Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

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

Masip Quintana, Robert; Valveny Llobet, Ernest, dir. Aplicació mòbil basada en una calculadora per obtenir la petjada de carboni dels aliments. 2023. (958 Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/272816

under the terms of the BY-NC-ND license

Aplicació mòbil basada en una calculadora per obtenir la petjada de carboni dels aliments

Robert Masip Quintana

08/02/2023

Resum– En aquest article, s'exposa el desenvolupament dut a terme per tal de realitzar una aplicació per a dispositius mòbils. Aquesta aplicació té els objectius de facilitar als usuaris, un seguiment de quin és l'impacte que crea la petjada de carboni dels aliments que consumeixen en el seu dia a dia. L'aplicatiu que s'ha dut a terme durant el projecte té la intenció de què la introducció i representació dels valors sigui senzilla i amigable, i que el procés d'aprenentatge esdevingui el menor possible, també incorpora ajuts per a qui ho necessiti. Els usuaris poden calcular la petjada de carboni introduint un aliment desitjat i la quantitat, l'aplicatiu mostrarà diferents valors sobre l'impacte ambiental d'aquell aliment, tant menteix l'aplicació permet registrar usuaris o tenir un compte personal per visualitzar les darreres cerques. L'aplicació ha estat implementada emprant Net Core i Xamarin. Per a la gestió del projecte s'ha seguit el model en cascada.

Paraules clau– Calculadora, software, aplicació mobil, petjada de carboni, API, Android

Abstract– In this article, the development carried out in order to make an application for mobile devices is presented. This application aims to help users monitor the impact of the carbon footprint of the food they consume on a daily basis. The application that has been carried out during the project aims to make the introduction and representation of the values simple and friendly, and that the learning process becomes as short as possible, it also incorporates aids for those who need it. Users can calculate the carbon footprint by entering a desired food and the amount, the application will show different values about the environmental impact of that food, so much lies the application allows users to register or have a personal account to view the last searches . The application has been implemented using Net Core and Xamarin. For project management, the waterfall model has been followed.

Keywords– Calculator, software, mobile application, carbon footprint, API, Android

1 Introducció - Context del treball

Aquest treball s'emmarca en la proposta de la realització d'una calculadora per calcular la petjada de carboni dels aliments. La idea principal ha sigut proposada per FAADA o "Fundació per a l'Assessorament i Acció en Defensa dels Animals"que és una fundació privada, independent i sense ànim de lucre fundada l'any 2004. Els seus principals objectius són canviar els hàbits de la societat promovent el respecte pels animals en l'àmbit social, legal, educatiu i aprendre a conviure respectuosament amb totes les espècies animals, sense abusar d'elles, i relacionant-se des del respecte, fomentant una relació d'igual a igual, sense cap mena de discriminació. Aquests propòsits es materialitzen fent accions en els àmbits socials, educatius i legals.

El projecte te un fort component per conscienciar a la població de l'ús que se'n fa del menjar i fer una crítica constructiva del que és la indústria alimentària, que actualment és completament insostenible, genera un impacte sobre el medi ambient molt significant sent juntament amb la pesca (després de les energies fòssils) el sector que té

• E-mail de contacte: 1459441@uab.cat
• Mencció realitzada: Enginyeria del Software
• Treball tutoritzat per: Ernest Valveny Llobet(Department of Computer Science)
• Curs 2022/23

un major impacte. Afecta el funcionament i a la regulació de la Terra pel fet que altera processos biogeofísics, com són: el canvi climàtic, el canvi en l'ús del sòl, l'ús d'aigua dolça, els cicles del nitrogen o del fòsfor i la pèrdua de la biodiversitat. Aquesta alteració pot causar canvis irreversibles, permanents i potencialment catastròfics en el sistema terrestre.

L'objectiu principal del projecte és crear una aplicació mòbil basada en una calculadora que principalment calcularà la petjada de carboni(CO_2) dels aliments en tota la cadena des de la producció fins que arriba a les nostres mans fent èmfasi en calcular l'impacte en la producció, el transport i l'empaquetament. L'aplicació també oferirà altres característiques com permetre als usuaris registrats consultar l'historial dels aliments consumits, fer equivalències de la reducció de la petjada de carboni segons els aliments que consumeixen. D'aquesta manera podrem mostrar a la ciutadania l'impacte que tenen els aliments consumits.

Per contextualitzar sobre el que és la petjada de carboni, és el nostre impacte sobre el planeta expressat com la quantitat de gasos d'efecte hivernacle que produïm. L'augment dels nivells d'aquests gasos a l'atmosfera està provocant un canvi climàtic global. La indústria alimentària n'és un factor important. Dels diversos gasos que hi contribueixen, el de major impacte és sense dubte el diòxid de carboni (CO_2), representa el 55% del total. La petjada de carboni mitjana d'un ciutadà europeu és de 7 tones de CO_2 a l'any.

Per tant la realització d'aquest projecte consisteix a crear una aplicació que permeti el càlcul de l'impacte ambiental de diferents combinacions d'aliments, alhora que serveix de conscienciació de l'impacte que tenen els aliments que consumim per tal d'evitar un conjunt d'emissions de gasos d'efecte hivernacle innecessaris i insostenibles.

2 Objectius

El principal objectiu del projecte és crear una aplicació mòbil per a l'associació FAADA, aquesta aplicació ha de servir per calcular la petjada de carboni de certs aliments introduïts per l'usuari de l'aplicació. A més els usuaris podran consultar informació rellevant sobre altres característics de l'impacte que té la producció d'aquell aliment (consum aigua, fòsfor, metres quadrats de terreny, etc). L'usuari també tindrà la possibilitat de registrar-se a l'aplicació així podent tenir un registre dels aliments consultats amb anterioritat i tenint un registre de l'impacte que té els aliments que consumeix.

Complemntariament a l'objectiu principal també tenim objectius secundaris:

- Dissenyar les diferents interfícies que conformaran la part visual de l'aplicació. Es podrà considerar dur a terme millores estètiques en etapes avançades del projecte).
- Creació i modelat d'una base de dades per guardar la informació (emissions CO_2, terres de cultiu fetes servir, aigua feta ús, etc.) sobre l'impacte ambiental dels aliments.
- Desplegar un back-end clar i entenedor per realitzar el manteniment de les dades de l'aplicació.

