



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Ingeniería de Software

**Desarrollo de frontend web para
grabación de ficheros de audio en tiempo
real**

Autor:

Daniel Figueroa Fernández



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Ingeniería de Software

**Desarrollo de frontend web para
grabación de ficheros de audio en tiempo
real**

Autor:

Daniel Figueroa Fernández

Tutores:

Valentín Cardeñoso Payo

Cristian Tejedor García

A mi familia que me ha apoyado y ha hecho esto posible.

Agradecimientos

Muchas personas me han ayudado y apoyado para que este proyecto salga adelante. Entre ellas quiero mencionar a mi familia y amigos por apoyarme durante todos estos meses, pero sobre todo quiero agradecer a mis tutores Cristian Tejedor García y Valentín Cardeñoso Payo por su ayuda y dedicación, sin la cual la realización de este proyecto no habría sido posible.

Resumen

En la actualidad, el dominio de idiomas como el inglés o el español se ha convertido en una habilidad crucial en una sociedad cada vez más global. Como resultado, ha habido un aumento en la demanda de herramientas que apoyen el aprendizaje de idiomas en diversas áreas. Una de las tecnologías con mayor potencial para mejorar aspectos como la pronunciación es el reconocimiento automático de habla. Esta tecnología permite practicar y recibir retroalimentación precisa. Sin embargo, para que las personas sin conocimientos informáticos puedan aprovechar sistemas tan complejos, es fundamental crear una interfaz intuitiva y fácil de usar. Con este objetivo en mente, surge este proyecto, que busca desarrollar un frontend web capaz de grabar audio y enviar solicitudes al servicio para obtener transcripciones de audio en tiempo real. Para ello, se desarrollará un frontend web que tenga las capacidades de grabar audio y realizar peticiones al servicio para obtener su transcripción en tiempo real.

Abstract

Currently, language proficiency in languages such as English or Spanish has become a crucial skill in an increasingly global society. As a result, there has been an increase in demand for tools that support language learning in various areas. One technology with great potential to improve aspects such as pronunciation is automatic speech recognition. This technology allows for practice and precise feedback. However, in order for people without computer knowledge to take advantage of such complex systems, it is essential to create an intuitive and user-friendly interface. With this goal in mind, this project aims to develop a web frontend capable of recording audio and sending requests to the service to obtain real-time transcriptions from audio files. To achieve this, a web frontend will be developed with the capabilities of recording audio and making requests to the service to obtain its real-time transcription.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Metodología	2
1.5. Estructura de la memoria	3
2. Planificación	5
2.1. Planificación inicial	5
2.2. Entorno tecnológico	6
2.2.1. Herramientas utilizadas	6
2.2.2. Entorno de trabajo	7
2.3. Estimación de riesgos	10
2.4. Estimación de costes	12
2.4.1. Salario del trabajador	12
2.4.2. Costes indirectos aplicados al espacio de trabajo	12
2.4.3. Costes del hardware	12
2.4.4. Costes del software y servicios	13
2.4.5. Costes totales	13
2.5. Planificación final	14
3. Estado de la cuestión	17
3.1. Otras tecnologías ASR en el mercado	18
3.1.1. Google Speech-to-Text	18
3.1.2. Kaldi	18
3.1.3. Modelos ASR End2end	19
3.2. Entorno y herramientas tecnológicas	20
3.2.1. Angular	20
3.2.2. Node.js	20
3.2.3. NPM	21
3.2.4. Docker	21
3.2.5. Nginx	22
3.2.6. Visual Studio Code	22
3.2.7. GitLab	22

3.2.8. Balsamiq Wireframes	23
3.2.9. Swagger	23
3.2.10. BabelEdit	24
3.2.11. Overleaf	24
4. Descripción de las iteraciones	25
4.1. Iteración 1	25
4.2. Iteración 2	26
4.2.1. Descripción de requisitos	26
4.2.2. Pruebas de concepto	29
4.2.3. Voice Activity Detection	30
4.2.4. Maquetación de las pantallas	32
4.3. Iteración 3	34
4.4. Iteración 4	35
4.4.1. Implementación de la nueva funcionalidad	35
4.4.2. Diseños finales y diseño modular	38
4.4.3. Pruebas funcionales y principales problemas	43
4.5. Iteración 5	44
4.5.1. Pruebas	44
4.5.2. Despliegue	44
4.5.3. Redacción de la memoria	46
5. Estado final de la aplicación	47
5.1. Diagramas de análisis	47
5.1.1. Historias de usuario	47
5.1.2. Modelo de dominio	54
5.1.3. Diagramas de actividad	57
5.2. Estructura del proyecto	72
5.2.1. Estructura del <i>frontend</i>	72
5.2.2. Arquitectura y patrones de diseño	75
5.3. Diagramas de diseño	76
5.3.1. Diagrama de paquetes	76
5.3.2. Diagrama de despliegue	77
6. Pruebas	79
6.1. Test de funcionalidad	79
6.2. Test de usabilidad	80
6.2.1. Conclusiones de los test de usabilidad	82
7. Conclusiones	83
7.1. Trabajo futuro	84

Apéndices	87
A. Acrónimos	87
B. Manual de despliegue	89
B.1. Instalación de git y clonación del repositorio	89
B.2. Instalación de docker	90
B.3. Creación de la imagen de Docker y despliegue en un contenedor	91
C. Manual de configuración	93
C.1. Cambiar los valores por defecto	93
C.2. Cambiar lista de traducciones	94
C.3. Cambiar el certificado SSL	95
D. Manual de usuario	97
E. Pruebas de usabilidad con usuarios reales	101
E.1. Usuario 1	101
E.1.1. Prueba 1	101
E.1.2. Prueba 2	103
E.1.3. Prueba 3	105
E.1.4. Preguntas	107
E.2. Usuario 2	107
E.2.1. Prueba 1	107
E.2.2. Prueba 2	109
E.2.3. Prueba 3	111
E.2.4. Preguntas	113
E.3. Usuario 3	113
E.3.1. Prueba 1	113
E.3.2. Prueba 2	115
E.3.3. Prueba 3	117
E.3.4. Preguntas	119
E.4. Usuario 4	119
E.4.1. Prueba 1	119
E.4.2. Prueba 2	121
E.4.3. Prueba 3	123
E.4.4. Preguntas	125
Bibliografía	129

Índice de figuras

1.1. Esquema de un desarrollo iterativo e incremental [6]	3
2.1. Distribución de las iteraciones según sus semanas de desarrollo.	6
2.2. Comparación de la estimación original con la duración real.	14
3.1. Logo de Angular.	20
3.2. Logo de Node.js.	21
3.3. Logo de NPM.	21
3.4. Logo de Docker.	21
3.5. Logo de Nginx.	22
3.6. Logo de VSC.	22
3.7. Logo de GitLab.	22
3.8. Logo de Balsamiq.	23
3.9. Logo de Swagger.	23
3.10. Logo de BabelEdit.	24
3.11. Logo de Overleaf.	24
4.1. opus-recorder vs recorderjs vs recordrtc.	29
4.2. node-vad vs vad vs voice-activity-detection.	29
4.3. Mock-up del inicio de sesión.	32
4.4. Mock-up del registro.	32
4.5. Mock-up de la página principal.	33
4.6. Mock-up del historial.	33
4.7. Diseño de la página principal.	38
4.8. Diseño de la página principal en la vista para móvil.	39
4.9. Diseño del inicio de sesión.	40
4.10. Diseño del inicio de sesión en la vista para móvil.	40
4.11. Diseño del registro.	41
4.12. Diseño del registro en la vista para móvil.	41
4.13. Diseño del historial.	42
4.14. Diseño del historial en la vista para móvil.	43
4.15. Aviso al intentar acceder a la aplicación.	45
5.1. Diagrama de dominio.	54
5.2. Diagrama de actividad de HU01.	58

5.3. Diagrama de actividad de HU02.	59
5.4. Diagrama de actividad de HU03.	60
5.5. Diagrama de actividad de HU04.	61
5.6. Diagrama de actividad de HU05.	62
5.7. Diagrama de actividad de HU06.	63
5.8. Diagrama de actividad de HU07.	64
5.9. Diagrama de actividad de HU08.	65
5.10. Diagrama de actividad de HU09.	67
5.11. Diagrama de actividad de HU10.	68
5.12. Diagrama de actividad de HU11.	69
5.13. Diagrama de actividad de HU12.	70
5.14. Diagrama de actividad de HU13.	71
5.15. Estructura del proyecto Angular.	72
5.16. Estructura de la carpeta app.	74
5.17. Diagrama de paquetes.	76
5.18. Diagrama de despliegue.	77
B.1. Resultado del comando docker run hello-world.	90
B.2. Resultado del comando docker container ls.	92
D.1. Pantalla de inicio de sesión.	97
D.2. Pantalla de registro.	98
D.3. Pantalla de inicio.	98
D.4. Pantalla de inicio mientras se graba un audio.	99
D.5. Aviso que se muestra al superar el límite de caracteres.	99
D.6. Pantalla del historial.	100

Índice de tablas

2.1. Ordenador utilizado en desarrollo.	8
2.2. Smartphone utilizado en desarrollo.	8
2.3. Servidor utilizado para el despliegue.	9
2.4. Tabla de riesgos.	10
2.5. Tabla de planes de acción para los riesgos.	11
2.6. Tabla de estimación de costes del hardware.	13
2.7. Tabla de estimación de costes totales.	13
4.1. Tabla de requisitos funcionales.	27
4.2. Tabla de requisitos no funcionales.	28
5.1. Historia de usuario HU01.	48
5.2. Historia de usuario HU02.	48
5.3. Historia de usuario HU03.	48
5.4. Historia de usuario HU04.	49
5.5. Historia de usuario HU05.	49
5.6. Historia de usuario HU06.	49
5.7. Historia de usuario HU07.	50
5.8. Historia de usuario HU08.	50
5.9. Historia de usuario HU09.	51
5.10. Historia de usuario HU10.	51
5.11. Historia de usuario HU11.	52
5.12. Historia de usuario HU012.	52
5.13. Historia de usuario HU13.	53
5.14. Entidad Usuario.	54
5.15. Entidad ArchivoAudio.	55
5.16. Entidad EnvioAudio.	55
5.17. Entidad Idioma.	55
5.18. Entidad TranscripcionAudio.	56
5.19. Entidad Palabra.	56
6.1. Plantilla utilizada para los test de usabilidad.	80
6.2. Perfiles de usuario.	81
E.1. Resultados del usuario 1 en la acción 1.	101

E.2. Resultados del usuario 1 en la acción 2.	102
E.3. Resultados del usuario 1 en la acción 3.	102
E.4. Resultados del usuario 1 en la acción 1.	103
E.5. Resultados del usuario 1 en la acción 2.	103
E.6. Resultados del usuario 1 en la acción 3.	103
E.7. Resultados del usuario 1 en la acción 4.	104
E.8. Resultados del usuario 1 en la acción 1.	105
E.9. Resultados del usuario 1 en la acción 2.	105
E.10.Resultados del usuario 1 en la acción 3.	105
E.11.Resultados del usuario 1 en la acción 4.	106
E.12.Resultados del usuario 1 en la acción 5.	106
E.13.Resultados del usuario 2 en la acción 1.	108
E.14.Resultados del usuario 2 en la acción 2.	108
E.15.Resultados del usuario 2 en la acción 3.	108
E.16.Resultados del usuario 2 en la acción 1.	109
E.17.Resultados del usuario 2 en la acción 2.	109
E.18.Resultados del usuario 2 en la acción 3.	109
E.19.Resultados del usuario 2 en la acción 4.	110
E.20.Resultados del usuario 2 en la acción 1.	111
E.21.Resultados del usuario 2 en la acción 2.	111
E.22.Resultados del usuario 2 en la acción 3.	111
E.23.Resultados del usuario 2 en la acción 4.	112
E.24.Resultados del usuario 2 en la acción 5.	112
E.25.Resultados del usuario 3 en la acción 1.	114
E.26.Resultados del usuario 3 en la acción 2.	114
E.27.Resultados del usuario 3 en la acción 3.	114
E.28.Resultados del usuario 3 en la acción 1.	115
E.29.Resultados del usuario 3 en la acción 2.	115
E.30.Resultados del usuario 3 en la acción 3.	115
E.31.Resultados del usuario 3 en la acción 4.	116
E.32.Resultados del usuario 3 en la acción 1.	117
E.33.Resultados del usuario 3 en la acción 2.	117
E.34.Resultados del usuario 3 en la acción 3.	117
E.35.Resultados del usuario 3 en la acción 4.	118
E.36.Resultados del usuario 3 en la acción 5.	118
E.37.Resultados del usuario 4 en la acción 1.	120
E.38.Resultados del usuario 4 en la acción 2.	120
E.39.Resultados del usuario 4 en la acción 3.	120
E.40.Resultados del usuario 4 en la acción 1.	121
E.41.Resultados del usuario 4 en la acción 2.	121
E.42.Resultados del usuario 4 en la acción 3.	121
E.43.Resultados del usuario 4 en la acción 4.	122
E.44.Resultados del usuario 4 en la acción 1.	123

ÍNDICE DE TABLAS

E.45.Resultados del usuario 4 en la acción 2.	123
E.46.Resultados del usuario 4 en la acción 3.	123
E.47.Resultados del usuario 4 en la acción 4.	124
E.48.Resultados del usuario 4 en la acción 5.	124

Capítulo 1

Introducción

1.1. Contexto

En un mundo en el que la importancia de conocer idiomas como el inglés o el español va en aumento, cada día más usuarios buscan formas de poder practicar y mejorar su dominio de estos idiomas. Una de las partes más difíciles de trabajar es la pronunciación, que requiere tanto conocimiento del idioma para saber como suenan las palabras, como una gran cantidad de horas de práctica.

Teniendo en cuenta la cantidad de trabajo que requiere esta mejora de la pronunciación, cada vez más personas interesadas en aprender un idioma buscan herramientas que les permitan hacerlo de forma más fácil y rápida. Aprovechando este aumento en la demanda y un rápido desarrollo de la tecnología, están apareciendo herramientas que utilizan ASR (Reconocimiento Automático de Habla), que permiten a estas personas practicar y recibir retroalimentación en cualquier momento.

Este proyecto surge de una propuesta de Trabajo Fin de Grado (TFG) publicada por Cristian Tejedor García en la página web de ofertas de TFG [1] de la Escuela de Ingeniería Informática de Valladolid, bajo el nombre de "Desarrollo de frontend web para grabación de ficheros de audio en tiempo real". El objetivo de ésta es construir un frontend web para el grupo ECA-SIMM [2]. Este frontend tendrá como objetivo permitir a los usuarios interactuar de forma sencilla con un servicio web creado anteriormente. Además, este servicio web será transparente al usuario a la hora de incluir diferentes servicios API de ASR.

Uno de los servicios de los que el frontend desarrollado obtendrá el reconocimiento automático del habla en tiempo real fue creado por el propio Cristian Tejedor García como parte de tu trabajo como postdoc en el CLST - Centre for Language and Speech Technology de Radboud University [3], y se encuentra disponible disponible mediante una API restringida desde mayo de 2022 [4], para proyectos de investigación internacionales relacionados con el reconocimiento automático del habla en el CLST.

1.2. Motivación

La interacción persona-computador es una parte de la informática que muchas veces se pasa por alto, pero que puede marcar la diferencia entre que un producto sea un éxito o un completo fracaso. Crear una interfaz clara y que proporcione al usuario toda la información necesaria para el uso de una aplicación (de cualquier tipo, y en el caso de este TFG en particular, una aplicación web) tendrá como objetivo que la satisfacción del usuario aumente.

Este TFG pretende desarrollar un frontend web que haga el servicio web anteriormente mencionado más accesible y fácil de manejar, dando una interfaz que muestre los datos de una forma más legible, y permitiendo realizar tareas como grabar audio y enviarlo al servidor para obtener su transcripción en tiempo real de forma sencilla. Además, la aplicación web utilizará diseño modular para poder ser utilizada de forma cómoda en la mayoría de dispositivos más comunes en la actualidad, como podrían ser smartphones, tablets u ordenadores.

Como otra ventaja, este frontend no tiene por qué utilizarse exclusivamente para el servicio que se ha proporcionado como base de este TFG, sino que además el grupo ECA-SIMM podría utilizarlo como interfaz para otras API que proporcionen servicios de ASR en tiempo real, realizando cambios mínimos.

1.3. Objetivos

El objetivo general de el proyecto es es construir un frontend web que permita grabar audio y enviarlo a un servidor en tiempo real, mostrando la transcripción devuelta por el mismo, así como ver un histórico de las grabaciones del usuario. Este objetivo general se puede dividir en:

- **Objetivo 1:** diseñar las interfaces de las diferentes vistas que formarán parte de la aplicación web, haciendo énfasis en su usabilidad.
- **Objetivo 2:** desarrollar la funcionalidad necesaria para la grabación de audio y su envío al servidor. Esto incluye tanto la grabación de audio y su transformación a archivo para poder enviarse, como la integración de un sistema de VAR (Voice Activity Recognition).
- **Objetivo 3:** desarrollar la aplicación web que permita grabar y enviar audio al servidor en tiempo real, así como ver y descargar el histórico de grabaciones del usuario.

1.4. Metodología

Para el desarrollo de este proyecto se ha decidido aplicar una metodología iterativa e incremental [5]. Esto se debe principalmente a que esta metodología da una mayor flexibilidad que otros modelos como podría ser el de cascada, lo cual puede ser muy beneficioso teniendo en cuenta la situación del estudiante. Además, permite la entrega al cliente de nueva funcionalidad en cada iteración, lo cual

permite recibir mejor feedback y de forma más frecuente, además de que proporciona más claridad sobre el ritmo al que se está desarrollando el proyecto con respecto a la planificación. Por otro lado, otras metodologías que confieren ventajas similares, como podrían ser las ágiles, no tienen un modo realista de ser aplicadas en un equipo formado por una única persona.



Figura 1.1: Esquema de un desarrollo iterativo e incremental [6]

En la figura 1.1 se puede ver un gráfico que representa los pasos a seguir durante cada una de las iteraciones de las que se compone un desarrollo iterativo e incremental, que comienzan por la creación de un planteamiento o planificación de la iteración a partir de la planificación inicial, para después definir los requerimientos que se deben cumplir al final de la iteración. Tras esto se realiza el análisis y el diseño de las partes de la aplicación que se van a desarrollar en esta iteración, para posteriormente poder hacer la implementación de las mismas. Por último, se realizan las pruebas necesarias para comprobar si la nueva funcionalidad funciona correctamente, y se realiza una evaluación del resultado para poder comenzar la siguiente iteración. En el capítulo 4 se describe en detalle todo lo ocurrido durante cada una de las iteraciones de las que se compone este proyecto.

1.5. Estructura de la memoria

- **Introducción:** se expone el proyecto desde una perspectiva general, hablando del contexto y los objetivos del proyecto, de la metodología aplicada y la estructura de la memoria.
- **Planificación:** descripción y estimación de las iteraciones, estimación del coste del proyecto y descripción de los riesgos identificados y sus planes de acción correspondientes.
- **Estado de la cuestión:** descripción de las bases del proyecto, así como de tecnologías similares que ya están en el mercado, y de las herramientas utilizadas.
- **Descripción de las iteraciones:** se describe el trabajo realizado en cada una de las iteraciones.

- **Estado final de la aplicación:** se describe el análisis y diseño de la aplicación, contiene todos los diagramas de análisis y diseño con sus correspondientes aclaraciones.
- **Pruebas:** descripción de las pruebas realizadas, incluye las pruebas de funcionalidad y las de usabilidad.
- **Conclusiones:** se enumeran las conclusiones sacadas del proyecto, así como los conocimientos obtenidos y posibles mejoras de cara al futuro.
- **Apéndices:** distintos documentos adicionales como el manual de instalación y despliegue.
- **Bibliografía:** referencias a páginas web, artículos de investigación, libros o proyectos consultados a lo largo del desarrollo del proyecto.

Capítulo 2

Planificación

2.1. Planificación inicial

Este TFG se ha llevado a cabo en el transcurso del segundo cuatrimestre del curso 2022-2023, comenzando el día 14 de febrero. Durante este tiempo, el alumno también ha estado realizando las prácticas en empresa, por lo que el número de horas diarias de trabajo para este TFG de lunes a viernes es de 3 horas aproximadamente, que se compensará con horas extra durante el fin de semana para asegurar un progreso adecuado en el proyecto. Se va a utilizar una metodología basada en iteraciones, repartidas de la siguiente forma:

- **1ª Iteración (1 semana - 10 horas)**

Esta primera etapa consistirá principalmente en una primera reunión con el tutor del proyecto para hacer una definición general de los requisitos y funcionalidades del producto, así como de unas pequeñas pruebas con las tecnologías y herramientas escogidas (en este caso angular y docker), para comprobar que son adecuadas para la realización del proyecto.

- **2ª Iteración (2 semanas - 30 horas)**

A lo largo de esta iteración se realizarán pruebas más extensivas, con el objetivo de lograr una mayor familiarización con las tecnologías. Además, se realizará la maquetación de la aplicación.

- **3ª Iteración (4 semanas - 60 horas)**

El objetivo principal de esta iteración es obtener un prototipo funcional de la aplicación, sobre el cual se pueda trabajar para realizar mejoras, y del cual se pedirá retroalimentación.

- **4ª Iteración (7 semanas - 100 horas)**

Como se puede observar, esta iteración será la más larga, puesto que su objetivo es terminar de crear la aplicación, implementando toda la funcionalidad que no se haya implementado en la iteración anterior, y realizando las mejoras oportunas sobre el prototipo.

- **5ª Iteración (3 semanas - 45 horas)**

Esta última iteración consistirá en la realización de todas las pruebas necesarias para asegurar

que se cumplen los requisitos correctamente (y responsive design/diseño). Por otro lado, durante esta iteración se completará la redacción de la memoria.

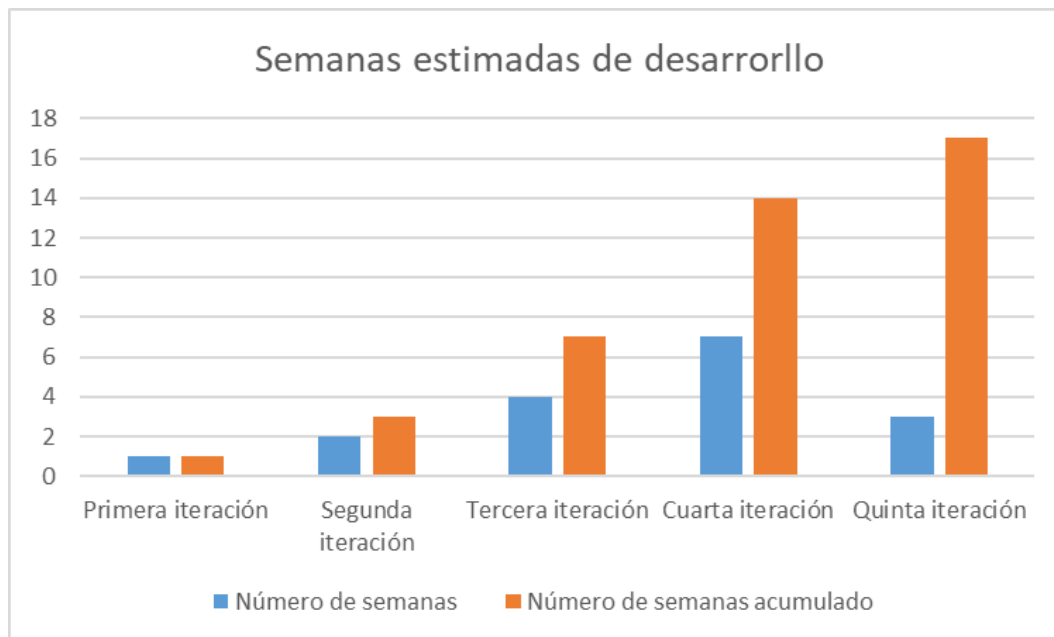


Figura 2.1: Distribución de las iteraciones según sus semanas de desarrollo.

2.2. Entorno tecnológico

En esta sección se enumeran las herramientas utilizadas para el desarrollo, tanto el software como los equipos.

2.2.1. Herramientas utilizadas

A continuación, se listan los programas y utilidades empleadas:

- **Angular [7]**: Herramienta basada en TypeScript utilizada para el desarrollo del proyecto.
- **Astah [8]**: Creación de diagramas UML.
- **BabelEdit [9]**: Herramienta para facilitar la traducción del proyecto.
- **Balsamiq Wireframes [10]**: Herramienta para la maquetación inicial de las pantallas.
- **Docker [11]**: Plataforma utilizada para facilitar el despliegue de la aplicación.
- **Git for Windows [12]**: Control de versiones del proyecto.
- **GitLab [13]**: Repositorio web del proyecto.
- **Google Chrome [14]**: Navegador opción de vista para móvil.

Planificación

- **Jitsi Meet [15]**: Herramienta de videollamada utilizada para la realización de las reuniones con los tutores.
- **Microsoft Excel [16]**: Para apuntar las horas invertidas cada día y en qué tareas.
- **Nginx [17]**: Servidor para la aplicación.
- **Node.js [18]**: Herramienta JavaScript utilizada para la compilación del proyecto, tanto durante el desarrollo como para el despliegue final.
- **NPM [19]**: Gestor de paquetes para Node.js que permite instalar paquetes de forma mas sencilla.
- **Outlook [20]**: Servicio de correo electrónico utilizado para la comunicación con los tutores.
- **Overleaf [21]**: Editor LaTeX en la nube.
- **Swagger [22]**: Herramienta para la documentación de las API y generación de código relacionado con la misma.
- **Visual Studio Code [23]**: IDE para desarrollo.
- **Windows Notepad [24]**: Para tomar apuntes durante las reuniones con los tutores.

Librerías relevantes utilizadas:

- **Bootstrap [25]**: Conjunto de herramientas para el diseño de sitios y aplicaciones web.
- **ngx-translate [26]**: Librería utilizada para la internacionalización.
- **Recordrtc [27]**: Librería utilizada para la grabación de audio.
- **rxjs [28]**: Librería de programación reactiva basada en eventos y Observables.
- **Voice Activity Detection [29]**: Librería implementada en JavaScript utilizada para la implementación del Voice Activity Recognition.

2.2.2. Entorno de trabajo

En las tablas 2.1 y 2.2 se describen los equipos utilizados a lo largo del proyecto, tanto para el desarrollo como para las pruebas de la aplicación, y en la tabla 2.3 se describe la máquina virtual utilizada para el despliegue en el servidor.

Ordenador portátil	
Asus ROG Zephyrus M16	
Hardware	
Procesador	Intel Core i9-12900H a 2.5GHz
RAM	32GB
SSD	1TB
GPU	NVIDIA GeForce RTX 3070 Ti
Software	
Sistema operativo	Windows 11 Home
Arquitectura	64 bits

Tabla 2.1: Ordenador utilizado en desarrollo.

Smartphone	
Samsung galaxy s10e	
Hardware	
Procesador	Samsung Exynos 9820 (Octa-core, 8 nm)
RAM	6GB
Almacenamiento	128GB
GPU	Samsung Exynos 9820
Pantalla	142,2,x 69,9 x 7,9 mm
Software	
Sistema operativo	Android 12
Arquitectura	64 bits

Tabla 2.2: Smartphone utilizado en desarrollo.

Servidor	
Maquina virtual	
Hardware	
Procesador	Intel KVM proccessor
RAM	8GB
HDD	12GB
Software	
Sistema operativo	Ubuntu 22.04.2 LTS
Arquitectura	64 bits

Tabla 2.3: Servidor utilizado para el despliegue.

2.3. Estimación de riesgos

En esta sección se van a enumerar los posibles riesgos que pueden afectar al correcto desarrollo del proyecto. Además de la descripción se incluye la probabilidad de que ocurra (en tanto por uno) y los días de retraso que podría acarrear.

Por otro lado, en la tabla 2.5 se muestran los planes de acción a llevar a cabo para minimizar el impacto de cada uno de los riesgos en caso de que se cumplan. También se muestra la exposición al riesgo, calculada al multiplicar la probabilidad de que el riesgo suceda por el retraso que se estima que provocaría.

ID	Riesgo	Probabilidad	Descripción	Retraso
R01	Enfermedad	0,2	Una enfermedad imposibilita al desarrollador trabajar durante un tiempo	5
R02	Falta de experiencia con las librerías utilizadas	0,5	Dificultad en el desarrollo causada por el desconocimiento de las librerías utilizadas	10
R03	Falta de experiencia con el tratamiento de audio	0,6	Dificultad en el desarrollo causada por el desconocimiento en cuanto al tratamiento de audio	10
R04	Planificación inadecuada	0,7	Error al estimar riesgos o tiempos de desarrollo	15
R05	Problemas con el equipo informático	0,2	Fallo del algún dispositivo utilizado para desarrollar el proyecto	2
R06	Modificación de los requisitos	0,5	Cambio en los requisitos que implique rehacer parte del proyecto, o que aumenten su escala	7

Tabla 2.4: Tabla de riesgos.

ID	Riesgo	Exposición al riesgo	Plan de acción
R01	Enfermedad	1	Hacer horas extras una vez finalizado el transcurso de la enfermedad
R02	Falta de experiencia con las librerías utilizadas	5	Buscar en la documentación y hacer horas extra para recuperar el tiempo perdido, o cambiar de librerías en caso de no ser capaz de encontrar una solución
R03	Falta de experiencia con el tratamiento de audio	6	Buscar en la documentación y hacer horas extra para recuperar el tiempo perdido
R04	Planificación inadecuada	10,5	Rehacer la planificación y hacer horas extras
R05	Problemas con el equipo informático	0,4	Utilizar otro ordenador
R06	Modificación de los requisitos	3,5	Programar teniendo en cuenta que puede haber cambios, haciendo código escalable

Tabla 2.5: Tabla de planes de acción para los riesgos.

2.4. Estimación de costes

En esta sección se va a realizar una estimación de los costes del proyecto. Estos costes incluyen el salario bruto del desarrollador, los costes de los equipos y licencias. Todas estas cantidades están contabilizadas con IVA.

2.4.1. Salario del trabajador

Ya que este proyecto se ha realizado para una asignatura, en concreto al de TFG, se estimará el coste para un único desarrollador junior. Para realizar esta estimación se tomará el sueldo medio para un programador junior en España, que según glassdoor [30] es de 19.000 euros anuales. Suponiendo que el número total de horas de trabajo a lo largo del año sean 1.800, el salario bruto resultante de esta estimación será de 10,55€/h. Como para la realización del proyecto han sido necesarias 350h, el coste procedente del salario del trabajador será de 3692,50€.

2.4.2. Costes indirectos aplicados al espacio de trabajo

El proyecto se ha realizado de forma íntegra en el domicilio personal del trabajador. Partiendo desde esto como base, el cálculo de gastos se realizará estimando el coste de la vivienda y los gastos complementarios de la misma. Tomando en cuenta tanto el tamaño como la zona, el precio del alquiler en las condiciones actuales del mercado resulta en 550 euros al mes, lo cual teniendo en cuenta un tiempo de desarrollo medio de 3h/día, resultan en un coste de 343,75€.

A estos gastos hay que añadir los gastos necesarios para el correcto desarrollo del proyecto, estos son principalmente la luz y el internet, teniendo en cuenta el número de horas de trabajo, el precio de la factura del internet (30€ al mes) y el precio del KW/h (que oscila entre los 0,12€ y los 0,24€ dependiendo de la hora. Puesto que el desarrollo se ha realizado durante distintos horarios, incluyendo tanto el más caro como el más barato, se han tomado 0,18€ de valor medio), resultan en un coste aproximado de 200€. Este valor sumado al coste asociado al alquiler resultan en un total de 543,75€.

