

Utilitat web que suggereix receptes a partir d'ingredients

Memòria de Projecte Final de Grau
Grau Multimèdia
Especialitat d'Enginyeria Web

Autor: Marc Tallón Ortega

Consultor: Ignasi Lorente Puchades
Professor: Carlos Casado Martínez

07 de gener del 2019



Aquest document està subjecte a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

El codi font fruit d'aquest treball està subjecte a una llicència [GNU General Public License](https://www.gnu.org/licenses/gpl-3.0.html)

Resum

Aquest projecte sorgeix com a resposta a un problema que qualsevol persona s'ha trobat eventualment: voler cuinar seguint una recepta i trobar-se sense els ingredients necessaris. I és que, sovint, els llibres de cuina estan plens de receptes amb ingredients exòtics o que, simplement, un no compta amb ells en el moment oportú. Caldria un llibre de receptes que s'adaptés als ingredients que es tenen en aquell moment, fet poc probable.

La tecnologia, però, ens permet posar remei a aquesta situació. Aquest projecte fa ús de les bases de dades, la tecnologia web i l'accés a internet per millorar la qualitat de vida dels usuaris. Mitjançant una consulta des de l'aplicació amb els ingredients disponibles, l'usuari rep un llistat de receptes que contenen aquests ingredients. Igualment, les pot filtrar per tipus de menjar o ordenar-les segons uns criteris.

L'objectiu principal d'aquesta aplicació és fer-li més còmoda la vida als usuaris i millorar els seus hàbits d'alimentació.

Abstract

This project arises as a response to a problem that anyone has ever encountered: wanting to cook a recipe but not having the necessary ingredients. And, often, cookbooks are full of recipes with exotic ingredients or, simply, one does not count on them at the right time. There should be a recipe book that adapts to the ingredients that are available at that time, which is unlikely.

Technology, however, allows us to remedy this situation. This project uses databases, web technology and internet access to improve the users life quality. Using the application to query the available ingredients, the user receives a list of recipes that contain these ingredients. You can also filter them by type of food or sort them according to criteria.

The main objective of this application is to make users life easier and improve their eating habits.

Índex

1. Introducció/Prefaci.....	07
2. Objectius.....	08
2.1 Principals.....	08
2.2 Secundaris.....	08
3. Continguts.....	09
4. Metodologia.....	10
5. Arquitectura de l'aplicació/sistema/servei.....	11
6. Plataforma de desenvolupament.....	15
7. Planificació.....	16
8. Procés de treball/desenvolupament.....	17
8.1 Primera entrega (2 de setembre).....	17
8.2 Segona entrega (21 d'octubre).....	17
8.3 Tercera entrega (9 de desembre).....	18
8.4 Entrega final (7 de gener).....	18
10. Diagrames UML.....	19
11. Prototips.....	21
11.1 Lo-Fi.....	21
11.2 Hi-Fi.....	23
14. Tests.....	25
21. Instruccions d'instal·lació/implantació.....	27
22. Instruccions d'ús.....	29
24. Projecció a futur.....	30
25. Pressupost.....	31
28. Conclusió/-ns.....	32
Annex 1. Lliurables del projecte.....	33
Annex 2. Bibliografia.....	34
Annex 3. Vita.....	35

Figures i taules

Llistat d'imatges, taules, gràfics, diagrames, etc., numerades, amb títols i les pàgines on apareixen.

Índex de figures

Figura 1: Arquitectura Client-Servidor.....	11
Figura 2: Patró Clean Architecture.....	12
Figura 3: Patró MVC.....	13
Figura 4: Diagrama de Gantt.....	16
Figura 5: Diagrama de Classes.....	19
Figura 6: Diagrama de casos d'ús.....	19
Figura 7: Diagrama de casos d'activitat.....	20
Figura 8: Wireframes del navegador web.....	21
Figura 9: Wireframes de la versió mòbil.....	22
Figura 10: Mockups per a navegador web.....	23
Figura 11: Mockups per a dispositius mòbil.....	24

Índex de taules

Taula 1: User Acceptance Tests.....	26
Taula 2: Pressupost del projecte.....	31

1. Introducció/Prefaci

Cada cop més, l'accelerat ritme de vida ens aboca a invertir més i més temps en les nostres feines i altres obligacions. A moltes persones se'ls hi fa difícil disposar de temps fins i tot per les coses més bàsiques, i això es tradueix en pèrdua de qualitat de vida. Afortunadament, les noves tecnologies han permès en els darrers anys la proliferació d'aplicacions de tota mena que ajuden a automatitzar certes tasques i a fer la vida de moltes persones més senzilla. La majoria d'aplicacions, ja siguin web o mòbils, pensades per ajudar a organitzar tasques, enregistrar dades sobre el rendiment esportiu, controlar el pes o monitoritzar el somni de l'usuari tenen un impacte positiu en la seva qualitat de vida.

Una problemàtica molt habitual en persones amb poc temps és la mala alimentació. Per a moltes persones el fet de no disposar d'ingredients suficients, la indecisió a l'hora de decidir què cuinar, o una barreja de les dues els acaba abocant a haver de recórrer a plats precuinats, a menjar fora de casa o alimentar-se de manera poc saludable. Sovint el problema no és tant la mandra de posar-s'hi a cuinar com d'haver de decidir què cuinar o haver de sortir a comprar els ingredients necessaris per preparar el plat que es té al cap.

Aquest treball pretén respondre a aquestes necessitats i, per aconseguir-ho, es proposa la creació d'una aplicació web pensada per ajudar els usuaris a escollir una recepta a partir dels ingredients dels què disposa en el moment de fer la consulta. El resultat de la consulta serà una llista de receptes ordenades per rellevància que ajudaran a l'usuari a prendre la decisió que més li convingui i estalviar-se temps i mal de caps.

2. Objectius

2.1 Principals

- Desenvolupar una aplicació servidor amb connexió a base de dades.
- Desenvolupar una aplicació client amb tecnologia web.
- Desenvolupar una aplicació client amb tecnologia mòbil (híbrida).
- Aplicar conceptes de disseny UI/UX a les aplicacions client.

2.2 Secundaris

- Desplegar el projecte en una màquina accessible per internet.
- Posar en pràctica la programació orientada a objectes i els patrons de disseny.
- Posar en pràctica patrons d'arquitectura.

3. Continguts

El Treball Final de Grau està format pels següents continguts:

- La present memòria de treball en què es defineixen:
 - Les metodologies de treball
 - L'arquitectura del sistema d'informació
 - Les tecnologies emprades en la implementació del producte i el procés de treball
 - Els documents resultat de l'anàlisi tècnic i del disseny gràfic
 - Els documents de seguiment del projecte (planificació, pressupost, UATs)
 - Les guies d'ús i implantació del projecte
- El producte resultant, dividit en:
 - Aplicació servidor
 - Aplicació client
- El disseny de les aplicacions client:
 - Logotip, paleta de colors i tipografia
 - Wireframes
 - Mockups

4. Metodologia

L'execució del projecte es fonamenta en l'aplicació d'unes metodologies de treball que permeten assegurar l'èxit del mateix, el bon funcionament del producte resultant a l'hora que donen flexibilitat davant possibles factors de risc. Algunes d'aquestes metodologies són:

Desenvolupament incremental iteratiu:

El desenvolupament iteratiu és un procès de desenvolupament de software molt aplicat en empreses modernes que busquen una metodologia més flexible i eficient que les habituals metodologies seqüencials o en cascada. La flexibilitat s'aconsegueix dividint les tasques en petites etapes iteratives, de manera que és més fàcil detectar errors en el disseny i solucionar-los abans que es propaguin a la iteració següent.