3 Planificació

Per poder desenvolupar aquest projecte, es va determinar una llista d'activitats que es van agrupar en diverses fases. Aquestes activitats es van planificar rigorosament en funció de la seva dificultat:

- **Captura i documentació de requisits:** Primera fase del projecte on s'ha fet una anàlisi dels requeriments, s'ha avaluat l'estat de l'art, s'ha creat un prototip i s'han realitzat diversos diagrames de suport.
- **Disseny del sistema:** Segona fase del projecte on s'ha creat la base de dades, s'ha realitzat la connexió amb l'aplicació i s'ha implementat el codi per al back-end per a satisfer els requisits.
- **Disseny App:** Tercera fase del projecte on s'ha realitzat pa part del codi que correspon al front-end.
- **Execució:** Quarta fase del projecte, s'ha provat la comunicació del front-end amb el back-end a través de la base de dades i el servidor.
- **Creació part dinamica del Front-end:** Aquesta part s'ha destinat a la creació i proves de la part dinàmica de l'aplicació, que la comunicació del client-servidor es realitzés correctament.
- **Tancament del projecte:** S'ha realitzat la memòria, la documentació necessària i la presentació del projecte posant fi al treball.

4 Metodologia

Per desenvolupar el seguit de tasques plantejades en el projecte s'ha utilitzat la metodologia Waterfall[1][2]. Metodologia que consisteix a dur a terme un projecte en forma de cascada, això significa que cada nova fase no es comença fins que no s'ha completat l'anterior. Aquesta metodologia consta de sis etapes diferenciades que esdevenen una rere l'altra. Aquestes etapes són: Captura i documentació de requisits, disseny del sistema, disseny de l'app, codificació, test i manteniment.

S'ha escollit aquesta metodologia vers altres alternatives del tipus agile, a causa que el desenvolupament del projecte va a càrrec d'un únic individu i la paral·lelització de les tasques és pràcticament inexistent. El fet d'aplicar una metodologia no cíclica o iterativa té els seus inconvenients. La presa de requisits inicial ha de ser tan acurada com sigui possible per tal d'estalviar temps i esforç en etapes més avançades del projecte. De la mateixa forma, començar a dissenyar l'App sense donar per finalitzada i validada la presa de requisits funcionals i secundaris pot produir la inversió de temps en dissenys i prototipus finalment descartats.

5 Eines i tecnologies

A continuació es farà una breu descripció o de les principals eines i tecnologies escollides per a l'execució del projecte.

5.1 Marvel App

Marvel App és una eina en línia que permet crear prototips visuals per tal de facilitar el disseny de la interfície d'usuari en projectes de desenvolupament de programari. També permet programar les transicions entre les diverses pantalles que conformen una aplicació, així com els elements que activen aquestes transicions. Aquest aspecte permet que es puguin crear prototips destinats al testatge per part dels usuaris. En el nostre cas, utilitzarem aquesta eina per dissenyar un mockup que mostri la disposició dels elements dins les diferents pantalles. No usarem aquest mockup per a realitzar proves d'usuaris pel fet que es vol obtenir una aplicació que no disposar d'una gran varietat de transicions i pantalles.
L'elecció d'aquest programa radica principalment en el fet que s'ha utilitzat en projectes anteriors i es coneixen les funcionalitats principals de la interfície, i la versió gratuïta disposa de les funcionalitats necessàries per realitzar el prototip visual que es requereix.

5.2 Visual estudio

L'aplicació serà desplegada per a Android, com a entorn de desenvolupament integrat utilitzaré l'eina de Microsoft Andoid Studio, el qual és un entorn que ens permetrà crear i desenvolupar el nostre programari a més permet integrar projectes Xamarin.

5.3 Xamarin

Xamarin és una plataforma de codi obert que permet compilar aplicacions multiplataforma com Android, iOS, Windows en el mateix codi de programació, està escrit en C#. Per treballar amb la part del back-end i la base de dades que és on es guarda tota la informació, usaré .NET Core . ASP.NET Core és un framework de codi obert i multiplataforma per a la creació d'aplicacions modernes connectades a Internet, com ara aplicacions web i APIs Web. També ha sigut creat per ser multiplataforma.

5.4 Somee

Per tal de tenir l'aplicació en línia i no tenir el projecte en local host depenem d'un allotjament web, un allotjament web proveeix als usuaris d'internet un sistema per poder emmagatzemar informació, imatges, o qualsevol contingut accessible via web. En el meu cas del meu projecte he allotjat la part del back-end, per poder tenir la comunicació client-servidor amb la base de dades.

5.5 Mongodb

Quan desenvolupem una app mòbil on es necessita que l'usuari guardi informació de forma permanent, com ara una llista de tasques, generalment aquesta informació es guarda en una base de dades. Es recomana que l'app no es connecti directament a la base de dades, sinó mitjançant serveis REST API.

La plataforma escollida per la base de dades en si serà, MongoDB que és un sistema de base de dades NoSQL, orientat a documents de codi obert i escrit a C++, que en lloc de guardar les dades a taules ho fa en estructures de dades BSON (similar a JSON) amb un esquema dinàmic. Com que és un projecte de codi obert, els seus binaris estan disponibles per als sistemes operatius Windows, GNU/Linux, OS X i és usat en múltiples projectes o implementacions en empreses.

6 Disseny

La validació dels requisits del projecte dóna pas a la fase de disseny, destinada a estudiar i enginyar com materialitzar els requisits definits en el producte que es vol elaborar. L'elaboració de l'esbós del disseny ha marcat l'inici d'aquesta fase. A causa de la necessitat de familiarització tant amb una tecnologia relativament nova com amb unes plataformes desconegudes en termes de desenvolupament de software, el disseny de l'aplicació ha estat molt lligat amb una fase de preparació on s'han assolit nous coneixements sobre NetCore o Xamarin, s'han provat components potencialment útils i s'ha estudiat el potencial del framework per tal de predir els punts del desenvolupament més problemàtics i com es podrien enfocar.

