

# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Grau en Enginyeria Informàtica**

## **DESENVOLUPAMENT D'UNA APP PER A GESTIONAR LA COMUNICACIÓ D'UNA ORGANITZACIÓ**

**Memòria**

**SERGI TUBERT  
PONENT: ALFONS PALACIOS**

PRIMAVERA 2019



TecnoCampus  
Mataró-Maresme







## Resum

L'objectiu d'aquest projecte consisteix en desenvolupar una plataforma que permet millorar els sistemes actuals d'intercanvi d'informació dins d'organitzacions, deixant de banda així les aplicacions de tipus xat i els butlletins informatius en paper. A més a més, ofereix un sistema de votació per ajudar a la presa de decisions dels usuaris.

Per a fer-ho es desenvolupa una aplicació web progressiva (PWA), una solució multiplataforma amb la finalitat de cobrir les necessitats del màxim nombre d'usuaris.

## Resumen

El objetivo de este proyecto consiste en desarrollar una plataforma que permite mejorar los sistemas actuales de intercambio de información dentro de organizaciones, dejando de lado las aplicaciones tipo chat y los boletines informativos en papel. Además, ofrece un sistema de votación para ayudar en la toma de decisiones de los usuarios.

Para hacerlo se desarrolla una aplicación web progresiva (PWA), una solución multiplataforma con la finalidad de dar cobertura a las necesidades del máximo número de usuarios.

## Abstract

The main goal of this project is to develop a platform to improve the actual sharing information systems inside an organization, setting aside the chat-like applications and leaving the paper-based bulletins. In addition, it offers a voting system to help the users take decisions.

To achieve it, it develops a progressive web application (PWA) as a multiplatform solution in order to cover the needs of the maximum number of users.









# Índex

Índex.....	I
Índex de figures.....	III
Glossari .....	V
1. Introducció i contextualització .....	1
1.1. Estructura del document.....	1
1.2. Actors implicats.....	1
1.3. Context actual.....	2
1.4. Proposta.....	2
2. Abast .....	5
2.1. Objectius .....	5
2.2. Metodologia .....	5
2.2.1. Divisió de fases .....	5
2.2.2. Control de versions i repositori .....	8
3. Planificació.....	9
3.1. Descripció de fases.....	9
3.2. Recursos i gestió econòmica .....	10
3.2.1. Recursos i costos humans.....	11
3.2.2. Recursos i costos de hardware.....	12
3.2.3. Recursos i costos de software.....	12
4. Requeriments i anàlisi del sistema d'informació .....	15
4.1. Captura de requeriments.....	15
4.2. Regles de negoci.....	16
4.3. Casos d'ús .....	16
4.4. Wireframing .....	18
4.5. Estudi dels sistemes de votació .....	18
5. Implementació: Base de dades .....	21
5.1. Sistema gestor de base dades .....	21
5.2. Disseny del model de dades .....	23
5.3. Definició de les taules .....	23
6. Implementació: API RESTful .....	29
6.1. Entorn de desenvolupament .....	30
6.2. Seguretat.....	32
6.3. Middlewares.....	35
6.4. Endpoints de la API.....	36

7.	Implementació: Aplicació .....	45
7.1.	Estudi d'opcions .....	45
7.2.	Elecció.....	48
7.3.	Entorn de desenvolupament .....	48
7.4.	Dos conceptes claus de React.....	49
7.5.	Llibreries de suport principals.....	49
7.6.	Funcionalitats de l'aplicació.....	51
8.	Funcionalitats PWA .....	67
8.1.	Manifest.....	67
8.2.	Javascript i service workers.....	68
8.2.1.	Caching i suport offline .....	69
8.2.2.	Notificacions push.....	69
9.	Servidor i hosting .....	71
9.1.	Configuració prèvia del servidor .....	71
9.2.	Instal·lació i configuració dels serveis .....	73
10.	Planificació final.....	75
10.1.	Desviació temporal.....	75
10.2.	Imprevistos .....	75
10.2.1.	Suport pels dispositius Apple .....	75
10.2.2.	Descàrrega d'arxius .....	76
11.	Sostenibilitat .....	77
11.1.	Econòmica.....	77
11.2.	Mediambiental.....	77
12.	Lleis i regulacions .....	79
13.	Conclusions .....	81
13.1.	Conclusions personals .....	81
14.	Bibliografia.....	83
	Annex I: Fitxes de casos d'ús.....	85
	Annex II: Wireframes.....	99
	Annex III: Contingut del CD-ROM .....	103

## Índex de figures

Figura 3.1 Taula de costos de recursos humans .....	12
Figura 3.2 Taula de costos materials .....	12
Figura 4.1 Diagrama de casos d'ús del membre.....	17
Figura 4.2 Diagrama de casos d'ús del gestor .....	17
Figura 4.3 diagrama de casos d'ús de l'administrador .....	18
Figura 5.1 Model de base de dades.....	22
Figura 6.1 Arquitectura tradicional client-servidor .....	33
Figura 6.2 arquitectura client-restful api .....	33
Figura 6.3 habilitar el permís d'accés a peticions d'origen extern.....	34
Figura 6.4 verificació del token d'autorització.....	35
Figura 7.1 Flux d'una aplicació Redux .....	50
Figura 7.2 Aplicació: Pantalla d'inici de sessió .....	51
Figura 7.3 Llistat de missatges .....	52
Figura 7.4 Llistat sense missatges .....	53
Figura 7.5 Detall d'un missatge.....	54
Figura 7.6 Enviament de missatge.....	55
Figura 7.7 llistat d'arxius.....	56
Figura 7.8 Llistat de votacions .....	57
Figura 7.9 Creació d'una votació .....	58
Figura 7.10 Realització d'una votació.....	59
Figura 7.11 Veure el resultat d'una votació .....	60
Figura 7.12 Menú lateral .....	61
Figura 7.13 llistat de gestió de grups.....	62
Figura 7.14 Creació d'un grup.....	63
Figura 7.15 Afegir USUARIS/GRUPS A GRUPS .....	64
Figura 7.16 llistat d'usuaris .....	64
Figura 7.17 Creació d'un usuari .....	65
Figura 9.1 configuració del firewall .....	71
Figura 9.2 configuració de registres .....	72
Figura 9.3 servidors de noms.....	72
Figura 9.4 arquitectura client-servidor .....	73



## Glossari

<b>Target</b>	Fa referència als destinataris al que pretén arribar un servei o un producte.
<b>Stack</b>	Conjunt d'eines de software que funcionen conjuntament formant una plataforma integrada.
<b>Sprint</b>	És el període de temps en el que es du a terme una tasca.
<b>Framework</b>	Entorn o marc de treball. És un conjunt estandarditzat de pràctiques per enfocar i resoldre una problemàtica en particular.
<b>Endpoint</b>	En serveis web, és cada URL a través de la qual pots accedir a diferents funcionalitats.
<b>Token</b>	Fitxa que serveix per identificar i autenticar un usuari.



# 1. Introducció i contextualització

Aquest projecte és un Treball de Fi de Grau (TFG) del Grau en Enginyeria Informàtica de l'Escola Superior Politècnica del Tecnocampus Mataró-Maresme.

## 1.1. Estructura del document

La memòria d'aquest projecte està dividida en quinze capítols i sis annexos. El primer capítol tracta d'introduir la temàtica i contextualitzar el projecte. El capítol 2 defineix els objectius del projecte, així com la metodologia que s'ha seguit. El capítol 3 tracta la planificació inicial, descrivint les fases i estudiant els recursos necessaris per dur a terme el treball. En el capítol 4 s'introdueixen els requeriments capturats i tot l'estudi previ al desenvolupament, es defineixen els casos d'ús i els esbossos de la interfície. El capítol 5 resumeix com es du a terme la persistència de les dades en aquest projecte. El capítol 6 explica com s'ha implementat l'API RESTful i quines mesures de seguretat s'han pres. El capítol 7 tracta com s'han desenvolupat les funcionalitats de l'aplicació web. El capítol 8 parla sobre les funcionalitats que ofereixen les aplicacions web progressives i com s'han aplicat en el projecte. El capítol 9 defineix on i com s'allotja l'aplicació. El capítol 10 resumeix els imprevistos a la planificació inicial. El capítol 11 tracta sobre la sostenibilitat econòmica i mediambiental d'aquest projecte i el capítol 12 tracta sobre les lleis i regulacions que l'afecten. El capítol 13 reuneix les conclusions. Per últim, el capítol 14 recull les fonts consultades en una bibliografia.

A continuació, l'annex 1 recull les fitxes de cas d'ús definides. L'annex 2 reuneix els esbossos de la interfície que s'han dissenyat. L'annex 3 defineix el contingut del CD-ROM.

## 1.2. Actors implicats

Com en qualsevol projecte que es realitza, s'impliquen directe o indirectament una sèrie de persones.

- Desenvolupador

El desenvolupador i, en aquest cas, també l'autor d'aquest projecte és l'actor principal. És l'encarregat tant d'implementar la solució software com de redactar i preparar tota la documentació relacionada amb el projecte.

- Tutor del projecte

El tutor del projecte és l'Alfons Palacios, Coordinador del Grau en Enginyeria Informàtica del Tecnocampus de Mataró. La seva responsabilitat recau en assessorar i donar suport a l'autor del projecte durant totes les fases del projecte, així com dur a terme un seguiment continu amb l'objectiu d'assolir l'objectiu final.

- Usuaris finals

Qualsevol usuari que contracti aquest servei pot beneficiar-se d'ell. El *target* objectiu al que es vol arribar són organitzacions de persones que necessitin una eina especialitzada en gestionar la comunicació interna.

### 1.3. Context actual

Actualment l'intercanvi d'informació està a l'ordre del dia, se'n mouen grans quantitats. Un dels principals problemes dins les organitzacions és que aquesta informació no està centralitzada i pot no arribar de la manera més adient a tots els usuaris.

### 1.4. Proposta

Tenint en compte la problemàtica actual esmentada en el punt anterior es vol desenvolupar una solució software en forma d'aplicació per tal de facilitar i garantir la correcta comunicació dins d'un col·lectiu, i tenir la informació a l'abast en qualsevol moment i en qualsevol lloc.

Per tal de garantir la correcta rebuda d'informació es proposa una solució que pugui controlar la correcta rebuda de la informació a tots els integrants dels grups i elimini la comunicació en paper.

Amb l'objectiu d'ajudar a les organitzacions a decidir sobre temes de propi interès també es proposa un sistema de votació segur pels integrants dels grups a través d'enquestes personalitzades. Tots els usuaris podran consultar els resultats d'aquestes votacions en qualsevol moment.



Per últim, amb l'objectiu d'arribar al màxim públic possible es vol adoptar una solució multiplataforma, amb el que donar suport tant per dispositius iOS com per dispositius Android.



## 2. Abast

### 2.1. Objectius

L'objectiu d'aquest projecte consisteix en dissenyar i implementar una solució software per ajudar a gestionar i automatitzar la comunicació dins d'una organització.

Aquest servei pretén que els usuaris finals, de manera senzilla i directa, puguin estar informats de manera eficient, així com intercanviar documents i arribar a acords grupals a través d'un sistema de votació.

No es vol enfocar aquest servei com una aplicació de missatgeria instantània, sinó com una font de consulta i intercanvi d'informació centralitzada dins d'un grup exclusiu.

Els objectius principals del projecte són:

- Analitzar la problemàtica de la comunicació actual dins d'una organització.
- Desenvolupar un servei en forma d'aplicació multiplataforma.
- Ha de permetre saber amb seguretat que la informació ha estat rebuda correctament pels integrants de l'organització.

### 2.2. Metodologia

Per dur a terme aquest projecte i tenir una visió global prèvia, s'ha decidit dividir el projecte en cinc fases: fase inicial, fase d'anàlisi i especificació, fase de disseny i implementació, fase de desplegament i proves i fase de documentació.

#### 2.2.1. Divisió de fases

##### **Fase inicial**

S'ha realitzat un estudi previ sobre l'estat d'art tecnològic i serveis similars ja existents. S'ha de reconèixer la problemàtica actual i s'han de definir els objectius, les bases del projecte i capturar els requeriments necessaris de l'aplicació. És molt important definir amb claredat tots i cada un dels requeriments per evitar confusions i incongruències durant les fases conseqüents.

## **Fase d'anàlisi i especificació**

Un cop finalitzada la primera fase, s'han analitzat els requeriments adquirits durant la fase inicial, tant els requeriments funcionals com les especificacions tècniques. Durant aquesta fase s'ha decidit quin *stack* tecnològic s'utilitzarà, tant pel desenvolupament de la interfície de l'aplicació com pel desenvolupament de la API.

S'han definit els casos d'ús per establir els comportaments de l'aplicació, descrivint els processos, els actors, els sistemes i les interaccions entre ells. Juntament amb les regles de negoci s'han completat els requisits previs necessaris. També ha estat important prototipar les vistes de l'aplicació tenint en compte estàndards d'usabilitat i accessibilitat.

Durant aquesta fase, s'ha pres un temps per a la formació en les tecnologies que s'han decidit utilitzar; així com dur a terme la configuració adient de l'entorn.

## **Fase de disseny i implementació**

Un cop s'han decidit totes les eines a utilitzar de cara al desenvolupament del projecte, s'ha començat la implementació de cada una de les parts que formen el servei complet. Per dur a terme la implementació del sistema, s'ha adaptat la metodologia àgil SCRUM (coneguda principalment per treballar en equip) a un sol individu. La metodologia SCRUM és una tècnica per a la coordinació del treball en equip sense tenir un pla definit i tancat, i que s'enfoca en complir petits objectius a curt termini, anomenats *sprints*.

A continuació es defineixen les pautes que s'han seguit per adaptar aquesta metodologia. S'han tingut en compte els següents punts:

- Reunions constants amb el tutor del projecte, per aconseguir tenir el feedback necessari del que s'ha fet i el es té previst fer.
- Amb els requeriments capturats en les fases anteriors, s'ha obtingut un *backlog* de totes les tasques fetes i pendents de fer. Aquest *backlog* s'ha anat actualitzant durant els procés de desenvolupament.
- S'han definit *sprints* amb el subconjunt de tasques definides en el *backlog*. S'han definit objectius assequibles a curt termini.

- El resultat d'un *sprint* és la suma de totes les tasques completades durant l'*sprint*, integrades amb tots els *sprints* anteriors. Cal que al final de cada *sprint* el producte sigui funcional i usable, independentment de si es decideix posar-lo en producció o no.

#### **a- Base de dades**

Degut al tipus d'aplicació que s'ha desenvolupat, on diferents components del sistema d'informació tenen dades que es poden relacionar directament entre si, s'ha optat per un tipus de base de dades relacional, ja que faciliten la cerca i la organització d'aquestes dades.

#### **b- Aplicació i Back End API**

S'ha optat per desenvolupar tant la API com la interfície de l'aplicació alhora, implementant primerament els casos d'ús independents. Un cop desenvolupats s'han ampliat amb els casos d'ús que tenien dependències amb els prèviament implementats.

Per cada cas d'ús ha fet falta implementar tant la interfície gràfica com tots els *endpoints* necessaris per a la seva utilització.

#### **Fase de desplegament i proves**

Un cop finalitzada la implementació, cal desplegar-la públicament. S'ha contractat un servidor on s'allotjarà l'aplicació i un domini. Una vegada adquirits aquests serveis, s'ha realitzat la configuració del servidor, instal·lant els paquets i dependències necessàries i configurant-les pel seu correcte funcionament.

S'ha clonat el repositori git del codi font de totes les versions de l'aplicació. Això ens permetrà facilitar el desplegament de noves versions així com tenir sempre una còpia de seguretat actualitzada.

#### **Fase de documentació**

Durant les fases prèviament esmentades s'ha anat documentant progressivament l'avanç del projecte. Tots els prototips visuals, diagrames, esquemes o documentació tècnica s'han inclòs a la memòria final del projecte.

### 2.2.2. Control de versions i repositori

En tot projecte de software, és imprescindible tenir una còpia de seguretat del codi font. En el cas que es guardés el codi en una màquina local i aquesta s'espatllés, significaria perdre tot el treball realitzat i haver de tornar a començar.

Git és un sistema de control de versions que permet mantenir un historial cronològic, organitzat a través d'operacions *commit*, de tots els canvis que s'han fet sobre el projecte.

S'ha utilitzat Github com a plataforma web per allotjar la nostra aplicació. Com tota plataforma web, Github funciona en un servidor remot, pel ha permès tenir aquest repositori com a còpia de seguretat del projecte.

Tenir el codi font allotjat en un repositori Git facilita l'accés al desenvolupador, ja que pot accedir al repositori en qualsevol moment i des de qualsevol màquina.

Durant tota la fase d'implementació de l'aplicació s'ha utilitzat Git com a eina de suport. S'ha creat un sol repositori que conté tant el codi de la interfície de l'aplicació com de la API que li proporciona les dades necessàries.

## 3. Planificació

Tot projecte requereix d'una planificació temporal inicial amb l'objectiu de definir les fases i el temps que s'estima dedicar a cada una d'elles.

Aquest projecte té una durada aproximada de cinc mesos, un període comprès entre el 9 de gener de 2019 i el 13 de juny de 2019.

En aquesta planificació inicial s'ha acotat cada fase a un període de temps determinat, tot i que queden obertes a modificacions durant el transcurs del projecte ja que poden sorgir imprevistos. Cada fase s'ha dividit en una sèrie de subtasques.

### 3.1. Descripció de fases

#### **Fase Inicial**

La primera fase consisteix en una recollida d'informació prèvia al desenvolupament de l'aplicació. Té una durada d'un mes, comprèn el període entre el 9 de gener de 2019 i el 9 de febrer de 2019. Cal definir ordenadament:

- Objectius del projecte.
- Estudi previ del context, de les tecnologies actuals i necessitats d'informació.
- Captura de requeriments funcionals i tecnològics.
- Estructura del projecte i recursos humans, temporals i materials
- Pressupostar costos
- Anàlisi de viabilitat tècnica, econòmica i mediambiental

#### **Fase d'anàlisi i especificació**

La segona fase depèn directament del treball realitzat a la fase inicial. Té una durada d'un mes, que comprèn entre el 9 de febrer de 2019 i el 9 de març de 2019. En primer lloc s'analitzaran els requeriments i s'especifica el comportament i les interaccions del sistema a través de la definició dels casos d'ús. Es preveu aproximadament una setmana de treball.

Un cop definits els casos d'ús i tenint clar quin comportament ha de tenir l'aplicació, s'ha començat a prototipar visualment l'aplicació. També es s'ha definit el model de dades sobre el que treballa l'aplicació. Es preveu aproximadament una setmana de treball.

Per últim i prèviament a la fase d'implementació, l'estudiant ha pres aproximadament dues setmanes per formar-se correctament amb les tecnologies a usar. Això ha permès que s'hagi familiaritzat amb l'entorn de treball i hagi establert un marc de treball i de bones pràctiques.

### **Fase de disseny i implementació**

La tercera comprèn un període d'una mica més de dos mesos. Es preveu que compregui del 9 de març fins el 15 de maig de 2019. Degut a la metodologia especificada anteriorment, s'ha fragmentat aquesta fase en *sprints*, una vegada s'hagin definit i tancat tots els casos d'ús de l'aplicació. Cada *sprint* ha tingut una durada concreta i serà independent de les anteriors.

