

**Roger Lluch López**

**Desenvolupament d'una aplicació que permeti planificar un entrenament de resistència**

**TREBALL DE FI DE GRAU**

**Dirigit per Sergi Calatayud Masip**

**Grau d'Enginyeria Informàtica**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona  
2024**

**Resum.**

El present projecte se centra en la creació d'una aplicació web destinada a la gestió i assignació d'entrenaments personalitzats en un entorn educatiu. La justificació del projecte rau en la necessitat d'oferir una eina eficient que permeti als instructors gestionar entrenaments de manera organitzada i personalitzada, així com millorar la interacció i el seguiment dels alumnes. La metodologia aplicada en aquest projecte es basa en que l'entrenador introdueix cançons a l'aplicació a partir d'un URL de YouTube i al backend es descarrega temporalment l'àudio per analitzar-lo i afegir la informació a la base de dades, incloent-hi dades com els BPM i la seva durada. A més, l'entrenador podrà crear entrenaments basant-se en la intensitat de les cançons per poder crear entrenaments personalitzats i assignar-los als alumnes que desitgi.

La tecnologia utilitzada inclou un stack basat en Node.js per al backend, amb una base de dades SQLite per a la gestió de la informació, dos scripts de Python per descarregar i analitzar l'àudio i Angular per al frontend.

Els resultats obtinguts mostren el correcte funcionament que es va plantejar per a l'aplicació en permetre als instructors crear i assignar entrenaments de manera dinàmica i adaptada a les necessitats dels alumnes, mentre que aquests poden accedir fàcilment als entrenaments que els han estat assignats. A més, es van implementar mecanismes de seguretat bàsics per a la gestió d'accés, garantint que només els usuaris autoritzats puguin realitzar certes accions, millorant així la integritat i confidencialitat de la informació gestionada per l'aplicació.

**Resumen.**

El presente proyecto se centra en la creación de una aplicación web destinada a la gestión y asignación de entrenamientos personalizados en un entorno educativo. La justificación del proyecto radica en la necesidad de ofrecer una herramienta eficiente que permita a los instructores gestionar entrenamientos de manera organizada y personalizada, así como mejorar la interacción y el seguimiento de los alumnos.

La metodología aplicada en este proyecto se basa en que el entrenador introduce canciones en la aplicación a partir de un URL de YouTube y en el backend se descarga temporalmente el audio para analizarlo y añadir la información a la base de datos, incluyendo datos como los BPM y su duración. Además, el entrenador podrá crear entrenamientos basándose en la intensidad de las canciones para poder crear entrenamientos personalizados y asignarlos a los alumnos que desee.

La tecnología utilizada incluye un stack basado en Node.js para el backend, con una base de datos SQLite para la gestión de la información, dos scripts de Python para descargar y analizar el audio, y Angular para el frontend.

Los resultados obtenidos muestran el correcto funcionamiento que se planteó para la aplicación al permitir a los instructores crear y asignar entrenamientos de manera dinámica y adaptada a las necesidades de los alumnos, mientras que estos pueden acceder fácilmente a los entrenamientos que les han sido asignados. Además, se implementaron mecanismos de seguridad básicos para la gestión de acceso, garantizando que solo los usuarios autorizados puedan realizar ciertas acciones, mejorando así la integridad y confidencialidad de la información gestionada por la aplicación.

**Abstract.**

The present project focuses on the creation of a web application aimed at managing and assigning personalized training sessions in an educational environment. The justification for the project lies in the need to offer an efficient tool that allows instructors to manage training sessions in an organized and personalized manner, as well as improve the interaction and monitoring of students.

The methodology applied in this project is based on the trainer entering songs into the application using a YouTube URL, and in the backend, the audio is temporarily downloaded to analyze it and add the information to the database, including data such as BPM and duration. Additionally, the trainer will be able to create training sessions based on the intensity of the songs to create personalized workouts and assign them to the desired students.

The technology used includes a stack based on Node.js for the backend, with an SQLite database for information management, two Python scripts for downloading and analyzing the audio, and Angular for the frontend.

The results obtained show the correct functionality that was proposed for the application by allowing instructors to create and assign training sessions dynamically and tailored to the needs of the students, while the students can easily access the training sessions assigned to them. Furthermore, basic security mechanisms were implemented for access management, ensuring that only authorized users can perform certain actions, thus improving the integrity and confidentiality of the information managed by the application.

# Índex

<b>1</b>	<b>INTRODUCCIÓ.....</b>	<b>6</b>
1.1	CONTEXT I JUSTIFICACIÓ DEL PROJECTE.....	6
<b>2</b>	<b>DESCRIPCIÓ DEL PROJECTE I OBJECTIUS DES DEL PUNT DE VISTA FORMATIU.....</b>	<b>8</b>
2.1	PREVISIONS D'ÚS.....	9
<b>3</b>	<b>REQUISITS.....</b>	<b>10</b>
3.1	REQUISITS FUNCIONALS.....	10
3.1.1	<i>Anàlisi dels Requisits Funcionals.....</i>	<i>11</i>
3.2	REQUISITS NO FUNCIONALS.....	27
3.3	REQUISITS ADDICIONALS.....	28
<b>4</b>	<b>DISSENY.....</b>	<b>29</b>
4.1	DISTRIBUCIÓ TEMPORAL.....	29
4.2	ARQUITECTURA DE L'APLICACIÓ.....	30
4.2.1	<i>Capa de Presentació (Frontend).....</i>	<i>30</i>
4.2.2	<i>Capa de Lògica de l'Aplicació (Backend).....</i>	<i>33</i>
4.2.3	<i>Capa de Dades (Base de Dades).....</i>	<i>37</i>
<b>5</b>	<b>IMPLEMENTACIÓ.....</b>	<b>39</b>
5.1	DETALLS DE LA TECNOLOGIA IMPLEMENTADA.....	39
5.1.1	<i>Fontend: Angular, HTML, CSS i TypeScript.....</i>	<i>39</i>
5.1.2	<i>Backend: Node.js, Express o HTTP.....</i>	<i>40</i>
5.1.3	<i>Base de Dades: SQL, SQLite.....</i>	<i>40</i>
5.2	ALGORITMES ESPECÍFICS IMPLEMENTATS.....	41
5.2.1	<i>Algoritme de selecció de música segons els BPM.....</i>	<i>41</i>
5.2.2	<i>Algoritme de descàrrega i anàlisi d'àudio.....</i>	<i>43</i>
5.2.3	<i>Algoritme d'autenticació d'usuaris i gestió de rols.....</i>	<i>45</i>
5.2.4	<i>Algoritme de reproducció contínua de música per a un entrenament.....</i>	<i>46</i>
5.2.5	<i>Algoritmes d'assignació i desassignació d'entrenaments a usuaris.....</i>	<i>50</i>
5.2.6	<i>Algoritme de creació d'entrenaments.....</i>	<i>53</i>
5.2.7	<i>Algoritme d'eliminació d'entrenaments.....</i>	<i>56</i>
5.2.8	<i>Algoritme de cerca i filtratge de cançons a la llista.....</i>	<i>57</i>
<b>6</b>	<b>AVALUACIÓ.....</b>	<b>59</b>
6.1	AVALUACIÓ DELS REQUISITS FUNCIONALS.....	59
6.2	AVALUACIÓ DELS REQUISITS NO FUNCIONALS.....	63
<b>7</b>	<b>PROCÉS NORMAL D'EXECUCIÓ.....</b>	<b>65</b>
7.1	INICI SERVIDORS.....	65
7.2	VISTA ROL "ALUMNE".....	66
7.3	VISTA ROL "PROFESSOR" O "ADMINISTRADOR".....	69
<b>8</b>	<b>CONCLUSIONS.....</b>	<b>78</b>
<b>9</b>	<b>BIBLIOGRAFIA.....</b>	<b>79</b>
<b>10</b>	<b>ANNEXES.....</b>	<b>80</b>
10.1	GUIA DE POSADA EN MARXA.....	80
10.2	CODI DE L'APLICACIÓ.....	83

**Índex de taules**

TAULA 1. FUNCIONS COMPONENT “SELECCIONAR-ACCIÓ” .....	30
TAULA 2. FUNCIONS COMPONENT “LOGIN” .....	31
TAULA 3. FUNCIONS COMPONENT “SELECCIONAR-MUSICA” .....	31
TAULA 4. FUNCIONS COMPONENT “PUJAR-MUSICA” .....	31
TAULA 5. FUNCIONS COMPONENT “SELECCIONAR-ENTRENAMENT” .....	32
TAULA 6. FUNCIONS COMPONENT “CREAR-ENTRENAMENT” .....	33
TAULA 7. FUNCIONS <i>DELETE</i> DEL BACKEND.....	33
TAULA 8. FUNCIONS <i>POST</i> DEL BACKEND.....	34
TAULA 9. FUNCIONS <i>GET</i> DEL BACKEND.....	34
TAULA 10. TAULES DE LA BASE DE DADES.....	37

## Índex de figures

FIGURA 1. EXEMPLE DE L'ANÀLISI INCORRECTE DE LA CANÇÓ "TILL I COLAPSE" A LA WEB WWW.SONGBPM.COM.....	6
FIGURA 2. BPM DE LA CANÇÓ <i>TILL I COLLAPSE</i> SEGONS EL PROGRAMA REKORDBOX.....	8
FIGURA 3. BPM DE LA CANÇÓ <i>TILL I COLLAPSE</i> DESPRÉS DE SER ANALITZADA AMB EL SCRIPT DE PYTHON.....	8
FIGURA 4. DIAGRAMA DE SEQÜÈNCIES RF1.....	11
FIGURA 5. DIAGRAMA D'ACTIVITATS RF1.....	12
FIGURA 6. DIAGRAMA DE SEQÜÈNCIES RF2.....	13
FIGURA 7. DIAGRAMA D'ACTIVITATS RF2.....	13
FIGURA 8. DIAGRAMA DE SEQÜÈNCIES RF3.....	14
FIGURA 9. DIAGRAMA D'ACTIVITATS RF3.....	15
FIGURA 10. DIAGRAMA DE SEQÜÈNCIES RF4.....	17
FIGURA 11. DIAGRAMA D'ACTIVITATS RF4.....	18
FIGURA 12. DIAGRAMA DE SEQÜÈNCIES RF5.....	19
FIGURA 13. DIAGRAMA D'ACTIVITATS RF5.....	20
FIGURA 14. DIAGRAMA DE SEQÜÈNCIES RF6.1.....	21
FIGURA 15. DIAGRAMA D'ACTIVITATS RF6.1.....	21
FIGURA 16. DIAGRAMA DE SEQÜÈNCIES RF6.2.....	22
FIGURA 17. DIAGRAMA D'ACTIVITATS RF6.2.....	22
FIGURA 18. DIAGRAMA DE SEQÜÈNCIES RF6.3.....	23
FIGURA 19. DIAGRAMA D'ACTIVITATS RF6.3.....	24
FIGURA 20. DIAGRAMA DE SEQÜÈNCIES RF6.4.....	25
FIGURA 21. DIAGRAMA D'ACTIVITATS RF6.4.....	25
FIGURA 22. DIAGRAMA DE SEQÜÈNCIES RF6.5.....	26
FIGURA 23. DIAGRAMA D'ACTIVITATS RF6.5.....	27
FIGURA 24. DISTRIBUCIÓ TEMPORAL.....	29
FIGURA 25. ESQUEMA DE L'ARQUITECTURA DEL PROJECTE.....	30
FIGURA 26. ESTRUCTURA DE LA BASE DE DADES.....	38
FIGURA 27. INICIALITZACIÓ DEL BACKEND.....	65
FIGURA 28. FRONTEND COMPILANT-SE.....	65
FIGURA 29. FRONTEND COMPILAT CORRECTAMENT.....	65
FIGURA 30. PÀGINA D'INICI DE SESSIÓ.....	66
FIGURA 31. CREDENCIALS DE L'USUARI <i>ALBERT</i> .....	66
FIGURA 32. REGISTRE DE L'USUARI <i>ALBERT</i> A LA TAULA "USUARIS".....	66
FIGURA 33. REGISTRES AMB L'ID DEL USUARI <i>ALBERT</i> (ID 2) A LA TAULA "USERENTR".....	67
FIGURA 34. VISTA DE L'USUARI <i>ALBERT</i> UN COP ESTÀ CARREGADA CORRECTAMENT.....	67
FIGURA 35. DETALLS DE L'ENTRENAMENT "TEST 4".....	68
FIGURA 36. ENTRENAMENT "TEST 4" REPRODUINT LA PRIMERA CANÇÓ DE L'ENTRENAMENT.....	68
FIGURA 37. VISTA QUAN L'USUARI <i>ALBERT</i> INTENTA ACCEDIR A LA SECCIÓ DEL PANNEL PER A SELECCIONAR ACCIÓ EDITANT L'URL.....	69
FIGURA 38. CREDENCIALS USUARI <i>SERGI</i> .....	69
FIGURA 39. REGISTRE DE L'USUARI <i>SERGI</i> A LA TAULA "USUARIS".....	70
FIGURA 40. VISTA DE L'ENTRENADOR EN INICIAR SESSIÓ.....	70
FIGURA 41. VISTA DE LA LLISTA DE CANÇONS.....	70
FIGURA 42. CANÇONS QUE COINCIDEIXEN AMB ELS FILTRES D'INTENSITAT MITJANA I BPM ENTRE 70 I 100.....	71
FIGURA 43. VISTA DE LA WEB MENTRE DESCARREGA I ANALITZA L'ÀUDIO.....	71
FIGURA 44. CANÇÓ ANALITZADA I LLESTA PER A SER INSERIDA A LA BASE DE DADES.....	72
FIGURA 45. COMPROVACIÓ DE QUÈ LA CANÇÓ S'HA INSERIT CORRECTAMENT A LA BASE DE DADES.....	72
FIGURA 46. INTRODUCCIÓ DELS PARÀMETRES DE L'ENTRENAMENT.....	73
FIGURA 47. LLISTA DE CANÇONS PROPOSADA PER A L'ENTRENAMENT CREAT.....	74
FIGURA 48. ASSIGNACIÓ DE L'ENTRENAMENT A L'ALUMNE <i>ALBERT</i> .....	75
FIGURA 49. VISTA DELS ENTRENAMENTS PER A L'USUARI <i>ALBERT</i> .....	75
FIGURA 50. ELIMINACIÓ DE L'ENTRENAMENT.....	76
FIGURA 51. VISTA DELS ENTRENAMENTS PER AL PROFESSOR DESPRÉS D'ELIMINAR L'ENTRENAMENT <i>TEST EXECUCIÓ</i> .....	76
FIGURA 52. VISTA DELS ENTRENAMENTS PER A L'USUARI <i>ALBERT</i> DESPRÉS D'ELIMINAR L'ENTRENAMENT <i>TEST EXECUCIÓ</i> .....	77
FIGURA 53. PÀGINA DE DESCÀRREGA DE NODE.JS.....	80
FIGURA 54. SELECCIÓ DE LA CARPETA D'INSTAL·LACIÓ PER A NODE.JS.....	80

FIGURA 55. PROCÉS D'INSTAL·LACIÓ DE NODE.JS.....	81
FIGURA 56. EXTENSIÓ RECOMANADA PER A APLICACIONS D'ANGULAR.....	81
FIGURA 57. COMANDA PER A INSTAL·LAR ELS PAQUETS NECESSARIS DEL FRONTEND.....	82
FIGURA 58. COMANDES PER A NAVEGAR A LA CARPETA DEL BACKEND I INSTAL·LAR ELS PAQUETS NECESSARIS.....	82
FIGURA 59. COMANDA PER A INICIAR EL SERVIDOR DEL BACKEND.....	82
FIGURA 60. COMANDA PER A INICIAR EL SERVIDOR DE DESENVOLUPAMENT DEL FRONTEND.....	83

## Índex de codis

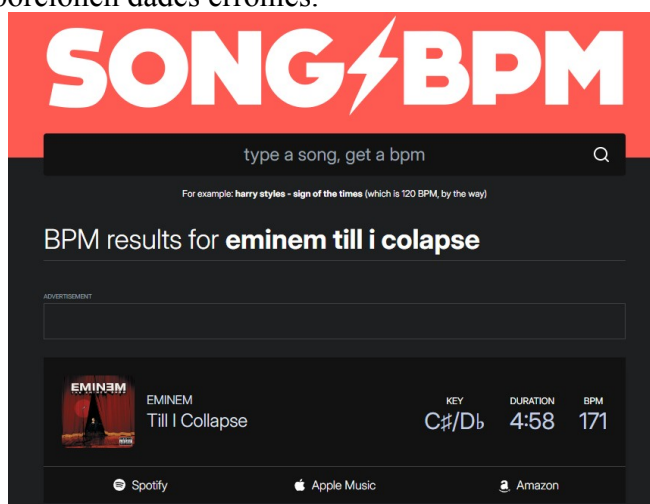
CODI 1. SCRIPT DESCARREGARAUDIO.PY.....	35
CODI 2. SCRIPT ANALITZARAUDIO.PY.....	36
CODI 3. SCRIPT XIFRARPASS.JS.....	37
CODI 4. FUNCIO PLENARCANCONS() DEL FITXER CREAR-ENTRENAMENT.COMPONENT.TS.....	42
CODI 5. FUNCIO VALIDARCAMPS() DEL FITXER CREAR-ENTRENAMENT.COMPONENT.TS.....	42
CODI 6. CODI PER OBTENIR LES CANÇONS DE LA INTENSITAT SELECCIONADA DEL FITXER SERVER.JS.....	43
CODI 7. FUNCIO BUSCARCANCO() DEL FITXER PUJAR-MUSICA.COMPONENT.TS.....	44
CODI 8. CODI PER A DESCARREGAR I ANALITZAR L'ÀUDIO DE L'URL PROPORCIONAT DEL FITXER SERVER.JS.....	45
CODI 9. FUNCIO LOGIN() DEL FITXER LOGIN.COMPONENT.TS.....	45
} ) ; CODI 10. CODI PER A COMPROVAR LES CREDENCIALS DEL FITXER SERVER.JS.....	46
CODI 11. FUNCIO REPRODUIRENTRENAMENT() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	47
CODI 12. FITXER YOUTUBE.D.TS.....	48
CODI 13. FUNCIO INCIARAPIYOUTUBE() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	48
CODI 14. FUNCIO CARREGARVIDEO() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	49
CODI 15. FUNCIO CANVIESTAT() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	49
CODI 16. FUNCIO VIDEOID() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	49
CODI 17. CODI PER A OBTENIR EN ORDRE LA LLISTA DE CANÇONS D'UN ENTRENAMENT DEL FITXER SERVER.JS.....	49
CODI 18. FUNCIO MOSTRARALUMNES() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	50
CODI 19. FUNCIO ASSIGNARENTRENAMENT() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	50
CODI 20. FUNCIO MOSTRARASSIGNATS() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	51
CODI 21. FUNCIO DESASSIGNARENTRENAMENT() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	51
CODI 22. CODI PER A CONSULTAR USUARIS QUE NO TENEN L'ENTRENAMENT ASSIGNAT DEL FITXER SERVER.TS.....	52
CODI 23. CODI PER A CONSULTAR ASSIGNAR L'ENTRENAMENT A UN ALUMNE DEL FITXER SERVER.TS.....	52
CODI 24. CODI PER A CONSULTAR USUARIS QUE TENEN L'ENTRENAMENT ASSIGNAT DEL FITXER SERVER.TS.....	52
CODI 25. CODI PER A CONSULAR DESASSIGNAR L'ENTRENAMENT A UN ALUMNE DEL FITXER SERVER.TS.....	53
CODI 26. FUNCIO CREARENTRENAMENT() DEL FITXER CREAR-ENTRENAMENT.COMPONENT.TS.....	54
CODI 27. CODI PER A CREAR UN ENTRENAMENT DEL FITXER SERVER.JS.....	55
CODI 28. CODI PER A INSERIR L'ORDRE DE LES CANÇONS DEL FITXER SERVER.JS.....	55
CODI 29. FUNCIO ELIMINARENTRENAMENT() DEL FITXER SELECCIONAR-ENTRENAMENT.COMPONENT.TS.....	56
CODI 30. CODI PER ELIMINAR UN ENTRENAMENT DEL FITXER SERVER.JS.....	57
CODI 31. FUNCIO FILTRARLLISTA() DEL FITXER SELECCIONAR-MUSICA.COMPONENT.TS.....	57
CODI 32. FUNCIO APLICARFILTRES() DEL FITXER SELECCIONAR-MUSICA.COMPONENT.TS.....	58
CODI 33. FUNCIO NETEJARFILTRES() DEL FITXER SELECCIONAR-MUSICA.COMPONENT.TS.....	58

# 1 Introducció

## 1.1 Context i justificació del projecte

En l'era digital actual, els entrenadors s'enfronten al repte de personalitzar entrenaments que no només siguin efectius sinó també motivadors per als seus clients. Un element clau en aquesta personalització és l'ús de la música, que s'ha demostrat que té un gran impacte en la motivació i el rendiment dels entrenaments. No obstant això, la selecció manual de música que s'ajusti a les necessitats específiques de cada entrenament, com el ritme o la intensitat, pot ser una tasca complexa i que requereix molt de temps.

Aquest procés, que implica triar manualment cada cançó segons la intensitat requerida, pot esdevenir llarg i tediós, sobretot quan s'ha de fer per a múltiples usuaris. A més, les eines disponibles actualment per obtenir informació precisa sobre els BPM<sup>1</sup> de les cançons no sempre són fiables, ja que s'ha detectat que algunes pàgines web especialitzades proporcionen dades errònies.



**Figura 1.** Exemple de l'anàlisi incorrecte de la cançó “*till i colapse*” a la web [www.songbpm.com](http://www.songbpm.com).

Aquesta inexactitud pot afectar directament la qualitat de l'entrenament, pel fet que la música incorrecta pot alterar el ritme i l'eficàcia de les activitats físiques.

Aquest projecte neix amb l'objectiu de solucionar aquestes problemàtiques mitjançant el desenvolupament d'una aplicació web que permeti als entrenadors automatitzar la creació i l'assignació d'entrenaments amb música. L'eina facilita la creació d'entrenaments basant-se en criteris objectius com la durada i la intensitat de l'activitat, proporcionant així una experiència d'usuari optimitzada tant per a entrenadors com per a esportistes. Amb aquesta aplicació, els entrenadors poden crear entrenaments de manera ràpida i eficaç, estalviant temps i garantint una major coherència en les sessions que dissenyen.

El projecte es fonamenta en la necessitat de millorar la precisió i l'eficiència en la gestió dels entrenaments, alhora que es resolen els problemes derivats de la informació incorrecta sobre els BPM. Mitjançant la implementació d'un sistema propi per analitzar els BPM de les cançons, l'aplicació assegura que la música seleccionada s'ajusta perfectament a cada fase de l'entrenament, millorant així l'experiència general i la motivació dels usuaris.

En definitiva, aquest projecte no només té l'objectiu de facilitar la feina dels entrenadors, sinó que també aporta una solució tecnològica innovadora que potencia la

<sup>1</sup> BPM(Beats Per Minute): unitat que mesura el ritme d'una cançó, indicant quants polsos es produeixen per minut.

qualitat dels entrenaments personalitzats i la seva adaptació a les necessitats específiques de cada usuari.

## 2 Descripció del projecte i objectius des del punt de vista formatiu

Aquest projecte té com a objectiu el desenvolupament d'una aplicació web destinada a facilitar la creació i l'assignació d'entrenaments personalitzats amb música per part dels entrenadors. L'objectiu del desenvolupament s'ha centrat a proporcionar una eina que resolgui les necessitats específiques dels entrenadors, cosa que els permet optimitzar el procés de selecció de cançons per a les diferents fases dels entrenaments basant-se en criteris com la durada i la intensitat de l'exercici.

L'aplicació ofereix un entorn intuïtiu i fàcil d'usar on els entrenadors poden crear entrenaments de manera eficaç, assignar-los a diversos usuaris i garantir que la música seleccionada s'ajusti perfectament a cada moment de l'entrenament. A més, el sistema inclou la capacitat de detectar i corregir les errades freqüents en la informació dels BPM que es troba en altres plataformes, assegurant així una experiència més precisa i fiable. Per a assegurar els resultats, s'han comparat amb eines específiques d'àudio com poden ser els softwares **TraktorPro** o **Rekordbox**.



**Figura 2.** BPM de la cançó *Till i collapse* segons el programa **RekordBox**.

id	títol	duració	bpm
Filtro	till i coll	Filtro	Filtro
2	Till I Collapse	297.9066666666667	86.1328125

**Figura 3.** BPM de la cançó *Till i collapse* després de ser analitzada amb el script de Python.

Des del punt de vista formatiu, aquest projecte té diversos punts clau que personalment opino que es veuran desenvolupats:

- **Desenvolupament d'habilitats tècniques:** A través de la implementació d'aquest projecte, s'adquirirà experiència pràctica en el desenvolupament d'aplicacions web, incloent-hi la integració de tecnologies front-end i back-end, el disseny de bases de dades i la gestió d'APIs externes.
- **Resolució de problemes reals:** El projecte aborda un problema real que afecta els entrenadors, proporcionant una solució tecnològica innovadora que podria ser aplicable en un context professional real.
- **Gestió de projectes:** Aquest projecte em permetrà desenvolupar habilitats en la planificació, execució i avaluació de projectes de desenvolupament de software.
- **Aprofundiment en noves tecnologies:** Tindrè l'oportunitat de treballar amb APIs externes i dades musicals, implementant algorismes per a l'anàlisi de BPM i la seva integració en la generació en entrenaments, cosa que suposa un repte tècnic significatiu.

<sup>2</sup> API (Application Programming Interface): Conjunt de regles i protocols que permeten a diferents aplicacions comunicar-se entre elles i intercanviar dades.

Així, aquest projecte no només té una rellevància pràctica significativa, sinó que també contribueix de manera important a la meua formació tècnica i professional en l'àmbit de l'enginyeria informàtica.

## **2.1 Previsions d'ús**

L'aplicació està dissenyada per ser utilitzada principalment per entrenadors personals, instructors de classes dirigides, i altres professionals del fitness que necessiten crear entrenaments personalitzats per a grups o individus. A més, pot ser útil per a centres d'esport i gimnasos que volen oferir un valor afegit als seus serveis, permetent als seus entrenadors dissenyar sessions d'entrenament altament personalitzades amb facilitat.

