

**Josep Calero Santo**

**APLICACIÓ DE VOLUNTARIAT DE TORTOSA**

**TREBALL DE FI DE GRAU**

**Dirigit per Pere Millán Marco**

**Grau d'Enginyeria Informàtica**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2021**



**Resum.**

Aquesta memòria tracta el procés de creació d'una aplicació des del punt de vista d'un enginyer de software. El projecte forma part del programa Aprenentatge Servei (APS) de la URV, que permet realitzar un Treball de Fi de Grau amb implicació social sobre el territori.

Es tracta d'una aplicació per a Android, iOS i Web que permet veure l'oferta d'activitats de voluntariat i les entitats que s'hi dediquen a la ciutat de Tortosa. Es pot veure la informació de les activitats disponibles, així com quina entitat les organitza, dades de contacte, dates, xarxes socials... de cara a que el voluntari es posi en contacte amb les entitats. A més de ser accessible per a l'usuari que vulgui ser voluntari. També inclou un accés restringit per a que un administrador pugui modificar les dades que es mostren a l'aplicació.

La metodologia de desenvolupament ha seguit els diferents passos de l'enginyeria del software: anàlisi dels requisits, disseny, implementació i avaluació. S'ha utilitzat el kit de desenvolupament Flutter, que permet crear aplicacions multiplataforma. Per emmagatzemar dades, s'ha utilitzat la plataforma Firebase, que té molta sinergia amb Flutter, facilitant el procés de desenvolupament i manteniment.

El projecte s'ha realitzat amb l'objectiu de crear un impacte sobre una comunitat, facilitant l'accés a la informació, possiblement la tasca més important d'un enginyer de software.

**Resumen.**

Esta memoria trata el proceso de creación de una aplicación desde el punto de vista de un ingeniero de software. El proyecto forma parte del programa Aprendizaje Servicio (APS) de la URV, que permite realizar un Trabajo de Fin de Grado con implicación social sobre el territorio.

Se trata de una aplicación para Android, iOS y Web que permite ver la oferta de actividades de voluntariado y las entidades que se dedican en la ciudad de Tortosa. Se puede ver la información de las actividades disponibles, así como qué entidad las organiza, datos de contacto, fechas, redes sociales... de cara a que el voluntario se ponga en contacto con las entidades. Además de ser accesible para el usuario que quiera ser voluntario, también incluye un acceso restringido para que un administrador pueda modificar los datos que se muestran en la aplicación.

La metodología de desarrollo ha seguido los diferentes pasos de la ingeniería de software: análisis de los requisitos, diseño, implementación y evaluación. Se ha utilizado el kit de desarrollo Flutter, que permite crear aplicaciones multiplataforma. Para almacenar datos, se ha utilizado la plataforma Firebase, que tiene mucha sinergia con Flutter, facilitando el proceso de desarrollo y mantenimiento.

El proyecto se ha realizado con el objetivo de crear un impacto sobre una comunidad, facilitando el acceso a la información, posiblemente la labor más importante de un ingeniero de software.

**Abstract.**

This memory deals with the process of creating an application from the point of view of a software engineer. The project is part of the Learning-Service Program (APS) from the URV, which allow students to make a bachelor thesis with social involvement over the territory.

It is an Android, iOS, and Web application that allow users to view volunteer activities and their entities in the city of Tortosa. Users can see information about the available activities as well as which entity organizes them, contact data, dates, social media... for the volunteer to contact entities. In addition to being accessible to the user who wants to be a volunteer, it also includes restricted access for an administrator to modify the application-data shown.

The development methodology has followed the different steps of software engineering: analysis of requirements, design, implementation and evaluation. The Flutter development kit has been used to create a cross-platform application. Also, the Firebase platform has been used to store data, which has much synergy with Flutter, hence facilitating the development and maintenance processes.

The project has been performed with the aim of creating an impact on a community, facilitating access to information, possibly the most important work of a software engineer.

# Índex

<b>1</b>	<b>INTRODUCCIÓ</b>	<b>11</b>
<b>2</b>	<b>DESCRIPCIÓ</b>	<b>12</b>
2.1	APLICACIONS SIMILARS	13
2.1.1	<i>European Solidarity Corps</i>	13
2.1.2	<i>Moviliza-T</i>	13
<b>3</b>	<b>REQUISITS</b>	<b>14</b>
3.1	GUIONS	14
3.1.1	<i>Guió de l'usuari</i>	14
3.1.2	<i>Guió de l'administrador</i>	14
3.1.3	<i>Altres funcions</i>	14
3.2	REQUISITS FUNCIONALS	15
3.2.1	<i>Diagrama de cassos d'ús versió 1</i>	15
3.2.2	<i>Especificació textual dels cassos d'ús: Versió 1 de l'aplicació</i>	15
3.2.3	<i>Diagrama de cassos d'ús versió 2</i>	19
3.2.4	<i>Especificació textual dels cassos d'ús: Versió 2 de l'aplicació</i>	19
3.2.5	<i>Especificació textual dels cassos d'ús: Versió 3 de l'aplicació</i>	21
3.3	REQUISITS NO FUNCIONALS	23
3.3.1	<i>Requisits de memòria</i>	23
3.3.2	<i>Requisits de rendiment</i>	24
3.3.3	<i>Requisits d'usabilitat</i>	24
3.3.4	<i>Requisits de fiabilitat</i>	24
3.3.5	<i>Requisits de portabilitat</i>	25
<b>4</b>	<b>ANÀLISI DELS REQUISITS</b>	<b>26</b>
4.1	DIAGRAMA DE CLASSES	26
4.2	DIAGRAMES DE SEQÜENCIES DELS CASSOS D'ÚS	26
<b>5</b>	<b>DISSENY</b>	<b>37</b>
5.1	TECNOLOGIES CONSIDERADES	37
5.1.1	<i>Kotlin per al desenvolupament Android</i>	37
5.1.2	<i>Tecnologies multiplataforma</i>	37
5.2	TECNOLOGIES A UTILITZAR	39
5.2.1	<i>Flutter</i>	39
5.2.2	<i>Arquitectura: BLoC</i>	39
5.2.3	<i>Serveis de Firebase</i>	40
5.2.4	<i>Control de versions: GitHub</i>	42
5.3	PROTOTIPS DE LA INTERFÍCIE GRÀFICA: VERSIÓ 1	42
5.4	PROTOTIPS DE LA INTERFÍCIE GRÀFICA: VERSIÓ 2	47
<b>6</b>	<b>DESENVOLUPAMENT</b>	<b>51</b>
6.1	ITERACIÓ 1: EXPERIMENTACIÓ	51
6.2	ITERACIÓ 2: ESTRUCTURA I FUNCIONALITATS BASE DE L'APLICACIÓ	52
6.3	ITERACIÓ 3: AMPLIACIONS DE L'ESTRUCTURA	54
6.4	ITERACIÓ 4: MILLORES VISUALS	59
6.5	ITERACIÓ 5: IMPLEMENTACIÓ PER A IOS	62
6.6	ITERACIÓ 6: REORGANITZACIÓ DE LA INTERFÍCIE D'USUARI	62
6.7	ITERACIÓ 7: REORGANITZACIÓ DEL CODI	66
6.8	ITERACIÓ 8: TORNADA AL DESENVOLUPAMENT A IOS	67
<b>7</b>	<b>IMPLEMENTACIÓ</b>	<b>69</b>
7.1	ESPECIFICACIONS ADDICIONALS	69
7.1.1	<i>Diagrama de classes</i>	69
7.1.2	<i>Requisits no implementats</i>	70

7.1.3	<i>Requisits modificats</i>	70
7.1.4	<i>Requisits afegits</i>	70
7.2	ESTRUCTURA DEL PROJECTE: DIRECTORIS I FITXERS	71
7.2.1	<i>Fitxer main.dart</i>	71
7.2.2	<i>Directori data</i>	71
7.2.3	<i>Directori services</i>	72
7.2.4	<i>Directori screens</i>	72
7.2.5	<i>Directori utils</i>	74
7.2.6	<i>Directori adminspecific</i>	74
7.3	ALGORITMES ESPECÍFICS DE LES PARTS MÉS COMPLEXES	75
7.3.1	<i>Patró BLoC implementat</i>	75
7.3.2	<i>Cerca amb actualització dinàmica</i>	77
7.3.3	<i>Pujar imatges a Firebase Storage</i>	78
<b>8</b>	<b>AVALUACIÓ</b>	<b>82</b>
8.1	TESTS FUNCIONALS	82
8.1.1	<i>Seqüència de tests per a usuaris</i>	82
8.1.2	<i>Seqüència de tests per a administradors</i>	84
8.2	TESTS NO FUNCIONALS	85
8.2.1	<i>Test de memòria</i>	85
8.2.2	<i>Test de rendiment</i>	85
8.2.3	<i>Test d'usabilitat</i>	87
8.2.4	<i>Test de fiabilitat</i>	87
8.2.5	<i>Test de portabilitat</i>	87
<b>9</b>	<b>INSTAL·LACIÓ I MANTENIMENT</b>	<b>88</b>
9.1	CONFIGURACIÓ DE L'ENTORN DE DESENVOLUPAMENT	88
9.2	INSTAL·LACIÓ	88
9.2.1	<i>Instal·lació de Firebase</i>	88
9.2.2	<i>Instal·lació Web</i>	88
9.2.3	<i>Instal·lació Android</i>	89
9.2.4	<i>iOS</i>	89
9.3	MANTENIMENT	89
9.3.1	<i>Manteniment de Firebase</i>	89
9.3.2	<i>Modificacions en el codi</i>	90
<b>10</b>	<b>CONCLUSIONS</b>	<b>91</b>
10.1	POSSIBLES AMPLIACIONS FUTURES	92
<b>11</b>	<b>REFERÈNCIES</b>	<b>93</b>
<b>12</b>	<b>ANNEXES</b>	<b>98</b>

# Índex de taules

TAULA 1. REQUISIT 00. IDENTIFICACIÓ.....	15
TAULA 2. REQUISIT 01. LLISTA D'ACTIVITATS .....	16
TAULA 3. REQUISIT 02. CERCAR .....	16
TAULA 4. REQUISIT 03. DETALLS D'ACTIVITAT .....	16
TAULA 5. REQUISIT 04. FILTRAR.....	17
TAULA 6. REQUISIT 05. LLISTAT D'ENTITATS .....	17
TAULA 7. REQUISIT 06. CONSULTAR INFORMACIÓ SOBRE L'APP .....	17
TAULA 8. REQUISIT 07. CONSULTAR INFORMACIÓ SOBRE L'APP .....	17
TAULA 9. REQUISIT 08. AFEGIR ACTIVITAT .....	18
TAULA 10. REQUISIT 09. ESBORRAR ACTIVITAT .....	18
TAULA 11. REQUISIT 10. ACTIVITATS PREFERIDES .....	20
TAULA 12. REQUISIT 11. OCULTAR ACTIVITATS .....	20
TAULA 13. REQUISIT 12. EDITAR ACTIVITATS.....	20
TAULA 14. REQUISIT 13. DESTACAR ACTIVITATS .....	20
TAULA 15. REQUISIT 14. LLISTA D'ACTIVITATS DINS D'UNA ENTITAT .....	20
TAULA 16. REQUISIT 15. FILTRES AVANÇATS.....	21
TAULA 17. REQUISIT 16. ACTIVITAT AMB IMATGE .....	21
TAULA 18. REQUISIT 17. ENTITAT AMB IMATGE.....	21
TAULA 19. REQUISIT 18. ORDENACIÓ .....	21
TAULA 20. REQUISIT 19. NOTIFICACIONS PUSH .....	22
TAULA 21. REQUISIT 20. TUTORIAL INICIAL .....	22
TAULA 22. REQUISIT 21. REGISTRE I IDENTIFICACIÓ D'USUARIS.....	22
TAULA 23. REQUISIT 22. LOCALITZACIÓ D'ACTIVITATS EN UN MAPA .....	22
TAULA 24. REQUISIT 23. MÚLTIPLES IDIOMES.....	23
TAULA 25. REQUISIT 24. ENTITATS FAVORITES.....	23
TAULA 26. REQUISIT 25. IMATGES AVANÇADES .....	23
TAULA 27. REQUISIT 03. DETALLS D'ACTIVITAT (ACTUALITZAT).....	70
TAULA 28. REQUISIT 26. DETALLS D'ENTITAT .....	70
TAULA 29. REQUISIT 27. PÀGINA PRINCIPAL.....	71
TAULA 30. REQUISIT 28. FILTRE SEGONS PRESENCIALITAT .....	71

# Índex de figures

FIGURA 1. ESQUEMA AGILE .....	12
FIGURA 2. DIAGRAMA DE CASSOS D'ÚS VERSIÓ 1 .....	15
FIGURA 3. DIAGRAMA DE CASSOS D'ÚS VERSIÓ 2 .....	19
FIGURA 4. PROTOTIP DE FILTRES AVANÇATS .....	26
FIGURA 5. SEQÜENCIA DEL CAS D'ÚS 00: IDENTIFICACIÓ .....	27
FIGURA 6. SEQÜENCIA DEL CAS D'ÚS 01: LLISTA D'ACTIVITATS .....	28
FIGURA 7. SEQÜENCIA DEL CAS D'ÚS 02: CERCAR .....	28
FIGURA 8. SEQÜENCIA DEL CAS D'ÚS 03: DETALLS D'ACTIVITAT .....	29
FIGURA 9. SEQÜENCIA DEL CAS D'ÚS 04: FILTRAR PER TIPUS .....	29
FIGURA 10. SEQÜENCIA DEL CAS D'ÚS 05: LLISTAT D'ENTITATS .....	30
FIGURA 11. SEQÜENCIA DEL CAS D'ÚS 06: CONSULTAR INFORMACIÓ SOBRE L'APP .....	30
FIGURA 12. SEQÜENCIA DEL CAS D'ÚS 07: FORMULARI PER FEEDBACK .....	31
FIGURA 13. SEQÜENCIA DEL CAS D'ÚS 08: AFEGIR ACTIVITAT .....	32
FIGURA 14. SEQÜENCIA DEL CAS D'ÚS 09: ESBORRAR ACTIVITAT .....	32
FIGURA 15. SEQÜENCIA DEL CAS D'ÚS 10: ACTIVITATS PREFERIDES .....	32
FIGURA 16. SEQÜENCIA DEL CAS D'ÚS 11: OCULTAR ACTIVITATS .....	33
FIGURA 17. SEQÜENCIA DEL CAS D'ÚS 12: EDITAR ACTIVITATS .....	33
FIGURA 18. SEQÜENCIA DEL CAS D'ÚS 13: DESTACAR ACTIVITATS .....	34
FIGURA 19. SEQÜENCIA DEL CAS D'ÚS 14: LLISTA D'ACTIVITATS DINS DE L'ENTITAT .....	34
FIGURA 20. SEQÜENCIA DEL CAS D'ÚS 15: FILTRES AVANÇATS .....	35
FIGURA 21. SEQÜENCIA DEL CAS D'ÚS 16: ACTIVITAT AMB IMATGE .....	35
FIGURA 22. SEQÜENCIA DEL CAS D'ÚS 17: ENTITAT AMB IMATGE .....	36
FIGURA 23. DIAGRAMA DE L'ARQUITECTURA BLOC .....	40
FIGURA 24. ELECCIÓ ENTRE CLOUD FIRESTORE I REALTIME DATABASE .....	41
FIGURA 25. PROTOTIP LLISTA D'ACTIVITATS .....	43
FIGURA 26. PROTOTIP CERCADOR I FILTRE .....	43
FIGURA 27. PROTOTIP DETALLS D'ACTIVITAT .....	44
FIGURA 28. PROTOTIP MENÚ .....	44
FIGURA 29. PROTOTIP IDENTIFICACIÓ ADMINISTRADOR .....	45
FIGURA 30. PROTOTIP LLISTA D'ACTIVITATS D'ADMINISTRADOR .....	45
FIGURA 31. PROTOTIP CREAR ACTIVITAT .....	46
FIGURA 32. PROTOTIP LLISTA D'ENTITATS .....	47
FIGURA 33. PROTOTIP ACTIVITATS FAVORITES .....	48
FIGURA 34. PROTOTIP FILTRES AVANÇATS .....	48
FIGURA 35. PROTOTIP AMB MÉS OPCIONS PER A L'ADMINISTRADOR .....	49
FIGURA 36. PROTOTIP D'ACTIVITATS DESTACADES .....	49
FIGURA 37. PROTOTIP LLISTA D'ACTIVITATS D'UNA ENTITAT .....	50
FIGURA 38. PROTOTIP INTERACTIU CERCA I MENÚ .....	53
FIGURA 39. PROTOTIP INTERACTIU WEB .....	54
FIGURA 40. PROTOTIP INTERACTIU DETALLS D'ACTIVITAT .....	55
FIGURA 41. FULL DE RUTA FLUTTER .....	56
FIGURA 42. PROTOTIP INTERACTIU D'INICIAR SESSIÓ .....	56
FIGURA 43. PROTOTIP INTERACTIU DE CERCA FUNCIONAL .....	57
FIGURA 44. PROTOTIP INTERACTIU LLISTA D'ENTITATS I FORMULARI .....	58
FIGURA 45. PROTOTIP INTERACTIU VISIBILITAT D'ACTIVITATS .....	58
FIGURA 46. PROTOTIP INTERACTIU DEL FORMULARI D'ACTIVITATS .....	59
FIGURA 47. PROTOTIP INTERACTIU FILTRE DE TIPUS .....	60
FIGURA 48. PROTOTIP INTERACTIU WEB .....	60
FIGURA 49. PROTOTIP INTERACTIU DETALLS D'ACTIVITAT AMB IMATGE .....	61
FIGURA 50. PROTOTIP INTERACTIU AMB ERRORS DE LA LLISTA .....	61
FIGURA 51. ACTUALITZACIÓ DE LA PANTALLA PRINCIPAL DE WALLAPOP .....	63
FIGURA 52. PANTALLES PRINCIPALS DE REVOLUT I YAHOO FINANCE .....	63
FIGURA 53. PROTOTIP INTERACTIU AMB MENÚ INFERIOR .....	64
FIGURA 54. PROTOTIP INTERACTIU AMB CERCA DINÀMICA .....	65
FIGURA 55. PROTOTIP INTERACTIU AMB FILTRE DE PRESENCIALITAT I TIPUS .....	65
FIGURA 56. PROTOTIP INTERACTIU AMB IMATGE I VÍDEO D'ENTITAT .....	66

FIGURA 57. DIAGRAMA DE CLASSES ACTUALITZAT.....	69
FIGURA 58. ESTRUCTURA DE LES DADES A FIREBASE FIRESTORE.....	77
FIGURA 59. PROTOTIP INTERACTIU PUJANT IMATGES I PERMISSOS .....	79
FIGURA 60. PROTOTIP INTERACTIU PUJANT IMATGES A WEB.....	80
FIGURA 61. RENDIMENT GRÀFIC .....	86
FIGURA 62. RENDIMENT DE MEMÒRIA.....	86
FIGURA 63. RENDIMENT GRÀFIC NEXUS 5 .....	87

## Índex de codis

CODI 1. MAIN.....	53
CODI 2. OPERACIÓ D'ESBORRAT AMB FIREBASE .....	55
CODI 3. STRINGS .....	67
CODI 4. ATRIBUTS D'ACTIVITY .....	71
CODI 5. ATRIBUTS D'ENTITAT .....	72
CODI 6. FITXER ADMIN.DART .....	72
CODI 7. HOMEPAGE SIMPLIFICAT .....	75
CODI 8. DESCARREGA D'ACTIVITATS DES DE FIREBASE .....	76
CODI 9. HOMELIST SIMPLIFICAT .....	76
CODI 10. CERCA DINÀMICA.....	78
CODI 11. PUJADA D'IMATGES A MÒBIL I WEB.....	81

## 1 Introducció

Aquest projecte forma part del Programa Aprenentatge Servei (APS<sup>1</sup>), una proposta educativa que combina l'aprenentatge amb el servei a la comunitat. És un programa que té com a objectiu formar professionals socialment responsables dins del mateix territori. La majoria de projectes APS col·laboren amb una entitat sense ànim de lucre per solucionar una necessitat real. Dins d'aquest, s'ha pogut realitzar un Treball de Fi de Grau amb un impacte directe sobre una comunitat.

La ciutat de Tortosa engega el 2010 un procés comunitari amb l'objectiu de millorar la convivència i cohesió social de la ciutat. A dia d'avui, compta amb una àmplia xarxa d'entitats socials sense ànim de lucre amb estretes relacions entre elles. Aquestes necessiten més participació comunitària per tal d'enfortir la incidència social de les seves tasques, així com la corresponsabilitat ciutadana.

Aquest projecte proposa el disseny d'una aplicació interactiva multiplataforma per tal de posar en contacte entitats socials de la ciutat i persones interessades en realitzar activitats de voluntariat.

En un principi, es volia proposar un desenvolupament només per a Android, però en un projecte amb el propòsit de portar més gent a la participació ciutadana, una aplicació accessible des de qualsevol lloc és més convenient. Per això, és accessible tan des d'una aplicació per a dispositius Android i iPhone, com des d'una web.

Des del punt de vista de l'enginyeria informàtica, el cas ideal seria poder crear una aplicació que funcionés en aquestes plataformes, i no crear 3 aplicacions diferents. El repte és que les tecnologies de desenvolupament multiplataforma són molt recents, no estan completes, però tenen un futur molt prometedor.

---

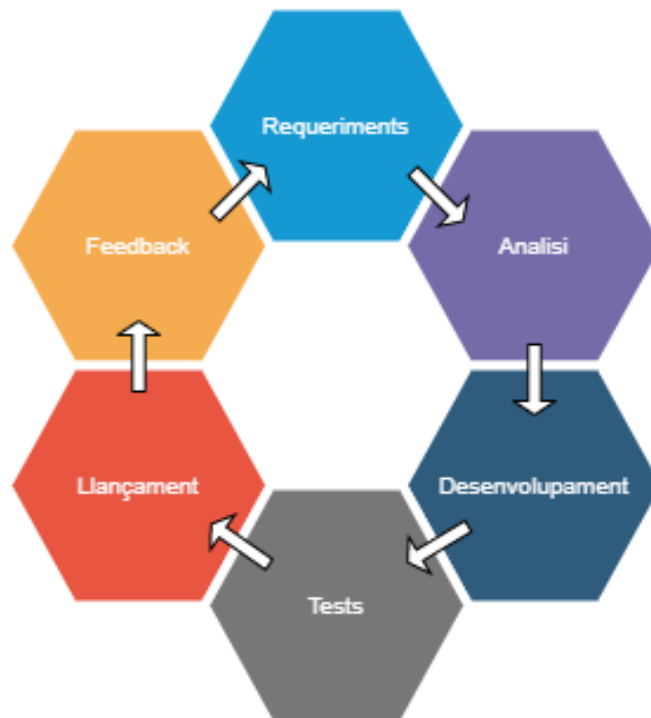
<sup>1</sup>APS = Aprenentatge Servei

## 2 Descripció

El projecte consisteix en la creació d'una aplicació per a Android, iOS<sup>2</sup> i Web. Aquesta contindrà totes les activitats de voluntariat recollides per a que qualsevol persona pugui consultar-les. Les activitats tindran una descripció i informació sobre les entitats involucrades, horaris i dades de contacte, entre d'altres. L'usuari podrà posar-se en contacte amb el responsable o entitat que organitza l'activitat a través del número de telèfon i/o correu electrònic especificats a les dades de contacte.

Una part difícil d'aquest tipus de sistemes és el manteniment. En aquest cas, la pròpia aplicació tindrà una àrea per a administradors. L'administrador haurà d'autenticar-se i des d'allà podrà afegir, esborrar o modificar qualsevol activitat. Com sempre que parlem d'aquest sistema, es podrà accedir des de multitud de dispositius.

Es planteja un projecte a realitzar per *sprints*, seguint la metodologia Agile. S'ha cregut que és el millor model per a aquest projecte, ja que es pot desenvolupar una versió, presentarla per a que els usuaris la puguin utilitzar, recollir *feedback*<sup>3</sup>, i millorar-la en la direcció que l'usuari prefereixi. En aquest projecte, cada *sprint* està representat com a una versió de l'aplicació. Cada versió inclou noves funcionalitats i millora o amplia les anteriors. Durant els següents apartats es farà referència a aquests *sprints* pels que va passar l'aplicació. Cada *sprint* suposa uns nous requeriments, una nova anàlisi, una nova etapa de desenvolupament, més tests, una nova versió a lliurar i *feedback* a recollir.



**Figura 1. Esquema Agile**

---

<sup>2</sup> Es tracta del sistema operatiu creat per Apple utilitzat als seus telèfons, els iPhone.

<sup>3</sup> Feedback és sinònim de respostes, comentaris, crítiques i retroalimentació.

## 2.1 Aplicacions similars

S'ha fet una recerca d'aplicacions de temàtica similar, disponibles tant a nivell nacional com internacional.

### 2.1.1 *European Solidarity Corps*

És el programa de voluntariat de la Comissió Europea. Només tenen [una pàgina web des d'on es poden consultar](#) (Cercador de oportunitats de voluntariat de European Solidarity Corps, sense data). Per defecte, la pàgina principal té un llistat de voluntariats ordenats segons quan s'han afegit a la plataforma. També pot ordenar el llistat segons la data que comença l'activitat o la data fins la qual es pot enviar una sol·licitud per participar (*deadline*). Té un cercador amb filtres per país, dates i tipus.

Quan es clica a una activitat, s'obre una activitat amb una descripció, dates, lloc, tipologia i perfil desitjat del voluntari.

Tot i que el nostre projecte no és internacional, l'aspecte i alguns dels atributs de les activitats podrien servir com a font d'inspiració.

### 2.1.2 *Moviliza-T*

Per tornar a territori nacional, tenim un projecte d'innovació social que té aplicacions per a iOS i Android però no tenen versió Web. Aquesta plataforma té activitats de voluntariat a tot el territori espanyol. Quan s'inicia, es poden veure una sèrie d'activitats destacades. Té un menú amb el que es pot accedir al llistat complet d'activitats o entitats, entre altres opcions. Com que és una aplicació que funciona a un ampli territori, és pot visualitzar un llistat o un mapa. Cada activitat té un logo i una descripció. Es pot escriure i enviar un formulari d'inscripció des de la pròpia aplicació. [Tenen una pàgina web, però no té cap d'aquestes funcionalitats](#). (Lloc Web de Moviliza-T, sense data)

## **3 Requisits**

Requisits: documentació dels requisits funcionals (guions, expressats i documentats com casos d'ús) i requisits no funcionals.

### **3.1 Guions**

#### ***3.1.1 Guió de l'usuari***

Accedir a una llista general amb totes les activitats de voluntariat de la zona. Cada voluntariat està identificat per un codi i també conté informació sobre el nom, descripció, entitats organitzadores (1 o més), horari, dades de contacte, tipus d'activitat, data de començament, data de finalització i data d'expiració. La data de començament i finalització son informació que no només s'ensenyen a l'usuari, no afecten al comportament de l'aplicació. La data d'expiració es tracta de la data que farà que una activitat deixi de ser visible. Accedir a aquests detalls complets de cada activitat de voluntariat. Fer cerca de text sobre tota la informació de l'activitat i veure una llista amb els resultats que coincideixin. Filtrar segons el tipus de voluntariat. Accedir al llistat complet d'entitats. Consultar informació sobre qui ha fet l'aplicació, qui la manté i com s'actualitza. Donar *feedback* per escrit als administradors.

#### ***3.1.2 Guió de l'administrador***

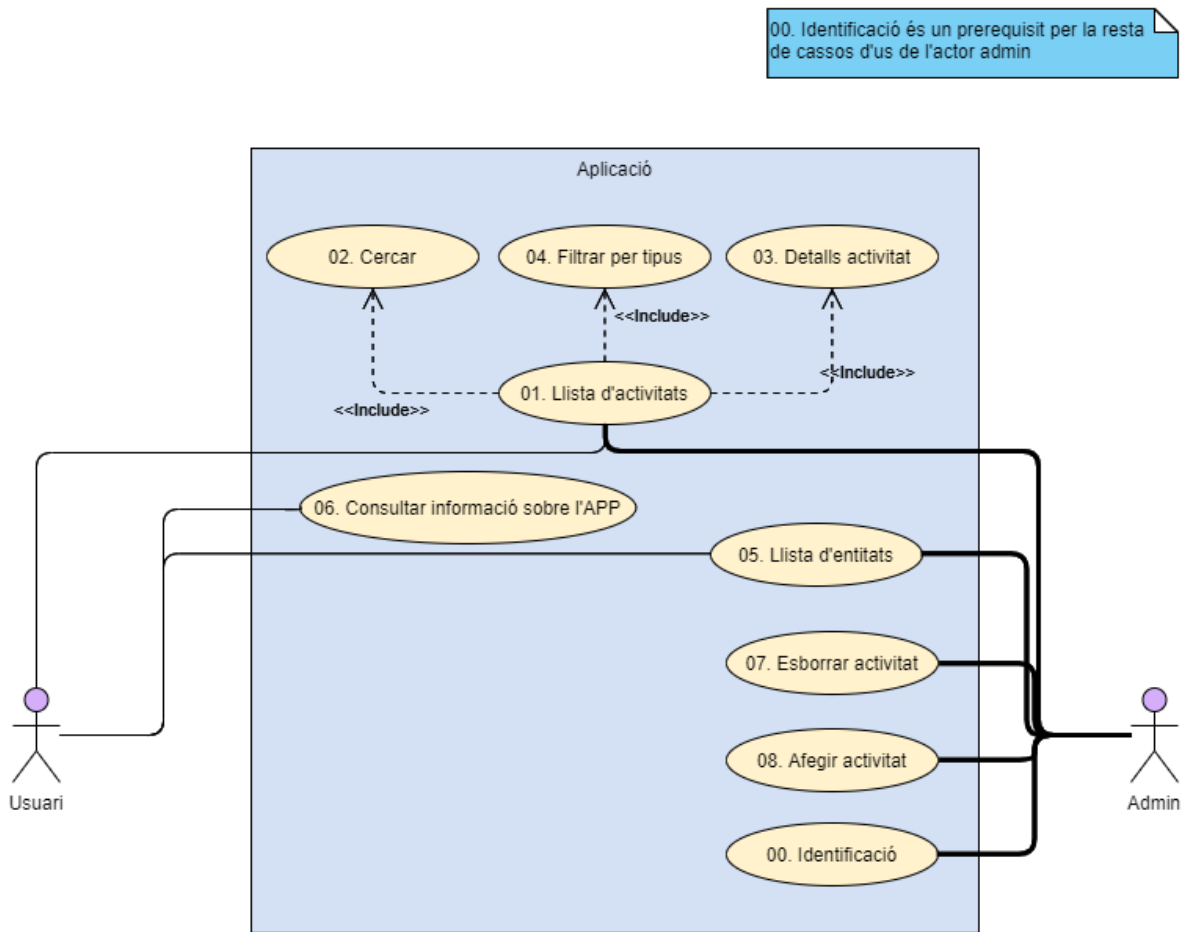
Accedir a la llista general d'activitats, de la mateixa forma que l'Usuari. Accedir als detalls de cada activitat de voluntariat. Afegir una activitat introduint el nom, descripció, entitats organitzadores (1 o més), horari, dades de contacte, tipus d'activitat, data de començament, data de finalització i data d'expiració. Esborrar una activitat. Fer una cerca de text sobre tota la informació de l'activitat i veure una llista amb els resultats. Filtrat segons el tipus de voluntariat. Accedir al llistat complet d'entitats, cadascuna identificada per un identificador. Afegir una entitat a partir del seu nom.

#### ***3.1.3 Altres funcions***

Mentre que els usuaris no necessiten estar autenticats per utilitzar l'aplicació, l'administrador necessita identificar-se amb contrasenya. Els usuaris no tenen dades. Els administradors estan identificats amb un correu electrònic i una contrasenya.

## 3.2 Requisits funcionals

### 3.2.1 Diagrama de cassos d'ús versió 1



**Figura 2. Diagrama de cassos d'ús versió 1**

### 3.2.2 Especificació textual dels cassos d'ús: Versió 1 de l'aplicació

<b>Títol</b>	00. Identificació
<b>Versió aplicació</b>	1.0
<b>Actor</b>	Administrador
<b>Prerequisits</b>	Cap
<b>Descripció</b>	L'administrador ha d'introduir el nom d'usuari i la contrasenya per poder accedir a la resta de funcionalitats de l'aplicació.

**Taula 1. Requisit 00. Identificació**

<b>Títol</b>	01. Llista d'activitats
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Es visualitza un llistat amb totes les activitats disponibles a la base de dades. Cada activitat està representada amb una targeta amb un títol, una descripció i una foto.

**Taula 2.** Requisit 01. Llista d'activitats

<b>Títol</b>	02. Cercar
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Es visualitza una barra per poder introduir text. Quan es realitza la cerca dels termes introduïts, es busca que els termes coincideixin amb algun dels atributs de les activitats i es mostra el resultat en forma d'una nova llista d'activitats. Cada activitat està representada amb una targeta amb un títol, una descripció i una foto.

**Taula 3.** Requisit 02. Cercar

<b>Títol</b>	03. Detalls d'activitat
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	A partir d'un llistat d'activitats, es pot entrar a veure els detalls complets de l'activitat. Aquests detalls són el nom, descripció, entitats organitzadores (1 o més), horari, dades de contacte, data de començament, data de finalització, data d'expiració i tipus d'activitat. Dins de les dades de contacte, qualsevol correu electrònic o telèfon ha de tenir un hiperenllaç per poder enviar un correu o trucar directament.

**Taula 4.** Requisit 03. Detalls d'activitat

<b>Títol</b>	04. Filtrar per tipus
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador

<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Es pot seleccionar un tipus d'activitat i llavors, el llistat d'activitats només mostrarà les activitats amb aquest tipus.

**Taula 5.** Requisit 04. Filtrar

<b>Títol</b>	05. Llistat d'entitats
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Es pot visualitzar un llistat amb totes les entitats de la base de dades.

**Taula 6.** Requisit 05. Llistat d'entitats

<b>Títol</b>	06. Consultar informació sobre l'APP
<b>Versió</b>	1.0
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Es pot visualitzar un text amb tota la informació sobre la creació i el suport de l'aplicació.

**Taula 7.** Requisit 06. Consultar informació sobre l'APP

<b>Títol</b>	07. Formulari per <i>feedback</i>
<b>Versió</b>	1.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Com que aquest és un projecte dinàmic, en el que es fàcil ampliar l'aplicació i actualitzar-se a noves versions, un sistema per recollir <i>feedback</i> dels usuaris és molt adient. Aquesta funcionalitat consistiria en un formulari on es pot introduir text i l'usuari pot explicar les seves incidències i opinions.

**Taula 8.** Requisit 07. Consultar informació sobre l'APP

<b>Títol</b>	08. Afegir activitat
<b>Versió</b>	1.0
<b>Actor</b>	Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat

<b>Descripció</b>	Es pot emplenar un formulari per especificar tots els atributs de l'activitat i afegir-la a la base de dades.
-------------------	---

**Taula 9.** Requisit 08. Afegir activitat

<b>Títol</b>	09. Esborrar activitat
<b>Versió</b>	1.0
<b>Actor</b>	Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	L'administrador pot esborrar activitats concretes del servidor. Cap actor podrà tornar a veure aquestes activitats esborrades.

