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

Casas Barco, Aleix; Bolta Torrell, Helena, dir. Planificador Gastronòmic : Organitza els teus àpats setmanals. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290068>

under the terms of the  license

Planificador Gastronòmic: Organitza els teus àpats setmanals

Aleix Casas Barco

Resum— Aquest document detalla el procés que s'ha seguit per a desenvolupar una aplicació web que a ajuda als usuaris en l'organització de menús saludables i variats, i en la preparació de llistes de la compra. El propòsit és crear una base de dades de receptes, ingredients i usuaris, una API per comunicar amb el front-end, una interfície d'usuari intuïtiva, i permetre als usuaris crear i visualitzar les receptes pròpies i d'altres, així com generar la llista de la compra. Utilitza una metodologia àgil adaptada a un sol desenvolupador, amb tecnologies com ReactJS, Tailwind CSS, NodeJS, i Figma. El projecte es divideix en les fases d'anàlisi de mercat, disseny del model de dades, desenvolupament de funcionalitats i desplegament.

Paraules clau— Organització d'àpats, Àpats saludables i variats, Llista de la compra, Aplicació Web, Interfície d'usuari intuïtiva, Metodologia àgil

Abstract— This document details the process that has been followed to develop a web application that helps users organize healthy and varied menus and prepare shopping lists. The goal is to create a database of recipes, ingredients and users, an API to communicate with the front end, an intuitive user interface, and allow users to create and view their own and other recipes, as well as generate the shopping list. It uses an agile methodology adapted to a single developer, with technologies such as ReactJS, Tailwind CSS, NodeJS, and Figma. The project is divided into the phases of market analysis, data model design, functionality development and deployment.

Index Terms— Meal organization, Healthy and varied meals, Shopping list, Web application, Intuitive user interface, Agile methodology

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

La realització d'aquest projecte sorgeix de la necessitat en què m'he trobat durant aquests anys de carrera vivint en un pis d'estudiants. Aquest fet ha provocat que jo m'hagi d'organitzar tots els àpats que faig durant la setmana per anar a comprar els ingredients necessaris i no haver d'anar un cop i un altre al supermercat.

És per això que fins a aquest moment, he estat apuntant els àpats de cada setmana en un bloc de notes al mòbil, després de decidir que menjaria cada dia intentant seguir una dieta equilibrada i variada, havia d'anar apuntant un per un tots els ingredients necessaris per a preparar els plats i finalment, anar al supermercat a comprar. Aquestes tasques s'han convertit en molt monòtones i repetitives i al final, m'he trobat amb el fet que sempre acabo repetint els mateixos plats setmana rere setmana.

Durant un temps, vaig estar provant diferents aplicacions i pàgines web per a substituir aquestes tasques, però cap

d'aquestes va satisfer les meves necessitats, ja que jo volia poder crear les meves pròpies receptes. Això em va portar a pensar que "Que millor que dissenyar i desenvolupar jo la meva pròpia aplicació que s'adapti a totes les meves necessitats?". Va ser en aquest moment, quan després de donar-li unes quantes voltes a què volia exactament i de reflexionar sobre si tenia els coneixements necessaris per a desenvolupar el projecte, vaig presentar la proposta d'aquest treball de final de grau.

2 OBJECTIUS

Aquest treball de final de grau es centra en desenvolupar una aplicació web completa composta per un back-end i un front-end que ofereixi una solució eficaç i versàtil als estudiants que viuen fora de casa durant la setmana per a ajudar-los a organitzar els seus àpats de manera saludable, variada i sense complicacions. Els objectius principals del projecte són els següents:

- Dissenyar i crear una base de dades de receptes, usuaris i ingredients per a poder guardar tota la informació necessària del projecte.
- Desenvolupar una API que comuniqui la informació de la base de dades amb el front-end de l'aplicació per a així mostrar només la informació necessària.

- E-mail de contacte: acasasbarco@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Helena Boltà Torrell (Àrea de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2023/24

- Desenvolupar una interfície d'usuari intuïtiva, moderna i fàcil d'utilitzar que permeti a l'usuari visualitzar les receptes i els ingredients.
- Proporcionar a l'usuari la possibilitat de visualitzar la planificació setmanal d'àpats de diferents formes així com accedir a les receptes.
- Oferir a l'usuari la possibilitat de crear les seves pròpies receptes i visualitzar les dels altres usuaris.
- Permetre a l'usuari l'opció de generar la llista de la compra amb els ingredients que formen part de les receptes que aquest, ha escollit per a la pròxima setmana.

Per a poder completar amb èxit tots aquests objectius i que aquests es converteixin en funcionalitats de l'aplicació, també s'haurà de desenvolupar un sistema de registre i inici de sessió d'usuaris perquè aquests puguin guardar i recuperar totes les seves dades en el cas de tancar la sessió.

3 ESTAT DE L'ART

La necessitat de crear una nova aplicació per a l'organització dels àpats setmanals es justifica per diverses raons clau, principalment les limitacions trobades en les solucions existents al mercat. Tot i que hi ha nombroses aplicacions i plataformes moltes d'elles, com "MyFitnessPal"^[1] i "Yummly"^[2], ofereixen funcions útils per al seguiment nutricional i la recerca de receptes, però fallen en personalització i flexibilitat a l'hora de fer la planificació dels àpats o gestionar les receptes pròpies dels usuaris. Altres, com "Paprika"^[3] i "Mealime"^[4] que faciliten la planificació dels àpats, poden ser menys intuïtives o no integren completament la planificació dels àpats amb la creació de llistes de la compra.

L'aplicació desenvolupada en aquest projecte busca omplir aquest buit proporcionant una solució més completa i adaptable. A diferència de les altres aplicacions, permet als usuaris no només accedir a receptes, sinó que també crear i compartir les seves, a més d'organitzar àpats saludables i generar la llista de la compra basada en la selecció. La seva interfície d'usuari intuïtiva i la seva capacitat per a personalitzar l'experiència d'usuari la fan destacar en el mercat.

4 METODOLOGIA DE DESENVOLUPAMENT

Per a aconseguir complir amb tots els objectius del projecte, s'ha decidit utilitzar una metodologia de desenvolupament àgil que incorpora molts elements de Scrum^[5], però que s'adapta a un sol desenvolupador. Aquesta adaptació permet l'entrega incremental de productes de software en cicles curts de temps anomenats Sprints, centrant tots els rols de Scrum en la mateixa persona i elimina les reunions de planificació, diàries, de revisió i de retrospectiva del sprint. S'ha pres aquesta decisió perquè, tenint en compte que tot el projecte serà desenvolupat per una sola persona i en un període de temps relativament curt

d'uns 5 mesos, adaptar aquesta metodologia que ja s'ha usat amb anterioritat ofereix una gran flexibilitat i adaptabilitat a l'hora d'ajustar i reorganitzar les tasques a mesura que es va avançant el projecte. A més, permet fer servir una metodologia ja coneguda per a centrar-se en el desenvolupament i no haver de fer-ho en aprendre una nova metodologia.

