



Phonocat, disseny i desenvolupament d'una aplicació web de gestió de col·leccions de vinils

Llorenç Isern Martín
Grau d'Enginyeria informàtica
Desenvolupament Web

Antoni Oller Arcas

Juny 2021



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Phonocat, disseny i desenvolupament d'una aplicació web de gestió de col·leccions de vinils</i>
Nom de l'autor:	<i>Llorenç Isern Martín</i>
Nom del consultor:	<i>Antoni Oller Arcas</i>
Data de lliurament (mm/aaaa):	<i>06/2021</i>
Àrea del Treball Final:	<i>Desenvolupament Web</i>
Titulació:	<i>Grau d'Enginyeria informàtica</i>

Resum del Treball (màxim 250 paraules):

La indústria musical en l'actualitat, obté els seus beneficis principalment dels serveis de reproducció en línia i plataformes basades en subscripcions. No obstant això, encara hi ha un format físic, el disc de vinil, que no tan sols resisteix davant d'aquest fet, sinó que incrementa les seves vendes exponencialment cada any.

Aquest projecte neix amb el propòsit de crear un espai digital que permeti satisfer les necessitats d'aquest segment del mercat, així com connectar als seus usuaris entre si.

Phonocat s'ha desenvolupat de manera seqüencial mitjançant una metodologia en cascada amb retroalimentació. Aquest document permet veure en detall les diferents fases del projecte, des de l'anàlisi, el disseny i la implementació fins al desplegament al núvol del producte final.

L'aplicació s'ha implementat des de zero amb un perfil full-stack, que inclou algunes de les tecnologies web més populars avui dia.

Com ha resultat, s'ha obtingut un producte mínim viable integrat a plataformes al núvol, que es connecta amb altres serveis per alimentar-se de fonts externes. L'eina permet als usuaris, entre altres funcionalitats, registrar i gestionar les seves col·leccions de discos, anunciar-los en venda i fer ofertes de compra mitjançant una interfície web responsiva i multi-idioma que proporciona una experiència d'usuari òptima.

Abstract (in English, 250 words or less):

Today, the music industry makes most of its money from subscription-based streaming services. However, there is still one physical media, the vinyl record, which is increasing its sales every year against all odds.

This project was started with the aim of building a digital space where this market segment can meet its needs, as well as connecting its users with each other.

Phonocat has been developed with a linear approach using the waterfall model, allowing cycle iterations when necessary. This document describes in detail the different steps of this project, from the analysis, design, and implementation to cloud deployment.

This web application was made from scratch using a full stack development profile, which includes some of the most popular and demanded web technologies these days.

The result is an MVP application running in a cloud server that feeds from external sources in order to fetch data. This product allows its users to store and manage their records, put them on sale and make offers for other albums through a responsive, multi-language user interface that provides an optimal user experience.

Paraules clau (entre 4 i 8):

Fullstack, SPA, Node, React, Vinils

Agraïments

Als meus pares, que em van inculcar la
importància de continuar aprenent.

A la Gloria,
per fer-me costat en tot moment.

Índex

1	Introducció.....	1
1.1	Context i justificació del Treball.....	1
1.2	Objectius del Treball.....	2
1.3	Enfocament i mètode seguit.....	3
1.4	Planificació del Treball	4
1.5	Breu sumari de productes obtinguts.....	9
1.6	Breu descripció dels altres capítols de la memòria	9
2	Anàlisi i disseny.....	10
2.1	Descripció de l'aplicació	10
2.2	Actors principals	11
2.3	Anàlisi de requisits	12
2.4	Abast.....	24
2.5	Disseny del sistema	25
2.6	Prototipatge UI	30
3	Implementació.....	32
3.1	Background tecnològic.....	32
3.2	Desenvolupament del backend	37
3.3	Desenvolupament del frontend	42
3.4	Integració al núvol	50
3.5	Resultat del producte	52
4	Conclusions.....	58
4.1	Valoració del projecte.....	58
4.2	Línies de treball futur.....	58
5	Glossari.....	60
6	Bibliografia	62
	Annex I. Especificació API REST Phonocat.....	64

Annex II. Manual d'instal·lació.....	74
Annex III. Llibre d'estil	76

Llista de figures

Figura 1.1: Evolució de vendes de discos de vinil als EUA	1
Figura 1.2: Model en cascada amb iteració de cicles	3
Figura 1.3: Diagrama de Gantt planificació inicial	5
Figura 2.1: Diagrama de casos d'ús	13
Figura 2.2: Matriu de prioritització de requisits.....	24
Figura 2.3: Entitats model de dades.....	26
Figura 2.4: Punt de vista de la computació	28
Figura 2.5: Diagrama de seqüència - Afegir LP.....	29
Figura 2.6: Prototip lo-fi Col·lecció LP	30
Figura 2.7: Prototip lo-fi afegir LP pas 1	30
Figura 2.8: Prototip lo-fi afegir LP pas 2	30
Figura 2.9: Prototip lo-fi afegir LP pas 3	30
Figura 2.10: Prototip lo-fi perfil d'usuari.....	31
Figura 2.11: Prototip lo-fi LPs en venda	31
Figura 2.12: Prototip lo-fi Ofertes	31
Figura 2.13: Prototip lo-fi llistat d'usuaris.....	31
Figura 2.14: Prototip lo-fi vistes dispositiu mòbil.....	31
Figura 3.1: Arquitectura tecnològica de l'aplicació	36
Figura 3.2: Organització codi font backend	37
Figura 3.3: Execució middlewares pujar portada LP	39
Figura 3.4: Organització del codi font client	42
Figura 3.5: Components React perfil d'usuari	43
Figura 3.6: Flux redux-saga.....	47

Figura 3.7: Estat aplicació petició col·lecció LP	47
Figura 3.8: Estat aplicació col·lecció LP carregada satisfactòriament.....	48
Figura 3.9: Validacions formulari registre	49
Figura 3.10: Aplicació desplegada al núvol	51
Figura 3.11: Landing page	52
Figura 3.12: Landing page mòbil	52
Figura 3.13: Login	53
Figura 3.14: Registre	53
Figura 3.15: Col·lecció de discos	53
Figura 3.16: Col·lecció buida.....	53
Figura 3.17: Cercar àlbum.....	54
Figura 3.18: 3.18: Afegir LP.....	54
Figura 3.19: Portada suggerida	54
Figura 3.20: Anunciar LP en venda	54
Figura 3.21: LP anunciat	54
Figura 3.22: Llistat de discos en venda	55
Figura 3.23: Filtratge anuncis	55
Figura 3.24: Realitzar oferta	55
Figura 3.25: Oferta realitzada.....	55
Figura 3.26: Ofertes rebudes.....	56
Figura 3.27: Valorar transacció	56
Figura 3.28: Ofertes realitzades	56
Figura 3.29: Llistat d'usuaris.....	56
Figura 3.30: Detall d'usuari.....	56
Figura 3.31: Top menu	57
Figura 3.32: User profile.....	57

Llista de taules

Taula 1.1: Objectius de l'aplicació	2
Taula 1.2: Riscos identificats.....	6
Taula 1.3: Estimació de temps	8
Taula 3.1: Comparativa Angular React	34

Llista de fragments de codi

Fragment de codi 3.1: endpoint routing.....	38
Fragment de codi 3.2: Exemple controlador.....	38
Fragment de codi 3.3: Generació JWT.....	40
Fragment de codi 3.4: Pre-hook mongoose, encriptar password	41
Fragment de codi 3.5: control estat llenguatge aplicació.....	44
Fragment de codi 3.6: Exemple ús redux-saga	46
Fragment de codi 3.7: Esquema de validació formulari registre.....	49
Fragment de codi 3.8: adaptació entorn producció	51
Fragment de codi 3.9: utilitat pujar fitxers cloudinary	52

1 Introducció

Aquest capítol conté una descripció del context i justificació del treball, els seus objectius, la metodologia emprada i el pla de treball. Tanmateix, s'enuncia un llistat dels productes obtinguts fruit de la seva implementació i es descriu breument el contingut de la resta de capítols d'aquesta memòria.

1.1 Context i justificació del Treball

És innegable que el mercat actual pel que fa a la indústria discogràfica cada cop es tendeix més a l'aparició de plataformes de reproducció en línia que permeten als seus clients consumir continguts a canvi de subscripcions temporals en lloc de la pròpia venda de discos de música. No obstant això, existeix un nínxol de mercat d'apassionats de la música que per diversos motius entre els quals s'inclouen la nostàlgia, el col·leccionisme o el simple fet de voler escoltar discos d'altres temps en el format en el qual es van concebre, prefereixen compaginar l'ús d'aquests serveis amb la compra de discos en format físic.

En particular, des de fa uns anys, es pot observar com un d'aquests formats, el vinil, està vivint un ressorgiment i com en contrast a altres formats en declivi com els discos digitals, cada any incrementa substancialment el seu volum de vendes¹. És més, no sols els grans segells discogràfics llancen al mercat noves reedicions de grans èxits de la música en discos de 33 rpm, sinó que arreu del món apareixen petits negocis destinats a fabricar discos de vinil d'artistes menys populars pel gran públic.

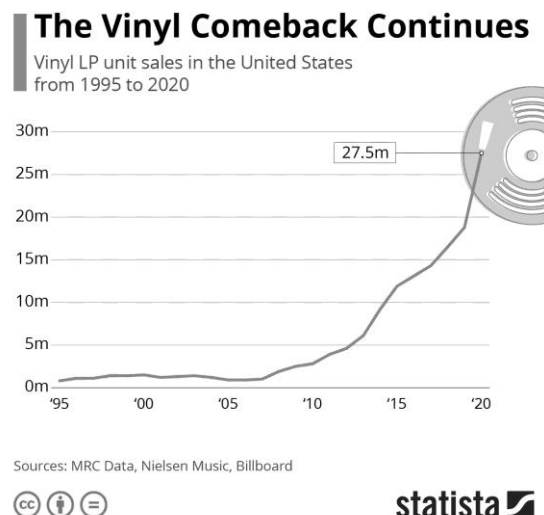


Figura 1.1: Evolució de vendes de discos de vinil als EUA

¹ Ritcher F.(2021, 18 de gener), Vinyl sales in the United States. Statista [article en línia] <https://www.statista.com/chart/7699/lp-sales-in-the-united-states/>

Aquest treball sorgeix amb el propòsit de dissenyar i crear una aplicació que permeti satisfer les necessitats d'aquests usuaris potencials, com són el registre, manteniment i ampliació de les seves col·leccions d'àlbums, i amb l'ambició d'alimentar aquest segment de la indústria i contribuir a la seva perdurabilitat.

1.2 Objectius del Treball

A aquest apartat s'enumeren els objectius del treball, tant pel que fa al producte resultant com en relació amb els objectius acadèmics i personals de l'autor del treball.

Objectius de l'aplicació ordenats per prioritat:

Objectiu	Prioritat
Dissenyar i implementar una aplicació web que serveixi com a eina de gestió de col·leccions de discos de vinil.	Alta
Definir el producte mínim viable de l'aplicació	Alta
Implementar un flux d'autenticació i autorització segur	Alta
Exposar diferents serveis mitjançant una API REST (estratègia B2C).	Alta
Connectar l'aplicació amb serveis externs per alimentar-se de fonts de dades foranies (estratègia B2B).	Alta
Fer servir una base de dades NoSQL orientada a documents	Alta
Presentar una interfície web atractiva i responsiva	Alta
Fer ús de tecnologies escalables davant l'augment d'usuaris	Alta
Afegir funcionalitats de mercadeig a l'aplicació que fomentin la venda i l'intercanvi de segona mà.	Alta
Internacionalitzar l'aplicació a més d'un idioma i regió.	Mitjana
Afegir funcionalitats socials a l'aplicació per interconnectar els seus usuaris.	Mitjana
Crear una interfície d'usuari atractiva basada en Material Design ^[2]	Mitjana
Mantenir el codi font de l'aplicació amb l'ús d'una eina de control de versions	Mitjana
Explorar diferents opcions per desplegar l'aplicació al núvol.	Mitjana

Taula 1.1: Objectius de l'aplicació

Objectius acadèmics i professionals:

- Consolidar i posar en pràctica les competències transversals del grau així com els coneixements assolits en desenvolupament de programari i tecnologies web.
- Explorar i aprofundir en les tendències actuals del desenvolupament web.
- Adquirir noves competències en algunes de les tecnologies web més populars en l'actualitat.
- Planificar, gestionar i documentar un projecte TIC d'inici a fi.

1.3 Enfocament i mètode seguit

L'aplicació s'ha concebut com a un producte nou. Tot i que existeixen solucions tecnològiques completes que permeten escurçar el temps de desenvolupament, sovint van unides a una menor flexibilitat. Tanmateix, alguns dels objectius del treball són investigar quines eines i llenguatges de programació són tendència actualment al desenvolupament web i aprofundir en elles.

Per a aquest projecte, s'ha previst implementar una solució a mida que afavoreixin l'experiència d'usuari. En conseqüència, s'ha dedicat una part important del projecte a decidir quines eines tecnològiques s'hi adaptaven millor i s'han fet servir per desenvolupar l'aplicació des de zero.

Com a metodologia^[1] s'han investigat dues de les estratègies més rellevants en la gestió de projectes, el model en cascada i la metodologia àgil. S'ha decidit fer servir el model en cascada, ja que no es preveuen canvis importants respecte als requeriments inicials. Aquest model ens permet desenvolupar el projecte d'una manera lineal. Com veurem a l'apartat "Planificació del Treball", les fites temporals de cada cicle han pres com a referència el model d'avaluació de l'assignatura.

Amb tot, el model s'ha modificat lleugerament per permetre la iteració dels cicles anteriors quan l'aplicació ho ha requerit, d'aquesta manera s'obté un model híbrid i es guanya agilitat en el desenvolupament. Es pot parlar doncs d'un model en cascada amb iteració de cicles.

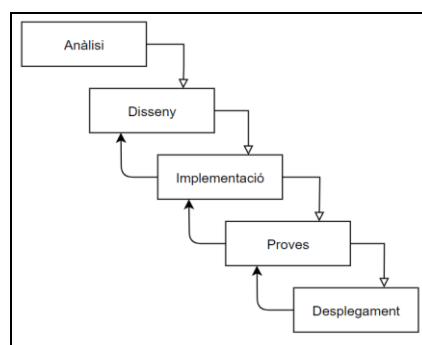


Figura 1.2: Model en cascada amb iteració de cicles

1.4 Planificació del Treball

A continuació es detalla la planificació inicial del projecte i es fa una valoració dels riscos i possibles desviacions que s'han previst. Seguidament es mostren els recursos que s'han destinat a la seva execució.

1.4.1 Planificació inicial

La planificació inicial del treball es va executar fent servir com a guia la duració de l'assignatura del TFG de desenvolupament web i prenent com a orientació les següents proves d'avaluació continuada.

- **PAC1:** Elaboració del pla de treball. Inclou la descripció del projecte, el context i justificació, els objectius que es volen assolir, la planificació temporal inicial i una proposta de definició tecnològica del projecte.
- **PAC2:** Anàlisi funcional i disseny tecnològic. A aquesta PAC es treballarà l'anàlisi i disseny de l'aplicació mitjançant la definició dels requeriments, el model de casos d'ús, diagrames UML, una primera especificació dels endpoints de la API REST i un primer prototip gràfic a baix nivell del programa. A més, es decidirà l'arquitectura tecnològica definitiva juntament amb una prova de concepte.
- **PAC3:** Implementació de l'aplicació. Aquesta fase se centra en el desenvolupament del programari web, tant al seu vessant de servidor com d'interfície d'usuari. A més, es farà una bateria de proves manuals d'acceptació per avaluar el producte.
- **Lliurament final:** Confecció de la memòria del TFG, el vídeo de presentació virtual i l'informe d'auto avaluació de competències transversals.

Aquestes fites es van desglossar en tasques i es va crear un diagrama de Gantt amb l'objectiu de poder visualitzar la seva distribució en el temps. Convé ressaltar que aquesta planificació inicial va servir com a guia i que s'han produït algunes modificacions al llarg del projecte que es comentaran a aquest mateix capítol.

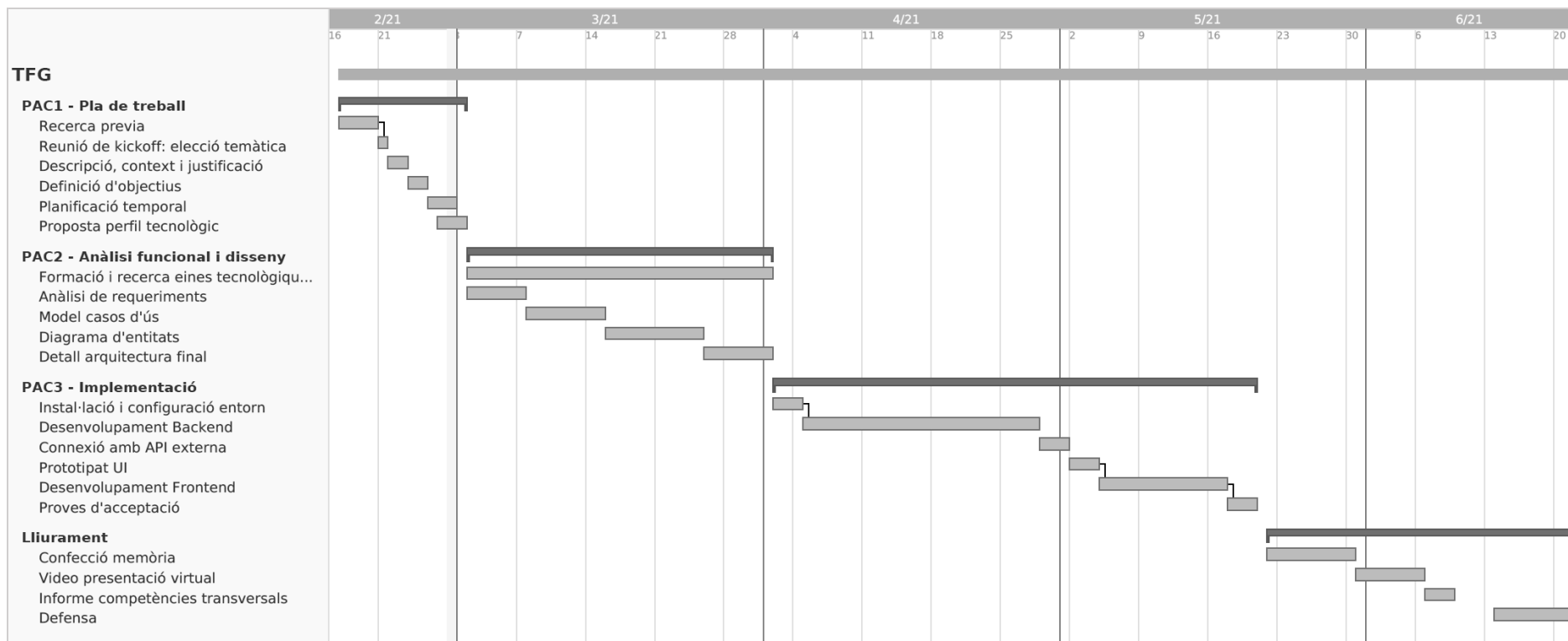


Figura 1.3: Diagrama de Gantt planificació inicial²

A la gràfica (figura 1.3) s'inclou la distribució temporal de cadascuna d'aquestes tasques en blocs setmanals compresos entre l'inici del treball el 17 de febrer fins al final de la defensa el 21 de juny de 2021.

² Diagrama generat amb l'eina teamgantt <<https://www.teamgantt.com/>>

1.4.2 Valoració de riscos

Al llarg del desenvolupament d'un projecte poden sorgir imprevistos i obstacles que alterin la planificació i/o la realització del projecte. Per aquest motiu, és recomanable identificar i preveure quins riscos es poden presentar i quines mesures s'hauran de dur a terme per tal de minimitzar el seu impacte.

Per a aquest projecte, s'han identificat els següents:

Codi	Risc	Probabilitat	Impacte	Mesures correctores
R01	Desviació de la planificació degut a l'ús de tecnologies desconegudes	Alta	Alt	Dedicar un temps de la planificació a la formació i recerca de les eines a emprar i fer una planificació conservadora
R02	Abast excessiu pel temps disponible	Alta	Mitjà	Definir un producte MVP amb les funcionalitats principals
R03	Necessitat de canviar de tecnologia o llibreries	Mitjana	Mitjà	Fer un estudi previ de les tecnologies i fer servir metodologies de programació que no estiguin excessivament lligades a una tecnologia o llibreria
R04	Fallades o avaries tècniques que derivin en pèrdua del progrés del treball	Mitjana	Baix	Emmagatzemar còpies del codi font i dels documents a plataformes al núvol
R05	Modificacions i canvis que provoquin fallades	Alta	Baix	Fer servir eines de control de versions
R05	Indisposició temporal de l'autor del treball per malaltia o motius personals	Indeterminada	Baix	En una situació de pandèmia s'ha de valorar la possibilitat de caure malalt o que algú de l'entorn ho faci. En conseqüència, és convenient extreure les mesures de precaució

Taula 1.2: Riscos identificats

1.4.3 Desviacions respecte la planificació inicial

La planificació inicial ha resultat ser bastant acurada en la majoria de les tasques. Hi ha dos punts, però, que han resultat més costosos en recursos del que inicialment es va definir.

En primer lloc, la recerca i formació s'ha allargat més del que es va valorar. A més, l'entorn de desenvolupament que finalment s'ha escollit s'integra amb un ecosistema de llibreries de diferent magnitud segons les problemàtiques que s'han volgut resoldre. Conseqüentment, la consulta de documentació, recerca

de tutorials i estudi d'alternatives ha estat una constant al llarg del cicle d'implementació del producte.

Un altre aspecte que ha resultat ser més costós del que es va preveure, ha estat el desenvolupament del frontal web. En retrospectiva, és lògic, ja que les tecnologies emprades fan un ús extensiu de la part client de l'aplicació.

Aquestes desviacions han tingut com a origen el fet que l'autor del treball era neòfit en moltes de les tecnologies emprades, no obstant això, aquests contratemps ja es van tenir en compte a la valoració de riscos, la qual cosa ha ajudat a minimitzar el seu impacte. Val la pena dir que formen part de les lliçons apreses que s'analitzen a les conclusions del projecte.

1.4.4 Recursos

Recursos Humans

La realització del projecte s'ha dut a terme per l'autor del projecte que ha fet alhora de gestor del projecte, dissenyador, desenvolupador i validador. També s'ha comptat amb l'orientació del consultor del TFG.

Recursos materials

Maquinari

- Ordinador personal AMD Ryzen 5 1600 amb 16GB de RAM i amb connexió a internet: Aquest PC s'ha fet servir per desenvolupar i documentar el projecte així com per elaborar la presentació.
- Dispositiu mòbil amb connexió a internet: Aquest dispositiu s'ha fet servir per realitzar proves de la interfície responsiva.

Infraestructures externes

- Cloudinary: Magatzem d'imatges i contingut multimèdia al núvol que s'ha fet servir per emmagatzemar les imatges pujades pels usuaris.
- Heroku: Plataforma al núvol que s'ha fet servir com a servidor per desplegar una versió pública de l'aplicació.
- MongoDB Atlas: Base de dades orientada a documents allotjada al núvol.
- GitHub: Servei per emmagatzemar el codi mitjançant l'eina de control de versions Git.

Programari:

- Windows 10 professional 64bit: Sistema operatiu desenvolupat per Microsoft.

