



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicació web per a refugis d'animals

Treball Fi de Grau

Grau en Enginyeria Informàtica

Autor: Artur Monter Anduix

Tutor: Oliver Gil, José Salvador

2020-2021

Resum

En aquest treball mostrarem el desenvolupament d'una aplicació web la qual tractarà, per un costat, d'ajudar en l'adopció d'animals abandonats que es troben en protectores, i per l'altre, facilitar la recerca de mascotes perdudes.

Aquesta web, per una banda, permetrà als usuaris poder elegir dintre d'un gran catàleg d'animals en adopció, ja que la seua finalitat serà agrupar totes les protectores d'animals, que donen el seu vistiplau, en una base de dades conjunta. Per altra permetrà crear alertes quan algú perda un animal, o quan se'n trobe algun perdut pel carrer, per tal de facilitar la cerca de mascotes desaparegudes.

Paraules claus: base de dades, aplicació web, web.

Resumen

En este trabajo mostraremos el desarrollo de una aplicación web la cual tratará, por un lado, de ayudar en la adopción de animales abandonados que se encuentran en protectoras, y por otro facilitar la búsqueda de mascotas perdidas.

Esta web, por una parte permitirá poder elegir dentro de un gran catálogo de animales en adopción, ya que su finalidad será agrupar a todas las protectoras de animales, que den su consentimiento, en una base de datos conjunta. Por otra permitirá crear alertas cuando se pierda un animal, o cuando se encuentre uno perdido por la calle, para facilitar la búsqueda de mascotas desaparecidas.

Palabras clave : base de datos, aplicación web, web.



Abstract

In this project we will show the development of a web application aiming to help in the adoption of abandoned animals living in shelters, as well as to facilitate the search for lost pets.

This web, on one hand, will allow its users to choose among a wide variety of animals up for adoption, since its primary goal is to bring together every shelter that agrees to it, into a collective data base. On the other hand, it will also make possible the creation of alarms whenever someone loses a pet, or finds a missing animal on the street, aiming to make the search for missing animals easier.

Keywords : data base, web application, web.

Taula de continguts

1.Introducció.....	9
1.1.Motivació.....	9
1.2. Objectius	10
2.Estat de l'art	11
2.1.Llenguatge de programació per a servidors.....	11
2.2.Aplicacions relacionades	12
3.Especificació de requisits	14
3.1.Introducció.....	14
3.2.Descripció general	15
3.3.Requisits específics	18
4.Anàlisi.....	20
4.1.Diagrama de classes	20
4.2.Casos d'ús	21
5.Disseny.....	25
5.1.Capa de presentació.....	25
5.2.Capa de persistència.	28
5.3. Capa lògica	30
6.Details de la implementació.....	33
6.1.Tecnologies utilitzades	33
6.2.Llibreries auxiliars	34
6.3.Plataformes de desenvolupament	34
6.3.Estructura dels fitxers.....	36
6.4.API REST.....	37
7.Proves	39
8.Conclusions.....	44
9.Treball futur a realitzar	45
9.1.Desplegament.....	45
9.2.Contactar amb les protectores.....	45

9.3.Creació de l'App mòbil complementària.....	46
10.Bibliografia	48

Taula de figures

Figura número 1.Llistat d'animals.	12
Figura 2. Fitxa d'informació d l'animal.	13
Figura 3. Formulari consulta informació.	13
Figura 4. Diagrama de classes.	20
Figura número 5. Mock up llista d'animals.	25
Figura número 6. Mock up fitxa d'animals.	26
Figura número 7. Mock up crear/mostrar alerta.	26
Figura número 8. Mock up perfil de l'usuari.	27
Figura número 9. Esquema base de dades.	30
Figura número 10 VISUAL STUDIO CODE	35
Figura número 11 GitHub desktoP	35
Figura número 16 Llista d'animals	39
Figura número 17 llista d>alertes	39
Figura número 18 fitxa del animal.	40
Figura número 19 formulari de registre.	41
Figura número 20 afegir animal.	41
Figura número 21 creació d>alertes	42
Figura número 22 Mostrar alerta.	42
Figura número 23 EDITAR PERFIL.	43
Figura número 11 mock UP MENÚ	46
Fig. núm. 12 i 13 mock up crear alerta i vincular animal.	47
Figura número 14 i 15 mock up llista ALERTES i alerta.	47

1. Introducció

Vivim en una època en què la lluita contra molts dels problemes presents a la nostra societat, com poden ser el racisme, la intolerància o masclisme, avança a passos gegantins. Aquest gran avanç ve de la mà de les tecnologies, gràcies a la gran visibilitat que aconsegueixen donar-los.

Tanmateix, no podem dir que la lluita haja acabat, ens queden molts altres problemes que encara que semblen menys importants estan molt presents en la nostra societat. Un d'aquests problemes és l'abandonament dels animals, ja siga voluntàriament o involuntària (pèrdua i no recuperació).

Per això, observant l'ajuda tan gran que ens ofereix la tecnologia per a facilitar la resolució de molts dels nostres problemes, perquè no utilitzar-la també per ajudar, per un costat a eixos animals que no es mereixen viure tota una vida dins d'una gàbia i per l'altre a eixes famílies que tant pateixen a causa de la pèrdua d'eixe membre "pelut" de la seua família?

És evident que l'abandonament de mascotes representa un gran problema en tot el món. Molta gent busca un nou membre per a la seua família i per les modes i el consumisme prefereixen comprar un animal abans d'adoptar-lo, creant així un mercat de tracta d'animals que no sempre compleix les lleis establertes. Aquest dos problemes tenen una solució conjunta i la tecnologia ens la proporcionarà

1.1. Motivació

Per això un dels objectius que busquem en aquest projecte és unir totes les protectores d'animals en un mateix lloc per a facilitar i agilitzar aquest complicat procés que pot ser el d'adopció d'un animal, i intentar evitar la compra i la cria de nous animals quan ja n'existeixen molts que busquen una llar on viure.

Altre problema que també ens agradaria abordar està relacionat amb les mascotes, i no és altre que seua pèrdua. Qui no ha perdut mai un animal i ha passat moments d'angoixa? Quan et trobes en eixa situació no saps ben bé què fer, a part de denunciar. Desgraciadament la policia moltes vegades no té ni el temps ni els recursos necessaris per a realitzar la recerca, l'única alternativa que queda és enganxar cartells pel poble amb la foto de l'animal.

Açò en la societat tan tecnològica en la qual vivim sembla "prehistòric". Per aquesta raó crear un lloc on puguem demanar ajuda d'una manera ràpida i eficient és tan necessari.

El que busquem és aprofitar el temps emprat i desenvolupar un projecte que pugui beneficiar la societat. D'aquesta manera proporcionarem una ferramenta gratuïta i útil que ajude a unir les distintes comunitats que existeixen al voltant de les protectores d'animals i així donar-los una major visibilitat, i si es possible disminuir la seua càrrega de treball.

1.2. Objectius

Al llarg d'aquest projecte busquem complir tres objectius:

Crear una ferramenta perquè tot usuari que utilitzi aquesta aplicació pugui veure d'una forma senzilla una llista de tots els animals en adopció de les protectores que en vulguen formar part.

Que l'usuari pugui veure tota l'informació rellevant sobre qualsevol animal d'una forma clara i detallada.

Que els usuaris registrats tinguin un lloc on demanar ajuda en la recerca o mitjançant alertes en cas de pèrdua d'un animal, o de trobada.

2. Estat de l'art

En aquest apartat parlarem en primer lloc, dels principals llenguatges de programació utilitzats a l'hora de desenvolupar aplicacions web i farem una ullada a les aplicacions similars que es poden trobar per Internet.

2.1. Llenguatge de programació per a servidors

Existeix una gran varietat de llenguatges de programació per al desenvolupament d'aplicacions web, entre els quals podem destacar els següents:

ASP.NET: és un entorn de programació per a aplicacions web desenvolupat i comercialitzat per Microsoft. La seua primera versió fou llançada en l'any 2002, i és la successora de la tecnologia Active Server Page (ASP). És utilitzada pels programadors i dissenyadors per a construir llocs web, aplicacions web i serveis web XML. ASP.NET està construïda sobre el Common Language Runtime, el qual permet als programadors utilitzar qualsevol llenguatge de programació admès per .NET Framework, com podrien ser C++, Visual Basic, C#, entre altres.

Java: és un llenguatge de programació orientat a objectes, el qual va ser desenvolupat per Sun Microsystems en els anys noranta, i adquirit posteriorment per Oracle. Un dels principals avantatges de Java és la seua gran portabilitat, es poden realitzar programes executables en qualsevol context o ambient. El framework més conegut i utilitzat s'anomena Spring MVC, el seu principal avantatge és que ens ajuda a crear qualsevol de les capes de l'aplicació.

Ruby: és un llenguatge de programació interpretat, reflexiu i orientat a objectes creat per Yukihiro "Matz" Matsumoto, presentat en 1995. Combina una sintaxi inspirada en Python i Perl amb característiques de programació orientades a objectes similars a Smalltalk. La seua implementació oficial és distribuïda baix una llicència de software lliure. El framework utilitzat per a la programació web s'anomena Rails, Ruby on Rails o RoR.

Perl: és un llenguatge de programació dissenyat per Larry Wall l'any 1987. Perl agafa característiques C, del llenguatge interpretat bourne shell (sh), AWK, sed, Lisp i, en menor mesura, de molts altres llenguatges. Perl segueix sent àmpliament emprat gràcies a la seua destresa a l'hora del processament de textos i per no comptar amb les limitacions que tenen altres llenguatges d'script. Existeixen molts frameworks per al desenvolupament web dins del llenguatge Perl entre els quals es poden trobar Mojolicious 2.0, Jifty o Gantry.

PHP: és un llenguatge de programació adaptat especialment al desenvolupament web. Creat el 1994 pel programador danés-canadenc Rasmus Lerdorf. PHP significava originàriament Personal Home Page (pàgina personal), però ara significa "inicialisme recursiu" PHP: Hypertext Preprocessor. Gràcies a la gran adaptació del llenguatge per al desenvolupament web existeixen diversos frameworks per realitzar projectes webs com poden ser Symfony o Laravel entre d'altres.



Python: es tracta d'un llenguatge de programació multiparadigma, és a dir, suporta parcialment l'orientació a objectes, la programació imperativa i, en menor grau, la programació funcional. És un llenguatge interpretat, dinàmic i multiplataforma. Una de les característiques que ha ajudat en la gran extensió d'aquest llenguatge és la facilitat que dona a l'hora de redactar i llegir el seu codi. El framework més conegut per al desenvolupament web és l'anomenat com a Django i és el que emprarem en el nostre projecte.

2.2. Aplicacions relacionades

No hem aconseguit trobar cap aplicació que desenvolupi els punts que volem desenvolupar en aquest projecte. No obstant això hem seleccionat una pàgina web d'una protectora d'animals per veure com se sol treballar en aquest camp, ja que totes solen seguir el mateix patró. La pàgina web triada és la de la protectora anomenada SVAP situada en San Antonio de Benagéber.

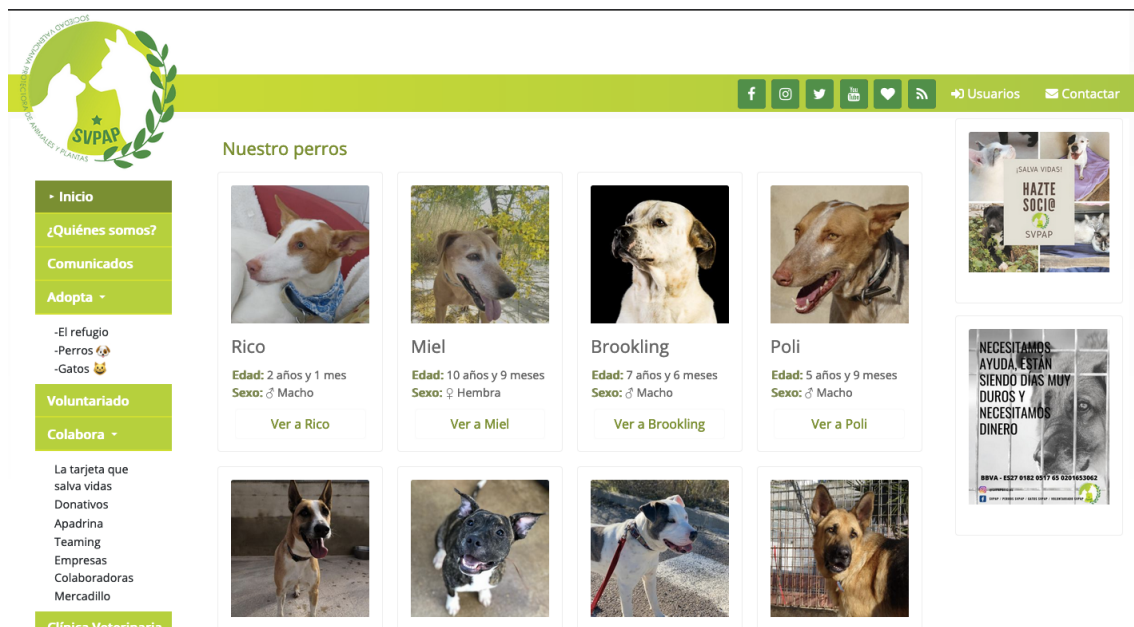


FIGURA NÚMERO 1.LLISTAT D'ANIMALS.