**Taula 10.** Requisit 09. Esborrar activitat

### 3.2.3 Diagrama de cassos d'ús versió 2

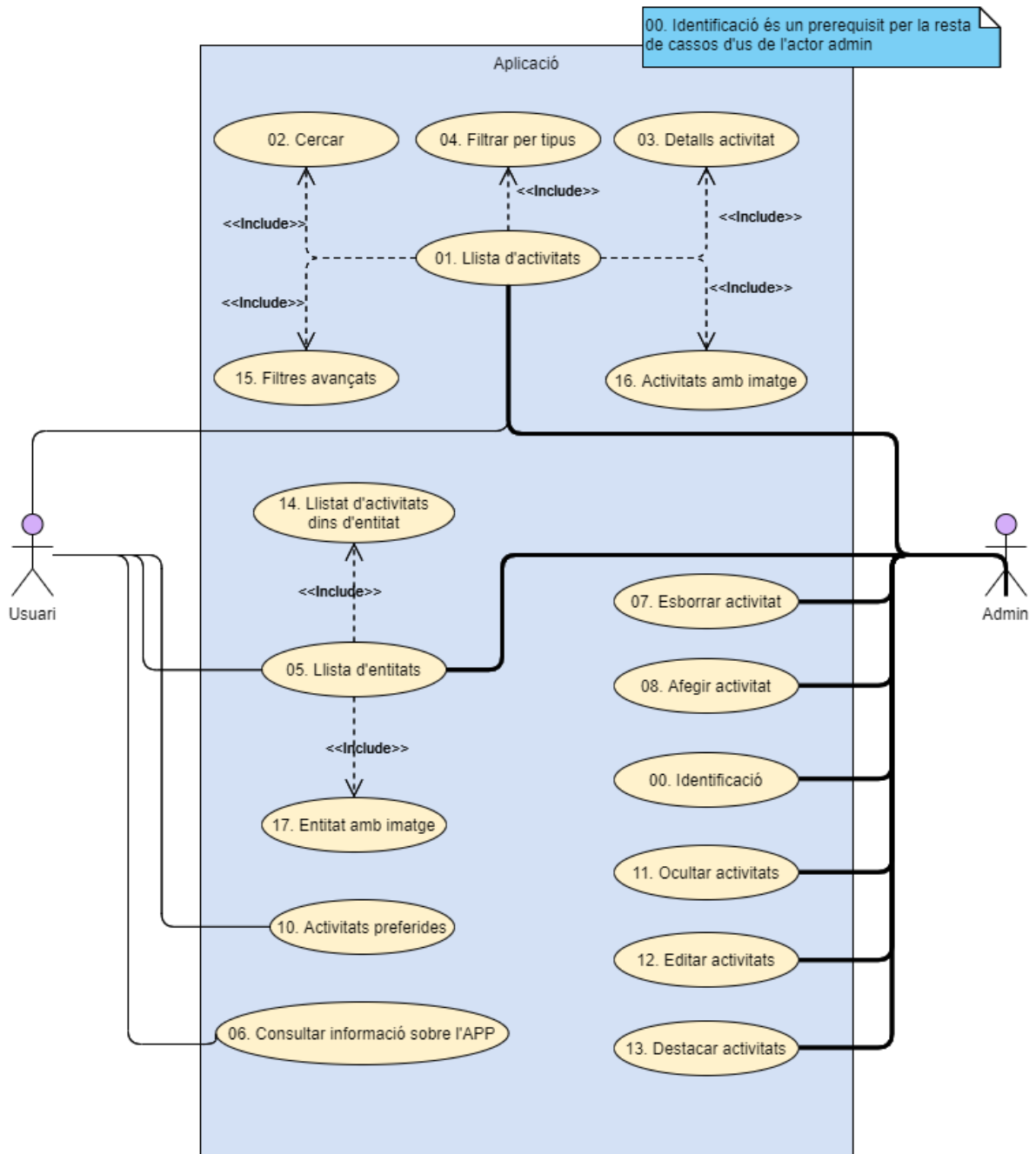


Figura 3. Diagrama de cassos d'ús versió 2

### 3.2.4 Especificació textual dels cassos d'ús: Versió 2 de l'aplicació

<b>Títol</b>	10. Activitats preferides
<b>Versió</b>	2.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Cada element de la llista principal d'activitats tindrà l'opció d'afegir-lo a favorits. D'aquesta forma, cada usuari podrà tenir la seva pròpia

	l·lista d'activitats preferides. Aquesta informaci3 es guardarà a la mem3ria interna de l'aplicaci3 i no al servidor.
--	---

**Taula 11.** Requisit 10. Activitats preferides

<b>Títol</b>	11. Ocultar activitats
<b>Versió</b>	2.0
<b>Actor</b>	Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripci3</b>	L'administrador pot ocultar activitats per a que no siguin visibles per a la resta d'usuaris. L'administrador podrà seguir veient l'activitat de forma normal. També és possible retornar una activitat al seu estat "visible a tots els usuaris".

**Taula 12.** Requisit 11. Ocultar activitats

<b>Títol</b>	12. Editar activitats
<b>Versió</b>	2.0
<b>Actor</b>	Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripci3</b>	L'administrador pot editar la informaci3 d'una activitat utilitzant un formulari similar a l'utilitzat per crear-la.

**Taula 13.** Requisit 12. Editar activitats

<b>Títol</b>	13. Destacar activitats
<b>Versió</b>	2.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripci3</b>	L'administrador pot fer que una activitat estigui destacada. Aix3 significa que els usuaris la veuran al principi de la l·lista d'activitats.

**Taula 14.** Requisit 13. Destacar activitats

<b>Títol</b>	14. L·lista d'activitats dins d'una entitat
<b>Versió</b>	2.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripci3</b>	A partir de la l·lista d'entitats, els actors poden accedir a la l·lista d'activitats organitzades per una entitat.

**Taula 15.** Requisit 14. L·lista d'activitats dins d'una entitat

<b>Títol</b>	15. Filtres avançats
<b>Versió</b>	2.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	De la mateixa forma que hi ha filtres segons tipus, aquesta implementació introdueix nous filtres. Alguns exemples són: <ul style="list-style-type: none"> <li>- Segons disponibilitat</li> <li>- Segons data de començament i final</li> <li>- Segons horari</li> </ul>

**Taula 16.** Requisit 15. Filtres avançats

<b>Títol</b>	16. Activitat amb imatge
<b>Versió</b>	2.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Els detalls de l'activitat tenen una imatge. L'administrador pot penjar la imatge a través de la seva àrea d'administrador. L'usuari no pot canviar la imatge.

**Taula 17.** Requisit 16. Activitat amb imatge

<b>Títol</b>	17. Entitat amb imatge
<b>Versió</b>	2.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	A la llista d'entitats, cada entitat té una imatge. L'administrador pot assignar-les.

**Taula 18.** Requisit 17. Entitat amb imatge

### 3.2.5 Especificació textual dels cassos d'ús: Versió 3 de l'aplicació

<b>Títol</b>	18. Ordenació
<b>Versió</b>	3.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Permetre l'ordenació segons algun criteri (data, nom, tipus...). L'actor podrà seleccionar quin és el criteri i llavors la llista d'activitats s'ordenarà d'aquesta forma.

**Taula 19.** Requisit 18. Ordenació

<b>Títol</b>	19. Notificacions push
<b>Versió</b>	3.0
<b>Actor</b>	Administrador, Usuari
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Permetre que l'administrador pugui enviar una notificació a tots els usuaris de l'aplicació. Aquesta notificació ha d'estar enllaçada a una activitat. Per tant, l'usuari pot prémer la notificació per accedir als detalls de l'activitat.

**Taula 20.** Requisit 19. Notificacions push

<b>Títol</b>	20. Tutorial inicial
<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	La primera vegada que l'usuari entra a la aplicació es mostra un tutorial guiat i interactiu sobre les funcionalitats principals de l'aplicació. El tutorial no s'ha de repetir mai, només es mostra la primera vegada.

**Taula 21.** Requisit 20. Tutorial inicial

<b>Títol</b>	21. Registre i identificació d'usuaris
<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Permetre que els usuaris puguin registrar-se i identificar-se dins de l'aplicació. D'aquesta forma, les activitats preferides passarien a guardar-se al servidor, dins de la informació de cada usuari registrat.

**Taula 22.** Requisit 21. Registre i identificació d'usuaris

<b>Títol</b>	22. Localització d'activitats en un mapa
<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Igual que es poden veure les activitats en forma de llista, aquesta funcionalitat representaria en un mapa cada activitat al lloc on es realitza.

**Taula 23.** Requisit 22. Localització d'activitats en un mapa

<b>Títol</b>	23. Múltiples idiomes
--------------	-----------------------

<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Permetre que l'aplicació estigui en múltiples idiomes (Català, Castellà, Anglès)

**Taula 24.** Requisit 23. Múltiples idiomes

<b>Títol</b>	24. Entitats favorites
<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Cada element de la llista d'entitats tindrà l'opció d'afegir-lo a favorits. D'aquesta forma, cada usuari podrà tenir la seva pròpia llista de d'entitats preferides.

**Taula 25.** Requisit 24. Entitats favorites

<b>Títol</b>	25. Imatges avançades
<b>Versió</b>	3.0
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Des del punt de vista de l'administrador, permetre que una activitat tingui més d'una imatge, que es puguin retallar i seleccionar la qualitat d'aquestes.  Des del punt de vista de l'usuari, es podran veure les imatges i fer zoom sobre aquestes.

**Taula 26.** Requisit 25. Imatges avançades

### 3.3 Requisits no funcionals

#### 3.3.1 Requisits de memòria

En referència a la memòria només influïm les aplicacions d'Android i iOS. Tenim 2 parts, la memòria que ocupen els instal·ladors, i la memòria de l'aplicació.

L'instal·lador APK<sup>4</sup> per a Android hauria d'ocupar menys de 50MB, l'anterior límit per aplicacions. Ara es tracta de 100MB. Si es sobrepassen els 100MB, [poden haver-hi complicacions per pujar l'aplicació a Google Play.](#) (Android Developer Guides for

---

<sup>4</sup> APK = Android Application Package. És el format de fitxer utilitzat per instal·lar una aplicació a Android.

[APK Expansion Files, sense data](#)) Tot i així, es deixarà el límit a 50MB. Aquest límit també s'aplicarà a iOS.

L'aplicació ja instal·lada hauria d'ocupar el mínim possible, per tal que la gent la pugui tenir instal·lada sense fer compromisos. Per això hauria d'ocupar no més de 200MB.

En cas que les mides anteriors es sobrepassin, també es plantejarà dividir l'aplicació en dos: una aplicació pels usuaris i una altre pels administradors. Tot i així, la mida de les aplicacions als smartphones ja no és tan important com el rendiment, especialment avui en dia que cada vegada és veuen més telèfons amb 128 i 256 GB.

### **3.3.2 Requisits de rendiment**

Els temps de resposta a les peticions de l'usuari han molt ràpids. L'optimització de la memòria juga una part important per tenir un bon rendiment, però no és l'únic factor. L'aplicació hauria de mantenir els 60 fotogrames per segon a tots els dispositius i inclús anar més ràpid a aquells dispositius que ho permetin.

### **3.3.3 Requisits d'usabilitat**

L'aplicació seguirà les [directives d'usabilitat de Material Design](#), (Guidelines Material Design, sense data) molt utilitzada a sistemes Android. Aquest llenguatge de disseny simplifica l'ús de components (botons, barres de text, fotos...). Per la simplicitat i el fet de que es pugui assemblar a altres aplicacions ja fetes (i no només les de Google), crec que és un conjunt de directives adien. [Material Design es recomana també per Web \(Material Design for Web, sense data\)](#) i s'utilitzarà d'aquesta forma.

Encara que Material Design es pot utilitzar a iOS [i així s'expressa a la pagina web de les directives de Material Design](#), (Material Design for iOS, sense data) el sistema operatiu de Apple té les seves pròpies directives de disseny i d'usabilitat: Flat Design. Flat Design (o Flat) no es molt diferent, en el sentit de que proposa unes directives que fan que la usabilitat sigui més simple i recordi a altres aplicacions d'iOS. Crec que adientment, s'hauria d'utilitzar Flat per la versió d'iOS, però donat que altres desenvolupadors han utilitzat Material a iOS en el passat, el que causaria menys contratemps durant el procés de desenvolupament seria utilitzar Material per a totes les versions de l'aplicació.

Tot i així, crec que el pas de Material a Flat per la versió d'iOS és un pas a fer de cara a futures versions de l'aplicació. Com que al nostre país, la gran majoria de gent té un dispositiu Android, aquesta es una altre raó per prioritzar la versió d'Android. Això sí, sense deixar enrere a la minoria que podria utilitzar la plataforma des del dispositiu d'Apple.

### **3.3.4 Requisits de fiabilitat**

L'aplicació no ha de guardar dades sobre els usuaris i per tant, no hi ha cap dada que es pugui comprometre des del punt de vista de l'usuari.

Des del punt de vista de l'administrador, si que es guarda un correu electrònic i una contrasenya, per tant, aquestes dades han de complir la EU General Data Protection Regulation (GDPR)<sup>5</sup>.

---

<sup>5</sup> El Reglament General de Protecció de Dades és el conjunt de regulacions de la llei de la Unió europea que tracten la protecció de dades i la privacitat.

### ***3.3.5 Requisites de portabilitat***

Com ja s'ha dit prèviament, l'aplicació serà accessible des de qualsevol navegador web, i tindrà aplicacions per Android i Apple. D'aquesta forma es podrà arribar a tothom qui tingui connexió a internet.

## 4 Anàlisi dels requisits

Anàlisi dels requisits funcionals (al menys, diagrama de classes i de seqüència dels casos d'ús)

### 4.1 Diagrama de classes

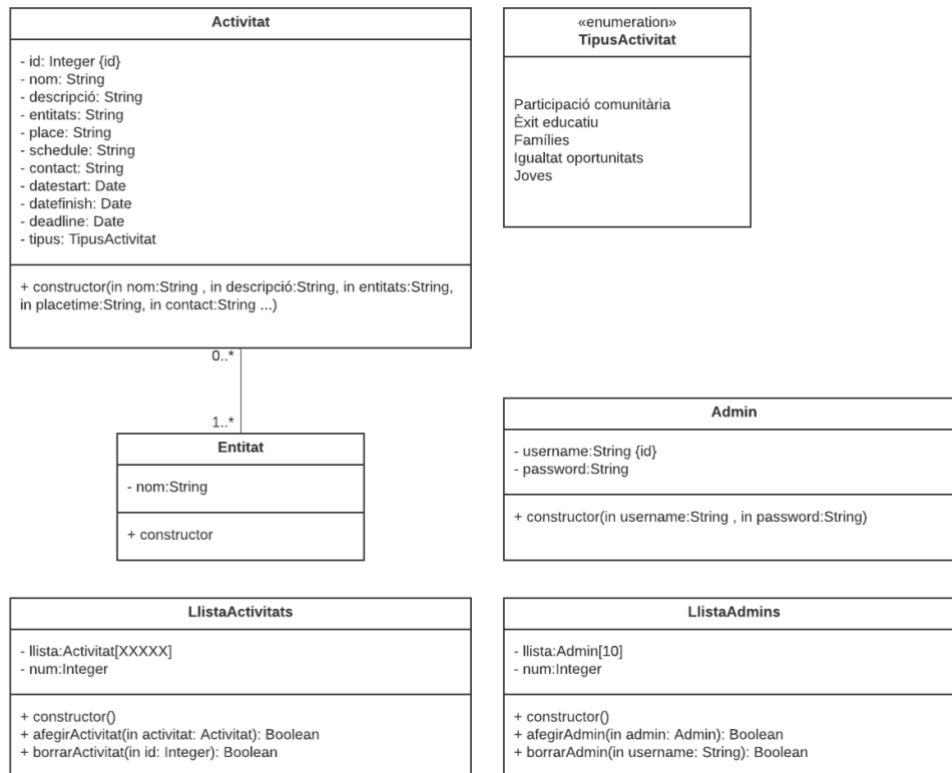


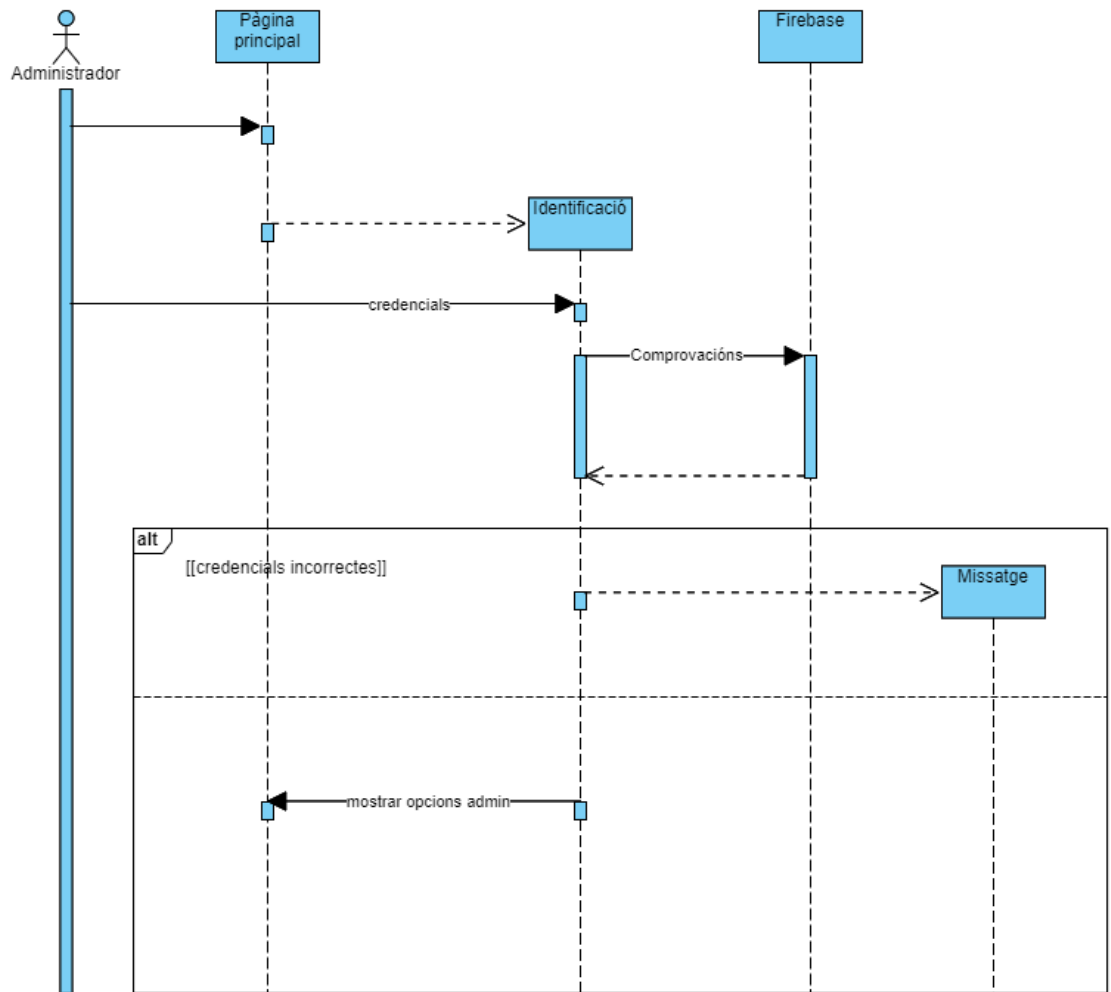
Figura 4. Prototip de filtres avançats

### 4.2 Diagrames de seqüències dels cassos d'ús

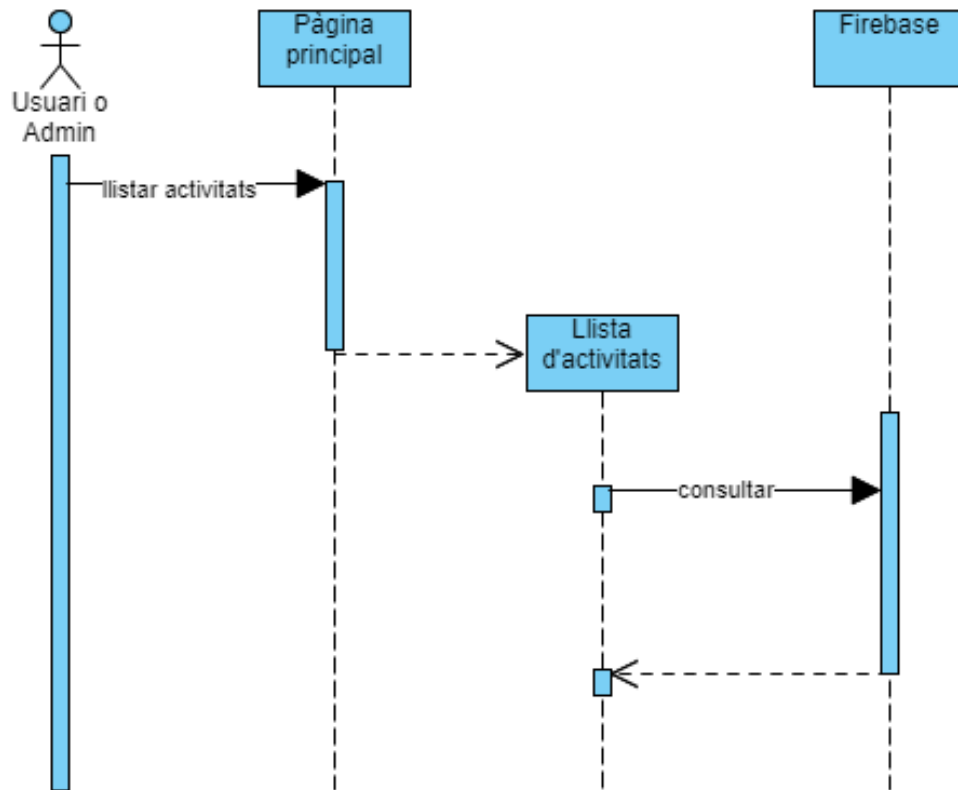
Segons el que es va estudiar a l'assignatura d'Anàlisi i Disseny d'Aplicacions, un diagrama de seqüències és un diagrama que representa els missatges intercanviats entre parts de l'aplicació en funció del temps. Les accions de cada cas d'ús comencen a dalt i acaben a baix conforme passa el temps.

Cada part de l'aplicació té una línia de vida (línia vertical discontinua) que es crea al principi del cas d'ús o es creada per altres parts de l'aplicació. Els missatges estan representats per línies horitzontals cap a altres parts de l'aplicació. Es poden utilitzar condicionals amb les caixes **alt**, que simbolitzen un condicional amb una o dos condicions.

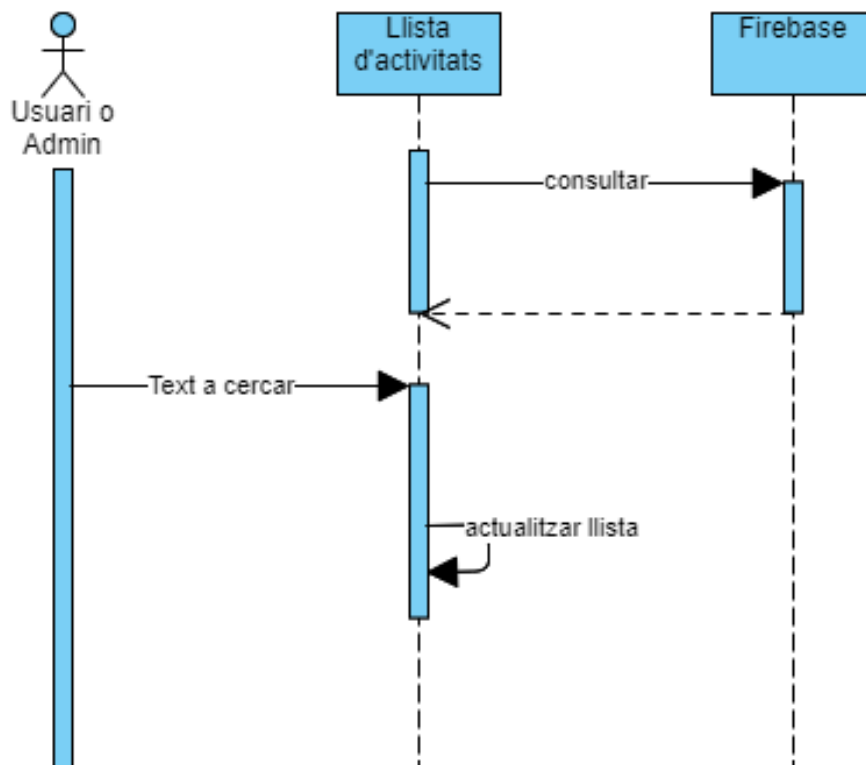
Per fer els diagrames de seqüència de cassos d'ús, s'ha utilitzat [una eina online anomenada Online Visual Paradigm](#). (Online Visual Paradigm, sense data)



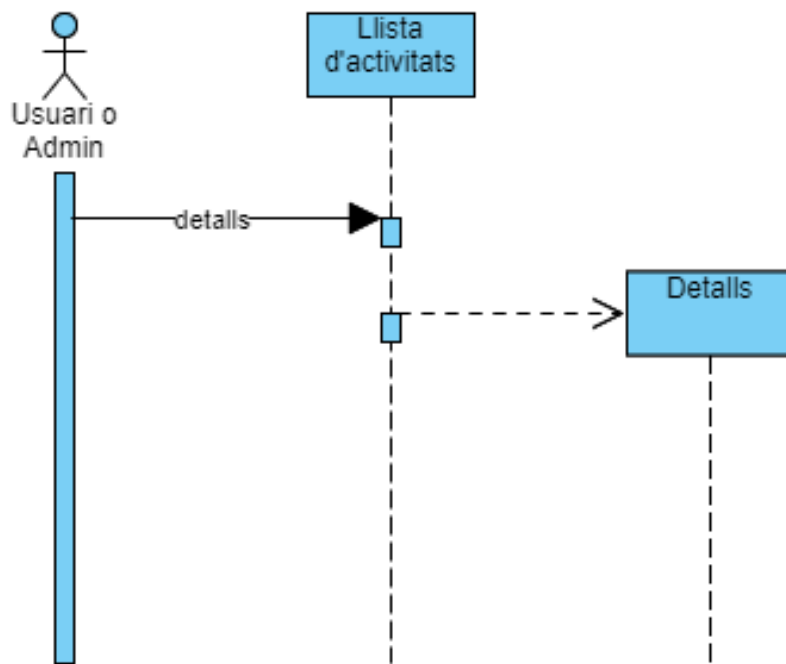
**Figura 5.** Seqüència del cas d'ús 00: Identificació



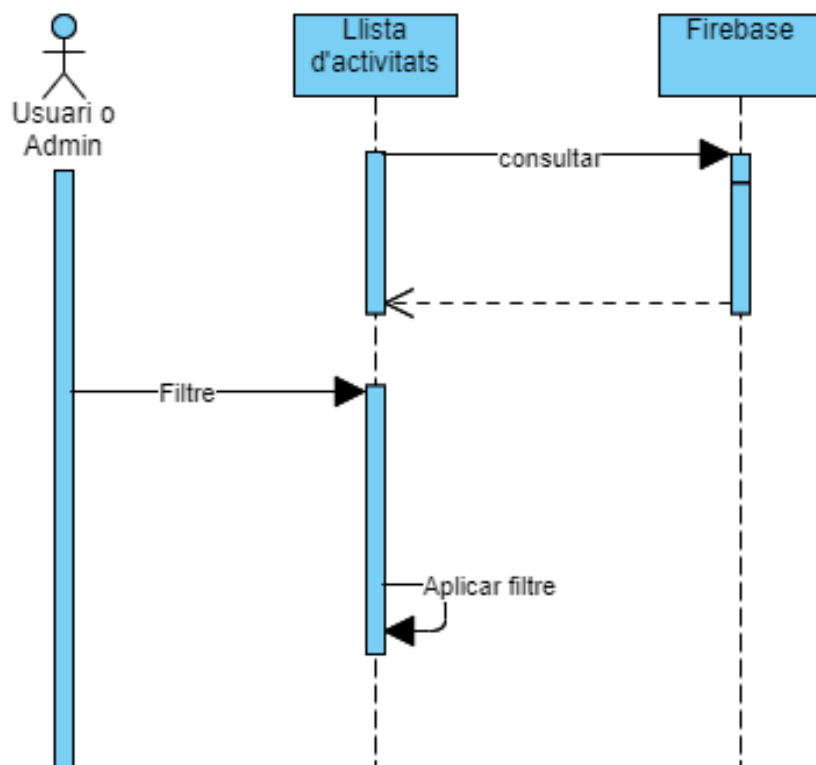
**Figura 6.** Seqüència del cas d'ús 01: Llista d'activitats



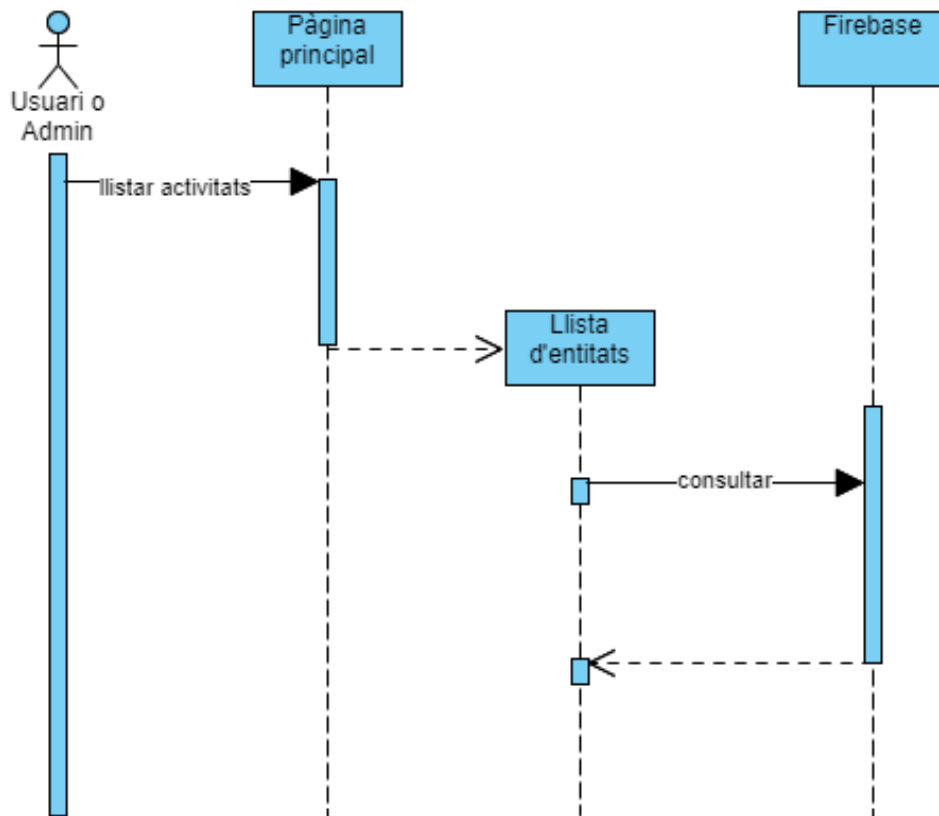
**Figura 7.** Seqüència del cas d'ús 02: Cercar



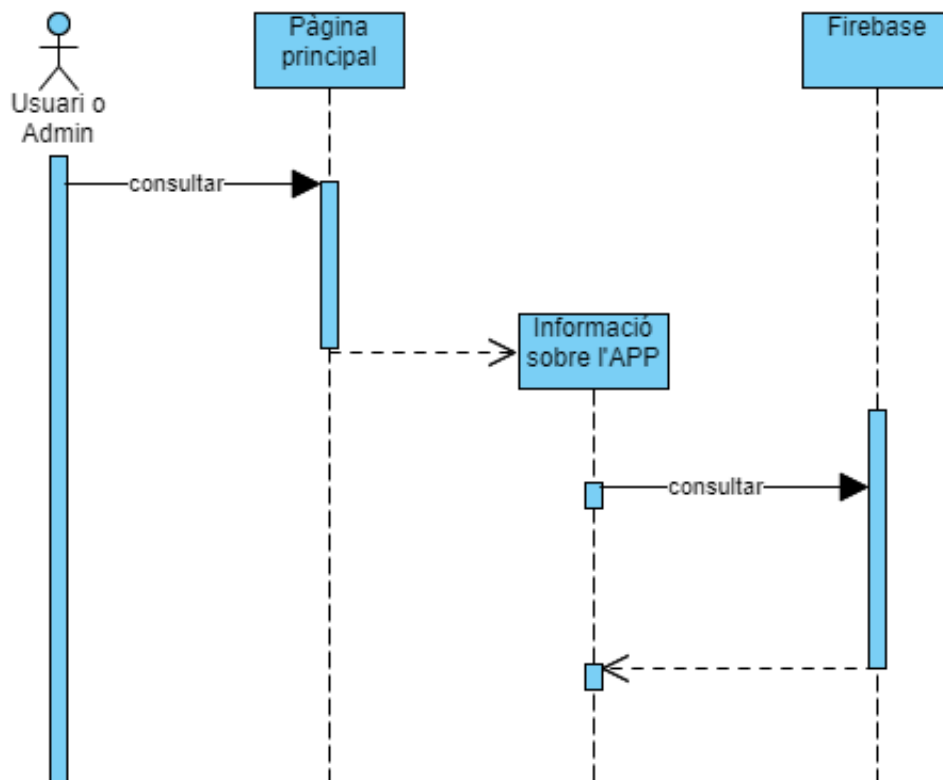
**Figura 8.** Seqüència del cas d'ús 03: Detalls d'activitat



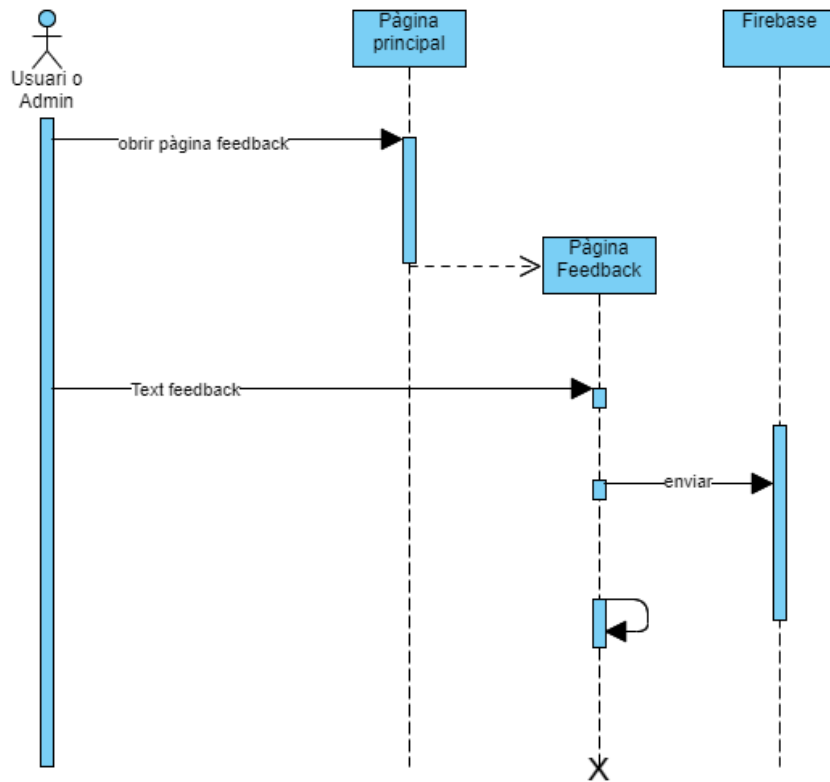
**Figura 9.** Seqüència del cas d'ús 04: Filtrar per tipus



