



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

**APLICACIÓ MÒBIL PEL SEGUIMENT DE
PROJECTES D'APRENENTATGE SERVEI**

Gerard Rouras Jubany

Directors: Eloi Puertas i Prats, Anna Maria Escofet Roig
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 24 de gener de 2021

Agraïments

Vull agrair tant a Eloi Puertas i Prats com a Anna Maria Escofet Roig per tota l'ajuda, orientació i recomanacions que m'han fet durant el desenvolupament de l'aplicació, també a Laura Rubio Serrano per la seva orientació en les dues sessions de seguiment que va poder assistir.

A la secció d'Aprenentatge Servei de la facultat d'educació, que ha anat provant l'aplicació ajudant a millorar-la i que ha dissenyat el logotip final de l'aplicació.

A totes les persones que han provat l'aplicació tant en la plataforma Android com IOS, que m'ha permès poder trobar errors i millorar els aspectes d'usabilitat i de disseny.

Finalment, agrair a la meva família, amics i companys que m'han recolzat i ajudat per poder fer possible aquest projecte.

Resum

Aquest projecte neix de la necessitat de la secció d'Aprenentatge Servei de la facultat d'educació. La creació d'una aplicació mòbil que permeti fer un seguiment més directe, eficaç, comunicatiu i intuïtiu dels diferents projectes i treballs que es desenvolupen durant el curs, donant una solució a la inconnexió que es produeix a vegades entre entitats, professors i alumnes, i així reduir els passos necessaris per una comunicació més directa entre les diferents parts.

L'aplicació mòbil està dissenyada per complir totes aquelles necessitats amb una corba d'aprenentatge baixa, això s'aconsegueix fent que la usabilitat amb la interfície de l'aplicació sigui similar a la d'altres aplicacions que els usuaris fan servir en el dia a dia, però afegint les funcionalitats necessàries. A més a més, està pensada per a que estigui disponible per al major nombre de dispositius mòbils, per tant, en les plataformes Android i IOS.

Resumen

Este proyecto nace de las necesidades por parte de la sección de Aprenentatge Servei de la facultad de educación. La creación de una aplicación móvil que permita hacer un seguimiento más directo, eficaz, comunicativo e intuitivo de los diferentes proyectos y trabajos que se desarrollan durante el curso, dando una solución a la inconnexión que se produce a veces entre entidades, profesores y alumnos, así reduciendo los pasos necesarios para una comunicación más directa entre las diferentes partes.

La aplicación móvil está diseñada para cumplir todas aquellas necesidades con una curva de aprendizaje baja, esto se consigue haciendo que la usabilidad con la interfaz de la aplicación sea similar en gran medida a otras aplicaciones que lo usuarios puedan utilizar en el día a día, pero añadiendo las funcionalidades necesarias. Además, está pensada de forma que esté disponible para el mayor número de dispositivos móviles, por tanto, en las plataformas Android e IOS.

Abstract

This project is born to meet the needs of Aprenentatge Servei from the education department. Creating a mobile application that allows making a faster, solid, communicative, and intuitive tracing of the different projects and works that are being developed during the academic year, not only solving the recurrent disconnection between the entities, professors, and students but also reducing the necessary steps to make more direct communication between the different parts.

The application is designed to fulfill all the requirements using an easier learning curve thanks to a common application interface usability and adding the necessary functionalities. Moreover, it has been created to be available for the largest number of devices like these that use Android and IOS.

ÍNDEX

1.	INTRODUCCIÓ	5
1.1	CONTEXT I MOTIVACIÓ	5
1.2	OBJECTIUS	5
1.3	ESTRUCTURA	6
2.	TREBALLS PREVIS.....	7
2.1	RECERCA DE TECNOLOGIES	7
3.	ANÀLISI.....	9
3.1	TIPUS DE REQUERIMENTS	9
3.2	ROLS D'USUARI	10
3.3	USER STORIES	11
3.4	PLANIFICACIÓ.....	16
3.5	COST DEL PROJECTE	20
4.	DISSENY	23
4.1	ARQUITECTURA I MODEL DE DADES	23
4.2	ESTRUCTURA DE DADES	23
4.3	MODEL RELACIONAL.....	27
4.4	PATRONS DE DISSENY	29
4.5	DIAGRAMES DE COMPONENTS	29
5.	IMPLEMENTACIÓ.....	32
5.1	ALGORITMES.....	33
6.	RESULTATS	35
6.1	TESTING.....	35
6.2	INTERFÍCIE D'USUARI	36
7.	CONCLUSIÓ.....	41
7.1	TREBALLS FUTURS.....	41
8.	BIBLIOGRAFIA	42
9.	ANNEX.....	43
9.1	GLOSSARI.....	43
9.2	MANUAL D'ÚS	44
9.2.1	INSTAL·LACIÓ.....	44
9.2.2	GUIA GENERAL.....	44

1. INTRODUCCIÓ

1.1 Context i Motivació

La secció d'aprenentatge servei és una proposta de caràcter docent i de recerca que integra tant l'aprenentatge que s'adquireix en l'entorn acadèmic, com el de servei per la comunitat en un sol projecte, que gràcies al seu desenvolupament, permet als estudiants formar-se treballant sobre necessitats reals.

Ens trobem en una era on gairebé tot es troba digitalitzat i cada cop és més important adaptar les necessitats del dia a dia a un entorn on els usuaris puguin accedir d'una manera més ràpida a les seves tasques, treballs i projectes que duen a terme en llocs físics com el seu treball o estudis.

D'aquí sorgeix la idea d'una aplicació capaç d'adaptar les noves necessitats digitals amb les de l'entorn d'aprenentatge servei aconseguint centralitzar en un únic lloc el seguiment dels projectes que es desenvolupen durant un curs amb la interacció entre els usuaris participants, d'una manera poc invasiva i que es trobi disponible en el màxim nombre de situacions.

1.2 Objectius

L'objectiu principal des d'un primer moment ha estat poder tenir una aplicació mòbil multi-plataforma (Android i IOS) que fos capaç d'encapsular el seguiment del desenvolupament dels projectes amb la comunicació entre els participants, com s'ha esmentat a la introducció.

En poques paraules, els usuaris han de tenir la possibilitat de comunicar-se amb altres participants d'un mateix projecte mitjançant un sistema de missatgeria, per tant l'estructura s'ha de basar en una sala de xat que alhora tingui les funcionalitats necessàries per permetre fer el seguiment educatiu.

L'alumne ha de tenir la possibilitat de poder notificar al professor i a l'entitat que ha assistit al lloc requerit en un cert horari per realitzar el seu treball o projecte. Això ho farà mitjançant una actualització o post, que consisteix en la introducció d'una explicació, un àudio i una imatge en una pàgina enllaçada a la sala de xat, on a més a més es podran visualitzar totes les actualitzacions creades de manera que es pugui fer el seguiment corresponent. L'horari està marcat amb una tasca mitjançant un calendari que podran veure tots els participants i aquesta, pot ser creada tant pel professor o com per l'entitat. A més a més, per cada tasca o actualització creada s'ha de veure un missatge d'avís a la sala de xat d'aquell grup de treball.

A mesura que el projecte es va anar desenvolupant es van anar afegint nous objectius i es va anar modificant els anteriors fins a completar la llista. Entre els més importants es troben els següents:

- Xats més generals on es trobin inclosos tots els alumnes que tinguin un projecte amb el mateix professor i entitat (incloent els dos en aquest xat), permetent una comunicació d'aspectes més generals. Dins d'aquests, existeix la possibilitat d'anar a un xat més reduït on només es trobi l'alumne, professor i entitat participants d'un treball o projecte específic, és a dir, la sala que hi havia com objectiu inicial.
- Xats individuals, permetent així als professors i entitats comunicar-se de manera més específica i individual amb altres usuaris, incloent els alumnes.
- Tots els arxius enviats (imatges i àudios) en el xats on l'usuari n'és participant es puguin visualitzar en una pàgina apart, d'una manera més ordenada i més clara.
- Per poder crear una comunicació més completa en el xats ha d'haver la possibilitat d'afegir imatges i àudios, a més a més de poder compartir els arxius rebuts a qualsevol xarxa social, e-mail o núvol.
- Un sistema de notificacions.

- Oferir la possibilitat de canviar l'estil de colors de l'aplicació per donar més llibertat als usuaris.

1.3 Estructura

Els següents apartats que es mostraran a la memòria són aquests:

2. Treballs previs: S'exposen els treballs i recerques fetes abans de començar amb el desenvolupament del projecte
3. Anàlisi: En aquest apartat s'exposen i s'expliquen els diferents requeriments demanats per el desenvolupament del projecte, els rols d'usuari, la planificació mitjançant una sèrie de US a partir d'aquests requeriments i el cost del projecte.
4. Disseny: Es mostra el disseny, els patrons i l'estructura que s'ha seguit principalment en l'àmbit de les dades
5. Implementació: S'analitzen els algorismes utilitzats i la implementació de la vista.
6. Resultats: Es mostren els resultats obtinguts tant del testing com de l'aplicació final un cop s'ha finalitzat la implementació.
7. Conclusió: S'expliquen les conclusions i el treball futur del projecte.
8. Bibliografia: Es mostren les diferents fonts consultades per el desenvolupament del projecte.
9. Annex: S'exposa el glossari de la memòria i el manual d'ús de l'aplicació.

2. TREBALLS PREVIS

En aquest projecte, com anteriorment s'ha exposat, s'ha de construir una aplicació mòbil de missatgeria i educativa per a les plataformes IOS i Android.

L'estructura que volem fer es divideix en dues parts, una secció de xats amb el seu respectiu entorn i un altre de funcionalitats que es poden trobar a qualsevol pàgina educativa com per exemple, la creació de tasques en un calendari o la tramesa d'elements a qualificar. Per tant, s'havia de trobar un recurs per poder encapsular les dues parts de la manera més homogènia possible, per tal de que la fluïdesa i la usabilitat no es trobessin afectades.

Al no existir una aplicació similar s'havia de dissenyar una de manera personalitzada, però sense allunyar-se del disseny d'aplicacions de tipus de missatgeria com poden ser Telegram i WhatsApp, i dels de tipus d'educació com pot ser la pàgina Moodle. Per tant, es va pensar que el disseny prioritari de la nostra aplicació fos d'un estil similar al d'aplicacions de missatgeria com les abans esmentades, i a partir d'aquest afegir les funcionalitats educatives de manera no invasiva.

Prioritzar aquest estil dona bastants més possibilitats alhora d'afegir funcionalitats, com per exemple tenir xats que alhora siguin un grup de treball, permetent fer trameses relacionades amb l'educació mitjançant una pàgina enllaçada, fent que el número de passos per fer aquesta acció fos similar al d'afegir qualsevol fitxer en un xat de qualsevol altre aplicació.

2.1 Recerca de tecnologies

Inicialment es va fer una recerca per trobar la manera d'exportar l'aplicació programada per la plataforma d'Android a IOS, però podrien sorgir bastants problemes ja que la implementació nativa a IOS requereix de d'interfície XCode, un entorn bastant diferent al que requereix Android (Android Studio), a més a més la traducció havia de ser manual, el que provocaria fer alhora dues aplicacions per una.

Per tant, l'altre solució era programar una aplicació que mitjançant un entorn de desenvolupament d'aplicacions amb un llenguatge de programació en específic fos compatible per ambdues plataformes un cop compilada. De manera que vaig extreure les opcions més comunes: Flutter, React Native i Ionic.

Flutter (2017), és un framework de codi obert creat per Google que utilitza el llenguatge de programació anomenat Dart (creat específicament per substituir Javascript) que s'empra per treballar des del client, per tant sense la necessitat d'utilitzar un framework de suport extern. Té continguda tant la vista com la lògica en un mateix fitxer Dart, tractant-se doncs, d'un framework bastant encapsulat on cada pàgina o component és independent.

El framework ens dona la possibilitat de córrer l'aplicació en les plataformes Android, IOS i Web.



En el cas d'Android és necessari tenir instal·lat al nostre ordinador Android Studio. Això ens permet la possibilitat d'execució directe des de la consola o interfície, sempre i quan hi hagi un emulador o dispositiu disponible, utilitzant la comanda *flutter run*.

Pel que fa a l'IOS és necessari XCode, aquest programa està només disponible per als ordinadors que tenen com a sistema operatiu MacOS i com en el cas anterior també ens permet la possibilitat d'execució mitjançant consola o interfície, sempre i quan hi hagi un emulador o dispositiu disponible, utilitzant la comanda *flutter run*.

Respecte a Web és necessari que s'hagi habilitat el suport web, mitjançant la comanda *flutter config --enable-web*, i posteriorment utilitzar la comanda *flutter run* seleccionant el navegador desitjat.



React Native (2015), és un framework de codi obert creat per Facebook que utilitza el llenguatge de programació Javascript i Npm associat a Node, és a dir, es tracta d'un framework que utilitza llenguatges bastant comuns de l'àmbit web, encara que intenta no allunyar-se d'un rendiment proper al d'una implementació nativa. Utilitza un sistema de components que marquen les diferents pàgines creades, aquestes, de manera similar a Flutter contenen tant la vista com la lògica en el mateix fitxer Javascript.

El framework ens permet córrer l'aplicació en Android i IOS, en el cas d'emulació Web és necessària la instal·lació de paquets externs a l'aplicació. Per poder emular en dispositius mòbils es necessita Android Studio i XCode, a més a més en ambdós casos tenim la possibilitat d'execució via consola mitjançant la comanda `react-native run-android --simulator="device"` i `react-native run-ios --simulator="device"` sempre i quan hi hagi un emulador o dispositiu disponible.

Ionic (2013) és un framework de codi obert que permet desenvolupar aplicacions híbrides utilitzant Html5, Css, Javascript i Cordova, a més a més de donar la possibilitat d'escollir entre react.js, angular.js o vue.js per gestionar les aplicacions. Cordova permet utilitzar les tecnologies abans esmentades, evitant el desenvolupament natiu de cada plataforma mòbil individualment, encara que brinda l'oportunitat d'accés a components nadius dels dispositius mòbils que poden ser necessaris per a la implementació de certes funcionalitats de l'aplicació.



Ionic utilitza Npm associat a node, per la vista fa servir Html5 i Css, i per la lògica utilitza Typescript per a la implementació d'angular.js, react.js o vue.js. Per tant aplica un desenvolupament semblant a aplicacions web però adaptada a aplicacions mòbil.

- Vue.js, Angular.js o React.js tenen com objectiu facilitar la construcció de les interfícies, les tres estan preparades per treballar amb components.

El framework ens dona la possibilitat de córrer l'aplicació en les plataformes Android, IOS i Web. Per córrer tant en Android i IOS és necessari, com en el casos anteriors, de la instal·lació d'Android Studio i XCode. Per tant, tenim la possibilitat d'execució mitjançant una comanda de consola, `ionic cordova run android` o `ionic cordova run ios`, en cas de que vulguem un dispositiu extern s'ha d'afegir al final – `device`. En el l'execució en Web és tant senzill com utilitzar la comanda `ionic serve` sense la necessitat de Cordova.

Un cop trobades les possibilitats d'implementació de la nostra aplicació utilitzant els diferents frameworks, era necessari trobar la manera de poder provar aquestes aplicacions en un entorn real.

- **Android:** mitjançant Android Studio tenim la possibilitat de crear i signar una aplicació per poder distribuir-la gratuïtament.
- **IOS:** mitjançant XCode podem crear i signar una aplicació, però a diferència d'Android Studio, és necessària una llicència de pagament per la distribució d'aquesta, per tant impossibilitant el testeig en un gran nombre de dispositius de manera gratuïta i provocant així un contratemps en el nostre desenvolupament a gran escala absolutament fiable per aquesta plataforma.

Entre totes les tecnologies cercades prèviament es va escollir el framework Ionic. Inicialment es va plantejar acompanyar-lo amb react.js però finalment l'opció escollida va ser angular.js ja que en el moment d'inici de desenvolupament es tenia una documentació més completa merament pel fet de ser bastant més antic, a més a més Ionic estava plantejat inicialment per ser acompanyat per aquest, el que fa que la resta de les implementacions estiguessin traduïdes a partir d'aquest.

3. ANÀLISI

Per poder desenvolupar una aplicació és necessari una llista de requeriments creats per el client, aquests varien depenent dels rols de cada usuari, ja que cadascú té unes funcionalitats i accions permeses. Es van establir en la primera reunió feta el 09/10/2020, encara que es van anar augmentant a mesura que es feien les reunions.

3.1 Tipus de requeriments

Requeriments d'usuaris

Se'ns va demanar les funcionalitats d'inici de sessió i de registre, tot això utilitzant un correu electrònic personal, a més a més de donar la possibilitat de reiniciar la contrasenya de l'usuari mitjançant l'enviament d'un correu electrònic de confirmació per tal d'evitar haver d'eliminar el compte si l'usuari no aconseguia recordar-la, ja que no pot haver cap accés de la contrasenya per temes de seguretat. Un altre funcionalitat va ser donar permís a l'usuari per modificar les seves dades inicialment introduïdes, com poden ser el nom i cognom, contrasenya, correu electrònic i imatge de perfil, permetent així una millor flexibilitat en aquest aspecte. Un cop l'usuari creés el compte, donar la possibilitat d'escollir la facultat en la que l'usuari és integrant. Per acabar era necessari que hi hagués l'opció de tancar una sessió activa.

Requeriments de missatgeria

La necessitat d'un servei de missatgeria entre persones o entre grups, tot això incloent imatges i àudios. Per implementar aquest servei era necessari la creació de sales de xats, donant la possibilitat de separar-les depenent del tipus.

- Xats individuals: creació i eliminació de xats de missatgeria entre dos usuaris.
- Xats de projectes: creació i eliminació de xats de missatgeria entre participants d'un projecte. Amb possibilitat de modificació de la imatge de perfil.
- Xats de grups de treballs: creació i eliminació de xats de missatgeria reduïda entre participants d'un grup de treball.

En aquest servei a més d'enviar missatges, s'havia de donar el poder a un usuari d'eliminar els seus missatges, encara que fos una imatge o àudio i la possibilitat d'eliminar tots els missatges del xat que es volgués, amb l'objectiu de millorar el rendiment d'aquests o per al seguiment del calendari acadèmic.

