

**Grau en Enginyeria Informàtica de Gestió i Sistemes  
d'Informació**

**APLICACIÓ DE RECOLZAMENT A LES DIETES  
MÈDIQUES**

**Memòria**

**César Villarroya Ollero**  
**TUTORA: Montse Rabassa**

2021/2022



## **Dedicatòria**

Dedico aquest TFG a totes les persones que necessitin ajuda per seguir una determinada  
dieta mèdica.



## **Agraïments**

Agraïments a la Lola Robles per tota la informació proporcionada i la seva ajuda amb les dietes mèdiques.



## **Abstract**

This project aims to design and implement a tool to help follow a certain medical diet. The application must allow the user to choose a certain diet based on the dietician's guidelines and from there, they are shown the permitted and prohibited foods and a series of menu proposals to follow that must be varied depending on the day necessary for the diet. A mobile part must be created by the end user and a web part to be able to manage the contents by the dietitian. The mobile part must allow the products to be classified by certain diets by analyzing the barcodes of the products.

## **Resum**

Aquest projecte té com a objectiu dissenyar i implementar una eina d'ajuda a seguir una determinada dieta mèdica. L'aplicació ha de permetre a l'usuari escollir una determinada dieta en funció de les directrius del o la dietista i a partir d'aquí que se li mostrin els aliments permesos, els prohibits i una sèrie de propostes de menús a seguir que han de ser variats en funció dels dies que cal la dieta. S'ha de confeccionar una part mòbil per l'usuari final i una part web per tal de poder administrar els continguts per part del dietista. La part mòbil ha de permetre classificar els productes per dietes determinades analitzant els codis de barres dels productes.

## **Resumen**

Este proyecto tiene como objetivo diseñar e implementar una herramienta de ayuda a seguir una determinada dieta médica. La aplicación debe permitir al usuario escoger una determinada dieta en función de las directrices del o la dietista ya partir de ahí que se le muestren los alimentos permitidos, los prohibidos y una serie de propuestas de menús a seguir que deben ser variados en función de los días necesarios para la dieta. Se debe confeccionar una parte móvil por el usuario final y una parte web para poder administrar los contenidos por parte del dietista. La parte móvil debe permitir clasificar los productos por dietas determinadas analizando los códigos de barras de los productos.





# Índex

Índex de figures .....	III
Índex de taules .....	VII
Glossari de termes .....	IX
1. Objecte del projecte.....	1
2. Estudi previ: context, antecedents i necessitats d'informació.....	3
2.1. Context.....	3
2.2. Antecedents.....	4
2.3. Anàlisi de mercat i competència .....	5
3. Objectius i Abast .....	23
3.1. Objectius .....	23
3.2. Abast .....	25
4. Metodologia .....	27
5. Definició de requeriments funcionals i tecnològics .....	29
5.1. Requeriments de la plataforma web.....	29
5.2. Requeriments de la aplicació mòbil.....	30
6. Implementació del projecte .....	33
6.1. Aplicació web .....	33
6.1.1. Dissenys part web .....	36
6.1.2. Comentaris sobre la implementació .....	73
6.2. Aplicació mòbil.....	83
6.2.1. Dissenys de l'aplicació mòbil .....	83
6.2.2. Comentaris sobre la implementació .....	100
7. Conclusions .....	103
8. Ampliacions .....	105
9. Bibliografia.....	107



## Índex de figures

Fig. 2.3.1. Pantalla d'inici de Yazio. Font: [4].....	6
Fig. 2.3.2. Pantalla d'inici de Yuka. Font: [5].....	8
Fig. 2.3.3. Pantalla d'inici de MyRealFood. Font: [6].....	10
Fig. 2.3.4. Pantalla d'inici de MyFitnessPal. Font: [7].....	12
Fig. 2.3.5. Pantalla d'inici de HealthifyMe. Font: [8].....	14
Fig. 6.1.1 Distribució de carpetes del projecte. Font: Elaboració pròpia.....	33
Fig. 6.1.2 Fitxers dins la carpeta front-end. Font: Elaboració pròpia.....	34
Fig. 6.1.3 Fitxers dins la carpeta tfg. Font: Elaboració pròpia.....	34
Fig. 6.1.4 Imatges dins de la carpeta imatges. Font: Elaboració pròpia.....	35
Fig. 6.1.1.1 Diagrama BPMN sobre iniciar sessió. Font: Elaboració pròpia.....	37
Fig. 6.1.1.2 Diagrama BPMN sobre la visualització de les dietes. Font: Elaboració pròpia.....	38
Fig. 6.1.1.3 Diagrama BPMN sobre les característiques d'un aliment 1. Font: Elaboració pròpia.....	40
Fig. 6.1.1.4 Diagrama BPMN sobre les característiques d'un aliment 2. Font: Elaboració pròpia.....	41
Fig. 6.1.1.5 Diagrama BPMN sobre esborrar una dieta 1. Font: Elaboració pròpia.....	43
Fig. 6.1.1.6 Diagrama BPMN sobre esborrar una dieta 2. Font: Elaboració pròpia.....	44
Fig. 6.1.1.7 Diagrama BPMN sobre editar una malaltia com a administrador 1. Font: Elaboració pròpia.....	46
Fig. 6.1.1.8 Diagrama BPMN sobre editar una malaltia com a administrador 2. Font: Elaboració pròpia.....	47
Fig. 6.1.1.9 Diagrama BPMN sobre afegir un aliment 1. Font: Elaboració pròpia.....	49
Fig. 6.1.1.10 Diagrama BPMN sobre afegir un aliment 1. Font: Elaboració pròpia.....	49
Fig. 6.1.1.11 Disseny relacional de la base de dades part de l'usuari. Font: Elaboració pròpia.....	50
Fig. 6.1.1.12 Disseny relacional de la base de dades part de la dieta. Font: Elaboració pròpia.....	51
Fig. 6.1.1.13 Disseny relacional de la base de dades part dels grups alimentaris. Font: Elaboració pròpia.....	52
Fig. 6.1.1.14 Disseny relacional de la base de dades part dels aliments. Font: Elaboració pròpia.....	53

Fig. 6.1.1.15 Taules de la base de dades. Font: Elaboració pròpia.....	55
Fig. 6.1.1.16 Diagrama de classes de les dietes. Font: Elaboració pròpia.....	56
Fig. 6.1.1.17 Dibuix del patró capes. Font: Elaboració pròpia.....	57
Fig. 6.1.1.18 Distribució de carpetes de les capes. Font: Elaboració pròpia.....	58
Fig. 6.1.1.19 Fitxers de la capa de domini. Font: Elaboració pròpia.....	59
Fig. 6.1.1.20 Fitxers de la capa de persistència. Font: Elaboració pròpia.....	59
Fig. 6.1.1.21 Fitxers de la capa d'aplicació. Font: Elaboració pròpia.....	60
Fig. 6.1.1.22 Fitxers de la capa de l'api. Font: Elaboració pròpia.....	61
Fig. 6.1.1.23 Disseny pantalla inicial. Font: Elaboració pròpia.....	66
Fig. 6.1.1.24 Disseny pantalla de dietes. Font: Elaboració pròpia.....	67
Fig. 6.1.1.25 Disseny pantalla de grups i subgrups alimentaris. Font: Elaboració Pròpia .....	67
Fig. 6.1.1.26 Disseny pantalla d'aliments. Font: Elaboració pròpia.....	68
Fig. 6.1.1.27 Disseny pantalla amb les característiques d'un aliment. Font: Elaboració pròpia.....	68
Fig. 6.1.1.28 Disseny pantalla de login. Font: Elaboració pròpia.....	69
Fig. 6.1.1.29 Disseny pantalla d'administració. Font: Elaboració pròpia.....	69
Fig. 6.1.1.30 Pantalla inicial. Font: Elaboració pròpia.....	70
Fig. 6.1.1.31 Pantalla de dietes. Font: Elaboració pròpia.....	70
Fig. 6.1.1.32 Pantalla d'aliments. Font: Elaboració pròpia.....	71
Fig. 6.1.1.33 Pantalla de les característiques d'un aliment. Font: Elaboració pròpia.....	71
Fig. 6.1.1.34 Pantalla d'administració. Font: Elaboració pròpia.....	72
Fig. 6.1.1.35 Pantalla de login. Font: Elaboració pròpia.....	72
Fig. 6.1.2.1 Part de contingut de la classe RestController. Font: Elaboració pròpia.....	73
Fig. 6.1.2.2 Part de contingut de la classe Controller 1. Font: Elaboració pròpia.....	74
Fig. 6.1.2.3 Part de contingut de la classe Controller 2. Font: Elaboració pròpia.....	75
Fig. 6.1.2.4 Contingut d'una interfície. Font: Elaboració pròpia.....	75
Fig. 6.1.2.5 Contingut d'un element del domini. Font: Elaboració pròpia.....	76
Fig. 6.1.2.6 Contingut d'exemple d'un DTO. Font: Elaboració pròpia.....	77
Fig. 6.1.2.7 Contingut d'exemple d'un DAO 1. Font: Elaboració pròpia.....	78
Fig. 6.1.2.8 Contingut d'exemple d'un DAO 2. Font: Elaboració pròpia.....	79
Fig. 6.1.2.9 Contingut del fitxer pom. Font: Elaboració pròpia.....	80
Fig. 6.1.2.10 Dependència per poder connectar el mysql amb el Spring. Font: Elaboració pròpia.....	80

Fig. 6.1.2.11 Dependència per poder connectar a la base de dades.	
Font: Elaboració pròpia.....	81
Fig. 6.1.2.12 Propietats per poder connectar amb mysql. Font: Elaboració pròpia.....	81
Fig. 6.1.2.13 Exemple d'us de fetch amb GET. Font: Elaboració pròpia.....	82
Fig. 6.1.2.14 Exemple d'us de fetch amb DELETE. Font: Elaboració pròpia.....	82
Fig. 6.2.1.1 Digrama BPMN sobre iniciar sessió. Font: Elaboració pròpia .....	84
Fig. 6.2.1.2 Digrama BPMN sobre canviar les dades de l'usuari. Font: Elaboració pròpia .....	86
Fig. 6.2.1.3 Digrama BPMN sobre consultar les característiques d'un aliment mitjançant el buscador 1. Font: Elaboració pròpia .....	88
Fig. 6.2.1.4 Digrama BPMN sobre consultar les característiques d'un aliment mitjançant el buscador 2. Font: Elaboració pròpia .....	88
Fig. 6.2.1.5 Digrama BPMN sobre consultar aliments d'una variant del menú 1. Font: Elaboració pròpia .....	90
Fig. 6.2.1.6 Digrama BPMN sobre consultar aliments d'una variant del menú 2. Font: Elaboració pròpia .....	91
Fig. 6.2.1.7 Digrama BPMN sobre generar un menú dinàmic. Font: Elaboració pròpia .....	93
Fig. 6.2.1.8 Disseny de la pantalla d'inici de sessió. Font: Elaboració pròpia .....	94
Fig. 6.2.1.9 Disseny de la pantalla inicial. Font: Elaboració pròpia .....	94
Fig. 6.2.1.10 Disseny del buscador. Font: Elaboració pròpia.....	95
Fig. 6.2.1.11 Disseny de la pantalla de les dades personals Font: Elaboració pròpia .....	95
Fig. 6.2.1.12 Disseny de la pantalla de creació de menús Font: Elaboració pròpia.....	96
Fig. 6.2.1.13 Disseny de la pantalla del menú generat per dies. Font: Elaboració pròpia .....	96
Fig. 6.2.1.14 Disseny de la pantalla de les variants de la dietista. Font: Elaboració pròpia.....	97
Fig. 6.2.1.15 Disseny de la pantalla dels plats de cada variant. Font: Elaboració pròpia .....	97
Fig. 6.2.1.16 Disseny de la pantalla de la llista de la compra. Font: Elaboració pròpia .....	98
Fig. 6.2.1.17 Disseny de la pantalla de la llista de la compra d'un menú. Font: Elaboració pròpia .....	98

Fig. 6.2.1.18 Disseny de la pantalla de les característiques d'un aliment. Font: Elaboració pròpia.....	99
Fig. 6.2.2.1 Exemple funció amb Filter. Font: Elaboració pròpia .....	100
Fig. 6.2.2.2 Exemple funció amb Collect. Font: Elaboració pròpia .....	100
Fig. 6.2.2.3 Exemple funció amb Search. Font: Elaboració pròpia .....	101
Fig. 6.2.2.4 Exemple funció per llegir el codi de barres d'un aliment. Font: Elaboració pròpia .....	101

## **Índex de taules**

Taula 2.3.1. Comparativa de funcionalitats de les aplicacions estudiades.....	18
---	----





## Glossari de termes

TFG	Treball Final de Grau
app	Aplicació mòbil
SCRUM	Metodologia Àgil
BMPN	Business Process Modeling Notation
IDE	Integrated Development Environment
API	Application Programming Interface
REST	Representational State Transfer
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
IA	Intel·ligència Artificial
DTO	Data Transfer Object
DAO	Data Access Object



## **1. Objecte del projecte**

L'objectiu de dur a terme el projecte és poder donar suport a les persones amb necessitats de control alimentàries o que tenen algun problema mèdic perquè entenguin que poden o no poden menjar segons la dieta mèdica que han de seguir.

El producte final està destinat per a dietistes de cara a poder facilitar la seva feina i la dels seus pacients. Avui en dia existeixen molts tipus de dietes i molta abundància d'aliments amb una gran quantitat de característiques, per tant, s'ha de fer un gran esforç per poder memoritzar o registrar tal diversitat de dades. La finalitat de fer el producte és agilitzar la cerca d'informació sobre dietes mèdiques, aliments i menús.

A l'aplicació resultant, el o la dietista dona accés als pacients per tal de poder usar l'aplicació mòbil sense cap dificultat. El pacient pot analitzar codis de barres per poder informar-se sobre les característiques dels aliments com greixos, calories, minerals, vitamines... També pot seleccionar les dietes mèdiques que vulgui tenir en compte d'entre les facilitades per la dietista. De cada una té l'opció de consultar els aliments permesos i prohibits i visualitzar propostes de menús a continuar depenent de la dieta i les preferències de l'usuari.

L'objectiu pel pacient final és que entengui perquè no pot menjar algunes coses i per què. El producte final d'aquest projecte no és en cap cas un substitutiu de la feina del dietista, la seva funció principal és de fer-li un suport a la seva feina i facilitar-li el procés de creació de dietes i menús.

Un fi del projecte és poder crear una aplicació que agrupi moltes funcionalitats diferents en un mateix producte. Actualment, l'àmbit de les dietes s'ha anat explotant a mesura que la necessitat de cuidar la salut ha anat augmentant. Avui en dia quasi tothom té accés a mòbil o a la web, per tant, qualsevol usuari ha de tenir accés a l'aplicació. El món de les dietes ha evolucionat molt, en conseqüència, es fa un producte d'acord amb les dietes mèdiques actuals.



## **2. Estudi previ: context, antecedents i necessitats d'informació**

S'ha fet un estudi previ a crear el producte analitzant els productes semblants al mercat i les seves funcionalitats, necessitats dels pacients i dietes mèdiques existents i més usades.

### **2.1 Context**

S'ha trobat davant una situació d'àmbit mèdic en què pacients de centres mèdics, hospitals o qualsevol persona hagi de seguir una determinada dieta mèdica. Els pacients han de saber escollir correctament quins aliments han de consumir i quins no, per poder seguir la seva dieta per no ocasionar problemes en el cos. Després d'una recerca d'informació des de diverses fonts i després d'una reunió amb la dietista, s'ha determinat els alguns tipus de dietes mèdiques que són més usades, això es necessitarà de cara al fet que els pacients puguin seguir la seva dieta:

- Hipocalòriques: Són dietes que contenen aliments amb baixa densitat energètica, alta densitat nutricional i d'alt volum. Són dietes indicades pels que pateixen obesitat, hipotiroïdisme, malalties cardiovasculars, diabetis, etc.
- Hipercalòriques: Les dietes hipercalòriques són aquelles que contenen una aportació energètica més gran del que necessita un individu. Són dietes indicades pels que pateixen cremats, traumatismes, malnutricions, hipertiroïdisme o càncer. En cas de tenir candidiasis també seria recomanable.
- Hipoglucèmica [1]: Aquesta dieta és una dieta d'aliments baixos en glúcids. Aquesta dieta és sobretot per tractar amb la diabetis de tipus 2.
- Hipolipídica [2]: Està indicada per a pacients amb colesterol i triglicèrids alts. Consisteix a reduir la ingesta de lípids amb menys del 25% de greixos a la dieta. És una dieta per problemes fetge, malalties cardiovasculars, ictus, úlcera digestiva, acidesa estomacal.
- Hipocolesterolemiant [3]: És una dieta especialment per baixar el colesterol de pacients amb colesterol alt. L'objectiu és ingerir menys de 200 mg al dia.

