

**FIB**



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# Aplicació mòbil per a l'organització dels voluntaris de APAN

*Autor:* Ignasi Escolà Piñol

*Directora:* Àlicia Ageno

*Especialitat:* Computació

*Data:* 14/01/2019

# Índex

Índex	2
1. Introducció	5
1.1 Introducció al problema	5
1.2 Contextualització	5
1.3 Actors implicats	6
1.3.1 Desenvolupador del algoritme i disseny de l'aplicació.	6
1.3.2 Beta Tester	6
1.3.3 Usuaris	6
2. Estudi previ i gestió del projecte	7
2.1 Estat de l'art	7
2.1.1 Forward checking	8
2.1.2 Minimal forward checking	8
2.2 Abast	12
2.2.1 Formulació del problema: distribució dels gossos i restriccions necessàries.	12
2.2.2 Objectius i altres requeriments	15
2.2.3 Obstacles	15
<i>Data entrega</i>	15
<i>Poca experiència a l'hora de desenvolupar aplicacions per dispositius mòbils.</i>	15
<i>Poca experiència en la creació d'interfícies d'usuari.</i>	16
<i>Aparició de canvis i/o noves funcionalitats.</i>	16
2.3 Metodologia i rigor	17
2.3.1 Mètodes de treball: Cicles desenvolupament + Feedback del director	17
2.3.2 Eines de seguiment	17
2.3.3 Mètodes de validació	17
2.4 Planificació temporal	18
2.4.1 Definició de les tasques	18
2.4.1.1 Anàlisi del projecte	18
2.4.1.2 Disseny del projecte	18
2.4.1.3 Desenvolupament del projecte	19
2.4.1.4 Fita final	19
2.4.1.5 Descripció de les tasques	20
2.4.1.6 Duració de les tasques	22
2.4.1.7 Dependències	23
2.4.1.8 Diagrama de Gantt	25
2.4.1.9 Recursos	27
Hardware	27
Software	27
2.4.2 Valoració d'alternatives i pla d'acció	28

2.5. Gestió Econòmica	30
2.5.1 Identificació i estimació dels costos	30
2.5.1.1 Pressupost de recursos humans	30
2.5.1.2 Pressupost del <i>hardware</i>	33
2.5.1.3 Pressupost del software	34
2.5.1.4 Despeses indirectes	34
2.5.1.5 Despeses imprevistes	35
2.5.1.6 Cost total del projecte	35
2.5.2 Control de gestió	36
2.5.3 Informe de sostenibilitat	37
2.5.3.1 Dimensió econòmica	37
2.5.3.2 Dimensió social	37
2.5.3.3 Dimensió ambiental	38
2.5.4. Autoavaluació del domini actual de la competència de sostenibilitat	38
3. Anàlisi de requisits	40
3.1 Requisits funcionals	40
3.1.1 Gestió gossos	40
3.1.2 Gestió voluntaris	43
3.1.3 Gestió distribució gossos	46
3.1.4 Gestió llistes	47
3.2 Requisits no funcionals	50
3.2.1 Requisits d'usabilitat	50
3.2.2 Requisits de rendiment	50
3.2.3 Requisits de manteniment i suport	50
3.2.4 Requisits legals	51
4. Disseny aplicació	52
4.1 Capa de Vistes	53
4.2 Capa de Presentador	69
4.3 Capa de Model	70
5. Algoritme	73
5.1 Descripció CSP	73
5.2 Alternatives	74
5.3 Solució adoptada	76
5.3.1 Variables	76
5.3.2 Dominis	76
5.3.3 Restriccions	77
5.3.4 Estructures	78
5.4 Pseudocodi	79
5.5 Problemes trobats	85
6. Disseny proves	87
7. Planificació, anàlisi de costos i sostenibilitat finals	91

7.1 Modificacions realitzades a la planificació	91
7.2 Metodologia final	93
7.3 Anàlisi econòmic final	93
8.Conclusions	98
8.1 Assoliment del objectius	98
8.2 Futures extensions	99
8.3 Lleis i regulacions	101
8.4 Justificació de les competències tècniques	101
8.5 Integració de coneixements	102
8.6 Valoració personal	103
9.Bibliografia	104
10.Annex	105
10.1 Manual de l'usuari	105
Pantalla inicial	105
Gestió gossos	106
Crear gos	107
Consultar gos	109
Editar gos	110
Gestió voluntaris	111
Crear voluntari	112
Consultar voluntari	113
Editar voluntari	114
Gestió distribució	115
Consultar distribució	116
Editar distribució	117
Gestió llistes	118
Configurar llista	119
Configurar passejos	120
Editar configuració	121
Crear llista/ Veure última llista	122
Editar llista	123

# 1. Introducció

## 1.1 Introducció al problema

Des de fa cosa d'un any i mig sóc voluntari a la protectora d'animals APAN (Òdena, Igualada) [1]. La feina dels voluntaris del cap de setmana es basa principalment a treure a passejar els gossos de la protectora, ja que durant la setmana no hi ha suficients mans per fer-ho. A part, s'intenta dedicar una estona als gossos més porucs per fer teràpia. Per tant, per aprofitar al màxim les hores que els voluntaris estan a la protectora hi ha d'haver algú que organitza els gossos que surten a passejar. L'objectiu és treure a passejar el màxim de gossos possibles. Ens dividim en 2 grups, el grup de dissabtes i el grup de diumenges, i s'intenta que cada gos surti entre 20-30 minuts tant dissabte com diumenge però no sempre és possible aconseguir que surtin tots a passejar.

Amb els voluntaris disponibles es fan llistes per organitzar i repartir qui ha de treure cada gos i en quin ordre però hi ha altres paràmetres a tenir en compte i el responsable de fer les llistes ha de dedicar fins a 2 hores cada setmana per tal de fer unes llistes que compleixin tots els requisits que fan falta.

## 1.2 Contextualització

La idea d'aquest projecte és crear una aplicació Android que a partir de:

1. La informació dels gossos
2. La llista dels voluntaris que assistiran i la informació d'aquests
3. La informació sobre a quines gàbies es troba cada gos

que introdueixi l'usuari permeti crear llistes de forma fàcil i dinàmica. Es farà servir un algorisme de satisfacció de restriccions per tal d'aconseguir que es compleixin totes les condicions que es tenen en compte actualment, com per exemple que tots els gossos d'una mateixa gàbia surten durant el mateix passeig, que hi ha gossos de diferents gàbies que no poden coincidir durant el mateix passeig, tenir en compte les preferències dels voluntaris sempre que sigui possible. Aquestes s'expliquen de forma més detallada en l'apartat on es formula el problema A part d'estalviar temps als voluntaris que fan les llistes cada setmana, agilitzarà el procés de refer les llistes a causa dels imprevistos que sempre apareixen a última hora.

## 1.3 Actors implicats

### 1.3.1 Desenvolupador del algoritme i disseny de l'aplicació.

Seré jo mateix l'encarregat de les tasques de desenvolupament i disseny, ja que sóc l'única persona encarregada d'aquest projecte.

### 1.3.2 Beta Tester

Les tasques de beta tester seran realitzades per mi mateix amb l'ajuda d'alguns dels voluntaris que solen encarregar-se de fer les llistes. Com a futurs usuaris em reportaran els aspectes a millorar i/o arreglar de l'aplicació.

### 1.3.3 Usuaris

Tots els voluntaris de la protectora seran els futurs usuaris, ja que a part de la funcionalitat de la creació de llistes per tal d'organitzar els passejos, a partir de la informació que es necessita per la creació d'aquestes es pot crear un mapa general de la distribució dels gossos a la protectora per tal de saber de forma ràpida on es troben els gossos que s'han de treure a passejar i aspectes a tenir en compte per cada gos en particular.

## 2. Estudi previ i gestió del projecte

### 2.1 Estat de l'art

El principal objectiu de fer les llistes és aconseguir que surtin a passejar el màxim de gossos possibles, ja que parlem de gossos que surten a passejar 30 minuts els dissabtes i uns altres 30 els diumenges. La resta de dies de la setmana alguns d'ells, els que estan a les gàbies més petites, surten 10-15 minuts a una zona comuna mentre se'ls hi neteja la gàbia. Per tant, el principal objectiu dels voluntaris es satisfer les necessitats físiques dels gossos encara que siguin només 2 dies a la setmana.

Pel que fa a altres necessitats com, per exemple, les necessitats emocionals i socials, la falta de voluntaris fa que sigui pràcticament impossible de treballar. Les protectores d'animals són llocs on els animals solen estar molt estressats, nerviosos. Per tant, una de les normes bàsiques pels voluntaris durant el passeig és que els gossos de diferents gàbies s'apropin el mínim possible per evitar enfrontaments, sempre es deixa un distància de seguretat d'uns 5 metres. En canvi, amb els gossos porucs que no surten a passejar o que no es deixen tocar si que s'hi treballa una vegada s'acaben els passejos per tal que perdin les pors i poder treure'ls a passejar en un futur i així tenir l'opció de que siguin adoptats.

L'algorisme que s'ha de desenvolupar és un problema de satisfacció de restriccions (CSP). Aquests tipus d'algorismes es basen en que a partir d'un estat inicial i unes restriccions s'arriba a una possible solució en el cas que n'hi hagi alguna. Els algorismes més bàsics es dediquen a buscar una solució provant totes les combinacions amb tots els valors possibles fins que s'obté una solució vàlida. Els algorismes més complexos intenten no explorar totes les solucions, detectant a partir d'una solució parcial tots els valors incompatibles que ja no és necessari provar o eliminant des de bon principi valors que mai poden trobar-se en la solució final.

L'algorisme més bàsic a l'hora de resoldre aquests tipus de problemes és el backtracking cronològic simple [2]. Aquest algorisme és el que en el pitjor cas comprova totes les combinacions amb tots els valors possibles. Simplement el que fa, és assignar un valor i quan arriba a una solució parcial on no es compleixen totes les

restriccions es fa un pas endarrere i es prova un altre valor entre els valors possibles. Tot i arribar a una solució vàlida el procés per trobar-la és bastant ineficient.

Dels algorismes que ja fan una poda per no analitzar totes les combinacions possibles tenim el Forward Checking [3] i el Minimal Forward Checking [4]. Definicions bàsiques que cal saber per tal d'entendre algorismes SCP:

- Problema: està format per un conjunt de variables.
- Domini: cada variable té associat un domini. Aquest domini representen tots els valors possibles que pot prendre la variable.
- Restriccions: restringeix els valors que les variables poden prendre simultàniament. La solució ha de complir totes les restriccions.
- Solució: és el resultat que s'obté quan cada variable del problema se li ha instanciat un valor i aquests valors compleixen totes i cadascuna de les restriccions.

### 2.1.1 Forward checking

En cada etapa de la búsqueda, es comprova a partir de la variable actual, tots els valors de les futures variables que no compleixen les restriccions definides pel problema. Els valors de les futures variables que són inconsistents amb l'assignació actual seran temporalment eliminades dels dominis pertinents. Si el domini d'una variable futura es queda buida, la instanciació de la variable actual es desfà i es prova amb un valor nou. Si cap dels valors es consistent llavors és quan es fa servir el backtracking cronològic.

### 2.1.2 Minimal forward checking

L'algoritme del Minimal forward checking és una versió del Forward Checking que retarda el dur a terme totes les comprovacions de consistències que s'han explicat en el punt anterior fins que és absolutament necessari. En comptes de comprovar les inconsistències de les variables futures a partir de la instanciació de la variable actual, aquest algoritme només fa comprovacions fins que troba el primer valor que és consistent. Després, si l'algoritme retrocedeix llavors es fan la resta de comprovacions amb la resta de valors que encara no s'han comprovat. D'aquesta



manera el número de comprovacions que fa el Minimal forward checking és el mateix que en el forward checking en el pitjor cas.

A part, hi ha altres algorismes que són capaços de fer una poda de dominis eliminant valors no només en el nivell actual de búsqueda, també a la resta de nivells com és el cas del Extended Forward Checking.

A més a més, per resoldre problemes de satisfacció de restriccions hi ha algorismes totalment diferents que busquen una solució a partir de la idea dels conflictes.

En el nostre problema l'eficiència del algorisme no és la màxima prioritat, només ens cal trobar una solució vàlida. Durant el desenvolupament del projecte es començarà implementant un backtracking cronològic simple. Quan ja tinguem una versió funcional s'analitzarà l'estat del projecte i en cas de què es disposi del temps suficient s'optarà per desenvolupar un d'aquests dos algorismes més eficients. Així a l'hora de crear les llistes més complexes (per quantitat de gossos, quantitat de voluntaris ...) l'usuari no haurà d'esperar tant per obtenir la solució.

Pel que fa a tecnologies ja existents que resolen problemes de satisfacció de restriccions és molt fàcil trobar softwares a la xarxa que tenen com a objectiu la gestió/creació d'horaris o l'administració de tasques. Tot i resoldre aquests tipus de problemes de formes similars cada problema és diferent a la resta degut a la definició del problema i les restriccions que s'hi tenen en compte. Alguns d'aquests softwares més utilitzats són *Schedule Perfect* [5] i *Monday* [6]

Schedule Perfect: software fet servir per gestió/creació d'horaris. Si analitzem el producte podem veure que es tracta d'un software on els punts a destacar serien:

- Flexible: permet crear horaris seguint unes restriccions globals que sempre s'han de complir i unes restriccions que es poden adaptar segons la disponibilitat de l'empleat, les preferències de torn, ubicació ...
- Fet a mida: Treballa al costat del client per tal que es compleixin els requisits del dia a dia
- Estalvia temps mitjançant l'algorisme que crea els horaris

Aquest software s'apropa força al que es vol aconseguir en aquest projecte. En el nostre cas l'algoritme que es desenvoluparà serà flexible ja que tindrà restriccions obligatòries i d'altres que es compliran sempre que es puguin, com les preferències dels voluntaris envers els gossos. En el nostre cas no es farà un algoritme que posteriorment es pugui adaptar a qualsevol problema, sinó que l'algoritme que es desenvoluparà sempre treballarà en el mateix problema.

Monday: software utilitzat per l'administració de tasques. El que es pot destacar d'aquest software és la interfície d'usuari. És fàcil d'entendre, flexible i customitzable. El problema que es tracta en aquest software no s'assembla gaire en el que es necessita en aquest projecte i després d'analitzar les diferents funcionalitats hem vist que el més interessant que podem extreure és la interfície d'usuari que fan servir

Aquestes eines ens poden servir principalment per veure el tipus d'interfícies d'usuari, analitzar els punts forts i febles i fer servir els punts forts a l'hora de dissenyar la interfície d'usuari de la nostra aplicació

El que sí podem trobar i pot ser de molta utilitat són diferents eines open-source anomenades com 'solucionadors de restriccions' (*constraints solver*). Algunes d'elles:

Minion: és un solucionador de restriccions molt potent que s'adapta al tamany del problema encara que aquest sigui molt gran. La diferència entre Minion[8] i alguns dels softwares que s'hi assembles és que aquests segons per tal de proporcionar més opcions de modelat i de resolució fan que cada vegada siguin més complexos en usabilitat i pitjors en rendiment. Sembla un software interessant al que profunditzar per veure fins a quin punt ens pot ajudar durant el desenvolupament del projecte.

OptaPlanner: és un solucionador de restriccions [9]. Serveix per una gran quantitat de problemes com planificació de recursos empresarials, la classificació d'empleats, assignació de tasques, programació de tasques, entre d'altres. A partir d'un conjunt limitat de recursos limitats (empleats, temps, diners) s'ofereixen plannings més eficients per millorar la qualitat del servei i reduir costos. En aquests cas, aquest software sembla més encarat a optimitzar una solució inicial que no pas a crear una solució des d'un escenari buit.

Després de veure les necessitats dels gossos de la protectora a tenir en compte a l'hora de crear l'algoritme per tal de crear les llistes, d'analitzar alguns dels softwares del mercat que treballen en productes que s'assemblen en cert punt al que volem fer i de fer una ullada als diferents solucionadors de restriccions que hem trobat rec que podem fer servir aquesta informació per crear un aplicació mòbil nova amb l'objectiu de crear les llistes dels gossos.

Aquesta aplicació serà feta a mida pel problema actual que es defineix en el següent punt. També serà una eina flexible ja que es podrà treballar en diferents restriccions, unes globals que sempre s'hauran de complir i unes altres que el seu compliment dependrà de la situació. Tot això per tal de complir l'objectiu principal de l'aplicació, que és el d'estalviar temps als futurs usuaris per tal que crear les llistes d'aquesta forma permeti que no s'hagi de malgastar temps cada setmana per fer aquest mateix procés manual. De l'anàlisi que hem durant aquest apartat, extraurem les idees més interessants dels 2 softwares del mercat dels que hem parlat i donarem una ullada més endavant als softwares open-source per tal de veure si ens poden facilitar la feina durant el desenvolupament del projecte.

## 2.2 Abast

### 2.2.1 Formulació del problema: distribució dels gossos i restriccions necessàries.

El problema que se'ns presenta és un problema de satisfacció de restriccions. A partir d'un estat inicial i un conjunt de restriccions hem d'arribar a una solució (no és una solució única) que compleixi amb totes les restriccions definides.

En aquest projecte el problema que es vol solucionar és que a partir de les dades dels gossos de la protectora, la seva distribució a les gàbies i els voluntaris que han confirmat la seva assistència es creï una llista on s'indiqui els gossos que ha de treure a passejar cada voluntari i l'ordre en què ho ha de fer.

La protectora consta de 3 zones separades on hi ha gàbies amb gossos:

**Quarantes:** Gàbies petites amb els gossos que fa poc que són a la protectora. Per falta de lloc en aquestes gàbies sempre hi ha algun gos que porta més temps a la protectora que la resta i ja se'l pot treure a passejar.

**Patis:** Recintes grans amb 3 o més gossos que, si hi ha suficients voluntaris, també se'ls treu a passejar.

**Xenils:** Gàbies petites amb 1 o 2 gossos. Són els gossos que es prioritzen a l'hora de sortir a passejar a causa de la mida de les gàbies. Cada dia, un dels voluntaris que va a la protectora no passeja gossos i s'encarrega de netejar les gàbies d'aquesta zona. S'aprofita quan els gossos estan passejant per netejar les seves gàbies. Com es pot veure en l'esquema de sota, aquestes gàbies comuniquen amb una zona comuna també tancada. Per aquesta zona es deixen sortir els gossos que no es deixen tocar i/o no es poden treure a passejar. La llista resultant també ha d'indicar els gossos que han de sortir per aquesta zona interior i en quin ordre s'ha de fer.



Figura 1. Esquema de la distribució de les gàbies de xenils

A l'hora de fer una llista, els paràmetres que es tenen en compte actualment, són els següents:

1. Per cada passeig, el voluntari que neteja ha de tenir temps suficient per netejar les gàbies dels gossos que estan passejant i dels que surten per la zona interior. Si hi ha voluntaris suficients, s'aprofita per treure gossos de quarantenes o patis.
2. Hi ha gàbies on conviuen gossos que poden sortir a passejar amb gossos que surten per la zona interior, per tant, s'ha de tenir en compte que estiguin fora de la gàbia durant el mateix passeig.
3. Hi ha gossos dels que només surten per la zona interior que es porten bé entre ells tot i conviure en gàbies separades i se'ls fa coincidir perquè puguin interactuar. Per tant, s'ha de complir el punt 2 per cada un d'ells.
4. Hi ha gossos que no es porten bé entre ells i no poden coincidir durant en el mateix passeig. Això es fa per evitar problemes i depenent del tamany i la força del gos és arriscat.
5. Cada gos té la seva història. Hi ha un percentatge bastant petit dels gossos que surten a passejar que no poden sortir amb qualsevol persona. Són gossos

que necessiten un nivell de confiança elevat amb la persona que els passeja. A l'hora d'assignar aquests tipus de gossos s'ha de tenir clar els únics voluntaris que se'n poden encarregar.

6. Si en una gàbia hi ha més d'un gos que surt a passejar, llavors surten tots en el mateix passeig.
7. Cada voluntari té unes preferències/gustos. Hi ha gossos que agraden més o hi ha gossos que fan més respecte. S'intenta que cada voluntari tregui a passejar els gossos que més li agraden tot i que a vegades no és possible.

A part, com la quantitat de gossos per gàbia varia, és difícil tenir en compte tots els punts anteriors i que a part quadrin els números, és a dir, que surtin el màxim de gossos possibles i que cada voluntari faci els mateixos passejos que la resta, etc.

Tenir tot això en compte a l'hora de fer una d'aquestes llistes manualment provoca que la persona que la fa ha de perdre molt de temps. Es podria pensar en tenir una llista base o seguir un patró però les llistes resultants mai s'assemblen d'una setmana a una altra ja que:

1. Els voluntaris no sempre poden venir, per tant, cada setmana son combinacions de gent diferent i la quantitat de voluntaris també varia.
2. El voluntari que li toca netejar es decideix de forma rotativa a no ser que algú demani expressament que vol netejar.
3. Cada setmana entren i surten gossos.
4. Els gossos no sempre estan a la mateixa gàbia. Com la quantitat de gàbies no és suficient per tots els gossos que hi ha s'intenta posar en la mateixa gàbia gossos que estiguin vivint sols o s'intenta fer grups grans de gossos en la zona de patis.

L'altra gran problema que ens trobem quasi cada cap de setmana es que sempre hi ha canvis d'última hora:

1. Gent que falla a última hora
2. Algú fa tard i no pot fer tots els passejos
3. Algun dels voluntaris ha d'ajudar amb una tasca de la protectora més urgent.
4. Venen possibles adoptants i treuen a passejar algun dels gossos i coincideix amb el que havia de treure un voluntari.

## 2.2.2 Objectius i altres requeriments

L'abast del projecte és definir, dissenyar i implementar una aplicació mòbil que permeti crear llistes per tal d'organitzar els passejos que han de fer el voluntaris a partir de les dades i aplicant les restriccions que s'han explicat anteriorment en aquest document. Tot i així hi ha certes limitacions:

- Desenvolupament de l'aplicació mòbil per dispositius Android: degut al temps i recursos limitats per dur a terme el projecte, no es podrà desenvolupar l'aplicació per totes les aplicacions. S'ha triat Android ja que és la que dona menys restriccions i limitacions a l'hora de desenvolupar una aplicació. També perquè en Android es conta amb un mínim d'experiència prèvia.
- Funcionalitats bàsiques: També degut al temps limitat la prioritat principal és tenir un algorisme funcionant amb les restriccions obligatòries i amb les restriccions secundàries més utilitzades actualment i tenir una interfície gràfica amb tots els elements necessaris per poder tenir tota la informació necessària per poder crear llistes que siguin correctes. En el cas que es disposi del temps suficient s'afegiran les funcionalitats extres.

## 2.2.3 Obstacles

Ara analitzarem els obstacles que poden anar apareixent durant el projecte i les possibles solucions per tal que no afecti negativament al resultat final del projecte.

### *Data entrega*

Com que el temps per treballar en el projecte és limitat, cal planificar objectius setmanals a complir i mantenir molt contacte amb el director del projecte per tal que validi que els passos que s'estan seguint són els correctes.

### *Poca experiència a l'hora de desenvolupar aplicacions per dispositius mòbils.*

Degut a que el nivell a l'hora de crear aplicacions per dispositius mòbils és un nivell bàsic s'hauran de fer cursos en línia i buscar informació per tal d'aprendre a com dissenyar i desenvolupar una aplicació des de zero.

*Poca experiència en la creació d'interfícies d'usuari.*

Com en el punt anterior, falta experiència en el disseny i implementació d'interfícies d'usuari. Per resoldre aquest obstacle haurem de buscar informació sobre els punts ha tenir en compte a l'hora de dissenyar una interfície d'usuari i agafar els punts més interessants de les aplicacions del mercat que resolen problemes semblants.

*Aparició de canvis i/o noves funcionalitats.*

En el cas que apareguin canvis i/o noves funcionalitats s'haurà d'analitzar l'estat en el que es troba el projecte per tal de veure si és possible introduir aquests canvis o s'han de descartar per falta de temps.



## 2.3 Metodologia i rigor

Com disposem d'un temps limitat hem de treballar amb una de les metodologies de treball de caràcter àgil com per exemple, SCRUM.



Figura 2: Estructura de la metodologia SCRUM

### 2.3.1 Mètodes de treball: Cicles desenvolupament + Feedback del director

Com s'ha comentat en l'apartat anterior disposem d'un temps limitat per treballar en el projecte. Cal fer cicles de desenvolupament curts amb objectius setmanals i fer un seguiment en les reunions amb el director del projecte per arreglar els errors el més ràpid possible i assegurar-nos que el projecte està en un estat correcte per poder arribar a la data d'entrega.

### 2.3.2 Eines de seguiment

El seguiment de l'elaboració del projecte es farà a partir del control de versions i de les actes de les reunions setmanals amb el director del projecte. Pel control de versions s'utilitzarà el Git del qual disposa l'Android Studio.

### 2.3.3 Mètodes de validació

Pel que fa al disseny de l'algorisme, cada versió nova ha d'anar acompanyada de jocs de proves per tal de comprovar que el funcionament és totalment correcte.

Pel que fa a la interfície d'usuari, i en una fase avançada del projecte es mantindrà un contacte diari amb els voluntaris de la protectora que estiguin provant l'aplicació per tal que puguin reportar els possibles errors el més ràpid possible.

## 2.4 Planificació temporal

### 2.4.1 Definició de les tasques

El projecte té una durada estimada d'uns 4 mesos aproximadament (del 14 de setembre del 2018 al 22 de gener de 2019). Al projecte se li han de dedicar unes 475 hores (130 dies) de feina. L'objectiu és tenir el TFG acabat pel dia 22/01/2019 abans de la lectura del TFG. La idea es treballar una mitja de 5 hores al dia. La durada real del projecte es veurà modificada per les desviacions que poden anar apareixent durant la realització d'aquest. El seguiment setmanal amb el director del projecte té com un dels principals objectius, detectar aquestes desviacions per tal de prendre les decisions necessàries per tal de complir amb la data final estimada, sempre que sigui possible.

