 2, 3, 4]

1. Banana Pi BPI-M4
2. Raspberry Pi zero
3. Raspberry Pi 3 Model B
4. Raspberry Pi 4 Model B

Como podemos ver, la familia Raspberry es la favorita. Finalmente, optamos por la Raspberry Pi 4B debido a que la diferencia de coste entre la versión 3 y la 4 era mínima y la Zero para este proyecto era un poco escasa. Descartamos la Banana Pi por su precio, que era algo superior al de las otras alternativas y por la cantidad de información y proyectos que hay desarollados con la familia Raspberry.

2.1.2 Micrófono y Altavoz

Para el desarrollo del proyecto se ha empleado un micrófono USB estándar y un altavoz estándar también propio. Se ha seleccionado esta configuración ya que las posibilidades po-

drían ser muy variadas y no hay ninguna cerrada: micrófono y altavoz bluetooth, micrófono y altavoz de locutor...

Cabe destacar que el micro ha de ser USB o Bluetooth ya que el puerto minijack que posee la raspberry es únicamente de salida.

2.2 Software del proyecto

En esta sección comentaremos el software empleado en este proyecto y la tarea que desempeñamos con él.

2.2.1 Raspberry OS

Es un sistema operativo basado en unix que puede correr sobre placas Raspberry. Se ha escogido este y no otro, por ejemplo, Android, debido a las limitaciones de este a la escucha (ver apartado 1.1).

2.2.2 Java

Es un lenguaje [5] de programación orientado a objetos y que se ejecuta sobre una máquina virtual Java, lo cual favorece mucho la integración. Se ha empleado este lenguaje porque era un requisito del proyecto

2.2.3 Eclipse

Eclipse es el IDE que hemos escogido debido al conocimiento de sus funciones, que se adaptan perfectamente al proyecto y a su integración con Maven.

2.2.4 Maven

Es una herramienta muy potente y fácil de usar que empleamos para gestionar las dependencias y no tener que descargar manualmente las librerías y sus versiones.

2.2.5 Git

Es un software de control de versiones que empleamos, sobre todo, en la parte de desarrollo del proyecto.

2.3 Librerías del proyecto

En esta sección analizaremos las librerías de tipo STT y TTS que se podrían usar para llevar a cabo este proyecto.

2.3.1 Voice2Json

Voice2Json [6]es una herramienta que se puede usar como transcriptor pero debido a su funcionamiento, que es por Intents, no sirve como librería STT ya que no cumple los requisitos 3.1. También analizamos sus posibles usos para implementar otras funciones como el despertador pero su acierto era muy malo en comparación con otras librerías además de tener un entrenamiento muy tedioso, dependencias complicadas de instalar y utilidades con bastantes errores por lo que se descartó.

2.3.2 CMUSphinx

Es una librería muy completa [7] para la función STT. Este precisamente puede ser el problema, sus grandes dimensiones. En un equipo de sobremesa, corre de forma ágil aunque tarda en iniciarse pero transcribe moderadamente bien. En un equipo como la raspberry las transcripciones son nefastas y se demoran hasta los dos minutos por lo que es inviable. Esta posee una 'hermana pequeña' llamada pocketSphinx, empleada para Raspberry pero la implementación en un proyecto Java sería compleja por lo que exploramos otras opciones.

2.3.3 Vosk

Precisamente en la documentación de la librería anterior encontramos Vosk. Es una librería [8] de tipo STT muy ligera y de fácil integración con Java y, según diversos proyectos web, tiene muy buen desempeño.

Para funcionar solo necesita el modelo de lenguaje del idioma que necesitemos, en nuestro caso, castellano.

2.3.4 FreeTTS

Es una librería [9] que funciona comosintetizador de voz. Está integrado con Java además de que está escrito en este lenguaje. Su funcionamiento es muy sencillo.

2.3.5 Porcupine

En contraste con las otras librerías aquí mencionadas, Porcupine no es una librería [10] STT o TTS sino que está hecha para detectar una/s determinada/s palabras en una conversación.

La parte vital del funcionamiento es el modelo de lenguaje en el que esten esas palabras, en nuestro caso, también castellano además de la o las palabras clave que querramos usar.

Capítulo 3

Análisis y Metodología

En este capítulo discutiremos los distintos requisitos de este proyecto.

3.1 Requisitos funcionales

RF-1 La aplicación ha de entender un discurso natural y real en la lengua de comunicación.

RF-2 La aplicación deberá transcribir correctamente el discurso del usuario.

RF-3 La aplicación ha de ser capaz de sintetizar palabras entendibles para una persona.

RF-4 La aplicación ha de ser capaz de enviar sus transcripciones al servidor de sinvad así como de interpretar sus respuestas.

RF-5 La aplicación ha de comenzar a funcionar de forma activa únicamente cuando se diga la palabra clave.

3.2 Requisitos no funcionales

RNF-1 La aplicación ha de correr en hardware abierto: en este caso, correrá sobre una Raspberry pi 4B de 2GB.

RNF-2 La aplicación ha de emplear software libre.

RNF-3 La aplicación estará escrita en lenguaje Java.

RNF-4 El precio total de los componentes será en torno a un smartphone de gama media: su cometido sería sustituír a un teléfono móvil con la app de Sinvad por lo que su presupuesto deberá ser similar.

RNF-5 La aplicación será fácilmente replicable: debe poder instalarse y configurarse fácilmente en varias Raspberrys.

RNF-6 La aplicación ha de ser rápida: debe ser capaz de realizar la función del usuario en un tiempo rápido para que este la considere útil. En caso contrario, no la usaría.

RNF-7 La aplicación ha de ser intuitiva y sencilla: debería poder ser usada por un trabajador común sin una formación concreta.

RNF-8 La aplicación deberá acceder a la red únicamente para interactuar con Sinvad.

RNF-9 La configuración de harware escogida ha de caber en una caja estándar para Raspberry.

3.3 Actores

Son todos los entes que hacen uso de nuestro sistema. En nuestro caso distinguimos dos:

- **Usuario de la aplicación**: Sería cada uno de los trabajadores de la empresa, como los obreros de una fábrica o un auxiliar en una farmacia. Interactuarán con el sistema empleando únicamente su voz.
- **Administrador de la aplicación**: Es los encargado de la instalación y el mantenimiento de la aplicación en la raspberry. Puede emplear otros periféricos para interactuar con la raspberry como pantalla, ratón o teclado.

3.4 Metodología de desarrollo

En esta sección explicaremos el concepto de metodologías ágiles e introduciremos cuál hemos usado para llevar a cabo este proyecto.

3.4.1 Metodologías ágiles

Las metodologías ágiles son un marco de trabajo que nos permite adaptar la forma de trabajo a nuestras condiciones de proyecto y es especialmente útil cuando este es muy abierto y las tareas no están perfectamente definidas. Su objetivo es mostrar al cliente algo en un tiempo relativamente corto para que pueda juzgarlo, decidir si le gusta, si se quiere hacer cambios sobre este. Es una forma sencilla de implicar al cliente en el proceso de desarrollo y de facilitar cambios ya que en otras metodologías tradicionales como la de cascada, estos son muy tediosos.[11, 12, 13, 14]

3.4.2 Definición de la metodología

La metodología usada para el proyecto no tiene un nombre propio pero podríamos definirla como una pseudo-SCRUM. Se han detallado una lista de tareas o de sprints entre el

cliente, representado por Juan, que también hace de asesor experto, al igual que Carlos, director de este trabajo. Por otra parte, el alumno realiza el papel de Scrub Master y de equipo de desarrollo.

El seguimiento de estas tareas es monitorizado con reuniones periódicas, cuya frecuencia fue variando a lo largo del proyecto siendo semanales o diarias. En estas, el esquema de la reunión es el siguiente:

1. ¿Qué tarea tenías para ayer?
2. ¿Qué tarea realizaste ayer?
3. ¿Qué problemas encontraste ayer?
4. ¿Cuáles son los próximos pasos?

La clave es que el tiempo de estas reuniones sea breve y se trabaje autónomamente entre estas. En caso de no poder, se fijaba una reunión individual.

Capítulo 4

Análisis del Hardware y Software

En este capítulo profundizaremos en las distintas alternativas de hardware y software listadas (en la sección 2.1 y 2.2).

4.1 Elección de la placa

Son numerosas las alternativas en el mercado pero, debido a su enorme popularidad, la Raspberry Pi era la favorita aunque no la única. En esta sección comentaremos algunos pros y contras de cada una de ellas y como nos podrían influír en nuestros requisitos.

4.1.1 Raspberry Pi 4B+

Teniendo en cuenta las características hardware de la Raspberry Pi 4B[4], destacamos:

- Posee Bluetooth 5.0, de bajo consumo de energía, lo que disminuiría el gasto de energía en el caso de emplear micrófono o altavoz bluetooth.
- Posee varias configuraciones de RAM, lo que nos permitiría escoger en caso de que el proyecto escalase. Inicialmente, para este proyecto, 2GB de tipo DDR4 son suficientes.
- El micrófono no puede ser conectado por el puerto minijack de 3,5mm ya que en la tarjeta de audio de la raspberry es sólo de salida.
- El precio de compra es alrededor de 43€.

4.1.2 Raspberry Pi 3B+

Es la versión anterior a la raspberry pi 4. Por diversos motivos como su precio, muy similar al de la siguiente generación, su memoria RAM, que es de 1GB DDR2, un tipo bastante antiguo con unas frecuencias muy bajas comparadas con una DDR4. Por otra parte, su versión del Bluetooth es 4.2 lo que provoca que su consumo sea mayor que en la version 4B.

Por ello, pese a que en un primer momento la analizamos, no merece la pena si la comparamos con su sucesora.

4.1.3 Raspberry Pi Zero

Es la versión más sencilla de la familia Raspberry y es muy empleada en proyectos de transcripción de voz. Con relación a nuestro proyecto, destacamos lo siguiente:

- Su consumo de energía es ridículo en comparación con sus sucesores.
- Es muy económica, menos de 20€.
- Sus especificaciones con respecto a la memoria RAM y el procesador son muy justas o hasta insuficientes.
- En algunas versiones no trae bluetooth de forma nativa. En otras lo trae pero no es 5.0 por lo que el consumo de energía sería mayor
- Solo posee un conector USB-OTG para conectar nuestros periféricos como altavoz o micrófono.

4.1.4 Banana PI MPI-M4

Es una placa similar a la Raspberry Pi 4B pero con un precio en torno a los 65€. Debido a sus características similares, mayor precio, menor fama, su memoria RAM únicamente de 1GB o 2GB y que no dispone de bluetooth BLE hace que tampoco consideremos esta placa en la decisión final.

4.1.5 Conclusiones

Finalmente, hemos escogido la Raspberry Pi 4B por su precio, conectividad en general (Bluetooth BLE, USBs de sobra y miniJack de 3,5”), por su memoria RAM y capacidad y, pensando también en futuros usos. Como comentabamos, la Raspberry Pi Zero, posee una limitada memoria RAM, poco almacenamiento y un procesador con un único core por lo que sus especificaciones se quedarían muy justas para el proyecto actual.

4.2 Elección de micrófono y altavoces

Teniendo en cuenta los requisitos plasmados por el cliente, escogeremos el micrófono y los altavoces que se debería usar en las pruebas en entorno real. Para el desarrollo y las pruebas de librerías se empleará un micrófono USB genérico y un altavoz estándar con alimentación por USB y un puerto de conexión minijack de 3,5mm.

Figura 4.1: Micrófono USB escogido para pruebas en entorno real

4.2.1 Micrófono

Las posibilidades de elección son casi infinitas teniendo en cuenta todos los estándares de conexión de la raspberry y las formas de micrófonos. Por ello, remitiéndonos a los requisitos no funcionaels (en sección 3.2) escogimos únicamente un micrófono que menos visible fuese pero que estuviese fuera de la carcasa de la raspberry para mejorar la calidad del sonido. Por ello, un micrófono con esta forma nos pareció muy interesante: 4.1.

4.2.2 Altavoz

El altavoz escogido es un conjunto de dos altavoces genéricos, con alimentación por USB y conexión minijack de 3,5mm para el conexión de audio, similares a los empleados para pruebas. Se prefirió estos a unos integrados en la raspberry porque darían un mejor y más

claro sonido, facilitando la interacción con el usuario.

4.3 Librerías STT

Las librerías de transcripción son una parte vital del proyecto por lo que debemos de estudiar todas las posibilidades y asegurar fielmente que el desempeño de ellas es correcto puesto que si la transcripción falla, la aplicación no será util.

A continuación presentamos una relación de las librerías analizadas y probadas.

4.3.1 Voice2Json

Voice2Json es la primera librería que analizamos para implementar su uso. Tiene buenas opiniones, es fácil de usar, es rápida en su tarea. Su funcionamiento, como explicamos en 2.3.1, es mediante intents. Esto significa que hemos de definir previamente una lista fija de acciones que puede interpretar. Por ello, descartamos esta librería como trasncriptor ya que nuestro prototipo ha de ser capaz de transcribir lenguaje natural. Con esto, no descartamos totalmente su uso en el proyecto, simplemente le asignamos otra posible función: la de despertador. En un primer lugar, con lo visto anteriormente, podríamos definir una tarea que simplemente estuviese representada por nuestra palabra clave, de tal forma que se activase cuando la escuchase. Esto fue descartado casi al instante ya que esta librería, forzaba los intents, es decir, si solo había una acción apuntada, la de despertar, la escogía siempre por lo que el sistema estaría permanentemente despierto por lo que este módulo sería inútil. Buceando en la documentación, encontramos otra interesante función, llamada wait-wake. Esta sería, tal cual, la que estamos buscando, la de un sistema que espera hasta detectar una palabra clave y cuando la detecta, nos avisa.

Como podemos ver en la documentación [6], posee una palabra predefinida, ”hey microft”. Su desempeño no es malo pero necesitamos una palabra distinta, por lo que la siguiente idea es entrenar una palabra clave.

Para ello, se detalla como hacerlo en la implementación. En mi opinión, la instalación es compleja ya que varias dependencias ocultas, no se detallan en la instación y hay que instalarlas manualmente. Una vez instalado, podemos proceder a entrenar nuestro sistema. Para ello, como podemos ver en la documentación [15], debemos recoger grabaciones de la palabra clave y de la palabras parecidas para que el sistema aprenda a diferenciarlas, esto lo podemos hacer con la utilidad de la librería o con una externa.

