
This is the **published version** of the bachelor thesis:

Martínez Pérez, Agustín; Prim i Sabrià, Marta, dir. Geolocalització de farmàcies segons especialització o formació. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264117>

under the terms of the  license

Geolocalització de farmàcies segons especialització o formació

Agustín Martínez Pérez

Resum– Actualment, existeixen diverses aplicacions de navegació per a dispositius mòbils que permeten calcular i traçar una ruta entre la posició geogràfica dels usuaris i la destinació seleccionada. Però, l'enfocament d'aquestes és general i no es plantegen com a objectiu ajudar als usuaris en el moment d'obtenir informació detallada sobre les farmàcies. Així, amb el present projecte, s'ha desenvolupat amb React Native una aplicació mòbil per ajudar als usuaris a trobar les farmàcies segons l'especialització o formació d'aquestes. A més, per permetre a les farmàcies introduir i gestionar la seva informació, s'ha creat una aplicació web amb Vue 3. Finalment, per gestionar totes les dades del sistema, s'ha desenvolupat un servei de *backend* amb Node.js i una base de dades MongoDB per emmagatzemar la informació. Per tant, s'ha creat un ecosistema d'aplicacions que permet als usuaris trobar les farmàcies que poden cobrir les seves necessitats més específiques.

Paraules clau– Càlcul de ruta, especialització, farmàcia, formació, geocodificació, geolocalització, mapa interactiu.

Abstract– Nowadays, there are several navigation applications for mobile devices that calculate and trace routes between the users' geographical position and the selected destination. But, their approach is broader and does not aim to help users find detailed information about pharmacies. Thus, with this project, a mobile application has been developed with React Native to help users find pharmacies according to their speciality or workers' training. In addition, to allow pharmacies to insert and manage their information, a web application has been created with Vue 3. Finally, to manage all the system data, a backend service has been developed with Node.js and a MongoDB database to store all the information. Therefore, an ecosystem of applications has been created that allows users to find the pharmacies that can cover their most specific needs.

Keywords– Route calculation, speciality, pharmacy, education, geocoding, geolocation, interactive map.



1 INTRODUCCIÓ

L'ESPECIALITZACIÓ de les farmàcies és cada vegada més habitual a causa de la competència existent, a més d'altres factors com, per exemple, la formació dels professionals. Per tant, per poder obtenir una rendibilitat econòmica, les farmàcies han de trobar noves formes de descobrir potencials clients i fidelitzar-los. La primera possibilitat que es podrien plantejar aquestes és distribuir

continguts publicitaris als mitjans audiovisuals de consum massiu, com podria ser la televisió o la ràdio. El problema principal que es presenta en aquest punt és que, encara que la legislació espanyola (article 27) [1] no prohibeix explícitament la publicitat dels centres i establiments sanitaris, grup que inclou també a les farmàcies, la legislació d'algunes comunitats autònomes com Aragó, Astúries, Canàries, Cantàbria, Extremadura, Galícia, Comunitat de Madrid i Múrcia sí que ho prohibeix expressament. També Google prohibeix la promoció i publicitat de farmàcies de manera *on-line* (a través de Google Ads) [2] a Espanya, independentment de la comunitat autònoma a la qual estigui ubicat l'establiment farmacèutic.

- E-mail de contacte: agustin.martinezp@autonoma.cat
- Menció realitzada: *Tecnologies de la Informació*
- Treball tutoritzat per: *Marta Prim Sabrià (Departament de Microelectrònica i Sistemes Electrònics)*
- Curs 2021/22

Per tant, a causa dels factors exposats, l'especialització i assessorament personalitzat als clients és una de les maneres que les farmàcies utilitzen per aconseguir nous clients amb necessitats específiques, arribant a convertir-los en cli-

ents habituals de l'establiment.

D'altra banda, per a les persones que requereixen tractaments més concrets, aquestes prohibicions suposen una dificultat extra per determinar quin és el camp de treball en el qual es focalitza cada farmàcia. A més, el nombre de farmàcies que ofereixen serveis específics com l'ortopèdia o l'audiometria, entre d'altres, és reduït a causa de l'obligació, per part dels professionals que hi treballen, de disposar de titulacions específiques en aquests camps. En conseqüència, encara és més complicat, per a les persones que requereixen aquests tractaments, trobar una farmàcia que pugui cobrir les seves necessitats particulars.

En referència a l'estructura del present document, a continuació, s'exposen els objectius d'aquest treball, un resum d'altres serveis o aplicacions similars que existeixen actualment i una descripció de la metodologia i el seguiment que s'ha dut a terme durant el desenvolupament del projecte. Posteriorment, es defineixen en detall tots els components que formen el sistema dissenyat i, finalment, es realitza una anàlisi dels resultats obtinguts i unes conclusions sobre aquest projecte, el qual rep el nom de PharmApp.

2 OBJECTIUS

L'objectiu principal del projecte és intentar facilitar als usuaris el procés de recerca de farmàcies segons la seva especialització o formació, permetent-los així trobar les farmàcies que puguin arribar a cobrir les seves necessitats més específiques. A més, per a les farmàcies, també es pretén facilitar la tasca de decidir quina informació podran trobar els usuaris sobre aquestes, oferint una plataforma a través de la qual sigui senzill introduir informació, eliminar-la i modificar-la en qualsevol moment. Finalment, cal destacar que aquest sistema ha d'estar preparat per a desplegar-se en un entorn real i ha de ser prou simple i clar per poder ser utilitzat per persones que no han tingut cap contacte amb les respectives aplicacions ni informació prèvia al respecte.

Aleshores, per permetre a les farmàcies introduir i gestionar la seva informació, tant general (com el nom, telèfon de contacte, etcètera) com l'especialització, s'ha desenvolupat un frontal al qual es pot accedir mitjançant un navegador web convencional. Aquesta decisió es basa en el fet que la gran majoria de farmàcies utilitzen ordinadors per gestionar-se i els treballadors tenen accés a ells durant tota la seva jornada laboral. A més, tot i la facilitat de què es disposa en el moment d'accedir a l'aplicació a través d'un navegador web instal·lat a un computador portàtil o de sobretaula, aquest frontal també està adaptat per a dispositius mòbils amb mides i resolucions de pantalla molt diverses. Per tant, el disseny és *responsive* per permetre l'accés al servei des de pràcticament qualsevol dispositiu connectat a Internet.

En referència als clients, per aconseguir l'objectiu de facilitar la tasca de trobar informació sobre les farmàcies, s'ha desenvolupat una aplicació per a *smartphone*, ja que la gran majoria de persones posseeixen i utilitzen un telèfon intel·ligent a diari. Així, aquesta aplicació dona accés a tota la informació rellevant que les farmàcies facin pública dins d'aquest sistema. A més, mostra un mapa interactiu amb el qual, els clients, podran visualitzar la ubicació de les farmàcies, filtrar els resultats segons l'especialització que sigui d'interès a cada moment i, també, es permet calcu-

lar la ruta òptima per arribar des de la ubicació actual de la persona fins a la farmàcia seleccionada, facilitant encara més el procés d'accedir a una farmàcia que pugui aportar les solucions necessàries.

3 ESTAT DE L'ART

Actualment, existeixen grans companyies com Google i Apple, entre d'altres, que han desenvolupat aplicacions de mapes utilitzades per milions de persones diàriament [3]. Durant l'any 2019, Google Maps va tenir 126,2 milions d'usuaris mensuals a la seva aplicació per telèfons intel·ligents només als Estats Units d'Amèrica (EUA), i més de 1000 milions d'usuaris mensuals arreu del planeta [4]. També Apple va tenir una gran quantitat d'usuaris mensuals a la seva aplicació Apple Maps, arribant als 29,3 milions d'usuaris als EUA. Una altra aplicació de mapes és Waze, la qual va comprar Google l'any 2013 i, durant el 2019, va tenir 31 milions d'usuaris mensuals, també als EUA.

La diferència entre aquestes aplicacions i la que es presenta en aquest treball és la seva finalitat, ja que les tres mencionades estan enfocades al públic general i serveixen per ajudar als usuaris a arribar al seu destí, independentment del tipus que sigui. En canvi, no s'ha desenvolupat, fins al moment, una aplicació de mapes que estigui destinada a trobar farmàcies segons l'especialització o la formació dels treballadors d'aquestes. Així, fins ara, els clients han de navegar pels llocs web de les diferents farmàcies per aconseguir l'especialització de cadascuna d'elles, però no hi ha cap directori estandarditzat. Per tant, descobrir aquesta informació sobre les farmàcies és una tasca relativament complexa en alguns casos, i encara més per a les persones amb pocs coneixements sobre tecnologia.

4 METODOLOGIA

Durant el Grau d'Enginyeria Informàtica s'imparteixen diferents metodologies que són útils (i recomanables) per gestionar correctament els recursos, tant materials com no materials, de què es disposa per al desenvolupament d'un projecte de *software*. Algunes de les més destacades són SCRUM, Kanban, Lean Development o eXtreme Programming.

