

Universitat de Girona
Escola Politècnica Superior

Grau en Enginyeria Informàtica

PROJECTE FINAL DE GRAU

**Learnum: Aplicació Multiplataforma d'una
xarxa social d'estudiants**

Autor:
Marc Torres Vilagut

Tutors:
Dr. Soler Maso, Josep

MEMÒRIA

Convocatòria:
Setembre 2023

Departament:
Informàtica, Matemàtica Aplicada i Estadística

Projecte: Projecte Final de Grau
Document: Memòria
Títol: Learnum: Aplicació Multiplataforma d'una xarxa social d'estudiants
Autor: Marc Torres Vilagut
Data: Setembre 2023

Estudi:
Grau en Enginyeria Informàtica
Universitat de Girona

Supervisor 1:
Dr. Soler Maso, Josep
Universitat de Girona
Email: josep.soler@udg.edu
Web: Link

Agraïments

Al meu tutor Josep Soler, per la seva ajuda amb aquest projecte i com a tutor de les pràctiques a l'empresa on actualment treballo.

Al professorat de la universitat per tot el que m'han ensenyat aquests anys.

A DoceoSoftware, l'empresa on treballo, per la flexibilitat que m'han donat per ajudar-me a acabar els estudis.

Als amics i companys que m'han ajudat en etapes difícils de la carrera.

A la meua família per tot el suport que m'han donat i els esforços que han fet al llarg de la carrera per poder permetre'm estar on estic avui.

I en memòria del meu pare, qui va ser el meu guia en aquesta vida i sempre em va aconsellar en moments de dubtes.

Índex

Agraïments	iii
Índex de figures	ix
Índex de taules	xiii
1 Introducció i objectius del projecte	1
1.1 Idea darrere Learnum	1
1.2 Motivacions i Propòsit	1
1.3 Objectius	2
2 Estudi de viabilitat	4
2.1 Pressupost i recursos humans	4
2.2 Viabilitat	5
3 Metodologia	6
3.1 Filosofia Agile	6
3.2 Metodologia Kanban	6
3.3 Beneficis de Kanban	7
3.4 Ús de Kanban	7
4 Planificació	9
4.1 Mètode de planificació	9
4.2 Flux de treball i tasques	10
4.2.1 Fase d'estudi i preparació	10
4.2.2 Fase de desenvolupament	10
4.2.3 Fase final	11
5 Marc de treball i conceptes previs	12
5.1 Entorns de desenvolupament integrats	12
5.2 Emuladors	13
5.3 NodeJS i Node Package Manager	13
5.4 API Rest	14
5.5 React Native	15
6 Requisits del sistema	16
6.1 Requisits de les aplicacions	16
6.2 Requisits dels dispositius	17
7 Estudis i decisions	18
7.1 Visual Studio Code	18
7.1.1 Extensions	19

7.2	API Rest	20
7.2.1	XAMPP	20
7.2.2	Express	21
7.2.3	Sequelize	21
7.2.4	Postman	22
7.3	React	23
7.3.1	DOM Virtual	23
7.3.2	JSX	24
7.3.3	Components	25
	Propietats	25
7.3.4	<i>State</i>	26
7.3.5	<i>Hooks</i>	26
	<i>useState</i>	26
	<i>useEffect</i>	27
	<i>Custom hooks</i>	28
7.4	React Native	28
7.4.1	Components	29
7.5	Emuladors: Android Studio i Xcode	29
7.6	Expo	30
7.6.1	Expo router	30
7.6.2	Expo Go	30
8	Anàlisi i disseny del sistema	32
8.1	Actors i mòduls	32
8.2	Anàlisi del sistema	33
8.2.1	Model de dades	33
8.2.2	Funcionalitats	34
	Llistats de funcionalitats	34
	Diagrama de casos d'ús	35
8.3	Disseny	38
8.3.1	Diagrama relacional de la base de dades	38
	Taula model <i>User</i>	39
	Taula model <i>UserFollower</i>	40
	Taula model <i>Forum</i>	40
	Taula model <i>ForumMember</i>	40
	Taula model <i>ForumPost</i>	41
	Taula model <i>PostLike</i>	42
	Taula model <i>PostMedia</i>	42
	Taula model <i>PostComment</i>	43
	Taula model <i>CommentLike</i>	43
	Taula model <i>Task</i>	44
	Taula model <i>Chat</i>	44
	Taula model <i>ChatsMember</i>	45
	Taula model <i>Message</i>	45
	Taula model <i>MessageMedia</i>	46
8.3.2	Funcionalitats: Fitxes de cas d'ús i diagrames d'activitat	46
	Registrar nou compte d'usuari	47
	Iniciar sessió	49
	Tancar sessió	50
	Visualitzar perfil d'usuari	51
	Editar dades del perfil personal	52

	Seguir usuari	54
	Deixar de seguir a un usuari	55
	Crear nou fòrum	56
	Afegir publicació a un fòrum	58
	Visualitzar llistat de publicacions	60
	Afegir interacció de “m’agrada” a una publicació	61
	Eliminar interacció de “m’agrada” a una publicació	62
	Afegir comentari a una publicació	63
	Visualitzar llistat de comentaris d’una publicació	64
	Afegir interacció de “m’agrada” a un comentari	65
	Eliminar interacció de “m’agrada” a un comentari	66
	Crear tasca	67
	Visualitzar llistat de tasques	68
	Editar tasca	69
	Marcar tasca com “completada”	71
	Desmarcar tasca com “completada”	72
	Eliminar tasca	73
	Obrir sales de xat de missatgeria directa amb un usuari seguit	74
	Crear sala de xat de grup	75
	Afegir usuari seguit a xats de grup	76
	Enviar missatge a un xat	77
	Visualitzar llistat de missatges d’un xat	78
	Eliminar publicació	79
	Eliminar comentari	80
	Editar dades del xat de grup	82
	Eliminar missatge	84
8.3.3	Endpoints API	86
8.3.4	Interfície d’usuari	86
9	Implementació i proves	91
9.1	Configuració bàsica dels projectes	91
9.1.1	Node.js i npm	91
9.1.2	GitHub. Repositoris	92
9.1.3	Visual Studio Code. Entorns de treball	93
9.2	API	93
9.2.1	Configuració	93
	Variables d’entorn	93
	Connexió amb la base de dades	94
	Preparació <i>endpoints</i>	95
	Inicialització del procés del servidor	96
9.2.2	Implementació dels models de dades	97
	Models de dades	97
	Associacions	100
	<i>Hooks</i>	101
9.2.3	Proves	102
9.2.4	Problemes i solucions	104
	Autenticació d’usuaris	104
	Obtenció d’usuaris que no siguin membres d’un xat	105
9.3	Aplicació per dispositius mòbils	106
9.3.1	Configuració	106
	Expo	106

Expo Router	107
9.3.2 Proves	108
9.3.3 Problemes i solucions	109
Crides a l'API	109
Mantenir sessió iniciada	109
Paginació de peticions	111
<i>Splash Screen</i>	112
10 Implantació i resultats	114
10.1 Apartat legal de l'aplicació	114
10.2 <i>Splash screen</i>	115
10.3 Accés a la plataforma	115
10.4 Pantalla d'inici de l'aplicació	120
10.5 Perfil d'usuari	122
10.5.1 Perfil personal	122
10.5.2 Perfil d'un altre usuari	125
10.5.3 Visualitzar seguidor i usuaris seguits	127
10.6 Pantalla de creació de publicacions	128
10.7 Pantalla de creació de fòrums	129
10.8 Pantalla d'una publicació	130
10.9 Pantalla d'un fòrum	132
10.10 Pantalla de tasques	135
10.11 Pantalla de xats	140
10.12 Pantalla d'un xat	142
10.12.1 Pantalla d'un xat de grup	144
11 Conclusions	147
12 Treball futur	149
Bibliografia	150
A Annex	156
A.1 Repositoris GitHub	156
A.2 Estructura de fitxers del projecte LearnumAPI	157
A.3 Estructura de fitxers del projecte LearnumMobile	158
A.4 Models de dades de l'API definits amb Sequelize	159
A.4.1 Mòdul d' Usuaris	159
User	159
UserFollower	159
A.4.2 Mòdul de Fòrums	160
Fòrum	160
ForumMember	160
ForumPost	161
PostLike	161
PostMedia	161
PostComment	162
CommentLike	162
A.4.3 Mòdul de Tasques	163
Task	163
A.4.4 Mòdul de Xats	163
Chat	163

	ChatMember	164
	Message	164
	MessageMedia	164
A.5	Documentació apartat legal	166
A.5.1	Avís Legal (<i>Legal Notice</i>)	166
A.5.2	Política de Privacitat (<i>Privacy Policy</i>)	166
A.5.3	Termes i condicions d'ús (<i>Terms and Conditions of Use</i>)	169

Índex de figures

5.2	Logo de l'IDE Visual Studio Code [3]	13
7.1	Captura de pantalla de l'entorn de desenvolupament de Visual Studio Code	19
7.2	Captura de pantalla del panell d'extensions de l'entorn de desenvolupament de Visual Studio Code	19
7.3	Captura de pantalla del panell de control de XAMPP	20
7.4	Captura de pantalla del panell de control local de PHPMyAdmin	21
7.5	Captura de pantalla del client REST Postman	23
7.6	Diagrama explicatiu del funcionament del DOM per actualitzar els continguts visibles de l'aplicació web	24
7.7	Diagrama explicatiu del funcionament del Virtual DOM per actualitzar els continguts visibles de l'aplicació web	24
7.8	Captura de pantalla d'instàncies dels emuladors per Android (dreta) i iOS (esquerra)	30
7.9	Captura de pantalla de l'aplicació Expo Go	31
8.1	Diagrama del model conceptual de dades en el que es defineixen les associacions de les dades que formaran part de la base de dades	33
8.2	Diagrama casos d'ús per l'actor: Usuari No Registrat	35
8.3	Diagrama casos d'ús per l'actor: Usuari Registrat	36
8.4	Diagrama casos d'ús per l'actor: Usuari Moderador d'un Fòrum	37
8.5	Diagrama casos d'ús per l'actor: Usuari Administrador d'un Xat	37
8.6	Diagrama relacional del model físic de dades per les entitats de la base de dades	38
8.7	Diagrama d'activitat del cas d'ús per registrar un compte d'usuari	48
8.8	Diagrama d'activitat del cas d'ús per l'inici de sessió	49
8.9	Diagrama d'activitat del cas d'ús per tancar la sessió	50
8.10	Diagrama d'activitat del cas d'ús per visualitzar el perfil d'un usuari	51
8.11	per editar les dades del perfil personal de l'usuari que ha iniciat sessió	53
8.12	Diagrama d'activitat del cas d'ús seguir a un usuari	54
8.13	Diagrama d'activitat del cas d'ús per deixar de seguir a un usuari	55
8.14	Diagrama d'activitat del cas d'ús per crear un nou fòrum	57
8.15	Diagrama d'activitat del cas d'ús per crear una nova publicació i afegir-la a un fòrum	59
8.16	Diagrama d'activitat del cas d'ús per la visualització de les publicacions del sistema	60
8.17	Diagrama d'activitat del cas d'ús per reaccionar a una publicació amb un "m'agrada"	61
8.18	Diagrama d'activitat del cas d'ús per eliminar la reacció de "m'agrada" d'una publicació	62

8.19	Diagrama d'activitat del cas d'ús per afegir un comentari a una publicació	63
8.20	Diagrama d'activitat del cas d'ús per la visualització dels comentaris d'una publicació	64
8.21	Diagrama d'activitat del cas d'ús per reaccionar a un comentari amb un "m'agrada"	65
8.22	Diagrama d'activitat del cas d'ús per eliminar la reacció de "m'agrada" d'un comentari	66
8.23	Diagrama d'activitat del cas d'ús per la creació d'una nova tasca	67
8.24	Diagrama d'activitat del cas d'ús per la visualització de les tasques de l'usuari que ha iniciat sessió	68
8.25	Diagrama d'activitat del cas d'ús per l'edició dels continguts d'una tasca	70
8.26	Diagrama d'activitat del cas d'ús per marcar l'estat d'una tasca com a "completada"	71
8.27	Diagrama d'activitat del cas d'ús per desmarcar l'estat d'una tasca de "completada" a "no completada"	72
8.28	Diagrama d'activitat del cas d'ús per l'eliminació d'una tasca de l'usuari	73
8.29	Diagrama d'activitat del cas d'ús per accedir al canal de missatgeria directa de l'usuari que ha iniciat sessió amb un altre usuari.	74
8.30	Diagrama d'activitat del cas d'ús per la creació d'un nou xat de grup .	75
8.31	Diagrama d'activitat del cas d'ús per afegir un nou membre a un xat de grup	76
8.32	Diagrama d'activitat del cas d'ús per enviar un missatge per un canal de xat	77
8.33	Diagrama d'activitat del cas d'ús per la visualització dels missatges d'un xat	78
8.34	Diagrama d'activitat del cas d'ús per l'eliminació d'una publicació per part d'un moderador d'un fòrum	79
8.35	Diagrama d'activitat del cas d'ús per l'eliminació d'un comentari per part d'un moderador d'un fòrum	81
8.36	Diagrama d'activitat del cas d'ús per l'edició de les dades relatives a un xat	83
8.37	Diagrama d'activitat del cas d'ús per l'eliminació d'un missatge per part d'un administrador d'un xat.	85
8.38	Recreació del dibuix conceptual per la pàgina d'inici de sessió de l'aplicació <i>Learnum</i> per dispositius mòbils	87
8.39	Recreació del dibuix conceptual per la pàgina per enregistrar un nou compte d'usuari per l'aplicació <i>Learnum</i> per dispositius mòbils	87
8.40	Recreació del dibuix conceptual per la pàgina d'inici de l'aplicació <i>Learnum</i> per dispositius mòbils	88
8.41	Recreació del dibuix conceptual per una pàgina d'una sala de xat de l'aplicació <i>Learnum</i> per dispositius mòbils	88
8.42	Icona per l'aplicació <i>Learnum</i>	89
8.43	<i>Favicon</i> per l'aplicació <i>Learnum</i>	89
8.44	Icona adaptativa per l'aplicació <i>Learnum</i>	89
8.45	Imatge per la <i>Splash Screen</i>	90
9.1	Instal·lador de <i>NodeJS</i> i <i>npm</i> per <i>Windows</i>	91
9.2	Captura de pantalla de la pàgina de GitHub amb els repositoris creats per cada part del projecte.	92

9.3	Captura de pantalla de la clonació del repositori <i>LearnumMobile</i> pel desenvolupament de la plataforma Learnum per dispositius mòbils.	92
9.4	Captura de pantalla dels continguts del fitxer <i>.env</i> de l'entorn de desenvolupament per l'API REST de la plataforma Learnum	94
9.5	Captura de pantalla dels continguts del fitxer <i>app.js</i> del projecte d'API REST per la plataforma Learnum	95
9.6	Captura de pantalla de la línia d'ordres amb l'execució del procés de l'API REST de la plataforma Learnum amb el <i>script "dev"</i>	97
9.7	Captura de pantalla del codi de definició del model de dades d' <i>Usuaris</i> de l'API	98
9.8	Captura de pantalla del codi de definició del model de dades dels <i>Membres d'un Xat</i> de l'API	100
9.9	Captura de pantalla de la implementació de l'associació entre el model d'usuari (User) i el model de xat (Chat) per afegir la relació <i>membre d'un xat</i>	101
9.10	Captura de pantalla de la implementació d'un <i>hook</i> pel model Post-Like per actualitzar el comptador de <i>likes</i> d'una publicació	102
9.11	Captura de pantalla de POSTMAN executant la petició a l'API per l'inici de sessió de l'usuari administrador	103
9.12	Captura de pantalla de POSTMAN executant la petició a l'API per recuperar els usuaris del servidor	103
9.13	Captura de pantalla de POSTMAN executant la petició a l'API per recuperar els usuaris als quals l'usuari administrador segueix	104
9.14	Captura de pantalla de l'execució de la comanda <i>npx create-expo-app</i> per la creació d'un projecte d'Expo amb l'opció de plantilla buida per començar des de zero.	106
9.15	Captures de pantalla del directori de les pantalles de l'aplicació de vista per mòbils de Learnum	107
9.16	Captura de pantalla de l'execució del procés de l'aplicació Expo per la vista per dispositius mòbils de Learnum	109
9.17	Captura de pantalla de la inicialització de l' <i>ApiManager</i> , una instància del gestor de peticions d' <i>axios</i> per efectuar crides a l'API de Learnum	110
9.18	Captura de pantalla del mètode per comprovar si un usuari disposa d'una sessió iniciada al dispositiu amb un token JWT.	110
9.19	Captura de pantalla del <i>custom hook</i> <i>usePagination</i> per afegir el comportament de paginació a les llistes de React Native	112
10.1	Captures de pantalla de les seccions amb els documents relatius a la legislació per la plataforma Learnum	114
10.2	Captura de pantalla de la <i>Splash Screen</i>	115
10.3	Captura de pantalla de la pantalla d'inici de sessió de l'aplicació Learnum per mòbils	116
10.4	Captura de pantalla de la pantalla de registre d'un nou compte d'usuari	117
10.5	Captura de pantalla del control d'error al formulari de la pantalla d'inici de sessió.	118
10.6	Captura de pantalla del control d'error al formulari de la pantalla de registre d'un nou compte d'usuari	119
10.7	Captura de pantalla de la pantalla principal de l'aplicació Learnum per mòbils (<i>Feed</i>)	120
10.8	Captura de pantalla de la pantalla principal mostren el comportament d'afegir reaccions de "m'agrada" a una publicació	121

10.9	Captura de pantalla del perfil personal d'usuari	122
10.10	Captura de pantalla de la pàgina d'edició de les dades de l'usuari	123
10.11	Captura de pantalla del control d'errors als formularis d'edició de les dades del perfil de l'usuari	124
10.12	Captura de pantalla del perfil d'un usuari	125
10.13	Captura de pantalla del perfil d'un usuari	126
10.14	Captura de pantalla dels llistats de seguidors o d'usuaris seguits d'un usuari	127
10.15	Captura de pantalla de la pàgina de creació de publicacions	128
10.16	Captura de pantalla de la pàgina de creació de fòrums	129
10.17	Captura de pantalla de la pàgina d'una publicació que s'ha reaccionat com que l'hi agrada a l'usuari	130
10.18	Captura de pantalla del comportament d'afegir un comentari a una publicació a la pàgina d'aquesta.	131
10.19	Captura de pantalla de la pàgina d'un fòrum	132
10.20	Captura de pantalla de la pàgina d'un fòrum amb el comportament per unir-se com a membre d'aquest	133
10.21	Captura de pantalla del llistat de membres d'un fòrum.	134
10.22	Captura de pantalla de la llista de tasques de l'usuari	135
10.23	Captures de pantalla de la pàgina de creació de tasques	136
10.24	Captura de pantalla de pàgina d'edició d'una tasca	137
10.25	Captures de pantalla del comportament d'assignació de l'estat "completada" a una tasca del llistat de tasques	138
10.26	Captura de pantalla de l'avís de confirmació de l'usuari per la funcionalitat per eliminar tasques	139
10.27	Captura de pantalla del llistat de xats de l'usuari	140
10.28	Captures de pantalla del comportament de la creació d'un nou xat de grup	141
10.29	Captura de pantalla d'un xat de missatgeria directa	142
10.30	Captura de pantalla del comportament d'enviar un missatge per un canal de xat.	143
10.31	Captura de pantalla de la sala d'un xat de grup	144
10.32	Captura de pantalla del llistat de membres d'un xat de grup	145
10.33	Captures de pantalla del comportament per afegir nous membres a un xat de grup	146
A.1	Captura de pantalla de l'estructura de carpetes utilitzada per l'aplicació del servidor API REST del projecte <i>Learnum</i>	157
A.2	Captura de pantalla de l'estructura de carpetes utilitzada per l'aplicació de la vista per dispositius mòbils del projecte <i>Learnum</i>	158

Índex de taules

8.1	Taula de funcionalitats requerides per l'actor <i>usuari no registrat</i>	34
8.2	Taula de funcionalitats requerides per l'actor <i>usuari registrat</i>	35
8.3	Taula de funcionalitats requerides per l'actor <i>usuari moderador de fòrum</i>	35
8.4	Taula de funcionalitats requerides per l'actor <i>usuari administrador de xat de grup</i>	35
8.5	Descripció dels camps de la taula de la base de dades pel model de dades <i>User</i>	39
8.6	Descripció dels camps de la taula de la base de dades pel model de dades <i>UserFollower</i>	40
8.7	Descripció dels camps de la taula de la base de dades pel model de dades <i>Forum</i>	40
8.8	Descripció dels camps de la taula de la base de dades pel model de dades <i>ForumMember</i>	41
8.9	Descripció dels camps de la taula de la base de dades pel model de dades <i>ForumPost</i>	41
8.10	Descripció dels camps de la taula de la base de dades pel model de dades <i>PostLike</i>	42
8.11	Descripció dels camps de la taula de la base de dades pel model de dades <i>PostMedia</i>	42
8.12	Descripció dels camps de la taula de la base de dades pel model de dades <i>PostComment</i>	43
8.13	Descripció dels camps de la taula de la base de dades pel model de dades <i>CommentLike</i>	43
8.14	Descripció dels camps de la taula de la base de dades pel model de dades <i>Task</i>	44
8.15	Descripció dels camps de la taula de la base de dades pel model de dades <i>Chat</i>	44
8.16	Descripció dels camps de la taula de la base de dades pel model de dades <i>ChatsMember</i>	45
8.17	Descripció dels camps de la taula de la base de dades pel model de dades <i>Message</i>	45
8.18	Descripció dels camps de la taula de la base de dades pel model de dades <i>MessageMedia</i>	46
8.19	Full del cas d'ús per registrar un compte d'usuari	47
8.20	Full del cas d'ús per l'inici de sessió	49
8.21	Full del cas d'ús per tancar la sessió	50
8.22	Full del cas d'ús per visualitzar el perfil d'un usuari	51
8.23	Full del cas d'ús per editar les dades del perfil personal de l'usuari que ha iniciat sessió	52
8.24	Full del cas d'ús per seguir a un usuari	54
8.25	Full del cas d'ús per deixar de seguir a un usuari	55

8.26	Full del cas d'ús per crear un nou fòrum	56
8.27	Full del cas d'ús per crear una nova publicació i afegir-la a un fòrum	58
8.28	Full del cas d'ús per la visualització de les publicacions del sistema	60
8.29	Full del cas d'ús per reaccionar a una publicació amb un "m'agrada"	61
8.30	Full del cas d'ús per eliminar la reacció de "m'agrada" d'una publicació	62
8.31	Full del cas d'ús per afegir un comentari a una publicació	63
8.32	Full del cas d'ús per la visualització dels comentaris d'una publicació	64
8.33	Full del cas d'ús per reaccionar a un comentari amb un "m'agrada"	65
8.34	Full del cas d'ús per eliminar la reacció de "m'agrada" d'un comentari	66
8.35	Full del cas d'ús per la creació d'una nova tasca	67
8.36	Full del cas d'ús per la visualització de les tasques de l'usuari que ha iniciat sessió	68
8.37	Full del cas d'ús per l'edició dels continguts d'una tasca	69
8.38	Full del cas d'ús per marcar l'estat d'una tasca com a "completada"	71
8.39	Full del cas d'ús per desmarcar l'estat d'una tasca de "completada" a "no completada"	72
8.40	Full del cas d'ús per l'eliminació d'una tasca de l'usuari	73
8.41	Full del cas d'ús per accedir al canal de missatgeria directa de l'usuari que ha iniciat sessió amb un altre usuari	74
8.42	Full del cas d'ús per la creació d'un nou xat de grup	75
8.43	Full del cas d'ús per afegir un nou membre a un xat de grup	76
8.44	Full del cas d'ús per enviar un missatge per un canal de xat	77
8.45	Full del cas d'ús per la visualització dels missatges d'un xat	78
8.46	Full del cas d'ús per l'eliminació d'una publicació per part d'un moderador d'un fòrum	79
8.47	Full del cas d'ús per l'eliminació d'un comentari per part d'un moderador d'un fòrum	80
8.48	Full del cas d'ús per l'edició de les dades relatives a un xat	82
8.49	Full del cas d'ús per l'eliminació d'un missatge per part d'un administrador d'un xat	84

Capítol 1

Introducció i objectius del projecte

Aquest primer capítol servirà com a presentació de la idea darrere de les aplicacions que es desenvoluparan en aquest projecte, així com les motivacions i objectius que s'intentaran assolir per aquestes.

1.1 Idea darrere Learnum

Molts estudiants ens trobem amb la necessitat d'haver de contactar amb algú per resoldre dubtes i problemes en diverses ocasions. Per resoldre les nostres qüestions tendim a utilitzar diferents eines de comunicació que trobem a la disposició de tothom en qualsevol moment del dia. Entre aquestes eines podem citar: - els correus electrònics, que permeten una comunicació directa i individual, per exemple entre un estudiant i un professor - les aplicacions de missatgeria instantània, que permeten als estudiants difondre els seus dubtes a tot un grup d'estudiants que es poden trobar amb el mateix tipus de traves o solucions per aquestes - i finalment disposem dels fòrums, plataformes que ens permeten interactuar amb un abast més gran d'usuaris a Internet que poden tenir els mateixos dubtes.

Entre les eines que esmentem podem observar la potència de la qual disposen els fòrums a l'hora d'establir aquest tipus de comunicacions. És per aquest motiu que neix l'interès per aquests i la motivació per presentar una nova proposta de fòrum a l'abast dels estudiants: **Learnum**.

1.2 Motivacions i Propòsit

Learnum, com ja s'ha esmentat, està motivada per la potència de la qual disposen els fòrums com a via de comunicació a l'hora d'establir converses, debats i resolucions de dubtes per part dels seus usuaris. Així mateix, també neix d'una motivació pel desenvolupament d'aquest projecte en plataformes com *Reddit*, que presenten una solució de fòrum fusionat amb una xarxa social que dota d'una interacció més atractiva a un gran grup d'usuaris.

A més de les motivacions relatives a l'admiració per aquestes plataformes, també existeix un altre motiu darrere d'aquesta proposta: el desenvolupament d'aplicacions multiplataforma. La primera pregunta que pot sorgir podria ser "per què tecnologies multiplataforma?". Tot i que l'opció més simple per posar a disposició de tothom la plataforma de Learnum seria la d'una plataforma web, existeix un interès

en l'aprenentatge de nous coneixements sobre aquestes tecnologies que actualment es troben liderant el mercat.

En aquest projecte, s'espera poder desenvolupar dues aplicacions multiplataforma: una que ens permeti disposar de la plataforma per dispositius mòbils, aprenent el procés de desenvolupar amb React Native com a Framework de treball i una altra per dispositius d'escriptori, aprenent a utilitzar també el Framework d'Electron.js.

El motiu principal per la utilització d'aquests dos Frameworks en comptes d'altres que ofereix el mercat és la capacitat de desenvolupar amb JavaScript, un llenguatge de programació pel qual dispeno de més experiència per tal de facilitar l'accés i reduir la corba d'aprenentatge que suposaria l'estudi d'altres tecnologies de les quals no dispeno de cap mena d'experiència prèvia.

Aleshores, el propòsit de Learnum és ser una plataforma on estudiants, professors o qualsevol usuari pugui aprendre sobre temes concrets, interactuar amb altres usuaris per debatre temes d'interès, dubtes, problemes o simplement xatejar, afegint també funcionalitats relatives a les xarxes socials com les interaccions amb publicacions.

1.3 Objectius

Ara que ja dispenem de la idea general darrere del projecte, l'objectiu d'aquest TFG serà desenvolupar l'anàlisi, disseny i implementació d'aquestes dues aplicacions que formaran el projecte Learnum.

Els principals requisits que la plataforma intenta assolir són:

- Creació d'un compte d'usuari bàsica que permeti la utilització de l'aplicació.
- Permetre iniciar sessió i sortir d'aquesta en qualsevol moment.
- Permetre la modificació de les dades personals dels usuaris.
- Creació de fòrums.
- Permetre als usuaris unir-se com a membres dels fòrums.
- Creació de publicacions pels fòrums disponibles.
- Permetre als usuaris comentar en les publicacions.
- Permetre donar interaccions tant a les publicacions com als comentaris dels fòrums amb *likes*.
- Permetre a usuaris seguir a altres usuaris.
- Permetre als usuaris establir un canal de comunicació de missatgeria directa amb els usuaris que segueixen.
- Permetre als usuaris crear sales de xats grupals.

- Permetre als usuaris que siguin membres d'un grup, afegir a nous usuaris que segueixen al grup.
- Enviar missatges tant pels xats de missatgeria directa com els grupals.
- Creació de tasques.
- Visualització de les tasques dels usuaris.
- Permetre donar permisos de moderador a membres d'un fòrum.
- Permetre als membres moderadors d'un fòrum eliminar publicacions.
- Permetre als membres moderadors d'un fòrum eliminar comentaris.
- Permetre donar permisos d'administrador a membres d'un xat grupal.
- Permetre als administradors d'un xat grupal eliminar missatges.

Originalment, es varen establir més funcionalitats per les aplicacions. Tot i que no han estat descartades en el desenvolupament de l'aplicació, aquestes han estat deixades de banda com a objectius secundaris. Un cop s'acabi la implementació de les funcionalitats principals del projecte es donarà pas a la integració d'aquestes funcionalitats secundàries. Les funcionalitats són:

- Administració del calendari dels usuaris.
- Creació i administració dels apunts dels usuaris.
- Creació i administració d'esdeveniments dels usuaris.
- Creació i administració d'enquestes públiques.
- Creació i administració de múltiples llistes de tasques.

Capítol 2

Estudi de viabilitat

Abans de començar a desenvolupar un projecte a gran escala s'ha d'estudiar la viabilitat tant econòmica com tecnològica. En aquest capítol es farà l'estudi simulant el pressupost relatiu a aquest projecte, i la viabilitat de completar aquest.

2.1 Pressupost i recursos humans

Per fer el pressupost de la plataforma tindrem en consideració diferents punts: les eines utilitzades pel desenvolupament, el temps estimat que es requerirà pel desenvolupament de la plataforma, i finalment la mà d'obra requerida pel projecte.

Com s'ha comentat, començarem primer amb les eines. Afortunadament, no requerim cap entorn de desenvolupament (IDE) específic, per tant, podem emprar l'IDE gratuït *Visual Studio Code*, així mateix també haurem de fer ús de *Node.js* que també està sotmesa a la mateixa llicència que *Visual Studio Code*, la llicència MIT, i en conseqüència no ens suposa cap mena d'augment al pressupost, així com l'ús dels mòduls necessaris obtinguts a partir del *Node Package Manager* o *npm* que disposin d'una llicència d'ús gratuït. Per altra banda, s'espera que s'hagin d'utilitzar tecnologies de serveis externs com *Firebase* pel desenvolupament de parts específiques del projecte que, si bé inicialment amb una base d'usuaris reduïda pot no suposar un increment al pressupost, a mesura que ens trobem amb un augment en el nombre d'usuaris de les aplicacions s'hauran de considerar les quotes que aquestes tecnologies presentin com a part del pressupost del projecte.

Movent-nos al següent punt, es pot fer una estimació amb el temps dedicat al projecte del temps aproximat que requeriria el projecte per la seva finalització. Actualment, s'han dedicat al voltant d'unes 300 hores de desenvolupament pel desenvolupament del servei API Rest i la vista per dispositius mòbils. D'aquestes hores, unes 30 hores han estat dedicades a l'estudi de les eines, 20 hores en el disseny i 50 hores al desenvolupament del servei API Rest, 170 al desenvolupament de l'aplicació per dispositius mòbils i 30 hores a proves de correctesa de les noves funcionalitats. Tenint en consideració aquesta informació podem fer una estimació del temps que restaria poder completar el projecte amb totes les seves parts entre 450 i 500 hores.

Aquest projecte està desenvolupat per una persona, per tant, el pressupost serà relatiu a aquest nombre de persones com a recursos humans. La plataforma que es desenvolupa en aquest projecte requereix programadors full stack, utilitzarem aleshores el salari mitjà d'un programador full stack junior (entre 15 € a 20 € l'hora) per fer el càlcul del cost total amb l'estimació d'hores establerta anteriorment. Com a

estimació aproximada el cost total del projecte es trobaria en el rang de 7500 € a 9000 €, amb una mitjana de 8250 €, juntament amb les possibles quotes de serveis externs en cas que la quantitat d'usuaris augmenti significativament.

2.2 Viabilitat

A l'hora de plantejar la viabilitat del projecte, hem de considerar les tres parts que constitueixen el projecte de les que hem estat parlant: el servei API Rest, i les aplicacions que serviran de vistes per la plataforma per dispositius mòbils i dispositius d'escriptori. Tecnològicament parlant, en el mercat hi ha moltes plataformes d'aquest tipus i els entorns en què treballaré estan àmpliament documentats, amb la qual cosa la seva viabilitat està garantida.

A causa de les dimensions del projecte i la reduïda mà d'obra, s'espera no poder completar el 100% dels objectius funcionals del projecte i, per tant, es farà una divisió dels recursos per poder presentar el màxim possible com a resultat parcial al final d'aquest projecte.

Capítol 3

Metodologia

Les metodologies són pràctiques que permeten la correcta organització del desenvolupament d'un projecte. En aquest capítol es presenten algunes metodologies existents de l'àmbit Agile utilitzades en diferents projectes reals, seguida de l'explicació de la metodologia escollida i els avantatges que presenta a l'hora de desenvolupar aquest projecte.

3.1 Filosofia Agile

Per tal d'escollir una metodologia pel desenvolupament del projecte, es varen tenir en consideració els valors i principis Agile. Els valors que Agile ens proposa són la prioritització de:

- Individu i interaccions per sobre dels processos i eines.
- Programari funcional per sobre de documentació comprensiva.
- Col·laboració amb el client per sobre de negociacions contractuals.
- Resposta davant del canvi sobre seguir un pla.

Aquests valors, juntament amb els 12 principis Agile, ens posen en escena una mentalitat flexible i constant pel desenvolupament del nostre programari. Podem trobar diverses metodologies que segueixen aquesta filosofia, entre les quals les més utilitzades són:

- SCRUM: una metodologia basada en "sprints" (entregues parcials del projecte) curts d'una o dues setmanes dissenyada per grups de treball petits.
- Kanban: una metodologia basada en tasques que segueixen un flux de treball per mitjà de taulers.
- Programació Extrema (XP): metodologia basada en cicles curts amb moltes versions a la disposició dels clients per accelerar i millorar la productivitat.

3.2 Metodologia Kanban

D'entre les metodologies esmentades, en aquest projecte s'utilitzarà la metodologia Kanban per la utilització d'aquesta a la meua feina i la comoditat obtinguda a l'hora

de desenvolupar amb aquest tipus de metodologia gràcies al kanban dels projectes que proporciona la plataforma *GitHub*.

Tal com s'ha esmentat prèviament, la metodologia Kanban es basa en una organització de tasques en taulers de forma que es pot mantenir un seguiment de l'estat d'aquestes. Aquestes tasques són relatives al desenvolupament del projecte, i haurien de poder completar-se en terminis curts de temps per poder presentar les noves funcionalitats als clients al més aviat possible. També es poden donar diferents prioritats a diferents tasques per poder establir un ordre en què es desenvolupen aquestes sigui per complexitat de tasques o per peticions amb més urgència per part dels clients.

3.3 Beneficis de Kanban

Entre els beneficis que podem obtenir a l'hora d'utilitzar aquesta metodologia podem considerar la facilitat per visualitzar l'estat del projecte, factor que permet una millor organització en la càrrega de treball de cada treballador en el desenvolupament del projecte, evitant càrregues excessives o insuficients per aquests.

Amb aquesta metodologia es pot aconseguir una eficiència molt alta en el flux de treball del projecte gràcies a la idea de no començar noves tasques fins a acabar les actualment assignades, reduint el desenvolupament de múltiples tasques a l'hora amb una dedicació inferior en cadascuna d'elles.

També, i per acabar, podem considerar la facilitat per començar a fer ús d'aquest tipus de metodologia gràcies als kanbans proporcionats per *GitHub* o altres plataformes com hem esmentat prèviament com un factor molt beneficiós per començar a treballar de forma immediata.

3.4 Ús de Kanban

Pel que fa al projecte *Learnum*, la metodologia Kanban s'ha desenvolupat amb l'eina de Kanban subministrada per *GitHub* com ja hem comentat, aprofitant no només de les eines de control de versions sinó també de les eines de gestió de projectes. Aquest kanban conté aleshores tasques relatives a les tres parts del projecte: tasques relatives al servei API Rest, tasques relatives a l'aplicació per dispositius mòbils, i tasques relatives a l'aplicació per escriptori.

Inicialment, es va definir les fases de producció de les tasques. Aquestes fases són els estats o cicle de vida pel qual les tasques passen al llarg del seu desenvolupament. Pel tauler kanban d'aquest projecte es va establir un cicle de 4 columnes:

- La columna inicial *To Do* o per fer, en aquesta columna s'afegeixen totes les tasques noves que encara no s'hagin començat a desenvolupar i per les quals encara s'ha de fer una estimació del temps que trigaran a completar-se,
- la següent columna és l'estat *In Progress* o en progrés, l'estat de la qual defineix les tasques per les quals s'ha començat la seva implementació i per les quals se'ls hi ha donat un temps estimat de finalització,

- la tercera columna és l'estat de revisió, la columna *In Review*, les tasques en aquesta columna estan en fase de prova, comprovant la correctesa de les funcionalitats implementades en les tasques,
- i l'última columna, la columna *Done* o complet, on es troben totes les tasques finalitzades.

Un cop establert el flux de treball del projecte, es varen establir unes tasques inicials per posar en marxa el desenvolupament del projecte. Posteriorment, i a mesura que el projecte avançava, es van anar afegint més tasques relatives a funcionalitats parcials o senceres dels objectius del projecte relacionades a noves funcionalitats o de ja implementades parcialment en tasques completades.

La metodologia Kanban utilitza els límits de "work in progress" (o WIP) que delimiten el nombre de tasques en desenvolupament simultani per tal d'establir una correcta càrrega de treball en tot moment que algunes tasques puguin generar. Com a límit en el desenvolupament d'aquest projecte es va establir un màxim de 3 tasques en procés d'implementació simultànies per tal d'evitar els problemes que pot ocasionar el *multitasking* en la correctesa de les funcionalitats presentades.

Capítol 4

Planificació

Un cop escollida una metodologia amb la qual treballar per aquest projecte, cal planificar el desenvolupament d'aquest. Per tant, en el següent capítol es busca explicar la planificació efectuada pel projecte, així com el flux de desenvolupament que s'ha portat a l'hora d'implementar les funcionalitats de les aplicacions.

4.1 Mètode de planificació

Per tal d'elaborar una planificació coherent de les tasques a realitzar pel nostre projecte, disposem de diferents formes per fer-ho. Una proposta de planificació popular és l'anomenada *Gantt Chart* o Diagrama de Gantt, on es defineix una estimació de la duració del desenvolupament de les tasques al llarg d'un termini, dotant a cada tasca d'un temps específic pel seu desenvolupament. Si bé aquest tipus de planificació pot donar bons resultats, ja que permet establir uns límits de temps en el que les tasques han de ser presentades, sovint també suposa haver de deixar tasques parcialment incompletes per poder seguir el ritme de desenvolupament.

Un altre tipus de planificació, que considero més adient pel tipus de metodologia utilitzada, és la *Kanban Planning* o Planificació Kanban. Inicialment en aquest tipus de planificació, es fa un estudi previ de les tasques que s'hauran de desenvolupar i s'afegeixen al tauler kanban. Aquest tauler es pot visualitzar com una línia temporal, on les tasques per fer són tasques que s'hauran de fer en algun moment al futur, les tasques en procés de fer-se són l'actualitat, i les tasques que ja s'han completat són tasques que s'han fet en un moment passat. Aleshores, al tauler kanban s'hauran d'afegir les tasques amb una estimació del temps que requerirà el compliment de cada tasca.

Si bé la planificació Kanban es basa en el mateix concepte de fer una planificació del temps que es requerirà per cada tasca com amb la planificació del diagrama de Gantt, amb aquesta no estem fixant dates, sinó aproximacions dels dies que es requeriran per la implementació de les tasques, donant flexibilitat a l'hora d'afegir noves tasques, reorganitzar les tasques en cas d'haver-hi algun canvi de prioritat sobtat, i no haver de cenyir-nos a dates preestablertes, sinó aprofitar l'experiència obtinguda amb tasques anteriors per fer una millor estimació del temps per les següents tasques a desenvolupar.

4.2 Flux de treball i tasques

Com ja s'ha comentat en l'apartat anterior, el desenvolupament del projecte ha estat recolzat per una estructura de quatre columnes al tauler kanban que defineixen els estats del flux de treball en què es troben les tasques que contenen. A forma de recordatori, aquestes columnes són: la columna de tasques per començar a fer, la columna de tasques que s'estan implementant actualment, la columna de tasques que estan en revisió per mitjà de proves i la columna de tasques completades. Per poder fer la planificació de les tasques es van dividir en 3 fases.

4.2.1 Fase d'estudi i preparació

Aquesta primera fase del projecte, és en la qual es va fer un estudi previ de les plataformes per les quals es desenvoluparia el projecte, i és on varen decidir quins serien els requisits del projecte i les tasques inicials per poder començar el desenvolupament del projecte.

Una vegada definits els requisits es va fer el disseny de la base de dades que donaria suport a les aplicacions. Tot seguit es va fer un estudi dels casos d'ús que ens trobaríem a la plataforma com a actors.

Pel que fa a les plataformes, es va fer un estudi previ de les tecnologies *NodeJS*, *Express*, i *Sequelize* pel desenvolupament del servidor de servei API Rest. En acabar aquest estudi, es varen examinar les tecnologies relacionades a les vistes: *React*, *React Native*, *Expo*, i *Electron.js*. Entrarem en més detall en aquestes tecnologies en els següents apartats.

Per finalitzar l'estudi es va prendre la decisió de començar el desenvolupament de la vista de la plataforma pel seu vessant d'aplicació per dispositius mòbils, i a posteriori el desenvolupament de l'aplicació per dispositius d'escriptori.

Aquesta decisió es va prendre perquè es va considerar que el canvi constant de Frameworks pel desenvolupament de les aplicacions del projecte (a més de canviar d'entorn per la feina on treballa), podrien suposar un problema a l'hora de reajustar la forma de pensar cada cop que es comencés cada tasca. A més, també es va considerar que seria una millor opció començar pel desenvolupament de l'aplicació per dispositius mòbils, ja que una aplicació amb intenció de funcionar com una xarxa social a part de com a fòrum obtindria millor recepció per dispositius d'aquest tipus.

Un cop finalitzat l'estudi previ de les plataformes, comença la preparació dels repositoris per tal de mantenir un control de versions de les aplicacions. L'eina utilitzada per fer aquest control ja ha estat esmentada prèviament i es tracta de GitHub.

4.2.2 Fase de desenvolupament

En aquesta es desenvolupa tot el flux del desenvolupament del projecte pel que fa a la implementació de les plataformes en ordre i de forma cíclica:

- Plantejament de la nova tasca a desenvolupar tasca.
- Implementació funcionalitat.

- Proves relatives i depuració de la funcionalitat implementada.
- Presentació i integració de la nova funcionalitat al projecte.

Aquesta és la fase principal i més ampla del projecte, ja que s'estén des de l'inici del desenvolupament del projecte fins al final d'aquest. Per tant, l'objectiu d'aquesta fase és assolir la implementació de les funcionalitats establertes prèviament:

- Integració de creació de compte d'usuari, inici de sessió i tancament de sessió.
- Administració del perfil de l'usuari.
- Administració de les relacions de seguiment entre usuaris.
- Creació i administració de fòrums.
- Creació i administració de publicacions.
- Creació i administració de comentaris.
- Creació i administració de tasques.
- Creació i administració de xats de missatgeria directa entre dos usuaris.
- Creació i administració de xats de grup entre múltiples usuaris.
- Integració dels permisos pels membres d'un fòrum.
- Integració dels permisos pels membres d'un grup.

4.2.3 Fase final

Per finalitzar, es troba la fase final del desenvolupament del projecte. Aquest apartat se centra en la comprovació de la correctesa de les aplicacions, afegir els fitxers de multimèdia necessaris i el desplegament de les aplicacions al núvol, així com la redacció final de la memòria del projecte.

Capítol 5

Marc de treball i conceptes previs

En aquest capítol es presentarà l'explicació del marc de treball del projecte, en altres paraules, els entorns i eines utilitzades al projecte, així com conceptes previs relatius a aquestes per la compressió d'algunes explicacions futures en aquest document.

5.1 Entorns de desenvolupament integrats

Qualsevol persona pot agafar el bloc de notes del sistema i començar a escriure codi que, a posteriori, conformarà part de la seva aplicació. Si bé això és cert, no treu el fet que pot ser complicat el desenvolupament d'una aplicació a mitjana o gran escala amb un bloc de notes, ja que pot ser una tasca tediosa. Actualment, la gran majoria de desenvolupadors utilitzen un tipus de programari que els hi facilita l'estructuració dels projectes, optimització de temps gràcies a funcionalitats integrades i, en general, una millor experiència a l'hora de "picar codi". Aquestes eines són els denominats entorns de desenvolupament integrats o *IDE* de les sigles en anglès.

Els entorns de desenvolupament proporcionen d'una col·lecció d'eines i processos als enginyers de programari per tal de, com s'ha dit abans, millorar la seva experiència a l'hora d'implementar les seves idees. Podem trobar-nos entorns de desenvolupament completament dedicats a un llenguatge (o conjunt de llenguatges específics) com pot ser Android Studio, que tal com el seu nom indica és un IDE centrat en el desenvolupament d'aplicacions Android amb llenguatges com JAVA o Kotlin, o PyCharm, dedicat a la programació en Python.



(A) Logo de l'IDE Android Studio [1]



(B) Logo de l'IDE Pycharm [2]

Per altra banda, també disposem d'entorns de desenvolupament que no són dirigits a un mercat en concret de desenvolupament de programari. Aquests permeten una

flexibilitat absoluta o quasi absoluta per tal de configurar, a preferència del desenvolupador, l'entorn amb les eines, processos, i fins i tot l'aparença d'aquest. L'exemple més popular per aquest tipus d'IDE podria ser Visual Studio Code.



FIGURA 5.2: Logo de l'IDE Visual Studio Code [3]

5.2 Emuladors

Quan ens trobem desenvolupant una aplicació hi ha una part molt important, i és la fase de proves de l'aplicació. Aquestes proves es poden fer utilitzant directament algun dispositiu del qual un disposi, i instal·lant l'aplicació desenvolupada per tal de provar-la, però per desgràcia no sempre es disposa de dispositius, ni de mòbils ni d'escriptori, per tots els sistemes operatius disponibles per cada tipus de dispositius pel qual es vol desenvolupar. En aquestes situacions, ens trobem amb eines molt potents i usades sovint a l'hora de desenvolupar aplicacions, especialment per dispositius mòbils: els emuladors.

Els emuladors són un tipus de programari que, com el seu nom indica emulen, o imiten entorns i comportaments de sistemes operatius d'un dispositiu. Aquests tipus d'aplicacions ens són especialment útils a l'hora de procedir amb la fase de proves, ja que ens permeten treballar en entorns dels quals és possible que no disposem, com podria ser un mòbil iOS per algú que tingui un mòbil Android, o un dispositiu Windows pels usuaris de Mac.

Entre els emuladors disponibles al mercat podem trobar per diferents tipus de dispositius, com mòbils, consoles, d'escriptori, entre altres. Alguns dels més famosos, especialment pel desenvolupament d'aplicacions, poden ser: Android Studio, amb el seu gestor de dispositius; Genymotion, un plugin disponible per Android Studio que també permet l'emulació de dispositius Android; o XCode, amb el seu emulador iOS integrat.

5.3 NodeJS i Node Package Manager

En el passat, JavaScript era un llenguatge només disponible a l'hora de fer desenvolupament web, això era degut al fet que només els navegadors web disposaven d'un motor que transformava el codi JavaScript escrit per la plataforma web a llenguatge comprensible per la màquina. De codi obert i multiplataforma, NodeJS és un motor basat en esdeveniments asíncrons que serveix com la proposta per solucionar aquest problema, permetent l'ús de JavaScript en aplicacions en fer que la màquina sigui capaç d'entendre el contingut del codi escrit. Aquest motor ens permet aleshores el desenvolupament d'aplicacions escalables del costat del servidor.

Per altra banda, el Node Package Manager o, amb la seva abreviatura per sigles més coneguda, *npm* és una llibreria o registre públic on els usuaris poden afegir paquets o mòduls que permeten afegir funcionalitats concretes a projectes. A més, *npm* afegeix entre les seves funcionalitats un conjunt de crides útils per utilitzar en la línia d'ordres o terminal per tal d'interactuar amb el repositori públic i administrar els paquets instal·lats en el projecte o globalment en el dispositiu.



(A) Logo de NodeJS [4]



(B) Logo de Node Package Manager (npm) [5]

5.4 API Rest

Quan volem comunicar les nostres aplicacions a un servidor que emmagatzema la informació relativa a aquestes disposem de les API de les sigles en anglès *Application Programming Interfaces*. Aquest tipus d'aplicacions serveix com a pont entre dues aplicacions per comunicar-se, com per exemple, entre la vista d'una aplicació i el servidor de la base de dades d'aquest.

Per altra banda, una API Rest, també coneguda com a API RESTful, són API que segueixen un conjunt de directrius o principis del disseny d'arquitectura REST (Representational State Transfer). Aquests principis defineixen una API Rest com una API que compleix:

- Interfície uniforme. Totes les peticions per obtenir el contingut d'un recurs han de seguir la mateixa estructura, vinguin d'on vinguin per assegurar que tot conjunt de claus úniques sempre atorgarà la recuperació del mateix recurs. Aquestes claus són considerades en conjunt la **URI** del recurs recuperat.
- Desacoblament client-servidor. L'aplicació de la vista o client ha de ser completament independent de l'aplicació del servidor. En altres paraules, ha de mantenir el seu correcte funcionament només coneixent els URI necessaris per a la recuperació de recursos del servidor per mitjà de l'API, sense disposar d'una connexió directa a aquest servidor. De la mateixa manera, el servidor tampoc haurà de ser capaç d'interactuar directament amb el client de la nostra aplicació per modificar-ho.
- Sense estat. L'API no ha de mantenir cap mena d'estat, cada petició que rep ha de ser tractada individualment i independent de les anteriors, evitant emmagatzemar informació d'aquestes que es pugui reutilitzar a posteriori. Per tant, les peticions han de contenir tota la informació necessària per ser processada correctament.

- *Cacheable*. Ocasionalment, ens trobem que una petició ha de recuperar múltiples cops la informació d'un mateix recurs. Per tal d'evitar haver de fer la mateixa petició al servidor múltiples cops, per la qual cosa aquesta informació hauria de poder emmagatzemar-se en una memòria cau.
- Arquitectura del sistema en capes. La idea darrera d'aquesta arquitectura és la divisió dels diferents mòduls de l'aplicació en capes on cada una d'aquestes tingui una tasca concreta, per exemple una capa de validació.

5.5 React Native

Les aplicacions multiplataforma per dispositius mòbils disposen de diverses eines pel seu desenvolupament, com poden ser alguns Frameworks com podrien ser *Flutter* o *React Native* entre altres.

La decisió de quina tecnologia utilitzar per al desenvolupament de l'aplicació està subjecta a diferents aspectes com la comoditat amb el llenguatge pel qual està dissenyat el Framework, l'accessibilitat d'aquest, les funcionalitats que volem assolir en la nostra aplicació, etc.

