

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE APLICACIÓN MÓVIL DE STREAMING DE AUDIOLIBROS BASADO EN ARQUITECTURA DE MICROSERVICIOS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

CARLOS FELIPE ANDRADE VALLADARES

carlos.andrade03@epn.edu.ec

JOSSELYN VIOLETA BRUSIL LLANOS

josselyn.brusil@epn.edu.ec

DIRECTOR: Msc. Victor Vicente Velepucha Bonett

victor.velepucha@epn.edu.ec

CODIRECTOR: PhD. Pamela Catherine Flores Naranjo

pamela.flores@epn.edu.ec

Quito, 03 de enero del 2023

DECLARACIÓN

Nosotros, Carlos Felipe Andrade Valladares y Josselyn Violeta Brusil Llanos, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Andrade Valladares Carlos Felipe



Brusil Llanos Josselyn Violeta

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por los estudiantes Andrade Valladares Carlos Felipe y Brusil Llanos Josselyn Violeta, bajo nuestra supervisión.



Msc. Victor Velepucha
DIRECTOR DE PROYECTO

PhD. Pamela Flores
CO- DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco a toda mi familia, por sus consejos y por el apoyo que fueron pilares fundamentales en mi proceso universitario.

Agradezco al Ing. Victor Velepucha y la Dr. Pamela Flores, por todo su tiempo, experiencia y seguimiento brindado a lo largo de este proyecto, su sabiduría me dio un nuevo enfoque en la realización de mis proyectos.

Agradezco a mi compañera de Tesis, Violeta Brusil, por su confianza y todo el esfuerzo invertido en este proyecto, compartir las clases y el proyecto de titulación ha sido una experiencia gratificante.

Agradezco a mis compañeros y amigos que hice a lo largo de mi estancia en la Facultad de Sistemas con los que compartí momentos de alegría y momentos complicados que supimos llevar adelante.

Agradezco a la Escuela Politécnica Nacional, por brindarme educación de calidad, que me servirá en mi futura vida como profesional y por enseñarme valores que aplicare en mi vida personal.

Felipe Andrade

DEDICATORIA

Quiero dedicar este trabajo de titulación a:

A mis padres, Eliana y Carlos por todo el amor, sacrificio, esfuerzo y confianza que invirtieron en mí y en mi educación.

A mi hermana, Pamela por su preocupación e interés.

A mis tías, Bertha, Dalia, Amparo y Rocío, a mi primo Alex por todo el apoyo, interés, y consejos que los aproveche sabiamente.

Felipe Andrade

AGRADECIMIENTO

Agradezco a Dios y a mis padres Silvia Llanos y Marcelo Brusil por su infinito amor y por permitirme tener esta gran experiencia. Quisiera agradecer a mi familia Brusil Llanos por ser mi respaldo desde el inicio por su esfuerzo y perseverancia. Igualmente, a Daniel por es mi apoyo incondicional, a mis amigos y compañeros por ser parte de una de las mejores épocas de mi vida.

También quisiera agradecer a mis tutores, Msc. Victor Velepucha y la PhD. Pamela Flores por su apoyo durante el desarrollo de este proyecto gracias por su tiempo y dedicación brindados; a mi compañero Felipe por su esfuerzo y dedicación.

Sin más, quisiera agradecer a la Escuela Politécnica Nacional por permitirme crecer en sus aulas y ser mi alma mater. Finalmente, a todos mis maestros que fueron parte de este gran recorrido.

Violeta Brusil

DEDICATORIA

A mis padres, mis hermanos, mis sobrinos por creer siempre en mi en ser ese apoyo en los momentos difíciles.

A Daniel por estar durante esta culminación de esta experiencia.

Violeta Brusil

ÍNDICE DE CONTENIDO

DECLARACIÓN	ii
CERTIFICACIÓN	iii
AGRADECIMIENTO.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
DEDICATORIA.....	vii
ÍNDICE DE CONTENIDO.....	viii
LISTA DE FIGURAS	xii
LISTA DE TABLAS.....	xix
RESUMEN	xxi
ABSTRACT	xxii
CAPÍTULO 1 – INTRODUCCIÓN	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivo General.....	2
1.2.2. Objetivos Específicos	2
CAPÍTULO 2 – MARCO TEÓRICO.....	3
2.1 Arquitectura monolítica	3
2.2 Arquitectura de microservicios	4
2.2.1 Principios de arquitectura de microservicios	5
2.2.2 Características de los microservicios	6
2.2.3 Ventajas de microservicios.....	6
2.2.4 Desventajas de microservicios	7
2.2.5 Arquitectura de comunicación directa cliente – microservicio	7

2.3	Patrón modelo - vista - modelo vista.....	8
2.4	Aplicación móvil	9
2.5	Aplicación web	9
2.6	JSON.....	10
2.7	API	10
2.8	Backend	11
2.9	Frontend.....	11
2.10	T-Shirt Sizing	11
2.11	Mockups.....	11
2.12	Base de datos SQL.....	12
2.13	Base de datos NoSQL	12
2.14	Scrum.....	13
2.14.1	Roles de Scrum.....	13
2.14.2	Elementos de Scrum	14
2.14.3	Reuniones Scrum.....	14
2.15	Pruebas Funcionales	15
2.15.1	Pruebas de aceptación.....	15
2.16	Pruebas no funcionales	15
2.16.1	Pruebas de carga	16
2.16.2	Pruebas de rendimiento	16
2.17	Herramientas utilizadas	16
2.18	Hábitos de lectura en la sociedad.....	25
2.19	Impacto de la tecnología en la sociedad	26
2.20	Stakeholders.....	26
CAPÍTULO 3 – DESARROLLO DE LA APLICACIÓN.....		28

3.1 FASE EXPLORATORIA	28
3.1.1. Alcance del proyecto.....	28
3.1.2. Planificación del proyecto	28
3.2. FASE DE INICIALIZACIÓN	28
3.2.1. Historias de Usuario.....	28
3.2.2. Product Backlog.....	29
3.2.3. Planificación de Sprints.....	32
3.2.4. Ejecución de los Sprints.....	34
3.2.4.1. Sprint 0.....	34
3.2.4.2. Sprint 1	59
3.2.4.3. Sprint 2.....	65
3.2.4.4. Sprint 3.....	74
3.2.4.5. Sprint 4.....	79
3.2.4.6. Sprint 5.....	84
3.2.4.7. Sprint 6.....	92
CAPÍTULO 4 – ANÁLISIS DE RESULTADOS.....	101
4.1. Ejecución de pruebas del Sprint 1	101
4.2. Ejecución de pruebas del Sprint 2	123
4.3. Ejecución de pruebas del Sprint 3	134
4.4. Ejecución de pruebas del Sprint 4	147
4.5. Ejecución de pruebas del Sprint 5	156
4.6. Ejecución de pruebas del Sprint 6	178
CAPÍTULO 5 – CONCLUSIONES Y RECOMENDACIONES	206
5.1. Conclusiones	206
5.2. Recomendaciones.....	207

REFERENCIAS BIBLIOGRÁFICAS209

LISTA DE FIGURAS

Figura 1 - Arquitectura Monolítica [8]	3
Figura 2 - Arquitectura Monolítica y Arquitectura con Microservicios [8]	4
Figura 3 - Arquitectura de Comunicación Cliente – Microservicio [12].....	8
Figura 4 - Patrón MVVM [14].....	9
Figura 5 - Instrucción SQL [22]	12
Figura 6 - Proceso SCRUM [24].....	13
Figura 7 - Funcionamiento Docker [45]	23
Figura 8 - Arquitectura de microservicios del proyecto	36
Figura 9 - Repositorio Git del microservicio AudioBookCatalogMicroservice	37
Figura 10 - Repositorio Git de UserPlayListMicroservice.....	38
Figura 11 - Repositorio Git del aplicativo web.....	38
Figura 12 - Repositorio Git del aplicativo móvil	39
Figura 13 - Mockup de la pantalla de inicio de sesión del aplicativo web	40
Figura 14 - Mockup de la pantalla de registro de usuarios al aplicativo web	40
Figura 15 - Mockup de la pantalla inicial para usuario autor.....	41
Figura 16 - Mockup de la pantalla inicial para usuario administrador	42
Figura 17 - Mockup de la pantalla para añadir audiolibros	43
Figura 18 - Mockup de la pantalla para editar audiolibros	44
Figura 19 - Mockup de la pantalla para editar información del usuario autor	45
Figura 20 - Mockup de la pantalla de inicio de sesión de la aplicación móvil	46
Figura 21 - Mockup de la pantalla de registro de usuarios a la aplicación móvil	47
Figura 22 - Mockup de la pantalla inicial de la aplicación móvil.....	48
Figura 23 - Mockup de la pantalla para buscar audiolibros.....	49
Figura 24 - Mockup de la pantalla para ver las listas de reproducción	50
Figura 25 - Mockup de la pantalla para crear una lista de reproducción	51
Figura 26 - Mockup de la pantalla para ver información del audiolibro.....	52
Figura 27 - Mockup de la pantalla del reproductor de audiolibros	53
Figura 28 - Mockup de la pantalla del menú de opciones del audiolibro	54

Figura 29 - Mockup de la pantalla para añadir un audiolibro a una lista de reproducción	55
Figura 30 - Mockup de la pantalla para reseñar un audiolibro	56
Figura 31 - Mockup de la pantalla de configuración.....	57
Figura 32 - Mockup de la pantalla para editar información de usuario desde la aplicación móvil.....	58
Figura 33 - Base catalog-microservice contenida en el Docker.....	61
Figura 34 - Atributos de la entidad AudioBook.....	62
Figura 35 - Atributos de la entidad Review	62
Figura 36 - Archivo Dockerfile del microservicio “AudioBookCatalogMicroservice” ...	63
Figura 37 - Comando para construir la imagen “audiobook-catalog-microservice”	64
Figura 38 - Microservicio AudioBookCatalogMicroservice contenido en el Docker Desktop	64
Figura 39 - Comando para la creación del proyecto en React [43].....	67
Figura 40 - Creación del proyecto en la consola de Firebase [40].....	68
Figura 41 - Integración de Firebase con React [27].....	69
Figura 42 - Base userPlayList-microservice contenida en el Docker[45].....	81
Figura 43 - Atributos de la entidad PlayList.....	81
Figura 44 - Atributos de la entidad IdAudioBook.....	82
Figura 45 – Archivo Dockerfile del microservicio “UserPlayListMicroservice”	83
Figura 46 – Comando para construir la imagen “audiobook-catalog-microservice” ...	83
Figura 47 - Microservicio UserPlayListMicroservice contenido en el Docker Desktop	83
Figura 48 - Creación del proyecto en XCode [27].....	86
Figura 49 – Configuración del proyecto AudityAppMobile	87
Figura 50 – Archivo User.....	88
Figura 51 - Configuración del hilo para la prueba de la petición Add AudioBook	113
Figura 52 - Configuración de la respuesta Add AudioBook	113
Figura 53 - HTTP Header Manager para la petición Add AudioBook	114
Figura 54 – Informe resumido de la petición Add AudioBook	114

Figura 55 - Árbol de resultados de la prueba de carga de la petición Add AudioBook	115
Figura 56 – Gráfico de los resultados de la prueba para la petición Add AudioBook	116
Figura 57 - Configuración del hilo para la prueba en JMeter de la petición Get All AudioBooks	117
Figura 58 - Configuración de la respuesta Get All AudioBooks	117
Figura 59 - Summary Report de la petición Get All AudioBooks.....	118
Figura 60 - Árbol de resultados de la prueba para la petición Get All AudioBooks..	118
Figura 61 - Gráfico de resultado de la prueba para la petición Get All AudioBooks	119
Figura 62 - Configuración del hilo para la prueba de la petición Search AudioBooks	120
Figura 63 - Configuración de la respuesta Search AudioBooks Per Author	120
Figura 64 - Configuración de la respuesta Search AudioBoks Per Title	121
Figura 65 - Configuración de la respuesta Search AudioBooks Per Gender	121
Figura 66 - Summary Report de la petición Search AudioBooks	122
Figura 67 – Árbol de resultados de la prueba para la petición Search AudioBooks	122
Figura 68 - Gráfico de resultado de la prueba para la petición Search AudioBooks	123
Figura 69 – Registro de un usuario autor fallido desde el aplicativo web	124
Figura 70 – Selección de una imagen para foto de perfil desde el aplicativo web...	125
Figura 71 – Registro de usuario autor correcto desde el aplicativo web.....	125
Figura 72 - Pantalla de inicio de sesión del aplicativo web	126
Figura 73 - Pantalla principal del aplicativo web	126
Figura 74 - Inicio de sesión con Google desde el aplicativo web.....	127
Figura 75 - Inicio de sesión fallido desde el aplicativo web.....	128
Figura 76 - Inicio de sesión exitoso desde el aplicativo web.....	128
Figura 77 – Actualización de la información personal del usuario autor desde el aplicativo web.....	129
Figura 78 – Pantalla principal cuando el usuario haya editado su información personal	129

Figura 79– Pantalla de inicio de sesión del usuario administrador desde el aplicativo web.....	130
Figura 80 – Pantalla principal del usuario administrador	131
Figura 81 – Pantalla para habilitar/deshabilitar la cuenta de un usuario.....	131
Figura 82 – Configuración del hilo para la prueba de carga de la página de inicio del aplicativo web.....	132
Figura 83 – Árbol de resultados de las peticiones que el usuario realiza al iniciar sesión en el aplicativo web	133
Figura 84 – Informe resumido del plan de prueba a la página de inicio de sesión ..	133
Figura 85 – Gráfico del resultado del plan de prueba de la página de inicio de sesión	134
Figura 86 – Agregación de un nuevo audiolibro.....	136
Figura 87 – Pantalla principal del usuario autor al agregar un audiolibro desde el aplicativo web.....	137
Figura 88 – Actualización de la información de un audiolibro desde el aplicativo web	138
Figura 89 – Pantalla principal del usuario autor después de editar la información de un audiolibro	138
Figura 90– Eliminación de un audiolibro desde el aplicativo web	139
Figura 91 – Grupo de hilos para la prueba de carga a la página de añadir audiolibro	140
Figura 92 – Árbol de resultados de la prueba a la página de añadir audiolibro	140
Figura 93 – Informe resumido de la prueba a la página de añadir audiolibro	141
Figura 94 – Gráfico de la prueba de carga a la página de añadir audiolibro	141
Figura 95 – Grupo de hilos de la prueba de carga a la página de editar audiolibro.	142
Figura 96 – Árbol de resultados de la prueba a la página de editar audiolibro	143
Figura 97 – Árbol de resultados de la prueba a la página de editar audiolibro	143
Figura 98 – Gráfico de la prueba de carga a la página de editar audiolibro	144
Figura 99 – Grupo de hilos de la prueba a la página de eliminar audiolibro	145
Figura 100 – Árbol de resultados de la prueba a la página de eliminar audiolibro ..	145
Figura 101 – Árbol de resultados de la prueba a la página de eliminar audiolibro ..	146

Figura 102 – Gráfico de la prueba de carga a la página de eliminar audiolibro.....	146
Figura 103 - Configuración del hilo para la prueba en JMeter de la petición Create Playlist.....	151
Figura 104 - Configuración de la respuesta Create Playlist.....	151
Figura 105 - HTTP Header Manager para la petición Create Playlist.....	151
Figura 106 - Summary Report de la petición Create Playlist	152
Figura 107 - Árbol de resultados de la prueba de la petición Create Playlist	152
Figura 108 – Gráfico de los resultados de la prueba para la petición Create Playlist	153
Figura 109 - Configuración del hilo para la prueba de la petición Add AudioBook To Playlist.....	154
Figura 110 - Configuración de la respuesta Add AudioBook To Playlist.....	154
Figura 111 - Summary Report de la petición Add AudioBook To Playlist.....	154
Figura 112 - Árbol de resultados de la prueba de la petición Add AudioBook To Playlist	155
Figura 113 – Gráfico de los resultados de la prueba para la petición Add AudioBook To Playlist.....	156
Figura 114 – Registro de usuario final inválido desde la aplicación móvil	157
Figura 115– Registro de usuario final válido desde la aplicación móvil	158
Figura 116 – Pantalla principal de la aplicación móvil.....	159
Figura 117 - Inicio de sesión con Google desde la aplicación móvil.....	160
Figura 118 – Selección de cuenta de Google desde la aplicación móvil para el registro	161
Figura 119 – Pantalla principal de la aplicación móvil ingresando con la cuenta de Google.....	162
Figura 120 – Inicio de sesión del usuario final cuando no ingresa los campos solicitados	163
Figura 121 – Pantalla de inicio de sesión cuando el usuario final ingresa los campos solicitados.....	164
Figura 122 – Inicio de sesión exitoso desde la aplicación móvil	165
Figura 123 – Pantalla de configuración de la aplicación móvil.....	166

Figura 124 – Mensaje informativo cuando el usuario final edita su información personal	167
Figura 125 – Pantalla de configuración con la información actualizada	168
Figura 126 – Pantalla principal de la herramienta Apptim [40].....	169
Figura 127 – Selección del dispositivo en donde se realizará la prueba	170
Figura 128 – Selección de la aplicación a testear.....	171
Figura 129 – Configuración del tiempo de duración de la prueba a la aplicación móvil	172
Figura 130 – Sesión exploratoria de la herramienta Apptim	173
Figura 131 – Datos relevantes capturados durante la prueba	174
Figura 132 – Video de la ejecución de la prueba	174
Figura 133 – Tabla de resumen de las métricas evaluadas por Apptim	175
Figura 134 – Información de la sesión	175
Figura 135 – Gráfico del uso de la CPU y el recuento de hilos.....	176
Figura 136 – Memoria utilizada por la aplicación móvil.....	176
Figura 137 – Gráfico de la red cuando envía y recibe bytes.....	177
Figura 138 – Gráfico del rendimiento de la renderización FPS.....	177
Figura 139 – Gráfico del almacenamiento disponible	178
Figura 140 – Pantalla principal de la aplicación móvil eligiendo el icono de buscar	179
Figura 141 – Búsqueda por nombre de autor desde la aplicación móvil	180
Figura 142 – Búsqueda por título de audiolibro desde la aplicación móvil	182
Figura 143 – Búsqueda por género del audiolibro desde la aplicación móvil	184
Figura 144 – Pantalla principal de la aplicación móvil seleccionando un audiolibro	185
Figura 145 – Pantalla de información de un audiolibro	186
Figura 146 – Pantalla de reproducción de un audiolibro.....	187
Figura 147 – Pantalla principal seleccionando el icono de “Mi biblioteca”	188
Figura 148 – Pantalla de “Mi librería” seleccionando el icono para crear una lista de reproducción.....	189
Figura 149 – Creación de una lista de reproducción.....	190
Figura 150 – Mensaje cuando el usuario excede el número de caracteres permitidos al crear una lista de reproducción	191

Figura 151 – Pantalla del reproductor de un audiolibro seleccionando el icono de los tres puntos.....	192
Figura 152 – Selección de la opción Añadir a lista de reproducción.....	193
Figura 153 – Agregación de un audiolibro a una lista de reproducción	194
Figura 154 – Mensaje de alerta cuando el usuario final desee agregar el mismo libro a la misma lista de reproducción	195
Figura 155 – Selección de la opción Reseña.....	196
Figura 156 – Reseña a un audiolibro	197
Figura 157 – Mensaje de advertencia cuando el comentario exceda su número de caracteres permitido.....	198
Figura 158 – Sesión exploratoria de la herramienta Apptim de la segunda prueba	199
Figura 159 – Datos relevantes capturados durante la segunda prueba	200
Figura 160 – Video de la ejecución de la segunda prueba	201
Figura 161 – Tabla de resumen de las métricas evaluadas por Apptim durante la segunda prueba	201
Figura 162 – Información de la sesión de la segunda prueba	202
Figura 163– Gráfico del uso de la CPU y el recuento de hilos de la segunda prueba	203
Figura 164 – Memoria utilizada por la aplicación móvil de la segunda prueba.....	203
Figura 165 – Gráfico de la red cuando envía y recibe bytes de la segunda prueba	204
Figura 166 – Gráfico del rendimiento de la renderización FPS de la segunda prueba	205
Figura 167 – Gráfico del almacenamiento disponible de la segunda prueba	205

LISTA DE TABLAS

Tabla 1 - Partes Interesadas	27
Tabla 2 - Roles de Scrum.....	28
Tabla 3 – Estimación aproximada de tiempo con la técnica T-Shirt Sizing.....	29
Tabla 4 – Product Backlog	32
Tabla 5 - Planificación de Sprints.....	33
Tabla 6 – Sprint 1 Planning	61
Tabla 7 – Sprint 1 Review	65
Tabla 8 – Sprint 2 Planning	67
Tabla 9 - Criterios de aceptación de la historia de usuario HU01 – 01	71
Tabla 10 – Criterios de aceptación de la historia de usuario HU01 – 02	71
Tabla 11 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 01 ...	72
Tabla 12 – Criterios de aceptación cumplidos de la historia de usuario HU02 – 02 ..	72
Tabla 13 – Criterios de aceptación cumplidos de la historia de usuario HU02 – 05 ..	73
Tabla 14 – Criterios de aceptación cumplidos de la historia de usuario HU07 – 01 ..	74
Tabla 15 – Sprint 3 Planning	76
Tabla 16 – Criterios de aceptación de la historia de usuario HU03 - 01	78
Tabla 17 – Criterios de aceptación de la historia de usuario HU03 - 02	78
Tabla 18 – Criterios de aceptación de la historia de usuario HU03 - 03	79
Tabla 19 – Sprint 4 Planning	80
Tabla 20 – Sprint 4 Review	84
Tabla 21 – Sprint 5 Planning	86
Tabla 22 – Criterios de aceptación de la historia de usuario HU01 - 03	89
Tabla 23 – Criterios de aceptación de la historia de usuario HU01 - 04	90
Tabla 24 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 03 ...	90
Tabla 25 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 04 ...	91
Tabla 26 – Sprint 6 Planning	94
Tabla 27 – Criterios de aceptación de la historia de usuario HU04 - 01	95
Tabla 28 – Criterios de aceptación de la historia de usuario HU04 - 02	96
Tabla 29 – Criterios de aceptación cumplidos de la historia de usuario HU04 - 03 ...	97
Tabla 30 – Criterios de aceptación cumplidos de la historia de usuario HU05 - 01 ...	97

Tabla 31 – Criterios de aceptación cumplidos de la historia de usuario HU06 - 01 ...	98
Tabla 32 – Criterios de aceptación cumplidos de la historia de usuario HU06 - 02 ...	99
Tabla 33 – Criterios de aceptación cumplidos de la historia de usuario HU05 – 02 ..	99
Tabla 34 - Petición POST para añadir un audiolibro	102
Tabla 35 - Petición PUT para actualizar un audiolibro	104
Tabla 36 - Petición GET para obtener todos los audiolibros agregados.....	105
Tabla 37 - Petición GET para buscar audiolibros por nombre de autor	106
Tabla 38 - Petición GET para buscar audiolibros por título de audiolibro	107
Tabla 39 - Petición GET para buscar audiolibros por género	108
Tabla 40 - Petición GET para añadir una reseña a un audiolibro	109
Tabla 41 - Petición GET para buscar un audiolibro por su id.....	110
Tabla 42 - Petición DELETE para eliminar un audiolibro	111
Tabla 43 - Petición GET para buscar un audiolibro por su id con respuesta null	112
Tabla 44 - Petición POST para crear una lista de reproducción	148
Tabla 45 - Petición POST para agregar un audiolibro a una lista de reproducción .	149
Tabla 46 - Petición GET para obtener las listas de reproducción	150

RESUMEN

El objetivo de este trabajo de investigación es diseñar y desarrollar una aplicación que se integre al reciente avance tecnológico y a los hábitos de lectura de la sociedad, es por lo que el tema principal de esta aplicación es de streaming de audiolibros construida bajo la lógica de una arquitectura de microservicios.

Para el desarrollo de esta aplicación hemos utilizado como base el marco de trabajo Scrum, dividiendo el desarrollo en sprints, usando Git y GitHub como controlador de versiones del proyecto, la parte de frontend está conformada por una aplicación móvil esta desarrollada con sistema operativo iOS, para el usuario final y una aplicación web desarrollada en React para los usuarios autores y administrador.

Como backend tenemos la arquitectura de microservicios desarrollada con el framework SpringBoot con el lenguaje de programación Java donde cada microservicio tiene su propia base de datos levantada en PostgreSQL alojados en Docker. Además, se usó la plataforma Firebase para la autenticación de usuarios y el almacenamiento de imágenes y audios que tendrá la aplicación móvil y web.

Para terminar con un correcto desarrollo de la aplicación, se realizó la etapa de pruebas funcionales y no funcionales que se realizó con las aplicaciones de Jmeter, BlazeMeter para evaluar la parte de backend y Apptim para la parte de pruebas de la aplicación móvil.

Palabras Clave: Aplicación, Docker, IOS, Microservicios, Móvil, React, Web.

ABSTRACT

The objective of this research work is to design and develop an application that integrates the recent technological advancement and reading habits of society, that is why the main theme of this application is streaming audiobooks built under the logic of a microservices architecture.

For the development of this application, we have used as a basis the Scrum framework, dividing the development in sprints, using Git and GitHub as version controller of the project, the frontend part consists of a mobile application is developed with iOS operating system, for the reader users and a web application developed in React for the part of the administrator user.

As backend we have the microservices architecture developed with the SpringBoot framework that works with a model view controller and the Java programming language and each microservice has a database raised in MySQL hosted in Docker.

To finish with a correct development of the application, the stage of functional and non-functional testing was performed with Jmeter, BlazeMeter applications to evaluate the backend and Apptim for the testing part of the mobile application.

Keywords: Application, Docker, IOS, Microservices, Mobile, React, Web.

CAPÍTULO 1 – INTRODUCCIÓN

1.1. Planteamiento del problema

Leer un libro es un hábito que implica tiempo y dedicación y, en ocasiones, las actividades diarias nos impiden desarrollar este hábito [1], la adquisición de conocimientos a través de los libros tiene un valor único, debido a que mantiene la mente activa, estimula diversos procesos cognitivos, estimula la memoria y la imaginación e incrementa el vocabulario. Al mantener el cerebro activo, ayuda a prevenir el Alzheimer y otras enfermedades degenerativas; desarrolla el pensamiento analítico potenciando la concentración [2].

El uso de los libros electrónicos contiene un cierto grado de distracción al momento de comprender textos en profundidad, es decir, la comprensión de la lectura suele ser menor si se lee en un ordenador, tableta o un dispositivo móvil. Por otro lado, la necesidad de utilizar el desplazamiento vertical del texto aumenta el efecto negativo sobre el interés en leer un libro [3].

Escuchar un audiolibro es una alternativa a leer un libro electrónico de la manera tradicional, debido a múltiples factores como, por ejemplo, el tiempo empleado en leer un libro electrónico es mayor al tiempo empleado en escuchar un audiolibro, lo que implica un incremento de libros leídos en un periodo de tiempo. Otro factor a su favor es la facilidad de disfrutar de cualquier libro al mismo tiempo que se realiza otras actividades repetitivas y que demandan un mínimo nivel de concentración [4]. Los audiolibros están disponibles como archivos de audio digital se los puede utilizar en dispositivos como reproductores de mp3, tablets, smartphones, laptops o computadoras de escritorio. Las principales plataformas a donde se tiene acceso a los audiolibros son: Audible, iBooks y Project Gutenberg.

Actualmente las arquitecturas de software que soportan audiolibros están diseñadas a partir de arquitecturas tradicionales, como la arquitectura monolítica que implica una barrera en la actualización o cambio en el código lo que requiere un nuevo lanzamiento de la aplicación por completo [5], al momento no se han

encontrado aplicaciones de streaming de audiolibros basadas en una arquitectura de microservicios, a pesar de ser una arquitectura basada que permite un mejor desarrollo, distribución y rendimiento del producto ya que cualquier cambio se lo puede realizar en un microservicio específico.

Conforme a estos antecedentes se propone realizar el desarrollo de una aplicación móvil de streaming de audiolibros, que servirá a los estudiantes de la facultad de Ingeniería en Sistemas Informáticos y de Computación como una alternativa para introducirse en la literatura, la aplicación estará basada en una arquitectura de microservicios con la finalidad de obtener un software de calidad con mayor rapidez desarrollando múltiples microservicios de forma simultánea [6].

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una aplicación móvil de streaming de audiolibros y un aplicativo web basados en una arquitectura de microservicios utilizando el marco de trabajo SCRUM.

1.2.2. Objetivos Específicos

- Identificar las historias de usuarios para la aplicación móvil de streaming de audiolibros y del aplicativo web.
- Diseñar la arquitectura de los microservicios de la aplicación móvil de streaming de audiolibros y del aplicativo web.
- Implementar el aplicativo web y la aplicación móvil de streaming de audiolibros que permitan incrementar el interés sobre los libros.
- Realizar pruebas funcionales de la aplicación móvil y del aplicativo web con el fin de obtener métricas de usabilidad.
- Realizar pruebas no funcionales de la aplicación móvil y del aplicativo web con el fin de obtener métricas de rendimiento.

CAPÍTULO 2 – MARCO TEÓRICO

2.1 Arquitectura monolítica

Una aplicación o un sistema construido con una arquitectura monolítica, describe una funcionalidad en donde todo el sistema es un solo programa ejecutable que contiene acceso de datos, interfaces de usuario, lógica de negocios, bases de datos, etc en una sola capa como podemos ver en la **Figura 1**[7].

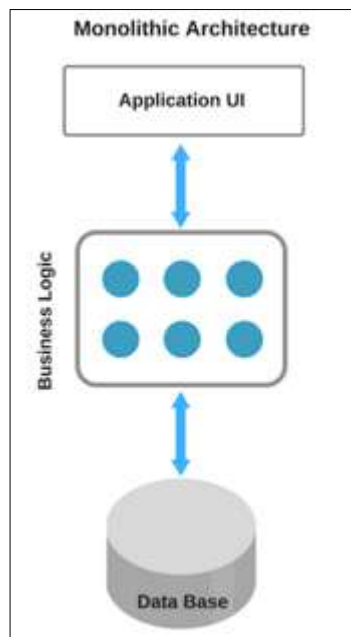


Figura 1 - Arquitectura Monolítica [8]

Esta arquitectura implica que al momento de implementar cambios en los módulos de la aplicación se debería desplegar nuevamente la aplicación completa, y el acceso a los datos se complica debido que, al tener todos los datos que utiliza la lógica de la aplicación están alojados en una sola base de datos [7].

Las aplicaciones elaboradas con esta arquitectura monolítica se ejecutan en un solo servidor local y están desarrolladas con un solo lenguaje de programación y una sola tecnología, lo que dificulta implementar una escalabilidad de la aplicación [7].

2.2 Arquitectura de microservicios

Una arquitectura de microservicios es un tipo de arquitectura que desarrolla una aplicación como un conjunto de servicios pequeños llamados microservicios. Dentro de la arquitectura de microservicios, cada uno de ellos es un servicio único el mismo que desempeña una única funcionalidad de negocio de la aplicación permitiendo gestionar tareas independientes, donde cada microservicio se comunica con otros a través de interfaces sencillas.

Existen más conceptos para describir microservicios, Amazon Web Services, en su página oficial especifica que: “Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes. Las arquitecturas de microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelera el tiempo de comercialización de las nuevas características” [5].

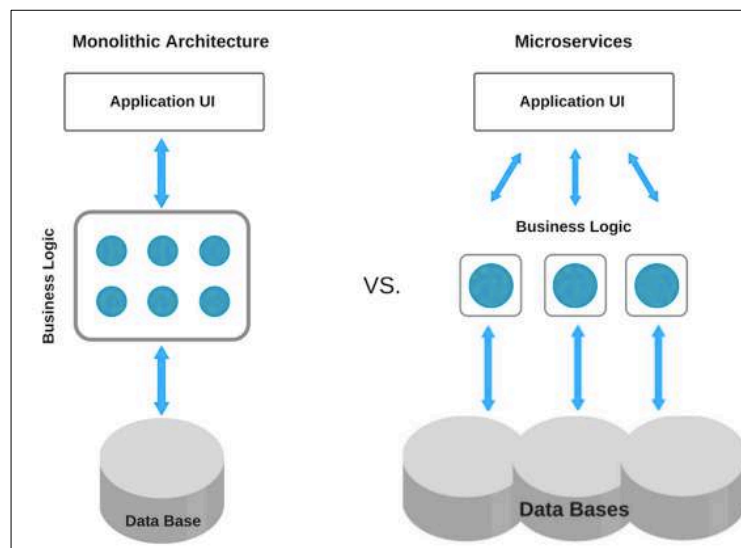


Figura 2 - Arquitectura Monolítica y Arquitectura con Microservicios [8]

Como podemos observar en la **Figura 2**, la arquitectura monolítica, parte izquierda, contiene toda la lógica del negocio en su solo componente, la arquitectura de microservicios está constituido de varios componentes y cada uno de ellos

conforman la lógica de negocio de la aplicación, lo cual permite solucionar el problema de interdependencia.

Una de las grandes ventajas de los microservicios es que sus componentes son totalmente encapsulados, permitiendo que cada servicio sea autónomo e independiente y encargado de una funcionalidad de negocio, además cada microservicio puede ser desarrollado e implementado de manera independiente, de igual manera su arquitectura interna, lenguaje de programación y tecnología pueden ser diferentes entre cada servicio.

2.2.1 Principios de arquitectura de microservicios

Los microservicios se enfocan en base a los límites del negocio, cada uno debe ser desplegado de manera aislada en una plataforma como servicio (PAAS) y sus comunicaciones son mediante llamadas en red. Newman [9], menciona que es posible elegir que principios adoptar, a continuación, se menciona los principios más relevantes [10]:

- **Modelar entorno a conceptos de negocios:** en base a la experiencia de interfaces basadas en límites del contexto de negocio son más estables, al ser modelados en dominios los cambios de los procesos de negocio se reflejan fácilmente.
- **Adoptar la cultura de automatización:** fortalecer una cultura de automatización con herramientas que soporten los microservicios, además se debe adoptar la integración continua para una rápida retroalimentación.
- **Esconder detalles internos de implementación:** definir los límites del negocio permitirá un buen soporte. Los servicios deben proveer APIS que permitan utilizar diferentes tecnologías, también, estos servicios deben ocultar su base de datos para evitar el acoplamiento.
- **Descentralizar todas las cosas:** buscar constantemente los cambios para delegar el control de los servicios a los equipos maximizando la autonomía de los microservicios.
- **Despliegue independiente:** hay que lograr que los microservicios puedan ser desplegados por ellos mismos aun cuando existan cambios deben

coexistir versiones diferentes permitiendo incrementar la autonomía de los equipos y optimizar la velocidad de nuevas características.

- **Aislar las fallas:** contar con un plan de fallas para que la arquitectura sea más resistente caso contrario el sistema fallará haciéndolo más frágil. Comprender el impacto de que pasaría si una parte del sistema se comporta erróneamente, a veces se debe sacrificar la disponibilidad o consistencia.
- **Altamente observables:** usar una vista holística y un monitoreo con transacciones de prueba que simulen un comportamiento real, es importante agregar logs y usar IDS correlacionados que permitan el rastreo de las llamadas del sistema.

2.2.2 Características de los microservicios

A continuación, se detallan las principales características de los microservicios [11]:

- **Alto nivel de desacoplamiento:** cada microservicio tiene su propio dominio, comunicándose y compartiendo datos con los demás.
- **Nombres únicos:** cada microservicio tiene su propio nombre para poder identificarlo fácilmente.
- **Resiliencia:** en caso de que se presente un problema, se podrá iniciar en otra máquina de forma que no se pierda ninguna información.
- **Independencia:** su implementación, escalado y actualización es de manera independiente.

2.2.3 Ventajas de microservicios

Las ventajas de los microservicios son lo suficientemente fuertes como para que algunas grandes empresas como Amazon, Netflix y eBay hayan adoptado esta arquitectura. En comparación con las estructuras de diseño más monolíticas, los microservicios ofrecen [11]:

- **Facilidad de comprensión:** con una mayor simplicidad, los desarrolladores pueden entender mejor la funcionalidad de un servicio.
- **Despliegues más pequeños y rápidos:** con despliegues más rápidos permite comenzar a explorar los beneficios del despliegue continuo.

- **Escalabilidad:** dado que sus servicios están separados, puede escalar más fácilmente en cada servicio, en lugar de toda la aplicación.
- **Total libertad:** del desarrollador a la hora de desplegar o desarrollar servicios de manera independiente.

2.2.4 Desventajas de microservicios

La principal desventaja de los microservicios es la complejidad que tiene cualquier sistema distribuido [11]:

- **La comunicación entre servicios es compleja:** dado que los servicios son independientes, hay que manejar cuidadosamente las peticiones que viajan entre los módulos.
- **Más servicios equivalen a más recursos:** por la gestión de múltiples bases de datos y transacciones es necesario tener los recursos suficientes para tener un óptimo desarrollo.
- **Las pruebas globales son difíciles:** para probar una aplicación basada en microservicios se debe confirmar que cada servicio sea dependiente antes de realizar las pruebas.

2.2.5 Arquitectura de comunicación directa cliente – microservicio

Al momento de crear aplicaciones con microservicios se debe tener en cuenta la comunicación entre las aplicaciones (web o móvil) y los microservicios (backend), considerando el tamaño y la escalabilidad de las aplicaciones, por ejemplo, cómo podemos ver en la **Figura 3** para una aplicación sencilla sin tantos microservicios, la comunicación directa entre cliente – microservicio sería suficiente ya que cada cliente web o móvil podría realizar request directamente a los microservicios que necesita respuesta [12].

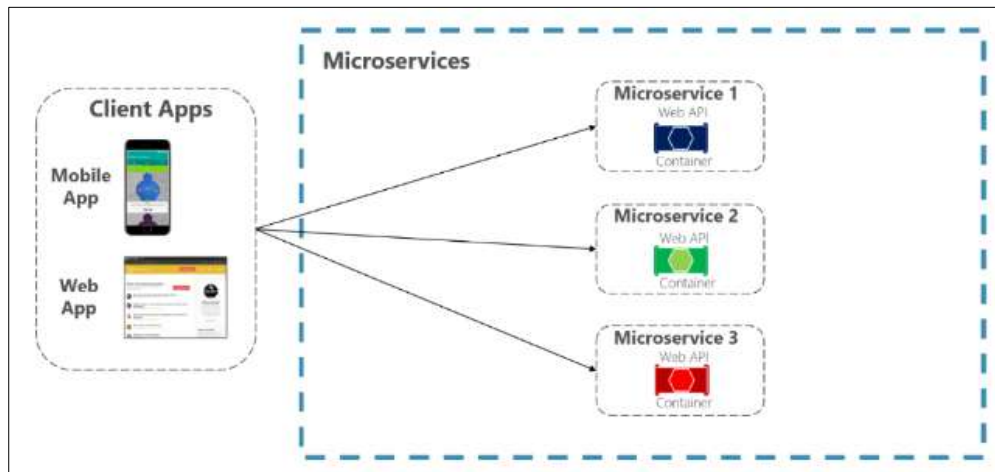


Figura 3 - Arquitectura de Comunicación Cliente – Microservicio [12]

Por otro lado, para aplicaciones con un nivel de escalabilidad mayor, que exista un balanceador de carga en medio del cliente y los microservicios sería una mejor opción, el balanceador de carga se encargaría de distribuir las solicitudes de los clientes hacia los microservicios, brindando un orden y equilibrio en los request del cliente, así como también proporcionando seguridad al tener una terminación SSL [12].

2.3 Patrón modelo - vista - modelo vista

El patrón MVVM (Modelo – vista – modelo vista) es un patrón de desarrollo de aplicaciones móviles iOS que se centra en un desarrollo separando la parte del view y model (interfaz de usuario de la parte de la lógica de negocios). Uno de los principales beneficios de este patrón es evitar conflictos y errores en el desarrollo y facilitar la etapa de pruebas [13].

Trabajar con este patrón de arquitectura permite crear planes de prueba incluso después de iniciada la etapa de desarrollo del software también presenta mejoras en características como reutilización de código, lo que permite una mejor colaboración entre los desarrolladores, en la creación de los diferentes módulos de la aplicación [13].

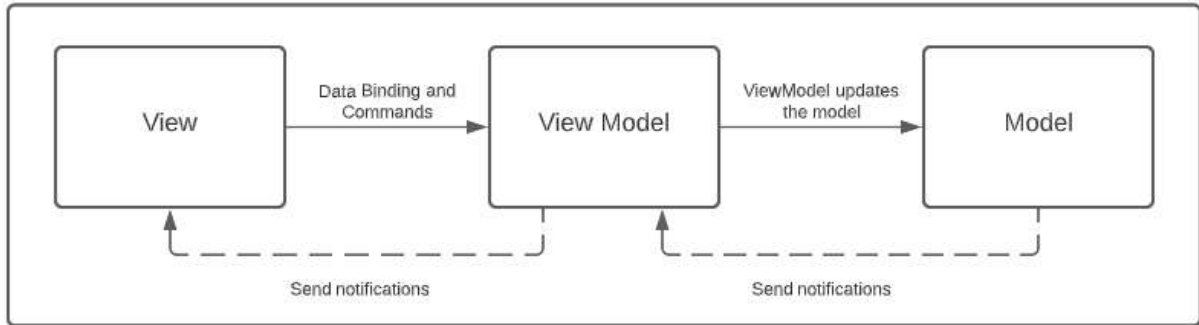


Figura 4 - Patrón MVVM [14]

2.4 Aplicación móvil

“Una aplicación móvil consta esencialmente de dos partes: las aplicaciones nativas y las webs móviles. Sin importar el tipo de aplicación que se decida usar, ambas deben proporcionar la misma calidad de información a los usuarios que las utilizan” [15].

Las aplicaciones creadas bajo el nombre de aplicaciones nativas nos permiten tener acceso a un gran número de productos, exclusivos o no, de las diferentes bibliotecas con las que fueron desarrolladas, las bibliotecas para este desarrollo pertenecen a las tiendas de aplicaciones de iTunes Store y Google Play Store [15].

Al momento de optar por desarrollar una aplicación móvil nativa, se obtiene una gran ventaja, que es la personalización de las bibliotecas para una mejor información, productos y nuevos servicios, garantizando que la aplicación llegue a sus usuarios principales, sin intermediarios de terceros [15]

También se debe considerar algunos aspectos al momento de desarrollar aplicaciones móviles, estos son el costo de desarrollo, el espacio de navegación y el lenguaje de programación, como sabemos los lenguajes de programación cambian dependiendo del sistema operativo del dispositivo, lo que implica un esfuerzo extra por parte del desarrollador al pensar la programación de la misma función para diferentes dispositivos [15].

2.5 Aplicación web

Aplicación web es el nombre que se le asigna a las aplicaciones que se comunican con un protocolo de internet (protocolo HTTP), que pertenece a la familia de

protocolo de comunicación TCP/IP y facilita la comunicación y la transmisión de datos [16].

Trabajan comúnmente con un modelo cliente/servidor, en el cual el cliente es la parte que interactúa con el usuario, y lo hace mediante un navegador web, un explorador o un visualizador. Por lo general el cliente de una página web está desarrollado en lenguaje HTML, con funcionalidades que pueden estar desarrolladas en JavaScript, la mayoría de las aplicaciones web cuentan con una serie de plugins que permiten visualizar y tener acceso a otro tipo de contenidos [16].

Por otro lado, el servidor es la parte donde se almacenan y se procesan los datos, este programa está pendiente y en espera de todas las solicitudes de transacciones que envía el cliente web mediante el protocolo de comunicación, el procesamiento de datos se puede dar, mediante scripts que se ejecutan en el lado del servidor, otra opción de procesamiento de datos es mediante recursos adicionales como por ejemplo, documentos, enlaces o archivos multimedia, todo esto se encuentra disponible en respuesta a la solicitud del cliente web [16].

2.6 JSON

Por sus siglas en inglés significa “Notación de Objetos de JavaScript”, es un formato utilizado para el intercambio de datos entre programas, independientemente del lenguaje de programación en el que se esté desarrollando la aplicación. Es un formato universal, lo que quiere decir que todos los lenguajes de programación lo soportan [17].

2.7 API

Por sus siglas significa “Interfaz de Programación de Aplicaciones” y su objetivo principal es permitir la comunicación entre componentes de las aplicaciones de software, siguiendo una arquitectura cliente - servidor, su funcionamiento es sencillo, el cliente es el que envía la petición y el servidor es el que recibe y envía una respuesta a esa petición [18].

2.8 Backend

Es una capa de acceso a datos, es decir se encarga del procesamiento de los datos, y trabaja en conjunto con el frontend, un backend desarrollado correctamente debe tener ciertas características de seguridad, base de datos y framework, ya que se encarga de manipular los datos desde que llegan del frontend hasta que son almacenados en la base y consumidos por las diferentes APIs de la aplicación [19].

2.9 Frontend

Es la parte de la aplicación denominada la parte de presentación, es decir es una interfaz visual que está en contacto con el usuario y permite que este pueda interactuar con el sitio web, realizando las peticiones o solicitudes y receptando los datos requeridos por las solicitudes para ser enviados al backend [19].

2.10 T-Shirt Sizing

Es un sistema que sirve para estimar y planificar la capacidad de un proyecto, esta herramienta nos ayudará a determinar que tarea o funcionalidad va a requerir mayor esfuerzo, este sistema trae beneficios para cualquier proyecto, debido a que puede ser usado por cualquier miembro del equipo, por ejemplo, los colaboradores individuales que mantienen una mejor comunicación sobre las prioridades, los líderes del proyecto pueden evaluar la capacidad del equipo, y los miembros del equipo que entienden que tarea realizar y la fecha de entrega de la misma [20].

2.11 Mockups

Es un diseño de una pantalla de una aplicación web o de una aplicación móvil, sirve para presentar todos los elementos finales que tendrán las pantallas en los dispositivos después de su desarrollo, no es funcional, es decir, que solo es una presentación estática del diseño de las pantallas [21].

El objetivo de los mockups en el diseño de aplicaciones es muy importante, es la manera en la que el diseñador prueba como queda la combinación y el funcionamiento de todos los elementos visuales considerados para la aplicación,

sin tener que realizar cambios y pruebas en el código en el proceso de desarrollo de la aplicación [21].

2.12 Base de datos SQL

Lenguaje de Consulta Estructurado o por sus siglas en inglés SQL permite gestionar y administrar los datos y las relaciones que existen entre ellos. Este tipo de lenguaje se usa también para diseñar, crear y modificar las tablas en las bases de datos de tipo relacional en las cuales dichas tablas tienen columnas con nombre únicos [22].

Al ser un estándar reconocido internacionalmente el uso de este lenguaje demanda una sintaxis específica que está formada por elementos del idioma inglés y varios elementos de la sintaxis de Visual Basic para aplicaciones [22].

```
SELECT Last_Name  
FROM Contacts  
WHERE First_Name = 'Mary';
```

Figura 5 - Instrucción SQL [22]

2.13 Base de datos NoSQL

Su nombre hace referencia a base de datos no relacionales, es decir, una base de datos que no está estructurada a base de tablas, estas bases no SQL trabajan con documentos para almacenar y recuperar los datos en formatos diferentes a las tablas [23].

Estas bases permiten tener un mejor desempeño al trabajar con aplicaciones modernas, debido que estas generan una gran cantidad de datos complejos y que cambian constantemente. Una ventaja de estas bases de datos es que se puede agregar nuevos datos sin tener que realizar una modificación en el diseño del esquema de la base de datos [23].

2.14 Scrum



Figura 6 - Proceso SCRUM [24]

Es un marco de trabajo ágil, que permite manejar de manera organizada las tareas contando con una planificación necesaria para lograrlas mediante las reuniones diarias asegurando que el proceso está en la etapa correcta, este marco de trabajo nos ayuda a trabajar exitosamente gracias a la división, asignación de roles y tareas apreciando su progreso de una manera transparente y acorde a lo establecido [24].

Para comprender la forma de trabajo de Scrum se debe explicar los componentes más importantes que son los roles y los elementos:

2.14.1 Roles de Scrum

- **Product Owner:** es el encargado de elaborar el Product Backlog, que contiene las ideas del cliente ordenadas por prioridad, es la persona que entiende el modelo de negocio del cliente, conoce el producto del cliente y toma las decisiones.
- **ScrumMaster:** es la persona que se encarga de que la metodología y el modelo de desarrollo operen de una manera funcional. Se encarga de eliminar los inconvenientes que se presenten durante el desarrollo del producto o que entorpezca el proceso, tiene interacción directa con los desarrolladores y el cliente.

- **Equipo de desarrollo:** es un equipo conformado por una cantidad de 5-9 personas encargadas de llevar a cabo el proceso de creación del producto la toma de decisiones y la organización la realizan centrándose en conseguir su objetivo.
- **Usuario:** es la persona que va a utilizar el producto al final.
- **Stakeholders:** es la persona o personas que se interesan en el desarrollo del producto, son los que obtendrán un beneficio del producto.
- **Manager:** son los encargados de tomar las decisiones finales, colaborando en el planteamiento de objetivos y de los requisitos del producto.

2.14.2 Elementos de Scrum

- **Product Backlog:** es la lista de necesidades del cliente.
- **Sprint Backlog:** es el listado de tareas que se realizaran en el sprint.
- **Incremento:** es la entrega al final de un sprint, la parte finalizada y totalmente funcional del sprint.

2.14.3 Reuniones Scrum

- **Planificación del Backlog:** se crea el documento con las necesidades del cliente los requisitos del sistema ordenados por prioridad, en este momento se planifica el Sprint 0, en el cual se plantean los objetivos en esa iteración.
- **Seguimiento del Sprint:** en esta etapa se realizan reuniones diarias en las que se evalúan tres cosas:
 - a) ¿Qué trabajo se realizó?
 - b) ¿Qué trabajo se realizará en la nueva reunión?
 - c) ¿Qué inconvenientes se han presentado y que se debe solucionar para poder continuar con el desarrollo del proyecto?
- **Revisión del Sprint:** al momento de terminar un Sprint se realiza una evaluación del incremento o entrega que se obtuvo de dicho Sprint, así como también se presentan los resultados finales con el objetivo de mejorar el feedback del cliente.

2.15 Pruebas Funcionales

Son un tipo de pruebas que tienen como objetivo determinar el correcto comportamiento del software, centrándose en los requisitos empresariales de la aplicación y si cumplimiento, es decir, solo verifican el resultado final de una acción, tomando en cuenta desde el punto de vista del usuario final del producto [25].

El proceso para probar las funciones del software se realiza analizando las especificaciones del software para luego elaborar los casos de prueba, con esto cada caso de prueba se realiza ingresando los datos de entrada que requiere cada función del producto e interpretando y entendiendo las salidas. Este tipo de pruebas se centra en demostrar inconsistencias entre la especificación inicial de las funciones del producto de software y el resultado final después de su desarrollo, por ejemplo:

2.15.1 Pruebas de aceptación

Este tipo de pruebas tienen como objetivo probar y verificar la calidad del software y que este cumpla con las expectativas planteadas por el cliente, y que vayan de acuerdo con los requisitos definidos al inicio del proyecto, las pruebas de aceptación tienen lugar antes de que la aplicación pase a la etapa de producción [25].

El procedimiento para realizar estas pruebas es con un “test de aceptación”, que una vez realizado permite definir el nivel de calidad de la aplicación. Como el objetivo principal de estas pruebas es verificar la calidad y que la aplicación este apta para el uso, se basan en especificaciones de requerimientos de usuario, especificaciones de requerimientos del sistema y procesos de negocio [25].

2.16 Pruebas no funcionales

Este tipo de pruebas, como su nombre lo indica se encargan de medir aspectos no funcionales del software como, por ejemplo: usabilidad, el rendimiento, seguridad, fiabilidad, eficiencia y compatibilidad. Entre algunos ejemplos de este tipo de pruebas tenemos:

2.16.1 Pruebas de carga

Este tipo de prueba se centra en la medición del sistema y su comportamiento al momento de administrar o generar una carga en la aplicación, la carga aplicada al sistema para la prueba puede ser diferente, por ejemplo, la cantidad de usuarios trabajando simultáneamente en el programa, el número de peticiones o request aplicados al producto [26].

2.16.2 Pruebas de rendimiento

Este tipo de pruebas tienen como propósito medir cuál es la velocidad de procesamiento de peticiones del sistema y el tiempo máximo que se demora al dar una respuesta en un ambiente de trabajo que asemeja a la vida real. Determinando problemas de concurrencia o escalabilidad que pueda presentar el resultado final del producto de software [26].

2.17 Herramientas utilizadas

❖ Para el desarrollo de aplicaciones web

- **React**

Es una herramienta que se usa en el desarrollo de aplicaciones web, específicamente una librería que se utiliza para crear interfaces de usuario, brindando al desarrollador la opción de diseñar estas interfaces para cada estado de su aplicación, basándose en componentes encapsulados que manejan su propio estado y que el desarrollador puede convertir en interfaces.

El funcionamiento de React es simple, una vez diseñadas y codificadas las interfaces, React se encarga de renderizar y actualizar todos los componentes conforme los datos cambian con el tiempo [27].

- **SpringBoot**

En el desarrollo de aplicaciones, es un framework de java que trabaja basándose en un modelo Vista-Controlador, este framework permite la integración más fácil con otros proyectos de Spring, posee servidores de

aplicación, por lo que su ejecución se facilita y no es necesario la creación de un WAR, la configuración de librerías de Spring o de aplicaciones de terceros se realiza de forma automática, estas características en conjunto permiten una fácil creación y despliegue de servicios REST, convirtiendo a Spring Boot en un buena opción para desarrollo de backend de las aplicaciones [28].

- **Eclipse IDE**

Es uno de los más usados por desarrolladores, el entorno de desarrollo integrado, conocido por sus siglas como “IDE”, de eclipse presenta una versión de escritorio y una versión en la nube, lo que lo hace muy versátil y fácil de usar para los desarrolladores de aplicaciones en java [29].

Eclipse se combina con diferentes idiomas y varias funciones en cualquiera de los paquetes de java, el IDE de Eclipse en la nube permite desarrollar aplicaciones donde quiera que te encuentres, los desarrolladores utilizan el navegador como entorno de desarrollo, instalar paquetes de escritorio par un mejor desarrollo de aplicaciones Java, JavaScript, CSS y HTML.

- **Visual Studio Code**

Es un editor de código, principalmente utilizado para desarrollo de aplicaciones web y JavaScript, posee extensiones que son compatibles con casi cualquier lenguaje de programación, brinda resaltado de sintaxis, funciones de autocompletar, terminaciones de tipos de variables, funciones y módulos.

Este IDE viene optimizado para trabajar y ser compatible con Git, permite revisar el código, detectar diferencias y solucionarlas directamente desde el editor de código, este IDE permite desarrollar en Java, Python, C/C++, C#, PHP, HTML/CSS, entre otros [30].

❖ **Herramientas para el desarrollo de aplicaciones móviles**

- **Xcode**

Es una aplicación que permite a todos los desarrolladores de Apple crear aplicaciones para Mac, Apple TV, iPad, iPhone y Apple Watch, el IDE de Apple ofrece una gran cantidad de facilidades, por ejemplo, compiladores de lenguajes como: Swift, C, C++ y Objective-C. Para que las aplicaciones creadas sean sencillas y divertidas al momento de desarrollar [31].

Al momento de trabajar con Xcode los desarrolladores cuentan con entorno que posee un flujo de trabajo unificado que permite diseñar, codificar, probar y eliminar errores de la aplicación.

Las características más importantes de Xcode son:

- Utiliza el lenguaje de programación Swift que es más seguro, moderno y rápido al momento de desarrollar.
- Los errores en la codificación se muestran en vivo mientras se crea la aplicación.
- Identifica los cuellos de botella en el rendimiento de la aplicación y proporciona formas de identificar la causa.

- **iOS**

Es un sistema operativo creado y utilizado por Apple, su nombre proviene de las siglas de "iPhone Operative System", el objetivo principal era utilizarlo en el iPhone, pero también fue utilizado en diferentes dispositivos como el iPod y en tabletas iPad, hasta la salida del iPadOS. La diferencia con el sistema operativo Android de Google es que iOS solo puede ser utilizado en los dispositivos de Apple, de igual manera nos permite agregar funciones adicionales a las que vienen por defecto en el iPhone, para lo cual podemos ingresar a la tienda de aplicaciones llamada AppStore y descargar aplicaciones con funciones que se necesite para el iPhone [32].

La última versión de este sistema operativo trae nuevas funcionalidades que facilitaran la interacción y el uso de los dispositivos que cuenten con este sistema operativo. Por ejemplo, la seguridad es importante para los usuario y desarrolladores por igual, brindando a los usuarios un respaldo de su información mediante datos móviles y no solo mediante la red WIFI [33].

Para la creación de aplicaciones en un entorno Apple, los denominados Apple developers aprender y se especializan en un lenguaje de programación específico llamado Swift.

- **Swift**

Es un lenguaje de programación muy versátil, seguro, e interactivo para escribir software de manera precisa y rápida para teléfonos, computadoras, servidores o cualquier dispositivo sea capaz y necesite de la ejecución de código, al tratarse de un lenguaje con una comunidad de código abierto, las diferentes actualizaciones y versiones proporcionadas por los diferentes colaboradores han llevado a que el compilador sea optimizado para el rendimiento y el lenguaje sea optimizado para el desarrollo, brindando a los Apple developers un ambiente de desarrollo seguro con buen rendimiento [34].

El lenguaje Swift es tan interactivo que permite cambios en el código y visualizar los resultados de forma inmediata, sin tener la necesidad de crear y ejecutar una aplicación cada vez que se experimente o se agregue nuevas funcionalidades en el código. Es por esta razón que este lenguaje se convierte en una gran opción para los desarrolladores novatos o los nuevos programadores en camino a ser Apple developers.

Swift agrupa de cierta manera buenas prácticas o patrones de programación con el objetivo de definir y dar solución a grandes errores o a errores comunes para los desarrolladores, entre algunas de ellas tenemos:

- Las variables se inicializan antes de ser usadas.
- La memoria se gestiona de manera automática.
- Los enteros se comprueban en busca de desbordamiento.

- La gestión de errores permite la recuperación en caso de fallas.

El código realizado en Swift está optimizado para sacar el mayor provecho a la nueva tecnología del hardware de las computadoras, esto hace que Swift sea una buena opción para cualquier aplicación que el desarrollador tenga en mente, desde un "Hello World", a un videojuego, o hasta todo un sistema operativo [35].

- **Swift UI**

Es un framework moderno y revolucionario para diseñar y crear interfaces de usuario con sintaxis de programación del lenguaje Swift [36]

Swift UI ofrece reemplazo dinámico, ya que la aplicación está constantemente siendo codificada y ejecutada, se puede ver y modificar los componentes y sus características en tiempo real. Permite crear layouts avanzados que pueden ser reutilizados para ahorrar tiempo y mantener una homogeneidad en las interfaces de usuario.

- ❖ **Herramientas para el control de versiones**

- **Git**

Es una aplicación que te permite gestionar las versiones de cualquier aplicativo móvil o web en desarrollo. Git es controlador de versiones más usado en la actualidad por los desarrolladores de grandes empresas. Una de las grandes ventajas de git es que al ser de código abierto el sistema es gratuito, otra ventaja es su poco espacio y su gran eficiencia, lo que lo hace una buena opción para el desarrollo de proyectos grandes y pequeños [37].

Git ofrece codificación basada en roles, es decir crea una rama para cualquier fase, por ejemplo, de producción, u otra en la que se realicen pequeñas pruebas, o ramas para cada función de la aplicación.

- **GitHub**

Es un entorno de control de versiones, que permite al desarrollador contribuir en proyectos de una manera rápida, segura y en un entorno amigable y

automático que facilita el trabajo. La seguridad se ofrece al momento de escribir código, se puede revisar automáticamente cada cambio realizado e identificar las posibles vulnerabilidades [38].

Con GitHub puedes gestionar los proyectos de la mejor manera, manteniendo las características, los errores bien organizados, para que todos los equipos de trabajo puedan acceder y tener conocimiento, se puede planificar reuniones y coordinar iniciativas para el desarrollo o solución de errores con elementos gráficos como tablas, pizarras digitales y listas de tareas.

- **Sourcetree**

Es una aplicación que permite simplificar como el desarrollador interactúa con los repositorios de Git, con el objetivo de que se centre en la codificación y desarrollo de sus aplicaciones, dejando de lado las líneas de comandos, facilitando la administración y visualización de repositorios a través de la GUI que ofrece la aplicación [39].

- ❖ **Herramientas para almacenamiento y autenticación**

- **Firebase**

Firebase ofrece una base de datos fácil de usar con una rápida lectura y escritura de datos, también permite almacenar datos y organizarlos con el objetivo de facilitar la verificación y las actualizaciones de los datos de las aplicaciones móviles [40].