Per al control de versions del codi del projecte, s'utilitzarà GIT amb un repositori de GitHub on es guardaran i gestionaran totes les versions del codi. A més, se seguirà la següent política de branques que facilitarà l'organització del repositori: la branca principal contindrà el codi estable i funcional del projecte i s'anirà actualitzant amb les branques de les funcionalitats que seran desenvolupades paral·lelament i de forma aïllada sense afectar el codi de la branca principal fins que la funcionalitat estigui totalment desenvolupada i provada. Aquesta política assegurarà que el codi es mantingui organitzat i facilitarà un seguiment clar dels canvis realitzats en cada funcionalitat.

Pel que fa a l'organització del projecte, es farà servir la plataforma Monday per a recopilar i especificar totes les tasques a fer. Aquesta plataforma t'ofereix la possibilitat de tenir diverses pestanyes en diferents formats com poden ser taules, Diagrames de Gantt, Kanban o Cronogrames que et faciliten molt la visualització de tota la informació del projecte i de cadascun dels sprints que aquest tindrà.

Subelemento	Planificació	Estat
Login i Registre d'usuaris	23 oct. - 5 nov.	Done
Planificació setmanal d'àpats	6 - 19 nov.	Done
Sidebar	6 - 19 nov.	Done
Llista de la compra i recerca d'ingr...	20 nov. - 3 dic.	Done
Crear receptes	4 - 17 dic.	Working on it
Pàgina "Mis Recetas"	18 - 31 dic.	To Do
Pàgina "Explorar Recetas"	18 - 31 dic.	To Do

Il·lustració 1: Vista de les tasques de desenvolupament a la plataforma Monday

5 PLANIFICACIÓ

El desenvolupament del projecte es dividirà en 4 etapes, cadascuna d'aquestes amb les seves pròpies tasques i objectius. A continuació, es detallen les principals fases de desenvolupament que seguirà el projecte:

En primer lloc, trobem la fase d'Anàlisi de Mercat, aquesta és la que inicia el projecte amb la recopilació d'informació rellevant del mercat. A més inclou de definició dels objectius del projecte i l'anàlisi de requisits. Aquesta fase ja s'està portant a terme i acaba amb l'entrega d'aquest informe el dia 8 d'octubre.

A continuació, es durà a terme la fase de disseny del model de dades que formarà el primer sprint del projecte. Aquesta estarà formada per la recerca de la base de dades més adient, el disseny de l'estructura i creació d'aquesta i per la recerca i obtenció de receptes i ingredients per a

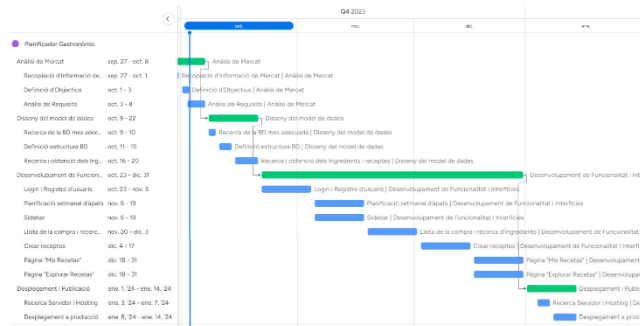
finalment, inserir-los a la base de dades seleccionada mitjançant scripts que també seran creats durant aquesta fase.

La tercera fase del projecte és la de desenvolupament i estarà formada pels següents sprints on es desenvoluparan totes les funcionalitats:

- El segon sprint començarà el dia 23 d'octubre i acabarà el 5 de novembre i en aquest es desenvoluparà tot el relacionat amb el login i el registre dels usuaris a l'aplicació.
- El tercer sprint anirà del dia 6 al 19 de novembre i estarà format per la pàgina de planificació dels àpats setmanals i el sidebar de l'aplicació.
- En el quart sprint, es desenvoluparan les funcionalitats relacionades amb la llista de la compra i en afegir ingredients de forma manual a aquesta. Està planificat entre els dies 20 de novembre i 3 de desembre.
- El cinquè sprint començarà el dia 4 de desembre i acabarà el 17 i en aquest es desenvoluparà tot el necessari per a crear receptes.
- Finalment, el sisè i últim sprint d'aquesta fase de desenvolupament, anirà del dia 18 al 31 de desembre i consistirà a desenvolupar les pàgines de "mis recetas" i "explorar recetas".

En cadascun d'aquests sprints de desenvolupament, dissenyarà les interfícies necessàries amb Figma, es crearan els endpoints del back-end i també es crearan totes les funcionalitats del front-end. Finalment, també es duran a terme tests unitaris de cadascun dels endpoints del back-end i tests manuals i d'usuari per a les interfícies del front-end.

Per acabar, trobem la fase de desplegament i publicació de l'aplicació, aquesta formarà el setè i últim sprint que anirà del dia 1 al 14 de gener. Durant aquestes setmanes, es buscarà un servidor per allotjar tots els serveis de l'aplicació. A continuació es pot veure aquesta planificació de totes les fases del projecte en format de diagrama de Gantt:



Il·lustració 2: Diagrama de Gantt del projecte

6 ANÀLISI

6.1 Anàlisi de requisits

A partir dels objectius definits per al projecte i de les necessitats tècniques que aquest requereix, s'ha elaborat un llistat de requisits. S'ha utilitzat la metodologia MoS-CoW^[6] per a prioritzar aquests requisits segons la seva importància en el desenvolupament del projecte^[7] agrupant-los en les següents quatre categories:

- (M)ust have: Essencial pel funcionament i la integritat del sistema.
- (S)hould have: Afegeix valor a l'aplicació i, per tant, s'hauria de fer.
- (C)ould have: Afegeix valor, però no afectaria en res si no es fes en aquest moment, s'afegeix al backlog.
- (W)on't: No és imprescindible, però es pot afegir al backlog per si hi ha temps i no incrementa l'esforç.

A continuació es troben alguns dels requisits més importants, es pot trobar el llistat complet a l'Annex A1.

Taula 1: Requisits Funcionals

TIPUS	REQUISITS FUNCIONALS	PRIORITAT
RF-4	L'usuari ha de poder crear una planificació inicial dels àpats de la setmana	M
RF-4.1	L'usuari ha de poder editar aquesta planificació en qualsevol moment	S
RF-4.4	L'usuari ha de poder buscar receptes a la Web App	S
RF-5	L'usuari ha de poder generar la llista de la compra per als àpats d'aquella setmana de forma automàtica	M
RF-6	L'usuari ha de poder crear les seves pròpies receptes	M
RF-7	L'usuari ha de poder explorar receptes d'altres usuaris	C
RF-9	L'usuari ha de poder veure els detalls de les receptes	S

Taula 2: Requisits No Funcionals

TIPUS	REQUISITS NO FUNCIONALS	PRIORITAT
RNF-1 Estètic	La interfície d'usuari ha de ser intuïtiva per a l'usuari	S
RNF-2 Rendiment	El temps de càrrega inicial de la Web App ha de ser inferior a 2 segons	S
RNF-3 Seguretat	Les contrasenyes dels usuaris han d'emmagatzemar-se de forma segura	M
RNF-4 Estètic	L'aplicació ha d'estar dissenyada per a dispositius mòbils però també ha de ser suportada per a ordinadors	S

6.2 Elecció del Servei de Base de Dades

Per a seleccionar el millor servei de base de dades per al projecte, s'ha elaborat una comparativa centrada en diferents serveis que ofereixen les següents característiques:

- Autenticació d'usuaris amb Google o altres proveïdors externs
- Emmagatzematge per a les imatges dels usuaris
- Connexió ràpida i fàcil amb una api pròpia desenvolupada amb NodeJS.