En aquest treball, pel seu vessant per dispositius mòbils, he emprat *React Native* pel desenvolupament d'aquest. Podríem dir que *React Native* és una expansió de la idea de la llibreria de *React*, pel que fa al desenvolupament de la interfície d'usuari per mitjà de components, per la creació d'aplicacions per a mòbils.

Capítol 6

Requisits del sistema

El principal objectiu a l'hora de desenvolupar una aplicació és que funcioni. Per tal d'establir uns criteris per determinar el nivell de completesa d'aquesta aplicació es defineixen un conjunt de requisits tècnics que l'aplicació ha de complir en la seva primera versió final.

L'objectiu d'aquest capítol és definir aquest conjunt de requisits funcionals i no funcionals que hauria de complir l'aplicació resultant del projecte per poder considerar-la una versió final.

6.1 Requisits de les aplicacions

Prèviament, i al llarg del projecte, s'han anat establint diverses funcionalitats de les quals els usuaris de l'aplicació haurien de poder gaudir al final del desenvolupament del projecte. Aquestes funcionalitats són:

- Els **usuaris no registrats** podran crear un nou compte d'usuari.
- Els **usuaris registrats** podran:
 - Iniciar sessió.
 - Tancar sessió.
 - Visualitzar la informació del seu perfil d'usuari.
 - Editar la informació del seu perfil d'usuari.
 - Crear nous fòrums.
 - Crear noves publicacions pels fòrums.
 - Reaccionar a publicacions.
 - Comentar en publicacions.
 - Reaccionar a comentaris.
 - Crear tasques.
 - Obrir sales de xat amb usuaris que l'usuari segueix.

- Crear nous xats de grups.
- Poder afegir nous usuaris seguits.
- Enviar missatges pels xats.
- Per altra banda, i considerant-los usuaris especials amb funcionalitats addicionals, els **usuaris administradors de xats** de grups haurien de poder:
 - Eliminar missatges.
 - Editar dades del grup.
- Finalment, també considerem als **usuaris amb privilegis de moderador de fòrums** com a usuaris especials amb funcionalitats addicionals:
 - Eliminar posts.
 - Eliminar comentaris.

En aquesta llista també es pot visualitzar amb facilitat els *actors* de la plataforma Learnum.

6.2 Requisits dels dispositius

Pel que fa a l'ús de les aplicacions pels dispositius els quals aquestes estaran disponibles ens trobem que: els dispositius mòbils hauran de disposar d'una versió del sistema Android 5 o superior, o una versió del sistema iOS 13 o superior. Aquests requisits són presentats degut a l'ús del SDK d'Expo pel desenvolupament.

Per altra banda, els dispositius d'escriptori hauran de disposar com a mínim d'una de les següents versions de sistemes operatius:

- Windows 7 o superior, tant per binaris de 32 com 64 bits.
- Mac OS 10.10 o superior, només binaris de 64 bits.
- Linux. De les diferents distribucions que disposa, l'única versió assegurada a funcionar, ja que disposa dels binaris necessaris són les versions d'Ubuntu 12.04 o superiors.

Pel que fa al hardware, pels dispositius d'escriptori es recomana almenys 512 MB de memòria RAM.

Capítol 7

Estudis i decisions

Amb la finalitat de desenvolupar les aplicacions relatives al projecte, s'ha d'escollir el conjunt d'eines que s'utilitzaran al llarg d'aquest.

Per tal de poder prendre una bona decisió de les eines escollides cal fer un estudi previ d'aquestes, que és el que es pretén fer en aquest capítol a l'hora de prendre una decisió.

7.1 Visual Studio Code

Tal com es va anunciar a l'apartat 5.1 amb l'explicació dels entorns de desenvolupaments, qualsevol persona pot programar en un bloc de notes, però perquè fer-ho quan disposem d'eines tan potents com són els IDE. Per tant, pel desenvolupament d'aquest projecte, s'ha utilitzat un entorn de desenvolupament ja mencionat: Visual Studio Code.

Visual Studio Code és un entorn de desenvolupament d'edició de codi que permet la implementació d'aplicacions i serveis en diversos llenguatges de programació. Visual Studio Code, normalment abreviat VSCode, és un entorn personalitzable per tota classe de desenvolupament que es necessiti. Aquesta personalització s'obté gràcies a les extensions proporcionades per la comunitat de VSCode.

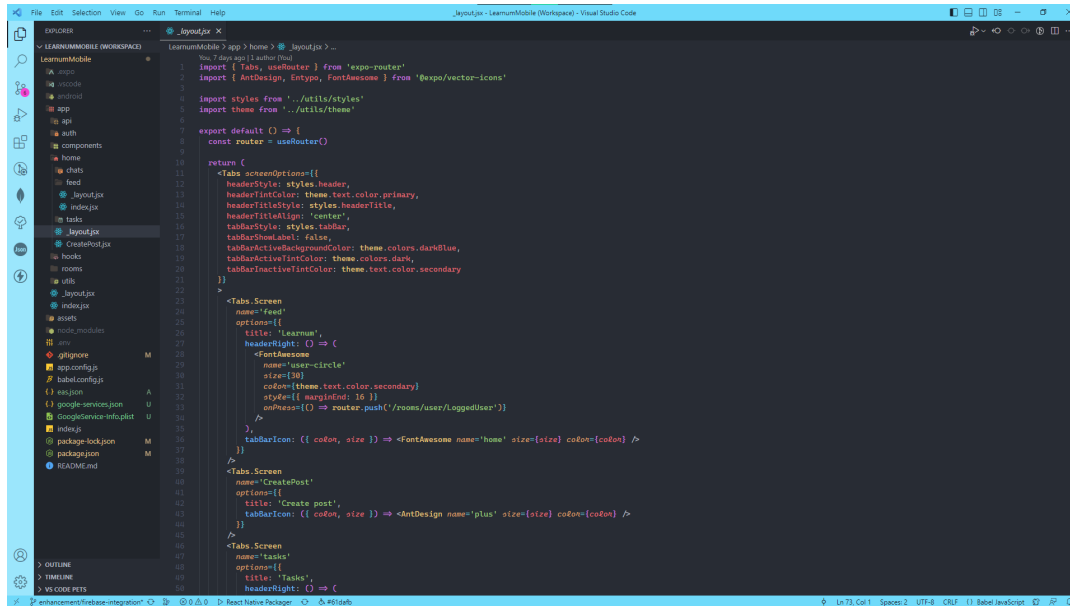


FIGURA 7.1: Captura de pantalla de l'entorn de desenvolupament de Visual Studio Code

7.1.1 Extensions

Les extensions de VSCode són plugins o add-ons proporcionats per membres o organitzacions que formen part de la comunitat de la plataforma, i afegeixen noves funcionalitats a l'entorn de desenvolupament. Aquestes funcionalitats poden anar des de: personalitzar com es veu l'entorn, afegint algun tipus de tema o canviant els colors; fins depuradors per llenguatges o altres eines de suport pel desenvolupament com per exemple un Client REST o una GUI per interactuar amb el repositori de GitHub en cas de trobar-nos en un projecte que disposi d'un repositori integrat al seu directori.

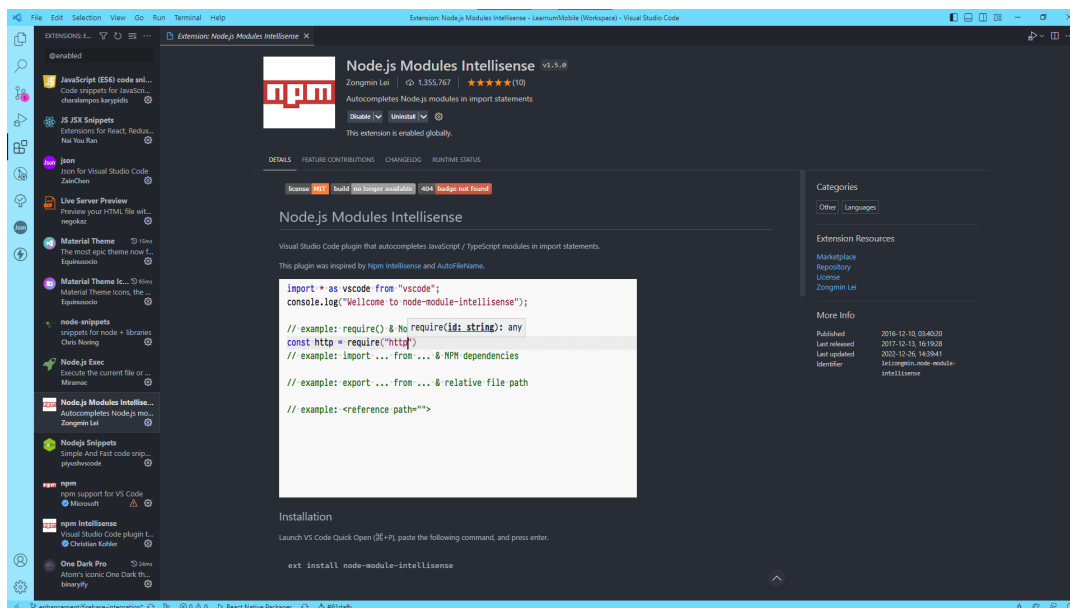


FIGURA 7.2: Captura de pantalla del panell d'extensions de l'entorn de desenvolupament de Visual Studio Code

És degut a la comoditat que suposen aquestes extensions que es troben en constant desenvolupament per una gran comunitat, i la flexibilitat amb la qual doten als usuaris, que l'elecció d'IDE pel desenvolupament d'aquest projecte ha estat, en efecte, Visual Studio Code.

7.2 API Rest

7.2.1 XAMPP

Per tal d'establir una connexió amb un servidor, primer n'hem de disposar d'un. Normalment, les aplicacions es troben allotjades al núvol, en servidors externs per tal que les aplicacions relacionades a aquesta puguin accedir-hi. A l'hora de desenvolupar, per altra banda, no ens interessa haver de pagar un manteniment d'una aplicació que encara està en fase de construcció. Per això mateix disposem dels servidors locals, plataformes independents que permeten interactuar amb bases de dades en l'àmbit local del dispositiu on aquest es trobin instal·lades.

El servidor amb el qual es vol treballar inicialment és un que disposi de base de dades tipus *MySQL*. Afortunadament, disposem de **XAMPP**, una aplicació de programari lliure que permet la utilització d'un servidor local per mitjà d'un servidor web *Apache* amb una base de dades *MySQL*.

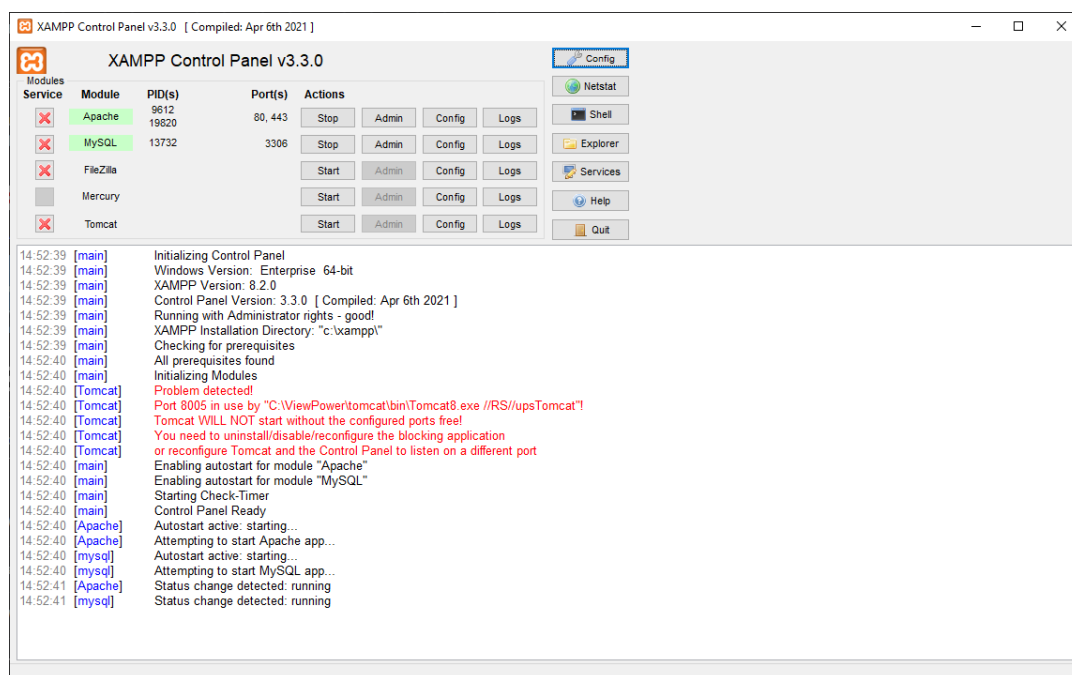


FIGURA 7.3: Captura de pantalla del panell de control de XAMPP

Si bé XAMPP és una aplicació orientada a servidors PHP, és possible utilitzar el servidor local que subministra amb altres llenguatges i Frameworks com és el cas d'aquest projecte. Això, sumat amb la comoditat a l'hora de preparar l'entorn pel servidor local que proporciona, són dos factors molt importants a l'hora d'escollir aquest software per poder definir la base de dades local de Learnum.

7.2.2 Express

Express és un Framework Web escrit en JavaScript que forma part dels mòduls disponibles a NodeJS per mitjà del npm que proporciona diferents eines i funcionalitats per la creació de l'API Rest de l'aplicació del projecte:

- *Routing*. Organització per mitjà d'URLs amb encaminament de peticions HTTP.
- Renderització de vistes dedicades a les rutes per visualitzar les respostes.
- Configuració de propietats com el port utilitzat a la connexió.
- Addició de *middlewares* per afegir capes a les rutes de les peticions per integrar funcionalitats intermèdies com la verificació de l'autenticació d'usuaris.

És l'opció més popular a l'hora de desenvolupar aplicacions API, per tant, disposa d'una colossal comunitat que suposa un gran suport a l'hora de treballar amb aquest Framework per qualsevol dubte o problema que sorgeixi amb aquest.

7.2.3 Sequelize

Un cop establerta la connexió amb la base de dades de XAMPP per mitjà d'Express, cal decidir com definir les taules de la base de dades. En treballar amb XAMPP podem accedir directament a la pàgina de *phpmyadmin*, la qual proporciona una GUI per interactuar amb les bases de dades del servidor local i les taules que aquestes contenen.

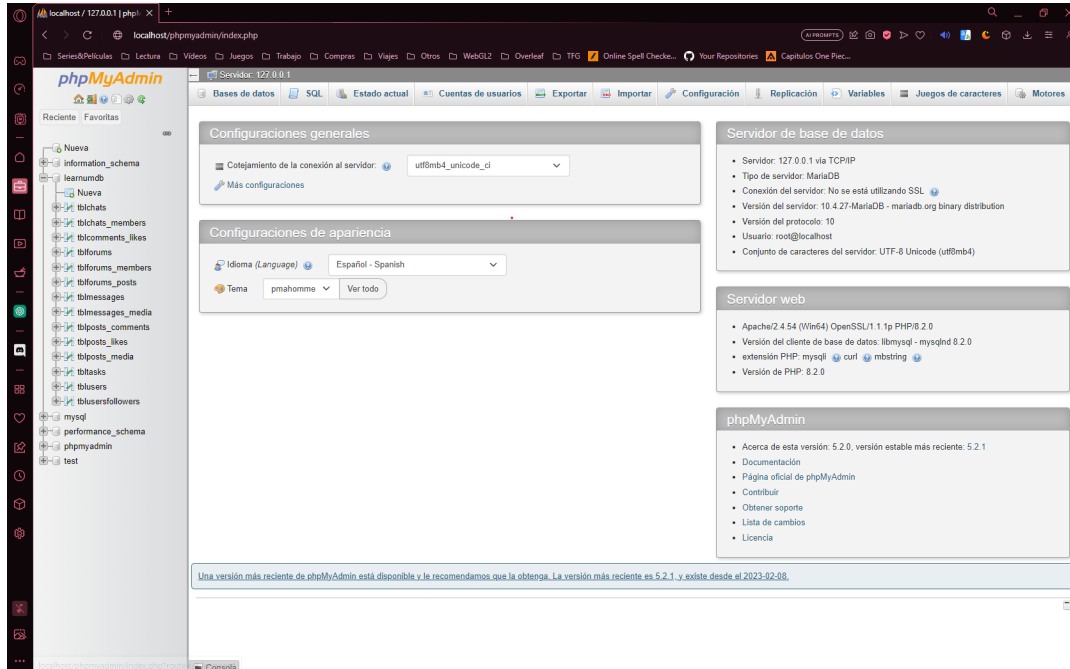


FIGURA 7.4: Captura de pantalla del panell de control local de PHPMyAdmin

Dins d'aquesta web podrem definir la base de dades i les taules que la conformen

amb les columnes que contenen. En cas d'automatitzar la creació de tot aquest contingut, també podem crear un *script* SQL, un fitxer amb peticions escrites en llenguatge SQL, que generi tot els apartats mencionats important-ho en la secció "Importar" en PHPMyAdmin.

Totes les opcions mencionades fins ara són acceptables i funcionals, per tant, es podria definir la base de dades d'aquesta forma manual, però també hi ha una altra opció: el paquet de NodeJS, **Sequelize**.

Sequelize és un ORM (*Object Relational Mapping*), un model de programació que transforma les taules i tasques més bàsiques relatives a bases de dades en crides simplifiades. Sequelize, que està dedicada a bases de tipus SQL com poden ser MySQL o MariaDB, permet entre altres funcions:

- La creació de **models** que defineixen les taules de la base de dades,
- la definició de relacions entre models que associen les taules que representen,
- l'execució de crides CRUD (**C**reate, **R**ead, **U**psert and **D**elete) sobre els models definits,
- i la definició de **hooks** pels models que defineixen els disparadors d'esdeveniment, per exemple per a quan una nova entrada per un model és creada a la seva taula.

Per tant, i amb consideració dels avantatges que suposa la simplificació de la creació d'una base de dades per mitjà d'aquest tipus d'eines, Sequelize és l'opció escollida a l'hora de definir i interactuar amb la base de dades de la nostra aplicació API REST.

7.2.4 Postman

Per tal de comprovar que s'està fent un bon desenvolupament de l'API podem testear-la de dues formes: per mitjà d'una aplicació de vista per la qual estem desenvolupant l'API. Forma que, si bé funciona, implica que hem de disposar d'una aplicació ja feta, o almenys que estigui prou desenvolupada, per poder utilitzar-la. Una altra opció que ens trobem són els **clients REST**.

Els clients REST són eines que permeten fer peticions a serveis API que remots o que estiguem desenvolupant per mitjà de URIs. De les diferents opcions que es disposen com a clients REST s'ha escollit la utilització de **Postman** per l'experiència obtinguda per aquest client en diferents projectes.

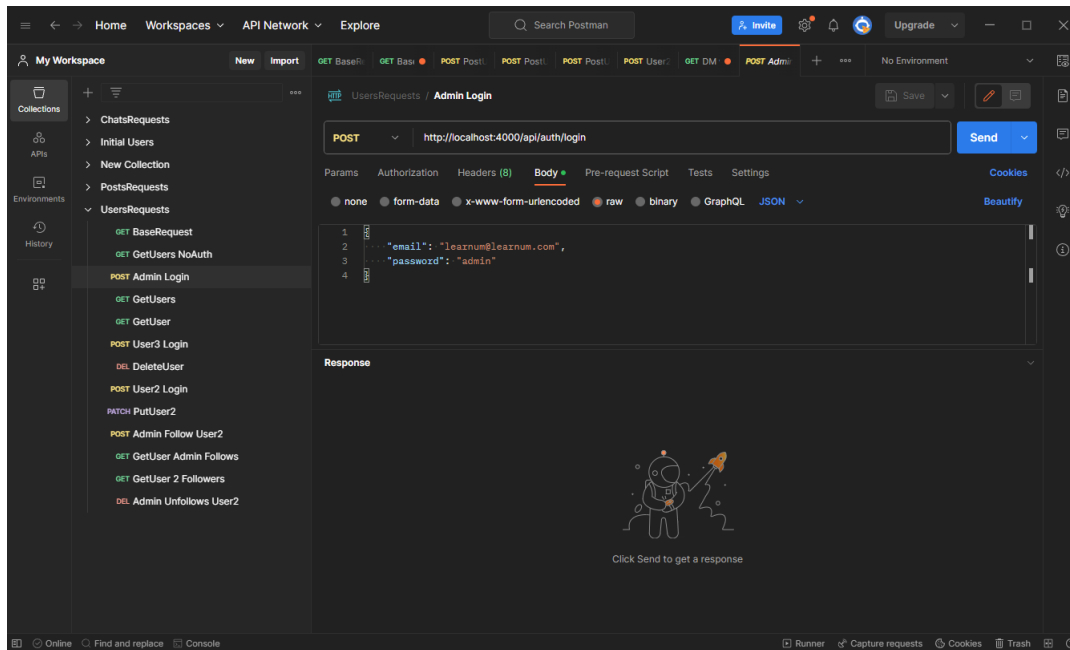


FIGURA 7.5: Captura de pantalla del client REST Postman

Postman ens proporciona una GUI simple que permet la creació de col·leccions de peticions, cadascuna amb el seu propi contingut: tipus de petició HTTP, informació del cos (*body*) de la petició, informació de la capçalera de la petició com *tokens* d'autorització, entre altres.

7.3 React

React és una llibreria desenvolupada per Facebook (Meta), que permet el desenvolupament de pàgines web per mitjà de components. Tal com vàrem dir al capítol 5, secció 5.5, aquesta llibreria és la base de React Native, la tecnologia utilitzada per aquest projecte, i ens cal estudiar-la abans de treballar amb React Native, ja que estableix molts dels conceptes i les eines utilitzades al Framework.

7.3.1 DOM Virtual

Quan ens trobem treballant en una aplicació web, molts de cops ens trobarem amb el DOM (*Document Object Model*). Aquest és un model o representació gràfica, normalment imaginada en forma de diagrama d'arbre, creada pels navegadors que mostra els continguts organitzats d'un document o pàgina. Si en qualsevol moment es produeix un esdeveniment o canvi d'estat que alteri l'estructura de la pàgina, serà aquest DOM sobre el qual s'apliquin els canvis necessaris per mostrar la nova vista. Aquest canvi pot ser una càrrega significativa a l'hora de mostrar els nous continguts de la pàgina, ja que per tal de modificar-ho, s'ha de renderitzar tots els elements que formen part del diagrama d'arbre del DOM, cosa que és costosa.

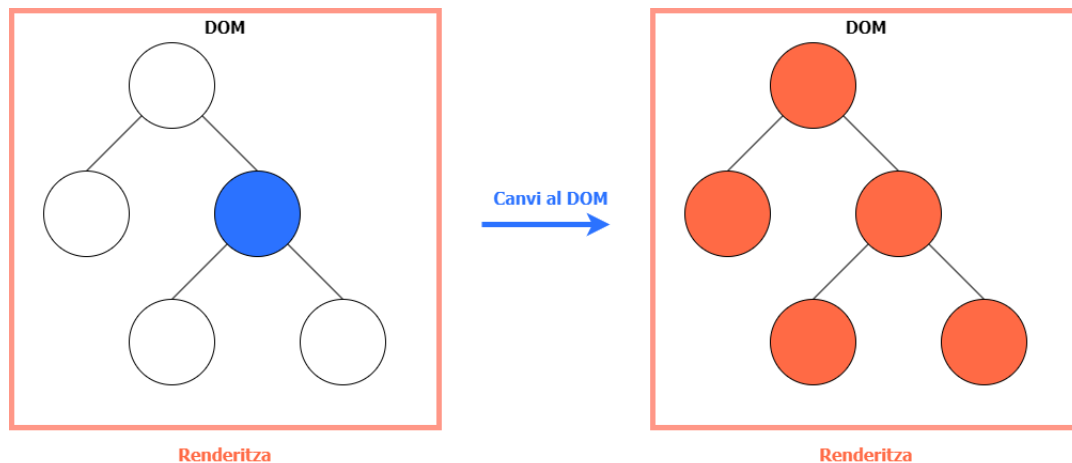


FIGURA 7.6: Diagrama explicatiu del funcionament del DOM per actualitzar els continguts visibles de l'aplicació web

Per combatre aquesta actualització tan costosa de la vista de la pàgina d'una aplicació web per un possible canvi d'una dimensió minúscula, disposem del *Virtual DOM*. El Virtual DOM, o VDOM, és una eina que permet una millor eficiència a l'hora d'interactuar amb els estats i esdeveniments de l'aplicació web. Aquest, emmagatzema una representació del DOM a memòria, que utilitzarà per fer un càlcul per trobar la manera més eficient possible de tornar a renderitzar el DOM amb el menor nombre de canvis possibles.

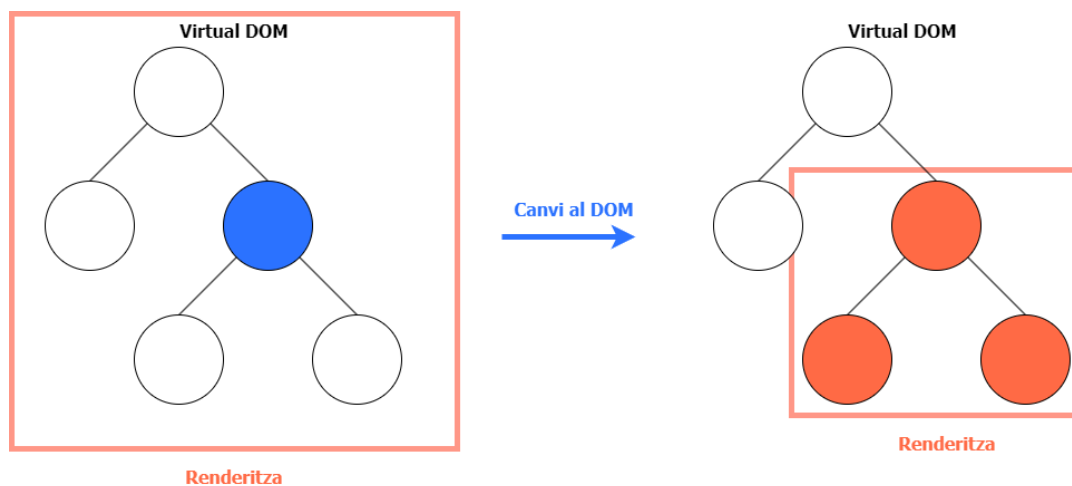


FIGURA 7.7: Diagrama explicatiu del funcionament del Virtual DOM per actualitzar els continguts visibles de l'aplicació web

7.3.2 JSX

El llenguatge de marcatge d'hipertext o HTML de les sigles en anglès, és un codi utilitzat per l'estructuració i desplegament de la vista d'una pàgina web, i presenta un conjunt d'etiquetes específiques per la visualització de diferents tipus de continguts com: llistes, imatges, peus de pàgines i molt més.

Per altra banda, si bé React no fa ús d'HTML per l'escriptura dels seus components, fa servir un llenguatge de marcatge similar. Aquest llenguatge de marcatge és una

extensió de la sintaxi per JavaScript i és coneguda com a *JavaScript Syntactic Extension*, *JSX* o JavaScript XML. Aquesta extensió ens permet escriure etiquetes similars a les que ens trobaríem amb HTML, proporcionant llistes, capçaleres, peus de pàgina, seccions, i més, amb la possibilitat d'intercalar aquestes etiquetes amb blocs d'expressions JavaScript encapsulat entre claus.

```
1 const Greeting = () => {  
2   const name = "Admin"  
3   return <h1>Hello {name}!</h1>  
4 }
```

El codi anterior és un exemple d'un codi escrit en JSX. Aquest codi és la definició d'un component. Després veurem que és exactament un component, però el que ens interessa d'aquest exemple és la sintaxi. Com podem veure, aquest codi retorna un contingut similar al que trobaríem en el codi HTML d'una pàgina web, però amb la particularitat de trobar-se amb un bloc delimitat per claus tal com s'ha esmentat abans. Aquest bloc conté una expressió JavaScript, més específicament l'accés a una variable `name`, que s'interpretarà i com a part del resultat final del component. Per aquest exemple, el resultat final serà `Hello Admin!`.

7.3.3 Components

Coneixem com a components en React a trossos de codi reutilitzable en diferents parts d'una aplicació. Per qüestions de disseny, quan es defineixen un component, el nom donat a aquest ha de ser majúscula.

En conceptes generals, els components i les funcions tenen el mateix objectiu de retornar un resultat a partir de valors d'entrada, mantenint en tot moment la seva independència.

Serveixen principalment per dividir fragments de la vista en parts reutilitzables com ja s'ha comentat. Un exemple d'un component podria trobar-se en una llista. Pot existir una llista on cada element d'aquesta estigui representada per un component que proporcioni una vista concreta.

```
1 const ListItem = () => {  
2   return <p>Item</p>  
3 }
```

Propietats

A l'hora d'utilitzar un component, i mantenint l'exemple de l'element d'una llista, podem voler modificar aquest per mostrar diferent informació amb aquest. Les propietats, o *props* com normalment s'anomenen en l'àmbit de React, són arguments que reben els components per definir els seus atributs i permetre una personalització del component des de fora sense haver de fer un nou component per un altre element del mateix tipus.

```
1 const ListItem = (props) => {  
2   const { text } = props  
3   return <p>{text}</p>  
4 }
```

En l'exemple de dalt, que és l'extensió de l'anterior, s'ha modificat el component de l'element d'una llista per tal de proporcionar una propietat `text`, recuperada utilitzant *desconstrucció* de l'objecte, i que permet des de fora de l'element donar un valor per diferenciar a cada *item*.

Per poder fer aquesta assignació de les propietats externes al component, la sintaxi és la següent:

```
1 const List = () => {
2   return (
3     <ul>
4       <ListItem text="Element 1"/>
5       <ListItem text="Element 2"/>
6     </ul>
7   )
8 }
```

7.3.4 State

React té un funcionament concret: renderitzar diferents components per generar una vista. Podem utilitzar eines com les propietats per donar valors a diferents instàncies d'un mateix component, però, i a l'hora d'afegir una funcionalitat com un comptador?

Els components de React, per naturalesa del disseny, no són dinàmics, un cop renderitzats aquests es mantenen amb el mateix contingut. Però, i si volem fer un component que ens serveixi per mantenir constància d'un comptador? La proposta de React per solucionar aquest problema són els estats.

L'estat d'un component, és un objecte que es defineix com a part del component, i que emmagatzema informació local d'aquest. Aquesta informació no pot ser enviada a altres components, i per tant és privada per cada component. Abans s'usava la definició de components per mitjà de classes de React per tal de poder afegir un estat a aquests, però en l'actualitat disposem d'una nova eina que ens proporciona una altra forma de definir els estats dels components i els seus comportaments de forma més elegant i intuïtiva: els *hooks*.

7.3.5 Hooks

Com ja s'ha esmentat abans, els *hooks* són eines que permeten proporcionar a un component d'estats i comportaments relatius a aquests. Aquests tendeixen a seguir dues regles:

1. Han de ser utilitzats en el nivell més alt d'un component.
2. El seu ús ha de ser incondicional dins d'un component.

Si bé disposem de diversos *hooks* proporcionats per la llibreria de React, n'hi ha dos en concret dels que cal parlar atès al seu ús extens: *useState* i *useEffect*.

useState

El hook **useState** ens proporciona la capacitat de dotar a un component qualsevol d'un estat i un mètode modificador per assignar nous valors a l'estat. En modificar

el valor de l'estat amb el mètode proporcionat, la vista del component on l'estat sigui utilitzat tornarà a ser renderitzada utilitzant el Virtual DOM de manera que poder visualitzar canvis dinàmics en l'aplicació. Per poder entendre millor l'explicat, a continuació veurem el codi d'un component que simularà el comportament d'un comptador.

```
1 const Counter = () => {
2   const [count, setCount] = useState(0)
3
4   const clickHandler = () => {
5     setCount(count + 1)
6   }
7
8   return (
9     <div>
10      <p>{count}</p>
11      <button onClick={clickHandler}>+</button>
12    </div>
13  )
14 }
```

Primer, i per posar en context, el component de l'exemple és un comptador que cada cop que es prem el botó + augmenta la quantitat comptada. En aquest definim l'estat del component amb el hook `useState` donant un valor inicial de 0. Aquest ens retorna l'estat i el mètode per modificar l'estat: `setCount`.

Com podem veure, el mètode `setCount` s'utilitza per augmentar el nombre de cops que el botó s'ha pres, augmentant en 1 el valor anterior de l'estat. Un cop aquest és actualitzat, React per mitjà del Virtual DOM calcularà la millor forma de tornar a renderitzar el component per tal de mantenir una eficiència alta, i en acabar, actualitzarà el paràgraf `<p>` del component pel nou valor modificat de l'estat `count`.

Per tant, podem veure com aquest estat ens és una eina molt potent a l'hora de desenvolupar lògica pels nostres components.

useEffect

Aquest hook, si bé no dota d'estat al component com ho fa el hook `useState`, ens permet dotar d'"efectes secundaris" als nostres components. Aquest efecte secundari s'executarà després del primer cop que es renderitzi el component, i cada cop que aquest sigui actualitzat, i abasta diferents funcionalitats, però una molt important és la recuperació d'informació via *fetch* d'una API.

Quan diem que el hook s'executarà cada cop que s'actualitzi, ens referim a un llistat de valors pel qual el hook dependrà, i que en cas d'actualitzar-se, dispararan l'execució d'aquest hook. Les dependències dels hooks s'indiquen en una llista la qual contindrà tots els estats o propietats utilitzades en el *callback* de l'`useEffect`. Per tant, la sintaxi del hook `useEffect` és:

```
1 useEffect(() => {
2   // Codi a executar
3 }, [ /* Llista de dependències */ ])
```

Custom hooks

Si bé les eines proporcionades per React ens suposen una eina molt potent, a vegades podem trobar-nos amb una constant reutilització del mateix comportament dels hooks per diferents components provocant una duplicació del codi. Per tal de solucionar aquest problema disposem dels coneguts hooks personalitzats o *custom hooks*.

Aquests són mètodes que simulen el comportament dels hooks de React, per afegir funcionalitats personalitzades als components. Per entendre millor aquest concepte podem presentar un exemple.

```
1 const useCount = (initialCount) => {  
2   const [count, setCount] = useState(initialCount)  
3  
4   const increment = () => setCount(count + 1)  
5  
6   return [count, increment]  
7 }
```

Per qüestions de disseny, els noms dels custom hooks implementats han de contenir el prefix “use” per poder identificar-los com a hooks, i ho podem observar en el nou custom hook creat: **useCount**.

El custom hook useCount ens serveix com a extensió de l'exemple presentat en l'explicació de l'useState, i ens proporciona, per mitjà de l'array retornat, les propietats necessàries per afegir la funcionalitat del comptador al component. Per tant, passem del codi anterior al següent:

```
1 const Counter = () => {  
2   const [count, increment] = useCount(0)  
3   return (  
4     <div>  
5       <p>{count}</p>  
6       <button onClick={increment}>+</button>  
7     </div>  
8   )  
9 }
```

Podem veure que l'exemple anterior es redueix significativament en mida, ja que la lògica del *handler* del esdeveniment de clic al botó és directament el retornat pel custom hook useCount que s'ha implementat.

Aquests tipus de hooks també presenten un avantatge, i és que com la lògica per modificar els estats d'aquests és interna, facilita el manteniment i validació dels valors pels quals s'actualitzen aquests estats siguin correctes pel degut funcionament del comportament del hook.

7.4 React Native

Ara que ja disposem del coneixement de les bases de React presentades a l'apartat anterior, podem endinsar-nos en React Native.

React Native, també desenvolupat per Meta (anteriorment Facebook), és un Framework dedicat a la implementació d'aplicacions multiplataforma per dispositius mòbils amb sistema Android o iOS.

7.4.1 Components

Igual que amb React, React Native també ens proporciona d'un conjunt de Components prefabricats que ens proporcionen diferents funcionalitats.

Ara bé, en React els components prefabricats eren escassos i s'implementaven amb les etiquetes HTML-like de JSX, amb React Native no és el cas. En React Native ens trobem que per tal de crear els components hem de fer ús dels components prefabricats que ens proporciona, i no les etiquetes JSX.

Això és degut al fet que aquests components proporcionats són interpretats en temps d'execució per React Native als seus corresponents components nadius per tal de mostrar la vista correcta en cada dispositiu.

Ja hem deixat clar que React Native disposa de molts components prefabricats a la disposició dels desenvolupadors. Donem un cop d'ull a alguns dels més bàsics.

- **View.** El component `View` és el més bàsic de tots, i proporciona la vista d'un contenidor similar al `div` d'HTML amb estil de *flexbox*.
- **Text.** Aquest component, com també l'anterior, és essencial pel desenvolupament, ja que aquest és l'únic component prefabricat que permet contenir text. Si bé a React podem afegir text en els paràgrafs, *spans*, capçaleres, etc. A React Native només disposem d'aquest component per aquesta funcionalitat.
- **FlatList.** Com el seu nom indica, aquest component presenta la funcionalitat per implementar llistes d'elements renderitzats per mitjà de components, i disposen d'un ampli abast de propietats amb les quals personalitzar la llista.

Cal mencionar que, tot i que la idea darrere d'aquest tipus de tecnologia és la d'escriure un cop i disposar de la funcionalitat tot arreu, React Native també presenta una funcionalitat que ens permet establir diferències de comportament, organització, o estils, relatius a cada plataforma sense haver d'escriure codi natiu per aquestes.

7.5 Emuladors: Android Studio i Xcode

Després d'afegir noves funcionalitats a la plataforma cal comprovar el correcte funcionament d'aquestes. Una de les eines utilitzades per fer aquesta fase de proves, són els emuladors. Tal com ja s'ha explicat prèviament, aquests permeten imitar entorns de sistemes operatius en dispositius virtuals on podrem instal·lar i provar l'aplicació.

Pel que fa a l'entorn iOS, com no es disposa d'un dispositiu mòbil amb aquest sistema, s'ha fet ús de l'emulador subministrat per XCode en un portàtil Mac. Gràcies a aquest, s'ha pogut fer proves per observar possibles anomalies entre la versió de l'aplicació per mòbils Android i la versió per dispositius iOS.

Per altra banda, per fer les proves en els dispositius Android, en disposar d'un dispositiu Android físic, s'han pogut efectuar tant les proves en aquest per mitjà d'Expo, del qual parlarem més endavant, i també s'ha fet ús dels emuladors subministrats per Android Studio amb el seu gestor de dispositius que permet la configuració de diferents dispositius virtuals per l'entorn Android per diferents versions.



FIGURA 7.8: Captura de pantalla d'instàncies dels emuladors per Android (dreta) i iOS (esquerra)

7.6 Expo

Expo és un SDK, en altres paraules un conjunt d'eines, llibreries i serveis que permeten la simplificació del codi de React Native utilitzat per les aplicacions que desenvolupem. També proporciona un emulador propi per l'execució d'aplicacions de React Native des de dispositius físics per mitjà d'una aplicació.

7.6.1 Expo router

Expo Router és un paquet disponible per Expo que afegeix un encaminador basat en fitxers disponible per aplicacions de React Native.

Permet la configuració de la navegació per l'aplicació mitjançant l'organització de les diferents pàgines o pantalles d'aquesta en carpetes.

7.6.2 Expo Go

Expo Go és una aplicació disponible per dispositius de sistemes Android i iOS que permet l'emulació d'aplicacions desenvolupades amb React Native i l'SDK d'Expo.

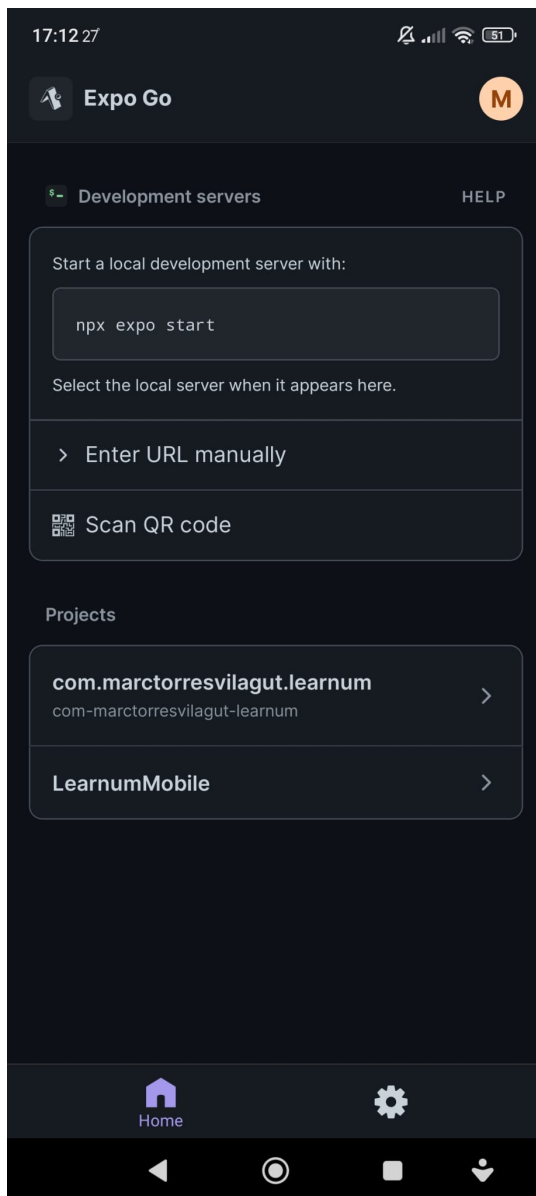


FIGURA 7.9: Captura de pantalla de l'aplicació Expo Go

Capítol 8

Anàlisi i disseny del sistema

Al desenvolupament d'un projecte cal determinar quines són les parts íntegres de l'aplicació com: quins tipus d'usuaris poblaran l'aplicació, les dades que s'hauran d'emmagatzemar, així com la distribució i relació d'aquestes i les funcionalitats del sistema. En aquest capítol es busca fer l'anàlisi d'aquestes parts, seguit del disseny del modelatge de dades en la base de dades de l'aplicació i el disseny de les funcionalitats d'aquesta.

8.1 Actors i mòduls

Primer de tot establirem, amb el que ja s'ha descrit anteriorment en el projecte, els actors que disposa l'aplicació. A partir dels objectius funcionals descrits, podem definir els diferents autors que conformaran la població de l'aplicació. Aquests són:

- **Usuari no registrat.** Aquest disposarà de funcionalitats reduïdes. De fet, l'única funcionalitat que tindrà disponible serà la de registrar-se com a membre de l'aplicació per poder desbloquejar algunes funcionalitats.
- **Usuari registrat.** Aquests actors podran utilitzar gran part de les funcionalitats disponibles per l'aplicació i són la forma més bàsica d'usuari.
- **Moderador de Fòrum.** Els moderadors són actors que estenen les funcionalitats de les quals disposa un usuari registrat bàsic. Aquestes funcionalitats són relatives als fòrums pels quals l'usuari és moderador i cap altre més.
- **Administrador de Grup.** Els administradors de grups, com també els moderadors de fòrums, són actors que estenen les funcionalitats bàsiques dels usuaris registrats, afegint noves funcionalitats de control sobre els xats de grups pels quals l'usuari és administrador.

Cal indicar que els rols dels actors de *moderador de fòrum* i *administrador de grup* no són mútuament exclusius. En altres paraules, per més que un usuari registrat sigui promocionat a administrador de xat de grup, això no significa que aquest usuari no pugui ser promocionat a moderador de fòrums o administrador d'un altre grup diferent.

Així mateix, també diferenciarem les funcionalitats entre els diferents mòduls: **Usuaris**, mòdul relacionat amb totes les funcionalitats relacionades amb la interacció

d'usuaris amb si mateixos o altres usuaris; **Fòrums**, mòdul relacionat amb les funcionalitats relatives als fòrums, posts i comentaris; **Tasques**, mòdul relacionat a les accions que pot fer un usuari amb les seves tasques; i finalment, **Xats**, totes aquelles funcionalitats relatives a creació i administració de xats.

8.2 Anàlisi del sistema

Com s'ha pogut observar en l'apartat anterior, l'aplicació Learnum està conformada per diversos mòduls o blocs. Cadascun d'aquests mòduls representa una part íntegra de l'aplicació i disposa de les seves pròpies funcionalitats als diferents actors i utilitzaran el seu propi conjunt de dades.

8.2.1 Model de dades

Una vegada definits els requisits que havia de satisfer l'aplicació vam fer una anàlisi de les dades que transitaran per l'aplicació. Per tant, ara visualitzarem les associacions entre aquestes dades.

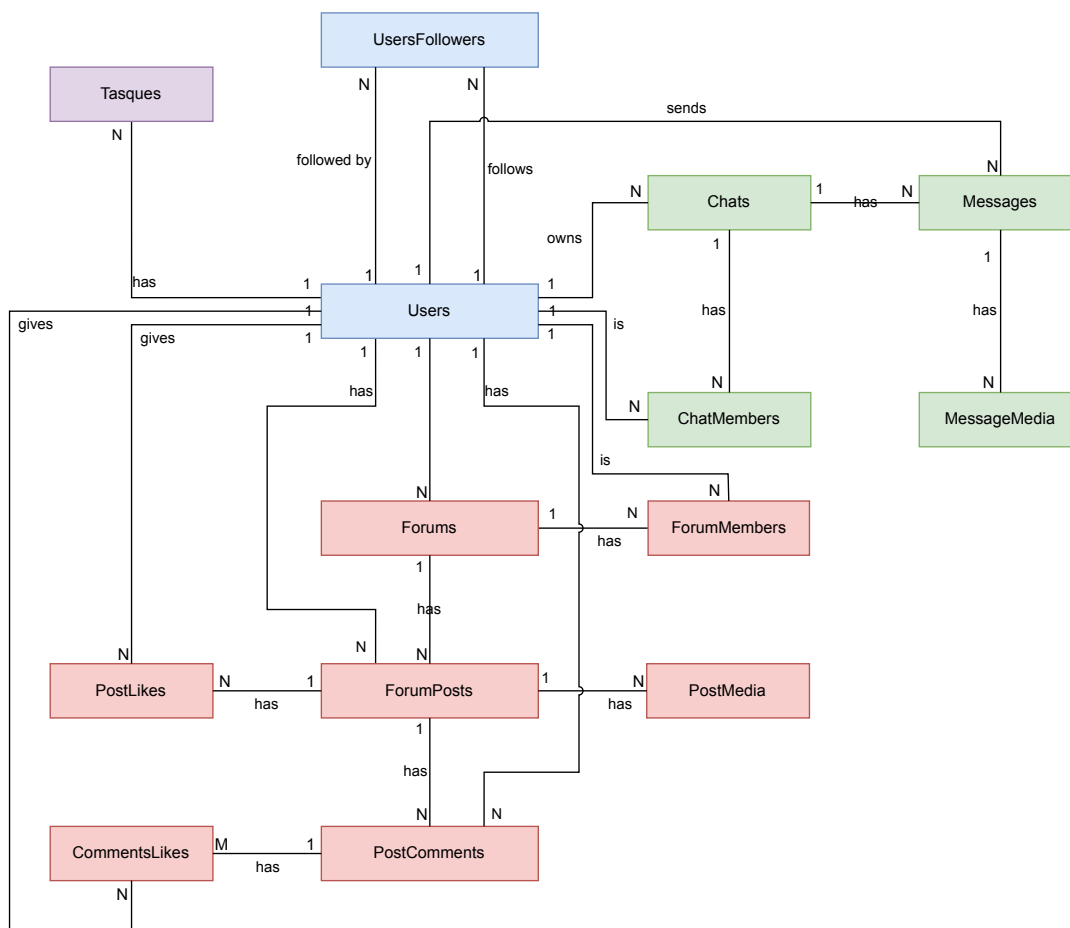


FIGURA 8.1: Diagrama del model conceptual de dades en el que es defineixen les associacions de les dades que formaran part de la base de dades

Al diagrama 8.1 podem observar les dades organitzades per mòdul de funcionalitat. En el diagrama podem veure: el mòdul d'usuari, representat per un color blau; el mòdul de fòrums, representat en color vermell; el mòdul de tasques, visualitzat amb

el color morat; i finalment, el mòdul de xats, associat al color verd. Es mantindrà aquests colors per representar cada mòdul al llarg d'aquesta secció per facilitar la lectura.

Així mateix, també podem visualitzar les relacions a la que estan sotmeses les dades del sistema. Per exemple podem veure que els usuaris es troben associats amb les tasques amb una relació 1:n (un usuari pot tenir múltiples tasques).

8.2.2 Funcionalitats

Llistats de funcionalitats

A continuació explorarem els requisits que l'aplicació ha de complir amb vista a satisfer les necessitats dels usuaris. A continuació es presentaran un conjunt de taules que definiran les diferents funcionalitats que l'aplicació hauria de disposar pels diferents actors. Per diferenciar cada tipus de funcionalitat, s'ha afegit una columna per identificar de quin mòdul de l'aplicació forma part.

