

Claudia Altadill Alvarez

APLICACIÓ D'INFORMACIÓ NUTRICIONAL

TREBALL DE FI DE GRAU

dirigit per la Dra. M. Àngels Moncusí Mercadé

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

**Tarragona**

**2024**

---

## **Agraïments**

Aquest Projecte de Fi de Grau és el resultat de mesos de dedicació i aprenentatge. Vull expressar el meu sincer agraïment a totes les persones que han contribuït a fer-lo possible.

En primer lloc, vull agrair a la meva tutora, M. Àngels Moncusí Mercadé, per la seva ajuda en definir millor la meva idea inicial i per guiar-me durant tot el procés.

També vull donar les gràcies a la meva família pel seu suport incondicional, a la meva millor amiga per corregir qualsevol errada d'aquest document, a la meva parella per la seva paciència i suport durant els moments de bloqueig en el projecte, i als meus amics de la universitat pel seu constant acompanyament.

---

## **Resum**

Aquest Treball de Fi de Grau es centra amb el desenvolupament d'una aplicació mòbil en l'àmbit de la nutrició. La seva funció principal és proporcionar informació sobre els macronutrients d'una recepta, i permet obtenir aquesta informació per una ració individual, ajustant-se a la quantitat de persones o a la quantitat de grams que es vol consumir.

La metodologia de desenvolupament ha seguit els diferents passos de l'enginyeria del Software: anàlisis dels requisits, disseny, implementació i avaluació. El projecte ha estat programat amb Flutter per al front-end, Python per al back-end i s'ha utilitzat PostgreSQL per a la gestió de la base de dades.

## **Resumen**

Este Trabajo de Fin de Grado se centra en el desarrollo de una aplicación móvil en el ámbito de la nutrición. Su función principal es proporcionar información sobre los macronutrientes de una receta, permitiendo obtener esta información para una ración individual, ajustándose a la cantidad de personas o a la cantidad en gramos que se desea consumir.

La metodología de desarrollo ha seguido los distintos pasos de la ingeniería del Software: análisis de requisitos, diseño, implementación y evaluación. El proyecto ha sido programado con Flutter para el front-end, Python para el back-end y se ha utilizado PostgreSQL para la gestión de la base de datos.

## **Abstract**

This Final Degree Project focuses on the development of a mobile application in the field of nutrition. Its main function is to provide information

---

about the macronutrients of a recipe, and to give this information for an individual dish, based on the number of people or the amount in grams that you want to consume.

The development methodology followed the different steps of Software engineering: requirements analysis, design, implementation and evaluation. The project was programmed using Flutter for the front-end, Python for the back-end and PostgreSQL was used for database management.

---

# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
<b>2</b>	<b>Objectius</b>	<b>2</b>
<b>3</b>	<b>Estudi previ</b>	<b>3</b>
<b>4</b>	<b>Planificació</b>	<b>5</b>
<b>5</b>	<b>Requisits de l'aplicació</b>	<b>7</b>
5.1	Requisits funcionals . . . . .	7
5.1.1	Diagrama de casos d'ús . . . . .	8
5.1.2	Especificació textual i seqüències dels casos d'ús . . . . .	8
5.2	Requisits no funcionals . . . . .	24
5.2.1	Requisit d'usabilitat . . . . .	24
5.2.2	Requisit de rendiment . . . . .	25
5.2.3	Requisit de seguretat . . . . .	26
5.2.4	Requisit de portabilitat . . . . .	26
5.2.5	Requisit de mantenibilitat . . . . .	27
5.2.6	Requisit d'escabilitat . . . . .	27
<b>6</b>	<b>Disseny</b>	<b>28</b>
6.1	Arquitectura de l'aplicació . . . . .	28
6.2	Disseny de la interfície de l'usuari . . . . .	31
6.2.1	Prototips . . . . .	32

---

6.2.2	Disseny . . . . .	33
6.3	Disseny de classes . . . . .	36
6.3.1	Diagrama de classes . . . . .	38
6.4	Disseny de la persistència de dades . . . . .	38
6.5	Disseny de l'API . . . . .	39
<b>7</b>	<b>Implementació</b>	<b>42</b>
7.1	Detalls de la tecnologia . . . . .	42
7.1.1	Back-end . . . . .	42
7.1.2	Front-end . . . . .	45
7.2	Algoritmes específics . . . . .	46
7.2.1	Mètode upload . . . . .	46
7.2.2	Mètode historial i list . . . . .	49
7.2.3	Mètode checkSharedRecipes . . . . .	49
7.3	Estructura i interfície d'usuari . . . . .	49
7.3.1	Classe main . . . . .	50
7.3.2	Classe MyHomePage . . . . .	50
7.3.3	Classe LoginForm . . . . .	51
7.3.4	Classe ConditionalForm . . . . .	52
7.3.5	Classe HistorialPage . . . . .	54
7.3.6	Classe ListFoodPage . . . . .	55
7.3.7	Classe InfoPage . . . . .	55
7.3.8	Classe ReceiptPublicPage . . . . .	56
7.3.9	Classe DeleteUserPage . . . . .	56

---

7.3.10	Classe HelpPage . . . . .	57
7.4	Altres implementacions . . . . .	57
7.4.1	Base de dades ingredients . . . . .	57
7.4.2	Connexió ChatGpt . . . . .	58
<b>8</b>	<b>Problemes d'implementació</b>	<b>59</b>
8.1	Intel·ligència artificial . . . . .	59
8.2	Amazon web services . . . . .	59
8.3	Dades per a la base de dades d'aliments . . . . .	60
<b>9</b>	<b>Avaluació</b>	<b>61</b>
9.1	Avaluació del projecte . . . . .	61
9.2	Avaluació dels costos . . . . .	64
<b>10</b>	<b>Legislació i protecció de dades</b>	<b>66</b>
<b>11</b>	<b>Conclusions</b>	<b>67</b>
11.1	Aportació de la carrera al projecte . . . . .	68
11.2	Aportació personal del projecte . . . . .	69
11.3	Ampliacions futures . . . . .	70
<b>12</b>	<b>Bibliografia</b>	<b>72</b>
<b>13</b>	<b>Annexos</b>	<b>75</b>
13.1	Video de l'aplicació . . . . .	75
13.2	Codis . . . . .	75
13.2.1	obtenerCantidadIngrediente . . . . .	75

---

13.2.2	obtenirMacros . . . . .	76
13.2.3	Codi d'inserir dades des d'un excel . . . . .	77

---

## Índex d'imatges

1	Diagrama de Gantt (part 1) . . . . .	5
2	Diagrama de Gantt (part 2) . . . . .	6
3	Diagrama de Gantt (part 3) . . . . .	6
4	Diagrama de casos d'ús . . . . .	8
5	Seqüència del cas d'ús 00.IniciarSessió . . . . .	9
6	Seqüència del cas d'ús 01.Registrar-se . . . . .	11
7	Seqüència del cas d'ús 02.IntroduirIngredients . . . . .	12
8	Seqüència del cas d'ús 03.VisualitzarIngredients . . . . .	14
9	Seqüència del cas d'ús 04.VisualitzarHistorial . . . . .	16
10	Seqüència del cas d'ús 05.CercarRecepta . . . . .	17
11	Seqüència del cas d'ús 06.EliminarRecepta . . . . .	18
12	Seqüència del cas d'ús 07.MacronutrientsIndividuals . . . . .	19
13	Seqüència del cas d'ús 08.VisualitzarInformacióRecepta . . . . .	20
14	Seqüència del cas d'ús 09.EliminarUsuari . . . . .	21
15	Seqüència del cas d'ús 10.VisualitzarReceptesPúbliques . . . . .	23
16	Seqüència del cas d'ús 11.TancarSessió . . . . .	24
17	Diagrama arquitectura 3 capes . . . . .	28
18	Pantalla inici de sessió . . . . .	32
19	Pantalla registrar-se . . . . .	32
20	Pantalla principal . . . . .	32
21	Pantalla historial . . . . .	32
22	Pantalla informació . . . . .	33

---

23	Pantalla inici de sessió . . . . .	33
24	Pantalla registrar-se . . . . .	33
25	Pantalla principal text . . . . .	34
26	Pantalla principal pdf . . . . .	34
27	Pantalla principal imatge . . . . .	34
28	Pantalla historial de receptes . . . . .	34
29	Pantalla historial de receptes (2) . . . . .	34
30	Pantalla informació recepta . . . . .	35
31	Pantalla informació recepta (2) . . . . .	35
32	Pantalla informació recepta (3) . . . . .	35
33	Pantalla receptes públiques . . . . .	35
34	Pantalla receptes públiques (aplicant un filtre) . . . . .	35
35	Pantalla informació recepta d'un altre usuari . . . . .	35
36	Pantalla llistat ingredients . . . . .	36
37	Pantalla d'ajuda . . . . .	36
38	Pantalla eliminar usuari . . . . .	36
39	Diagrama de classes . . . . .	38
40	Diagrama base de dades . . . . .	39

---

## Índex de taules

1	Descripció de les proves i resultats del sistema . . . . .	61
---	--	----

---

## Índex de codis

1	Extracció de text des d'un PDF . . . . .	47
2	Extracció de text des d'una imatge . . . . .	47
3	Filtratge de línies . . . . .	47
4	Diccionari d'unitats . . . . .	48
5	Consulta mètode checkSharedRecipes . . . . .	49
6	Mètode d'enciptació de la contrasenya . . . . .	51
7	Línia per extraure el path . . . . .	52
8	Funcions per obtenir imatges . . . . .	53
9	Algoritme de cerca en l'historial . . . . .	54
10	Navegació historial a l'informació de la recepta . . . . .	54
11	Connexió base de dades . . . . .	57
12	Funció per obtenir la quantitat i l'ingredient de la llista . .	75
13	Funció per obtenir els macronutrients . . . . .	76
14	Insertar dades de l'excel d'ingredients . . . . .	78

---

# 1 Introducció

En l'actualitat, l'alimentació saludable i la gestió dels macronutrients són temes d'interès per a la majoria de les persones. Hi ha una necessitat creixent d'eines nutricionals per millorar els coneixements de la informació nutricional del que consumim i controlar certs aspectes segons l'interès i el perfil de cada persona.

Per exemple, el control dels hidrats de carboni és crucial per a moltes persones, especialment aquelles amb diabetis, per tal de gestionar els nivells de glucosa en sang. D'altra banda, hi ha persones que desitgen mantenir o reduir el seu percentatge de greix corporal, o bé augmentar la massa muscular, per a les quals és essencial tenir un control de les calories i un coneixement precís d'altres nutrients essencials per a assolir els seus objectius.

Tot i que aquests perfils són diversos, comparteixen un objectiu comú: l'anàlisi de la informació nutricional. Encara que es poden usar recursos com llistes, cerques a internet o diverses aplicacions disponibles en el mercat, aquests no aborden el repte d'obtenir la informació d'un conjunt de nutrients d'una recepta.

Aquest projecte consisteix en el desenvolupament d'una aplicació que, a partir de la introducció dels ingredients d'una recepta, proporcionarà informació detallada sobre els cinc macronutrients principals: hidrats de carboni, energia, proteïna, greix i fibra. A més, permetrà obtenir aquesta informació de manera individualitzada per a cada aliment.

L'aplicació també oferirà la possibilitat de calcular la porció individual consumida per cada persona i inclourà altres funcionalitats, com la consulta de receptes guardades, tant les pròpies com les d'altres usuaris, aquestes últimes només es podran visualitzar si estan publicades.

---

## 2 Objectius

Tot treball té diferents objectius i motivacions que són claus per assolir els resultats desitjats. En el meu cas, és important destacar la motivació que m'ha portat a escollir aquest tema abans de remarcar els objectius específics que vull aconseguir amb aquest treball.

L'elecció d'aquest tema està influït pel fet que soc una persona amb diabetis tipus 1, i en el meu dia a dia he de controlar els hidrats de carboni que consumeixo. Aquesta necessitat personal m'ha portat a enfocar-me en la importància de la informació nutricional en les receptes, amb l'objectiu de proporcionar una eina que permeti consultar els macronutrients d'aquestes i obtenir la informació essencial que moltes persones necessiten en la seva vida quotidiana.

L'objectiu principal d'aquest treball és desenvolupar una eina pràctica que permeti als usuaris consultar la informació nutricional de diverses receptes, així com guardar i revisar aquelles que ja han estat consultades. Aquesta eina no només serà útil per a persones que han de controlar les quantitats dels carbohidrats, com les persones diabètiques, sinó també per a qualsevol persona interessada en mantenir una dieta equilibrada i saludable.

Objectius específics:

- Desenvolupar una aplicació mòbil.
- Crear una base de dades amb informació dels macronutrients per ingredients i habilitar la capacitat de guardar les receptes dels usuaris per crear un llistat accessible.
- Desenvolupar la lògica de l'aplicació i implementar-la de manera funcional amb el front-end.
- Permetre obtenir la informació dels ingredients amb diferents formats: PDF, text i imatge.

---

### 3 Estudi previ

Tot i tenir una idea per al meu projecte, primer vaig haver d'investigar si ja existia alguna aplicació o pàgina web que realitzés el mateix. Aquest pas em va permetre determinar si la idea era realment innovadora en el mercat o si ja estava coberta per altres solucions.

Durant l'estudi, vaig identificar diverses aplicacions relacionades amb la nutrició, amb els següents objectius:

- **Fitia:** aplicació que facilita la planificació nutricional amb plans personalitzats d'alimentació. Permet establir objectius nutricionals específics i ajusta les recomanacions segons les necessitats i preferències del usuari. A més, ofereix seguiment del progrés i ajustos al pla en funció dels resultats obtinguts.
- **Nootric:** crea un perfil nutricional personalitzat basat en els hàbits alimentaris del usuari. Genera recomanacions i ajustos dietètics per aconseguir metes com la pèrdua de pes o la millora del benestar general. Inclou recursos educatius per ajudar en la presa de decisions sobre la dieta.
- **Oorenji:** ajuda a planificar l'alimentació amb recomanacions de porcions adequades i un equilibri nutricional òptim. Proporciona guies per combinar aliments de manera saludable i ajustar les porcions segons les necessitats individuals. Inclou eines per al seguiment de la ingesta nutricional.
- **8fit:** combina entrenaments físics amb plans de nutrició per a un enfocament integral de la salut. Ofereix rutines d'exercici personalitzades i guies de menjar per maximitzar els resultats. També proporciona seguiment del progrés i ajustos basats en els assoliments i necessitats del usuari.
- **Yuka:** permet escanejar codis de barres de productes alimentaris per

---

obtenir informació sobre la seva composició. Avalua els productes segons la qualitat nutricional i ingredients, oferint alternatives més saludables quan calgui. És útil per millorar la dieta evitant productes amb components poc saludables.

He observat que la majoria de les aplicacions de nutrició es centren a oferir plans nutricionals predefinitos per a millorar l'alimentació i estil de vida de l'usuari. Aquestes aplicacions proporcionen receptes i suggereixen combinacions d'aliments basades en l'equilibri dels macronutrients. No obstant això, la meva proposta és diferent. Pretenc permetre als usuaris introduir les seves pròpies receptes, així com obtenir informació específica sobre els macronutrients de cada recepta.

Encara que la meva aplicació també permetrà consultar la informació nutricional d'altres receptes, com les guardades per l'usuari o les compartides per altres, el que realment diferencia la meva proposta és la llibertat que ofereix a l'usuari per decidir quines receptes utilitzar. Aquesta flexibilitat és un aspecte clau que no es troba en les alternatives actuals.

Adicionalment, vull destacar que en la majoria de les aplicacions mencionades, les funcionalitats disponibles per als usuaris són limitades. Moltes opcions que permeten una major personalització estan reservades per a versions de pagament. La meva aportació personalitzada es manté accessible de manera gratuïta en tot moment, ja que la meva idea és proporcionar aquesta personalització i flexibilitat sense costos addicionals.

Per acabar, també cal remarcar la importància de dissenyar una interfície gràfica d'usuari intuïtiva que tingui en compte les diverses necessitats del públic objectiu, des de persones amb diabetis fins a esportistes o aquests interessats en mantenir una dieta equilibrada, així com els principis d'accessibilitat i de UX. Aquest enfocament garantirà que l'aplicació sigui no només tècnicament viable, sinó també atractiva, fàcil i ràpida d'utilitzar per a un ampli ventall d'usuaris.

## 4 Planificació

A continuació es pot observar la planificació que he seguit per dur a terme el projecte. Cal destacar que hi ha certs períodes en els quals no s'ha realitzat cap avançament, ja que estava prioritzant altres assignatures. Un cop finalitzades aquestes al llarg de l'estiu, he aconseguit progressar fluidament amb el projecte.

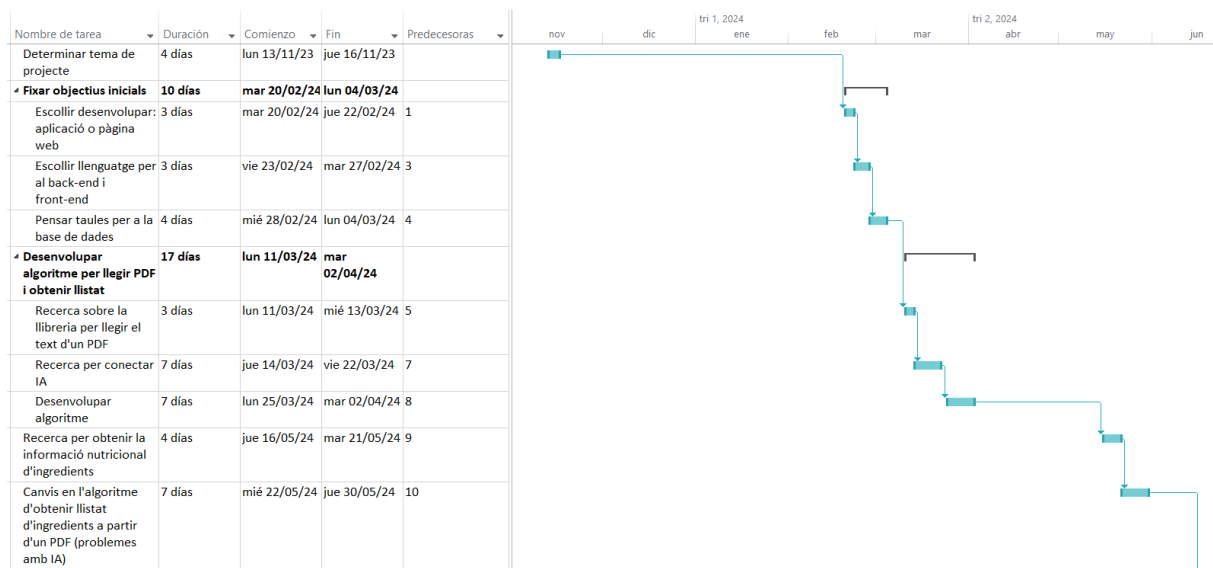


Figure 1: Diagrama de Gantt (part 1)

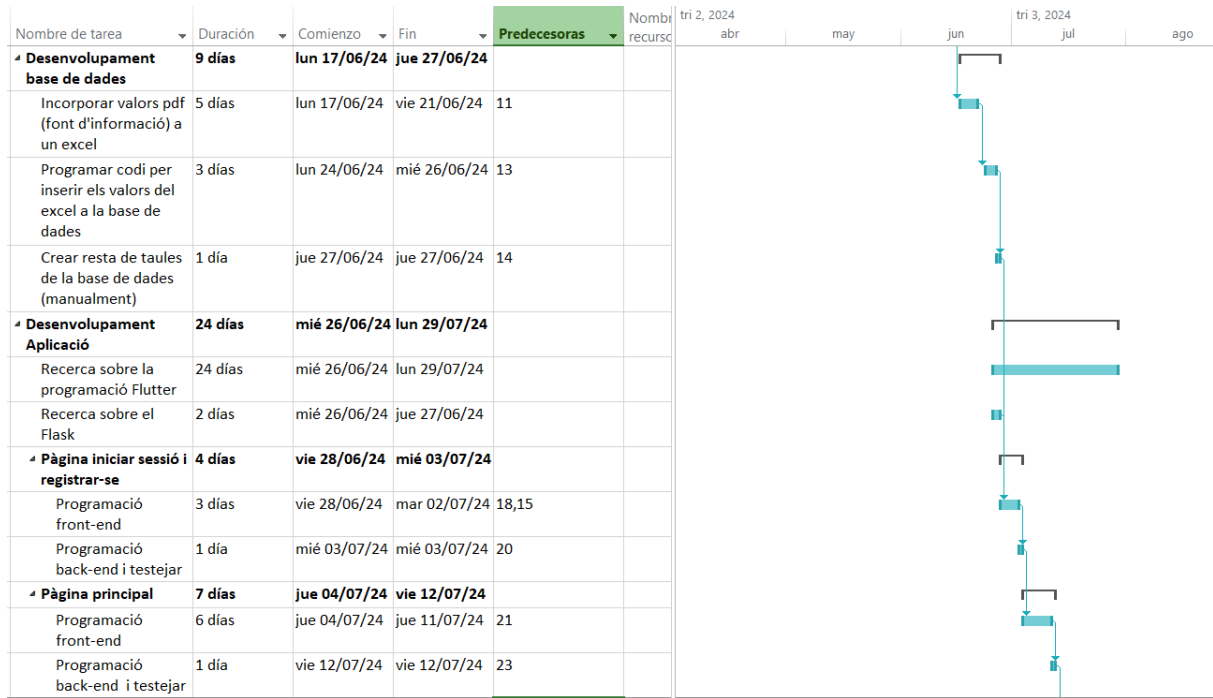


Figure 2: Diagrama de Gantt (part 2)