- Rica en greixos essencials omega 3 i 6: Indicat en dietes colesterol alt i pacients amb colesterol "bo" baix.
- Hiperlipídica: Règim alimentari caracteritzat per contenir més del 35% de l'energia en forma de greixos.
- Dietes controlades en proteïnes, micronutrients, líquids o residus
- Dietes progressives, sobretot en embarassades
- Dietes toves, líquides, semi líquides o semi sòlides
- Dietes de fàcil digestió

Les dietes anteriors són les dietes mèdiques principals, però n'hi ha de moltes més. Durant aquest projecte només es fan servir dues dietes, ja que es tracta de crear un primer prototip. Per consultar amb més detall mirar l'annex.

## **1.2 Antecedents**

Tant els centres mèdics com els pacients necessiten tenir un suport per poder seguir les dietes correctament, sobretot quan no es té coneixement ben definit sobre el tema. S'ha demanat fer un producte per poder ajudar en aquesta situació i així poder contribuir a realitzar les dietes correctament.

Es comença de zero completament, sense cap diagrama ni cap mena de sistema d'informació ja existent.

### **1.3 Anàlisi de mercat i competència**

Els programaris dietoterapèutics constitueixen actualment una eina bàsica en el tractament dietètic de pacients, per tant, ara en aquests temps estan molt de moda l'ús d'aquestes eines. Molta gent acostuma a utilitzar-les per fins esportius o per baixar o pujar de pes. S'ha buscat aplicacions similars de cara a poder estudiar el funcionament d'aquestes i comparar-la amb la idea proposada i poder comprovar el mercat per la proposta demanada. També per poder agafar noves idees i que el projecte sigui més eficient.

Els programaris analitzats han estat:

Yazio, Yuka, MyRealFood, MyfitnessPal, HealthifyMe, Diabetic Diet

Cada programari estudiat té els seus avantatges i els seus inconvenients que s'analitzen a continuació.

## Yazio [4]

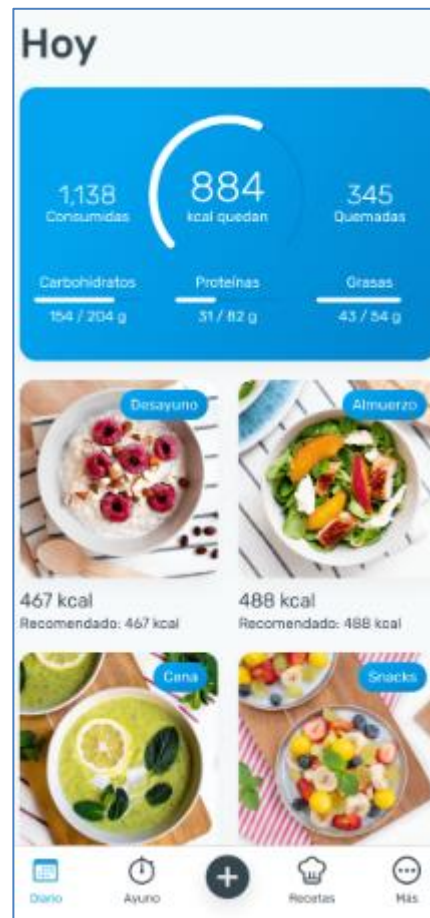


Fig. 2.3.1. Pantalla d'inici de Yazio. Font: [4]

Yazio és una app per poder controlar l'alimentació i que es pugui perdre de pes, o augmentar-lo, depenent de les pretensions de cada usuari.

És una aplicació molt coneguda des de fa un temps que hi va haver el gran boom de les aplicacions per controlar les dietes i l'alimentació. És una aplicació bastant completa, conté comptador de calories, consells de dejuni intermitent, es pot escollir diferents plans nutricionals, entre ells receptes, consells diaris i tasques que proposen per poder aconseguir l'objectiu amb més rapidesa. Amb aquests es pot controlar tot el que es menja durant el dia així com el que es beu.

Per aquesta aplicació és molt important saber la finalitat de l'usuari, pot ser guanyar o perdre pes. També té en compte el seguiment d'una dieta específica, sigui vegetariana, vegana, piscitariana o omnívora per tenir en compte les receptes o aliments a recomanar. Pregunta tant el pes com l'altura, edat i el sexe. Conté un cercador d'aliments perquè es pugui cercar amb comoditat quin aliment es vol menjar i saber les seves característiques, així com calories o valor nutricional. Conté una base de dades oberta al públic [10] on es poden consultar els



diferents aliments així com les seves característiques nutricionals.

Té en compte les passes que es fan de cara a restar les calories cremades durant un dia. És una aplicació amb moltes funcionalitats que fins fa poc es va fer de pagament, el qual ha fet que els usuaris no la descarreguin, limitant les descàrregues. Es pot apreciar la seva interfície a la figura 2.3.1.

## Yuka [5]

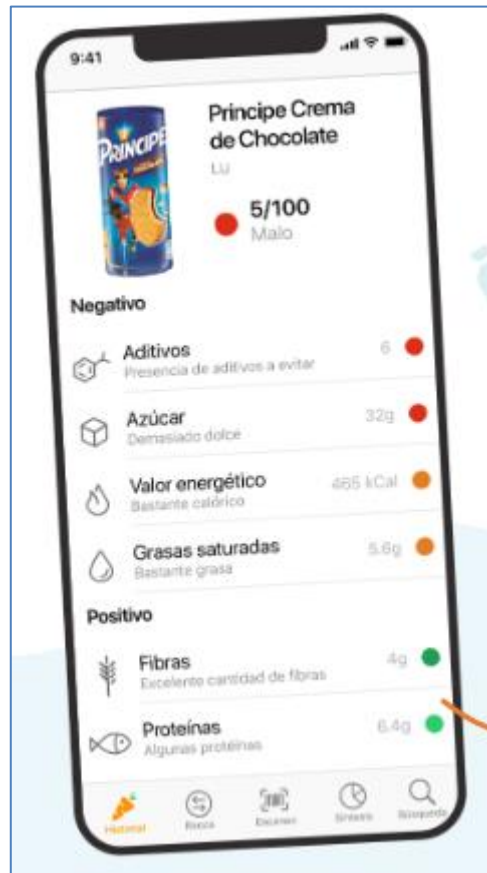


Fig. 2.3.2. Pantalla d'inici de Yuka. Font: [5]

Yuka és una app exclusivament per escanejar codis de barres de productes, aliments i cosmètics.

Permet escanejar els codis de barres de tota mena de productes, ja que té un nivell de coneixement de marques bastant profund i ofereix informació detallada del seu impacte a la salut, tant en l'àmbit negatiu com positiu.

Una eina molt útil quan es va al supermercat i es vol saber informació del que es compra, quan és bo per a la salut.

Una cosa molt positiva és que posa una nota, un tant per cent, de quant saludable és aquest producte. És molt específic, fins i tot indica els productes determinats que contenen additius, dels quals t'informa de quins són bons, regulars o dolents segons les seves regles. També indica els que contenen molta sal, el valor energètic, les fibres, quins contenen sucre, proteïnes o grasses bones o dolentes.

Després conté un sistema de recomanació en què d'un determinat producte et dona propostes d'aliments més saludables. Té registrats els productes que has analitzat per poder consultar-los altre cop. No té cap mena de registre, es guarda tot a la base de dades del telèfon. És una aplicació molt simple, però molt eficient en el que es vol aconseguir. Es pot apreciar la seva interfície a la figura 2.3.2.

## MyRealFood [6]

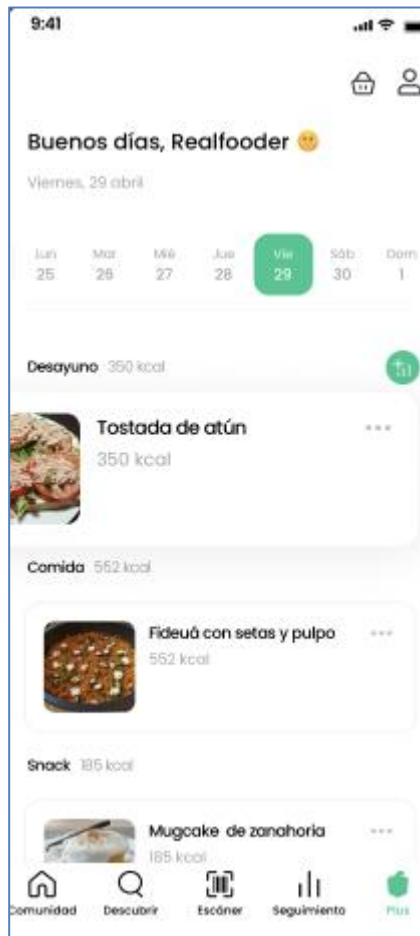


Fig. 2.3.3. Pantalla d'inici de MyRealFood. Font: [6]

L'app, a més de disposar de l'opció d'escanejar productes i determinar els menjars "reals" o processats. Compta amb una extensa base de dades de receptes saludables que s'han d'incloure a tota dieta sana.

Conté un apartat de registre i d'inici de sessió a diferència de Yuka. Inclou l'opció de registrar-se amb Google o amb un correu. Conté l'opció d'analitzar codis de barres dels aliments com l'anterior.

Detecta els aliments baixos en sal, amb fibra, proteïna, baix en grasses, baix en sucre... Addicionalment, conté alguns ingredients del producte i un apartat per saber en quin supermercat pots comprar-lo.

Igualment, té funcionalitats per poder registrar els menjars que es fa cada dia per esmorzar, dinar o sopar i permet registrar les mesures per comprovar els objectius complets com el pes, mesura del pit, cintura i maluc.

Recomana a l'usuari què menjar, hi ha un conjunt de tipus d'aliments com per exemple oli, pasta i arròs, begudes vegetals... i de cada tipus recomana aliments alternatius als considerats com a processats, ultraprocessats...

I per últim, que les altres aplicacions no tenen, conté una comunitat que és semblant a una xarxa social en què els usuaris fan posts de diferents menjars o receptes parlant de les seves experiències personals. Es pot dir que aquesta és una barreja de les dues anteriors. Es pot apreciar la seva interfície a la figura 2.3.3.

## MyfitnessPal [7]

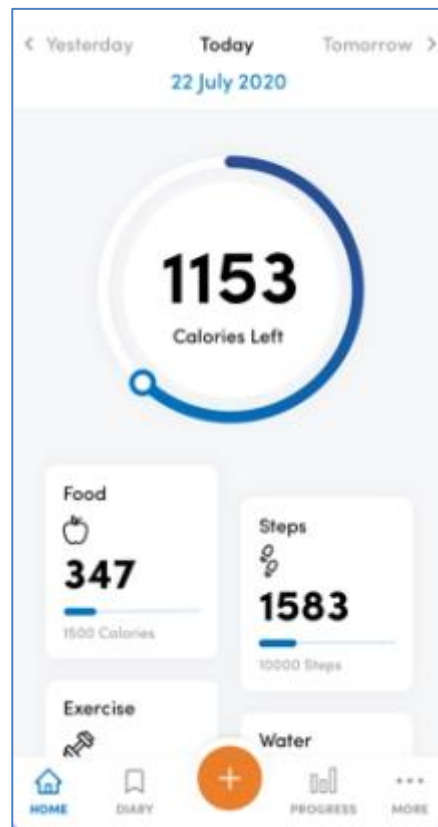


Fig. 2.3.4. Pantalla d'inici de MyFitnessPal. Font: [7]

És una app més que conté un comptador de calories per perdre pes o assolir el desitjat amb el menor esforç. Disposa d'una base de dades d'aliments amb més de 3.000.000 d'aliments amb diferents menjars i exercicis. Facilita les relacions socials, ja que permet afegir amics i compartir els objectius i progressos.

Conté un registre d'esmorzar, dinar i sopar on es pot posar tot el que s'ha menjat i comptar correctament les calories ingerides durant el dia. Es pot veure com es progressa amb el comptador de calories i uns gràfics diaris.

Té una opció per poder apuntar els propòsits, ja siguin objectius de calories, carbohidrats, proteïnes, grasses, pes, etc. Segons el que s'ha menjat indica el total dels carbohidrats, proteïnes, grasses ingerides, vitamines, sucres, minerals... Això fa que es tingui un resum molt complet.

Conté l'opció de crear receptes, menjars i aliments quan no hi són a la base de dades. Inclou l'opció de contar les passes que es fan de cara a poder calcular millor les calories cremades.

També conté una comunitat per poder visualitzar consells d'altres usuaris sobre alimentació, nutrició, forma física, receptes, etc.

Es poden agregar amics i poder parlar amb ells mitjançant missatgeria, això en cap de les anteriors s'ha vist. Per últim, conté una funció per poder gestionar recordatoris per poder decidir les hores del menjar o similars. És molt completa, però no té en compte les dietes. Es pot apreciar la seva interfície a la figura 2.3.4.

## HealthifyMe [8]

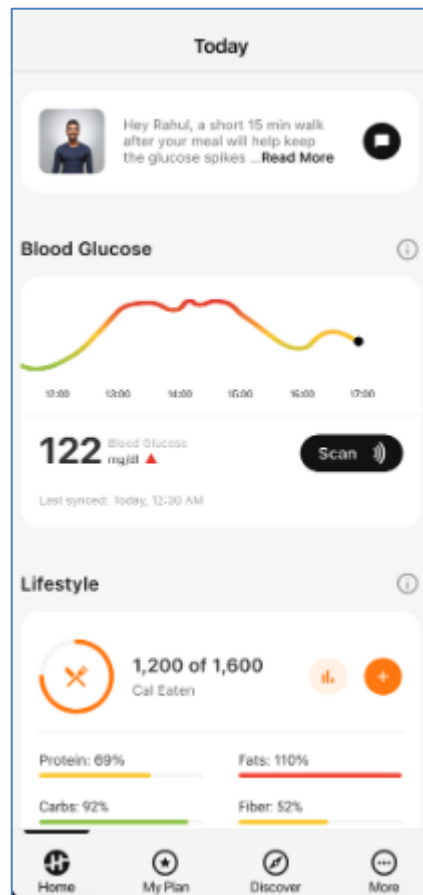


Fig. 2.3.5. Pantalla d'inici de HealthifyMe. Font: [8]

HealthifyMe és una plataforma digital índia de salut i benestar que ofereix serveis com ara el seguiment de calories, entrenament individual de nutrició i fitness, i plans de dieta i entrenament. L'aplicació adopta un enfocament holístic de seguiment de l'estil de vida per mantenir els usuaris compromesos i motivats.

Té configuracions d'exercici a casa, comptador de calories, guanyar múscul, pèrdua de pes, guanyar pes, pla de dietes i guies d'entrenadors personals. Per tal de poder guardar els progressos té un sistema d'inici de sessió per poder identificar-se. Dona l'opció de fer-ho per mòbil, correu electrònic, Facebook o Google. Durant el registre s'ha d'indicar nom, ciutat d'origen, gènere, edat, l'activitat física, altura, pes, els objectius de pes, dona l'opció d'indicar alguna condició mèdica com per exemple diabetis, colesterol, hipertensió, tiroides, alguna deficiència física...