Requeriments en l'apartat educatiu

Implementar una sèrie de funcionalitats per fer el seguiment dels projectes que els alumnes van desenvolupant durant el curs. Per tant, com s'ha explicat en els treballs previs, s'havia d'integrar el servei de missatgeria amb el d'educació de manera que les dues coses formessin part d'un mateix entorn, és a dir, que un grup de treball fos un xat amb un apartat centrat en l'educació. Aquests requeriments són:

- Fer actualitzacions mitjançant la creació d'un post, afegint un text, imatge i àudio.
- Una pàgina on es poguessin veure totes les actualitzacions creades en un grup de treball.
- La creació d'una pàgina que contingues un calendari enllaçat a un grup de treball.
- Crear tasques al calendari enllaçant a un grup de treball.

Requeriments en l'administració de fitxers:

Implementar una sèrie de funcionalitats relacionades amb la visualització i administració de fitxers, entre les quals s'havia de permetre la visualització d'una imatge o la reproducció d'un àudio a la pàgina on es trobessin, a més a més donar la possibilitat mitjançant la creació d'una pàgina apart de visualitzar o reproduir tots els fitxers enviats i rebuts en qualsevol xat en que l'usuari es trobés inclòs. Era important com a requeriments no funcionals, que els fitxers que s'eliminassin de l'aplicació també ho fessin a la base de dades, com una bona pràctica i per reforçar la seguretat. També era necessari poder compartir els fitxers que volguéssim a qualsevol xarxa social, sistema de núvol o correu electrònic.

Requeriments de la usabilitat de l'aplicació:

Per la usabilitat de l'aplicació se'ns va demanar poder cercar paraules en la majoria de pàgines de l'aplicació per simplificar la localització d'algun element com pot ser un missatge en un xat, un usuari en la llista, un grup, un projecte o un xat individual. A més a més era necessari la implementació d'un canvi d'estil mitjançant el canvi de colors de l'aplicació amb la possibilitat per part d'un administrador de crear-ne de nous directament des de l'aplicació.

Com a part important, era necessari la creació d'una pàgina d'ajuda per poder guiar a qualsevol usuari que tingués algun dubte amb algun apartat de l'aplicació o busqués la manera de fer alguna acció específica, amb la possibilitat de modificar-la de manera externa i no estàtica.

La creació d'un sistema de notificacions, per poder rebre els missatges de forma indirecta dels seus corresponents xats, projectes o grups de treball, incloses les noves tasques creades al calendari i les actualitzacions d'aquest últim.

Requeriments d'administració de l'aplicació

A part dels requeriments d'administració inclosos en missatgeria, era necessari oferir la possibilitat de canviar el rol o facultat d'un usuari, això es necessari ja que es pot donar el cas que un usuari esculli equivocadament la seva facultat, o per assignar el seu rol inicial a un usuari.

Per part dels usuaris es va donar l'opció de silenciar notificacions de manera individual de cada xat o de manera global, a més a més de poder veure els usuaris de la facultat o dels grups de treball que es trobin inclosos.

Finalment donar la possibilitat als usuaris de veure la llista d'usuaris de la facultat o del grup de treball.

3.2 Rols d'usuari

- Súper-administrador: Només pot ser assignat des de la base de dades per seguretat, té el deure d'administrar l'aplicació en les diferents facultats.
- Administrador: És assignat pel súper administrador, té el deure d'administrar l'aplicació en la facultat on es troba.
- Professor: És assignat pel súper-administrador o administrador, pensat per poder administrar i fer el seguiment dels seus grups de treballs del món real.
- Entitat: És assignat pel súper-administrador o administrador, pensat per poder fer el seguiment dels seus grups de treballs del món real.
- Alumne: S'assigna automàticament al crear un compte, pensat per desenvolupar els diferents treballs assignats del món real.

- Usuari sense sessió iniciada: Qualsevol usuari que encara no ha iniciat sessió a l'aplicació.

3.3 User Stories

L'objectiu de les User Stories és poder organitzar de manera clara els requeriments juntament amb els rols d'usuari de l'aplicació per poder utilitzar-los posteriorment durant el desenvolupament mitjançant el procediment explicat a l'apartat 3.4.

Trobem dos tipus de User Stories, aquestes poden ser èpiques o no. Les èpiques són històries que requereixen una gran quantitat de treball i que al mateix temps comporten l'execució de sub-històries més petites. A més a més és necessari un criteri d'acceptació per cadascuna.

La numeració ve donada per l'ordre en què es va fer la selecció dels requeriments durant el desenvolupament.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Usuari sense sessió iniciada	Iniciar sessió a l'aplicació	Poder entrar al meu compte	Sí	Inici de sessió utilitzant una sèrie de credencials per tenir accés a la interfície pròpia de l'usuari	US01
Usuari sense sessió iniciada	Creació d'un compte	Poder crear un compte nou	Si	Creació de compte utilitzant un correu electrònic personal i una contrasenya secreta.	US02
Usuari amb sessió iniciada	Tancar sessió	Poder tancar una sessió activa	Si	L'usuari ha de tenir una sessió activa.	US03
Usuari sense sessió iniciada	Recuperar contrasenya	Poder recuperar la contrasenya del meu compte en cas d'oblidar-me	Si	L'usuari ha de tenir un compte existent i permetre el canvi de contrasenya mitjançant un correu enviat.	US04
Usuari amb sessió iniciada	Seleccionar la meua facultat	Estar en la interfície pròpia de la facultat	No	L'usuari ha d'iniciar sessió per primer cop	US36
Usuari amb sessió iniciada	Canviar les meves dades de perfil	Actualitzar les meves dades o per algun altre motiu	No	Si l'usuari vol canviar el correu o la contrasenya, és necessari que introdueixi la contrasenya actual abans, a més a més ha de confirmar-ho mitjançant correu enviat	US05

Figura 3.3.1: User Stories del bloc de requeriments d'usuaris. Font: Pròpia

Les User Stories anteriors són extrems del bloc de requeriments d'usuari. Les històries com a usuari sense sessió iniciada només són necessàries un cop s'inicia l'aplicació per primer cop o si l'usuari ha decidit tancar la sessió actual.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Administrador o Súper Administrador	Crear un projecte	Crear un projecte que sigui necessari per la secció d' ApS	No	L'usuari ha de crear el projecte mitjançant la introducció d'un títol.	US06
Administrador o Súper Administrador	Eliminar un projecte	Eliminar un projecte que ja no sigui necessari.	Si	Ha d'existir el projecte per procedir a l'eliminació d'aquest i dels grups de treball enllaçats amb els seus fitxers, tant dels xats com dels posts.	US07

Administrador o Súper Administrador	Afegir participants a un projecte	Afegir qualsevol usuari de la facultat que sigui necessari per a aquell projecte	No	L'usuari a afegir s'ha de trobar en la mateixa facultat i que no estigui com a participant en el projecte.	US08
Administrador o Súper Administrador	Eliminar participants d'un projecte	Poder eliminar qualsevol usuari que estigui inclòs al projecte per qualsevol motiu	No	L'usuari ha de poder eliminar un altre usuari del projecte sempre i quan en sigui participant.	US09
Administrador o Súper Administrador	Modificació de la imatge de perfil d'un projecte	Poder canviar quan vulgui la imatge de perfil d'un projecte existent	No	Permetre a l'usuari el canvi de la imatge de perfil d'un projecte sempre i quan existeixi.	US10
Usuaris amb sessió iniciada	Veure els projectes assignats	Poder veure els projectes dels quals en sóc participant per poder entrar	No	L'usuari ha de ser participant d'algun projecte de la mateixa facultat.	US11

Figura 3.3.2: User Stories del bloc de missatgeria de la secció de projectes. Font: Pròpia

En les User Stories anteriors tenim continguts el requeriments de l'apartat de projectes en el bloc de missatgeria. Veiem que totes les accions que tenen a veure amb l'administració de projectes estan delegades per als rols d'administradors i súper administradors. Això es degut a que és necessari tenir un control d'aquests projectes per tal que s'afegeixin i s'eliminin en el moment que existeixi la necessitat per part de la secció d'Aprenentatge Servei, que sol ser a inici i final del curs, fent així que el professorat no hagi de fer tot el treball. A més a més, s'havia d'implementar una interfície per visualitzar tots els projectes assignats.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Administrador, Súper Administrador o Professor	Crear un grup de treball	Poder crear un grup per fer treballs de manera més reduïda dins d'un projecte ja existent.	No	L'usuari ha de poder crear un grup de treball amb la introducció d'un nom determinat i dins d'un projecte creat amb anterioritat.	US30
Administrador, Súper Administrador o Professor	Eliminar un grup de treball	Poder eliminar un grup de treball ja existent en cas de que ja no sigui necessari continuar amb el desenvolupament	Sí	Ha d'existir el grup de treball per procedir a l'eliminació d'aquest i dels fitxers enllaçats, tant del xat com dels posts.	US31
Administrador, Súper Administrador o Professor	Afegir participants a un grup de treball	Poder afegir un usuari participant d'un projecte que sigui necessari en aquell grup de treball	No	Per afegir un usuari és necessari que aquest es trobi inclòs al projecte però que no estigui ja al grup de treball.	US32
Administrador, Súper Administrador o Professor	Eliminar participants d'un grup de treball	Poder eliminar un usuari participant d'un grup de treball per qualsevol motiu	No	Per eliminar un usuari d'un grup de treball aquest n'ha de ser participant.	US33
Usuaris amb sessió iniciada	Veure els grups de treballs assignats	Poder veure els grups de treballs als quals n'estic assignat per poder entrar	No	L'usuari ha de ser participant d'aquell grup de treball i del projecte pare.	US34

Figura 3.3.3: User Stories del bloc de missatgeria de la secció de grups de treballs. Font: Pròpia

En aquestes User Stories s'afegeix el rol de professor, donant-li la capacitat d'administrar els grups de treball d'una manera similar a la que es faria en una classe, encara que, sempre amb la possibilitat d'un administrador

de fer aquesta tasca. La visualització dels grups està pensada per fer-se mitjançant una pàgina enllaçada al projecte per tal de tenir-los tots organitzats.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Administrador, Súper Administrador, Professor o Entitat	Creació d'un xat individual	Comunicar-me de manera directa amb altres usuaris de la facultat	No	L'usuari ha de poder crear un xat amb un altre usuari si aquest està a la mateixa facultat. És necessari també enviar un primer missatge per a que es faci la creació del xat.	US14
Administrador, Súper Administrador, Professor o Entitat	Eliminació d'un xat individual	Poder eliminar un xat amb un altre usuari on ja no vulgui seguir amb la comunicació	Sí	Ha d'existir el xat individual per procedir a l'eliminació d'aquest i dels fitxers enllaçats.	US15
Usuaris amb sessió iniciada	Veure els xats individuals propis	Poder veure els xats individuals que tinc actius per poder entrar	No	L'usuari ha tenir alguna conversa del tipus usuari-usuari per poder veure els xats.	US16

Figura 3.3.4: User Stories del bloc de missatgeria de la secció de xats individuals. Font: Pròpia

Trobem les User Stories dels xats individuals del bloc de missatgeria, com es veu el rol d'alumne no té cap poder en l'administració d'aquests. Això està pensat per evitar que l'alumne pugui utilitzar l'apartat de xats de manera lliure i no controlada amb els problemes que podria ocasionar. Per tant, ha de ser el professor o entitat qui es posi en contacte amb ell en el moment que desitgin per el motiu que sigui.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Usuari amb sessió iniciada	Enviar missatges en un xat	Poder establir una conversa mitjançant l'ús de text, àudios o imatges en un xat existent	No	Permetre a l'usuari enviar missatges en els xats on es trobi inclòs. Per part dels administradors no fa falta aquest requisits.	US12
Usuari amb sessió iniciada	Eliminar missatges en un xat	Poder eliminar missatges enviats quan ho necessiti per qualsevol motiu.	Sí	Per poder eliminar missatges han d'haver estat enviats pel mateix usuari sense tenir el temps des de l'enviament	US13
Administrador o Súper Administrador	Eliminar tots els missatges d'un projecte/grup	Millorar el rendiment de la pàgina.	Sí	És necessari que hi hagin missatges per poder eliminar-los.	US45

Figura 3.3.5: User Stories del bloc de missatgeria de la secció d'administració d'enviament de missatges. Font: Pròpia

Les últimes User Stories del bloc de missatgeria són de l'apartat d'administració d'enviament de missatges, com es pot veure està pensat per donar llibertat a qualsevol usuari que hagi enviat missatges, evitant així imatges, àudio i missatges de text enviats per equivocació. A més a més, es va pensar delegar l'acció d'eliminar els missatges als administradors per tal de que només fos necessària en cas de que hagués problemes de rendiment per un nombre elevat de missatges en aquell xat.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Alumne	Crear una actualització mitjançant un post	Poder informar que he anat al lloc i fet el treball que se m'ha assignat	Sí	L'usuari ha d'estar com a participant d'un grup de treball, a més a més ha d'afegir una imatge, àudio i text.	US17
Alumne,	Eliminar una actualització creada	Poder eliminar en cas de que hagi creat un post equivocadament.	No	L'usuari ha d'estar com a participant d'un grup de treball i haver creat algun post per procedir a l'eliminació.	US18
Alumne, Professor o Entitat	Veure els posts creats	Poder fer un seguiment dels posts que s'han anat creant durant el desenvolupament.	No	L'usuari ha d'estar com a participant d'un grup de treball, a més a més, l'alumne ha d'haver creat algun post amb anterioritat.	US19
Professor, Entitat o Alumne	Veure el calendari del grup de treball amb les seves tasques.	Poder veure el que hi ha programat en un calendari.	No	L'usuari ha d'estar com a participant d'un grup de treball.	US20
Professor o Entitat	Afegir una tasca al calendari	Informar a l'alumne de les noves tasques que ha de fer	Sí	L'usuari ha d'estar com a participant d'un grup de treball i que l'horari introduït sigui vàlid.	US21
Professor o Entitat	Eliminar una tasca al calendari	Eliminar-la en cas d'equivocació	No	L'usuari ha d'estar com a participant d'un grup de treball i que hi hagi alguna tasca creada	US22

Figura 3.3.6: User Stories del bloc d'educació. Font: Pròpia

En aquestes User Stories donem la responsabilitat als professors i entitats d'afegir les tasques que un alumne hagi de fer en el futur per tal de que aquest envii els seus posts diaris amb la confirmació de que ha fet el seu treball, de manera que es pugui fer un seguiment més directe del que fa l'alumne.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Usuari amb sessió iniciada	Veure una imatge en pantalla completa	Poder veure amb més detall aquella imatge que es troba a la pàgina	No	Hi ha d'haver alguna imatge a la pàgina on es troba l'usuari per, mitjançant un clic, obrir la visualització completa.	US26
Usuari amb sessió iniciada	Reproduir un àudio	Poder escoltar un àudio que es troba a la pàgina	Sí	Hi ha d'haver algun àudio a la pàgina on es troba l'usuari.	US27
Usuari amb sessió iniciada	Compartir un element (Imatge/Àudio/Text)	Enviar l'element desitjat per ús personal o per compartir amb algú	Sí	Hi ha d'haver alguna imatge o àudio a la pàgina que es trobi. En el cas de text ha de ser a la pàgina d'un post creat	US38
Usuari amb sessió iniciada	Veure tots els fitxers enviats o rebuts	Poder tenir-los organitzats per trobar-los amb més facilitat	Sí	Ha d'estar inclòs en alguna sala de xat, projecte o grup de treball	US23

Figura 3.3.7: User Stories del bloc d'administració de fitxers. Font: Pròpia

En les User Stories anteriors podem veure el sistema d'administrador dels fitxers enviats. Qualsevol usuari que tingui a la vista algun fitxer d'àudio o imatge té la possibilitat de reproduir-lo, visualitzar-lo o compartir-lo. Per compartir un text és necessari que sigui en la pàgina d'un post creat per un alumne, es dona aquesta possibilitat per a que un professor o entitat pugui traspasar aquest element per avaluar.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Usuari amb sessió iniciada	Cercar un text a la pàgina	Simplificar la cerca d'elements d'alguna pàgina on hi pugui haver multitud.	No	La pàgina en qüestió ha de tenir la possibilitat de fer aquesta cerca, a més a més en aquest cas ha d'haver missatges, projectes, usuaris o grups de treball en la llista de la pàgina	US41
Usuari amb sessió iniciada	Veure la pàgina d'ajuda	Buscar resposta a alguna cosa que no se'n pot trobar	No		US28
Súper-Administrador	Modificar la pàgina d'ajuda	Enriquir-la amb més informació o millorar-la	No	El súper administrador ha de tenir les credencials del GitHub on es troba la pàgina per poder modificar-la de manera manual, ja que es troba fora de la pàgina	US29
Usuari amb sessió iniciada	Modificar l'estil de l'aplicació	Canviar l'estil actual pel motiu que sigui.	No	Ha d'haver estils creats amb anterioritat per poder seleccionar-ne un nou	US39
Súper-Administrador	Afegir nous estils a l'aplicació	Donar més possibilitats als usuaris de l'aplicació	No	L'usuari ha de seleccionar tots els colors dels elements de l'aplicació que es mostrin per modificar	US40
Usuari amb sessió iniciada	Rebre notificacions	Estar al dia amb els missatges que m'arribin o els posts/tasques que es creïn en un grup de treball	Si	L'usuari ha d'estar en algun projecte, grup de treball o tenir algun xat individual actiu per rebre les notificacions, a més a més l'aplicació ha d'estar engegada per rebre les notificacions.	US42
Usuari amb sessió iniciada	Silenciar notificacions d'una sala	Perquè no m'interessen	No	L'usuari ha de seleccionar manualment el projecte, xat o grup que vol silenciar.	US43
Usuari amb sessió iniciada	Activar notificacions d'una sala	Perquè ara m'interessen	No	L'usuari ha de seleccionar manualment el projecte, xat o grup que tingui les notificacions silenciades per activar-les	US44

Figura 3.3.8: User Stories del bloc de usabilitat. Font: Pròpia

Aquestes User Stories estan enfocades per donar llibertat a l'usuari i millorar la seva usabilitat, les histories que estan basades en creació i modificació estan delegades al súper-administrador, això es degut a que són d'importància i no es canviaran molt de sovint. La pàgina d'ajuda es localitza fora de l'aplicació per tal de poder modificar-la completament sense haver de modificar el codi de l'aplicació, permetent així la integritat d'aquesta i que hi hagi més flexibilitat alhora de dur els canvis requerits en aquesta pàgina.