2.4.3. Costes del hardware

Para la realización de la estimación de los costes asociados a los equipos utilizados, se tendrá en cuenta el coste en el momento de compra, junto con la proporción entre el número de horas de uso de cada equipo para la realización del proyecto y el número de horas de uso totales del mismo.

Planificación

Equipo	Coste en el momento de compra	Nº de horas del equipo	Nº de horas de uso	Coste total
Portátil	2000€	2.000	350	350€
Smartphone	500€	9.500	350	18,42€
Total				368,42€

Tabla 2.6: Tabla de estimación de costes del hardware.

2.4.4. Costes del software y servicios

Aunque varias de las herramientas utilizadas son de pago, la mayoría de las mismas contaban con planes gratuitos que permiten utilizar suficiente funcionalidad como para realizar el desarrollo del proyecto, u ofrecen periodos de prueba lo suficientemente largos como para poder realizar todas las tareas para las cuales eran necesarias dichas herramientas. Sin embargo, no todas contaban con estas ventajas, resultando en los siguientes gastos:

2.4.4.1. Astah

El coste de Astah professional para una única persona es de 7,50€ al mes. Solo fue necesario su uso durante un mes, por lo que el coste es de solo 7,50€.

2.4.5. Costes totales

En la siguiente tabla se puede ver la estimación del coste total del proyecto, junto al de cada una de sus partes.

	Coste
Salario desarrollador	3692,50€
Espacio de trabajo	543,75€
Hardware	368,42€
Software y servicios	7,50€
Total	4612,17€

Tabla 2.7: Tabla de estimación de costes totales.

2.5. Planificación final

La planificación final ha sufrido ligeros cambios respecto a la planificación estimada. En la figura 2.2 se puede observar la comparación entre el número de semanas acumuladas estimadas y el número real, indicando como las 17 semanas planeadas inicialmente se acabaron alargando a causa de diversos retrasos hasta llegar a 20. A lo largo de estas 20 semanas se han realizado un total de 12 reuniones con los tutores, utilizadas para recibir feedback y planificar más en detalle dentro de cada iteración, para reducir el número de retrasos al máximo.

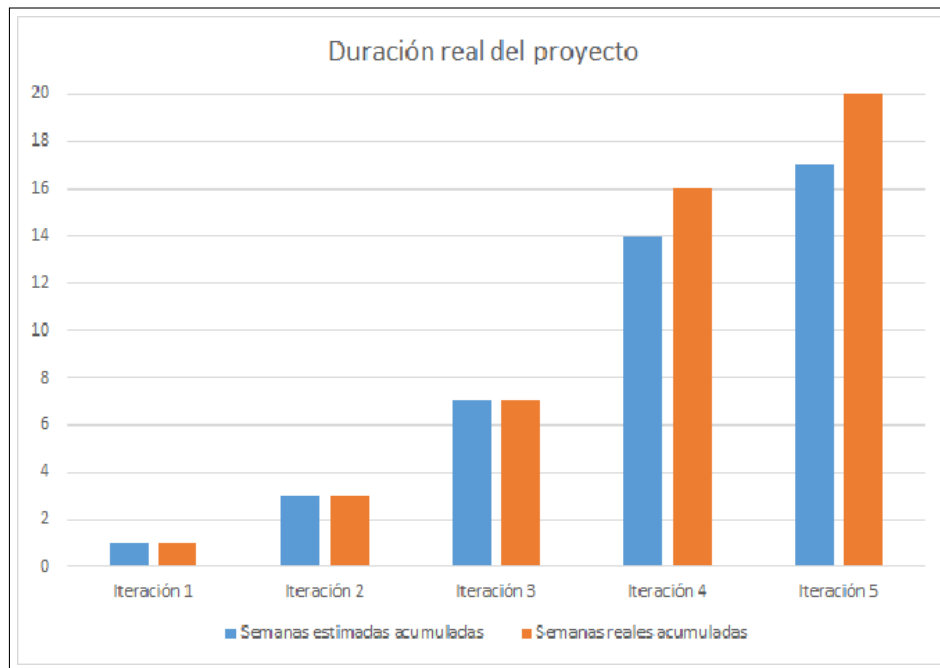


Figura 2.2: Comparación de la estimación original con la duración real.

Este aumento en la duración del proyecto también se ha visto reflejado en la cantidad de horas invertidas en la realización del proyecto. En total se han llevado a cabo 350 horas de trabajo, que superan con creces las 300 horas mínimas que son necesarias para la realización de un TFG. Dichas horas se han distribuido de la siguiente forma a lo largo de las iteraciones:

- 1ª Iteración: 1 semana - 10 horas
- 2ª Iteración: 2 semanas - 25 horas
- 3ª Iteración: 4 semanas - 75 horas
- 4ª Iteración: 9 semanas - 155 horas
- 5ª Iteración: 4 semanas - 85 horas

En cuanto a las razones por las cuales no se ha cumplido con la planificación estimada inicialmente, los diversos retrasos han sido a consecuencia de varios de los riesgos indicados en el apartado 2.3. Estos requisitos, en un orden de mayor a menor importancia son: R06,R04,R02 y R03. Además, estos problemas aparecieron principalmente en la cuarta iteración, lo que provocó que no tuviera suficiente tiempo como para compensar el tiempo perdido, provocando un retraso sustancial en la fecha final del

proyecto.

Durante las dos primeras iteraciones no se manifestó ninguno de los riesgos, lo cual era esperable al tratarse principalmente de pruebas con las tecnologías, aunque hubiera otras tareas como la planificación inicial, o la maquetación de las pantallas.

Ya durante la tercera iteración se manifestó el primero de los riesgos en afectar al correcto desarrollo del proyecto, este fue el riesgo R03, es decir, dificultad en el desarrollo causada por la falta de experiencia en cuanto al tratamiento de audio. A pesar de haber realizado pruebas durante la segunda iteración, se tardó más de lo esperado en implementar algunas de las funcionalidades, como separar el audio en trozos más pequeños para poder enviarlos al servidor. Como ya se ha mencionado antes, este riesgo fue el que tuvo menor repercusión, de unos tres días. Aplicando el plan de acción asociado se pudo volver a encauzar el proyecto.

El siguiente riesgo en aparecer fue también el siguiente con menos repercusión, es decir, el R02. Este riesgo, que hace referencia a la falta de experiencia con las librerías utilizadas, apareció durante el final de la tercera iteración y la mayoría de la cuarta. Los problemas asociados a este riesgo tuvieron que ver principalmente con la implementación de los timers, que se mencionan varias veces en los apartados 4.3 y 4.4. Esto provocó no solo tener que cambiar que librería se utilizaba para la implementación de los mismos hasta 3 veces, sino que también provocó retrasos de aproximadamente una semana.

El siguiente en aparecer fue con diferencia el más problemático de todos, el R06, es decir, la modificación de requisitos. El principal problema con este riesgo es que apareció al final de la cuarta iteración, dejando muy poco margen para arreglar los problemas causados por el mismo. Si bien no hubo ningún cambio sustancial de los requisitos establecidos durante las primeras semanas del proyecto, se crearon dos nuevos requisitos funcionales, explicados más en detalle en el apartado 4.4.

El primero de los requisitos en aparecer fue el que trata sobre mostrar el progreso del envío de audio al servidor y obtención de la transcripción del mismo, este cambio por sí solo fué el menor de los problemas de este riesgo, causando un retraso de solo 2 días.

El otro requisito que se añadió fue el que trata de dar la posibilidad al usuario de cambiar el idioma de la página sobre la marcha, lo supuso un gran problema, pues implicaba tener que elegir entre añadir un servidor como firebase que se encargara de servir las distintas versiones de la página al usuario (lo cual añadiría complejidad al proyecto, y podría complicar el despliegue), o rehacer de cero la internacionalización de la página utilizando una librería que permitiera cumplir el requisito si necesidad de programas o servidores externos. Ambas opciones iban a suponer una gran pérdida de tiempo, por lo cual se eligió la segunda, que no aumentaba la complejidad del proyecto, esto provocó un retraso de cerca de una semana.

Asociados a los cambios a la interfaz necesarios para cumplir con los 2 anteriores requisitos, y al feedback recibido de los tutores, fue necesario también un rediseño todas las pantallas, y aunque algunas solo tenían pequeños cambios como el inicio de sesión y el registro, otras tuvieron grandes modificaciones, como la página principal. Aunque este rediseño no es un cambio en los requisitos

propiamente dicho, es en gran parte causa de los mismos, por lo que he decidido agrupar el retraso causado por este rediseño junto al de los nuevos requisitos, añadiendo una semana más al total.

Sumando todos los retrasos causados por el riesgo R06 se obtiene una suma de 17 días, lo cual es la razón por la que he indicado el riesgo R04 (Planificación inadecuada) como otro de los importantes. Esto se centra más en que la estimación de los riesgos no ha sido correcta. Esto es muy evidente tomando como ejemplo el propio Riesgo R06, que tenía una posibilidad estimada de aparecer de un 50 %, y sin embargo ha aparecido dos veces, y un retraso estimado de 7 días, y si embargo ha causado 17. A esto se le puede añadir que el plan de acción asociado al mismo no ha sido óptimo, puesto que no ha sido capaz de compensar o disminuir estos retrasos. Si bien la mala planificación no ha causado retrasos en sí, ha supuesto que las consecuencias de los anteriores hayan sido mayores de las que habrían sido con una mejor planificación.

También se puede ver que además de los retrasos indicados anteriormente, ha aparecido uno nuevo durante la quinta iteración. Este retraso ha sido el resultado de un problema que no se previó en los riesgos iniciales: los problemas tras el despliegue.

Estos problemas se explicarán en mas detalle en el apartado correspondiente a la quinta iteración, pero el resumen es que hasta el momento del despliegue la aplicación solo se había probado de forma local, para lo cual no se veía ningún problema. Sin embargo al desplegar se vio que la parte de grabar audio no funcionaba, esto era debido a que los navegadores modernos no permiten grabar audio a través de HTTP por motivos de seguridad. Esto provocó un retraso de 4 días cambiando la aplicación para que funcionara a través de HTTPS. Si bien este fallo se podría pensar en un principio que podría caer bajo el riesgo R03 (falta de experiencia con el tratamiento de audio), la mayoría del retraso viene de necesitar tiempo para aprender a desplegar la aplicación de modo que permita HTTPS.

La última conclusión que se puede sacar de comparar la planificación inicial con la final es que se ha sido demasiado optimista en cuanto a la estimación de la cantidad de horas asignadas para cada iteración, especialmente en la última.

Capítulo 3

Estado de la cuestión

Como se ha mencionado en la introducción, en este TFG se parte de la base de un servicio web que ofrece ASR en tiempo real [4]. Esta API está formada por 7 endpoints (direcciones de la API que se encargan de dar respuesta a las distintas peticiones) con la siguiente funcionalidad:

1. Subir un archivo de audio desde el almacenamiento local.
2. Subir un archivo de audio desde la nube al servidor.
3. Obtener al transcripción de un archivo de audio, pudiendo especificar algunos parámetros, como el idioma o la intensidad (que proporciona resultados más precisos a cambio de unos mayores tiempos de respuesta).
4. Obtener un listado de todos los archivos de audio de un usuario, junto a sus respectivas transcripciones.
5. Posibilidad de borrar los datos de audio tras obtener la transcripción, de acuerdo con el GDPR (Reglamento General de Protección de Datos) [31].
6. Iniciar sesión.
7. Registro, con confirmación por correo electrónico.

Este servicio web está inspirado en el TFG de Mario Panadero Palazuelos [32] en el hecho de que se trabaja en una aplicación educativa con servicios API con el fin de mejorar la pronunciación, en el de Juan García Diéguez [33] por utilizar una API REST en el contexto de las aplicaciones educativas centradas en la mejora de la pronunciación, y en el de Antonio Gamazo Ferrero [34] en el desarrollo de una aplicación educativa multiplataforma con el objetivo de ayudar a mejorar la pronunciación mediante un frontend móvil y un backend web mediante API. Además, también está inspirado en los artículos [35, 36, 37] por el hecho de que se utilizan APIs de terceros en tiempo real para obtener la transcripción de habla en aplicaciones educativas para mejora de la pronunciación. En este caso, se desea tener un servidor propio para poder gestionar los datos sensibles de la voz de los usuarios de manera privada.

La API utilizada en este TFG tiene una serie de características y cualidades que será útil tener en cuenta a lo largo del desarrollo. Las más relevantes son la posibilidad de elegir el modelo ASR (de entre un gran número de idiomas) y la intensidad sobre la marcha, su baja latencia que permitirá obtener la transcripción en tiempo real, su compatibilidad con cualquier dispositivo gracias uso del protocolo HTTP, y su seguridad, que utiliza jsonwebtoken (JWT) para la autenticación de los usuarios y un sistema de tickets para limitar la cantidad de peticiones que estos pueden realizar cada hora.

3.1. Otras tecnologías ASR en el mercado

El servicio descrito anteriormente no es el único que implementa una API que hace uso de la tecnología ASR. De hecho, existen varias opciones comerciales o open source que tiene una funcionalidad similar.

3.1.1. Google Speech-to-Text

Google STT [38] es la opción más popular, y permite convertir la voz a texto de forma precisa gracias a la tecnología de IA de Google. Entre sus principales ventajas se encuentran la variedad de dominios especializados en diferentes funciones, como recibir comandos por voz, o realizar transcripciones, la capacidad de ejecutarse de forma local, sin necesidad de acceso a internet, y la variedad de idiomas que soporta, llegando a superar los 125.

Como contrapartida, esta API de google no es gratuita, y cobra por cada minuto de audio [39], teniendo acceso a los 60 primeros de cada mes de forma gratuita. Este coste puede ser pequeño para aplicaciones con un número pequeño de usuarios, pero según aumenta la cantidad de personas utilizando la aplicación, y por lo tanto la cantidad de tiempo de audio, el precio se dispara.

3.1.2. Kaldi

Kaldi [40] es un software open source escrito en C++ y lanzado bajo la licencia pública de Apache, se puede utilizar de forma gratuita y es altamente personalizable, pudiendo extender o modificar su funcionalidad.

El principal problema de Kaldi es su escasez en cuanto a modelos ya entrenados para diferentes idiomas, en su página de modelos [41] se puede ver que solo cuentan con 14, y que solamente están entrenados para inglés o mandarín. Además, este software está orientado a expertos, por lo que requiere un gran conocimiento en el campo de los sistemas ASR para poder utilizarlo, aunque se utilicen los modelos disponibles para descargar.

3.1.3. Modelos ASR End2end

Un tipo de modelos que se está volviendo cada vez más popular en el ámbito del ASR son los modelos End2end (extremo a extremo), que son sistemas que transforman la información de entrada en información de salida utilizando un sistema de aprendizaje neuronal único. Estos sistemas utilizan Deep Learning para aprender a asociar los sonidos con sus correspondientes transcripciones. Algunos de los más relevantes son:

3.1.3.1. Whisper

Whisper [42] es un sistema opens source de ASR propiedad de openAI. Su principal característica es su robustez, ya que ha sido entrenado con una gran cantidad de datos (680.000 horas). Es capaz de realizar transcripciones en varios idiomas, además de traducir dichos idiomas al inglés.

3.1.3.2. wav2vec2.0

Wav2vec2.0 [43] es un modelo ASR muy popular, que fue lanzado en septiembre de 2020 por Meta AI, este modelo fue entrenado utilizando CTC (Connectionist Temporal Classification), y su principal ventaja es que no necesita que se utilice un modelo de lenguaje o diccionario externo para generar transcripciones de audio, aunque se puede utilizar para obtener mejores resultados. Además, cuenta con un modelo multilingüaje [44], que soporta 56 idiomas distintos.

3.2. Entorno y herramientas tecnológicas

En esta sección se exponen las tecnologías implicadas en el desarrollo de este TFG y sus características principales. Como breve resumen, para el desarrollo del frontend se se ha utilizado Angular, que necesita de Node.js para la compilación. Para facilitar la instalación de librerías se ha utilizado NPM, y para hacer el despliegue se ha utilizado Docker, el cual en este caso hace uso de imágenes tanto de Node como de Nginx.

Además, a modo de apoyo se han utilizado otras tecnologías: Visual Studio Code como IDE, Overleaf para redactar la memoria, Gitlab para alojar el repositorio git, Balsamiq para realizar la maquetación inicial de las pantallas, BabelEdit para facilitar la traducción y Swagger para ver y probar la API del servidor, así como generar código correspondiente a la misma de forma automática.

3.2.1. Angular

Angular [7] es un framework para el diseño de aplicaciones y una plataforma de desarrollo construida en TypeScript pensada para crear aplicaciones web eficientes y sofisticadas. Las aplicaciones Angular están formadas principalmente por componentes, formados por una clase TypeScript (.ts), un selector de CSS que define como se usa el componente en las plantillas, una plantilla HTML que indica como se debe renderizar el componente y de forma opcional una serie de hojas de estilo CSS, que cambian la apariencia de los elementos de la plantilla HTML.



Figura 3.1: Logo de Angular.

En este proyecto se ha utilizado para desarrollar el frontend por completo, con la ayuda de algunas librerías que se explicarán en más detalle a lo largo de las iteraciones, cuando vayan siendo relevantes.

3.2.2. Node.js

Node.js [18] es un entorno de ejecución de JavaScript orientado a eventos asíncronos, diseñado para crear aplicaciones network escalables. Permite atender muchas conexiones de forma simultánea, activando la devolución de llamada (callback) para cada una de ellas, y durmiéndose si no tiene más trabajo que realizar.



Figura 3.2: Logo de Node.js.

En este proyecto, se utiliza principalmente para facilitar la construcción de la aplicación angular, tanto para poder usar npm, que facilita la instalación de paquetes como para el CLI (Interfaz de Línea de Comandos) de Angular. Además, en el Dockerfile se utilizará una imagen de Node 18 para compilar el proyecto.

3.2.3. NPM



Figura 3.3: Logo de NPM.

NPM [19] (node package manager) es un gestor de paquetes para Node.js creado en 2009 como un proyecto Open Source. Consiste de una gran colección de paquetes y aplicaciones, y una serie de comandos que permite instalar y publicar dichos paquetes. Su principal utilidad en este proyecto será la de ayudar a instalar las diferentes librerías que se acaben usando durante el desarrollo.

3.2.4. Docker



Figura 3.4: Logo de Docker.

Docker [11] es una plataforma que permite desarrollar, transportar y ejecutar aplicaciones. La principal ventaja de Docker es que permite separar las aplicaciones de la infraestructura, permitiendo un despliegue fácil y rápido. Esto se consigue gracias a la posibilidad de empaquetar las aplicaciones en entornos aislados, llamados contenedores. Estos contenedores tienen varias ventajas:

- La posibilidad de ejecutar varios contenedores a la vez en la misma máquina, lo cual permite por ejemplo, tener un contenedor que contenga el backend de una aplicación, y otro que contenga el frontend.
- Son ligeros, por lo que no consumen demasiados recursos.

- Contienen todo lo necesario para ejecutar la aplicación, por lo que no necesitan de instalaciones extras además del propio Docker.
- Se pueden compartir de forma sencilla.

3.2.5. Nginx

Nginx [17] es un software que principalmente hace las funciones de un servidor web, aunque tiene otras funcionalidades como el balanceo de cargas, y la capacidad de servir como un servidor proxy para correo electrónico. Una de las características principales con las que fue diseñado es el alto rendimiento, lo cual ha aumentado en gran medida su popularidad desde su lanzamiento en 2004.



Figura 3.5: Logo de Nginx.

En este proyecto se utilizará como servidor web dentro del contenedor de Docker que contenga la aplicación final, y para configurar los certificados SSL que permitan acceder a dicha aplicación a través de HTTPS.

3.2.6. Visual Studio Code



Figura 3.6: Logo de VSC.

Visual Studio Code [23] (VSC) es el IDE con el que se va a desarrollar el proyecto, en parte por que el alumno ya tiene experiencia utilizando este programa, pero además por la gran cantidad de extensiones de las que dispone, además de ser rápido, funcionar con una gran cantidad de lenguajes de programación y archivos de distintas extensiones (por ejemplo, es capaz de reconocer la estructura del Dockerfile), además de tener otras herramientas, como una terminal que permite lanzar comandos como los que se utilizan para instalar librerías, compilar el proyecto Angular y crear imágenes de Docker.

3.2.7. GitLab



Figura 3.7: Logo de GitLab.

Gitlab [13] es el repositorio git que se ha utilizado en este proyecto, los repositorios git son útiles tanto para mantener versiones de nuestro proyecto accesibles, como para compartirlo y que otros usuarios puedan tanto verlo como descargarlo, ya sea para ejecutarlo o para participar en el desarrollo del mismo.

El principal objetivo en este proyecto será el de compartir el proyecto de forma sencilla y asegurando que se trata de la última versión, ya sea con los tutores, o con el servidor donde se va a alojar para poder desplegarlo.

3.2.8. Balsamiq Wireframes



Figura 3.8: Logo de Balsamiq.

Balsamiq Wireframes [10] es una herramienta que permite hacer bocetos rápidos para diseñar la interfaz de aplicaciones, tanto móviles como para ordenador, sus principales ventajas son que es muy sencillo y rápido, por lo que evita la necesidad de gastar mucho tiempo durante el proceso de maquetación de las pantallas, y su facilidad de uso, lo que permite utilizarlo sin necesidad de experiencia previa ni tutoriales.

3.2.9. Swagger



Figura 3.9: Logo de Swagger.

Swagger [22] se compone de una serie de herramientas para desarrolladores de APIs, las partes más relevantes para este proyecto son:

- **documentación:** Es la herramienta que se usó tanto para explicar el funcionamiento de la API antes de comenzar a usarla, como para solucionar dudas sobre el funcionamiento de la misma durante el desarrollo.
- **Probar la API:** Swagger permite probar las llamadas de la API, lo cual es útil para ver las respuestas del servidor para una llamada real. Además se utilizó durante las primeras iteraciones para hacer las llamadas que pudieran ser necesarias y que no estaban implementadas todavía en el código (Para más información ver el apartado Iteraciones).

- **Swagger editor [45]:** Esta es una herramienta muy poderosa, con varias funcionalidades, pero la más relevante para este proyecto es la capacidad de generar el código necesario para utilizar la API en una gran variedad de lenguajes de programación, tanto para el cliente como para el servidor, aunque en este caso solo fue necesario utilizarlo para el cliente.

3.2.10. BabelEdit



Figura 3.10: Logo de BabelEdit.

BabelEdit [9] es una herramienta diseñada para encargarse del manejo de los archivos de traducción, aunque funciona con archivos de varias extensiones, en este proyecto solo es utilizará para archivos .json, proporciona una interfaz para editarlos de forma sencilla, además de incorporar el uso de varios traductores online para agilizar el proceso de traducir.

3.2.11. Overleaf



Figura 3.11: Logo de Overleaf.

Overleaf [21] es un editor de LaTeX online, fácil de usar y que permite la colaboración de varias personas en tiempo real, permitiendo compilar de forma fácil y rápida para ver el resultado a medida que se va progresando en la redacción. Ha sido la herramienta utilizada para la redacción de la memoria.

Capítulo 4

Descripción de las iteraciones

En este capítulo se expone el trabajo realizado durante las diferentes iteraciones a lo largo del desarrollo del proyecto. Entre otras cosas, se van a comentar los requisitos del proyecto, las pruebas realizadas con las tecnologías, el prototipo implementado, el diseño de las pantallas, y por último, de su implementación.

4.1. Iteración 1

Para iniciar el proyecto, se produce una reunión del desarrollador con el cliente, en este caso, el tutor del TFG. Esta primera reunión ocurrió el día 14 de febrero de 2023 y marca el inicio del proyecto, durante la misma se exponen las características del proyecto de manera general, y se establecen una serie de pasos a seguir durante el desarrollo del mismo.

Tras esta reunión, ya con una idea general de la dimensión del proyecto, así como de las funcionalidades a implementar, se definen las iteraciones en las que se dividirá el proyecto, y se realiza una estimación de su duración. Esta planificación se puede ver detalladamente en la sección 2.1 de la memoria.

Por otro lado, también se seleccionan las tecnologías que se planean utilizar para el desarrollo (Angular y Docker), y se realiza la instalación de todo lo necesario para poder utilizarlas. Además, se realizan unas pequeñas pruebas para comprobar que dichas tecnologías permiten implementar toda la funcionalidad indicada en la reunión inicial. Estas pruebas incluyen la creación de un proyecto 'Hola Mundo' y ejecutarlo desde Docker, así como comprobar que es posible grabar audio utilizando Angular.

4.2. Iteración 2

En esta segunda iteración se continúa la familiarización con las tecnologías elegidas para la realización del proyecto. Para esto, se realizarán unos pequeños programas relacionados con la grabación del audio. El objetivo de estas pruebas es por un lado adquirir los conocimientos necesarios para iniciar el desarrollo del proyecto, y por otro, encontrar y probar las librerías que se van a utilizar tanto para grabar el audio, como para implementar el VAR (Voice Activity Recognition). Además, se realizará la maquetación inicial de todas las pantallas, y se especificarán los requisitos de forma más precisa.

4.2.1. Descripción de requisitos

Al inicio de esta iteración, y coincidiendo con el final de la anterior, se realizó una reunión con el tutor. En dicha reunión no solo se revisó el resultado de la iteración anterior y se planificó esta, sino que además se indicaron de forma mucho más precisa los requisitos de la aplicación.

4.2.1.1. Requisitos funcionales

A continuación se describen los diferentes requisitos funcionales del proyecto. Estos requisitos describen la funcionalidad que debe cumplir el sistema que se va a implementar. En la tabla 4.1 se muestran estos requisitos con su descripción y su prioridad.

Descripción de las iteraciones

ID	Nombre	Descripción	Prioridad
RF01	Registro de un usuario	El sistema permitirá crear nuevos usuarios	Alta
RF02	Identificación de un usuario	El sistema permitirá que los usuarios existentes se identifiquen	Alta
RF03	Cerrar sesión	El sistema permitirá que los usuarios cierren la sesión	Alta
RF04	Grabar audio	El sistema permitirá grabar audio y enviarlo al servidor para obtener su transcripción	Alta
RF05	Elegir idioma	El sistema permitirá elegir el idioma para el cual obtener la transcripción	Alta
RF06	Nombre de la grabación	El sistema permitirá ponerle nombre a la grabación	Alta
RF07	Elegir intensidad	El sistema permitirá cambiar el nivel de intensidad del servidor	Alta
RF08	Elegir guardar audio	El sistema permitirá si se guarda el audio en el servidor	Media
RF09	Limpiar resultado	El sistema permitirá al usuario limpiar la transcripción	Alta
RF10	Mostrar volumen	El sistema mostrará al usuario el nivel de audio que está recibiendo	Alta
RF11	Mostrar historial	El sistema permitirá mostrar una tabla con el histórico del usuario	Media
RF12	Filtros	El sistema permitirá filtrar los datos del historial	Alta
RF13	Descarga historial	El sistema permitirá al usuario descargar su historial	Media

Tabla 4.1: Tabla de requisitos funcionales.

4.2.1.2. Requisitos no funcionales

A continuación, se procede a describir los requisitos no funcionales. Estos definen las características técnicas del sistema. En la tabla 4.2 se muestran con una estructura similar a la utilizada para los funcionales solo que en este caso se indica la relevancia, en una escala de "Crítica", "Deseable" o "Baja".

ID	Nombre	Descripción	Relevancia
RNF01	Multiplataforma	El sistema se podrá utilizar en cualquier plataforma que pueda utilizar Docker	Crítica
RNF02	Formato de descarga historial	El sistema utilizará un formato JSON para la descarga del historial	Deseable
RNF03	Formato audio	El sistema utilizará el formato .wav para los audios	Crítico
RNF04	Traducción	El sistema deberá ser sencillo de traducir a otros idiomas (i18n), y deberá contar con traducciones al inglés estadounidense y al español de España	Crítico
RNF05	Notificación de errores	El sistema informará al usuario por pantalla en caso de que éste cometa algún error, como introducir unas credenciales de inicio de sesión erróneas	Crítica
RNF06	Facilidad de instalación	Se debe poder instalar el sistema siguiendo los pasos del manual de instalación	Deseable
RNF07	Diseño modular	El sistema cumplirá los estándares del responsive design para su correcta visualización en dispositivos con pantallas pequeñas (como móviles y tablets)	Deseable
RNF08	Voice activity recognition	El sistema debe detectar cuando el usuario para de hablar para enviar el audio al servidor	Crítica
RNF09	Tiempo límite de audio	El sistema deberá tener un límite de tiempo de 5s, a partir del cual el audio se enviará al servidor automáticamente	Crítica
RNF10	Tiempo límite de silencio	El sistema deberá tener un límite de tiempo de 0,7s desde que el usuario pare de hablar, a partir del cual el audio se debe enviar al servidor automáticamente	Crítica
RNF11	Límite de caracteres	El resultado tendrá un límite de 500 caracteres	Deseable

Tabla 4.2: Tabla de requisitos no funcionales.

4.2.1.3. Requisitos de información

El último tipo de requisitos que falta por mencionar son los requisitos de información, que son aquellos que especifican cuales son los datos que se van a guardar en el sistema. Debido a la naturaleza del proyecto (crear un frontend para un servidor backend ya existente) no va a ser necesario guardar ningún dato, sino que será el servidor el que guarde o no, y con el formato que decida, los datos que el enviemos.

4.2.2. Pruebas de concepto

Tanto las pruebas realizadas durante esta iteración, como el proyecto definitivo se realizaron utilizando *Visual Studio Code* [23], y partieron de la base de las pruebas de la primera iteración. Tras probar varias librerías de grabación de audio, al final se decidió que la más indicada para la realización del proyecto era *recordrtc* [27], principalmente por su popularidad, lo cual facilitará encontrar soluciones a los problemas que puedan aparecer más tarde. esta popularidad se puede comprobar en páginas como *npm trends* [46] o *snykAdvisor* [47]:

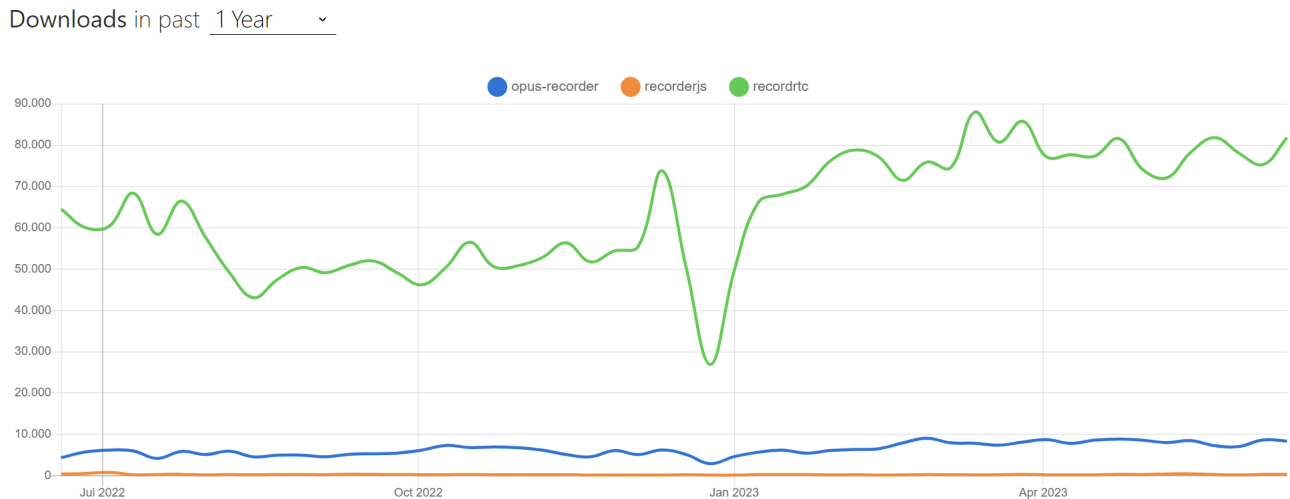


Figura 4.1: opus-recorder vs recorderjs vs recordrtc.