Més endavant es detalla com es resoldran les eventuais desviacions que es trobin durant el desenvolupament del projecte

#### 2.4.1.1 Anàlisi del projecte

En la fase d'anàlisi del projecte caldrà definir els requisits no funcionals i les restriccions en les que es veurà implicat el projecte. A més a més, es definiran les diferents funcionalitats (casos d'ús), l'algoritme de satisfacció de restriccions (CSP<sup>1</sup>) a implementar i el model del comportament del sistema

#### 2.4.1.2 Disseny del projecte

Pel que fa al disseny caldrà definir la interfície de l'usuari de l'aplicació aplicant els coneixements apresos aquests anys i l'arquitectura del software, és a dir, el conjunt de components dels que constarà el sistema, sent un dels components principals l'algoritme de creació de llistes.

---

<sup>1</sup> CSP: Problemes de satisfacció de restriccions

#### 2.4.1.3 Desenvolupament del projecte

La implementació es dividirà en 2 parts, per una banda, s'implementarà l'algoritme CSP i una vegada ja es tingui una versió funcional que contempli la majoria de les restriccions necessàries, es començarà el desenvolupament de la interfície d'usuari de l'aplicació mòbil. Això implicarà que s'aniran fent modificacions a l'algoritme durant el desenvolupament de la interfície. D'aquesta manera ens assegurem d'arribar a la data final del desenvolupament amb un algoritme funcional que contempla totes les restriccions que s'havien definit en un principi i una interfície d'usuari que compleix amb tots els casos d'ús i funcionalitats definides. En el seguiment setmanal amb el director del projecte s'analitzaran quins punts prioritzar i els canvis o funcionalitats opcionals que es poden desenvolupar o, en cas que no es puguin complir les dates, els punts que s'han de deixar pendents.

#### 2.4.1.4 Fita final

A la fita final es presentarà la memòria que inclourà el prototip de l'aplicació que funcionarà amb l'algoritme de satisfacció de restriccions implementat durant el procés de desenvolupament. A més a més, a la memòria s'explicarà el funcionament principal del algoritme acompanyat del seu pseudocodi i com a annex, una guia d'ús de l'aplicació.

### 2.4.1.5 Descripció de les tasques

Al treballar amb una metodologia àgil com el SCRUM no hi ha cap manera de definir unes tasques per tal que aquestes es vagin aplicant de forma iterativa. El que si que podem definir són unes tasques genèriques que sabem del cert que mantenen entre elles una relació de dependència. Les tasques més concretes i objectius a complir es definiran de forma setmanal i s'hauran d'adaptar en el cas que aparegui algun problema o hi hagi qualsevol tipus de desviació.

Les tasques genèriques que es defineixen a continuació tenen relacions de dependència que fa que no es puguin realitzar de forma paral·lela:

1. Realització de la fita inicial
2. Anàlisi
  - 2.1. Anàlisi dels diferents algorismes per resoldre problemes de satisfacció de restriccions per trobar el que millor s'adapta al nostre problema i satisfà els requisits de complexitat i eficiència.
  - 2.2. Anàlisi de diferents interfícies d'usuari de softwares que resolen problemes del mateix tipus per veure els punts forts i febles.
3. Disseny
  - 3.1. Disseny del algoritme. Cal decidir quin de tots els algorismes SCP que existeixen s'implementarà i es definiran l'estat inicial del problema i les restriccions que l'algoritme haurà de tenir en compte.
  - 3.2. Disseny de la interfície d'usuari. A partir de l'anàlisi anterior es farà el disseny de la interfície tenint en compte tots els casos d'ús que s'hagin definit.
4. Implementació
  - 4.1. Implementació del algorisme que s'aplicarà per tal de crear les llistes
    - 4.1.1. Implementació de l'estructura bàsica de l'algoritme que contempla les restriccions més bàsiques del problema.
    - 4.1.2. Adaptacions de l'algoritme i desenvolupament de la resta de restriccions definides per aquest problema.
  - 4.2. Desenvolupament de la interfície d'usuari
    - 4.2.1. Desenvolupament de la part de la interfície que s'encarrega de d'executar l'algoritme i de mostrar/modificar la solució final.

- 4.2.2. Desenvolupament de la part de la interfície que s'encarrega de l'entrada i consulta de les dades
- 4.3. Parametrització del algoritme. Un cop els 2 desenvolupament anteriors estiguin en un estat avançat s'haurà d'implementar la forma en que la interfície crida a l'algoritme i com aquest retorna la solució.
- 4.4. Implementació de funcionalitats addicionals. En el cas que es disposi del temps suficient s'implementaran funcionalitats extra en la interfície d'usuari. Principalment es tracta de funcionalitats que augmenten la informació que l'aplicació mostrarà a l'usuari. No són funcionalitats directament relacionades amb l'algoritme.
5. Fase de proves
  - 5.1. Fase de proves inicial. Principalment es faran proves de que el funcionament de l'algoritme és el correcte i que les restriccions s'apliquen tal com s'havien definit.
  - 5.2. Fase de proves amb els futurs usuaris quan l'aplicació estigui en un estat bastant avançat (Beta). Proves generals on es faran proves de la interfície de l'aplicació i s'acabaran d'arreglar els possibles errors de l'algoritme que puguin aparèixer.
6. Redacció de la memòria, l'annex i la documentació que s'ha fet durant el desenvolupament del projecte.

### 2.4.1.6 Duració de les tasques

Tasca	Duració(h)	Inici	Fi	Dependències
<b>1.Gestió de projectes</b>	<b>60</b>			
1.1 Abast del projecte i contextualització	24	14/09/2018	25/09/2018	-
1.2 Planificació temporal	8	26/09/2018	01/10/2018	1.1
1.3 Gestió econòmica i sostenibilitat	9	02/10/2018	8/10/2018	1.2
1.4 Plec de condicions	16	09/10/2018	15/10/2018	1.3
<b>2.Anàlisi del projecte</b>	<b>35</b>			
2.1 Anàlisi algorismes SCP	20	16/10/2018	19/10/2018	1.3
2.2 Anàlisi UI <sup>2</sup>	15	20/10/2018	23/10/2018	1.3
<b>3.Disseny del projecte</b>	<b>25</b>			
3.1 Disseny algoritme	7	20/10/2018	21/10/2018	2.1
3.2 Disseny UI	18	24/10/2018	27/11/2018	2.2
<b>4.Implementació del projecte</b>	<b>215</b>			
<i>4.1 Implementació algoritme</i>	80			
4.1.1 Estructura bàsica i restriccions obligatòries	50	28/10/2018	8/11/2018	3.1
4.1.2 Adaptacions del algoritme i resta de restriccions	30	18/11/2018	25/11/2018	4.1.1
<i>4.2 Desenvolupament UI</i>	90			
4.2.1 Desenvolupament de la UI relacionada amb l'execució del algoritme	35	9/11/2018	17/11/2018	3.2
4.2.2 Desenvolupament de la part de la UI d'entrada/consulta de dades	55	1/12/2018	13/12/2018	4.2.1
4.3 Parametrització	25	14/12/2018	19/12/1028	4.1.1-4.2.1
4.4 Funcions addicionals	20	20/12/2018	25/12/2018	4.1.2-4.2.2-4.3
<b>5.Fase de proves</b>	<b>50</b>			
5.1 Fase de proves inicial	20	26/11/2018	30/11/2018	4.1.2

<sup>2</sup> User Interface

5.2 Fase de proves(Beta)	30	26/12/2018	03/01/2019	5.1-4.2.2-4.3
<b>6.Redacció de la memòria, l'annex</b>	<b>30</b>	04/01/2019	14/01/2019	-
<b>7.Reunions de control</b>	<b>30</b>			
7.1 Seguiment tasques implementades	10	Cada setmana	Cada setmana	-
7.2 Anàlisi de l'estat del projecte i compliment de les dates	10	Cada setmana	Cada setmana	7.1
7.3 Marcar objectius per la setmana següent	10	Cada setmana	Cada setmana	7.2
<b>8.Documentació</b>	<b>30</b>			
8.1 Documentació fase de anàlisi	6	16/10/2018	23/10/2018	
8.2 Documentació fase de disseny	7	20/10/2018	27/10/2018	
8.3 Documentació fase de desenvolupament	8	28/10/2018	25/12/2018	
8.4 Documentació fase de proves	9	26/11/2018	3/01/2019	
<b>Total</b>	<b>475</b>			

Taula 1.Duració aproximada de les tasques

#### 2.4.1.7 Dependències

Com podem veure a la *Taula 1* hi ha tasques que depenen d'altres per tal de poder començar-les. En aquest apartat veurem en més detall els motius d'aquestes dependències. Agafant els valors de la taula anterior:

##### Anàlisi

Una vegada feta la documentació inicial ja es pot començar amb l'anàlisi:

1.3.3 < 2.1

1.3.3 < 2.2

## Disseny

Un cop fet l'anàlisi del algoritme ja és pot fer el seu disseny: 2.1 < 3.1

Un cop fet l'anàlisi de la interfície ja és pot fer el disseny: 2.2 < 3.2

## Desenvolupament

Quan ja s'ha fet el disseny del algoritme ja és pot implementar l'estructura d'aquest i el desenvolupament de les restriccions bàsiques. Una vegada feta aquesta versió inicial ja es podrà treballar en fer la resta de restriccions i aplicar els canvis necessaris al algoritme:

3.1 < 4.1.1 < 4.1.2

El mateix farem per la interfície. Començarem desenvolupant la part de la interfície que s'encarregarà de configurar els paràmetres i la crida del algoritme. Quan aquesta part de la interfície estigui pràcticament enllestida ja es podrà començar amb la part de la interfície més dedicada a l'entrada i consulta de dades .

2.2 < 4.2.1 < 4.2.2

En el punt que ja s'hagin implementat la versió inicial de l'algoritme i la part de la interfície que fa la crida del algoritme ja es podrà treballar en connectar aquestes dues parts implementant quins paràmetres la interfície haurà de passar al algoritme i quina informació haurà de retornar aquest a la interfície per tal de mostrar el resultat de l'execució al usuari.

4.1.1 + 4.2.1 < 4.3

En el cas que es puguin implementar canvis o noves funcionalitats es faran una vegada els punts anteriors s'hagin acabat de desenvolupar.

4.1.2 + 4.2.2 + 4.3 < 4.4

## Proves

La primera fase de proves només provarà el funcionament del algoritme per tant només cal tenir aquest acabat d'implementar.

4.1.2 < 5.1

La segona fase de proves ja necessita que tant la interfície d'usuari com l'algoritme estiguin pràcticament acabat ja que en aquest punt la resta de voluntaris/futurs usuaris



ja han de ser capaços de poder provar totes les funcionalitats definides en el disseny de l'aplicació.

$$5.1+4.2.2+4.3 < 5.2$$

### 2.4.1.8 Diagrama de Gantt

App	123d	9/14/18 8:00 AM	1/14/19 5:00 PM	0%	475h	\$0	1
1. Gestió de projectes	32d	9/14/18 8:00 AM	10/15/18 5:00 PM	0%	60h	\$0	2
1.1 Abast del projecte i contextualització	12d	9/14/18 8:00 AM	9/25/18 5:00 PM	0%	24h	\$0	3
1.2 Planificació temporal	6d	9/26/18 8:00 AM	10/1/18 5:00 PM	0%	10h	\$0	4
1.3 Gestió econòmica i sostenibilitat	7d	10/2/18 8:00 AM	10/8/18 5:00 PM	0%	10h	\$0	5
1.4 Plec de condicions (Bloc especialitats)	7d	10/9/18 8:00 AM	10/15/18 5:00 PM	0%	16h	\$0	6
2. Anàlisis del projecte	8d	10/16/18 8:00 AM	10/23/18 5:00 PM	0%	35h	\$0	7
2.1 Anàlisi algorismes SCP	4d	10/16/18 8:00 AM	10/19/18 5:00 PM	0%	20h	\$0	8
2.2 Anàlisi UI	4d	10/20/18 8:00 AM	10/23/18 5:00 PM	0%	15h	\$0	9
3. Disseny del projecte	8d	10/20/18 8:00 AM	10/27/18 5:00 PM	0%	25h	\$0	10
3.1 Disseny algorisme	2d	10/20/18 8:00 AM	10/21/18 5:00 PM	0%	7h	\$0	11
3.2 Disseny UI	4d	10/24/18 8:00 AM	10/27/18 5:00 PM	0%	18h	\$0	12
4. Implementació del projecte	59d	10/28/18 8:00 AM	12/25/18 5:00 PM	0%	215h	\$0	13
4.1 Implementació algorisme	29d	10/28/18 8:00 AM	11/25/18 5:00 PM	0%	80h	\$0	14
4.1.1 Estructura bàsica	12d	10/28/18 8:00 AM	11/8/18 5:00 PM	0%	50h	\$0	15
4.1.2 Adaptacions i resta de restriccions	8d	11/18/18 8:00 AM	11/25/18 5:00 PM	0%	30h	\$0	16
4.2 Desenvolupament UI	35d	11/9/18 8:00 AM	12/13/18 5:00 PM	0%	90h	\$0	17
4.2.1 Desenvolupament UI part relacionada amb al...	9d	11/9/18 8:00 AM	11/17/18 5:00 PM	0%	35h	\$0	18
4.2.2 Desenvolupament UI part consulta/entrada da...	13d	12/1/18 8:00 AM	12/13/18 5:00 PM	0%	55h	\$0	19
4.3 Parametrització	6d	12/14/18 8:00 AM	12/19/18 5:00 PM	0%	25h	\$0	20
4.4 Funcions addicionals	6d	12/20/18 8:00 AM	12/25/18 5:00 PM	0%	20h	\$0	21
5. Fase de proves	39d	11/26/18 8:00 AM	1/3/19 5:00 PM	0%	50h	\$0	22
5.1 Fase de proves inicial	5d	11/26/18 8:00 AM	11/30/18 5:00 PM	0%	20h	\$0	23
5.2 Fase de proves(Beta)	9d	12/26/18 8:00 AM	1/3/19 5:00 PM	0%	30h	\$0	24
6. Redacció de la memòria, l'annex	11d	1/4/19 8:00 AM	1/14/19 5:00 PM	0%	60h	\$0	25
7. Reunions de control		Setmanalment	Setmanalment	0%	30h	\$0	26
7.1 Seguiment tasques implementades				0%	10h	\$0	27
7.2 Anàlisi estat projecte i compliment de les dates				0%	10h	\$0	28
7.3 Marcar següent objectius				0%	10h	\$0	29
8. Documentació	80d	10/16/18 8:00 AM	1/3/19 5:00 PM	0%	0h	\$0	30
8.1 Documentació anàlisi	8d	10/16/18 8:00 AM	10/23/18 5:00 PM	0%	0h	\$0	31
8.2 Documentació disseny	8d	10/20/18 8:00 AM	10/27/18 5:00 PM	0%	0h	\$0	32
8.3 Documentació desenvolupament	59d	10/28/18 8:00 AM	12/25/18 5:00 PM	0%	0h	\$0	33
8.4 Documentació proves	39d	11/26/18 8:00 AM	1/3/19 5:00 PM	0%	0h	\$0	34

Figura 3. Taula amb les tasques del diagrama de Gantt

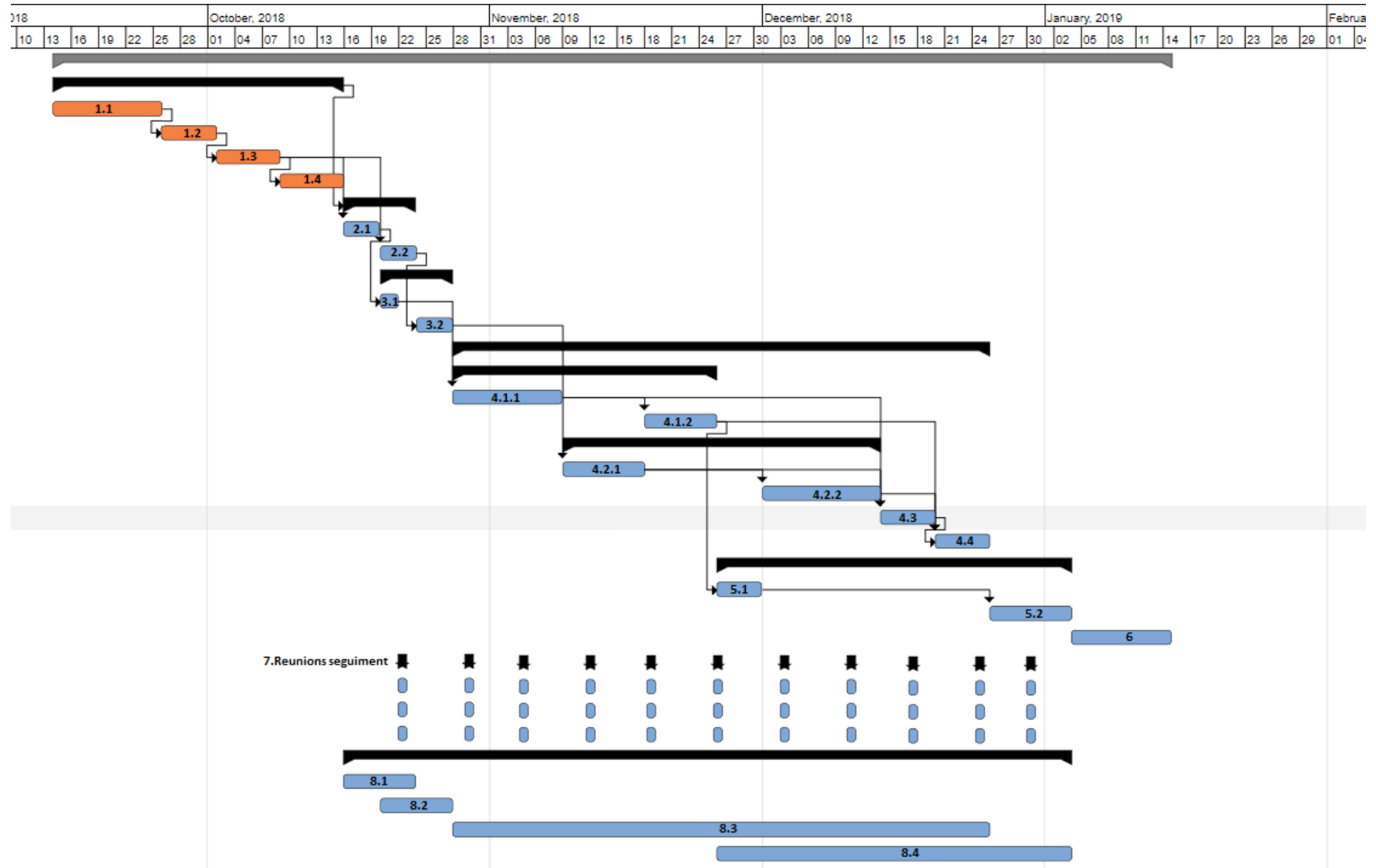


Figura 4. Diagrama de Gannt

### 2.4.1.9 Recursos

Per al desenvolupament d'aquest projecte són necessàries les següents eines de hardware i software.

#### *Hardware*

El hardware que s'utilitzarà és:

1. Ordinador de sobretaula
2. Le Leeco X626 4GB 64GB 21MP Helio X20 Deca-core 5.5" FHD Android 6.0 telèfono 4GP04-X626G



#### *Software*

Les eines de software que s'utilitzaran són:

1. Android Studio
2. JustInMind Prototyper
3. Office 365 Personal
4. Windows 10 Home
5. Google Chrome
6. Google Drive



## 2.4.2 Valoració d'alternatives i pla d'acció

Durant el desenvolupament del projecte poden aparèixer diferents problemes i incidències que poden afectar a les dates indicades en la planificació i provocar certs desviaments.

En la fase d'implementació és on ens podem trobar amb els problemes més rellevants. Per això a les diferents tasques d'aquesta fase ja s'hi han afegit hores de més, així de bon principi ja ens estem assegurant de que les incidències que apareguin no implicaran un incompliment de les dates planificades. Igualment hem d'estar preparats per si arriba un moment on no es poden complir les dates.

Com ja s'ha explicat anteriorment, els obstacles que poden sorgir en el desenvolupament del projecte:

1. Falta de temps per acabar el projecte. Com que no es disposen de moltes hores diàries per dedicar al TFG és un dels punts de més preocupació. Per tant, s'haurà de controlar la planificació i complir amb les dates. A part, es faran reunions setmanals amb el director del projecte per fer un seguiment per tal de saber en tot moment que el projecte es podrà acabar en l'estat esperat el dia de la data d'entrega. En cas que sigui difícil acabar el projecte pel dia de la data final s'hauran d'incrementar les hores diàries que s'han de dedicar al projecte.
2. Poca experiència a l'hora de desenvolupar aplicacions per dispositius mòbils. Es faran cursos online i desenvolupament d'aplicacions senzilles per tal de provar els continguts apresos abans de començar amb el desenvolupament de l'aplicació d'aquest projecte. Com es disposa de poc temps fins que comença el desenvolupament del projecte, haurem de dedicar unes 50 hores a aprendre a treballar amb el Android Studio.
3. Aparició de canvis i/o noves funcionalitats. En cas que apareguin canvis o altres funcionalitats es faran anàlisis per saber si es disposa del temps suficient per implementar-ho. Les hores que es dedicaran a implementar els canvis o les noves funcionalitats no ho podem saber ja que dependrà de com es vagi desenvolupant el projecte.

4. Poca experiència en la creació d'interfícies d'usuari. Com en el punt 2) es faran cursos online i es consultarà documentació per tal d'acabar desenvolupant una bona interfície d'usuari. Com disposem de poc temps fins que comença el desenvolupament de la interfície, haurem de dedicar unes 30 hores a aprendre els punts a tenir en compte per desenvolupar una bona interfície.

## 2.5. Gestió Econòmica

### 2.5.1 Identificació i estimació dels costos

En aquest punt farem una estimació dels elements que formen part del pressupost del projecte. Es tindran en compte tots els elements detallats en els punts anteriors i els recursos que s'han utilitzat, per exemple, les hores dedicades al projecte, el *hardware*, el *software* i despeses indirectes.

Els elements sobre els que calcularem el pressupost sorgeixen a partir del que s'ha definit al diagrama de Gantt vist anteriorment.

#### 2.5.1.1 Pressupost de recursos humans

Aquest projecte, com en la gran majoria, hi participen diferents persones que ocupen diferents rols dins d'aquest. En aquest cas com el desenvolupament sencer del projecte el fa una sola persona, aquesta serà l'encarregada de fer tots els rols. A la *Taula2* detallem els diferents rols i el preu per hora de cada rol. Aquests preus s'han consultat a *Indeed* [10] on es mostren els salaris mitjans dels diferents rols a Espanya.

Rol	Preu per hora
Cap de projecte	25€/h
Analista	19€/h
Dissenyador	17€/h
Programador	14€/h
BetaTester	18€/h

Taula2: Cost de cada rol.

Per calcular les despeses directes del projecte haurem de fer un estimació del cost per cada tasca del diagrama de Gannt:

Tasca	Rol/s implicat/s	Hores	Cost
<b>1.Gestió de projectes</b>	-	<b>60</b>	-
1.1 Abast del projecte i contextualització	Cap de projecte	24	600€
1.2 Planificació temporal	Cap de projecte	9	220€
1.3 Gestió econòmica i sostenibilitat	Cap de projecte	10	250€
1.4 Plec de condicions	Cap de projecte	17	425€
<b>2.Anàlisis del projecte</b>	-	<b>35</b>	-
2.1 Anàlisi algorismes SCP	Analista (80%)	20	304€
	Cap de projecte (20%)		100€
2.2 Anàlisi UI <sup>3</sup>	Analista (80%)	15	228€
	Cap de projecte (20%)		75€
<b>3.Disseny del projecte</b>	-	<b>25</b>	-
3.1 Disseny algoritme	Dissenyador(80%)	7	95,2€
	Cap de projecte (20%)		35€
3.2 Disseny UI	Dissenyador(80%)	18	244,8€
	Cap de projecte(20%)		90€
<b>4.Implementació del projecte</b>	-	<b>215</b>	-
<i>4.1 Implementació algoritme</i>	-	80	-
4.1.1 Estructura bàsica i restriccions obligatòries	Programador(90%)	50	630€
	Cap de projecte(10%)		125€
4.1.2 Adaptacions del algoritme i resta de restriccions	Programador(90%)	30	378€
	Cap de projecte(10%)		75€
<i>4.2 Desenvolupament UI</i>	-	90	-

<sup>3</sup> User Interface

4.2.1 Desenvolupament de la UI relacionada amb l'execució del algoritme	Programador(90%)	35	441€
	Cap de projecte(10%)		87,5€
4.2.2 Desenvolupament de la part de la UI d'entrada/consulta de dades	Programador(90%)	55	693€
	Cap de projecte(10%)		137,5€
4.3 Parametrització	Programador(90%)	25	315€
	Cap de projecte(10%)		62,5€
4.4 Funcions addicionals	Programador(90%)	20	252€
	Cap de projecte(10%)		50€
<b>5.Fase de proves</b>	-	<b>50</b>	-
5.1 Fase de proves inicial	Beta tester(90%)	20	324€
	Cap de projecte(10%)		50€
5.2 Fase de proves(Beta)	Beta tester(90%)	30	486€
	Cap de projecte(10%)		75€
<b>6.Redacció de la memòria, l'annex</b>	Cap de projecte	<b>30</b>	750€
<b>7.Reunions de control</b>	-	<b>30</b>	-
7.1 Seguiment tasques implementades	Cap de projecte	10	250€
7.2 Anàlisi de l'estat del projecte i compliment de les dates	Cap de projecte	10	250€
7.3 Marcar objectius per la setmana següent	Cap de projecte	10	250€
<b>8.Documentació</b>	-	<b>30</b>	-
8.1 Documentació fase de anàlisi	Cap de projecte	6	150€
8.2 Documentació fase de disseny	Cap de projecte	7	175€



8.3 Documentació fase de desenvolupament	Cap de projecte	8	200€
8.4 Documentació fase de proves	Cap de projecte	9	225€
<b>Total</b>		<b>475</b>	<b>9.103,5€</b>

Taula 3: Pressupost recursos humans

### 2.5.1.2 Pressupost del *hardware*

Per al desenvolupament del projecte es farà servir una ordinador i per les proves un dispositiu mòbil:

Producte	Cost	Vida útil	Amortització total estimada(4 mesos)
Ordinador de sobretaula	600€	4 anys	50€
Smartphone	125€	3 anys	13,88€
<b>TOTAL</b>			<b>53,88€</b>

Taula 4: Pressupost del *hardware*

### 2.5.1.3 Pressupost del software

Pel desenvolupament d'aquest projecte s'ha fet servir el següent *software*:

Producte	Cost	Vida útil	Amortització total estimada(4 mesos)
Android Studio	0,00€	-	0,00€
JustInMind Prototyper	0,00€(Versió prova)	-	0,00€
Office 365 Personal	69€	1 any	23€
Windows 10 Home	145€	2 anys	24,16€
Google Chrome	0,00€	-	0,00€
Google Drive	0,00€	-	0,00€
<b>TOTAL</b>			<b>47,16€</b>

Taula 5: Pressupost del *software*

### 2.5.1.4 Despeses indirectes

En aquest apartat es tenen en compte les despeses associades al desenvolupament del projecte ja siguin l'accés a internet, l'electricitat consumida, la contractació d'un *hosting* o les llicències per poder publicar l'aplicació de *Android* a *Google Play Store*:

Producte/Servei	Cost	Període	Total estimat
Energia elèctrica	0,143 €/KWh	475h	67,93€
Accés a internet	47,73€	5 mesos	238,65€
Transport	20€	5 mesos	100€
Llicència Android	21,67€	Indefinit	21,67€
Servidor i domini (Digital Ocean)	5,44€	5 mesos	27,20€
<b>TOTAL</b>			<b>455,45€</b>

Taula 6: Despeses indirectes

### 2.5.1.5 Despeses imprevistes

Els únics imprevistos amb els que ens podríem trobar que ens suposessin unes despeses addicionals a part de les ja calculades seria que durant el desenvolupament del projecte fallés o l'ordinador o el mòbil. En aquest cas s'hauria de sumar el cost de la reparació (uns 300€ de mitja). Si un dels 2 aparells quedés temporalment inactiu es passaria a treballar amb un altre ordinador o un altre dispositiu mòbil per tal de poder seguir amb el desenvolupament normal del projecte. La probabilitat de que falli un dels 2 aparells és del 10%. Per tant les despeses imprevistes seran 30€.