Com a...	Vull	Per tal de	Èpica	Criteri d'acceptació	Nº US
Súper-Administrador	Canviar la facultat d'un usuari	Moure'l a la facultat que pertany per haver seleccionat incorrectament la seva facultat a l'inici.	No	Estar a la pàgina d'usuaris de la facultat i que hi hagi usuaris diferents al meu	US36
Administrador o Súper-Administrador	Canviar el rol d'un usuari	Assignar el rol que li correspon realment a aquell usuari	Si	Estar a la pàgina d'usuaris de la facultat i que hi hagi usuaris diferents al meu	US25

Alumnes	Veure els integrants de tots els grups de treballs on estic inclòs	Tenir millor organitzats els usuaris relacionats amb mi	No	Estar en algun grup de treball	US35
Administrador, Súper-Administrador, Professor o Entitat	Veure els usuaris de tota la facultat	Tenir millor organitzats tots els usuaris, poder crear xats des d'allà i canviar el rol/facultat d'aquests en cas de ser administradors	No	Que hi hagi usuaris diferents al meu	US24
Súper-Administrador	Canviar d'interfície de facultat	Administrar altres facultats	Sí		US37

Figura 3.3.9: User Stories del bloc d'administració. Font: Pròpia

Aquestes últimes User Stories es basen en l'àmbit d'administració de l'aplicació, era necessari no permetre als alumnes poder veure tots els usuaris de la facultat ja que no hauria de ser d'interès al no poder fer cap acció sobre aquests a diferència de la resta d'usuaris.

3.4 Planificació

Durant el desenvolupament del projecte s'ha utilitzat la metodologia àgil Scrum que utilitza un sistema de Sprints. Aquest sistema està plantejat amb l'objectiu de poder organitzar i dividir les tasques que s'han d'implementar en el projecte. En el nostre cas, cada Sprint tenia una durada de 2-3 o 4 setmanes on al final s'organitzava una reunió que estava dividida en dues parts, la primera constava d'una explicació amb la mostra del avanços que s'havien fet fins aquell moment, la segona part constava de la planificació, on es decidien les User Stories que s'afegirien al desenvolupament. Com a part important, cada cop que s'acabava un Sprint es compartia un enllaç de descàrrega de l'aplicació per poder instal·lar-la en qualsevol dispositiu Android. Això era necessari perquè tant els tutors com la secció d'Aprenentatge servei proveïssin l'aplicació ajudant a detectar els errors i els elements a millorar. Cal matisar que el fitxer que es compartia era d'extensió .apk, és a dir, un fitxer que només es pot instal·lar en dispositius Android, ja que com s'ha esmentat en el capítol 2.1, les proves en dispositius IOS només es poden fer de manera gratuïta connectant el dispositiu mòbil al mateix ordinador on s'està desenvolupant el projecte, per tant totes les proves realitzades en aquesta plataforma han sigut realitzades només per mi.

Es van fer un total de 6 reunions, per tant, 5 Sprints:

Nº Sprint	Data inicial	Data final
1	09/10/2020	30/10/2020
2	30/10/2020	13/11/2020
2	13/11/2020	27/11/2020
4	27/11/2020	11/12/2020
5	11/12/2020	08/01/2021

Figura 3.4.1: Calendari dels Sprints. Font: Pròpia

Les User Stories es van anar distribuint en els diferents Sprints, a més a més es van anar afegint de noves cada cop que es feia la reunió de final-inici del Sprint.

Sprint 1:

Aquest va ser el primer Sprint on es van deixar clars els requeriments inicials i es va començar amb el desenvolupament del projecte. En aquest es van implementar alguns requeriments com són la creació d'un compte, l'inici de sessió, la recuperació de contrasenya, la modificació de dades de perfil, la creació i eliminació de projectes inclòs l'enviament de missatges en el xat, afegir usuaris al projecte, poder veure els projectes assignats i també es va afegir la creació d'un menú per poder navegar entre les diferents pàgines. Per tant aquestes US:

US01	Creació d'un compte
US02	Iniciar sessió a l'aplicació
US03	Tancar sessió
US04	Recuperar contrasenya
US05	Canviar les meves dades de perfil
US06	Crear projecte
US07	Eliminar un projecte
US08	Afegir usuaris a un projecte
US11	Veure els projectes assignats
US12	Enviar missatges en el xat

Figura 3.4.2: US Sprint 1. Font: Pròpia

Sprint 2:

En aquest Sprint no es va aconseguir implementar tots els requeriments previstos. Els que no es van poder acabar es van traslladar a un altre Sprint, entre aquests està la creació d'àudios que per problemes amb la creació d'un fitxer a partir d'aquest no es podia enviar correctament. La resta de requeriments que SI es van implementar són la creació a part i eliminació de xats individuals, poder canviar la imatge de perfil d'un projecte, eliminar usuaris d'un projecte, es va permetre la possibilitat d'eliminar missatges propis de qualsevol sala i poder visualitzar en pantalla completa una imatge. A més a més es va implementar el sistema de rols per poder controlar les accions de cada usuari que s'aplicarien en totes les funcionalitats vinents. Per tant aquestes US:

US09	Eliminar participants d'un projecte
US10	Modificació de la imatge de perfil d'un projecte
US13	Eliminar missatges en un xat
US14	Creació d'un xat individual
US15	Eliminació d'un xat individual
US16	Veure els xats individuals propis
US25	Canviar el rol d'un usuari

US26	Veure una imatge en pantalla completa
------	---------------------------------------

Figura 3.4.2: US Sprint 2. Font: Pròpia

Sprint 3:

Al Sprint 3 es va implementar els requeriments de visualitzar tots els fitxers rebuts en un xat assignat, es va afegir el sistema de facultats necessari per separar les interfícies, per aquest motiu es necessitava el rol del súper administrador per controlar totes les facultats, l'opció de crear i eliminar grups de treball a partir dels projectes a part de visualitzar-los en una pàgina enllaçada, l'opció de creació d'actualitzacions mitjançant post. Es va permetre també veure la llista d'usuaris de la facultat o en el cas dels alumnes dels seus grups de treball i finalment la possibilitat per part d'un súper administrador de moure's entre facultats. A més a més es van fer millores en la implementació de parts de codi i en l'aspecte visual. Per cada post creat s'enviava un missatge al grup de treball per informar d'un nou post afegit. Per tant aquestes US:

US17	Crear una actualització mitjançant un post
US18	Eliminar una actualització creada
US19	Veure els posts creats
US23	Veure tots els fitxers enviats o rebuts
US24	Veure els usuaris de tota la facultat
US30	Crear un grup de treball
US31	Eliminar un grup de treball
US32	Afegir participants a un grup de treball
US33	Eliminar participants a un grup de treball
US34	Veure els grups de treballs assignats
US35	Veure els integrants de tots els grups de treballs on estic inclòs
US36	Canviar la facultat d'un usuari
US37	Canviar d'interfície de facultat

Figura 3.4.3: US Sprint 3. Font: Pròpia

Sprint 4:

Per aquest Sprint es va desenvolupar el calendari i la possibilitat d'afegir i eliminar les tasques necessàries, la implementació d'un buscador, la reproducció dels àudios i la compartició de fitxers. Com es pot veure en termes de US no es van implementar gaires, ja que en aquest Sprint em vaig centrar en la solució de problemes i en l'optimització que requeria l'aplicació per tal de polir-la i preparar-la per les implementacions més complexes que es farien al final. Per cada tasca afegida al calendari s'enviava un missatge al grup de treball per informar d'aquesta.

US20	Veure el calendari del grup de treball amb les seves tasques
US21	Afegir una tasca al calendari
US22	Eliminar una tasca al calendari
US27	Reproducció d'àudios
US38	Compartir un element (Imatge/Àudio/Text)
US41	Cercar un text a la pàgina

Figura 3.4.4: US Sprint 4. Font: Pròpia

Sprint 5:

Aquest és l'últim Sprint que es va fer i on més coses dels requeriments es van implementar, ja que la durada d'aquest era d'unes 4 setmanes. A més a més de contenir alguns dels requeriments més complexos també es va haver de comprovar i adaptar l'aplicació a IOS ja que des d'aquell moment només es feien les proves a la plataforma d'Android. Es va implementar el sistema de notificacions en local que contenia la possibilitat de silenciar-les o activar-les per cada sala, la visualització i modificació de la pàgina d'ajuda, donar la possibilitat a un súper administrador d'eliminar els missatges d'algun grup per millorar el rendiment i finalment es va donar la possibilitat de crear i modificar l'estil de l'aplicació. Sobretot es van afegir requeriments no funcionals basats en la usabilitat.

US28	Veure la pàgina d'ajuda
US29	Modificar la pàgina d'ajuda
US39	Modificar l'estil de l'aplicació
US40	Afegir nous estils a l'aplicació
US42	Rebre notificacions
US43	Silenciar notificacions d'una sala
US44	Activar notificacions d'una sala
US45	Eliminar tots els missatges d'un projecte/grup

Figura 3.4.5: US Sprint 5. Font: Pròpia

3.4.1 Diagrama de Gantt

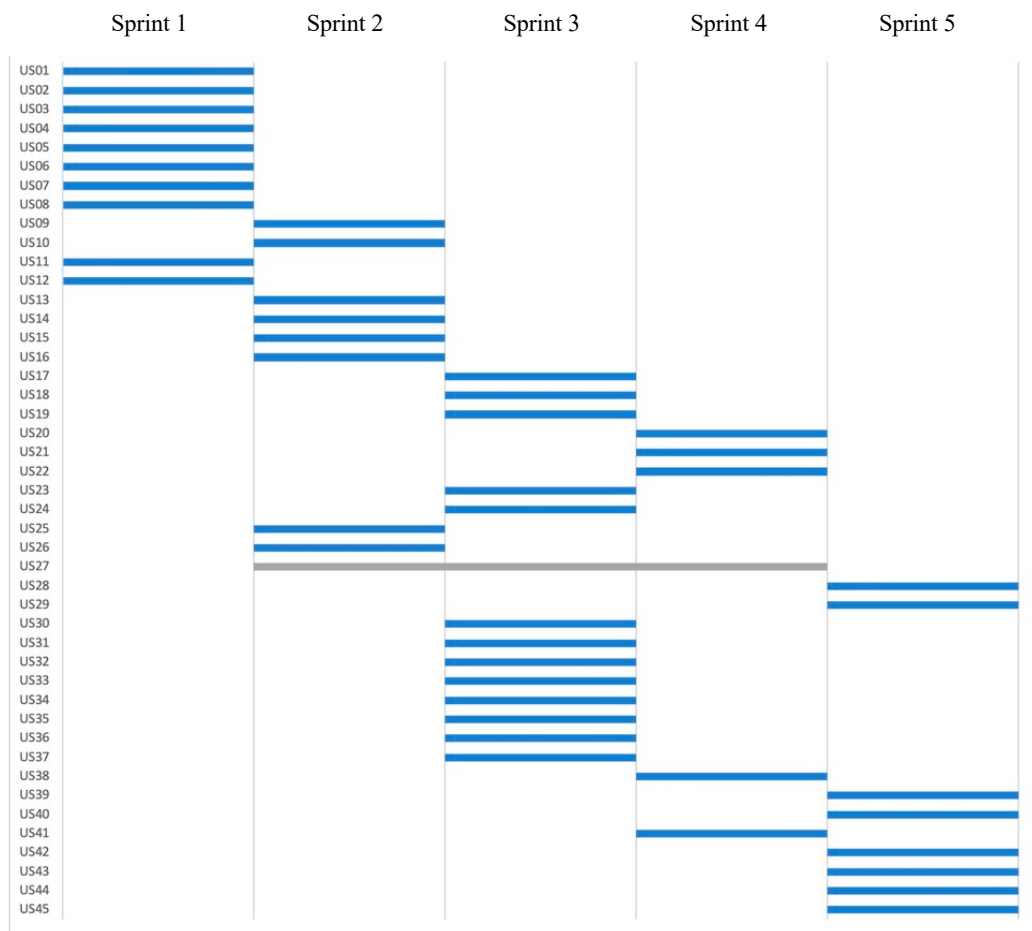


Figura 3.4.1.1: Diagrama de Gantt. Font: Pròpia

El diagrama està format per totes les User Stories fetes a cada Sprint per tal de veure millor l'ordre que s'ha seguit durant el desenvolupament del projecte. La US que està en gris, vol dir que es va iniciar a un Sprint, però es va allargar durant uns altres dos.

3.5 Cost del projecte

Firebase és la base de dades que vam escollir per al projecte. Aquesta base de dades és gratuïta sempre i quan no es superin un nombre d'operacions a la subjaent Firestore o la mida màxima d'emmagatzematge a la subjaent Storage. El límit d'operacions pot ser diari, mensual o global, en cas de superar-lo és necessari tenir el pla de pagament Blaze activat i en cas contrari l'aplicació deixarà de fer les operacions de la part on s'hagi superat. Aquest pla ens farà pagar cada cop que es superi un límit en un rang de temps marcat, per tant un cop superat aquest rang el comptador torna a 0. Aquí hi ha un enllaç amb l'explicació completa del servei i tots els costos per si es vol una informació completa i detallada: <https://firebase.google.com/pricing>.

Així doncs, el cost de la base de dades dependrà de l'ús que es doni regularment a l'aplicació, en les estimacions donades, pensant en un ús intensiu i en els serveis que nosaltres utilitzem (Firestore, Storage), el cost mensual seria:

- 50 Gb de fitxers (Àudios i Imatges) emmagatzemats extra = $0.026\$$ (Cada 5 Gb extres) $\cdot 50\text{Gb} = 1,30 \$$
- 2.30 Gb d'ample da banda diari = $0.108\$$ (Cada Gb extra) $\cdot 1.30$ (Ja que ens donen 1 Gb gratuït) $\cdot 30$ Dies = $4,22 \$$
- 250.000 Lectures diàries = $0.06\$$ (Cada 100.000 lectures extres) $\cdot 2$ (Ja que ens donen 50.000 lectures al dia gratuïtes) $\cdot 30$ Dies = $3.6\$$
- 1 Gb d'emmagatzematge extra de documents = $0.18\$$
- Total = $1.30 + 4,22 + 3,60 + 0.18 = 9.30 \$$

Tipus	Preu (Mensual)
75 Gb de fitxers	1,30 \$
2.65 Gb d'ample de banda diari	4.22 \$
250.000 lectures diàries	3.60 \$
1 Gb extra de documents	0.18 \$
Total	9.30 \$ = 111,6 \$ (Anuals)

Figures 3.5.1: Cost mensual base de dades. Font: Pròpia

Tot això de forma estimada. Respecte a l'emmagatzematge, el límit gratuït són 5 Gb estàtics, és a dir, no es reinicia amb el temps, per tant seria interessant eliminar-lo per cada curs d'estudis. Hi ha elements que, en cas de superar els seus límits, també incrementarien el preu però, en totes el temps que he estat desenvolupant l'aplicació no he vist la possibilitat de que es poguessin superar.

Per pujar l'aplicació a la Google Play (La tenda de la majoria de dispositius Android), s'ha de tenir un compte de desenvolupador que té un cost d'uns 25\$ que només s'ha de pagar un cop. Per altre part, en el cas de IOS mitjançant la App Store, es necessita un compte de desenvolupador que té un cost de 99\$ anuals.

Tipus	Preu
Deploy Google Play	25 \$
Deploy App Store	99 \$ (Anuals)
Total	115 \$ primer any, 99\$ la resta d'anys

Figures 3.5.2: Cost del deploy de l'aplicació. Font: Pròpia

El cost total dels dos tipus d'elements anteriors ens el donen en dòlars, s'hauria de transformar posteriorment depenen del estat en el que es trobi la moneda en el moment que es vulgui continuar amb el projecte. Encara que normalment es sol fer la conversió directa i no pel preu de moneda actual.

En referència al cost humà, és a dir, pel treball de desenvolupament i manteniment, s'ha de tenir en compte que, tot i que com a treballador seria de Junior (8 Euros la hora), el treball fet seria d'un àmbit més extens que implicaria a més tipus de treballadors, per tant estimem uns 15 euros la hora.

Tenint en compte que el treball està estimat en 14 setmanes (es va començar a finals d'octubre) les quals 2 s'utilitzaran per la memòria (per tant no es compten) , amb un treball diari de 6 hores en 3.75 dies, el cost total

humà que hi ha hagut en el desenvolupament és d'uns 4050 euros. Per tant allò també serà el preu final del desenvolupament ja que per la base de dades s'ha utilitzat el pla gratuït.

Tipus	Preu total
Cost humà	4050 Euros
Total del desenvolupament	4050 Euros

Figures 3.5.3: Cost del desenvolupament. Font: Pròpia

Per tant, de cost de desenvolupament tenim els 4050 euros, i de cost futur que s'haurà de fer en cas de que es vulgui mantenir l'aplicació i donar-li un ús constants és:

Tipus	Preu
Firebase	111,6\$ anuals
Deploy	115\$ Primer any, 99\$ la resta d'anys
Total	226 \$ primer any, 210 \$ la resta d'anys

Figures 3.5.4: Cost de manteniment de l'aplicació. Font: Pròpia

4. DISSENY

Com s'ha esmentat en el punt 2.1 es va decidir per utilitzar la tecnologia Ionic juntament amb el framework Angular.js. Això, com veurem ara, determina l'estructura que tindrà el nostre projecte. En aquest apartat podrem veure aquesta arquitectura amb el models de la base de dades, l'estructura de les dades, els diagrames de classes i els patrons utilitzats.

4.1 Arquitectura i model de dades

L'estructura del projecte que es seguirà és el de MVC (model-vista-controlador) ja que angular.js utilitza aquest patró. Això ens permet que les diferents parts de l'aplicació siguin independents entre si. Aquesta estructura està dividida entre plantilles, components i serveis. Els serveis són una classe global que s'instancien només un cop i són útils per poder utilitzar les seves funcions en el component que ho requereixi. Els components són l'estructura d'una pàgina que conté l'estil, la vista, les plantilles, etc. Les plantilles o templates són elements que s'utilitzen com a vista d'un component per tal de mostrar la pàgina d'aquest.

Aquesta estructura està enllaçada directament amb una base de dades d'un servidor extern de Google. La base de dades és Firestore que pertany a l'entorn de Google Firebase, és de tipus no relacional en forma de documents. Aquests documents estan creats utilitzant un model de dades de tipus JSON, per tant les operacions es fan utilitzant aquest format fent que les dades de la nostra aplicació hagin d'estar adaptades, a més a més es troben continguts en col·leccions per tal d'organitzar-los i classificar-los. Al tenir una aquesta base de dades externa no necessitem d'un backend pur ja que ho gestiona la mateixa, incloent la seguretat, això sí, amb una lògica creada manualment.

Per tant el nostre projecte en definició està distribuït en tres parts: La part de la lògica de l'aplicació que es comunica amb la base de dades i fa les operacions necessàries per la vista, la vista de l'aplicació que és traduïda per Ionic amb l'objectiu de mostrar totes les pàgines i finalment la base de dades que és l'encarregada de rebre les peticions i enviar les respostes.