Amb l'experiència obtinguda durant els estudis, l'experiència laboral acumulada fins al moment i la tipologia d'aquest projecte, s'ha considerat que la metodologia que millor s'ajusta per fer el seguiment del desenvolupament i l'organització de les tasques és Kanban. Això és degut al fet que la informació es mostra d'una manera molt visual i permet, de manera ràpida, situar-se al projecte i comprendre l'estat actual d'aquest. Per treure encara més profit d'aplicar aquesta metodologia de treball, s'ha mantingut un tauler Kanban en format digital, emprant l'aplicació Trello. En aquest tauler s'han afegit les llistes que es considera que permeten visualitzar, de la millor manera possible, l'estat global del desenvolupament del projecte. Concretament, les llistes que formen el tauler són les següents:

- *Backlog*: serveix per emmagatzemar inicialment totes les tasques que s'han de desenvolupar per poder aconseguir els objectius definits per al projecte, però

no conté les tasques que s'han de desenvolupar en el *sprint* actual.

- **To-Do:** conté les tasques a desenvolupar en el *sprint* actual, però que no s'està treballant en elles encara.
- **Doing:** emmagatzema les tasques en les quals s'està treballant en aquell moment concret.
- **Testing:** conté les tasques que determinen quins components del sistema han estat finalitzats i, per tant, s'està comprovant el correcte funcionament.
- **Done:** serveix per emmagatzemar totes les tasques que ja han estat finalitzades durant aquest *sprint* i, a més, han passat satisfactòriament les proves de funcionament.
- **Delivered:** conté totes les tasques finalitzades i que han estat entregades. Generalment, s'afegeixen les tasques a aquesta llista en el moment en què es fa el lliurament d'un informe, ja que actuen com fites aconseguides i disposen de dates molt concretes definides al començament del projecte.

A continuació, s'inclou una captura (Figura 1) de l'estat del tauler Kanban en el moment de l'entrega del present informe. Com es pot observar, la captura mostra com aquest tauler serveix per estructurar les tasques a realitzar i aporta uns grans beneficis, oferint una visió general de l'estat del projecte.

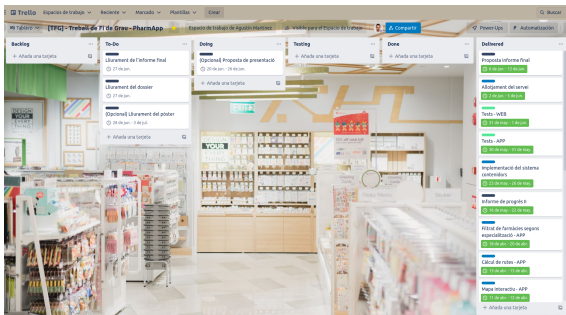


Fig. 1: Estat del tauler Kanban gestionat amb Trello.

Adicionalment, per controlar si aquesta planificació s'està seguint durant el transcurs del projecte o, per contra, si hi ha una diferència respecte de les dates previstes per a l'execució de les tasques i el temps de què es disposa, s'ha combinat la gestió de les tasques del tauler Kanban amb un Diagrama de Gantt. A continuació, es mostra la imatge (Figura 2) del diagrama amb la planificació temporal de les tasques:

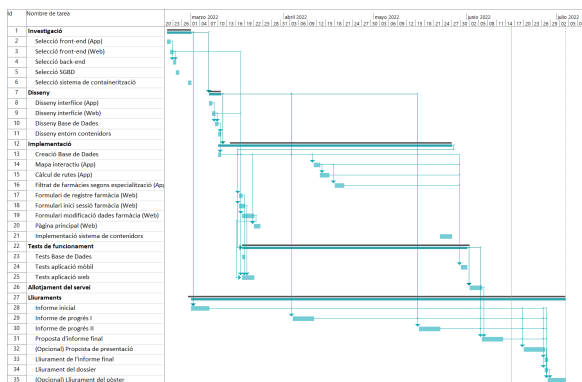


Fig. 2: Diagrama de Gantt del projecte.

4.1 Planificació

Després de definir tots els requisits del projecte és necessari planificar com totes les tasques a desenvolupar s'han de distribuir per poder lliurar el projecte dins del termini de temps de què es disposa. Per millorar la planificació ha estat útil crear fases de desenvolupament amb objectius ben definits. Per això s'han marcat tres fites que han permès conèixer si el desenvolupament del projecte està transcorrent de la forma prevista o no.

La primera fase ha finalitzat amb l'entrega de l'Informe Inicial del Projecte, el dia 6 de març de 2022. Durant aquesta, s'han dut a terme principalment tasques d'investigació, mitjançant les quals s'han examinat les diverses tècniques i eines de desenvolupament i s'han seleccionat aquelles considerades òptimes per poder desenvolupar amb èxit aquest projecte. Posteriorment, s'ha creat l'esquema del sistema complet, indicant les relacions existents de comunicació entre cadascun dels mòduls implicats.

La segona fase ha finalitzat amb l'entrega de l'Informe de Progrés I, el dia 10 d'abril de 2022. Inicialment, per completar-la, era necessari finalitzar el desenvolupament de la base de dades i, pràcticament, de l'aplicació mòbil sencera. El problema que es planteja en aquest punt és que, durant el desenvolupament d'aquesta segona fase, és necessari poder obtenir les ubicacions de les farmàcies per mostrar-les al mapa de l'aplicació mòbil. És per aquest motiu que es va decidir aturar temporalment el desenvolupament d'aquesta i centrar-se en el servei web. En conseqüència, en permetre a les farmàcies poder registrar la seva informació a través de l'aplicació web, el desenvolupament de l'aplicació mòbil és molt més senzill, ja que es disposa, per tant, d'informació de les farmàcies semblant a la que pot ser utilitzada en un entorn d'execució real. Així, les modificacions produïdes en l'ordre de desenvolupament s'han aplicat amb l'objectiu de millorar l'eficiència durant el transcurs posterior del projecte.

Finalment, la tercera fase finalitza amb l'entrega de l'Informe de Progrés II, el dia 22 de maig de 2022. Durant aquesta s'ha completat el desenvolupament de l'aplicació web, corregint alguns errors i, també, s'ha completat la creació de l'aplicació mòbil. Així doncs, en aquest moment, el projecte es troba a la seva fase final i està llest per poder funcionar correctament en un entorn local.

Després de completar aquestes tres fases, el projecte és completament funcional. Per tant, a partir d'aquest moment, l'objectiu ha estat millorar l'experiència i el desplegament dels components que formen el sistema. Així, s'han desplegat els serveis a través de contenidors Docker i s'han allotjat al servidor virtual privat d'OVH, tal com es descriu als apartats 8 i 9.

5 DISSENY DE LA INTERFÍCIE

Tot i que aquest projecte està centrat en el desenvolupament tècnic del sistema que permet als usuaris trobar informació rellevant sobre les farmàcies, és necessari dissenyar la interfície visual abans de començar a construir el projecte, ja que així es defineixen unes línies clares que determinaran l'aspecte final dels diferents components i, també, ajudaran a fixar quines són les funcions necessàries que s'han de desenvolupar. A més, aquest disseny inicial permet crear

una interfície que sigui fàcil d'utilitzar, tant pels treballadors de les farmàcies que gestionaran la informació d'aquestes a través de l'aplicació web, com pels clients que utilitzin l'aplicació mòbil.

Així, per dissenyar aquests prototips (Apèndix A.2), s'ha fet servir Figma, un editor de gràfics vectorials molt utilitzat per dissenyadors d'aplicacions, atès que els permet generar pràcticament qualsevol interfície que puguin imaginar. A més, ofereix la possibilitat de generar prototips funcionals que simulen el comportament final del sistema, permetent així poder rebre *feedback* per part de potencials clients abans de començar amb el desenvolupament i, també, com ja s'ha comentat, permet als desenvolupadors conèixer quin ha de ser el comportament final del sistema que han de crear.

6 DESENVOLUPAMENT

La solució proposada per poder aconseguir l'objectiu d'aquest projecte consisteix a desenvolupar una aplicació que pugui ser instal·lada en un *smartphone*, una aplicació a la qual es pugui accedir a través d'un navegador web, i un sistema que permeti emmagatzemar i gestionar tota la informació i les peticions que aquestes dues aplicacions generin. Per tant, aquest projecte consisteix en el desenvolupament *full-stack* del servei que pretén solucionar els problemes descrits a l'apartat d'introducció.

A continuació, s'inclou el diagrama (Figura 3) que mostra gràficament quins són els diferents components que formen el sistema, els quals estan descrits en profunditat als següents apartats.