En termes tècnics, es preveu que l'aplicació serà utilitzada en un entorn web, accessible tant des de dispositius d'escriptori com mòbils. L'ús de tecnologies de desenvolupament web modernes permetrà que l'aplicació sigui escalable, segura i fàcil de mantenir. A més, en un futur l'aplicació es podria integrar amb altres sistemes o plataformes de fitness existents, com ara aplicacions de monitoratge de l'activitat física.

En resum, aquest projecte ofereix una solució innovadora a un problema comú en el sector de l'esport, amb el potencial de millorar significativament la manera en què els entrenadors personals i altres professionals gestionen i personalitzen els entrenaments per als seus clients.

### 3 Requisites

En aquest apartat es presenten els diferents requisits que han guiat el desenvolupament de l'aplicació.

Primerament, es detallaran els requisits funcionals, que defineixen les funcionalitats específiques que l'aplicació ha de complir. A continuació, s'inclourà una anàlisi d'aquests amb diagrames d'activitats i de seqüències.

Després, es descriuran els requisits no funcionals, que fan referència a les qualitats generals del sistema, com l'eficiència, la seguretat, i la usabilitat.

Finalment, es presentaran els requisits addicionals, que plantegen possibles millores i funcionalitats futures per ampliar les capacitats de l'aplicació.

#### 3.1 Requisites funcionals

- **RF1. Autenticació d'usuari:** L'aplicació ha de permetre als usuaris iniciar sessió introduint el seu nom d'usuari i contrasenya. Només els usuaris autenticats poden accedir a les funcionalitats de l'aplicació en funció del seu rol.
- **RF2. Mostrar llista de cançons:** L'aplicació ha de permetre als entrenadors i administradors consultar la llista de les cançons emmagatzemades a la base de dades. D'aquesta manera, poden decidir si hi ha suficients cançons per a l'entrenament o si es desitja afegir-ne de noves.
- **RF3. Descàrrega i anàlisi de música:** L'aplicació ha de permetre la descàrrega de música des de YouTube i l'anàlisi dels BPM i altres característiques de les cançons a partir de l'URL, per tal de poder-les classificar segons la seva intensitat.
- **RF4. Creació d'entrenaments personalitzats:** L'aplicació ha de permetre als entrenadors poder crear entrenaments personalitzats amb les activitats, intensitat i música personalitzades. Introduint únicament el temps i la intensitat de les activitats, l'aplicació ha d'omplir les activitats amb cançons fins que s'arribi, com a mínim, al temps seleccionat. Una vegada s'ha omplert la llista, s'ha de donar la possibilitat d'editar-la.
- **RF5. Mostrar llista entrenaments en funció del rol:** L'aplicació ha de permetre als usuaris amb rol "alumne" veure únicament els entrenaments que tenen assignats, mentre que els entrenadors i administradors podran veure tots els entrenaments.
- **RF6. Gestió d'entrenaments:** L'aplicació ha de permetre la gestió dels entrenaments segons el rol de l'usuari, incloent-hi les següents funcionalitats:
  - **Mostrar detalls d'un entrenament:** Visualitzar les activitats que componen l'entrenament, la durada de cada una d'elles i les cançons associades.
  - **Reproduir un entrenament:** Permetre la reproducció de les cançons d'un entrenament de manera ininterrompuda, una darrere l'altra.
  - **Assignar o desassignar entrenaments:** Facilitar als entrenadors i administradors assignar o desassignar entrenaments als alumnes quan sigui necessari.
  - **Eliminar entrenaments:** Permetre als entrenadors i administradors eliminar entrenaments del sistema quan aquests ja no siguin necessaris.

### 3.1.1 Anàlisi dels Requisits Funcionals

#### RF1. Autenticació d'usuari:

- Resum de la funcionalitat: L'aplicació permet als usuaris iniciar sessió introduint el seu nom d'usuari i contrasenya. Només els usuaris autenticats poden accedir a les funcionalitats de l'aplicació segons el seu rol.
- Paràmetres d'entrada: Nom d'usuari, Contrasenya.
- Paràmetres de sortida: Token d'autenticació, Rol de l'usuari.
- Actors: Alumne, Professor, Administrador.
- Precondició: Haver estat afegit a la base de dades per l'administrador.
- Postcondició: L'usuari és redirigit a la pàgina corresponent segons el seu rol.
- Procés normal principal:
  1. L'usuari introdueix el seu nom d'usuari i la contrasenya.
  2. El sistema verifica les credencials.
  3. Si les credencials són correctes, el sistema genera un token d'autenticació i identifica el rol de l'usuari. En cas ser incorrectes, mostra un missatge d'error i torna a demanar les credencials.
  4. L'usuari és redirigit a la pàgina principal corresponent al seu rol (alumne o professor/administrador).
- Alternatives del procés i excepcions:
  - Si les credencials són incorrectes, el sistema mostra un missatge d'error.

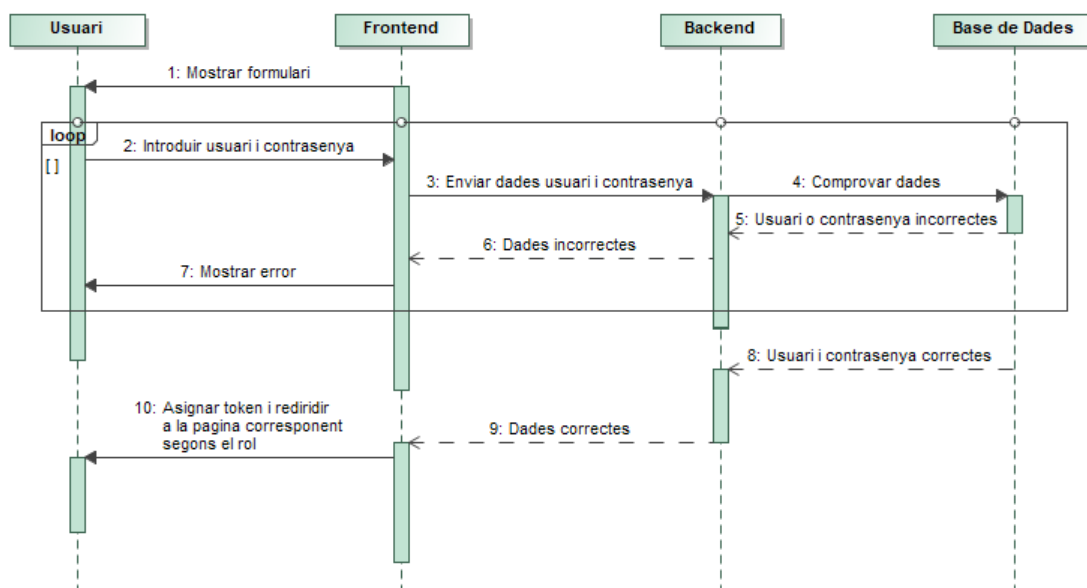
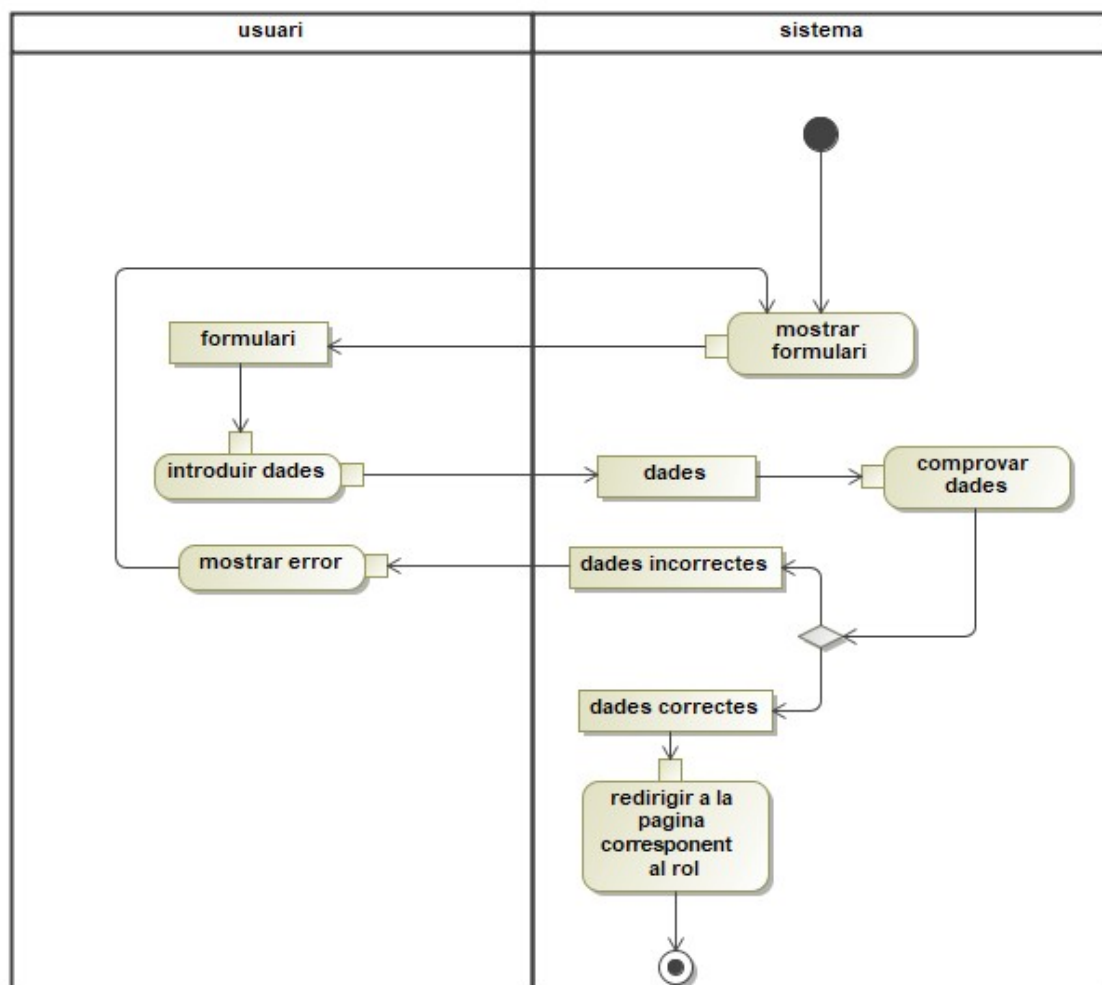


Figura 4. Diagrama de seqüències RF1.



**Figura 5.** Diagrama d'activitats RF1.

### RF2. Mostrar llista de cançons:

- Resum de la funcionalitat: L'aplicació permet als entrenadors i administradors consultar la llista de cançons emmagatzemades a la base de dades. Això permet als usuaris valorar si hi ha suficients cançons per a l'entrenament o si cal afegir-ne de noves.
- Paràmetres d'entrada: Cap.
- Paràmetres de sortida: Llista de cançons (títol, duració, BPM, intensitat).
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1).
- Postcondició: Es mostra la llista de cançons disponibles.
- Procés normal principal:
  1. L'usuari accedeix a la secció de consultar la llista de cançons.
  2. El sistema consulta la base de dades i recupera la llista de cançons.
  3. La llista de cançons es mostra a l'usuari.
- Alternatives del procés i excepcions:
  - Si no es recuperen cançons, es mostra un missatge d'error.

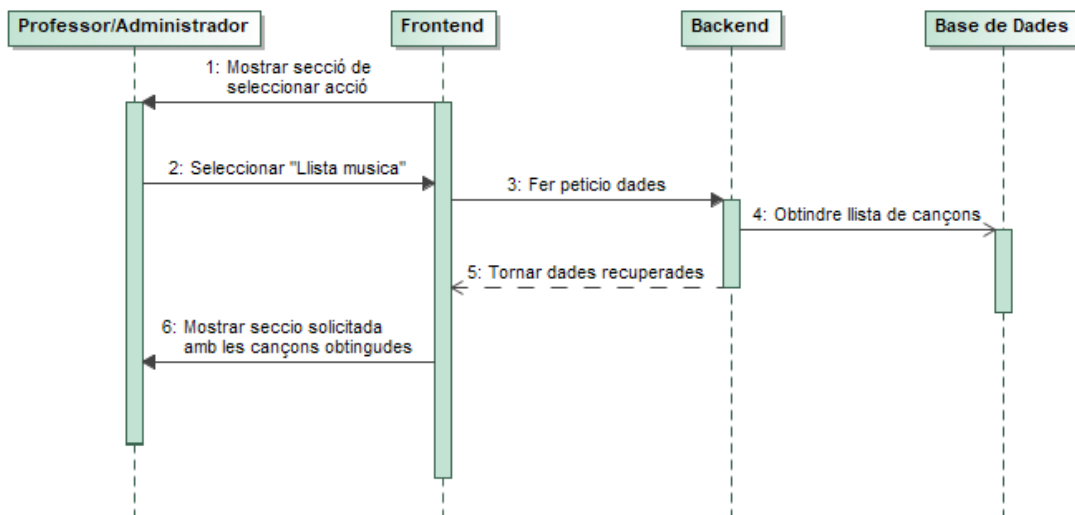


Figura 6. Diagrama de seqüències RF2.

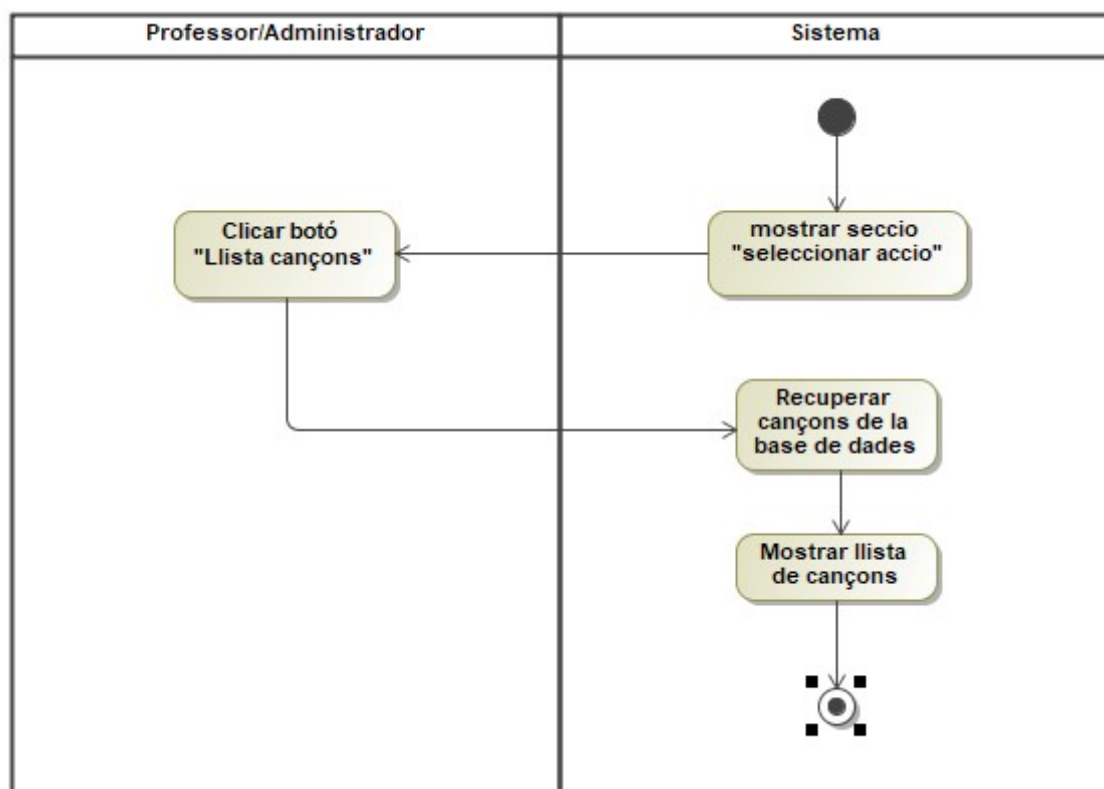
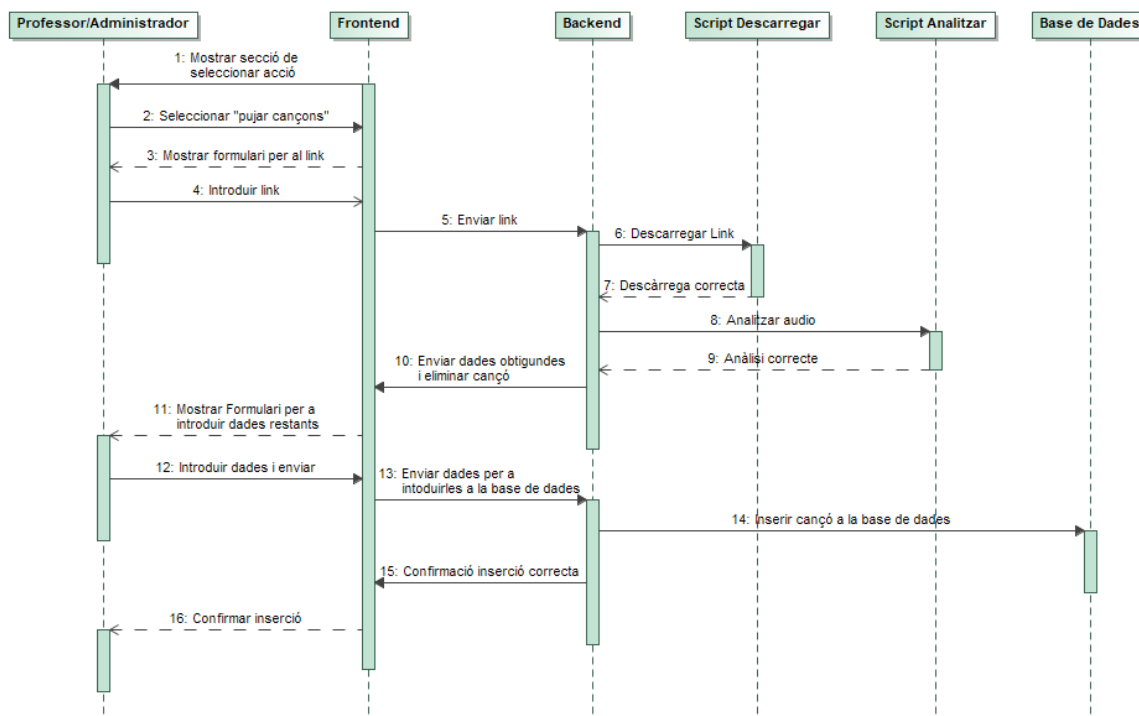


Figura 7. Diagrama d'activitats RF2.

**RF3. Descarrega i anàlisi de música:**

- Resum de la funcionalitat: L'aplicació permet als entrenadors i administradors descarregar música des de YouTube i analitzar les característiques de les cançons, com els BPM, per poder classificar-les segons la intensitat.
- Paràmetres d'entrada: Enllaç de YouTube.
- Paràmetres de sortida: Títol de la cançó, BPM, Duració, Intensitat.
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1).
- Postcondició: La cançó es guarda a la base de dades amb les dades analitzades.
- Procés normal principal:
  1. L'usuari accedeix a la secció de pujar cançons.
  2. L'usuari introdueix l'enllaç de YouTube.
  3. El sistema descarrega l'àudio del vídeo.
  4. El sistema analitza l'àudio per obtenir els BPM i altres característiques.
  5. La cançó i les dades analitzades es desen a la base de dades.
  6. Es confirma a l'usuari que la cançó s'ha afegit correctament.
- Alternatives del procés i excepcions:
  - Si l'enllaç de YouTube no és vàlid, es mostra un missatge d'error.
  - Si hi ha un error en la descàrrega o l'anàlisi de l'àudio, es mostra un missatge d'error i es permet a l'usuari intentar-ho de nou.

**Figura 8.** Diagrama de seqüències RF3.

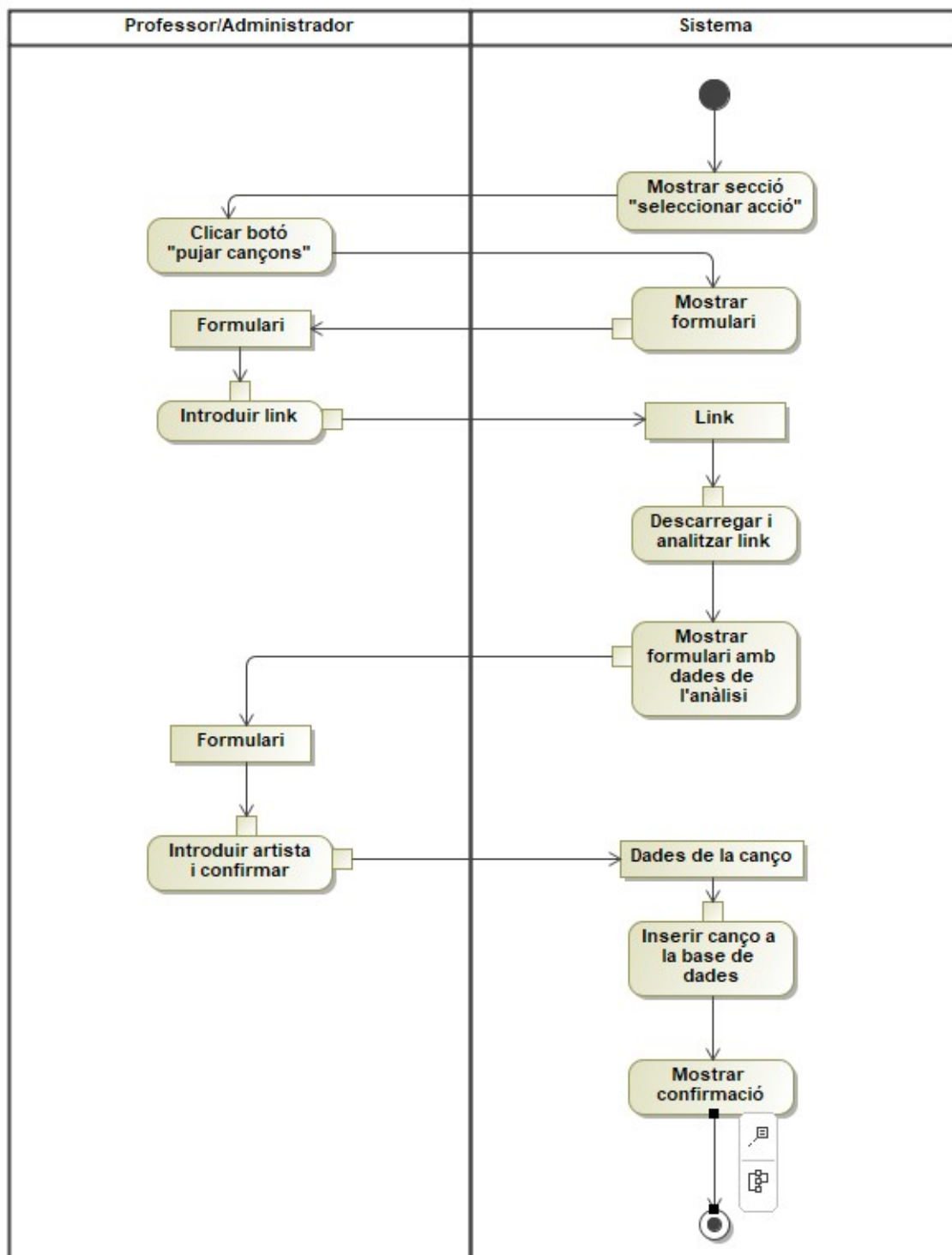


Figura 9. Diagrama d'activitats RF3.

**RF4. Creació d'entrenaments personalitzats:**

- Resum de la funcionalitat: L'aplicació ha de permetre als entrenadors poder crear entrenaments personalitzats amb les activitats, intensitat i música personalitzades. Introduint únicament el temps i la intensitat de les activitats, l'aplicació ha d'omplir les activitats amb cançons fins que s'arribi, com a mínim, al temps seleccionat. Una vegada s'ha omplert la llista, s'ha de donar la possibilitat d'editar cançons.
- Paràmetres d'entrada: Nom de l'entrenament, activitats (temps, intensitat).
- Paràmetres de sortida: Confirmació de la creació de l'entrenament.
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1).
- Postcondició: L'entrenament es desa a la base de dades.
- Procés normal principal:
  1. L'usuari accedeix a la secció de creació d'entrenaments.
  2. Es defineix el nom de l'entrenament i es configuren les activitats del mateix indicant el nombre d'activitats, i el temps i la intensitat d'aquestes.
  3. El sistema replena automàticament el temps de les activitats amb cançons.
  4. L'usuari confirma l'entrenament.
  5. Es guarda l'entrenament a la base de dades.
  6. Es confirma a l'usuari que l'entrenament s'ha creat correctament.
- Alternatives del procés i excepcions:
  - Si l'usuari ho desitja, pot editar les cançons abans de confirmar l'entrenament.
  - Si algun dels camps obligatoris no està complet, es mostra un missatge d'error indicant que s'han d'omplir.
  - Si hi ha un error en desar l'entrenament, es mostra un missatge d'error.