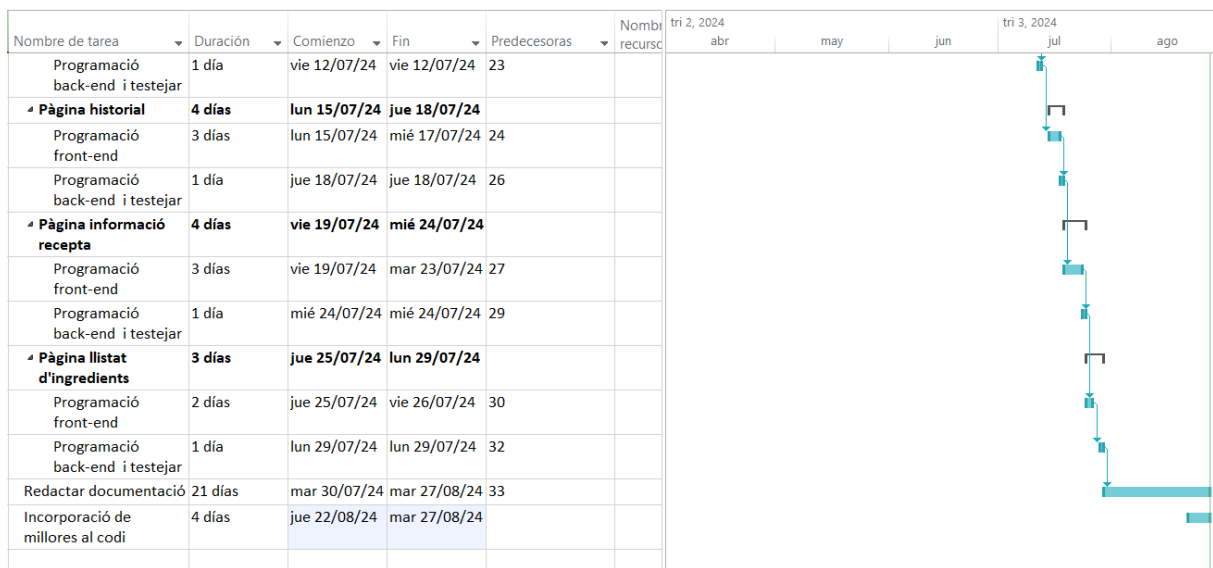


Figure 3: Diagrama de Gantt (part 3)

---

## 5 Requisits de l'aplicació

En aquest apartat tractaré els requisits funcionals amb el diagrama i l'especificació textual dels casos d'ús, com també els requisits no funcionals del projecte.

### 5.1 Requisits funcionals

Els requisits funcionals són els processos que ha de fer l'aplicació per satisfer les necessitats dels usuaris. Les funcionalitats que té l'aplicació i realitza l'usuari són les següents:

- Iniciar sessió o registrar-se.
- Introduir els ingredients d'una recepta, escollint el format: text, PDF o imatge.
- Obtenir el resultat dels macronutrients de la recepta introduïda.
- Veure l'historial de les receptes que s'han consultat.
- Eliminar una recepta que no volen consultar més.
- Observar la informació (nom, ingredients i macronutrients) de les receptes.
- Calcular els macronutrients de porcions individuals.
- Veure el llistat d'ingredients que hi ha en la base de dades.
- Dels ingredients obtenir els macronutrients a partir dels grams que volem.
- Cercar receptes d'altres usuaris i votar si han estat del seu gust.
- Cercar per caràcters o aplicar filtres a les llistes de receptes.
- Eliminar l'usuari de la base de dades.
- Tancar sessió.

### 5.1.1 Diagrama de casos d'ús

En l'aplicació només hi ha un tipus d'usuari que executa les tasques mencionades anteriorment, per aquest motiu, al diagrama s'observa a un únic actor, l'usuari, que interactua amb l'aplicació.

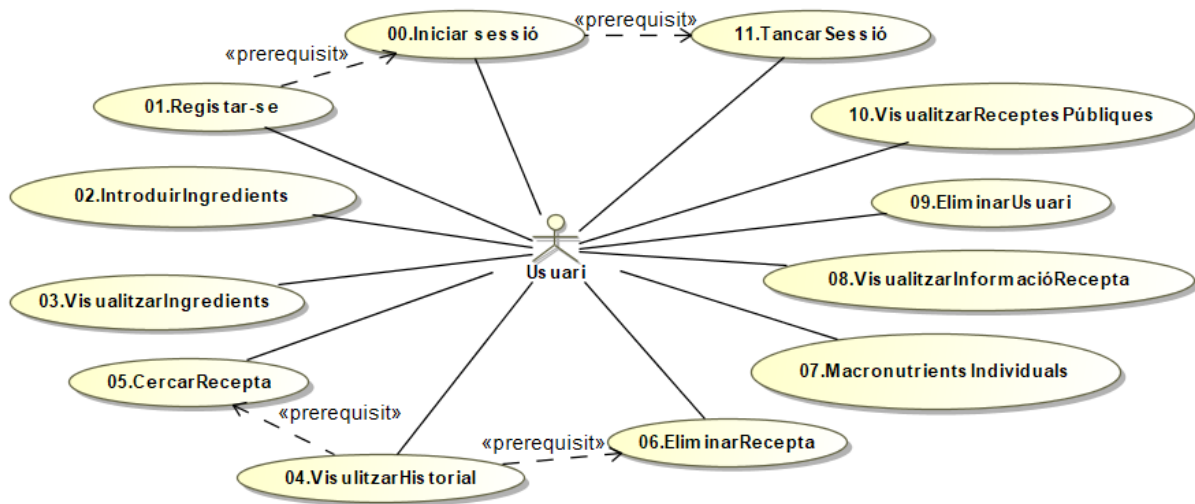


Figure 4: Diagrama de casos d'ús

### 5.1.2 Especificació textual i seqüències dels casos d'ús

#### Cd'ú 00.IniciarSessió

Resum de la funcionalitat: identificar a l'usuari i entrar a l'aplicació.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: tenir a l'Usuari registrat a l'aplicació.

Postcondició: s'ha identificat l'Usuari al sistema i té accés a la seva informació guardada (historial de receptes).

Procés normal principal:

1. El sistema demana el nom d'Usuari i la contrasenya de l'Usuari.
2. L'Usuari introdueix les dades demanades.
3. El sistema consulta les credencials.
4. El sistema presenta les funcions que pot realitzar l'Usuari.

Alternatives de procés i excepcions:

3a Les credencials son incorrectes.

3a1 El sistema mostra un missatge d'error.

3a2 El sistema torna al pas 1.

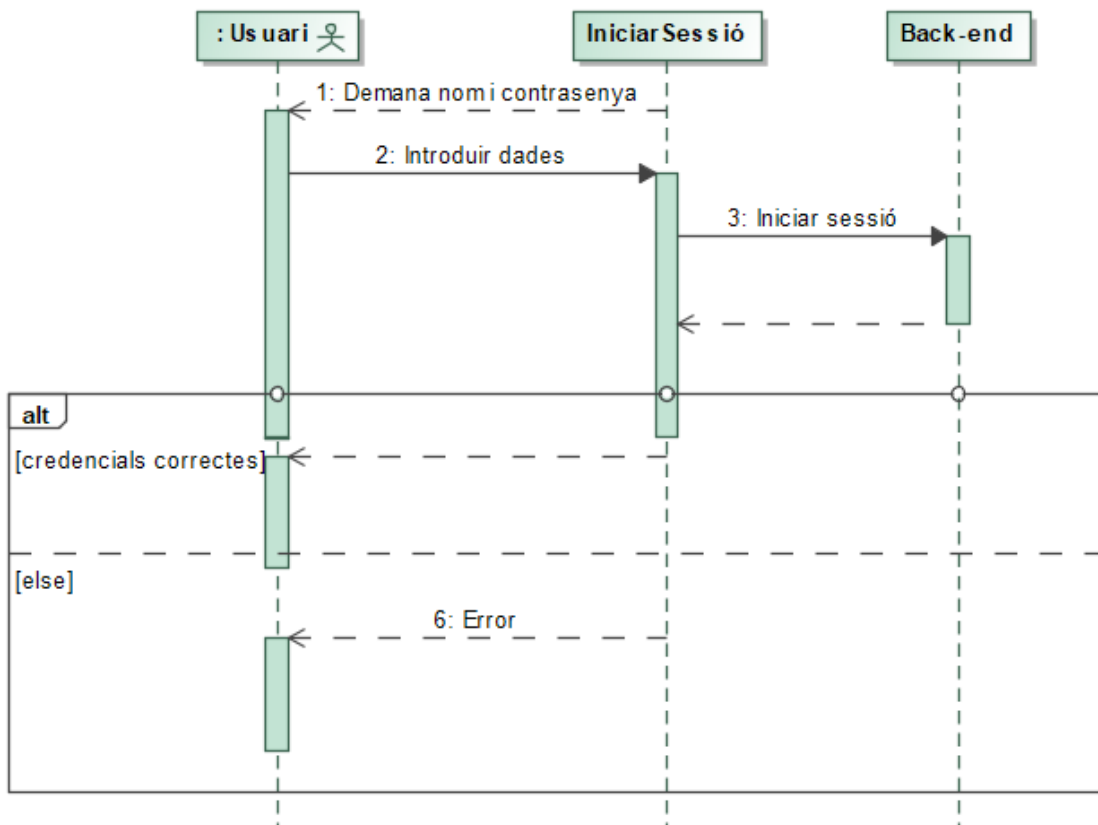


Figure 5: Seqüència del cas d'ús 00.IniciarSessió

## Cd'ú 01.Registrar-se

Resum de la funcionalitat: registrar a un nou Usuari i donar accés a l'aplicació.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: s'ha registrat a l'Usuari al sistema i té accés a aquest.

Procés normal principal:

- 
1. El sistema demana el nom, la contrasenya i la confirmació d'aquesta a l'Usuari.
  2. L'Usuari introdueix les dades demanades.
  3. El sistema registra el nouUsuari a la base de dades.
  4. El sistema presenta les funcions que pot realitzar l'Usuari.

Alternatives de procés i excepcions:

3a L'usuari introduït ja existeix.

3a1 El sistema mostra un missatge d'error.

3a2 El sistema torna al pas 1.

3b La contrasenya i la verificació d'aquesta no són iguals.

3a1 El sistema mostra un missatge d'error.

3a2 El sistema torna al pas 1.

3c La contrasenya no es posa amb el format adequat.

3a1 El sistema mostra un missatge d'error i indica quin format a de tenir la contrasenya.

3a2 El sistema torna al pas 1.

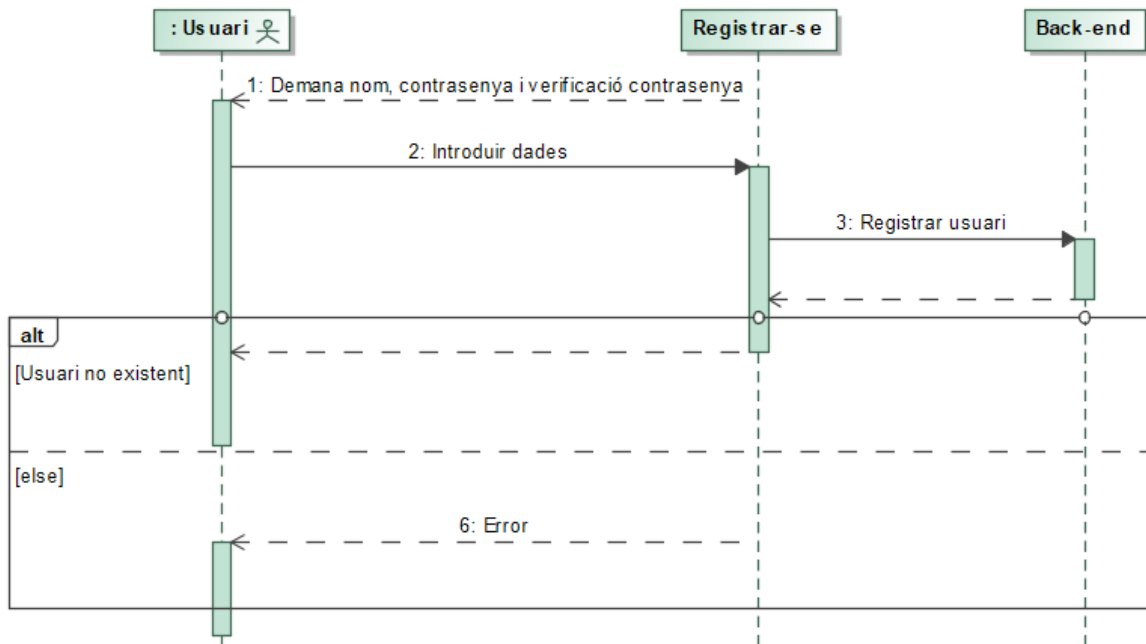


Figure 6: Seqüència del cas d'ús 01.Registrar-se

## Cd'ú 02.IntroduirIngredients

Resum de la funcionalitat: introduir els ingredients de la recepta segons el format indicat.

Paràmetres d'entrada: format d'ingredient.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: visualitzar els macronutrients de la recepta un cop s'han introduït les dades dels ingredients.

Procés normal principal:

1. El sistema mostra la pantalla d'inici amb els botons per escollir el format.
2. L'Usuari selecciona el format.
3. El sistema mostra els elements visuals per a la introducció de les dades en funció del format.
4. El sistema demana el nom de la recepta, introduir les dades amb el

format pertinent i si es vol publicar la recepta.

5. L'Usuari introdueix les dades demanades.
6. El sistema filtra els ingredients i la seva quantitat a partir de la informació introduïda.
7. El sistema calcula els macronutrients a partir de les dades aconseguïdes anteriorment i consultant la informació d'aquests a la base de dades.
8. El sistema guarda a la base de dades el nom, el llistat d'ingredients, el llistat de macronutrients, l'identificador de l'Usuari que consulta la recepta, els grams de la recepta, si és comparteix i el nombre d'ingredients.
9. El sistema mostra els macronutrients de la recepta.

Alternatives de procés i excepcions: cap.

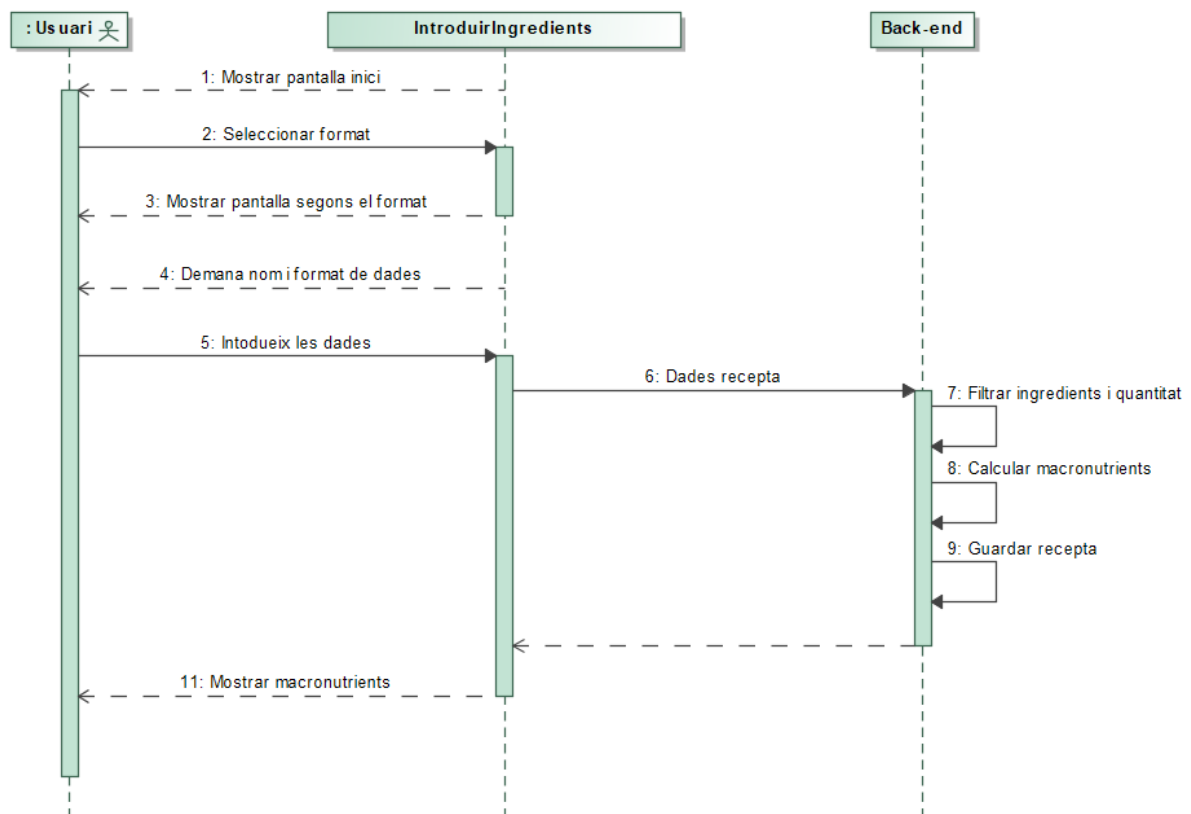


Figure 7: Seqüència del cas d'ús 02.IntroduirIngredients

---

### **Cd'ú 03. VisualitzarIngredients**

Resum de la funcionalitat: mostrar la llista d'ingredients que hi ha a la base de dades i proporcionar la seva informació.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: s'obté el llistat d'ingredients de la base de dades.

Procés normal principal:

1. El sistema consulta la taula d'aliments per obtenir la informació d'aquests.
2. El sistema mostra el llistat trobat d'ingredients.

Alternatives de procés i excepcions:

2a L'Usuari ha seleccionat un ingredient de la llista.

2a1 El sistema mostra la informació dels macronutrients i els grams de la informació.

2b L'Usuari vol realitzar una cerca.

2b1 L'Usuari introdueix els caràcters que vol cercar.

2b2 El sistema retorna el llistat dels ingredients que comencen pels caràcters introduïts.

2c L'Usuari canvia els grams d'un ingredient.

2c1 El sistema mostra la informació dels macronutrients en funció els grams que s'han introduït.