### 2.5.1.6 Cost total del projecte

Concepte	Cost
Despeses directes	9.103,5€
Despeses indirectes	556,49€
Contingència(5%)	483€
Despeses imprevistes	30,00€
<b>TOTAL</b>	<b>10.172,99€</b>

Taula 7: Resum del cost del projecte

### 2.5.2 Control de gestió

Per tal de portar un control de les hores que treballa un rol concret, es crearà una taula que s'actualitzarà diàriament. En aquesta taula s'indicarà la tasca realitzada, el rol, i les hores que s'hi han dedicat.

Dia	Cap de projecte	Analista	Dissenyador	Programador	Tester
05/10/2018	2h	2h	-	-	-
06/10/2018	2h	-	2h	-	-

Taula 8: Control de hores/rol

Durant el transcurs del projecte i al finalitzar-lo es sumaran les hores que s'han treballat amb cada rol i d'aquesta forma es podran detectar desviacions respecte a l'estimació inicial. Si hi ha desviacions, caldrà prendre les decisions oportunes a tenir en compte durant la resta de projecte. Aquesta eina ens servirà per tenir un control dinàmic del projecte.

Al finalitzar el projecte obtindrem les hores reals de cada rol per tal de calcular les despeses directes reals i calcular les diferents desviacions a partir de les dades reals i les dades de la planificació.

Rol	Hores estimades	Hores reals	Cost estimat	Cost real
Analista				
Tester				

Taula 9: Control de desviacions

### 2.5.3 Informe de sostenibilitat

A l'hora de parlar sobre la sostenibilitat del projecte, s'han d'avaluar i analitzar l'impacte del projecte en 3 àmbits diferents, que són l'econòmic, el social i ambiental.

#### 2.5.3.1 Dimensió econòmica

En aquest projecte s'ha fet un anàlisi de les despeses generals com són les materials i els recursos. N'hi ha d'altres que no s'han tingut en compte ja que apareixerien en una etapa posterior de la vida útil del producte que s'ha desenvolupat com podrien ser les despeses per fer el manteniment o per fer actualitzacions.

Si comparem la nostra planificació amb altres projectes semblants podem veure que les despeses estan tan ajustades que el projecte és viable econòmicament parlant. A partir d'aquests altres projectes hem agafat la informació necessària per tal de planificar la durada de les tasques en períodes de temps semblant. Totes les tasques que s'han de fer durant el desenvolupament del projecte es podrien fer de forma més ràpida si disposéssim de més persones.

#### 2.5.3.2 Dimensió social

En referència a la situació social, un dels principals problemes de la protectora, a part de la gran quantitat d'animals abandonats a aquest país durant l'any en comparació amb altres països d'Europa, és la falta de personal. Com la majoria de recursos dels que disposa una protectora va destinada a la millora de la qualitat de vida dels gossos, d'alimentar-los, de vacunar-los, de medicar-los moltes vegades no és pot contractar la quantitat de gent que seria necessària per ocupar-se de tots els animals. Per aquest motiu existeix el voluntariat, les persones que ens comprometen i som voluntaris d'una protectora tenim com a objectiu ajudar amb tot el que sigui necessari però el temps que dediquem a la protectora segueix sent insuficient per ser capaços de fer tota la feina que hi ha en una protectora.

La realització d'aquest projecte cobreix una necessitat real. El fet de poder-nos organitzar millor i de forma més ràpida significa que es disposa de més temps per tal de dedicar-lo a altres tasques importants i/o urgents de la protectora. D'altra banda els actuals encarregats de fer les llistes no hauran de dedicar-hi tant de temps i si hi ha canvis d'última hora podran modificar-ho de forma ràpida i àgil.

Una vegada hi hagi l'aplicació en funcionament la qualitat del voluntariat millorarà substancialment. En primer lloc els responsables de crear les llistes agrairan no haver de perdre hores entre setmana per fer les llistes. Els dies de voluntariat si falla algú a última hora, es podrà obtenir una llista modificada al moment. En segon lloc, una de les característiques del algoritme serà maximitzar el nombre de passejos on el voluntari treu un dels seus gossos preferits. Això provocarà que els voluntaris marxin de la protectora més satisfets. Per últim, amb una bona organització es disposarà de més temps per treballar amb gossos porucs. Aquest és un dels punts que més ens preocupa actualment als voluntaris, ja que gossos que no es deixen tocar o no poden sortir a passejar és molt difícil que surtin adoptats, per tant, ens hem de marcar com a objectiu que els dies de voluntariat tots els gossos porucs o insegurs vagin millorant poc a poc i perdent les pors. L'objectiu d'aquest projecte és que l'aplicació resultant, a part de millorar l'organització, faci que els voluntaris estiguin més satisfets i per tant, vinguin a la protectora de forma més freqüent i així tenir cada dia les mans suficients per tal de poder fer tots els passejos i a més poder treballar amb els gossos que no poden sortir a passejar.

### 2.5.3.3 Dimensió ambiental

Durant l'elaboració del projecte s'utilitzen diferents recursos que afecten al medi ambient. En totes les fases del projecte s'utilitza principalment electricitat, ja sigui el ordinador amb el que es treballarà per desenvolupar el projecte, o el router per poder-me connectar a internet, o el servidor que es necessitarà contractar per tenir-hi una base de dades.

Actualment, abans de desenvolupar el projecte les llistes de la protectora es fan amb paper i bolígraf. Aquest projecte informatitzarà aquest procés fent que es gastin menys paper, plàstics i tinta però en canvi, es consumirà més energia ja que la consulta de dades i la creació de llistes es farà des d'un dispositiu mòbil.

### 2.5.4. Autoavaluació del domini actual de la competència de sostenibilitat

Dels diferents apartats que es tracten en l'enquesta n'hi ha alguns que he pogut valorar amb una alta puntuació ja que pregunta sobre temes que s'han tractat en aquest document i n'hi ha d'altres dels que només he mirat una mica d'informació.

Pel que fa al anàlisi de la sostenibilitat des de els àmbits ambiental, social i econòmic i al càlcul de costos directes i indirectes he après durant la redacció d'aquest document els passos a seguir per tal d'arribar a calcular un pressupost teòric del projecte i a partir d'aquest pressupost fer un anàlisi en els diferents àmbits.

A part, també relacionat amb l'àmbit social que hem comentat en aquest document, l'objectiu d'aquest projecte i de tots els projectes que he desenvolupat o desenvoluparé en un futur tenen com a finalitat tenir un impacte positiu i millorar el bé comú de la societat.

D'altra banda, hi ha apartats dels que es parla en l'enquesta que en el moment de fer-la no sabia de que s'estava parlant com per exemple, la justícia social, l'economia circular, la problemàtica d'accessibilitat, l'ergonomia, la equitat, la diversitat o la transparència. Una vegada feta l'enquesta he buscat informació sobre cada un d'aquests punts però no he profunditat en com s'ha de tenir en compte aquests aspectes a l'hora de fer la documentació del projecte.

El que també desconeixia és l'existència dels principis deontològics relacionats amb la sostenibilitat. De fet desconeixia que existien principis deontològics per a cada professió per tal de saber si es du a terme de forma correcta certa activitat professional. Una vega feta l'enquesta he buscat els principis deontològics dels enginyers informàtics i els punts que fan referència a la sostenibilitat.

### 3.Anàlisi de requisits

#### 3.1 Requisits funcionals

##### 3.1.1 Gestió gossos

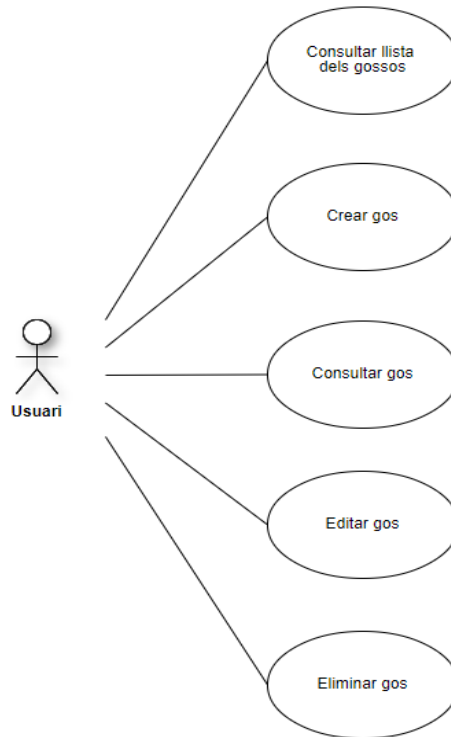


Figura 5. Diagrama de cas d'ús de la gestió de gossos

Nom	Consultar llista dels gossos
Actor primari	Usuari
Precondició	
Disparador	L'usuari sol·licita consultar la llista dels gossos
Escenari principal	1- L'usuari selecciona a la pantalla inicial l'opció de veure els gossos. 2- El sistema mostra una pantalla nova on hi ha la llista amb tots els gossos.
Extensions	



Nom	Crear gos
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla on hi ha la llista de tots els gossos.
Disparador	L'usuari clica al element per tal de crear un gos nou.
Escenari principal	1- L'usuari selecciona a la pantalla l'element inicial de la llista que és el que permet crear un gos nou.  2- El sistema mostra una pantalla amb el formulari on s'han d'introduir les dades del gos nou
Extensions	Permetre la modificació de la imatge del gos per una que hi hagi a la galeria de fotos del mòbil.  No permetre crear el gos si no s'han introduït les dades necessàries.

Nom	Consultar gos
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla on hi ha la llista de tots els gossos.
Disparador	L'usuari clica a un gos de la llista.
Escenari principal	1- L'usuari selecciona a la pantalla un gos determinat de la llista.  2- El sistema mostra una pantalla amb les dades del gos.
Extensions	

Nom	Editar gos
Actor primari	Usuari
Precondició	L'usuari es troba al perfil d'un dels gossos
Disparador	L'usuari clica al botó d'editar gos.
Escenari principal	1- L'usuari clica al botó d'editar gos.  2- El sistema mostra una pantalla amb el formulari on es poden modificar les dades
Extensions	Permetre la modificació de la imatge del gos per una que hi hagi a la galeria de fotos del mòbil.  No permetre editar el gos si les dades modificades no són correctes.

Nom	Eliminar gos
Actor primari	Usuari
Precondició	L'usuari es troba editant el gos
Disparador	L'usuari clica al botó d'eliminar el gos.
Escenari principal	1- L'usuari clica al botó d'eliminar el gos. 2- El sistema esborra el gos.
Extensions	Mostrar un missatge de confirmació abans d'esborrar el gos

### 3.1.2 Gestió voluntaris

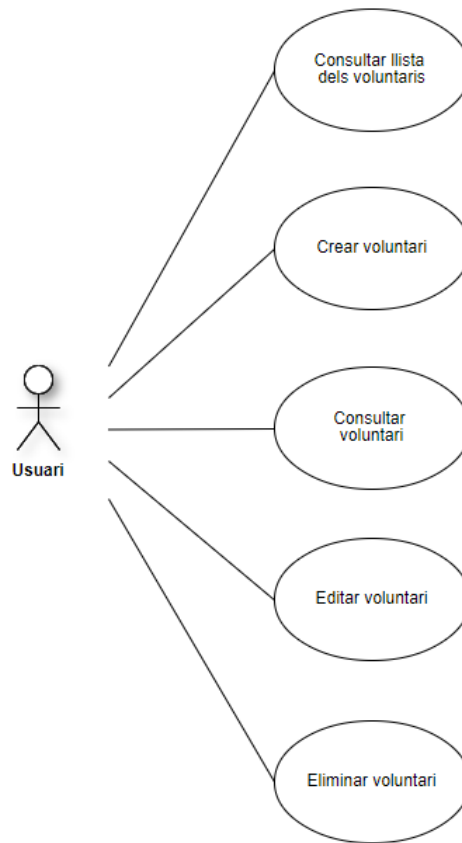


Figura 6. Diagrama de cas d'ús de la gestió de voluntaris

Nom	Consultar llista dels voluntaris
Actor primari	Usuari
Precondició	
Disparador	L'usuari sol·licita consultar la llista dels voluntaris
Escenari principal	1- L'usuari selecciona a la pantalla inicial l'opció de veure els voluntaris. 2- El sistema mostra una pantalla nova on hi ha la llista amb tots els voluntaris.
Extensions	

Nom	Crear voluntari
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla on hi ha la llista de tots els voluntaris.
Disparador	L'usuari clica al element per tal de crear un voluntari nou.
Escenari principal	1- L'usuari selecciona a la pantalla l'element inicial de la llista que és el que permet crear un voluntari nou.  2- El sistema mostra una pantalla amb el formulari on s'han d'introduir les dades del voluntari nou
Extensions	Permetre la modificació de la imatge del voluntari per una que hi hagi a la galeria de fotos del mòbil.  No permetre crear el voluntari si no s'han introduït les dades necessàries.

Nom	Consultar voluntari
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla on hi ha la llista de tots els voluntaris.
Disparador	L'usuari clica a un voluntari de la llista.
Escenari principal	1- L'usuari selecciona a la pantalla un voluntari determinat de la llista.  2- El sistema mostra una pantalla amb les dades del voluntari.
Extensions	

Nom	Editar voluntari
Actor primari	Usuari
Precondició	L'usuari es troba al perfil d'un dels voluntaris
Disparador	L'usuari clica al botó d'editar voluntari.
Escenari principal	1- L'usuari clica al botó d'editar voluntari. 2- El sistema mostra una pantalla amb el formulari on es poden modificar les dades
Extensions	Permetre la modificació de la imatge del voluntari per una que hi hagi a la galeria de fotos del mòbil.  No permetre editar el voluntari si les dades modificades no són correctes.

Nom	Eliminar voluntari
Actor primari	Usuari
Precondició	L'usuari es troba editant el voluntari
Disparador	L'usuari clica al botó d'eliminar el voluntari.
Escenari principal	1- L'usuari clica al botó d'eliminar el voluntari. 2- El sistema esborra el voluntari.
Extensions	Mostrar un missatge de confirmació abans d'esborrar el voluntari

### 3.1.3 Gestió distribució gossos

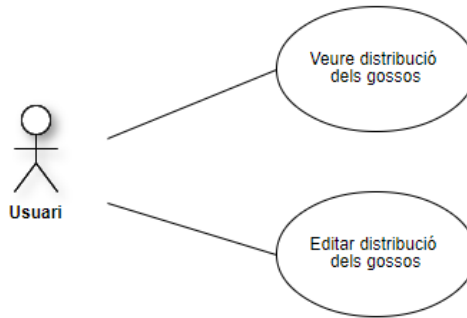


Figura 7. Diagrama de cas d'ús de la gestió de la distribució de gossos

Nom	Veure distribució dels gossos
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla de distribució de gossos
Disparador	L'usuari clica al botó de veure distribució
Escenari principal	1- L'usuari clica al botó de veure la distribució dels gossos 2- El sistema mostra per a cada zona de la protectora, un mapa amb les gàbies, i els gossos que es troben en cada gàbia de la zona que s'estigui consultant.
Extensions	

Nom	Editar distribució dels gossos
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla de distribució de gossos
Disparador	L'usuari clica al botó d'editar distribució
Escenari principal	1- L'usuari clica al botó d'editar distribució 2- El sistema mostra per a cada zona els gossos que es troben en cada gàbia de la zona que s'estigui consultant. Es poden moure els gossos d'una gàbia a qualsevol altra gàbia de la mateixa zona o de les altres zones.
Extensions	

### 3.1.4 Gestió llistes

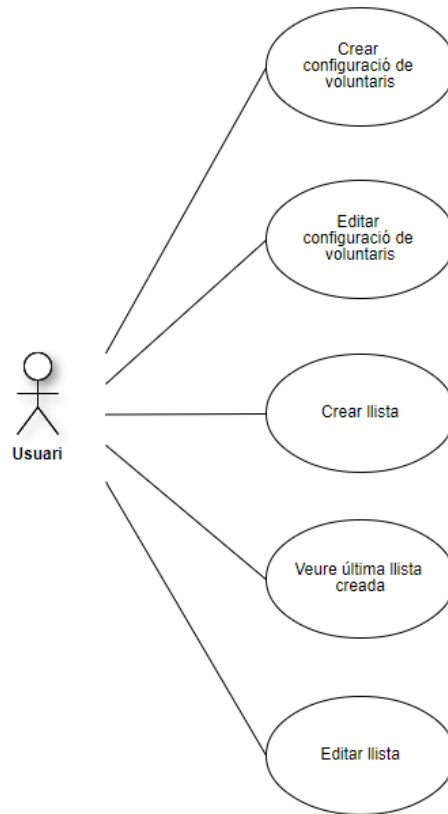


Figura 8. Diagrama de cas d'ús de la gestió de la distribució de gossos

Nom	Crear configuració de voluntaris
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla de gestió de llistes
Disparador	L'usuari clica al botó de crear una nova configuració.
Escenari principal	<p>1- L'usuari clica al botó de crear una nova configuració.</p> <p>2- El sistema mostra una nova pantalla amb una llista amb tots els voluntaris. L'usuari ha de seleccionar els voluntaris que s'hauran d'afegir a la configuració.</p> <p>3- A la pantalla següent, per cada voluntari seleccionat, l'usuari ha de seleccionar quins passejos farà, i quin és el voluntari que netejarà.</p>
Extensions	

Nom	Editar configuració de voluntaris
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla de gestió de llistes. Ha d'existir una configuració de voluntaris ja creada.
Disparador	L'usuari clica al botó d'editar la configuració.
Escenari principal	<p>1- L'usuari clica al botó d'editar la configuració</p> <p>2- El sistema mostra una nova pantalla amb 2 llistes, una amb els voluntaris de la configuració ja existent, i l'altra la llista de la resta de. L'usuari ha de seleccionar els voluntaris que vol afegir/treure de la configuració.</p> <p>3- A la pantalla següent, per cada voluntari seleccionat, l'usuari ha de seleccionar quins passejos farà, i quin és el voluntari que netejarà.</p>
Extensions	

Nom	Crear llista
Actor primari	Usuari
Precondició	L'usuari es troba a la pantalla de gestió de llistes. Ha d'existir una configuració de voluntaris ja creada.
Disparador	L'usuari clica al botó de crear llista.
Escenari principal	<p>1- L'usuari clica al botó de crear llista.</p> <p>2- El sistema, a partir de l'última configuració creada i a partir de la informació dels gossos, voluntaris i la distribució dels gossos a les gàbies, executa l'algoritme per tal de crear una llista que compleixi totes les restriccions necessàries.</p> <p>3- En una nova pantalla es mostra la llista de passejos on es mostra per cada voluntari de la configuració que s'ha fet servir, els gossos que ha de treure a passejar i l'ordre en que els ha de treure. A més, pel voluntari que neteja, es mostra en quin ordre s'han de treure els gossos de la gàbia que no surten a passejar i que es queden a la zona interior de xenils.</p>
Extensions	



Nom	Veure última llista creada
Actor primari	Usuari
Precondició	Ha d'existir una llista ja creada.
Disparador	L'usuari clica al botó de veure l'última llista.
Escenari principal	1- L'usuari clica al botó de veure l'última llista. 2- El sistema mostra en una nova pantalla l'última llista creada. Es mostren tant els gossos que surten a passejar com els gossos que surten per la zona interior de xenils.
Extensions	

Nom	Editar llista
Actor primari	Usuari
Precondició	L'usuari es troba consultant una llista.
Disparador	L'usuari clica al botó d'editar la llista.
Escenari principal	1- L'usuari clica al botó d'editar la llista. 2- El sistema mostra en una nova pantalla per una banda la taula de passejos i per l'altra la taula de neteja amb els gossos que surten per la zona interior de xenils. A més es mostren els gossos que no han estat assignats a cap taula. 3- L'usuari pot moure qualsevol gos a qualsevol posició d'una de les 2 taules o a la llista de gossos assignats. A més, per tal que l'usuari sigui conscient de que els canvis que fa manualment són els correctes, al costat del nom de cada gos hi apareix un cercle que pot tenir diferents colors respecte l'error que pugui donar-se*.
Extensions	

\*Els diferents errors que es poden trobar quan es modifica manualment la llista són:

- 1.Vermell: hi ha com a mínim un gos que comparteix gàbia amb el que té el color vermell que no està assignat al mateix passeig que a la resta.
- 2.Blau: s'ha assignat un gos 'especial' a un voluntari que no el té com a preferit.
- 3.Taronja: s'ha assignat un gos que no surt a passejar a la taula de passejos.
- 4.Verd: no es dona cap dels anteriors errors

A l'hora de fer canvis manuals a la llista ens podem trobar que un gos compleixi més d'un dels errors explicats. L'aplicació informarà de tots els errors que tingui cada gos.

## 3.2 Requisits no funcionals

### 3.2.1 Requisits d'usabilitat

- L'aplicació ha de tenir una interfície intuïtiva i fàcil de fer servir. L'objectiu de l'aplicació és facilitar el fet de fer les llistes que fins ara es feien de forma manual, per tant, la configuració de les dades, la configuració de l'algoritme i l'edició de la llista creada (en cas que es modifiqui) ha de ser ràpid.
- No existiran rols pels usuaris. Tots els usuaris podran accedir a totes les funcionalitats de l'aplicació.
- El sistema s'ha de poder usar només entrar-hi, sense cap mena de tutorial o manual d'usuari. Les categories i els diferents apartats de l'aplicació han de tenir noms clars que descriguin la informació que s'està mostrant per pantalla.

### 3.2.2 Requisits de rendiment

- L'aplicació ha de mostrar/guardar les dades gairebé en temps real per tal que l'usuari no s'hagi d'esperar per realitzar les tasques.

### 3.2.3 Requisits de manteniment i suport

- L'aplicació ha d'estar preparada per a futures millores i ampliacions. Una de les primeres millores que s'implementaran quan es disposi de temps serà llogar una base de dades externa on s'hi pugui pujar i/o baixar la configuració de gossos, voluntaris i distribució per tal que els canvis sobretot de gossos d'una setmana a una altra només els hagi de fer un usuari i que la resta només necessitin consultar la base de dades externa. Això implicarà que fer les llistes serà encara més ràpid.

A part, per evitar que un usuari nou pugui a la base de dades externa una configuració no desitjada, s'afegirà una contrasenya per tal de que l'actualització de la base de dades externa només la puguin fer els usuaris als que se li faciliti aquesta contrasenya.

- L'aplicació s'ha de poder executar en qualsevol dispositiu mòbil en Android.
- En un futur també està pensat adaptar la app per iOS en el cas que hi hagués molts voluntaris de la protectora que no poguessin fer servir la app perquè no disposen d'un dispositiu mòbil Android.

### 3.2.4 Requisits legals

- Tota la informació de l'aplicació s'ha de tractar de manera que compleixi amb la Llei Orgànica de Protecció de Dades (LOPD)

## 4. Disseny aplicació

Donat que en aquest treball s'ha creat una aplicació per dispositius Android s'ha fet servir la arquitectura MVP (Model-View-Presenter) que és la més recomanada a l'hora de treballar amb aquest tipus de projectes [10].

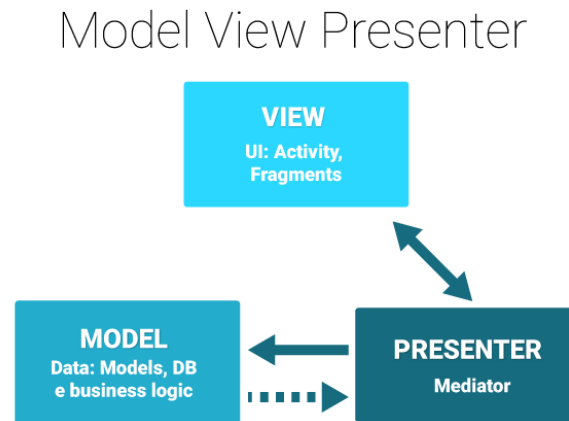


Figura 9. Arquitectura MVP

L'arquitectura MVP consisteix en dividir el sistema en una capa de Vistes, una altra de Presentador i una de Model. El funcionament de cada capa és el següent:

- Capa de Vistes: una interfície passiva que mostra les dades y afegix a una cua les accions del usuari al presentador. En Android, aquesta capa sol estar representada per les classes Activity, Fragment i View.
- Capa de Presentador: Un intermediari que recupera les dades de la capa del Model i els mostra a la vista corresponent. A més també processa les accions del usuari que li envia la vista
- Capa de Model: és una capa que conté la lògica empresarial, controla la creació i modificació de les dades i controla la forma com s'emmagatzemen les dades. En Android, és la capa que accedeix a les dades com la API de la base de dades interna de l'aplicació o la API del servidor remot.