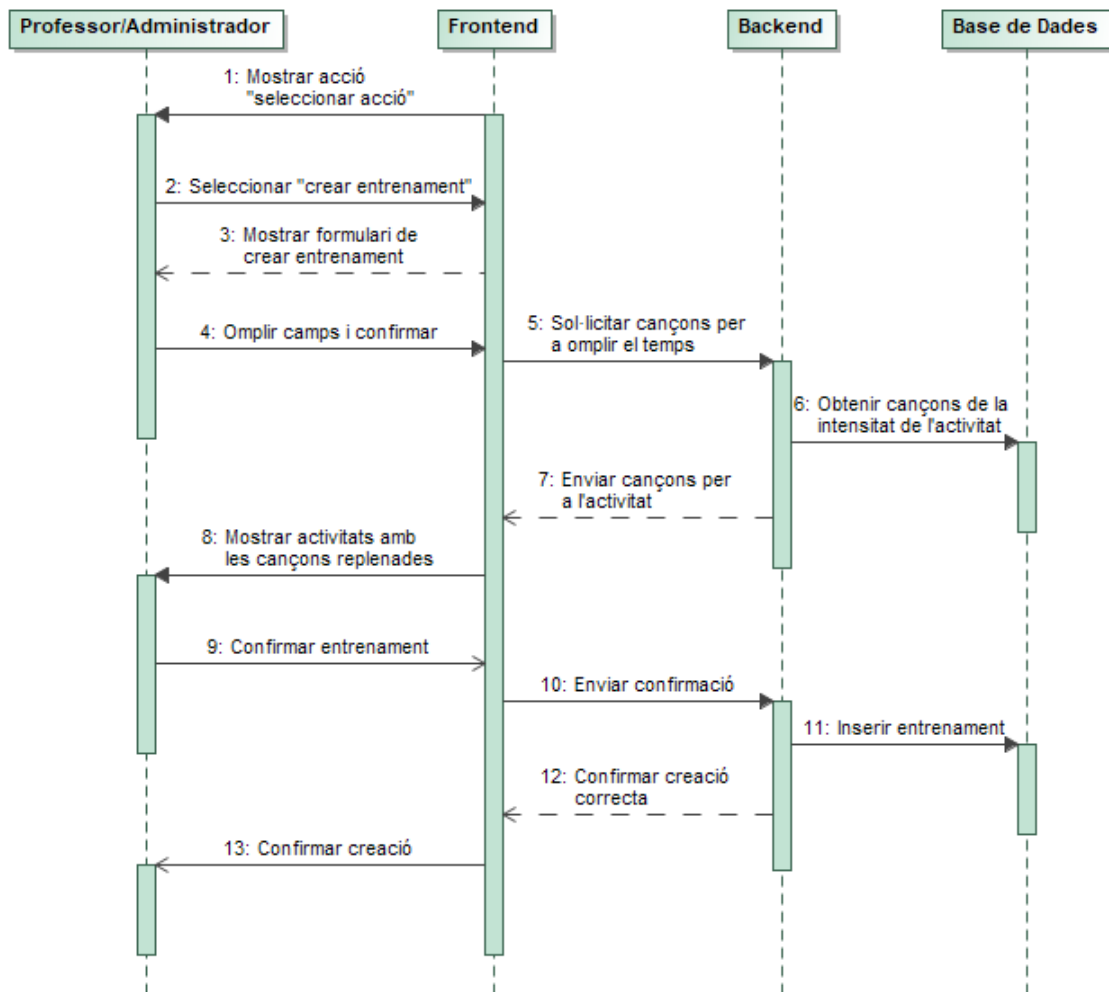


Figura 10. Diagrama de seqüències RF4.

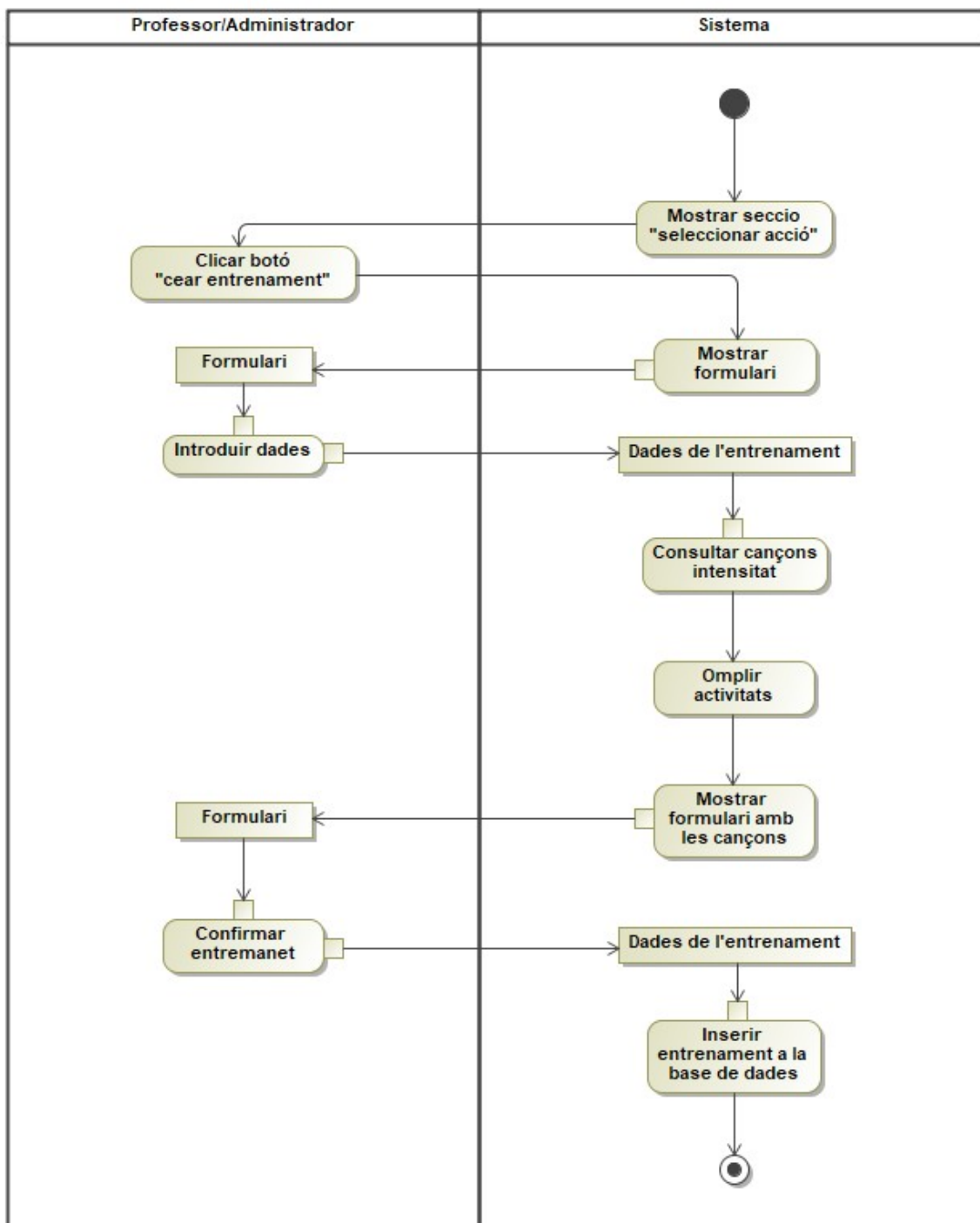


Figura 11. Diagrama d'activitats RF4.

**RF5. Mostrar llista d'entrenaments en funció del rol:**

- Resum de la funcionalitat: L'aplicació permet als usuaris amb rol "alumne" veure només els entrenaments assignats, mentre que els entrenadors i administradors poden veure tots els entrenaments disponibles.
- Paràmetres d'entrada: Cap.
- Paràmetres de sortida: Llista d'entrenaments segons el rol de l'usuari.
- Actors: Alumne, Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1).
- Postcondició: Es mostra la llista d'entrenaments adequada segons el rol de l'usuari.
- Procés normal principal:
  - *Professors i administradors:*
    1. L'usuari accedeix a la secció d'entrenaments.
    2. El sistema consulta la base de dades per a recuperar la llista d'entrenaments.
    3. Es mostra la llista d'entrenaments a l'usuari.
  - *Alumnes:*
    1. Un cop iniciada sessió l'alumne és redirigit directament a aquesta secció.
    2. El sistema consulta els entrenaments assignats a l'alumne i recupera la llista dels entrenaments.
    3. Es mostra la llista d'entrenaments a l'usuari.
- Alternatives del procés i excepcions:
  - Si hi ha un error recuperant la llista d'entrenaments es mostra un missatge d'error.

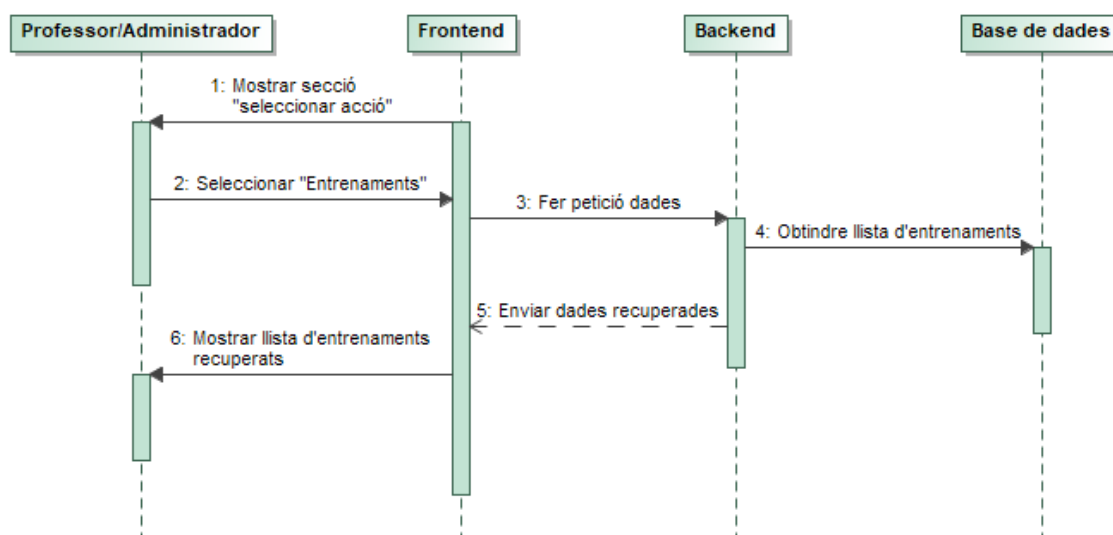


Figura 12. Diagrama de seqüències RF5.

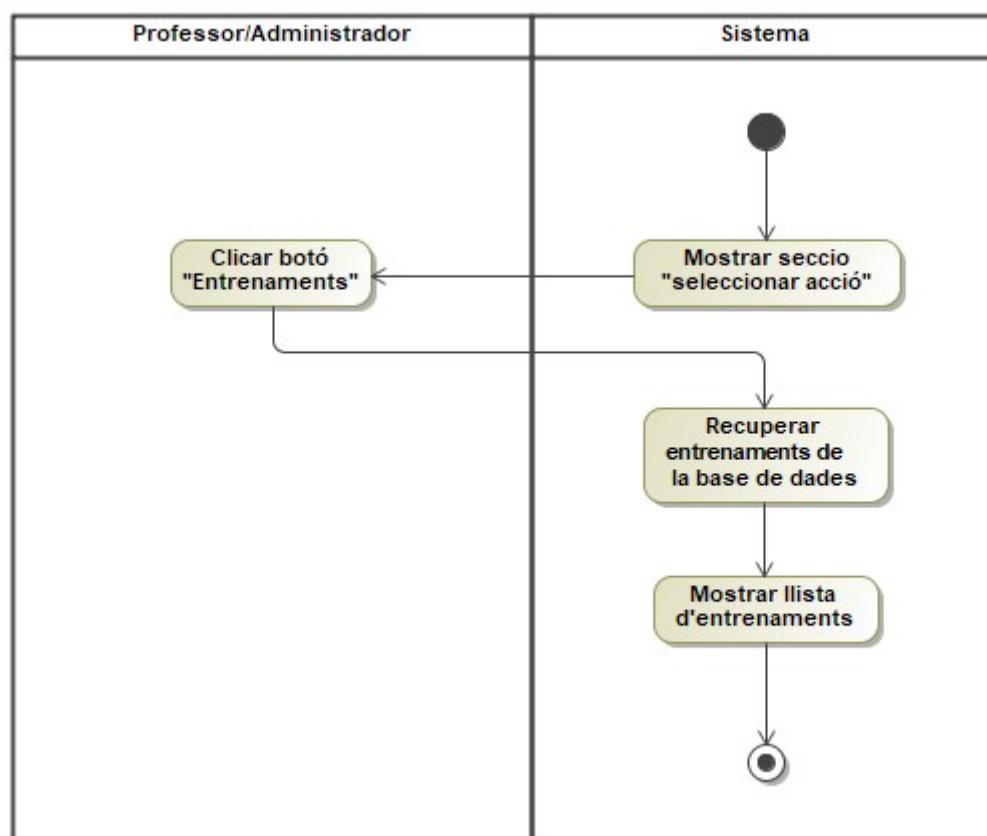


Figura 13. Diagrama d'activitats RF5.

#### RF6. Gestió d'entrenaments:

Donat que aquest requisit inclou diverses funcionalitats relacionades amb la gestió d'entrenaments, es detallarà a continuació cada subfunció suposant que l'usuari ja ha accedit a la secció de consultar entrenaments:

##### RF6.1. *Mostrar detalls d'un entrenament*

- Resum de la funcionalitat: L'aplicació permet visualitzar les activitats que componen l'entrenament, la durada de cadascuna d'elles i les cançons associades.
- Paràmetres d'entrada: Identificador de l'entrenament.
- Paràmetres de sortida: Activitats que componen l'entrenament i, durada i cançons de les activitats.
- Actors: Alumne, Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1) i haver accedit a la secció d'entrenaments (RF5).
- Postcondició: Es mostren els detalls de l'entrenament.
- Procés normal principal:
  1. L'usuari fa clic al botó "mostrar" de la fila de l'entrenament desitjat per a mostrar els detalls de l'entrenament.
  2. El sistema recupera els detalls de l'entrenament de la base de dades.
  3. Es mostren els detalls de l'entrenament en la part inferior de la pàgina.
- Alternatives del procés i excepcions:
  - Si no es poden recuperar els detalls de l'entrenament, es mostra un missatge d'error.

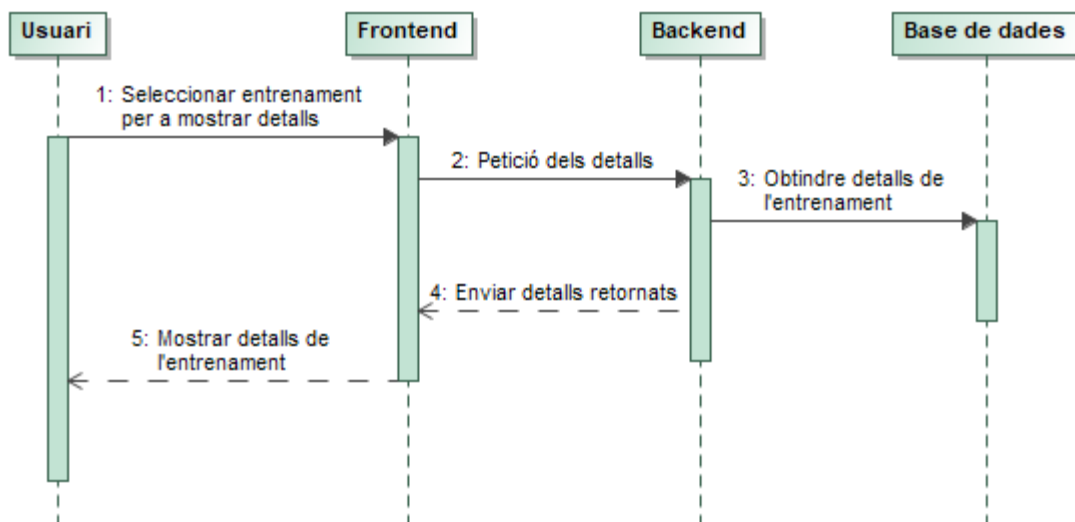


Figura 14. Diagrama de seqüències RF6.1

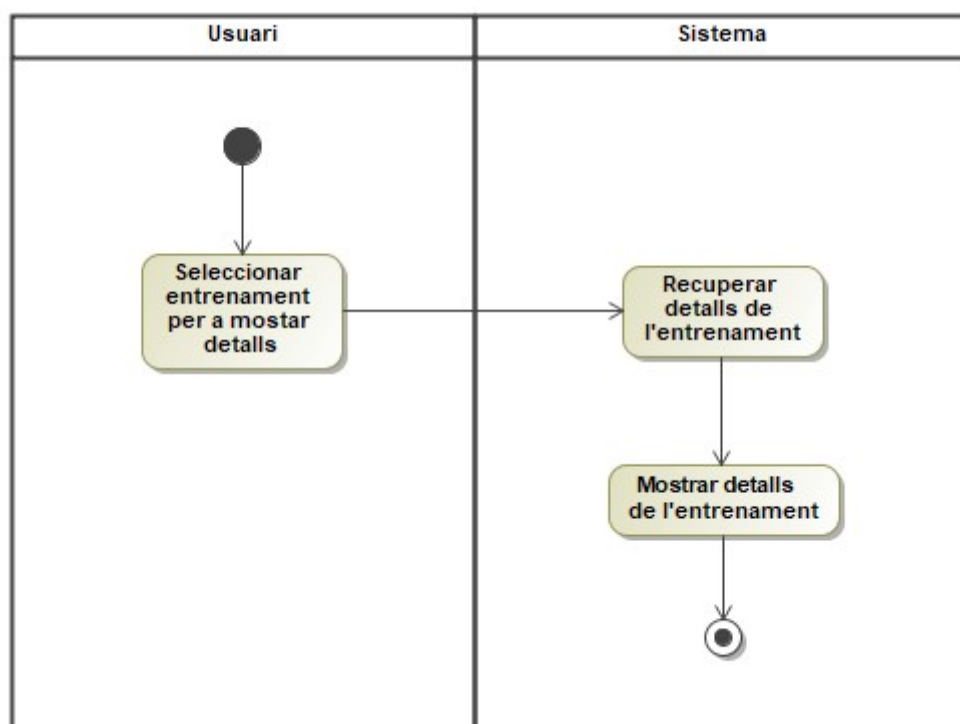


Figura 15. Diagrama d'activitats RF6.1.

### RF6.2. Reproduir un entrenament

- Resum de la funcionalitat: L'aplicació permet la reproducció de les cançons d'un entrenament de manera ininterrompuda, una darrere l'altra.
- Paràmetres d'entrada: Identificador de l'entrenament.
- Paràmetres de sortida: Reproducció de les cançons de l'entrenament de manera ininterrompuda en ordre.
- Actors: Alumne, Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1), haver accedit a la secció d'entrenaments (RF5).
- Postcondició: Les cançons es reproduïen en ordre.

- Procés normal principal:
  1. L'usuari fa clic al botó “reproduir” de la fila de l'entrenament desitjat per a reproduir la música d'aquest.
  2. El sistema recupera els URLs de les cançons associades a l'entrenament amb l'orde que correspon per a cada una.
  3. Es reproduïxen les cançons una darrere l'altra.
- Alternatives del procés i excepcions:
  - Si hi ha algun problema amb l'URL d'alguna cançó es pausa el reproductor.
  - Bug conegut no solucionat: Si el reproductor no està carregat a l'entrar en la secció, és necessari refrescar la pàgina amb “F5”.

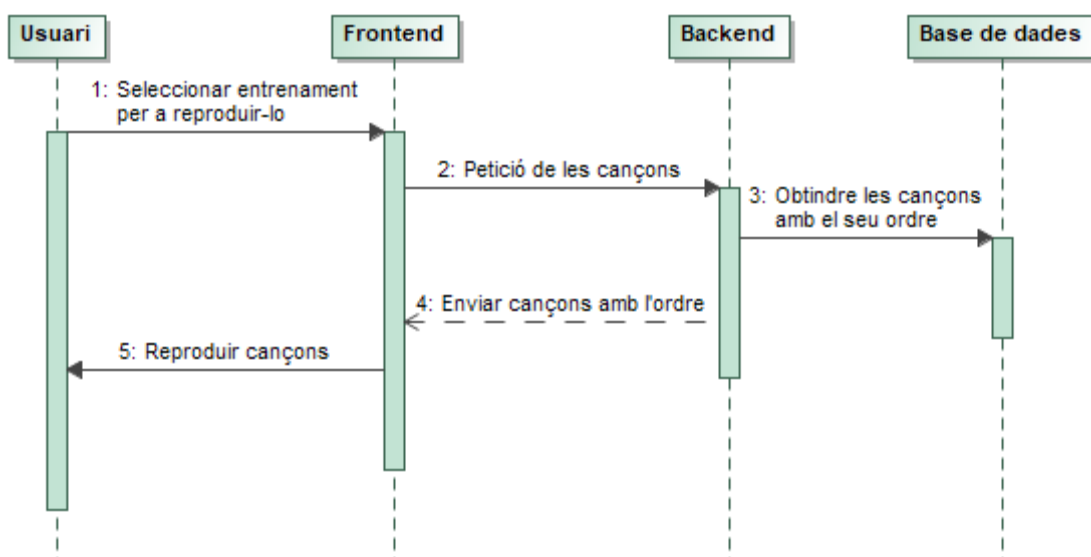


Figura 16. Diagrama de seqüències RF6.2.

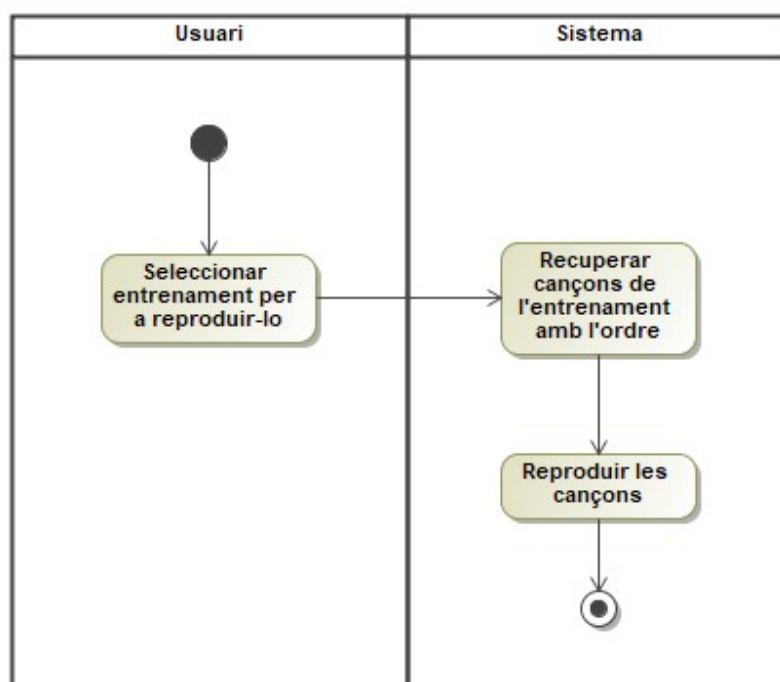
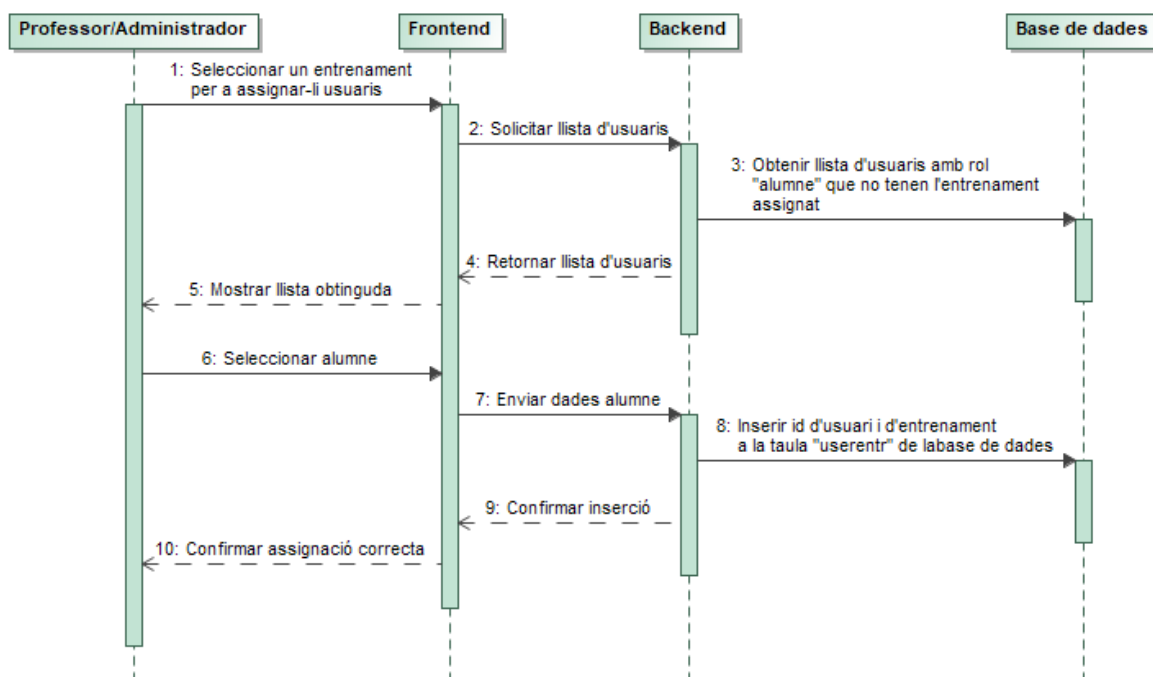
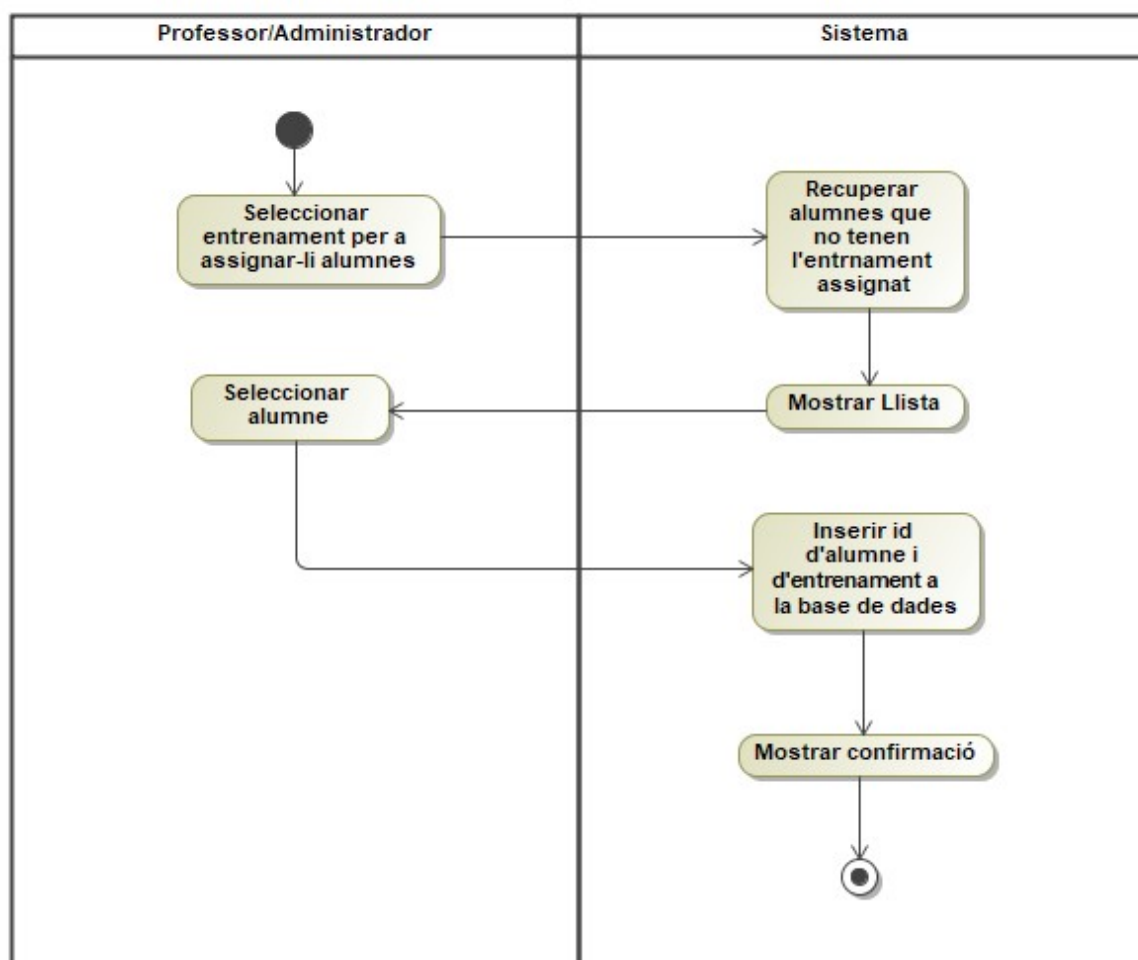


Figura 17. Diagrama d'activitats RF6.2.