En la captura anterior (figura núm.1) podem observar com mostren els distints animals que es troben esperant per a ser adoptats. Quan cliques en qualsevol dels animals la pàgina et dirigeix a la pantalla que podem observar en la captura de pantalla número 2.

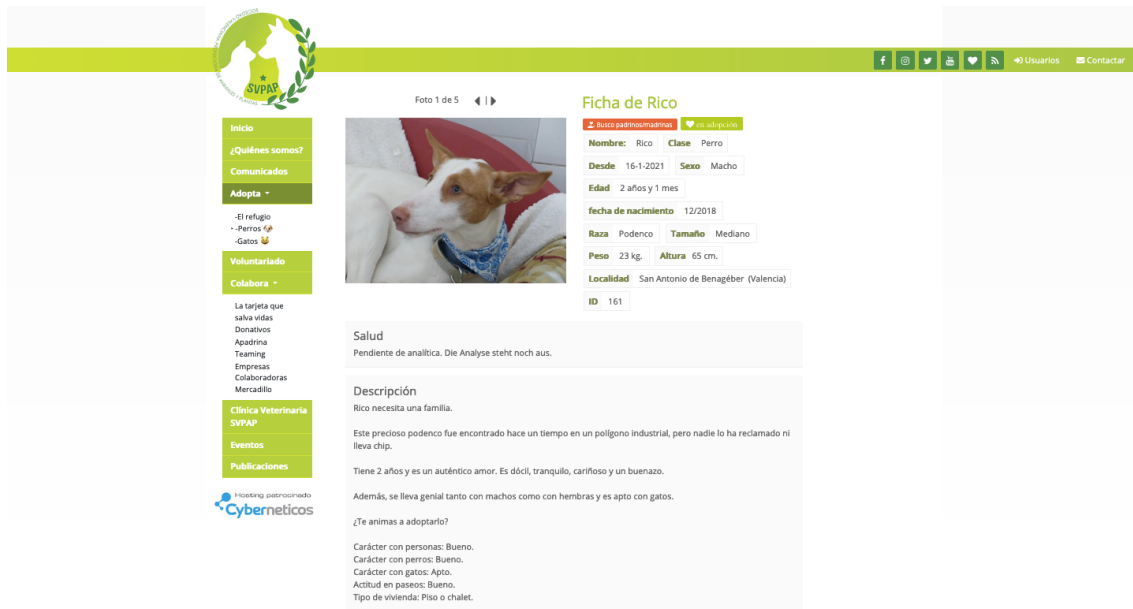


FIGURA 2. FITXA D'INFORMACIÓ D L'ANIMAL.

En aquesta captura (figura núm. 2) es poden observar diverses imatges del gos amb tota la informació útil necessària per a les persones que estiguen interessades en aquest animal. Entre aquesta informació podem trobar el nom, pes, grandària, el seu comportament, etc. Si fem scroll, i baixem trobarem la captura de pantalla número 3 on podem observar un formulari, el qual ens dona distintes opcions, podrem rebre més informació de l'animal, apadrinar-lo o adoptar-lo segons les nostres necessitats.

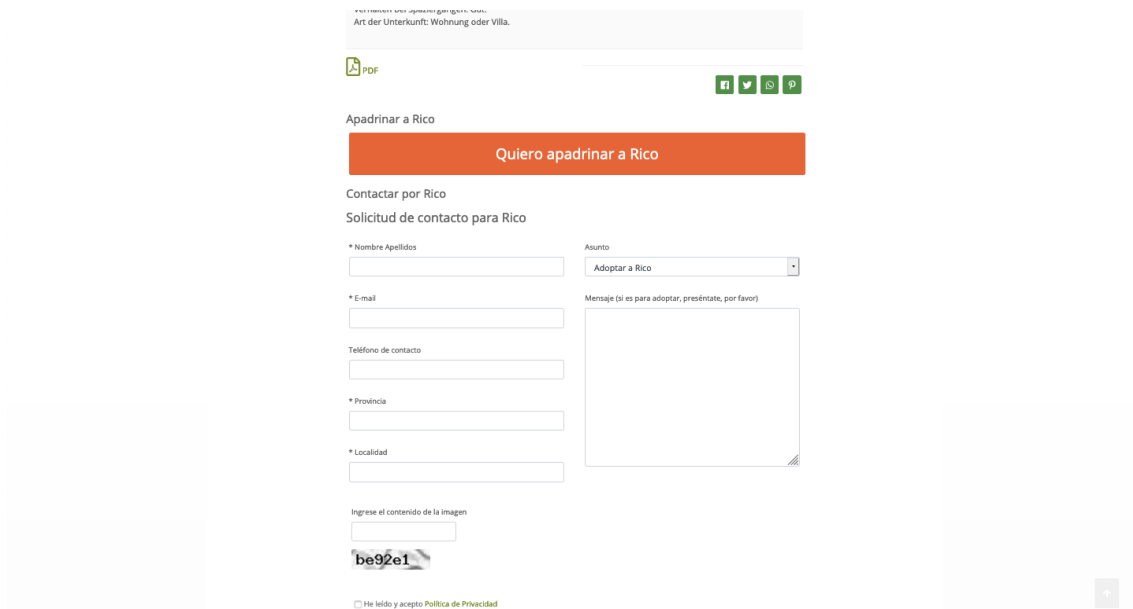


FIGURA 3. FORMULARI CONSULTA INFORMACIÓ.

Si naveguem per la pàgina també podem trobar informació de com podem ajudar, ja siga mitjançant donacions monetàries o de materials, com poden ser medicaments, aliments o de construcció. Una altra forma d'ajudar és mitjançant un voluntariat, encara que l'actual situació de la COVID-19 ha paralytitzat en gran mesura aquest tipus d'ajuda.

3. Especificació de requisits

En aquest punt tractarem d'exposar el projecte a realitzar d'una forma més clara i detallada. Començarem descrivint les funcionalitats que contindrà l'aplicació web i els distints usuaris que en formaran part. També farem una llista de definicions de certs temes més específics la qual ens ajudarà a entendre millor el projecte.

3.1. Introducció

3.1.1. Propòsits i abast

Com hem dit més amunt, el propòsit principal d'aquest projecte no és altre que el de facilitar la tasca de les protectores d'animals i intentar alleugerir la seua càrrega de treball, per una part donant-los una major visibilitat i per l'altra intentant aconseguir que tota persona que perda la seua mascota la pugui trobar amb facilitat abans que arribe a qualsevol protectora.

La pàgina web oferirà a qualsevol persona que la visite l'opció de poder buscar animals en estat d'adopció i obtenir-ne una xicoteta descripció, a més farà de nexa d'unió entre aquests usuaris i les protectores que vulguen formar part de la web. Al mateix temps donarà opció, als usuaris registrats, de crear alertes quan algun dels seus animals s'haja perdut o si se'n troben algun pel carrer.

També oferirà una API Rest per al consum d'una aplicació mòbil que pensem desenvolupar en un futur per a facilitar la creació de les alertes i la seua localització. Al llarg del projecte mostrarem mock ups i explicarem quin procediment volem seguir per al seu desenvolupament encara que no ho farem d'una manera molt extensa ja que no és l'objectiu d'aquesta memòria.

Resumint, els objectius que volem aconseguir són els següents:

- Visualització d'una llista amb tots els animals.

- Una xicoteta fitxa dels animals i un enllaç a la pàgina de la protectora.

- Creació d'alertes, tant d'animals perduts, com trobats.

- API Rest per a la futura app mòbil.

3.1.2. Definicions i acrònims

En aquest apartat farem una xicoteta explicació de diversos termes que han aparegut o apareixeran al llarg d'aquesta memòria per a fer més fàcil la seua comprensió.

Definicions

- Geolocalització: és la capacitat per obtenir la ubicació geogràfica real d'un objecte, com un radar, un telèfon mòbil o un ordinador connectat a Internet.
- Framework: la traducció al valencià seria “entorn de treball o marc de treball”, i és el conjunt estandarditzat de conceptes, pràctiques i criteris per a enfocar un tipus de problemàtica particular que serveix com a referència per a enfrontar i resoldre nous problemes d'una índole similar, facilitant així la tasca dels programadors.
- Script: designa un progrma relativament simple el qual automatitza l'execució de tasques.

Acrònims

- BD: data base base de dades.
- API Rest: Representational State Transfer transferència d'estat representacional (rest).
- PaaS: Plataform as a service Plataforma com a servei.

3.2. Descripció general

En aquest apartat farem una descripció abstracta del projecte i les seues funcionalitats.

3.2.1. Perspectiva del producte

L'objectiu principal d'aquest projecte és desenvolupar una aplicació web la qual facilita l'adopció o l'apadrinament d'animals que es troben en protectores.

Altre objectiu molt important dins del projecte és d'agilitzar les recerques de les mascotes i aprofitar tots els recursos que tenim al nostre abast per a ajudar en aquesta tasca.

3.2.2.Funcionalitat del producte

Les funcions que formaran part de l'aplicació web són les següents:

Gestió dels usuaris: podran registrar-se en l'aplicació com a usuaris base, també donarem a les protectores o a determinats voluntaris l'opció de crear-se un perfil especial, d'aquesta manera tindran més funcionalitats.

Gestió de llista d'animals en adopció: està reservada als perfils de les protectores perquè actualitzen la llista d'animals en adopció.

Gestió d'alertes d'animals: els usuaris registrats podran crear alertes quan perden o troben un animal.

Gestió d'animals propis: els usuaris registrats podran vincular al seu perfil les seues mascotes per a facilitar la creació d'alertes en cas que s'extravien.

3.2.3.Característiques del producte

En l'aplicació podem observar tres tipus d'usuaris clarament diferenciats: administradors, protectores i usuaris base, els quals optaran a determinades funcionalitats segons les seues característiques.

Administradors: són aquells que vetlaran pel bon ús de l'aplicació i actuaran si veuen que qualsevol usuari en fa un ús inapropiat, com puga ser la creació d'alertes falses o la utilització de l'aplicació per a lucrar-se.

Protectores: són aquells voluntaris de les protectores que accepten entrar dins de l'aplicació.

Usuaris base: són tots aquells que vulguen beneficiar-se de les funcionalitats que ofereix aquest projecte, com és l'adopció d'un animal, o ser ajudat o ajudar a l'hora de buscar-ne algun perdut.

Gestió dels usuaris:

Procés	Usuaris	Protectores	Administrador
Crear compte	X	X	X
Iniciar sessió	X	X	X

Gestió d'animals propis:

Procés	Usuaris	Protectores	Administrador
Vincular mascota	X		

Gestió de llista d'animals:

Procés	Usuaris	Protectores	Administrador
Crear animal		X	X
Eliminar animal		X	X
Vincular animal	X	X	X

Gestió d'alertes d'animals:

Procés	Usuaris	Protectores	Administrador
Crear alerta	X	X	X
Eliminar alerta (pròpia)	X	X	X
Visitar alerta	X	X	X

3.2.4. Restriccions del sistema

Aquesta aplicació web ha sigut ideada per a ser usada en navegadors web d'ordinador, no l'hem feta *responsive* per a dispositius mòbils per dues raons: la primera és que la llista dels animals no tindria molt de sentit en un pantalla xicoteta ja que no es podrien diferenciar bé ni els animals ni les seues dades; la segona, i més important, és que com bé hem dit anteriorment, tenim pensat dissenyar una aplicació mòbil per a facilitar la creació de les alertes, utilitzant la càmera i el gps del mòbil, per reduir el temps des que es perd l'animal fins que es crea l'alerta.

3.2.5. Dependències del sistema

El sistema haurà de ser desplegat en un servidor en el qual puguen ser instal·lats tant python i el seu framework django, com docker, en el qual crearem el servidor per a la missatgeria instantània.

També comptarem amb una dependència externa ja que hem utilitzat l'API de Google Maps per a facilitar la creació d'alertes i millorar-ne la visualització. Confiam en Google per la seua disponibilitat en tot moment.

3.3.Requisits específics

En el següent apartat desenvoluparem els aspectes imprescindibles que haurà de tindre la nostra aplicació web per a considerar-la funcional de manera plena.

3.3.1.Requisits específics.

En el següent apartat desenvoluparem els aspectes imprescindibles que haurà de tindre la nostra aplicació web per a considerar-la funcional de manera plena.

Requisits d'interface externs

Aspectes visuals que haurà d'acomplir el sistema:

- **IR-RO1:** Idioma. L'idioma principal serà el valencià, amb perspectiva de futur d'incloure el castellà.
- **IR-RO2:** Compatibilitat amb els principals navegadors del mercat. (Safari, Microsoft Edge, Mozilla Firefox, Google Chrome).