- Visual Code Studio: Editor de codi font que s'ha emprat per desenvolupar el backend i el frontend.
- Postman: Aplicació client de la qual s'ha fet us per testejar l'API de l'aplicació.
- Microsoft Office: L'editor de text Microsoft Word i el magatzem al núvol OneDrive s'han fet servir per crear la documentació del projecte.
- Google Chrome amb extensions: Navegador web per accedir als recursos electrònics. L'extensió Pocket s'ha fet servir per desar i etiquetar els recursos web més rellevants.
- MagicDraw: Creació de diagrames UML.
- Gimp: Editor d'imatges.
- Draw.io: Altres diagrames i gràfics.
- Pixabay i Unsplash: Bancs d'imatges lliures de drets d'autor.
- Figma: Eina de prototipatge

1.4.5 Estimació de temps

La següent taula conté una valoració aproximada en hores del temps que ha portat desenvolupar el projecte, desglossada en tasques principals.

Tasca	Estimació de temps en hores
Gestió inicial del projecte: Context, objectius i planificació	22
Anàlisi de requisits i disseny funcional	45
Formació i recerca	75
Prototipat	15
Preparació entorn	4
Implementació Backend	80
Implementació Frontend	220
Proves	5
Redacció memòria	30
Elaboració presentació	16
Total	542

Taula 1.3: Estimació de temps

1.5 Breu sumari de productes obtinguts

Com a resultat del desenvolupament del present TFG, s'han obtingut els següents productes:

- Memòria final del TFG: Aquest document sintetitza tot el treball que s'ha dut a terme al llarg del treball de fi de grau.
- Especificació de l'API REST: L'Annex I inclou l'especificació de l'API de l'aplicació.
- Manual d'instal·lació a entorns locals: L'Annex II conté el manual d'instal·lació del programari per a executar l'aplicació a un entorn local
- Guia d'usuari: l'Annex III mostra una guia d'usuari amb captures de l'aplicació.
- Llibre d'estil: l'Annex IV indica quines convencions d'estil s'han fet servir pel desenvolupament de la interfície d'usuari.
- Codi font del backend i frontend de l'aplicació: Aquest treball inclou el codi font de l'aplicació, que també està publicat al repositori: <https://github.com/llorenc-isern-tfg/phonocat>
- L'aplicació s'ha desplegat a un contenidor d'aplicacions del servei al núvol Heroku. <https://phonocat.herokuapp.com/>
- Presentació en vídeo del TFG

1.6 Breu descripció dels altres capítols de la memòria

Els següents capítols de la memòria s'han estructurat de la següent manera:

Capítol 2, Anàlisi i disseny: Aquest capítol aprofundeix en l'anàlisi i disseny de l'aplicació amb elements com la descripció funcional, l'anàlisi de requisits, l'abast final del producte, el disseny arquitectònic del sistema i el prototipat en baixa fidelitat de l'aplicació.

Capítol 3, Implementació: El capítol 3 descriu el procés d'investigació amb el qual s'han escollit les tecnologies finals del producte i detalla alguns dels aspectes rellevants de la implementació del backend i frontend de l'aplicació. A més, conté una breu descripció de les utilitats que s'han fet servir a cada mòdul. Per acabar, fa referència a alguns aspectes de la integració al núvol de l'aplicació.

Capítol 4, Conclusions: Les conclusions fan una reflexió sobre el resultat del treball en relació amb els objectius que s'han definit, quines lliçons s'han après al llarg del desenvolupament de l'aplicació i quines millores futures i línies d'investigació es podrien aplicar al producte.

2 Anàlisi i disseny

Fins ara, s'han descrit el context i els objectius del producte que es vol desenvolupar. Aquest apartat detalla les característiques de l'aplicació des dels seus diferents punts de vista. Inclou, a més, una descripció d'alt nivell de l'aplicació, els actors principals involucrats, l'anàlisi de requisits funcionals i no funcionals, els casos d'ús, i l'especificació del model de dades i de l'arquitectura.

2.1 Descripció de l'aplicació

L'objectiu d'aquest apartat és descriure l'aplicació Phonocat, un espai digital que permetrà als seus usuaris registrar les seves col·leccions de discos i que a més, els connectarà mitjançant funcionalitats socials i de mercadeig.

A Phonocat s'hi podran registrar i identificar per mitjà d'un accés públic els usuaris que ho desitgin, ja sigui fent ús d'un correu electrònic personal que haurà de ser verificat per l'aplicació o bé mitjançant un compte de Google. Per completar el registre se'ls hi demanarà un nom d'usuari que serà visible a l'aplicació. L'usuari ha de poder introduir opcionalment la seva data de naixement, una imatge de perfil, el seu número de telèfon i el seu país de residència, a més s'ha de poder seleccionar en quin idioma es vol visualitzar l'aplicació. Un usuari podrà consultar aquestes dades i modificar-les sempre que ho consideri oportú, a excepció del nom d'usuari que serà únic i quedarà vinculat a la seva adreça electrònica.

Tanmateix, els usuaris identificats a Phonocat podran registrar i gestionar digitalment la seva col·lecció de discos o LP en vinil. Dels discos s'enregistrarà el títol, el segell discogràfic, la data d'inserció, el gènere musical, la condició física del disc, si té un format de reproducció en estèreo o mono, el nombre de discos que componen l'àlbum i la qualitat de premsa del vinil en grams. A fi de proporcionar una millor experiència d'usuari, l'aplicació es connectarà amb serveis externs que recuperaran algunes d'aquestes dades quan sigui possible i n'afegiran d'altres com l'autor principal de l'LP, les cançons que conté o fins i tot la imatge de la seva portada. A més, els usuaris podran afegir una puntuació i una ressenya dels seus discos i podran decidir si són visibles a les consultes dels altres usuaris o no. Els usuaris, podran consultar el detall dels seus discos enregistrats en qualsevol moment i, a més a més, podran editar les seves dades o eliminar-los amb la condició que no s'hagin posat abans en venda.

Altrament, Phonocat proporcionarà una plataforma de mercadeig on els usuaris podran posar en venda els discos de la seva col·lecció indicant el preu desitjat. Així mateix, podran consultar quins discos estan en venda per altres usuaris i fer ofertes per un import que l'usuari propietari del disc podrà acceptar o rebutjar. En cas d'acceptar la oferta, cadascun dels dos usuaris involucrats en la transacció podrà valorar-la amb una puntuació. Una vegada la transacció hagi estat completada, és a dir, valorada pels dos usuaris, el disc passarà de la col·lecció de l'usuari venedor a la de l'usuari comprador automàticament. Cal tenir present que els usuaris podran retirar un disc que han posat en venda sempre que aquest no tingui una oferta acceptada.

Per potenciar aquestes funcionalitats de mercadeig, l'aplicació inclourà les següents funcionalitats socials. D'una banda, els usuaris podran seguir a altres usuaris amb els que tinguin afinitat per poder consultar les seves col·leccions. De l'altra, l'aplicació proporcionarà un sistema de missatgeria que permetrà que un usuari pugui enviar i rebre missatges d'un altre usuari de l'aplicació.

L'aplicació serà accessible des de navegadors web i s'adaptarà a les principals resolucions de dispositius mòbils, tauletes i ordinadors. Tot i això, la seva implementació ha de considerar la possibilitat de que en un futur sigui accessible mitjançant aplicacions natives dels diferents sistemes operatius d'aquests dispositius mòbils. A més, l'aplicació ha d'estar disponible en català, castellà i anglès i ha d'estar preparada per fer front a un possible augment d'usuaris. En conseqüència, ha de fer ús de tecnologies escalables.

2.2 Actors principals

Els actors principals, també anomenats stakeholders, són persones i entitats que estan interessades en el projecte o que tenen algun impacte sobre aquest. Com veurem a continuació això no només inclou a l'usuari final de l'aplicació, sinó que s'han identificat els següents actors principals:

- **Desenvolupador del projecte:** Aquest projecte forma part d'un treball de fi de grau d'enginyeria informàtica de la Universitat Oberta de Catalunya. Com a autor d'aquest projecte i únic desenvolupador assumeixo totes les funcions pel que fa al disseny, implementació, validació i documentació del projecte. Una de les principals motivacions a l'hora d'haver escollit un projecte d'aquestes característiques ha estat poder posar en pràctica i ampliar els coneixements adquirits al grau i combinar-los amb una de les meves aficions.
- **Personal docent:** El personal docent de l'àrea d'aquest TFG i en especial el professor Antoni Oller Arcas, acompanyarà, guiarà i ajudarà al desenvolupador d'aquest projecte davant els diferents obstacles que puguin sorgir.
- **Usuari final:** Des d'un punt de vista funcional, el principal actor principal són els usuaris finals de l'aplicació. El seu interès sorgeix de la necessitat d'enregistrar digitalment la seva col·lecció musical d'una manera atractiva i fàcilment consultable i/o de la intenció de formar part d'una comunitat on posar en venda, adquirir nous discs i compartir els seus gustos amb altres usuaris.
- **Entitats externes:** Per últim, l'aplicació farà ús de diferents serveis externs per aprofitar-se de fonts de dades externes amb l'objectiu de millorar l'experiència dels usuaris finals. En concret, es faran servir les API de LastFM^[3] i Discogs^[4]. L'aplicació també interactuarà amb l'API de Google^[5] per tal de poder identificar usuaris al sistema mitjançant autenticació OAuth2. A més, es farà servir l'API de cloudinary per que els recursos multimèdia que pugen els usuaris es desin al núvol.

Aquestes organitzacions han exposat diferents serveis web amb l'interès d'estendre les seves respectives comunitats. A més, tindran impacte sobre el projecte i el funcionament de l'aplicació Phonocat.

2.3 Anàlisi de requisits

Un cop descrits els objectius de l'aplicació i els actors principals involucrats, es llisten els requisits funcionals i no funcionals de l'aplicació amb el propòsit d'identificar i documentar les necessitats que els stakeholders tenen sobre el sistema.

2.3.1 Requisits funcionals

Els requisits funcionals ens permeten descriure les funcionalitats que ha de proporcionar l'aplicació. A continuació, s'enumeren mitjançant casos d'ús amb un estil exhaustiu i formal. S'ha decidit no vincular en excés la descripció dels casos d'ús amb la interfície d'usuari per tal de no limitar les opcions d'implementació.

Els casos d'ús s'han agrupat en paquets segons el tipus de funcionalitat que descriuen. Els paquets permeten afegir modularitat a l'aplicació i ens serviran per organitzar els diferents components del sistema. S'han definit quatre packages principals:

- **User:** Agrupa les funcionalitats relacionades amb el registre, autenticació i el perfil d'usuari dins del package User.
- **Album:** Unifica les operacions relacionades amb la col·lecció de discos dels usuaris dins del package Album.
- **Market:** Aquest package conté els casos d'ús relacionats amb la compra/venda de discos entre usuaris de l'aplicació.
- **Social:** Inclou les funcionalitats relacionades amb la interacció social entre usuaris.

El següent diagrama (figura 2.1), mostra el conjunt de casos d'ús agrupats per aquests paquets.

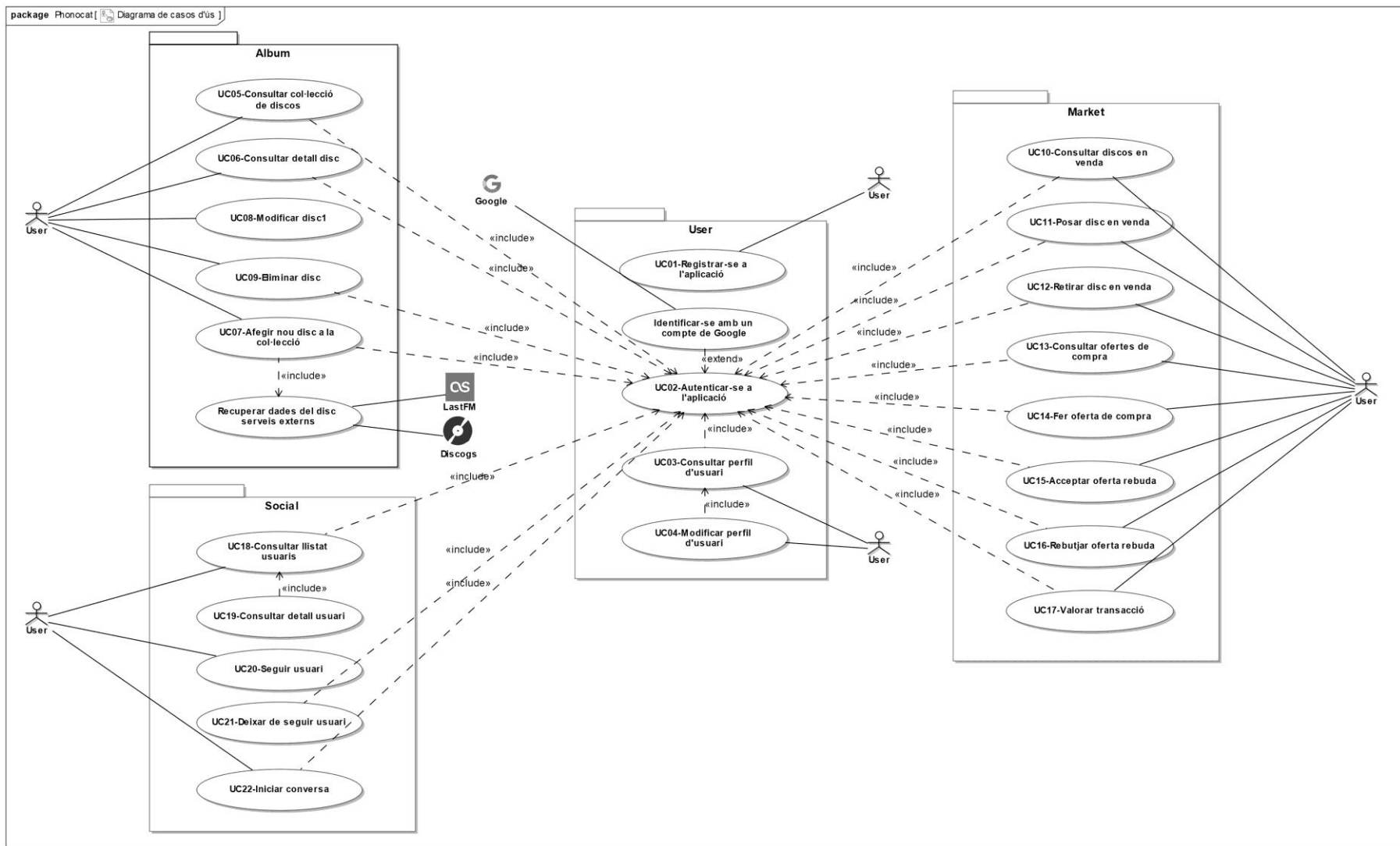


Figura 2.1: Diagrama de casos d'ús

A continuació es llisten els casos d'ús detallats i agrupats per package.

Package User

Cas d'ús	UC01 - Registrar-se a l'aplicació
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol obtenir permís per accedir a l'aplicació
Precondició	-
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari queda enregistrat a l'aplicació i rep un correu electrònic de confirmació
Escenari principal d'èxit	
<ol style="list-style-type: none">1. L'usuari accedeix al formulari de registre de l'aplicació2. Emplena les dades sol·licitades i envia el formulari3. El sistema envia un correu de verificació del compte a l'usuari4. L'usuari confirma el compte mitjançant l'enllaç del correu electrònic5. El sistema envia un correu de confirmació de registre a l'usuari	
Extensions	
-	
Cas d'ús	UC02 - Autenticar-se a l'aplicació
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol accedir a l'aplicació
Precondició	L'usuari ha d'estar enregistrat al sistema (veure cas UC01) o ser propietari d'un compte de Google
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari accedeix a la seva pàgina d'inici de l'aplicació
Escenari principal d'èxit	
<ol style="list-style-type: none">1. L'usuari accedeix al formulari d'inici de sessió de l'aplicació2. L'usuari introdueix el seu correu electrònic i contrasenya enregistrats a l'aplicació3. Accedeix a la seva pàgina d'inici	
Extensions	
<ol style="list-style-type: none">2.a. Les credencials no són vàlides<ol style="list-style-type: none">2.a.1. Es mostra un missatge d'error2.a.2. Es retorna al pas 1 del cas d'ús2.b. L'usuari s'autentica amb les dades d'un compte de Google<ol style="list-style-type: none">2.b.1 L'aplicació sol·licita les dades de contacte obligatòries2.b.2 L'usuari emplena i confirma les dades2.b.3 El cas d'us finalitza amb el pas 3 de l'escenari principal d'èxit2.c L'usuari prem l'enllaç "No recordo la meua contrasenya"<ol style="list-style-type: none">2.c.1 L'aplicació envia un correu electrònic a l'usuari amb un enllaç de restauració de credencials.2.c.2 L'usuari accedeix a l'enllaç i modifica les seves credencials2.c.3 Es retorna al pas 1 del cas d'ús	

Cas d'ús	UC03 - Tancar sessió
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol tancar la seva sessió
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	La sessió de l'usuari es tanca i aquest accedeix a la landing page de l'aplicació
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari selecciona finalitzar la seva sessió 2. El sistema finalitza la sessió i mostra la landing page de l'aplicació 	
Extensions	
-	

Cas d'ús	UC04 - Consultar el perfil d'usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar el seu perfil personal
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra per pantalla les dades personals de l'usuari
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari s'autentica a l'aplicació i accedeix al seu perfil personal 	
Extensions	
-	

Cas d'ús	UC05 - Modificar el perfil d'usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol modificar el seu perfil a l'aplicació
Precondició	L'usuari ha accedit al seu perfil personal (veure cas UC04)
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari rep un missatge de confirmació per pantalla i les dades del seu perfil es persisteixen a base de dades
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari accedeix al seu perfil personal 2. Modifica les dades editables i confirma el formulari 3. L'aplicació mostra una confirmació de l'acció per pantalla 	
Extensions	
-	

Package Album

Cas d'ús	UC06 - Consultar col·lecció de discos
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar la seva col·lecció personal
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	Els discos de l'usuari es mostren per pantalla
Escenari principal d'èxit	
	<ol style="list-style-type: none">1. L'usuari selecciona consultar la seva col·lecció de discos2. El sistema mostra el llistat de discos
Extensions	
	<ol style="list-style-type: none">2.a. L'usuari no té cap disc afegit<ol style="list-style-type: none">2.a.1. Es mostra un missatge d'avertència informant que no hi ha discos2.b. L'usuari filtra la seva col·lecció per diferents atributs com el gènere musical, condició etc.
Cas d'ús	UC07 - Consultar detall disc
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar les dades d'un disc
Precondició	L'usuari accedeix a la seva col·lecció de discos (veure cas UC06)
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra les dades del disc per pantalla
Escenari principal d'èxit	
	<ol style="list-style-type: none">1. L'usuari selecciona un disc de la seva col·lecció de discos2. El sistema mostra el detall del disc
Extensions	
	<ol style="list-style-type: none">2.a. El disc està en venda<ol style="list-style-type: none">2.a.1. Les dades es mostren només en mode de consulta amb l'opció de retirar l'anunci de venda.
Cas d'ús	UC08 - Afegir nou disc a la col·lecció
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol afegir un nou disc a la seva col·lecció
Precondició	L'usuari accedeix a la seva col·lecció de discos (veure cas UC06)
Garanties mínimes	El disc s'afegeix a la col·lecció de l'usuari amb les dades obligatòries
Garanties en cas d'èxit	El disc s'afegeix a la col·lecció de l'usuari amb dades provinents de sistemes externs com les cançons i la seva durada
Escenari principal d'èxit	
	<ol style="list-style-type: none">1. L'usuari accedeix a la seva col·lecció de discos i selecciona l'opció afegir disc2. Emplena les dades del disc3. Opcionalment, l'usuari puja una imatge de la caràtula del seu disc4. L'usuari confirma les dades5. El sistema es connecta amb fonts externes per recuperar les cançons del disc

6. L'aplicació mostra un missatge de confirmació i persisteix les dades

Extensions

- 4.a. L'usuari decideix demanar a l'aplicació una imatge del disc
 - 4.a.1. L'aplicació es connecta amb un servei extern per recuperar una imatge coincident amb les dades del disc.
 - 4.a.2 Es mostra la primera imatge recuperada a l'usuari o cap en cas de no trobar-ne
 - 4.a.2.1 L'usuari confirma o rebutja la proposta
 - 4.a.2.2 es continua amb el pas 5 de l'escenari principal
- 5.a. No es troben dades a les fonts externes
 - 3.a.1. El disc es desa igualment, es continua amb el pas 6 de l'escenari principal

Cas d'ús	UC09 - Modificar disc
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol modificar les dades d'un disc
Precondició	L'usuari consulta el llistat de discos (veure cas UC06) i selecciona editar-ne un
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra confirmació i les noves dades del disc es persisteixen
Escenari principal d'èxit	
1. L'usuari selecciona editar un disc de la seva col·lecció de discos	
2. L'usuari modifica les dades del disc i accepta el formulari	
3. El sistema mostra un missatge de confirmació	
Extensions	
2.a. El disc està en venda	
2.a.1. Les dades es mostren en mode de consulta amb l'opció de retirar l'anunci de venda.	

Cas d'ús	UC10 - Eliminar disc
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol eliminar un disc de la seva col·lecció
Precondició	L'usuari accedeix a la seva col·lecció de discos (veure cas UC06)
Garanties mínimes	-
Garanties en cas d'èxit	El disc s'elimina de la col·lecció
Escenari principal d'èxit	
1. L'usuari selecciona un disc de la seva col·lecció de discos	
2. L'usuari selecciona eliminar el disc	
3. El sistema demana una confirmació a l'usuari	
4. L'usuari confirma l'acció	
5. El disc s'elimina de la col·lecció de l'usuari	
Extensions	
3.a. El disc està en venda	
2.a.1. Es mostra una advertència	

Package Market

Cas d'ús	UC11 - Consultar discos en venda
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari consultar els discos que altres usuaris tenen en venda
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	Es mostra un llistat dels discos anunciats en venda
Escenari principal d'èxit	1. L'usuari s'autentica a l'aplicació i navega a la consulta de discos en venda 2. El sistema mostra els discos en venda d'altres usuaris ordenats per data descendent
Extensions	2.a. L'usuari filtra els resultats per atributs com preu o gènere musical.
Cas d'ús	UC12 - Posar disc en venda
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol anunciar en venda un disc de la seva col·lecció
Precondició	L'usuari consulta un disc de la seva col·lecció (veure cas UC07) que encara no està en venda
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra confirmació i el disc es fa visible a la consulta de discos en venda
Escenari principal d'èxit	1. L'usuari selecciona un disc de la seva col·lecció de discos 2. L'usuari selecciona posar en venda el disc i indica un preu desitjat 3. El sistema mostra un missatge de confirmació i fa visible el disc per altres usuaris a la consulta de discos en venda
Extensions	-
Cas d'ús	UC13 - Retirar disc en venda
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol retirar un disc que ha posat en venda prèviament
Precondició	L'usuari consulta un disc de la seva col·lecció (veure cas UC07) que ha estat publicat en venda El disc no pertany a una transacció finalitzada satisfactòriament
Garanties mínimes	-
Garanties en cas d'èxit	Les possibles ofertes existents sobre el disc queden cancel·lades i el disc s'elimina de la consulta de discos en venda
Escenari principal d'èxit	1. L'usuari selecciona un disc en venda de la seva col·lecció

2. L'usuari selecciona retirar l'anunci de disc en venda
3. El sistema cancel·la totes les ofertes existents sobre aquest disc
3. El sistema mostra un missatge de confirmació i retira el disc de la consulta de discos en venda

Extensions

-

Cas d'ús	UC14 - Consultar ofertes de compra
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar les ofertes que ha rebut pels seus discos en venda o que ha enviat a altres usuaris
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra les ofertes per pantalla
Escenari principal d'èxit	

1. L'usuari accedeix a l'aplicació i navega a la consulta de sol·licituds de compra
2. El sistema mostra les ofertes per pantalla

Extensions

-

Cas d'ús	UC15 - Fer oferta de compra
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol fer una oferta de compra d'un disc en venda
Precondició	L'usuari accedeix a la consulta de discos en venda (veure cas UC11)
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari propietari del disc rep una notificació amb la oferta de compra
Escenari principal d'èxit	

1. L'usuari accedeix a la consulta de discos en venda i selecciona un disc per examinar-ne els detalls
2. L'usuari introdueix un preu d'oferta i envia una sol·licitud de compra
3. El sistema mostra una confirmació i envia una notificació a l'usuari propietari del disc

Extensions

-

Cas d'ús	UC16 - Acceptar oferta rebuda
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol acceptar una oferta de compra rebuda
Precondició	L'usuari accedeix a la consulta d'ofertes (veure cas UC14)
Garanties mínimes	-
Garanties en cas d'èxit	L'oferta queda en estat acceptat, el disc en estat reservat i l'usuari que ha fet la oferta rep una notificació d'acceptació
Escenari principal d'èxit	

1. L'usuari accedeix a la seva consulta d'ofertes i selecciona una oferta rebuda
2. L'usuari confirma la oferta
3. El sistema rebutja automàticament la resta d'ofertes pel mateix disc si n'hi ha
4. El sistema crea una transacció pendent de valoració pels dos usuaris involucrats i els notifica

Extensions

-

Cas d'ús	UC17 - Rebutjar oferta rebuda
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol rebutjar una oferta de compra rebuda
Precondició	L'usuari accedeix a la consulta d'ofertes (veure cas UC14)
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari que ha fet la oferta rep una notificació de rebuig

Escenari principal d'èxit

1. L'usuari selecciona una de les ofertes rebudes
2. L'usuari rebutja la oferta
3. L'usuari que ha fet la oferta rep una notificació de rebuig i la oferta queda cancel·lada.