### **Fase de desplegament i proves**

Es preveu que aquesta fase no comporti més de dues setmanes, compresa del 15 de maig de 2019 al 31 de maig de 2019. Té una seqüència de tasques molt concreta:

- Contractar hosting.
- Contractar domini.
- Configurar el servidor i instal·lar les dependències necessàries.
- Desplegar l'aplicació

**Nota:** És possible que un cop es tingui un producte mínimament funcional durant la fase anterior, s'avanci aquesta fase i s'incorpori a l'anterior. Això permet una integració continua de les funcionalitats i es poden realitzar les proves directament en el servidor de producció.

### **Fase de documentació**

Tot i que durant l'execució de les fases anteriors s'han anat documentant progressivament tots els avenços, l'últim període, comprès del 1 de juny de 2019 al 13 de juny de 2019, s'ha dedicat a finalitzar la documentació i a preparar la defensa del projecte.

## **3.2. Recursos i gestió econòmica**

Alhora de desenvolupar un projecte es necessiten una sèrie de recursos que permeten dur-lo a terme. Una vegada plantejat el projecte cal realitzar un estudi econòmic dels recursos per determinar la seva viabilitat.



Es defineixen com: recursos humans, recursos de hardware i recursos de software. Per aquest projecte en concret seran els següents:

### 3.2.1. Recursos i costos humans

- L'estudiant del Grau en Enginyeria Informàtica és l'encarregat de l'anàlisi, el disseny i la implementació del projecte.
- El tutor del projecte ha de donar suport a l'estudiant i realitzar un seguiment durant el temps d'execució del projecte.

#### **Cost en recursos humans**

Per a calcular el cost del recurs humà es necessiten dues dades principals: el número d'hores invertides i el preu brut total per una hora de treball. Partint del sou brut anual s'ha extrapolat aquesta dada per a trobar quin és el preu/hora aproximat.

Segons Indeed, un dels portals per a buscar feina més importants i que mou aproximadament uns 250 milions d'usuaris mensuals únics, el sou brut d'un analista programador junior a Espanya és de **18.931€**, estimat a partir de 363 fonts obtingudes de les empreses, usuaris i feines del portal els últims 36 mesos. La última data d'actualització i per tant amb la que està basada el sou esmentat és l'1 de febrer de 2019.

Tal i com indica la Figura 3.1, el cost total d'una hora de treball d'un programador junior és de **13,99€**.

Partint de que el número d'hores estimades per dur a terme el projecte és de 500 hores, això comporta un cost humà total mínim de:

$$\text{Cost humà} = 500 \text{ hores} \cdot 13,99\text{€/hora}$$

$$\text{Cost humà} = 6.995\text{€}$$


Concepte	Percentatge		Hores treball anuals
<b>Salari anual brut</b>		18.931,00 €	
			 (Ctrl) ▾
Seguretat social empresa	33,00%	6.247,23 €	
<b>Cost salari empresa anual</b>		25.178,23 €	
<b>Cost preu/hora empresa</b>		<b>13,99 €</b>	1800

Figura 3.1 Taula de costos de recursos humans

Producte	Vida útil (anys)	Cost	Amortització
Lenovo Thinkpad E470	6	1.100€	91,67€
Pantalla Samsung 24"	6	150€	12,50€
Impressió + encuadernat		20€	20€
<b>Total</b>			<b>124,17€</b>

Figura 3.2 Taula de costos materials

### 3.2.2. Recursos i costos de hardware

- Portàtil Lenovo Thinkpad E470, 16GB de memòria RAM, disc dur SSD de 256GB i targeta gràfica NVIDIA Geforce 940MX de 2gb dedicats.
- Pantalla Samsung 24”.

#### Cost en recursos materials

La quota d'amortització dels recursos materials que s'han utilitzat durant el projecte es resumeix en la Figura 3.2. S'ha calculat amb la fórmula: amortització = cost vida útil · 0,5, on 0,5 prové de la durada del projecte, aproximadament sis mesos.

El cost d'impressió i encuadernat s'ha calculat per una extensió aproximada d'unes cent pàgines.

### 3.2.3. Recursos i costos de software

- Sistema operatiu – Ubuntu 18.04 LTS
- Base de dades – MySQL
- Visual Studio Code com a editor de codi.

- Git (plataforma Github) com a sistema de control de versions.
- Insomnia com a eina per fer peticions HTTP de manera senzilla. Ajuda en el desenvolupament de les API.

**Cost en software**

Tot el programari i eines utilitzades pel desenvolupament del projecte són de codi obert i gratuïtes.



## 4. Requeriments i anàlisi del sistema d'informació

### 4.1. Captura de requeriments

#### Requeriments funcionals

L'aplicació ha de complir les següents especificacions funcionals:

- Permetre als usuaris tenir una sessió única pròpia.
- Ha de definir tres nivells de perfils: administradors, gestors i membres. Els nivells superiors són una extensió dels nivells inferiors.
- Permetre als administradors crear i gestionar perfils d'usuari.
- Permetre els gestors:
  - o Crear i gestionar grups.
  - o Afegir usuaris o grups a grups.
  - o Enviar comunicats a grups.
  - o Crear votacions
- Permetre als membres:
  - o Consultar la informació.
  - o Enviar documentació
  - o Respondre votacions
- Mostrar de manera eficaç i clara la informació.
- Treballar amb un tipus de comunicació unidireccional i asimètrica.
- Informar els usuaris corresponents a través de notificacions push.
- Oferir un sistema de votació segur.
- Validar que els components del grup reben correctament la informació.

Els requeriments mínims que ha de complir el sistema de votació són els següents:

- Crear les preguntes i respostes de les enquestes.
- Afegir i eliminar preguntes i respostes a les preguntes
- Consultar els resultats de les enquestes

## Requeriments tècnics

Els requeriments tècnics que ha de complir l'aplicació són els definits a continuació:

- Ser compatible amb iOS i Android (multiplataforma).
- Oferir servei offline per a la consulta de dades.

## 4.2. Regles de negoci

- RN-1 L'usuari té una sessió única.
- RN-2 S'ha de permetre afegir usuaris o altres grups a grups.
- RN-3 Un usuari només pot accedir als comunicats dels grups que pertany actualment.
- RN-4 Un usuari pot formar part de grups diferents.
- RN-5 L'usuari podrà votar sempre i quan estigui convidat a la votació.
- RN-6 L'usuari podrà votar sempre i quan no ho hagi fet anteriorment.
- RN-7 Els noms dels grups han de ser únics per client.
- RN-8 No s'eliminaran dades, es desactivaran en mesura del possible.
- RN-9 A l'eliminar un usuari, s'eliminaran les dades sensibles d'acord amb la LOPD i es substituiran per dades fictícies.
- RN-10 Els noms d'usuari han de ser únics per client.

## 4.3. Casos d'ús

A partir dels requeriments capturats, s'han definit els casos d'ús que s'implementaran en aquesta aplicació. Han sorgit un total de 14 casos d'ús llistats a continuació en forma de diagrama de casos d'ús. La descripció de les fitxes dels casos d'ús es pot consultar a l'annex 1.

- La Figura 4.1 mostra els casos d'ús de l'actor amb perfil de membre.
- La Figura 4.2 mostra els casos d'ús de l'actor amb perfil de gestor.
- La Figura 4.3 mostra els casos d'ús de l'actor amb perfil d'administrador.

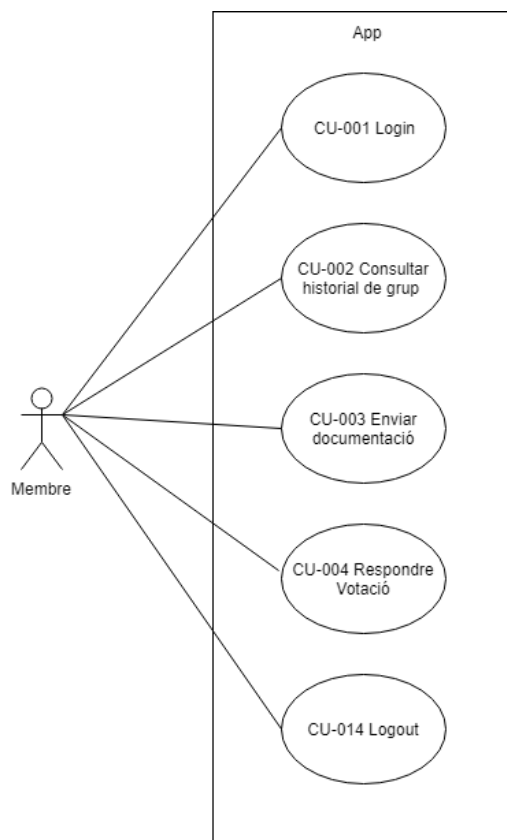


FIGURA 4.1 DIAGRAMA DE CASOS D'ÚS DEL MEMBRE

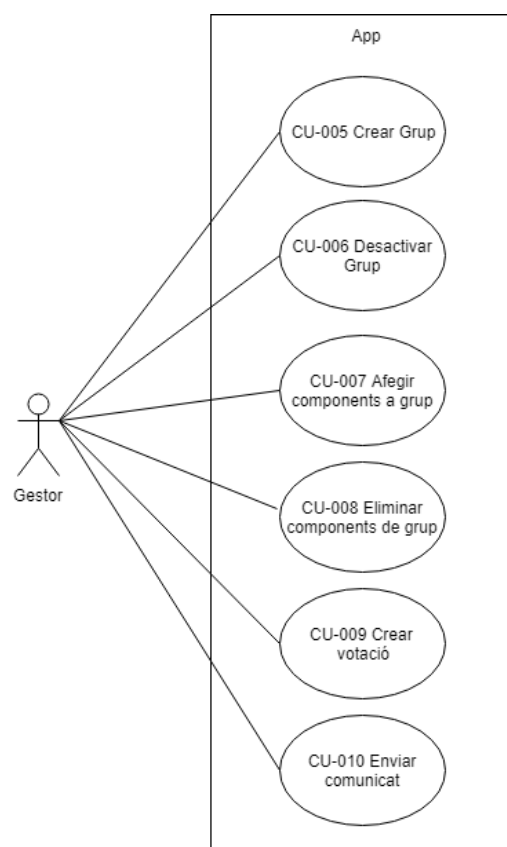


FIGURA 4.2 DIAGRAMA DE CASOS D'ÚS DEL GESTOR

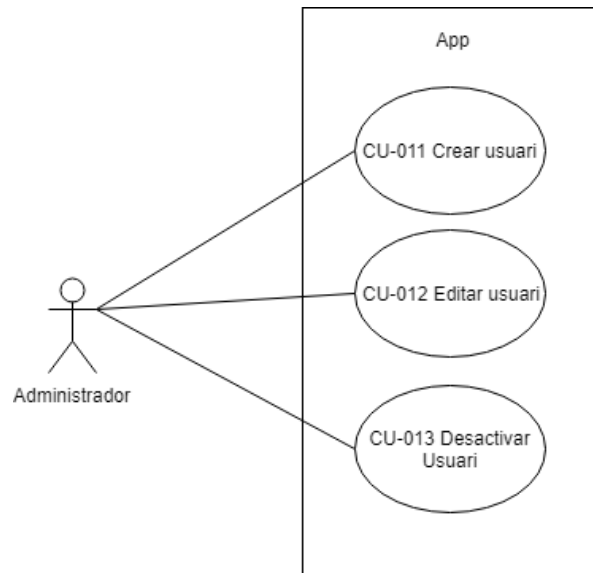


FIGURA 4.3 DIAGRAMA DE CASOS D'ÚS DE L'ADMINISTRADOR

#### 4.4. Wireframing

Una vegada definit els casos d'ús que s'han d'implementar i amb l'objectiu de tenir una idea prèvia al desenvolupament de l'aplicació s'han dissenyat *wireframes* de totes les pantalles.

Els *wireframes* de les pantalles juntament amb les definicions dels casos d'ús permeten tenir una idea molt aproximada de què ha de fer cada pantalla i com serà la navegació entre elles.

Aquests *wireframes* han estat dissenyats a mà i són a l'annex 2.

#### 4.5. Estudi dels sistemes de votació

##### Propostes sistema de votació

Per a implementar el sistema de votació, s'han investigat serveis externs de votació segurs ja existents per veure si es poden integrar amb el nostre sistema, facilitant així la tasca. La principal característica que es busca en aquest servei és que disposi d'una API REST o similar per a la gestió de les enquestes: creació de les preguntes i consulta de les respostes i resultats. Es disposa de tres opcions a estudiar per a la implementació del sistema de votació.



**a- Integrar completament les enquestes a l'aplicació via API REST**

La primera opció és trobar un servei API REST amb el que podem integrar completament i de manera transparent la nostra aplicació. La gestió de la informació es faria a través de peticions HTTP als mètodes de l'API.

**b- Generar un fitxer de paràmetres des de l'aplicació i enviar-lo**

La segona opció és generar un fitxer per a la creació d'una enquesta amb les preguntes i respostes corresponents a través d'un formulari dins de l'aplicació. Enviar el fitxer generat a un servei extern que processaria la informació per a generar el formulari. Posteriorment, ens permetria consultar l'enquesta ja generada per a que els usuaris puguin respondre-la a través d'una API o incrustant-la a través d'un iframe.

**c- Implementació d'un model propi**

Si cap de les opcions anteriors satisfà les necessitats de l'aplicació es desenvoluparà un model propi que compleixi amb tots els requeriments. S'ha de garantir la correcta votació anònima de les parts implicades, sabent en tot moment qui però no què ha votat.

**Propostes d'externalització del sistema de votació**

S'han estudiat diferents serveis ja existents per a intentar cobrir aquesta funcionalitat amb garanties. Els serveis d'enquesta consultats han estat: LimeSurvey, SurveyMonkey, OpenAgora y SurveyPal. Un dels principals problemes que s'ha trobat és que cap d'aquests serveis ens ofereix el servei tal i com el necessitem.

LimeSurvey, OpenAgora i SurveyMonkey ens facilita una API per a la gestió de les enquestes, però ho fan tenint en compte que cada usuari ha de tenir un compte creat en el seu servei. No es vol obligar a l'usuari a haver de crear-se un compte extern a la nostre aplicació per a poder accedir a les votacions. Això minvaria de manera crítica l'experiència de l'usuari i disminueix la sensació d'estar usant un sistema segur i privat.

Per altre banda, SurveyPal facilita la interfície completa de creació i gestió d'enquestes i votacions a través de portals web. De la mateixa manera que el grup anterior, això limita una bona experiència per l'usuari.

## 5. Implementació: Base de dades

En aquesta secció es detalla la funcionalitat referent a la persistència de les dades. Aquest projecte, com a sistema d'informació que és, requereix guardar totes les dades pel posterior tractament d'elles.

És imprescindible mantenir una bona integritat de la base de dades a través de regles *check* i de nul·labilitat de camps. També s'ha assegurat la integritat referencial a través de claus primàries i claus foranes.

S'ha dissenyat una base de dades MySQL allotjada en el núvol. Totes les dades han de poder ser consumides en qualsevol moment per l'aplicació a través de la API.

### 5.1. Sistema gestor de base dades

MySQL és un Sistema Gestor de Base de Dades Relacional (SGBDR) open source sota una llicència GNU General Public License, tot i que també està disponible sota llicències privatives. MySQL és la base de dades open source més popular ja que s'integra de manera fàcil amb qualsevol tipus d'aplicació.

Els punts forts pels quals MySQL satisfà les necessitats requerides pel tipus d'aplicació com a sistema gestor de base de dades són els següents:

- Senzillesa d'ús i manteniment: Des de un procés d'instal·lació lineal a una administració i un manteniment senzill fan que qualsevol desenvolupador pugui aprendre tot el necessari per gestionar la base de dades.
- Suportada per una gran comunitat facilita al desenvolupador trobar solucions a possibles problemes.
- Tenir menys característiques que altres SGBDR permet córrer MySQL sobre hardware de menys cost. No té un cost en hardware on no es necessita realment.
- Multiplataforma: Es pot instal·lar tant en servidors Linux com Windows.
- Satisfà les propietats ACID (transaccions atòmiques, consistents, aïllades i durables) requerides per una aplicació bàsicament transaccional.

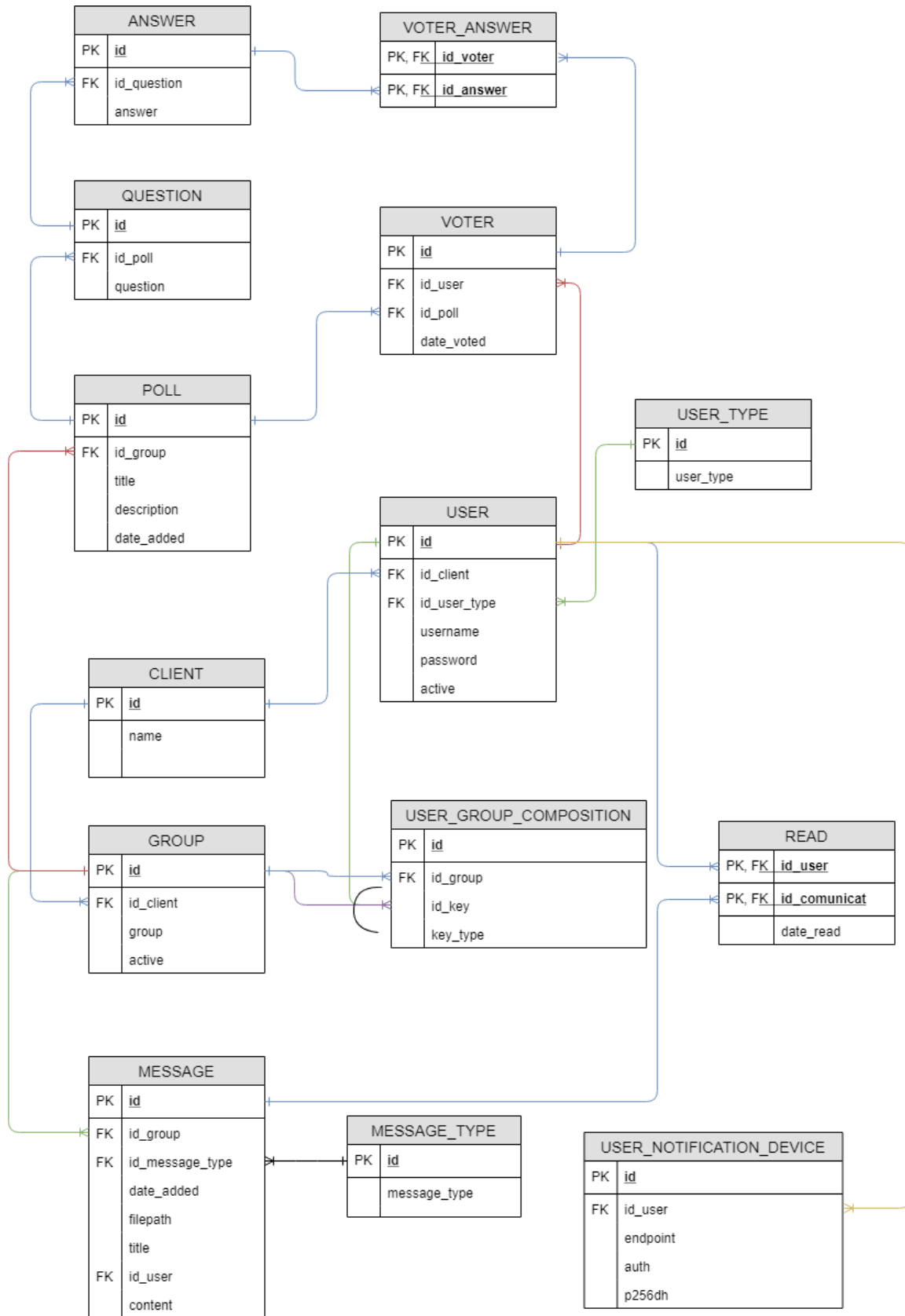


FIGURA 5.1 MODEL DE BASE DE DADES

## 5.2. Disseny del model de dades

Un cop s'han definit tant els requisits com els casos d'ús de l'aplicació, s'ha implementat el model de dades que utilitza el sistema d'informació.