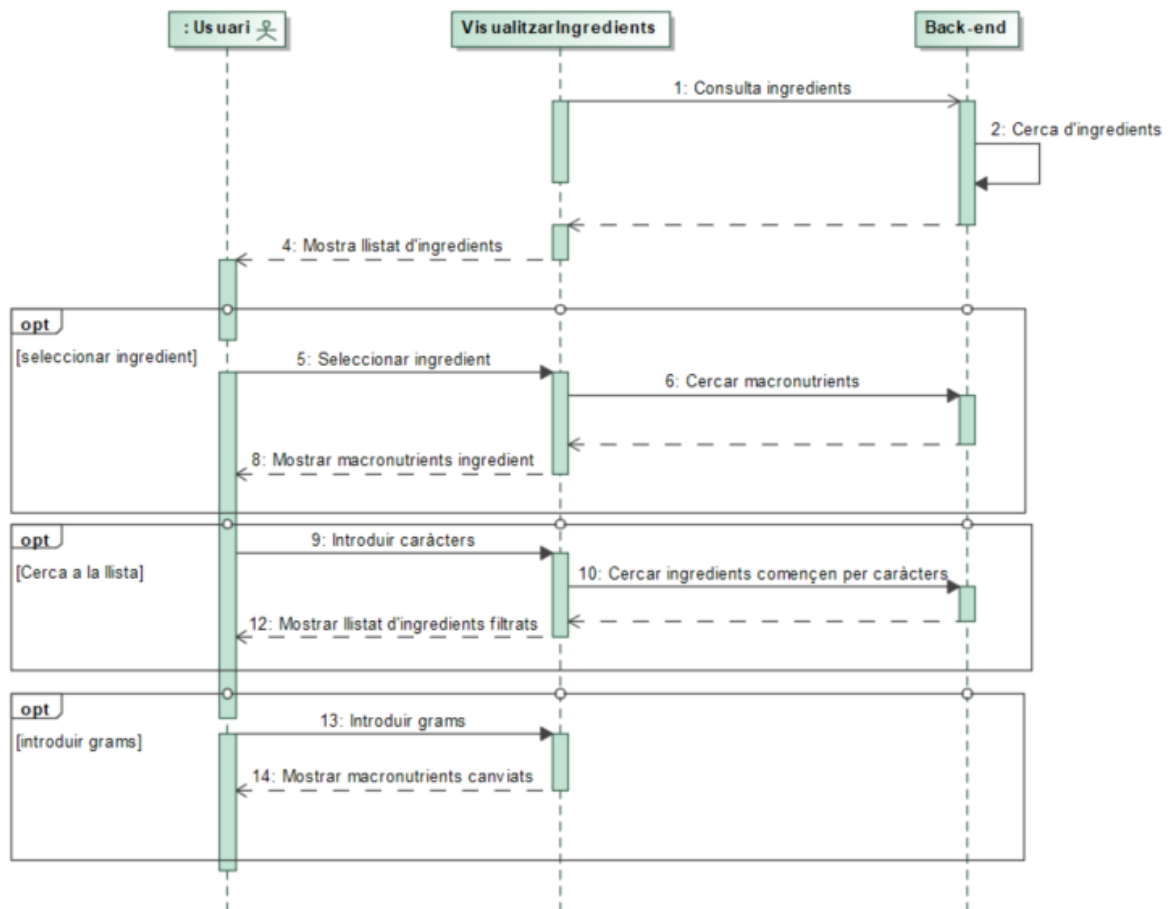


Figure 8: Seqüència del cas d'ús 03.VisualitzarIngredients

## Cd'ú 04.VisualitzarHistorial

Resum de la funcionalitat: mostrar les receptes consultades per l'Usuari.

Paràmetres d'entrada: identificador d'Usuari.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: visualitzar l'historial de les receptes que han estat consultades per l'Usuari o un text que indica que encara no se n'ha consultat cap.

Procés normal principal:

1. El sistema consulta a la base de dades les receptes que han estat guardades per l'Usuari.

---

2. El sistema mostra el llistat de receptes o un text si no hi ha res guardat.

Alternatives de procés i excepcions:

2a L'Usuari ha seleccionat per cercar les receptes (només si hi ha receptes a l'historial).

2a1 El sistema executa el cd'ú 05.CercarRecepta.

2b L'Usuari selecciona una recepta del llistat (només si hi ha receptes a l'historial).

2b1 El sistema executa el cd'ú 08.VisualitzarInformacióRecepta.

2c L'Usuari elimina una recepta del llistat (només si hi ha receptes al historial).

2c1 El sistema executa el cd'ú 06.EliminarRecepta.

2d L'Usuari utilitza un filtre per al llistat (només si hi ha receptes al historial).

2d1 El sistema modifica el llistat en funció el filtre.

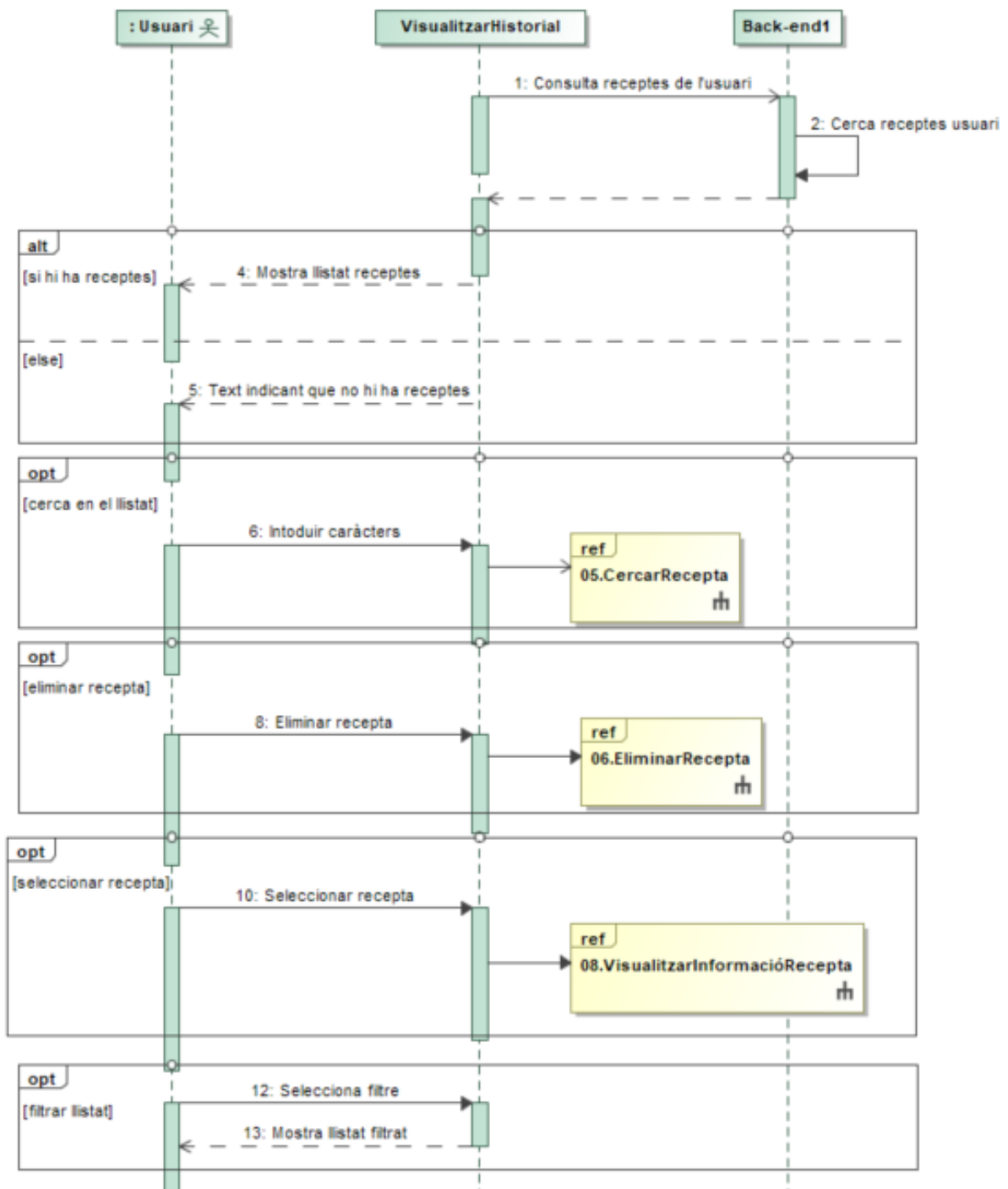


Figure 9: Seqüència del cas d'ús 04.VisualitzarHistorial

### Cd'ú 05.CercarRecepta

Resum de la funcionalitat: cercar per caràcters inicials d'una recepta o el nom d'aquesta.

Paràmetres d'entrada: caràcters introduïts.

Paràmetres de sortida: llistat de receptes filtrades.

---

Usuaris: Usuari.

Precondició: visualitzar l'historial de receptes i tenir receptes a la llista.

Postcondició: observar el llistat filtrat de les receptes.

Procés normal principal:

1. L'Usuari introdueix els caràcters.
2. El sistema cerca a la base de dades les receptes que coincideixen a l'inici amb els caràcters introduïts.
3. El sistema mostra el llistat (si n'hi ha) de les receptes filtrades.

Alternatives de procés i excepcions: cap.

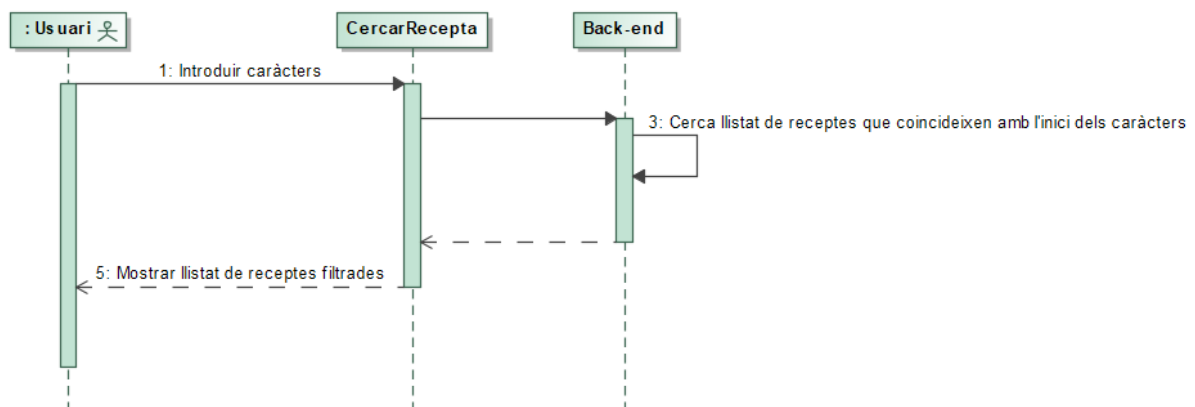


Figure 10: Seqüència del cas d'ús 05.CercarRecepta

## Cd'ú 06.EliminarRecepta

Resum de la funcionalitat: eliminar la recepta escollida de la llista de receptes de l'Usuari.

Paràmetres d'entrada: recepta a eliminar.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: visualitzar l'historial de receptes i tenir receptes a la llista.

Postcondició: s'ha eliminat la recepta relacionada amb aquest Usuari de la base de dades.

Procés normal principal:

1. L'Usuari selecciona la recepta que vol eliminar.

2. El sistema la busca a la base de dades amb nom i Usuari, i l'elimina.

Alternatives de procés i excepcions: cap.

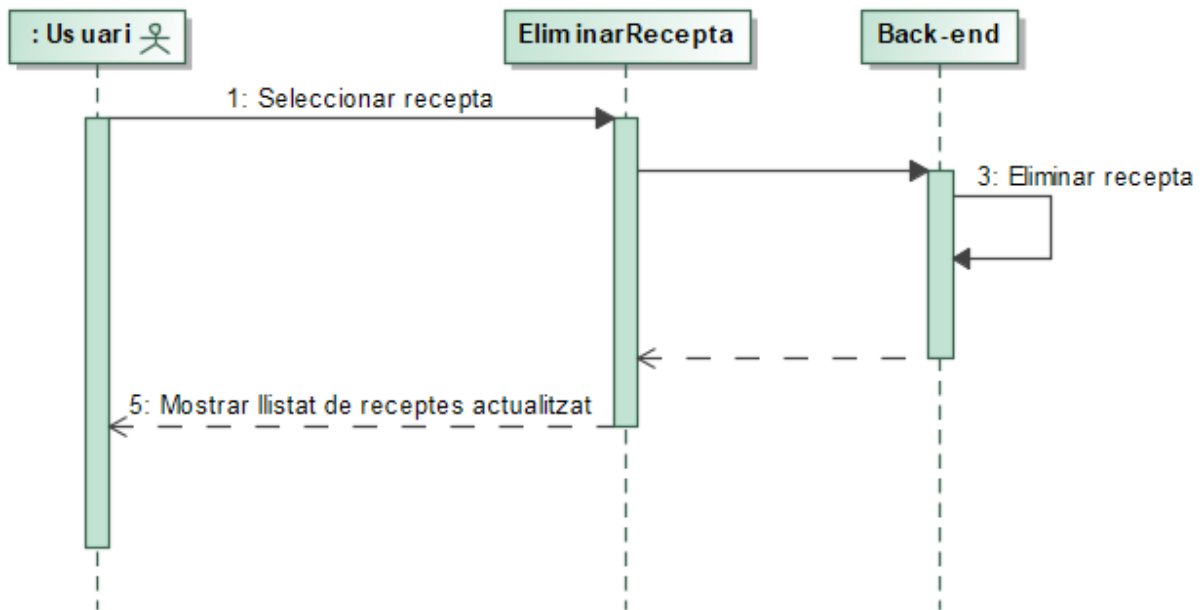


Figure 11: Seqüència del cas d'ús 06.EliminarRecepta

## Cd'ú 07.MacronutrientsIndividuals

Resum de la funcionalitat: calcular la quantitat orientativa de la recepta de manera individual.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: tenir una recepta introduïda.

Postcondició: obtenir els valors dels macronutrients per una ració individual.

Procés normal principal:

1. El sistema mostra les dues formes de calcular les racions individuals: nombre de persones o quantitat a menjar.
2. L'Usuari selecciona una de les dues opcions.
3. El sistema mostra la informació nutricional resultant.

Alternatives de procés i excepcions:

2a L'Usuari ha seleccionat el càlcul per persona.

2a1 L'Usuari introdueix el nombre de persones.

2a2 El sistema calcula el nou resultat dividint els valors per la dada introduïda per l'Usuari.

2b L'Usuari ha seleccionat el càlcul per grams.

2b1 L'Usuari introdueix el nombre de grams que consumirà.

2b2 El sistema calcula el nou resultat tenint en compte els grams totals i la dada introduïda per l'Usuari.

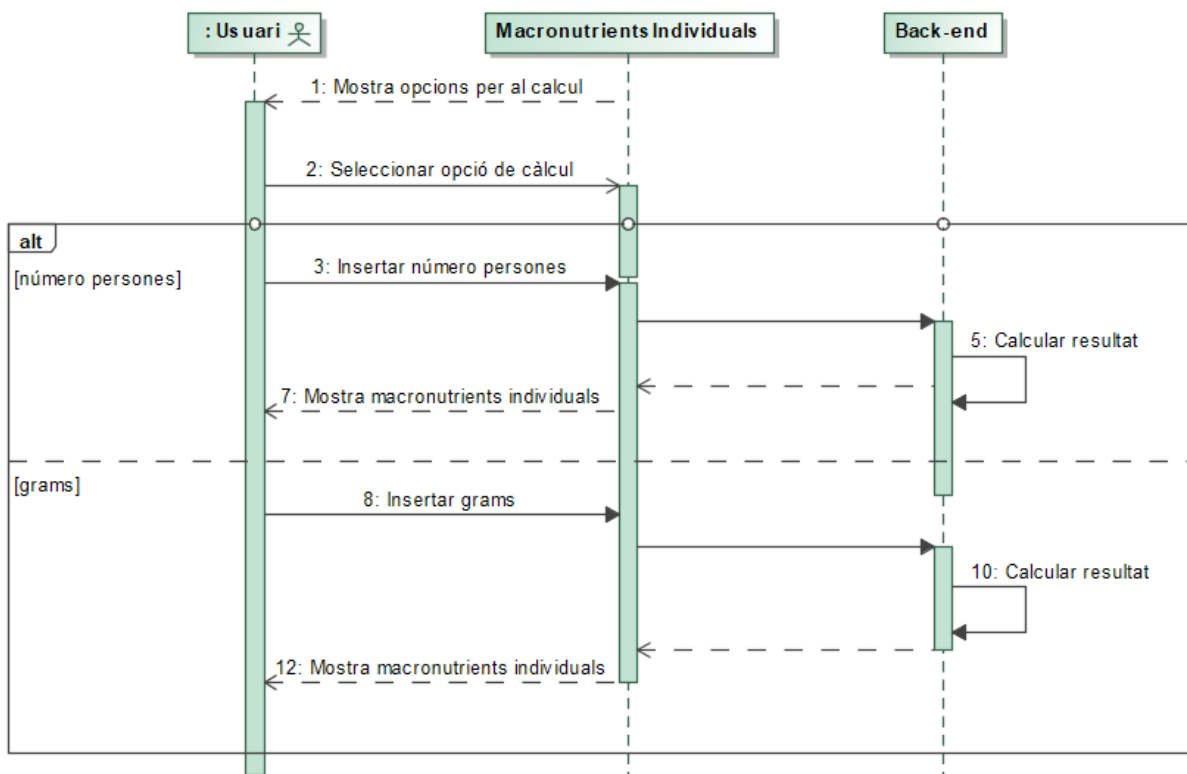


Figure 12: Seqüència del cas d'ús 07.MacronutrientsIndividuals

## Cd'ú 08.VisualitzarInformacióRecepta

Resum de la funcionalitat: mostrar la informació d'una recepta.

Paràmetres d'entrada: recepta.

Paràmetres de sortida: cap.

Usuaris: Usuari.

---

Precondició: cap.

Postcondició: s'obté les dades d'una recepta (nom, llistat ingredients, llistat macronutrients).

Procés normal principal:

1. El sistema cerca la recepta a la base de dades.
2. El sistema mostra les dades de la recepta a l'Usuari.

Alternatives de procés i excepcions: cap.

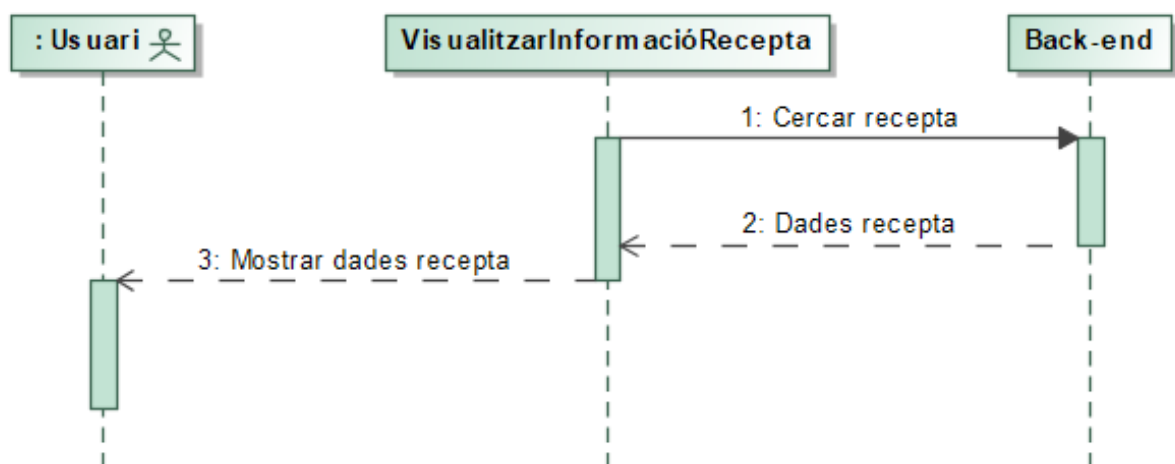


Figure 13: Seqüència del cas d'ús 08.VisualitzarInformacióRecepta

## Cd'ú 09.EliminarUsuari

Resum de la funcionalitat: eliminar l'Usuari del sistema.

Paràmetres d'entrada: Usuari.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: eliminar a l'usuari de la base de dades.

Procés normal principal:

1. L'Usuari ha seleccionat l'opció d'eliminar-se.
2. El sistema cerca si l'Usuari té receptes públiques i mostra missatge abans d'eliminar.

3. L'Usuari selecciona l'opció: eliminar totes les receptes o sols les no publicades.
4. El sistema cerca a l'Usuari i l'elimina de la base de dades.
5. El sistema executa el cd'ú 00.IniciarSessió.

Alternatives de procés i excepcions: cap.