Extensions

-

Cas d'ús	UC18 - Valorar transacció
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol valorar una transacció en la qual està involucrat
Precondició	L'usuari accedeix a la consulta d'ofertes (veure cas UC14)
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari ha participat a una compra o venda d'un disc

Escenari principal d'èxit

1. L'usuari selecciona una de les ofertes rebudes o enviades en estat acceptat
2. L'usuari valora la transacció
3. El sistema afegeix la valoració a l'altre usuari involucrat

Extensions

- 2.a. La transacció ha estat valorada per l'usuari comprador
 - 2.a.1 El disc s'afegeix a la seva col·lecció

Package Social

Cas d'ús	UC19 - Consultar usuaris
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar altres usuaris de l'aplicació
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)

Garanties mínimes	-
Garanties en cas d'èxit	Es mostra un llistat d'usuaris
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari selecciona la consulta d'usuaris 2. El sistema mostra un llistat d'usuaris de l'aplicació ordenat a l'atzar 	
Extensions	
-	

Cas d'ús	UC20 - Consultar detall usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol consultar les dades d'un altre usuari
Precondició	L'usuari accedeix al llistat d'usuaris (veure cas UC19)
Garanties mínimes	-
Garanties en cas d'èxit	El sistema mostra les dades del disc per pantalla
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari selecciona un usuari del llistat d'usuaris 2. El sistema mostra les dades de l'usuari 	
Extensions	
-	

Cas d'ús	UC21 - Seguir usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol afegir un usuari als seus usuaris favorits
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	Es mostra una confirmació
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari accedeix a la consulta d'usuaris (veure cas UC19) i selecciona veure un 2. El sistema mostra les dades públiques de l'usuari i els seus discos visibles 3. L'usuari selecciona seguir aquest usuari 4. El sistema vincula l'usuari i l'afegeix al llistat d'usuaris favorits 	
Extensions	
-	

Cas d'ús	UC22 - Deixar de seguir usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol afegir un usuari als seus usuaris favorits
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02) i ja segueix a l'usuari (veure cas UC21)
Garanties mínimes	-

Garanties en cas d'èxit	Es mostra una confirmació
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari accedeix a la consulta d'usuaris (veure cas UC19) i selecciona veure un al qual ja segueix 2. El sistema mostra les dades públiques de l'usuari i els seus discos visibles 3. L'usuari selecciona deixar de seguir aquest usuari 4. El sistema elimina el vincle del llistat d'usuaris favorits 	
Extensions	
-	

Cas d'ús	UC21 - Iniciar conversa amb usuari
Actor principal	Usuari de l'aplicació
Àmbit	Aplicació web Phonocat
Stakeholders i interessos	L'usuari vol contactar un usuari amb discos en venda
Precondició	L'usuari s'ha autenticat al sistema (veure cas UC02)
Garanties mínimes	-
Garanties en cas d'èxit	L'usuari destinatari rep el missatge
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari accedeix a la consulta general d'usuaris (veure cas UC19) i selecciona un usuari al qual segueix 2. L'usuari selecciona iniciar una conversa amb aquest usuari 3. El sistema mostra una pantalla de xat 4. L'usuari envia un missatge 5. L'usuari destinatari rep el missatge 	
Extensions	
-	

2.3.2 Requisits no funcionals

Els requisits no funcionals són requisits de producte que, en contrast als requisits funcionals que descriuen el comportament del sistema, ens permeten descriure la qualitat esperada de l'aplicació. S'han llistat els següents requisits a partir de la plantilla d'especificació Volere ^[6].

- Requisits de presentació (Look and Feel)

<u>Codi Volere</u>	<u>Requeriment</u>
10a. Aparença	L'aplicació ha de tenir un estil visual atractiu i entenedor en concordança a altres aplicacions coetànies.
10a. Aparença	A totes les pantalles del sistema i comunicacions per correu electrònic, ha de ser present un com a mínim un element d'identitat visual de l'aplicació, el logo de Phonocat.
10b. Estils	Els estils han de ser consistents entre les diferents vistes.

- Requisits d'usabilitat i humanitat (Usability and Humanity)

Codi Volere

Requeriment

11a. Facilitat d'ús

L'aplicació ha de ser intuïtiva i de fàcil ús per usuaris de més de 13 anys.

11a. Facilitat d'ús

L'aplicació ha d'ajudar a prevenir accions no desitjades amb l'ús de mecanismes com missatges de confirmació.

11b.
Personalització i internacionalització

Els usuaris podran seleccionar la seva preferència d'idioma entre català, castellà i anglès.

11b.
Personalització i internacionalització

Les dates i números s'han de visualitzar en un format concordant amb l'idioma seleccionat

11c. Aprenentatge

L'aplicació s'ha de poder fer servir per usuaris acostumats a les noves tecnologies sense cap tipus de formació addicional.

11d. Comprensió i cortesia

L'aplicació ha de fer ús d'icones i expressions fàcilment identificables pels usuaris potencials de l'aplicació.

11e. Accessibilitat

L'aplicació farà ús d'elements com etiquetes i descripcions dels elements visuals que ajudin als usuaris amb diversitat funcional a fer servir l'aplicació.

- Requisits de rendiment (Performance)

Codi Volere

Requeriment

12g. Escalabilitat o extensibilitat

El sistema ha de fer ús de tecnologies fàcilment escalables davant l'augment d'usuaris

- Requisits operacionals i d'entorn (Operational and Environment)

Codi Volere

Requeriment

13a. Entorn físic esperat

L'aplicació ha de ser usable a dispositius connectats a internet que disposin de navegador web, pantalla i interfície d'entrada, com telèfons intel·ligents, tauletes i ordinadors portàtils o de sobretaula.

- Requisits de manteniment i suport (Maintainability and Support)

Codi Volere

Requeriment

14c. Adaptabilitat

L'aplicació ha de fer ús d'un estil responsiu que s'adapti a diferents resolucions i formats de pantalla.

- Requisits de seguretat (Security)

Codi Volere Requeriment

15a. Accés Només han de poder fer servir les funcionalitats privades del sistema els usuaris autenticats a l'aplicació.

15c. Privacitat Les contrasenyes dels usuaris s'han d'emmagatzemar mitjançant un sistema d'encriptació segur.

- Requisits culturals (Cultural)

Codi Volere Requeriment

16b. Diversitat cultural i inclusió L'aplicació no farà ús d'expressions ofensives.

- Requisits legals (Compliance)

Codi Volere Requeriment

17a. Compliment legal L'aplicació ha de complir les condicions legals dels serveis externs dels quals fa ús.

2.4 Abast

Una vegada s'han definit els objectius i els requisits, es proposa un MVP (Minimum Viable Product)^[7] amb les característiques que s'han considerat primordials. S'han valorat els recursos, termini disponible i riscos identificats a la planificació del projecte.

Amb el propòsit d'identificar quines funcionalitats s'haurien d'incloure al producte del TFG, s'ha creat una matriu de prioritització amb els principals requisits funcionals i no funcionals que afecten la implementació.

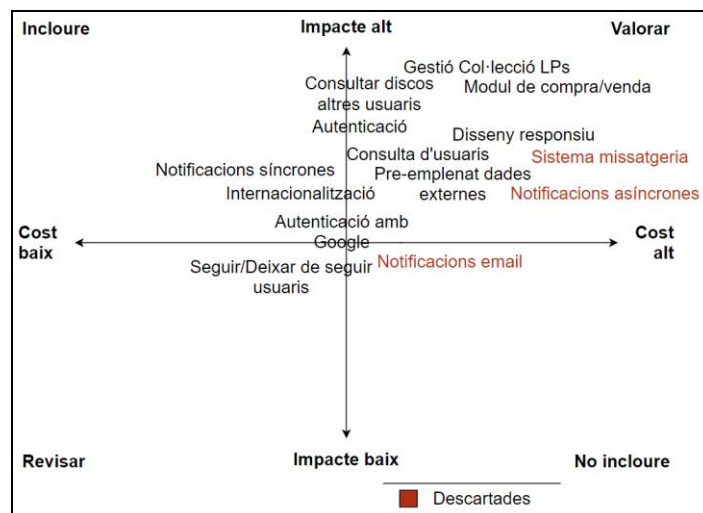


Figura 2.2: Matriu de prioritització de requisits

A la matriu es pot observar que la majoria de funcionalitats identificades tenen un impacte que tendeix a ser alt, però també un cost que tendeix a ser elevat. Aquestes funcionalitats són candidates a ser valorades, ja que en un projecte amb recursos i temps limitat poden implicar endarreriments. D'aquest grup s'ha decidit descartar el sistema de missatgeria instantània i les notificacions asíncrones, ja que implica explorar com treballar amb web-sockets i integrar-ho correctament amb la interfície d'usuari i resta de tecnologies. No obstant això, es considera que tindrien un impacte positiu a l'aplicació, per aquest motiu s'inclouran a l'apartat de futures millores.

Un altre quadrant on caldria revisar quines funcionalitats s'inclouen, és el quadrant inferior esquerra, que conté les funcionalitats amb un impacte més aviat baix, però també un cost que tendeix a la baixa. Aquestes funcionalitats s'han deixat pel final donant prioritat a aquelles que tenen més impacte. Finalment, s'han inclòs al producte gràcies al fet que no han requerit un cost massa elevat.

Per últim, d'una banda es pot observar un quadrant amb funcionalitats amb baix cost i alt impacte, que s'han inclòs directament al producte. De l'altra, el quadrant amb aquelles funcionalitats que tenen un cost alt i un impacte baix, d'aquestes s'ha descartat el sistema de notificacions per correu electrònic.

2.5 Disseny del sistema

Prenent com a referència els requisits descrits a l'apartat anterior, s'ha decidit fer servir una arquitectura heterogènia que combina les següents arquitectures:

- **Arquitectura client servidor:** El producte serà presentat com a aplicació web, la qual cosa afavoreix aquest tipus d'arquitectura.
- **Arquitectura en capes:** L'aplicació es dividirà en capes. La capa de presentació estarà fortament desacoblada de la capa de negoci, que consistirà en un servei web. A més, el model de dades estarà representat a la capa d'integració. Aquesta arquitectura propicia l'escalabilitat i redueix l'acoblament del sistema. Convé ressaltar que la implementació de la capa de presentació serà independent, estarà totalment desacoblada de la capa de negoci i integració i fins i tot es podrà executar a un servidor diferent.
- **Arquitectura SOA:** L'aplicació exposarà una API REST amb el conjunt de serveis necessaris per satisfer els requisits funcionals i que podrà ser consumida per altres aplicacions i/o per futures aplicacions mòbils natives de Phonocat. A més, l'aplicació es comunicarà amb diferents APIs públiques de les quals es nodrirà amb l'objectiu d'obtenir dades externes que aportin un valor afegit. Es tracta, per tant, d'una arquitectura orientada a serveis o SOA.

2.5.1 Punt de vista de la informació

L'objectiu del punt de vista de la informació és el de descriure en detall la informació amb la qual treballarà el sistema i el tipus de dades que farà servir.

Esquema invariant

L'esquema invariant de classes UML ens permet representar les estructures de dades amb les quals treballa l'aplicació, algunes de les seves restriccions i com es relacionen entre si.

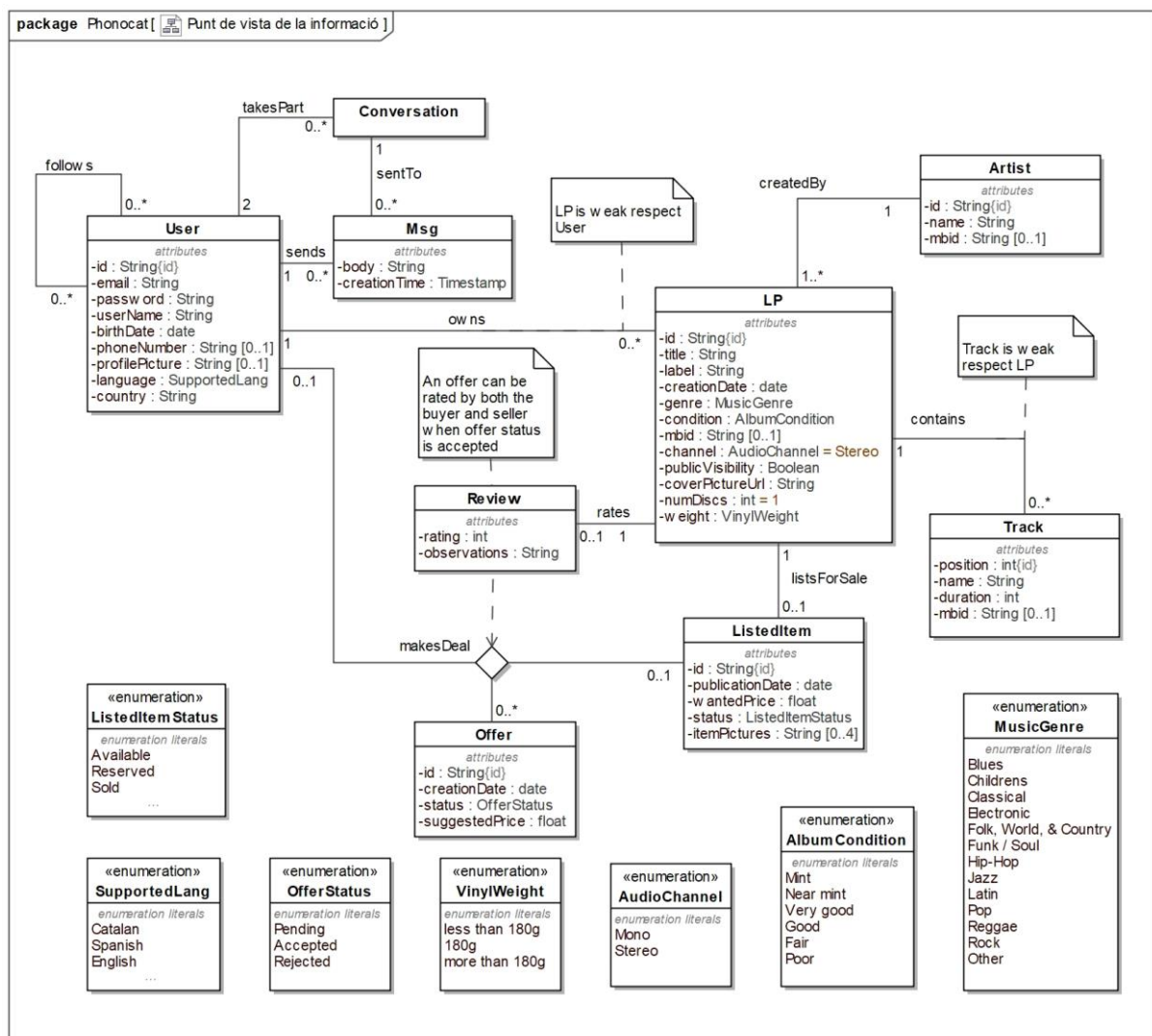


Figura 2.3: Entitats model de dades

Descripció de les entitats i les seves relacions

- **User**: Representa l'usuari identificat al sistema. Conté les seves dades personals i d'identificació. Un usuari pot ser propietari d'un o més discos (LP) de la seva col·lecció i pot tenir vinculades múltiples ofertes (Offer) i participar en n converses.

- **LP:** Representa un disc de la col·lecció d'un usuari. Conté entre d'altres, diverses dades pròpies del disc com el seu títol, autor, portada, segell discogràfic etc. També conté altres dades que detallen el seu estat com la condició física o imatges del disc. Un disc ha estat creat per un autor principal (Artist) i pot contenir el detall de les diferents cançons que el componen (Track). A més, un LP pot estar o no llistat com a disc en venda (ListedItem). Un LP conté un atribut mbid que serveix d'identificador per a alguns dels serveis externs dels quals en farà ús l'aplicació.
- **MainArtist:** Conté el nom i un mbid de l'autor d'un o més discs del sistema.
- **Track:** Conté diversos camps com la posició, nom i duració d'un LP del sistema. Com a LP i MainArtist, s'emmagatzema l'identificador mbid per connectar amb serveis externs.
- **ListedItem:** Representa un disc posat en venda amb el preu desitjat pel comprador. Un disc en venda pot rebre múltiples ofertes (Offer).
- **Offer:** L'entitat Offer, representa una oferta feta per un usuari sobre un disc en venda. Conté un import proposat de compra i un estat. Un usuari pot fer múltiples ofertes, però una oferta només pot estar vinculada a un disc en venda(ListedItem).
- **Msg:** Aquesta entitat conté el cos i l'instant de temps d'un missatge enviat per un usuari.
- **SupportedLand:** Enumerador amb els llenguatges suportats per l'aplicació.
- **ListedItemStatus:** Enumerador amb els possibles estats d'un disc en venda (Disponible si no te cap oferta confirmada, Reservat si te una oferta acceptada sense valorar i Venut si la transacció ja s'ha valorat pels usuaris involucrats).
- **OfferStatus:** Enumerador amb els estats pels que pot passar una oferta.
- **VinylWeight:** Enumerador amb els diferents formats de qualitat de premsa d'un vinyl pel que fa al seu pes.
- **AlbumCondition:** Enumerador amb la qualitat de conservació del disc.
- **MusicGenre:** Enumerador amb els diferents gèneres de música amb els que es pot catalogar un LP.

2.5.2 Punt de vista de la computació

L'objectiu del punt de vista de la computació és el de descriure com s'organitzen i descomponen els diferents components arquitectònics del sistema. El següent diagrama correspon a una arquitectura encara independent de les tecnologies que es faran servir. Com a representació global del sistema es mostren els components de les diferents capes amb les seves interfícies col·lapsades.

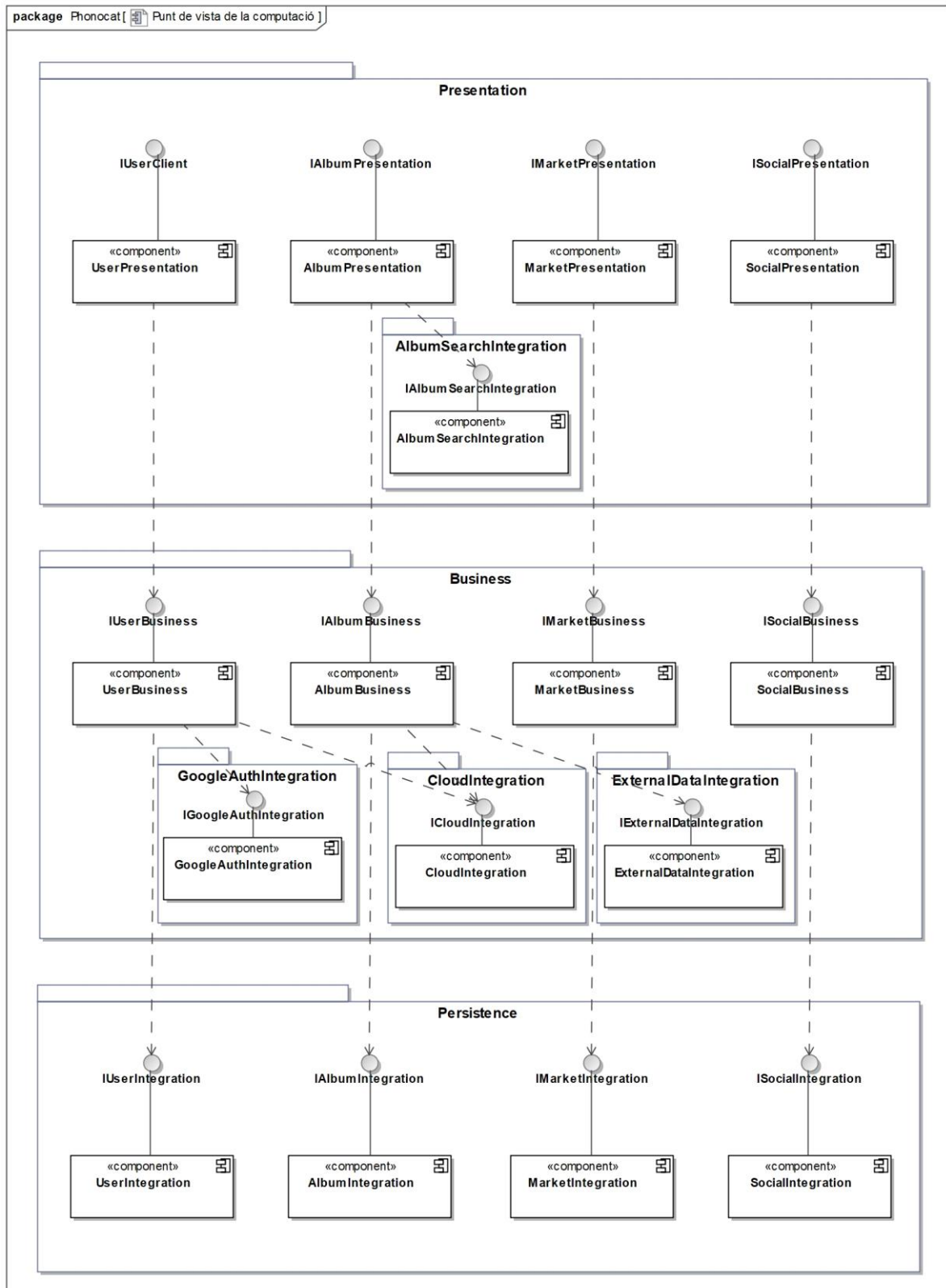


Figura 2.4: Punt de vista de la computació

Els components de la capa de presentació es comunicaran amb els serveis que exposa la capa de negoci. En conseqüència, les interfícies dels components de les dues capes contenen els mètodes de l'API REST, que es pot consultar el detall a l'Annex I. Especificació API REST Phonocat.