El almacenamiento y gestión en tiempo real de los datos es posible en cualquier aplicación, desarrollada en cualquier sistema operativo como Android, iOS y las denominadas “Web- based applications” o aplicaciones web. Firebase provee distintos servicios y facilidades para un mejor desarrollo de tus aplicaciones, por ejemplo:

- **Cloud Firestore**

Es el servicio de base de datos NoSQL en la nube, que mantiene los datos sincronizados en tiempo real entre todos los clientes, con integración sin

interrupciones y sin importar el dispositivo móvil o web, todo esto gracias a sus características de flexibilidad de datos, asistencia sin conexión, actualizaciones en tiempo real, consultas expresivas y escalabilidad al momento de desarrollar las aplicaciones web, móvil o servidores [41].

- **Firestore Authentication**

La seguridad es importante en todas las aplicaciones móviles, este servicio de firestore nos permite identificar al usuario que está intentando acceder, esto permite guardar la información en la nube y permitir el acceso de esta en todos los dispositivos del usuario [42].

La autenticación de firestore permiten integraciones que facilitan el acceso del usuario a la aplicación, por ejemplo: Google, Twitter, Facebook, número de teléfono, mediante contraseñas o proveedores de identidad. Su funcionamiento es simple, para acceder a la aplicación se debe obtener las credenciales de acceso y estas pueden ser el correo electrónico (usuario de Facebook, Twitter), o un token otorgado por la entidad proveedora.

- **Cloud Storage**

Este servicio nos permite almacenar los datos sin importar el tipo, es decir, los datos utilizados y almacenados por la aplicación pueden ser imágenes, videos, audio, direcciones, texto, números, etc. Este servicio brinda seguridad en cualquier operación de carga y descarga de los datos, adicionalmente permite el ahorro de ancho de banda a los usuarios, debido a que, si la conexión es baja o presenta latencias, el usuario puede reintentar la operación desde donde la dejó [43].

- **PostgreSQL**

Es un programa que permite la gestión de una base de datos, PostgreSQL es el más avanzado sistema de base de datos relacional, una de sus ventajas es su código abierto, lo que lo hace muy fácil de acceder y de operar por parte de

los usuarios, este sistema ofrece confiabilidad de los datos, características fuertes para las operaciones CRUD en la base de datos y un alto rendimiento.

PostgreSQL brinda soporte para la creación de triggers, permitiendo una mejor administración de eventos y generando acciones cuando estos se disparan, presenta una alta concurrencia, es decir, que la base de datos puede atender las peticiones de varios clientes al mismo tiempo, mostrando la misma información sin presentar bloqueos [44].

❖ Para dockerización

- **Docker**

Es una plataforma que trabaja colaborativamente con AWS, que facilita a los desarrolladores con la creación, prueba e implementación de aplicaciones en la nube, también ofrece una administración más confiable para ejecutar aplicaciones de pequeña o gran escala. Docker funciona empaquetando el software o las aplicaciones en contenedores, estos incluyen todas las funciones para la ejecución de dicha aplicación, como las librerías, el código y las herramientas del sistema [45].

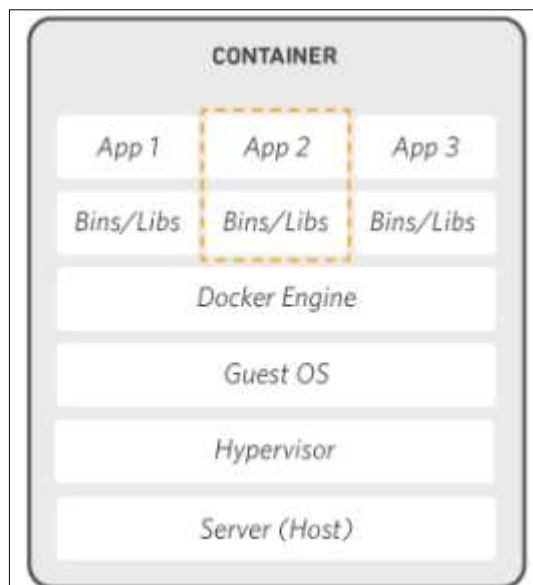


Figura 7 - Funcionamiento Docker [45]

❖ **Para realizar pruebas no funcionales**

- **Jmeter**

Es una herramienta que permite realizar testing a aplicaciones, con la finalidad de determinar sus puntos de quiebre, límites, rendimiento, comportamiento funcional. Las funcionalidades de esta aplicación son: Diseñar un test-plan con los objetivos a evaluar, ejecutar esa prueba plan y mostrar de distintas formas los resultados que arroja el test plan. Las pruebas que se pueden realizar con Jmeter son diversas como, realizar una carga pesada al servidor de la aplicación [46].

- **BlazeMeter**

Es una herramienta de pruebas de software, utilizada para probar APIs, aplicaciones móviles y aplicaciones web, esta aplicación fue optimizada para que los desarrolladores puedan realizar pruebas a sus aplicaciones, ejecutando planes de prueba, recibir reportes, gráficos con los resultados de las pruebas.

Esta aplicación ofrece planes de pruebas de carga, pruebas para funcionamiento, performance, código, monitoreo de APIs y pruebas de datos. Esta herramienta es muy versátil para los desarrolladores ya que se puede ejecutar en un entorno local y en la nube [47].

- **Postman**

Es una aplicación que te permite probar las HTTP request de tu programa con una mayor facilidad y mediante una interfaz gráfica amigable con el usuario, al ser un cliente HTTP la realización de pruebas API son mucho más sencillas para los desarrolladores y optimiza el tiempo de ejecución de pruebas. Una de las ventajas de esta aplicación es la creación de colecciones y diferentes ambientes para realizar pruebas.

Postman tiene varios métodos con los que los desarrolladores pueden realizar las pruebas de sus aplicaciones, entre los más utilizados tenemos: GET, POST, PUT, PATCH, DELETE [48]

- **Apptim**

Es un sistema creado para desarrolladores poner a prueba sus aplicaciones, con el fin de analizar errores y evaluar su impacto, para evitar que se activen problemas críticos. Este sistema trabaja directamente con el renderizado de la aplicación, tiempos, recursos, el consumo de energía, etc.

Apptim ofrece pruebas automáticas que los desarrolladores pueden usar en sus aplicaciones que corren en dispositivos locales y también en aplicaciones alojadas en dispositivos en la nube [49]

- ❖ **Para realizar los mockups**

- **JustinMind**

Es una aplicación utilizada por desarrolladores, UX, diseñadores, analistas de negocios que permite realizar el prototipo o mockup de cualquier aplicación web o móvil con gran facilidad y de una forma intuitiva para el usuario. Se puede crear los mockups solo arrastrando los componentes a la interfaz visual, sin tener que codificar nada.

En JustinMind se pueden crear prototipos más interactivos para los desarrolladores, simulan “clicks” y algunas funciones básicas, la biblioteca de componentes de esta aplicación es tan amplia que se puede crear aplicaciones móviles Android o IOS y crear una excelente experiencia de usuario [50].

2.18 Hábitos de lectura en la sociedad

El hábito de lectura ha ido disminuyendo de generación en generación, los niños y jóvenes, hoy en día asocian a la lectura como una obligación, una actividad que deben hacer o tareas en colegios y universidades, los adultos se excusan de falta de tiempo debido a sus trabajos y actividades diarias. Debido a esto las personas prefieren invertir su tiempo libre en acciones como revisar sus redes sociales o pasar viendo el teléfono móvil y no invierten en la lectura como una oportunidad de desarrollo personal.

Una consecuencia de esto es que en los hogares, escuelas y universidades exista menor cantidad de libros lo que disminuye el acceso a la literatura, sumado a esto el avance de la tecnología y las facilidades que ofrecen las aplicaciones en tendencia, complican el tener que decidir entre llevarse el teléfono móvil o un libro [51].

2.19 Impacto de la tecnología en la sociedad

Tomando en cuenta los beneficios que trae la tecnología en los aspectos sociales como la economía y en mayor medida en la educación, siendo parte importante en el desarrollo de los estudiantes, brindando facilidad de acceso a materiales académicos y como una importante fuente de capacitación y alfabetización digital, también se puede observar cambios en la forma de comunicación de la sociedad como reuniones virtuales o participación en comunidades de videojuegos. Es por esto por lo que las personas optan por utilizar más, los dispositivos [52].

Por tal motivo las empresas desarrolladoras de aplicaciones han optado por ofrecer al mercado aplicaciones que se adapten a las nuevas necesidades de las personas en campos como desarrollo personal y educación y lo han hecho con aplicaciones de reproducción de audiolibros.

Para este proyecto hemos realizado una comparación y un análisis de varios proyectos similares de software multimedia para dispositivos móviles, audiolibros educativos en Java y Android. Como resultado de esa investigación se determinó que la mayoría de las aplicaciones trabajan con actores y en lo que respecta al almacenamiento estos sistemas trabajan con una base de datos relacional con tablas, de forma local [53].

2.20 Stakeholders

La siguiente tabla define a las partes interesadas en la aplicación móvil y el aplicativo web (también denominados stakeholders):

Partes interesadas	
Stakeholder	Descripción
Usuario administrador	El usuario administrador podrá utilizar el aplicativo web para: iniciar sesión, ver todos los usuarios registrados y habilitar o deshabilitar sus cuentas de ser necesario.
Usuario autor	El usuario autor podrá utilizar el aplicativo web para: crear cuenta, iniciar sesión, ver su biblioteca personalizada en la cual mostrará un listado de los audiolibros que haya añadido, además podrá eliminar o editarlos, por último, podrá actualizar su información personal.
Usuario final	El usuario final podrá utilizar la aplicación móvil para: crear cuenta, iniciar sesión, ver los audiolibros del repositorio general, buscar audiolibros por nombre de autor, título, o género; reproducir un audiolibro, escribir una reseña sobre un audiolibro y calificarlo, ver su perfil y a su vez editar su información.

Tabla 1 - Partes Interesadas

CAPÍTULO 3 – DESARROLLO DE LA APLICACIÓN

3.1 FASE EXPLORATORIA

3.1.1. Alcance del proyecto

Este proyecto desarrolla un aplicativo web para el uso de los usuarios autores en la cual podrán añadir, eliminar y actualizar un audiolibro del repositorio, además de una aplicación móvil para el uso de los usuarios finales donde podrán reproducir, buscar un audiolibro y a su vez crear listas de reproducción. Dichas aplicaciones están desarrolladas bajo una arquitectura de microservicios con una comunicación cliente – microservicio.

3.1.2. Planificación del proyecto

Durante el proceso de desarrollo del producto se realizaron reuniones mediante la plataforma TEAMS con el magister Víctor Vicente Velepucha Msc., director del presente proyecto. Estas reuniones sirvieron para obtener los requerimientos necesarios de la aplicación móvil verificando que los criterios de aceptación se cumplan para cada requerimiento.

Se definieron los siguientes roles para el desarrollo del proyecto, siguiendo la metodología SCRUM.

ROLES DE SCRUM	
Scrum Master	Vicente Velepucha
Product Owner	Vicente Velepucha
Scrum Team (Equipo de desarrollo)	Felipe Andrade Violeta Brusil

Tabla 2 - Roles de Scrum

3.2. FASE DE INICIALIZACIÓN

3.2.1. Historias de Usuario

Para el desarrollo de este proyecto se realizaron reuniones con el Product Owner donde se manifestó la necesidad de crear un aplicativo web donde los usuarios

autores subirán audiolibros, el usuario administrador llevará un control de los usuarios registrados y una aplicación móvil donde los usuarios finales podrán utilizar todas sus funcionalidades. De ahí surgieron las historias épicas las cuales se dividieron en historias de usuario más pequeñas y manejables para el equipo de desarrollo. Dando un total de 20 historias de usuario para la obtención del producto final. El detalle de las historias épicas se encuentra en el **Anexo 1**.

Para las historias de usuario se definieron criterios de aceptación, mismos que son considerados criterios para comprobar si una historia de usuario fue desarrollada según la expectativa del dueño del producto. En el **Anexo 1** se detallan las historias de usuario con sus respectivos escenarios.

3.2.2. Product Backlog

Tras definir las historias de usuario por el equipo de desarrollo, se procedió a darles una estimación aproximada de tiempo con la técnica del tamaño de camisetas o “T-Shirt Sizing” [20], ya que después de un análisis se estableció el orden en el cual serian ejecutadas este análisis se obtuvo mediante la experiencia previa del equipo de desarrollo

La estimación aproximada se describe en la **Tabla 3**:

ESTIMACIÓN APROXIMADA DE TIEMPO	
T-Shirt Sizing	Valoración
S	<2hr
M	<4hr
L	<8hr
XL	>8hr

Tabla 3 – Estimación aproximada de tiempo con la técnica T-Shirt Sizing

Una vez definidas las estimaciones de cada una de las historias de usuario se definió el Product Backlog y el orden en el que se desarrolló cada parte de los

microservicios, el aplicativo web y la aplicación móvil, a continuación, se detalla en la **Tabla 4**:

PRODUCT BACKLOG				
	Código	Nombre	Estimación T-Shirt Sizing	
BACKEND AUDIOBOOK - CATALOGMICROSERVICE	HU03 - 01	Añadir audiolibro al repositorio	S	
	HU03 - 02	Eliminar audiolibro del repositorio	S	
	HU03 - 03	Editar la información de un audiolibro previamente añadido	S	
	HU04 - 01	Buscar los audiolibros existentes por autor	S	
	HU04 - 02	Buscar los audiolibros existentes por título de audiolibro	S	
	HU04 - 03	Buscar los audiolibros existentes por género	S	
	HU05 - 02	Reseñar un audiolibro	S	
FRONTEND APLICATIVO WEB	AUTHENTICATION FIREBASE	HU01 - 01	Crear cuenta como usuario autor con email y contraseña	M
		HU01 - 02	Crear cuenta como usuario autor con Google	M
		HU02 - 01	Autenticación usuario autor	M
		HU02 - 02	Editar información personal desde el aplicativo web	M
		HU02 - 05	Autenticación usuario administrador	M
		HU07 - 01	Habilitar o deshabilitar la cuenta de un usuario	M

	CONEXIÓN BACKEND	HU03 - 01	Añadir audiolibro al repositorio	M
		HU03 - 02	Eliminar audiolibro del repositorio	M
		HU03 - 03	Editar la información de un audiolibro previamente añadido	M
BACKEND USERPLAYLIST MICROSERVICE		HU06 - 01	Crear una lista de reproducción	S
		HU06 - 02	Agregar audiolibros a una lista de reproducción	S
FRONTEND APLICACIÓN MÓVIL	AUTHENTICATION FIREBASE	HU01 - 03	Crear cuenta como usuario final con email y contraseña	M
		HU01 - 04	Crear cuenta como usuario final con Google	M
		HU02 - 03	Autenticación usuario final	M
		HU02 - 04	Editar información personal desde la aplicación móvil	M
	CONEXIÓN BACKEND	HU04 - 01	Buscar los audiolibros existentes por autor	M
		HU04 - 02	Buscar los audiolibros existentes por título de audiolibro.	M
		HU04 - 03	Buscar los audiolibros existentes por género	M
		HU05 - 01	Reproducir un audiolibro	L
		HU06 - 01	Crear una lista de reproducción	S
		HU06 - 02	Agregar audiolibros a una lista de reproducción	M

		HU05 - 02	Reseñar un audiolibro	M
--	--	-----------	-----------------------	---

Tabla 4 – Product Backlog

3.2.3. Planificación de Sprints

Para el desarrollo del proyecto se planificó un total de 6 sprints para cumplir con el Product Backlog (desde sprint 1 al sprint 6), más 1 sprint adicional. Dedicado al diseño de la arquitectura de microservicios y configuración de repositorios de Git [37], y al diseño de mockups [21] para el aplicativo web como para la aplicación móvil.

SPRINTS						
0	1	2	3	4	5	6
Diseño de arquitectura de microservicios y diseño de mockups de las aplicaciones	Implementación del microservicio AudioBook-Catalog-Microservice	Implementación de frontend del aplicativo web y autenticación de usuarios	Conexión del microservicio AudioBook-Catalog-Microservice con frontend del aplicativo web	Implementación del microservicio UserPlayList-Microservice	Implementación de frontend de la aplicación móvil y autenticación de usuarios	Conexión de los microservicios con frontend de la aplicación móvil
	HU03 - 01	HU01 - 01	HU03 - 01	HU06 - 01	HU01 - 03	HU04 - 01
	HU03 - 02	HU01 - 02	HU03 - 02	HU06 - 02	HU01 - 04	HU04 - 02
	HU03 - 03	HU02 - 01	HU03 - 03	-	HU02 - 03	HU04 - 03
	HU04 - 01	HU02 - 02	-	-	HU02 - 04	HU05 - 01
	HU04 - 02	HU02 - 05	-	-	-	HU06 - 01
	HU04 - 03	HU07 - 01	-	-	-	HU06 - 02
	HU05 - 02	-	-	-	-	HU05 - 02

Tabla 5 - Planificación de Sprints

3.2.4. Ejecución de los Sprints

La ejecución de los Sprints se realizó en relación con la Planificación de Sprints como se muestra en la **Tabla 5**. A continuación se indica el detalle de cada uno:

3.2.4.1. Sprint 0

El Sprint 0 se enfocó en el diseño de la arquitectura de microservicios y de los mockups para la aplicación móvil como del aplicativo web

- **Objetivo del Sprint**

Diseñar la arquitectura de microservicios y mockups de la aplicación móvil y del aplicativo web.

- **Planning**

En este sprint se definirá el diseño de la arquitectura de microservicios para el proyecto, determinando el entorno y las herramientas que se utilizarán para el desarrollo del backend y frontend; igual se creará los repositorios Git para el control de versiones del proyecto. Además, se realizará los mockups del aplicativo web y de la aplicación móvil para lo cual se hará uso del aplicativo Justinmind [50].

- **Ejecución del Sprint 0**

- **Arquitectura de microservicios del proyecto**

Se definió que el proyecto tendrá una arquitectura de comunicación directa cliente microservicio, en este enfoque, cada microservicio tiene un puerto distinto [12], permitiendo la comunicación con cada uno de los clientes.

Para nuestro caso la arquitectura tendrá dos clientes un aplicativo web basado en React [27] y una aplicación móvil IOS basado en SwiftUI [34], el diagrama los representa como un dispositivo y un computador. Las aplicaciones utilizan la internet para tener comunicación con los microservicios los mismos que estarán contenidos en un Docker [45]: el aplicativo web se comunicará con el microservicio **“AudioBookCatalogMicroservice”**, el cual será responsable de

que el usuario autor pueda agregar, eliminar o actualizar un audiolibro en una base de datos PostgreSQL [44] la misma que estará contenida en un Docker [45].

La aplicación móvil se comunicará con los microservicios **“AudioBookCatalogMicroservice”**, que accederá a la información de los audiolibros y a su vez obtener los archivos multimedia de los mismos que se encontrarán almacenados en el **“Storage”** de Firebase [40]; y con **“UserPlayListMicroservice”**, el cual permitirá que el usuario final pueda crear listas de reproducción y agregar audiolibros guardando la información en una base de datos PostgreSQL [44] la misma que estará contenida en un Docker [45].

Para la autenticación de usuarios de las dos aplicaciones se utilizará **“Firebase Authentication”**, el mismo que permitirá autenticar a los diferentes usuarios guardando los registros en la base de datos **“Cloud Firestore”** de Firebase [40]. Esta arquitectura esta detallada en la siguiente figura:

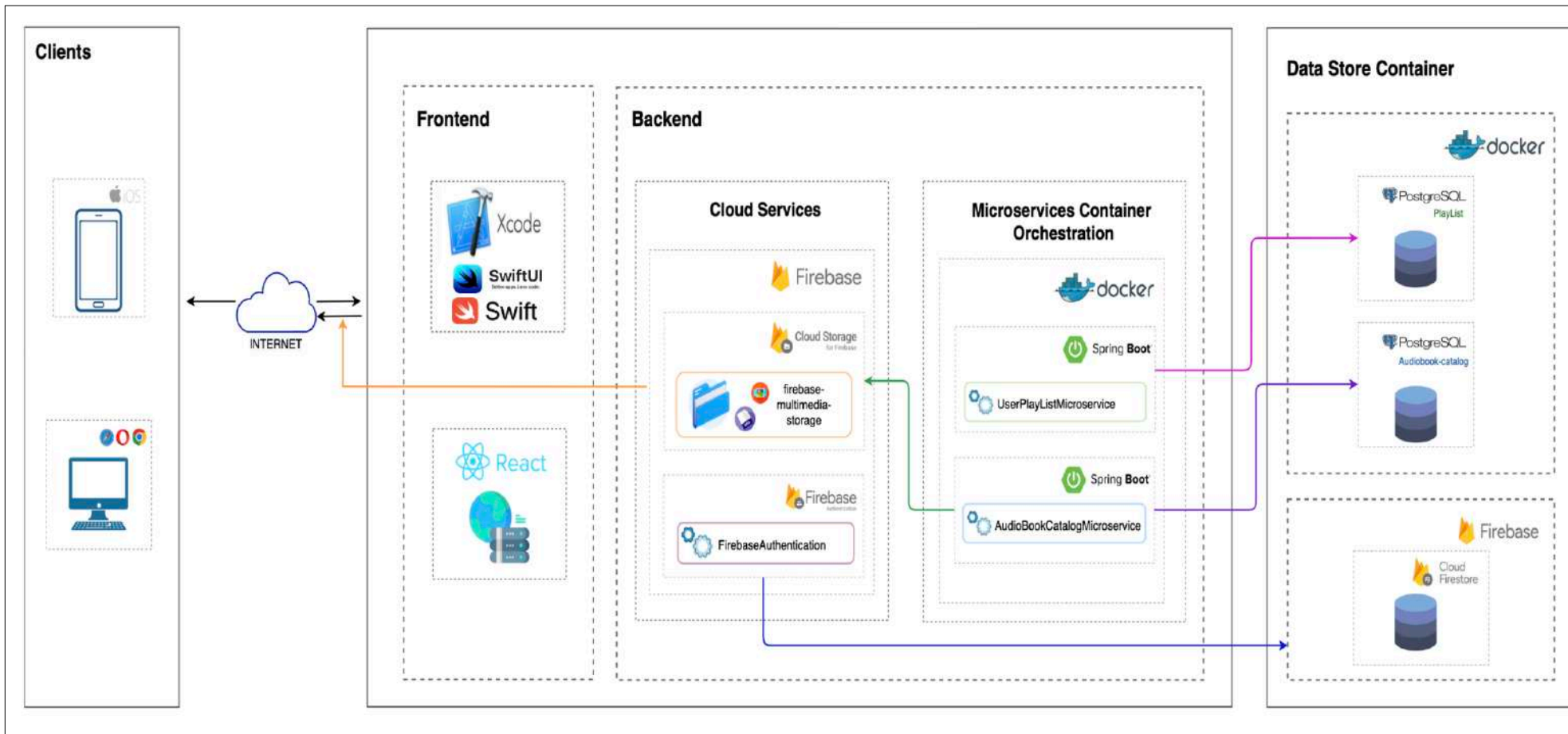


Figura 8 - Arquitectura de microservicios del proyecto

- **Repositorio Git y control de versiones**

Para el control de versiones de este proyecto se creó 4 repositorios en GitHub [38], que permitirá observar todos los cambios al código a lo largo del proyecto. El backend tiene dos repositorios para los microservicios, así mismo como el frontend para las dos aplicaciones respectivamente como lo podemos observar a continuación:

a) Repositorio del microservicio AudioBookCatalogMicroservice (Backend)

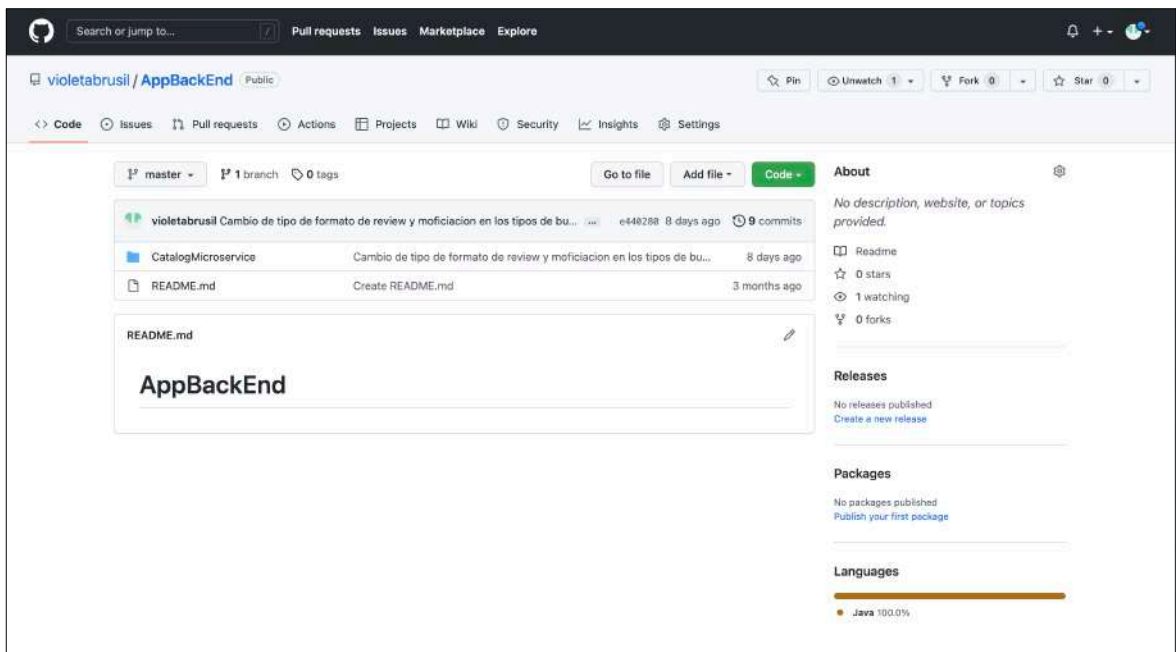


Figura 9 - Repositorio Git del microservicio AudioBookCatalogMicroservice

b) Repositorio del microservicio UserPlayListMicroservice (Backend)

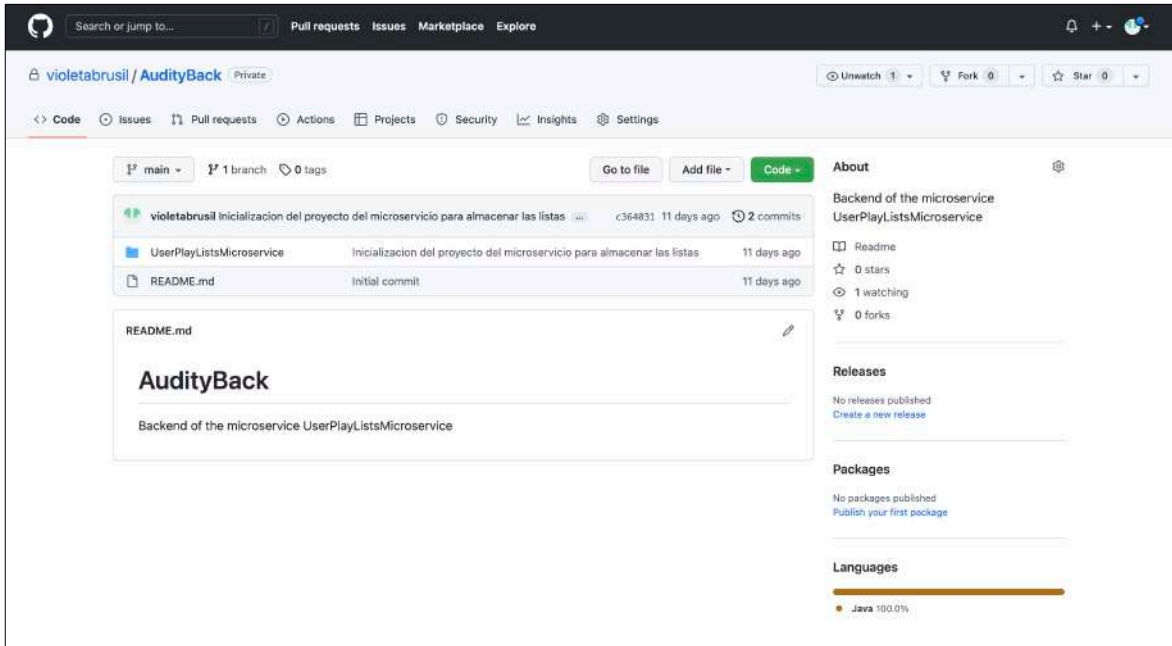


Figura 10 - Repositorio Git de UserPlayListMicroservice

c) Repositorio del aplicativo web (Frontend)

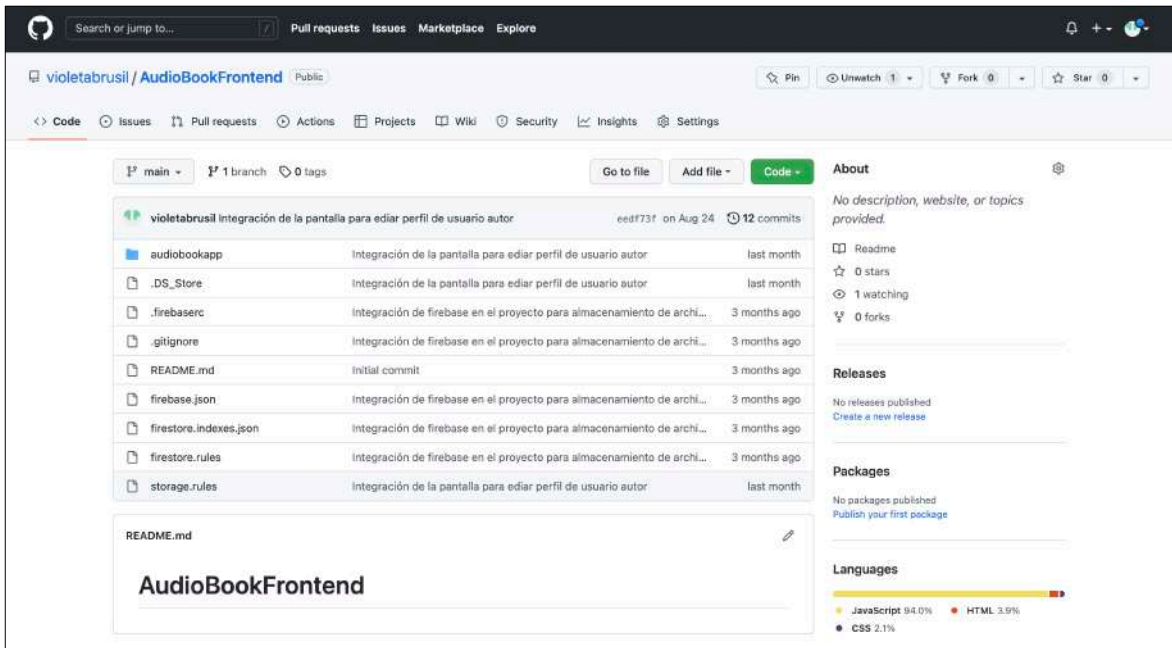


Figura 11 - Repositorio Git del aplicativo web

d) Repositorio de la aplicación móvil (Frontend)

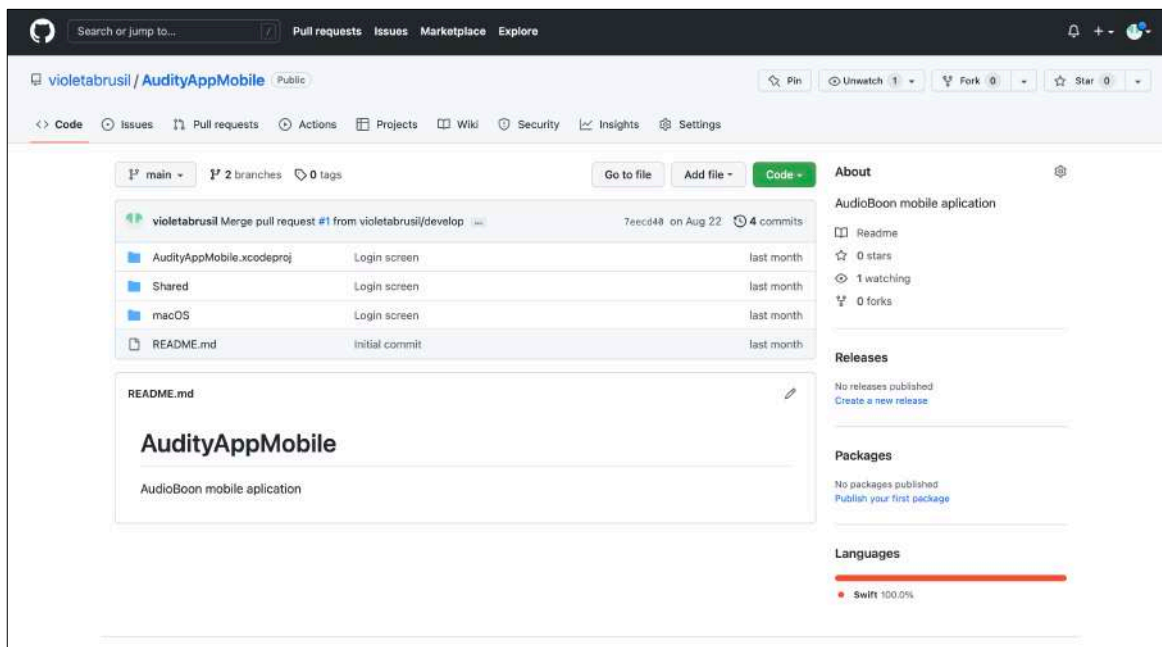


Figura 12 - Repositorio Git del aplicativo móvil

En el **Anexo 2** se encuentra los enlaces para acceder a cada repositorio.

- **Instalación de la herramienta Justinmind**

Para la instalación de la herramienta se realizó la descarga en la página oficial de Justinmind (<https://www.justinmind.com>), una vez que finalizó la descarga se realizó la instalación correspondiente.

- **Diseño del prototipo del aplicativo web**

Se creó un archivo para el diseño del aplicativo web el cual será utilizado por los usuarios autor y administrador, a continuación, se detalla cada una de las pantallas:

A. Pantalla de inicio de sesión del aplicativo web: desde la cual será el ingreso de los usuarios registrados de tipo autor y administrador al aplicativo web, una vez que realice el ingreso podrá acceder a su respectiva pantalla inicial.

Iniciar sesión

Email

Contraseña

Ingresar

Ingresar con Google

[¿Es nuevo aquí? Cree una cuenta](#)

Figura 13 - Mockup de la pantalla de inicio de sesión del aplicativo web

B. Pantalla de registro de usuarios al aplicativo web: desde la cual los usuarios podrán registrarse.

Regístrate gratis para empezar

Email

Contraseña

Nombre de usuario

Registrar

Figura 14 - Mockup de la pantalla de registro de usuarios al aplicativo web

C. Pantalla inicial para usuario autor: desde la cual el usuario podrá añadir eliminar o actualizar un audiolibro.

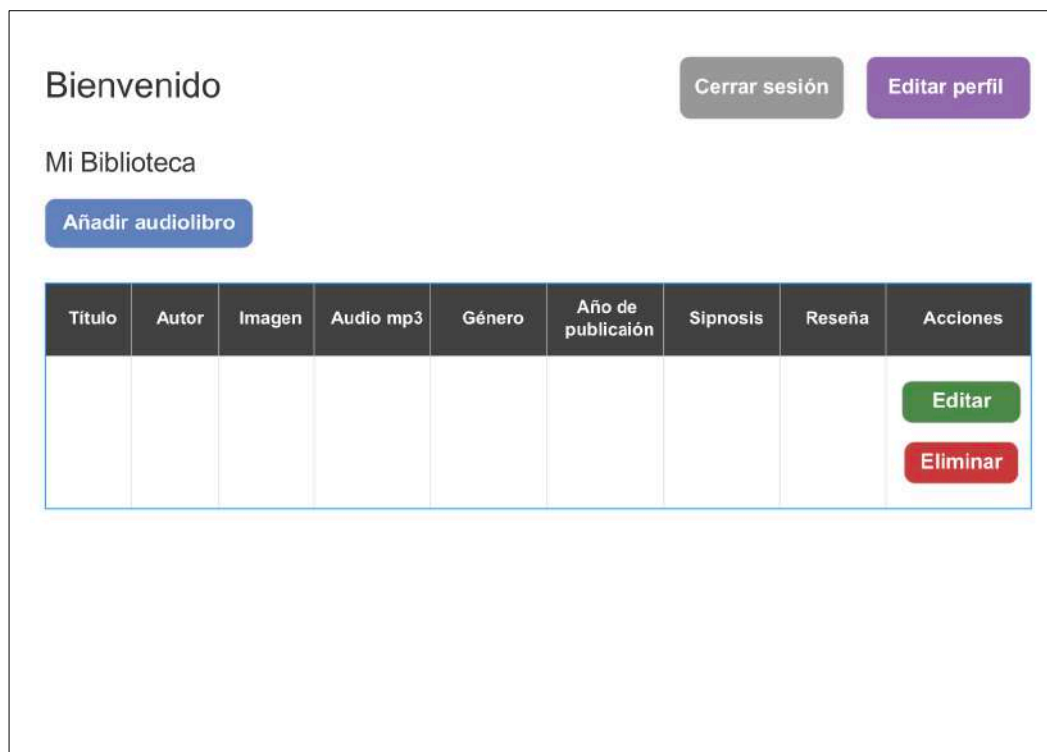


Figura 15 - Mockup de la pantalla inicial para usuario autor

D. Pantalla inicial para usuario administrador: desde la cual el usuario podrá habilitar o deshabilitar la cuenta de un usuario registrado.

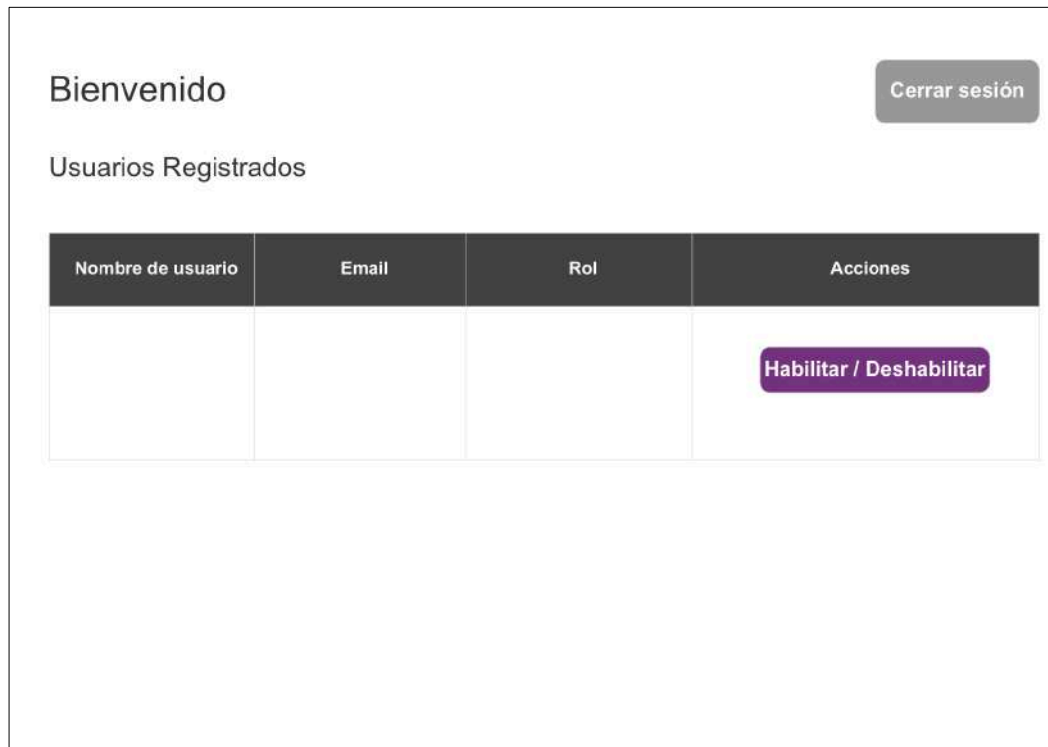


Figura 16 - Mockup de la pantalla inicial para usuario administrador

E. Pantalla para añadir audiolibro: desde la cual el usuario autor podrá añadir un nuevo audiolibro.

Añadir Audiolibro

Título

Autor

Síntesis

Género
Seleccione el género

Imagen

Audio

Año de publicación

Figura 17 - Mockup de la pantalla para añadir audiolibros

F. Pantalla para editar audiolibro: desde la cual el usuario autor podrá editar la información del audiolibro que elija.

Editar Audiolibro

Título

Autor

Sínpnosis

Género
Seleccione el género

Imagen

Audio

Año de publicación

Figura 18 - Mockup de la pantalla para editar audiolibros

G. Pantalla para editar información del usuario autor: desde la cual el usuario podrá editar su información personal.

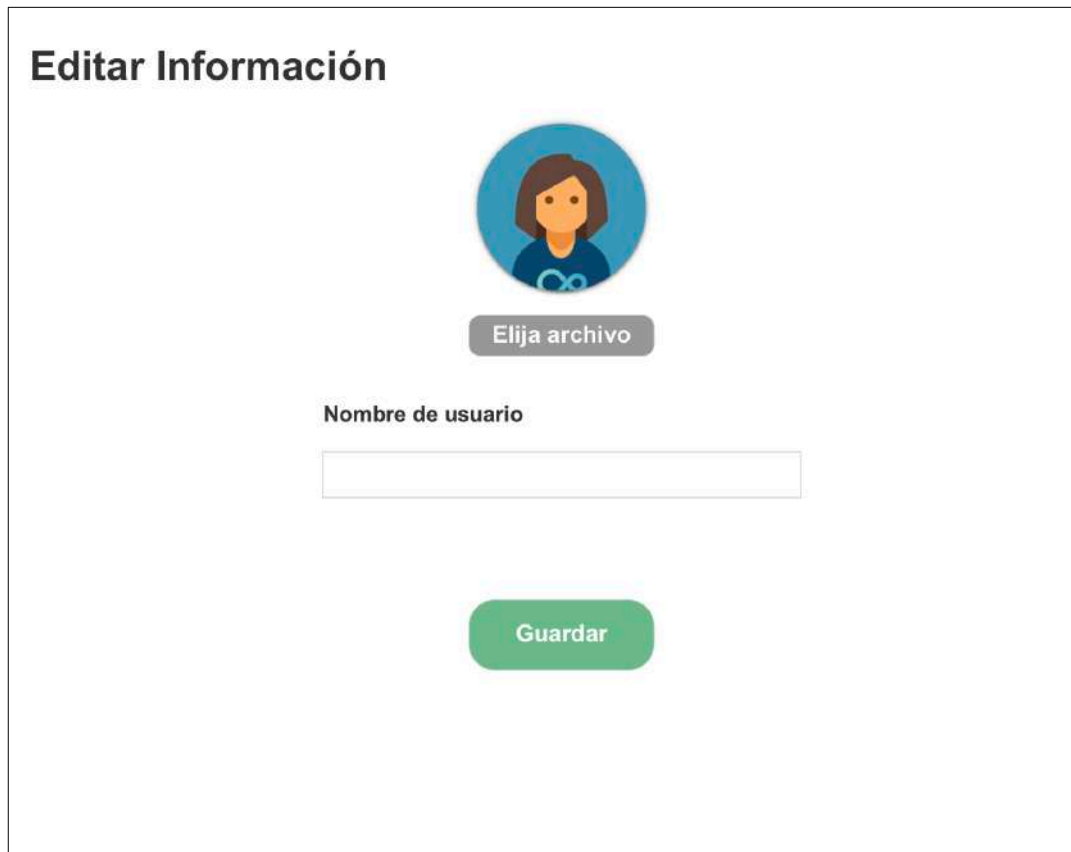


Figura 19 - Mockup de la pantalla para editar información del usuario autor

- **Diseño del prototipo de la aplicación móvil**

Se creó otro archivo para el diseño de la aplicación móvil la cual será utilizado por los usuarios finales, a continuación, se detalla cada una de las pantallas:

A. Pantalla de inicio de sesión de la aplicación móvil: desde la cual el usuario final podrá iniciar sesión a la aplicación móvil.



Figura 20 - Mockup de la pantalla de inicio de sesión de la aplicación móvil

B. Pantalla de registro de usuarios a la aplicación móvil: desde la cual los usuarios podrán registrarse.

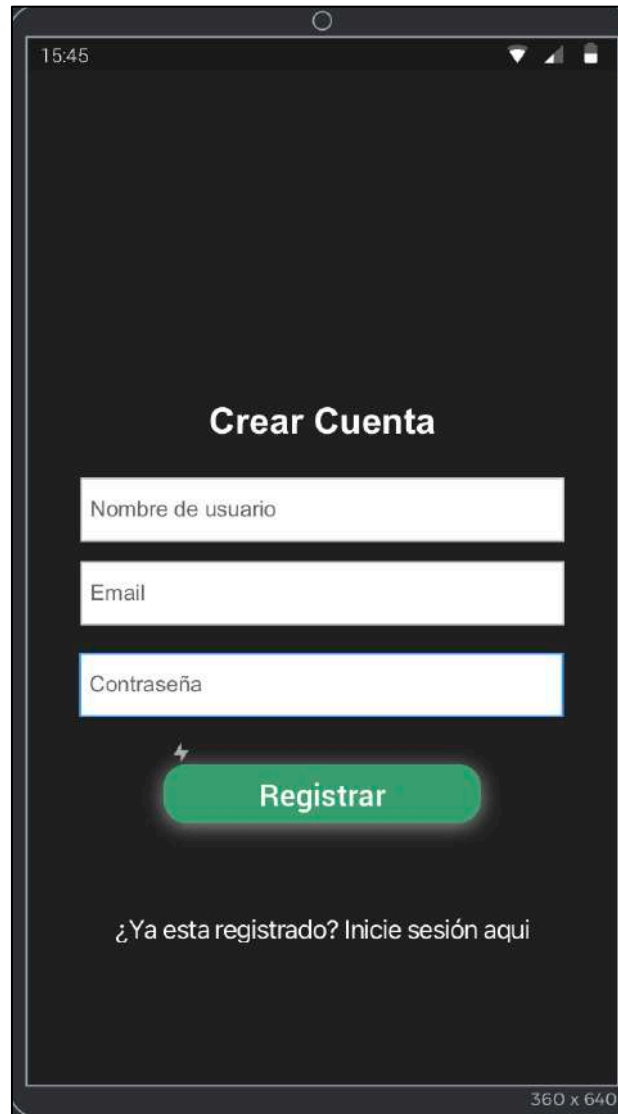


Figura 21 - Mockup de la pantalla de registro de usuarios a la aplicación móvil

C. Pantalla inicial de la aplicación móvil: desde la cual el usuario podrá observar los audiolibros añadidos y utilizar las diferentes acciones como ir a la configuración, buscar, y ver mi biblioteca



Figura 22 - Mockup de la pantalla inicial de la aplicación móvil

D. Pantalla para buscar audiolibros: desde la cual el usuario podrá buscar los audiolibros por título de audiolibro, nombre de autor y género

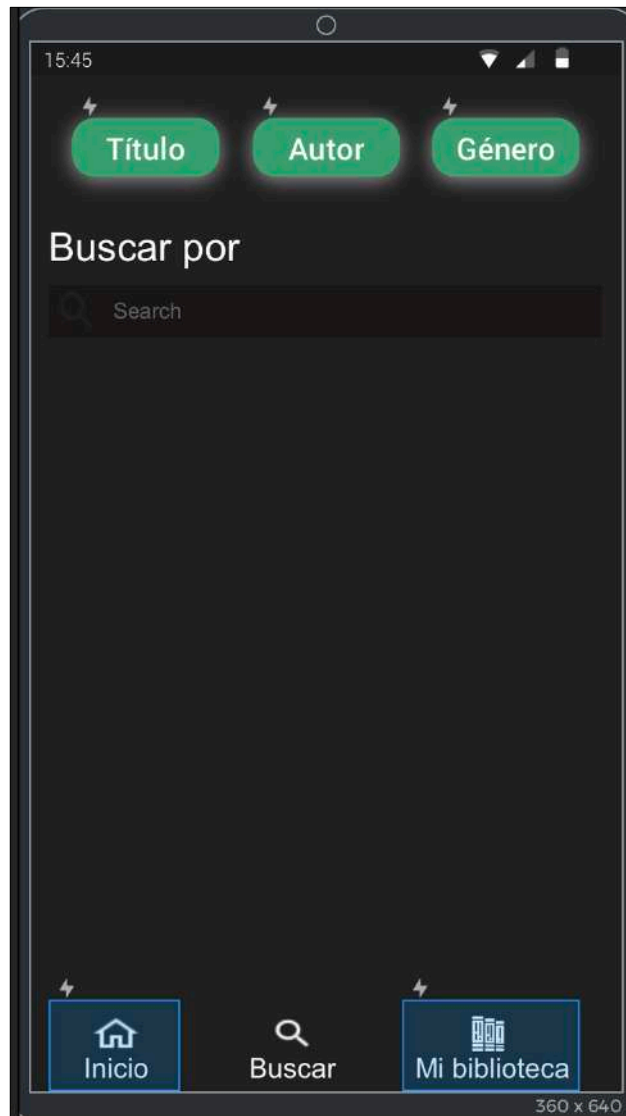


Figura 23 - Mockup de la pantalla para buscar audiolibros

- E. Pantalla para ver las listas de reproducción:** desde la cual el usuario podrá observar las listas de reproducción que haya creado, además de poder redirigirse a la pantalla para buscar listas de reproducción o crear una nueva lista de reproducción.



Figura 24 - Mockup de la pantalla para ver las listas de reproducción

F. Pantalla para crear una lista de reproducción: desde la cual el usuario podrá crear una nueva lista de reproducción.

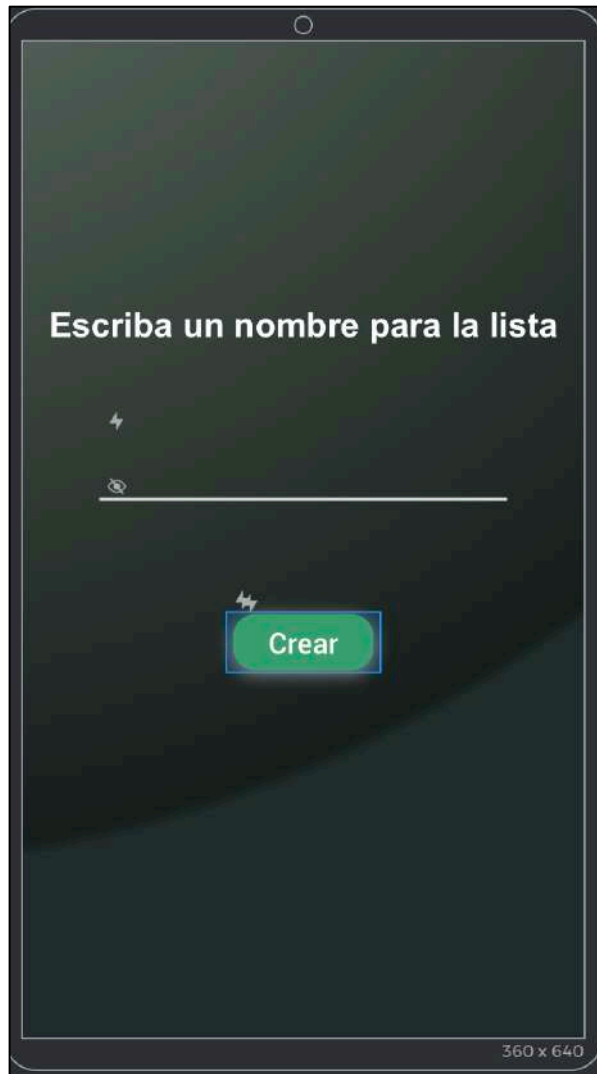


Figura 25 - Mockup de la pantalla para crear una lista de reproducción

G. Pantalla para ver información del audiolibro: desde la cual el usuario podrá observar la información del audiolibro seleccionado, además de redireccionar a la pantalla del reproductor.

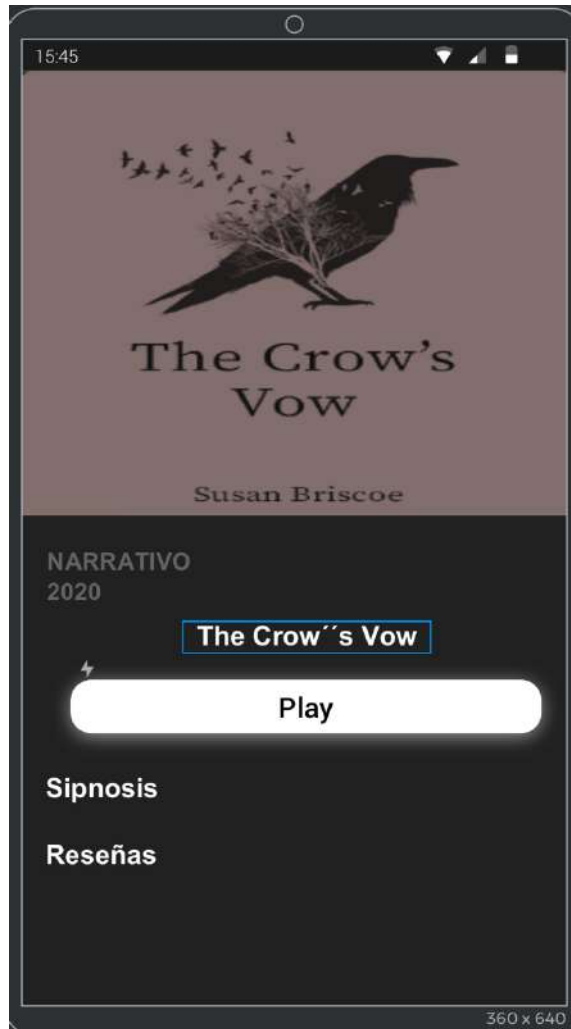


Figura 26 - Mockup de la pantalla para ver información del audiolibro

H. Pantalla del reproductor de audiolibros: desde la cual la aplicación móvil reproducirá el audiolibro seleccionado por el usuario, además de redireccionarse a otra pantalla en donde tendrá las opciones para añadir el audiolibro a una lista de reproducción o escribir una reseña.

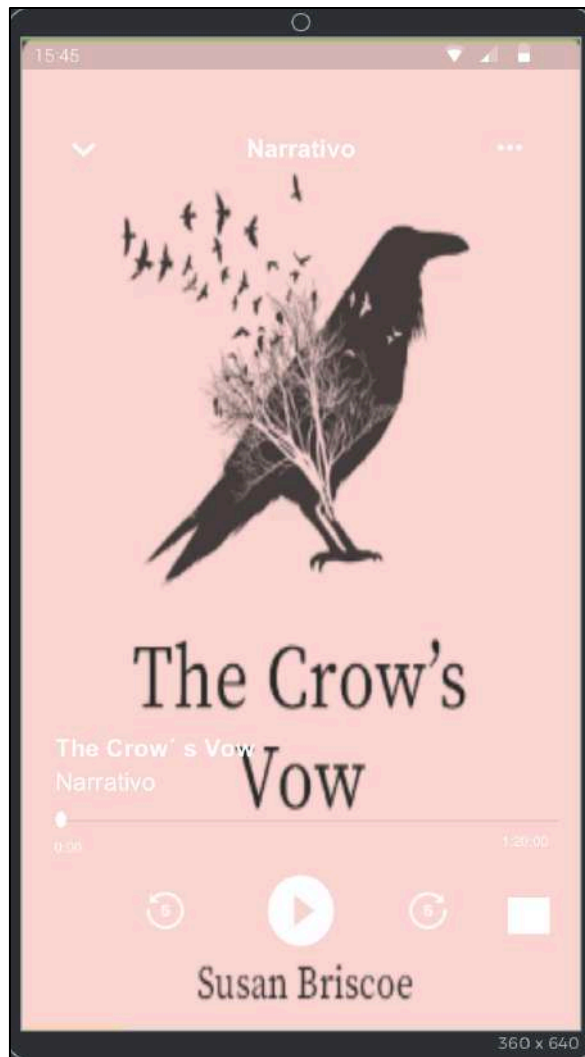


Figura 27 - Mockup de la pantalla del reproductor de audiolibros

- I. **Pantalla del menú de opciones del audiolibro:** desde la cual el usuario podrá elegir las opciones de añadir a lista de reproducción o reseñar el audiolibro.



Figura 28 - Mockup de la pantalla del menú de opciones del audiolibro

- J. Pantalla para añadir un audiolibro a una lista de reproducción:** desde la cual el usuario podrá agregar un audiolibro a una lista de reproducción a su elección, además de redireccionarse a la pantalla para crear una nueva lista de reproducción.



Figura 29 - Mockup de la pantalla para añadir un audiolibro a una lista de reproducción

K. Pantalla para reseñar un audiolibro: desde la cual el usuario podrá escribir una reseña sobre el audiolibro seleccionado.

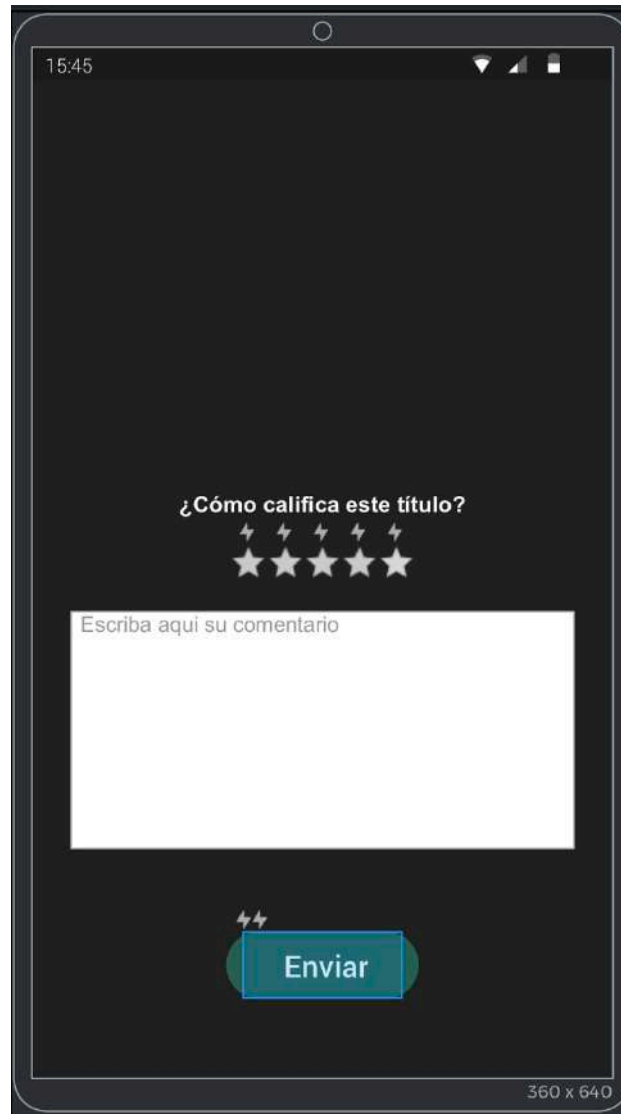


Figura 30 - Mockup de la pantalla para reseñar un audiolibro

- L. Pantalla de configuración:** desde la cual el usuario podrá ver su información personal, además de poder redireccionarse a la pantalla para editar sus datos.