Tras escoger una librería de grabación de audio, y de realizar una serie de pruebas para comprobar su funcionamiento, el siguiente paso fue escoger librería para el VAR, de las cuales se escogió *voice activity detection* (VAD). Según *npm trends* [48] tiene mejores estadísticas de uso que otras librerías similares, pero además su documentación junto con su ejemplo de uso en GitHub facilitaron en gran medida comprobar su funcionamiento, tanto individualmente, como en conjunto con *recordrtc*.

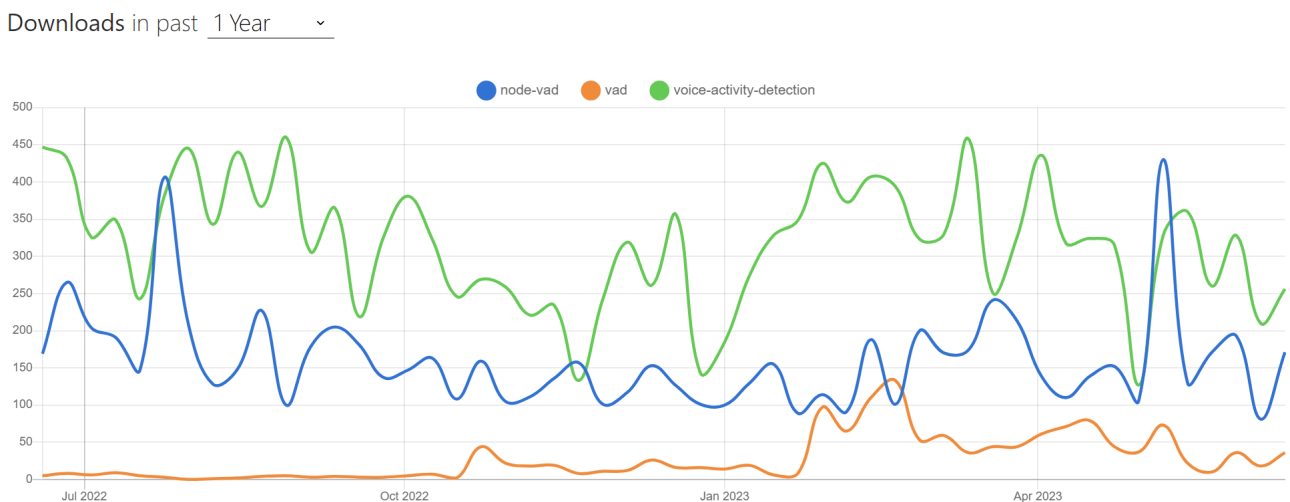


Figura 4.2: node-vad vs vad vs voice-activity-detection.

La última (y la más compleja) de las pruebas que se realizó consistía en un sencillo indicador que mostraba si se estaba hablando, junto con un número que indicaba el volumen que se estaba recibiendo en el micrófono, en una escala entre 0 y 1. Esta prueba nos permitió saber que era posible implementar el VAR, además de tener un modo de indicar el volumen al usuario.

4.2.3. Voice Activity Detection

Ya que se va a utilizar una librería externa para el VAR (que es una parte bastante relevante del proyecto) es necesario comprender bien su funcionamiento y todas las opciones de su configuración. Esto nos permitirá sacarle el mayor partido posible y obtener los mejores resultados.

VAD se compone de un único archivo escrito en JavaScript (.js), que exporta una única función, que creará el objeto VAD. Esta función recibe un objeto `AudioContext`, un stream de audio, y un por último, un parámetro con la configuración.

El objeto `AudioContext` representa un gráfico de procesamiento de audio, formado por módulos de audio, representados por objetos del tipo `AudioNode`. En cuanto al stream, es simplemente una secuencia de valores a lo largo del tiempo, que en nuestro caso contendrá el audio que se está grabando con el micrófono. Si bien ambos objetos se pueden configurar mas allá de los cambios que haga el propio VAD al aplicar su configuración, estos cambios no son especialmente relevantes, y están más relacionados con la parte correspondiente a la grabación de audio que con el propio VAD.

El tercer objeto, y el más importante de comprender es el que contiene la configuración. Los valores que añadamos a este objeto sustituirán a los de la configuración por defecto, manteniendo el resto, por lo que lo siguiente que debemos hacer es analizarla:

```
1 var defaults = {
2   fftSize: 1024,
3   bufferLen: 1024,
4   smoothingTimeConstant: 0.2,
5   minCaptureFreq: 85,           // in Hz
6   maxCaptureFreq: 255,         // in Hz
7   noiseCaptureDuration: 1000, // in ms
8   minNoiseLevel: 0.3,          // from 0 to 1
9   maxNoiseLevel: 0.7,          // from 0 to 1
10  avgNoiseMultiplier: 1.2,
11  onVoiceStart: function() {
12  },
13  onVoiceStop: function() {
14  },
15  onUpdate: function(val) {
16  }
17  };
```

Como se puede ver, se trata de una lista con los valores de algunas propiedades, y tres funciones vacías. Veamos qué es cada uno de estos valores:

- **minCaptureFreq/maxCaptureFreq:** Las frecuencias mínima y máxima para detectar audio, los

Descripción de las iteraciones

valores por defecto representan el rango de frecuencias de la voz humana, por lo que no ha sido necesario modificarlos.

- **noiseCaptureDuration:** El tiempo que tiene que pasar para empezar a contar un sonido como voz, para intentar evitar que el sonido ambiente se detecte como habla humana. Puesto que queremos que nuestra aplicación trabaje en tiempo real, se ha reducido a 200ms, de modo que se pueden seguir evitando pequeños ruidos en el ambiente, pero no haya un retraso sustancial desde que el usuario empieza a hablar hasta que se le empieza a dar feedback (en forma del indicador de volumen).
- **minNoiseLevel/maxNoiseLevel:** En una escala de entre 0 y 1, los niveles mínimo y máximo de ruido durante el transcurso de *noiseCaptureDuration*. Se ha reducido el valor mínimo a 0.2, porque durante las pruebas realizadas a lo largo del desarrollo se detectó que dependiendo del micrófono utilizado, un bajo volumen provocaba problemas.
- **avgNoiseMultiplier:** Es un multiplicador que se le aplica la nivel de ruido necesario para cambiar el estado entre activo e inactivo. No ha sido necesario modificarlo con respecto a su valor por defecto.
- **fftSize, bufferLen y smoothingTimeConstant:** Son distintas propiedades que se utilizan en la creación de un analizador (*AnalyserNode* [49]), en el caso de *fftSize* y *smoothingTimeConstant*, y de un procesador de scripts (*ScriptProcessorNode* [50]) en caso de *bufferLen*. Ambos a partir del *AudioContext* recibido como parámetro. *BufferLen* representa el tamaño del buffer de entrada y salida, *fftSize* el de la transformación de Fourier que se utiliza para calcular el dominio de las frecuencias, y por último *smoothingTimeConstant* [51] es un número que se utiliza para suavizar las transiciones entre valores. De estos tres campos solo se ha cambiado el último a 0.1, puesto que no necesitamos de ese suavizado.
- **onVoiceStart, onVoiceStop y onUpdate:** Son tres funciones que por defecto están vacías, puesto que su propósito es ser sobrescritas para añadir la funcionalidad que queramos. *onVoiceStart* se activará cuando se detecte que el usuario ha comenzado a hablar, *onVoiceStop* cuando pare, y *onUpdate* cada vez que llegue un valor del stream.

En cuanto al funcionamiento de la librería en sí, lo primero que hace es aplicar los valores de la de la configuración, después, cada vez que llega un valor del stream se utiliza el analizador previamente mencionado para obtener los datos de las frecuencias, se comprueba con contadores si ha pasado el tiempo suficiente para cambiar de estado (entre usuario hablando y usuario callado) y en caso de que sí, se cambia el estado y se llama a la función correspondiente (*onVoiceStart/onVoiceStop*), por último se llama a la función *onUpdate*.

Además, cuenta con funciones que permiten conectar y desconectar la instancia del VAD, así como destruirla cuando se haya terminado de usar.

4.2.4. Maquetación de las pantallas

Durante esta iteración se realizaron también los mock-ups de las 4 vistas que tendrá nuestra aplicación, para ello se utilizó la herramienta *Balsamiq wireframes* [10], obteniendo los siguientes resultados:

■ Inicio de sesión

Esta será primera página a la que acceda el usuario, y consiste de un inicio de sesión simple, con campos para usuario y contraseña, que permitirá visitar la página de registro, o iniciar sesión e ir a la principal:

Mock-up del inicio de sesión. El formulario contiene dos campos de texto: 'Usuario' y 'Contraseña'. Debajo de los campos hay dos botones: 'Iniciar sesión' y 'Registrarse'.

Audioserver - (C) Daniel Figeroa Fernandez. 2023

Figura 4.3: Mock-up del inicio de sesión.

■ Registro

La siguiente página es la de registro, que funcionará de manera similar a la de inicio de sesión, pero añadiendo un campo para introducir el correo electrónico:

Mock-up del registro. El formulario contiene tres campos de texto: 'Usuario', 'Contraseña' y 'Email'. Debajo de los campos hay un botón: 'Registrarse'.

Audioserver - (C) Daniel Figeroa Fernandez. 2023

Figura 4.4: Mock-up del registro.

■ Página principal

Esta es la página principal de la web, es en la que se encuentra la mayoría de la funcionalidad que no esté relacionada con el historial, y es desde la que se graba el audio y se envía al servidor:

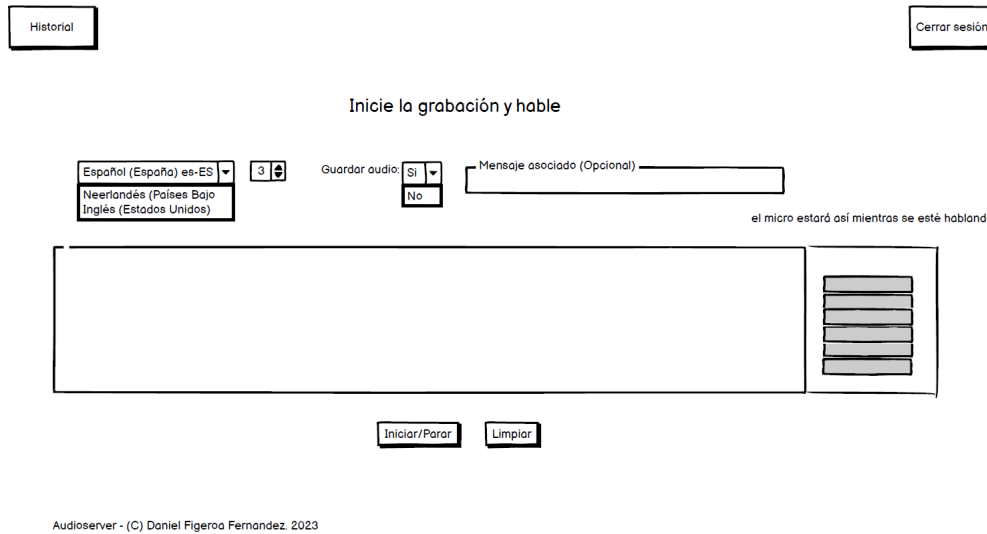


Figura 4.5: Mock-up de la página principal.

■ **Historial**

La última página es el historial, que mostrará en una tabla el histórico del usuario:

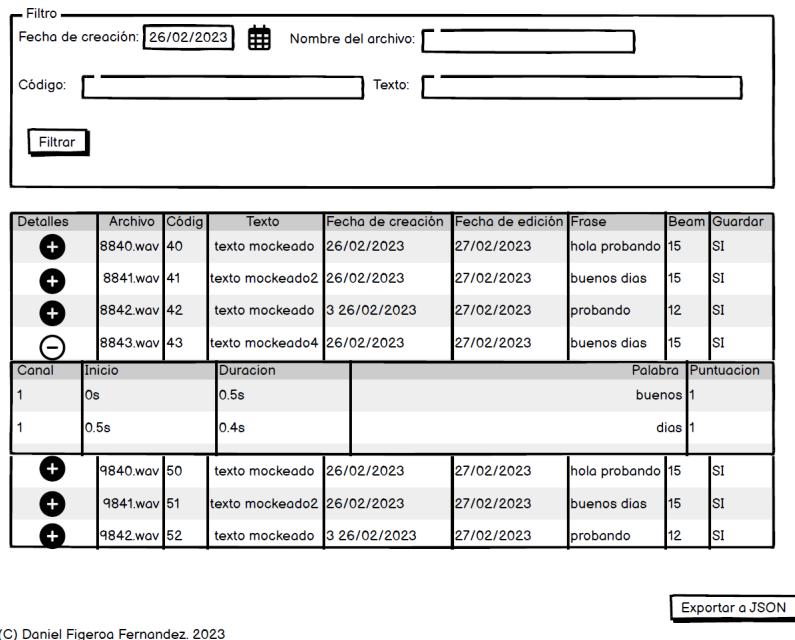


Figura 4.6: Mock-up del historial.

Esta maquetación corresponde a la idea inicial del sistema, y si bien se pretende que el sistema final sea parecido a los diseños de esta sección, es susceptible a cambios, ya sean causados por feedback recibido de los tutores, o por cambios en los requisitos que dejen estos diseños obsoletos.

4.3. Iteración 3

El objetivo principal de esta iteración es la creación de un prototipo funcional que sirva como base de la aplicación, y a partir del cual se pueda empezar a recibir feedback más directo. Dicho prototipo solo contará con la funcionalidad de la página principal, y aún no cumplirá todos los requisitos no funcionales, como el diseño modular ni la internacionalización.

Este prototipo permitirá grabar audio y que se suba al servidor, obteniendo su transcripción y mostrándola por pantalla, borrar la transcripción de la pantalla, y cambiar las 4 configuraciones que tiene el usuario (idioma, intensidad, nombre de la grabación, y guardar audio). Además, mostrará un indicador del volumen al que está hablando el usuario, y ya tendrá implementados los timers que envían el audio al servidor pasada la cantidad de tiempo indicada.

Durante esta iteración se realizaron dos reuniones con el tutor, una para planificar la iteración, y otra intermedia más adelante para comprobar el progreso y realizar las modificaciones necesarias. En la primera de ellas se me dio acceso a la API del servidor, además de una explicación sobre el funcionamiento de cada uno de los métodos de dicha API REST. Por otro lado, lo más relevante de la segunda reunión fueron las ideas de añadir colores al indicador de volumen, y de crear un fichero de constantes. Este fichero permitirá cambiar algunos parámetros de la aplicación de forma sencilla, como pueden ser la duración de los timers para enviar audio al servidor, o el idioma seleccionado por defecto, por ejemplo.

Centrándonos más en la parte del desarrollo, se comenzó creando el componente de la página principal, añadiendo los elementos necesarios (botones, campos de texto, etc) y dándole forma utilizando Bootstrap [25] para que siguieran el diseño creado en la parte maquetación de la iteración anterior. Lo siguiente que se implementó fue el indicador de volumen, seguido de las llamadas al servidor para obtener las transcripciones de los audios y por último de los timers.

En lo relacionado con las llamadas al servidor, una de las cosas más importantes que se tuvieron que hacer fue cambiar los archivos de las grabaciones a formato .wav para que el servidor los pudiera procesar correctamente, además de modificar la configuración de recordrtc y de VAD para que el audio se entendiera bien (y por lo tanto resultara en transcripciones más precisas).

Los principales problemas encontrados durante el desarrollo de esta iteración tuvieron que ver con el indicador de volumen no actualizándose lo suficientemente rápido, y con los timers, que son causantes de problemas de forma recurrente.

4.4. Iteración 4

De acuerdo con la planificación inicial, en esta iteración se desarrollará el resto de la funcionalidad de la aplicación, y por ello, su duración será la mayor de todas. Debido a esta extensión a lo largo del tiempo, será necesario realizar varias reuniones con los tutores, que permitan detectar problemas, ausencias de funcionalidad, o aspectos de la aplicación que se puedan mejorar, así como asegurar que el ritmo al que progresa el desarrollo es el adecuado.

La primera de estas reuniones se realiza justo al comienzo de la iteración, y principalmente se centra en la revisión del prototipo creado en la iteración anterior. En ella se detectó que los timers no funcionaban correctamente, por lo que será necesario utilizar parte del tiempo de desarrollo de esta iteración para solucionar esos problemas.

A lo largo del resto de reuniones se fueron planteando algunas de las propuestas de mejora por parte de los tutores. Estas propuestas de mejora, en conjunto con los cambios en algunas de las funcionalidades previamente especificadas (como añadir más opciones a los filtros en el historial), y la creación de dos nuevos requisitos funcionales, provocaron grandes cambios en lo referido al diseño de la interfaz (incluyendo un rediseño casi completo de la página principal).

Los nuevos requisitos añadidos durante esta iteración son ambos de carácter funcional, y consisten en lo siguiente:

- **RF14: Mostrar envío**, El sistema mostrará al usuario el progreso del envío del audio al servidor. Este requisito, de prioridad media, se añadió aproximadamente a mitad de la iteración.
- **RF15: Cambiar idioma**. El sistema permitirá al usuario cambiar el idioma de la página. Este requisito, de prioridad alta, se añadió en la última reunión antes del fin de la iteración, y causó varios problemas, que se explicarán con más detalle en las siguientes secciones.

4.4.1. Implementación de la nueva funcionalidad

La primera parte de la funcionalidad asignada a esta iteración que se desarrolló fue el inicio de sesión. Esto es debido a que el servidor tiene implementadas funciones de seguridad que obligan al usuario a haber iniciado sesión y tener un token de acceso válido para poder hacer llamadas al mismo (como por ejemplo las necesarias para subir un audio y obtener su transcripción).

En cuanto el funcionamiento de dichos tokens, tanto la generación de los mismos, como su comprobación ya están implementadas en el servidor, por lo que para nosotros solo es necesario realizar una llamada al servicio de inicio de sesión través de la API REST para obtener el token, y después guardarlo para enviarlo en el resto de llamadas. Este paso en el prototipo de la iteración anterior se debía hacer manualmente.

Tras acabar el inicio de sesión, y de dedicar tiempo a corregir algunos fallos, se pasó a desarrollar la parte más compleja de la iteración, el historial. La principal complicación del historial es que cada entrada tiene un formato como este:

```

1 {
2     "filename": "1644830598840.wav",
3     "code": "1",
4     "text": "custom text",
5     "nbest": "dit is een test",
6     "ctm": [
7         {
8             "channel": 1,
9             "tbegin": 0,
10            "duration": 0.33,
11            "word": "dit",
12            "score": 1
13        },
14        {
15            "channel": 1,
16            "tbegin": 0.33,
17            "duration": 0.08,
18            "word": "is",
19            "score": 1
20        },
21        {
22            "channel": 1,
23            "tbegin": 0.41,
24            "duration": 0.03,
25            "word": "een",
26            "score": 1
27        },
28        {
29            "channel": 1,
30            "tbegin": 0.44,
31            "duration": 0.66,
32            "word": "test",
33            "score": 1
34        }
35    ],
36    "beam": 15,
37    "keep": true,
38    "created_at": "2022-02-14T09:23:18.930Z",
39    "updated_at": "2022-02-14T09:23:23.334Z"
40 },

```

Esta cantidad de información que mostrar y este formato hicieron que fuera necesario que cada entrada de la tabla tuviera una subtabla en la que mostrar los contenidos pertenecientes a "ctm".

Una vez que se consiguió crear la tabla, se pasó a la implementación de los filtros. Estos filtros causaron el problema de requerir tener cargadas una cantidad que podía llegar a ser muy grande de entradas para poder filtrar correctamente, lo cual podía provocar que se hicieran demasiadas llamadas al servidor (que solo es capaz de devolver las entradas de 100 en 100) además de un tiempo de espera muy largo para que se carguen una gran cantidad de páginas que posiblemente no se quieran visualizar. Para solucionarlo, se implementó un límite (que puede ser cambiado desde el fichero de constantes) a partir del cual no se cargan mas entradas automáticamente, y se añadió un botón a la interfaz que permitirá al usuario cargar más en caso de necesidad, junto con un mensaje explicativo.

Tras la implementación de los filtros, se pasó a algunas funcionalidades más sencillas, como el

Descripción de las iteraciones

cierre de sesión, la descarga del historial como JSON, el manejo de errores en el inicio de sesión y la corrección de algunos fallos que se fueron encontrando durante las pruebas la aplicación, como un error que hacía que no se parara la grabación correctamente.

En este punto ocurrió la reunión que introdujo el nuevo requisito de mostrar el progreso de los envíos de audio al servidor, junto con el añadido de nuevas opciones a los filtros. Durante el periodo entre esta reunión y la siguiente (que será la última de la iteración), se realizaron dichos cambios, junto con la creación de versiones de la aplicación en español e inglés para cumplir el requisito RNF04 (es decir, que el sistema sea fácil de traducir, aplicando i18n). Para ello, se utilizó la implementación por defecto de i18n del propio Angular, que permite elegir el idioma en el despliegue, y utiliza archivos que siguen un formato similar a este:

```
1 <trans-unit id="introductionHeader" datatype="html">
2   <source>Hello i18n!</source>
3   <target>Hola i18n!</target>
4 </trans-unit>
```

Tras terminar esta parte, se realizó una última reunión con los tutores, en la cual se plantearon cambios a la interfaz de la página principal y se añadió el nuevo requisito funcional de permitir al usuario cambiar el idioma de la página de forma dinámica. Tras esto se implementó el registro de usuarios y se pasó a implementar el selector de idiomas.

Este selector de idiomas resultó ser muy complejo de implementar si se seguía haciendo la traducción del i18n de la misma manera que se estaba haciendo antes. Esto se debe a que sería necesario un servidor como firebase que guardara todas las versiones de la página en los diferentes idiomas y que las fuera sirviendo al cliente según fueran necesarias. Para evitar esta complejidad, se pasó a utilizar *ngx-translate* [26], que es una librería que no solo nos permite cambiar de idioma fácil y rápidamente de forma dinámica, sino que además utiliza archivos .json, como por ejemplo:

```
1 {
2   "main.hello": "Hello World",
3   "main.bye": "Goodbye",
4 }
```

```
1 {
2   "main.hello": "Hola Mundo",
3   "main.bye": "Adios",
4 }
```

Utilizar estos archivos tiene la ventaja de que son mucho más fáciles de leer y editar para un traductor que no tenga conocimientos de informática. Además, para facilitar y acelerar la tarea de traducir la página, he utilizado una aplicación llamada BabelEdit [9], que proporciona una interfaz para crear los ids de cada cadena y editar sus traducciones, además de proporcionar sugerencias utilizando varios traductores online.

Una vez que toda la funcionalidad de la aplicación estaba completa, se pasó a dar los últimos retoques visuales y a hacer las modificaciones necesarias para que estas siguieran el diseño modular.

4.4.2. Diseños finales y diseño modular

Para adecuarse a los nuevos requisitos y para implementar algunas de las propuestas de mejora recibidas en las reuniones con los tutores, fue necesario hacer varios cambios a los diseños de las páginas:

■ Página principal:

Se trata de la página que más ha cambiado con respecto a su maquetación, se ha añadido la opción del menú de cambiar de idioma, se han cambiado los nombres y la colocación de los diferentes campos de la configuración, se ha mejorado el indicador de volumen y se ha añadido el de subida, que antes del rediseño del indicador de volumen se encontraba bajo el resultado y era mucho más grande. Además, se han hecho pequeñas mejoras como añadir colores e iconos a los botones, o mostrar el nombre de usuario.



Figura 4.7: Diseño de la página principal.

En cuanto a la vista para móvil, en general se han agrandado y centrado todos los elementos de la interfaz, para facilitar su uso:

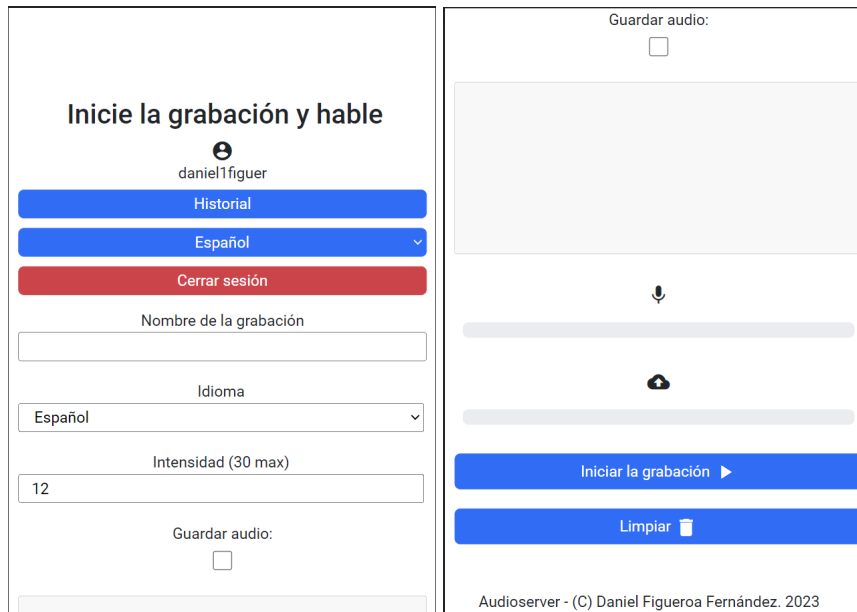
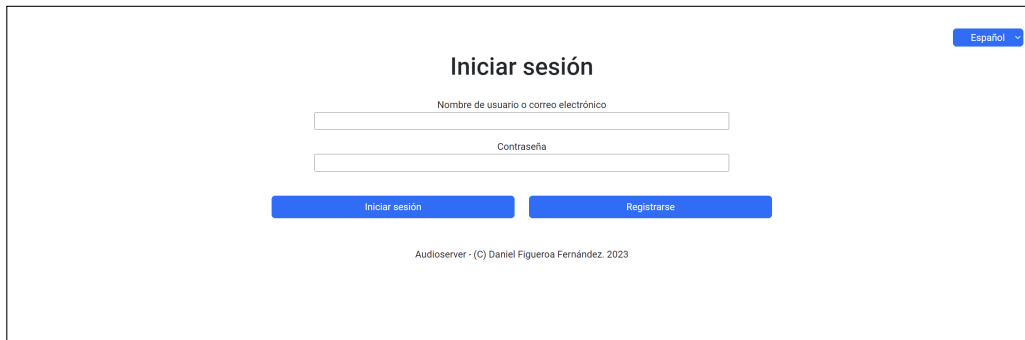


Figura 4.8: Diseño de la página principal en la vista para móvil.

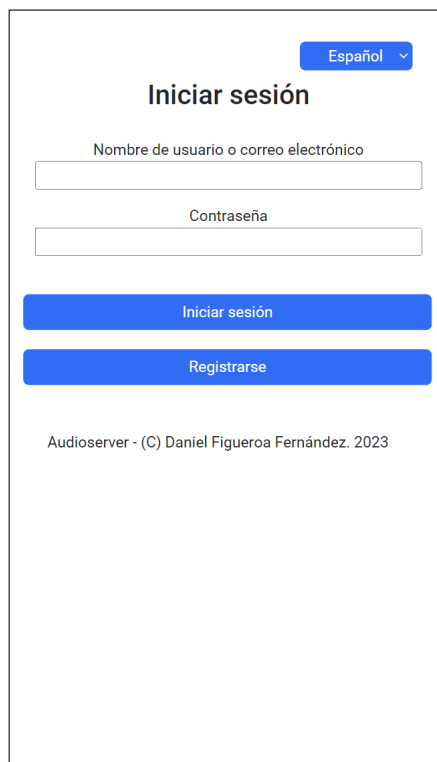
■ Inicio de sesión

Se trata de la pantalla más simple en cualquiera de sus dos versiones, y su única modificación con respecto a la maquetación inicial ha sido añadir el selector de idioma.



Este diagrama muestra la interfaz de usuario para el inicio de sesión en una versión de escritorio. El título "Iniciar sesión" está centrado en la parte superior. En la esquina superior derecha hay un selector de idioma que muestra "Español" con una flecha hacia abajo. Debajo del título, hay un campo de entrada etiquetado "Nombre de usuario o correo electrónico" y un campo de entrada etiquetado "Contraseña". Debajo de estos campos hay dos botones azules: "Iniciar sesión" a la izquierda y "Registrarse" a la derecha. En la parte inferior del formulario, se muestra el texto "Audioserver - (C) Daniel Figueroa Fernández. 2023".

Figura 4.9: Diseño del inicio de sesión.

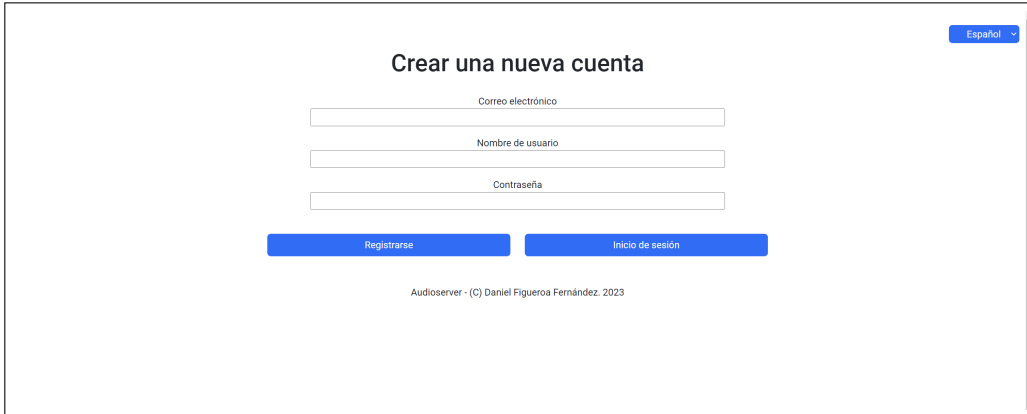


Este diagrama muestra la interfaz de usuario para el inicio de sesión en una versión para móvil. El selector de idioma "Español" con una flecha hacia abajo está ahora en la parte superior izquierda. El título "Iniciar sesión" sigue centrado. Los campos de entrada para "Nombre de usuario o correo electrónico" y "Contraseña" están apilados uno encima del otro. Los botones "Iniciar sesión" y "Registrarse" también están apilados uno encima del otro. El texto de copyright "Audioserver - (C) Daniel Figueroa Fernández. 2023" permanece en la parte inferior.

Figura 4.10: Diseño del inicio de sesión en la vista para móvil.