Usuari No Registrat	
Funcionalitat	Mòdul
Registrar nou compte d'usuari	Usuaris

TAULA 8.1: Taula de funcionalitats requerides per l'actor *usuari no registrat*

Usuari Registrat	
Funcionalitat	Mòdul
Iniciar sessió	Usuaris
Tancar sessió	Usuaris
Visualitzar perfil d'usuari	Usuaris
Editar dades del perfil personal	Usuaris
Seguir un usuari (Afegir relació de seguiment amb un altre usuari)	Usuaris
Deixar de seguir un usuari (Eliminar relació de seguiment establerta amb un altre usuari)	Usuaris
Crear nou fòrum	Fòrums
Afegir publicació a un fòrum	Fòrums
Visualitzar llistat de publicacions	Fòrums
Afegir interacció de "m'agrada" a una publicació	Fòrums
Eliminar interacció de "m'agrada" a una publicació	Fòrums
Afegir comentari a una publicació	Fòrums
Visualitzar llistat de comentaris	Fòrums
Afegir interacció de "m'agrada" a un comentari	Fòrums
Eliminar interacció de "m'agrada" a un comentari	Fòrums
Crear tasca	Tasques
Visualitzar llista de tasques	Tasques
Editar tasca	Tasques
Marcar tasca com "completada"	Tasques
Desmarcar tasca com "completada"	Tasques
Eliminar tasca	Tasques

Obrir sales de xat de missatgeria directa amb un usuari seguit	Xats
Crear sala de xat de grup	Xats
Afegir usuari seguit a xats de grup	Xats
Enviar missatge a un xat	Xats
Visualitzar llistat de missatges d'un xat	Xats

TAULA 8.2: Taula de funcionalitats requerides per l'actor *usuari registrat*

Usuari Moderador de Fòrum	
Funcionalitat	Mòdul
Eliminar publicació	Fòrums
Eliminar comentari	Fòrums

TAULA 8.3: Taula de funcionalitats requerides per l'actor *usuari moderador de fòrum*

Usuari Administrador de Xat	
Funcionalitat	Mòdul
Editar dades del xat de grup	Xats
Eliminar missatge	Xats

TAULA 8.4: Taula de funcionalitats requerides per l'actor *usuari administrador de xat de grup*

Diagrama de casos d'ús

A partir de les funcionalitats descrites en les taules podem elaborar el diagrama de casos d'ús per als actors de l'aplicació. Primer el de l'actor *usuari no registrat*:

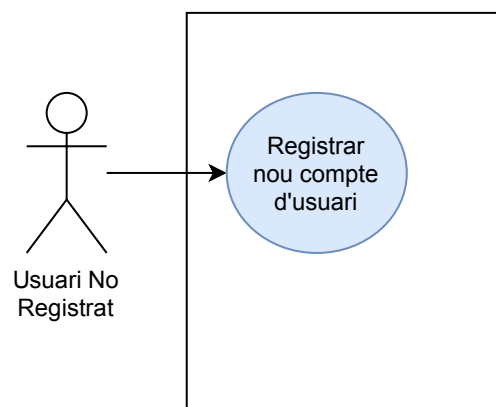


FIGURA 8.2: Diagrama casos d'ús per l'actor: Usuari No Registrat

Després l'actor *usuari registrat*:



FIGURA 8.3: Diagrama casos d'ús per l'actor: Usuari Registrat

Continuem amb l'actor *moderador de fòrum*

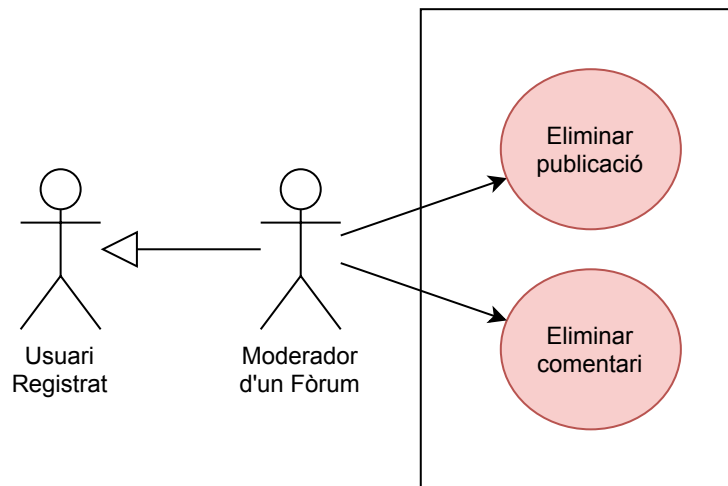


FIGURA 8.4: Diagrama casos d'ús per l'actor: Usuari Moderador d'un Fòrum

I, per finalitzar, el diagrama per l'actor *administrador de xat*

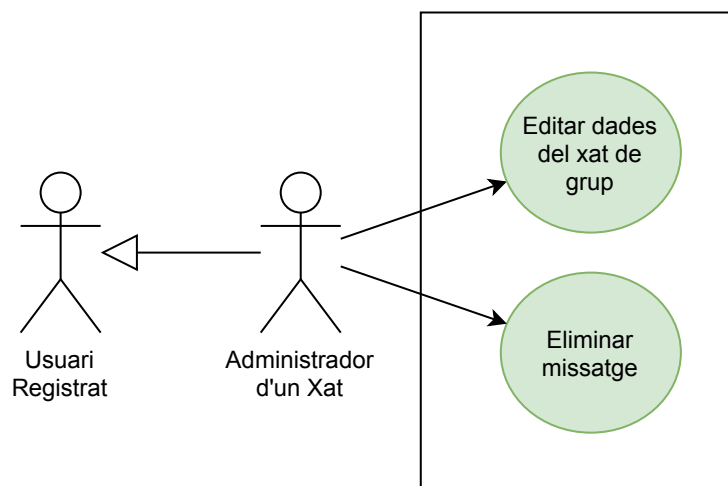


FIGURA 8.5: Diagrama casos d'ús per l'actor: Usuari Administrador d'un Xat

8.3 Disseny

8.3.1 Diagrama relacional de la base de dades

El diagrama de la figura 8.1 ens presentava l'associació entre les diferents dades de l'aplicació. A continuació s'expandirà el diagrama anterior, per tal de mostrar els continguts d'aquestes estructures de dades i les eines utilitzades per connectar les associacions.

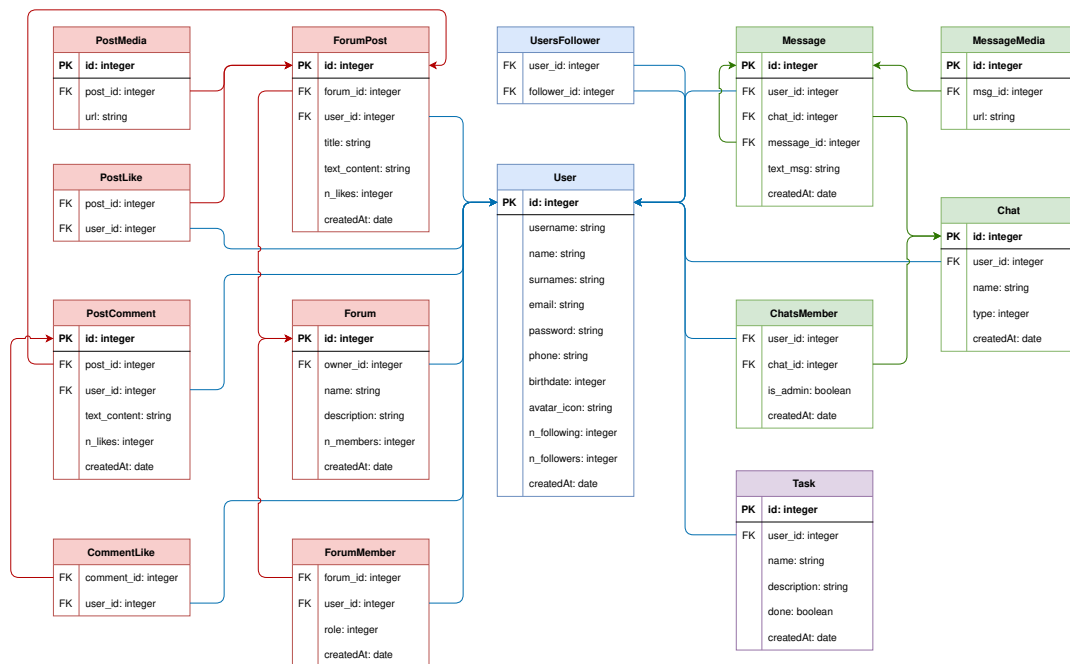


FIGURA 8.6: Diagrama relacional del model físic de dades per les entitats de la base de dades

En aquest diagrama podem veure la mateixa distribució de colors per delimitar cada mòdul de continguts. Així mateix, per distingir amb més facilitat les relacions entre les taules, també s'ha establert la mateixa distribució de color per les relacions que es dirigeixin a un mòdul. Per exemple, una clau forana que faci referència a una clau principal d'un usuari serà indicada de color blau, i una clau forana que faci referència a una clau principal d'un xat serà indicada de color verd.

Cadascun dels blocs de dades del diagrama representa una taula a la base de dades, i les propietats d'aquests blocs defineixen les columnes d'aquestes taules.

Per tal d'establir les associacions a la base de dades s'ha utilitzat les claus foranes. A l'hora de dissenyar relacions $n:m$, com pot ser la dels *Usuaris* i *Xats*, aquesta s'ha representat amb una taula intermèdia de membres del xat (*ChatMember*) per tal de reduir la relació $n:m$ a dues relacions $1:n$.

Algunes d'aquestes taules, com per exemple les dels models *PostMedia* i *MessageMedia*, han estat dissenyades pensant en el futur de l'aplicació amb funcionalitats que es volen implementar un cop finalitzada la implementació de les funcionalitats inicials del projecte.

Taula model *User*

La taula del model de dades *User* conté la informació personal relativa al compte dels usuaris. De la seva informació hem d'assegurar un cert nivell de seguretat per tal que atacants no puguin obtenir fàcilment la contrasenya de l'usuari, ja que amb aquesta podran accedir a la resta d'informació.

Per tal de donar una capa de seguretat al camp de la contrasenya, en comptes de guardar aquesta directament a la taula, s'utilitzen eines d'encryptació per a generar un *hash* de la contrasenya que es guardarà a la taula i dificultarà la recuperació d'aquesta per part dels atacants.

Aquesta taula també conté els comptadors de seguidors i usuaris seguits que ens permetrà tenir un recompte d'aquest més econòmic que no pas comptar la quantitat d'aquests cada cop que es vulguin saber.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un usuari.
username		<i>STRING</i>	Nom pel qual l'usuari serà reconegut a l'aplicació.
name		<i>STRING</i>	Nom de la persona darrere de l'usuari.
surnames		<i>STRING</i>	Cognoms de la persona darrere de l'usuari.
email		<i>STRING</i>	Adreça de correu electrònic de l'usuari.
password		<i>STRING</i>	<i>Hash</i> de la contrasenya de l'usuari per accedir al seu compte.
phone		<i>STRING</i>	Número de telèfon mòbil de l'usuari.
birthdate		<i>INTEGER</i>	Marca de temps en format UNIX <i>TIMESTAMP</i> de la data de naixement de l'usuari.
avatar_icon		<i>STRING</i>	URL del fitxer assignat per l'usuari com a avatar.
n_following		<i>INTEGER</i>	Comptador del nombre d'usuaris que segueix l'usuari.
n_followers		<i>INTEGER</i>	Comptador del nombre d'usuaris que segueixen a l'usuari.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació del compte de l'usuari.

TAULA 8.5: Descripció dels camps de la taula de la base de dades pel model de dades *User*

Taula model *UserFollower*

La taula del model *UserFollower* ens permet definir les relacions de seguiment entre els usuaris amb relacions 1:*n* en comptes de relacions *n:m*.

En afegir o eliminar entrades per aquesta taula, els comptadors de la taula d'usuaris relatius al nombre de seguidors per l'usuari seguit, i usuaris seguits per l'usuari seguidor, s'actualitzaran automàticament.

Nom del camp	Clau primària/forana	Tipus	Descripció
user_id	primària i forana	INTEGER	Identificador de l'usuari seguit a la taula d'usuaris.
follower_id	primària i forana	INTEGER	Identificador de l'usuari seguidor a la taula d'usuaris.

TAULA 8.6: Descripció dels camps de la taula de la base de dades pel model de dades *UserFollower*

Taula model *Forum*

La taula pels fòrums conté la informació bàsica d'un fòrum, així com quin usuari ha creat el fòrum, i el nombre de membres amb un comptador similar al del nombre de seguidors pel model d'usuaris.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	INTEGER	Valor únic autogenerat per la base de dades que identifica a un fòrum.
owner_id	forana	INTEGER	Identificador de l'usuari que va crear el fòrum.
name		STRING	Nom del fòrum.
description		STRING	Descripció del fòrum.
n_members		INTEGER	Comptador del nombre d'usuaris que són membres del fòrum.
createdAt		DATE	Camp autogenerat per <i>Sequelize</i> que indica la data de creació del fòrum.

TAULA 8.7: Descripció dels camps de la taula de la base de dades pel model de dades *Forum*

Taula model *ForumMember*

La taula del model *ForumMember* ens permet definir les relacions de pertinença dels usuaris amb un fòrum fent ús relacions 1:*n* en comptes de relacions *n:m*.

En aquesta taula també es conté el tipus de rol que l'usuari representa dins del fòrum pel qual és membre, com per exemple membre bàsic o moderador de fòrum.

En afegir o eliminar entrades per aquesta taula, el comptador de la taula de fòrums relatiu al nombre d'usuaris que són membres del fòrum s'actualitzarà automàticament.

Nom del camp	Clau primària/forana	Tipus	Descripció
forum_id	primària i forana	<i>INTEGER</i>	Identificador del fòrum.
user_id	primària i forana	<i>INTEGER</i>	Identificador de l'usuari membre del fòrum.
role		<i>INTEGER</i>	Identificador del rol de l'usuari al fòrum.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data en la qual l'usuari es va unir al fòrum.

TAULA 8.8: Descripció dels camps de la taula de la base de dades pel model de dades *ForumMember*

Taula model *ForumPost*

La taula del model *ForumPost* conté la informació relativa a la descripció i identificació de les publicacions dels fòrums i els usuaris creadors d'aquestes.

La taula també disposa d'un comptador de *likes* amb un funcionament similar al del comptador de membres d'un fòrum a la taula de fòrums.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a una publicació.
forum_id	forana	<i>INTEGER</i>	Identificador del fòrum al qual pertany la publicació.
user_id	forana	<i>INTEGER</i>	Identificador de l'usuari que ha creat la publicació.
title		<i>STRING</i>	Títol de la publicació.
text_content		<i>STRING</i>	Contingut de text de la publicació.
n_likes		<i>INTEGER</i>	Comptador del nombre de reaccions de "m'agrada" que els usuaris han donat a la publicació.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació de la publicació.

TAULA 8.9: Descripció dels camps de la taula de la base de dades pel model de dades *ForumPost*

Taula model *PostLike*

La taula del model *PostLike* ens permet definir les relacions entre les publicacions i les reaccions rebudes per part dels usuaris fent ús relacions 1:*n* en comptes de relacions *n:m*.

En afegir o eliminar entrades per aquesta taula, el comptador de la taula de publicacions relatiu al nombre de “m’agrada” que ha rebut una publicació s’actualitzarà automàticament.

Nom del camp	Clau primària/forana	Tipus	Descripció
post_id	primària i forana	<i>INTEGER</i>	Identificador de la publicació que rep la reacció.
user_id	primària i forana	<i>INTEGER</i>	Identificador de l’usuari que ha donat la reacció.

TAULA 8.10: Descripció dels camps de la taula de la base de dades pel model de dades *PostLike*

Taula model *PostMedia*

Un objectiu futur del projecte és la incorporació d’imatges i vídeos per les publicacions, per tal de mantenir un control de quins fitxers multimèdia pertanyen a quina publicació, es va dissenyar la taula del model de dades *PostMedia*, que conté els URL relatiu a fitxers multimèdia relacionats a una publicació per mitjà del seu identificador.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un fitxer multimèdia d’una publicació.
post_id	forana	<i>INTEGER</i>	Identificador de la publicació a la qual pertany el fitxer multimèdia.
url		<i>STRING</i>	Adreça on es troba guardat el contingut del fitxer multimèdia.

TAULA 8.11: Descripció dels camps de la taula de la base de dades pel model de dades *PostMedia*

Taula model *PostComment*

Els comentaris que els usuaris fan a les publicacions són emmagatzemats en la taula del model *PostComment*. Aquesta taula, a part del contingut del comentari, conté les relacions a la publicació a la qual s'ha fet el comentari, així com quin usuari ha fet el comentari, i el nombre de reaccions de "m'agrada" que ha rebut.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un comentari.
post_id	forana	<i>INTEGER</i>	Identificador de la publicació al qual pertany el comentari.
user_id	forana	<i>INTEGER</i>	Identificador de l'usuari que ha creat el comentari.
text_content		<i>STRING</i>	Contingut de text del comentari.
n_likes		<i>INTEGER</i>	Comptador del nombre de reaccions de "m'agrada" que els usuaris han donat al comentari.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació del comentari.

TAULA 8.12: Descripció dels camps de la taula de la base de dades pel model de dades *PostComment*

Taula model *CommentLike*

La taula del model *CommentLike*, de la mateixa manera que el model *PostLike*, ens permet definir les relacions entre els comentaris i les reaccions rebudes per part dels usuaris fent ús relacions 1:n en comptes de relacions n:m.

En afegir o eliminar entrades per aquesta taula, el comptador de la taula de comentaris relatiu al nombre de "m'agrada" que ha rebut un comentari s'actualitzarà automàticament.

Nom del camp	Clau primària/forana	Tipus	Descripció
comment_id	primària i forana	<i>INTEGER</i>	Identificador del comentari que rep la reacció.
user_id	primària i forana	<i>INTEGER</i>	Identificador de l'usuari que dona la reacció.

TAULA 8.13: Descripció dels camps de la taula de la base de dades pel model de dades *CommentLike*

Taula model *Task*

La taula del model *Task*, conté la informació relativa a les tasques dels usuaris del sistema, relacionant cada tasca amb un usuari.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a una tasca.
user_id	forana	<i>INTEGER</i>	Identificador de l'usuari que ha creat la tasca.
name		<i>STRING</i>	Nom de la tasca.
description		<i>STRING</i>	Descripció de la tasca.
done		<i>BOOLEAN</i>	Indicador de si la tasca ha estat completada o no.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació de la tasca.

TAULA 8.14: Descripció dels camps de la taula de la base de dades pel model de dades *Task*

Taula model *Chat*

La taula pel model de dades *Chats* conté la informació bàsica dels xats, així com quin usuari ha creat el xat i el tipus de xat que és. Els xats poden ser de tipus de missatgeria directa, entre un usuari i un altre, i de sala de grup, per la comunicació entre múltiples usuaris alhora.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un xat.
user_id	forana	<i>INTEGER</i>	Identificador de l'usuari que ha creat el xat.
name		<i>STRING</i>	Nom del xat.
type		<i>INTEGER</i>	Tipus del xat. Pot ser de missatgeria directa o de grup.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació del xat.

TAULA 8.15: Descripció dels camps de la taula de la base de dades pel model de dades *Chat*

Taula model *ChatsMember*

La taula del model *ChatsMember* ens permet definir les relacions de pertinença dels usuaris amb un xat fent ús relacions 1:n en comptes de relacions n:m.

En aquesta taula també es conté un camp per identificar si l'usuari és considerat un administrador del xat o no.

Nom del camp	Clau primària/forana	Tipus	Descripció
user_id	primària i forana	<i>INTEGER</i>	Identificador de l'usuari que pertany al xat.
chat_id	primària i forana	<i>INTEGER</i>	Identificador del xat al qual pertany l'usuari.
is_admin		<i>BOOLEAN</i>	Indicador de si l'usuari és un administrador del xat o no.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data en la qual l'usuari es va unir al xat.

TAULA 8.16: Descripció dels camps de la taula de la base de dades pel model de dades *ChatsMember*

Taula model *Message*

Els missatges dels xats s'emmagatzemen a la taula del model *Messages*. Aquesta taula conté els continguts textuais del missatge així com les relacions de quin usuari ha enviat el missatge, i a quin xat. A més, en un futur es vol afegir la funcionalitat de respondre a missatges als xats, per tant, també conté la relació amb un altre missatge.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un missatge.
user_id	forana	<i>INTEGER</i>	Identificador de l'usuari que ha enviat el missatge.
chat_id	forana	<i>INTEGER</i>	Identificador del xat on s'ha enviat el missatge.
message_id	forana	<i>INTEGER</i>	Identificador del missatge al qual respon el missatge.
text_msg		<i>STRING</i>	Contingut de text del missatge.
createdAt		<i>DATE</i>	Camp autogenerat per <i>Sequelize</i> que indica la data de creació de la tasca.

TAULA 8.17: Descripció dels camps de la taula de la base de dades pel model de dades *Message*

Taula model *MessageMedia*

De la mateixa manera que amb les publicacions, en un futur es vol poder enviar imatges i vídeos pels missatges d'un canal de xat. Per tal de mantenir el control de quin fitxer multimèdia pertany a quin missatge, es va dissenyar la taula del model *MessageMedia*, que conté els URL a fitxers multimèdia relacionats a un missatge d'un xat per mitjà del seu identificador.

Nom del camp	Clau primària/forana	Tipus	Descripció
id	primària	<i>INTEGER</i>	Valor únic autogenerat per la base de dades que identifica a un fitxer multimèdia d'un missatge.
msg_id	forana	<i>INTEGER</i>	Identificador del missatge a la qual pertany el fitxer multimèdia.
url		<i>STRING</i>	Adreça on es troba guardat el contingut del fitxer multimèdia.

TAULA 8.18: Descripció dels camps de la taula de la base de dades pel model de dades *MessageMedia*

8.3.2 Funcionalitats: Fitxes de cas d'ús i diagrames d'activitat

De la mateixa manera que s'ha fet en l'apartat anterior, expandint del diagrama del model de dades, en aquest apartat es farà el disseny dels processos relatius a les funcionalitats de l'aplicació en respecte als diagrames de casos d'ús: 8.2, 8.3, 8.4, i 8.5.

Registrar nou compte d'usuari

Cas d'ús	Registrar nou compte d'usuari.
Descripció	Un usuari no registrat crea un nou compte d'usuari per poder iniciar sessió a l'aplicació.
Actors	Usuari no registrat
Pre	Nou usuari no registrat.
Post	El nou usuari ha creat correctament un compte nou.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari visualitza la pàgina d'inici de sessió. 2. Usuari accedeix a la pàgina de registre de compte amb el botó per anar-hi. 3. Usuari entra les dades requerides al formulari pel registre del seu compte: Nom d'usuari, correu electrònic, contrasenya, confirmació de contrasenya. També pot entrar dades opcionals: Nom i cognoms. 4. Usuari envia el formulari amb les dades entrades. 5. Compte d'usuari és creada correctament i redirigit a la pantalla d'inici de sessió.
Flux Alternatiu	<ol style="list-style-type: none"> 1. Usuari envia el formulari sense alguna de les dades requerides, amb un format incorrecte, o les contrasenyes no coincideixen. 2. Es mostra un missatge d'error indicat quin és l'error ocorregut.

TAULA 8.19: Full del cas d'ús per registrar un compte d'usuari

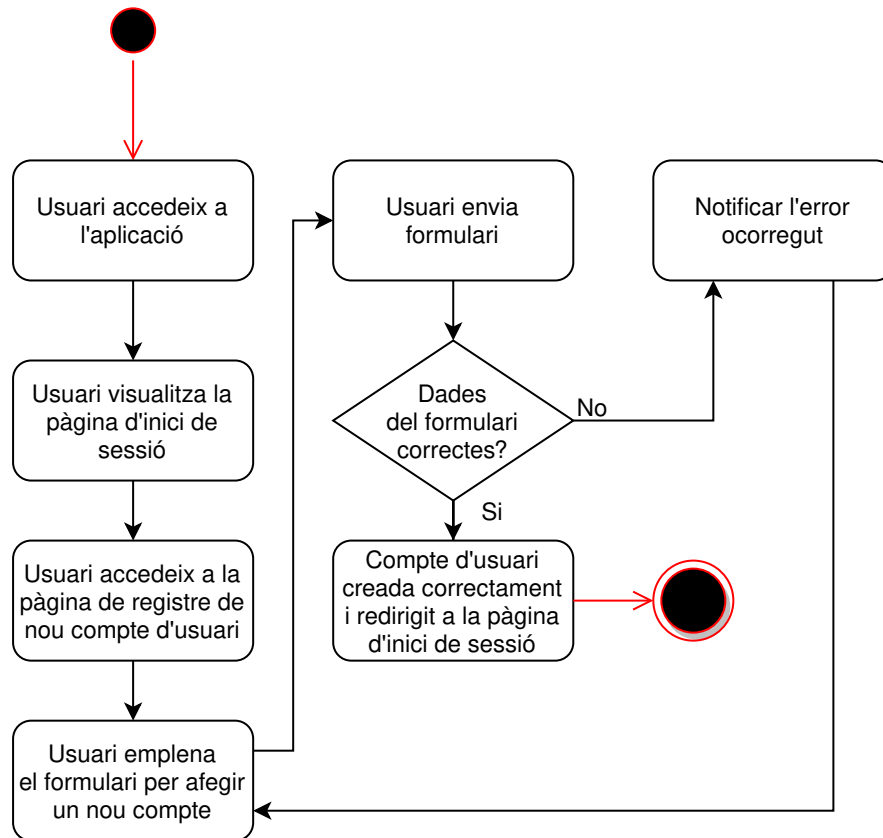


FIGURA 8.7: Diagrama d'activitat del cas d'ús per registrar un compte d'usuari

Iniciar sessió

Cas d'ús	Iniciar sessió.
Descripció	Usuari registrat inicia sessió amb les seves credencials.
Actors	Usuari registrat
Pre	Usuari disposa d'un compte registrat.
Post	Usuari inicia sessió correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari visualitza la pàgina d'inici de sessió. 2.Usuari entra les credencials (correu electrònic i contrasenya) al formulari d'inici de sessió. 3.Usuari envia el formulari. 4.Usuari inicia sessió correctament amb les credencials enviades.
Flux Alternatiu	<ol style="list-style-type: none"> 1.Usuari no entra algun dels camps del formulari o algun dels camps no coincideix amb les credencials guardades a la base de dades. 2. Es mostra un missatge d'error al formulari per indicar quin error ha ocorregut a l'usuari.

TAULA 8.20: Full del cas d'ús per l'inici de sessió

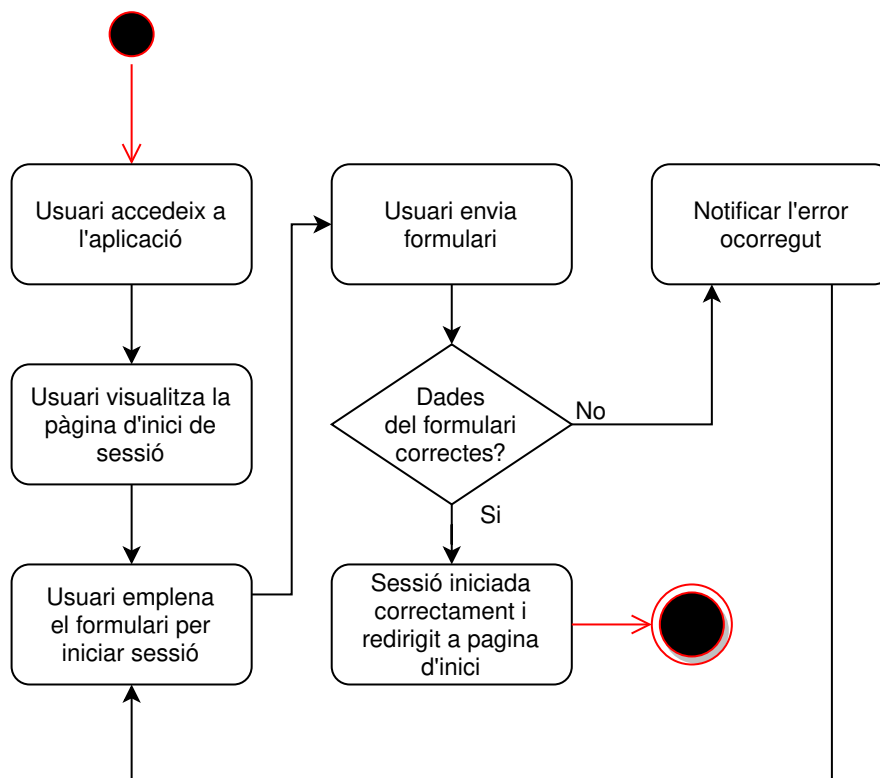


FIGURA 8.8: Diagrama d'activitat del cas d'ús per l'inici de sessió

Tancar sessió

Cas d'ús	Tancar sessió.
Descripció	Usuari tanca correctament la sessió.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari tanca la sessió correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari visualitza la pàgina principal de l'aplicació. 2.Usuari es dirigeix al seu perfil d'usuari. 3.Usuari prem el botó per tancar la sessió. 4.Sessió de l'usuari es tanca correctament.

TAULA 8.21: Full del cas d'ús per tancar la sessió

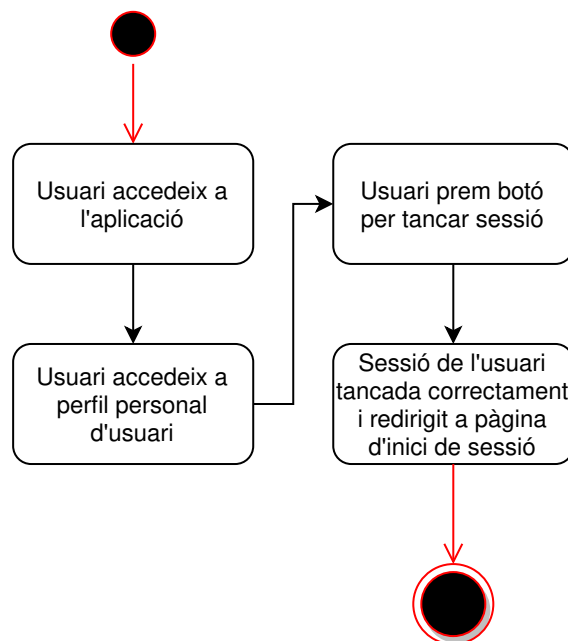


FIGURA 8.9: Diagrama d'activitat del cas d'ús per tancar la sessió

Visualitzar perfil d'usuari

Cas d'ús	Visualitzar perfil d'usuari.
Descripció	Usuari es dirigeix a un perfil d'usuari.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari visualitza les dades de l'usuari del perfil obert.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari prem algun enllaç al perfil d'algun usuari del sistema. 2.Usuari és redirigit al perfil de l'usuari premut i visualitza les seves dades.
Flux Alternatiu	<ol style="list-style-type: none"> 1.Usuari prem algun enllaç al seu perfil personal. 2.Usuari és redirigit al seu perfil personal correctament.

TAULA 8.22: Full del cas d'ús per visualitzar el perfil d'un usuari

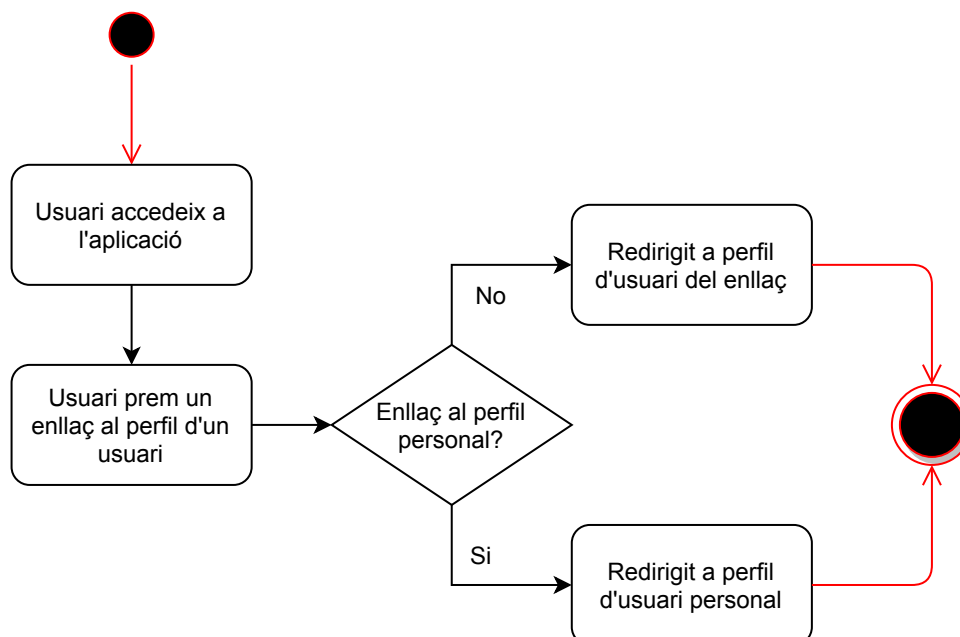


FIGURA 8.10: Diagrama d'activitat del cas d'ús per visualitzar el perfil d'un usuari

Editar dades del perfil personal

Cas d'ús	Editar dades del perfil personal.
Descripció	Usuari edita les dades personals i credencials del seu compte.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari ha editat les dades personals o credencials correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari prem enllaç a perfil personal. 2.Usuari prem enllaç a pàgina per editar dades personals. 3.Usuari modifica i emplena les dades que vol modificar al formulari per editar les dades del seu perfil. 4.Usuari envia les noves dades entrades al formulari. 5. Les dades de l'usuari s'actualitzen amb les noves dades entrades correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1. L'usuari entra alguna de les dades amb un format incorrecte o elimina alguna de les credencials requerides pel sistema. 2.Usuari intenta enviar el formulari. 3. Es mostra un missatge amb l'error ocorregut a l'usuari.

TAULA 8.23: Full del cas d'ús per editar les dades del perfil personal de l'usuari que ha iniciat sessió

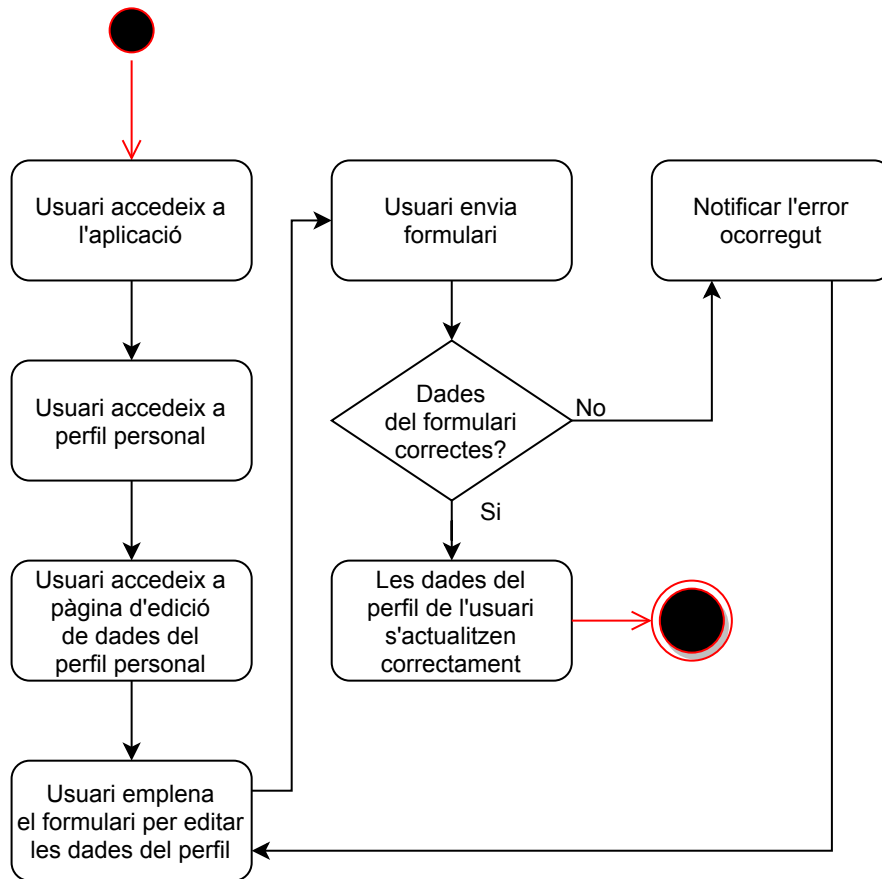


FIGURA 8.11: per editar les dades del perfil personal de l'usuari que ha iniciat sessió

Seguir usuari

Cas d'ús	Seguir usuari.
Descripció	Usuari afegeix una relació de seguiment amb un altre usuari del sistema.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari ha començat a seguir a l'altre usuari correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari prem un enllaç al perfil d'un usuari. 2.Usuari visualitza el perfil de l'usuari. 3.Usuari prem el botó per seguir a l'usuari. 4.Usuari comença a seguir a l'altre usuari correctament.

TAULA 8.24: Full del cas d'ús per seguir a un usuari

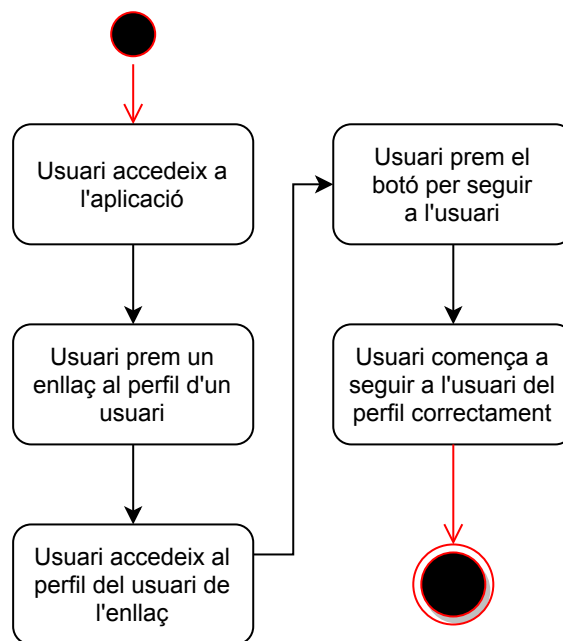


FIGURA 8.12: Diagrama d'activitat del cas d'ús seguir a un usuari

Deixar de seguir a un usuari

Cas d'ús	Deixar de seguir a un usuari.
Descripció	Usuari afegeix una relació de seguiment amb un altre usuari del sistema.
Actors	Usuari registrat
Pre	<ol style="list-style-type: none"> 1.Usuari ha iniciat sessió. 2.Usuari segueix a l'altre usuari.
Post	Usuari deixa de seguir a l'altre usuari correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari prem un enllaç al perfil d'un usuari. 2.Usuari visualitza el perfil de l'usuari. 3.Usuari prem el botó per deixar de seguir a l'usuari. 4.Usuari deixa de seguir a l'altre usuari correctament.

TAULA 8.25: Full del cas d'ús per deixar de seguir a un usuari

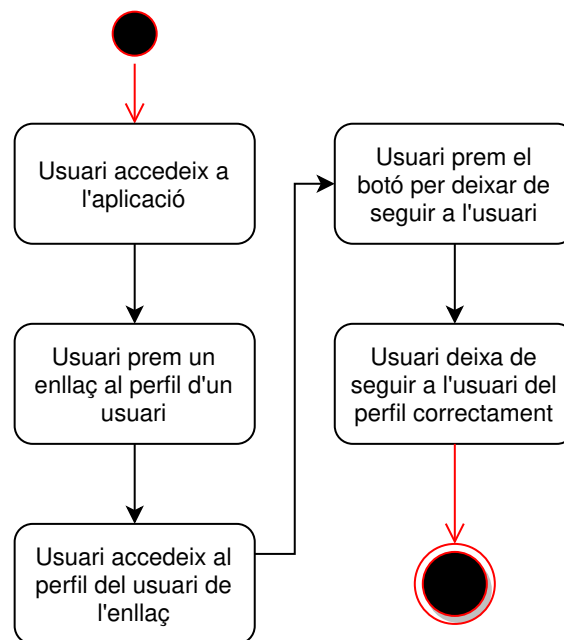


FIGURA 8.13: Diagrama d'activitat del cas d'ús per deixar de seguir a un usuari

Crear nou fòrum

Cas d'ús	Crear nou fòrum.
Descripció	Usuari crea un nou fòrum.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Nou fòrum creat correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix a la pàgina per crear una nova publicació. 2. Usuari selecciona opció per crear nou fòrum. 3. Usuari accedeix a la pàgina per crear un nou fòrum. 4. Usuari emplena el formulari amb les dades del fòrum: nom (obligatori), i descripció. 5. Usuari envia el formulari per crear el fòrum. 6. Fòrum es crea correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1. Usuari intenta enviar el formulari per crear el fòrum sense el camp de nom. 2. Es mostra un missatge amb l'error ocorregut.

TAULA 8.26: Full del cas d'ús per crear un nou fòrum

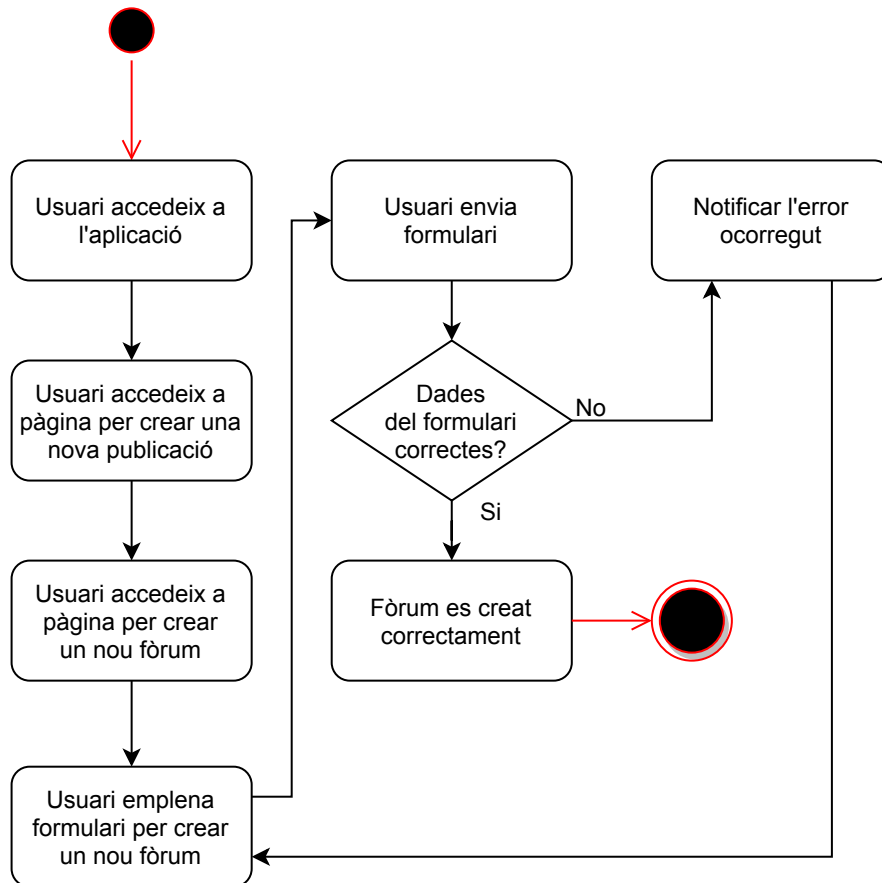


FIGURA 8.14: Diagrama d'activitat del cas d'ús per crear un nou fòrum

Afegir publicació a un fòrum

Cas d'ús	Afegir publicació a un fòrum.
Descripció	Creació i addició d'una nova publicació a un fòrum.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Nova publicació afegida al fòrum seleccionat.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina per crear una nova publicació. 2.Usuari emplena el formulari amb les dades relatives a la publicació: títol (obligatori), i contingut de text. 3.Usuari seleccionar fòrum on afegir la publicació. 4.Usuari envia formulari amb les dades entrades per crear una nova publicació. 5.Publicació es crea correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1.Usuari no entra el títol o no selecciona un fòrum on afegir la publicació. 2. Es mostra un missatge amb l'error ocorregut.

TAULA 8.27: Full del cas d'ús per crear una nova publicació i afegir-la a un fòrum

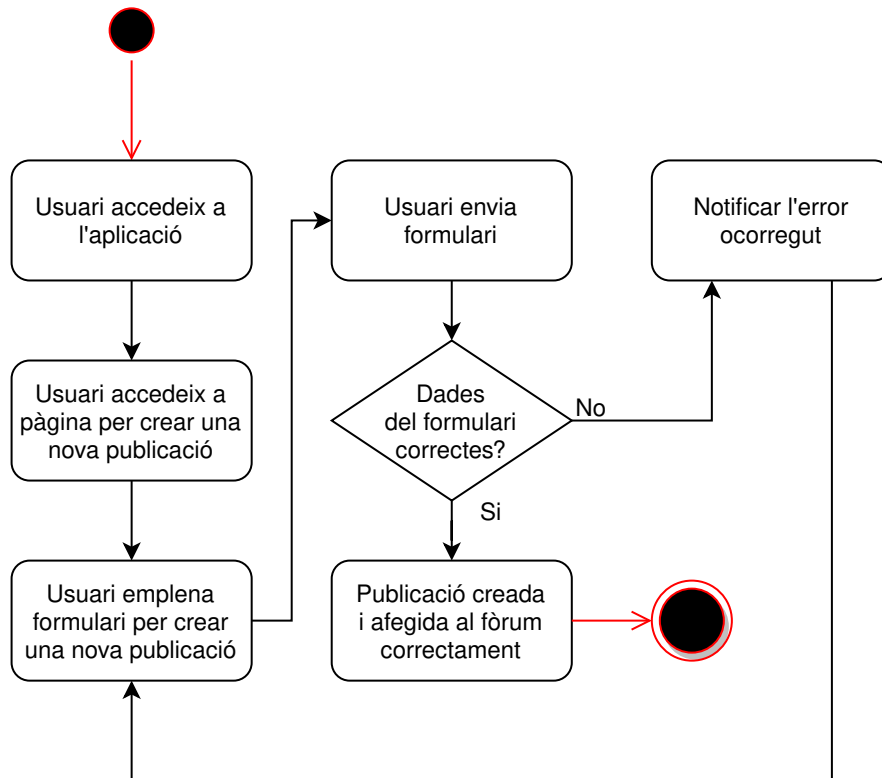


FIGURA 8.15: Diagrama d'activitat del cas d'ús per crear una nova publicació i afegir-la a un fòrum

Visualitzar llistat de publicacions

Cas d'ús	Visualitzar llistat de publicacions.
Descripció	Carrega i visualització del llistat de publicacions del sistema.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari visualitza el llistat de publicacions.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari accedeix a la pàgina principal de l'aplicació. 2. Es carrega la primera pàgina de publicacions. 3. L'usuari visualitza correctament les publicacions. 4. Si l'usuari es desplaça al final de la llista. <ol style="list-style-type: none"> (a) Si és l'última pàgina. <ul style="list-style-type: none"> • Finalitza procés del cas d'ús. (b) FiSi (c) Carrega la següent pàgina. (d) Retornar al pas 3. 5. FiSi

TAULA 8.28: Full del cas d'ús per la visualització de les publicacions del sistema

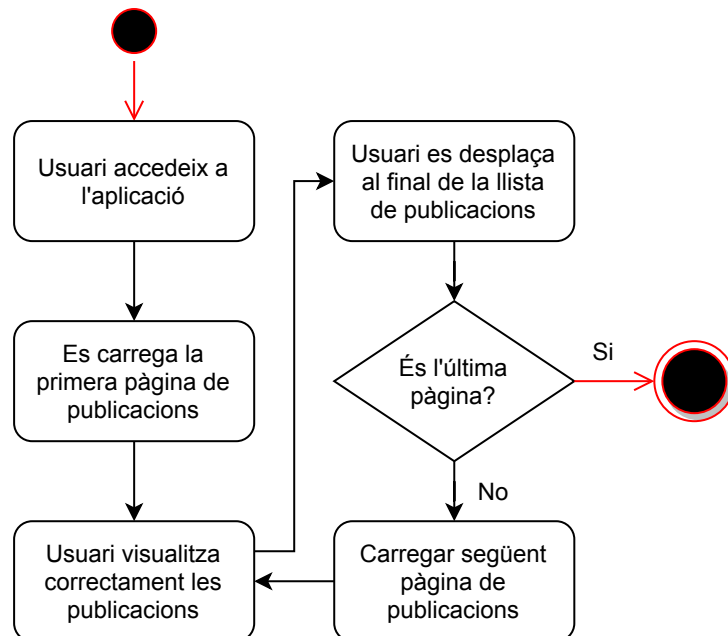


FIGURA 8.16: Diagrama d'activitat del cas d'ús per la visualització de les publicacions del sistema

Afegir interacció de "m'agrada" a una publicació

Cas d'ús	Afegir interacció de "m'agrada" a una publicació.
Descripció	Afegir una interacció de "m'agrada" per part de l'usuari a una publicació.
Actors	Usuari registrat
Pre	<ul style="list-style-type: none"> • Usuari ha iniciat sessió. • L'usuari no té marcada la publicació com "m'agrada".
Post	Usuari afegeix reacció de "m'agrada" a la publicació correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari visualitza la carta o pàgina d'una publicació. 2. Usuari prem el botó de "m'agrada" de la publicació. 3. S'afegeix la reacció de "m'agrada" de l'usuari per la publicació correctament.

TAULA 8.29: Full del cas d'ús per reaccionar a una publicació amb un "m'agrada"

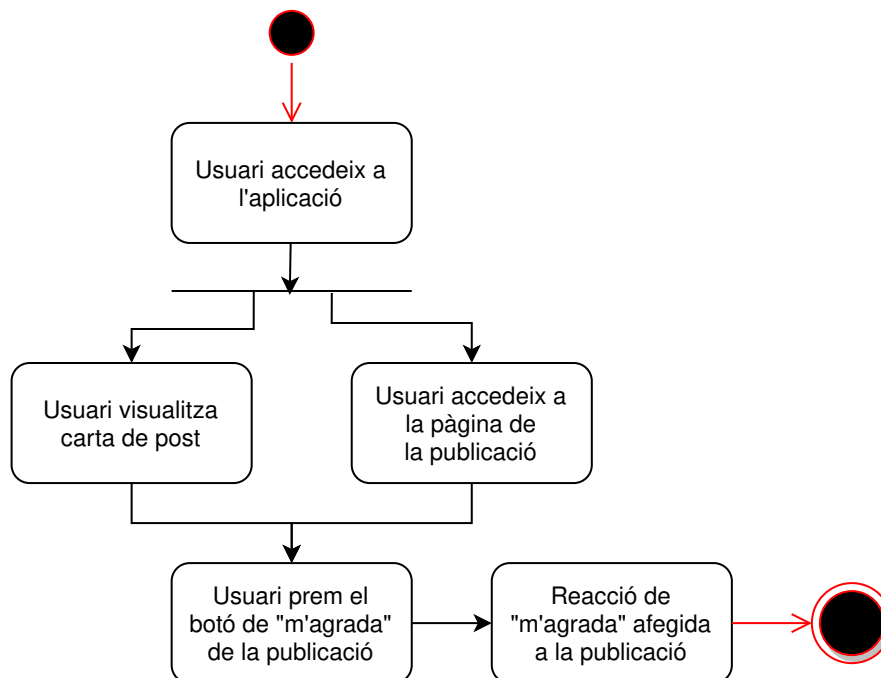


FIGURA 8.17: Diagrama d'activitat del cas d'ús per reaccionar a una publicació amb un "m'agrada"

Eliminar interacció de "m'agrada" a una publicació

Cas d'ús	Eliminar interacció de "m'agrada" a una publicació.
Descripció	Usuari elimina una interacció de "m'agrada" establerta prèviament a una publicació.
Actors	Usuari registrat
Pre	<ul style="list-style-type: none"> • Usuari ha iniciat sessió. • L'usuari té marcada la publicació com "m'agrada".
Post	Usuari elimina la seva reacció de "m'agrada" de la publicació correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari visualitza la carta o pàgina d'una publicació. 2. Usuari prem el botó de "m'agrada" de la publicació. 3. S'elimina la reacció de "m'agrada" de l'usuari per la publicació correctament.

TAULA 8.30: Full del cas d'ús per eliminar la reacció de "m'agrada" d'una publicació

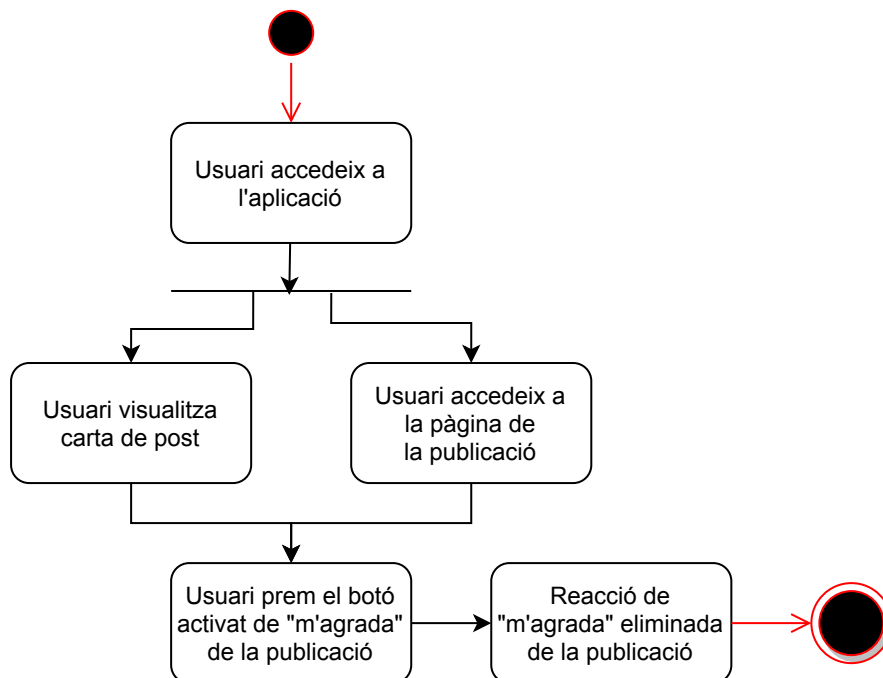


FIGURA 8.18: Diagrama d'activitat del cas d'ús per eliminar la reacció de "m'agrada" d'una publicació

Afegir comentari a una publicació

Cas d'ús	Afegir comentari a una publicació.
Descripció	Usuari afegeix un comentari a una publicació.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari afegeix un comentari a una publicació correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a pàgina d'una publicació. 2.Usuari emplena el formulari per afegir un nou comentari amb el contingut de text d'aquest. 3.Usuari envia formulari per afegir un nou comentari a la publicació. 4. Comentari es crea i afegeix correctament a la publicació.