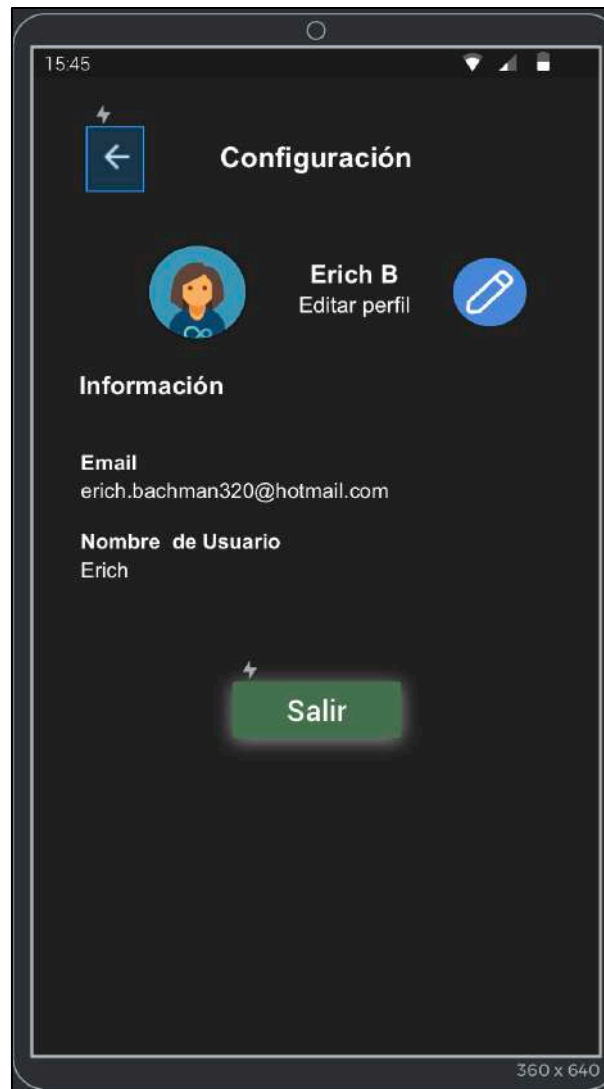


Figura 31 - Mockup de la pantalla de configuración

M. Pantalla para editar información del usuario: desde la cual el usuario podrá editar sus datos personales.

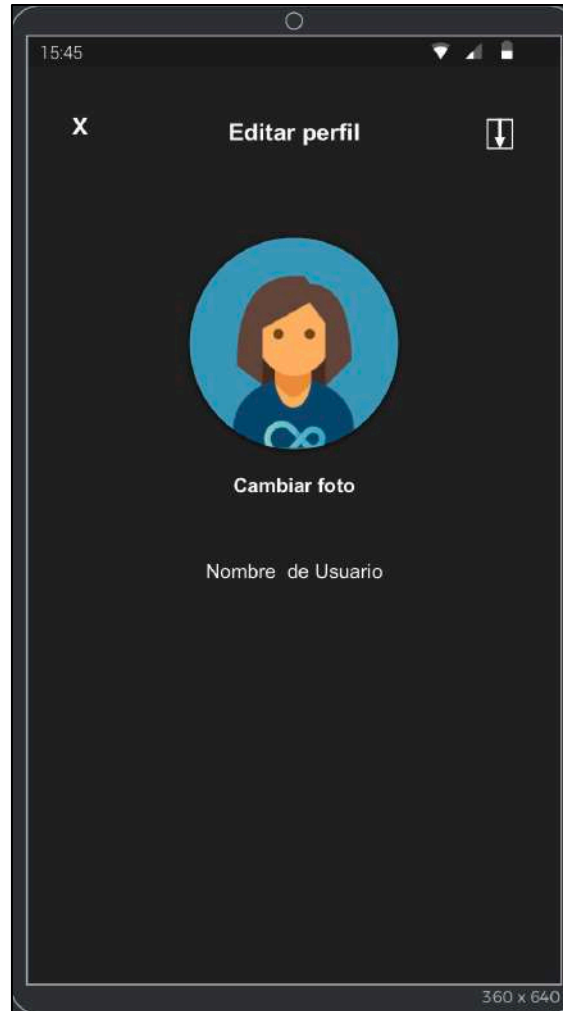


Figura 32 - Mockup de la pantalla para editar información de usuario desde la aplicación móvil

- **Sprint 0 Review**

Tras la finalización de este Sprint se logró obtener la arquitectura en el cual el equipo deberá basarse para empezar con el desarrollo del proyecto, y estar consciente del flujo de trabajo que tiene que seguir para cumplir con sus objetivos. De igual manera se logró obtener los mockups del aplicativo web como de la aplicación móvil, además del flujo que tendrá cada una de las aplicaciones cuando el usuario interactúe cuando las utilice.

- **Sprint 0 Retrospective**

- **¿Qué salió bien?**

Se logró diseñar la arquitectura de microservicios de manera correcta, definir las herramientas que se utilizaran para el desarrollo del backend y del frontend del proyecto y crear los repositorios Git para el control de versiones. De igual manera se logró diseñar los mockups de las aplicaciones de manera correcta, definir cuál será la navegación que tendrán las mismas al momento de ser utilizadas por los usuarios.

- **¿Qué se puede mejorar?**

Hay que familiarizarse (leer documentación) con las herramientas a utilizar ya que son nuevas para el equipo de trabajo. En cuanto a los mockups requieren de una paleta de colores específica para cuando se realice el desarrollo en las respectivas herramientas.

3.2.4.2. Sprint 1

- **Objetivo del Sprint**

Implementar el microservicio ***AudioBookCatalogMicroservice*** para probar desde la parte del backend las historias de usuario correspondientes.

- **Sprint 1 Planning**

En este sprint se planea implementar el microservicio ***“AudioBookCatalogMicroservice”*** con el framework de SpringBoot, que permitirá probar desde la parte del backend las historias de usuario relacionadas con: añadir, eliminar, actualizar un audiolibro; buscar por título de audiolibro, por nombre de autor o género y reseñar un audiolibro. Finalmente, las pruebas se realizarán con la herramienta Postman [34].

Sprint 1 Planning

Código	Nombre	Actividades
HU03 - 01	Añadir audiolibro al repositorio	Crear la base de datos en un Docker.
		Crear la entidad Audiolibro con sus respectivos atributos.
		Crear el método para añadir un nuevo audiolibro al repositorio.
		Crear el API que permita añadir un audiolibro.
HU03 - 02	Eliminar audiolibro del repositorio	Crear el método para eliminar un audiolibro.
		Crear el API que permita eliminar un audiolibro.
HU03 - 03	Editar la información de un audiolibro previamente añadido	Crear el método para editar un audiolibro.
		Crear el API que permita editar un audiolibro.
HU04 - 01	Buscar los audiolibros existentes por autor	Crear el método para buscar un audiolibro por autor.
		Crear el API que permita buscar un audiolibro por autor.
HU04 - 02	Buscar los audiolibros existentes por título de audiolibro	Crear el método para buscar un audiolibro por su título.
		Crear el API que permita buscar un audiolibro por su título.

HU04 - 03	Buscar los audiolibros existentes por género	Crear el método para buscar un audiolibro por género.
		Crear el API que permita buscar un audiolibro por género.
HU05 - 02	Reseñar un audiolibro	Crear la entidad Review con sus respectivos atributos.
		Crear el método para realizar una reseña a un audiolibro.
		Crear el API que permite realizar una reseña a un audiolibro.

Tabla 6 – Sprint 1 Planning

- **Ejecución del Sprint 2**

- **Preparación del entorno**

Primeramente, se creó una base de datos PostgreSQL [44] llamada “**catalog-microservice**” dentro del Docker [45], como lo podemos observar en la siguiente imagen:

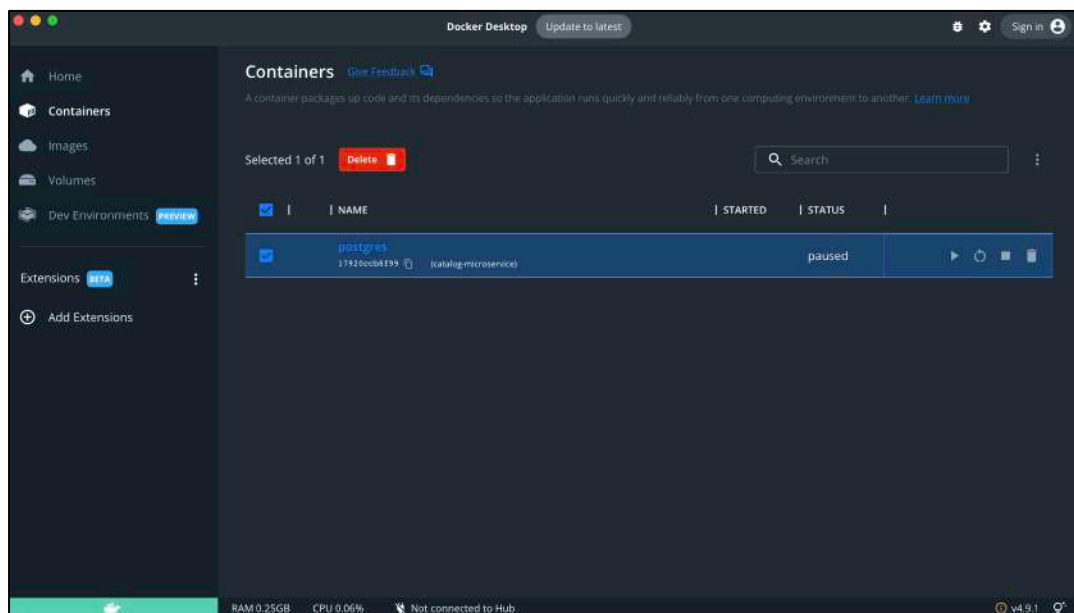


Figura 33 - Base catalog-microservice contenida en el Docker

- **Implementación del microservicio AudioBookCatalogMicroservice con SpringBoot**

Una vez que el Docker [45] contenga la base de datos, se procedió a crear un nuevo proyecto en Eclipse [29] con el framework SpringBoot [28] con el nombre “**AudioBookMicroservice**”, se crearon los siguientes paquetes:

A. com.thesis.audiobookcatalogmicroservice.audiobook: el cual contiene la clase **AudioBook** que será nuestra entidad principal la misma que posee los siguientes atributos:

```
private Integer idAudioBook;  
private String titleAudioBook;  
private String author;  
private String sipnosis;  
private String urlImage;  
private String urlAudio;  
private String gender;  
private String yearOfPublication;  
private Set<Review> reviews;  
private String userId;
```

Figura 34 - Atributos de la entidad AudioBook

En esta clase igualmente se encuentra el constructor y los métodos getters y setters para acceder a sus atributos.

B. com.thesis.catalogmicroservice.audiobook.review: el cual contiene la clase **Review** entidad que se relaciona con la clase **AudioBook**, la misma que posee los siguientes atributos:

```
private Integer idReview;  
private String userId;  
private String comment;  
private String rating;  
private String idAudioBook;
```

Figura 35 - Atributos de la entidad Review

- C. com.thesis.catalogmicroservice.repository:** el cual tiene dos clases (interfaces) con el nombre “AudioBookRepository” y “ReviewRepository” respectivamente, las mismas que poseen una extensión **JpaRepository** que contiene la API para las operaciones básicas de CRUD.
- D. com.thesis.catalogmicroservice.service:** el cual contiene la clase **AudioBookService** que será encargada de almacenar los servicios (métodos) para agregar un nuevo audiolibro, obtener los audiolibros agregados, eliminar, actualizar audiolibros; buscar por nombre de autor, título o género y reseñar un audiolibro. Podemos observar el método para buscar un audiolibro en el Anexo 3.
- E. com.thesis.catalogmicroservice.utils:** el cual contiene la clase **SearchType** de tipo enum, donde se declaró un conjunto de constantes con los tipos de búsqueda de audiolibros.
- F. com.thesis.catalogmicroservice.rest:** contiene la clase **AudioBookREST** con la anotación **@RestController** que permite manejar todas las APIs REST como las peticiones GET, POST, Delete y PUT. Esta clase llama a los métodos de la clase **AudioBookService**. En el Anexo 3 se puede observar el método POST para agregar un audiolibro al repositorio.

Al terminar con el desarrollo se procedió a construir la imagen Docker para ejecutar el microservicio para la cual se agregó el archivo **Dockerfile** con la siguiente configuración:

```
FROM openjdk:8-jdk-alpine
EXPOSE 8080
ADD /build/libs/CatalogMicroservice-0.0.1-SNAPSHOT.jar .
CMD ["java", "-jar" , "CatalogMicroservice-0.0.1-SNAPSHOT.jar"]
```

Figura 36 - Archivo Dockerfile del microservicio “AudioBookCatalogMicroservice”

Se ejecutó el siguiente comando para crear la imagen en el contenedor con el nombre **“audiobook-catalog-microservice”**:

```
docker build -f Dockerfile -t audiobook-catalog-microservice .
```

Figura 37 - Comando para construir la imagen "audiobook-catalog-microservice"

Para comprobar que el comando se ejecutó correctamente se abrió Docker Desktop donde podremos visualizar la imagen creada:

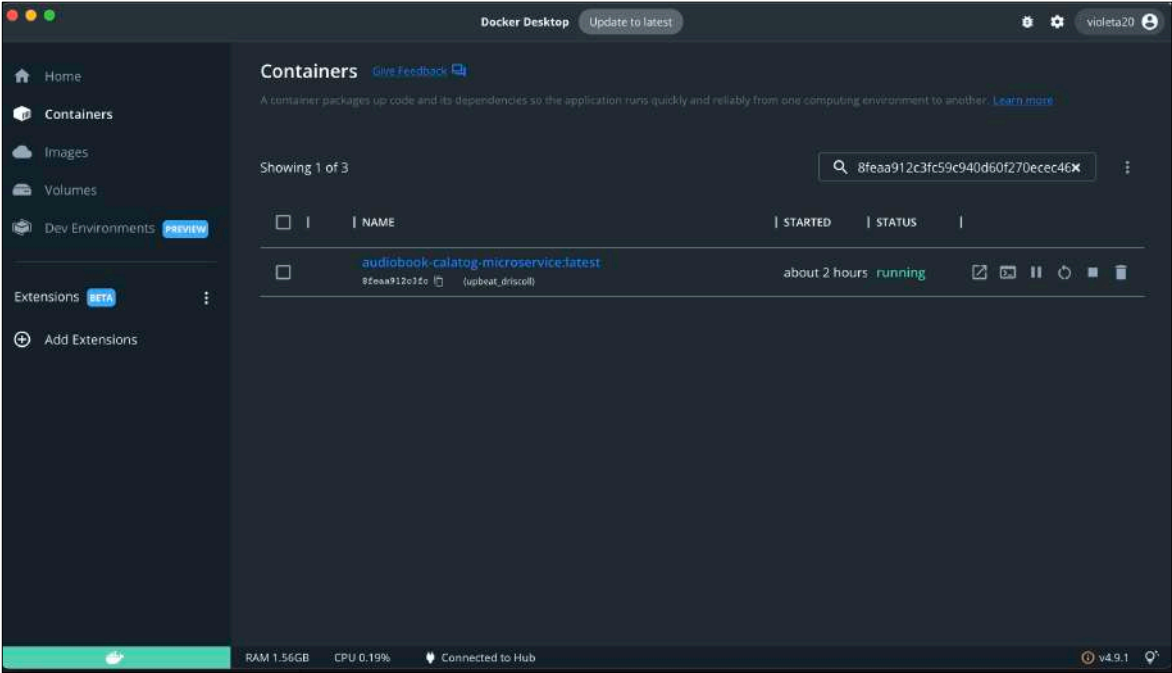


Figura 38 - Microservicio AudioBookCatalogMicroservice contenido en el Docker Desktop

- **Sprint 1 Review**

Se cumplió con el objetivo de crear el microservicio "AudioBookCatalogMicroservice" con su respectiva base de datos, además se realizó las siguientes historias de usuarios:

Historia de usuario	Cumplido
Se añadió un audiolibro al repositorio.	Sí
Se eliminó audiolibro del repositorio.	Sí
Se editó la información de un audiolibro previamente añadido.	Sí
Se buscó los audiolibros existentes por autor.	Sí

Se buscó los audiolibros existentes por título de audiolibro.	Sí
Se buscó los audiolibros existentes por género.	Sí
Se reseñó un audiolibro.	Sí

Tabla 7 – Sprint 1 Review

- **Sprint 1 Retrospective**

- **¿Qué salió bien?**

Se obtuvo las peticiones REST para las diferentes historias de usuario comprobando su funcionalidad con la herramienta POSTMAN.

- **¿Qué se puede mejorar?**

Para el siguiente sprint se espera integrar el microservicio “**AudioBookCatalogMicroservice**” para que el usuario pueda añadir, editar o eliminar audiolibros desde el aplicativo web.

3.2.4.3. Sprint 2

- **Objetivo**

Desarrollar el frontend del aplicativo web basado para realizar la autenticación de usuario de tipo autor y administrador.

- **Sprint 2 Planning**

En este sprint se planea desarrollar el frontend del aplicativo web basado en React [27], y a su vez realizar la autenticación de usuarios de tipo autor y administrador con Firebase Authentication [42]. Para cumplir con las siguientes historias de usuario:

Sprint 2 Planning

Código	Nombre	Actividades
HU01 - 01	Crear cuenta como usuario autor con email y contraseña	Instalar el paquete de firebase para poder utilizar Firebase Authencation y Firebase Storage.
		Realizar la configuración del proyecto de React con Firebase.
		Crear el método para que el usuario se registre al aplicativo web con su email y contraseña.
		Crear las validaciones respectivas para el formulario de registro.
		Diseñar la página para el registro de usuarios.
HU01 - 02	Crear cuenta como usuario autor con Google	Crear el método para que el usuario se registre con su cuenta de Google.
HU02 - 01	Autenticación usuario autor	Crear el método para la autenticación de usuarios al aplicativo web.
		Crear las validaciones respectivas para el formulario de autenticación.
		Diseñar la página para la autenticación de usuarios.
		Diseñar la página principal para el usuario autor.

HU02 - 02	Editar información personal desde el aplicativo web	Crear el método para el usuario pueda editar su información personal.
		Diseñar la página para editar su información.
HU02 - 05	Autenticación usuario administrador	Diseñar la página principal del usuario administrador.
HU07 - 01	Habilitar o deshabilitar la cuenta de un usuario	Crear una tabla con el listado de los usuarios registrado en el aplicativo web.
		Crear el método para deshabilitar o habilitar la cuenta de un usuario.

Tabla 8 – Sprint 2 Planning

- **Ejecución del Sprint 2**

- A. Creación del proyecto AudityWebFrontend**

Para crear el proyecto se utilizó la herramienta Visual Studio Code [30], en la cual se inició una nueva terminal y se ejecutó el comando:

```
npx create-react-app AudityWebFrontend
```

Figura 39 - Comando para la creación del proyecto en React [43]

Una vez que concluyó la creación del proyecto, se agregaron los siguientes directorios:

- **Context:** contiene el archivo *authContext.js* el cual es el encargado de la comunicación con Firebase [40] para la autenticación, registro de usuarios y carga y descarga de multimedia al Storage [43].
- **Pages:** se subdivide en los siguientes directorios:

- **authentication:** contiene los archivos del diseño de pantallas que se comunican con el servicio de Firebase [40] para la autenticación y registro de usuarios.
- **informationUser:** contiene los archivos del diseño de pantallas que se comunican con el archivo **authContext.js**.

En el Anexo 4, se muestra un ejemplo del archivo del diseño de la pantalla Login, teniendo en cuenta que el diseño es similar para las demás pantallas.

B. Integración de Firebase al proyecto

Se creó un nuevo proyecto desde la consola de Firebase [40] con el nombre de “**firebase-multimedia-storage**”

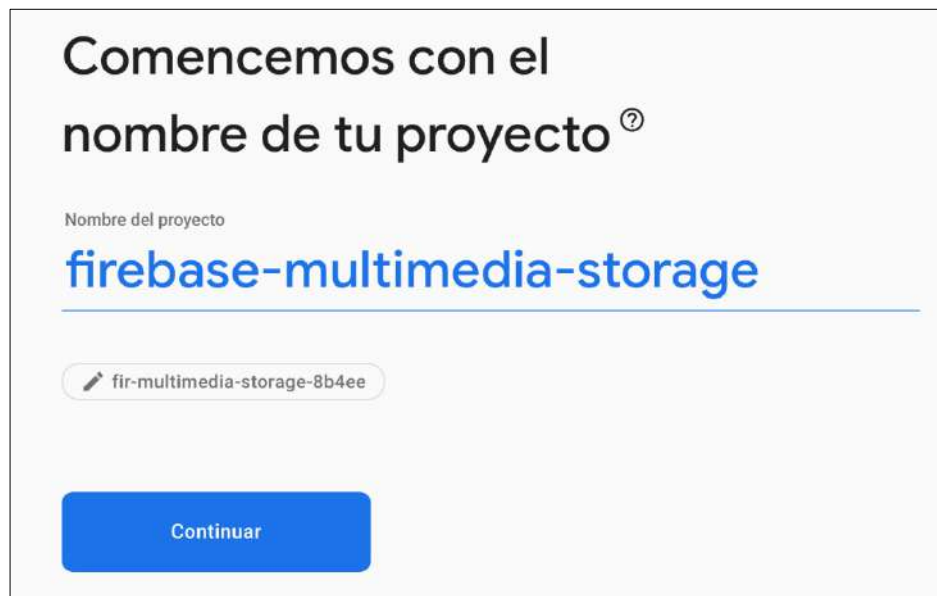


Figura 40 - Creación del proyecto en la consola de Firebase [40]

Una vez creado el proyecto se agregó la aplicación anteriormente creada en React [27] y en nueva terminal se ejecutó el comando para instalar Firebase [40] en nuestro proyecto, como lo muestra la siguiente imagen:

Sobrenombre de la app
react-multimedia-storage ✎

ID de la app ⓘ
1:1090037862392:web:5dba6283846359f90f100d

[Vincular a un sitio de Firebase Hosting](#)

Configuración del SDK

npm ⓘ CDN ⓘ Config ⓘ

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK:

```
$ npm install firebase
```

Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyB10XbCUdbzBdZpdL1c8jpEMsoFxtPidjk",
  authDomain: "fir-multimedia-storage.firebaseio.com",
  projectId: "fir-multimedia-storage",
  storageBucket: "fir-multimedia-storage.appspot.com",
  messagingSenderId: "1090037862392",
  appId: "1:1090037862392:web:5dba6283846359f90f100d",
  measurementId: "G-00ML9HQ5WS"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Nota: Esta opción utiliza el [SDK de JavaScript modular](#), que proporciona un tamaño reducido del SDK.

Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)

Figura 41 - Integración de Firebase con React [27]

Al concluir la instalación, se siguió las instrucciones que muestra la **Figura 59** para utilizar los productos de Firebase [27].

C. Implementación de Firebase con React

En el archivo **authContext.js** se encuentran servicios de Firebase [27] para realizar el registro y la autenticación de los usuarios al aplicativo web, en el Anexo 4 se puede observar con detalle cada uno de los métodos, a continuación, se detallan cada servicio utilizado:

- **createUserWithEmailAndPassword:** función que permite guardar la información del registro de un usuario en la base de Firebase “**Cloud Firestore**” [40]. Invocando el servicio que nos otorga cloud firestore (createUserWithEmailAndPassword).
- **signInWithEmailAndPassword:** función que permite que un usuario registrado pueda iniciar sesión con su email y contraseña. Invocando al servicio que nos otorga cloud firestore (signInWithEmailAndPassword).
- **loginWithGoogle:** función que permite que un usuario se registre e inicie sesión con su cuenta de Google, invocando al servicio de cloud firestore (signInWithPopup).

Posteriormente se realizó el diseño de las pantallas para iniciar sesión, registro de usuarios, pantalla principal para usuario autor, pantalla principal para usuario administrador y la pantalla para editar la información personal del usuario. Además, se configuró el archivo **App.js** el cual es responsable de almacenar las rutas de navegación que tendrá el aplicativo web.

- **Sprint 2 Review**

Al finalizar con la ejecución del sprint los objetivos se cumplieron correctamente, a continuación, se muestra un resumen de los criterios de aceptación de cada historia de usuario que se realizaron:

Código	Nombre
HU01 - 01	Crear cuenta como usuario autor con email y contraseña
Criterios de aceptación	
El email aceptó el formato xxx@xxx.com .	Cumplido Sí

La contraseña admitió como máximo 8 caracteres.	Sí
El nombre de usuario admitió como máximo 8 caracteres.	Sí
El formato de la imagen aceptó jpg, jpeg p png.	Sí

Tabla 9 - Criterios de aceptación de la historia de usuario HU01 – 01

Código	Nombre
HU01 - 02	Crear cuenta como usuario autor con Google
Criterio de aceptación	
Cumplido	
Cuando el usuario final seleccionó el icono de “Ingresa con Google” , ingresó la cuenta a registrar. Una vez que el usuario indicó la cuenta fue redireccionado a la página principal del aplicativo web.	
Sí	

Tabla 10 – Criterios de aceptación de la historia de usuario HU01 – 02

Código	Nombre
HU02 - 01	Autenticación usuario autor
Criterio de aceptación	
Cumplido	
El campo de ingreso correspondiente al email no aceptó vacíos.	
Sí	
El campo de ingreso correspondiente a la contraseña no aceptó vacíos.	
Sí	

Si las credenciales de acceso eran incorrectas (email o contraseña), y se seleccionó el ícono " Ingresar ", la aplicación mostró un mensaje de error.	Sí
Si las credenciales de acceso eran correctas (email y contraseña), y se seleccionó el ícono " Ingresar ", la aplicación permitió el acceso.	Sí

Tabla 11 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 01

Código	Nombre
HU02 - 02	Editar información personal desde el aplicativo web
Criterio de aceptación	
Cumplido	
Quando el usuario autor inició sesión en el aplicativo web y seleccionó el icono " Editar Perfil ", lo redireccionó a otra pantalla donde se mostró sus datos personales; los campos se habilitaron para su edición.	Sí
Quando el usuario autor terminó la edición, seleccionó el icono de " Guardar ".	Sí

Tabla 12 – Criterios de aceptación cumplidos de la historia de usuario HU02 – 02

Código	Nombre
HU02 - 05	Autenticación usuario administrador
Criterio de aceptación	
Cumplido	

El campo de ingreso correspondiente al email no aceptará vacíos	Sí
El campo de ingreso correspondiente a la contraseña no aceptará vacíos.	Sí
Si las credenciales de acceso son correctas (email y contraseña), y haya seleccionado el ícono " Ingresar ", entonces la aplicación le permitirá el acceso	Sí
Si las credenciales de acceso son incorrectas (email o contraseña), y haya seleccionado el ícono " Ingresar ", entonces la aplicación mostrará un mensaje de error	Sí

Tabla 13 – Criterios de aceptación cumplidos de la historia de usuario HU02 – 05

Código	Nombre	
HU07 - 01	Habilitar o deshabilitar la cuenta de un usuario	
Criterio de aceptación		Cumplido
El usuario deberá ingresar al aplicativo web con el usuario de tipo administrador		Sí
El sistema mostrará: <ul style="list-style-type: none"> ○ Nombre de usuario ○ Email ○ Rol ○ Acciones 		Sí

<p>Cuando el usuario administrador haya iniciado sesión, el sistema le mostrará los usuarios registrados con las opciones de habilitar o deshabilitar, el usuario deberá seleccionar la acción que desee realizar.</p>	<p>Sí</p>
--	-----------

Tabla 14 – Criterios de aceptación cumplidos de la historia de usuario HU07 – 01

- **Sprint 2 Retrospective**

- **¿Qué salió bien?**

El equipo de desarrollo pudo familiarizarse rápidamente con la librería React [27] para el desarrollo del aplicativo web.

- **¿Qué se puede mejorar?**

Se probaron la autenticación usando el SDK de Firebase Authentication, se puede también utilizar más proveedores como Facebook, Twitter, Microsoft, Apple lo cual permitiría a los usuarios tener más variedad en las formas de autenticación.

3.2.4.4. Sprint 3

- **Objetivo**

Conectar el microservicio **“AudioBookCatalogMicroservice”** con el frontend desarrollado para que el usuario autor pueda añadir, eliminar o actualizar un audiolibro en el repositorio desde el aplicativo web.

- **Sprint 3 Planning**

En este sprint se planea conectar el microservicio **“AudioBookCatalogMicroservice”** con el frontend desarrollado anteriormente en React [27] donde el usuario autor pueda añadir, eliminar o actualizar un audiolibro en el repositorio desde el aplicativo web.

Sprint 3 Planning

Código	Nombre	Actividades
HU03 - 01	Añadir audiolibro al repositorio	Diseñar la página para añadir un nuevo audiolibro.
		Obtener un listado de los audiolibros agregados para ser presentados en la página principal del usuario autor.
		Crear las validaciones respectivas para el formulario de añadir un audiolibro.
		Crear el método API para llamar al microservicio del backend encargado de añadir un audiolibro.
		Crear el método API para llamar al microservicio del backend encargado de obtener todos los audiolibros agregados.
HU03 - 02	Eliminar audiolibro del repositorio	Crear el método para eliminar un audiolibro del repositorio.
		Crear el método API para llamar al microservicio del backend encargado de eliminar un audiolibro.
HU03 - 03	Editar la información de un audiolibro previamente añadido	Diseñar la página para editar un audiolibro.
		Crear el método para editar un audiolibro.

		Crear el método API para llamar al microservicio den backend encargado de editar un audiolibro.
--	--	---

Tabla 15 – Sprint 3 Planning

- **Ejecución del Sprint 3**

En el proyecto anteriormente creado con React [27] se creó el directorio “**service**” el cual contendrá el archivo que permitirá la configuración de la conexión con el microservicio al aplicativo web y una carpeta “**audioBook**” dentro del mismo “**pages**”, el cual almacenará los archivos para el desarrollo de las pantallas de añadir, eliminar o editar un audiolibro.

- **Service:** esta carpeta contiene el archivo “**AudioBookService.js**” el mismo que permite la conexión con el microservicio “**AudioBookCatalogMicroservice**” creando las llamadas a las API que permitirán obtener todos los audiolibros agregados, añadir un audiolibro, editar un audiolibro anteriormente agregado, eliminar un audiolibro.
- **AudioBook:** esta carpeta tiene los siguientes archivos:
 - **AddAudioBook:** contiene el diseño de la pantalla desde la cual el usuario podrá añadir un nuevo libro cuando haya llenado el formulario correspondiente y seleccione la opción de guardar la misma que hace llamado a la función del backend de “AudioBookCatalogMicroservice”.
 - **AudioBookList:** contiene el diseño de la pantalla desde la cual el usuario podrá observar una lista de los audiolibros que haya agregado, a continuación, se muestra la función que permite obtener todos los audiolibros del usuario autor haya iniciado sesión en el aplicativo web. Este archivo también contiene la función que permite eliminar el audiolibro que el usuario autor seleccione.
 - **EditAudioBook:** contiene el diseño de la pantalla desde la cual el usuario podrá editar un audiolibro, a continuación, se muestra la

función que permite editar la información específica del audiolibro seleccionado por el usuario autor.

En el Anexo 5 se puede observar un detalle de lo mencionado anteriormente.

- **Sprint 3 Review**

Al finalizar con la ejecución del sprint los objetivos se cumplieron correctamente, a continuación, se muestra un resumen de los criterios de aceptación de cada historia de usuario que se realizaron:

Código	Nombre	
HU03 - 01	Añadir audiolibro al repositorio	
	Criterios de aceptación	Cumplido
	El usuario con el rol de autor ingresó al aplicativo web.	Sí
	Dado que el usuario autor ingresó al aplicativo web y seleccionó el icono de “Añadir audiolibro” , se redireccionó a la pantalla para agregar un audiolibro.	Sí
	El título aceptó un máximo de 50 caracteres admitiendo alfanuméricos.	Sí
	El nombre del autor aceptó un máximo de 20 caracteres.	Sí
	La sinopsis aceptó caracteres alfanuméricos.	Sí
	El usuario seleccionó un género para el audiolibro.	Sí
	La imagen aceptó el formato jpeg, jpg o png.	Sí

El archivo por subir aceptó el formato mp3.	Sí
El año de publicación aceptó alfanuméricos.	Sí

Tabla 16 – Criterios de aceptación de la historia de usuario HU03 - 01

Código	Nombre
HU03 - 02	Editar la información de un audiolibro previamente añadido
Criterios de aceptación	
	Cumplido
El usuario con el rol de autor ingresó al aplicativo web.	Sí
Cuando el usuario autor inició sesión el aplicativo web le mostró los audiolibros de su autoría.	Sí
Para editar la información del audiolibro el usuario seleccionó el icono “Editar” y el aplicativo web le mostró la información previamente registrada donde editó los siguientes campos: <ul style="list-style-type: none"> ○ Sinopsis ○ Imagen ○ Audio 	Sí

Tabla 17 – Criterios de aceptación de la historia de usuario HU03 - 02

Código	Nombre
HU03 - 03	Eliminar audiolibro del repositorio
Criterios de aceptación	
	Cumplido

El usuario con el rol de autor ingresó al aplicativo web.	Sí
Cuando el usuario autor inició sesión el aplicativo web mostró los audiolibros de su autoría.	Sí
Para eliminar el audiolibro el usuario autor seleccionó el icono “Eliminar” .	Sí

Tabla 18 – Criterios de aceptación de la historia de usuario HU03 - 03

- **Sprint 3 Retrospective**

- **¿Qué salió bien?**

El equipo de desarrollo logró desarrollar el frontend para probar el backend del microservicio **“AudioBookCatalogMicroservice”** creado anteriormente teniendo un mejor enfoque de lo que se desea que realice el aplicativo web.

- **¿Qué se puede mejorar?**

Para los siguientes sprints se espera integrar el microservicio **“AudioBookCatalogMicroservice”** con la aplicación móvil.

3.2.4.5. Sprint 4

- **Objetivo**

Implementar el microservicio **“UserPlayListMicroservice”** para probar desde la parte del backend sus historias de usuario correspondientes.

- **Sprint 4 Planning**

En este sprint se planea implementar el microservicio **“UserPlayListMicroservice”**, que permitirá probar desde la parte del backend las historias de usuario relacionadas con: crear una lista de

reproducción y agregar audiolibros a una lista de reproducción. Finalmente, las pruebas se realizarán con la herramienta Postman [14].

Sprint 4 Planning		
Código	Nombre	Actividades
HU06 - 01	Crear una lista de reproducción	Crear la base de datos en un Docker.
		Crear la entidad PlayList con sus respectivos atributos.
		Crear el método para crear una lista de reproducción.
		Crear el API que permita crear una lista de reproducción.
HU06 - 02	Agregar audiolibros a una lista de reproducción	Crear el método para agregar a una lista de reproducción.
		Crear el API que permita agregar una lista de reproducción.

Tabla 19 – Sprint 4 Planning

- **Ejecución del Sprint 4**

- **Preparación del entorno**

Primeramente, se creó una base de datos PostgreSQL [44] llamada “**userPlayList-microservice**” dentro del Docker [45], como lo podemos observar en la siguiente imagen:

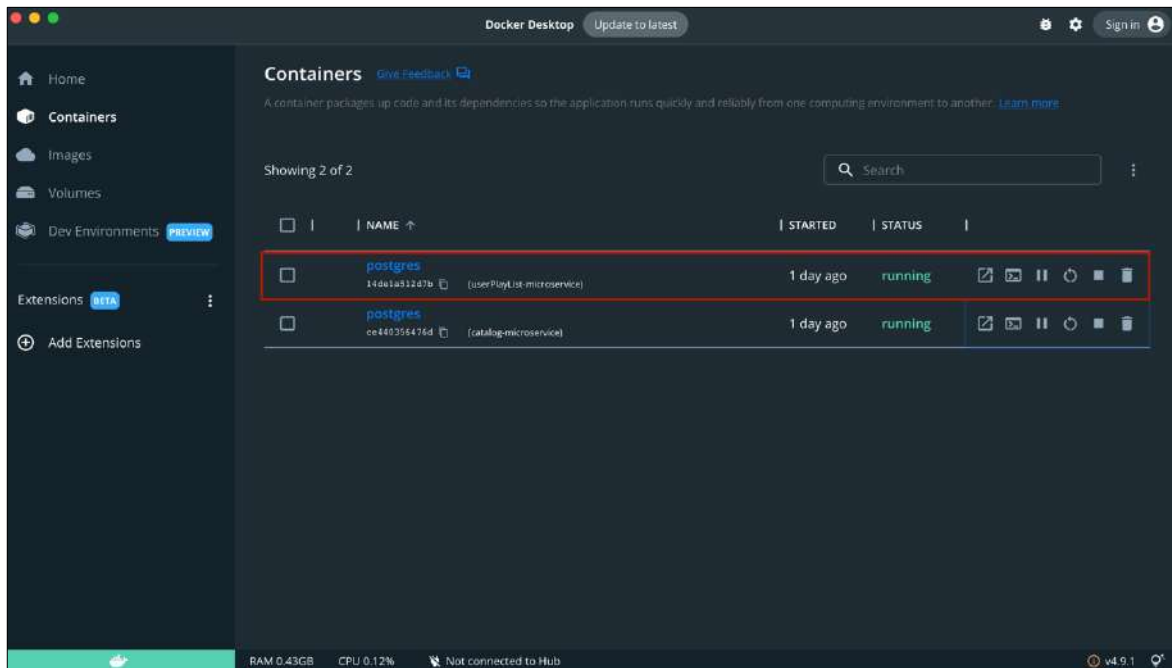


Figura 42 - Base userPlaylist-microservice contenida en el Docker[45]

- **Implementación del microservicio UserPlaylistMicroservice con SpringBoot**

Una vez que el Docker [45] contenga la base de datos, se procedió a crear un nuevo proyecto en Eclipse [29] con el framework SpringBoot [28] con el nombre “**UserPlayListsMicroservice**”, se crearon los siguientes paquetes:

A. com.thesis.userplaylistmicroservice.library: el cual contiene la clase **PlayList** que será nuestra entidad principal la misma que posee los siguientes atributos:

```
private Integer idPlayList;
private String namePlayList;
private String userId;
private Set<IdAudioBook> idAudioBooks;
```

Figura 43 - Atributos de la entidad PlayList

En esta clase igualmente se encuentra el constructor y los métodos getters y setters para acceder a sus atributos.

B. com.thesis.userplaylistmicroservice.library.audioBook: el cual contiene la clase **IdAudioBook** entidad que se relaciona con la clase **PlayList**, la misma que posee los siguientes atributos:

```
private Integer idAudioBookPlayList;  
private String idAudioBook;
```

Figura 44 - Atributos de la entidad *IdAudioBook*

C. com.thesis.userplaylistmicroservice.repository: el cual tiene dos clases (interfaces) con el nombre “PlayListRepository” y “IdAudioBookRepository” respectivamente, las mismas que poseen una extensión **JpaRepository** que contiene la API para las operaciones básicas de CRUD.

D. com.thesis.userplaylistmicroservice.service: el cual contiene la clase **PlayListService** que será encargada de almacenar los servicios (métodos) para crear una lista de reproducción y agregar audiolibros a la misma.

E. com.thesis.userplaylistmicroservice.utils: el cual contiene la clase **SearchType** de tipo enum, donde se declaró un conjunto de constantes con los tipos de búsqueda por título y usuario.

F. com.thesis.userplaylistmicroservice.rest: contiene la clase **PlayListREST** con la anotación **@RestController** que permite manejar todas las APIs REST como las peticiones GET, POST, Delete. Esta clase llama a los métodos de la clase **PlayListService**.

Al terminar con el desarrollo se procedió a construir la imagen Docker para ejecutar el microservicio para la cual se agregó el archivo **Dockerfile** con la siguiente configuración:

```
FROM openjdk:8-jdk-alpine  
EXPOSE 9090  
ADD /build/libs/UserPlayListsMicroservice-0.0.1-  
SNAPSHOT.jar .
```



```
CMD ["java", "-jar" , "UserPlayListsMicroservice-0.0.1-SNAPSHOT.jar"]
```

Figura 45 – Archivo Dockerfile del microservicio “UserPlayListMicroservice”

Se ejecutó el siguiente comando para crear la imagen en el contenedor con el nombre “**user-playlist-microservice**”:

```
docker build -f Dockerfile -t user-playlist-microservice .
```

Figura 46 – Comando para construir la imagen “audiobook-catalog-microservice”

Para comprobar que el comando se ejecutó correctamente se abrió Docker Desktop donde podremos visualizar la imagen creada:

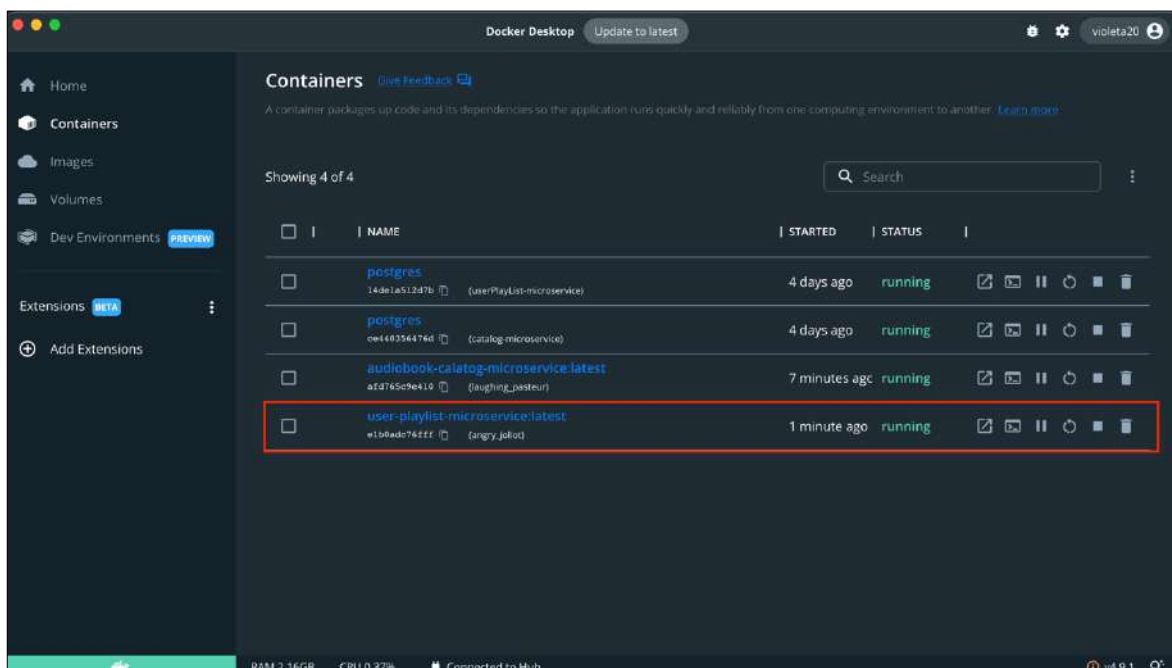


Figura 47 - Microservicio UserPlayListMicroservice contenido en el Docker Desktop

- **Sprint 4 Review**

Se cumplió con el objetivo de crear el microservicio “**UserPlayListMicroservice**” con su respectiva base de datos, además de realizar las siguientes historias de usuarios:

Historia de usuario	Cumplido
Se creó una lista de reproducción	Sí
Se agregó audiolibros a una lista de reproducción	Sí

Tabla 20 – Sprint 4 Review

- **Sprint 4 Retrospective**

- **¿Qué salió bien?**

Se obtuvo las peticiones REST para las diferentes historias de usuario comprobando su funcionalidad con la herramienta POSTMAN.

- **¿Qué se puede mejorar?**

Crear el aplicativo web y móvil con la finalidad de probar el microservicio “*UserPlayListMicroservice*” para comprobar su funcionalidad.

3.2.4.6. Sprint 5

- **Objetivo**

Desarrollar el frontend del aplicativo móvil para realizar la autenticación de usuarios de tipo final con Firebase Authentication.

- **Sprint 5 Planning**

En este sprint se planea desarrollar el frontend del aplicativo móvil basado en SwiftUI [27], y a su vez realizar la autenticación de usuarios de tipo final con Firebase Authentication [42].

Sprint 5 Planning

Código	Nombre	Actividades
HU01 - 03	Crear cuenta como usuario final con email y contraseña	Instalar el paquete de firebase para poder utilizar Firebase Authencation y Firebase Storage.
		Realizar la configuración del proyecto de SwiftUI con Firebase.
		Crear el método para que el usuario se registre a la aplicación móvil con su email y contraseña.
		Crear las validaciones respectivas para el formulario de registro.
		Diseñar la página para el registro de usuarios.
HU01 - 04	Crear cuenta como usuario final con Google	Crear el método para que el usuario se registre con su cuenta de Google.
HU02 - 03	Autenticación usuario final	Crear el método para la autenticación de usuarios al aplicativo web.
		Crear las validaciones respectivas para el formulario de autenticación.
		Diseñar la página para la autenticación de usuarios.
		Diseñar la página principal de la aplicación móvil.

HU02 - 04	Editar información personal desde el aplicativo web	Crear el método para el usuario pueda editar su información personal.
		Diseñar la página para editar la información del usuario final.

Tabla 21 – Sprint 5 Planning

- **Ejecución del Sprint 5**

A. Creación del proyecto AudityAppMobile

Para crear el proyecto se utilizó la herramienta XCode [30] donde se seleccionó la plantilla y el tipo de aplicación como se muestra en la siguiente imagen:

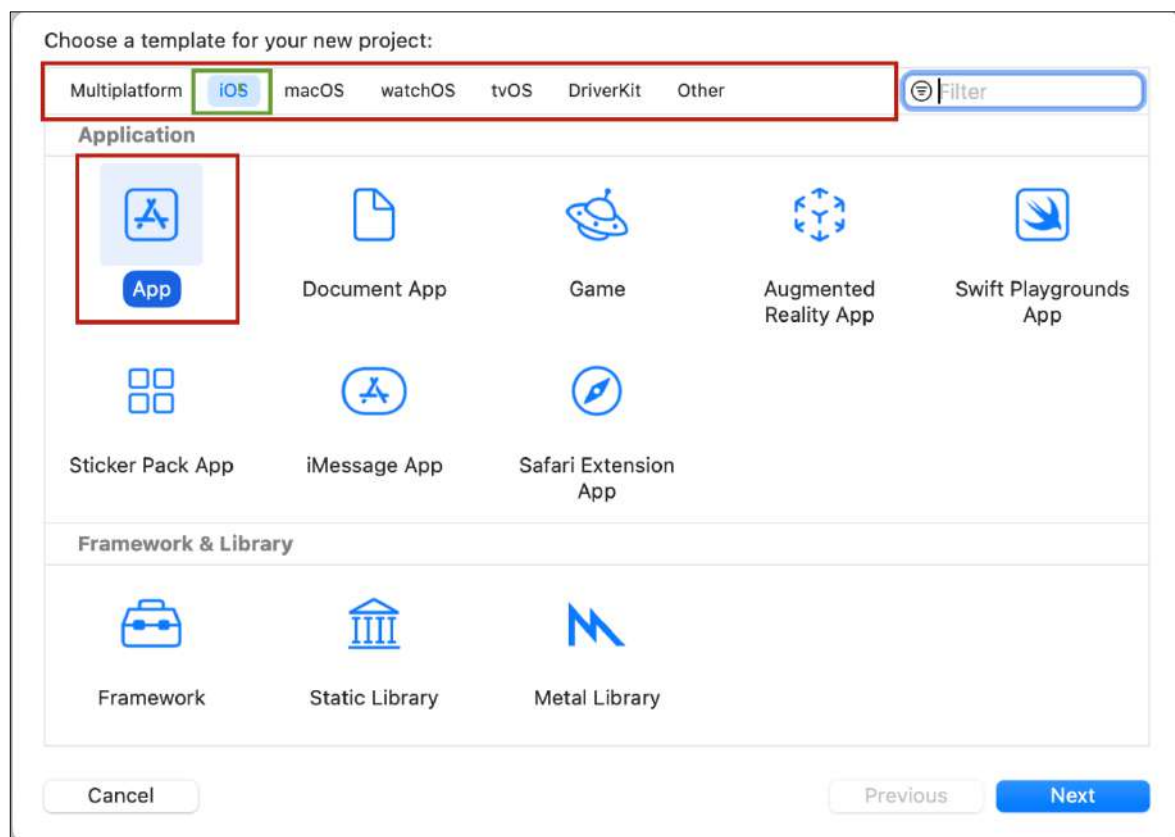


Figura 48 - Creación del proyecto en XCode [27]

A continuación, se le asignó un nombre al proyecto eligiendo el lenguaje y la interfaz en la cual se desarrollará la aplicación móvil:

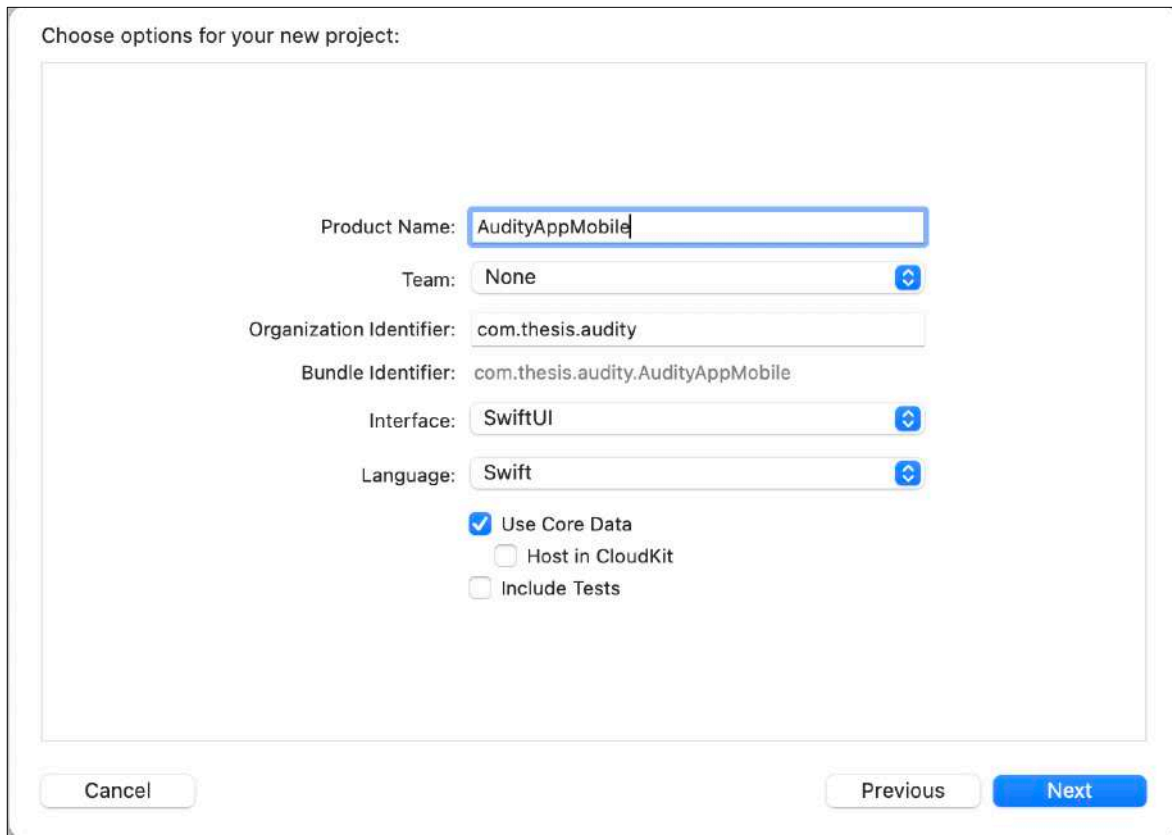


Figura 49 – Configuración del proyecto AudityAppMobile

Una vez que concluyó la creación del proyecto, se agregaron los siguientes directorios:

- **Model:** contiene los modelos que se tendrá durante todo el desarrollo de la aplicación móvil, como **“User”** con los atributos del autor final. Como se observa en la siguiente imagen:

```

import Foundation

struct User: Codable {

    var access: String = "true"
    var email: String = ""
    var password: String = ""
    var photo: String = ""
    var rol: String = "final"
    var userName: String = ""
}

```

Figura 50 – Archivo User

- **Firestore:** se subdivide en los siguientes directorios:
 - **Views:** contiene los archivos del diseño de pantallas que se comunican con el servicio de Firebase [40] para la autenticación y registro de usuarios.
 - **Auth:** contiene el archivo “**UserViewModel**” encargado de la creación y autenticación de usuarios mediante Firebase [40].

En el Anexo 6 se encuentra el diseño de la pantalla del registro de usuarios, teniendo en cuenta que el desarrollo es similar para las demás pantallas:

B. Implementación de Firebase con SwiftUI

En el archivo **UserViewModel** se encuentran servicios de Firebase [27] para realizar el registro y la autenticación de los usuarios a la aplicación móvil, en el Anexo 6 se encuentra un detalle de cada método realizado, a continuación, se detallan cada servicio utilizado:

- **createUser:** función que permite guardar la información del registro de un usuario en la base de Firebase “**Cloud Firestore**” [40]. Invocando el servicio que nos otorga cloud firestore (createUser).

- **signIn:** función que permite que un usuario registrado pueda iniciar sesión con su email y contraseña. Invocando al servicio que nos otorga cloud firestore (signIn).
- **loginWithGoogle:** función que permite que un usuario se registre e inicie sesión con su cuenta de Google, invocando al servicio de cloud firestore (GIDSignIn).

Posteriormente se realizó el diseño de las pantallas para iniciar sesión, registro de usuarios, pantalla principal de la aplicación móvil y la pantalla para editar la información personal del usuario final.

- **Sprint 5 Review**

Al finalizar con la ejecución del sprint los objetivos se cumplieron correctamente, a continuación, se muestra un resumen de los criterios de aceptación por cada historia de usuario realizada:

Código	Nombre	
HU01 - 03	Crear cuenta como usuario final con email y contraseña	
	Criterios de aceptación	Cumplido
	El email admitió el formato xxx@xxx.com	Sí
	La contraseña aceptó como máximo 8 caracteres	Sí
	El nombre de usuario aceptó como máximo 8 caracteres	Sí

Tabla 22 – Criterios de aceptación de la historia de usuario HU01 - 03

Código	Nombre	
HU01 - 04	Crear cuenta como usuario final con Google	

Criterio de aceptación	Cumplido
Cuando el usuario final seleccionó el icono de “Ingresa con Google” , indicó la cuenta a registrar. La aplicación móvil le redireccionó a la pantalla principal.	Sí

Tabla 23 – Criterios de aceptación de la historia de usuario HU01 - 04

Código	Nombre
HU02 - 03	Autenticación usuario final
Criterio de aceptación	Cumplido
El campo de ingreso correspondiente al email no aceptó vacíos.	Sí
El campo de ingreso correspondiente a la contraseña no aceptó vacíos.	Sí
Si las credenciales de acceso eran incorrectas (email o contraseña) y seleccionó el ícono “Ingresar” , la aplicación mostró un mensaje de error.	Sí
Si las credenciales de acceso eran correctas (email y contraseña) y seleccionó el ícono “Ingresar” , la aplicación le permitió el acceso.	Sí

Tabla 24 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 03

Código	Nombre
HU02 - 04	Editar información personal desde la aplicación móvil

Criterio de aceptación	Cumplido
Cuando el usuario final inició sesión en la aplicación móvil deberá y seleccionó el icono de “Configuración” , lo redireccionó a otra pantalla donde se presentó sus datos personales, el usuario final seleccionó el icono de “Editar” .	Sí
La aplicación móvil le mostró sus datos personales habilitados para su edición.	Sí
Cuando el usuario final terminó la edición, seleccionó el icono de “Guardar” .	Sí

Tabla 25 – Criterios de aceptación cumplidos de la historia de usuario HU02 - 04

- **Sprint 5 Retrospective**

- **¿Qué salió bien?**

El equipo de desarrollo pudo familiarizarse con la herramienta XCode para el desarrollo de la aplicación móvil y la integración de la librería de Firebase para lograr el registro y la autenticación de usuarios.

- **¿Qué se puede mejorar?**

En este sprint se probó la autenticación en el aplicativo móvil usando Firebase Authentication, sin embargo, el aplicativo móvil al ejecutarse en iOS, consideramos que una mejora futura sería añadir la autenticación con Apple.

3.2.4.7. Sprint 6

- **Objetivo**

Conectar los microservicios **“AudioBookCatalogMicroservice”** y **“UserPlayListMicroservice”** con la aplicación móvil para probar la funcionalidad final del sistema.

- **Sprint 6 Planning**

En este sprint se planea conectar los microservicios **“AudioBookCatalogMicroservice”** y **“UserPlayListMicroservice”** con el frontend desarrollado anteriormente en SwiftUI [27] donde el usuario autor pueda buscar audiolibros por nombre de autor, título de audiolibro o por género, reproducir audiolibros, crear listas de reproducción, agregar audiolibros a una lista de reproducción y reseñar un audiolibro desde la aplicación móvil.

Sprint 6 Planning		
Código	Nombre	Actividades
HU04 - 01	Buscar los audiolibros existentes por autor	Diseñar la pantalla para buscar audiolibros.
		Crear el método para buscar un audiolibro por autor.
		Crear el método API para llamar al microservicio “AudioBookCatalogMicroservice” encargado de buscar un audiolibro por su autor.
HU04 - 02	Buscar los audiolibros existentes por título de audiolibro.	Crear el método para buscar un audiolibro su título.

		<p>Crear el método API para llamar al microservicio “AudioBookCatalogMicroservice” encargado de buscar un audiolibro por su título.</p>
HU04 - 03	Buscar los audiolibros existentes por género	<p>Crear el método para buscar un audiolibro por género.</p>
		<p>Crear el método API para llamar al microservicio “AudioBookCatalogMicroservice” encargado de buscar un audiolibro por su género.</p>
HU05 - 01	Reproducir un audiolibro	<p>Diseñar la pantalla de reproducción de audiolibros.</p>
		<p>Crear el método para reproducir un audiolibro que el usuario escoja al azar.</p>
HU06 - 01	Crear una lista de reproducción	<p>Diseñar la pantalla para la creación de una lista de reproducción</p>
		<p>Crear el método para crear una lista de reproducción.</p>
		<p>Crear el método API para llamar al microservicio “UserPlayListMicroservice” encargado de crear una lista de reproducción.</p>
HU06 - 02	Agregar audiolibros a una lista de reproducción	<p>Crear el método para agregar un audiolibro a una lista de reproducción.</p>

		<p>Crear le método API para llamar al microservicio</p> <p>“UserPlayListMicroservice” encargado de agregar un audiolibro a una lista de reproducción.</p>
HU05 - 02	Reseñar un audiolibro	Diseñar la pantalla para reseñar un audiolibro.
		Crear el método para reseñar un audiolibro.
		<p>Crear le método API para llamar al microservicio</p> <p>“AudioBookCataloMicroservice” encargado de reseñar un audiolibro.</p>

Tabla 26 – Sprint 6 Planning

- **Ejecución del Sprint 6**

En el proyecto anteriormente creado con SwiftUI [27] se crearon los siguientes directorios:

- **Service:** el cual contiene los siguientes archivos:
 - **“AudioBookViewModel”:** permite la configuración de la conexión con el microservicio **“AudioBookCatalogMicroservice”**, en el Anexo 7 se puede observar la función para buscar un audiolibro por nombre de autor:
 - **“PlaylistViewModel”** permite la configuración de la conexión con el microservicio **“UserPlayListMicroservice”**, la siguiente imagen ilustra la función para crear una nueva lista de reproducción.
- **Shared:** contiene los archivos para el diseño de las diferentes pantallas que tendrá la aplicación móvil en el Anexo 7, se encuentra el desarrollo de la pantalla para reproducir un audiolibro recordando que la codificación es similar para las demás pantallas:

Posteriormente en el directorio **“Model”**, creado en el Sprint 5, se agregó el archivo **“PlayList”** con los atributos de una lista de reproducción.