Una vez hecho, procedemos al entreno inicial. Despues de cada entrenamiento, se puede probar el desempeño de la palabra en tiempo real y, para sorpresa de nadie, con un único entrenamiento su comportamiento es nefasto ya que detectaba la palabra en un 60 por ciento. Por tanto, la idea es ir incorporando nuevas grabaciones, tanto de palabras clave como de

palabras parecidas para ir reduciendo falsas activaciones y aumentando las verdaderas. Esto no pudo ser posible ya que, al intentar realizar un segundo entreno, la librería fallaba en ejecución y se cerraba abruptamente por lo que queda descartada.

4.3.2 Porcupine

Es una muy buena alternativa al despertador de Voice2Json, tiene buenas críticas y además, está integrada con Java, lo que facilitaría enormemente la realización del proyecto.

Su uso implica una conexión con un servidor Porcupine por lo que necesita una apikey pero, según podemos ver en la documentación[10], su uso es sencillo. Para su funcionamiento necesitaremos registrarnos, lo cual nos permite entrenar tres palabras clave cada treinta días y acceso con la apikey desde tres dispositivos distintos. La elección de la palabra clave es una decisión importante. Barajándolo con Juan acordamos que la palabra sería ”sinvad” o ”hola sinvad”. Esto no es posible con esta utilidad ya que sólo permite palabras en castellano por lo que decidimos que la palabra clave fuese ”hola beta”. Una vez fijada la palabra clave, procedemos a explicar que necesita para funcionar. Como dijimos antes, necesitamos un archivo que contiene el modelo de lenguaje al que pertenece la palabra que va a escuchar además de otro archivo por cada palabra clave que tenga que identificar, en nuestro caso solo una. Para obtener esta palabra, podemos conseguirlo en la consola de picovoice. Para ello tenemos que ’entrenar’ la palabra en esta consola pero esto es transparente para nosotros y muy rápido. Debemos seleccionar el idioma y el sistema operativo en el que la vamos a emplear. Es muy importante destacar que la palabra solo sirve para la versión de porcupine actual, es decir, si hoy entreno una palabra y la versión actual de porcupine es 2.1.0 esta solo se podrá usar con esa versión de la librería, no con anteriores ni siguientes. Otros parámetro relevante de porcupine es la sensibilidad. Como podemos ver, en unas primeras pruebas (en sección 7.2), una sensibilidad muy alta provoca un gran número de falsos positivos haciendo que el sentido de esta librería sea inútil.

Con un ajuste de este parámetro y repitiendo las pruebas podemos ver como el funcionamiento de la librería es muy bueno, casi brillante, y así lo incorporaremos a nuestro proyecto.

4.3.3 CMUSphinx

CmuSphinx es una gran librería de transcripción muy utilizada e integrada con Java. Su funcionamiento es bastante más complejo que otras y es menos intuitiva pero sus opiniones son buenas por lo que decidimos probarla. Para funcionar solo necesita un modelo de lenguaje, un modelo acústico y un diccionario, todo esto para el idioma que queramos. Después de una gran búsqueda encontramos un modelo de castellano.

Una vez con ese modelo, esta librería proporciona dos clases diferentes para la transcripción, una como origen un fichero y otra como origen el audio recopilado en vivo del micró-

fono. Finalmente, empleamos esta para las pruebas y la implementación de nuestro transcriptor. Uno de los grandes beneficios, pero en nuestro caso, impedimento es que la gestión de la recopilación del audio del micro es totalmente opaca para nosotros, la gestiona la librería, lo que nos hace perder control sobre que micrófono usar, por ejemplo.Su primer arranque es bastante lento, alrededor de 12 segundos pero luego la transcripción es fluída en el pc de desarrollo.

Una vez hechas las pruebas (en sección 7.3.3, podemos ver que el acierto es moderado. Sus resultados empiezan a ser erróneos conforme va creciendo el número de palabras pero muchos de estos errores se dan por entradas que no se encuentran en el diccionario por lo que, teniendo en cuenta que si hay algún pequeño fallo de transcripción, sinvad lo entendería de igual forma, decidimos emplear esta librería.

4.3.4 Vosk

Vosk es un otro transcriptor también integrado con Java. Tiene soporte para más de 20 idiomas, entre ellos el castellano que es el que necesitamos. Su funcionamiento es mucho más intuitivo que sphinx pero, por contra, la gestión de recolección del audio debemos de implementarla nosotros. Su modelo de funcionamiento es infinitamente más ligero que CMUSphinx y en, un primer contacto, más rápido y acertado. Para su funcionamiento, esta librería necesita únicamente el modelo. Además, puedes obtener los resultados parcieles de la transcripción en lugar de los finales, que es lo que hicimos en el proyecto ya que son más exactos. Así, decidimos probar la librería (ver sección 7.3.2). Como podemos ver en estas, el resultado es muy bueno, significativamente mejor que en CMUSphinx por lo que también emplearemos este transcriptor.

Capítulo 5

Planificación y recursos

En este capítulo explicaremos la planificación inicial del proyecto, los recusos disponibles y los cambios ocurridos durante el desarrollo.

5.1 Hoja de ruta y planificación temporal

En la realización del proyecto podemos distinguir las siguientes tareas:

1. Documentación y análisis de requisitos: primera toma de contacto con el proyecto. Se definen los requisitos funcionales y no funcionales del mismo.
2. Diseño del sistema: se estudia si es factible darle soporte a esos requisitos y se decide cómo hacerlo.
3. Investigación y elección del hardware: se buscan las posibles alternativas a hardware, atendiendo a los requisitos.
4. Investigación y elección de las librerías STT y TTS: búsqueda de las diversas librerías.
5. Desarrollo de un prototipo.
6. Ejecución de pruebas finales.

Como podemos ver, tenemos 6 tareas y la duración del proyecto ha sido de 10 meses y 18 días (329 días), iniciándose este el 23 de marzo y finalizando el 15 de febrero. Teniendo en cuenta estos datos, aquí tenemos una posible distribución de los tiempos en la tabla 5.1. Para el cálculo horario tomaremos una media de 2 horas por día así que el tiempo sería de unas 658 horas.

Hablando del tiempo de reunión con los tutores fijaremos que de esos 330 días, retirando días laborables y festivos nos quedarían en torno a 40 reuniones con una media de 15 minutos por reunión nos da un total de 10 horas. Si sumamos además el tiempo de reuniones concretas, estimandas en 15 horas nos da un total de 25 horas.

Tarea	Días	Horas
Tarea 1	30 días	60 horas
Tarea 2	59 días	118 horas
Tarea 3	30 días	60 horas
Tarea 4	70 días	140 horas
Tarea 5	70 días	140 horas
Tarea 6	70 días	140 horas
Total	329 días	658 horas

Tabla 5.1: Distribución temporal del proyecto en función de las tareas.

5.2 Recursos del proyecto

Podemos distinguir tres tipos en este proyecto: recursos humanos, materiales y técnicos.

5.2.1 Recursos humanos

Podemos distinguir las siguientes categorías:

- **Jefe de proyecto**: Rol desempeñado por el alumno. Es el que toma y asume las decisiones importantes del proyecto. En este caso el uso o no de una librería, de una placa, de un lenguaje de programación…
- **Analista**: Rol desempeñado por el alumno. Realiza tareas de análisis, diseño y se encargará de las pruebas de aceptación.
- **Supervisor**: Este rol es representado por Carlos. Son asesores, sirven de comunicación entre el equipo y el cliente y tienen un papel fundamental en la memoria del proyecto.
- **Consultor**: Este rol lo desempeña Juan, de la empresa Srec Solutions, especifica qué es lo que busca el cliente de forma continuada mediante reuniones periódicas, en este caso, dailies.
- **Programador**: Rol desempeñado por el alumno. Es el que desarrolla el proyecto y realiza las pruebas de unidad.

5.2.2 Recursos materiales

Aquí podemos ver un listado de los recursos materiales:

- Ordenador de desarrollo.
- Raspberry Pi 4B 2GB, cargador y tarjeta microSD.
- Micrófono USB genérico.

- Altavoz genérico con alimentación USB y conexión Minijack 3.5.

5.2.3 Recursos software

Aquí podemos ver un listado de los recursos técnicos:

- Java SE 1.8
- Apache Maven
- IDE Eclipse
- Raspberry PI OS
- Librería Vosk 0.3.33
- Librería FreeTTS 1.2.2
- Librería Porcupine 2.1.0

5.3 Estimación de costes

Para estimar los costes del proyecto tendremos que establecer el coste en personal, recursos materiales y herramientas software.

Teniendo en cuenta los tiempos detallados en la sección 5.1 y estimando un precio por hora de 20€ en el caso de las horas de dedicación por parte del alumno y de 35€ en el caso de los directores del proyecto, calculamos los costes humanos como vemos en la tabla 5.2.

Personal	**Coste por hora**	**Horas dedicadas**	**Total**
Jefe de proyecto	50 €/hora	50 horas	2500 €
Analista	40 €/hora	60 horas	2400 €
Programador	30 €/hora	448 horas	13440 €
Consultores	45 €/hora	25 horas	1125 €
Total	–	–	19.465€

Tabla 5.2: Costes de los recursos humanos en el proyecto.

Además, hemos de sumar en el apartado de recursos materiales los siguientes elementos:

- Ordenador de desarrollo: 400€. No se imputará su coste porque ya se encuentra amortizado, al tener más de 4 años.
- Raspberry Pi 4B: 43,95€
- Alimentador Raspberry: 8,95€

- Micrófono: 4,95€
- Altavoz: 4,95€
- **Total recursos materiales imputables al proyecto:** 62,80€

Con respecto a los recursos y herramientas software, no hemos consignado ninguno, porque hemos empleado software libre o software sin necesidad de licencia. Por lo que el precio total de los recursos software es de 0€.

Así, el coste total del proyecto asciende a 19.527,80€.

Capítulo 6

Arquitectura e implementación

En este capítulo hablaremos de cómo se compone la arquitectura de la app, de sus distintos módulos y de cómo se comunican entre estos.

6.1 Arquitectura general

Para la arquitectura de la aplicación se ha decidido una arquitectura modular. Esta se basa en el concepto de módulo, que es una parte del programa que tiene una tarea o tareas concretas. Así, teóricamente podríamos intercambiar un módulo que no tenga el resultado esperado por otro y para el resto de módulos este cambio no debería de ser perceptible. En nuestra aplicación se compone principalmente de cinco módulos:

- Módulo Transcriptor
- Módulo Despertador
- Módulo Sintetizador
- Módulo de Comunicaciones
- Módulo Coordinador

Además posee un dispositivo de entrada, un micrófono, uno de salida, el altavoz y dos ficheros de configuración. Los módulos mencionados anteriormente interactúan de la siguiente forma: el micrófono recoge la voz y la envía a los módulos transcriptor y despertador. El módulo despertador, cuando recibe determinada palabra, manda una señal al coordinador. Este, manda una señal al transcriptor. Por su parte, el transcriptor convierte la voz que le llega del micro a texto continuamente y, solamente cuando recibe la señal del transcriptor, envía el texto que ha obtenido al coordinador. Este módulo, cuando obtiene un texto, lo manda al módulo de comunicaciones que es el que lo envía a Sinvad y el encargado de obtener una respuesta,

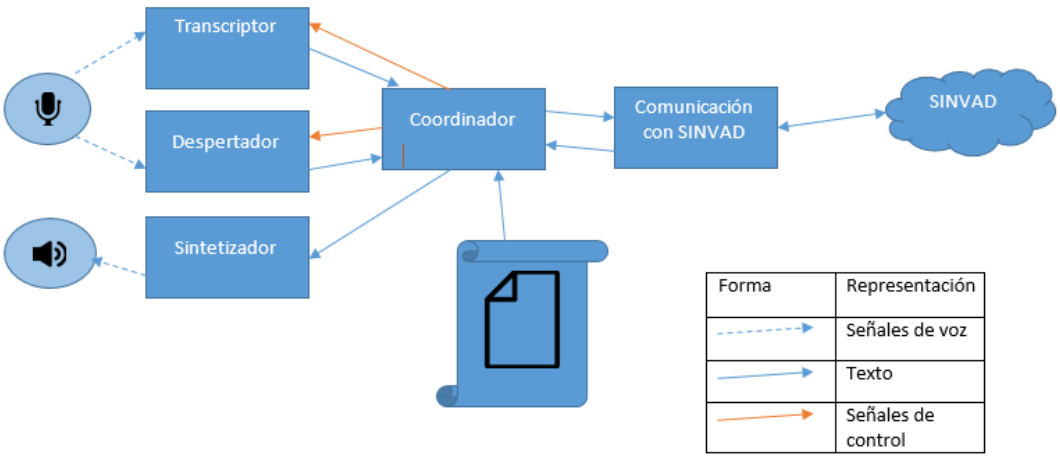

Figura 6.1: Representación de la arquitectura de la app.

también en forma de texto. Esta vuelve a ser procesada por el coordinador, que manda el texto de la respuesta al sintetizador. Este módulo, usa el altavoz para convertir a voz la respuesta y reproducirla por el altavoz. Toda esta interacción se esboza de forma más clara en la figura 6.1.

Por último, es importante recalcar que esta arquitectura permite cierto grado de paralelismo en el cual, el transcriptor y el despertador pueden trabajar al mismo tiempo, controlados por el coordinador. Esta es una idea muy potente que eliminaría retrasos en las transcripciones y que llevaremos a cabo en nuestro proyecto pero primero, hemos de definir en qué consiste cada módulo, cuáles son sus tareas, con quién interactúa y por qué.

6.1.1 Módulo despertador

Es el encargado de determinar cuándo se está interactuando con el equipo, de cuando las palabras que se están diciendo corresponden a un comando que se quiere enviar. Esta función la realiza escuchando repetidamente buscando una palabra clave, que es la que hace que el equipo se despierte. En este caso, es "hola beta" pero se podrían añadir cuantas quisiesen.

Este módulo recibe, desde el inicio hasta el fin de la ejecución, la lectura del micrófono, que es la que analiza para detectar la palabra clave. Una vez ocurre, se le comunica al módulo coordinador y este, avisa que ha recibido la comunicación y manda al primero a que vuelva a buscar si se ha dicho la palabra clave Para llevar a cabo esta tarea se hemos barajado distintas opciones pero debido a errores en otras librerías, mayor complejidad en la integración y, sobre todo, al buen desempeño de Porcupine, decidimos implementar solo esta (ver sección 4.3.2).