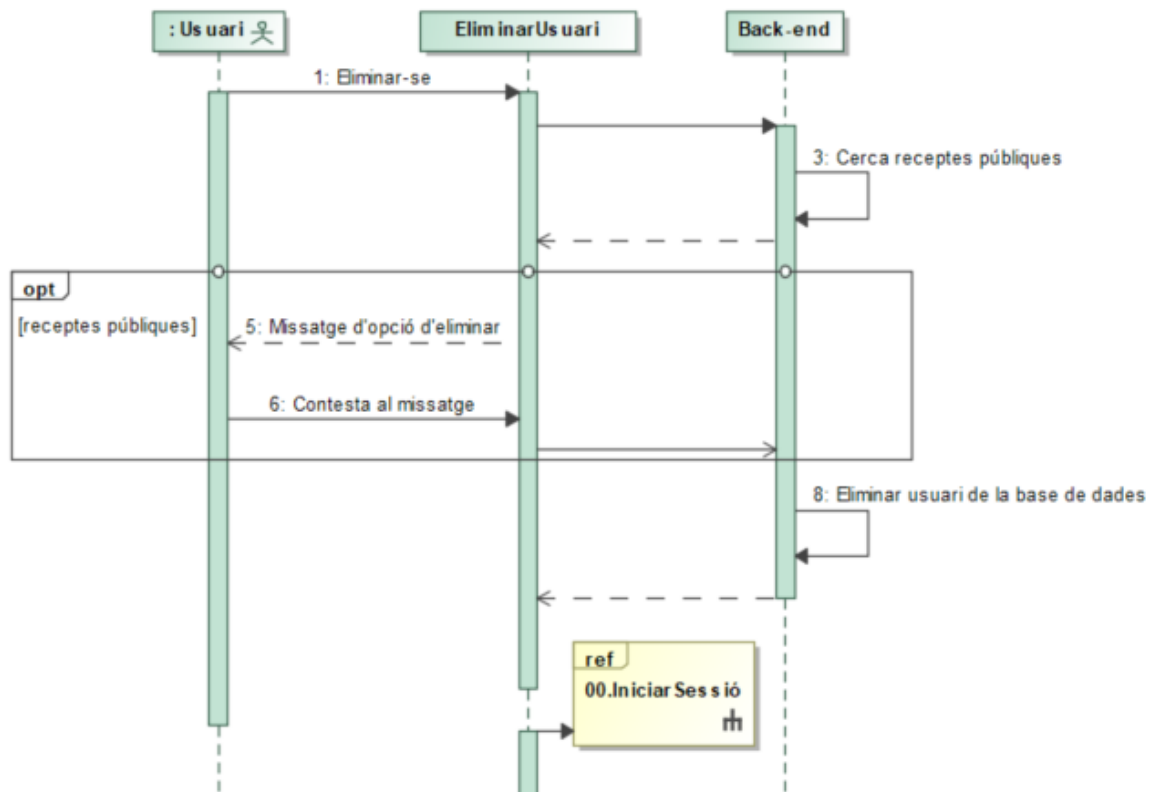


Figure 14: Seqüència del cas d'ús 09.EliminarUsuari

## Cd'ú 10.VisualitzarReceptesPúbliques

Resum de la funcionalitat: mostrar les receptes publicades pels usuaris.

Paràmetres d'entrada: identificador de l'Usuari.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: cap.

Postcondició: visualitzar l'historial de les receptes públiques o un text que indica que encara no se n'ha publicat cap.

Procés normal principal:

- 
1. El sistema consulta a la base de dades les receptes que estan compartides i no son de l'Usuari.
  2. El sistema mostra el llistat de receptes o un text si no hi ha res guardat.

Alternatives de procés i excepcions:

2a L'Usuari ha seleccionat per cercar les receptes.

2a1 El sistema executa el cd'ú 05.CercarRecepta.

2b L'Usuari selecciona una recepta del llistat.

2b1 El sistema executa el cd'ú 08.VisualitzarInformacióRecepta.

2c L'Usuari utilitza un filtre per al llistat (només si hi ha receptes al historial).

2c1 El sistema modifica el llistat en funció el filtre.

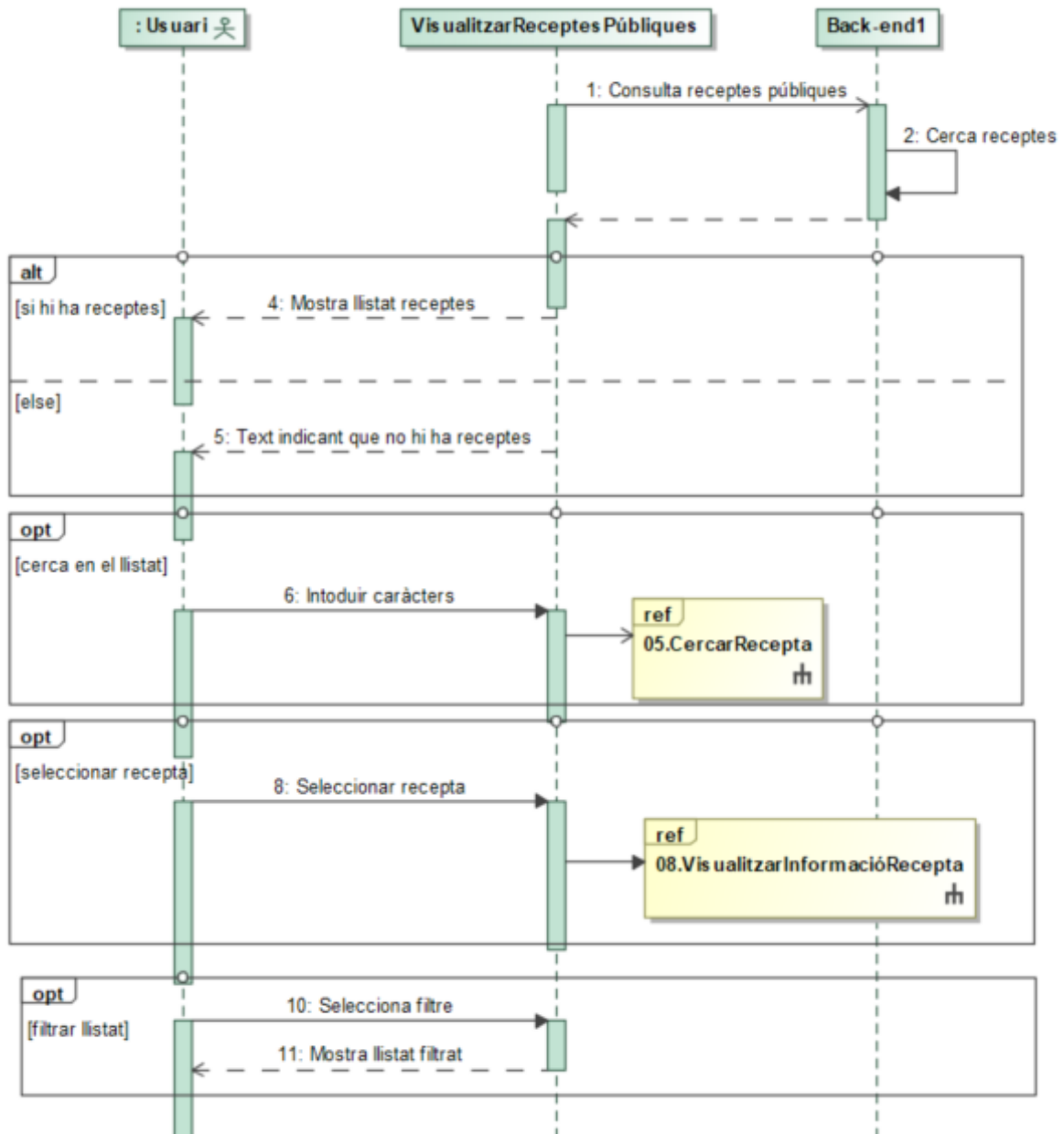


Figure 15: Seqüència del cas d'ús 10.VisualitzarReceptesPúbliques

## Cd'ú 11.TancarSessió

Resum de la funcionalitat: tancar la sessió activa de l'Usuari.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Usuaris: Usuari.

Precondició: tenir a l'Usuari actiu a l'aplicació.

Postcondició: tancar la sessió i tornar a la pàgina d'entrada de l'aplicació.

Procés normal principal:

1. L'Usuari ha seleccionat l'opció de tancar sessió.
2. El sistema deixa de recordar l'identificador de l'Usuari actiu.
3. El sistema executa el cd'ú 00.IniciarSessió.

Alternatives de procés i excepcions: cap.

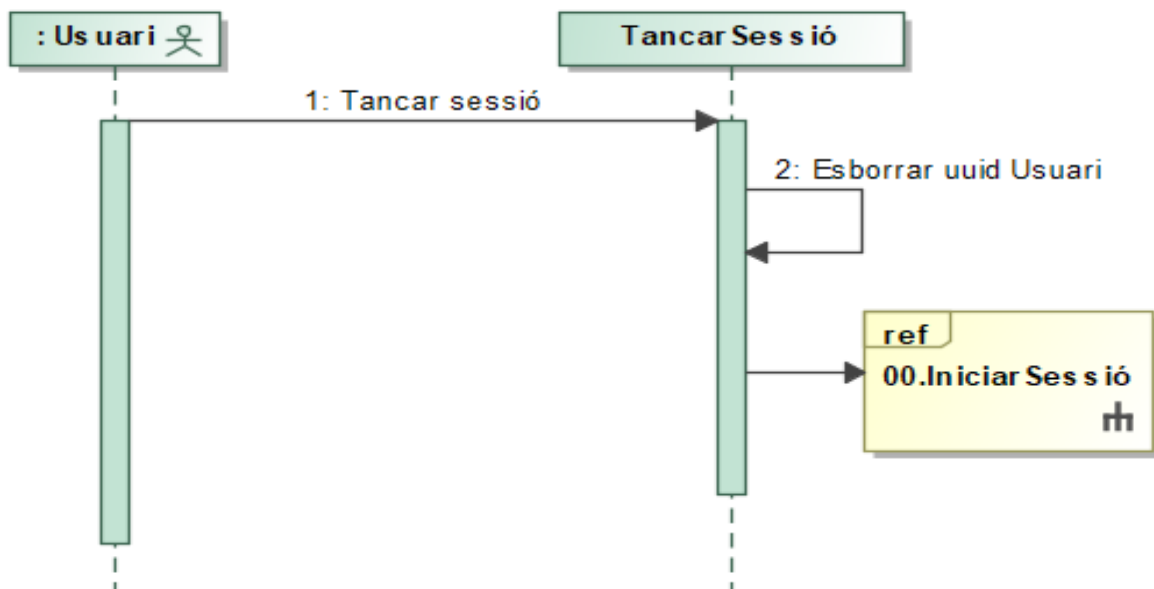


Figure 16: Seqüència del cas d'ús 11.TancarSessió

## 5.2 Requisits no funcionals

### 5.2.1 Requisit d'usabilitat

L'aplicació ha de ser intuïtiva i fàcil d'utilitzar per a tota mena d'usuaris, independentment del seu nivell tècnic o grup d'edat. La interfície d'usuari, desenvolupada amb Flutter, ha de ser senzilla i clara, oferint una experiència de navegació fluida que permeti als usuaris completar les seves tasques sense complicacions ni confusió.

Els usuaris han de poder introduir receptes o ingredients i obtenir els resultats dels macronutrients amb el mínim esforç, gràcies a l'organització ben estructurada dels elements a la pantalla, com les opcions clares per a introduir dades (Text, PDF, Imatge) i botons grans i fàcilment identificables.

---

A més, l'aplicació ha de guiar l'usuari en els passos necessaris per utilitzar funcions específiques, com la pujada d'imatges, PDF, o la consulta de l'historial de receptes, assegurant-se de que les funcionalitats són fàcilment accessibles i comprensibles. La disponibilitat de guies visuals i instruccions ha de garantir que els usuaris poden interactuar amb l'aplicació de manera efectiva i eficient, fomentant així la confiança i l'eficiència en la seva utilització.

Aquest requisit es considerarà complert si els usuaris poden utilitzar totes les funcionalitats de l'aplicació sense necessitat de suport addicional, contribuint a una experiència d'usuari òptima i una major satisfacció.

### **5.2.2 Requisit de rendiment**

L'aplicació ha de garantir un rendiment òptim en tot moment, assegurant temps de resposta adequats per a totes les operacions, especialment en la generació i consulta de macronutrients. És fonamental que la comunicació entre el front-end, desenvolupat amb Flutter, i el back-end, que utilitza Flask en Python, es realitzi de manera fluida i eficient, evitant qualsevol tipus de retard que pugui afectar l'experiència de l'usuari.

A més, s'ha de comprovar que les consultes a la base de dades PostgreSQL es processen de manera ràpida i sense interrupcions. El sistema ha de ser capaç de manejar múltiples peticions simultànies sense que això afecti el rendiment general de l'aplicació. Això inclou mantenir un temps de resposta consistent fins i tot quan hi hagi un augment en la càrrega de treball.

Aquest requisit es considerarà complert si els temps de resposta per a les operacions clau, com ara la generació de macronutrients i la consulta d'historials, es mantenen dins dels llindars establerts durant les proves de càrrega. La prioritat és assegurar una experiència d'usuari contínua i sense interrupcions, independentment de la demanda en temps real.

---

### 5.2.3 Requisit de seguretat

L'aplicació ha de garantir la seguretat de les dades de l'usuari, especialment tenint en compte que emmagatzema informació personal i l'historial de consultes. La seguretat s'ha de basar en les següents mesures:

- **Xifrat de la contrasenya:** Les contrasenyes dels usuaris es xifren utilitzant l'algorisme SHA-256. La contrasenya es converteix primer a bytes amb codificació UTF-8 i després es genera un hash SHA-256 a partir d'aquests bytes. Aquest hash és emmagatzemat en lloc de la contrasenya en text pla, garantint que les contrasenyes no es guardin de manera vulnerable.
- **Generació aleatòria de l'identificador d'usuari:** Els identificadors d'usuari es generen de manera aleatòria per evitar que es pugui deduir informació sobre altres usuaris o accedir a les dades d'usuari mitjançant una simple enumeració de comptes.
- **Accés restringit a les dades:** Els usuaris només poden accedir a les seves pròpies receptes, garantint que no es pugui accedir a dades d'altres usuaris.

Aquests requisits asseguren un nivell bàsic de seguretat per a la protecció de les dades de l'usuari i per prevenir accions malicioses. És recomanable revisar i actualitzar regularment les mesures de seguretat per abordar noves amenaces i vulnerabilitats que puguin sorgir.

### 5.2.4 Requisit de portabilitat

L'aplicació ha de ser compatible amb els sistemes operatius Android i iOS, gràcies a l'ús de Flutter, que permet desenvolupar aplicacions per a ambdues plataformes amb un únic codi base. Ha de funcionar correctament tant en emuladors com en dispositius mòbils físics.

---

A més, l'aplicació ha de poder ser fàcilment transferida i configurada en diferents entorns de desenvolupament i producció. Les dependències com el servidor Flask i la base de dades PostgreSQL han de ser senzilles de configurar i executar en diverses plataformes sense necessitat de canvis significatius.

### **5.2.5 Requisit de mantenibilitat**

L'aplicació ha de ser dissenyada per facilitar la seva mantenibilitat i actualització. Tant el codi del frontend en Flutter com el del backend en Flask han de seguir les millors pràctiques de programació, amb una estructura clara i ben documentada. Això permetrà als desenvolupadors fer modificacions, actualitzacions i correccions de manera eficient i sense complicacions.

A més, l'arquitectura del sistema ha de permetre afegir noves funcionalitats i resoldre problemes amb mínims ajustos. La utilització de patrons de disseny adequats i la separació de responsabilitats entre el front-end i el back-end facilitaran aquests canvis, assegurant que les actualitzacions es facin de manera segura i sense afectar la resta del sistema.

### **5.2.6 Requisit d'escalabilitat**

L'aplicació ha de ser dissenyada per manejar l'augment potencial en la càrrega de treball i el nombre d'usuaris sense perdre rendiment. Encara que les proves actuals s'han realitzat amb un usuari actiu, l'arquitectura del sistema ha de permetre una escalabilitat efectiva. Això inclou la capacitat de gestionar més usuaris i dades a mesura que l'aplicació creixi en el futur.

El sistema ha de ser dissenyat tenint en compte que podrà ser necessari augmentar els recursos, com ara la potència del servidor o l'emmagatzematge, i ajustar la base de dades per manejar una major quantitat de dades. La infraestructura ha de ser flexible i adaptar-se a les necessitats canviants dels usuaris, garantint que l'aplicació mantingui un bon rendiment mentre creix.

---

## 6 Disseny

En aquest apartat es detallen els diferents aspectes referents a l'arquitectura de l'aplicació, el disseny de la interfície de l'usuari, disseny de les classes, el disseny de la base de dades i el disseny de l'API.

### 6.1 Arquitectura de l'aplicació

Com a tot projecte que desenvolupa una aplicació hi han tres capes que intervenen en l'arquitectura: capa de presentació (front-end), capa de lògica de negoci (back-end) i capa de dades (base de dades). L'arquitectura de tres capes permet una separació clara de responsabilitats i facilita la mantenibilitat, escalabilitat i reutilització de codi. Aquestes tres capes es comuniquen de la següent forma:

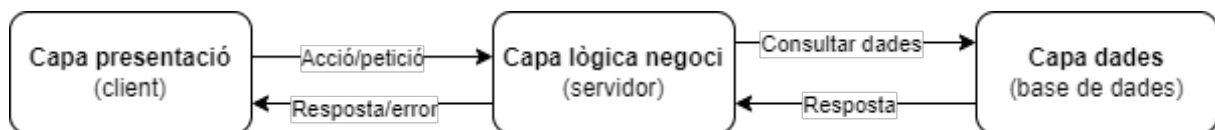


Figure 17: Diagrama arquitectura 3 capes

#### Capa de presentació

Per desenvolupar la interfície d'usuari de l'aplicació he utilitzat el llenguatge Flutter. Em vaig decantar per aquest pel fet que, en un únic codi, es pot adaptar tant a dispositius Android com iOS. La capacitat d'aquest llenguatge per crear aplicacions natives per ambdues plataformes des d'una sola base de codi em va semblar una avantatge clau en comparació amb altres opcions.

Tot i no haver-lo utilitzat mai, vaig decidir explorar-lo per les seves nombroses ventatges. En altres assignatures, havia treballat amb Android Studio i React, però vaig escollir Flutter després de fer una investigació exhaustiva sobre les seves possibilitats i els beneficis que podia aportar al projecte. He mencionat que em va cridar l'atenció el fet de poder ser util-

---

itzat per les dues plataformes, però React també pot ser usat per ambdues. Amb això, els motius a destacar per la meva tria són els següents:

- Compila directament a codi natiu, cosa que proporciona un rendiment gairebé idèntic al d'una aplicació nativa.
- Els widgets propis asseguren una aparença i comportament uniformes a totes les plataformes, facilitant un control total sobre el disseny.
- La funcionalitat "Hot Reload" permet veure els canvis a l'instant sense perdre l'estat de l'aplicació i accelerant el procés de desenvolupament i depuració.
- Permet crear interfícies d'usuari altament personalitzades i complexes amb gran facilitat, gràcies al seu motor gràfic.

### **Capa de lògica de negoci**

Per a la lògica de negoci, vaig decantar-me pel llenguatge Python, ja que m'ofereix una flexibilitat en el desenvolupament d'algoritmes i processos de dades. En un inici, em vaig plantejar connectar-me a l'API de ChatGpt per aconseguir amb més facilitat el llistat d'ingredients que es troben a un PDF. Tot i això, a causa del cost associat, vaig investigar l'ús de llibreries de Python per extraure el text i després desenvolupar un algoritme per obtenir el llistat.

En la recerca de llibreries de Python, em vaig trobar amb PyPDF2 per manipular PDFs. A posteriori, quan vaig incorporar el tractament d'imatges, vaig haver de buscar quina llibreria utilitzar, i vaig trobar pytesseract. Ambdues m'han agilitzat la feina de tractament dels diferents formats de dades per obtenir el llistat d'ingredients.

Posteriorment, vaig investigar com connectar el front-end amb Flutter i el back-end amb Python. Vaig esbrinar que Flask, un microframework, permet la comunicació entre aquestes dues capes. També facilita mantenir la coherència en el llenguatge utilitzat i aprofita la seva simplicitat per

---

construir una API RESTFUL que permet la interacció entre l'aplicació mòbil i el servidor de manera eficient i segura.

Encara que podria haver optat per un llenguatge més conegut en l'entorn acadèmic, com Java, vaig considerar que Python m'oferia una major flexibilitat en la programació. Concretament, aquesta flexibilitat es reflecteix en la facilitat de tractar les dades i la rapidesa del desenvolupament, ja que té una sintaxi senzilla i una gran varietat de llibreries per agilitzar tasques específiques, com les dues mencionades anteriorment.