- **Sprint 6 Review**

Al finalizar con la ejecución del sprint los objetivos se cumplieron correctamente, a continuación, se muestra un resumen de los criterios de aceptación por cada historia de usuario realizada:

Código	Nombre	
HU04 - 01	Buscar los audiolibros existentes por autor	
Criterios de aceptación		Cumplido
El mínimo de caracteres para realizar una búsqueda fue de 3 caracteres.		Sí
Se mostró: <ul style="list-style-type: none"> ○ Portada del audiolibro ○ Título del audiolibro ○ Autor ○ Fecha de publicación 		Sí
Cuando el usuario final inicio sesión en la aplicación móvil y seleccionó el icono de “Buscar” de la barra inferior, escribió el nombre del autor en la barra de búsqueda, la aplicación móvil mostró un listado de los audiolibros relacionados con la búsqueda.		Sí

Tabla 27 – Criterios de aceptación de la historia de usuario HU04 - 01

Código	Nombre
HU04 - 02	Buscar los audiolibros existentes por título de audiolibro
Criterios de aceptación	
	Cumplido
El mínimo de caracteres para realizar una búsqueda fue de 3 caracteres.	Sí
Se mostró: <ul style="list-style-type: none"> ○ Portada del audiolibro ○ Título del audiolibro ○ Autor ○ Fecha de publicación 	Sí
Cuando el usuario final inició sesión en la aplicación móvil y seleccionó el icono de “ Buscar ” de la barra inferior, escribió el nombre del autor en la barra de búsqueda, la aplicación móvil mostró un listado de los audiolibros relacionados con la búsqueda.	Sí

Tabla 28 – Criterios de aceptación de la historia de usuario HU04 - 02

Código	Nombre
HU04 - 03	Buscar los audiolibros existentes por género
Criterios de aceptación	
	Cumplido
El mínimo de caracteres para realizar una búsqueda fue de 3 caracteres.	Sí

<p>Se mostró:</p> <ul style="list-style-type: none"> ○ Portada del audiolibro ○ Título del audiolibro ○ Autor ○ Fecha de publicación 	Sí
<p>Cuando el usuario final haya iniciado sesión en la aplicación móvil y seleccionó el icono de “Buscar” de la barra inferior, escribió el nombre del autor en la barra de búsqueda, la aplicación móvil mostró un listado de los audiolibros relacionados con la búsqueda.</p>	Sí

Tabla 29 – Criterios de aceptación cumplidos de la historia de usuario HU04 - 03

Código	Nombre
HU05 - 01	Reproducir un audiolibro
Criterio de aceptación	Cumplido
<p>Cuando el usuario final inició sesión en la aplicación móvil y escogió un audiolibro fue redireccionado a una pantalla donde seleccionó el icono de “Play”, para reproducir el audiolibro.</p>	Sí

Tabla 30 – Criterios de aceptación cumplidos de la historia de usuario HU05 - 01

Código	Nombre
HU06 - 01	Crear una lista de reproducción
Criterio de aceptación	Cumplido

<p>Cuando el usuario final inició sesión en la aplicación móvil y seleccionó el icono “Mi biblioteca”, de la barra inferior, y posteriormente el icono de “+”, la aplicación móvil le redireccionó a la pantalla para crear una lista de reproducción.</p>	Sí
<p>El usuario llenó los siguientes campos:</p> <ul style="list-style-type: none"> ○ Nombre de la lista <ul style="list-style-type: none"> ▪ El campo admitió un máximo de 15 caracteres. ▪ El campo aceptó caracteres alfanuméricos y especiales. 	Sí
<p>Una vez que el usuario final haya llenó el campo correspondiente seleccionó el icono “Crear”.</p>	Sí

Tabla 31 – Criterios de aceptación cumplidos de la historia de usuario HU06 - 01

Código	Nombre						
HU06 - 02	Agregar audiolibros a una lista de reproducción						
	<table border="1"> <thead> <tr> <th>Criterio de aceptación</th> <th>Cumplido</th> </tr> </thead> <tbody> <tr> <td> <p>Cuando el usuario final inició sesión en la aplicación móvil y escogió un audiolibro, posteriormente seleccionó el icono de los tres puntos, la aplicación móvil lo redireccionó a otra pantalla en donde eligió la opción “Agregar a lista de reproducción”, optando por cualquier lista creada anteriormente.</p> </td> <td>Sí</td> </tr> <tr> <td> <p>El audiolibro fue añadido en la lista de reproducción seleccionada.</p> </td> <td>Sí</td> </tr> </tbody> </table>	Criterio de aceptación	Cumplido	<p>Cuando el usuario final inició sesión en la aplicación móvil y escogió un audiolibro, posteriormente seleccionó el icono de los tres puntos, la aplicación móvil lo redireccionó a otra pantalla en donde eligió la opción “Agregar a lista de reproducción”, optando por cualquier lista creada anteriormente.</p>	Sí	<p>El audiolibro fue añadido en la lista de reproducción seleccionada.</p>	Sí
Criterio de aceptación	Cumplido						
<p>Cuando el usuario final inició sesión en la aplicación móvil y escogió un audiolibro, posteriormente seleccionó el icono de los tres puntos, la aplicación móvil lo redireccionó a otra pantalla en donde eligió la opción “Agregar a lista de reproducción”, optando por cualquier lista creada anteriormente.</p>	Sí						
<p>El audiolibro fue añadido en la lista de reproducción seleccionada.</p>	Sí						

La aplicación móvil no permitió agregar el mismo audiolibro a una misma lista de reproducción.	Sí
--	----

Tabla 32 – Criterios de aceptación cumplidos de la historia de usuario HU06 - 02

Código	Nombre	
HU05 - 02	Reseñar un audiolibro	
Criterio de aceptación		Cumplido
<p>Cuando el usuario final inició sesión en la aplicación móvil escogió un audiolibro, posteriormente seleccionó el icono de los tres puntos, la aplicación móvil lo redireccionó a otra pantalla en donde eligió la opción “Reseña”, donde llenó los siguientes campos:</p> <ul style="list-style-type: none"> ○ Puntuación <ul style="list-style-type: none"> ▪ La puntuación estuvo representada por estrellas en un rango de una a cinco estrellas. ○ Comentario <ul style="list-style-type: none"> ▪ El comentario aceptó una longitud máxima de 200 caracteres ▪ El campo admitió caracteres alfanuméricos y especiales. 		Sí
<p>Cuando el usuario final concluyó seleccionó el icono “Enviar”, para guardar la reseña del audiolibro.</p>		Sí

Tabla 33 – Criterios de aceptación cumplidos de la historia de usuario HU05 – 02

- **Sprint 6 Retrospective**
 - ¿Qué salió bien?

El uso del patrón MVVM [10] facilitó el desarrollo final de la aplicación móvil ya que nos permitió reutilizar componentes en diferentes partes del desarrollo obteniendo un código limpio, reutilizable y mantenible.

- **¿Qué se puede mejorar?**

Como mejora futura se puede crear más microservicios que potencien tanto la aplicación web como la móvil, de tal forma que se añadan más funcionalidad importante al usuario. Por ejemplo, en el desarrollo de los microservicios notamos que se puede crear a futuro un microservicio que permita a un usuario suscribirse al contenido de un autor.

CAPÍTULO 4 – ANÁLISIS DE RESULTADOS

Tras completar la ejecución del Sprint 1 al 6, se realizaron pruebas funcionales [25], como las pruebas de aceptación [25] donde se verificó que cada sprint cumpla con los objetivos esperados, además se realizaron pruebas no funcionales [26] haciendo uso de diferentes herramientas para la realización de pruebas de rendimiento [26] y de carga [26], a continuación, se detalla cada una de las pruebas realizadas al finalizar cada Sprint:

4.1. Ejecución de pruebas del Sprint 1

- **Pruebas funcionales del Sprint 1**

Una vez que se concluyó la programación del microservicio **“AudioBookCatalogMicroservice”** se realizaron las pruebas correspondientes para verificar que las peticiones cumplan con las historias de usuario a continuación se detalla las peticiones realizadas:

A. Añadir audiolibro: permite agregar un nuevo audiolibro en la base de datos, se realizó una petición POST que recibe un objeto JSON [14] de tipo audiolibro con los atributos correspondientes al audiolibro.

MÉTODO	POST
URL	http://localhost:8080/api/audioBook/addAudioBook
BODY	<pre>{ "titleAudioBook": "Violeta", "author": "Isabel Allende", "sipnosis": "novela", "urlImage": "", "urlAudio": "", "gender": "Novela", "yearOfPublication": "2019" }</pre>
RESPONSE	

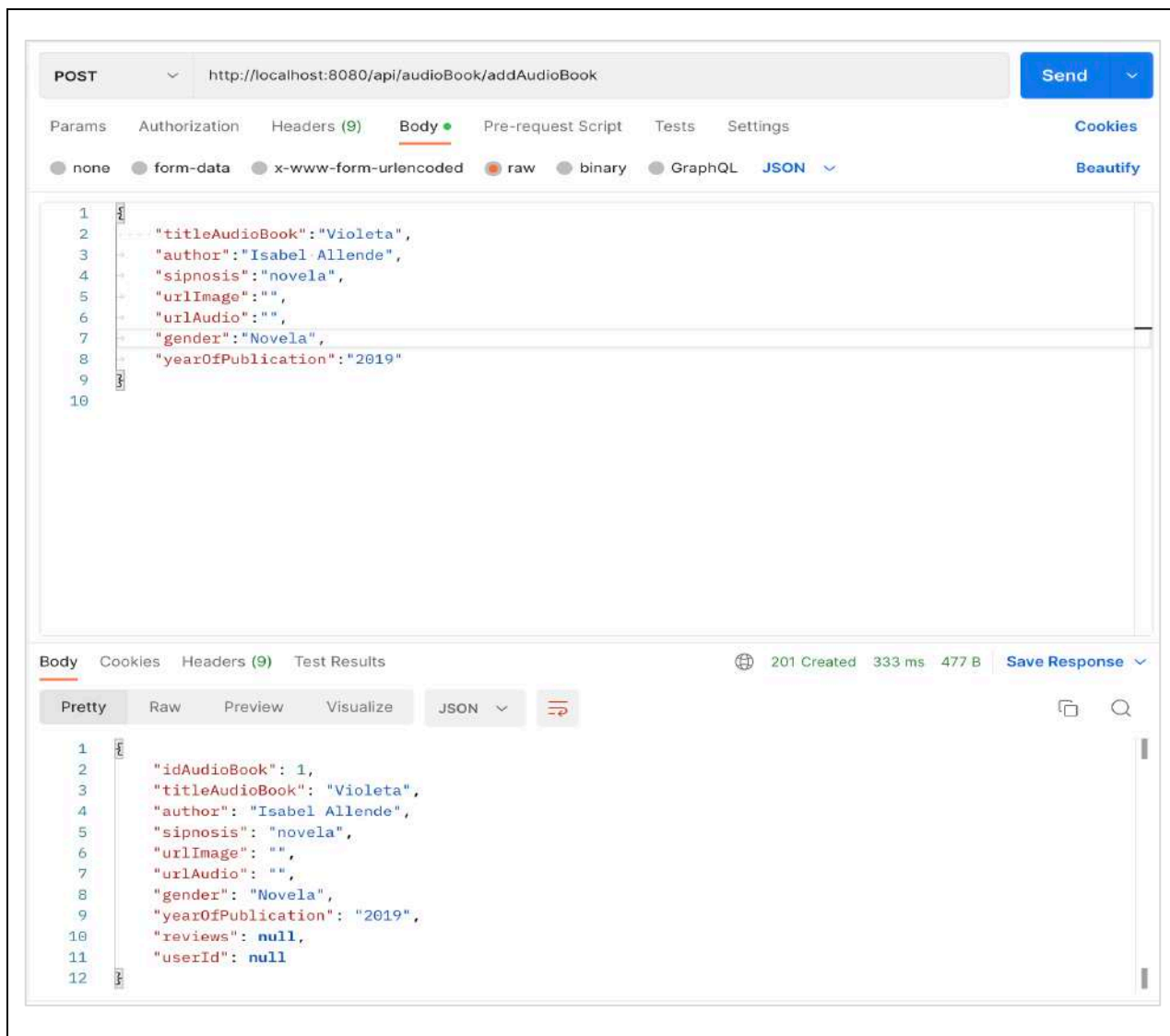


Tabla 34 - Petición POST para añadir un audiolibro

Como se observa en la **Tabla 34** el libro fue agregado exitosamente.

B. Actualizar audiolibro: permite actualizar la información de un audiolibro, se realiza una petición PUT que recibe un objeto de tipo audiolibro.

MÉTODO	PUT
URL	http://localhost:8080/api/audioBook/updateAudioBook
BODY	<pre> { "idAudioBook": 1, "titleAudioBook": "Violeta", "author": "Isabel Allende", </pre>

```
"sipnosis": "La épica y emocionante historia de una
mujer cuya vida abarca los momentos históricos más
relevantes del siglo XX",
"urlImage":
"https://www.google.com/imgres?imgurl=https%3A%2F%2Ffoli
teraria.com%2Fwpcontent%2Fuploads%2F2021%2F06%2FVioleta.jp
g&imgrefurl=https%3A%2F%2Ffoliteraria.com%2F2021%2F06%2F
25%2Fvioleta",
"urlAudio":
"https://www.google.com/imgres?imgurl=https%3A%2F%2Ffoli
teraria.com%2Fwpcontent%2Fuploads%2F2021%2F06%2FVioleta.jp
g&imgrefurl=https%3A%2F%2Ffoliteraria.com%2F2",
"gender": "Romance",
"yearOfPublication": "2020"
}
```

RESPONSE

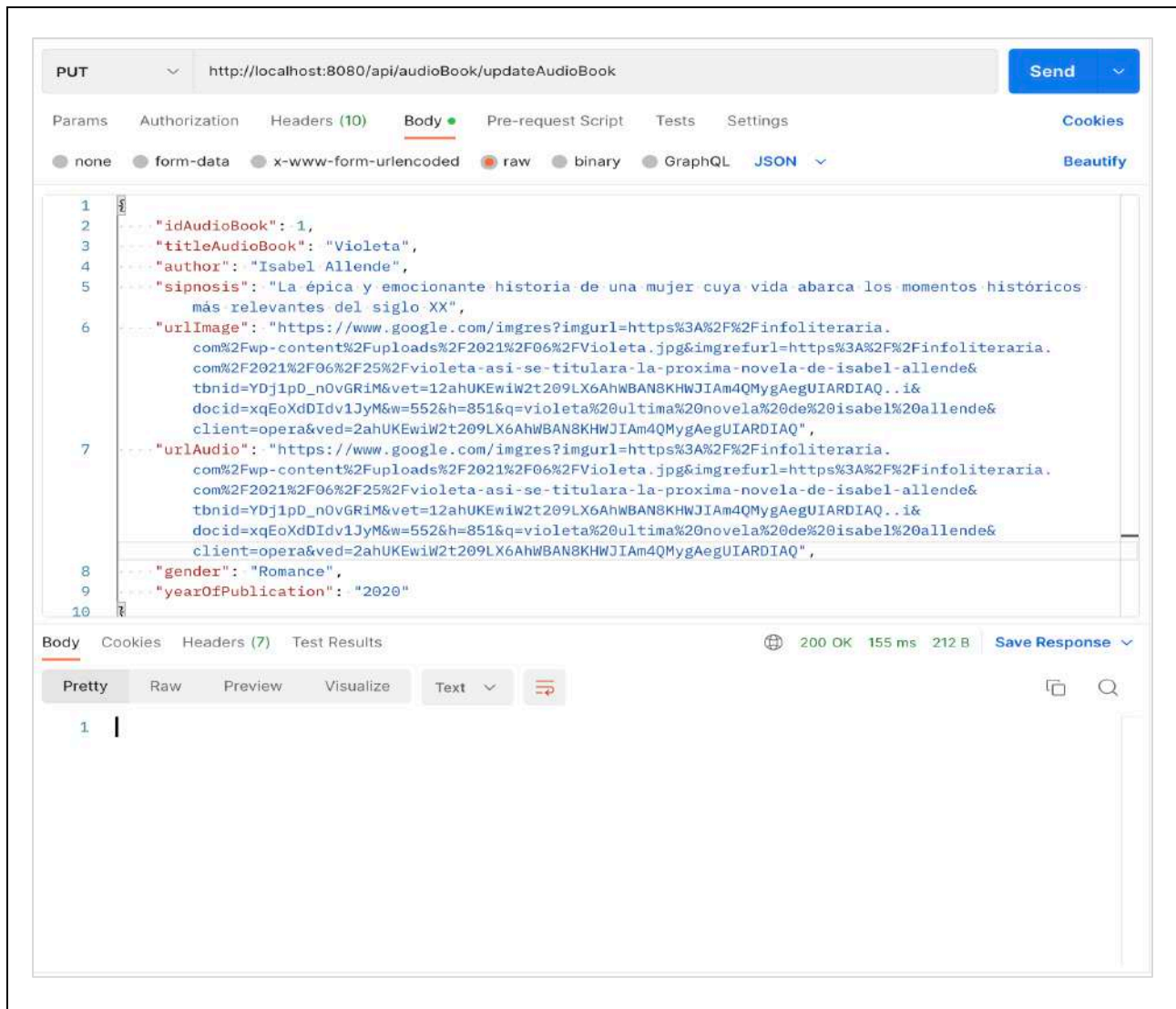


Tabla 35 - Petición PUT para actualizar un audiolibro

Para comprobar que la actualización fue exitosa, se realizó una petición GET para obtener la lista de todos los audiolibros.

MÉTODO	GET
URL	http://localhost:8080/api/audioBook/getAllAudioBooks
RESPONSE	

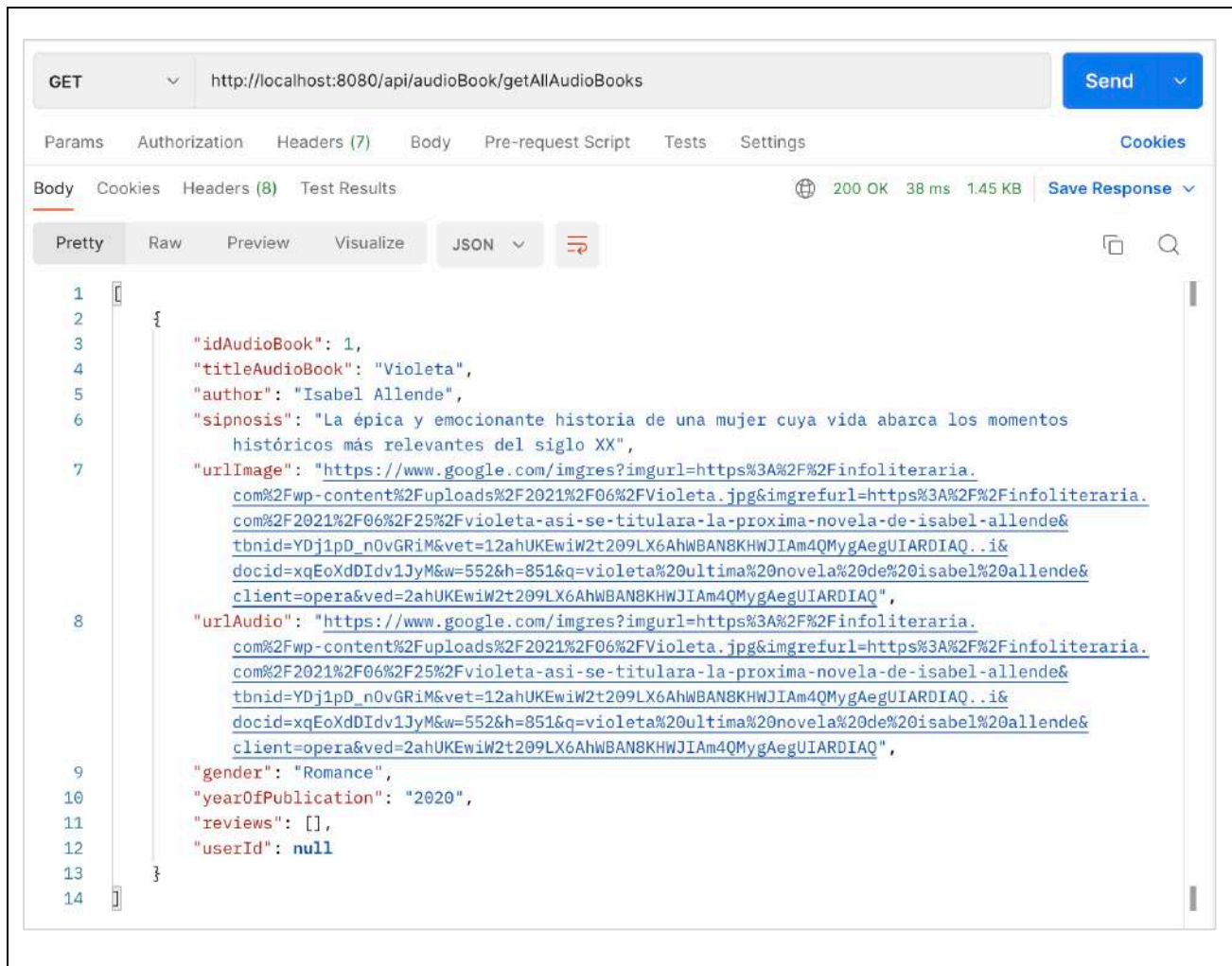


Tabla 36 - Petición GET para obtener todos los audiolibros agregados

Como se observa en la **Tabla 36** la actualización se realizó de manera exitosa.

C. Buscar por nombre de autor: permite buscar un audiolibro por el nombre del autor, se realizó una petición GET que recibe dos variables:

- Tipo de búsqueda “**SEARCH_PER_AUTHOR**”
- Nombre del autor

MÉTODO	GET
URL	http://localhost:8080/api/audioBook/searchByAuthor/SEARCH_PER_AUTHOR/Isabel

VARIABLES

- SEARCH_PER_AUTHOR
- Isabel

Response

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/api/audioBook/searchByAuthor/SEARCH_PER_AUTHOR/Isabel
- Status:** 200 OK, 44 ms, 1.45 KB
- Response Format:** JSON
- Response Body (Pretty):**

```

1  {
2
3    "idAudioBook": 1,
4    "titleAudioBook": "Violeta",
5    "author": "Isabel Allende",
6    "sipnosis": "La épica y emocionante historia de una mujer cuya vida abarca los momentos
7      históricos más relevantes del siglo XX",
8    "urlImage": "https://www.google.com/imgres?imgurl=https%3A%2F%2Finfofiteraria.
9      com%2Fwp-content%2Fuploads%2F2021%2F06%2FVioleta.jpg&imgrefurl=https%3A%2F%2Finfofiteraria.
10     com%2F2021%2F06%2F25%2Fvioleta-asi-se-titulara-la-proxima-novela-de-isabel-allende&
11     tbnid=YDjlpD_n0vGRiM&vet=12ahUKEwiW2t209LX6AhWBAN8KHWJIam4QMygAegUIARDIAQ..i&
12     docid=xqEoXdDIdv1JyM&w=552&h=851&q=violeta%20ultima%20novela%20de%20isabel%20allende&
13     client=opera&ved=2ahUKEwiW2t209LX6AhWBAN8KHWJIam4QMygAegUIARDIAQ",
14    "urlAudio": "https://www.google.com/imgres?imgurl=https%3A%2F%2Finfofiteraria.
15      com%2Fwp-content%2Fuploads%2F2021%2F06%2FVioleta.jpg&imgrefurl=https%3A%2F%2Finfofiteraria.
16     com%2F2021%2F06%2F25%2Fvioleta-asi-se-titulara-la-proxima-novela-de-isabel-allende&
17     tbnid=YDjlpD_n0vGRiM&vet=12ahUKEwiW2t209LX6AhWBAN8KHWJIam4QMygAegUIARDIAQ..i&
18     docid=xqEoXdDIdv1JyM&w=552&h=851&q=violeta%20ultima%20novela%20de%20isabel%20allende&
19     client=opera&ved=2ahUKEwiW2t209LX6AhWBAN8KHWJIam4QMygAegUIARDIAQ",
20    "gender": "Romance",
21    "yearOfPublication": "2020",
22    "reviews": [],
23    "userId": null
24  }
```

Tabla 37 - Petición GET para buscar audiolibros por nombre de autor

D. Buscar por título de audiolibro: permite buscar un audiolibro por su título, se realizó una petición GET que recibe dos variables:

- Tipo de búsqueda en este caso será “**SEARCH_PER_TITLE**”
- Título del audiolibro

MÉTODO

GET

URL	http://localhost:8080/api/audioBook/searchByAuthor/SEARCH_PER_TITL E/violeta
VARIABLES	<ul style="list-style-type: none"> ○ SEARCH_PER_TITLE ○ Violeta

RESPONSE

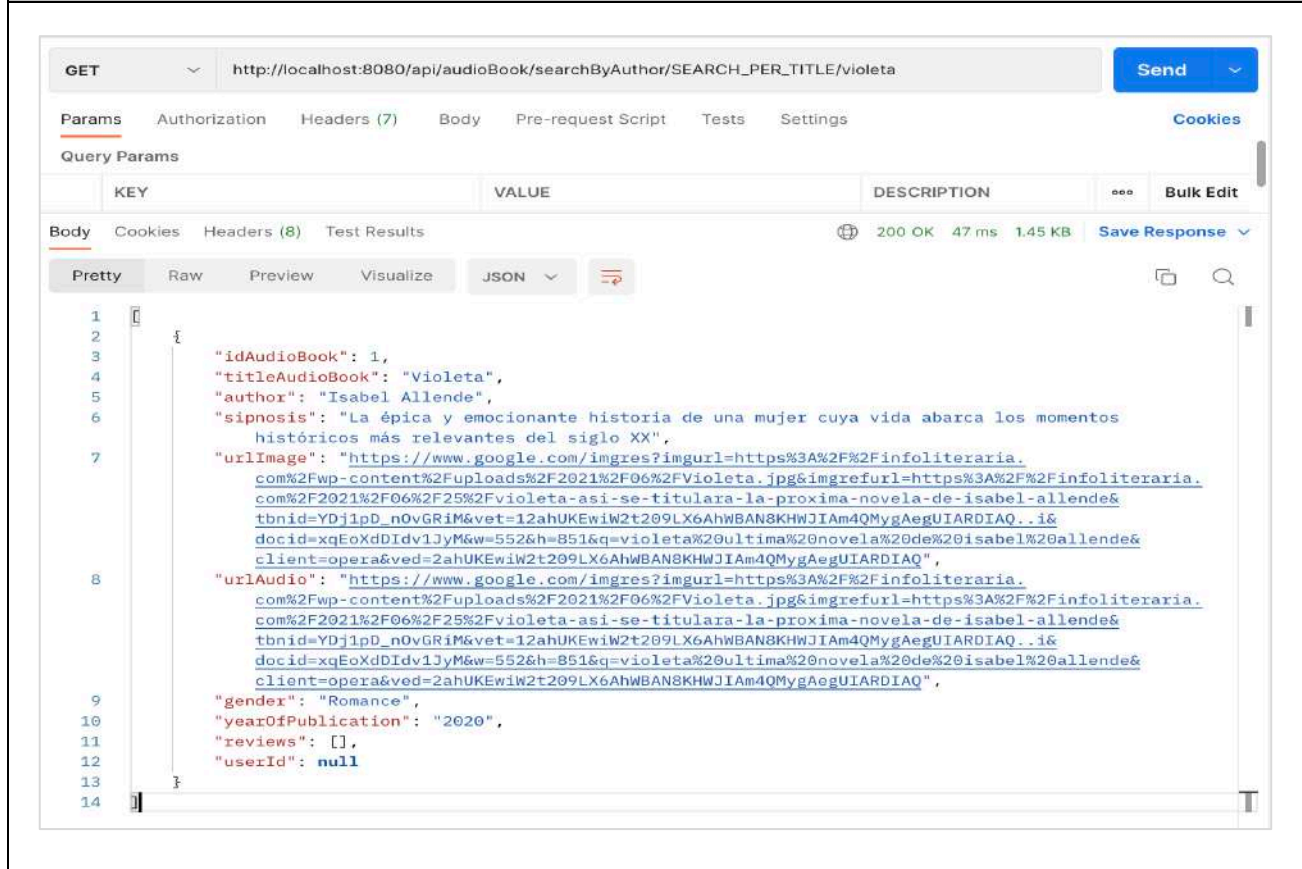


Tabla 38 - Petición GET para buscar audiolibros por título de audiolibro

E. Buscar por género: permite buscar un audiolibro por su género, se realizó una petición GET que recibe dos variables:

- Tipo de búsqueda “**SEARCH_PER_GENDER**”
- Género del audiolibro

MÉTODO	GET
URL	http://localhost:8080/api/audioBook/searchByAuthor/SEARCH_PER_GENDER/romance
VARIABLES	<ul style="list-style-type: none"> ○ SEARCH_PER_GENDER ○ Romance

RESPONSE

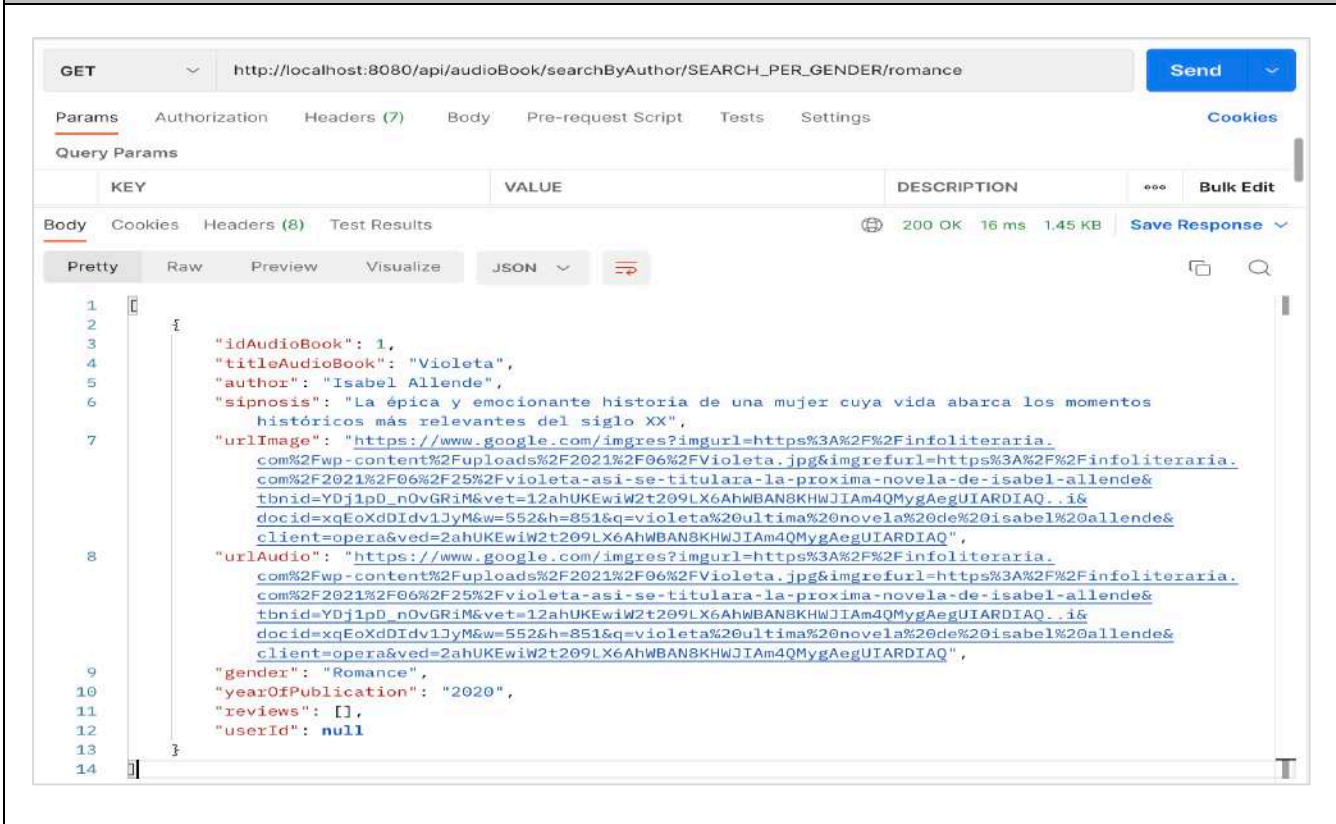


Tabla 39 - Petición GET para buscar audiolibros por género

F. Agregar reseña: permite añadir una reseña a un audiolibro en específico, se realiza una petición POST que recibe un objeto con los atributos de la clase Review y una variable con el id del audiolibro.

MÉTODO	POST
---------------	------

URL	http://localhost:8080/api/audioBook/addReview/1
VARIABLES	o 1
BODY	<pre>{ "userId": "2", "comment": "Interesante novela", "rating": "4" }</pre>

RESPONSE

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/api/audioBook/addReview/1
- Body (Request):**

```
{
  "userId": "2",
  "comment": "Interesante novela",
  "rating": "4"
}
```
- Status:** 200 OK, 77 ms, 341 B
- Body (Response):**

```
{
  "idReview": 2,
  "userId": "2",
  "comment": "Interesante novela",
  "rating": "4"
}
```

Tabla 40 - Petición GET para añadir una reseña a un audiolibro

Para comprobar que la reseña se agregó correctamente se realizó una petición GET para buscar un audiolibro por su id, la petición recibe como variable el id del audiolibro

MÉTODO	GET
---------------	-----

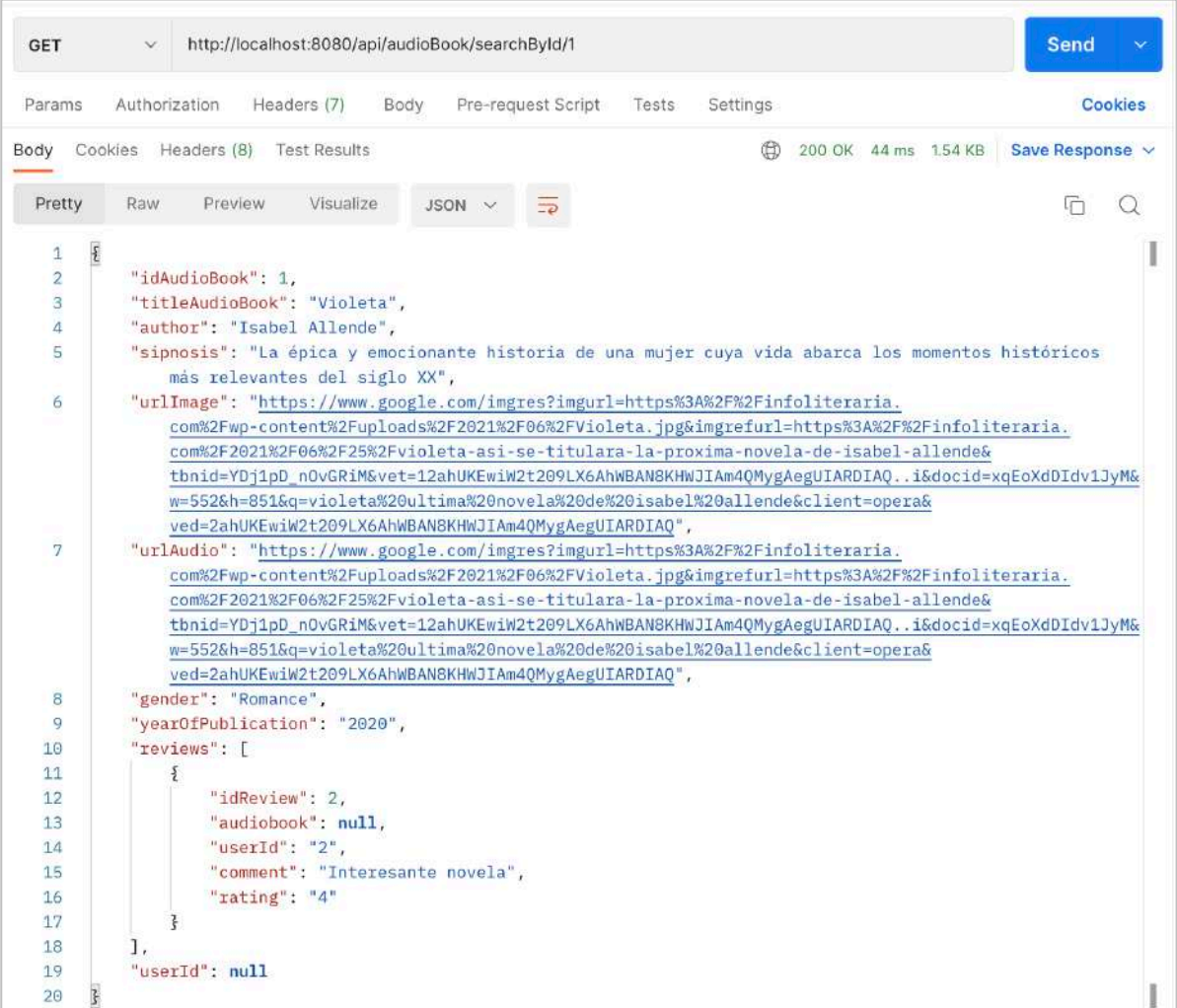
URL	http://localhost:8080/api/audioBook/searchById/1
VARIABLES	o 1
RESPONSE	
 <pre> 1 { 2 "idAudioBook": 1, 3 "titleAudioBook": "Violeta", 4 "author": "Isabel Allende", 5 "sipnosis": "La épica y emocionante historia de una mujer cuya vida abarca los momentos históricos 6 más relevantes del siglo XX", 7 "urlImage": "https://www.google.com/imgres?imgurl=https%3A%2F%2Ffinfoliteraria. 8 com%2Fwp-content%2Fuploads%2F2021%2F06%2Fvioleta.jpg&imgrefurl=https%3A%2F%2Ffinfoliteraria. 9 com%2F2021%2F06%2F25%2Fvioleta-asi-se-titulara-la-proxima-novela-de-isabel-allende& 10 tbnid=YDj1pD_n0vGRiM&vet=12ahUKEwiW2t209LX6AhWBAN8KHwJIAm4QMygAegUIARDIAQ..i&docid=xqEoXd0Idv1JyM& 11 w=552&h=851&q=violeta%20ultima%20novela%20de%20isabel%20allende&client=opera& 12 ved=2ahUKEwiW2t209LX6AhWBAN8KHwJIAm4QMygAegUIARDIAQ", 13 "urlAudio": "https://www.google.com/imgres?imgurl=https%3A%2F%2Ffinfoliteraria. 14 com%2Fwp-content%2Fuploads%2F2021%2F06%2Fvioleta.jpg&imgrefurl=https%3A%2F%2Ffinfoliteraria. 15 com%2F2021%2F06%2F25%2Fvioleta-asi-se-titulara-la-proxima-novela-de-isabel-allende& 16 tbnid=YDj1pD_n0vGRiM&vet=12ahUKEwiW2t209LX6AhWBAN8KHwJIAm4QMygAegUIARDIAQ..i&docid=xqEoXd0Idv1JyM& 17 w=552&h=851&q=violeta%20ultima%20novela%20de%20isabel%20allende&client=opera& 18 ved=2ahUKEwiW2t209LX6AhWBAN8KHwJIAm4QMygAegUIARDIAQ", 19 "gender": "Romance", 20 "yearOfPublication": "2020", 21 "reviews": [22 { 23 "idReview": 2, 24 "audiobook": null, 25 "userId": "2", 26 "comment": "Interesante novela", 27 "rating": "4" 28 } 29], 30 "userId": null 31 } </pre>	

Tabla 41 - Petición GET para buscar un audiolibro por su id

Como podemos observar en la **Tabla 41** la reseña se agregó correctamente.

- G. Eliminar un audiolibro:** permite eliminar un audiolibro mediante su id, se realizó una petición DELETE que recibe como variable el id del audiolibro que se desee eliminar.

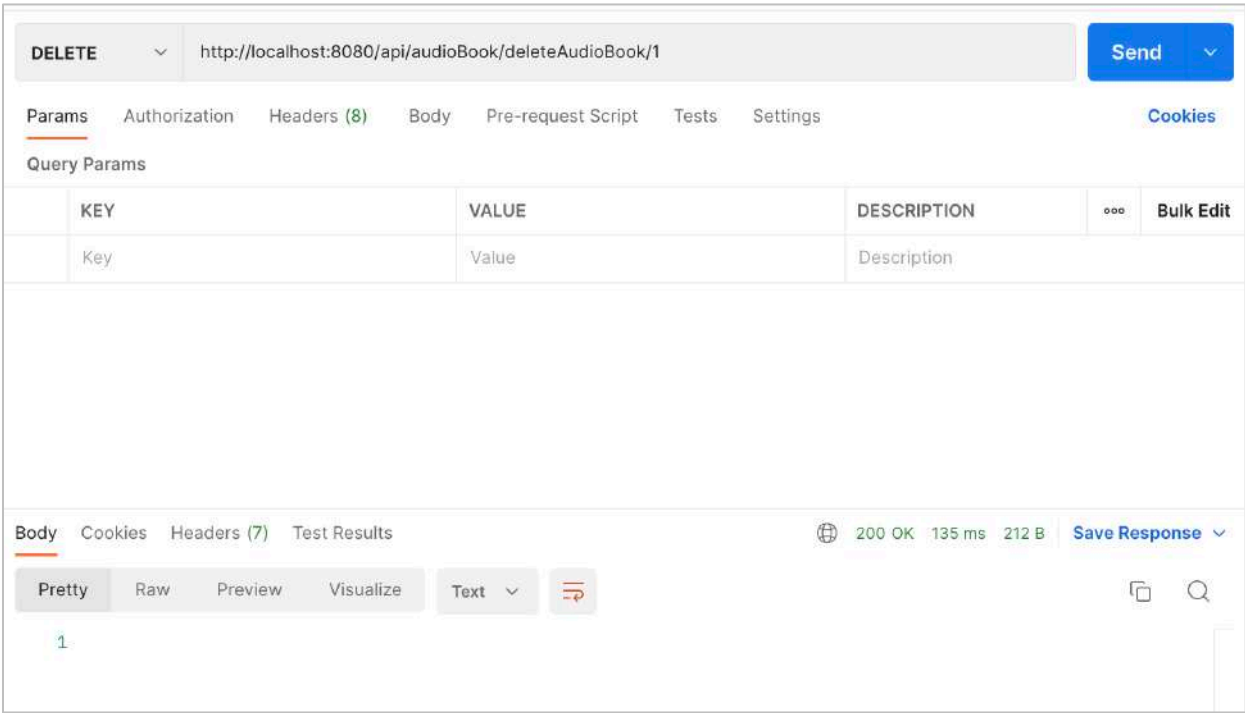
MÉTODO	DELETE										
URL	http://localhost:8080/api/audioBook/deleteAudioBook/1										
VARIABLES	○ 1										
RESPONSE											
 <p>The screenshot shows a REST client interface. At the top, the method is set to 'DELETE' and the URL is 'http://localhost:8080/api/audioBook/deleteAudioBook/1'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Query Params' section is visible with a table:</p> <table border="1"> <thead> <tr> <th>KEY</th> <th>VALUE</th> <th>DESCRIPTION</th> <th>...</th> <th>Bulk Edit</th> </tr> </thead> <tbody> <tr> <td>Key</td> <td>Value</td> <td>Description</td> <td></td> <td></td> </tr> </tbody> </table> <p>At the bottom, the 'Body' tab is selected, showing a response of '1'. The status bar indicates '200 OK', '135 ms', and '212 B'.</p>		KEY	VALUE	DESCRIPTION	...	Bulk Edit	Key	Value	Description		
KEY	VALUE	DESCRIPTION	...	Bulk Edit							
Key	Value	Description									

Tabla 42 - Petición DELETE para eliminar un audiolibro

Para comprobar que el audiolibro se eliminó correctamente se hizo una petición GET para buscar un audiolibro por su id.

MÉTODO	GET
URL	http://localhost:8080/api/audioBook/getAllAudioBooks
VARIABLES	○ 1

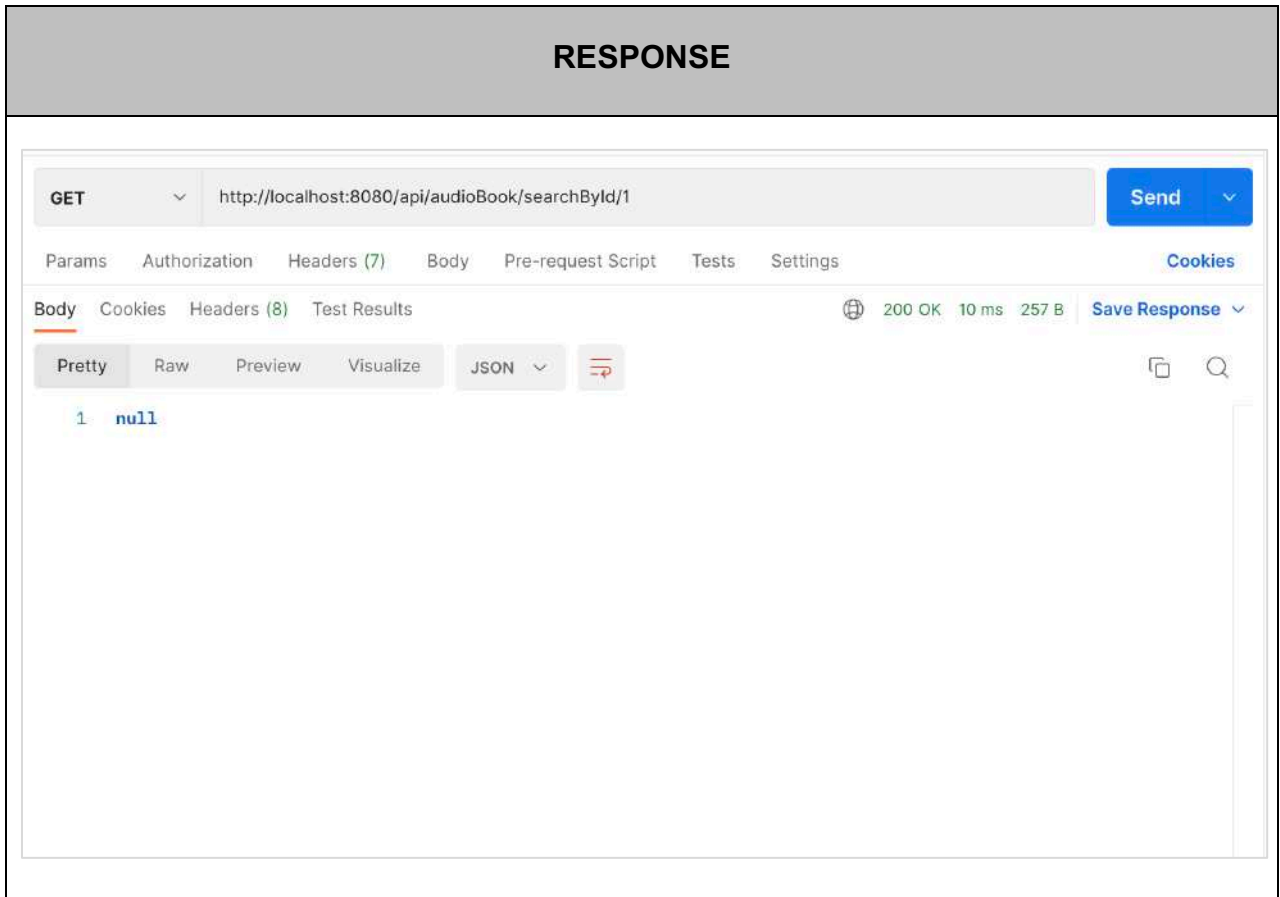


Tabla 43 - Petición GET para buscar un audiolibro por su id con respuesta null

Como se observa en la **Tabla 43** la respuesta es null ya que el audiolibro se eliminó exitosamente.

- **Pruebas no funcionales del Sprint 1**

Las pruebas de rendimiento [26] se las realizó a tres de los métodos del microservicio **AudioBookCatalogMicroservice**, considerando que son los más relevantes, para las pruebas se utilizó la herramienta JMeter [46], a continuación, se detalla el proceso de cada plan de prueba realizado:

A. Añadir audiolibro: se realizó la configuración del hilo en donde se indicó que la prueba soporte 100 usuarios cada 10 segundos.

Thread Group

Name: POST

Comments:

Action to be taken after a Sampler error:

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users): 100

Ramp-up period (seconds): 10

Loop Count: Infinite 1

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Figura 51 - Configuración del hilo para la prueba de la petición Add AudioBook

Se agregó una respuesta HTTP de tipo POST y el objeto de tipo audiolibro que deberá ser enviado durante las peticiones.

HTTP Request

Name: HTTP Request Add AudioBook

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: localhost Port Number: 8080

HTTP Request

POST Path: /api/audioBook/addAudioBook Content encoding: UTF-8

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

```

1: {
2:   "titleAudioBook": "Violeta",
3:   "author": "Isabel Allende",
4:   "synopsis": "novela",
5:   "urlImage": "",
6:   "urlAudio": "",
7:   "gender": "Novela",
8:   "yearOfPublication": "2019"
9: }
10:

```

Figura 52 - Configuración de la respuesta Add AudioBook

Como la petición espera recibir un objeto en formato JSON, a nuestra prueba se agregó un gestor de cabeceras HTTP (HTTP Header Manager).

HTTP Header Manager

Name:

Comments:

Headers Stored in the Header Manager

Name:	Value
Content-Type	application/json

Figura 53 - HTTP Header Manager para la petición Add AudioBook

A continuación, se detallan los resultados obtenidos:

- **Informe resumido:** se visualizó que la prueba fue exitosa ya que durante la ejecución no dio ningún error, obteniendo un valor máximo de respuesta de 17 milisegundos.

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename: Log/Display Only: Errors Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s...	Sent KB/sec	Avg. Bytes
HTTP Reques...	100	12	9	17	1.42	0.00%	10.1/sec	4.78	3.61	485.0
TOTAL	100	12	9	17	1.42	0.00%	10.1/sec	4.78	3.61	485.0

Figura 54 – Informe resumido de la petición Add AudioBook

- **Árbol de resultados:** desde el cual se observa las peticiones que se realizan logrando visualizar los datos que se envía y la respuesta que recibe cada una de ellas.

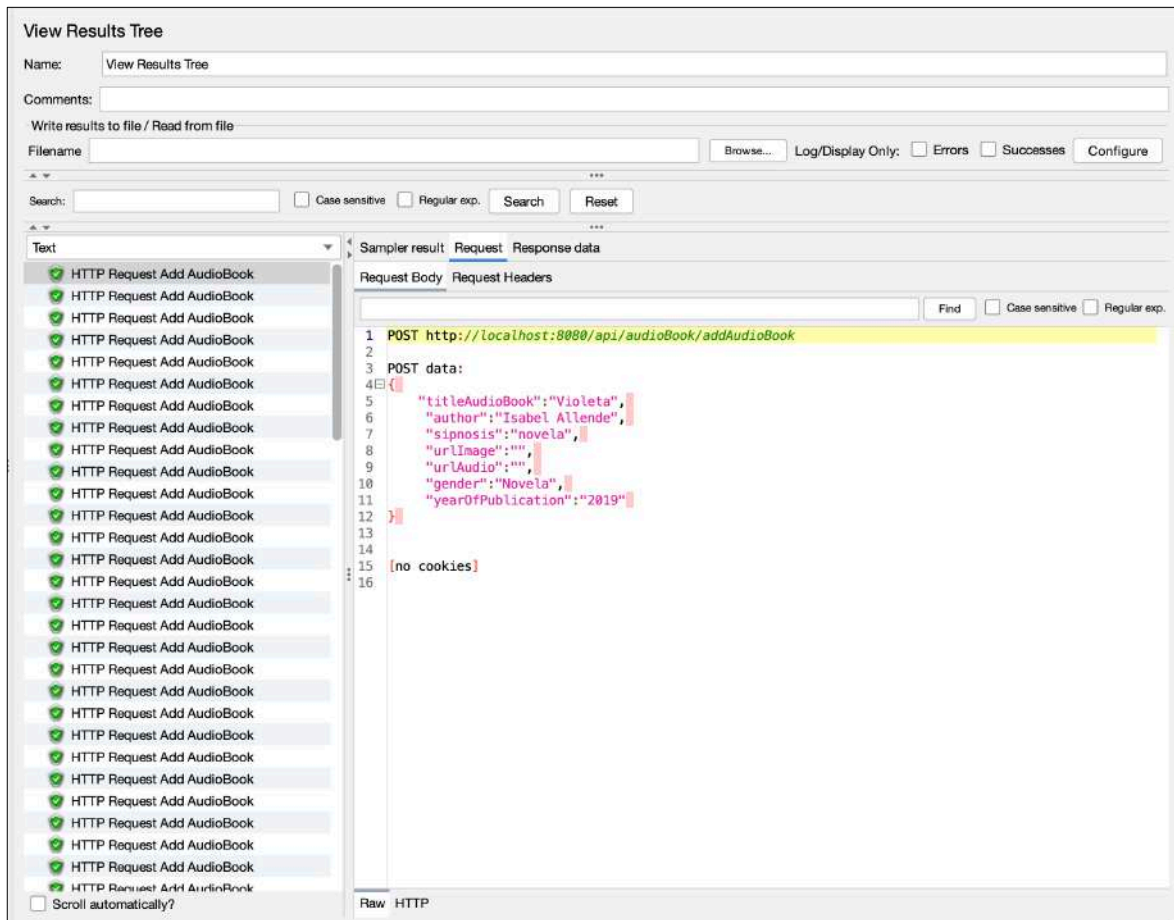


Figura 55 - Árbol de resultados de la prueba de carga de la petición Add AudioBook

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 605.144 peticiones por minuto con una media de 12 y una mediana de 13.

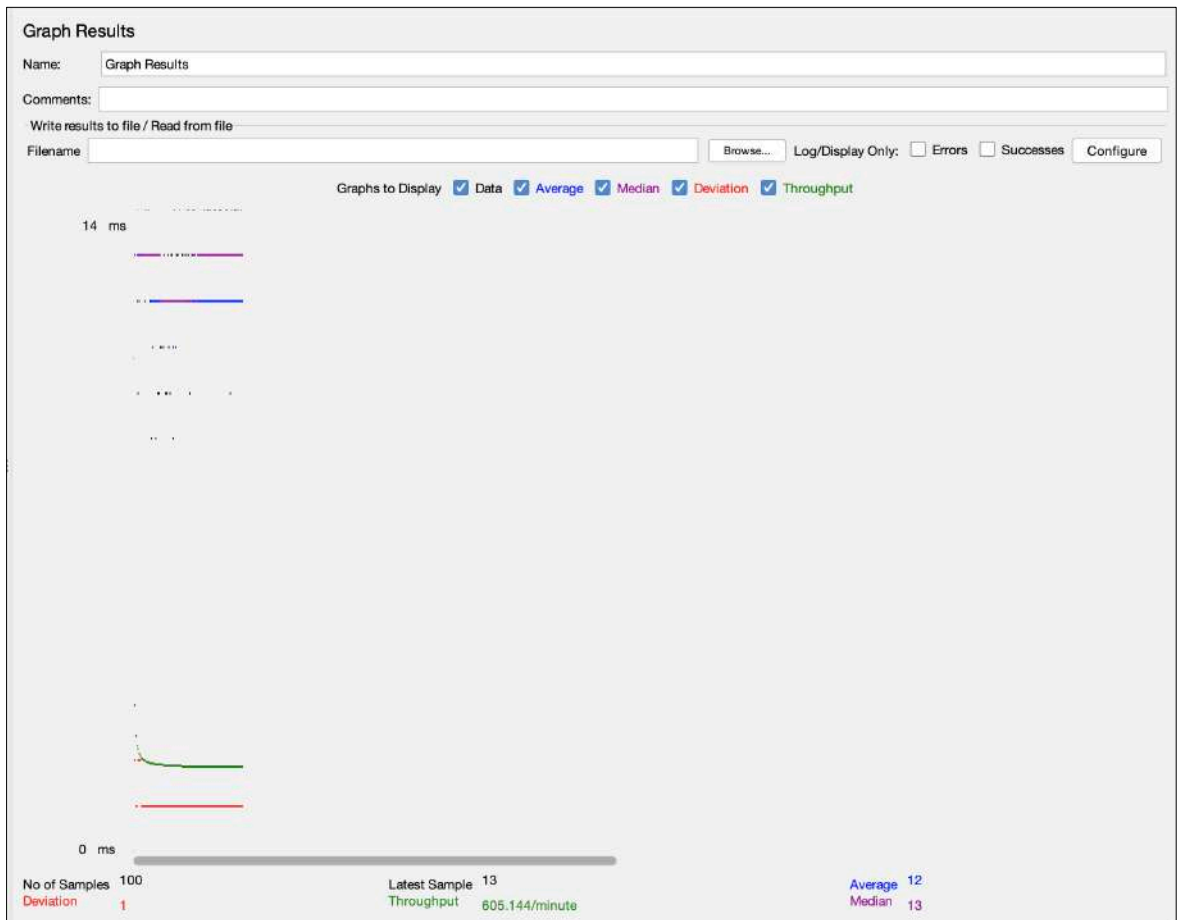


Figura 56 – Gráfico de los resultados de la prueba para la petición Add AudioBook

Con los datos obtenidos se llegó a la conclusión que el servidor responde correctamente al hilo configurado ya que el valor del tiempo de respuesta máximo en segundos es de 0.017 y el rendimiento de 10.1 segundos.

B. Obtener todos los audiolibros: se realizó la configuración del hilo en donde se indicó que la prueba soporte 100 usuarios cada 10 segundos.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Figura 57 - Configuración del hilo para la prueba en JMeter de la petición Get All AudioBooks

Se agregó una respuesta HTTP de tipo GET.

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

GET Path: Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figura 58 - Configuración de la respuesta Get All AudioBooks

A continuación, se detallan los resultados obtenidos:

- **Informe resumido:** donde se visualizó que la prueba de realizó fue exitosa ya que durante la ejecución no dio ningún error.

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: Errors Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s...	Sent KB/sec	Avg. Bytes
HTTP Reques...	100	197	190	234	8.05	0.00%	9.9/sec	665.81	1.37	68820.0
TOTAL	100	197	190	234	8.05	0.00%	9.9/sec	665.81	1.37	68820.0

Figura 59 - Summary Report de la petición Get All AudioBooks

- **Árbol de resultados:** desde el cual se observa las peticiones que se realizan y el valor que recibe cada una de ellas.

View Results Tree

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: Errors Successes

Search: Case sensitive Regular exp.

Sampler result: Request: Response data

Response Body: Response headers

Find Case sensitive Regular exp.

```

[{"idAudioBook":1,"titleAudioBook":"Alicia en el paÃs de las maravillas","author":"Lewis Carroll","sipnosis":"La historia cuenta cÃmo una niÃa llamada Al

```

Scroll automatically?

Figura 60 - Árbol de resultados de la prueba para la petición Get All AudioBooks

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 594.413 peticiones por minuto con una media de 196 y una mediana de 197.

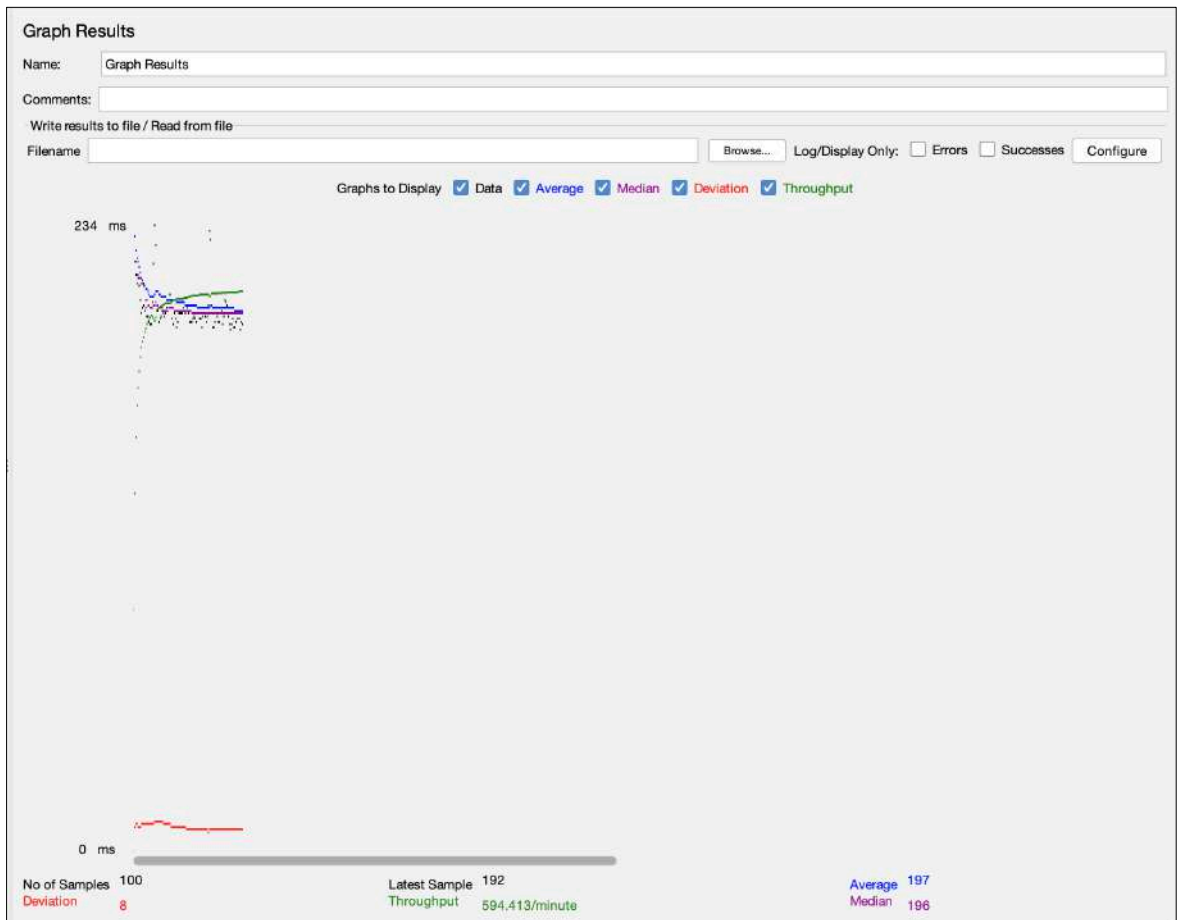


Figura 61 - Gráfico de resultado de la prueba para la petición Get All AudioBooks

Con los datos obtenidos se llegó a la conclusión que el servidor responde correctamente al hilo configurado ya que el valor del tiempo de respuesta máximo en segundos es de 0.234 y el rendimiento de 9.9 segundos.