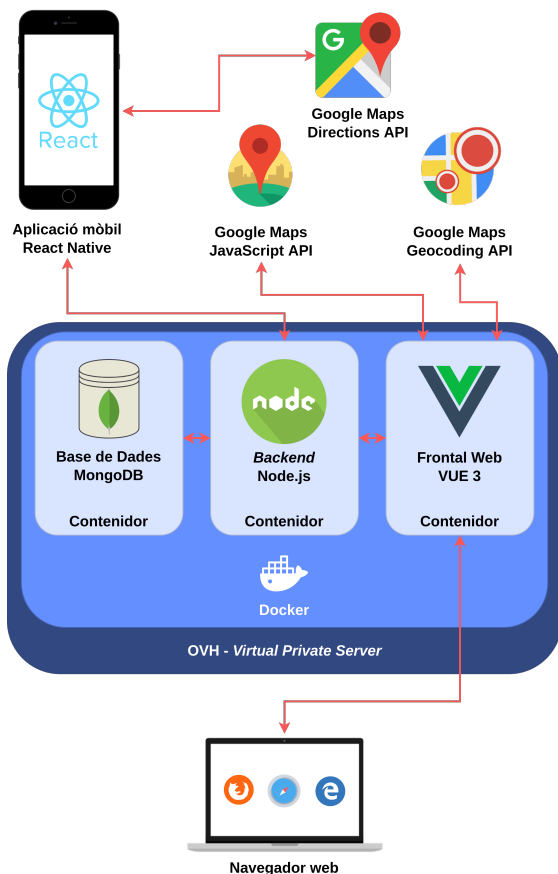


Fig. 3: Components que formen el sistema.

6.1 Base de dades

Per emmagatzemar la informació que utilitzaran les aplicacions de PharmApp s'utilitza MongoDB, un Sistema Gestor de Bases de Dades NoSQL de codi obert que ofereix una gran versatilitat i velocitat. A més, requereix pocs recursos computacionals i es complementa perfectament amb aplicacions desenvolupades en JavaScript, com és el cas tant de l'aplicació mòbil com de l'aplicació web i el *backend*.

Pel que fa a les col·leccions definides per emmagatzemar les dades, s'han creat dues diferents. La primera, que rep el nom de *specialities*, té l'objectiu d'emmagatzemar la informació de totes les especialitats que poden estar associades a una farmàcia, i està formada per documents amb una estructura molt simple. Concretament, cada document només emmagatzemarà l'identificador únic de cada especialitat i el nom d'aquesta.

La segona col·lecció, anomenada *pharmacies*, serveix per emmagatzemar tota la informació que les farmàcies introdueixen a través de l'aplicació web. L'estructura dels documents que emmagatzema aquesta col·lecció és més complexa, ja que conté els següents camps per emmagatzemar informació: identificador únic, nom de la farmàcia, especialitat, telèfon, lloc web, correu electrònic, ciutat, carrer, codi postal, país, adreça completa, latitud, longitud, horari d'obertura al públic, identificador dels serveis de Google (*Place ID*), contrasenya (s'emmagatzema un *hash* d'aquesta), la *salt* (és la llavor que s'utilitza com a entrada per calcular el *hash* de la contrasenya, evitant així la possibilitat de realitzar atacs de *Rainbow Tables* [5] contra la base de dades del projecte) i, finalment, un camp per detectar si és una farmàcia l'establiment que s'ha registrat.

6.2 Aplicació web — PharmApp

L'aplicació web, o *frontend*, de PharmApp té l'objectiu de ser l'eina que utilitzen les farmàcies per gestionar la seva informació a la plataforma i per configurar quines dades es mostraran als usuaris que utilitzin l'aplicació mòbil per trobar-les.

Aquesta s'ha desenvolupat utilitzant VUE 3, que és un *framework* [6] de JavaScript que permet crear les interfícies d'usuari dels llocs web. S'ha decidit fer servir aquest entorn per dos motius, principalment. El primer és la facilitat i l'alt rendiment que ofereix en el moment de crear aplicacions del tipus *Single Page Application* (SPA), les quals permeten crear interfícies molt interactives i que no necessiten recarregar la pàgina completa en el moment en què les dades s'actualitzen o es navega entre els diferents recursos de l'aplicació.

A causa de la naturalesa de Vue.js pel que fa a la creació de SPAs, és necessària la utilització de Vue Router per permetre a l'usuari navegar entre els diferents components que formen l'aplicació. És així ja que, en cap moment, es recarregarà la pàgina en sol·licitar un nou recurs a l'aplicació (sempre que l'usuari no premi el botó de recarregar la pàgina manualment). Així, per poder mostrar els components dinàmics amb els quals interactuarà l'usuari a través del seu navegador web, es fa servir el *tag* `<router-view>`. A més, per permetre a l'usuari navegar entre aquests components, com si es tractés de pàgines diferents, s'utilitza el *tag* `<router-link>`, el qual subs-

titueix a la clàssica àncora (<a>) d'HTML.

En referència a l'estructura interna dels fitxers que defineixen els components que formaran el frontal del lloc web, aquests segueixen una arquitectura del tipus *Single-File Components* [7]. Aquest format dels fitxers encapsula la lògica del component (definida amb JavaScript), la part visual (anomenada plantilla o *template*, i definida amb HTML) i, també, els estils que s'aplicaran als elements que formen la plantilla (definits amb CSS).

El segon motiu pel qual s'ha decidit utilitzar aquest *framework* és perquè disposa d'una gran comunitat de desenvolupadors que ajuden a resoldre possibles errors de programació i dubtes que puguin sorgir durant aquesta fase. Així, Vue.js s'ha convertit en un dels *frameworks* més utilitzats [8] per al desenvolupament web.

6.2.1 Google Maps JavaScript API i Google Maps Geocoding API

El primer pas que una farmàcia ha de seguir per començar a formar part de PharmApp és registrar-se. Per fer-ho, el formulari que ofereix l'aplicació web requereix completar-lo amb diferents dades d'aquesta nova farmàcia. Seguidament, amb aquesta informació proporcionada, s'utilitza la Google Maps Geocoding API [9] per tal d'obtenir la informació de geolocalització a partir de l'adreça, la ciutat, el codi postal i el nom de l'establiment. Addicionalment, aquesta API serveix per comprovar si l'establiment que s'està intentant registrar és realment una farmàcia, ja que la resposta conté una llista de termes que defineixen l'activitat que realitza el comerç o institució. A continuació, es mostra una imatge (Figura 4) que compara les paraules clau obtingudes en realitzar la cerca d'una farmàcia respecte de les paraules clau obtingudes a partir de la cerca d'un establiment no farmacèutic, com és la Universitat Autònoma de Barcelona:

Farmàcia	Universitat
0: "establishment"	0: "establishment"
1: "health"	1: "point_of_interest"
2: "pharmacy"	2: "university"
3: "point_of_interest"	length: 3
4: "store"	
length: 5	

Fig. 4: Paraules clau resultants del procés de geolocalització.

Una vegada geolocalitzat l'establiment i, només, si aquest és una farmàcia, es fa servir la funció *getDetails* de la Google Maps JavaScript API [10] per poder aconseguir la informació detallada de l'establiment com, per exemple, l'horari d'obertura al públic, el qual es podrà modificar posteriorment a través de l'aplicació web de PharmApp.

Com es pot observar, l'obtenció d'informació de geolocalització i sobre els establiments recau sobre la plataforma de Google. Existeixen, principalment, dos motius per la utilització d'un servei extern com aquest. En primer lloc, cal crear un mecanisme per realitzar una verificació molt concreta que permeti determinar si una entitat que es registra és realment una farmàcia. Per dur a terme aquesta tasca, seria necessari tenir accés a la llista completa d'establiments sanitaris que ofereixen serveis farmacèutics, però no és senzill d'obtenir i varia en funció del país en el qual es troba l'establiment. El segon motiu, i derivat d'aquest primer, és la gran cobertura de geocodificació i mapes que

ofereix Google públicament, ja que proporciona dades sobre pràcticament la totalitat de països del planeta [11]. Així, es pot comprovar amb un alt percentatge d'encert si un establiment és una farmàcia realment, independentment del país des d'on es registri.

6.2.2 Descripció del funcionament de l'aplicació

Una vegada registrada la nova farmàcia, aquesta podrà utilitzar el formulari d'inici de sessió que ofereix l'aplicació per accedir al seu compte.

A la pantalla principal es mostrarà tota la informació que la farmàcia ha introduït durant el procés de registre o que hagi modificat posteriorment. Així, sense haver de navegar entre diferents pestanyes, es pot obtenir una visió general de les dades que els usuaris poden visualitzar sobre aquella farmàcia concreta quan utilitzin l'aplicació mòbil de PharmApp.

També, sense haver de navegar a altres components del lloc web, es permet editar tota la informació d'una manera molt senzilla: a la part inferior de la pantalla apareix un botó que permet habilitar l'edició de la farmàcia. En prémer-lo, els camps de text s'habilitaran i es permetrà modificar qualsevol dada. També, en habilitar l'edició, apareixen quatre nous botons: el primer permet canviar la contrasenya, i redirigirà a una altra pàgina que conté el formulari per modificar-la. El segon botó permet cancel·lar l'edició. Per tant, les dades modificades no s'emmagatzemaran a la base de dades. El tercer botó permet eliminar el compte; com és una acció delicada i no pot desfer-se, requereix una segona confirmació per part de l'usuari. Així, en cas de ser una equivocació, la farmàcia no s'esborrarà per error i, en cas de voler eliminar-la definitivament, només caldrà tornar a prémer un botó. El quart i últim botó serveix per guardar els canvis i actualitzar la informació a la base de dades.

Així doncs, exceptuant la utilització del botó d'eliminar la farmàcia, en finalitzar el procés d'edició i desar els canvis es retornarà a l'usuari a la pàgina principal perquè pugui visualitzar tota la seva informació de nou.