**RF6.3. Assignar un entrenament a un alumne.**

- Resum de la funcionalitat: L'aplicació permet als entrenadors i administradors assignar entrenaments a usuaris específics.
- Paràmetres d'entrada: Identificador de l'entrenament, Identificador de l'usuari.
- Paràmetres de sortida: Confirmació d'acció realitzada correctament.
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1) i haver accedit a la secció d'entrenaments (RF5).
- Postcondició: L'entrenament s'assigna a l'alumne.
- Procés normal principal:
  1. El professor o administrador fa clic al botó “assignar” de la fila de l'entrenament corresponent.
  2. Es mostra la llista d'alumnes que no tenen l'entrenament assignat.
  3. L'usuari tria a quin alumne vol assignar l'entrenament.
  4. El sistema insereix els ids de l'usuari i de l'entrenament a la base de dades.
  5. Es confirma que l'entrenament s'ha assignat amb èxit.
- Alternatives del procés i excepcions:
  - Si no s'ha assignat l'entrenament, es mostra un missatge d'error.

**Figura 18.** Diagrama de seqüències RF6.3.



**Figura 19.** Diagrama d'activitats RF6.3

#### ***RF6.4. Desassignar un entrenament a un alumne.***

- Resum de la funcionalitat: L'aplicació permet als entrenadors i administradors desassignar entrenaments d'usuaris específics.
- Paràmetres d'entrada: Identificador de l'entrenament, Identificador de l'usuari.
- Paràmetres de sortida: Confirmació d'acció realitzada correctament.
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1) i haver accedit a la secció d'entrenaments (RF5).
- Postcondició: L'entrenament es desassigna de l'alumne.
- Procés normal principal:
  1. El professor o administrador fa clic al botó “desassignar” de la fila de l'entrenament corresponent.
  2. Es mostra la llista d'alumnes que tenen l'entrenament assignat.
  3. L'usuari tria a quin alumne vol desassignar l'entrenament.
  4. El sistema elimina els ids de l'usuari i de l'entrenament de la base de dades.
  5. Es confirma que l'entrenament s'ha desassignat amb èxit.
- Alternatives del procés i excepcions:
  - Si no s'ha desassignat l'entrenament, es mostra un missatge d'error.

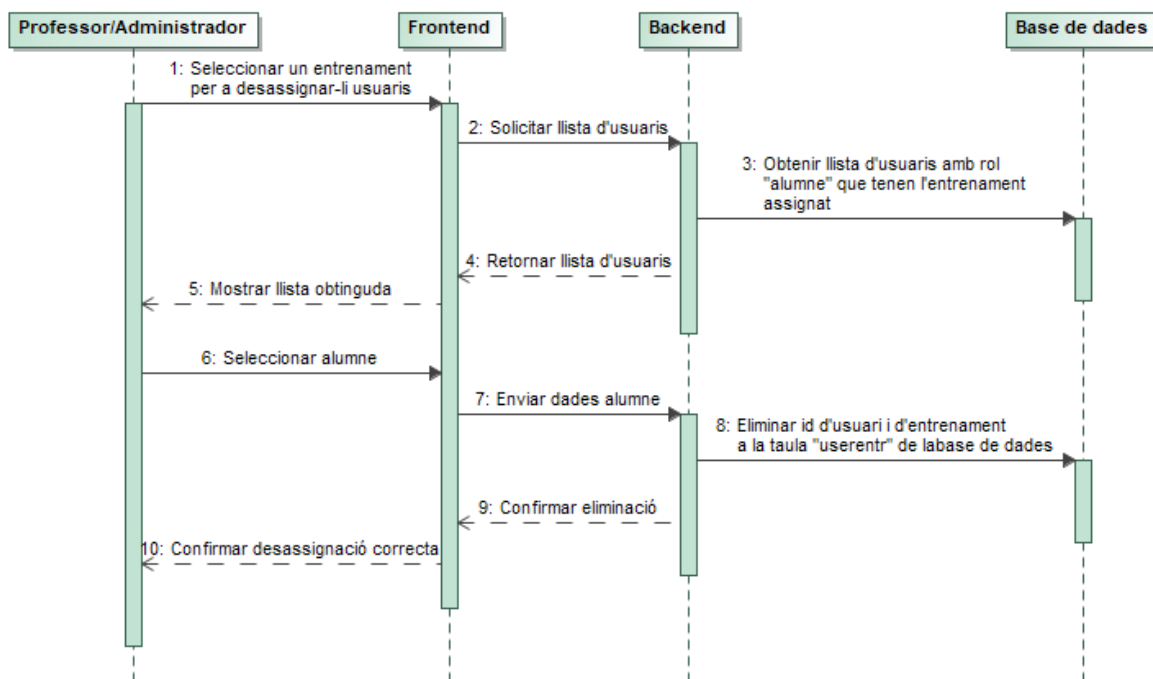


Figura 20. Diagrama de seqüències RF6.4.

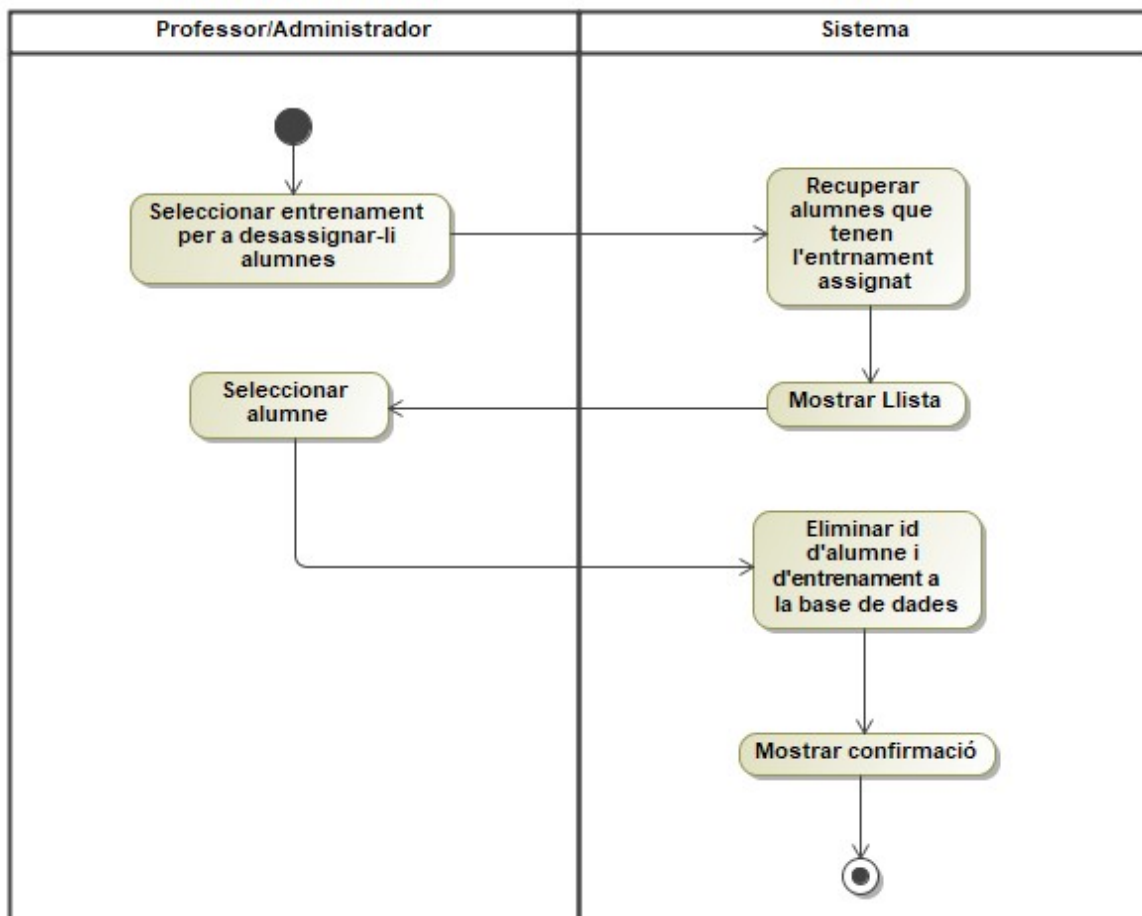
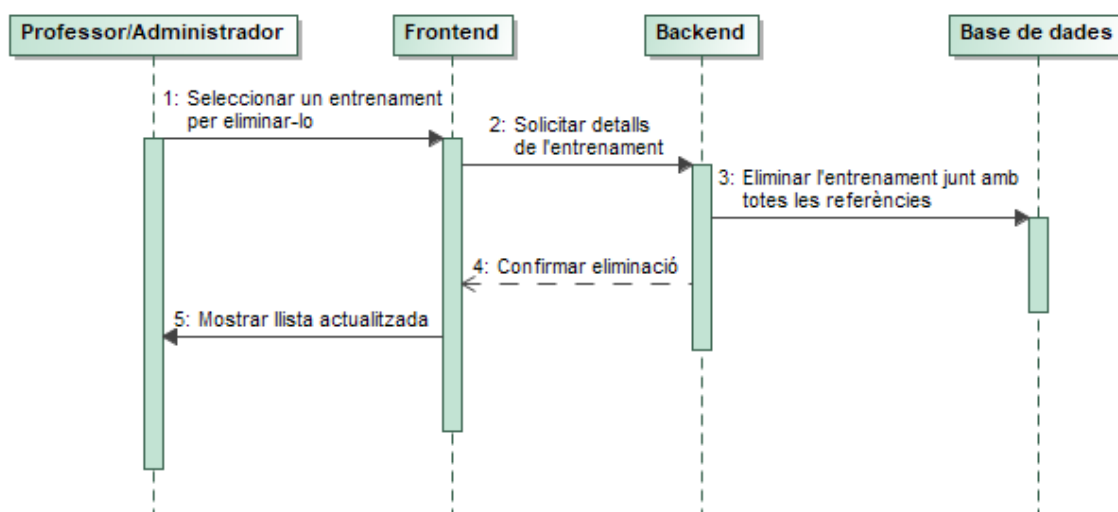


Figura 21. Diagrama d'activitats RF6.4.

**RF6.5. Eliminar un entrenament.**

- Resum de la funcionalitat: L'aplicació permet als entrenadors i administradors eliminar entrenaments del sistema.
- Paràmetres d'entrada: Identificador de l'entrenament.
- Paràmetres de sortida: Llista d'entrenaments actualitzada.
- Actors: Professor, Administrador.
- Precondició: Haver iniciat sessió (RF1) i haver accedit a la secció d'entrenaments (RF5).
- Postcondició: L'entrenament s'elimina junt amb tots els registres que el contenen de les taules “activitats”, “ordre” i “userentr” que fan referència al mateix.
- Procés normal principal:
  1. L'usuari fa clic al botó “eliminar” de la fila de l'entrenament que es vol eliminar.
  2. El sistema elimina l'entrenament eliminant també els registres dels usuaris que el tenen assignat, els de les activitats que les componen i els de l'ordre de les cançons.
  3. S'actualitza la taula d'entrenaments i es mostren els entrenaments restants.
- Alternatives del procés i excepcions:
  - Si l'entrenament no es pot eliminar, es mostra un missatge d'error.



**Figura 22.** Diagrama de seqüències RF6.5.

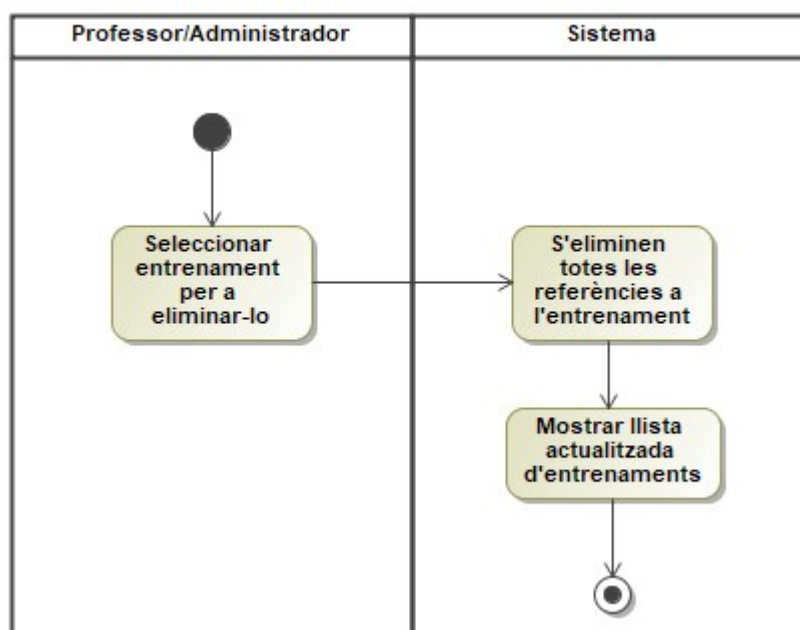


Figura 23. Diagrama d'activitats RF6.5.

### 3.2 Requisits no funcionals

- **RNF1. Usabilitat:** L'aplicació ha de ser intuïtiva i fàcil d'utilitzar. Els entrenadors han de poder crear, editar, i assignar entrenaments sense necessitat d'una formació extensa.
- **RNF2. Eficiència de processament d'àudio:** L'aplicació ha de processar la descàrrega i l'anàlisi de música de manera eficient amb temps de resposta adequats per a una experiència fluida.
- **RNF3. Gestió correcta de l'eliminació d'entrenaments:** L'aplicació ha d'assegurar la integritat referencial fent esborrat en cascada dels registres associats a l'entrenament que no estiguin en ús, de manera que es garanteixi que l'eliminació d'un entrenament no deixa registres orfes a la base de dades.
- **RNF4. Compatibilitat amb diferents navegadors:** L'aplicació ha de ser compatible amb diferents navegadors web, garantint una experiència consistent i sense errors.
- **RNF5. Seguretat de les dades:** L'aplicació ha de garantir la seguretat de les dades personals i sensibles, utilitzant mètodes de xifrat per a l'emmagatzematge de contrasenyes i altres dades crítiques.
- **RNF6. Responsivitat:** L'aplicació ha de ser *responsive*<sup>3</sup>, adaptant-se a diferents resolucions i dispositius, incloent-hi ordinadors, tauletes i telèfons mòbils.
- **RNF7. Compatibilitat amb diferents dispositius:** L'aplicació haurà de ser una aplicació web no nativa per a evitar dependències dels dispositius. D'aquesta manera, es facilita el manteniment i es redueix la complexitat de desenvolupament, ja que no es requereix una aplicació diferent per a cada dispositiu.(Android, IOS, Windows...).

<sup>3</sup> Responsive: Disseny web que s'adapta automàticament a diferents mides de pantalla i dispositius, oferint una experiència d'usuari òptima tant en ordinadors com en mòbils i tauletes.

### 3.3 Requisits addicionals

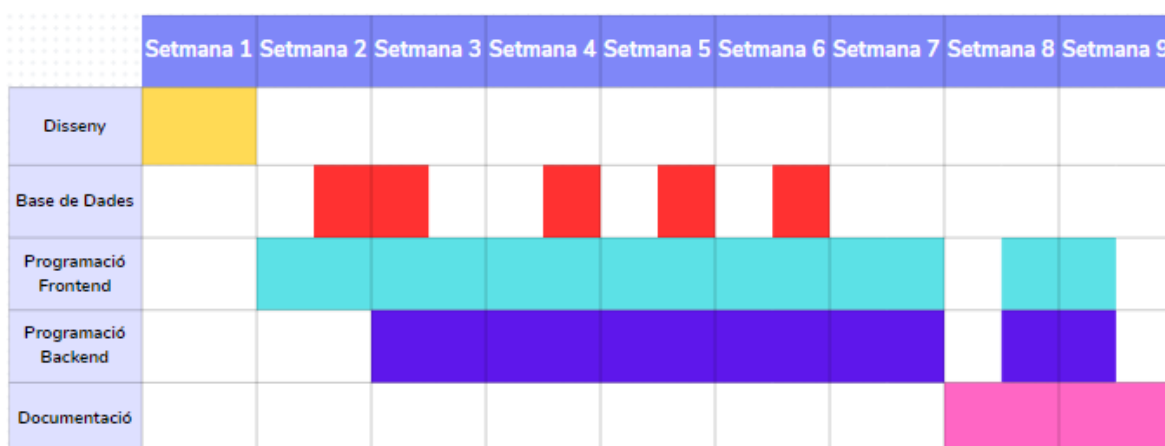
- **RA1. Integració amb diferents plataformes de streaming:** En actualitzacions futures, l'aplicació podria implementar integració amb diferents plataformes de streaming d'àudio, com Spotify, Apple Music o Tidal(entre d'altres), per tal que els usuaris puguin reproduir la música amb la seva aplicació preferida.
- **RA2. Suport per a múltiples entrenadors:** Actualment, l'aplicació pot ser utilitzada per múltiples entrenadors, però aquests tenen accés a tots els alumnes indistintament a l'hora d'assignar o desassignar alumnes. En futures versions, es podria afegir una nova taula de relacions que permeti associar cada entrenador amb els seus propis alumnes. Això permetria que cada entrenador pugui gestionar, assignar i desassignar entrenaments només als seus esportistes.
- **RA3. Gestió d'usuaris:** En versions futures, l'aplicació podria implementar un sistema complet de gestió d'usuaris, permetent, per exemple, que els administradors puguin crear, editar i eliminar usuaris, així com assignar rols per a no haver de crear-los directament a la base de dades.
- **RA4. Integració amb aplicacions d'esport:** En una futura actualització, l'aplicació es podria sincronitzar amb aplicacions com Strava per tal d'oferir una experiència completa amb el progrés realitzat o amb aplicacions que proposen exercicis com pot ser Nike Training Club per a fer més entretingut l'entrenament proposat.
- **RA5. Millora de l'estil visual de l'aplicació:** En una propera actualització, es podria considerar millorar l'estil visual de l'aplicació web per oferir una interfície d'usuari més moderna i atractiva, ja que aquest aspecte no es va poder desenvolupar completament a causa de limitacions de temps durant la realització del projecte.

## 4 Disseny

En aquest apartat s'explicaran les decisions de disseny que s'han implementat. S'inclourà també una taula del cost temporal del desenvolupament del projecte, informació del disseny de la GUI<sup>4</sup>, la lògica del backend i la distribució de les dades.

### 4.1 Distribució temporal

Abans de descriure el disseny de l'aplicació, es pot veure en la imatge inferior quin ha estat el cost temporal de cada una de les parts d'aquest projecte.



**Figura 24.** Distribució temporal.

La distribució temporal reflecteix l'evolució del projecte des del disseny inicial fins a la documentació final. En les primeres setmanes, es va donar prioritat al disseny de l'aplicació i al disseny web, assegurant una base sòlida per al desenvolupament posterior. El treball amb la base de dades es va realitzar en paral·lel, permetent que tant el backend com el frontend poguessin ser desenvolupats de manera eficient un cop el disseny i la lògica de dades estiguessin establerts.

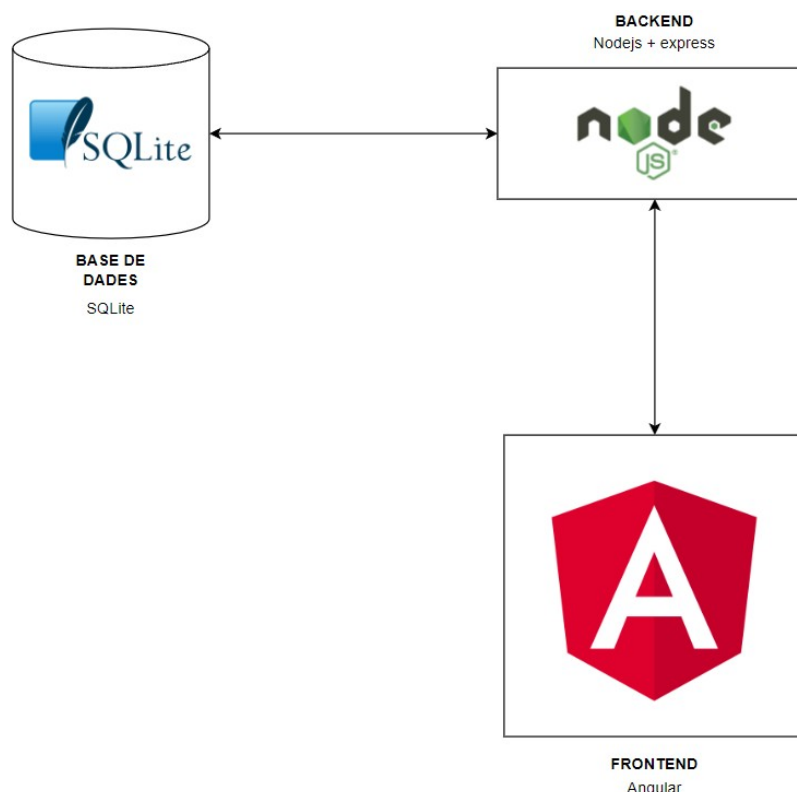
A mesura que avançava el projecte, el desenvolupament del frontend i del backend es va intensificar, amb la integració de la base de dades i la creació de les funcionalitats principals. Aquesta fase va requerir un esforç per tal de coordinar totes les capes de l'aplicació per assegurar que el sistema funcionés de manera cohesiva i sense problemes.

Finalment, les darreres setmanes es van dedicar a la documentació del projecte, assegurant que tots els aspectes del desenvolupament estiguessin adequadament registrats i explicats. Aquesta fase també va incloure la revisió i refinament de les funcionalitats desenvolupades, per garantir que l'aplicació complís amb els requisits funcionals i no funcionals definits inicialment.

<sup>4</sup> GUI (Graphical User Interface): Sistema que permet als usuaris interactuar amb un dispositiu mitjançant elements visuals com finestres, icones i menús, en lloc de línies de comandes.

## 4.2 Arquitectura de l'aplicació

A continuació es detallarà un esquema amb l'estructura bàsica de l'aplicació i l'ordre que es va seguir per a desenvolupar-la.



**Figura 25.** Esquema de l'arquitectura del projecte.

### 4.2.1 Capa de Presentació (Frontend)

La capa de presentació està desenvolupada amb Angular, un framework de front-end basat en TypeScript. Aquesta capa s'encarrega de la interacció amb l'usuari, presentant la informació de manera dinàmica i permetent l'entrada de dades. Angular facilita la creació de components reutilitzables i modulars, com formularis per a la creació d'entrenaments, llistes de cançons, i altres elements visuals de l'aplicació.

Per a desenvolupar aquesta capa es va optar per una GUI neta i senzilla que prioritza l'eficàcia del rendiment sobre l'estètica, ja que, està pensada per a ser utilitzada sense necessitar una formació extensa i ha de ser intuïtiva per a nous entrenadors i alumnes.

El Frontend està dividit en diferents components que inclouen funcions per a interactuar amb el backend i tractar les dades.

A continuació, es detallaran els components i les funcions que inclou cada un d'ells amb una petita descripció del que fan aquestes.

#### Component "seleccionar-acció":

Funció	Descripció
<b>navigateTo</b>	Navegar d'un component a l'altre
<b>logout</b>	Tancar sessió, allibera el token i navega a la pàgina d'iniciar sessió
<b>obtindreRol</b>	A partir del token de l'usuari, obté el rol del mateix

**Taula 1.** Funcions component "seleccionar-acció".

**Component “login”:**

<b>Funció</b>	<b>Descripció</b>
<b>login</b>	Comprova el nom d'usuari i contrasenya per a iniciar sessió. També mira el rol de l'usuari per tal de redigir-lo a la pàgina corresponent

**Taula 2.** Funcions component “login”.**Component “seleccionar-musica”:**

En aquest component es repeteixen les funcions “*navigateTo*”, “*logout*” i “*obtindreRol*”, motiu pel qual no s'explicaran.