S'ha buscat la màxima integritat de les dades dins del model a través de claus foranes. Degut a la política definida a les regles de negoci del projecte, no s'eliminen registres d'entitats que no siguin febles. En aquest cas, es desactiven. No s'han hagut d'aplicar polítiques restrictives alhora d'eliminar o actualitzar registres en aquestes taules, ja que no es perd mai la relació de la clau forana. El disseny de base de dades es pot veure a la Figura 5.1.

L'eina utilitzada per a dibuixar el diagrama entitat-relació és draw.io, un software de diagrames online gratuït.

## 5.3. Definició de les taules

Una vegada s'ha validat el disseny del model de dades resultant perquè suporti tots els requeriments capturats, s'han definit un total de tretze taules. L'objectiu concret de cada taula es defineix de la següent manera:

### **USER\_TYPE**

La taula `USER_TYPE` defineix els tipus d'usuari que existeixen dins de l'aplicació. La formen els camps `id`, un identificador únic com a clau primària i el camp `user_type` que és el nom del tipus d'usuari.

### **USER**

La taula `USER` és un dels components principals del model de dades. Emmagatzema les dades dels usuaris amb l'objectiu de que puguin accedir a l'aplicació. Tenir un usuari actiu és un requisit indispensable per a poder utilitzar l'aplicació.

El camp `active` és un camp booleà que permet restringir l'accés a l'aplicació en el cas de que el seu valor sigui 0.

Els camps *username* i *password* són els encarregats de donar accés a l'aplicació, sempre i quan sigui un usuari amb el camp *active* amb valor 1.

El camp *id\_user\_type* és una clau forana a la taula `USER_TYPE` i permet assignar un valor equivalent a un rol dins l'aplicació.

El camp *id\_client* permet classificar els usuaris per clients. És una clau forana a la taula `CLIENT`.

## **GROUP**

La taula `GROUP` guarda la informació bàsica de cada grup.

Cada grup té un *id* com a identificador únic (clau primària), el camp *group\_name* per emmagatzemar el nom i un camp *active* booleà per distingir grups actius d'inactius.

El camp *id\_client* permet classificar els grups per clients. És una clau forana a la taula `CLIENT`.

## **CLIENT**

La taula `CLIENT` permet poder utilitzar la mateixa instància de la base de dades per a tots els clients/usuaris. Permet classificar les taules `USER` i `GROUP`, les dues taules principals del sistema d'informació i d'on hi deriven la resta de les dades.

Consta d'un *id* com a identificador únic i clau primària i el camp *name* per definir-li un nom únic.

## **USER\_GROUP\_COMPOSITION**

Aquesta taula permet realitzar l'assignació d'usuaris a grups i/o assignar grups a grups pare, podent crear jerarquies de grups estil arbre.

El camp *id* defineix la seva clau primària.

El camp *key\_type* funciona com un “arc exclusiu”, pot prendre el valor 1 o 2. És el camp que permet diferenciar entre si és una relació “grup pare” – usuari o una relació “grup pare” – “grup fill”.

El camp *id\_key* defineix l'*id* corresponent a l'usuari o grup, dependent del valor *key\_type*.

El camp *id\_group* defineix quin és el grup pare en el qual se li assigna un usuari o un grup, dependent del valor del camp *key\_type*. És una clau forana a la taula GROUP.

## **MESSAGE\_TYPE**

La taula MESSAGE\_TYPE defineix els tipus de missatge que es poden enviar dins de l'aplicació. La formen els camps *id*, un identificador únic com a clau primària i el camp *message\_type* que és el tipus de missatge. Permet classificar els comunicats com a missatge o arxiu.

## **MESSAGE**

La taula MESSAGE emmagatzema les dades de tots els comunicats enviats a cada grup. El camp *id* defineix l'identificador únic de la taula. El camp *id\_message\_type* és una clau forana a la taula MESSAGE\_TYPE i defineix el tipus de comunicat emmagatzemat.

El camp *id\_group* és una clau forana a la taula GROUP i defineix quin és el grup destinatari del missatge.

El camp *id\_user* és una clau forana a la taula USER i defineix quin és l'usuari que ha enviat el missatge.

El camp *date\_added* guarda la data en que es crea el missatge. Té un valor per defecte a la data i hora del moment en que s'insereix el registre.

El camp *title* permet definir un títol principal per a cada comunicat enviat.

El camp *content* permet definir el cos del comunicat a enviar, sempre i quan sigui de tipus missatge.

El camp *filepath* defineix la ruta de l'arxiu enviat, sempre i quan el tipus de missatge sigui arxiu.

## **READ**

La taula READ permet tenir un registre de totes les lectures dels usuaris sobre cada missatge.

Consta dels camps *id\_message* que és una clau forana a la taula MESSAGE i *id\_user* que és una clau forana a la taula USER. El camp *date\_read* és un camp nul·lable per defecte. Quan un usuari llegeix un missatge, s'actualitza amb la data de lectura, marcant que el missatge ha estat llegit per l'usuari.

## **POLL**

La taula POLL defineix la informació base de cada enquesta. Cada enquesta té un *id* com identificador únic, un *id\_group* que és una clau forana a la taula GROUP i defineix sobre quin grup es vol fer l'enquesta. El camp *title* defineix un títol curt per identificar l'enquesta i el camp *description*, que és opcional, permet afegir una petita descripció.

El camp *date\_added* guarda la data del moment en què s'insereix el registre.

## **QUESTION**

Aquesta taula guarda la informació de les preguntes de cada enquesta. Cada pregunta té un *id* com a identificador únic, un *id\_poll* que és una clau forana a la taula POLL i indica a quina enquesta pertany. El camp *question* guarda la cadena de text de la pregunta corresponent.

## **ANSWER**

La taula ANSWER emmagatzema la informació referent a cada resposta. Cada resposta té un *id* com a identificador únic, un *id\_question* que és una clau forana a la taula QUESTION i indica que és una possible resposta a aquella pregunta. El camp *answer* guarda la cadena de text de la resposta corresponent.

## **VOTER**

La taula VOTER guarda la informació de tots els votants d'una enquesta. Funciona com una mena de val per a un usuari en una votació. Cada registre es compon d'un *id* com a identificador únic, un *id\_user* que és una clau forana a la taula USER per identificar qui és l'usuari que vota i un *id\_poll* que és una clau forana a la taula POLL per identificar l'enquesta. Per últim, un camp *date\_voted* que pot ser nul·lable i que guarda la data en què l'usuari ha realitzat la votació, privant-lo així de que pugui votar de nou.



## **VOTER\_ANSWER**

La taula VOTER\_ANSWER guarda les respostes seleccionades per cada VOTER. Consta d'una clau primària composta de *id\_answer*, que alhora és una clau forana de la taula ANSWER i de *id\_voter*, que alhora és una clau forana de la taula VOTER.

D'aquesta manera es guarden les respostes que voten els usuaris per cada enquesta, podent fer informes per veure els resultats.

## **USER\_NOTIFICATION\_TOKEN**

La taula USER\_NOTIFICATION\_TOKEN guarda els identificadors dels dispositius de cada usuari que han donat permís per rebre notificacions push.

Els camps que componen la taula són *id* com a identificador únic, *id\_usuari* per a poder saber a qui se li ha d'enviar cada notificació, el camp *endpoint* que emmagatzema la URL del dispositiu el qual s'enviaran les peticions i els camps *auth* i *p256dh* que guarden claus de credibilitat necessàries per l'enviament de les notificacions del servidor al client.



## 6. Implementació: API RESTful

Un dels elements claus de l'arquitectura del projecte és el desenvolupament d'una API RESTful, que s'encarrega de comunicar l'aplicació amb la base de dades a través de peticions HTTP.

La API proporciona tot el conjunt de funcions i procediments per accedir i tractar les dades des de serveis externs.

REST (Transferència d'estat representacional) és un tipus d'arquitectura de software que defineix un conjunt de normes a seguir alhora de construir un servei web.

Les API RESTful es caracteritzen per no conservar cap estat entre peticions diferents, ja que cadascuna és independent de les altres. Cada petició al servidor ha de contenir tota la informació necessària per a poder ser entesa i no pot dependre de cap estat guardat prèviament en el servidor.

### Recursos

S'entén per recurs qualsevol element d'informació el qual s'hi pot accedir a través d'un identificador. La representació d'aquest recurs en una marca de temps concreta és el seu estat.

En aquest projecte s'entén que cada registre de les taules de la base de dades és un recurs ja que aplica la definició prèviament esmentada.

### Comunicació amb la API

Cada petició HTTP a l'API consta de 3 elements principals:

- Una URI identificativa del recurs.
- Un mètode HTTP: GET, POST, PUT, PATCH o DELETE.
  - o GET: És un mètode segur. Significa que accedir a un recurs a través d'aquest mètode no ha de resultar en un canvi en el recurs.

- GET, PUT, DELETE: Són mètodes idempotents. Significa que fer peticions amb els mateixos paràmetres d'entrada no tindrà cap efecte addicional sobre el recurs resultant.
- Un tipus de contingut (MIME type).

La resposta de la API RESTful sempre retorna un codi HTTP que indica l'estat resultant i un cos en format JSON, si s'escau.

És molt important fer un ús correcte dels codis de resposta HTTP per a que les aplicacions consumidores de la API puguin tractar els resultats obtinguts de manera eficient.

Els codis de resposta HTTP es poden classificar en:

- 1xx (Respostes informatives): El servidor ha rebut la capçalera de la petició correctament i el client ha de procedir a enviar el cos de la petició.
- 2xx (Peticions correctes): Indica que la petició ha estat rebuda, entesa i acceptada correctament.
- 3xx (Redireccions): Indica que el client ha de prendre una acció addicional per completar l'acció, normalment una redirecció.
- 4xx (Error de client): La petició no es pot processar, en el cas de que l'error provingui del client.
- 5xx (Error de servidor): El servidor falla al processar una petició vàlida per part del client.

## 6.1. Entorn de desenvolupament

La API RESTful que forma tota la part del *backend* de l'aplicació s'ha desenvolupat amb Node.js.

El punt clau en l'elecció del llenguatge en l'entorn del servidor ha estat el tipus de tractament de dades que s'ha de fer en aquest projecte. Sent una aplicació que es basa en operacions CRUD concurrents, es necessita d'un entorn que satisfaci aquestes necessitats.

Node.js està pensat per a construir aplicacions ràpides, no bloquejants i escalables i és capaç de tractar moltes connexions simultànies amb un bon rendiment.

El flux principal que segueix una petició a la API és el següent:

- El client fa una petició a la API.
- La API llegeix i interpreta els paràmetres necessaris de la petició.
- Opcionalment llegeix o escriu a la base de dades.
- Construeix una resposta i la retorna al client.

Fent referència al tipus de dades sobre el qual treballa la API, tant les dades de la petició com de la resposta són, en general, cadenes de text en format JSON, el qual fa que aquestes tinguin un volum petit.

Cal esmentar el gestor de paquets propi de Node.js: NPM. Es pot accedir a aquests paquets de manera pública a través d'un repositori online i compten amb versió i gestió de dependències.

## Framework Express

Express és un *framework* web per a Node.js que proveeix al desenvolupador un conjunt de funcionalitats sobre el qual construir un servei web. Una de les característiques més importants que aporta pel desenvolupament de la API és la facilitat amb la que es poden definir rutes i vincular-les a una funcionalitat concreta d'un controlador. Permet la opció d'aplicar funcions de *middleware* a certes rutes i que actuen com a interceptores de les peticions.

Dos dels components principals que utilitza aquest *framework* són els l'objecte *request* (petició) que conté tota la informació referent a la petició HTTP (paràmetres, capçaleres, cos) i l'objecte *response* (resposta) que conté tota la informació a retornar després de processar la informació.

## 6.2. Seguretat

Tota aplicació que no és d'ús públic, requereix la implementació de certs mecanismes per a garantir la privacitat i la protecció de les dades. De la mateixa manera, les dades sensibles, com poden ser les contrasenyes, han d'estar encriptades

### Encriptació de contrasenyes

S'ha utilitzat la funció de hashing **bcrypt** per a encriptar les contrasenyes dels usuaris. Aquest mètode d'encriptació incorpora un valor aleatori anomenat *salt* que s'utilitza alhora de formar el *hash*. El *salt* evita que dues contrasenyes iguals generin el mateix valor de *hash*.

El paquet **bcrypt** per a Node.js proporciona una API tant per l'encriptació com per la verificació de la contrasenya. S'ha instal·lat a través de NPM.

### JWT (JSON Web Tokens)

S'han protegit tots els endpoints de la API amb JSON Web Token. JWT és un estàndard obert per a la creació de *tokens* d'accés.

Per a la creació del *token* s'utilitzen les dades de l'usuari (*username*, *password*, *id\_client* i *id\_user\_type*) juntament amb una clau secreta del servidor.

Aquests *tokens* són generats pel servidor quan l'usuari logeja a l'aplicació i són retornats al client, que el guarda localment. Un dels avantatges d'aquest mètode de protecció és que alhora de verificar el *token* cada vegada que l'usuari intenta fer una petició a la API a un recurs o a una ruta protegida, el servidor sap tant si és un usuari vàlid com de quin tipus d'usuari es tracta, afegint un altre nivell de protecció: el rol de l'usuari.

Qualsevol petició a la API per part de l'usuari la fa afegint a la petició HTTP la capçalera *Authorization* i utilitzant l'esquema *Bearer*. La capçalera d'autenticació té el següent format:

```
Authorization: Bearer <token>
```

El paquet **jsonwebtoken** per a Node.js proporciona una API per a la signatura i verificació de tokens JWT. S'ha instal·lat a través de NPM.

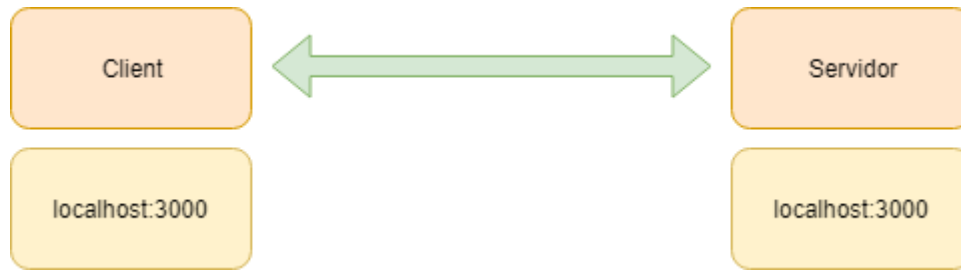


FIGURA 6.1 ARQUITECTURA TRADICIONAL CLIENT-SERVIDOR



FIGURA 6.2 ARQUITECTURA CLIENT-RESTFUL API

### Cross-Origin Resource Sharing (CORS)

L'intercanvi de recursos d'origen creuat (CORS) és un mecanisme que utilitza unes capçaleres HTTP addicionals per permetre l'accés als recursos a un origen diferent que el propi servidor.

En una arquitectura tradicional on el client fa una petició al servidor i aquest li retorna una pàgina HTML no existeix aquest problema ja que estàs accedint als recursos del servidor des de el propi servidor.

Ara bé, per una RESTful API no és el cas. El més normal és que client i servidor tinguin diferents orígens. Inclòs un port diferent dins del mateix servidor es considera un origen diferent. En aquest cas, fer una petició a un recurs del servidor falla. Com a RESTful api però, es vol permetre aquest accés des de orígens externs, per tal de que els clients puguin consumir les dades.

Per a fer-ho, s'han d'enviar una sèrie de capçaleres HTTP del servidor al client que donen accés als recursos sol·licitats. Les capçaleres a enviar són:

```
app.use((req, res, next) => {
  res.header("Access-Control-Allow-Origin", "*");
  res.header(
    "Access-Control-Allow-Headers",
    "Origin, X-Requested-With, Content-type, Accept, Authorization"
  );
  if(req.method === 'OPTIONS') {
    res.header("Access-Control-Allow-Methods", "GET, POST, PUT, PATCH, DELETE");
    return res.status(HTTP_CODES.OK).json({});
  }
  next();
});
```

FIGURA 6.3 HABILITAR EL PERMÍS D'ACCÉS A PETICIONS D'ORIGEN EXTERN

- Access-Control-Allow-Origin per filtrar quins orígens vols permetre que consumeixin recursos.
- Access-Control-Allow-Headers per filtrar quines capçaleres HTTP es permeten rebre.
- Access-Control-Allow-Methods per filtrar quins mètodes de petició HTTP es permeten.

La Figura 6.3 mostra el fragment de codi que aplica aquests permisos.

Alhora de fer una petició, el navegador envia primer una petició amb mètode OPTIONS amb l'objectiu de que el servidor verifiqui el client com a consumidor. Si la resposta és vàlida, procedeix amb la petició real a efectuar.

## Base de dades

Per a realitzar tant la connexió com les operacions de lectura i escriptura a la base de dades, s'ha utilitzat el paquet **mysql** per a Node.js que ofereix un conjunt d'eines per a la interacció amb la base de dades.

Una tècnica de seguretat automàtica que proporciona aquest paquet és l'escapament de caràcters. És de vital importància escapar els caràcters enviats pel client abans de realitzar una consulta a la base de dades, amb l'objectiu d'evitar atacs d'injecció d'SQL. En cas de no fer-ho, un atacant pot enviar consultes SQL com a paràmetre i extreure informació no desitjada o sensible de la base de dades.



### 6.3. Middlewares

S'han utilitzat funcions de *middleware* com a mecanisme de filtració de les peticions que entren a la API. Aquestes funcions de *middleware* protegeixen les rutes de l'aplicació a certs usuaris consumidors de la API.

S'han creat tres funcions de *middleware* diferents. A continuació, es detalla l'objectiu de cada una d'elles.

#### CheckAuth

Aquesta funció de *middleware* protegeix tots els *endpoints* de la API, a excepció del cas de *login*. S'encarrega de validar que la petició HTTP conté la capçalera *Authorization* i verifica el valor del *token* JWT, tal i com s'ha esmentat en l'apartat anterior.

En el cas de que la petició entrant no contingui aquesta capçalera automàticament retorna una resposta HTTP al client amb el codi d'estat 401, equivalent a *Unauthorized* (No autoritzat).