Requisits funcionals

Com el projecte té dos objectius clarament diferenciats, l'adopció o apadrinament dels animals i l'ajuda a l'hora de realitzar la recerca d'una mascota perduda, hem considerat d'ajuda separar els requisits funcionals en dos grups segons el seu objectiu principal:

Requisits comuns a les dos funcionalitats

- **CR-RO1.** Registrar-se o fer login. El sistema cal que deixe registrar a qualsevol persona que estiga disposada a donar les seues dades i acceptar la política de privacitat de l'aplicació. Referint-se als comptes de les protectores, seran creats pels administradors per a major seguretat.
- **CR-RO2.** Editar perfil. Els usuaris registrats tindran l'opció de modificar o afegir certes dades com poden ser una foto de perfil, nom i cognoms (nom de la protectora), un telèfon de contacte o una localització de referència per a rebre alertes.

Adopció/apadrinament d'animals

- **AR-Ro1.** Divisió animals. El sistema ha de donar tres opcions a l'hora de mostrar animals. La primera, tots junts; la segona, una llista sols de gossos i la tercera només de gats, ja que aquests dos animals són els més presents en les protectores d'animals.
- **AR-Ro2.** Fitxa animals. Al fer clic sobre un animal de la llista l'aplicació ha de mostrar una fitxa amb la informació rellevant.
- **AR-Ro3.** Afegir a preferits. L'usuari ha d'estar registrat. Dins de la fitxa de l'animal apareixerà un botó el qual, al ser seleccionat, guardarà l'animal a la llista de preferits de l'usuari, la qual apareixerà en el perfil de l'usuari per a poder fer un seguiment de l'animal.
- **AR-Ro4.** Afegir/eliminar animal. El sistema ha de permetre a les protectores afegir un nou animal, o eliminar-lo quan ho consideren oportú.

Recerca d'animals

- **RR-Ro1.** Crear alerta. El sistema ha de permetre crear una alerta als usuaris registrats, aquesta alerta pot ser de dos tipus, quan es troba un animal o quan es perd.
- **RR-Ro2.** Mostrar llista d'alertes. El sistema mostrarà en tot moment en una barra lateral una llista de les últimes alertes que s'han produït, ja siguin d'animals perduts o trobats. També donarà opció a visualitzar una llista diferent per a cada tipus d'alerta on apareixeran totes, no sols les més recents.
- **RR-Ro3.** Vincular mascota. Un usuari registrat podrà vincular les seues mascotes en l'aplicació per a agilitzar la creació d'una alerta en cas de pèrdua.
- **RR-Ro4.** Alerta. El sistema ha de mostrar tota la informació d'una alerta, com per exemple l'animal vinculat a l'alerta, l'última localització on va ser vist, i un xat en viu que podrà ser utilitzat per controlar la cerca per part dels diferents participants.

Serveis API

- **AP-Ro1.** Oferir llista d'alertes. Cal retornar la llista de les alertes guardades en el sistema.

Requisits del rendiment

En un principi no existeix cap requisit de rendiment, encara que l'aplicació haurà de respondre a les peticions dels usuaris en menys d'un segon perquè l'usuari no siga conscient de la demora i acomplir així els estàndards.

4. Anàlisi

En aquest apartat, realitzarem una anàlisi més exhaustiva per a poder fer-nos una idea més clara i precisa del producte final.

4.1. Diagrama de classes

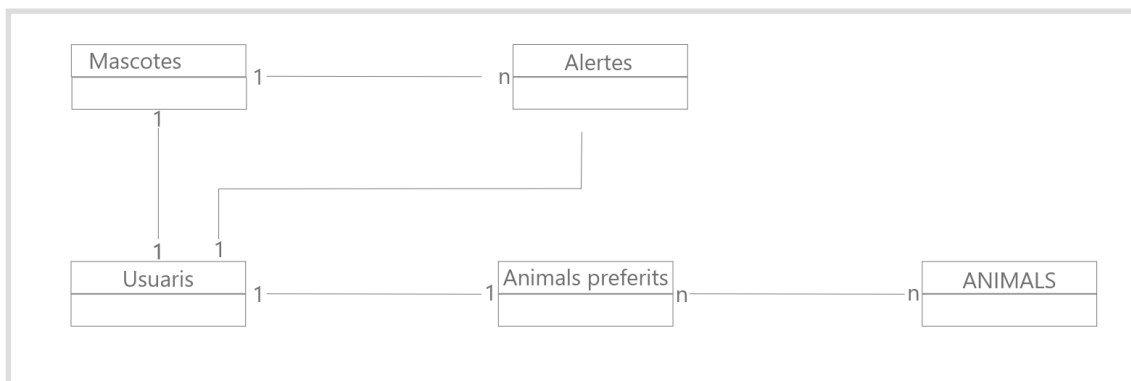


FIGURA 4. DIAGRAMA DE CLASSES.

En el diagrama de classes que podem observar en la figura anterior podem veure que els usuaris poden vincular tantes mascotes com tinguen, però aquestes mascotes sols poden ser vinculades a un sol usuari. També podem observar que cada usuari conté una llista de preferits la qual pot contenir tants animals com l'usuari desitge.

Els animals en adopció poden estar en moltes llistes de preferits a la mateixa vegada.

Per altra banda, una alerta sols pot estar relacionada amb una mascota, és a dir, que per a mascotes distintes l'usuari haurà de crear alertes distintes. Una mateixa mascota pot estar relacionada amb moltes alertes, d'igual manera un usuari pot estar relacionat amb moltes alertes però una alerta sols pot estar relacionada amb un usuari.

4.2.Casos d'ús

Cas d'ús 1: Mostrar llista d'animals	
Actors	Tots els usuaris.
Objectius	Obtenir una llista dels animals en adopció.
Resum	Es farà una llista dels animals en adopció diferenciant pel color del marc de la foto entre gats i gossos, també es podrà seleccionar perquè sols mostre gossos o gats.
Precondició	L'usuari haurà d'haver entrat en la pàgina web.
Postcondició	El sistema retornarà una llista dels distints animals en adopció.
Requisits associats	AR-R01

Cas d'ús 2: Registrar-se com a usuari	
Actors	Usuaris.
Objectius	Obtenir un compte d'usuari per a poder crear alertes de gossos perduts i fer-ne el seguiment.
Resum	Per a registrar-se els usuaris hauran d'elegir un nom d'usuari, una contrasenya, la qual caldrà que confirmen, i adjuntar una direcció de correu vàlida per a facilitar la comunicació amb l'aplicació.
Precondició	No pot existir un compte amb el mateix nom d'usuari o email associats.
Postcondició	Es guardarà l'usuari amb les seues credencials en la base de dades. L'usuari serà redirigit a la llista dels animals.
Requisits associats	CR-R01

Cas d'ús 3: Afegir informació al perfil de l'usuari	
Actors	Usuaris.
Objectius	Completar la informació dels usuaris per agilitzar la creació de les alertes.
Resum	L'usuari podrà afegir informació: dades personals - el seu nom complet i telèfon- localització... també podrà vincular les seues mascotes .
Precondició	No pot existir un compte amb el mateix nom d'usuari o email associats.
Postcondició	Es guardaran les noves dades de l'usuari a la base de dades.
Requisits associats	CR-R02,RR-R03.

Cas d'ús 4: Registrar-se com a refugi	
Actors	Treballadors d'un refugi
Objectius	Obtenir un compte de protectora per poder crear/eliminar animals.
Resum	Per a registrar-se els treballadors hauran d'insertar el nom de la institució, una contrasenya, que caldrà que confirmen, i adjuntar una direcció de correu vàlida per a facilitar la comunicació amb l'aplicació.
Precondició	El treballador haurà d'haver entrat en la pàgina web, i demanar la validació del compte a un administrador.
Postcondició	Es guardarà la protectora amb les seues credencials en la base de dades. L'usuari serà redirigit a la llista dels animals.
Requisits associats	CR-R01.

Cas d'ús 5: Afegir un animal	
Actors	Treballadors d'un refugi.
Objectius	Afegir un animal a la base de dades de l'aplicació.
Resum	Mitjançant un formulari els treballadors de les protectores podran afegir un animal perquè es mostre a la llista d'animals.
Precondició	El treballador haurà d'estar registrat en un compte de protectora.
Postcondició	L'animal apareixerà en la llista dels animals per tal que tots els usuaris puguin veure la seua informació.
Requisits associats	AR-R04

Cas d'ús 6: Crear alerta	
Actors	Tots els usuaris registrats.
Objectius	Crear una alerta perquè es mostre a l'aplicació.
Resum	Mitjançant un formulari els usuaris podran crear una alerta per a indicar que han extraviat una mascota o que han trobat un animal solt pel carrer.
Precondició	El usuaris hauran d'estar registrats.
Postcondició	L'alerta es mostrarà dins de l'aplicació en els llocs pertinents.
Requisits associats	RR-R01

Cas d'ús 7: Fitxa de l'animal	
Actors	Tots els usuaris.
Objectius	Mostrar les dades dels animals.
Resum	Es mostraran totes les dades de l'animal que hem considerat rellevants, com poden ser el pes, l'altura, raça i comportament.
Precondició	Fer clic en qualsevol animal de la llista.
Postcondició	
Requisits associats	AR.R02

Cas d'ús 8: Guardar a preferits	
Actors	Usuaris registrats.
Objectius	Poder guardar un animal que ens agrade.
Resum	En entrar dins de la fitxa d'un animal, un usuari registrat podrà guardar aquest animal en la seua llista d'animals preferits i així podrà accedir a la seua informació d'una manera més ràpida, ja que aquesta llista es trobarà en el seu perfil.
Precondició	Obrir la fitxa de qualsevol animal i fer clic al botó de preferits.
Postcondició	Si hem apretat el botó de preferits l'animal es podrà veure en la llista dins del perfil de l'usuari.
Requisits associats	AR.R03

Cas d'ús 9: Mostrar llistat d'alertes	
Actors	Tots els usuaris.
Objectius	Obtenir una llista de les alertes.
Resum	Es farà una llista de les alertes, diferenciant entre alertes d'animals perduts i animals trobats
Precondició	L'usuari haurà d'haver entrat en la pàgina web.
Postcondició	Si l'usuari està registrat podrà observar tota la informació de l'alerta fent click sobre ella.
Requisits associats	RR-R02

Cas d'ús 10: Mostrar alerta	
Actors	Usuaris registrats.
Objectius	Obtenir informació sobre les alertes.
Resum	Es mostrarà tota la informació de l'alerta i si aquesta es sobre un animal perdut, es podrà utilitzar un xat en viu per a tot aquell que vulga aportar informació útil o ajuda.
Precondició	L'usuari haurà d'estar registrat i haurà d'haver fet clic sobre l'alerta de la qual vol visualitzar la informació.
Postcondició	
Requisits associats	RR-RO2

5. Disseny

En aquest apartat farem una explicació més detallada del projecte realitzat. Començarem fent una xicoteta introducció i després passarem a parlar de cada una de les tres capes de les que consta el projecte, la capa de presentació, la de persistència i la de lògica.

5.1. Capa de presentació

Per a la capa de presentació vam fer uns mock-ups mitjançant la ferramenta ADOBE XD, la qual ens permet crear les diferents pantalles d'una forma ràpida i intuïtiva. Vam optar per aquesta ferramenta ja que no volíem utilitzar cap tipus de plantilla, per a poder personalitzar l'estil de la pàgina tot el que desitgem. A l'hora de fer els mock-ups no vam decidir quins colors emprariem en la nostra aplicació ja que teníem més d'una opció en ment.

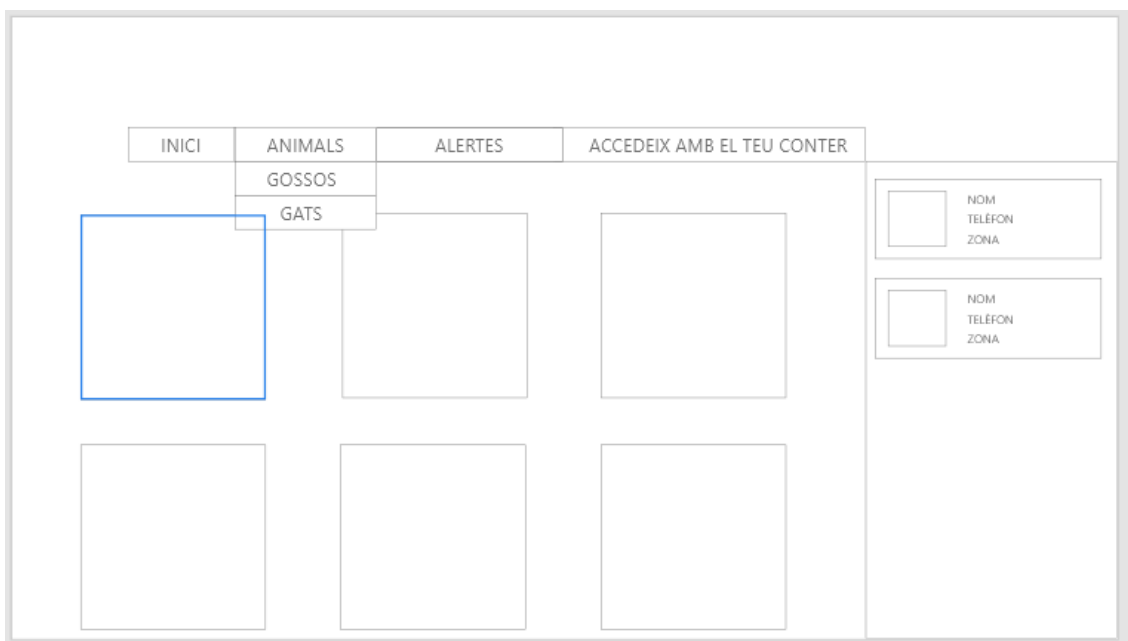


FIGURA NÚMERO 5. MOCK UP LLISTA D'ANIMALS.