<b>Funció</b>	<b>Descripció</b>
<b>ngOnInit</b>	Funció d'Angular que s'executa una vegada el component es crea i s'ha inicialitzat. Molt útil per a carregar dades o configuracions abans de mostrar la vista de la pàgina.
<b>convertirMinuts</b>	Convertir els segons que dura la cançó a minuts.
<b>getCancons</b>	Es comunica amb el Backend per a obtenir la llista de cançons.
<b>filtrarLlista</b>	Filtrar cançons pel seu nom.
<b>menuFiltres</b>	Obrir i tancar el menú de filtres.
<b>aplicarFiltres</b>	Aplicar els filtres seleccionats al menú.
<b>netejarFiltres</b>	Netejar els filtres

**Taula 3.** Funcions component “seleccionar-musica”.**Component “pujar-musica”:**

En aquest component es repeteixen les funcions “*navigateTo*”, “*logout*”, “*obtindreRol*” i “*convertirMinuts*”, motiu pel qual no s'explicaran.

<b>Funció</b>	<b>Descripció</b>
<b>buscarCanco</b>	Es comunica amb el Backend enviant-li l'URL de Youtube per a descarregar-lo i analitzar-lo. Rep com a resposta el títol, la duració i els BPM de la cançó.
<b>pujarCanco</b>	Es comunica amb el Backend per a inserir a la base de dades la cançó analitzada.
<b>calcularIntensitat</b>	Calcula la intensitat de la cançó depenent dels BPM d'aquesta.

**Taula 4.** Funcions component “pujar-musica”.**Component “seleccionar-entrenament”:**

En aquest component es repeteixen les funcions “*navigateTo*”, “*logout*”, “*obtindreRol*” i “*ngOnInit*”, motiu pel qual no s'explicaran.

<b>Funció</b>	<b>Descripció</b>
<b>obtindreEntrenaments</b>	Es comunica amb el Backend per a carregar la llista d'entrenaments. Depenent del rol de l'usuari mostra la llista completa o únicament els entrenaments assignats a l'alumne.

<b>obtindreIdUser</b>	Obtenir l'id del usuari per a carregar els entrenaments que té assignats.
<b>mostrarEntrenament</b>	Es comunica amb el Backend per a obtenir els detalls de l'entrenament i mostrar-los a l'usuari.
<b>eliminarEntrenament</b>	Es comunica amb el Backend per a eliminar l'entrenament.
<b>reproduirEntrenament</b>	Es comunica amb el Backend per a obtenir l'ordre de les cançons de l'entrenament.
<b>iniciarAPIYouTube</b>	Carregar el reproductor de Youtube per a reproduir els vídeos.
<b>carregarVideo</b>	Carregar el vídeo per a reproduir-lo a partir del seu id.
<b>canviEstat</b>	Detecta quan acaba un vídeo i carrega el següent.
<b>videoID</b>	Obtenir l'id del vídeo per a poder carregar-lo a l'API.
<b>mostrarAlumnes</b>	Es comunica amb el Backend per a mostrar la llista dels alumnes que no tenen l'entrenament assignat.
<b>mostrarAssignats</b>	Es comunica amb el Backend per a mostrar la llista dels alumnes que tenen l'entrenament assignat.
<b>assignarEntrenament</b>	Es comunica amb el Backend per a afegir l'id de l'usuari i de l'entrenament a la taula "userentr"
<b>desassignarEntrenament</b>	Es comunica amb el Backend per a eliminar l'id de l'usuari i de l'entrenament de la taula "userentr"

**Taula 5.** Funcions component "seleccionar-entrenament".

#### Component "crear-entrenament":

En aquest component es repeteixen les funcions "*navigateTo*", "*logout*", "*obtindreRol*" i "*convertirMinuts*", motiu pel qual no s'explicaran.

<b>Funció</b>	<b>Descripció</b>
<b>afegirActivitat</b>	Afegir una nova activitat al formulari de creació.
<b>eliminarActivitat</b>	Eliminar l'activitat seleccionada del formulari.
<b>actualitzarDuracio</b>	Actualitzar la duració de l'entrenament quan es canvia el temps d'una activitat o s'elimina una activitat.
<b>validarCamps</b>	Comprovar que tots els camps s'han omplert correctament abans d'omplir les activitats amb cançons.
<b>crearEntrenament</b>	Es comunica amb el Backend per a inserir l'entrenament i l'ordre de les cançons a la base de dades.
<b>plenarCançons</b>	Buscar cançons de la intensitat seleccionada per a l'activitat i afegeix cançons a l'activitat fins a arribar, com a mínim, a la durada seleccionada.
<b>editarCanco</b>	Es comunica amb el Backend per a mostrar la llista de cançons amb la mateixa intensitat que la cançó seleccionada i deixa la possibilitat de canviar-la per una altra.

<b>afegirCanco</b>	Es comunica amb el Backend per a mostrar la llista de cançons amb la mateixa intensitat de l'activitat i així poder afegir una nova cançó a l'entrenament.
<b>eliminarCanco</b>	Eliminar la cançó de l'activitat i actualitzar la durada d'aquesta.
<b>tancarSelectorCancons</b>	Tanca la llista que es mostra a l'editar o afegir cançons a l'activitat.
<b>seleccionarNovaCanco</b>	Edita la cançó seleccionada o afegeix una nova cançó a la llista.
<b>confirmarCancons</b>	Confirmar les cançons i crida a la funció <b>crearEntrenament</b> per a afegir l'entrenament a la base de dades.

**Taula 6.** Funcions component “crear-entrenament”.

#### 4.2.2 Capa de Lògica de l'Aplicació (Backend)

La capa de lògica d'aplicació està implementada amb Node.js i Express. Aquesta capa gestiona les peticions del frontend, processa les dades, aplica la lògica i interactua amb la base de dades. És responsable de funcionalitats com la descàrrega i anàlisi de música, l'autenticació d'usuaris, la creació i gestió d'entrenaments, entre d'altres.

El backend està estructurat per gestionar les rutes d'API que permeten al frontend interactuar amb la base de dades. A través d'aquestes rutes, es poden efectuar operacions CRUD<sup>s</sup>.

A continuació es presenta un resum de les funcions principals incloses en el backend classificades pel tipus d'operació que realitzen:

##### Operacions *delete*:

<b>Funció</b>	<b>Descripció</b>
<b>/api/entrenament/:id</b>	Elimina un entrenament del sistema, incloent-hi les seves activitats, l'ordre de les cançons i les relacions dels usuaris que el tenen assignat a la base de dades.
<b>/api/desassignar-entrenament/:iduser/:identre</b>	Elimina la relació entre un alumne i un entrenament, desassignant-lo de l'alumne especificat.

**Taula 7.** Funcions *delete* del Backend.

##### Operacions *post*:

<b>Funció</b>	<b>Descripció</b>
<b>/api/entrenament</b>	Crea un nou entrenament al sistema amb les activitats associades.
<b>/api/ordre</b>	Afegeix un ordre específic de les cançons a una activitat dins d'un entrenament.
<b>/api/buscar-canco</b>	Permet la descàrrega d'una cançó des de YouTube, així com l'anàlisi dels seus BPM i altres metadades per a la seva classificació.
<b>/api/pujar-canco</b>	Insereix una nova cançó a la base de dades amb les metadades obtingudes, com el títol, la duració i els BPM.
<b>/api/login</b>	Autentica un usuari amb el seu nom d'usuari i

<sup>s</sup> CRUD (Create, Read, Update, Delete): Són les operacions bàsiques per gestionar dades en una base de dades o aplicació.

	contrasenya, retornant un token de sessió si les credencials són correctes.
<b>/api/assignar-entrenament</b>	Assigna un entrenament a un alumne específic, registrant la relació a la base de dades.

Taula 8. Funcions *post* del Backend.**Operacions *get*:**

<b>Funció</b>	<b>Descripció</b>
<b>/api/cançons</b>	Retorna una llista de totes les cançons emmagatzemades a la base de dades.
<b>/api/entrenaments</b>	Retorna una llista de tots els entrenaments disponibles al sistema.
<b>/api/entrenaments-alumne</b>	Retorna una llista dels entrenaments assignats a un alumne en particular.
<b>/api/entrenament/:id</b>	Retorna els detalls específics d'un entrenament, incloent-hi les activitats i cançons associades.
<b>/api/cancoINT</b>	Retorna una llista de cançons filtrades per la intensitat especificada en la consulta.
<b>/api/reproduirEntrenament/:id/</b>	Retorna l'ordre i la llista de cançons associades a un entrenament, per poder ser reproduïdes en ordre.
<b>/api/usuaris-alumnes/:identre</b>	Retorna una llista dels alumnes que no tenen l'entrenament especificat assignat.
<b>/api/usuaris-assignats/:identre</b>	Retorna una llista dels alumnes que tenen l'entrenament especificat assignat.

Taula 9. Funcions *get* del Backend.

A més a més, el backend integra dos scripts de Python que són essencials per al funcionament de l'aplicació, ja que s'encarregaran de dues de les funcions més importants de l'aplicació, que són la descàrrega i l'anàlisi de la música, i un script Node.js que s'executa manualment des de consola per a xifrar les contrasenyes abans d'inserir-les a la base de dades.

Per a la integració dels scripts de Python amb Node.js, es criden mitjançant la funció *exec* del mòdul *child\_process*. Això permet executar els scripts externs de Python i capturar-ne els resultats per a usar-los dins de l'aplicació.

D'altra banda, per a xifrar les contrasenyes és necessari executar manualment el script amb la contrasenya a xifrar per a obtenir el hash de la contrasenya xifrada.

Així mateix, es procedirà a explicar la funció de cada script i les llibreries utilitzades dins d'aquest junt amb el codi.

**Script *descarregarAudio.py***

- **Funció:** A partir de l'enllaç proporcionat per l'usuari, es descarrega l'àudio del vídeo en format MP3 i obté el títol.
- **Llibreries utilitzades:**
  - *yt\_dlp*: És una versió avançada de la llibreria *youtube-dl*. Aquesta llibreria és una eina per descarregar vídeos i àudios de YouTube. En aquest cas, *yt\_dlp* s'utilitza per extreure i descarregar l'àudio del vídeo en el millor format disponible.

- *Sys*: Utilitzada per a gestionar arguments de la línia de comandes. Es fa servir per a rebre l'URL<sup>6</sup> de Youtube i el path<sup>7</sup> on es descarregarà l'àudio com a paràmetres de l'script.

- **Codi:**

```
import sys
from yt_dlp import YoutubeDL

def descarregarAudio(youtube_url, output_path):
    ydl_opts = {
        'format': 'bestaudio/best',
        'outtmpl': output_path,
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192',
        }],
    }

    with YoutubeDL(ydl_opts) as ydl:
        info_dict = ydl.extract_info(youtube_url, download=True)
        title = info_dict.get('title', None)
        return title

if __name__ == "__main__":

    youtube_url = sys.argv[1]
    output_path = sys.argv[2]

    try:
        title = descarregarAudio(youtube_url, output_path)
        print(f"{title}")
    except Exception as e:
        print(f"Error al descarregar l'audio: {e}")
        sys.exit(1)
```

**Codi 1.** Script `descarregarAudio.py`.

### Script `analitzarAudio.py`

- **Funció:** Aquest script rep com a entrada el path del fitxer d'àudio descarregat anteriorment i així poder analitzar-lo per a obtenir els BPM i la duració. Aquests paràmetres són essencials per a poder classificar l'àudio correctament.
- **Llibreries utilitzades:**
  - *Librosa*: Llibreria utilitzada per a l'anàlisi d'àudio en Python. Es fa servir per a carregar l'àudio, calcular la seva durada i determinar els BPM.
  - *Json*: Llibreria que permet la serialització de les dades en format JSON, facilitant la comunicació entre l'script i el backend.

---

<sup>6</sup> URL (Uniform Resource Location): Adreça que identifica i localitza una pàgina web o recurs en Internet.

<sup>7</sup> Path: Ruta que especifica la ubicació d'un fitxer o directori dins d'un sistema de fitxers.

- Sys: De manera similar a l'anterior, s'utilitza per a rebre el path del fitxer d'àudio que es desitja analitzar.

- **Codi:**

```
import sys
import librosa
import json

def analyze_audio(audio_path):
    try:
        y, sr = librosa.load(audio_path)
        duracio = librosa.get_duration(y=y, sr=sr)
        bpm, _ = librosa.beat.beat_track(y=y, sr=sr)
        return float(duracio), float(bpm)
    except Exception as e:
        print(f"Error analitzant l\'audio: {e}", file=sys.stderr)
        return None, None

if __name__ == "__main__":
    audio_path = sys.argv[1]
    duracio, bpm = analyze_audio(audio_path)
    if duracio is not None and bpm is not None:
        print(json.dumps({"duracio": duracio, "bpm": bpm}))
    else:
        print(json.dumps({"error": "Error analitzant l\'audio"}))
```

### Codi 2. Script analitzarAudio.py.

#### Script xifrarPass.js

- **Funció:** Aquest script genera un hash segur a partir d'una contrasenya proporcionada com a paràmetre. El hash resultant és adequat per ser emmagatzemat de manera segura a la base de dades.
- **Llibreries utilitzades:**
  - *bcrypt*: Llibreria que implementa l'algoritme de hashing bcrypt, que es basa en l'algoritme Blowfish per generar un hash segur de la contrasenya.
- **Codi:**

```
const bcrypt = require('bcrypt');

const contrasenyaText = process.argv[2];

if (!contrasenyaText) {
    console.error('Has de proporcionar la contrasenya com a paràmetre.');
```

```
    process.exit(1);
}

bcrypt.hash(contrasenyaText, 10, (err, contrasenyaXifrada) => {
    if (err) {
        console.error('Error en xifrar la contrasenya:', err);
        process.exit(1);
    } else {
```

```

        console.log('La contrasenya xifrada és:',
contrasenyaXifrada);
        process.exit(0);
    }
});

```

### Codi 3. Script xifrarPass.js

#### 4.2.3 Capa de Dades (Base de Dades)

La capa de persistència de dades està dissenyada utilitzant SQLite, una base de dades lleugera i fàcil d'integrar amb aplicacions petites i mitjanes. Aquesta capa s'encarrega d'emmagatzemar de manera persistent tota la informació necessària per al funcionament de l'aplicació, incloent-hi usuaris, entrenaments, activitats, cançons i les relacions entre aquests elements.

Dues coses que s'han tingut en compte a l'hora de crear la base de dades:

- **Relacions entre taules:** La base de dades està dissenyada amb claus foranes per mantenir la integritat referencial de les dades. Per exemple, la taula *activitats* té una clau forana *identrenament* que apunta a la taula *entrenaments*, garantint així que cada activitat estigui associada a un entrenament vàlid.
- **Integritat referencial:** La integritat referencial es manté mitjançant les claus foranes i restriccions a nivell de base de dades per evitar la inserció o eliminació de dades que podrien comprometre la coherència de les dades. Per exemple, no es pot eliminar un entrenament si té activitats associades.

A continuació, es presenten les taules principals de la base de dades i les seves funcions.

Taula	Descripció
<b>activitats</b>	Conté les activitats que componen cada entrenament, incloent-hi la duració, intensitat i l'entrenament al qual pertanyen.
<b>cançons</b>	Emmagatzema les cançons disponibles per a la creació d'entrenaments.
<b>entrenaments</b>	Conté la informació sobre els entrenaments creats.
<b>ordre</b>	Defineix l'ordre de les cançons, relacionant-les amb l'activitat i l'entrenament que pertanyen.
<b>userentr</b>	Guarda les relacions entre els usuaris i els entrenaments que tenen assignats.
<b>usuaris</b>	Emmagatzema les dades dels usuaris de l'aplicació.

**Taula 10.** Taules de la Base de Dades.

Tot seguit, es procedirà a explicar en detall les taules, definint els camps de cada una d'aquestes, les claus principals i les claus foranes de cada una.

#### Detalls de les taules:

- **activitats**
  - **Camps:** idactivitat, identrenament, temps, intensitat
  - **Claus primàries:** idactivitat, identrenament
  - **Claus foranes:** *identrenament* que referència al camp *id* de la taula *entrenaments*.

- **cançons**
  - **Camps:** id, títol, duració, bpm, intensitat, artista, link
  - **Clau primària:** id
- **entrenaments**
  - **Camps:** id, nom, duració
  - **Clau primària:** id
- **ordre**
  - **Camps:** identre, idact, idcanco, ordre
  - **Clau primària:** identre, idact, idcanco, ordre
  - **Claus foranes:** *identre* que referència al camp *id* de la taula *entrenaments*, *idact* que referència al camp *id* de la taula *activitats*, *idcanco* que referència al camp *id* de la taula *cançons*.
- **userentr**
  - **Camps:** iduser, identre
  - **Clau primària:** iduser, identre
  - **Claus foranes:** *identre* que referència al camp *id* de la taula *entrenaments*, *iduser* que referència al camp *id* de la taula *usuaris*.
- **usuaris**
  - **Camps:** id, nom, cognoms, rol, user, pass.
  - **Clau primària:** id

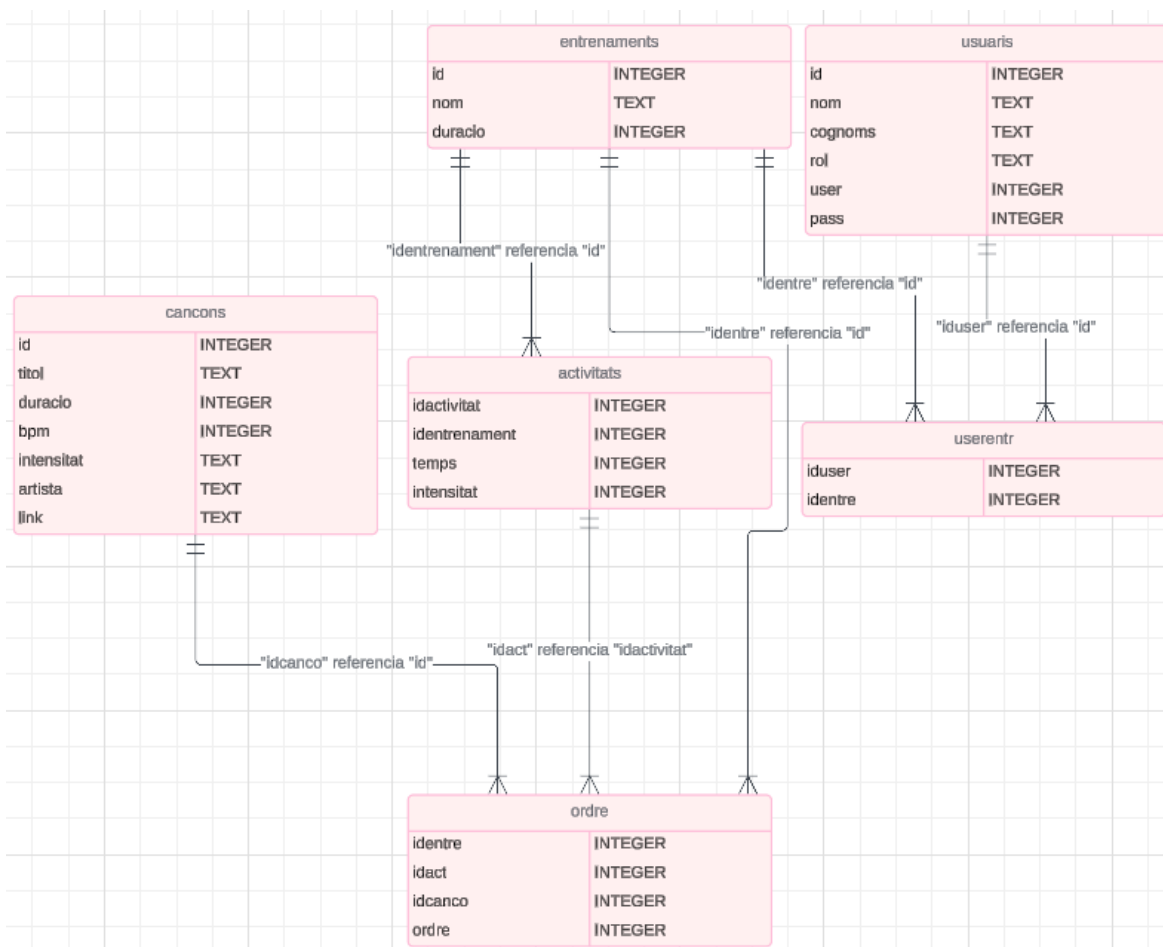


Figura 26. Estructura de la base de dades.

## 5 Implementació

En aquest apartat es descriuen el client, el servidor i la interacció entre ells, tenint en compte la base de dades i les comunicacions amb els scripts de Python per a l'anàlisi i la descàrrega de música.

El procés s'inicia quan un usuari interactua amb l'aplicació Angular, la qual s'encarrega de mostrar la interfície gràfica i gestionar les accions de l'usuari. Quan un usuari sol·licita una acció específica, com ara la descàrrega d'una cançó o la creació d'un entrenament, el client envia una petició HTTP al servidor backend, que està implementat amb Node.js i Express.

El servidor, que roman obert per a processar aquestes peticions, rep la sol·licitud i la gestiona adequadament. Per exemple, en el cas de la descàrrega d'una cançó, el servidor invoca un script de Python que utilitza la llibreria *yt\_dlp* per a descarregar l'àudio d'un vídeo de YouTube. Un cop descarregat, un altre script de Python, utilitzant la llibreria *librosa*, analitza l'arxiu d'àudio per a extreure'n la durada i el BPM. Aquestes dades s'envien de tornada al servidor, que les processa i les emmagatzema a la base de dades SQLite.

Pel que fa a la gestió dels entrenaments, el servidor s'encarrega de rebre les peticions per a crear, editar, o eliminar entrenaments. Aquestes operacions impliquen la interacció amb la base de dades per a assegurar que els entrenaments es creen amb les activitats i cançons adequades. El servidor també gestiona la reproducció de les cançons, enviant al client la informació necessària per a reproduir les pistes en l'ordre correcte.

A més, l'autenticació dels usuaris es gestiona a través de tokens JWT<sup>8</sup>, assegurant que només els usuaris autenticats poden accedir a les funcionalitats de l'aplicació segons el seu rol (alumne, professor o administrador).

### 5.1 Detalls de la tecnologia implementada

#### 5.1.1 Frontend: Angular, HTML, CSS i TypeScript

El frontend de l'aplicació s'ha desenvolupat utilitzant diverses tecnologies que permeten la creació d'una interfície d'usuari interactiva i fàcil d'utilitzar. A continuació, es descriuran les tecnologies utilitzades:

- **Angular:** És un framework de desenvolupament d'aplicacions web basat en TypeScript, desenvolupat i mantingut per Google. Angular és una plataforma i un framework per a crear aplicacions d'una sola pàgina en el costat del client usant HTML i TypeScript. Angular facilita també la implementació de serveis per gestionar la lògica de l'aplicació i la comunicació amb el backend a través de HTTP.
- **HTML (HyperText Markup Language):** És el llenguatge de marcat estàndard per crear pàgines web. HTML descriu l'estructura d'una pàgina web a través d'una sèrie d'elements, els quals indiquen al navegador com ha de mostrar el contingut. Aquests elements d'HTML etiqueten les diferents parts del contingut.
- **CSS (Cascading Style Sheet):** És el llenguatge que descriu com s'han de mostrar els elements HTML en pantalla, paper o en altres mitjans. CSS estalvia molta feina, ja que pot controlar el disseny de múltiples pàgines web de manera simultània.

---

<sup>8</sup> Token JWT: Formats compactes i segurs per a l'autenticació, que contenen informació codificada sobre l'usuari i s'utilitzen per verificar identitats de manera segura en aplicacions web.

- **TypeScript:** És un llenguatge de programació de codi obert desenvolupat per Microsoft. És un superconjunt de JavaScript que afegeix la capacitat de treballar amb tipus estàtics i suport per a la programació orientada a objectes amb classes. TypeScript permet detectar errors en el codi de manera anticipada durant el desenvolupament i proporciona eines avançades per a l'autocompletat i la navegació del codi.

### 5.1.2 Backend: Node.js, Express o HTTP

El backend de l'aplicació gestiona la lògica del servidor, l'accés a la base de dades i la comunicació amb el frontend. Per a aquest propòsit, s'han utilitzat les següents tecnologies:

- **Node.js:** És un entorn d'execució de JavaScript orientat a esdeveniments asíncrons, dissenyat per construir aplicacions de xarxa escalables. Opera en un sol fil d'execució utilitzant un bucle d'esdeveniments, cosa que permet gestionar múltiples connexions de manera eficient sense bloquejar l'execució. Això fa que sigui ideal per al desenvolupament de servidors web.
- **Express:** Express és un framework minimalista i flexible per a Node.js que simplifica la creació d'aplicacions web i APIs. Proporciona una àmplia gamma de funcionalitats per gestionar rutes, middleware, sessions, i altres aspectes essencials d'una aplicació web.
- **HTTP (HyperText Transfer Protocol):** És un protocol subjacent utilitzat per la World Wide Web. L'HTTP defineix com es formaten i transmeten els missatges i quines accions han de prendre els servidors web i navegadors en resposta a diversos comandos. Les peticions HTTP permeten que el frontend sol·liciti dades al backend (mitjançant operacions GET, POST, DELETE, etc.), i que el backend retorni les dades sol·licitades o processi operacions específiques, com ara la creació o modificació de registres en la base de dades.

### 5.1.3 Base de Dades: SQL, SQLite

La persistència de dades s'ha implementat mitjançant l'ús de SQLite, una base de dades relacional que emmagatzema totes les dades en un únic fitxer.

- **SQL (Structured Query Language):** És un llenguatge de programació per a emmagatzemar i processar informació en una base de dades relacional. Una base de dades relacional emmagatzema informació en forma de taula, amb files i columnes que representen diferents atributs de dades i les diverses relacions entre els valors de dades. Pot usar les instruccions SQL per a emmagatzemar, actualitzar, eliminar, buscar i recuperar informació de la base de dades.
- **SQLite:** És una biblioteca de programari que implementa un sistema de gestió de bases de dades relacional lleugera. És autònoma, sense servidor, zero-configuració i transaccional. SQLite està dissenyada per ser eficient i fàcil d'incorporar en aplicacions, amb fitxers de bases de dades que són simplement fitxers ordinaris en el sistema. Gràcies a la seva simplicitat i robustesa, és àmpliament utilitzada en aplicacions mòbils, dispositius integrats i navegadors web. Ja que guarda totes les dades en un sol fitxer, facilita la gestió i distribució de la base de dades.