6.1.2 Módulo transcriptor

Este módulo es el encargado de convertir las señales del micro, es decir, las palabras que pronuncia una o más personas, en texto. El transcriptor empieza a funcionar al arranque de la aplicación, como el despertador. Este detalle se hace para evitar el delay que supone iniciar y parar el transcriptor cada vez que se diga la palabra clave ya que tarda alrededor de 4 segundos en iniciarse. Así, el transcriptor lleva a cabo su tarea desde el inicio y, cuando recibe la señal del módulo coordinador de que se ha detectado la palabra clave, empieza a almacenar lo que escucha en lugar de desecharlo. Este decide, también, que terminará el comando cuando se produzca un silencio significativo o cuando el comando dure un número elevado de segundos. Ambos parámetros pueden ser modificables. Cuando llega este momento, el transcriptor guarda el texto que ha recogido, lo encapsula en un comando y se lo manda al módulo coordinador.

Para implementar esta función se ha hecho uso de vosk (apartado 4.3.4).

6.1.3 Módulo coordinador

Este módulo es la pieza clave de la aplicación por sus dos principales tareas: la primera es recuperar los parámetros de configuración de los distintos módulos de un fichero de texto previamente definido en el código y la segunda, coordinar el funcionamiento del resto de módulos, especialmente los que trabajan de forma paralela ya que, en ciertos casos, acceden de forma concurrente a varios recursos.

Anteriormente hemos definido cómo se comunica con el despertador y el transcriptor por lo que, partiendo desde este punto, el coordinador tiene un comando ya elaborado que el usuario desea enviar a Sinvad. Así, este módulo le manda el comando al módulo de comunicaciones y recibe la respuesta que le envía Sinvad. Luego, el coordinador le envía la respuesta al sintetizador y este es el que se encarga de sintetizarla. Además, este módulo es el que se encarga de la interacción con el usuario que, aunque este no tiene muchas funciones, debe saber cuando el equipo ha detectado la palabra clave y empieza a escuchar activamente y cuando termina de hacerlo.

6.1.4 Módulo de comunicaciones

Es el punto de unión entre la aplicación y sinvad y se encarga de enviar los comandos a la apide sinvad así como de interpretar y elaborar la respuesta a estos.

Su funcionamiento es muy sencillo, este módulo recibe los comandos preparados para enviar por parte del módulo coordinador y los envía a sinvad. Luego, encapsula el texto recibido en una respuesta e interpreta el éxito o no de la operación, es decir, si el comando se ha recibido correctamente, si el comando existía ... Por último, envía esta respuesta al transcriptor.

6.1.5 Módulo sintetizador

Es el encargado de verbalizar, a través de una voz artificial, el texto escrito.

En concreto, en la aplicación, el coordinador le manda la respuesta que ha de transcribir y este, transforma el texto a voz y se lo manda al altavoz para que la reproduzca.

Para implementar esta función se hace uso de la librería FreeTTS (en apartado 2.3.4), cuyo funcionamiento es muy sencillo. Sólo necesita el archivo del idioma de voz a transcribir, en este caso hemos escogido ”kevin16”.

6.2 Implementación

Ya que hemos definido cuáles son las tareas de cada módulo, ahora vamos a explicar cómo las hemos implementado. Para ello analizaremos paquete por paquete sus clases, atributos y métodos así como otros datos de interés.

6.2.1 Idea general

La idea principal de implementación es que cada módulo sea implementado por un paquete. Además, se requiere un paralelismo entre el funcionamiento del módulo transcriptor y despertador que ha de gestionar el coordinador. Por otra parte, la idea de ejecución es que cada uno de los tres módulos anteriores corra en un thread, siendo el thread del coordinador el que realice las tareas de comunicación y sintetizado de la respuesta, debido a que no es necesario ni lógico que estos módulos trabajen en paralelo con los primeros. Para seguir, es importante aclarar una diferencia conceptual con respecto a la arquitectura. Los módulos transcriptor y despertador poseen acceso al micro pero deben capturar audio en formatos distintos puesto que no tienen la misma tarea. Por ello, en lugar de usar un único micrófono finalmente se han usado dos, uno para cada módulo. Por último, la lectura del fichero de configuración se descentralizará y será realizada por todos los módulos que lo necesiten, ya que todos poseen parámetros y es más natural y sencilla esta implementación.

6.2.2 Paquete Login

Este paquete contiene la clase Login, empleada para la identificación contra el servidor de Sinvad.

Como podemos ver, simplemente almacena tres atributos que se emplean en la identificación con Sinvad. Son de relevancia puesto que determinan que comandos tenemos disponibles para nuestra comunicación.

Figura 6.2: Clases del paquete login

6.2.3 Paquete utils

Este paquete contiene funciones generales necesarias para el resto de módulos del proyecto como, por ejemplo, la lectura de parámetros.

En la figura 6.3. podemos ver las clases que lo conforman. A continuación prodecemos a explicar cada una de ellas. En primer lugar, ConfigurationParametersManager, cuya función es leer los dos principales archivos de configuración de texto, uno que contiene información del login de sinvad y otro de parámetros de configuración de la app, ambos en formato clave-valor.En adelante,podemos identificar cuando un parámetro es leído del archivo de configuración en una clase si su nombre está escrito en mayúscula y es de tipo String.

Para seguir, la función getParameters lee un determinado archivo en función del booleano lookinglogin y almacena estos parámetros en el mapa login o parameters según corresponda.

En tercer lugar, tenemos la función getParameter, que recibe la clave de un parámetro y simplemente devuelve su valor. Para ello busca en la variable parameters.

Por último, la función getLogin, que fija el booleano lookinglogin a true, llama a getParameters para leer del fichero de login, deposita estos datos en el mapa login y luego emplea getParameter para recuperarlos y construir el login.

Otra de las clases que tenemos en este paquete es JsonClientCommandDtoConversor, que es la encargada de preparar los objetos Command para enviarlos a la api de Sinvad.

En primer lugar tenemos la función toClientDto, que se encarga de convertir un objeto Command a ClientCommandDto. Esta conversión se da porque, en determinadas circunstan-

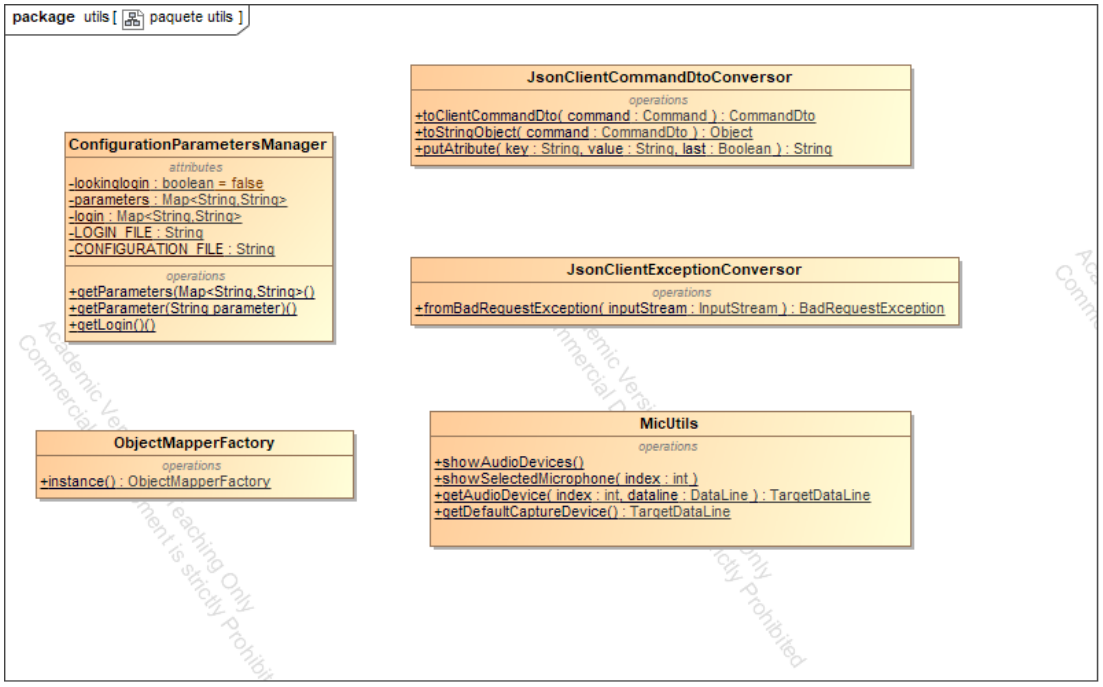

Figura 6.3: Clases del paquete utils

cias, no se quiere almacenar los mismos datos de un objeto en la capa modelo que en la de comunicación con la api. Un ejemplo de esto sería un atributo apikey, solo necesario para la comunicación con la api.

Para seguir, tendríamos toStringObject, que formatea un commandDto para que pueda ser leído por la api de Sinvad. Este formateo es en texto plano pero siguiendo las reglas de Notación de Objetos de JavaScript (JSON).

Por último tenemos putAtribute, que se encargaría de añadir un par clave valor al objeto que queremos enviar a la api.

Si queremos hablar de las respuestas, en este caso fallidas, de la interpretación de estos comandos tendríamos que hablar de la clase JsonClientExceptionConversor, que es la que se encarga de convertir las respuestas fallidas según la excepción que sea.

En este caso, como solo hay un posible error, deserializa el objeto JSON a una excepción de tipo BadRequestException, empleando también un ObjectMapper, que obtiene de la factoría.

Para terminar con este paquete, tenemos la clase MicUtils, que contiene algunos procedimientos para facilitar el manejo del micro, usadas fundamentalmente por el transcriptor y el despertador.

Estos, necesitan conocer, de todos los dispositivos de audio que tiene nuestro ordenador, cuales son aptos para la escucha, teniendo en cuenta, entre otras cosas, que el micrófono soporte el formato de audio que necesitamos para cada módulo. Esto es lo que nos proporcio-

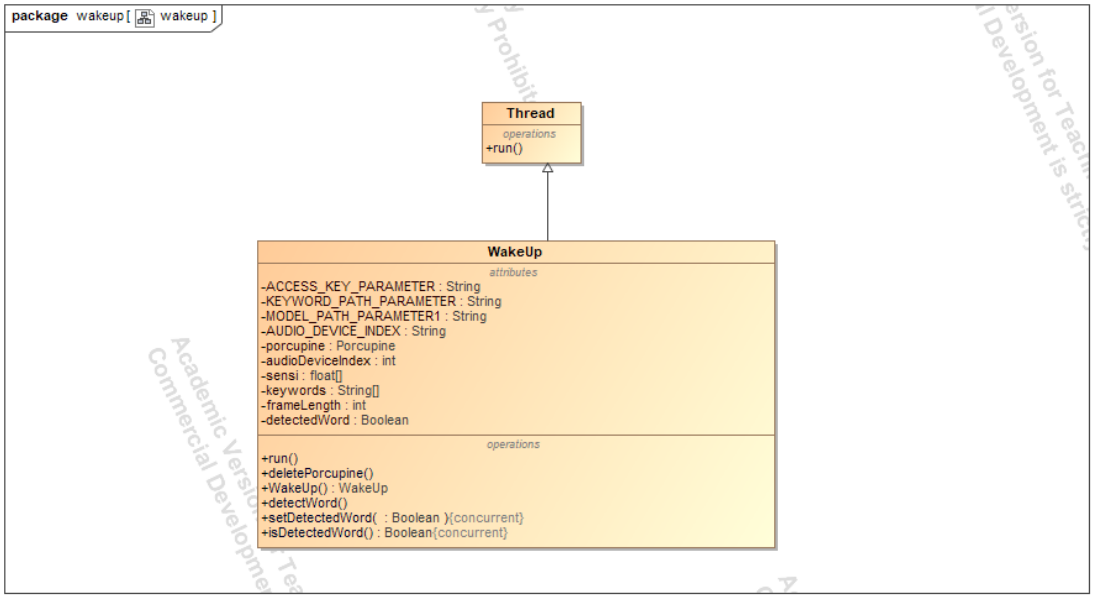

Figura 6.4: Diagrama de clases del paquete WakeUp

naría la función showAudioDevices, asignando un índice a cada micrófono válido. Por otro lado, showSelectedMicrophone mostraría las características del micrófono que se encuentra en el índice que nosotros le indiquemos.

Gracias a la siguiente función, getAudioDevice, seleccionaríamos de entre todos los dispositivos de audio disponibles en nuestro ordenador, el que queremos para escuchar, que lo indicamos con el índice. Por ello, nos devuelve un objeto TargetDataLine, que es el acceso al micrófono.

En caso de que este acceso al micro falle o de que no tengamos ningún micrófono disponible que cumpla nuestras necesidades de formato, este procedimiento delegará en getDefaulCaptureDevice, que como su propio nombre indica, obtendrá la entrada de audio por defecto del sistema. Si esta también falla, se mostrará un error por pantalla y se disparará la excepción de la clase LineUnavailableException. [16, 17]

6.2.4 Paquete WakeUp

Este paquete contiene toda la lógica del módulo despertador, cuyas funciones son realizadas como ya vimos anteriormente en la sección 6.1.1 por la librería porcupine.

Como podemos ver,está compuesto por una única clase que contiene un objeto Porcupine, nuestro despertador y sus parámetros necesarios de creación: el archivo que contiene el modelo de lenguaje, el archivo que contiene el modelo de la palabra clave, entrenado por la consola de porcupine;el índice del micro por el que se va a hacer la escucha y el apikey. Estos son leídos del archivo de configuración. Además, junto con el array de floats que contiene las

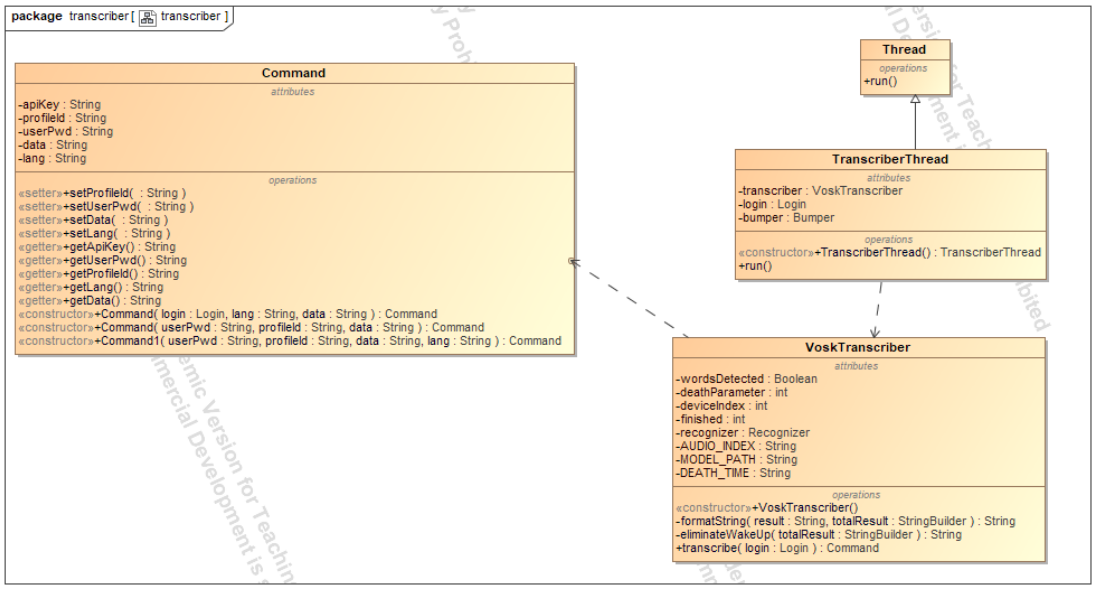

Figura 6.5: Diagrama de clases del paquete Transcriber

sensibilidades con respecto a nuestra palabra clave, en el constructor de la clase,creamos el objeto porcupine.