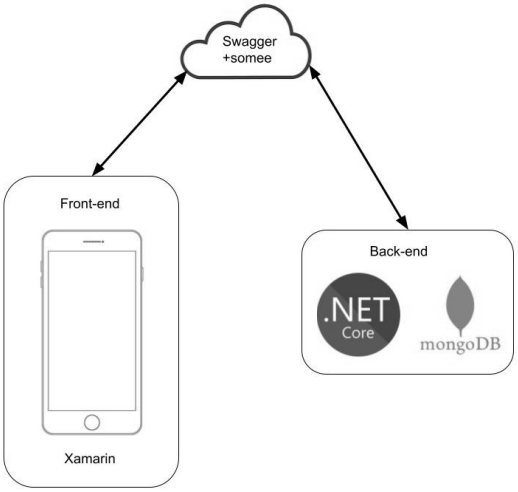

Fig. 1: Diagrama d'estructura

6.1 Interfíci d'usuari

S'ha realitzat un mockup utilitzant l'eina Marvel amb l'objectiu de plantejar la disposició dels elements a les diferents pantalles que conformen l'aplicació per tal que la interfície d'usuari quedi definida prèviament a la implementació, no havent de prendre decisions importants relacionades amb el disseny de la interfície durant aquesta etapa.

6.2 Arquitectura

Seguint la planificació explicada a l'apartat de planificació sobre la fase d'execució, s'explicaran a continuació els avenços del projecte respecte a la base de dades a l'API REST.

6.3 Disseny i implementació de la base de dades

La base de dades utilitzada per a la realització d'aquest projecte serà MongoDB. Inicialment, es va proposar fer servir MySQL, però finalment s'ha optat per l'opció de MongoDB, ja que ofereix més avantatges en projectes que estiguin bastants en el núvol. MongoDB és un sistema de base de dades NoSQL, orientat a documents i de codi obert. En lloc de desar les dades en taules, tal com es fa a les bases de dades relacionals, MongoDB guarda estructures de dades BSON (una especificació similar a JSON) amb un esquema dinàmic, fent que la integració de les dades en certes aplicacions sigui més fàcil i ràpida.

El disseny final de la base de dades consta de dues col·leccions amb els seus atributs corresponents. Aquestes són:

- **Pels aliments:** id(aleatori que crea automàticament MongoDB), name, ració, emisionsEH, terresCultiu, aigua, nitrogenUsat, fosforUsat, emissionsCO2
- **Pels usuaris:** id(aleatori que crea automàticament MongoDB), nom usuari, edat usuari, poblacio usuari, email usuari, password usuari

La col·lecció pels aliments està destinada a guardar informació relativa a aquests, amb aquesta informació es realitzaran els càlculs de la petjada de carboni amb les dades que haurà introduït l'usuari (grams de cada aliment). La col·lecció pels usuaris estarà destinada a organitzar la informació dels usuaris perquè es puguin registrar a l'aplicació i guardar informació rellevant sobre les seves dades a l'aplicació.

Després de revisar la configuració de la base de dades i d'elaborar noves funcions a l'API, s'ha decidit fer una sèrie de canvis pel que fa a les col·leccions i pel que fa a les dades que s'emmagatzemen i com s'emmagatzemen. El principal problema que ha provocat que s'hagin hagut de fer canvis és que el plantejament que es va fer al principi del projecte no va ser el correcte, ja que no es va pensar de la manera correcta. És a dir, MongoDB és una base de dades No-SQL i, per manca d'experiència i informació, la manera com s'anaven a emmagatzemar les dades va ser pensada com si fos una base de dades relacional i SQL. Precisament, la raó per la qual es va escollir MongoDB és perquè aquest projecte no requeria grans relacions entre les dades. En el nostre cas l'aplicació només relaciona els aliments cercats pels usuaris amb l'Id de l'usuari.

6.4 API REST

Per a la creació de l'API REST[3], es va decidir que el llenguatge a utilitzar per programar-la seria NetCore. Aquesta decisió s'ha pres d'acord amb la facilitat d'ús i l'ampli coneixement sobre aquest llenguatge per part del desenvolupador de programari d'aquest projecte, facilitant així la implementació de les funcionalitats de l'API REST.

Una API REST és un conjunt de regles que defineixen com poden les aplicacions o els dispositius connectar-se entre si i comunicar-se. En el nivell més bàsic una API és un mecanisme que permet a una aplicació accedir a recursos dins una altra aplicació. L'aplicació o servei que realitza l'accés s'anomena client i l'aplicació o servei que conté el recurs s'anomena servidor.

Les API REST es comuniquen a través de sol·licituds HTTP per executar funcions a la base de dades, com per exemple crear usuaris, esborrar usuaris, actualitzar informació usuaris dins d'un recurs. Per exemple una API REST utilitza sol·licitud GET per recuperar un registre (informació d'un usuari), POST per crear-lo, PUT per actualitzar informació i la sol·licitud DELETE per suprimir-lo.

6.4.1 Patró de disseny MVC

Com es descriu a la planificació original d'aquest projecte, una de les tasques és la de refactoritzar el codi utilitzant el patró de disseny de programari MVC. Per això s'han creat tres carpetes: controladors[4], models i vistes per al back-end. La carpeta "controladors" conté dos fitxers, un per cada col·lecció. Dins d'aquests es defineixen tots els mètodes de petició necessaris per al funcionament correcte de l'API. La carpeta "models" conté també dos fitxers (un per col·lecció), en què es defineixen els esquemes de cada col·lecció de la base de dades. Finalment, la carpeta "vistes" conté un fitxer "login.hbs" amb què podem comprovar de manera gràfica el funcionament correcte de l'autenticació per part dels usuaris registrats a la base de dades.

6.4.2 Mètodes de petició per a API

En aquest apartat s'han implementat tots els mètodes de petició[5] en dues col·leccions, tal com es descriu a la planificació del projecte. Les col·leccions implementades al complet són la col·lecció usuaris i la d'aliments. La col·lecció usuaris conté dos mètodes GET (un per llistar tots els registres que hi hagi a la taula i un altre per cercar pel camp ID), dos mètodes POST (un per al registre i un altre per a l'autenticació), un mètode PUT (per editar usuaris registrats) i un mètode DELETE (per eliminar usuaris). La col·lecció aliemnts conté dos mètodes GET (un per llistar tots els registres que hi hagi a la base de dades i un altre per cercar-los pel camp ID), un mètode POST (per guardar un nou aliment a la base de dades), un mètode PUT (per editar els aliments a la base de dades) i un mètode DELETE (per eliminar aliments).