Una de les concrecions d'aquest mètode de desenvolupament de software són les metodologies àgils (agile).

Agile:

Com en qualsevol metodologia iterativa, els processos agile estan formats per petites iteracions que no pretenen obtenir un producte totalment acabat, sinó un producte funcional, sense errors i que aporta valor. Cadascuna de les iteracions d'un desenvolupament àgil consta de: planificació, anàlisi, disseny, implementació, testing i documentació.

Dins les metodologies agile una de les més esteses actualment és Scrum, que és la que segueix aquest projecte.

Scrum:

Scrum és una de les metodologies àgils més emprades en el desenvolupament de software actualment. Algunes de les seves característiques són:

- El producte es desenvolupa de manera incremental, enlloc de definir perfectament com serà el producte acabat abans de començar la fase d'implementació.
- Posa l'èmfasi en el coneixement, l'autogestió dels membres de l'equip i la comunicació abans que en els processos.
- Permet el flux de treball en paral·lel en comptes de seqüencial.

En aquest projecte no es poden aplicar tots els conceptes de la metodologia Scrum perquè l'equip de treball està format per una sola persona, que fa de Scrum master, Product owner i developer team a l'hora. Però, en la mesura del possible, el seguiment de les tasques a realitzar es farà segons aquesta metodologia, agrupant les funcionalitats en sprints, amb històries d'usuari i organitzant les tasques en tres categories: backlog, in progress i done.

TDD:

El desenvolupament de les aplicacions es farà seguint el mètode Test Driven Development (TDD) o desenvolupament dirigit per tests. Aquest mètode de treball proposa la implementació primer de tests per assegurar la qualitat del codi resultant. Primer s'escriu un test unitari com a resposta a un cas d'ús o una funcionalitat concreta que es vol desenvolupar. S'executa el test que fallarà perquè encara no s'ha implementat el codi. Iterativament s'afegeix codi fins que el test s'executa sense errors.

D'aquesta manera la qualitat del codi queda assegurada i, a mida que es van afegint funcionalitats, s'evita la creació de bugs com a dany col·lateral.

Control de versions amb Git:

Les versions produïdes durant les diverses iteracions es persistiran mitjançant una eina de gestió de versions, en aquest cas Git. D'aquesta manera s'assegura que no es perdrà codi i que sempre es podrà tornar a un estat anterior en cas que hi hagi algun error important.

5. Arquitectura de l'aplicació/sistema/servei

Arquitectura del sistema:

El sistema està basat en l'arquitectura client-servidor, en què una aplicació client envia peticions mitjançant el protocol HTTP a una aplicació servidor que, un cop processada, li retorna la resposta. A continuació podem veure un esquema del seu funcionament:

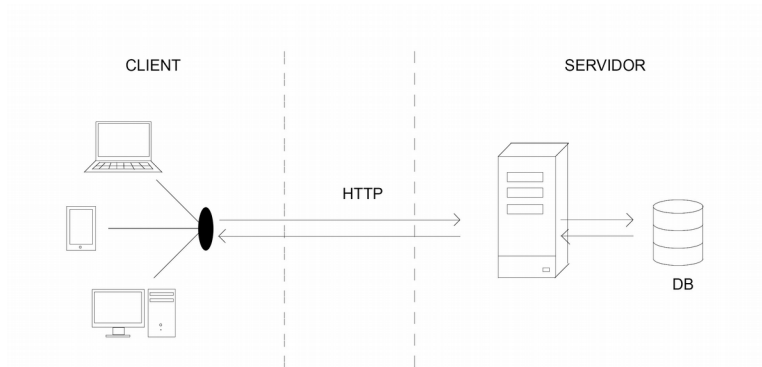


Figura 1: Arquitectura Client-Servidor

L'aplicació servidor o Backend:

Clean Architecture

El servidor està estructurat seguint el patró d'arquitectura del software Clean Architecture, que defineix la separació d'una aplicació per capes segons les seves responsabilitats, a on les capes més interiors són agnòstiques respecte de les més exteriors (en desconeixen la seva implementació). En les capes més exteriors se situen les implementacions específiques segons les tecnologies escollides, de manera que el sistema és més flexible a l'hora de fer canvis i, també, més escalable.

Aquest seria un esquema de la Clean Architecture:

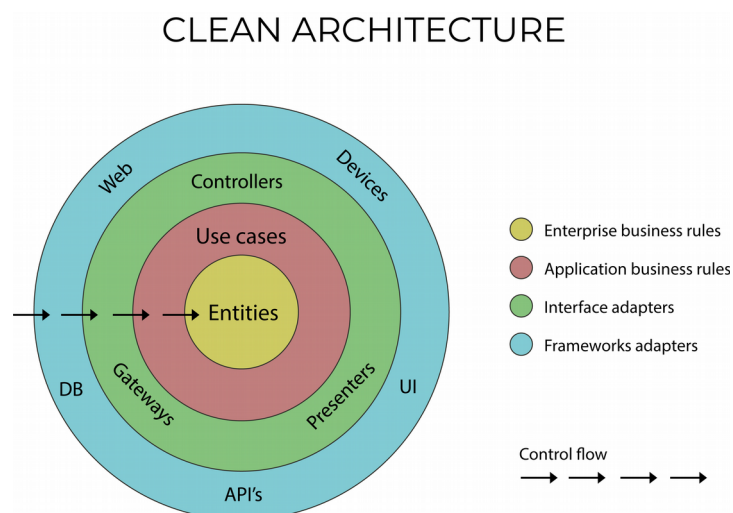


Figura 2: Patró Clean Architecture

Clean Architecture aplicat a aquest projecte

En concret, el backend d'aquest projecte està dividit en les següents 4 capes, de més interior a més exterior:

Core

Aquest mòdul conté les entitats de domini. Les entitats de domini són aquelles que modelen objectes del domini de l'aplicació, com les receptes culinàries en aquest projecte. Són totalment independents de la resta del software.

Usecases

Aquest mòdul conté la lògica de negoci de l'aplicació, que determina les seves funcionalitats bàsiques. També és l'encarregat de definir una API amb els punts d'entrada que rebrà les peticions HTTP. Aquests punts d'entrada depenen només de les entitats definides en el mòdul core.

Dataproviders

Aquest mòdul conté entitats de base de dades i repositoris d'accés a dades. Les entitats modelen cadascun dels objectes de la base de dades fent servir el framework JPA. Els repositoris implementen els punts d'entrada definits en el mòdul Usecases, accedint a la base de dades per llegir-la o modificar-la mitjançant el framework Spring Data. Les entitats que modelen objectes de la base de dades són traduïdes (mapping) a les corresponents entitats de domini que hi han al mòdul Core. Per tant, aquest mòdul depèn del mòdul Usecases i del mòdul Core.