Con este objeto creado, y el índice del micrófono a usar en la variable audioIndexDevice, la llamada a la función detectWord haría la tarea principal de este módulo, la cual es realizar una escucha, analizarla y modificar la variable detectedWord, a la cual se accede concurrentemente utilizando la función setDetectedWord.

Por último, y vital para este módulo, observamos que esta clase es hija de Thread además de que sobreescribe el procedimiento run para conseguir el paralelismo entre los módulos. Esta función simplemente ejecutaría la función detectWord.

6.2.5 Paquete Transcriber

Este paquete contiene toda las clases necesarias para cumplir las funciones del módulo transcriptor, para las cuales hemos hecho uso , en un primer momento, de CMUSphinx (ver sección 4.3.3)y, posteriormente, de Vosk (ver sección 4.3.4).

En primer lugar, Command representa los comandos que transcribimos y enviamos a Sinvad. Por ello, esta clase posee atributos como data, que representa el texto que transcribe y lang, que indica el idioma de la transcripción. El resto de atributos son necesarios para identificación de nuestra app en la API de Sinvad.

Ahora bien, ¿quién produce estos comandos? En el diagrama de clases 6.5 podemos descubrir que es VoskTranscriber.

Esta clase es realmente la encargada de obtener audio del micrófono y transcribirlo. Pa-

ra esto, como dijimos anteriormente debe crear un objeto Recognizer, implementado en la librería de vosk. Este objeto necesita únicamente la carpeta donde se encuentra el modelo de lenguaje. Por otra parte, tenemos el índice del micrófono que usaremos para la escucha y deathParameter, que luego explicaremos su significado.

Transcribe es la función encargada de realizar la transcripción. Para ello, necesita un objeto Login (ver sección 6.2.2), que almacena los datos del usuario que elabora el comando. Se implementa como un bucle que obtiene un fragmento de audio del micrófono, que procesa el recognizer y devuelve un resultado parcial. Si ese resultado no es vacío se almacena en la transcripción final empleando las funciones de formateo de String y si lo es se incrementa uno la variable finished. Las condición de parada es sencilla: cuando se hayan detectado un número suficiente de silencios seguidos, que es lo que representa deathParameter y además se haya transcrito alguna palabra, ya que si no el transcriptor podría cortarse antes de que empezasemos a hablar debido a que cada uno de estos ciclos dura menos de un segundo. Esta condición hace que podamos adaptarnos tanto a comandos breve como a comandos más largos de hasta 30 segundos.

Por último, para implementar el paralelismo en este módulo vemos como, a diferencia del despertador, hemos creado una clase llamada TranscriberThread para realizar esta tarea. Esta posee un transcriptor de tipo VoskTranscriptor, un Login y un Bumper para comunicarse con el módulo coordinador. Como podemos ver, es hija de Thread e implementa el procedimiento run, que simplemente transcribe infinitamente y comprueba si la escucha está activada y , si lo está, lo envía al módulo main empleando el Bumper.

Como detalle de implementación es necesario comentar que, para que la clase Vosk funcione, la frecuencia de muestreo ha de ser muy alta, para que esta librería funcione.

6.2.6 Paquete Bumper

En este paquete encontraremos la clase Bumper, necesaria para la comunicación concurrente entre el trancriptor y el coordinador.

A diferencia de las comunicaciones con el despertador, estas son más complejas ya que se intercambia texto además de que se tiene que sincronizar correctamente cuando se inicia o detiene una transcripción además de gestionar una posible concurrencia. Por ello, hemos implementado esta clase.

Esta se ha implementado como una máquina de estados [18] con 3 posibles,que podemos ver en la figura 6.7 y procedemos a detallar:

1. El despertador aún no ha recibido la palabra clave por lo que la trancripción se deshecha.

2. Empieza la escucha activa porque el despertador así se lo indicó al main. Hasta que no se fije el comando con sendCommand seguiremos en este estado.

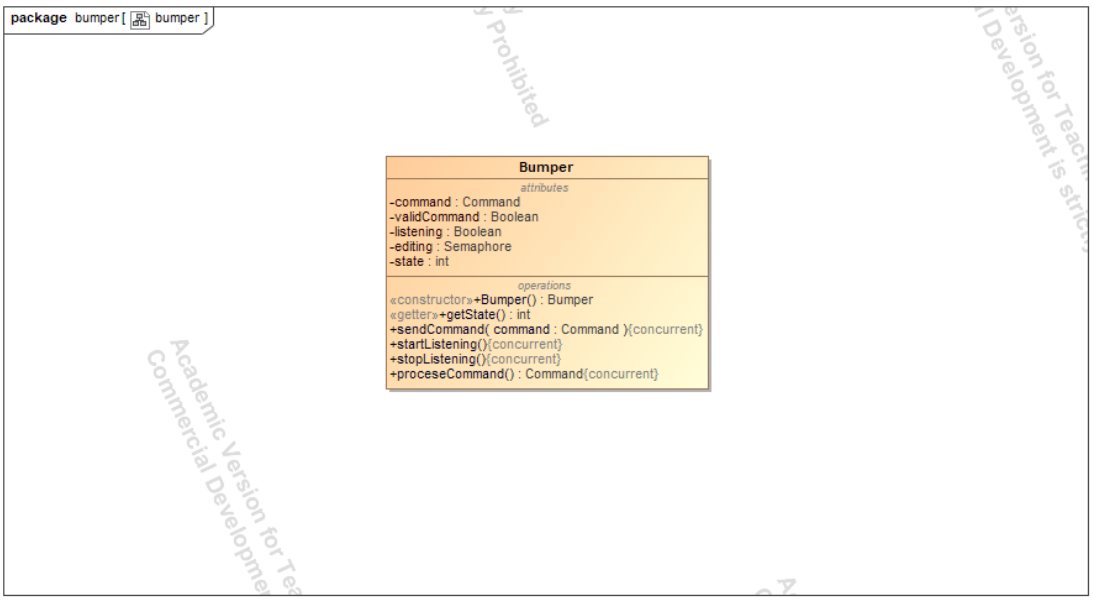

Figura 6.6: Diagrama de clases del paquete Bumper

3. La escucha ha finalizado y se ha obtenido un comando válido que el coordinador puede leer y enviar a la API de Sinvad.

   Los procedimientos sendCommand, proceseCommand y startListening son los que provocan estos cambios de estado.

   Como podemos ver, este objeto posee un semáforo inicializado a uno, por lo que funciona como un mutex. Así, todas las modificaciones que se realicen en este objeto deberán ser secuenciales.

6.2.7 Paquete apiClient

Este paquete contiene las clases necesarias para la comunicación entre nuestra aplicación y la API de Sinvad.

Para implementar estas funcionalidades haremos uso del patrón DTO y factoría.

En primer lugar, ClientCommandDto es el DTO de solo lectura de nuestro objeto Command. Este objeto facilita la comunicación ya que como se ve es un simple almacén de datos y, aunque la clase command también lo es, en un futuro podría no ser así. Como un DTO de lectura, solo contiene getters y setters para todos sus atributos, excepto para la apikey, que es fija al crear el objeto.

Para seguir, tenemos la clase que implementa el servicio, es decir, RestClientApiService. Como podemos deducir por el nombre de la clase, el servicio proporcionado por sinvad es de tipo REST.

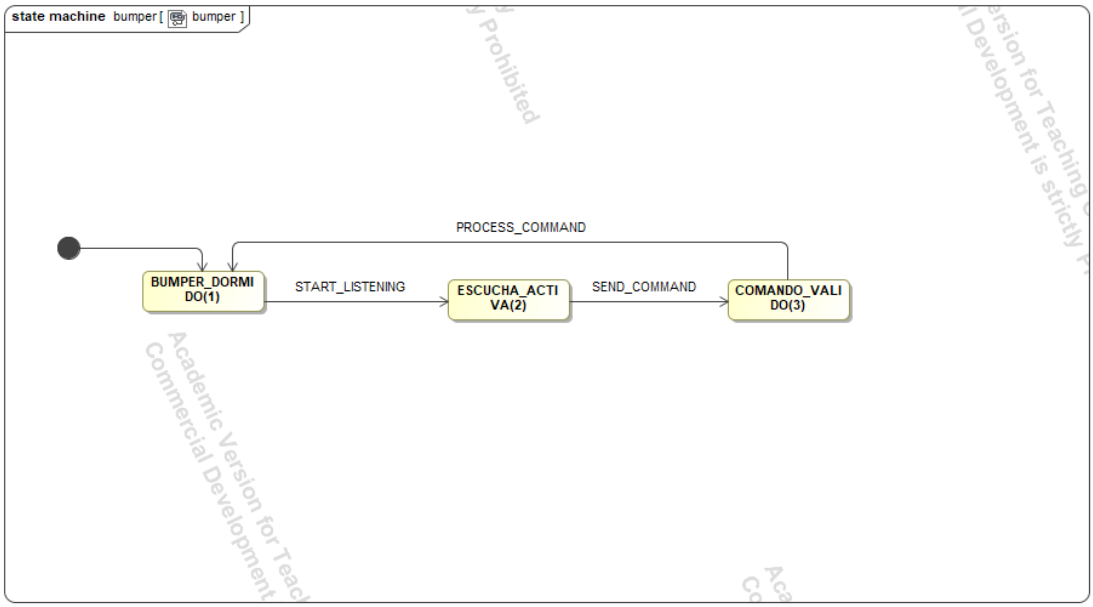

Figura 6.7: Diagrama de estados de la clase Bumper

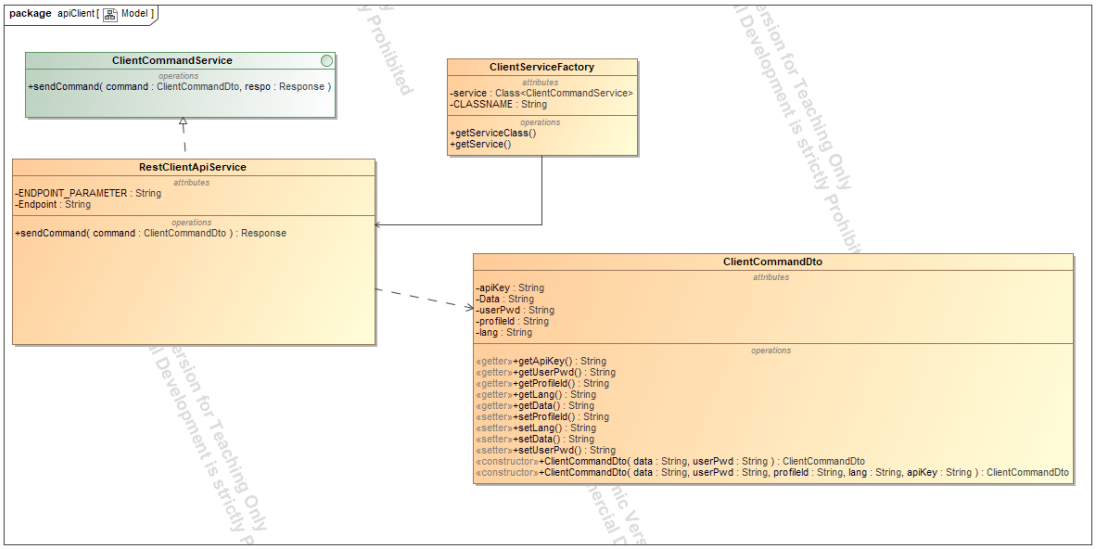

Figura 6.8: Diagrama de clases del paquete ApiClient

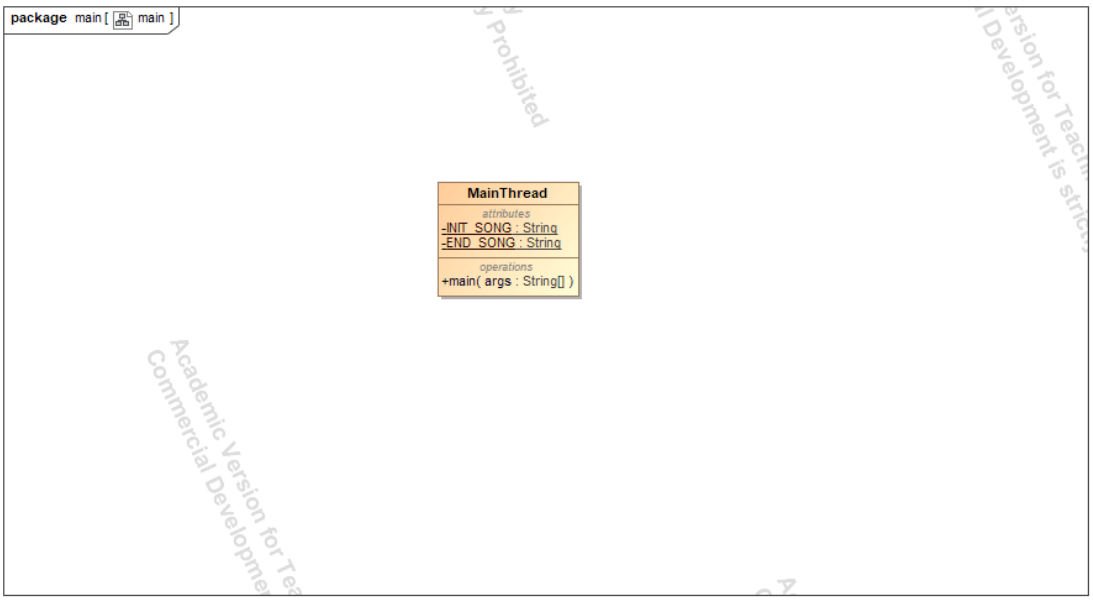

Figura 6.9: Diagrama de clases del paquete Main

Como indicamos arriba, para implementar el servicio, hemos usado el patrón factoría por lo que para obtener una instancia de este, delegamos en ClientServiceFactory, que es el que realmente lo crea. En el archivo de configuración especificamos el nombre del servicio que queremos instanciar. En caso de que hubiesemos implementado varios, podríamos escoger cual usar a través de este parámetro.