6.4.3 Documentació API REST amb Swagger

La documentació d'una API[6] fa referència al contingut tècnic amb instruccions clares sobre com funciona una API, les seves capacitats i com fer-la servir. No importa que tan bona sigui una API per crear i estendre un servei de programari, ja que aquesta podria ser inutilitzable si els desenvolupadors no poden entendre com funciona. L'objectiu de la documentació d'aquesta API REST és treballar com una font de referència precisa capaç de descriure l'API en profunditat i actuar com a eina d'ensenyament i una guia per ajudar els usuaris a familiaritzar-se amb el programari i poder-lo utilitzar. Un manual complet que conté tota la informació necessària per treballar amb una API específica,

com ara funcions, arguments, tipus de retorn i classes.

El document també inclou exemples per donar suport a la informació. Aquest ha de ser fàcilment entenedora per als usuaris o desenvolupadors que requereixin el seu ús. Per trobar la llibreria que millor s'adeqüi al nostre projecte, s'ha buscat informació a Internet sobre com documentar una API. En fer-ho, el 90% dels resultats que van aparèixer, tots parlaven sobre "swagger-jsdoc". Per tant, es va buscar informació específicament per a aquesta llibreria. I, finalment, a causa de la fàcil implementació i del resultat final que ofereix a l'usuari, s'ha decidit fer servir "swagger-jsdoc"per documentar l'API REST d'aquest projecte[7].

Amb aquesta llibreria es poden documentar tots els enpoints de Xamarin utilitzant l'especificació Swagger OpenAPI 3 sense escriure fitxers YAML o JSON. Es poden escriure comentaris jsdoc sobre cada enpoint i la llibreria crearà la interfície d'usuari swagger. A la Fig.2. es pot veure un exemple de com documentar l'Schema d'una col·lecció de la base de dades.

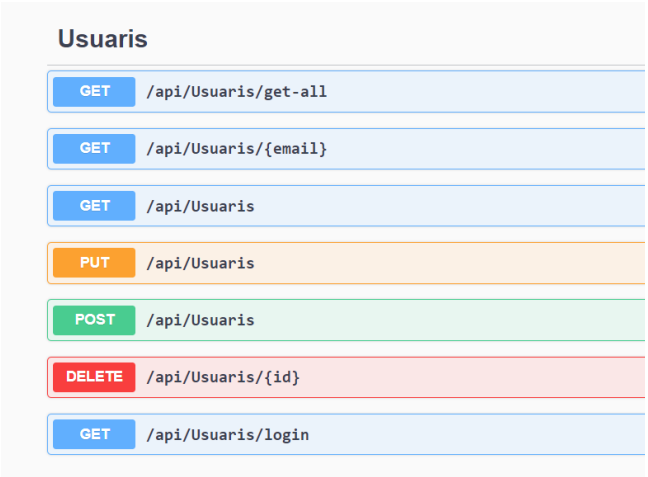

Fig. 2: Endpoint Col·lecció Usuari

Fig. 3: Resultat documnetació Schema

7 Implementació front-end

Durant aquesta fase s'han materialitzat els requisits seguint tant les decisions de disseny com la planificació estipulada. Existeixen funcionalitats no considerades durant la planificació del projecte, i d'altres que finalment s'ha decidit no implementar com la comparació de la petjada de carboni entre aliments.

Per entendre millor aquesta part cal esmentar que a l'aplicatiu existeixen tres tipus d'usuaris, els usuaris que no estiguin registrats, que accediran a l'aplicació clicant al logo de l'associació Faada de la Figura 4, també existeixen els usuaris que estiguin registrats a l'aplicació, que es podran registrar com es mostra a la Figura 6 i podran accedir a l'aplicació introduint les seves credencials com es mostra a la Figura 5. També existeix un tercer grup d'usuaris que són els que s'encarregaran del manteniment de l'aplicació. Les accions que realitza cada grup d'usuaris es pot observar a l'apart de l'apèndix als casos d'ús. Finalment, també cal comentar que les dades d'on s'extreuen els resultats de l'aplicació i per fer els càlculs provenen d'un full Excel que em va cedir l'associació Faada.

A continuació s'esmenten els principals aspectes desenvolupats:

- Inici de sessió
- Registre Usuari
- Pantalla calculadora
- Llista aliments
- Perfil usuari i editar perfil
- Pantalla configuració
- Guia ràpida

7.1 Gestió de pantalles

A Xamarin la gestió de les diferents pantalles de l'aplicació funciona com una pila. Quan iniciem l'aplicació, la pantalla que establim com a principal s'afegeix com el primer element de la pila. A mesura que es navega per l'aplicació i es visiten noves pantalles, aquestes es van afegint a la part superior de la pila i quan es vol tornar a la pantalla anterior, la pantalla actual s'elimina de la pila i es carrega l'últim element d'aquesta, que serà la darrera pantalla visitada, i que romandrà en el mateix estat en què estava l'últim cop que es va visitar.

L'aplicatiu consta de vàries pantalles diferents que anirem descrivint i que l'usuari podrà visitar, en realitzar una determinada acció i que corresponen als principals components desenvolupats per l'aplicatiu: Inici de sessió (Inici de sessió), Signin (registre d'usuari), calculadora (pantalla principal), perfil (l'usuari podrà consular i modificar informació relativa a ell), editar perfil (formulari per editar informació de l'usuari) i les pantalles relatives als aliments.

El punt d'entrada de l'aplicatiu és el component App. Aquest s'encarrega principalment de cridar a la resta de components perquè cadascú realitzi la seva funció. En el cas que ens ocupa, en comptes de renderitzar els components directament invoca al component AppNavigator que gestionarà les transicions entre pantalles.