4.2 Estructura de dades

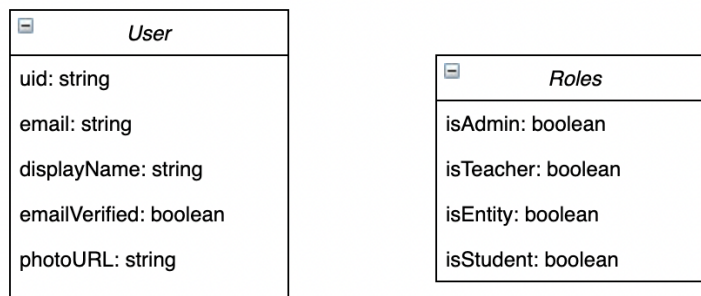
Per poder estructurar la nostra base de dades s'ha d'utilitzar una certa organització de les dades per tal de permetre fer les operacions corresponents per a què la nostra aplicació pugui funcionar correctament i en la seva totalitat. Una de les coses negatives d'una base de dades no relacional és que no es pot detectar els canvis des d'una col·lecció diferent, per tant es va haver d'anar guardant les claus o identificadors dels elements que volia tenir a mà en les col·leccions on fos necessari.

La idea de l'aplicació és emmagatzemar absolutament totes les dades a la base de dades facilitant no dependre gairebé del emmagatzematge intern dels dispositius mòbils. Per tant, era necessari dissenyar una estructura de dades eficaç i de connexió lògica.

Utilitzem una sèrie d'interfícies on trobem declarades unes taules que posteriorment s'utilitzen en la creació de documents de les diferents col·leccions. Aquestes són:

User.interface.ts

Firebase ens demana que alhora de la creació d'un usuari hi hagi uns camps específics, el que ens permet a nosaltres poder modificar-los, però no afegir-ne de nous, per tant, s'ha d'inserir un camp nou com pot ser el de facultat o una taula diferent, s'ha de fer a part i després enllaçar-ho per tal de crear un document únic per a la col·lecció d'usuaris. Per als rols d'usuaris vam crear una taula a part dins de la mateixa interfície.



Figures 4.2.1: model de User.interface. Font: Pròpia

De la taula User, uid és l'identificador de l'usuari, per tant del document que el conté, email és el correu electrònic que ha escollit, displayName el nom i cognom personals, emailVerified és el camp on controlem si l'usuari ha confirmat el seu compte per tal de permetre iniciar sessió o no i photoURL és el camp que utilitzarà per afegir la imatge de perfil que vulgui. Per part de la taula de Roles, isAdmin (és administrador?), isTeacher (és professor?), isEntity (és entitat) i isStudent (és estudiant?) assignaran el rol actual de l'usuari, estan continguts en una taula ja que per exemple, un professor pot ser un administrador i és necessari tenir-los tots a la mà.

Chat.interface.ts

En el nostre cas els usuaris no tenen com a camp els xats individuals i dels projectes assignats (L'aplicació està adaptada per poder fer-ho), per tant els documents dels xats són els que contenen els usuaris. Hem creat dues taules, una que construeix un xat d'un projecte o de tipus individual, i una taula que conté l'usuari amb el seu identificador uid, de manera que hi hagi control en els participants.

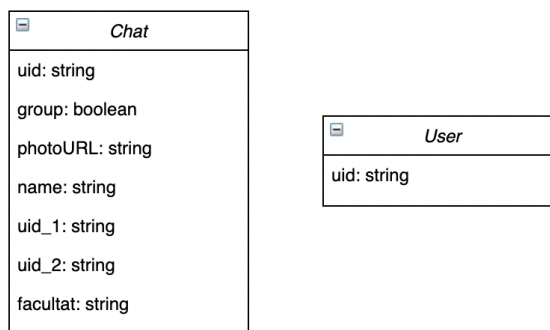


Figura 4.2.2: model de Chat.interface. Font: Pròpia

Com veurem al model relacional, un usuari s'afegeix al document dels xats mitjançant una ArrayList creada directament. Uid és l'identificador del xat i també del document que el conté. El camp group defineix si el xat és

individual o un projecte, en cas de que sigui individual només tindrà en compte els camps uid_1, uid_2 i el de facultat, ja que la imatge de grup i el nom depenen de l'altre participant del xat. Vaig afegir uid_1 i uid_2 per no haver de llegir mitjançant un bucle l'array d'usuaris del xat, ja que només hi haurien dos, de manera que es millorés l'eficiència. El camp de facultat serveix per poder organitzar els xats i mostrar-los només en la facultat on han estat creats. En la part de la taula de User només tenim uid ja que la resta de dades com s'ha esmentat anteriorment es llegeixen de la taula de User.interface.

Work.interface.ts

En el cas de la interfície de Work trobem una taula que conté les dades necessàries per la creació d'un grup de treball. Serà necessari que primerament s'hagi creat un xat de projecte i per després crear el grup de treball.

<i>Work</i>
id: string
photoURL: string
name: string
facultat: string

Figura 4.2.3: model de Work.interface. Font: Pròpia

Tenim un identificador del grup de treball el qual és el mateix que el del document contenidor, el camp de photoURL on nosaltres no hem activat la possibilitat de modificar la imatge del grup en aquest cas però amb la possibilitat de poder permetre en un futur si es volgués, el nom del grup i la facultat on es troba el treball.

Messages.interface.ts

Com diu el seu nom aquesta interfície conté la taula per l'administració dels missatges que s'envien en qualsevol sala de xat de l'aplicació. Al ser una taula que utilitzaran els documents tant per xats/projectes com per grups de treballs era necessari crear una interfície només per aquesta.

<i>Messages</i>
id: Date
content: string
type: string
date: number
user: string
name: string

Figura 4.2.4: model de Messages.interface. Font: Pròpia

Aquesta taula és una mica més complexa que la resta. Trobem un identificador que com es pot veure és de tipus Date, sent imprescindible per missatges de tipus àudio i imatge, ja que els arxius a la Firebase Storage es guarden amb el nom de la data de creació. Per tant, alhora d'eliminar un missatge, és necessari que també ho fes l'arxiu i per aconseguir això hem tenir la data exacta. El content és el contingut del missatge, pot ser l'escrit d'un usuari per enviar o una direcció URL de l'arxiu enviat. També disposem d'un camp date de tipus número per tal de poder ordenar els missatges per data de creació i els xats d'aquella sala per l'últim missatge rebut. El

camp user amb l'identificador de l'usuari que ha enviat el missatge i finalment tenim name que és el nom de l'usuari que ha enviat el missatge.

Events.interface.ts

És necessari tenir una taula que contingui les dades d'una tasca creada per part de l'entitat o professor, per tal de poder sincronitzar el calendari del grup de treball amb una llista de tasques creades, permetent així que tots els participants el puguin visualitzar.

<i>Event</i>
title: string
startTime: Date
endTime: Date
allDay: boolean

Figura 4.2.5: model de Events.interface. Font: Pròpia

No trobem un identificador, en aquest cas vaig preferir utilitzar l'identificador propi del document sense assignar-lo a la taula, ja que les accions que només s'executen són les de la creació i eliminació de la tasca, per tant no era necessari tenir un identificador per seleccionar individualment la tasca en un altre pàgina que no fos la del calendari. Title és la descripció que volem posar a una tasca, startTime la data on comença i endTime on acaba, finalment allDay per dir si és una tasca que dura tot el dia o no. En la nostra aplicació hem preferit mantenir tasques curtes ja que ajuda a millorar l'organització, però com es veu hi ha la opció de poder implementar tasques que durin tot el dia si es volgués en el futur.

Post.interface.ts

En la taula d'aquesta interfície trobem els camps de les dades necessàries per poder crear una actualització mitjançant un Post.

<i>Post</i>
id: Date
photoURL: string
audio: string
text: string
date: number
user: string
name: string

Figura 4.2.6: model de Post.interface. Font: Pròpia

Troblem un identificador de tipus Date per ordenar a la base de dades els posts per data, photoURL que contindrà la direcció de l'imatge creada per el Post, audio que contindrà la direcció de l'àudio creat pel Post, text que conté el text que ha escrit l'usuari per explicar el treball fet i també tenim també una data (date) que aquesta és útil per l'administració en part de la vista, user amb l'identificador de l'alumne i name amb el nom del alumne.

Color.interface.ts

Aquesta interfície té definida la taula de colors que es trobarà com a document dins de la col·lecció de colors. S'utilitza per canviar l'estil de l'aplicació amb la utilització de colors en certs components de la interfície de l'aplicació. Aquesta taula serveix tant per crear nous estils com per carregar-los a l'usuari evitant no haver de canviar l'estil cada cop que s'obre l'aplicació. Només es poden canviar alguns colors per tal de no donar peu a errors alhora d'escollir els colors fent que l'estil que es creés, fes inutilitzable l'aplicació.

Colors
id: string
maincolor: string
secondarycolor: string
thirdcolor: string
invertedcolor: string
chatcolor: string
privatecolor: string
titlecolor: string

Figura 4.2.7: model de Color.interface. Font: Pròpia

Els colors de la taula són valors en hexadecimal. Id és l'identificador del color, per tant el del document que el conté, maincolor és el color principal de l'aplicació com per exemple pel header de l'aplicació, secondarycolor és el que s'utilitza per al fons del contingut d'algunes pàgines, thidcolor s'utilitza per al fons d'alguns selectors d'imatges, invertedcolor és el color necessari per el tipus de rol d'usuari del perfil i de la facultat del menú, chatcolor és el que es fa servir per al fons del contingut dels xats individuals i de projectes, privatecolor és el color de fons del contingut dels xats de grups de treballs i titlecolor s'utilitza pels títols de l'aplicació.

Altres dades

Per la creació de documents també utilitzem altres camps de dades. Per als documents d'usuaris tenim un camp de facultat de tipus string, un altre de tipus Color que assigna un estil a l'aplicació de l'usuari per tal de carregar-lo cada cop que iniciï l'aplicació i finalment dos ArrayList, un per als identificadors dels xats silencisats i l'altre per als identificadors dels grups de treballs assignats a aquell usuari.

Per part dels documents dels xats/projectes i grups de treball tenim els mateixos, dos ArrayList, una de missatges per als objectes de la taula de Messages de Message.interface i l'altre els objectes de la taula User de Chat.interface (comparteixo taula per no repetir encara que sigui d'un altre interfície).

Els identificadors, com abans he esmentat, són necessaris per poder llegir la informació d'altres col·leccions.

4.3 Model relacional

Aquí mostrarem el diagrama amb el model relacional de la nostra base de dades de Firestore, la part de Firebase Storage es troba fora del diagrama del model relacional i és on s'emmagatzemen els fitxers que després utilitzem mitjançant unes direccions URL per tal de mostrar-los i descarregar-los, com veurem a l'apartat d'implementació 5.0.

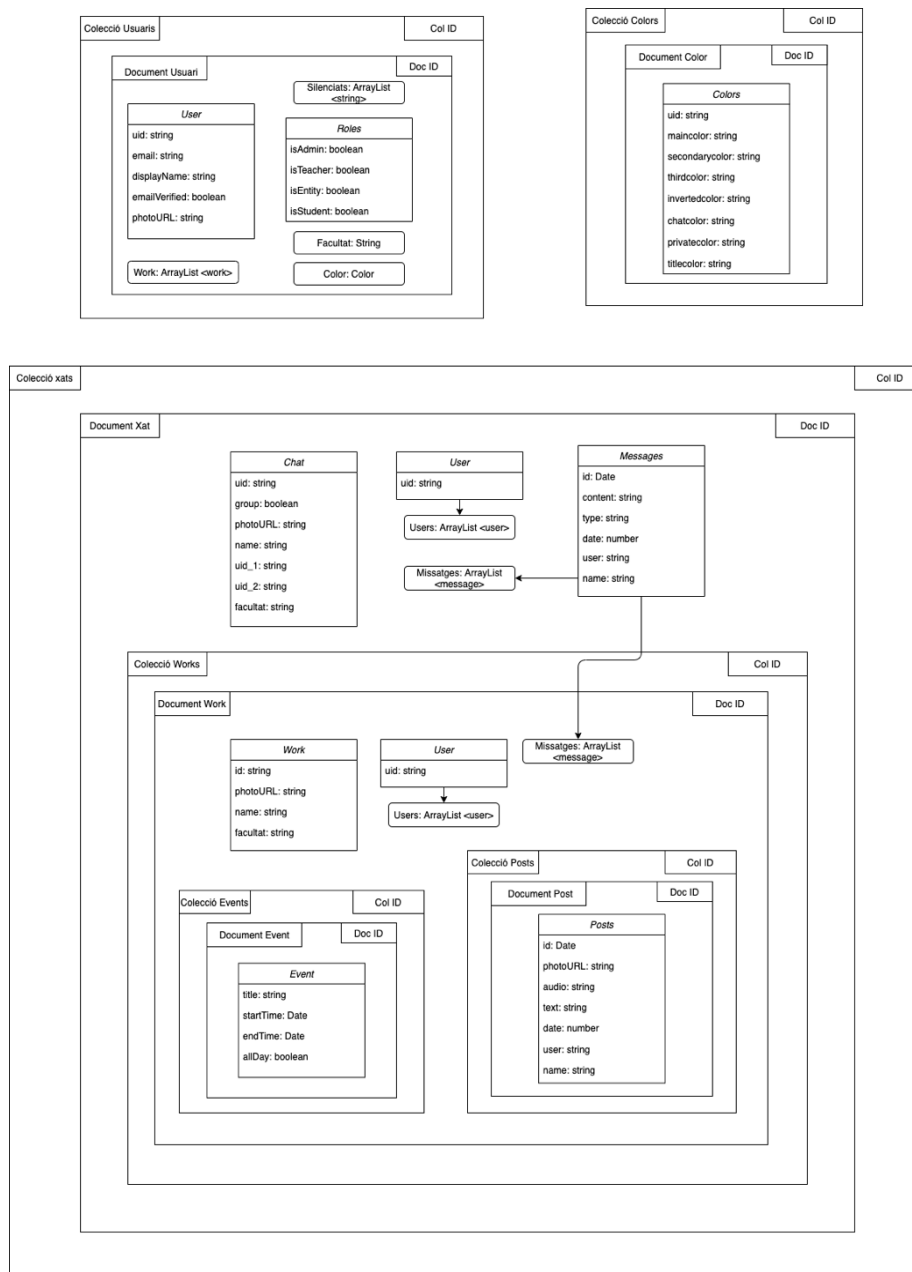


Figura 4.3.1: model relacional de la base de dades. Font: Pròpia

Com podem veure tenim tres col·leccions, la dels usuaris, la dels colors i la dels xats. Dins de cadascuna trobem una sèrie de documents que serveixen per omplir cada col·lecció. A diferència de la resta, en els documents de xats tenim inclosa la col·lecció de Works (grups de treball) i dins dels documents d'aquest, trobem incloses les col·leccions d'Events (tasques) i de Posts que al no ser una base de dades relacional és l'única manera de poder tenir continguts certs elements que han d'estar relacionats entre ells, és a dir, que cada col·lecció depèn del document que la conté, si el document d'un xat s'elimina, les col·leccions que contingui també ho faran automàticament.

Al no haver la possibilitat d'ús de claus foranies, fa que utilitzem uns ArrayList que contenen els identificadors dels documents de xats silenciats de l'usuari i els dels treballs, perquè en cas de que vulguem llegir la informació d'aquestes col·leccions puguem utilitzar aquests identificadors. El mateix passa amb els colors, en aquest cas si es borra el color en la base de dades si l'usuari l'està utilitzant, no s'eliminarà fins que canviï d'estil.

4.4 Patrons de disseny

Com s'ha esmentat en l'apartat d'Arquitectura, per aquest projecte utilitzem el patró de MVC (model-vista i controlador) per tal de poder millorar la independència dels components.

El patró més important que vaig utilitzar va ser el de Observable, l'objectiu d'aquest patró és detectar els canvis d'estat d'un objecte per notificar-los a les funcions dependents. Nosaltres utilitzàvem una base de dades externa de forma que hi hagués constantment canvis produïts per usuaris diferents al propi, era necessari estar escoltant constantment per si es produïa algun. Per exemple, si un usuari era afegit a un xat era essencial que a aquest li sortís el xat a la seva interfície i pogués començar a fer les accions relacionades.

4.5 Diagrames de components

En la nostra aplicació trobem una estructura que es divideix en diferents parts. Per una part tenim les pàgines, on estan els components de les pàgines de l'aplicació amb la seva vista i la seva lògica, trobem també un modal que utilitzem per mostrar les imatges en pantalla completa i una part de Services on estan les classes que s'instancien només un cop i proveeixen a les pàgines els serveis necessaris. Tenim també una carpeta anomenada Shared que conté algunes classes que ens ajuden a encapsular algunes funcions que s'utilitzen a gairebé totes les pàgines per tal de no haver de repetir-les, al igual que els serveis. Es disposa també d'algunes que s'instancien només un cop però no donen accés a les funcions de la base de dades. Finalment trobem un component app-component que és la classe mare de l'aplicació.

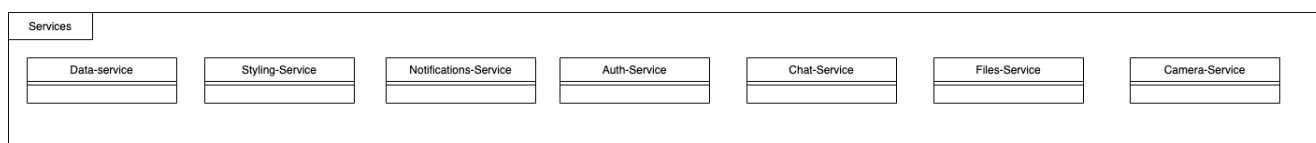


Figura 4.5.1: Serveis . Font: Pròpia

- Data-service serveix per ordenar les llistes rebudes de la base de dades.
- Styling-service és el servei on trobem les funcions que són necessàries per l'administració dels estils, com la creació, lectura, modificació o eliminació.
- Notifications-service es troben les funcions necessàries per enviar i rebre les notificacions de forma local.
- Auth-service és el servei que controla la col·lecció de Users i tots els seus derivats.
- Chat-service funciona de forma similar a l'anterior servei però aquest cop controla la col·lecció i documents dels Xats, projectes, grups de treballs, posts i events.
- Files-service té una part on es controla la connexió amb la firebase Storage amb els fitxers d'àudio que s'envien i un altre on es creen els arxius per tal de reproduir-los o compartir-los.
- Càmera-service controla l'element de càmera nativa d'un dispositiu mòbil i de manera semblant a files-service controla la connexió amb la firebase Storage però en aquest cas dels fitxers de tipus imatge.