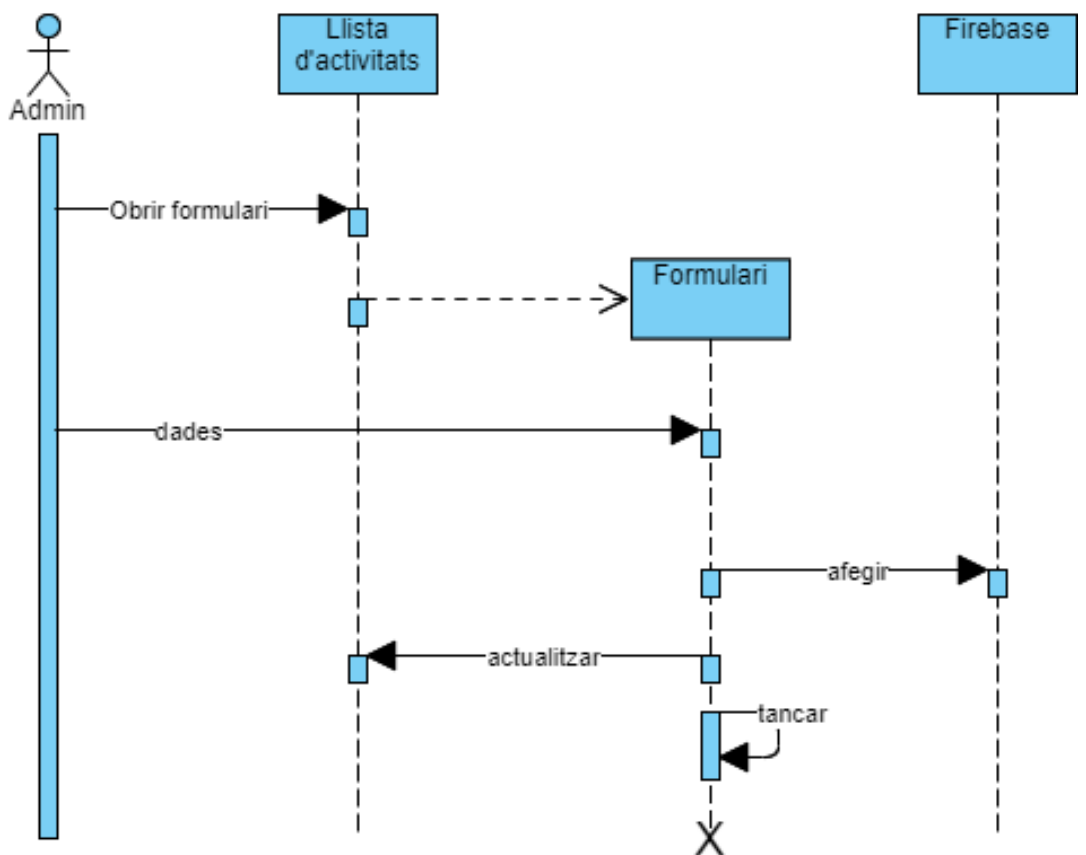
**Figura 10.** Seqüència del cas d'ús 05: Llistat d'entitats



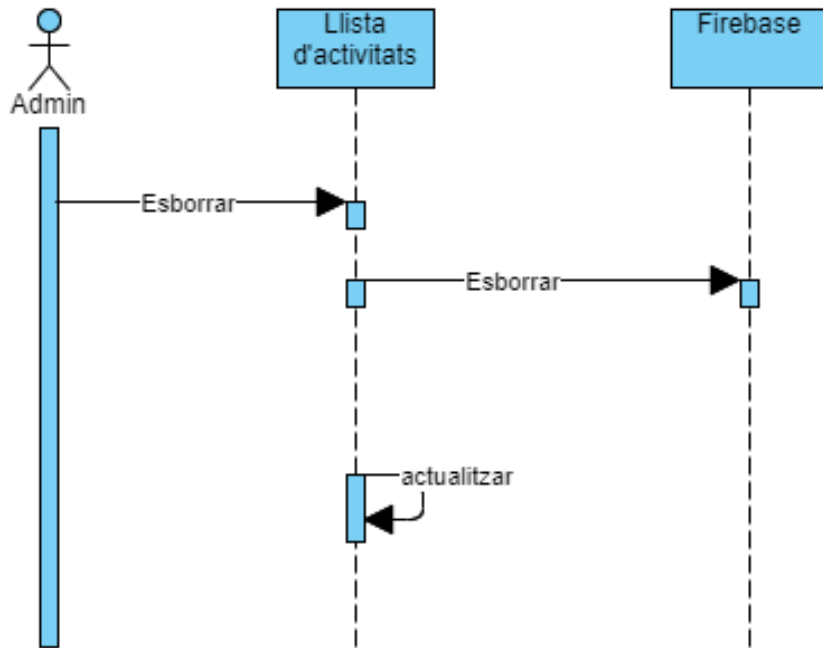
**Figura 11.** Seqüència del cas d'ús 06: Consultar informació sobre l'APP



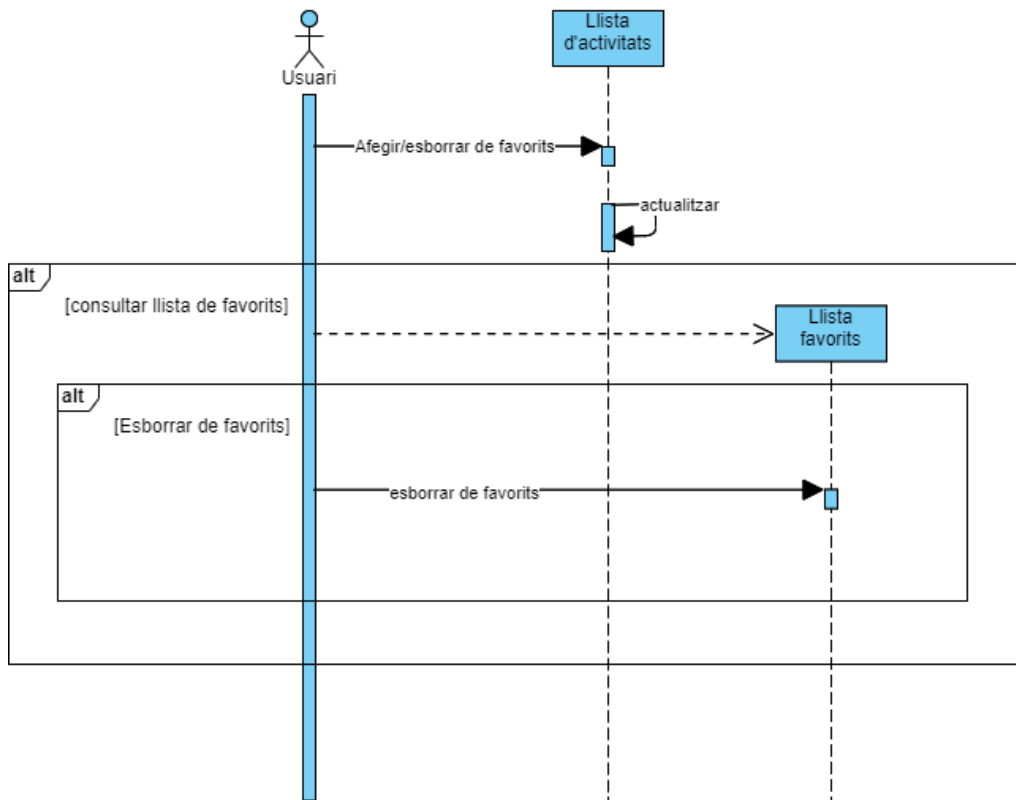
**Figura 12.** Seqüència del cas d'ús 07: Formulari per feedback



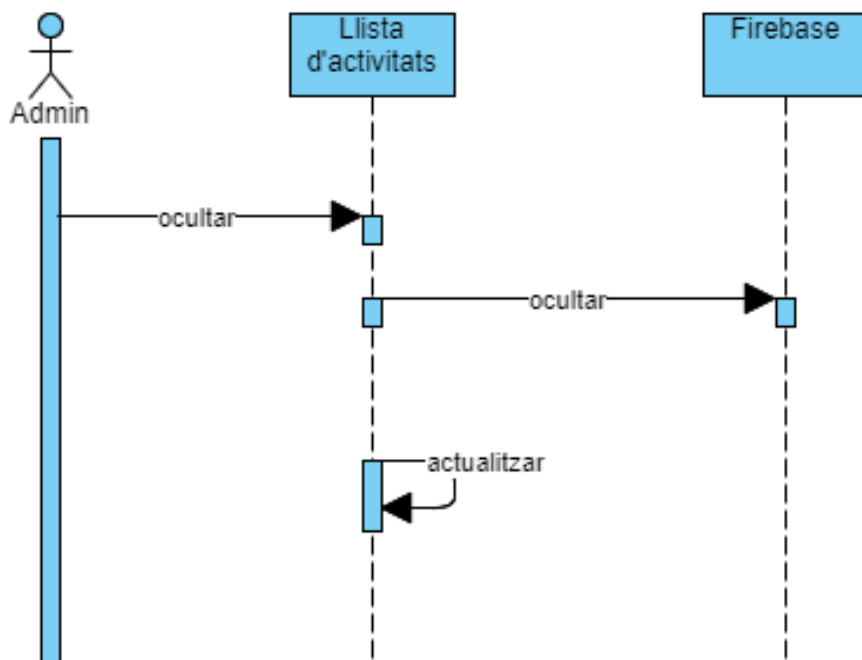
**Figura 13.** Seqüència del cas d'ús 08: Afegir activitat



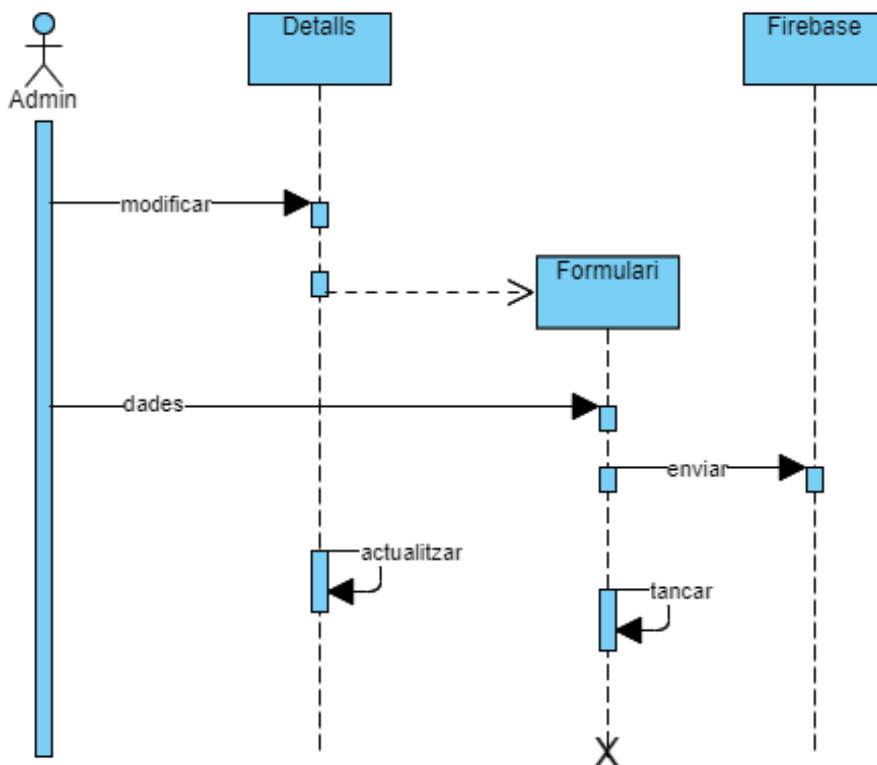
**Figura 14.** Seqüència del cas d'ús 09: Esborrar activitat



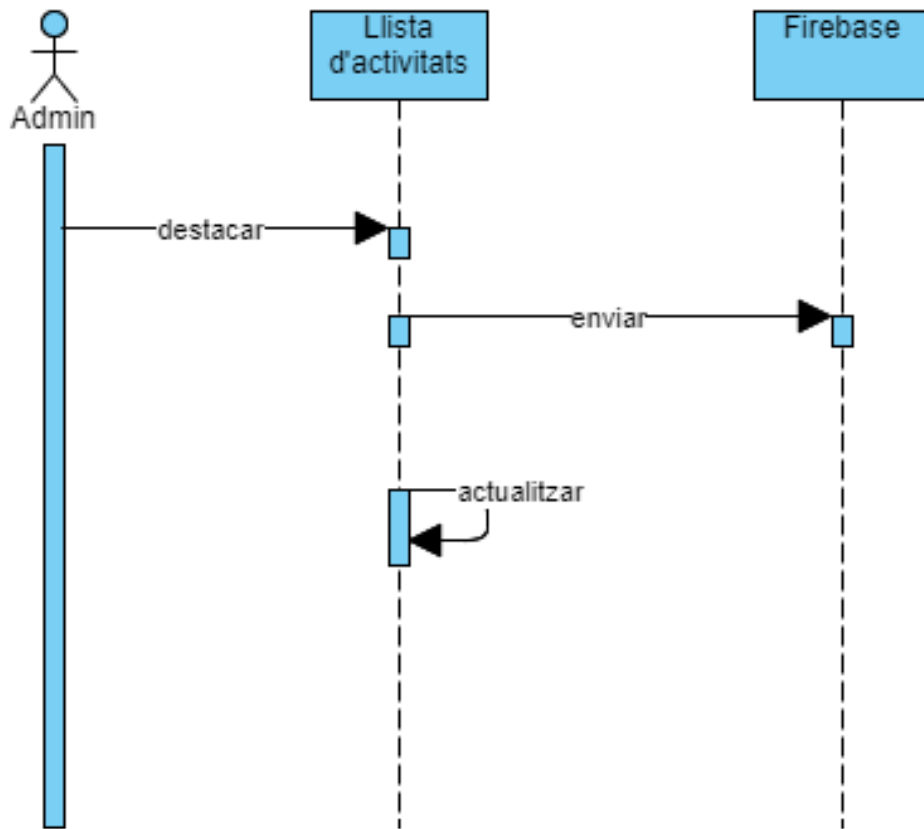
**Figura 15.** Seqüència del cas d'ús 10: Activitats preferides



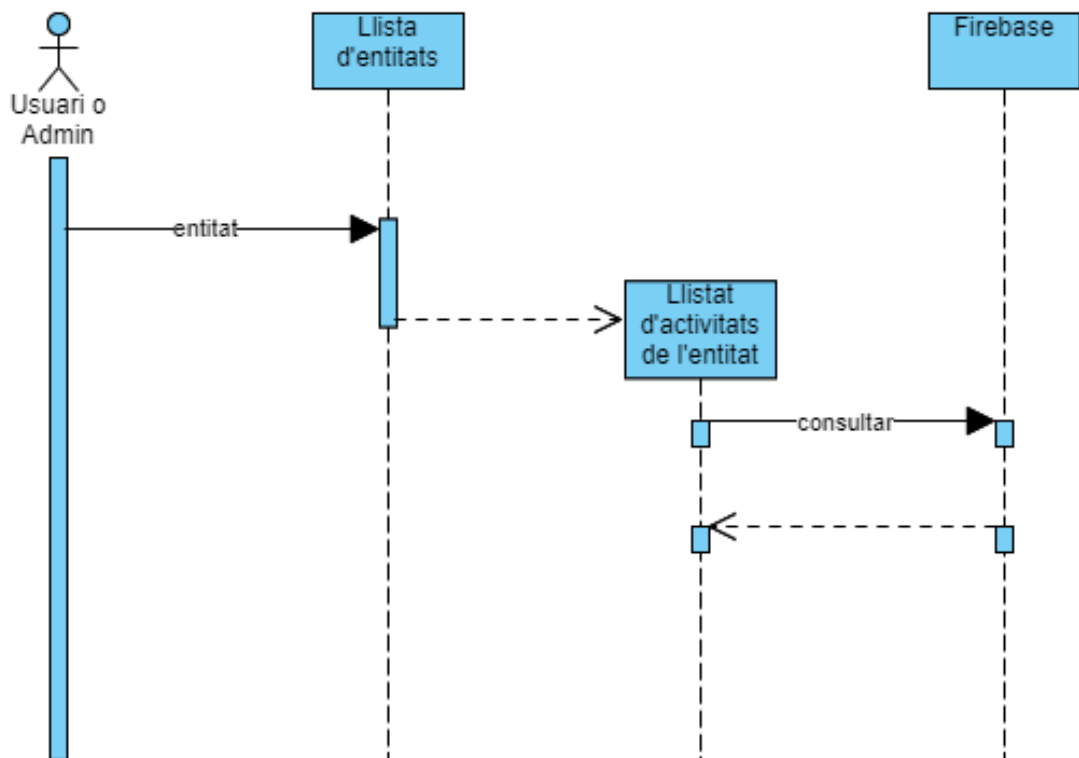
**Figura 16.** Seqüència del cas d'ús 11: Ocultar activitats



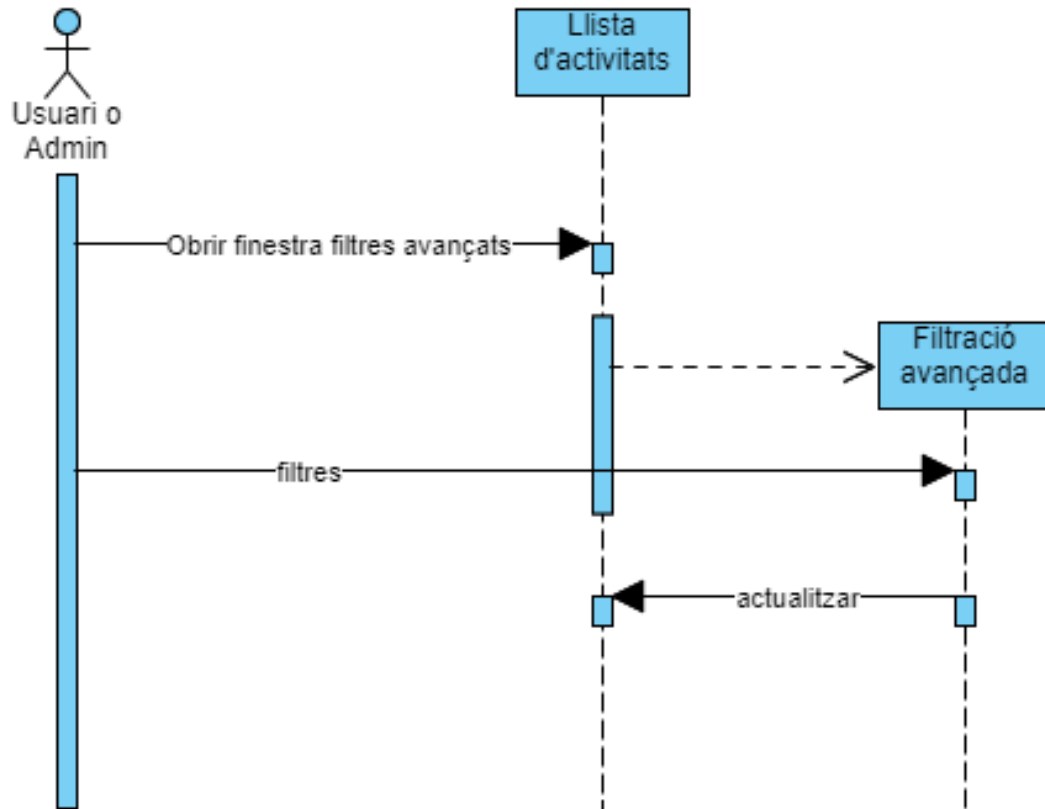
**Figura 17.** Seqüència del cas d'ús 12: Editar activitats



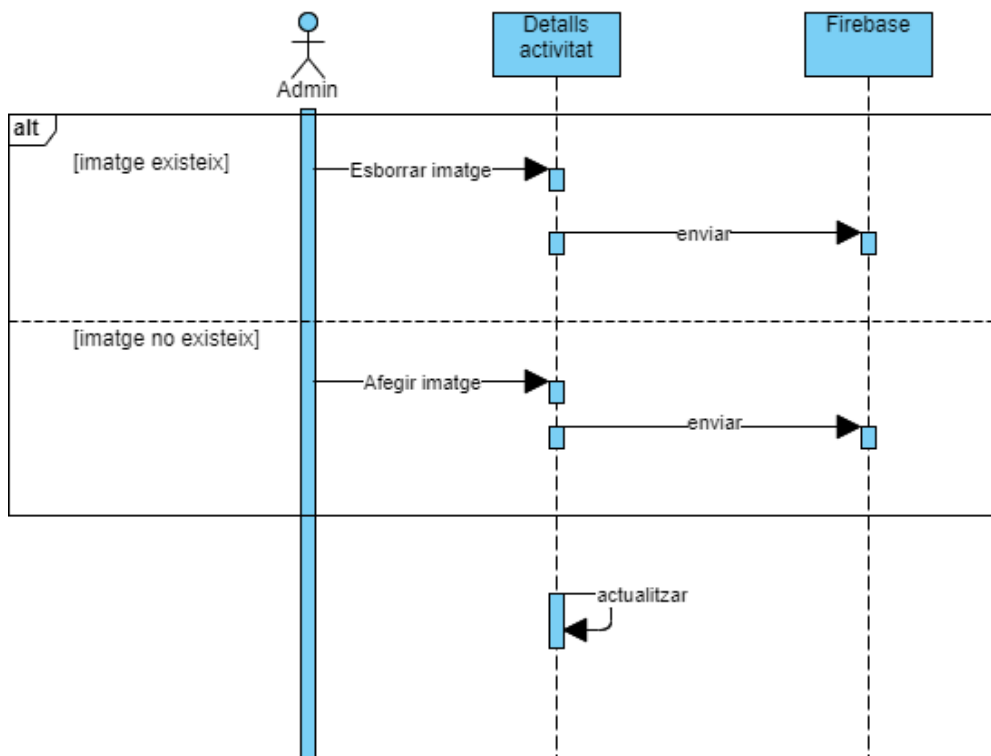
**Figura 18.** Seqüència del cas d'ús 13: Destacar activitats



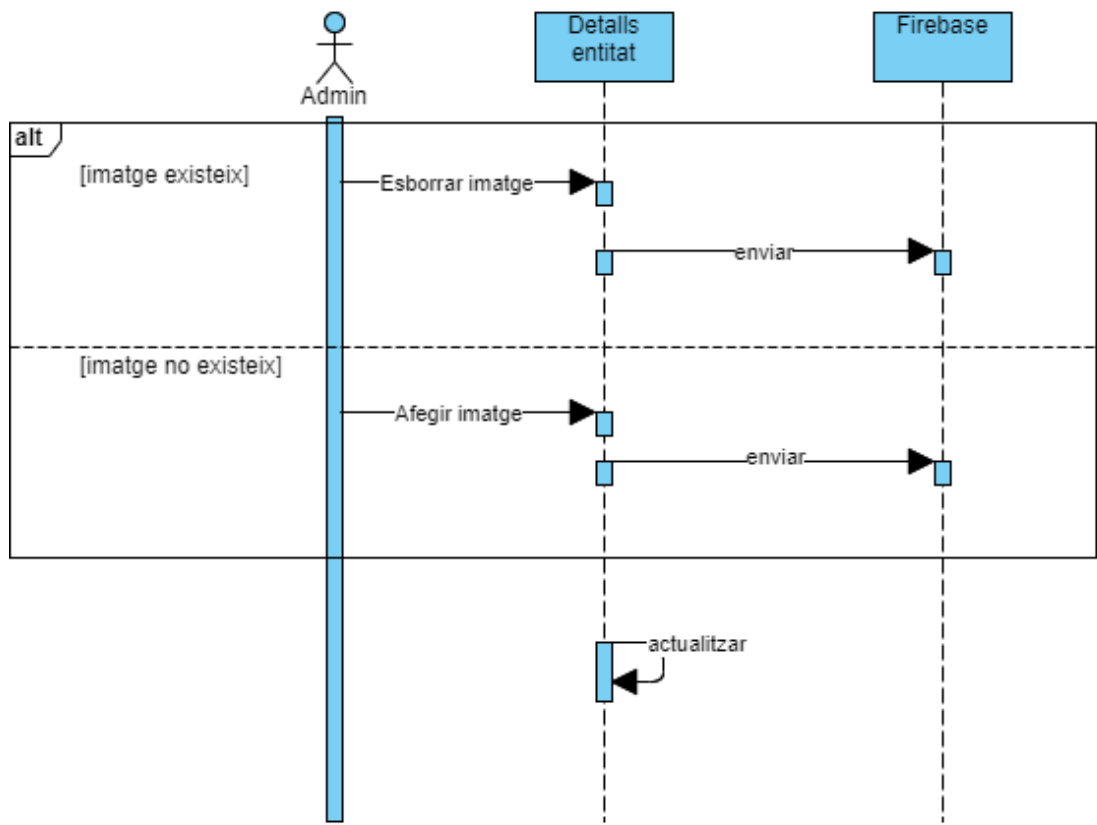
**Figura 19.** Seqüència del cas d'ús 14: Llista d'activitats dins de l'entitat



**Figura 20.** Seqüència del cas d'ús 15: Filtres avançats



**Figura 21.** Seqüència del cas d'ús 16: Activitat amb imatge



**Figura 22.** Seqüència del cas d'ús 17: Entitat amb imatge

## 5 Disseny

Disseny. Arquitectura de l'aplicació, disseny de la interfície gràfica, disseny de la persistència de dades (BD)...

### 5.1 Tecnologies considerades

La elecció de la tecnologia, es remunta a uns mesos abans de la obtenció dels requisits complets.

#### 5.1.1 Kotlin per al desenvolupament Android

Durant l'assignatura de Aplicacions Mòbils i Encastades, es va fer una aplicació Android utilitzant Java. En aquesta assignatura també es va donar una introducció al nou llenguatge oficial de Android: Kotlin. Aquest llenguatge que es promocionava com un nou llenguatge orientat a objectes que ho simplifica tot. Un llenguatge s'ha anat millorant amb els anys fins que a 2019 es va convertir en el llenguatge de referència per desenvolupament Android. Té una documentació excel·lent, com de costum als llenguatges i entorns de desenvolupament de Google. A la consulta anual de Stackoverflow figura com [el quart llenguatge més agradat i el sisè més buscat a programadors per les empreses](#). (2020 Developer Survey StackOverflow, sense data)

Quan vaig presentar la sol·licitud per fer aquest projecte, l'ajuntament demanava una aplicació, sense especificar la plataforma. Vaig pensar en que a Catalunya la gran majoria de usuaris utilitzen Android i d'altra banda, Kotlin és un llenguatge amb el que em vaig quedar amb ganes durant l'assignatura d'Aplicacions Mòbils i Encastades.

Durant el mes abans de tindre la primera reunió amb l'ajuntament de Tortosa, vaig fer alguns prototips d'aplicacions per practicar amb l'entorn d'Android i Kotlin. També vaig estar utilitzant Firebase pel emmagatzemament de dades. Firebase és una plataforma (també de Google) que facilita molt l'emmagatzemament de dades per dispositius mòbils. És va fer pensant en Android i és part de l'ecosistema de desenvolupament de Google, en el que se li posen les coses fàcils al programador.

La idea d'utilitzar Kotlin va canviar quan es van reconsiderar les necessitats del projecte.

#### 5.1.2 Tecnologies multiplataforma

Després de rebre una reunió amb l'ajuntament i rebre uns requisits més complets, es va considerar la possibilitat de les tecnologies multiplataforma.

El motiu pel qual l'ajuntament de Tortosa vol realitzar aquest projecte és per promoure les activitats de voluntariat a la ciutat. Una aplicació d'aquest tipus podria fer que més gent participés en les activitats. Per tant, pot ser que fer una aplicació només per Android, es quedi curt en l'abast d'usuaris als que arriba.

Una opció per arribar a més persones seria programar en paral·lel una pagina web, una aplicació per Android i un aplicació per iOS. D'aquesta forma estaríem arribant al 100% dels usuaris de dispositius mòbils i ordinadors. Seria una solució perfecta, això si, amb un procés de desenvolupament molt extens.

L'altre opció passa per utilitzar una tecnologia que pugui crear una aplicació fàcilment transportable a totes 3 plataformes a partir del mateix codi. Aquests tipus de tecnologia existeixen des de fa pocs anys i [les més populars són les següents](#). (Statistia: Cross-platform

mobile frameworks used by software developers worldwide in 2019 and 2020, sense data) He escollit fer un anàlisi de les 5 més populars, tot i que hi ha altres al mercat.

- Xamarin: tot i que ara està perdent popularitat, aquest es un dels primers frameworks per desenvolupar en mòbils. Com que es part del ecosistema de Microsoft .NET, utilitza C#. Crec que la opinió dels programadors és important i les opinions no són les millors [a la enquesta de 2020 de Stackoverflow](#). (2020 Developer Survey StackOverflow, sense data)
- Cordova/PhoneGap: es tracta d'altre dels frameworks més antics, que fins i tot ha canviat de nom i va tindre molta popularitat entre el 2010 i el 2015. Si l'anterior framework no tenia les millors opinions, aquí tenim al framework menys buscat per les empreses i el segon més temut pels usuaris.
- Ionic: un projecte molt interessant, donat que és basa en open-source com a filosofia. Està construït pel damunt de AngularJS i Cordova. És la tercera tecnologia multiplataforma més utilitzada.
- React Native: Aquesta es la plataforma més gran i més utilitzada. Utilitza Javascript, un dels llenguatges [més populars i buscat per les empreses](#). Dos aplicacions que utilitzen aquest framework són Facebook i Instagram, ja que la plataforma va ser creat per la companyia de Mark Zuckerberg. Porta 5 anys en el mercat, suficient per tindre la comunitat més gran de desenvolupadors multiplataforma i una documentació excel·lent. De fet, està dins del [top 10 repositoris de GitHub<sup>6</sup> a Juny de 2021](#). (Repositories Ranking Gitstar, sense data) És el rei en aquest tipus de tecnologies, tot i que podria canviar en poc temps.
- Flutter: aquesta plataforma porta dos anys en el mercat i amb aquest poc temps ja ha igualat (o potser superat quan s'estigui llegint això) la quota de mercat de React Native. A principis de 2020, amb només un any al mercat, ja era [el tercer framework més agradat pels usuaris de Stackoverflow](#). (2020 Developer Survey StackOverflow, sense data) Una tecnologia que al igual que Android, pertany a Google. Utilitza el seu propi llenguatge, Dart, que s'especialitza en el multiplataforma. Té una bona documentació, tot i que hi ha parts del framework que no estan complertes i encara tenen errors. És una plataforma molt nova, tot i que es millora dia a dia, es una tecnologia més a arriscada d'utilitzar. El seu suport per Android i iOS és estable, però [el suport web segueix en beta](#). (Web support for Flutter, sense data)

Donat que React Native i Flutter tenen les bases d'usuaris més grans, molt bones opinions i dues de les [Big Tech<sup>7</sup>](#), (Tech, sense data) semblen les dues millors apostes pel futur.

---

<sup>6</sup> Github és un servei de Hosting de repositoris on emmagatzemar codi, compartir-lo i administrar-lo.

<sup>7</sup> Grup de les 5 companyes de l'informació dominants: Google, Facebook, Apple, Amazon i Microsoft.

## 5.2 Tecnologies a utilitzar

### 5.2.1 Flutter

He escollit Flutter ja que utilitza Android Studio pel desenvolupament, al igual que Kotlin i té molt bona integració amb Firebase. Android Studio és el IDE <sup>8</sup>utilitzat per desenvolupar aplicacions per dispositius mòbils d'Android, i té ple suport pel que es necessita per desenvolupar amb Flutter. Que són tecnologies que vaig poder utilitzar prèviament. A més, Android Studio té una integració molt bona amb GitHub, cosa que permetria fer un sistema de control de les versions de l'aplicació de una forma més fluïda.

Flutter en sí és un SDK <sup>9</sup>que permet escriure tot el codi amb el seu llenguatge, Dart i crear instal·ladors per Android i iOS. Funciona d'una manera semblant per la web, creant un únic fitxer font a partir d'una combinació de DOM<sup>10</sup>, Canvas element i CSS<sup>11</sup>. Compila el codi Dart a JavaScript. El resultat és una pagina web accessible des de navegadors i dispositius mòbils. (Flutter Web, sense data)

Els elements visuals, estructurals i interactius de les interfície d'usuari es basen en widgets. Cada element en pantalla és un widget i es construeixen arbres de widgets, és a dir, que hi ha widgets dins d'altres widgets que emmagatzemen altres widgets.

Com ja s'ha comentat, Firebase i Flutter tenen una bona relació mitjançant el plugin oficial FlutterFire, que facilita les crides al servidor de Firebase. D'aquesta forma, Firebase emmagatzema les dades de l'aplicació, i des de Flutter es fan una crida que retorna la informació i s'actualitzen els widgets.

La utilització de les dades al núvol de Firebase també és una bona oportunitat per aplicar Programació Reactiva. [La programació reactiva es un paradigma enfocat en treballar amb fluxos de dades asíncrons](#) (Didierboelens: Introduction to the notions of Streams, Bloc and Reactive Programming. Theory and practical examples., sense data). Hi ha diferents models de programació reactiva, però Flutter utilitza BLoC.

### 5.2.2 Arquitectura: BLoC

La programació reactiva és un paradigma que s'enfoca en fluxos de dades asíncrons, permetent que les dades es propaguin generant canvis en l'aplicació. Es a dir, els elements de l'aplicació reaccionen a les dades, executant una sèrie d'esdeveniments. El patró BLoC<sup>12</sup> (Business Logic Component) és la forma d'implementar programació reactiva a Flutter..

BLoC és una arquitectura del software amb 3 parts que ajuda a separar la lògia de negoci de la interfície d'usuari.

---

<sup>8</sup> Integrated Evelopment Environment és una eina de desenvolupament de software, que el facilita a traves d'una interfície gràfica i altres eines.

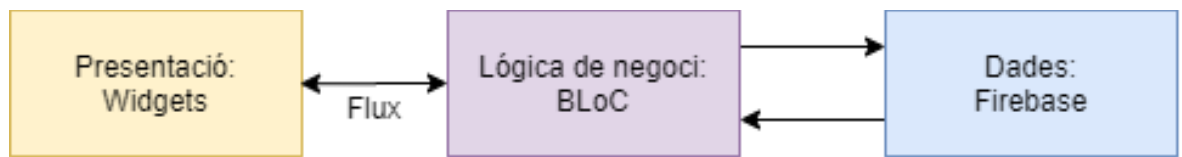
<sup>9</sup> Software Development Kit és una col·lecció d'eines que faciliten el desenvolupament d'aplicacions amb la inclusió d'un compilador i eines de testatge.