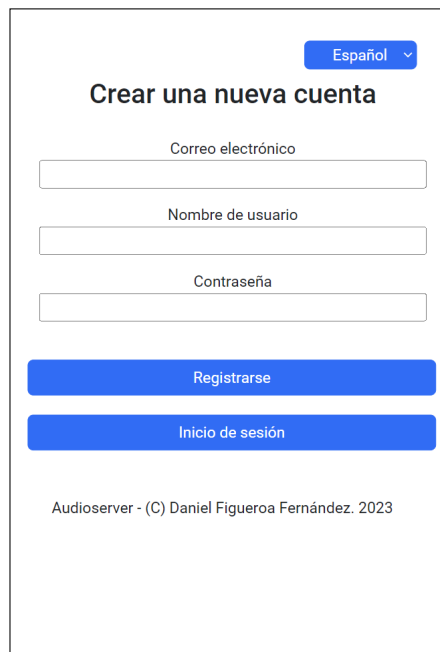
■ Registro

Se trata de una vista muy parecida a la de inicio de sesión, de modo que tampoco ha tenido grandes cambios. Se han ordenado los campos para que sean más intuitivos, y se ha añadido el selector de idioma.



The image shows a desktop view of a registration form titled "Crear una nueva cuenta". In the top right corner, there is a language selector button labeled "Español" with a downward arrow. The form contains three input fields: "Correo electrónico", "Nombre de usuario", and "Contraseña". Below these fields are two blue buttons: "Registrarse" and "Inicio de sesión". At the bottom center, there is a small copyright notice: "Audioserver - (C) Daniel Figueroa Fernández. 2023".

Figura 4.11: Diseño del registro.



The image shows a mobile view of the same registration form. The layout is vertically oriented. The language selector "Español" is at the top right. The title "Crear una nueva cuenta" is centered. The input fields for "Correo electrónico", "Nombre de usuario", and "Contraseña" are stacked vertically. Below them are the "Registrarse" and "Inicio de sesión" buttons, also stacked vertically. The copyright notice "Audioserver - (C) Daniel Figueroa Fernández. 2023" is at the bottom.

Figura 4.12: Diseño del registro en la vista para móvil.

- Historial** A pesar de su complejidad, el historial ha recibido una cantidad relativamente pequeña de cambios comparado con su diseño inicial, siendo los más importantes el añadido del botón para cargar más entradas, los nuevos campos del filtro, y como en el resto de vistas, el selector de volumen.

The screenshot displays the history page of the application. At the top right, there is a user profile for 'daniel1figuer' with buttons for 'Inicio', 'Español', and 'Cerrar sesión'. Below this, there are search filters for 'Fecha de creación', 'Fecha de actualización', 'Nombre de la grabación', 'Nombre del archivo', and 'Resultado'. A 'Filtro' button is centered below the filters. The main content is a table of recordings with columns: Código, Nombre de la grabación, Resultado, Guardar audio, Intensidad, Nombre del archivo, Fecha de creación, and Fecha de actualización. Below the table, there is a detailed view of a recording with columns: Canal, Inicio, Duración, Palabra, and Puntuación. At the bottom, there is a 'Descargar JSON' button, a message 'Se han cargado las primeras 1000 entradas de 3026', and a 'Cargar 1000 más' button. The footer contains 'Audioserver - (C) Daniel Figueroa Fernández. 2023'.

Código	Nombre de la grabación	Resultado	Guardar audio	Intensidad	Nombre del archivo	Fecha de creación	Fecha de actualización
9	string	HOLA SOY DANIEL ESTOY PROBANDO EL DESPEJE	true	12	1678201800993.wav	2023-03-07T15:10:01.225Z	2023-03-07T15:12:21.767Z
9	texto	PROBANDO PROA	true	12	1680515662763.wav	2023-04-03T09:54:22.920Z	2023-04-03T09:55:31.376Z
Canal	Inicio	Duración	Palabra	Puntuación			
1	0	0.49	PROBANDO	0.79			
1	0.5	0.17	PROA	0.87			
9	texto	PROBANDO APROBANDO OJO PARA QUE FUNCIONE TODO	true	12	1680517138097.wav	2023-04-03T10:18:58.315Z	2023-04-03T10:19:28.801Z

Canal	Inicio	Duración	Palabra	Puntuación			
1	0	0.49	PROBANDO	0.79			
1	0.5	0.17	PROA	0.87			
9	texto	PROBANDO APROBANDO OJO PARA QUE FUNCIONE TODO	true	12	1680517138097.wav	2023-04-03T10:18:58.315Z	2023-04-03T10:19:28.801Z
9	por que eres obligatorio	PARA PROBANDO	true	12	1680605850006.wav	2023-04-04T10:57:30.268Z	2023-04-04T10:57:32.873Z
9	pruebas	HOLA PROBANDO QUE TAL	true	12	1680605943688.wav	2023-04-04T10:59:03.908Z	2023-04-04T10:59:06.491Z
9	pruebas	<UNK> PRUEBA BUENOS DÍAS	true	12	1680606089732.wav	2023-04-04T11:01:29.954Z	2023-04-04T11:01:32.512Z
3	pruebas	true	true	12	1680606104634.wav	2023-04-04T11:01:44.709Z	2023-04-04T11:01:46.940Z
3	pruebas	mhm	true	12	1680606109716.wav	2023-04-04T11:01:49.803Z	2023-04-04T11:01:51.635Z
3	pruebas	true	true	12	1680606114710.wav	2023-04-04T11:01:54.796Z	2023-04-04T11:01:56.582Z
3	pruebas	huh	true	12	1680606119781.wav	2023-04-04T11:01:59.910Z	2023-04-04T11:02:01.726Z

Items per page: 10 1 - 10 of 417

Descargar JSON Se han cargado las primeras 1000 entradas de 3026 Cargar 1000 más

Audioserver - (C) Daniel Figueroa Fernández. 2023

Figura 4.13: Diseño del historial.

La vista para móviles del historial sigue los mismos principios que la de la página principal, haciendo los elementos tan grandes como sea posible, y centrándolos.

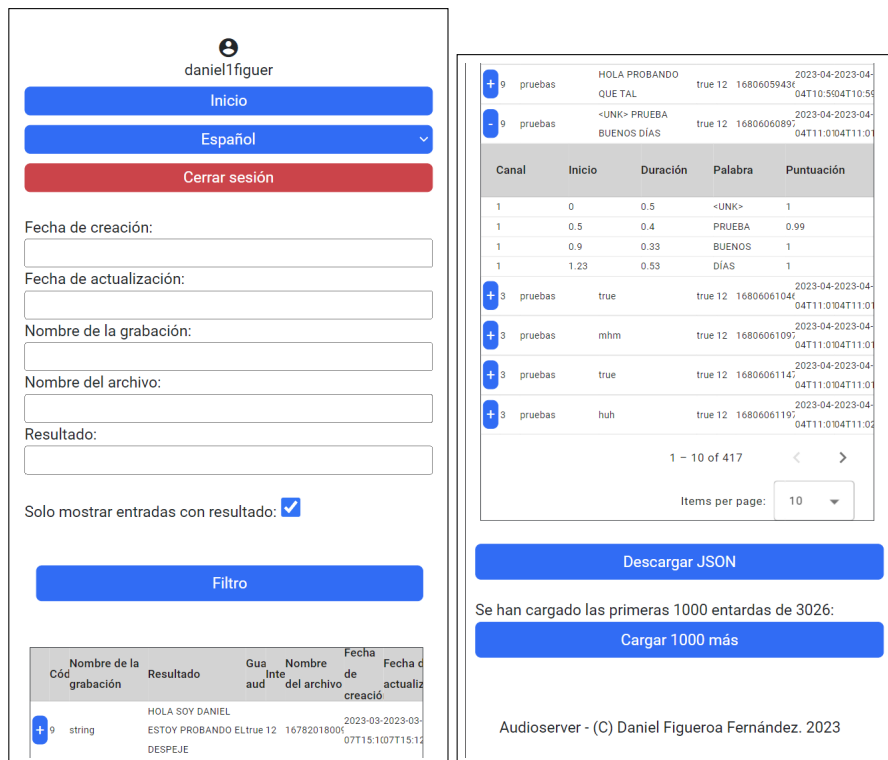


Figura 4.14: Diseño del historial en la vista para móvil.

4.4.3. Pruebas funcionales y principales problemas

Durante el transcurso de la iteración, la aplicación se han ido probando para tratar de minimizar todos los posibles errores. Estas pruebas consisten en realizar una serie de acciones como las haría un usuario y comprobar si los resultados son los esperados, y si encontramos algo que no funcione correctamente por el camino. Un ejemplo de estas pruebas podría ser iniciar una grabación, decir un par frases observando el micro y el indicador de subida para comprobar que funcionan, parar la grabación, comprobar que se muestra el resultado por pantalla, ir al historial y buscar con los filtros las entradas correspondientes.

A lo largo de estas pruebas se han encontrado numerosos problemas, entre los cuales destacan los relacionados con los timers, y los que tienen que ver con parar la grabación, ambos por ocurrir de manera recurrente.

Los problemas al parar la grabación venían en su mayoría por no parar correctamente las grabaciones o el VAD, y que esto causara problemas en las siguientes grabaciones, que podían variar desde fallos totales a errores más sutiles como parpadeos en el indicador de volumen, aunque también aparecieron en otros contextos, como por ejemplo al llegar al límite de caracteres del resultado, lo que para la grabación forzosamente, pudiendo causar problemas dependiendo del estado del sistema en el momento de la parada.

El otro gran tipo de problemas son los relacionados con los contadores, que por ciertas incompatibilidades los de algunas librerías dejaban congelada la página, o que por cómo se estaban usando

causaban retrasos mayores a los intencionados.

4.5. Iteración 5

En esta última iteración, se procede al despliegue, realización de pruebas y redacción de la memoria.

4.5.1. Pruebas

Aunque durante el transcurso de todas las iteraciones que involucran la implementación de nueva funcionalidad se han realizado pruebas de funcionalidad, es en esta iteración, tras tener toda la funcionalidad implementada, en la que se han hecho las pruebas finales.

Estas pruebas han sido de dos tipos:

- **Test de funcionalidad:** pruebas que buscan comprobar el correcto funcionamiento de la aplicación, se han realizado tanto en local como sobre la aplicación desplegada en un servidor, y tanto para la vista para ordenador como para la de móviles. Estos test han permitido encontrar y corregir bugs como botones que no funcionaban en la vista para móviles, o validaciones de datos de entrada que eran necesarias, y sin embargo no se realizaban.
- **Test de usabilidad:** pruebas que buscan evaluar la usabilidad y el diseño de las pantallas de la aplicación, para su realización se han utilizado 4 usuarios reales que desconocían la aplicación y se les ha pedido que realicen diversas tareas, anotando su rendimiento y opiniones. Esto ha permitido plantear diversas mejoras al diseño de la interfaz de usuario.

Para ver en más detalle tanto el planteamiento como las conclusiones de estas pruebas, ver el capítulo 6.

4.5.2. Despliegue

Se despliega la aplicación en una máquina virtual de la UVa siguiendo los pasos indicados en el manual del apéndice B, este despliegue inicial no supuso muchos problemas mas allá de elegir bien los puertos para el contenedor de Docker. Tras finalizar este despliegue se pudo comprobar que la aplicación no funcionaba correctamente, puesto que lanzaba un error al tratar de grabar audio.

Tras investigar un poco se descubrió que este error era causado por intentar grabar audio a través de HTTP, lo que no causaba problemas para una aplicación local (que es como se fue probando la aplicación durante el desarrollo), pero sí al intentar hacerlo desde un servidor. Esto se debe a que por motivos de seguridad los navegadores no permiten grabar audio a través de HTTP, para evitar problemas de privacidad.

Descripción de las iteraciones

Aunque se buscó una solución a nivel de código, al poco tiempo fue obvio que iba a ser necesario añadir un certificado SSL que permitiera acceder a la aplicación a través de HTTPS. Para esto se generó un nuevo certificado autofirmado, lo cual por un lado es bueno, porque se pudo generar gratis y de forma relativamente rápida, además de ser suficiente para poder acceder a la aplicación utilizando HTTPS, pero por otro lado tiene el inconveniente de que los navegadores mostrará un mensaje como este:

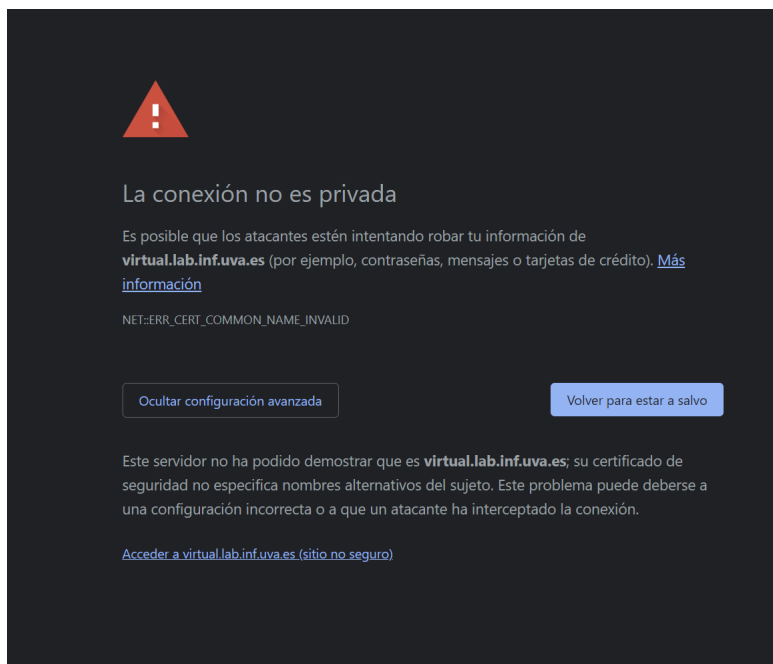


Figura 4.15: Aviso al intentar acceder a la aplicación.

Aunque podremos acceder a la aplicación igualmente a través de la configuración avanzada. El siguiente paso, y el que necesitó más tiempo fue configurar angular y docker para poder hacer el despliegue utilizando esos certificados, para ello se creó una nueva carpeta dentro del proyecto para contener los archivos necesarios (el certificado y la clave), se creó el archivo nginx-config.conf, con el siguiente contenido:

```
1 server {
2     listen          443 ssl;
3     ssl_certificate  "ssl/server.crt";
4     ssl_certificate_key "ssl/server.key";
5     root /usr/share/nginx/html;
6     index index.html;
7     location / {
8         try_files $uri $uri/ =404;
9     }
10 }
```

Este fichero principalmente selecciona el puerto 443, que es el puerto por defecto de HTTPS, y añade tanto la clave como el certificado al despliegue. Por último, se realizaron las modificaciones necesarias al Dockerfile para que utilice esta configuración.

4.5.3. Redacción de la memoria

Se finaliza la redacción de la memoria, en Overleaf utilizando LaTeX, para ello, se hará uso de las información anotada durante el desarrollo del proyecto. Esta información ha sido guardada tanto utilizando el bloc de notas para realizar apuntes durante las reuniones, como en Excel guardando la lista de tareas que se han ido realizando, además de los comentarios del código.

Capítulo 5

Estado final de la aplicación

5.1. Diagramas de análisis

A lo largo de este capítulo se van a abordar los temas referentes tanto al análisis como al diseño de la aplicación, incluyendo todos los diagramas relevantes para cada parte.

5.1.1. Historias de usuario

En esta sección se detallarán todas las historias de usuario del proyecto, es decir, explicaciones generales e informales de la funcionalidad del software, desde el punto de vista del usuario. Para cada una de estas historias, se mostrará la siguiente información:

- **ID:** Un identificador único para cada historia de usuario.
- **Nombre:** Nombre de cada historia de usuario.
- **Prioridad:** Prioridad de la funcionalidad de dicha historia, entre alta, media y baja.
- **Riesgo:** Nivel de consecuencias en caso de fallo en dicha historia, entre alto, medio y bajo.
- **Descripción:** Descripción de la funcionalidad que se quiere que la aplicación cumpla.
- **Prerrequisitos:** Enumeración de los posibles prerrequisitos de la historia, en caso de que los tenga.
- **Validación:** Condiciones necesarias para que la historia de usuario se considere completada de forma satisfactoria.

ID	HU01
Nombre	Registro
Prioridad	Media
Riesgo	Medio
Descripción	Como usuario, quiero poder registrarme.
Prerrequisitos	Ninguno.
Validación	<ul style="list-style-type: none"> – Quiero poder registrarme en el sistema con un nombre de usuario, contraseña y correo electrónico. – Quiero que el usuario y correo electrónico sean únicos. – Que a tras confirmar el correo se cree el usuario y se pueda iniciar sesión con él.

Tabla 5.1: Historia de usuario HU01.

ID	HU02
Nombre	Inicio de sesión
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder iniciar sesión con mis credenciales en el sistema.
Prerrequisitos	Estar registrado en el sistema.
Validación	<ul style="list-style-type: none"> – Quiero poder iniciar sesión con un nombre de usuario y una contraseña.

Tabla 5.2: Historia de usuario HU02.

ID	HU03
Nombre	Cambiar idioma de la página
Prioridad	Media
Riesgo	Bajo
Descripción	Como usuario, quiero poder cambiar el idioma de la página.
Prerrequisitos	Existe una traducción de la página para el idioma seleccionado.
Validación	<ul style="list-style-type: none"> – Que el idioma de la página se cambie al instante y sin errores.

Tabla 5.3: Historia de usuario HU03.

Estado final de la aplicación

ID	HU04
Nombre	Seleccionar idioma de la transcripción
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder seleccionar el idioma para el cual obtener la transcripción de los audios.
Prerrequisitos	Haber iniciado sesión.
Validación	<ul style="list-style-type: none">– Que en los siguientes audios se obtenga la transcripción para el idioma seleccionado.

Tabla 5.4: Historia de usuario HU04.

ID	HU05
Nombre	Selección de intensidad
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder elegir la intensidad (calidad) de la transcripción.
Prerrequisitos	Haber iniciado sesión.
Validación	<ul style="list-style-type: none">– Que en los siguientes audios se obtenga la transcripción con el nivel de intensidad seleccionado.

Tabla 5.5: Historia de usuario HU05.

ID	HU06
Nombre	Elegir guardado
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder elegir si mis audios se guardan en el servidor.
Prerrequisitos	Haber iniciado sesión.
Validación	<ul style="list-style-type: none">– Que para los siguientes audios se guarden o no en el servidor, dependiendo de mi elección.

Tabla 5.6: Historia de usuario HU06.

ID	HU07
Nombre	Nombrar grabación
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder poner nombre a la grabación.
Prerrequisitos	Haber iniciado sesión.
Validación	<ul style="list-style-type: none"> – Que los siguientes audios que se graben tengan el nombre de grabación indicado.

Tabla 5.7: Historia de usuario HU07.

ID	HU08
Nombre	Historial
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder ver un historial con los datos de todos mis audios.
Prerrequisitos	<ul style="list-style-type: none"> ▪ Haber iniciado sesión. ▪ Haber grabado algún audio.
Validación	<ul style="list-style-type: none"> – Que se muestre la información del historial.

Tabla 5.8: Historia de usuario HU08.

ID	HU09
Nombre	Obtención de la transcripción en tiempo real
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder grabar audios y obtener su transcripción en tiempo real.
Prerrequisitos	<ul style="list-style-type: none"> ■ Haber iniciado sesión. ■ Haber concedido los permisos para utilizar el micrófono.
Validación	<ul style="list-style-type: none"> – Que se me indique el volumen con el que estoy hablando. – Que se me indique el progreso del envío del audio al servidor. – Que tras finalizarse ese envío se muestre por pantalla la transcripción de lo que he dicho. – Que se pueda ver la información de ese audio en el historial.

Tabla 5.9: Historia de usuario HU09.

ID	HU10
Nombre	Filtros
Prioridad	Media
Riesgo	Alto
Descripción	Como usuario, quiero poder filtrar la información de mi historial.
Prerrequisitos	<ul style="list-style-type: none"> ■ Haber iniciado sesión. ■ Haber grabado algún audio.
Validación	<ul style="list-style-type: none"> – Que en el historial solo se muestren los datos de los audios que concuerden con lo que haya introducido en el filtro.

Tabla 5.10: Historia de usuario HU10.

ID	HU11
Nombre	Cerrar sesión
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder cerrar sesión.
Prerrequisitos	Haber iniciado sesión.
Validación	<ul style="list-style-type: none"> – Que mi sesión se cierre y no pueda interactuar con la aplicación hasta haber iniciado sesión de nuevo.

Tabla 5.11: Historia de usuario HU11.

ID	HU12
Nombre	Descargar historial
Prioridad	Baja
Riesgo	Bajo
Descripción	Como usuario, quiero poder descargar mi historial.
Prerrequisitos	<ul style="list-style-type: none"> ■ Haber iniciado sesión. ■ Haber grabado algún audio.
Validación	<ul style="list-style-type: none"> – Que se descargue un archivo JSON con la información del historial. – Que se descargue solo la parte filtrada.

Tabla 5.12: Historia de usuario HU012.

ID	HU13
Nombre	Limpiar resultado
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario, quiero poder limpiar el resultado de mi pantalla.
Prerrequisitos	<ul style="list-style-type: none">■ Haber iniciado sesión.■ Tener el resultado de la transcripción de un audio en mi pantalla.
Validación	<ul style="list-style-type: none">– Que se borre el resultado actual de la pantalla.

Tabla 5.13: Historia de usuario HU13.

5.1.2. Modelo de dominio

En la figura 5.1 se puede ver el diagrama de dominio de la aplicación, que indica las entidades que la componen, sus relaciones y sus multiplicidades.

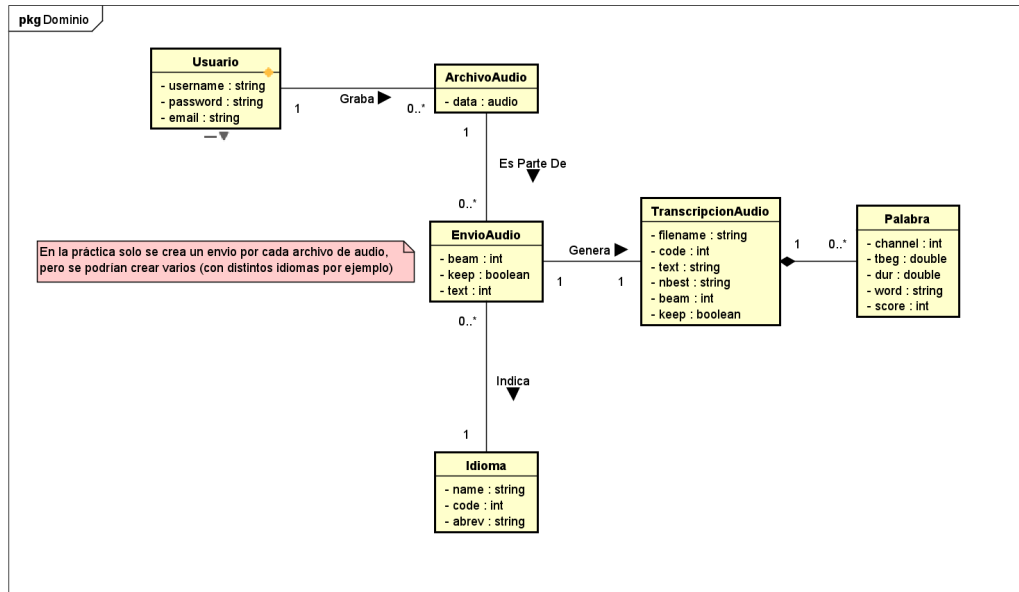


Figura 5.1: Diagrama de dominio.

Como se puede ver, el modelo de dominio de la aplicación es relativamente sencillo, esto se debe a que por la naturaleza del proyecto (la de crear un frontend web que trabaja con audio), la mayoría del tiempo se ha invertido en la creación de la propia web (diseño de las interfaces, facilidad de uso y tratamiento de audio), y no tanto en la complejidad de la aplicación.

En las siguientes tablas se describen en detalle las entidades que aparecen en el diagrama de dominio, con su descripción, responsabilidades y atributos:

Usuario	
Descripción	Modela un usuario que se ha registrado en la aplicación.
Responsabilidades	Modelar los detalles de un usuario que utiliza la aplicación, junto con sus datos de identificación.
Atributos	<ul style="list-style-type: none"> – username: Nombre de usuario único de cada usuario. – password: Contraseña que el usuario utilizará para iniciar sesión. – email: Correo electrónico del usuario.

Tabla 5.14: Entidad Usuario.

ArchivoAudio	
Descripción	Entidad que modela los archivos de audio del sistema.
Responsabilidades	Modelar los archivos de audio que graban los usuarios.
Atributos	<ul style="list-style-type: none"> – data: el archivo de audio que ha grabado el usuario.

Tabla 5.15: Entidad ArchivoAudio.

EnvioAudio	
Descripción	Modela los envíos de audio que realizará la aplicación al servidor.
Responsabilidades	Modelar el contenido de las peticiones que hará el sistema al servidor al enviar audio y obtener su transcripción.
Atributos	<ul style="list-style-type: none"> – beam: Nivel de intensidad con el que se quiere obtener la transcripción. – keep: Indicador que refleja si el usuario quiere que se guarde el audio en el servidor tras obtener su transcripción. – text: Nombre de la grabación.

Tabla 5.16: Entidad EnvioAudio.

Idioma	
Descripción	Modela los distintos idiomas de la aplicación .
Responsabilidades	Modelar los idiomas seleccionables por el usuario.
Atributos	<ul style="list-style-type: none"> – name: Nombre del idioma que se mostrará por pantalla. – code: Número que identifica el idioma para el que se quiere obtener la transcripción al enviarlo al servidor. – abrev: código del idioma según la normativa BCP 47 (de RFC 4646), por ejemplo, es-ES o en-US.

Tabla 5.17: Entidad Idioma.

TranscripcionAudio	
Descripción	Modela la transcripción de un audio.
Responsabilidades	Modelar el resultado de pedir la transcripción de un audio al servidor, con todos los parámetros que se mostrarán en el historial.
Atributos	<ul style="list-style-type: none"> – filename: Nombre único con el que el archivo de audio que ha generado esta transcripción se guardó en el servidor. – code: Código que envía el servidor para indicar si el resultado ha sido exitoso. – text: Nombre de la grabación a la que pertenece el audio usado para generar la transcripción. – nbest: Resultado de la transcripción. – beam: Nivel de intensidad utilizado. – keep: Indicador que refleja si el usuario ha decidido que se guarde el audio en el servidor tras obtener su transcripción o no.

Tabla 5.18: Entidad TranscripcionAudio.

Palabra	
Descripción	Modela las palabras de la transcripción.
Responsabilidades	Modelar la información de cada palabra de una transcripción para poder verla en el historial.
Atributos	<ul style="list-style-type: none"> – channel: El canal de audio a través del cual se ha obtenido esa palabra. – tbeg: Momento en el que comienza la palabra. – dur: Duración. – word: Palabra obtenida. – score: Puntuación asignada por el servidor.

Tabla 5.19: Entidad Palabra.

5.1.2.1. Relaciones entre entidades

Empezaremos por la entidad **Usuario**, que como se ha indicado en las tablas anteriores, representa a cada uno de los usuarios de la aplicación. Cuando uno de estos usuarios inicie la grabación, se comenzarán a crear archivos de audio (es decir, las entidades **ArchivoAudio**), cada una de ellas contendrá un pequeño fragmento de audio, de 5 segundos o menos. El objetivo de la aplicación es obtener las transcripciones de estos audios, por los que para cada uno de estos archivos se generará una entidad del tipo **EnvioAudio**, que es el objeto que se enviará al servidor para generar la transcripción. Cada uno de estos **EnvioAudio** indica al servidor para qué **Idioma** quiere obtener la transcripción.

Como se menciona en la anotación del diagrama, en la actualidad solo se crea un único envío por cada archivo, pero se podrían generar varios para enviarlos al servidor con distintos parámetros si así se quisiera (por ejemplo para intentar autodetectar el idioma).

Una vez que realiza la petición al servidor, se genera una **TranscripcionAudio** a partir del **EnvioAudio**, la cual contiene los datos de cada palabra del audio, en entidades del tipo **Palabra**, además de la información general del audio.

5.1.3. Diagramas de actividad

En esta sección se presentan los diagramas de actividad correspondientes a las historias de usuario descritas en el apartado 5.1.1. Puesto que para la realización de este proyecto no se ha tenido acceso a la lógica del servicio del backend contra el que se lanzan las llamadas (y por lo tanto se desconoce exactamente que operaciones realiza) en estos diagramas se detallarán las acciones realizadas en el frontend, junto con una simplificación de las realizadas por el servidor del backend. Esta simplificación tiene como objetivo justificar las respuestas que se obtienen del servidor, no explicar su funcionamiento.

Además, se mostrarán tanto la secuencia normal, como la que representa el flujo de acciones en caso de aparecer algún error, y se supondrá que se cumplen los prerrequisitos indicados en el apartado 5.1.1.

5.1.3.1. HU01 - Registro

En la figura 5.2 se muestra el diagrama de actividad de la historia de usuario HU01. En ella, el usuario accede a la página del registro, rellena los datos necesarios, y pulsa enviar. A partir de aquí, el frontend hace una petición al backend enviando los datos proporcionados por el usuario. El backend devuelve un mensaje de éxito o uno de error, los cuales son mostrados al usuario por parte del frontend. Si el mensaje era de fallo, ese mensaje contendrá el motivo, por lo que el usuario podrá volver a introducir datos, corrigiendo la causa del error. Por otro lado, si el registro se ha realizado exitosamente, desde el frontend se le avisa al usuario de que revise su correo electrónico para encontrar un email enviado por el backend, que le permite confirmar su dirección de correo electrónico, completando el registro.

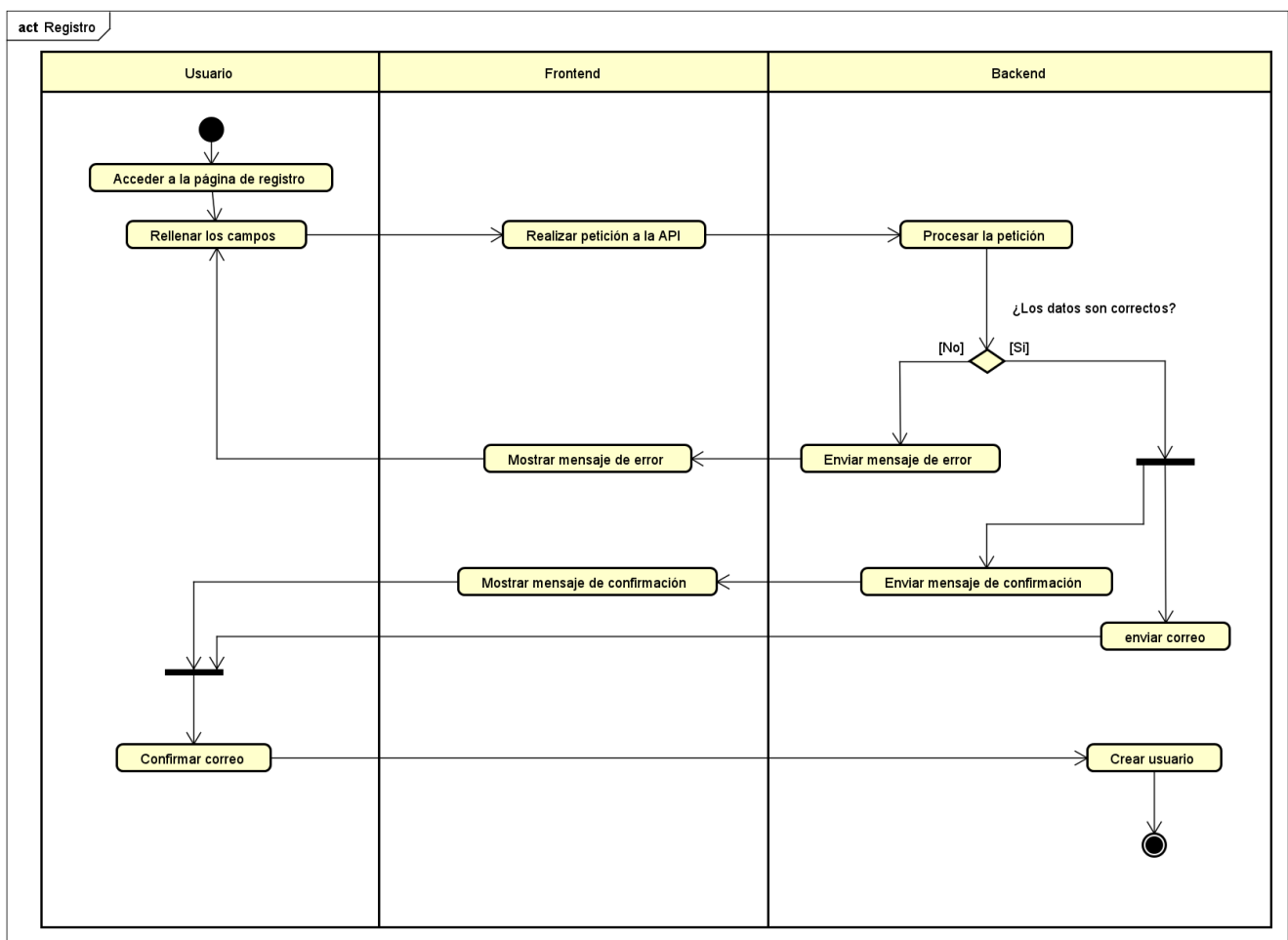


Figura 5.2: Diagrama de actividad de HU01.