A continuació, s'inclou una captura (Figura 5) on es pot observar la simplicitat de la disposició dels components i la facilitat que tenen els professionals que treballen a les farmàcies per modificar qualsevol dada introduïda a la plataforma:

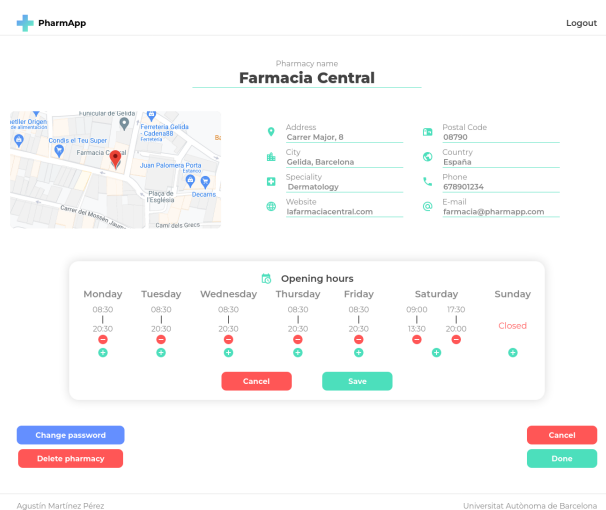


Fig. 5: Pantalla d'edició de la farmàcia a l'aplicació web.

6.3 Aplicació mòbil — PharmApp

Aquesta aplicació té l'objectiu de ser la interfície que poden utilitzar els usuaris per trobar les farmàcies d'interès i visualitzar la seva posició geogràfica sobre un mapa interactiu.

El desenvolupament d'aquesta ha estat mitjançant React Native, que és un *framework* de JavaScript que permet crear aplicacions per diferents sistemes operatius i fer que aquestes s'executin com si es tractés d'un desenvolupament natiu, quant a rendiment i disseny de les aplicacions. Gràcies a aquestes característiques, React Native ha aconseguit generar una gran comunitat de suport [12], pel que l'elevat rendiment, la modularitat dels components i la gran comunitat de desenvolupadors que el suporten han estat factors determinants en l'elecció d'aquest *framework*.

El problema que ha sorgit en aquest punt és que, per poder desenvolupar una aplicació per *smartphones* basats en iOS, és a dir, telèfons intel·ligents iPhone, és necessari utilitzar un computador de la família Mac, comercialitzats també per Apple. Per tant, com que no es disposa d'un computador d'aquest tipus, ja que tot el projecte es desenvolupa fent servir el sistema operatiu Ubuntu 20.04, no es pot realitzar la compilació de l'aplicació per a dispositius basats en iOS. Tot i això, el desenvolupament amb React Native permet fer ús del mateix codi font per compilar les aplicacions d'iOS i Android. Per tant, només caldria processar el codi desenvolupat durant el projecte en un computador Mac, dur a terme alguns canvis mínims a diversos fitxers de configuració i l'aplicació estaria completament llesta per ser compilada també per iOS, sense haver de fer grans modificacions al codi font original.

6.3.1 Google Directions API

Respecte a la ruta entre l'usuari i la farmàcia seleccionada, es mostra al mapa una línia que servirà de guia en el moment del desplaçament. Per realitzar el càlcul d'aquesta, entre els dos punts seleccionats (la ubicació de l'usuari i la farmàcia a la qual es vol arribar), s'ha utilitzat la llibreria React Native Maps Directions [13]. A través d'aquesta es permet a React Native interactuar amb l'API Google Maps Directions [14], la qual s'ocupa de determinar la ruta òptima entre ambdós punts i segons el mètode de desplaçament seleccionat.

A causa del fet que es poden seleccionar diversos modes de desplaçament (com per exemple a peu o mitjançant un vehicle motoritzat) les rutes han de ser calculades de forma diferent. Així, en el moment de realitzar la petició a l'API de Google perquè retorni la ruta òptima, s'ha d'especificar el mètode de desplaçament desitjat. Per tant, a través de la interfície de l'aplicació, l'usuari tindrà l'opció de seleccionar el mitjà de transport que utilitzarà per dur a terme el seu desplaçament.

6.3.2 Descripció del funcionament de l'aplicació

Respecte al funcionament de l'aplicació, inicialment es mostra un mapa interactiu que permet al client visualitzar la seva posició geogràfica en temps real, la qual cosa permetrà seguir la ruta des de la seva posició fins a la farmàcia que desitgi. També, a la mateixa pantalla principal, es mostra un

selector d'especialitats que permet a l'usuari seleccionar-ne una de la llista. Una vegada seleccionada, es localitzen i es mostren al mapa totes les farmàcies que ofereixen serveis d'aquella especialitat. A més, en aquest mateix instant, apareix una llista horitzontal que conté la informació rellevant de les farmàcies, en forma de targetes, amb les quals l'usuari pot interactuar. A cadascuna d'aquestes targetes hi ha un botó d'informació que, en prémer-lo, obrirà una nova pestanya on es mostrarà tota la informació que hi ha registrada sobre la farmàcia i que pot ser d'utilitat per al client. Si, en canvi, se selecciona una farmàcia de la llista, el mapa de l'aplicació se centrarà en aquesta i es mostraran dues icones per començar la ruta des de la posició actual, sigui a peu o utilitzant un vehicle motoritzat que tingui permès circular per les vies públiques, segons l'elecció de l'usuari.

La decisió d'incorporar únicament aquests dos mètodes de circulació és, bàsicament, per simplicitat. A causa de la distribució amb què es mostren els diferents components del *layout* de l'aplicació mòbil, incorporar més opcions degradaria l'experiència d'usuari, ja que els elements haurien de ser massa petits per poder-los visualitzar a la vegada. L'únic inconvenient d'aquesta decisió és la impossibilitat de calcular la ruta en transport públic. Però, aquest podria ser un punt de millora per a l'aplicació en el futur.

Seguidament, es mostren dues captures (Figures 6a i 6b) de l'aplicació mòbil, mitjançant les quals es pot observar la vista principal en el moment de filtrar les farmàcies segons la seva especialització i, també, es pot visualitzar com es mostra la ruta i la posició de l'usuari a sobre del mapa:

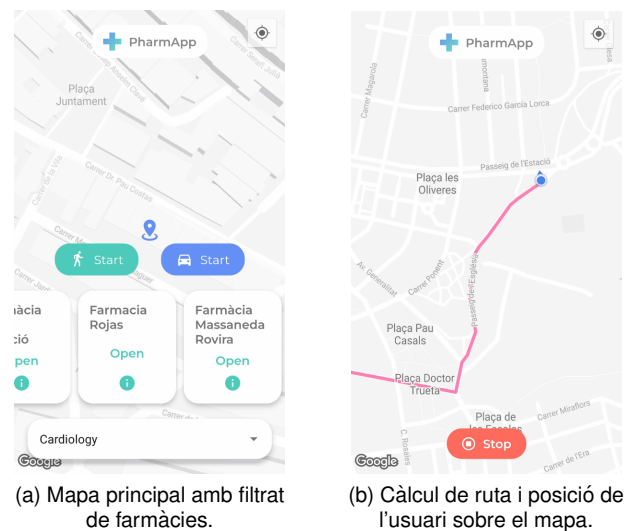


Fig. 6: Captures de l'aplicació mòbil.

6.4 Gestió de les peticions — Backend

Per poder gestionar tota la informació generada per l'aplicació web, a més de poder donar resposta a la demanda de dades que es produeix des d'ambdues aplicacions (web i mòbil) és necessari crear un sistema eficient que sigui capaç de retornar la resposta esperada en el mínim temps possible.

Així doncs, el *backend* té, principalment, dos objectius molt clars: el primer és gestionar i donar resposta a totes les peticions, tant les rebudes de l'aplicació web com les rebudes de l'aplicació mòbil. El segon objectiu és comunicar-se

amb la base de dades per tal de realitzar lectures i escriptures d'informació, ja que és l'únic component del sistema que té connexió directa amb aquesta.

Per poder aconseguir el primer objectiu de poder rebre, gestionar i donar resposta a les dues aplicacions, s'ha construït una *Application Programming Interface* del tipus *Representational State Transfer*, també coneguda com a API REST (o RESTful) [15]. Per a la seva construcció i l'òptima gestió de les peticions en funció del *Uniform Resource Identifier* (URI) requerit, s'ha utilitzat el *framework* Express, el qual està inclòs dins del conjunt de programari MEAN [16], juntament amb MongoDB, Angular.js i Node.js. Gràcies a la utilització de Node.js com a entorn d'execució de JavaScript al *backend*, la integració amb Express és completa. També, i com que Node.js està especialment dissenyat per gestionar events asíncrons, ofereix una alta velocitat de resposta als clients, a més de permetre crear aplicacions altament escalables horitzontalment.

Finalment, per aconseguir el segon objectiu de poder realitzar les lectures, insercions i altres modificacions a la base de dades MongoDB, s'han definit models de dades amb Node.js. Concretament, s'ha utilitzat Mongoose, que és una eina de modelatge de dades (*Object Data Modeling*, ODM) que permet el tractament d'aquestes i la comunicació entre el *backend* i MongoDB durant les operacions de lectura i escriptura d'informació.