<sup>10</sup> DOM = Document Object Model. Llenguatge de programació que descriu l'estructura d'un document.

<sup>11</sup> CSS = Cascading Style Sheets. Llenguatge de programació que descriu la presentació d'un document.

<sup>12</sup> BLoC = Business Logic Component

El resultat d'utilitzar BLoC és la següent arquitectura basada en 3 parts:



**Figura 23.** Diagrama de l'arquitectura BLoC

BLoC gestiona la lògica del negoci, sent un mediador entre les dades que arriben al flux des de un repositori local o remot (en el nostre cas, Firebase) i la informació que es mostra als widgets (Medium: Flutter State Management: BLoC Pattern, sense data). El patró BLoC permet que un widget escolti al flux de dades o una seqüència d'esdeveniments, ja que està subscript a aquest flux. Cada esdeveniment es una dada i es transfereixen totes les disponibles si el widget està subscript. Després, aquest widget conté widgets fill que són capaços d'accedir-hi al flux, i que finalment s'actualitzaran en base a aquestes dades.

Per implementar correctament BLoC, la lògica de negoci ha de:

- Ser independent de la presentació tant com es pugui. En altre paraules, els components de la IU només haurien de preocupar-se per les coses de la IU i no pel negoci.
- Tenir Streams exclusius per l'input i l'output.
- Ser independent de la plataforma

Això permet canviar la lògica de negoci sense molt d'impacte sobre la presentació, i al revés.

### 5.2.3 Serveis de Firebase

Firebase ofereix multitud de serveis (o productes, com diuen ells). Per desenvolupar l'aplicació es farà us de molts d'aquests. Tots són gratis fins un nombre de sol·licituds màxim, a partir dels quals es comença a pagar segons l'ús. [Els preus es poden consultar a la pagina web](#) (Firebase Pricing plans, sense data). Té uns estàndards de seguretat que van en línia amb els GDPR que aquest projecte busca.

#### 5.2.3.1 Cloud Firestore

Firebase Firestore és un dels dos tipus de bases de dades que ofereix Firebase. És flexible i escalable, pensat per l'emmagatzematge de dades en dispositius mòbils. Treballa amb col·leccions de documents, semblant a JSON<sup>13</sup>, però sense ser-ho. Està pensada per emmagatzemar tot tipus i estructures de dades, fins les jerarquies més complexes.

Com he dit, hi ha dos tipus de bases de dades, l'altre és Realtime Database. Es tracta de la base de dades original de Firebase. [Utilitza JSON i és la més simple d'utilitzar](#) (Choose a Database: Cloud Firestore or Realtime Database, sense data). Realtime Database també està dedicada per operacions més simples, sense ordenació ni transaccions avançades.

---

<sup>13</sup> JSON = JavaScript Object Notation. Es tracta de un format de representació d'informació que per la seva simplicitat és fàcil d'entendre i escriure tant per humans com per ordinadors.

Funci3n de la base de datos	Si no necesitas realizar consultas, transacciones ni ordenamientos avanzados, te recomendamos <a href="#">Realtime Database</a> .
Operaciones con datos	Si tu app envia un flujo de pequefias actualizaciones, como en una app de pizarra digital, te recomendamos utilizar <a href="#">Realtime Database</a> .
Modelo de datos	Para datos JSON no estructurados, recomendamos <a href="#">Realtime Database</a> .
Disponibilidad	Cuando se acepta una disponibilidad muy alta, pero que no llega a ser crtica, recomendamos <a href="#">Cloud Firestore</a> o <a href="#">Realtime Database</a> .
Consultas sin conexi3n en datos locales	Para capacidades de consulta sofisticadas en datos locales cuando el usuario no tiene conexi3n, recomendamos <a href="#">Cloud Firestore</a> .
Cantidad de instancias de base de datos	Si necesitas una sola base de datos, te recomendamos <a href="#">Cloud Firestore</a> o <a href="#">Realtime Database</a> .

**Figura 24.** Elecci3n entre Cloud Firestore i Realtime Database

En un principi, Realtime Database semblava la m3s adient per aquest projecte, donat que les dades a guardar no s3n massa complexes. L'3nic problema 3s que Flutter encara no es compatible en totes les plataformes amb Firebase. Igual que la versi3n de web encara estava en beta (cosa que va canviar amb l'arribada de Flutter 2), [Realtime Database encara no funciona en web](#) (FlutterFire, sense data). Realtime Database funciona perfectament per Android i iOS, per3 s'estaria deixant de banda la versi3n web, al menys, fins que els desenvolupadors implementessin la funcionalitat.

Per aquesta ra3, s'utilitza Cloud Firestore, que 3s compatible amb totes les plataformes.

#### 5.2.3.2 Firebase Storage

Aquest 3s el servei de emmagatzematge d'objectes de Firebase. Amb aquest, es poden emmagatzemar i descarregar tot tipus de contingut multim3dia.

3s ideal pel nostre prop3sit: imatges.

#### 5.2.3.3 Firebase Authentication

El servei d'autenticaci3n de Firebase emmagatzema les credencials dels usuaris de forma segura i t3 multitud de sistemes per registrar-se i iniciar sessi3n (tel3fon, contrasenya, correu electr3nic, Google, Facebook...). Tamb3 t3 sistemes de recuperaci3n de contrasenya, com per exemple, enviar un correu per canviar-la.

L'administrador de l'aplicaci3n del projecte podria iniciar sessi3n amb les credencials emmagatzemades a Firebase Authentication, 3s a dir, correu electr3nic i contrasenya.

#### 5.2.3.4 Firebase Messaging

Un dels requisits de la versi3n 3.0 implementa les notificacions push. Aquest servei aporta una infraestructura per l'enviament de notificacions d'aquest tipus en aplicacions m3bils. No funciona a la web, per3 tampoc 3s un problema, ja que l'objectiu es poder enviar notificacions a les aplicacions en Android i iOS,

### 5.2.3.5 Firebase Hosting

A través de Firebase també es pot hostejar una aplicació web. De tots els serveis analitzats, aquest és l'únic que no és recomanable de cara a una versió final, donat que l'opció gratuïta només arriba a 360 MB/dia, cosa que en arribar a cert nombre de sol·licituds, és fàcil de sobrepassar.

Això sí, és un sistema molt recomanable per fer versions de prova, compartir-les i obtenir feedback.

### 5.2.3.6 Risc: massa dependència amb Firebase

Firebase és un servei pensat en el futur, que posa les coses fàcils als programadors i ofereixen una qualitat excel·lent, inclús a l'usuari que no paga. Actualment, les quotes són molt raonables i, exceptuant Hosting, és difícil arribar al sostre de la quota gratuïta.

El problema vindria si aquestes quotes canvien en un futur, si el que abans era gratuït, ara no ho és. O si els serveis actuals passessin a no existir. Firebase [ja ha fet algun canvi als costos prèviament](#) (StartUpsVentureCapital: Firebase Costs Increased by 7,000%! , sense data) i per donar un altre exemple, Google Photos (que abans era totalment gratuït), [tindrà més limitacions en el pla gratuït](#) (amid, sense data). Google no té por a fer aquests canvis i és molt possible que hi hagi més en el futur.

Crec que la dependència amb aquest servei és una debilitat, que ve compensada amb les facilitats donades. Per aquesta raó, la utilització de Firebase està feta amb consciència dels seus possibles problemes.

## 5.2.4 Control de versions: GitHub

GitHub és un proveïdor de serveis i eines per al desenvolupament de software. S'ha utilitzat GitHub per fer un control de les versions de l'aplicació. D'aquesta forma, no es pot perdre el codi, ja que està publicat al nuvol.

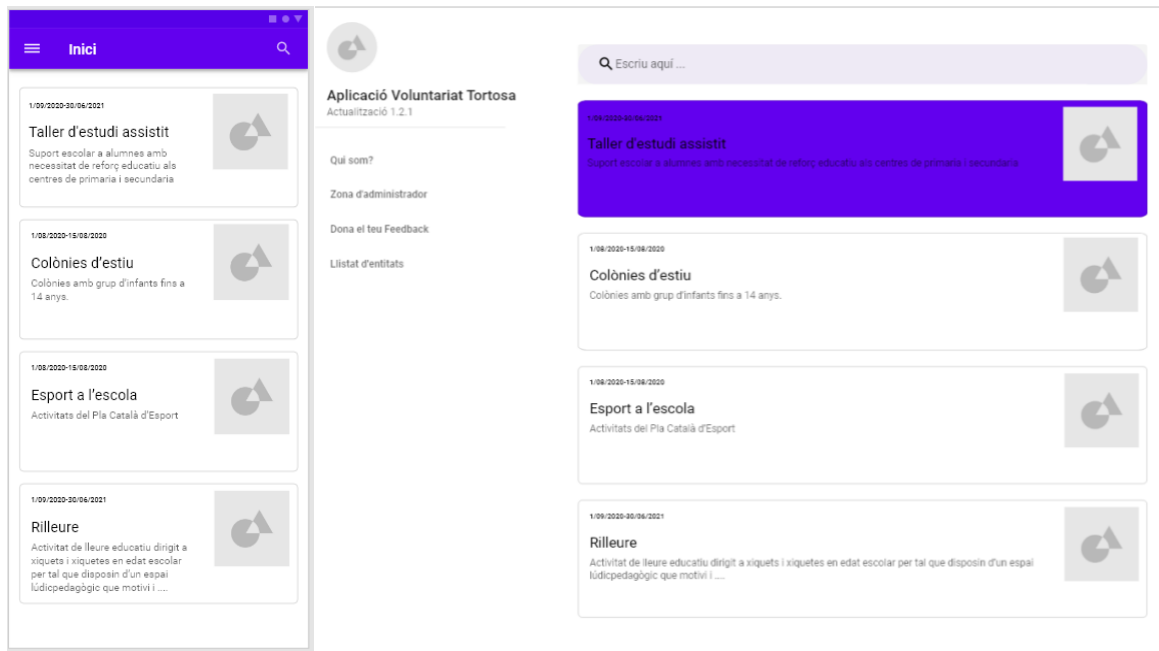
També s'ha tret un profit d'altres funcionalitats de la plataforma. Com que el codi de Flutter es desenvolupa a través de GitHub, es pot veure de forma transparent en que estan treballant els programadors de Flutter. D'aquesta forma, es pot veure quins problemes venen per part de la pròpia plataforma.

A més, GitHub té la funcionalitat de crear projectes, organitzar tasques i problemes en el temps. D'aquesta forma sempre es té un control del que falta a fer.

## 5.3 Prototips de la interfície gràfica: Versió 1

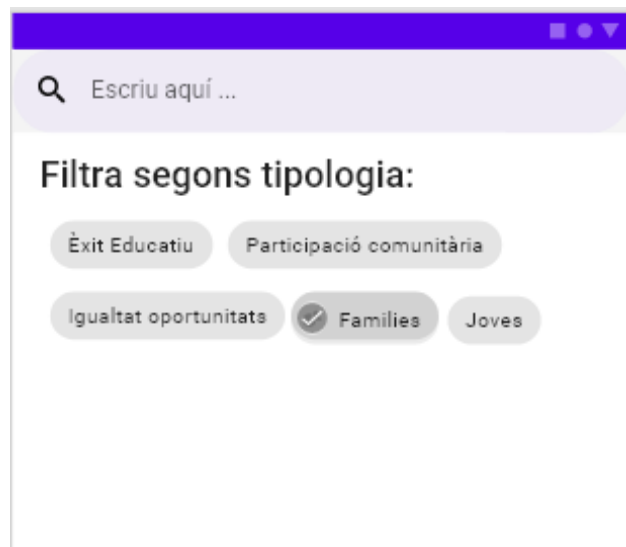
Per poder crear uns prototips sense interacció i que tinguin un aspecte semblant al resultat final, s'ha utilitzat Adobe XD. És una eina especialitzada en fer prototips per diferents plataformes. En el cas de Flutter, té fins i tot una eina per generar codi automàticament a partir del prototip (tot i que no s'ha utilitzat en aquest projecte).

- A l'esquerra la versió d'Android, a la dreta la d'escriptori. A dalt a l'esquerra està el menú i a dalt a la dreta el cercador d'activitats. Al mig es troba la llista d'activitats, clicant una es passa a la informació detallada d'aquesta.



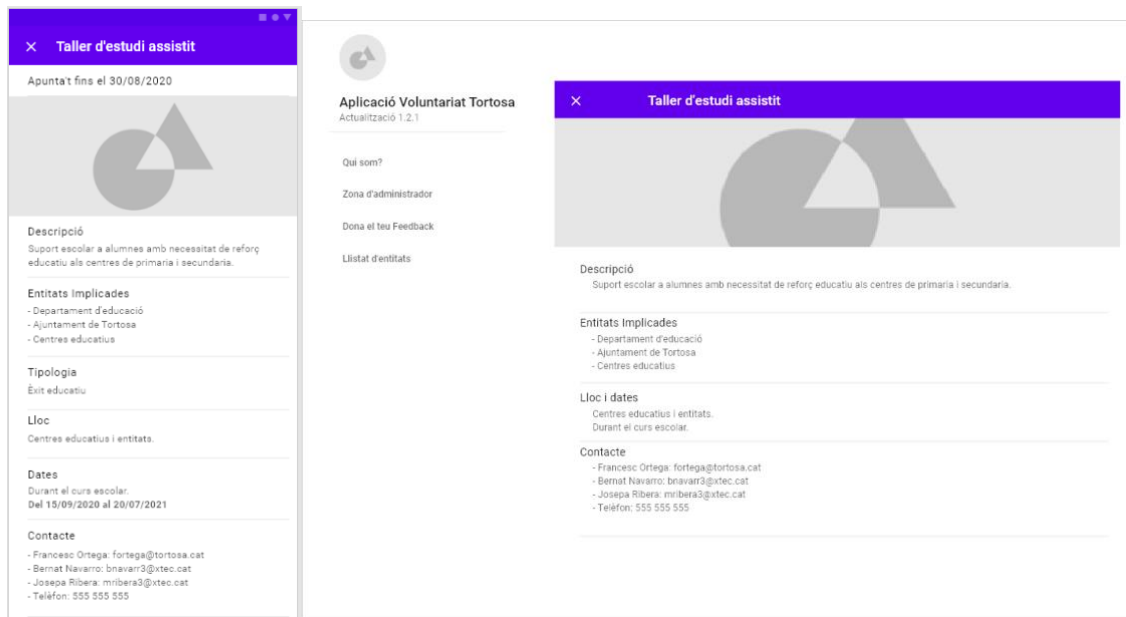
**Figura 25.** Prototip Llista d'activitats

- Es podrà introduir un o més termes per buscar si coincideixen amb algun camp de les activitats. També està el filtre segons la tipologia.



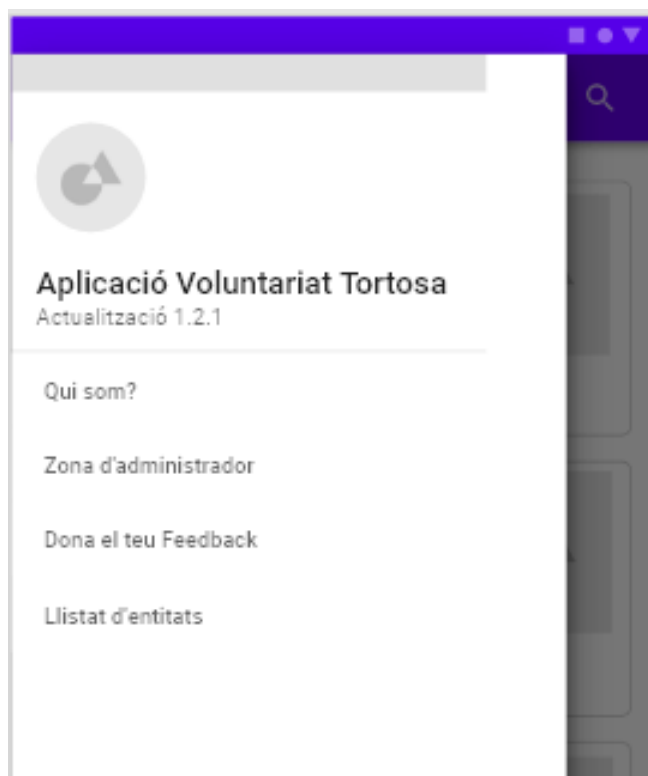
**Figura 26.** Prototip Cercador i Filtre

- Prototip Detalls Activitat Es pot veure una imatge a la part superior dels detalls i un llistat amb els atributs de l'activitat. Pulsar la creueta porta al llistat anterior. S'ha d'aclarir que els atributs que es veuen al prototips no tenen perque ser els finals.



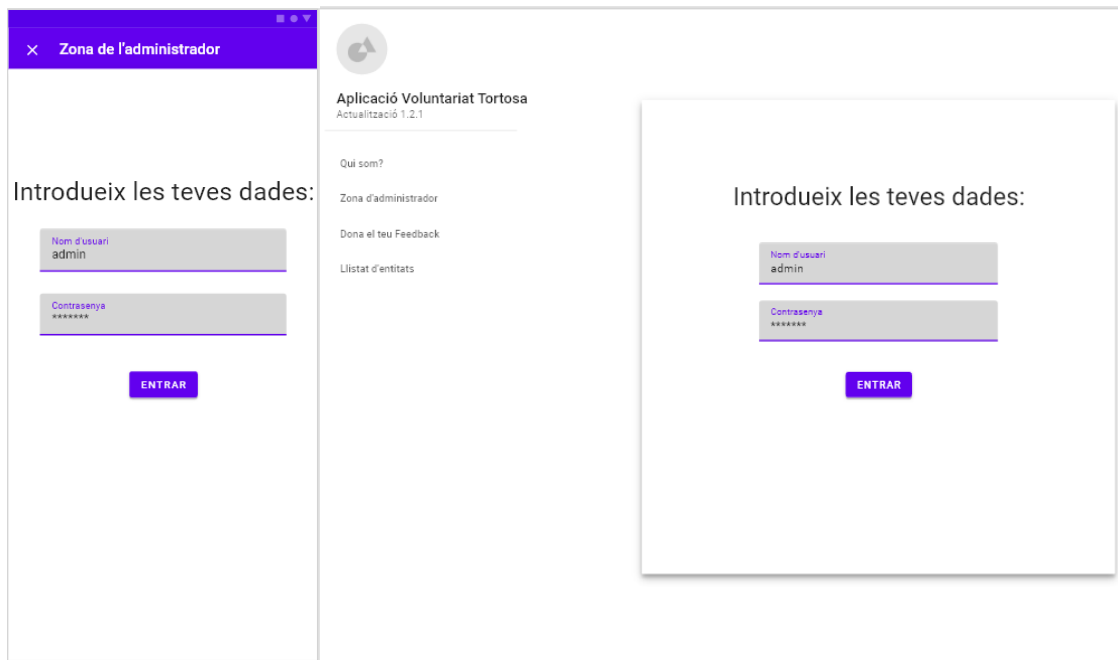
**Figura 27.** Prototip Detalls d'Activitat

- Per poder accedir a la zona de l'administrador, accedir a la llista completa d'entitats o veure informació general de la plataforma o enviar feedback.



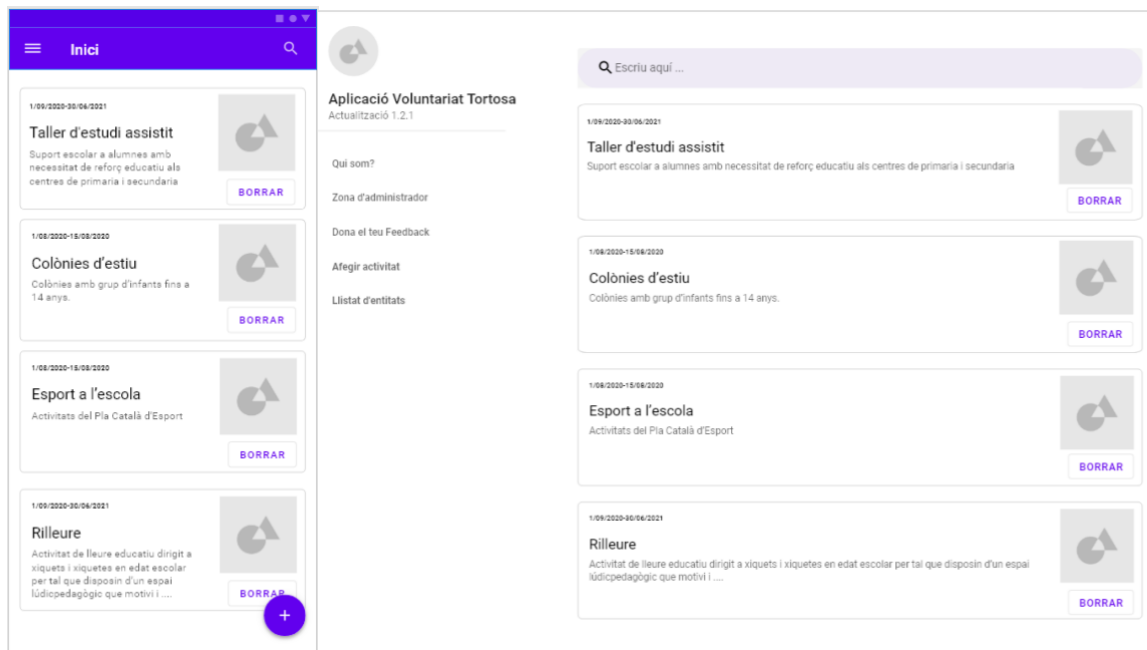
**Figura 28.** Prototip Menú

- L'administrador pot introduir les dades d'autenticació als camps de text i enviar la informació al servidor per ser analitzada.



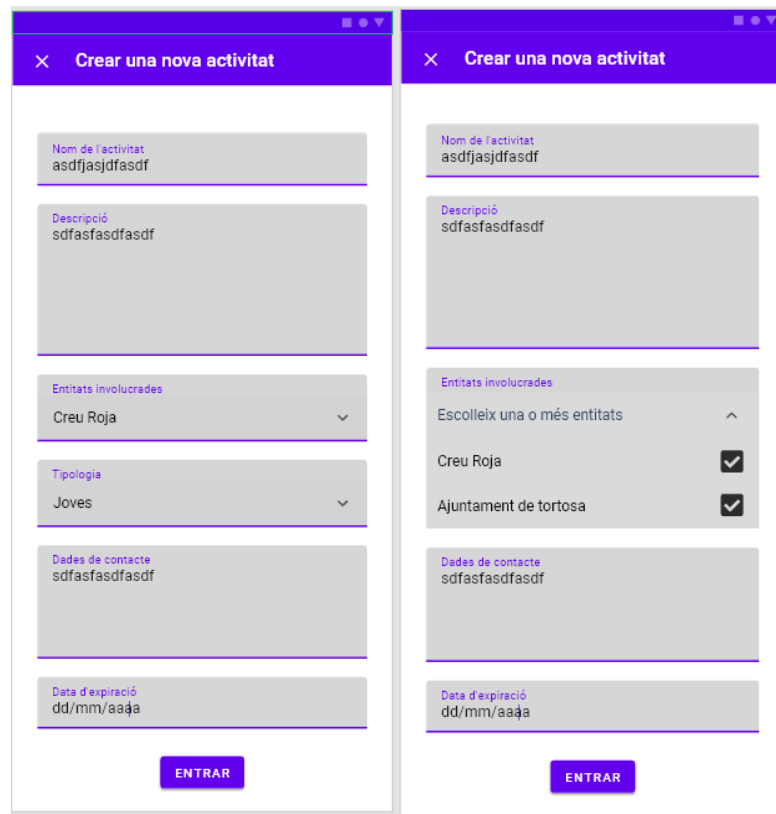
**Figura 29.** Prototip Identificació Administrador

- Es molt semblant a la pantalla principal de l'usuari, però s'afegeixen les opcions d'afegir i esborrar activitats.



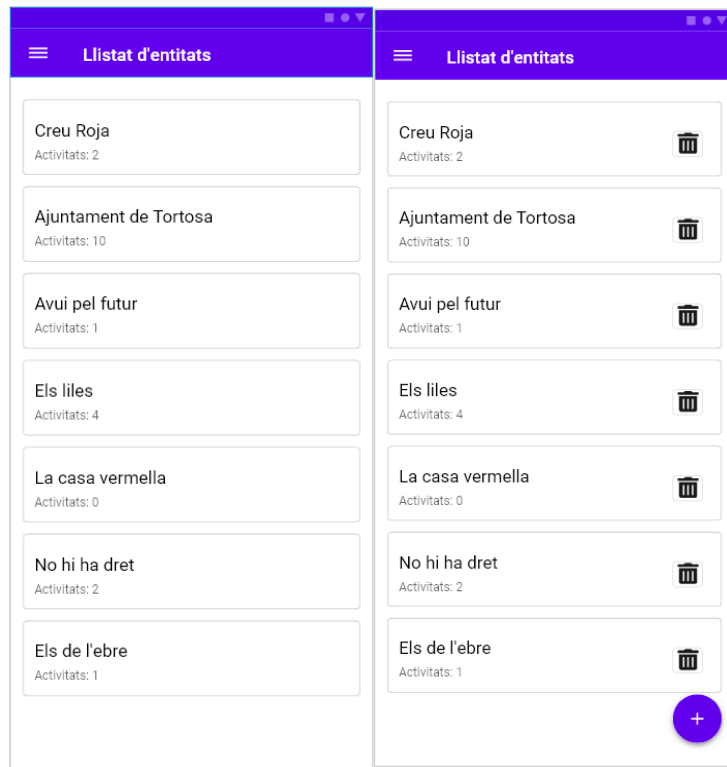
**Figura 30.** Prototip Llista d'activitats d'administrador

- A l'exemple no estan tots els camps que s'introduiran, a la pràctica hi haurà totes les activitats. Les entitats tindran uns checkboxes per seleccionar quines són les involucrades.



**Figura 31.** Prototip Crear Activitat

- A l'esquerra la part de l'usuari normal; a la dreta la de l'administrador, amb els botnos d'afegir i esborrar entitats. Per afegir una entitat, només es preguntarà el nom.

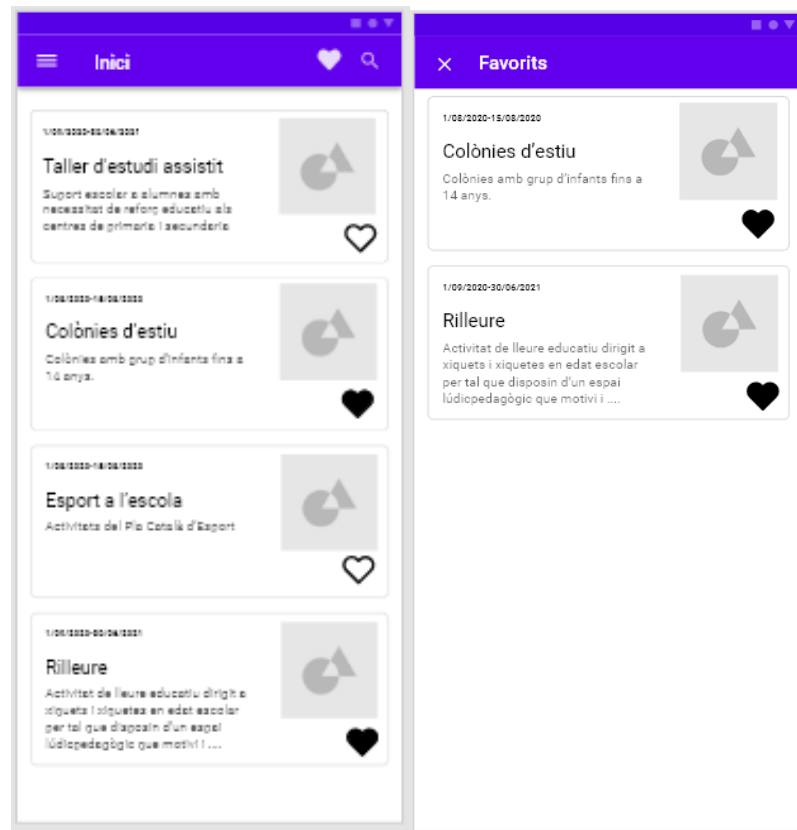


**Figura 32.** Prototip Llista d'Entitats

#### 5.4 Prototips de la interfície gràfica: Versió 2

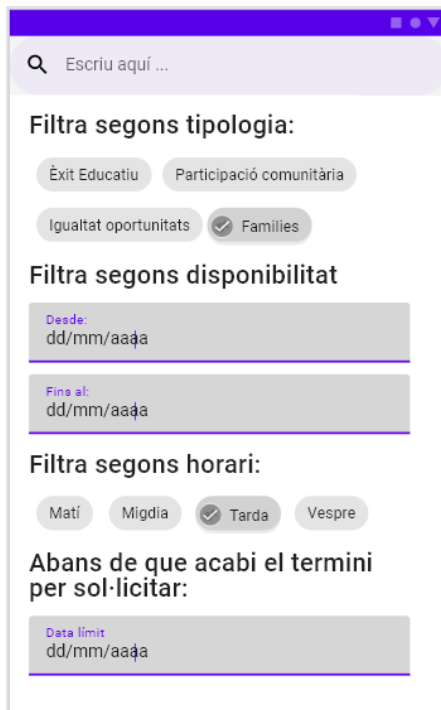
Aquests són els prototips fets a partir de les funcionalitats que inclouria la versió 2. Quan les funcionalitats de la versió 1 estiguessin implementades, aquests serien els prototips d'algunes de les funcionalitats a implementar a continuació. En aquest apartat s'inclouen prototips de les millores a fer per sobre de la versió 1, i no suposa una substitució de les funcionalitats ja implementades a la versió anterior.

A l'esquerra es pot veure que s'inclouen a la pantalla principal uns cors que representen si un usuari té (o no té) una activitat afegida a favorits. També es pot veure a la dreta que es pot accedir a una llista d'activitats favorites.



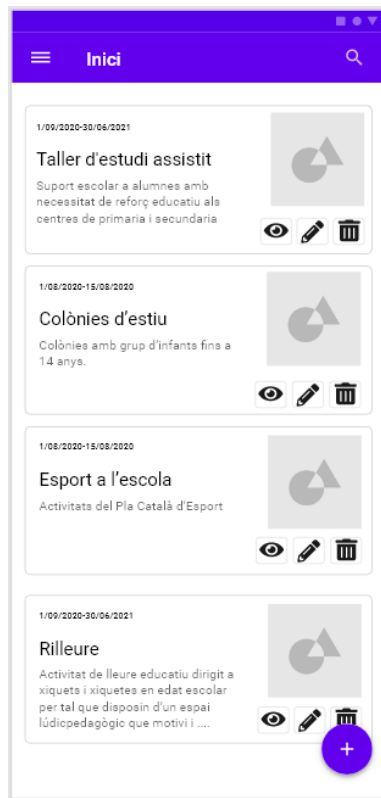
**Figura 33.** Prototip activitats favorites

A continuació es poden veure els filtres avançats, segons una multitud d'atributs.



**Figura 34.** Prototip filtres avançats

L'administrador podrà ocultar (ull), editar (llapis) o esborrar (brossa) una activitat.



**Figura 35.** Prototip amb més opcions per a l'administrador

Les activitats destacades apareixen les primeres a dalt i en vermell.



**Figura 36.** Prototip d'activitats destacades

Visualització de les activitats organitzades per una entitat en concret.



**Figura 37.** Prototip Llista d'activitats d'una entitat

## 6 Desenvolupament

A continuació es detallen cadascuna de les iteracions que s'han fet a través de la metodologia Agile.

### 6.1 Iteració 1: Experimentació

*Duració: 21 de Octubre al 4 de Novembre de 2020*

Aquesta iteració té com a objectiu aprendre a els conceptes basics de les tecnologies a utilitzar. Aquesta primera etapa està marcada pels tutorials i guies per aprendre a utilitzar Flutter i el seu llenguatge (Dart) mentre es fa la pròpia aplicació en comptes d'aprendre primer i desenvolupar després. D'aquesta forma es pot agilitzar el desenvolupament i em podria enfocar en aquelles funcionalitats que tindrien la versió final.

Vaig seguir [un tutorial](#) (Youtube: Flutter Tutorial for Beginners by The Net Ninja, sense data) que em va engegar en les possibilitats que tenia Flutter, i el seu IDE<sup>14</sup>, Android Studio. El vídeo (i codi) està fet per un dels canals més importants de tutorials en noves tecnologies software, [The Net Ninja](#) (Youtube: The Net Ninja, sense data). A part d'ensenyar a configurar el Android Studio per desenvolupar i algunes de les operacions més simple de Dart, també instrueix sobre els widgets més simples. Aquests inclouen llistes, colors, botons, icones, imatges, caselles de text... Una de les qüestions més interessants que vaig aprendre va ser que existeixen dos tipus de widgets:

- **Stateful widgets:** el estat del widget pot canviar en temps d'execució. Es poden canviar tots els elements (widgets) que hi ha dins del widget sense problema.
- **Stateless widgets:** el estat del widget no canvia mentre s'executa el propi widget. És a dir, es podria interactuar amb el widget (i els widgets que hi ha dins d'aquest), però no es pot modificar cap text, color, icona, botó... en temps d'execució. En altres paraules, es pot prémer un botó que activa una funcionalitat o pantalla, però no es pot canviar el color del botó en temps d'execució.

A la [documentació de Flutter](#) (Flutter for Android developers, sense data), es parla de que els dos widgets es comporten de la mateixa forma en el seu nucli. La única diferencia es que Stateful widgets tenen un estat que guarda una informació que pot ser potencialment canviada en temps d'execució. La regla aquí és: si el widget canvia (quan l'usuari interactua amb ell, per exemple), es Stateful. Però, **si el fill d'un widget reacciona a un canvi, el widget pare pot seguir sent Stateless mentre que el pare no reaccioni a un canvi**. Això implica que podrien haver-hi tot tipus de widgets fill dins d'un pare.

Stateful widgets tenen un avantatge a l'hora de desenvolupar, i es que permet fer recarregues en calent (hot-reload). Això significa que si estem executant el programa i modifiquem el codi de un stateful widget, els canvis es poden aplicar i veure sense reiniciar l'aplicació. Quan s'invoca un hot-reload, es recompila la llibreria amb el codi canviat i es enviat a la Dart Virtual Machine. Aquesta actualitza les classes i reconstrueix el widget i tots els seus fills. D'aquesta forma, els canvis es veuen casi a l'instant, molt més ràpid que tornar a compilar tota l'aplicació i reiniciar-la.

---

<sup>14</sup> IDE = Integrated Development Environment

Anteriorment, parlàvem de com [Realtime Database encara no funciona en web \(FlutterFire, sense data\)](#), cosa que en aquestes primeres fases no es sabia. Aquest principi es va realitzar seguint un [tutorial](#) (Youtube: Flutter and Firebase by Santos Enoque, sense data) per saber fer una aplicació simple de Flutter utilitzant operacions CRUD<sup>15</sup> amb Realtime Database. Posteriorment, era necessari fer una migració d'un sistema a l'altre, canvi que no va ser molt dramàtic.