5.1.3.2. HU02 - Inicio de sesión

En la figura 5.3 se muestra el diagrama de actividad de la historia de usuario HU02. En ella, el usuario accede a la página de inicio de sesión, introduce sus credenciales, y el frontend las envía al backend a través de una petición. Si las credenciales introducidas por el usuario son correctas, entonces el backend devolverá un token de autenticación, que se guardará en el frontend para utilizarlo en las peticiones que se realicen a partir de ahora, y después se le mostrará al usuario la página de inicio. Por otro lado, si las credenciales de inicio de sesión son erróneas, el backend se lo notificará al frontend, que informará al usuario de que los datos introducidos no son correctos y le dejará volver a intentarlo.

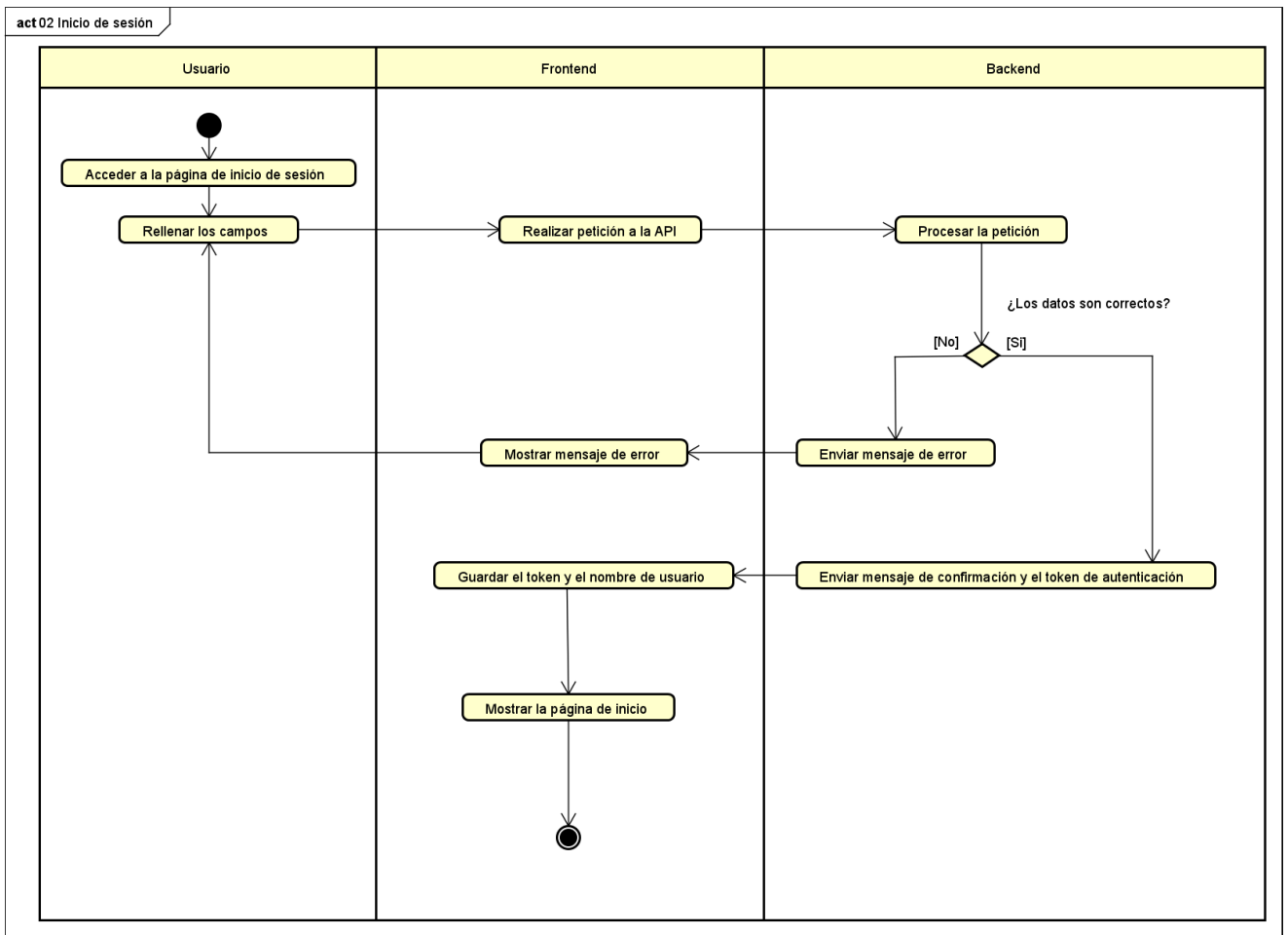


Figura 5.3: Diagrama de actividad de HU02.

5.1.3.3. HU03 - Cambiar idioma de la página

En la figura 5.4 se muestra el diagrama de actividad de la historia de usuario HU03. En ella, el usuario accede a cualquiera de las vistas de la página, y elige uno de los idiomas del selector. Tras esto, el frontend cambia de idioma al seleccionado, con la excepción del título de la pestaña, que no se actualiza a la vez que el resto de la página y debe forzarse una vez que se ha traducido lo demás.

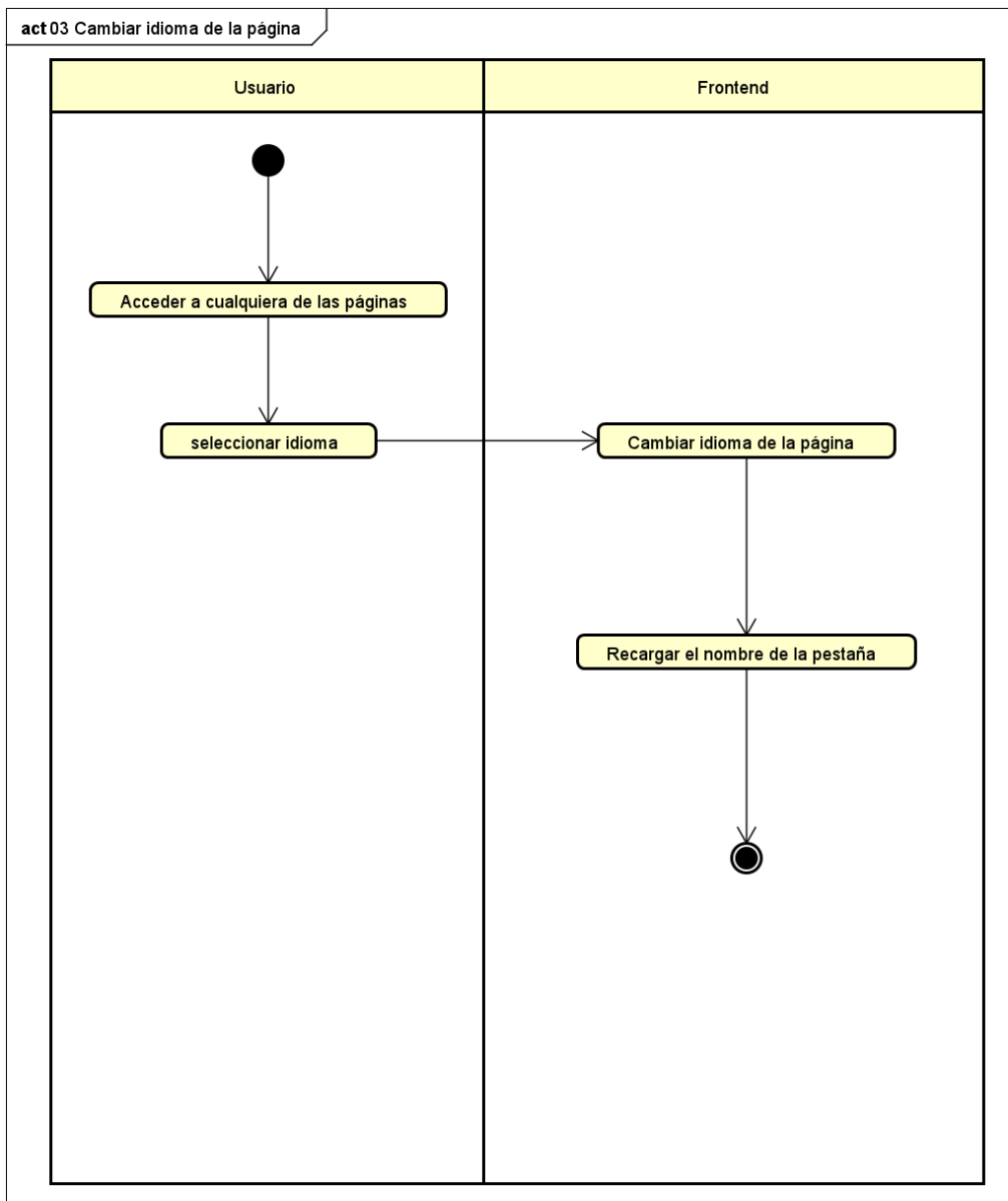


Figura 5.4: Diagrama de actividad de HU03.

5.1.3.4. HU04 - Seleccionar idioma de la transcripción

En la figura 5.5 se muestra el diagrama de actividad de la historia de usuario HU04. En ella, el usuario accede a la página principal, y utilizando el selector de idioma ubicado en la configuración de la grabación, elige un idioma. Tras esto, el frontend muestra qué idioma está seleccionado.

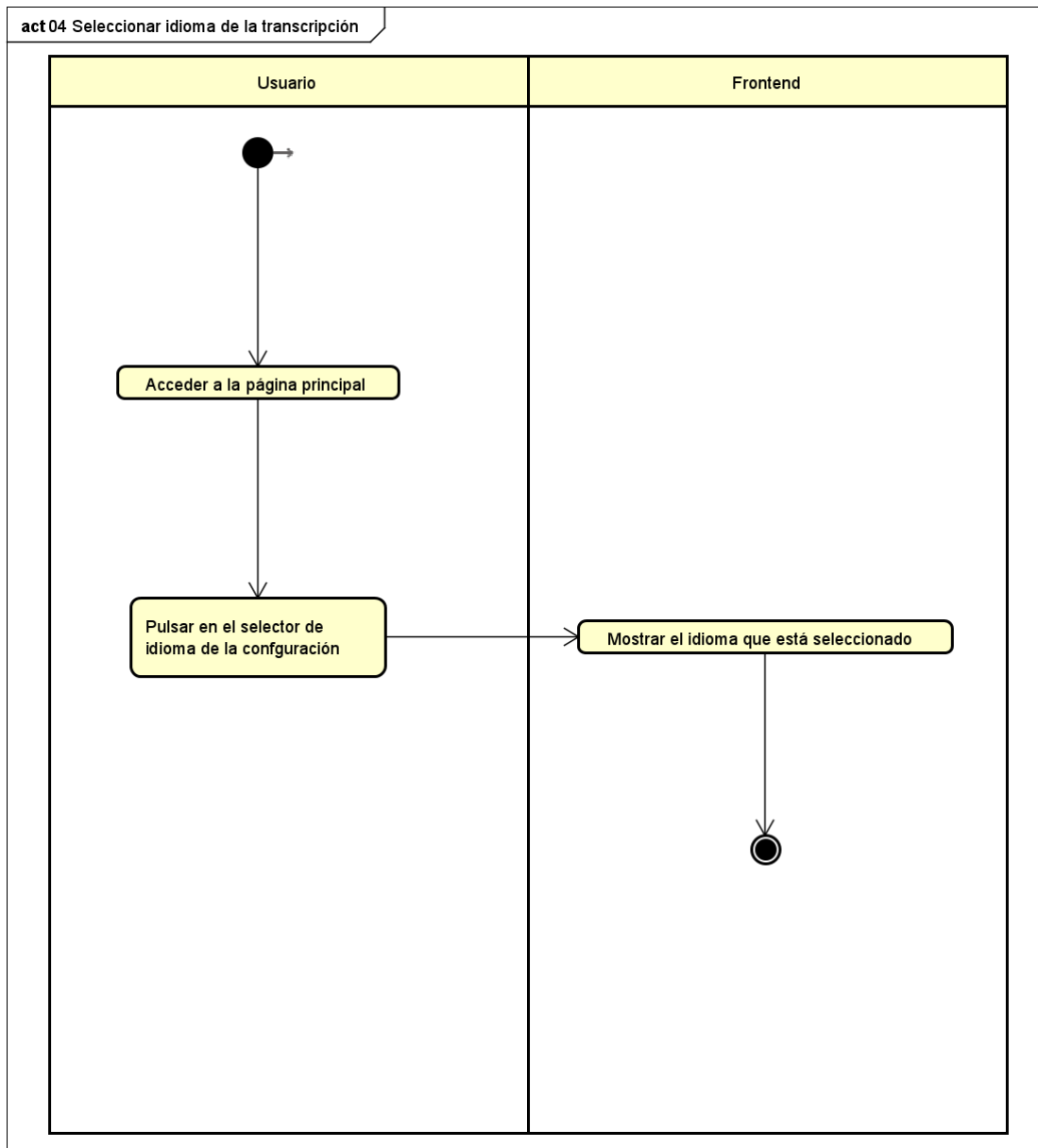


Figura 5.5: Diagrama de actividad de HU04.

5.1.3.5. HU05 - Selección de intensidad

En la figura 5.6 se muestra el diagrama de actividad de la historia de usuario HU05. En ella, el usuario accede a la página principal, y utilizando el selector de intensidad ubicado en la configuración de la grabación elige un valor. Tras esto, el frontend muestra qué valor está seleccionado.

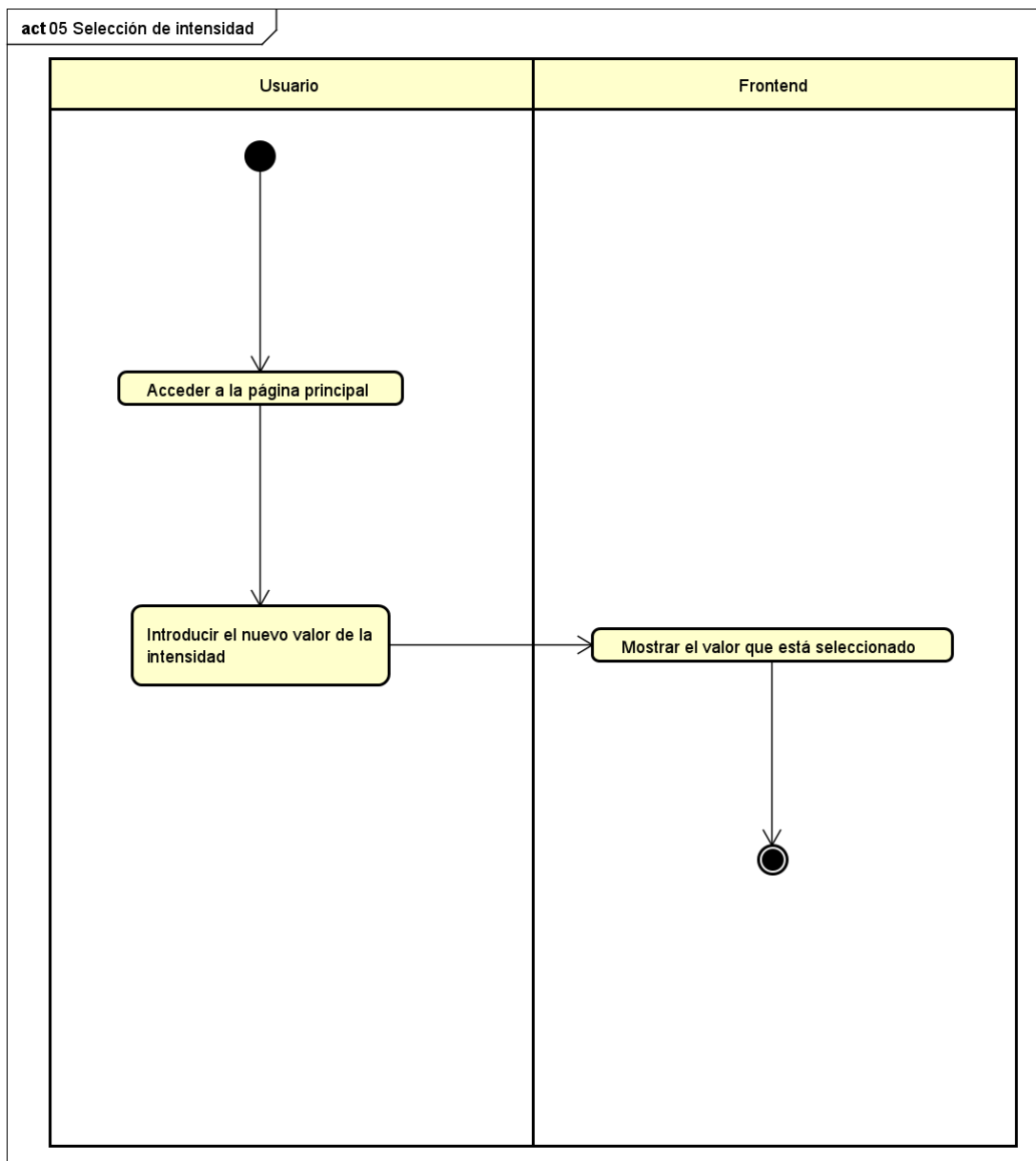


Figura 5.6: Diagrama de actividad de HU05.

5.1.3.6. HU06 - Elegir guardado

En la figura 5.7 se muestra el diagrama de actividad de la historia de usuario HU06. En ella, el usuario accede a la página principal, y selecciona si quiere o no que sus audios se guarden en el servidor. Tras esto, el frontend muestra la configuración actual.

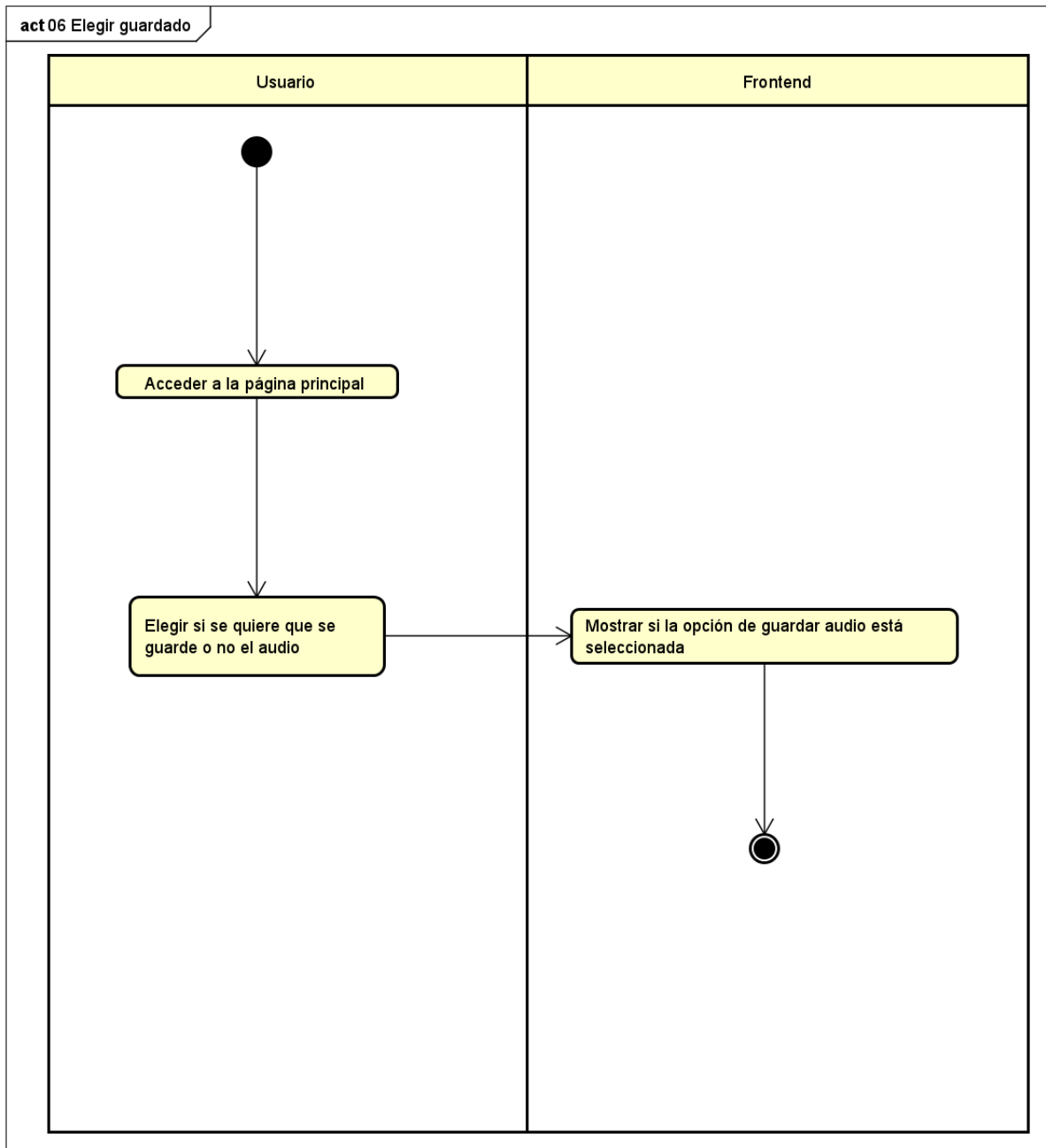


Figura 5.7: Diagrama de actividad de HU06.

5.1.3.7. HU07 - Nombrar grabación

En la figura 5.8 se muestra el diagrama de actividad de la historia de usuario HU07. En ella, el usuario accede a la página principal, y introduce el nombre que quiere darle a la siguiente grabación. Tras esto, el frontend muestra el nombre seleccionado actualmente.

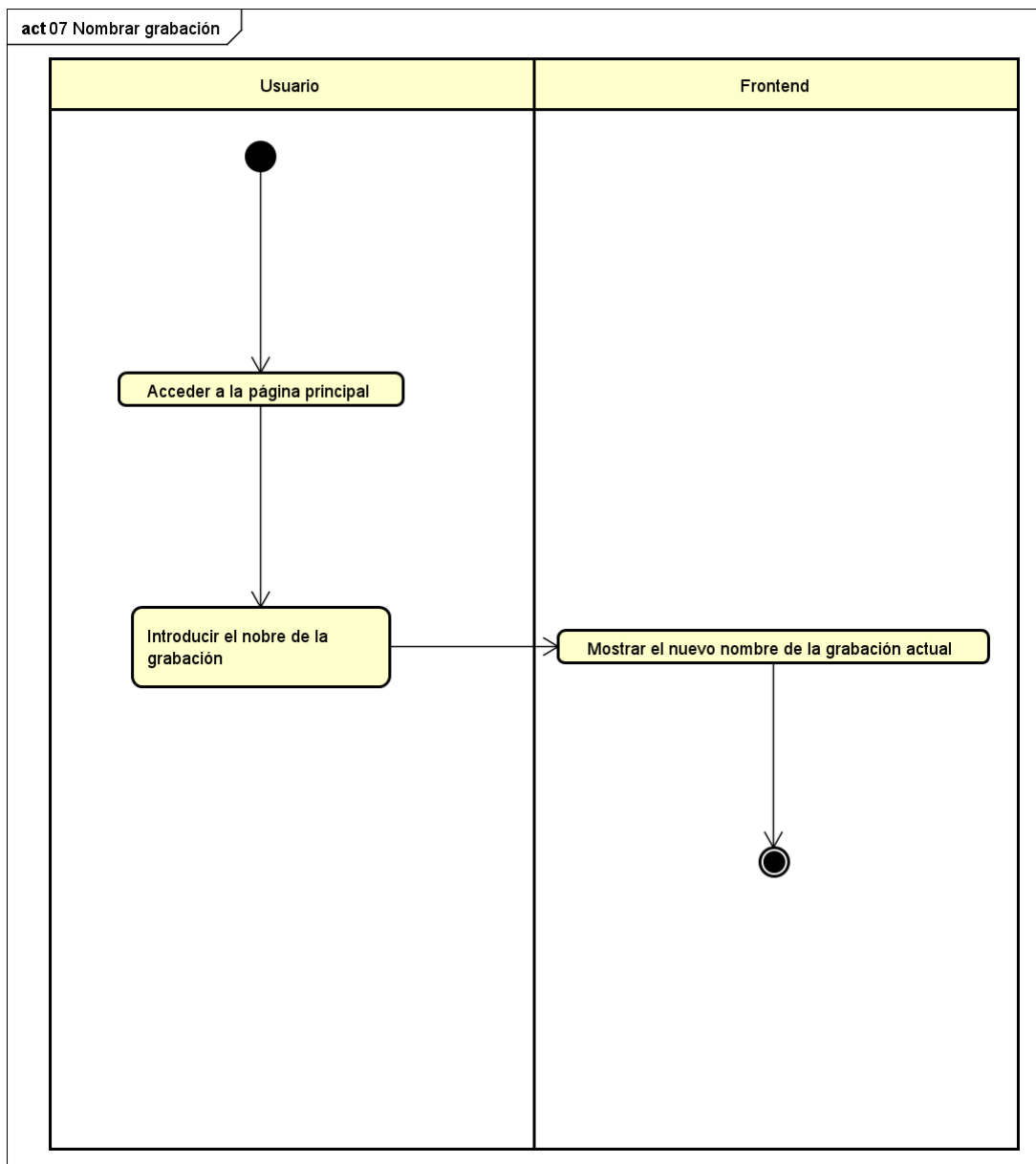


Figura 5.8: Diagrama de actividad de HU07.

5.1.3.8. HU08 - Historial

En la figura 5.9 se muestra el diagrama de actividad de la historia de usuario HU08. En ella, el usuario accede a la página del historial, lo que provoca que el frontend muestre la vista correspondiente, y que haga la llamada a la API para obtener la primera página del historial (100 resultados), cuando reciba la respuesta, guardará esos resultados, y los filtrará con los valores actuales de los filtros, si se ha llegado al límite de entradas del historial fijado en la configuración de la aplicación o no quedan más entradas en el servidor, se actualiza la tabla con los resultados y se muestra al usuario. En caso contrario se hace una nueva petición para obtener la primera de las páginas restantes.

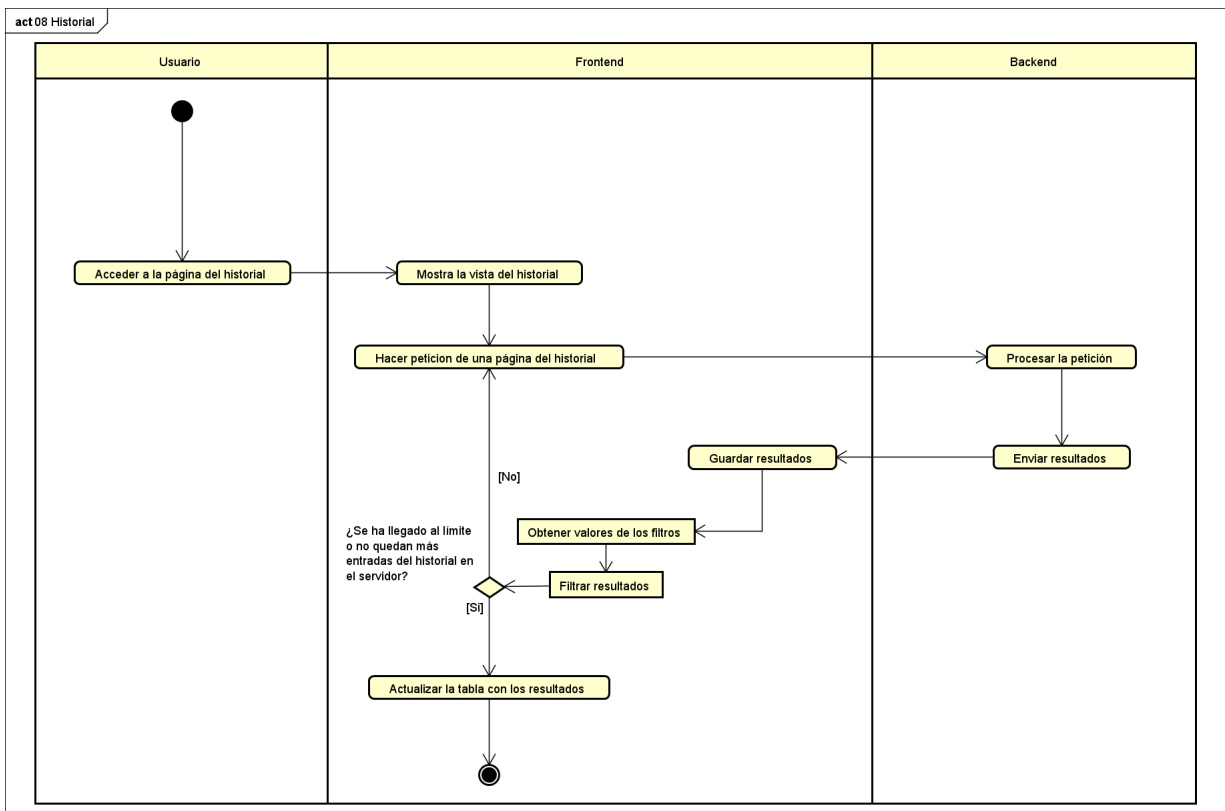


Figura 5.9: Diagrama de actividad de HU08.

5.1.3.9. HU09 - Obtención de la transcripción en tiempo real

En la figura 5.10 se muestra el diagrama de actividad de la historia de usuario HU09. Como se puede ver, es un diagrama de actividades más complejo de todas las historias de usuario, que representa la funcionalidad principal de la aplicación. Para comenzar, el usuario accede a la página principal y comienza a hablar. A partir de aquí, el frontend cambia el botón de iniciar grabación por el de pararla, y comienza un bucle que se repetirá cada 5 segundos (siempre que el usuario no intervenga).