Les funcions bàsiques són calcular les calories totals que s'ha ingerit durant el dia, sigui esmorzar, dinar o sopar. Es té en compte les calories que es cremen durant l'exercici i les



hores en les quals s'adorm. Hi ha articles dins l'aplicació de persones que hi han pujat imatges o consells sobre les seves experiències.

Els usuaris amb subscripcions prèmium poden triar entre un equip de nutricionistes certificats interns, entrenadors de fitness i entrenadors de ioga. Alternativament, els usuaris poden accedir a la nutricionista mitjançant IA per obtenir ajuda. La part negativa d'aquesta aplicació és que quasi tot el que és interessant és de pagament, tota la resta és molt bàsic. Es pot apreciar la seva interfície a la figura 2.3.5.

## **Diabetic Diet**

Aquesta és una aplicació molt més específica que les anteriors. Buscant aquest exemple s'ha volgut veure quines funcionalitats conté una aplicació d'una dieta mèdica determinada, en aquest cas la dieta per persones diabètiques.

Ens dona moltes opcions, però totes elles són informatives. Té informació sobre que és la diabetis, tractament, guia per als diabètics, exercicis, consells per perdre pes, menjars que poden menjar els diabètics, proposa un menú setmanal perquè puguin saber què menjaramb alguns plats i com preparar-los. També dona informació sobre que és una dieta diabètica i alguns dubtes comuns resolts.

Després té en compte preferències sobre els menjars, com salats o dolços, es mostra receptes a fer amb els ingredients i la preparació a més del valor nutricional del plat determinat.

Aquesta aplicació és una guia molt bàsica i estàtica, no té en compte els gustos de l'usuari ni les seves preferències. Ha servit per veure quin tipus d'informació es busca per una dieta determinada.

## **Anàlisi i conclusió**

Després de fer una anàlisi exhaustiu, s'han detectat una sèrie de requisits i funcionalitats comunes, en què s'analitzen les funcionalitats de cada aplicació i s'ha arribat a una sèrie de conclusions.

De cada un s'analitzen quins dels següents requisits es compleixen:

- Analitzar codis de barres
- Incloure dietes registrades
- Afegir receptes
- Contenir opcions de pagament
- Definir objectius de l'usuari
- Tenir en compte l'alimentació de l'usuari
- Preguntar dades físiques a l'usuari
- Contenir un registre d'aliments
- Donar informació dels aliments
- Identificar usuaris
- Registrar alimentació diària
- Recomanar alternatives als aliments
- Contenir una comunitat
- Registrar amics i permetre enviar missatges
- Comptar passes
- Avisar als usuaris de l'hora de menjar
- Incloure exercicis pels usuaris
- Controlar condicions mèdiques
- Informar sobre dietes mèdiques
- Proposar menús

A continuació a la taula 2.3.1 es pot veure els paràmetres a comprovar de cada aplicació estudiada.

	Yazio	Yuka	MyRealFood	MyFitnessPal	HealthifyMe	Diabetic Diet
Analitzar codis de barres	No	Sí	Sí	No	No	No
Incloure dietes registrades	No	No	No	No	No	Sí
Afegir receptes	Sí	No	No	Sí	No	Sí
Contenir opcions de pagament	Sí	No	No	No	Sí	No
Definir objectius de l'usuari	Sí	No	No	Sí	Sí	No
Tenir en compte l'alimentació de l'usuari	No	No	No	No	No	Sí
Preguntar dades físiques a l'usuari	Sí	No	No	No	Sí	No

	Yazio	Yuka	MyRealFood	MyFitnessPal	HealthifyMe	Diabetic Diet
Contenir un registre d'aliments	Sí	Sí	Sí	Sí	No	No
Donar informació dels aliments	Sí	Sí	Sí	Sí	No	Sí
Identificar usuaris	No	No	Sí	No	Sí	No
Registrar alimentació diària	Sí	No	Sí	Sí	Sí	No
Recomanar alternatives als aliments	No	Sí	Sí	No	No	Sí
Contenir una comunitat	No	No	Sí	Sí	Sí	No
Registrar amics i permetre enviar missatges	No	No	No	Sí	No	No

	Yazio	Yuka	MyRealFood	MyFitnessPal	HealthifyMe	Diabetic Diet
Comptar calories	Sí	No	Si	Sí	Sí	No
Avisar als usuaris de l'hora de menjar	No	No	No	Sí	No	No
Incloure exercicis pels usuaris	No	No	No	Sí	Sí	Sí
Controlar condicions mèdiques	No	No	No	No	Sí	No
Informar sobre dietes mèdiques	No	No	No	No	No	Sí
Proposar menús	No	No	No	No	No	Sí

Taula 2.3.1. Comparativa de funcionalitats de les aplicacions estudiades. Font: Elaboració pròpia.

Després d'analitzar totes aquestes aplicacions es destaca que són semblants entre elles, però amb petites diferències. S'han agafat idees sobre cadascun dels exemples estudiats i també possibles funcionalitats pel projecte com per exemple la consideració de l'alimentació de l'usuari, informar a l'usuari sobre les dietes mèdiques, proposar menús personalitzats, recomanar alternatives d'aliments, analitzar codis de barres o tenir un registre d'aliments.

No hi ha cap aplicació existent en el mercat actual que sigui específicament centrat en dietes mèdiques. La gran majoria van centrades en el menjar saludable i en determinats objectius de guanyar pes o perdre pes. Per aquest tipus de situacions no es necessiten les dietes mèdiques, seguint un hàbit de menjar saludable és suficient i altres tipus de dietes en què no és de l'àmbit del projecte.

Hi ha exemples de dietes mèdiques, així i tot, només en dietes determinades com en el cas de l'aplicació pels diabètics, això fa entendre que no hi ha una aplicació que tingui un conjunt de dietes mèdiques per a diferents pacients, només dietes en concret. Per tant, és una gran oportunitat per crear el producte desitjat i poder tenir un lloc al mercat sense competència directa.





### **3. Objectius i Abast**

L'objectiu del projecte és crear un producte que permeti als pacients saber que poden o no poden menjar segons una determinada dieta mèdica. També un producte pel o la dietista en què permeti administrar tota la informació que vulgui manegar així com tipus de dietes, aliments...

#### **3.1 Objectius**

##### **Objectius del client**

El client en aquest projecte és un determinat dietista o nutricionista en què el seu objectiu és poder proporcionar als seus pacients un servei de dietes. El servei simplifica la tasca de recerca de dades per les dues parts, ja sigui per poder buscar informació sobre dietes o els aliments permesos o no permesos.

En concret, els objectius específics són:

- Proporcionar informació sobre les necessitats dels pacients quant a necessitats alimentàries.
- Aconseguir un producte que funcioni correctament, sense falles i funcional per tal de donar suport al dietista.
- Oferir un bon servei proporcionant funcionalitats adients de l'aplicació.
- Realitzar el procés de generar dietes personalitzades de cada pacient i que es faci correctament segons les necessitats de cada pacient.
- Disposar de diferents aliments i aliments alternatius per a una dieta determinada
- Portar un registre dels pacients del dietista o la dietista.

## **Objectius del producte**

L'objectiu del producte és poder garantir que tots els requeriments per part del client es compleixin i que inclogui totes les funcionalitats demanades pel client.

- Registrar tots els pacients del dietista
- Mostrar dietes mèdiques existents sota el control de la dietista.
- Determinar els aliments permesos i no permesos d'una dieta determinada.
- Generar menús recomanables per seguir una dieta determinada
- Proposar un calendari amb diferents menús per cada dia de la setmana.
- Administrar els diferents aliments o productes i els continguts.
- Analitzar codis de barres de productes.
- Classificar els productes segons les dietes existents.
- Classificar correctament els aliments tenint en compte les necessitats de l'usuari, així com dieta, alimentació, preferències, gustos...

## **Objectius de l'usuari final**

L'usuari final és en aquest cas els pacients dels dietistes, vol saber segons la dieta que li han assignat, què poden menjar i què no, conèixer les propietats dels aliments que vulgui analitzar, quins menús es recomanen a la setmana per poder seguir la dieta sense problemes i restringir quins aliments no es pot menjar i poder buscar altres alternatives.

- Saber quins aliments es pot menjar o no segons la dieta escollida.
- Escollir la dieta desitjada dins de l'aplicació, que hi sigui present.
- Tenir diferents alternatives d'aliments per la dieta.
- Visualitzar, crear i modificar els menús que es vulgui per la dieta.
- Restringir els aliments que no agraden o no es poden menjar, ja que no entraran dins la seva dieta

- Escanejar els codis de barres de tots els menjars.
- Satisfer les necessitats com a pacient.
- Tenir coneixement de tots els menjars i les seves propietats.

## 3.2 Abast

L'abast del projecte és crear un producte que tingui dues tecnologies, tecnologia web i tecnologia mòbil. La web és bàsicament per definir dietes, poder introduir nous productes i administrar el contingut d'interès pels usuaris.

Aquesta ha de diferenciar entre administradors o dietista i pacients. L'administrador és l'únic que pot modificar la base de dades, el pacient només pot accedir a manera de consultes per visualitzar menús, aliments, etc. El contingut mòbil permet escanejar codis de barres d'aliments i de cada un determinar les seves característiques nutricionals i classificar-los per una dieta determinada.

Aquest és un prototip que mostra les funcionalitats desitjades pel dietista. Ha de permetre a l'usuari final escollir una dieta determinada i a partir de la dieta que hagi seleccionat se li mostrin els aliments que pot i que no pot menjar així com una sèrie de propostes de menús a seguir que han de ser variats en funció dels dies a la setmana que cal seguir la dieta.

El principal objectiu és començar com a mínim amb dues dietes mèdiques i a mesura que avanci el projecte afegir-ne més.

No es pretén treballar de forma exhaustiva totes les possibles dietes mèdiques



## 4. Metodologia

Aquest projecte és desenvolupat amb metodologies Àgils. El projecte es porta a terme mitjançant SCRUM com a procés de software, en el qual es fan esprints setmanals per poder determinar les tasques que es fan durant aquest projecte. Al final de cada esprint es determina de les tasques plantejades quines s'han realitzat correctament i a partir dels resultats es plantegen els nous esprints i les noves tasques a realitzar. Les tasques a fer es decideixen a partir d'històries d'usuari amb diferents prioritats per dur a terme durant els esprints.

La cerca d'informació d'aquest projecte es fonamenta en trobar informació sobre les dietes mèdiques existents, per a qui són aquestes dietes i el tipus de pacients, informació sobre cada dieta com per exemple qui les usa, perquè les usa, que és el que fa que no puguin menjar amb normalitat, quin problema té cada pacient per usar una dieta determinada, quin és l'element que fa que no puguin menjar normal, quins aliments estan permesos i quins no, recomanacions d'aliments dins de cada dieta, etc.

Per consultar dades es busquen bases de dades d'aliments amb les seves característiques nutricionals així com calories, greix, proteïnes, vitamines, etc. Sobretot és important a l'hora de cercar que la informació sigui contrastada per la dietista. També es fa ús de la informació proporcionada per una nutricionista professional que proporciona informació vàlida per poder acomplir el projecte.

A continuació un cop s'ha buscat informació sobre el producte la següent tasca és la definició del projecte, en el qual es defineixen els objectius, tant com del client, de l'usuari final, pel producte o pel projecte. Es fa la definició de requisits, en el qual se celebra una reunió amb el client i es determinen els requisits funcionals, els no funcionals i els requeriments del negoci. El següent pas és realitzar un estudi sobre els costos del projecte, siguin salaris, llicències o programaris. Un cop els requeriments ja estan definits i revisats ja es poden fer els diagrames BPMN dels processos en els quals participen el producte i els models preliminars necessaris per entendre els processos que gestiona el producte, així com diagrames de procés, diagrames de casos d'ús, diagrames de classes i diagrames de seqüència.

Un cop s'ha fet la definició del projecte i ja s'ha determinat que el projecte és viable. A continuació cal fer el pla de projecte. En el pla de projecte cal definir els objectius del projecte, tant per l'usuari final, el client, o el producte. S'ha de definir els criteris d'acceptació, en què es determina segons quines condicions el projecte és acceptat o rebutjat, sobretot se centra en la viabilitat i els recursos. Cal definir les fases que es duen a terme durant el projecte, anàlisi, desenvolupament, implantació, manteniment ... I les tasques que es desenvolupen durant cada fase del projecte. És important definir l'equip que participa durant el projecte i quina tasca fa cadascun en cada esprint.

A partir de les tasques a fer, es crea un calendari del projecte amb els sprints planejats amb les tasques a fer per cadascú. També és necessari crear un pla de control de qualitat per poder analitzar amb exactitud el que s'ha creat.

Després del pla de projecte comença la fase d'anàlisi i disseny, primer per la plataforma i després els dissenys necessaris per al correcte funcionament del sistema.

Durant el disseny i preparació de la plataforma es tenen en compte els elements de hardware i software, així com l'entorn de desenvolupament, control de versions, eines de gestió del projecte, pla d'instal·lació de la plataforma...

L'anàlisi i disseny del producte és la part més delicada del projecte, ha d'estar tot ben dissenyat. Durant aquesta etapa es dissenyen tots els diagrames de classes, disseny d'interfícies d'usuari, disseny de bases de dades, disseny de crides a l'API, disseny de testos per comprovar el funcionament de les funcionalitats...

Un cop finalitzat el procés de disseny, es procedeix a la seva codificació. Durant aquesta etapa es codifiquen tots els mòduls definits en l'etapa anterior. A continuació es procedeix a realitzar un testing de tot el codi per comprovar que tot funciona. Un cop l'aplicació ha passat els tests de comprovació, ja es pot fer l'entrega i s'espera l'acceptació de la primera versió dels mòduls.

Un cop el projecte està tancat, és crucial dur a terme la documentació final del projecte.

## **5. Definició de requeriments funcionals i tecnològics**

### **5.1 Requeriments de la plataforma web**

#### **Requeriments funcionals**

- Registrar usuaris
- Identificar usuaris
- Mostrar el sistema de registre d'aliments
- Guardar els aliments a la base de dades
- Mostrar aliments registrats per tipus amb les seves característiques
- Mostrar informació sobre les dietes mèdiques
- Registrar, esborrar o modificar qualsevol aliment
- Modificar informació de l'usuari
- Donar de baixa un usuari
- Mostrar les dades de l'usuari
- Donar d'alta, modificar o eliminar una dieta
- Validar les propostes d'aliments per part dels usuaris

#### **Requeriments no funcionals**

- Optimitzar el rendiment per permetre una gran quantitat d'usuaris
- Optimitzar l'eficiència
- Implementar un sistema segur
- Poder accedir a la plataforma des de qualsevol navegador sense problema
- Complir amb la llei de protecció de dades
- Mostrar els resultats de la web en menys de 5 segons
- Configurar la interfície per ser intuïtiva per l'usuari final
- Disposar d'una interfície web

- Comptar amb una disponibilitat del 99% de les vegades que un usuari hi vulgui accedir

## **Requeriments del negoci**

- Només poden donar d'alta o de baixa usuaris els usuaris administradors
- Només poden registrar, esborrar o modificar aliments els usuaris administradors
- Només poden donar d'alta, modificar o eliminar una dieta els usuaris administradors
- Només poden validar noves propostes d'aliments els usuaris administradors
- Només poden veure les opcions de registre els usuaris administradors

## **5.2 Requeriments de la aplicació mòbil**

### **Requeriments funcionals**

- Detectar codis de barres
- Llegir codis de barres
- Mostrar dietes mèdiques vinculades a l'expedient de cada pacient
- Identificar usuaris
- Modificar informació de l'usuari
- Mostrar contingut depèn de l'usuari
- Mostrar camps de registre i d'inici de sessió
- Mostrar camps per omplir com el nom, l'edat, pes, altura, alimentació o preferències
- Mostrar aliments permesos segons la dieta determinada
- Mostrar aliments no permesos segons la dieta determinada
- Mostrar els aliments existents
- Registrar el producte que s'ha escanejat si no hi és a la base de dades
- Classificar el producte escanejat segons la dieta a què pertanyen



- Comprovar que el producte escanejat és a la base de dades
- Crear menús diaris tenint en compte les preferències i la dieta de l'usuari
- Generar una llista de la compra segons el menú generat dinàmicament

### **Requeriments no funcionals**

- Optimitzar el rendiment al màxim per permetre una gran quantitat d'usuaris
- Optimitzar l'eficiència
- Implementar un sistema segur
- Poder accedir a la plataforma des de qualsevol navegador sense problema
- Complir amb la llei de protecció de dades
- Mostrar els resultats de la web en menys de 5 segons
- Configurar la interfície per ser intuïtiva per l'usuari final
- Disposar d'una interfície web
- Comptar amb una disponibilitat del 99% de les vegades que un usuari hi vulgui accedir



## 6. Implementació del projecte

### 6.1 Aplicació web

S'ha separat la implementació del projecte en dues grans parts: el backend, que és la part interna del programari en el qual s'ha creat la base de dades, API REST, i s'han creat els diagrames corresponents així com diagrama de classes i diagrama de la base de dades. A continuació hi ha el frontend, el qual és la part visible per l'usuari. S'ha fet dissenys de les diferents pàgines web així com la funcionalitat d'entre elles. S'ha creat la web perquè sigui usable i agradable, tenint així un disseny senzill i intuïtiu de cara a l'usuari final.