En la figura número 5 podem veure per una part el menú amb les opcions desplegable que apareixeran quan l'usuari passe el cursor per damunt de la paraula ANIMALS, també podem observar la distribució que tindrà la llista dels animals, on els quadrats que apareixen sense res contindran les seues imatges, totes les llistes de l'aplicació seguiran el mateix estil, estaran constituïdes per quadrats amb les imatges pertinents. Per últim a la dreta de la imatge podem observar com apareixerà la llista d'alertes, aquesta serà comuna per a totes les pantalles.

Quan l'usuari clique en qualsevol de les imatges dels animals apareixerà la seua fitxa que podem veure en la figura número 6.

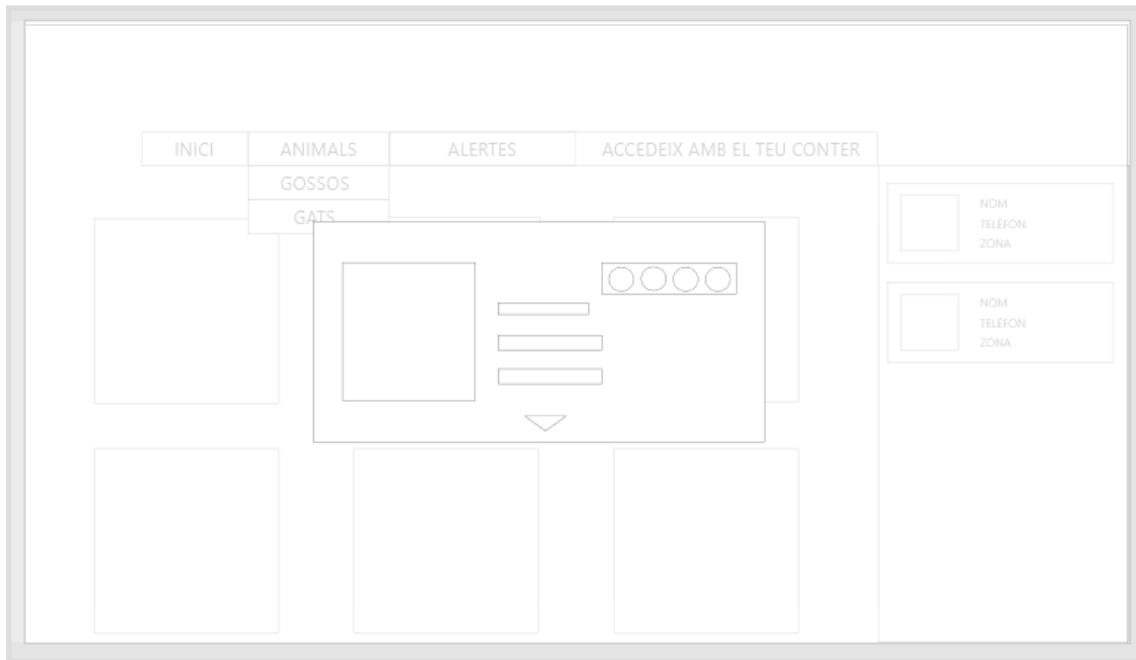


FIGURA NÚMERO 6. MOCK UP FITXA D'ANIMALS.

En aquesta fitxa podem observar quatre parts clarament diferenciades. El quadrat que observem a l'esquerra és la foto de l'animal, els tres rectangles que es troben al costat de la foto són el lloc on es trobarà la informació important del gos. El requadre de d'alt a la dreta, podem veure que conté quatre cercles, aquests cercles contindran icones amb colors diferents perquè l'usuari pugui diferenciar-los ràpidament. I per últim el triangle que s'observa a la part de baix de la fitxa, serà un botó, que en ser premut farà créixer la fitxa per a mostrar la informació addicional de l'animal.

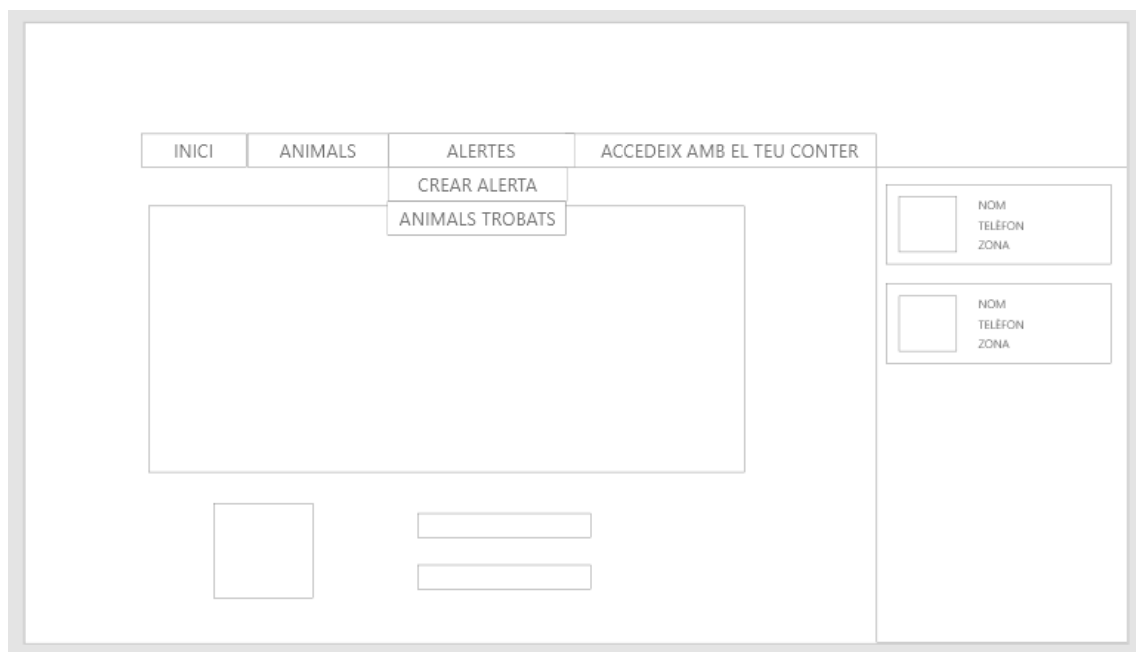


FIGURA NÚMERO 7. MOCK UP CREAR/MOSTRAR ALERTA.

En la figura número 7, podem observar l'estructura que seguiran tant la pantalla per a la creació d'alertes, com la pantalla de mostrar alertes. Ambdues tindran un disseny paregut, ja que comptaran amb un mapa (rectangle superior), i en la part de baix apareixeran un formulari, en el cas de la creació de l'alerta, o les dades importants, en el cas de mostrar alerta.

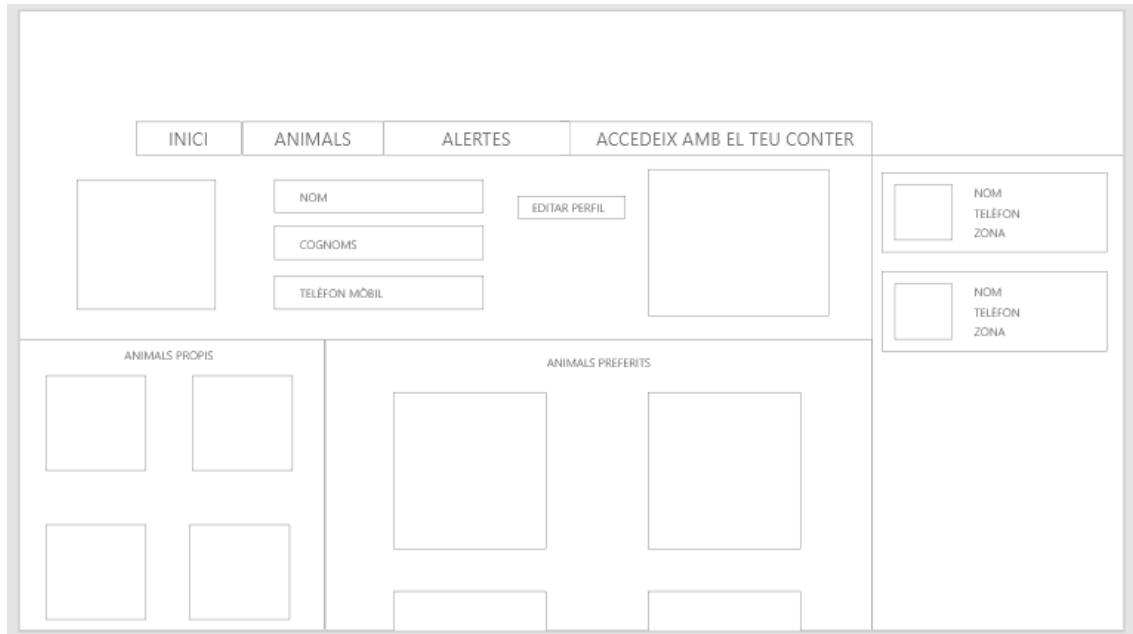


FIGURA NÚMERO 8. MOCK UP PERFIL DE L'USUARI.

Per finalitzar amb la capa de presentació, la figura número 8 mostra com serà la pantalla de perfil de l'usuari, on es podran editar els camps que apareixen en la part superior de la pantalla. Les dos llistes que es poden veure, tant la d'animals preferits com animals propis podran contindre tants animals com l'usuari desitge i es podrà accedir a tots els animals mitjançant un scroll infinit. El rectangle que es troba en la part superior esquerra contindrà la foto de l'usuari.

No cal dir que les captures de pantalles explicades amb anterioritat són mockups, és a dir, en el desenvolupament del projecte s'han fet tots els canvis estètics que hem considerat rellevants per a la bona presentació i funcionament de l'aplicació.

5.2. Capa de persistència.

En la descripció dividirem els camps en dos grups: obligatoris, sense aquesta informació la BD no podrà fer la creació de l'objecte, i optatius.

Usuaris

Per a l'identificació dels usuaris guardarem en la BD les següents dades:

Obligatoris:

- ID: Identificador únic.
- Nom: nom d'usuari.
- Email: correu electrònic de l'usuari.
- Contrasenya: contrasenya xifrada de l'usuari.
- Protectora: serà un boolean que en cas de ser perfil de protectora el seu valor serà positiu, per defecte serà negatiu.

Opcionals:

- Foto: la foto de perfil de l'usuari.
- Telèfon: número de telèfon de l'usuari
- Latitud: la latitud de la localització elegida per l'usuari.
- Longitud: La longitud de la localització elegida per l'usuari.

Animals

Per a l'identificació dels Animals:

Obligatoris:

- ID: identificador únic.
- Nom: nom de l'animal.
- Foto: foto de l'animal.
- Raça: raça de l'animal.
- Edat: edat de l'animal.
- Pes: pes de l'animal.
- Altura: altura de l'animal.
- Grandària: grandària de l'animal.
- Tipus: espècie de l'animal, hi haurà dues opcions, gos o gat.
- Protectora: serà un boolean, el valor del qual serà positiu si l'animal és d'una protectora, o negatiu si és una mascota.

Optatius:

- URL protectora: en el cas que l'animal siga d'una protectora, es guardarà l'enllaç a la seua pàgina web, en cas que existisca.
- PPP: un boolean, que serà positiu si l'animal està considerat de raça perillosa.
- Actitud cap a les persones: l'actitud que té l'animal enfront a les persones.
- Actitud davant dels gossos: l'actitud que té l'animal enfront als gossos.
- Actitud davant dels gats: l'actitud que té l'animal enfront als gats.
- Pors: les possibles pors de l'animal.
- Malalties: les possibles malalties de l'animal.

Els usuaris podran tindre dos llistes diferents, la de les mascotes o animals propis, i la d'animals preferits, les dos tenen una estructura similar, i tots els seus camps són obligatoris:

Llista mascotes

- ID: identificador de la llista.
- ID usuari: identificador de l'usuari.
- ID animals: llista dels identificadors dels animals vinculats per l'usuari.

Animals preferits

- ID: identificador de la llista.
- ID usuari: identificador de l'usuari.
- Animals: llista dels animals seleccionats com a preferits per l'usuari.