TAULA 8.31: Full del cas d'ús per afegir un comentari a una publicació

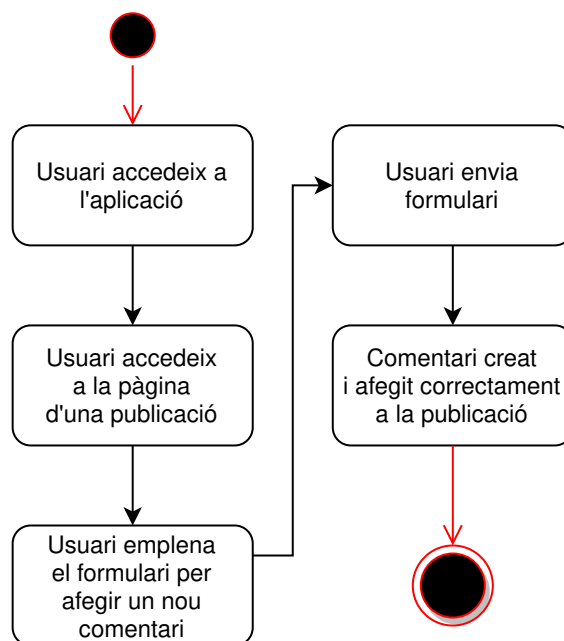


FIGURA 8.19: Diagrama d'activitat del cas d'ús per afegir un comentari a una publicació

Visualitzar llistat de comentaris d'una publicació

Cas d'ús	Visualitzar llista de comentaris.
Descripció	Carrega i visualització del llistat de comentaris d'una publicació.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari visualitza el llistat de publicacions.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina d'una publicació. 2. Es carrega la primera pàgina de comentaris. 3. L'usuari visualitza correctament els comentaris. 4. Si l'usuari es desplaça al final de la llista. <ol style="list-style-type: none"> (a) Si és l'última pàgina. <ul style="list-style-type: none"> • Finalitza procés del cas d'ús. (b) FiSi (c) Carrega la següent pàgina. (d) Retornar al pas 3. 5. FiSi

TAULA 8.32: Full del cas d'ús per la visualització dels comentaris d'una publicació

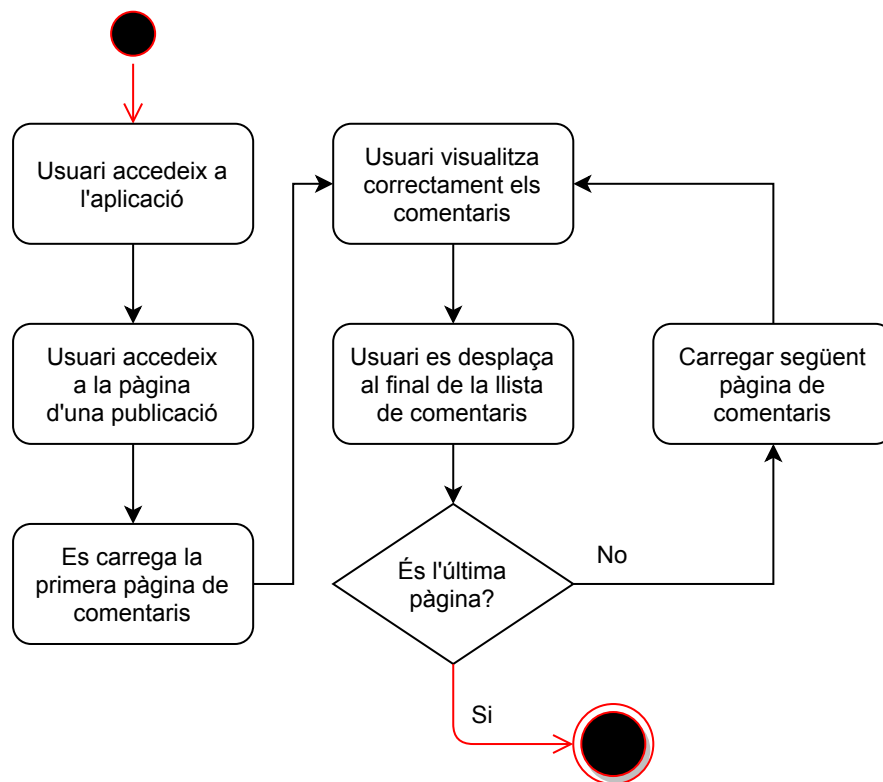


FIGURA 8.20: Diagrama d'activitat del cas d'ús per la visualització dels comentaris d'una publicació

Afegir interacció de "m'agrada" a un comentari

Cas d'ús	Afegir interacció de "m'agrada" a un comentari.
Descripció	Usuari afegeix una reacció de "m'agrada" a un comentari.
Actors	Usuari registrat
Pre	<ol style="list-style-type: none"> 1.Usuari ha iniciat sessió. 2. L'usuari no té marcat el comentari com "m'agrada".
Post	Usuari afegeix la reacció de "m'agrada" a un comentari correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina d'una publicació. 2.Usuari prem el botó de "m'agrada" d'un comentari. 3. S'afegeix la reacció de "m'agrada" de l'usuari pel comentari correctament.

TAULA 8.33: Full del cas d'ús per reaccionar a un comentari amb un "m'agrada"

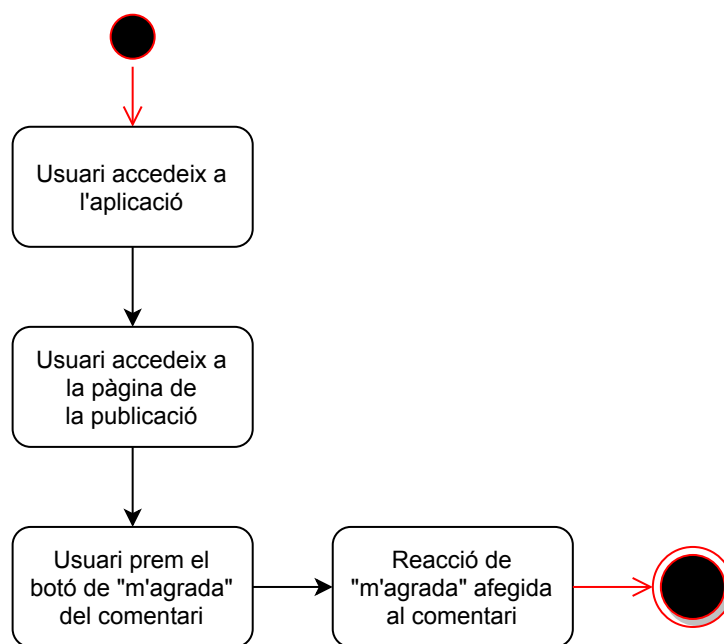


FIGURA 8.21: Diagrama d'activitat del cas d'ús per reaccionar a un comentari amb un "m'agrada"

Eliminar interacció de "m'agrada" a un comentari

Cas d'ús	Eliminar interacció de "m'agrada" a un comentari.
Descripció	Usuari elimina una interacció de "m'agrada" establerta prèviament a un comentari.
Actors	Usuari registrat
Pre	<ol style="list-style-type: none"> 1.Usuari ha iniciat sessió. 2. L'usuari té marcat el comentari com "m'agrada".
Post	Usuari elimina la seva reacció de "m'agrada" del comentari correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari visualitza la carta o pàgina d'una publicació. 2.Usuari prem el botó de "m'agrada" de la publicació. 3. S'elimina la reacció de "m'agrada" de l'usuari per la publicació correctament.

TAULA 8.34: Full del cas d'ús per eliminar la reacció de "m'agrada" d'un comentari

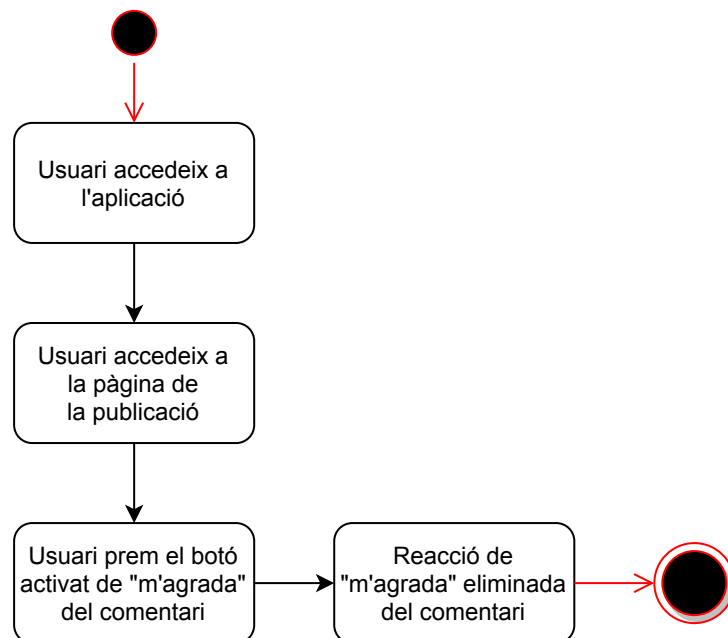


FIGURA 8.22: Diagrama d'activitat del cas d'ús per eliminar la reacció de "m'agrada" d'un comentari

Crear tasca

Cas d'ús	Crear tasca.
Descripció	L'usuari crea una nova tasca pel seu llistat de tasques.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Es crea la nova tasca pel llistat de tasques de l'usuari correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix al llistat de tasques. 2. Usuari prem el botó per afegir una nova tasca. 3. Usuari accedeix a pàgina per afegir tasca. 4. Usuari emplena el formulari per afegir una nova tasca amb: nom (obligatori), i descripció. 5. Usuari envia el formulari per crear nova tasca amb les dades entrades. 6. Tasca es crea i s'afegeix al llistat de tasques de l'usuari correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1. Usuari intenta enviar formulari sense emplenar el camp del nom de la tasca. 2. Es mostra un missatge amb l'error ocorregut.

TAULA 8.35: Full del cas d'ús per la creació d'una nova tasca

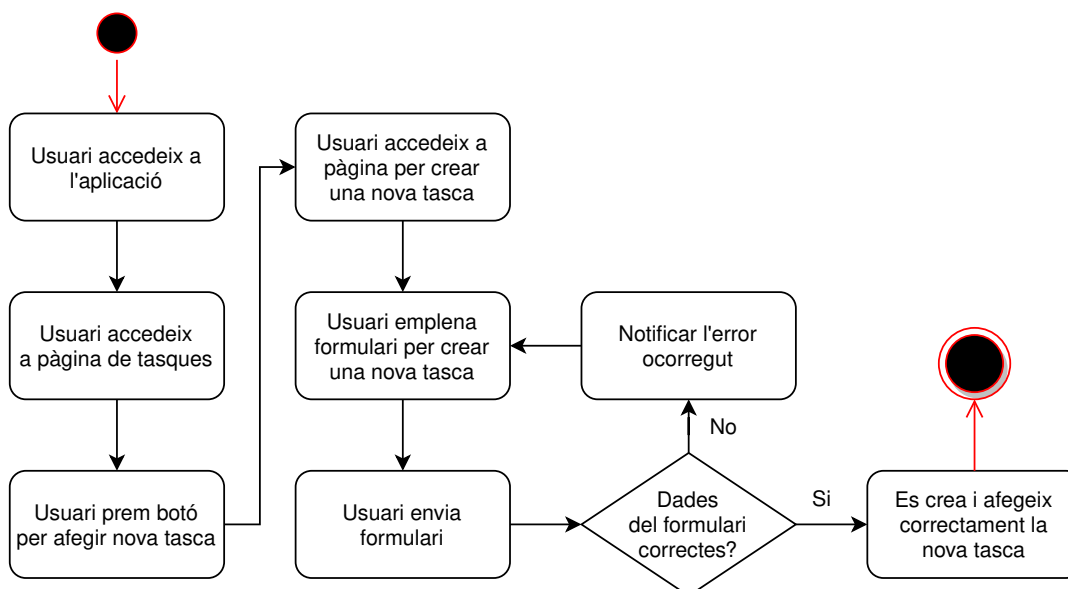


FIGURA 8.23: Diagrama d'activitat del cas d'ús per la creació d'una nova tasca

Visualitzar llistat de tasques

Cas d'ús	Visualitzar llistat de tasques.
Descripció	Carrega i visualització del llistat de tasques de l'usuari.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari visualitza el llistat de tasques correctament.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari accedeix a la pàgina de tasques de l'aplicació. 2. Es carrega la primera pàgina de tasques. 3. L'usuari visualitza correctament les tasques. 4. Si l'usuari es desplaça al final de la llista. <ol style="list-style-type: none"> (a) Si és l'última pàgina. <ul style="list-style-type: none"> • Finalitza procés del cas d'ús. (b) FiSi (c) Carrega la següent pàgina. (d) Retornar al pas 3. 5. FiSi

TAULA 8.36: Full del cas d'ús per la visualització de les tasques de l'usuari que ha iniciat sessió

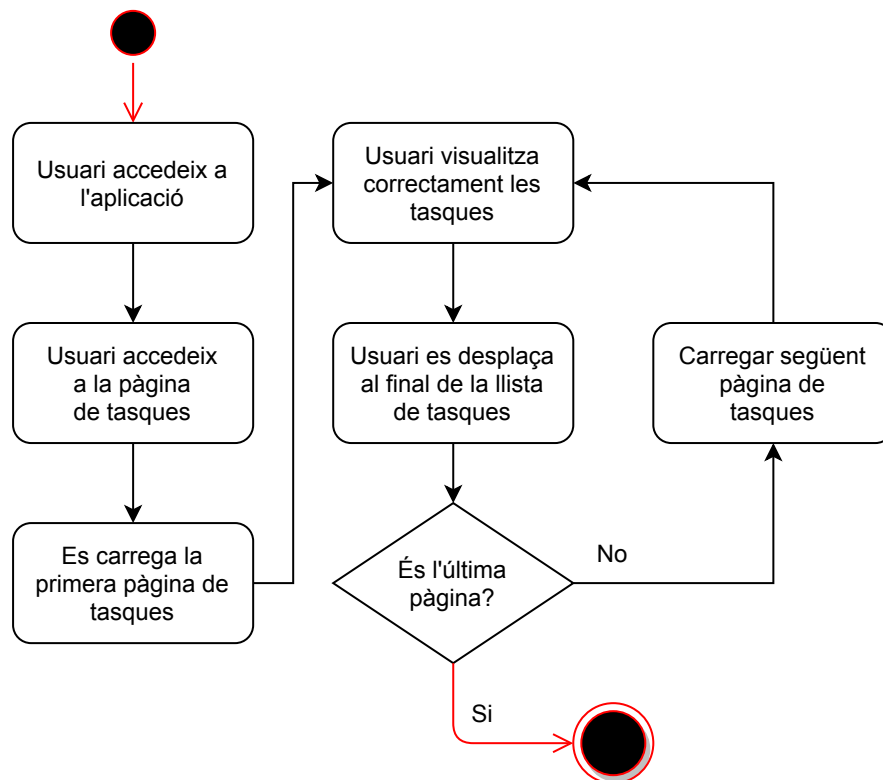


FIGURA 8.24: Diagrama d'activitat del cas d'ús per la visualització de les tasques de l'usuari que ha iniciat sessió

Editar tasca

Cas d'ús	Editar tasca.
Descripció	Usuari edita les dades relatives a una tasca.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari modifica les dades relatives a una tasca correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix a la pàgina del llistat de tasques. 2. Usuari prem en la carta d'una tasca. 3. Usuari accedeix a la pàgina d'edició de la tasca seleccionada. 4. Usuari modifica les dades del formulari de la tasca: nom i descripció. 5. Usuari envia formulari amb les noves dades per editar la tasca. 6. Les dades de la tasca editada són actualitzades correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1. Usuari intenta enviar formulari sense emplenar el camp nom. 2. Es mostra un missatge amb l'error ocorregut a l'usuari.

TAULA 8.37: Full del cas d'ús per l'edició dels continguts d'una tasca

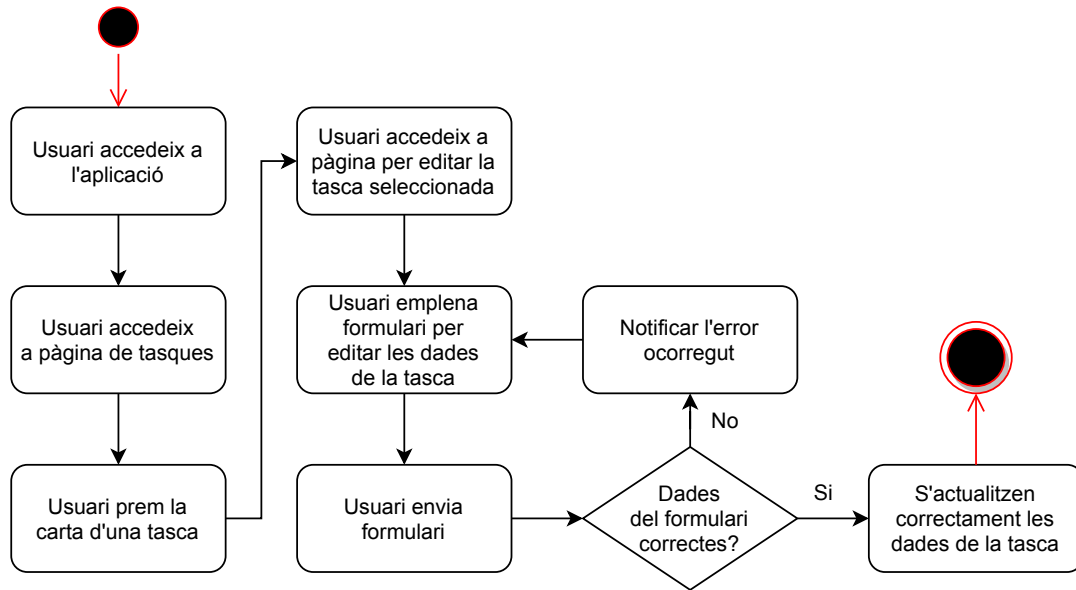


FIGURA 8.25: Diagrama d'activitat del cas d'ús per l'edició dels continguts d'una tasca

Marcar tasca com "completada"

Cas d'ús	Marcar tasca com "completada".
Descripció	Usuari marca una tasca com "completada".
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari marca una tasca com "completada" correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina del llistat de tasques. 2.Usuari prem el botó per marcar l'estat de la tasca. 3. Tasca canvia d'estat a "completada" correctament.

TAULA 8.38: Full del cas d'ús per marcar l'estat d'una tasca com a "completada"

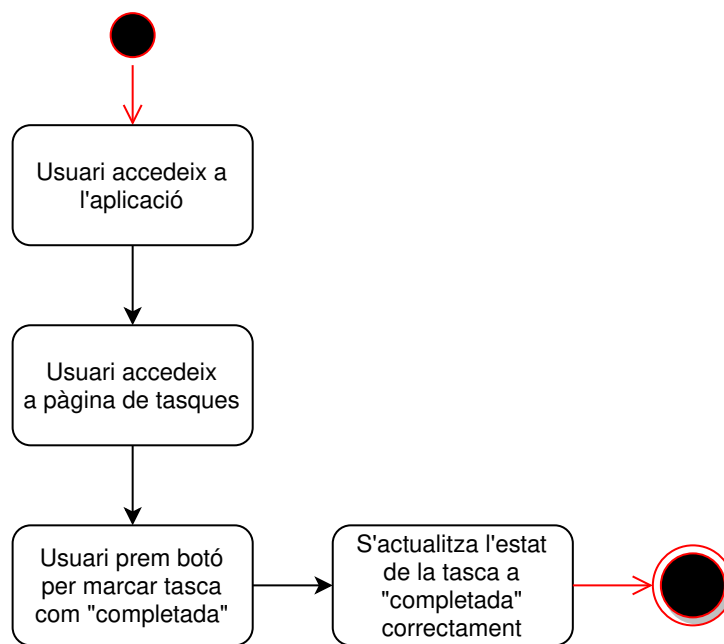


FIGURA 8.26: Diagrama d'activitat del cas d'ús per marcar l'estat d'una tasca com a "completada"

Desmarcar tasca com "completada"

Cas d'ús	Desmarcar tasca com "completada".
Descripció	Usuari elimina l'estat de "completada" a una tasca.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari marca una tasca com "completada" correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina del llistat de tasques. 2.Usuari prem el botó per marcar l'estat de la tasca. 3. Tasca canvia d'estat a "no completada" correctament.

TAULA 8.39: Full del cas d'ús per desmarcar l'estat d'una tasca de "completada" a "no completada"

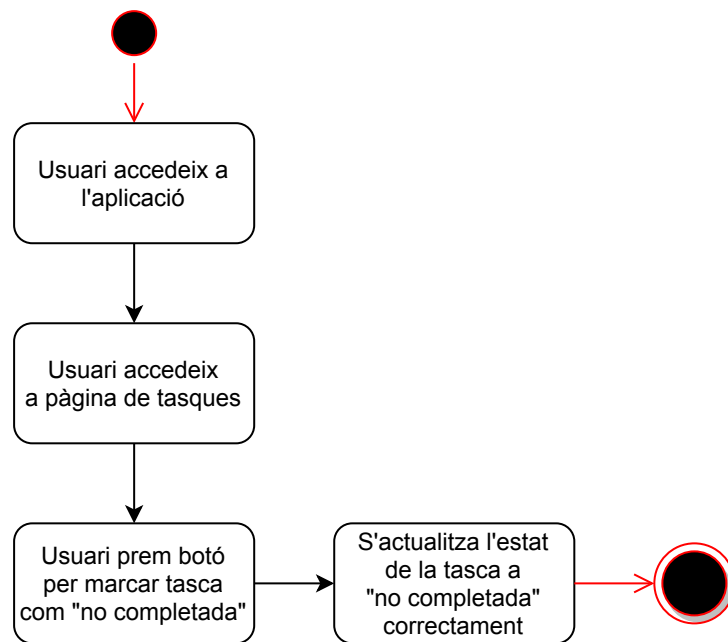


FIGURA 8.27: Diagrama d'activitat del cas d'ús per desmarcar l'estat d'una tasca de "completada" a "no completada"

Eliminar tasca

Cas d'ús	Eliminar tasca.
Descripció	Usuari elimina una tasca.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari elimina una tasca correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina del llistat de tasques. 2.Usuari prem el botó per eliminar una tasca. 3. Es mostra avis de confirmació de l'usuari. 4. Si l'usuari confirma l'acció. <ul style="list-style-type: none"> • Tasca s'elimina del llistat i del sistema correctament. 5. Altrament. <ul style="list-style-type: none"> • Es cancel·la l'eliminació de la tasca i acaba el cas d'ús. 6. FiSi

TAULA 8.40: Full del cas d'ús per l'eliminació d'una tasca de l'usuari

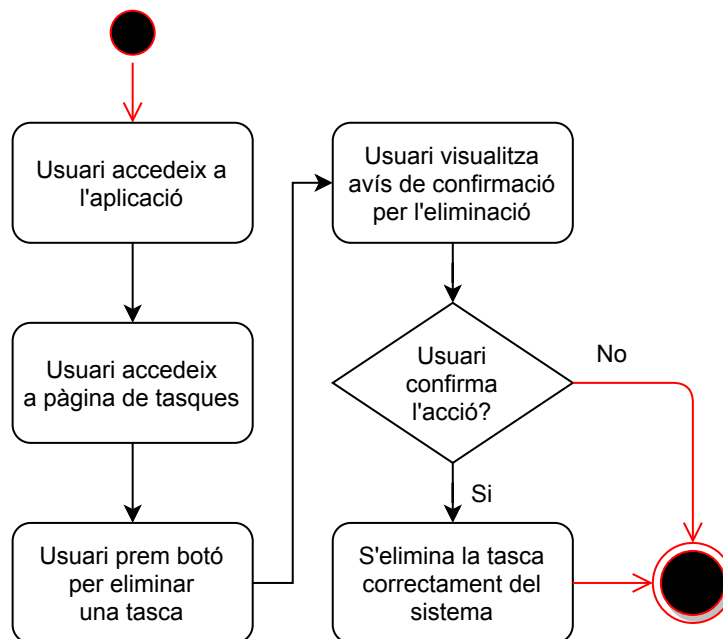


FIGURA 8.28: Diagrama d'activitat del cas d'ús per l'eliminació d'una tasca de l'usuari

Obrir sales de xat de missatgeria directa amb un usuari seguit

Cas d'ús	Obrir sales de xat de missatgeria directa amb un usuari seguit.
Descripció	Usuari obre un canal de missatgeria directa amb un usuari al qual segueix.
Actors	Usuari registrat
Pre	<ol style="list-style-type: none"> 1.Usuari ha iniciat sessió. 2.Usuari segueix a l'usuari pel qual es vol obrir el canal.
Post	Canal de missatgeria directa entre els usuaris creat correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix al perfil de l'usuari. 2.Usuari prem el botó per dirigir-se al canal de missatgeria directa amb l'altre usuari. 3. Si el canal no existeix. <ol style="list-style-type: none"> (a) Es crea el canal de missatgeria directa amb l'usuari correctament. 4. FiSi 5.Usuari és redirigit a la sala de xat pel canal de missatgeria directa amb l'usuari.

TAULA 8.41: Full del cas d'ús per accedir al canal de missatgeria directa de l'usuari que ha iniciat sessió amb un altre usuari

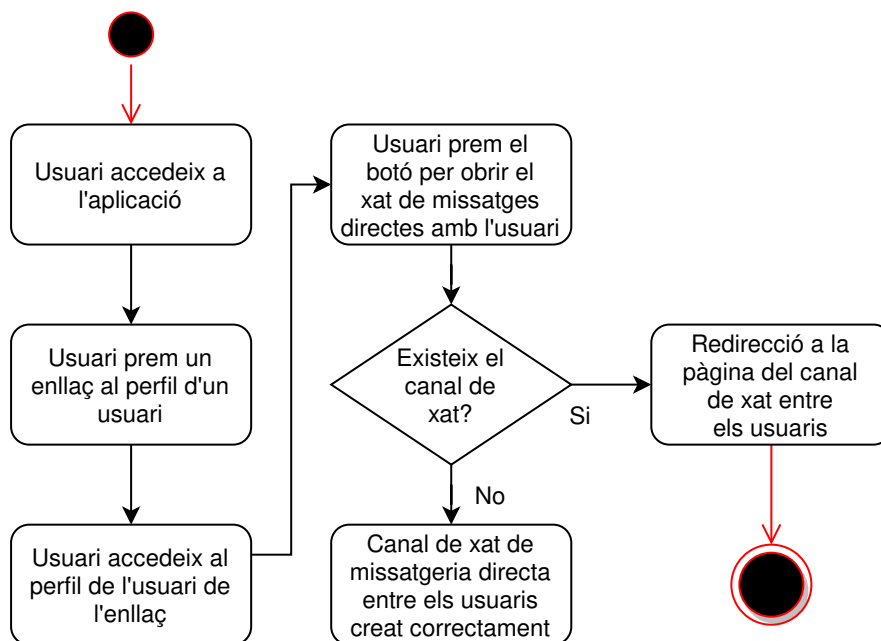


FIGURA 8.29: Diagrama d'activitat del cas d'ús per accedir al canal de missatgeria directa de l'usuari que ha iniciat sessió amb un altre usuari.

Crear sala de xat de grup

Cas d'ús	Crear sala de xat de grup.
Descripció	Usuari crea un xat grupal.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari crea un nou xat de grup correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina de xats de l'aplicació. 2.Usuari prem el botó per afegir un nou xat de grup. 3.Usuari accedeix a la pàgina de creació de xats de grup. 4.Usuari emplena el formulari per crear un xat de grup amb: el nom (obligatori). 5.Usuari envia el formulari per crear el xat de grup. 6. Xat de grup és creat correctament amb les dades del formulari.
Flux Alternatiu	<ol style="list-style-type: none"> 1.Usuari intenta enviar el formulari sense emplenar el camp de nom. 2. Es mostra un missatge amb l'error ocorregut.

TAULA 8.42: Full del cas d'ús per la creació d'un nou xat de grup

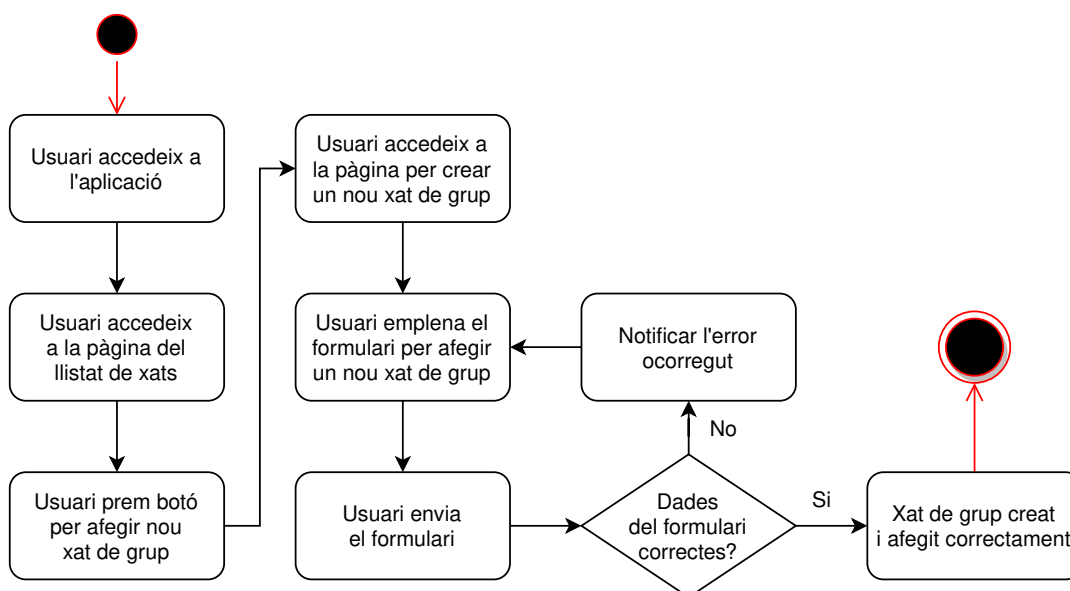


FIGURA 8.30: Diagrama d'activitat del cas d'ús per la creació d'un nou xat de grup

Afegir usuari seguit a xats de grup

Cas d'ús	Afegir usuari seguit a xats de grup.
Descripció	Usuari afegeix usuaris seguits a una sala de xat grupal.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari afegeix un usuari seguit com a nou membre del xat de grup correctament.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina de xats de l'aplicació. 2.Usuari accedeix a la sala d'un xat de grup. 3.Usuari prem l'opció de visualitzar la llista de membres del xat. 4.Usuari accedeix a la llista de membres del grup. 5.Usuari prem l'opció d'afegir nous membres. 6.Usuari accedeix a llista per afegir usuaris seguits al xat. 7.Usuari prem botó per afegir un usuari seguit al xat. 8.S'afegeix l'usuari seleccionat correctament al xat de grup.

TAULA 8.43: Full del cas d'ús per afegir un nou membre a un xat de grup

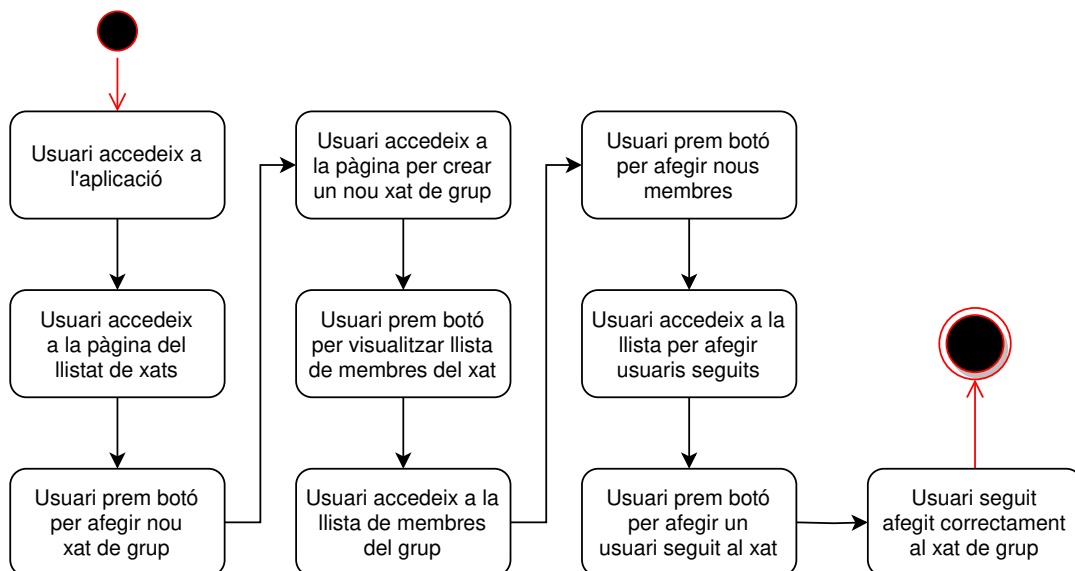


FIGURA 8.31: Diagrama d'activitat del cas d'ús per afegir un nou membre a un xat de grup

Enviar missatge a un xat

Cas d'ús	Enviar missatge a un xat.
Descripció	Usuari envia un missatge per un canal de xat a altres usuaris.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari afegeix un nou missatge a un xat correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix a la pàgina de xats de l'aplicació. 2. Usuari accedeix a la sala d'un xat. 3. Usuari empena el formulari per afegir un nou missatge amb el seu contingut de text. 4. Usuari envia el formulari per afegir el missatge. 5. El missatge s'afegeix correctament al text. 6. Es notifica als membres del xat del nou missatge.

TAULA 8.44: Full del cas d'ús per enviar un missatge per un canal de xat

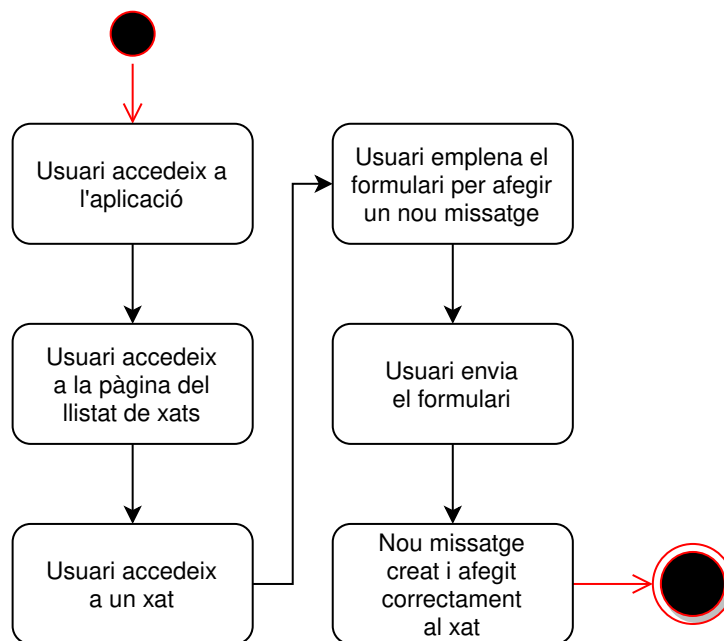


FIGURA 8.32: Diagrama d'activitat del cas d'ús per enviar un missatge per un canal de xat

Visualitzar llistat de missatges d'un xat

Cas d'ús	Visualitzar llistat de missatges d'un xat.
Descripció	Carrega i visualització del llistat de missatges d'un xat.
Actors	Usuari registrat
Pre	Usuari ha iniciat sessió.
Post	Usuari visualitza el llistat de missatges.
Flux Principal	<ol style="list-style-type: none"> 1.Usuari accedeix a la pàgina de xats de l'aplicació. 2.Usuari accedeix a la sala d'un xat. 3. Es carrega la primera pàgina de missatges. 4. L'usuari visualitza correctament els missatges. 5. Si l'usuari es desplaça al final de la llista. <ol style="list-style-type: none"> (a) Si és l'última pàgina. <ul style="list-style-type: none"> • Finalitza procés del cas d'ús. (b) FiSi (c) Carrega la següent pàgina. (d) Retornar al pas 4. 6. FiSi

TAULA 8.45: Full del cas d'ús per la visualització dels missatges d'un xat

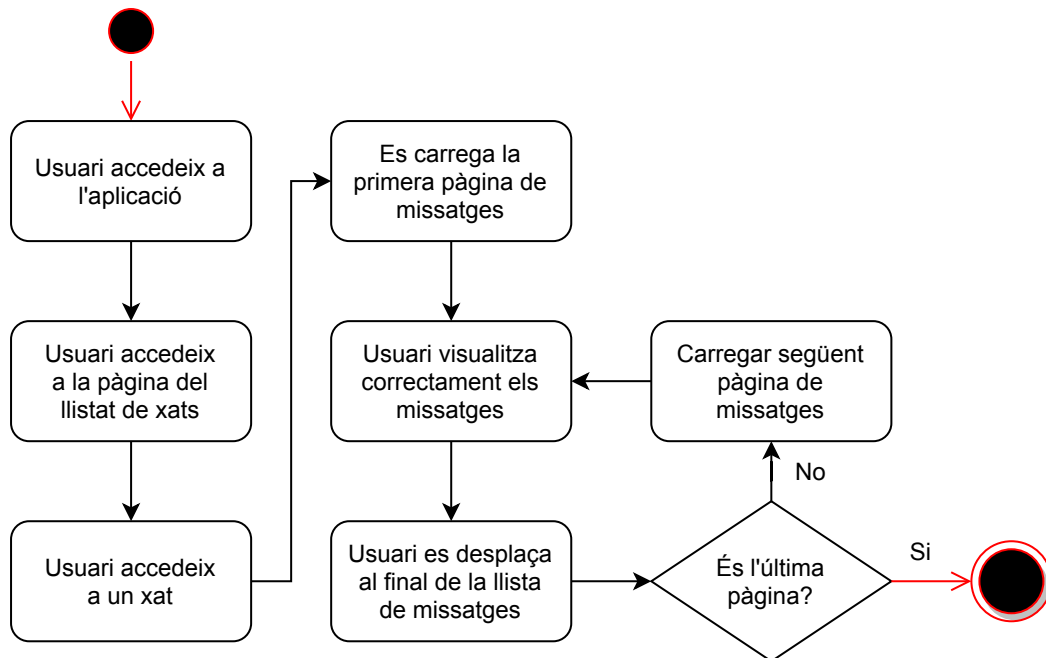


FIGURA 8.33: Diagrama d'activitat del cas d'ús per la visualització dels missatges d'un xat

Eliminar publicació

Cas d'ús	Eliminar publicació.
Descripció	Usuari amb permisos de moderació sobre un fòrum elimina una publicació.
Actors	Moderador d'un Fòrum
Pre	<ol style="list-style-type: none"> 1. Usuari ha iniciat sessió. 2. Usuari disposa de permisos de moderació sobre el fòrum de la publicació.
Post	Publicació eliminada correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Moderador accedeix a la pàgina d'una publicació. 2. Moderador prem el botó per eliminar la publicació. 3. Es mostra avis de confirmació. 4. Si el moderador confirma l'acció. <ul style="list-style-type: none"> • La publicació s'elimina del sistema correctament. 5. Altrament. <ul style="list-style-type: none"> • Es cancel·la l'eliminació de la publicació. 6. FiSi

TAULA 8.46: Full del cas d'ús per l'eliminació d'una publicació per part d'un moderador d'un fòrum

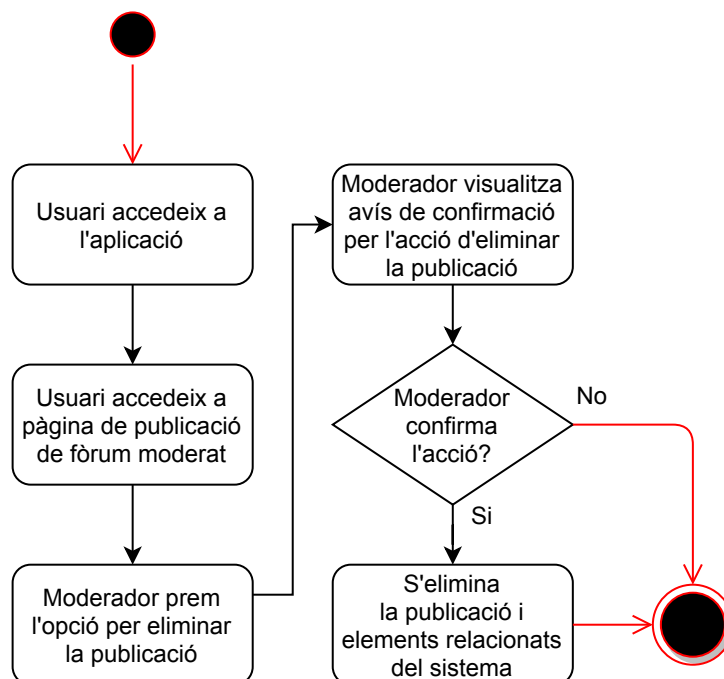


FIGURA 8.34: Diagrama d'activitat del cas d'ús per l'eliminació d'una publicació per part d'un moderador d'un fòrum

Eliminar comentari

Cas d'ús	Eliminar comentari.
Descripció	Usuari amb permisos de moderació sobre un fòrum elimina un comentari fet a una publicació.
Actors	Moderador d'un Fòrum
Pre	<ol style="list-style-type: none"> 1. Usuari ha iniciat sessió. 2. Usuari disposa de permisos de moderació sobre el fòrum de la publicació.
Post	Comentari eliminat correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Moderador accedeix a la pàgina d'una publicació. 2. Moderador selecciona un comentari. 3. Moderador prem el botó per eliminar el comentari. 4. Es mostra avis de confirmació. 5. Si el moderador confirma l'acció. <ul style="list-style-type: none"> • El comentari s'elimina del llistat i el sistema correctament. 6. Altrament. <ul style="list-style-type: none"> • Es cancel·la l'eliminació del comentari. 7. FiSi

TAULA 8.47: Full del cas d'ús per l'eliminació d'un comentari per part d'un moderador d'un fòrum

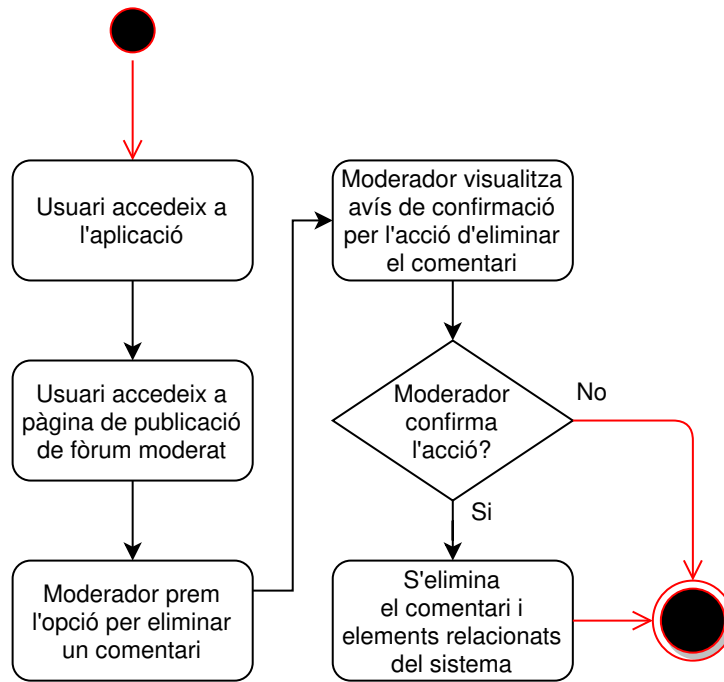


FIGURA 8.35: Diagrama d'activitat del cas d'ús per l'eliminació d'un comentari per part d'un moderador d'un fòrum

Editar dades del xat de grup

Cas d'ús	Editar dades del xat de grup.
Descripció	Usuari amb permisos d'administració sobre un xat modifica les dades relatives a aquest.
Actors	Administrador d'un Xat
Pre	<ol style="list-style-type: none"> 1. Usuari ha iniciat sessió. 2. Usuari disposa de permisos d'administració sobre el xat a editar.
Post	Dades del xat modificades.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix a la pàgina de xats de l'aplicació. 2. Usuari accedeix a la pàgina d'un xat on sigui Administrador. 3. Administrador prem el botó per editar les dades del xat. 4. Administrador accedeix a pàgina per editar les dades del xat. 5. Administrador emplena el formulari amb les dades relatives al xat: nom (obligatori). 6. Administrador envia el formulari per editar les dades del xat. 7. Les dades del xat s'actualitzen correctament.
Flux Alternatiu	<ol style="list-style-type: none"> 1. Usuari intenta enviar formulari sense el camp nom. 2. Es mostra un missatge amb l'error ocorregut.

TAULA 8.48: Full del cas d'ús per l'edició de les dades relatives a un xat

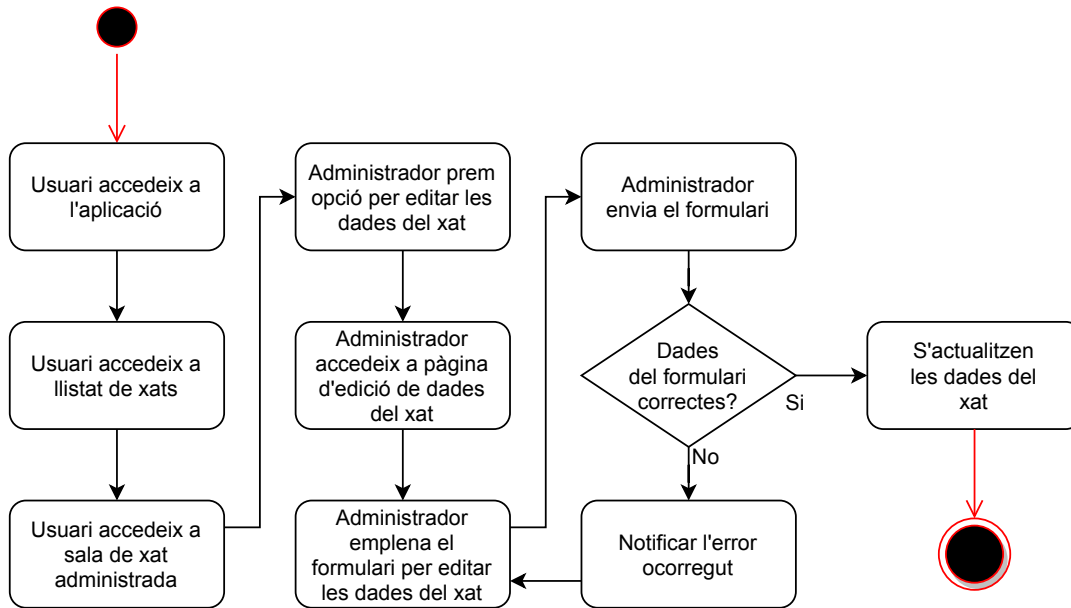


FIGURA 8.36: Diagrama d'activitat del cas d'ús per l'edició de les dades relatives a un xat

Eliminar missatge

Cas d'ús	Eliminar missatge.
Descripció	Usuari amb permisos d'administració sobre el xat elimina un missatge.
Actors	Administrador d'un Xat
Pre	<ol style="list-style-type: none"> 1. Usuari ha iniciat sessió. 2. Usuari disposa de permisos d'administració sobre el xat del missatge.
Post	Missatge eliminat correctament.
Flux Principal	<ol style="list-style-type: none"> 1. Usuari accedeix a la pàgina de xats de l'aplicació. 2. Usuari accedeix a la pàgina d'un xat on sigui Administrador. 3. Administrador selecciona un missatge. 4. Administrador prem el botó per eliminar el missatge. 5. Es mostra avis de confirmació. 6. Si l'administrador confirma l'acció. <ul style="list-style-type: none"> • El missatge s'elimina del llistat i el sistema correctament. 7. Altrament. <ul style="list-style-type: none"> • Es cancel·la l'eliminació del missatge. 8. FiSi

TAULA 8.49: Full del cas d'ús per l'eliminació d'un missatge per part d'un administrador d'un xat

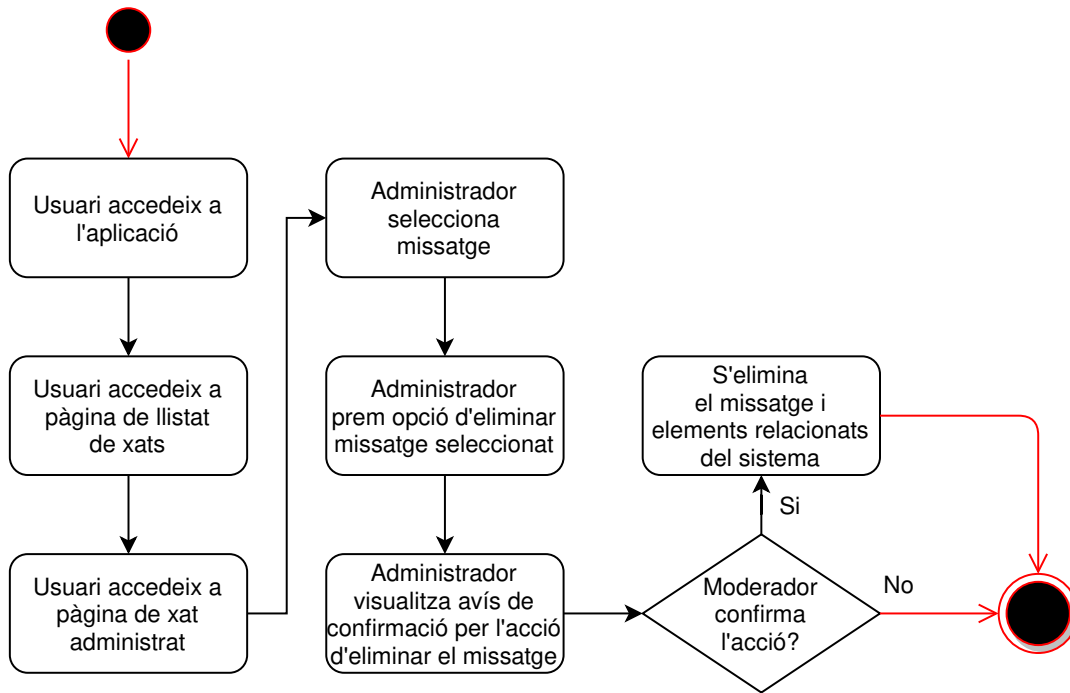


FIGURA 8.37: Diagrama d'activitat del cas d'ús per l'eliminació d'un missatge per part d'un administrador d'un xat.

8.3.3 Endpoints API

Una part crucial del desenvolupament d'una API és la definició dels *endpoints* d'aquesta. Per tal d'estructurar els endpoints per l'API desenvolupada pel projecte es va decidir fer una extensió de la divisió per mòduls pels casos d'ús. Aquesta divisió va resultar en les següents agrupacions d'endpoints:

- Mòdul **Usuaris**. Aquest mòdul veu repartit els seus endpoints en dos grups:
 - el grup *auth*, per les peticions relatives a l'autenticació dels usuaris,
 - i el grup *users*, amb les peticions relatives a la informació de l'usuari.
- Mòdul **Fòrums**. Aquest mòdul se subdivideix en 3 grups:
 - el grup *fòrums* que conté els endpoints relatius a la informació dels fòrums,
 - el grup *posts* que disposa de les rutes per les peticions relatives a les publicacions,
 - i finalment *comments* disposa dels endpoints per les peticions relatives als comentaris.
- Mòdul **Tasques**. El mòdul de les tasques és possible de mantenir en un mateix grup, per tant, els endpoints d'aquest mòdul es troben al grup *tasks*.
- Mòdul **Xats**. Igual que amb el mòdul de les tasques, el mòdul dels xats s'ha mantingut en un mateix grup d'endpoints: el grup *chats*.