S'ha dissenyat perquè la web sigui compatible en tots els navegadors.

S'ha fet la distribució de carpetes on es pot veure a la figura 6.1.1: *tfg*, que és on s'ubica tot el backend, *imatges*, que és on es guarden les imatges necessàries, i *front-end*, que és on s'hi guarden les diferents pantalles necessàries per a la visualització web. Dins del frontend hi ha fitxers HTML, CSS i Javascript.

front-end	03/04/2022 12:36	Carpeta de archivos
imatges	06/04/2022 11:19	Carpeta de archivos
tfg	27/02/2022 22:07	Carpeta de archivos

Fig. 6.1.1 Distribució de carpetes del projecte. Font: Elaboració pròpia.

## **Front-End:**

A la figura 6.1.2 es pot apreciar el contingut de la carpeta *front-end*. Dins podem observar els diferents fitxers HTML, on els quals es fa la maquetació web, CSS, que dona els estils a la pàgina o JS, que dona les funcionalitats de cada pàgina.

administracio.css	03/04/2022 12:03	Documento de ho...	3 KB
administracio.html	18/04/2022 19:51	Chrome HTML Do...	121 KB
aliment.css	24/03/2022 20:19	Documento de ho...	2 KB
aliment.html	17/04/2022 17:04	Chrome HTML Do...	11 KB
aliments.css	22/03/2022 19:52	Documento de ho...	3 KB
aliments.html	17/04/2022 17:01	Chrome HTML Do...	5 KB
dietes.css	20/03/2022 20:27	Documento de ho...	1 KB
dietes.html	17/04/2022 17:05	Chrome HTML Do...	4 KB
formulari.css	31/03/2022 19:36	Documento de ho...	12 KB
grupalimentari.html	17/04/2022 17:04	Chrome HTML Do...	5 KB
index.css	17/03/2022 19:52	Documento de ho...	1 KB
index.html	06/04/2022 11:25	Chrome HTML Do...	6 KB
index.js	15/03/2022 20:16	Archivo JS	0 KB
login.css	03/04/2022 13:03	Documento de ho...	1 KB
login.html	03/04/2022 14:26	Chrome HTML Do...	6 KB
login.js	16/03/2022 19:25	Archivo JS	0 KB
subgrupalimentari.html	17/04/2022 17:01	Chrome HTML Do...	5 KB

Fig. 6.1.2 Fitxers dins la carpeta front-end. Font: Elaboració pròpia.

## **TFG:**

A la figura 6.1.3 es pot observar els diferents fitxers usats en el backend dins la carpeta *tfg*. Es pot observar el fitxer pom, on hi són les dependències del projecte, un connector mysql per poder connectar la base de dades amb Spring...

.idea	20/04/2022 19:51	Carpeta de archivos	
.settings	16/02/2022 18:54	Carpeta de archivos	
src	16/02/2022 18:50	Carpeta de archivos	
target	27/02/2022 21:56	Carpeta de archivos	
.classpath	16/02/2022 18:54	Archivo CLASSPATH	2 KB
.project	16/02/2022 18:54	Archivo PROJECT	1 KB
HELP.md	16/02/2022 18:50	Archivo MD	1 KB
mvnw	16/02/2022 18:50	Archivo	11 KB
mvnw.cmd	16/02/2022 18:50	Script de comand...	7 KB
mysql-connector-java-8.0.27.jar	27/02/2022 21:06	Executable Jar File	2.418 KB
pom.xml	27/02/2022 22:06	Archivo XML	3 KB
tfg.iml	27/02/2022 22:07	Archivo IML	11 KB

Fig. 6.1.3 Fitxers dins la carpeta tfg. Font: Elaboració pròpia.

**Imatges:**

A la figura 6.1.4 es pot observar les diferents imatges usades en el front-end, així com per afegir, esborrar o editar, i algunes imatges de dietes i colors. Totes aquestes imatges són fetes servir a la web, totes són de lliure disposició.



Fig. 6.1.4 Imatges dins de la carpeta imatges. Font: Elaboració pròpia.

### 6.1.1 Dissenys part web

S'ha fet un estudi de les diferents funcionalitats que es necessiten per fer aquesta part web de l'aplicació i per poder acomplir la tasca mencionada s'han fet casos d'ús de les possibles tasques que fa l'usuari que usa l'aplicació web.

S'han fet casos d'ús sobre la consulta de dietes, aliments i administració de la plataforma. Els casos d'ús de la plataforma són molt semblants, per tant, es mostren alguns exemples genèrics així com afegir, editar o esborrar informació. També s'ha fet un diagrama BPMN per cada cas d'ús perquè quedi clar com funciona el sistema. Tots els següents diagrames s'ha fet mitjançant l'eina BPMN.io [14].

#### Casos d'ús i diagrames BPMN

##### Cas N°1:

- **Nom:** Iniciar sessió
- **Descripció:** L'usuari vol iniciar sessió
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina web
- **Postcondicions:** L'usuari ha iniciat sessió

##### Flux Normal:

1. L'usuari accedeix a la pàgina web
2. L'usuari polsa sobre el botó de login del menú
3. El sistema li mostra a l'usuari la pantalla de login amb un formulari per posar el nom d'usuari i la contrasenya
4. L'usuari introdueix les dades
5. L'usuari polsa el botó de login
6. El sistema comprova si és un usuari registrat
7. El sistema mostra la pàgina d'inici amb el botó d'administració visible

**Flux Alternatiu:**

1. L'usuari no omple tots els camps del formulari i fa que no es mostri el botó de login
2. L'usuari no està registrat, per tant, no va a la pàgina d'inici

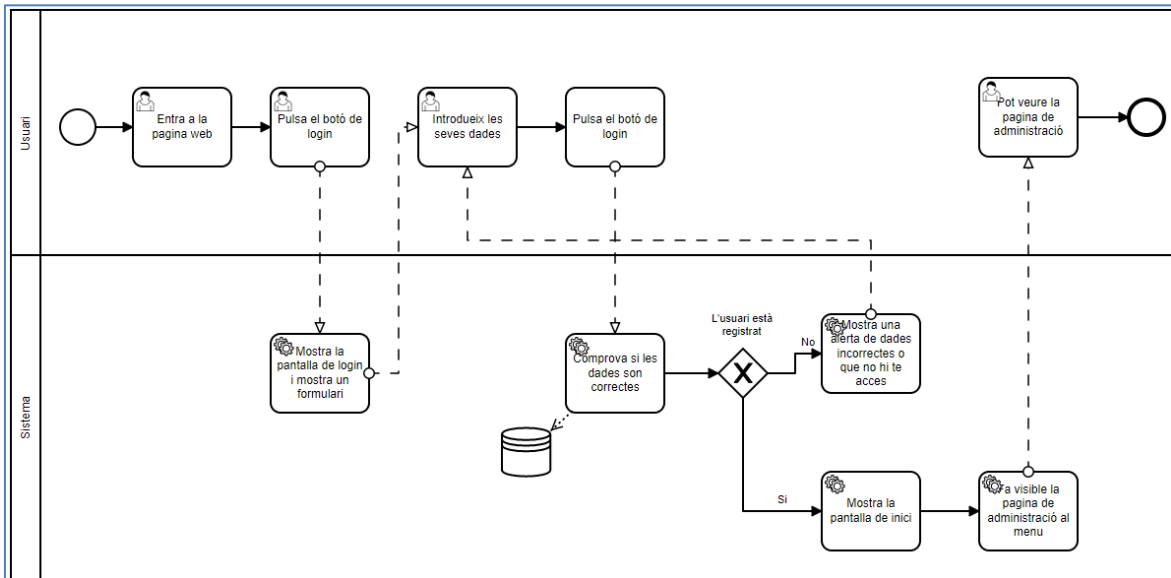
**Diagrama BPMN:**

Fig. 6.1.1.1 Diagrama BPMN sobre iniciar sessió. Font: Elaboració pròpia.

**Cas N°2:**

- **Nom:** Visualitzar les dietes disponibles
- **Descripció:** L'usuari vol veure les dietes disponibles a la base de dades
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina web
- **Postcondicions:** L'usuari ha consultat les dietes

**Flux Normal:**

1. L'usuari accedeix a la pàgina web
2. L'usuari polsa sobre el botó de dietes del menú
3. El sistema busca les dietes disponible

4. El sistema mostra la pantalla de dietes amb les dietes disponibles

**Flux Alternatiu:**

1. El sistema no troba dietes a la base de dades

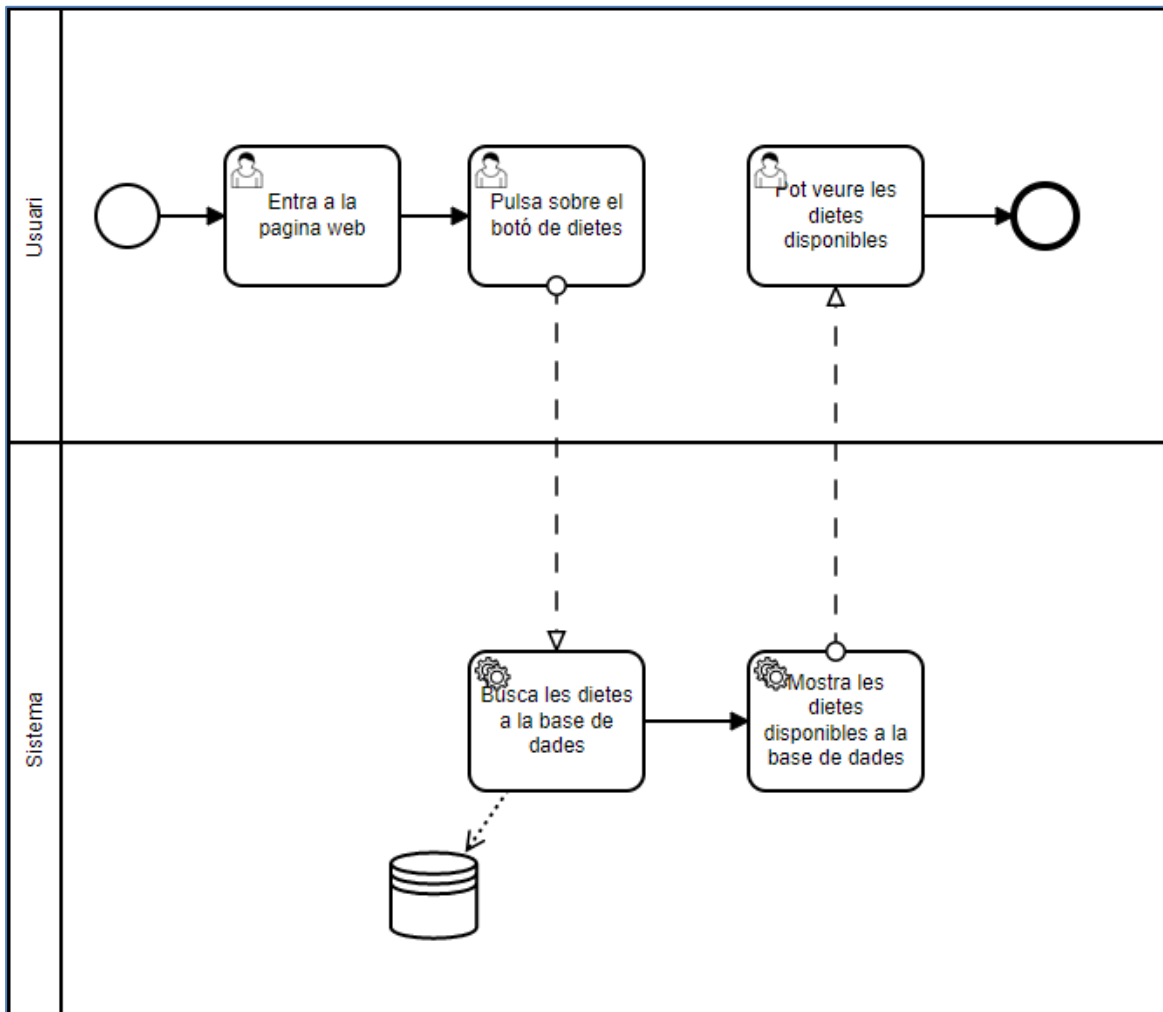
**Diagrama BPMN:**

Fig. 6.1.1.2 Diagrama BPMN sobre la visualització de les dietes. Font: Elaboració pròpia.



**Cas N°3:**

- **Nom:** Veure les característiques d'un aliment
- **Descripció:** L'usuari vol veure les característiques d'un aliment de la base de dades
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina web
- **Postcondicions:** L'usuari ha consultat un aliment

**Flux Normal:**

1. L'usuari accedeix a la pàgina web
2. L'usuari polsa sobre el botó d'aliments
3. El sistema busca els grups alimentaris de la base de dades
4. El sistema mostra a l'usuari els grups alimentaris disponibles
5. L'usuari polsa un grup alimentari
6. El sistema busca els subgrups alimentaris del grup seleccionat
7. El sistema mostra a l'usuari els subgrups alimentaris del grup seleccionat
8. L'usuari polsa un subgrup alimentari
9. El sistema busca els aliments del subgrup seleccionat
10. El sistema mostra a l'usuari els aliments disponibles del subgrup
11. L'usuari polsa sobre l'aliment desitjat
12. El sistema busca l'aliment seleccionat
13. El sistema mostra a l'usuari l'aliment seleccionat
14. L'usuari pot veure les característiques de l'aliment seleccionat

**Flux Alternatiu:**

1. El sistema no troba grups a la base de dades
2. El sistema no troba subgrups a la base de dades
3. El sistema no troba aliments a la base de dades

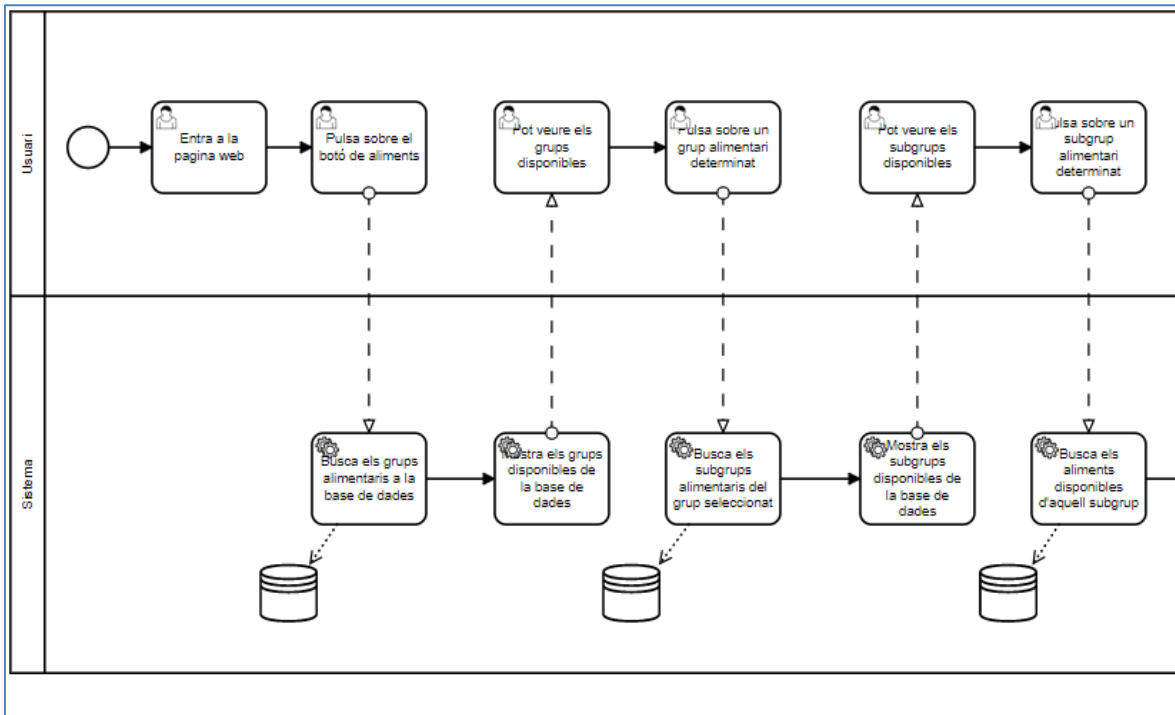
**Diagrama BPMN:**

Fig. 6.1.1.3 Diagrama BPMN sobre les característiques d'un aliment 1. Font: Elaboració pròpia.