Alertes

Per a identificar les alertes, tots els seus camps són obligatoris:

- ID: identificador de l'alerta.
- Animal: ID de l'animal del qual es fa l'alerta.
- Creador: ID de l'usuari que a creat l'alerta.
- Estat: estat de l'alerta, hi haurà dues opcions possibles, perdut o trobat.
- Telèfon: número de telèfon de contacte del creador de l'alerta.
- Hora: hora de creació de l'alerta.
- Latitud: latitud de la localització de l'alerta.
- Longitud: longitud de la localització de l'alerta.

En la figura número 9 podem veure d'una manera gràfica els punts explicats amb anterioritat:

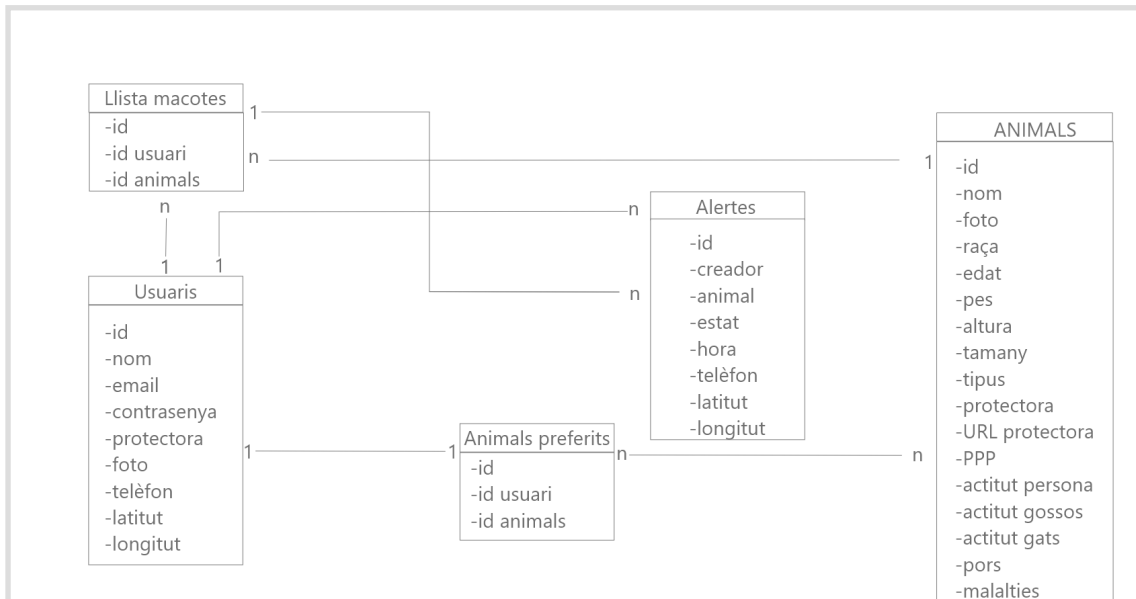


FIGURA NÚMERO 9. ESQUEMA BASE DE DADES.

5.3. Capa lògica

En aquest punt explicarem les funcionalitats del projecte les quals vindran definides per els distints casos d'usos.

CU1 - Mostrar llista d'animals. L'usuari accedirà a la pàgina d'animals mitjançant el menú. En passar el cursor sobre la paraula "Animal", apareixeran els dos tipus d'animals disponibles en la BD.

Si l'usuari fa clic sobre la paraula "Animal", el sistema farà una recerca en la BD, i mostrarà tots els animals que pertanyen a una protectora, si fa clic en "Gossos" el sistema seleccionarà sols els animals que siguin gossos i en fer clic en "Gats" mostrarà sols els gats.

Quan el sistema mostre les imatges dels animals, si l'usuari clica sobre una foto, el sistema mostrarà la fitxa de l'animal.

CU2 i CU4 - Registrar-se com a usuari. El formulari per a registrar-se conté quatre camps: nom d'usuari, email de contacte, contrasenya i confirmació de la contrasenya. Quan l'usuari envie el formulari, el sistema comprovarà, mitjançant una funció de javascript, que cap camp es trobe buit, que la contrasenya i la seua confirmació coincidisquen i que la direcció de email siga vàlida. Si tot està correcte, el formulari s'enviarà al servidor el qual farà l'última comprovació, que no existeix un usuari amb el mateix nom.

Si no hi ha cap error l'usuari serà registrat en la BD.

Si es troba qualsevol error el sistema donarà una xicoteta explicació a l'usuari perquè corregisca el possible error.

Per a la creació d'un compte d'usuari de protectora, després de la creació d'un compte regular, el treballador haurà de contactar amb un administrador mitjançant el correu especificat per al canvi de rol.

CU3 - Afegir informació en el perfil de l'usuari. Una vegada l'usuari es registre, li apareixerà una finestra on podrà seleccionar l'opció d'afegir informació al perfil. Si per qualsevol causa l'usuari no vol afegir la informació en aquest moment sempre ho podrà fer clicant en l'opció de perfil del menú.

Una vegada estiga en la pàgina d'edició de perfil, l'usuari podrà omplir un formulari, el qual constarà de cinc camps. Els dos primers són el nom i cognoms. El tercer és un número de telèfon de contacte. Els dos últims seran transparents a l'usuari ja que seran coordenades geogràfiques (latitud i longitud). Per a seleccionar-les l'usuari podrà col·locar una marca dins d'un mapa perquè el sistema les calcule automàticament. Aquestes coordenades s'utilitzaran per a seleccionar quines alertes seran mostrades a l'usuari.

No es farà cap comprovació ja que no es necessari omplir cap dels camps, l'usuari pot completar aquells camps que considere importants.

L'usuari també podrà vincular les seues mascotes emplenant un formulari el qual consta dels següents camps: nom, edat, raça, foto, pes, altura, tipus d'animal, grandària i si està considerat animal perillós. El sistema avisarà l'usuari en cas de faltar qualsevol dels camps, de la mateixa manera que en el formulari de registre.

CU5 - Afegir un animal. Aquesta funcionalitat està reservada per als usuaris que siguin protectores. Una vegada l'usuari tinga un compte amb els permisos necessaris, en l'opció del menú "Animals", en passar el cursor per damunt li apareixerà una nova opció anomenada "Afegir animal". En seleccionar-la, apareixerà un formulari el qual comptarà amb disset camps dividits en tres subgrups. El primer subgrup és indispensable per a la creació de l'animal, els altres dos són complementaris.

El primer grup constarà d'informació essencial d'un animal com pot ser, el nom, l'edat, la raça, una foto, el tipus (se és gos o gat), la grandària, el pes, l'altura i si està considerat raça perillosa (ppp). El segon grup serà per tenir informació sobre el comportament de l'animal, constarà de tres camps, el comportament amb les persones, amb els gossos i amb els gats. Per finalitzar, l'últim subgrup contindrà informació extra com per exemple possibles pors o malalties que patisca l'animal.

CU 6 - Afegir alerta. Una vegada l'usuari estiga registrat tindrà l'opció de crear una alerta. Per fer-ho haurà de fer clic en l'opció del menú de "crear alerta". Si l'usuari ja té vinculat l'animal sobre el qual vol crear l'alerta, sols caldrà que el seleccione; si no el té vinculat, haurà d'omplir el formulari "vincular animal" que hem explicat en el tercer cas

d'ús. A més de seleccionar l'animal, l'usuari tindrà l'opció de moure una bandera en un mapa (mateix funcionament que google maps) per marcar l'últim lloc on s'ha vist l'animal si aquest s'ha perdut, o el lloc on s'ha trobat. Els dos últims camps són, primer un número de telèfon per a poder contactar amb el creador de l'alerta, i segon l'estat de l'alerta, és a dir, trobat o perdut.

CU 7 - Fitxa del animal. Quan l'usuari de la web clique sobre la imatge d'un animal, apareixerà una finestra modal en la qual apareixeran totes les dades guardades en la BD sobre l'animal seleccionat, aquestes dades seran aquelles que introduí el treballador de la protectora entre les quals podem diferenciar tres grups. Al primer grup trobarem el nom, l'edat, l'altura (cm) i si està considerat de raça perillosa. El segon grup constarà de quatre icones amb canvi de fons per fer l'escala, la informació que donaran aquestes icones són el comportament, tant amb humans com amb gats i gossos, i la grandària de l'animal. El tercer grup serà un desplegable el qual apareixerà baix de la fitxa i mostrarà les pors i les malalties de l'animal. Segons el tipus de l'animal la fitxa serà d'un color o de l'altre.

CU 8 - Guardar preferits. Quan un usuari registrat, es troba en la fitxa d'un animal, tindrà l'opció de guardar-lo a la seua llista d'animals preferits fent clic en l'estrela que apareix al costat de la foto. Es podrà accedir a aquesta de manera directa des de la pantalla de perfil de l'usuari, perquè així l'usuari pugui comparar els animals que més li agraden d'una forma ràpida.

CU 9 - Llista d'alertes. En tot moment, independentment de la pantalla on es trobe l'usuari, a la dreta de la pantalla apareixerà una llista d'alertes, el tipus de les quals podrem distingir segons el color: roig significaria animal perdut i verd animal trobat. Cada alerta conté una foto de l'animal, a més d'informació útil, el nom de l'animal perdut (indiferent si és un avís d'animal trobat), la zona on s'ha localitzat o va ser vist per última vegada i el telèfon de contacte. Si l'usuari està registrat en fer clic en qualsevol de les alertes el sistema el redirigirà a una pantalla amb la informació completa de l'alerta.

L'usuari registrat també podrà accedir a una llista d'alertes específica per a cada tipus d'alerta mitjançant el menú, en passar el cursor sobre la paraula "Alertes" apareixeran les dues opcions "Alertes perduts", "Alertes trobats".

CU 10.- Mostrar alerta. Com ja hem explicat amb anterioritat, quan fem clic sobre una alerta el sistema ens redirigirà a una pantalla on rebrem més informació sobre l'alerta. En aquesta pantalla podrem observar en un mapa la localització de la l'alerta, una xicoteta descripció de l'animal, un telèfon de contacte i un xat en viu per a poder facilitar la possible tasca, ja siga en la recerca de l'animal o per a ficar en contacte el propietari de la mascota i la persona que l'ha trobada.

6. Detalls de la implementació

Al llarg d'aquest apartat ens centrarem en parlar de la part tècnica del projecte, és a dir, parlarem de la tecnologia utilitzada en la seua implementació, quines llibreries auxiliars hem utilitzat per a facilitar aquesta implementació. També parlarem de quines eines hem emprat per a la creació del codi i acabarem mostrant i explicant el per què de l'organització del directoris del projecte.

6.1. Tecnologies utilitzades

Com ja hem explicat en el tercer capítol existeix una gran varietat de llenguatges de programació per al desenvolupament d'aplicacions web, les quals s'adapten a les diverses necessitats que pot tindre cada projecte. En el nostre cas hem emprat python.

Python i Django

Python és un llenguatge interpretat, dinàmic i multiplataforma que se centra molt en l'estructura del codi, amb una comunitat molt àmplia i activa gràcies a la qual es poden trobar una gran quantitat de llibres i tutorials, els quals poden facilitar molt la tasca al programador. Per aquesta raó i per l'existència del framework Django hem optat per aquest llenguatge de programació.

Django és un marc de treball d'alt nivell que facilita en gran mesura el disseny de la nostra aplicació web, gràcies al qual el programador sols haurà de centrar-se en la programació a alt nivell i no caldrà que utilitze el seu temps en tasques com podria ser la connexió amb una base de dades ja que django se n'encarrega automàticament. Tres dels seus avantatges més grans són, la rapidesa amb la qual es pot programar l'aplicació web, la seua gran escalabilitat i sobretot per la seguretat que incorpora, en una època en què l'índex d'atacs informàtics és tan alt. Quant a la seguretat, Django ofereix protecció enfront diversos atacs, com poden ser la protecció contra falsificacions de sol·licitud entre llocs (CSRF), la qual és molt fàcil d'implementar, el programador sols haurà d'incorporar un token anomenat CSRF en tots els formularis de la seua aplicació, ja que si rep un formulari que no el continga, django el descartarà automàticament.

Deixant de banda aquests avantatges, Django també permet llançar l'aplicació de forma local executant en una consola la instrucció `python manage.py runserver`, cosa que agilitza en gran manera la programació. Sols caldrà escriure l'url `127.0.0.1:8000` per a veure en temps real tots els canvis que es fan en el codi.



6.2.Llibreries auxiliars

Com ja hem explicat més amunt, un dels avantatges de Django és la seua gran comunitat i la gran varietat de llibreries les quals faciliten molt la tasca del programador. Com no podria ser d'altra manera en el nostre projecte hem emprat diverses llibreries les quals descriurem a continuació:

Font awesome: aquesta llibreria l'hem emprada per a poder utilitzar icones ja dissenyades, ja que en la seua versió gratuïta ens ofería un catàleg ampli on elegir i una gran facilitat a l'hora d'utilitzar-les al llarg del nostre projecte.

Channels: aquest llibreria l'hem emprada per poder desenvolupar el xat de missatgeria instantània ja que ens permet utilitzar websockets, entre altres noves funcionalitats.

Django.-google-maps: com el seu nom indica aquesta llibreria auxiliar ens ajuda a l'hora de connectar el nostre projecte amb l'API de google maps, per així poder beneficiar-nos de tots els avantatges que ens atorga.