El bucle consiste en comenzar a grabar audio e iniciar el Voice Activity Recognition (VAR), que será lo que detecte si el usuario se calla durante cierto tiempo, y lo que actualizará el indicador de volumen. Una vez que pasen los 5 segundos (que es el límite que se le ha decidido poner a la duración de los audios) se parará la grabación y el VAR, y mientras esta se procesa, se iniciará una grabación nueva.

En cuanto a los pasos para procesar la grabación, lo primero que se hace es comprobar si el límite de caracteres del resultado ha sido superado, en cuyo caso se sabe que la parada ha sido forzosa (debido a que en la iteración anterior del bucle se haya superado el límite), y por lo tanto no es necesario procesarla. Una vez que ya se sabe que sí que es necesario realizar la petición a la API, se crea el archivo de audio para enviarlo al servidor, se obtienen el nombre de la grabación, el nivel de intensidad, la decisión sobre si guardar o no el audio en el servidor y el idioma para el cual se quiere obtener la transcripción (que fueron elegidos en las historias de usuario HU04, HU05, HU06 y HU07), y se realiza la primera petición al backend.

La respuesta a esta primera petición no contiene la transcripción del audio, pero sí confirma que se ha subido correctamente. Al recibir esta respuesta, el frontend prepara y envía la segunda petición, la cual sí que devolverá la transcripción. Tras recibir esta respuesta, se suma el resultado obtenido al que ya se estuviera mostrando por pantalla, y se comprueba si se han superado el máximo de caracteres, en cuyo caso se bloquea el botón de comenzar grabación, y se para la que esté activa en ese momento.

En cuanto a las otras dos acciones del usuario, si éste se calla durante más de 0,7 segundos después de hablar, se terminará la iteración del bucle antes de tiempo, es decir, se parará y procesará la grabación actual, a la vez que se crea una nueva. Por otro lado, cuando el usuario decida pulsar el botón de parar grabación, se parará y procesará la grabación actual, pero sin iniciar una nueva.

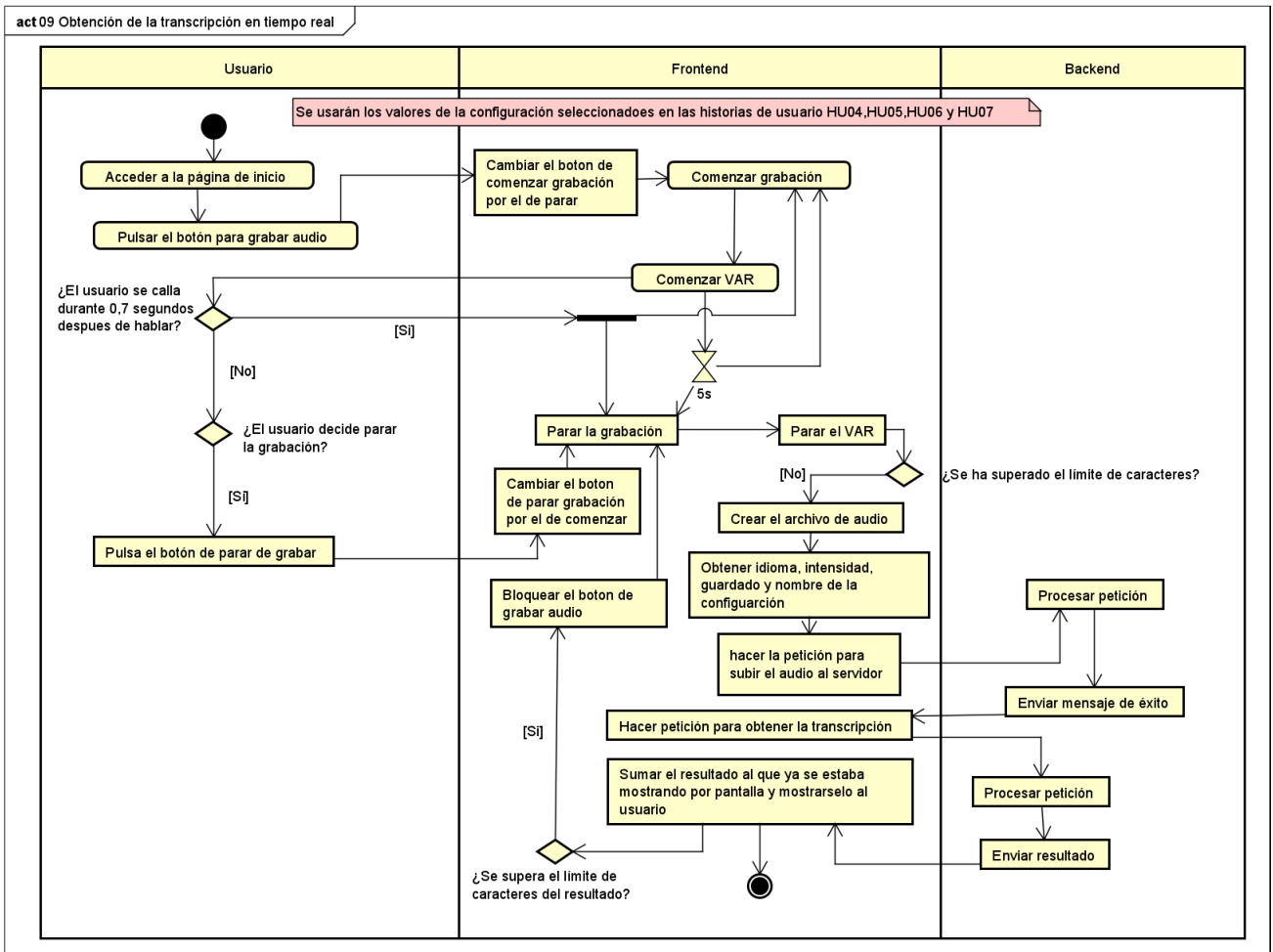


Figura 5.10: Diagrama de actividad de HU09.

5.1.3.10. HU10 - Filtros

En la figura 5.11 se muestra el diagrama de actividad de la historia de usuario HU10. En ella, el usuario accede a la página del historial (lo que implica cumplir la historia de usuario HU08), introduce los valores por los que quiere filtrar los resultados y pulsa el botón de filtrar. Una vez que se pulsa este botón, el frontend obtiene esos valores y filtra los resultados que tiene guardados, para después actualizar la tabla con estos mismos.

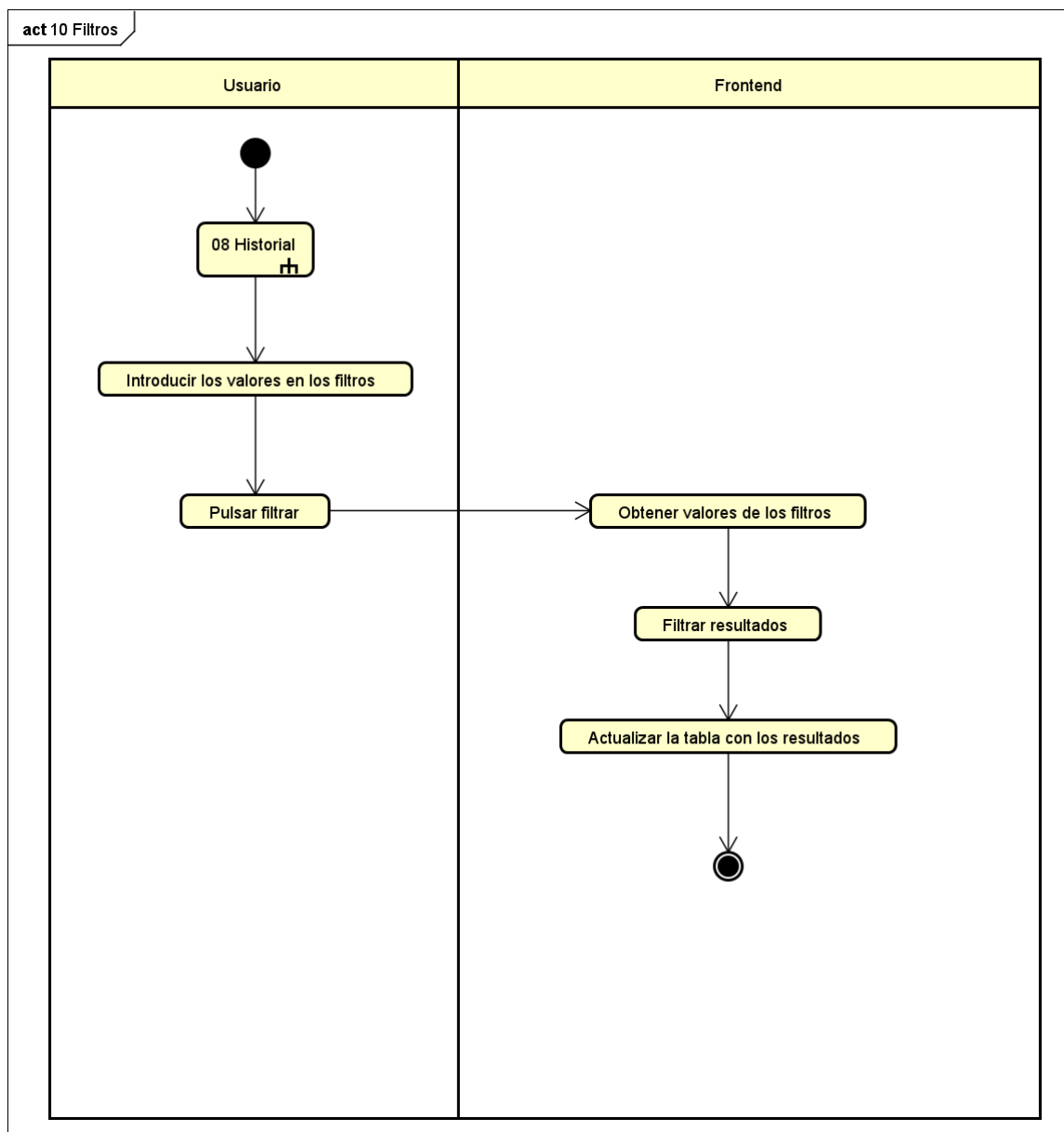


Figura 5.11: Diagrama de actividad de HU10.

5.1.3.11. HU11 - Cerrar sesión

En la figura 5.12 se muestra el diagrama de actividad de la historia de usuario HU11. En ella, desde cualquiera de las vistas a las que se puede acceder estando identificado ante el sistema (inicio e historial) el usuario pulsa el botón de cerrar sesión. Tras esto, el frontend borra el token y el nombre de usuario que tenía guardados para poder hacer peticiones a la API, y hace una redirección a la página de inicio de sesión.

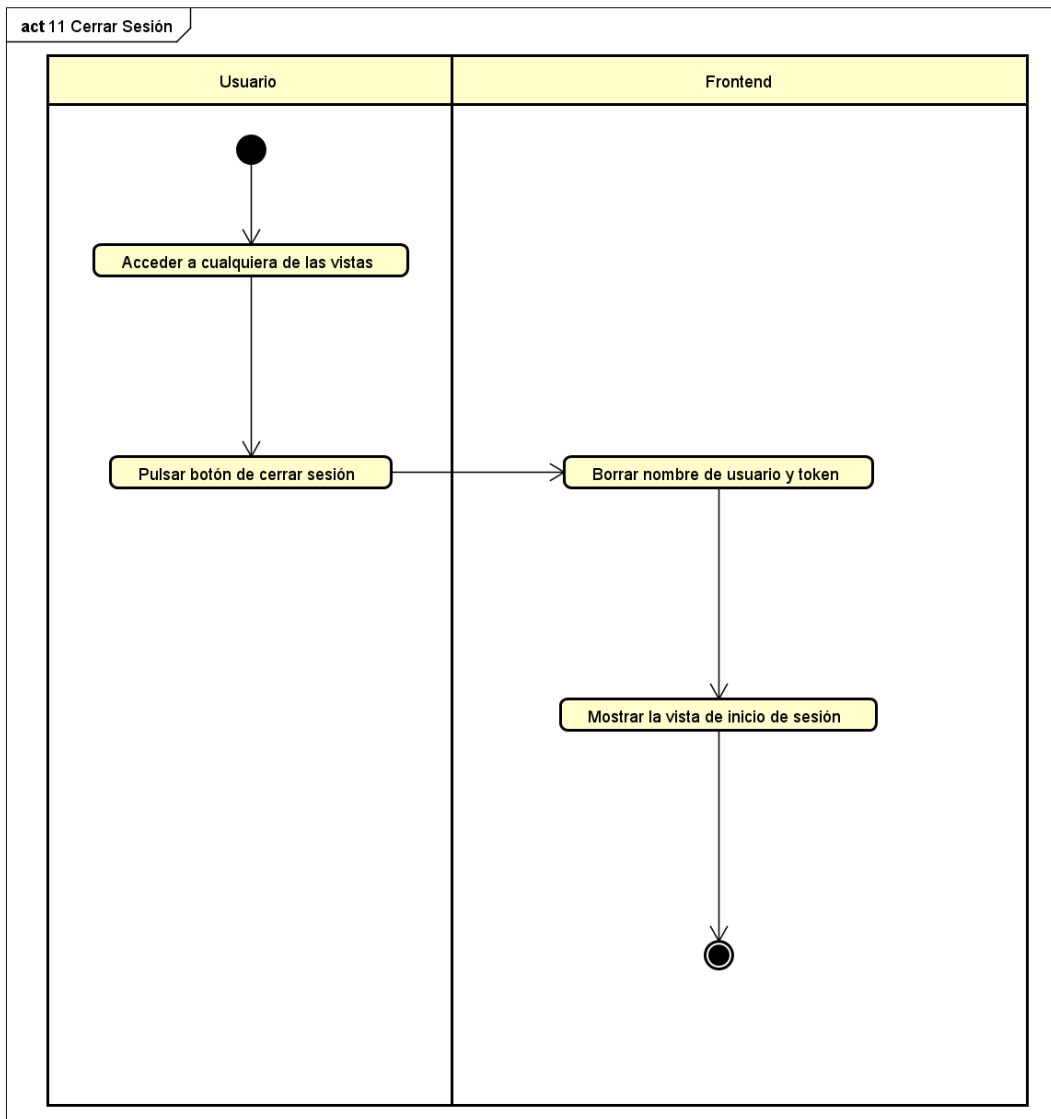


Figura 5.12: Diagrama de actividad de HU11.

5.1.3.12. HU12 - Descargar historial

En la figura 5.13 se muestra el diagrama de actividad de la historia de usuario HU12. En ella, el usuario accede a la página del historial (lo que implica cumplir la historia de usuario HU08), y pulsa el botón de descargar, pudiendo haber filtrado los resultados. Tras esto, el frontend crea un archivo .json con los datos que están en la tabla de resultados, y lo descarga.

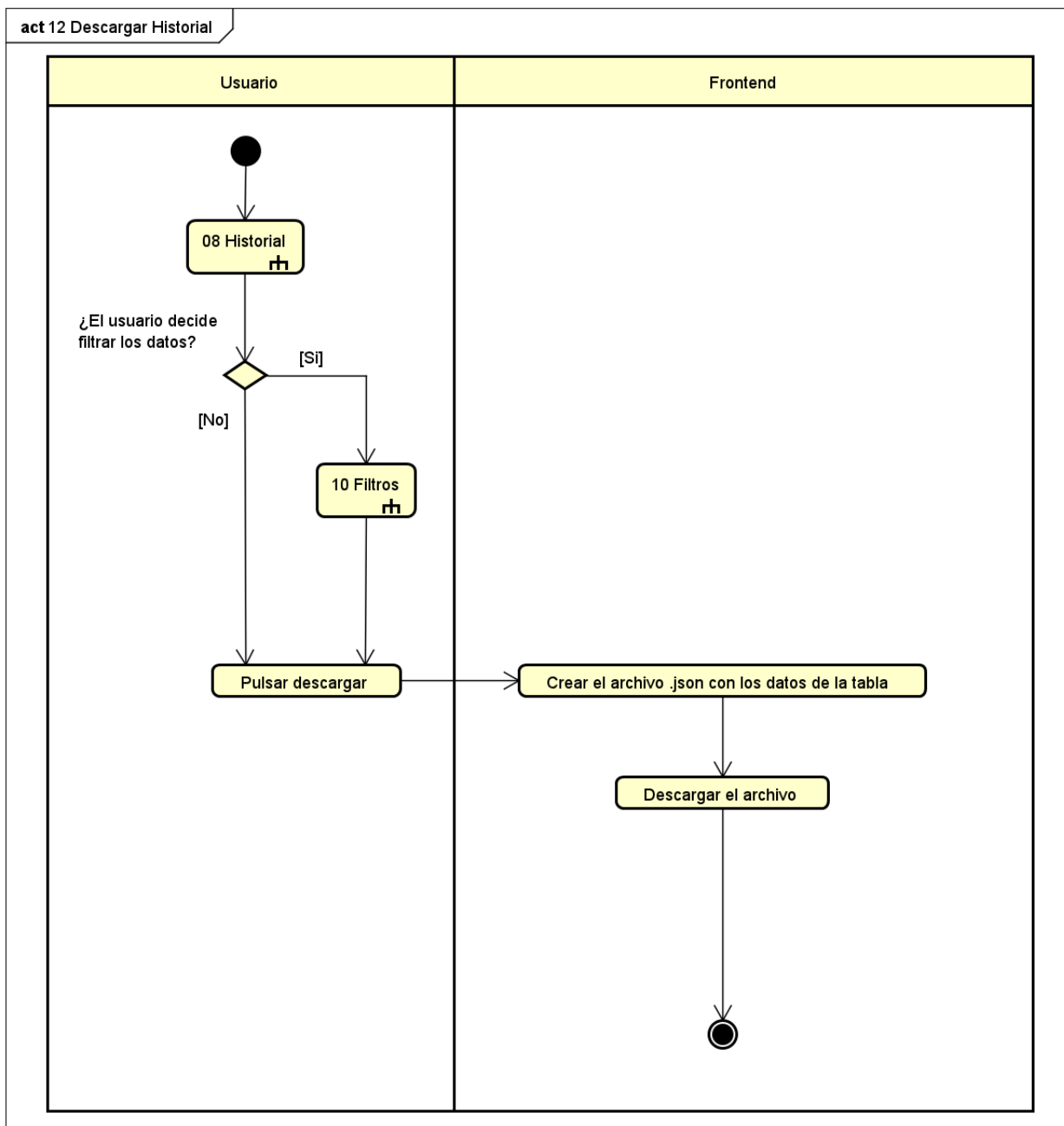


Figura 5.13: Diagrama de actividad de HU12.

5.1.3.13. HU13 - Limpiar resultado

En la figura 5.14 se muestra el diagrama de actividad de la historia de usuario HU13. En ella, tras completar las acciones de la historia HU09 (para que haya un resultado que eliminar), el usuario pulsa el botón de limpiar resultado. En respuesta a esto, el frontend deja en blanco el resultado y rehabilita el botón de iniciar grabación.

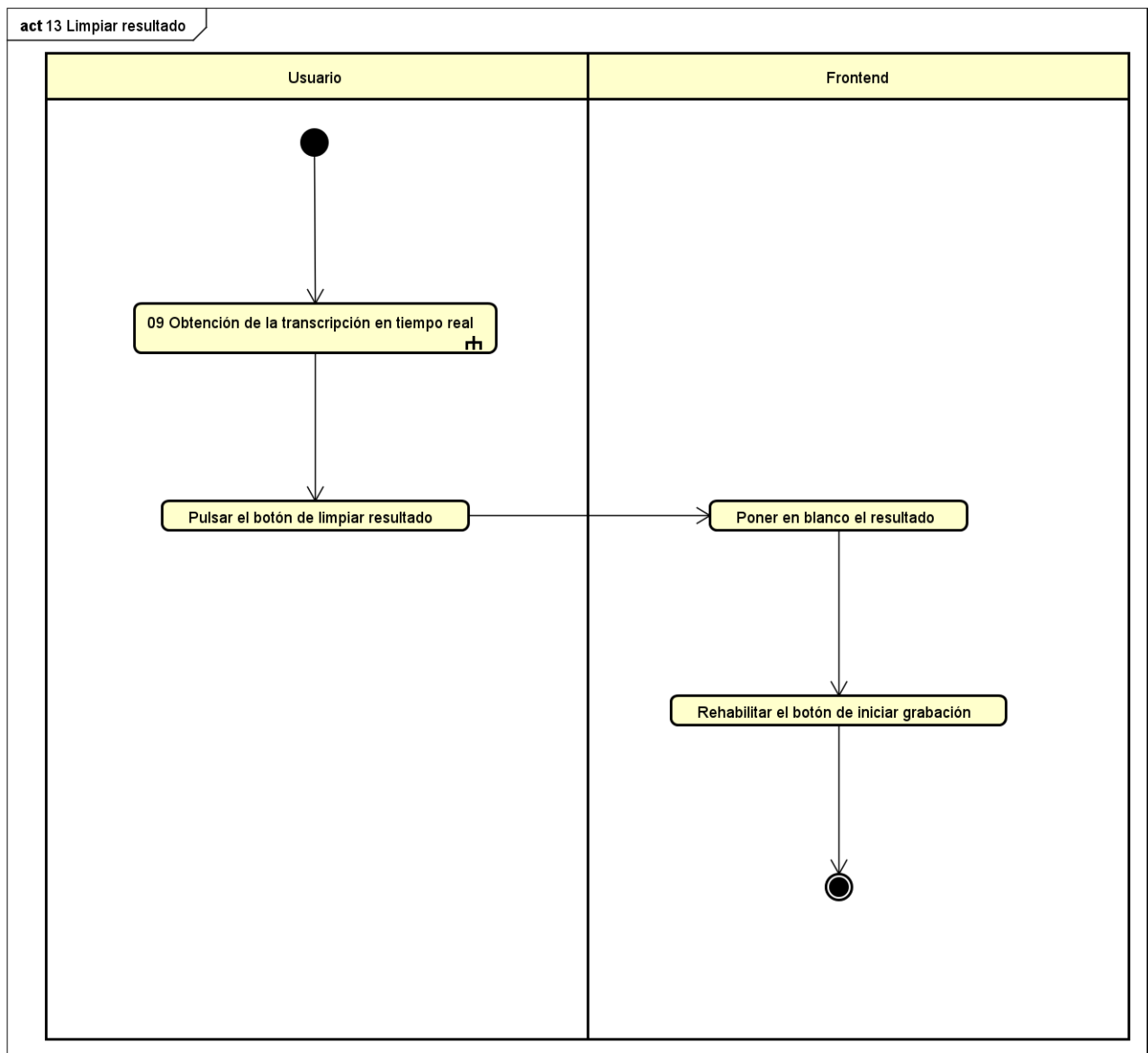


Figura 5.14: Diagrama de actividad de HU13.

5.2. Estructura del proyecto

5.2.1. Estructura del *frontend*

Puesto que se trata de un proyecto Angular, presenta la estructura característica de los mismos, a la que se le han añadido las carpetas y archivos necesarios para implementar la funcionalidad, como los que se utilizan para la traducción.

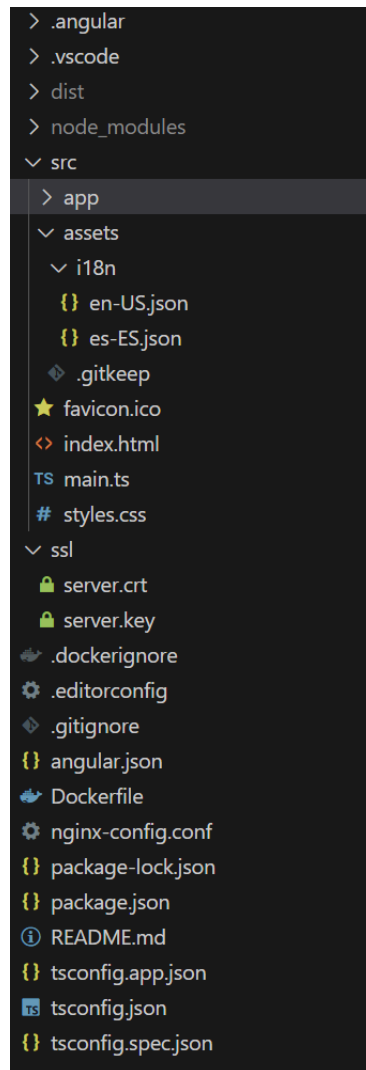


Figura 5.15: Estructura del proyecto Angular.

Muchos de los archivos y

- **.angular**: Guarda archivos generados automáticamente al compilar.
- **.vscode**: Guarda archivos .json que hacen referencia a las preferencias y a la configuración de Visual Studio Code.
- **dist**: Guarda los archivos resultantes de la compilación del proyecto.

Estado final de la aplicación

- **node_modules**: Guarda ficheros de Angular y de las librerías .

- **src**: Contiene toda la parte relevante del proyecto, a excepción de algunos ficheros de configuración. Sus elementos más relevantes son:
 - **app**: Carpeta que contiene el código de la aplicación, como esta carpeta tiene tanta importancia, se explicará mas en detalle en la sección [].
 - **assets**: Contiene una única carpeta llamada i18n, que a su vez contiene los archivos de traducción del proyecto. En este caso solo hay archivos para el inglés estadounidense y el español de España, ambos en formato .json.
 - **styles.css**: Hoja de estilo general de la aplicación.

- **ssl**: Contiene el certificado y la calve necesarios para poder acceder a la aplicación a través de HTTPS.

- **Dockerfile**: Archivo que indica a Docker los pasos que debe seguir para generar una imagen del proyecto.

- **nginx-config**: Archivo de configuración de nginx, que establece algunas de las propiedades que tendrá nuestro servidor web cuando se despliegue desde un contenedor de Docker, como que certificados utilizará.

- **README.md**: Archivo de texto que contiene únicamente: "Este es el repositorio para el TFG del alumno Daniel Figueroa Fernández, Para obtener instrucciones sobre como instalar y desplegar el proyecto, lea el manual de instalación".

El resto de archivos que no se han mencionado son de configuración, y son generados de forma automática, de modo que no son relevantes

5.2.1.1. Estructura de la carpeta app

Esta carpeta es la que contiene el código (y por tanto, la lógica) del proyecto, tiene la siguiente estructura:

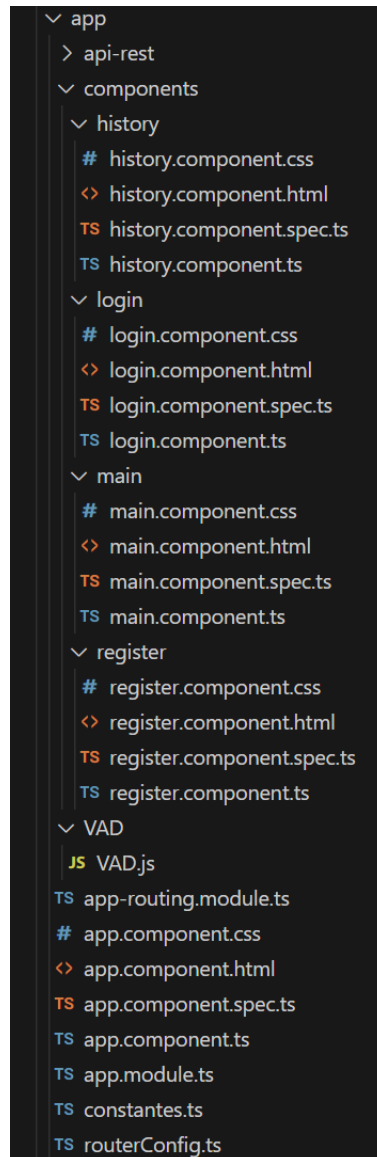


Figura 5.16: Estructura de la carpeta app.

Del contenido de la carpeta destacan los siguientes elementos:

- **api-rest**: Carpeta generada utilizando swagger, contiene las clases con los métodos necesarios para hacer llamadas a la API.
- **src**: Carpeta que contiene los componentes que conforman al aplicación, que son:
 - **history**: Contiene la plantilla HTML, la lógica y la hoja de estilos utilizadas en la vista del historial.
 - **login**: Contiene la plantilla HTML, la lógica y la hoja de estilos utilizadas en la vista de inicio de sesión.
 - **main**: Contiene la plantilla HTML, la lógica y la hoja de estilos utilizadas en la vista principal (en la que se puede grabar audio).

- **register**: Contiene la plantilla HTML, la lógica y la hoja de estilos utilizadas en la vista de registro.
- **VAD**: Contiene el archivo JavaScript que implementa la librería Voice Activity Detection, el contenido de este archivo no ha sido modificado.
- **Dockerfile**: Archivo que indica a Docker los pasos que debe seguir para generar una imagen del proyecto.
- **config.ts**: Archivo que contiene los valores configurables del proyecto, como los valores por defecto de la configuración de la grabación de audio, o la lista de idiomas en los que se puede visualizar la página.

El resto de elementos corresponden al componente por defecto de angular (que en este caso solo incorpora parte de la lógica utilizada para la internacionalización) o son archivos de configuración generados automáticamente.

5.2.2. Arquitectura y patrones de diseño

En este apartado se comentaran los patrones utilizados en el desarrollo del proyecto.

5.2.2.1. Módulo

El patrón módulo [52] se utiliza en la creación de objetos, permitiendo que éstos sean privados o públicos. Esta diferenciación simplifica el acceso, y modificación de los mismos, permitiendo exportar datos de un módulo a otro, pero evitando los problemas de hacer que todas las variables sean de alcance global. Puesto que se aplica al crear cualquier objeto TypeScript (que a su vez lo hereda de JavaScript), se ha utilizado a lo largo de toda la aplicación.

5.2.2.2. Singleton

El patrón singleton [53] de diseño consiste en asegurar la creación de una instancia única de una clase, de tal forma que todas las operaciones que involucren a esa clase lo harán con la misma instancia. Como estamos usando Angular, este patrón se va a aplicar en cualquier proyecto, puesto que el propio Angular lo utiliza para algunas clases de las más importantes. Además veces que este patrón se ha aplicado de forma automática, también se ha utilizado de forma manual para clases como la del servicio de inicio de sesión.

5.2.2.3. Observer

El patrón observer [54] es un patrón de diseño se aplica al utilizar promesas y observadores, que son bastante comunes en TypeScript, especialmente si se utiliza la librería RxJS (como es el caso en este proyecto). Sirve para tener código asíncrono, y poder esperar a que ocurran ciertos eventos, como la finalización de una función. Este patrón se ha aplicado en varios lugares del código, como en el inicio de las grabaciones, los timers, y en las llamadas a la API.

5.3. Diagramas de diseño

5.3.1. Diagrama de paquetes

A continuación, en la figura 5.17 se puede ver una representación de los paquetes que componen la aplicación. El hecho de que no haya una separación por capas está causado por el hecho de que para este proyecto solo se ha desarrollado el frontend, mientras que el backend y su persistencia ya estaban creados, y no formaban parte de los objetivos de este proyecto.