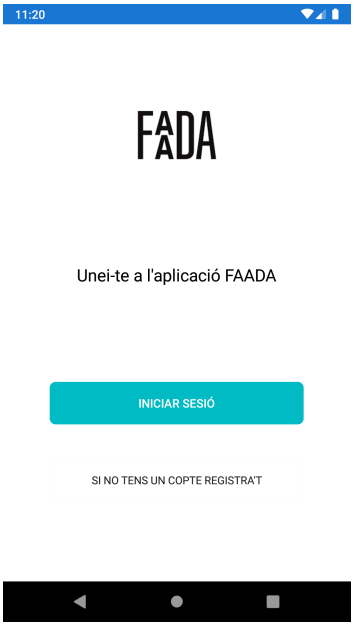

Fig. 4: Pantalla a l'inicar l'aplicació

7.2 Inici de sessió

Com en qualsevol aplicació tenim una pantalla inicial que dona la benvinguda a l'usuari a l'aplicació (Fig. 4). En aquesta pantalla l'usuari pot iniciar sessió o registrar-se en el cas que ho desitgi, si no es vol registrar pot fer clic al logo de l'associació Fadda. En el cas que estigui registrat i vulgui introduir les seves credencials (mail i contrasenya) ho podrà fer com es veu a la(Fig. 5) i polsar sobre el botó iniciar sessió, a continuació el codi desencadena un seguit d'accions, es crida a la funció login. Aquesta funció s'encarrega de realitzar una validació de les dades introduïdes per l'usuari mitjançant c#. Comprova que les dades introduïdes s'ajusten al format estipulat abans que siguin enviades a l'API. Un cop s'ha validat el format de les dades, aquestes queden llestes per poder ser enviades a l'API per comprovar que l'usuari existeix i que les dades són correctes. Finalment, si l'inici de sessió resulta que és erroni, s'informarà l'usuari a traves d'un pop-up del motiu pel qual no pot iniciar sessió i si l'inici de sessió és correcte, es redirigeix a l'usuari a la pantalla Calcualdora.

7.3 Registrar usuari

Al formulari d'inici de sessió es mostra un text on es proposa a l'usuari registrar-se en el cas que encara no ho hagi fet. Si es polsa sobre la paraula registrar-se, es crida a la funció signin, que redirigeix a l'usuari cap a la pantalla Signin.
Aquesta pantalla (Fig. 6) sol·licita a l'usuari la informació necessària per a registrar-se al sistema; nom usuari, email, edat i contrasenya. En prémer el boto registrar-se es crida a la funció signin, i tal com es fa amb el formulari d'inici

Fig. 5: Pantalla iniciar sessió

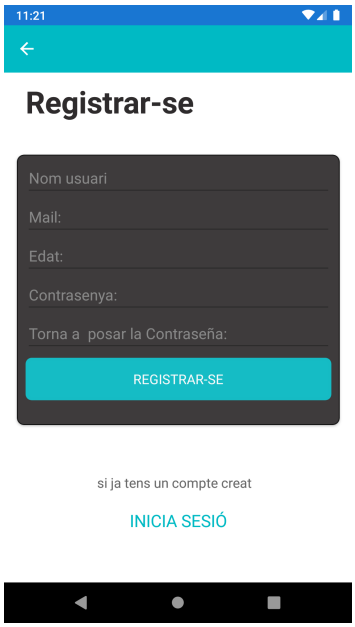

Fig. 6: Pantalla registrar usuari

de sessió, es valida el correcte format de les dades, per ser enviades posteriorment a l'API. Si el registre és correcte, aparèixer a un pop-up notificant a l'usuari, i se'l redirigirà a la pantalla inici de sessió quan aquest es tanqui, si l'enregistrament resulta erroni, es mostrarà un pop-up a l'usuari informant-lo del motiu de l'error.

7.4 Calculadora

Com a pantalla més important de l'aplicació tenim la pantalla que anomenarem calculadora on es realitzaran els càlculs dels aliments, on podrem indicar l'aliment que volem calcular i la quantitat, i l'aplicació en retornarà la petja de carboni i informació extra. Amb el nom de l'aliment i la quantitat d'aquest que vulgui cercar l'usuari, s'agafarà l'id del nom de l'aliment que s'hagi indicat, es realitzarà una petició a la base de dades a través de

l'api, aquesta retornarà amb un GET, tota la informació sobre l'aliment que tenim guardat a la base de dades (emissions CO_2, nitrogen usat, terres de cultiu usades...,), amb aquestes dades es realitzaran uns càlculs d'acord amb la porció en grams de l'aliment que hagi indicat l'usuari, finalment després de realitzar els càlculs es mostraran per pantalla (Fig.7). També voldria afegir que pels usuaris registrats aquesta informació es guardarà en la pantalla de perfil de cada usuari.

En aquesta pantalla quan l'usuari entra el producte es fa una crida a una funció per validar que les dades siguin correctes, en el cas que no siguin vàlides es mostrarà un pop-up indicant l'error(com un aliment que no estigui guardat a la base de dades), si les dades són vàlides es realitzà tot el procés esmentat anteriorment.
Finalment també podem veure a la barra superior tenim les opcions de tornar endarrere, anar al perfil de l'usuari, si està registrat i també podem veure una icona que serveix com a guia ràpida per navegar per aquesta pantalla i entendre d'una forma més senzilla que vol dir cada element que veiem a la pantalla.

Fig. 7: Pantalla calculadora

7.5 Llista aliments

Per completar la pantalla de la calculadora he afegit a l'aplicació una altra pantalla conjunta a la de la calculadora on es podrà visualitzar en un format de llista (Fig. 8) tots els aliments que existeixen a l'aplicació i que es podem consultar, els valors d'emissions d'aquests aliments estan calculats segons la ració mitjana de consum, aquest valor mitjà ha sigut extret d'un Excel que em va proporcionar l'associació FADDA, i que és el document d'on he tret tota la informació dels aliments i les emissions que fan referència a ells. Internament, aquesta pantalla fa una petició a la base de dades a través de l'API de tots els aliments que existeixen en la base de dades i els mostra per pantalla en format llista. Ta, bé cal comentar que els aliments els afegiran els usuaris administradors que serà l'associació FADAA, aquest procés es realitzarà a través d'una pantalla exclusiva pels administradors anomenada "afegir aliment".