A partir d'aquí, es van seguir tutorials per seguir aprenent sobre Flutter de la mà de Firebase Firestore. Ja que el primer que vaig seguir, de The Net Ninja, va ser molt útil, vaig seguir [un segon tutorial sobre precisament el que estava buscant](#) (Youtube: Flutter & Firebase App Build by The Net Ninja, sense data). També vaig seguir [un altre tutorial](#) (Youtube Flutter Blog App Using Firestore - Build ios & Android App by Master Coding, sense data) d'un altre canal amb bona reputació sobre desenvolupament.

## 6.2 Iteració 2: Estructura i funcionalitats base de l'aplicació

Duració: 4 de Novembre al 18 de Novembre de 2020

Per facilitar el desenvolupament, es va començar a programant partint l'aplicació en 2: una part per les funcionalitats de l'usuari i un altre per les funcionalitats de l'administrador. Això implicaria tindre codi repetit, però facilitaria tindre aquestes dues parts separades i independents mentre no es sapigués com manegar aquest codi comú. Tindre codi en comú pot suposar un risc de seguretat, ja que si no es fes de forma correcte, un usuari podria accedir a la part de l'administrador. Aquesta forma d'estructurar el codi canviaria en quant es trobés una forma de solucionar aquest possible problema.

Es va realitzar un primer prototip a presentar a l'ajuntament amb esquelet/esquema del que podria fer l'aplicació. Aquest prototip incloïa les següents funcionalitats:

- Llistat d'activitats amb cadascuna el seu títol i part de la descripció. Les activitats destacades apareixen a la part superior de la llista i envoltades de vermell.
- Barra superior amb un [Drawer](#) (Navigation Drawer Material Design, sense data) que permet accedir a la informació de l'aplicació (Qui som?) i la zona de l'administrador. Aquesta barra superior també inclou un botó per cercar.

El codi que hi ha darrere d'aquesta implementació comença amb un main.dart que, després de fer unes inicialitzacions, crida a la pagina principal (HomePage), que a l'hora està envoltada en una MaterialApp, requerida per crear una aplicació amb Material Design.

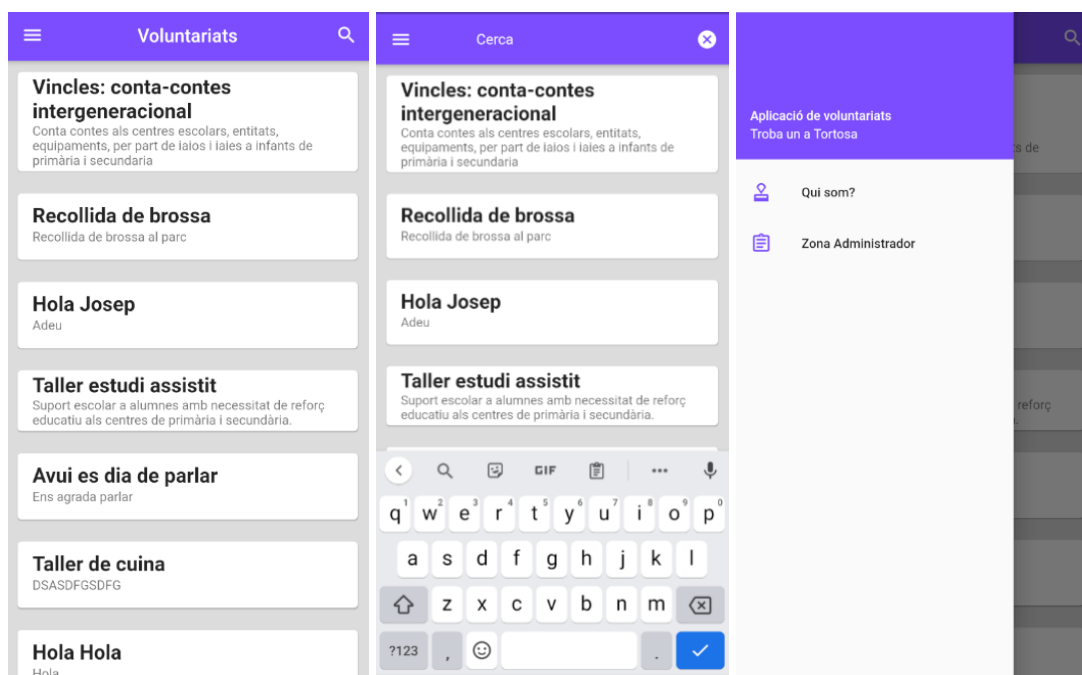
```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  Widget build(BuildContext context) {
    return MaterialApp(home: HomePage());
  }
}
```

---

<sup>15</sup> Conjunt d'operacions de creació (Create), lectura (Read), actualització (Update) i eliminació (Delete).

## Codi 1. Main

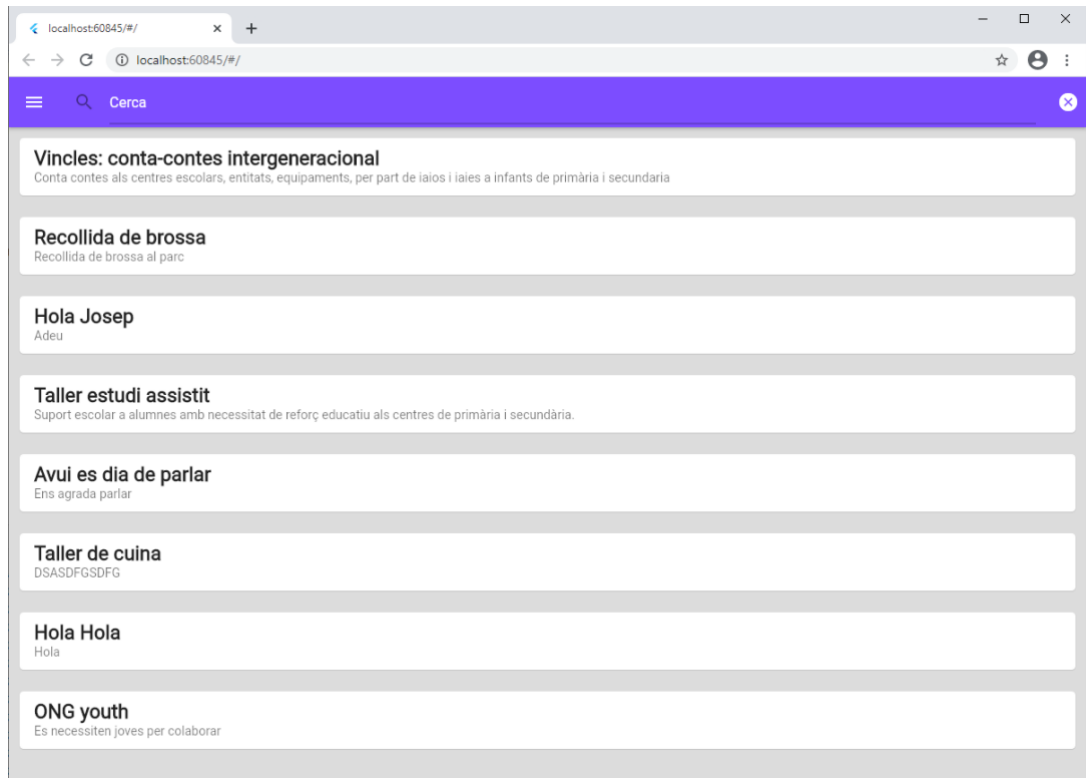
Aquesta pagina principal serà Stateful, ja que segurament serà una pantalla que canviï en quant se li apliquin filtres o cerques. El widget pare que s'ha utilitzat a la majoria de pantalles ha sigut Scaffold. Scaffold implementa la estructura basica de Material Design, diseny escollit per implementar a l'aplicació. Aquesta estructura inclou el Drawer, la barra superior o el cos de la pantalla, entre d'altres.



**Figura 38.** Prototip interactiu cerca i menú

També cal parlar de les dependències amb llibreries i widgets externs. Flutter incorpora unes funcionalitats i permet que altres desenvolupadors creïn els seus propis widgets i llibreries més o menys complexes. Per exemple, cadascun dels serveis de Flutter (firestore, authentication, storage...) té el seu propi paquet a Flutter ([cloud\\_firestore](#) (Pub.dev cloud\_firestore, sense data), [firebase\\_auth](#) (Pub.dev firebase\_auth, sense data), [firebase\\_storage](#) (Pub.dev firebase\_storage, sense data)...) i una versió d'aquest paquet. Totes aquestes llibreries amb les seves respectives versions es declaren a un fitxer anomenat pubspec.yaml.

Per últim, aquest prototip també funcionava en web, probat des de google chrome per Android i Windows 10. De moment el focus estaria a Android i Web, deixant de banda la versió d'iOS.



**Figura 39.** Prototip interactiu web

El desenvolupament del primer prototip va suposar la creació dels ciments al damunt d'on es construiria l'aplicació.

### 6.3 Iteració 3: Ampliacions de l'estructura

*Duració: 18 de Novembre a 2 de Desembre de 2020*

Donat que des de que es va crear Flutter, és un dels repositoris amb més participació i expectació a Github, la creació de un repositori per l'aplicació és el pas a fer. Flutter es un projecte en desenvolupament i si en algun moment tinc un problema amb un widget, podré fer referència al meu repositori dins d'un issue del repositori de Flutter. A més, al estar dins de la plataforma, és una oportunitat per seguir de prop els canvis que s'estan fent en el SDK<sup>16</sup>. Per això, tot el codi del projecte està escrit en anglès, per millorar la llegibilitat del codi per si el vull compartir amb un estranger (p.e. per solucionar un error). Si en qualsevol moment es perdés el meu repositori local, Github és una manera fàcil de recuperar el codi. També és una bona practica de cara a aquest treball, ja que es pot contrastar la informació en aquest document amb el que s'ha fet en cada moment en procés de desenvolupament.

Una vegada ja estava feta l'estructura de l'aplicació i l'arquitectura BLoC ja implementada, l'objectiu va ser incorporar la resta de funcionalitats una a una.

Primer es van implementar els detalls de l'activitat amb descripció, entitats, tipus, dates, lloc, horari i dades de contacte. Tota aquesta informació ja està a Firestore, dins de cada document de la col·lecció d'activitats. Des de la part de l'administrador, aquesta

---

<sup>16</sup> SDK = Software Development Kit. És un conjunt d'eines per desenvolupar aplicacions. Flutter és un SDK.

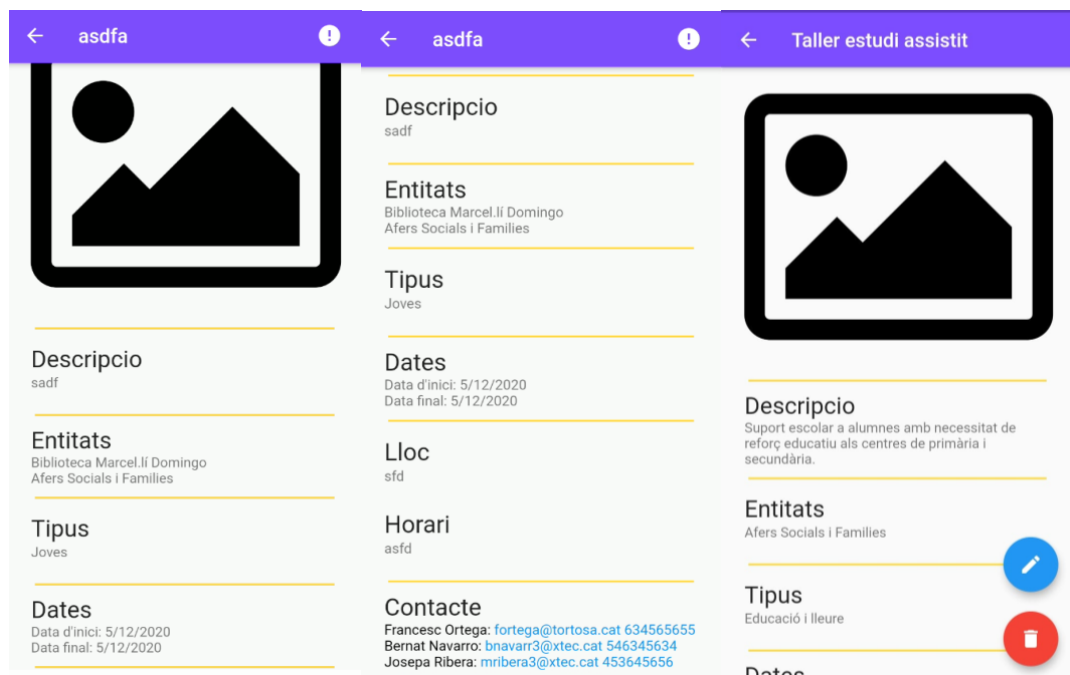
pantalla es veu amb dos botons inferiors per modificar la informació de l'activitat o esborrar-la de la base de dades, idea que vaig extreure [d'aquí](#) (Github Flutter-Tutorials by Cheetahtcoding, sense data). El codi per esborrar un document de Firestore es simple, em vaig inspirar en un tutorial i aquest és el codi.

```
CollectionReference ref =  
Firestore.instance.collection("Activities");  
deleteActivity(Activity activity) async {  
  await ref.doc(activity.id).delete();  
}
```

## Codi 2. Operació d'esborrat amb Firebase

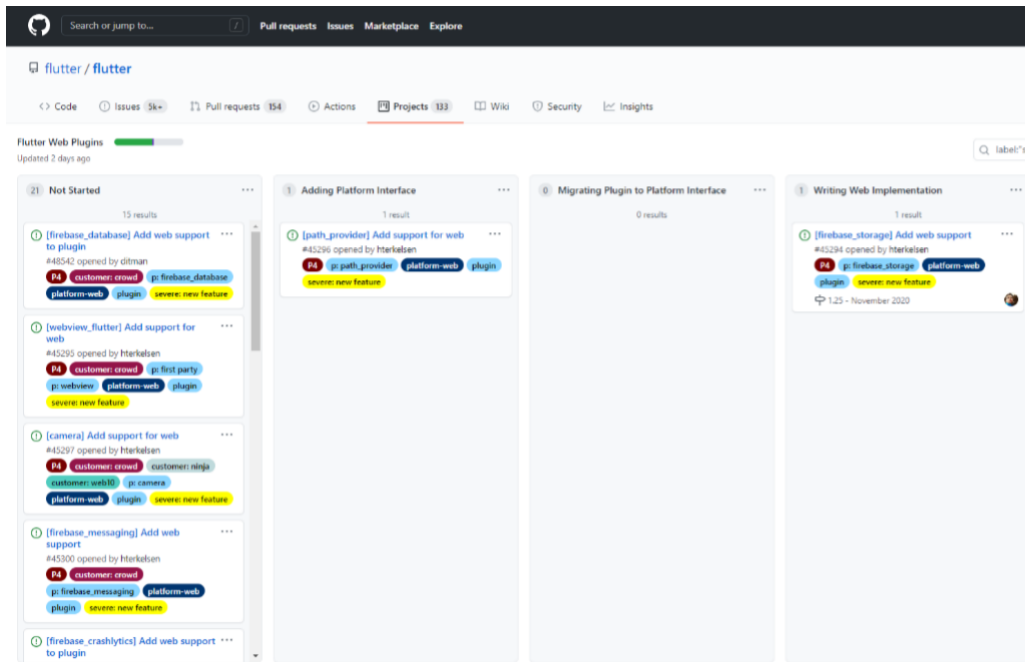
Es tracta d'una funció asíncrona, així el codi es principal es pot continuar executant sense que s'hagi d'esperar a que la funció s'executi al 100%.

S'ha utilitzat el widget extern [Linkable](#) (Pub.dev linkable, sense data) per poder fer que automàticament es reconeixin els correus i telèfons del contacte. Es pot apreciar que les dades contacte estan en blau, que significa que té interacció. Si es crica el numero de telèfon des del mòbil, s'obrirà l'aplicació per trucar i si es crica el correu, s'obrirà l'aplicació del correu.



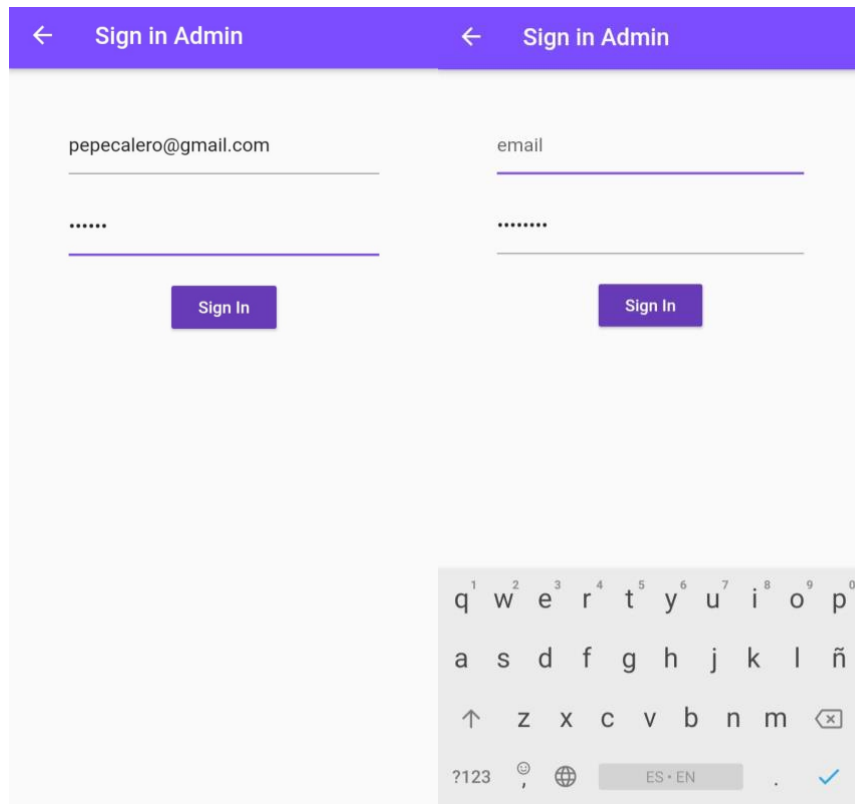
**Figura 40.** Prototip interactiu detalls d'activitat

La imatge dels detalls no estava implementada, donat que en aquell moment encara no estava implementat Firebase Storage per la web. Tot i així, el problema només era qüestió de paciència, ja que en [la fulla de ruta de Flutter](#), (Github FlutterFire Roadmap, sense data) es marcava el suport de Firebase Storage a web com a funcionalitat en la que s'estava treballant. Hi havia un conjunt de funcionalitats pendents de ser començades (p.e. suport web per Firebase Realtime Database) i altres com Path Provider o Firebase Storage en les que sí.



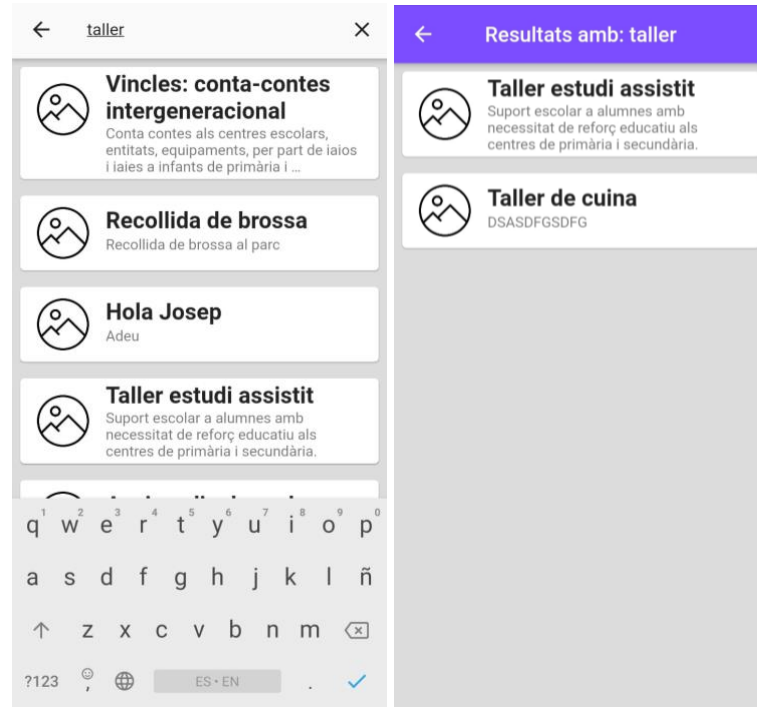
**Figura 41.** Full de ruta Flutter

Per accedir a la zona de l'administrador, es mostra una pantalla per introduir les credencials: correu electrònic i contrasenya. També m'he inspirat en [un tutorial de The Net Ninja](#). (Ninja, sense data)



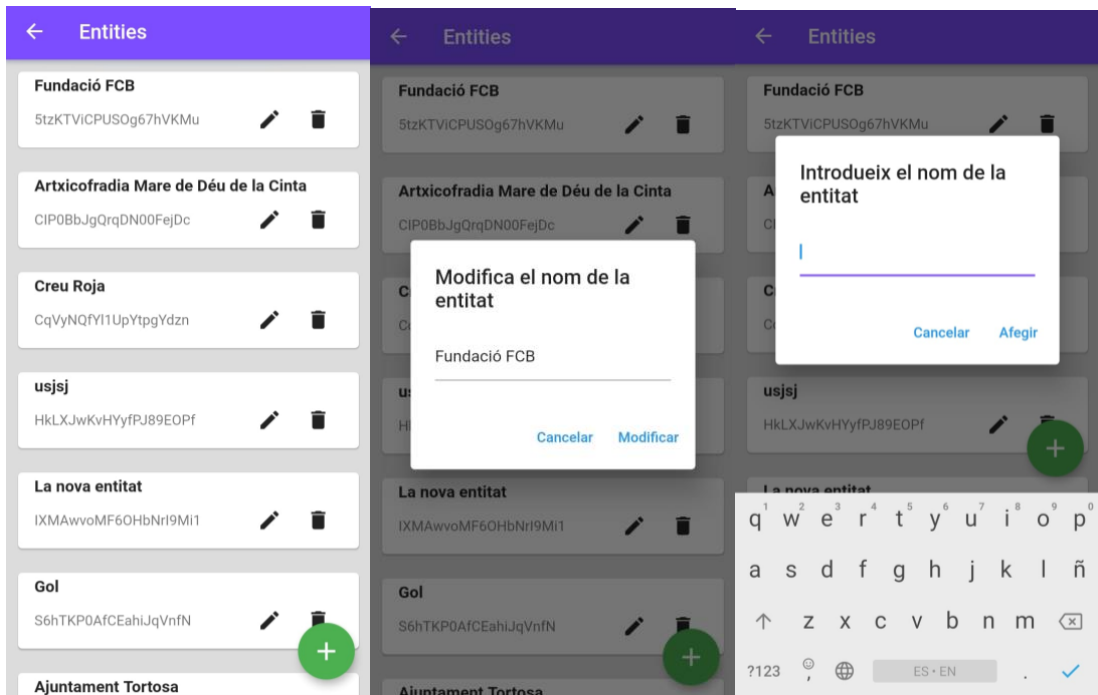
**Figura 42.** Prototip interactiu d'iniciar sessió

Anteriorment es tenia un botó de cerca, però sense funcionalitat. En aquest punt, es pot introduir una paraula/es i s'obre una nova pantalla amb un llistat d'activitats que coincideixen. Aquesta barra de cerca no està inclosa dins del codi Flutter i utilitza un paquet extern (widget), anomenat [flutter\\_search\\_bar](#) (Pub.dev flutter\_search\_bar, sense data), creat per un programador de la comunitat.



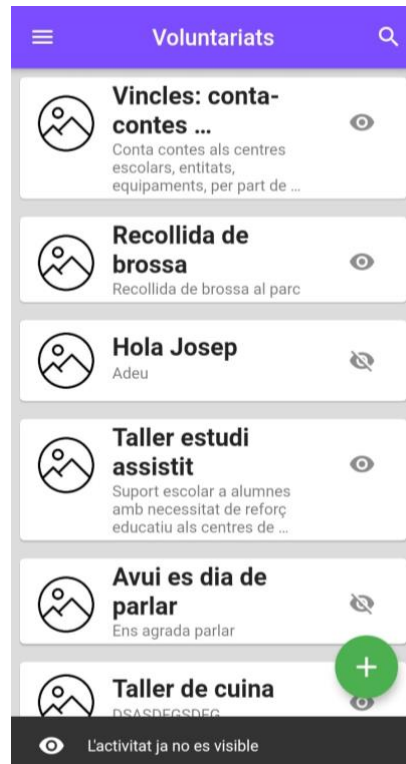
**Figura 43.** Prototip interactiu de cerca funcional

S'afegeix el llistat d'entitats, amb la possibilitat (des del punt de vista del administrador) d'afegir noves entitats a partir del seu nom, modificar el nom i esborrar-les. Els diàlegs que apareixen per modificar o introduir el nom de l'entitat s'han inspirat en [un tutorial de RetroPortal Studio](#). (Studio, sense data)



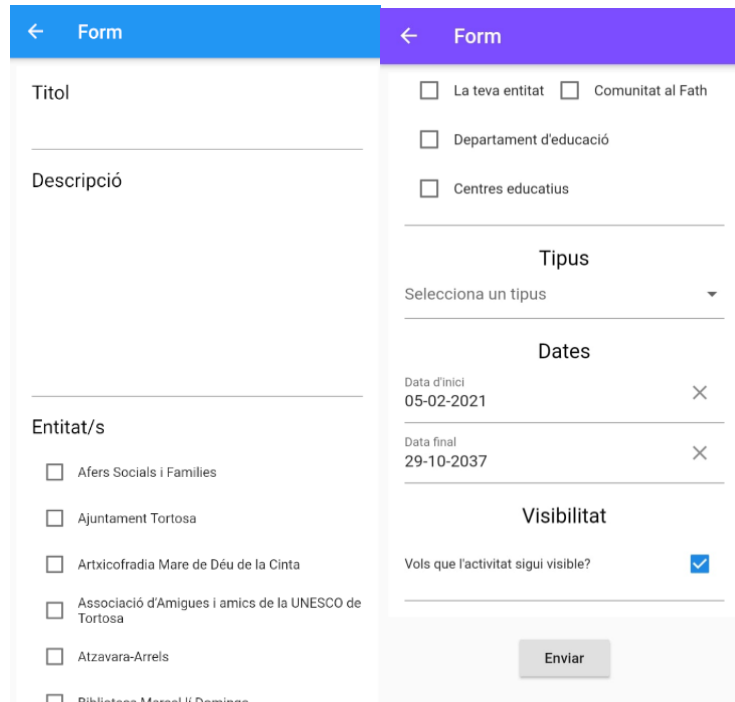
**Figura 44.** Prototip interactiu llista d'entitats i formulari

S'afegeix la possibilitat de ocultar activitats des de l'administrador. En cas que no es vulgui que una activitat sigui visible pels usuaris, es pot ocultar per més tard modificar-la. La informació sobre si una activitat es pot veure o no, està emmagatzemada a Firestore i es tracta d'un camp més de cada activitat.



**Figura 45.** Prototip interactiu visibilitat d'activitats

A la zona d'administrador s'afegeix un formulari per indicar tota la informació d'una activitat, de cara a afegir-la a la base de dades o modificar la informació de l'activitat per actualitzar-la. Aquest formulari utilitza un paquet extern creat per la comunitat anomenat [flutter\\_form\\_builder](#) (Pub.dev flutter\_form\_builder, sense data). Flutter té el seu propi Form, però s'ha escollit aquesta forma externa ja que incloïx més formes de validar els camps, que van més enllà de si un camp es buit o no. Per conèixer sobre les possibilitats de flutter\_form\_builder vaig seguir un [tutorial](#). (SyntacOps, sense data)

The image shows two side-by-side screenshots of a mobile application form titled "Form". The left screenshot shows the form's input fields: "Títol", "Descripció", and "Entitat/s". The "Entitat/s" field contains a list of checkboxes with labels: "Afers Socials i Famílies", "Ajuntament Tortosa", "Artxicrofradia Mare de Déu de la Cinta", "Associació d'Amigues i amics de la UNESCO de Tortosa", "Atzavara-Arrels", and "Biblioteques Marcel·lí Domínguez". The right screenshot shows the form's validation and submission options. It includes checkboxes for "La teva entitat" and "Comunitat al Fath", a dropdown menu for "Tipus" with the text "Selecciona un tipus", and date pickers for "Data d'inici" (05-02-2021) and "Data final" (29-10-2037). There is also a checkbox for "Visibilitat" with the text "Vols que l'activitat sigui visible?" which is checked. An "Enviar" button is at the bottom.

**Figura 46.** Prototip interactiu del formulari d'activitats

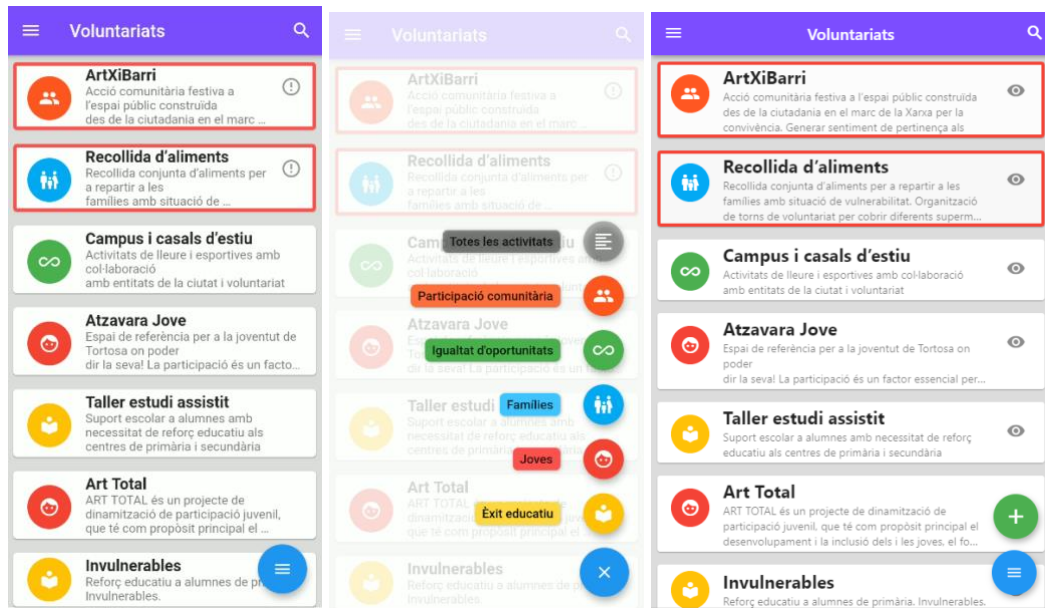
#### 6.4 Iteració 4: Millores visuals

*Duració: 2 de Desembre a 29 de Desembre de 2020*

En aquest punt, es va incorporar una nova eina de treball: Stackoverflow. Com que l'aplicació es fa cada vegada més complexa, en aquest apartat es veurà com d'util pot ser.

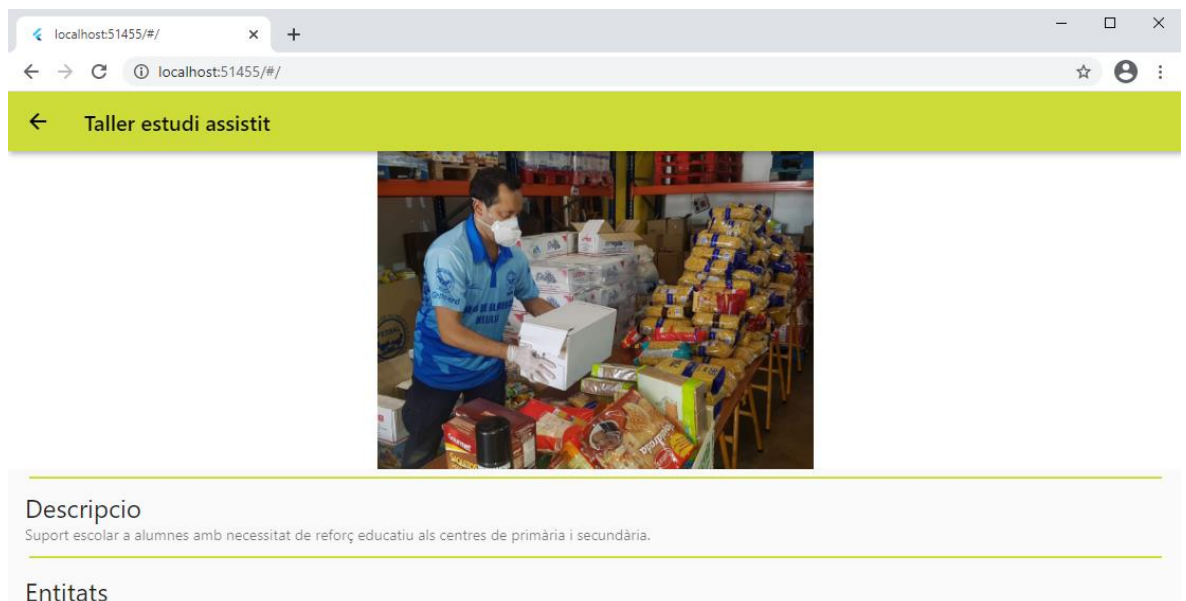
Es va afegir un filtre a la part inferior de la pantalla per poder escollir un tipus, cadascun representat per un color. El filtre està fet a partir d'un [Speed Dial](#) (Material Design Speed Dial Guidelines, sense data), que, té associat [un widget implementat per la comunitat](#) (Pub.dev flutter\_speed\_dial, sense data). També es destaquen algunes activitats, mitjançant un quadre vermell i una icona amb un signe d'exclamació (!). Les activitats destacades apareixen a la part superior de la pantalla. Posteriorment també s'afegiria una previsualització de la imatge de l'activitat al cercle de la targeta.

De cara a la part de l'administrador, es va afegir un botó al damunt l'altre, cosa que no va ser gens fàcil i ho vaig fer gràcies a [l'ajuda que vaig obtenir a través d'una pregunta a StackOverflow](#) (Calero, sense data).

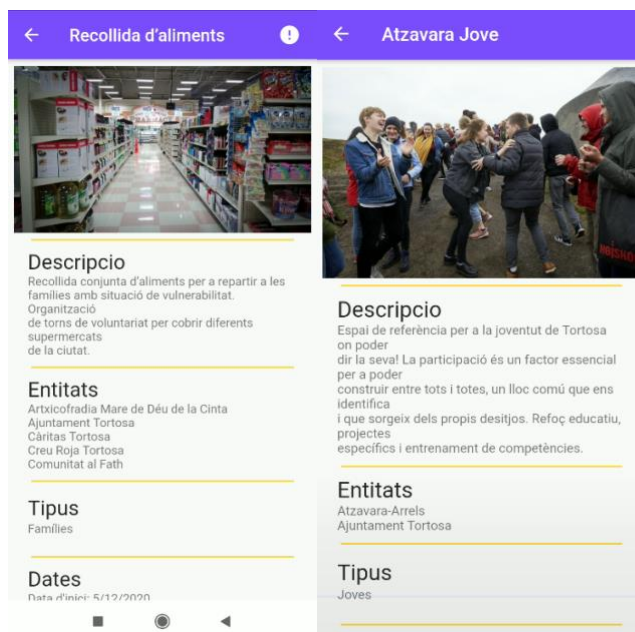


**Figura 47.** Prototip interactiu filtre de tipus

Finalment es va donar suport a Firebase Storage per Flutter web, que es va traduir en la possibilitat de veure imatges a la versió d'Android i web.