A partir d'aquests criteris, s'ha fet una recerca dels diferents serveis gratuïts més populars i s'ha analitzat el que ofereix cadascun d'aquests.

En aquesta comparativa en forma de taula que es pot observar a l'Annex A2, es poden observar aspectes com les característiques mencionades anteriorment per als serveis de Firebase, AWS^[8], DynamoDB, MongoDB Atlas i Supabase.

Després d'aquest anàlisi, s'ha optat per Firebase com la millor opció de servei de base de dades per a aquest projecte. Aquesta elecció es basa en la seva capacitat per satisfer les necessitats específiques del projecte. Firebase ofereix bases de dades, autenticació d'usuaris i emmagatzematge d'imatges, i el seu gran avantatge és la facilitat d'ús sense requerir configuració addicional per a obtenir tots els serveis. A més, la versió gratuïta de Firebase proporciona els recursos necessaris per a les transaccions i operacions esperades per a la meua Web App, la qual cosa la converteix en una opció eficient i econòmica per a aquest projecte.

Això permetrà centrar-se en el desenvolupament sense haver-se de preocupar pel manteniment de la infraestructura, un aspecte fonamental per a l'èxit del projecte.

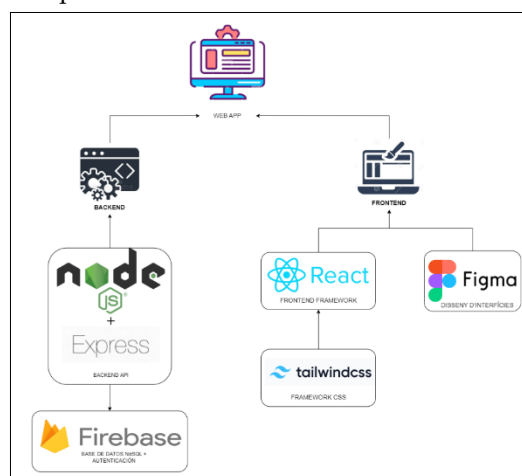
6.3 Tecnologies utilitzades

Per a seleccionar les tecnologies de desenvolupament per a aquest projecte, s'ha fet una elecció considerant les diferents opcions a partir de les tecnologies ja conegudes i les que podien oferir millors solucions. Sempre s'ha tingut en compte la prioritat de minimitzar la corba d'aprenentatge que suposaria aprendre alguna tecnologia des de zero per a aquest projecte. Les tecnologies seleccionades han estat les següents:

- Per al front-end de l'aplicació s'utilitzarà la llibreria de JavaScript, ReactJS que a part de ser conegut per la seva eficiència i flexibilitat a l'hora de crear interfícies d'usuari, és el més popular i suportat del món i, per tant, disposa de moltes llibreries que ajuden als desenvolupadors a simplificar moltes de les funcionalitats a crear. També s'usarà Tailwind CSS com a framework de disseny per a aplicar estils a les interfícies web de forma àgil i optimitzada.
- El back-end de l'aplicació serà desenvolupat amb NodeJS i el framework Express, ja que és realment veloç i ens permetrà desenvolupar tots els endpoints necessaris

de forma fàcil, ja que utilitza JavaScript i, per tant, la seva corba d'aprenentatge serà mínima. Aquesta API s'encarregarà de realitzar peticions sobre la base de dades de Firebase. Per al disseny de les interfícies d'usuari, s'usarà Figma, aquesta eina gratuïta permetrà crear prototips i dissenyar les interfícies.

En resum, aquest conjunt d'eines i tecnologies, ha estat seleccionat per la seva eficàcia, eficiència, popularitat i capacitat per a satisfer les necessitats del projecte. A continuació es pot observar un diagrama de components que reflecteix totes les tecnologies que s'utilitzaran de forma gràfica i esquemàtica:



Il·lustració 3: Diagrama de components de l'aplicació

En aquest diagrama, creat amb Draw.io^[9], podem visualitzar les tecnologies i frameworks que s'utilitzaran tant al front-end com al back-end del projecte formant una Web App completa.

6.4 Disseny estructura Base de Dades

Després d'haver seleccionat el servei de base de dades, va ser el moment de dissenyar l'estructura d'aquesta. Com que Firebase Firestore és una base de dades estructurada en documents i col·leccions, el disseny d'aquesta pot ser molt flexible, però s'ha optat per a un disseny bastant rígid per a estalviar-se consultes i així reduir el temps de resposta de les peticions al back-end. Aquesta decisió s'ha pres considerant que el nombre de dades que tindrà l'aplicació serà petit i, per tant, no ocuparà molta memòria el fet de tenir alguna informació repetida en diversos documents.

L'esquema resultant amb l'estructura seleccionada es pot observar a l'Annex A3. En aquest esquema podem veure com hi haurà tres col·leccions diferents:

- user: guardarà la informació de l'usuari per a iniciar sessió, les seves dades personals, la seva foto de perfil, la llista de la compra i el llistat de receptes preferides en diversos subdocuments i finalment, la planificació setmanal de l'usuari en un altre document format alhora per altres documents.

- ingredient: contindrà la informació de tots els ingredients de la base de dades, aquesta estarà composta per l'identificador de cada ingredient, el seu nom, la imatge i les calories que conté una quantitat determinada de cada ingredient.
- recipe: guardarà la informació de totes les receptes que estaran a l'aplicació. Cadascuna d'aquestes tindrà un títol, descripció, ingredients en format text, ingredients enllaçats amb la taula de la base de dades, passos per a preparar la recepta, dificultat, temps de preparació, una imatge, calories, un enllaç de referència i finalment l'identificador de l'usuari que l'ha creat.

6.5 Disseny de la interfície gràfica

Després d'analitzar tots els requisits de l'aplicació i, per tant, conèixer les funcionalitats que tindrà, és el moment de dissenyar les interfícies de l'aplicació. Fer a fer-ho s'ha utilitzat Figma, s'han realitzat els dissenys de les pantalles que tindrà l'aplicació buscant crear una experiència d'usuari intuïtiva i agradable. És per això que s'ha escollit una gamma de colors atractiva que crida l'atenció de l'usuari gràcies als seus tons blaus i taronges.