## 5.2 Algoritmes específics implementats

En aquest apartat es descriuen els algoritmes més rellevants que s'han implementat per a desenvolupar les funcionalitats clau de l'aplicació i s'inclourà també el codi. Cada algoritme està dissenyat per a abordar un aspecte específic del projecte, optimitzant el rendiment i assegurant la correcta execució de les tasques requerides per l'aplicació.

### 5.2.1 Algoritme de selecció de música segons els BPM

Aquest algoritme té com a objectiu seleccionar les cançons més adequades per a cada activitat d'un entrenament, basant-se en el ritme desitjat (BPM). Quan un entrenador defineix la intensitat d'una activitat, l'algoritme cerca a la base de dades les cançons que coincideixen amb els BPM requerits. Si no hi ha suficients cançons per a cobrir la durada de l'activitat, l'algoritme repeteix cançons de manera que el ritme es mantingui constant al llarg de l'exercici. Això permet garantir que la música seleccionada sigui congruent amb l'objectiu de l'entrenament.

- **Codi Frontend:**

Codi per a omplir la llista amb cançons de la intensitat seleccionada.

```
plenarCancons() {
  if (!this.validarCamps()) {
    return;
  }

  const seleccions = this.activitats.map((activitat: any) => {
    console.log(`Buscant cançons per a la intensitat: ${
      {activitat.intensitat}`);
    return this.http.get<any[]>(`/api/cancoINT?intensitat=${
      {activitat.intensitat}`).toPromise();
  });
});

Promise.all(seleccions).then((resultats: any[]) => {
  console.log('Resultats de les cerques de cançons:',
resultats);
  let duracioTotalEntrenament = 0;
  this.canconsSeleccionades = resultats.map((cancons, index) =>
{
  let canconsSeleccionades: any[] = [];
  let tempsTotal = 0;
  let indexCanco = 0;

  while (tempsTotal < this.activitats[index].temps * 60) {
    if (indexCanco >= cancons.length) {
      indexCanco = 0;
    }

    const canco = cancons[indexCanco];
    canconsSeleccionades.push(canco);
    tempsTotal += canco.duracio;
    indexCanco++;
  }

  //Actualitzar la duració de l'activitat
  this.activitats[index].temps = Math.round(tempsTotal / 60);
}
```

```

        duracioTotalEntrenament += this.activitats[index].temps;
        console.log(`Cançons seleccionades per a l'activitat $
{index}:`, canconsSeleccionades);
        return {
            activitat: this.activitats[index],
            cancons: canconsSeleccionades
        };
    });

    this.duracioEntrenament = duracioTotalEntrenament;
    this.revisat = true;
}).catch(error => {
    console.error('Error cercant cançons:', error);
});
}

```

**Codi 4.** Funció `plenarCancons()` del fitxer `crear-entrenament.component.ts`.

Codi per a validar que tots els camps necessaris per a crear l'entrenament estan omplerts correctament.

```

validarCamps(): boolean {
    if(!this.nom.trim()) {
        this.missatgeError = 'És obligatori indicar un nom per a
l\'entrenament';
        return false;
    }

    for (const activitat of this.activitats) {
        if(activitat.temps <= 0) {
            this.missatgeError = 'La duració de les activitats ha de ser
superior a 0';
            return false;
        }
        if(!activitat.intensitat) {
            this.missatgeError = 'És obligatori indicar una intensitat
per a les activitats';
            return false;
        }
    }
    this.missatgeError = '';
    return true;
}

```

**Codi 5.** Funció `validarCamps()` del fitxer `crear-entrenament.component.ts`.

- **Codi backend:**

Codi per obtenir les cançons de la intensitat seleccionada del fitxer server.js.

```
app.get('/api/cancoINT', (req, res) => {
  const { intensitat } = req.query;
  if (!intensitat) {
    res.status(400).json({ error: 'Intensitat no
proporcionada' });
    return;
  }

  const sqlObtenirCancons = 'SELECT * FROM cancons WHERE
intensitat = ?';
  db.all(sqlObtenirCancons, [intensitat], (err, rows) => {
    if (err) {
      res.status(400).json({ error: `Error en la consulta de
cançons: ${err.message}` });
      return;
    }
    res.json(rows);
  });
});
```

**Codi 6.** Codi per obtenir les cançons de la intensitat seleccionada del fitxer server.js.

### 5.2.2 Algoritme de descàrrega i anàlisi d'àudio

Aquest algoritme s'encarrega de descarregar cançons des de YouTube i analitzar-les per obtenir-ne informació crucial com els BPM i la durada. Mitjançant l'script de Python *descarregarAudio.py*, es descarrega l'àudio del vídeo seleccionat amb la llibreria *yt-dlp*. A continuació, l'script *analitzarAudio.py* utilitza la llibreria *librosa* per a processar l'arxiu d'àudio, extraient els BPM i la durada de la pista, i a continuació, elimina l'àudio per a no ocupar espai en el servidor. La informació obtinguda s'emmagatzema posteriorment a la base de dades.

- **Codi Frontend:**

Codi necessari per a enviar l'URL de Youtube al backend.

```
buscarCanco() {
  this.isLoading = true;
  this.mostrarFormulari = false;
  this.http.post({message: string, path: string, bpm: number,
duracio: number, title: string})('/api/buscar-canco', { url:
this.youtubeLink }).subscribe({
  next: (response) => {
    this.titol = response.title;
    this.duracio = response.duracio;
    this.bpm = response.bpm;
    this.isLoading = false;
    this.mostrarFormulari = true;
  }, error: (error) => {
    console.error('Error buscant video:', error);
    this.missatgeError = 'Error al descarregar o analitzar
l\'audio. Verifica el link i torna a intentar-ho.';
    this.isLoading = false;
  });
}
```

```

    }
  });
}

```

**Codi 7.** Funció buscarCanco() del fitxer pujar-musica.component.ts.

- **Codi Backend:**

Codi necessari per a executar els dos scripts i així poder descarregar un àudio a partir d'un URL de Youtube per així poder analitzar-lo per tal d'obtenir el títol, els BPM i la duració. Un cop analitzat l'àudio, s'elimina l'àudio del servidor.

```

app.post('/api/buscar-canco', (req, res) => {
  const { url } = req.body;
  const outputFilename = `audio`;
  const outputPath = path.join('tempaudio', outputFilename);

  console.log('Descarregant àudio de:', url, 'a:', outputPath);
  const descarregarAudio = `python descarregarAudio.py ${url} ${outputPath}`;

  exec(descarregarAudio, (error, stdout, stderr) => {
    if (error) {
      console.error(`Error en descarregar l'àudio: ${error}`);
      return res.status(500).json({ error: `Error en descarregar l'àudio` });
    }

    console.log('Àudio descarregat correctament, iniciant anàlisi...');
    const analitzarAudio = `python analitzarAudio.py ${outputPath}.mp3`;
    const titolVideo = stdout.split("\n").slice(-2, -1)[0];
    exec(analitzarAudio, (error, stdout, stderr) => {
      if (error) {
        console.error(`Error en analitzar l'àudio: ${error}`);
        return res.status(500).json({ error: `Error en analitzar l'àudio` });
      }

      const resultatAnàlisi = JSON.parse(stdout);
      if (resultatAnàlisi.error) {
        console.error(`Error en l'anàlisi de l'àudio: ${resultatAnàlisi.error}`);
        return res.status(500).json({ error: resultatAnàlisi.error });
      }

      const { bpm, duracio } = resultatAnàlisi;
      console.log('Resultats de l'anàlisi:', resultatAnàlisi);

      fs.unlink(`${outputPath}.mp3`, (unlinkError) => {
        if (unlinkError) {

```

```

                                console.error(`Error en eliminar el fitxer
d'àudio: ${unlinkError}`);
                                } else {
                                        console.log('Fitxer d'àudio eliminat
correctament');
                                }
                                });
                                res.json({
                                        message: 'Àudio descarregat i analitzat',
                                        path: outputPath,
                                        bpm,
                                        duracio,
                                        title: titolVideo,
                                });
                                });
                                });
                                });

```

**Codi 8.** Codi per a descarregar i analitzar l'àudio de l'URL proporcionat del fitxer `server.js`.

### 5.2.3 Algoritme d'autenticació d'usuaris i gestió de rols

Aquest algoritme gestiona l'accés a l'aplicació i la gestió de sessions dels usuaris autenticats, basant-se en els rols definits (alumne, entrenador, administrador). Durant el procés d'autenticació, es genera un token JWT que s'emmagatzema i s'utilitza per verificar l'autenticació en cada petició posterior. L'algoritme determina les funcionalitats a les quals cada usuari pot accedir segons el seu rol, com ara la creació i eliminació d'entrenaments o l'assignació de sessions d'entrenament a altres usuaris.

- **Codi Frontend:**

Codi per a enviar les credencials de l'usuari al backend i poder redirigir-lo a la pàgina adient en funció del seu rol.

```

login() {
    this.http.post<any>('/api/login', { user: this.user, pass:
this.pass }).subscribe({next: res => {
        localStorage.setItem('token', res.token);
        const payload = JSON.parse(atob(res.token.split('.')[1]));
        if (payload.rol === 'alumne') {
            this.router.navigate(['/entrenaments-assignats']);
        } else {
            this.router.navigate(['/accio']);
        }
    }},
    error: error => {
        console.error('Error en l'inici de sessió:', error);
        alert('Usuari o contrasenya incorrectes.');
```

**Codi 9.** Funció `login()` del fitxer `login.component.ts`.

- **Codi Backend:**

Codi per a comprovar les credencials de l'usuari i donar-li un token per a poder obtenir informació sobre el rol de l'usuari.

```
app.post('/api/login', (req, res) => {
  const { user, pass } = req.body;
  const sqlConsultaUser = 'SELECT * FROM usuaris WHERE user = ?';
  db.get(sqlConsultaUser, [user], (err, row) => {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    if (row) {
      bcrypt.compare(pass, row.pass, (err, isMatch) => {
        if (err) {
          res.status(500).json({ error: 'Error al
verificar la contrasenya' });
          return;
        }
        if (isMatch) {
          const token = jwt.sign({ id: row.id, rol:
row.rol }, SECRET_KEY);
          res.json({ token, rol: row.rol });
        } else {
          res.status(401).json({ error: 'Contrasenya
incorrecta.' });
        }
      });
    } else {
      res.status(401).json({ error: 'Usuari o contrasenya
incorrectes.' });
    }
  });
});
```

}); **Codi 10.** Codi per a comprovar les credencials del fitxer server.js.

### 5.2.4 Algoritme de reproducció contínua de música per a un entrenament

Aquest algoritme controla la reproducció contínua de música durant un entrenament. Quan un usuari inicia un entrenament, l'algoritme s'encarrega de reproduir les cançons seleccionades en l'ordre especificat, sense interrupcions entre elles. Això es fa carregant cada cançó successivament i iniciant-ne la reproducció en quant acaba l'anterior, assegurant una experiència d'entrenament fluida. Aquest algoritme s'ha implementat amb l'API oficial de Youtube per a incloure un reproductor en pàgines web, cosa la qual el fa fiable enfront a possibles canvis que pugui patir l'aplicació de Youtube.

- **Codi Frontend:**

Codi per a recuperar les cançons de l'entrenament en ordre i carregar la primera.

```
reproduirEntrenament(entrenament: any) {
  this.selectedEntrenament = entrenament;
  const id = entrenament.id;
  this.http.get<any[]>(`/api/reproduirEntrenament/${
{id}`).subscribe({next: data => {
  this.reproductorDetails = data;
```

```

        this.currentSongIndex = 0;
        this.mostrarReproductor = true;
        this.carregarVideo();
    },
    error: error => {
        console.error('Error obtenint detalls de l\'entrenament:',
error);
    }
});
}

```

**Codi 11.** Funció reproduirEntrenament() del fitxer seleccionar-entrenament.component.ts.

Declaració de les interfícies i classes de l'API de YouTube per permetre una integració més senzilla amb TypeScript. Per a això, serà necessari crear un nou fitxer anomenat *youtube.d.ts*.

```

declare namespace YT {
    class Player {
        constructor(elementId: string, options: PlayerOptions);
        loadVideoById(videoId: string): void;
        playVideo(): void;
        pauseVideo(): void;
    }

    interface PlayerOptions {
        height?: string;
        width?: string;
        videoId?: string;
        events?: {
            onReady?: (event: PlayerEvent) => void;
            onStateChange?: (event: OnStateChangeEvent) => void;
        };
        playerVars?: PlayerVars;
    }

    interface PlayerVars {
        autoplay?: number;
        controls?: number;
        modestbranding?: number;
        showinfo?: number;
        rel?: number;
    }

    interface PlayerEvent {
        target: Player;
    }

    interface OnStateChangeEvent extends PlayerEvent {
        data: number;
    }

    const PlayerState: {

```

```

    ENDED: number;
    PLAYING: number;
    PAUSED: number;
    BUFFERING: number;
    CUED: number;
  };
}

```

### Codi 12. Fitxer youtube.d.ts.

Codi per a carregar l'API de YouTube i inicialitzar el reproductor quan l'API està llesta.

```

inciarAPIYouTube() {
  const tag = document.createElement('script');
  tag.src = 'https://www.youtube.com/iframe_api';
  const firstScriptTag = document.getElementsByTagName('script')
[0];
  if (firstScriptTag) {
    firstScriptTag.parentNode?.insertBefore(tag, firstScriptTag);
  } else {
    console.error('Error al inserir el script de YouTube API');
    return;
  }

  window.onYouTubeIframeAPIReady = () => {
    this.player = new window.YT.Player('player', {
      events: {
        'onStateChange': this.canviEstat.bind(this),
      },
      playerVars: {
        autoplay: 1,
        controls: 1,
      },
    });
  };
}

```

### Codi 13. Funció inciarAPIYouTube() del fitxer seleccionar-entrenament.component.ts.

Codi per a carregar i reproduir el vídeo actual seleccionat de la llista d'entrenament.

```

carregarVideo() {
  if (this.reproductorDetails.length > 0 && this.player && typeof
this.player.loadVideoById === 'function') {
    const song = this.reproductorDetails[this.currentSongIndex];
    const videoId = this.videoID(song.link);
    if (videoId) {
      this.player.loadVideoById(videoId);
    }
  } else {
    console.error('Reproductor no inicialitzat o no s\'ha trobat
el videoId');
  }
}

```

}

**Codi 14.** Funció `carregarVideo()` del fitxer `seleccionar-entrenament.component.ts`.

Codi per a detectar quan el vídeo acaba i així poder carregar el següent.

```
canviEstat(event: YT.OnStateChangeEvent) {
  if (event.data === YT.PlayerState.ENDED) {
    this.currentSongIndex++;
    if (this.currentSongIndex < this.reproductorDetails.length) {
      this.carregarVideo();
    }
  }
}
```

**Codi 15.** Funció `canviEstat()` del fitxer `seleccionar-entrenament.component.ts`.

Codi per obtenir l'id del vídeo a partir de l'URL per tal de poder carregar-lo en el reproductor.

```
videoID(url: string): string {
  const videoIdMatch = url.match(/(?:v=|^&+)\/);
  return videoIdMatch ? videoIdMatch[1] : '';
}
```

**Codi 16.** Funció `videoID()` del fitxer `seleccionar-entrenament.component.ts`.

- **Codi Backend:**

Codi per a consultar a la base de dades la llista de les cançons que componen un entrenament en l'ordre de reproducció assignat.

```
app.get('/api/reproduirEntrenament/:id/', (req, res) => {
  const { id } = req.params;
  const sqlConsultaOrdre = `SELECT activitats.idactivitat,
activitats.intensitat, ordre.ordre, cancons.link
FROM activitats
JOIN ordre ON activitats.idactivitat
= ordre.idact AND activitats.identrenament = ordre.identre
JOIN cancons ON ordre.idcanco =
cancons.id
WHERE activitats.identrenament = ?
ORDER BY activitats.idactivitat,
ordre.ordre`;
  db.all(sqlConsultaOrdre, [id], (err, rows) => {
    if (err) {
      return res.status(500).json({ error: err.message });
    }
    res.json(rows);
  });
});
```

**Codi 17.** Codi per a obtenir en ordre la llista de cançons d'un entrenament del fitxer `server.js`.

### 5.2.5 Algoritmes d'assignació i desassignació d'entrenaments a usuaris

Aquest algoritme facilita l'assignació i desassignació d'entrenaments a usuaris específics. Quan un entrenador o administrador decideix assignar un entrenament a un alumne, l'algoritme mostra una llista dels alumnes que no tenen entrenament assignat i en seleccionar-ne un, crea un enllaç entre l'usuari i l'entrenament a la base de dades. De manera similar, quan es desassigna un entrenament, l'algoritme elimina aquest enllaç, garantint que l'entrenament ja no aparegui a la llista d'entrenaments de l'usuari.

- **Codi Frontend:**

Codi per a comunicar-se amb el Backend i mostrar la llista d'alumnes que no tenen l'entrenament assignat.

```
mostrarAlumnes(entrenament: any) {
  this.selectedEntrenament = entrenament;
  this.http.get<any[]>(`/api/usuaris-alumnes/${
entrenament.id}`)
.subscribe({next: data => {
  this.alumnes = data;
  this.mostrarAssignar = true;
  this.mostrarDesassignar = false;
},
error: error => {
  console.error('Error obtenint alumnes:', error);
}
});
}
```

**Codi 18.** Funció `mostrarAlumnes()` del fitxer `seleccionar-entrenament.component.ts`.

Codi per enviar al Backend l'id d'usuari i entrenament i així poder inserir-los a la base de dades.

```
assignarEntrenament(alumneId: number) {
  if (!this.selectedEntrenament) {
    console.error('No hi ha entrenament seleccionat');
    return;
  }
  const body = { iduser: alumneId, identre:
this.selectedEntrenament.id };
  this.http.post('/api/assignar-entrenament',
body).subscribe({next: () => {
  alert('Entrenament assignat amb èxit');
  this.mostrarAssignar = false;
},
error: error => {
  console.error('Error assignant entrenament:', error);
}
});
}
```

**Codi 19.** Funció `assignarEntrenament()` del fitxer `seleccionar-entrenament.component.ts`.

Codi per a comunicar-se amb el Backend i mostrar la llista d'alumnes que tenen l'entrenament assignat.

```
mostrarAssignats(entrenament: any) {
  this.selectedEntrenament = entrenament;
  this.http.get<any[]>(`/api/usuarios-assignats/${
entrenament.id}`).subscribe({next: data => {
  this.alumnesAssignats = data;
  this.mostrarAssignar = false;
  this.mostrarDesassignar = true;
},
error: error => {
  console.error('Error obtenint alumnes assignats:', error);
}
});
}
```

**Codi 20.** Funció `mostrarAssignats()` del fitxer `seleccionar-entrenament.component.ts`.

Codi per enviar al Backend l'id d'usuari i entrenament i així poder eliminar-los de la base de dades.

```
desassignarEntrenament(alumneId: number) {
  if (!this.selectedEntrenament) {
    console.error('No hi ha entrenament seleccionat');
    return;
  }
  const body = { iduser: alumneId, identre:
this.selectedEntrenament.id };
  this.http.delete(`/api/desassignar-entrenament/${alumneId}/${
this.selectedEntrenament.id}`).subscribe({next: () => {
  alert('Entrenament desassignat amb èxit');
  this.mostrarDesassignar = false;
},
error: error => {
  console.error('Error desassignant entrenament:', error);
}
});
}
```

**Codi 21.** Funció `desassignarEntrenament()` del fitxer `seleccionar-entrenament.component.ts`.

- **Codi Backend:**

Codi per a consultar la llista d'usuaris que no tenen assignat un entrenament en concret.

```
app.get('/api/usuarios-alumnes/:identre', (req, res) => {
  const { identre } = req.params;
  const sqlObtenirAlumnes = `SELECT * FROM usuarios
WHERE rol = "alumne"
AND id NOT IN (SELECT iduser FROM
userentr WHERE identre = ?)`;
```

```

    db.all(sqlObtenirAlumnes, [identre], (err, rows) => {
      if (err) {
        res.status(400).json({ error: err.message });
        return;
      }
      res.json(rows);
    });
  });
});

```

**Codi 22.** Codi per a consultar usuaris que no tenen l'entrenament assignat del fitxer server.ts.

Codi per a inserir l'id de l'alumne i l'entrenament a la base de dades de forma que l'entrenament quedi assignat a l'alumne.

```

app.post('/api/assignar-entrenament', (req, res) => {
  const { iduser, identre } = req.body;
  const sqlAssignarEntrenament = 'INSERT INTO userentr (iduser,
identre) VALUES (?, ?)';
  db.run(sqlAssignarEntrenament, [iduser, identre], (err) => {
    if (err) {
      res.status(500).json({ error: `Error assignant
entrenament: ${err.message}` });
      return;
    }
    res.json({ message: 'Entrenament assignat amb èxit' });
  });
});

```

**Codi 23.** Codi per a consultar assignar l'entrenament a un alumne del fitxer server.ts.

Codi per a consultar la llista d'usuaris que tenen assignat un entrenament en concret.

```

app.get('/api/usuaris-assignats/:identre', (req, res) => {
  const { identre } = req.params;
  const sqlObtenirAssignats = `
    SELECT usuaris.* FROM usuaris
    JOIN userentr ON usuaris.id = userentr.iduser
    WHERE userentr.identre = ?
  `;
  db.all(sqlObtenirAssignats, [identre], (err, rows) => {
    if (err) {
      res.status(400).json({ error: err.message });
      return;
    }
    res.json(rows);
  });
});

```

**Codi 24.** Codi per a consultar usuaris que tenen l'entrenament assignat del fitxer server.ts.

Codi per a eliminar l'id de l'alumne i l'entrenament a la base de dades de forma que l'entrenament quedi desassignat de l'alumne.

```
app.delete('/api/desassignar-entrenament/:iduser/:identre', (req,
res) => {
  const { iduser, identre } = req.params;
  const sqlDesassignarEntrenament = 'DELETE FROM userentr WHERE
iduser = ? AND identre = ?';
  db.run(sqlDesassignarEntrenament, [iduser, identre], (err) => {
    if (err) {
      res.status(500).json({ error: `Error desassignant
entrenament: ${err.message}` });
      return;
    }
    res.json({ message: 'Entrenament desassignat amb èxit' });
  });
});
```

**Codi 25.** Codi per a consular desassignar l'entrenament a un alumne del fitxer server.ts.

### 5.2.6 Algorisme de creació d'entrenaments

Aquest algorisme permet als entrenadors crear entrenaments personalitzats seleccionant la durada, intensitat i música per a cada activitat. Una vegada l'entrenador ha concretat el nombre d'activitats i la durada i la intensitat d'aquestes, l'algorisme proposa una llista de cançons i dona la capacitat d'editar la llista ja sigui canviant, afegint o eliminant cançons, permetent així crear un entrenament completament personalitzat si es desitja. Les configuracions de cada entrenament i l'ordre de les cançons d'aquest es guarden a la base de dades, facilitant la seva posterior reproducció.

- **Codi Frontend:**

Codi per a enviar al backend les dades de l'entrenament, així com les activitats que el componen i les cançons per les quals estan formades.

```
crearEntrenament() {
  if (!this.revisat) {
    this.plenarCancons();
    return;
  }

  if (!this.canconsSeleccionades ||
this.canconsSeleccionades.length === 0) {
    this.missatgeError = 'No s\'han seleccionat cançons per a
les activitats';
    return;
  }

  const entrenament = {
    nom: this.nom,
    activitats: this.canconsSeleccionades.map((elementLlista: any,
indexActivitat: number) => ({
      temps: elementLlista.activitat.temps,
      intensitat: elementLlista.activitat.intensitat,
      idact: indexActivitat + 1,
      cancons: elementLlista.cancons.map((canco: any,
indexCanco: number) => ({
```

```

        idcanco: canco.id,
        ordre: indexCanco + 1,
    )))
    )))
};

this.http.post('/api/entrenament', entrenament).subscribe({next:
response => {
    const identre = (response as { id: number }).id;
    entrenament.activitats.forEach((activitat: any) => {
        activitat.cancons.forEach((canco: any) => {
            this.http.post('api/ordre', {
                identre,
                idact: activitat.idact,
                idcanco: canco.idcanco,
                ordre: canco.ordre
            }).subscribe({next: response => {
                console.log('Cançó inserida amb l\'ordre
corresponent', response);
            }, error: error => {
                console.error('Error creant l\'ordre de les cançons:',
error);
            }
        });
    });
});
});

console.log('Entrenament creat:', response);
// Netejar formulari
this.nom = '';
this.duracioEntrenament = 0;
this.activitats = [{temps: 0, intensitat: ''}];
this.canconsSeleccionades = [];
this.revisat = false;
this.navigateTo('entrenaments');
},
error: error => {
    console.error('Error creant entrenament:', error);
}
});
}
}

```

**Codi 26.** Funció crearEntrenament() del fitxer crear-entrenament.component.ts.

- **Codi Backend:**

Codi per a inserir a la base de dades els detalls de l'entrenament i les activitats que el componen.

```

app.post('/api/entrenament', (req, res) => {
    console.log('Sol·licitud rebuda:', req.body);
    const { nom, activitats } = req.body;
    const duracio = activitats.reduce((total, activitat) => total +
activitat.temps, 0);

```

```

    const sqlInserirEntrenament = 'INSERT INTO entrenaments (nom,
duracio) VALUES (?, ?)';

    db.run(sqlInserirEntrenament, [nom, duracio], function (err) {
        if (err) {
            res.status(400).json({ error: `Error en crear
l'entrenament: ${err.message}` });
            return;
        }
        const idEntrenament = this.lastID;
        let idactivitat = 1;
        const sqlInserirActivitats = 'INSERT INTO activitats
(identrenament, idactivitat, temps, intensitat) VALUES (?, ?, ?, ?)';
        for (const activitat of activitats) {
            db.run(sqlInserirActivitats, [idEntrenament,
idactivitat, activitat.temps, activitat.intensitat], function (err) {
                if (err) {
                    res.status(400).json({ error: `Error en crear
activitat: ${err.message}` });
                    return;
                }
            });
            idactivitat++;
        }
        res.status(201).json({ id: idEntrenament, message:
`Entrenament creat amb èxit amb l'ID: ${idEntrenament}` });
    });
});

```

### **Codi 27.** Codi per a crear un entrenament del fitxer server.js.

Codi per a inserir les cançons que componen cada activitat de l'entrenament i l'ordre d'aquestes dins de cada activitat.

```

app.post('/api/ordre', (req, res) => {
    const { identre, idact, idcanco, ordre } = req.body;
    console.log('Dades rebudes per inserir en ordre:', { identre,
idact, idcanco, ordre });

    const sqlInserirOrdre = 'INSERT INTO ordre (identre, idact,
idcanco, ordre) VALUES (?, ?, ?, ?)';
    db.run(sqlInserirOrdre, [identre, idact, idcanco, ordre], (err)
=> {
        if (err) {
            console.error(`Error al inserir ordre: ${err.message}`);
            res.status(500).json({ error: `Error al inserir ordre: $
${err.message}` });
            return;
        }
        res.json({ message: 'Ordre inserit amb èxit' });
    });
});

```

### **Codi 28.** Codi per a inserir l'ordre de les cançons del fitxer server.js.

### 5.2.7 Algoritme d'eliminació d'entrenaments

Aquest algoritme gestiona l'eliminació d'entrenaments del sistema. Quan un entrenador o administrador decideix eliminar un entrenament, l'algoritme s'encarrega de suprimir totes les dades associades a aquest entrenament, incloent-hi les activitats, l'ordre de les cançons assignades, i les relacions amb els usuaris. Això assegura que no quedin dades residuals a la base de dades, mantenint la integritat i neteja de la informació.