**Figura 48.** Prototip interactiu web

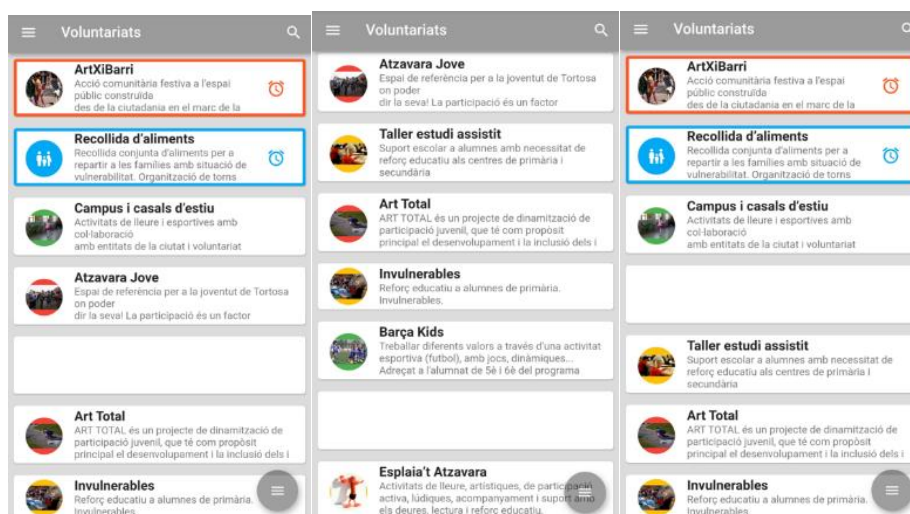


**Figura 49.** Prototip interactiu detalls d'activitat amb imatge

Una vegada es podien veure aquestes imatges a partir de links cap a imatges emmagatzemades a Firebase Storage, venia el problema de emmagatzemar aquestes imatges des de la pròpia aplicació. Pujar una imatge des d'Android és molt diferent a pujar-la des de iOS o web. A l'apartat d'algorismes específics es cobreix com es soluciona aquest problema.

Aquestes imatges també es poden previsualitzar a cada targeta de la llista principal d'activitats. La forma circular s'ha inspirat en [aquesta pregunta de Stack Overflow](#) (Prithish, sense data). També s'ha incorporat que les imatges més grans, i que tarden més en carregar, tinguin un efecte de carrega, [com es pot veure a la guia de Flutter](#). (Flutter Fade in images with a placeholder, sense data)

Un error que va sorgir durant aquestes setmanes va ser el següent:



**Figura 50.** Prototip interactiu amb errors de la llista

Algunes targetes de la llista desapareixien i segur que no era problema de la implementació pròpia, havia de ser problema del SDK i una de les ultimes actualitzacions. Vaig obrir un [issue](#) (Flutter issue #72461, sense data) a Flutter i es va poder seguir a través d'un [issue](#) (Flutter issue #72171, sense data) creat uns dies abans. Finalment es va arreglar amb una actualització.

No tot es queda amb millores (o problemes, com acabem de veure) visuals, si no que també hi ha novetats al backend. Ja que en aquest punt, s'afegeix la informació de contacte, lloc i horari de l'activitat. També s'afegeix una altra dada que està emparellada amb la visibilitat de l'activitat: la data de visibilitat. D'aquesta forma, es pot ocultar una activitat o no, però lo que es segur es que passada la data de visibilitat, l'activitat es deixarà de veure. Això permet que l'administrador pugui declarar de forma automatitzada quan fer desaparèixer una activitat, sense necessitat de ser borrada, i poder-la reutilitzar en qualsevol moment.

Finalment, també s'afegeix una icona i un text de mostra de l'aplicació, [seguint aquest tutorial de Medium](#). (Rana, sense data)

## 6.5 Iteració 5: Implementació per a iOS

*Duració: 30 de desembre de 2020 a 20 de gener de 2021*

Per desenvolupar la aplicació per iOS es necessita un ordinador iMac<sup>17</sup>, com que no en tinc, vaig seguir [un tutorial](#) (Wikigain How to Install macOS Catalina on VMware on Windows 10 – PC, sense data) per instal·lar una imatge virtual de macOS<sup>18</sup> Catalina 10.15.5 a VMWare per Windows 10. Vaig escollir la versió Catalina ja que ja portava més d'un any al mercat i la versió 10.15.5 era bastant estable.

Segurament la millor forma de implementar l'aplicació es a través d'un equip d'Apple físic i no virtual. L'experiència implementant aquesta aplicació amb una imatge virtual, en el meu cas, va estar plena de llargs temps d'espera, configuracions, instal·lacions, solució d'errors i en definitiva, poca programació.

## 6.6 Iteració 6: Reorganització de la interfície d'usuari

*Duració: 20 de gener a 3 de març de 2021*

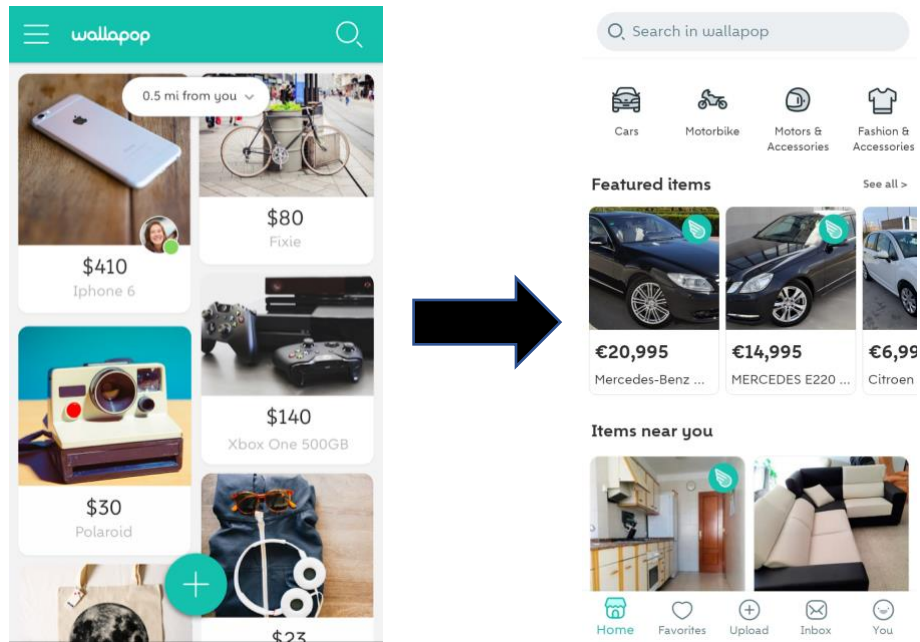
Personalment, utilitzo bastant Wallapop, una aplicació de compra-venta d'objectes de segona mà. També utilitzo [Yahoo Finance](#) (Yahoo Finance PlayStore, sense data), una aplicació de notícies del mont financer. [Revolut](#) (Revolut Playstore, sense data), l'aplicació del meu banc també es una de les aplicacions al meu telèfon. Sorprenentment, aquestes tres aplicacions, que tracten temes molt diferents, han servit d'inspiració per canviar l'estructura de l'aplicació.

En concret [Wallapop](#) (Wallapop Playstore, sense data) va fer el següent canvi en gener de 2021:

---

<sup>17</sup> Ordinador del fabricant Apple

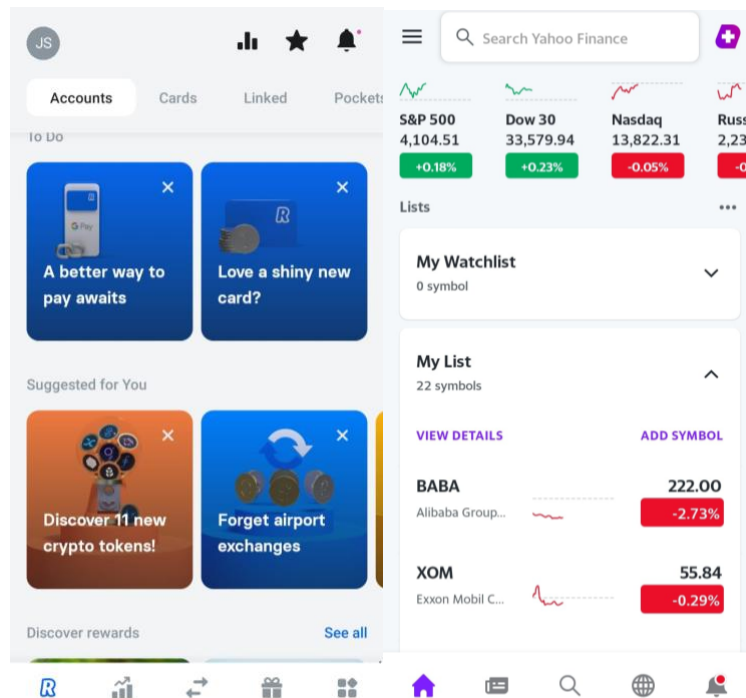
<sup>18</sup> Sistema Operatiu dels ordinadors iMac del fabricant Apple.



**Figura 51.** Actualització de la pantalla principal de Wallapop

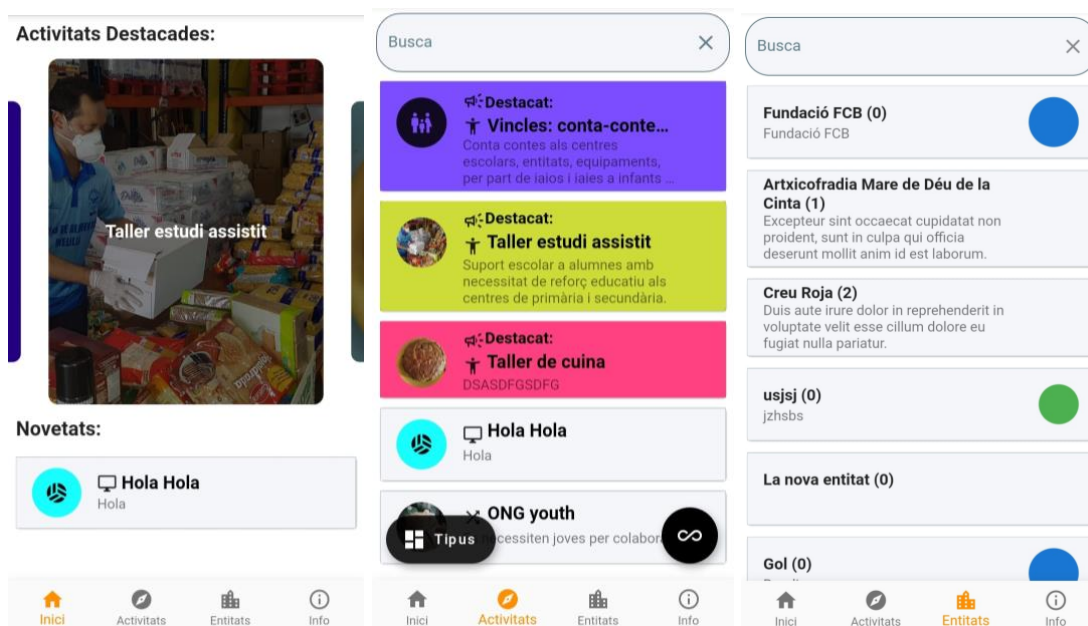
La diferència està en l'abandonament del Drawer superior, per canviar-lo per una barra inferior que fa de menú de l'aplicació.

Yahoo Finance i Revolut també tenen aquesta barra inferior:



**Figura 52.** Pantalles principals de Revolut i Yahoo Finance

La forma de organitzar apartats d'aquestes tres aplicacions es van traduir en un canvi molt semblant a l'aplicació:

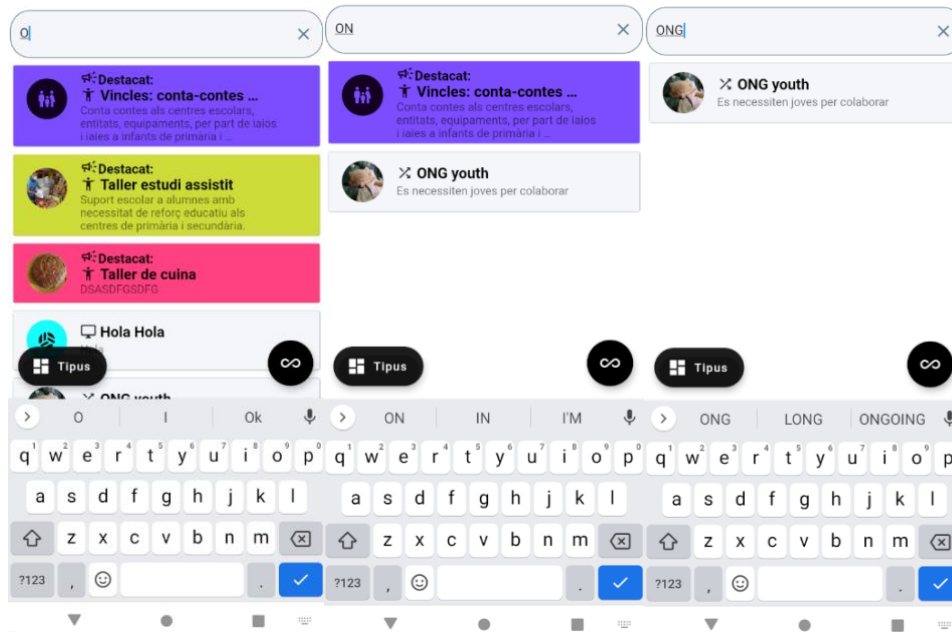


**Figura 53.** Prototip interactiu amb menú inferior

S'afegeix una barra inferior de navegació amb 4 apartats: Inici, Activitats, Entitats i info (-rmació). D'aquesta forma, el AppBar seria prescindible i aprofitar millor el espai de la pantalla.

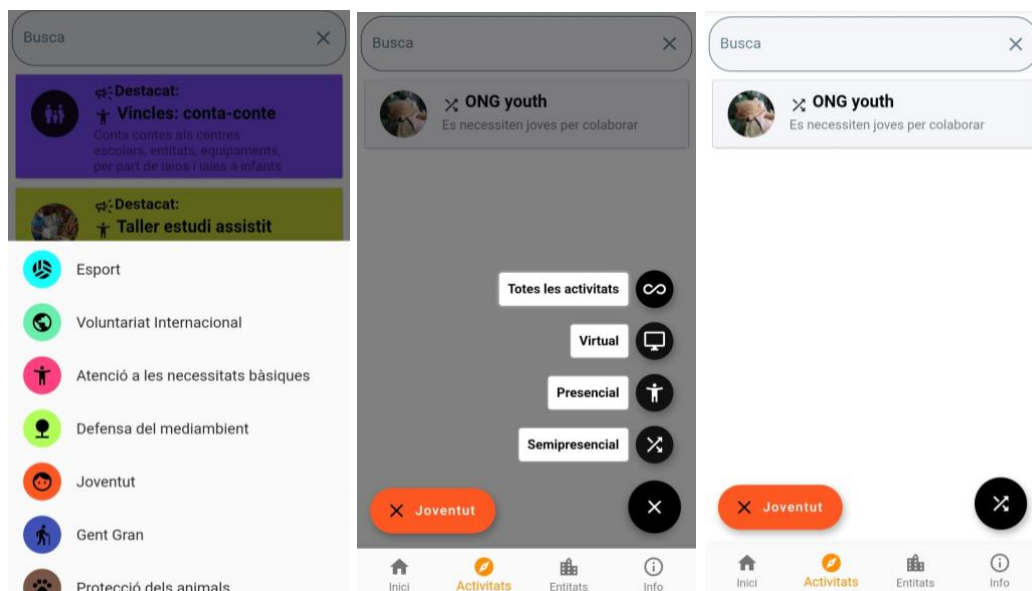
A la pantalla d'inici es mostren aquelles activitats noves, segons la data que han sigut publicades. Aquesta data s'afegeix quan es crea l'activitat. També apareixen en gran les activitats destacades en un carrousel que va canviant automaticament la activitat mostrada, pero que també es pot lliscar per fer el mateix. S'ha fet inspirat en la solució proposada en [aquest video](#) (Youtube: Flutter Tutorial - Carousel Slider - Loop & Infinite ListView, sense data) i [aquest](#) (Code, sense data). Per defecte es mostra la imatge de l'activitat, però si no hi hagués imatge, es mostraria el color del tipus de fons, enfosquit de [la manera proposada a aquesta pregunta de Stackoverflow](#) (jazzbnp, sense data).

La barra de cerca es canvia per una que mostra l'informació de forma dinàmica, dins de la mateixa pantalla i que reacciona conforme s'introdueixen lletres. L'estil de la barra [s'ha inspirat en aquest tutorial](#) (Way, sense data) i la programació que hi ha darrere s'ha inspirat [en aquest](#) (Addict, sense data).



**Figura 54.** Prototip interactiu amb cerca dinàmica

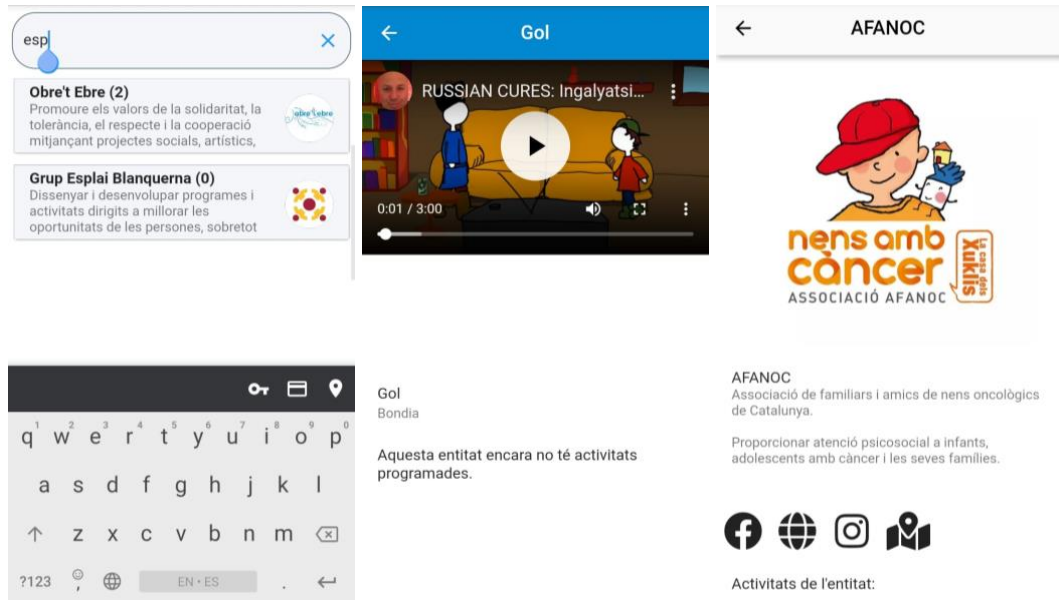
Es modifica el filtre de tipus per a incloure nous tipus. A les recomanacions de Material Design, un botó de tipus FAB no hauria de tindre més de 6 elements i es necessitaven 11. La nova solució va passar per l'anomenat Bottom Sheet, un desplegable on es poden escollir una de les 11 opcions. Per fer el Bottom Sheet, m'he inspirat [en aquest tutorial \(Youtube: Flutter & Firebase App Build by The Net Ninja, sense data\)](#), [aquest article \(Fernando, sense data\)](#), [aquest codi \(IshanFx, sense data\)](#) i [aquesta pregunta a Stackoverflow \(Mary, sense data\)](#). Tot i així, el botó FAB s'inclouria per poder filtrar segons l'activitat es presencial, semipresencial o virtual, adaptant l'aplicació a la situació del SARS-CoV-2.



**Figura 55.** Prototip interactiu amb filtre de presencialitat i tipus

La cerca funciona de la mateixa forma amb la pestanya d'entitats, però sense filtres. A cada entitat se li afegeix la possibilitat de veure una descripció, una foto/logo, reproduir un video i de tindre accés a les xarxes socials i localització en Google Maps de l'entitat. Les

icones de xarxes social si Google Maps venen del paquet d'icones [Font Awesome Flutter](#) (font\_awesome\_flutter, sense data). També es pot escollir el color de la barra superior de la entitat. Per fer que el color del nom de l'entitat també canviï amb el canvi de color (per evitar colors foscos sobre foscos o clars sobre clars), s'utilitza [la solució proposada en aquesta pregunta de Stackoverflow](#) (user6274128, sense data).



**Figura 56.** Prototip interactiu amb imatge i vídeo d'entitat

## 6.7 Iteració 7: Reorganització del codi

*Duració: 3 de març a 14 d'abril de 2021*

Aquest és el moment en que [s'anuncia Flutter 2](#) (Announcing Flutter 2, sense data). La gran novetat és que Flutter web surt del seu estat com a Beta i es transforma en estable. Cosa que podia semblar com a molt, però no ho era. Es va tractar de una actualització més, on el 99% del codi és compatible. Realment el més afecta a aquest projecte és el fet de que alguns plugins i paquets externs deixessin de ser compatibles, cosa que va suposar la espera de 1-2 setmanes a que es possessin al dia. Finalment, quan tots els paquets estaven actualitzats per funcionar a Flutter 2, es va actualitzar tot de cop: el SDK, Dart, plugins i paquets externs.

En aquest punt, l'aplicació està bastant provada i no té errors (a simple vista). Un dels problemes que té l'aplicació de base és la quantitat de codi repetit. Des del principi, s'han fet dues aplicacions (una per usuaris i altra per administradors) en una. Ara que els coneixements sobre Flutter són més grans, és un bon moment de fer aquest canvi, que és bastant simple.

La clau és portar tot allò comú a una mateixa carpeta (screens) i deixar de banda tots aquells fitxers que siguin específics de una part o l'altra de l'aplicació. També la creació d'una carpeta per utilitats (utils), amb totes les funcions externes que s'utilitzen diferents cops a molts llocs del codi. Aquests canvis van reduir el numero de linees de codi d'art de 5754 a 3686, millorant la estructura del codi i la seva llegibilitat. En principi, aquest canvi no s'ha traduït en una aplicació més lleugera ni ràpida.

Ara bé, un dels possibles problemes de l'aplicació és que un usuari normal accedeixi a la part de l'administrador. La solució és simple, que passa per l'us d'un booleà que sempre té valor fals. Només pot canviar a verdader si un usuari introdueix de forma correcta les

credecials. D'aquesta forma, quan s'entra a la part comuna, es comprova si aquest booleà es cert o fals i ensenya una part o altra de l'aplicació.

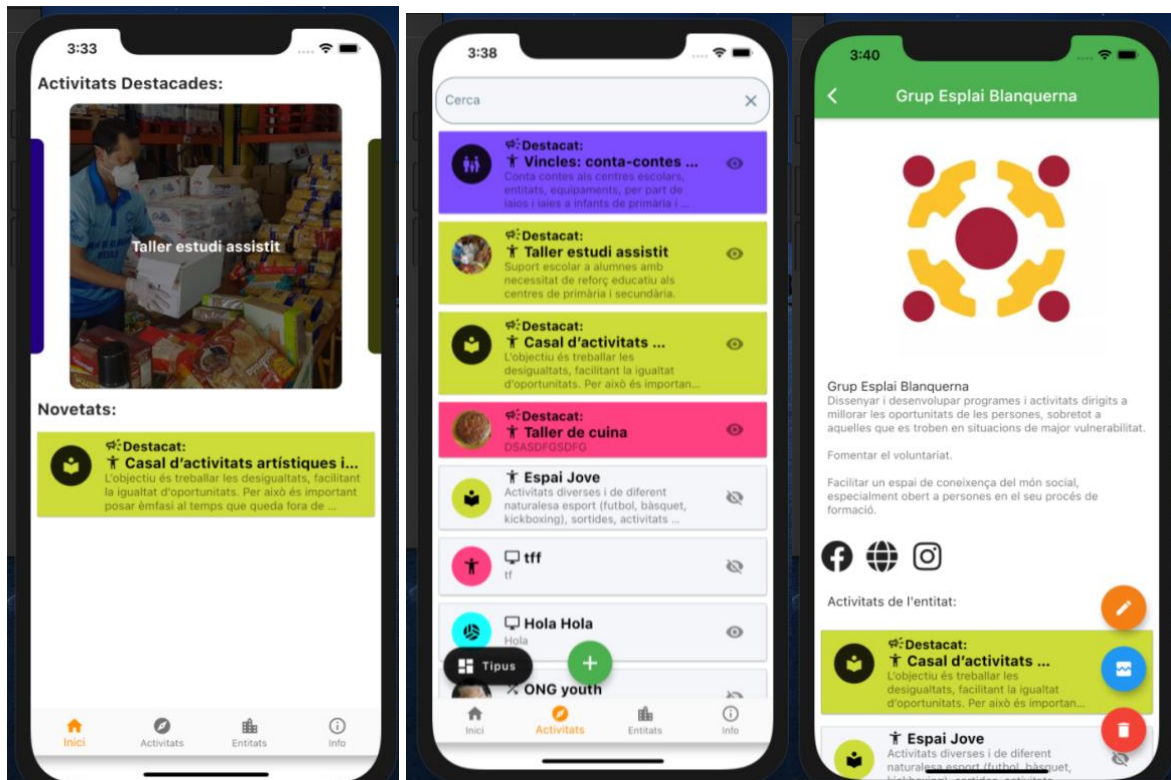
També s'ha modificat la implementació dels Strings per centralitzar-los tots en un mateix fitxer. [Aquesta es la manera recomenada per Flutter.](#)

```
class Strings {
  static const String type = 'Tipus';
  static const String typeSss = 'Serveis Sociosanitaris';
  static const String typeFamilia = 'Atenció i suport a les famílies';
  static const String typeEducacio = 'Educació i lleure';
  static const String typeEsport = 'Esport';
  static const String typeBasic = 'Atenció a les necessitats bàsiques';
  static const String typeInter = 'Voluntariat Internacional';
  static const String typeMedi = 'Defensa del mediambient';
  static const String typeJove = 'Joventut';
  static const String typeGran = 'Gent Gran';
  static const String typeAnimal = 'Protecció dels animals';
  static const String typeCult = 'Cultura';
  ...
}
```

Codi 3. Strings

### 6.8 Iteració 8: Tornada al desenvolupament a iOS

La prova a iOS va ser caòtica i aquest segon intent va pel mateix camí. Desenvolupar utilitzant una imatge virtual de iOS dins d'una imatge virtual d'un iMac no és fàcil. Després de provar sort amb moltes documentacions i tutorials, la solució en el meu cas va ser [seguir aquest simple tutorial](#). Els tests s'han fet a un dispositiu iPhone 12 Pro Max i aquestes són les captures de com es veu a aquest dispositiu:



També s'ha provat a un iPad Air de 4ta generació.



## 7 Implementació

### 7.1 Especificacions addicionals

Tal i com sol passar en aquest tipus de desenvolupaments, les especificacions canvien i s'afegeixen nous objectius. Quan s'utilitza la metodologia àgil, en cada iteració es recull feedback, que té altes probabilitats de canviar en algun sentit l'aplicació. També, el fet que Flutter sigui un SDK en desenvolupament, implica que poden haver-hi noves possibilitats de les que es pugui beneficiar el projecte. En definitiva, s'han afegit noves funcionalitats que no estaven contemplades a l'anàlisi funcional, i que aquí repassem per veure que ha canviat.

#### 7.1.1 Diagrama de classes

El diagrama de classes exposat a l'apartat de disseny passaria a ser el següent:

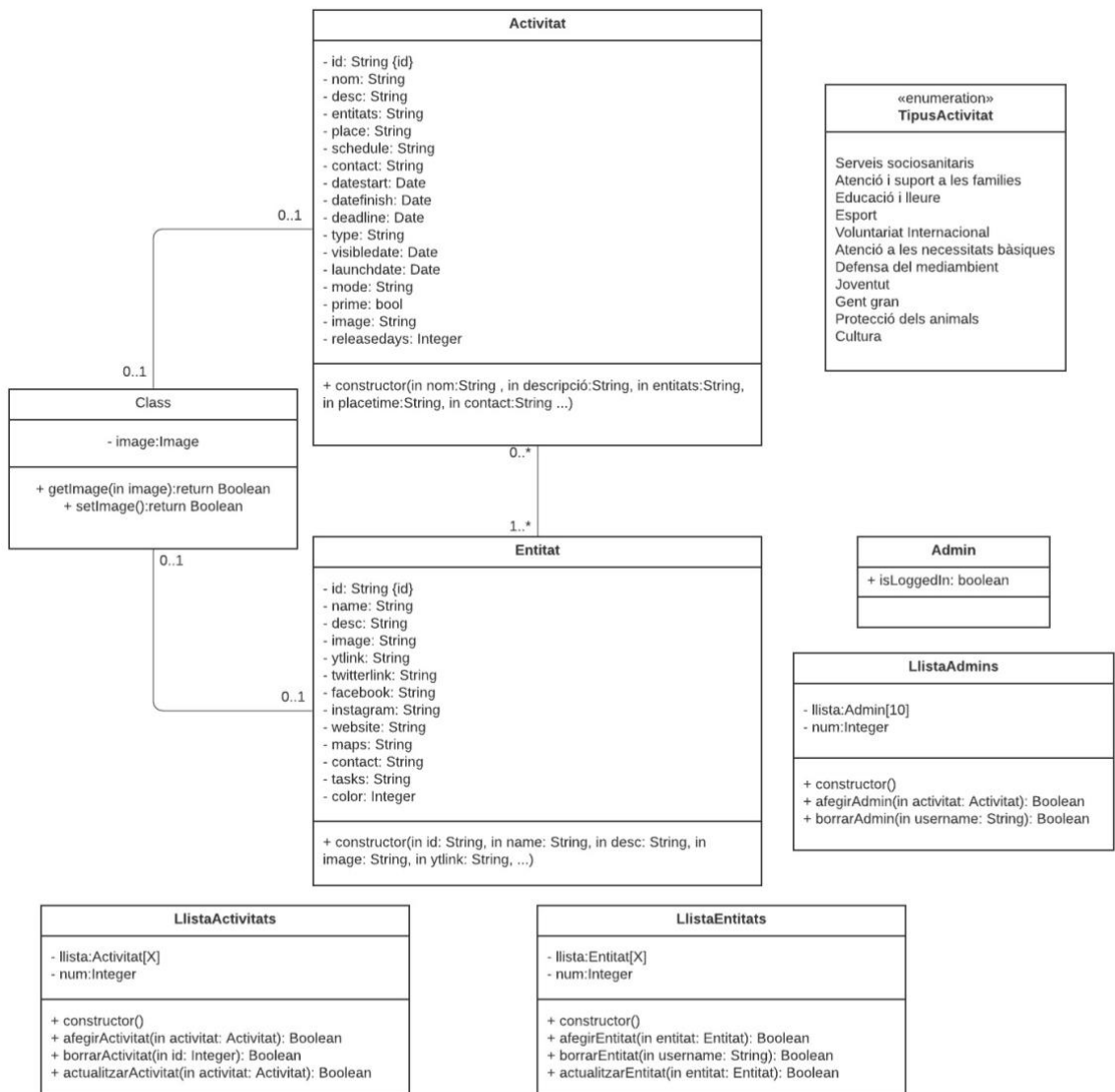


Figura 57. Diagrama de classes actualitzat

### 7.1.2 Requisits no implementats

S'han implementat totes les funcionalitats de la versió 1 i 2, exceptuant l'opció de seleccionar activitats com a preferides. Les funcionalitats de la versió 3 es deixarien per un futur desenvolupament.

El requisit 10. Activitats preferides no s'ha introduït ja que els usuaris no podrien guardar aquestes activitats en diferents plataformes, si no que es quedarien restringides a ser part de la informació local. S'ha cregut que el perdre aquesta llista de preferits, a causa de un canvi de dispositiu, podria suposar una frustració per l'usuari. A continuació es declaren els requisits nous, incorporats a posteriori en l'aplicació. Aquests requisits nous resulten del feedback recollit durant cada iteració del procés de desenvolupament.

### 7.1.3 Requisits modificats

<b>Títol</b>	03. Detalls activitat (actualitzat)
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	A partir d'un llistat d'activitats, es pot entrar a veure els detalls complets de l'activitat. Aquests detalls són el nom, descripció, entitats organitzadores (1 o més), horari, dades de contacte, data de començament, data de finalització, data d'expiració, tipus d'activitat, <b>imatge, presencialitat (presencial, virtual o semipresencial), si està destacada, lloc, dia de publicació i dies que estarà com a novetat.</b> Dins de les dades de contacte, qualsevol correu electrònic o telèfon ha de tindre un hiperenllaç per poder enviar un correu o trucar directament.

**Taula 27.** Requisit 03. Detalls d'activitat (actualitzat)

### 7.1.4 Requisits afegits

Els requisits afegits no han passat per tot el procés de desenvolupament. És a dir, no s'ha fet un anàlisi ni disseny d'aquests requisits, si no que s'han implementat de manera directe. S'ha vist durant el desenvolupament que una vegada l'aplicació ja té certes funcionalitats nucli, la expansió mitjançant nous requisits es bastant simple, i no és necessari passar pels processos de anàlisi ni disseny.

<b>Títol</b>	26. Detalls entitat
<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	A partir d'un llistat d'entitats, es pot entrar a veure els detalls complets de l'entitat. Aquests detalls són el nom, descripció, activitats que organitza (0 o més), foto, vídeo i enllaços a les xarxes socials (facebook, instagram, twitter o pagina web). També s'inclou un enllaç a Google Maps per saber la localització de l'entitat en un mapa.

**Taula 28.** Requisit 26. Detalls d'entitat

<b>Títol</b>	27. Pàgina principal
--------------	----------------------

<b>Actor</b>	Usuari, Administrador
<b>Prerequisits</b>	L'administrador ha d'estar identificat
<b>Descripció</b>	Pantalla que mostra un llistat d'activitats destacades i una altra de activitats noves (recentment afegides).

**Taula 29.** Requisit 27. Pàgina principal

<b>Títol</b>	28. Filtre segons presencialitat
<b>Actor</b>	Usuari
<b>Prerequisits</b>	Cap
<b>Descripció</b>	Filtre per poder escollir si es volen veure les activitats presencials, semipresencials o virtuals de la llista d'activitats. Funció especialment dedicada a la crisi del COVID-19.

**Taula 30.** Requisit 28. Filtre segons presencialitat

## 7.2 Estructura del projecte: directoris i fitxers

### 7.2.1 Fitxer *main.dart*

És el fitxer amb el widget pare de l'aplicació, del que cauen la resta. Fa les inicialitzacions del Flutter engine i Firebase per després fer la crida al widget Home.