Per veure com es comporten computacionalment alguns dels components, es mostra com a exemple el diagrama de seqüència de la inserció d'un nou LP a la col·lecció d'un usuari. S'ha escollit aquesta funcionalitat perquè s'executa en diferents passos i per dur-se a terme requereix la participació de múltiples components. El flux d'interacció és el següent:

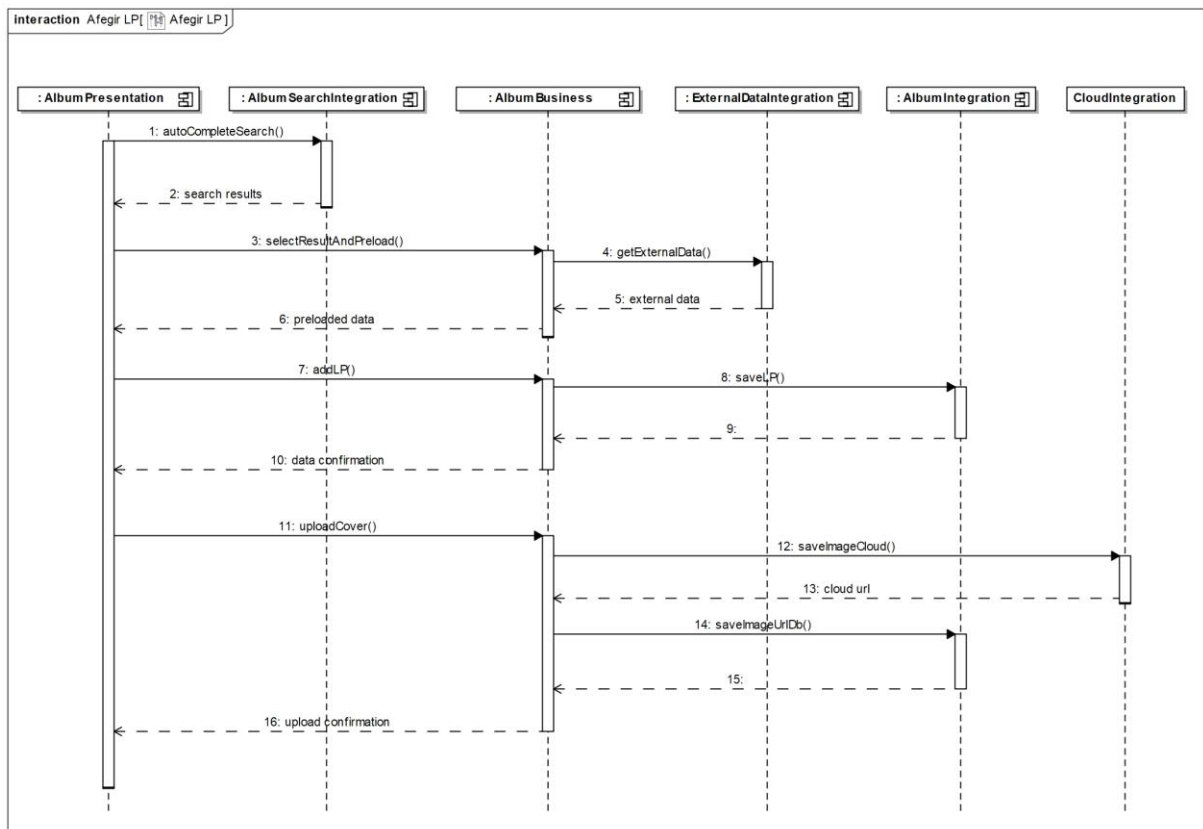


Figura 2.5: Diagrama de seqüència - Afegir LP

L'usuari introdueix un terme de cerca del LP que vol introduir al sistema. La capa de presentació es comunica directament amb un servei de cerca de termes. Aquest pas es fa directament a la presentació i no al negoci, ja que es tracta d'una cerca amb autocompletat de dades, aquest servei és públic i no s'exposen dades privades i d'aquesta manera es redueix el temps de resposta, la qual cosa proporciona una experiència d'usuari favorable. A continuació l'usuari selecciona un dels resultats de cerca i crida a un servei de l'aplicació per pre-emplenar dades del disc. En aquest cas sí que es fa a través de la capa de negoci, ja que es fan diferents operacions al servidor per pre-emplenar les dades i en aquest cas el temps de resposta no és tan crític com abans. De nou a la capa de presentació l'usuari pot modificar aquestes

dades suggerides o afegir-ne de noves i desar l'àlbum. Per a aquest pas es torna a comunicar amb la capa de negoci que desa les dades mitjançant la capa de presentació i retorna una confirmació a l'usuari. En darrer lloc, l'usuari pot pujar una portada del disc, per fer-ho fa una petició a la capa de negoci, que es comunica amb els serveis d'integració del núvol per emmagatzemar la imatge i posteriorment desa la url de la imatge del núvol a base de dades.

2.6 Prototipatge UI

De nou prenent com a referència la planificació inicial i els recursos disponibles, s'han elaborat diversos prototips de baixa fidelitat^[8] que serviran com a guia per implementar la interfície d'usuari de l'aplicació.

Els prototips de baixa fidelitat són útils com a representació esquemàtica dels elements generals del sistema sense arribar al detall de la interfície. S'han prototipat les següents vistes que, si bé no reflecteixen tota la interfície de l'aplicació, són de gran utilitat de cara a visualitzar globalment com serà el producte i com s'han de distribuir els diferents components gràfics a la fase d'implementació.

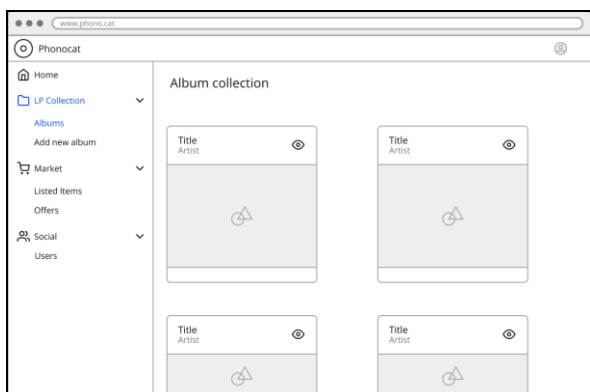


Figura 2.6: Prototip lo-fi Col·lecció LP

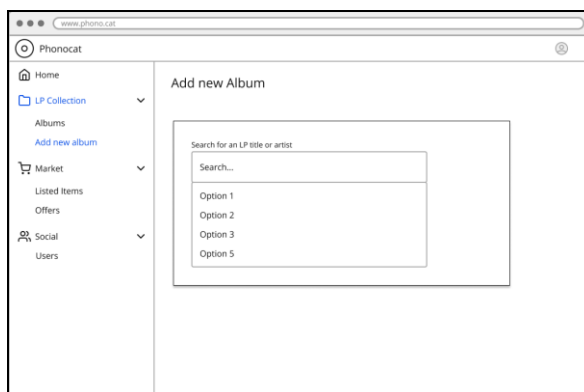


Figura 2.7: Prototip lo-fi afegir LP pas 1

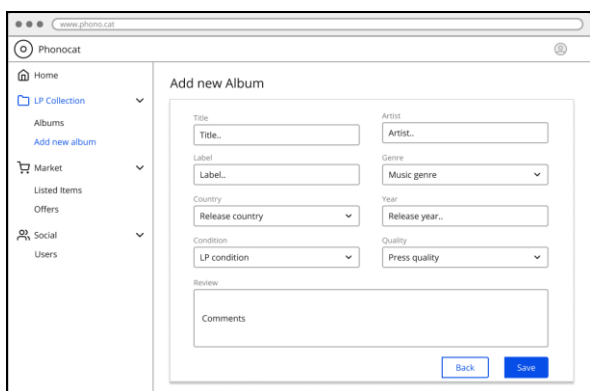


Figura 2.8: Protitip lo-fi afegir LP pas 2

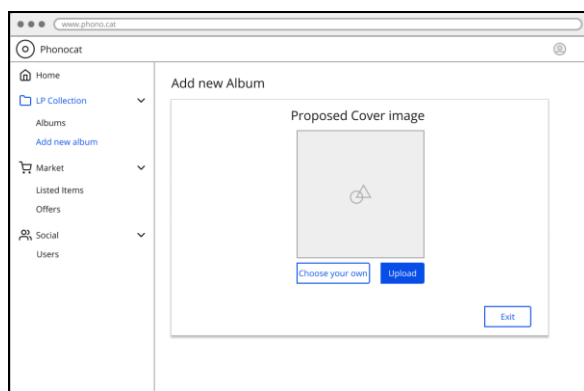


Figura 2.9: Prototip lo-fi afegir LP pas 3

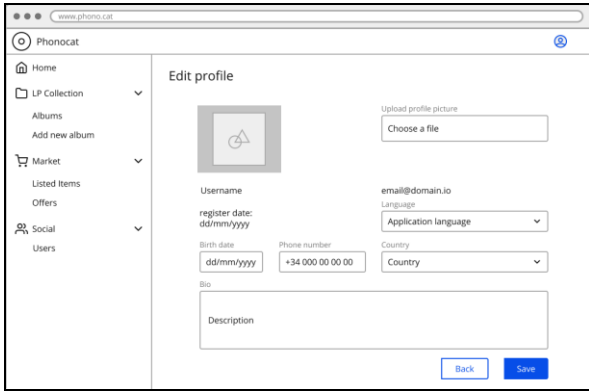


Figura 2.10: Prototip lo-fi perfil d'usuari

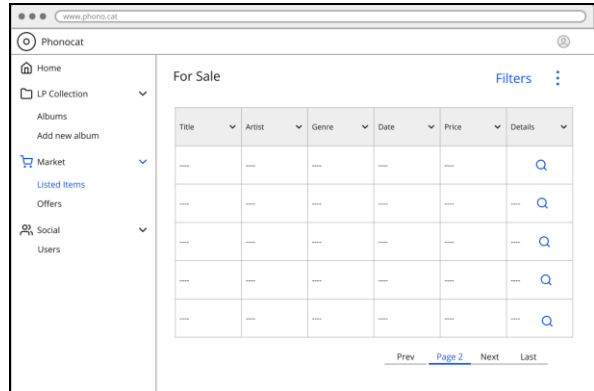


Figura 2.11: Prototip lo-fi LPs en venda

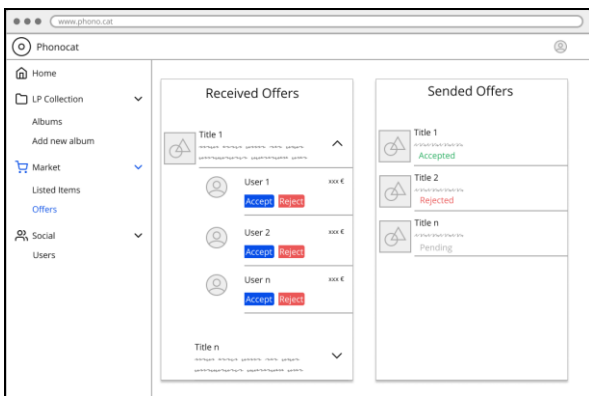


Figura 2.12: Prototip lo-fi Ofertes

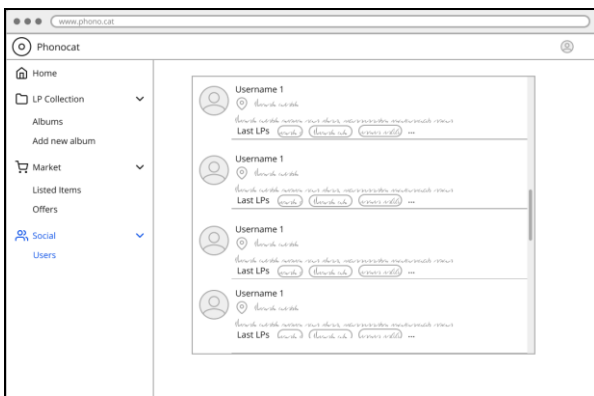


Figura 2.13: Prototip lo-fi llistat d'usuaris

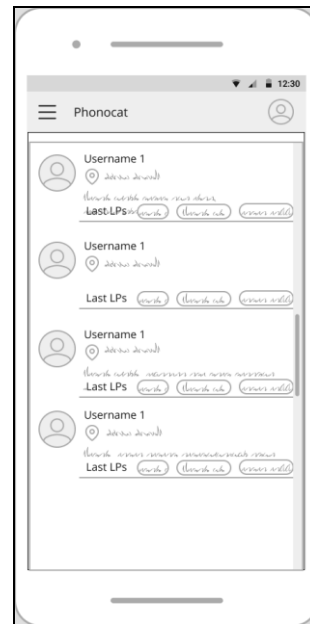
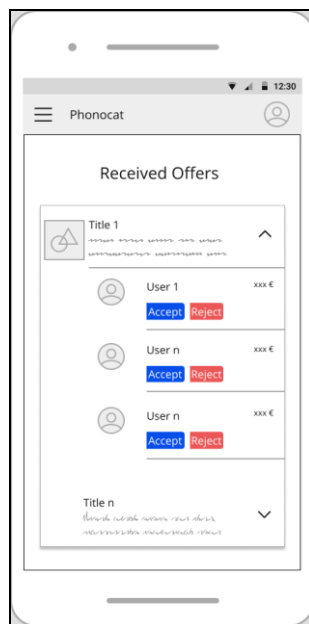
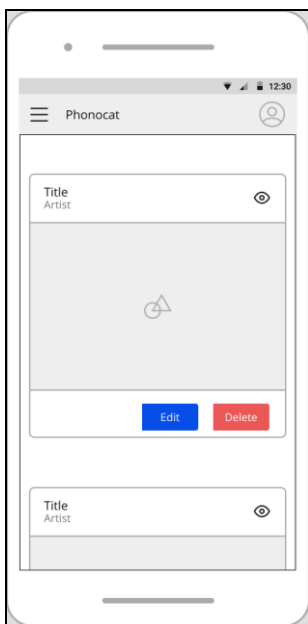


Figura 2.14: Prototip lo-fi vistes dispositiu mòbil

3 Implementació

3.1 Background tecnològic

Abans de descriure com s'ha implementat el projecte, s'ha considerat convenient dedicar aquest subapartat a descriure el conjunt d'eines que s'han fet servir, perquè s'han escollit en pro d'altres solucions i com s'integren en el conjunt de l'arquitectura.

Tot i que els coneixements acadèmics i professionals de l'autor del treball se centren principalment en programació orientada a objectes i bases de dades relacionals, un dels objectius principals del treball era adquirir noves competències en algunes de les tecnologies web més populars en l'actualitat. Tanmateix, el projecte s'ha implementat per una sola persona, per aquest fet s'ha dut a terme recerca sobre tecnologies que afavoreixin un perfil full-stack^[9], on el desenvolupador s'encarrega d'implementar tant el programari del back-end o servidor, com el front-end o client.

Com ha resultat d'aquesta investigació s'ha decidit fer servir eines basades en el llenguatge de programació JavaScript, ja que actualment existeixen multitud de solucions tant per desenvolupar el client com el servidor que es basen en aquesta tecnologia. D'aquesta manera tot i que cada eina conté les seves característiques pròpies, el llenguatge de programació en el que es basen és comú a tota l'aplicació.

Un altre tret significatiu que es pot extreure del disseny de l'aplicació, és que la capa de negoci s'exposarà mitjançant una API REST^[10]. Amb aquest condicionant present s'ha decidit fer servir Node.js, pel fet que té una arquitectura basada en esdeveniments que permet executar el codi de manera asíncrona. A més, Node es basa en JavaScript i proporciona un entorn lleuger amb alta escalabilitat que permet donar servei davant un nombre elevat d'accessos concurrents, aquest fet també compleix un dels objectius de l'aplicació, que és la capacitat d'adaptar-se davant un augment d'usuaris.

Amb el propòsit de simplificar el desenvolupament i fer ús de bones pràctiques pel que fa a l'organització del codi del servidor, s'ha fet servir Express, un dels frameworks més populars de Node.js. De manera paral·lela, express ens permet fer servir middlewares^[12], els middlewares son funcionalitats que tenen accés a la petició i a la resposta de les crides HTTP i ens permeten integrar d'una manera fàcil utilitats per controlar, entre d'altres, l'autenticació o la gestió d'errors.

Com a base de dades, s'ha decidit fer servir una base de dades NoSQL^[11] que s'adapti fàcilment a les eines abans descrites i que treballi amb documents JSON, que és el format principal mitjançant el qual s'intercanviaran les dades entre el client i el servidor. Això permet una comunicació lleugera i amb un format proper a JavaScript que ens facilitarà la lectura i escriptura d'aquestes dades a les diferents capes de l'aplicació. Amb aquestes característiques presents, s'ha decidit fer servir MongoDB, que a més de satisfer aquests requisits, s'integra fàcilment al núvol i de manera gratuïta.

Per desenvolupar el front-end, es van identificar dues solucions, Angular i React. Tot i que conceptualment són eines diferents; Angular és un framework complet de desenvolupament i React una llibreria feta servir per desenvolupar interfícies d'usuari; les dues solucions s'adapten adientment a la resta de tecnologies que es faran servir i ens permeten desenvolupar la interfície d'usuari de l'aplicació. Cadascuna té, però, els seus inconvenients i avantatges. Per decidir quina fer servir al projecte, es va realitzar una comparació punt per punt amb alguns dels aspectes més rellevants a l'hora de fer-ne servir una o altra.

	Angular	React	Veredict
Popularitat	Angular és una eina de desenvolupament web amb una popularitat elevada i una comunitat ampla.	React ha guanyat popularitat especialment al darrer dels últims anys i es posiciona com a una de les solucions referents a l'actualitat	Segons l'enquesta a desenvolupadors del sector del portal StackOverflow l'any 2020 ^[12] , React guanya en popularitat a Angular amb un 35,9% d'ús enfront a un 25,1% per part d'Angular. Una popularitat superior significa més recursos i exemples disponibles, fet que pot facilitar el desenvolupament del projecte.
Corva d'aprenentatge	Angular té una corva d'aprenentatge elevada, en tant que es tracta d'un framework complet.	React és fàcil de començar a fer servir, no obstant això la corva es fa més inclinada segons les llibreries amb les quals l'integrem.	React te com a avantatge que és més directe començar a desenvolupar. Per si sol no satisfà totes les necessitats d'una aplicació amb certa complexitat, per tant cal integrar-ho amb altres eines. Angular, d'altra banda, requereix una preparació inicial superior, però conté per si mateixa solucions per la majoria de necessitats d'una aplicació web. Amb tot, ambdues eines estan equilibrades en aquest sentit.
Flexibilitat	Sistema més rígid però amb major consistència del codi.	Sistema altament flexible, permet integrar-se amb multitud d'eines d'una manera robusta. Permet definir una estructura del codi lliure.	Pel que fa a la flexibilitat, React permet adaptar-se amb components externs amb molta facilitat. Per a aquest projecte, s'ha valorat positivament la capacitat d'adaptació amb components externs que facilitin el desenvolupament.
Projecció laboral	Oferta laboral actual consolidada amb salaris atractius	Oferta laboral amb tendència creixent amb salaris atractius	Un dels objectius del treball és adquirir noves competències, s'ha valorat que aquestes estiguin demandades. La creixent popularitat de React fa que el mercat tendeixi a

			demanar més professionals especialitzats en aquesta llibreria. La mitja salarial dels desenvolupadors d'Angular i React és equilibrada amb un lleuger avantatge per a React ³ .
--	--	--	--

Taula 3.1: Comparativa Angular React

Com a resultat d'aquesta comparativa, s'ha decidit fer servir ReactJS. A més, aquesta llibreria s'integra nativament amb Material-UI, un conjunt de components basats en el disseny material^[2] que permeten desenvolupar interfícies d'usuari reactives. La majoria de components gràfics s'han pogut desenvolupar exclusivament amb l'ús d'aquesta llibreria, que en alguns casos s'ha combinat amb els llenguatges HTML i CSS.

React per si sol presentava algunes limitacions en la gestió de l'estat, ja que requeria que aquest viatgés entre components mitjançant la transferència de propietats o funcions de callback, això va canviar amb l'aparició de React Context que permet mantenir un estat global de l'aplicació. No obstant això, per tal de mantenir l'estat global s'ha fet servir la llibreria Redux, que afavoreix fer ús de bones pràctiques a l'hora de treballar amb l'estat i a més conté eines que permeten depurar l'aplicació. Tanmateix, per gestionar la navegació de l'aplicació s'ha fet ús de la llibreria React Router. Aquestes llibreries es descriuran en detall a l'apartat d'implementació del front-end.

Un altre aspecte tecnològic sobre el qual es va fer recerca, és la possibilitat de desplegar l'aplicació al núvol. S'ha decidit fer servir el servidor d'aplicacions Heroku per la seva facilitat d'integració amb la resta de tecnologies i pel seu cost. A diferència d'altres plataformes de cloud computing, que requereixen informar les dades de pagament i comencen a cobrar quan s'excedeix un volum de peticions, Heroku ofereix una capa totalment gratuïta. Un altre aspecte rellevant és el fet que Heroku permet desplegar l'aplicació mitjançant l'eina de control de versions Git mitjançant un CLI^[13]. Aquest pas simplifica significativament les tasques d'integració.

Per contra, un dels punts febles que es van identificar amb aquesta solució, és que Heroku no permet l'escriptura de documents al sistema de fitxers del servidor, o més acuradament, els seus contenidors d'aplicacions tenen un sistema de fitxers efímer, que s'esborra periòdicament. Diverses funcionalitats de l'aplicació requereixen que els usuaris puguin pujar imatges, per exemple del seu perfil, dels seus discos o d'un anunci de venda. Per evitar aquest obstacle, es van identificar dues possibles solucions.

³ What's the Average JavaScript Developer Salary in 2021?. DAXX [Comparativa salaris] <https://www.daxx.com/blog/development-trends/javascript-market-trends-average-salaries>

La primera solució consisteix a desar les imatges com a binari a la base de dades, com a avantatge es pot implementar sense la necessitat de cap eina extra, per contra augmenta la càrrega de treball a les consultes de base de dades i el volum de dades a la transferència d'informació, entre el servidor i el client, ja que les imatges han de viatjar codificades en format base64. La segona solució i la que finalment s'ha fet servir és la integració amb una plataforma d'emmagatzematge al núvol. Aquesta solució requereix integrar l'arquitectura amb un nou servei extern, però a canvi allibera la base de dades de tot el volum referent a contingut multimèdia i ens permet fer viatjar aquests recursos indicant tan sols una URL. De les diferents opcions que s'han valorat, el cost ha estat de nou un aspecte determinant. S'ha escollit Cloudinary que permet emmagatzemar contingut multimèdia al núvol de manera gratuïta i s'integra amb Node.js fàcilment ^[14].

En darrer lloc, s'han fet servir multitud de llibreries i components instal·lats mitjançant el repositori npm que es descriuran als apartats d'implementació del front-end i el back-end.

Recapitulant, es farà ús dels següents llenguatges de programació, frameworks, llibreries i solucions tecnològiques:

- **JavaScript:** Llenguatge de programació lleuger que tradicionalment es feia servir principalment a la part client de les aplicacions web, però que en l'actualitat també s'usa per desenvolupar el costat servidor. La seva revisió ES6 permet l'ús de funcionalitats com desestructuració (deconstructing), promeses (promises) i funcions fletxa (arrow functions) de les quals s'ha fet un ús extensiu per a implementar aquest projecte.
- **JSON:** Format lleuger per emmagatzemar i transportar dades, el seu nom s'obté de les sigles de **JavaScript Object Notation**.
- **Node.js**^[15]: Entorn en temps d'execució asíncron per a la capa de servidor basat en Javascript que té una arquitectura basada en esdeveniments. En general es tracta d'un entorn lleuger que permet construir aplicacions escalables amb accessos concurrents.
- **Express**^[16]: Framework que proporciona un conjunt de característiques per organitzar i construir aplicacions web amb Node.js. Està especialment dissenyat per desenvolupar aplicacions web sota el patró MVC i APIs.
- **MongoDb**^[17]: Base de dades que treballa amb documents en format JSON. L'ODM Mongoose, proveeix d'una solució directa, basada en esquemes per modelar les dades a una aplicació Node.js.
- **React**^[18]: Llibreria creada per Facebook desenvolupada sobre JavaScript que permet desenvolupar interfícies d'usuari reactives d'una forma modular a partir de components. Cada component està encapsulat i per si mateix conté tant elements de presentació gràfica com programació de funcionalitats

- **Material UI**^[19]: Llibreria de components gràfics per React basats en Material Design.
- **HTML**: Prové de les sigles **H**yper**T**ext **M**arkup **L**anguage, és un dels formats més populars per a la construcció de documents web.
- **CSS**: Prové del nom en àngles **C**ascading **S**tyle **S**heets,
- **Npm**: Repositori de programari JavaScript open-source amb més de 800.000 paquets de codi. Es tracta del sistema de gestió de paquets per defecte de Node.js.
- **Heroku**^[20]: Plataforma com a servei (PaaS) al núvol que permet compilar, administrar i executar aplicacions.
- **Cloudinary**: Servei d'emmagatzematge al núvol de recursos multimèdia.
- **MongoDb Atlas**: servei al núvol que ofereix una base de dades MongoDB.
- **Git**: Eina de control de versions amb una àmplia comunitat que s'ha convertit en un estàndard de facto.