8.3.4 Interfície d'usuari

Sens dubte, un dels punts més importants en desenvolupar una aplicació actualment és el disseny gràfic d'aquesta. Hi ha infinitat d'eines que disposen d'eines per dissenyar els esbossos inicials per la interfície d'usuari de l'aplicació a desenvolupar.

Si bé les eines proporcionades per aquest tipus de programari són molt útils, pel disseny de la interfície d'usuari de l'aplicació desenvolupada en aquest projecte s'ha preferit fer a paper. No hi ha un motiu especial darrere d'aquesta decisió fora de la comoditat de poder plasmar una idea visual en qualsevol moment i en qualsevol lloc sempre que es disposi d'un llapis i un full.

Per desgràcia la qualitat de les imatges escanejades no és bona, per tant, s'ha optat per fer la reconstrucció, amb programari d'edició d'imatge, d'alguns dels dissenys més significatius, que són els que es presentaran en aquest apartat. També s'han dissenyat alguns *assets* per l'aplicació que també es mostren a continuació.

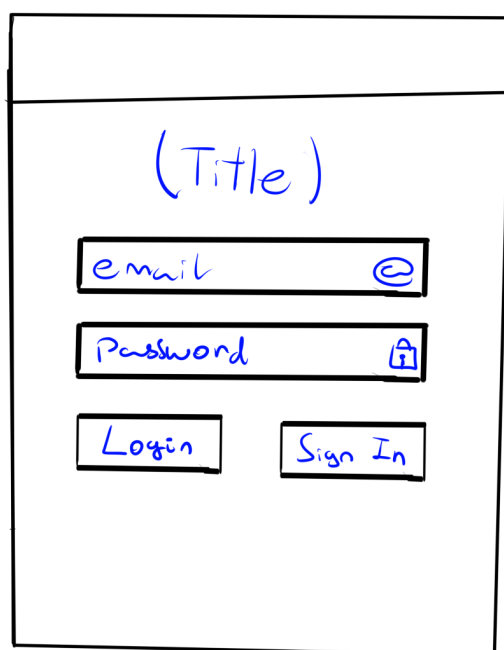


FIGURA 8.38: Recreació del dibuix conceptual per la pàgina d'inici de sessió de l'aplicació *Learnum* per dispositius mòbils



FIGURA 8.39: Recreació del dibuix conceptual per la pàgina per en-registrar un nou compte d'usuari per l'aplicació *Learnum* per dispositius mòbils

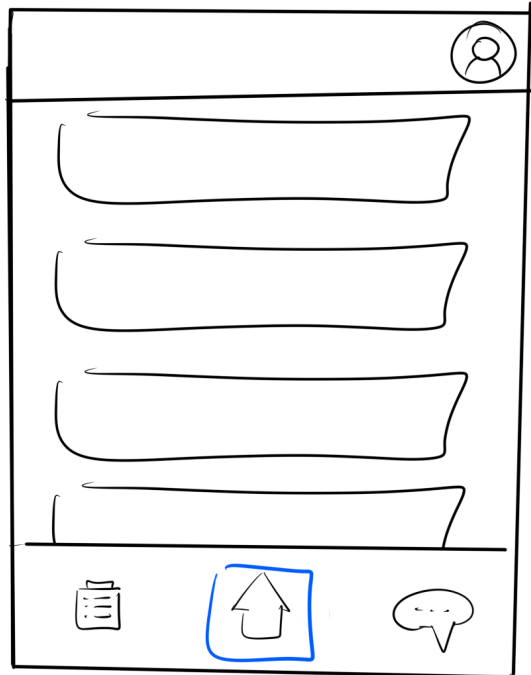


FIGURA 8.40: Recreació del dibuix conceptual per la pàgina d'inici de l'aplicació *Learnum* per dispositius mòbils

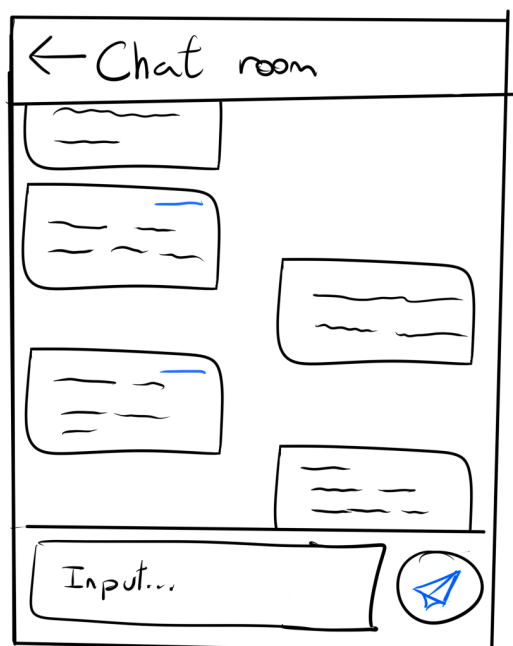


FIGURA 8.41: Recreació del dibuix conceptual per una pàgina d'una sala de xat de l'aplicació *Learnum* per dispositius mòbils



FIGURA 8.42: Icona per l'aplicació *Learnum*



FIGURA 8.43: Favicon per l'aplicació *Learnum*



FIGURA 8.44: Icona adaptativa per l'aplicació *Learnum*



FIGURA 8.45: Imatge per la *Splash Screen*

Capítol 9

Implementació i proves

En aquest capítol ens trobarem amb el cos principal del projecte. Aquí veurem des de com s'ha fet la configuració dels projectes amb els entorns, eines i control, fins a problemes que s'han trobat a l'hora de desenvolupar l'aplicació.

9.1 Configuració bàsica dels projectes

9.1.1 Node.js i npm

Si bé cada part de la plataforma Learnum utilitza eines pròpies per les necessitats que ens trobem a cadascuna, aquestes comparteixen una base comuna en *NodeJS* i *npm*. Aleshores, el primer pas per començar a treballar va ser la instal·lació del Framework i del gestor de paquets.

Per tal d'instal·lar tant NodeJS com npm en la màquina on s'ha desenvolupat el projecte (dispositiu *Windows*), es va fer servir l'instal·lador disponible a la pàgina de *NodeJS.org* [6]. Afortunadament, la instal·lació de NodeJS i npm va en conjunt amb el fitxer descarregat, així que, seguint les instruccions de l'executable (figura 9.1) es varen instal·lar tots dos correctament.

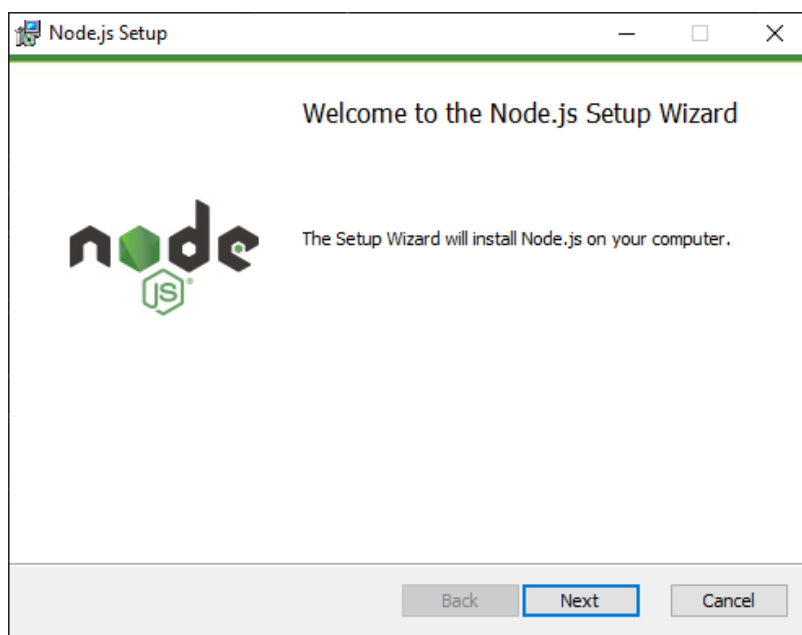


FIGURA 9.1: Instal·lador de *NodeJS* i *npm* per *Windows*

9.1.2 GitHub. Repositoris

A l'hora de mantenir un control sobre el contingut que s'està desenvolupant en vers del qual ja s'ha acabat de desenvolupar, els softwares de control de versions són eines molt útils. GitHub, el software de control de versions utilitzat en aquest projecte, disposa d'una gran potència en el sector pel qual és dirigit, facilitant la configuració d'un repositori a un parell de clics en una pàgina. Amb objectiu de mantenir un ordre per cada part del projecte, cada una d'aquestes parts (API, vista mòbils i vista d'escriptori) es va dividir en repositoris propis.

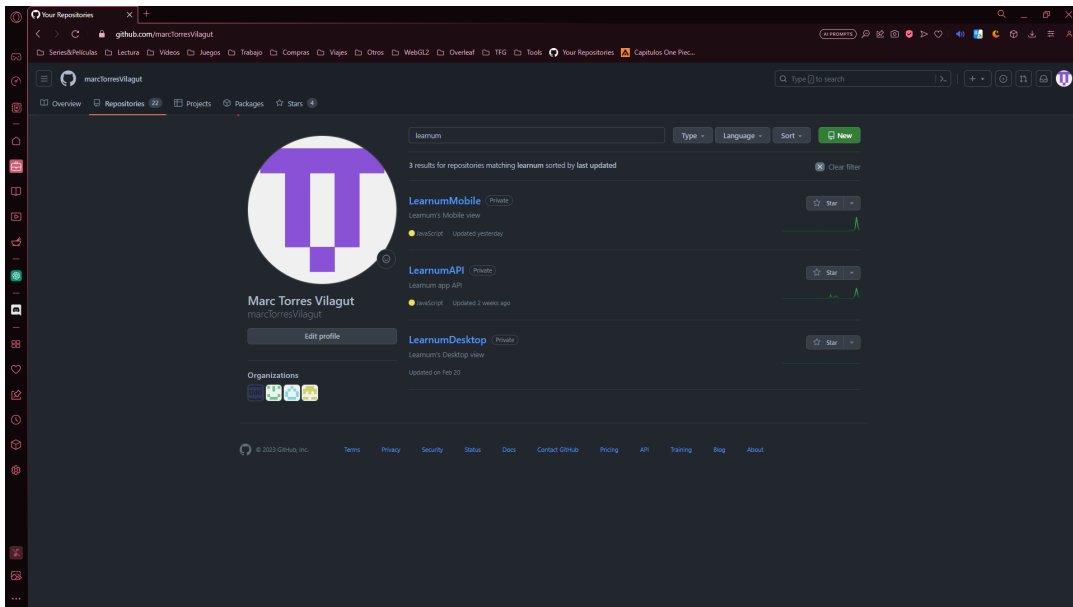


FIGURA 9.2: Captura de pantalla de la pàgina de GitHub amb els repositoris creats per cada part del projecte.

Un cop creats els repositoris, només queda clonar-los al dispositiu local i ja estaran disponibles per poder començar a desenvolupar. Per fer la clonació es va utilitzar l'eina de línia d'ordres *Git Bash*.

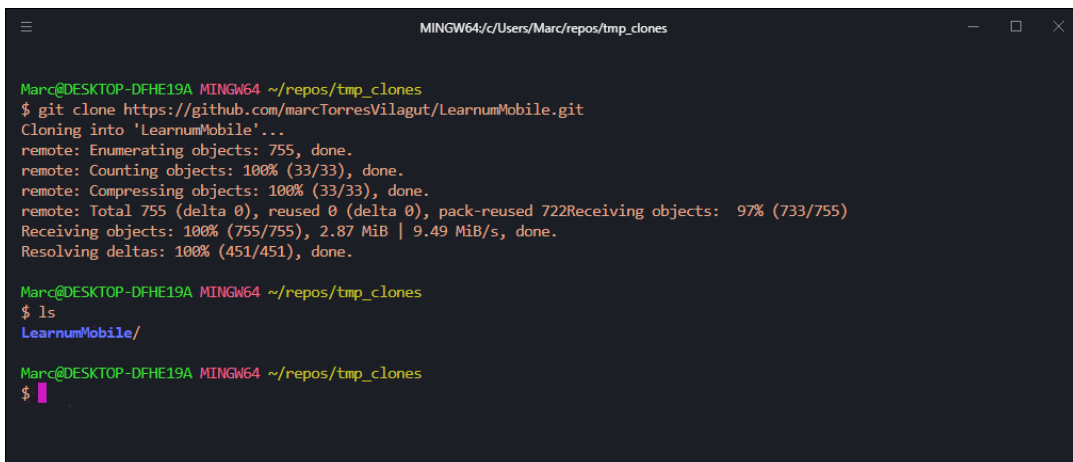


FIGURA 9.3: Captura de pantalla de la clonació del repositori *LearnumMobile* pel desenvolupament de la plataforma Learnum per dispositius mòbils.

9.1.3 Visual Studio Code. Entorns de treball

Per tal de preparar els entorns de treball de Visual Studio Code es varen instal·lar diferents extensions per facilitar el desenvolupament per les diferents parts de l'aplicació. Entre les diferents extensions instal·lades podem trobar algunes com:

- **ES7+ React/Redux/React-Native snippets.** Extensió per afegir *snippets* per agilitzar el procés de “picar codi”.
- **GitLens — Git supercharged.** Extensió que integra funcionalitats de GitHub/-GitBash en projectes inicialitzats com repositoris de GitHub.
- **Error Lens.** Extensió que permet visualitzar més fàcilment els errors al codi.

També es va aprofitar l'ús de *NodeJS* i *npm* per tal d'instal·lar dependències relatives a *ESLint*, un tipus d'eina coneguda com a *linters* que permet la millora de codi. Aquest tipus d'eina són molt convenients per establir un estil de codi en un projecte, ja que permet la definició de regles que, un cop es guarda un document, aquest és formatat amb les regles establertes. Un exemple podria ser: “Totes les cadenes de text han de ser definides amb cometes simples”. Si s'ha configurat correctament, un cop guardem un fitxer on existeixin cadenes de text amb cometes dobles, aquestes es convertiran en cometes simples.

Són un tipus d'eina potent, ja que pot solucionar molts problemes, i s'ha utilitzat també per mantenir el codi net i formatat de forma uniforme al llarg del projecte, facilitant la lectura d'aquest.

9.2 API

9.2.1 Configuració

Per donar suport a les explicacions del següent apartat, a l'apèndix A del projecte es pot trobar una captura de pantalla amb l'estructura utilitzada pels fitxers d'aquesta part del projecte (figura A.1).

A l'hora de posar en marxa el servidor de l'API REST hi ha diversos punts que s'han de tenir en consideració:

1. La recuperació de valors de l'entorn per la configuració del servidor.
2. La connexió amb la base de dades.
3. Preparació de les rutes (*endpoints*) per les peticions.
4. Inicialització del procés d'escolta per peticions.

Variables d'entorn

Per qüestions de seguretat, quan ens trobem en el desenvolupament d'una aplicació on la informació dels usuaris es trobarà viatjant constantment entre diversos dispositius i serveis, és indispensable mantenir un cert tipus de dades delicades amagades. Una forma d'obtenir aquest secretisme en aquestes dades són els fitxers *.env*, els

quals permeten la definició de variables d'entorn que només estaran disponibles en el dispositiu on es trobi desplegada l'aplicació.

Com per aquest projecte ens trobem en el cas mencionat, i per mantenir dades com les credencials per la connexió a la base de dades ocultes, es va definir un fitxer `.env` per mantenir les dades ocultes. El contingut d'aquest fitxer és el següent:

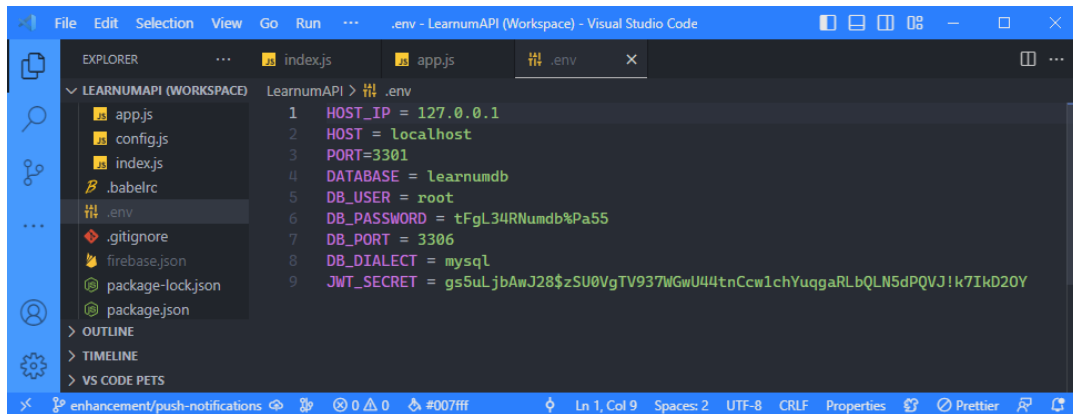


FIGURA 9.4: Captura de pantalla dels continguts del fitxer `.env` de l'entorn de desenvolupament per l'API REST de la plataforma Learnum

En la figura 9.4 podem observar diferents dades importants pel funcionament de l'aplicació. Entre aquestes podem destacar:

- **DATABASE:** El nom de la base de dades de l'aplicació.
- **DB_PASSWORD:** Contrasenya utilitzada per la connexió amb la base de dades del servidor local de l'aplicació.
- **JWT_SECRET:** Clau emprada per la codificació de la informació mitjançant *JSON Web Tokens*.

Connexió amb la base de dades

Un cop que ja disposem de les credencials necessàries per a la connexió amb la base de dades de l'aplicació, es pot fer el següent pas. En el cas d'aquest projecte, per tal d'establir la connexió es va utilitzar l'ORM Sequelize esmentat prèviament. Aquest, amb les credencials establertes al fitxer de variables d'entorn, ens permet establir la connexió amb la base de dades de forma molt compacte i simple.

```

1 const sequelize = new Sequelize(config.database, config.user, config.
    password, {
2   host: config.host,
3   dialect: config.db_dialect,
4   logging: logger.logging
5 })

```

Amb aquestes poques línies de codi es crea una instància d'un objecte *Sequelize* amb les credencials establertes al fitxer `.env` que ens estableix la connexió amb la base de dades. Per tal de sincronitzar les taules de la base de dades amb els models definits per Sequelize podem fer-ho de la següent manera:


```

1 sequelize.sync({ force: false })
2   .then(() => {
3     console.log('Sequelize: Synchronized tables')
4   })
5   .catch((error) => {
6     console.error('Error: Unable to synchronize Database tables with
7     Sequelize')
8     console.error('[ERROR]', error)
9   })

```

Els models Sequelize definits per les dades o taules de la base de dades són una part important del desenvolupament de l'API, i s'expliquen més endavant en aquest capítol.

Preparació endpoints

Com ja sabem, les API REST satisfan sol·licituds fetes per clients. Aquestes peticions arriben en format d'URI que defineixen una ruta única per cada recurs disponible. Afortunadament amb *NodeJS* disposem del paquet *Express.js*. Tal com es va explicar anteriorment al capítol 7, secció 7.2.3, *Express.js* ens dota d'eines d'encaminament per la implementació d'aquestes URIs amb els *routers*.

Per tant, per tal de definir les rutes per cada grup de peticions establides al capítol 8, secció 8.3.3, es va escriure el següent codi amb el Framework d'Express:

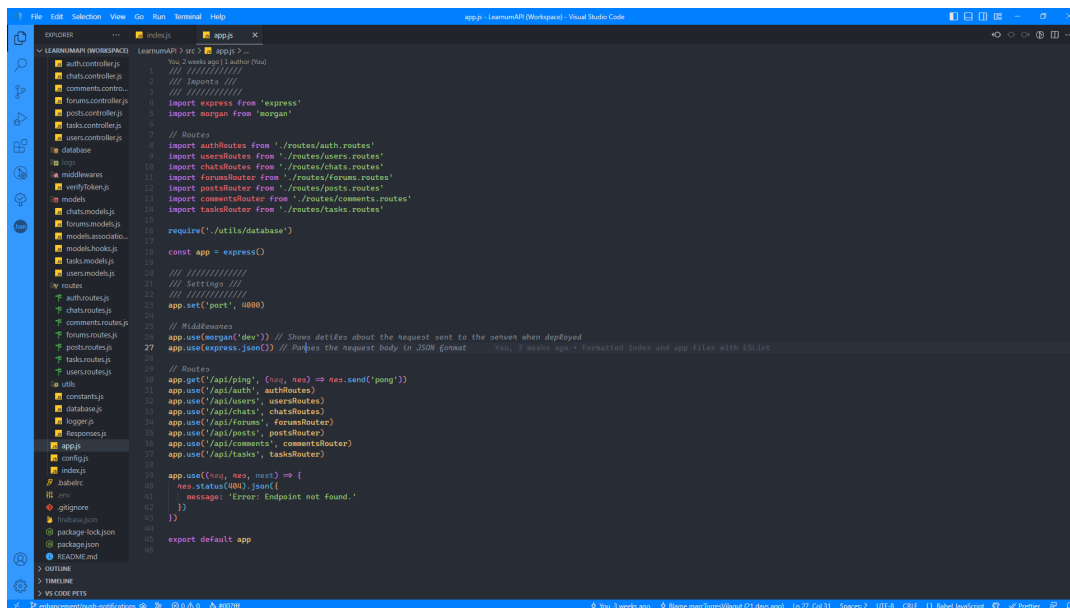


FIGURA 9.5: Captura de pantalla dels continguts del fitxer *app.js* del projecte d'API REST per la plataforma Learnum

En la figura 9.5 podem observar l'ús del mètode *use* per a la instància de l'aplicació *express*. Aquest mètode ens permet definir instàncies d'encaminadors (*routers*) per utilitzar per a un conjunt de rutes. Per exemple, les rutes que comencen amb el prefix `"/api/tasks"`, utilitzen l'encaminador *tasksRouter*, importat del fitxer *routes/task.routes.js*, el codi del qual és el següent:

```

1 import { Router } from 'express'
2 import verifyToken from '../middlewares/verifyToken'

```

```

3 import { TasksController } from '../controllers/tasks.controller'
4 const router = Router()
5
6 router.get('/', verifyToken, TasksController.getTasks)
7 router.post('/', verifyToken, TasksController.addTask)
8 router.get('/:id', verifyToken, TasksController.getTaskByID)
9 router.patch('/:id', verifyToken, TasksController.updateTask)
10 router.delete('/:id', verifyToken, TasksController.deleteTask)
11
12 export default router

```

En el codi podem observar com cada ruta disposa d'una combinació de camí i verb *HTTP* pròpia, i disposen del seu propi comportament per satisfer la petició feta pel client.

Inicialització del procés del servidor

Només queda un últim pas per poder començar a utilitzar l'API REST, i aquest és inicialitzar el procés del servidor. Per tal de fer-ho es va escriure, en el fitxer *index.js* el següent fragment de codi que inicialitza el procés d'escolta de l'aplicació express en el port definit per aquesta:

```

1 import app from './app'
2
3 const main = () => {
4   app.listen(app.get('port'))
5   console.log('Server on port ${app.get('port')}')
6 }
7
8 main()

```

Per tal d'executar el procés des de la línia de comandes es va aprofitar els *scripts* de NodeJS per tal de definir els següents tipus d'execucions al fitxer *package.json*:

```

1 "scripts": {
2   "babel-node": "babel-node --presets=@babel/preset-env",
3   "dev": "nodemon --exec npm run babel-node src/index.js",
4   "start": "npm run babel-node src/index.js",
5   "populate": "npm run babel-node src/database/populateDB.js"
6 },

```

- **babel-node**, interpreta i compila el codi executat amb els *presets* i *plugins* de Babel.
- **dev**, inicialitza el servei de l'API REST utilitzant el paquet *nodemon* per tal que el procés es reinicialitzi cada vegada que es fa un canvi en el projecte.
- **start**, inicialitza el servei de l'API REST sense *nodemon*.
- **populate**, executa un script que emplena la base de dades amb dades prefabricades per disposar informació inicial amb la qual testear l'aplicació. Aquest script fa servir eines de Sequelize.

Per tant, per tal d'inicialitzar el procés només cal executar aquests *scripts* amb *npm* i ja disposarem de totes les funcionalitats presentades per l'API REST.



```
C:\WINDOWS\system32\cmd.exe
Marc@DESKTOP-DFHE19A MINGW64 ~/repos/uni/TFG/LearnumAPI (dev)
$ npm run dev

> learnumapi@1.0.0 dev
> nodemon --exec npm run babel-node src/index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `npm run babel-node src/index.js`

> learnumapi@1.0.0 babel-node
> babel-node --presets=@babel/preset-env src/index.js

Server on port 4000
Sequelize: Synchronized tables
```

FIGURA 9.6: Captura de pantalla de la línia d'ordres amb l'execució del procés de l'API REST de la plataforma Learnum amb el *script* "dev"

Cal tenir en consideració que per tal que l'aplicació es pugui connectar correctament amb la base de dades local, *XAMPP* ha d'estar en obert amb el servei de la base de dades en execució.

9.2.2 Implementació dels models de dades

Sequelize disposa de diverses eines que ens fan la vida més fàcil a l'hora de desenvolupar el disseny d'una base de dades, així com les relacions entre les entitats que formen part d'aquesta. En aquest apartat es donarà un cop d'ull a la implementació d'alguns dels models de dades utilitzats en l'aplicació de l'API REST per establir connexió amb les taules de la base de dades, i altres eines utilitzades per les relacions d'aquestes.

Models de dades

Una de les classes principals de Sequelize és la classe *Model*. Aquesta ens permet la definició de models de dades, donant estructura als atributs que formen les columnes d'aquests models en la base de dades, així com el nom de les taules i altra informació extra autogenerada per Sequelize. A continuació veurem alguns dels models definits per les dades de l'API.

```
1  /**
2   * @param {Sequelize} sequelize
3   * @param {Sequelize.DataTypes} types
4   *
5   * @return {Sequelize.Model} The users DB model
6   */
7  const UserModel = (sequelize, types) => {
8    const User = sequelize.define('user', {
9      id: {
10       type: types.INTEGER,
11       primaryKey: true,
12       autoIncrement: true,
13       allowNull: false
14     },
15     username: {
16       type: types.STRING,
17       allowNull: false
18     },
19     name: types.STRING,
20     surnames: types.STRING,
21     email: {
22       type: types.STRING,
23       allowNull: false,
24       unique: true
25     },
26     password: {
27       type: types.STRING,
28       allowNull: false
29     },
30     phone: types.STRING,
31     birthdate: types.INTEGER,
32     avatar_icon: types.STRING,
33     n_following: {
34       type: types.INTEGER,
35       defaultValue: 0
36     },
37     n_followers: {
38       type: types.INTEGER,
39       defaultValue: 0
40     }
41   }, {
42     updatedAt: false,
43     freezeTableName: false,
44     tableName: 'tblusers'
45   })
46 }
```

FIGURA 9.7: Captura de pantalla del codi de definició del model de dades d'*Usuaris* de l'API

En la figura 9.7 podem veure la definició del model d'usuari. Aquest, en sincronitzar Sequelize, ens genera una taula a la base de dades associada a aquest model amb les columnes establertes com atributs. Podem veure que la definició disposa de tres paràmetres: el nom del model, els atributs del model, i configuració extra del model.

En el segon bloc podem veure que cada atribut té un conjunt de propietats que permet una configuració de la columna per la taula. Per exemple, la columna *email* haurà de contenir valors de tipus cadena de text, sense valors nuls, i únics, o sigui que no poden haver-hi dues entrades per la taula amb el mateix correu.

Per altra banda, al tercer bloc de configuració extra podem visualitzar les següents propietats:

- **updateAt:** Sequelize afegeix per defecte columnes de *timestamps* que permeten mantenir un control de quan una entrada a la taula s'ha creat (*createdAt*) o actualitzat (*updatedAt*). Aquesta propietat, assignada amb valor fals, ens permet evitar que afegeixi la columna *updatedAt* a la taula.
- **freezeTableName:** Per defecte Sequelize dona com a nom a les taules el nom assignat al model al primer paràmetre pluralitzat. Per poder donar un nom personalitzat a les taules, aquesta propietat desactiva aquesta assignació automàtica permetent assignar un nom personalitzat a la taula del model amb la següent propietat.
- **tableName:** Defineix el nom de la taula del model a la base de dades.

```

28  /**
29   * @param {Sequelize} sequelize
30   * @param {Sequelize.DataTypes} types
31   *
32   * @return {Sequelize.Model} The chat's members DB model
33   */
34  const ChatMemberModel = (sequelize, types) => {
35    return sequelize.define('chat_member', {
36      user_id: {
37        type: types.INTEGER,
38        primaryKey: true,
39        allowNull: false
40      },
41      chat_id: {
42        type: types.INTEGER,
43        primaryKey: true,
44        allowNull: false
45      },
46      is_admin: types.BOOLEAN
47    }, {
48      updatedAt: false,
49      freezeTableName: false,
50      tableName: 'tblchats_members'
51    })
52  }
53

```

FIGURA 9.8: Captura de pantalla del codi de definició del model de dades dels *Membres d'un Xat* de l'API

Un altre model interessant és el presentat a la figura 9.8, on podem visualitzar la definició d'un model intermediari per la relació entre el model *user* i el model *chat*, per definir la relació dels membres d'un xat. Aquest tipus de models han estat definits per implementar, entre altres, les relacions *n:m* entre: usuaris i xats, usuaris i fòrums, usuaris i altres usuaris (seguidors).

Per poder visualitzar la resta de models definits per l'API, veure l'annex A, secció A.4.

Associacions

Per tal d'afegir l'adjectiu "relacional" a la base de dades, Sequelize disposa d'eines conegudes com a *associations*. Aquestes ens permeten definir relacions entre els models definits. Per poder explicar aquesta funcionalitat, continuarem a partir de l'exemple de model presentat a la figura 9.8.

```

50
51 ✓   /// CHATS - USERS (Members)
52     // Chat's Members: A Chat can have many Users Members
53 ✓   Chat.belongsToMany(User, {
54     through: ChatMember,
55     foreignkey: 'chat_id',
56     otherKey: 'user_id',
57     as: 'chatMembers'
58   })
59     // User Chats: A User can be a Member of many Chats
60 ✓   User.belongsToMany(Chat, {
61     through: ChatMember,
62     foreignkey: 'user_id',
63     otherKey: 'chat_id',
64     as: 'userChats'
65   })
66

```

FIGURA 9.9: Captura de pantalla de la implementació de l'associació entre el model d'usuari (**User**) i el model de xat (**Chat**) per afegir la relació *membre d'un xat*

A la figura 9.9 podem observar la creació de l'associació entre dos models a través d'un tercer model. La definició de l'associació es divideix en 3 passos:

1. Definir la primera direcció de la relació: En aquest cas es defineix inicialment que el model **Chat** pot contenir diverses instàncies del model *User*.
2. Definir la direcció inversa de la relació: Després es defineix que el model **User** pot pertànyer a diversos models *Chats*.
3. Definim la taula intermediària: En ambdues associacions s'ha afegit com a paràmetre la propietat *through* que ens permet definir a través de quin model s'ha de fer l'associació. En altres paraules, a través de quina taula s'ha d'associar aquests dos models. En aquest cas, aquest model intermediari és el model **ChatMember**.

Les associacions són eines molt potents. Aquestes no només ens afegixen les relacions entre les taules de la base de dades, sinó que també afegixen mètodes "màgics", que permeten interactuar des d'un model amb els models associats a aquest, per exemple recuperant amb un usuari tots els seus seguidors, o tots els xats d'un usuari. Els noms d'aquests "mètodes màgics" es generen a partir de la propietat *as* de la definició de la relació. Per exemple per la figura 9.9: el model **User**, que té la propietat *as* assignada com "userChats", rebrà el mètode *getUserChats* entre d'altres.

Hooks

L'última eina de Sequelize de la que parlarem en aquest apartat són els *hooks*. Sense confondre'ls amb els hooks de react, aquestes eines ens permeten la definició dels disparadors de les taules de la base de dades. A forma de resum, un disparador és una funció o *query* que s'executarà cada cop que un esdeveniment succeeixi a la taula

per la qual estigui definit aquest. Aquesta eina és de molta ajuda per definir diferents comportaments automatitzats a la base de dades. En aquest projecte s’han utilitzat principalment per mantenir un control dels comptadors de diferents models.

Alguns models, com ja hem dit, disposen de comptadors. Alguns exemples d’aquest són les publicacions (model **ForumPost**), que disposa d’un comptador de *likes* que ha rebut la publicació. Una opció pot ser actualitzar aquesta columna manualment cada cop que s’afegeixi una nova reacció de “m’agrada” per una publicació, però per tal d’aprofitar al màxim Sequelize s’ha optat per definir disparadors que ens actualitzin automàticament, en aquest cas, cada cop que s’afegeixi o s’elimini alguna reacció.

```

52
53   PostLike.afterCreate('increment_post_likes_counter', async (postLike, options) => {
54     const post = await Post.findByPk(postLike.post_id)
55     await post.increment('n_likes')
56   })
57   PostLike.beforeDestroy('decrement_post_likes_counter', async (postLike, options) => {
58     const post = await Post.findByPk(postLike.post_id)
59     await post.decrement('n_likes')
60   })

```

FIGURA 9.10: Captura de pantalla de la implementació d’un *hook* pel model **PostLike** per actualitzar el comptador de *likes* d’una publicació

A la figura 9.10 podem observar com es defineix un *hook* per un model de Sequelize. En aquest cas, s’està afegint dos disparadors: un per a quan es creï una instància d’una reacció de “m’agrada” per una publicació (*INSERT*), i un altre per a quan s’elimini una reacció per una publicació (*DELETE*). A més, a l’haver definit l’atribut *n_likes* del model **ForumPost** com un enter, disposem dels mètodes `increment` i `decrement` que ens permeten fer l’operació d’incrementar i reduir el valor del comptador sense haver de definir la *query* per actualitzar l’entrada de la taula.

9.2.3 Proves

Un cop ja s’ha configurat i implementat les funcionalitats que es trobaven en desenvolupament, toca la fase de proves. En aquesta fase es va utilitzar tant POSTMAN com l’aplicació de la vista per tal de *debuggar* l’API. POSTMAN, com ja s’ha mencionat prèviament, és un client REST que ens permet fer peticions directes a una API sense necessitat de disposar de l’aplicació de la vista en el moment de fer les proves. Per tal de provar les funcionalitats s’han definit un conjunt de peticions. A continuació, es mostraran algunes imatges de les peticions utilitzades:

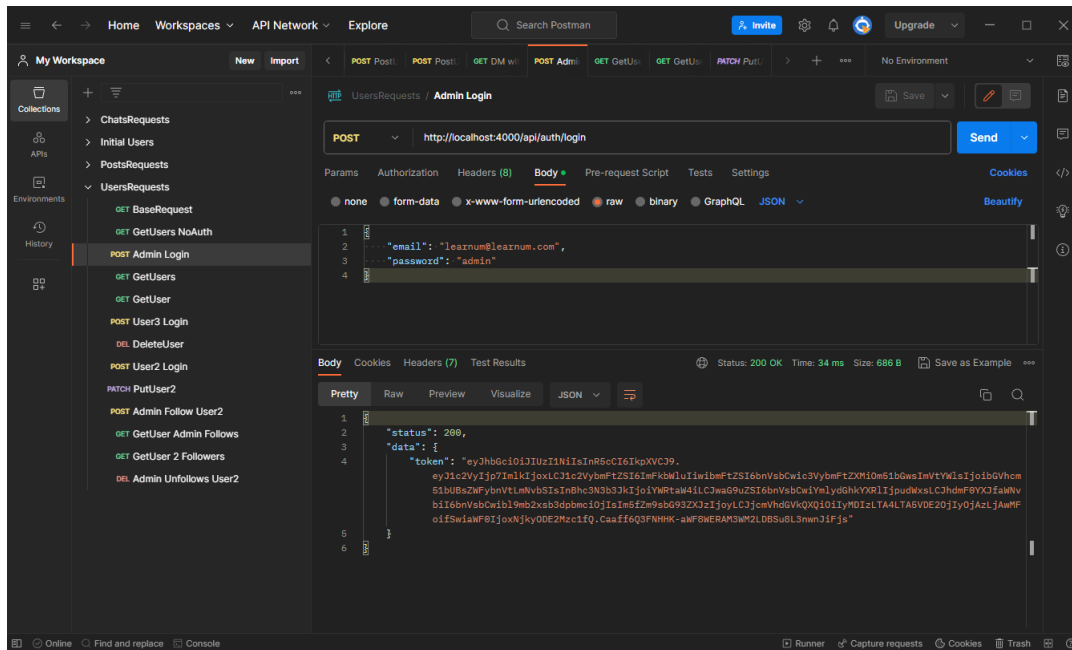


FIGURA 9.11: Captura de pantalla de POSTMAN executant la petició a l'API per l'inici de sessió de l'usuari administrador

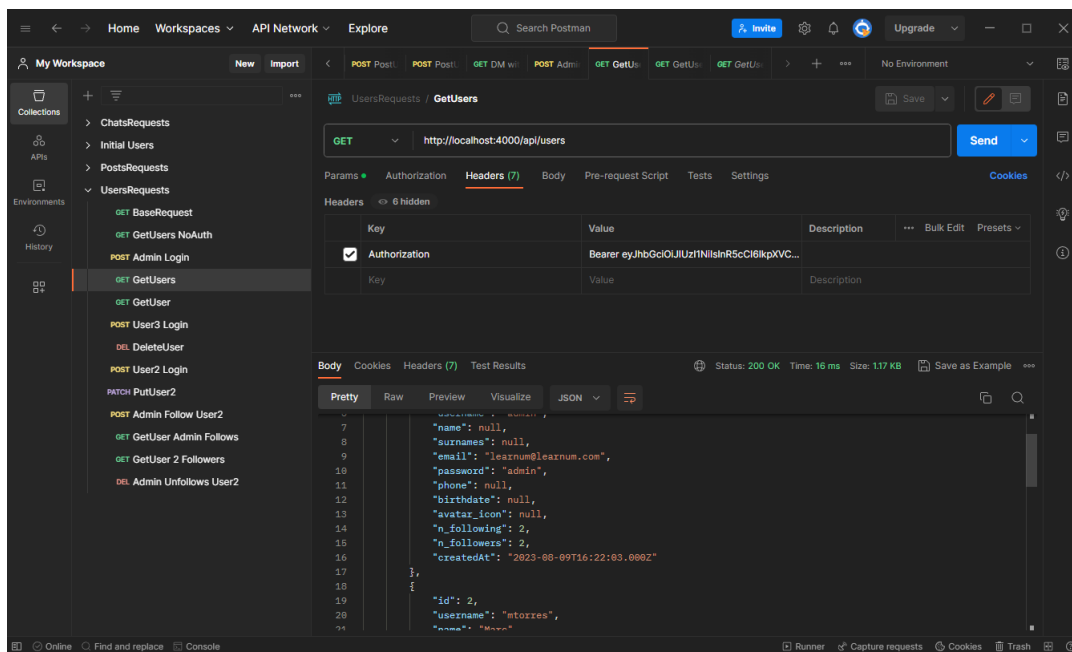


FIGURA 9.12: Captura de pantalla de POSTMAN executant la petició a l'API per recuperar els usuaris del servidor

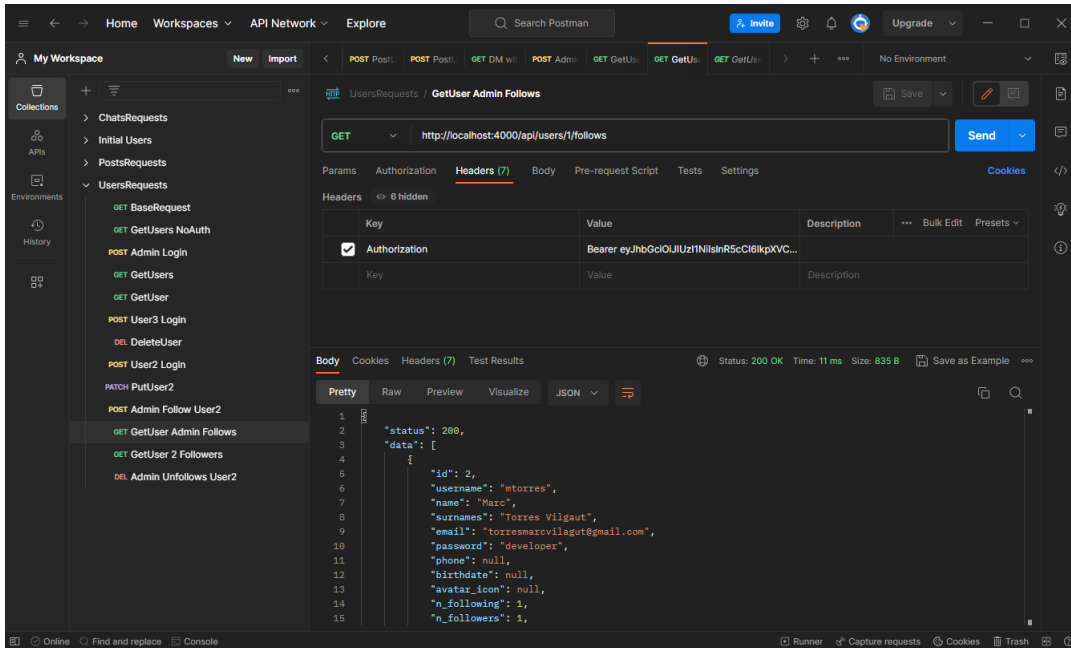


FIGURA 9.13: Captura de pantalla de POSTMAN executant la petició a l'API per recuperar els usuaris als quals l'usuari administrador segueix

9.2.4 Problemes i solucions

Al llarg del desenvolupament de l'API han aparegut alguns problemes que han requerit un estudi extra a l'hora de solucionar-los. En aquest apartat es presentaran aquests problemes i la solució que s'ha implementat per aquests.

Autenticació d'usuaris

Quan ens trobem amb la funcionalitat d'inici de sessió per una aplicació hem de tenir en consideració que l'usuari no ha d'haver d'iniciar sessió cada cop que entri a l'aplicació. Per fer això possible disposem de diverses opcions.

La primera opció que pot sorgir com a idea per emmagatzemar les dades de l'inici de sessió pot ser l'ús de *cookies* associat a una entrada a la base de dades per una taula específica per mantenir les sessions dels usuaris. Tot i que aquesta és una bona opció, per la implementació de l'autenticació per aquest projecte s'ha optat per la utilització dels **JSON Web Token (JWT)**.

JWT és un estàndard que permet la codificació d'informació per garantir seguretat a l'hora de transmetre aquesta. L'ús d'aquesta ens permet generar *tokens*, utilitzant una clau privada coneguda com a *JWT Secret* que és pròpia de l'aplicació i es troba configurada al fitxer `.env` com s'ha presentat anteriorment.

Per tal de guardar la informació de la sessió de l'usuari, es genera un *token* de JWT a partir d'informació de l'usuari que s'enviarà a la vista on s'emmagatzemarà per poder enviar-ho amb cada petició a l'API per poder autenticar de quin usuari prové la petició. També s'ha afegit un *middleware* per les rutes d'Express que verificarà la identitat de l'usuari que ha fet la petició. Els *middlewares* són fragments de codi que s'executen sobre les peticions abans d'executar el procés de servei principal

per la ruta. Per exemple en la ruta `router.get('/', verifyToken, PostsController.getPosts)`, el mètode `verifyToken` serveix com a *middleware* abans d'executar-se la funció principal `PostController.getPosts` per recuperar la informació de les publicacions del sistema.

Obtenció d'usuaris que no siguin membres d'un xat

Si bé obtenir els usuaris que forment part d'un xat pot ser una tasca simple amb la utilització de Sequelize i les associacions establertes anteriorment, l'invers pot suposar un problema. En aquest cas ens trobem amb la funcionalitat d'afegir nous membres a un xat de grup. Per tal de poder afegir a algun usuari a un xat de grup ha d'existir una relació de seguiment de l'usuari que vol afegir amb l'usuari que serà afegit (usuari A ha de seguir a l'usuari B per poder afegir-ho a un grup). Per poder abreviar a l'hora de nomenar a aquest grup d'usuaris que poden ser afegits a un grup se'ls hi ha decidit anomenar com "*outsiders*" (forasters en anglès).

Per tal de poder recuperar a aquests *outsiders* es va presentar la següent implementació:

```

1 const getChatOutsiders = async (request, response) => {
2   let { limit, page } = request.query
3   page = (page) ? parseInt(page) : 1
4   limit = (limit) ? parseInt(limit) : 10
5   const offset = limit * (page - 1)
6   try {
7     Jwt.verify(request.token, config.jwt_secret, async (err, authData) =>
8     {
9       if (err) return Responses.Error(response, 401, 'Error: Unauthorized
10      request from unknown user.')
11       const authUser = authData.user
12
13       const { id } = request.params || null
14       if (!id) return Responses.Error(response, 400, 'Error: The request
15      must contain the ID of the chat to search.', 'id')
16       if (isNaN(id)) return Responses.Error(response, 400, 'Error: The ID
17      of the user requested must be a number.', 'id')
18
19       const chat = await ChatModel.findById(id)
20       if (!chat) return Responses.Error(response, 404, 'Error: Chat ${id}
21      not found.', 'id')
22
23       const user = await UserModel.findById(authUser.id)
24       const followedUsers = await user.getFollows({ limit, offset })
25
26       const outsiders = await Promise.all(
27         followedUsers.map(async (user) => {
28           const { user_followers: followers, ...outsiderData } = user.
29           dataValues
30
31           if (!(await chat.hasChatMember(user))) return outsiderData
32           else return null
33         })
34       )
35       const resOutsiders = outsiders.filter(outsider => outsider)
36
37       return Responses.Success(response, resOutsiders)
38     })
39   } catch (error) {
40     return Responses.Error(response, 500, error.message)
41   }
42 }
```

36 }

Tot i que he d'admetre que no és la solució més elegant, i probablement tampoc la més eficient, ha estat el *workaround* que vaig poder implementar per aquest problema. El procés de la solució es basa en 4 passos:

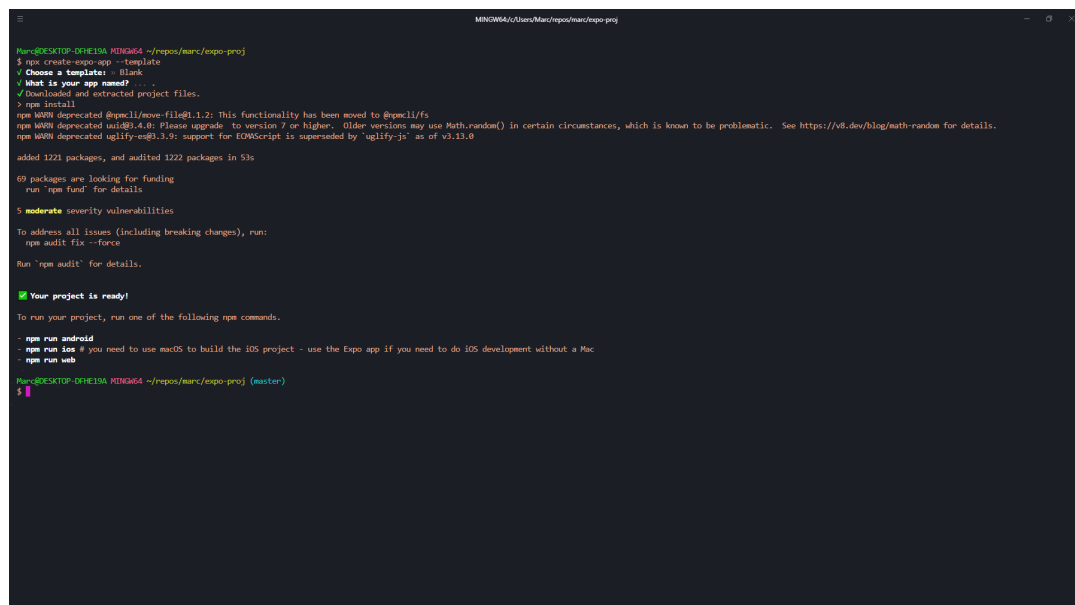
1. Recuperar la instància del model del xat pel qual s'estan recuperant els *outsiders*.
2. Recuperar la instància del model de l'usuari (Usuari A) que ha fet la petició.
3. Recuperar els usuaris que l'usuari A segueix.
4. *Mapejar* la informació dels usuaris seguits per tal de recuperar només aquells que no siguin membres del xat.

9.3 Aplicació per dispositius mòbils

9.3.1 Configuració

Expo

Una aplicació per mòbils es pot desenvolupada perfectament només amb React Native i els paquets dels quals disposa amb *npm*, però la utilització del SDK d'Expo ha presentat un gran suport a l'hora de desenvolupar i també a l'hora de provar les funcionalitats gràcies a l'eina d'Expo Go. Per tal d'utilitzar Expo per inicialitzar el projecte disposem de la comanda *create-expo-app* que proporciona l'opció d'inicialitzar el projecte buit o amb algunes estructures de fitxer ja definides.



```

MINGW64~/Users/Marc/repos/marc/expo-proj
Marc@DESKTOP-DHE13A MINGW64 ~/repos/marc/expo-proj
$ npm create-expo-app --template
✔ Choose a template? blank
✔ What is your app named? ...
✔ Downloaded and extracted project files.
$ npm install
npm WARN deprecated @pnpm/move-file@1.1.2: This functionality has been moved to @pnpm/fs
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uglifyjs@3.19: support for ECMAScript is superseded by 'uglify-js' as of v3.13.0
added 1221 packages, and audited 1222 packages in 53s
69 packages are looking for funding
  run npm fund for details
5 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.

✔ Your project is ready!

To run your project, run one of the following npm commands:

- npm run android
- npm run ios if you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web

Marc@DESKTOP-DHE13A MINGW64 ~/repos/marc/expo-proj (master)
$

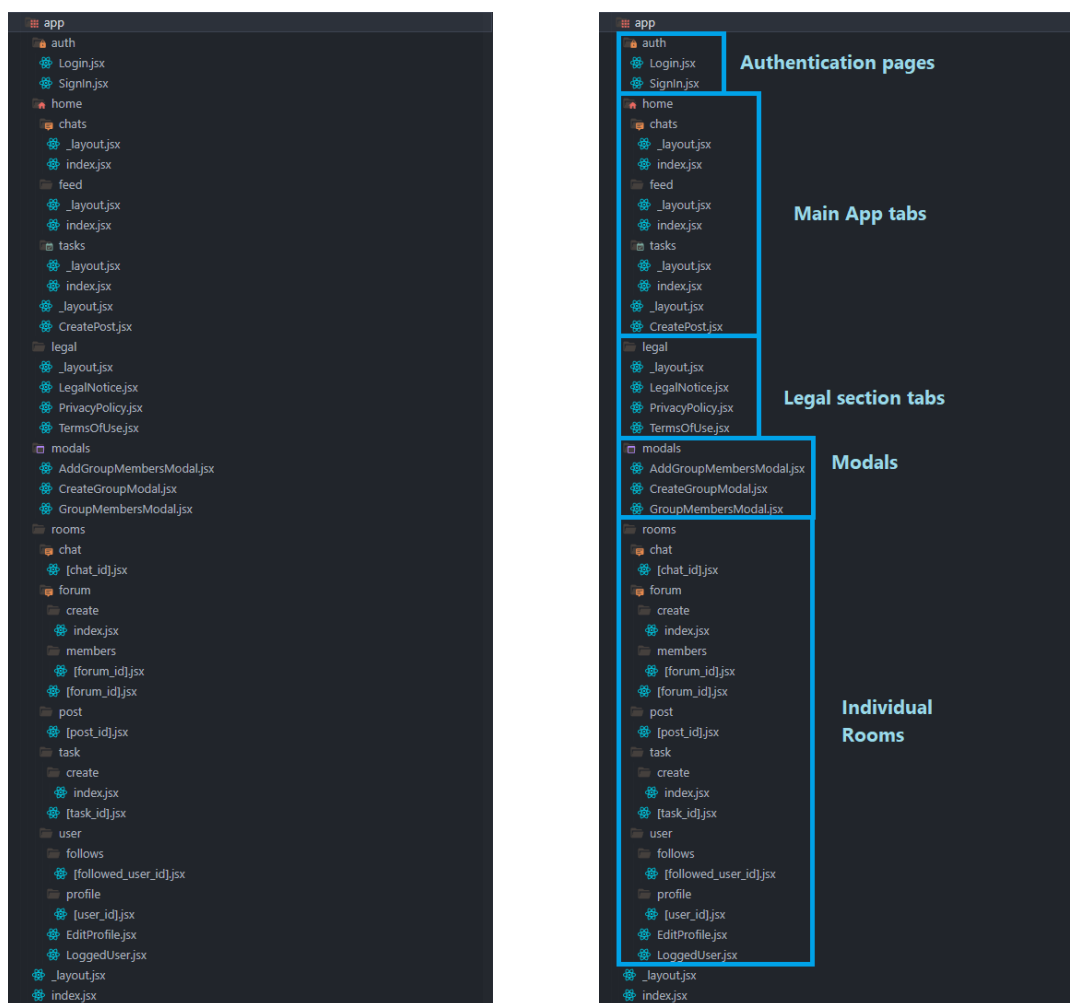
```

FIGURA 9.14: Captura de pantalla de l'execució de la comanda *npm create-expo-app* per la creació d'un projecte d'Expo amb l'opció de plantilla buida per començar des de zero.