C. Buscar audiolibros: se realizó la prueba con los tres tipos de búsqueda: por título de audiolibro, por nombre de autor y por género del audiolibro, la configuración del hilo se lo hizo con 100 usuarios por 10 segundos.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Figura 62 - Configuración del hilo para la prueba de la petición Search AudioBooks

Se agregó las respuestas HTTP de tipo GET por cada una de las búsquedas, como se muestra a continuación:

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

GET Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figura 63 - Configuración de la respuesta Search AudioBooks Per Author

HTTP Request

Name: HTTP Request HTTP Request Search AudioBook Per Title

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

GET Path: api/audioBook/searchByAuthor/SEARCH_PER_TITLE/alicia Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figura 64 - Configuración de la respuesta Search AudioBoks Per Title

HTTP Request

Name: HTTP Request Search AudioBook Per Gender

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

GET Path: api/audioBook/searchByAuthor/SEARCH_PER_GENDER/ciencia Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figura 65 - Configuración de la respuesta Search AudioBooks Per Gender

A continuación, se detalla los resultados obtenidos:

- **Informe resumido:** donde se visualizó que la prueba de realizó fue exitosa ya que durante la ejecución no dio ningún error.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Send KB/sec	Avg. Bytes
Search_Per_Author	100	8384	1021	16837	4670.82	0.00%	3.7/sec	1703.97	0.60	466580.6
Search_Per_Title	100	11213	2064	16450	3809.87	0.00%	3.0/sec	1390.47	0.47	482154.0
Search_Per_Gender	100	9054	1001	16238	2882.46	0.00%	2.6/sec	1255.58	0.43	488340.9
TOTAL	300	9550	1001	16837	4041.91	0.00%	7.3/sec	3397.10	1.17	479018.5

Figura 66 - Summary Report de la petición Search AudioBooks

- **Árbol de resultados:** desde el cual se observa las peticiones que se realizan y el valor que recibe cada una de ellas.

The screenshot shows the 'View Results Tree' window. On the left, a tree view lists multiple 'HTTP Request Search Audi' items. The right pane shows the 'Response Body' for a selected request, containing a JSON object with details about an audiobook titled 'Alicia en el país de las maravillas' by Lewis Carroll. The JSON includes fields for title, author, synopsis, image URL, and other metadata.

Figura 67 – Árbol de resultados de la prueba para la petición Search AudioBooks

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 1812.141 peticiones por minuto con una media de 8 y una mediana de 8.

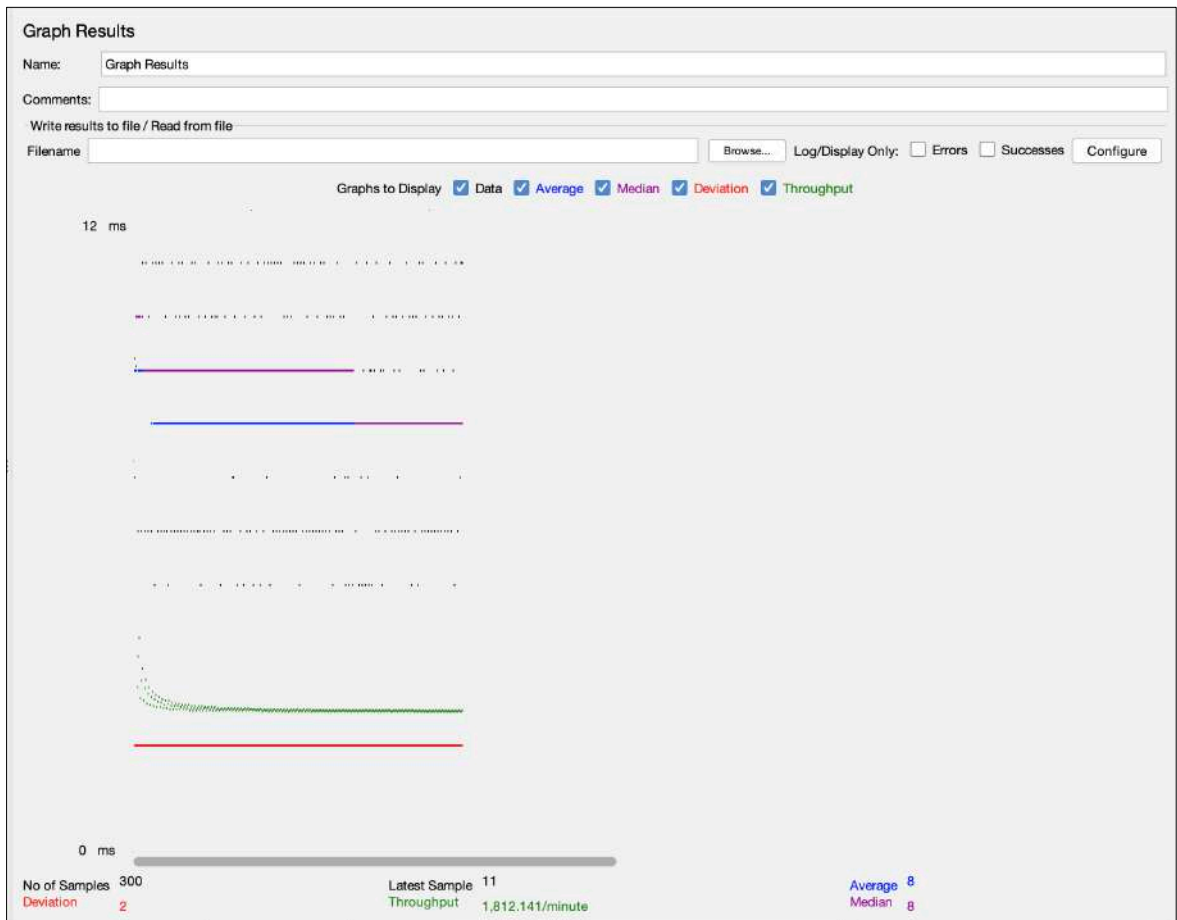


Figura 68 - Gráfico de resultado de la prueba para la petición Search AudioBooks

Con los datos obtenidos se llegó a la conclusión que el servidor responde correctamente el hilo configurado ya que el valor del tiempo de respuesta máximo total por las tres peticiones es 0.014 segundos y el rendimiento total por las tres peticiones es 30.2 segundos.

4.2. Ejecución de pruebas del Sprint 2

- **Pruebas funcionales del Sprint 2**

Se realizaron pruebas de aceptación [25] a cada historia de usuario de la autenticación y registro de usuarios de tipo autor, a continuación, se indica los resultados obtenidos por cada criterio de aceptación verificando su cumplimiento

A. Creación de cuentas con usuario y contraseña desde el aplicativo web:

donde el usuario de tipo autor crea una cuenta, cumpliendo los siguientes criterios de aceptación:

- El email debe estar en el formato xxx@xxx.com
- La contraseña contendrá como máximo 8 caracteres
- El nombre de usuario contendrá como máximo 8 caracteres
- El usuario elegirá subir una foto desde su equipo
- El formato de la imagen será jpg, jpeg o png

Cuando el usuario ingrese al aplicativo web seleccione la opción “**¿Es nuevo aquí? Cree una cuenta**”, se redireccionará a la pantalla de registro, en la cual debe llenar los campos solicitados de manera correcta, si el usuario ingresa incorrectamente el aplicativo web no le permitirá crear la cuenta hasta que los campos sean válidos. Como se observa a continuación:

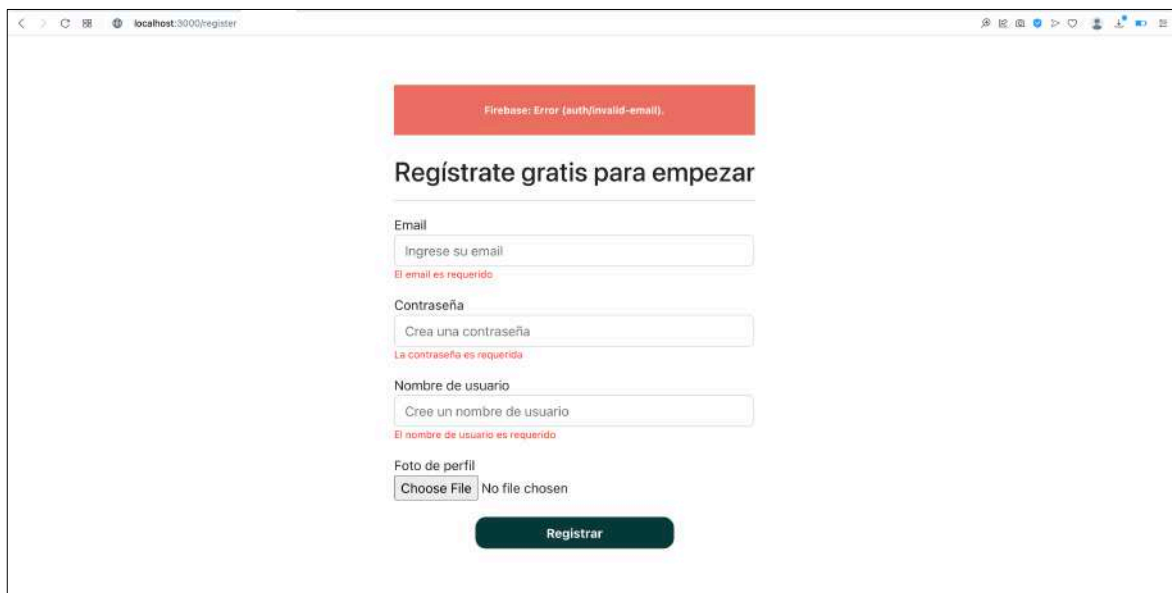


Figura 69 – Registro de un usuario autor fallido desde el aplicativo web

En caso de que el usuario desee elegir una imagen cuando realice el registro el aplicativo web abrirá una imagen emergente en donde solo podrá seleccionar imágenes en formato jpeg, jpg o png como se observa a continuación:

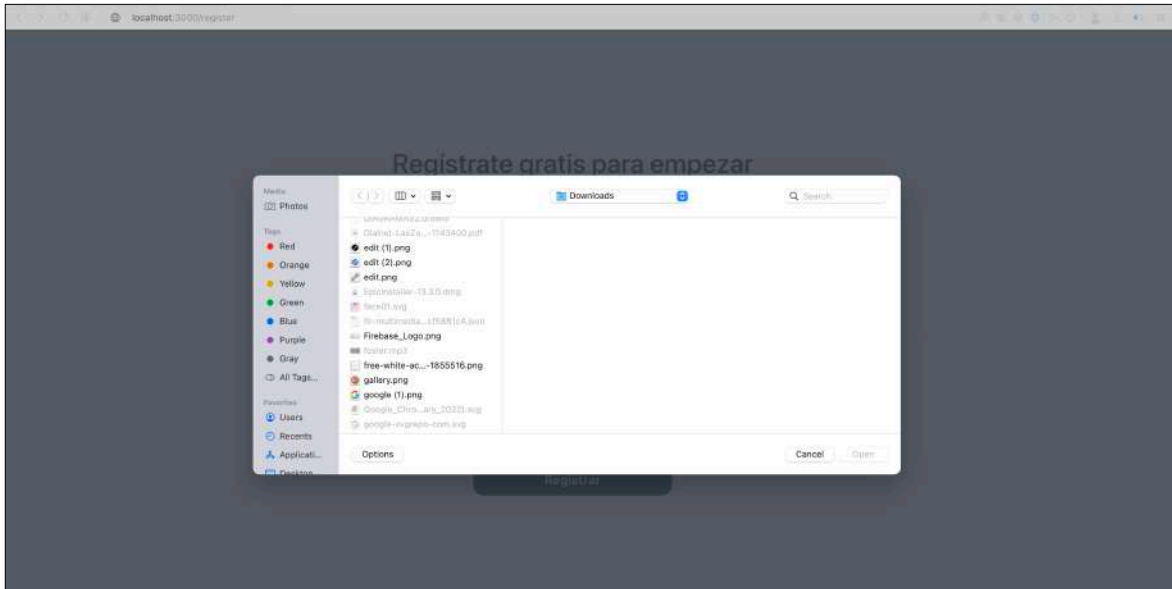


Figura 70 – Selección de una imagen para foto de perfil desde el aplicativo web

Si el usuario ingresa correctamente los campos solicitados el aplicativo web le permite el registro:

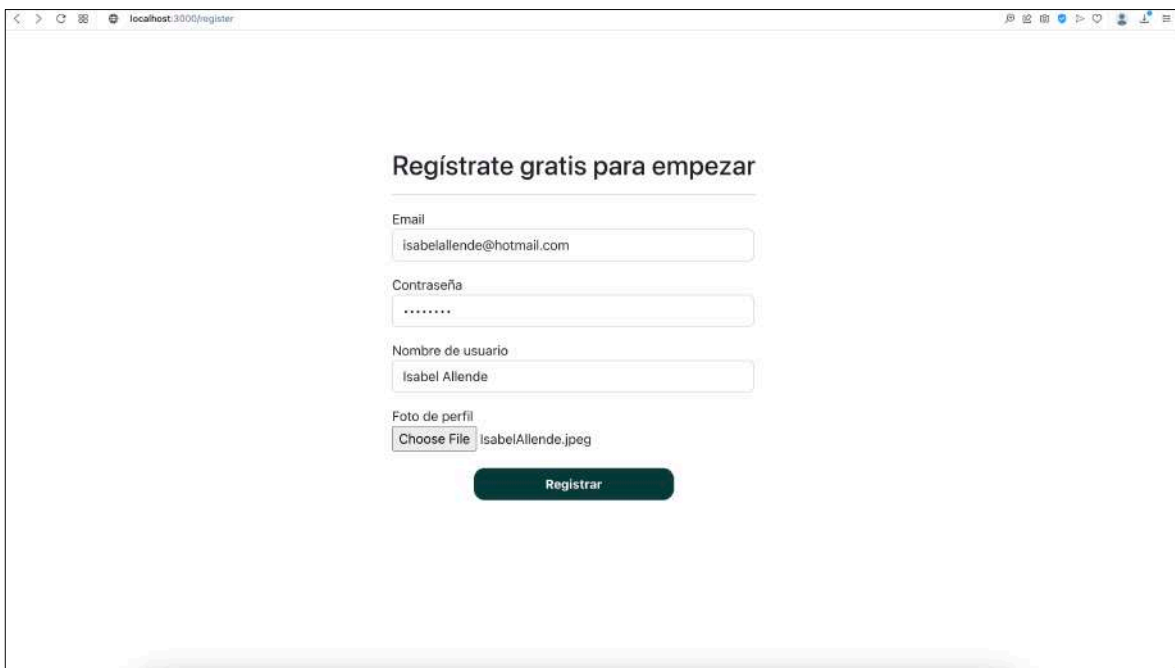


Figura 71 – Registro de usuario autor correcto desde el aplicativo web

El aplicativo web lo redirecciona a la pantalla de inicio de sesión donde deberá ingresar sus credenciales:

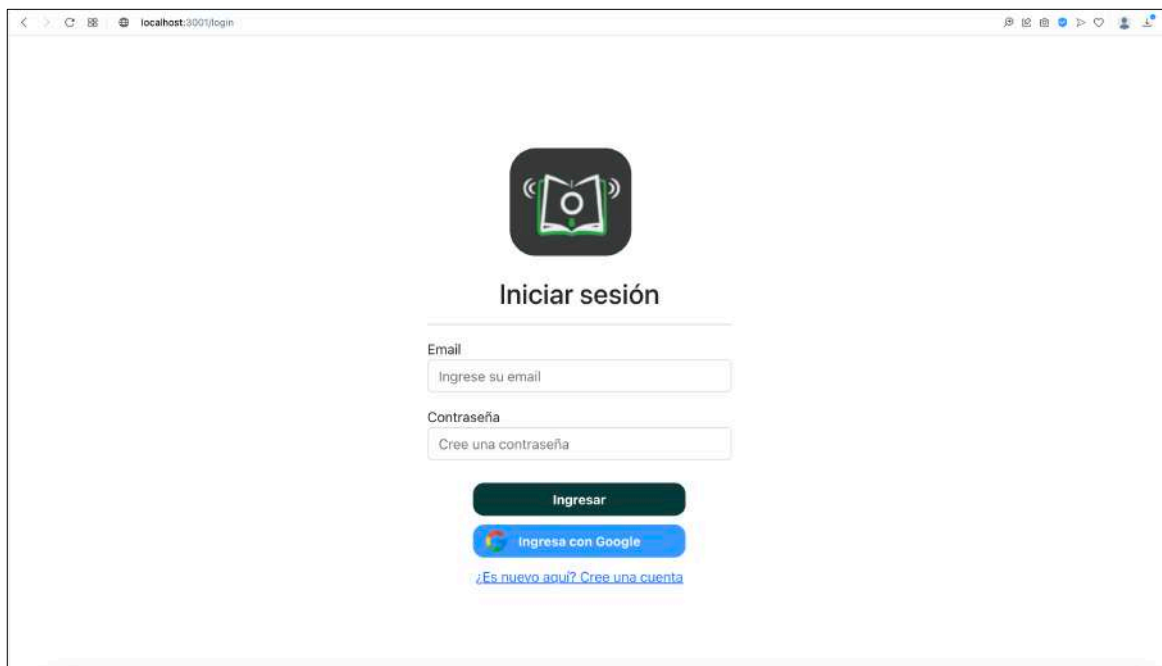


Figura 72 - Pantalla de inicio de sesión del aplicativo web

Una vez que haya iniciado sesión podrá observar la pantalla principal antes de que el usuario añada un audiolibro.

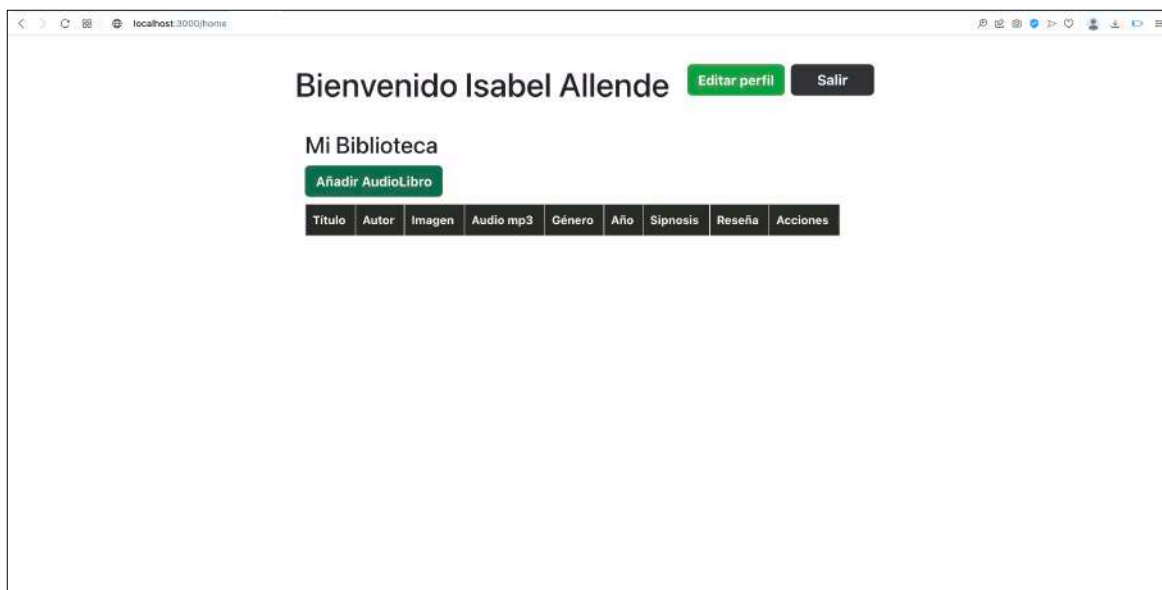


Figura 73 - Pantalla principal del aplicativo web

B. Creación de cuentas con Google desde el aplicativo web: donde el usuario de tipo autor puede crear una cuenta con su correo de Google, seleccionando la opción **“Ingresa con Google”**.

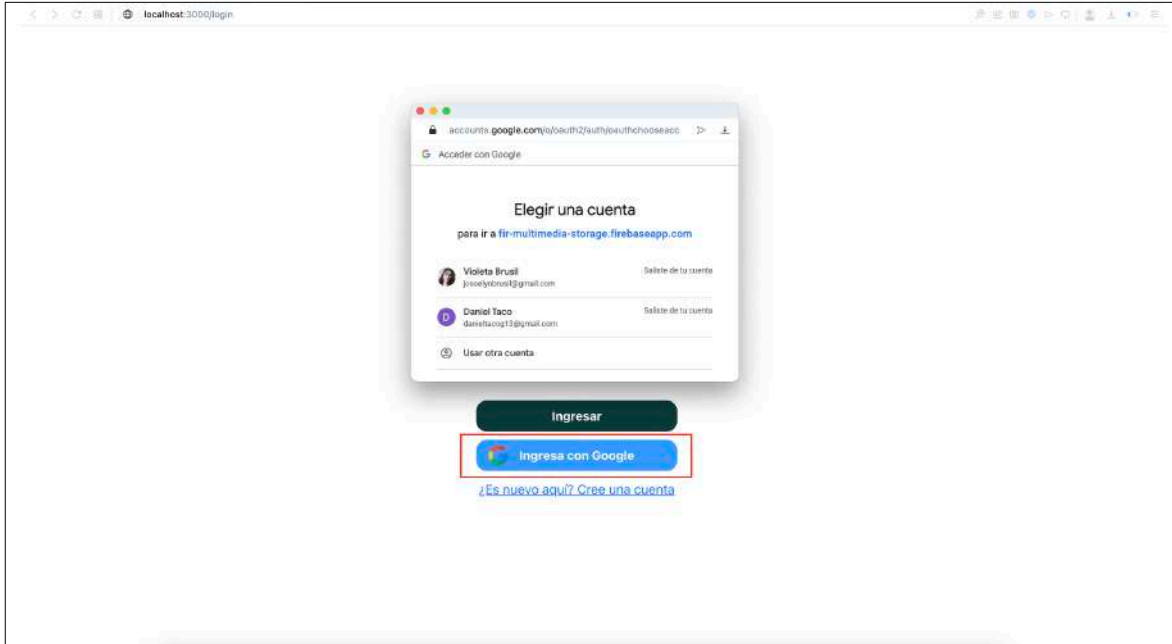


Figura 74 - Inicio de sesión con Google desde el aplicativo web

Como se observa en la **Figura 74**, el usuario selecciona la cuenta que desea registrar y posteriormente ingresa su contraseña para concluir con el registro. Al terminar el proceso el aplicativo web lo redirecciona a la pantalla principal personalizada por cada usuario como se observó en la **Figura 73**.

C. Autenticación de usuarios desde el aplicativo web: dado el usuario se registró de manera exitosa podrá ingresar al aplicativo web con su email y contraseña desde la pantalla de inicio de sesión. Si el usuario selecciona la opción **“Ingresar”** sin llenar los campos correspondientes a email y contraseña, el aplicativo web le indicará que son requeridos para iniciar sesión, como se observa a continuación:

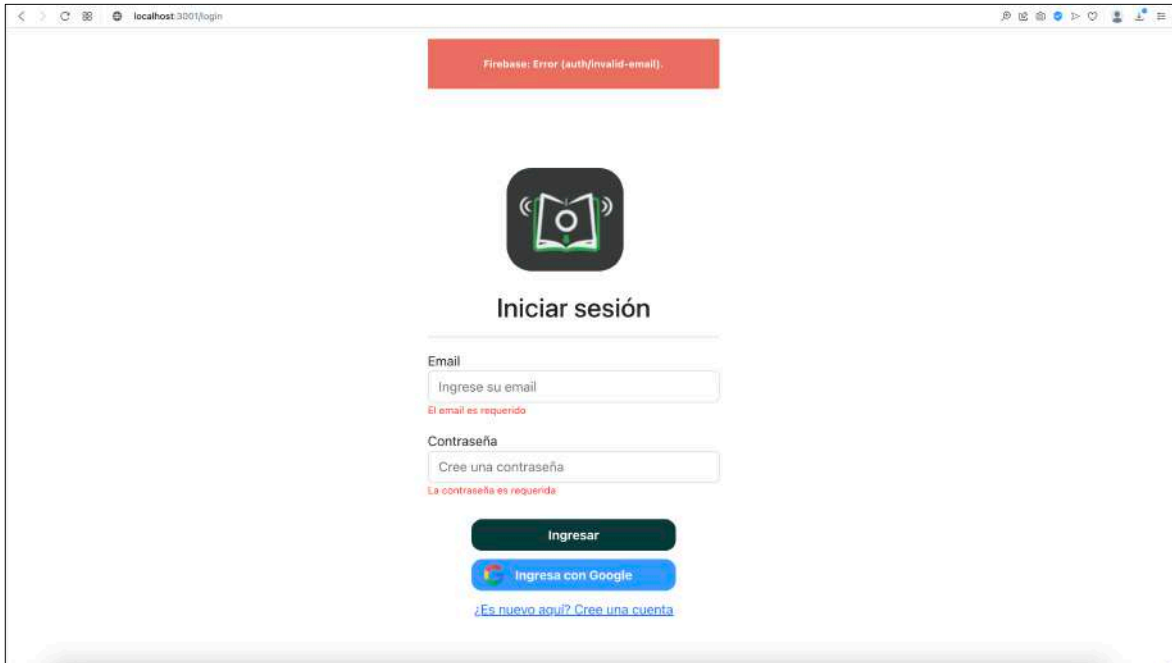


Figura 75 - Inicio de sesión fallido desde el aplicativo web

Si el usuario ingresa los datos correctamente el aplicativo web le permitirá el ingreso:

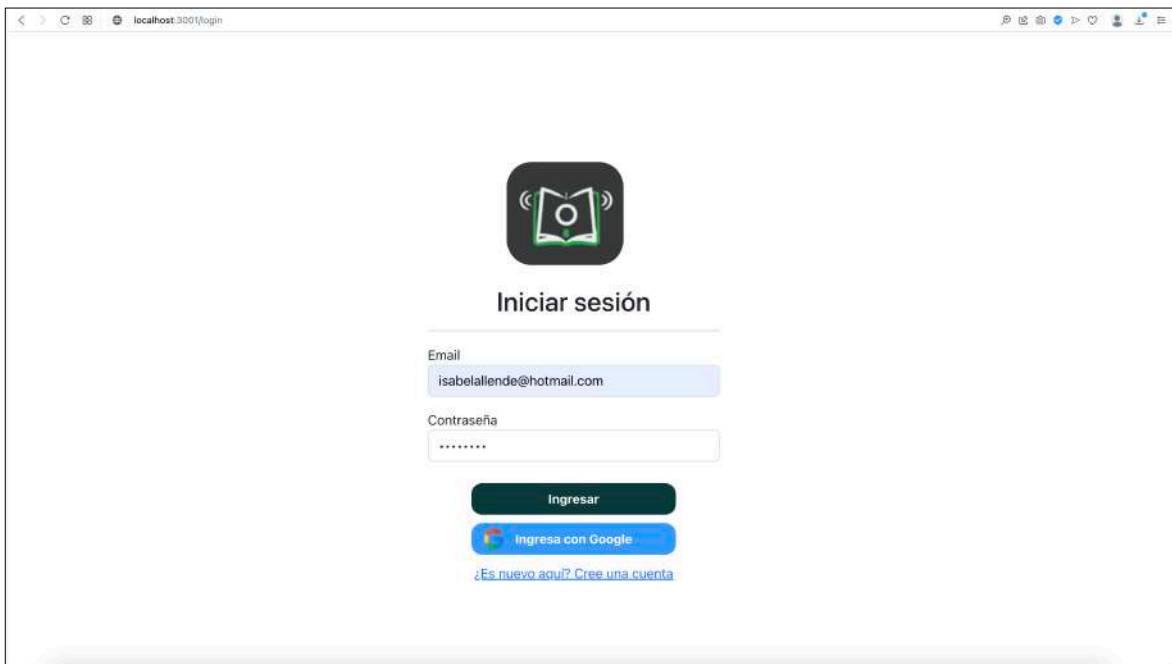


Figura 76 - Inicio de sesión exitoso desde el aplicativo web

D. Editar información personal desde el aplicativo web: dado que el usuario seleccione la opción **“Editar perfil”**, el aplicativo web lo redirecciona a la pantalla de editar información en la que se encuentra los datos del usuario que podrá editar. Como se muestra a continuación:

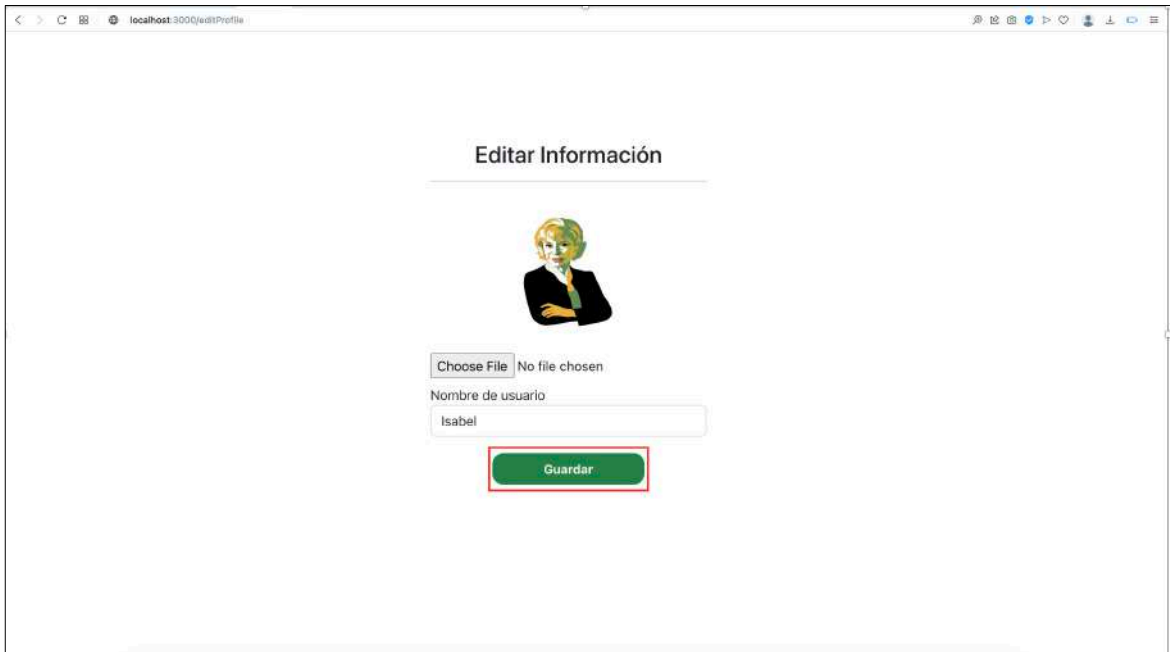


Figura 77 – Actualización de la información personal del usuario autor desde el aplicativo web

Una vez que el usuario haya modificado sus datos selecciona la opción **“Guardar”**, el aplicativo web lo redirecciona a la pantalla principal donde se observa los cambios realizados:

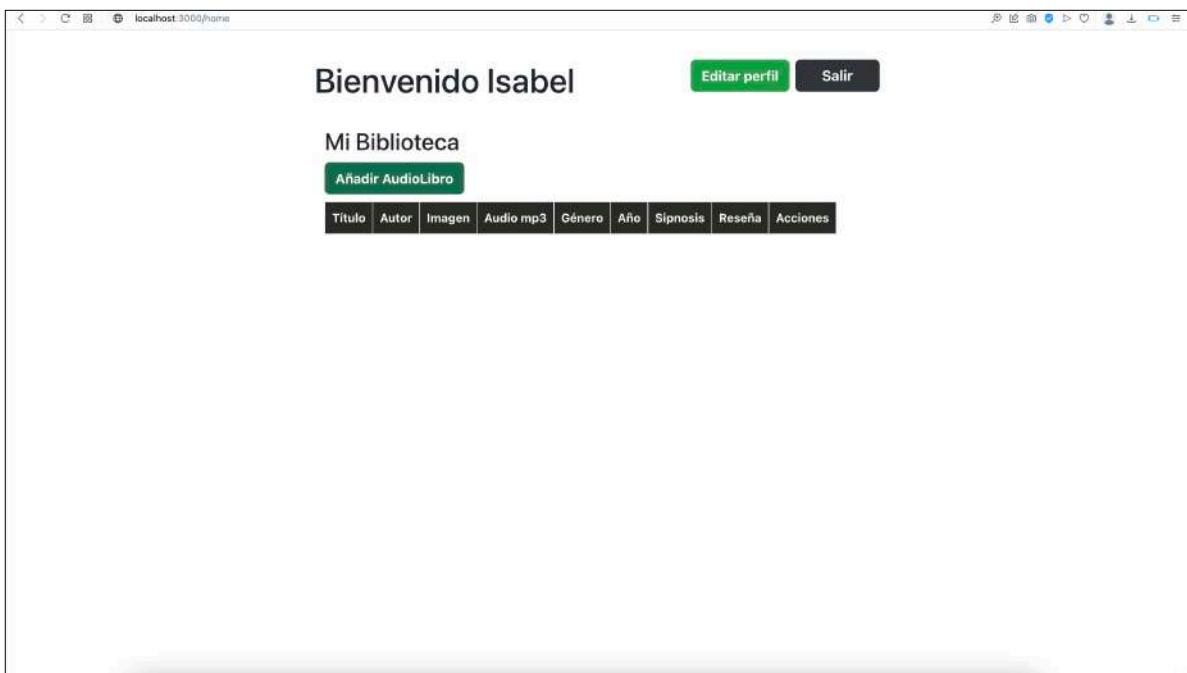


Figura 78 – Pantalla principal cuando el usuario haya editado su información personal

E. Autenticación usuario administrador: en este caso el usuario administrador se encuentra registrado con el correo (administrador@hotmail.com) desde la base de firebase, por lo tanto, para su autenticación al aplicativo web debe ingresar su email y contraseña. Una vez que llene los campos solicitados deberá seleccionar la opción **“Ingresar”**.

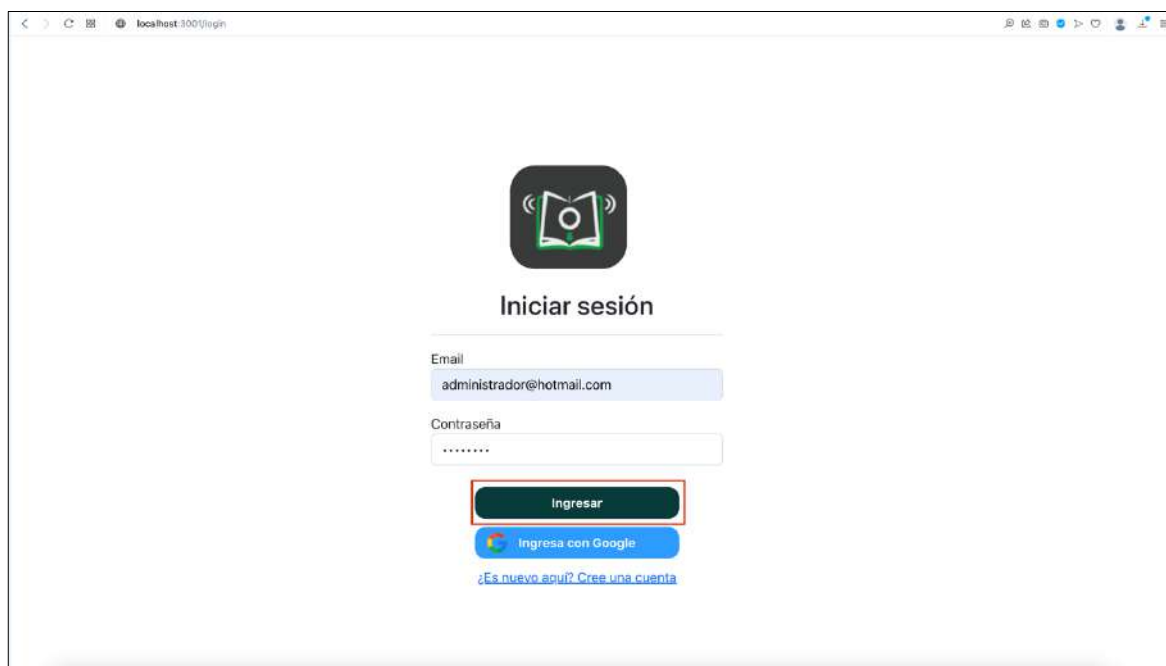


Figura 79– Pantalla de inicio de sesión del usuario administrador desde el aplicativo web

Dado que el usuario ingreso al aplicativo web este le redireccionará a la pantalla principal para el usuario administrador como se observa en la siguiente imagen:

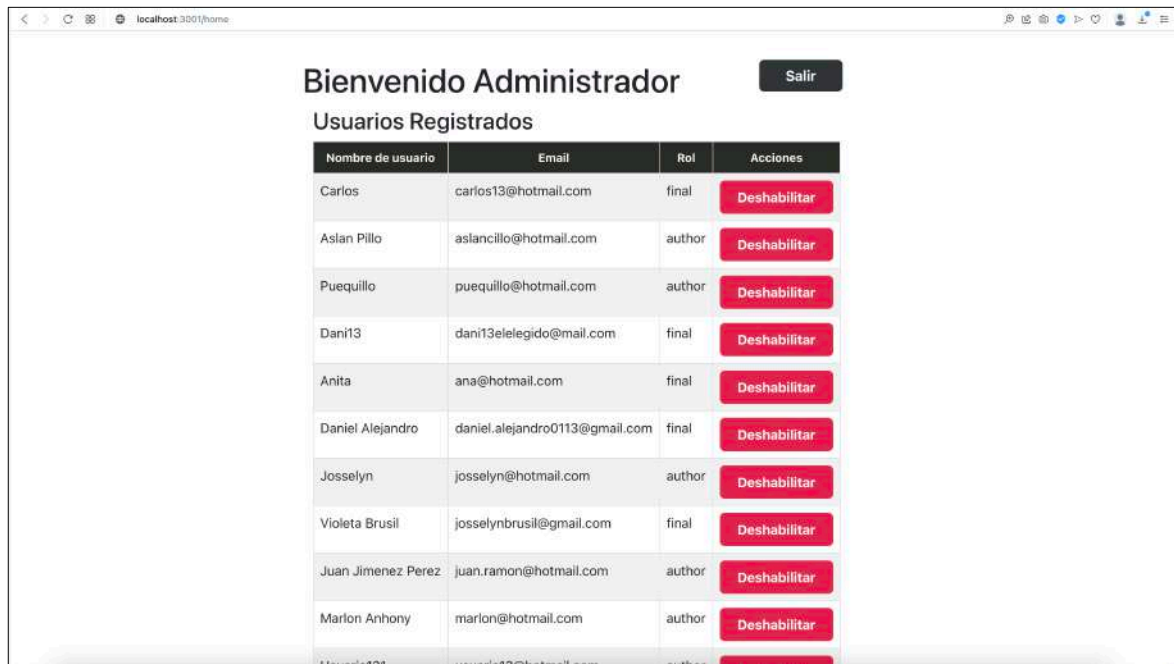


Figura 80 – Pantalla principal del usuario administrador

F. Habilitar / deshabilitar cuentas: cuando el usuario administrador haya ingresado al aplicativo web con su email y contraseña podrá observar la lista de todos los usuarios registrados, con la opción de habilitar o deshabilitar la cuenta seleccionando la opción **“Deshabilitar”**, la misma que cambiara a **“Habilitar”**.

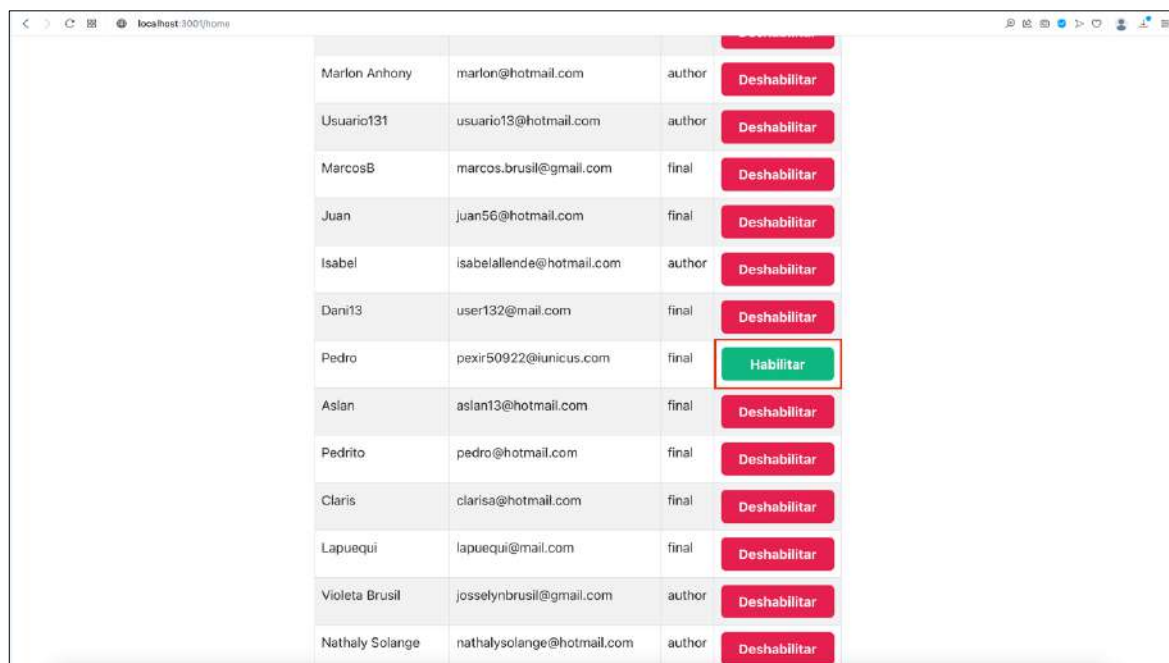
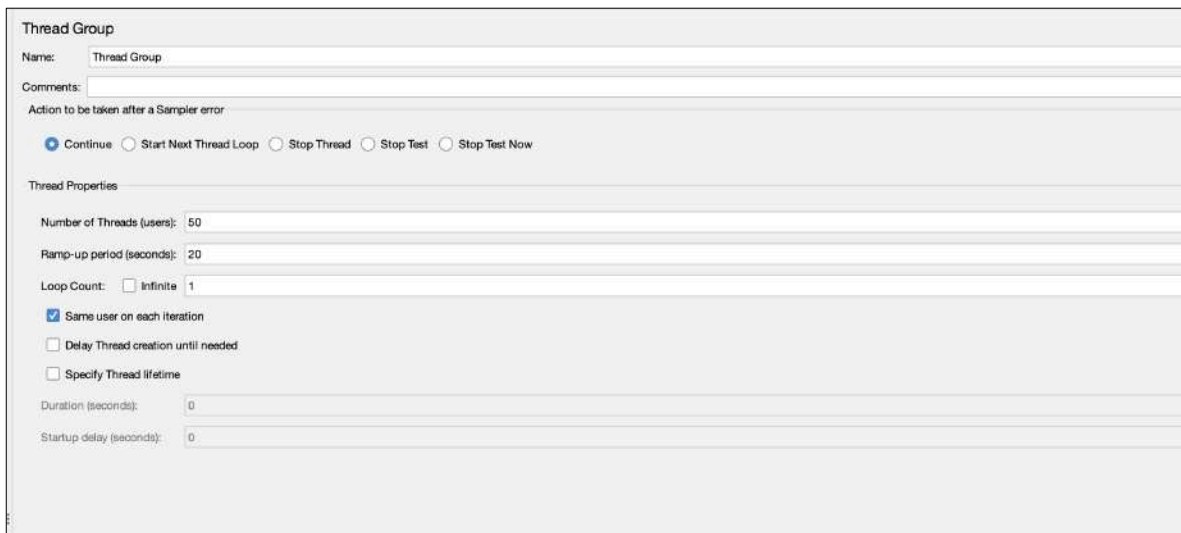


Figura 81 – Pantalla para habilitar/deshabilitar la cuenta de un usuario

- **Pruebas no funcionales del Sprint 2**

La prueba de carga [26] se realizó a la página de inicio de sesión del aplicativo web con la extensión para Chrome de BlazeMeter [47], la misma que permite realizar una grabación de la interacción cuando un usuario inicia sesión guardando los datos en un archivo, el cual fue ejecutado con JMeter [46] obteniendo los siguientes resultados:

- **Grupo de hilos:** para la prueba se configuro el grupo de hilos con 50 usuarios en un periodo de 20 segundos.



The image shows the 'Thread Group' configuration window in Apache JMeter. The 'Name' field is set to 'Thread Group'. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. In the 'Thread Properties' section, 'Number of Threads (users)' is set to 50, 'Ramp-up period (seconds)' is set to 20, and 'Loop Count' is set to 1. The 'Same user on each iteration' checkbox is checked. Other options like 'Delay Thread creation until needed' and 'Specify Thread lifetime' are unchecked. 'Duration (seconds)' and 'Startup delay (seconds)' are both set to 0.

Figura 82 – Configuración del hilo para la prueba de carga de la página de inicio del aplicativo web

- **Árbol de resultados:** desde el cual se observa las peticiones que el usuario realiza para acceder al aplicativo web cuando inicia sesión.

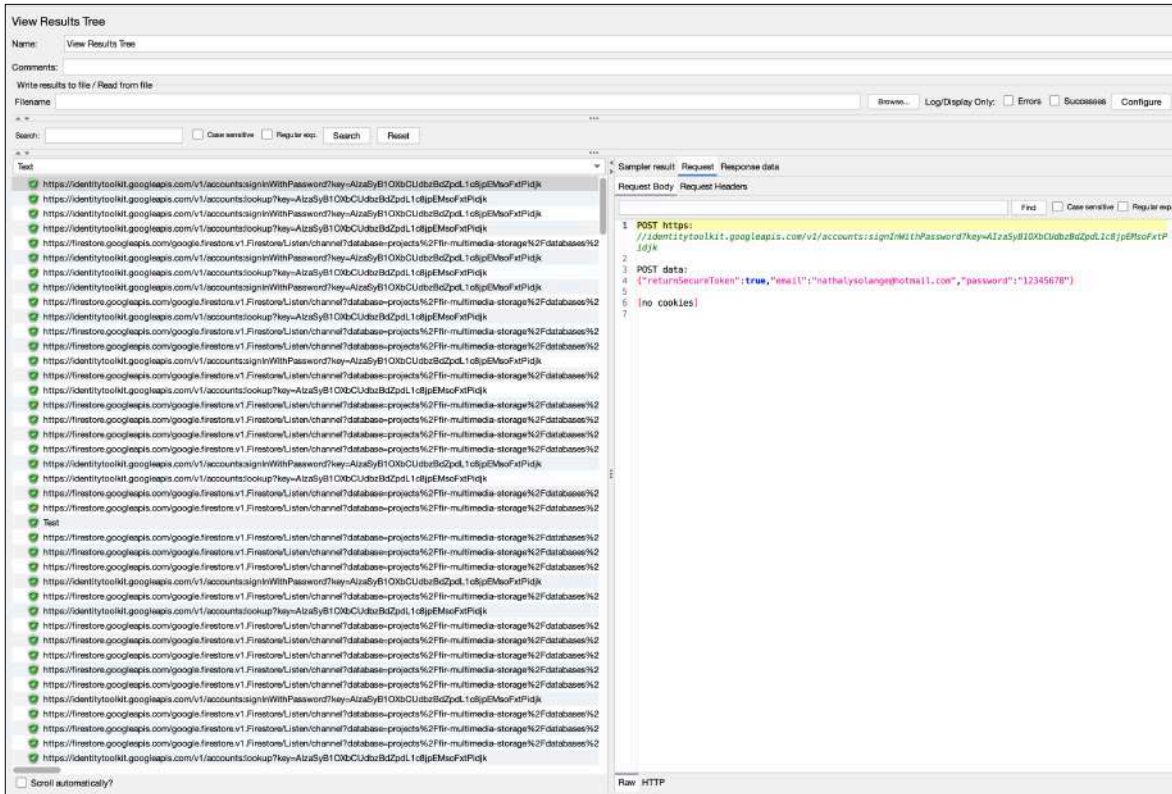


Figura 83 – Árbol de resultados de las peticiones que el usuario realiza al iniciar sesión en el aplicativo web

- **Informe resumido:** se visualiza que cada petición que el usuario realiza se hace de manera exitosa, obteniendo el siguiente resultado:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://identitytoolkit.googleapis.c...	50	295	212	483	76.42	0.00%	2.5/sec	5.10	1.68	2069.0
https://identitytoolkit.googleapis.c...	50	127	101	173	25.54	0.00%	2.6/sec	3.28	3.92	1307.0
https://firestore.googleapis.com/gp...	50	462	182	678	117.45	0.00%	2.5/sec	1.42	5.98	574.0
https://firestore.googleapis.com/gp...	50	333	73	470	107.89	0.00%	2.5/sec	1.42	2.15	574.0
https://firestore.googleapis.com/gp...	50	327	87	538	117.07	0.00%	2.5/sec	1.41	2.15	573.9
https://firestore.googleapis.com/gp...	50	346	70	515	108.03	0.00%	2.5/sec	1.42	2.62	574.0
https://firestore.googleapis.com/gp...	50	359	133	489	93.56	0.00%	2.5/sec	1.41	2.15	573.9
Test	50	2256	1408	3004	354.18	0.00%	2.3/sec	13.79	18.39	6239.8
TOTAL	400	564	87	3004	863.74	0.00%	18.1/sec	27.88	38.77	1580.0

Figura 84 – Informe resumido del plan de prueba a la página de inicio de sesión

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 1086.415 peticiones por minuto con una media de 373 y una mediana de 584.

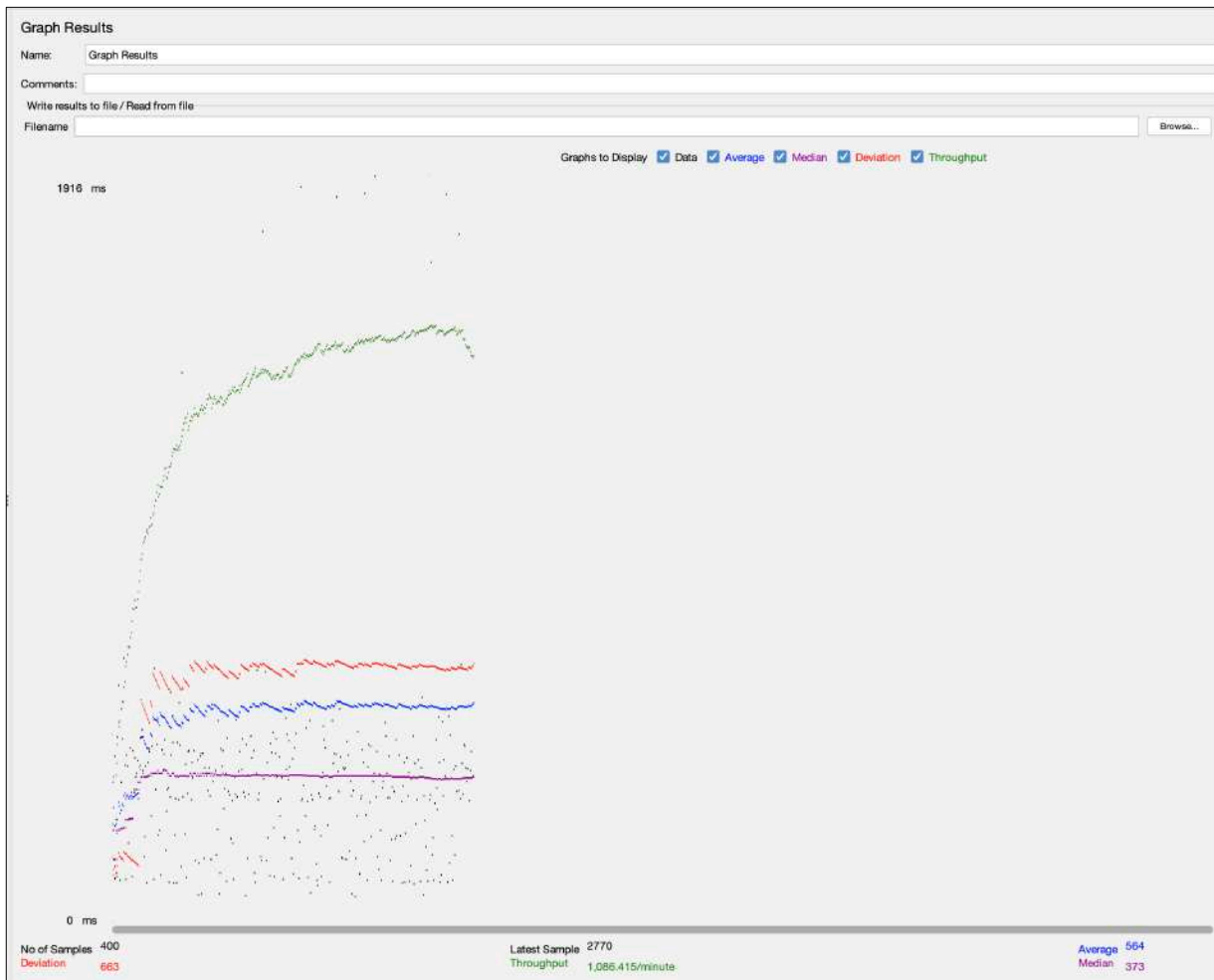


Figura 85 – Gráfico del resultado del plan de prueba de la página de inicio de sesión

Con los datos obtenidos se llegó a la conclusión que los servicios del aplicativo web responden correctamente al hilo configurado ya que el valor del tiempo máximo de respuesta total en segundos es de 3.024 y el rendimiento total de 18.1 segundos.

4.3. Ejecución de pruebas del Sprint 3

- **Pruebas funcionales del Sprint 3**

Se realizaron pruebas de aceptación a las historias de usuario donde el autor podrá añadir, eliminar o editar un audiolibro desde el aplicativo web:

A. Añadir un audiolibro: el usuario autor debe llenar los campos correspondientes a la información del audiolibro cumpliendo los siguientes criterios de aceptación:

- Título del audiolibro

- El título contendrá un máximo de 50 caracteres
- El campo admitirá caracteres alfanuméricos
- Autor
 - El nombre del autor contendrá un máximo de 20 caracteres
- Sinopsis
 - El campo admitirá caracteres alfanuméricos
- Género
 - El usuario deberá seleccionar un género para el audiolibro
- Portada del audiolibro
 - La imagen deberá ser de formato jpeg, jpg o png.
- Archivo mp3
 - El archivo por subir deberá ser de formato mp3
- Año de publicación
 - El campo admitirá caracteres alfanuméricos

Cuando el usuario autor ingrese al aplicativo web con su email y contraseña se redirige a la pantalla principal en donde al seleccionar la opción **“Añadir audiolibro”**, muestra la pantalla para añadir un nuevo audiolibro llenando los campos solicitados como se muestra a continuación:

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/addAudiobook'. The page title is 'Añadir Audiolibro'. A green 'Guardar' button is located in the top right corner. The form fields are as follows:

- Título:** Violeta 1920
- Autor:** Isabel Allende
- Sipnosis:** Violeta, la primera niña de una familia de cinco hermanos, viene al mundo en 1920. Desde su nacimiento su vida estará marcada por acontecimientos extraordinarios, desde la Gripe Española de 1920 hasta la pandemia de Covid de 2020. Recorriendo así, a través de Violeta, cien años de historia.
- Género:** Novela
- Imagen:** A small image of the book cover for 'Violeta' by Isabel Allende is displayed. Below it is a 'Choose File' button with the filename 'violeta.jpeg'.
- Audio:** A 'Choose File' button with the filename 'Violeta.mp3'.
- Año de publicación:** 2022

Figura 86 – Agregación de un nuevo audiolibro

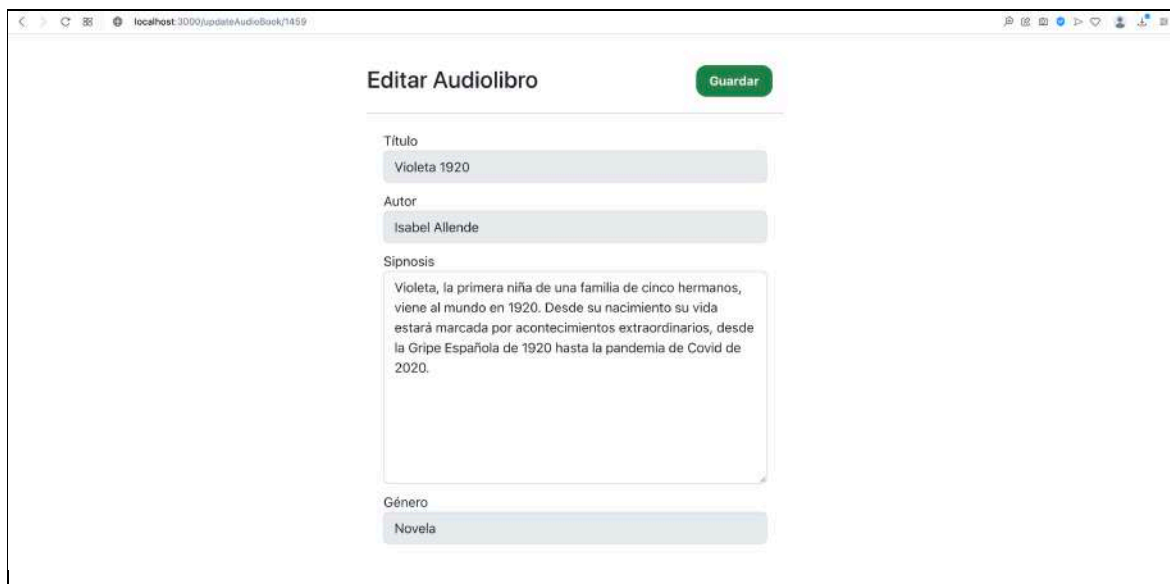
Nota: Los campos título y nombre de autor tienen una validación que admiten un número máximo de caracteres en caso de exceder el aplicativo web no le permite el tipeo de más caracteres.

Una vez que el usuario autor haya llenado los campos deberá seleccionar la opción **“Guardar”**, el aplicativo le redirecciona a la pantalla principal en donde aparecerá el audiolibro agregado recientemente como se puede observar en la siguiente imagen:



Figura 87 – Pantalla principal del usuario autor al agregar un audiolibro desde el aplicativo web

B. Editar un audiolibro: para editar la información de un audiolibro el usuario autor debe seleccionar la opción **“Editar”**, el aplicativo web le redireccionará a una pantalla en donde se habilita los campos que podrá editar como se muestra a continuación:



Género

Imagen


Choose File

Audio
 Choose File

Año de publicación

Figura 88 – Actualización de la información de un audiolibro desde el aplicativo web

Dado que el usuario autor actualice la información del audiolibro debe seleccionar la opción **“Guardar”**, el aplicativo web le redirecciona a la pantalla principal donde podrá observar los cambios realizados como se observa a continuación:

Bienvenido Isabel [Editar perfil](#) [Salir](#)

Mi Biblioteca [Añadir AudioLibro](#)


Titulo	Autor	Imagen	Audio mp3	Género	Año	Sipnosis	Reseña	Acciones
Violeta 1920	Isabel Allende		▶ 0:00 / 4:17	Novela	2022	Violeta, la primera niña de una familia de cinco hermanos, viene al mundo en 1920. Desde su nacimiento su vida estará marcada por acontecimientos extraordinarios, desde la Gripe Española de 1920 hasta la pandemia de Covid de 2020.	Not reviewed	Editar Eliminar

Figura 89 – Pantalla principal del usuario autor después de editar la información de un audiolibro

C. Eliminar un audiolibro: para eliminar un audiolibro el usuario autor debe seleccionar la opción **“Eliminar”** y el audiolibro será eliminado del repositorio. El aplicativo web le mostrará el siguiente mensaje **“El audiolibro fue eliminado exitosamente”**.

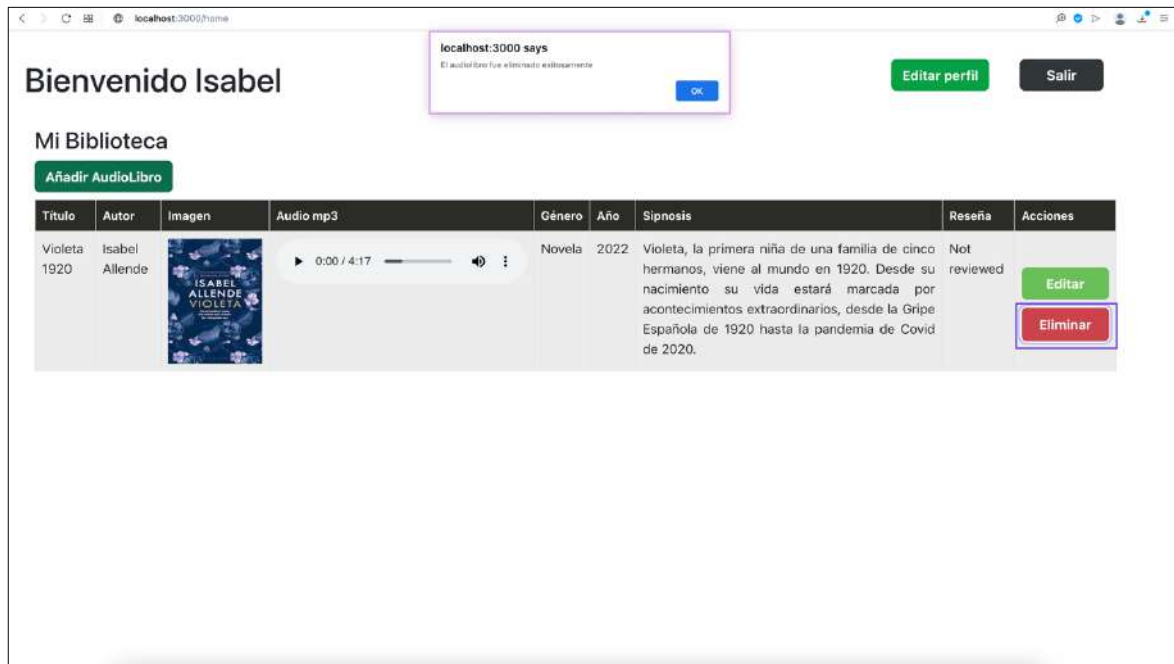


Figura 90– Eliminación de un audiolibro desde el aplicativo web

Cuando el usuario autor elimine el audiolibro, el aplicativo web le redirecciona nuevamente a la pantalla principal.