Delivery

Aquest mòdul és l'encarregat de connectar tota l'aplicació i controlar-ne la seva execució. Defineix objectes de transferència (DTO's) per a les respostes i peticions que s'atendran i serviran des del servidor via API REST. Els DTO's són traduïts a les corresponents entitats del mòdul Core. Fa ús dels frameworks Spring Boot i Spring Web i té dependències sobre el mòdul Usecases i sobre el mòdul Core.

L'aplicació client o Frontend:

MVC

L'aplicació client és una solució híbrida basada en el framework de javascript Ionic, que a l'hora fa ús del framework Angular. Angular utilitza una arquitectura Model-Vista-Controlador (MVC), una arquitectura que separa les dades d'una aplicació, la interfície d'usuari i la lògica de control. D'aquesta manera s'aconsegueix assegurar la independència entre la vista i el model, reduint l'impacte dels canvis en el disseny de la interfície gràfica.

El model conté les entitats de domini, la lògica de negoci del domini i els repositoris de persistència. La vista presenta de manera atractiva i intuïtiva el contingut de l'aplicació i els mecanismes d'interacció. El controlador fa de connector i gestiona el flux d'informació entre les dues capes anteriors. També s'ocupa de traduir la informació als formats necessaris.

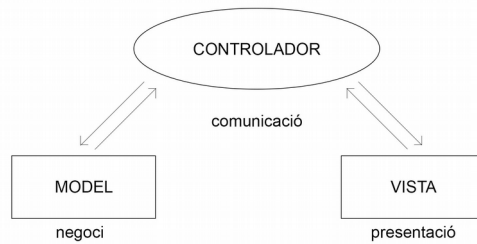


Figura 3: Patró MVC

Component-based architecture

L'estructura del frontend d'aquest projecte, seguint la guia d'estil dels projectes Angular, organitza el codi per components (Component-based architecture). Les architectures basades en components agrupen els elements amb funcionalitats similars, de manera que asseguren l'encapsulació i el desacoblament. Aquest enfocament permet, sobretot, la reutilització dels components en una mateixa aplicació. Un exemple de component podria ser la barra de navegació d'una web, que acostuma a ser comuna a tota l'aplicació i, per tant, es pot separar en un paquet que serà reaprofitat allà on calgui.

MVC i Component-based architecture en aquest projecte

L'aplicació client d'Aquest projecte segueix una estructura basada en components i, per tant, contindrà un mòdul per a cada element de l'aplicació que tingui funcionalitats similars. A més, dins de cada mòdul, el component quedarà dividit en un arxiu typescript que farà de controlador i dos arxius html i css que definiran les vistes. Els models i els repositoris quedaran en un mòdul a banda que serà compartit per tota l'aplicació. A tall d'exemple, aquesta seria l'estructura de l'aplicació per al component *recipe*:

`src/app/recipe/recipe.component.ts`

Arxiu de tipus Typescript que fa de controlador.

`src/app/recipe/recipe.component.html`

Arxiu HTML que defineix l'estructura de la vista.

`src/app/recipe/recipe.component.scss`

Arxiu SCSS que defineix el disseny de la vista.

`src/shared/models/recipe.ts`

Arxiu de tipus Typescript que representa l'entitat de domini *recipe*.

`src/shared/providers/recipe.provider.ts`

Arxiu de tipus Typescript que s'encarrega de fer les peticions al servidor i traduir les respostes a l'entitat corresponent.

La base de dades:

NoSQL

La base de dades escollida per a la capa de persistència del servidor és una base de dades NoSQL, concretament Elasticsearch. Les bases de dades NoSQL no emmagatzemen les dades en forma de taules amb files indexades com en els sistemes de bases de dades relacionals. En el cas concret d'Elasticsearch, les dades s'emmagatzemen en documents amb format JSON. Això fa possible que les cerques de text siguin molt flexibles i àgils.

6. Plataforma de desenvolupament

Els recursos tecnològics emprats en aquest projecte són els següents:

Recursos de software:

Linux: El sistema operatiu linux és l'escollit per a fer tot el desenvolupament.

Intellij idea i webstorm: Els IDE's utilitzats per al desenvolupament backend i frontend són, respectivament, IDEA i Webstorm, d'Intellij.

Git i Github: Les versions es gestionen amb l'eina de control de versions Git i el repositori Github.

Docker: La creació de l'entorn de desenvolupament es gestiona i automatitza mitjançant els contenidors de l'eina Docker.

Apache Tomcat: El servidor d'aplicacions escollit per a l'aplicació servidor és Apache Tomcat

Maven: La gestió de dependències es fa a través de Maven

Spring: Amb Spring s'agilitza la creació i execució d'una API REST

JPA: El framework Java Persistence API s'utilitza per a la comunicació amb base de dades

Kotlin, JDK: El llenguatge de programació escollit és Kotlin, que s'executa sobre el JDK

Elasticsearch i Kibana: La base de dades escollida és Elasticsearch. Es fa servir Kibana com a eina gràfica només per ajudar durant el desenvolupament.

Node: El framework de javascript Node serveix per a la gestió de dependències de l'aplicació client

Ionic i Angular: El framework Ionic que, a l'hora fa servir Angular, és l'escollit per al desenvolupament de l'aplicació client.

Typescript: El llenguatge de programació escollit és Typescript, que és el que fa servir Angular i, per tant, Ionic.

Recursos de hardware:

Raspberry: Es preveu la possible necessitat de desplegar l'aplicació servidor en una Raspberry Pi.

PC: El desenvolupament es fa a través d'un PC amb linux. També servirà per a fer testing del client web.

Mac: El desplegament de l'aplicació client IOS es fa des d'un ordinador Mac.

Smartphone android: El testing del client mòbil es fa en un dispositiu amb SO Android

Smartphone ios: El testing del client mòbil es fa en un dispositiu amb IOS

7. Planificació

Diagrama de Gantt amb la planificació detallada del projecte.