Algunes de les pantalles que s'han dissenyat es poden veure a continuació, aquestes són les de la planificació setmanal i la de crear receptes.



Il·lustració 4: Disseny pantalla planificació setmanal i crear receptes

6.6 Obtenció i Filtratge de Dades

Un cop sabia la informació i, per tant, els camps dels ingredients que necessitava per a obtenir una base de dades completa, va ser el moment de fer recerca de la millor forma d'obtenir els ingredients de cuina en castellà així com una imatge i les seves calories. Després d'investigar i navegar per diferents pàgines web, vaig trobar amb la indicada. Aquesta era la de Nutritionix^[10], una base de

dades de productes, restaurants i ingredients de cuina.

Aquest últim apartat era el que necessitava i afortunadament, tenia la seva versió en castellà. Com que els resultats de tots els ingredients es mostraven en una llista amb resultats paginats i tenia una longitud total de 2332, la forma més fàcil d'aconseguir-los tots va ser Scrapejant aquesta pàgina web. Aquesta podria haver estat una tasca molt complexa, però per sort, l'extensió de Google Chrome de Ginee Data Scraper^[11] va facilitar molt la feina, ja que amb una petita configuració i una mica de temps i proves, vaig obtenir tots els resultats en una cadena de text.

Un cop vaig tenir tots els ingredients, el següent pas va ser netejar totes aquestes dades eliminant les repetides i totes aquelles que eren productes que pots trobar en un supermercat relacionat amb menjar, però que realment no són ingredients. Aquesta va ser una feina bastant lenta i pesada, ja que l'única forma de fer-ho i assegurar-se que estigués bé era de forma manual. Finalment, després d'algunes hores eliminant ingredients, el resultat final va estar de 848 en lloc dels 2332 inicials. A partir d'aquí l'última feina que quedava per als ingredients va estar enumerar-los a partir del camp "id" i guardar-los en el format JSON preparat per a inserir a la base de dades.

La feina amb les receptes va ser totalment diferent, ja que després de buscar per moltes pàgines diferents, cap podia proporcionar tots els camps que necessitava per a l'estructura de dades seleccionada. És per això que la solució que vaig escollir va ser demanar-li al ChatGPT^[12] que escrigués receptes de cuina amb tots els camps que havia pensat amb anterioritat. El resultat em va sorprendre gratament, ja que va ser capaç d'escriure 30 receptes diferents amb molts detalls i tot el que jo necessitava. Tot i això, quedava la feina més laboriosa amb les receptes. Aquesta era parsejar els ingredients que formaven les receptes en format string amb subdocuments que contenen la informació dels ingredients recopilats anteriorment. Aquesta també va estar una feina manual, ja que després de provar-ho amb diversos scripts creats amb JavaScript, el resultat no era gens satisfactori i al tractar-se de "només" 30 receptes, vaig optar per fer-ho manualment.

Finalment, després de dures tasques de recerca i manipulació de totes les dades per a tenir-les amb el format desitjat per a inserir a la base de dades, la feina que quedava era la d'elaborar un petit script en JavaScript que es connectés a Firebase i inserís les dades en les diferents col·leccions.

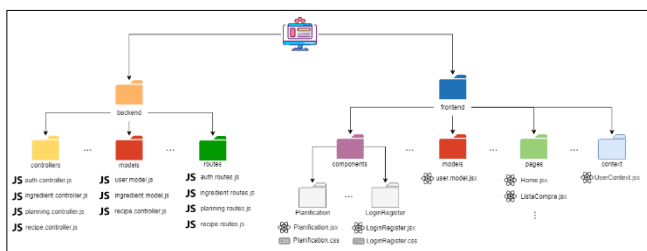
7. DESENVOLUPAMENT

Un cop tenim totes les dades necessàries per a desenvolupar el projecte, és el moment de començar amb el desenvolupament el qual s'ha dividit en diferents sprints on es desenvolupava un o diversos components que conformen l'aplicació.

7.1 Configuració inicial i estructuració del projecte

En aquesta primera fase de desenvolupament, s'ha establert la infraestructura bàsica del projecte, començant amb la creació del codi i la creació del repositori de Git sincronitzat amb GitHub. El back-end s'ha desenvolupat amb NodeJS, s'ha definit l'estructura de carpetes, la creació dels models de dades, la configuració de les rutes i la connexió amb Firebase. Per al front-end, s'ha optat per ReactJS amb Vite, integrant TailwindCSS i NextUI.

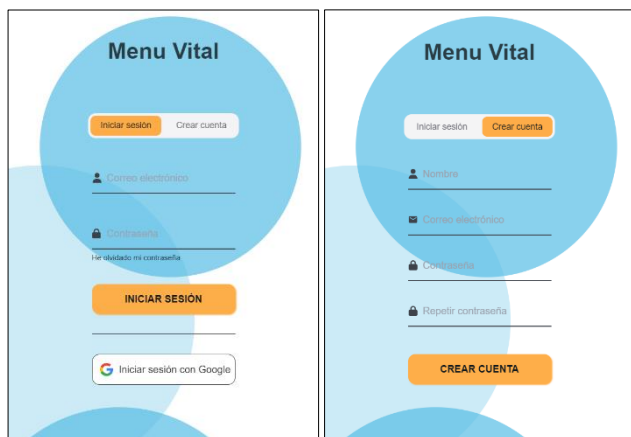
Tant en el back-end com en el front-end, s'ha definit una estructura organitzada de fitxers i components. El back-end inclou controladors per a diferents tipus de peticions i fitxers de rutes, mentre que el front-end disposa de carpetes per a components, pàgines, i un sistema de rutes protegides per a l'autenticació d'usuaris. A continuació, es pot veure un esquema que representa l'estructura utilitzada.



Il·lustració 5: Diagrama d'estructura de fitxers

7.2 Login i registre d'usuaris

La primera tasca de desenvolupament va ser la de crear les pàgines de login i registre d'usuaris. Per a això s'han creat diferents endpoints al back-end per a registrar usuaris, inicial la sessió, restablir la contrasenya i iniciar sessió amb Google. Finalment, s'han creat els dissenys d'aquesta pàgina al front-end així com el del formulari per a restablir la contrasenya. El resultat d'aquesta pàgina és el següent:



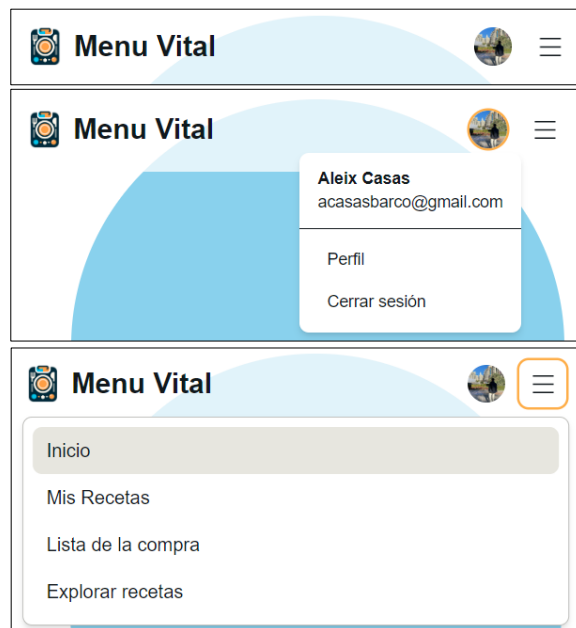
Il·lustració 6: Pàgines d'inici de sessió i registre d'usuaris

7.3 Component NavBar

Aquest component és el que es mostra a la part superior de totes les pàgines de l'aplicació un cop has iniciat sessió, permetent a l'usuari navegar entre pàgines i tancar la sessió. S'ha creat un Context de React, anomenat

UserContext, que permet compartir les dades de l'usuari entre components sense passar-les com a propietats. Aquest context s'utilitza per a gestionar l'autenticació de l'usuari i la seva informació.