### **Capa de dades**

Per gestionar les dades de l'aplicació, vaig decidir utilitzar PostgreSQL com a sistema de bases de dades. Aquesta elecció es va basar en diverses consideracions importants. En primer lloc, PostgreSQL és una base de dades relacional que ofereix robustesa i escalabilitat, aspectes crucials per garantir que l'aplicació pugui manejar un volum significatiu de dades de manera eficient i fiable.

Una de les raons per les quals vaig triar PostgreSQL és la seva interfície administrativa, que és molt visual i intuïtiva, cosa que em permet gestionar les dades de manera més còmoda i ràpida. Ja havia treballat amb PostgreSQL a una altra assignatura, fet que em va proporcionar la confiança necessària per continuar utilitzant-la en aquest projecte.

A més, PostgreSQL s'integra fàcilment amb Python, cosa que em permet mantenir una coherència tecnològica al llarg de tot el projecte. Encara que hi ha altres opcions viables tant en bases de dades relacionals com no relacionals, vaig considerar que PostgreSQL era la millor elecció per a aquest projecte, tenint en compte les necessitats específiques de la meva aplicació i la meva experiència prèvia amb aquesta eina.

---

## 6.2 Disseny de la interfície de l'usuari

El disseny de la interfície gràfica ha estat pensat en tot moment per ser senzill i intuïtiu, amb l'objectiu de facilitar l'ús de l'aplicació a les persones usuàries. Abans de la implementació del codi, vaig dissenyar les pantalles principals: inici de sessió, registre d'usuari, la pantalla principal (on l'usuari pot seleccionar la forma en què introduirà els ingredients de la recepta), l'historial de les receptes i informació de la recepta.

Posteriorment, durant la fase d'implementació, es van afegir altres pantalles, com la visualització dels ingredients emmagatzemats a la base de dades i el llistat de receptes publicades pels usuaris. També es van incorporar noves funcionalitats a les pantalles dissenyades, com ara el càlcul dels macronutrients individuals a la pantalla que mostra la informació dels ingredients, així com la possibilitat de votar les receptes publicades per altres usuaris, filtrar la llista de receptes i eliminar l'usuari.

Inicialment, els dissenys van ser esbossats en una llibreta, però per tal de fer-los més visuals, vaig utilitzar Figma, una aplicació per dissenyar interfícies.

## 6.2.1 Prototips

NOM APLICACIÓ

### Iniciar sessió

✓ iniciar session    registrarse

Nombre

Contraseña

iniciar sessió

Figure 18: Pantalla inici de sessió

NOM APLICACIÓ

### Registrarse

iniciar session    ✓ registrarse

Nombre

Contraseña

Confirmar contraseña

registrarse

Figure 19: Pantalla registrar-se

NOM APLICACIÓ

✓ Text    PDF    Imagen

Nombre receta

Format introduir text de la recepta

Resposta amb la informació dels macronutrients

Figure 20: Pantalla principal

NOM APLICACIÓ

Buscar en el historial    Q

🗑 Receta 1    →

🗑 Receta 2    →

🗑 Receta 3    →

Figure 21: Pantalla historial

---

NOM APLICACIÓ

# Nom recepta

Llistat d'ingredients

Informació dels macronutrients

Figure 22: Pantalla informació

## 6.2.2 Disseny

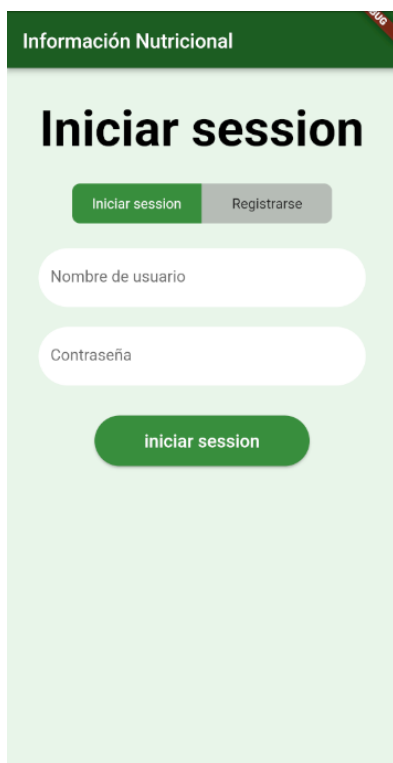


Figure 23: Pantalla inici de sessió

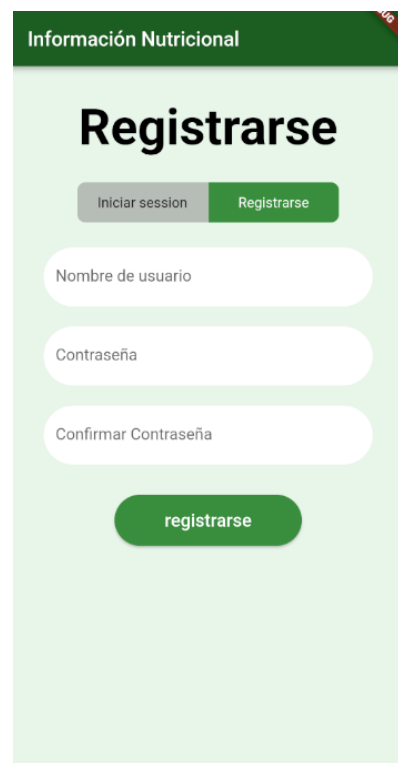


Figure 24: Pantalla registrar-se

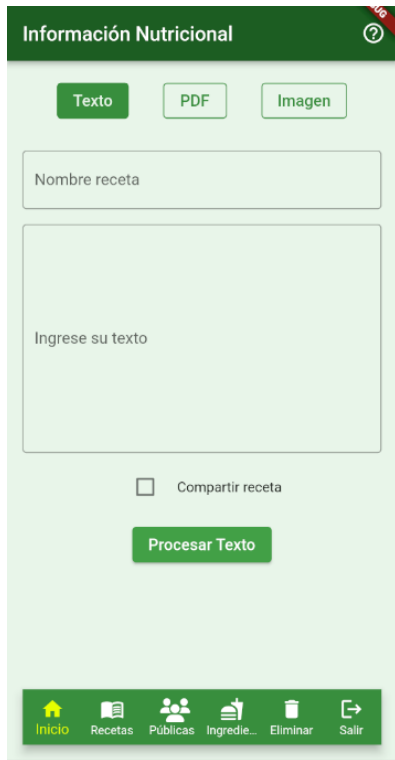


Figure 25: Pantalla principal text

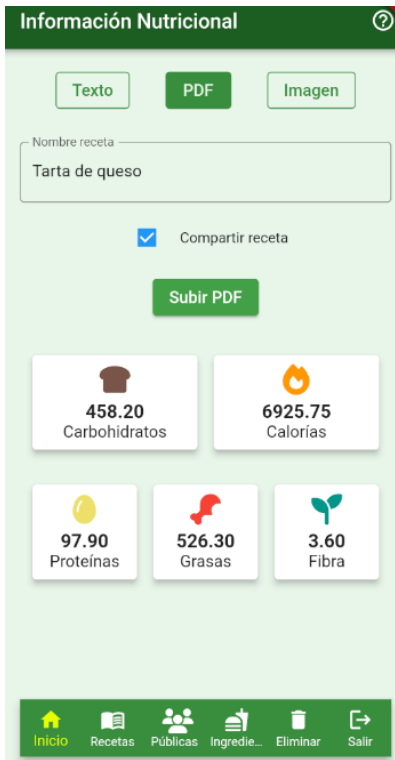


Figure 26: Pantalla principal pdf

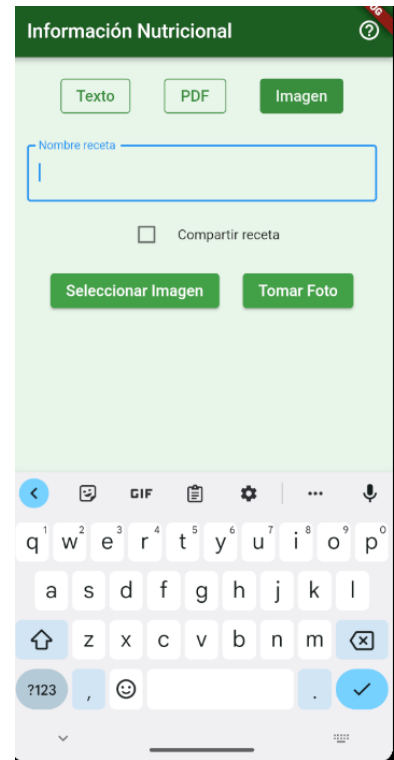


Figure 27: Pantalla principal imatge

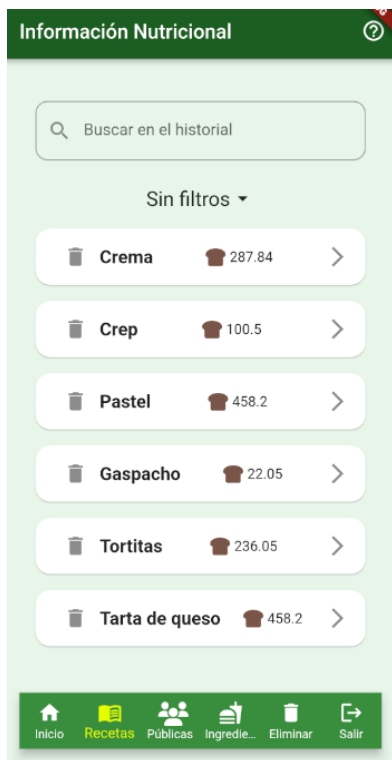


Figure 28: Pantalla historial de recetas

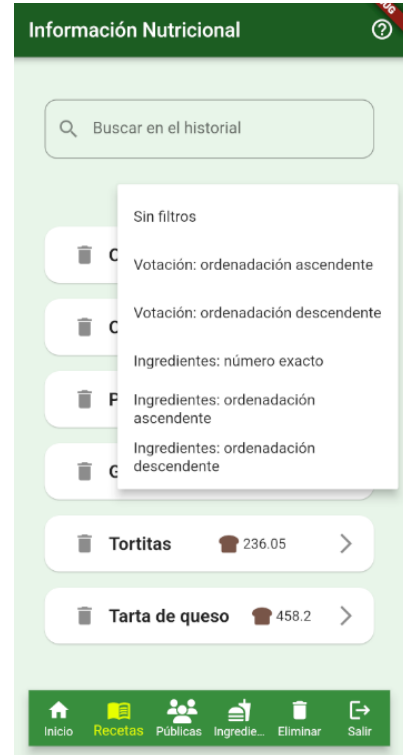


Figure 29: Pantalla historial de recetas (2)

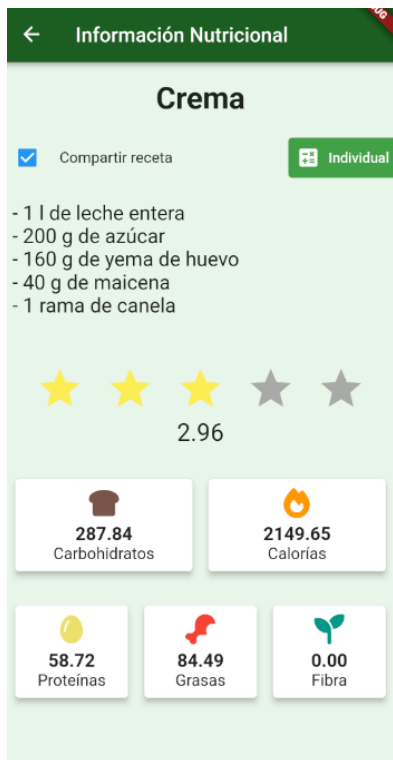


Figure 30: Pantalla informació recepta

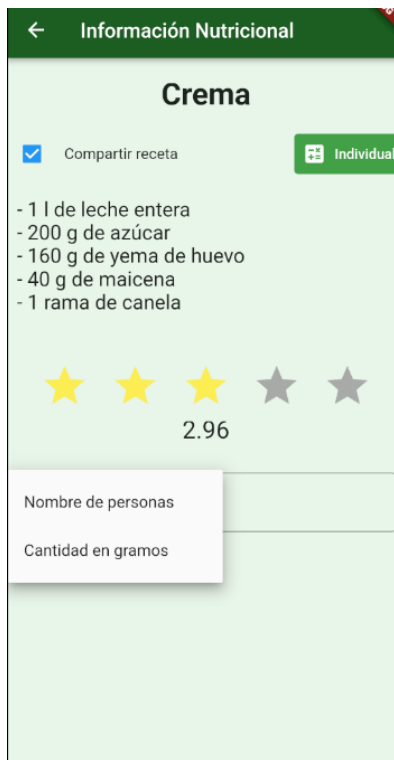


Figure 31: Pantalla informació recepta (2)

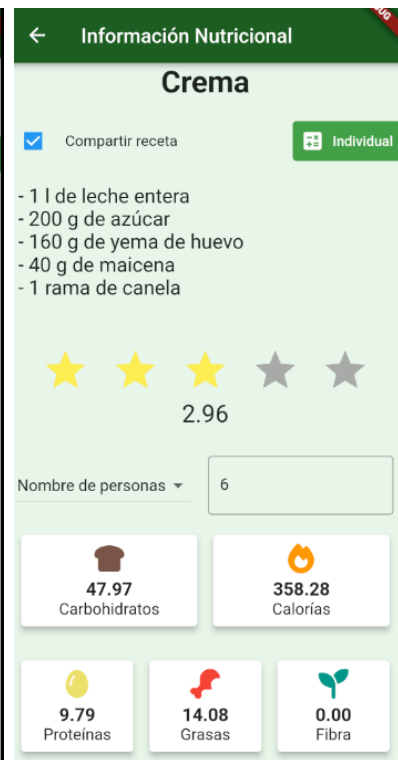


Figure 32: Pantalla informació recepta (3)



Figure 33: Pantalla receptes públiques



Figure 34: Pantalla receptes públiques (aplicant un filtre)

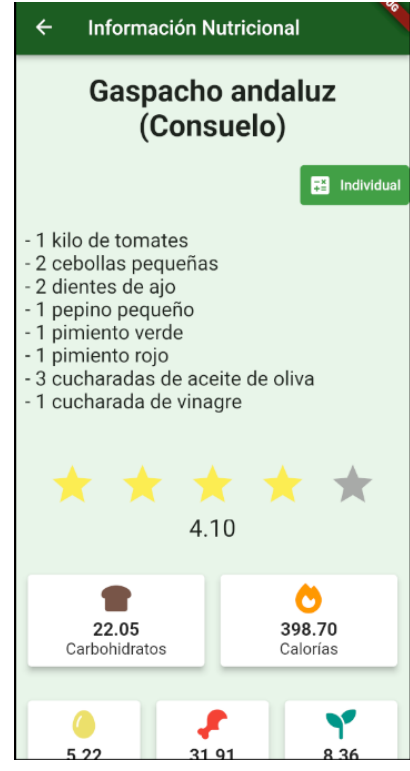


Figure 35: Pantalla informació recepta d'un altre usuari

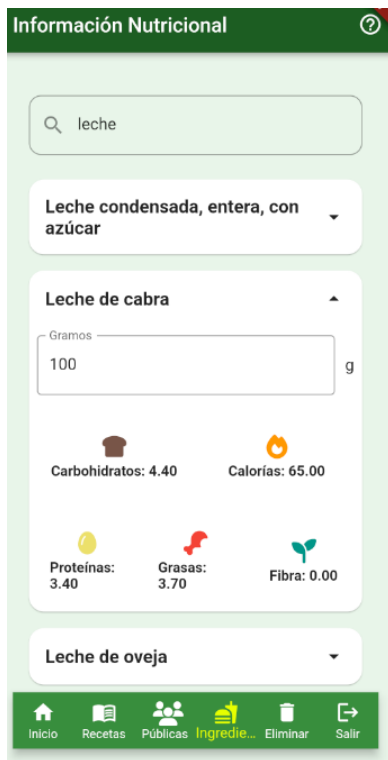


Figure 36: Pantalla llistat ingredients



Figure 37: Pantalla d'ajuda

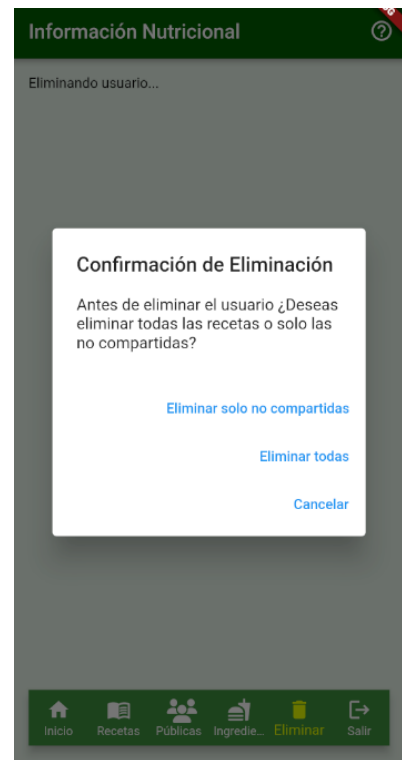


Figure 38: Pantalla eliminar usuari

### 6.3 Disseny de classes

El back-end es basa en els mètodes de l'API, per tant, no té cap classe perquè se centra en la gestió de sol·licituds i dades. El disseny de classes es reflecteix exclusivament al front-end, on cada pantalla es considera una classe, amb la següent estructura i funcionalitat principal:

- **main:** pàgina principal que decideix si mostrar LoginForm o ConditionalForm en funció si hi ha un usuari actiu.
- **MyHomePage:** s'utilitza com a medidora entre les diferents pàgines accessibles dins de l'aplicació: ConditionalForm, HistorialPage i ListFoodPage.
- **LoginForm:** formulari per iniciar sessió o registrar un nou usuari.
- **ConditionalForm:** pàgina principal de l'aplicació on permet seleccionar el format de les dades (text, imatge o pdf) i s'obté els macronu-

---

trients.

- **HistorialPage:** pàgina per consultar les receptes guardades per l'usuari i inclou funcionalitats per cercar receptes específiques.
- **ListFoodPage:** pàgina que mostra els ingredients de la base de dades, amb informació detallada sobre cada ingredient.
- **InfoPage:** mostra la informació detallada d'una recepta.
- **ReceiptPublicPage:** pàgina que permet veure les receptes publicades per altres usuaris, amb funcionalitats per cercar i ordenar la llista.
- **DeleteUserPage:** pàgina intermediària que apareix abans d'eliminar un usuari, per comprovar si hi ha receptes publicades i decidir si s'han d'eliminar.
- **HelpPage:** pàgina que ofereix informació sobre les funcionalitats disponibles a l'aplicació, amb l'objectiu d'ajudar l'usuari.