6.4.1 JSON Web Tokens (JWT)

Àmpliament utilitzats dins de l'àmbit del desenvolupament web i definits a l'estàndard RFC 7519 [17], els *JSON Web Tokens* (coneguts també com a JWT) són un mecanisme de representació de dades que permet compactar-les i, així, optimitzar el procés d'intercanvi d'informació entre les parts implicades durant la comunicació HTTP.

Més concretament, i tal com s'exposa a la seva definició, un *token* del tipus JWT és una cadena de caràcters alfanumèrics que representa un conjunt de dades incloses dins d'un objecte del tipus *JavaScript Object Notation* (JSON), el qual pot estar codificat seguint l'estàndard *JSON Web Signature* (JWS) [18] o xifrat utilitzant tècniques de *JSON Web Encryption* (JWE) [19]. Així, la seva finalitat és la representació d'informació en entorns on l'espai és reduït o està limitat, com poden ser les capçaleres d'autenticació dels missatges HTTP o els paràmetres inclosos als URI en el moment de l'enviament de les dades cap al servidor.

Una característica diferencial que aporten aquests *tokens* respecte d'altres o, simplement, de la transferència habitual de dades contingudes a les capçaleres dels missatges intercanviats mitjançant el protocol HTTP, és que la secció que conté les dades a transportar (anomenada *payload*) està signada amb una clau secreta de l'entitat que genera el *token*, que en aquest cas és el *backend*. Per tant, qualsevol modificació en el contingut del JWT intercanviat entre el servidor i el client, per mínima que sigui, invalidarà la signatura i el *token* estarà revelant la seva invalidesa. Com a resultat, la seva petició no es processarà i el sistema, en general, serà més segur.

A continuació, es mostra una imatge (Figura 7) on es pot observar tant el *token* codificat com la informació en format JSON que conté:

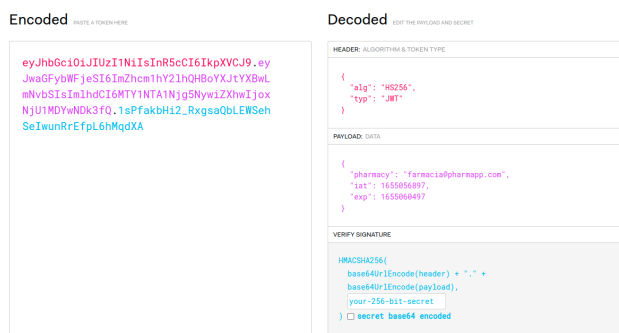


Fig. 7: JWT codificat (esquerra) i descodificat (dreta).

L'estructura del JWT codificat consta de tres parts separades per un punt i, cadascuna d'elles, es codifica per separat utilitzant *Base64URL* [20]. La primera és la capçalera o *header* del *token*, i aquesta s'encarrega d'emmagatzemar el tipus de *token* (especificat amb la clau "typ" al JSON descodificat) i l'algorisme utilitzat per generar la signatura, el qual s'especifica amb la clau "alg".

La segona part d'aquest *token* és el cos, càrrega o *payload* i conté la informació. Cadascuna de les dades que es transporten al *payload* reben el nom de *claims*, i són declaracions sobre una entitat, que generalment fan referència a un usuari. Com es pot observar, el *token* mostrat a la Figura 7 conté tres *claims*: la primera, "pharmacy", conté l'adreça de correu electrònic de la farmàcia que ha rebut aquest JWT en el moment de registrar-se o iniciar sessió. També existeixen dues *claims* addicionals: "iat" i "exp". La primera significa "Issued At" i, com el seu nom indica conté, en format Epoch, la data a la qual es va emetre el JWT. La segona *claim* significa "Expires" i indica la data, també en format Epoch, a partir de la qual aquest *token* deixarà de ser vàlid.

Finalment, la tercera part del JWT conté la signatura d'aquest, i és a partir de la qual es pot comprovar si el contingut és original o, per contra, si s'ha modificat.

En referència a la utilització dels JWT dins de l'entorn de PharmApp, cal destacar que aquests *tokens* tenen tres funcions principals: identificació de les farmàcies per part del *backend* mentre utilitzen el servei web, limitació de les accions que aquestes poden dur a terme i, finalment, simulació d'un estat utilitzant un protocol sense estat (com és HTTP).

Primerament, la identificació de les farmàcies és possible gràcies al fet que, en el moment de generar un nou *token*, ja sigui durant el registre d'una nova farmàcia o en el moment en què inicia sessió a l'aplicació web, s'inclou l'adreça de correu electrònic de la farmàcia dins del *payload* d'aquest *token* que s'està generant. Així, en futures peticions al *backend*, aquest podrà accedir al contingut del JWT rebut a la petició i podrà identificar unívocament a cadascuna de les farmàcies que estiguin registrades a la plataforma de PharmApp. En conseqüència, aquesta identificació permet també la limitació d'accés a recursos, com poden ser les diferents rutes que ofereix l'API REST del *backend*.

Finalment, la simulació d'un estat utilitzant un protocol sense estat com HTTP és possible gràcies al fet que, el JWT assignat a cadascuna de les farmàcies, s'envia com una *cookie* al client en el moment d'iniciar sessió o registrar-se. Per tant, el client emmagatzemarà aquesta *cookie* al seu navegador durant tota la sessió i podrà navegar entre les diferents pàgines del servei web de PharmApp sense necessitat d'haver d'iniciar sessió amb cada moviment entre pàgines.

En el moment en què el client finalitzi la seva sessió, aquesta *cookie* s'esborrarà del seu navegador i s'invalidarà al servidor, el qual generarà un *token* nou quan la farmàcia torni a iniciar sessió.

7 GOOGLE CLOUD PLATFORM

Com s'ha descrit als apartats anteriors relatius a les aplicacions web i mòbil, s'utilitzen tres serveis que ofereix Google: Maps Directions API, Maps JavaScript API i Maps Geocoding API. Totes elles formen part de Google Cloud Platform.

En aquest àmbit dels serveis al *cloud*, Google actua com un proveïdor de recursos de computació i d'infraestructures de Tecnologies de la Informació (TI) al núvol, oferint així la possibilitat d'utilitzar els seus serveis per al desenvolupament de nou programari. Aquest fet facilita molt la creació de noves aplicacions que, com és el cas de PharmApp, si s'hagués de desenvolupar cadascun dels components que formen el sistema, no seria possible a causa de la complexitat i el temps requerit, a més de la immensa quantitat de dades que posseeix Google i que posa a disposició dels clients que facin servir els seus serveis.

Per tal de poder utilitzar aquestes infraestructures i oferir les aplicacions de PharmApp als clients i farmàcies, és necessari obtenir una clau, o *API Key*, que s'enviarà a totes les peticions que es realitzin per part de les aplicacions. Així, Google pot identificar quins serveis pot fer servir cada client i quin ha estat el nombre de vegades que s'ha usat.

Cal destacar que, encara que per a la realització d'aquest projecte s'ha disposat d'un pla gratuït durant tot el desenvolupament, la utilització d'aquests serveis de Google comporta un cost econòmic. Així, el preu dels serveis utilitzats és de 4,66 € per cada 1000 peticions a cadascun dels serveis. Per tant, és necessari registrar una targeta bancària per assegurar que no se'n farà un mal ús dels sistemes o, en cas de superar el nombre de peticions màximes, el desenvolupador (o l'entitat que es faci càrrec del cost) haurà d'abonar a Google la quantitat establerta segons la utilització dels serveis.

8 ALLOTJAMENT DEL SERVEI

Per allotjar tant el servei de *frontend* que utilitzaran les farmàcies per gestionar la seva informació com el *backend* que dona resposta a les peticions de les dues aplicacions (mòbil i web) i la base de dades de PharmApp, s'ha decidit utilitzar un servei de *Virtual Private Server* [21] (també conegut com a *VPS*) al *cloud* que ofereix l'empresa OVH. Aquesta decisió es basa en el concepte inicial de PharmApp, que és crear un servei que sigui funcional i estigui operatiu per a utilitzar en el món real. Així, OVH ofereix serveis de *VPS* que, encara que són de pagament, el cost econòmic que suposen és molt reduït i, a més, ofereix la garantia que el funcionament serà l'esperat per a un projecte com PharmApp durant la seva fase inicial de desplegament.

Concretament, d'entre els 4 diferents plans que ofereix OVH sobre serveis *VPS*, s'ha seleccionat el pla "Value", que és el nom que rep el servidor privat virtual més econòmic, i té un cost de \$5.52/mes. Aquest servidor té les següents característiques:

- Processador: 1 vCore.
- Memòria RAM: 2 GB.
- Unitat d'emmagatzematge: 40 GB SSD NVMe.
- Amplada de banda: 250 Mb/s i trànsit il·limitat.
- Adreçament IP: 1 adreça IPv4 estàtica i 1 rang d'adreces /128 del tipus IPv6.
- Sistema operatiu: es pot escollir entre CentOS, Debian, Fedora, Ubuntu, AlmaLinux i RockyLinux.
- *Service Level Agreement*: assegura que el temps durant el qual el *VPS* estarà operatiu és, com a mínim, del 99,9%.
- Altres serveis inclosos: Docker, protecció contra atacs DDoS, KVM, accés com a superusuari (*root*) i còpies de seguretat automatitzades.
- Altres serveis opcionals (inclosos al preu): CPANEL, Plesk o Windows Server.