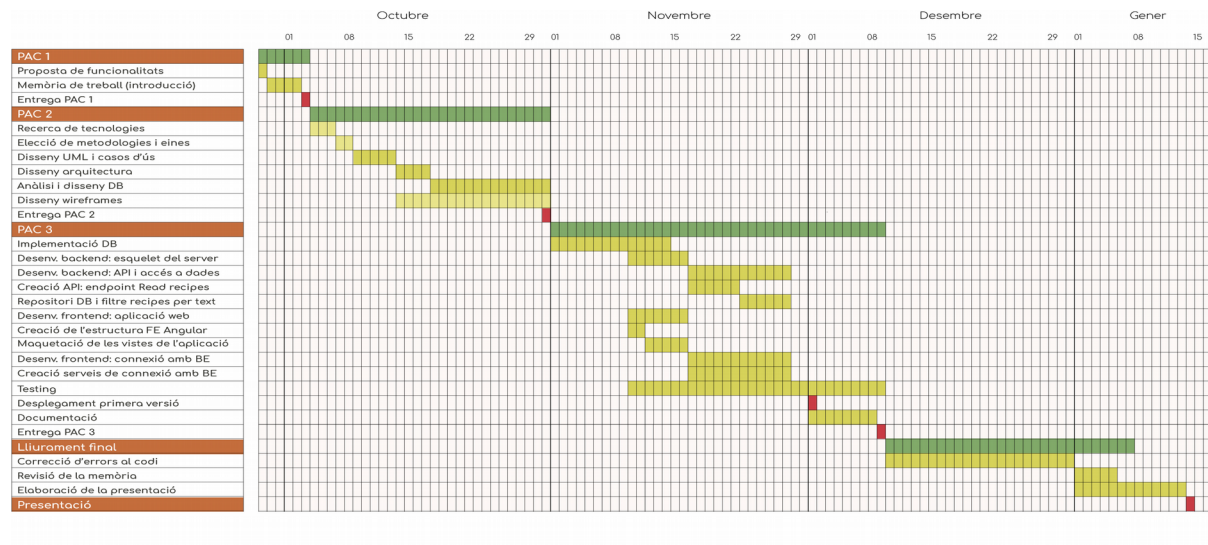


Figura 4: Diagrama de Gantt

7.1 Dates clau

- 02 de setembre: entrega de l'estructura de la memòria, funcionalitats i planificació del projecte.
- 31 d'octubre: entrega dels dissenys de DB, arquitectura, UML i wireframes.
- 09 de desembre: desplegament de la primera versió del projecte.
- 07 de gener: entrega de la memòria acabada.
- 14 de gener: presentació del treball.

8. Procés de treball/desenvolupament

A continuació es relaciona el procés de treball d'aquest projecte estructurat de manera cronològica per entregues i fases:

8.1 Primera entrega (2 de setembre)

Definició del projecte

Durant aquesta fase s'identifiquen els problemes a resoldre i es defineix una possible implementació de la solució. D'aquesta manera s'obté una idea general del projecte, els objectius a assolir i els criteris d'acceptació que permeten identificar el projecte com un cas d'èxit.

En aquest treball l'objectiu general del projecte ha estat identificat com la realització d'una aplicació web i mòbil per ajudar a escollir receptes que coincideixin amb els ingredients de què disposa l'usuari. La funcionalitat principal de l'aplicació és la cerca de receptes filtrades i ordenades per ingredients.

Durant aquesta primera fase, també s'ha definit l'estructura de la memòria del treball i s'ha fet una feina de recerca de les possibles tecnologies a emprar en el desenvolupament del projecte.

8.2 Segona entrega (21 d'octubre)

Disseny UI

Abans de posar-s'hi a treballar en el disseny i l'estructura de l'aplicació s'ha volgut començar a definir les principals funcionalitats de les aplicacions clients. Aquest enfocament ajuda a identificar funcionalitats que altrament podrien passar desapercebudes.

Durant la fase de disseny UI s'han creat els wireframes que representa l'estructura de cadascuna de les vistes de les aplicacions client, així com les accions que podrà realitzar l'usuari a través d'elles.

Després s'ha realitzat un prototip amb Photoshop de les vistes amb la major fidelitat possible. El procés de creació dels mokups s'ha desenvolupat de manera iterativa, demanant opinió a un grup de possibles usuaris en cada iteració, fins a obtenir el resultat final, disponible més endavant en aquest treball.

Disseny DB i UML

S'ha considerat oportú començar per la definició de la base de dades i el model de domini de l'aplicació.

El primer pas d'aquesta fase ha estat la recerca d'una base de dades adient per a la realització de consultes no indexades complexes i eficients. El següent pas ha estat la identificació de l'estructura de dades més oportuna per a l'emmagatzematge de la informació que representa els objectes de domini.

Un cop identificada l'estructura de la informació, s'ha definit el diagrama de classes i l'esquema de base de dades.

Disseny de l'arquitectura de l'aplicació

En aquesta etapa s'ha acabat de fer recerca sobre les tecnologies i entorns més adients per al desenvolupament de l'aplicació, així com els patrons d'arquitectura a seguir.

Un cop feta la recerca s'ha fet una prova de concepte que recull l'estructura principal de l'arquitectura tant per l'aplicació servidor com per les aplicacions client.

8.3 Tercera entrega (9 de desembre)

Definició i configuració de sistemes

En aquesta primera fase de la implementació del codi del projecte s'ha fet la configuració dels sistemes informàtics i l'entorn de desenvolupament, instal·lant totes les eines necessàries per les fases següents.

Entre d'altres tasques, s'ha creat un projecte amb una estructura base per a cada aplicació (client i servidor), s'ha configurat una eina de contenidors (Docker) amb tot el programari necessari per al desenvolupament i s'ha creat un repositori a Github per allotjar les diferents versions del codi.

Desenvolupament del servidor

Un cop configurat l'entorn de treball i creada l'estructura base del servidor, s'han anat desenvolupant les funcionalitats necessàries per assolir els requeriments. En aquesta fase s'ha configurat el servidor, l'accés a la base de dades, la publicació d'uns punts d'entrada via API per a cada cas d'ús, i altres funcionalitats secundàries, com la publicació de l'accés a recursos (imatges de les receptes) del servidor.

Tota aquesta fase de desenvolupament s'ha fet tenint present la metodologia TDD, és a dir, desenvolupant tests unitaris que donen resposta a una funcionalitat encara no implementada per a, acte seguit, implementar-la i fer que els tests passin.

Desenvolupament de les aplicacions client

El desenvolupament de les aplicacions client ha passat per dues fases principalment:

- 1 – La implementació de les vistes segons els mockups generats en la fase de disseny UI.
- 2 – La connexió de la lògica de l'aplicació amb les dades del servidor via API.

La implementació de les vistes s'ha fet de manera iterativa i primer per mòbils. En aplicar els dissenys als navegadors webs la planificació s'ha desviat lleugerament d'hores, però sense conseqüències pel projecte ja que en fases prèvies hi va haver superàvit.

8.4 Entrega final (7 de gener)

Testing i Quality Assurance

En aquesta última fase s'ha iterat sobre les funcionalitats ja implementades per solucionar els errors que s'han anat trobant i que s'han documentat en el document UAT (User Acceptance Testing)

Finalització de la Memòria de Treball i presentació

La resta del temps d'aquesta darrera fase s'ha dedicat a acabar la present Memòria de Treball, així com preparar els documents que serviran per a la seva presentació

9. Diagrames UML

Diagrama de classes

El diagrama de classes de l'aplicació és molt senzill perquè només consta d'una classe principal que modela la recepta i una auxiliar que defineix la imatge de la mateixa:

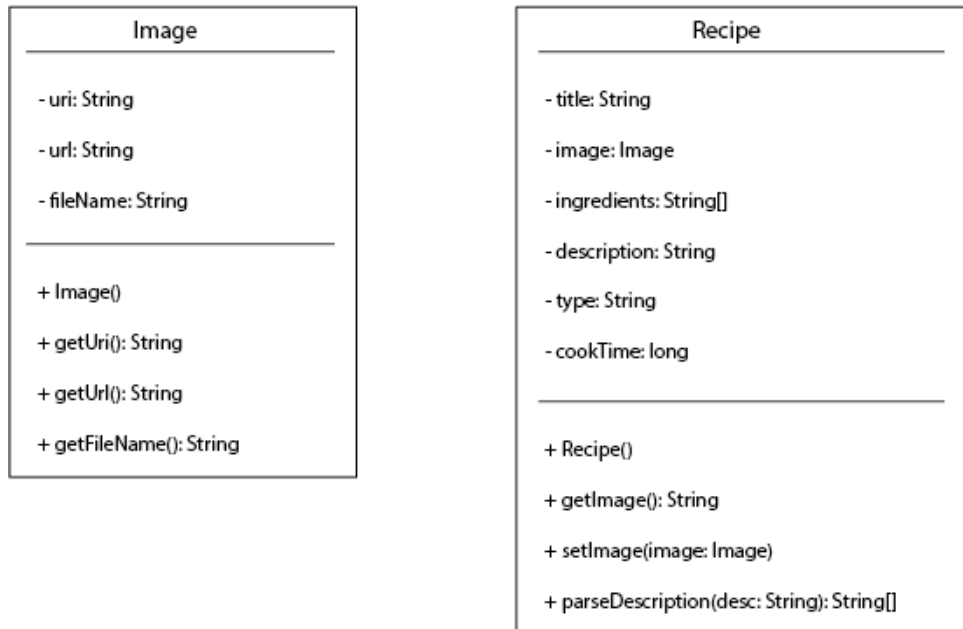


Figura 5: Diagrama de classes

Diagrama casos d'ús:

L'únic actor d'aquest sistema informàtic és l'usuari de l'aplicació client. I l'únic cas d'ús que pot fer és buscar receptes, com es mostra en el següent diagrama:

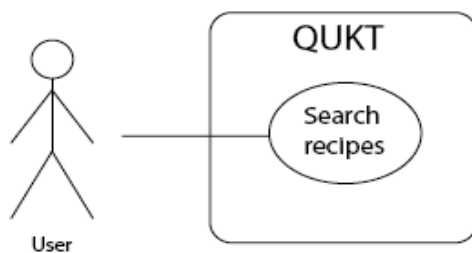


Figura 6: Diagrama de casos d'ús

Diagrama activitat:

Aquest és el diagrama d'activitat de l'aplicació:

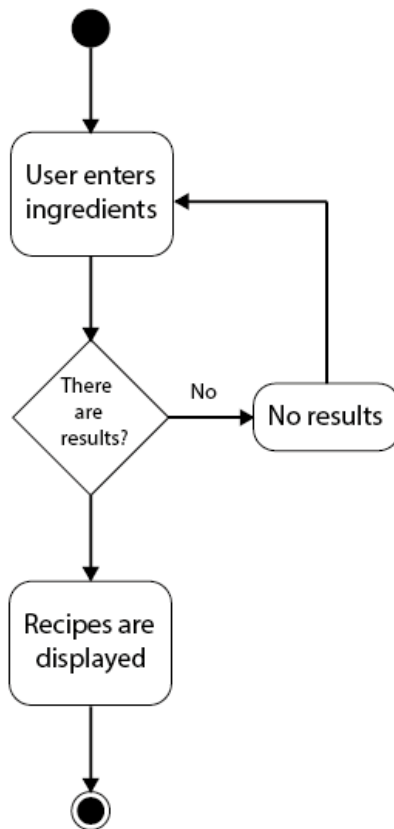


Figura 7: Diagrama d'activitat

10. Prototips

Esbossos, croquis, models, prototips, etc., creats durant el procés de treball, incloent especialment:

10.1 Prototips Lo-Fi

Wireframes per a navegador web

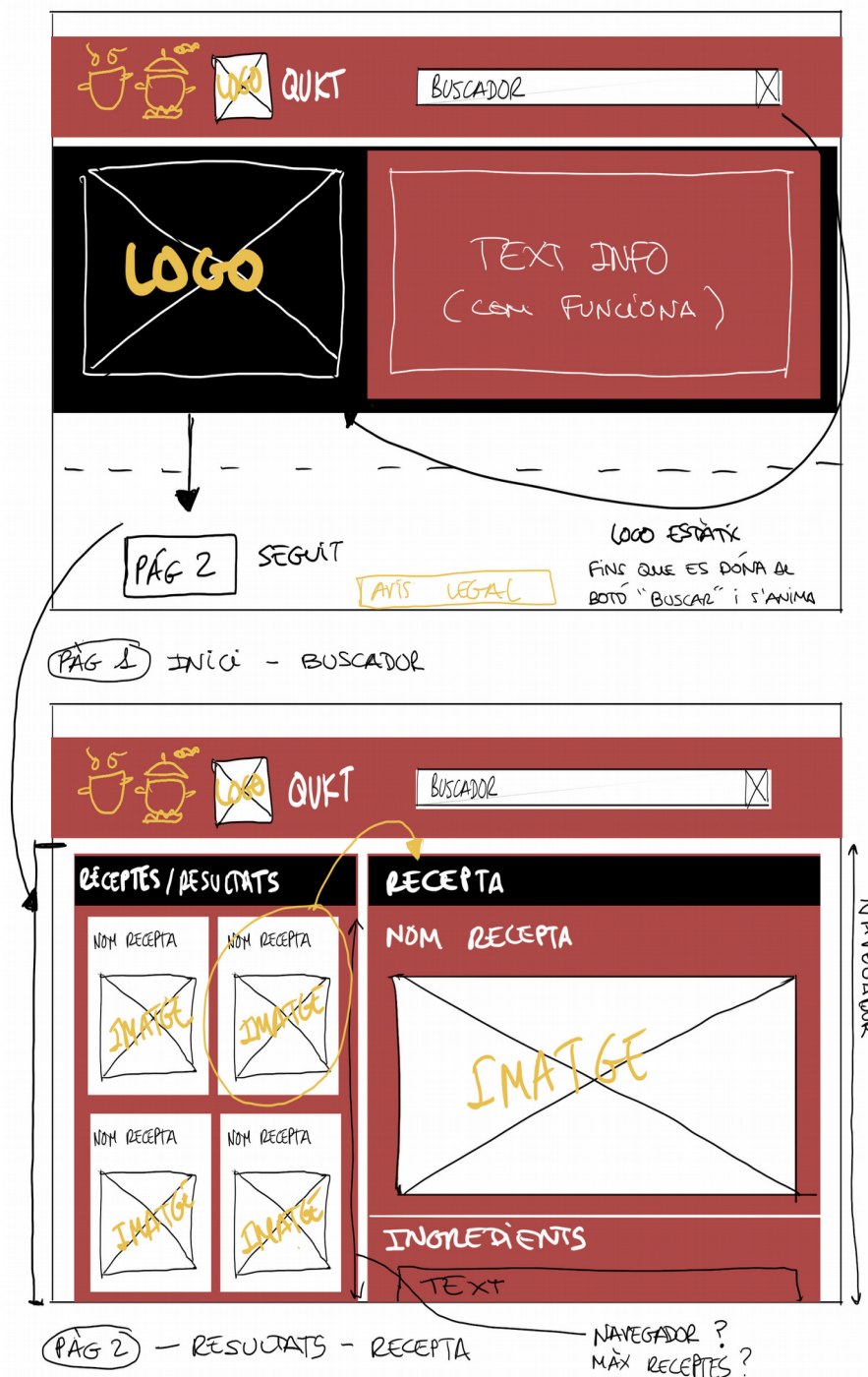


Figura 8: Wireframes del navegador web

Wireframes per a dispositius mòbils

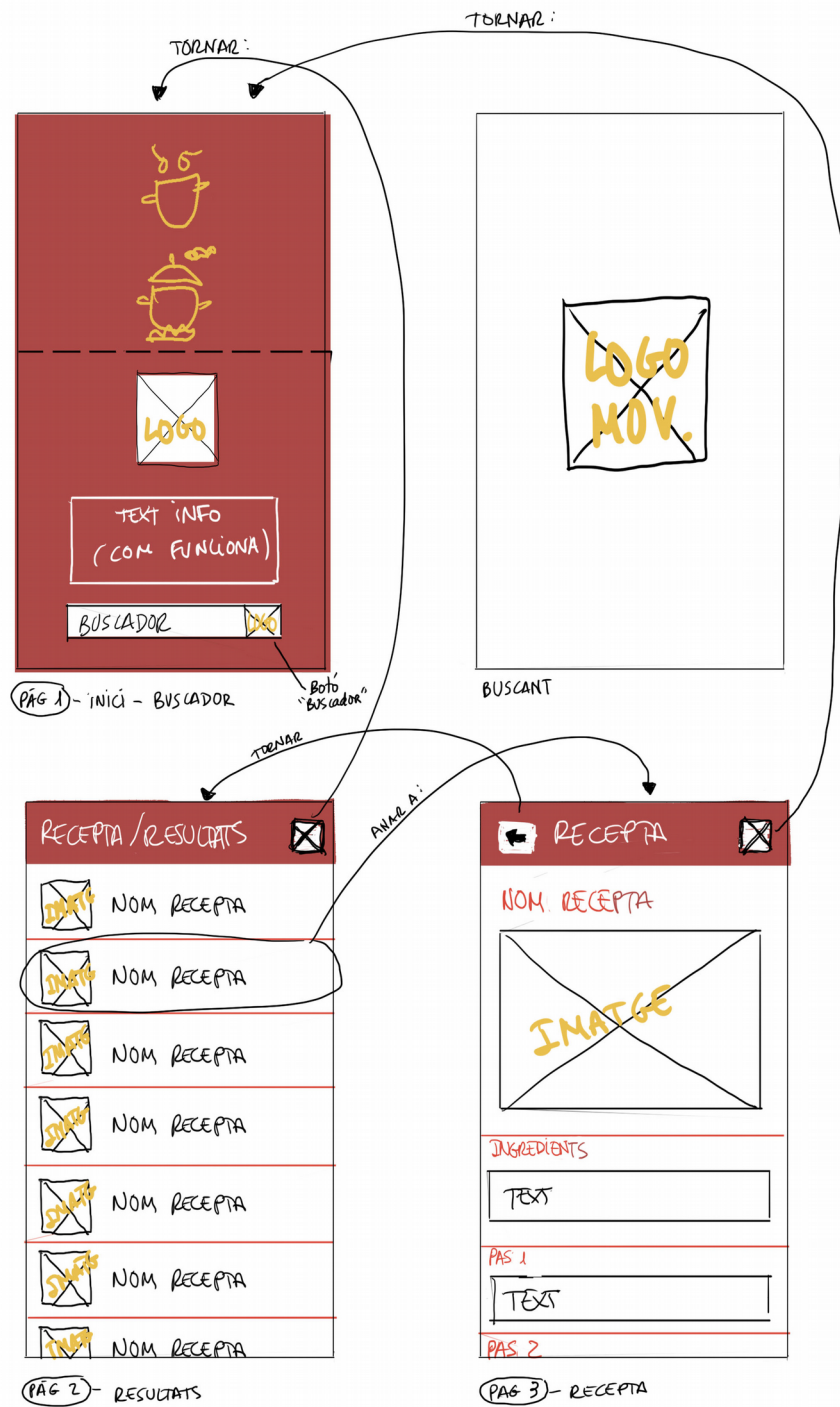


Figura 9: Wireframes dels dispositius mòbils

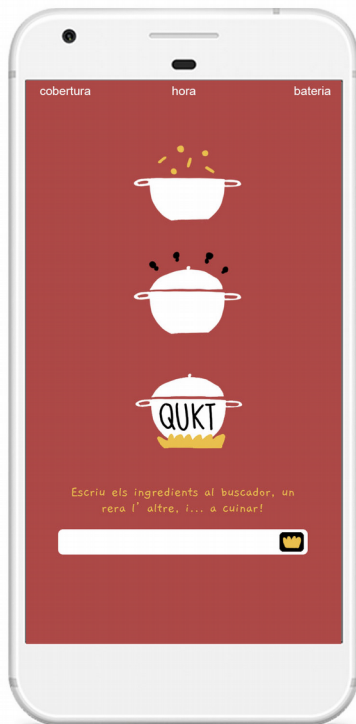
10.2 Prototips Hi-Fi

Mockups per a navegador web

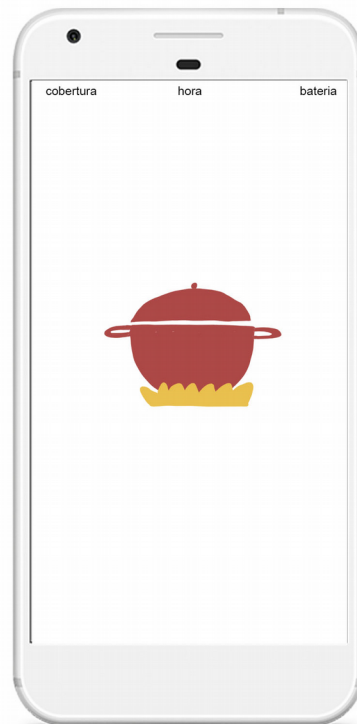


Figura 10: Mockups per a navegador web

Mockups per a dispositius mòbils



1. INICI - BUSCADOR



2. BUSCADOR



3. RESULTATS RECEPTES



4. RECEPTE

Figura 11: Mockups per a dispositius mòbil

11. User Acceptance Testing

Descripció dels tests i proves realitzades per posar a prova el treball respecte a la seva funcionalitat, rendiment, utilitat, etc., indicant, si s'escau, quines eines, persones i procediments/protocols s'han aplicat. Incloure, per exemple, tests de:

- Usuari
- Usabilitat
- Seguretat

Taula de criteris d'acceptació

Dispositiu	Tipus	Test d'usuari	Passos a seguir	Criteri d'èxit	Acceptació		Comentaris
					Previ	Actual	
Android	Inici	Veure el logo de l'aplicació en la pantalla de càrrega	Obrir l'app per primer cop o després de tancar-la manualment	Es mostra la pantalla de càrrega amb el logo de l'app. Posteriorment es mostra la pantalla inicial		Acceptat	
Android	Cerca	Cercar receptes sense introduir text	Obrir l'app. Prémer el botó de cerca sense indicar ingredients.	Surt un missatge indicant a l'usuari que ha d'introduir algun ingredient	Rebutjat		Falta incloure el missatge
Android	Cerca	Fer una cerca	Obrir l'app. Introduir text. Prémer el botó de cerca	Es mostra una animació de càrrega.	Rebutjat		L'animació es mostra, però amb fons blanc. Ha de ser transparent
Android	Cerca	Cercar receptes	Obrir l'app. Introduir ingredients separats per comes. Prémer el botó de cerca	Es mostra una nova vista amb un llistat de receptes ordenades per ordre d'importància, segons el número de coincidències d'ingredients.		Acceptat	
Android	Llistat de receptes	Navegar pel llistat de receptes	Obrir l'app. Fer una cerca que retorni resultats	Es mostra una llista amb totes les receptes resultants de la cerca, en la que es veu el títol i la imatge. En fer clic sobre una recepta s'obre una vista de detall d'aquesta recepta		Acceptat	
Android	Llistat de receptes	Tornar a fer una cerca	Des de la vista del llistat de receptes: Fer clic a la icona superior dreta (Home) Fer clic en el botó "enrere" del dispositiu	Es torna a mostrar la vista inicial, la de cerca		Acceptat	
Android	Detall de recepta	Veure el detall d'una recepta	Des de la vista del llistat de receptes: Fer clic en una recepta del llistat	Es mostra la vista de detall de la recepta clicada. Es mostra el títol, la imatge, els ingredients i la descripció de la recepta		Acceptat	
Android	Detall de recepta	Tornar a fer una cerca	Des de la vista de detall d'una recepta: Fer clic a la icona	Es torna a mostrar la vista inicial, la de cerca		Acceptat	