## 4.1 Capa de Vistes

En aquest apartat, es mostrarà el mapa navegacional de l'aplicació i es mostrarà cada pantalla i s'explicarà la seva finalitat.

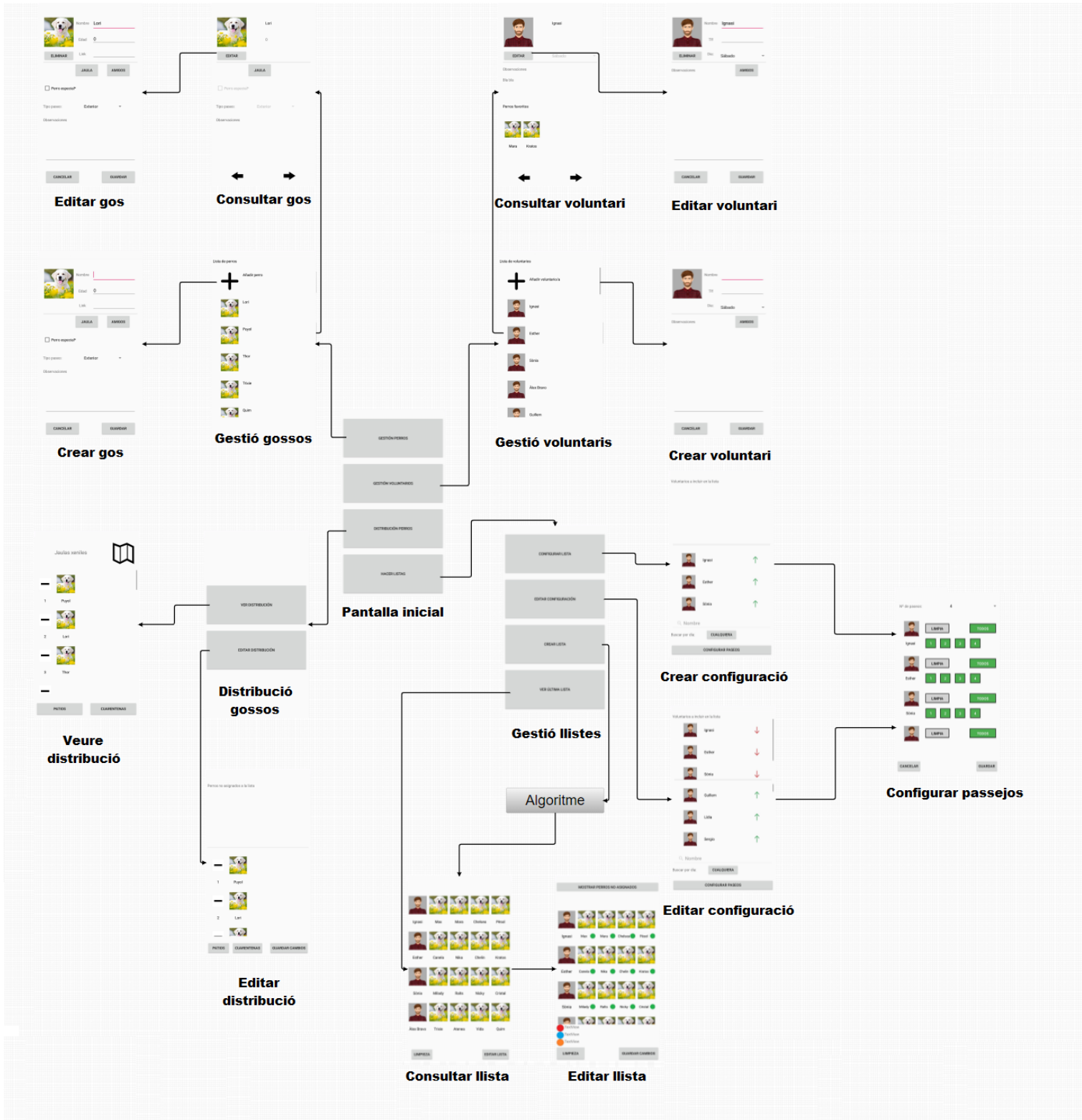


Figura 10. Mapa navegacional de l'aplicació

Com que el temps que he tingut per desenvolupar el projecte ha sigut limitat ja que ho he hagut de compaginar amb la meva feina actual i la prioritat era tenir un algoritme que funcionés i que complís totes les restriccions que tenen en compte les persones que actualment fan aquesta feina a mà, el disseny de totes les pantalles és el més bàsic possible. Totes les pantalles tenen el mínim per tal que els usuaris en la versió que es presentarà de l'aplicació ja puguin treballar-hi.

Millorar el disseny d'aquestes pantalles és una de les futures extensions d'aquest projecte. Sobretot caldrà decidir els patrons de colors que hauran de tenir les pantalles i afegir moltes animacions que ajudarien a fer la interfície més intuïtiva per l'usuari.

Pel que fa a les imatges dels gossos i voluntaris i ja s'ha explicat també en els requisits funcionals, s'ha marcat com a futura extensió poder editar cada gos i cada voluntari amb fotos diferents, que s'obtidran de la galeria de fotos del dispositiu mòbil.

*Pantalla inicial*



Figura 11. Pantalla inicial

Quan s'obra l'aplicació és la primera pantalla que es pot veure. Si és la primera vegada que s'engega després de ser instal·lada s'inicialitzarà la base de dades i s'introduiran les bases de dades bàsiques.

*Gestió de gossos*

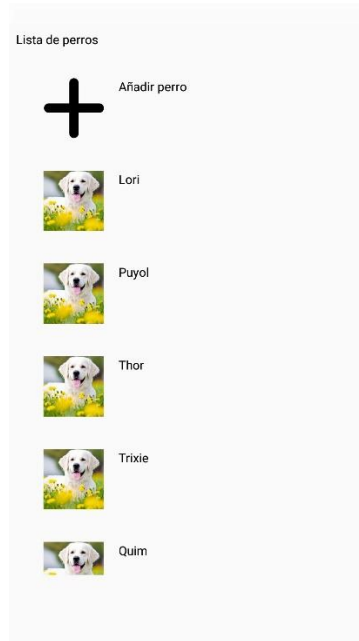


Figura 12. Gestió de gossos

En aquesta pantalla se'ns mostra tots els gossos introduïts a l'aplicació. Es pot seleccionar qualsevol gos de la llista, o bé crear un gos nou clicant al primer element de la llista.

## Crear gos



Figura 13. Crear gos

A l'hora de crear un gos, se'ns obra un formulari amb totes les dades que es poden definir per a cada gos. Aquestes dades són:

- Nom del gos (Camp obligatori)
- Edat del gos
- Link del gos a la pàgina web d'APAN [0]
- Es tracta d'un gos especial? Com a gos especial ens referim a gossos que només poden treure a passejar unes persones concretes ja sigui perquè el gos és molt poruc, és desconfiat amb la gent que no coneix o es difícil de manejar durant el passeig. Si es marca un gos com a especial, a l'hora de crear la llista aquest tipus de gossos només s'assignaran a persones que tinguin a aquest gos com a preferit.
- Tipus passeig. Les opcions poden ser:
  - Surt a passejar ('Exterior' a la app)
  - Surt per la zona interior de xenils ('Interior' a la app)
  - No surt de la gàbia ('No sale' a la app)
- Observacions. Informació extra per tal de donar els principals consells o pautes a seguir a l'hora de treure a passejar el gos.



- Gàbia. La gàbia on es troba el gos s'informa en la part de l'aplicació on es fa la gestió de la distribució. En aquestes pantalles apareix simplement per mostrar la informació de forma ràpida al usuari.
- Amics. S'informen tots els gossos amb el que el gos que s'està creant en aquest cas pot compartir la zona interior de xenils, independentment que estiguin o no a la mateixa gàbia.

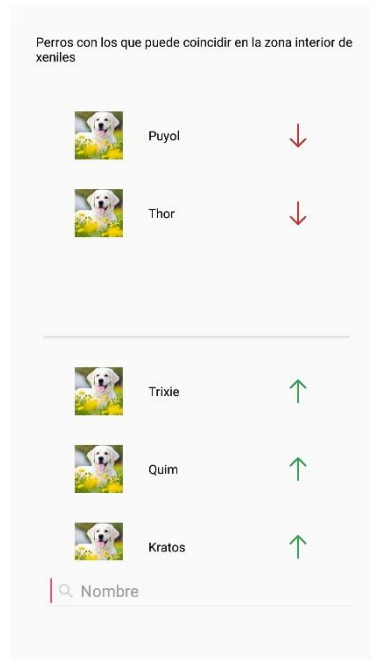


Figura 14. Editar gossos amb els que pot coincidir a la zona interior de xenils

### Consultar gos

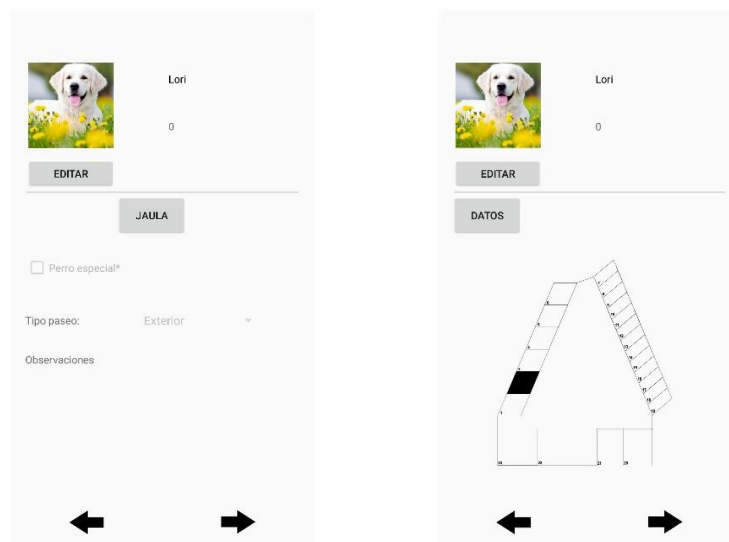


Figura 15. Consultar gos. Dades del gos i gàbia on està ubicat

En aquesta pantalla podem veure les dades del gos i la gàbia on el podem trobar. Llavors podem anar a editar el gos, podem tornar a la pantalla anterior o podem anar veient els diferents perfils dels gossos mitjançant les fletxes que veiem a la part inferior de la pantalla. No es pot editar cap camp.

### *Editar gos*




Figura 16. Editar gos

El funcionament de la pantalla es el mateix que a l'hora de crear un gos però en aquest cas en els camps tenim la informació que té actualment el gos que estem editant. També podem eliminar el gos.

Les pantalles de gestió, creació, edició i consulta dels voluntaris funciona exactament igual que les que hem vist pels gossos.

## Gestió de voluntaris

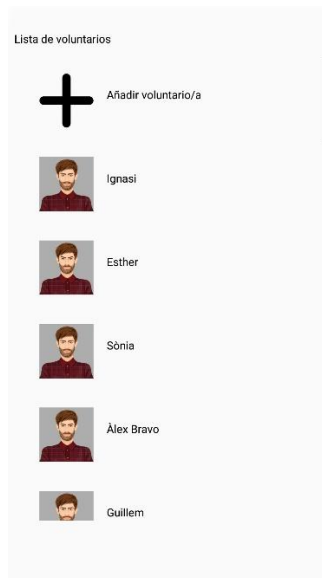


Figura 17. Gestió de voluntaris

## Crear voluntari

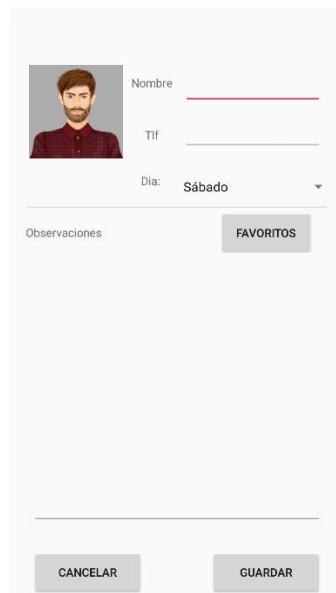


Figura 18. Crear voluntari

A l'hora de crear un voluntari, se'ns obra un formulari amb totes les dades que es poden definir per a cada voluntari. Aquestes dades són:

- Nom del voluntari (Camp obligatori)
- Telèfon del voluntari
- Dia de la setmana quan el voluntari sol venir de voluntari (Dissabtes/Diumenges)
- Es tracta d'un gos especial? Com a gos especial ens referim a gossos que només poden treure a passejar unes persones concretes ja sigui perquè el gos és molt poruc, és desconfiat amb la gent que no coneix o es difícil de manejar durant el passeig. Si es marca un gos com a especial, a l'hora de crear la llista aquest tipus de gossos només s'assignaran a persones que tinguin a aquest gos com a preferit.
- Tipus passeig. Les opcions poden ser:
  - Surt a passejar ('Exterior' a la app)
  - Surt per la zona interior de xenils ('Interior' a la app)
  - No surt de la gàbia ('No sale' a la app)
- Observacions. Informació extra.
- Favorits. S'informen tots els gossos favorits del voluntari.

### Consultar voluntari

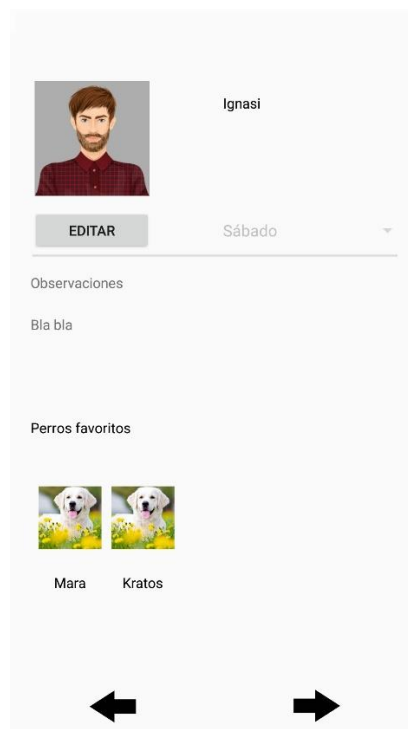
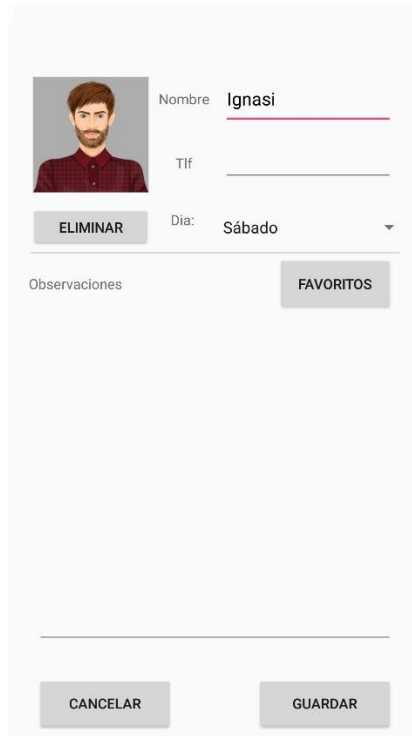


Figura 19. Consultar voluntari

*Editar voluntari*



Nombre **Ignasi**

Tif \_\_\_\_\_

ELIMINAR Dia: Sábado ▾

Observaciones

FAVORITOS

CANCELAR GUARDAR

Figura 20. Editar voluntari

*Gestió distribució dels gossos*



VER DISTRIBUCIÓN

EDITAR DISTRIBUCIÓN

Figura 21. Gestió distribució dels gossos

*Veure distribució*

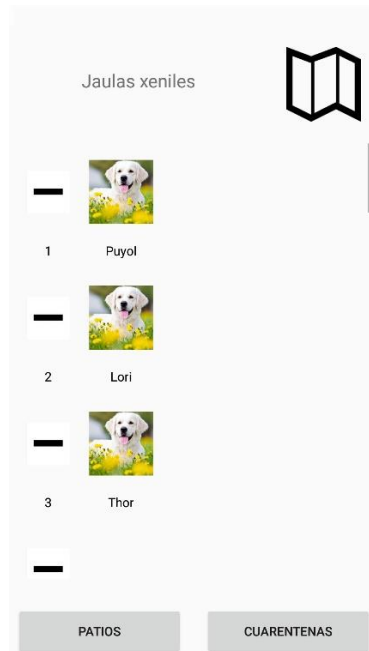


Figura 22. Veure distribució

Per a cada zona de gàbies es mostra quins gossos hi ha a cada gàbia. Per tal de saber a quina gàbia correspon cada número, a la part superior de la pantalla es mostra un mapa on es mostren les gàbies de la zona que s'està consultant.

*Editar distribució*

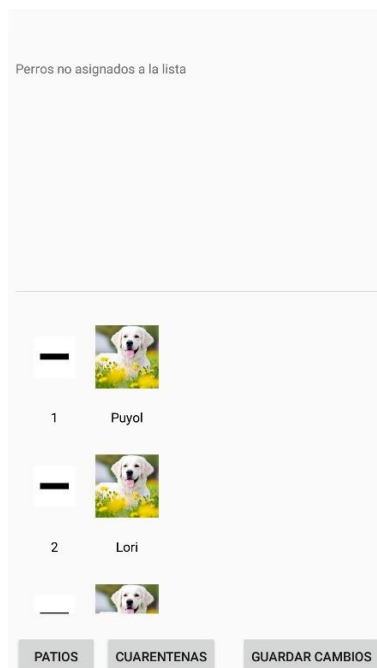


Figura 23. Editar distribució

En aquesta pantalla podem reassignar qualsevol gos a qualsevol gàbia i assignar els gossos que no estan assignats a cap gàbia.

### *Gestió llistes*



Figura 24. Gestió llistes

En aquesta pantalla si no existeix cap configuració creada per la llista només es veurà el primer botó i últim botó. En cas que existeixi una configuració per la creació de la llista aquesta es podrà editar o crear una nova llista.

### Configurar llista

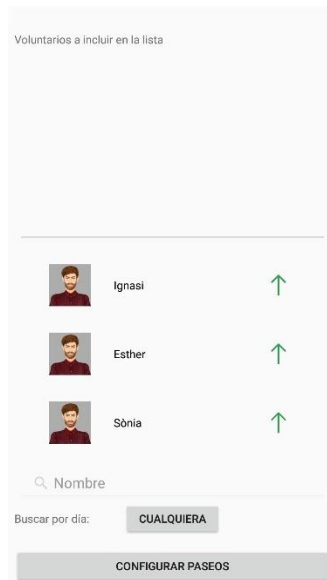


Figura 25. Configurar llista

En aquesta pantalla hem de seleccionar els voluntaris que es volen afegir a la llista/es que es creïn posteriorment. Podem filtrar per nom o per dia de voluntariat per fer d'una forma més ràpida aquesta selecció.

### Configurar passejos



Figura 26. Configurar passejos



En aquesta pantalla ens apareixent tots els voluntaris seleccionats en la pantalla anterior. Hem de configurar el total de passejos que es faran, dels passejos totals quins farà cada voluntari i quin és el voluntari que neteja. Quan es guarda o es cancel·la es retorna a la pantalla de 'Gestió llistes'.

*Editar configuració llista*

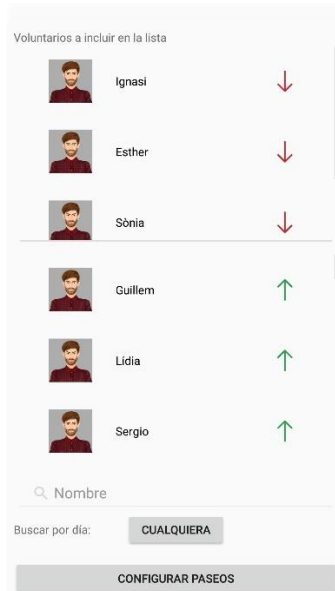


Figura 27. Editar configuració llista

A l'hora d'editar la configuració apareixen com a seleccionats els voluntaris de la configuració existent i a l'hora de configurar passejos també surten seleccionats els passejos de la configuració existent.

*Crear Llista / Veure última llista*

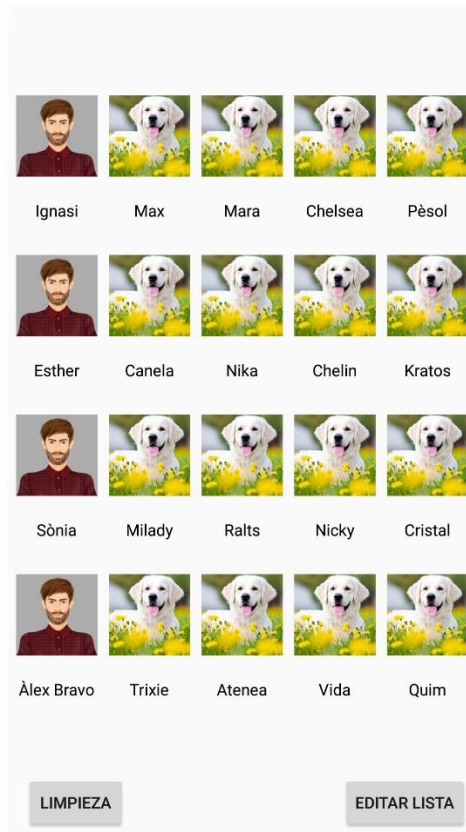


Figura 28. Crear Llista / Veure última llista

Tant el botó de crear llista com la de veure última llista de la pantalla de 'Gestió llistes' obren la mateixa pantalla amb la diferencia que el botó de crear llista primer executa l'algoritme a partir de la configuració de llista existent, les dades dels gossos i voluntaris i les dades de la distribució dels gossos a les gàbies.

*Editar llista*

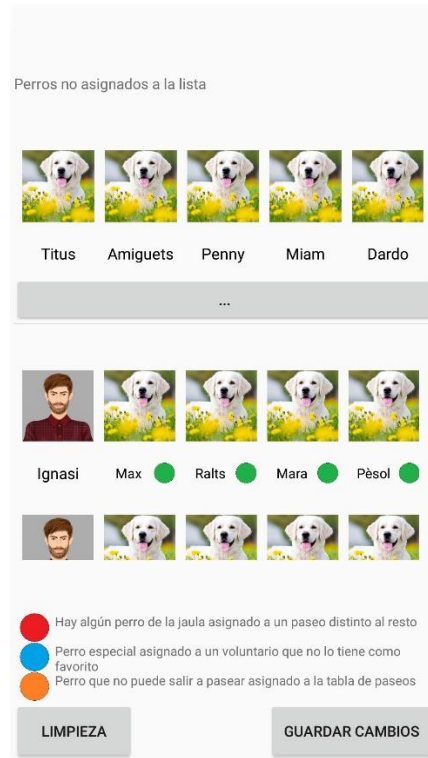


Figura 29. Editar llista

En aquesta pantalla podem modificar la llista que s'ha obtingut de l'algoritme o la última versió de la llista existent. En el cas que els canvis que faci l'usuari facin que la llista resultant sigui incorrecta (no compleixi amb les restriccions) a mesura que es vagin fent canvis es mostraran a la pantalla els possibles errors. D'aquesta manera l'usuari sempre és conscient si la llista es correcta o no. Tot i així, sempre es podrà guardar la llista independentment dels errors que hi hagi ja que es pot donar el cas que per un motiu casual s'hagi de fer un canvi que faci que la llista sigui incorrecta pel que s'ha definit a l'aplicació però correcta pel que fa al escenari real.

Per exemple, si tenim l'escenari d'una gàbia amb 3 gossos on els 3 surten a passejar per l'exterior l'algoritme el que farà serà assignar els 3 al mateix passeig. Si durant la jornada de voluntariat, una família s'interessa per un d'aquests gossos i el volen anar a passejar durant el passeig que li tocava passejar-lo al voluntari, llavors per tal que el voluntari no es quedi sense fer res, se li assignaria un altra gos d'una altra gàbia i el gos que ha sortit a passejar amb la família que s'ha interessat amb ell a l'aplicació es posaria com a no assignat, per tant com dels 3 gossos, només n'hi ha 2 assignats a la taula de passejos ens sortiria que tindríem el primer error, el vermell.

Aquests casos que apareixen de forma eventual l'algoritme i l'aplicació no ho tindran mai en compte perquè són escenaris que es donen molt de tant en tant, i amb l'edició manual de la llista ja es pot solucionar aquests casos tot i que la app digui que hi ha errors en la llista.

## 4.2 Capa de Presentador

Aquest projecte s'ha desenvolupat mitjançant la programació orientada a objectes. En aquesta capa és on es treballa amb els objectes definits ja, o bé després de recuperar les dades de la base de dades o després de recuperar les accions de la capa de vistes. En aquest apartat es mostrarà el diagrama de classes definit.