Com es pot observar, tot i tenir un preu molt reduït, les especificacions del servei permeten oferir el servei de PharmApp públicament i sense problemes de rendiment durant la fase inicial del desplegament.

Una vegada iniciat i configurat el servidor, el servei de *frontend* que permet gestionar les farmàcies està disponible a través de l'adreça IP 37.187.198.170 i el port 8080.

9 CONTAINERITZACIÓ DELS COMPONENTS

Per aconseguir l'objectiu de poder desplegar els diferents components en un entorn real i oferir els serveis públicament, s'ha considerat que la containerització dels elements que formen el sistema és l'opció adequada, ja que els avantatges que aporta la utilització de contenidors per al desplegament d'aplicacions són múltiples. Per exemple, l'eficiència en el consum de recursos respecte als mecanismes de virtualització tradicional, la capacitat d'escalar horitzontalment els sistemes respecte d'un esquema escalable verticalment (el qual consumeix molts recursos econòmics, materials i energètics) i, a més, permet reiniciar els serveis de manera molt ràpida.

Respecte de l'eficiència, aquesta és deguda principalment a què tots els contenidors utilitzen el sistema operatiu de la màquina que els conté. Per tant, no es genera sobrecàrrega en aquest sentit i, a més, els contenidors només fan ús de les llibreries necessàries per al funcionament del servei que ofereixen, reduint així l'espai d'emmagatzematge necessari.

L'escalabilitat horitzontal consisteix a desplegar múltiples instàncies del mateix servei per distribuir la càrrega de treball, la qual cosa permet abordar els pics de demanda de manera dinàmica i eficient. També, en els moments en què la demanda és menor, es poden eliminar instàncies fàcilment, reduint així el cost econòmic, el consum de recursos computacionals i el consum energètic, en contraposició a l'esquema tradicional de posseir molts recursos que treballen a ple rendiment constantment i independentment de la demanda. També s'elimina el punt únic de fallada, ja que es pot oferir més d'una instància a la vegada per cada servei, que contrasta amb la idea de tenir un servidor central que gestiona totes les peticions.

Altrament, per als sistemes que ofereixen els serveis mitjançant contenidors, la facilitat i alta velocitat de recu-

peració davant caigudes és un factor determinant, atès que iniciar de nou els serveis és un procés ràpid i eficient.

9.1 Creació de les imatges

Per poder instanciar contenidors Docker que ofereixin els serveis desenvolupats s'han de generar imatges d'aquests. Per dur a terme aquesta tasca és necessari crear un Dockerfile, el qual és un fitxer de text sense format que conté les instruccions necessàries per crear la nova imatge del servei, executant-se aquestes de forma seqüencial. Així, estableixen quines accions s'han de dur a terme per construir una imatge del servei que, posteriorment, serà instanciat en un o més contenidors.

Una vegada generat el fitxer Dockerfile d'un servei concret, la imatge s'ha de generar amb la comanda "docker build" (veure Apèndix A.1 per accedir a les comandes completes).

9.2 Enviament de les imatges al VPS

Pel fet que el desenvolupament de les aplicacions es realitza localment i no al VPS, és necessari transferir les imatges dels contenidors al servidor per poder oferir el servei públicament. Per dur a terme aquesta tasca s'ha utilitzat l'eina *OpenSSH Secure File Copy*, coneguda com SCP. Aquesta permet transferir dades, mitjançant el protocol *Secure Shell* (SSH), entre diferents computadors connectats a la xarxa.

Per tant, després de generar les imatges dels contenidors, com s'ha descrit a l'apartat anterior, és necessari generar un fitxer que es pugui transferir al VPS. La comanda "docker save" emmagatzema una o diverses imatges Docker a un fitxer del tipus Tar. Després de generar aquest fitxer es procedeix a transferir-lo al VPS utilitzant SCP.

Una vegada rebuts els fitxers al VPS, s'han de carregar les imatges dels contenidors Docker. La comanda "docker load" carrega una imatge a partir d'un fitxer Tar.

9.3 Execució dels contenidors al VPS

Per gestionar l'execució de tots els contenidors necessaris per oferir el servei de PharmApp als usuaris i a les farmàcies s'utilitza Docker Compose. Aquesta eina que ofereix Docker és un orquestrador de contenidors que permet definir i controlar com s'executaran els diferents contenidors dins d'un entorn local. Així, permet també crear una xarxa virtual per permetre la comunicació, de manera ràpida i eficient, entre els contenidors que formen part d'aquesta. Totes aquestes configuracions, tant les relacionades amb els contenidors com les que fan referència a la xarxa virtual, es defineixen a un fitxer escrit en format YAML.

Finalment, una vegada carregades les imatges dels contenidors al servei Docker instal·lat al VPS, aquests es poden gestionar de manera eficient i senzilla a través de la comanda "docker compose".

10 SEGURETAT DEL SISTEMA

Per crear un sistema segur i evitar possibles fugites d'informació s'han utilitzat diverses tècniques que prevenen dife-

rents tipus d'atacs informàtics.

El primer mecanisme, el qual ja s'ha descrit amb més profunditat a l'apartat 6.4.1, és la utilització dels JWT. Amb aquests, s'aconsegueix identificar unívocament a les farmàcies que realitzen peticions al servei de *backend* per limitar les accions que aquestes poden dur a terme al sistema com, per exemple, modificar dades de les diferents farmàcies registrades al sistema.

Un altre dels mecanismes emprats per prevenir possibles riscos és la utilització de *password salting*. Amb aquesta tècnica es poden prevenir els atacs de diccionari precomputat, els atacs de *Rainbow Tables* i, també, es fa pràcticament impossible completar atacs de força bruta, ja que s'augmenta notablement el nombre de càlculs necessaris per trobar el *hash* generat en concatenar la *salt* i la contrasenya de l'usuari en el moment de xifrar-la. Per generar aquesta *salt* i el xifratge de la contrasenya s'ha utilitzat l'algorisme *Bcrypt* [22].

Finalment, per evitar atacs d'injecció de codi a l'aplicació web es realitza un procés de desinfecció o *sanitizing* de tots els camps d'entrada de dades, utilitzant l'eina *Validator.js* [23]. Així, s'eliminen alguns dels caràcters que s'introdueixen i que són considerats com a maliciosos. Per tant, el possible codi introduït no s'arribarà a executar.

11 CONCLUSIONS I RESULTATS

Amb la finalització d'aquest informe es tanca l'etapa de desenvolupament del projecte, i es pot afirmar que s'han assolit tots els objectius plantejats a l'inici d'aquest. Així, s'ha creat amb èxit una aplicació mòbil multiplataforma per als usuaris, una aplicació web per gestionar la informació de les farmàcies i un servei de *backend* que permet a les dues aplicacions comunicar-se i modificar la informació continguda a la base de dades. Per tant, el servei creat facilita als usuaris de l'aplicació mòbil la tasca de trobar farmàcies d'interès que puguin atendre les seves necessitats de la millor manera possible. A més, l'aplicació web, permet a les farmàcies afegir la seva informació a la plataforma de manera senzilla i ràpida i, també, ajuda a descobrir i arribar directament a un públic més concret, que és un dels motius principals pel qual aquestes han pres la decisió d'especialitzar-se cada vegada més.

Adicionalment, tal com s'ha comentat a l'apartat 9, la containerització dels components aporta uns grans beneficis i deixa preparats els diferents components del sistema per poder realitzar les ampliacions futures i permetre escalar el projecte si, en algun moment, augmenta la demanda per part dels usuaris.

Finalment, cal destacar també que una planificació adequada de les tasques i recursos, a més de les eines indicades per realitzar el seguiment d'aquesta, són essencials en el desenvolupament de qualsevol projecte però, especialment, per als de l'àmbit tecnològic. Així, la planificació creada a l'inici d'aquest treball i les seves posteriors modificacions, han permès desenvolupar tots els components que formen el sistema dins del termini establert.

En referència al treball futur, cal destacar que una de les possibles millores que es podria implementar i que permetria exprémer al màxim els avantatges que la replicació i gestió automàtica de contenidors aporta seria la distribució dels contenidors a diferents servidors, creant així un clúster

que evitaria tenir un únic punt de fallada com és, en aquest moment, el VPS contractat. Però, per aconseguir-ho, seria necessari disposar de diverses màquines on replicar i distribuir la càrrega de treball. Per tant, com que no es disposa d'aquesta infraestructura, s'ha decidit inicialment oferir el servei de PharmApp mitjançant un dels servidors privats virtuals que ofereix OVH i que, com s'ha exposat, compleix amb els requeriments necessaris per dur a terme el primer desplegament de la plataforma.

Una altra millora seria la compilació de l'aplicació per al sistema operatiu iOS ja que, en no disposar d'un computador de la família Mac, no s'ha pogut instal·lar l'aplicació a telèfons intel·ligents iPhone.