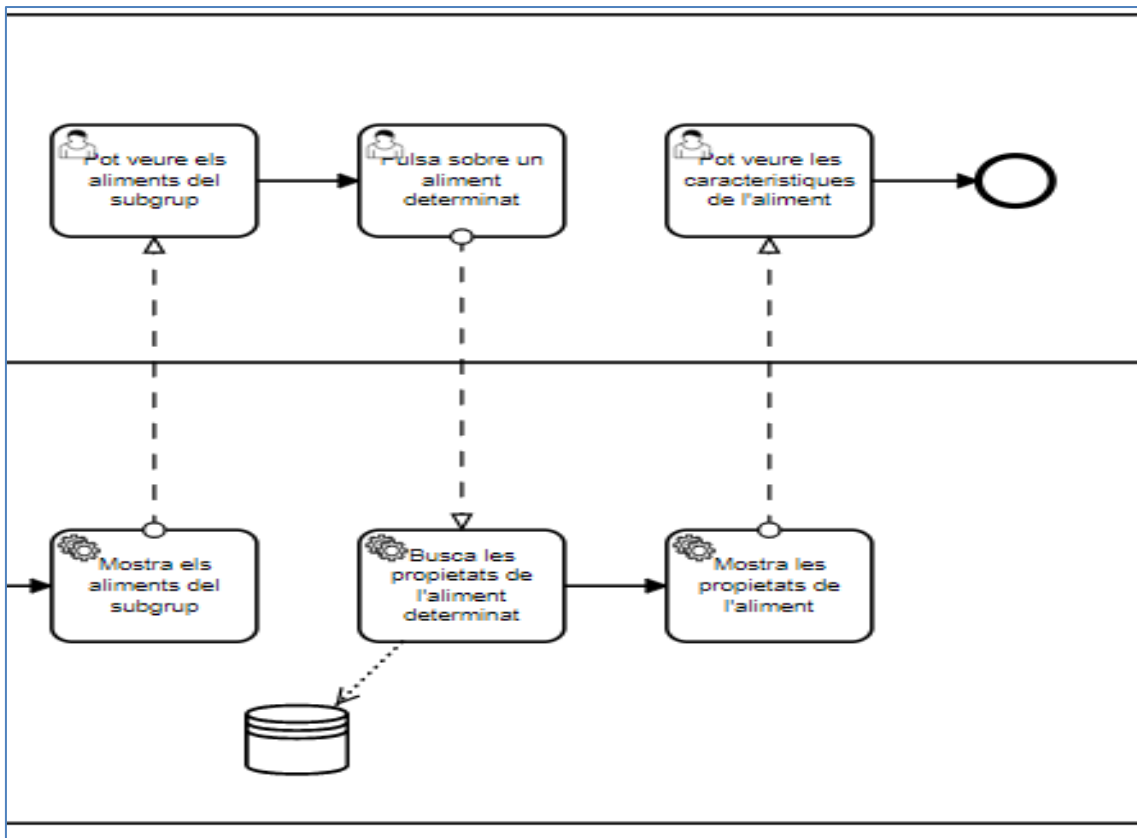


Fig. 6.1.1.4 Diagrama BPMN sobre les característiques d'un aliment 2. Font: Elaboració pròpia.

**Cas N°4:**

- **Nom:** Esborrar una dieta com a usuari administrador
- **Descripció:** L'usuari administrador vol esborrar una dieta
- **Actor:** Usuari administrador
- **Precondicions:** L'usuari es troba a la pàgina web i ja ha iniciat sessió
- **Postcondicions:** L'usuari ha esborrat una dieta
- **Flux Normal:**
  1. L'usuari accedeix a la pàgina web
  2. L'usuari polsa sobre el botó d'administració
  3. El sistema mostra la pantalla d'administració
  4. El sistema busca els usuaris disponibles i els mostra
  5. L'usuari polsa sobre el botó de dietes
  6. El sistema busca les dietes disponibles i les mostra
  7. L'usuari selecciona una dieta a la taula de l'esquerra
  8. L'usuari polsa el botó d'esborrar
  9. El sistema busca la dieta seleccionada i la borra
  10. L'usuari pot veure que la dieta seleccionada ha sigut eliminada

**Flux Alternatiu:**

1. No hi ha usuaris disponibles
2. No hi ha dietes disponibles
3. No es pot esborrar la dieta

**Diagrama BPMN:**

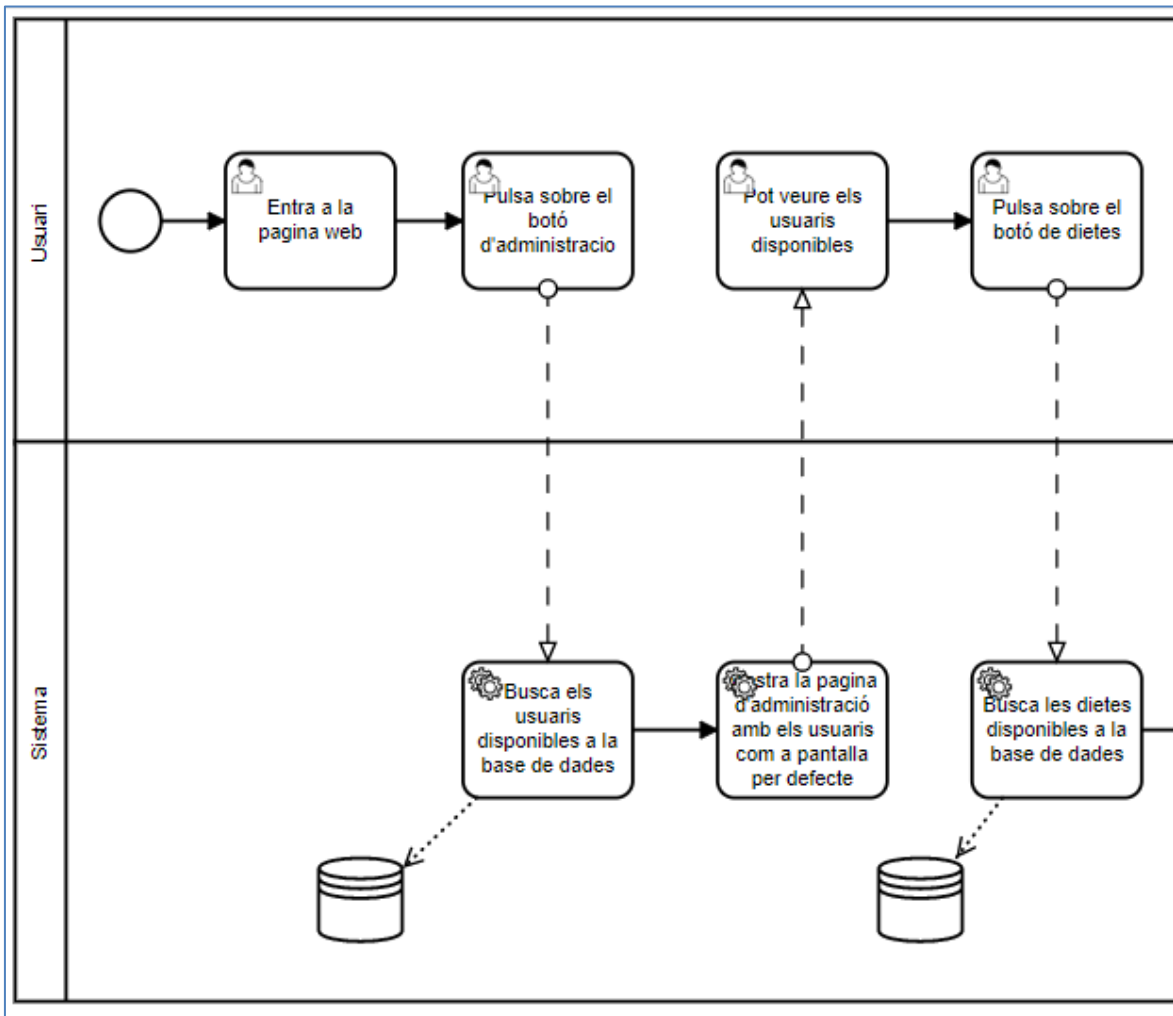


Fig. 6.1.1.5 Diagrama BPMN sobre esborrar una dieta 1. Font: Elaboració pròpia.

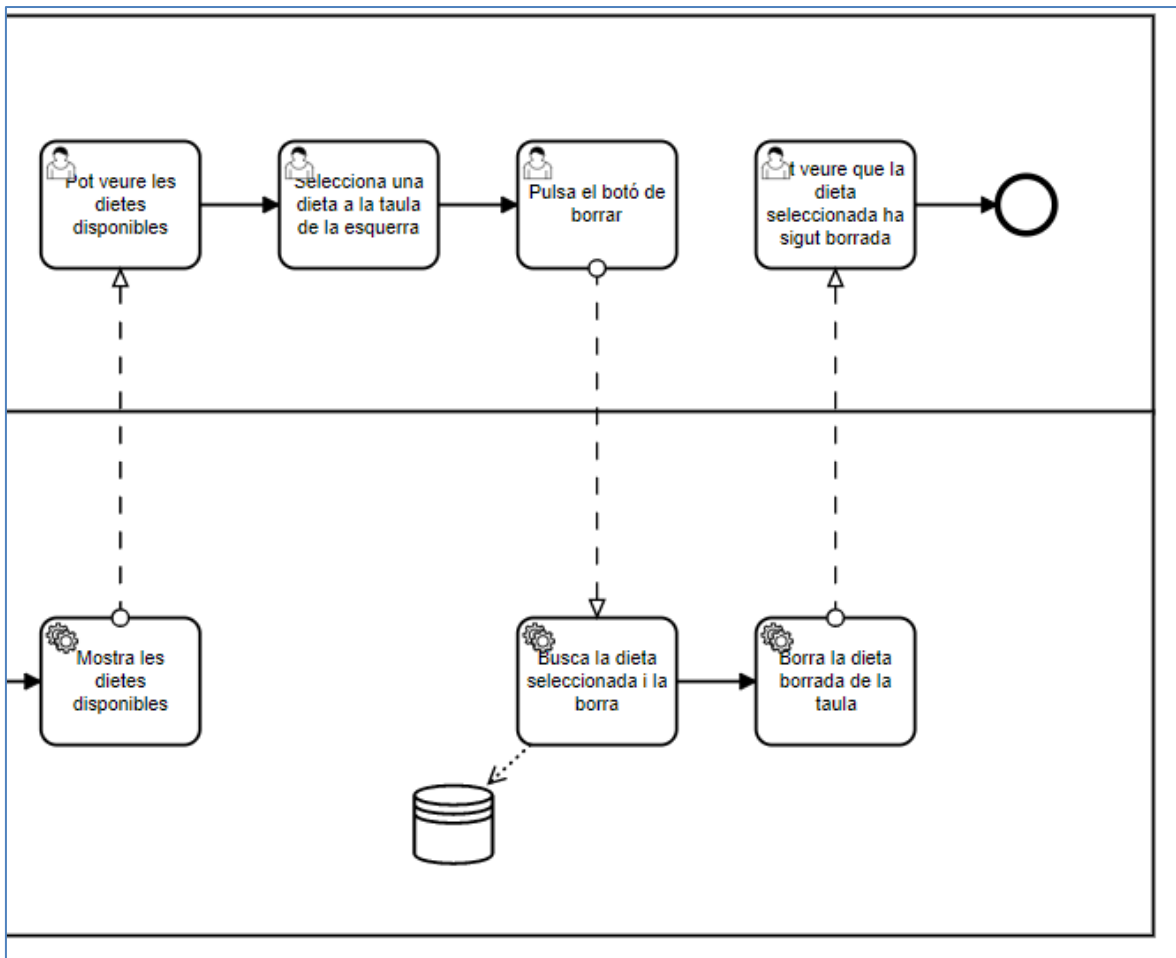


Fig. 6.1.1.6 Diagrama BPMN sobre esborrar una dieta 2. Font: Elaboració pròpia.

**Cas N°5:**

- **Nom:** Editar una malaltia com a usuari administrador
- **Descripció:** L'usuari administrador vol editar una malaltia
- **Actor:** Usuari administrador
- **Precondicions:** L'usuari es troba a la pàgina web i ja ha iniciat sessió
- **Postcondicions:** L'usuari ha actualitzat una malaltia

**Flux Normal:**

1. L'usuari accedeix a la pàgina web
2. L'usuari polsa sobre el botó d'administració
3. El sistema mostra la pantalla d'administració
4. El sistema busca els usuaris disponibles i els mostra
5. L'usuari polsa sobre el botó de malalties
6. El sistema busca les malalties disponibles i les mostra
7. L'usuari selecciona una malaltia a la taula de l'esquerra
8. L'usuari polsa el botó d'editar
9. El sistema permet editar les dades de la dreta
10. L'usuari canvia les dades que vulgui canviar
11. L'usuari polsa acceptar
12. El sistema busca la malaltia i actualitza les dades
13. El sistema buida el formulari i el deshabilita
14. L'usuari pot veure que les dades han sigut actualitzades

**Flux Alternatiu:**

1. No hi ha usuaris disponibles
2. No hi ha malalties disponibles

3. No es pot actualitzar la malaltia
4. No hi posa tots els camps

### Diagrama BPMN:

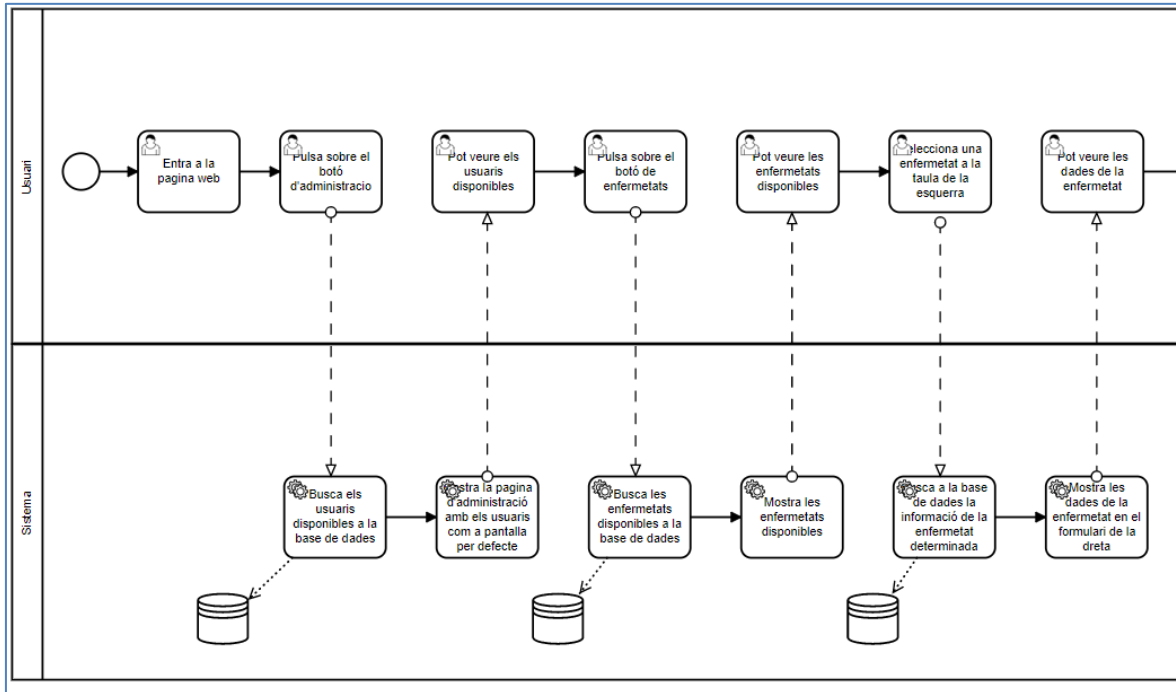


Fig. 6.1.1.7 Diagrama BPMN sobre editar una malaltia com a administrador 1. Font: Elaboració pròpia.