- **Codi Frontend:**

Codi per a enviar al backend l'id d'un entrenament per a poder eliminar-lo.

```
eliminarEntrenament(id: number) {
  if (this.rol !== 'professor' && this.rol !== 'admin') {
    alert('Els alumnes no poden eliminar entrenaments.');
```

```
=> {
  this.http.delete(`/api/entrenament/${id}`).subscribe({next: data
  this.obtindreEntrenaments();
}, error: error => {
  console.error('Error eliminant l\'entrenament:', error);
});
}
```

**Codi 29.** Funció eliminarEntrenament() del fitxer seleccionar-entrenament.component.ts.

- **Codi Backend:**

Codi per a eliminar tots els registres de l'entrenament que es troben a les taules *userentr*, *entrenaments*, *activitats* i *ordre*.

```
app.delete('/api/entrenament/:id', (req, res) => {
  const { id } = req.params;
  const sqlEliminarEntrenament = 'DELETE FROM entrenaments WHERE
id = ?';
  const sqlEliminarActivitats = 'DELETE FROM activitats WHERE
identrenament = ?';
  const sqlEliminarOrdre = 'DELETE FROM ordre WHERE identre = ?';
  const sqlEliminarUserEntr = 'DELETE FROM userentr WHERE identre
= ?';

  db.serialize(() => {
    db.run(sqlEliminarEntrenament, [id], (err) => {
      if (err) {
        res.status(400).json({ error: `Error en eliminar
l'entrenament: ${err.message}` });
        return;
      }
      db.run(sqlEliminarActivitats, [id], (err) => {
        if (err) {
          res.status(400).json({ error: `Error en eliminar
activitats: ${err.message}` });
          return;
        }
        db.run(sqlEliminarOrdre, [id], (err) => {
```



```

        this.intensitatsSeleccionades.add('Baix');
    }

    this.llistaCancons = this.cancons.filter(canco =>
        canco.bpm >= this.rangBPM[0] &&
        canco.bpm <= this.rangBPM[1] &&
        (this.intensitatsSeleccionades.size === 0 ||
this.intensitatsSeleccionades.has(canco.intensitat)
    );
    this.obrirMenu = false;
}

```

**Codi 32.** Funció aplicarFiltres() del fitxer seleccionar-musica.component.ts.

Codi per a netejar el filtres del menú i mostrar totes les cançons.

```

netejarFiltres() {
    this.rangBPM = [0, 200];
    this.intensitatsSeleccionades.clear();
    this.intensitatAlta = false;
    this.intensitatMitja = false;
    this.intensitatBaixa = false;
    this.llistaCancons = [...this.cancons];
    this.obrirMenu = false;
}

```

**Codi 33.** Funció netejarFiltres() del fitxer seleccionar-musica.component.ts.

## 6 Avaluació

En aquest apartat es presenta una avaluació del sistema desenvolupat, tant pel que fa als requisits funcionals com als no funcionals. L'objectiu d'aquesta secció és validar que l'aplicació compleix amb els criteris establerts durant la fase de disseny, assegurant així que el producte final s'ajusta a les necessitats dels usuaris i als objectius definits inicialment. A través de l'anàlisi dels requisits funcionals, es comentarà quines funcionalitats implementades operen correctament i quines no han complert els objectius proposats, comprovant alhora que es mantenen els estàndards de qualitat en termes de rendiment, seguretat i usabilitat. Aquesta avaluació permetrà identificar possibles punts de millora i garantir que l'aplicació ofereix una experiència d'usuari òptima.

### 6.1 Avaluació dels requisits funcionals

En aquesta secció, es realitzarà una avaluació detallada dels requisits funcionals del sistema. Es presentaran els casos de prova corresponents a cadascun dels requisits funcionals identificats prèviament, amb l'objectiu de verificar que cada funció de l'aplicació es comporta de manera correcta segons les especificacions. Cada prova inclourà una descripció dels objectius, les entrades i sortides esperades, així com els passos detallats per a la seva execució i els resultats obtinguts. Aquest procés permetrà assegurar que totes les funcionalitats crítiques de l'aplicació han estat implementades correctament i que estan preparades per a ser utilitzades en un entorn real.

#### RF1. Autenticació d'usuari

- **Objectiu:** Verificar que els usuaris puguin iniciar sessió correctament introduint el seu nom d'usuari i contrasenya, i que només els usuaris autenticats puguin accedir a les funcionalitats de l'aplicació segons el seu rol.
- **Entrades:** Nom d'usuari, contrasenya.
- **Sortides Esperades:** Amb credencials correctes s'ha de dirigir als alumnes a la secció d'entrenaments mentre que als professors i administradors se'ls ha de redirigir al panell per a seleccionar acció. En cas de credencials incorrectes s'ha de mostrar un missatge d'error i deixar a l'usuari tornar a intentar-ho.
- **Passos de la prova:**
  1. Introduir nom d'usuari i contrasenya.
  2. Fer clic al botó "Iniciar sessió".
  3. En cas d'haver introduït unes credencials correctes, comprovar que es redirigeix als usuaris a la pàgina adient al seu rol.
  4. En cas d'haver introduït unes credencials incorrectes, comprovar que es mostra el missatge d'error i es deixa tornar a intentar-ho.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que els usuaris amb credencials vàlides accedeixen a l'aplicació, mentre que els intents amb credencials incorrectes resulten en un missatge d'error.

#### RF2. Mostrar llista de cançons

- **Objectiu:** Comprovar que els entrenadors i administradors puguin consultar la llista completa de cançons emmagatzemades a la base de dades i que poden usar els filtres correctament.
- **Entrades:** Accés a la secció de la llista de cançons, diferents opcions de filtres.
- **Sortida Esperada:** Visualització de la llista completa a l'accedir a la secció i visualització de les cançons que compleixen únicament els filtres seleccionats.

- **Passos de la prova:**
  1. Accedir a la secció de cançons des del panell principal.
  2. Verificar que es mostren totes les cançons de la base de dades.
  3. Aplicar diferents filtres.
  4. Comprovar que es mostren únicament les cançons que compleixen els filtres seleccionats.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que l'accedir a la secció de la llista de cançons s'observen totes les cançons de la base de dades i una vegada s'han aplicat filtres, es mostren únicament les cançons que els compleixen.

### RF3. Descàrrega i anàlisi de música

- **Objectiu:** Verificar que l'aplicació permeti la descàrrega de música des de YouTube i l'anàlisi dels BPM i altres característiques de les cançons.
- **Entrada:** URL de Youtube.
- **Sortides Esperades:** Descàrrega correcta de l'àudio en format *mp3*, anàlisi correcte dels BPM i duració, correcte emmagatzematge a la base de dades.
- **Passos de la prova:**
  1. Accedir a la secció de pujar música.
  2. Introduir l'URL de Youtube i fer clic a "Descarregar"
  3. Comprovar al servidor que la cançó es descarrega, s'analitza i s'elimina correctament.
  4. Comprovar que es mostren el BPM correctes i altres detalls com la durada de l'àudio o el títol del vídeo a Youtube.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que les cançons es descarreguen correctament, s'analitzen les dades necessàries i s'eliminen del servidor. De la mateixa manera, el formulari es mostra amb les dades completes correctament.

### RF4. Creació d'entrenaments personalitzats

- **Objectiu:** Assegurar que els entrenadors poden crear entrenaments personalitzats amb les activitats, intensitat i música personalitzades. Introduint únicament el temps i la intensitat de les activitats, l'aplicació ha d'omplir les activitats amb cançons fins que s'arribi, com a mínim, al temps seleccionat. Una vegada s'ha omplert la llista, s'ha de donar la possibilitat d'editar cançons.
- **Entrades:** Detalls de l'entrenament (nom i nombre d'activitats), detalls de les activitats (intensitat i duració).
- **Sortides Esperades:** En cas de detalls incorrectes mostrar un error, correcte emplenament del temps de les activitats, correcta edició de la llista de cançons, correcta creació de l'entrenament.
- **Passos de la prova:**
  1. Accedir a la secció de creació d'entrenaments.
  2. Introduir el nom de l'entrenament i crear diferents activitats sense omplir tots els camps obligatoris.
  3. Obtenir missatge d'error informant que s'han d'omplir tots els camps.
  4. Omplir tots els camps requerits.
  5. Comprovar que es mostra la llista de cançons a la part inferior i que, com a mínim, arriben al temps especificat per a cada activitat.
  6. Comprovar que es poden afegir, eliminar i editar cançons.
  7. Confirmar l'entrenament i comprovar que s'ha creat correctament mostrant-se a la llista d'entrenaments.

- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que en cas de no introduir tots els camps requerits no es mostra la llista de cançons i es mostra un error indicant que s'han d'omplir tots els camps. Una vegada s'han omplert els camps les activitats s'emplenen correctament i que les cançons es poden editar correctament. Finalment, un cop creat, es pot comprovar que l'entrenament s'ha creat correctament mostrant els detalls d'aquest.

#### **RF5. Mostrar llista d'entrenaments en funció del rol**

- **Objectiu:** Verificar que els usuaris amb rol d'alumne veuen únicament els entrenaments assignats, mentre que entrenadors i administradors poden veure tots els entrenaments.
- **Entrades:** En cas de tenir el rol "alumne" iniciar sessió, en cas de tenir rol "professor" o "administrador" accedir a la secció d'entrenaments.
- **Sortida Esperada:** Visualització dels entrenaments depenent del rol de l'usuari.
- **Passos de la prova:**
  1. Iniciar sessió com a alumne
  2. Verificar que es mostra directament la secció d'entrenaments i únicament es veuen els entrenaments assignats.
  3. Tancar sessió i iniciar de nou com a professor.
  4. Accedir a la secció d'entrenaments i comprovar que es visualitzen tots els entrenaments de la base de dades.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que es mostra la llista corresponen segons el rol de l'usuari.

#### **RF6. Gestió d'entrenaments**

Aquest requisit funcional cobreix diverses funcions relacionades amb la gestió dels entrenaments dins de l'aplicació. Aquestes funcions van dirigides a usuaris amb rol d'entrenador o administrador i inclouen la visualització dels detalls d'un entrenament, la reproducció de les cançons assignades a un entrenament, l'assignació o desassignació d'entrenaments a alumnes, i l'eliminació d'entrenaments del sistema. Tot seguit, s'explicarà cada funció de manera individual.

##### **RF6.1. Mostrar detalls d'un entrenament**

- **Objectiu:** Permetre als usuaris visualitzar els detalls complets d'un entrenament específic, incloent-hi les activitats, la durada de cada activitat i les cançons associades.
- **Entrada:** Clicar el botó "mostrar" de la fila de l'entrenament desitjat.
- **Sortida Esperada:** Visualització en la part inferior els detalls complets de l'entrenament, incloent-hi les activitats, la seva durada, intensitat, i les cançons assignades.
- **Passos de la prova:**
  1. Clicar botó "mostrar" de l'entrenament desitjat.
  2. Verificar que els detalls es mostren correctament.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que es mostren els detalls de l'entrenament correctament.

**RF6.2. Reproduir un entrenament**

- **Objectiu:** Permetre la reproducció de les cançons associades a un entrenament de manera ininterrompuda, seguint l'ordre establert per l'entrenador.
- **Entrada:** Pantalla de reproducció de cançons amb el reproductor carregat des que s'accedeix a la pàgina, clicar el botó “reproduir” de la fila de l'entrenament.
- **Sortida Esperada:** Reproducció de l'entrenament sense interrupcions iniciant la següent cançó quan l'actual acaba.
- **Passos de la prova:**
  1. Fer clic al botó “Reproduir” de l'entrenament desitjat.
  2. Comprovar que la cançó es reproduïx correctament.
  3. Moure's al final de la cançó per a comprovar que es passa a la següent en finalitzar.
  4. Comprovar que les cançons es reproduïxen totes, una activitat darrere de l'altra.
- **Resultats obtinguts:** Aquest requisit no s'ha assolit correctament, ja que l'accedir a la secció el reproductor no es veu a la pàgina i és necessari recarregar la pàgina per a poder visualitzar el reproductor. Una vegada el reproductor està carregat, les cançons es reproduïxen de manera correcta.

**RF6.3. Assignar o desassignar entrenaments**

- **Objectiu:** Permetre als entrenadors i administradors assignar entrenaments a usuaris amb rol d'alumne, així com desassignar entrenaments quan sigui necessari.
- **Entrada:** Clicar el botó “assignar” o “desassignar”, seleccionar un usuari de la llista mostrada.
- **Sortida Esperada:** Assignació o desassignació correcta en funció de l'opció seleccionada.
- **Passos de la prova:**
  1. Clicar el botó “Assignar” de l'entrenament desitjat.
  2. De la llista d'alumnes de la part inferior, seleccionar a quin es vol assignar l'entrenament.
  3. Comprovar que l'entrenament s'ha assignat correctament. Això es pot fer directament fent clic al botó “Desassignar” per a mostrar la llista d'alumnes assignats, o bé es pot iniciar sessió amb l'usuari que s'ha assignat l'entrenament per a comprovar si és visible a la llista.
  4. Clicar el botó “Desassignar” de l'entrenament desitjat.
  5. De la llista d'alumnes que el tenen assignat, escollir a quin es vol desassignar.
  6. Comprovar que l'entrenament s'ha desassignat correctament. De la mateixa manera que l'anterior, aquesta acció es pot fer directament fent clic al botó “Assignar” per a mostrar la llista d'alumnes que no tenen l'entrenament assignat, o bé es pot iniciar sessió amb l'usuari que s'ha desassignat l'entrenament per a comprovar si és visible a la llista.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que els entrenaments s'assignen o desassignen correctament als alumnes. De la mateixa manera, les llistes que es mostren són correctes mostrant únicament els alumnes que tenen o no l'entrenament assignat.

**RF6.4. Eliminar entrenaments**

- **Objectiu:** Permetre als entrenadors i administradors eliminar entrenaments que ja no siguin necessaris del sistema, incloent-hi tota la informació associada a l'entrenament.
- **Entrada:** Clicar el botó “Eliminar” de l'entrenament desitjat.
- **Sortida Esperada:** L'entrenament és eliminat completament de la base de dades (incloent-hi les referències de les taules *activitats*, *userentr* i *ordre*), i desapareix de la llista d'entrenaments disponibles.
- **Passos de la prova:**
  1. Clicar el botó “Eliminar” de l'entrenament que es vol eliminar.
  2. Comprovar que s'actualitza la llista i l'entrenament ja no apareix en la llista i que totes les referències a aquest s'han eliminat.
- **Resultats obtinguts:** Aquest requisit s'ha assolit correctament, ja que l'entrenament s'elimina completament de la base de dades i s'actualitza la llista per a mostrar-la amb els entrenaments restants.

**6.2 Avaluació dels Requisits No Funcionals**

En aquest apartat s'avaluen els requisits no funcionals definits per a l'aplicació, amb l'objectiu de determinar si compleixen amb els estàndards establerts. Aquesta avaluació es basa en proves específiques realitzades durant el desenvolupament i la implementació del sistema.

**RNF1. Usabilitat**

- **Objectiu:** Verificar que l'aplicació és intuïtiva i fàcil d'utilitzar, permetent als entrenadors crear, editar i assignar entrenaments sense necessitat de formació extensa.
- **Passos de la prova:** Realitzar proves amb usuaris sense experiència prèvia en l'aplicació, observant la seva capacitat per completar les tasques sense dificultats.
- **Resultats:** Els usuaris han pogut navegar per l'aplicació i utilitzar les seves funcionalitats bàsiques sense problemes, confirmant que el sistema compleix amb el requisit.

**RNF2. Eficiència de processament d'àudio**

- **Objectiu:** Assegurar que l'aplicació processa la descàrrega i l'anàlisi de música de manera eficient, amb temps de resposta adequats.
- **Passos de la prova:** Mesurar el temps de resposta en la descàrrega i anàlisi de diverses cançons.
- **Resultats:** El sistema ha respost dins del marge de temps esperat en la majoria dels casos, sense un temps d'espera molt elevat.

**RNF3. Gestió correcta de l'eliminació d'entrenaments**

- **Objectiu:** Verificar que l'eliminació d'un entrenament elimina correctament totes les dades associades.
- **Passos de la prova:** Eliminar diversos entrenaments i verificar que no queda cap dada residual en la base de dades.
- **Resultats:** Els entrenaments s'eliminen completament sense deixar cap fila que tingui el seu id.

**RNF4. Compatibilitat amb diferents navegadors**

- **Objectiu:** Garantir que l'aplicació funciona correctament en diferents navegadors web.
- **Passos de la prova:** Dur a terme proves d'ús en diferents navegadors.

- **Resultats:** L'aplicació ha funcionat correctament en tots els navegadors que s'ha testejat.

#### **RNF5. Seguretat de les dades**

- **Objectiu:** Assegurar que les dades personals i sensibles estan protegides, amb especial atenció al xifratge de contrasenyes.
- **Passos de la prova:** Revisar el mètode d'emmagatzematge de contrasenyes en la base de dades.
- **Resultats:** Les contrasenyes estan xifrades mitjançant l'algoritme bcrypt, assegurant un alt nivell de seguretat per a les credencials dels usuaris.

#### **RNF6. Responsivitat**

- **Objectiu:** Assegurar que l'aplicació s'adapta correctament a diferents resolucions i dispositius.
- **Passos de la prova:** Provar l'aplicació en diferents resolucions, incloent-hi pantalles de dispositius mòbils.
- **Resultats:** S'ha observat que no totes les parts de l'aplicació són totalment responsives, encara que es poden utilitzar en dispositius mòbils i tauletes.

#### **RNF7. Compatibilitat amb diferents dispositius**

- **Objectiu:** Assegurar que independentment del sistema operatiu l'aplicació funciona correctament.
- **Passos de la prova:** Provar l'aplicació en diferents ordinadors amb diferents sistemes operatius.
- **Resultats:** L'aplicació ha funcionat correctament en tots els sistemes que s'ha testejat.

## 7 Procés normal d'execució

En aquest apartat es procedirà a explicar el procés d'execució de l'aplicació tenint en compte que ja es tenen tots els requisits instal·lats i la base de dades creada correctament. Es començarà per l'arrencament dels servidors i continuant amb les diferents vistes depenent del rol de l'usuari que iniciï sessió.

### 7.1 Inici servidors

1. Iniciar el servidor del backend en un terminal. Si tot és correcte, es veurà un missatge de confirmació.

**Comanda:** `node nomDelFitxerBackend.js`

```
PS E:\OneDrive - URV\5e curs xd - TFG\Web\TFG> cd .\backend\  
PS E:\OneDrive - URV\5e curs xd - TFG\Web\TFG\backend> node server.js  
Servidor en funcionament al port http://localhost:3000  
Connectat a la base de dades amb èxit!
```

**Figura 27.** Inicialització del backend.

2. Iniciar el servidor de desenvolupament del frontend en un nou terminal al mateix temps que el backend. Si s'ha iniciat correctament, es veurà una confirmació amb l'URL que podem ingressar al navegador.

**Comanda:** `npm start`

Un cop introduïda la comanda, es començarà el procés de "Building", on Angular compila el codi TypeScript en JavaScript i prepara tots els fitxers necessaris per executar l'aplicació en el navegador.

```
PS E:\OneDrive - URV\5e curs xd - TFG\Web\TFG> npm start  
  
> tfg@0.0.0 start  
> ng serve --proxy-config src/proxy.conf.json  
  
⋄ Building...
```

**Figura 28.** Frontend compilant-se.

Un cop s'ha compilat el frontend, veurem que al terminal la confirmació i l'URL al que hem de dirigir-nos per tal de poder accedir a l'aplicació.

```
Application bundle generation complete. [7.791 seconds]  
  
Watch mode enabled. Watching for file changes...  
NOTE: Raw file sizes do not reflect development server per-request transformations.  
→ Local: http://localhost:4200/  
→ press h + enter to show help
```

**Figura 29.** Frontend compilat correctament.

3. Ara que els servidors estan iniciats, ens podem dirigir a l'URL <http://localhost:4200/> per tal d'accedir a l'aplicació.

**Figura 30.** Pàgina d'inici de sessió.

A continuació veurem les dues possibles vistes de l'aplicació. Per un costat la vista de l'usuari amb rol alumne i per l'altre la vista del que tenen rol de professor o administrador.

## 7.2 Vista rol “alumne”

1. Introduïm les credencials d'un usuari que tingui rol “alumne” amb entrenaments assignats. En aquest cas iniciarem sessió amb l'usuari *albert* que com es pot comprovar per les imatges de la base de dades, té el rol “alumne” i 3 entrenaments assignats.

**Figura 31.** Credencials de l'usuari *albert*.

id	nom	cognoms	rol	user	pass
Filtro	Filtro	Filtro	Filtro	ert ✕	Filtro
2	Albert	Carbonell Marti	alumne	albert	123

**Figura 32.** Registre de l'usuari *albert* a la taula “usuaris”.

iduser	identre
2 <input type="checkbox"/>	Filtro
2	50
2	53
2	54

**Figura 33.** Registres amb l'id del usuari *albert* (id 2) a la taula “userentr”.

2. Comprovar que un cop s’ha iniciat sessió la pàgina que es mostra és la secció d’entrenaments i s’ha carregat correctament amb el reproductor de Youtube.

The screenshot shows a web browser at localhost:4200/entrenaments-assignats. A 'Tancar sessió' button is visible. Below it is the heading 'Entrenaments' and a table with the following data:

ID	Nom	Duració Total	Accions
50	test 4	39	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>
53	Test ordre 2	27	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>
54	123	5	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>

Below the table is a large black rectangle representing a video player, with a red play button icon in the center.

**Figura 34.** Vista de l’usuari *albert* un cop està carregada correctament.

3. A continuació es triarà l'opció de mostrar els detalls d'un entrenament. En aquest cas, s'ha decidit mostrar l'entrenament anomenat "test 4".

### Entrenaments

ID	Nom	Duració Total	Accions
50	test 4	39	Mostrar Reproduir
53	Test ordre 2	27	Mostrar Reproduir
54	123	5	Mostrar Reproduir

#### Detalls de l'entrenament seleccionat:

**Nom:** test 4

**Duració Total:** 39 mins

#### Cançons per l'entrenament seleccionat:

ID	Temps	Intensitat	Cançons
1	13	Baix	Till I Collapse, Still D.R.E., ATWA (Official Audio)
2	12	Mig	Yes Sir, I Can Boogie, Starman, Ingovernable
3	14	Alt	Los Perros , Toro, Romantika

Figura 35. Detalls de l'entrenament "test 4".

4. Finalment, triarem reproduir l'entrenament del qual s'han mostrat els detalls i es pot comprovar que les cançons mostrades a la llista de detalls són les que es reproduïxen.

#### Detalls de l'entrenament seleccionat:

**Nom:** test 4

**Duració Total:** 39 mins

#### Cançons per l'entrenament seleccionat:

ID	Temps	Intensitat	Cançons
1	13	Baix	Till I Collapse, Still D.R.E., ATWA (Official Audio)
2	12	Mig	Yes Sir, I Can Boogie, Starman, Ingovernable
3	14	Alt	Los Perros , Toro, Romantika



Figura 36. Entrenament "test 4" reproduint la primera cançó de l'entrenament.

5. Si amb el rol alumne intentem accedir editant l'URL a qualsevol altra secció que no sigui la d'entrenaments, es mostrarà un missatge d'error.



ERROR: No tens permisos per accedir a aquesta pàgina.

**Figura 37.** Vista quan l'usuari *albert* intenta accedir a la secció del panell per a seleccionar acció editant l'URL.

### 7.3 Vista rol “professor” o “administrador”

1. Introduïm les credencials d'un usuari que tingui rol “professor”. En aquest cas iniciarem sessió amb l'usuari *sergi* que com es pot comprovar per les imatges de la base de dades, té el rol “professor”.

A login form with the title "Login". It contains two input fields: the first contains the text "sergi", and the second contains three dots. Below the fields is a black button with the text "Login" in white.

**Figura 38.** Credencials usuari *sergi*.

id	nom	cognoms	rol	user	pass
Filtro	Filtro	Filtro	Filtro	gi <input checked="" type="checkbox"/>	Filtro
3	Sergi	Calatayud Masip	professor	sergi	123

Figura 39. Registre de l'usuari *sergi* a la taula "usuaris".

- Comprovar que un cop s'ha iniciat sessió es mostra el panell per seleccionar acció.