Per altre banda, si el valor enviat a la capçalera no passa el procés de validació, automàticament retorna una resposta HTTP al client amb el codi d'estat 403, equivalent a *Forbidden* (Prohibit).

```
module.exports = (req, res, next) => {
  try{
    const bearerHeader = req.headers.authorization;
    if (typeof bearerHeader === 'undefined') {
      res.sendStatus(HTTP_CODES.FORBIDDEN);
    }
    const bearerToken = bearerHeader.split(" ")[1];
    const decoded = jwt.verify(bearerToken, process.env.SECRET_KEY);
    req.token = decoded;
    next();
  } catch (error) {
    return res.status(HTTP_CODES.UNAUTHORIZED).json({
      message : 'Authentication failed'
    });
  }
};
```

FIGURA 6.4 VERIFICACIÓ DEL TOKEN D'AUTORITZACIÓ

En el cas de que el *token* sigui vàlid, afegeix a l'objecte *request* les dades de l'usuari, extretes del *token* descodificat, i procedeix en l'execució.

Es pot veure la implementació en la Figura 6.4.

### CheckGestor i CheckAdmin

Aquestes dues funcions de middleware funcionen de la mateixa manera i serveixen per protegir aquells endpoints on només hi poden accedir els usuaris que són del tipus **Gestor** i **Administrador**, respectivament.

Aquestes funcions verifiquen que el tipus d'usuari de l'usuari que fa la petició té permís per accedir-hi.

En el cas de el tipus d'usuari no sigui vàlid, retorna una resposta HTTP al client amb codi d'estat 401, equivalent a *Unauthorized* (No autoritzat). Per altre banda, si el tipus d'usuari que realitza la petició és vàlid, la petició procedirà en l'execució.

Cal apuntar que aquests *middlewares* s'han de cridar sempre posteriorment al *middleware* de **CheckAuth**, ja que aquest és l'encarregat de descodificar el *token* i afegir les dades de l'usuari a l'objecte *request*. En cas contrari, el tipus d'usuari serà NULL i, tal i com s'espera, retorna una resposta HTTP al client amb codi d'estat 401, equivalent a *Unauthorized*.

## 6.4. Endpoints de la API

A continuació s'expliquen tots els *endpoints* que ofereix l'API per satisfer les necessitats de l'aplicació. Per a detallar cada petició s'ha seguit el següent format:

<Mètode HTTP> - <Middleware aplicat> - <Ruta relativa/:paràmetres>

<Cos de la petició>

<Explicació>

## Autenticació

### - **POST /auth/login**

*username, password*

Valida que les dades enviades siguin correctes. En cas d'èxit, genera un *token* JWT i retorna una resposta HTTP amb codi d'estat 200 (*OK*) i un JSON amb les dades de l'usuari i el *token* generat. En cas d'error, retorna una resposta HTTP amb codi d'estat 401 (*Unauthorized*) i un missatge d'error.

## Usuaris

### - **GET – checkAuth, checkAdmin - /user**

Recupera l'*id\_client* de l'usuari que fa la petició i retorna un llistat en format JSON de tots els usuaris d'aquell client. En cas de no existir, retorna una resposta amb codi d'estat 404 (*Not found*) i un missatge d'error.

### - **GET – checkAuth, checkAdmin - /user/:id\_user**

Retorna la informació en format JSON de l'usuari amb l'identificador equivalent al enviat per paràmetres. En cas de no existir, retorna una resposta HTTP amb codi d'estat 404 (*Not found*).

### - **GET – checkAuth - /user/:id\_user/groups**

Retorna un llistat en format JSON dels grups i grups pares si s'escau, els quals pertanyen a l'usuari amb l'identificador equivalent al enviat per paràmetres. En cas de no formar part de cap grup, retorna una resposta HTTP amb codi d'estat 404 (*Not found*).

### - **POST – checkAuth, checkAdmin - /user**

*username, password, id\_user\_type*

Crea un usuari amb la informació enviada en el cos de la petició. Aquest nou usuari té el mateix *id\_client* que el de l'usuari administrador que ha fet la petició. Retorna una resposta

HTTP amb codi d'estat 201 (*Created*) i les dades del nou usuari en format JSON. Si existeix un usuari amb l'username enviat, retorna una resposta HTTP amb codi d'estat 409 (*Conflict*) i un missatge d'error.

- **PATCH – checkAuth, checkAdmin - /user/:id\_user**

*username, id\_user\_type*

Permet editar l'usuari amb l'identificador al enviat per paràmetres. Retorna una resposta HTTP amb codi d'estat 200 (*OK*) i la informació actualitzada en format JSON. Si ja existeix un usuari amb l'username enviat, retornar una resposta HTTP amb codi d'estat 409 (*Conflict*) i un missatge d'error.

- **PATCH – checkAuth, checkAdmin - /user/enable/:id\_user**

Permet activar l'usuari amb l'identificador equivalent al enviat per paràmetres. Si no existeix l'usuari amb l'identificador sol·licitat retorna una resposta HTTP amb codi 404 (*Not found*) i un missatge d'error.

- **DELETE – checkAuth, checkAdmin - /user/:id\_user**

Permet desactivar l'usuari amb l'identificador equivalent al enviat per paràmetres. Si no existeix l'usuari amb l'identificador sol·licitat retorna una resposta HTTP amb codi 404 (*Not found*) i un missatge d'error.

## **Grups**

- **GET – checkAuth, checkGestor - /groups**

Retorna un llistat en format JSON amb tota la informació dels grups que pertanyen al id client de l'usuari que realitza la petició. Si no existeix cap grup, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth, checkGestor - /groups/:id\_group**

Retorna la informació en format JSON del grup amb l'identificador equivalent al enviat per paràmetres. Si no existeix cap grup amb aquest identificador, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /groups/:id\_group/messages**

Retorna un llistat en format JSON amb tots els comunicats enviats al grup amb l'identificador equivalent al enviat per paràmetres. Per a cada comunicat indica si ha estat llegit per l'usuari que realitza la petició. En cas de que no hi hagi cap comunicat disponible, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /groups/:id\_group/files**

Retorna un llistat en format JSON amb tots els arxius enviats al grup amb l'identificador equivalent al enviat per paràmetres. En cas de que no hi hagi cap arxiu disponible, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /groups/:id\_group/polls**

Retorna un llistat en format JSON amb totes les enquestes del grup amb l'identificador equivalent al enviat per paràmetres. Per a cada enquesta indica si ha estat l'usuari que realitza la petició ha votat o no. En cas de que no hi hagi cap enquesta disponible, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /groups/:id\_group/groups**

Retorna un llistat en format JSON de tots els grups amb el mateix id client de l'usuari que realitza la petició, i que l'identificador del grup no és el que és enviat per paràmetres. A més, per a cada grup indica si ja és fill del grup enviat per paràmetres. Si no existeix cap grup, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /groups/:id\_group/users**

Retorna un llistat en format JSON de tots els usuaris amb el mateix id\_client de l'usuari que realitza la petició. A més, per a cada usuari indica si ja forma part del grup enviat per

paràmetres. Si no existeix cap usuari, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **POST – checkAuth, checkGestor - /groups**

***group\_name***

Crear un grup amb el nom especificat. Aquest nou grup té el mateix `id_client` que el de l'usuari administrador que ha fet la petició. Retorna una resposta HTTP amb codi d'estat 201 (*Created*) i les dades del nou grup en format JSON. Si no existeix un grup amb el nom del grup enviat, retorna una resposta HTTP amb codi d'estat 409 (*Conflict*) i un missatge d'error.

- **POST – checkAuth, checkGestor - /groups/:id\_group/manage-users/:id\_user**

Valida que l'usuari amb l'identificador enviat per paràmetres no formi part del grup amb l'identificador també enviat per paràmetres. Si és vàlid, l'afegeix. Retorna una resposta HTTP amb codi d'estat 201 (*Created*) i un JSON amb les dades. En cas que ja formi part, retorna una resposta HTTP amb codi d'estat 409 (*Conflict*) i un missatge d'error.

- **POST – checkAuth, checkGestor - /groups/:id\_group/manage-groups/:id\_subgroup**

Valida que el subgrup amb l'identificador enviat per paràmetres no formi part del grup pare amb l'identificador també enviat per paràmetres. Si és vàlid, l'afegeix. Retorna una resposta HTTP amb codi d'estat 201 (*Created*) i un JSON amb les dades. En cas que ja formi part, retorna una resposta HTTP amb codi d'estat 409 (*Conflict*) i un missatge d'error.

- **PATCH – checkAuth, checkGestor - /groups/enable/:id\_group**

Permet activar el grup amb l'identificador equivalent al enviat per paràmetres. Si no existeix el grup amb l'identificador sol·licitat retorna una resposta HTTP amb codi 404 (*Not found*) i un missatge d'error.

- **DELETE – checkAuth, checkGestor - /groups/:id\_group/manage-users/:id\_user**

Elimina l'usuari amb l'identificador enviat per paràmetres del grup amb l'identificador també enviat per paràmetres. En cas de que no en formi part, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **DELETE – checkAuth, checkGestor - /groups/:id\_group/manage-groups/:id\_subgroup**

Elimina el subgrup amb l'identificador enviat per paràmetres del grup pare amb l'identificador també enviat per paràmetres. En cas de que no en formi part, retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **DELETE – checkAuth, checkGestor - /groups/:id\_group**

Permet desactivar el grup amb l'identificador equivalent al enviat per paràmetres. Si no existeix el grup amb l'identificador sol·licitat retorna una resposta HTTP amb codi 404 (*Not found*) i un missatge d'error.

## Comunicats

- **GET – checkAuth, checkGestor - /messages/:id\_message**

Retorna la informació en format JSON del comunicat amb l'identificador equivalent al enviat per paràmetres. En cas de no existir, retorna una resposta HTTP amb codi d'estat 404 (*Not found*).

- **POST – checkAuth - /messages**

*title, content, id\_group*

Crea un comunicat amb les dades enviades en el cos de la petició. Selecciona tots els usuaris del grup destinatari d'aquest missatge, així com tots els usuaris que formen part dels grups fills del grup enviat i afegeix els registres a la taula de lectures amb *date\_read* amb valor NULL. Retorna una resposta HTTP amb un codi d'estat 201 (*Created*) i la informació del missatge en format JSON.

- **PATCH – checkAuth - /messages/:id\_message/read**

Recupera l'id de l'usuari que fa la petició i marca el missatge amb l'identificador equivalent a l'enviat per paràmetres com a llegit per aquest usuari.

## **Arxius**

- **POST – checkAuth - /files**

*title, id\_group, file*

Guarda l'arxiu enviat per l'usuari en el servidor. Un cop guardat l'arxiu, crea un comunicat amb les dades enviades en el cos de la petició i retorna una resposta HTTP amb codi d'estat 201 (*Created*).

## **Enquestes**

- **GET – checkAuth, checkGestor - /polls/:id\_poll**

Retorna la informació en format JSON de l'enquesta amb l'identificador equivalent al enviat per paràmetres. En cas de no existir retorna una resposta HTTP amb codi d'estat 404 (*Not found*) i un missatge d'error.

- **GET – checkAuth - /polls/:id\_poll/42tore42ons**

Retorna un llistat en format JSON amb totes les preguntes i les possibles respostes de l'enquesta amb l'identificador equivalent al enviat per paràmetres.

- **GET – checkAuth - /polls/:id\_poll/answers**

Retorna un llistat en format JSON amb totes les respostes enviades i el número de vegades que ha estat votada pels usuaris de l'enquesta amb l'identificador equivalent al enviat per paràmetres.



- **POST – checkAuth, checkGestor - /polls**

*title, description, 43tore43ons, id\_group*

Crea l'enquesta al grup sol·licitat amb les preguntes i les respostes enviades. Selecciona els usuaris pertinents del grup sol·licitat que seran els candidats de la votació (*Voters*) i crea els registres. Retorna una resposta HTTP amb codi d'estat 201 (*Created*) i un JSON amb la informació de l'enquesta.

- **POST – checkAuth - /polls/:id\_poll/answers**

*answers*

Recupera l'id de l'usuari que fa la petició i comprova, juntament amb l'id de l'enquesta enviada per paràmetres, que aquest usuari té permís per a votar o que no ha votat encara. En cas de que no tingui permís, retorna una resposta HTTP amb codi d'estat 403 (*Forbidden*) i un missatge d'error. En cas correcte, guarda les respostes de l'usuari i actualitza el camp *date\_voted* del votador, de manera que aquest usuari ja no pot tornar a votar per a aquesta enquesta.



## 7. Implementació: Aplicació

Tal i com s'ha definit anteriorment, l'aplicació que ha de satisfer les necessitats dels usuaris finals ha de ser multiplataforma, amb l'objectiu d'oferir aquest servei al màxim número de persones.

### 7.1. Estudi d'opcions

#### **Opció A: Aplicacions natives iOS/Android**

La primera opció a valorar és el desenvolupament per separat de dues aplicacions natives. El principal avantatge de les solucions natives és que disposen del millor rendiment en comparació amb les altres alternatives. Al ser programades exclusivament per la seva plataforma gaudeixen d'un nivell d'optimització més elevat, sobretot en aplicacions que han de suportar grans càrregues de dades.

Les aplicacions natives també permeten al 100% l'ús dels recursos del telèfon, ja siguin funcionalitats avançades de la càmera, sensors de proximitat o l'accés a informació sensible del dispositiu, com podria ser l'agenda de contactes.

El desavantatge més clar d'aquesta opció és l'elevat cost de desenvolupament. Per una banda es necessiten dos entorns de desenvolupament i llenguatges completament diferents per programar cada solució: una solució iOS (Swift) i una solució Android (Java o Kotlin). Directament relacionat amb el problema esmentat, això significa també un augment en el temps de desenvolupament, ja que s'han de desenvolupar per separat. Per últim i no menys important, mencionar el cost de manteniment que tindrà cada aplicació per separat.

Una altre característica que clau en l'elecció de *l'stack* tecnològic a utilitzar és que les aplicacions natives es descarreguen i s'actualitzen a través de les botigues corresponents a cada sistema operatiu, AppStore per a iOS i PlayStore per a Android. Això fa que com a desenvolupador t'hagis d'adaptar a les polítiques pròpies de cada empresa per separat, així com pagar les llicències pertinents i obligar als usuaris a passar per la botiga corresponent per descarregar o actualitzar l'aplicació.

## Opció B: Aplicació híbrida iOS/Android

Una altre opció a valorar es el desenvolupament d'una aplicació híbrida. Aquest tipus d'aplicacions solen compilar el mateix codi font tant per Android com per iOS. Cal esmentar que al ser aplicacions compilades a codi natiu requereixen fer tant la descàrrega com l'actualització de l'aplicació a través de la Store pertinent.

### React Native

Tecnologia/Framework desenvolupat per Facebook. Utilitza el llenguatge Javascript de la mà de la llibreria React. És la solució més popular i una de les més madures ja que té un ecosistema i una comunitat molt important.

### NativeScript

Aquesta tecnologia també utilitza Javascript com a llenguatge per programar les aplicacions. Dins del mateix llenguatge Javascript, ofereix suport per Angular, Vue.js i Javascript pur, cada una independent de l'altre.

Ofereix una sèrie de *plugins*, desenvolupats pel mateix equip de NativeScript, per accedir a les característiques natives d'els dispositius. És la solució menys popular dins la comunitat ja que és relativament nova i és immadura.

### Flutter

Aquesta nova tecnologia desenvolupada per Google és tant un *framework* del llenguatge Dart com un SDK (Software development kit). A diferència de les altres opcions, Flutter proveeix una sèrie de components ja preparats per crear les interfícies gràfiques basats en Material Design.

Necessita de paquets externs per accedir a algunes funcionalitats natives. Degut a la seva immaduresa es troben a faltar funcionalitats avançades dels dispositius.

És una de les solucions més innovadores i a la vegada més populars degut a que Google hi està apostant molt.

## **Ionic**

Ionic treballa de manera diferent a les tecnologies esmentades anteriorment. Aquesta tecnologia permet crear d'una simple aplicació web que és embolcallada per una aplicació real nativa que conté un *webview*. És una solució molt senzilla ja que realment és una aplicació web.

Proveeix de paquets externs per accedir a les funcionalitats natives dels dispositius.

## **Opció C: Aplicació Web Progressiva (PWA)**

La última opció a proposar és una de les tecnologies que està revolucionant el concepte d'aplicació mòbil. Les aplicacions web progressives, d'aquí en endavant PWA, són aplicacions web dissenyades per treballar com aplicacions natives.

Les PWA poden ser afegides al escriptori dels dispositius mòbils, ofereixen una experiència molt similar a una aplicació nativa: treballen pantalla completa, funcionen sense connexió, permeten notificacions push al dispositiu, entre d'altres.

L'avantatge principal d'aquest tipus d'aplicacions és que funcionen de la mateixa forma que les aplicacions web clàssiques de tal manera que qualsevol navegador de qualsevol dispositiu les pot executar. Això permet utilitzar un mateix codi font per a tots els dispositius, tant Android, iOS o desktop.

Un dels avantatges més importants respecte les altres solucions proposades anteriorment és que al ser una aplicació web no requereix cap tipus d'instal·lació i/o actualització a través de les botigues corresponents, l'usuari sempre treballa amb l'última versió desplegada en el servidor.

Un dels obstacles que es poden trobar als usuaris és com saber de quina manera s'afegeix la PWA a l'escriptori. Els sistemes Android automatitzen aquesta feina mostrant un avís quan accedeixes a una PWA, amb un accés directe per instal·lar-la. Per contra, els dispositius iOS no permeten aquesta funcionalitat nativament, pel qual ho han de fer manualment.

El desavantatge més clar que tenen aquestes aplicacions és que de moment no permeten enviar notificacions push en els dispositius iOS. Això es deu a les polítiques establertes per

Apple. Tot i que en les últimes actualitzacions del sistema operatiu han anat donant més i més suport a les PWA, cal esperar per veure com reaccionen a aquesta funcionalitat.

## 7.2. Elecció

Tenint com un dels requeriments tècnics principals que aquest servei sigui una solució multiplataforma i considerant les opcions tècniques prèviament esmentades, s'ha optat per desenvolupar una PWA. Aquesta tècnica de desenvolupament permet a les aplicacions web oferir una experiència molt similar a la d'una aplicació nativa. També és important esmentar que al no ser una aplicació que requereix de funcionalitats avançades del dispositiu mòbil, la PWA s'adapta perfectament en aquest cas. A més, permet oferir als usuaris la consulta de la informació i suport *offline*.

Els avantatges clau respecte les altres tecnologies són:

- Mateix codi font per navegador clàssic, iOS i Android.
- Independència de les botigues i les polítiques d'Apple i Google.
- Els usuaris sempre gaudeixen de la versió actual del servidor, pel qual no hauran d'actualitzar manualment l'aplicació.

El terme PWA no és una tecnologia en concret, sinó que agrupa una sèrie de característiques independents que fan gaudir a una aplicació web de característiques d'aplicacions natives. Per enriquir l'experiència de l'usuari amb una navegabilitat més fluida i amb més sensació d'aplicació, desenvoluparem la PWA amb React, que ens facilitarà la creació d'una SPA (Single Page Application).

## 7.3. Entorn de desenvolupament

React és una llibreria de Javascript per a la creació d'interfícies d'usuari. Un dels avantatges que ofereix la construcció d'una SPA (Single Page Application) és la manera en què treballa.