Expo Router

Per la navegació de l'aplicació disposem de diverses opcions. D'entre les més conegudes tenim *React Navigation* i *React Router*. Per altra banda, Expo disposa d'un paquet conegut com a *Expo Router*, construït per sobre de *React Navigation*, que facilita la implementació de la navegació per mitjà de sistemes de fitxers, així com un *custom hook* i components que ens permeten crear rutines amb comportaments de redirecció de l'encaminament.

Aleshores, i amb l'ajuda d'Expo Router, la vista d'aquest projecte s'estructura en el sistema de fitxers que es pot observar a la figura 9.15.



(A) Estructura de la vista utilitzant Expo Router

(B) Estructura de la vista amb anotacions per indicar els continguts

FIGURA 9.15: Captures de pantalla del directori de les pantalles de l'aplicació de vista per mòbils de Learnum

A la subfigura 9.15b podem observar els subdirectoris del directori principal de la vista *app* amb anotacions que indiquen el contingut d'aquest. Ens trobem amb els següents subdirectoris:

- **auth.** Les pantalles per l'autenticació dels usuaris amb inici de sessió o registre de nou compte d'usuari es troben en aquest subdirectori.

- **home.** Subdirectori principal que conté les pantalles de la navegació per *tabs* o pestanyes.
- **legal.** Subdirectori amb les pantalles per la navegació per *tabs* o pestanyes de la secció d'avisos legals de l'aplicació.
- **modals.** Aquest subdirectori conté les pantalles de modals de l'aplicació.
- **rooms.** El subdirectori més extens. Aquest conté les diferents *rooms* o sales per visualitzar la informació de diferents elements de l'aplicació. Un exemple és una sala de xat, la pantalla de la qual es troba al fitxer `rooms/chat/[chat_id].jsx` on `[chat_id]` representa un valor dinàmic. Si redirigim l'aplicació a la ruta `rooms/chat/4.jsx`, podríem visualitzar la sala del xat amb identificador 4.

9.3.2 Proves

Per tal de provar les funcionalitats de l'aplicació simplement s'ha utilitzat aquesta a mesura que es desenvolupava i s'afegien noves funcionalitats. Aquestes proves de les funcionalitats per aquesta part del projecte han estat efectuades en diversos dispositius:

- Dispositiu físic amb Android 13 utilitzant Expo Go.
- Emulador Pixel 3a amb Android 14.
- Emulador iPhone 14 amb iOS 16.

Expo Go és una aplicació disponible a la *PlayStore* de *Google* o la *AppStore* d'*Apple*. Aquesta aplicació permet l'execució d'aplicacions desenvolupades amb l'SDK d'Expo, sigui usant Expo Go en si o amb una instal·lació d'una *build* desenvolupada pel sistema del dispositiu on s'estigui executant.

Per tal d'executar l'aplicació de la vista de Learnum, només s'ha de disposar de l'aplicació d'Expo Go instal·lada a un dispositiu mòbil i escanejar des d'aquesta aplicació el codi QR que apareix quan s'executa el servei per l'aplicació d'Expo. Un cop escanejat, començarà a compilar l'aplicació i executarà una instància al dispositiu mòbil.

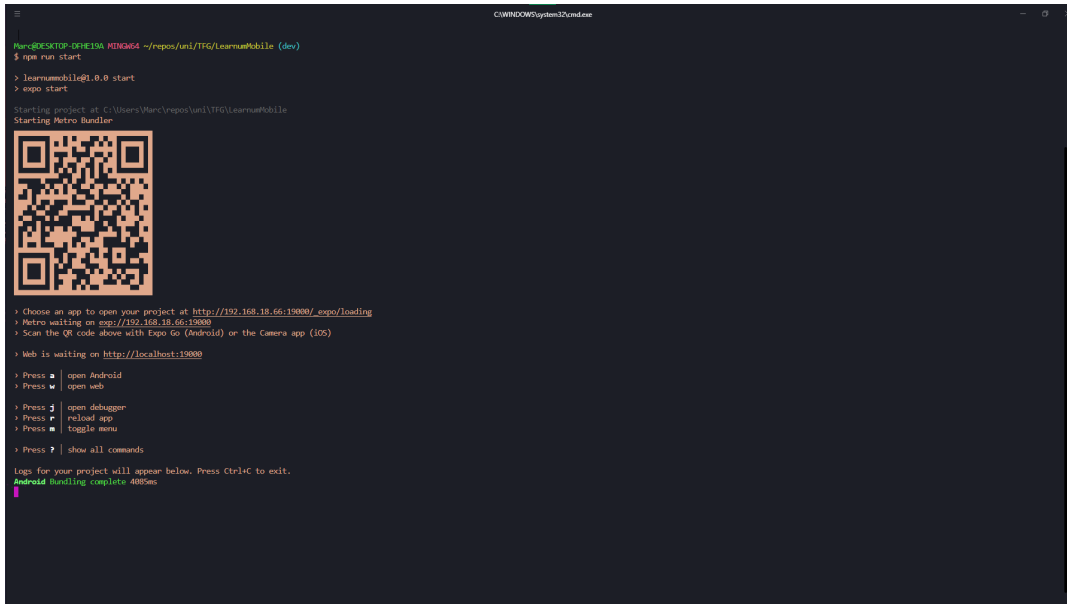


FIGURA 9.16: Captura de pantalla de l'execució del procés de l'aplicació Expo per la vista per dispositius mòbils de Learnum

9.3.3 Problemes i solucions

Crides a l'API

Efectuar crides a una API pot ser una tasca que, si bé no és complexa, pot ser tediosa. Per tal d'agilitzar el procés de petició d'informació a l'API s'ha fet ús d'**Axios**, un paquet de npm que permet la configuració d'un gestor de peticions que facilita la sintaxi a l'hora d'escriure aquestes peticions.

Axios ens permet configurar el gestor donant una configuració inicial a aquest per mitjà de propietats. En aquest podem definir un URL base a la que les peticions seran enviades, aquest URL hauria de ser la ruta a l'API de Learnum per tal de poder fer les peticions a aquesta. També podem assignar capçaleres i els tipus de respostes admeses.

Mantenir sessió iniciada

Igual que quan parlàvem de la implementació de l'API i els problemes relacionats amb l'autenticació dels usuaris, en aquesta part del projecte també van sorgir problemes relacionats.

El primer problema que va sorgir era l'emmagatzematge del *token* d'autenticació. Si bé quan efectuem la petició d'inici de sessió, aquesta ens retorna correctament aquest token, s'ha de tenir compte a l'hora de guardar-ho, ja que aquest, tot i estar codificat, pot continuar contenint informació delicada per l'usuari.

Per tant, per tal de guardar el token de forma segura, es va fer ús del paquet *expo-secure-store*, el qual utilitza de fons tecnologies d'emmagatzematge segur de les plataformes natives en les quals s'executi el codi.

Un cop disposem del token guardat en un emmagatzematge segur, cal afegir la lògica per mantenir la sessió iniciada cada cop que s'engegui l'aplicació.

```

1  import axios from 'axios'
2  import Constants from '../utils/Constants'
3
4  const ApiManager = axios.create({
5    baseURL: Constants.API_PATH,
6    responseType: 'json',
7    withCredentials: true,
8    headers: {
9      Accept: 'application/json',
10     'Content-Type': 'application/json'
11   }
12 })
13

```

FIGURA 9.17: Captura de pantalla de la inicialització de l'*ApiManager*, una instància del gestor de peticions d'*axios* per efectuar crides a l'API de Learnum

Per tal de fer aquest control es va afegir un nou mètode pel *custom hook* per l'autenticació d'usuaris que comprova l'existència del token de sessió d'usuari a l'emmagatzematge segur. Si el token existeix, disposem de la informació necessària per poder accedir a l'API amb les credencials de l'usuari amb la sessió iniciada.

```

14
15  const checkLogin = async () => {
16    const token = await secureStorage.get(Constants.CLIENT_TOKEN_KEY)
17    if (token === null) return false
18
19    // Set to the headers of the API Manager's requests
20    ApiManager.defaults.headers.common.Authorization = `Bearer ${token}`
21
22    return true
23  }
24

```

FIGURA 9.18: Captura de pantalla del mètode per comprovar si un usuari disposa d'una sessió iniciada al dispositiu amb un token JWT.

L'execució d'aquest mètode de comprovació de la sessió es durà a terme quan s'obri l'aplicació com un comportament asíncron de pantalla de càrrega.

Només queda un petit detall. A l'API, tret de les peticions de registre i inici de sessió, la resta de peticions requereixen l'existència del token *Authorization* a les capçaleres de la petició. Per tal d'afegir aquest token a cada petició podríem recuperar el token cada cop que es vulgui afegir, però ja que disposem de la instància d'*Axios* creada prèviament, podem assignar el token com a part de les capçaleres de les peticions efectuades per l'*ApiManager* de la següent manera (presentada també a la figura 9.18):

```

1 // Set to the headers of the API Manager's requests
2 ApiManager.defaults.headers.common.Authorization = `Bearer ${token}`

```


Paginació de peticions

Carregar molta informació de cop pot suposar una càrrega innecessària tant per la vista a l'hora de renderitzar-ho, com pel servidor API a l'hora d'enviar tota la informació al client. Aquest cas ens el trobem diversos cops en aquesta aplicació i, per tant, s'ha utilitzat la paginació en diverses parts d'aquesta.

Per posar-nos en context, un exemple de cas on s'ha vist necessària la utilització de la paginació ha estat la recuperació de publicacions per la pàgina d'inici de l'aplicació.

La pàgina d'inici de l'aplicació consta de 3 funcions principals:

- Servir de pàgina principal per l'aplicació.
- Mostrar un llistat de publicacions a manera de *feed*.
- Proporcionar l'accés al perfil personal de l'usuari.

Quan volem recuperar la informació de les publicacions pel *feed* de la pantalla d'inici, no ens interessa recuperar totes les publicacions del sistema, ja que suposaria una càrrega de feina tant per l'API com per la vista molt alta a la llarga, a més que encara que recuperem totes les publicacions a la vista només es podran visualitzar unes quantes.

Per tant, per tal de solucionar aquest problema es va afegir la paginació amb la implementació del *custom hook*: **usePagination**. Aquest defineix els comportaments necessaris per afegir la funcionalitat de paginació als components *FlatList* de React Native. Per exemple, cada cop que un usuari arribi al final d'una llista, aquesta efectuarà una nova petició a l'API per recuperar una nova pàgina. Aquest comportament continuarà executant-se en el mateix ordre fins a arribar a l'última pàgina. També, si l'usuari prefereix refrescar la llista, pot fer-ho gràcies a aquest *custom hook*.

```

1  import { useState } from 'react'
2
3  const usePagination = ({ nextPageCallback = null, resetPageCallback = null }) => {
4    const [page, setPage] = useState(1)
5    const [lastPage, setLastPage] = useState(false)
6
7    const nextPage = () => {
8      if (!lastPage) {
9        setPage(page + 1)
10       if (nextPageCallback) nextPageCallback()
11     }
12   }
13
14   const lastPageReached = () => {
15     setLastPage(true)
16   }
17
18   const resetPage = () => {
19     setPage(1)
20     setLastPage(false)
21     if (resetPageCallback) resetPageCallback()
22   }
23
24   return [
25     page,
26     {
27       nextPage,
28       lastPageReached,
29       resetPage
30     }
31   ]
32 }
33

```

FIGURA 9.19: Captura de pantalla del *custom hook* usePagination per afegir el comportament de paginació a les llistes de React Native

Splash Screen

Quan una aplicació s'obre pot haver de carregar informació necessària per al correcte funcionament d'aquesta. La pàgina d'inici de les aplicacions tendeixen a utilitzar aquesta informació, per tant, abans de mostrar aquesta pàgina, es mostra una *Splash Screen*. Aquesta pantalla ens serveix per avisar als usuaris que l'aplicació s'està carregant.

Per tal d'implantar la *Splash Screen* es va utilitzar les eines que proporciona el paquet de *Expo Router*. Amb aquest podem recuperar el component **SplashScreen** que ens permet la configuració d'aquesta pantalla. El component disposa de dos mètodes que s'ha utilitzat per a la implementació:

- `preventAutoHideAsync`. Aquest mètode evita que la *Splash Screen* s'amagui un cop la navegació del projecte s'hagi acabat de muntar.
- `hideAsync`. En cridar-se aquest mètode, amaga la *Splash Screen* si aquesta no s'havia amagat prèviament.

Per tal d'efectuar la comprovació correcta de si un usuari disposa d'una sessió iniciada, es va afegir un *hook* `useEffect` de React per tal de fer la càrrega asíncrona de la informació necessària per a l'aplicació. El codi de la implementació és el següent:

```
1 SplashScreen.preventAutoHideAsync() // Prevent SplashScreen from hiding
2
3 export default function App () {
4   const router = useRouter()
5   const { checkLogin } = useAuth()
6   useEffect(() => {
7     (async () => {
8       try {
9         const isLoggedIn = await checkLogin()
10
11         await new Promise(() => setTimeout(() => {
12           // When all loading is setup, unmount
13           // the splash screen component.
14           SplashScreen.hideAsync() // Hide SplashScreen
15           router.replace(isLoggedIn ? '/home' : '/auth/Login') // Redirect
16           user to correct screen
17           }, 1000))
18         } catch (err) {
19           console.warn(err)
20         }
21       })()
22     }, [checkLogin, router])
23
24     return (
25       <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center'
26         }}>
27         <ActivityIndicator size='large' />
28       </View>
29     )
30   })
31 }
```

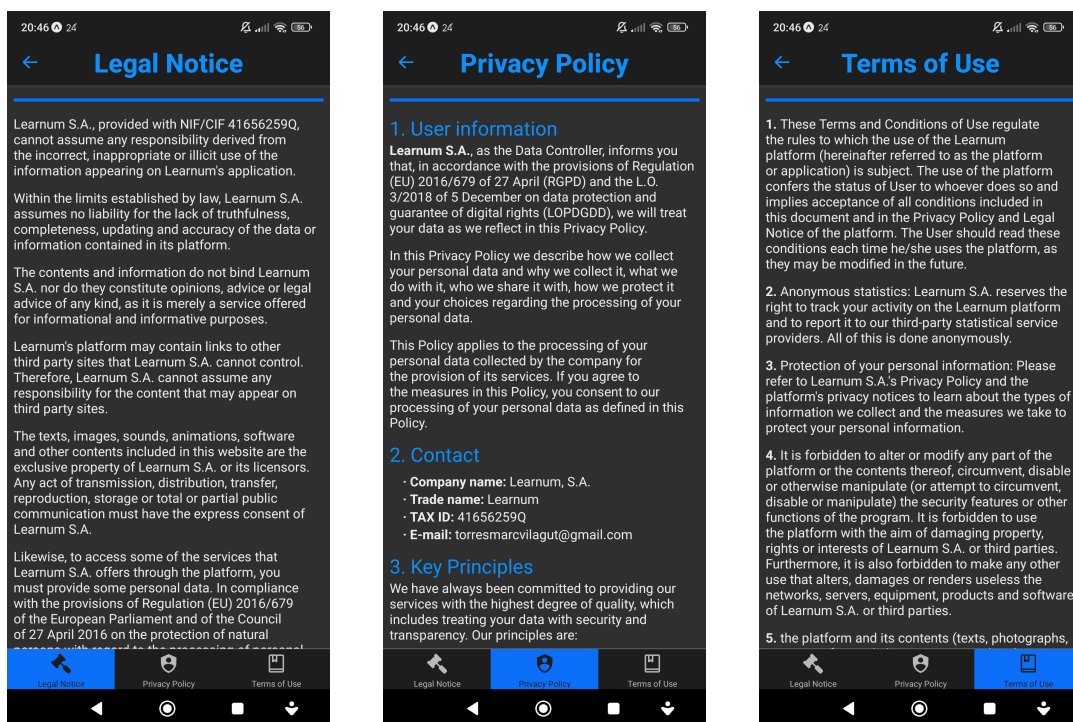

Capítol 10

Implantació i resultats

En aquest capítol visualitzarem els resultats obtinguts de les funcionalitats implementades en aquest projecte i com han estat implantades a la vista amb la qual interactuaran els usuaris de la plataforma.

10.1 Apartat legal de l'aplicació

La legislació actual ens obliga a prendre mesures de seguretat per la privacitat de les dades personals dels usuaris de la plataforma, així com indicar l'ús d'aquestes dades i demanar el consentiment dels usuaris. En compliment d'aquestes obligacions, s'han pres mesures de seguretat i s'ha afegit una secció a l'aplicació on els usuaris podran informar-se dels apartats legals de la plataforma. Aquests apartats també es troben disponibles a l'annex A, secció A.5, d'aquest document.



(A) Avís legal de l'aplicació Learnum

(B) Política de privacitat de Learnum

(C) Termes o condicions d'ús de la plataforma

FIGURA 10.1: Captures de pantalla de les seccions amb els documents relatius a la legislació per la plataforma Learnum

10.2 *Splash screen*

Primer de tot disposem de la pàgina de càrrega inicial, la *Splash Screen*. Com s'ha esmentat prèviament, aquesta pàgina fa la comprovació de l'existència d'alguna sessió iniciada per algun usuari al dispositiu.



FIGURA 10.2: Captura de pantalla de la *Splash Screen*

10.3 Accés a la plataforma

Quan un usuari vol accedir a la plataforma disposa de dues opcions: - si disposa d'un compte d'usuari, en la pantalla d'inici de sessió poden entrar a l'aplicació amb les seves credencials - per altra banda, si aquest és un nou usuari sense compte, des de la pantalla d'inici de sessió, pot accedir a la pantalla de registre i crear un nou compte amb les credencials requerides i altres d'opcionals.

Quan un usuari vol accedir a l'aplicació per dispositius mòbils, aquest pot fer-ho accedint a la pàgina d'inici de sessió i entrant les seves credencials. Aquesta pantalla serà la primera pantalla a carregar quan l'aplicació acabi de carregar-se a la *Splash Screen* si no s'ha trobat una sessió iniciada.

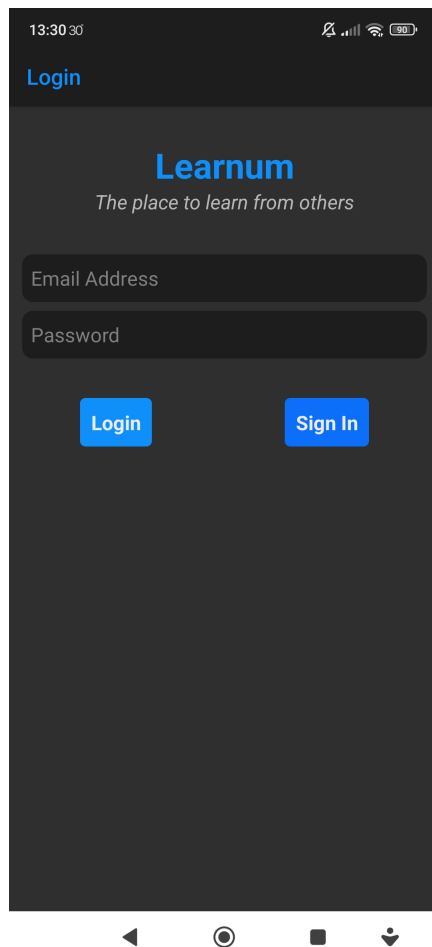


FIGURA 10.3: Captura de pantalla de la pantalla d'inici de sessió de l'aplicació Learnum per mòbils

Si, per altra banda, l'usuari que vol accedir no disposa encara d'un compte, aquest pot accedir a la pantalla de registre prement el botó "Sign In" que podem veure a la figura 10.3.

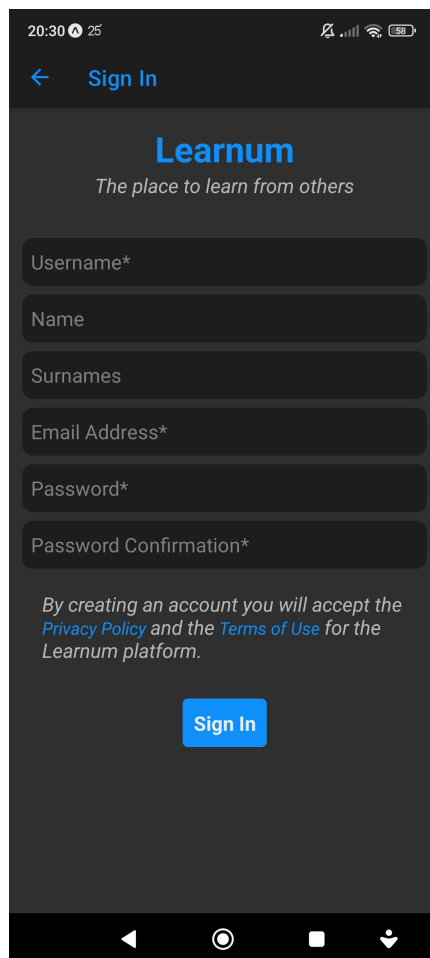
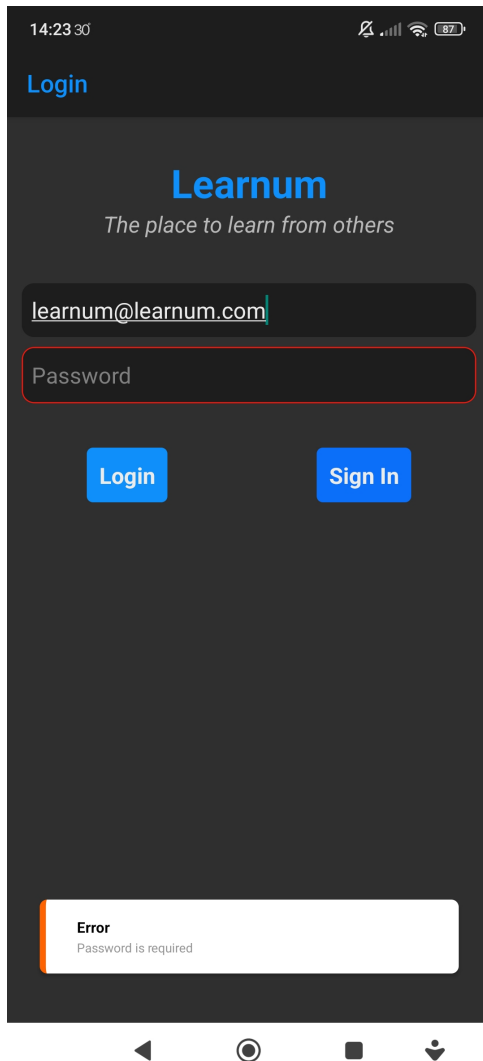
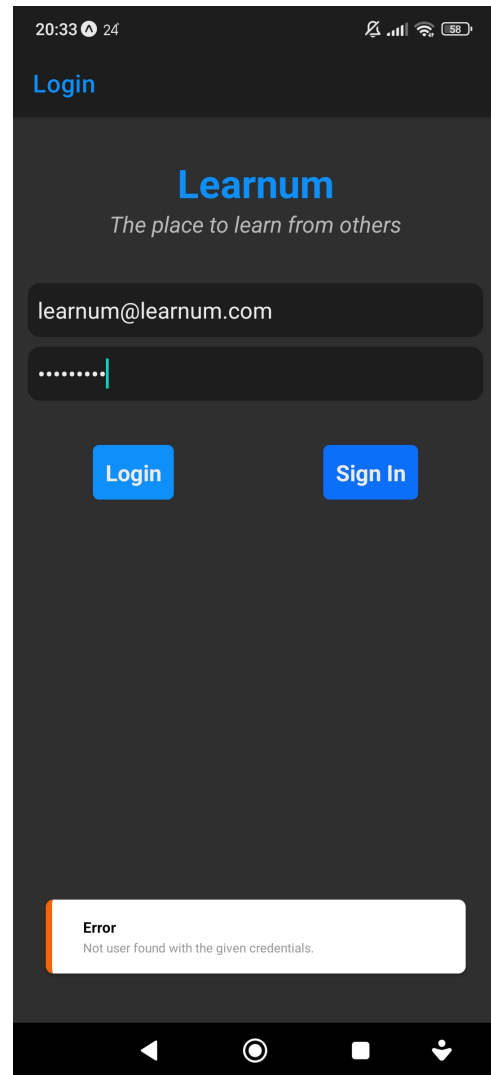
Aquesta és una captura de pantalla d'un dispositiu mòbil que mostra la pantalla de registre d'un nou compte d'usuari a l'aplicació Learnum. La pantalla té un fons fosc i text blanc i blau. A la part superior esquerra, hi ha l'hora 20:30 i la data 25. A la part superior dreta, hi ha els icones de senyal, Wi-Fi i bateria. A sota, hi ha un botó de retrocedir i el text "Sign In" en blau. El títol principal és "Learnum" en blau, amb el subtítol "The place to learn from others" en blanc. Hi ha sis camps de text: "Username*", "Name", "Surnames", "Email Address*", "Password*" i "Password Confirmation*". A sota dels camps, hi ha un text que diu: "By creating an account you will accept the [Privacy Policy](#) and the [Terms of Use](#) for the Learnum platform." A la part inferior, hi ha un botó "Sign In" en blau. A la part inferior de la pantalla, hi ha els icones de navegació d'Android.

FIGURA 10.4: Captura de pantalla de la pantalla de registre d'un nou compte d'usuari

En cas d'entrar alguna dada incorrecta o invàlida als formularis, aquest mostraran l'error ocorregut a l'usuari per tal que pugui rectificar-ho.



(A) No s'ha entrat el camp de contrasenya que és requerit



(B) No s'ha trobat un compte amb les credencials entrades

FIGURA 10.5: Captura de pantalla del control d'error al formulari de la pantalla d'inici de sessió.

20:34 24

← Sign In

Learnum
The place to learn from others

u1959683

Name

Surnames

u1959683@campus.udg.edu

.....

Password Confirmation*

By creating an account you will accept the [Privacy Policy](#) and the [Terms of Use](#) for the Learnum platform.

Sign In

Error
The password must be confirmed

(A) Confirmació de contrasenya és requerida

20:34 24

← Sign In

Learnum
The place to learn from others

u1959683

Name

Surnames

u1959683@campus.udg.edu

.....

.....

By creating an account you will accept the [Privacy Policy](#) and the [Terms of Use](#) for the Learnum platform.

Sign In

Error
Passwords must match

(B) Les contrasenyes entrades no coincideixen

FIGURA 10.6: Captura de pantalla del control d'error al formulari de la pantalla de registre d'un nou compte d'usuari

10.4 Pantalla d'inici de l'aplicació

Un cop l'usuari accedeix a la plataforma, aquest és redirigit a la pantalla principal de l'aplicació, també coneguda com a *feed*, on disposarem de diverses funcionalitats entre les quals trobem la navegació per mitjà del menú de navegació per pestanyes a la part inferior de la pantalla, la navegació cap al perfil personal de l'usuari que ha iniciat sessió per mitjà d'un botó de la capçalera que redirigeix a l'usuari al seu perfil (👤), i la navegació a la secció on es troben els documents d'informació legal relativa a l'aplicació amb el botó de la capçalera d'informació (ℹ️).

La funcionalitat més important a mencionar en aquesta pantalla és la visualització del llistat de les últimes publicacions fetes a l'aplicació. La càrrega del llistat de publicacions està fet a partir de la solució presentada al capítol 9, secció 9.3.3, on s'explicava el problema de la paginació.

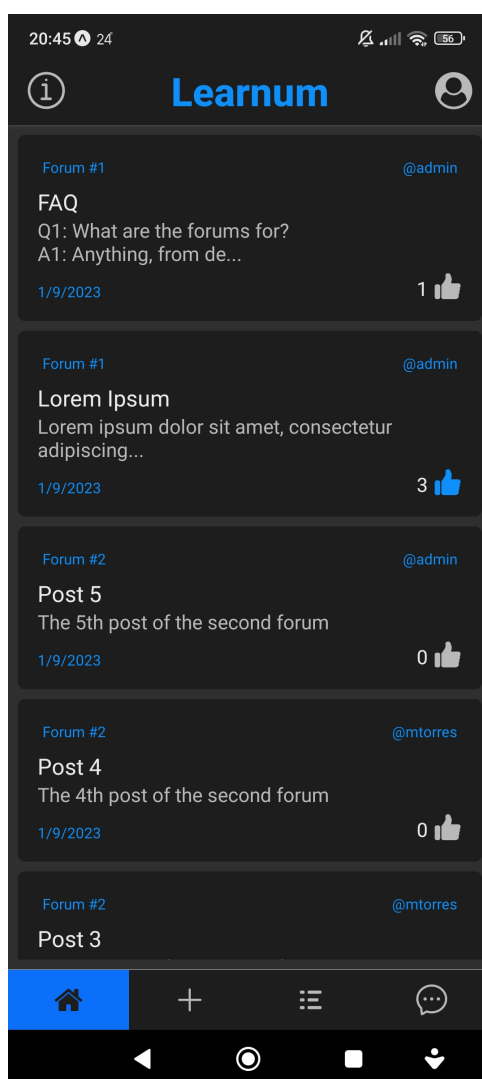
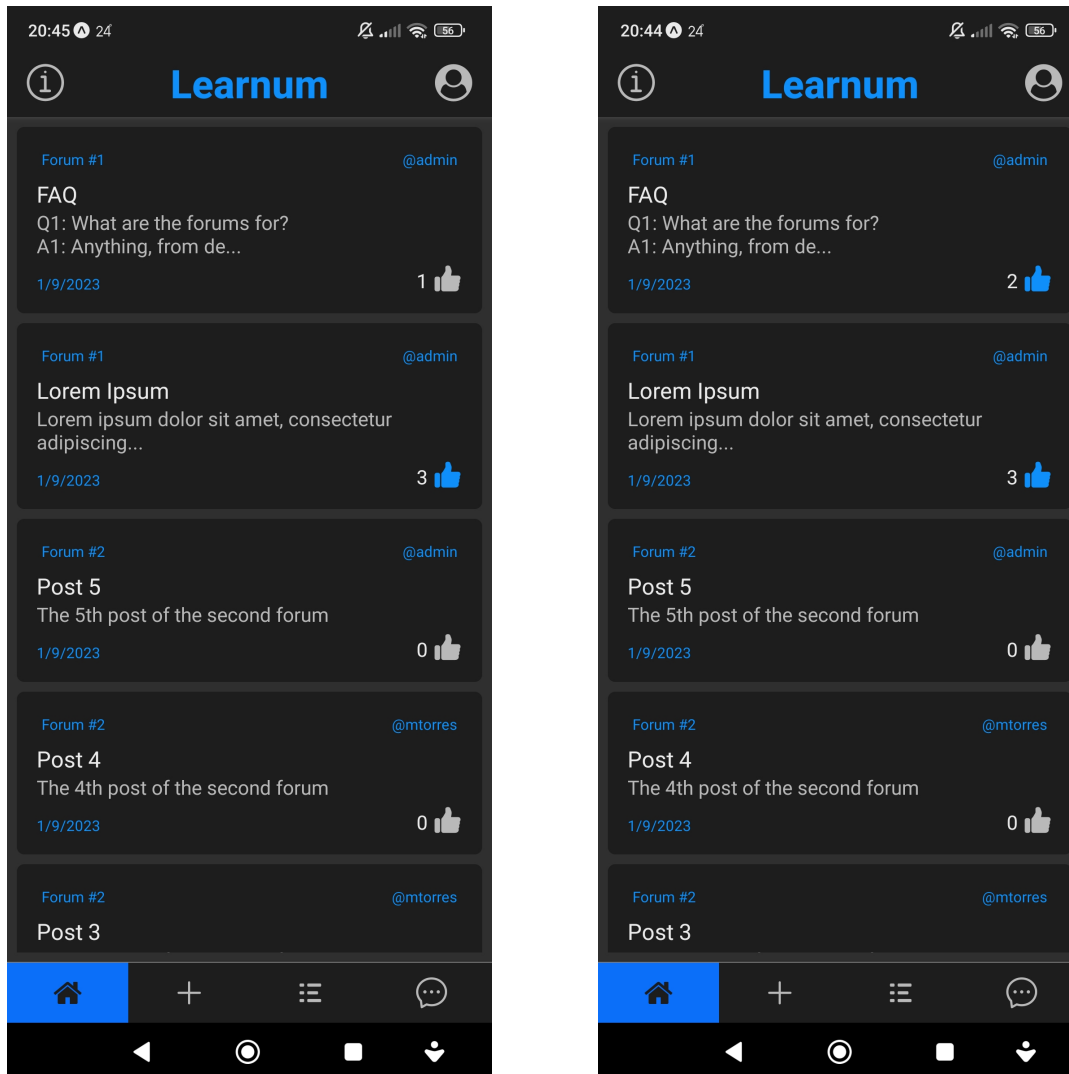


FIGURA 10.7: Captura de pantalla de la pantalla principal de l'aplicació Learnum per mòbils (*Feed*)

El llistat de publicacions està format per les targetes de les publicacions. Aquestes targetes proporcionen la navegació a les pàgines específiques de les respectives publicacions, així com la funcionalitat per afegir o eliminar una reacció de “m’agrada” a aquestes.



(A) Vista prèvia a afegir la reacció de “m’agrada”

(B) Vista posterior a l'addició de la reacció a la publicació

FIGURA 10.8: Captura de pantalla de la pantalla principal mostren el comportament d'afegir reaccions de “m’agrada” a una publicació

També cal indicar que les targetes de les publicacions disposen d'una capçalera, com podem veure a la figura 10.7, en la que es presenten els noms de: el fòrum on s'ha publicat (esquerra) i el nom de l'usuari que va crear la publicació (dreta). Aquests dos camps són enllaços que permeten a l'usuari navegar a la pàgina del fòrum de la publicació i al perfil de l'usuari que ha fet la publicació respectivament.

10.5 Perfil d'usuari

El perfil d'un usuari disposa de dos tipus de vistes: la vista pel perfil personal de l'usuari que ha iniciat sessió, i la vista pels perfils dels altres usuaris. Si bé aquestes dues vistes presenten diferències, també disposen d'utilitats compartides entre les dues, com pot ser la visualització d'informació de l'usuari com el seu nom d'usuari (*username*), correu electrònic, data de naixement, etc. Els perfils també disposen d'un llistat de les publicacions que l'usuari ha fet i per la qual s'apliquen les mateixes funcionalitats que la llista presentada a la pantalla d'inici.

10.5.1 Perfil personal

Com ja s'ha comentat, la vista del perfil pot ser de dos tipus, en aquesta secció donarem un cop d'ull al perfil personal de l'usuari connectat. En aquest trobem l'accés a la pantalla d'edició de la informació personal de l'usuari.

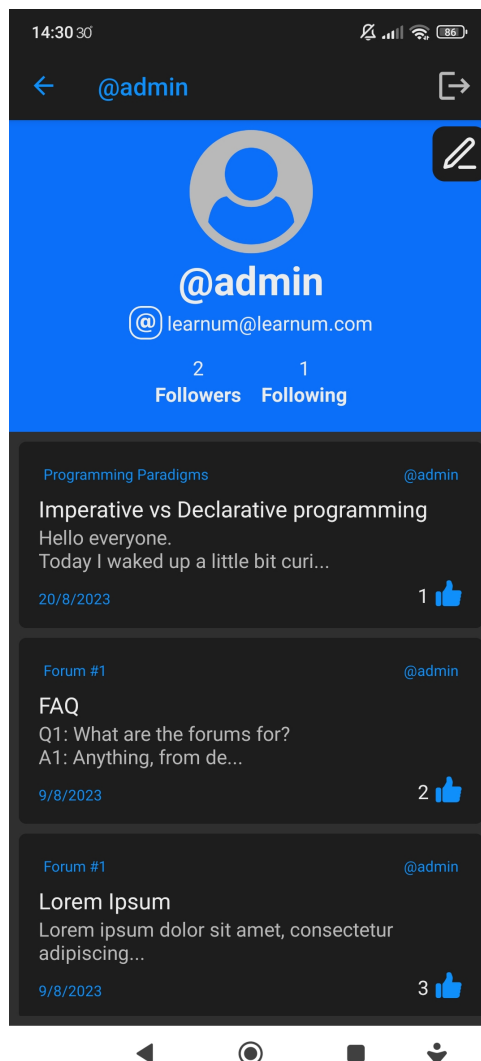


FIGURA 10.9: Captura de pantalla del perfil personal d'usuari

A la pàgina d'edició de les dades del perfil personal de l'usuari ens trobem amb dos formularis dividits per una franja. El formulari superior permet la modificació de les dades de l'usuari visibles a altres usuaris com poden ser el nom i cognoms, o el número de telèfon. Per altra banda, al formulari inferior ens trobem amb l'opció per canviar la contrasenya de l'usuari. Per tal de poder aplicar el canvi per la nova clau del compte de l'usuari, aquest ha d'entrar, la nova contrasenya, la contrasenya actual, i, per seguretat, la confirmació de la contrasenya actual.

La imatge mostra una captura de pantalla d'un dispositiu mòbil amb un fons fosc. A la part superior, hi ha l'estatus de l'hora (14:33:30), el nivell de bateria (86%) i els símbols de senyal i Wi-Fi. Just a sota, hi ha un botó de retrocedir i el títol 'Edit admin profile'. El formulari 'Edit info' té un ícon de desatjar a la dreta i conté els camps: 'admin' (pre-omplert), 'Name', 'Surnames', 'learnum@learnum.com', 'Phone' (amb un interruptor desactivat) i 'Birthdate' (amb un ícon de tancar). Una línia horitzontal blava separa aquest formulari del 'Change Password', que té els camps: 'Password*', 'Password Confirmation*' i 'New Password*', i un botó 'Change' a la part inferior.

FIGURA 10.10: Captura de pantalla de la pàgina d'edició de les dades de l'usuari

Aquests formularis també disposen de controls d'errors.

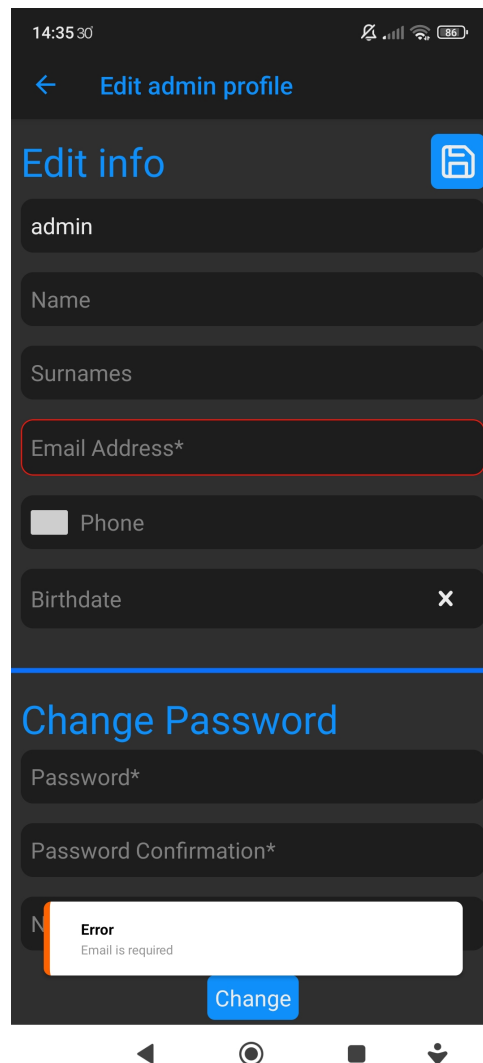


FIGURA 10.11: Captura de pantalla del control d'errors als formularis d'edició de les dades del perfil de l'usuari

10.5.2 Perfil d'un altre usuari

Tal com hem comentat, aquesta vista també disposa de la visualització d'informació de l'usuari, així com les publicacions que ha fet. A més, en aquesta pàgina disposem d'algunes funcionalitats més.

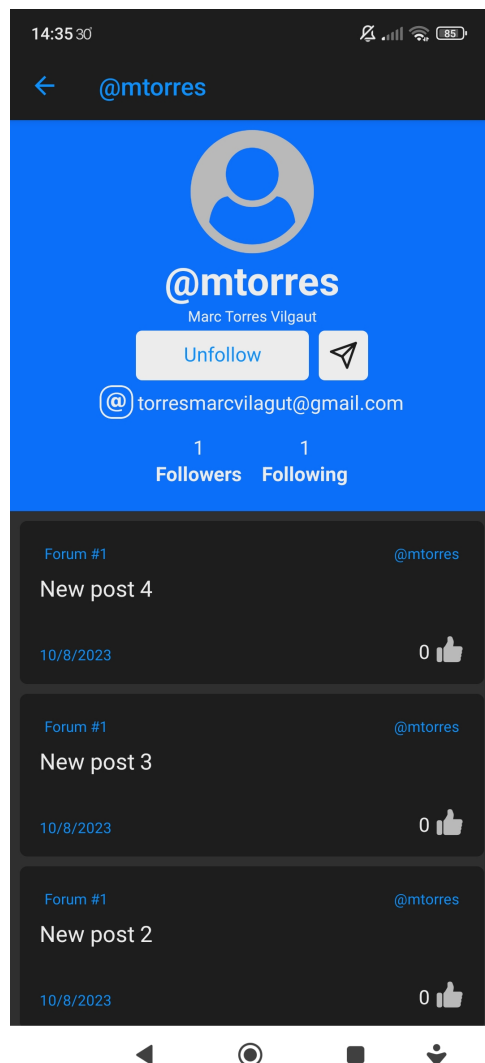
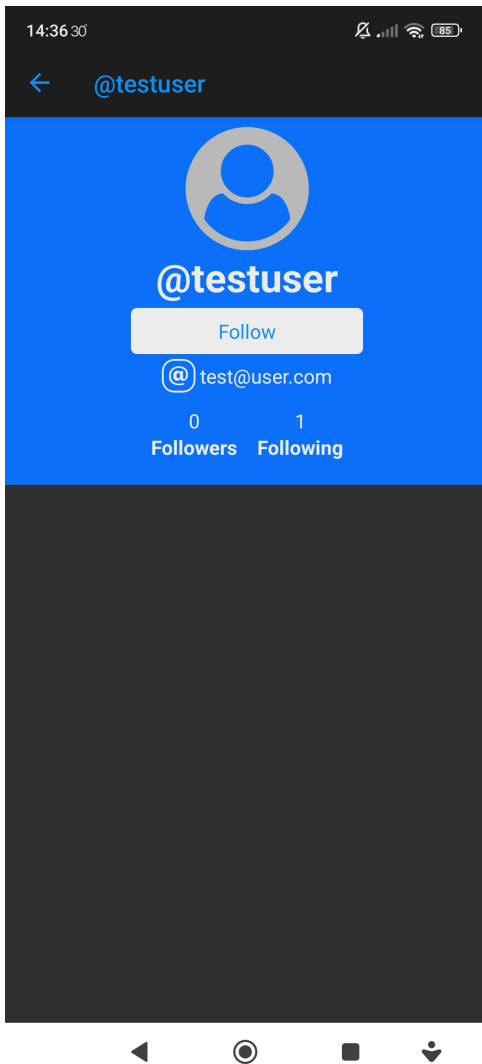
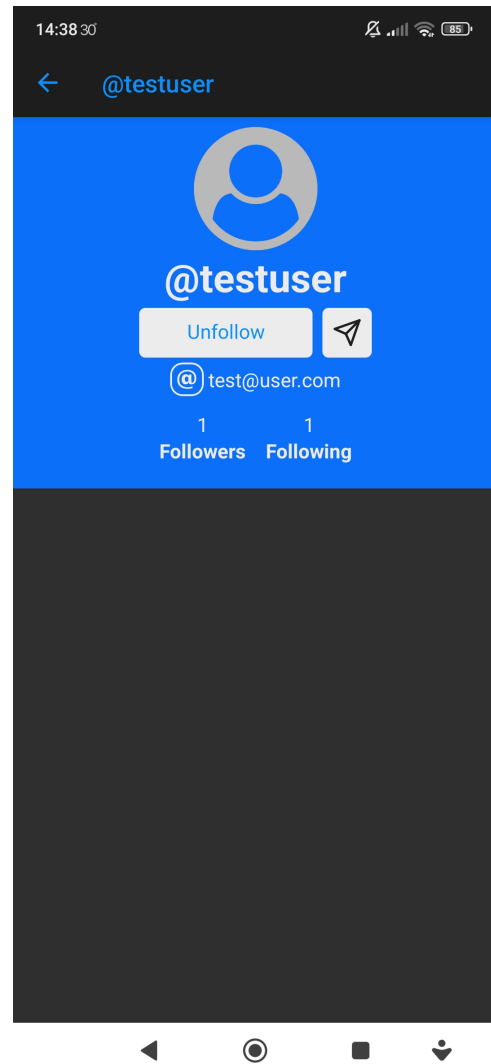


FIGURA 10.12: Captura de pantalla del perfil d'un usuari

La primera funcionalitat a mencionar és la de seguir o deixar de seguir a l'usuari a qui pertany el perfil que s'està visualitzant. Per tal de fer-ho, sota el seu nom d'usuari, i nom complet en cas de tenir-ho assignat, disposem del botó que ens proporciona aquesta funcionalitat. Un cop premem el botó de "Follow" es començarà a seguir a aquest usuari, i el botó s'actualitzarà per mostrar l'opció de deixar de seguir a l'usuari, així com mostrar l'opció d'obrir un canal de missatgeria directa amb aquest usuari.



(A) Vista del perfil quan no se segueix a l'usuari

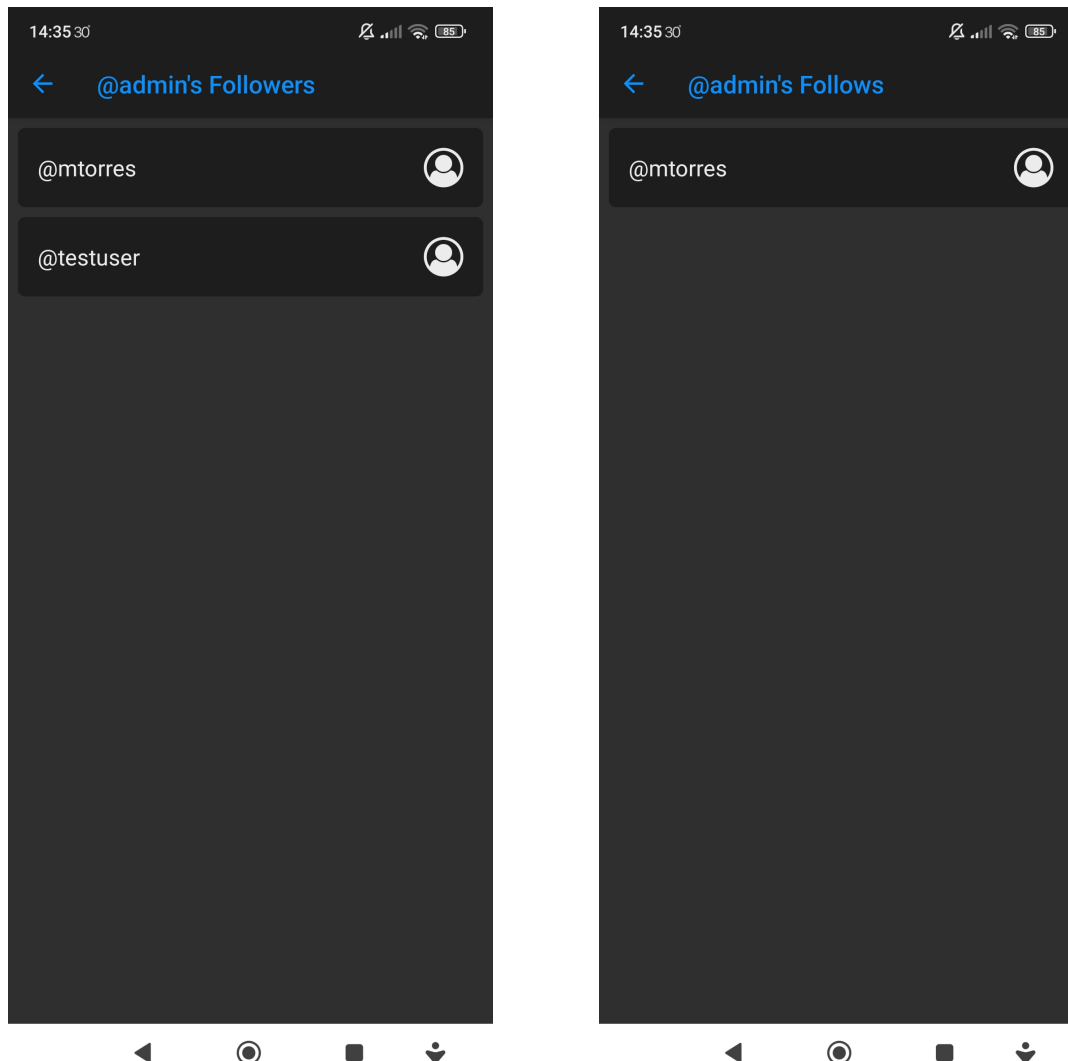


(B) Vista del perfil un cop es comença a seguir a l'usuari

FIGURA 10.13: Captura de pantalla del perfil d'un usuari

10.5.3 Visualitzar seguidor i usuaris seguits

Una altra funcionalitat compartida als dos tipus de vistes del perfil d'usuari és la de poder veure el llistat d'usuaris que segueix l'usuari i el llistat d'usuaris que l'usuari segueix. Per tal d'accedir-hi es pot fer prement els comptadors d'usuaris seguidors (*Followers*) i d'usuaris seguits (*Following*) que redirigiran a l'usuari a la pàgina amb el llistat de seguidors i d'usuaris seguits respectivament.