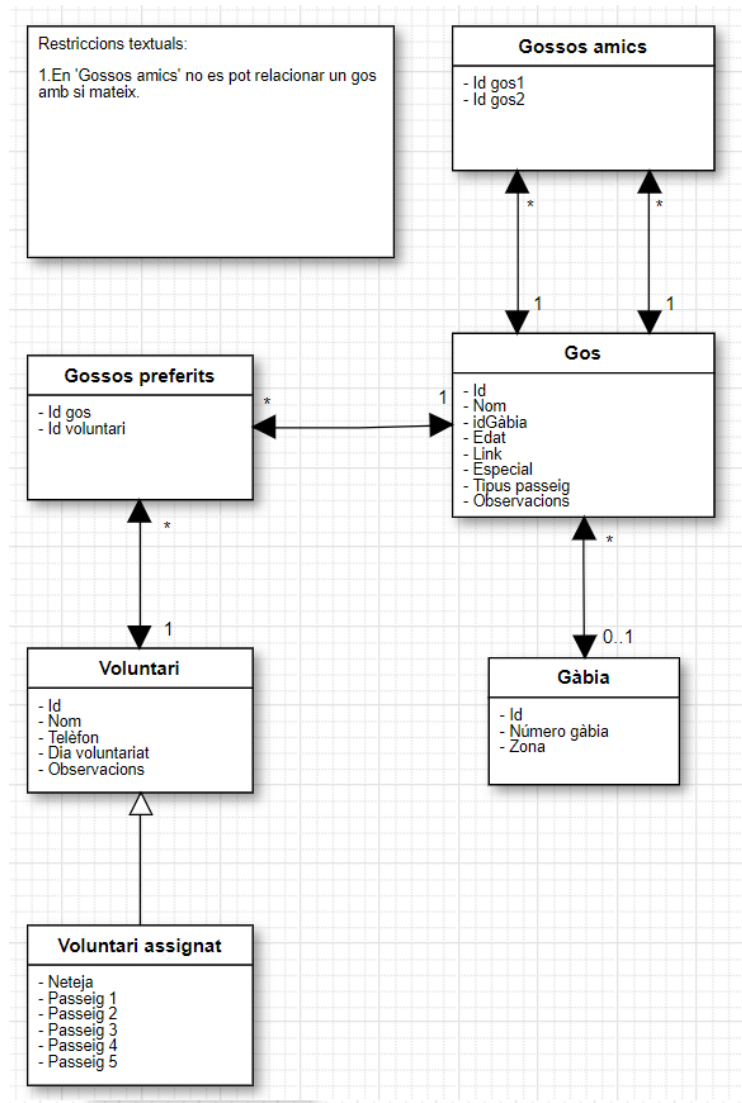


Figura 30. Diagrama de classes

## 4.3 Capa de Model

En la versió actual de l'aplicació les dades s'emmagatzemen en una base interna ubicada en el mateix dispositiu mòbil on s'executa l'aplicació. A partir d'aquesta base de dades, es generen les diferents instàncies dels objectes que hem vist al diagrama de classes. A continuació es mostra el contingut de les taules que hi ha a la base de dades.

*Taula Gossos:* en aquesta taula s'emmagatzema les dades principals de cada gos.

- ID: número únic assignat a cada entrada.
- Nom: nom del gos.
- ID gàbia: id de la gàbia on el gos està assignat en cas que estigui assignat a alguna gàbia.
- Edat: edat del gos.
- Link: link de la pàgina d'APAN on veure el gos.
- Especial: booleà que ens indica si un gos és especial. Com ja s'ha explicat anteriorment, si un gos es marca com a especial a l'hora de fer la llista aquest tipus de gossos només s'assignaran a voluntaris que els tinguin com a gossos preferits. En cas que no hi hagi cap voluntari que compleixi aquest requisit, no s'assignarà aquest gos.
- Tipus passeig: determina si el gos surt a passejar, només surt per la zona interior de xenils o no surt de la gàbia en cap cas.
- Observacions. Informació extra sobre el gos, o sobre pautes a seguir i aspectes a tenir en compte a l'hora de manipular o passejar el gos.

*Taula Voluntaris:* en aquesta taula s'emmagatzema les dades principals de cada voluntari.

- ID: número únic assignat a cada entrada.
- Nom: nom del voluntari.
- Telèfon: telèfon del voluntari.
- Dia voluntariat: informa sobre el dia en que el voluntari sol venir a fer la jornada de voluntariat (Dissabtes o Diumenges)
- Observacions: informació extra.

*Taula Gàbies:* en aquesta taula s'emmagatzema les dades principals de cada gàbia

- ID: número únic assignat a cada entrada.
- Número: número de la gàbia.
- Zona: zona on es troba la gàbia: xenils, patis o quarantenes.

*Taula Gossos preferits:* en aquesta taula s'emmagatzema les relacions entre els voluntaris i els gossos preferits de cada voluntari

- ID: número únic assignat a cada entrada.
- ID voluntari: identificador del voluntari.
- ID gos: identificador del gos

*Taula Gossos amics:* en aquesta taula s'emmagatzema les relacions entre els gossos que si es troben en diferents gàbies poden compartir la zona interior de xenils ja que es porten bé entre ells.

- ID: número únic assignat a cada entrada.
- ID gos 1: identificador del primer gos.
- ID gos 2: identificador del segon gos.

El 'ID gos 1' sempre serà diferents al 'ID gos 2'.

*Taula Passejos:* en aquesta taula s'emmagatzema la informació de la taula de passejos de la llista creada.

- ID: número únic assignat a cada entrada.
- Fila: fila on es troba l'element en la taula de passejos
- Columna: columna on es troba l'element en la taula de passejos.
- ID gos: identificador del gos situat en la fila i columna informades en la taula de passejos.
- ID voluntari: identificador del voluntari situat en la fila i columna informades en la taula de passejos.

*Taula Neteja:* en aquesta taula s'emmagatzema la informació de la taula de neteja de la llista creada. En aquesta taula es guarden els gossos que el voluntari que neteja ha de deixar sortir de la gàbia per la zona interior de passejos mentre la resta de gossos de la taula de passejos surten a passejar.

- ID: número únic assignat a cada entrada.
- Fila: fila on es troba l'element en la taula de neteja
- Columna: columna on es troba l'element en la taula de neteja.
- ID gos: identificador del gos situat en la fila i columna informades en la taula de neteja.

*Taula Voluntaris Seleccionats:* en aquesta taula s'emmagatzema la informació de l'última configuració creada per fer servir en la propera creació d'una llista.

- ID: número únic assignat a cada entrada
- ID voluntari: identificador del voluntari.
- Nom: nom del voluntari.
- Neteja: booleà que informa si el voluntari seleccionat neteja o passeja
- Passeig 1: booleà que informa si el voluntari seleccionat fa el primer passeig.
- Passeig 2: booleà que informa si el voluntari seleccionat fa el segon passeig.
- Passeig 3: booleà que informa si el voluntari seleccionat fa el tercer passeig.
- Passeig 4: booleà que informa si el voluntari seleccionat fa el quart passeig.
- Passeig 5: booleà que informa si el voluntari seleccionat fa el cinquè passeig.

Pel que fa a la funcionalitat de veure l'última llista creada/ editada es necessita guardar l'estat de les dades en l'instant on s'ha creat o modificat la llista per tal de mostrar tota la informació d'aquesta llista encara que s'afegeixin/eliminïn dades. Per tant, el que s'ha fet es tenir les taules duplicades (excepte la de gàbies que sempre té la mateixa informació). D'aquesta manera sempre podem veure l'última llista creada independentment dels canvis que es facin. Aquesta última llista s'actualitzarà quan l'usuari creï una llista nova.



## 5. Algoritme

En aquest apartat veurem l'anàlisi de diferents algoritmes que es van analitzar per tal d'implementar a posteriori el nostre propi algoritme per resoldre el problema que es planteja i la descripció de la versió final del algoritme.

### 5.1 Descripció CSP

Els problemes CSP són una de les alternatives existents que poder modelar el nostre problema i és per l'alternativa que ens hem decantat. Els problemes CSP són problemes NP. Per resoldre el problema que es presenta en aquest projecte s'utilitzarà algun dels algoritmes existents que s'utilitzen per resoldre CSPs.

Un problema de satisfacció de restriccions es pot representar de la següent manera:

1. Un conjunt  $X$  de  $n$  variables  $\{x_1, \dots, x_n\}$
2. Una tupla de dominis finits  $D = \langle D_1, \dots, D_n \rangle$  on el  $i$ -ésim element  $D_i$  és el domini que conté els possibles valors que es poden assignar a la variable  $x_i$ .
3. Un conjunt finit de restriccions  $C = \{c_1, c_2, \dots, c_p\}$ . Cada restricció definida sobre un conjunt de  $k$  variables restringeix els valors que aquestes variables poden tenir per tal d'arribar a una solució vàlida. Totes les restriccions definides en un CSP són conjuntives, és a dir, que una solució al problema ha de complir totes les restriccions.

La instanciació d'una variable és una tupla variable-valor  $(x_i, a_i)$  on a la variable  $x_i$  se li assigna el valor  $a_i$  tal que  $\{a_i \in D_i\}$ .

Una solució a un problema CSP és una assignació  $(a_1, a_2, \dots, a_n)$  de valors a totes les variables del conjunt  $X$  de tal manera que es satisfacin totes les restriccions del conjunt  $C$  del CSP. Un CSP és consistent si té com a mínim una solució.

## 5.2 Alternatives

A l'hora d'analitzar quin algoritme s'havia d'implementar per tal de resoldre el problema que es vol resoldre en aquest projecte es van estudiar els següent algoritmes:

Backtracking simple: algoritme més bàsic per resoldre problemes CSP. Tot i ser un algoritme que el seu funcionament és totalment correcte alhora és molt ineficient. Aquest algoritme realitza una exploració en profunditat de l'espai de búsqueda, instanciant successivament les variables y comprovant a cada nova instanciació que es compleixin totes les restriccions definides. Si aquestes es compleixen es segueix amb la instanciació d'una nova variable. En el cas contrari, s'intenta assignar un nou valor a la última variable instanciada si es possible. Si la última variable instanciada no té més valors al domini es retrocedeix i s'assigna un nou valor a la penúltima variable assignada. Això fa que en el pitjor cas, la solució la trobi després de provar totes les combinacions possibles entre les variables i els valors de cada domini.

Forward checking: Aquest algoritme és una variant del anterior. Per evitar fer tota la exploració en profunditat de l'espai de búsqueda, en cada etapa de la búsqueda, es comproba a partir de la variable actual, tots els valors de les futures variables que no compleixen les restriccions definides pel problema. Els valors de les futures variables que són inconsistents amb l'assignació actual seran temporalment eliminades dels dominis pertinents. Si el domini d'una variable futura es queda buida, la instanciació de la variable actual es desfà i es prova amb un valor nou. Si cap dels valors es consistent llavors és quan es fa servir el backtracking cronològic.

Minimal forward checking: L'algoritme del Minimal forward checking és una versió del Forward Checking que retarda el fet de dur a terme totes les comprovacions de consistències que s'han explicat en el punt anterior fins que és absolutament necessari. En comptes de comprovar les inconsistències de les variables futures a partir de la instanciació de la variable actual, aquest algoritme només fa comprovacions fins que troba el primer valor que és consistent. Després, si l'algoritme retrocedeix llavors es fan la resta de comprovacions amb la resta de valors que encara no s'han comprovat. D'aquesta manera el número de comprovacions que fa el Minimal forward checking és el mateix que en el forward checking en el pitjor cas.

En aquest projecte ens hem basat en buscar un algoritme CSP de les alternatives més clàssiques que existeixen per tal de resoldre el problema que ens ocupa. Però aquests no són els únics algoritmes que s'haguessin pogut fer servir en la implementació d'aquest projecte. Es tracta d'un problema complex i obert i tot i que s'han contemplat aquestes alternatives més clàssiques, sabem que existeixen altres opcions que utilitzen tècniques diferents, com per exemple:

- Algoritmes de l'estil Hill climbing o algoritmes genètics que utilitzen tècniques adaptatives. És a dir, que es comença la búsqueda en un punt aleatori de l'espai de búsqueda i mitjançant heurístics el van modificant fins a trobar una solució després d'unes determinades iteracions. Alguns d'aquests algoritmes no asseguren trobar una solució.
- CSP difusos. S'utilitzen quan existeixen dominis en els que no hi ha una descripció suficient limitada de les restriccions. El concepte de satisfactibilitat de les restriccions passa a ser una qüestió de grau. També es podria parlar de problemes on existeixen restriccions fortes i restriccions dèbils.
- CSP numèrics. Problemes definits amb dominis infinits, requereixen l'aplicació de tècniques matemàtiques específiques per la seva resolució.

## 5.3 Solució adoptada

L'algoritme que s'ha implementat finalment consisteix en un backtracking cronològic simple. Per evitar fer comprovacions de casos inconsistents o per evitar fer l'exploració en profunditat en zones de l'espai de búsqüeda on ja es pot comprovar que no es trobarà cap solució es faran servir les tècniques de backjumping i de forward checking. A més a més, s'utilitzaran heurístics per tal de treballar amb dominis ordenats.

L'objectiu de l'algoritme és que, a partir de la informació dels gossos, la informació dels voluntaris que s'han d'assignar a la llista i la informació de la distribució dels gossos a la protectora, s'obtinguin 2 taules diferents:

1. La taula de passejos on hi haurà la informació de quin gossos s'han assignat a cada voluntari i en l'ordre que els haurà de treure a passejar.
2. La taula de *neteja* on hi haurà la informació dels gossos que s'han de deixar sortir per la zona interior de xenils mentre la resta de gossos passen.

### 5.3.1 Variables

Les variables que s'han definit pel resoldre el problema són:

1. Per la matriu de passejos:  $\{w_{ij}\}$ ,  $i = 0 \dots P-1$ ,  $j = 0 \dots V-1$  on  $P \rightarrow$  quantitat de passejos i  $V \rightarrow$  quantitat de voluntaris.
2. Per la llista de gossos que surten per la zona interior de xenils:  $\{c_i\}$ ,  $i = 0 \dots P-1$  on  $P \rightarrow$  nombre de passejos

### 5.3.2 Dominis

Els dominis son diferents per a cada tipus de variable:

1. Per la matriu de passejos, cada posició de la matriu ( $w_{ij}$ ) tindrà com a domini:  $\{gos_1, gos_2, gos_3, \dots, gos_N\}$  on  $gos_i$  ( $0 \leq i \leq N$ ) és un gos que pot sortir a passejar i on no hi ha cap parell de gossos d'aquest conjunt que estiguin a la mateixa gàbia. Tots els gossos que poden sortir a passejar i no estan al domini s'afegiran a la taula de passejos quan s'assigni el gos de la gàbia que si està al domini. Això es fa per evitar crides innecessàries al mètode de backtracking

2. Per la llista de gossos que surten per la zona interior de xenils, cada posició de la llista tindrà com a domini un conjunt de subllistes ordenades segons els tipus que s'expliquen a continuació:
  1. Tots els subgrups de gossos amics entre ells i de gàbies diferents que poden coincidir a la zona interior de xenils
  2. Per a cada gàbia de xenils, tindrem un subgrup amb tots els gossos que només surten per la zona interior de xenils.
  3. Per a cada gàbia de xenils, tindrem un subgrup amb tots els gossos de la gàbia. Així en cas que no es puguin assignar els gossos que surten a passejar a la llista de passejos perquè no compleixen les restriccions, s'assignaran a la llista de neteja perquè surtin per la zona interior junt amb els companys de la gàbia que també surten per la zona interior.

El domini final per a cada posició de la llista serà una concatenació de totes les llistes de tipus 1,2 i 3 en aquest mateix ordre.

### 5.3.3 Restriccions

Les restriccions que hem de tenir en compte en l'algoritme són:

Obligatòries:

1. Tots els gossos d'una mateixa gàbia han de sortir durant el mateix passeig.
2. Si s'assigna un gos a la zona interior al passeig  $i$ , llavors, els gossos que surten a passejar que comparteixen gàbia amb aquest primer han de sortir a passejar en el passeig  $i$ .

Complementàries:

3. Per facilitar la feina al voluntari que neteja, no s'assignaran més de 6 gàbies de xenils a cap dels passejos.
4. Hi ha gossos especials que només els poden treure a passejar uns voluntaris determinats. Aquest tipus de gossos només s'assignaran si hi ha algun voluntari que el tingui com a preferit.
5. Hi ha gossos de gàbies diferents i que s'assignen per sortir a la zona interior de xenils que es porten bé entre ells. Aquests es poden assignar durant el mateix passeig, per tant, cal aplicar la restricció número 2 per totes les gàbies implicades.

Opcionals:

6. Les preferències de cada voluntari. S'ha de maximitzar el nombre de passejos on el voluntari  $v$  passeja un dels seus gossos preferits.

### 5.3.4 Estructures

Les estructures que fa servir l'algoritme són les següents:

1. Matriu  $P \times V$  on  $P$  = nombre de passejos i  $V$  = nombre de voluntaris. A cada posició de la matriu hi tindrem la variable  $w_{ij}$ .
2. Matriu  $P \times V$  de dominis on  $P$  = nombre de passejos i  $V$  = nombre de voluntaris. A cada posició de la matriu hi tindrem una llista amb tots els gossos que podrien anar a la posició  $w_{ij}$  de la matriu solució anterior.
3. Llista de gossos que surten per la zona interior de xenils. A cada posició de la llista hi tindrem una llista de gossos.
4. Llista de dominis dels gossos que surten per la zona interior de xenils. A cada posició de la llista hi tindrem una llista de llistes de gossos aptes per assignar a la llista anterior.
5. Un graf no dirigit amb pesos. El graf tindrà tants vèrtexs com gossos tinguem. Cada vèrtex(gos) tindrà una relació amb la resta de gossos que surten a passejar o perl la zona interior de xenils. El significat de la relació entre 2 vèrtexs depèn del pes de l'aresta:
  1. Si el pes = 1 llavors són gossos compatibles
  2. Si el pes = 2 llavors són gossos incompatibles
  3. Si el pes = 3 llavors els 2 gossos formen part de la mateixa gàbia.
  4. Si el pes = 4 llavors els 2 gossos surten per la zona interior de xenils, són de diferents gàbies, però es porten bé entre ells.

Aquest graf s'utilitzarà per fer el forward checking (quan s'assigni un gos s'eliminaran de la resta de dominis del mateix passeig tots els gossos incompatibles al gos assignat) i a més es farà servir a l'hora de crear els dominis. El domini s'ordenarà per zona(mitjançant la llista del punt següent) i s'ordenarà per quantitat de gossos que surten a passejar de cada gàbia (mitjançant aquest graf)

6. Llista de totes les gàbies on a la  $i$ -èsima posició de la llista hi tindrem una llista amb tots els gossos que viuen a la gàbia  $i$ .
7. Per poder crear el primer tipus de subllistes dels dominis de la taula de neteges necessitem saber quines gàbies són compatibles, és a dir, necessitem saber els subgrups de gàbies on tots els gossos que surten per la zona interior de xenils es porten tots bé entre ells. Per tal de fer això, s'ha definit un nou graf a partir del que es crea al punt 5 però en aquest cas cada node del graf és una gàbia. El significat de la relació entre 2 vèrtexs en aquest graf també dependrà del pes:
  1. Si el pes = 1. Les gàbies són incompatibles. No tots els gossos de les gàbies a assignar a la zona interior es porten bé entre ells.
  2. Si el pes = 2. Com a mínim una de les 2 gàbies no té cap gos que s'hagi d'assignar per la zona interior de xenils.
  3. Si el pes = 3. Tots els gossos a assignar a la zona interior de les 2 gàbies es porten bé entre ells.

## 5.4 Pseudocodi

En aquest apartat es mostra el pseudocodi de la funció de Backtracking + Validació que s'utilitza per fer les llistes. Són les funcions més importants per veure el funcionament del algoritme. A més a més es mostra el pseudocodi de la funció que crea els subgrups que poden compartir la zona interior de xenils a l'hora de crear el domini de la llista de neteja.

Funció de Backtracking + Validació: la idea principal és que s'assignen els gossos per passejos i per cada passeig primer assignem els gossos de la taula de neteja (el que el voluntari que neteja a de deixar sortir els gossos indicats de la gàbia perquè estiguin a la zona interior de xenils) i després els gossos de la taula de passejos. Per tal d'assignar gossos a la taula de neteja es fan 2 comprovacions: per totes les gàbies dels gossos assignats es comprova que els gossos exteriors d'aquestes gàbies es puguin assignar a la taula de passejos en aquell mateix passeig. Després també es comprova, que d'aquests gossos exteriors a assignar, cada gos que estigui definit com a especial es pugui assignar a un voluntari que el tingui a la llista de gossos preferits.

Una vegada assignats els gossos de x passeig es comencen a assignar gàbies a la taula de passejos. Quan s'assigna una gàbia a un determinat passeig es comprova:

1. Que els gossos especials de la gàbia assignada més els gossos especials pendents d'assignar tenint en compte els gossos assignats a la taula de neteja es puguin assignar cada un a un voluntari que el tingui com a preferit.
2. En el cas que s'hagin assignat els gossos de més d'una gàbia a la taula de neteja al passeig al que hem d'assignar gossos a passejar es comprova si els gossos ja assignats al passeig + els gossos de la gàbia assignada + els gossos que surten a passejar que queden pendents d'assignar tenint en compte les gàbies assignades a la taula de neteja sigui menors o igual al nombre de passejos màxims al passeig on estem assignant la gàbia.
3. En el cas que s'hagin assignat els gossos d'una sola gàbia a la taula de neteja.
  - a. Si els gossos que surten a passejar s'han assignat a la taula de neteja o els gossos que surten a passejar ja estan assignats al passeig es comprova que els gossos assignats al passeig + els gossos de la gàbia assignada sigui menor o igual al nombre de passejos màxims al passeig on estem assignant la gàbia.
  - b. Si els gossos que surten a passejar de la gàbia assignada a la taula de passejos es comprova que els gossos ja assignats + els gossos de la gàbia assignada + els gossos pendents d'assignar de la gàbia assignada a la taula de neteja sigui menor o igual al nombre de passejos màxims al passeig on estem assignant la gàbia.
  - c. Si la gàbia que s'assigna conté gossos que surten per la zona interior de xenils i aquesta gàbia no es l'assignada a la taula de neteja en el passeig on estem assignant la gàbia, no es deixa assignar aquesta gàbia.
  - d. Es comprova que la quantitat de gàbies de xenils assignades sigui inferior o igual a 6.

Quan la validació d'una gàbia a una de les 2 taules és correcta si no s'ha arribat a assignar un gos a l'última posició de la taula de passejos, s'actualitzen els posteriors dominis. A l'hora d'actualitzar els dominis, no només s'elimina la gàbia assignada, sinó que a més, s'eliminen tots els gossos incompatibles amb la gàbia assignada (fent



ús del graf que hem definit en el punt 5 de l'apartat d'Estructures). Una vegada actualitzat el domini es crida a la funció de backtracking per la primera posició posterior sense gos assignat (en el cas de la taula de passejos, quan la validació es correcta, primer s'afegeixen la resta de gossos de la gàbia al passeig. Parlem d'assignar una gàbia però realment només s'assigna un dels gossos de la gàbia però la funció de validar ja té en compte tots els gossos de la gàbia que surten a passejar).

En el cas que la validació sigui incorrecta. Si el domini d'aquella posició té més valors, es crida la funció de backtracking perquè provi d'assignar una altra gàbia a la mateixa posició de la taula. En el cas que no hi hagin més valors, es desassigna la última gàbia assignada, s'actualitza el domini de la posició actual amb tots els valors possibles (s'eliminen els que no són possibles per evitar crides a la funció innecessàries) i es crida la funció de backtracking per la primera posició de les anteriors a l'actual que no té cap gos assignat. A l'hora de desassignar gossos també es té en compte la idea principal. Si la columna actual de la taula de passejos és la columna 0, llavors el que es provarà de reassignar, és un nou valor a la taula de neteja. Si el que desassignem és la gàbia de la taula de neteja, s'assignarà un valor nou a l'última gàbia assignada al passeig anterior al actual.

```
Crida inicial: Backtracking(0,0,0);
```

```
procediment Backtracking(Posició taula neteja , Fila taula passejos, Columna taula passejos) ; si Posició taula neteja == Fila taula passejos s'assignen gossos a la taula de neteja si Posició taula neteja > Fila taula passejos s'assigna gos a la taula de passejos
```

```
iClean <- Posició taula neteja
iRow <- Fila taula passejos
iColumn <- Columna taula passejos
```

**inici**

```
dominiBuit = DominiBuit(iClean,iRow,iColumn)
; En el cas que hi hagi menys gossos que passejos i es permetin assignacions buides hem de consultar si el domini és buit
```

```
Selecció(iClean,iRow,iColumn,dominiBuit)
; Assigna un gos a la taula de passejos o una llista de gossos a la taula de neteja
```

```
si Validació(iClean,iRow,iColumn,dominiBuit)
  AsignarRestaDeLaGàbia(iClean,iRow,iColumn,dominiBuit)
  ActualitzarDomini(iClean,iRow,iColumn,dominiBuit)
  ; S'elimina del domini de passejos o del de neteja el/els gos/gossos assignat/s de les següents posicions a assignar
  si (S'ha assignat un gos/gossos a la taula de passejos)
```

```

    gossosAfeigits = ObtenirNumeroGossosAfeigits();
    si (iClean == Quantitat passejos && iRow == Quantitat passejos - 1 && iColumn +
gossosAfeigits >= Quantitat de passejos del passeig(iRow))
        No fer res -> Tant la taula de neteja com la de passejos han assignat un valor
a cada una de les posicions -> Es una SOLUCIÓ
    si no
        si (Assignat gos/gossos a taula neteja)
            Backtracking(iClean + 1,iRow,iColumn)
        si no
            si(iColumn + gossosAfeigits >= Quantitat de passejos del passeig(iRow))
                Backtracking(iClean, iRow +1, 0)
            si no
                Backtracking(iClean, iRow, iColumn + gossosAfeigits)
            fi si
        fi si
    fi si
si no
    si !DominiBuit(iClean, iRow, iColumn)
        ; Encara queden valor al domini. S'assigna un nou valor a la mateixa posició
de la taula
        Backtracking(iClean, iRow, iColumn)

    si no
        si (iClean == 0 && iRow == 0 && iColumn == 0)
            Posició inicial de les 2 taules
            return; No existeix solució
        si no
            si (Quantitat de gossos < Total passejos a assignar)
                ; Si hi ha menys gossos que passejos a assignar s'han de permetre els
valors buits a la taula de passejos
                si(Fila de la taula de passejos plena)
                    Backtracking(iClean, iRow + 1, iColumn)
                si no
                    Backtracking(iClean, iRow, iColumn + 1)
                fi si
            si (iColumn == 0 && iClean > iRow)
                ; Al passeig i no s'ha pogut assignar cap gos vàlid a la taula de
passejos. Fem backtracking d'aquests gos. S'ha de modificar el/s
gos/gossos assignats a la taula de neteja.
                Desassignem el gos que s'acaba d'assignar de la taula de passejos.
                Actualitzem el domini de la taula de passejos en aquesta
posició
                ; de tal manera que hi hagi de nou tots els gossos no assignats a la
taula de passejos
                Eliminem de la taula de neteja la llista de l'última posició ja
que s'han d'assignar gossos de nou
                Backtracking(iClean - 1, iRow, iColumn)
            si (iColumn == 0)
                ; Al passeig i no s'ha pogut assignar cap gos/gossos vàlids a la taula
de neteja. Fem backtracking d'aquest/s gos/gosos. S'ha de modificar els
gossos de l'última gàbia assignada al passeig i-1
                Actualitzem el domini de la taula de neteja
                Backtracking(iClean, iRow - 1, Quantitat passejos(iRow) -1)
            si
no
                ; Eliminem els gossos de l'última gàbia assignada i assignem una gàbia
al mateix passeig
                gossosÚltimaGàbiaAssignada=ObtenirNumeroGossosÚltimaGàbiaAssignada();
                Actualitzem el domini de la taula de passejos de les posicions on hem
desassignat gossos i de les posicions sense gossos assignats
                Backtracking(iClean, iRow, iColumn - gossosÚltimaGàbiaAssignada)
            fi si
        fi si
    fi si
fi si
fi si
fi