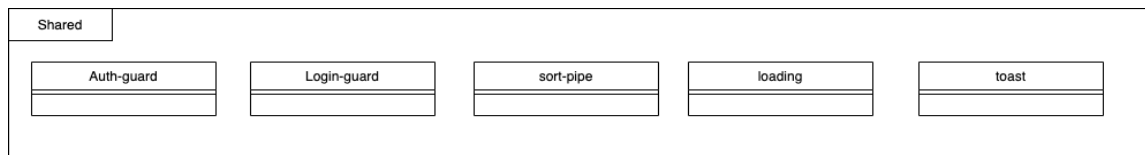


Figura 4.5.2: Shared . Font: Pròpia

Auth-guard s'utilitza en el Routing-module de l'aplicació. En aquest Routing trobem totes les direccions de la pàgina i mitjançant aquest guard controlem que només s'accedeixi a la pàgina que volem, controlant si l'usuari té la sessió iniciada i si l'usuari ja ha seleccionat la facultat del seu compte. D'una manera similar funciona Login-Guard. Sort-pipe és una classe que s'utilitza només per ordenar una llista des de la vista i no des de la lògica. Loading és una classe que és útil alhora de fer accions a la base de dades per tal de fer esperar l'usuari fins que s'hagi completat una operació. Per acabar, Toast ens permet treure missatges a l'aplicació de tipus toast com diu el seu nom.

Al ser una aplicació basada en aquest sistema de components fa que hi hagi independència entre la resta de components.

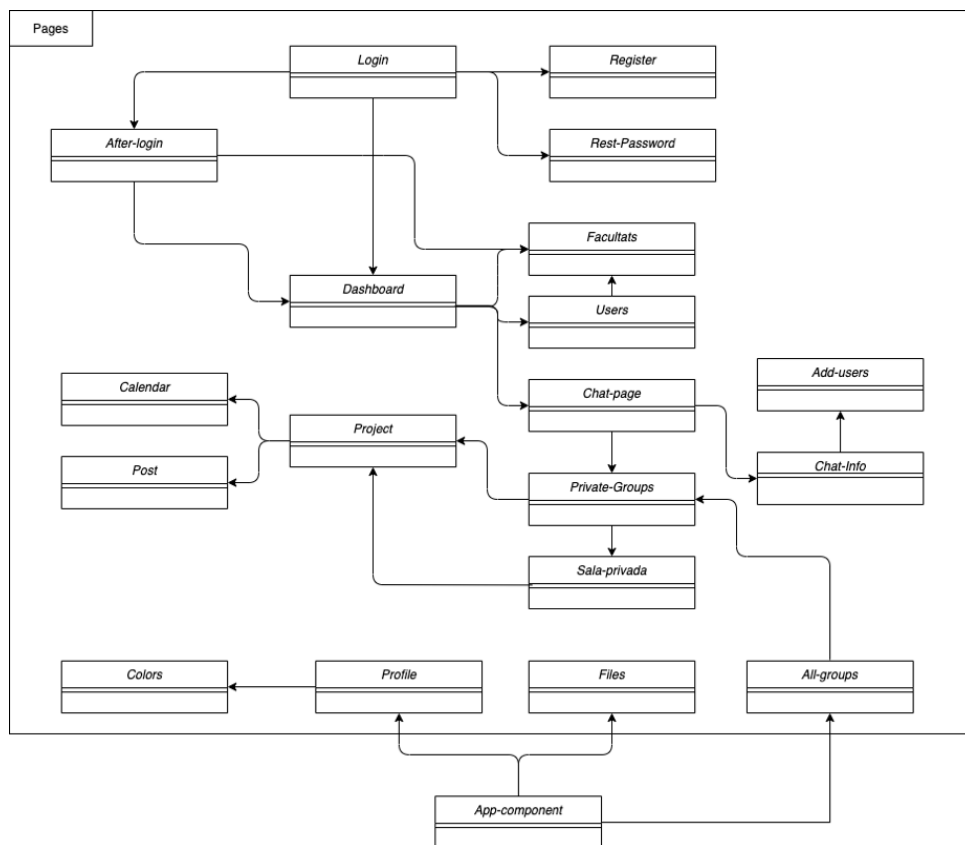


Figura 4.5.3: Diagrama de components de les pàgines. Font: Pròpia

Les pàgines de l'aplicació són les següents:

- Login: Per iniciar sessió a l'aplicació.
- Register: Ens permet crear un nou compte.
- Reset-password: Ens dona la possibilitat de recuperar/reiniciar la contrasenya.
- After-login: Ens redirigeix a la pàgina de facultats per poder seleccionar per primer cop la facultat.

- Dashboard: Pàgina principal de l'aplicació on es mostren tots els xats i projectes, a més a més és el component on es controlen tots els canvis als xats per tal d'enviar les notificacions a l'usuari de tipus local.
- Facultats: Pàgina auxiliar que ens ajuda a escollir una facultat, ja sigui per canviar-la o afegir-la.
- Users: Llista d'usuaris de la facultat o en cas d'alumnes, llista de tots els participants de grups de treball.
- Chat-page: Pàgina on es visualitza el xat individual o de projecte.
- Chat-info: Pàgina auxiliar on es veuen els usuaris d'un xat anterior i ens dóna diferents opcions.
- Private-groups: Els grups de treballs d'un projecte.
- Sala-privada: Pàgina on es visualitza el xat d'un grup de treball.
- Projecte: Secció educativa d'un grup de treball, ens permet veure la llista dels posts i crear-ne un.
- Calendar: Calendari sincronitzat d'un grup de treball.
- Post: Pàgina on es visualitza un post creat.
- Files: Es visualitza la llista de fitxers rebuts i enviats en els diferents xats d'on un és integrant.
- Profile: Pàgina de perfil que ens permet modificar els camps personals del nostre compte.
- Colors: Ens permet seleccionar i crear un estil (súper-administrador) per la nostra aplicació.
- All-groups: Visualització només dels projectes.
- Add-users: Llista d'usuaris per afegir-los en un projecte o grup de treball.

En aquest diagrama tenim la distribució dels components de les pàgines i l'ordre en el que s'instancien. S'ha de tenir en compte que totes les relacions són 1..1 ja que aquest sistema de components utilitza les pàgines com un únic objecte, per tant encara que hi hagi un nombre elevat de xats, es carregarà en la mateixa pàgina de chat-page.

Com es pot veure tenim el App-Component fora de la carpeta de pàgines i en aquest, trobem el menú de l'aplicació que ens permet anar cap a les pàgines senyalades. La ruta inicial està assignada a la pàgina de login, però en cas de tenir la sessió iniciada ens dirigirà directament a dashboard. Si és el primer cop que iniciem sessió a l'aplicació, se'ns redigirà a la pàgina de after-login que al mateix temps aquesta ens redigirà cap a la de facultats per seleccionar la nostra facultat. Tenim la opció d'anar a la pàgina de projecte sense passar per la sala del grup de treball (sala-privada) des de la pàgina de all-groups.

5. IMPLEMENTACIÓ

El framework Ionic utilitza les tecnologies de HTML i Css per la part de la vista i Typescript per la part de la lògica. Com es pot veure, la programació de la vista és bastant similar a la d'un desenvolupament web, encara que, ens dóna l'oportunitat de poder utilitzar una sèrie de components pre-dissenyats i d'alguns elements que no es podrien trobar en un desenvolupament web tradicional. Si utilitzem aquests components amb el framework de Angular tenen aquesta estructura:

```
<ion-x > </ion-x> (x es el component que volem, pot ser button o qualsevol altre)
```

Ens deixa afegir algunes característiques personalitzades, ajudant-nos a simplificar el disseny de l'estil de l'aplicació per tal de intentar utilitzar el menys possible la part de Css. Un exemple pot ser el de seleccionar l'estil d'un botó:

```
<ion-button shape="round">Inici de Sessió</ion-button>
```

Es va haver d'utilitzar per les pàgines amb molts components un sistema de grid o quadrícula responsiva que empra elements de columnes(col) i files(row) que ens ajuden a distribuir els elements de manera independent i ordenada, sense haver d'afegir la mida de cadascun d'aquests elements, fet que ens permet reduir la complexitat de codi i els problemes que pot donar quan s'afegeix algun component nou. Encara que en dos casos en particular, el del calendari i la barra inferior que s'utilitza per enviar missatges als xats, sí es va utilitzar el sistema no responsiu de columnes i files, que implica haver d'afegir la mida o size d'aquest. El sistema de grid implementat té aquesta estructura:

```
<ion-grid>
  <ion-row>
    <ion-col></ion-col>
  <ion-row>
</ion-grid>
```

Tenint en compte aquestes funcionalitats, es va implementar de forma auxiliar els estils Css, ja que ens ajuda a que hi hagi més possibilitats alhora de fer el disseny i l'estructura de la part visual per tal de que sigui compatible per al major nombre de dispositius mòbils tenint en compte que hi ha molta diversitat de mida de pantalles. A més a més per la part de la pàgina externa d'ajuda es va utilitzar una plantilla bàsica de bootstrap per la implementació d'aquesta, mitjançant l'ús de Javascript, css, html5 i jquery.

En el cas de la lògica, primerament era necessari instal·lar tots els complements i paquets que s'han d'utilitzar per al desenvolupament. Per la instal·lació es va utilitzar el sistema de paquets NPM de node i el de Ionic + Cordova. Aquests complements són per poder accedir als sistemes dels dispositius mòbils com poden ser la càmera, el micròfon, etc. També són necessaris per l'administració i entorn de fitxers, les notificacions, el mode background, el calendari, la selecció de colors per crear l'estil, per detectar la regió on es troba el dispositiu alhora de saber el fus horari, per poder ordenar una llista directament a la vista, pel control del teclat i finalment per la integració de la base de dades Firebase i el seu entorn. Tots aquests paquets per poder utilitzar-los s'han de declarar manualment a *app.module.ts*.

Per poder transmetre dades entre pàgines utilitzàvem la funció *router.navigate*, on s'introdueix la URL de la direcció de la pàgina on volem anar i els valors que volem passar a aquella. Això és útil perquè la pàgina destinant rebi les dades per poder actualitzar la pàgina. Per exemple, en cas dels xats és necessari que des de fora es notifiqui quin és el xat i si és un grup o no per tal que la pàgina s'actualitzi i mostri les dades i així poder enviar i rebre els missatges del xat que es vol.

5.1 Algoritmes

Es va haver d'implementar una sèrie d'algoritmes per les diverses funcionalitats de l'aplicació. A l'haver un gran nombre, faré una selecció dels més importants o amb una estructura en comú.

Com s'ha esmentat en l'apartat 4.4, utilitzem una sèrie d'observables que apunten directament a unes direccions de la base de dades per tal de poder notificar els canvis que es produeixin. Aquests observables els creem utilitzant la funció *subscribe* de Angular.js, aquesta estarà apuntant a aquell objecte fins que nosaltres decidim que ho deixi de fer. El objectes que es creen des de la base de dades són administrats pels diferents serveis que hi ha a l'aplicació, amb l'objectiu de tenir una millor independència entre les direccions.

Per exemple, si volem agafar un sala de xat en concret l'haurem d'escollar per tal de no perdre els canvis que es puguin produir, utilitzant pseudocodi:

```
Entry : Id
subscribe chatRoom with that Id then
.      chats = chatfromFirebase
```

Figura 5.1.1: Pseudocodi funcionalitat subscribe. Font: Pròpia

Com es pot observar escoltem el *chatfromfirebase* i l'assignem al xat de la classe on estem. Aquesta variable és del tipus *Chat* de *chat.interface* que conté tots els valors. En cas de que algun d'aquests es modifiqués, el mètode automàticament, enviarà el nou valor. Aquest mètode sol estar contingut en les inicialitzacions dels components per tal d'escollar-los encara que es surti de la pàgina, però també hi ha la possibilitat d'escollar-lo o no cada cop que s'entri o es surti de la pàgina.

Per utilitzar aquests *subscribe*, com s'ha esmentat anteriorment, utilitzem unes funcions per retornar el objectes, en aquest cas es tracta d'una sala de xat normal i es farà des de *chat.service.ts*, per tant retornem d'aquest mode:

```
Function: getChatRoom()
Entry : Id
Exit : The chat room with that id
return valuechanges Document with that id from chatRooms Collection
```

Figura 5.1.2: Pseudocodi retorn de xat mitjançant *valuechanges*. Font: Pròpia

Com podem observar utilitzem *valuechanges*, ens permet poder utilitzar la funció *subscribe* després per aquest l'objecte per tal de veure el canvis que es van anant fent a aquell objecte.

```

Function: getChatRooms()
Entry : None
Exit : All the ChatRooms
return snapshotChanges Collection chatRooms then
.       return Document map then
.       assign Document.id to Map data.id then
.       return Data changed

```

Figura 5.1.3: Pseudocodi retorn de tots els xats. Font: Pròpia

Utilitzem aquest algoritme per retornar molts elements des de la base de dades, com poden ser tots els xats, grups de treballs, etc. Aquest és diferent a l'anterior, ja que li afegim a cada document un identificador per no haver de dependre exclusivament d'un identificador assignat a una taula dins dels documents, a més a més utilitzem `snapshotChanges` que ens permet poder canviar l'identificador, però també observar els diferents canvis que es produeixen a l'objecte de la mateixa manera que ho fa `valueChanges`. Aquest algoritme no modifica el de `subscribe`, per tant, fa que no s'hagi de modificar la funció de qui el crida, però si tindrà la opció d'anar llegint els diversos identificadors durant el recorregut d'una llista.

Mitjançant l'estructura d'aquests d'algoritmes d'observació i retorn de dades de la base de dades s'ha construït gairebé tota l'aplicació. Per exemple, en el cas de les notificacions el que fem és llegir els grups de treball que té l'usuari assignat mitjançant recorregut de tots els xats que es troba com integrant, per tal de estar observant si l'han afegit a algun grup, projecte, un missatge nou a un xat o grup de treball i per veure si algun integrant a afegit una tasca o un post dins d'algun grup de treball, comprovant amb llistes externes si hi ha algun canvi amb la llista que s'ha creat nova per als canvis. Aquestes notificacions s'envien utilitzant el paquet `LocalNotifications` de Ionic, i com es pot deduir de l'explicació anterior, les notificacions les envia el mateix dispositiu mòbil, és a dir, són en local, el que produeix que si l'aplicació està tancada no es pugui rebre aquestes notificacions. Es va implementar també la funcionalitat de `Background`, on s'habilita que l'aplicació en cas d'estar de fons no es tanqui automàticament i per tant, rebre les notificacions.

Per l'administració i creació de fitxers s'utilitza `Storage` de firebase. Quan nosaltres volem descarregar un fitxer enviat, ho fem mitjançant la funció `getDownloadURL`, on s'afegeix la direcció a la carpeta on s'ha guardat i també ens retorna la direcció web d'on es troba el fitxer desitjat. En el cas d'imatges utilitzem un string de base64 per transformar la imatge i pujar-la com a string a la base de dades. Aquesta imatge es crea des del dispositiu, pot ser mitjançant la càmera o la galeria (Ens donarà l'opció d'acceptar els permisos si l'aplicació s'està utilitzant des de Android) i es va posar una qualitat estàtica per tal de no excedir amb facilitat el límit mensual d'emmagatzematge estimat. A l'hora de visualitzar-la, la base de dades ens permet fer-ho utilitzant la URL mitjançant la funció anterior `getDownloadURL`, en canvi, pels fitxers d'àudio s'ha de descarregar el fitxer d'aquella direcció i es guarda com a arxiu al dispositiu mòbil. A partir d'aquell moment l'aplicació podrà reproduir o compartir el fitxer utilitzant el paquet de `Media` de Ionic. Respecte a aquests fitxers d'àudio, per poder pujar-los a la `Storage`, és necessari crear un `Blob` a partir del fitxer transformat en un `Array buffer` utilitzant la funció `readAsArrayBuffer`.

6. RESULTATS

En aquest apartat veurem els diferents resultats obtinguts després de la implementació de l'aplicació. Trobarem la interfície final i els resultats dels test obtinguts durant tot el desenvolupament de l'aplicació.

6.1 Testing

Com s'ha esmentat en l'apartat 3.4, a final de cada Sprint es compartia l'aplicació actualitzada. Gràcies a aquestes proves fetes per diversos usuaris es van aconseguir solucionar alguns errors relacionats amb la implementació lògica de l'aplicació, a més a més es va obtenir una sèrie d'opinions respecte al disseny de la interfície de l'aplicació que van ajudar a millorar-la.

Errors trobats durant el testing

Error	Solucionat	Solució
Redirecció a la pàgina de dashboard cada cop que es canviava un camp del perfil d'usuari.		Cancelar subscripció del nostre usuari a Auth.guard.
Canvis no efectius de camps modificats al perfil.		Canvi d'implementació de la modificació d'aquests camps a la firestore.
No reproducció d'àudios a IOS.		Canviar el tipus d'URL del fitxer de descàrrega creat i l'extensió del fitxer a .wav
No s'eliminaven les col·leccions i documents de la firestore que es trobaven com fills d'un altre col·lecció en cas de que aquesta s'eliminés.		Eliminar també tots els fills quan s'eliminés una col·lecció.
Data de tasca incorrecta en el missatge d'alerta alhora de crear una tasca al calendari.		S'utilitzava una funció incorrecte de l'entorn Date per al mes de la data. Canvi de funció.
Error en la conjunció dels noms de les facultats que comencen per vocal, com educació, informació i medis audiovisuals o economia i empresa,		Corregir la conjunció.
L'aplicació es tancava si s'intentava enviar una imatge o àudio a Android.		Per les noves versions d'Android és necessari afegir un permís al manifest per tal de permetre l'ús d'aquests elements.
Fitxers no ordenats per data.		Fer una funció d'ordenació per data d'afegit.

Figura 6.1.1: Taula d'errors durant el testing. Font: Pròpia

Recomanacions

Recomanació	Implementat	Adaptació de la recomanació
Ordenació de xats per tipus.		Donar opció de selecció de xats individuals o projectes.
En la pàgina de xat quan es rebi un missatge baixi automàticament o notifiqui amb una bombolla al estil WhatsApp o Telegram.		Quan es rebi un missatge baixarà automàticament si estem en una posició propera al final del xat, si no es veurà una bombolla que avisarà d'un nou missatge.
Canvi en la implementació de l'observació dels xats propis a la dashboard.		
Un cop s'entri a una sala de xat ens posicioni als últims missatges.		No es pot fer immediatament. Ens posicionarà al final un cop s'hagin carregat tots els missatges.
Es vegi l'hora de l'últim missatge fora del xat.		
Canviar disseny de les imatges que es pre-visualitzen als xats.		Diferent a IOS i Android, aquest primer es veurà com thumbnail.
Disseny més intuïtiu en la navegació entre part educativa i de missatgeria.		Es va separar d'una manera més clara les dues parts, a més a més de donar la possibilitat d'anar a ambdues parts sense haver de passar per l'altre.
Càrrega per parts dels missatges d'un xat.		