Por otro lado, tenemos la interfaz que define la comunicación con la API, ClientCommandService. Tanto el servicio ya implementado como otros hipotéticos futuros servicios deben implementar esta interfaz.

Así, para el coordinador, es transparente el tipo concreto de servicio que implementa ya que para obtener una instancia de este llama a la factoría y, como todos implementan la interfaz ClientCommandService, todos tienen esas funciones disponibles.

6.2.8 Paquete Main

Este paquete contiene la clase Main que implementa las funcionalidades del módulo Coordinador.

Como podemos ver, tiene un único procedimiento llamado Main, que es nuestra Main Class. Aquí contiene todo el código para coordinar el funcionamiento de todos los módulos así como de proporcionar feedback al usuario.

Para proporcionar esta funcionalidad había varias posibilidades. La primera, usar el sintetizador para decir algo similar a ”hola” o ”empiezo la escucha” y ”adiós” o ”fin” cuando se termine esta. La segunda, y la implementada, unos sonidos breves, que se pueden cambiar

según nuestras necesidades.

Ahora, pasamos a detallar en qué consiste la funcion main. Primero, creamos todos los objetos necesarios que representan los módulos de la app: despertador, transcriptor, sintetizador y de conexión con Sinvad además de los necesarios para la comunicación como son el bumper, un objeto Command y otro Response. Luego, creamos los objetos Clip de inicio y fin, cuya ruta obtenemos del fichero de configuración, abrimos los archivos y dejamos listo para poder reproducirlos.

Todos los objetos necesarios han sido creados por lo que entraremos en un bucle infinito en el cual fijaremos el frame de inicio de los sonidos en 0, ya que si no solo se reproducirían una vez. Luego, esperaremos de forma indefinida hasta que el wakeUp haya despertado.

En este momento, comprobaremos si se ha detectado la palabra, reproduciremos el sonido de inicio, indicaremos al módulo despertador que hemos recibido la señal y notificaremos al transcriptor que empieza la escucha activa a través del bumper, que entra en el estado 2 (en la figura 6.7 podemos ver sus posibles estados). Después de esto, el coordinador queda esperando hasta que el transcriptor produzca un comando válido.

Cuando el Transcriber acaba la transcripción e indica que el comando es válido, el coordinador obtiene el comando, emite el sonido de fin y notifica al transcriptor que lo ha recibido y se lo envía a Sinvad.

Por último, espera su respuesta, la sintetiza y volvemos al inicio del bucle.

Como podemos ver, no hemos creado un thread para cada módulo ya que el módulo coordinador puede ejecutar estas funciones secuencialmente y no tendría sentido empezar la transcripción de la respuesta antes de tenerla.

6.2.9 Paquete Synthesizer

Este paquete contiene las clases necesarias para el funcionamiento del módulo sintetizador. Para implementar esta tarea se ha hecho uso de la librería FreeTTS.

Como podemos ver en la figura 6.10, está compuesto de dos clases. La primera, Response, representa las respuestas se producen al enviar un comando a la API de Sinvad. Como podemos ver, contiene la respuesta y el comando que la generó.

Por otro lado, el encargado de verbalizar esa respuesta escrita es la clase sintetizador. Su funcionamiento es muy sencillo y mucho más opaco para el programador. En este caso, creamos el objeto Voice, seleccionando una voz mediante el fichero de parámetros.

A diferencia del transcriptor o del despertador, no necesito operar con ningún objeto SourceDataLine, que representaría un altavoz sino que invoco a la función talk y es ella la que hace este trabajo.

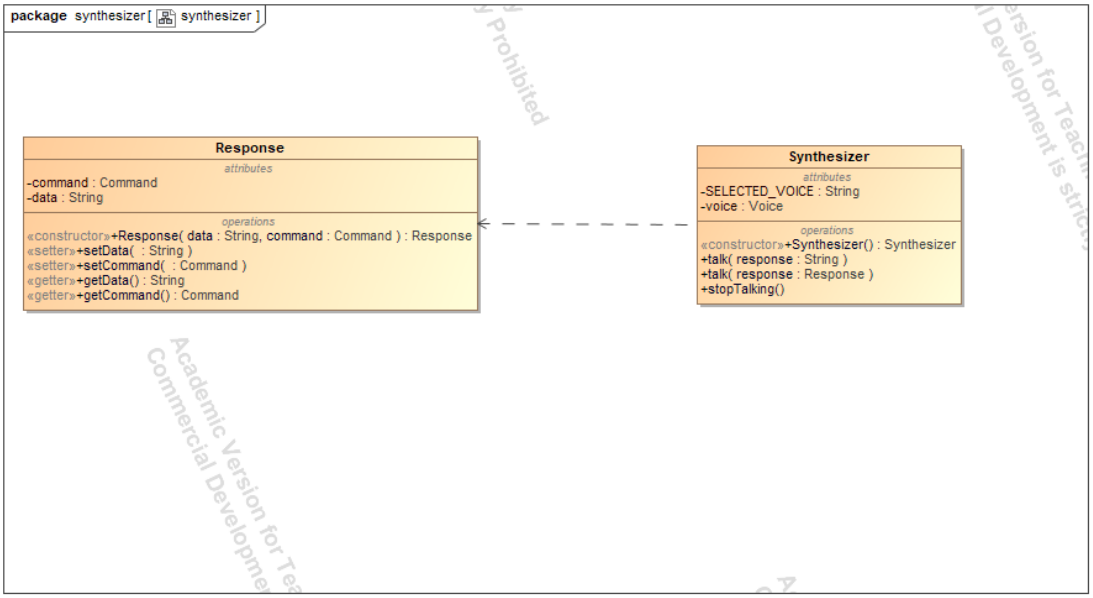

Figura 6.10: Diagrama de clases del paquete Synthesizer

6.2.10 Paquete Exceptions

Este paquete contiene únicamente las excepciones que se pueden dar por los posibles errores en la interacción del usuario con la API de Sinvad.

Este servidor, cuando recibe un comando desconocido o cuando indicamos parámetros de login incorrecto, nos devuelve un error 401 acompañado de un objeto JSON indicando los motivos.

Usando las clases de conversión de excepciones del paquete utils 6.2.3, se realiza la conversión de Objeto JSON a texto plano, se dispara la excepción BadRequestException y se envía el mensaje de error al coordinador para que se lo notifique al usuario.

Si esta conversión falla, se dispararía la segunda excepción, ParsingException, que también se convertiría y se seguiría el mismo protocolo que con la anterior.

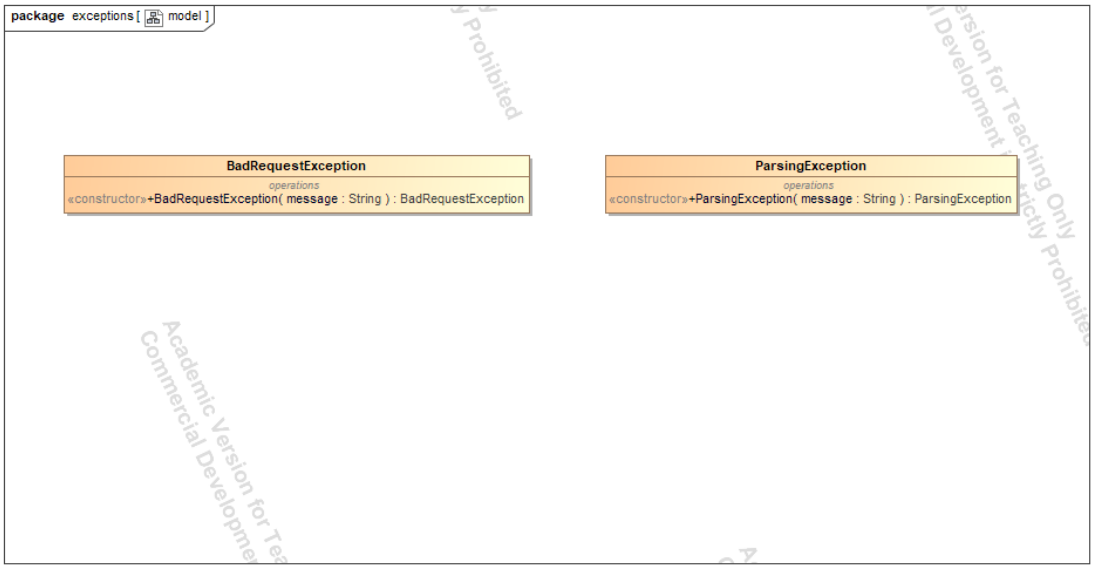

Figura 6.11: Diagrama de clases del paquete Exceptions

Capítulo 7

Pruebas

En esta sección detallaremos en qué han consistido las pruebas realizadas a los distintos módulos de nuestro sistema, los resultados obtenidos y que podemos extraer de estos.

7.1 Pruebas de micrófono

En un proyecto de este tipo, el papel del micrófono es muy importante ya que uno de baja calidad puede provocar que todo nuestro trabajo a nivel software no valga por un hardware deficiente. Por ello, realizamos esta prueba.

7.1.1 Definición de la prueba

Se conectará el micrófono a un ordenador o a un teléfono móvil y, empleando el asistente de google, se intentará que transcriba 3 frases acordadas:

- Quedan 4 unidades de xilofonos.
- ¿Cuál es la existencia de chopo disponible?
- ¿Cómo va la tarea de cortar el castaño?

Para cuantificar el acierto del sistema, podríamos utilizar múltiples estrategias como calcular la distancia entre el resultado esperado y el recibido pero emplearemos un método más simple. Este consiste en asignar un punto por cada palabra que transcribe correctamente y dividir entre el total, lo cual nos entregará una tasa de acierto. Las ubicaciones del micro y condiciones de ruido fueron variando a lo largo de la prueba. En la tabla 7.1 podemos ver esta tasa de acierto, en porcentaje.

Distancia	dB	Intento 1	Intento 2	Intento 3
0m	40	100	100	100
0m	50	100	100	86
0m	60	100	100	100
0m	70	100	86	86
0m	80	100	86	86
1m recto	40	100	100	100
1m recto	50	100	100	100
1m recto	60	86	100	100
1m recto	70	15	57	15
1m recto	80	15	0	0
1m 45°izq	40	100	100	100
1m 45°izq	50	86	100	86
1m 45°izq	60	86	71	71
1m 45°izq	70	15	15	0
1m 45°izq	80	0	0	0
1m 45°dcha	40	100	100	100
1m 45°dcha	50	86	71	100
1m 45°dcha	60	86	71	71
1m 45°dcha	70	15	29	15
1m 45°dcha	80	0	0	0
2m recto	40	100	86	86
2m recto	50	86	86	86
2m recto	60	86	71	71
2m recto	70	14	14	0
2m recto	80	0	0	0
2m 45°izda	40	86	86	71
2m 45°izda	50	86	71	50
2m 45°izda	60	42	42	71
2m 45°izda	70	0	0	15
2m 45°izda	80	0	0	0
2m 45°dcha	40	71	86	100
2m 45°dcha	50	71	71	57
2m 45°dcha	60	0	42	71
2m 45°dcha	70	0	0	0
2m 45°dcha	80	0	0	0
1,5m y obstáculo a 1m	40	86	86	71
1,5m y obstáculo a 1m	50	86	71	86
1,5m y obstáculo a 1m	60	71	86	71
1,5m y obstáculo a 1m	70	14	0	0
1,5m y obstáculo a 1m	80	0	0	0

Tabla 7.1: Resultados de las pruebas de micrófono.

7.1.2 Resultados y conclusiones

Como podemos ver en la tabla, el sistema tiene una tasa de acierto muy buena cuando estamos cerca, incluso en condiciones de mucho ruido. A medida que nos alejamos y el ambiente se hace más ruidoso, cae esta tasa de acierto hasta el 0, lo cual es lógico. Atendiendo a la tasa de acierto global, este micrófono es apto para nuestro sistema.

7.2 Pruebas de la librería despertador

Como especificamos anteriormente, nuestro sistema posee un módulo despertador6.1.1, activado por una palabra clave. Por tanto, debemos de comprobar cómo se comporta ante esta, ante palabras similares y si esta librería cumple los requisitos. Es importante, tanto que se active cuando se dice la palabra como que no lo haga cuando no se dice esta.

7.2.1 Definición y ejecución de la prueba

Se llevará un monólogo con el micrófono a menos de 50cm del locutor y en condiciones de ruido normales(50dB) y este dirá, en primer lugar, la palabra clave 20 veces entremezclado con el discurso, lo repetirá y comprobará si el módulo despierta. Luego, en una segunda fase, dirá palabras similares a la palabra clave, lo repetirá y comprobará que el módulo no se despierta. Esta prueba nos ayudará a determinar la sensibilidad a la palabra clave.

Nºintento	Despiertos	Tasa Acierto	Dormidos	Tasa fallo
Intento 1	19	95	1	5
Intento 2	18	90	2	10

Tabla 7.2: Pruebas de despertado con sensibilidad 1.

Nºintento	Despiertos	Tasa Acierto	Dormidos	Tasa fallo
Intento 1	14	70	6	30
Intento 2	13	65	7	35

Tabla 7.3: Pruebas de NO despertado con sensibilidad 1.

Como podemos ver, la tasa de fallo en el caso de palabras similares es bastante alta. Probaremos ajustando la sensibilidad y repitiendo las pruebas

Nºintento	Despiertos	Tasa Acierto	Dormidos	Tasa fallo
Intento 1	18	90	2	10
Intento 2	18	90	2	10

Tabla 7.4: Pruebas de despertado con sensibilidad 0,5.

NºIntento	Despiertos	Tasa Acierto	Dormidos	Tasa fallo
Intento 1	0	100	20	0
Intento 2	1	95	19	5

Tabla 7.5: Pruebas de NO despertado con sensibilidad 0.5.