- **Pruebas no funcionales del Sprint 3**

La prueba de carga [26] se realizó a la interacción del usuario autor donde añade, edite o elimine un audiolibro desde el aplicativo web con la extensión para Chrome de BlazeMeter [47], el cual fue ejecutado con JMeter [46] obteniendo los siguientes resultados:

A. Añadir audiolibro:

- **Grupo de hilos:** para la prueba se configuró el grupo de hilos con 50 usuarios en un periodo de 20 segundos.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Figura 91 – Grupo de hilos para la prueba de carga a la página de añadir audiolibro

- **Árbol de resultados:** desde el cual se observa las peticiones que realiza el usuario cuando añade un nuevo audiolibro desde el aplicativo web.

View Results Tree

Name:

Comments:

Write results to file / Read from file

Filename: Log/Display Only: Errors Successes

Search: Case sensitive Regular exp.

Text

- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook
- ✓ http://localhost:8080/api/audiobook/getAllAudioBoo
- ✓ Test
- ✓ http://localhost:8080/api/audiobook/addAudioBook

Scroll automatically?

Sampler result Request Response data

Request Body Request Headers

1 POST http://localhost:8080/api/audiobook/addAudioBook

2

3 POST data:

4 {"titleAudioBook":"Violeta","author":"Isabel Allende","sinopsis":
"La Épica y emocionante historia de una mujer cuya vida abarca los momentos histÁricos máis relevantes del sigl
o XX.", "gender":"Novela", "urlImage":
"https://firebasestorage.googleapis.com/v0/b/fir-multimedia-storage.appspot.com/o/violeta.jpeg?alt=media&token=ec
9966c7-c138-45a2-9334-7dca7b03a579", "urlAudio":
"https://firebasestorage.googleapis.com/v0/b/fir-multimedia-storage.appspot.com/o/Violeta.mp3?alt=media&token=7d1
c07cd-f454-42c4-b5e3-5a5c9febcbdf", "yearOfPublication":"2022", "userId":"g7vp00pWPMWkLZ5NM4M4yM7nr1"}

5 [no cookies]

6

7

Raw HTTP

Figura 92 – Árbol de resultados de la prueba a la página de añadir audiolibro

- **Informe resumido:** se visualiza que cada petición que el usuario realiza se hace de manera exitosa, obteniendo el siguiente resultado:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://localhost...	50	10	8	30	3.04	0.00%	2.5/sec	2.28	2.58	916.0
http://localhost...	50	58	37	81	11.96	0.00%	2.5/sec	118.82	0.98	47890.0
Test	50	69	46	94	12.72	0.00%	2.5/sec	121.04	3.55	48806.0
TOTAL	150	46	8	94	27.33	0.00%	7.6/sec	242.07	7.11	32537.3

Figura 93 – Informe resumido de la prueba a la página de añadir audiolibro

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 457.108 peticiones por minuto con una media de 53 y una mediana de 46.

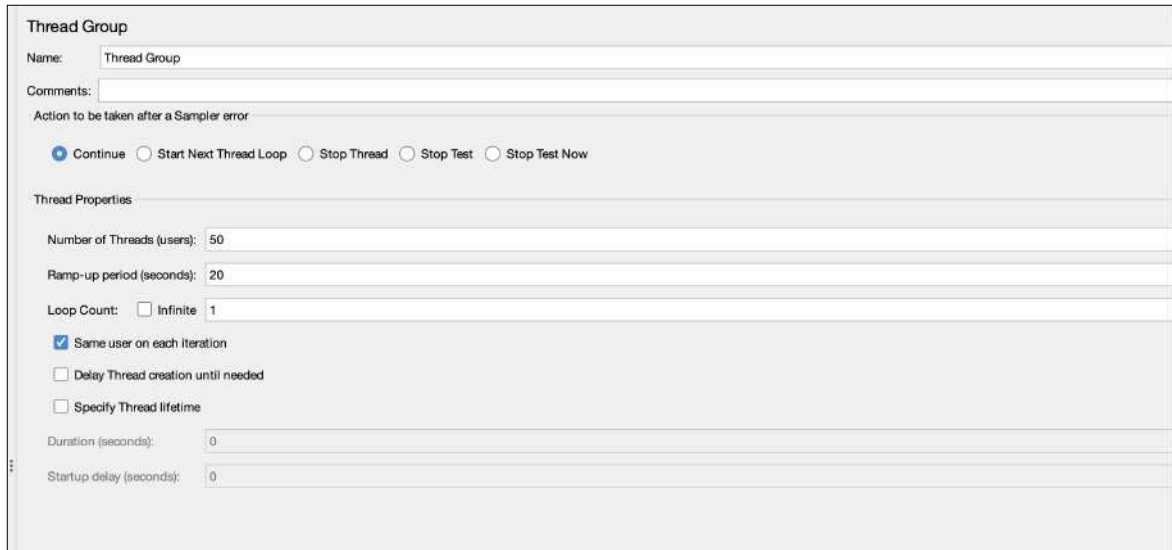


Figura 94 – Gráfico de la prueba de carga a la página de añadir audiolibro

Con los datos obtenidos se llegó a la conclusión el aplicativo web responde correctamente al hilo configurado ya que el valor del tiempo máximo de respuesta total en segundos es de 0.094 y el rendimiento total de 7.6 segundos.

B. Editar audiolibro:

- **Grupo de hilos:** para la prueba se configuro el grupo de hilos con 50 usuarios en un periodo de 20 segundos.



The screenshot shows the 'Thread Group' configuration window in JMeter. The 'Name' field is set to 'Thread Group'. The 'Comments' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users): 50', 'Ramp-up period (seconds): 20', 'Loop Count: 1' (with 'Infinite' unchecked), 'Same user on each iteration' checked, 'Delay Thread creation until needed' unchecked, and 'Specify Thread lifetime' unchecked. The 'Duration (seconds): 0' and 'Startup delay (seconds): 0' fields are also visible.

Figura 95 – Grupo de hilos de la prueba de carga a la página de editar audiolibro

- **Árbol de resultados:** desde el cual se observa las peticiones que realiza el usuario cuando edita un audiolibro desde el aplicativo web.

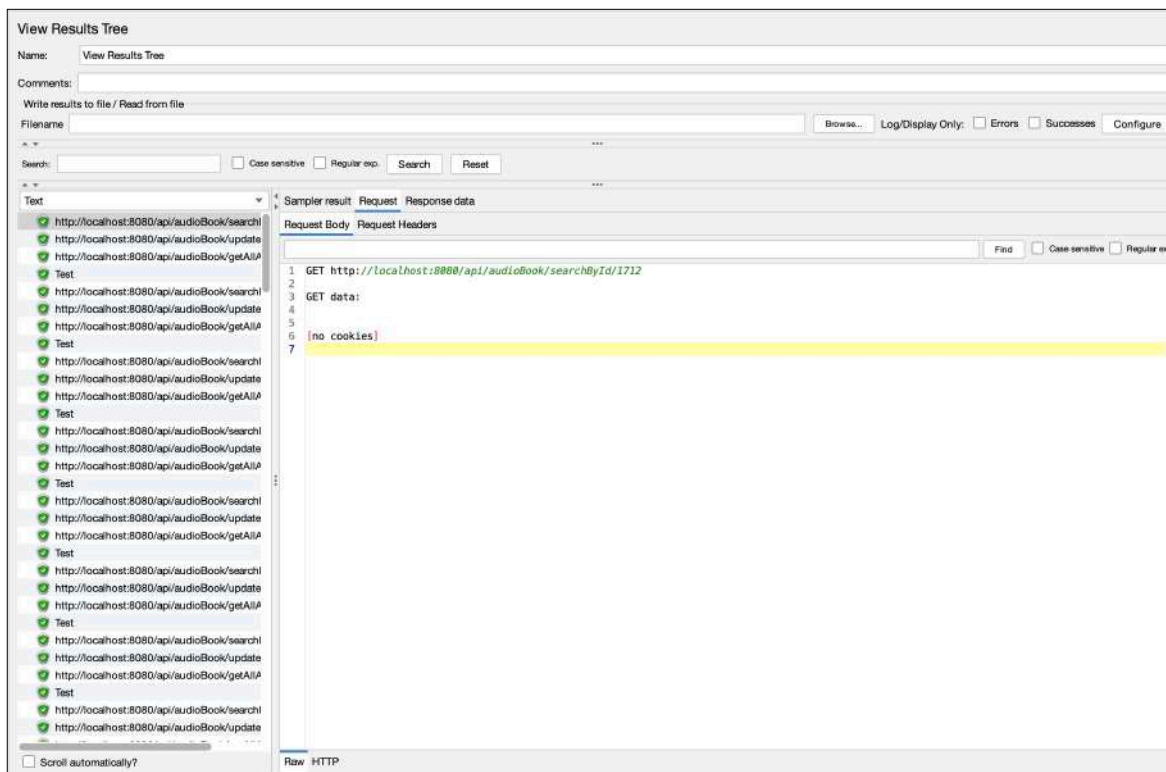


Figura 96 – Árbol de resultados de la prueba a la página de editar audiolibro

- **Informe resumido:** se visualiza que cada petición que el usuario realiza se hace de manera exitosa, obteniendo el siguiente resultado:

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://localhost:80...	50	6	5	8	0.74	0.00%	2.5/sec	2.29	0.99	918.0
http://localhost:80...	50	2	2	4	0.65	0.00%	2.6/sec	0.52	2.72	210.0
http://localhost:80...	50	3	2	5	0.64	0.00%	2.6/sec	2.29	0.99	920.0
Test	50	12	10	15	1.09	0.00%	2.5/sec	5.10	4.70	2048.0
TOTAL	200	6	2	15	3.96	0.00%	10.2/sec	10.20	9.39	1024.0

Figura 97 – Árbol de resultados de la prueba a la página de editar audiolibro

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 611.777 peticiones por minuto con una media de 5 y una mediana de 6.



Figura 98 – Gráfico de la prueba de carga a la página de editar audiolibro

Con los datos obtenidos se llegó a la conclusión el aplicativo web responde correctamente al hilo configurado ya que el valor del tiempo máximo de respuesta total en segundos es de 0.015 y el rendimiento total de 10.2 segundos.

C. Eliminar audiolibro:

- **Grupo de hilos:** para la prueba se configuro el grupo de hilos con 50 usuarios en un periodo de 20 segundos.

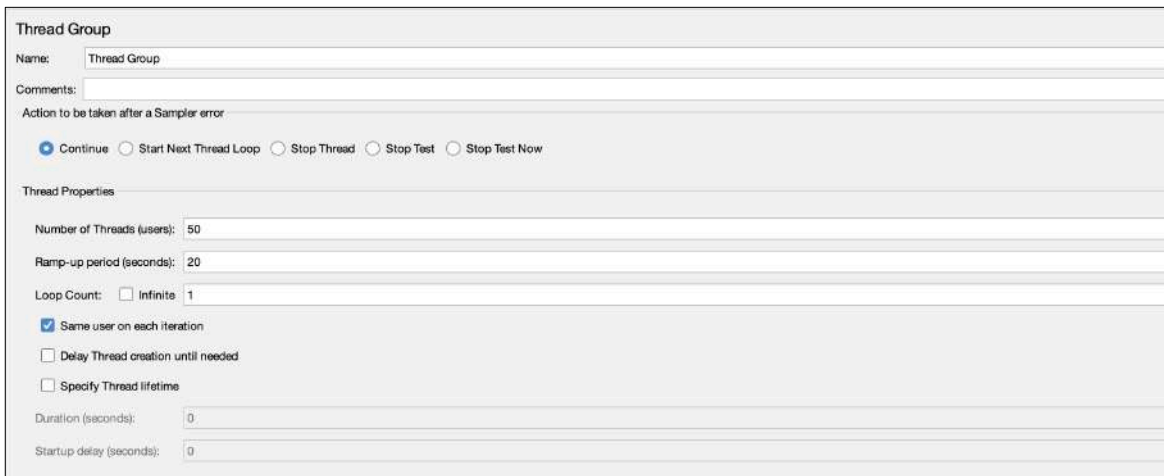


Figura 99 – Grupo de hilos de la prueba a la página de eliminar audiolibro

- **Árbol de resultados:** desde el cual se observa las peticiones que realiza el usuario cuando edita un audiolibro desde el aplicativo web.

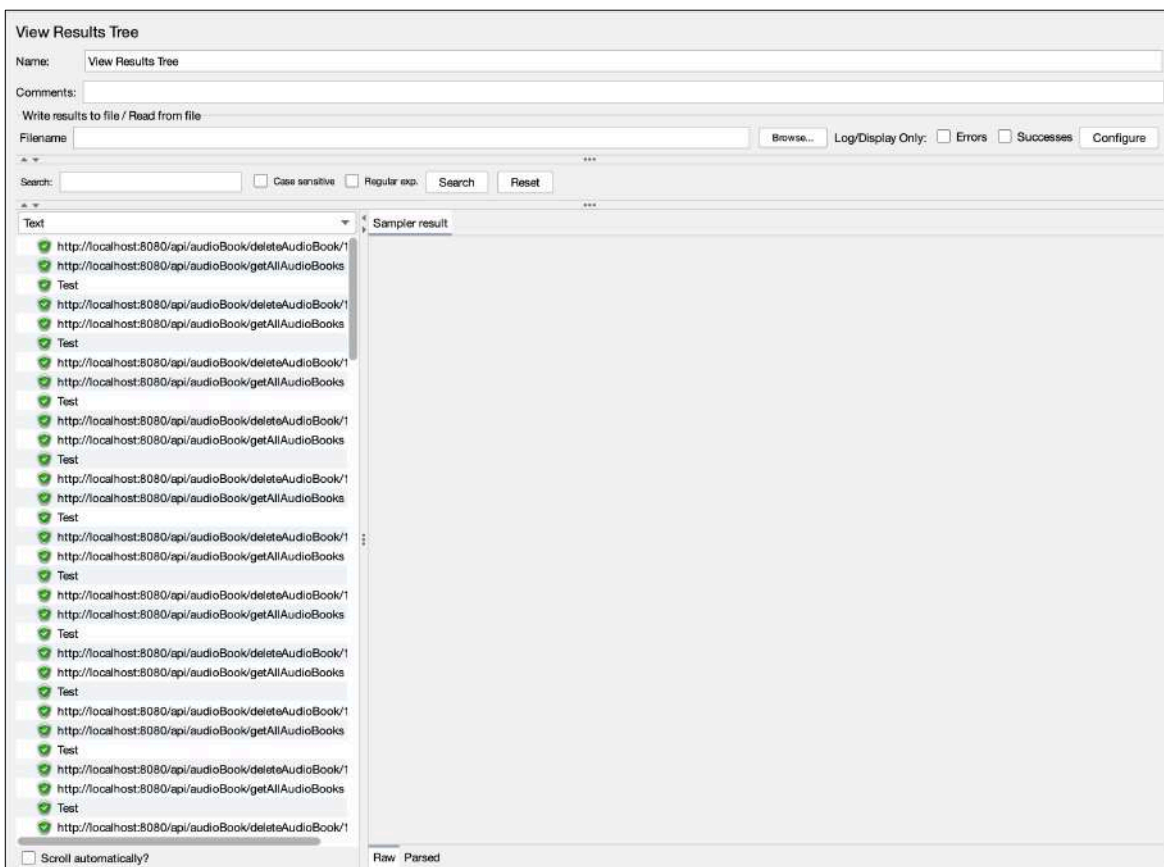


Figura 100 – Árbol de resultados de la prueba a la página de eliminar audiolibro

- **Informe resumido:** se visualiza que cada petición que el usuario realiza se hace de manera exitosa, obteniendo el siguiente resultado:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://localhost:...	50	4	3	26	3.08	0.00%	2.6/sec	0.52	1.21	210.0
http://localhost:...	50	2	2	3	0.50	0.00%	2.6/sec	0.66	0.99	263.0
Test	50	7	5	29	3.19	0.00%	2.5/sec	1.18	2.20	473.0
TOTAL	150	4	2	29	3.24	0.00%	7.6/sec	2.36	4.40	315.3

Figura 101 – Árbol de resultados de la prueba a la página de eliminar audiolibro

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 458.95 peticiones por minuto con una media de 4 y una mediana de 4.

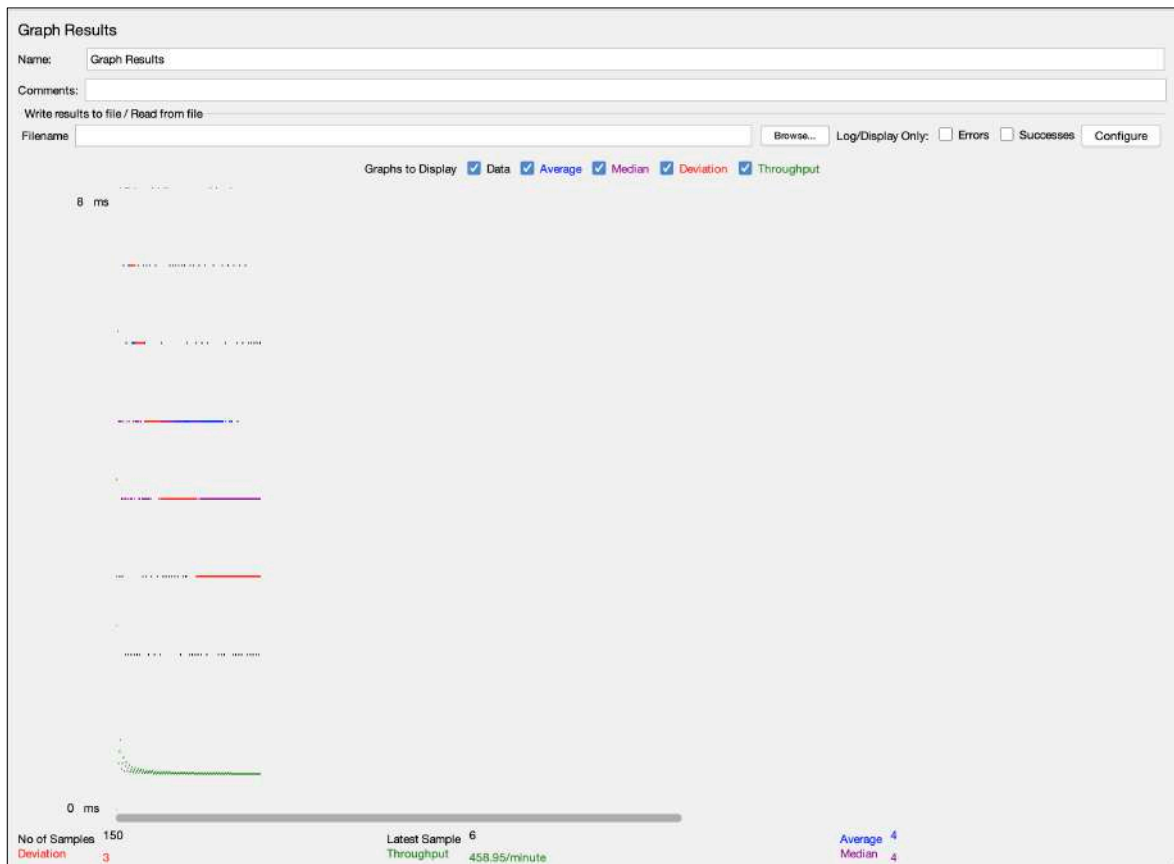


Figura 102 – Gráfico de la prueba de carga a la página de eliminar audiolibro

Con los datos obtenidos se llegó a la conclusión el aplicativo web responde correctamente al hilo configurado ya que el valor del tiempo máximo de respuesta total en segundos es de 0.029 y el rendimiento total de 7.6 segundos.

4.4. Ejecución de pruebas del Sprint 4

- **Pruebas funcionales del Sprint 4**

Una vez que se concluyó la programación del microservicio se realizaron las pruebas correspondientes para verificar que las peticiones cumplan con las historias de usuario a continuación se detalla las peticiones realizadas:

A. Crear una lista de reproducción: permite crear una nueva lista de reproducción en la base de datos, se realizó una petición POST que recibe un objeto de tipo playList con los atributos correspondientes a la lista de reproducción.

MÉTODO	POST
URL	http://localhost:9090/api/playList/createPlayList
BODY	<pre>{ "namePlayList": "Novelas", "userId": "Y2wZmG94YGcj1f4q0D8G63lYZxX2" }</pre>
RESPONSE	

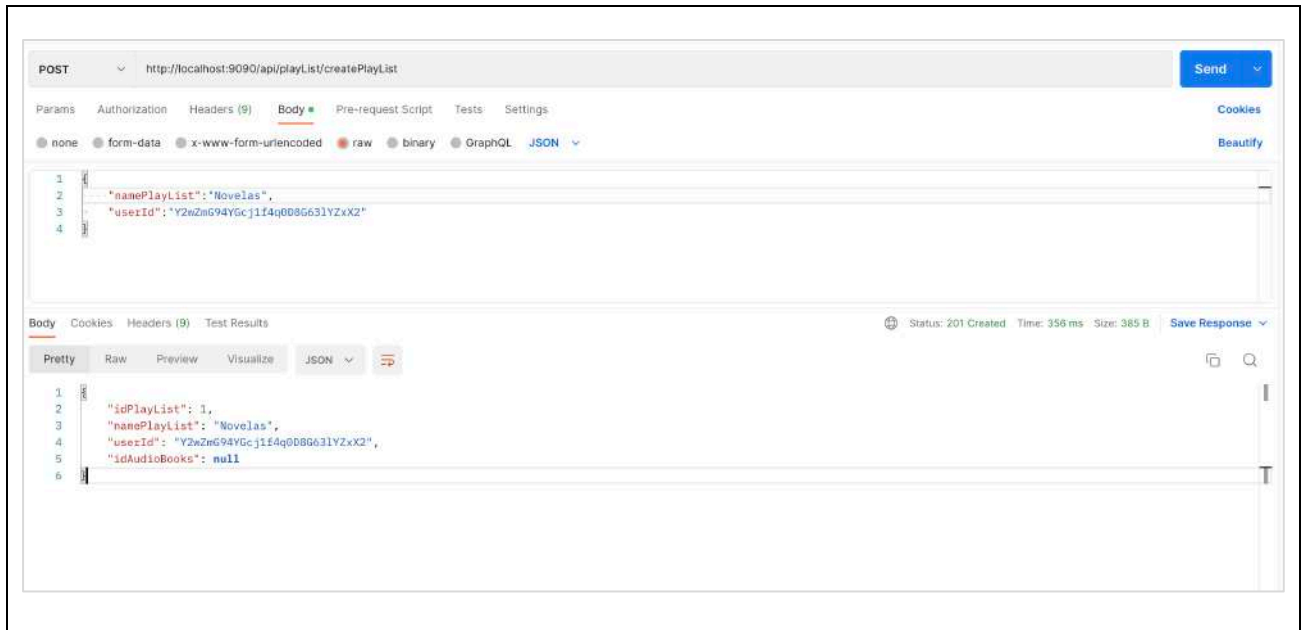


Tabla 44 - Petición POST para crear una lista de reproducción

Como se observa en la **Tabla 44** la lista de reproducción fue creada exitosamente.

B. Agregar audiolibros a una lista de reproducción: permite agregar audiolibros a una lista de reproducción, la petición POST recibe como variable el id de la lista de reproducción y el id de audiolibro que será agregado.

MÉTODO	POST
URL	http://localhost:9090/api/playList/addAudioBookToPlayList/1
VARIABLES	<ul style="list-style-type: none"> 1
BODY	<pre> { "idAudioBook": "8" } </pre>
Response	

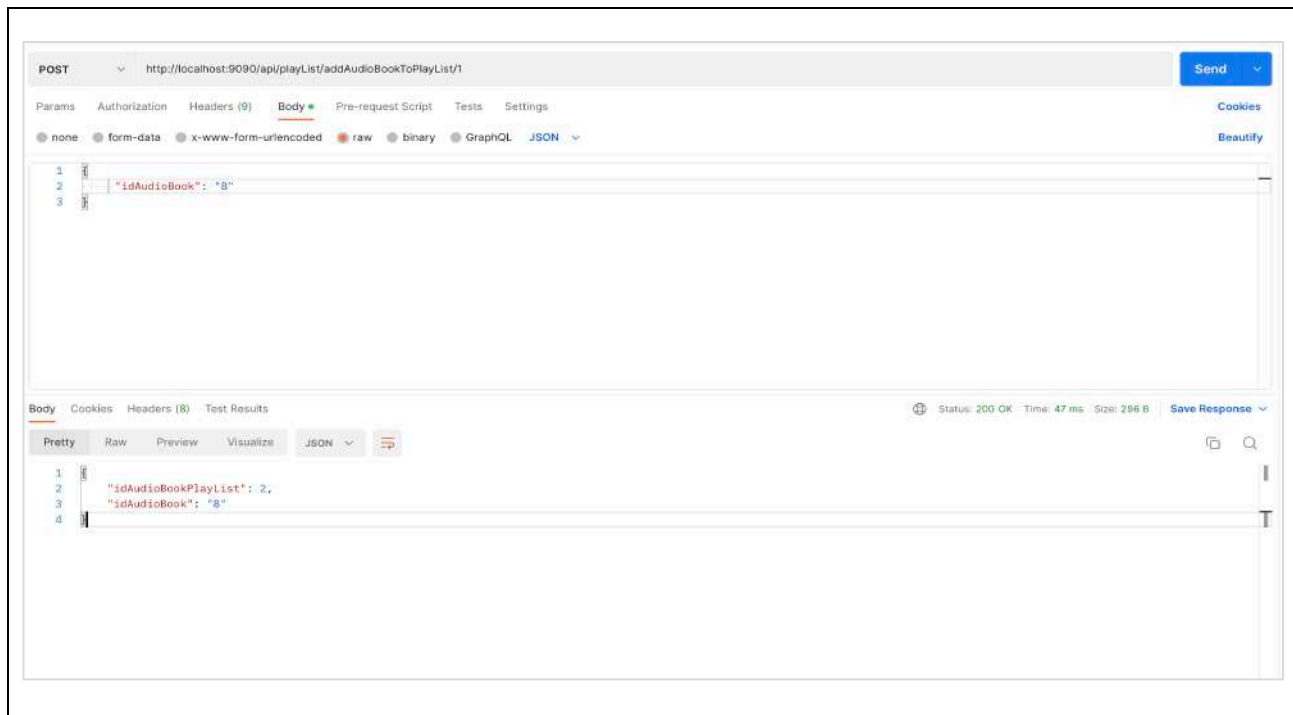


Tabla 45 - Petición POST para agregar un audiolibro a una lista de reproducción

Para comprobar que el audiolibro se agregó correctamente a la lista de reproducción creada anteriormente se realizó una petición GET para obtener todas las listas de reproducción:

MÉTODO	GET
URL	http://localhost:9090/api/playList/getAllPlayLists
RESPONSE	

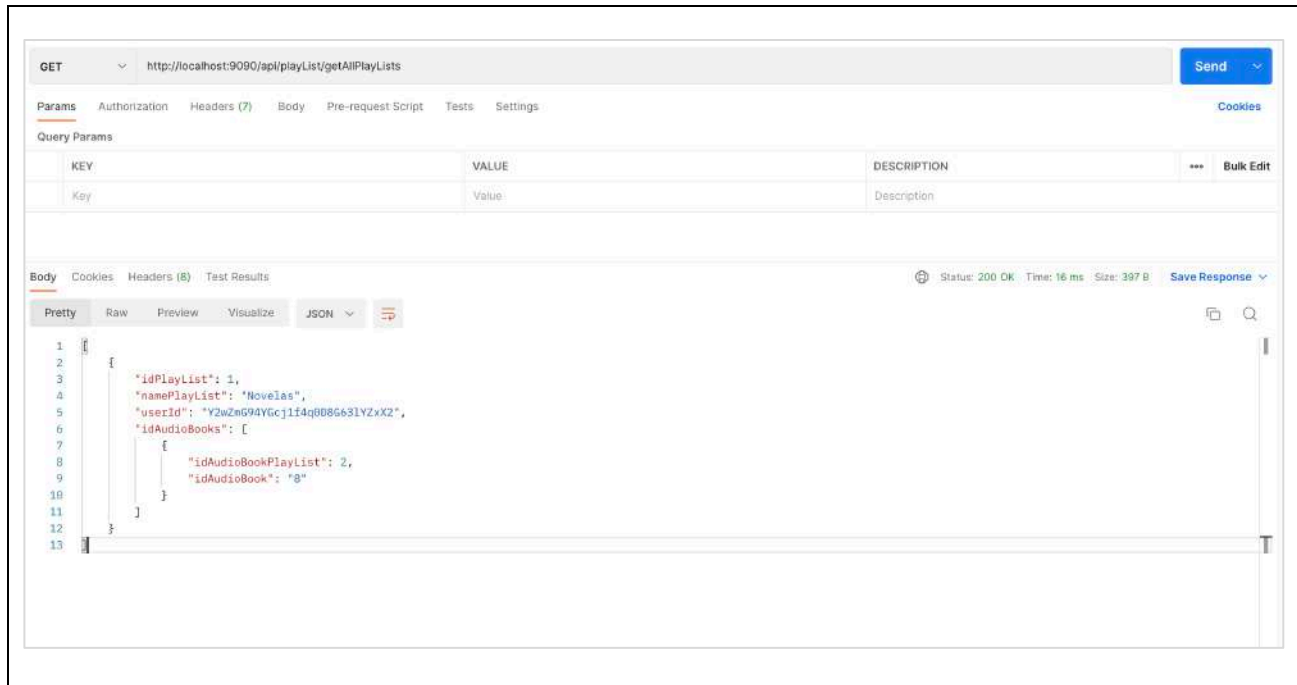


Tabla 46 - Petición GET para obtener las listas de reproducción

- **Pruebas no funcionales del Sprint 4**

Las pruebas de rendimiento [26] se las realizó al microservicio **UserPlaylistMicroservice**, para las pruebas se utilizó la herramienta JMeter [46], a continuación, se detalla el proceso de cada plan de prueba realizado:

- A. Crear lista de reproducción:** se realizó la configuración del hilo en donde se indicó que la prueba soporte 100 usuarios cada 10 segundos.

Figura 103 - Configuración del hilo para la prueba en JMeter de la petición Create Playlist

Se agregó una respuesta HTTP de tipo POST y el objeto de tipo playList que deberá ser enviado durante las peticiones.

Figura 104 - Configuración de la respuesta Create Playlist

Como la petición espera recibir un objeto en formato JSON, a nuestra prueba se agregó un gestor de cabeceras HTTP (HTTP Header Manager).

Name	Value
Content-Type	application/json

Figura 105 - HTTP Header Manager para la petición Create Playlist

A continuación, se detallan los resultados obtenidos:

- **Informe resumido:** donde se visualizó que la prueba de realizó fue exitosa ya que durante la ejecución no dio ningún error, llegando a la conclusión que la petición soporta los 100 usuarios cada 10 segundos.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Create Playlist	100	11	9	22	1.42	0.00%	10.1/sec	3.85	2.65	390.9
TOTAL	100	11	9	22	1.42	0.00%	10.1/sec	3.85	2.65	390.9

Figura 106 - Summary Report de la petición Create Playlist

- **Árbol de resultados:** desde el cual se observa las peticiones que se realizan logrando visualizar los datos que se envía y la respuesta que recibe cada una de ellas.

```

1 POST http://localhost:9090/api/playlist/createPlaylist
2
3 POST data:
4 {
5   "namePlaylist": "Novelas",
6   "userId": "Y2w2mG94Y6cJ1f4q086631Y2xx2"
7 }
8
9 [no cookies]
10
  
```

Figura 107 - Árbol de resultados de la prueba de la petición Create Playlist

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 605.327 peticiones por minuto con una media de 11 y una mediana de 11.

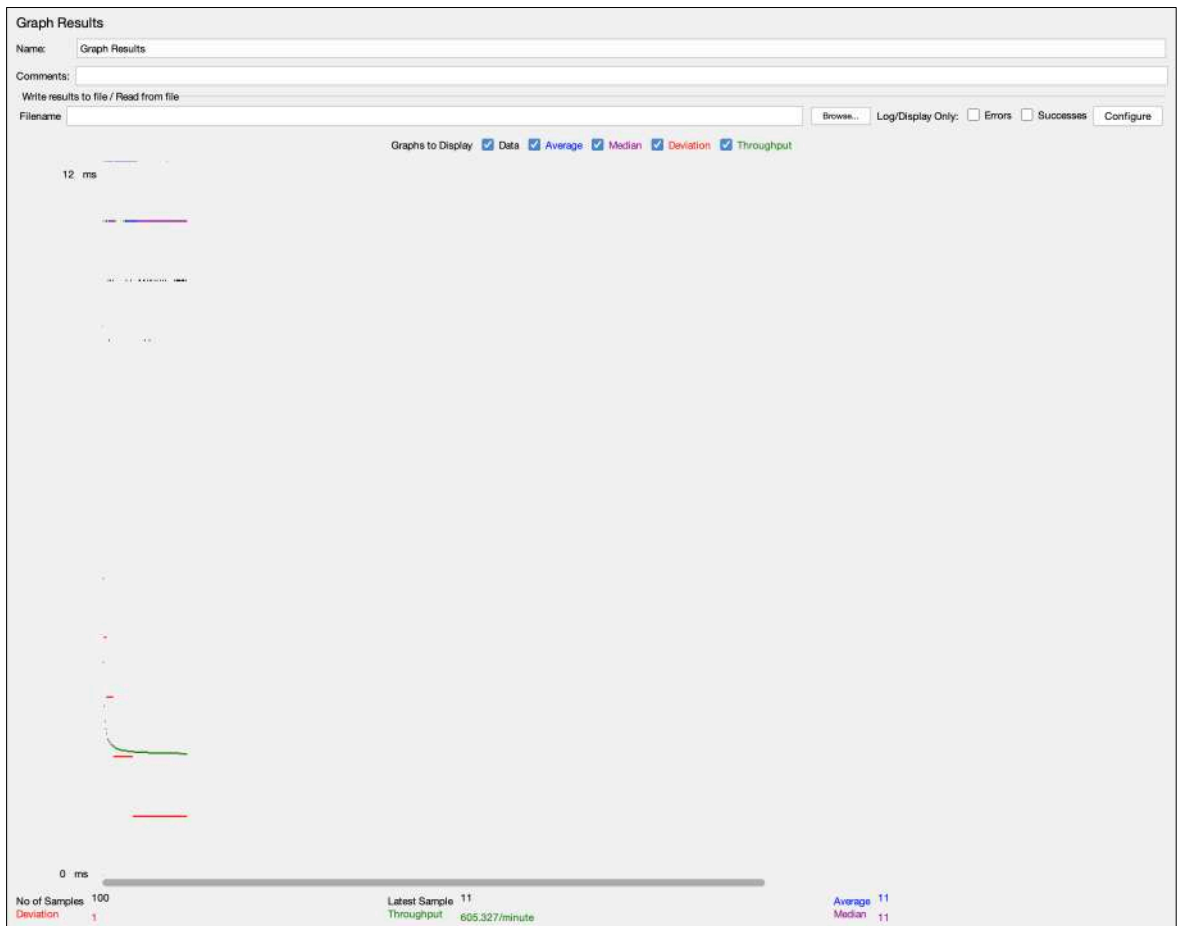


Figura 108 – Gráfico de los resultados de la prueba para la petición Create Playlist

Con los datos obtenidos se llegó a la conclusión que el servidor del microservicio **“UserPlaylistMicroservice”** responde correctamente al hilo configurado ya que el valor del tiempo máximo de respuesta en segundos es de 0.022 y el rendimiento total de 10.1 segundos.

B. Agregar audiolibros a una lista de reproducción: se realizó la configuración del hilo en donde se indicó que la prueba soporte 100 usuarios cada 10 segundos.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error:

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Figura 109 - Configuración del hilo para la prueba de la petición Add AudioBook To PlayList

Se agregó una respuesta HTTP de tipo GET

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: Port Number:

HTTP Request

GET Path: Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals

Figura 110 - Configuración de la respuesta Add AudioBook To PlayList

A continuación, se detallan los resultados obtenidos:

- **Informe resumido:** donde se visualizó que la prueba de realizó fue exitosa ya que durante la ejecución no dio ningún error, llegando a la conclusión que la petición soporta los 100 usuarios cada 10 segundos.

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configura

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ADD AUDIOBOOK TO..	100	136	58	1106	206.86	0.00%	10.0/sec	102.43	1.37	10466.0
TOTAL:	100	136	58	1106	206.86	0.00%	10.0/sec	102.43	1.37	10466.0

Figura 111 - Summary Report de la petición Add AudioBook To PlayList

- **Árbol de resultados:** desde el cual se observa las peticiones que se realizan logrando visualizar los datos que se envía y la respuesta que recibe cada una de ellas.

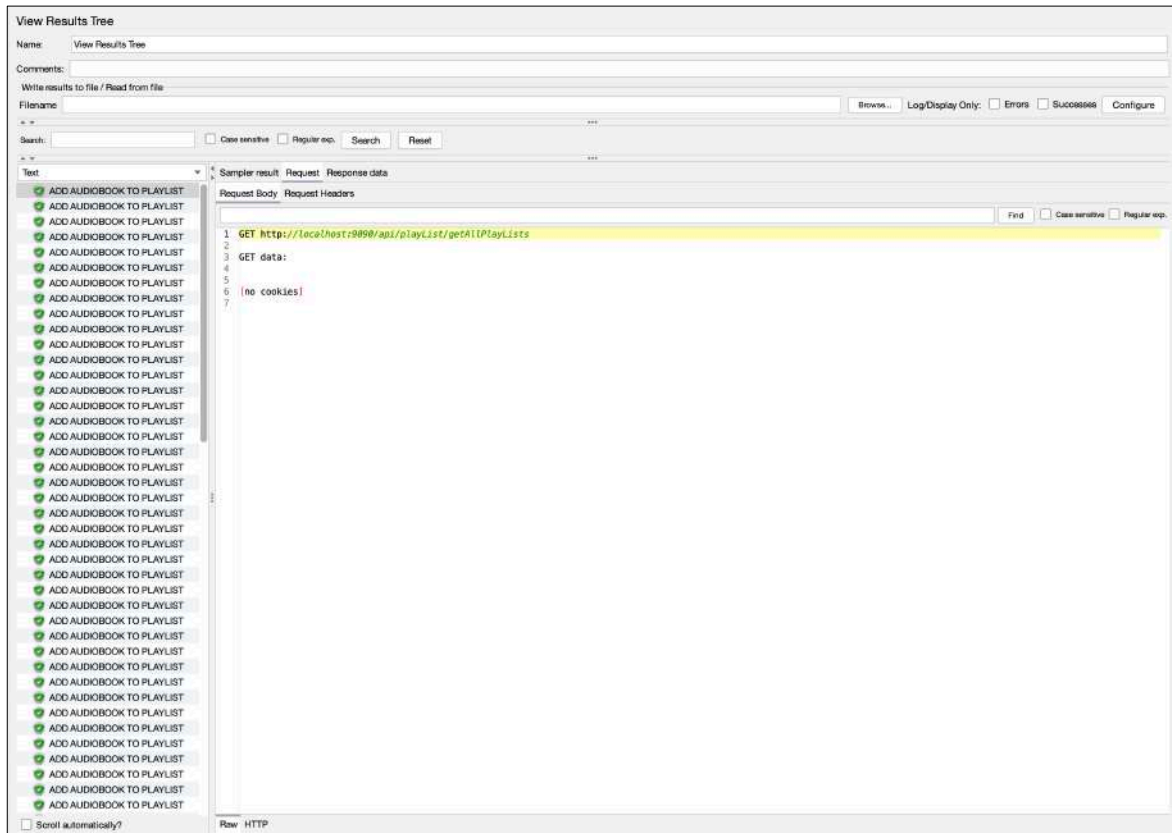


Figura 112 - Árbol de resultados de la prueba de la petición Add AudioBook To PlayList

- **Gráfico:** el gráfico traza el tiempo de respuesta en milisegundos y el tiempo transcurrido, el número de solicitudes que fue capaz de procesar fueron 601.323 peticiones por minuto con una media de 72 y una mediana de 136.

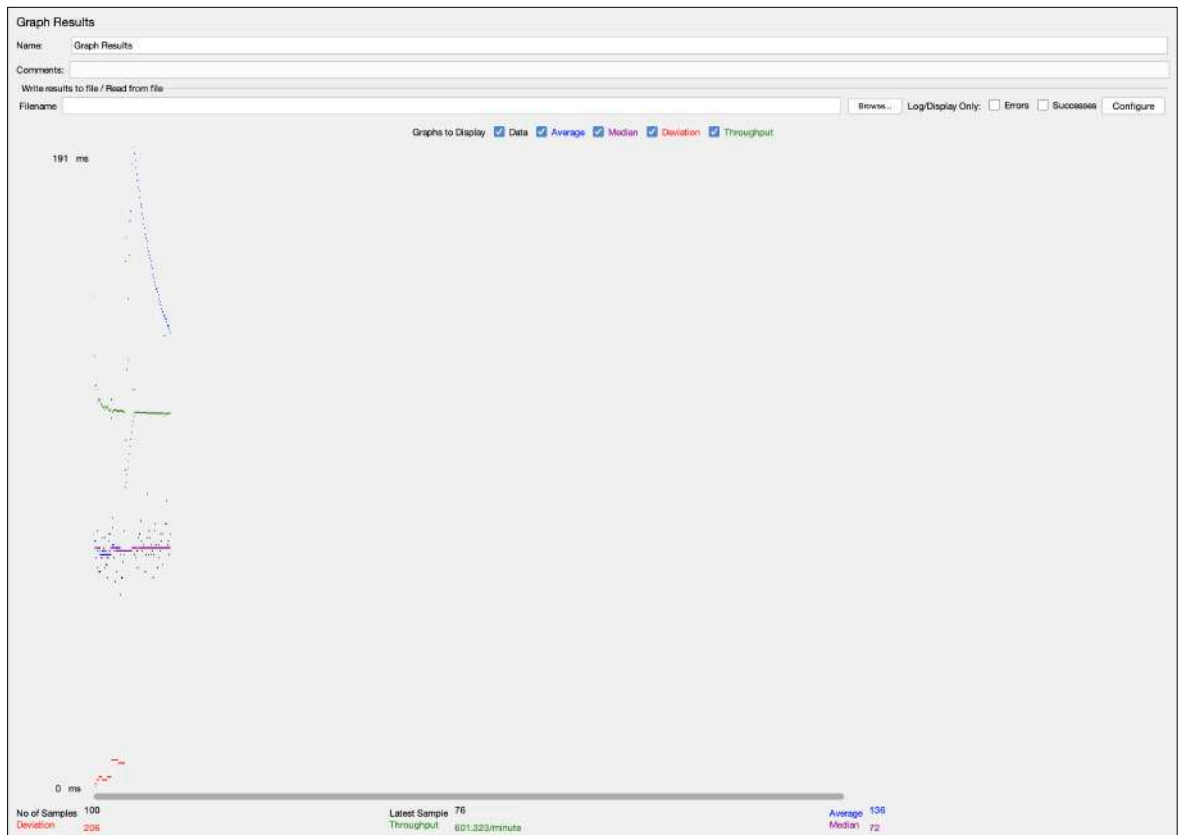


Figura 113 – Gráfico de los resultados de la prueba para la petición Add AudioBook To Playlist

Con los datos obtenidos se llegó a la conclusión que el servidor del microservicio **“UserPlaylistMicroservice”** responde correctamente al hilo configurado ya que el valor del tiempo de respuesta en segundos es de 1.106 y el rendimiento total de 10.0 segundos.

4.5. Ejecución de pruebas del Sprint 5

- **Pruebas funcionales del Sprint 5**

Al finalizar con la ejecución del Sprint se realizaron las pruebas de aceptación [25] a cada historia de usuario para la autenticación y registro del usuario final, a continuación, se detallan los resultados obtenidos:

A. Creación de cuentas con usuario y contraseña desde la aplicación

móvil: donde el usuario de tipo final crea una cuenta, cumpliendo los siguientes criterios de aceptación:

- El email debe estar en el formato xxx@xxx.com
- La contraseña contendrá como máximo 8 caracteres.

- El nombre de usuario contendrá como máximo 8 caracteres.

Cuando el usuario ingrese a la aplicación móvil y seleccione la opción “**¿Es nuevo aquí? Cree una cuenta**”, se redireccionará a la pantalla de registro, en la cual debe llenar los campos solicitados de manera correcta, si el usuario no ingresa los datos o no cumplen con los criterios de aceptación la aplicación móvil no le permitirá el registro como se observa en la siguiente imagen:

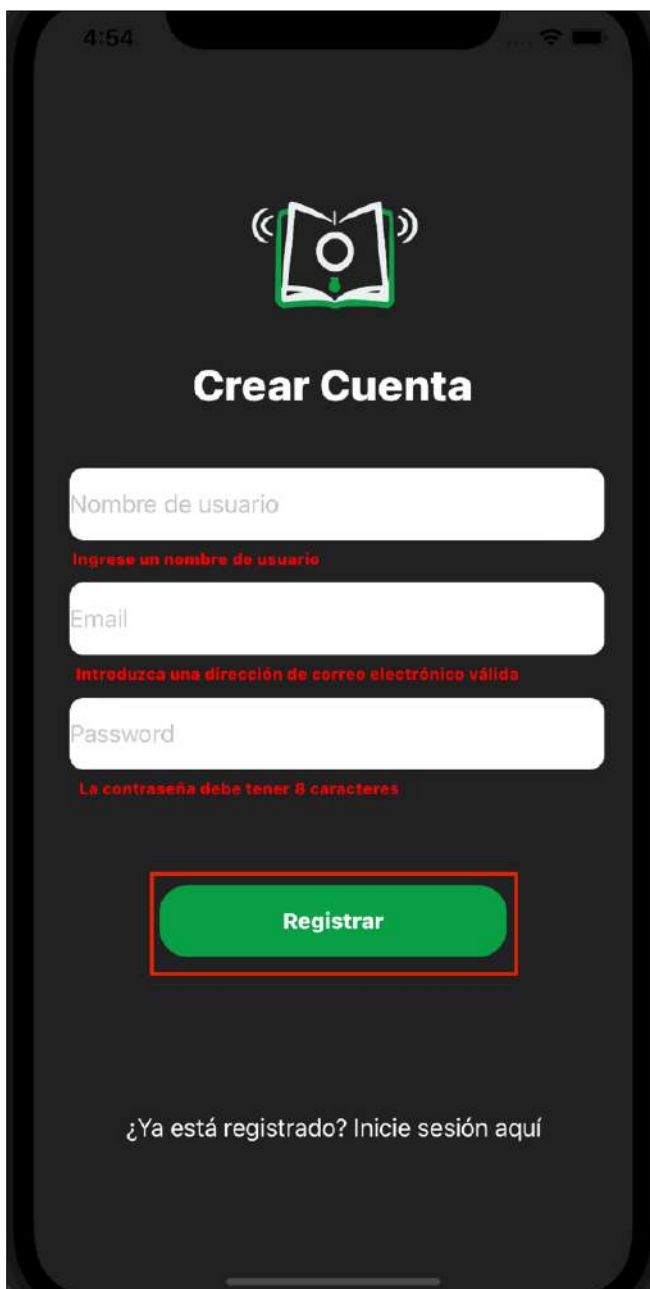


Figura 114 – Registro de usuario final inválido desde la aplicación móvil

Si el usuario ingresa correctamente los campos solicitados la aplicación móvil le permite el registro y le mostrará el siguiente mensaje **“Registro completado exitosamente”**.



Figura 115– Registro de usuario final válido desde la aplicación móvil

Cuando el usuario final se encuentre registrado la aplicación móvil le redireccionara a la pantalla principal como se muestra en la siguiente imagen:



Figura 116 – Pantalla principal de la aplicación móvil

- B. Creación de cuentas con Google desde la aplicación móvil:** donde el usuario de tipo final puede crear una cuenta con su correo de Google, seleccionando la opción **“Ingresa con Google”**, además de otorgarle los permisos correspondientes como se muestra a continuación:

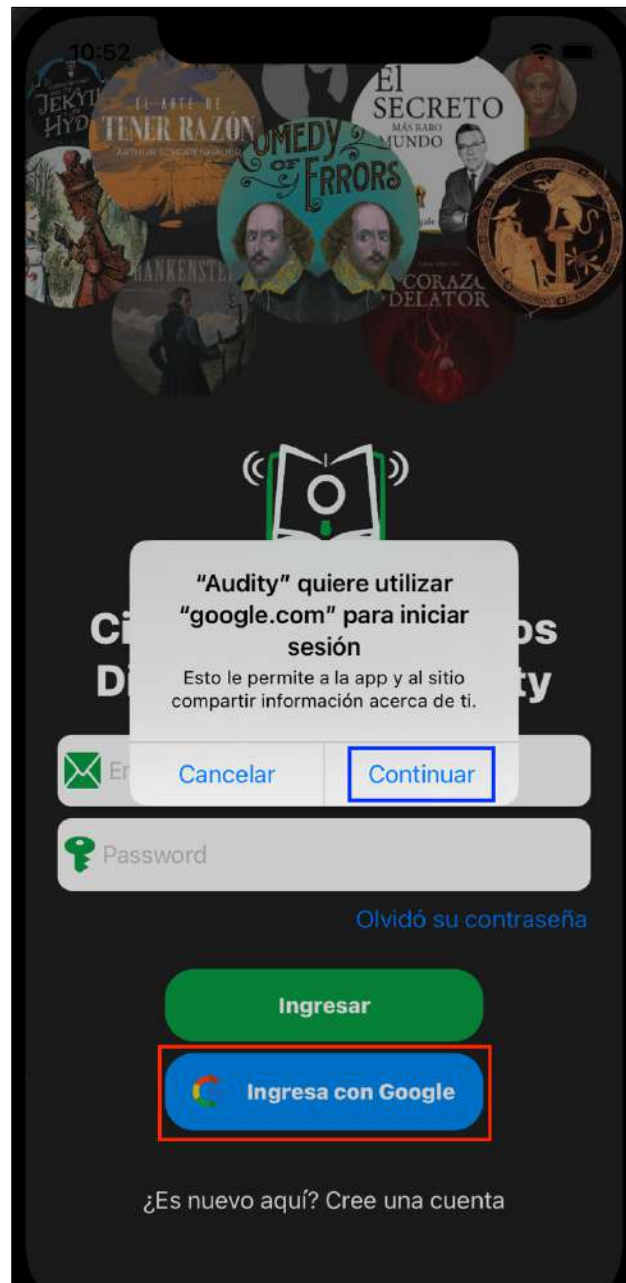


Figura 117 - Inicio de sesión con Google desde la aplicación móvil

El usuario final debe elegir la cuenta que desea registrar:

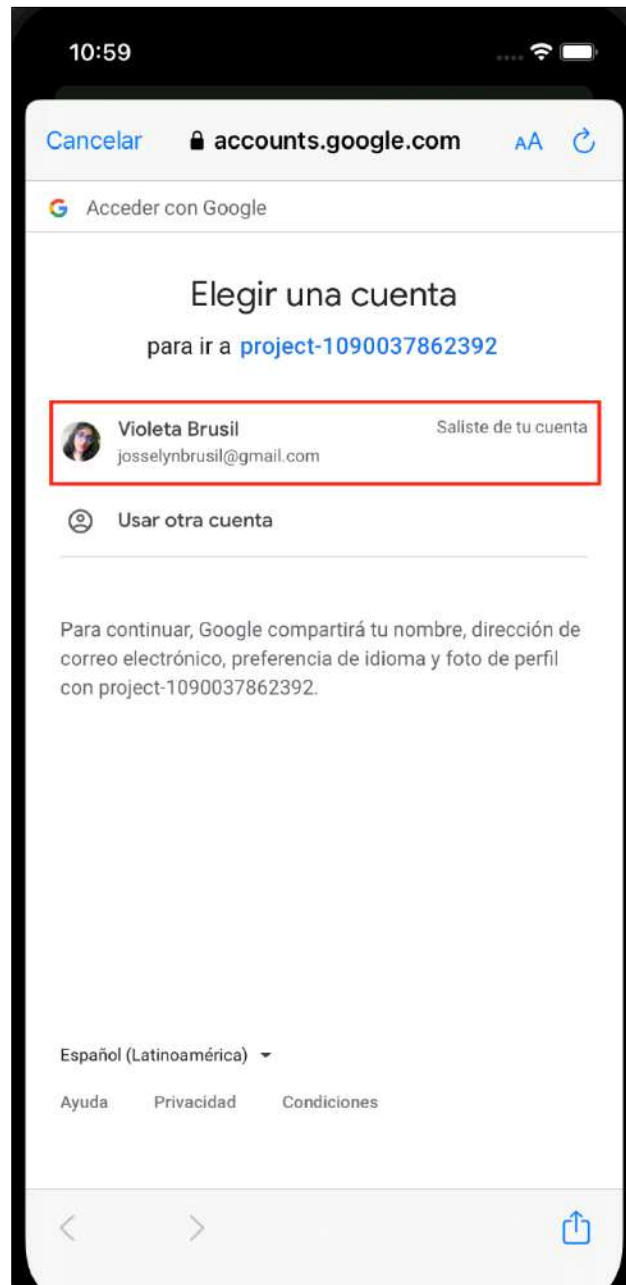


Figura 118 – Selección de cuenta de Google desde la aplicación móvil para el registro

Una vez que el usuario elija la cuenta a registrar deberá ingresar la contraseña y finalizar el registro, la aplicación móvil le redireccionará a la pantalla principal como se observa a continuación:

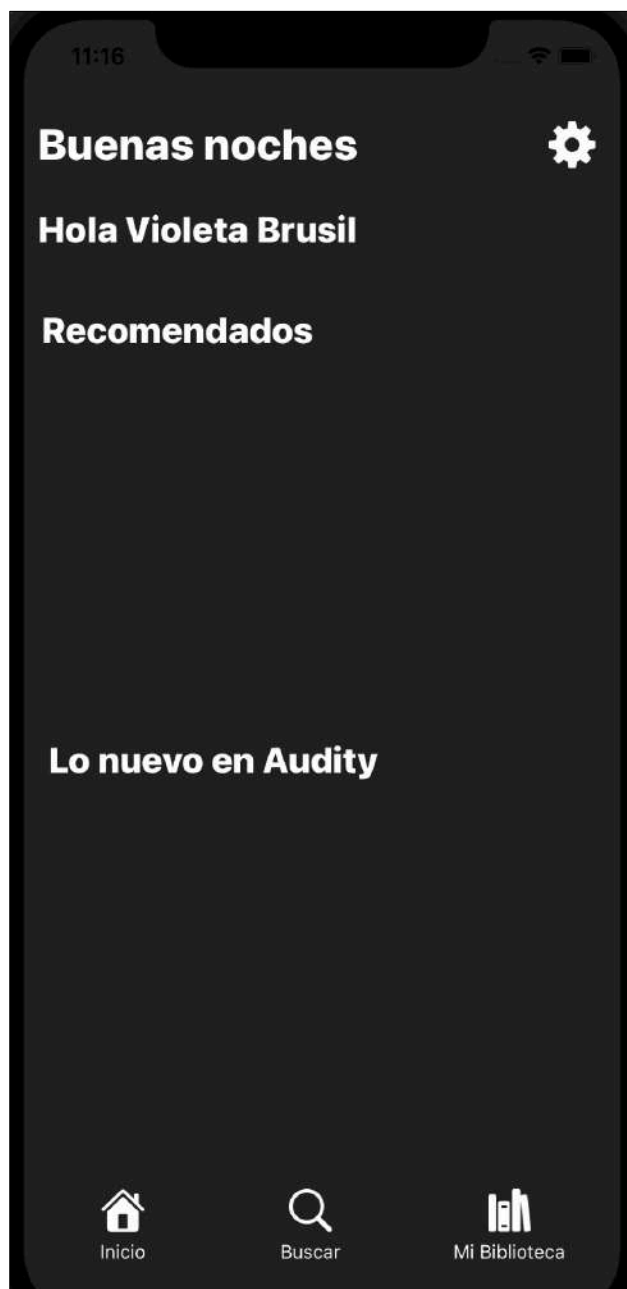


Figura 119 – Pantalla principal de la aplicación móvil ingresando con la cuenta de Google

C. Autenticación de usuarios desde la aplicación móvil: dado el usuario se registró de manera exitosa podrá ingresar al aplicativo web con su email y contraseña desde la pantalla de inicio de sesión. Si el usuario selecciona la opción **“Ingresar”** sin llenar los campos correspondientes a email y contraseña, la aplicación móvil le indicará que son requeridos para iniciar sesión, como se observa a continuación:

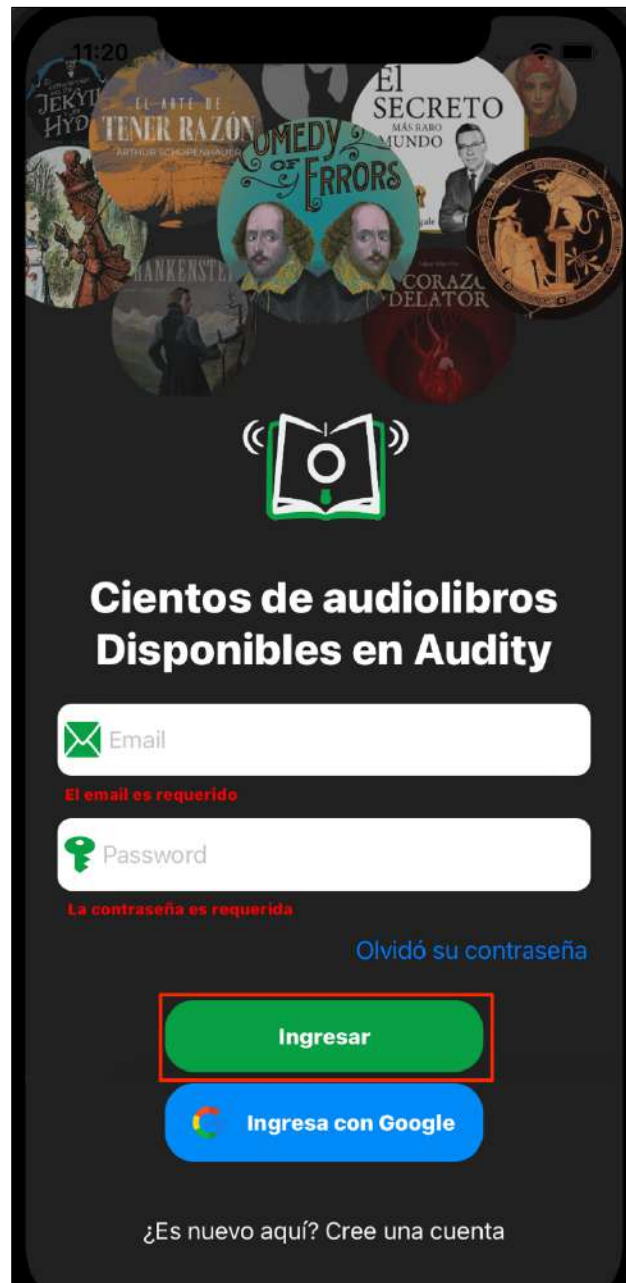


Figura 120 – Inicio de sesión del usuario final cuando no ingresa los campos solicitados

Si el usuario ingresa los datos correctamente la aplicación móvil le permitirá el ingreso, siempre y cuando ya se encuentre registrado, como se observa a continuación:

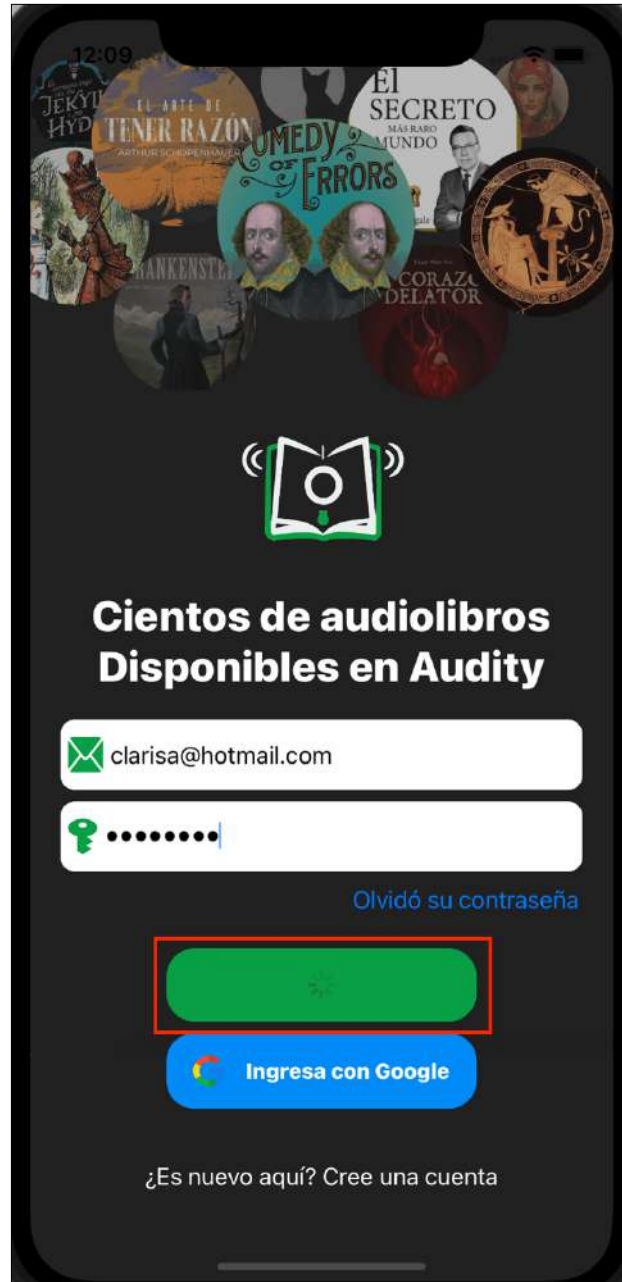


Figura 121 – Pantalla de inicio de sesión cuando el usuario final ingresa los campos solicitados

En caso de que el usuario ingrese incorrectamente su email o contraseña la aplicación móvil le mostrara un mensaje de error negándole el ingreso:

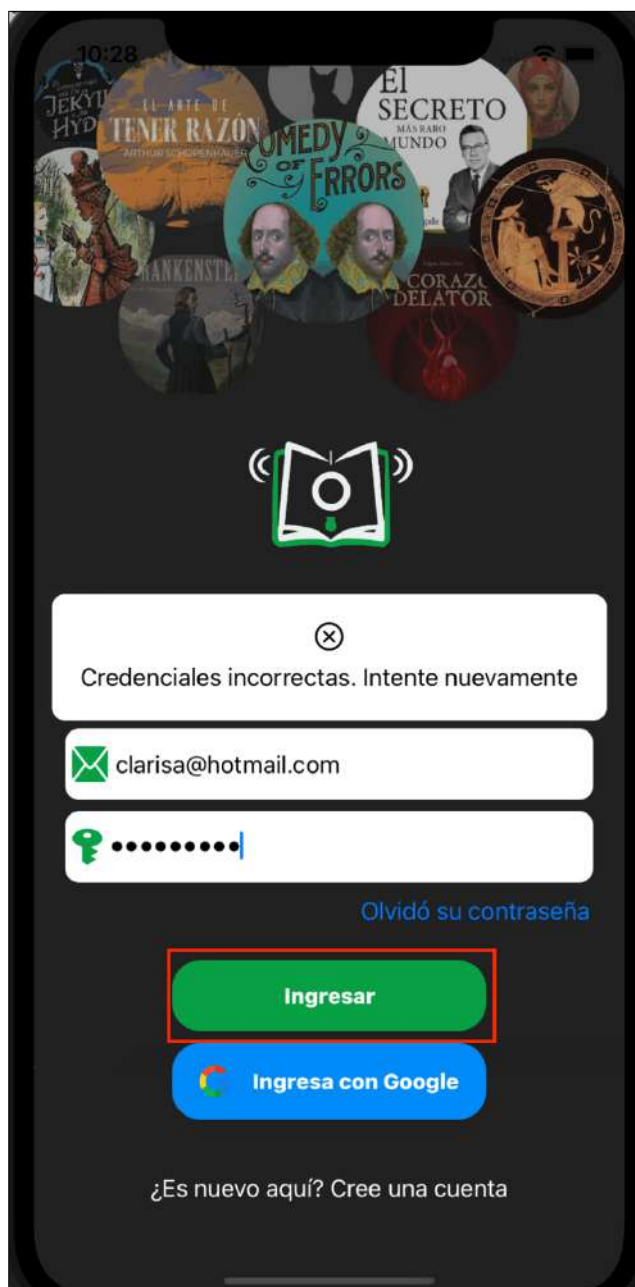


Figura 122 – Inicio de sesión exitoso desde la aplicación móvil

D. Editar información personal desde la aplicación móvil: dado que el usuario seleccione el icono de **“Configuración”**, la aplicación móvil lo redirecciona a la pantalla de configuración, en la cual se puede observar la información personal del usuario:

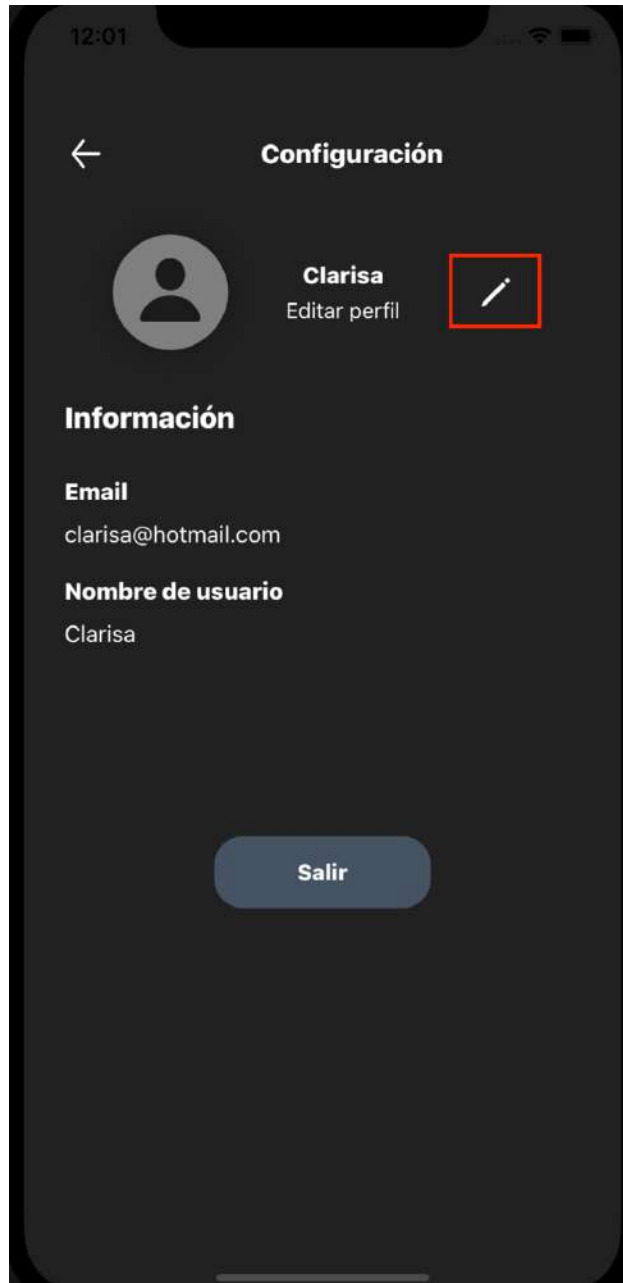


Figura 123 – Pantalla de configuración de la aplicación móvil

El usuario debe seleccionar el icono de **“Editar”** como se observa en la **Figura 123**, la aplicación móvil le redireccionará a la pantalla de editar perfil donde podrá cambiar el nombre de usuario o su foto de perfil. Una vez que el usuario haya modificado sus datos seleccionará el icono **“Guardar”** y se le presentará el siguiente mensaje:

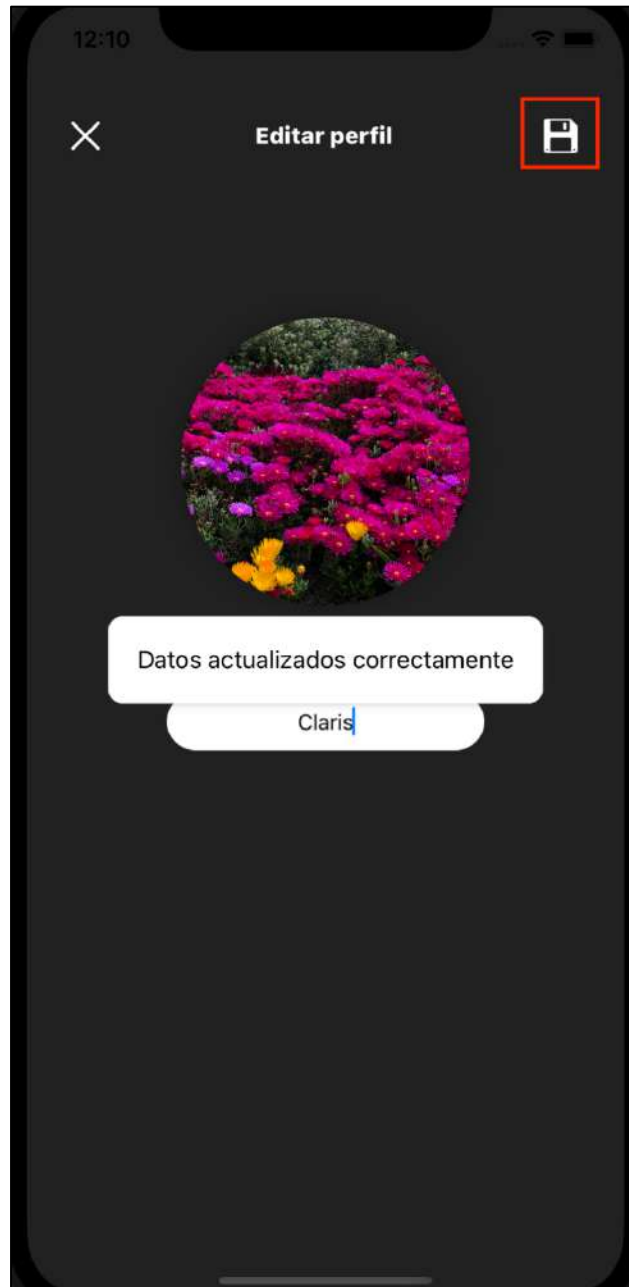


Figura 124 – Mensaje informativo cuando el usuario final edita su información personal

Inmediatamente la aplicación móvil le redirecciona nuevamente a la pantalla anterior donde se podrá observar los datos actualizados como se observa a continuación:

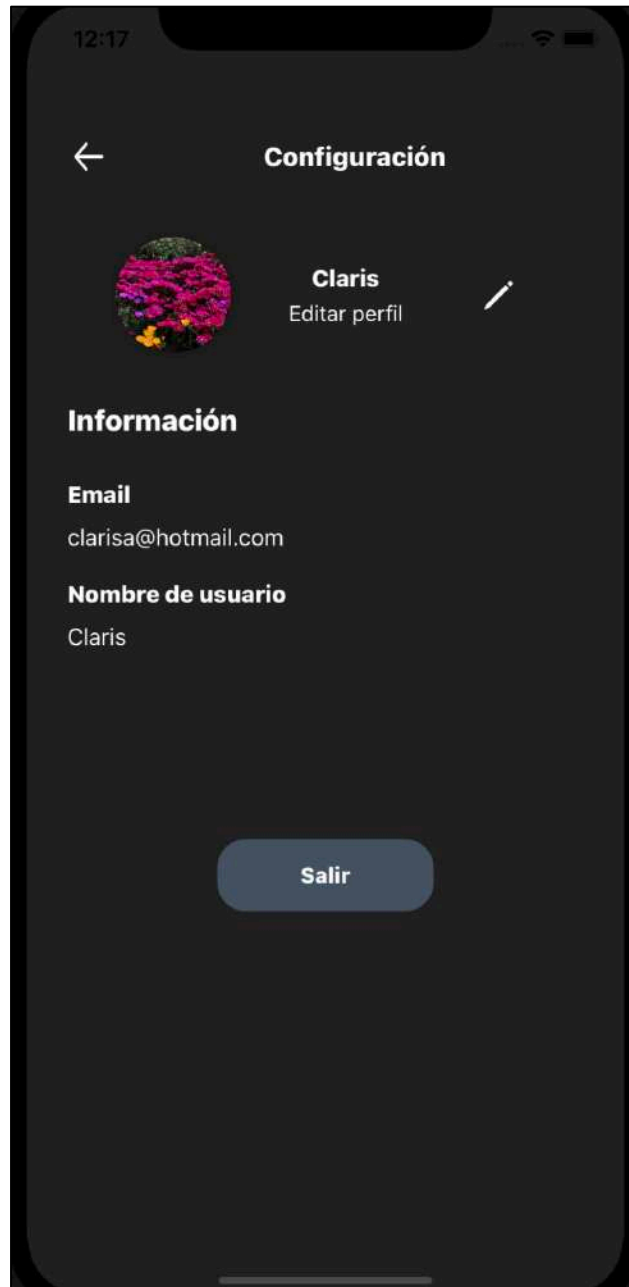


Figura 125 – Pantalla de configuración con la información actualizada

- **Pruebas no funcionales del Sprint 5**

La prueba de rendimiento [26] se realizó con la herramienta Apptim [40], la cual permite obtener un análisis sobre el rendimiento de la aplicación móvil en tiempo real, para empezar con la prueba se siguieron los siguientes pasos:

A. Iniciar una nueva prueba: al abrir la herramienta se seleccionó la opción **“Start a new test”**.

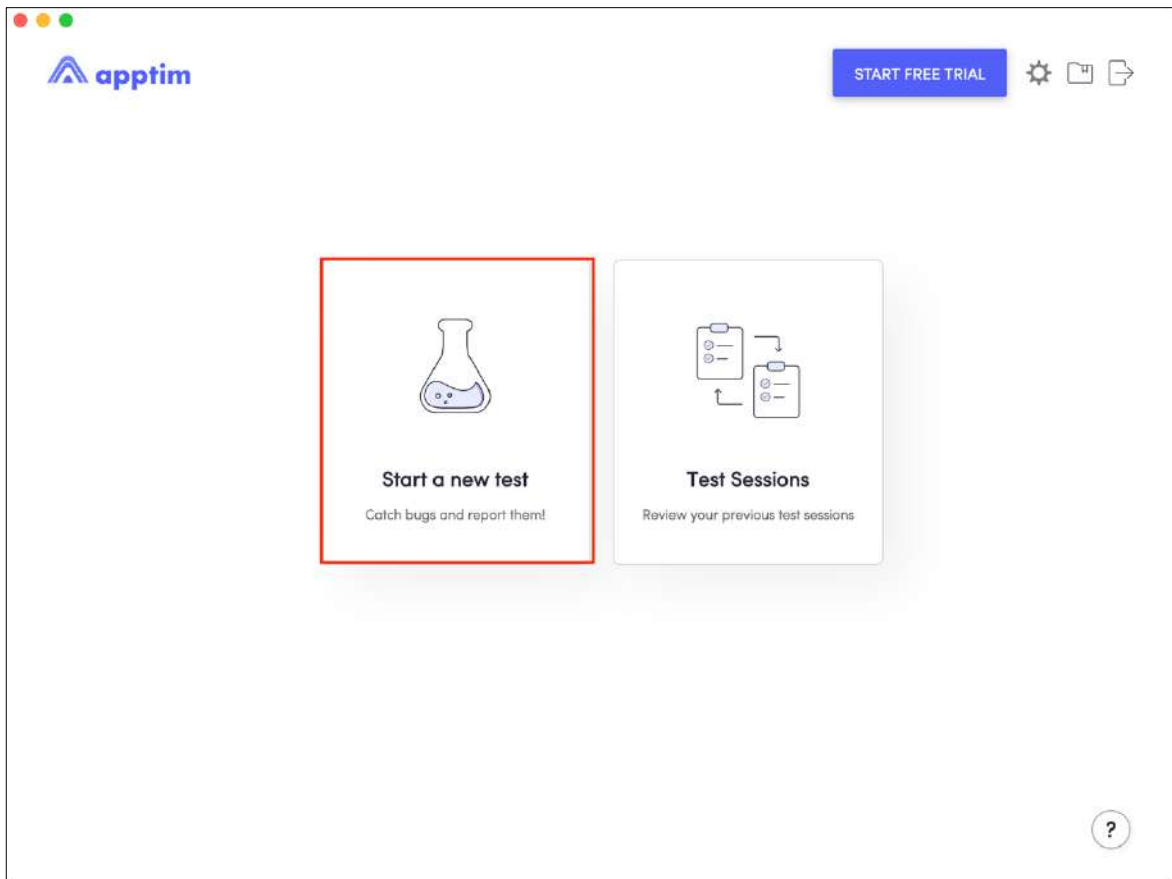


Figura 126 – Pantalla principal de la herramienta Apptim [40]

B. Seleccionar dispositivo: el siguiente paso es seleccionar el dispositivo con el cual se realizará la prueba.

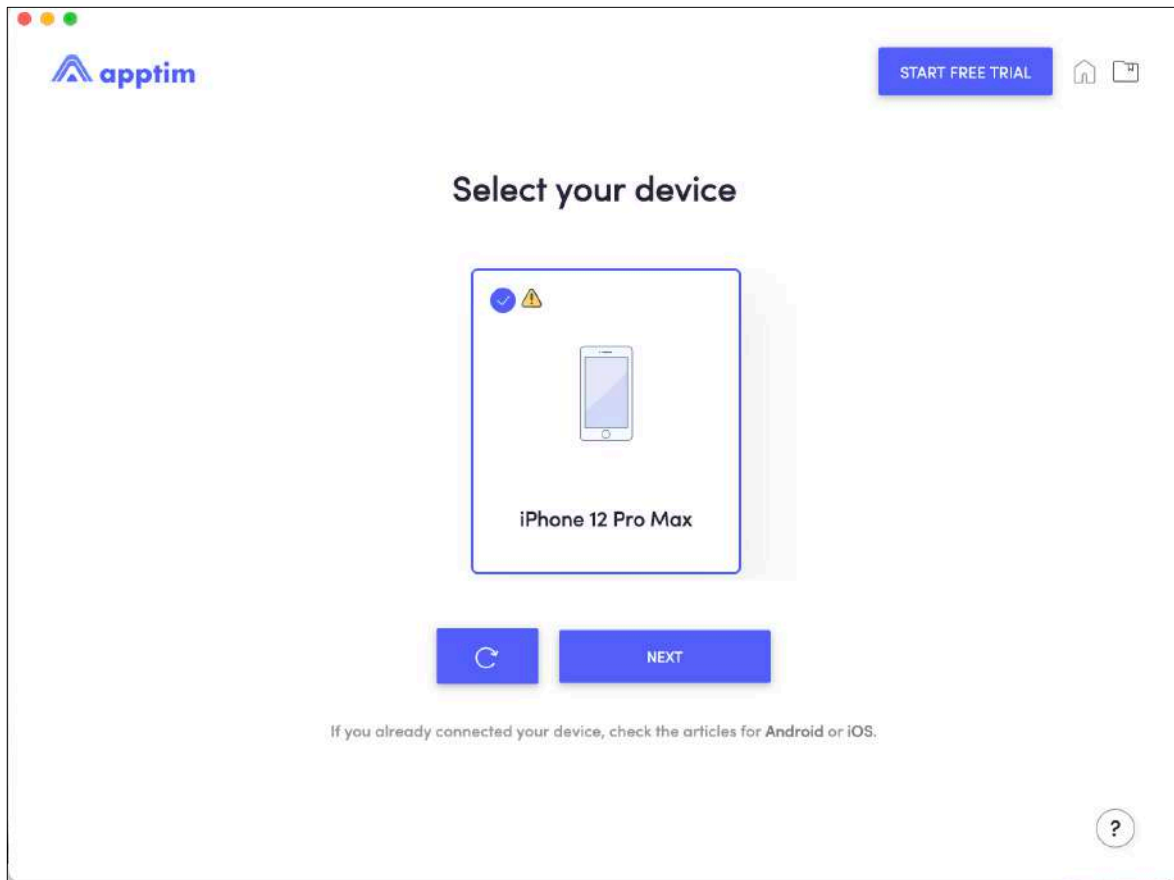


Figura 127 – Selección del dispositivo en donde se realizará la prueba

C. Seleccionar aplicación a testear: se seleccionó la aplicación **“Audity”**

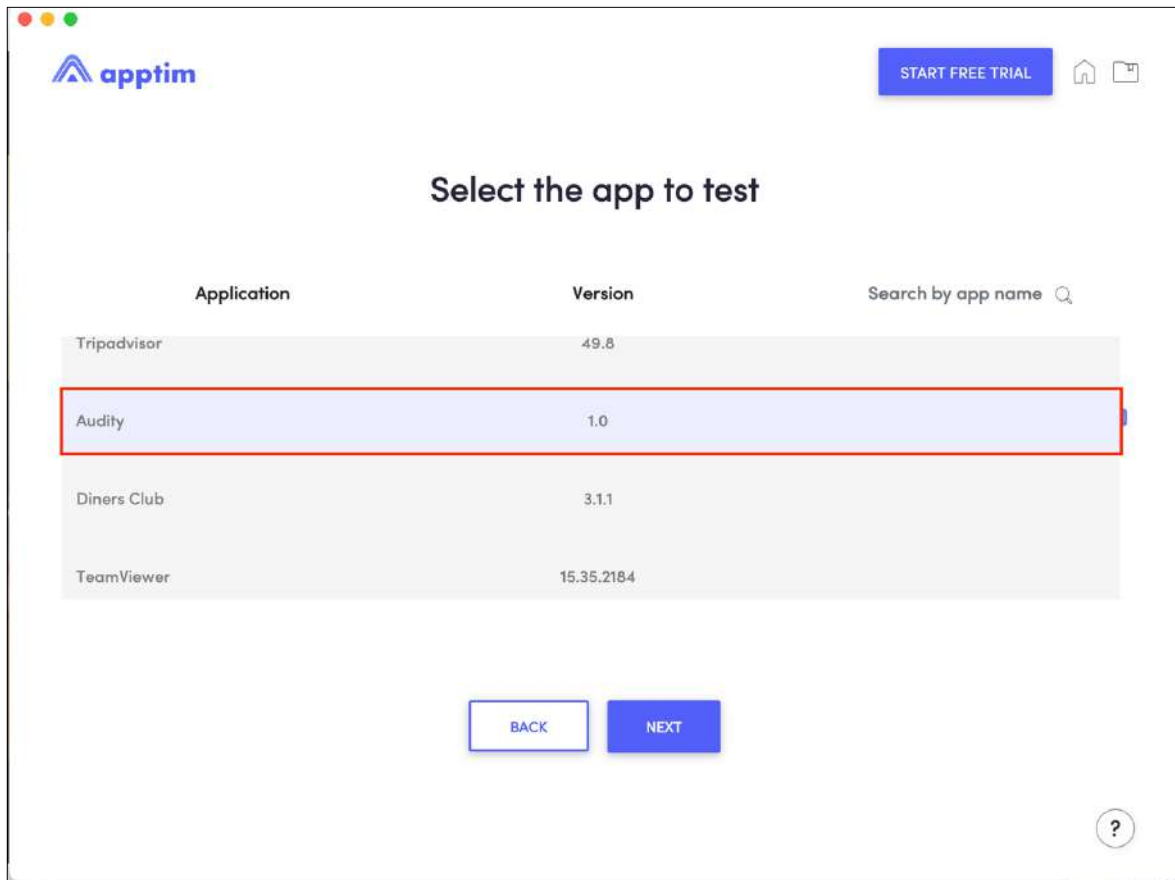


Figura 128 – Selección de la aplicación a testear

A continuación, se indicó el tiempo de duración que tendrá la prueba a ejecutar:

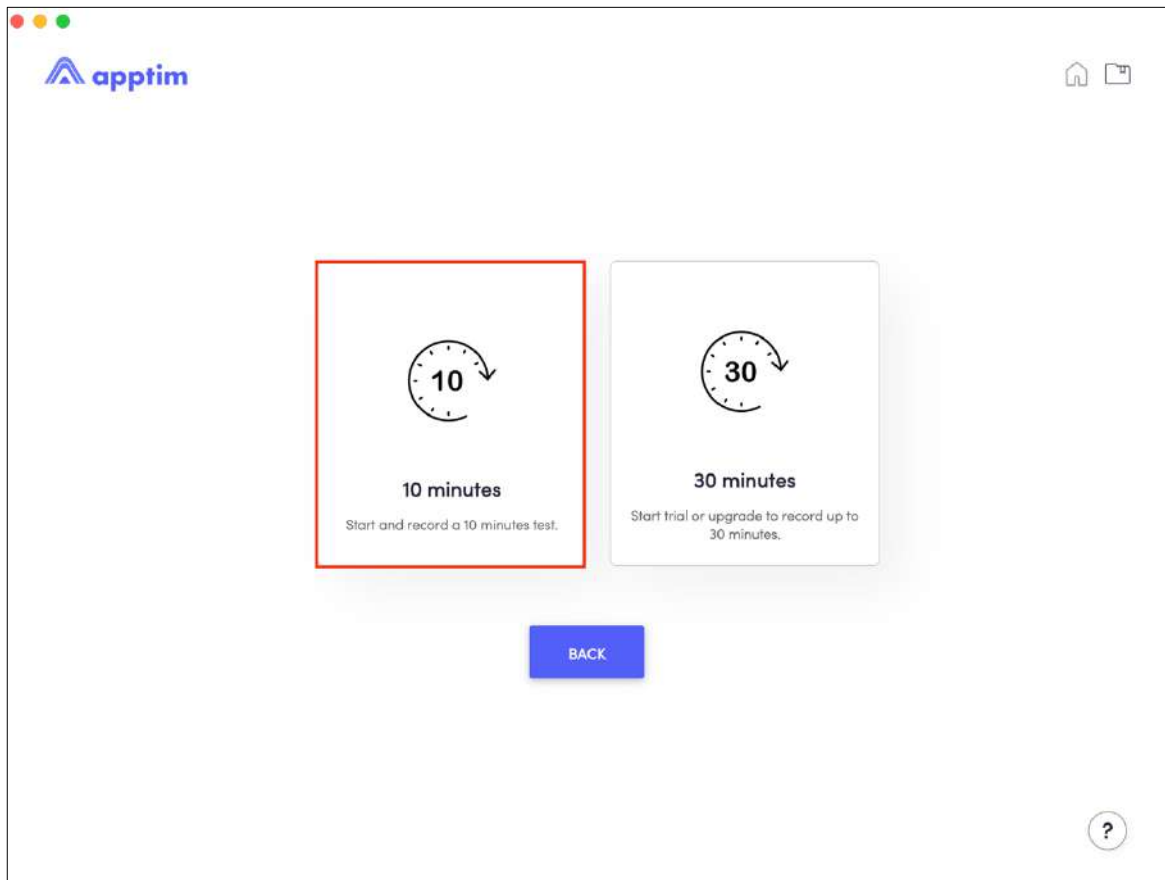


Figura 129 – Configuración del tiempo de duración de la prueba a la aplicación móvil

La herramienta empezará con la ejecución de la prueba mientras el usuario utiliza la aplicación móvil en el dispositivo seleccionado cuando termine la ejecución nos mostrará la sesión exploratoria donde podremos obtener un resumen de la prueba de rendimiento. En esta sesión podremos observar el tipo de dispositivo que se utilizó, la hora de inicio, el tiempo de duración.

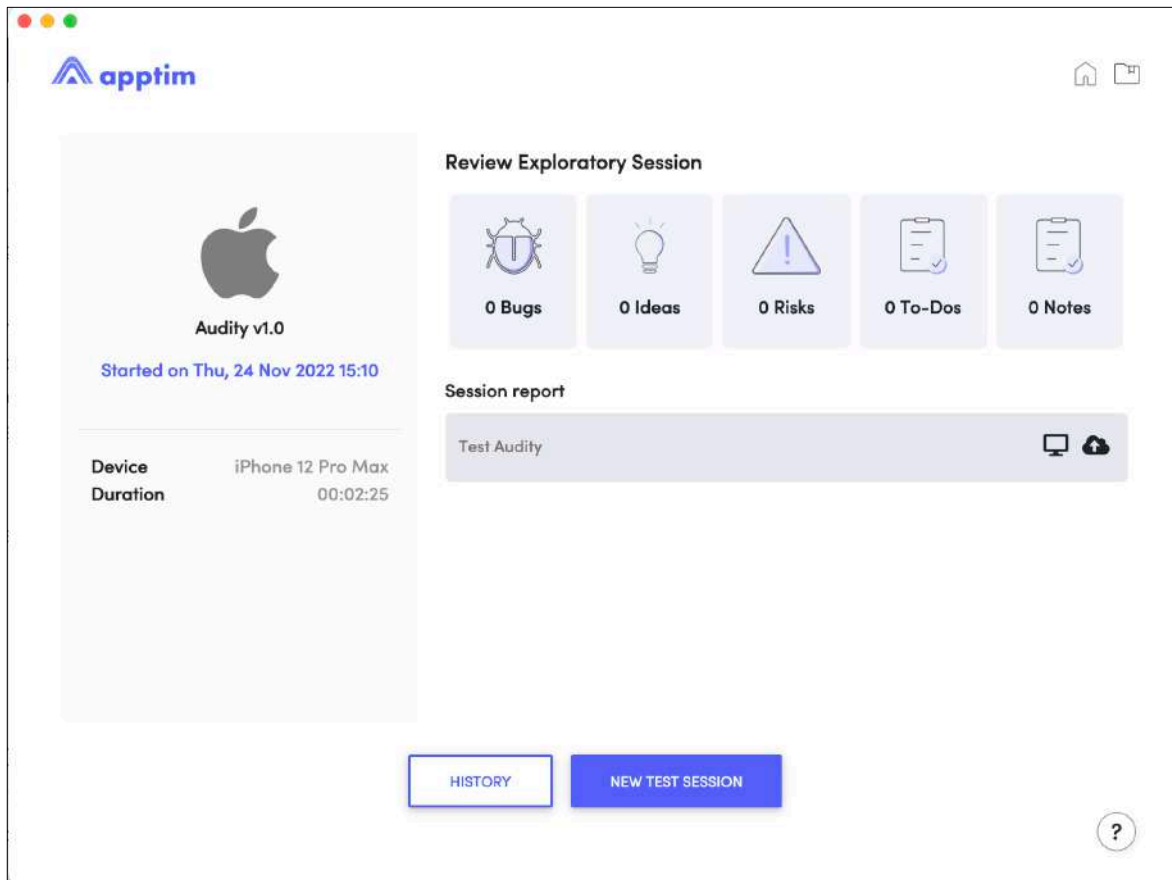


Figura 130 – Sesión exploratoria de la herramienta Apptim

A continuación, se detalla el informe y sus subsecciones que se obtuvieron al finalizar la prueba:

- A. Resumen del informe:** es la sección más importante, contiene datos del rendimiento medido por Apptim, los mismos que se muestran en la parte superior incluyendo el uso medio de la CPU, el uso medio de la memoria, el promedio de FPS y cualquier error o excepción capturado durante la sesión de prueba:

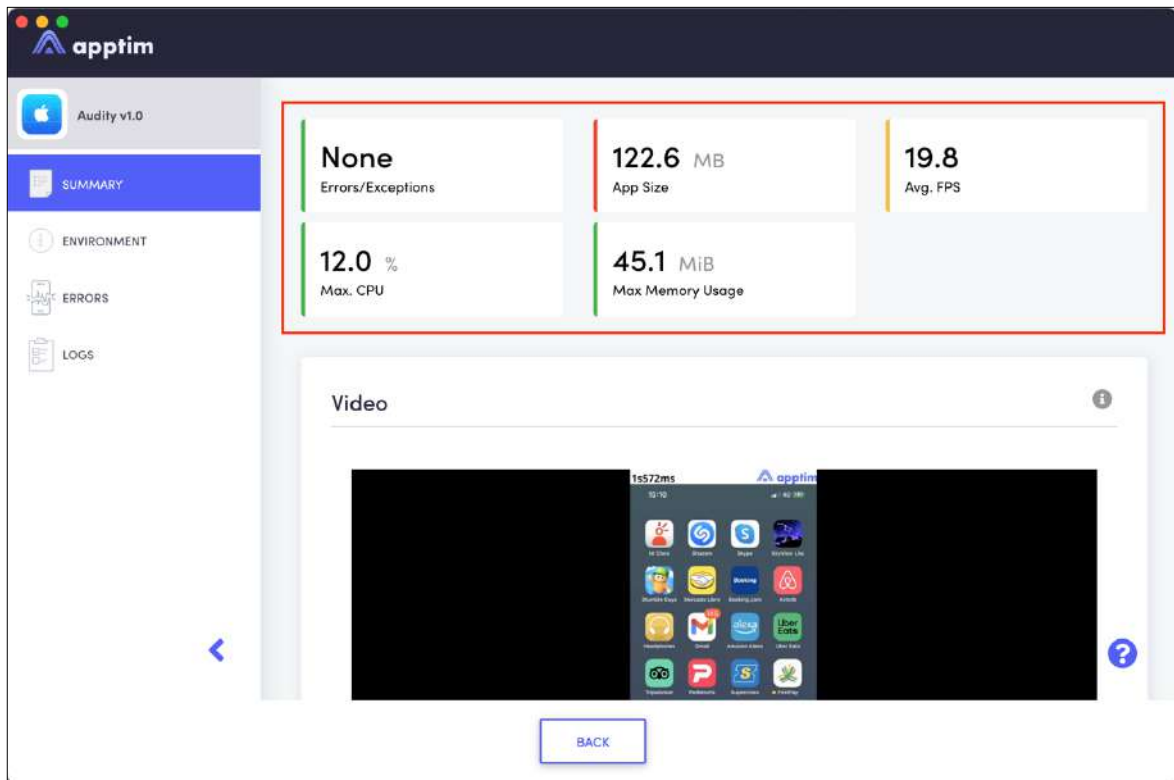


Figura 131 – Datos relevantes capturados durante la prueba

- **Video:** se genera al final de la sesión de la prueba y se acelera para facilitar la navegación.

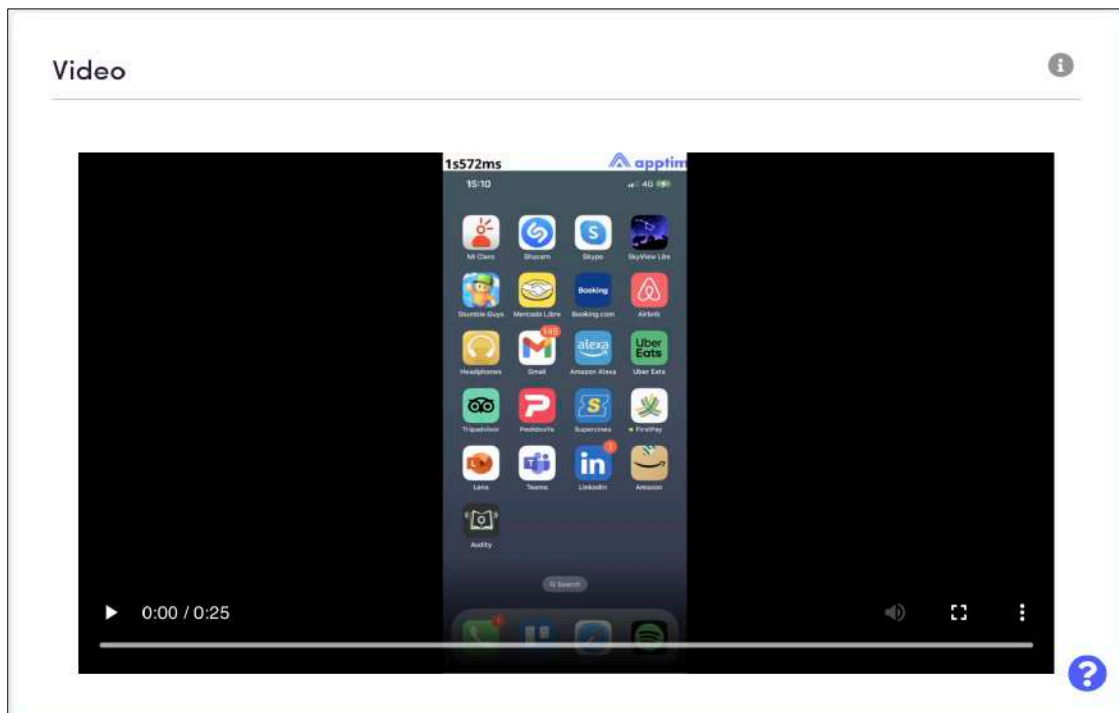


Figura 132 – Video de la ejecución de la prueba

- **La tabla de resumen:** contiene una lista de cada métrica evaluada con respecto a los umbrales predeterminados establecidos por Apptim [40], que se basan en las mejores prácticas de Apple (iOS), junto con otros puntos de referencia del mercado que tienen en cuenta la fragmentación de los dispositivos.



Figura 133 – Tabla de resumen de las métricas evaluadas por Apptim

- **Información de la sesión:** contiene detalles sobre la sesión de la prueba, como el nombre de la prueba, duración, hora de inicio, hora de finalización y el dispositivo utilizado.

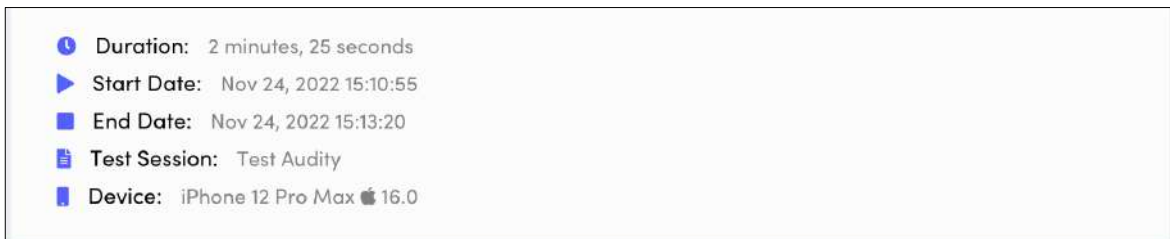


Figura 134 – Información de la sesión

- **Recursos:** contiene los siguientes gráficos:
 - **CPU:** el grafico superior muestra el uso de la CPU durante la sesión de la prueba, el grafico inferior indica el recuento de los hilos cuando se crean y se destruyen, por lo que es de esperar que este valor aumente con las operaciones que exigen más procesamiento en el dispositivo



Figura 135 – Gráfico del uso de la CPU y el recuento de hilos

- **Memoria:** es uno de los recursos más valiosos en cualquier entorno de desarrollo de software, pero es aún más valioso en entornos móviles donde la memoria física es limitada.

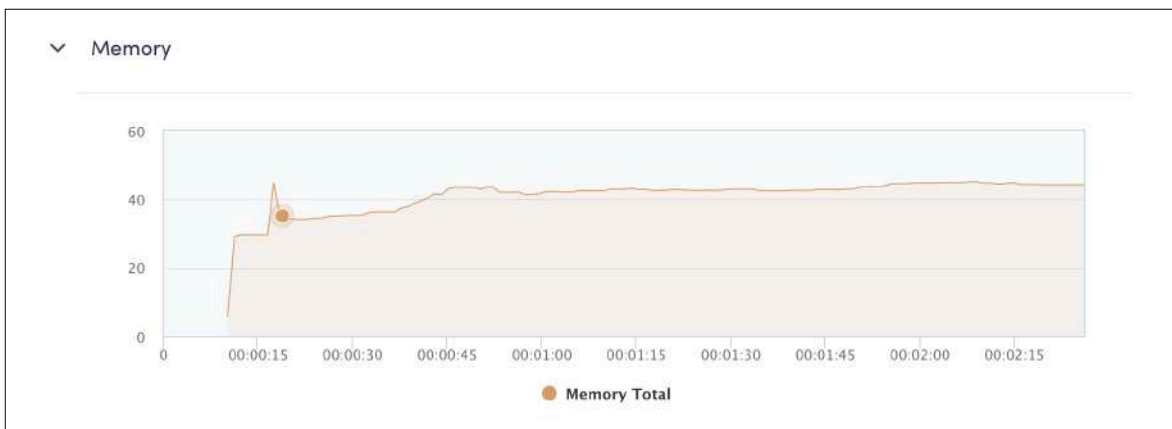


Figura 136 – Memoria utilizada por la aplicación móvil

- **Red:** la aplicación móvil y los servicios del sistema envían o reciben frecuentemente datos de la red a la que está conectado (3G, LTE, 5G, Wifi, etc). Apptim captura la cantidad de bytes enviados como se observa

en la parte superior del gráfico y recibidos desde el dispositivo móvil durante la sesión de prueba ilustrados en la parte inferior del gráfico.



Figura 137 – Gráfico de la red cuando envía y recibe bytes

- **Render:** el dibujo en los dispositivos móviles ocurre cada ~16ms, esto significa que la aplicación móvil debe renderizar a ~60 FPS (fotogramas por segundo).

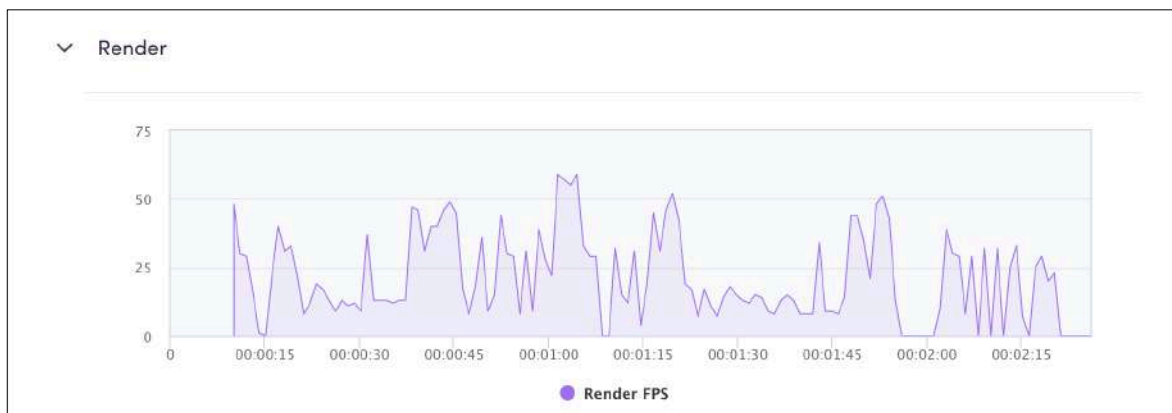


Figura 138 – Gráfico del rendimiento de la renderización FPS

- **Almacenamiento:** cantidad de almacenamiento que está actualmente disponible para su uso.

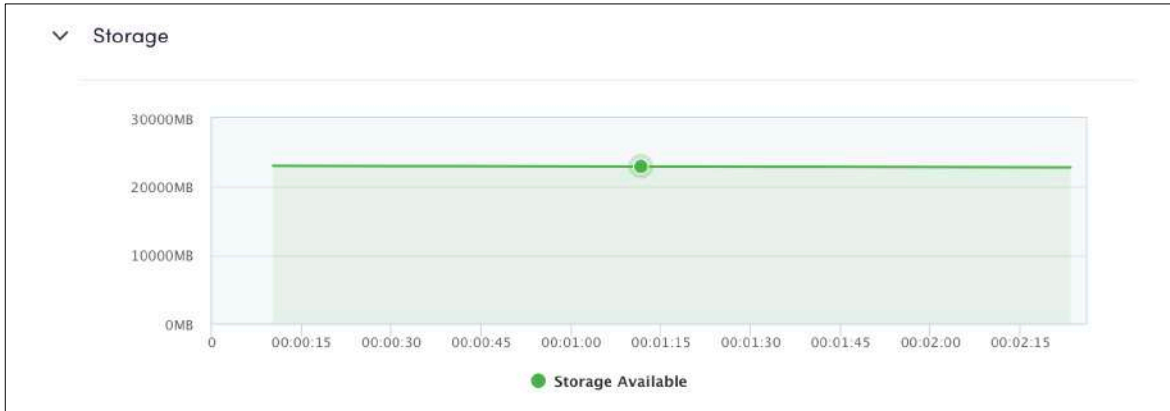


Figura 139 – Gráfico del almacenamiento disponible

4.6. Ejecución de pruebas del Sprint 6

Para la ejecución de este conjunto de pruebas se utilizó un data set con 31 audiolibros.

- **Pruebas funcionales del Sprint 6**

Al finalizar con la ejecución del Sprint se realizaron las pruebas de aceptación [25] a las historias de usuario del usuario final agregadas a la aplicación móvil, a continuación, se detalla cada una de ellas:

A. Buscar los audiolibros por autor: cumpliendo con los siguientes criterios de aceptación:

- El mínimo de caracteres para realizar una búsqueda será de 3 caracteres
- Se mostrará:
 - Portada del audiolibro
 - Título del audiolibro
 - Autor
 - Fecha de publicación

Cuando el usuario final haya iniciado sesión en la aplicación móvil y seleccione el icono de “**Buscar**” de la barra inferior como se muestra en la siguiente imagen:

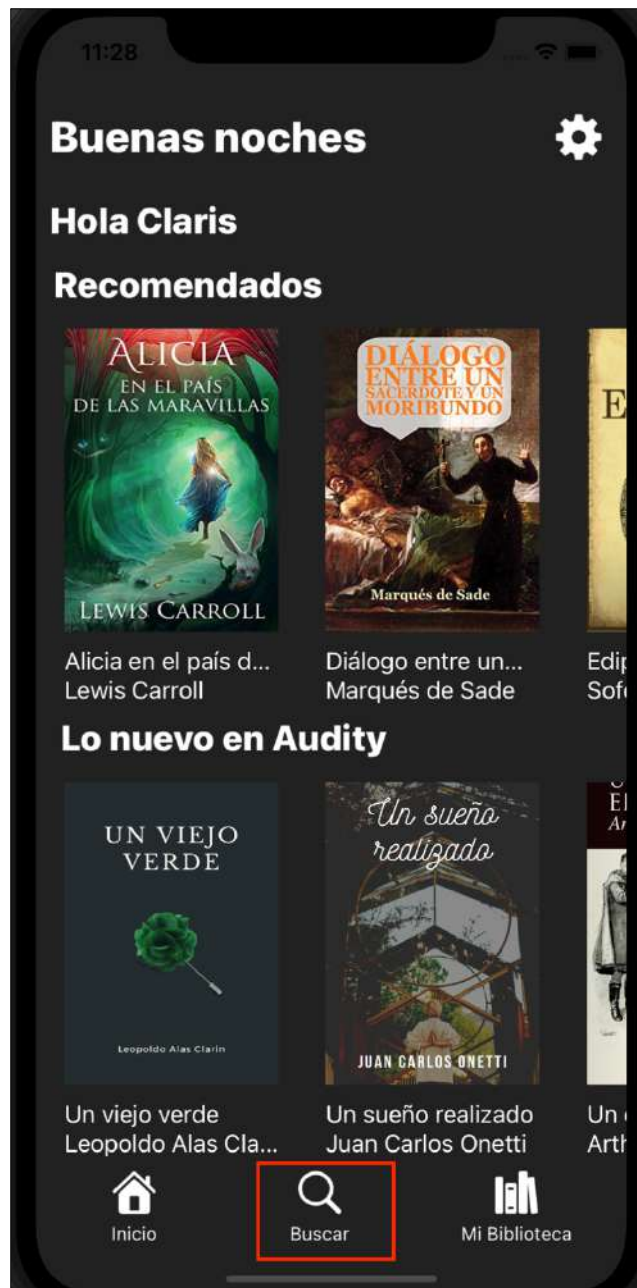


Figura 140 – Pantalla principal de la aplicación móvil eligiendo el icono de buscar

La aplicación móvil le redireccionará a la pantalla de buscar donde el usuario final deberá seleccionar la opción (Autor), en la barra de búsqueda escribirá el nombre del autor mostrando los resultados como se observa a continuación:

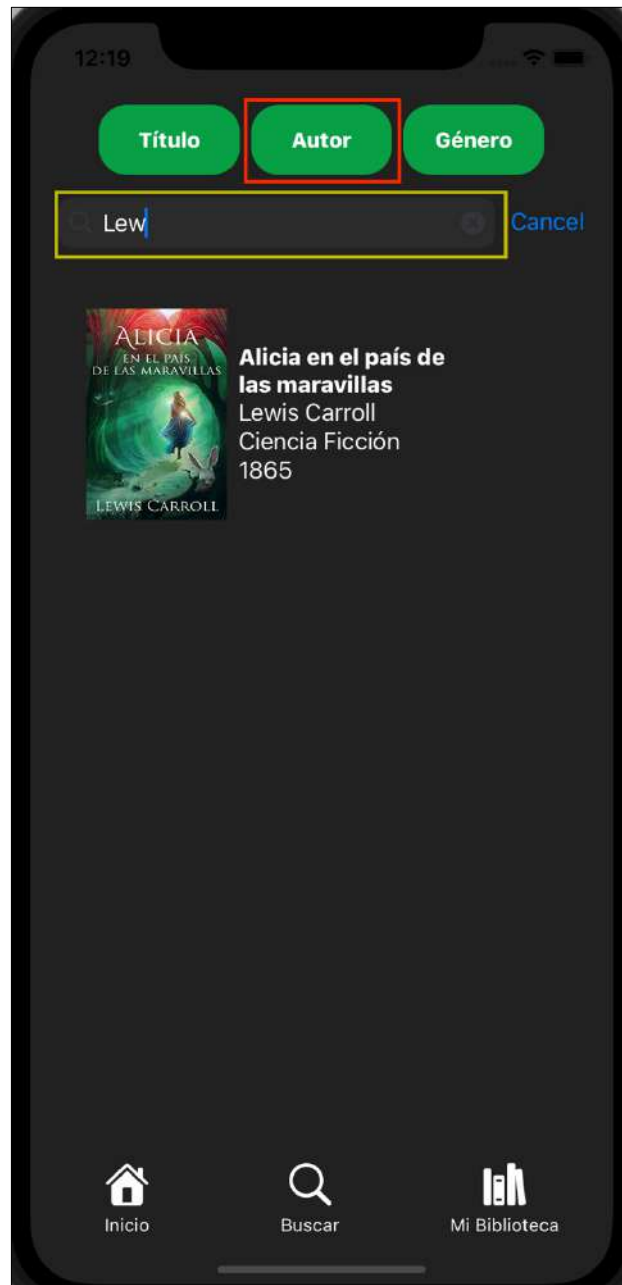


Figura 141 – Búsqueda por nombre de autor desde la aplicación móvil

B. Buscar los audiolibros existentes por título de audiolibro: cumpliendo con los siguientes criterios de aceptación:

- El mínimo de caracteres para realizar una búsqueda será de 3 caracteres
- Se mostrará:
 - Portada del audiolibro
 - Título del audiolibro

- Autor
- Fecha de publicación

Cuando el usuario final haya iniciado sesión en la aplicación móvil y seleccione el icono de “**Buscar**” de la barra inferior como lo indica la **Figura 140**, la aplicación móvil le redireccionará a la pantalla de buscar donde el usuario final deberá seleccionar la opción (Título), en la barra de búsqueda escribirá el título del audiolibro mostrándole los audiolibros que coincidan con la búsqueda como se observa a continuación:

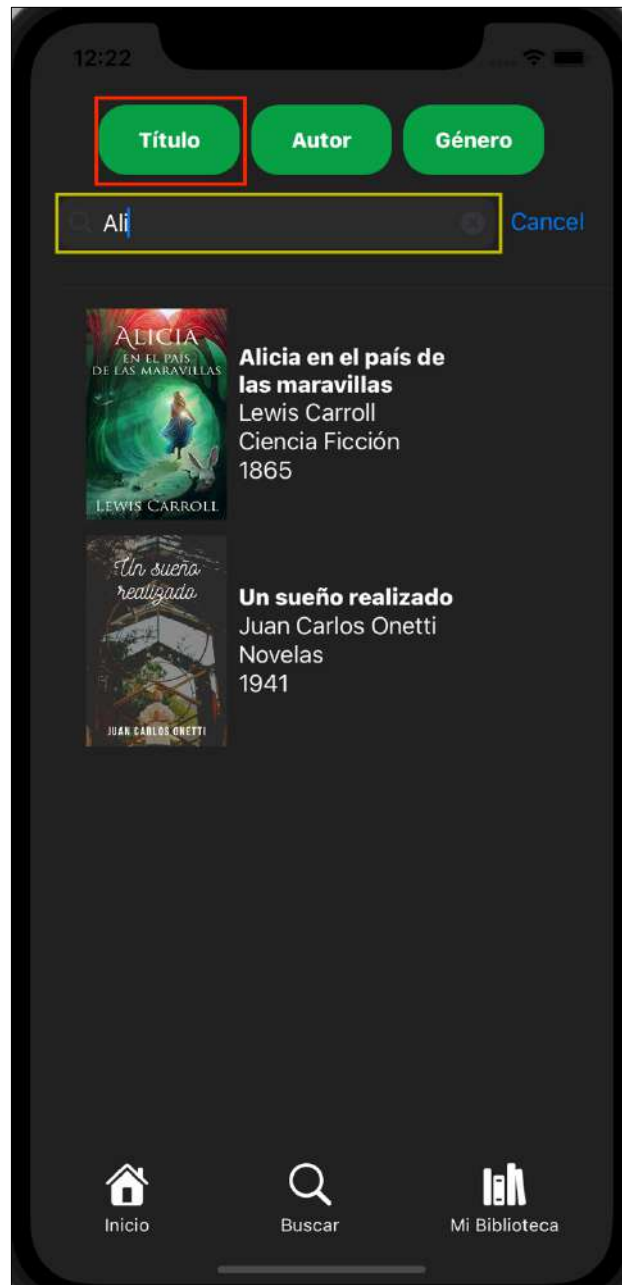


Figura 142 – Búsqueda por título de audiolibro desde la aplicación móvil

C. Buscar los audiolibros existentes por género: cumpliendo con los siguientes criterios de aceptación:

- El mínimo de caracteres para realizar una búsqueda será de 3 caracteres
- Se mostrará:
 - Portada del audiolibro

- Título del audiolibro
- Autor
- Fecha de publicación

Cuando el usuario final haya iniciado sesión en la aplicación móvil y seleccione el icono de “**Buscar**” de la barra inferior como lo indica la **Figura 140**, la aplicación móvil le redireccionará a la pantalla de buscar donde el usuario final deberá seleccionar la opción (Género), en la barra de búsqueda escribirá el género del audiolibro mostrándole los audiolibros que coincidan con la búsqueda como se observa a continuación:

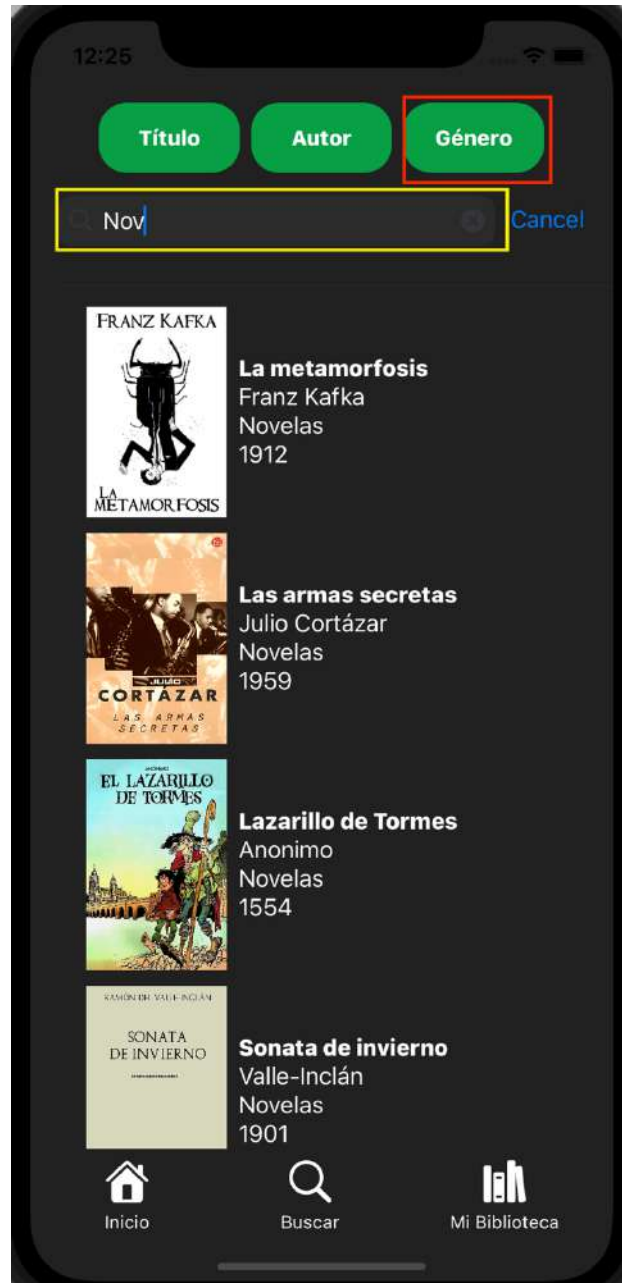


Figura 143 – Búsqueda por género del audiolibro desde la aplicación móvil

D. Reproducir un audiolibro: Cuando el usuario final haya iniciado sesión en la aplicación móvil podrá seleccionar cualquier libro a su elección de la sesión de recomendados o lo nuevo:

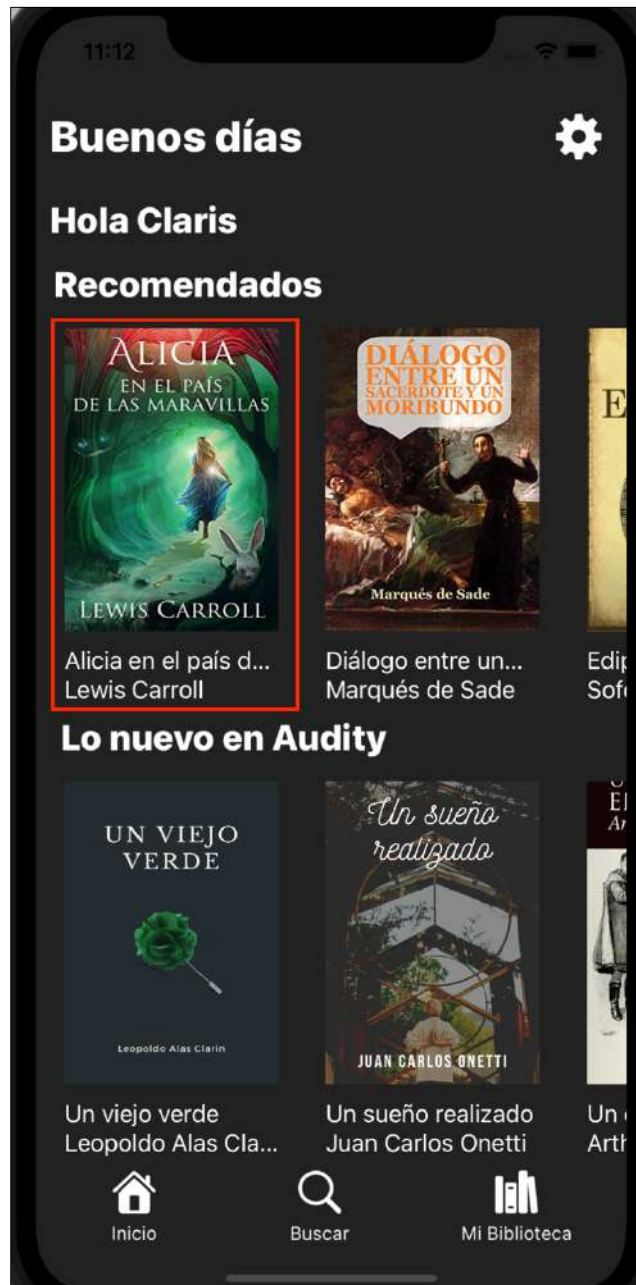


Figura 144 – Pantalla principal de la aplicación móvil seleccionando un audiolibro

Una vez que el usuario final hay seleccionado el audiolibro es redireccionado a la pantalla de información, la misma que contiene los datos del audiolibro como su género, año de publicación, título del audiolibro, sinopsis y la reseñas. Para reproducirlo debe seleccionar el icono de **“Play”**.