```

```

procediment Validació(Posició taula neteja , Fila taula passejos, Columna taula
passejos, dominiBuit)

```

```

retorna booleà
inici
si dominiBuit
    si(Posició taula neteja == Fila taula passejos)
        ; A la taula de neteja sempre s'ha d'assignar una subllista de gossos
        retorna fals
    si(Posició taula neteja == Fila taula passejos && Quantitat gossos <
Quantitat
passejos)
        ; Si hi ha més passejos que gossos hem de permetre les assignacions
sense gossos a la taula de passejos
        retorna cert
    si no
        retorna fals
si(Posició taula neteja == Fila taula passejos)
    ; S'han assignat gossos a la taula de neteja
    gossosPerAssignar = Obtenir gossos que s'hauran d'assignar a la taula de passejos
    ; Si sabem els gossos que s'han assignat a la taula de neteja podem obtenir
quins gossos dels que surten a passejar s'hauran d'assignar obligatoriament
al passeig 'Fila taula passejos'
    gossosEspecialsPerAssignar = Obtenir gossos especials que s'hauran d'assignar a
la taula de passejos

si(EsPodenAssignarEspecials(gossosEspecialsPerAssignar))
    ; Un gos especial només es pot assignar si hi ha algun voluntari que el té com
a preferit.
    retorna fals
si(gossosPerAssignar > Quantitat passejos del passeig(Fila taula passejos))
    ; No es poden assignar tots els gossos al passeig
    retorna fals
fi si
si no ; S'ha assignat un gos a la taula de passejos
    gàbiesAssignades = ObtenirGàbiesJaAssignades()
    quantitatGossosAssignats = ObtenirGossosJaAssignats()
    ; S'inclouen tots els gossos que surten a passejar de la gàbia de l'últim gos
assignat
    gossosEspecialsPerAssignar = Obtenir gossos especials que s'hauran d'assignar a
la taula de passejos

si(EsPodenAssignarEspecials(gossosEspecialsPerAssignar))
    ; Un gos especial només es pot assignar si hi ha algun voluntari que el té com
a preferit.
    retorna fals
si(Hi ha més d'una gàbia assignada a la taula de neteja al passeig(Fila taula
passejos))
    gossosPerAssignar = Obtenir gossos que s'hauran d'assignar a la taula de
passejos
    si(cuantitatGossosAssignats + gossosPerAssignar > Quantitat passejos del
passeig(Fila taula passejos))
        retorna fals
    fi si

    si(Si només hi ha una gàbia assignada a la taula de neteja i els gossos que surten
a passejar que comparteixen gàbia amb el gos/gossos assignats a la taula de neteja
ja estan assignats && quantitatGossosAssignats > Quantitat passejos del passeig(Fila
taula passejos))
        retorna fals
    si(Si només hi ha una gàbia assignada a la taula de neteja i els gossos que surten
a passejar que comparteixen gàbia amb el gos/gossos assignats a la taula de neteja
NO estan assignats && quantitatGossosAssignats + gossos que surten a passejar que

```

```

comparteixen gàbia amb el gos/gossos assignats a la taula de neteja > Quantitat
passejos del passeig(Fila taula passejos))
    retorna fals
    si(Si només hi ha una gàbia assignada a la taula de neteja i s'hi han assignats
tots els gossos (els que surten a passejar i els que no && quantitatGossosAssignats
> Quantitat passejos del passeig(Fila taula passejos))
        retorna fals
    si(Els gossos que surten per la zona interior de la gàbia del gos que s'acaba
d'assignar no estan assignats a la taula de neteja)
        retorna fals
    si(Si s'han assignat més de 6 gàbies de la zona de xenils)
        retorna fals
    fi si
    retorna cert

fi

```

Per saber si es poden assignar tots els gossos especials a un determinat passeig:

```

procediment EsPodenAssignarEspecials(gossosEspecialsPerAssignar)

retorna booleà
inici
    si (per cada gos especial de gossosEspecialsPerAssignar hi ha un voluntari que
el té com a preferit i el tamany de l'unió de voluntaris > Quantitat gossos en
gossosEspecialsPerAssignar)
        retorna cert
    sino
        retorna fals
fi

```

Pel que fa a la construcció del domini de la taula de neteja per trobar els conjunts de gàbies amb gossos que es porten bé entre ells i per tant, poden coincidir a la zona interior de xenils s'ha implementat un algorisme que a partir del graf de gàbies fa una búsqueda en profunditat i busca tot els subconjunts diferents de gàbies amb gossos interiors i compatibles.

Per cada vèrtex del graf es fa la crida: *ObtenirGrupGabiesDomini(iVertex, vertexs, vertexsVisitats, gàbiesGrupActual, gossosGrupActual)*

```

global GrafJaulas
global GrupsJaTrobats
procediment      ObtenirGrupGabiesDomini(vertex,          vertexs,          vertexsVisitats,
gàbiesGrupActual, gossosGrupActual)

inici
    vertexsVisitats = Marcar 'vertex' com a visitat

    arestes = ObtenirAristasPorPeso(GrafJaulas, pes == 3)

    Per cada aresta
        segonVertex = aresta.ObtenirVertexDestí(vertex)
        si (vertexsVisitats no conté segonVertex)

```

```

//Mirar si el segonVertex no es incompatible amb cap gàbia de la llista
'gàbiesGrupActual'
arestesSegonVertex = ObtenerAristasPorPeso(GrafJaulas, pes == 3)
Comprovar que segonVertex tingui arestes a totes les gàbies de
'gàbiesGrupActual'
jaExisteixGrup = Comprovar si GrupsJaTrobats té un grup igual a
gàbiesGrupActual

si (segonVertex és compatible amb les gàbies de vertexsVisitats i
!jaExisteixGrup)
  Afegir a gàbiesGrupActual la gàbia del vertex actual
  Afegir a gossosGrupActual tots els gossos que s'han d'assignar a la
zona interior de la gàbia del vertex actual
  Afegir a GrupsJaTrobats la llista gossosGrupActual
  ObtenerGrupGabiesDomini(segonVertex, vertexs, vertexsVisitats,
gàbiesGrupActual, gossosGrupActual)
fi si
fi bucle

Treure de gàbiesGrupActual la gàbia del vertex 'iVertex'
Treure de gossosGrupActual tots els gossos que s'han d'assignar a la zona interior de
la gàbia del vertex 'iVertex'
Fi

```

El codi de l'aplicació es podrà trobar al meu GitHub[11].

## 5.5 Problemes trobats

Durant la implementació de l'algoritme es van trobar diferents problemes que es van anar solucionant a mesura que s'anava fent la implementació.

En un principi a cada crida del backtracking l'assignació de gossos a la taula de neteja funcionava igual que en la versió final però quan s'assignaven gossos a la taula de passejos només se n'assignava un a cada crida del mètode 'Backtracking', no existia el concepte d'assignar/desassignar tots els gossos d'una gàbia. Això provocava que quan un gos no es podia assignar en un determinat passeig (per tant, la gàbia no es podia assignar al passeig), els gossos de la gàbia no es desassignaven fins que s'havien provat d'assignar totes les combinacions dels gossos de la gàbia que ja es sabia que no es podia assignar. Degut a això, quan s'executava l'algoritme i s'havien de desassignar diferents gàbies, es feien tantes crides de backtracking que la 'stack trace' s'omplia i es parava l'execució de l'algoritme. Llavors es quan es va introduir la tècnica de **backjumping** i el concepte d'assignar/desassignar gàbia. Al domini de la taula de passejos es va posar només un únic gos dels que passeja per gàbia. Llavors quan s'assigna qualsevol gos, si compleix les restriccions del mètode 'Validació'

s'afegeixen els companys de gàbia del gos assignat que surten a passejar (ja que en la validació ja es comprova si tots els companys poden ser assignats).

Per evitar trobar-nos amb problemes semblants de que durant l'execució de l'algoritme ens quedem sense 'stack trace' s'ha minimitzat la crida a la funció de 'Backtracking' tant com ha estat possible. Mesures que s'han pres a part de la ja explicada anteriorment:

- 1. Ordenació de dominis:** Els dominis de la taula de passejos estan ordenats de gàbies amb més gossos a gàbies amb menys gossos. D'aquesta manera si la taula de passejos està buida, si es vol assignar una gàbia amb molts gossos el més probable es que hi hagi espai suficient per assignar-hi tots els gossos. Com un dels requisits a l'hora de fer la llista es que s'han de prioritzar els gossos que estan a la zona de xenils llavors el que tenim es que cada domini consta d'una doble ordenació. Primer està ordenat per zona (on els primers gossos de la llista sempre seran gossos que estan a la zona de xenils) i després per tamany de gossos a la gàbia de forma decreixent.
- 2. Forward Checking:** Quan s'assigna un gos a la taula de passejos (i els companys de la gàbia que també surten a passejar) a l'hora d'actualitzar els dominis de les següents posicions de la taula, s'eliminen tots els gossos incompatibles amb els gossos assignats (fent ús del graf de gossos).

## 6. Disseny proves

En aquest apartat es detallaran com s'han realitzat les proves per tal de comprovar el funcionament de l'algoritme i de l'aplicació en general. Per fer aquestes proves, s'han seguit les 2 fases que s'han explicat en la planificació:

- Fase de proves de l'algoritme: des de l'aplicació es va crear un apartat de test on es pot provar el funcionament de l'algoritme i les diferents restriccions que s'han implementat. Totes les dades necessàries, per a cada test, s'assignen de forma aleatòria, sense modificar les dades originals, per tal de veure que l'algoritme funciona sempre de forma correcta:
  - Configuració de passejos: es comprova si l'algoritme assigna de forma correcta els passejos configurats a cada voluntari seleccionat.
  - Gossos preferits: es comprova que l'algoritme una vegada a assignat els gossos d'un passeig, assigna cada gos amb un voluntari que el tingui com a preferit si aquest encara no se li ha assignat cap altre gos. Els gossos que no hi hagi cap voluntari que els tingui com a preferit, s'assignen els últims als voluntaris que no tenen gossos assignats.
  - Gossos amics: es comprova que l'algoritme agrupi de forma correcta els gossos de diferents gàbies de xenils que poden compartir la zona interior.
  - Gossos especials: es comprova que l'algoritme assigna els gossos especials de forma correcta. Semblant al test dels gossos preferits però en aquest cas el gos marcat com a especial ha d'estar sempre amb un voluntari que el tingui com a preferit.
- Fase de proves de l'aplicació: es comproven que totes les funcionalitats de l'aplicació funcionin de forma correcta. En aquest cas ja s'ha distribuït la versió de l'aplicació als futurs usuaris per tal de fer proves en casos reals i així arreglar els possibles errors amb els que ens puguem trobar.

*Test: Configuració de passejos*

En la primera taula tenim quins voluntaris s'han seleccionats de forma aleatòria i a quins passejos se'ls hi ha d'assignar un gos (també es fa de forma aleatòria). En la taula de sota, tenim la llista solució una vegada passat l'algoritme amb la configuració de passejos de la primera taula.

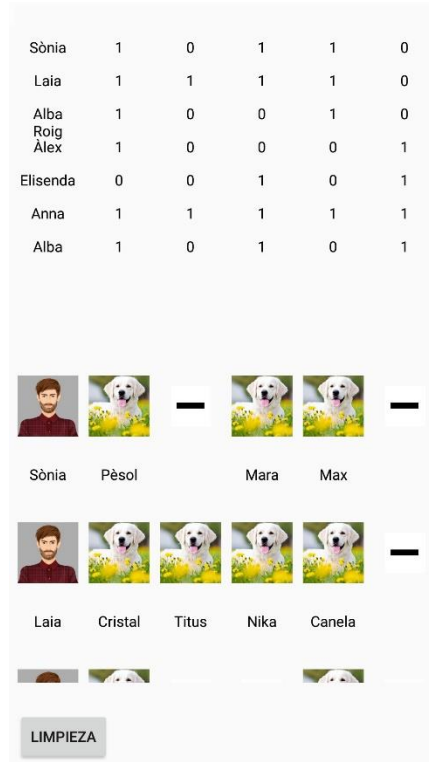


Figura 31. Test: Configuració de passejos

*Test: Gossos preferits*

En la primera taula tenim per cada voluntari seleccionat de forma aleatòria quins són els seus gossos preferits (assignats també de forma aleatòria). En la taula de sota, tenim la llista solució una vegada passat l'algoritme amb la configuració gossos preferits de la primera taula. En aquest cas, per cada gos assignat a un passeig aquest es reassigna al mateix passeig però a un voluntari que el tingui com a preferit i que no tingui encara cap gos assignat. Si no es dona aquest cas, es reassigna aquest gos al mateix passeig i a qualsevol voluntari que encara no tingui cap gos assignat.



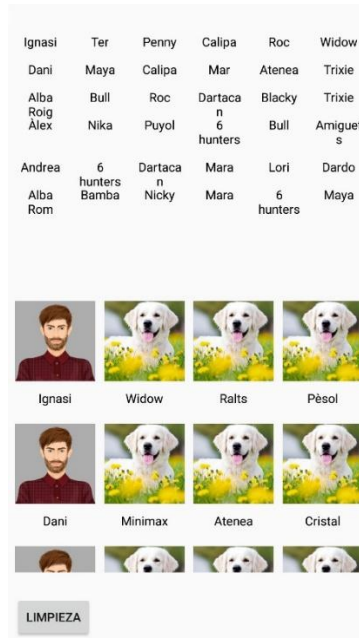


Figura 32. Test: Gossos preferits

*Test: Gossos amics*

En la primera taula tenim, pels gossos que només surten per la zona interior de xenils i són de diferents gàbies, les relacions d'amistat entre aquests gossos assignat de forma aleatòria. En la taula de neteja de la solució resultant podem veure que les assignacions dels diferents subgrups de gossos de diferents gàbies compatibles que poden sortir junts per la zona interior de xenils. En cas que no hi hagi cap subgrup el funcionament a l'hora d'assignar gossos a la taula de neteja és el que es fa per defecte.



Figura 33. Test: Gossos amics

*Test: Gossos especials*

En la primera taula tenim els gossos que s’han marcar com a gossos especials de forma aleatòria. A la segona taula per cada voluntari seleccionat de forma aleatòria quins són els seus gossos preferits (assignats també de forma aleatòria). En la taula de sota, tenim la llista solució una vegada passat l’algoritme amb la configuració gossos preferits de la primera taula. En aquest cas, per cada gos especial, aquest s’assigna a un dels voluntaris que el tingui com a preferit. A diferència dels gossos preferits, si un gos especial està assignat a un determinat passeig, l’algoritme ja s’ha assegurat que aquest voluntari existeix. Per tant, sempre es podrà reassignar. Llavors per cada gos no especial es fa el mateix que s’ha fet en el test de gossos preferits.

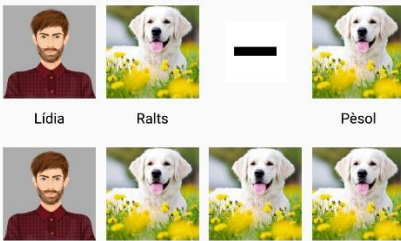
Lori	Puyol	Thor	Trixie	Maxi	Milady
Lídia	Nicky	Mar	Mara	Penny	Widow
Asun	Mara	Kratos	Widow	Nika	Chelsea
Anna B.	Nika	Mar	Dartaca n	Miam	Roc
Montse	Nicky	Dolça	Roc	Miam	Vida
Mambo	Blacky	Toones	Quim	Mar	Bull
					
<input type="button" value="LIMPIEZA"/>					

Figura 34. Test: Gossos especials

# 7. Planificació, anàlisi de costos i sostenibilitat finals

## 7.1 Modificacions realitzades a la planificació

Per tal de veure les diferències entre la planificació inicial, i la realitat recuperem el diagrama de Gantt de l'apartat de planificació inicial del estudi previ:

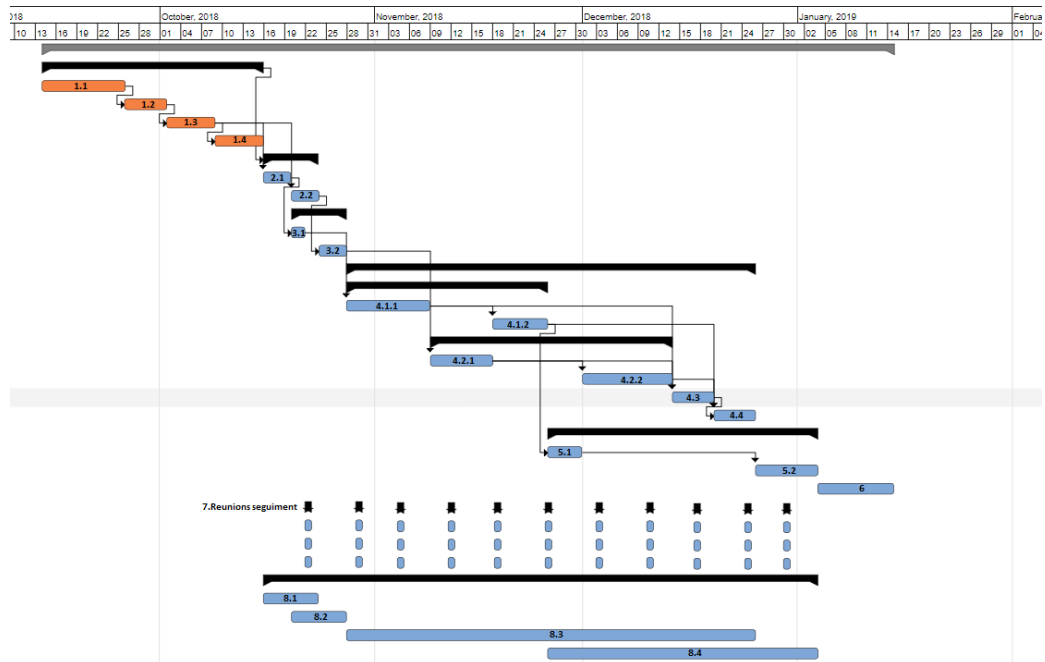


Figura 35: Diagrama de Gantt de la planificació inicial

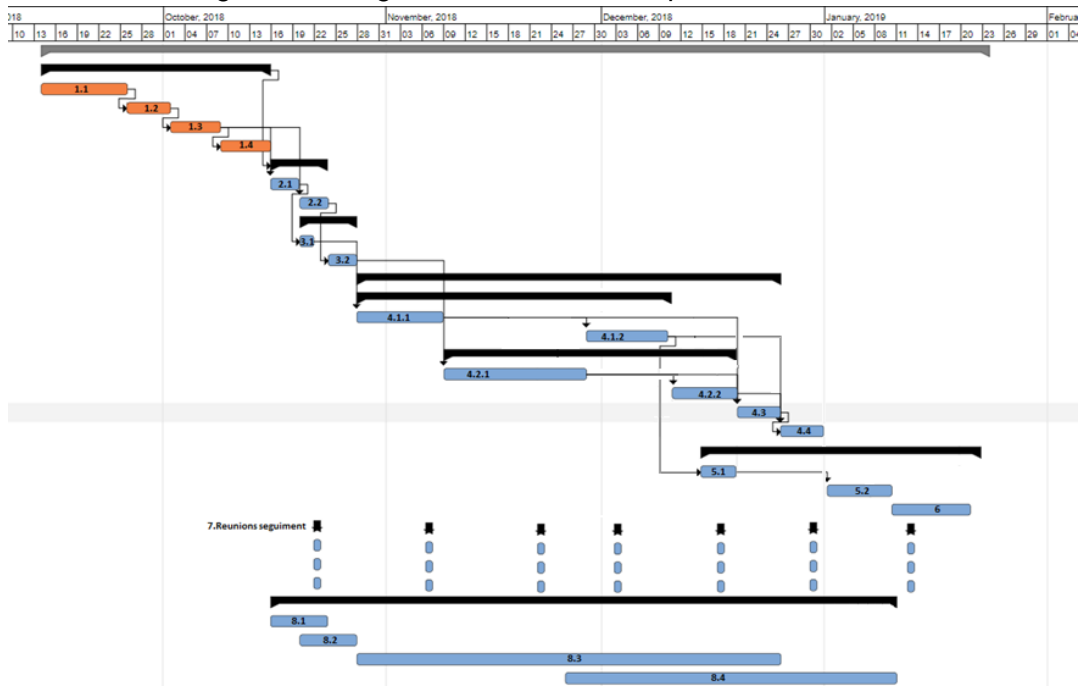


Figura 36. Diagrama del desenvolupament real.

Com podem veure en el segon Diagrama de Gannt, durant el desenvolupament del projecte hi ha hagut uns parell de modificacions:

- La tasca 4.2.2 (implementar la part de la interfície gràfica que està relacionada amb la creació de llistes) es va retrassar més del esperat. Aquest retràs va ser provocat a una sèrie de complicacions que es van trobar amb el funcionament d'algun dels elements i events de la pantalla bàsicament per desconeixença i falta de coneixements a l'hora de desenvolupar aplicacions Android.
- Degut al retràs de la tasca 4.2.2 s'han dedicat menys hores a desenvolupar la resta d'interfície gràfica. Tot i així s'ha pogut desenvolupar el mínim necessari per poder tenir una versió 100% funcional. El que queda pendent de desenvolupament són totes les tasques que fan referència a l'estil de l'aplicació i aspectes visuals i animacions de components perquè l'aplicació sigui més atractiva i còmoda de fer servir per l'usuari.
- A més, s'ha afegit una última reunió amb el director del TFG per acabar de fer els últims retocs de l'aplicació i de la memòria.

Pel que fa al control d'hores que es va plantejar per tal de poder controlar les hores que ha dedicat cada rol al projecte i veure les diferents desviacions, el resultat una vegada acabat el projecte és el següent:

Rol	Hores estimades	Hores reals	Cost estimat	Cost real
Cap de projecte	188,5	195,5	4.637,5€	4.887,5€
Analista	28	28	532€	532€
Dissenyador	20	20	340€	340€
Programador	193,5	256,5	2.709€	3.591€
BetaTester	45	45	810€	810€

Taula 10: Control de desviacions

## 7.2 Metodologia final

La metodologia emprada en aquest projecte ha sigut la mateixa que es va indicar inicialment, el SCRUM, tot i que no s'ha aplicat de la mateixa forma que es volia en un principi. Això es deu a que no s'han pogut realitzar tants desplaçaments a Barcelona com es volia fer en un principi per tal de fer reunions setmanals o quinzenals amb el director del projecte.

Pel que fa al seguiment del desenvolupament del projecte, el que s'ha fet finalment han estat reunions presencials cada 3 o 4 setmanes, una al final de cada tasca de la implementació. En aquestes reunions analitzàvem l'estat de la última tasca realitzada, l'estat del projecte, la feina pendent i els possibles canvis de planificació per tal d'arribar a la data d'entrega. Tot i així, si que s'ha fet de forma setmanal un correu explicant l'estat de la tasca que s'estava desenvolupant, els problemes trobats i les possibles solucions.

Pel que fa el seguiment del projecte amb el client (els voluntaris de la protectora) quan ja es va disposar de la part de la interfície d'usuari dedicada a la configuració i consulta del algoritme i l'edició de les llistes resultants, se'ls hi va començar a distribuir de forma setmanal l'última versió de l'aplicació. A principis d'any ja se'ls hi va distribuir una versió funcional per tal de que poguessin començar a fer les llistes amb l'aplicació tot i que quedaven pendents alguna que altra funcionalitat, com per exemple, faltava la part de mostrar les restriccions que s'incompleixen a l'hora de modificar una llista.

## 7.3 Anàlisi econòmic final

Després d'haver detallat les modificacions que ha patit la planificació inicial del projecte, podem estimar el cost econòmic final de tot el projecte.