A més, s'ha creat un nou endpoints en l'API per permetre que l'usuari tanqui la sessió, eliminant les seves dades del front-end. El resultat d'aquest component amb els seus desplegable oberts és el següent:



Il·lustració 7: Component NavBar amb els seus desplegables

En aquest moment em vaig adonar de la necessitat de reiniciar sessió a cada canvi de pàgina des de la NavBar, ja que la sessió no es guardava, resultant incòmoda. Per solucionar-ho, després d'investigar, s'ha escollit crear un token i enviar-lo en una cookie quan s'inicia sessió des del back-end. A cada petició que es realitza des del front-end cap a l'API, s'envia una cookie. Per garantir la seguretat, s'ha implementat un middleware que s'activa abans de donar resposta. Aquest middleware s'encarrega de verificar el token inclòs en la cookie, permetent desxifrar la informació de l'usuari i retornar les dades pertinents. En cas que el token no sigui vàlid, el back-end emet un error d'autenticació, provocant el tancament automàtic de la sessió en el front-end, així reforçant la seguretat de l'aplicació.

7.4 Planificació setmanal i recerca de receptes

El component principal que es mostra un cop l'usuari entra a l'aplicació, és el de la planificació setmanal dels àpats. En aquest component es mostren els àpats d'esmorzar, dinar i sopar per a cadascun dels dies de la setmana. En cadascun d'aquests àpats, l'usuari té la possibilitat d'afegir receptes gràcies al fet que s'obre un "modal" que et permet buscar en les receptes de l'aplicació. Un cop s'ha afegit alguna recepta, l'usuari també té la possibilitat d'eliminar-la de la planificació en el cas que hagi canviat d'opinió o d'afegir altres receptes en un mateix àpat. Finalment, l'usuari també té la possibilitat d'eliminar tots els àpats o

d'afegir tots els ingredients a la llista de la compra.

Per a crear tot aquest component s'han desenvolupat els següents endpoints a la API:

- GET /planning : Per a obtenir la planificació setmanal de l'usuari autenticat
- POST /add-meal : Per a afegir una recepta a la planificació de l'usuari en l'àpat especificat en el cos de la petició
- POST /delete-meal : Per a eliminar una recepta de la planificació de l'usuari en l'àpat especificat en el cos de la petició
- DELETE /delete-planning : Per a eliminar totes les receptes de la planificació setmanal de l'usuari
- POST /add-planning : Per a afegir tots els ingredients de les receptes a la llista de la compra
- GET /recipes: Per a afegir tots els ingredients de les receptes a la llista de la compra

Per a utilitzar tots aquests endpoints, s'han creat les funcions pertinents al front-end que s'encarreguen d'aconseguir i gestionar les dades que aquestes retornen. També s'ha creat algunes funcions per a renderitzar algunes parts d'aquesta pantalla. Com que en els requisits del projecte s'especificava que la planificació s'havia de poder veure de dues formes diferents, s'ha creat dues vistes diferents i el menú per a poder afegir i buscar les receptes. El resultat d'aquest component és el següent:



Il·lustració 8: Component de planificació i popup d'afegir receptes



Il·lustració 9: Vista alternativa del component de planificació

7.5 Llista de la compra i recerca d'ingredients

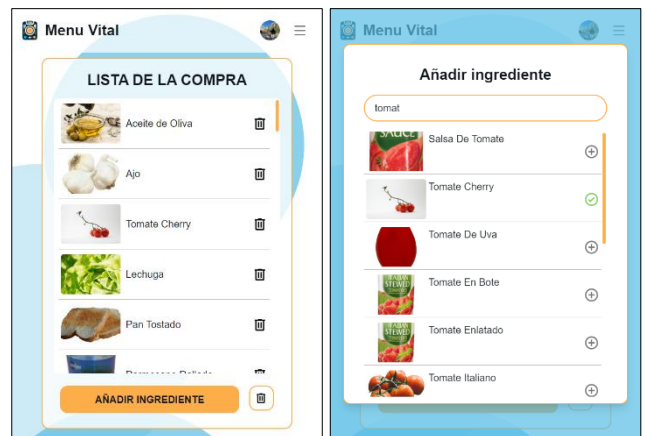
El component de la llista de la compra ha de permetre a l'usuari visualitzar-la de forma intuïtiva així com eliminar o afegir-hi ingredients.

Per a aconseguir-ho s'han creat endpoints per a obtenir, afegir i eliminar ingredients de la llista de la compra de l'usuari. Perquè l'usuari pugui seleccionar els ingredients que vol afegir a la llista, també s'ha creat un "pop up" que apareix quan l'usuari clica al botó d'afegir ingredient. Aquest component s'encarrega d'aconseguir tots els ingredients disponibles mitjançant una petició a un nou endpoint del back-end i és el front-end qui permet a l'usuari buscar entre aquests. S'ha optat per a fer la funció que s'encarrega de filtrar les dades dels ingredients al front-end per a així evitar que s'esgoti la quota de lectura de registres de Google Firebase i evitar pagar una versió superior d'aquest servei per a obtenir més lectures diàries.