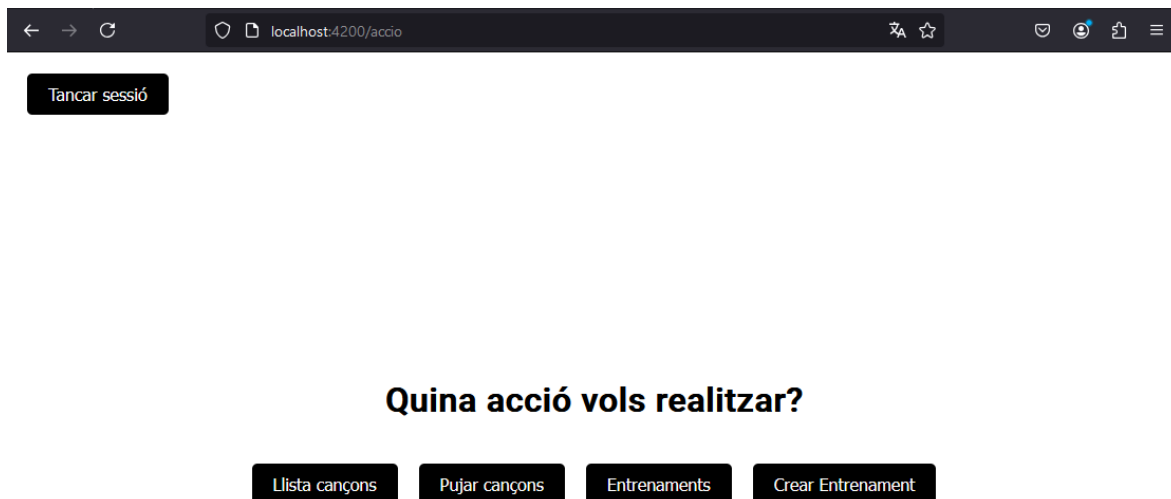


Figura 40. Vista de l'entrenador en iniciar sessió.

- Accedir a la secció de la llista de cançons per a poder buscar cançons si fos necessari.

Títol	Duració(min)	BPM	Intensitat	Artista	Link
Los Perros	3:19	143.5546875	Alt	Arde Bogota (Andres Campo & K-Style Remix)	https://www.youtube.com/watch?v=yL80hazGak
Till I Collapse	4:57	86.1328125	Baix	Eminem	https://www.youtube.com/watch?v=P3_Zs-oRLo
Still D.R.E.	4:50	92.28515625	Baix	Dr. Dreft, Snoop Dogg	https://www.youtube.com/watch?v=-CL6n0fJZpk
Yes Sir I Can Boogie	4:40	123.046875	Mig	Baccara	https://www.youtube.com/watch?v=5b-zHhZMIQ
Starman	4:14	99.38401442307692	Mig	David Bowie	https://www.youtube.com/watch?v=sBKE3MNMNM
Ingovernable	3:06	103.359375	Mig	C. Tangana, Gipsy Kings, Nicolás Reyes, Tonino Ballardo	https://www.youtube.com/watch?v=UzPKP3KpGQ
Toro	6:11	151.9990888235293	Alt	El columbia asesino (I HATE MODELS remix)	https://www.youtube.com/watch?v=dqEVSitwgZo
Ballantines	5:19	92.28515625	Baix	Voladores del verso	https://www.youtube.com/watch?v=atYQ1qhsuE
Romantika	4:10	161.4990234375	Alt	Brutalismus 3000	https://www.youtube.com/watch?v=jR9D0WKTfKo
ATWA (Official Audio)	2:57	89.10290948275862	Baix	System Of A Down	https://www.youtube.com/watch?v=g3k3j5t5Xg
Toxicity (Official HD Video)	3:44	117.45383522727273	Mig	System Of A Down	https://www.youtube.com/watch?v=yw80MvYLI
FE-A-I-A-I-O (Official Audio)	3:10	135.99917763157896	Alt	System Of A Down	https://www.youtube.com/watch?v=eud0we5Rz-Q
Radio (Official Audio)	4:10	143.5546875	Alt	System Of A Down	https://www.youtube.com/watch?v=DmmYAXBPhg
Cigaro (Official Audio)	2:13	107.666015625	Mig	System Of A Down	https://www.youtube.com/watch?v=L4M98Z2Zpgl
Pota salvaje	3:26	135.99917763157896	Alt	Isabel Aaun	https://www.youtube.com/watch?v=CZ-zSADfSw
Eye Of The Tiger (Official HD Video)	4:04	107.666015625	Mig	Survivor	https://www.youtube.com/watch?v=bFjPfnesV4

Figura 41. Vista de la llista de cançons.

4. Aplicar un o més filtres i comprovar que les cançons que es mostren coincideixen amb aquests.

The screenshot shows a web application interface. At the top left, there are two buttons: "Tancar sessió" and "Inici". Below them is the heading "Llista cançons". A search bar contains the text "Cerca cançó". Below the search bar is a table with the following data:

Títol	Duracio(min)	BPM	Intensitat	Artista	Link
Starman	4:14	99.38401442307692	Mig	David Bowie	https://www.youtube.com/watch?v=aBKEt3MhNM

On the right side, there is a sidebar titled "Filtres" with a close button "X". Under "Intensitat", there are three radio buttons: "Alt" (unchecked), "Mig" (checked), and "Baix" (unchecked). Under "BPM", there are two sliders and the text "BPM Range: 70 - 100". At the bottom of the sidebar are two buttons: "Aplicar Filtres" and "Netejar Filtres".

**Figura 42.** Cançons que coincideixen amb els filtres d'intensitat mitjana i bpm entre 70 i 100.

5. A continuació s'accedirà a la secció de pujar música, tornant a l'inici amb el botó "inici" de la part superior esquerra. En aquesta secció s'introduirà l'URL del vídeo *The Beatles – I want to hold your hand* per a pujar-la a la base de dades.

The screenshot shows a web browser window with the address bar displaying "localhost:4200/pujar". The page has a heading "Llista cançons" in a dark box. Below it is a search bar containing the URL "www.youtube.com/watch?v=NCTzkaL2t\_Y" and a "Cercar" button. Below the search bar, there is a message: "Espera mentre es descarrega i s'analitza la canço..."

**Figura 43.** Vista de la web mentre descarrega i analitza l'àudio.

6. A continuació es mostraran els detalls de la cançó, es podrà editar el títol si així es desitja i s'haurà d'emplenar el camp d'artista (és obligatori i no es pot obtenir a partir de les dades del vídeo). En aquest cas s'editarà el títol del vídeo per a eliminar la part de "The Beatles" i s'inclourà aquesta informació al camp d'artista. Finalment, es farà clic al botó "Pujar cançó" per a inserir-la a la base de dades.

Lista cançons

www.youtube.com/watch?v=ipADNIW7yBM **Cercar**

I Want To Hold your Hand [HD]  
The Beatles

Duració: 2:23 minuts  
BPM: 129.19921875

**Pujar cançó** **Cancelar**

Tingues en compte que els rangs de BPM per a determinar el ritme seran els següents:

Ritme "Baix": menys de 95 BPM  
Ritme "Mig": entre 95 i 130 BPM  
Ritme "Alt": mes de 130 BPM

**Figura 44.** Cançó analitzada i llesta per a ser inserida a la base de dades.

Tancar sessió

Inici

**Llista cançons** **Pujar cançó**

i want **Cerca** **Filtres**

Títol	Duracio(min)	BPM	Intensitat	Artista	Link
I Want To Hold your Hand [HD]	2:23	129.19921875	Mig	The Beatles	https://www.youtube.com/watch?v=ipADNIW7yBM

**Figura 45.** Comprovació de què la cançó s'ha inserit correctament a la base de dades.

7. Tot seguit, anirem a la secció de creació d'entrenaments i crearem un entrenament anomenat "test execució" amb dues activitats. La primera tindrà una intensitat baixa i una durada de 15 minuts i la segona, una intensitat mitjana i 10 minuts de duració.

**Entrenaments** **Crear Entrenament**

**Nom:**

**Duració Total (mins):**

**Activitat 1**

**Temps (mins):**

**Intensitat:**

**Eliminar Activitat**

**Activitat 2**

**Temps (mins):**

**Intensitat:**

**Eliminar Activitat**

**Afegir Activitat** **Crear Entrenament**

**Figura 46.** Introducció dels paràmetres de l'entrenament.

8. A continuació es procedirà a fer clic al botó “Crear entrenament” per a veure les cançons proposades i si es desitja es podran editar. Cal recordar que l’aplicació emplenarà el temps amb cançons fins a arribar al temps determinat. Si s’excedeix el temps determinat pel professor, s’actualitza el temps perquè es pugui comprovar el temps de cada activitat i el total.

## Revisar Cançons Seleccionades

**Activitat 1: Baix**

Temps: 15 mins

- Till I Collapse (4:57 mins)
- Still D.R.E. (4:50 mins)
- Ballantines (5:19 mins)

**Activitat 2: Mig**

Temps: 12 mins

- Yes Sir, I Can Boogie (4:40 mins)
- Starman (4:14 mins)
- Ingobernable (3:06 mins)

**Figura 47.** Llista de cançons proposades per a l’entrenament creat.

9. Finalment, es fa clic a “Confirmar” i l’entrenament s’inserirà a la base de dades i l’usuari serà redirigit a la llista d’entrenaments. Un cop aquí, es pot comprovar que l’entrenament existeix i es poden realitzar les opcions comentades als Requisits Funcionals. En aquest cas, com ja s’han mostrat i reproduït entrenament amb els usuaris alumnes, s’assignarà l’entrenament a l’alumne *albert* perquè aquest el tingui disponible.

## Entrenaments

ID	Nom	Duració Total	Accions				
49	test repro2	22	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
50	test 4	39	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
53	Test ordre 2	27	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
54	123	5	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
57	test 420	51	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
58	test execució	27	Mostrar	Reproduir	Assignar	Desassignar	Eliminar

### Assignar entrenament a un alumne

Albert	Assignar
Marti	Assignar

**Figura 48.** Assignació de l’entrenament a l’alumne *albert*.

## Entrenaments

ID	Nom	Duració Total	Accions	
50	test 4	39	Mostrar	Reproduir
53	Test ordre 2	27	Mostrar	Reproduir
54	123	5	Mostrar	Reproduir
58	test execució	27	Mostrar	Reproduir

**Figura 49.** Vista dels entrenaments per a l’usuari *albert*.

10. Un cop l'entrenament ja no sigui necessari, l'entrenador el pot eliminar si no necessita usar-lo més.

Inici

Tancar sessió

## Entrenaments

ID	Nom	Duració Total	Accions				
49	test repro2	22	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
50	test 4	39	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
53	Test ordre 2	27	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
54	123	5	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
57	test 420	51	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
58	test execució	27	Mostrar	Reproduir	Assignar	Desassignar	Eliminar

Figura 50. Eliminació de l'entrenament.

## Entrenaments

ID	Nom	Duració Total	Accions				
49	test repro2	22	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
50	test 4	39	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
53	Test ordre 2	27	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
54	123	5	Mostrar	Reproduir	Assignar	Desassignar	Eliminar
57	test 420	51	Mostrar	Reproduir	Assignar	Desassignar	Eliminar

Figura 51. Vista dels entrenaments per al professor després d'eliminar l'entrenament *test execució*.

## Entrenaments

ID	Nom	Duració Total	Accions
50	test 4	39	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>
53	Test ordre 2	27	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>
54	123	5	<input type="button" value="Mostrar"/> <input type="button" value="Reproduir"/>

**Figura 52.** Vista dels entrenaments per a l'usuari *albert* després d'eliminar l'entrenament *test execució*.

## 8 Conclusions

L'objectiu principal d'aquest projecte era desenvolupar una aplicació web que permetés als entrenadors crear, gestionar i assignar entrenaments personalitzats amb música seleccionada segons els BPM. Aquest objectiu s'ha complert satisfactòriament, i l'aplicació resultant ofereix una solució eficaç per a la gestió d'entrenaments, facilitant la tasca dels entrenadors i millorant l'experiència dels usuaris finals.

A diferència d'altres projectes acadèmics, aquest projecte té una aplicació directa a la vida diària, la qual cosa ha estat una motivació significativa durant tot el procés de desenvolupament. La possibilitat de crear una eina pràctica que pugui ser utilitzada en el món real ha estat un dels principals motors per a enfrontar els reptes que han sorgit.

Una de les principals dificultats ha estat la necessitat d'aprendre i aplicar Angular, un framework amb el qual no tenia experiència prèvia. La formació ha estat completament autodidacta, fet que ha afegit un nivell de complexitat extra al desenvolupament. No obstant això, aquest desafiament ha resultat en l'adquisició de noves habilitats que han estat clau per a la implementació d'una interfície d'usuari moderna i funcional.

Durant el procés, he pogut aplicar molts dels coneixements adquirits durant el grau. Les assignatures de bases de dades han estat fonamentals per a la gestió eficient de la informació en a la base de dades, mentre que les relacionades amb programació han proporcionat una base sòlida per a crear l'estructura del codi. Finalment, l'experiència adquirida en l'assignatura de sistemes de comerç electrònic, on vaig treballar amb Vue.js i Node.js, m'ha estat molt útil per a entendre i implementar correctament l'arquitectura del projecte.

El desenvolupament d'aquest projecte també ha servit per enfortir la meua capacitat de resolució de problemes i gestió del temps, sobretot tenint en compte la naturalesa autodidacta del procés. La necessitat d'autogestionar l'aprenentatge i d'aplicar-lo en un projecte real m'ha ajudat a preparar-me millor per als desafiaments professionals que es presentaran en el futur.

## 9 Bibliografia

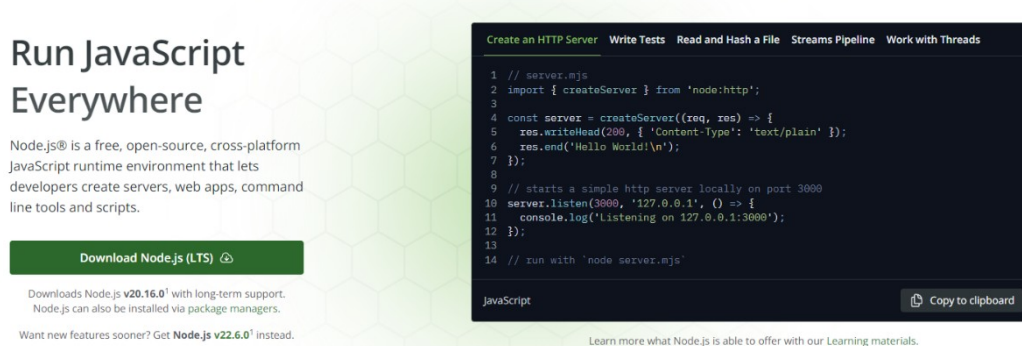
- [1] Pàgina Web <https://docs.angular.lat/guide/architecture>. [Definició Angular]
- [2] Pàgina Web [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp). [Definició HTML]
- [3] Pàgina Web [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp). [Definició CSS]
- [4] Pàgina Web <https://www.typescriptlang.org>. [Definició Typescript]
- [5] Pàgina Web <https://nodejs.org/en/about>. [Definició Node.js]
- [6] Pàgina Web <https://expressjs.com>. [Definició Express]
- [7] Pàgina Web <https://www.arimetrics.com/glosario-digital/http-2>. [Definició HTTP]
- [8] Pàgina Web <https://aws.amazon.com/es/what-is/sql/>. [Definició SQL]
- [9] Pàgina Web <https://www.sqlite.org/about.html>. [Definició SQLite]
- [10] Pàgina Web <https://github.com/yt-dlp/yt-dlp>. [Llibreria Python]
- [11] Pàgina Web [https://librosa.org/doc/latest/generated/librosa.beat.beat\\_track.html](https://librosa.org/doc/latest/generated/librosa.beat.beat_track.html). [Llibreria Python]
- [12] Pàgina Web [https://developers.google.com/youtube/iframe\\_api\\_reference?hl=es](https://developers.google.com/youtube/iframe_api_reference?hl=es). [Llibreria Python]
- [13] Pàgina Web <https://github.com/RogerLluch/TFG>. [Repositori amb el codi font]

## 10 Annexes

### 10.1 Guia de posada en marxa

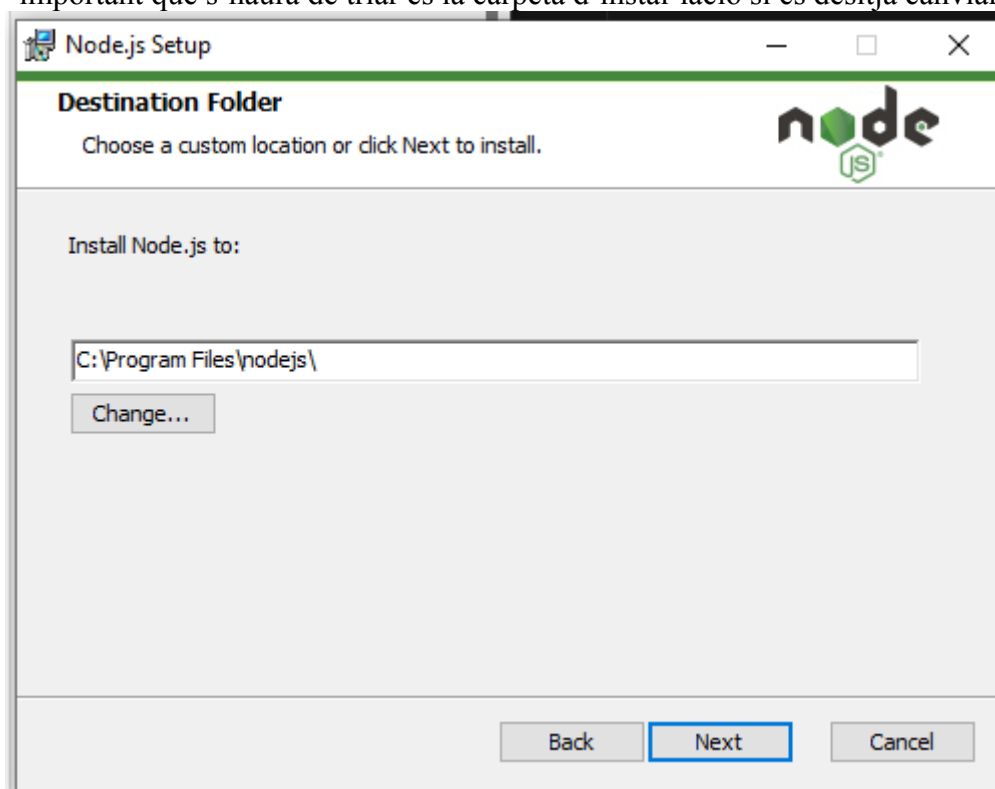
En aquesta secció, es proporcionarà una guia detallada per posar en marxa per primera vegada l'aplicació en un ordinador amb el sistema operatiu Windows 10. Aquesta guia parteix de la base que l'usuari ja té instal·lat *Visual Studio Code* i ha descarregat la carpeta completa del projecte.

1. Anar a la web oficial per a descarregar *Node.js* (<https://nodejs.org/en>).

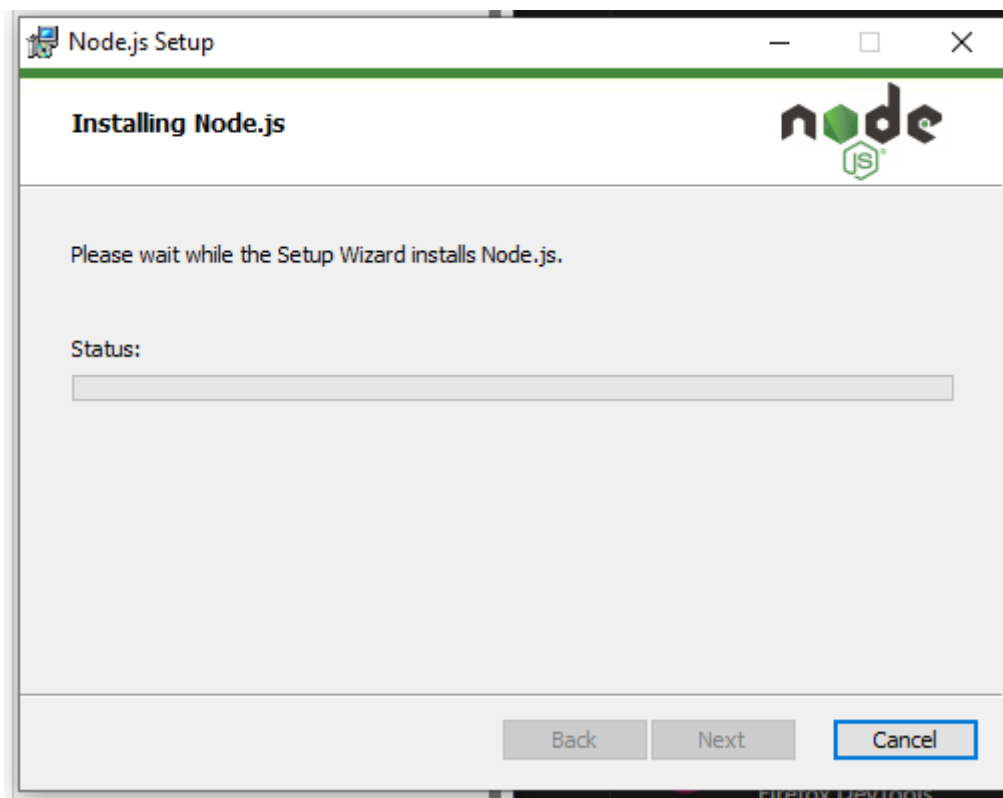


**Figura 53.** Pàgina de descàrrega de Node.js.

2. El procés d'instal·lació de *Node.js* és similar a altres programes, on l'única selecció important que s'haurà de triar és la carpeta d'instal·lació si es desitja canviar.

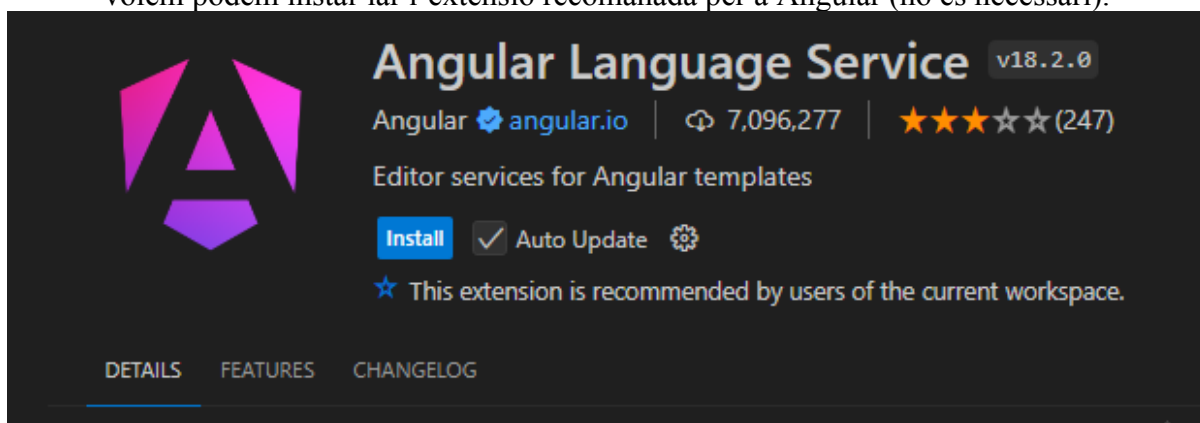


**Figura 54.** Selecció de la carpeta d'instal·lació per a Node.js.



**Figura 55.** Procés d'instal·lació de Node.js.

3. Un cop instal·lat Node.js, obrim la carpeta del projecte amb *Visual Studio Code* i si volem podem instal·lar l'extensió recomanada per a Angular (no és necessari).



**Figura 56.** Extensió recomanada per a aplicacions d'Angular.

4. A continuació, obrirem un terminal a la carpeta arrel del projecte per a instal·lar els paquets necessaris executant la comanda `npm install`.

```
PS C:\Users\roger\Desktop\TFG> npm install

removed 39 packages, and audited 1029 packages in 4s

144 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 10.8.1 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice
PS C:\Users\roger\Desktop\TFG> █
```

**Figura 57.** Comanda per a instal·lar els paquets necessaris del frontend.

5. Tot seguit, obrirem un nou terminal i ens dirigirem a la carpeta **backend** amb la comanda `cd .\backend\` per a tornar a executar la comanda `npm install`.

```
PS C:\Users\roger\Desktop\TFG> cd .\backend\
PS C:\Users\roger\Desktop\TFG\backend> npm install

up to date, audited 206 packages in 1s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\roger\Desktop\TFG\backend> █
```

**Figura 58.** Comandes per a navegar a la carpeta del backend i instal·lar els paquets necessaris.

6. En aquest punt, ja tindrem tots els paquets necessaris per a executar l'aplicació i podrem executar les comandes mostrades en l'apartat anterior (anomenat Procés normal d'execució) per a arrancar els servidors.

```
PS C:\Users\roger\Desktop\TFG\backend> node server.js
Servidor en funcionament al port http://localhost:3000
Connectat a la base de dades amb èxit!
█
```

**Figura 59.** Comanda per a iniciar el servidor del backend.

```

PS C:\Users\roger\Desktop\TFG> npm start

> tfg@0.0.0 start
> ng serve --proxy-config src/proxy.conf.json

Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.dev/cli/analytics.

no
Global setting: not set
Local setting: disabled
Effective status: disabled
Initial chunk files | Names          | Raw size
main.js             | main           | 95.88 kB |
polyfills.js       | polyfills     | 90.23 kB |
styles.css         | styles        | 76.22 kB |

| Initial total | 262.33 kB

Application bundle generation complete. [3.746 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
Re-optimizing dependencies because vite config has changed
  → Local:   http://localhost:4200/
  → press h + enter to show help

```

**Figura 60.** Comanda per a iniciar el servidor de desenvolupament del frontend.

## 10.2 Codi de l'aplicació

En aquest apartat, es proporciona un enllaç al repositori de GitHub que conté el codi complet de l'aplicació desenvolupada. Aquest codi inclou totes les funcionalitats descrites en aquest projecte, així com els diferents mòduls i components que han estat implementats. Al fitxer *README.md* es trobaran les instruccions necessàries per a iniciar l'aplicació.

Es pot accedir al repositori fent clic [aquí](#).