Fig. 8: Pantalla llista productes

7.6 Perfil usuari i editar perfil

En aquesta pantalla es mostra informació relativa als usuaris registrats, els usuaris que no tinguin un compte a l'aplicació no tindrà accés a tenir un perfil dins l'aplicació, el que es pot veure és informació personal dels usuaris, aquesta informació es pot editar com es veu a la (Fig. 10), també existeix un apartat on cada usuari podrà veure un llistat dels aliments consultats amb anterioritat(Fig. 9).

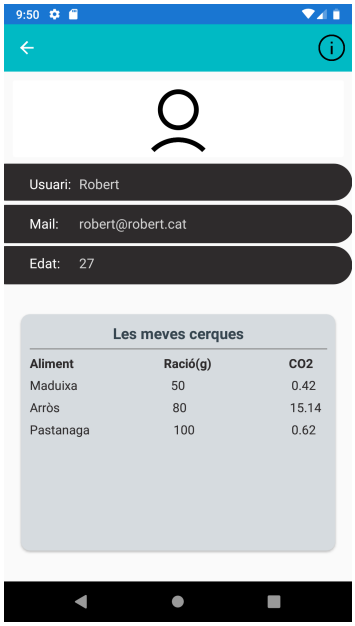

Fig. 9: Pantalla perfil usuari

7.7 Configuració

Per continuar amb la pila de mostra de les pantalles més importants de l'aplicació cal esmentar la creació d'una pantalla de configuració (Fig. 11) del compte de cada usuari, a aquesta pantalla si accedirà prement al logo de l'usuari de

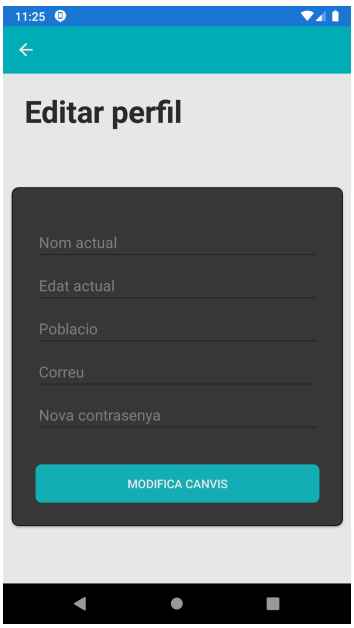

Fig. 10: Pantalla editar perfil usuari

la (Fig. 9) Aquesta pantalla té la intenció que cada usuari pugui gestionar el seu compte, primerament es mostrarà l'opció d'editar el seu perfil (Fig. 10), també tindrà l'opció de revisar la guia ràpida de les pantalles més rellevants de l'aplicació, també existeix una altra pantalla per posar-se en contacte amb l'associació Fadda a través de les seves xarxes socials, així com conèixer quin és el seu propòsit. Com en qualsevol aplicació l'usuari tindrà l'opció de tanca la sessió de l'aplicatiu i esborrar el seu compte, aquesta acció també elimina el compte de la base de dades.

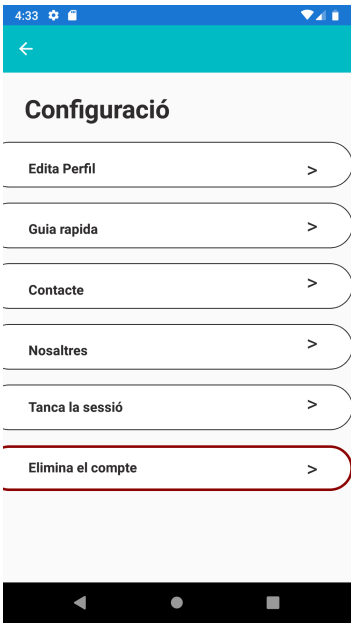

Fig. 11: Pantalla configuració

7.8 Guia ràpida

Per acabar el fil de mostra de les pantalles de l'aplicació vull esmentar la creació d'una llista de pantalles de guia ràpida(Fig. 12) que es podrà accedir des de la pantalla d'inici o de la pantalla de configuració del perfil de cada usuari. Aquesta aplicació ha sigut pensada i desenvolupada per ser senzilla i amigable, però penso que no està de més posar detall en les amb més dificultat que no saben ben bé que han de fer en descarregar-se l'aplicació i és per això que les pantalles principals de l'aplicació a la part de la barra superior es pot veure una icona d'informació en clicar aquesta icona apareixerà una pantalla exactament igual, però on apareixeran icones amb el logo d'informació en clicar en aquestes icones apareixerà un pop-up amb informació sobre l'acció que realitza cada element de la pàgina en concret.

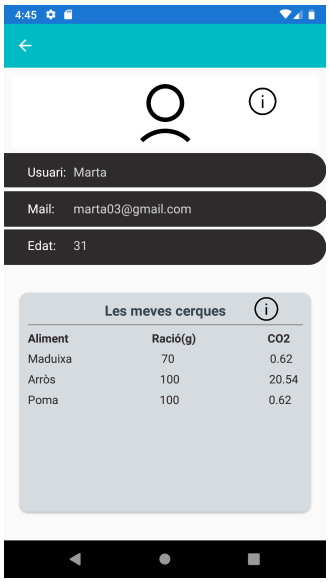

Fig. 12: Pantalla guia rapida "perfil usuari"

8 Conclusions

Aquesta secció està enfocada a elaborar una anàlisi del treball dut a terme durant el projecte, podent així extreure conclusions sobre el nivell d'èxit de la planificació i els requisits. Durant aquesta etapa s'elaborarà el document final del projecte per tal de ser lliurat, i finalment és dura a terme una presentació del nostre projecte.

Durant l'execució d'aquest projecte, s'han vist tecnologies, eines i plataformes molt interessants amb diferents serveis d'un gran interès per l'aplicació a desenvolupar, ja que el món del desenvolupament d'aplicacions mòbils és molt gran i amb moltes alternatives, que està evolucionant constantment. Considerant-ho tot, pel desenvolupament d'aquest projecte, a la fase inicial, s'ha escollit el framework de net core pel back-end i Xamarin pel front-end, que ens permet desplegar aplicacions mòbils multiplataforma.