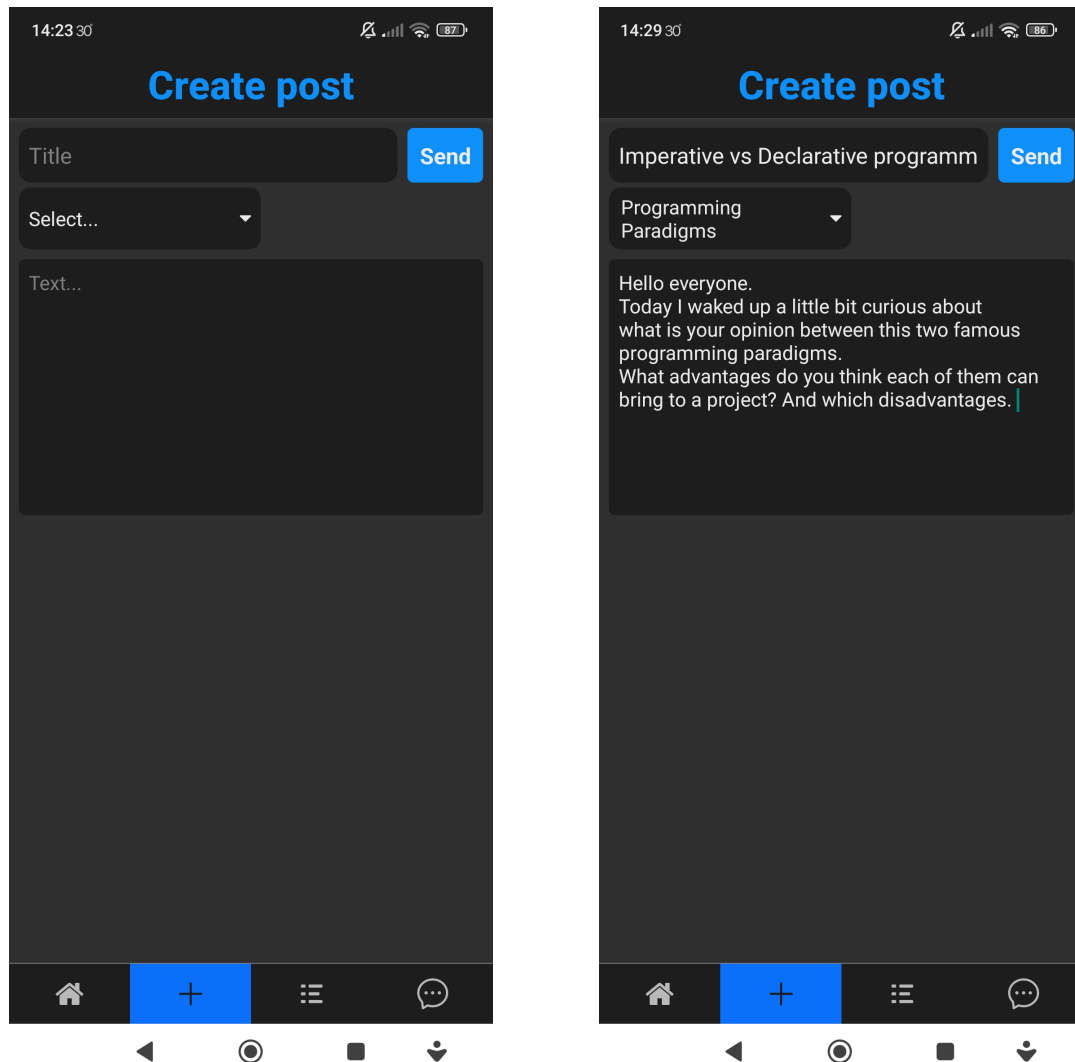
(A) Llistat d'usuaris seguidors

(B) Llistat d'usuaris seguits

FIGURA 10.14: Captura de pantalla dels llistats de seguidors o d'usuaris seguits d'un usuari

10.6 Pantalla de creació de publicacions

Els usuaris poden crear noves publicacions dirigint-se a la pàgina de creació de publicacions. Aquesta és accessible a partir del menú de navegació per pestanyes inferior, i presenta un formulari on l'usuari haurà de descriure la publicació amb títol i una descripció d'acord, a més d'indicar a quin fòrum volen que la publicació sigui afegida.



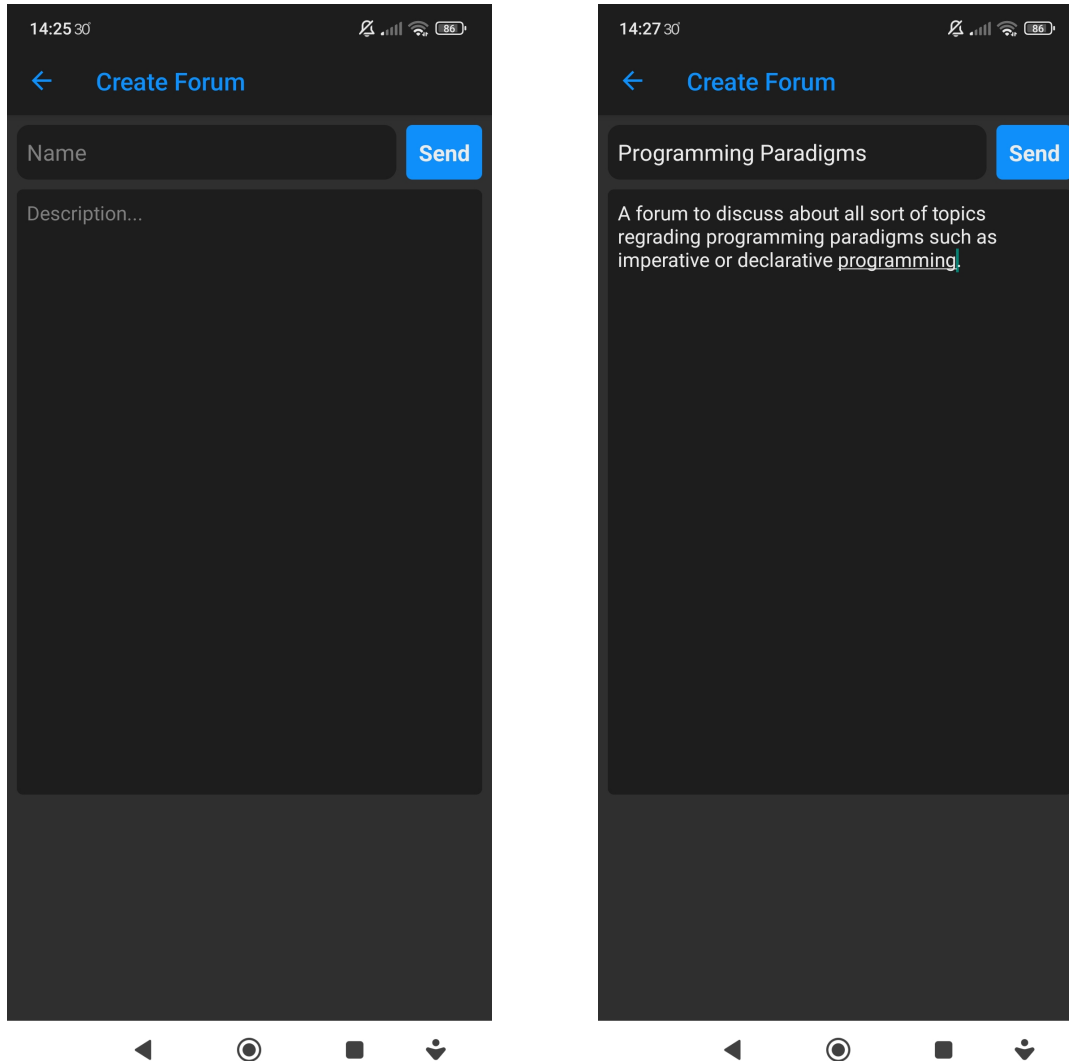
(A) Estat inicial de la pàgina

(B) Dades entrades per la publicació

FIGURA 10.15: Captura de pantalla de la pàgina de creació de publicacions

10.7 Pantalla de creació de fòrums

Quan un usuari vol crear una publicació, disposa d'un desplegable per escollir a quin fòrum vol afegir-la. Si l'usuari no troba un fòrum adient al tema tractat per la publicació, aquest disposa de l'opció de crear un nou fòrum. Aquesta opció redirigeix a l'usuari a la pàgina de creació de fòrums en la que haurà d'entrar el nom del fòrum, i la descripció per indicar quins temes s'haurien de tractar en aquest.



(A) Estat inicial de la pàgina

(B) Dades entrades pel fòrum

FIGURA 10.16: Captura de pantalla de la pàgina de creació de fòrums

10.8 Pantalla d'una publicació

Prement les targetes d'un llistat de publicacions podem accedir a la pàgina específica de la publicació. En aquesta podem visualitzar la descripció completa de la publicació, així com la mateixa informació que trobem a les targetes: el comptador de *likes*, i la capçalera amb els enllaços a la pàgina del fòrum i el de l'usuari que l'ha creat. A part d'això, també disposem dels comentaris que els usuaris han pogut afegir a la publicació.

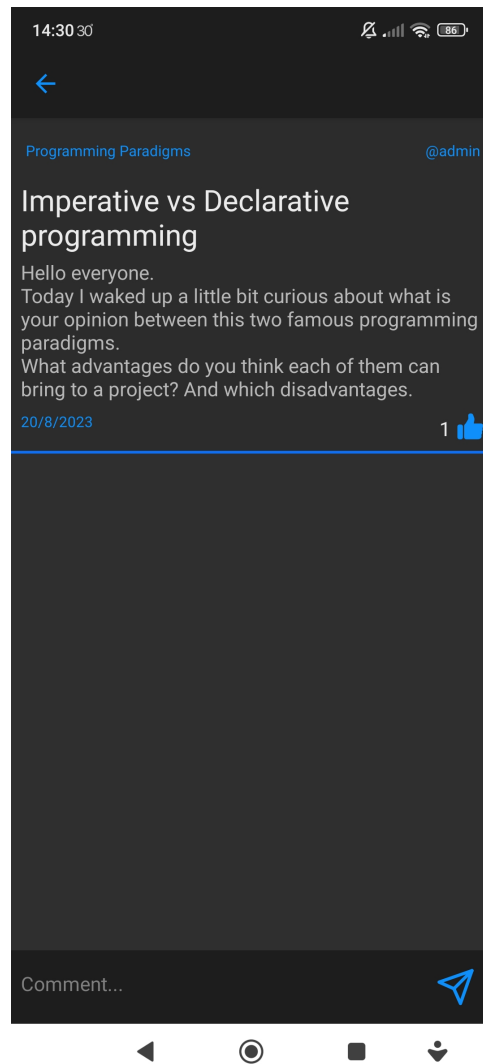
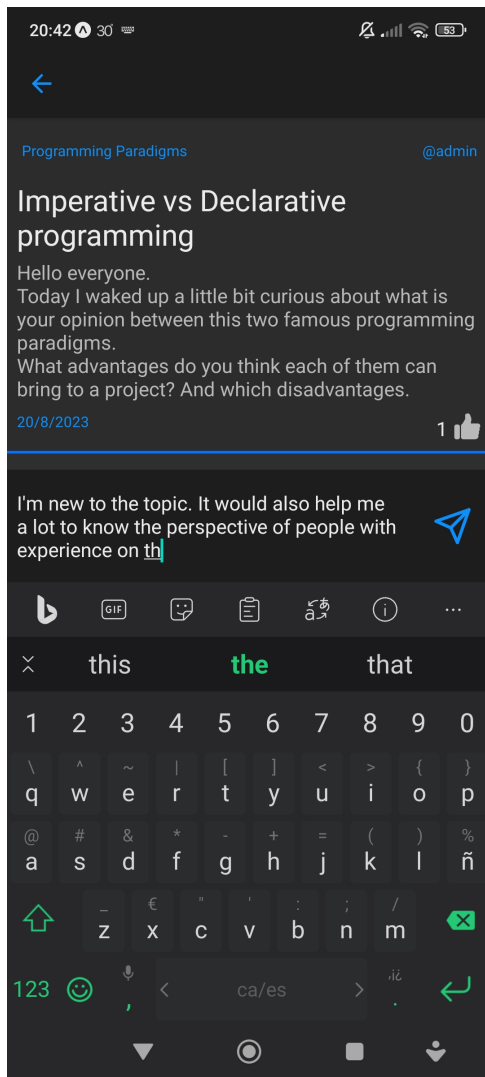
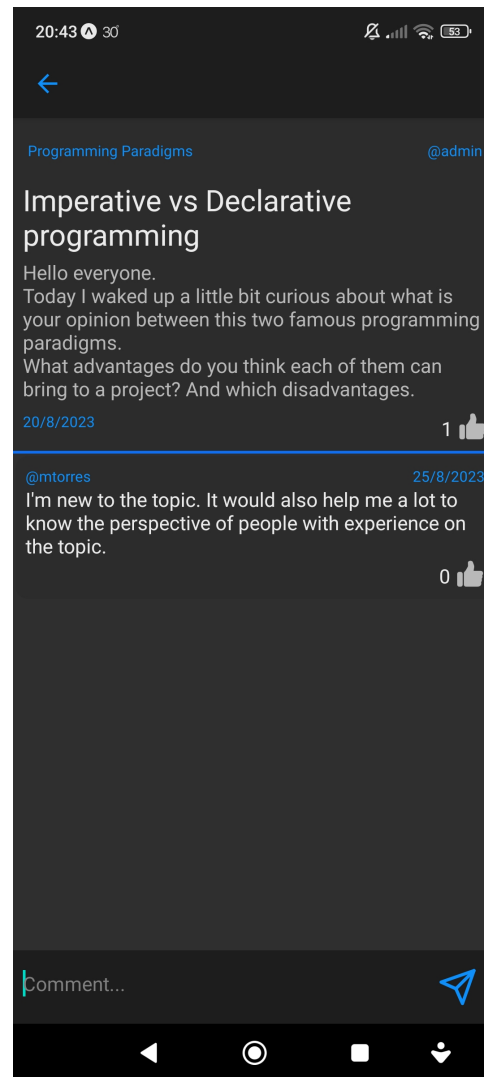


FIGURA 10.17: Captura de pantalla de la pàgina d'una publicació que s'ha reaccionat com que l'hi agrada a l'usuari

Quan un usuari vulgui afegir un comentari, només ha d'entrar el contingut de text d'aquest a l'entrada que es troba a la part inferior de la pantalla, i aquest s'afegirà a la publicació.



(A) Escrivint un comentari



(B) Comentari afegit a la publicació

FIGURA 10.18: Captura de pantalla del comportament d'afegir un comentari a una publicació a la pàgina d'aquesta.

El comportament dels comptadors de *likes* dels comentaris és el mateix que el que ens trobem amb els comptadors de *likes* d'una publicació.

10.9 Pantalla d'un fòrum

La pàgina d'un fòrum ens permet visualitzar les publicacions fetes en aquesta, així com l'opció d'unir-se o deixar de ser membre d'aquest i veure el llistat de membres del fòrum.

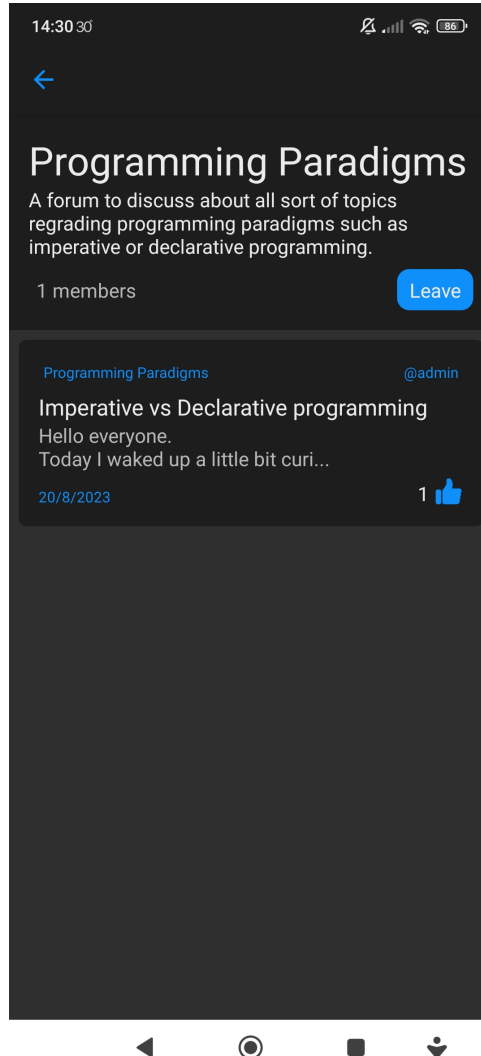
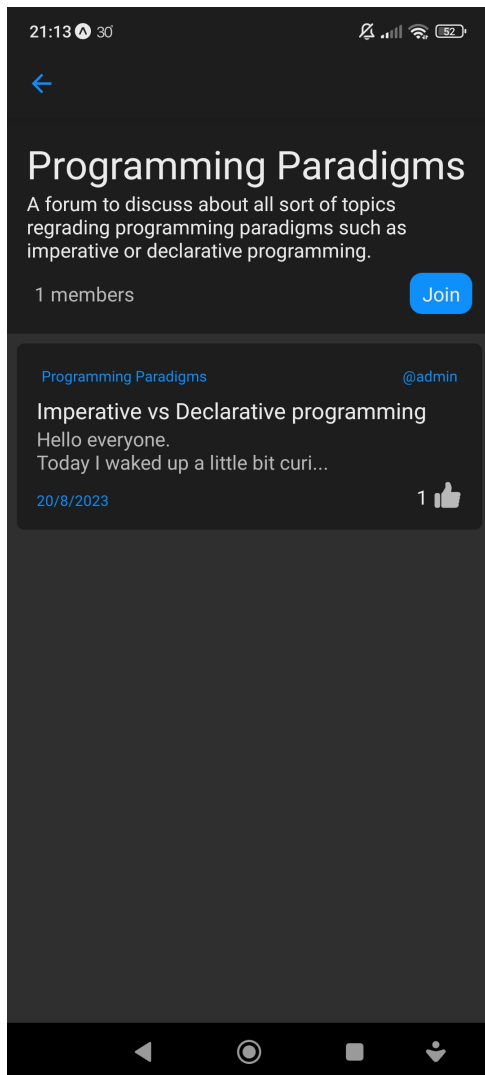
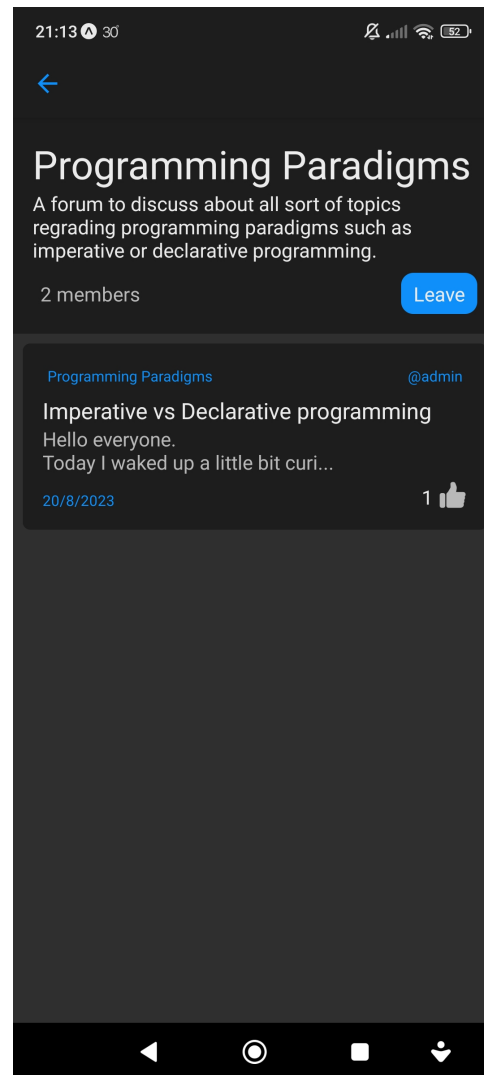


FIGURA 10.19: Captura de pantalla de la pàgina d'un fòrum

Per tal d'unir-se a un fòrum, l'usuari haurà d'accedir a la pàgina d'aquest. Un cop en aquesta haurà de prémer el botó d'unir-se (**Join**) i se l'afegirà com a nou membre del fòrum.



(A) Abans d'unir-se



(B) Després d'unir-se

FIGURA 10.20: Captura de pantalla de la pàgina d'un fòrum amb el comportament per unir-se com a membre d'aquest

Per visualitzar el llistat de membres només cal prémer el comptador de membres que podem trobar a l'esquerra del botó d'unir-se/abandonar el fòrum. Aquest ens redirigirà a la pàgina amb el llistat dels membres del fòrum. En cas de prémer alguna de les targetes dels usuaris membres del fòrum, l'usuari serà redirigit al perfil d'usuari del membre seleccionat.

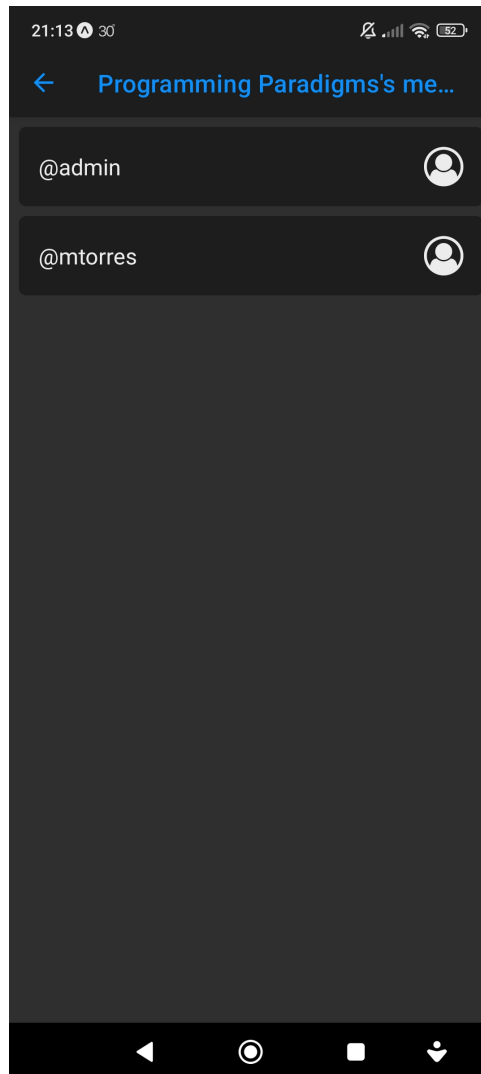


FIGURA 10.21: Captura de pantalla del llistat de membres d'un fòrum.

10.10 Pantalla de tasques

Cada usuari de l'aplicació disposa de la seva pròpia llista de tasques. Aquesta llista és accessible per mitjà del menú de navegació per pestanyes inferior.

Un cop accedida a la pantalla de les tasques, l'usuari podrà visualitzar les seves tasques. Aquestes es mostraran en format de targetes amb el nom, l'indicador de si està completada i el botó per eliminar-la.

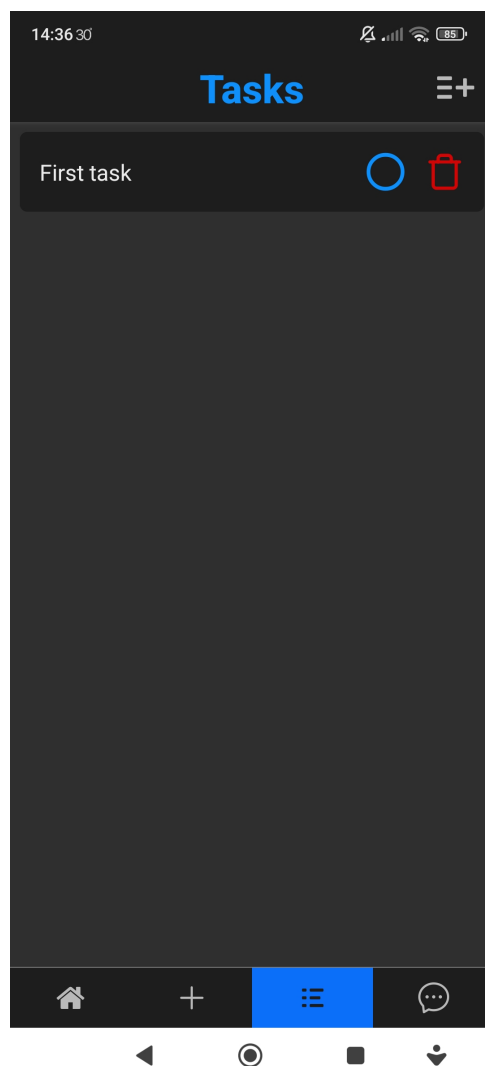
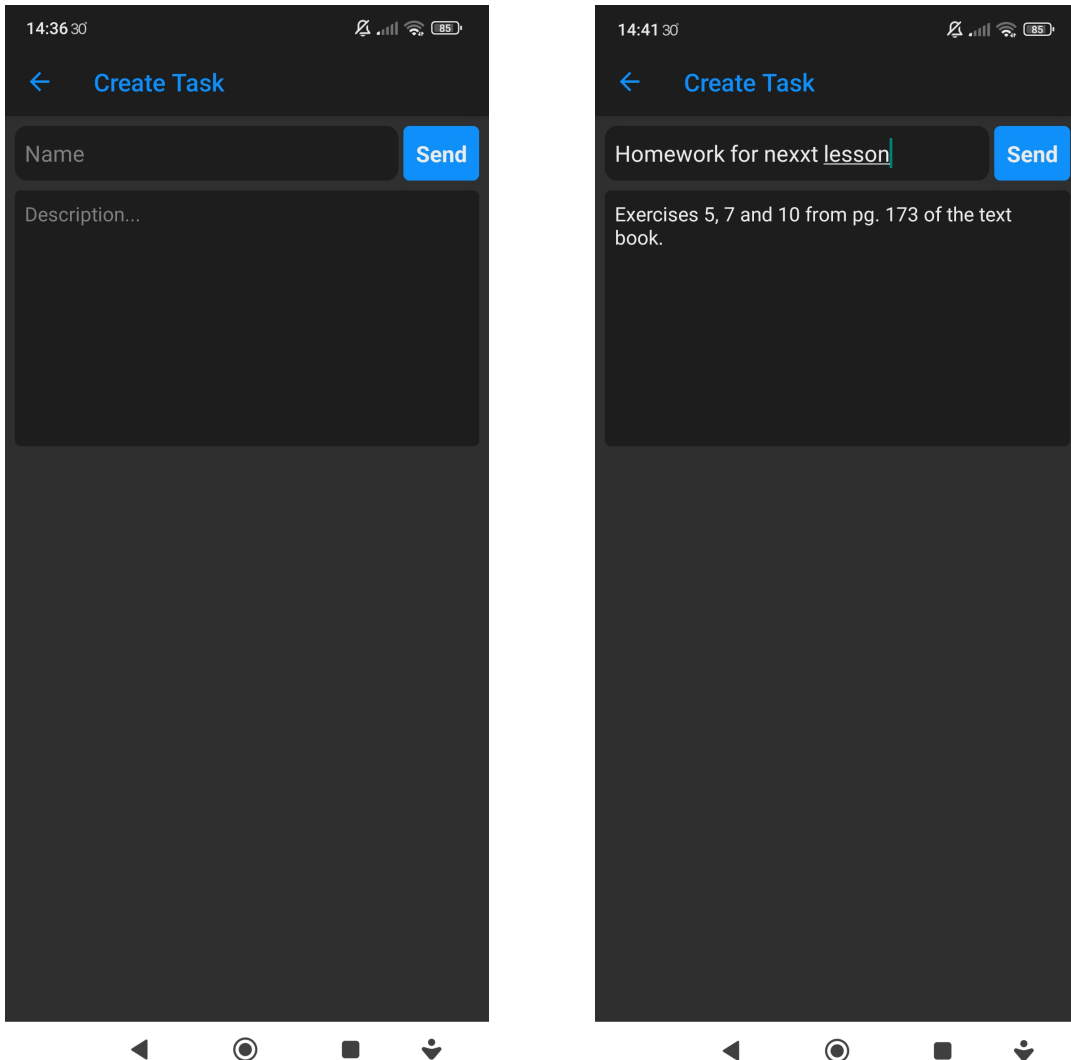


FIGURA 10.22: Captura de pantalla de la llista de tasques de l'usuari

Per tal de crear una nova tasca, els usuaris poden accedir a la pàgina de creació de tasques per mitjà del botó de la capçalera per afegir noves tasques (E+). En aquesta pantalla l'usuari haurà d'entrar el nom i, de forma opcional, una descripció de la tasca.



(A) Estat inicial de la pàgina

(B) Dades entrades pel formulari

FIGURA 10.23: Captures de pantalla de la pàgina de creació de tasques

Aquesta tasca creada s'afegeix al llistat de tasques de l'usuari i es mostra a la pàgina de tasques. En cas de voler modificar les dades relatives a la tasca, prement la targeta d'una tasca del llistat, es redirigirà a l'usuari a la pàgina d'edició de la tasca seleccionada on es presentarà el mateix formulari que a l'hora de crear la tasca, però amb els valors carregats del nom i la descripció de la tasca.

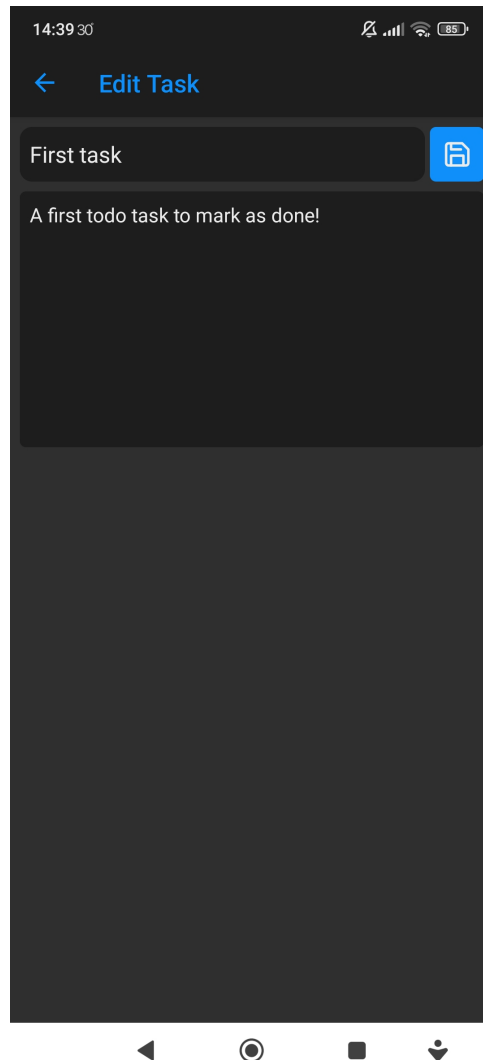
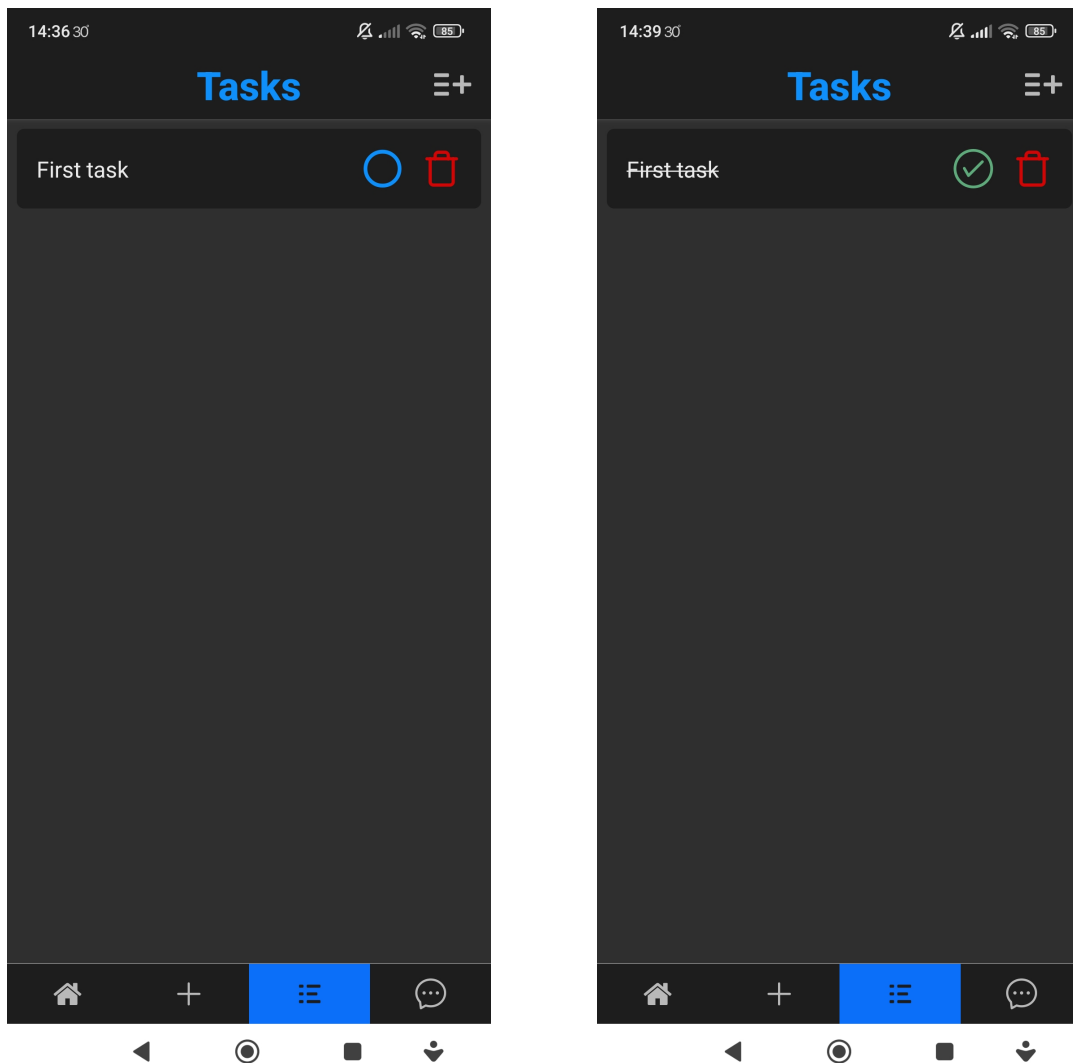


FIGURA 10.24: Captura de pantalla de pàgina d'edició d'una tasca

Una altra funcionalitat del llistat de tasques és el de marcar i desmarcar aquestes amb l'estat de "completada". Per tal de canviar l'estat de la tasca, l'usuari haurà de prémer al botó de l'estat de la tasca (○) per alternar el valor actual d'aquest.



(A) Tasca amb estat "incompleta"

(B) Assignat l'estat "completada" a la tasca

FIGURA 10.25: Captures de pantalla del comportament d'assignació de l'estat "completada" a una tasca del llistat de tasques

Finalment, si l'usuari té tasques completades, o simplement vol eliminar alguna tasca, pot prémer el botó per eliminar una tasca (🗑️). Aquest presentà un avís a l'usuari abans d'executar la comanda per tal d'assegurar que la petició és correcta i l'usuari vol eliminar la tasca. Si l'usuari confirma l'acció, aquesta tasca s'elimina de la llista.

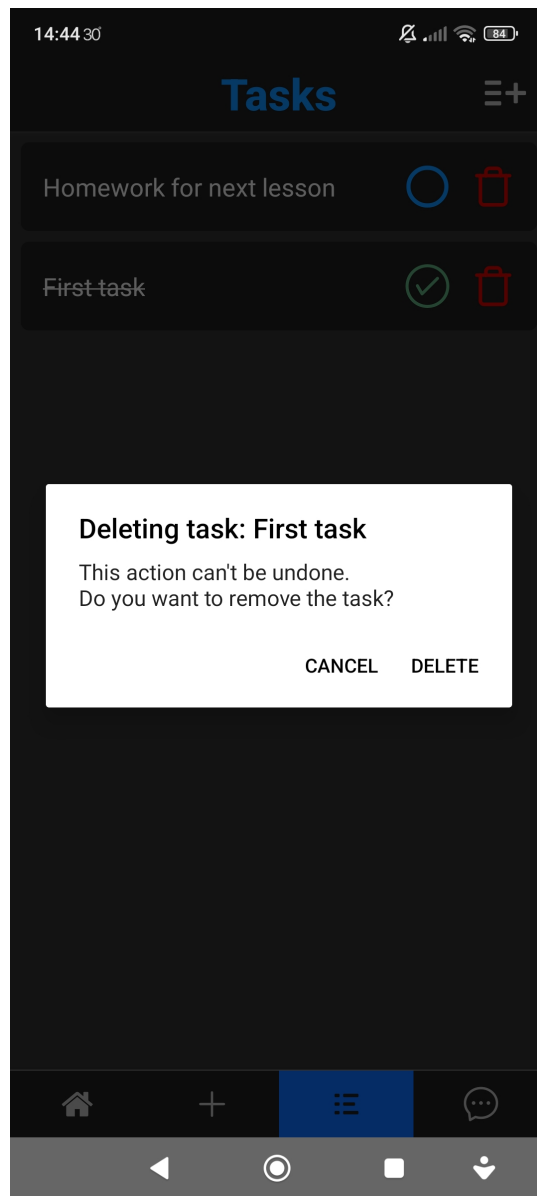


FIGURA 10.26: Captura de pantalla de l'avís de confirmació de l'usuari per la funcionalitat per eliminar tasques

10.11 Pantalla de xats

L'última pàgina accessible pel menú de navegació per pestanyes inferior és el del llistat de xats de l'usuari. En aquest llistat es mostren els diferents xats que l'usuari té, ja siguin de missatgeria directa amb un altre usuari, o xats de grup. En aquesta pàgina també trobem l'opció per poder crear nous xats de grup amb el botó de la capçalera (👤+).

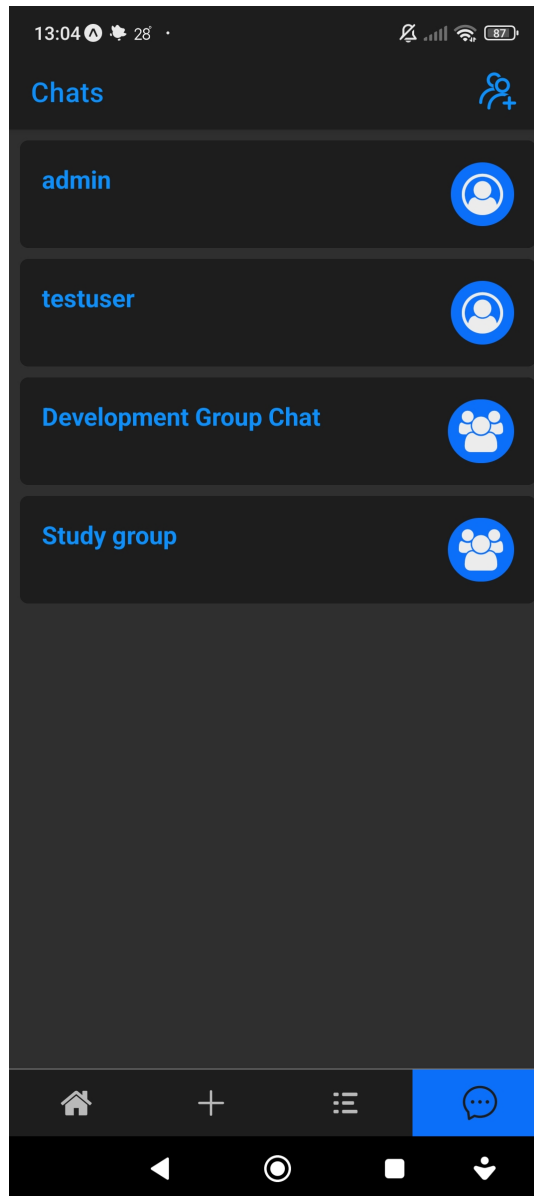
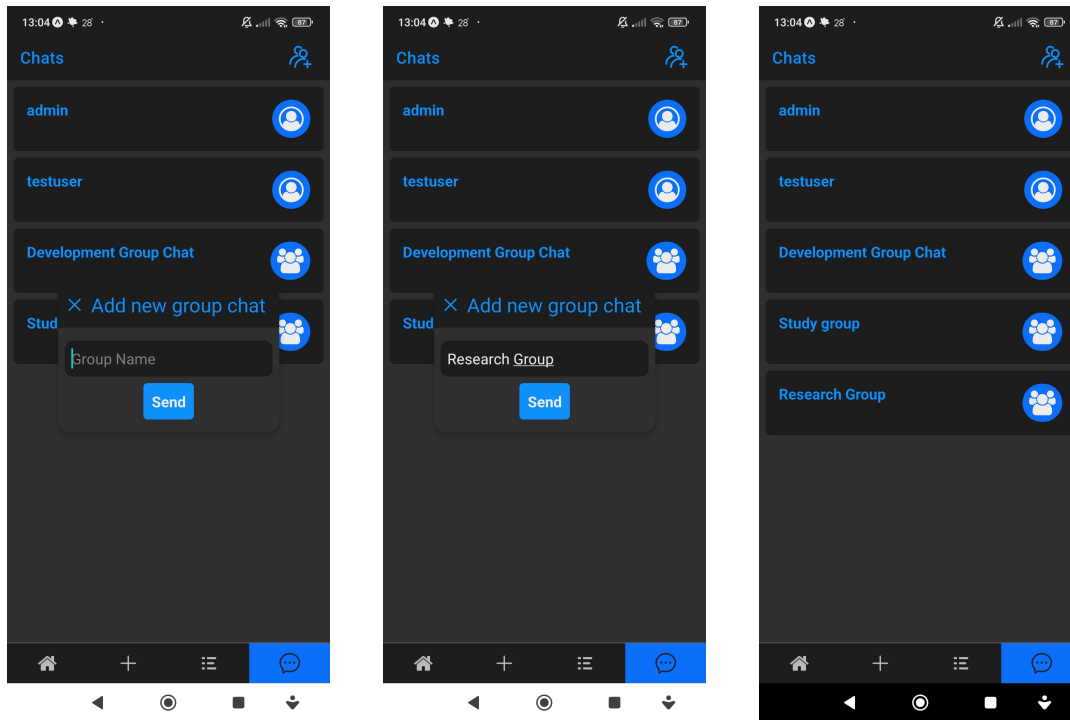


FIGURA 10.27: Captura de pantalla del llistat de xats de l'usuari

En prémer en el botó de creació de xat, es carregarà el modal per la creació de xats on l'usuari haurà d'entrar el nom pel nou grup. Un cop creat, aquest s'afegirà al llistat de xats.



(A) Estat inicial del modal

(B) Nom entrat pel nou grup

(C) Nou grup afegit al llistat

FIGURA 10.28: Captures de pantalla del comportament de la creació d'un nou xat de grup

10.12 Pantalla d'un xat

Igual que amb les pantalles dels perfils d'usuaris, les pantalles de xats disposen de dues vistes, una per veure un canal de missatgeria directa entre dos usuaris, i un altre per veure un canal d'un grup d'usuaris. Podem considerar els xats de missatgeria directa com la base, i els canals de grups la seva extensió.

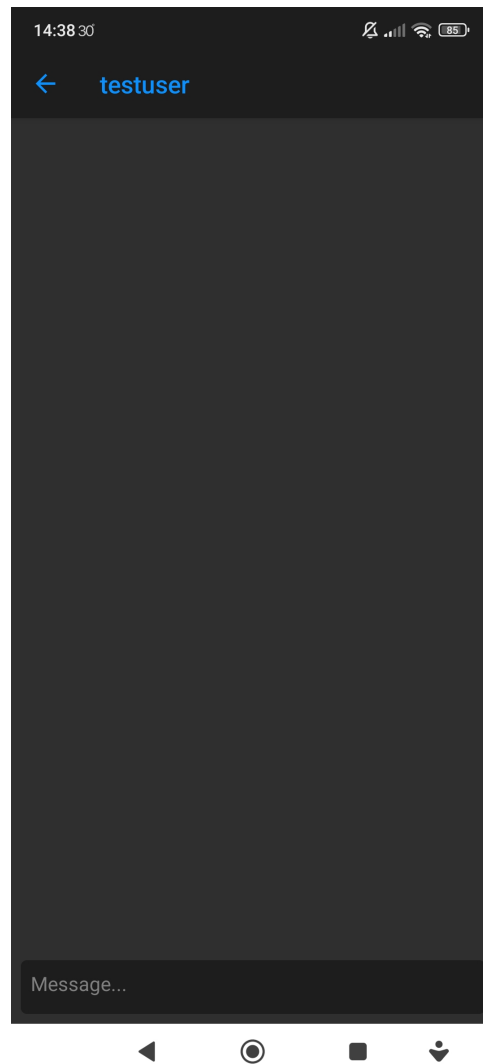
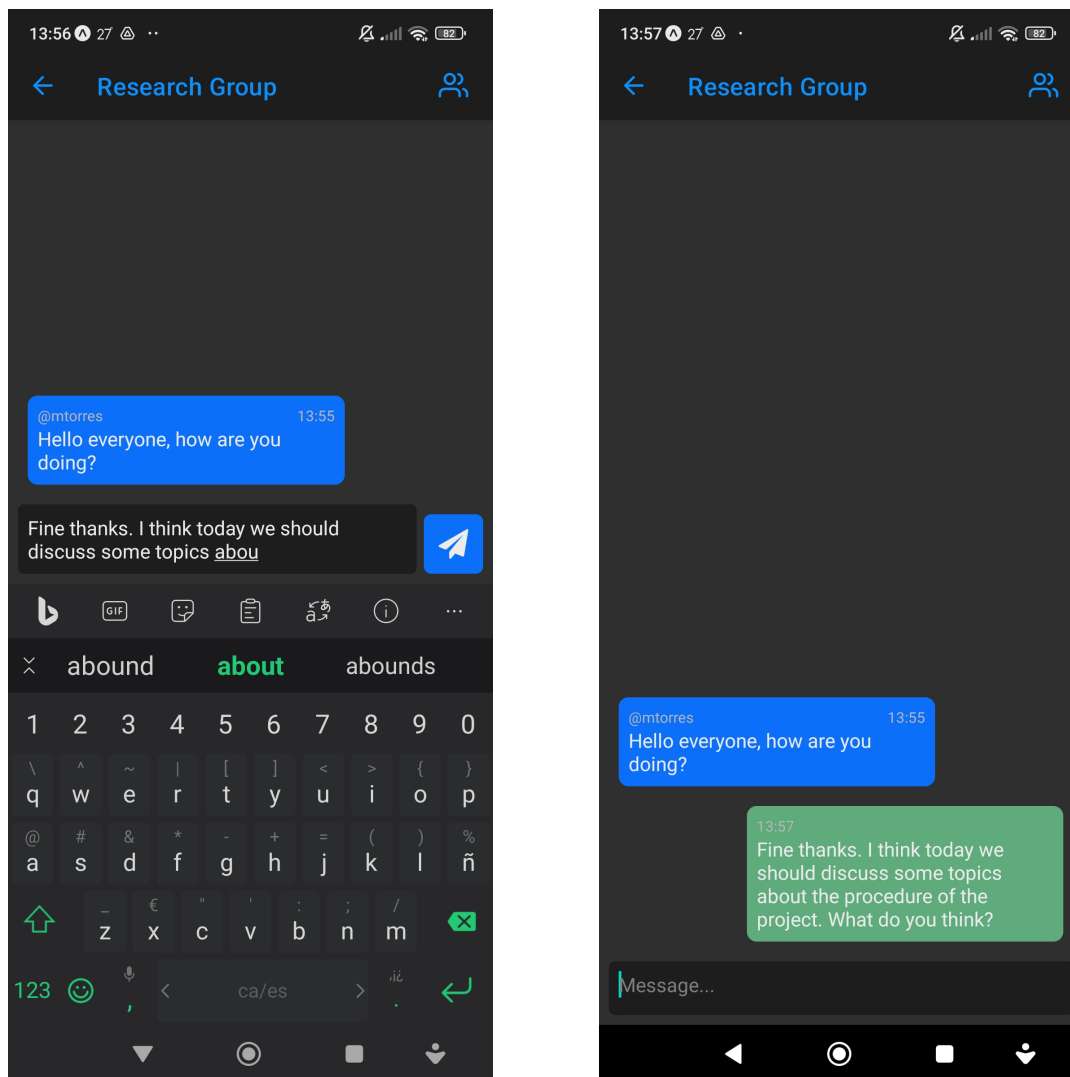


FIGURA 10.29: Captura de pantalla d'un xat de missatgeria directa

Com a qualsevol xat, els usuaris poden escriure i enviar missatges. Aquests missatges s'afegeixen aleshores al llistat de missatges del xat i altres usuaris poden visualitzar-los.



(A) Escrivint el missatge

(B) Missatge afegit al xat

FIGURA 10.30: Captura de pantalla del comportament d'enviar un missatge per un canal de xat.

10.12.1 Pantalla d'un xat de grup

A diferència dels canals de missatgeria directa, els xats de grup disposen d'algunes funcionalitats més a part d'enviar i visualitzar els missatges d'aquest. En aquest, els usuaris del xat, podran veure el llistat de membres del xat i afegir-n'hi de nous.

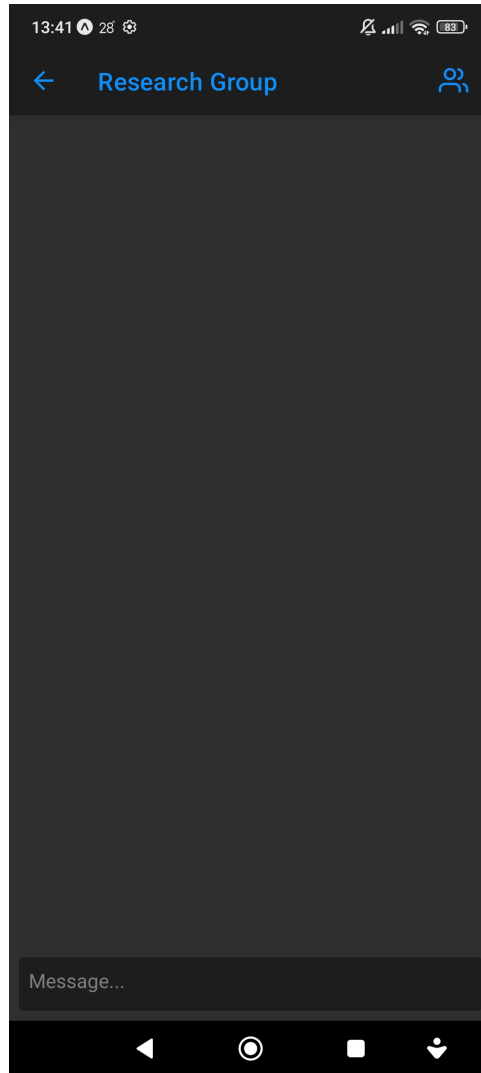


FIGURA 10.31: Captura de pantalla de la sala d'un xat de grup

Per tal de visualitzar el llistat de membres, l'usuari haurà de prémer el botó de la capçalera (👤). Aquest obrirà un model en format de pantalla on apareixerà el llistat de les targetes dels usuaris membres del xat. Prement aquestes targetes, l'usuari serà redirigit al perfil de l'usuari seleccionat.