Aquests tipus de col·leccions d'eines o stacks, sovint s'identifiquen amb acrònims generats amb les inicials de les tecnologies emprades. En el cas d'aquest projecte podem parlar d'un stack MERN que prové de les eines **M**ongo**D**b, **E**xpress, **R**ect i **N**ode. La següent imatge mostra una representació esquemàtica d'aquesta arquitectura:

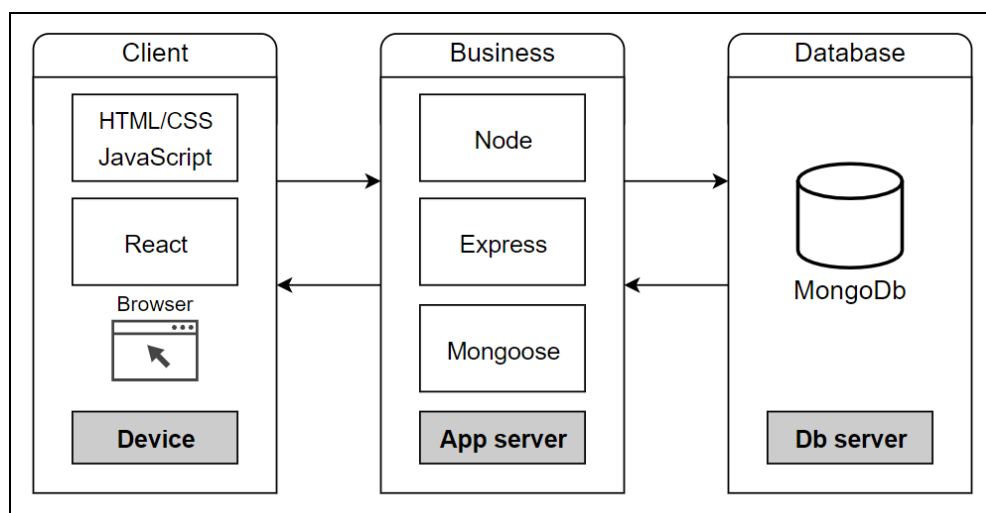


Figura 3.1: Arquitectura tecnològica de l'aplicació

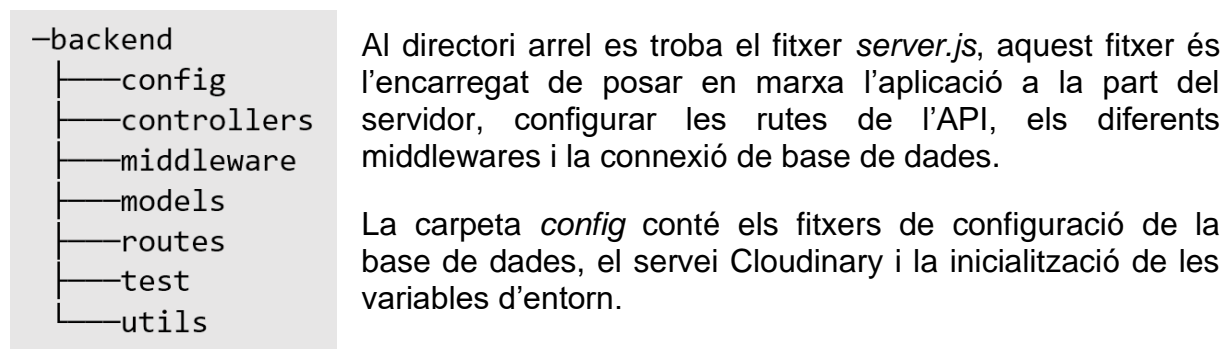
3.2 Desenvolupament del backend

A aquest apartat, es descriu com s'ha organitzat el codi del servidor, quina és la funció dels diferents components, així com les utilitats que s'han fet servir i quins són alguns dels aspectes més rellevants de la implementació.

3.2.1 Organització del codi font del servidor

A causa de la flexibilitat d'aquest entorn de desenvolupament, no existeix un estàndard alhora d'estructurar el codi. No obstant això, s'han seguit algunes de les pràctiques i convencions més freqüents per tal d'organitzar el codi font^[21]. El codi del backend s'organitza en els següents directoris:

Figura 3.2: Organització codi font backend



La carpeta *routes* conté un fitxer per cadascun dels paquets definits a la fase de disseny on es capturen les URL dels endpoints de l'API. Cada captura d'una ruta del servei web està associada a un mètode d'un controlador dins del directori *controllers* i pot ser processat per un o varis middlewares. Aquests fitxers contenen lògica de negoci dels diferents endpoints de l'API i també existeix un per paquet.

Pel que fa al directori *middleware*, dins s'inclou el codi dels middlewares d'autenticació i autorització, control d'errors i pujada de fitxers.

El directori *models* conté els esquemes de mongoose que defineixen les entitats del model de dades de l'aplicació.

La carpeta *test* conté scripts destinats a provar l'aplicació. En concret, s'inclou un script que permet generar usuaris de prova amb dades de perfil aleatòries mitjançant la llibreria *faker*⁴.

En darrer lloc, el directori *utils* conté diverses utilitats que es fan servir a diversos llocs de l'aplicació, com la generació de tokens JWT, la normalització de les imatges que es puguen al núvol o la traducció de dades entre el format dels serveis externs i el de la pròpia aplicació.

⁴ Faker, llibreria de generació de dades massives
<https://github.com/Marak/Faker.js#readme>

3.2.2 Implementació de l'API REST

El framework Express, entre les seves funcionalitats, conté la classe Router, que permet encaminar les peticions HTTP d'una manera modular i permet l'ús de middlewares. Com a exemple general de com s'implementen els mètodes de l'API es mostren els següents fragments de codi de l'aplicació. El fitxer `phonocat/backend/routers/lp-routes.js`, mostra a tall d'exemple, com es capturen les peticions dins l'aplicació:

```
1 import express from 'express'
2
3 ...
4 const router = express.Router()
5 ...
6 router.get('/users/:username/lps', passport.authenticate("jwt", {
7   session: false }), getLPs)
8 ...
```

Fragment de codi 3.1: endpoint routing

Aquest fragment de codi correspon a la captura de les peticions a la ruta `/users/:username/lps`. L'etiqueta `:username` fa referència al nom de l'usuari pel qual es volen recuperar els discos. La secció de codi de la línia 6 que fa referència a `passport` és l'execució del middleware d'autenticació. Per últim, és crida com a middleware al mètode `getLPs`. Aquest mètode forma part del controlador `phonocat/backend/controllers/lp-controllers.js`, que conté la lògica de negoci d'aquest servei:

```
1 const getLPs = asyncHandler(async (req, res) => {
2   const paramUser = await User.findOne({ username:
3   req.params.username })
4   if (paramUser) {
5     ...
6     res.send(lpsSummary)
7   } else {
8     res.status(404)
9     ...
10  }
11 }
```

Fragment de codi 3.2: Exemple controlador

Com es pot observar a la línia 3, el nom de l'usuari es recupera a l'objecte request del mètode. Aquest patró ens permet desacoblar la programació del llistat de URL i s'ha seguit per a la resta de mètodes de l'API.

3.2.3 Middlewares

Al llarg d'aquest apartat s'han fet diverses referències a uns components de la implementació anomenats middlewares. A Express, els middlewares tenen accés a la petició i la resposta (`request`, `response`), i a la següent funció del cicle de petició/resposta de l'aplicació.

Express inclou la possibilitat de fer servir middlewares a nivell d'aplicació, de direccionalment o de gestió d'errors. Això és de gran utilitat per tal d'incloure funcions dins del flux d'una petició.

Posem com a exemple l'operació per pujar la portada d'un disc, que fa ús de la llibreria passport com a middleware per controlar l'autorització de l'usuari i de la llibreria multer com a middleware per a controlar el format i restriccions del fitxer que s'envia a la petició. En primer lloc, el middleware d'autenticació s'encarrega de validar que la petició conté un token d'autorització vàlid i no exhaurit. A continuació, el middleware de control de fitxers valida diverses restriccions del fitxer, que sigui de tipus imatge, i d'una mida inferior a 10MB.

```
router.post('/lps/:id/cover/', passport.authenticate("jwt", {
  session: false }),
  multerUploadSingleImage('cover'), uploadLPCover)
```

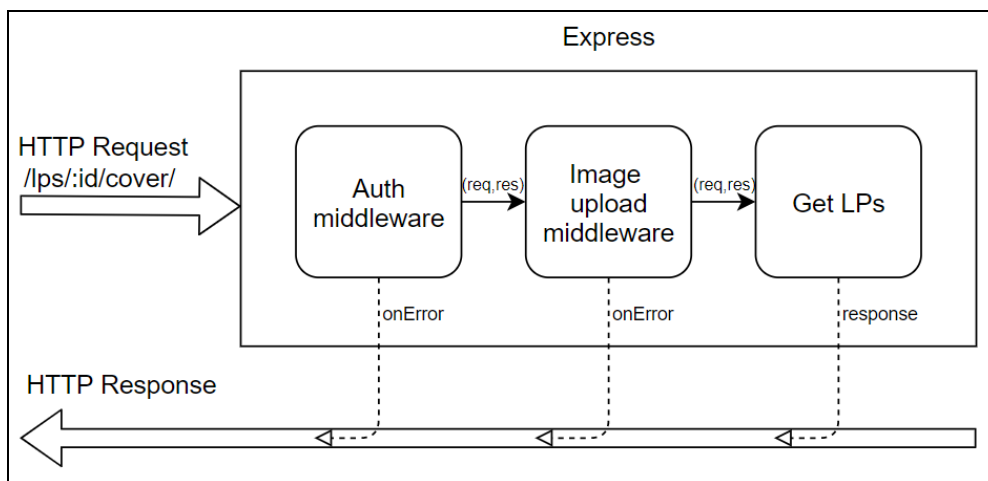


Figura 3.3: Execució middlewares pujar portada LP

Aquest flux ens permet observar com es comporten les peticions al servidor fins que es retorna la resposta.

3.2.4 Autenticació, autorització i seguretat

Amb fi de controlar que les peticions s'executin per usuaris autenticats al sistema a aquells mètodes de l'API que això ho requereixen, s'ha investigat quin mètode fer servir que ens permeti identificar d'una forma segura quin usuari està fent les peticions i si s'ha identificat correctament. Els serveis de Phonocat no emmagatzemen l'estat, en conseqüència cal fer servir un mètode on el consumidor proporcioni les seves dades a cada petició d'una manera segura.

Un format que ens permet transmetre aquest tipus d'informació són els JSON Web Tokens o JWT. JWT és un estàndard que permet transmetre segurament informació, compacta i autocontenida en format JSON.

L'API de Phonocat fa ús d'aquests tokens d'accés per controlar l'autorització dels usuaris de la següent manera:

Quan l'usuari s'autentica amb les seves credencials el servidor genera un token JWT amb una validesa de quinze dies. Aquest token es genera mitjançant la llibreria a partir de l'id de l'usuari i una clau privada que només coneix el servidor de l'aplicació.

```
1 import jwt from 'jsonwebtoken'
2 ...
3
4 /**
5  * Creates a JWT token based on user id
6  * @param {*} id user Id
7  * @returns JWT token
8  */
9 export const createToken = (id) => {
10   return jwt.sign({ id }, process.env.JWT_SECRET, {
11     expiresIn: process.env.JWT_EXPIRATION_TIME,
12   })
13 }
```

Fragment de codi 3.3: Generació JWT

El fragment de codi 3.3 conté la funció de generació de JWT^[22]. Aquest mètode retorna una cadena de text del següent tipus:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyYWM2ZjI4NjY2YWM4MDAwNGEzYWYwZiIsIm1hdCI6MTYyMjkxOTk3OSwiZXhwIjoxNjE0MjE1OTc5fQ.1XI6f891AhmOTaK60xs3SfA8KeSvqR4gY1se  
tCdjrUI
```

Aquest token intel·ligible a primera vista conté tres elements que ens permeten validar la identitat de l'usuari que fa la petició i la validesa del seu accés:

- Una capçalera que indica el format i algoritme d'encriptació.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```
- Un cos anomenat payload amb l'id de l'usuari i les dates d'expedició i exhauriment del token.

```
{
  "id": "60ac6f28666af80004a3af0f",
  "iat": 1622919979,
  "exp": 1624215979
}
```
- La signatura feta amb la clau secreta.

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  SECRET_KEY
```

)

Un altre aspecte referent a la seguretat de la implementació del backend, té a veure amb l'emmagatzematge dels passwords. S'ha fet servir un hook de mongoose per encriptar el password abans de desar-lo a base de dades cada vegada que s'insereix un nou usuari. Els hooks a mongoose funcionen com a middleware i són funcions que s'executen abans o després del mètode al qual fan referència.

```
1 userSchema.pre('save', async function (next) {
2   const user = this
3   if (user.isModified('password')) {
4     const salt = await bcrypt.genSalt(10)
5     user.password = await bcrypt.hash(this.password, salt)
6   }
7   next()
8 })
```

Fragment de codi 3.4: Pre-hook mongoose, encriptar password

3.2.5 Llistat d'utilitats backend

Les següents llibreries s'han fet servir per a resoldre o facilitar les següents funcionalitats del backend:

- *axios*: client per fer crides a l'API.
- *bcryptjs*: encriptació els passwords.
- *cloudinary*: integració amb l'API de Cloudinary.
- *country-list*: Llistat de països en format ISO 3166-1 alpha-2.
- *dotenv*: variables d'entorn.
- *express*: framework de node.js.
- *express-async-handler*: gestió dels errors als controladors.
- *express-validator*: validació de dades.
- *google-auth-library*: autenticació amb Google.
- *jsonwebtoken*: Creació de JWT.
- *mongoose*: ODM per MongoDB.
- *mongoose-aggregate-paginate-v2*: utilitat per paginar i filtrar cerques amb agregats.
- *mongoose-paginate-v2*: utilitat per paginar cerques normals.
- *morgan*: Eina de loggin de les peticions HTTP.
- *multer*: middleware per crides multipart/form-data.
- *passport*: middleware d'autenticació.
- *passport-google-oauth20*: estratègia d'autenticació amb Google Oauth per passport.
- *passport-google-oauth20*: estratègia d'autorització amb JWT per passport
- *sharp*: processament i normalització de les imatges que puja l'usuari.
- *streamifier*: utilitat per treballar amb streams sense deixar fitxers físics al servidor.
- *uuid*: generació d'identificadors únics.
- *validator*: validació i sanejament de cadenes de text.

Utilitats per a l'entorn de desenvolupament

- `concurrently`: Permet iniciar simultàniament el servidor backend i frontend.
- `faker`: Generació massiva de dades aleatòries.
- `nodemon`: Reinicia el servidor automàticament en detectar canvis.

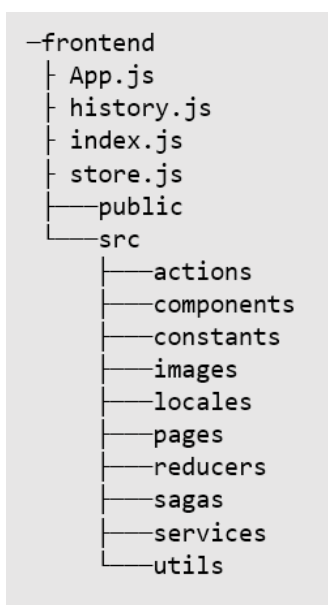
3.3 Desenvolupament del frontend

A continuació, es descriu l'estructura del codi font de l'aplicació client, quin és el propòsit dels fitxers continguts a cada directori, les utilitats més rellevants de les que s'ha fet ús i es mostren alguns fragments de codi d' exemple.

3.3.1 Organització del codi font del client

L'aplicació client s'ha creat amb `Create-React-App`^[23] i s'estructura de la següent manera:

Figura 3.4: Organització del codi font client



En primer lloc, el directori arrel conté un directori anomenat *public* on es troben els fitxers públics de l'aplicació. Això inclou el fitxer *index.html*, que és el punt d'entrada i única pàgina web de l'aplicació. Això és degut a que React construeix Single Page Applications (SPA), on el contingut de la vista es carrega dinàmicament davant les accions de l'usuari sense necessitat de refrescar la pàgina ni redirigir-ne a noves. Al directori font, també se situen diversos fitxers i components amb característiques globals pel que fa al mòdul client. Els fitxers *index.js* i *App.js* contenen respectivament la instrucció de react que s'encarrega de renderitzar els components a la pàgina web i el component principal de l'aplicació que retorna els components en relació a la ruta sol·licitada per l'usuari. El fitxer *history.js* configura l'històric de navegació per poder realitzar accions com redireccions o tornar enrere i el fitxer *store.js* té a veure amb l'emmagatzematge de l'estat de l'aplicació.

El directori *pages* conté components de React que fan referència a una vista determinada. Aquests components estan formats alhora per un conjunt de subcomponents ubicats a la carpeta *components*.

Els directoris *actions*, *reducers* i *sagas* tenen a veure amb el flux de dades de react-redux per gestionar les peticions d'accions al servidor i la modificació i recuperació de l'estat. Tanmateix, el directori *services* conté les crides a l'API de Phonocat.

A la carpeta *locales* s'inclouen tots els fitxers necessaris per a la configuració de la internacionalització^[31] de l'aplicació així com els fitxers de traduccions.

Per últim, el directori *constants* conté variables constants i literals que es fan servir a diversos recursos, i d'aquesta per evitar errors tipogràfics. El directori *images* conté recursos gràfics estàtics i el directori *utils* conté utilitats compartides.

3.3.2 Components de React

Els components de React són peces independents i reutilitzables que permeten construir la interfície d'usuari d'una manera modular. Conceptualment es poden interpretar com a funcions de javascript que reben unes propietats com a paràmetre i retornen codi HTML o altres elements de React amb l'objectiu de descriure el que ha d'aparèixer per pantalla.

Per fer-se una millor idea de com es combinen els components a React per implementar la interfície d'usuari a continuació es mostra la pantalla de perfil d'usuari de l'aplicació amb els diferents components senyalitzats.

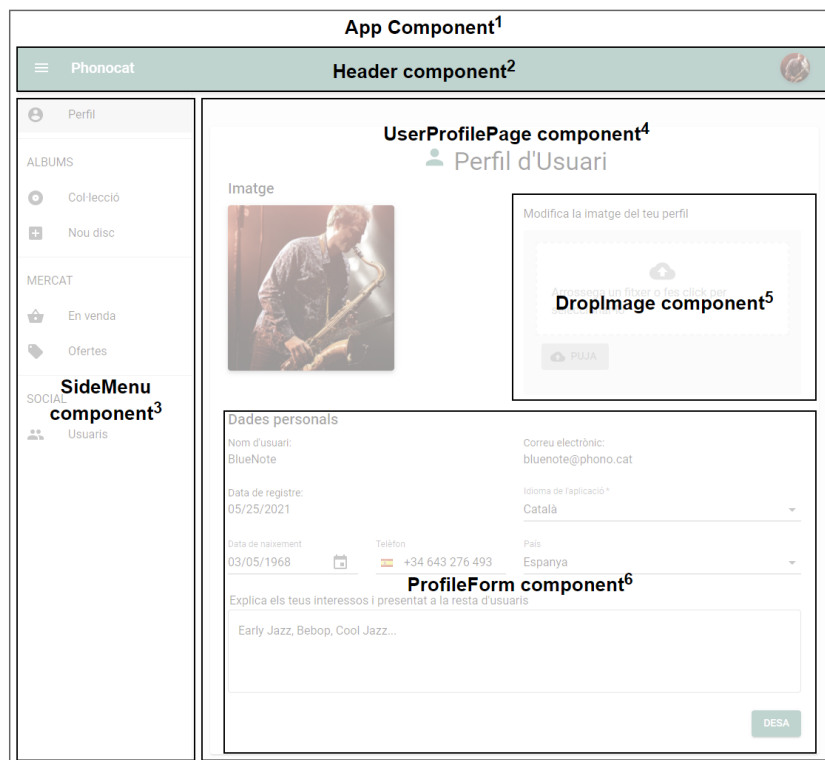


Figura 3.5: Components React perfil d'usuari

Per a aquesta pàgina, es fan servir diversos components que formen la interfície d'usuari. Alguns d'ells, la capçalera, el menú lateral o el component per pujar imatges entren en la categoria de components compartits, i es fan servir a més vistes de l'aplicació.

A React, els components es poden implementar de dues maneres diferents, mitjançant components funcionals o classes^[24]. La principal diferència que hi ha entre els dos mètodes és la sintaxi, els components funcionals són funcions de

JavaScript mentre que els components de classe són classes de JavaScript ES6 que estenen la classe `React.Component` i implementen un mètode de renderització.

A aquest projecte, s'ha decidit fer servir la sintaxi funcional, ja que presenta alguns avantatges. En primer lloc, els components funcionals són més fàcils de llegir que els components de classe i, a més, contenen menys codi. Tanmateix, faciliten la tasca de separar la lògica funcional dels elements de presentació. A més, amb la incorporació dels hooks `useState` i `useEffect` des de fa ja algunes versions de React, es pot controlar el cicle de vida d'aquests i usar l'estat. Amb aquest afegit, els components funcionals no presenten cap mancança davant els components de classe i a més a més, degut a la seva simplicitat, són més eficients. De fet, actualment és la manera de programar que recomana la pròpia llibreria.

El següent codi simplificat mostra com es fa servir la funció `useEffect` per controlar l'estat d'un component.

```
1  const Header = ({...}) => {
2
3    ...
4
5    const auth = useSelector((state) => state.auth)
6    const { userInfo } = auth
7
8    const [language, setLanguage] = useState(auth.userInfo ?
9    auth.userInfo.language : matchBrowserLanguage(i18n.languages[0]))
10
11   useEffect(() => {
12     if (userInfo)
13       setLanguage(userInfo.language)
14   }, [userInfo])
15
16   useEffect(() => {
17     i18n.changeLanguage(language)
18   }, [language])
```

Fragment de codi 3.5: control estat llenguatge aplicació

Aquesta lògica està inclòs al component de la capçalera de l'aplicació. A la funció `useEffect` de la línia 11 d'aquest fragment (veure fragment de codi 3.5), es mostra com cada vegada que es modifica la informació de l'usuari, per exemple en el cas de que aquest faci login, també es modifica l'estat del component amb l'idioma que l'usuari tingui associat com a idioma preferit. En cas que el llenguatge es modifiqui, ja sigui pel motiu anteriorment descrit o per una selecció directa del llenguatge, la funció `useEffect` de la línia 16 detecta que la variable d'estat `language` ha canviat, i s'encarrega de fer la crida a la funció d'internacionalització de la llibreria `i18next` per tal d'efectuar el canvi d'idioma.