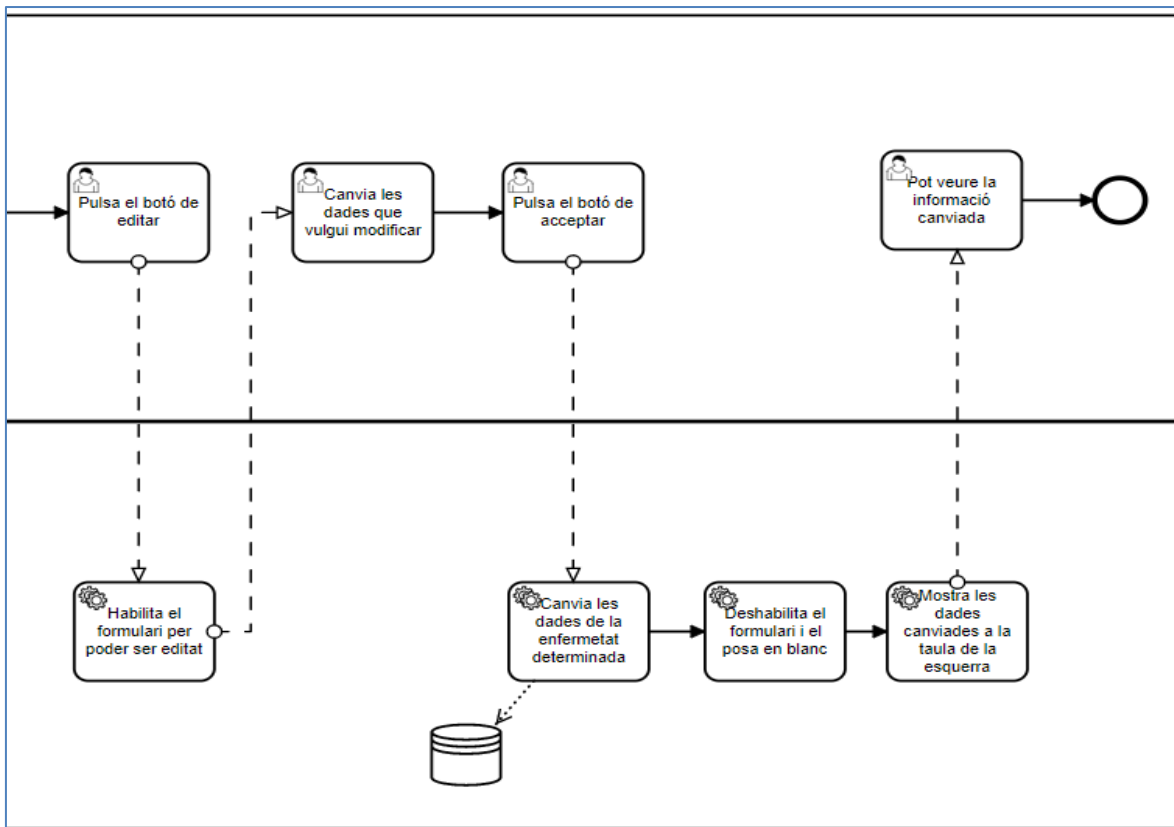


Fig. 6.1.1.5 Diagrama BPMN sobre editar una malaltia com a administrador 2. Font: Elaboració pròpia.

**Cas N°6:**

- **Nom:** Afegir un aliment com a usuari administrador
- **Descripció:** L'usuari administrador vol afegir un aliment
- **Actor:** Usuari administrador
- **Precondicions:** L'usuari es troba a la pàgina web i ja ha iniciat sessió
- **Postcondicions:** L'usuari ha afegit un aliment

**Flux Normal:**

1. L'usuari accedeix a la pàgina web
2. L'usuari polsa sobre el botó d'administració
3. El sistema mostra la pantalla d'administració
4. El sistema busca els usuaris disponibles i els mostra
5. L'usuari polsa sobre el botó d'aliments
6. El sistema busca els aliments disponibles i els mostra
7. L'usuari polsa el botó d'afegir
8. El sistema permet editar les dades de la dreta
9. L'usuari afegeix les dades de l'aliment
10. L'usuari polsa acceptar
11. El sistema afegeix el nou aliment
12. El sistema buida el formulari i el deshabilita
13. L'usuari pot veure que les dades han sigut afegides

**Flux Alternatiu:**

1. No hi ha usuaris disponibles
2. No hi ha aliments disponibles
3. No es pot afegir l'aliment

4. No hi posa tots els camps

Diagrama BPMN:

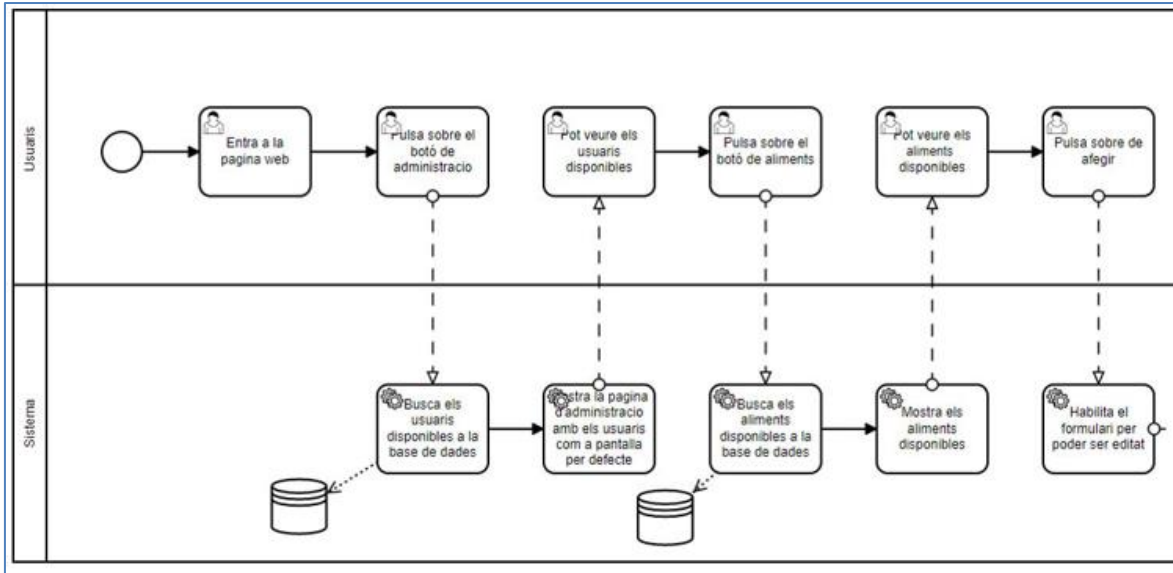


Fig. 6.1.1.9 Diagrama BPMN sobre afegir un aliment 1. Font: Elaboració pròpia.

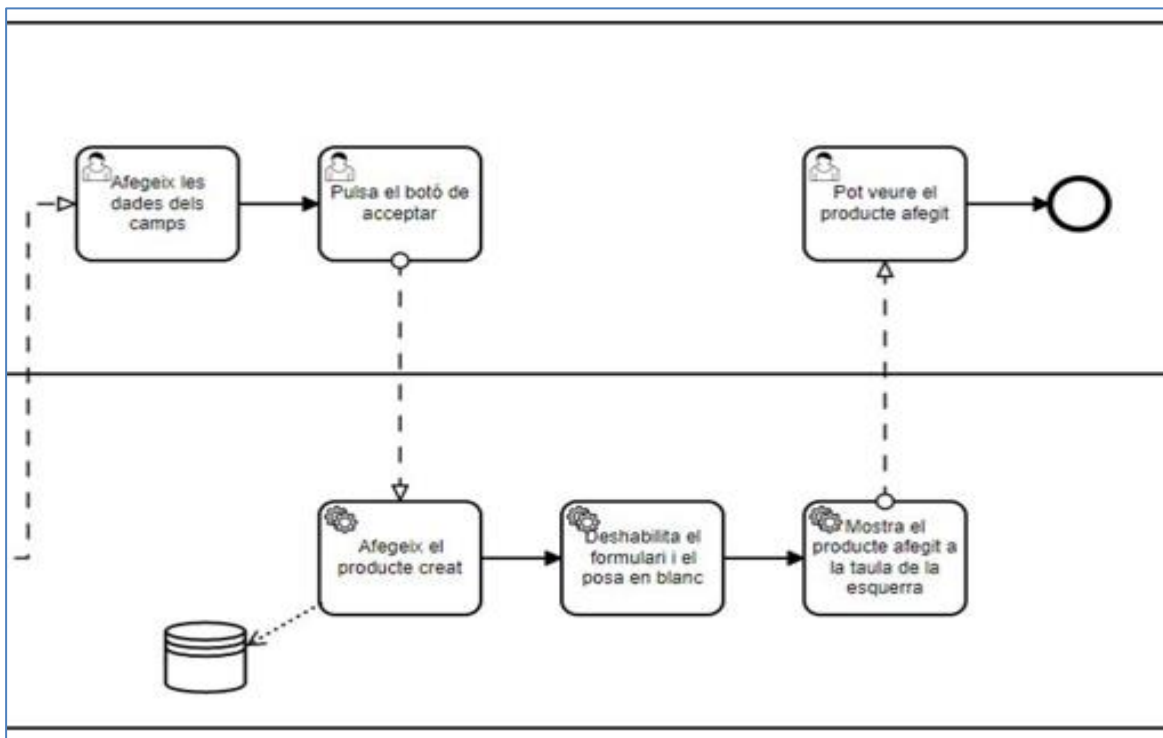


Fig. 6.1.1.10 Diagrama BPMN sobre afegir un aliment 2. Font: Elaboració pròpia.

## Disseny de base de dades

El que és més primordial d'aquesta part és la base de dades, és el cor d'aquesta aplicació web. Si no funciona correctament, no es pot interactuar amb la interfície gràfica. Per fer el disseny i la implementació de la base de dades s'ha utilitzat MYSQL Workbrench.

A continuació es mostra el disseny de la base de dades per parts donada la quantitat de taules que conté:

### Disseny de la part de l'usuari

A la figura 6.1.1.11 es poden visualitzar les entitats *usuari*, en la qual es guarda la informació de l'usuari, *enfermetat*, en la qual es guarda la informació de les malalties, *usuari\_has\_dieta*, en la qual es guarden les dietes d'un usuari i *usuari\_has\_enfermetat*, on es guarden les malalties d'un usuari.

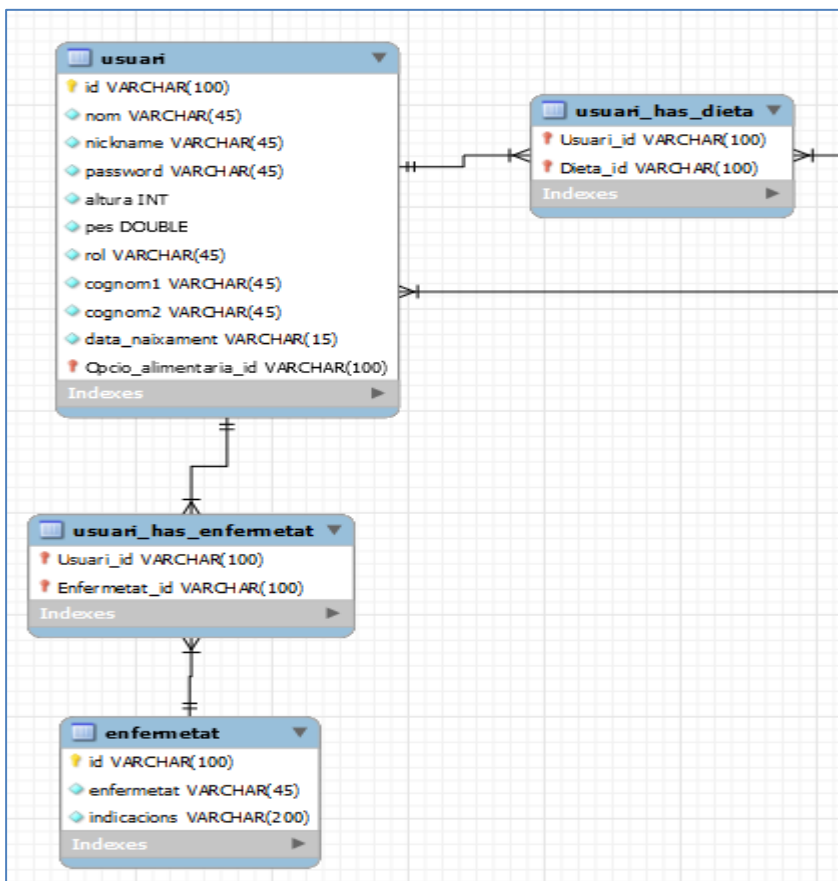


Fig. 6.1.1.11 Disseny relacional de la base de dades part de l'usuari. Font: Elaboració pròpia.

**Disseny de la part de la dieta:**

A la figura 6.1.1.12 es poden visualitzar les entitats *dieta*, en la qual es guarda la informació de les dietes, *menu*, en la qual es guarda la informació dels menús, *moment\_de\_menjar*, en la qual es guarda la informació de les diferents variants dels menús, *menu\_has\_moment\_de\_menjar*, on es guarden les variants de cada menú, *moment\_de\_menjar\_has\_aliment*, on es guarden els aliments de cada variant i *dieta\_has\_grup\_alimentari*, on es guarden tots els grups alimentaris d'una dieta determinada.

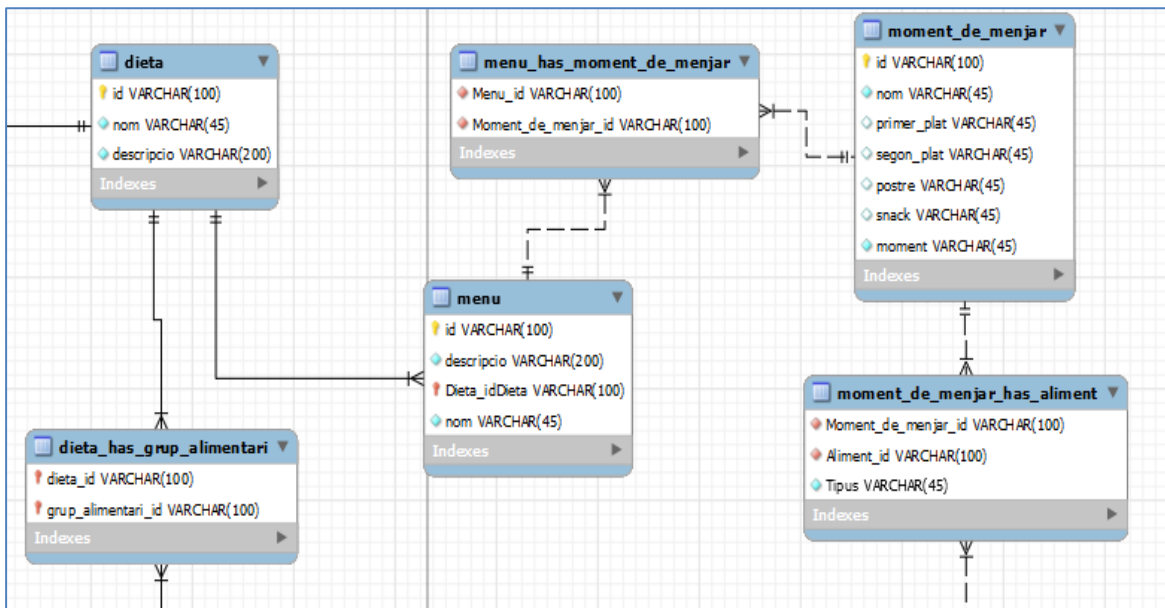


Fig. 6.1.1.12 Disseny relacional de la base de dades part de la dieta. Font: Elaboració pròpia.