			superior dreta (Home)			
--	--	--	-----------------------	--	--	--

Dispositiu	Tipus	Test d'usuari	Passos a seguir	Criteri d'èxit	Acceptació		Comentaris
					Previ	Actual	
Web	Cerca	Cercar receptes sense introduir text	Obrir la web Prémer el botó de cerca sense indicar ingredients.	Surt un missatge indicant a l'usuari que ha d'introduir algun ingredient	Rebutjat		Falta incloure el missatge. En prémer la tecla "Intro" la cerca s'hauria de fer també
Web	Cerca	Fer una cerca	Obrir la web. Introduir text. Prémer el botó de cerca	Es mostra una animació de càrrega.	Rebutjat		L'animació es mostra, però amb fons blanc. Ha de ser transparent
Web	Cerca	Cercar receptes	Obrir la web. Introduir ingredients separats per comes. Prémer el botó de cerca	Es mostra una nova vista amb el detall del primer resultat de la cerca, incloent el títol, la imatge, els ingredients i la descripció de la recepta.		Acceptat	
Web	Resultat de cerca	Avançar pel llistat de receptes	Des de la vista dels resultats de la cerca: Fer clic en la fletxa a la dreta del títol	Es mostra el detall de la següent recepta de la llista de resultats. Si no hi han més receptes, la vista no canvia		Acceptat	
Web	Resultat de cerca	Retrocedir pel llistat de receptes	Des de la vista dels resultats de la cerca: Fer clic en la fletxa a l'esquerra del títol	Es mostra el detall de la següent recepta de la llista de resultats. Si no hi han més receptes, la vista no canvia		Acceptat	

12. Instruccions d'instal·lació/implantació

12.1 Producció

L'aplicació d'aquest projecte encara no ha estat desplegada en producció. Es preveu fer-ho en un futur.

12.2 Desenvolupament

A continuació es detallen els passos a seguir per a tenir l'aplicació funcionant en local per al seu desenvolupament. És necessari tenir funcionant l'aplicació servidor i la client.

Consideracions prèvies:

- En ambdós casos s'assumeix que el sistema operatiu és Linux.
- Cal tenir instal·lat l'última versió del JDK i l'SDK d'Android, així com haver configurat les variables d'entorn JAVA_HOME I ANDROID_HOME
- És aconsellable tenir instal·lat un IDE adequat per a cada aplicació. Es recomana la instal·lació d'Intellij IDEA per al servidor i Webstorm + Android Studio per als clients.

Servidor

Per a instal·lar aquesta aplicació en un entorn de desenvolupament és necessari disposar del codi, que està publicat en el següent repositori de github:

<https://github.com/matallonor/qukt-server>

Un cop descarregat el codi en local, és necessari instal·lar les eines docker i docker-compose.

Després s'ha d'executar la comanda:

```
~$ docker-compose up -d
```

I, per últim, compilar i executar l'aplicació. Això es pot fer de dues maneres:

- Des d'un IDE com IntelliJ IDEA, seleccionant l'arxiu d'entrada al programa (que es troba a [delivery/src/main/kotlin/com/mtallon/qukt/delivery/App.kt](#)) i executant-lo amb F10.

- O des d'una terminal amb la comanda:

```
~$ ./mvnw package && java -jar delivery/target/delivery-1.0.0-SNAPSHOT.jar
```

Client

Per a instal·lar aquesta aplicació en un entorn de desenvolupament és necessari disposar del codi, que està publicat en el següent repositori de github:

<https://github.com/matallonor/qukt-app>

Un cop descarregat el codi en local, és necessari instal·lar les eines Node, npm i nvm, així com tenir instal·lat l'última versió del JDK i l'SDK d'Android.

Primer escollim la versió de l'eina npm que volem utilitzar, mitjançant nvm:

```
~$ nvm use
```

A continuació instal·larem Ionic fent servir npm amb les comandes següents:

```
~$ npm install -g cordova  
~$ npm install -g ionic
```

Després procedim a instal·lar totes les dependències situades en l'arxiu package.json amb:

```
~$ npm install
```

I, per últim, compilar i executar l'aplicació amb la comanda:

```
~$ npm run ionic:serve
```

Si tot ha anat bé. Si anem al navegador i busquem la direcció <https://localhost:8100> veurem la nostra app funcionant.

13. Instruccions d'ús

L'aplicació client d'aquest projecte es basa en una sola funcionalitat, que permet a l'usuari introduir en una barra de cerca els ingredients que té disponibles i, en enviar la cerca, el servidor retorna una llista de receptes que s'ajusten a aquests ingredients.

Així que l'usuari tan sols haurà de descarregar l'aplicació mòbil en la Play Store o obrir un navegador web i navegar a la url de la aplicació web. Un cop en la vista principal, li apareixerà una barra de cerca en la que pot escriure ingredients. Quan envii la cerca s'obrirà una nova vista amb el llistat de resultats.

L'ús d'aquesta aplicació és tan senzill que no s'ha vist necessari incloure dins la pròpia aplicació client una secció de tutorial. En tot cas, sí s'ha vist convenient incloure un petit text sobre la barra de cerca que informa a l'usuari que ha d'introduir els ingredients de què disposa separats per comes.

14. Projecció a futur

Un cop assolits els principals objectius del projecte, es preveu en les futures fases del mateix l'ampliació d'algunes de les seves funcionalitats.

Afegir receptes

Actualment la base de dades consta d'unes poques receptes introduïdes a ma, que són suficients per a donar per tancada aquesta fase inicial del projecte. No obstant això, és evident que per aconseguir que l'aplicació sigui realment útil pels usuaris en calen moltes més.

Amb aquest motiu es preveu la necessitat d'afegir una nova funcionalitat que permetrà la creació de noves receptes a la base de dades còmodament des de la web i l'aplicació mòbil.

Recepta del dia

L'aplicació està ideada per ajudar els usuaris a prendre decisions a l'hora de preparar un plat amb el què disposen, però sembla oportú que també els hi serveixi d'inspiració en general a l'hora de cuinar.

Es preveu la creació d'una nova vista en les aplicacions clients que mostrin la recepta del dia, una recepta escollida de manera aleatòria.