7.2.2 Resultados y Conclusiones

Como podemos ver en las tablas 7.2, 7.3, 7.4 y 7.5, estas pruebas nos han permitido ajustar la sensibilidad a la palabra para mejorar el funcionamiento de este módulo. Así, reduciendo su sensibilidad hemos permitido reducir la tasa de fallo en la segunda prueba hasta, casi, erradicarla, mientras la tasa de acierto en la primera apenas se ve afectada.

7.3 Pruebas de la librería transcriptora

El modelo transcriptor es de especial relevancia en este proyecto, ya que una mala elección puede hacer que nuestra aplicación no entienda correctamente nuestro discurso y provocar que sea lenta e ineficiente, lo cual iría contra el proyecto. Por eso, probaremos el funcionamiento de la librería de transcripción Vosk y decidiremos si es apta para el proyecto.

7.3.1 Definición de la prueba

Para evaluar el acierto de la librería se ha diseñado la siguiente prueba. Se seleccionará una lista de frases y se transcribirán. Se comparará el resulta esperado con el recibido. Estas pruebas se ejecutarán a 50cm del micrófono y con un nivel de ruido de alrededor de 50dB.

7.3.2 Resultados y conclusiones

Para comprender la tabla 7.6 debemos saber que si en la celda consta una ”v” significa que la palabra esperada y la transcrita coinciden. En caso de que aparezcan puntos suspensivos,estos indican que las palabras transcritas y las esperadas coinciden hasta ese punto o a partir de ese punto. Como podemos verm el desempeño de la librería es muy bueno, en estas condiciones los errores son mínimos, por lo que los pocos que haya en la transcripción podrían ser subsanados por Sinvad fácilmente.

7.3.3 Pruebas de la librería transcriptora CMUSphinx

Como dijimos anteriormente, este módulo es de especial relevancia por lo que se requiere un buen desempeño para hacer la aplicación útil.

Para evaluar el trabajo de esta librería se han diseñado unas exhaustivas pruebas. Estas consisten en definir 100 frases de distintas longitudes para que el transcriptor las interprete.

Palabra	Intento 1	Intento 2	Intento 3
Sí	v	v	v
No	v	v	v
Muy Bien	v	v	v
Muy Mal	v	v	v
Donde esta	v	v	v
Hola Beta	v	v	Hola Betas
Habitación cinco	v	v	v
Cuando llega	v	v	v
No hay herramientas necesarias	... necesaria	v	v
El ordenador esta estropeado	v	v	El ordenador está ..
El personal de limpieza de la planta	v	...plata	v
Cuales de las plantas anteriores no han sido limpiadas	v	Cuál es ...	v
La segunda habitación	v	v	v
El total es de dos euros	v	v	v

Tabla 7.6: Pruebas de transcripción con librería Vosk.

Esta prueba se hará bajo tres condiciones distintas de ruido: con ruido alto, en torno a 80 dB, con ruido medio, como podría haber en una oficina (55dB) y con ruido bajo, menos de 40db.

En estas tablas recogemos los resultados de las pruebas. Como podemos ver, son muy densos por lo que dividimos cada prueba en dos tablas.

La representación de la información de las tablas es el mismo que en las pruebas de la otra librería:

- La letra ”v” representa que el resultado esperado
- Los puntos suspensivos representan que la transcripción es la esperada hasta o desde ese punto.

A continuación, podemos ver las pruebas en condiciones de ruido alto: 7.1 y 7.2. En estas tablas podemos ver las de ruido medio 7.3 y 7.4. Por último, en estas podemos ver los resultados de la prueba con ruido bajo 7.5 y 7.6.

En conclusión,el rendimiento es aceptable, teniendo en cuenta que Sinvad podrá corregir esos errores en la transcripción, por lo que decidimos emplear esta librería.

7.4 Prueba de la aplicación completa

Una vez ha concluído el desarrollo de todas las funcionalidades de la aplicación se han de realizar varias pruebas para comprobar que el funcionamiento de todos los módulos en conjunto es el adecuado.

Núm	FRASE	INTENTO 1	INTENTO 2	INTENTO 3
1	Sí	v	v	v
2	No	v	v	v
3	Menos	v	v	De ellos
4	Para	v	v	v
5	Más	v	v	v
6	Muy Bien	Fue el	v	v
7	Muy Mal	v	v	Hola
8	Sigue Probando	Si acabando	v	Si acabando
9	Donde esta	v	v	Reja
10	Cuando llega	v	v	cuando era
11	La habitación primera	v	v	La habitacion y eh
12	La casa azul	v	v	La taza apure
13	La segunda habitación	v	v	v
14	La primera recepción	La primera deteccion	La primera detencion	La primera deteccion
15	La última piscina	v	v	v
16	La cuarta planta del hotel	v	v	v
17	El cuarto de la plancha	El cuarto de la planta	El cuarto de la planta	El cuarto de la planta
18	La piscina de la planta	v	v	la cristina de la planta
19	El waterpolo de mi casa	pero tengo que ir alla	pero tengo que ir alla	pero tengo que ir alla
20	Las mesas del primer recibidor	las metas del primer recibido	las mesas del primer recibido	las mesas del primer recibido
21	La bolsa de compra	v	v	v
22	Mi mochila de clase	v	v	v
23	Esta libre el baño	la libre el baño	La libre el baño	Libre el baño
24	La casa del vecino	v	v	La caza el vecino
25	El cuarto del vecino	v	v	v
26	La sintaxis del idioma	la sin taxis del idioma	la sin taxis del idioma	la cinta seis del idioma
27	La llave de casa	v	v	la llave de paz
28	La llave del porche	la llave del por che	la llave del por che	la llave del por che
29	El primer día aquí	v	v	v
30	El segundo día aquí	v	el segundo dia a aquí	el segundo dia si
31	El total de habitaciones	v	v	el total eh habitaciones
32	Dos días lluviosos aquí	dos dias lluviosas aqui	dos dias lluviosas aqui	dias lluviosas aqui
33	Los cascos no funcionan	los cascos de julian	los cascos romper	los cascos no murió
34	La fregona no limpia	la pregona o limpia	la pregona o limpia	la pregona o limpia
35	La habitacion esta sucia	v	v	v
36	La habitacion esta limpia	v	v	v
37	No hay herramientas necesarias	v	v	v
38	El ordenador esta estropeado	El ordenador esta estupendo	El ordenador esta estupendo	El ordenador esta estupendo
39	El choque fue grande	v	choque fue grande	el choque fue el hambre
40	Reordena la habitacion ya	me ordena la habitacion alla	me ordena la habitacion alla	readaptacion vida
41	No quedan helados arriba	v	v	no quedan heladas ari
42	El personal de limpieza de la planta	v	v	persona de limpieza de planta
43	Las habitaciones de la segunda planta	v	v	las habitaciones de la semana
44	El total es de un euro	v	tales de un euro	v
45	El total es de dos euros	el total es de dos seguros	el total es de dos seguros	el total es de dos seguros
46	Donde se encuentra la tercera planta	v	v	v
47	Donde se encuentra la cuarta planta	v	v	v
48	Donde se encuentra la primera planta	v	v	v
49	El caballo y sus tres herraduras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras
50	No queda chocolate blanco en esta planta	v	v	v
51	Hay zapatos pequeños en este lugar	v	v	v
52	Mi cama no esta en la habitación	mi camara esta en la habitacion	mi camara esta en la habitacion	mi cama no esta en la habitacion

Figura 7.1: Resultados de las pruebas con la librería CMUSphinx en un entorno muy ruidoso.

53	Las casas de la urbanización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca
54	Donde se encuentra el centro del hotel	dónde se encuentra el centro lo tela	v	v
55	La musica del hall no es agradable	la música del joven lois árabe	la música del joven lois árabe	la música del joven lois árabe
56	El portátil no es lo suficientemente potente	v	el ordenador no es lo solicite mente potente	v
57	El trofeo del campeonato esta expuesto	v	v	v
58	No hay existencias de madera de chopo	no hay existencia es de embarazo oh	pero hay existencia es de madera de chocó	no hay existencia es de madera de chocó
59	Alumbra el recibidor de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta
60	Donde esta el gerente del hotel	v	v	v
61	Cuando llega el gerente del hotel	v	v	v
62	Cual de las habitaciones de la planta esta limpia	v	v	v
63	Cuantas habitaciones hay en la planta de arriba	v	v	v
64	Que haces para jugar tan bien al waterpolo	que antes para jugar también agua truco lo	que antes para jugar también agua truco lo	qué antes para jugar también agua truco lo
65	Ese teléfono android no funciona tan bien como el otro	ese teléfono andrade logra también como el otro	ese teléfono andrade logra también como el otro	ese teléfono andrade logra también como el otro
66	El aspirador no se encuentra en la primera planta	Él aspira dora no se encuentra en la primera planta	Él aspira dora no se encuentra en la primera planta	Él aspira dora no se encuentra en la primera planta
67	La habitación trescientos catorce aun no ha sido limpiada	la habitación trescientos catorce echa humo ha sido limpiada	la habitación trescientos catorce echa humo ha sido limpiada	la habitación trescientos catorce a humo así da miedo
68	El responsable de la planta uno no llega hasta mañana	v	v	v
69	Cuando llega el gerente del hotel a la ciudad	cuando llega el gerente del hotel a la tienda	v	v
70	Busco recepcionista y gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel
71	Tres millones doscientos cuarenta y cuatro millones de euros	v	v	v
72	Sintonizar el programa de la primera en todas las televisiones	v	v	v
73	Limpiar los uniformes de todo el personal del hotel	v	v	v
74	Instalar los programas del pack universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador
75	Cuantos días quedan para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes
76	Faltan treinta y cuatro días para el siguiente pedido	v	v	v
77	Falta personal para cubrir los puestos el lunes	v	v	v
78	La sala de espectáculos está vacía los domingos	la sala de espectáculos estaba hacia los domingos	v	v
79	Las dioptrías de las gafas son de dos y media	las dio cria de las cartas son de dos y media	las dio cria de las gracias son de dos y media	las dio cria de las cartas son de dos y media
80	Cantar y bailar es imprescindible para conseguir el puesto	v	v	v
81	La decoración del recibidor se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros
82	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros
83	El hotel se encuentra a solo cinco kilometros del centro	el hotel se encuentra solo a ti no pierdo	el hotel se encuentra solo a ti no pierdo	v
84	En este establecimiento se habla castellano y gallego	... se habla castellanos y la diego	... se habla castellanos y la diego	...se habla castellanos y la diego
85	Subir la temperatura de la habitación a veinte grados	...a veinte lados	... haga integrados	llamada personal de la recepcion de la platna
86	Llamar al personal de la recepción de la planta	... de detección de la planta	... de detección de la planta	v
87	Pintar la habitación cien de color negro azabache	...bien de color negro acabarse	...bien de color negro a la base	... de negro a tope
88	Cuales de las plantas anteriores no han sido limpiadas	cuáles de las plantas anteriores no han sido olimpiadas	cuáles de las plantas anteriores no han sido olimpiadas	no han sido olimpiadas
89	Las habitaciones de la primera planta no han sido limpiadas por Álvaro	...no he sido limpias hola Álvaro	... no he sido olimpiadas por Álvaro	... no han sido olimpiadas por Álvaro
90	El responsable de la segunda planta no llega hasta las dos	v	v	v
91	No hay existencias de sábanas para las camas de la planta	pero hay existencia es de sábanas...	lobaina existencia ste sábanas...	como lo hiciste decías para las camaras de la planta
92	A cuanto asciende la suma de todos los salarios del personal	v	v	v
93	Buscar en youtube musica de ambiente y reproducirla en el dispositivo	buscaré a otra chica era teresa periférico	busca y mi auto posada irás a elegir	busca y mi auto posada irás a elegir
94	No hay electricidad, agua o gas en ninguna parte del hotel	pero hay electricidad a paga bas en una de él	de ser atacadas	de ser atacadas ...
95	El quinto hotel de la cadena será abierto esta semana en Zamora	... bezamora	... bezamora	... bezamora
96	El aire acondicionado y la televisión de esta planta están en desuso	v	v	v
97	Dos millones cuatrocientos cuarenta y seis doscientos veintidós millones de	...eu nos costo	... eu nos costo	...eu nos costo

Figura 7.2: Resultados de las pruebas con la librería CMUSphinx en un entorno muy ruidoso (cont).

Núm	FRASE	INTENTO 1	INTENTO 2	INTENTO 3
1	Sí	v	v	v
2	No	v	v	v
3	Menos	v	v	De ellos
4	Para	v	v	v
5	Más	v	v	v
6	Muy Bien	Fue el	v	v
7	Muy Mal	v	v	Hola
8	Sigue Probando	Si acabando	v	Si acabando
9	Donde esta	v	v	Reja
10	Cuando llega	v	v	cuando era
11	La habitación primera	v	v	La habitacion y eh
12	La casa azul	v	v	La taza apure
13	La segunda habitación	v	v	v
14	La primera recepción	La primera deteccion	La primera deteccion	La primera deteccion
15	La última piscina	v	v	v
16	La cuarta planta del hotel	v	v	v
17	El cuarto de la plancha	El cuarto de la planta	El cuarto de la planta	El cuarto de la planta
18	La piscina de la planta	v	v	la cristina de la planta
19	El waterpolo de mi casa	pero tengo que ir alla	pero tengo que ir alla	pero tengo que ir alla
20	Las mesas del primer recibidor	las metas del primer recibido	las mesas del primer recibido	las mesas del primer recibido
21	La bolsa de compra	v	v	v
22	Mi mochila de clase	v	v	v
23	Esta libre el baño	la libre el baño	La libre el baño	Libre el baño
24	La casa del vecino	v	v	La caza el vecino
25	El cuarto del vecino	v	v	v
26	La sintaxis del idioma	la sin taxis del idioma	la sin taxis del idioma	la cinta seis del idioma
27	La llave de casa	v	v	la llave de paz
28	La llave del porche	la llave del por che	la llave del por che	la llave del por che
29	El primer dia aqui	v	v	v
30	El segundo dia aqui	v	el segundo dia a aqui	el segundo dia si
31	El total de habitaciones	v	v	el total eh habitaciones
32	Dos dias lluviosos aqui	dos dias lluviosas aqui	dos dias lluviosas aqui	dias lluviosas aqui
33	Los cascos no funcionan	los cascos de julian	los cascos romper	los cascos no murió
34	La fregona no limpia	la pregona no limpia	la pregona o limpia	la pregona o limpia
35	La habitacion esta sucia	v	v	v
36	La habitacion esta limpia	v	v	v
37	No hay herramientas necesarias	v	v	v
38	El ordenador esta estropeado	El ordenador esta estupendo	El ordenador esta estupendo	El ordenador esta estupendo
39	El choque fue grande	v	choque fue grande	el choque fue el hambre
40	Reordena la habitacion ya	me ordena la habitacion alla	me ordena la habitacion alla	readaptacion vida
41	No quedan helados arriba	v	v	no quedan heladas ari
42	El personal de limpieza de la planta	v	v	persona de limpieza de planta
43	Las habitaciones de la segunda planta	v	v	las habitaciones de la semana
44	El total es de un euro	v	tales de un euro	v
45	El total es de dos euros	el total es de dos seguros	el total es de dos seguros	el total es de dos seguros
46	Donde se encuentra la tercera planta	v	v	v
47	Donde se encuentra la cuarta planta	v	v	v
48	Donde se encuentra la primera planta	v	v	v
49	El caballo y sus tres herraduras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras
50	No queda chocolate blanco en esta planta	v	v	v
51	Hay zapatos pequeños en este lugar	v	v	v
52	Mi cama no esta en la habitación	mi camara esta en la habitacion	mi camara esta en la habitacion	mi cama no esta en la habitacion

Figura 7.3: Resultados de las pruebas con la librería CMUSphinx en un entorno de ruido medio.