Les SPA es caracteritzen per gestionar de manera més ràpida les interaccions de l'usuari que les webs tradicionals. Tot i tenir una càrrega inicial més significativa que una web tradicional degut a que es carrega a la vegada tota la interfície en el client, en aplicacions on l'usuari hi

té molta interacció és molt més eficient ja que les càrregues posteriors són molt més ràpides; les dades es carreguen del servidor de manera asíncrona.

El llenguatge que utilitza React per el desenvolupament és Javascript. De la mateixa manera que amb el desenvolupament de la API, s'han gestionat les dependències a través de NPM.

Per a començar el desenvolupament amb React, s'ha instal·lat una eina anomenada **create-react-app**, gestionada per Facebook, i que proveeix al programador d'un esquelet ja configurat sobre el qual començar a desenvolupar, com si es tractés d'un framework. S'ha instal·lat globalment a la nostre màquina a través de NPM.

## 7.4. Dos conceptes claus de React

Un dels conceptes claus de React són els **components**. Els components permeten dividir la interfície d'usuari en peces independents i reutilitzables. A més, els components reaccionen a canvis d'estat de l'aplicació, actualitzant-se automàticament.

El nucli d'un component de React és l'**estat**. L'estat és un objecte que, a través de les seves propietats, determina com es comporta i com es renderitza cada component.

## 7.5. Llibreries de suport principals

### Material UI

Aquesta llibreria proporciona una sèrie de components visuals basats en el disseny Material de Google. Facilita la tasca al programador alhora de crear les interfícies visuals. A més, dóna una sensació similar a una aplicació nativa.

### Redux i React-Redux

La llibreria **Redux** permet centralitzar tot l'estat de l'aplicació en un sol objecte. En aplicacions SPA complexes, la correcte manipulació de l'estat és important per evitar comportaments estranys de l'aplicació. Centralitzant la gestió de l'estat en un sol objecte i manipulant-lo a través d'accions, permet tenir un control sobre aquest.

Cal apuntar que la llibreria Redux no està lligada a React. Per això s'ha necessitat la llibreria **React-Redux**, que permet lligar aquest contenidor d'estat a l'aplicació React. La manté el mateix equip de Redux, oferint sempre versions actualitzades de les APIs de Redux i React.

És important tenir clars els conceptes principals de Redux per entendre el cicle de vida d'una aplicació Redux.

- Store: És el contenidor on es guarda tot l'estat de l'aplicació.
- Action: Representa un tipus d'event. Un objecte *action* té el següent format:
  - {type: MUST, payload: OPTIONAL}
- Action Creator: És una funció que retorna un objecte *action*.
- Reducer: Especificquen com canvia l'estat de l'aplicació en resposta a una acció enviada a la *store*. Els reducers només calculen el següent estat i ha de ser totalment previsible: cridar-lo varies vegades amb les mateixes entrades ha de produir sempre la mateixa sortida.

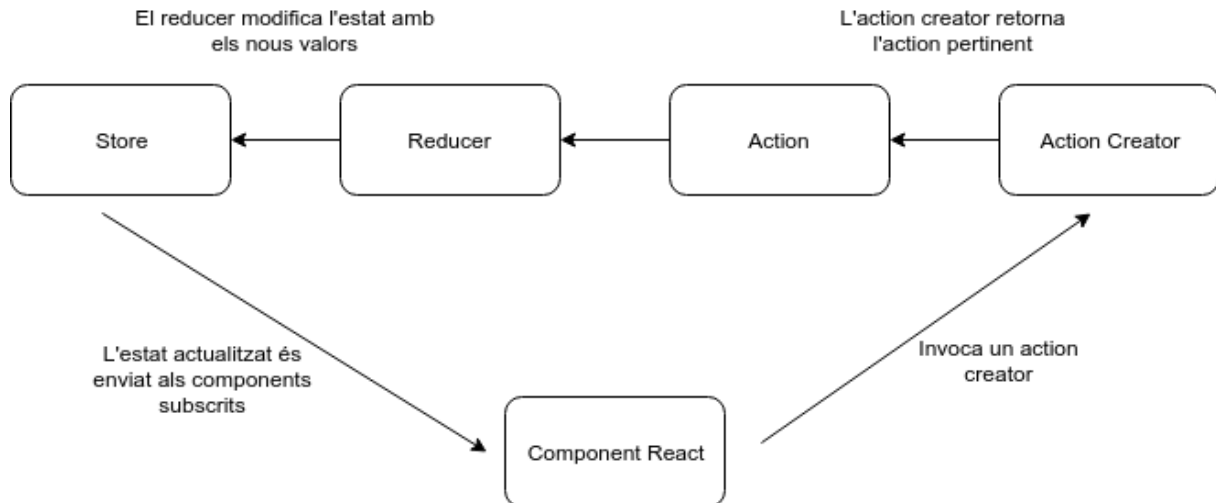


Figura 7.1 Flux d'una aplicació Redux



## 7.6. Funcionalitats de l'aplicació

Una vegada definits tots els requeriments i casos d'ús que té l'aplicació i s'ha decidit la tecnologia a utilitzar, s'han començat a desenvolupar de forma incremental tots els mòduls que integraran l'aplicació final.

A continuació es detallen, juntament amb imatges explicatives, les funcionalitats de l'aplicació.

### Inici de sessió

El punt d'entrada a l'aplicació per a tots els usuaris és la pantalla d'inici de sessió. Aquesta consta d'un petit formulari que demana l'usuari i la contrasenya per accedir, tal i com mostra la Figura 7.2. En cas d'utilitzar unes claus errònies, llença un missatge d'error. En cas satisfactori, l'usuari pot accedir a les funcionalitats permeses, segons el tipus d'usuari que és.

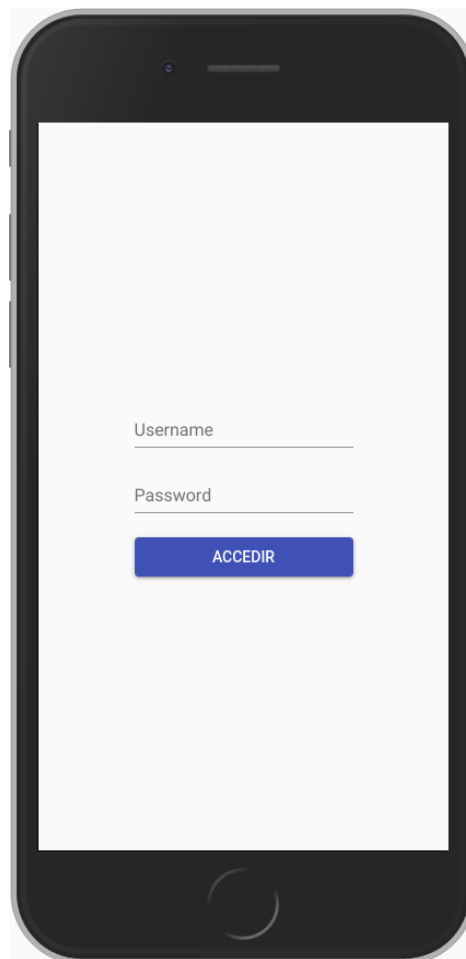


FIGURA 7.2 APLICACIÓ: PANTALLA D'INICI DE SESSIÓ

## Pantalla d'inici

Una vegada l'usuari ha accedit satisfactòriament, es mostra la vista principal de l'aplicació. Aquesta pantalla consta de diferents components, tal i com es pot veure en la Figura 7.3.

Per una banda, la barra superior que està formada per:

- Un botó que desplega un menú lateral que mostra diferents opcions depenent del rol de l'usuari que ha accedit a l'aplicació i un menú desplegable .
- Un menú desplegable que conté tots els grups els quals forma part l'usuari.

Per altre banda, un menú de navegació permet navegar a les pantalles de missatges, arxius i votacions.

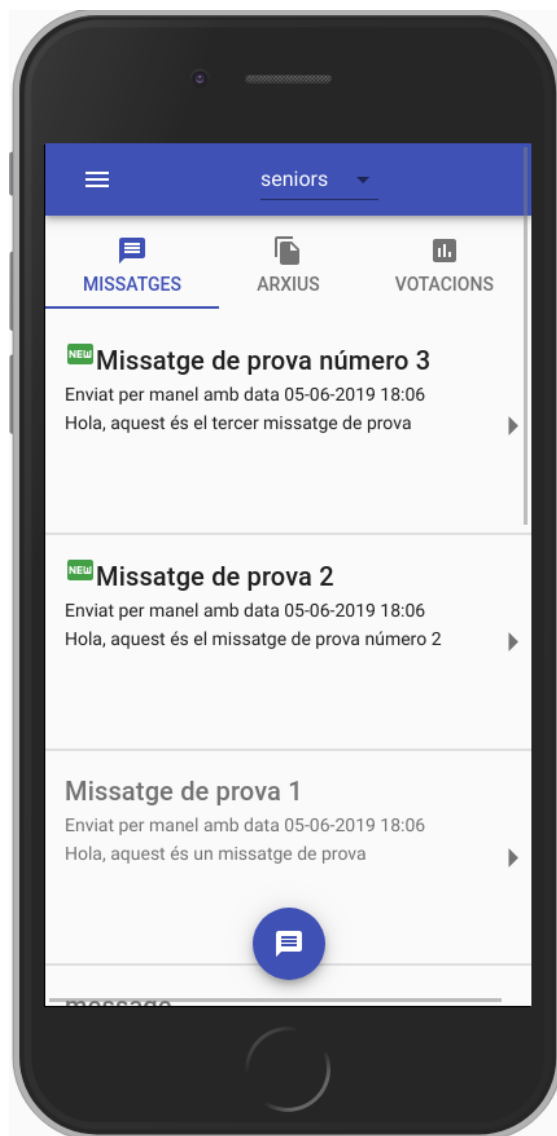


FIGURA 7.3 LLISTAT DE MISSATGES

## Llistat de missatges

Quan es clica sobre l'ítem Missatges del menú de navegació, l'aplicació mostra el llistat de missatges del grup seleccionat en el menú superior. En cas de canviar de grup, els missatges s'actualitzen automàticament amb els del nou grup.

Aquest llistat mostra els missatges de dues maneres, depenent de si l'usuari ja ha llegit el missatge o no. En cas de que no l'hagi llegit, mostra un petit rètol avisant de que és un missatge nou. Si l'usuari ja l'ha llegit, l'ítem de la llista de missatges es mostra en un to de color més suau. Es pot veure en la Figura 7.3.

Si no existeix cap missatge pel grup seleccionat, l'aplicació mostra un missatge avisant que no existeix cap comunicat, tal i com es pot veure en la Figura 7.4.

Si l'usuari té, com a mínim, el rol de Gestor, es mostra un botó a la part inferior central per enviar un nou missatge al grup seleccionat.

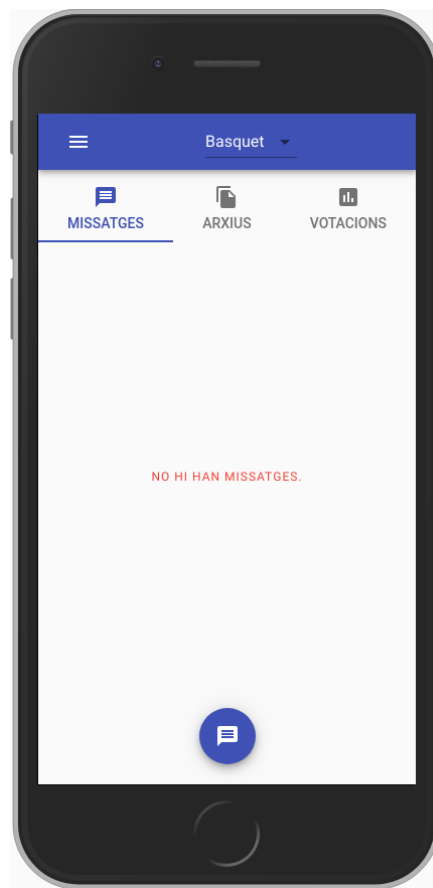


FIGURA 7.4 LLISTAT SENSE MISSATGES



**FIGURA 7.5 DETALL D'UN MISSATGE**

### **Veure detall d'un missatge**

Fent clic a un ítem de la llista de missatges, es mostra una pantalla simple amb el detall d'aquest, tal i com es pot veure en la Figura 7.5.

### **Enviar un missatge**

A aquesta pantalla només hi poden accedir els usuaris que tinguin, com a mínim, el rol de Gestor. Permet omplir un formulari per a enviar un missatge al grup seleccionat en la barra superior, tal i com es pot veure en la Figura 7.6. No deixa enviar el formulari si no passa la validació.



FIGURA 7.6 ENVIAMENT DE MISSATGE

### Llistat d'arxius

Quan es clica sobre l'ítem Arxius del menú de navegació, l'aplicació mostra el llistat d'arxius enviats al grup seleccionat en el menú superior. En cas de canviar de grup, els arxius s'actualitzarien automàticament amb els del nou grup.

Cada ítem del llistat és un arxiu. En mostra la informació principal i un botó per a descarregar-lo, tal i com es pot veure en la Figura 7.7

Si no s'ha enviat cap arxiu al grup seleccionat, l'aplicació mostra un missatge avisant de que no existeix cap arxiu.

Es mostra un botó a la part inferior central per a enviar un arxiu al grup seleccionat. Qualsevol usuari que sigui membre del grup té permís per enviar.

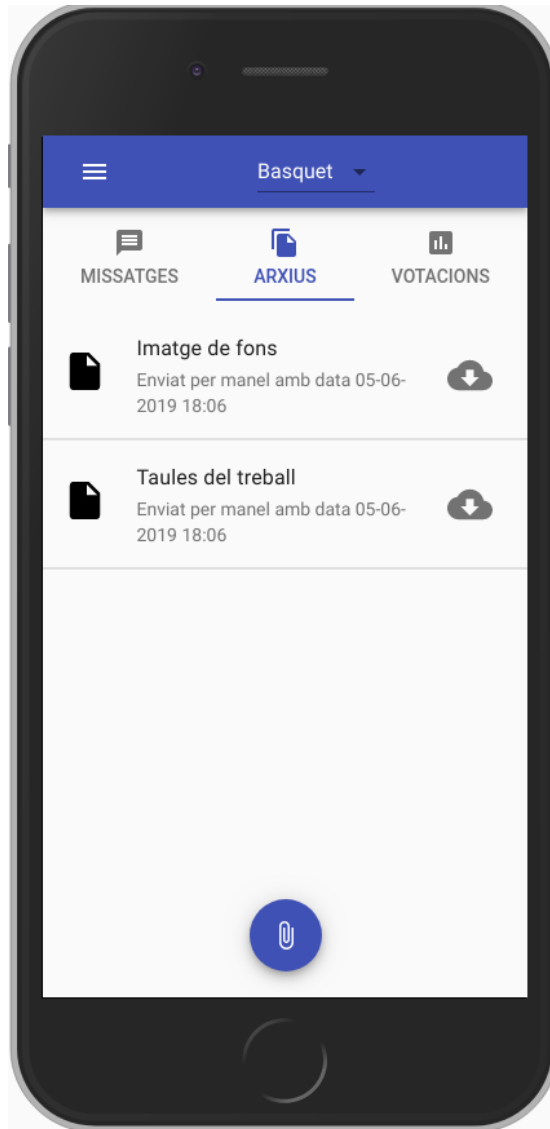


FIGURA 7.7 LLISTAT D'ARXIU

### Llistat de votacions

Quan es clica sobre l'ítem Votacions del menú de navegació, l'aplicació mostra el llistat d'enquestes enviades al grup seleccionat en el menú superior. En cas de canviar de grup, les enquestes s'actualitzarien automàticament amb els del nou grup.

Cada ítem del llistat és una enquesta diferent. Tal i com mostra la Figura 7.8, es mostra la informació principal i dos possibles botons:

- **Veure resultats**
- Si l'usuari no ha votat aquella enquesta i té permís per a fer-ho, mostra el botó **Votar**

Si no s'ha creat cap enquesta pel grup seleccionat, l'aplicació mostra un missatge avisant de que no existeix cap.

Si l'usuari té, com a mínim, el rol de Gestor, es mostra un botó a la part inferior central per a crear una nova enquesta al grup seleccionat.

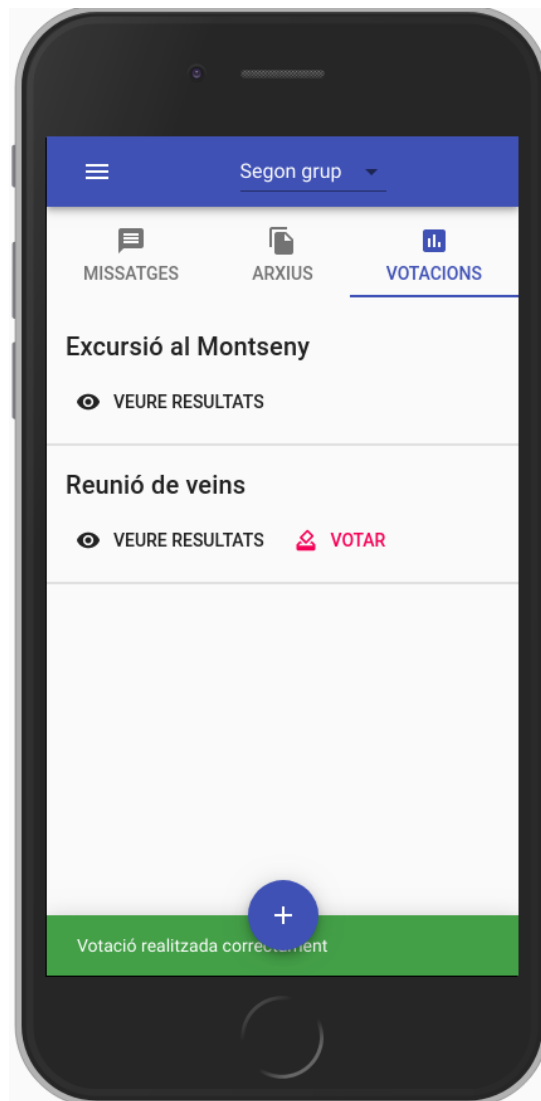


FIGURA 7.8 LLISTAT DE VOTACIONS

### Crear una votació

Aquesta pantalla permet la creació completa d'una enquesta. Permet definir un títol i una descripció opcional per identificar-la. S'ha desenvolupat un formulari dinàmic per afegir o eliminar preguntes i respostes, tal i com es pot veure en la Figura 7.9.



FIGURA 7.9 CREACIÓ D'UNA VOTACIÓ

### Realitzar votació

Genera un formulari i permet a un usuari vàlid per a votar en aquella enquesta respondre les preguntes. Obliga a l'usuari a respondre-les totes abans de guardar-les. En cas contrari, la votació no es guarda i l'usuari manté la validesa per a votar més endavant. Es pot veure en la Figura 7.10.