AGRAÏMENTS

Voldria agrair, en primer lloc, a la meva família i amics, ja que sense el seu suport durant aquesta etapa no hagués estat possible continuar en moltes ocasions. També a la tutora d'aquest treball, Marta Prim, pel seguiment realitzat durant tot el projecte. Finalment, voldria agrair a totes les professores, professors i altres persones que, durant el transcurs del Grau, han dedicat el seu temps a formar-nos i a contribuir en la recerca d'un futur millor.

REFERÈNCIES

- [1] Llei 14/1986, de 25 d'abril, General de Sanitat. Boletín Oficial del Estado. Espanya. 25 d'abril de 1986. [En línia]. Disponible a: <https://www.boe.es/buscar/pdf/1986/BOE-A-1986-10499-consolidado.pdf>
- [2] "Healthcare and medicines", Google. [En línia]. Disponible a: <https://support.google.com/adspolicy/answer/176031?hl=en>
- [3] "Leading US Map and Navigation Smartphone Apps", eMarketer, 15 de març de 2020. [En línia]. Disponible a: <https://www.emarketer.com/chart/234831/leading-us-map-navigation-smartphone-apps-ranked-by-monthly-unique-users-aug-2019>
- [4] E. Russell, "9 things to know about Google's maps data: Beyond the Map", Google Cloud, 30 de setembre de 2019. [En línia]. Disponible a: <https://cloud.google.com/blog/products/maps-platform/9-things-know-about-googles-maps-data-beyond-map>
- [5] "Rainbow Table Attack", Beyond Identity. [En línia]. Disponible a: <https://www.beyondidentity.com/glossary/rainbow-table-attack>
- [6] "What Is a Framework?", Codecademy News, 23 de setembre de 2021. [En línia]. Disponible a: <https://www.codecademy.com/resources/blog/what-is-a-framework/>.
- [7] "Single-File Components", Vue.js. [En línia]. Disponible a: <https://vuejs.org/guide/introduction.html#single-file-components>
- [8] L. Sujay, "Most used web frameworks among developers worldwide, as of 2021", Statista. 23 de febrer de 2022. [En línia]. Disponible a: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>
- [9] "Geocoding API", Google. 13 de maig de 2022. [En línia]. Disponible a: <https://developers.google.com/maps/documentation/geocoding/overview>
- [10] "Place Details - Maps JavaScript API", Google. 8 de juny de 2022. [En línia]. Disponible a: <https://developers.google.com/maps/documentation/javascript/examples/place-details>
- [11] "Google Maps Platform Coverage Details", Google. 13 de maig de 2022. [En línia]. Disponible a: <https://developers.google.com/maps/coverage>
- [12] L. S. Vailshery, "Cross-platform mobile frameworks used by global developers 2021", Statista, 21 de febrer de 2022. [En línia]. Disponible a: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- [13] B. Van Damme, "React Native Maps Directions", Github. 30 d'abril de 2022. [En línia]. Disponible a: <https://github.com/bramus/react-native-maps-directions>
- [14] "Directions API overview", Google. 13 de maig de 2022. [En línia]. Disponible a: <https://developers.google.com/maps/documentation/directions/overview>
- [15] "What is a REST API?", IBM Cloud Education. 6 d'abril de 2021. [En línia]. Disponible a: <https://www.ibm.com/cloud/learn/rest-apis>
- [16] "The MEAN Stack", Github. 2 de febrer de 2017. [En línia]. Disponible a: <https://web.archive.org/web/20190306144532/http://mean.io/>
- [17] M. Jones, J. Bradley i N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519. Maig de 2015. [En línia]. Disponible a: <https://www.rfc-editor.org/info/rfc7519>.
- [18] M. Jones, J. Bradley i N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515. Maig de 2015. [En línia]. Disponible a: <https://www.rfc-editor.org/info/rfc7515>
- [19] M. Jones i J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516. Maig de 2015. [En línia]. Disponible a: <https://www.rfc-editor.org/info/rfc7516>
- [20] "Base64URL", Base64 Guru. [En línia]. Disponible a: <https://base64.guru/standards/base64url>
- [21] "Virtual Private Server", IBM Cloud Education. [En línia]. Disponible a: <https://www.ibm.com/cloud/learn/vps>
- [22] N. Provos i D. Mazieres, "Bcrypt Algorithm", Usenix. 28 d'abril de 1999. [En línia]. Disponible a: <https://www.usenix.org/legacy/events/usenix99/provos/provos.html/node1.html>
- [23] C. O'Hara i A. Nandaa, "validator.js", npm. 22 d'abril de 2022. [En línia]. Disponible a: <https://www.npmjs.com/package/validator>

APÈNDIX

A.1 Creació dels contenidors

En aquest apartat s'especifiquen les comandes utilitzades per a la creació de les imatges Docker que s'utilitzaran per instanciar nous contenidors i executar-los al VPS contractat a OVH. Així és com s'ofereix el servei de PharmApp públicament.

A.1.1 Creació de les imatges

Com el desenvolupament es realitza dins de l'entorn local, la creació de les imatges dels contenidors Docker també es realitza dins d'aquest mateix entorn, mitjançant els fitxers Dockerfile. A continuació, es mostra el contingut del Dockerfile que permet crear la imatge del contenidor que allotjarà l'aplicació web:

```
# build stage
FROM node:lts-alpine as build-stage
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# production stage
FROM nginx:stable-alpine
  as production-stage
COPY --from=build-stage /app/dist
  /usr/share/nginx/html
RUN rm /etc/nginx/conf.d/default.conf
COPY nginx/nginx.conf
  /etc/nginx/conf.d
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

El Dockerfile que permet generar la imatge del *backend* és el següent:

```
FROM node:16
# Crea el directori de l'aplicació
WORKDIR /usr/src/app
# Copia les dependències del projecte
COPY package*.json ./
# Instal·la les dependències per la
# fase de producció
RUN npm ci --only=production
# Copia a la imatge l'executable
COPY . .
EXPOSE 8080
CMD [ "node", "server.js" ]
```

Després de definir les instruccions dels fitxers Dockerfile, es pot procedir a generar les imatges. Per generar la imatge de l'aplicació web, des de l'arrel del directori que conté els fitxers del servei, cal executar la següent comanda: “docker build -t pharmapp-web .”.

També, per generar la imatge del *backend*, cal situar-se a l'arrel del directori que conté els fitxers del servei. Seguidament s'ha de procedir a executar la següent comanda: “docker build -t pharmapp-backend .”.

Finalment, per poder enviar aquestes imatges al VPS de OVH, cal generar un fitxer Tar. Per fer-ho, cal executar les següents comandes:

```
“docker save pharmapp-web >
  pharmapp-web.tar”
“docker save pharmapp-backend >
  pharmapp-backend.tar”
```

A.1.2 Enviament de les imatges al VPS

Després de generar els fitxers amb les imatges s'ha de procedir a enviar-les al servidor VPS. Per dur a terme aquesta tasca s'utilitzen les següents comandes;

```
“scp pharmapp-web.tar
  debian@37.187.198.170:/home/debian”
“scp pharmapp-backend.tar
  debian@37.187.198.170:/home/debian”
```

Per poder utilitzar aquests fitxers rebuts al VPS com imatges que permetin instanciar contenidors, és necessari carregar-les a Docker. Per dur a terme aquesta tasca, és necessari executar les següents comandes:

```
“docker load < pharmapp-web.tar”
“docker load < pharmapp-backend.tar”
```

A.1.3 Execució dels contenidors al VPS

Per poder executar tots els contenidors necessaris per oferir el servei de PharmApp i, a més, fer-ho de forma eficient, s'ha creat el fitxer docker-compose.yml, el qual conté les següents instruccions:

```
version: '3.1'
services:
  mongo:
    image: mongo
    restart: always
    container_name: pharmapp-mongodb
    ports:
      - 27017:27017
    volumes:
      - $HOME/PharmApp/mongodb_data:/data/db
    command: [--auth]
    networks:
      - local

  backend:
    image: pharmapp-backend
    restart: always
    container_name: pharmapp-backend
    ports:
      - 3000:3000
    depends_on:
      - mongo
      - web
    networks:
      - local

  web:
    image: pharmapp-web
    restart: always
    container_name: pharmapp-web
    ports:
      - 8080:80
    depends_on:
      - mongo
    networks:
      - local