Auto-completar ingredients

El sistema actual per introduir ingredients, tot i que funcional, és poc usable. L'usuari ha d'introduir en format text el nom dels ingredients separats per comes. Això fa que el procés sigui feixuc i que si l'usuari s'equivoca en unes lletres la cerca no retorni resultats.

Per a solucionar aquest problema es preveu l'ampliació de les funcionalitats per auto-completar el camp de text dels ingredients a mida que l'usuari escriu. Cada vegada que s'introdueixi una lletra es farà una consulta a base de dades que retornarà el llistat d'ingredients que coincideixin amb el text introduït per l'usuari.

15. Pressupost

Pressupost detallat del projecte:

Equip humà			
Tasca	Recurs	Hores	Cost (20€/h)
Definició i gestió del projecte	Marc Tallón	20	400
Gestió metodologia àgil (SCRUM master)	Marc Tallón	2	40
Disseny UI	Marc Tallón	4	80
Definició i configuració de sistemes	Marc Tallón	10	200
Disseny i implementació de base de dades	Marc Tallón	20	400
Disseny de l'arquitectura de l'aplicació	Marc Tallón	12	240
Desenvolupament del servidor	Marc Tallón	8	160
Desenvolupament de les aplicacions client	Marc Tallón	32	640
Testing i Quality Assurance	Marc Tallón	12	240
Manteniment i desenvolupament correctiu	Marc Tallón	8	160
TOTAL equip humà		128	2560
Equipament tècnic (Software i hardware)			
Concepte	Cost/hora *	Hores d'ús	Cost aprox.
Entorns de desenv. IntelliJ IDEA i Webstorm			Gratuït per a estudiants
PC	0,4	128	51
Mac	0,43	20	8,6
Smartphone Android	0,1	50	5
Raspberry	0,01	128	1,2
Total equipament tècnic			65,8
Altres despeses			
Concepte	Cost mensual	Durada del servei	Cost aprox.
Connexió a internet	40	3 mesos	120
Altres (Lloguer despatx, llum, etc)	500	3 mesos	1500
Total altres despeses			1620
Contingència del 10%			10%(4245,8) = 424,58
TOTAL PROJECTE			4670,38

* El cost per hora de l'equipament s'ha calculat com a una aproximació de la fórmula:
Preu del producte / hores de vida útil aproximades

16. Conclusions

Havent finalitzat el projecte en el temps previst i amb les funcionalitats plantejades a l'inici, es pot concloure que, en termes generals, el projecte ha estat un èxit. S'han seguit de manera escrupolosa les dates límit estipulades en la fase de planificació i s'han assolit els objectius principals del projecte.

No ha estat, però, un camí lliure d'obstacles. Com en qualsevol projecte de software hi han hagut moments difícils en què les coses no surten tan ràpid com estaven previstes, però s'ha pogut gestionar bé gràcies a que la planificació inicial es va fer amb realisme.

Han estat especialment útils els coneixements adquirits al llarg del Grau en matèria de gestió de projectes. S'han procurat posar en pràctica els conceptes apresos en l'assignatura Gestió de Projectes, en especial els que tenen a veure amb les metodologies àgils, sempre tenint present que tots els rols els ha exercit una única persona.

També s'ha procurat tenir cura del disseny i la usabilitat de l'aplicació, aplicant els conceptes de composició visual, teoria del color, animació, arquitectura de la informació, etc.

I, naturalment, han estat imprescindibles per a la compleció del projecte els coneixements adquirits en matèria d'Enginyeria del Software, Programació Orientada a Objectes, Patrons de disseny i Bases de Dades. És en aquesta àrea, pot ser, en què s'ha hagut de fer més recerca i en què s'han ampliat més els continguts del Grau de Multimèdia.

El producte final, tot i estar lluny encara de ser un producte de mercat, compleix amb els objectius plantejats a l'inici i serveix de punt de partida per a desenvolupar un producte de qualitat, escalable i mantenible.

Annex 1. Lliurables del projecte

Juntament amb aquesta carpeta s'adjunten els següents arxius:

Elements gràfics del software

Són els arxius png que es fan servir en el codi del projecte per mostrar el logo, l'animació i els botons.

Es troben en el directori "Annexos/Assets".

Diagrames

Són els diagrames que s'han anat creant durant la fase d'anàlisi i disseny del projecte (Diagrama de classes, d'activitat i de casos d'ús).

Es troben en el directori "Annexos/Diagrames"

Diagrama de Gantt

És el diagrama de Gantt que es va elaborar durant la fase de planificació del projecte.

Es troba en el directori "Annexos/Gantt"

Esquema de l'arquitectura del servidor

És l'esquema que defineix el patró d'arquitectura en què s'ha basat el desenvolupament del servidor (Clean Architecture)

Es troba en el directori "Annexos/Arquitectura Servidor"

Wireframes

Són els wireframes tant de web com de mòbil que es van generar durant l'etapa de disseny UI/UX.

Es troben en el directori "Annexos/Wireframes"

Mockups

Són els mockups tant de web com de mòbil que es van generar durant l'etapa de disseny UI/UX.

Es troben en el directori "Annexos/Mockups"

Annex 2. Bibliografia

- Samuel, S. Bocutiu, S. (2017). Programming Kotlin. Packt publishing
- McConnell, S. (2004). Code Complete, 2nd edition. Microsoft Press
- C. Martin, R. (2008). Clean Code. Prentice Hall
- C. Martin, R. (2011). The Clean Coder. Prentice Hall
- C. Martin, R. (2017). Clean Architecture. Prentice Hall
- Freeman, E. Robson, E. (2004). Head First Design Patterns. O'Reilly Media
- Beck, K (2002). Test-Driven Development By Example. Addison-Wesley
-
- Elastic Team. (2018). Elastic Stack and Product Documentation
Obtingut d'Elastic: <https://www.elastic.co/guide/index.html>
- Docker Inc. (2018). Docker Documentation
Obtingut de Docker docs: <https://docs.docker.com/>
- Pivotal Software. (2018). Spring Framework Documentation
Obtingut de Spring.io: <https://spring.io/docs>
- Google. (2018). Angular Documentation
Obtingut d'Angular.io: <https://angular.io/docs>
- Ionic Framework Team. (2018). Ionic Framework Documentation
Obtingut d'Ionicframework.com: <https://ionicframework.com/docs/>

Annex 3. Vita

L'autor d'aquest treball, Marc Tallón Ortega, és desenvolupador de software professional i, a data del present document, estudiant del Grau de Multimèdia amb especialització en enginyeria web en la Universitat Oberta de Catalunya.

Va iniciar la seva carrera en l'àmbit de les TIC l'any 2014, després de cursar un CFGS de desenvolupament d'aplicacions multiplataforma. A data del present document l'autor ha treballat desenvolupant software per les següents empreses:

- Euromaster 2000 (Desenvolupador backend Java, 2014)
- WorktoCloud (Desenvolupador Android, 2014)
- Ricoh (Desenvolupador backend Java, 2015 – 2016)
- Nakima (Desenvolupador backend Java i frontend Javascript, 2017 - 2019)

Per a més informació visitar el seu perfil de LinkedIn a:

<https://www.linkedin.com/in/marc-tall%C3%B3n-ortega-36077596/>