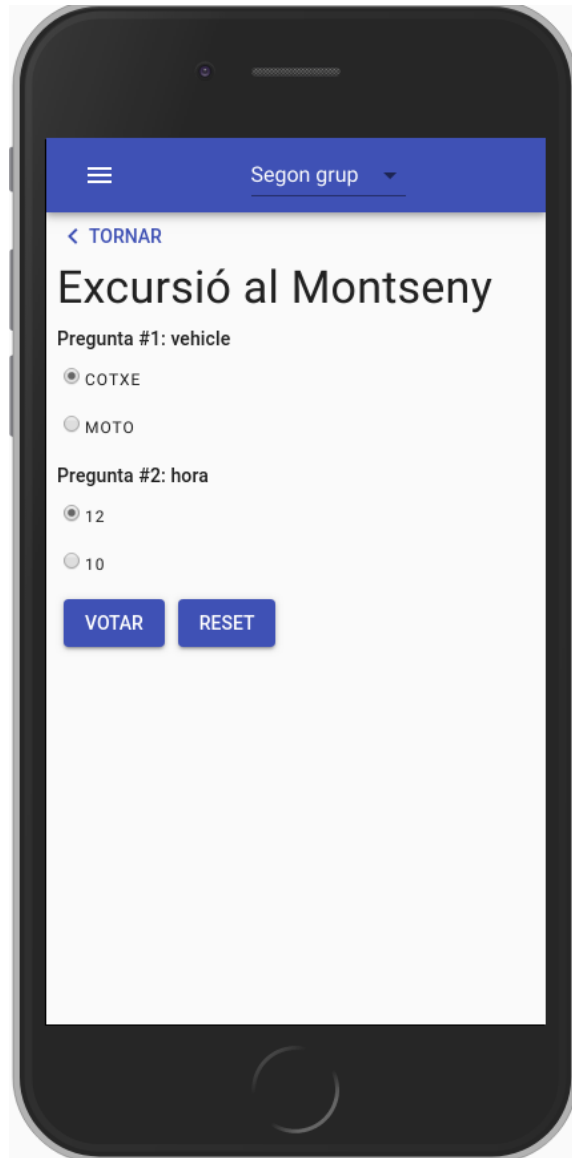


FIGURA 7.10 REALITZACIÓ D'UNA VOTACIÓ

**Veure resultat d'una votació**

Aquesta pantalla només s'hi pot accedir una vegada existeix, com a mínim, una resposta a aquella votació guardada. Si no existeix cap resposta en el moment d'accés, envia a l'usuari al llistat de votacions. Permet veure per a cada pregunta un gràfic amb les respostes i una llegenda, tal i com es pot veure en la Figura 7.11.

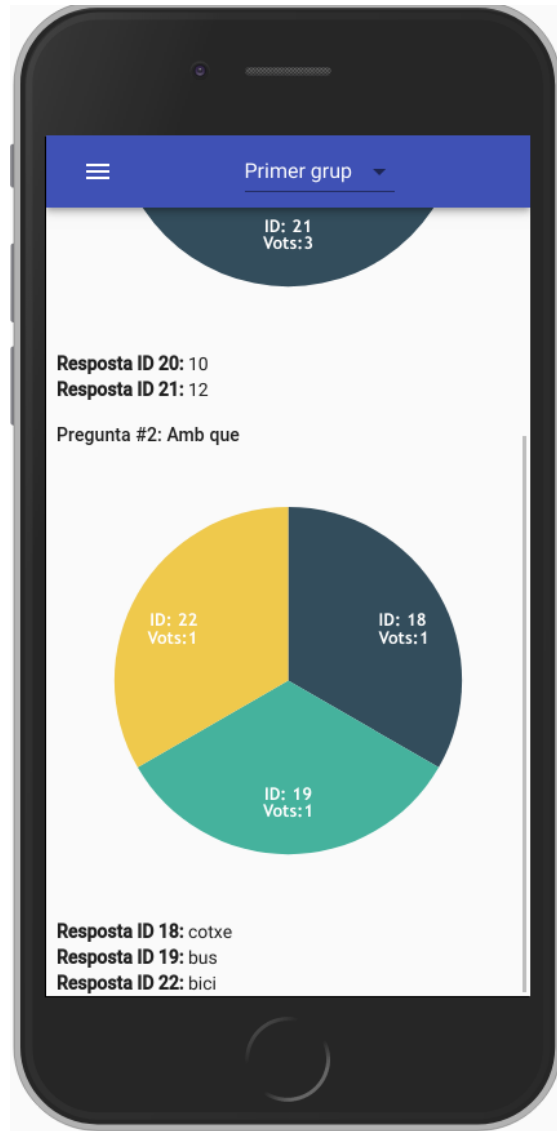


FIGURA 7.11 VEURE EL RESULTAT D'UNA VOTACIÓ

### Menú lateral

Des del menú desplegable lateral es pot accedir a les pantalles de gestió de grups i de gestió d'usuaris, tal i com es mostra en la Figura 7.12.

El botó de **Grups** només el veuen els usuaris que tenen, com a mínim, el rol de gestor.

Per altre banda, el botó d'**Usuaris** només el veuen els usuaris administradors.

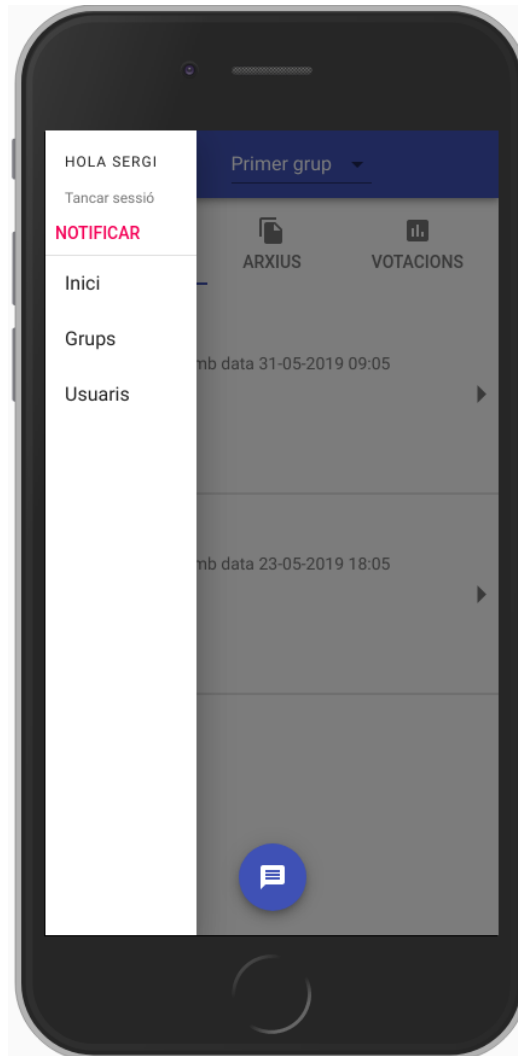


FIGURA 7.12 MENÚ LATERAL

### Llistat de gestió de grups

Aquesta pantalla permet gestionar tots els grups del client el qual pertany l'usuari. Tal i com s'ha comentat anteriorment, només poden accedir a aquesta pantalla els usuaris que, com a mínim, tenen el rol de Gestor. En cas contrari, són redirigits a la pantalla d'inici.

Permet activar i desactivar grups a través d'un interruptor. També és el punt d'entrada a la pantalla per a gestionar els integrants del grup. Es pot veure en la Figura 7.13.

A la part inferior central hi ha un botó per accedir a la creació de grups.

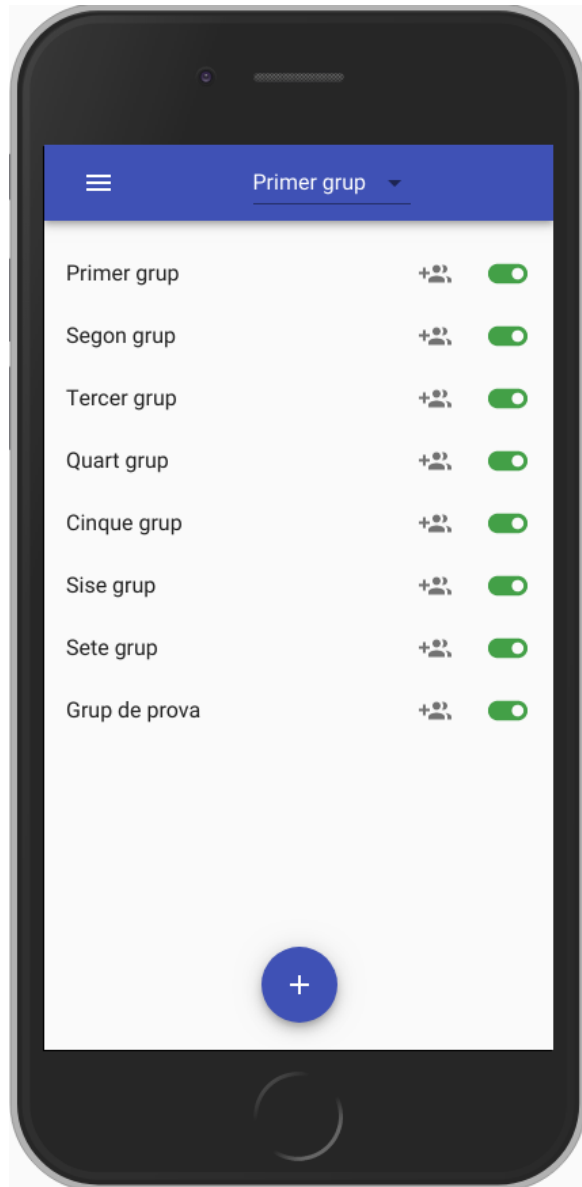


FIGURA 7.13 LLISTAT DE GESTIÓ DE GRUPS

### Crear un grup

Aquesta pantalla mostra un formulari per a crear un grup nou del client el qual pertany l'usuari, tal i com mostra la Figura 7.14. En cas de que el nom del grup ja existeixi, llençarà un missatge d'error.

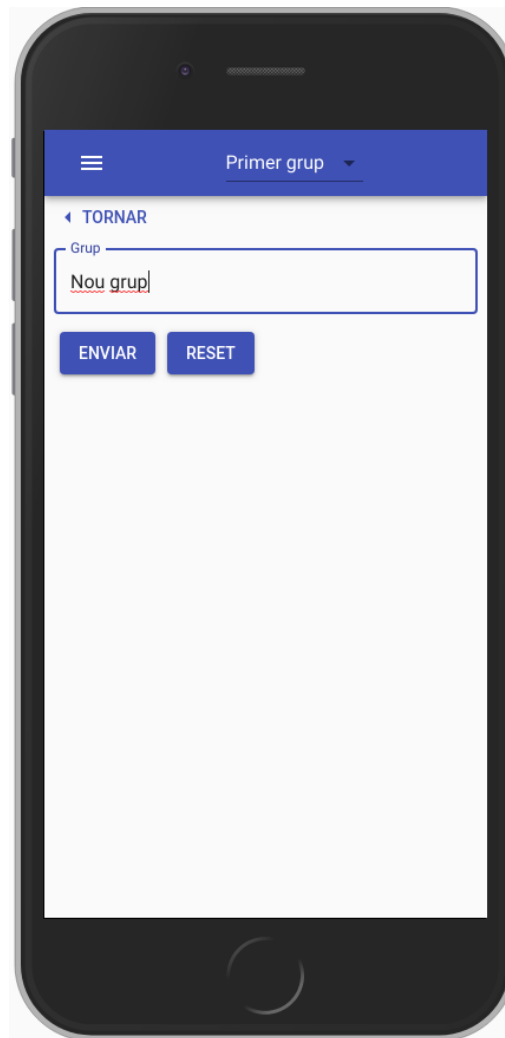


FIGURA 7.14 CREACIÓ D'UN GRUP

### **Afegir usuaris/grups a grups**

Aquesta pantalla permet gestionar els integrants del grup, tant usuaris com grups de grups a través d'interruptors. S'han separat els usuaris dels grups en dues pestanyes diferents, tal i com es mostra en la Figura 7.15.

### **Llistat de gestió d'usuaris**

Aquesta pantalla permet gestionar tots els usuaris del client el qual pertany l'usuari. Tal i com s'ha comentat anteriorment, només poden accedir a aquesta pantalla els usuaris que tenen el rol d'administrador. En cas contrari, són redirigits a la pantalla d'inici.

Permet activar i desactivar usuaris a través d'un interruptor. Es pot veure en la Figura 7.16.

A la part inferior central hi ha un botó per accedir a la creació d'usuaris.

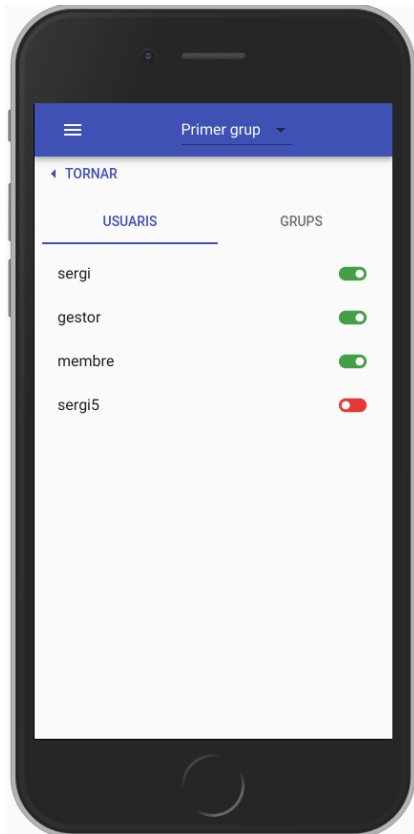


FIGURA 7.15 AFEGIR USUARIS/GRUPS A GRUPS

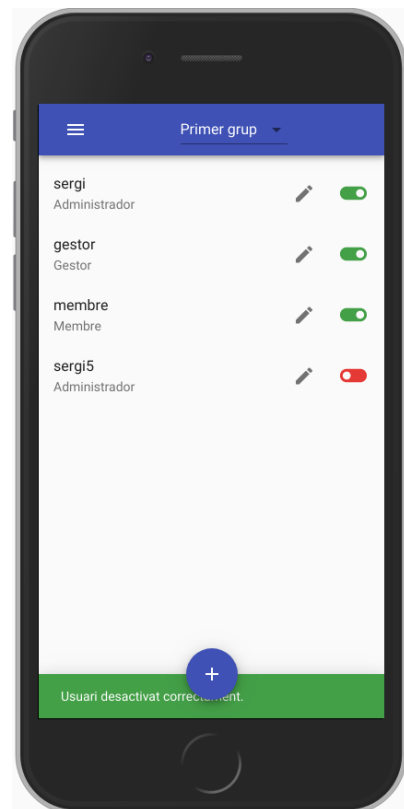
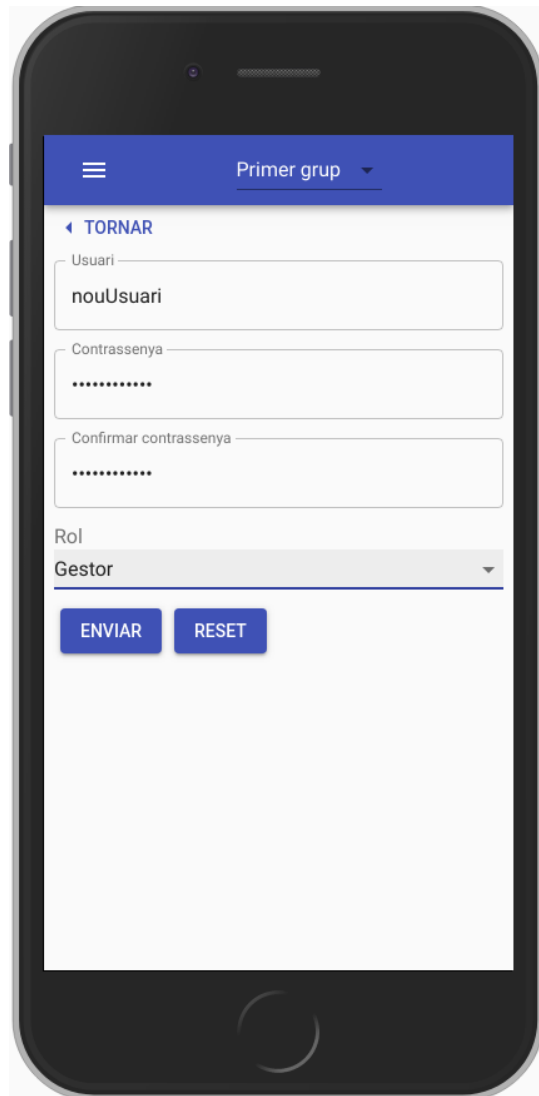


FIGURA 7.16 LLISTAT D'USUARIS

## Crear usuari

Aquesta pantalla mostra un formulari per a crear un usuari nou del client el qual pertany l'usuari, tal i com mostra la Figura 7.14. En cas de que el nom de l'usuari ja existeixi, llença un missatge d'error.



The image shows a smartphone screen displaying a user creation form. The form is titled "Primer grup" and has a blue header bar with a hamburger menu icon on the left. Below the header, there is a section titled "TORNAR" with a back arrow. The form contains three input fields: "Usuari" with the text "nouUsuari", "Contrassenya" with a masked password "\*\*\*\*\*", and "Confirmar contrassenya" with a masked password "\*\*\*\*\*". Below these fields is a "Rol" dropdown menu currently set to "Gestor". At the bottom of the form are two blue buttons: "ENVIAR" and "RESET".

FIGURA 7.17 CREACIÓ D'UN USUARI





## 8. Funcionalitats PWA

Una vegada implementats els casos d'ús de l'aplicació web, s'han afegit les funcionalitats progressives de les PWA. Són varies les tecnologies que permeten proveir funcionalitats considerades natives a les aplicacions web.

Per aquest projecte en concret s'han implementat tres d'aquestes tecnologies, molt relacionades entre elles, que s'expliquen en els següents apartats:

- S'ha definit un web **Manifest**.
- S'ha implementat un **service worker**.
- S'han implementat **notificacions push** als dispositius.

Anteriorment, s'ha fet referència a la eina **create-react-app**. Aquesta eina incorpora de sèrie dues d'aquestes tecnologies, facilitant molt la feina al desenvolupador ja que l'esquelet ve preparat per a funcionar com una PWA. Les dues tecnologies que incorpora són el **manifest** i el **service worker**.

### 8.1. Manifest

El **manifest** és un arxiu en format JSON que es troba a l'arrel del projecte. Aquest arxiu permet fer les aplicacions web instal·lables en els escriptoris dels dispositius mòbils.

L'arxiu manifest conté una sèrie de propietats que són interpretades pel navegador quan accedim a l'aplicació. És important que implementi correctament aquestes propietats per tal d'oferir la millor experiència a l'usuari.

Les propietats que s'han considerat mínimes per oferir la millor experiència són:

- **name**: És el nom que es mostra a la pantalla de càrrega al obrir l'aplicació una vegada instal·lada.
- **short\_name**: És el nom que es mostra, per defecte, sota la icona de l'aplicació.
- **display**: Aquesta propietat ha de prendre el valor de *standalone*. És la propietat que fa l'aplicació web semblar-se a una aplicació nativa. Una vegada s'ha instal·lat la aplicació a l'escriptori, la aplicació s'executa a pantalla completa.
- **background\_color**: Defineix el color de fons de la pantalla de càrrega.