### 6.3.1 Diagrama de classes

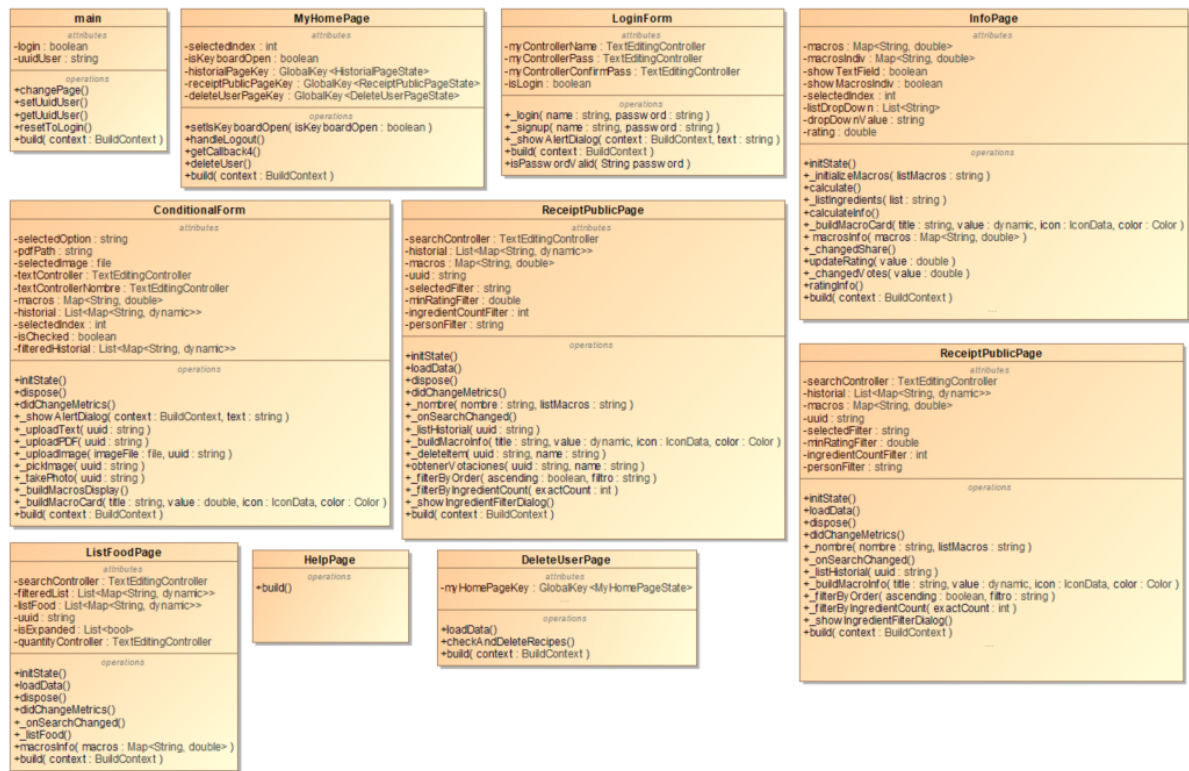


Figure 39: Diagrama de classes

### 6.4 Disseny de la persistència de dades

La base de dades amb PostgreSQL m'ha permès aconseguir la persistència de les dades. Tot i que està formada per quatre taules independents, em permet manipular la informació de manera adequada:

- **alimentos:** aquí es guarda la informació dels macronutrients (calories, hidrats de carboni, fibra, proteïna i greix) dels aliments per 100 grams.
- **equivalencias:** aquesta taula, tot i estar molt "hardcodejada", s'ha creat per especificar quantitats concretes d'alguns aliments com ara ous o cogombre "petit", en descripcions específiques.
- **historial:** emmagatzema les receptes dels usuaris amb tota la informació relacionada, incloent-hi els ingredients, els macronutrients,

---

i l'usuari que les ha guardat, identificat pel seu identificador únic. També emmagatzema dades sobre si l'usuari ha publicat la recepta, el pes d'aquesta en grams, i la mitjana de les votacions rebudes.

- **usuarios:** guarda la informació necessària dels usuaris que utilitzen l'aplicació. Per raons de privacitat, la clau d'accés d'aquests usuaris està encriptada.

A continuació, es pot observar el diagrama de les taules mencionades de la base de dades, on es pot observar el format dels atributs:

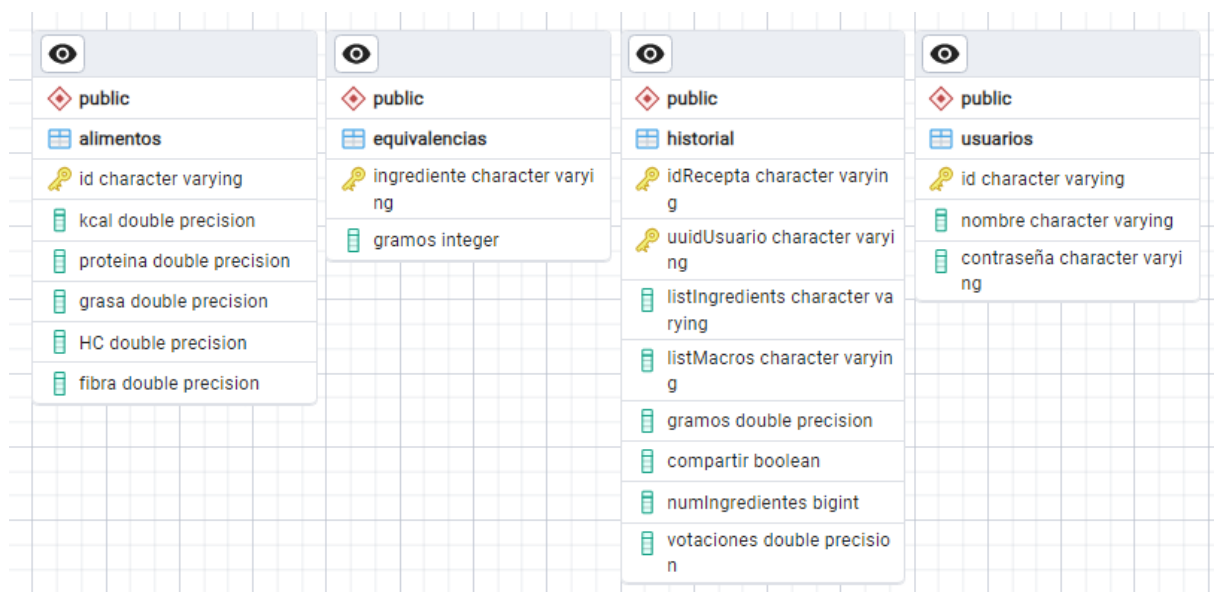


Figure 40: Diagrama base de dades

## 6.5 Disseny de l'API

La comunicació entre el client i el servidor es realitza mitjançant els següents mètodes de l'API:

- **POST /upload:** guarda la informació pertinent a una recepta publicada per l'usuari actiu. Realitza certes tasques en funció del tipus de dades (text, pdf o imatge), però el resultat final és el mateix. El cos de la sol·licitud ha de tenir el tipus de dades que s'envia, la recepta, el nom de la recepta i l'identificador de l'usuari. Retorna els

---

macronutrients de la recepta que s'ha introduït o el codi d'error si hi ha un problema en alguna part de l'execució.

- **POST /login:** comprova les credencials de l'usuari per permetre l'accés al sistema. El cos de la sol·licitud ha de tenir el contingut del nom de l'usuari i la contrasenya. Retorna informació de l'usuari en cas d'autenticació correcta i un codi d'error 404 si les credencials són incorrectes.
- **POST /signup:** registra un nou usuari al sistema, si no n'existeix un altre amb el mateix nom. El cos de la sol·licitud té el nom d'usuari, la contrasenya i la comprovació d'aquesta. Retorna l'identificador de l'usuari que s'ha generat de manera automàtica amb el format UUID.
- **GET /historial:** retorna el llistat de les receptes de l'usuari actiu. Tot i ser un metode GET s'ha de proporcionar l'identificador de l'usuari actiu, i la seva resposta és el llistat de les receptes d'aquest.
- **GET /list:** cerca i retorna tots els ingredients disponibles a la base de dades. No requereix cap dada al cos de la sol·licitud i la seva resposta és el llistat d'ingredients.
- **DELETE /deleteItem:** elimina la recepta especificada per l'usuari actiu. El cos de la sol·licitud té l'identificador de l'usuari i el nom de la recepta, la seva resposta és el llistat modificat de les receptes.
- **GET /receiptPublic:** retorna el llistat de les receptes que estan publicades, excloent-hi les de l'usuari actual, ja que l'objectiu és visualitzar receptes d'altres usuaris.
- **PUT /changedShare:** modifica l'estat de compartició de la recepta de l'usuari, invertint el valor actual. Si la recepta és pública, passa a ser privada, i viceversa.
- **GET /checkSharedRecipes:** consulta si l'usuari té receptes publicades.

- 
- **DELETE /deleteNonSharedRecipes:** elimina de la base de dades només aquelles receptes que no són públiques.
  - **DELETE /deleteAllRecipes:** elimina totes les receptes de l'usuari.
  - **DELETE /deleteUser:** elimina l'usuari de la base de dades.
  - **GET /getVotes:** obté el valor de les votacions de la recepta de l'usuari que s'està consultant.
  - **PUT /changedVotes:** actualitza el valor de les votacions de la recepta de l'usuari que s'està consultant.

---

## 7 Implementació

En aquest apartat es troben els detalls de la tecnologia, algorismes específics i com s'ha desenvolupat el projecte diferenciant-ho per fases.

### 7.1 Detalls de la tecnologia

Tot i haver esmentat quins llenguatges he utilitzat tant per al front-end com per al back-end, en aquest apartat aprofundiré en els detalls de les llibreries i paquets que han estat essencials en el desenvolupament.

#### 7.1.1 Back-end

A continuació, detallo les llibreries que he utilitzat en Python per al desenvolupament del back-end, juntament amb una explicació del motiu pel qual s'han emprat:

- **re:**

Mòdul de Python que s'utilitza per manipular expressions regulars. L'he utilitzat per identificar i extreure seccions específiques del text, concretament l'extracció del llistat d'ingredients a partir d'un PDF o d'imatges que s'han convertit en text.

- **PyPDF2:**

Aquesta llibreria permet la lectura i manipulació de fitxers PDF, l'he utilitzat per extreure el text d'aquests i poder processar-los.

- **psycopg2:**

Llibreria que m'ha permès la connexió i interacció amb la base de dades PostgreSQL des de Python, concretament l'he utilitzat per executar consultes d'emmagatzematge d'informació en les diverses taules de la base de dades.

---

- **Flask:**

És un microframework que permet la comunicació entre el back-end i el front-end.

- **request:**

S'utilitza per gestionar les dades enviades pel client en sol·licituds HTTP.

- **jsonify:**

Serveix per retornar respostes en format JSON, és utilitzat quan és vol enviar dades amb un format i no sols una única dada, per exemple: enviant al client les dades pertinents de la recepta on hi ha el nom, llistat d'ingredients i llistat de macronutrients.

- **CORS:**

Mòdul del paquet *flask\_cors* que s'utilitza per habilitar el CORS (Cross-Origin Resource Sharing) a les API de Flask, la qual cosa permet que les sol·licituds HTTP de recursos es puguin realitzar des d'orients diferents. Això ha estat utilitzat per permetre al front-end interactuar amb el back-end que està allotjat a un servidor diferent.

- **secure\_filename:**

És una funció de *werkzeug.utils* que s'utilitza per assegurar que els noms de fitxer carregats siguin segurs i no puguin causar problemes de seguretat. Aquesta funció ha estat crucial per gestionar de manera segura les càrregues de fitxers a través del sistema de fitxers del servidor.

- **os:**

És una llibreria estàndard de Python que proporciona funcions per interactuar amb el sistema operatiu, com la creació de directoris o la manipulació de fitxers i rutes. En aquest projecte, s'ha utilitzat per gestionar els directoris de càrrega de fitxers i comprovar que existeixen abans de desar-hi documents.

---

- **uuid:**

Llibreria que genera identificadors únics universals (UUIDs). L'he utilitzat per generar identificadors únics per als usuaris i evitar tenir un identificador continu per garantir la seguretat de l'usuari.

- **pytesseract:**

Interfície de Python utilitzada al motor de reconeixement òptic de caràcters (OCR) Tesseract. S'ha utilitzat per extreure el text d'imatges, la qual cosa permet processar receptes que s'introduïxin en format fotografia.

- **Image:**

El mòdul de la llibreria PIL s'ha utilitzat conjuntament amb la llibreria anterior per obrir i manipular les imatges abans de processar-les amb OCR.

També voldria esmentar les llibreries que he utilitzat per incorporar els aliments a la base de dades. Les dades s'han extret d'un Excel que vaig crear a partir d'un PDF utilitzat per estudiants de nutrició. També he creat un Excel per construir la taula d'equivalències, la qual he utilitzat per indicar els pesos d'ingredients que no poden estar especificats de manera precisa, com ara els ous. Tot i que aquests fitxers són independents, també mereixen una menció:

- **pandas:**

Llibreria essencial per a la manipulació de dades en forma de DataFrame. L'he utilitzat per llegir els fitxers Excel i processar les dades abans de carregar-les a la base de dades.

- **sqlalchemy (create\_engine, MetaData, Table, Column, String, Float):**

Aquest conjunt de components de SQLAlchemy permet la creació i manipulació de bases de dades a través d'un ORM (Object-Relational Mapping). L'he utilitzat per definir l'estructura de la taula d'aliments i gestionar la connexió a la base de dades.

---

- **sqlalchemy.exc (SQLAlchemyError):**

Part del paquet SQLAlchemy, aquesta classe d'errors permet gestionar excepcions durant les operacions amb la base de dades. L'he utilitzada per capturar i gestionar errors en la creació de taules i la inserció de dades.

### 7.1.2 Front-end

Paquets de Flutter que s'han usat i el motiu del seu ús:

- **http:** Aquest paquet s'utilitza per realitzar sol·licituds HTTP com GET, POST, DELETE, etc., permetent la comunicació amb l'API. Ha estat fonamental per establir una relació de petició i resposta entre el front-end i el back-end.
- **font\_awesome\_flutter:** Permet afegir icones atractives, millorant l'experiència de l'usuari i aconseguint una interfície intuïtiva i visualment atractiva. Inclou una àmplia col·lecció d'icones d'una de les biblioteques més utilitzades en el disseny d'interfícies.
- **file\_picker:** Permet als usuaris seleccionar fitxers PDF des del sistema de fitxers. Ha estat útil per manipular el format de dades d'una recepta a partir d'un PDF.
- **image\_picker:** Aquest paquet permet als usuaris seleccionar imatges de la galeria o capturar-ne de noves amb la càmera del dispositiu. Ha estat fonamental per obtenir receptes en format d'imatge i poder-les processar.
- **crypto:** Aquest paquet inclou una varietat d'algoritmes criptogràfics com hash (MD5, SHA-1, SHA-256, etc.) i encriptació. Ha estat crucial per assegurar que les dades sensibles, com les contrasenyes dels usuaris, no puguin ser manipulades o interceptades per terceres persones.

- 
- **toggle\_switch:** Proporciona una manera fàcil d'implementar botons de tipus "toggle". L'he utilitzat a les pàgines d'inici de sessió i registre per permetre l'alternança entre aquestes dues pàgines, oferint a l'usuari una experiència clara i visualment atractiva.
  - **flutter\_pannable\_rating\_bar:** Permet als usuaris seleccionar una puntuació mitjançant una interfície interactiva. En el meu cas, l'he utilitzat per mostrar cinc estrelles que representen la puntuació d'una recepta. Els usuaris poden tocar o arrossegat sobre les estrelles per ajustar la seva valoració, i la puntuació seleccionada es guarda i s'envia a la base de dades.

## 7.2 Algoritmes específics

En aquest apartat tractaré els algoritmes més importants o destacables del projecte, especialment aquells que formen part de la lògica implementada al back-end. La major part de la complexitat del sistema es troba en aquesta part del projecte, on es gestionen i processen les dades a través de l'API.

L'API amb els seus mètodes està detallada en la part pertinent del disseny. En aquesta secció, em centraré en donar més detall sobre els algoritmes més rellevants que processen i manipulen les dades, destacant el funcionament intern d'aquests mètodes.

### 7.2.1 Mètode upload

Aquest mètode s'utilitza per processar els ingredients de les receptes i obtenir els macronutrients corresponents de la base de dades PostgreSQL. Si la informació es proporciona en format text, ja disposem del llistat d'ingredients. No obstant això, quan s'empra el format PDF o imatge, primer cal extreure el text per després obtenir la llista.

En el cas dels PDFs, s'ha utilitzat la llibreria PyPDF2 per processar el text

---

pàgina per pàgina, amb el següent codi:

```
def extract_text_from_pdf(pdf_path):
    text = ""
    with open(pdf_path, "rb") as file:
        reader = PyPDF2.PdfReader(file)
        num_pages = len(reader.pages)
        for page_num in range(num_pages):
            page = reader.pages[page_num]
            text += page.extract_text()
    return text
```

Listing 1: Extracció de text des d'un PDF

Per a imatges, s'ha fet servir la llibreria pytesseract juntament amb PIL per obrir la imatge i extreure'n el text. A més, s'ha configurat l'idioma a espanyol per permetre els accents, ja que de moment els textos que processa el prototip són en castellà:

```
pytesseract.pytesseract.tesseract_cmd = r'C:\Program
Files\Tesseract-OCR\tesseract.exe'
image = Image.open(filepath)
text = pytesseract.image_to_string(image, lang="spa")
lines = text.splitlines()
\end{minted}
```

Listing 2: Extracció de text des d'una imatge

En tots dos casos, s'ha filtrat el text per obtenir únicament el que es troba entre l'apartat d'ingredients i el de preparació, obtenint així un filtratge de línies. Aquest procediment és òptim per a la majoria de receptes, on el text està dividit en les seccions esmentades; en cas contrari, la localització de la llista d'ingredients es complica. Un cop tenim el llistat, també eliminem caràcters innecessaris com guions o espais en blanc que no corresponen.

El filtratge de línies es fa amb una expressió regular que identifica els patrons comuns en la majoria de receptes. Aquesta expressió facilita l'extracció precisa de la informació necessària:

```
patron_ingredientes = r"(?=Ingredientes|INGREDIENTES)\s*(.+?)
(?:=Preparaci n|PREPARACI N|Elaboraci n|ELABORACI N|Receta|Pasos
```

---

```
|Procedimiento|$)"
```

Listing 3: Filtratge de línies

En aquest punt, ja disposem del llistat d'ingredients per aconseguir la informació específica. La funció `obtenerCantidadIngrediente` processa les tres parts essencials: primer, es troba la quantitat; després, la mesura; i finalment, l'ingredient. Com que la informació a la base de dades es processa en grams, primer es realitza una conversió utilitzant un diccionari d'unitats, com es mostra a continuació:

```
unidades_medida = {
    'g': 1,
    'gr': 1,
    'grs.': 1,
    'gramos': 1,
    'kilo': 1000,
    'kg': 1000,
    'l': 1000,
    'litro': 1000,
    'ml': 1,
    'cucharadas': 10,
    'cucharada': 10,
    'rama': 5,
    'sobre': 14,
    'dientes': 5,
    'tarros': 125
}
```

Listing 4: Diccionari d'unitats

Després de processar aquesta informació, es procedeix a obtenir els macronutrients de la recepta mitjançant la funció `obtenerMacros`. Aquesta funció busca a la base de dades els ingredients corresponents per obtenir-ne la informació nutricional i calcular els valors en funció de la quantitat del producte a la recepta. Cal destacar que l'algoritme pot afinar la cerca segons les paraules clau que empri l'usuari (ex. llet condensada o desnatada), amb l'objectiu de trobar l'ingredient que millor s'adapti a les propietats necessàries.

---

### 7.2.2 Mètode historial i list

Aquests dos mètodes són destacables per la seva utilitat a l'hora de guardar les dades necessàries en un format JSON, facilitant així la manipulació a la part del front-end. A la funció historial, de cada recepta se'n extreu el nom, el llistat d'ingredients, el llistat de macronutrients, entre altres valors importants. Pel que fa al mètode list, aquest emmagatzema de manera individual el nom de l'ingredient i els seus macronutrients: kcal, proteïnes, greixos, hidrats de carboni i fibra.

### 7.2.3 Mètode checkSharedRecipes

En aquest mètode vull destacar la consulta a la base de dades que he realitzat. En aquest cas, no he utilitzat un *SELECT* com en altres consultes, sinó que l'he modificat per comprovar si existeix almenys una instància que compleixi el filtre. Concretament, la consulta és la següent:

```
cur.execute(  
    "SELECT EXISTS(  
        SELECT 1  
        FROM historial  
        WHERE \"compartir\"=true and \"uuidUsuario\"=%s)"  
    , (uuid,)  
)
```

Listing 5: Consulta mètode checkSharedRecipes

## 7.3 Estructura i interfície d'usuari

En el cas del front-end, no hi ha una lògica de dades sinó el procesament de peticions per part de l'usuari i la resposta que es rep del servidor. Per tant, en aquest apartat entraré en detall sobre els elements visuals utilitzats juntament amb el codi més rellevant de les diverses classes anteriorment esmentades.

---

### 7.3.1 Classe main

És la primera classe que es mostra a l'aplicació. Aquesta controla, mitjançant una variable booleana, si s'ha de mostrar la pantalla d'inici de sessió/registre o la pantalla inicial de l'aplicació. Aquesta funció que permet canviar l'estat es accessible des de qualsevol pàgina, ja que la passem a la resta de classes com a paràmetre.

A més, aquesta classe conté la variable de l'identificador de l'usuari, així com la funció per canviar-lo i obtenir-ne el valor. S'ha inclòs aquí per facilitar-ne la manipulació des de qualsevol classe, ja que pot ser necessari accedir-hi en diverses parts de l'aplicació.