Figura 6.1.2: Taules de recomanacions . Font: Pròpia

Les recomanacions que es troben apuntades a la taula, són recomanacions que en el seu moment vaig trobar interessants i que podrien millorar l'experiència d'usuari. Les que no vaig poder complir les afegeixo com a treball futur.

Com es pot veure vaig aconseguir reduir els possibles problemes mitjançant el testeig de forma individual, per tal de proveir l'aplicació en les millors condicions possibles.

6.2 Interfície d'usuari

Aquí mostrem el resultat final de les pàgines implementades a la interfície d'usuari. Només es trobaran les pàgines finals amb les màximes funcionalitats possibles, és a dir, les que es mostren quan un usuari té el rol d'usuari necessari per fer les màximes accions a aquella pàgina. Les funcionalitats com crear un grup, un projecte, un xat individual, etc. I la navegació es troba a la part de manual d'usuari de l'annex.

La següent interfície es igual en les dues plataformes, menys per la part de les sales de xat que a IOS les imatges es carreguen amb un petit thumbnail en comptes de la imatge completa.

AS^{UB}
FACULTAT D'EDUCACIÓ

Correu electrònic
ub@ub.edu

Contrasenya
Contrasenya

No tens compte? Crea una aquí

He oblidat la meua contrasenya

INICIAR SESSIÓ

Figura 6.2.1: Inici sessió. Font: Pròpia.

AS^{UB}
FACULTAT D'EDUCACIÓ

Nom i cognom
Nom i cognom

Correu electrònic
ub@ub.edu

Contrasenya
Contrasenya

Repeteix contrasenya
Repetix la contrasenya

He llegit i accepto els termes i condicions d'ús ☐

Tornar (Inici de sessió)

CREAR COMPTE

Figura 6.2.2: Crear compte. Font: Pròpia

Recuperar contrasenya

Correu electrònic
ub@ub.edu

Tornar a la pàgina principal

SOLICITAR REINICI DE CONTRASENYA

Figura 6.2.3: Recuperar contrasenya Font: Pròpia

Dashboard

TOTS PROJECTES INDIVIDUALS

Pol Collell	10/01/2021 16:17
Demo demo	18/12/2020 23:01
ASTE	11/12/2020 12:53
Anna	13/11/2020 10:59
INFO	Sense missatges

Figura 6.2.4: Dashboard. Font: Pròpia.

Facultat d'Educació

- Dashboard
- Projectes
- Fitxers
- Configuració
- Ajuda
- Tancar Sessió

Figura 6.2.5: Menú de l'aplicació. Font: Pròpia

Facultat d'Educació

Anna	Administrador principal
Demo demo	Alumne
Gerard Rouras	Administrador principal
Pol Collell	Alumne
Usuari Random	Alumne
Xènia Sánchez	Professor
Àngel Valdueza	Professor

Figura 6.2.6: Llista d'usuaris. Font: Pròpia

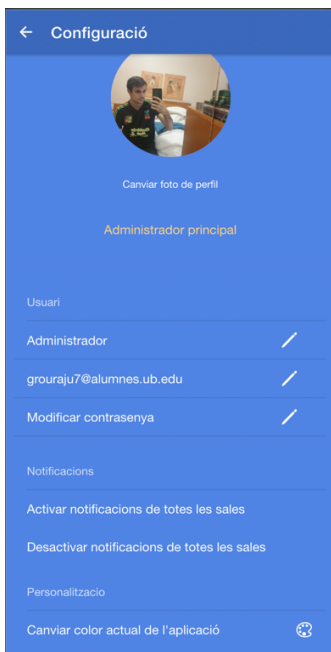


Figura 6.2.7: Configuració/Perfil. Font: Pròpia.

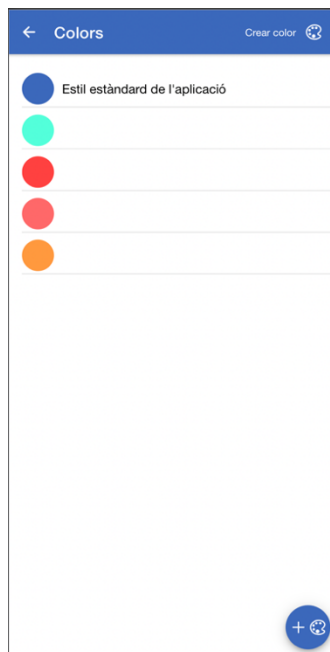


Figura 6.2.8: Selecció d'estils. Font: Pròpia.

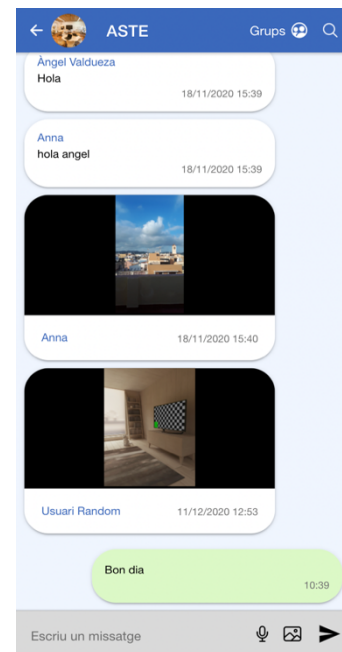


Figura 6.2.9: Sala individual o projecte. Font: Pròpia

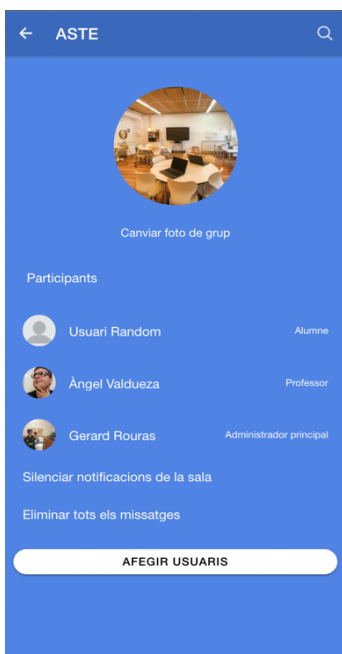


Figura 6.2.10: Informació xat/projecte. Font: Pròpia.



Figura 6.2.11: Modal Visualització imatge. Font: Pròpia.

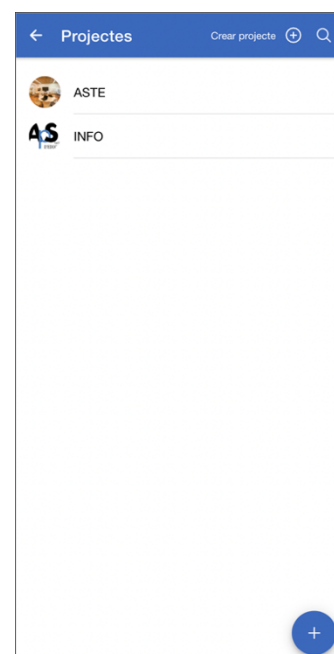


Figura 6.2.12: Llista de Projectes. Font: Pròpia



Figura 6.2.13: Llista de grups de treball. Font: Pròpia

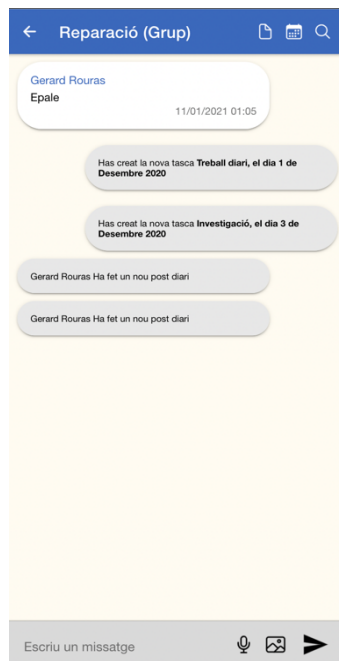


Figura 6.2.14: Sala del grup de treball. Font: Pròpia.

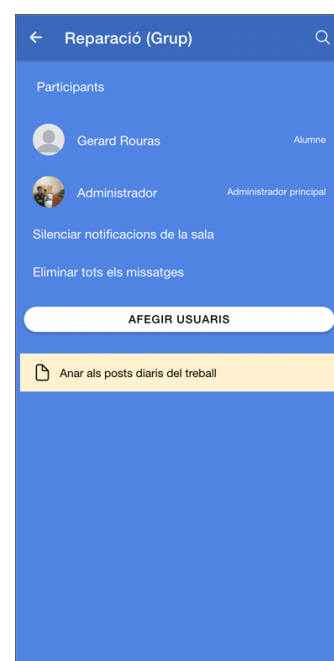


Figura 6.2.15: Informació grup de treball Font: Pròpia.



Figura 6.2.16: Part educativa dels grups . Font: Pròpia

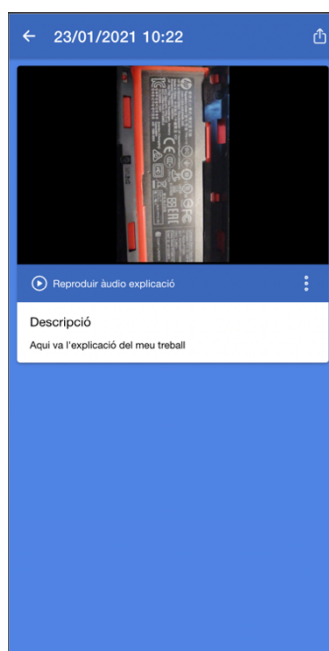


Figura 6.2.17: Visualització d'un post. Font: Pròpia



Figura 6.2.18: Calendari del grup. Font: Pròpia

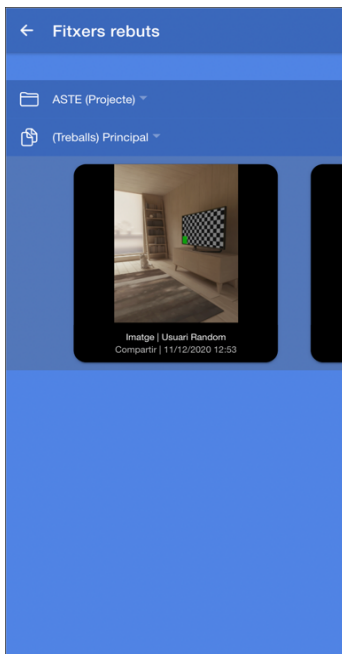


Figura 6.2.19: Visualització dels fitxers. Font: Pròpia

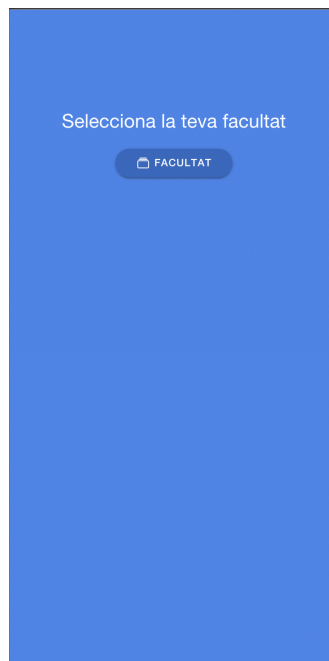


Figura 6.2.20: Inici de sessió per primer cop. Font: Pròpia

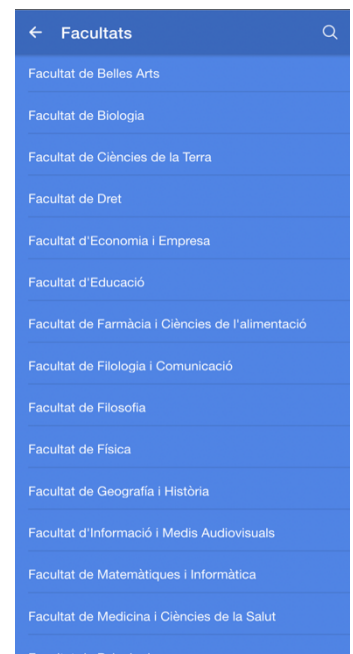


Figura 6.2.21: Selecció de facultat. Font: Pròpia.

Ajuda
Informació general
Pàgines
Alumnes
Professors/Entitats
Administradors

Informació general

L'aplicació consisteix en una eina de suport pel seguiment d'activitats a la secció d'aprenentatge de servei, a més a més de proporcionar facilitats per a la comunicació entre alumnes, professors i entitats mitjançant l'ús de xats i pàgines específiques.

Projectes

Integrants d'un projecte més general amb possibilitat de diversos alumnes, professors i entitats. Només fa la funció de xat i dona la possibilitat de crear grups de treball.

Grups de treball

Integrants d'un projecte separats en petits grups de treball (Alumne-Professor-Entitat) per al seguiment de l'alumne corresponent, aquí es poden fer posts i tasques al calendari. A més a més té la possibilitat de fer funció de xat.

Notificacions

Les notificacions són en local, és a dir, per poder rebre és necessari tenir la aplicació oberta, ja sigui utilitzant-la o tenir-la de fons sense tancar-la.

Figura 6.1.2: Pàgina d'ajuda externa. Font: Pròpia

7. CONCLUSIÓ

Partint dels objectius inicialment exposats, s'ha aconseguit implementar una primera versió de l'aplicació de manera que fos multi-plataforma amb la capacitat de poder suportar un gran nombre de dispositius, tot això amb un disseny intuïtiu, fàcil i ràpid d'entendre i d'utilitzar.

Per aconseguir fer aquest projecte ha estat necessari integrar i aprendre les noves tecnologies partint d'uns coneixements previs. Inicialment havia treballat amb Javascript, JQuery i Php, Html5 i Css, tots ells apresos entre les assignatures de Computació Orientada al Web, Factors Humans i Computació, Software Distribuït i Enginyeria del Software. Així doncs, com es pot veure, vaig poder aplicar certs coneixements de Html5 i Css, i el fet d'haver treballat amb Javascript em va ajudar a poder assolir amb més rapidesa Typescript, per la seva estructura similar. La resta de tecnologies i funcionalitats les vaig anar aprenent i integrant en el transcurs de l'elaboració del projecte, aconseguint acabar-lo amb èxit.

L'aplicació està distribuïda i està disponible per la secció d'aprenentatge servei per tal de distribuir-la als estudiants que utilitzin la plataforma d'Android, i començar a treballar amb ella, podent detectar llavors, errors que no s'havien trobat fins ara, a més a més de proveir un feedback per millorar-la.

7.1 Treballs futurs

Depenent dels resultats que s'obtinguin de la utilització durant els pròxims mesos i dels errors que puguin sorgir, s'haurà de modificar, afegir o eliminar certes coses de l'aplicació per tal de que compleixi totes les necessitats noves que apareguin. Així doncs, hauria d'haver un manteniment darrere per poder anar administrant el que vagi sorgint. A més a més, hi han implementacions que s'haurien volgut afegir però que no es va poder fer durant el desenvolupament i que podrien millorar encara més l'aplicació en termes d'optimització o usabilitat, aquestes poden ser:

- Canviar l'observació del tots els xats de la base de dades feta a la pàgina principal, per només els xats que un estigui com integrant, evitant així haver de llegir els canvis als xats que no són d'interès d'aquell usuari.
- Càrrega per parts dels xats, estil WhatsApp o Telegram, carregant només els últims 15 missatges que s'hagin enviat amb la possibilitat d'accedir als més antics pujant cap a dalt del xat. Aquesta funcionalitat milloraria el rendiment d'una sala que contingui molts missatges, ja que la càrrega es fa mitjançant un Array.
- Millora en la reproducció d'àudios, per tal de poder veure quant de temps li queda per acabar de reproduir, anar al temps que es vulgui o pausar sense que es reinici la reproducció a 0 quan es vulgui continuar.

S'hauria de llençar l'aplicació en les plataformes Android i IOS a fi que l'aplicació es pogués instal·lar en qualsevol moment des de les tendes d'aquestes dues, amb la possibilitat d'anar actualitzant l'aplicació cada cert temps en lloc de crear un enllaç cada cop que es faci un canvi i anar-lo passant a tothom. Per Android com es va veure en cost, la llicència és més reduïda de preu i només s'ha de pagar un cop, en IOS, per contra, sí que s'hauria d'invertir una mica més de diners.

8. BIBLIOGRAFIA

- [1] “Pàgina oficial Ionic, documentació i instal·lació” <https://ionicframework.com/>
- [2] “Pàgina oficial Node, instal·lació inicial” <https://nodejs.org/es/>
- [3] “Instal·lació Visual Studio” <https://code.visualstudio.com/>
- [4] “Pàgina oficial Cordova, documentació” <https://cordova.apache.org/docs/en/latest/>
- [5] “Ionic + Angular.js tutorials” <https://devdactic.com/>
- [6] “Ionic + Angular.js tutorials” <https://medium.com/>
- [7] “Repositori projecte i Repositori pàgina d’ajuda (privats)” <https://github.com/>
- [8] “Instal·lació de paquet de calendari auxiliar” <https://www.npmjs.com/package/ion2-calendar>
- [9] “Fòrum de solució de problemes de Ionic” <https://forum.ionicframework.com/>
- [10] “Pàgina oficial Android Studio, instal·lació” <https://developer.android.com/studio>
- [11] “Pàgina oficial XCode, instal·lació” <https://developer.apple.com/xcode/>

9. ANNEX

En aquest annex inclourem la documentació extra del projecte on trobarem un glossari i un manual d'ús de l'aplicació.

9.1 Glossari

- Framework: Un framework és un esquema o patró que ens ofereix l'entorn per el desenvolupament d'una aplicació
- Android: Sistema operatiu de Google per a dispositius mòbils,
- IOS: Sistema operatiu d'Apple per a dispositius mòbils de la marca.
- Android Studio: Entorn de desenvolupament d'aplicacions per la plataforma Android.
- XCode: Entorn de desenvolupament d'aplicacions per la plataforma IOS.
- Backend: S'encarrega d'administrar els processos necessaris perquè l'aplicació funcioni correctament.
- US: User Stories.
- User Stories: Històries relacionades amb els usuaris necessàries per una especificació més concreta dels requeriments.
- Sprint: Interval de temps durant el que es desenvolupa un increment de funcionalitats d'un producte o aplicació de forma que sigui utilitzable.
- JSON: JavaScript Object Notation, format de text per l'intercanvi de dades.
- Patró de disseny: Tècniques per resoldre problemes comuns durant el desenvolupament.
- Toast: Missatge que es mostra per pantalla durant uns segons.
- Date: Tipus de dades que s'utilitza per emmagatzemar una data amb format YYYY-mm-dd
- Base64: Grup d'esquemes de codificació de binari a text que es representen mitjançant una cadena ASCII.
- Testing: Proves sobre una aplicació de forma exhaustiva.