Figura 145 – Pantalla de información de un audiolibro

La aplicación móvil lo redirige a la pantalla de reproducción donde podrá interactuar con la barra de reproducción en donde se encuentran los botones de pausar/reproducir, adelantar o retroceder 10 segundos, volver a reproducirlo o regresar a la pantalla anterior:



Figura 146 – Pantalla de reproducción de un audiolibro

E. Crear una lista de reproducción: Cuando el usuario final haya iniciado sesión y seleccione el icono **“Mi biblioteca”**, de la barra inferior:

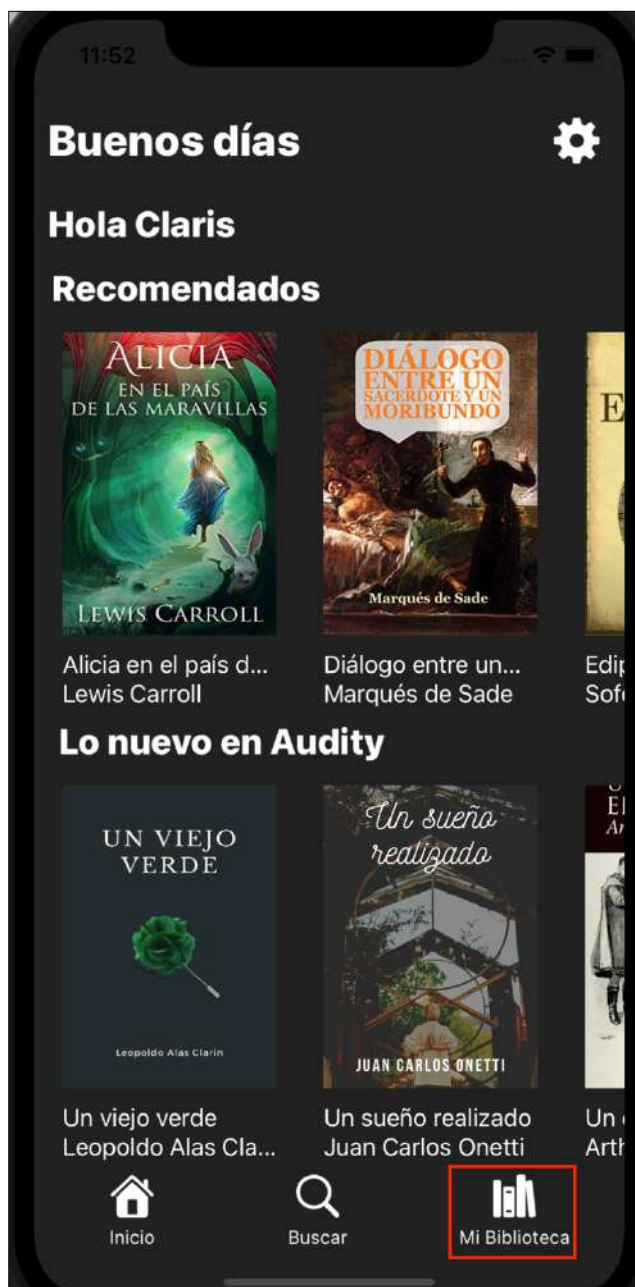


Figura 147 – Pantalla principal seleccionando el icono de "Mi biblioteca"

Posteriormente la aplicación móvil le redireccionará a la pantalla en donde se mostrarán las listas de reproducción creadas por el usuario, para crear una nueva lista debe seleccionar el icono de "+":

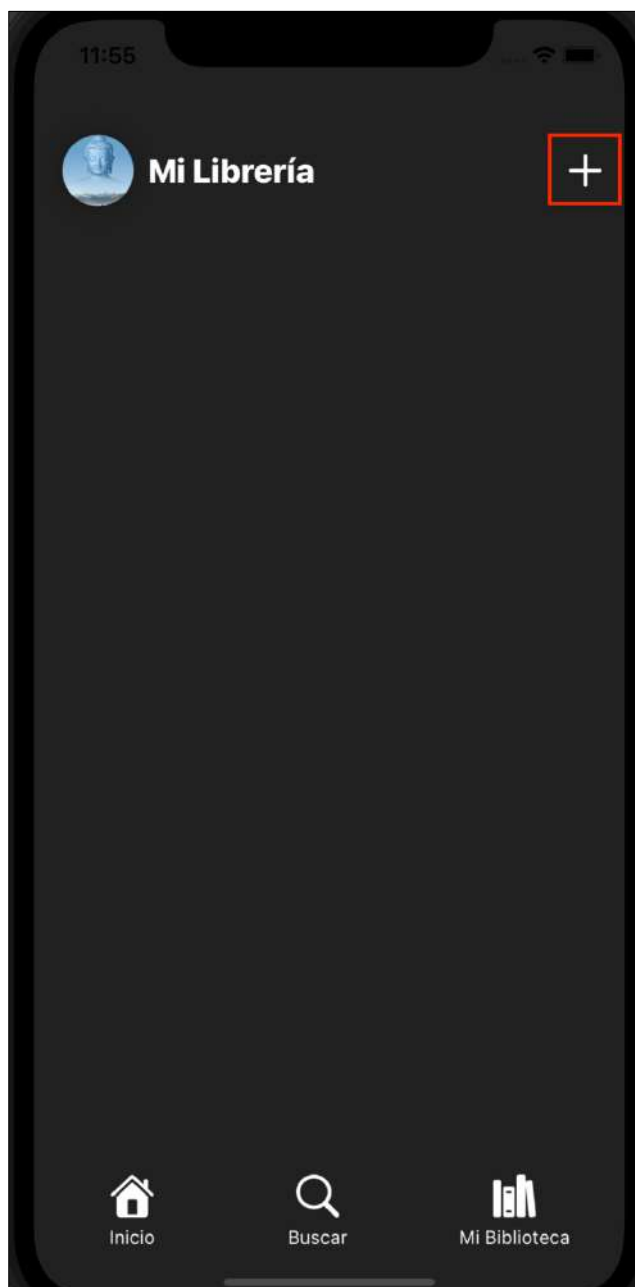


Figura 148 – Pantalla de “Mi librería” seleccionando el icono para crear una lista de reproducción

Se le presentará la pantalla para crear una lista de reproducción donde el usuario deberá llenar los siguientes campos:

- Nombre de la lista
 - El campo tendrá un máximo de 15 caracteres.
 - El campo permitirá caracteres alfanuméricos y especiales.

Una vez que el usuario final haya llenado el campo correspondiente deberá seleccionar el icono “**Crear**”. La aplicación móvil le mostrará el mensaje “**Lista de reproducción creada**”.

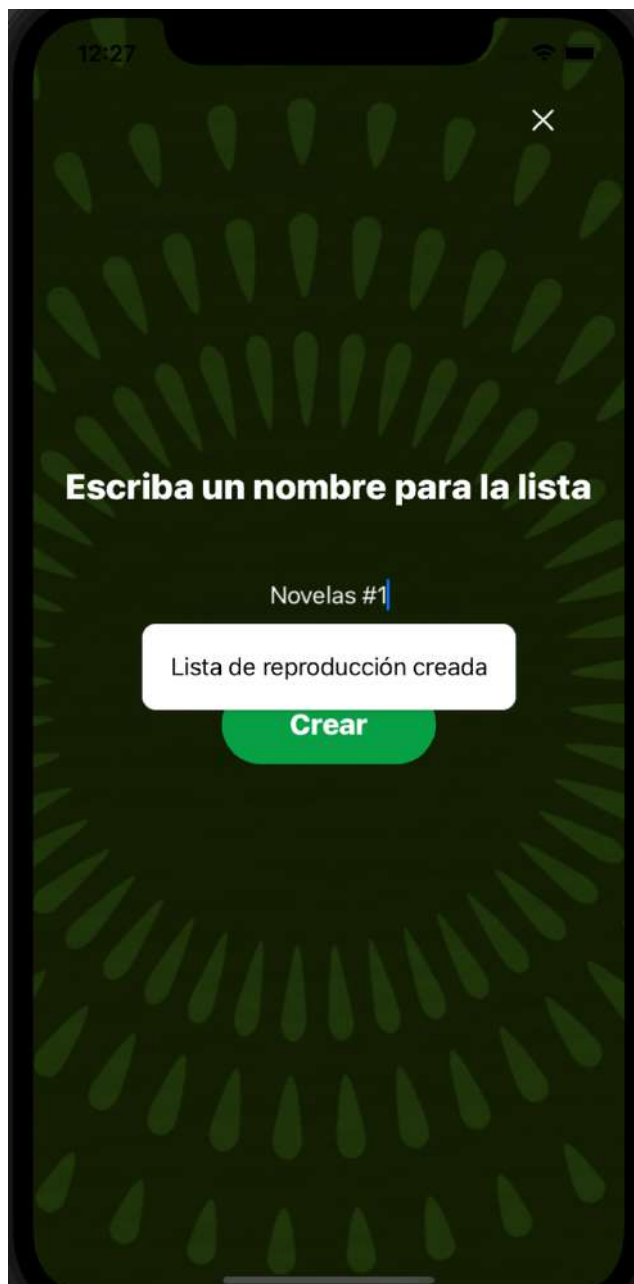


Figura 149 – Creación de una lista de reproducción

En caso de que el usuario exceda el número de caracteres permitidos la aplicación móvil le mostrará el siguiente mensaje “Límite de caracteres alcanzado”

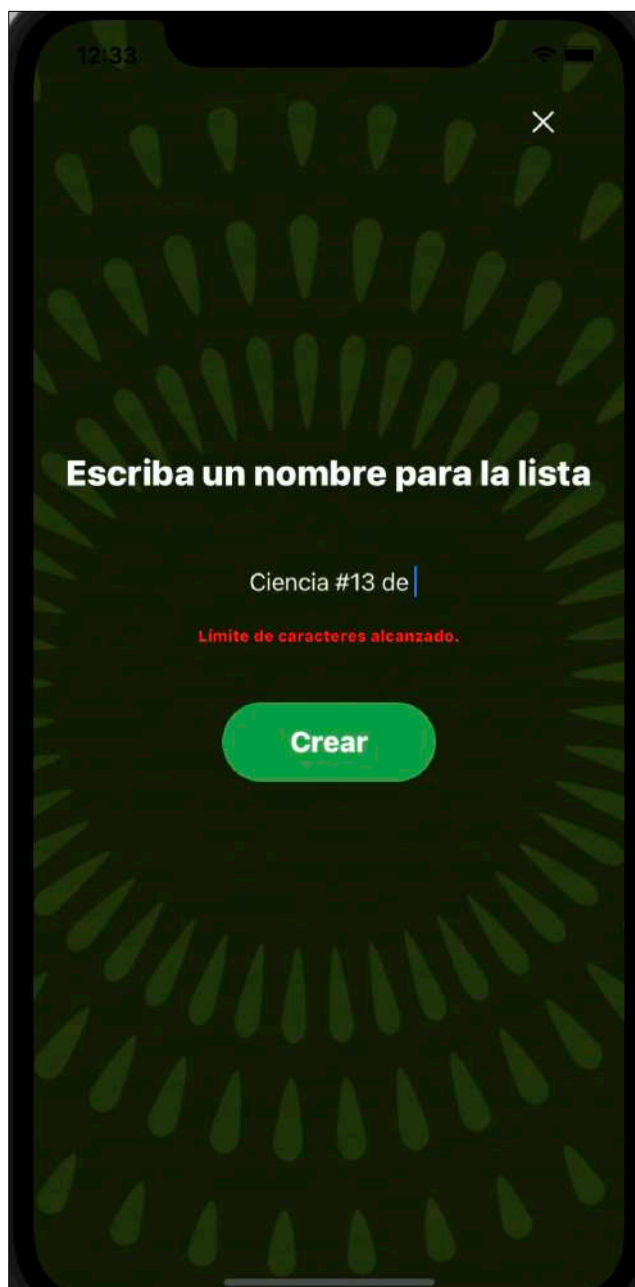


Figura 150 – Mensaje cuando el usuario excede el número de caracteres permitidos al crear una lista de reproducción

- **Agregar audiolibros a una lista de reproducción:** cuando el usuario final ingrese a la aplicación móvil, seleccione un audiolibro y elija el icono “Play” como se observa en la **Figura 145**, el usuario debe seleccionar el icono de los tres puntos de la pantalla del reproductor de audiolibros:



Figura 151 – Pantalla del reproductor de un audiolibro seleccionando el icono de los tres puntos

Posteriormente la aplicación móvil le redireccionará a una pantalla donde deberá seleccionar la opción **“Agregar a lista de reproducción”**.



Figura 152 – Selección de la opción Añadir a lista de reproducción

La aplicación móvil le redireccionará a la pantalla en donde el usuario final podrá agregar el audiolibro a una lista de reproducción seleccionando la lista de reproducción a su elección la aplicación móvil le mostrará el siguiente mensaje **“Audiolibro agregado”**.

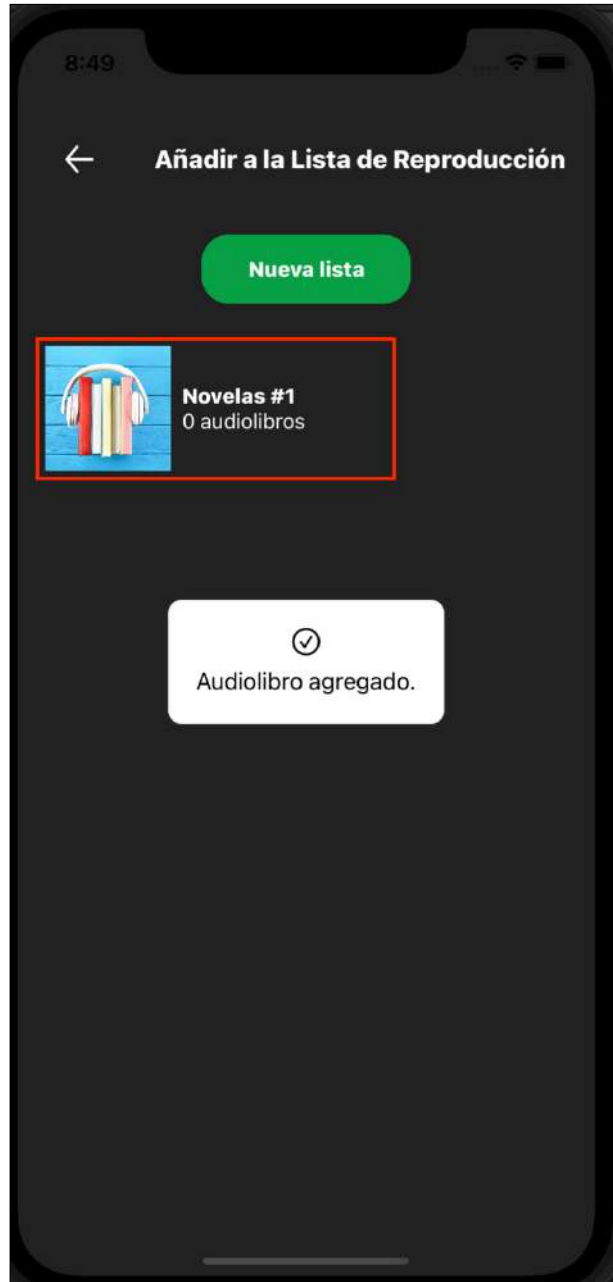


Figura 153 – Agregación de un audiolibro a una lista de reproducción

En caso de que el usuario final desee agregar el mismo audiolibro a la misma lista de reproducción, la aplicación móvil no le permitirá agregarlo mostrándole el siguiente mensaje **“El audiolibro ya se encuentra agregado”**:

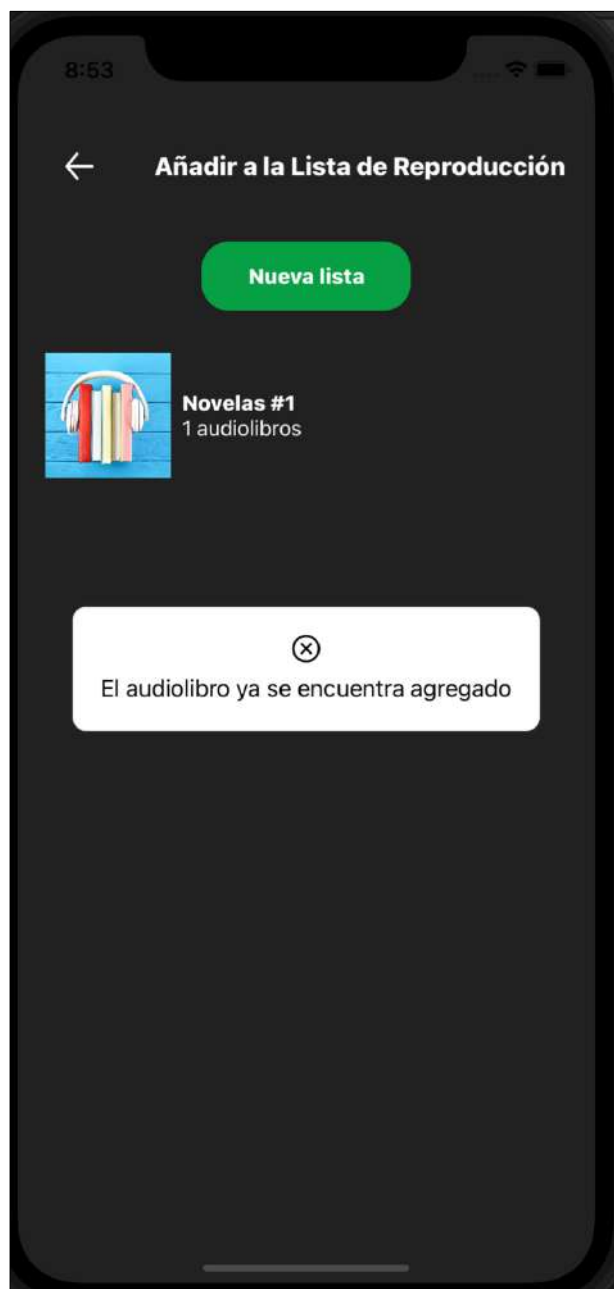


Figura 154 – Mensaje de alerta cuando el usuario final desee agregar el mismo libro a la misma lista de reproducción

F. Reseñar un audiolibro: cuando el usuario final ingrese a la aplicación móvil y elija un audiolibro como se observa en la **Figura 145**, se dirigirá a la pantalla del reproductor y deberá seleccionar el icono de los tres puntos como se observa en la **Figura 155**, y seleccionará la opción **“Reseña”**:

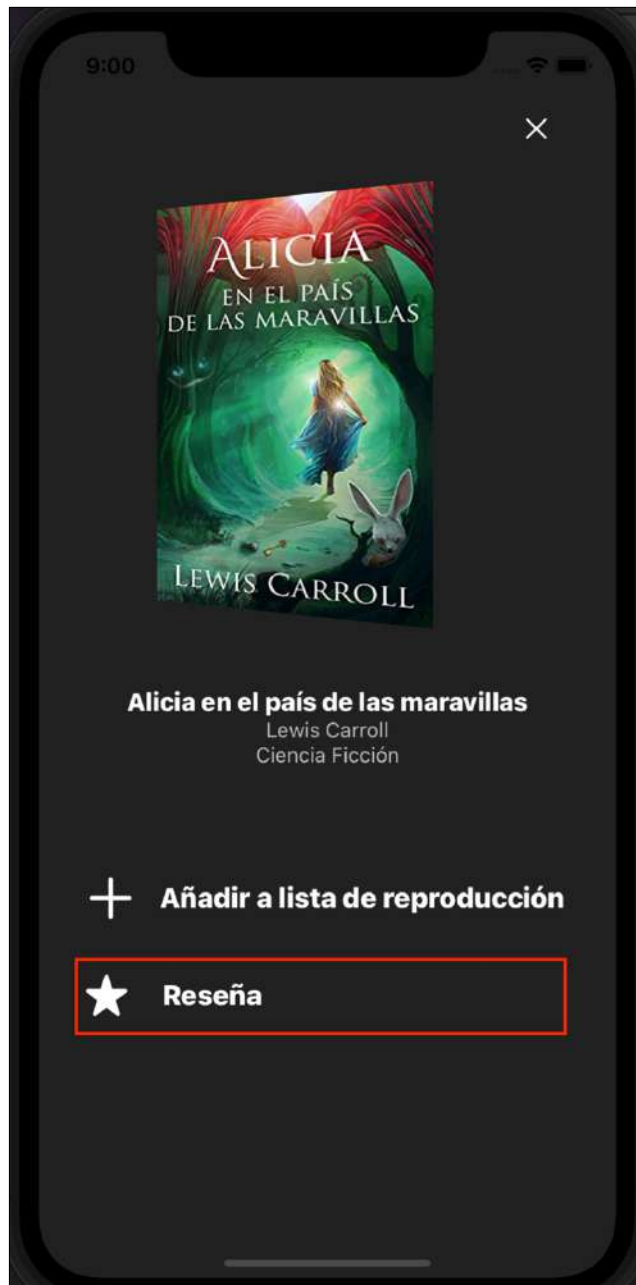


Figura 155 – Selección de la opción Reseña

La aplicación móvil le dirige a la pantalla de reseñas donde el usuario final deberá llenar los siguientes campos:

- Puntuación
 - La puntuación estará representada por estrellas en un rango de una a cinco estrellas.
- Comentario
 - El comentario tendrá una longitud máxima de 200 caracteres

- El campo permitirá caracteres alfanuméricos y especiales

Una vez que el usuario final haya concluido deberá seleccionar el icono **“Enviar”**, para guardar la reseña del audiolibro.



Figura 156 – Reseña a un audiolibro

En caso de el usuario final ingrese un comentario que exceda el límite permitido la aplicación móvil le mostrara un mensaje de advertencia:

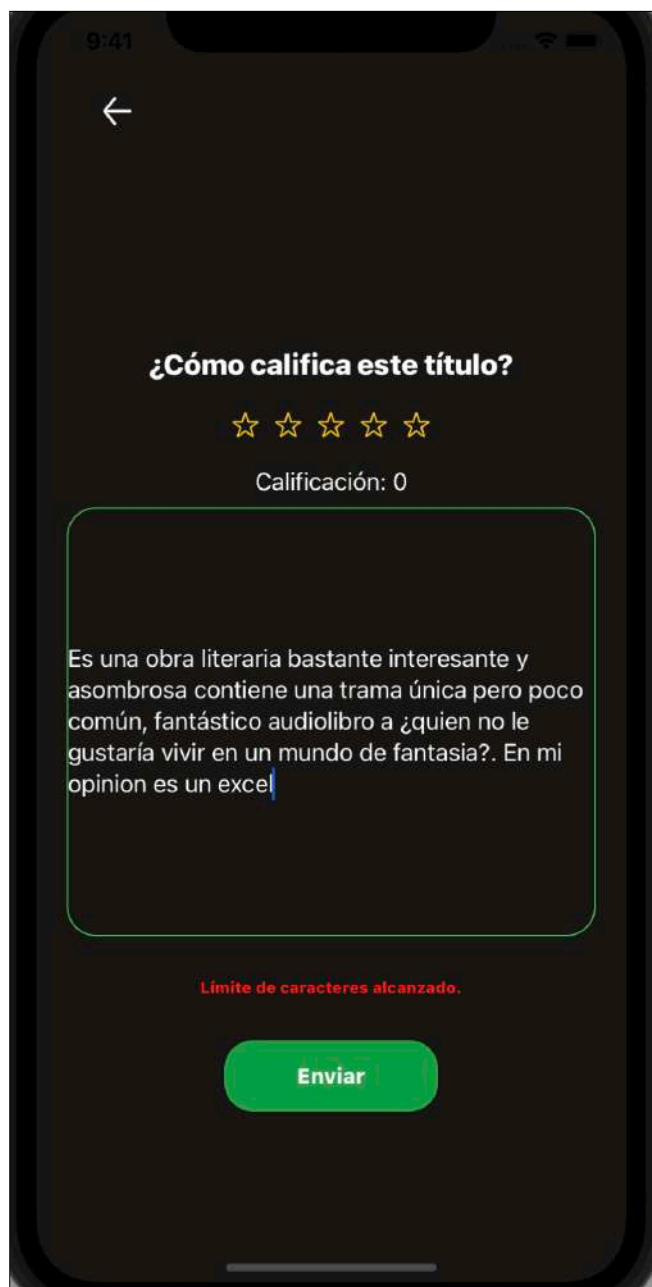


Figura 157 – Mensaje de advertencia cuando el comentario exceda su número de caracteres permitido

- **Pruebas no funcionales del Sprint 6**

La prueba de rendimiento [26] se realizó con la herramienta Apptim [40], para crear la prueba se siguieron los pasos utilizados en el Sprint 6 de la sección de Pruebas no funcionales de la aplicación móvil.

La herramienta empezó con la ejecución de la prueba mientras el usuario utiliza la aplicación móvil en el dispositivo seleccionado cuando termine la ejecución nos mostrará la sesión exploratoria donde podremos obtener un resumen de la prueba de rendimiento. En esta sesión podremos observar el tipo de dispositivo que se utilizó, la hora de inicio, el tiempo de duración.

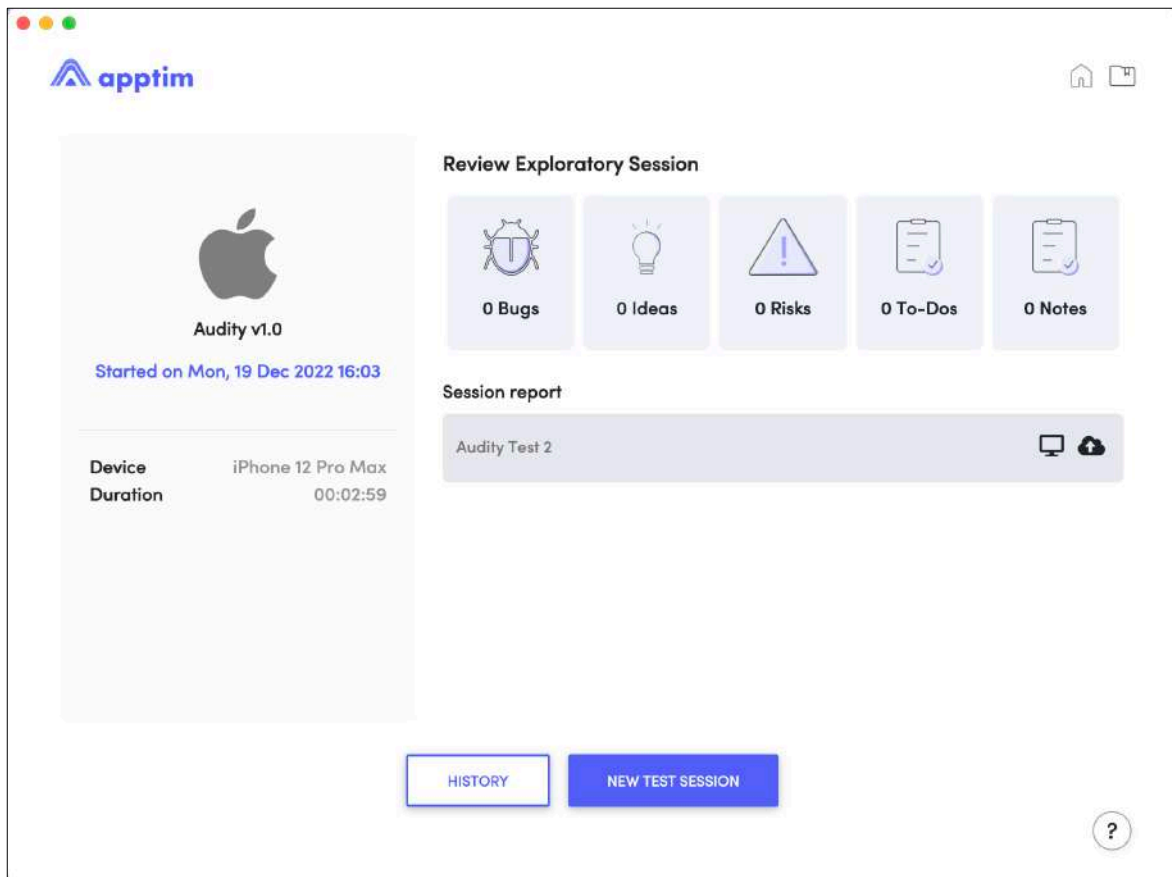


Figura 158 – Sesión exploratoria de la herramienta Apptim de la segunda prueba

A continuación, se detalla el informe y sus subsecciones que se obtuvieron al finalizar la prueba:

A. Resumen del informe: es la sección más importante, contiene datos del rendimiento medido por Apptim, los mismos que se muestran en la parte superior incluyendo el uso medio de la CPU, el uso medio de la memoria, el promedio de FPS y cualquier error o excepción capturado durante la sesión de prueba:

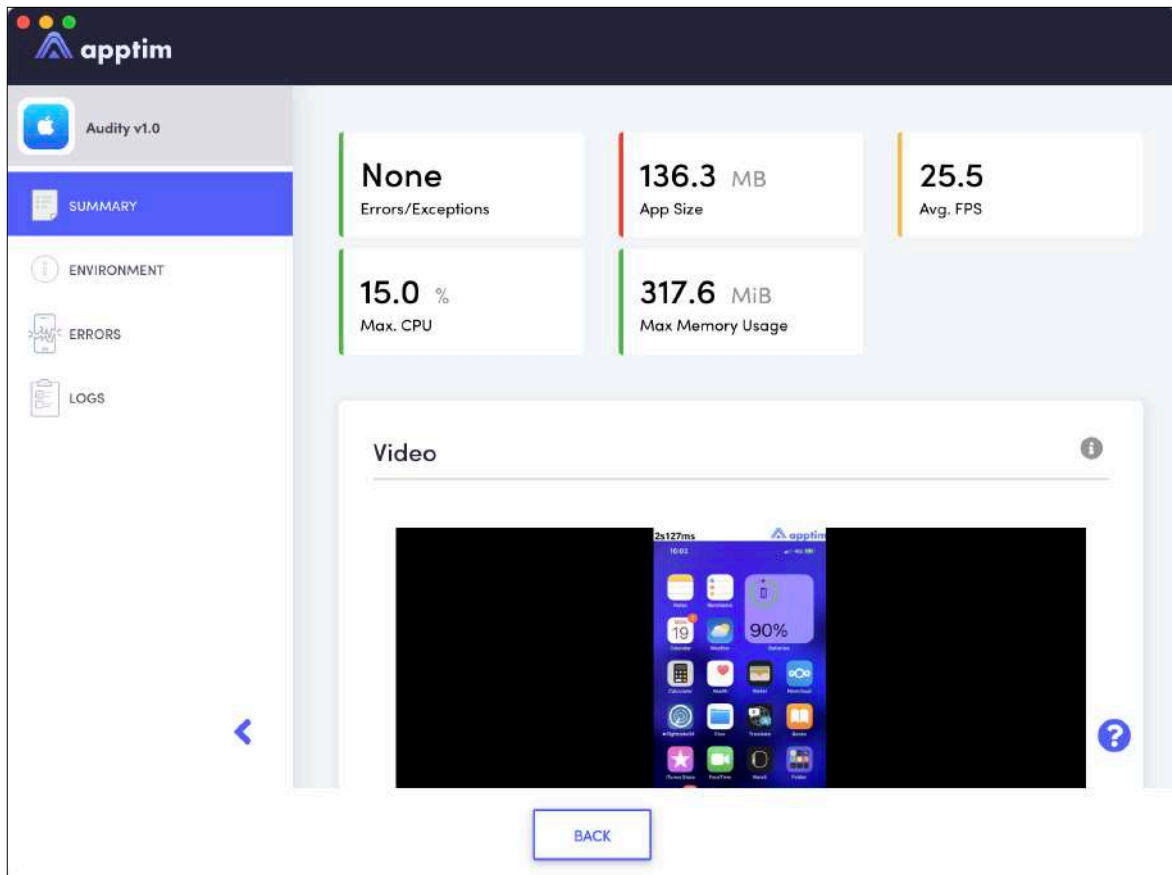


Figura 159 – Datos relevantes capturados durante la segunda prueba

- **Video:** se genera al final de la sesión de la prueba y se acelera para facilitar la navegación.

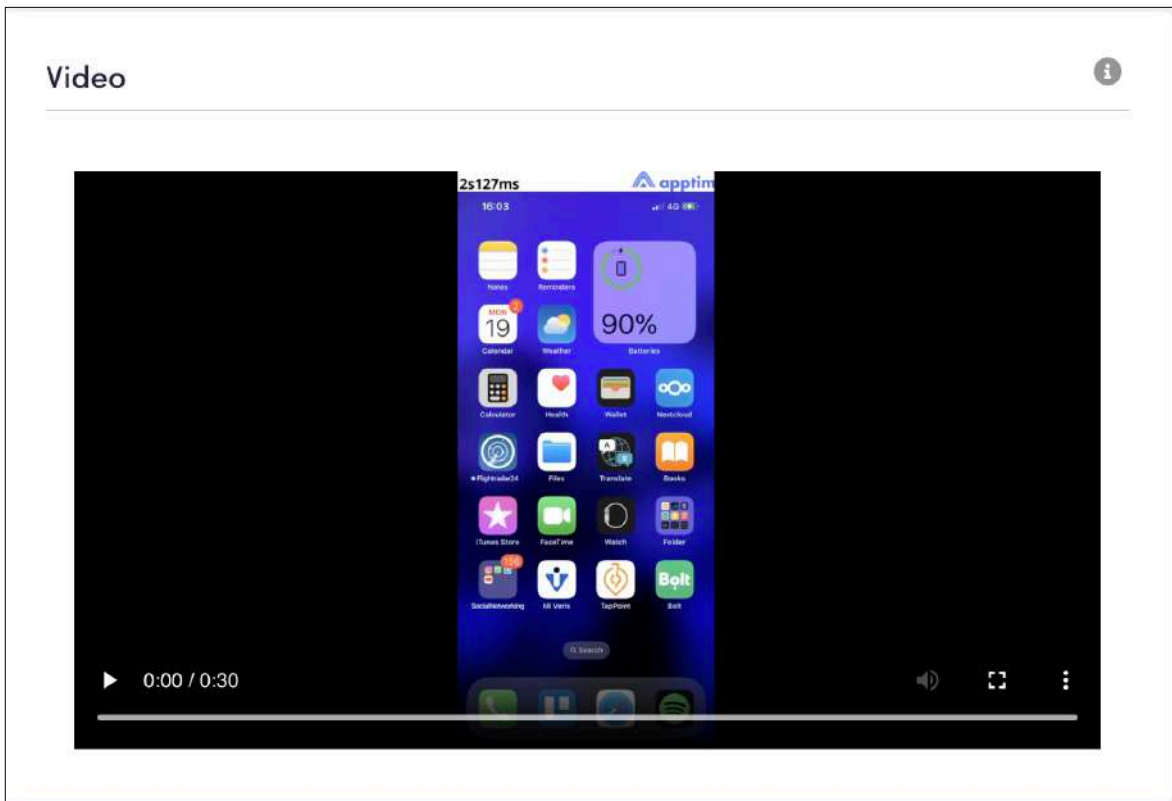


Figura 160 – Video de la ejecución de la segunda prueba

- **La tabla de resumen:** contiene una lista de cada métrica evaluada con respecto a los umbrales predeterminados establecidos por Apptim [40], que se basan en las mejores prácticas de Apple (iOS), junto con otros puntos de referencia del mercado que tienen en cuenta la fragmentación de los dispositivos.



Figura 161 – Tabla de resumen de las métricas evaluadas por Apptim durante la segunda prueba

- **Información de la sesión:** contiene detalles sobre la sesión de la prueba, como el nombre de la prueba, duración, hora de inicio, hora de finalización y el dispositivo utilizado.

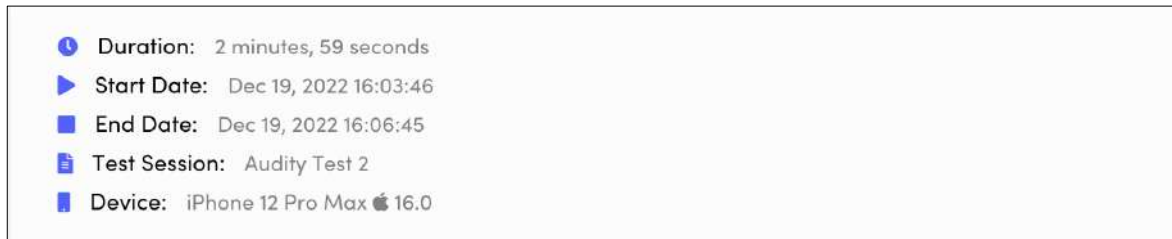


Figura 162 – Información de la sesión de la segunda prueba

- **Recursos:** contiene los siguientes gráficos:
 - **CPU:** el grafico superior muestra el uso de la CPU durante la sesión de la prueba, el grafico inferior indica el recuento de los hilos cuando se crean y se destruyen, por lo que es de esperar que este valor aumente con las operaciones que exigen más procesamiento en el dispositivo.



Figura 163– Gráfico del uso de la CPU y el recuento de hilos de la segunda prueba

- **Memoria:** es uno de los recursos más valiosos en cualquier entorno de desarrollo de software, pero es aún más valioso en entornos móviles donde la memoria física es limitada.

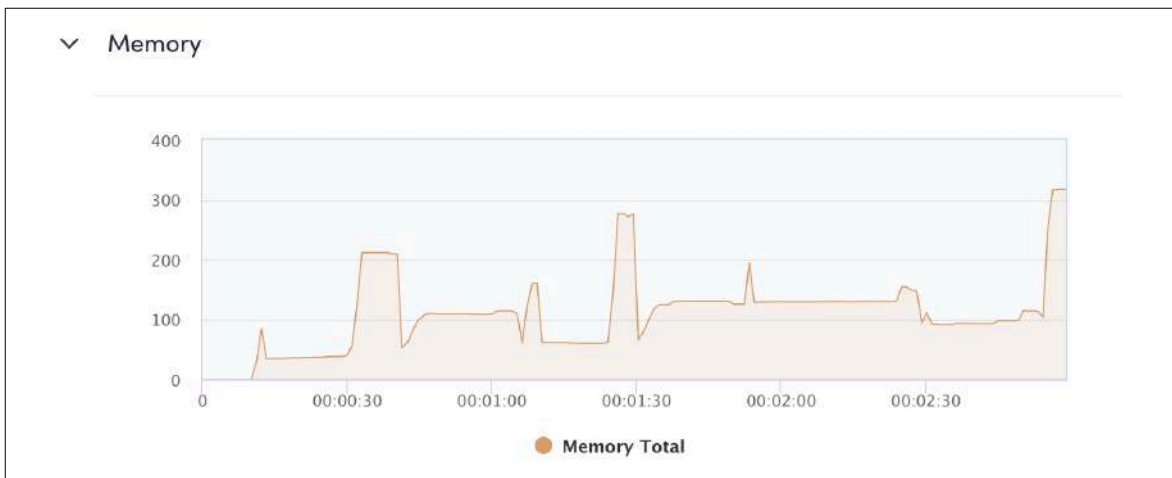


Figura 164 – Memoria utilizada por la aplicación móvil de la segunda prueba

- **Red:** la aplicación móvil y los servicios del sistema envían o reciben frecuentemente datos de la red a la que está conectado (3G, LTE, 5G, Wifi, etc). Apptim [40] captura la cantidad de bytes enviados como se observa en la parte superior del gráfico y recibidos desde el dispositivo móvil durante la sesión de prueba ilustrados en la parte inferior del gráfico.

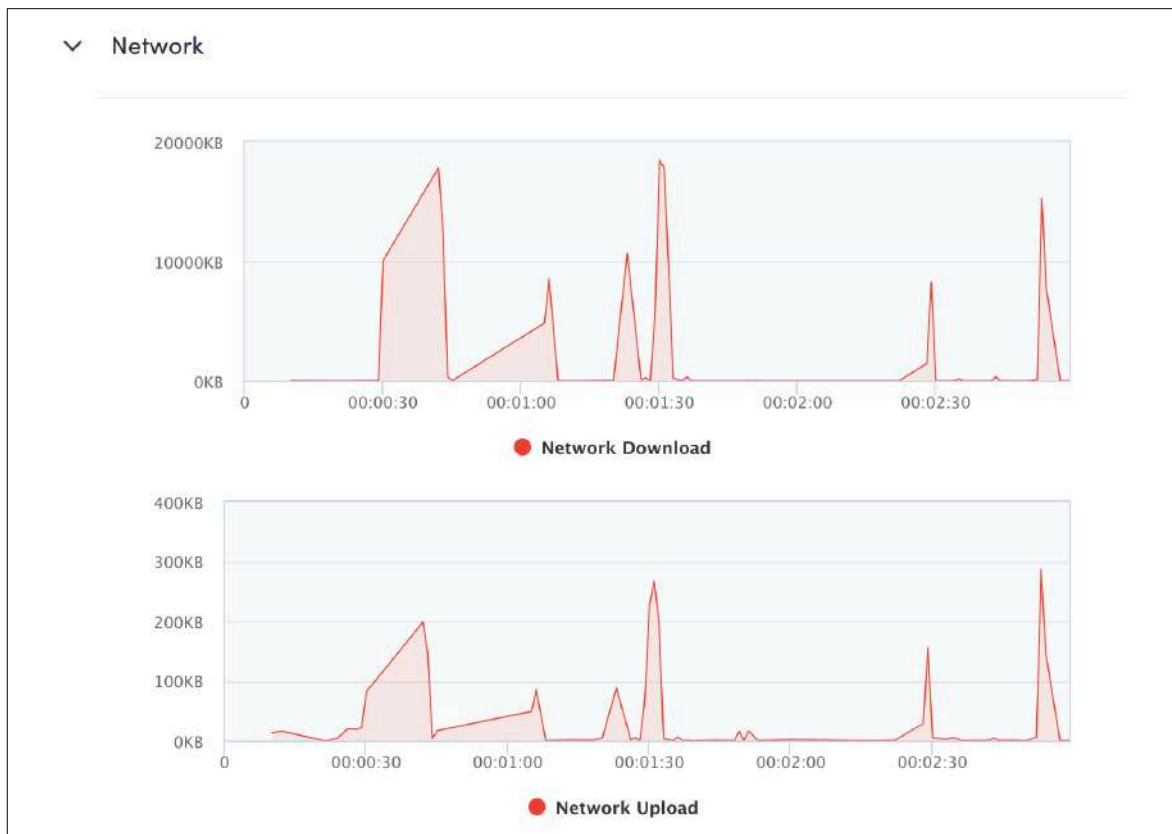


Figura 165 – Gráfico de la red cuando envía y recibe bytes de la segunda prueba

- **Render:** el dibujo en los dispositivos móviles ocurre cada $\sim 16\text{ms}$, esto significa que la aplicación móvil debe renderizar a $\sim 60\text{ FPS}$ (fotogramas por segundo).

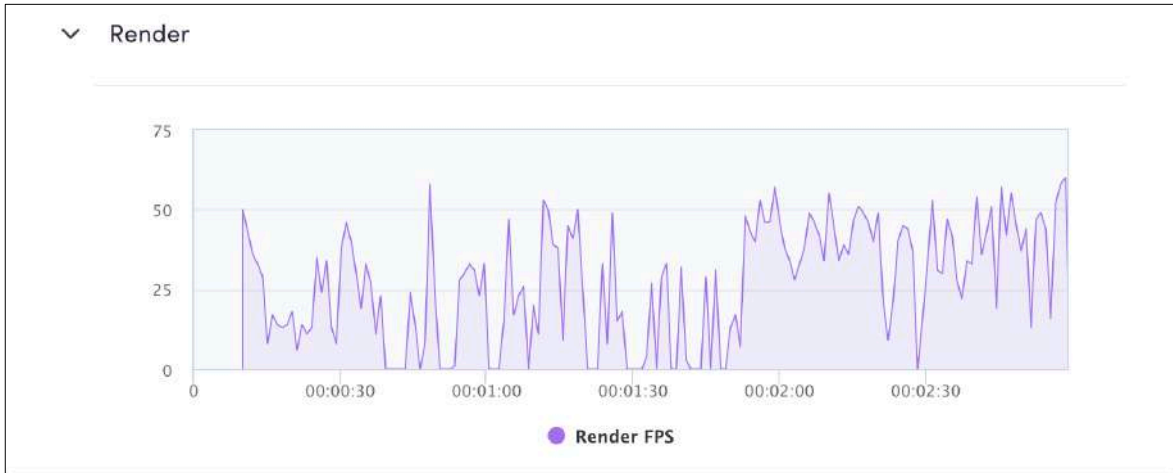


Figura 166 – Gráfico del rendimiento de la renderización FPS de la segunda prueba

- **Almacenamiento:** cantidad de almacenamiento que está actualmente disponible para su uso.

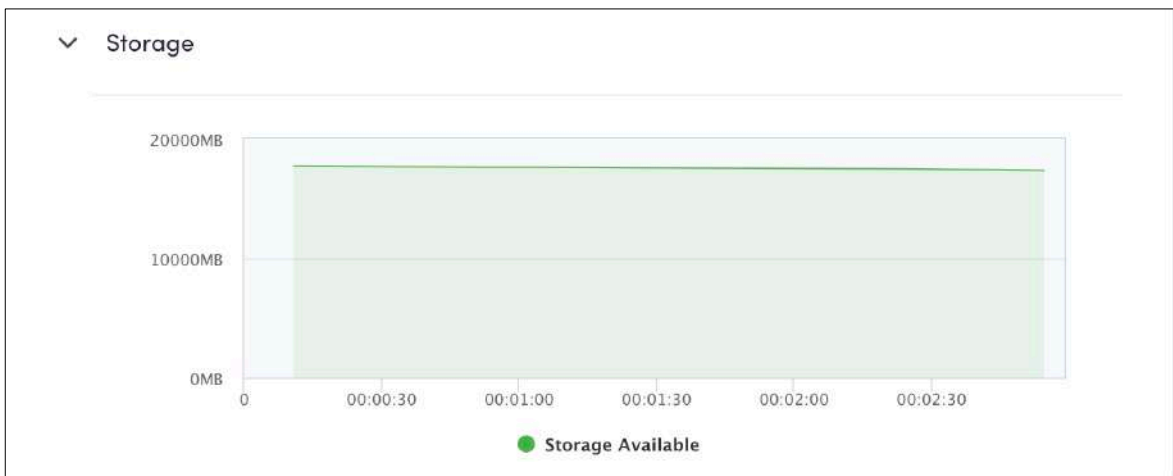


Figura 167 – Gráfico del almacenamiento disponible de la segunda prueba

CAPÍTULO 5 – CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El equipo de desarrollo definió las historias de usuario clasificándolas por cada actor del sistema obteniendo una mejor abstracción de las necesidades que tendría cada uno durante el uso de la aplicación móvil y del aplicativo web, estas historias de usuario definieron nuestro producto backlog estimándolas mediante el uso de la técnica T-Shirt Sizing, llegando a conclusión que esta técnica de estimación de tiempo, combinada con la experiencia del equipo de desarrollo es conveniente para proyectos similares.
- La arquitectura de microservicios utilizada en este proyecto tiene una comunicación directa cliente – microservicio que utilizó dos microservicios uno para manejar los audiolibros y otro para las listas de reproducción, por lo que llegamos a la conclusión de que el uso de una arquitectura de microservicios es ideal para el desarrollo de este proyecto, así como también de aplicaciones que no tienen un gran tamaño o gran cantidad de componentes.
- El diseño e implementación de la aplicación móvil como del aplicativo web se las realizó utilizando tecnologías nuevas y de vanguardia como lo es Firebase, React y SwiftUI con lo que podemos concluir que la brecha entre tecnologías es cada vez menor.
- La arquitectura de microservicios utilizada en este proyecto permitió trabajar con diferentes tecnologías de base de datos como PostgreSQL y Firebase para el almacenamiento de los datos, por lo que podemos concluir que las aplicaciones desarrolladas con esta arquitectura pueden acoplarse a las necesidades y a los diferentes tipos de datos y no centrar su desarrollo en un solo tipo de base de datos.
- Las pruebas funcionales que se realizaron a la aplicación móvil como al aplicativo web permitieron obtener un correcto comportamiento del sistema ya que cada prueba realizada se centró en los criterios de aceptación definidos por cada historia de usuario verificando que los

resultados finales eran los esperados, por lo que concluimos que las pruebas funcionales son clave realizar en cada iteración o sprint del proyecto, tomando en cuenta que en cada sprint obtenemos un producto incremental.

- Las pruebas no funcionales fueron realizadas con distintas herramientas como JMeter la cual permitió realizar pruebas de rendimiento a los microservicios obteniendo datos muy favorables para el backend; BlazeMeter junto con JMeter permitió realizar pruebas de carga para el aplicativo web en donde se realizó una grabación de la ejecución de prueba teniendo valores de tiempo de respuesta bajo y rendimiento alto por lo que podemos concluir que nuestra aplicación trabaja dentro de los rangos aceptables establecidos por las métricas de JMeter.
- En cuanto a las pruebas no funcionales de la aplicación móvil se utilizó la herramienta Apptim la cual realiza pruebas de rendimiento a aplicaciones móviles midiendo diferentes métricas preestablecidas concluyendo que las métricas obtenidas en los resultados de nuestra aplicación se basan en las mejores prácticas, en este caso del desarrollo de aplicaciones Apple, obteniendo valores dentro del rango aceptable.
- Como resultado del desarrollo de este proyecto podemos concluir que el uso de una aplicación móvil de audiolibros interactiva puede ser una manera de plantear una solución que permita generar un interés por los libros y atacar la problemática de los niños y jóvenes sobre los hábitos de lectura.

5.2. Recomendaciones

- Se recomienda que al utilizar Docker como contenedor de imágenes ya sea para los microservicios o las bases de datos crear un Docker Hub, el mismo que es un repositorio público en la nube lo cual permitirá evitar pérdidas de las imágenes creadas localmente.

- Es muy importante utilizar y familiarizarse con un sistema de versionamiento como Git desde el inicio del desarrollo de cualquier tipo de software para evitar pérdidas de información.
- Se recomienda en proyectos similares implementar una arquitectura de microservicios ya que su implementación de cada servicio es autónoma y su comunicación con otros servicios se la puede realizar a través de interfaces sencillas.
- Realizar pruebas funcionales y no funcionales a un proyecto de software es muy recomendable para que permita comprobar las características y a su vez mejorar si es necesario el resultado esperado por el usuario lo que nos da como resultado es un producto con un número de errores bajo obteniendo un código más limpio.

REFERENCIAS BIBLIOGRÁFICAS

- [1] E. A. Nuñez, “Qué son Microservicios y ejemplos reales de uso ,” *OpenWebinars*, Apr. 19, 2016. <https://openwebinars.net/blog/microservicios-que-son/> (accessed Mar. 17, 2022).
- [2] E. Novoseltseva, “Beneficios Y Ejemplos de la implementación de los microservicios,” Apr. 18, 2020. <https://apiumhub.com/es/tech-blog-barcelona/los-microservicios/> (accessed Mar. 17, 2022).
- [3] P. Delgado, L. Salmerón, and C. Vargas, “La lectura digital, en desventaja ,” Dec. 2019. <https://www.investigacionyciencia.es/revistas/mente-y-cerebro/el-inconsciente-sale-a-la-luz-783/la-lectura-digital-en-desventaja-18011> (accessed Mar. 17, 2022).
- [4] S. Garcia-Bulle, “Cinco ventajas didácticas de los audiolibros — Observatorio | Instituto para el Futuro de la Educación,” Oct. 10, 2019. <https://observatorio.tec.mx/edu-news/ventajas-didacticas-de-los-audiolibros> (accessed Mar. 17, 2022).
- [5] AWS, “¿Qué son los microservicios? ,” 2021. <https://aws.amazon.com/es/microservices/> (accessed Mar. 17, 2022).
- [6] Decide, “Arquitectura de microservicios: qué es, ventajas y desventajas,” Sep. 03, 2019. <https://decidesoluciones.es/arquitectura-de-microservicios/> (accessed Mar. 17, 2022).
- [7] I. Alexis Fernando Saransig Chiza, “ANÁLISIS DE RENDIMIENTO ENTRE UNA ARQUITECTURA MONOLÍTICA Y UNA ARQUITECTURA DE MICROSERVICIOS–TECNOLOGÍA BASADA EN CONTENEDORES,” Ibarra, 2018.
- [8] AltexSoft, “Using Microservices for Legacy System Modernization.” <https://www.altexsoft.com/blog/engineering/using-microservices-for-legacy-system-modernization/> (accessed Dec. 26, 2022).
- [9] S. Newman, “Monolith to Microservices Evolutionary Patterns to Transform Your Monolith,” 2019.
- [10] P. E. De la Cruz Vélez de Villa, M. H. Espinoza Ramirez, and O. Cuba Estrella, “Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional

- de Obstetricia de la UNMSM,” *Hamut’Ay*, vol. 6, no. 2, pp. 141–158, 2019, doi: 10.21503/hamu.v6i2.1781.
- [11] E. Fernández, “Que son los Microservicios. Características y Ventajas,” Jun. 16, 2021. <https://www.incentro.com/es-ES/blog/que-son-microservicios> (accessed May 04, 2022).
- [12] Microsoft Learn, “Diferencias entre el patrón de puerta de enlace de API y la comunicación directa de cliente a microservicio.” <https://learn.microsoft.com/es-es/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern> (accessed Oct. 08, 2022).
- [13] C. Anderson, “The Model-View-ViewModel (MVVM) Design Pattern,” 2012.
- [14] Microsoft Learn, “El patrón Model-View-ViewModel - Xamarin |.” <https://learn.microsoft.com/es-es/xamarin/xamarin-forms/enterprise-application-patterns/mvvm> (accessed Dec. 19, 2022).
- [15] R. Alberto Garita-Araya, “E-Ciencias de la Información Tecnología Móvil: desarrollo de sistemas y aplicaciones para las Unidades de Información,” 2013. [Online]. Available: <http://ebci.ucr.ac.crhttp://revistaebci.ucr.ac.cr/>
- [16] S. Lujan, “Programación de Aplicaciones Web: Historia, Principios Básicos y Clientes Web.,” 2002.
- [17] “JSON.” <https://www.json.org/json-es.html> (accessed Dec. 20, 2022).
- [18] AWS, “¿Qué es una API? - Guía sobre las API para principiantes -.” <https://aws.amazon.com/es/what-is/api/> (accessed Dec. 20, 2022).
- [19] P. Fin De Carrera, D. V. Trujillo, and I. G. Carrasco, “Grado en Ingeniería Informática,” 2014.
- [20] Asana, “Descubre cómo las tallas de camisetas pueden ayudarte con la estimación de los proyectos.” <https://asana.com/es/resources/t-shirt-sizing> (accessed Sep. 28, 2022).
- [21] Clique Studios, “¿Qué es una maqueta? (+Cómo crear una maqueta en 2022).” <https://cliquestudios.com/mockups/> (accessed Oct. 08, 2022).
- [22] Microsoft, “Access SQL: conceptos básicos, vocabulario y sintaxis.” <https://support.microsoft.com/es-es/office/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671#bm1> (accessed Sep. 27, 2022).

- [23] A. Fowler, "NoSQL For Dummies." [https://books.google.com.ec/books?id=g_QwBgAAQBAJ&pg=PA11&dq=no sql&hl=es&sa=X&ved=2ahUKEwiTvoKO4Lb6AhX2STABHf2VC5kQ6AF6BA gDEAI#v=onepage&q=nosql&f=false](https://books.google.com.ec/books?id=g_QwBgAAQBAJ&pg=PA11&dq=no+sql&hl=es&sa=X&ved=2ahUKEwiTvoKO4Lb6AhX2STABHf2VC5kQ6AF6BAgDEAI#v=onepage&q=nosql&f=false) (accessed Sep. 28, 2022).
- [24] M. Trigas, "Metodología Scrum."
- [25] H. Guerrero, "Gestión de las Pruebas Funcionales."
- [26] M. Sánchez, "Pruebas de Software. Fundamentos y Técnicas," 2015.
- [27] A. Banks and E. Porcello, "Learning React: Functional Web Development with React and Redux." <https://books.google.com.ec/books?hl=es&lr=&id=ycTADgAAQBAJ&oi=fnd&pg=PP1&dq=react+&ots=cwWOOkB-hQ&sig=egj8jj7KzON1F3Ti2fXpDijRX4M#v=onepage&q=react&f=false> (accessed Jul. 11, 2022).
- [28] E. Haro, T. Guarda, A. O. Zambrano Peñaherrera, and G. Ninahualpa Quiña, "Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot."
- [29] The Eclipse Foundation, "Eclipse Desktop & Web IDEs." <https://www.eclipse.org/ide/> (accessed Oct. 13, 2022).
- [30] Visual Studio Code, "Visual Studio Code - Code Editing. Redefined." <https://code.visualstudio.com/#meet-intellisense> (accessed Oct. 13, 2022).
- [31] App Store, "Xcode en Mac." <https://apps.apple.com/es/app/xcode/id497799835?mt=12> (accessed Nov. 12, 2022).
- [32] Documentación para desarrolladores de Apple, "Notas de la versión de iOS y iPadOS." <https://developer.apple.com/documentation/ios-ipados-release-notes> (accessed Sep. 12, 2022).
- [33] Documentación para desarrolladores de Apple, "Notas de la versión de iOS 16." <https://developer.apple.com/documentation/ios-ipados-release-notes/ios-16-release-notes> (accessed Sep. 13, 2022).
- [34] Docs Swift, "Compatibilidad de versiones: el lenguaje de programación Swift (Swift 5.7)." <https://docs.swift.org/swift-book/GuidedTour/Compatibility.html> (accessed Sep. 13, 2022).

- [35] Hacking with Swift, “What’s new in Swift 5.7.”
<https://www.hackingwithswift.com/articles/249/whats-new-in-swift-5-7>
(accessed Sep. 13, 2022).
- [36] Apple Developer, “SwiftUI Overview - Xcode.”
<https://developer.apple.com/xcode/swiftui/> (accessed Nov. 12, 2022).
- [37] Git, “Git.” <https://git-scm.com/> (accessed Oct. 12, 2022).
- [38] GitHub, “Features | GitHub ·.” <https://github.com/features> (accessed Nov. 12, 2022).
- [39] Sourcetree app, “Sourcetree | Free Git GUI for Mac and Windows.”
<https://www.sourcetreeapp.com/> (accessed Oct. 08, 2022).
- [40] B. Stonehem, “Google Android Firebase: Learning the Basics.”
https://books.google.com.ec/books?hl=es&lr=&id=JeeODAAAQBAJ&oi=fnd&pg=PP10&dq=firebase&ots=tm8Nw-RBH_&sig=0DXZSyA4_Whb1dhhuHdvANBTeoQ#v=onepage&q&f=false
(accessed Jul. 11, 2022).
- [41] Firebase Documentation, “Cloud Firestore.”
<https://firebase.google.com/docs/firestore> (accessed Jul. 11, 2022).
- [42] Firebase Documentation, “Firebase Authentication.”
<https://firebase.google.com/docs/auth> (accessed Jul. 11, 2022).
- [43] Firebase Documentation, “Cloud Storage para Firebase.”
<https://firebase.google.com/docs/storage> (accessed Jul. 11, 2022).
- [44] PostgreSQL, “PostgreSQL: The world’s most advanced open source database.” <https://www.postgresql.org/> (accessed Oct. 13, 2022).
- [45] AWS, “Contenedores de Docker | ¿Qué es Docker?”
<https://aws.amazon.com/es/docker/> (accessed Jul. 21, 2022).
- [46] Apache JMeter™, “Apache JMeter.” <https://jmeter.apache.org/> (accessed Oct. 05, 2022).
- [47] BlazeMeter by Perforce, “BlazeMeter Continuous Testing.”
<https://www.blazemeter.com/> (accessed Oct. 23, 2022).
- [48] Blog Postman, “Las primeras 5 cosas que debe probar si es nuevo en Postman.” <https://blog.postman.com/postman-first-5-things-to-try/> (accessed Oct. 08, 2022).

- [49] Apptim, "Mobile Performance Testing - Mobile App Testing Tool." <https://www.apptim.com/> (accessed Nov. 30, 2022).
- [50] Justinmind, "Free prototyping tool for web & mobile apps." <https://www.justinmind.com/> (accessed Oct. 08, 2022).
- [51] S. Salazar and D. Ponce, "HÁBITOS DE LECTURA."
- [52] M. Chávez, "Conceptos, Clasificación, Evolucion, efectos de las TICS, ventajas y desventajas, comunidades virtuales, impacto y evolución de servicios. Aplicaciones.," 2019.
- [53] J. I. N. LIMAS CANO DARIO SEBASTIAN, "SOFTWARE MOVIL, AUDIOLIBROS EDUCATIVOS PARA NIÑOS DE 5 AÑOS," *UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS*, 2016.