53	Las casas de la urbanización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca	
54	Donde se encuentra el centro lo tela	dónde se encuentra el centro lo tela	v	v	
55	La música del hall no es agradable	la música del joven lois árabe	la música del joven lois árabe	la música del joven lois árabe	
56	El portátil no es lo suficientemente potente	el ordenador no es lo solicite mente potente	el ordenador no es lo solicite mente potente	v	
57	El trofeo del campeonato esta expuesto	v	v	v	
58	No hay existencias de madera de chopo	pero hay existencia es de madera de chocó	pero hay existencia es de madera de chocó	no hay existencia es de madera de chocó	
59	Alumbra el recibidor de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta	
60	Donde esta el gerente del hotel	v	v	v	
61	Cuando llega el gerente del hotel	v	v	v	
62	Cual de las habitaciones de la planta esta limpia	v	v	v	
63	Cuantas habitaciones hay en la planta de arriba	v	v	v	
64	Que hacer para jugar bien al waterpolo	que antes para jugar también agua truco lo	que antes para jugar también agua truco lo	que antes para jugar también agua truco lo	
65	Este teléfono android no funciona bien como el otro	ese teléfono android llega también como el otro	ese teléfono android llega también como el otro	ese teléfono android llega también como el otro	
66	La habitación trescientos cinco aun no ha sido limpiada	la habitación no se encuentra en la primera planta	la habitación no se encuentra en la primera planta	la habitación no se encuentra en la primera planta	
67	La habitación trescientos cinco aun no ha sido limpiada	la habitación trescientos cinco echa humo ha sido limpiada	la habitación trescientos cinco echa humo ha sido limpiada	la habitación trescientos cinco echa humo ha sido limpiada	
68	El responsable de la planta uno no llega hasta mañana	v	v	v	
69	Cuando llega el gerente del hotel a la ciudad	cuando llega el gerente del hotel a la tienda	v	v	
70	Busco recepcionista y gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	
71	Tres millones doscientos cuarenta y cuatro millones de euros	v	v	v	
72	Sintonizar el programa de la primera en todas las televisiones	v	v	v	
73	Limpiar los uniformes de todo el personal del hotel	v	v	v	
74	Instalar los programas del pack universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador	
75	Cuantos días quedan para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes	
76	Faltan treinta y cuatro días para el siguiente pedido	v	v	v	
77	Falta personal para cubrir los puestos el lunes	v	v	v	
78	La sala de espectáculos está vacía los domingos	la sala de espectáculos estaba hacia los domingos	v	v	
79	Las dioptrías de las gafas son de dos y media	las dio cria de las cartas son de dos y media	las dio cria de las gracias son de dos y media	las dio cria de las cartas son de dos y media	
80	Cantar y bailar es imprescindible para conseguir el puesto	v	v	v	
81	La decoración del recibidor se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	
82	El cincuenta por ciento del salario es quinientos euros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros	
83	El hotel se encuentra a solo cinco kilometros del centro	el hotel se encuentra solo a ti no pierdo	el hotel se encuentra solo a ti no pierdo	v	
84	En este establecimiento se habla castellano y gallego	... se habla castellanos y la diego	... se habla castellanos y la diego	... se habla castellanos y la diego	
85	Subir la temperatura de la habitación a veinte grados	...a veinte lados	... haga integrados	llamada personal de la recepcion de la platna	
86	Llamar al personal de la recepción de la planta	... de detección de la planta	... de detección de la planta	v	
87	Pintar la habitación cien de color negro azabache	...bien de color negro a la base	...bien de color negro a la base	... de negro a tope	
88	Cuáles de las plantas anteriores no han sido limpiadas	cuálles de las plantas anteriores no han sido olimpiadas	cuálles de las plantas anteriores no han sido olimpiadas	no han sido olimpiadas	
89	Las habitaciones de la primera planta no han sido limpiadas por Álvaro	...no he sido limpias hola Álvaro	...no he sido olimpiadas por Álvaro	...no han sido olimpiadas por Álvaro	
90	El responsable de la segunda planta no llega hasta las dos	v	v	v	
91	No hay existencias de sabanas para las camas de la planta	pero hay existencia es de sábanas...	lobaina existencia ste sábanas...	como lo hicíste decías para las camaras de la planta	
92	A cuanto asciende la suma de todos los salarios del personal	v	v	v	
93	Buscar en youtube musica de ambiente y reproducirla en el dispositivo	buscaré a otra chica era teresa periférico	busca y mi auto posada irás a elegir	busca y mi auto posada irás a elegir	
94	No hay electricidad, agua o gas en ninguna parte del hotel	pero hay electricidad a paga bas en una de él	de ser atacadas ...	de ser atacadas ...	
95	El quinto hotel de la cadena será abierto esta semana en Zamora	... bezamora	... bezamora	... bezamora	
96	El aire acondicionado y la televisión de esta planta están en desuso	v	v	v	
97	Dos millones cuatrocientos cuarenta y seis doscientos veintidós millones de euros costó	...eu nos costo	...eu nos costo	...eu nos costo	

Figura 7.4: Resultados de las pruebas con la librería CMUSphinx en un entorno de ruido medio (cont).

Número	FRASE	INTENTO 1	INTENTO 2	INTENTO 3
1	Sí	v	v	v
2	No	v	v	v
3	Menos	v	v	v
4	Para	v	v	v
5	Más	v	v	v
6	Muy Bien	v	v	v
7	Muy Mal	v	v	v
8	Sigue Probando	v	v	v
9	Donde está	v	v	donde están
10	Cuando llega	v	v	cuando llegan
11	La habitación primera	v	v	v
12	La casa azul	v	v	v
13	La segunda habitación	v	v	v
14	La primera recepción	v	v	v
15	La última piscina	v	v	v
16	La cuarta planta del hotel	v	v	v
17	El cuarto de la plancha	El cuarto de la pl ancha	El cuarto de la pl ancha	El cuarto de la pl ancha
18	La piscina de la planta	v	v	v
19	El waterpolo de mi casa	Él igual te por lo de mi casa	el cuatro por lo de mi casa	Él igual te por lo de mi casa
20	Las mesas del primer recibidor	las mesas del primer recibidos	las mesas del primer recibidos	las mesas del primer recibido
21	La bolsa de compra	v	v	v
22	Mi mochila de clase	v	v	v
23	Esta libre el baño	v	v	v
24	La casa del vecino	v	v	v
25	El cuarto del vecino	v	v	v
26	La sintaxis del idioma	la sin taxis del idioma	la sin taxis del idioma	La cinta crisis del idioma
27	La llave de casa	v	v	la llave de paz
28	La llave del porche	la llave del por che	la llave del por che	la llave del por che
29	El primer día aquí	v	v	v
30	El segundo día aquí	v	v	v
31	El total de habitaciones	v	v	v
32	Dos días lluviosos aquí	dos días lluviosas aquí	dos días lluviosas aquí	días lluviosas aquí
33	Los cascos no funcionan	v	v	v
34	La fregona no limpia	la pregonan do limpia	la pregonan do limpia	la pregonan do limpia
35	La habitacion esta sucia	v	v	v
36	La habitacion esta limpia	v	v	v
37	No hay herramientas necesarias	v	v	v
38	El ordenador esta estropeado	el ordenador está está creado	el ordenador está está creado	el ordenador está creado
39	El choque fue grande	v	v	v
40	Reordena la habitacion ya	re ordenar la habitación de ella	re ordenar la habitación de ella	re ordenar la habitación de ella
41	No hay helados arriba	v	v	v
42	El personal de limpieza de la planta	v	v	v
43	Las habitaciones de la segunda planta	v	v	v
44	El total es de un euro	v	v	v
45	El total es de dos euros	el total es de dos seguros	el total es de dos seguros	el total es de dos seguros
46	Donde se encuentra la tercera planta	v	v	v
47	Donde se encuentra la cuarta planta	v	v	v
48	Donde se encuentra la primera planta	v	v	v
49	El caballo y sus tres herraduras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras	el caballo y sus tres querrá duras
50	No hay chocolate blanco en esta planta	v	v	v
51	Hay zapatos pequeños en este lugar	v	v	v
52	Mi cama no esta en la habitación	mi camara esta en la habitacion	mi camara esta en la habitacion	mi cama no esta en la habitacion

Figura 7.5: Resultados de las pruebas con la librería CMUSphinx en condiciones de poco ruido.

53	Las casas de la urbanización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca	las casas de la organización no están cerca
54	Donde se encuentra el centro del hotel	dónde se encuentra el centro lo tela	v	v
55	La música del hall no es agradable	la música del joven lois árabe	la música del joven lois árabe	la música del joven lois árabe
56	El portátil no es lo suficientemente potente	v	el ordenador no es lo soficite mente potente	v
57	El trofeo del campeonato esta expuesto	v	v	v
58	No hay existencias de madera de chopo	no hay existencia es de madera de chocó	pero hay existencia es de madera de chocó	no hay existencia es de madera de chocó
59	Alumbra el recibidor de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta	al umbral recibido de la primera planta
60	Donde esta el gerente del hotel	v	v	v
61	Cuando llega el gerente del hotel	v	v	v
62	Cual de las habitaciones de la planta esta limpia	v	v	v
63	Cuantas habitaciones hay en la planta de arriba	v	v	v
64	Que haces para jugar tan bien al waterpolo	que antes para jugar también agua truco lo	que antes para jugar también agua truco lo	que antes para jugar también agua truco lo
65	Ese teléfono android no funciona tan bien como el otro	ese teléfono android logra también como el otro	ese teléfono android logra también como el otro	ese teléfono android logra también como el otro
66	El aspirador no se encuentra en la primera planta	El aspira dora no se encuentra en la primera planta	El aspira dora no se encuentra en la primera planta	El aspira dora no se encuentra en la primera planta
67	La habitación trescientos catorce aun no ha sido limpiada	la habitación trescientos catorce echa humo ha sido limpiada	la habitación trescientos catorce echa humo ha sido limpiada	la habitación trescientos catorce a humo así da miedo
68	El responsable de la planta uno no llega hasta mañana	v	v	v
69	Cuando llega el gerente del hotel a la ciudad	cuando llega el gerente del hotel a la tienda	v	v
70	Busco recepcionista y gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel	bush corrección lista el gerente para trabajar en el hotel
71	Tres millones doscientos cuarenta y cuatro millones de euros	v	v	v
72	Sintonizar el programa de la primera en todas las televisiones	v	v	v
73	Limpiar los uniformes de todo el personal del hotel	v	v	v
74	Instalar los programas del pack universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador	instalar los programas del paco universidad en el ordenador
75	Cuantos días quedan para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes	cuantos días quedando para el final del mes
76	Faltan treinta y cuatro días para el siguiente pedido	v	v	v
77	Falta personal para cubrir los puestos el lunes	v	v	v
78	La sala de espectáculos está vacía los domingos	la sala de espectáculos estaba hacia los domingos	v	v
79	Las dioptrías de las gafas son de dos y media	las dio cria de las cartas son de dos y media	las dio cria de las gracias son de dos y media	las dio cria de las cartas son de dos y media
80	Cantar y bailar es imprescindible para conseguir el puesto	v	v	v
81	La decoración del recibidor se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros	la decoración del recibido se compone de tres cuadros
82	El cincuenta por ciento del salario es quinientos euros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros	El cincuenta por ciento del salario es quinientos seguros
83	El hotel se encuentra a solo cinco kilometros del centro	v	v	v
84	En este establecimiento se habla castellano y gallego	...diga llegó	...diga llegó	...diga llegó
85	Subir la temperatura de la habitación a veinte grados	...a veinte Heladas	v	llamada personal de la recepción de la planta
86	Llamar al personal de la recepción de la planta	v	v	v
87	Pintar la habitación cien de color negro azabache	pinta la habitación cien de negro alza ba che	...De negro a tope	... de negro a tope
88	Cuales de las plantas anteriores no han sido limpiadas	cual es de las plantas anteriores no han sido olimpiadas	no han sido olimpiadas	no han sido olimpiadas
89	Las habitaciones de la primera planta no han sido limpiadas por Álvaro	v	... no han sido limpia hadas por Álvaro	... no han sido olimpiadas por Álvaro
90	El responsable de la segunda planta no llega hasta las dos	v	v	v
91	No hay existencias de sábanas para las camas de la planta	pero hay existencia es de sábanas...	Idbains existencia ste sábanas...	como lo hicíte decías para las camaras de la planta
92	A cuanto asciende la suma de todos los salarios del personal	buscar es otra chica era teresa periférico	v	v
93	Buscar en youtube musica de ambiente y reproducirla en el dispositivo	busca y mi auto posada irás a elegir	busca y mi auto posada irás a elegir	busca y mi auto posada irás a elegir
94	No hay electricidad, aguno gas en ninguna parte del hotel	pero hay electricidad a paga las en una de él	de ser atacadas...	de ser atacadas ...
95	El quinto hotel de la cadena será abierto esta semana en Zamora	... bezamora	... bezamora	... bezamora
96	El aire acondicionado y la televisión de esta planta están en desuso	v	v	v
97	Dos millones cuatrocientos cuarenta y seis mil doscientos veintidós euros costó	...eu nos costo	...eu nos costo	...eu nos costo

Figura 7.6: Resultados de las pruebas con la librería CMUSphinx en condiciones de poco ruido(cont).