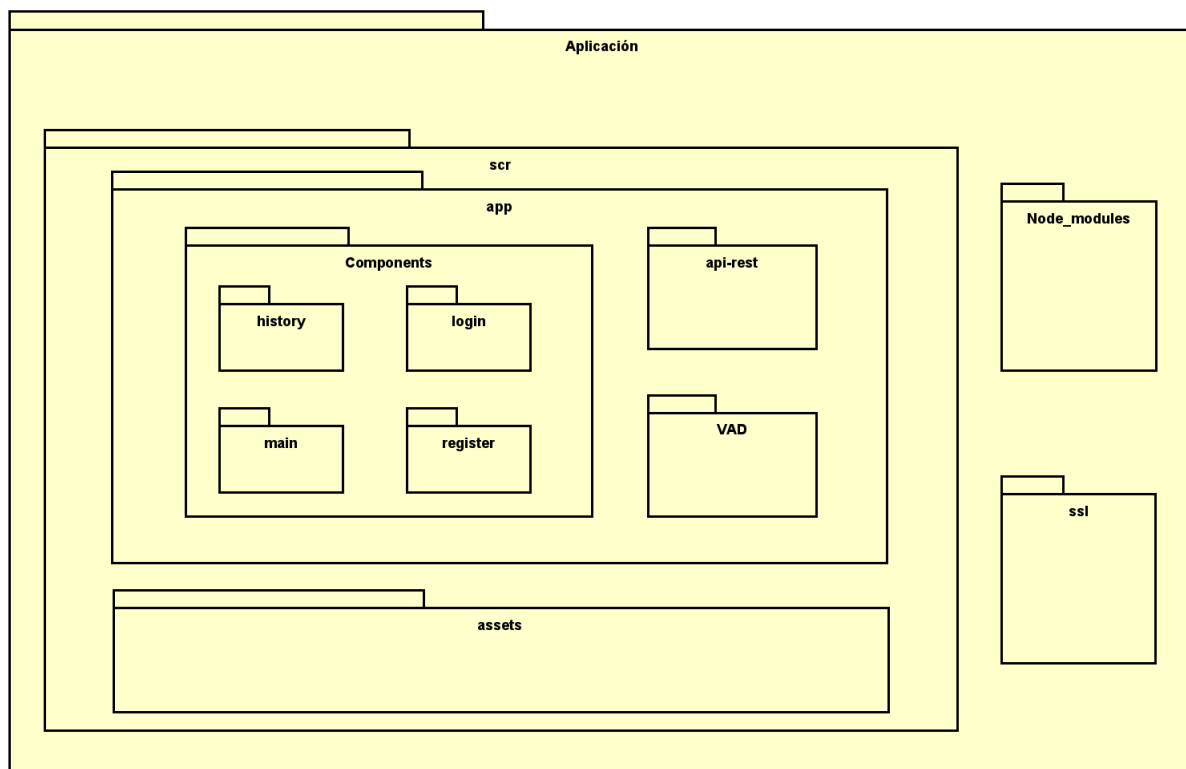


Figura 5.17: Diagrama de paquetes.

Al tratarse de un proyecto formado únicamente por un frontend, la estructura de paquetes que se puede ver en el diagrama es la misma que tenía el proyecto de Angular visto en el apartado anterior. En la aplicación se encuentran las carpetas ssl, node_modules y src. Esta última contiene una carpeta

Estado final de la aplicación

de assets (que guarda los archivos de traducción), y la carpeta app, que a su vez contiene las carpetas correspondientes a la api-rest, el VAD, y los componentes).

5.3.2. Diagrama de despliegue

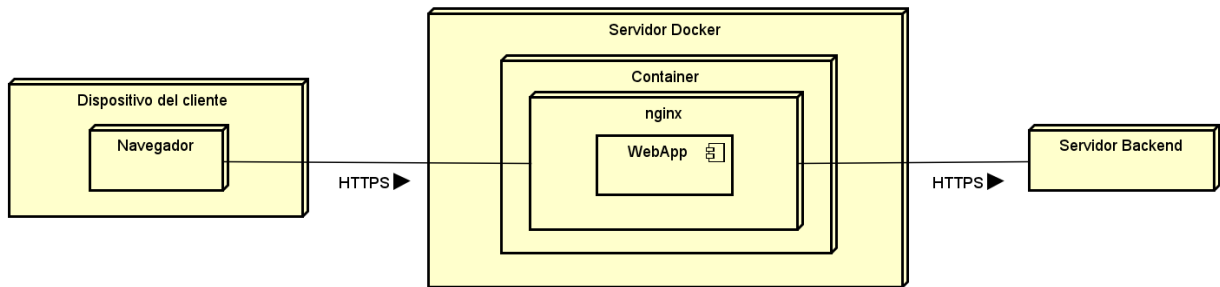


Figura 5.18: Diagrama de despliegue.

Como se puede observar en el diagrama de despliegue de la figura 5.18, un usuario se conectará un navegador en su dispositivo, el cual accederá a la aplicación a través de HTTPS. Esta aplicación se encuentra en un servidor que tiene en ejecución un contenedor de Docker, el cual contiene un servidor web Nginx que está ejecutando la aplicación. Esta aplicación se conectará a través de HTTPS con el servidor de backend para poder realizar las peticiones necesarias a la API. En el apéndice B se encuentra el manual de despliegue del sistema.

Capítulo 6

Pruebas

Si bien durante toda la realización del proyecto se han ido realizando algunas pruebas para comprobar que la funcionalidad implementada hasta el momento funciona correctamente, tras acabar el desarrollo de la aplicación es necesario realizar más pruebas para asegurar que la aplicación cumple con los estándares que se buscan. Para esto, no solo se han realizado pruebas de funcionalidad para comprobar si la aplicación tiene algún error que no permite realizar correctamente algunas de las historias de usuario, sino que también se han realizado test de usabilidad con usuarios reales, para comprobar que el diseño de las interfaces es adecuado.

6.1. Test de funcionalidad

El primer tipo de pruebas que se han realizado son las de funcionalidad, es decir, aquellas que comprueban si hay algún fallo en la aplicación. Estas pruebas se han realizado a mano, probando todas las opciones que ofrece la aplicación. Además, dichas pruebas se han realizado primero en la vista para ordenadores, y después en la de móviles, tanto en una versión local del proyecto, como en una desplegada en un servidor.

El objetivo de repetir las pruebas en el servidor es asegurar que el acceso a través de HTTPS a la página funcionaba correctamente y no influía en el resto de funcionalidad.

Para la realización de cada una de estas pruebas se ha probado el caso en el que todas las acciones se realizan correctamente por parte del usuario, pero también los casos en los que el usuario hace algo mal, como por ejemplo no introducir algún campo en el registro, o poner valores superiores al máximo en la configuración de intensidad.

Como resultado de la realización de estas pruebas, se encontraron validaciones que no se habían tenido en cuenta durante el desarrollo, como los valores erróneos para la intensidad mencionados anteriormente. También gracias a esto se detectó que para evitar problemas, lo mejor era bloquear la configuración mientras se estuvieran grabando audios.

Gracias a probar las vistas de móvil también, se detectó que el botón de cargar más entradas del historial no funcionaba en la misma a pesar de que sí lo hacía en la vista para escritorio, por lo que se pudo corregir.

6.2. Test de usabilidad

Estas pruebas se llevan a cabo con usuarios reales que desconocen la aplicación, y consisten en probar su funcionalidad como si fuera un caso real para ver como se desenvuelven, estos usuarios serán observados mientras realizan estas pruebas para poder anotar observaciones sobre su comportamiento al interactuar con la página. Tras realizar estas pruebas, se aprovechará para pedirles su opinión sobre el diseño y la usabilidad de la página.

Para la realización de estos test se plantean tres pruebas.

1. Consiste en registrarse, iniciar sesión, y cambiar de idioma la página a inglés, (después se les permite volver a cambiar el idioma a español para las siguientes pruebas).
2. Consiste en realizar dos grabaciones, una con los parámetros por defecto, y otra con intensidad 30, guardando el audio, y con el idioma y nombre de grabación que decida el usuario.
3. Consiste en ver el historial de audios, ver los detalles de un audio, filtrar los resultados de la segunda grabación, y descargar el resultado.

Para evaluar cada una de las acciones de los usuarios se se van a utilizar tablas como la siguiente:

Nº	Descripción	¿Completado?	Valoración
Métrica			
Observaciones			

Tabla 6.1: Plantilla utilizada para los test de usabilidad.

- **Nº:** número de la acción a evaluar.
- **Nombre:** nombre descriptivo de la acción.
- **¿Completado?:** indica si el usuario ha sido capaz de completar la acción con éxito.
- **Valoración:** puntuación sobre cómo se ha llevado a cabo la acción (1 a 5).
- **Métrica:** explicación de los valores que se han medido para valorar la acción.
- **Observaciones:** anotaciones sobre el comportamiento del usuario.

Pruebas

Además, tras finalizar todos los escenarios, a cada usuario se le realizarán las siguientes preguntas:

- ¿Te ha resultado intuitiva la interfaz?
- ¿Ha habido algún punto durante la realización de las tareas en el que no supieras como continuar?
- ¿El diseño de la aplicación te ha resultado adecuado?
- ¿Cambiarías o añadirías algo a la interfaz?
- ¿Quieres añadir alguna observación o comentario?

Además, no solo se probará la vista de escritorio, sino que la mitad de los usuarios probarán la vista para ordenador, y la otra mitad la vista para móviles. Después de evaluar la vista que les corresponda, se les mostrará la que no han probado (ordenador o móvil), y tras dejarles probarla un poco, se les realizarán más preguntas:

- ¿Sabrías realizar las tareas que has hecho antes, pero con este diseño de pantallas?
- ¿Cuál de las dos vistas te parece mas clara/intuitiva?

El razonamiento detrás de solo realizar estas preguntas en lugar de repetir las pruebas es que los usuarios estarían realizando las mismas acciones por segunda vez, y por lo tanto, se acordarían de lo que tienen que hacer, lo cual no permite valorar de forma correcta lo intuitiva que es esa vista en concreto.

Estos son algunos datos relevantes de los usuarios utilizados para hacer los test:

Nº	Edad	Nivel de conocimientos informáticos (0-5)	Vista que prueba
1	23	4	Ordenador
2	21	2	Móvil
3	15	2	Ordenador
4	51	3	Móvil

Tabla 6.2: Perfiles de usuario.

Puesto que esta aplicación no está necesariamente orientada hacia expertos, se han escogido perfiles con distinto nivel de conocimiento sobre informática, para ver como esa diferencia de conocimiento afecta al modo en el que interaccionan con la aplicación. Además, ya que se va a probar la versión para móviles, se han escogido dos perfiles muy distintos en cuanto a lo acostumbrados que están a utilizar el móvil.

En el apéndice E se encuentran los resultados de los test realizados por cada usuario.

6.2.1. Conclusiones de los test de usabilidad

Teniendo en cuenta los resultados obtenidos de la realización de estas pruebas, se puede destacar lo siguiente:

- La edad y el conocimiento de informática no parece afectar en gran medida a la tasa de éxito de estos usuarios durante los test.
- Como era esperable, el usuario que no está acostumbrado a utilizar móviles ha tardado más en realizar la mayoría de tareas.
- La única pantalla para la cual un usuario ha sido incapaz de realizar una tarea sin ayuda es la vista del historial, en su formato para móviles.
- La mayoría de los errores han aparecido en la vista del historial.
- Todos los usuarios prefieren la vista para ordenador.
- Los usuarios coinciden en que la tabla en la vista para móviles es demasiado pequeña.

Teniendo estos patrones en cuenta, además de los fallos o dudas puntuales que se han registrado a lo largo de las pruebas, se ha creado una lista con los posibles aspectos a mejorar de cara al futuro:

- El aspecto más claro en el que se puede mejorar la aplicación es en la tabla del historial, en especial en la vista para móviles, como posibles soluciones, a parte de buscar otro modo de mostrar la información de la tabla de forma clara en pantallas pequeñas, se podría hacer que las filas se desplieguen al tocarlas, sin necesidad del botón.
- También se podría mejorar la claridad en cuanto a los filtros, por ejemplo metiéndolos en un desplegable.
- Para evitar confusiones con el selector del idioma de la interfaz, se podía añadir un texto encima de los parámetros de configuración, que deje más claro que se trata de la configuración del audio.
- Otro cambio que podría ayudar a solucionar el problema del punto anterior, pero además ayudar a que el menú esté accesible todo el tiempo, sería ponerlo en un desplegable que se quede fijado en una esquina/lateral de la pantalla.

Capítulo 7

Conclusiones

En este capítulo final se exponen las diferentes ideas que se han ido recopilando a lo largo de la realización del proyecto. Además, se propondrán posibles mejoras y nuevas funcionalidades que puedan ayudar a mejorar el proyecto de cara a futuro.

En este TFG se ha desarrollado un frontend web con diseño modular, que permite la grabación de audio y la obtención de su transcripción en tiempo real, además de la visualización y descarga de un histórico con todos los datos de las grabaciones anteriores. Este frontend permite a los usuarios utilizar de forma fácil y rápida la tecnología de reconocimiento automático de voz que utiliza el servicio backend que se tomó como base del proyecto, ya sea con fines educativos, como la mejora de la pronunciación de un idioma que se está aprendiendo, o con fines investigativos.

Durante el proceso de desarrollo del proyecto se han descubierto nuevas herramientas y tecnologías, desde aquellas que se empezaron a utilizar en la primeras iteraciones para grabar audio por primera vez, hasta las que se aprendieron a utilizar durante las últimas etapas del desarrollo, como el uso de nginx para realizar las funciones de servidor web. Además, el aprendizaje no solo ha aparecido en forma de nuevas tecnologías que conocer y empezar a utilizar, sino que también se ha manifestado en forma de una mejora sustancial en el manejo de algunas de estas tecnologías que ya se habían utilizado durante la carrera, afianzando y ampliando los conocimientos que se tenían sobre las mismas. De todo este aprendizaje destacaría la mejora sustancial en el uso de Angular, que ha sido la herramienta principal utilizada para desarrollar el proyecto, y el mayor nivel de comprensión obtenido sobre el funcionamiento de Docker.

Pero el proyecto no solo se compone del desarrollo de la aplicación, sino que ha tenido una parte muy importante de planificación: desde la elección de una metodología adecuada para la situación, como la iterativa e incremental, que me ha permitido flexibilizar el tiempo de trabajo, además de poder presentar progreso en la aplicación final de forma periódica para recibir retroalimentación, hasta la planificación de las iteraciones y los riesgos, la dificultad de hacerla bien, y las consecuencias de que dicha planificación no sea del todo correcta.

Como última parte que quiero destacar, creo que es importante mencionar la realización de los test de usabilidad, estas pruebas con usuarios reales han arrojado luz a problemas que eran imposibles de

detectar desde el punto de vista del desarrollador, y las conclusiones sacadas de los mismos son una parte importante de las mejoras que hubiera sido interesante poder implementar, pero que tendrán que dejarse como trabajo a futuro por falta de tiempo.

Finalmente, en lo personal, la realización de este proyecto me ha resultado muy enriquecedora, desde planificar un proyecto con una duración de varios meses, hasta aplicar los conocimientos obtenidos durante la carrera a un caso real, y por supuesto, la obtención de experiencia y nuevos conocimientos mencionada anteriormente que serán de gran ayuda de cara a una futura carrera profesional. Aunque durante algunas partes de la realización del proyecto se encontraron una mayor dificultad y carga de trabajo que las esperadas, al final este proyecto ha resultado ser una experiencia muy gratificante, a la par que beneficiosa.

7.1. Trabajo futuro

En este apartado se recogen las áreas de mejora en las que se podría trabajar de cara a futuro para mejorar la aplicación, y que no se han podido implementar a lo largo de este proyecto por diversos motivos, desde la falta de tiempo, hasta la imposibilidad de realizar las modificaciones necesarias en el servicio de backend.

- **Mejoras en la interfaz:** principalmente aquellas mencionadas como conclusiones de las pruebas de usabilidad, para minimizar las dudas de los usuarios sobre como funciona la aplicación.

- **Nueva funcionalidad:** Algunas ideas sobre la funcionalidad que se podría haber implementado en caso de poder modificar el servicio de backend (o que éste ya tuviera la funcionalidad necesaria implementada) incluyen añadir la opción de borrar elementos del historial, o permitir escuchar los audios que se han decidido guardar en el servidor.

- **Mejorar la seguridad:** Con cambios como encriptar al contraseña al mandarla al servidor en la petición del login, y que este sea capaz de decodificarla.

Apéndices

Apéndice A

Acrónimos

- **API:** Application Programming Interface.
- **ASR:** Automatic Speech Recognition.
- **CLI:** Command Line Interface.
- **CLST:** Centre for Language and Speech Technology.
- **CPU:** Central Processing Unit.
- **CSS:** Cascading Style Sheets.
- **CTC:** Connectionist Temporal Classification.
- **ECA-SIMM:** Grupo de investigación de entornos de computación avanzada y sistemas de interacción multimodal.
- **GDPR:** General Data Protection Regulation.
- **GPU:** Graphics Processing Unit.
- **HTML:** Hypertext Markup Language.
- **HTTP:** Hypertext Transfer Protocol.
- **HTTPS:** Hypertext Transfer Protocol Secure.
- **IDE:** Integrated Development Environment.
- **IVA:** Impuesto sobre el Valor Añadido.
- **JSON:** JavaScript Object Notation.
- **JWT:** JSON Web Token.
- **TFG:** Trabajo de Fin de Grado.
- **RAM:** Random Access Memory.

-
- **REST**: Representational State Transfer.
 - **SSD**: Solid State Drive.
 - **SSL**: Secure Sockets Layer.
 - **STT**: Speech-to-text.
 - **VAD**: Voice Activity Detection.
 - **VAR**: Voice Activity Recognition.
 - **VSC**: Visual Studio Code.

Apéndice B

Manual de despliegue

En esta sección se detallan los pasos a seguir para la instalación y el despliegue del proyecto. Es importante destacar que en este caso el despliegue se ha realizado sobre una máquina virtual con un sistema operativo Linux (Ubuntu 22.04.2 LTS), por lo que puede haber algunas diferencias al realizar el despliegue en otros sistemas.

B.1. Instalación de git y clonación del repositorio

Todos los comandos que se indican a continuación se ejecutan desde una terminal de Linux. Lo primero que haremos será actualizar apt:

```
sudo apt-get update
```

Para poder clonar el repositorio es necesario tener instalado git, y para ello utilizaremos el siguiente comando:

```
sudo apt-get install git
```

Una vez tengamos instalado git, lo siguiente es situarnos en el directorio en el que queremos que se descargue el proyecto, y ejecutar:

```
git clone https://gitlab.inf.uva.es/danfigu/audioservervfgfront.git
```

Este comando puede tardar unos momentos, pero cuando finalice tendremos la última versión del proyecto descargada.

B.2. Instalación de docker

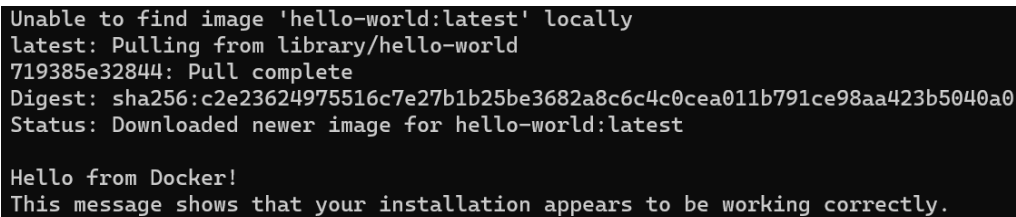
Para facilitar el despliegue de la aplicación, además de para permitirnos usar sus comandos para hacer algunos pequeños ajustes como cambiar el puerto en el que desplegamos nuestra aplicación, vamos a utilizar Docker. Para instalarlo vamos a utilizar los siguientes comandos:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

El primero de estos comandos obtendrá un script que proporciona el propio Docker para poder instalarlo de una forma más conveniente, mientras que el segundo lo ejecuta. Para poder ejecutar este segundo comando es necesario tener permisos para poder utilizar sudo. Una vez que se complete la ejecución de estos comandos, Docker ya debería estar instalado en el sistema. Para comprobarlo, podemos utilizar el comando:

```
sudo docker run hello-world
```

Que descarga una imagen de prueba, la ejecuta en un contenedor que muestra un mensaje de confirmación y termina. El resultado debería ser algo similar a esto:



```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:c2e23624975516c7e27b1b25be3682a8c6c4c0cea011b791ce98aa423b5040a0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Figura B.1: Resultado del comando docker run hello-world.

Lo siguiente que vamos a hacer es crear un grupo Unix llamado docker, y a añadir nuestro usuario a dicho grupo. Esto nos permitirá utilizar los comandos de Docker sin la necesidad de añadir sudo al inicio.

```
sudo groupadd docker
sudo usermod -aG docker <tu_nombre_de_usuario>
```

Tras ejecutar estos comandos, cerrar sesión y volverla abrir para que se actualicen los cambios, la instalación debería estar completa, y deberíamos poder ejecutar los comandos de Docker sin necesidad de sudo. Para comprobarlo se puede utilizar el comando:

```
docker run hello-world
```

Que debería dar un resultado como el que obtuvimos al lanzar el comando anteriormente utilizando sudo.

B.3. Creación de la imagen de Docker y despliegue en un contenedor

Si queremos cambiar la configuración del proyecto, por ejemplo para añadir una nueva traducción a la página, este es el momento para hacerlo. Las instrucciones sobre como realizar dichas modificaciones a la configuración se encuentran en el apéndice C.

Una vez que tenemos tanto el proyecto descargado y configurado como la instalación de Docker completada, podemos pasar a desplegar la aplicación. Para ello, el primer paso es crear la imagen de Docker, para lo cual debemos abrir una terminal de linux y situarnos dentro del directorio que descargamos en la sección B.1 (*audioservvertfgfront*).

```
cd audioservvertfgfront
```

Una vez que nos encontremos en el directorio del proyecto, utilizaremos el siguiente comando para generar la imagen del proyecto:

```
docker build -t audioservvertfgfront .
```

El proceso de crear la imagen puede tardar varios minutos, pero una vez que se haya completado, la imagen debería aparecer en el listado de imágenes de Docker, el cual podemos ver utilizando:

```
docker image ls
```

Una vez que ya se ha creado la imagen, el último paso es desplegarla en un contenedor, para lo cual utilizaremos el comando:

```
docker run -d -p 80:443 audioservvertfgfront
```

Este comando necesita un poco más de explicación, ya que podemos cambiar algunas de sus partes dependiendo de lo que necesitemos:

- **-d** Indica que el contenedor se ejecutará en segundo plano, y que mostrará por consola el id del contenedor, si preferimos que no se haga, podemos eliminar este modificador.
- **-p x:y** Enlaza el puerto 'y' del contenedor al 'x' del host. El puerto 80 es el puerto por defecto para las peticiones HTTP y el 443 el de las peticiones HTTPS. De modo que si se ejecuta el comando como en el ejemplo, las peticiones HTTPS que llegan por defecto al puerto 443 del host acaban en el 80 del contenedor. Al elegir estos valores es muy importante saber que para que la grabación de audio funcione, se tiene que acceder a la aplicación a través de HTTPS, es decir, el puerto 443.

Tras ejecutar este último comando la aplicación estará desplegada, y podremos acceder a ella a través de localhost utilizando del puerto que hayamos elegido, esto nos puede servir para comprobar

B.3. Creación de la imagen de Docker y despliegue en un contenedor

que todo funciona correctamente. Además podremos ver el contenedor que se ha creado en la lista de contenedores de Docker, la cual podemos ver al utilizar el comando:

```
docker container ls
```

Que devolverá algo parecido a lo mostrado en la siguiente imagen:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f7cef9c580d2	audioservertfgfront	"nginx -g 'daemon of..."	29 hours ago	Up 29 hours	0.0.0.0:20272->80/tcp	crank
y_yalow						

Figura B.2: Resultado del comando docker container ls.

Apéndice C

Manual de configuración

En este apéndice se detallan los pasos necesarios para modificar la configuración del proyecto. Cualquier modificación en la configuración debe realizarse previa al despliegue del mismo para que ésta tenga efecto.

Independientemente de lo que queramos modificar, el primer paso que debemos seguir es acceder al directorio del proyecto (ver Apéndice B.), con el comando

```
cd audioservertfgfront
```

A partir de aquí el procedimiento cambia dependiendo de lo que queramos hacer, las tres opciones principales de configuración son las siguientes:

- Cambiar los valores por defecto del proyecto, como la duración máxima de los audios.
- Modificar la lista de traducciones de la página.
- Cambiar el certificado SSL.

C.1. Cambiar los valores por defecto

Para ello solo hay que modificar el archivo llamado *config.ts*, que se encuentra en *./src/app/* y que contiene 9 parámetros:

- **timeAutoSend:** Indica cada cuanto tiempo se enviará el audio al servidor como máximo (en ms).
- **timeInactiveSend:** Indica el tiempo en ms que tiene que pasar desde que el usuario se calla hasta que se envía el audio al servidor.
- **beam:** Indica el valor por defecto del nivel de intensidad (un mayor valor indica mayores tiempos de respuesta, pero una transcripción más precisa), su valor máximo es 30.

- **language:** Indica el valor por defecto del idioma.
- **save:** Indica si por defecto se guardan o no los audios en el servidor (1 indica si y 0 no).
- **limiteHistorial:** Indica la cantidad de entradas del historial que se cargan a la vez.
- **filtrarSinResult:** Indica si se ocultan por defecto las entradas del historial que no tienen resultado.
- **idiomas:** Es una lista con los idiomas en los que estará disponible la página y su código, antes de modificarlo, leer el apartado C.2
- **idiomaDefault:** Es el código del idioma en el que se mostrará la página por defecto, debe estar en la lista de idiomas.

C.2. Cambiar lista de traducciones

La lista de las posibles traducciones de la página se encuentra en el archivo `./src/app/config.ts`, bajo el nombre de idiomas. Para eliminar un idioma es tan sencillo como borrarlo de la lista, siempre y cuando el idioma en cuestión no sea el `idiomaDefault`, en cuyo caso habrá que modificar este campo para elegir otro.

Para añadir un nuevo idioma primero es necesario crear un nuevo archivo de traducción, con el nombre `<código-del-idioma>.json`, por ejemplo, `en-US.json` para el inglés estadounidense, o `es-ES.json` para el español de España. Este archivo se deberá situar en la carpeta `./src/assets/i18n/`. En cuanto al contenido, esta es una parte del fichero de traducción del inglés estadounidense:

```
"history.beam": "Recognition level",
"history.channel": "Channel",
"history.code": "Code",
"history.createdAt": "Created at",
"history.downloadAsJSON": "Download as JSON",
"history.duration": "Duration",
"history.filename": "Filename",
"history.filter": "Filter",
"history.keep": "Keep audio",
"history.loadMore": "Load {{limite}} more",
```

La parte de la izquierda de los ':' es un identificador del texto, por ejemplo `"history.downloadAsJSON"` indica el texto del botón de descargar el historial como archivo `.json`. Estos identificadores no se deben modificar, y son los mismos para todos los idiomas, por lo que lo más sencillo es copiarlos de un archivo de traducción de otro idioma.

En cuanto a la parte de la derecha de los ':', se trata de la traducción, es decir, es el texto que se mostrará en el lugar indicado por la parte de la izquierda. Las únicas partes que no se deben modificar

de estos textos son las que se encuentran entre llaves, como `{{limite}}` en la última línea del ejemplo. Estas partes corresponden a nombres de variables dentro del código, y cambiarlas provocará fallos.

Una vez que ya tengamos nuestro archivo de traducción creado, lo siguiente es añadirlo a la lista, como ya se ha mencionado antes, esta lista se encuentra en el archivo `./src/app/config.ts`, bajo el nombre de idiomas. Cada elemento de esta lista se compone de dos partes:

- **code:** código del idioma según la normativa BCP 47 (de RFC 4646), debería coincidir con el nombre del archivo de traducción, pero sin la extensión `.json`
- **label:** nombre con el que aparecerá en el selector de idiomas

Una vez que hayamos añadido el idioma a la lista y redespuguemos la aplicación ya estará disponible desde el selector de idiomas.

C.3. Cambiar el certificado SSL

El certificado que se encuentra por defecto en el proyecto es un certificado autofirmado, por lo que los navegadores lanzarán avisos de que la conexión no es segura la intentar acceder a la página a través de HTTPS (lo cual es necesario para poder grabar audio). Si se quiere cambiar este certificado por uno que no cause estos problemas, estos son los pasos a seguir:

- Sustituir el certificado y la clave por los nuevos: los actuales se encuentran en la carpeta `./ssl`, con los nombres de `server.crt` y `server.key` respectivamente.
- En caso de que los nombres de los archivos nuevos no sean los mismos que los actuales, es necesario reflejar estos cambios en la configuración de nginx, modificando el archivo `./nginx-config.conf` para que su contenido sea el siguiente:

```
server {
    listen          443 ssl;
    ssl_certificate  "ssl/<<nombre-nuevo-certificado>>.crt";
    ssl_certificate_key "ssl/<<nombre-nueva-clave>>.key";
    root /usr/share/nginx/html;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

- Del mismo modo que en el punto anterior, es necesario actualizar los nombres de los archivos del certificado y la clave, pero esta vez en el archivo `./angular.json`.

Apéndice D

Manual de usuario

La primera pantalla que se va a ver al acceder a la aplicación es la de inicio de sesión (figura D.1), en ella podremos utilizar el selector de idioma de la parte superior para elegir el idioma de la página, iniciar sesión introduciendo nuestro nombre de usuario y contraseña si ya tenemos cuenta, o acceder a la página de registro para crear una nueva.



Español ▾

Iniciar sesión

Nombre de usuario o correo electrónico

Contraseña

Iniciar sesión

Registrarse

Audioserver - (C) Daniel Figueroa Fernández. 2023

Figura D.1: Pantalla de inicio de sesión.

Si accedemos a la página de registro (figura D.2) se nos presenta una vista muy similar a la anterior, pero que esta vez nos pide nombre de usuario, email y contraseña. Tras introducirlos y pulsar el botón de registro, si todos estos datos son válidos se nos enviará un correo electrónico para poder confirmar nuestra cuenta. Una vez que accedamos al enlace que aparecerá en este correo ya tendremos nuestra cuenta activa, y podremos volver a la página de inicio de sesión para acceder a la aplicación.

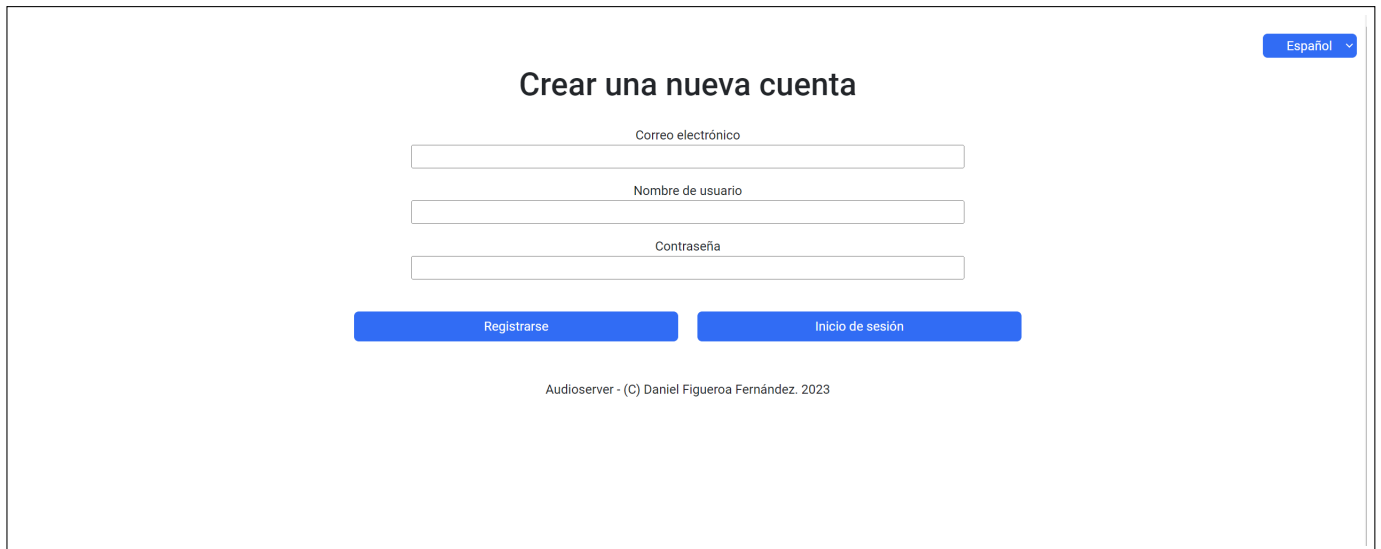


Figura D.2: Pantalla de registro.