Finalment, s'ha decidit millorar l'experiència de l'usuari en afegir o eliminar ingredients de la llista de la compra. Quan una d'aquestes peticions s'executa, es mostra una notificació flotant que desapareix al cap d'uns segons per a indicar el resultat de l'acció que l'usuari ha realitzat. El resultat és el següent:



Il·lustració 10: PopUps informatius al afegir ingredients a la llista de la compra

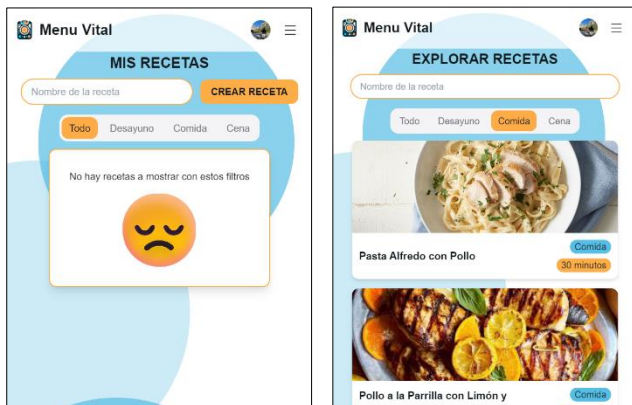


Il·lustració 11: Component llista de la compra i afegir-hi ingredients

7.6 Pàgina “Mis Recetas” i “Explorar Recetas”

Aquestes pàgines són les que permeten a l'usuari visualitzar les receptes de forma ordenada així com filtrar-les pel tipus d'àpat o buscar entre elles. La diferència entre elles és que en la pàgina de “Mis recetas” es mostren només les receptes que aquest usuari ha creat i es pot accedir a la pàgina per a crear una nova recepta. Per altra banda, a la pàgina “Explorar recetas” es mostren totes les receptes existents.

Per a desenvolupar aquestes pàgines s'ha creat els endpoints per a obtenir totes les receptes i les d'un usuari en concret així com un component que és el que renderitza la targeta d'informació de cada recepta. Tant el filtre de buscar per nom de la recepta com el de filtrar per àpat es realitzen al front-end per a evitar sobrecarregar la base de dades amb moltes lectures, ja que aquestes estan limitades. El resultat ha estat el següent:



Il·lustració 12: Pàgines "Mis recetas" i "Explorar recetas"

7.7 Crear receptes

Aquest component és el que permet a l'usuari crear una nova recepta. Està format per un formulari amb diferents camps que l'usuari ha d'omplir per a poder crear una recepta. Aquests camps són els següents: Títol, descripció, ingredients, instruccions, àpat, temps de preparació, dificultat i imatge.

Per a desenvolupar la funcionalitat, s'ha implementat una verificació dels camps del formulari tant al front-end com al back-end on s'envien totes les dades inclosa la imatge de la recepta. És des del back-end d'on s'utilitza el servei de Firebase Storage per a emmagatzemar la imatge als seus servidors i s'obté un link per a accedir a ella des de qualsevol part de l'aplicació.

Per a seleccionar els ingredients de la recepta, s'ha aprofitat el “modal” usat per la llista de la compra i s'ha creat una funció que s'encarrega d'escriure targetes amb els diferents ingredients dintre d'aquest camp del formulari i un botó per eliminar-los individualment. Quan es puja una imatge a l'aplicació abans de crear la recepta, aquesta es mostra per a poder comprovar que és correcta. El resultat d'aquesta pantalla ha estat el següent:

Il·lustració 13: Pàgina de creació de receptes

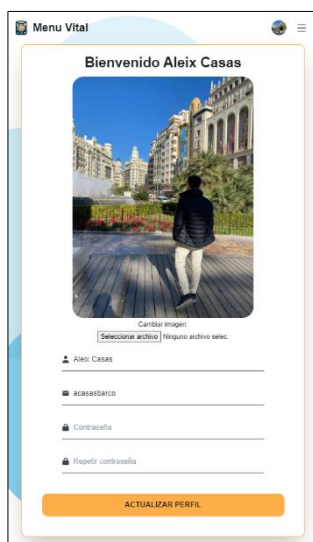
7.8 Detalls de les receptes

Aquesta pàgina és la que permet a l'usuari visualitzar tota la informació d'una recepta de forma ordenada i gràfica. En ella es mostren el títol, la imatge, els ingredients, la dificultat, el temps de preparació, l'àpat al qual pertany i les instruccions per a preparar-la. Per a accedir a aquesta pàgina, es pot fer fent click a la recepta pertinent des del menú d'explorar receptes o de les meves receptes i des de la pàgina de planificació fent click sobre la recepta. Per a desenvolupar aquesta pàgina s'ha creat un endpoint que retorna tota la informació de la recepta que se sol·licita. El resultat d'aquesta pàgina ha estat el següent:

Il·lustració 14: Pàgina de detalls d'una recepta

7.9 Perfil d'usuari

L'última pàgina de l'aplicació és la del perfil de l'usuari, aquesta ens permet canviar la fotografia de perfil o la contrasenya entre altres opcions. Per a desenvolupar-lo s'ha creat un formulari per a actualitzar la informació i s'ha utilitzat el mateix sistema que en la creació de receptes per a poder pujar la imatge de l'usuari. Pel que fa al canvi de contrasenya de l'usuari, aquesta s'envia al back-end de forma segura i posteriorment es fan ús de les funcions de Firebase Authentication per a gestionar les contrasenyes i els usuaris de forma segura i així s'aconsegueix modificar la contrasenya de l'usuari. El resultat d'aquesta pàgina és el següent:



Il·lustració 15: Pàgina del perfil de l'usuari

7.10 Test

Un cop acabades totes les funcionalitats de l'aplicació, ha estat el moment de provar que tot funcionés correctament i, per tant, de realitzar la fase de test. Per a fer-ho s'ha elaborat un test plan per a provar la API que el back-end. En aquest s'han identificat tots els endpoints i s'ha decidit dur a terme proves d'integració en tots ells i així assegurar que l'aplicació funcionés correctament.

Per alguns dels endpoints més importants de l'aplicació com són els de planificació, receptes i alguns d'ingredients, també s'han fet tests unitaris per a així assegurar cobrir gran part de les línies de codi i que es gestionen correctament les excepcions.

Tots aquests tests han seguit una estratègia de caixa negra que consisteixen en assegurar la qualitat del codi enfocant-se en la funcionalitat i el flux de treball sense conèixer els detalls interns del codi.

El resultat d'aquest pla de tests ha estat la creació de 45 casos de test, dividits en integració i unitaris. Pel que fa als unitaris, s'ha aconseguit cobrir un 100% de les línies de codi del controlador de planificació i un 96,4% del de receptes. Pel que fa al d'ingredients, com que només s'han dut a terme tests unitaris d'alguns dels endpoints, el

percentatge de línies cobertes és del 24%. A continuació es poden veure els resultats de l'execució dels tests:

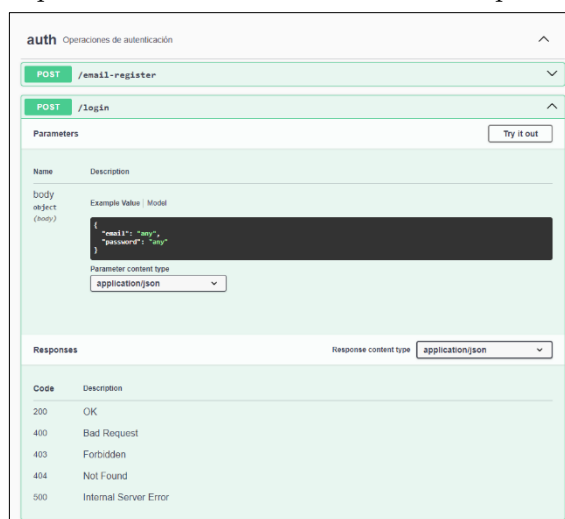
File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	67.33	50	60	70.21	
ingredient.controller.js	24.19	0	20	27.27	...50,54-96,101-112
planning.controller.js	100	100	100	100	
recipe.controller.js	96.42	100	100	96.36	72-73

Test Suites:	8 passed, 8 total
Tests:	45 passed, 45 total
Snapshots:	0 total
Time:	10.957 s, estimated 11 s
Ran all test suites.	