9.2 Manual d'ús

En aquest apartat podrem veure una guia d'ús de l'aplicació per part dels diferents usuaris.

9.2.1 Instal·lació

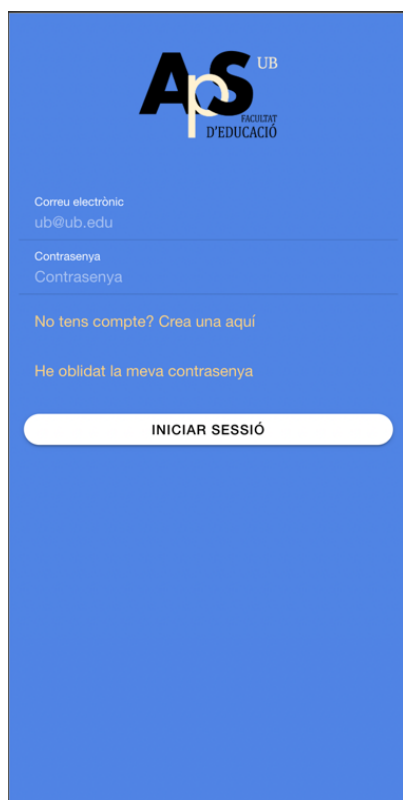
Per poder instal·lar l'aplicació, primerament s'ha de descarregar mitjançant el fitxer .Apk que s'ha proporcionat. Així doncs, és necessari que el dispositiu en el que es vulgui utilitzar l'aplicació sigui de la plataforma Android.

Un cop descarregada, si el dispositiu ens dona la possibilitat d'accedir al fitxer automàticament, l'obrim. En cas contrari s'haurà d'utilitzar un gestor de fitxers i accedir a la ubicació on es troba l'arxiu descarregat per poder obrir-lo i començar amb la instal·lació.

Un cop s'obri el fitxer, el dispositiu informará que l'aplicació no és de confiança o que és d'origen desconegut, això es deu a que és una aplicació que es troba fora de la Google Play. És possible que al mostrar el missatge no es pugui continuar amb la instal·lació, per habilitar-ho, doncs, s'ha de fer la següent acció: Configuració → Configuració avançada → Instal·lar aplicacions d'origen desconegut. Un cop començada la instal·lació hi ha la possibilitat de que Google informi que l'aplicació no pertany a la Google Play i es preguntarà si es vol continuar, en aquest cas s'ha de prémer el botó de acceptar. També existeix la possibilitat que s'hagi de desconnectar Google Play Protect, per fer això s'ha de fer la següent acció: Obrir Google Play → Play Protect Ajustos → Deshabilitar l'opció d'analitzar aplicacions amb Play Protect.

9.2.2 Guia general

Un cop instal·lada l'aplicació, en la següent guia explicarem els passos necessaris per poder utilitzar la interfície i les accions possibles depenent del rol d'usuari.



Aquesta és la pàgina inicial on es podrà iniciar sessió amb un compte creat amb anterioritat, en cas de que no es despongui d'una, s'ha de prémer el text on posa: No tens compte? Crea una aquí. En cas de que ens haguem oblidat de la nostra contrasenya, s'ha de prémer el text: He oblidat la meua contrasenya.

Figura 9.2.2.1: Inici de sessió. Font: Pròpia

AS^{UB}
FACULTAT
D'EDUCACIÓ

Nom i cognom
Nom i cognom

Correu electrònic
ub@ub.edu

Contrasenya
Contrasenya

Repeteix contrasenya
Repetix la contrasenya

He llegit i accepto els termes i condicions d'ús ☐

Tornar (Inici de sessió)

CREAR COMPTE

Trobem la pàgina de crear compte. Es necessari omplir tots els camps i acceptar els termes i condicions d'ús. S'ha de vigilar que el correu electrònic estigui correctament escrit i no s'hagi utilitzat per crear un altre compte. A més a més les contrasenyes han de coincidir entre elles. Si s'ha creat satisfactòriament s'haurà rebut un missatge per confirmar el compte des de la direcció: noreply@ubprojectes-716c8.firebaseio.com.

Figura 9.2.2.2: Crear compte. Font: Pròpia

Recuperar contrasenya

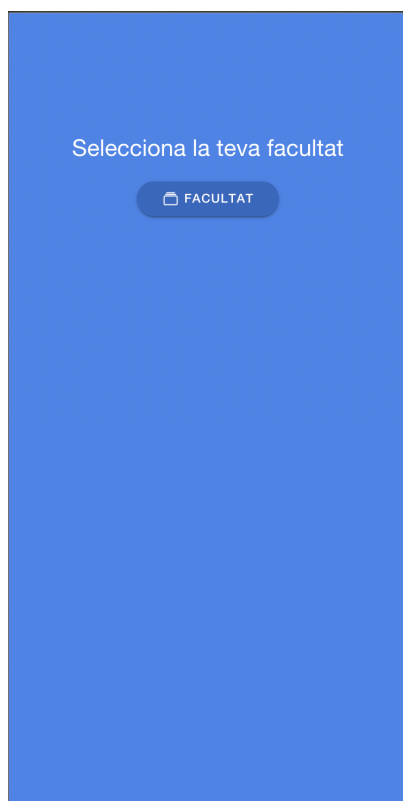
Correu electrònic
ub@ub.edu

Tornar a la pàgina principal

SOLICITAR REINICI DE CONTRASENYA

Si s'ha escollit recuperar contrasenya sortirà aquesta pàgina. Per dur a terme l'acció s'ha d'omplir el camp Correu electrònic amb el del compte desitjat i prémer el botó de sol·licitar reinici de contrasenya. Si el correu existeix, ens arribarà un correu amb l'adreça: noreply@ubprojectes-716c8.firebaseio.com que ens dirà si volem sol·licitar una de nova, si premem aquella opció ens enviarà a una pàgina pròpia de Google on haurem de posar una nova contrasenya i confirmar-la, i un cop fet ja es podrà accedir al compte amb la nova contrasenya.

Figura 9.2.2.3: Recuperar contrasenya. Font: Pròpia



Si és el primer cop que s'inicia sessió en l'aplicació sortirà aquesta pantalla on haurem de seleccionar la nostra facultat. Per fer-ho s'ha de prémer el botó facultat que ens mostrarà un llistat amb totes les facultats de la Universitat de Barcelona.

Figura 9.2.2.4: Seleccionar facultat. Font: Pròpia



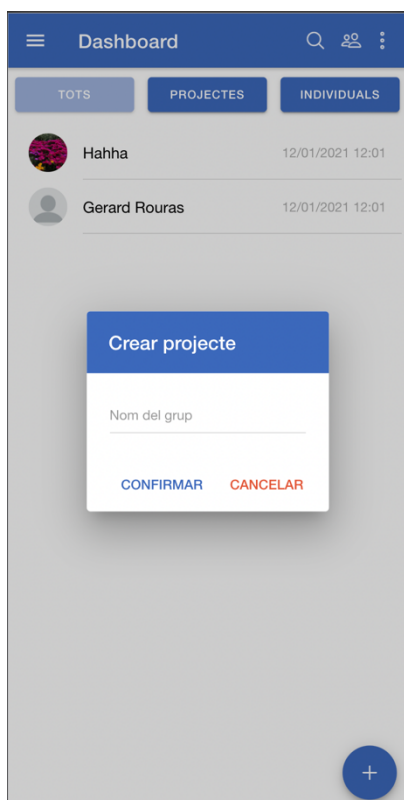
Aquesta pàgina surt cada cop que es vol seleccionar una facultat. Hi ha dos casos possibles: Quan es vol seleccionar una facultat per primer cop o quan un súper-administrador vol canviar la facultat d'un usuari. Per seleccionar la facultat desitjada només s'ha de prémer a aquella que es vulgui i sortirà un missatge preguntant si realment és la que es vol seleccionar. Hi ha l'opció de cercar la que es vulgui en cas de no es trobi mitjançant el botó de la part superior en forma de lupa, que obrirà un camp per escriure el text.

Figura 9.2.2.5: Llista de elecció de facultat. Font: Pròpia



En cas de que sigui la primera vegada que s'accedeix a l'aplicació tindrem el rol d'alumne, per tant haurem d'esperar a que un administrador assigni un altre rol en cas de necessitar-ho. Aquesta és la pàgina que es veu cada cop que s'inicia l'aplicació i com es pot veure no hi han projectes ni xats assignats al nostre usuari, per poder crear-ne haurem de ser administradors. En cas de ser súper-administrador o administrador es veuran tots els projectes creats a aquella facultat (no els xats individuals de la resta), en cas contrari només els projectes assignats, a més a més dels xats individuals propis, on prement sobre algun d'aquests redireccionarà cap a la sala de xat d'aquell. Des d'aquesta pàgina es pot anar al menú principal de l'aplicació mitjançant la icona de la part superior esquerra o lliscant la part esquerra de la pantalla cap a la dreta, cercar un projecte o xat amb la icona de la lupa, veure la llista d'usuaris de la facultat o dels grups de treball amb la icona de les dues persones o obrir un petit menú amb diverses opcions mitjançant els tres punts.

Figura 9.2.2.6: Dashboard. Font: Pròpia



Aquesta és la pàgina que es mostra en cas de ser administradors o súper-administradors, només canvia respecte a la de la resta de rols en que es pot crear projectes i eliminar-los. Per crear projectes s'ha de prémer el botó inferior dret amb el símbol de més o des del menú obert des de la icona superior dels tres punts. En cas de voler eliminar un xat (professor, entitat) o projecte (administrador – súper-administrador) s'ha de lliscar cap a la dreta l'element que es vulgui eliminar i ens sortirà l'opció; un cop s'accedeixi ens preguntarà si volem eliminar aquella sala definitivament, incloent tots els grups de treballs i fitxers enllaçats.

En aquest menú a més a més hi han altres opcions com canviar la facultat actual (En cas de súper-administrador), anar a configuració o tornar a carregar els xats. En cas de tenir el rol de professor, entitat o alumne, no hi han les dues primeres opcions.

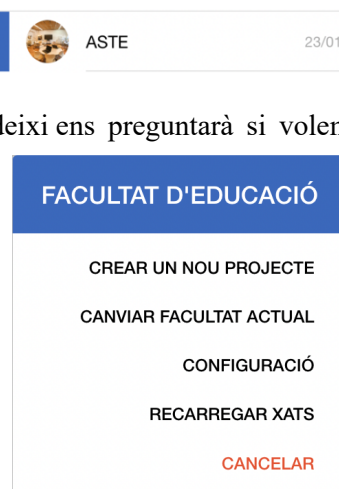
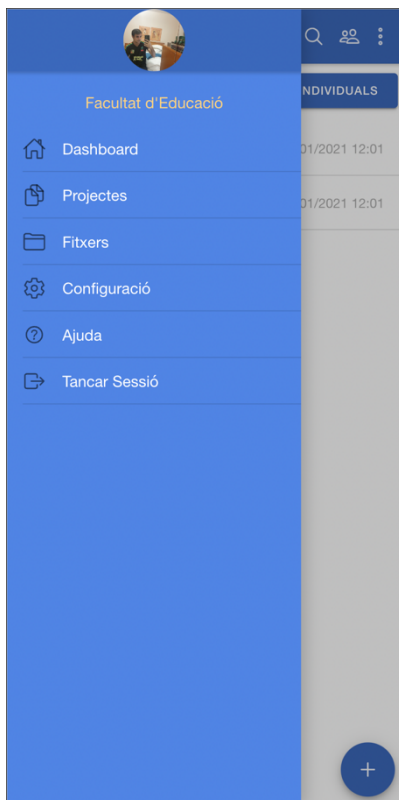
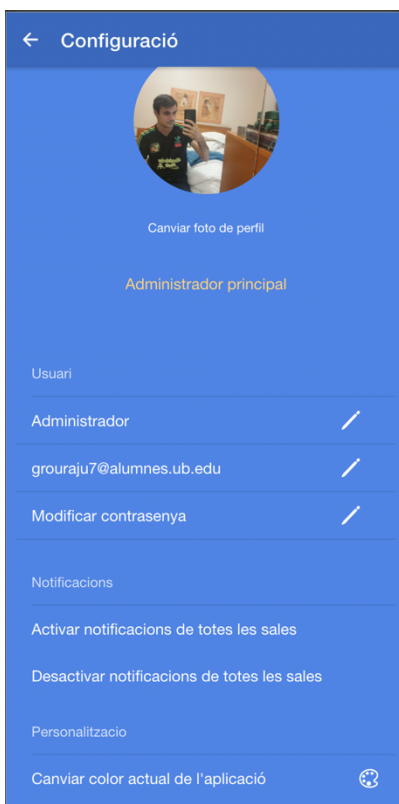


Figura 9.2.2.7: Dashboard + modal. Font: Pròpia



Aquest es el menú principal de l'aplicació, des d'aquí podem anar a diferents llocs, com la pantalla principal Dashboard, veure els nostres projectes assignats, el fitxers rebuts i enviats en les diferents sales de xat, configuració, anar a la pàgina externa d'ajuda o tancar la sessió que hi ha activa.

Figura 9.2.2.8: Menú principal. Font: Pròpia



En la pàgina de configuració veurem totes les nostres dades afegides alhora de la creació de compte i ens donarà l'opció de poder modificar-les. Per fer això s'ha de prémer a sobre del camp que es vulgui modificar o a la icona del llapis, un cop s'hagi fet s'obrirà un camp per escriure la dada modificada i per acceptar s'haurà de prémer l'icona amb el símbol de guardar que hi ha al costat dret. En cas de no voler canviar aquella dada es pot cancel·lar prement el botó de cancel·lar edició o la icona amb el símbol d'una creu. Si es vol modificar el correu electrònic o la contrasenya s'haurà de confirmar la contrasenya actual per dur aquella acció, això està pensat com a mètode de seguretat. A més a més tenim l'opció de desactivar i activar les notifikacions de totes les sales de xat que tinguem assignades i canviar el color actual de l'estil de l'aplicació

Usuari	
Gerard Rouras	/
grouaju7@alumnes.ub.edu	/
Nova contrasenya	
Nova contrasenya	o x
CANCEL·LAR EDICIÓ	

Figura 9.2.2.9: Configuració/Perfil. Font: Pròpia



Si es vol canviar l'estil es pot seleccionar un altre color de la llista, un cop es seleccioni s'aplicarà automàticament com es veu a la imatge de la pàgina. Les opcions de crear color només es trobaran disponibles per el súper-administrador, per tant el botó de la part inferior dreta i el text de la part superior dreta no es veuran en qualsevol altre rol. Per crear un nou color s'obrirà un formulari on s'haurà de seleccionar un color per cada tipus d'element de l'aplicació, aquest color pot ser en hexadecimal o seleccionat des de el selector de colors que s'obrirà. Un cop seleccionats tots els colors s'ha de prémer enviar un nou estil o cancel·lar edició.

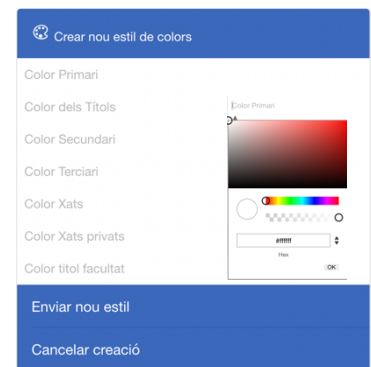
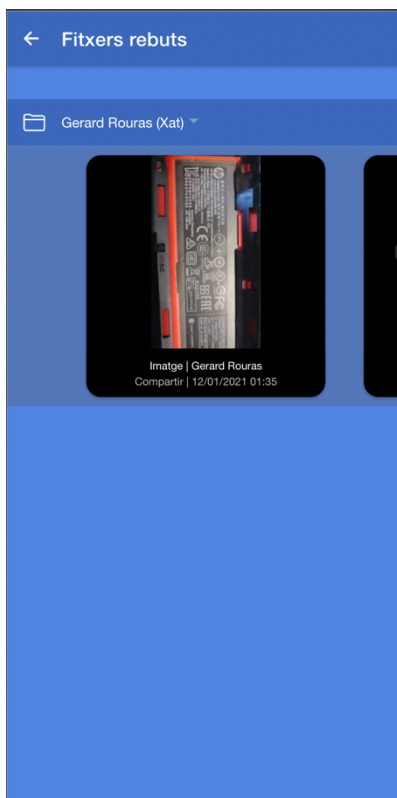


Figura 9.2.2.10: Selecció d'estil. Font: Pròpia



A la pàgina de fitxers rebuts podrem veure tots els fitxers de les sales de xat assignades com es pot veure a la imatge de la pàgina. Aquests fitxers estan ordenats per data d'enviament i tenim l'opció de visualitzar la imatge o reproduir/parar un àudio polsant a sobre del fitxer. També hi ha la possibilitat de compartir el fitxer prement al text inferior Compartir. Per seleccionar un xat, projecte o grup de treball s'ha de prémer a sobre d'on posa conservacions, en cas de la imatge ja està seleccionat un xat.



Figura 9.2.2.11: Pàgina fitxers. Font: Pròpia



La següent pàgina és per la visualització d'imatges en pantalla completa, s'activa un cop es prem sobre qualsevol imatge d'un xat o post i ens dona l'opció de compartir-la o d'ampliar-la. Per tancar només és necessari prémer el text Tornar.