Al principi del plantejament de l'aplicació tenia aquesta idea, ja que el framework Xamarin permet desplegar aplicacions pels diversos sistemes operatius amb una gran dependència de codi, cosa que permet no haver de fer el mateix treball dues vegades per cada sistema operatiu la realització del treball pels dos sistemes operatius es va descartar

al principi al dur a terme el planning del projecte per la curta durada d'aquest i perquè el fet de realitzar proves amb un sistema operatiu com iOS requereix d'un dispositiu amb aquest sistema operatiu i no tinc la possibilitat d'adquirir un dispositiu amb aquest SO durant diversos dies per realitzar les proves. Destacats aquests punts també vull esmentar que la planificació temporal inicial no va ser del tot correcta al principi i va patir certs canvis, ja que vaig posar moltes tasques al principi que requerien molt de temps i no vaig fer unes previsions del temps correcte, cosa que es va corregir per poder anar a temps d'acomplir les tasques futures i no patir més endarreriments en les següents etapes.

Els objectius del projecte plantejats a l'inici del projecte s'han aconseguit complir. S'ha assolit una aplicació senzilla i intuïtiva. També s'ha obtingut que l'aplicació sigui fàcil de consultar i visualitzar els resultats i no tingui una corba d'aprenentatge difícil.

Ha sigut gràcies a aquest projecte que he pogut aprofundir en els coneixements de netcore i xamarin, aprenent a realitzar una aplicació i fer una prova de com funciona la realització d'una aplicació real a partir d'una planificació prèvia en fases. Aquest projecte m'ha ensenyat a saber organitzar-me, a planificar i també a responsabilitzar-me d'un projecte d'aquesta magnitud. Gràcies a aquest projecte he pogut conèixer nous conceptes, aprofundir en conceptes que ja havia après abans. Finalment, aquest treball m'ha ajudat a veure tot el que he anat aprenent durant aquests anys i m'ha ensenyat que encara puc continuar aprenent i continuar creixent.

9 TREBALL FUTUR

Per concloure el document del projecte, s'ha considerat convenient dedicar una secció per mencionar tant possibles millores en l'aplicatiu, com noves funcionalitats i prestacions interessants d'afegir en futures versions.

Un aspecte interessant de millora de l'aplicació a l'hora de realitzar els càlculs de la petjada de carboni seria tindre en compte la procedència de cada aliment, per així tindre un valor més real de la petjada de carboni de l'aliment consultat.

Un altre punt que es podria millorar seria la de poder fer útil les dades dels usuaris, ja que amb aquesta aplicació es genera un gran volum de dades sobre les consultes que fan els usuaris, amb aquestes dades es podria fer estudis de quines són les conductes alimentàries dels usuaris, sempre mantenint l'anonimat d'aquests. Com a exemples que podrien ser útils les dades serien realitzar campanyes per conscienciar a la gent de l'impacte que tenen els aliments que consumeixen o que a l'hora de comprar aliments comprin aquells que siguin de proximitat.

Per finalitzar, també voldria mencionar que el projecte es podria adaptar a un entorn web, ja que la part del codi que correspon la back-end està preparat per incorporar una versió web per a ordinador, també un altre aspecte de treball futur seria incorporar més idiomes a l'aplicatiu per arribar a un ventall de persones més gran.

REFERÈNCIES

[1] CompraSoftware. "Metodología Waterfall: definición, pros y contres", [En línia], Disponible a:https://blog.comparasoftware.com/metodologia-waterfall/

[2] Instituto Europeo de Postgrado, "Metodología Waterfall: Modelo de gestión de proyectos en cascada", [En línia], Disponible a:https://www.iep.edu.es/metodologia-waterfall/

[3] Dot Net Office. "API Key Authentication in ASP.NET Core Web API", [En línia], Disponible a: https://www.dotnetoffice.com/2022/09/api-key-authentication-in-aspnet-core.html

[4] Learn, Microsoft. Çontroller action return types in ASP.NET Core web API", [En línia], Disponible a: https://learn.microsoft.com/en-us/aspnet/core/web-api/ action-return-types?view=aspnetcore-7.0

[5] C# Corner. "Frequently Used Status Code And How To Return Them From ASP.NET Core Web API", [En línia], Disponible a: https://www.c-sharpcorner.com/article/ frequently-used-status-code-and-how-to-return-them-from-asp-net-core-web -api/

[6] Documenting APIs. "Swagger UI tutorial", [En línia], Disponible a: https://idratherbewriting .com/learnapidoc/pubapis_swagger.html

[7] DEV. "Build a Node.js/Express REST API with MongoDB and Swagger", [En línia], Disponible a: https://dev.to/mikefmeyer/build-a-node-js-express-rest-api -with-mongodb-and-swagger-3de9

10 APÈNDIX

10.1 Requisits

En aquesta seccio es presenten els requisits de l'aplicació classificats entre funcionals i no funcionals.

10.1.1 Requisits funcionals

Els seguents requisits descriuen com s'ha de comportar l'aplicació en els diferents escenaris possibles.

RF1 Quan s'inicia l'aplicació per primer cop, el sistema ha de mostrar una pantalla de càrrega.

RF2 El sistema ha d'oferir una caixetí per introduir el nom de l'aliment.

RF3 El sistema ha de ser capaç de petjada dels aliments en CO2 .

RF4 El sistema ha de permetre retornar a la pantalla anterior en totes les pantalles.

RF5 El sistema ha de tindre una pantalla per comparar dos aliments.

RF6 Es sistema ha de permetre registrar als usuaris FAADA.

RF7 El sistema ha de permetre accedir als usuaris FAADA accedir a un registre de consultes dels aliments dels usuaris.

RF8 El sistema ha de permetre als administradors introduir nous aliments

RF9 Quan els administradors vulguin introduir nous aliments se'ls demanarà un títol, descripció i petjada d'aquell aliment.

RF10 El sistema ha de permetre registrar usuaris.

RF11 El sistema ha de mostrar un perfil pels usuaris registrats.

RF12 Els usuaris podran accedir al seu perfil.

RF13 El sistema ha de mostrar l'historial dels aliments calculats de cada usuari

RF14 El sistema ha de mostrar una pantalla del total de CO2 estalviat respecte a altres aliments.

RF15 El sistema ha de permetre als administradors editar/eliminar els aliments.

RF16 El sistema ha de permetre eliminar els comptes dels usuaris registrats.

RF17 El sistema ha de permetre als usuaris contactar amb els administradors.

RF18 El sistema ha de permetre a l'usuari registrar-se de forma opcional.