Tras iniciar sesión se nos redirigirá a la página principal (figura D.3). Una vez estemos en esta página, en el menú de la zona superior derecha se nos muestra nuestro nombre de usuario, y tenemos botones para ir al historial, cambiar el idioma de la página, y cerrar sesión. En el resto de la vista tenemos las diferentes opciones relacionadas con la grabación de audio y la obtención de su transcripción en tiempo real. A la izquierda podemos ver los selectores del idioma para el cual queremos obtener la transcripción y de la intensidad (que debe estar entre 1 y 30, y proporciona transcripciones más precisas, a cambio de un mayor tiempo de respuesta), junto con la opción para decidir si queremos que el archivo de audio que grabemos se guarde en el servidor. Además, en la parte central de la pantalla podemos añadirle un nombre a la grabación. Estos cuatro campos no se podrán modificar en mitad de una grabación, por lo que si queremos cambiarlos deberemos parar la grabación actual y cambiarlos antes de empezar una nueva.



Figura D.3: Pantalla de inicio.

Para grabar un audio simplemente hay que pulsar el botón de iniciar grabación y comenzar a hablar, y al poco tiempo empezará a mostrarse en la caja de texto la transcripción de lo que hemos dicho. Además, a la derecha de dicha caja de texto tendremos un indicador del volumen con el que estamos hablando, y otro del progreso de la subida del audio al servidor. Para parar la grabación simplemente tenemos que pulsar el botón de detener grabación que sustituirá al de iniciarla, como se muestra en la figura D.4.



Figura D.4: Pantalla de inicio mientras se graba un audio.

Para borrar el resultado de la pantalla solo hay que pulsar el botón limpiar, lo cual puede llegar a ser necesario en caso de llegar al límite de 500 caracteres del resultado, en cuyo caso se parará la grabación automáticamente y se mostrará un mensaje como el siguiente:

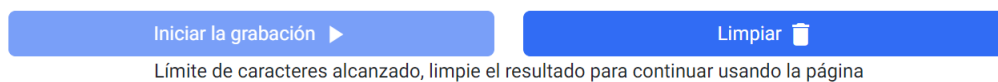



Figura D.5: Aviso que se muestra al superar el límite de caracteres.

La última vista que forma parte de la aplicación es el historial (figura D.6). Esta vista tiene un menú en la parte superior derecha que funciona igual que el de la vista principal, pero sustituyendo la opción de ir al historial por la de volver a la página de inicio. Además, cuenta con una tabla que contiene la información de cada audio que ha grabado el usuario, pudiendo ver los detalles de cada palabra de ese audio pulsando el botón "+" situado en el lado izquierdo de la tabla.



 daniel1figuer

Inicio

Español ▼

Cerrar sesión

Fecha de creación: <input style="width: 90%;" type="text"/>	Fecha de actualización: <input style="width: 90%;" type="text"/>	Nombre de la grabación: <input style="width: 90%;" type="text"/>
Nombre del archivo: <input style="width: 90%;" type="text"/>	Resultado: <input style="width: 90%;" type="text"/>	Solo mostrar entradas con resultado: <input checked="" type="checkbox"/>

Filtro

Código	Nombre de la grabación	Resultado	Guardar audio	Intensidad	Nombre del archivo	Fecha de creación	Fecha de actualización
+ 9	string	HOLA SOY DANIEL ESTOY PROBANDO EL DESPEJE	true	12	1678201800993.wav	2023-03-07T15:10:01.225Z	2023-03-07T15:12:21.767Z
- 9	texto	PROBANDO PROA	true	12	1680515662763.wav	2023-04-03T09:54:22.920Z	2023-04-03T09:55:31.376Z
Canal	Inicio	Duración	Palabra	Puntuación			
1	0	0.49	PROBANDO	0.79			
1	0.5	0.17	PROA	0.87			
+ 9	texto	PROBANDO APROBANDO OJO PARA QUE FUNCIONE TODO	true	12	1680517138097.wav	2023-04-03T10:18:58.315Z	2023-04-03T10:19:28.801Z

- 9	texto	PROBANDO PROA	true	12	1680515662763.wav	2023-04-03T09:54:22.920Z	2023-04-03T09:55:31.376Z
Canal	Inicio	Duración	Palabra	Puntuación			
1	0	0.49	PROBANDO	0.79			
1	0.5	0.17	PROA	0.87			
+ 9	texto	PROBANDO APROBANDO OJO PARA QUE FUNCIONE TODO	true	12	1680517138097.wav	2023-04-03T10:18:58.315Z	2023-04-03T10:19:28.801Z
+ 9	por que eres obligatorio	PARA PROBANDO	true	12	1680605850006.wav	2023-04-04T10:57:30.268Z	2023-04-04T10:57:32.873Z
+ 9	pruebas	HOLA PROBANDO QUE TAL	true	12	1680605943688.wav	2023-04-04T10:59:03.908Z	2023-04-04T10:59:06.491Z
+ 9	pruebas	<UNK> PRUEBA BUENOS DÍAS	true	12	1680606089732.wav	2023-04-04T11:01:29.954Z	2023-04-04T11:01:32.512Z
+ 3	pruebas	true	true	12	1680606104634.wav	2023-04-04T11:01:44.709Z	2023-04-04T11:01:46.940Z
+ 3	pruebas	mhm	true	12	1680606109716.wav	2023-04-04T11:01:49.803Z	2023-04-04T11:01:51.635Z
+ 3	pruebas	true	true	12	1680606114710.wav	2023-04-04T11:01:54.796Z	2023-04-04T11:01:56.582Z
+ 3	pruebas	huh	true	12	1680606119781.wav	2023-04-04T11:01:59.910Z	2023-04-04T11:02:01.726Z

Items per page: 10 1 - 10 of 417 < >

Descargar JSON
Se han cargado las primeras 1000 entardas de 3026: Cargar 1000 más

Audioserver - (C) Daniel Figueroa Fernández. 2023

Figura D.6: Pantalla del historial.

En la parte superior de la página se pueden ver los filtros, para usarlos primero hay que rellenar los campos por los que se quiera filtrar, y después pulsar el botón "Filtro". Por último, tenemos los dos botones que se encuentran bajo la tabla, el de la izquierda descarga el contenido de la tabla a un archivo .json (si el contenido de la tabla está filtrado, el contenido de este archivo también estará filtrado), mientras que el botón de la derecha cargará más entradas de la tabla. Este último solo aparecerá cuando no se hayan cargado todos los audios que hay en el servidor (solo se cargan automáticamente los 1000 primeros).

Apéndice E

Pruebas de usabilidad con usuarios reales

En este apéndice se muestran los resultados obtenidos durante la realización de las pruebas de usabilidad.

E.1. Usuario 1

Este usuario tiene el perfil de una persona joven, acostumbrada a usar la informática, pero sin llegar a ser un experto, probará la vista para ordenador.

E.1.1. Prueba 1

En las siguientes tablas se muestran los resultados obtenidos por el usuario 1 en la prueba 1:

Nº: 1	Nombre: Registrarse	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 20 segundos- Clicks/toques necesarios(mínimo 5): 5- Errores: No			
Observaciones <ul style="list-style-type: none">- Ninguna			

Tabla E.1: Resultados del usuario 1 en la acción 1.

Nº:2	Nombre: Iniciar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 10 segundos - Clicks/toques necesarios(mínimo 3): 3 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.2: Resultados del usuario 1 en la acción 2.

Nº:3	Nombre: Cambiar idioma de la página	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 2 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.3: Resultados del usuario 1 en la acción 3.

E.1.2. Prueba 2

En las siguientes tablas se muestran los resultados obtenidos por el usuario 1 en la prueba 2:

Nº:1	Nombre: Hacer una grabación con los valores por defecto	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 30 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.4: Resultados del usuario 1 en la acción 1.

Nº:2	Nombre: Borrar el resultado	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.5: Resultados del usuario 1 en la acción 2.

Nº:3	Nombre: Cambiar la configuración de la grabación (Idioma, intensidad, guardar audio y nombre de la grabación)	Completado: Sí	Valoración: 3
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 5 segundos - Clicks/toques necesarios(mínimo 5): 6 - Errores: Pulsó el selector de idioma de la página, en lugar del de la configuración 			
Observaciones			
<ul style="list-style-type: none"> - Se dio cuenta del error rápidamente y lo pudo corregir sin ayuda 			

Tabla E.6: Resultados del usuario 1 en la acción 3.

Nº:4	Nombre: Hacer una grabación con los nuevos valores	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 25 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones <ul style="list-style-type: none"> - Ninguna 			

Tabla E.7: Resultados del usuario 1 en la acción 4.

E.1.3. Prueba 3

En las siguientes tablas se muestran los resultados obtenidos por el usuario 1 en la prueba 3:

Nº:1	Nombre: Ver el historial	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 3 segundos- Clicks/toques necesarios(mínimo 1): 1- Errores: No			
Observaciones <ul style="list-style-type: none">- Ninguna			

Tabla E.8: Resultados del usuario 1 en la acción 1.

Nº:2	Nombre: Ver detalles de un audio	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 4 segundos- Clicks/toques necesarios(mínimo 1): 1- Errores: No			
Observaciones <ul style="list-style-type: none">- El usuario duda un poco, pero rápidamente se da cuenta de lo que tiene que hacer			

Tabla E.9: Resultados del usuario 1 en la acción 2.

Nº:3	Nombre: Filtrar segunda grabación	Completado: Sí	Valoración: 2
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 25 segundos- Clicks/toques necesarios(mínimo 2): 4- Errores: El usuario en un principio intentó filtrar por nombre de archivo			
Observaciones <ul style="list-style-type: none">- En cuanto se dio cuenta de su error, realizó la acción sin problemas			

Tabla E.10: Resultados del usuario 1 en la acción 3.

Nº:4	Nombre: Descargar JSON	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.11: Resultados del usuario 1 en la acción 4.

Nº:5	Nombre: Cerrar sesión	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 4 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - El usuario se encontraba en la parte baja de la vista del historial, por lo que no veía el botón de cerrar sesión hasta que volvió a la parte de arriba 			

Tabla E.12: Resultados del usuario 1 en la acción 5.

E.1.4. Preguntas

En esta sección se muestran las respuestas proporcionadas por el usuario 1 a una serie de preguntas que se le realizaron tras la realización de las pruebas:

- ¿Te ha resultado intuitiva la interfaz? Sí, en general era bastante clara
- ¿Ha habido algún punto durante la realización de las tareas en el que no supieras como continuar? No, tras equivocarme vi en seguida cómo se hacía
- ¿El diseño de la aplicación te ha resultado adecuado? Sí, en general sí
- ¿Cambiarías o añadirías algo a la interfaz? Algo que indique que el selector de idioma de la izquierda es el del audio
- ¿Quieres añadir alguna observación o comentario? No

Después, se les mostró la vista que no había probado (móvil), y tras probarla un poco, respondió lo siguiente:

- ¿Sabrías realizar las tareas que has hecho antes, pero con este diseño de pantallas? Sí, es muy parecido
- ¿Cuál de las dos vistas te parece mas clara/intuitiva? Las dos por igual, aunque la vista de ordenador es más cómoda para ver la tabla en una pantalla grande.

E.2. Usuario 2

Este usuario tiene el perfil de una persona joven, pero con pocos conocimientos de informática, pero que está muy acostumbrado a usar el móvil. probará la vista para móviles.

E.2.1. Prueba 1

En las siguientes tablas se muestran los resultados obtenidos por el usuario 2 en la prueba 1:

Nº: 1	Nombre: Registrarse	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 40 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.13: Resultados del usuario 2 en la acción 1.

Nº:2	Nombre: Iniciar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 20 segundos - Clicks/toques necesarios(mínimo 3): 3 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.14: Resultados del usuario 2 en la acción 2.

Nº:3	Nombre: Cambiar idioma de la página	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 2 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.15: Resultados del usuario 2 en la acción 3.

E.2.2. Prueba 2

En las siguientes tablas se muestran los resultados obtenidos por el usuario 2 en la prueba 2:

Nº:1	Nombre: Hacer una grabación con los valores por defecto	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 40 segundos - Clicks/toques necesarios(mínimo 2): 3 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Le pone nombre a la grabación aunque no se le pida - Tarda unos segundos en darse cuenta de que tiene que parar la grabación 			

Tabla E.16: Resultados del usuario 2 en la acción 1.

Nº:2	Nombre: Borrar el resultado	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.17: Resultados del usuario 2 en la acción 2.

Nº:3	Nombre: Cambiar la configuración de la grabación (Idioma, intensidad, guardar audio y nombre de la grabación)	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 5 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.18: Resultados del usuario 2 en la acción 3.

Nº:4	Nombre: Hacer una grabación con los nuevos valores	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 20 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones <ul style="list-style-type: none"> - Ninguna 			

Tabla E.19: Resultados del usuario 2 en la acción 4.

E.2.3. Prueba 3

En las siguientes tablas se muestran los resultados obtenidos por el usuario 2 en la prueba 3:

Nº:1	Nombre: Ver el historial	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.20: Resultados del usuario 2 en la acción 1.

Nº:2	Nombre: Ver detalles de un audio	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 20 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- El usuario tarda en encontrar el botón			

Tabla E.21: Resultados del usuario 2 en la acción 2.

Nº:3	Nombre: Filtrar segunda grabación	Completado: No	Valoración: 1
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 60 segundos - Clicks/toques necesarios(mínimo 2): 10 - Errores: El usuario no consigue realizar la tarea sin ayuda 			
Observaciones			
- Ninguna			

Tabla E.22: Resultados del usuario 2 en la acción 3.

Nº:4	Nombre: Descargar JSON	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 2 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.23: Resultados del usuario 2 en la acción 4.

Nº:5	Nombre: Cerrar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.24: Resultados del usuario 2 en la acción 5.

E.2.4. Preguntas

En esta sección se muestran las respuestas proporcionadas por el usuario 2 a una serie de preguntas que se le realizaron tras la realización de las pruebas:

- ¿Te ha resultado intuitiva la interfaz? Sí, no tiene mucha pérdida
- ¿Ha habido algún punto durante la realización de las tareas en el que no supieras como continuar? Sí, en el filtro
- ¿El diseño de la aplicación te ha resultado adecuado? Sí
- ¿Cambiarías o añadirías algo a la interfaz? Hacer el botón de los detalles de la tabla más grande
- ¿Quieres añadir alguna observación o comentario? No

Después, se les mostró la vista que no había probado (ordenador), y tras probarla un poco, respondió lo siguiente:

- ¿Sabrías realizar las tareas que has hecho antes, pero con este diseño de pantallas? Sí, probablemente mejor y más rápido
- ¿Cuál de las dos vistas te parece mas clara/intuitiva? La del ordenador

E.3. Usuario 3

Este usuario tiene el perfil de una persona joven, con unos conocimientos bajos en informática, probará la vista para ordenador.

E.3.1. Prueba 1

En las siguientes tablas se muestran los resultados obtenidos por el usuario 3 en la prueba 1:

Nº: 1	Nombre: Registrarse	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 35 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.25: Resultados del usuario 3 en la acción 1.

Nº:2	Nombre: Iniciar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 15 segundos - Clicks/toques necesarios(mínimo 3): 3 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.26: Resultados del usuario 3 en la acción 2.

Nº:3	Nombre: Cambiar idioma de la página	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.27: Resultados del usuario 3 en la acción 3.

E.3.2. Prueba 2

En las siguientes tablas se muestran los resultados obtenidos por el usuario 3 en la prueba 2:

Nº:1	Nombre: Hacer una grabación con los valores por defecto	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 35 segundos - Clicks/toques necesarios(mínimo 2): 3 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Le pone nombre a la grabación aunque no se le pida 			

Tabla E.28: Resultados del usuario 3 en la acción 1.

Nº:2	Nombre: Borrar el resultado	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.29: Resultados del usuario 3 en la acción 2.

Nº:3	Nombre: Cambiar la configuración de la grabación (Idioma, intensidad, guardar audio y nombre de la grabación)	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 15 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.30: Resultados del usuario 3 en la acción 3.

Nº:4	Nombre: Hacer una grabación con los nuevos valores	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 12 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones <ul style="list-style-type: none"> - Ninguna 			

Tabla E.31: Resultados del usuario 3 en la acción 4.

E.3.3. Prueba 3

En las siguientes tablas se muestran los resultados obtenidos por el usuario 3 en la prueba 3:

Nº:1	Nombre: Ver el historial	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 2 segundos- Clicks/toques necesarios(mínimo 1): 1- Errores: No			
Observaciones <ul style="list-style-type: none">- Ninguna			

Tabla E.32: Resultados del usuario 3 en la acción 1.

Nº:2	Nombre: Ver detalles de un audio	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 10 segundos- Clicks/toques necesarios(mínimo 1): 2- Errores: Pulsa en el audio, pero			
Observaciones <ul style="list-style-type: none">- Ninguna			

Tabla E.33: Resultados del usuario 3 en la acción 2.

Nº:3	Nombre: Filtrar segunda grabación	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none">- Tiempo dedicado a la tarea: 10 segundos- Clicks/toques necesarios(mínimo 2): 2- Errores: No			
Observaciones <ul style="list-style-type: none">- Ninguna			

Tabla E.34: Resultados del usuario 3 en la acción 3.

Nº:4	Nombre: Descargar JSON	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.35: Resultados del usuario 3 en la acción 4.

Nº:5	Nombre: Cerrar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 2 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.36: Resultados del usuario 3 en la acción 5.

E.3.4. Preguntas

En esta sección se muestran las respuestas proporcionadas por el usuario 3 a una serie de preguntas que se le realizaron tras la realización de las pruebas:

- ¿Te ha resultado intuitiva la interfaz? Sí
- ¿Ha habido algún punto durante la realización de las tareas en el que no supieras como continuar?
No, nunca
- ¿El diseño de la aplicación te ha resultado adecuado? Sí
- ¿Cambiarías o añadirías algo a la interfaz? No
- ¿Quieres añadir alguna observación o comentario? No

Después, se les mostró la vista que no había probado (ordenador), y tras probarla un poco, respondió lo siguiente:

- ¿Sabrías realizar las tareas que has hecho antes, pero con este diseño de pantallas? Probablemente sí
- ¿Cuál de las dos vistas te parece mas clara/intuitiva? La del ordenador, porque en el móvil la tabla no se ve muy bien si no lo tienes en horizontal.

E.4. Usuario 4

Este usuario tiene el perfil de una persona adulta, con unos conocimientos medios en informática, pero que no suele usar mucho el móvil, probará la vista para móvil.

E.4.1. Prueba 1

En las siguientes tablas se muestran los resultados obtenidos por el usuario 4 en la prueba 1:

Nº: 1	Nombre: Registrarse	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 60 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Duda si pulsar el botón para volver al inicio de sesión pero al final pulsa el botón correcto 			

Tabla E.37: Resultados del usuario 4 en la acción 1.

Nº:2	Nombre: Iniciar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 25 segundos - Clicks/toques necesarios(mínimo 3): 3 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.38: Resultados del usuario 4 en la acción 2.

Nº:3	Nombre: Cambiar idioma de la página	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 5 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.39: Resultados del usuario 4 en la acción 3.

E.4.2. Prueba 2

En las siguientes tablas se muestran los resultados obtenidos por el usuario 4 en la prueba 2:

Nº:1	Nombre: Hacer una grabación con los valores por defecto	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 45 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Tarda en darse cuenta de que tiene que parar la grabación 			

Tabla E.40: Resultados del usuario 4 en la acción 1.

Nº:2	Nombre: Borrar el resultado	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 1 segundo - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.41: Resultados del usuario 4 en la acción 2.

Nº:3	Nombre: Cambiar la configuración de la grabación (Idioma, intensidad, guardar audio y nombre de la grabación)	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 15 segundos - Clicks/toques necesarios(mínimo 5): 5 - Errores: No 			
Observaciones			
<ul style="list-style-type: none"> - Ninguna 			

Tabla E.42: Resultados del usuario 4 en la acción 3.

Nº:4	Nombre: Hacer una grabación con los nuevos valores	Completado: Sí	Valoración: 5
Métricas <ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 10 segundos - Clicks/toques necesarios(mínimo 2): 2 - Errores: No 			
Observaciones <ul style="list-style-type: none"> - Ninguna 			

Tabla E.43: Resultados del usuario 4 en la acción 4.

E.4.3. Prueba 3

En las siguientes tablas se muestran los resultados obtenidos por el usuario 3 en la prueba 3:

Nº:1	Nombre: Ver el historial	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.44: Resultados del usuario 4 en la acción 1.

Nº:2	Nombre: Ver detalles de un audio	Completado: Sí	Valoración: 3
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 10 segundos - Clicks/toques necesarios(mínimo 1): 2 - Errores: Pulsa en el audio, pero no en el botón + 			
Observaciones			
- Después de fallar, se da cuenta del botón y lo pulsa			

Tabla E.45: Resultados del usuario 4 en la acción 2.

Nº:3	Nombre: Filtrar segunda grabación	Completado: Sí	Valoración: 4
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 40 segundos - Clicks/toques necesarios(mínimo 2): 3 - Errores: Pulsó el botón de filtrar antes de llenar los filtros 			
Observaciones			
- Ninguna			

Tabla E.46: Resultados del usuario 4 en la acción 3.

Nº:4	Nombre: Descargar JSON	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.47: Resultados del usuario 4 en la acción 4.

Nº:5	Nombre: Cerrar sesión	Completado: Sí	Valoración: 5
Métricas			
<ul style="list-style-type: none"> - Tiempo dedicado a la tarea: 3 segundos - Clicks/toques necesarios(mínimo 1): 1 - Errores: No 			
Observaciones			
- Ninguna			

Tabla E.48: Resultados del usuario 4 en la acción 5.

E.4.4. Preguntas

En esta sección se muestran las respuestas proporcionadas por el usuario 4 a una serie de preguntas que se le realizaron tras la realización de las pruebas:

- ¿Te ha resultado intuitiva la interfaz? Sí
- ¿Ha habido algún punto durante la realización de las tareas en el que no supieras como continuar?
Cuando pulsé el botón que decía filtro, pero no hacía nada
- ¿El diseño de la aplicación te ha resultado adecuado? Sí
- ¿Cambiarías o añadirías algo a la interfaz? No
- ¿Quieres añadir alguna observación o comentario? No

Después, se les mostró la vista que no había probado (ordenador), y tras probarla un poco, respondió lo siguiente:

- ¿Sabrías realizar las tareas que has hecho antes, pero con este diseño de pantallas? Sí, mucho más fácil
- ¿Cuál de las dos vistas te parece mas clara/intuitiva? La del ordenador, en el móvil lo veo siempre todo muy pequeño y me cuesta.

Bibliografía

- [1] Ofertas TFG Escuela de Ingeniería Informática de Valladolid. Última visita: 2023-02-20. [Online]. Available: <https://www.fi.uva.es/tfg/ofertas.php>
- [2] ECA-SIMM. Última visita: 2023-07-09. [Online]. Available: <https://eca-simm.uva.es/>
- [3] Radboud University - CLST. Última visita: 2023-06-29. [Online]. Available: <https://www.ru.nl/clst/>
- [4] Audioserver. Última visita: 2023-06-30. [Online]. Available: <https://github.com/cristiantg/audioserver>
- [5] M. Cotterell and B. Hughes, *Software Project Management 5Th Edition*. McGraw-Hill Education, 2009, ch. 2.
- [6] Esquema desarrollo iterativo e incremental. Última visita: 2023-06-29. [Online]. Available: <http://aprendiendocositasdelssoftware.blogspot.com/p/desarrollo-iterativo-y-creciente.html>
- [7] Angular. Última visita: 2023-06-30. [Online]. Available: <https://angular.io/docs>
- [8] Astah. Última visita: 2023-07-01. [Online]. Available: <https://astah.net/>
- [9] Babeledit. Última visita: 2023-07-01. [Online]. Available: <https://www.codeandweb.com/babeledit>
- [10] Balsamiq wireframes. Última visita: 2023-07-01. [Online]. Available: <https://balsamiq.com/wireframes/>
- [11] Docker. Última visita: 2023-07-01. [Online]. Available: <https://docs.docker.com/get-started/overview/>
- [12] Git for windows. Última visita: 2023-07-01. [Online]. Available: <https://gitforwindows.org/>
- [13] Gitlab. Última visita: 2023-07-01. [Online]. Available: <https://about.gitlab.com/>
- [14] Google chrome. Última visita: 2023-07-01. [Online]. Available: https://www.google.com/intl/es_es/chrome/
- [15] Jitsi meet. Última visita: 2023-07-01. [Online]. Available: <https://meet.jit.si/>
- [16] Microsoft excel. Última visita: 2023-07-01. [Online]. Available: <https://www.microsoft.com/es-es/microsoft-365/excel>
- [17] Nginx. Última visita: 2023-07-01. [Online]. Available: <https://www.nginx.com/resources/glossary/nginx/>

- [18] Node.js. Última visita: 2023-07-01. [Online]. Available: <https://nodejs.org/es/about>
- [19] Npm. Última visita: 2023-07-01. [Online]. Available: <https://www.npmjs.com/about>
- [20] Outlook. Última visita: 2023-07-01. [Online]. Available: <https://www.microsoft.com/es-es/microsoft-365/outlook/email-and-calendar-software-microsoft-outlook>
- [21] Overleaf. Última visita: 2023-07-01. [Online]. Available: es.overleaf.com
- [22] Swagger. Última visita: 2023-07-01. [Online]. Available: <https://swagger.io/about/>
- [23] Visual studio code. Última visita: 2023-07-01. [Online]. Available: <https://code.visualstudio.com/docs/editor/whyvscode>
- [24] Windows notepad. Última visita: 2023-07-01. [Online]. Available: <https://apps.microsoft.com/store/detail/windows-notepad/9MSMLRH6LZF3?hl=es-es&gl=es>
- [25] Bootstrap. Última visita: 2023-07-01. [Online]. Available: <https://getbootstrap.com/>
- [26] Ngx-translate. Última visita: 2023-05-30. [Online]. Available: <https://github.com/ngx-translate/core>
- [27] Recordrtc. Última visita: 2023-06-20. [Online]. Available: <https://recordrtc.org/>
- [28] Rxjs. Última visita: 2023-04-17. [Online]. Available: <https://rxjs.dev/>
- [29] Voice activity detection. Última visita: 2023-06-20. [Online]. Available: <https://github.com/Jam3/voice-activity-detection>
- [30] Sueldo medio de un programado junior en españa, según glassdoor. Última visita: 2023-07-02. [Online]. Available: https://www.glassdoor.es/Sueldos/programador-junior-sueldo-SRCH_KO0,18.htm
- [31] Reglamento general de protección de datos. Última visita: 2023-07-09. [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection_es
- [32] Mario Panadero Palazuelos. Propuesta de matchmaking para aplicaciones educativas multijugador online. 2022. Trabajo Fin de Grado, Departamento de Informática, Universidad de Valladolid. Última visita: 2020-07-08. [Online]. Available: <https://uvadoc.uva.es/handle/10324/57343>
- [33] Juan García Diéguez. Migración de aplicación Android con Google Play Games a API REST. 2021. Trabajo Fin de Grado, Departamento de Informática, Universidad de Valladolid. Última visita: 2020-07-08. [Online]. Available: <https://uvadoc.uva.es/handle/10324/50373>
- [34] Antonio Gamazo Ferrero. Desarrollo de una aplicación educativa multiplataforma para el entrenamiento de la pronunciación de idiomas. 2021. Trabajo Fin de Grado, Departamento de Informática, Universidad de Valladolid. Última visita: 2020-07-08. [Online]. Available: <https://uvadoc.uva.es/handle/10324/47131>
- [35] C. Tejedor-García, V. Cardeñoso-Payo, E. Cámara-Arenas, C. González-Ferreras, and D. Escudero-Mancebo, "Playing around minimal pairs to improve pronunciation training," in *Proc. IFCASL*, ser. Feedback in Pronunciation Training Workshop, Saarland, Germany, Nov. 5–6, 2015.

- [36] C. Tejedor-García, V. Cardeñoso-Payo, M. J. Machuca, D. Escudero-Mancebo, A. Ríos, and T. Kimura, “Improving Pronunciation of Spanish as a Foreign Language for L1 Japanese Speakers with Japañol CAPT Tool,” in *Proc. IberSPEECH*, Barcelona, Spain, Nov. 21–23, 2018, pp. 97–101.
- [37] C. Tejedor-García, D. Escudero-Mancebo, V. Cardeñoso-Payo, and C. González-Ferreras, “Using challenges to enhance a learning game for pronunciation training of english as a second language,” *IEEE Access*, vol. 8, pp. 74 250–74 266, 2020.
- [38] Google stt. Última visita: 2023-07-07. [Online]. Available: <https://cloud.google.com/speech-to-text?hl=es>
- [39] Precios google stt. Última visita: 2023-07-07. [Online]. Available: <https://cloud.google.com/speech-to-text/pricing?hl=es>
- [40] Kaldi. Última visita: 2023-07-09. [Online]. Available: <https://kaldi-asr.org/>
- [41] Lista de modelos de kaldi. Última visita: 2023-07-09. [Online]. Available: <https://kaldi-asr.org/models.html>
- [42] Whisper. Última visita: 2023-07-10. [Online]. Available: <https://openai.com/research/whisper>
- [43] wav2vec2.0. Última visita: 2023-07-10. [Online]. Available: <https://huggingface.co/blog/wav2vec2-with-ngram>
- [44] Modelo multilinguaje wav2vec2.0. Última visita: 2023-07-10. [Online]. Available: <https://huggingface.co/voidful/wav2vec2-xlsr-multilingual-56>
- [45] Swagger editor. Última visita: 2023-07-01. [Online]. Available: <https://swagger.io/tools/swagger-editor/>
- [46] Comparación de recordrt con otras librerías similares. Última visita: 2023-06-19. [Online]. Available: <https://npmrends.com/opus-recorder-vs-recorderjs-vs-recordrtc>
- [47] recordrtc en synkadvisor. Última visita: 2023-06-19. [Online]. Available: <https://snyk.io/advisor/npm-package/recordrtc>
- [48] Comparación de voice activity detection con otras librerías similares. Última visita: 2023-06-19. [Online]. Available: <https://npmrends.com/node-vad-vs-vad-vs-voice-activity-detection>
- [49] Documentación de analysernode. Última visita: 2023-07-03. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/AnalyserNode>
- [50] Documentación de scriptprocessornode. Última visita: 2023-07-03. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/ScriptProcessorNode>
- [51] Documentación de smoothingtimeconstant. Última visita: 2023-07-03. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/AnalyserNode/smoothingTimeConstant>
- [52] Patrón módulo en javascript. Última visita: 2023-07-09. [Online]. Available: <https://javascriptpatterns.vercel.app/patterns/design-patterns/module-pattern>

- [53] Patrón módulo en angular. Última visita: 2023-07-09. [Online]. Available: <https://angular.io/guide/singleton-services>
- [54] Patrón observador en angular. Última visita: 2023-07-09. [Online]. Available: <https://codigofacilito.com/articulos/patron-observable>