### Disseny de la part dels grups alimentaris:

A la figura 6.1.1.13 es poden visualitzar les entitats *grup\_alimentari*, en la qual es guarda la informació dels grups alimentaris, *subgrup\_alimentari*, en la qual es guarda la informació dels subgrups alimentaris, *opcio\_alimentaria*, en la qual es guarda la informació de cada opció alimentaria i *opció\_alimentaria\_has\_subgrup\_alimentari* en què es guarden les opcions alimentàries de cada subgrup alimentari.

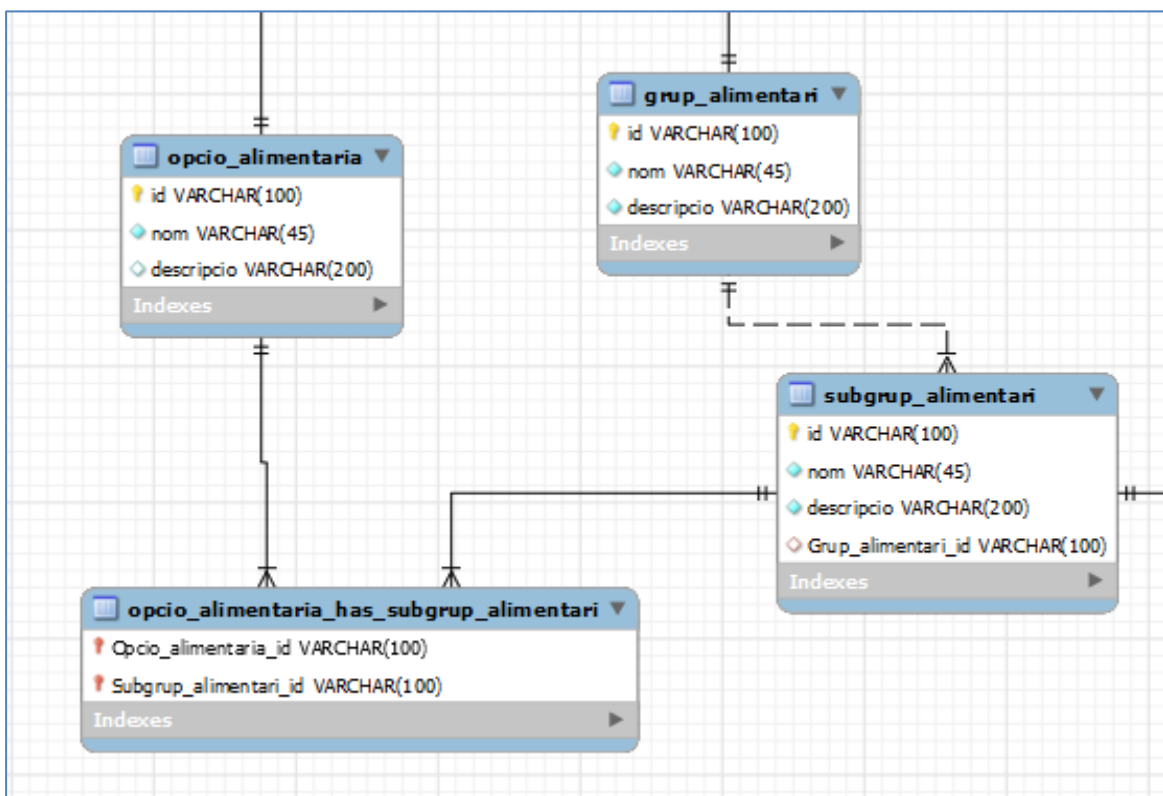


Fig. 6.1.1.13 Disseny relacional de la base de dades part dels grups alimentaris. Font: Elaboració pròpia.

### Disseny de la part dels aliments:

A la figura 6.1.1.14 es poden visualitzar les entitats *aliment*, en la qual es guarda la informació dels aliments, *vitamina*, en la qual es guarda la informació de les vitamines, *mineral*, en la qual es guarda la informació de cada mineral, *informació\_nutricional* en la qual es guarda la informació de la informació nutricional de cada aliment, *aliment\_has\_vitamina*, en la qual es guarden les vitamines de cada aliment, *aliment\_has\_mineral*, en la qual es guarden els minerals de cada aliment i *aliment\_has\_aliment*, on es guarden els aliments alternatius a cada aliment. A la taula *codis\_aliments\_reals* es guarda les equivalències entre el codi de l'aliment creat per la o el dietista i el codi de barres de l'aliment real per poder escanejar el producte i que surtin les característiques genèriques de l'aliment.

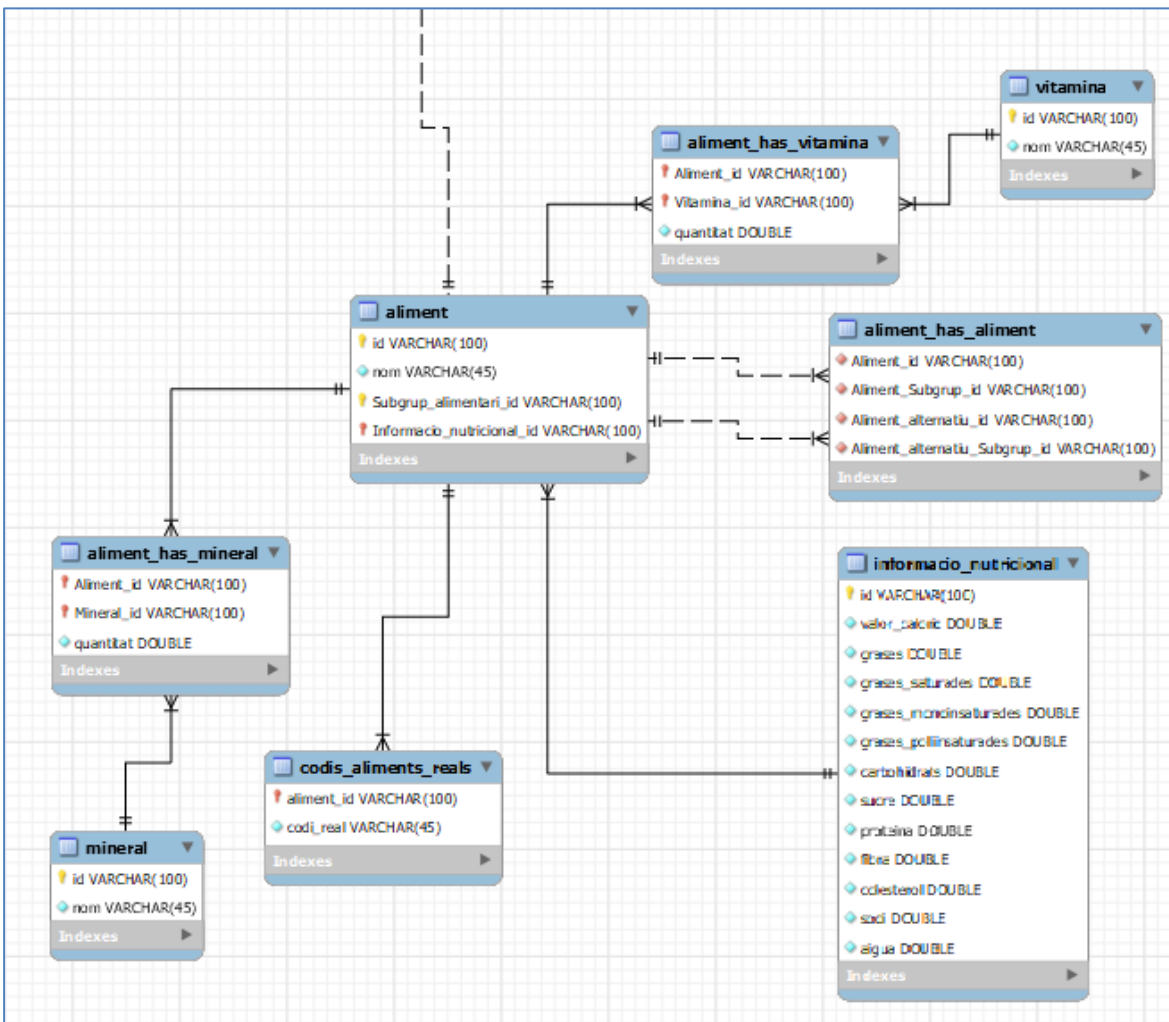


Fig. 6.1.1.14 Disseny relacional de la base de dades part dels aliments. Font: Elaboració pròpia.

S'han creat entitats per poder emmagatzemar les dades que es necessiten: dieta, menú, aliment, usuari, etc. Cada una amb els atributs que s'han considerat necessaris.

Cada usuari pot tenir una o més enfermetats i cada enfermetat pot tenir un o més usuaris. Cada usuari té assignada una opció alimentària i una opció alimentaria pot estar assignada a molts usuaris. Un usuari pot tenir un o més dietes assignades i una dieta pot tenir un o més usuaris assignats.

Cada dieta pot estar assignada a un o molts menús i un menú pot estar assignat a una dieta. Aquest menú té diferents parts així com primer plat, segon, postres o snacks, que estan definits en els moments de menjar, així com esmorzar, dinar, berenar i sopar. Cada àpat té assignat uns aliments en el que estan fets els menjars. Cada dieta té assignada un o més grups alimentaris i un grup alimentari té assignat una o moltes dietes.

Cada grup d'aliments té assignats uns subgrups i cada subgrup ha d'estar dins d'un grup alimentari. Cada subgrup té un o molts aliments i un aliment pertany a un subgrup. Cada subgrup té assignat una o moltes opcions alimentàries i cada opció alimentaria pot pertànyer a un o molts subgrups.

Cada aliment té assignat un valor nutricional i un valor nutricional pertany a un aliment, té una o moltes vitamines i minerals assignats i cadascun d'ells pot estar dins d'un o més aliments.

A continuació s'ha generat el codi en SQL a partir el model i s'han creat les entitats correctament, es pot apreciar en resultat a la figura 6.1.1.15.



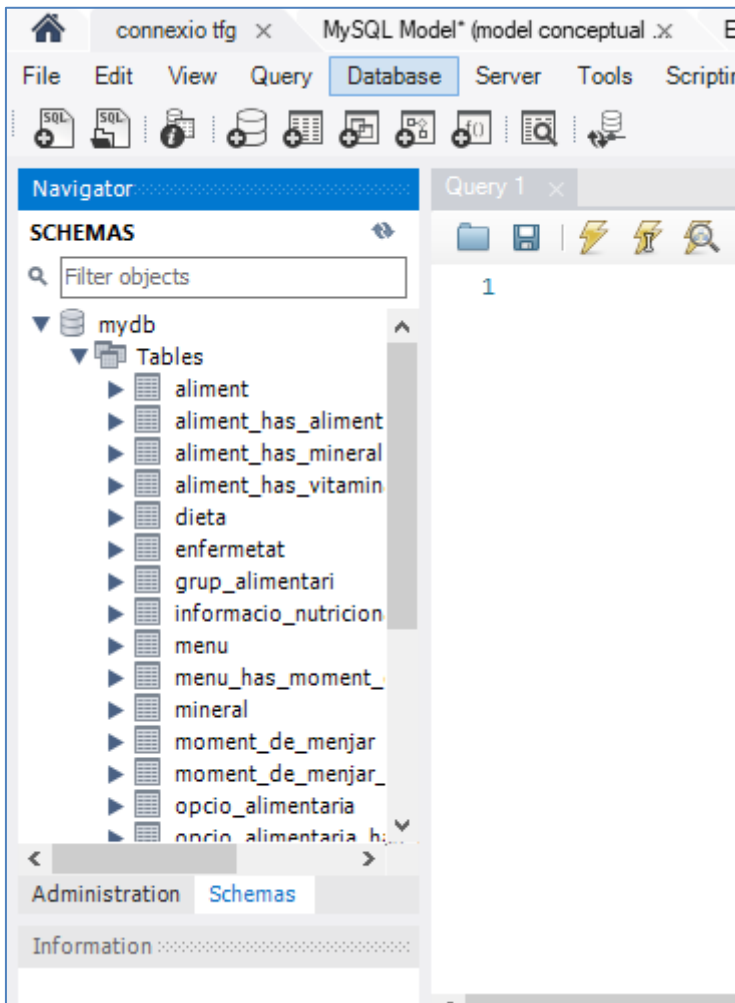


Fig. 6.1.1.15 Taules de la base de dades. Font: Elaboració pròpia.

## Diagrama de classes

Per poder entendre el funcionament intern de l'aplicació s'han realitzat diagrames de classes. En aquest document només es mostra el diagrama de classes de les dietes per poder entendre el funcionament del patró capes. Per fer els següents diagrames s'ha fet servir l'eina Lucidchart [13].

La interacció de les classes s'inicia amb el RestController, és des d'on es comunica l'api amb les entrades externes. Un cop s'ha fet la petició el RestController es comunica amb el Controller i aquest dependent de la petició es comunica amb un DAO determinat de cara a poder extreure dades de la base de dades. Els DAOs són els que s'encarreguen de comunicar-se amb la base de dades mitjançant un jdbcTemplate. Un jdbcTemplate és una classe interna de Java que permet la connexió entre bases de dades i el backend.

### Dietes:

A continuació a la figura 6.1.1.16 es pot observar l'exemple d'un diagrama de classes de les dietes:

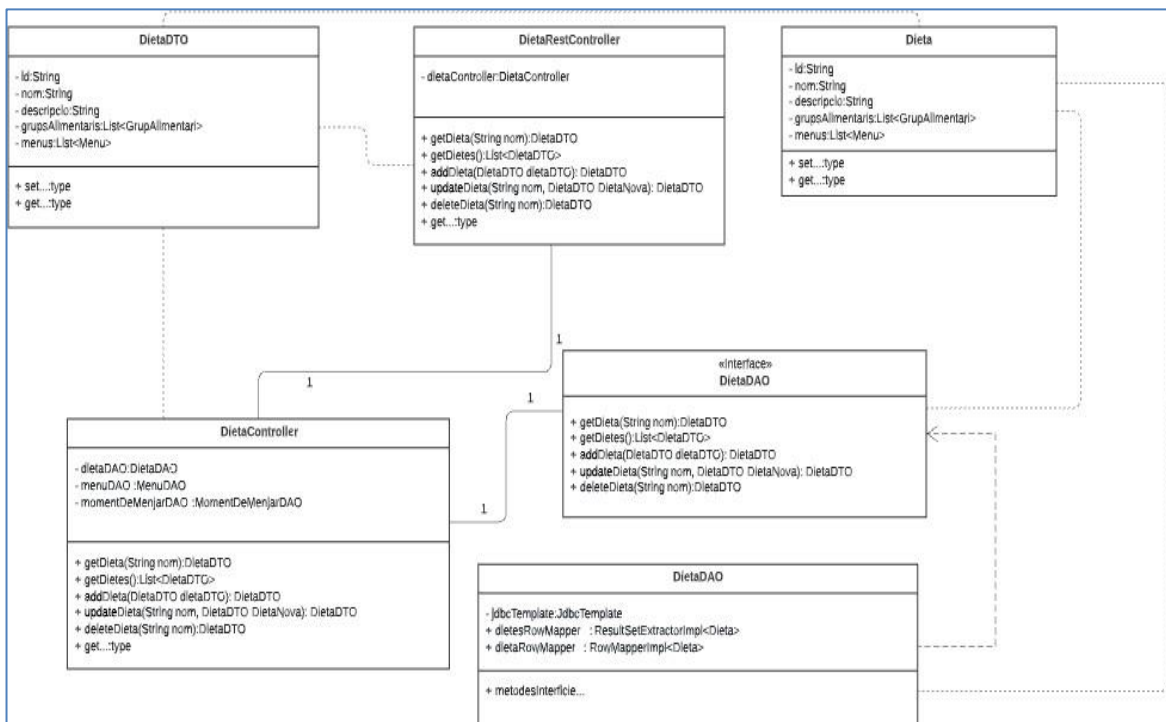


Fig. 6.1.1.16. Diagrama de classes de les dietes. Font: Elaboració pròpia.