3.3.3 Gestió de l'estat amb Redux i Saga

Com ja s'ha fet menció a l'apartat del background tecnològic, s'ha fet servir la llibreria Redux^[25] amb l'objectiu de gestionar i mantenir l'estat global dels components l'aplicació. Redux és una llibreria de JavaScript que s'integra fàcilment amb React i ajuda a mantenir la consistència de les dades.

Redux fa ús principalment de tres components per mantenir l'estat de l'aplicació:

- **Actions:** Les accions són objectes plans de JavaScript que s'envien des de l'aplicació al store o magatzem de l'estat. Com a convenció, per a aquesta aplicació sempre contenen dos atributs, un anomenat type que fa referència al tipus d'acció i és obligatori, i un altre anomenat payload que conté les dades que es volen associar a l'acció. Les accions s'envien mitjançant el mètode dispatch.
- **Reducers:** Els reducers són funcions que reben l'estat actual de l'aplicació i una acció. A partir d'aquestes dades modifiquen l'estat si és necessari i retornen el nou estat. Tenen algunes regles per evitar incoherències com per exemple no permetre modificar l'estat previ directament i retornar sempre una còpia amb les modificacions pertinents, tampoc han de contenir codi asíncron.
- **Store:** El store és l'objecte que manté l'estat actual de l'aplicació i permet accedir-hi. Convé ressaltar que només m'hi ha d'haver un a tota l'aplicació, que pot estar compost per múltiples reducers.

Per poder fer servir redux a l'aplicació, és necessari afegir un altre component. Moltes de les accions de Phonocat fan crides a l'API del servidor, per tant necessita fer crides asíncrones. En un primer moment es va fer ús del middleware Redux Thunk^[26], que permet realitzar aquestes crides i incloure lògica funcional als generadors d'accions. La seva simplicitat fa que sigui molt fàcil d'usar però també es converteix en un problema. Segons l'aplicació va anar creixent, els generadors d'accions, que haurien de ser funcions simples que creïn missatges simples, s'anaven tornant cada vegada més complexes. Per aquest motiu, és va valorar migrar a la llibreria Redux-Saga^[27]. Redux-Saga permet abstraure tota la lògica que conté les crides a l'API i altres funcionalitats asíncrones dins d'uns components anomenats sagas. Aquesta migració va ser relativament costosa, ja que saga fa ús de funcions generadores de JavaScript ES6, també anomenades funcions asterisc, que fins aquest moment no s'havien fet servir al projecte. No obstant això finalment ha resultat ser una bona decisió que ha permet tenir un codi més encapsulat, fàcil de mantenir i de provar.

Els següents fragments de codi simplificat (veure fragment 3.6) mostren com interactuen aquests elements quan es fa una petició per recuperar els LP de l'usuari de l'aplicació.

```
1  /* fitxer LpCollectionPage.js. El component de React en inicialitzar-se fa
2  servir el mètode //dispatch envia una acció de petició de recuperació dels
3  LP */
4  ...
```

```

5     useEffect(() => {
6         dispatch(lpCollectionRequest())
7     }, [])
8     ...
9     /* fitxer lpActions.js. El generador d'accions conté el codi per crear les
10 accions */
11     ...
12     const action = (type, payload = {}) => ({ type, payload })
13
14     export const lpCollectionRequest = () => action(LP_COLLECTION_REQUEST)
15     export const lpCollectionSuccess = (lps) => action(LP_COLLECTION_SUCCESS,
16     lps)
17     export const lpCollectionFail = error => action(LP_COLLECTION_FAIL, error)
18     ...
19     /* fitxer lpActions.js
20 La saga lpSaga observa totes les peticions de tipus LP_COLLECTION_REQUEST
21 amb l'objectiu de tractar-les. Quan detecta una, fa la crida a l'API del
22 servidor i si tot es correcte envia una nova acció de tipus
23 LP_COLLECTION_SUCCESS que conté associat el llistat de LPS */
24     ...
25     function* watchLpCollection() {
26         yield takeLatest(LP_COLLECTION_REQUEST, lpCollection)
27     }
28     ...
29     function* lpCollection() {
30         try{
31             ...
32             const { data } = yield call(getLpCollectionService, userInfo)
33             ...
34             yield put(lpActions.lpCollectionSuccess(lps))
35         } catch (error) {
36             ...
37         }
38     }
39     ...
40     /* fitxer lpReducers.js. El reductor processa les accions i modifica
41 l'estat en conseqüència */
42     export const lpCollectionReducer = (state = {}, action) => {
43         switch (action.type) {
44             case LP_COLLECTION_REQUEST:
45                 return { ...state, loading: true }
46             case LP_COLLECTION_SUCCESS:
47                 return { ...state, loading: false, lps: action.payload }
48             case LP_COLLECTION_FAIL:
49                 return { ...state, loading: false, error: action.payload }
50             ...
51         }
52     }

```

Fragment de codi 3.6: Exemple ús redux-saga

Aquest flux del sistema es representa conceptualment a la figura 3.6.

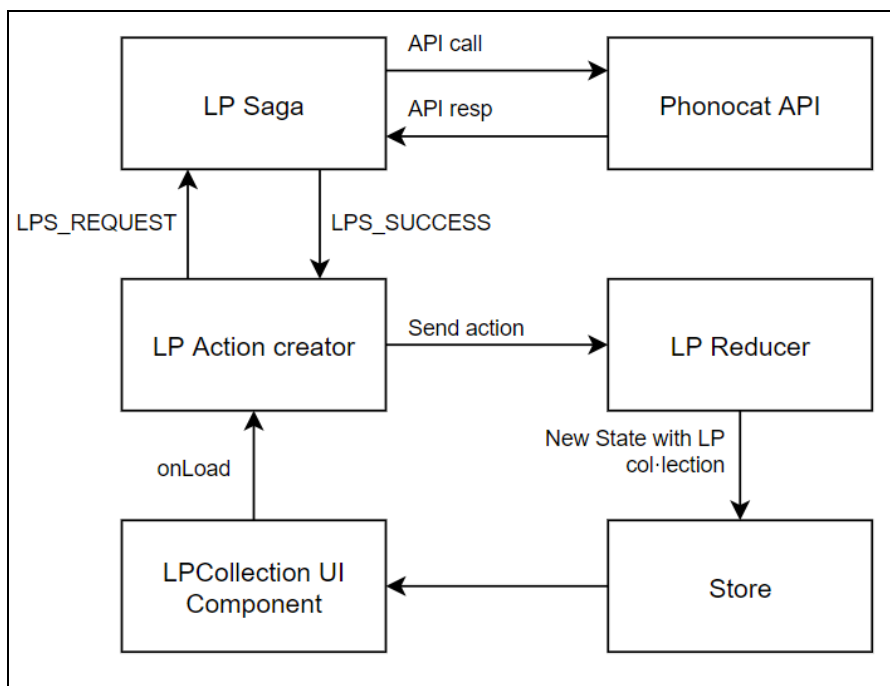


Figura 3.6: Flux redux-saga

Una utilitat que ha facilitat el desenvolupament d'aquest flux és l'extensió `redux-devtools-extension`, que juntament amb la extensió `ReduxDevTools` de Google Chrome, permet fer seguiment de l'estat de l'aplicació d'una manera visual. A les següents captures es pot observar com l'estat del reductor `lpCollection` passa de tenir l'atribut `loading` amb valor `true` (Figura 3.7), a l'atribut `loading` amb valor `false` i el llistat de `lps` emplenat (Figura 3.8).

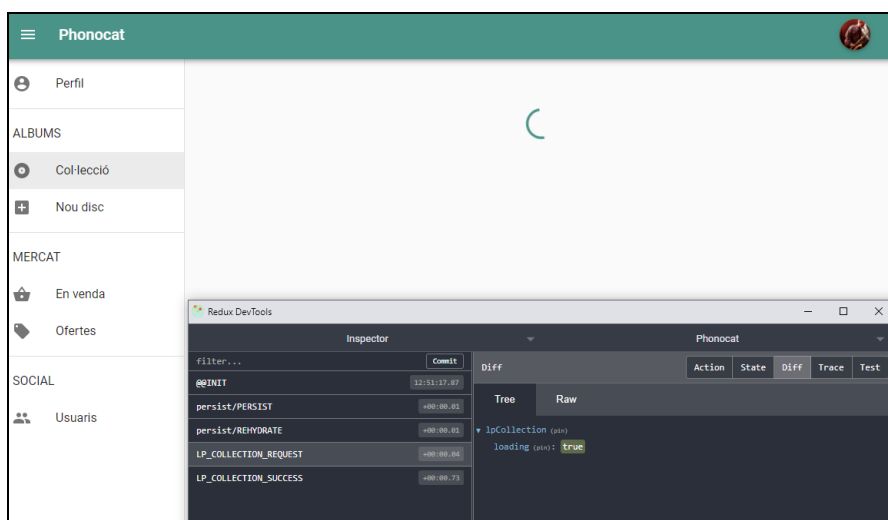


Figura 3.7: Estat aplicació petició col·lecció LP

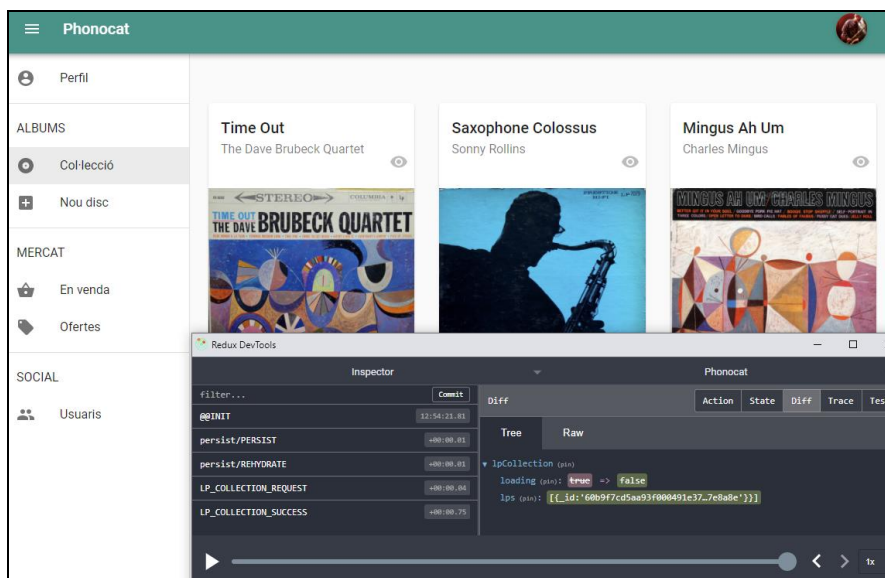


Figura 3.8: Estat aplicació col·lecció LP carregada satisfactòriament

3.3.4 Encaminament de rutes

Amb el propòsit de mantenir i controlar la navegació dins l'aplicació s'ha fet servir la llibreria React Router DOM ^[28] i un objecte històric dels accessos per tal de poder fer redireccions. El següent fragment de codi mostra com s'ha sobreescrit el component *Route* d'aquesta llibreria per tal de controlar tots aquells accessos que necessiten autenticació i retornar a l'usuari a la pàgina d'entrada si no està correctament autoritzat.

```

1 import { Route, Redirect } from 'react-router-dom';
2 ...
3 //Redirecciona a la pàgina d'inici si l'usuari no està autenticat
4 function PrivateRoute({ component: Component, ...rest }) {
5
6     const auth = useSelector((state) => state.auth)
7     const { userInfo } = auth
8
9     return <Route {...rest} render={props => (
10         userInfo ? <Component {...props} /> : <Redirect to={{
11     pathname: '/', state: { from: props.location } }} />
12     )} />
13 }...
```

3.3.5 Validació de formularis amb Formik i yup

Una altra llibreria que ha facilitat la implementació del frontend de l'aplicació és Formik, una de les llibreries de control de formularis més populars de React.

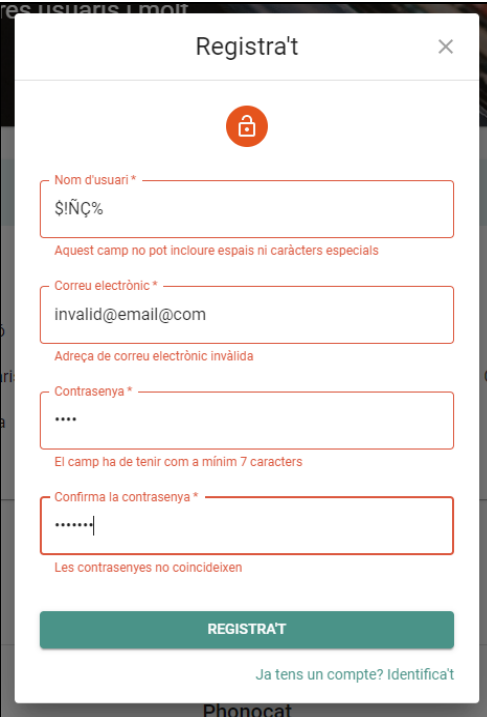
Inicialment es va valorar l'ús de ReactHookForms, que guanya en eficiència de renderització, però Formik va resultar adaptar-se millor a nivell de codi amb els components gràfics de Material-UI^[32], la qual cosa va ser determinant en la seva elecció. Un altre requisit que es va valorar va ser la integració amb yup, en aquest cas ambdues llibreries s'hi integren correctament.

Yup permet simplificar les tasques de validació mitjançant la definició d'esquemes. A mode d'exemple, el següent esquema mostra com s'efectua la validació del formulari de registre d'usuari:

```
1 const loginSchema = yup.object().shape({
2   username:
3   yup.string().required().min(USERNAME_MIN_LENGTH).matches(/^(\\d|\\w)+$/,
4   t('yup:noSpecialChars')),
5   email: yup.string().email().required(),
6   password: yup.string().required().min(PASSWORD_MIN_LENGTH),
7   password2: yup.string().required().oneOf([yup.ref('password'), null],
8   t('registerForm.differentPasswords'))
9 })
```

Fragment de codi 3.7: Esquema de validació formulari registre

Aquest esquema valida que el nom de l'usuari tingui com a mínim 5 caràcters, que no contingui caràcters especials, que el correu electrònic sigui vàlid, que el password tingui una longitud de com a mínim 7 caràcters i que els dos passwords coincideixin. Amb la seva integració amb Material-UI i Formik, el resultat quan s'incompleteixen aquestes validacions es mostra a la figura 3.8.



La imatge mostra una interfície d'usuari per a un formulari de registre titulat "Registra't". El formulari té quatre camps d'entrada amb errors de validació: 1. Camp "Nom d'usuari *": Conté el text "\$!NÇ%" i un missatge d'error "Aquest camp no pot incloure espais ni caràcters especials". 2. Camp "Correu electrònic *": Conté "invalid@email@com" i un missatge d'error "Adreça de correu electrònic invàlida". 3. Camp "Contrasenya *": Conté quatre punts i un missatge d'error "El camp ha de tenir com a mínim 7 caràcters". 4. Camp "Confirma la contrasenya *": Conté sis punts i un missatge d'error "Les contrasenyes no coincideixen". A sota dels camps hi ha un botó "REGISTRAT" i un enllaç "Ja tens un compte? Identifica't". El logotip "Phonocat" està a la part inferior.

Figura 3.9: Validacions formulari registre

3.3.6 Llistat d'utilitats frontend

Les següents llibreries s'han fet servir per a resoldre o facilitar les següents funcionalitats del backend:

- *@material-ui/core*: components Material UI.
- *@material-ui/icons*: icones Material UI.
- *@material-ui/core*: components Experimentals Material UI.
- *@material-ui/pickers*: selectors de dates Material UI.
- *autosuggest-highlight*: ressaltat de subcadena de text.
- *clsx*: generació dinàmica de noms de classes.
- *formik*: gestió de formularis.
- *i18next*: internacionalització.
- *i18next-browser-languagedetector*: detecció de l'idioma del navegador per a usuaris no identificats.
- *i18next-intervalplural-postprocessor*: processament de traduccions.
- *lodash*: utilitat per a treballar amb estructures de dades JavaScript.
- *material-ui-image*: càrrega atractiva d'imatges.
- *material-ui-phone-number*: input números de telèfon.
- *moment*: formats de temps.
- *react*: llibreria React.
- *react-dom*: llibreria de React per treballar amb el DOM.
- *react-dropzone*: component gràfic per pujar fitxers.
- *react-google-login*: component gràfic de login amb Google.
- *react-i18next*: hook de react per a la internacionalització.
- *react-infinite-scroll-component*: component de scroll infinit.
- *react-number-format*: format numèric.
- *react-redux*: integració de redux amb react.
- *react-router-dom*: encaminament de rutes.
- *react-scripts*: configuració de react.
- *react-slick*: carrusel d'imatges.
- *redux*: llibreria redux.
- *redux-persist*: persistència de l'estat amb redux.
- *redux-saga*: creació de sagas.
- *reselect*: optimització de redux per a components exigents ^[29].
- *timeago.js*: conversió d'unitats de temps en la seva representació textual.
- *yup*: validació de dades amb esquemes.
- *redux-devtools-extension*: utilitat per depurar l'estat de l'aplicació.

3.4 Integració al núvol

Com s'ha descrit a l'apartat del background tecnològic, durant la fase de recerca tecnològica es va explorar la possibilitat de desplegar l'aplicació al núvol. Com a resultat d'aquesta investigació, s'ha desplegat una versió funcional de l'aplicació al servidor Heroku. La integració amb aquesta plataforma ha resultat generalment senzilla, un dels seus punts forts és que permet pujar una nova versió de l'aplicació amb la instrucció PUSH de l'eina de control de versions Git.

No obstant això, per tal que l'aplicació funcione correctament s'han hagut d'indicar les variables d'entorn a l'administració de Heroku i s'ha hagut de fer una petita

adaptació al fitxer server.js del backend per tal de capturar les peticions que no siguin part de l'API i enviar-les al mòdul client.

```
if (process.env.NODE_ENV === 'production') {
  app.use(express.static(path.join(__dirname, '/frontend/build')))

  app.get('*', (req, res) =>
    res.sendFile(path.resolve(__dirname, 'frontend', 'build',
    'index.html'))
  )
}
```

Fragment de codi 3.8: adaptació entorn producció

La base de dades també s'allotja al núvol gracies a MongoDB Atlas, la realització d'aquest pas ha estat extremadament sencilla i només ha consistit en crear una instancia de MongoDB Atlas i substituir la url de la base de dades local per la nova.

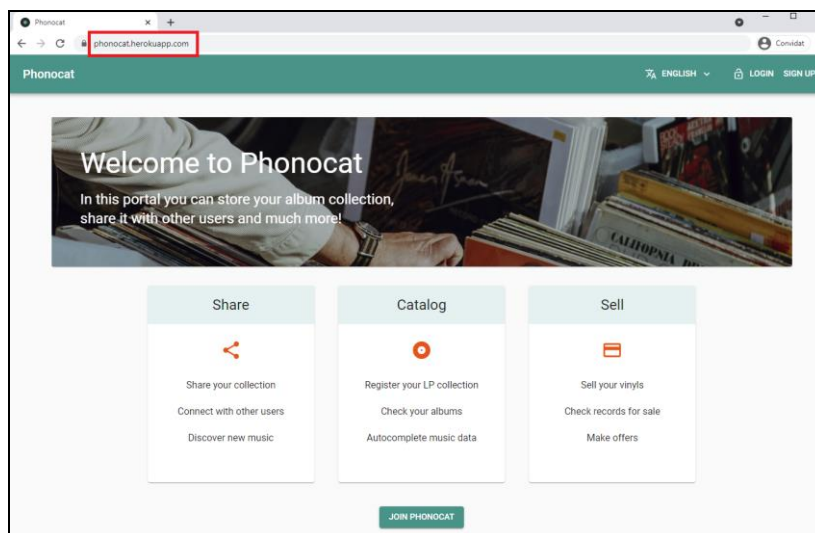


Figura 3.10: Aplicació desplegada al núvol

En darrer terme, s'ha decidit fer servir el servei al núvol Cloudinary^[14]. Aquest servei ha requerit algunes adaptacions al codi com la creació d'un middleware de Cloudinary per a node i la creació d'una utilitat per pujar els fitxers rebuts a l'API (veure fragment de codi 3.9).

```
1  /**
2   * Uploads an image to Cloudinary media storage service
3   * @param {*} buffer file buffer
4   * @param {*} folder destination folder
5   * @param {*} public_id resource public_id
6   * @returns Promise with service result
7   */
8  export const uploadImageBufferCloud = async (buffer, folder,
9  public_id) => {
10   return new Promise((resolve, reject) => {
11     const uploadStream = cloudinary.uploader.upload_stream(
```

```

12         {
13             folder,
14             public_id
15         },
16         (error, result) => {
17             if (result) resolve(result);
18             else reject(error);
19         },
20     );
21     streamifier.createReadStream(buffer).pipe(uploadStream);
22 })

```

Fragment de codi 3.9: utilitat pujar fitxers cloudinary

3.5 Resultat del producte

El propòsit d'aquest apartat és mostrar una visió global del resultat del producte mitjançant captures de l'aplicació.

3.5.1 Pagina d'entrada

La pagina d'inici és d'accés públic i conté els accessos per registrar-se, ifentificar-se i canviar d'idioma al català, castellà o anglès.

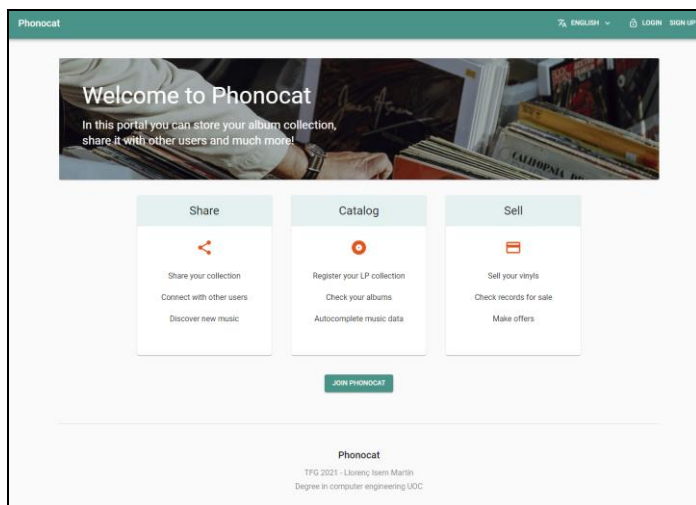


Figura 3.11: Landing page

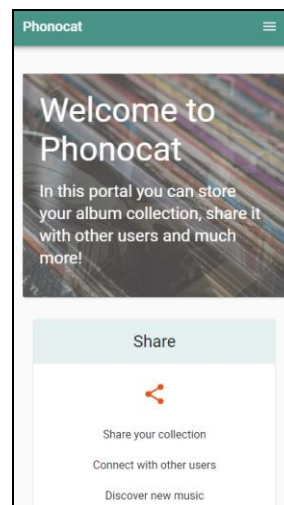


Figura 3.12: Landing page mòbil

3.5.2 Login/Registre

Els formularis de login i registre es mostren com a finestra emergent. L'aplicació també permet accedir amb un compte de Google, en aquest cas si es tracta del primer accés, es crea un nou usuari a la base de dades i se li assigna l'identificador de google d'aquest compte.