- icons: Permet definir les icones que utilitza l'aplicació. Es poden definir varies icones de varies mides per a diferents mides de pantalla.

Aquest arxiu manifest ha d'estar vinculat a l'aplicació web a través d'una etiqueta *link* a la capçalera de l'arxiu HTML. Tal com s'ha comentat, l'arxiu **manifest** ja ve incorporat per defecte amb l'esquelet de l'aplicació React. Només s'han hagut de modificar les dades per a què satisfessin les necessitats d'aquest projecte.

## 8.2. Javascript i service workers

Com qualsevol aplicació web, els fitxers Javascript són carregats per les pàgines HTML. Per entendre el seu funcionament, és clau saber de quina manera treballa Javascript. Javascript és un llenguatge que treballa sobre un únic *thread*. Això significa que alhora que, per exemple, visites una aplicació web i aquesta et retorna una pàgina HTML que executa Javascript, tot aquest codi Javascript carregat per la pàgina funciona sobre un únic *thread*. Tot i carregar diferents arxius Javascript, aquests compartiran el mateix *thread*. Aquest arxius Javascript són els que poden accedir al DOM i manipular-lo.

Per altre banda, els *service workers* també són arxius Javascript, però tenen accés a un grup diferent de característiques. És important saber que, de la mateixa manera, treballen sobre un únic *thread* separat. Aquest *thread* no és compartit amb el *thread* que utilitza el Javascript carregat pels fitxer HTML. Els *service workers* tenen el seu propi *thread* ja que treballen en segon pla, pel qual vol dir que estan desacoblats de les pàgines HTML i no poden, per exemple, accedir i manipular el DOM.

A més, els *service workers* segueixen treballant en segon pla tot i haver tancat totes les pàgines del navegador.

Aquesta manera de treballar en segon pla dels *service workers* s'aprofita per escoltar events. Aquests arxius tenen la capacitat d'escoltar tant events llençats per les pàgines HTML/Javascript com per events llençats des de un altre servidor, com per exemple, les notificacions web push.

### 8.2.1. Caching i suport offline

Per oferir una bona experiència d'usuari, una PWA ha de carregar la UI de forma gairebé instantània sense importar l'estat de connexió de l'aplicació.

Un avantatge d'utilitzar la eina **create-react-app** és que incorpora de sèrie tot el mecanisme de caché d'arxius estàtics a través del *service worker*. Quan l'usuari accedeix per primer cop a l'aplicació es guardaran a la caché del navegador aquests arxius, fent que les càrregues posteriors siguin més lleugeres. De la mateixa manera, l'aplicació podrà accedir a aquests arxius tot i no tenir connexió.

Per altre banda, les peticions que fa el client es van guardant de manera incremental a la Store Redux, de manera que si l'usuari no té connexió o té una connexió lenta, mostrarà les últimes dades guardades.

### 8.2.2. Notificacions push

Una de les funcionalitats clau de l'aplicació és poder enviar notificacions push als usuaris amb la finalitat de mantenir-los al dia de l'activitat dels grups el qual pertany. Per assolir-ho, s'ha implementat el sistema de notificacions push que ofereix la Web Push API.

Les notificacions push és un servei que ha d'estar sempre actiu escoltant les notificacions entrants. És per això que aquesta funcionalitat s'ha implementat en un *service worker*, ja que aquests funcionen en segon pla quan l'aplicació no està executant-se.

En aquest projecte s'han implementat notificacions push i per tant, avisa als usuaris corresponents, cada vegada que s'envia un missatge a un grup, cada vegada que es crea una votació nova i quan s'afegeix o s'elimina un usuari d'un grup.

Les notificacions push funcionen a través d'uns *endpoints* únics per dispositiu i navegador que ens proporciona la pròpia push API, els quals són les direccions on s'envien les notificacions des del servidor. Quan es sol·liciten permisos al usuari i aquest els accepta, es genera una nova clau de subscripció.

#### **Mecanisme de seguretat de les subscripcions a les notificacions**

S'ha implementat un sistema de seguretat per les notificacions push que protegeix els usuaris de l'enviament de notificacions push externes al nostre servidor *back end*.

Tal i com s'ha explicat anteriorment, les notificacions push s'envien individualment a una URL el qual cada usuari hi està subscript. Si per algun motiu algun usuari atacant amb males intencions pogués aconseguir accés a aquestes URLs, podria enviar notificacions no desitjades als usuaris des de un servidor extern a aquesta plataforma. És per això que s'ha assegurat que la única font vàlida per enviar notificacions és el propi servidor *back end*.

S'ha aconseguit a través d'un mecanisme anomenat VAPID que funciona a través d'una clau pública i una clau privada. Les subscripcions es creen tenint en compte la clau pública i les notificacions s'envien havent configurat prèviament aquestes dues claus. En el cas que s'enviessin notificacions als *endpoints* de cada usuari sense firmar, aquestes no serien rebudes per l'usuari. D'aquesta manera s'ha aconseguit protegir-los de contingut no desitjat.

## 9. Servidor i hosting

Com tota aplicació web, aquesta ha d'estar allotjada en un servidor per a que s'hi pugui accedir públicament. S'ha contractat un servidor a la plataforma Amazon Web Services (AWS) i un nom de domini a la plataforma Google Domains per a satisfer totes les necessitats requerides.

El nom de domini contractat és *sergitubert.dev*.

La IP pública del servidor contractat és 35.181.78.181.

Per a configurar correctament el servidor s'han seguit una sèrie d'instruccions per garantir el correcte funcionament de l'aplicació.

### 9.1. Configuració prèvia del servidor

Primerament s'ha configurat el *firewall* del servidor per tal de poder accedir a través de SSH, HTTP i HTTPS (Figura 9.1) i s'ha vinculat el nom de domini a la IP de la màquina. S'han configurat:

- Dos registres A, amb els noms *sergitubert.dev* i [www.sergitubert.dev](http://www.sergitubert.dev) i la IP pública del servidor com a valor.
- Els name servers (NS), assignats automàticament per Amazon.
- L'Start of Authority (SOA), també assignat automàticament per Amazon.

Es pot veure la configuració dels registres a la Figura 9.2


Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0

**FIGURA 9.1 CONFIGURACIÓ DEL FIREWALL**

<input type="checkbox"/>	Nombre	Tipo	Valor
<input type="checkbox"/>	sergitubert.dev.	A	35.181.78.185
<input type="checkbox"/>	sergitubert.dev.	NS	ns-370.awsdns-46.com. ns-1285.awsdns-32.org. ns-1608.awsdns-09.co.uk. ns-1016.awsdns-63.net.
<input type="checkbox"/>	sergitubert.dev.	SOA	ns-1285.awsdns-32.org. awsdns-hostmaster.amazo
<input type="checkbox"/>	www.sergitubert.dev.	A	35.181.78.185

**FIGURA 9.2 CONFIGURACIÓ DE REGISTRES**

- Usar los servidores de nombres de Google Domains  
 Utilizar servidores de nombres personalizados

Servidor de nombres 

ns-370.awsdns-46.com

ns-1285.awsdns-32.org

ns-1608.awsdns-09.co.uk

ns-1016.awsdns-63.net

**FIGURA 9.3 SERVIDORS DE NOMS**

Per últim, s'han vinculat els servidors de noms (registres NS) al domini, tal com mostra la Figura 9.3. Cal esmentar que la resolució de noms no és instantània, pot tardar fins a 72 hores a resoldre correctament un nom.

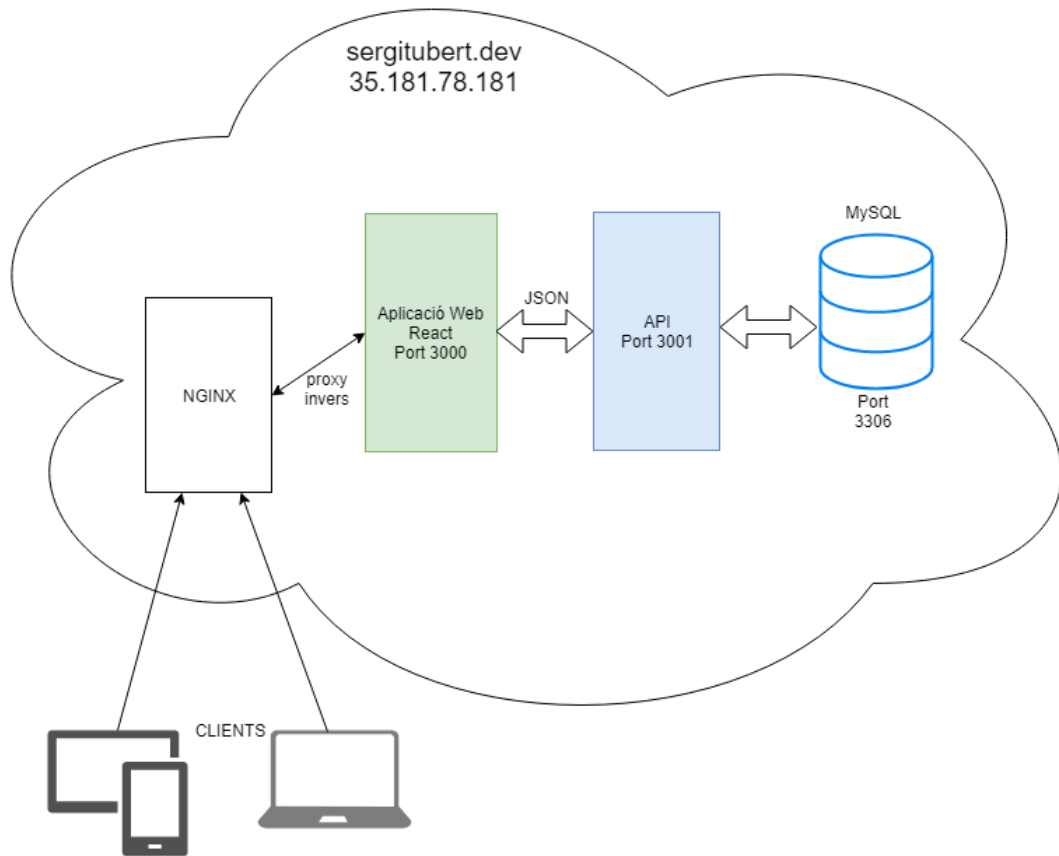


FIGURA 9.4 ARQUITECTURA CLIENT-SERVIDOR

## 9.2. Instal·lació i configuració dels serveis

### 1. MySQL

El primer servei a instal·lar ha estat el servidor MySQL degut a la seva facilitat i independència amb la resta de serveis. Aquest servei corre sobre el port 3306.

S'ha executat l'script de creació de base de dades importat des de la màquina de desenvolupament local per a crear el model que utilitza l'aplicació.

### 2. Nginx

S'ha utilitzat el servidor web Nginx com a proxy invers per a rebre les peticions externes i enviar-les a una xarxa interna del servidor.

Les dues aplicacions que se serveixen són l'aplicació web progressiva React en el port 3000 i la API RESTful que s'utilitza com a origen de dades en el port 3001.

El proxy invers ha servit per a poder fer d'intermediari entre els clients i l'aplicació servida en el port 3000, la única que s'hi pot accedir públicament.

### **3. Certificat HTTPS**

S'ha instal·lat **Certbot** per a gestionar els certificats HTTPS del servidor. **Certbot** automatitza la gestió dels certificats configurant el servidor web, en aquest cas NGINX, i facilitant-ne la seva creació.

Actualment és molt important disposar de certificats HTTPS per tal d'oferir la millor experiència a l'usuari. Per una part s'assegura que les peticions a la xarxa es realitzen de manera encriptada i no en text pla. Per altre banda, els navegadors d'avui en dia intercepten i limiten cada dia més la navegació per webs sense certificat HTTPS.

És un requisit indispensable enviar les notificacions push des de un domini HTTPS, en cas contrari, aquestes no s'enviaran. Per tant, és crític per aquesta plataforma disposar d'un certificat vigent.



## 10. Planificació final

### 10.1. Desviació temporal

A l'inici d'aquest projecte es va dur a terme una primera planificació inicial on s'estimava el temps de durada de cada una de les fases i en què consistia cadascuna d'elles, tal i com s'explica en l'apartat 3.1.

Durant el transcurs de la fase d'anàlisi i especificació es va produir una desviació temporal de dues setmanes, concretament durant la fase de formació en les tecnologies que s'utilitzen per a desenvolupar aquest projecte. S'ha necessitat d'una formació més extensa que la plantejada inicialment, cosa que va fer endarrerir l'inici de les següents fases.

Tot i això, no ha afectat els objectius finals d'aquest projecte que han seguit sent els mateixos.

### 10.2. Imprevistos

#### 10.2.1. Suport pels dispositius Apple

Tot i que cada vegada Apple va adaptant-se a aquestes noves tecnologies, sempre ho fa amb cert retard degut a les seves polítiques pròpies d'empresa, en contra del seu competidor i principal impulsor de les PWA, Google.

Les últimes actualitzacions tant de iOS com de Safari han anat millorat el suport cap a les PWA, tot i que encara estan orfes de funcionalitats importants. Cal esperar i veure com avancen les actualitzacions i quines polítiques o limitacions apliquen.

#### **Suport del manifest en el navegador Safari**

Hi ha una sèrie de propietats del manifest que el navegador Safari encara no interpreta correctament. Entre elles destaquen:

- Les icones
- Modificació de la pantalla de càrrega de l'aplicació

Tot i això i no acceptar-les a través del manifest, es pot donar suport al navegador Safari a aquestes propietats a través d'etiquetes *link* a la capçalera del fitxer HTML.

## **Notificacions push iOS**

A data d'avui, el sistema operatiu iOS i el seu navegador Safari no dona suport a la Push API, la API que permet enviar notificacions push al navegador. Per tant, les notificacions push de l'aplicació no funcionen en aquests dispositius.

### 10.2.2. **Descàrrega d'arxius**

Un altre imprevist que s'ha trobat alhora de desenvolupar l'aplicació és que no es permet la descàrrega d'arxius al dispositiu mòbil ja que les aplicacions web no tenen accés al sistema d'arxius del dispositiu.

## 11. Sostenibilitat

### 11.1. Econòmica

S'han avaluat els costos, per una banda els costos humans i per altre banda els costos materials. Aquests costos s'han calculat tenint en compte que l'encarregat de desenvolupar el projecte és una persona.

Aquest projecte es podria desenvolupar més ràpidament de dues formes:

- Individualment però dedicant una jornada completa de 8 hores diàries. En aquest cas el cost seria exactament el mateix però es realitzaria el doble de ràpid.
- Amb un grup més gran de persones, tot i que hi hauria un impacte en el cost per cada persona involucrada, es podria disminuir molt més el temps que a la meitat.

Fent referència a fonts d'ingrés, aquest projecte es podria monetitzar de manera que els clients paguessin llicències en funció del número d'usuaris que necessitin.

### 11.2. Mediambiental

S'ha considerat la despesa d'energia com a principal impacte ambiental durant el cicle del projecte i el cost de manteniment energètic necessari tant per a la màquina que es fa servir pel desenvolupament com per el servidor on està allotjada l'aplicació.

També cal esmentar els residus provinents dels recursos de hardware usats un cop finalitzada la seva vida útil, els quals s'hauran de prendre mesures de reciclatge dels seus components per a minimitzar l'impacte.

Per últim, cal esmentar el cost de desplaçament. Degut a que aquests es faran a través d'una motocicleta, tenen un impacte tant durant la vida útil (combustible i emissions) com quan arribi el seu final. Per reduir al màxim l'impacte, es procura una conducció eficient, tant per reduir l'ús de combustible com les emissions i passant els controls i exàmens adients. Un cop finalitzada la vida útil, s'enviarà a un desballestador per a que es reciclin el màxim de components possibles.



## 12. Lleis i regulacions

Principalment s'han de tenir en compte els tres aspectes següents:

- La Llei orgànica de protecció de dades de caràcter personal, coneguda com a LOPD, que té com a objectiu protegir les dades sensibles de les persones físiques. S'han de seguir una sèrie de pautes i definir processos per assolir el compliment d'aquesta llei, com per exemple l'encriptació de les dades que es posseeixen, l'intercanvi d'informació personal o que una persona física pugui sol·licitar l'eliminació de les seves dades del sistema.
- La propietat intel·lectual i drets d'autor.
- La Ley de Servicios de la Sociedad de la información y del comercio electrónico (LSSI). S'ha d'informar als usuaris de manera permanent, fàcil, directa i gratuïta informació de contacte i referent al prestador del servei, així com informar de la política de cookies.



## 13. Conclusions

L'objectiu d'aquest projecte era desenvolupar una plataforma per ajudar a gestionar i centralitzar la comunicació en una organització. Per assolir aquest objectiu s'ha creat una aplicació multiplataforma que permet als usuaris finals mantenir una via de contacte privada i controlada.

Tret dels imprevistos trobats durant el desenvolupament d'aquesta plataforma, el producte final compleix els requeriments inicials a excepció dels dispositius iOS que s'haurà d'estar atent a com procedeix Apple amb aquestes tecnologies.

Una de les dificultats principals ha estat l'entorn de desenvolupament tant de l'aplicació web com de la API. Tot i haver treballat amb Javascript prèviament, és un llenguatge que ha evolucionat molt els darrers anys i ja no només recau en la manipulació del DOM. En cap cas havia rebut cap formació universitària ni de React ni de Node.js.

Tot això juntament amb què tot i ser una aplicació web ha d'estar pensada per treballar com una aplicació mòbil nativa fa que s'hagi hagut de tenir especial cura de no implementar llibreries innecessàries que poguessin afectar el rendiment de l'aplicació.

Finalment però, s'ha aconseguit un producte funcional i que podria ser implementat en grups per valorar si satisfà les necessitats internes.

### 13.1. Conclusions personals

Amb motiu de tancament del projecte cal fer una valoració personal de tot el treball realitzat. Ara fa sis mesos vaig decidir capficar-me en aquest projecte ja que tinc molt d'interès en les tecnologies web i considerava molt atractiu un projecte d'aquest tipus per poder investigar i treballar amb tecnologies innovadores. Han estat sis mesos de molt treball des del primer dia, tant de treball d'anàlisi com de desenvolupament.

M'agradaria recalcar que les dificultats en quan a l'entorn de desenvolupament esmentades anteriorment, sempre les he vist com una oportunitat per formar-me en dues tecnologies molt demandades a dia d'avui com poden ser React i Node.js, sense deixar de banda les PWA que ja són present.

Una de les valoracions més positives que m'enduc és el fet d'haver estat capaç de superar el repte plantejat inicialment. Ser capaç de trencar un problema molt gran en peces petites que van encaixant i poc a poc prenen forma és una gran satisfacció.