### 7.3.2 Classe MyHomePage

Aquesta classe actua com a medidora entre les diferents pantalles de l'aplicació, ja que volem que a totes es mostri el menú de navegació. Per tant, aquesta classe està referenciada a la classe main i rep com a paràmetres les funcions esmentades anteriorment: canviar o obtenir l'identificador de l'usuari, així com actualitzar el valor de la variable booleana.

Per controlar la navegació entre pàgines, s'utilitza una variable índex. També es fa ús d'una variable booleana per determinar si el teclat està obert i, en cas que ho estigui, el menú no es mostra, ja que el teclat pot arribar a desplaçar els elements cap amunt, desestructurant el contingut de la pantalla.

A la pàgina de l'historial, per a poder actualitzar el llistat un cop s'hi vulgui accedir, s'ha fet servir una clau per tal d'identificar la pàgina de l'historial que permet executar una funció de la classe `HistorialPage`, concretament la d'actualitzar la llista de receptes guardades. Sense aquesta actualització, només es mostraria la llista carregada en iniciar l'aplicació.

---

Aquest canvi també s'hauria de considerar al llistat d'ingredients en cas d'actualització de la base de dades afegint-hi nous ingredients. Però, de moment, es manté la llista existent, que ja compta amb uns 700 ingredients. Això és una idea per a futures modificacions del prototip.

### 7.3.3 Classe LoginForm

Aquesta classe gestiona les pàgines d'inici de sessió i registre, controlant quina ha d'estar activa visualment amb un botó toggle (botó que permet seleccionar una opció o una altra).

En ambdós casos, es controla la informació introduïda amb un `TextEditingController` per cada espai on s'introdueix text, permetent així manipular-la. Cada acció crida el mètode de l'API corresponent: `login` o `signup`.

En el cas del `login`, un cop es prem el botó, s'executa el codi de l'API. En canvi, per registrar-se, abans d'executar el mètode de l'API, es comprova que el camp de la contrasenya coincideixi amb el de confirmació. Si el text és el mateix, s'executa l'API, i sinó, es mostra una advertència, igual que si hi ha un error (usuari ja existent o no existent en iniciar sessió). L'aplicació detecta aquests errors perquè l'API retorna un codi d'error si no s'efectua el flux esperat.

Cal destacar que la contrasenya de l'usuari s'envia encriptada per evitar que sigui manipulada per intrusos que puguin interceptar la comunicació entre les dues capes. Per obtenir el hash de la contrasenya, es fan servir les següents línies de codi:

```
final bytes = utf8.encode(password);  
final hash = sha256.convert(bytes).toString();
```

Listing 6: Mètode d'encriptació de la contrasenya

---

### 7.3.4 Classe ConditionalForm

Aquesta classe s'encarrega de processar la majoria de la lògica de l'aplicació, sent la pàgina principal on l'usuari pot escollir el tipus de dades (text, pdf o imatge) que proporcionar i extreure'n els macronutrients.

En funció del tipus, es mostren diferents widgets, encara que un cop processats, tots mostren els macronutrients de la mateixa manera. Tot i que cada un executa la mateixa funció, es diferencien en el tipus i les dades passades, i l'API s'encarrega de la lògica interna:

- **Text:**
  - Widgets: camps de text per insertar el nom i el llistat d'ingredients, botó per indicar que es vol processar l'introduït i botó per indicar si es vol compartir la recepta.
  - Funció: en el cos de la funció upload, es passen com a paràmetres el llistat d'ingredients, el nom de la recepta i l'identificador de l'usuari.
- **Pdf:**
  - Widgets: camp de text per introduir el nom de la recepta, un botó que permet buscar entre els fitxers del dispositiu el pdf de la recepta que volem processar i un botó per indicar si es vol compartir la recepta.
  - Funció: per passar fitxers PDF, es fa servir request.fields per afegir paràmetres individual i jeràrquicament, com el UUID, el nom i el fitxer.
  - Detectar pdf: en seleccionar el PDF, es busca el path amb la següent línia:

```
pdfPath = result.files.single.path
```

Listing 7: Línia per extraure el path

---

- **Imatge:**

- Widgets: camp de text per introduir el nom, dos botons per seleccionar una imatge del dispositiu o fer-ne una a l’instant i un botó per indicar si es vol compartir la recepta.
- Funció: els paràmetres es passen de la mateixa manera que en el cas del PDF, i l’API comprova si el fitxer és un PDF o una imatge.
- Detectar imatge: hi ha dues funcions per detectar la imatge; una per seleccionar-la del dispositiu i l’altra per fer-ne una amb la càmera. Les funcions són les següents:

```
Future<void> _pickImage(String uuid) async {
    final ImagePicker picker = ImagePicker();
    final XFile? image = await
        picker.pickImage(source:
            ImageSource.gallery);
    if (image != null) {
        setState(() {
            selectedImage = File(image.path);
        });
        await _uploadImage(selectedImage!, uuid);
    }
}
```

```
Future<void> _takePhoto(String uuid) async {
    final ImagePicker picker = ImagePicker();
    final XFile? photo = await
        picker.pickImage(source:
            ImageSource.camera);
    if (photo != null) {
        setState(() {
            selectedImage = File(photo.path);
        });
        await _uploadImage(selectedImage!, uuid);
    }
}
```

Listing 8: Funcions per obtenir imatges

---

### 7.3.5 Classe `HistorialPage`

En aquesta classe es mostra l'historial de receptes de l'usuari. Cada vegada que s'hi accedeix, es crida la funció que permet actualitzar la llista de receptes. També hi ha una funció que retorna informació ja coneguda (com el nom de la recepta), però que s'utilitza per obtenir el llistat de macronutrients i mostrar informació dels hidrats de carboni abans de seleccionar-ne alguna i obtenir més informació.

Per facilitar la cerca a la llista, hi ha un widget que filtra les receptes que comencen pels caràcters introduïts en el camp de cerca:

```
void _onSearchChanged() {
  setState(() {
    filteredHistorial = historial!
      .where((item) =>
        _nombre(item['nombre'], item['macros'])
          .toLowerCase()
          .startsWith(searchController.text.toLowerCase()))
      .toList();
  });
}
```

Listing 9: Algoritme de cerca en l'historial

A més, es permet eliminar receptes des d'aquesta pàgina o veure'n més informació. Per accedir a la pàgina d'informació, es fa servir la següent navegació:

```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => InfoPage(receta:
    historial![index])),
);
```

Listing 10: Navegació historial a l'informació de la recepta

També es pot ordenar el llistat de receptes en funció del filtre que es vulgui aplicar. Encara que actualment hi ha cinc opcions, en el futur es podrien afegir fàcilment nous filtres, com ara ordenar per valors de macronutrients

---

o pel temps de cocció, entre d'altres. A continuació, es presenten els filtres disponibles:

- Votació: ordenada de manera ascendent
- Votació: ordenada de manera descendent
- Ingredients: número exacte
- Ingredients: ordenada de manera ascendent
- Ingredients: ordenada de manera descendent

### 7.3.6 Classe ListFoodPage

Aquesta pàgina és similar a la de l'historial, ja que mostra el llistat d'ingredients que hi ha a la base de dades. No és necessari actualitzar la llista cada vegada que s'hi accedeix perquè, de moment, no hi ha cap funció que permeti modificar o afegir nous ingredients. La informació dels ingredients s'obté d'un PDF, que s'explica en el següent apartat.

També es pot obtenir informació detallada sobre un ingredient específic. En seleccionar-lo, es desplegarà la informació rellevant sobre els seus macronutrients. A més, en aquest moment es pot ajustar el valor dels grams que es visualitzen per adaptar-los a les nostres necessitats.

### 7.3.7 Classe InfoPage

Aquesta pàgina, a la qual s'accedeix des de l'historial, mostra les dades de la recepta: el nom, el llistat d'ingredients i el llistat de macronutrients. També inclou la funcionalitat de calcular els macronutrients en funció del nombre de persones o dels grams per ració individual.

En prémer el botó corresponent, apareix un desplegable que permet escollir com es vol fer el càlcul, i retorna el resultat ajustat. Si es torna a prémer el botó, es mostren de nou els macronutrients de la recepta completa.

---

Cal destacar que l'API retorna la informació en format JSON, cosa que permet manipular la informació de manera més àgil que amb una variable en format text.

També es pot votar i veure qui ha publicat la recepta si s'accedeix des de la classe ReceiptPublicPage. Si l'accés es fa des de HistorialPage, es comprova si la recepta ha estat publicada. En cas afirmatiu, es mostra la mitjana de les votacions. Si la recepta no està publicada, la mitjana de les votacions no es mostra. A més, es permet alternar entre publicar o no la recepta.

### 7.3.8 Classe ReceiptPublicPage

En quant a la implementació, és similar a la classe HistorialPage. No obstant això, la diferència principal és que no es poden eliminar les receptes del llistat, ja que aquesta tasca és exclusiva de l'usuari que ha publicat la recepta. A més, en accedir a la informació de les receptes, podem veure qui ha publicat cada recepta i donar-li *feedback* mitjançant la votació.

### 7.3.9 Classe DeleteUserPage

Inicialment, volia mostrar un missatge a l'usuari sobre la pàgina actual per confirmar si volia eliminar totes les receptes o només aquelles que no ha compartit, sense que l'usuari abandonés la pàgina. No obstant això, posteriorment vaig considerar més oportú implementar una pàgina intermèdia.

El flux de la classe intermèdia és el següent: primer es comprova si l'usuari té receptes publicades. Si és així, es notifica l'usuari sobre les opcions d'eliminació disponibles. A continuació, es crida al mètode de l'API corresponent per eliminar les receptes i, posteriorment, per eliminar l'usuari. Si l'usuari no té receptes, la pàgina intermèdia no es mostra i l'eliminació es realitza directament. Un cop completada l'eliminació, l'usuari és redirigit

---

al formulari d'inici de sessió o registre.

### 7.3.10 Classe `HelpPage`

La implementació d'aquesta classe ha estat senzilla, ja que té com a objectiu proporcionar informació addicional a l'usuari. Per tant, la classe es compon principalment d'elements visuals dissenyats per aportar aquesta informació de manera clara i efectiva.

## 7.4 Altres implementacions

### 7.4.1 Base de dades ingredients

Per obtenir la informació dels ingredients que hi ha a la base de dades, vaig utilitzar un PDF utilitzat per estudiants de nutrició, concretament el titulat *Tablas de composición de alimentos del CESNID*. Abans d'executar el programa que incorporava la informació, vaig haver de passar les dades a un fitxer Excel, que permet importar taules des d'un PDF.

Finalment, vaig unificar les taules que es detectaven per construir-ne una d'individual. També vaig eliminar les columnes que no eren necessàries, ja que només volia conservar les mínimes requerides per mostrar els macronutrients.

Per incorporar el fitxer Excel com a dades de la taula de la base de dades, he creat un script que permet la incorporació de dades. Primer s'ha hagut de connectar a la base de dades PostgreSQL:

```
db_params = {  
    'dbname': 'tfg',  
    'user': 'postgres',  
    'password': 'admin',  
    'host': 'localhost',  
    'port': '5432'
```

---

}

Listing 11: Connexió base de dades

Posteriorment s'han de seguir els següents passos, per aconseguir la lògica d'incorporar els ingredients:

1. Crear l'URL de la base de dades.
2. Crear el motor de connexió.
3. Llegir el fitxer Excel.
4. Crear la taula proporcionant el nom, les columnes i el tipus d'aquestes.
5. Introduir les dades a la taula.

#### 7.4.2 Connexió ChatGpt

Inicialment, l'algoritme per obtenir el llistat d'ingredients es basava en peticions a ChatGPT, passant-li el text del PDF i una pregunta específica per obtenir el llistat desitjat. Malauradament, però, no vaig poder continuar amb aquesta implementació perquè es va limitar la prova gratuïta que proporciona OpenAI sobre les seves APIs.

Aquesta solució hauria fet el prototip més funcional, ja que la implementació d'intel·ligència artificial permetia obtenir el llistat de manera més òptima i segura, assegurant-se de capturar tots els ingredients. També hauria permès manipular PDFs amb diverses receptes i extreure'n només els ingredients de la recepta necessària.

Tot i això, com amb el prototip actual, hi ha casos en què els PDFs no es detecten correctament a causa de la manera en què el text està estructurat, cosa que pot causar errors en l'extracció de text.

---

## 8 Problemes d'implementació

### 8.1 Intel · ligència artificial

Durant el desenvolupament del projecte, vaig decidir incorporar eines basades en intel · ligència artificial per a la manipulació de documents PDF i imatges, amb l'objectiu d'extreure'n el llistat d'ingredients de manera eficient. Inicialment, vaig explorar l'eina *ChatPDF*, que permet processar un arxiu PDF i respondre preguntes sobre el seu contingut. Tanmateix, no vaig poder connectar aquesta eina amb la meva solució perquè no proveeix una API pública.

Una altra opció que vaig considerar va ser la integració amb ChatGPT. En aquest cas, vaig aconseguir obtenir una clau per a la seva API i així integrar-ho amb èxit. El problema estava en què, la clau tenia un límit temporal gratuït, el qual un cop finalitzat, es requeria un pagament per continuar utilitzant l'eina. Això hauria provocat de no complir amb l'objectiu de construir una aplicació gratuïta per als usuaris.

### 8.2 Amazon web services

En les primeres fases del prototip, el servidor i la base de dades es van gestionar localment. No obstant això, per permetre l'accés des de dispositius externs, vaig decidir migrar aquests components al núvol. Tot i tenir les instàncies d'EC2 (servidor web) i RDS (base de dades) de AWS configurades, vaig trobar dificultats a l'hora de replicar o clonar les taules de la base de dades local mitjançant l'eina pgAdmin. L'error que es presentava era: *"Unable to connect to server: connection timeout expired"*. Tot i provar amb diverses solucions, cap va solucionar el problema completament.

---

### 8.3 Dades per a la base de dades d'aliments

Inicialment, vaig localitzar una pàgina web que contenia la base de dades d'ingredients d'Espanya (BEDCA), on es poden consultar les composicions dels aliments. Tanmateix, aquesta pàgina no oferia cap API que permetés extreure la informació de manera automatitzada, fet que em va portar a buscar altres alternatives.

Una altra opció que vaig trobar va ser un document PDF utilitzat per estudiants de nutrició amb informació d'un gran conjunt d'ingredients. Per evitar l'extracció manual de les dades, vaig intentar d'automatitzar el procés utilitzant intel·ligència artificial, concretament ChatGPT, per llegir el PDF i guardar només la informació necessària. Malauradament, l'eina només era capaç de processar uns 20 ingredients d'un total d'aproximadament 700, per una limitació de paràmetres del model.

Finalment, la solució que vaig implementar va ser automatitzar la tasca utilitzant una funcionalitat d'Excel que permet incorporar dades directament des d'un PDF. Aquesta funcionalitat processa cada pàgina com una taula separada, però vaig poder agrupar totes les taules i, posteriorment, eliminar les columnes innecessàries per conservar només les dades rellevants.

---

## 9 Avaluació

En aquest apartat tractaré d'avaluar si el projecte compleix amb els requisits establerts, duent a terme certes proves en l'aplicació i observant els seus resultats, així com valorant els costos del projecte.

### 9.1 Avaluació del projecte

He dissenyat les següents proves per verificar que el projecte compleix amb els requisits funcionals i que els resultats són els esperats. A la següent taula detallo les proves:

Table 1: Descripció de les proves i resultats del sistema

<b>Descripció</b>	<b>Resultat</b>	<b>Correcte</b>
Iniciar sessió amb usuari existent	El sistema mostra la pantalla d'inici de l'aplicació	Sí
Iniciar sessió amb nom d'usuari erroni	Missatge d'error on notifica que contrasenya o nom d'usari incorrecte.	Sí
Iniciar sessió amb un camp en blanc.	Missatge d'error on notifica que contrasenya o nom d'usuari incorrecte.	Sí
Regista un nou usuari.	El sistema mostra la pantalla d'inici de l'aplicació.	Sí
Regista un usuari existent.	Missatge d'error on notifica que ja existeix el nom d'usuari.	Sí
Regista un nou usuari amb comprovació de contrasenya diferent de la contrasenya.	Missatge d'error on notifica que la contrasenya de verificació no és la mateixa.	Sí

<b>Descripció</b>	<b>Resultat</b>	<b>Correcte</b>
Regista un nou usuari amb format de contrasenya erronú.	Missatge d'error on notifica que la contrasenya ha de seguir un format.	Sí
Tancar la sessió.	El sistema mostra la pantalla d'iniciar sessió.	Sí
Pàgina principal opció de text: introduir nom recepta i llistat d'ingredients.	El sistema mostra els macronutrients.	Sí
Pàgina principal opció de text: introduir nom recepta.	Notifica que no hi ha llistat ingredients	Sí
Pàgina principal opció de text: introduir llistat d'ingredients.	Notifica que no es pot guardar si no li posa un nom	Sí
Pàgina principal opció de pdf: introduir nom recepta i pdf.	El sistema mostra els macronutrients.	Sí
Pàgina principal opció de pdf: introduir només pdf.	Notifica que no es pot guardar si no li posa un nom	Sí
Pàgina principal opció d'imatge: introduir nom recepta i buscar/fer una foto.	El sistema mostra els macronutrients.	Sí
Pàgina principal opció d'imatge: buscar o fer una foto.	Notifica que no es pot guardar si no li posa un nom	Sí
Accedir a la pàgina de l'historial	El sistema mostra el llistat de les receptes introduïdes per l'usuari.	Sí
Pàgina historial: fer una cerca per caràcters	El sistema només mostra el llistat de les receptes introduïdes per l'usuari que comencin pels caràcters indicats.	Sí

<b>Descripció</b>	<b>Resultat</b>	<b>Correcte</b>
Pàgina historial: seleccionar una recepta.	El sistema mostra la pàgina d'informació d'aquesta.	Sí
Pàgina historial: eliminar una recepta.	El sistema elimina la recepta de la llista.	Sí
Pàgina historial: aplicar un filtre.	El sistema mostra la llista amb l'ordre segons el filtre.	Sí
Accedir a la pàgina d'informació d'una recepta.	El sistema mostra la informació d'aquesta: el nom de la recepta, el llistat d'ingredients i els macronutrients.	Sí
Pàgina informació recepta: calcular macronutrients individuals a l'indicar les persones.	El sistema mostra els macronutrients resultants per la ració individual.	Sí
Pàgina informació recepta: calcular macronutrients individuals a l'indicar els grams.	El sistema mostra els macronutrients resultants per la ració individual.	Sí
Pàgina informació recepta: accés des de receptes públiques i tocar les estrelles.	El sistema mostra el valor de la votació i el guarda a la recepta pertinent.	Sí
Accedir a la pàgina d'ingredients.	El sistema mostra el llistat d'ingredients que hi ha a la base de dades	Sí
Pàgina ingredients: fer una cerca per caràcters	El sistema només mostra el llistat d'ingredients que comencin pels caràcters indicats.	Sí
Pàgina ingredients: seleccionar un ingredient	El sistema mostra els macronutrients per 100gr d'aquest.	Sí

<b>Descripció</b>	<b>Resultat</b>	<b>Correcte</b>
Pàgina ingredients: canviar grams d'un ingredient	El sistema mostra els macronutrients pels grams introduïts.	Sí
Accedir a la pàgina de receptes publicades.	El sistema mostra el llistat de les receptes publicades.	Sí
Pàgina de receptes publicades: aplicar un filtre.	El sistema mostra el llistat de les receptes publicades de manera ordenada segons el filtre.	Sí
Eliminar usuari: amb receptes públicades	El sistema mostra un missatge per saber si vol eliminar totes les receptes o només les no publicades. Finalment, mostra la pàgina d'iniciar sessió	Sí
Eliminar usuari: amb receptes no públicades	El sistema elimina l'usuari i mostra la pàgina d'iniciar sessió.	Sí

## 9.2 Avaluació dels costos

El projecte actual s'ha plantejat com un prototip funcional, de manera que no ha suposat cap cost monetari associat amb el desenvolupament, ja que l'objectiu principal ha estat assolir una versió bàsica i operativa del sistema.

Tot el desenvolupament s'ha realitzat amb recursos propis, com el meu ordinador personal, en el qual he utilitzat eines com Visual Studio per a la programació, pgAdmin per a la gestió de la base de dades PostgreSQL, i GitHub per emmagatzemar el codi i gestionar les versions.