Tot i que la tasca que desenvolupament de la part d'interfície d'usuari relacionada amb l'algoritme es va retrassar pel que fa a la planificació els dies de programació totals es mantenen iguals. El que varia són les hores dedicades a la programació. Ja que en un principi es dedicava una mitja de 5 hores diàries al desenvolupament del

projecte, a finals de la tasca que es va retrassar es va a passar a fer entre 6/7 hores diàries dedicades a la programació per tal de poder arribar a la data d'entrega amb el prototip de l'aplicació amb totes les funcionalitats prèviament definides ja implementades. A partir de la segona setmana de gener s'ha tornat a les 5 hores diàries de feina dedicades a redactar la memòria del projecte. El cost real dels recursos humans al final ha quedat de la següent manera:

Tasca	Rol/s implicat/s	Hores	Cost
<b>1.Gestió de projectes</b>	-	<b>60</b>	-
1.1 Abast del projecte i contextualització	Cap de projecte	24	600€
1.2 Planificació temporal	Cap de projecte	9	225€
1.3 Gestió econòmica i sostenibilitat	Cap de projecte	10	250€
1.4 Plec de condicions	Cap de projecte	17	425€
<b>2.Anàlisis del projecte</b>	-	<b>35</b>	-
2.1 Anàlisi algorismes SCP	Analista (80%)	20	304€
	Cap de projecte (20%)		100€
2.2 Anàlisi UI <sup>4</sup>	Analista (80%)	15	228€
	Cap de projecte (20%)		75€
<b>3.Disseny del projecte</b>	-	<b>25</b>	-
3.1 Disseny algoritme	Dissenyador(80%)	7	95,2€
	Cap de projecte (20%)		35€
3.2 Disseny UI	Dissenyador(80%)	18	244,8€
	Cap de projecte(20%)		90€
<b>4.Implementació del projecte</b>	-	<b>285</b>	-
<i>4.1 Implementació algoritme</i>	-	<b>100</b>	-
4.1.1 Estructura bàsica i restriccions obligatòries	Programador(90%)	50	630€
	Cap de projecte(10%)		125€

---

<sup>4</sup> User Interface

4.1.2 Adaptacions del algoritme i resta de restriccions	Programador(90%)	<b>50</b>	630€
	Cap de projecte(10%)		125€
<b>4.2 Desenvolupament UI</b>	-	<b>140</b>	-
4.2.1 Desenvolupament de la UI relacionada amb l'execució del algoritme	Programador(90%)	<b>80</b>	1.008€
	Cap de projecte(10%)		200€
4.2.2 Desenvolupament de la part de la UI d'entrada/consulta de dades	Programador(90%)	<b>60</b>	756€
	Cap de projecte(10%)		150€
4.3 Parametrització	Programador(90%)	25	315€
	Cap de projecte(10%)		62,5€
4.4 Funcions addicionals	Programador(90%)	20	252€
	Cap de projecte(10%)		50€
<b>5.Fase de proves</b>	-	<b>50</b>	-
5.1 Fase de proves inicial	Beta tester(90%)	20	324€
	Cap de projecte(10%)		50€
5.2 Fase de proves(Beta)	Beta tester(90%)	30	486€
	Cap de projecte(10%)		75€
<b>6.Redacció de la memòria, l'annex</b>	Cap de projecte	<b>30</b>	750€
<b>7.Reunions de control</b>	-	<b>30</b>	-
7.1 Seguiment tasques implementades	Cap de projecte	10	250€
7.2 Anàlisi de l'estat del projecte i compliment de les dates	Cap de projecte	10	250€
7.3 Marcar objectius per la setmana següent	Cap de projecte	10	250€

<b>8.Documentació</b>	-	<b>30</b>	-
8.1 Documentació fase de anàlisi	Cap de projecte	6	150€
8.2 Documentació fase de disseny	Cap de projecte	7	175€
8.3 Documentació fase de desenvolupament	Cap de projecte	8	200€
8.4 Documentació fase de proves	Cap de projecte	9	225€
<b>Total</b>		<b>545</b>	<b>10.160,5€</b>

Taula 11: Cost real recursos humans

Respecte al cost de recursos materials, aquestes hores dedicades al projecte de més repercuteix mínimament en l'amortització del hardware indicat prèviament per tant es tracta d'un valor negligible. El mateix passa per l'accés a Internet ja que com el cost és el mateix independentment de les hores ja que la quota és un valor constant és un cost que tampoc em de revisar.

Respecte al les despeses indirectes si que cal revisar-les. S'ha de tenir en compte les hores extres per calcular el total del cost d'energia elèctrica utilitzada. A més s'ha de reduir el preu de transport a la meitat ja que no s'han fet tants desplaçaments que es volien fer en un principi. I per últim, el cost del servidor i domini de Digital Ocean es pot eliminar. Aquest servidor extern és el que s'utilitzarà per implementar una de les primeres extensions que es faran després de l'entrega del prototip de l'aplicació i per falta de temps ja es va definir com a futura extensió gairebé des del principi del desenvolupament del projecte i abans de començar amb la implementació de l'aplicació:



Producte/Servei	Cost	Període	Cost real estimat
Energia elèctrica	0,143 €/KWh	545h	77,94€
Accés a internet	47,73€	5 mesos	238,65€
Transport	10€	5 mesos	50€
Llicència Android	21,67€	Indefinit	21,67€
Servidor i domini (Digital Ocean)	-	-	-
<b>TOTAL</b>			<b>388,26€</b>

Taula 12: Cost real despeses indirectes

I pel que fa al cost real estimat del projecte, s'han tingut en compte les variacions de les hores totals dedicades a la implementació, les modificacions a les despeses indirectes reals i com no hem tingut cap tipus d'imprevist dels que s'havien tingut en compte en el pressupost inicial durant el desenvolupament del projecte no ho tindrem en compte a l'hora de calcular el cost final del projecte.

Concepte	Cost
Despeses directes	10.160,5€
Despeses indirectes	489,3€
Contingència(5%)	532,49€
Despeses imprevistes	0,00€
<b>TOTAL</b>	<b>11.182,29€</b>

Taula 13: Resum del cost final del projecte

D'aquesta manera, el cost total del projecte ha estat 11.182,29€ respecte als 10.172,99€ del pressupost inicial. Això fa que ens doni una desviació negativa de 1.009,3€.

## 8. Conclusions

### 8.1 Assoliment del objectius

Un cop finalitzat amb el desenvolupament del projecte i haver distribuït l'aplicació als voluntaris de la protectora podem concloure que s'han aconseguit amb èxit la majoria d'objectius. Tot i que de moment encara es tracta d'un prototip ja s'està fent servir l'aplicació desenvolupada en aquest projecte per resoldre el problema real i fer més fàcil l'organització dels voluntaris de la protectora. S'hauria pogut desenvolupar una interfície d'usuari més treballada però com ja s'ha comentat, la manca de coneixements, el curt període per desenvolupar el projecte i el poc temps disponible per treballar-hi ja que s'ha hagut de compaginar el desenvolupament de l'aplicació amb la feina actual va fer que es decidís implementar una interfície més bàsica. Això s'ha aconseguit i aquesta compleix amb totes les funcionalitats que s'havien descrit en un principi, per tant podem estar satisfets amb el resultat final.

Pel que fa a la planificació inicial del projecte, tot i patir desviació en una de les tasques de desenvolupament s'ha sapigut gestionar de forma correcta i no hi ha hagut cap impediment per a la finalització del projecte en la data fixada.

La versió proposada tot i que tingui molt marge de millora compleix amb tots els requisits definits, i pel que fa a l'algoritme que crea les llistes, s'ha aconseguit que tingui en compte totes les restriccions necessàries i que actualment es tenen en compte per fer cada llista.

## 8.2 Futures extensions

En aquest apartat es comentaran totes les millores o futures funcionalitats que s'afegiran a l'aplicació en els propers mesos, tot i que algunes d'aquestes millores ja s'han comentat en algun apartat d'aquesta memòria. Aquests canvis que veurem ja estaven com a funcionalitats extres des d'un bon començament ja que ja es va veure que era impossible implementar aquestes funcionalitats per la data d'entrega del projecte.

La primera millora que s'afegirà a l'aplicació és l'accés a una base de dades externes per tal de consultar/modificar l'estat actual de la protectora. Per estat actual de la protectora s'entén que són les dades de tots els gossos, tots els voluntaris i de la distribució de gossos per les gàbies de la protectora. D'aquesta manera, cada setmana, només caldrà que un usuari, dels que tingui permís per modificar les dades de la base de dades externa, faci les modificacions de les dades que han canviat a app i ho pugui a la base de dades externa. La resta d'usuaris, només s'hauran de baixar les dades actualitzades i ja poder començar a crear llistes o consultar les dades dels gossos o de la distribució d'aquests. A l'hora de consultar la informació dels gossos, cada voluntari podrà saber els gossos que traurà a passejar i per tant, ja sap què ha de tenir en compte a l'hora de treure cada gos, com l'aspecte, on es troba el gos i quin material necessitarà (arnés de la mida adequada, corretja llarga/curta, menjar per distreure el gos en les situacions en que s'altera massa...)

L'altra funcionalitat principal que des d'un bon principi ja es va definir com a futura extensió és el tema de les imatges configurables. En comptes de tenir les imatges per defecte pels gossos i voluntaris, l'usuari ha de ser capaç de poder agafar una foto de la seva galeria i posar-la a l'aplicació pel gos o voluntari que vulgui. D'aquesta manera si es va servir l'aplicació com a consulta de dades, entre la foto de cada gos i la informació sobre la gàbia a la que està, qualsevol voluntari podrà saber quin és cada gos sense haver de perdre el temps als cuidadors per preguntal-s'hi sobre els gossos nous o que no sabem en quina gàbia estan.

Es millorarà de forma general l'estil de les pantalles. Es definiran els colors principals i secundaris que ha de tenir la interfície. Es modificarà el disseny dels diferents botons. S'afegiran alertes abans de fer certes accions (per exemple, popups de confirmació al eliminar un gos o voluntari). S'afegiran animacions (per exemple, al editar llista, a l'hora de redimensionar les taules de passejos y neteja).

Es millorarà el funcionament de la pantalla d'editar les llistes. Actualment l'event per començar a moure un gos a una altra posició tarda més del desitjat en llançar-se. En cas que s'hagin de moure molts gossos, és incòmode fer aquests canvis amb el comportament actual. A més a més per tenir la informació completa dels gossos i de les gàbies on es troben s'ha pensat en afegir una funcionalitat per tal que al clicar qualsevol gos es mostri un llistat d'accions. De moment l'acció que s'ha pensat en afegir és redirigir-nos a una altra pantalla on es vegi el mapa de la gàbia on pertany el gos i a sota la llista de tots els gossos que comparteixen aquesta gàbia.

Pel que fa als mapes, en el prototip actual s'han creat de la forma més senzilla i ràpida possible i es mostren d'una forma molt simple. S'hauria d'estudiar de quina manera es podria fer d'una forma millor, però és un altre dels punts que es modificaran.

Pel que fa a futures extensions que es poden fer a molt llarg termini tenim la generalització del problema. L'aplicació creada està 100% pensada per utilitzar a la protectora APAN i a la forma de treballar dels voluntaris. En cas que es disposés de temps i de la informació necessària del funcionament d'altres protectores es podria generalitzar el problema de tal manera que:

- Els usuaris poguessin modificar l'estructura de gàbies i zones de gàbies.
- Generalitzar les restriccions actuals i les restriccions que apareguessin de reunir totes les diferents formes de treballar dels diferents voluntariats i permetre a l'usuari configurar l'algoritme de tal manera que es poguessin escollir les restriccions que es volen tenir en compte.

D'aquesta manera es podria distribuir l'aplicació a qualsevol protectora per tal de facilitar la feina a l'hora d'organitzar l'equip de voluntaris.

## 8.3 Lleis i regulacions

Una llei que hem de tenir en compte a l'hora de desenvolupar la app és la llei orgànica de protecció de dades (LOPD). En el moment que s'introdueixi informació personal dels voluntaris de la protectora o la app demani accedir a la galeria del dispositiu de l'usuari cal garantir la protecció de dades personals, les llibertats públiques i els drets fonamentals de les persones físiques. A més la app sol·licitarà el permís pertinent a l'usuari per poder accedir a la galeria del dispositiu per tal de poder afegir imatges a l'aplicació.

## 8.4 Justificació de les competències tècniques

CCO1: "Tenir un coneixement profund dels principis fonamentals i dels models de la computació i saber-los aplicar per a interpretar, seleccionar, valorar, modelar i crear nous conceptes, teories, usos i desenvolupaments tecnològics, relacionats amb la informàtica". En aquest projecte s'ha desenvolupat una aplicació per dispositius mòbils des de zero. S'han valorat diferents tipus d'algoritmes CSP i s'ha implementat l'algoritme que s'adaptava millor a aquest projecte, tenint en compte, el temps disponible per fer la implementació, com els requisits necessaris de complexitat i eficiència que es necessitaven complir. A més, s'han analitzat diferents tipus d'interfície d'usuaris per tal de dissenyar i desenvolupar una bona interfície per la nostra aplicació tot i que la versió actual del prototip encara tingui molt marge de millora.

CCO1.1: "Avaluar la complexitat computacional d'un problema, conèixer estratègies algorísmiques que puguin dur a la seva resolució, i recomanar, desenvolupar i implementar la que garanteixi el millor rendiment d'acord amb els requisits establerts". A partir del problema existent s'han avaluat les diferents alternatives per tal de resoldre el problema de la millor forma possible. Una vegada es va veure que el que es necessitava modelar era un algoritme CSP vam buscar l'algoritme que millor s'adaptés als requisits que s'havien de complir. Tot i que l'algoritme finalment implementat es basa en l'algoritme de backtracking també s'han utilitzat altres tècniques per tal de que l'algoritme sigués més eficient, fent que no s'explorés tot

l'espai de búsqueda i així aquest aconseguís obtenir la solució en un temps menor. Aquestes altres tècniques ja mencionades anteriorment, són el backjumping i el forward checking.

CCO2: “Desenvolupar de forma efectiva i eficient els algorismes i el software apropiats per a resoldre problemes complexos de computació”. Un dels objectius de l'aplicació, com ja s'ha explicat anteriorment era la implementació d'un algoritme per resoldre el problema que es planteja i que aquest funcionés correctament i amb la eficiència i complexitat que es van definir en el moment de fer l'anàlisi i el disseny d'aquest algoritme. Com hem pogut comprovar en les 2 fases de proves, l'algoritme té el comportament esperat. Les restriccions sempre es compleixen i aquestes es tenen en compte amb la prioritat prèviament definida.

La implementació de l'algoritme com hem vist a la planificació del projecte, s'ha desenvolupat en 2 fases. En un primer es va implementar la funció principal de backtracking i només es tenia en compte la restricció principal, la de que tots els gossos d'una gàbia s'han d'assignar al mateix passeig. Quan es va establir aquesta implementació i es van solucionar els diferents error llavors va ser quan en la segona fase de desenvolupament de l'algoritme es van afegir la resta de restriccions que aquest havia de tenir en compte.

## 8.5 Integració de coneixements

En el desenvolupament de l'aplicació s'han fet servir bastants coneixements que s'han ensenyat durant el grau, ja sigui la utilització d'estructures com grafs o l'aplicació d'algoritmes que s'ensenyaven a les primeres assignatures de programació. Pel que fa al desenvolupament d'una aplicació amb Android Studio només sabia la part més bàsica, la majoria ha estat aprenentatge autònom.

## 8.6 Valoració personal

Haver realitzat aquest projecte ha estat una gran experiència ja que, a part, d'obtenir una gran quantitat de coneixements a l'hora de crear aplicacions Android, també he après a planificar un projecte i gestionar els desviaments amb els que ens hem anat trobant. A més, saber que aquesta aplicació s'estarà utilitzant cada setmana i "no quedarà en un calaix" com la resta de treballs que he anat fet durant el meu pas per la FIB i que aquesta s'utilitzarà per millor la qualitat del voluntariat de la protectora fa que em senti molt satisfet amb la feina feta.

En part, saber que aquesta aplicació seria utilitzada en un futur ha estat una de les principals motivacions que he tingut per tirar el projecte endavant i dedicar-li totes les hores que li he dedicat. No ha estat gens fàcil compaginar aquest projecte amb la meva feina actual i amb la meva vida personal. De fet, quan em vaig endarrerir amb la tasca de desenvolupament de la primera part de la interfície per tal de poder arribar a tenir el prototip a la data d'entrega vaig estar un parell de setmanes que entre la feina i el projecte estava tot el dia programant. Per sort, només vam tenir una desviació respecte la planificació inicial i la bona gestió d'aquesta desviació ha fet que es pugui presentar el prototip amb totes les funcionalitats que es volien presentar implementades.

Tot i que s'han aconseguit els objectius principals del projecte, es seguirà treballant en el desenvolupament d'aquesta aplicació a un ritme diferent al que s'ha treballat durant aquests mesos per tal de tenir en un temps l'aplicació completa amb totes les funcionalitats que de moment han quedat com a futures extensions i també afegir les funcionalitats que puguin anar apareixent durant el pròxims mesos.

En definitiva, com ja he comentat, estic satisfet amb el treball realitzar i amb la versió del prototip actual de l'aplicació, ja que és exactament la idea que tenia del que volia que sigués aquesta aplicació.

## 9. Bibliografia

- [1] Protectora Apan. (2018). Apan. [online] Disponible a: <http://www.protectora-apan.org/ca/> [Accedit 20 Sep. 2018].
- [2] Manyà, F. and Gomes, C. (2018). Técnicas de resolución de problemas de satisfacción de restricciones. Extret de: <https://pdfs.semanticscholar.org/59f1/0fa90e2e3f5a2d490a81f0cbaceb03f0c87d.pdf>
- [3] Bacchus F. and Grove, A. On The Forward Checking Algorithm. Extret de: <http://www.cs.toronto.edu/~fbacchus/Papers/BGCP95.pdf>
- [4] Michael J. Dent & Robert E. Mercer. Minimal forward checking. Extret de: <https://pdfs.semanticscholar.org/a83c/8b7bdf0ca923be3f391afc394abefd99b5b1.pdf>
- [5] SchedulePerfect. (2018). SchedulePerfect: Auto / Rules Based Scheduling Software.. [online] Disponible a: <https://auto-schedule.scheduleperfect.com/> [Accedit el 5 Oct. 2018].
- [6] monday.com. (2018). monday - team management software | monday.com. [online] Disponible a: <https://monday.com/> [Accedit el 5 Oct. 2018].
- [7] Anon, (2018). [online] Disponible a: <https://constraintmodelling.org/minion/> [Accedit el 6 Oct. 2018].
- [8] OptaPlanner. (2018). Constraint satisfaction solver (Java™, Open Source). [online] Disponible a: <https://www.optaplanner.org> [Accedit el 6 Oct. 2018].
- [9] Indeed.es. (2018). Ofertas de trabajo, bolsa de trabajo | Buscar empleo en Indeed España. [online] Disponible a: <https://www.indeed.es/> [Accedit el 13 Oct. 2018].
- [10] Architecture patterns in Android – AndroidPub (2018). AndroidPub. [online] Disponible a: <https://android.jelise.eu/architecture-patterns-in-android-abf99f2b6f70> [Accedit el 10 Gen. 2019]
- [11] Github amb el codi de l'aplicació. GitHub [online] Disponible a: <https://github.com/IgnasiEP/APANListsAndroidApp> [Accedit el 20 Gen. 2019]



# 10. Annex

## 10.1 Manual de l'usuari

En aquest apartat s'explicarà el funcionament de cada pantalla de la versió actual de l'aplicació. Aquest manual també es distribuirà als nous usuaris de l'aplicació per tal de que sàpiguen tot el que es pot fer amb aquesta, totes les dades que es poden consultar i la forma en que s'ha de configurar i executar l'algoritme per tal d'obtenir les llistes. D'aquesta manera tots els usuaris tindran la informació necessària per fer servir l'aplicació de la forma més òptima.

### Pantalla inicial

Des d'aquesta pantalla inicial podem anar als diferents apartats de gestió de dades o a la part de l'aplicació per tal de crear les llistes:

1. Gestió gossos: permet consultar les dades de cada gos, i la gàbia a la que es troba. A més es poden afegir/modificar/eliminar gossos.
2. Gestió voluntaris: permet consultar les dades de cada voluntari. A més es poden afegir/modificar/eliminar voluntaris.
3. Distribució gossos: es pot veure la distribució dels gossos a les gàbies de la protectora i es pot modificar aquesta distribució.
4. Fer llistes: es va a la part de l'aplicació que es dedica a la configuració/creació i consulta de llistes.



Figura 37. Pantalla inicial

## Gestió gossos

En aquesta pantalla podem veure la llista de tots els gossos que té guardats l'aplicació.

1. Si cliquem el primer element de la llista s'obrirà el formulari que ens permet crear un nou gos.
2. Si cliquem qualsevol gos de la llista s'obrirà el perfil del gos clicat i es mostraran les seves dades.

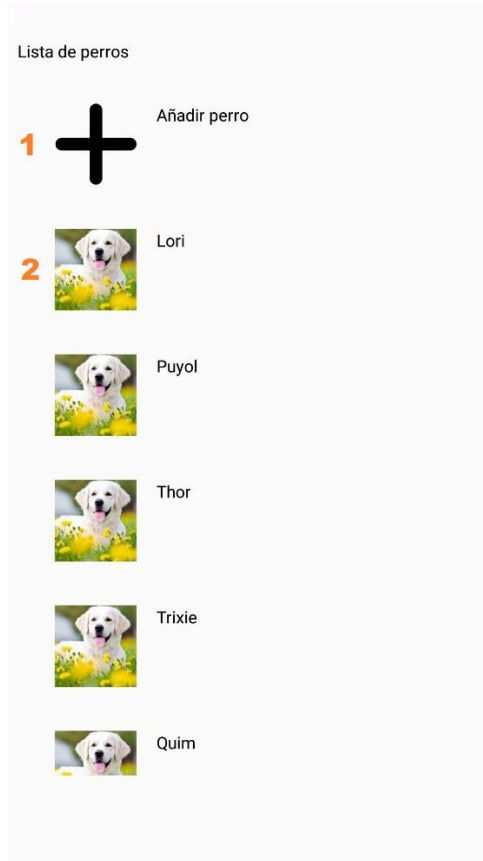



Figura 38. Gestió de gossos

## Crear gos

Al formulari de creació de gossos hi tenim els següents elements:

1. Nom del gos.
2. Edat del gos.
3. Link del gos a la pàgina web APAN.
4. Veure la gàbia on està assignat el gos. Per defecte el gos no s'assigna a cap gàbia. Per assignar-lo a una gàbia s'ha de fer des de l'apartat de 'Gestió de la distribució'.
5. Configurar els amics del gos que s'està creant. Al clicar s'obrirà la segona pantalla de la Figura 39. Per "amics" s'entenen els gossos que poden sortir per la zona interior de xenils però que estan assignats a gàbies diferents.
6. Marca per saber si el gos és especial o no (si es marca com a especial només es podrà assignar als voluntaris que el tinguin com a preferit).
7. Tipus de passeig:
  - a. Exterior: surt a passejar per fora de la protectora.
  - b. Interior: surt per la zona interior de xenils.
  - c. No surt: no surt de la gàbia.
8. Altres observacions a tenir en compte pel gos que s'està creant.
9. Cancel·lar i no crear el gos nou.
10. Crear el gos nou.
11. Dessignar el gos com a amic del que s'està creant.
12. Assignar el gos com a amic del que s'està creant.
13. Buscador per nom dels gossos.



Nombre  **1**

Edad  **2**

Link  **3**

---

**4** JAULA **5** AMIGOS

Perro especial\* **6**



Tipo paseo: Exterior **7**

Observaciones **8**




---

**9** CANCELAR **10** GUARDAR

Perros con los que puede coincidir en la zona interior de xeniles

	Puyol	<b>11</b> ↓
	Thor	↓

---

	Trixie	<b>12</b> ↑
	Quim	↑
	Kratos	↑

**13**

Figura 39. Crear gos

## Consultar gos

En aquesta pantalla tenim:

1. Al clicar s'obrirà el formulari per editar el gos
2. Es pot veure la gàbia a la que està assignat el gos en cas que estigui assignat a alguna.
3. Si el gos està assignat a una gàbia de xenils i té com a tipus de passeig 'Interior' es pot veure els gossos amb els que pot compartir la zona interior de xenils.
4. Tenint en compte la llista de la Figura 38, es consulta el gos que es troba a la posició anterior.
5. Tenint en compte la llista de la Figura 38, es consulta el gos que es troba a la posició posterior.




Lori  
0

**1** EDITAR

**2** JAULA **3**

Perro especial\*

Tipo paseo: Exterior ▼

Observaciones


**4** ←      → **5**

Figura 40. Consultar gos

## Editar gos

El funcionament d'aquesta pantalla és exactament el mateix que a l'hora de crear un gos. Les diferències principals són:

- Les dades del formulari s'omplen amb les del gos actual.
- 4. Al clicar s'elimina el gos.



The screenshot shows a form for editing a dog's profile. It includes a profile picture of a white dog, a name field 'Lori', an age field '0', and a 'Link' field. There are buttons for 'ELIMINAR', 'JAULA', and 'AMIGOS'. A checkbox for 'Perro especial\*' is present, along with a 'Tipo paseo:' dropdown menu set to 'Exterior'. An 'Observaciones' field is at the bottom. At the very bottom are 'CANCELAR' and 'GUARDAR' buttons. Orange numbers 1 through 11 are placed next to various elements to indicate their function or state.

Figura 41. Editar gos

## Gestió voluntaris

En aquesta pantalla podem veure la llista de tots els voluntaris que té guardats l'aplicació.

1. Si cliquem el primer element de la llista s'obrirà el formulari que ens permet crear un nou voluntari.
2. Si cliquem qualsevol voluntari de la llista s'obrirà el perfil del voluntari clicat i es mostraran les seves dades.