6.3.Plataformes de desenvolupament

Per al desenvolupament d'aquest projecte hem emprat l'editor de textos anomenat Visual Studio Code en la versió 1.52.1.

Hem elegit aquest editor de text ja que ja teníem certa pràctica amb ell, a més estalvia faena ja que incorpora una opció d'autocompletar. Així mateix, el propi editor diferència entre els distints llenguatges de programació, com pot ser html, css, javascript o python; i avisa dels possibles errors a l'hora de programar. També ajuda a fer un control de versions automàticament, i avisa quins arxius han sigut modificats des de l'últim commit. També permet al programador obrir terminals en la carpeta del projecte, cosa que facilita llançar el projecte en local, la migració de la base de dades entre altres tasques i executar les distintes ordres per a dur un bon control de versions. Altre avantatge d'aquest editor de text és la facilitat que té per a la instal·lació de plugins, dels quals hem utilitza els següents:

- REST CLIENT : en la versió 0.24.4 per a fer proves de l'API ràpidament.
- WAKATIME: en la versió 5.0.1 per a dur un seguiment del temps que hem dedicat a desenvolupar cada part del projecte.

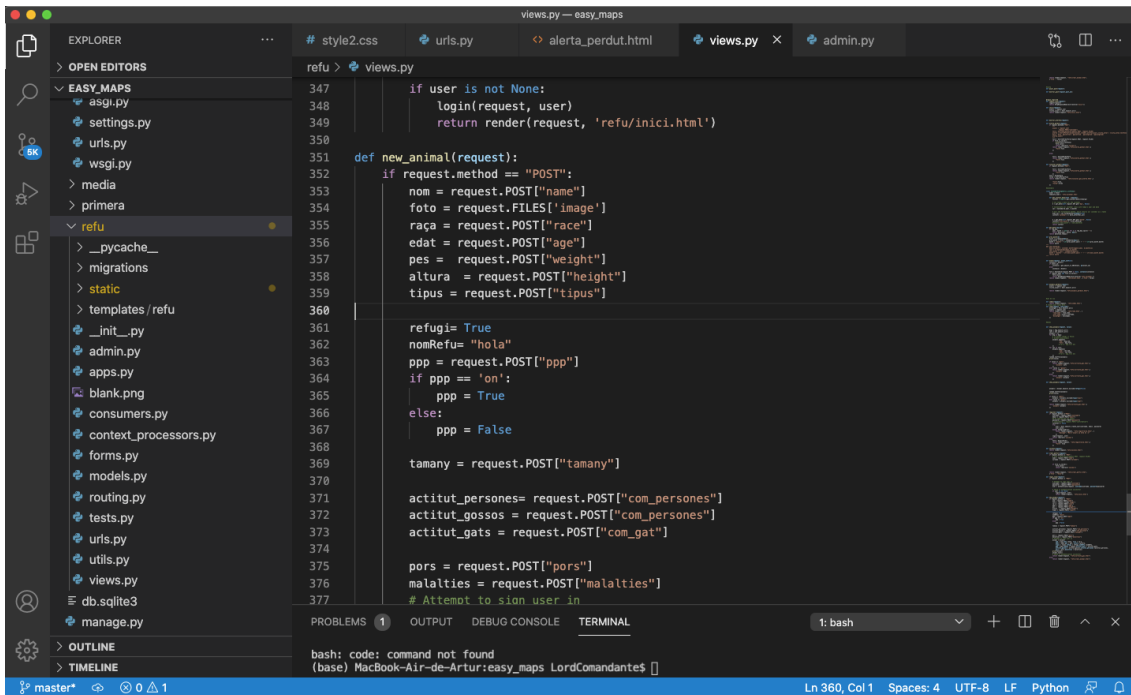


FIGURA NÚMERO 10 VISUAL STUDIO CODE

Hem elegit la ferramenta Github per a guardar el codi en el núvol i així no perdre tot el treball realitzat en cas d'avaría de l'ordinador emprat per l'autor, i també per a poder treballar i fer proves amb altres dispositius. Per al control de versió hem triat GitHub Desktop perquè coneixíem aquesta ferramenta emprada en projectes anteriors .

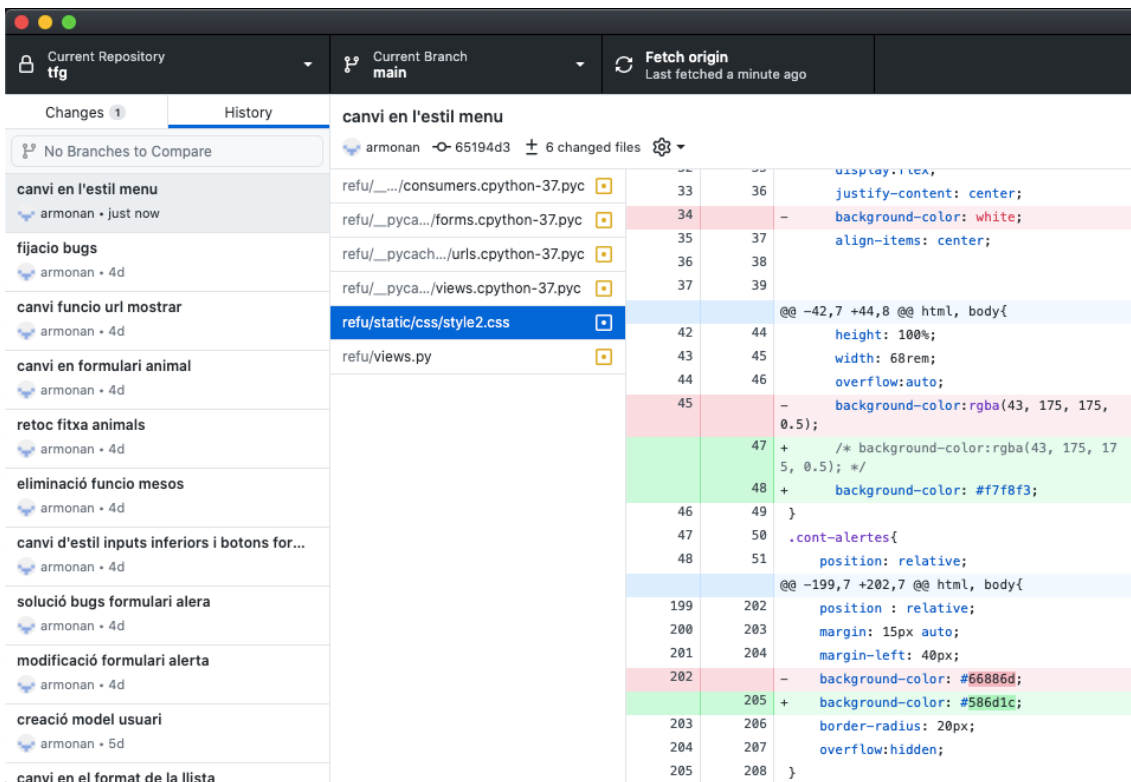


FIGURA NÚMERO 11 GITHUB DESKTOP



6.3. Estructura dels fitxers

La carpeta mestra s'anomena TFG i conté tots els fitxers i carpetes que formen aquesta aplicació web

Carpetes:

Approtectora: de creació automàtica, el seu nom coincideix amb el nom que vam inserir en la creació del projecte Django, dins d'aquesta carpeta es troben els diferents fitxers i carpetes de la nostra aplicació. En aquesta descripció sols parlarem d'aquells de creació pròpia o que considerem rellevants a l'hora de crear un projecte i obviarem aquells que són creats automàticament per Django. Entre tots els fitxers que conté els més importants són:

setting.py: En aquest fitxer creat automàticament per Django es guarda informació important del projecte com poden ser les claus secretes per xifrar, la clau de google per poder fer ús de la seua API, les aplicacions instal·lades, és a dir, les llibreries externes que hem emprat. També podem trobar la configuració de la BD.

url.py: en aquest fitxer podem trobar referències a tots els fitxers url.py que tinguem en la nostra aplicació, en el nostre cas sols tindrem un altre fitxer amb aquestes característiques.

Media: conté les imatges que es pujaran al servidor, dins d'aquesta carpeta podem diferenciar dos subcarpetes, "Animals" que contindrà les imatges referents als animals, i "Perfil" que contindrà les imatges de perfil de l'usuari.

Refugi: aquesta carpeta conté el gros de l'aplicació, dintre podem trobar tres carpetes i dotze fitxers entre els quals en destacarem quatre:

Carpetes:

Migrations: és una carpeta creada per django que conté fitxers per a la interacció entre l'aplicació i la BD.

Static: conté els fitxers estàtics de la nostra aplicació web, els quals s'agrupen en dos carpetes diferents:

CSS: conté les pàgines d'estils que hem creat al llarg del nostre projecte.

images: conté les imatges que apareixen en la nostra aplicació, com poden ser les icones.

Templates: conté tots els arxius html que componen el nostre projecte.

Fitxers:

models.py: conté els models que hem creat per a poder gestionar la persistència en la nostra base de dades.

views.py: conté les funcions que s'executaran per mostrar les vistes de la nostra aplicació, podrem observar tantes funcions com arxius html hem creat.

forms.py: conté tots els formularis que usarem en la nostra aplicació web, vinculats amb els seus models corresponents.

urls.py: aquest serà l'arxiu referenciat en l'arxiu anomenat d'igual manera, i conté una llista de les url que contindrà el nostre projecte vinculades amb les funcions corresponents de l'arxiu views.py

A més d'aquestes tres carpetes també trobarem dos fitxers, el primer anomenat db.sqlite3 que conté tota la informació que hem emmagatzemat en la nostra base de dades i l'arxiu manage.py que és creat automàticament per Django per a poder executar les diverses funcions que ens ofereix Django en la consola, com són crear un superusuari o arrancar el projecte en local.

6.4.API REST

Com que en un futur volem que aquesta aplicació web siga complementada amb una aplicació mòbil, hem creat una API perquè certa informació pugui ser consumida i la informació de l'aplicació mòbil pugui ser emmagatzemada en la BD del nostre projecte.

A continuació descriurem les diferents rutes que conformen la nostra API:

URL:/app/alertes

- GET: retornarà una llista amb totes les alertes que conté la BD.
- POST: guardarà l'arxiu JSON en la base de dades com una alerta.

URL:/app/alertes/perduts

- GET: retornarà una llista amb les alertes d'animals perduts que conté la BD.

URL:/app/alertes/trobats

- GET: retornarà una llista amb les alertes d'animals trobats que conté la BD.

URL:/app/alertes/{id_alerta}

- GET: contindrà el paràmetre id_alerta i retornarà l'alerta que coincidisca amb el paràmetre citat.

Tant la resposta rebuda en fer els GETS a les rutes dites amb anterioritat, com el format que ha de seguir el JSON per fer el POST, seguirà el següent format:

[{

 " id ":id alerta,

 " animal ": ID de l'animal del qual es fa l'alerta,

 " Creador ": ID de l'usuari que ha creat l'alerta,

 " Estat ": dos opcions possibles, trobat o perdut,



“Telèfon”: número de telèfon de contacte del creador de l'alerta,

“Hora”: hora de creació de l'alerta,

“Latitud”: latitud de la localització de l'alerta,

“Longitud”: longitud de la localització de l'alerta

}]

URL:/app/usuarios/login

- POST: la cridada post per a aquesta URL servirà als usuaris per a entrar amb el seu compte a l'aplicació mòbil, aquest compte serà el mateix que el que fan servir en l'aplicació web. El format que ha de seguir l'arxiu JSON és el següent:

{

“username”: nom d'usuari,

“password”: contrasenya de l'usuari

}

Com a resposta rebrà un arxiu JSON que contindrà un token, en cas que les credencials siguin correctes o null si no ho són.

URL:/app/usuarios/register

- POST: la cridada post per a aquesta URL serà utilitzada per a registrar-se tant en l'aplicació mòbil com en la web. El format que ha de seguir l'arxiu JSON és el següent:

{

“username”: nom d'usuari,

“password”: contrasenya de l'usuari,

“email”:el correu electrònic de l'usuari

}

7.Proves

En aquest capítol mostrarem el resultat del nostre projecte mitjançant imatges i unes breus explicacions.