---

Inicialment, vaig considerar la possibilitat d'integrar l'API de ChatGPT per millorar el processament d'extracció d'ingredients, especialment en la introducció de PDFs o imatges, així com per permetre l'addició de nous ingredients a la base de dades. Aquesta funcionalitat hauria suposat una inversió econòmica addicional, però com que l'objectiu era desenvolupar un prototip, he decidit no incloure la intel·ligència artificial en aquesta fase. No obstant això, no es descarta la seva integració i desenvolupament a l'aplicació en un futur.

A més, no he implementat cap solució per a l'accés remot al servidor, com seria el cas de desplegar el projecte en una plataforma com AWS. Si el projecte s'hagués desplegat en un servei d'allotjament, s'haurien d'haver tingut en compte els costos associats, com el manteniment del servidor, l'emmagatzematge de dades i l'ample de banda necessari.

---

## 10 Legislació i protecció de dades

En l'estat actual del prototip de la meva aplicació, l'única dada que necessita ser protegida és la contrasenya de l'usuari. Aquesta contrasenya es guarda i es transmet de manera segura entre el client i el servidor mitjançant tècniques de xifratge.

En aquesta primera versió, no es recull cap altra informació personal, ja que no és necessària per al funcionament bàsic de l'aplicació. No obstant això, en una futura ampliació que permeti la restauració de la contrasenya, es podria requerir informació addicional, com ara el correu electrònic, el qual també hauria de ser protegit d'acord amb les normatives vigents.

A més, hi ha la possibilitat que els usuaris pugui eliminar el seu compte si així ho desitgen. En aquest cas, totes les dades associades a l'usuari seran eliminades de la base de dades, garantint així un nivell adequat de protecció de la seva privacitat.

Actualment, no s'ha implementat una política de privacitat formal ni un procés d'acceptació per part de l'usuari. No obstant això, quan l'aplicació es llanci en la seva versió completa, els usuaris hauran d'acceptar la política de privacitat abans de registrar-se. Aquesta política explicarà detalladament com es recopilen, utilitzen i protegeixen les dades personals. A més, es desplegaran les mesures de seguretat necessàries per complir amb les normatives vigents de la Unió Europea i assegurar una protecció completa de les dades.

---

## 11 Conclusions

Aquest projecte ha estat un procés enriquidor en el qual he pogut veure com una idea inicial, malgrat no estar del tot definida, ha anat agafant forma gràcies a l'aplicació dels coneixements adquirits en diferents assignatures del grau.

L'objectiu principal, obtenir els macronutrients d'una recepta mitjançant tres formats de dades, ha estat assolit amb èxit, demostrant la funcionalitat del sistema desenvolupat i complint així amb el seu objectiu principal. Es podrien afegir noves funcionalitats, però seria important que aquestes no es desviessin del tema principal.

Destaco també que aquest projecte s'ha desenvolupat com una eina més personal per a la meva vida quotidiana, però he consultat persones properes que han indicat que també els sembla interessant el concepte de l'aplicació.

Mencionar que, tot i no haver treballat mai amb el framework Flutter, l'he trobat interessant per obtenir nous coneixements i aprendre més sobre el desenvolupament d'aplicacions amb versatilitat per a diferents tipus de dispositius (Android i iOS).

Tanmateix, he identificat certes limitacions que podrien superar-se amb millores futures, com la incorporació d'intel·ligència artificial, que permetria una major flexibilitat en la identificació d'ingredients i l'ampliació de la base de dades. Aquestes millores podrien ampliar l'abast de l'aplicació i fer-la encara més útil.

En definitiva, aquest treball m'ha permès consolidar els meus coneixements i m'ha brindat l'oportunitat de crear una aplicació pràctica i funcional, tot i les seves limitacions, amb un alt potencial de millora i aplicació futura.

---

## 11.1 Aportació de la carrera al projecte

Durant la carrera, he adquirit coneixements diversos sobre les diferents àrees de la informàtica. A continuació, destaco el conjunt d'assignatures que han estat més rellevants l'hora d'aportar coneixements per al desenvolupament d'aquesta tesis:

- **Anàlisi i disseny d'aplicacions:** aquesta assignatura ha estat fonamental per a la planificació i comprensió dels processos de desenvolupament d'aplicacions. M'ha permès sistematitzar les accions i estructures de l'aplicació abans de la seva programació, facilitant així un desenvolupament més organitzat i efectiu.
- **Aplicacions mòbils i encastades:** he adquirit coneixements bàsics sobre la programació d'aplicacions mòbils i la interacció amb aquestes, així com sobre el flux de programació basat en les interaccions dels usuaris i no només en una seqüència lineal d'instruccions.
- **Base de dades:** aquesta assignatura ha proporcionat una base sòlida en la manipulació de bases de dades, essencial per a la gestió i emmagatzematge eficient de dades en el projecte.
- **Sistemes d'informació en les organitzacions:** m'ha ofert coneixements avançats en la manipulació de bases de dades, amb un enfocament pràctic i actualitzat. Les pràctiques realitzades amb PostgreSQL han estat especialment valuoses per al desenvolupament del projecte.
- **Sistemes oberts i Sistemes de comerç electrònic:** en aquestes assignatures, he desenvolupat pàgines web on he treballat tant amb la capa del client com amb la del servidor, proporcionant-me experiència en la creació i gestió de serveis web.
- **Seguretat en xarxes:** aquesta assignatura m'ha ajudat en quant a les tècniques de protecció de dades sensibles mitjançant l'ús d'eines

---

d'encriptació, protocols segurs sobre la capa d'aplicació segons el model OSI, etc.

- **Sistemes distribuïts:** m'ha aportat coneixements sobre la coordinació de diferents elements que estan distribuïts en el Cloud (com és el cas del servidor i base de dades en la meva solució).
- **Projectes de sistemes informàtics:** ha estat útil per tal de planificar i gestionar el temps i recursos d'aquest projecte, assegurant així el correcte desenvolupament del treball.

## 11.2 Aportació personal del projecte

Aquest projecte no només serà una eina pràctica en la meva vida quotidiana, sinó que també m'ha proporcionat beneficis importants en diversos àmbits. En particular, l'aplicació em serà útil per controlar els hidrats de carboni presents en les receptes que preparo diàriament, ajudant-me a ajustar les unitats d'insulina necessàries.

A més, el projecte ha estat una oportunitat per posar en pràctica certs conceptes apresos durant la carrera, permetent-me aplicar coneixements sense la guia immediata de professors o manuals. He hagut de desenvolupar una eina des de zero, estudiant el mercat per assegurar-me que la idea fos innovadora i útil. Aquesta fase inicial va requerir una planificació acurada de l'estructura del projecte per garantir un desenvolupament ordenat i coherent.

A través del desenvolupament del projecte, he pogut aprofundir en la distribució i emmagatzematge de les dades necessàries per a l'aplicació, millorant les meves habilitats en la gestió de bases de dades. També he après a dissenyar l'aplicació tenint en compte l'experiència de l'usuari, assegurant que sigui visualment atractiva i funcional per a una àmplia gamma d'usuaris, independentment de la seva experiència o edat.

---

Encara que en altres assignatures havia observat la interacció entre els diferents elements d'una aplicació (client, servidor i dades), el desenvolupament d'aquest projecte m'ha permès entendre de manera més profunda com gestionar aquestes interaccions de manera efectiva.

Finalment, he adquirit nous coneixements en àrees com la incorporació d'intel·ligència artificial (tot i no acabar implementant-ho), la gestió de consultes mitjançant *API Keys*, i el processament de dades amb diversos mecanismes. Aquestes experiències han ampliat la meva comprensió sobre com optimitzar les funcionalitats de l'aplicació i millorar la seva eficiència.

### **11.3 Ampliacions futures**

En aquest treball de final de grau, s'ha desenvolupat un prototip de la meva idea inicial, el qual té el potencial d'incorporar noves funcionalitats per millorar l'experiència de l'usuari.

Una ampliació crucial seria la incorporació de la intel·ligència artificial. Encara que el prototip actual utilitza expressions regulars, l'ús de la intel·ligència artificial permetria introduir nous aliments que no es troben actualment a la base de dades. Això proporcionaria informació més precisa i milloraria la cerca de llistats d'ingredients, augmentant la precisió de la cerca i permetent filtrar per altres aspectes, com el temps necessari per elaborar una recepta.

També es podrien incorporar funcionalitats de recomanació de receptes basades en les preferències de l'usuari. Per exemple, si l'usuari és intolerant a la lactosa, es podrien suggerir receptes que no continguin lactosa. A més, si l'usuari desitja consumir una certa quantitat d'hidrats de carboni, l'aplicació podria oferir receptes que s'ajustin a aquests valors. Així mateix, seria important ampliar la base de dades de receptes, tant per als usuaris com per a les noves incorporacions, amb l'objectiu d'incrementar la varietat i disponibilitat de receptes. Com és lògic, aquesta nova funcionalitat aniria

---

acompanyada de l'ús de la intel·ligència artificial també, com s'ha explicat al paràgraf anterior.

Una ampliació interessant seria fer que el botó d'ajuda sigui més dinàmic, incorporant un chatbot que permeti fer preguntes sobre l'aplicació i obtenir la informació desitjada. Aquesta nova implementació estaria relacionada amb la intel·ligència artificial.

A més, per motivar els usuaris, es podrien afegir noves funcionalitats que premiïn les receptes més votades. Això incentivaria els usuaris a utilitzar més l'aplicació i a aprendre'n més sobre la informació dels aliments que consumeixen.

---

## 12 Bibliografía

API limits and quotas. Google for Developers.

<https://developers.google.com/photos/library/guides/api-limits-quotas>

Aplicaciones de nutrición: las mejores en 2024. (2023, February 15). El Blog de Lowi.

<https://www.lowi.es/blog/las-mejores-aplicaciones-de-nutricion/>

Base de datos BEDCA. Bedca.net. <https://www.bedca.net/bdpub/index.php>

ChatPDF Backend API - extract information and chat with PDFs. Chatpdf.com. <https://www.chatpdf.com/docs/api/backend?msclkid=6e756adc1a7c1dc22a0f048b5dcaad>

chinmaya121221 Follow Improve. (2023, October 12). Flutter - implement an image inside an

AlertDialog. GeeksforGeeks. <https://www.geeksforgeeks.org/flutter-implement-an-image-inside-an-alertdialog/>

<https://www.geeksforgeeks.org/flutter-implement-an-image-inside-an-alertdialog/>

Coding, A. [@amplifyabhicoding]. How to Upload Flutter PDF Files with a Single Click | PDF

| file upload | dio | amplifyabhi. Youtube. <https://www.youtube.com/watch?v=g834QMAqP0s>

Como obtener texto de una imagen? Stack Overflow en español.

<https://es.stackoverflow.com/questions/561623/como-obtener-texto-de-una-imagen>

Connecting to PostgreSQL Database Server. (2012, August 23). PostgreSQL Tutorial.

<https://www.postgresqltutorial.com/postgresql-getting-started/connect-to-postgresql-database/>

Flutter - Introduction. Tutorialspoint.com.

[https://www.tutorialspoint.com/flutter/flutter\\_introduction.htm](https://www.tutorialspoint.com/flutter/flutter_introduction.htm)

Flutter - toggle buttons. (2022, June 9). GeeksforGeeks.

<https://www.geeksforgeeks.org/flutter-toggle-buttons/>

Flutter/Dart Unhandled Exception: type “String” is not a subtype of type “Map” in type cast.

Stack Overflow. [https://stackoverflow.com/questions/69958335/](https://stackoverflow.com/questions/69958335/flutter-dart-unhandled-exception-type-string-is-not-a-subtype-of-type-mapst)

[flutter-dart-unhandled-exception-type-string-is-not-a-subtype-of-type-mapst](https://stackoverflow.com/questions/69958335/flutter-dart-unhandled-exception-type-string-is-not-a-subtype-of-type-mapst)

Flutter\_ pannable\_rating\_bar example. Dart Packages.

[https://pub.dev/packages/flutter\\_ pannable\\_rating\\_bar/example](https://pub.dev/packages/flutter_ pannable_rating_bar/example)

García, G. (2019, February 9). Acceder a un atributo de un StatefulWidget desde su State.

Guillermogarcia.es. <https://guillermogarcia.es/>

[acceder-a-un-atributo-de-un-statefulwidget-desde-su-state/](https://guillermogarcia.es/acceder-a-un-atributo-de-un-statefulwidget-desde-su-state/)

Gemini API quickstart. Google AI for Developers.

---

[https://ai.google.dev/tutorials/dart\\_quickstart](https://ai.google.dev/tutorials/dart_quickstart)

Icons class - material library - Dart API. Flutter.dev.

<https://api.flutter.dev/flutter/material/Icons-class.html>

Mansour, N. How to release your Flutter app for iOS and android. Instabug.com. <https://www.instabug.com/blog/how-to-release-your-flutter-app-for-ios-and-android>

old ssh client connection to AL2023 fails with “no hostkey alg.” Amazon Web Services, Inc. <https://repost.aws/questions/QU0Gh0VqHjQauzZ9Q8LQujjg/old-ssh-client-connection-to-al2023-fails-with-no-hostkey-alg>

Patel, J. (2024, July 12). Build iOS app with Flutter - Jignesh Patel. Medium.

<https://medium.com/@jigneshpatel24/build-ios-app-with-flutter-7dee52ef4b25>

PDFelement. (2023, February 27). ¿Cómo extraer texto de imágenes usando Python? Wondershare PDFelement.

<https://pdf.wondershare.es/ocr/extracting-text-from-image-python.html>

Python MySQL insert into. W3schools.com.

[https://www.w3schools.com/python/python\\_mysql\\_insert.asp](https://www.w3schools.com/python/python_mysql_insert.asp)

Quickstart — flask documentation (3.0.X). Palletsprojects.com.

<https://flask.palletsprojects.com/en/3.0.x/quickstart/>

Raza, F. Convertir captura de pantalla a texto con OCR en Python. Aspose.com.

<https://blog.aspose.com/es/ocr/convert-image-to-text-ocr-in-python/>

Run PostgreSQL queries from the command line. Stack Overflow. <https://stackoverflow.com/questions/19674456/run-postgresql-queries-from-the-command-line>

SearchBar class. Flutter.dev.

<https://api.flutter.dev/flutter/material/SearchBar-class.html>

Suryadevsingh. (2020, March 22). Deep dive into buttons & clickable widgets in flutter.

Suryadevsingh24032000. <https://medium.com/suryadevsingh24032000/buttons-clickable-widgets-in-flutter-2b2167fcfa23>

Tu primera app de Flutter. Google Codelabs. <https://codelabs.developers.google.com/codelabs/flutter-codelab-first?hl=es-419>

<https://codelabs.developers.google.com/codelabs/flutter-codelab-first?hl=es-419>

Amazon.com. [https://docs.aws.amazon.com/es\\_es/AmazonRDS/latest/UserGuide/TUT\\_WebAppWithRDS.html](https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.html)

[https://docs.aws.amazon.com/es\\_es/AmazonRDS/latest/UserGuide/TUT\\_WebAppWithRDS.html](https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.html)

Openai.com. <https://platform.openai.com/docs/overview>

---

Fitia.App. <https://fitia.app/es/es/calorias-informacion-nutricional/>

Deeplearning.Ai.

<https://learn.deeplearning.ai/courses/chatgpt-prompt-eng/lesson/1/introduction>

Openai.com. <https://platform.openai.com/docs/api-reference/introduction>

---

## 13 Annexos

### 13.1 Video de l'aplicació

En aquest apartat es troba l'enllaç al vídeo on es presenta una demostració del funcionament de l'aplicació.

<https://youtu.be/HDm1HNrxUKc>

### 13.2 Codis

En aquest apartat incloc alguna part de codi que és essencial o forma part de la lògica del projecte.

#### 13.2.1 obtenerCantidadIngrediente

```
def obtenerCantidadIngrediente(strLine, receta, recetaOrigin):
    partes = strLine.split()
    print("Line parts:", partes)
    dict = {}
    medida = ''
    cantidadOrigin = ''
    global gramos
    global numIngre

    if partes[0].isdigit():
        cantidadOrigin = int(partes[0])
        partes = partes[1:]
        if partes and partes[0] in unidades_medida:
            medida = partes[0]
            cantidad = cantidadOrigin * unidades_medida[medida]
            gramos = gramos + cantidad
            partes = partes[1:]
        else:
            ingrediente = ' '.join(partes)
            ingrediente = ingrediente.split('(')[0].strip()
            equivalencia =
                obtenerEquivalenciaIngrediente(ingrediente.lower())
```

---

```

        if equivalencia:
            cantidad = cantidadOrigin * equivalencia
            gramos = gramos + cantidad
        else:
            print(f"No equivalence found for {ingrediente}")
            return None
    else:
        return None

    ingrediente = ' '.join(partes)
    if ingrediente.startswith('de'):
        ingrediente = ingrediente[2:].strip()

    ingrediente = ingrediente.split('(')[0].strip()

    receta[ingrediente] = cantidad
    dict = {
        'ingrediente': ingrediente,
        'medida': medida,
        'cantidad': cantidadOrigin
    }
    recetaOrigin.append(dict)
    numIngre += 1
    return f"{cantidad} {ingrediente}".strip()

```

Listing 12: Funció per obtenir la quantitat i l'ingredient de la llista

### 13.2.2 obtenirMacros

```

def obtenirMacros(receta, macros):
    for ingrediente, cantidad in receta.items():
        ingrediente = ingrediente.capitalize()
        ingrediente_pattern = f"%{ingrediente}%"
        infos = obtenirIngredienteBD(ingrediente_pattern)
        afterInfos = infos

        if len(infos) == 0:
            partes = ingrediente.split()
            if partes[0].endswith('s'):
                ingrediente_pattern = f"%{partes[0][:-1]}%"
        else:

```

---

```

        ingrediente_pattern = f"%{partes[0]}%"
        infos = obtenerIngredienteBD(ingrediente_pattern)

    if len(infos) == 0:
        ingrediente_pattern = f"%{partes[0].lower()}%"
        infos = obtenerIngredienteBD(ingrediente_pattern)
    elif len(infos) > 2:
        if len(partes) == 1:
            partes.append('entero')
            ingrediente_pattern2 = f"%{partes[1]}%"
            cur.execute("SELECT * FROM alimentos WHERE id
                LIKE %s AND id LIKE %s", (ingrediente_pattern,
                    ingrediente_pattern2))
            infos = cur.fetchall()

elif len(infos) > 2:
    ingrediente_pattern = f"%{ingrediente},%"
    infos = obtenerIngredienteBD(ingrediente_pattern)

    if len(infos) == 0:
        infos = afterInfos

if infos:
    info = infos[0]

    id, kcal, proteina, grasa, HC, fibra = info
    macros['kcalTotal'] += proporcionarValores(cantidad,
        kcal)
    macros['HCTotal'] += proporcionarValores(cantidad,
        HC)
    macros['proteinaTotal'] +=
        proporcionarValores(cantidad, proteina)
    macros['fibraTotal'] +=
        proporcionarValores(cantidad, fibra)
    macros['grasaTotal'] +=
        proporcionarValores(cantidad, grasa)

```

Listing 13: Funció per obtenir els macronutrients

### 13.2.3 Codi d'inserir dades des d'un excel

---

```

db_url =
    f"postgresql+psycopg2://{db_params['user']}:{db_params['password']}
    @{db_params['host']}:{db_params['port']}/{db_params['dbname']}"

engine = create_engine(db_url)

df = pd.read_excel('../Taules_BD.xlsx')

df.replace('tr.', 0, inplace=True)

final_table_name = 'alimentos'

metadata = MetaData()

alimentos_table = Table(final_table_name, metadata,
    Column('id', String, primary_key=True),
    Column('kcal', Float),
    Column('proteina', Float),
    Column('grasa', Float),
    Column('HC', Float),
    Column('fibra', Float)
)

try:
    metadata.create_all(engine)
    print(f"Tabla '{final_table_name}' creada correctamente con
        la clave primaria")
except SQLAlchemyError as e:
    print(f"Error al crear la tabla '{final_table_name}': {e}")

try:
    df.to_sql(final_table_name, engine, if_exists='append',
        index=False)
    print(f"Datos insertados correctamente en la tabla
        '{final_table_name}'")
except SQLAlchemyError as e:
    print(f"Error al insertar datos en la tabla
        '{final_table_name}': {e}")

```

Listing 14: Insertar dades de l'excel d'ingredients