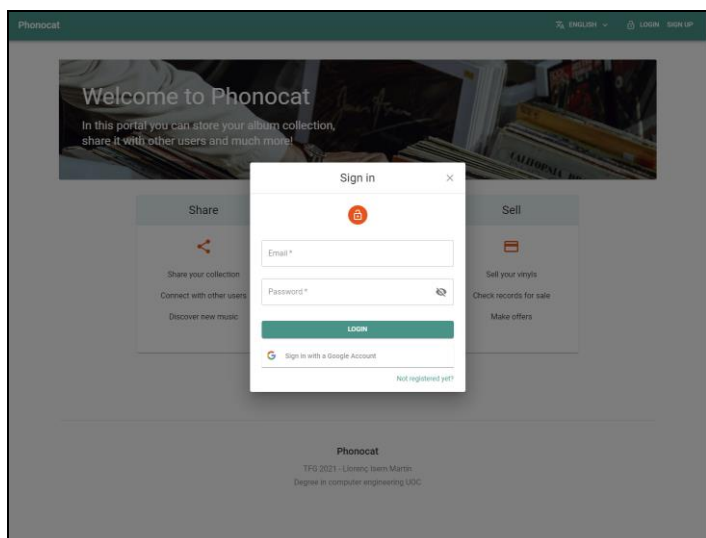


Figura 3.13: Login

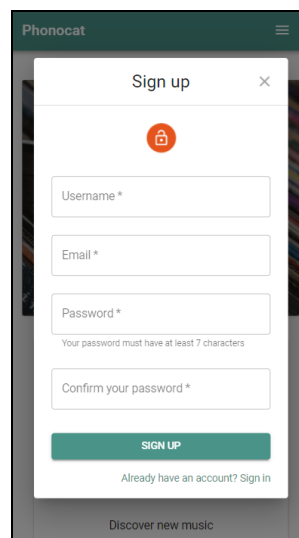


Figura 3.14: Registre

3.5.3 Col·lecció de LPs

Una vegada l'usuari s'ha identificat, es redirecciona automàticament a la seva col·lecció de LPs, on se li demanarà afegir-ne un de nou si encara no en té. Aquesta acció també es pot dur a terme en qualsevol moment mitjançant el menú lateral amb l'opció "Nou disc".

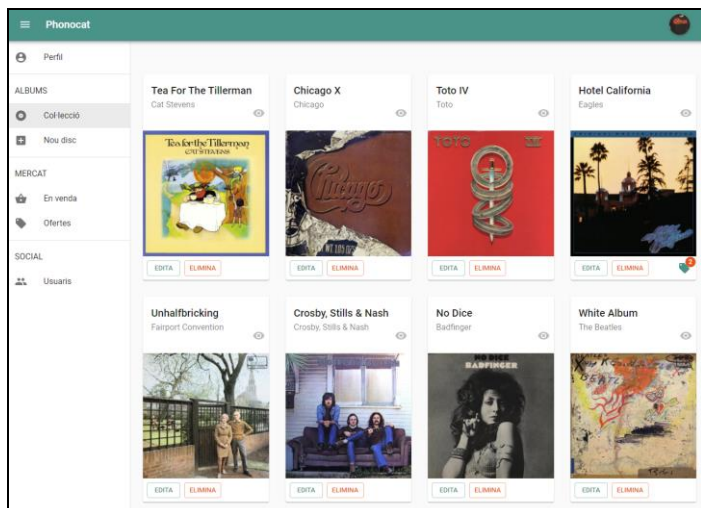


Figura 3.15: Col·lecció de discos

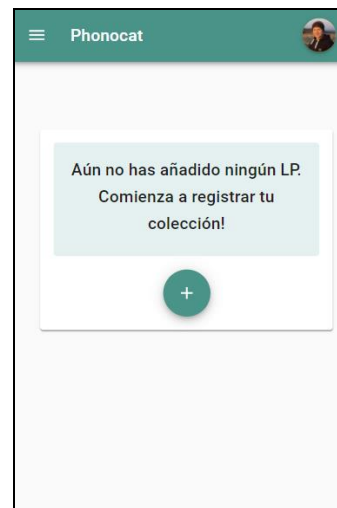


Figura 3.16: Col·lecció buida

3.5.4 Afegir LP

Per introduir un nou disc, l'aplicació fa ús de fonts externes per ajudar a l'usuari a omplir les dades del seu LP i a més, suggereix una imatge de portada.

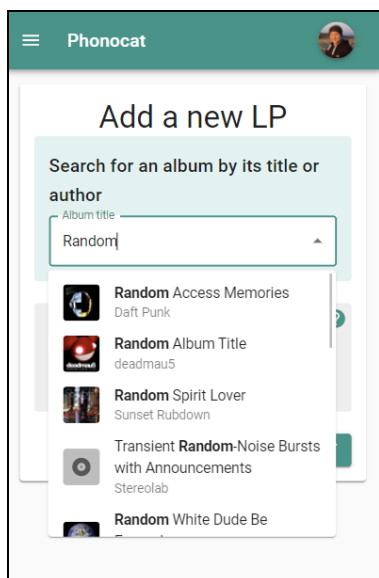


Figura 3.17: Cercar àlbum

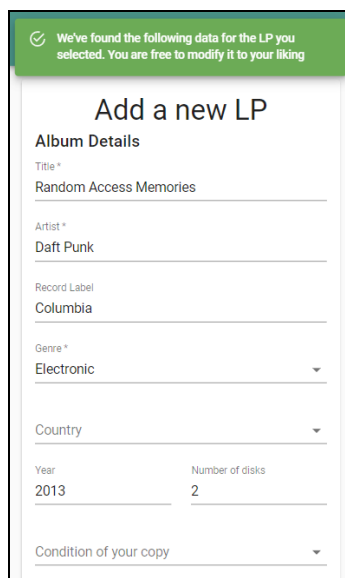


Figura 3.18: Afegir LP

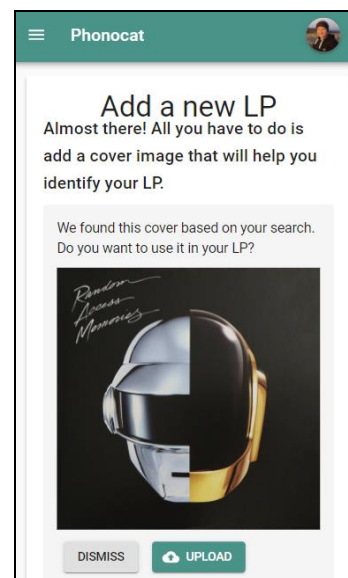


Figura 3.19: Portada suggerida

Els discos afegits es poden consultar, editar i eliminar. Entre les dades que s'introdueixen de cada disc s'inclouen el títol, autor, segell discogràfic, gènere, dades de l'estat físic del disc i la possibilitat de fer visible o ocult el disc a la resta d'usuaris. A més, els usuaris poden afegir ressenyes dels seus LPs i consultar les seves cançons.

3.5.5 Anunciar LP en venda

Els usuaris poden anunciar un LP de la seva col·lecció en venda a partir de la consulta del disc. Aquesta funcionalitat obliga a indicar el preu de venda desitjat i opcionalment permet pujar fins a 4 imatges associades a l'anunci.

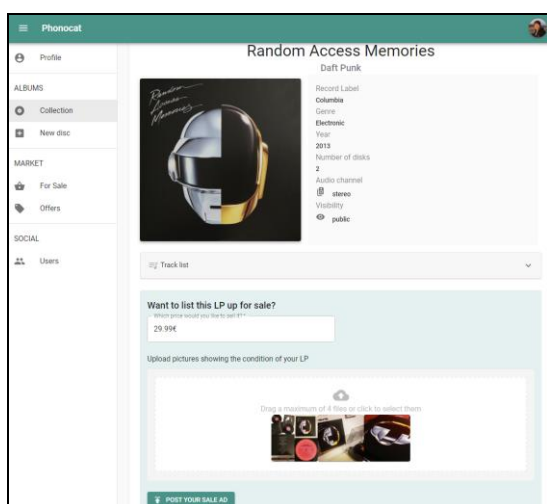


Figura 3.20: Anunciar LP en venda

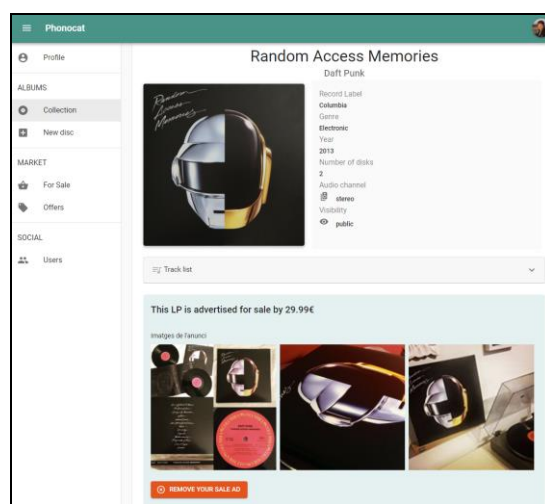
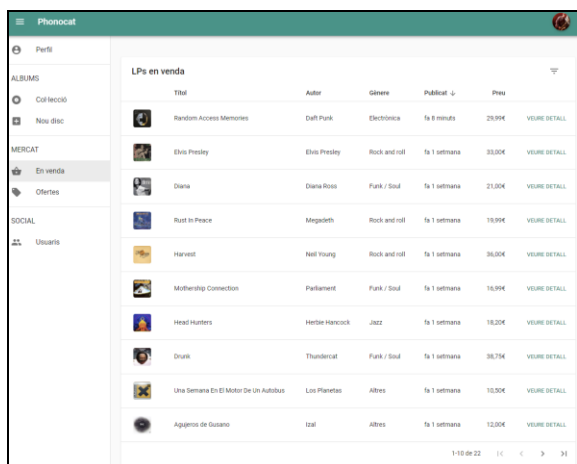


Figura 3.21: LP anunciat

3.5.6 Consultar LPs en venda

Els discos anunciats apareixen al punt de menú “En venda” a una taula paginada que permet ordenació pels diferents camps i filtratge per gènere musical o rang de preus.



Títol	Autor	Gènere	Publicat	Preu	VEURE DETALL
Random Access Memories	Daft Punk	Electrònica	fa 8 minuts	29,99€	VEURE DETALL
Elvis Presley	Elvis Presley	Rock and roll	fa 1 setmana	33,00€	VEURE DETALL
Diana	Diana Ross	Funk / Soul	fa 1 setmana	21,00€	VEURE DETALL
Rust In Peace	Megadeth	Rock and roll	fa 1 setmana	19,99€	VEURE DETALL
Harvest	Neil Young	Rock and roll	fa 1 setmana	36,00€	VEURE DETALL
Motherhip Connection	Parliament	Funk / Soul	fa 1 setmana	16,99€	VEURE DETALL
Head Hunters	Herbie Hancock	Jazz	fa 1 setmana	18,20€	VEURE DETALL
Drunk	Thundercat	Funk / Soul	fa 1 setmana	38,75€	VEURE DETALL
Una Semana En El Motor De Un Autobus	Los Planetas	Altres	fa 1 setmana	10,00€	VEURE DETALL
Aggroes de Susano	tzal	Altres	fa 1 setmana	12,00€	VEURE DETALL

Figura 3.22: Llistat de discos en venda

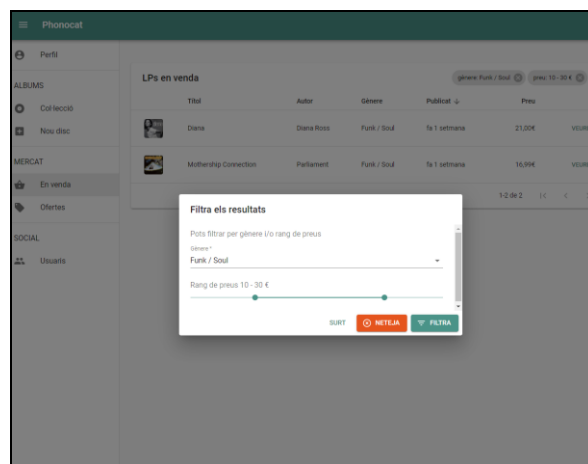


Figura 3.23: Filtratge anuncis

3.5.7 Fer ofertes de compra

Si es fa click a la portada d'un disc del llistat d'anuncis o al botó veure detall, es permet fer una oferta per aquest disc sempre que no en siguem ja els propietaris.

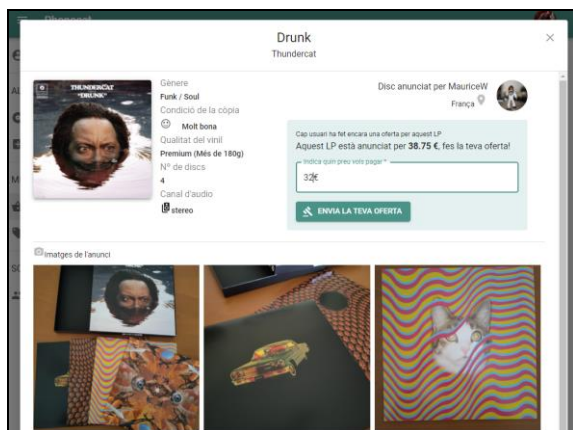


Figura 3.24: Realitzar oferta

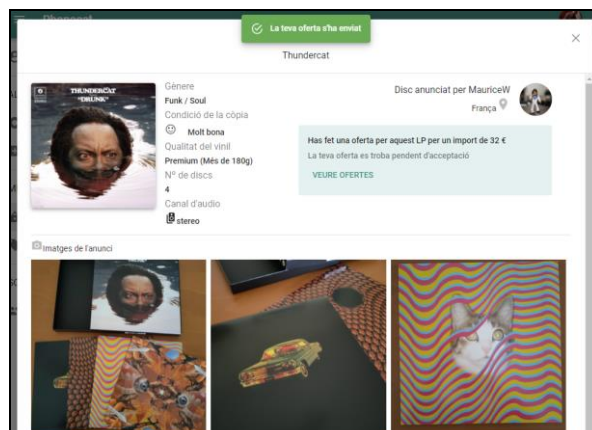


Figura 3.25: Oferta realitzada

3.5.8 Gestionar ofertes

Al punt de menú ‘Ofertes’, podem veure l'estat de les ofertes realitzades i acceptar o rebutjar les ofertes rebudes, així com valorar-les quan ja s'han acceptat. D'altra banda, es poden veure les ofertes realitzades i el seu estat actual.

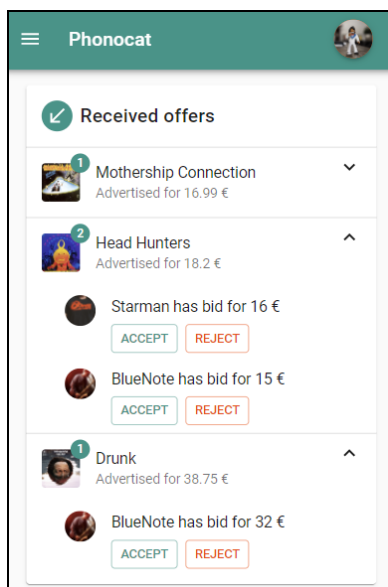


Figura 3.26: Ofertes rebudes

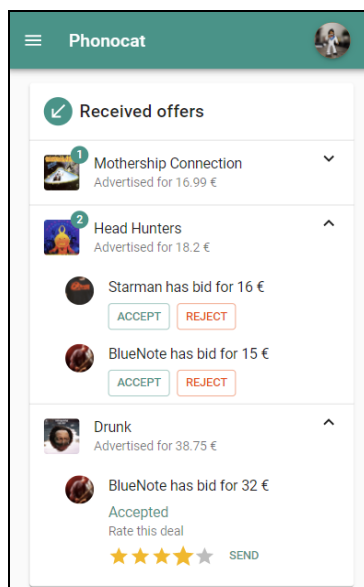


Figura 3.27: Valorar transacció

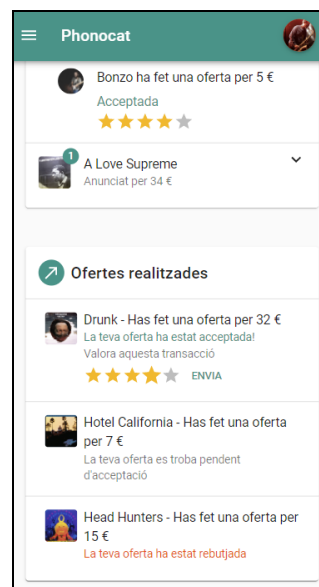


Figura 3.28: Ofertes realitzades

Quan l'usuari comprador i venedor han valorat la transacció d'un LP, el disc passa automàticament a la col·lecció del comprador.

3.5.9 Consultar discos d'altres usuaris

Per a consultar els discos d'altres usuaris cal accedir al punt de menú "Usuaris" del menú lateral. Aquesta pantalla mostra un llistat d'usuaris amb scroll infinit asíncron. En seleccionar un usuari s'accedeix al seu detall, on es poden veure les seves dades i el llistat de discos de la seva col·lecció sempre que tinguin visibilitat pública.

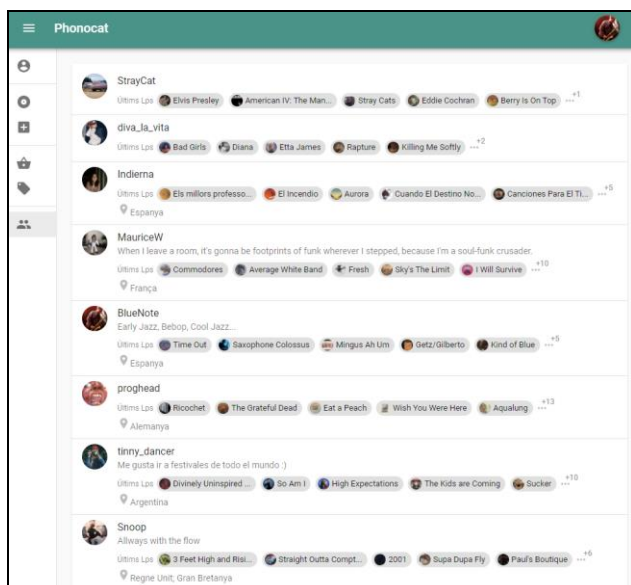


Figura 3.29: Llistat d'usuaris

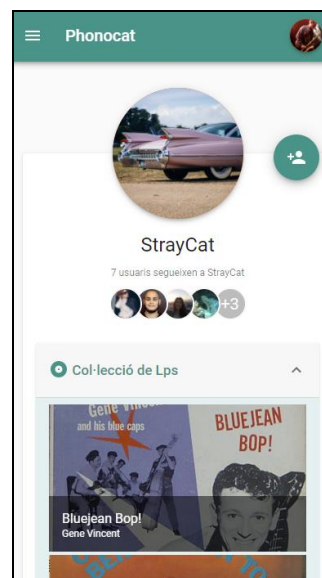


Figura 3.30: Detall d'usuari

Dins la vista de detall de l'usuari, aquest es pot seguir o deixar de seguir fent clic a la icona situada a la dreta de l'avatar.

3.5.10 Perfil d'usuari i sessió

Els usuaris poden accedir al seu perfil de dues maneres. Mitjançant l'opció "Perfil" del menú lateral o mitjançant el menú desplegable que s'obre en fer clic a l'avatar de la capçalera.

Dins del seu perfil els usuaris poden configurar l'idioma de l'aplicació i editar dades personals com la data de naixement o el seu país de residència.

D'altra banda, el menú desplegable superior també mostra l'opció de tancar la sessió. Aquesta acció esborra el token d'autorització de l'usuari i el redirigeix a la pàgina d'entrada.



Figura 3.31: Top menu

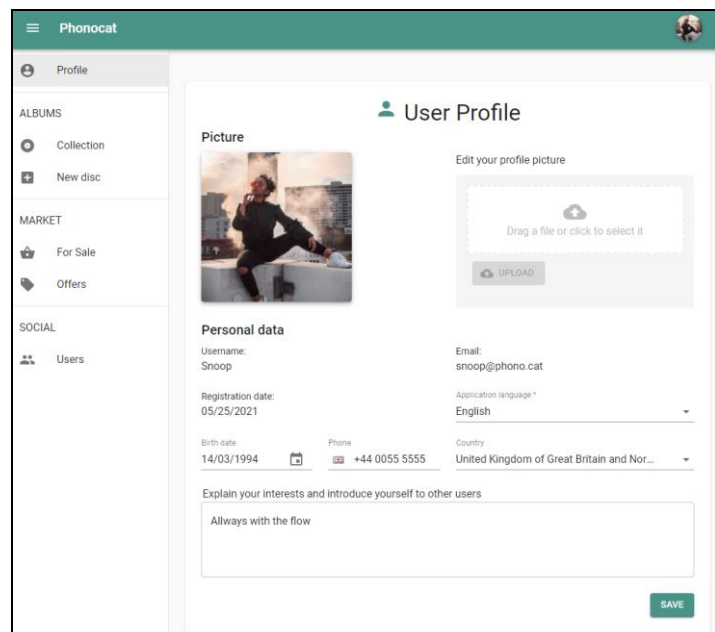


Figura 3.32: User profile

4 Conclusions

Aquest apartat conté una valoració global del resultat del projecte i la seva execució així com una descripció de les línies de treball futur que s'ofereixen prenent com a punt de partida la versió actual de l'aplicació.

4.1 Valoració del projecte

En primer lloc, el nivell de satisfacció pel que fa a l'assoliment dels objectius marcats a l'apartat 1.2 és molt elevat, especialment tenint present que el conjunt de tecnologies que s'han fet servir per desenvolupar el projecte eren, o bé totalment noves, o bé es disposava de coneixements elementals.

A partir de l'anàlisi de requisits inicial es va fer una valoració de l'abast i es va decidir descartar algunes funcionalitats com el mòdul de missatgeria instantània i el sistema de notificacions per correu electrònic. Aquesta decisió va resultar encertada, ja que l'abast inicial era massa ampli considerant els condicionants abans descrits.

Tanmateix, es valora positivament la planificació i metodologia de desenvolupament del projecte de manera global. No obstant això, la recerca i formació en el conjunt d'eines, llenguatges i llibreries que finalment s'han fet servir va portar més temps del previst. Conseqüentment, l'inici de la implementació es va veure afectat lleugerament.

Un altre obstacle que va sorgir al llarg de la fase de desenvolupament de l'aplicació va ser haver de canviar de tecnologia, com és el cas del canvi de *redux-thunk* a *redux-saga*, ja que la primera llibreria no va permetre satisfer alguns dels principis de bones pràctiques que s'han volgut mantenir pel que fa a la estructuració del codi.

Com a lliçons apreses, s'han adquirit coneixements avançats en un conjunt de tecnologies que sens dubte seran d'utilitat alhora d'afrontar nous projectes, especialment a l'àmbit del desenvolupament web. L'execució d'aquest treball, a més, ha servit per aprendre a valorar d'una manera més acurada els recursos que cal dedicar en l'aprenentatge i perfeccionament de noves eines. Un altre aspecte que s'ha senyalat com a lliçó apresada té a veure amb la valoració i dedicació dels recursos necessaris per a la implementació de la interfície d'usuari. Aquesta es va subestimar en un primer moment, tot i això, aquests obstacles es van poder superar i com a resultat, s'ha pogut obtenir un producte final que satisfà les expectatives inicials.

4.2 Línies de treball futur

Amb l'horitzó d'obtenir un producte més complet, seria convenient afegir el sistema de missatgeria instantània amb l'ús de *web-sockets*. L'ús d'aquesta tecnologia també permetrà la implementació d'un sistema de notificacions asíncron, que proporcioni una experiència d'usuari encara més òptima i atractiva.

Una millora que s'hagués incorporat si en el moment d'iniciar el projecte s'haguessin tingut els coneixements i experiència actuals, hauria estat la utilització del llenguatge *TypeScript*^[33] tant al servidor com al client. Aquest llenguatge és una ampliació de

javascript que el converteix en un llenguatge fortament tipat, la qual cosa proporciona un codi més comprensible i fàcil de verificar.

Un altre punt a explorar és l'aprofundiment en funcionalitats PWA, com l'ús de notifikacions a dispositius mòbils. En aquest sentit, es podria anar encara més lluny i desenvolupar una interfície nativa per aquest tipus de dispositius amb l'ús de React-Native. Això permetria ampliar les funcionalitats de l'aplicació. Per exemple, es podria fer ús dels sistemes de geo-localització que incorporen els telèfons intel·ligents per millorar el sistema de compra venda permetent als usuaris cercar anuncis de venda per proximitat.