## 14. Bibliografia

1. Academind [En línia] [Data: 15 / Gener / 2019]

<https://www.academind.com/learn/flutter/react-native-vs-flutter-vs-ionic-vs-nativescript-vs-pwa/>

2. React [En línia] [Data: 25 / Març / 2019]

<https://reactjs.org/>

3. Redux [En línia] [Data: 25 / Març / 2019]

<https://redux.js.org/>

4. Material UI [En línia] [Data: 26 / Març / 2019]

<https://material-ui.com/>

5. Express js [En línia] [Data: 3 / Abril / 2019]

<https://expressjs.com/>

6. JSON Web Tokens [En línia] [Data: 6 / Abril / 2019]

<https://jwt.io/>

7. Wikipedia [En línia] [Data: 6 / Abril / 2019]

[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

8. HTTP CORS [En línia] [Data: 12 / Abril / 2019]

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

9. Google Developers [En línia] [Data: 25 / Abril / 2019]

<https://developers.google.com/web/fundamentals/>

10. Service Workers [En línia] [Data: 25 / Abril / 2019]

<https://developers.google.com/web/fundamentals/primers/service-workers/>

11. Cloud Messaging [En línia] [Data: 25 / Abril / 2019]

<https://firebase.google.com/docs/cloud-messaging/>

12. Nginx [En línia] [Data: 27 / Maig / 2019]

<https://nginx.org/en/docs/>

13. Certbot [En línia] [Data: 27 / Maig / 2019]

<https://certbot.eff.org/about/>

## Annex I: Fitxes de casos d'ús

**Cas d'ús:** CU-001 Login

**Actors:** Membre, Gestor, Administrador, Client, Servidor

**Precondicions:** -

**Condicció d'èxit:** L'actor queda logejat a l'aplicació.

**Condicció de fracàs:** L'actor no es pot logejar correctament.

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Client** Mostra formulari de login.
3. **Usuari** Omple el formulari i prem el botó "Accedir".
4. **Servidor** Valida les dades introduïdes.
5. **Usuari** Accedeix a l'aplicació.

**Flux alternatiu:**

- 4.1.1 **Servidor** Les dades introduïdes no són correctes.
- 4.1.2 **Client** Mostra un missatge d'error i acaba el cas d'ús.
- 4.2 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.

**Cas d'ús:** CU-002 Consultar historial de comunicats del grup

**Actors:** Membre, Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari veu el llistat de comunicats referent a un grup.

**Condicció de fracàs:** -

**Dependències:** RN-004

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Selecciona el grup objectiu del menú superior.
3. **Usuari** Selecciona la icona referent a comunicats.
4. **Servidor** Retorna els comunicats del grup ordenats cronològicament.
5. **Client** Mostra el llistat de comunicats.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostrarà el llistat guardats en caché.

**Cas d'ús:** CU-003 Enviar documentació

**Actors:** Membre, Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari ha enviat un document als components del grup.

**Condicció de fracàs:** El document no es pot enviar.

**Dependències:** RN-002

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Selecciona el grup objectiu del menú superior.
3. **Usuari** Selecciona la icona referent a documentació.
4. **Client** Mostra la pantalla documentació.
5. **Usuari** Prem el botó d'enviar documentació.
6. **Client** Obre la finestra de selecció d'arxiu.
7. **Usuari** Selecciona l'arxiu a enviar i prem "Enviar".
8. **Servidor** Valida que l'arxiu enviat sigui vàlid.
9. **Client** Mostra missatge d'èxit i mostra el llistat de documents.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1.1 **Servidor** L'arxiu enviat no és vàlid.
- 8.1.2 **Client** Mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-004** Respondre votació

**Actors:** Membre, Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condió d'èxit:** El vot de l'usuari queda registrat.

**Condió de fracàs:** L'usuari no pot efectuar el vot.

**Dependències:** RN-006, RN-007

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Selecciona el grup objectiu del menú superior.
3. **Usuari** Selecciona la icona referent a votacions.
4. **Client** Mostra les votacions del grup seleccionat.
5. **Usuari** Selecciona la votació i prem "Votar".
6. **Servidor** Comprova que l'usuari té dret a vot.
7. **Client** Mostra el formulari de votació.
8. **Usuari** Omple el formulari i prem "Enviar".
9. **Servidor** Registra el vot.
10. **Client** Mostra missatge d'èxit i mostra el resultat de la votació.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 6.1 **Client** Si l'usuari no té dret a vot mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús:** CU-005 Crear grup

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari crear un nou grup.

**Condicció de fracàs:** No es pot crear un nou grup.

**Dependències:** RN-003, RN-008

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Grups→Nou grup.
4. **Client** Mostra formulari de creació de grup.
5. **Usuari** Omple el formulari i selecciona usuaris o grups que compondran el nou grup.
6. **Usuari** Prem "Crear".
7. **Servidor** Valida que les dades enviades siguin correctes.
8. **Client** Mostra missatge d'èxit i mostra llistat de grups.

**Flux alternatiu:**

- 6.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 7.1 **Client** Si les dades enviades no són vàlides mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús:** CU-006 Desactivar grup

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** El grup queda desactivat.

**Condicció de fracàs:** El grup no es pot desactivar.

**Dependències:** RN-009

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Grups.
4. **Client** Mostra llistat de grups.
5. **Usuari** Prem interruptor de desactivar.
8. **Servidor** Valida que l'usuari té permisos per realitzar l'acció.
9. **Client** Mostra missatge d'èxit i es mostra el llistat de grups.

**Flux alternatiu:**

- 4.1 **Client** Si no hi han grups mostra missatge informatiu i acaba el cas d'ús.
- 7.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1 **Client** Si l'usuari no té permisos mostra missatge d'error i acaba el cas d'ús.



**Cas d'ús:** CU-007 Afegir usuaris o grups a grup

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari o el grup s'ha afegit al grup.

**Condicció de fracàs:** No es pot afegir un usuari o grup al grup.

**Dependències:** RN-005

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Grups.
4. **Client** Mostra llistat de grups.
5. **Usuari** Selecciona el grup objectiu i prem Gestionar al grup.
6. **Client** Mostra usuaris i grups potencials.
7. **Usuari** Prem els interruptors dels usuaris o grups que vol afegir.
8. **Servidor** Valida que els usuaris seleccionats no pertanyin ja al grup.
9. **Client** Mostra missatge d'èxit i mostra el llistat de grups.

**Flux alternatiu:**

- 4.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 4.2 **Client** Si no hi han grups mostra missatge informatiu i acaba el cas d'ús.
- 8.1 **Client** Si no són vàlids mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-008** Eliminar usuari o grup d'un grup

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condió d'èxit:** L'usuari o el grup es desvincula del grup objectiu.

**Condió de fracàs:** L'usuari o el grup no es pot desvincular del grup objectiu.

**Dependències:** -

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Grups.
4. **Client** Mostra llistat de grups.
5. **Usuari** Selecciona el grup objectiu i prem Gestionar el grup.
6. **Client** Mostra els usuaris i grups potencials.
7. **Usuari** Prem els interruptors dels usuaris o grups que vol eliminar.
8. **Servidor** Valida que els usuaris seleccionats pertanyin al grup.
9. **Client** Mostra missatge d'èxit i mostra el llistat de grups.

**Flux alternatiu:**

- 4.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 4.2 **Client** Si no hi han grups mostra missatge informatiu i acaba el cas d'ús.
- 8.1 **Client** Si no són vàlids mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús:** CU-009 Crear votació

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari ha enviat un document als components del grup.

**Condicció de fracàs:** El document no es pot enviar.

**Dependències:** -

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Selecciona el grup objectiu del menú superior.
3. **Usuari** Selecciona la icona referent a votacions.
4. **Client** Mostra la pantalla de votacions.
5. **Usuari** Prem el botó de crear votació
6. **Client** Mostra el formulari de creació d'una votació.
7. **Usuari** Omple el formulari, definint les preguntes i les respostes per cada una d'elles.
8. **Usuari** Prem "Crear".
9. **Servidor** Valida la informació enviada.
10. **Client** Mostra missatge d'èxit i mostra el llistat de votacions.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 9.1.1 **Servidor** La informació enviada no és vàlida.
- 9.1.2 **Client** Mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-010** Enviar comunicat

**Actors:** Gestor, Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condió d'èxit:** S'envia un comunicat als usuaris del grup.

**Condió de fracàs:** El comunicat no es pot enviar.

**Dependències:** -

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Selecciona el grup objectiu del menú superior.
3. **Usuari** Selecciona la icona referent a comunicats.
4. **Client** Mostra la pantalla de comunicats.
5. **Usuari** Prem el botó de nou comunicat.
6. **Client** Mostra el formulari de nou comunicat.
7. **Usuari** Omple el formulari i prem "Enviar".
8. **Servidor** Valida que les dades siguin vàlides.
9. **Servidor** Envia notificació push.
10. **Client** Mostra missatge d'èxit i mostra el llistat de comunicats.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1.1 **Servidor** Les dades enviades no són vàlides.
- 8.1.2 **Client** Mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús:** CU-011 Crear usuari

**Actors:** Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** Es crea un nou usuari.

**Condicció de fracàs:** No es pot crea un nou usuari.

**Dependències:** RN-011

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Usuaris.
4. **Client** Mostra llistat d'usuaris.
5. **Usuari** Prem el botó de nou usuari.
6. **Client** Mostra el formulari de creació d'usuari.
7. **Usuari** Omple el formulari amb la informació corresponent i prem el botó "Crear".
8. **Servidor** Valida que les dades introduïdes siguin correctes.
9. **Client** Mostra missatge d'èxit i mostra llistat de grups.

**Flux alternatiu:**

- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1 **Client** Si les dades enviades no són vàlides mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-012** Editar usuari

**Actors:** Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condió d'èxit:** Les dades de l'usuari queden actualitzades.

**Condió de fracàs:** Les dades de l'usuari no es poden actualitzar.

**Dependències:** RN-011

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Usuaris.
4. **Client** Mostra llistat d'usuaris.
5. **Usuari** Selecciona l'usuari i prem el botó d'editar usuari.
6. **Client** Mostra el formulari d'edició amb les dades actuals de l'usuari.
7. **Usuari** Modifica les dades i prem "Guardar".
8. **Servidor** Valida que les dades introduïdes siguin correctes.
9. **Client** Mostra missatge d'èxit i mostra llistat de grups.

**Flux alternatiu:**

- 4.1 **Client** Si no hi han usuaris mostra missatge informatiu i acaba el cas d'ús.
- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1 **Client** Si les dades enviades no són vàlides mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-013** Eliminar usuari

**Actors:** Administrador, Client, Servidor

**Precondicions:** CU-001 Login

**Condicció d'èxit:** L'usuari queda desactivat.

**Condicció de fracàs:** L'usuari no es pot desactivar.

**Dependències:** RN-009, RN-010

**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona Usuaris.
4. **Client** Mostra llistat d'usuaris.
5. **Usuari** Selecciona l'usuari i prem l'interruptor.
8. **Servidor** Desactiva l'usuari.
9. **Client** Mostra missatge d'èxit i mostra llistat d'usuaris.

**Flux alternatiu:**

- 4.1 **Client** Si no hi han usuaris mostra missatge informatiu i acaba el cas d'ús.
- 5.1 **Client** Si no hi ha connexió a Internet mostra un missatge d'error i acaba el cas d'ús.
- 8.1 **Client** Si les dades enviades no són vàlides mostra missatge d'error i acaba el cas d'ús.

**Cas d'ús: CU-014 Logout****Actors:** Membre, Gestor, Administrador, Client**Precondicions:** CU-001 Login**Condió d'èxit:** L'usuari es desconnecta de l'aplicació.**Condió de fracàs:** -**Dependències:** -**Flux principal:**

1. **Usuari** Entra a l'aplicació.
2. **Usuari** Desplega el menú lateral.
3. **Usuari** Selecciona "Tancar sessió".
4. **Client** Mostra pantalla de login.






## Annex II: Wireframes

usuari \_\_\_\_\_

contrassenya \_\_\_\_\_

**ACCEDIR**

☰ Grup 1 v

 MISSATGES	 ARXIU	 VOTACIONS
--	--	--

Títol 1  
~~~~~ >

Títol 2  
~~~~~ >

Títol 3  
~~~~~ >  
~~~~~ (+)

|               |  |
|---------------|--|
| USUARI <      |  |
| TANCAR SESSIÓ |  |
| Imici         |  |
| Grups         |  |
| Usuaris       |  |

☰

◀ TORNAR

Títol del missatge

Cor del missatge ~~~~~

~~~~~

~~~~~

~~~~~

☰ Grup 1 v

◀ TORNAR

Títol del missatge

---

Cos del missatge

ENVIAR    RESET

☰ Grup 1 v

 MISSATGES	 ARXIVS	 VOTACIONS
---------------	------------	---------------

Document 1 ↓

Document 2 ↓

Document 3 ↓

+

☰ Grup 1 v

◀ TORNAR

Títol de la votació

---

Descripció

AFEGIR PREGUNTA

Pregunta 1

---

AFEGIR RESPOSTA

Resposta 1

Resposta 2

---

Pregunta 2

AFEGIR RESPOSTA

ENVIAR    RESET

☰ Grup 1 v

 MISSATGES	 ARXIVS	 VOTACIONS
---------------	------------	---------------

Votació 1

VEURE RESULTATS    VOTAR

---

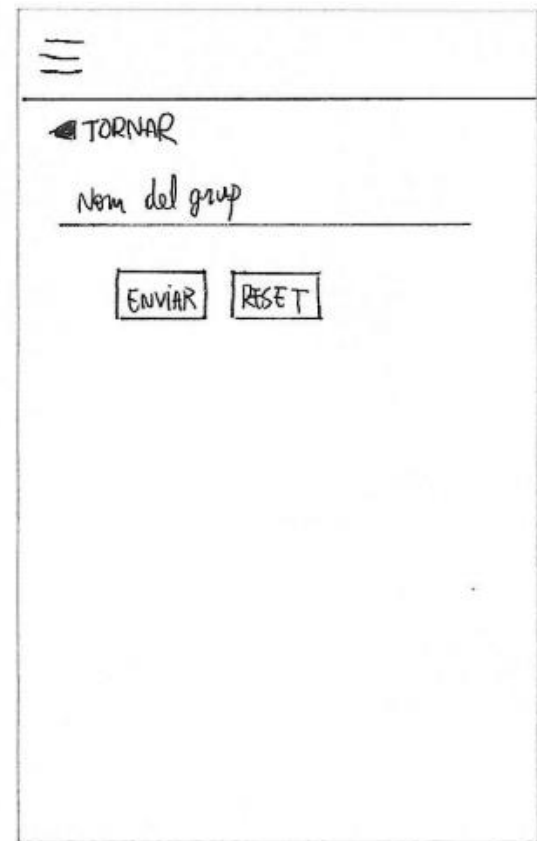
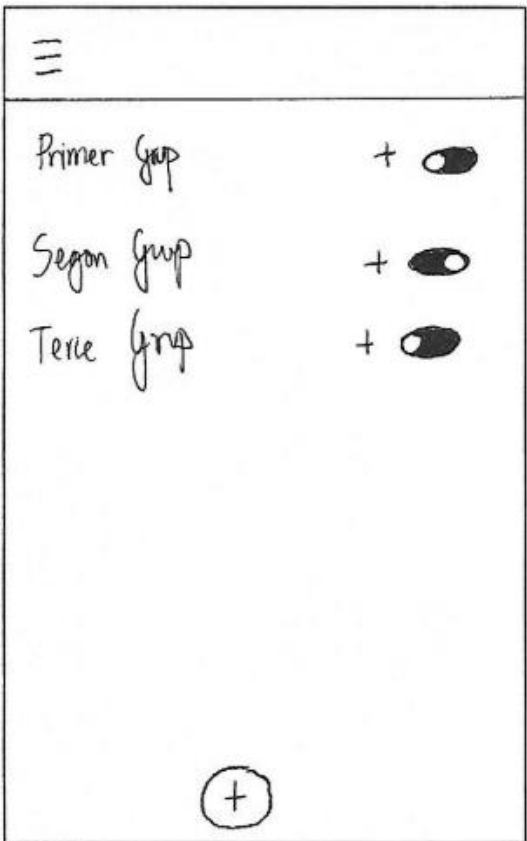
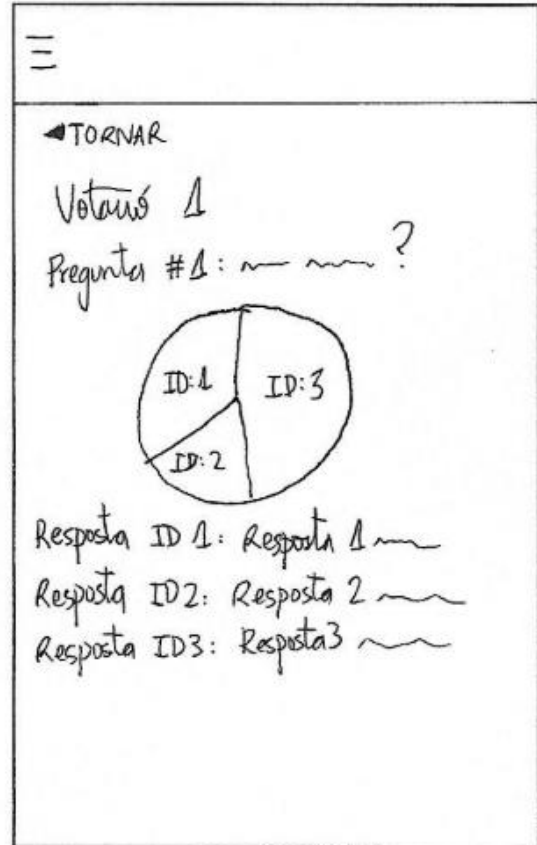
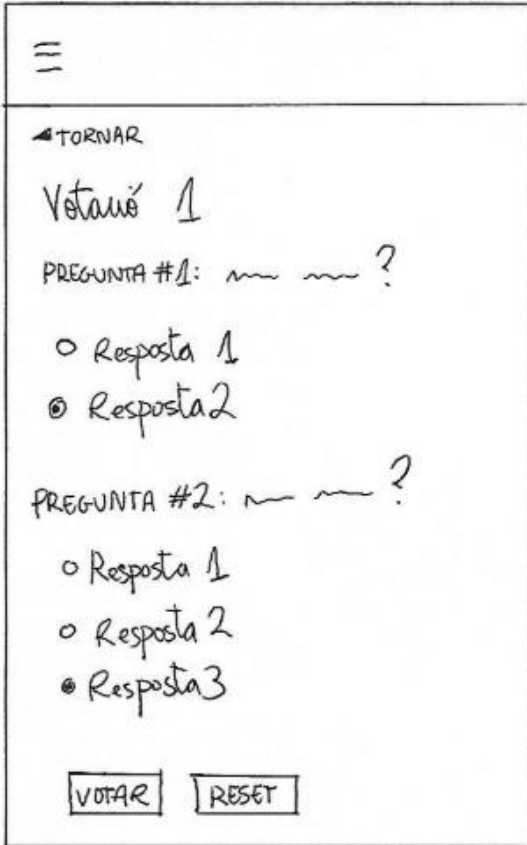
Votació 2

VEURE RESULTATS

---

Votació 3

VEURE RESULTATS    VOTAR



☰

◀ TORNAR

USUARIS      GRUPS

---

usuari 1     

usuari 2     

usuari 3     

☰

usuari 1      EDITAR

usuari 2      EDITAR

usuari 3      EDITAR

(+)

☰

◀ TORNAR

nom usuari

contrassenya

confirmar contrassenya

rol      v

ENVIAR    RESET

## Annex III: Contingut del CD-ROM

- Documentació del projecte (memòria)
- Script de creació de base de dades
- Codi font de l'aplicació client
- Codi font de l'API