Il·lustració 16: Resultat execució tests

7.11 Documentació de la API

Per a documentar tots els endpoints de la API, s'ha utilitzat Swagger^[13], una eina molt eficaç i utilitzada en aquest àmbit. Aquesta ens permet crear una documentació interactiva i fàcil d'entendre que permet als usuaris visualitzar i provar els endpoints. Aquesta documentació inclou detalls com els mètodes HTTP, paràmetres d'entrada, formats de resposta i descripció de cada un dels endpoints de la API organitzats en les diferents categories. A continuació es pot veure la documentació d'un dels endpoints:



Il·lustració 17: Documentació de l'endpoint de Login generada amb Swagger

La documentació completa amb tots els endpoints es pot veure a <https://menuvital.es/api/docs/>

7.11 Desplegament de l'aplicació

Per a desplegar l'aplicació, es va fer una recerca de diferents serveis que ofereixen servidors al núvol. Afortunadament, es va trobar que la llicència de GitHub Education, de la qual dispo de gràcies a ser estudiant, ofereix 200\$ de crèdit a la plataforma de DigitalOcean^[14] la qual disposa de serveis de hosting al Cloud.

En aquesta plataforma pots llogar servidors amb diferents especificacions variables per a poder escalar el servei en el cas que sigui necessari. Un cop disposava d'un servidor amb una CPU i 1GB de memòria, va ser el moment de configurar-lo per a poder servir l'aplicació que s'ha desenvolupat. Per garantir que el back-end de l'aplicació estigui

sempre actiu i accessible, s'ha fet ús de Nginx^[15] i PM2. Nginx actua com a servidor web i proxy invers, dirigint les sol·licituds entrants cap a l'aplicació correctament. D'altra banda, PM2 és una eina de gestió de processos que assegura que l'aplicació Node.js romanguí en execució contínua, reiniciant-la automàticament en cas de fallada o quan el servidor es reinicia.

Per a fer l'aplicació més accessible als usuaris, s'ha comprat el domini *menuvital.es*, se li ha assignat la direcció IP del servidor i s'han aconseguit els certificats SSL^[16] perquè la pàgina aparegui com a un lloc segur. També s'ha programat la renovació automàtica del certificat SSL mitjançant la configuració del servidor i s'ha redirigit totes les peticions que comencen per “/api” al port 3000 de l'aplicació que és on està escoltant la API que conforma el backend.

Un cop ha estat tot configurat, es pot trobar l'aplicació web en aquesta direcció: <https://menuvital.es/> o escanejant el codi QR^[17] que es mostra a l'Annex 4.

8 CONCLUSIÓ

Tot i que en la fase de planificació es va obviar algunes tasques que s'havien de portar a terme durant el projecte i es va menysprear la dificultat i la duració d'algunes tasques. La totalitat del projecte s'ha pogut realitzar correctament tot i patir alguns retards en diverses fases que s'han anat recuperant quan s'acostava la data de lliurament.

És per això que podem assegurar que no s'ha fet una bona estimació de la duració de cadascuna de les fases del projecte i que, per tant, tot i que no ha estat necessari modificar la planificació, s'hauria d'haver dedicat més temps i esforç a aquesta fase.

Pel que fa a les tasques de desenvolupament, aquestes s'han pogut completar satisfactòriament sense una dificultat molt elevada gràcies al fet que ja es tenien coneixements de les tecnologies que s'han utilitzat per a portar-lo a terme.

En conclusió, es pot assegurar que el projecte s'ha desenvolupat satisfactòriament, ja que s'ha complert la totalitat dels objectius que es van plantejar a l'inici del projecte.

BIBLIOGRAFIA

- [1] MyFitnessPal | MyFitnessPal. (s. f.). <https://www.myfitnesspal.com/es>
- [2] Yummly: personalized recipe recommendations and search. (n.d.). Yummly. <https://www.yummly.com/>
- [3] Paprika Recipe Manager for iOS, Mac, Android, and Windows. (n.d.). <https://www.paprikaapp.com/>
- [4] Mealime - meal planning app for healthy eating - Get it for free today! (2023, 30 setembre). Mealime. <https://www.mealime.com/>
- [5] Hotmart. (2023, 2 octubre). Metodología SCRUM: Qué es y cómo usar esta herramienta en tus proyectos. Blog Hotmart. <https://hotmart.com/es/blog/scrum>
- [6] Simões, C. (2020, July 14). MoSCoW. ¿Qué es y cómo priorizar en el desarrollo de tu aplicación? Blog ITDO - Agencia De Desarrollo Web, APPs Y Marketing En Barcelona. <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/>
- [7] Novoseltseva, E. (2023, 20 junio). Priorización de requerimientos de software | ApiumHub. Apiumhub. <https://apiumhub.com/es/tech-blog-barcelona/priorizacion-requerimientos-software/>
- [8] Amazon DynamoDB vs. Firebase Realtime Database vs. MongoDB comparison. (s. f.). <https://db-engines.com/en/system/Amazon+DynamoDB%3bFirebase+Realtime+Database%3bMongoDB>
- [9] Draw.io - free flowchart maker and diagrams online. (s. f.-b). <https://app.diagrams.net/>
- [10] Nutritionix. (s. f.). NutritionIX. <https://www.nutritionix.com/es/database/common-foods>
- [11] Easy Web Data Scraper. (s. f.). Chrome Web Store. <https://chrome.google.com/webstore/detail/easy-web-data-scraper/mndkmbnkepbhdkhlofdcfmgflbjggnl>
- [12] ChatGPT. (n.d.). <https://chat.openai.com/>
- [13] API documentation & design tools for teams | Swagger. (n.d.). <https://swagger.io/>
- [14] DigitalOcean | Cloud hosting for builders. (n.d.). <https://www.digitalocean.com/>
- [15] Glass, E. (2020, May 21). Cómo instalar Nginx en Ubuntu 20.04. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04-es>
- [16] Boucheron, B. (2020, June 11). Cómo proteger Nginx con Let's Encrypt en Ubuntu 20.04. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04-es>
- [17] Login - QR code Generator PRO. (s. f.). <https://app.qr-code-generator.com/>