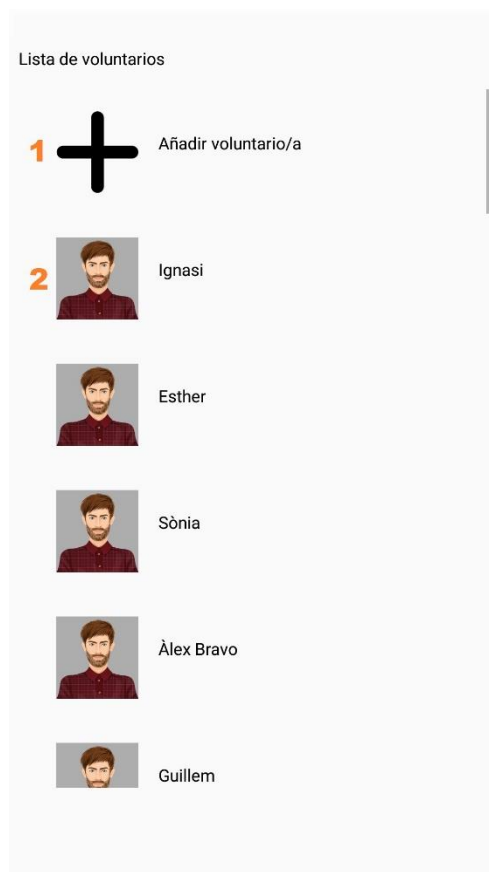
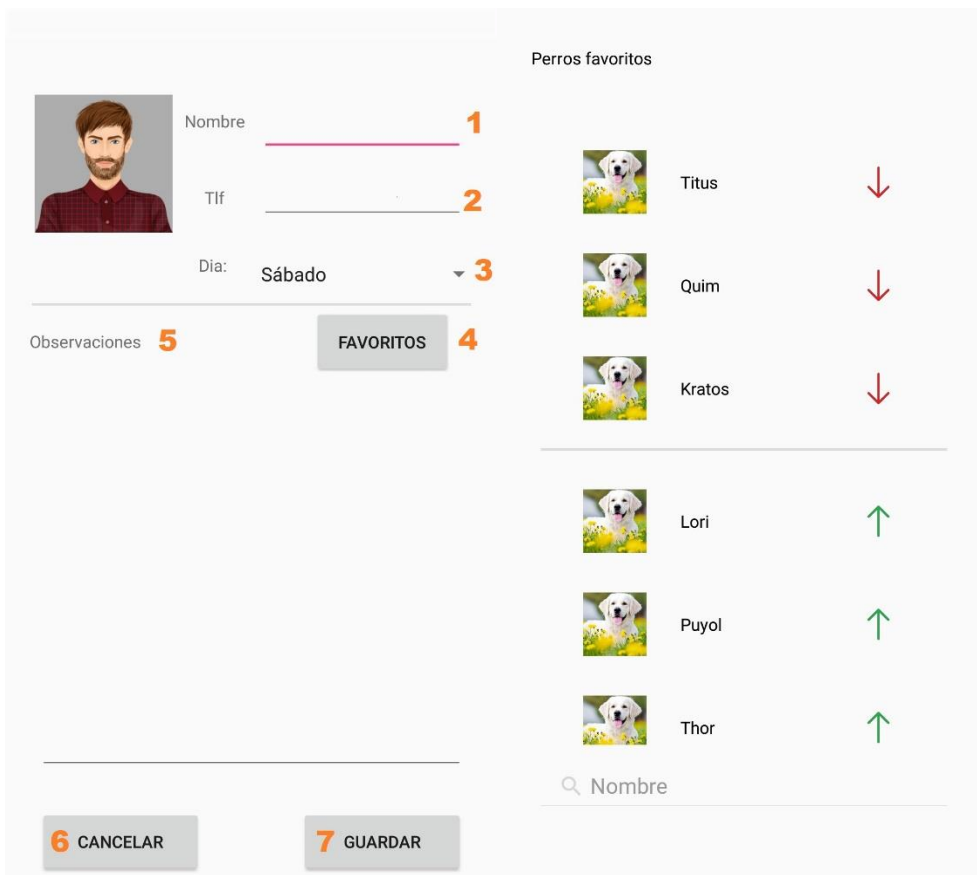


Figura 42. Gestió voluntaris

## Crear voluntari

Al formulari de creació de voluntaris hi tenim els següents elements:

1. Nom del voluntari.
2. Telèfon del voluntari.
3. Dia en el que sol fer el voluntariat.
4. Configurar els gossos preferits del nou voluntari.
5. Altres dades.
6. Cancel·lar la creació del nou voluntari.
7. Crear el nou voluntari.



The screenshot shows a web form for creating a volunteer. On the left, there is a profile picture of a man and a form with the following fields: 'Nombre' (labeled 1), 'Tif' (labeled 2), 'Dia:' with a dropdown menu set to 'Sábado' (labeled 3), and 'Observaciones' (labeled 5). Below these fields is a 'FAVORITOS' button (labeled 4). At the bottom of the form are two buttons: '6 CANCELAR' and '7 GUARDAR'. On the right side, there is a section titled 'Perros favoritos' containing a list of dogs: Titus, Quim, Kratos, Lori, Puyol, and Thor. Each dog entry has a small image and a red downward arrow (for Titus, Quim, Kratos) or a green upward arrow (for Lori, Puyol, Thor). Below the list is a search bar with the placeholder text 'Nombre'.

Figura 43. Crear voluntari



## Consultar voluntari

En aquesta pantalla hi tenim:

1. Al clicar entrem al formulari per editar el voluntari.
2. Tenint en compte la llista de la Figura 42, es consulta el voluntari que es troba a la posició anterior.
3. Tenint en compte la llista de la Figura 42, es consulta el voluntari que es troba a la posició posterior.

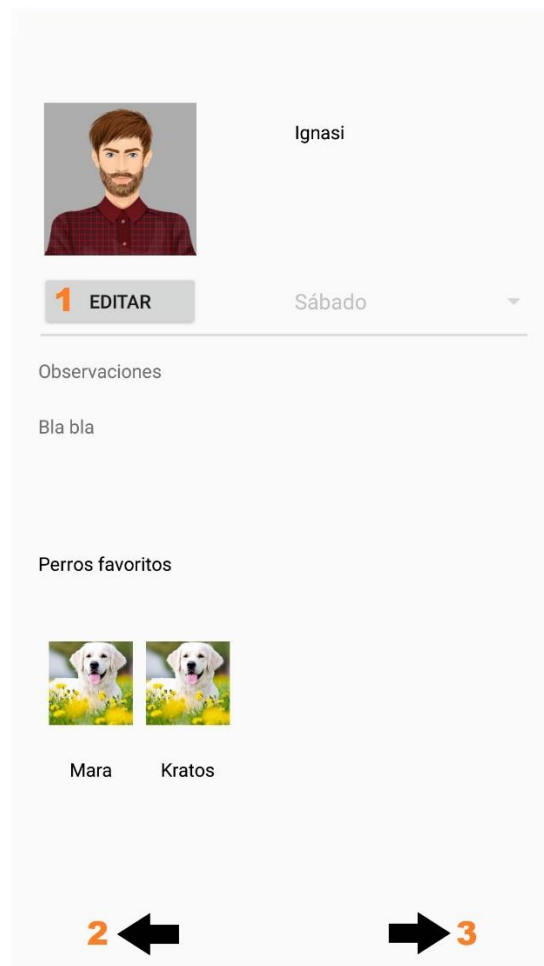
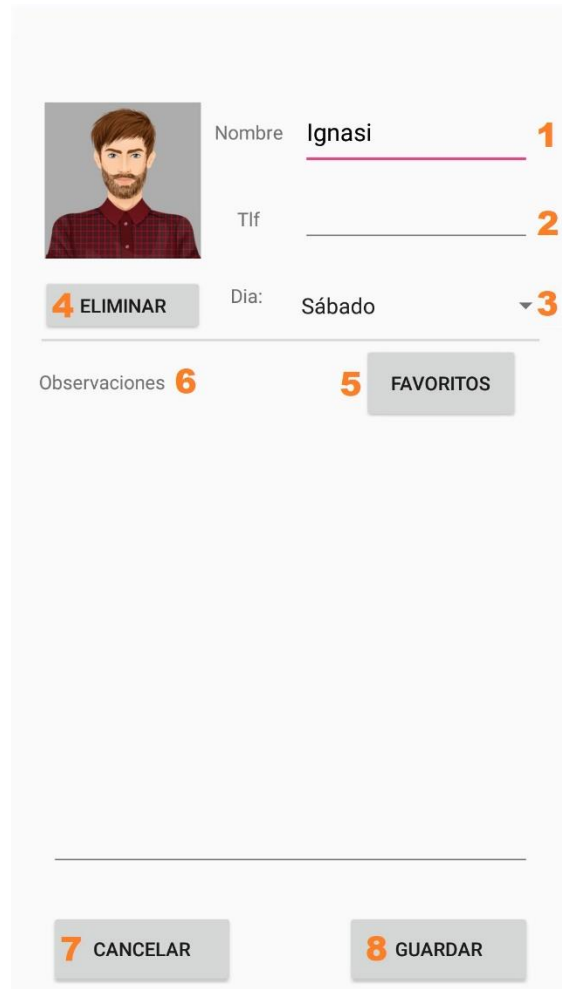


Figura 44. Consultar voluntari

## Editar voluntari

El funcionament d'aquesta pantalla és exactament el mateix que a l'hora de crear un voluntari. Les diferències principals són:

- Les dades del formulari s'omplen amb les del voluntari actual.
- 4. Al clicar s'elimina el voluntari.



The screenshot shows a mobile application interface for editing a volunteer. At the top left is a profile picture of a man with a beard. To the right of the picture are three input fields: 'Nombre' with the value 'Ignasi', 'Tlf' (empty), and 'Dia:' with a dropdown menu showing 'Sábado'. Below these fields is a button labeled '4 ELIMINAR'. Underneath is a section for 'Observaciones' (6) and a button labeled '5 FAVORITOS'. At the bottom of the form are two buttons: '7 CANCELAR' and '8 GUARDAR'. Orange numbers 1 through 8 are placed next to the corresponding elements to indicate their function or location.

Figura 45. Editar voluntari

## Gestió distribució

En aquesta pantalla hi tenim les opcions:

1. Consultar la distribució actual dels gossos a les gàbies.
2. Editar la distribució dels gossos a les gàbies.



Figura 46. Gestió distribució

## Consultar distribució

En aquesta pantalla hi tenim:

1. Al clicar, s'obrirà el mapa amb les gàbies de la zona actual que estem consultant. El número que surt al principi de cada fila de la taula és el número que correspon a la gàbia que podem veure al mapa.
2. Al clicar, es passa a consultar la zona de gàbies que s'indica en el text del botó.
3. Al clicar, es passa a consultar la zona de gàbies que s'indica en el text del botó.

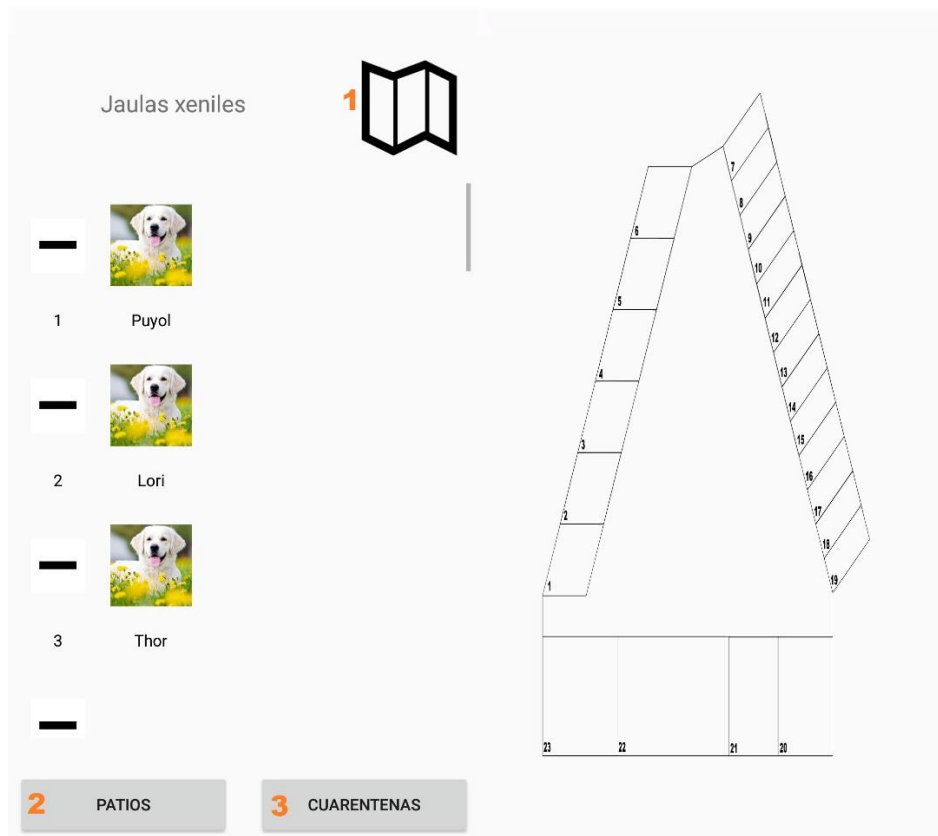


Figura 47. Veure distribució

## Editar distribució

En aquesta pantalla hi tenim:

1. Llista de tots els gossos que no tenen cap gàbia assignada.
2. Al deixar clicant qualsevol gos, aquest es podrà començar a arrossegar.
  - a. Si s'arrossega s'obre un altre gos, ja sigui de la taula visible o un gos de la llista de gossos no assignats, aquests intercanviaran posicions.
  - b. Si s'arrossega al final de la llista de gossos no assignats o en un espai en blanc de la taula visible aquest s'afegeix a la posició destí i en la posició original queda un espai en blanc.
  - c. Si mentre s'arrossega es passa pels botons (3 i 4) es canviarà la taula que es mostra per la taula de gàbies de la zona a la que es vol reassignar el gos. Si a la posició destí ja hi ha un gos el funcionament és com el del punt a)
3. Al clicar, es passa a consultar la zona de gàbies que s'indica en el text del botó.
4. Al clicar, es passa a consultar la zona de gàbies que s'indica en el text del botó.
5. Es guarden els canvis en la distribució.

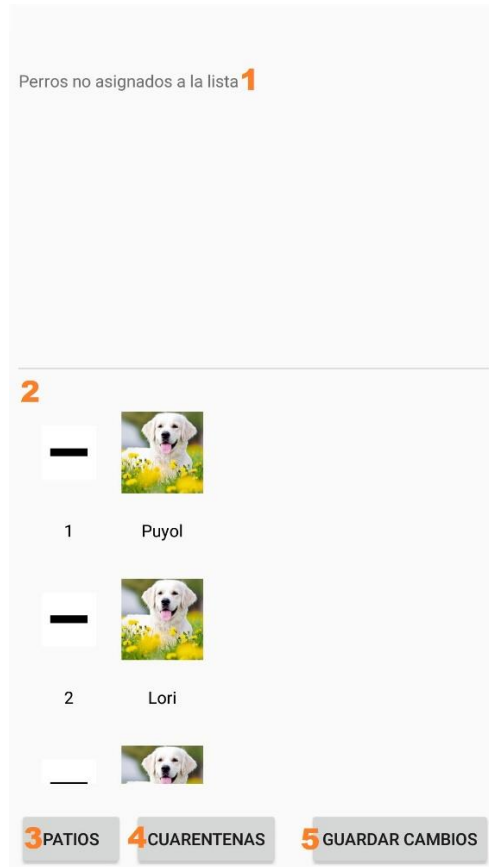


Figura 48. Editar distribució

## Gestió llistes

En aquesta pantalla hi tenim les següents opcions:

1. Crear configuració de la propera llista a crear
2. Editar configuració de la propera llista a crear (només visible si ja existeix una configuració)
3. Crear llista (només visible si ja existeix una configuració)
4. Veure última llista creada (només visible si existeix una llista)



Figura 49. Gestió llistes

## Configurar llista

A l'hora de fer la configuració de la llista primer hem de seleccionar els voluntaris que s'han de tenir en compte a l'hora de crear la següent llista:

1. Llista del voluntaris afegits a la configuració actual.
2. Al clicar, s'afegeix el voluntari a la configuració.
3. Buscar per nom.
4. Filtre de voluntaris per dia del voluntariat. Al clicar, s'alternen les opcions:
  - a. Qualsevol dia
  - b. Dissabte
  - c. Diumenge
5. Al clicar, es passa a la pantalla de configurar els passejos pels voluntaris seleccionats.

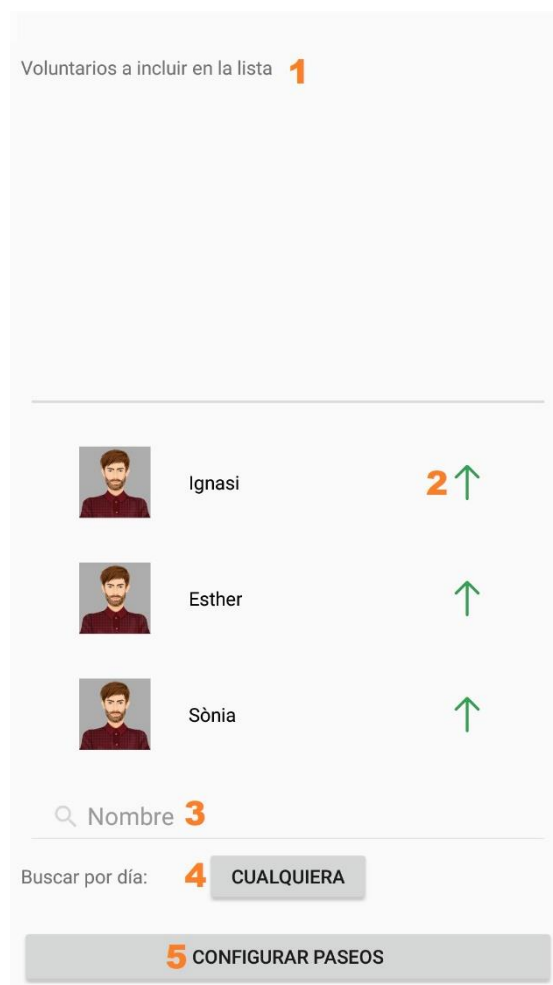



Figura 50. Seleccionar voluntaris

## Configurar pasejos


Una vegada seleccionats els voluntaris a incloure a la configuració, hem de configurar quins pasejos fa cada un d'aquests voluntaris:

1. Es tria s'hi la llista tindrà 3,4 o 5 pasejos
2. Per cada voluntari es seleccionen quins pasejos farà cada voluntari, o en el cas que li toqui netejar xenils es marcarà com a que neteja.
3. Es cancel·la la creació de la configuració.
4. Es guarda la nova configuració.


Nº de paseos: **1** 4

**2**  LIMPIA  TODOS


Ignasi  1  2  3  4

 LIMPIA  TODOS

Esther  1  2  3  4

 LIMPIA  TODOS

Sònia  1  2  3  4

 LIMPIA  TODOS

**3** CANCEL·LAR **4** GUARDAR

Figura 51. Configurar pasejos



## Editar configuració

El funcionament d'aquesta pantalla és exactament el mateix que el de la Figura 50. La diferència és que en aquest cas la llista de voluntaris de la configuració en comptes de ser buida apareix amb els voluntaris que ja formen part de la configuració:

- 1. Es treu el voluntari de la configuració
- 5. A l'hora de configurar els passejos, els voluntaris que ja formaven part de la configuració apareixeran amb la configuració de passejos actual.

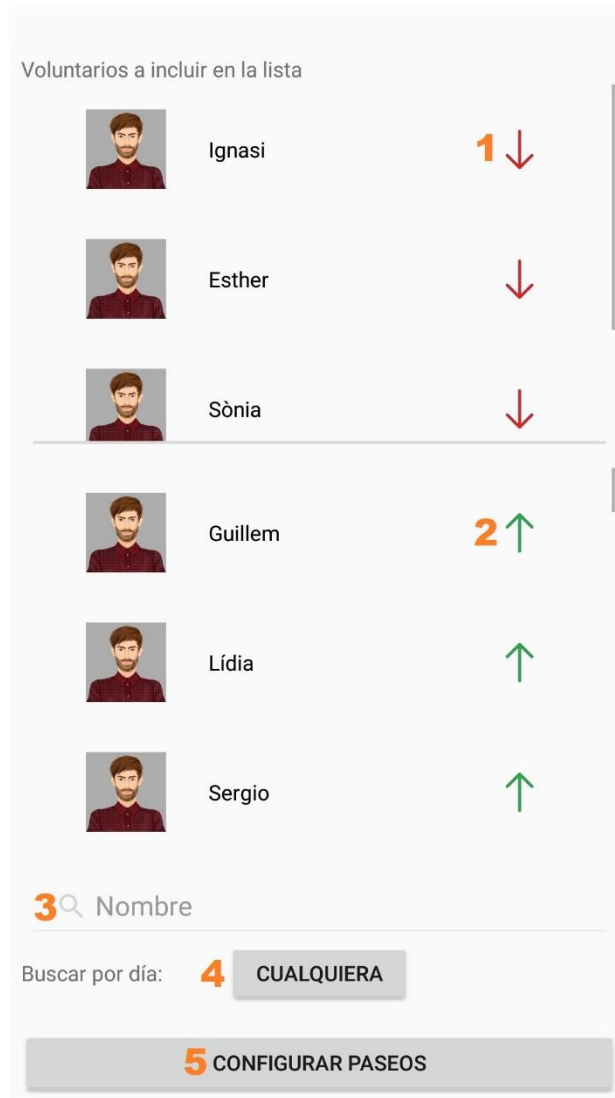


Figura 52. Editar configuració

## Crear llista/ Veure última llista

A l'hora de mostrar la llista (independentment si s'executa l'algoritme o no) observarem la següent pantalla:

1. Al clicar, veurem la llista dels gossos que surten per la zona interior de xenils mentre la resta de gossos passegen.
2. Al clicar, s'obrirà la pantalla que ens permet editar de forma manual la llista actual.

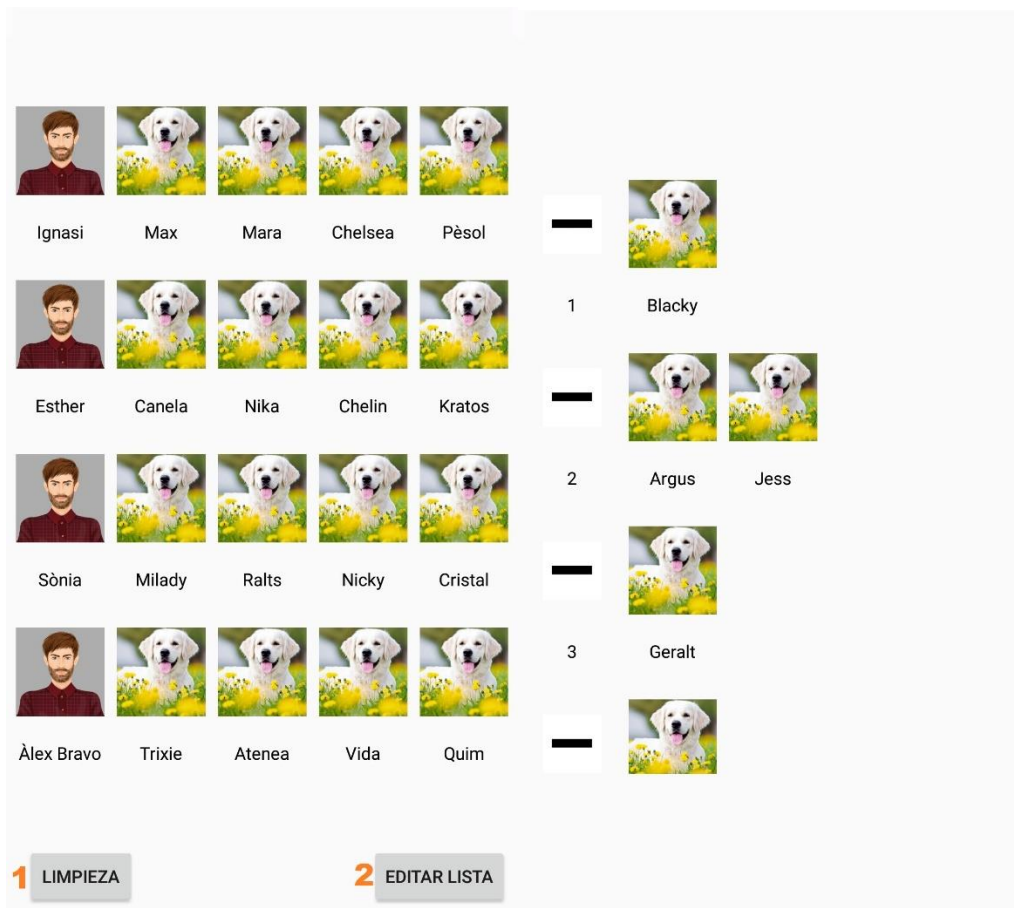


Figura 53. Crear/ veure última llista. Llista de passejos/ Llista neteja

## Editar llista

La pantalla per fer els canvis manuals de la llista és la més complexa:

1. Es mostra la llista de gossos que no estan assignats ni a la taula de passejos ni a la taula de neteja.
2. Cada gos, al clicar-lo es pot començar a arrossegar:
  - a. Si a la posició destí del gos arrossegat no hi ha cap altre gos aquest es posarà a la posició destí de la taula on es deixi d'arrossegar i a la posició original hi quedarà un espai buit.
  - b. Si a la posició destí hi ha un altre gos, el gos arrossegat i aquest segon intercanvien les posicions.
  - c. Si al arrossegar un gos es passa per sobre del botó 'Limpieza' o 'Paseos' (quan es mostra la llista de neteja). La llista que és visible s'ocultarà i la llista oculta apareixerà per pantalla. D'aquesta manera es poden moure gossos entre les 2 taules.
3. Informació dels errors. S'actualitza cada vegada que s'arrossega un gos.
4. Veure la llista de neteja.
5. Guardar els canvis.

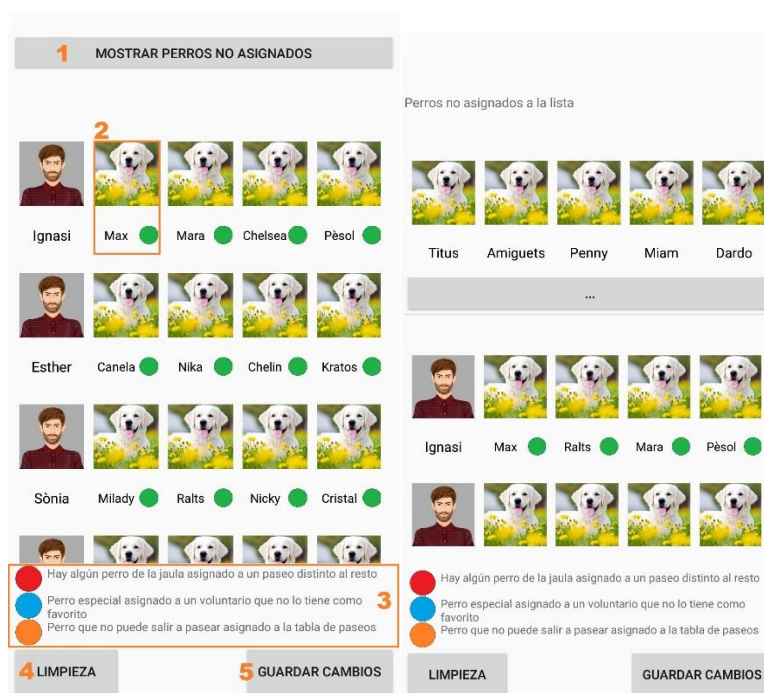


Figura 54. Editar llista