Figura 9.2.2.12: Modal visualització imatge. Font: Pròpia

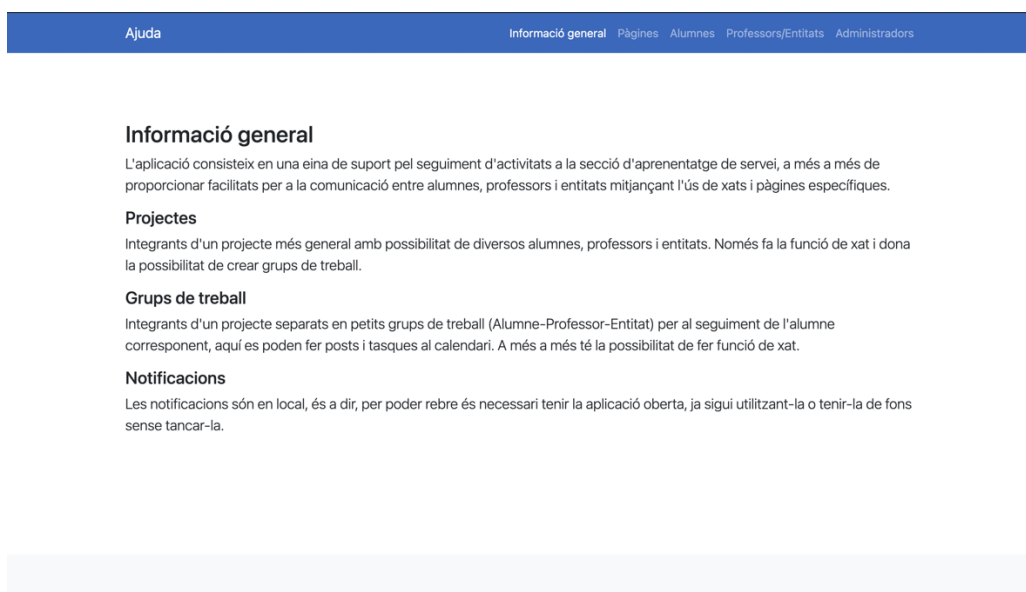


Figura 9.2.2.13: Pàgina d'ajuda. Font: Pròpia

La pàgina d'ajuda es troba de forma externa en servidor de github per tal de poder modificar-la. Mitjançant els botons de la part superior ens podem moure per aquesta, o directament baixant manualment la pàgina trobarem els diferents continguts.



La següent pàgina es mostra seleccionant al menú l'opció de projectes. En aquesta es visualitzen tots els projectes assignats o tots els projectes de la facultat en cas de ser administrador o súper-administrador. En cas de tenir un dels últims dos rols hi haurà la possibilitat de crear un projecte de la mateixa manera que des del dashboard, mitjançant l'ús les eines típiques, el botó amb el símbol del més o des de crear projecte de la barra superior. Per eliminar-lo s'ha de fer lliscant l'element cap a la dreta. A diferència de dashboard si es selecciona un projecte no redirigirà cap a la sala de xat, sinó cap els grups de treballs. Això està fet per crear un nou camí evitant haver de passar primer pel xat, així doncs, prioritant l'apartat educatiu. Es pot cercar un projecte mitjançant l'icona de la lupa. Per poder afegir usuaris a un projecte ha de ser des de la sala de xat del projecte.

Figura 9.2.2.14: Llista de projectes. Font: Pròpia



Aquí tenim la pàgina de grups de treballs, en aquesta pàgina es veuran tots els grups de treball d'un projecte i en cas de ser alumne només es veuran els grups de treball assignats. A diferència de les altres accions administratives, en aquest cas els professors tenen l'opció de crear i eliminar grups de treball. L'eliminació es fa amb l'acció de lliscar cap a la dreta. Aquesta pàgina s'obre si es prem un projecte des de la pàgina projectes o si es prem l'opció de grups des un xat de projecte, en cas de que sigui lo primer, si es selecciona un grup es redireccionarà a la pàgina educativa on es poden crear les actualitzacions i es troba el calendari, de maners que es pugui seguir el desenvolupament. Es pot cercar un grup de treball mitjançant l'icona de la lupa. Per poder afegir usuaris a un grup de treball ha de ser des de la sala de xat del grup de treball.

Figura 9.2.2.15: Llista de grups de treball. Font: Pròpia

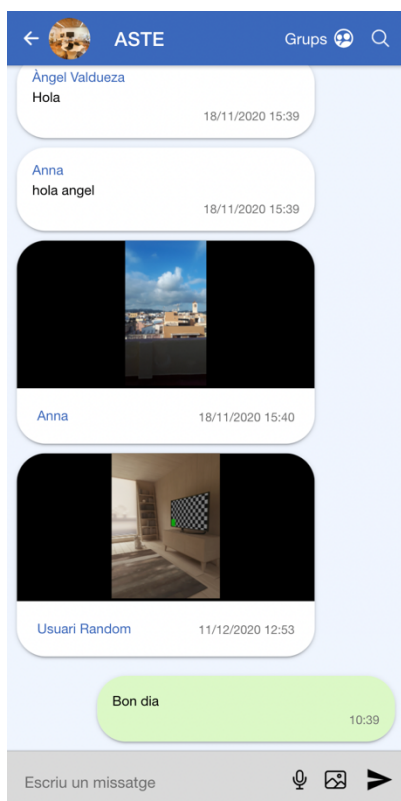


Figura 9.2.2.16: Sala de xat d'un projecte. Font: Pròpia

Aquesta és la pàgina de la sala de xat individual o de projecte, en aquesta imatge es tracta de l'últim. Existeix opció d'enviar un missatge mitjançant el camp de la barra inferior on es diu: Escriu un missatge, on un cop s'hagi escrit el que es vol, prement la icona de la fletxa negra s'enviarà a la resta de participants. També hi ha la possibilitat d'afegir un àudio o una imatge utilitzant els icones de micròfon i imatge. El micròfon que s'utilitzarà serà el propi del dispositiu, per les imatges hi ha l'opció de fer una fotografia o agafar-la de la galeria. Si es vol eliminar un missatge enviat per un mateix, s'ha de prémer dos cops sobre el missatge i ens donarà l'opció. En cas de que es vulgui anar als grups de treball s'ha de prémer el text on posa grups de la part superior i si es vol anar a la informació del xat o projecte s'ha de prémer el nom del xat.

Si s'està fora de la posició final dels missatges del xat, en cas de que arribés un missatge sortirà un botó informant que hi ha un nou missatge, i si es prem es baixarà automàticament als últims missatges.

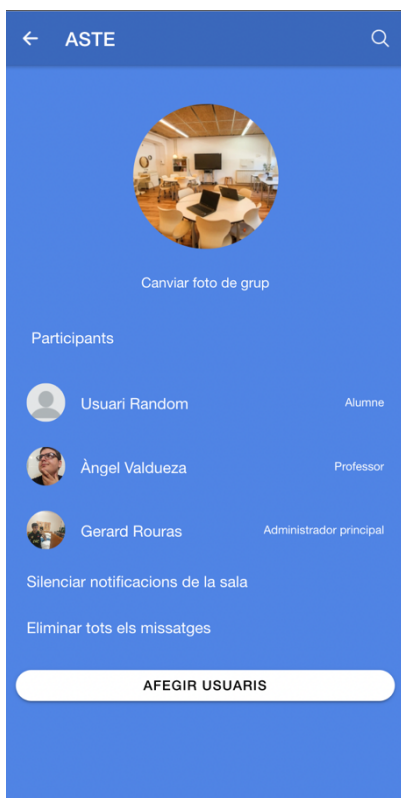


Figura 9.2.2.17: Informació d'un projecte. Font: Pròpia

Ara trobem la pàgina d'informació d'un projecte, en aquesta hi ha l'opció d'afegir usuaris utilitzant el botó afegir usuaris o eliminar-los llistant cap a la dreta l'usuari que es vol, tot això sent súper-administrador o administrador. També es pot cercar un participant. Com a usuari participant es poden silenciar les notifikacions d'aquella sala utilitzant el text Silenciar notifikacions de la sala o Activar notifikacions de la sala. En cas de ser súper-administrador hi haurà la possibilitat d'eliminar tots els missatges d'aquell xat.

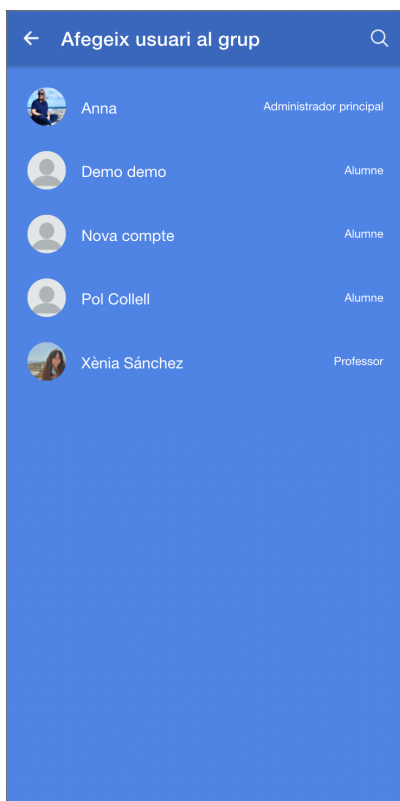


Figura 9.2.2.18: Llista d'usuaris per afegir. Font: Pròpia

La pàgina per afegir un usuari s'obrirà un cop es seleccioni l'opció d'afegir un usuari en un grup de treball o projecte. En cas de que sigui un projecte només es mostraran els usuaris de la facultat que no han estat inclosos en el projecte, en canvi, si es vol afegir un usuari a un grup de treball ens mostrarà els participants que es troben en el projecte superior però que no es troben inclosos en el grup de treball. Per poder afegir un usuari només és necessari seleccionar el que es vol. Hi ha la possibilitat de cercar l'usuari mitjançant l'icona de la lupa.

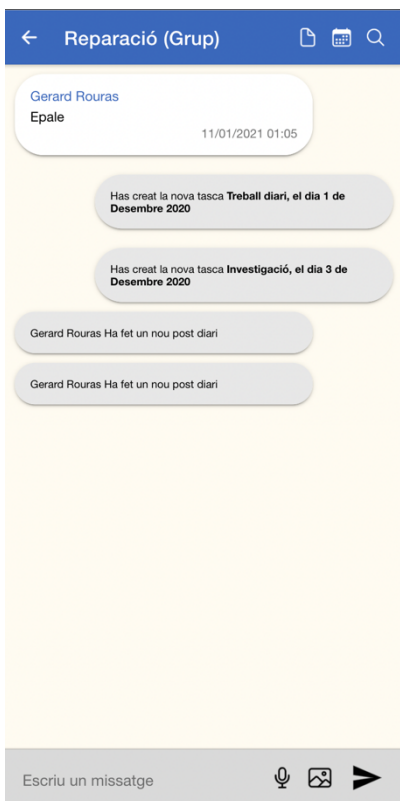
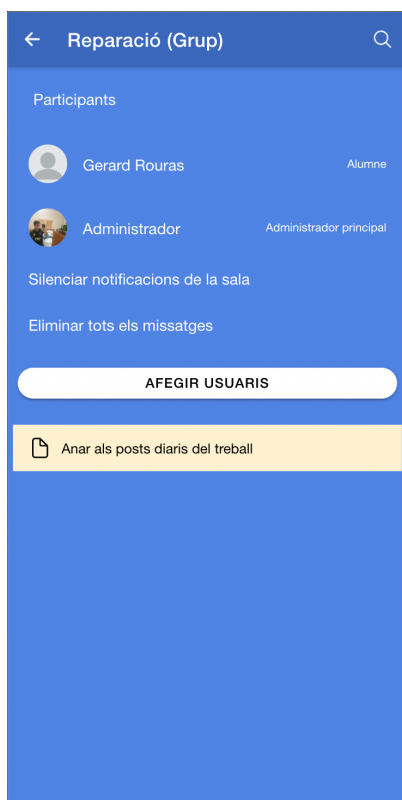


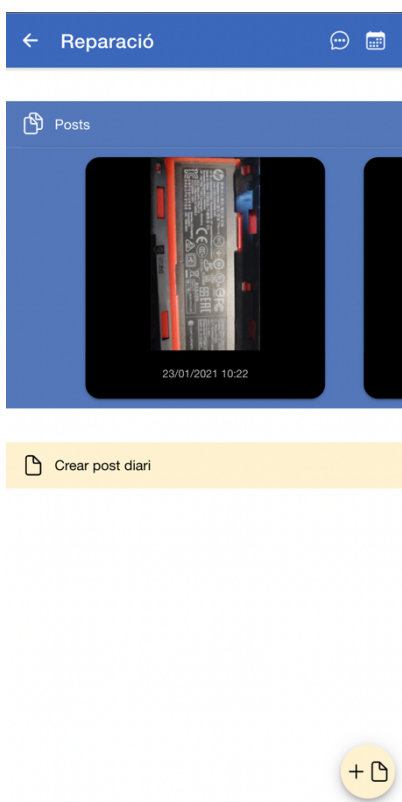
Figura 9.2.2.19: Sala de xat d'un grup de treball. Font: Pròpia

Aquesta pàgina conté la sala de xat d'un grup de treball, aquí s'enviaran i es rebran els missatges de notificació de creació d'un post diari per part de l'alumne o d'una tasca al calendari per part d'un professor o una entitat, si es prem alguna d'aquests missatges es redigirà cap al post creat o cap el calendari. Té les mateixes funcionalitats que la sala de xat individual o de projecte. Es pot anar cap la pàgina educativa on es troben els posts des de la icona amb un paper doblegat en la barra superior. També es pot anar al calendari del grup de treball des de l'icona del calendari. Hi ha la possibilitat de cercar missatges mitjançant la lupa de la barra.



Aquesta és la pàgina d'informació d'un grup de treball, és similar a la d'un projecte i es diferencia en que no es pot afegir o canviar la imatge de perfil de grup. Mitjançant el botó Anar als posts diaris del treball podem anar cap a la part educativa.

Figura 9.2.2.20: Informació d'un grup de treball. Font: Pròpia



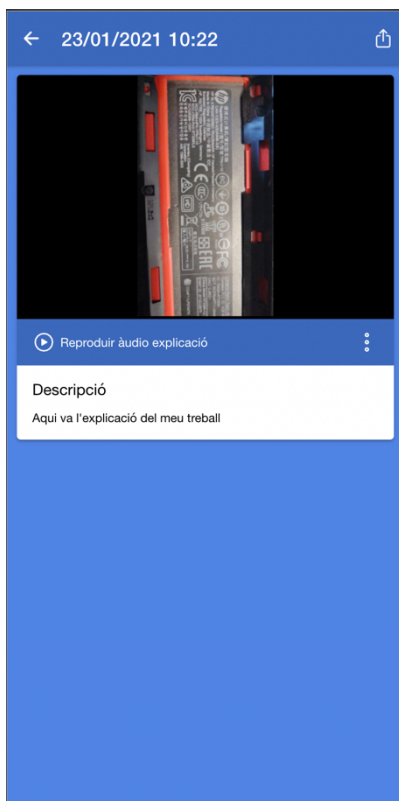
Aquí trobem la part educativa del grup de treball, es poden veure les diferents actualitzacions creades a partir d'un post, si es selecciona alguna ens redirigeix cap al post complet. En cas de que l'usuari sigui un alumne li sortiran les opcions d'afegir un post diari, aquestes són: el botó de la part inferior amb un més i un paper doblegat i el botó de crear post diari.

En cas de que s'hagi seleccionat l'opció de crear un post, s'obrirà un formulari on s'haurà d'afegir una imatge, un àudio i un text per explicar que s'ha fet aquell dia en l'horari marcat. Es pot enviar o cancel·lar aquell post. A més a més si s'ha afegit una imatge o àudio incorrecte es podrà eliminar sense haver-ho d'eliminar tot

Crear post diari
Afegir imatge
Afegir àudio
Explica el que has fet avui
Enviar post diari
Cancelar post diari

Des de aquesta pàgina es pot anar cap al xat del grup de treball mitjançant l'icona d'un xat a la barra superior i anar cap al calendari mitjançant l'icona de calendari

Figura 9.2.2.21: Secció educativa. Font: Pròpia



La pàgina següent mostra el contingut complet d'un post creat per un usuari, en cas de que l'usuari sigui un alumne hi haurà l'opció d'eliminar aquell post. També existeix la possibilitat de compartir la imatge, àudio o text del post utilitzant l'icona de la barra superior.

Figura 9.2.2.22: Post. Font: Pròpia



Aquí trobem el calendari dels grup de treball, on en cas de que l'usuari sigui una entitat o un professor tindrà l'opció de crear una tasca prement el boto inferior dret amb el símbol d'un més i un rellotge o des de la barra superior, a més a més es pot eliminar la tasca prement dos cops l'element que es vulgui. Per crear una tasca s'ha de seleccionar primer el dia que volem i després prémer el botó o text d'afegir tasca. Ens sortirà un formulari on s'haurà de posar una descripció de la tasca que ha de fer l'alumne i un rang d'hores amb un format correcte.

Crear tasca

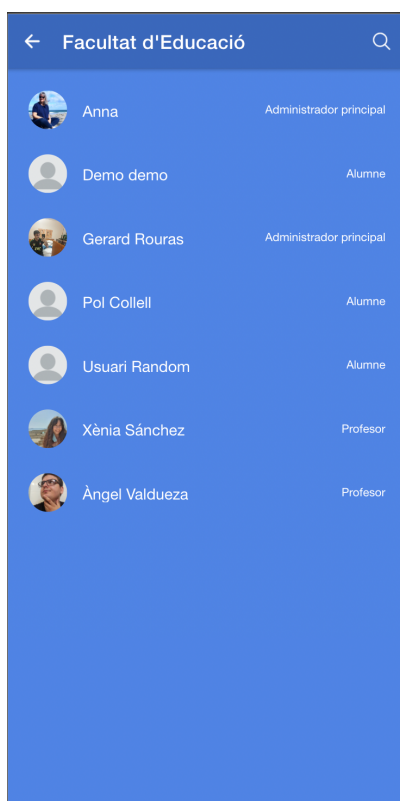
Títol de la tasca

Inici 00:00

Final: 00:00

CONFIRMAR
CANCELAR

Figura 9.2.2.23: Calendari. Font: Pròpia



Aquesta pàgina s'obre des de la dashboard, i mostra tots els usuaris de la facultat en cas de no ser un Alumne, en cas contrari mostrarà només els participants dels grups de treball on l'alumne n'és participant. En cas de súper-administrador es podrà canviar la facultat o rol d'un usuari lliscant cap a la dreta, en cas d'administrador només es pot canviar el rol d'un usuari. Hi ha la possibilitat de cercar un usuari mitjançant la lupa de la barra superior.

Figura 9.2.2.24: Llista d'usuaris. Font: Pròpia

Afegits

Com a afegits incorporem les notificacions, aquestes dependran de cada dispositiu mòbil, però el funcionament és el mateix, només arribaran les notificacions de les sales en les quals un no es trobi dins en aquell instant o que tingui silenciades. No es mostraran amb l'aplicació tancada, per tant, s'ha de tenir com a mínim de fons. Un cop surti una notificació, si es prem a sobre es redirigirà cap a la sala de xat d'on provingui la notificació, ja sigui individual, projecte o grup de treball.