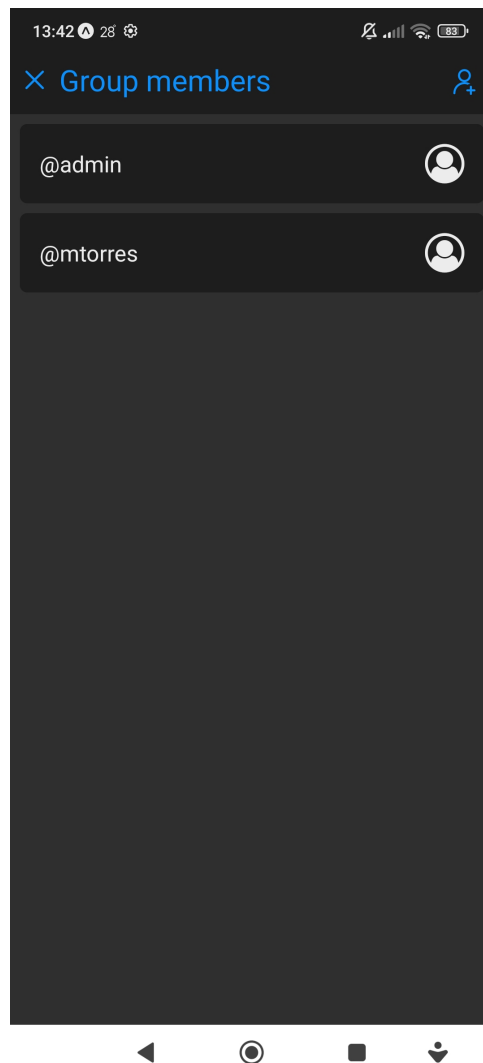


FIGURA 10.32: Captura de pantalla del llistat de membres d'un xat de grup

A partir d'aquest modal del llistat de membres, si un usuari vol afegir a algun membre nou al xat, prement el botó per afegir nous membres a la capçalera (👤+), s'obrirà un nou modal amb els usuaris seguits per l'usuari que vol afegir membres i que no siguin membres dels xats. Aquest llistat es genera amb la proposta dels *outsiders* presentada al capítol 9, secció 9.2.4, on s'explicava el problema relatiu a la recuperació d'aquests per la generació d'aquest llistat, així com la solució presentada.

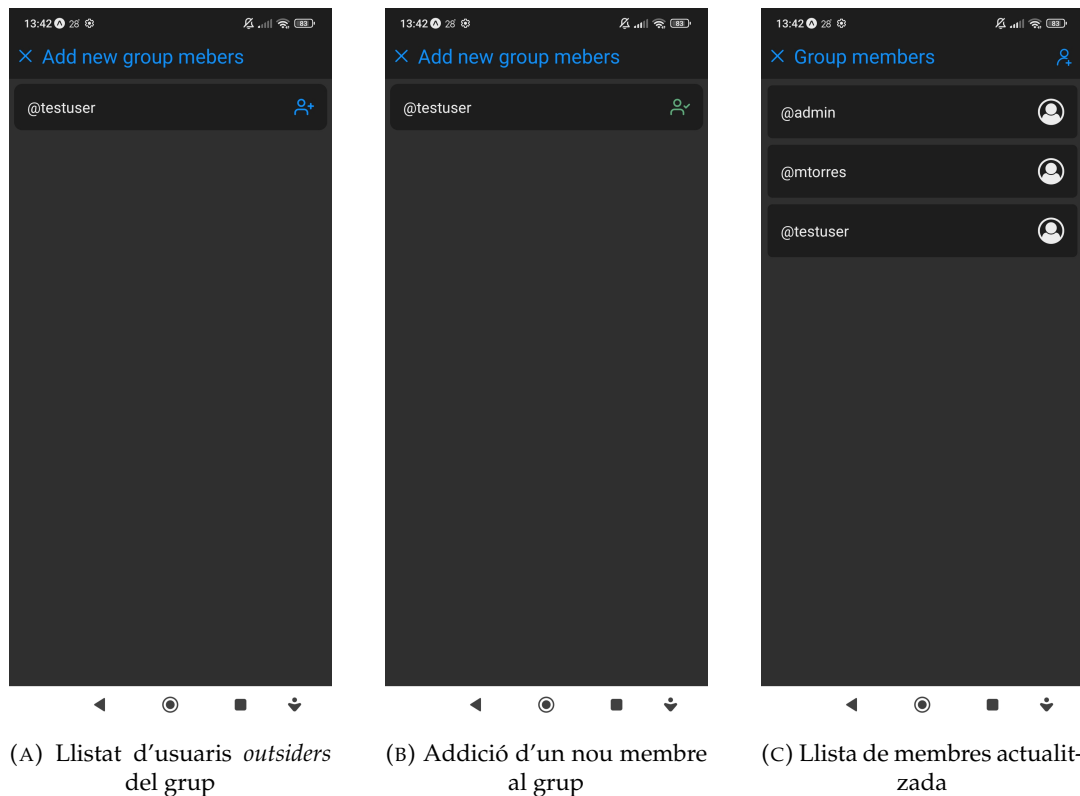


FIGURA 10.33: Captures de pantalla del comportament per afegir nous membres a un xat de grup

Com podem veure a la figura 10.33, on es mostra el procés per afegir nous membres a un xat de grup, per tal d'afegir a un nou membre, l'única cosa que l'usuari ha de fer al llistat d'*outsiders* és prémer el botó per afegir-ho al grup (👤+).

Capítol 11

Conclusions

Els coneixements obtinguts amb l'ORM *Sequelize* així com la llibreria d'*Express*, m'han ajudat al desenvolupament de l'API REST, i m'han permès descobrir una nova forma de desenvolupar aquestes. Per altra banda, considero que la llibreria *React*, com també el Framework *React Native*, han estat eines molt interessants, apassionants i satisfactòries d'aprendre, i crec que els coneixements adquirits per aquestes tecnologies em seran de molta utilitat per futurs projectes.

Fent una vista enrere, el resultat obtingut per a l'aplicació desenvolupada compleix en gran manera les especificacions i objectius establerts a l'inici del projecte. Si bé no s'ha aconseguit implementar les funcionalitats relatives als usuaris amb rols especials ni la interacció en temps real pels canals de xat, l'aplicació desenvolupada dotà als usuaris de les funcionalitats establertes inicialment pels usuaris bàsics.

Aquestes funcionalitats de les quals disposa l'aplicació pels seus usuaris, proporcionen la funció de fòrum per la qual inicialment es va decidir a desenvolupar aquest projecte. Els usuaris, aleshores, poden crear fòrums, publicacions i comentaris, així com interactuar amb aquestes publicacions i comentaris per mitjà de reaccions. Per altra banda, els usuaris també disposen de les relacions de seguiment entre ells, la qual cosa els permet establir una xarxa de seguiments amb altres usuaris amb els quals podran interactuar per missatgeria, ja sigui per xats de missatges directes com per canals de grup. Finalment, els usuaris disposen dels seus llistats de tasques, que si bé no és una funcionalitat comuna en l'àmbit d'aplicacions de tipus fòrums, crec que pot ser una bona eina auxiliar pels usuaris que decideixin enfocar l'ús d'aquesta aplicació en un àmbit més acadèmic per mantenir un registre de les tasques que han de fer.

Comparant la proposta desenvolupada amb aplicacions similars al mercat com poden ser *WhatsApp*, per les funcionalitats de missatgeria, o *9GAG*, per les funcionalitats de fòrums, o *Reddit*, pels dos tipus de funcionalitats, he d'admetre que encara veig marge de millorar per moltes funcionalitats com poden ser la creació de publicacions i fòrums, millores en presentació de la visualització de publicacions per ser més relatives als fòrums dels que els usuaris siguin membres o als usuaris que se segueixin, etc. Tot i això, puc dir que estic satisfet amb les funcionalitats desenvolupades que sí que han vist la llum en aquest projecte i han pogut estar implantades a l'aplicació final d'aquest.

Un altre punt a considerar és el plantejament inicial del projecte. Si bé no considero errada la decisió de desenvolupar aquest projecte amb *React Native*, i la voluntat

de també fer-ho amb *Electron.js*, ja que un dels objectius d'aquest projecte era l'aprenentatge d'aquest tipus d'eines multiplataforma, opino que en cas de tornar a fer el projecte des de zero, i amb un objectiu més comercial que d'aprenentatge, el desenvolupament inicial de la plataforma seria com una aplicació web. Com es va comentar a la introducció d'aquest projecte, una plataforma web seria l'opció més còmoda per posar a disposició de tothom aquesta plataforma, tot i que no s'abandonaria el desenvolupament de l'aplicació pels dispositius mòbils un cop finalitzada l'aplicació web.

En definitiva estic satisfet amb els coneixements i resultats obtinguts, amb ganes de continuar amb el projecte i també començar de nous utilitzant les tecnologies estudiades, i considero que el desenvolupament d'aquest projecte, tant la part d'implementació com la part de planificació i disseny, ha estat en conjunt una experiència enriquidora.

Capítol 12

Treball futur

Tal com s'ha mencionat breument a les conclusions del projecte (capítol 11), no s'han pogut implementar alguns objectius funcionals a la versió final del projecte. Per tant, com a objectiu principal del treball futur es troba la finalització d'aquests objectius originals.

La missatgeria en temps real dels xats ha estat una funcionalitat complicada d'implementar, i no ha pogut estar present finalment a les funcionalitats implantades per l'aplicació resultant d'aquest projecte. Per aquest motiu, aquesta és la primera i més important de les funcionalitats a implementar en la continuació del treball.

Seguida a aquesta funcionalitat, també caldrà afegir la implantació dels rols *moderadors de fòrums* i *administradors de xats* al sistema amb l'addició de les funcionalitats relatives. Aleshores, aquests rols disposen les eines per moderar o administrar els continguts de fòrums i xats respectivament, amb la possibilitat d'eliminació de continguts en cas que es facin publicacions que no compleixin les condicions establertes pels administradors d'un fòrum, o en cas d'enviament de missatges que ataquin els drets d'un altre usuari d'un grup.

Tot i que la implantació de la plataforma com a aplicació d'escriptori era un d'aquests objectius originals, pel desenvolupament de la plataforma accessible per dispositius d'escriptori s'ha optat per fer el desenvolupament de l'aplicació per navegadors web en la continuació del projecte. Per tal de desenvolupar aquesta aplicació, s'haurà de fer un estudi de viabilitat de la reutilització del projecte per dispositius mòbils utilitzant el paquet *react-native-web*, que atorga la implementació dels components de *React Native* per navegadors web, o si seria millor opció recrear el projecte només amb *React* pels navegadors web.

Per altra banda, les funcionalitats secundàries establertes també a l'inici del projecte, així com les funcionalitats esmentades a les conclusions que no formen part dels objectius inicials, també formen part d'aquest treball futur com també la millora de funcionalitats ja existents per fer-les més intuïtives pels usuaris. Dins d'aquestes funcionalitats es consideren opcions com:

- Administració de missatges propis d'un xat (edició o eliminació del missatge).
- Addició de recordatoris i dates de finalització de les tasques.
- Enviament de missatges d'àudio o imatges pels canals de xat.

Bibliografía

- [1] TM/[®]Google LLC. *Android Studio Icon*. URL: https://es.m.wikipedia.org/wiki/Archivo:Android_Studio_Trademark.svg.
- [2] JetBrains. *PyCharm Icon*. URL: https://commons.wikimedia.org/wiki/File:PyCharm_Icon.svg.
- [3] Microsoft. *Visual Studio Icon*. URL: https://es.m.wikipedia.org/wiki/Archivo:Visual_Studio_Code_1.35_icon.svg.
- [4] node.js authors. *NodeJS Icon*. URL: https://commons.wikimedia.org/wiki/File:Node.js_logo.svg.
- [5] Font Awesome. *NPM Icon*. URL: https://commons.wikimedia.org/wiki/File:Font_Awesome_5_brands_npm.svg.
- [6] NodeJS.org. *Node.js*. URL: <https://nodejs.org/es>.
- [7] Kanbanize. *Kanban Planning - Killing the Gantt Chart*. URL: <https://kanbanize.com/kanban-resources/kanban-software/what-is-kanban-planning>.
- [8] Talent. *Salario medio para Desarrollador Full Stack en España, 2023*. URL: <https://es.talent.com/salary?job=desarrollador+full+stack>.
- [9] NodeJS.org. *About | Node.js*. URL: <https://nodejs.org/en/about>.
- [10] NodeJS.org. *Wikipedia. Node.js*. URL: <https://es.wikipedia.org/wiki/Node.js>.
- [11] Diana. *Qué es Node.js: Casos de uso comunes y cómo instalarlo*. 2023. URL: <https://www.hostinger.es/tutoriales/que-es-node-js>.
- [12] Fireship. *Node.js Ultimate Beginner's Guide in 7 Easy Steps*. 2020. URL: <https://www.youtube.com/watch?v=ENrzd9HAZK4>.
- [13] Nicole Abramowski. *What is NPM? A Beginner's Guide*. 2022. URL: <https://careerfoundry.com/en/blog/web-development/what-is-npm/>.
- [14] Telusko. *What is NPM? | Node JS*. URL: <https://www.youtube.com/watch?v=M8nlgBQBxxo&list=PLsyeobzWxl7occsESx2X1E2R2Uw5wCoeG&index=130>.
- [15] *Introducción a Express/Node*. URL: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction.
- [16] avck. *What is minimum system requirements to run electron apps?* 2016. URL: <https://stackoverflow.com/questions/36306450/what-is-minimum-system-requirements-to-run-electron-apps>.
- [17] Traversy Media. *Create a Desktop App With JavaScript & Electron*. 2022. URL: <https://www.youtube.com/watch?v=ML743nrkMHw>.
- [18] *HTTP response status codes*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>.
- [19] IBM. *What is a REST API?* Inf. tèc. IBM. URL: <https://www.ibm.com/topics/rest-apis>.

- [20] Fireship. *RESTful APIs in 100 Seconds // Build an API from Scratch with Node.js Express*. URL: <https://www.youtube.com/watch?v=-MTSQjw5DrM>.
- [21] John Au-Yeung i Ryan Donovan. "Best practices for REST API design". A: *The Overflow* (2020). URL: <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>.
- [22] Geshan Manandhar. "Build a REST API with Node.js, Express, and MySQL". A: *LogRocket* (2023). URL: <https://blog.logrocket.com/build-rest-api-node-express-mysql/>.
- [23] ESIC. *El ORM como herramienta eficiente de trabajo*. Inf. tèc. ESIC Business & Marketing School, 2018. URL: <https://www.esic.edu/rethink/tecnologia/el-orm-como-herramienta-eficiente-de-trabajo>.
- [24] Alejandro Garcia Serna. "Agiliza tu desarrollo en Nodejs con el ORM sequelize". A: (). URL: <https://ed.team/blog/agiliza-tu-desarrollo-en-nodejs-con-el-orm-sequelize>.
- [25] Sequelize. *Getting Started*. URL: <https://sequelize.org/docs/v6/getting-started/>.
- [26] Sequelize. *References*. URL: <https://sequelize.org/api/v6/identifiers.html>.
- [27] Chinedu Orié. "Build API with GraphQL, Node.js, and Sequelize". A: *dev.to* (2020). URL: <https://dev.to/nedsoft/build-api-with-graphql-node-js-and-sequelize-5e8e>.
- [28] Anju. *How to implement many to many association in sequelize*. 2014. URL: <https://stackoverflow.com/questions/22958683/how-to-implement-many-to-many-association-in-sequelize>.
- [29] Guy Levin. "4 Most Used REST API Authentication Methods". A: *RestCase* (2019). URL: <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/>.
- [30] Fireship. *Session vs Token Authentication in 100 Seconds*. 2020. URL: <https://www.youtube.com/watch?v=UBUNrFtufWo>.
- [31] bezkoder. "Node.js Express: JWT example | Token Based Authentication & Authorization". A: *BezKoder* (2023). URL: <https://www.bezkoder.com/node-js-jwt-authentication-mysql/>.
- [32] Auth0. *jsonwebtoken*. URL: <https://www.npmjs.com/package/jsonwebtoken?activeTab=readme>.
- [33] Prashant Ram. "Difference between Session Cookies vs. JWT (JSON Web Tokens), for session management." A: *Medium* (2021). URL: <https://medium.com/@prashantramnyc/difference-between-session-cookies-vs-jwt-json-web-tokens-for-session-management-4be67d2f066e>.
- [34] Sam Meech-Ward. *Hashing Passwords in Node and Express using bcrypt*. 2022. URL: https://www.youtube.com/watch?v=AzA_LTDoFqY.
- [35] *Guía adaptación de la LOPD a LOPDGDD en 2021*. Ayuda Ley Protección Datos. URL: <https://ayudaleyprotecciondatos.es/lopdgdd/>.
- [36] *Plantilla Política de Privacidad 2023 Gratis*. Ceporros. URL: <https://ceporros.com/plantilla-politica-de-privacidad-2021-gratis/>.
- [37] Wilson Page. *Best way to store chat messages in a database?* 2011. URL: <https://stackoverflow.com/questions/7063051/best-way-to-store-chat-messages-in-a-database>.
- [38] sairfan. *Storing huge file collection in database*. 2021. URL: <https://dba.stackexchange.com/questions/298983/storing-huge-file-collection-in-database>.

- [39] Brent Ozar. "Store Files in a File System, Not a Relational Database". A: *Brent Ozar* (2021). URL: <https://www.brentozar.com/archive/2021/07/store-files-in-a-file-system-not-a-relational-database/>.
- [40] Sandokan El Cojo. *Sequelize: Include.where filtering by a 'parent' Model attribute*. 2015. URL: <https://stackoverflow.com/questions/30052254/sequelize-include-where-filtering-by-a-parent-model-attribute>.
- [41] itaied. *Sequelize pagination*. 2016. URL: <https://stackoverflow.com/questions/38211170/sequelize-pagination>.
- [42] midudev. *Cómo usar async/await con el método .map de los Arrays*. 2021. URL: <https://midu.dev/como-usar-async-await-con-array-prototype-map/>.
- [43] borisdiaukur. *How to use an include with attributes with sequelize?* 2014. URL: <https://stackoverflow.com/questions/21883484/how-to-use-an-include-with-attributes-with-sequelize>.
- [44] Jasmine Wang. "A more in-depth look at Sequelize's Magic Methods". A: *Medium* (2020). URL: <https://medium.com/@jsmney/a-more-in-depth-look-at-sequelizes-magic-methods-428928c70d58>.
- [45] thenetimp. *Prevent Sequelize from outputting SQL to the console on execution of query?* 2015. URL: <https://stackoverflow.com/questions/28927836/prevent-sequelize-from-outputting-sql-to-the-console-on-execution-of-query>.
- [46] Javier Alonso Delgado. *How to set up Winston loggin with Sequelize properly?* 2021. URL: <https://stackoverflow.com/questions/67211877/how-to-set-up-winston-loggin-with-sequelize-properly>.
- [47] NewtonCode. *How to get Winston daily file rotate to log to corresponding level file*. 2017. URL: <https://stackoverflow.com/questions/45613624/>.
- [48] Coding Note. "How To Create Instances With Associations In Sequelize". A: *Medium* (2022). URL: <https://medium.com/@codingnotes/how-to-create-instances-with-associations-in-sequelize-6664cb9fb984>.
- [49] Muhammad Adam. "Cómo crear una aplicación de calculadora usando React Native: un tutorial paso a paso". A: *freeCodeCamp* (2023). URL: www.freecodecamp.org/espanol/news/como-crear-una-aplicacion-de-calculadora-usando-react-native-tutorial-paso-a-paso/.
- [50] Tari Ibaba. "When Exactly is the useEffect Hook Called in React?" A: *Coding Beauty* (2022). URL: <https://codingbeautydev.com/blog/react-when-is-useeffect-called/>.
- [51] Yonas Enchalew. "Most Common React Hooks". A: *Linkedin* (2023). URL: <https://www.linkedin.com/pulse/most-common-react-hooks-yonas-enchalew>.
- [52] React. *Escribir marcado con JSX*. URL: <https://es.react.dev/learn/writing-markup-with-jsx>.
- [53] Rohit Sharma. "10 Basic Concepts About React You Should Learn". A: *Let's React* (). URL: <https://www.letsreact.org/10-basic-concepts-about-react-you-should-learn/>.
- [54] Kingsley Ubah. "React Native Networking – How To Perform API Requests In React Native using the FetchAPI". A: *freeCodeCamp* (2023). URL: <https://www.freecodecamp.org/news/react-native-networking-api-requests-using-fetchapi/>.

- [55] React Native. *Networking*. URL: <https://reactnative.dev/docs/network>.
- [56] React Native. *Core Components and Native Components*. URL: <https://reactnative.dev/docs/intro-react-native-components>.
- [57] Siddhant Varma. "React Native Project Structure: A Best Practices Guide". A: *waldo* (2022). URL: <https://www.waldo.com/blog/react-native-project-structure>.
- [58] Fireship. *I built a chat app in 7 minutes with React & Firebase*. 2020. URL: <https://www.youtube.com/watch?v=zQyrwxMPm88>.
- [59] Fireship. *10 React Hooks Explained // Plus Build your own from Scratch*. 2021. URL: <https://www.youtube.com/watch?v=TNhaIS0Uy6Q>.
- [60] rnfirebase.io. *React Native Firebase*. URL: <https://rnfirebase.io>.
- [61] *expo/vector-icons13.0.0*. URL: <https://icons.expo.fyi/Index>.
- [62] Madhur. *How to add strike through on Text in react native?* 2017. URL: <https://stackoverflow.com/questions/45833682/how-to-add-strike-through-on-text-in-react-native>.
- [63] Dan's React Native Lab. *How to Create an Expo App with React-Native Firebase*. 2023. URL: <https://www.youtube.com/watch?v=mZlKwRV4MC8>.
- [64] React Native. *Alert*. URL: <https://reactnative.dev/docs/alert>.
- [65] Akash J. *How to remove a item/index from a FlatList in react native?* 2018. URL: <https://stackoverflow.com/questions/53855424/how-to-remove-a-item-index-from-a-flatlist-in-react-native>.
- [66] Guna Shekar Proddaturi. "Push Notifications in a React Native Application using Firebase, Node.js and Express.js". A: *Medium* (2021). URL: <https://medium.com/tech-iiitg/push-notifications-in-a-react-native-application-using-firebase-node-js-and-express-js-67b3391b6c25>.
- [67] Umar Hassan. "Get the Hours and Minutes From a Date in JavaScript". A: *linuxhint* (2022). URL: <https://linuxhint.com/get-the-hours-and-minutes-from-a-date-in-javascript/>.
- [68] ADITYA GUPTA. "How to Check If a Date Is Today in JavaScript: A Detailed Guide". A: *Code for geek* (2023). URL: codeforgeek.com/how-to-check-date-is-today-javascript/.
- [69] javadog36. *How do I use .toLocaleTimeString() without displaying seconds?* 2013. URL: <https://stackoverflow.com/questions/17913681/how-do-i-use-tolocaletimestring-without-displaying-seconds>.
- [70] humkins. *Draw horizontal rule in React Native*. 2017. URL: <https://stackoverflow.com/questions/43380260/draw-horizontal-rule-in-react-native>.
- [71] Rodrigo Figueroa. "How to clear input form with Formik React or How to reset form after submit? Formik React". A: *Medium* (2020). URL: <https://medium.com/geekculture/how-to-clear-input-form-with-formik-react-47a6ce27c0d9>.
- [72] Expo. *Expo Constants*. URL: <https://docs.expo.dev/versions/latest/sdk/constants/>.
- [73] React Native. *ActivityIndicator*. URL: <https://reactnative.dev/docs/activityindicator>.
- [74] React Native. *Core Components and APIs*. URL: <https://reactnative.dev/docs/components-and-apis>.
- [75] React Native. *Pressable*. URL: <https://reactnative.dev/docs/pressable>.

- [76] Anilkumar iOS - ReactNative. *How to get previous route name from React Navigation*. 2019. URL: <https://stackoverflow.com/questions/55942600/how-to-get-previous-route-name-from-react-navigation>.
- [77] rendom. *react-navigation: Detect when screen, tabbar is activated / appear / focus / blur*. 2018. URL: <https://stackoverflow.com/questions/50290818/react-navigation-detect-when-screen-tabbar-is-activated-appear-focus-blur>.
- [78] rili-live. *react-native-phone-input*. URL: <https://www.npmjs.com/package/react-native-phone-input>.
- [79] Evanss. *Validate phone number with Yup?* 2018. URL: <https://stackoverflow.com/questions/52483260/validate-phone-number-with-yup>.
- [80] henninghall. *react-native-date-picker*. URL: <https://www.npmjs.com/package/react-native-date-picker>.
- [81] Shahid Rizwan. "Building custom hooks in React to fetch Data". A: <https://dev.to> (2021). URL: <https://dev.to/shaedrizwan/building-custom-hooks-in-react-to-fetch-data-4ig6>.
- [82] Coding with CharismaAji. *React Native Axios, AsyncStorage and JSON Web Token (JWT) for Login Authentication*. 2022. URL: <https://www.youtube.com/watch?v=hMpcOLgLdlw>.
- [83] Expo. *Splash screen*. URL: <https://docs.expo.dev/develop/user-interface/splash-screen/>.
- [84] Expo. *Expo SplashScreen*. URL: <https://docs.expo.dev/versions/latest/sdk/splash-screen/>.
- [85] Javatpoint. *React Native Positioning Element with Flex*. URL: <https://www.javatpoint.com/react-native-positioning-element-with-flex>.
- [86] AdelRedaa97. *react-native-select-dropdown*. URL: <https://github.com/AdelRedaa97/react-native-select-dropdown#readme>.
- [87] jquense. *Yup*. URL: <https://github.com/jquense/yup/tree/pre-v1>.
- [88] calintamas. *react-native-toast-message*. URL: <https://www.npmjs.com/package/react-native-toast-message>.
- [89] kerim Erkan. *Where to store user's credentials in React Native Mobile App*. 2020. URL: <https://stackoverflow.com/questions/63315433/where-to-store-users-credentials-in-react-native-mobile-app>.
- [90] React Navigation. *Getting started*. URL: <https://reactnavigation.org/docs/getting-started/>.
- [91] Manish Mandal. "Config ESLint and Prettier in Visual Studio Code for React js development". A: *Medium* (2020). URL: <https://medium.com/how-to-react/config-eslint-and-prettier-in-visual-studio-code-for-react-js-development-97bb2236b31a>.
- [92] Expo-router vs React Navigation. *u/deadclown*. 2023. URL: https://www.reddit.com/r/reactnative/comments/14apzyj/exporouter_vs_react_navigation/.
- [93] Md Moin. "React Native hide show password input". A: *Medium* (2018). URL: <https://mdmoin07.medium.com/react-native-hide-show-password-input-d4be4d0f70aa>.
- [94] Matan Kastel. "React Native Cookie Authentication". A: *Medium* (2021). URL: <https://javascript.plainenglish.io/react-native-cookie-authentication-83ef6e84ba70>.
- [95] notJust.dev. *Building Navigation in React Native with Expo Router*. 2023. URL: <https://www.youtube.com/watch?v=yyGS0adZdsU>.

-
- [96] notJust.dev. *Let's build WhatsApp with React Native and AWS Amplify [p4]*. 2022. URL: <https://www.youtube.com/watch?v=pqQGU-JsD7k>.
- [97] midudev. *CURSO REACT NATIVE desde CERO - Creamos una app paso a paso*. 2022. URL: <https://www.youtube.com/watch?v=qi87b6VcIHY>.
- [98] JavaScript Mastery. *Build and Deploy a React Native App | 2023 React Native Course Tutorial for Beginners*. 2023. URL: <https://www.youtube.com/watch?v=mJ3bGvy0WAY>.
- [99] Simon Grimm. *You're doing React Native Routing wrong - Expo File-Based Routing*. 2023. URL: <https://www.youtube.com/watch?v=RglRiycD0oQ>.
- [100] React Native. *Security*. URL: <https://reactnative.dev/docs/security>.
- [101] Expo. *Introduction*. URL: <https://expo.github.io/router/docs/>.
- [102] Peter Kelvin Torver. "Understanding Cross Origin Resource Sharing (CORS)". A: <https://dev.to> (2022). URL: <https://dev.to/techfortified/understanding-cross-origin-resource-sharing-cors-2jbg>.

Apèndix A

Annex

En aquest apartat es poden trobar alguns fragments d'informació extra que no indispensables per comprendre amb exactitud el projecte, però que poden donar suport a l'hora d'entendre alguns conceptes.

A.1 Repositoris GitHub

A continuació es troben els enllaços als repositoris de GitHub pel desenvolupament de la plataforma *Learnum*:

- <https://github.com/marcTorresVilagut/LearnumAPI>
- <https://github.com/marcTorresVilagut/LearnumMobile>

A.2 Estructura de fitxers del projecte LearnumAPI

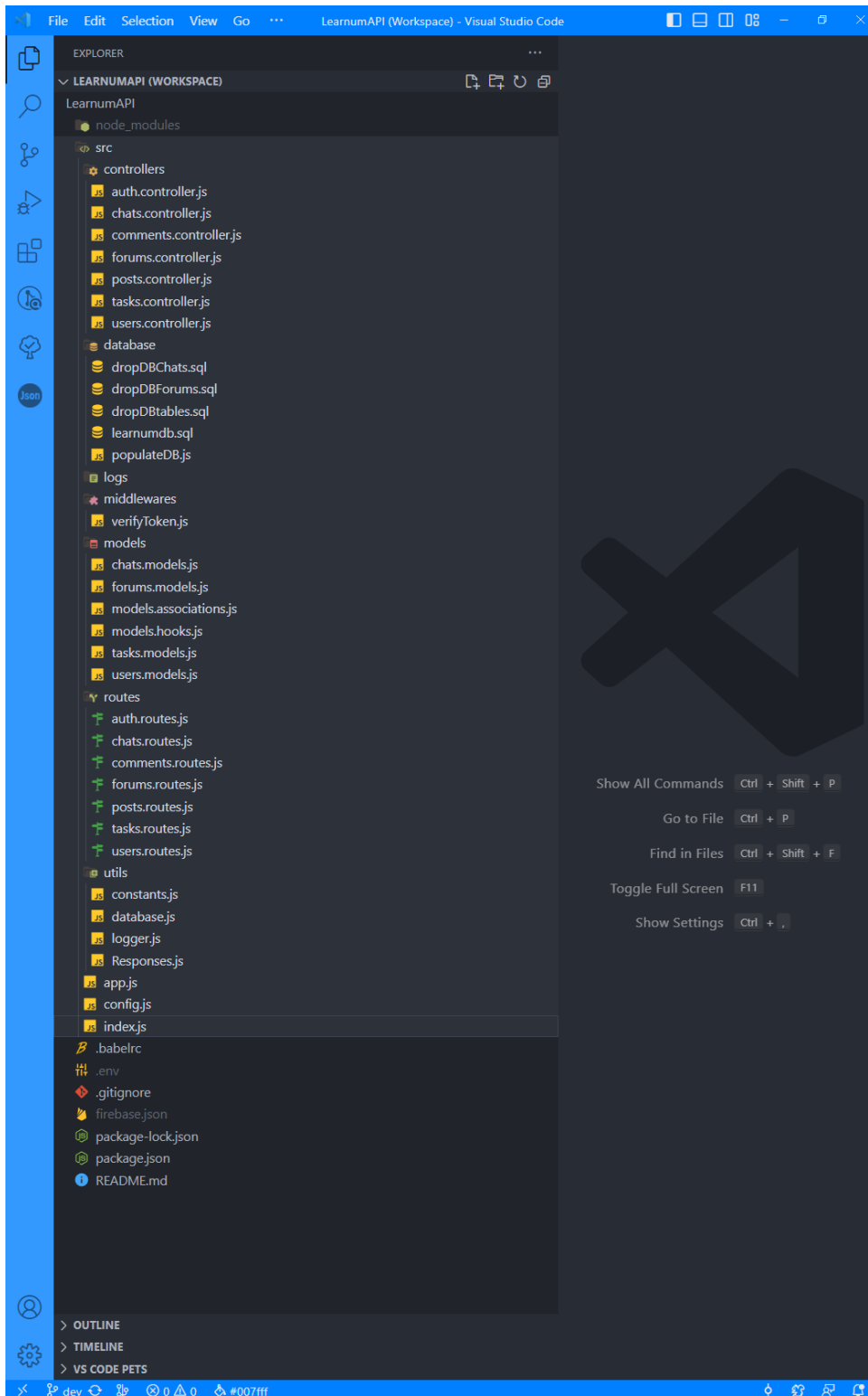


FIGURA A.1: Captura de pantalla de l'estructura de carpetes utilitzada per l'aplicació del servidor API REST del projecte *Learnum*

A.3 Estructura de fitxers del projecte LearnumMobile

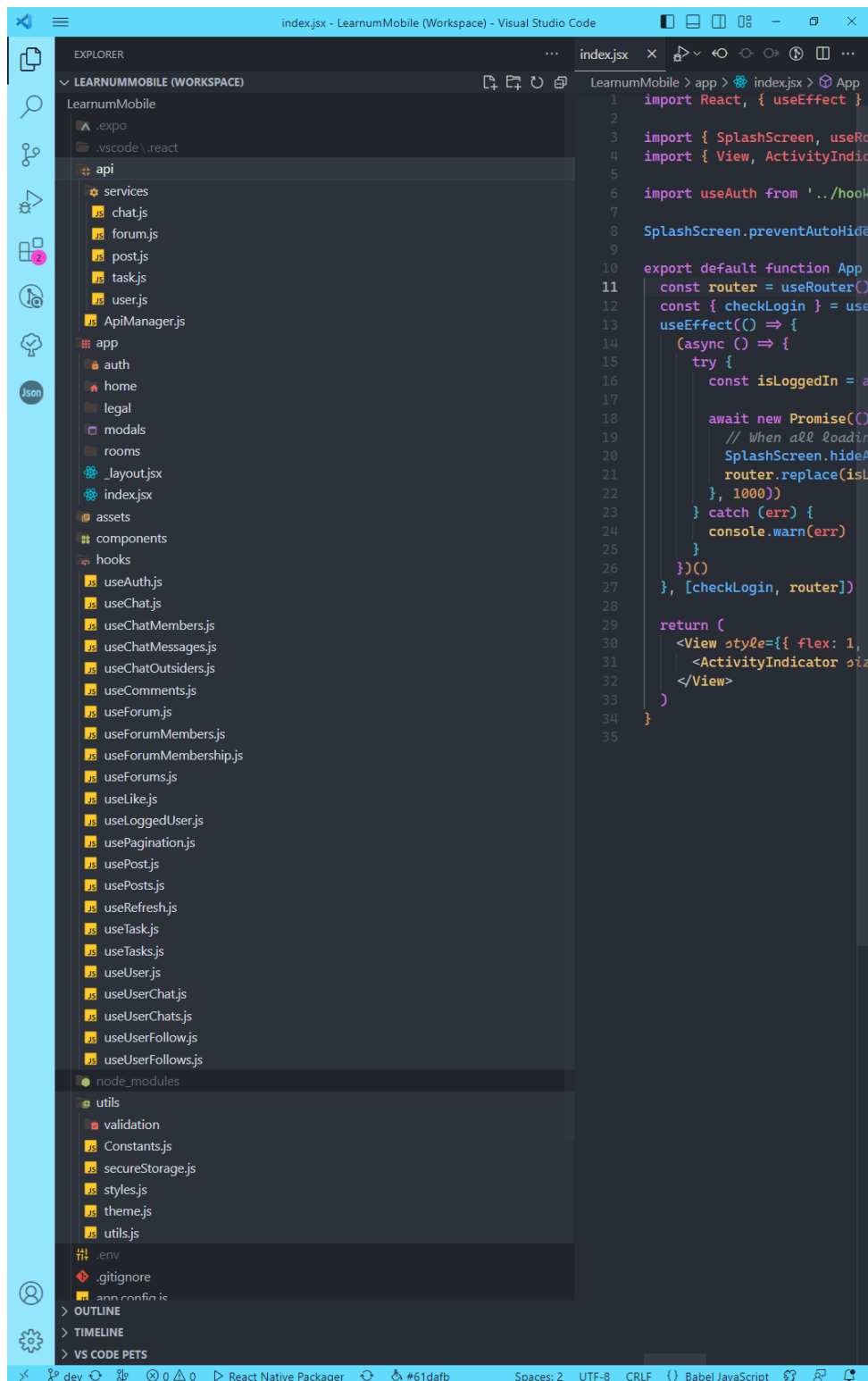


FIGURA A.2: Captura de pantalla de l'estructura de carpetes utilitzada per l'aplicació de la vista per dispositius mòbils del projecte *Learnum*

A.4 Models de dades de l'API definits amb Sequelize

A.4.1 Mòdul d'Usuaris

User

```
1 const UserModel = (sequelize, types) => {
2   const User = sequelize.define('user', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     username: {
10      type: types.STRING,
11      allowNull: false
12    },
13    name: types.STRING,
14    surnames: types.STRING,
15    email: {
16      type: types.STRING,
17      allowNull: false,
18      unique: true
19    },
20    password: {
21      type: types.STRING,
22      allowNull: false
23    },
24    phone: types.STRING,
25    birthdate: types.INTEGER,
26    avatar_icon: types.STRING,
27    n_following: {
28      type: types.INTEGER,
29      defaultValue: 0
30    },
31    n_followers: {
32      type: types.INTEGER,
33      defaultValue: 0
34    }
35  }, {
36    updatedAt: false,
37    freezeTableName: false,
38    tableName: 'tblusers'
39  })
40
41  return User
42 }
```

UserFollower

```
1 const UserFollowerModel = (sequelize, types) => {
2   const UserFollower = sequelize.define('user_followers', {
3     user_id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       allowNull: false
7     },
8     follower_id: {
9       type: types.INTEGER,
10      primaryKey: true,
11      allowNull: false
12    }
13  }
```

```
13 }, {
14   timestamps: false,
15   freezeTableName: false,
16   tableName: 'tblusersfollowers'
17 })
18
19 return UserFollower
20 }
```

A.4.2 Mòdul de Fòrums

Fòrum

```
1 const ForumModel = (sequelize, types) => {
2   return sequelize.define('forum', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     owner_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13    name: types.STRING,
14    description: types.STRING,
15    n_members: {
16      type: types.INTEGER,
17      allowNull: false,
18      defaultValue: 0
19    }
20  }, {
21    updatedAt: false,
22    freezeTableName: false,
23    tableName: 'tblforums'
24  })
25 }
```

ForumMember

```
1 const ForumMemberModel = (sequelize, types) => {
2   return sequelize.define('forum_member', {
3     forum_id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       allowNull: false
7     },
8     user_id: {
9       type: types.INTEGER,
10      primaryKey: true,
11      allowNull: false
12    },
13    role: types.INTEGER
14  }, {
15    updatedAt: false,
16    freezeTableName: false,
17    tableName: 'tblforums_members'
18  })
19 }
```

ForumPost

```
1 const ForumPostModel = (sequelize, types) => {
2   return sequelize.define('forum_post', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     forum_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13    user_id: {
14      type: types.INTEGER,
15      allowNull: false
16    },
17    title: types.STRING,
18    text_content: types.TEXT('long'),
19    n_likes: {
20      type: types.INTEGER,
21      allowNull: false,
22      defaultValue: 0
23    }
24  }, {
25    updatedAt: false,
26    freezeTableName: false,
27    tableName: 'tblforums_posts'
28  })
29 }
```

PostLike

```
1 const PostLikeModel = (sequelize, types) => {
2   return sequelize.define('post_like', {
3     post_id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       allowNull: false
7     },
8     user_id: {
9       type: types.INTEGER,
10      primaryKey: true,
11      allowNull: false
12    }
13  }, {
14    timestamps: false,
15    freezeTableName: false,
16    tableName: 'tblposts_likes'
17  })
18 }
```

PostMedia

```
1 const PostMediaModel = (sequelize, types) => {
2   return sequelize.define('post_media', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
```

```
9   post_id: {
10     type: types.INTEGER,
11     allowNull: false
12   },
13   url: types.STRING
14 }, {
15   timestamps: false,
16   freezeTableName: false,
17   tableName: 'tblposts_media'
18 })
19 }
```

PostComment

```
1 const PostCommentModel = (sequelize, types) => {
2   return sequelize.define('post_comment', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     post_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13    user_id: {
14      type: types.INTEGER,
15      allowNull: false
16    },
17    text_content: types.TEXT('long'),
18    n_likes: {
19      type: types.INTEGER,
20      allowNull: false,
21      defaultValue: 0
22    }
23  }, {
24    updatedAt: false,
25    freezeTableName: false,
26    tableName: 'tblposts_comments'
27  })
28 }
```

CommentLike

```
1 const CommentLikeModel = (sequelize, types) => {
2   return sequelize.define('comment_like', {
3     comment_id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       allowNull: false
7     },
8     user_id: {
9       type: types.INTEGER,
10      primaryKey: true,
11      allowNull: false
12    }
13  }, {
14    timestamps: false,
15    freezeTableName: false,
16    tableName: 'tblcomments_likes'
17  })
18 }
```



```
18 }
```

A.4.3 Mòdul de Tasques

Task

```
1 const TaskModel = (sequelize, types) => {
2   const Task = sequelize.define('task', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     user_id: { // Owner
10      type: types.INTEGER,
11      primaryKey: true,
12      allowNull: false
13    },
14    name: types.STRING,
15    description: types.TEXT('long'),
16    done: {
17      type: types.BOOLEAN,
18      allowNull: false,
19      defaultValue: false
20    }
21  }, {
22    updatedAt: false,
23    freezeTableName: false,
24    tableName: 'tbltasks'
25  })
26
27  return Task
28 }
```

A.4.4 Mòdul de Xats

Chat

```
1 const ChatModel = (sequelize, types) => {
2   return sequelize.define('chat', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     user_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13    name: types.STRING,
14    type: types.INTEGER
15  }, {
16    updatedAt: false,
17    freezeTableName: false,
18    tableName: 'tblchats'
19  })
20 }
```

ChatMember

```
1 const ChatMemberModel = (sequelize, types) => {
2   return sequelize.define('chat_member', {
3     user_id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       allowNull: false
7     },
8     chat_id: {
9       type: types.INTEGER,
10      primaryKey: true,
11      allowNull: false
12    },
13    is_admin: types.BOOLEAN
14  }, {
15    updatedAt: false,
16    freezeTableName: false,
17    tableName: 'tblchats_members'
18  })
19 }
```

Message

```
1 const MessageModel = (sequelize, types) => {
2   return sequelize.define('message', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     user_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13    chat_id: {
14      type: types.INTEGER,
15      allowNull: false
16    },
17    message_id: types.INTEGER,
18    text_msg: types.TEXT('long')
19  }, {
20    updatedAt: false,
21    freezeTableName: false,
22    tableName: 'tblmessages'
23  })
24 }
```

MessageMedia

```
1 const MessageMediaModel = (sequelize, types) => {
2   return sequelize.define('message_media', {
3     id: {
4       type: types.INTEGER,
5       primaryKey: true,
6       autoIncrement: true,
7       allowNull: false
8     },
9     msg_id: {
10      type: types.INTEGER,
11      allowNull: false
12    },
13  }, {
14    updatedAt: false,
15    freezeTableName: false,
16    tableName: 'tblmessage_media'
17  })
18 }
```

```
13     url: types.STRING
14   }, {
15     timestamps: false,
16     freezeTableName: false,
17     tableName: 'tblmessages_media'
18   })
19 }
```

A.5 Documentació apartat legal

En aquesta secció es presenten els documents relatius a la legislació per l'aplicació Learnum. La redacció d'aquests documents ha estat feta en anglès per l'aplicació.

A.5.1 *Avís Legal (Legal Notice)*

Learnum S.A., provided with NIF/CIF 41656259Q, cannot assume any responsibility derived from the incorrect, inappropriate or illicit use of the information appearing on Learnum's application.

Within the limits established by law, Learnum S.A. assumes no liability for the lack of truthfulness, completeness, updating and accuracy of the data or information contained in its platform.

The contents and information do not bind Learnum S.A. nor do they constitute opinions, advice or legal advice of any kind, as it is merely a service offered for informational and informative purposes.

Learnum's platform may contain links to other third party sites that Learnum S.A. cannot control. Therefore, Learnum S.A. cannot assume any responsibility for the content that may appear on third party sites.

The texts, images, sounds, animations, software and other contents included in this website are the exclusive property of Learnum S.A. or its licensors. Any act of transmission, distribution, transfer, reproduction, storage or total or partial public communication must have the express consent of Learnum S.A.

Likewise, to access some of the services that Learnum S.A. offers through the platform, you must provide some personal data. In compliance with the provisions of Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, we inform you that, by filling in these forms, your personal data will be incorporated and processed in Learnum's files in order to be able to provide and offer our services as well as to provide and offer our services. Learnum files in order to provide and offer our services, as well as to inform you of improvements to the application. We also inform you of the possibility of exercising your rights of access, rectification, cancellation and opposition of your personal data, free of charge by email to torresmarcvilagut@gmail.com.

A.5.2 *Política de Privacitat (Privacy Policy)*

1. User information

Learnum S.A., as the Data Controller, informs you that, in accordance with the provisions of Regulation (EU) 2016/679 of 27 April (RGPD) and the L.O. 3/2018 of 5 December on data protection and guarantee of digital rights (LOPDGDD), we will treat your data as we reflect in this Privacy Policy.

In this Privacy Policy we describe how we collect your personal data and why we collect it, what we do with it, who we share it with, how we protect it and your choices regarding the processing of your personal data.

This Policy applies to the processing of your personal data collected by the company for the provision of its services. If you agree to the measures in this Policy, you consent to our processing of your personal data as defined in this Policy.

2. Contact

- **Company name:** Learnum, S.A.
- **Trade name:** Learnum
- **CIF:** 41656259Q
- **E-mail:** torresmarcvilagut@gmail.com

3. Key Principles

We have always been committed to providing our services with the highest degree of quality, which includes treating your data with security and transparency. Our principles are:

- **Legality:** We will only collect your Personal Data for specified, explicit and legitimate purposes.
- **Data Minimization:** We limit the collection of Personal Data to what is strictly relevant and necessary for the purposes for which it was collected.
- **Purpose Limitation:** We will only collect your personal data for the stated purposes and only according to your wishes.
- **Accuracy:** We will keep your personal data accurate and up to date.
- **Data Security:** We implement appropriate technical and organizational measures commensurate with the risks to ensure that your data is not damaged, such as unauthorized disclosure or access, accidental or unlawful destruction or accidental loss or alteration and any other form of unlawful processing.
- **Access and Rectification:** We have the means for you to access or rectify your data whenever you consider it appropriate.
- **Retention:** We retain your personal data in a lawful and appropriate manner and only as long as it is necessary for the purposes for which it was collected.
- **International transfers:** Where your data is to be transferred outside the EU/E-EA it will be adequately protected.
- **Third parties:** Access and transfer of personal data to third parties are carried out in accordance with applicable laws and regulations and with appropriate contractual safeguards.
- **Direct Marketing and Cookies:** We comply with applicable advertising and cookie legislation.

4. Collection and processing of your Personal Data

The types of data that may be requested and processed are:

- Identifying data.

Whenever we request your Personal Data, we will clearly inform you what personal data we collect and for what purpose. In general, we collect and process your Personal Data for the purpose of:

- providing information, services, products, relevant information and news in the sector.
- sending communications.

5. Legitimacy

In accordance with the applicable data protection regulations, your personal data may be processed provided that:

- You have given us your consent for the purposes of processing. Of course, you may withdraw your consent at any time.
- By legal requirement.
- Because there is a legitimate interest that is not undermined by your privacy rights.
- Because it is necessary for the provision of our services through a contractual relationship between you and us.

6. Communication of Personal Data

The data may be communicated to companies related to Learnum, S.A. for the provision of various services as Data Processors. The company will not make any transfer, except by legal obligation.

7. Your Rights

In connection with the collection and processing of your personal data, you can contact us at any time to:

- Access your personal data and any other information indicated in Article 15.1 of the GDPR.
- Rectify your personal data that is inaccurate or incomplete in accordance with Article 16 of the GDPR.
- Delete your personal data in accordance with Article 17 of the GDPR.
- Limit the processing of your personal data in accordance with Article 18 of the GDPR.
- Request the portability of your data in accordance with Article 20 of the GDPR.

- Oppose the processing of your personal data in accordance with Article 21 of the GDPR.
- If you have given your consent for any specific purpose, you have the right to withdraw the consent given at any time, without affecting the lawfulness of the processing based on the consent prior to its withdrawal HR.

You may exercise these rights by sending a reasoned and accredited communication to torresmarcvilagut@gmail.com.

You also have the right to file a complaint with the competent supervisory authority (www.aepd.es) if you believe that the processing does not comply with the regulations in force.

8. Legal Information

The requirements of this Policy supplement, and do not replace, any other requirements existing under applicable data protection law, which shall prevail in any event.

This Policy is subject to periodic review and may be amended by the Company at any time. When this occurs, we will notify you of any changes and ask you to re-read the most recent version of our Policy and confirm your acceptance.

A.5.3 Termes i condicions d'ús (*Terms and Conditions of Use*)

1. These Terms and Conditions of Use regulate the rules to which the use of the Learnum platform (hereinafter referred to as the platform or application) is subject. The use of the platform confers the status of User to whoever does so and implies acceptance of all conditions included in this document and in the Privacy Policy and Legal Notice of the platform. The User should read these conditions each time he/she uses the platform, as they may be modified in the future.
2. Anonymous statistics: Learnum S.A. reserves the right to track your activity on the Learnum platform and to report it to our third-party statistical service providers. All of this is done anonymously.
3. Protection of your personal information: Please refer to Learnum S.A.'s Privacy Policy and the platform's privacy notices to learn about the types of information we collect and the measures we take to protect your personal information.
4. It is forbidden to alter or modify any part of the platform or the contents thereof, circumvent, disable or otherwise manipulate (or attempt to circumvent, disable or manipulate) the security features or other functions of the program. It is forbidden to use the platform with the aim of damaging property, rights or interests of Learnum S.A. or third parties. Furthermore, it is also forbidden to make any other use that alters, damages or renders useless the networks, servers, equipment, products and software of Learnum S.A. or third parties.

5. the platform and its contents (texts, photographs, images, software, links, contents, graphic design, source code, etc.), as well as trademarks and other distinctive signs, are the property of Learnum S.A. or third parties, the User not acquiring any right over them by the mere use of the platform. The User shall refrain from:
 - (a) Reproduce, copy, distribute, make available to third parties, publicly communicate, transform or modify the platform or its contents, except in cases contemplated by law or expressly authorized by Learnum S.A. or by the holder of such rights.
 - (b) Extract or reuse all or a substantial part of the contents of the platform.
6. Subject to the conditions set forth in the preceding paragraph, Learnum S.A. grants the User a non-exclusive, free license to use the platform for personal use, limited to the national territory and for an indefinite period of time. This license is also granted under the same terms with respect to updates and improvements made to the application. These licenses may be revoked by Learnum S.A. unilaterally at any time by simply notifying the User.
7. It is up to the User, in any case, to have adequate tools for the detection and disinfection of malicious programs or any other harmful computer element. Learnum S.A. is not responsible for any damage caused to computer equipment during the use of the platform. Likewise, Learnum S.A. shall not be liable for damages caused to Users when such damages are caused by failures or disconnections in telecommunications' networks that interrupt the service.
8. **IMPORTANT:** We may, without any obligation to you, modify these Terms of Use without notice at any time. Your continued use of the App following any modification to these Terms of Use will constitute your acceptance of such modifications. If you do not accept these Terms of Use or agree to be bound by them, you must not use the App or download or use any related software. Your use of the App is at your sole risk. We have no responsibility or liability for the deletion or failure to store or transmit any content or other information maintained or transmitted by the App. We are not responsible for the accuracy or reliability of any information or advice transmitted through the App. Furthermore, we may, at any time, limit or discontinue your use at our sole discretion. To the maximum extent permitted by law, in no event will we be liable for any related loss or damage.
9. The User agrees to make proper use of the platform, in accordance with the Law, with these Terms and Conditions of Use and with other regulations and instructions that, where appropriate, may be applicable. The User shall be liable to Learnum S.A. and third parties for any damages that may be caused by breach of these obligations.
10. These Terms and Conditions of Use are governed entirely by Spanish law. For the resolution of any dispute relating to their interpretation or application, the User expressly submits to the jurisdiction of the courts of Girona (Spain).