### 7.2.2 Directori *data*

#### 7.2.2.1 Fitxer *activity.dart*

Conté la classe *Activity*, on cada instància representa una activitat amb els seus atributs.

```
String id;
String title;
String desc;
String type;
String contact;
String place;
String schedule;
List<String> entities;
DateTime startDate;
DateTime finalDate;
DateTime visibleDate;
DateTime launchDate;
int releasedays;
bool visible;
bool prime;
String image;
String mode;
```

#### **Codi 4.** Atributs d'*Activity*

#### 7.2.2.2 Fitxer *entity.dart*

Conté la classe *Entity*, on cada instància representa una entitat amb els seus atributs.

```
String id;
```

```
String name;  
String desc;  
String image;  
String ytlink;  
String twitter;  
String facebook;  
String instagram;  
String website;  
String maps;  
String contact;  
String tasks;  
int color;
```

### **Codi 5. Atributs d'entitat**

#### 7.2.2.3 Fitxer admin.dart

Conté un booleà que és verdader o fals segons l'administrador està identificat o no. Serveix per comprovar en diferents parts de l'aplicació qui és l'actor que accedeix a l'aplicació.

```
bool isLoggedIn = false;
```

### **Codi 6. Fitxer admin.dart**

#### **7.2.3 Directori services**

##### *7.2.3.1.1 Arxiu activity\_service.dart*

Conté les operacions CRUD que utilitzen l'API<sup>19</sup> de Firestore per comunicar-se amb Firebase. S'encarrega de fer la conversió de dades entre el objectes de classe Activity a mapejos destinats a ser enviats. De la mateixa forma, transforma el tipus de dades rebut en objectes de classe Activity.

##### *7.2.3.1.2 Arxiu entity\_service.dart*

Funciona de la mateixa forma que activity\_service, com a intermediari entre l'aplicació i l'API de Firebase.

##### *7.2.3.1.3 Arxiu signin\_service.dart*

Només té un mètode, que es comunica amb l'API de Firebase Authentication, per verificar el correu electrònic i la contrasenya passades per paràmetre.

#### **7.2.4 Directori screens**

##### 7.2.4.1 Subdirectori screens/activities

##### *7.2.4.1.1 Arxiu activity\_details.dart*

Conté les operacions de modificació i esborrament de l'activitat. També té les operacions per esborrar o afegir una imatge. Es fa una crida a activity\_details\_body.dart, que conté el widget amb tota la informació de l'activitat.

---

<sup>19</sup> API = Application Programming Interface. Un codi o interfície que fa d'intermediari entre diferents aplicacions. Defineix una sèrie d'operacions per a que es comuniquin les aplicacions.

#### 7.2.4.1.2 *Arxiu activity\_details\_body.dart*

Mostra tota la informació de l'activitat, títol, descripció, contactes, dates...

#### 7.2.4.1.3 *Arxiu activity\_list.dart*

Aquesta es la llista d'activitats amb la seva barra de cerca. Combina els filtres amb la cerca per mostrar resultats.

#### 7.2.4.1.4 *Arxiu activity\_list\_tile.dart*

Construeix cada element/targeta de la llista i la informació que es mostra d'aquest a la previsualització de la llista.

#### 7.2.4.1.5 *Arxiu activity\_present\_entities.dart*

Aquest fitxer s'encarrega de consultar les activitats i entitats dins de la base de dades per veure quines entitats organitzen una activitat concreta. Una vegada s'han creuat les dades, es mostra una llista d'entitats. Està pensada per a que dins dels detalls d'una activitat, es puguin veure quines entitats organitzen aquesta activitat i poder accedir a la informació completa d'aquestes.

### 7.2.4.2 Subdirectori screens/Authentication

#### 7.2.4.2.1 *Arxiu signin\_screen.dart*

Es tracta de la pantalla que es mostra quan l'administrador vol iniciar sessió. Construeix un formulari amb un botó per enviar el correu electrònic i contrasenya. En cas que aquests siguin incorrectes, es mostra el missatge pertinent. En cas que les credencials siguin correctes, fa una crida a la pantalla principal Home des del punt de vista de l'administrador.

### 7.2.4.3 Subdirectori screens/entities

#### 7.2.4.3.1 *Arxiu entity\_details.dart*

Quan des de la llista d'entitats es clica una entitat, surten els detalls, els quals es construeixen dins de *entity\_details.dart*. Presenta per pantalla la informació de l'entitat, i en el cas que sigui l'administrador qui accedeix, es troben els botons per esborrar l'entitat o accedir al formulari de modificació.

#### 7.2.4.3.2 *Arxiu entity\_list.dart*

Es tracta de la llista d'entitats junt a la seva barra de cerca. Utilitza el patró BLoC per obtenir les entitats, junt a la cerca i filtres per ensenyar la llista.

#### 7.2.4.3.3 *Arxiu entity\_list\_tile.dart*

Representa la informació que es mostra en cada targeta d'entitat dins de la llista d'entitats.

#### 7.2.4.3.4 *Arxiu entity\_present\_activities.dart*

Des dels detalls de l'entitat, es crida a *entity\_present\_activities* per desplegar la llista d'activitats que hi ha assignades a aquesta entitat.

#### 7.2.4.3.5 *Arxiu socialnetworks.dart*

Des dels detalls es fa una crida a `socialnetworks.dart` per representar els botons de les xarxes socials de l'entitat.

#### *7.2.4.3.6 Arxiu `video_screen.dart`*

En cas que l'entitat tingui un vídeo, aquest arxiu fa la presentació del vídeo i els seus botons a més de comunicar-se amb [l'API de Youtube Player](#) (YouTube Player API Reference for iframe Embeds, sense data). Utilitza un plugin que facilita aquest procés, anomenat [youtube\\_player\\_inframe](#) (`youtube_player_iframe`, sense data).

#### *7.2.4.4 Subdirectori `screens/main`*

##### *7.2.4.4.1 Arxiu `carousel.dart`*

Com el seu propi nom indica, intenta simular un carrusel, fent lliscar elements per pantalla. Aquests elements són les activitats destacades, que aparèixer per defecte com a primera pantalla quan s'inicia l'aplicació. Aquest fitxer també inclou la llista de novetats.

##### *7.2.4.4.2 Arxiu `home.dart`*

Aquesta és el fitxer més complex i amb més línies de codi de l'aplicació, que a l'hora serveix de nucli per la resta de funcionalitats. S'encarrega de la barra de navegació i dels botons flotants (filtres o botó d'afegir).

### **7.2.5 Directori *utils***

#### *7.2.5.1 Arxiu `colorizer.dart`*

Conté widgets específics de colors i icones que s'utilitzen en diferents parts de l'aplicació. S'utilitza especialment per seleccionar els colors de cada tipus d'activitat de cara a aplicar els filtres.

#### *7.2.5.2 Arxiu `commonfunctions.dart`*

Conté funcions que s'utilitzen en moltes parts de l'aplicació per transformar tipus de dades, ordenar i aplicar filtres a llistes.

### **7.2.6 Directori *adminspecific***

#### *7.2.6.1 Subdirectori `activities`*

##### *7.2.6.1.1 Fitxer `activity_form_field.dart`*

Construeix l'estructura de tots els camps de text i selectors per totes les dades d'una activitat. També tots els validadors de les dades, per evitar que hi hagi camps buits o amb dades incorrectes.

##### *7.2.6.1.2 Fitxer `activity_form.dart`*

A part de ser el widget que hi ha per damunt del formulari, també fa de intermediari entre el `ActivityService` i el formulari. Una vegada l'usuari envia el formulari, d'es d'aquí es crida a `ActivityService` per a que es pugin les dades, ja sigui per modificar o crear.

#### *7.2.6.2 Subdirectori `entities`*

##### *7.2.6.2.1 Fitxer `entity_form_fields.dart`*

Similar al model de les activitats, aquesta classe conté el widget de l'estructura dels elements del formulari

#### 7.2.6.2.2 *Fitxer entity\_form.dart*

Es tracta del widget que està damunt de `entity_form_fields.dart`. Envia les dades del formulari `EntityService` per a que es crei o modifiqui una entitat.

### 7.3 Algoritmes específics de les parts més complexes

En aquest apartat s'explica com s'ha realitzat i el raonament que hi ha darrere de les parts més complexes de l'aplicació. Alguns dels algoritmes i codis exposats estan modificats en favor de les explicacions.

#### 7.3.1 *Patró BLoC implementat*

Encara que la explicació que es fa a continuació es fa des del punt de vista de les Activitats, la llista d'Entitats funciona de la mateixa forma. El codi exposat es el codi de una de les primeres versions de l'aplicació, i que encara que les següents varien una mica la organització d'aquest codi, els conceptes són els mateixos. Es tracta de un codi més simple (ja que en futures versions s'incorporen noves funcionalitats), i s'entén millor a l'hora de donar aquesta explicació.

L'element més important del patró BLoC és `Stream`, que treballa com una canonada. Quan s'inserta alguna cosa dins, aquesta flueix fins el final de la canonada. `Stream` funciona de la mateixa forma amb les dades. Per controlar aquest flux, s'utilitza normalment un `StreamController` o un `StreamProvider` (en el nostre cas s'utilitza el segon).

Per recollir el llistat de Activitats implementant l'arquitectura BLoC, em de fer que un `StreamProvider` estigui escoltant per veure si arriben activitats. `StreamProvider` és una classe que escolta els events d'un `Stream`, per tal de passar aquests events als fills en cas de novetat. Quan hi ha una novetat al `Stream`, es crea un event que actualitza els widget fills i durant aquesta actualització, es consulten les noves dades.

Els fills de `StreamProvider` poden accedir al contingut de la sequencia d'events. Per fer-ho, em embolicat `Scaffold` (widget que crea la estructura Material Design) en un `StreamProvider` i d'aquesta forma, desde el primer moment que s'executi l'aplicació, ja s'estarà escoltant per si arriben dades desde `Firestore`. Si hi ha una actualització, `StreamProvider` crea un event, els seus fills consulten la nova actualització i es reconstrueixen amb l'informació nova.

```
class HomePage extends StatefulWidget {
  _HomePageState createState() => _HomePageState();
}
class _HomePageState extends State<HomePage> {
  Widget build(BuildContext context) {
    return StreamProvider<List<Activity>>.value(
      value: ActivityService().activities,
      child: Scaffold(
        appBar: searchBar.build(context),
        drawer: new Drawer(...),
        body: HomeList(),
      ),
    );
  }
}
//...
```

#### Codi 7. HomePage simplificat

A continuació es pot veure el fragment que pregunta a Firebase Firestore, desde la classe `ActivityService`. Al principi no es treballa amb llistes d'activitats, si no que s'utilitzen `Streams`. Aquest `Stream` ve donat per la instància de la col·lecció d'activitats. Aquesta col·lecció està formada per documents, on cadascun conté la informació d'una activitat. Per transformar la informació d'aquest document en un tipus tractable fàcilment per l'aplicació, es fa mapeig a `String`, `int` o `Date`, que són tipus de dades de Dart.

```
CollectionReference ref = FirebaseFirestore.instance.  
                                collection("Activities");  
Stream<List<Activity>> get activities {  
    return ref.snapshots().map(_ActivitiesFromSnapshot);  
}
```

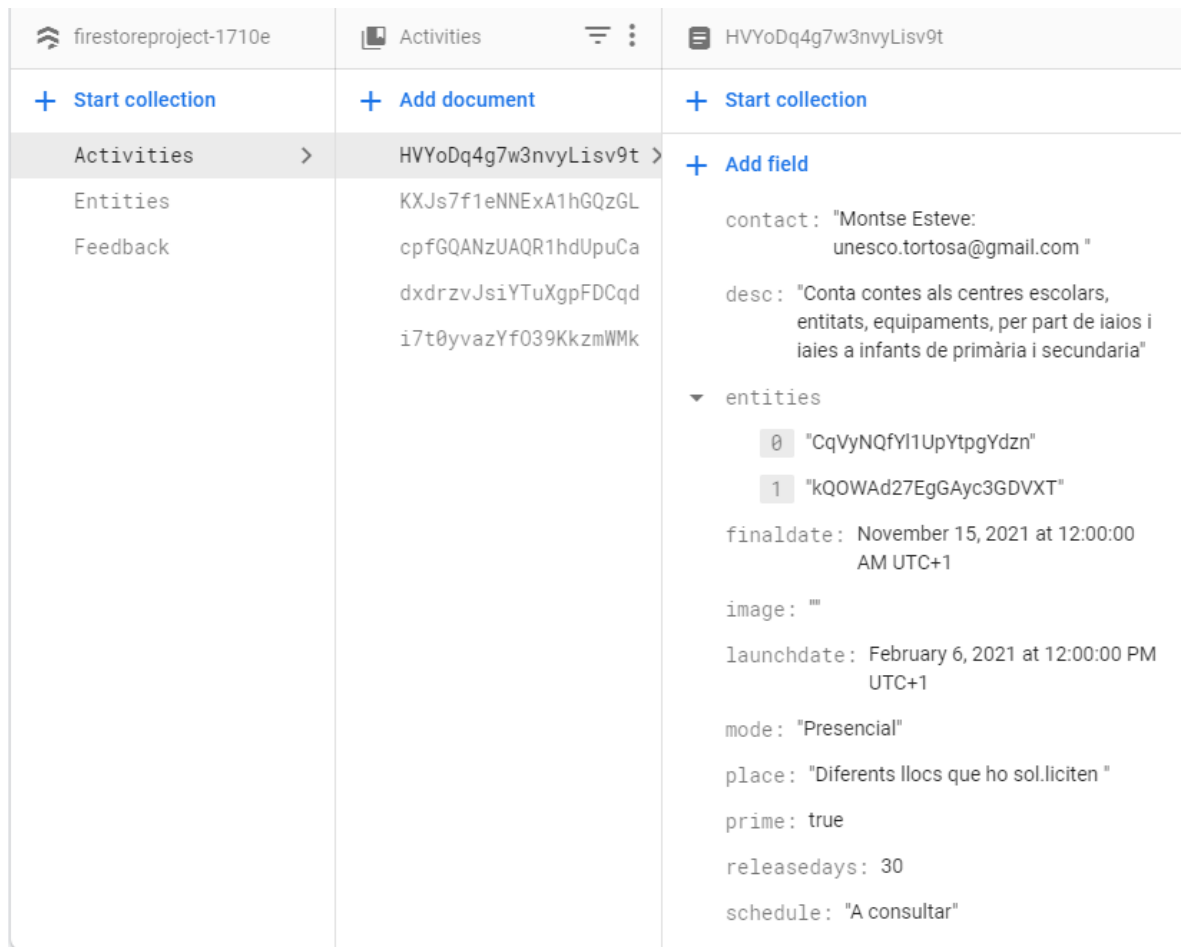
### **Codi 8. Descarrega d'activitats des de Firebase**

Si fos una llista, no es podria actualitzar si hi hagués algun canvi a la base de dades, però com que és un `Stream`, si hi ha una actualització o nova activitat, tots els usuaris la veuràn instantàniament. Aquesta informació es mostra a `HomeList`, on finalment es transforma el `Stream` en una llista (`List`).

```
List<Activity> list_activities=Provider.of<List<Activity>>(context);  
return ListView.builder(  
    itemCount: list_activities.length,  
    itemBuilder: (context, index){  
        return HomeListTile(activity: list_activities[index]);  
    }  
);
```

### **Codi 9. HomeList simplificat**

Tota la informació de les activitats ve de Firebase, on s'organitza la informació en col·leccions de documents. Cada col·lecció en el meu cas és un tipus d'objecte (`Activitat`) que pot emmagatzemar diferents objectes (`Activitats`) amb cadascun els seus camps (informació sobre l'activitat). Cada activitat està identificada per un identificador (p.e. `HVYoDq4g7w3nvyLisv9`) i té diferents camps amb el nom, contacte, descripció... Aquesta forma d'emmagatzemar la informació és fàcil de transformar en objectes Dart.



**Figura 58.** Estructura de les dades a Firebase Firestore

Dins de HomeListTile, l'únic que quedaria és mostrar la informació d'una activitat, representada per una targeta dins de la llista de la interfície.

### 7.3.2 Cerca amb actualització dinàmica

Al principi es va plantejar una cerca tradicional, en la que l'usuari introdueix un conjunt de caràcters, es prem un botó, es fa una cerca i s'ensenyen els resultats en una nova pantalla.

Per millorar la usabilitat i realitzar una millor cerca, es va proposar fer-la dinàmica, en aquest apartat s'explica com.

Tot comença amb un TextEditingController, que permet connectar el camp de text de la cerca amb altres parts de l'aplicació que reaccionen a canvis en el camp de text. Quan s'inicia el TextEditingController, se li afegeix un Listener, és a dir, se li afegeix una funció que s'executarà a cada canvi en el camp de text. Aquesta funció es diu onSearch i quan es crida, s'executa un setState(), que independentment del que es faci dins d'aquesta funció, actualitzarà l'estat del widget, reconstruint tots els widgets que tingui com a fills. En aquest cas, el que es fa dins de setState() és actualitzar la variable searchText, que filtra les activitats que tenen camps que contenen aquesta cadena de caràcters.

```
TextEditingController searchController;
String searchText;
Widget build(BuildContext context) {
  List<Entity> entitylist= Provider.of<List<Entity>>(context) ?? [];
```

```

var listActivities = Provider.of<List<Activity>>(context) ?? [];
listActivities=CommonFunctions.sortActivityPrimes(listActivities);
return Column(
  children: [
    TextFormField(
      controller: searchController,
      decoration: InputDecoration(...),
    ),
    Expanded(
      child: ListView.builder(
        itemCount: listActivities.length,
        itemBuilder: (context, index) {
          var act = CommonFunctions.toStringLowerCaseComplete(...);
          if((searchtext==null)|| (act.contains(searchtext)))
            {...}
          else return Container();
        }
      ),
    ),
  ],
);
}

void initState() {
  searchController = TextEditingController();
  searchController.text = "";
  searchController.addListener(_onSearch);
  super.initState();
}

void _onSearch() {
  setState(() {
    searchtext = searchController.text;
  });
}
}

```

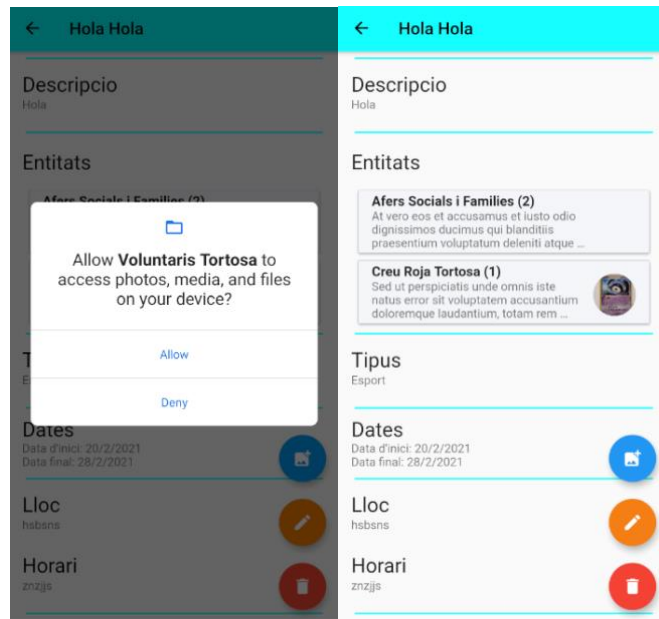
### **Codi 10. Cerca dinàmica**

Com que cada cop que s'introdueix un nou caràcter, s'actualitza el widget pare actualitzant la informació en pantalla, això permet l'usuari previsualitzar si el que està escrivint està disponible.

Aquesta cerca dinàmica té el risc de si hi ha massa activitats (o entitats), tardar massa en fer la cerca. En aquest cas, estem parlant de menys de 100 activitats, cosa que permet que el temps de cerca sigui pràcticament immediat.

#### **7.3.3 Pujar imatges a Firebase Storage**

Des de la part de l'administrador es va incloure un nou botó per afegir imatges, però el problema de pujar imatges és que és molt diferent fer-lo des de una web que des de una aplicació per mòbil. Per això, es va crear un algorisme que primer comprovi si s'està executant la versió web o la mòbil, per a partir d'aquí, actuar d'una forma o altra.

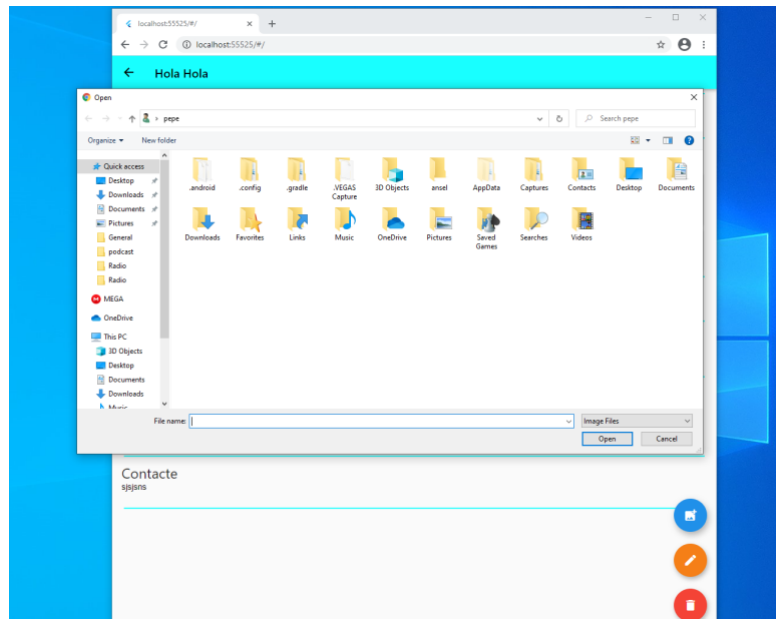


**Figura 59.** Prototip interactiu pujant imatges i permissos

Primer es va realitzar la part mòbil, que té com a principal diferència que s'ha de preguntar a l'usuari per permissos a accedir als fitxers del dispositiu. Per sol·licitar permissos s'ha utilitzat [permission handler](#) (`permission_handler`, sense data), que aporta una API per gestionar i demanar permissos, feta per [baseflow](#) (`Baseflow`, sense data). L'usuari pot escollir una imatge a partir del [plugin image picker](#), (`image_picker`, sense data) plugin fet per desenvolupadors de Flutter. La imatge es transforma en un objecte `File`, que es puja fàcilment al servidor de `Firebase Storage`. Aquesta operació retorna un URL i s'emmagatzema l'URL com a part de la informació de l'activitat (junt al seu nom, descripció, visibilitat...).

En un principi vaig intentar fer el mateix codi per web i mòbil, cosa que em va donar molts problemes. Aquest és un problema que no s'havia comentat abans a [StackOverflow](#), on [vaig publicar una pregunta i em van ajudar](#) (Santo, Error uploading a photo to Firebase storage with Flutter Web using Pickers, sense data). També em vaig beneficiar d'aquestes preguntes que amb [aquest error](#) (Flutter Web: firebase\_storage: MissingPluginException No implementation found for method StorageReference#putData, sense data) i [aquest](#) (Moros, sense data).

La part web no necessita cap permís i a partir d'aquí el procediment és semblant, tot i que amb algunes particularitats. En comptes d'utilitzar `image_picker`, s'utilitza `image_picker_for_web`, que es tracta d'una implementació web de l'anterior. L'objecte que es retorna quan es llegeix la imatge no és un `File`, sinó que un grup de Bytes. Tot i així, `Firebase Storage` té una funció que facilita pujar imatges d'aquesta forma.



**Figura 60.** Prototip interactiu pujant imatges a web

Per últim, s'ha de dir que aquesta funció es tracta de forma assíncrona i que d'aquesta forma, mentre es puja la imatge, l'usuari pot seguir interactuant amb l'aplicació.

```
uploadImage() async {
  final _storage = FirebaseStorage.instance;
  final _picker = ImagePicker();
  PickedFile image;
  if (kIsWeb) {
    image = await _picker.getImage(source: ImageSource.gallery);
    var bytes = await image.readAsBytes();
    if (image != null) {
      var ref = _storage.ref();
      var snapshot = await ref
        .child('descriptionimages/' + widget.activity.id)
        .putData(bytes);
      var downloadUrl = await snapshot.ref.getDownloadUrl();
      setState(() {
        widget.activity.image = downloadUrl;
        ActivityService().updateActivity(widget.activity);
      });
    } else print('No Path Received');
  } else {
    await Permission.photos.request();
    var permissionStatus = await Permission.photos.status;
    if (permissionStatus.isGranted) {
      image = await _picker.getImage(source: ImageSource.gallery);
      var file = File(image.path);
      if (image != null) {
        var snapshot = await _storage
          .ref()
          .child('descriptionimages/' + widget.activity.id)
          .putFile(file);
        var downloadUrl = await snapshot.ref.getDownloadUrl();
        setState(() {
          widget.activity.image = downloadUrl;
          ActivityService().updateActivity(widget.activity);
        });
      }
    }
  }
}
```

```
    } else print('No Path Received');  
  } else print('Grant Permissions and try again');  
}  
}
```

### **Codi 11. Pujada d'imatges a mòbil i web**

## 8 Avaluació

Avaluació. Disseny dels casos de prova que cobreixin els requisits i els resultats. Relació de les proves realitzades per avaluar cada part).

### 8.1 Tests funcionals

En aquest apartat s'analitza i es posa a prova que els requisits funcionals de l'aplicació es garanteixin, fins i tot en les situacions més adverses. Aquests cassos d'us s'han testejat i verificat a les 3 plataformes del projecte (Android, Web i iOS) de forma paral·lela.

Aquests tests s'han fet tenint en compte que no hi ha cap anomalia per part del servidor, és a dir, que hi ha connexió a internet i es pot accedir a les dades de Firebase. En cas que no funcionés Firebase o la connexió a aquest, l'aplicació no tindria dades.

#### 8.1.1 Seqüència de tests per a usuaris

Prova	Resultat
Accedir a la pantalla principal (amb el carrusel), poder veure les activitats destacades i novetats, assumint que hi ha a la base de dades.	Es veuen ✓
Accedir a la pantalla principal si no hi ha novetats, però si destacades.	S'ha programat per a que si no hi ha cap llista de novetats, no es mostri ni el títol de novetats. ✓
Accedir a la pantalla principal si no hi ha activitats destacades, però si hi ha destacades.	S'ha programat per a que si no hi ha cap activitat destacada, no es vegin cap i es vegi directament les novetats. ✓
Accedir a la pantalla principal si no hi ha activitats destacades ni activitats noves.	No es veuen ni novetats ni activitats destacades. L'aplicació s'ha programat per a que una de les dos al menys existeixi sempre. ✗
No hi ha cap activitat a la llista d'activitats	No es veu cap activitat. ✓
Accedir a la pantalla principal i veure una activitat oculta.	No es veu. ✓
Accedir a la llista d'activitats i veure les activitats destacades a dalt i la resta a baix.	Es veuen d'aquesta forma ✓
Accedir a la llista d'activitats i que no hi hagi activitats o totes estiguin ocultes.	No es veu cap activitat ✓
Cercar una paraula que està a algun dels camps d'una activitat, inclòs pel nom de l'entitat que l'organitza.	Es veu ✓
Cercar el una paraula que no coincideix	No es veu cap activitat ✓
Prémer el botó de creueta mentre es cerca	S'esborra la paraula cercada i es torna a veure el llistat complet d'activitats. ✓

Aplicar el filtre segons el tipus d'activitat o presencialitat si hi ha alguna activitat assignada a aquest tipus.	Es veu la llista d'activitats assignades a aquest tipus. ✓
Aplicar el filtre segons el tipus d'activitat o presencialitat si no hi ha cap activitat assignada a aquest tipus	Es veu una llista buida ✓
Si una activitat no té imatge	Es veu una icona que representa el tipus de l'activitat i en cas de veure els detalls de l'activitat, no es veurà cap imatge. ✓
Des dels detalls d'una activitat, clicar un correu electrònic o telèfon si el format es correcte.	S'obre l'aplicació del correu o de fer trucades. ✓
Des dels detalls d'una activitat, clicar un correu electrònic o telèfon si el format no és correcte.	Sempre que el format del telèfon sigui 123456789 (sense espais) i el correu electrònic sigui paraula@paraula.paraula, és reconeixerà correctament. Si no es correcte, no es pot clicar. ✗
Des dels detalls d'una activitat, clicar una entitat.	S'obren els detalls de l'entitat ✓
Filtrar llista segons un tipus que no té cap activitat assignada.	Es veu una llista buida ✓
La llista d'entitats està ordenada alfabèticament segons el nom de l'entitat	Sí ✓
Cercar el una paraula que està al títol, descripció o tasques de l'entitat	Es veu ✓
Cercar el una paraula que no coincideix al llistat d'entitats	No es veu cap entitat ✓
No hi ha cap activitat a la llista d'entitats	No es veu cap entitats. ✓
Aplicar filtres a la llista d'entitats	No es pot ✓
Prémer el botó de creueta mentre es cerca	S'esborra la paraula cercada i es torna a veure el llistat complet d'entitats. ✓
Si una entitat no té imatge	Es veu un cercle del color de l'entitat a la llista d'entats i no es veu cap imatge als detalls de l'entitat. ✓
Si una entitat no té imatge però si vídeo.	Es veu un cercle del color de l'entitat a la llista d'entats i no es veu cap imatge als detalls de l'entitat, però si un vídeo. ✓
Si una entitat té imatge i vídeo.	Es veu un cercle amb la imatge previsualitzada i en els detalls no es veu la imatge. En el lloc de la imatge, es veu el vídeo. ✓

Reproduir el vídeo minimitzat o en pantalla completa.	Es pot ✓
Obrir el vídeo a Youtube	Es pot ✓
Es treu el vídeo si es surt de la pàgina de l'entitat que l'està reproduint	Es treu ✓
Si una entitat té activitats, es mostra un llistat d'activitats. El numero que es previsualització a la llista coincideix amb el numero d'activitats.	Si ✓
Si una entitat no té activitats	Es mostra una explicació conforme no hi ha activitats disponibles a aquesta entitat. ✓
Si una entitat no té informació de contactes, xarxes socials o tasques	No es mostren aquests camps, però si altres. ✓
Enviar feedback sense contingut	S'ha d'introduir contingut al formulari de feedback. ✓
Clicar successivament en les entitats d'una activitat i en les activitats de les entitats.	Cap problema, tot i que per tornar al menú principal s'ha de clicar varies vegades enrere. S'ha fet una prova amb 30 iteracions, entrant en les activitats d'entitats i en les entitats de les activitats successivament. ✓

### 8.1.2 Seqüència de tests per a administradors

Gran part de les proves fetes a la part d'usuari és igual a la de administradors

Prova	Resultat
Ocultar activitat visible	Es pot fer. ✓
Fer visible una activitat que té la data de visibilitat abans d'avui	No es pot fer. Es mostra un missatge conforme s'ha de canviar la data de visibilitat. ✓
De un dia per l'altre, la data de visibilitat és passa.	La pròxima vegada que un usuari es connecti, sense que aquest ho sàpiga, es farà una comprovació i una operació que farà canviar l'estat de l'activitat a oculta. D'aquesta forma aquest usuari i tots els que vindran després, no veuran l'activitat. ✓
Fer visible una activitat que té la data de visibilitat després d'avui	Es pot fer. ✓
Crear o modificar una activitat amb el mateix nom que altra	Es pot fer. ✓
Crear o modificar una activitat amb més de 10 anys entre la data de començament i de final.	L'activitat serà permanent. ✓
Crear o modificar una activitat sense entitats	No es pot, l'activitat requereix de al menys una entitat. ✓

Pujar un arxiu que no és una imatge com a imatge d'una activitat	No es pot ✓
Recuperar una activitat o entitat esborrada	No es pot ✓
Recuperar una foto esborrada	No es pot ✓
Assignar un link incorrecte d'una xarxa social o vídeo de Youtube	El link no porta a enlloc. La aplicació no falla, però la redirecció és incorrecte ✗
Assignar una foto i un vídeo de Youtube a una entitat.	Es veu el logo a la llista i només el vídeo als detalls de l'entitat. ✓
Assignar algunes xarxes socials, però no totes.	Es veuen només les assignades. ✓
Ocultar o filtrar entitats	No es pot ✓

## 8.2 Tests no funcionals

Una vegada testejadades les funcionalitats de l'aplicació, aquí s'analitzaran els aspectes relacionats amb els requisits no funcionals.

### 8.2.1 Test de memòria

Als requisits de memòria es van establir 50 MB de màxim per l'APK de l'aplicació, cosa que s'ha aconseguit ja que actualment té una mida de uns 10 MB. No hi haurà cap problema per pujar l'aplicació a Google Play i a més, podem dir que es trigarà molt poc en instal·lar l'aplicació. L'aplicació **una vegada instal·lada** passa a ocupar uns 50 MB, cosa que tampoc suposa cap problema.

Aquest límit també s'aplicarà a iOS.

### 8.2.2 Test de rendiment

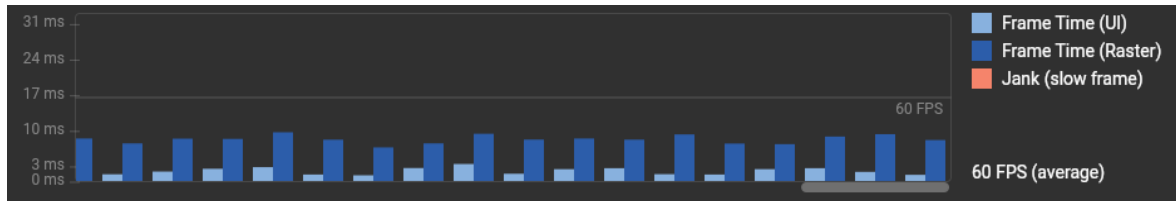
Flutter incorpora les seves pròpies eines per analitzar el rendiment gràfic, ús de memòria RAM, CPU, xarxa i mida de l'aplicació. Avanço que els resultats són molt satisfactoris.

Els tests s'han realitzat amb un mòbil de gama baixa, el Xiaomi Redmi 8 amb 3 gb de RAM, que es troba al mercat des de 2019 . S'ha cregut que aquesta ha sigut la millor referència pel tipus de telèfon més utilitzat a Espanya. A espanya Android té una quota de mercat del voltant del 90% (Android supera el 90% de cuota en España, sense data), es gasten de mitjana 228€ per un telèfon nou (Lejos de la gama alta: los españoles pagamos 228 euros de media en la compra de un smartphone, sense data) i es tarda de mitja uns 30 mesos en ser canviat (¿Cuánto tardamos los españoles en cambiar de móvil?, sense data). Per això el test que s'analitzarà és a aquest dispositiu.

Pel que fa al rendiment gràfic, s'ha aconseguit una mitja de 60 fotogrames per segon, cosa que porta fluïdesa durant l'ús de l'aplicació. El test s'ha fet mentre es tocaven diferents botons i s'ha aconseguit certa estabilitat en el temps de resposta. A la llegenda del gràfic inferior es pot veure que es tenen en compte 3 variables per mesurar el rendiment:

- **Frame Time:** temps d'execució del codi dart per presentar una pantalla
- **Frame time:** temps d'execució del codi gràfic per presentar una pantalla

- **Jank:** temps d'execució d'aquelles pantalles que tarden més de 16 ms en executar-se i construir-se. Quan es superen aquests 16 ms, no s'arriba als 60 fotogrames per segon.