networks:
  local:
    driver: bridge
```

Fig. 8: Contingut del fitxer docker-compose.yml

La comanda per iniciar tots els serveis definits dins d'aquest fitxer és “docker compose up -d”. La comanda per finalitzar l'execució ordenada d'aquests contenidors és “docker compose down”.

A.2 Prototips de les aplicacions

A.2.1 Prototips de l'aplicació mòbil

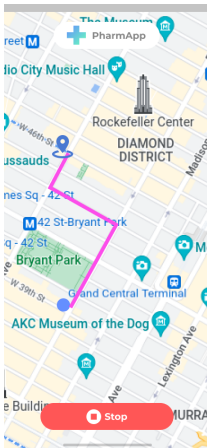


(a) Mapa principal.

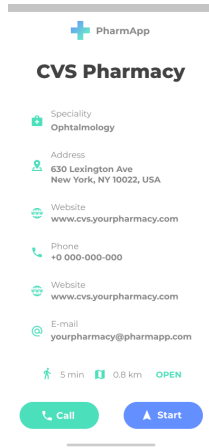


(b) Mapa principal amb filtrat de farmàcies.

Fig. 9: Prototips de l'aplicació mòbil.



(a) Càlcul de ruta i posició de l'usuari sobre el mapa.



(b) Informació detallada d'una farmàcia.

Fig. 10: Prototips de l'aplicació mòbil.

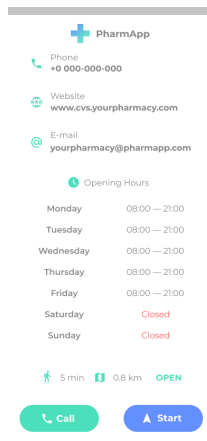


Fig. 11: Informació detallada d'una farmàcia.

A.2.2 Prototips de l'aplicació web

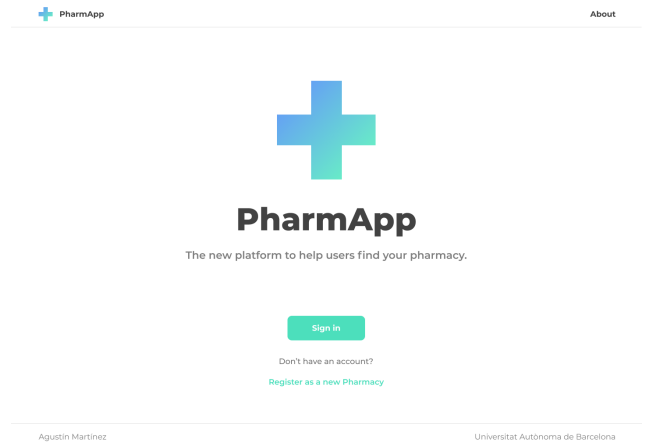


Fig. 12: Pantalla inicial.

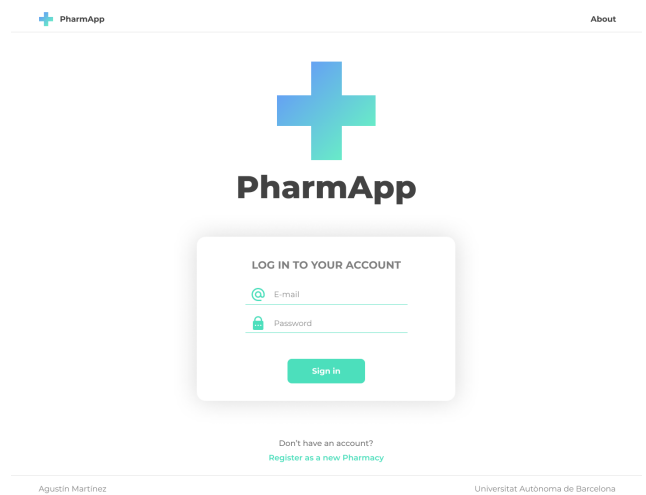


Fig. 13: Formulari d'inici de sessió.

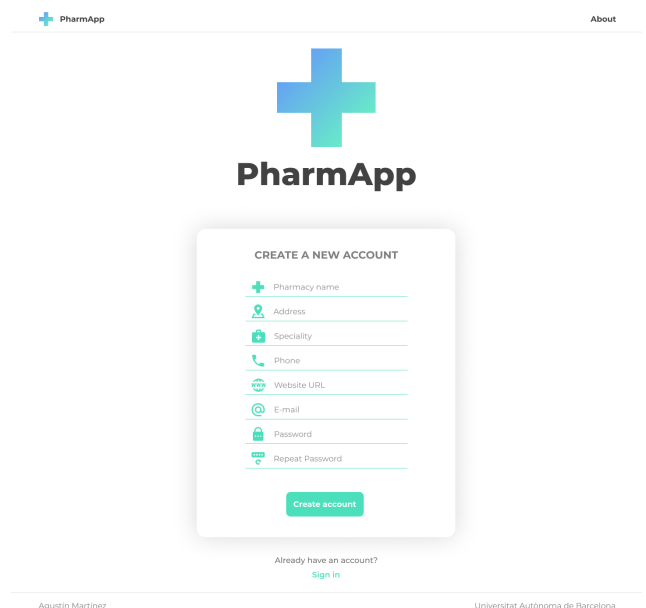


Fig. 14: Formulari de registre.

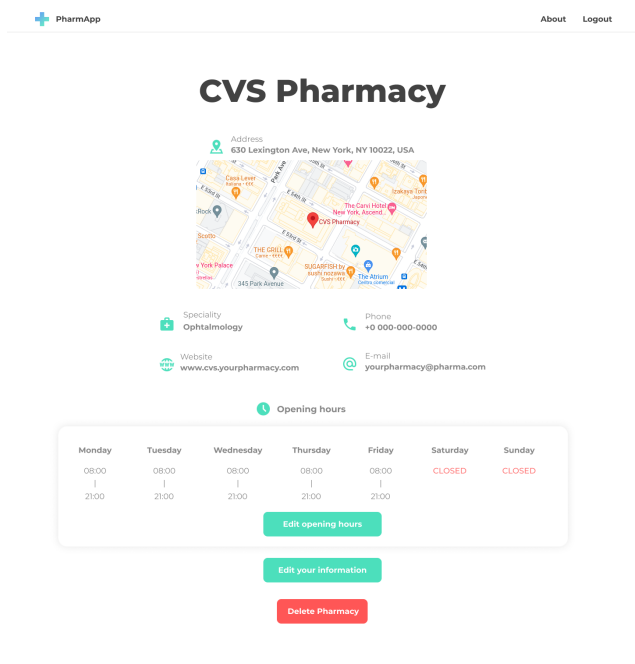


Fig. 15: Pantalla de gestió de la farmàcia.

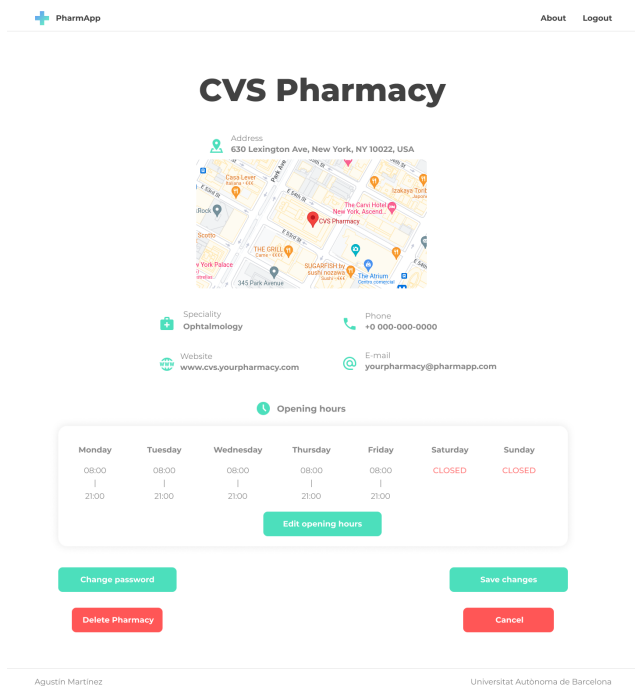


Fig. 16: Pantalla d'edició d'informació de la farmàcia.

A.3 Casos d'ús

A.3.1 Aplicació web — Gestió de la farmàcia

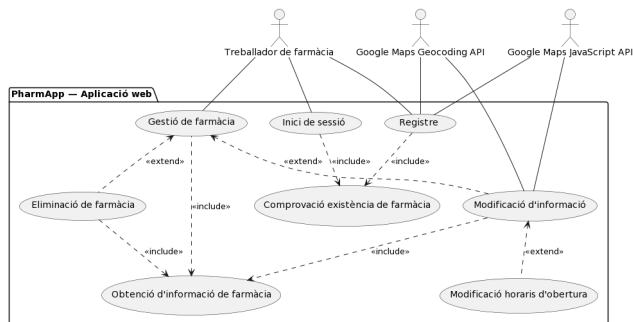


Fig. 17: Diagrama de cas d'ús. Gestió de la farmàcia per part d'un treballador d'aquesta.

A.3.2 Aplicació mòbil — Llistat i càlcul de ruta per part de l'usuari

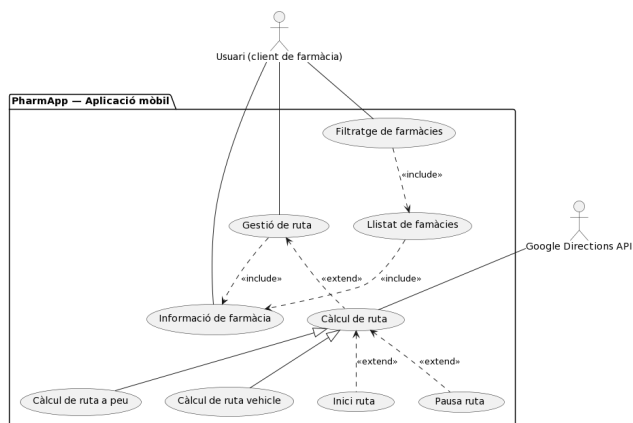


Fig. 18: Diagrama de cas d'ús. Filtratge de farmàcies i càlcul de ruta per part d'un client.