7.4.1 Definición de la prueba

Para comprobar que la aplicación en conjunto funciona correctamente hemos realizado una interacción normal con sinvad empleando el comando Habitación. Este comando es usado por el personal de mantenimiento de un hotel y, para una habitación concreta, le hace una serie de preguntas acerca de cómo es la habitación, qué tareas ha realizado.

7.5 Resultados y conclusiones

Una vez concluida la etapa de desarrollo, realizamos las pruebas finales de la aplicación con ambas librerías transcriptoras. La primera prueba, con CMUSphinx fue un rotundo fracaso. El desempeño de la librería en la Raspberry Pi 4B es muy malo, tarda en el arranque más ed 15 segundos y cada transcripción de media un minuto, algunas hasta dos. Esto hace que el transcriptor no sea capaz de interpretar ninguna palabra correctamente por el retraso que introduce. Además, provoca que otros módulos que si funcionan correctamente, como el de comunicaciones o el sintetizador sufran retrasos. Por tanto, decidimos investigar acerca de las causas. Para ello, empleamos el comando htop, para visualizar el consumo de CPU y memoria por parte de la aplicación. Así, podemos concluir que la librería CMUSPhinx ahora a la cpu de la Raspberry provocando una utilización de más del 99 por ciento en sus cuatro núcleos. Esto se debe, en parte, a que la transcripción con esta librería, crea varios subprocesos, que ahogan al procesador. Con respecto a esta librería, por todo lo comentado anteriormente, decidimos descartarla, es decir, su funcionalidad está implementada pero no puede ser utilizada por nuestra aplicación sin cambiar el código y recompilar.

En las segundas pruebas, con la librería Vosk, vemos una enormísima mejoría, que posibilitan una comunicación fluída con sinvad. Como la interacción con el sistema es por vía voz hemos decidido activar el modo log y mostrar en la consola la realización de la prueba. Este resultado se encuentra el anexo B y es importante destacar que esta salida por pantalla no está pensada para que la vea el usuario es únicamente con fines ilustrativos para la memoria.

Teniendo en cuenta lo anterior, podemos ver que la aplicación transcribe e interpreta casi correctamente toda la interacción que hemos hecho. Destacamos un error de trasncripción en el que confunde ”aspirada” con inspirada y un error conocido en el que elimina la primera letra de las palabras, que está mencionado en 8.2. Con una visión en conjunto de la iteracción podríamos decir que bajo condiciones de ruido normales cumple los requisitos iniciales, destacando que es rápida, fácil de usar y útil. La siguiente prueba sería necesaria pero no se ha podido realizar por falta de tiempo y es la prueba en un entorno real con el hardware escogido específicamente para el proyecto (ver sección 4.2

Capítulo 8

Conclusiones y líneas futuras

En este capítulo se presentará la situación final del trabajo, las lecciones aprendidas, la relación con las competencias de la titulación y de la mención y los pasos posteriores del proyecto.

8.1 Conclusiones

Si nos remontamos al inicio de la memoria, en la sección 1.2, podemos ver cuales eran los objetivos iniciales. Ahora, haremos una relación de las funcionalidades finales que se han conseguido:

- Se ha creado una aplicación que permite interactuar con el usuario solamente con su propia voz.
- Se ha conseguido que sea sencilla y rápida de usar.
- Se ha conseguido que su tasa de acierto sea alta, lo que provoca que también sea útil.
- La app es fácilmente replicable, su instalación es sencilla (como podemos ver en el manual, en A.1) y se podría automatizar.
- El coste material es inferior al de un smartphone de gama media.

Con respecto a la carrera en general, se ha aprendido cosas muy generales, entre ellas las siguientes:

- Se ha como aplicar todas las asignaturas vistas en la carrera en conjunto en algo de tal envergadura como un proyecto.
- Se ha aprendido a trabajar en equipo.
- Se ha aprendido a trabajar con distintas metodologías de trabajo, como SCRUM.

- Se ha aprendido a trabajar de forma autónoma.
- Se ha aprendido a tratar con el cliente.

Si hacemos énfasis en la mención de ingeniería de computadores podemos destacar las siguientes competencias.

- Se han aplicado conocimientos de como funcionan dispositivos de entrada y salida como micrófonos o altavoces.
- Se ha aprendido a identificar las distintas arquitecturas de procesadores y a determinar para qué es adecuada cada una.
- Se ha aprendido a seleccionar el hardware concreto para la funcionalidades que deseamos implementar

8.2 Errores conocidos

En las pruebas de la aplicación se ha detectado algún error de mínima importancia pero que cabe destacar.

En algunos casos, el transcriptor elimina las letras iniciales de los comandos. Esto, aunque podría ser problemático, en este proyecto concreto no es representativo ya que Sinvad puede interpretar correctamente el mensaje aunque le falte la primera letra.

8.3 Líneas futuras

Los próximos pasos de este proyecto deben ser solucionar los errores conocidos y desarrollar las pruebas en un entorno real.

Otros aspectos que se puedes ampliar serían:

- Soporte para otras lenguas: sería bastante sencillo de implementar, la mayoría de lenguas ya poseen modelos de lenguaje.
- Mejora del sintetizador: Su sonido es bastante metálico pero cumple su función.
- Definición de comandos básicos para que pueda tener algunas funciones sin estar conectado a la red.

Apéndices

Apéndice A

DOCUMENTACIÓN SINVADRASP

A.1 PRERREQUISITOS

- Tener una Raspberry Pi 4B de 2GB con un SO Raspbian Desktop versión lanzada 08/11/2021. Esto se puede conseguir con raspberry pi imager y descargando el archivo .zip que contiene el archivo .img que necesita este programa. Dicha versión puede descargarse aquí.

A.2 CONFIGURACIONES ANTERIORES

En la carpeta wifi, escribir nombre wifi y contraseña de la red con **UN ÚNICO ESPACIO** separando campos, por ejemplo:

```
wpa-ssid nombreredwifi
wpa-psk contraseñadelwifi
```

Luego, ejecutar como root el archivo script.sh, este configurará el wifi (empleando /etc/network/interfaces) e instalará java en su versión 11 si no está instalada ya. Por último, instalará una librería compartida de vosk y moverá sus archivos del directorio

```
/home/pi/.local/lib/python3.9/site-packages/vosk/libvosk.so
```

al directorio:

```
/usr/lib/libvosk.so.
```

En caso de que la instalación no se produzca en el primer directorio, mover manualmente. No sirve establecer links simbólicos con la libería compartida, hay que mover al directorio raíz. Si no se sabe dónde se ha instalado, volver a lanzar pip install vosk, como root, y devolverá la ruta dónde se ha instalado.

A.3 CONFIGURACIÓN DE SINVADRASP

A.3.1 CONFIGURACION DE LOGIN

Para hacer login introducir los parámetros de login userPwd y profileID en el archivo de configuración LoginParameters.properties. Este se encuentra comprimido en el archivo sinvadrasp-0.0.1-SNAPSHOT.jar. Para acceder abrir con un archivador de ficheros como el que viene por defecto en raspbian, editar los parámetros y guardar. A continuación, sin cerrar el archivador, clicar arriba en añadir archivo y seleccionar el archivo de login ya que lo hemos modificado. En esta imagen, dicho botón sería el primero del tercer grupo de botones empezando por la izquierda.

A.3.2 MÓDULO TRANSCRIPTOR (CMUSPHINX)

El diccionario base, el language model y el directorio del modelo se encuentran en la carpeta es-es. Estos tres parámetros son recogidos por la app en el archivo ConfigurationParameters.properties.

A.3.3 MÓDULO TRANSCRIPTOR (VOSK)

El modelo de lenguaje castellano se encuentra en la carpeta vosk. También posee un parámetro de Deathtime, que realmente mide los ciclos que pasan hasta que para de escuchar. Por último, se puede fijar el micrófono por el que se quiere escuchar a través de audioIndex. Estos tres parámetros son recogidos por la app en el archivo ConfigurationParameters.properties.

A.3.4 MÓDULO WAKE-UP(porcupine)

El archivo de funcionamiento para castellano así como el de la palabra clave “hola beta” en SO Windows, raspbian o Linux se encuentran en la carpeta porcupine. El archivo de funcionamiento y el de la palabra clave, dependiendo del sistema son recogidos por la app en el archivo ConfigurationParameters.properties. Además, se ha añadido un parámetro para seleccionar el micrófono que se quiere usar para la escucha de este módulo.

A.3.5 MÓDULO SINTETIZADOR(FREE-TTS)

El nombre de la voz del sintentizador se encuentra en ConfigurationParameters.properties. Para cambiarlo hay que acceder a dicho archivo.

A.3.6 SONIDOS DE INICIO Y FIN DE ESCUCHA

Estos archivos se encuentran en la carpeta sounds. Son recogidos por la app en el archivo ConfigurationParameters.properties.

A.3.7 MICRÓFONOS DISPONIBLES

En el archivo ConfigurationParameters.properties, el primer parámetro (Microphone.showMics=S) nos permite seleccionar si queremos ver al inicio de nuestra app los micrófonos que tenemos disponibles para usar. Para verlos introduciremos una “S”, para no verlos una “N” o cualquier otro carácter. La lista que despliegue nos dará el número que tenemos que meter en (WakeUp.DeviceIndex=) o en (VoksTranscriber.audioDeviceIndex=).

A.3.8 CAMBIO ENTRE TRANSCRIPTORES

Debido a la ineficiencia de CmuSphinx no se posibilita el cambio de transcriptor sin modificar el código. El transcriptor activo ahora mismo es el de vosk.

A.3.9 CONFIGURATIONPARAMETERSMANAGER

Todos los ítems que hemos comentado en esta lista se enlazan con la app mediante ConfigurationParameters.properties. Si se cambia el nombre, se reubican estos archivos o se mejoran, habrá que modificar ConfigurationParameters.properties. Para modificarlo se seguirá el mismo procedimiento que con LoginParameters.properties

Apéndice B

Resultado prueba aplicación

A continuación podemos ver el log generado de una interacción con la aplicación Sinvadrasp. Hay que hacer énfasis en que este feedback nunca se dará entre el usuario final y la app, es meramente ilustrativo.

```
   pi@raspberrypi:~/Desktop/sinvadrasprasp $ java -jar sinvadrasp.jar
Device 5: default [default]
Device 7: U091571550 [plughw:1,0]
Device 8: Microphone [plughw:2,0]
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:habitación tres
Enviando comando recibido: habitación tres
La respuesta es : "cambio de sábanas"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:cambio de sábanas
Enviando comando recibido: cambio de sábanas
La respuesta es : "mantenimiento"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:sin mantenimiento
Enviando comando recibido: sin mantenimiento
La respuesta es : "limpiar polvo"
Palabra detectada
```

```
1Transcribiendo...
3Comando preparado...
1Comando procesado...:sin limpieza de polvo
Enviando comando recibido: sin limpieza de polvo
La respuesta es : "aspirar"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:abitación inspirada
Enviando comando recibido: abitación inspirada
La respuesta es : "fregar"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:sin fregar
Enviando comando recibido: sin fregar
La respuesta es : "lavandería"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:sin lavandería
Enviando comando recibido: sin lavandería
La respuesta es : "interruptor de luces apagado"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:interruptor de luces apagado
Enviando comando recibido: interruptor de luces apagado
La respuesta es : "interruptor de calefacción y aire acondicionado apagado"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:interruptor de encendido
Enviando comando recibido: interruptor de encendido
La respuesta es : "reposición de papel higiénic"
Palabra detectada
1Transcribiendo...
```

```
3Comando preparado...
1Comando procesado...:sin reposición de papel higiénico
Enviando comando recibido: sin reposición de papel higiénico
La respuesta es : "comprobar almohadas y mantas"
Palabra detectada
1Transcribiendo...
3Comando preparado...
1Comando procesado...:sin almohadas
Enviando comando recibido: sin almohadas
La respuesta es : "has terminado en esta habitación muchas gracias"
```

Lista de acrónimos

API Interfaz de programación de aplicaciones. 30

DTO Objeto de transferencia de datos. 30

IDE Entorno de desarrollo integrado. 4

JSON Notación de Objetos de JavaScript. 26

REST transferencia de estado representacional. 30

STT Discurso a texto. 4, 5

TTS Texto a discurso. 4, 5

USB Bus Universal en Serie. 3, 4

Glosario

concurrencia Acceso simultáneo a un determinado recurso por dos o más threads diferentes. 29

módulo Parte de un programa que realiza una tarea concreta. . 21

palabra clave Palabra que provoca que se inicie la aplicación. . 7, 22

paralelismo Realización de varias tareas al mismo tiempo. . 24

sintetizador Programa capaz de verbalizar texto plano.. 5

thread Hilo de ejecución de un programa. 24

transcriptor Programa capaz de convertir palabras a texto.. 5

unix Sistema operativo que nace en 1969. En él se basan otros como Ubuntu, Debian, Raspberry OS.... 4

Bibliografía

[1] "Banana pi bpi-m4," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://wiki.banana-pi.org/Banana_Pi_BPI-M4

[2] "Raspberry pi zero: info," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.raspberrypi.com/products/raspberry-pi-zero/

[3] "Raspberry pi 3: info," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

[4] "Raspberry pi 4: info," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[5] "¿qué es java?" consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.java.com/es/download/help/whatis_java.html

[6] "Voice2json," consultado el 15 de febrero de 2022. [En línea]. Disponible en: http://voice2json.org/

[7] "Cmusphinx tutorial for developers," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://cmusphinx.github.io/wiki/tutorial/

[8] "Vosk," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://alphacephei.com/vosk/

[9] "¿qué es freetts?" consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://freetts.sourceforge.io/#what_is_freetts

[10] "Porcupine sdk introduction," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://picovoice.ai/docs/porcupine/

[11] "Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/

[12] "Scrum (desarrollo de software)," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)

[13] "Desarrollo ágil de software," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Desarrollo_ágil_de_software

[14] "¿qué es la metodología ágil?" consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://www.redhat.com/es/devops/what-is-agile-methodology

[15] "How to train your own wake word," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://github.com/MycroftAI/mycroft-precise/wiki/Training-your-own-wake-word

[16] "Interface targetdataline," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://docs.oracle.com/javase/7/docs/api/javax/sound/sampled/TargetDataLine.html

[17] "Interface dataline," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://docs.oracle.com/javase/7/docs/api/javax/sound/sampled/DataLine.html

[18] "Máquina de estados," consultado el 15 de febrero de 2022. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Máquina_de_estados