**Figura 61.** Rendiment gràfic

Durant el temps que s'ha conduït el test, obrint i tancant diferents pantalles, no s'ha trobat cap pantalla que acabi al Jank. Això es tradueix en que el rendiment és satisfactori, però ara veurem que és a costa de l'ús de memòria.

Al gràfic inferior es pot veure que es fa un ús de la memòria bastant estable durant l'ús de l'aplicació, tot i que hi ha excepcions. Al principi s'utilitzen uns 200MB de memòria RAM total, quan encara no s'han descarregat les imatges i les dades de Firebase. Però conforme arriben les dades, augmenta considerablement fins a uns 400 MB. També, quan es reproduïx un vídeo de YouTube des de dins l'aplicació, l'ús de memòria puja a uns 500 MB.



**Figura 62.** Rendiment de memòria

El rendiment gràfic i l'ús de memòria són els paràmetres més importants a l'hora d'analitzar el comportament d'una aplicació. Segurament la utilització de molts recursos de memòria es tradueix en un millor rendiment.

Tot i així, també s'ha analitzat el l'ús de CPU en busca d'algun procés que tardés més del compte. El resultat també són satisfactoris, ja que no s'ha trobat cap anomalia.

Finalment, respecte a les crides en la xarxa, un aspecte interessant a analitzar és el temps que es tarda en descarregar una imatge. S'han fet els tests utilitzant la velocitat 4G. Sota aquestes circumstàncies, la majoria d'imatges tarden entre 200 i 300 ms en descarregar-se des de que es fa la petició al servidor. Un temps que es podria millorar amb l'ús d'imatges comprimides, però que realment no ho necessiten, sobretot si tenim en compte la imminent arribada de la xarxa 5G.

### 8.2.3 Test d'usabilitat

L'aplicació ha implementat els widgets de Material Design i ha seguit les directrius d'usabilitat, tal i com s'ha vist reflectit a les decisions preses durant el desenvolupament. Totes les versions de l'aplicació utilitzen aquest model.

Idealment, en un futur, seria recomanable implementar Flat Design a per la versió d'iOS.

### 8.2.4 Test de fiabilitat

L'aplicació no guarda dades sobre els usuaris tal i com s'esperava.

En canvi l'administrador utilitza el seu correu i contrasenya, que assegurin que cap usuari extern pugui accedir-hi. El correu electrònic i contrasenya estan assegurats pel EU General Data Protection Regulation (GDPR) del servei proporcionat per Firebase. (Privacy and Security in Firebase, sense data)

### 8.2.5 Test de portabilitat

Donat que s'han testejat totes les funcionalitats en les 3 plataformes del projecte, es conclou que l'aplicació funciona plenament en Android, iOS i Web.

També s'han fet tests en multitud de versions Android i hi ha versions antigues en que l'aplicació no s'engega. S'han realitzat tests en dispositius virtuals de les versions Android 4.1, 4.2, 4.3 (Jelly Bean) i 4.4 Kit Kat. En cap d'aquestes ha funcionat l'aplicació, encara que si es possible instal·lar-la.

Els tests als dispositius que utilitzen versions Android 5.0 Lollipop en davant, l'aplicació funciona, tot i que a un pitjor rendiment. S'ha fet una prova a un dispositiu LG Nexus 5 (LG Nexus 5, sense data) virtual. Com es pot veure a la imatge, la majoria de pantalles tenen un temps de resposta que impedeixen arribar als 60 fotogrames per segon i es queden a 49.

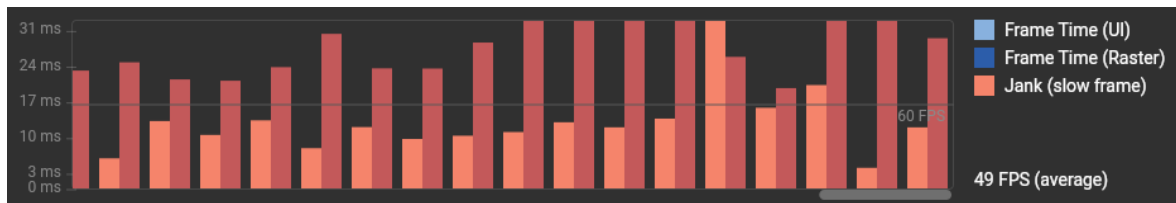


Figura 63. Rendiment gràfic Nexus 5

La mínima versió de iOS que soporta l'aplicació és iOS 10. El dispositiu més vell amb aquesta versió és l'iPhone 5, que porta al mercat des de 2012. S'ha provat amb el 5 i no ha funcionat, al menys des de un dispositiu virtual. Tot i així, el seu successor, el iPhone 5s sí. Tot i que no té el millor rendiment, si es pot instal·lar i utilitzar. També s'ha provat en més iPhones virtuals a partir del 5s i funcionen correctament.

La web ha sigut testejada a diferents navegadors (Chrome, Safari, Edge, Mozilla Firefox...) a diferents dispositius (telèfons, tabletas i ordinadors) i funciona correctament. Els elements en pantalla escalen correctament en tot tipus de dispositius des del navegador web.

## 9 Instal·lació i Manteniment

### 9.1 Configuració de l'entorn de desenvolupament

Per poder provar l'aplicació a partir del codi desenvolupat i crear noves versions de l'aplicació en un futur, s'han de fer una sèrie d'instal·lacions. El codi de l'aplicació és públic i és troba al [repositori de Github](#). (Santo, Repositori de Github amb el codi, sense data)

El millor és seguir el tutorial per instal·lar Flutter, Dart i Android Studio, disponible a [la pagina web de Flutter](#) (Install Flutter, sense data).

### 9.2 Instal·lació

#### 9.2.1 Instal·lació de Firebase

Per afegir dades a l'aplicació, s'utilitza la consola de Firebase. Aquesta consola té dos plans diferents, un gratuït (pla Spark) i un de pagament (pla Blaze). La informació complerta dels plans es troba a [la pagina web de Firebase](#). (Firebase Pricing, sense data) Aquí hi ha un resum:

- Si l'aplicació no té més de 300 usuaris diaris (sent conservadors), Spark hauria de ser suficient. En cas que aquest límit es superés, l'aplicació es bloquejaria fins el mes següent i això és un problema.
- Blaze necessita una targeta electrònica o compte bancari. Llavors, quan es supera el límit del pla gratuït, l'aplicació continuaria funcionant i es cobra segons l'ús de les dades. Aquí es pot veure [un exemple de quant costaria utilitzar aquest pla segons el nombre d'usuaris](#). (See a Cloud Firestore pricing example, sense data)

Per accedir a la consola de Firebase, s'han de donar permisos a la persona que vulgui accedir a través del seu correu electrònic de Google [d'aquesta forma](#). (Manage project members, sense data)

Des de la consola, es poden accedir als següents serveis utilitzats a l'aplicació:

- Authentication: aquí estan tots els administradors amb els seus correus i contrasenyes. Des d'aquí es poden afegir, deshabilitar o esborrar administradors. Les contrasenyes no es poden llegir ni canviar des d'aquí. Per canviar una contrasenya, s'ha d'enviar un correu a aquesta persona per a que la canviï. Per fer-ho, es clica a veure més (: ) sobre un administrador i es fa un reset de la contrasenya.
- Firestore: aquí es troben totes les dades en text.
- Storage: Aquí es troben les imatges de les activitats i entitats emmagatzemades.

#### 9.2.2 Instal·lació Web

Tal i com explica [la documentació de Flutter](#), per crear els fitxers de la pàgina web, bastaria amb executar la comanda flutter build web. Això genera els arxius de l'aplicació web dins del directori build/web del projecte. Per desplegar l'aplicació a un servidor es pot fer de dues formes. (Building the app for release, sense data)

##### 9.2.2.1 Firebase Hosting

Una opció de servidor que simplifica el treball és la de utilitzar el servei de Hosting de Firebase. No és una opció que surti econòmica, però que si simplifica el treball. Només s'han de fer una sèrie d'instal·lacions prèvies i un parell de comandes, [com es pot veure a aquest](#)

[tutorial](#) (Robledo, sense data). En cas que es vulgui també un domini propi, [es recomana seguir aquest tutorial](#) (Maan, sense data).

#### 9.2.2.2 Servidor propi

Seguint [aquest tutorial](#) (Abd-Alkrim, sense data), es podria desplegar l'aplicació web en un servidor dedicat, un VPS o un Shared Hosting amb NodeJs. A [la documentació de Flutter](#) (Building the app for release, sense data) suggereixen utilitzar també un Servidor Python o utilitzar [el paquet dhttp](#). (dhttpd, sense data)

#### 9.2.3 Instal·lació Android

Tota la informació sobre la creació de l'APK per a compartir sense que estigui a Google Play, es troba a [la documentació de Flutter](#) (Building the app for release, sense data). Per publicar l'aplicació a Google Play, s'ha de crear un compte de desenvolupador d'Android, que costa \$25, i publicar. Per més informació, recomano seguir [aquest tutorial](#) (Petra, sense data) i [la documentació de Android Developers](#). (Launch on Google Play, sense data)

#### 9.2.4 iOS

Per desplegar una aplicació a Apple App Store de iOS, el procés és més difícil i llarg. Es tracta en detall a [la documentació de Flutter](#). (Build and release an iOS app, sense data) En costos, publicar una aplicació són \$99.

### 9.3 Manteniment

El manteniment parteix de la instal·lació, una vegada ja s'ha desplegat l'aplicació a totes les plataformes i ja està operativa.

#### 9.3.1 Manteniment de Firebase

En cas que el codi de l'aplicació no es modifiqui, només s'hauria de fer un manteniment des de la consola de Firebase.

Primerament s'hauria de fer un control periòdic de la quantitat d'usuaris que es connecten a l'aplicació i els costos associats. Com he dit abans, el pla Blaze permet que es cobri segons la quantitat d'usuaris que es connecten a l'aplicació i s'hauria de controlar que no superés el pressupost de l'ajuntament. Des de la consola de Firebase es pot veure tota la informació dels usuaris diaris i el control de costos.

També s'ha de fer el manteniment dels serveis, alguns de manera puntual, altres periòdicament.

- Authentication: afegir administradors de forma puntual o canviar contrasenyes en cas que sigui necessari.
- Firestore: En cas que hi hagi algun problema, es pot fer una còpia de seguretat. S'ha de tenir el compte Blaze per poder fer còpies de seguretat periòdiques manualment. [Les instruccions es troben a la pàgina web de Firebase](#). (Export and import data, sense data)
- Storage: no es possible fer una còpia de seguretat del contingut a Firebase Storage. Si en un futur es volen fer còpies de seguretat, [es podrien fer utilitzant Google Cloud SDK, tal i com es descriu en aquest article](#). (Marcel, sense data)

### ***9.3.2 Modificacions en el codi***

En cas que es vulgui ampliar o solucionar un error de l'aplicació i no de la base de dades, s'hauria de modificar el codi. Aquestes modificacions comportarien tenir que reconstruir el projecte per actualitzar-lo a cadascuna de les plataformes.

## 10 Conclusions

Durant la carrera hi ha hagut molt de treball, però tinc la sensació de que aquest treball ha servit merament per a la meva educació. Han sigut hores de treball que no s'han materialitzat en cap millora cap a la societat ni en cap rendiment econòmic. Han sigut treballs que s'han fet i s'han guardat a un calaix. Per aquesta raó, jo volia que el meu treball de fi de grau es desmarqués d'aquest corrent i em vaig apuntar a fer aquest treball per poder crear una solució a un problema real d'algú. I no mentiré quan dic que m'agrada quan penso que un treball meu beneficia a la societat. Els projectes APS busquen que el treball de fi de grau es desmarqui d'aquest camí, donant-li un nou sentit al que fan els estudiants.

Durant el desenvolupament de l'aplicació, sempre hi havia algun problema a solucionar o alguna cosa a millorar. En el moment que estic escrivint aquesta documentació, el projecte encara no ha beneficiat a ningú i tinc moltes ganes de que això passi. La meva motivació per fer el projecte va ser aquest benefici comú. Han sigut uns mesos plens de treball i reunions periòdiques que ningú ha vist ni de les quals ningú s'ha beneficiat.

Tot i així, si trèiem la part més social de projecte, he de dir que els problemes als que m'he enfrontat han sigut suficientment interessants com per a que hagi merescut els mesos dedicats. L'aprenentatge d'una nova tecnologia que a més té una projecció de futur bastant prometedora segurament m'obrirà portes en el meu futur.

En realitat, aquesta tecnologia no es tan nova, ja que beu d'algunes de les que s'han utilitzat a varies assignatures de la carrera. El disseny de les pantalles de Interacció Persona Ordinador; enginyeria del software d'Anàlisi i Disseny d'Aplicacions; programació a smartphones d'Aplicacions Mòbils i Encastades; comunicació entre capa de presentació i bases de dades de Sistemes Oberts; entre d'altres. Es podria dir s'han ajuntat moltes idees d'altres tecnologies apreses durant els meus estudis i això m'ha permès crear aquesta aplicació.

També he après molt sobre la comunicació s'ha de tindre entre un enginyer i algú que no té tantes habilitats en aquest sentit. A vegades es crea una distancia molt gran entre el que una persona no experta vol d'una aplicació informàtica i el que l'enginyer pensa que es vol implementar. En alguns moments aquesta comunicació també ha sigut complicada, ja que si es volia programar una funcionalitat que depenia de la confirmació de l'ajuntament de Tortosa i aquest no responia en setmanes, el projecte s'endarrereixia. Com que aquest projecte ha sigut bastant llarg i ple de reunions periòdiques, aquestes dissonàncies s'han anat corregint sense molts problemes. Segurament hagués sigut molt diferent si s'hagués fet una aplicació directament a partir d'un enunciat.

Igual que espero que l'aplicació sigui utilitzada pels tortosins, també espero que això no acabi aquí. Una opció es aplicar-li la llicència de codi obert a aquest projecte per a que qualsevol pugui ampliar-lo o aplicar-ho a altres ajuntaments. Com que el codi s'ha escrit totalment en anglès, també pot tenir projecció fora del país. Una altra opció és ampliar el projecte en un treball de fi de grau d'un futur estudiant, de cara a millorar l'aplicació i el servei als futurs voluntaris.

## 10.1 Possibles ampliacions futures

Durant aquest projecte s'han recollit moltes formes de fer aquesta aplicació més completa. Des d'aquells cassos d'ús no implementats de la versió 2 ni 3, fins a aquells cassos d'ús que s'han considerat a posteriori.

Aquestes noves funcionalitats [s'han detallat a Github amb la forma d'un nou projecte](#). (htt)D'aquesta forma, un altre estudiant del grau podria fer un nou projecte per implementar aquestes ampliacions. Als issues del projecte de Github hi ha més detalls sobre com implementar cadascuna de les funcionalitats.

- **Ordenació d'activitats** segons algun criteri (data, nom, tipus...). L'actor podrà seleccionar quin es el criteri i la llista d'activitats s'ordenarà d'aquesta forma.
- **Enviar notificacions** a tots els usuaris sobre una activitat. L'usuari pot prémer la notificació per accedir als detalls de l'activitat.
- **Tutorial inicial:** la primera vegada que una persona utilitza l'aplicació en iOS o Android, s'ensenyarà un tutorial guiat.
- **Registre i identificació d'usuaris** per a que guardin les activitats preferides al seu compte. Permetre diferents formes d'inici de sessió (Google, Facebook, Twitter, correu electrònic...)
- **Implementar Flat Design per la versió iOS.** L'aplicació es veu molt bé a tots els dispositius, tot i que s'han seguit les directives de Material Design per a totes les versions. Per tal de que l'aplicació s'adapti a les directrius del medi d'iOS, l'aplicació hauria d'implementar l'estil Flat Design.
- **Localitzar les activitats en un mapa** d'on estan localitzades. Es podrien veure les activitats tant en una llista com al mapa.
- **Múltiples idiomes** per a que en qualsevol moment una persona pugui canviar l'idioma de l'aplicació.
- **Activitats i entitats favorites.** Permetre que els usuaris puguin seleccionar les seves activitats i entitats favorites per tenir-les més en compte i accessibles.
- **Imatges avançades:** permetre veure més d'una imatge per activitat i fer zoom sobre elles. Des del punt de vista de l'administrador, permetre que es puguin retallar les imatges.

## 11 Referències

- Recursos utilitzats (bibliografia, pàgines web, programari, hardware, etc...).
- (sense data). Recollit de <https://github.com/ZikBurns/voluntariat/projects/4>
- 2020 Developer Survey StackOverflow*. (sense data). Consultat el October / 2020, a <https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-languages-wanted>
- Abd-Alkrim, A. (sense data). *Deploy Flutter Web App in a Shared Hosting with NodeJs*. (Medium) Consultat el Abril / 2021, a <https://medium.com/firebits/deploy-flutter-web-app-in-a-shared-hosting-with-nodejs-5f7bc820b18d>
- Addict, P. (sense data). *Youtube: Searching a ListView with JSON data in Flutter - Programming Addict*. Recollit de <https://www.youtube.com/watch?v=EHT-QlGUoqY&ucbcb=1>
- amid, G. P. (sense data). (BBC) Consultat el November / 2020, a <https://www.bbc.com/news/technology-54919165>
- Android Developer Guides for APK Expansion Files*. (sense data). Consultat el Desembre / 2020, a <https://developer.android.com/google/play/expansion-files>
- Android supera el 90% de cuota en España*. (sense data). (Xataka mòvil) Consultat el Abril / 2021, a <https://www.xatakamovil.com/mercado/android-supera-90-cuota-espana-ios-cae-debajo-9-kantar>
- Announcing Flutter 2*. (sense data). (Flutter) Consultat el 3 / Març / 2021, a <https://developers.googleblog.com/2021/03/announcing-flutter-2.html>
- Baseflow*. (sense data). Consultat el Desembre / 2020, a <https://www.baseflow.com/>
- Build and release an iOS app*. (sense data). (Flutter) Consultat el Gener / 2021, a <https://flutter.dev/docs/deployment/ios>
- Building the app for release*. (sense data). (Flutter) Consultat el Desembre / 2020, a <https://flutter.dev/docs/deployment/web#building-the-app-for-release>
- Building the app for release*. (sense data). (Flutter) Consultat el Desembre / 2020, a <https://flutter.dev/docs/deployment/web#building-the-app-for-release>
- Building the app for release*. (sense data). (Flutter) Consultat el Desembre / 2020, a <https://flutter.dev/docs/deployment/android#building-the-app-for-release>
- Calero, J. (sense data). *StackOverflow: Is there a way to have a FloatingActionButton and a SpeedDial together?* Consultat el Desembre / 2020, a <https://stackoverflow.com/questions/65170712/is-there-a-way-to-have-a-floatingactionbutton-and-a-speeddial-together>
- Cercador de oportunitats de voluntariat de European Solidarity Corps*. (sense data). Consultat el Octubre / 2020, a [https://europa.eu/youth/go-abroad/volunteering/opportunities\\_en](https://europa.eu/youth/go-abroad/volunteering/opportunities_en)
- Choose a Database: Cloud Firestore or Realtime Database*. (sense data). Consultat el November / 2020, a <https://firebase.google.com/docs/database/rtdb-vs-firestore>
- Code, R. (sense data). *Carousel Slider Flutter - Flutter Tutorial 2021*. Consultat el Febrer / 2021, a [https://www.youtube.com/watch?v=5\\_RdsmUQkc4](https://www.youtube.com/watch?v=5_RdsmUQkc4)

*dhttpd*. (sense data). (Pub.dev) Consultat el Abril / 2021, a <https://pub.dev/packages/dhttpd>

*Didierboelens: Introduction to the notions of Streams, Bloc and Reactive Programming. Theory and practical examples*. (sense data). Consultat el November / 2020, a <https://www.didierboelens.com/2018/08/reactive-programming-streams-bloc/>

*Export and import data*. (sense data). (Firebase) Consultat el Abril / 2021, a <https://firebase.google.com/docs/firestore/manage-data/export-import>

Fernando, I. (sense data). *Medium: Flutter — Bottom Sheet*. Consultat el Febrer / 2021, a <https://medium.com/codechai/flutter-5-bottom-sheet-2d56bf9f3bc>

*Firebase Pricing*. (sense data). Consultat el October / 2020, a <https://firebase.google.com/pricing>

*Firebase Pricing plans*. (sense data). Consultat el October / 2020, a <https://firebase.google.com/pricing>

*Flutter Fade in images with a placeholder*. (sense data). Consultat el Desembre / 2020, a <https://flutter.dev/docs/cookbook/images/fading-in-images>

*Flutter for Android developers*. (sense data). Consultat el October / 2020, a <https://flutter.dev/docs/get-started/flutter-for/android-devs>

*Flutter issue #72171*. (sense data). Consultat el Desembre / 2020, a <https://github.com/flutter/flutter/issues/72171?web=1&wdLOR=c863EED3B-73C3-44E4-83C2-D60B18075CE6>

*Flutter issue #72461*. (sense data). Consultat el Desembre / 2020, a <https://github.com/flutter/flutter/issues/72461?web=1&wdLOR=c7BA10302-BE60-4FFB-8784-D1F49E310BCD>

*Flutter Web*. (sense data). Consultat el Octubre / 2020, a <https://flutter.dev/web>

*Flutter Web: firebase\_storage: MissingPluginException No implementation found for method StorageReference#putData*. (sense data). (Stack Overflow) Consultat el Desembre / 2020, a <https://stackoverflow.com/questions/63604931/flutter-web-firebase-storage-missingpluginexception-no-implementation-found-fo/63733238#63733238>

*FlutterFire*. (sense data). Consultat el November / 2020, a <https://firebase.flutter.dev/>

*font\_awesome\_flutter*. (sense data). (Pub.dev) Consultat el Març / 2021, a [https://pub.dev/packages/font\\_awesome\\_flutter](https://pub.dev/packages/font_awesome_flutter)

*Github FlutterFire Roadmap*. (sense data). Consultat el Novembre / 2020, a [https://github.com/flutter/flutter/projects/69?card\\_filter\\_query=label%3A%22severe%3A+new+feature%22&wdLOR=c0F8FB24C-F836-4279-B981-78B83179EB18](https://github.com/flutter/flutter/projects/69?card_filter_query=label%3A%22severe%3A+new+feature%22&wdLOR=c0F8FB24C-F836-4279-B981-78B83179EB18)

*Github Flutter-Tutorials by Cheetahcoding*. (sense data). Consultat el 2020 / November, a [https://github.com/cheetahcoding/Flutter-Tutorials/tree/firebase\\_series](https://github.com/cheetahcoding/Flutter-Tutorials/tree/firebase_series)

*Guidelines Material Design*. (sense data). Consultat el Desembre / 2020, a <https://material.io/design/guidelines-overview>

*image\_picker*. (sense data). (Pub.dev) Consultat el Desembre / 2020, a [https://pub.dev/packages/image\\_picker](https://pub.dev/packages/image_picker)

*Install Flutter*. (sense data). (Flutter) Consultat el Octubre / 2020, a <https://flutter.dev/docs/get-started/install>

- IshanFx. (sense data). *Github: flutter\_bottom\_sheet main.dart*. (Github) Consultat el Febrer / 2021, a [https://github.com/IshanFx/flutter\\_bottom\\_sheet/blob/master/lib/main.dart](https://github.com/IshanFx/flutter_bottom_sheet/blob/master/lib/main.dart)
- jazzbpn. (sense data). *StackOverflow: Programmatically Lighten or Darken a hex color in dart*. Consultat el Febrer / 2021, a <https://stackoverflow.com/questions/58360989/programmatically-lighten-or-darken-a-hex-color-in-dart>
- Launch on Google Play*. (sense data). (Android Developers) Consultat el Abril / 2021, a <https://developer.android.com/distribute/best-practices/launch>
- Lejos de la gama alta: los españoles pagamos 228 euros de media en la compra de un smartphone*. (sense data). (Xataka móvil) Consultat el Abril / 2021, a <https://www.xatakamovil.com/mercado/lejos-de-la-gama-alta-los-espanoles-pagamos-228-euros-de-media-en-la-compra-de-un-smartphone>
- LG Nexus 5*. (sense data). (GMSArena) Consultat el Abril / 2021, a [https://www.gsmarena.com/lg\\_nexus\\_5-5705.php](https://www.gsmarena.com/lg_nexus_5-5705.php)
- Lloc Web de Moviliza-T*. (sense data). Consultat el Octubre / 2020, a <https://www.movilizat.org/es>
- Maan, H. S. (sense data). *Deploying Flutter Web App to Firebase With a Custom Domain*. (Medium) Consultat el Abril / 2021, a <https://medium.com/geekculture/deploying-flutter-web-app-to-firebase-with-a-custom-domain-cb1da7337cf1>
- Manage project members*. (sense data). (Firebase) Consultat el Abril / 2021, a <https://support.google.com/firebase/answer/7000272?hl=en>
- Marcel, P. (sense data). *How to back-up Firebase Firestore and Firebase storage bucket*. (Medium) Consultat el Abril / 2021, a <https://medium.com/scalp/how-to-back-up-firebase-firestore-and-firebase-storage-bucket-b6d8dbc0cd7c>
- Mary. (sense data). *How do you adjust the height and borderRadius of a BottomSheet in Flutter?* (Stack Overflow) Consultat el Febrer / 2021, a <https://stackoverflow.com/questions/48968176/how-do-you-adjust-the-height-and-borderradius-of-a-bottomsheet-in-flutter>
- Material Design for iOS*. (sense data). Consultat el Desembre / 2020, a <https://material.io/develop/ios>
- Material Design for Web*. (sense data). Consultat el Desembre / 2020, a <https://material.io/develop/web>
- Material Design Speed Dial Guidelines*. (sense data). Consultat el Desembre / 2020, a <https://material.io/components/buttons-floating-action-button#types-of-transitions>
- Medium: Flutter State Management: BLoC Pattern*. (sense data). Consultat el November / 2020, a <https://medium.com/@aaron.chu/flutter-state-management-bloc-pattern-9cd6011c699#:~:text=If%20you%20know%20about%20the,no%20matter%20local%20or%20remote>
- Moros, M. (sense data). *Flutter Web: firebase\_storage: MissingPluginException No implementation found for method StorageReference#putData*. (Stack Overflow) Consultat el Desembre / 2020, a <https://stackoverflow.com/questions/63604931/flutter-web-firebase-storage-missingpluginexception-no-implementation-found-fo/63733238#63733238>

*Navigation Drawer Material Design.* (sense data). Consultat el Octubre / 2020, a <https://material.io/components/navigation-drawer>

Ninja, T. N. (sense data). *Youtube Flutter & Firebase App Tutorial #10- Sign In & Register Forms.* Recollit de [https://www.youtube.com/watch?app=desktop&v=\\_SHssHJJhAI&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&index=10&ucbcb=1](https://www.youtube.com/watch?app=desktop&v=_SHssHJJhAI&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&index=10&ucbcb=1)

*¿Cuánto tardamos los españoles en cambiar de móvil?* (sense data). (Informacion.es) Consultat el Abril / 2021, a <https://www.informacion.es/vida-y-estilo/tecnologia/2018/01/25/tardamos-espanoles-cambiar-movil-5822349.html>

*Online Visual Paradigm.* (sense data). Consultat el Desembre / 2020, a <https://online.visual-paradigm.com/>

*permission\_handler.* (sense data). (Pub.dev) Consultat el Decembre / 2020, a [https://pub.dev/packages/permission\\_handler](https://pub.dev/packages/permission_handler)

Petra, T. (sense data). *Deploying Flutter App To Playstore.* (Youtube) Consultat el Abril / 2021, a <https://www.youtube.com/watch?app=desktop&v=NFTaWkUtD34&ucbcb=1>

Pritish. (sense data). *StackOverflow: How to make FadeInImage Circular?* Consultat el Desembre / 2020, a <https://stackoverflow.com/questions/54059519/how-to-make-fadeinimage-circular>

*Privacy and Security in Firebase.* (sense data). Consultat el Abril / 2021, a <https://firebase.google.com/support/privacy>

*Pub.dev cloud\_firestore.* (sense data). Consultat el 2020 / Novembre, a [https://pub.dev/packages/cloud\\_firestore](https://pub.dev/packages/cloud_firestore)

*Pub.dev firebase\_auth.* (sense data). Consultat el 2020 / Novembre, a [https://pub.dev/packages/firebase\\_auth](https://pub.dev/packages/firebase_auth)

*Pub.dev firebase\_storage.* (sense data). Consultat el Novembre / 2020, a [https://pub.dev/packages/firebase\\_storage](https://pub.dev/packages/firebase_storage)

*Pub.dev flutter\_form\_builder.* (sense data). Consultat el Novembre / 2020, a [https://pub.dev/packages/flutter\\_form\\_builder](https://pub.dev/packages/flutter_form_builder)

*Pub.dev flutter\_search\_bar.* (sense data). Consultat el Novembre / 2020, a [https://pub.dev/packages/flutter\\_search\\_bar](https://pub.dev/packages/flutter_search_bar)

*Pub.dev flutter\_speed\_dial.* (sense data). Consultat el Desembre / 2020, a [https://pub.dev/packages/flutter\\_speed\\_dial](https://pub.dev/packages/flutter_speed_dial)

*Pub.dev linkable.* (sense data). Consultat el November / 2020, a <https://pub.dev/packages/linkable>

Rana, V. (sense data). *Medium: Change Application Name and Icon in Flutter project(Android and iOS).* Consultat el Desembre / 2020, a <https://medium.com/@vaibhavi.rana99/change-application-name-and-icon-in-flutter-bebbec297c57>

*Repositories Ranking Gitstar.* (sense data). Consultat el Desembre / 2020, a <https://gitstar-ranking.com/repositories>

- Revolut Playstore.* (sense data). Consultat el Gener / 2021, a <https://play.google.com/store/apps/details?id=com.revolut.revolut&hl=en&gl=US>
- Robledo, A. (sense data). *Use Firebase to host your Flutter app on the web.* (Medium) Consultat el Desembre / 2020, a <https://medium.com/flutter/must-try-use-firebase-to-host-your-flutter-app-on-the-web-852ee533a469>
- Santo, J. C. (sense data). *Error uploading a photo to Firebase storage with Flutter Web using Pickers.* Consultat el Desembre / 2020, a <https://stackoverflow.com/questions/65201535/error-uploading-a-photo-to-firebase-storage-with-flutter-web-using-pickers>
- Santo, J. C. (sense data). *Repositori de Github amb el codi.* Consultat el Abril / 2021, a <https://github.com/ZikBurns/voluntariat>
- See a Cloud Firestore pricing example.* (sense data). Consultat el October / 2020, a <https://firebase.google.com/docs/firestore/billing-example>
- StartUpsVentureCapital: Firebase Costs Increased by 7,000%!* (sense data). Consultat el November / 2020, a <https://startupsventurecapital.com/firebase-costs-increased-by-7-000-81dc0a27271d>
- Statista: Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020.* (sense data). Consultat el October / 2020, a <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- Studio, R. (sense data). *Youtube: Flutter - Creating Pop-up Dialog in Flutter! (Beginners).* Consultat el Novembre / 2020, a <https://www.youtube.com/watch?app=desktop&v=FGfhnS6skMQ&ucbcb=1>
- SyntacOps. (sense data). *Youtube: Flutter Forms Made Easy With Flutter FormBuilder Package (Outdated).* Consultat el Novembre / 2020, a <https://www.youtube.com/watch?v=7FBELQq808M&t=6s>
- Tech, W. B. (sense data). Consultat el December / 2020, a [https://en.wikipedia.org/wiki/Big\\_Tech](https://en.wikipedia.org/wiki/Big_Tech)
- user6274128. (sense data). *How to change Text color depending on background so that there is good contrast between the two colors.* (Stack Overflow) Consultat el Març / 2021, a <https://stackoverflow.com/questions/58432285/how-to-change-text-color-depending-on-background-so-that-there-is-good-contrast>
- Wallapop Playstore.* (sense data). Consultat el Gener / 2021, a <https://play.google.com/store/apps/details?id=com.wallapop&hl=en&gl=US>
- Way, T. F. (sense data). *Youtube: User Profile - Complete E-Commerce App Flutter UI - Ep 7 (The final one) - Speed Code.* Consultat el Febrer / 2021, a <https://www.youtube.com/watch?app=desktop&v=sOVgPx8ljaE&ucbcb=1>
- Web support for Flutter.* (sense data). Consultat el November / 2020, a <https://flutter.dev/web>
- Wikigain How to Install macOS Catalina on VMware on Windows 10 – PC.* (sense data). Consultat el Desembre / 2020, a <https://www.wikigain.com/install-macos-catalina-on-vmware-on-windows/>

*Xiaomi Redmi 8.* (sense data). (GMSArena) Consultat el Abril / 2021, a [https://www.gsmarena.com/xiaomi\\_redmi\\_8-9800.php](https://www.gsmarena.com/xiaomi_redmi_8-9800.php)

*Yahoo Finance PlayStore.* (sense data). Consultat el Gener / 2021, a <https://play.google.com/store/apps/details?id=com.yahoo.mobile.client.android.finance&hl=en&gl=US>

*Youtube Flutter Blog App Using Firestore - Build ios & Android App by Master Coding.* (sense data). Consultat el November / 2020, a [https://www.youtube.com/watch?app=desktop&v=\\_ed4783G5yY&ucbcb=1](https://www.youtube.com/watch?app=desktop&v=_ed4783G5yY&ucbcb=1)

*YouTube Player API Reference for iframe Embeds.* (sense data). Consultat el Març / 2021, a [https://developers.google.com/youtube/iframe\\_api\\_reference](https://developers.google.com/youtube/iframe_api_reference)

*Youtube: Flutter & Firebase App Build by The Net Ninja.* (sense data). Consultat el November / 2020, a <https://www.youtube.com/watch?v=sfA3NWDBPZ4&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&index=2>

*Youtube: Flutter and Firebase by Santos Enoque.* (sense data). Consultat el October / 2020, a [https://www.youtube.com/watch?app=desktop&v=StSZSUEAEgc&list=PLmnT6naTGy2R2n53EzzUYzpnONE0bHep\\_&ucbcb=1](https://www.youtube.com/watch?app=desktop&v=StSZSUEAEgc&list=PLmnT6naTGy2R2n53EzzUYzpnONE0bHep_&ucbcb=1)

*Youtube: Flutter Tutorial - Carousel Slider - Loop & Infinite ListView.* (sense data). Consultat el Febrer / 2021, a <https://www.youtube.com/watch?app=desktop&v=SQnmuulYWPU&ucbcb=1>

*Youtube: Flutter Tutorial for Beginners by The Net Ninja.* (sense data). Recollit de <https://www.youtube.com/watch?app=desktop&v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ&ucbcb=1>

*Youtube: The Net Ninja.* (sense data). Consultat el October / 2020, a <https://www.youtube.com/channel/UCW5YeuERMmlnqo4oq8vwUpg>

*youtube\_player\_iframe.* (sense data). (Pub.dev) Recollit de [https://pub.dev/packages/youtube\\_player\\_iframe](https://pub.dev/packages/youtube_player_iframe)