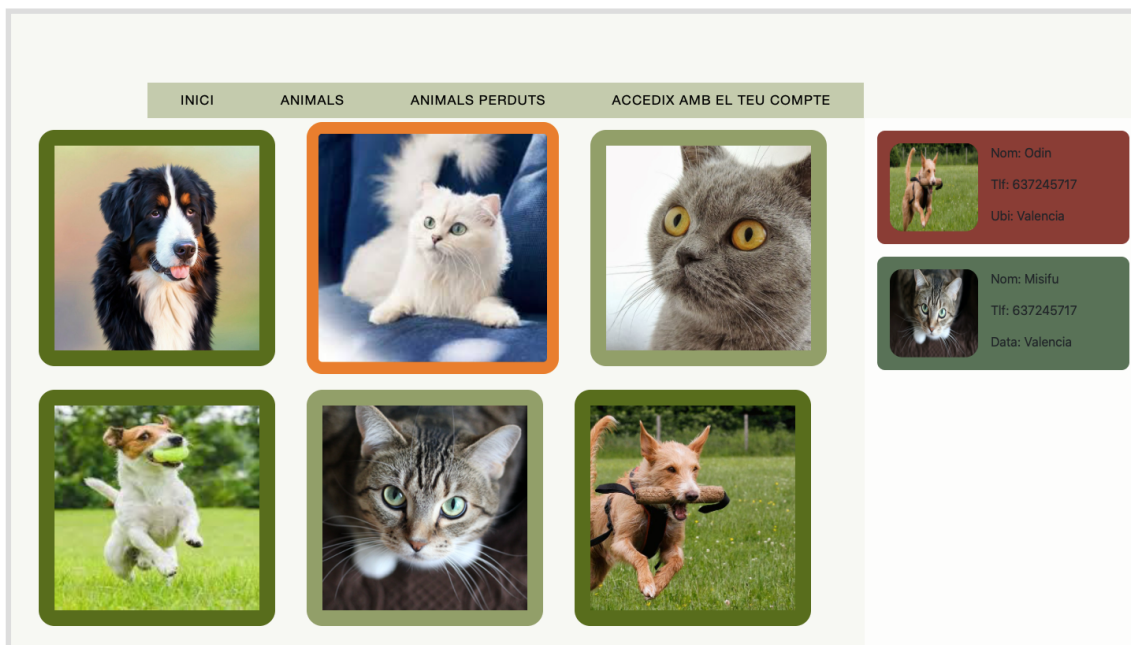


FIGURA NÚMERO 16 LLISTA D'ANIMALS

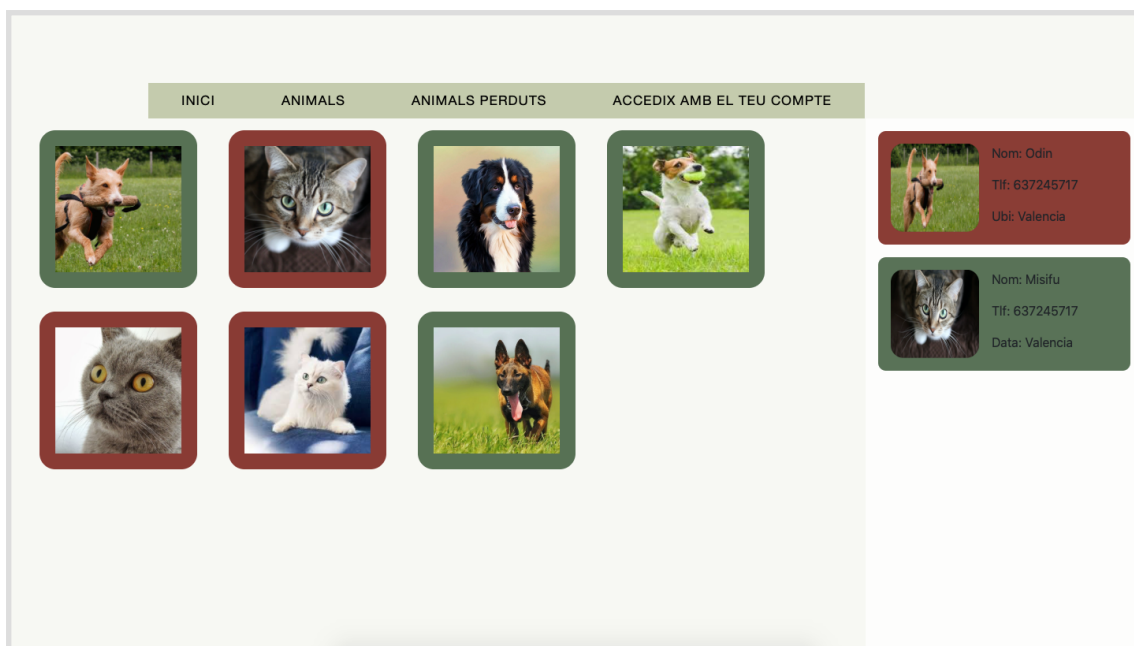


FIGURA NÚMERO 17 LLISTA D'ALERTES

En la figura número 16, podem observar la llista d'animals en adopció. Aquesta llista està formada per les imatges dels animals, emmarcades, el color del marc variarà segons el tipus d'animal, gat o gos. També podem observar que el marc canvia de color quan l'usuari passa el ratolí per damunt de la imatge per a donar-li un major dinamisme i perquè li resulte més fàcil situar-se dins de la pantalla. En la imatge que observem a continuació (número 17) podem veure que la llista d'alertes s'organitza de la mateixa manera que la d'adopcions només canvien els colors dels marcs.

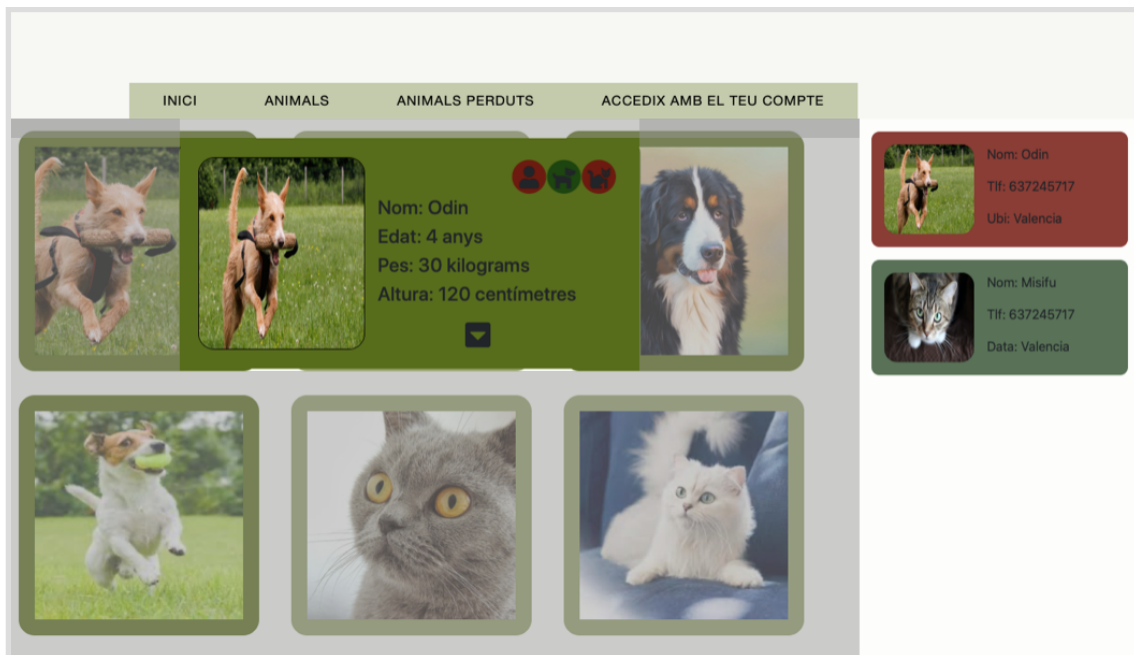


FIGURA NÚMERO 18 FITXA DEL ANIMAL.

En aquesta pantalla podem veure el que passa quan cliquem la imatge d'un animal, s'obri una finestra en la qual apareix tota la informació referent a l'animal emmagatzemada en la BD. Si ens fixem en la part dreta superior, podem observar tres icones, les quals informen sobre el comportament de l'animal amb les persones, els gossos i els gats, segons el color de fons, que podria ser roig, si l'actitud és dolenta, taronja si és normal, i verd si és bona. L'últim a comentar en aquesta finestra és el botó en forma de triangle situat en la part inferior, el qual en ser premut farà que apareguen les possibles pors i malalties de l'animal.

FIGURA NÚMERO 19 FORMULARI DE REGISTRE.

FIGURA NÚMERO 20 AFEGIR ANIMAL.

En la captura número 19 i captura número 20, podem observar com, tant el formulari de “registre” com el d’“afegir animal”, segueixen un estil semblant, l’única diferència és el nombre d’inputs dins del formulari, cosa que fa que el formulari d’“afegir animal” estiga dividit en dues columnes.

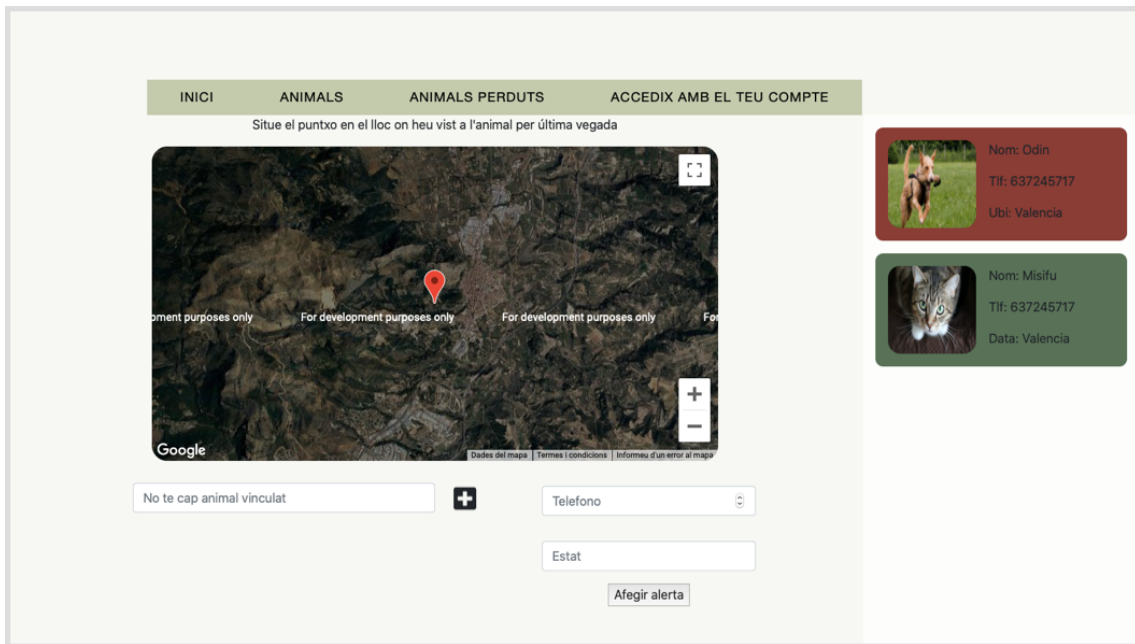


FIGURA NÚMERO 21 CREACIÓ D'ALERTES

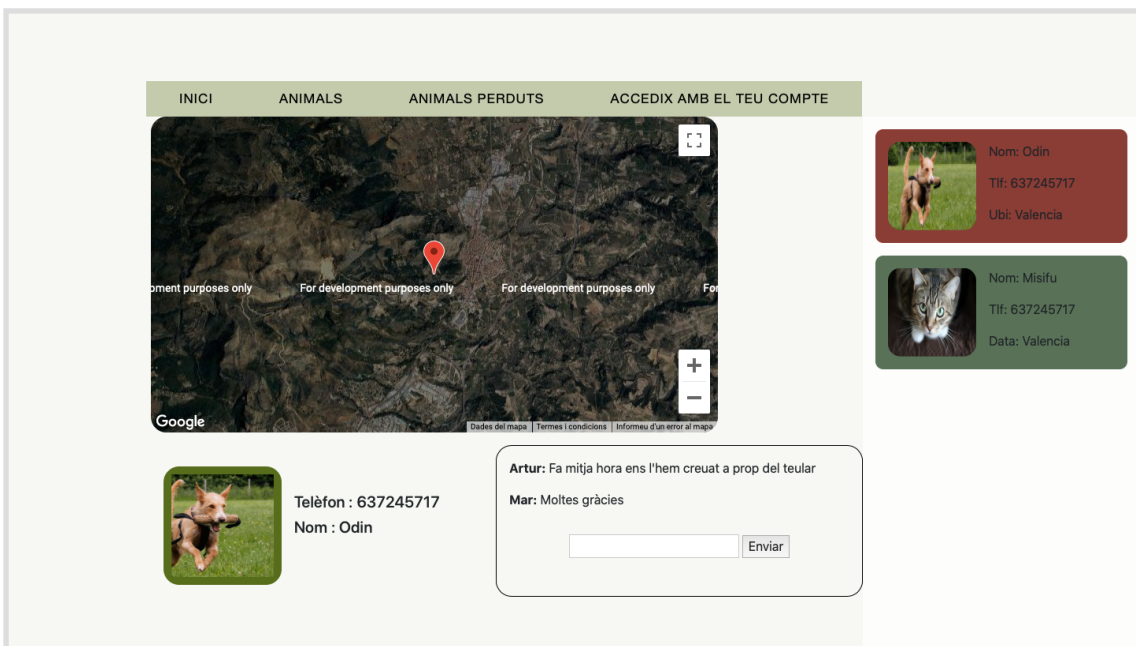


FIGURA NÚMERO 22 MOSTRAR ALERTA.