APÈNDIX

A1. LLISTAT DE REQUISITS

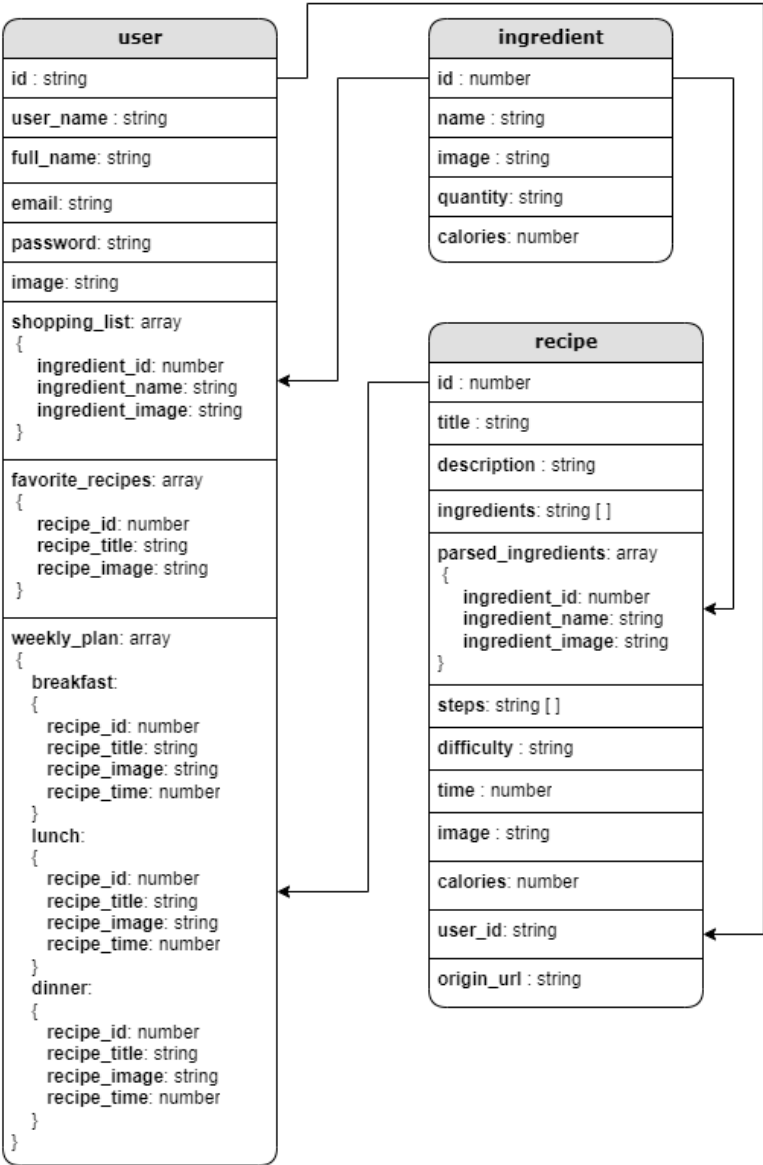
TIPUS	REQUISITS FUNCIONALS	PRIORITAT
RF-1	L'usuari ha de poder registrar-se a la Web App	M
RF-1.1	L'usuari ha de poder modificar el seu perfil	W
RF-2	L'usuari ha de poder iniciar sessió a la Web App	M
RF-2.1	L'usuari ha de poder tancar sessió a la Web App	S
RF-3	L'usuari ha de poder recuperar la seva contrasenya	S
RF-4	L'usuari ha de poder crear una planificació inicial dels àpats de la setmana	M
RF-4.1	L'usuari ha de poder editar aquesta planificació en qualsevol moment	S
RF-4.2	L'usuari ha de poder visualitzar la planificació setmanal en format de taula, ja sigui per columnes o per files (2 formes diferents)	S
RF-4.3	L'usuari ha de poder calcular les calories de cada àpat planificat	C
RF-4.4	L'usuari ha de poder buscar receptes a la Web App	S
RF-5	L'usuari ha de poder generar la llista de la compra per als àpats d'aquella setmana de forma automàtica	M
RF-5.1	L'usuari ha de poder visualitzar la llista de la compra en qualsevol moment	M
RF-5.2	L'usuari ha de poder afegir i eliminar ingredients d'aquesta llista de la compra	S
RF-5.3	L'usuari ha de poder buscar ingredients per a afegir a la llista de la compra i veure el seu icona o imatge	M
RF-5.4	L'usuari ha de poder buidar la llista de la compra	S
RF-6	L'usuari ha de poder crear les seves pròpies receptes	M
RF-6.1	L'usuari ha de poder afegir una imatge per a les seves receptes	M
RF-6.2	L'usuari ha de poder afegir una estimació de les calories de cada una de les seves receptes	C
RF-7	L'usuari ha de poder explorar receptes d'altres usuaris	C
RF-8	L'usuari ha de poder afegir i eliminar receptes a la llista dels seus preferits	C
RF-9	L'usuari ha de poder veure els detalls de les receptes	S

TIPUS	REQUISITS NO FUNCIONALS	PRIORITAT
RNF-1 Es-tètic	La interfície d'usuari ha de ser intuïtiva per a l'usuari	S
RNF-2 Rendiment	El temps de càrrega inicial de la Web App ha de ser inferior a 2 segons	S
RNF-3 Seguretat	Les contrasenyes dels usuaris han d'emmagatzemar-se de forma segura	M
RNF-4 Es-tètic	L'aplicació ha d'estar dissenyada per a dispositius mòbils però també ha de ser suportada per a ordinadors	S
RNF-5 Tèc-nic	El back-end haurà de ser implementat amb NodeJS i el framework Express	S
RNF-6 Tèc-nic	El front-end haurà de ser implementat amb ReactJS i TailwindCSS	S

A2. COMPARATIVA DE SERVEIS DE BASES DE DADES

	SERVEI DE BASE DE DADES				
	Firestore	AWS	DynamoDB	MongoDB Atlas	Supabase
Base de Dades	Firestore (NoSQL)	DynamoDB (NoSQL)	DynamoDB (NoSQL)	MongoDB (NoSQL)	PostgreSQL (Relacional)
Autenticació	Suportada	Requerirà configuració addicional	Requerirà implementació personalitzada	No és part de la plataforma	Suportada
Emmagatzematge	Firestore Storage	Requerirà configuració addicional	Requerirà configuració addicional	No inclòs	Requereix configuració addicional
Escalabilitat	Automàtica i fàcil	Escalable i personalitzable	Escalable i personalitzable	Escalable	Escalable
Rendiment	Temps de càrrega ràpid	Pot ser ràpid, requerirà configuració addicional	Pot ser ràpid, requerirà configuració addicional	Pot ser ràpid	Pot ser ràpid
Limitacions versió gratuïta	100 connexions simultànies 50.000 lectures i 20.000 escriptures diàries	No té restriccions de connexions simultànies 25 lectures i 25 escriptures per segon	No té restriccions de connexions simultànies 25 lectures i 25 escriptures per segon	500 connexions simultànies per No té restriccions de lectures o escriptures	No hi ha restriccions de connexions simultànies o de lectures i escriptures

A3. ESQUEMA DE L'ESTRUCTURA DE LA BASE DE DADES



A4. CODI QR DE L'APLICACIÓ