Finalment, una altra línia de treball a investigar és l'ús de microserveis que siguin funcionals per si sols, mitjançant tècniques de replicació de base de dades. D'aquesta manera es podria obtenir com a resultat una aplicació encara més flexible i escalable.

5 Glossari

API: Conjunt de definicions i protocols que es fa servir per a desenvolupar programari. De l'anglès Application Programming Interface.

B2B: Bussiness to Bussiness. Fa referència a intercanvi de dades entre el negoci de dues entitats.

B2C: Bussiness to Consumer. Fa referència a l'enviament de dades de l'aplicació al consumidor del producte.

Backend: Capa del costat del servidor d'una aplicació

Cloud: Servei accessible en línia.

DOM: De l'anglès Document Object Model. Estructura i conjunt d'objectes d'un document HTML.

Endpoint: URL de destí d'una arquitectura orientada a serveis.

Frontend: Capa de presentació d'una aplicació.

Full stack development: Desenvolupament que cobreix tots els aspectes d'una aplicació.

Hook: Interfície que permet incloure codi customitzat.

JSON: De l'anglès JavaScript Object Notation. Format destinat a emmagatzemar o transportar dades.

JWT: De l'anglès Json Web Token. Estandard per a transmetre informació d'una manera segura i lleugera que fa servir el format JSON.

Middleware; Programari que se situa enmig de diferents processos d'un sistema.

MVC: De l'anglès Model-View-Controller. Estil d'arquitectura que separa l'aplicació en tres components lògics: Vista, model i controlador.

MVP: De l'anglès Minimum Viable Product . Producte mínim viable.

NoSQL: Sistema de gestió de base de dades que difereix del model relacional pensat per afavorir el treball amb grans volums de dades

PaaS: Plataforma de computació al núvol que ofereix un entorn de desenvolupament, execució i administració d'aplicacions.

PWA: De l'anglès Progressive Web Application. Aplicació web que pretén proporcionar una experiència d'usuari similar a la obtinguda amb aplicacions natives.

REST: Representational State Transfer. Estil d'arquitectura de programari sense estat que serveix per crear aplicacions que mitjançant serveis web.

RESTful: Servei web que segueix l'arquitectura REST.

SOA: De l'anglès Service Oriented Architecture. Arquitectura basada en serveis.

UI: De l'anglès User Interface. Interfície d'usuari.

6 Bibliografia

- [1] Van Casteren, W. (2017 Febrer). "The Waterfall Model and the Agile Methodologies : A comparison by project characteristics" ResearchGate [publicació electrònica] <https://www.researchgate.net/publication/313768756_The_Waterfall_Model_and_the_Agile_Methodologies_A_comparison_by_project_characteristics> [última consulta 02/06/2021]
- [2] Material Design [guia de disseny web] <https://material.io/design> [última consulta 04/06/2021]
- [3] LastFM API <<https://www.last.fm/api>> [última consulta 25/05/2021]
- [4] Discogs API <<https://www.discogs.com/developers>> [última consulta 07/06/2021]
- [5] Google auth API <<https://developers.google.com/identity/protocols/oauth2>> [última consulta 17/05/2021]
- [6] Volere. Volere Requirements Specification Template [plantilla requisits] <<https://www.volere.org/templates/volere-requirements-specification-template/>> [última consulta 01/06/2021]
- [7] Planning a Minimum Viable Product: a Step-By-Step Guide. Clearbridge Mobile. Medium [publicació electrònica] <<https://medium.com/@ClrMobile/planning-a-minimum-viable-product-a-step-by-step-guide-6f387d657870>> [última consulta 03/06/2021]
- [8] Synder, C. Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces [llibre] San Francisco: Morgan Kaufmann
- [9] What is Full Stack? W3schools [recurs electrònic] <https://www.w3schools.com/whatis/whatis_fullstack.asp#:~:text=A%20full%20stack%20web%20developer,ASP%2C%20Python%2C%20or%20Node> [última consulta 02/06/2021]
- [10] What is a REST API? <<https://www.redhat.com/en/topics/api/what-is-a-rest-api>> [última consulta 28/05/2021]
- [11] NoSQL explained <<https://www.mongodb.com/nosql-explained>> [última consulta 04/06/2021]
- [12] Enquesta Stackoverflow <<https://insights.stackoverflow.com/survey/2020#technology-web-frameworks>> [última consulta 03/06/2021]
- [13] Heroku CLI <<https://devcenter.heroku.com/articles/heroku-cli>> [última consulta 08/06/2021]
- [14] Okpukoro, J. Uploading Images to Cloudinary Using Multer and ExpressJS. Medium. <<https://medium.com/@joeokpus/uploading-images-to-cloudinary-using-multer-and-expressjs-f0b9a4e14c54>> [última consulta 27/05/2021]
- [15] Node.js [documentació oficial] <<https://nodejs.org/es/docs/>> [última consulta 08/06/2021]
- [16] Express API [documentació oficial] <<https://expressjs.com/es/4x/api.html>> [última consulta 08/06/2021]
- [17] MongoDB API [documentació oficial] <<https://api.mongodb.com/>> [última consulta 08/06/2021]

- [18] React docs [documentació oficial] <<https://es.reactjs.org/docs/getting-started.html>> [última consulta 08/06/2021]
- [19] Material UI [documentació oficial] <<https://material-ui.com/>> [última consulta 08/06/2021]
- [20] Heroku with Node <<https://devcenter.heroku.com/articles/getting-started-with-nodejs>> [última consulta 08/06/2021]
- [21] Project structure for an Express REST API <<https://www.coreycleary.me/project-structure-for-an-express-rest-api-when-there-is-no-standard-way>> [última consulta 01/06/2021]
- [22] Peyrott, S. JWT Handbook [llibre electrònic] <https://assets.ctfassets.net/2ntc334xpx65/o5J4X472PQUI4ai6cAcqg/13a2611de03b2c8edb009c3ca14ae86b/jwt-handbook-v0_14_1.pdf> [última consulta 01/06/2021]
- [23] Create React App <<https://es.reactjs.org/docs/create-a-new-react-app.html>> [última consulta 08/06/2021]
- [24] Jötch, D. Functional vs Class-Components in React <<https://djoech.medium.com/functional-vs-class-components-in-react-231e3fbd7108>>
- [25] Redux <<https://es.redux.js.org/>> [última consulta 08/06/2021]
- [26] Redux Thunk <<https://github.com/reduxjs/redux-thunk>> [última consulta 08/06/2021]
- [27] Sagas vs. Thunk <<https://www.paradigmadigital.com/dev/sagas-vs-thunk/>> [última consulta 01/06/2021]
- [28] React Router DOM <<https://reactrouter.com/web/guides/quick-start>> [última consulta 08/06/2021]
- [29] Memoizing Selector Functions [documentació oficial] <<https://redux.js.org/tutorials/essentials/part-6-performance-normalization>> [última consulta 10/05/2021]
- [30] Khalfallah, H. NodeJS Rest API with Express, Passport, JWT and MongoDB. <<https://helabekhalfallah.medium.com/nodejs-rest-api-with-express-passport-jwt-and-mongodb-98e5f2fee496>> [última consulta 07/06/2021]
- [31] Shanika, W. How to internationalize a React application using i18next. <<https://lokalise.com/blog/how-to-internationalize-react-application-using-i18next/>> [última consulta 03/06/2021]
- [32] Formik and Material UI Integration <<https://www.paradigmadigital.com/dev/sagas-vs-thunk/>> [última consulta 01/06/2021]
- [33] TypeScript <<https://www.typescriptlang.org/>> [última consulta 08/06/2021]

Annex I. Especificació API REST Phonocat

User endpoints

Registrar usuari al sistema

Petició			
Mètode	POST	URL	/users
Accés	Públic		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
body	Cos Obligatori	Dades de l'usuari { "username" : " BlueNote", "email" : " bluenote@phono.cat", "password" : "uoc1234" }	
Respostes			
Codi	Descripció		
201	Dades de l'usuari		
409	Si l'usuari ja existeix o s'incompleix alguna validació retorna un missatge d'error		

Identificar usuari al sistema

Petició			
Mètode	POST	URL	/users/login
Accés	Públic		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
body	Cos Obligatori	Credencials de l'usuari { "email" : " bluenote@phono.cat ", "password" : " uoc1234" }	
Respostes			
Codi	Descripció		
200	retorna les dades de l'usuari i un token d'autenticació JWT.		
401	Si les credencials no són vàlides, retorna un missatge d'error.		

Identificar usuari al sistema mitjançant google

Petició			
Mètode	POST	URL	/login/google
Accés	Públic		
ContentType	application/json		
Respostes			
Codi	Descripció		
200	retorna les dades de l'usuari i un token d'autenticació JWT.		
401	Si l'autenticació falla, retorna "No autoritzat"		

Consultar perfil d'usuari

Petició			
Mètode	GET	URL	/users/{username}/profile
Accés	Autenticat. Si l'usuari autenticat no coincideix amb el de l'usuari a consultar, només es retornen les dades públiques		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari a consultar	
Respostes			
Codi	Descripció		
200	Dades de l'usuari		
404	Usuari no trobat		
401	No autoritzat		

Editar perfil d'usuari

Petició			
Mètode	PATCH	URL	/users/{username}/profile
Accés	Autenticat com a l'usuari a modificar		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta	Nom de l'usuari a editar	
body	Cos	Noves dades de l'usuari { phoneNumber: '+34 643 276 493', bio: 'I like early Jazz, Bebop, Cool Jazz...', language: 'ca', country: 'ES' }	
Respostes			
Codi	Descripció		
200	Dades de l'usuari editades		
404	Usuari no trobat		
403	Permís denegat		
401	No autoritzat		

Pujar imatge de perfil

Petició			
Mètode	POST	URL	/users/{username}/picture
Accés	Autenticat com a l'usuari a modificar		
ContentType	multipart/form-data		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta	Nom de l'usuari a editar	
picture	Fitxer	Fitxer amb la imatge de perfil	
Respostes			
Codi	Descripció		
200	Confirmació i URL de la imatge desada al núvol		

404	Usuari no trobat
403	Permís denegat
401	No autoritzat

Album endpoints

Consultar col·lecció d'LPs d'un usuari

Petició			
Mètode	GET	URL	/users/{username}/lps
Accés	Autenticat. Si l'usuari autenticat no coincideix amb l'usuari a consultar, només es retornen els discos visibles al públic		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari propietari del LP	
Respostes			
Codi	Descripció		
200	Array d'LPs de l'usuari en format resum		
404	Usuari no trobat		
401	No autoritzat		

Afegir LP a la col·lecció de l'usuari

Petició			
Mètode	POST	URL	/users/{username}/lps
Accés	Autenticat com a propietari de la col·lecció		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari propietari del LP	
body	Cos	Dades del LP <pre>{ "title": "A Love Supreme", "label": "Impulse!", "releaseYear": 1965, "genre": "jazz", "condition": "good", "isPublic": "true", "review": { "rating": 5, "comment": "Obra mestre, indispensable a qualsevol col·lecció de Jazz" } }</pre>	
Respostes			
Codi	Descripció		
201	Dades del LP inserit		
400	Si l'usuari ja existeix o s'incompleix alguna validació retorna un missatge d'error		
401	No autoritzat		

Consultar detall LP

Petició			
Mètode	GET	URL	/users/{username}/lps/{id}
Accés	Autenticat. Si l'usuari autenticat no és el propietari del LP, només es retornen les dades si el disc és públic		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari propietari del LP	
Id	Ruta Obligatori	Id del LP	
Respostes			
Codi	Descripció		
200	Dades del LP		
403	Permís denegat		
404	LP no trobat		
401	No autoritzat		

Editar LP

Petició			
Mètode	PATCH	URL	/users/{username}/lps/{id}
Accés	Autenticat com a propietari del LP		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari a consultar	
Id	Ruta Obligatori	Id del LP	
body	Cos Obligatori	Dades a modificar del LP { "condition": "mint", "isPublic": false }	
Respostes			
Codi	Descripció		
200	Dades del LP editat		
403	Permís denegat		
404	LP no trobat		
401	No autoritzat		

Eliminar LP

Petició			
Mètode	DELETE	URL	/users/{username}/lps/{id}
Accés	Autenticat com a propietari del LP		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta. Obligatori	Nom de l'usuari propietari del LP	

Id	Ruta. Obligatori	Id del LP
Respostes		
Codi	Descripció	
200	Confirmació LP esborrat	
403	Permís denegat	
404	LP no trobat	
401	No autoritzat	

Recuperar dades externes album

Petició			
Mètode	GET	URL	/album/search
Accés	Autenticat		
Paràmetres			
Nom	Tipus	Descripció/exemple	
title	Query Obligatori	Nom del disc del qual es volen recuperar dades	
artist	Query Opcional	Nom de l'autor del disc	
Respostes			
Codi	Descripció		
200	Dades recuperades de l'àlbum		
404	Dades no trobades		
400	Títol obligatori		
401	No autoritzat		

Pujar portada disc

Petició			
Mètode	POST	URL	/lps/{id}/picture
Accés	Autenticat com a l'usuari propietari del LP		
ContentType	multipart/form-data		
Paràmetres			
Nom	Tipus	Descripció/exemple	
id	Ruta Obligatori	Id del LP	
cover	Fitxer	Fitxer amb la imatge de la portada	
url	FormData	També podem pujar una imatge allotjada a una url	
Respostes			
Codi	Descripció		
200	Confirmació i URL de la imatge desada al núvol		
404	Lp no trobat		
403	Permís denegat		
401	No autoritzat		

Market endpoints

Consultar LPs en venda

Petició			
Mètode	GET	URL	/market/listeditems
Accés	Autenticat.		
Paràmetres			
Nom	Tipus	Descripció/exemple	
limit	Query	Registres per pàgina	
sortBy	Query	Camp d'ordenació. [createdAt, title, artist, genre, wantedPrice]	
sortOrder	Query	Ordre [asc, desc]	
genre	Query	Gènere musical	
minPrice	Query	Preu mínim	
maxPrice	Query	Preu màxim	
page	Query	Núm. pàgina	
Respostes			
Codi	Descripció		
200	Array de discos en venda		
401	No autoritzat		

Posar LP en venda

Petició			
Mètode	POST	URL	/lps/{lpId}/listforsale
Accés	Autenticat com a propietari del LP		
ContentType	multipart/form-data		
Paràmetres			
Nom	Tipus	Descripció/exemple	
lpId	Query Obligatori	Id del LP que es vol posar en venda	
wantedPrice	formData Obligatori	Import pel qual es posa en venda el disc	
picture	Fitxer Opcional de 0 a 4 fitxers	Imatges de l'anunci	
Respostes			
Codi	Descripció		
200	Item en venda		
403	Permís denegat		
401	No autoritzat		

Retirar LP en venda

Petició			
Mètode	DELETE	URL	/market/listeditems/{itemId}
Accés	Autenticat com a propietari del LP		
Paràmetres			
Nom	Tipus	Descripció/exemple	
itemId	Ruta Obligatori	Id de l'item en venda	

Respostes	
Codi	Descripció
200	Item eliminar
403	Permís denegat
404	Item no trobat
401	No autoritzat

Fer oferta a un LP en venda

Petició		
Mètode	POST	URL /market/listeditems/{itemId}/offer
Accés	Autenticat	
ContentType	application/json	
Paràmetres		
Nom	Tipus	Descripció/exemple
itemId	Ruta Obligatori	Id de l'item en venda
body	Cos Obligatori	Import pel qual es fa la oferta {"suggestedPrice": 10.50}
Respostes		
Codi	Descripció	
201	Oferta creada	
403	Permís denegat	
404	Item no trobat	
401	No autoritzat	

Consultar Ofertes rebudes

Petició		
Mètode	GET	URL /market/offers/received
Accés	Autenticat	
Respostes		
Codi	Descripció	
200	Array d'ofertes rebudes de l'usuari autenticat	
401	No autoritzat	

Consultar Ofertes realitzades

Petició		
Mètode	GET	URL /market/offers/sended
Accés	Autenticat	
Respostes		
Codi	Descripció	
200	Array d'ofertes fetes per l'usuari autenticat	
401	No autoritzat	

Acceptar oferta rebuda

Petició		
Mètode	PATCH	URL /market/offers/{offerId}/accept
Accés	Autenticat com a l'usuari venedor de l'item en venda	
Paràmetres		

Nom	Tipus	Descripció/exemple
offerId	Ruta Obligatori	Id de la oferta que es vol acceptar
Respostes		
Codi	Descripció	
200	Oferta acceptada	
403	Permís denegat	
404	Oferta no trobada	

Rebutjar oferta rebuda

Petició			
Mètode	PATCH	URL	/market/offers/{offerId}/reject
Accés	Autenticat com a l'usuari venedor de l'ítem en venda		
Paràmetres			
Nom	Tipus	Descripció/exemple	
offerId	Ruta Obligatori	Id de la oferta que es vol rebutjar	
Respostes			
Codi	Descripció		
200	Oferta rebutjada		
403	Permís denegat		
404	Oferta no trobada		
401	No autoritzat		

Valorar transacció

Petició			
Mètode	POST	URL	/market/offers/{offerId}/rate
Accés	Autenticat com a l'usuari venedor o comprador de l'ítem en venda		
ContentType	application/json		
Paràmetres			
Nom	Tipus	Descripció/exemple	
offerId	Ruta Obligatori	Id de la oferta que ha originat la transacció	
body	Cos Obligatori	Valoració de la transacció {"rating": 3}	
Respostes			
Codi	Descripció		
200	Array d'ofertes rebudes per l'ítem en venda		
404	Oferta no trobada		
405	Descripció error		
401	No autoritzat		

Social endpoints

Consultar usuaris

Petició			
Mètode	GET	URL	/users
Accés	Autenticat		
Paràmetres			
Nom	Tipus	Descripció/exemple	
query	Query	Paràmetres de cerca	
Respostes			
Codi	Descripció		
200	Llistat d'usuaris de l'aplicació		
401	No autoritzat		

Consultar detall usuari

Petició			
Mètode	GET	URL	/users/{username}
Accés	Autenticat		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari a consultar	
Respostes			
Codi	Descripció		
200	Detall de l'usuari amb els usuaris que el segueixen i la seva col·lecció pública de LPs		
404	Usuari no trobat		
401	No autoritzat		

Seguir usuari

Petició			
Mètode	POST	URL	/users/{username}/follow
Accés	Autenticat		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari a seguir	
Respostes			
Codi	Descripció		
200	Seguint usuari		
404	Usuari no trobat		
401	No autoritzat		

Deixar de seguir usuari

Petició			
Mètode	DELETE	URL	/users/{username}/unfollow
Accés	Autenticat		
Paràmetres			
Nom	Tipus	Descripció/exemple	
username	Ruta Obligatori	Nom de l'usuari a seguir	
Respostes			
Codi	Descripció		
200	Deixant de seguir usuari		
404	Usuari no trobat		
401	No autoritzat		

Annex II. Manual d'instal·lació

Execució al núvol

L'aplicació s'ha desplegat a la plataforma cloud Heroku, això permet provar l'aplicació directament sense necessitat d'instal·lar cap programari. Per accedir-hi cal introduir la següent URL a un navegador web:

<https://phonocat.herokuapp.com/>

Convé ressaltar la versió gratuïta de heroku descarrega recursos quan porten un temps sense rebre tràfic, en aquest sentit, és possible que a l'accedir a l'aplicació aquesta tardi uns quants segons en carregar-se.

Execució local

També existeix la possibilitat d'executar l'aplicació a un entorn local.

El codi font es troba publicat al següent repositori de Github:

<https://github.com/llorenc-isern-tfg/phonocat>

Pel seu funcionament, el programari necessita que es creïn els següents fitxers .env i es configurin diverses variables d'entorn.

Fitxer phonocat/.env

```
EXEC_ENV = development
PORT = 5000
MONGO_URI = url de connexió a una instància de MongoDB

JWT_SECRET = clau secreta per a encriptar JWT
JWT_EXPIRATION_TIME = temps expiració token

GOOGLE_OAUTH_CLIENT_ID = clau propia google auth
GOOGLE_OAUTH_CLIENT_SECRET = secret propi google auth

DISCOGS_API_URL = https://api.discogs.com
DISCOGS_API_KEY = clau propia discogs
DISCOGS_API_SECRET = secret propi google auth

CLOUDINARY_CLOUD_NAME = identificador cloudinary
CLOUDINARY_API_KEY = clau cloudinary
CLOUDINARY_API_SECRET = secret cloudinary
```

Fitxer phonocat/frontend/.env

```
REACT_APP_LASTFM_URL = https://ws.audioscrobbler.com/2.0
REACT_APP_LASTFM_KEY = key lastfm
REACT_APP_GOOGLE_CLIENT_ID = cliend id google auth
```

A l'entrega del TFG ja s'annexa el codi amb la inclusió aquests fitxers de configuració ".env". No s'han pujat al repositori de github pel fet de que contenen claus privades.

Per executar l'aplicació cal tenir instal·lat node 15 i npm 6 o superior i executar la comanda **npm install** als directoris 'phonocat' i 'phonocat/frontend'.

A continuació, des del directori arrel 'phonocat', cal executar la comanda:
npm run dev

Aquesta instrucció posa en marxa simultàniament el servidor node amb l'api backend i l'aplicació client amb react.

Per defecte, l'aplicació apunta a una instància de MongoDB Atlas compartida amb producció, no obstant això es pot canviar per una BDD local. En aquest cas, cal modificar el paràmetre MONGO_URI del fitxer phonocat/.env i engegar la instància local de base de dades. Les taules i altres objectes de BDD es generen automàticament a l'iniciar el servidor de l'aplicació.

Generació massiva d'usuaris

L'aplicació inclou un script de generació d'usuaris amb les seves dades de perfil bàsiques amb propòsit de prova. Aquesta generació es fa mitjançant la llibreria faker, que proporciona dades aleatòries de diversos tipus.

Per executar-ho només cal executar la següent instrucció al directori arrel de l'aplicació i tenir la instància de base de dades engegada:

```
npm run seed-users <num>
```

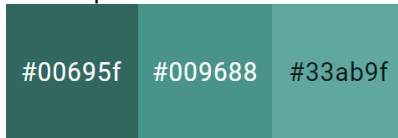
<num> fa referència al número d'usuaris que es volen generar.

Annex III. Llibre d'estil

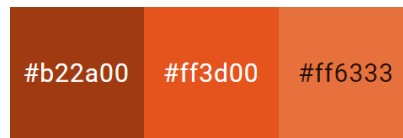
El present document marca les pautes essencials que s'han seguit pel disseny gràfic de l'aplicació Phonocat.

Paleta de colors

Color primari



Color secundari



Aquesta paleta de colors s'ha inspirat en la etiqueta que feia servir el cel-lebre segell discogràfic Atlantic Records entre els anys 1968 i 1975.



A.III.1: Galeta Atlantic Records



A.III.2: Favlcon Phonocat

Tipografia

Font Roboto:

<https://fonts.google.com/specimen/Roboto>

h1. Phonocat

h2. Phonocat

h3. Phonocat

h4. Phonocat

h5. Phonocat

h6. Phonocat

subtitle1. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quos blanditiis tenetur

subtitle2. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quos blanditiis tenetur

body1. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quos blanditiis tenetur unde suscipit, quam beatae rerum inventore consectetur, neque doloribus, cupiditate numquam dignissimos laborum fugiat deleniti? Eum quasi quidem quibusdam.

body2. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quos blanditiis tenetur unde suscipit, quam beatae rerum inventore consectetur, neque doloribus, cupiditate numquam dignissimos laborum fugiat deleniti? Eum quasi quidem quibusdam.

BUTTON TEXT

caption text

OVERLINE TEXT

Botons