En les següents dues captures de pantalla (núm 21 i 22), podem observar que, tant la creació de les alertes com la seua visualització, segueixen una estructura pareguda. En la part central podem veure un mapa que conté una marca per senyalar la ubicació on s'ha trobat o perdut l'animal. En la part inferior podem observar, per un costat el formulari citat en el cas d'ús número 6 per a poder crear l'alerta, en aquest formulari l'usuari caldrà que seleccione l'animal que s'ha perdut si ja el té vinculat, o vincular-lo punxant en la icona del signe més, a més haurà de seleccionar el número de telèfon i l'estat de l'alerta; per l'altre costat, en visualitzar una alerta, l'usuari podrà veure a

L'esquerra tota la informació vinculada a aquesta alerta i a la dreta el xat de missatgeria instantània.

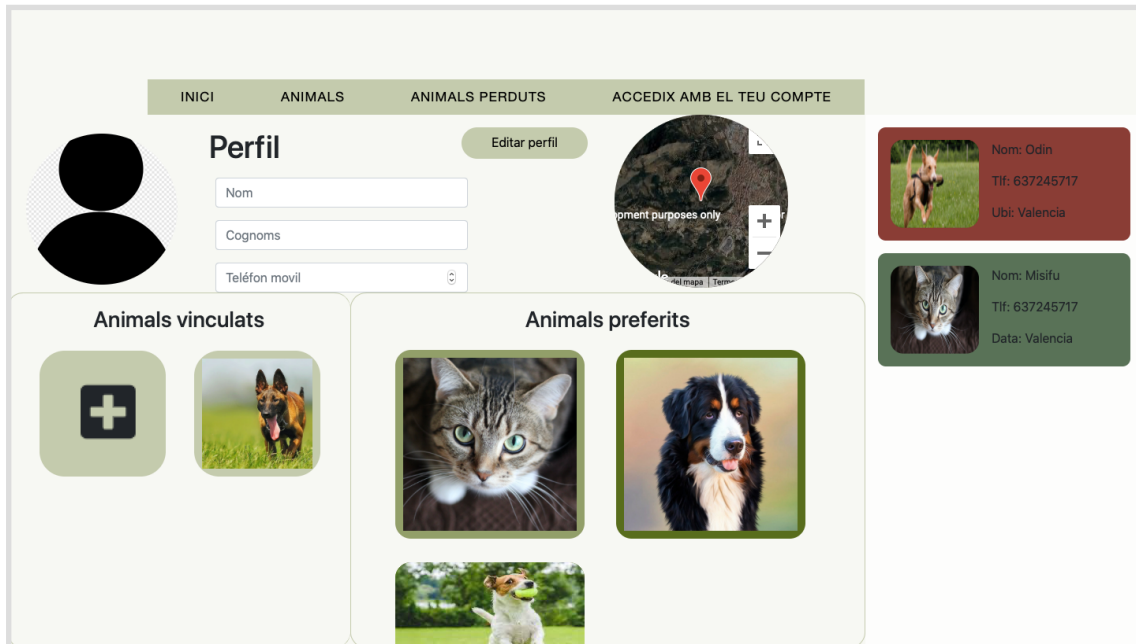


FIGURA NÚMERO 23 EDITAR PERFIL.

Per finalitzar, en l'última imatge podem veure la pantalla d'editar perfil, en la qual podem distingir dues parts. La primera, formada per un formulari i un mapa, es farà servir per completar la informació rellevant a l'hora de crear alertes. La segona part està formada per dues llistes, la primera de mascotes vinculades amb una icona per poder vincular-ne més, i la segona llista serà la d'animals preferits.

8. Conclusions

A l'hora de triar aquest projecte, érem conscients que no existia cap alternativa que cobrira les necessitats que volíem cobrir perquè, desgraciadament, com que no proporciona benefici econòmic, cap empresa privada ni estatal ha intentat cobrir-les. Sí que és veritat que existeix una gran comunitat d'amants dels animals, la qual cada vegada és més gran, tanmateix no compta amb la preparació ni les ferramentes necessàries per a anar un pas més enllà i així poder aprofitar al cent per cent els avantatges que ens atorguen les tecnologies.

Després de comentar la idea del projecte amb molta gent, veure que tots opinaven que era una bona idea, i que els animals sense casa són uns dels col·lectius no humans més vulnerables, començarem a comparar les distintes opcions que tenim per a complir els nostres objectius els qual vam identificar d'una forma molt clara.

Aquest objectius eren dos: el primer era maximitzar tot el possible la visibilitat i l'abast de les protectores per a evitar la compra venda d'animals i afavorir l'adopció; l'altre objectiu era intentar alleugerir la càrrega de treball de les protectores i evitar que animals amb famílies acabaren en aquests llocs perquè s'havien perdut modernitzant els mecanismes per a la recerca.

A partir d'aquest punt, vam marcar d'una manera més clara quines funcionalitats i requisits volíem que complira el nostre projecte. Després vam crear els distints casos d'ús i el diagrama de classes per desenvolupar el producte d'una forma coherent.

Una vegada acabada la part d'anàlisi del problema, i comparats els distints llenguatges de programació que es podien adaptar al nostre projecte, vam optar per la utilització de Python, el seu marc de treball Django ens facilitava molt la tasca de relacions en la base de dades, a més ja havíem desenvolupat més projectes emprant aquesta tecnologia i sabíem que ofería una gran quantitat d'avantages.

Deixant de banda els projectes passats, cap s'ha pogut comparar quant a mida amb aquest, cosa que fa que l'experiència personal que hem aconseguit amb aquest projecte siga molt positiva, perquè hem pogut aplicar molts dels coneixements apresos durant els anys en la Universitat en un problema real que preocupa a molta gent.

Per finalitzar, com ja hem comentat en l'apartat de treball a realitzar, encara ens queda gran part del treball per fer per a aconseguir l'objectiu que ens vam marcar però cada dia estem més prop de complir-lo i així poder ajudar en l'erradicació d'un problema no massa visible però, que d'una manera directa o indirecta, ens afecta a tots.

9. Treball futur a realitzar

Al no comptar amb tot el temps que ens haguera agradat per a poder desenvolupar aquest projecte en la seua totalitat, en aquest apartat parlarem dels passos que seguirem després de la redacció d'aquesta memòria per a poder compartir amb el món aquesta aplicació web.

9.1. Desplegament

Per a fer el desplegament de la nostra aplicació web, primerament haurem de contractar un servei d'host. Podem trobar una alta gamma de possibilitats a l'hora de contractar un servei de hosting, per a seleccionar quin s'adequa més a les característiques del nostre projecte haurem de tindre en compte molts aspectes, entre els quals destacarem els següents: quina càrrega computacional tindrà el nostre projecte, on es trobaran ubicats els servidors on desplegarem la nostra aplicació i des de quina localització accediran els possibles usuaris, quines ferramentes ens aporten per a poder fer ús dels servidors, quina escalabilitat ens ofereixen -tant horitzontalment amb el nombre de màquines com verticalment amb màquines més potents- i evidentment el cost.

En el nostre cas hem elegit Heroku per al futur desplegament. Heroku es un PaaS molt utilitzat en l'actualitat per la facilitat que aporta per a administrar, configurar i escalar l'aplicació, ja que per a desplegar-la sols s'haurà d'informar a Heroku de quin llenguatge s'ha utilitzat en el backend(part del servidor), o quina BD s'utilitzarà.

9.2. Contactar amb les protectores

Una tasca no relacionada amb la informàtica però essencial a l'hora de finalitzar el nostre projecte és la de contactar amb les distintes protectores per mostrar com aquesta aplicació les pot ajudar amb la seua tasca. Preferiblement per a realitzar aquesta tasca hauríem de concertar cites presencials per a fer una demostració als diferents administradors i treballadors per a explicar com s'utilitza i resoldre els possibles dubtes, i també per a poder demanar els permisos pertinents per fer ús de la informació de les protectores. Degut a la situació actual de pandèmia haurem de canviar el nostre *modus operandi* i concertar reunions virtuals mitjançant plataformes com Team o Skype.

9.3. Creació de l'App mòbil complementària

En la societat actual cada vegada menys gent utilitza els ordinadors ja que l'ús i les funcionalitats del mòbil estan creixent a passos gegantins, per això hem considerat que per a complir la funcionalitat referent a la pèrdua de les mascotes, el temps que es guanyarà creant l'alerta amb el mòbil, i sense haver d'esperar a arribar a casa o tindre un ordinador, pot ser vital. Per això una part del nostre projecte és desenvolupar una app complementària per a complir aquest requisit.

Com ja hem comentat en la introducció d'aquest apartat, comptem amb un temps limitat i no podem oblidar que encara som estudiants per la qual cosa el desenvolupament de la nostra aplicació mòbil haurà d'esperar, l'única cosa que podem mostrar en aquesta memòria són els mockups de les distintes pantalles i explicar com serà el seu futur ús.

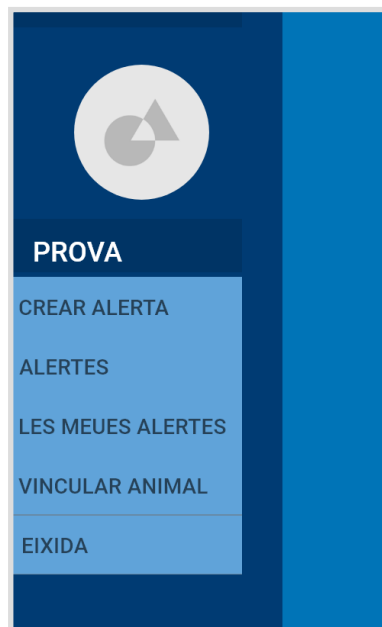


FIGURA NÚMERO 11 MOCK UP MENÚ

En el primer mock up podem observar com serà el menú de la nostra aplicació que es trobarà a l'esquerra i podrà ser visualitzat si l'usuari prem la pantalla i mou el dit cap a la dreta.

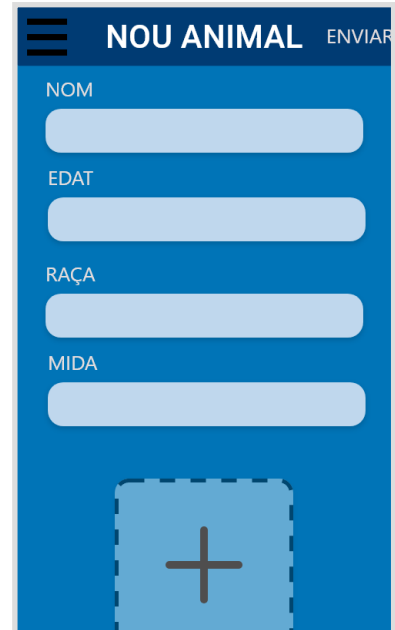


FIG. NÚM. 12 I 13 MOCK UP CREAR ALERTA I VINCULAR ANIMAL.

Els dos mock ups anteriors són el de “crear alerta” i “vincular animal” respectivament, els quals intenten copiar dins de les limitacions l’estructura que hem seguit en l’aplicació web.

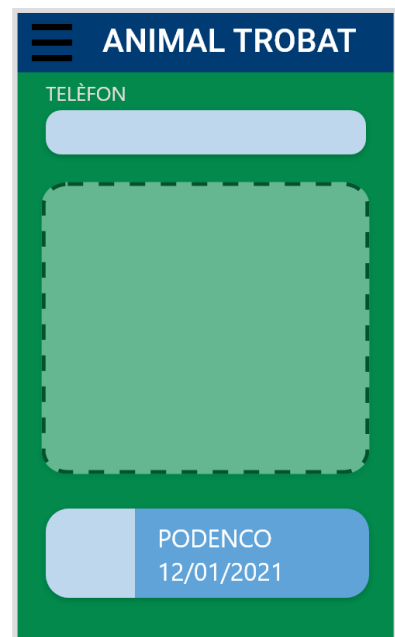


FIGURA NÚMERO 14 I 15 MOCK UP LLISTA ALERTES I ALERTA.

Per finalitzar podem observar la “llista d’alertes” que es farà servir tant per a mostrar totes les alertes com per a mostrar sols les alertes creades per l’usuari. I l’últim mock up és el que emprarem per a mostrar tota la informació de l’alerta, on el rectangle serà substituït per un mapa.

10. Bibliografia

THE WORLD'S LARGEST WEB DEVELOPER SITE

<https://www.w3schools.com/>

Django: The Web framework for perfectionists with deadlines

<https://www.djangoproject.com/>

HOLZNER, STEVEN | *Django* | Place of publication not identified Peachpit 2010 | 1st edition

Top 10 server-side programming languages in 2020

<https://ebs-integrator.com/blog/basics-top-10-server-side-programming-languages-in-2020/>

Introduction to the server side

https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction

SVAP- Sociedad Valenciana Protectora de Animales y Plantas

<https://svpap.org/>

JAISWAL, SANJEEV | KUMAR, RATAN, *Learning Django web development : from idea to prototype, a learner's guide for web development with the Django application framework* | Birmingham, England : Packt Publishing, 2015. | 1st edition