RF19 El sistema ha de mostrar informació extra sobre els aliments com(Terres de cultiu usades, aigua usada, nitrogen usat, fòsfor usat, kg de CO2 en transport).

RF20 Creació d'un manual d'instruccions per dur a terme el manteniment de l'aplicació mòbil.

RF21 El sistema ha de proveir a l'usuari crear alertes de la petjada de carboni que generaran durant un espai de temps.

10.1.2 Requisits no funcionals

RNF-1 Els administradors, les entitats i els usuaris únicament han de fer servir una aplicació.

RNF-2 El temps d'aprenentatge ha de ser menor de cinc minuts.

RNF-3 El sistema ha de comptar amb documentació (privadesa, etc.) degudament explicats.

RNF-4 El servidor i la base de dades s'han d'hostatjar a Infraestructures com a Serveis (IaaS).

RNF-5 La Infraestructura ha de poder escalar horitzontalment adaptant-se al nombre de peticions simultàniament.

RNF-6 Les pàgines s'han de renderitzar en la part del servidor.

RNF-7 La base de dades ha de permetre emmagatzemar fitxers.

RNF-8 La base de dades ha de ser capaç de gestionar 1 GB de dades.

RNF-9 Establir un color principal per identificar l'aplicació.

RNF-10 Les pantalles han de mantenir una estructura similar.

RNF-11 L'aplicació ha de ser user-friendly, intuïtiva i tenir un disseny responsive.

RNF-12 La Infraestructura com a Servei ha de ser cost zero.

RNF-13 S'ha de restringir l'accés a pàgines sense permisos.

RNF-14 Els usuaris no registrats, registrats i entitats han de tenir accés a accions de lectura de la base de dades.

RNF-15 Els permisos d'accés i modificació de la base de dades només poden ser canviats per l'administrador de la base de dades.

RNF-16 Els usuaris registrats han de tenir accés a accions d'escriptura de la base de dades.

RNF-17 La taxa d'errors comesos pel sistema haurà de ser menor de l'1% de les transaccions totals executades.

RNF-18 Tota operació a la base de dades s'ha de fer en un període de temps inferior a 15 segons. En cas d'arribar als 15 segons, l'operació es cancel·larà i saltarà una TimeOutException.

RNF-19 La base de dades ha de respondre les consultes amb un temps menor a 2 segons.

10.1.3 Restriccions

R-1 L'aplicació ha d'estar finalitzada abans del dia 08/02/2023.

R-2 Es limita el pressupost del projecte a 0€.

R-3 L'aplicació ha de poder ser utilitzada des de qualsevol dispositiu android.

10.2 Casos d'ús

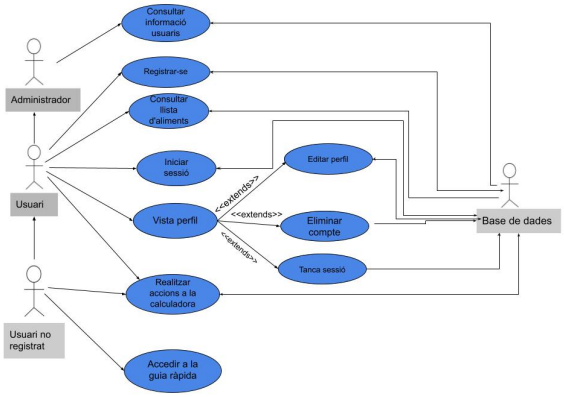

Fig. 13: Casos dús

10.3 Prototip

Prototips inicials dissenyats amb l'eina MarvelApp.

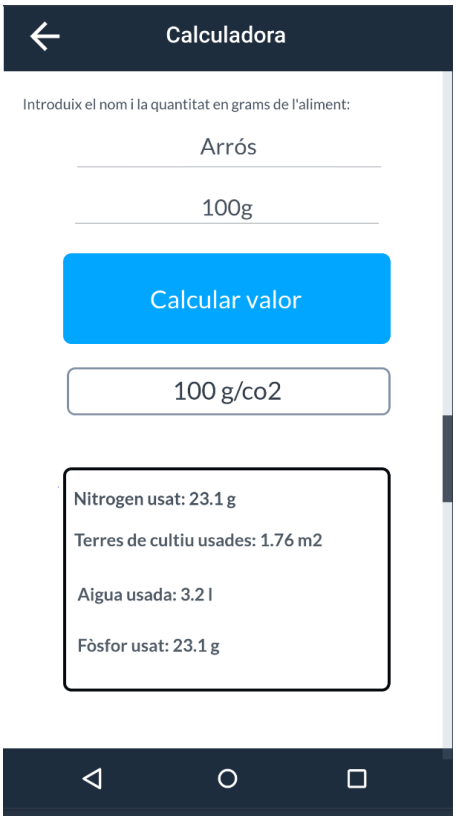

Fig. 14: Prototip pantalla calculadora

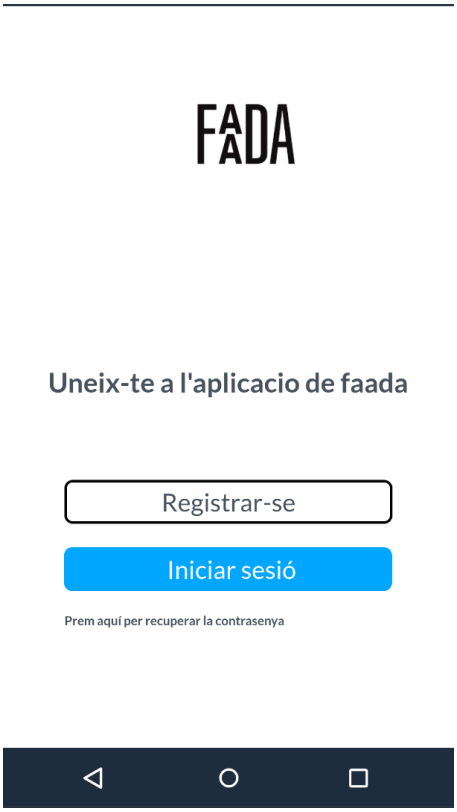

Fig. 15: Prototip pantalla Inicial

Fig. 16: Prototip pantalla perfil usuari