Per efectuar el backend de la gestió de la web s'ha utilitzat el patró capes per tal de connectar els diferents serveis amb la base de dades.

A continuació s'explica el funcionament del patró capes:

El domini són totes aquelles classes de les quals es guarda informació d'una dieta. Un DTO és una còpia d'aquesta classe del domini. Les classes del domini només es poden comunicar únicament amb les capes que tenen a sota o a sobre, per això es necessiten els DTOs que són els que s'encarreguen de portar la informació a les capes superiors. El domini només es pot comunicar amb els DAOs (accés a la base de dades) i amb els serveis que en aquest cas serien els control·lers. Els DTOs es comuniquen amb els control·lers i amb el RestController ja que es troba a la capa de serveis o d'aplicació.

Per poder comunicar-se amb la base de dades s'utilitzen els DAOs, que són dins la capa de persistència. Els RestControllers estan a la capa de presentació i els control·lers a la capa de serveis. Es pot observar a la figura 6.1.1.17 com seria el patró capes visualment.

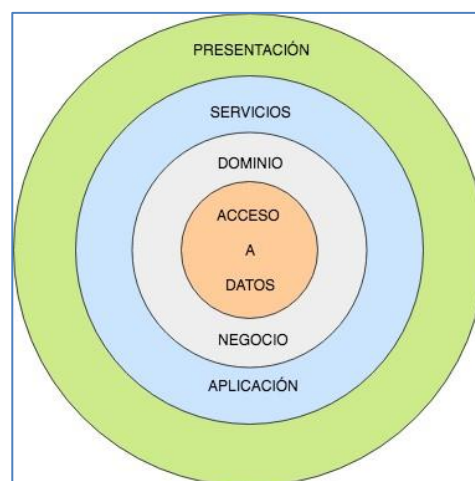


Fig. 6.1.1.17. Dibuix del patró capes. Font: [15]

A continuació es pot veure alguns exemples del contingut de cada capa:

A la figura 6.1.1.18 es pot apreciar la capa de presentació dins de la carpeta api, la capa de serveis està dins de la carpeta application, la capa del domini està dins de la capa de domain i la capa de accés a les dades és dins de la carpeta de persistence. Dins de cada capa es poden veure les diferents classes usades.

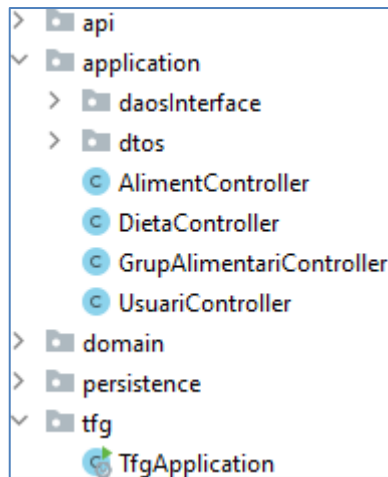


Fig. 6.1.1.18. Distribució de carpetes de les capes. Font: Elaboració pròpia.

**Domini:**

A la figura 6.1.1.19 es pot veure el contingut de la carpeta de la capa de domini. Hi ha totes les classes que guarden informació a la base de dades.

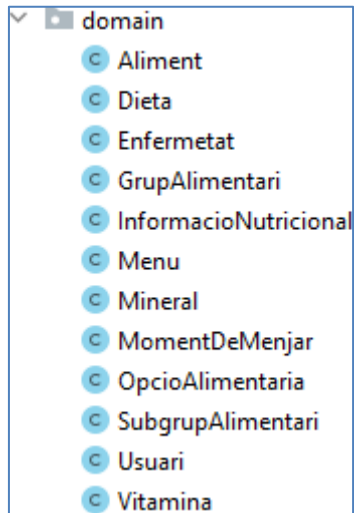


Fig. 6.1.1.19. Fitxers de la capa de domini. Font: Elaboració pròpia.

**Persistència:**

En la figura 6.1.1.20 podem observar el contingut de la capa de persistència. Hi ha totes les classes necessàries per poder connectar cada element del domini amb la base de dades

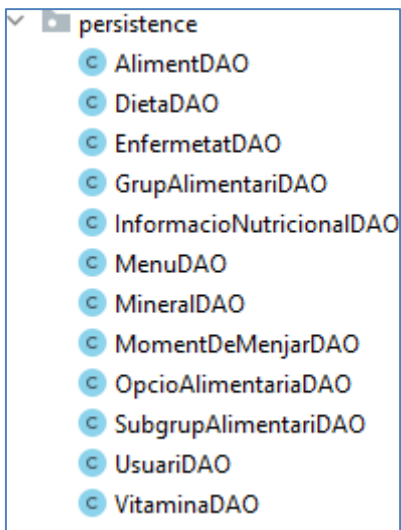


Fig. 6.1.1.20. Fitxers de la capa de persistència. Font: Elaboració pròpia.

### Aplicació:

En la figura 6.1.1.21 es pot apreciar el contingut de la capa d'aplicació. Hi ha la carpeta de les interfícies DAOs, els DTOs i els control·lers de cada element del Domini.

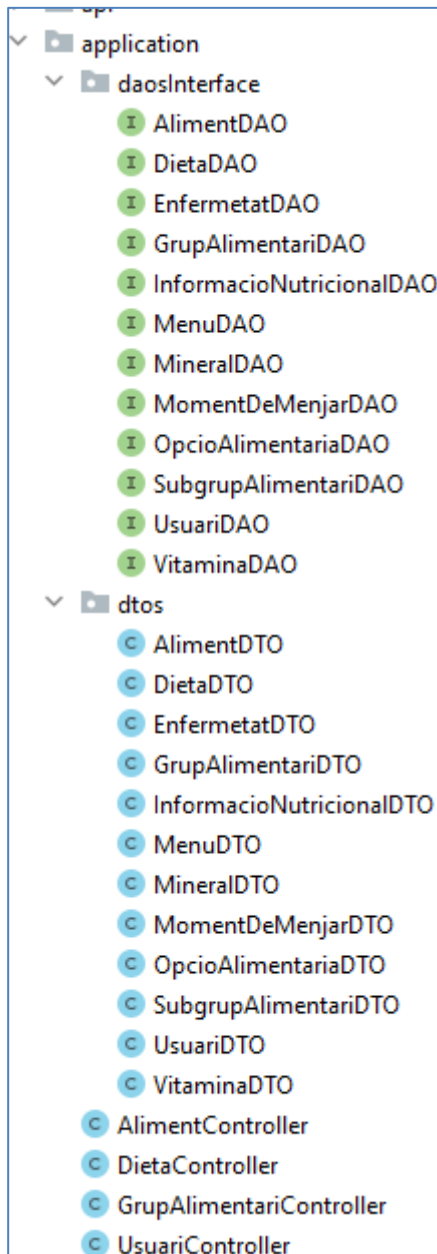


Fig. 6.1.1.21 Fitxers de la capa d'aplicació. Font: Elaboració pròpia.

**Api:**

A la figura 6.1.1.22 es pot apreciar el contingut de la capa d'aplicació. S'ha creat quatre classes, un per cada grup de taules de la base de dades.

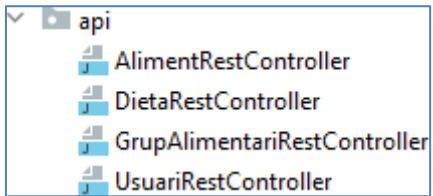


Fig. 6.1.1.22 Fitxers de la capa de l'api. Font: Elaboració pròpia

## **Desenvolupament de l'API REST**

Per la part de desenvolupament de l'API s'ha fet ús d'Spring, un framework de codi obert per Java.

A continuació es posen alguns punts d'accés que s'utilitzen a l'aplicació. Només es mostren els més representatius. En tots els casos hi ha punts d'accés per esborrar, afegir, modificar i obtenir dades de cada element del domini, ja siguin dietes, menús, grups alimentaris, subgrups, opcions alimentàries, usuaris, malalties, vitamines, minerals i aliments.

### **Per afegir un aliment a un subgrup alimentari**

Aquest punt de accés s'utilitza per afegir un aliment existent dins d'un subgrup alimentari.

<http://localhost:8080/aliment/{subgrup}>

- Funcionalitat: Afegir un aliment a un subgrup alimentari
- Tipus de petició: POST
- Body: Tipus AlimentDTO
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

### **Per afegir un aliment**

Aquest punt de accés s'utilitza per afegir un aliment a la base de dades.

<http://localhost:8080/aliment>

- Funcionalitat: Afegir un aliment
- Tipus de petició: POST
- Body: Tipus AlimentDTO
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.



### **Per obtenir els aliments**

Aquest punt de accés s'utilitza per obtenir tots els aliments de la base de dades.

<http://localhost:8080/aliment>

- Funcionalitat: Obtenir els aliments
- Tipus de petició: GET
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades dels Aliments en format JSON.

### **Per obtenir un aliment**

Aquest punt de accés s'utilitza per obtenir un aliment determinat de la base de dades.

<http://localhost:8080/aliment/{nom}>

- Funcionalitat: Obtenir un aliment
- Tipus de petició: GET
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

### **Per actualitzar un aliment**

Aquest punt de accés s'utilitza per actualitzar un aliment determinat de la base de dades.

<http://localhost:8080/alimentAlternatiu/{nom}>

- Funcionalitat: Actualitzar un aliment
- Tipus de petició: PUT
- Body: Tipus AlimentDTO
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte

- 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

### **Per esborrar un aliment**

Aquest punt de accés s'utilitza per esborrar un aliment determinat de la base de dades.

<http://localhost:8080/aliment/{nom}>

- Funcionalitat: Esborrar un aliment
- Tipus de petició: DELETE
- Caràcter: Públic
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

### **Per afegir un aliment alternatiu**

Aquest punt de accés s'utilitza per afegir un aliment alternatiu d'un aliment a la base de dades.

<http://localhost:8080/alimentAlternatiu/{nom}/{nomAlternatiu}>

- Funcionalitat: Afegir un aliment alternatiu a un aliment
- Tipus de petició: POST
- Caracter: Public
  - Resposta HTTP:
    - 200. Petició correcte
    - 500. Error de sistema
  - Resposta: Dades del Aliment en format JSON.

### **Per obtenir els aliments alternatius d'un aliment**

Aquest punt de accés s'utilitza per obtenir els aliments alternatius d'un aliment de la base de dades.

<http://localhost:8080/alimentAlternatiu/{nom}>

- Funcionalitat: Obtenir els aliments alternatius d'un aliment
- Tipus de petició: GET
- Caracter: Public
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

### **Per esborrar un aliment alternatiu**

Aquest punt de accés s'utilitza per esborrar un aliment alternatiu d'un aliment determinat de la base de dades.

<http://localhost:8080/alimentAlternatiu/{nom}/{nomAlternatiu}>

- Funcionalitat: Esborrar un aliment alternatiu
- Tipus de petició: DELETE
- Caracter: Public
- Resposta HTTP:
  - 200. Petició correcte
  - 500. Error de sistema
- Resposta: Dades del Aliment en format JSON.

## Dissenys del FrontEnd

En aquest apartat es poden apreciar tots els dissenys realitzats de la plataforma web.

S'ha dissenyat com és la vista de cada pantalla web de l'aplicació de cara l'usuari. S'ha pensat fer un disseny per la pàgina d'inici de sessió, dietes, aliments, grups i subgrups alimentaris característiques de cada aliment i la pàgina d'administració per la dietista.

Els dissenys s'han fet amb Lucidchart. Cada pantalla té un menú a dalt per poder navegar entre les pàgines tal com es poden veure a continuació:

### Pantalla inicial:

Com es pot apreciar a la figura 6.1.1.23, és la pantalla d'inici de l'aplicació. Es pot veure imatges informant els usuaris sobre el funcionament de l'aplicació amb fletxes que indiquen el menú per visualitzar més imatges. A baix hi ha contingut de l'aplicació mòbil en format text i alguna imatge.

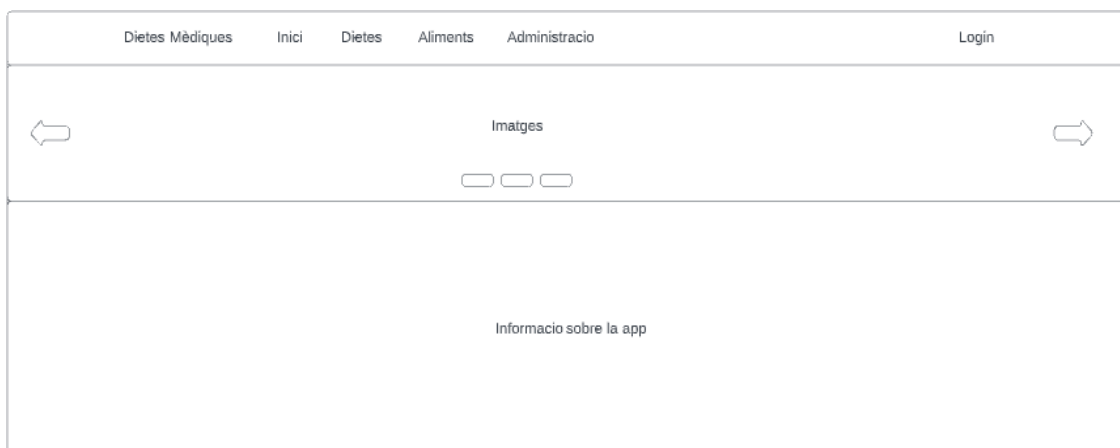


Fig. 6.1.1.23 Disseny pantalla inicial. Font: Elaboració pròpia

### **Pantalla de dietes:**

Com es pot observar a la figura 6.1.1.24, en aquesta pantalla es poden veure totes les dietes disponibles a la base de dades. Hi ha el nom de cada dieta amb una petita descripció sobre elles.



Fig. 6.1.1.24 Disseny pantalla de dietes. Font: Elaboració pròpia

### **Pantalla de Grups/Subgrups alimentaris:**

Com es pot veure a la figura 6.1.1.25, és el disseny de la pantalla de grups i subgrups alimentaris. Aquesta té uns cercles amb el nom de cada grup/subgrup al seu interior. Un cop es polsa cada cercle aquest el porta a la pantalla d'aliments o subgrups depèn d'on estigui l'usuari.



Fig. 6.1.1.25 Disseny pantalla de grups i subgrups alimentaris. Font: Elaboració pròpia

### Pantalla dels aliments

Com es pot apreciar a la figura 6.1.1.26, aquesta pantalla conté una llista amb dos columnes de tots els aliments determinats d'un subgrup alimentari. Aquesta llista només conté el nom de l'aliment.

The screenshot shows a web interface with a navigation bar at the top containing the following links: Dietes Mèdiques, Inici, Dietes, Aliments, Administracio, and Login. Below the navigation bar, the main content area is titled 'Aliments disponibles'. It contains a grid of 10 input fields arranged in two columns and five rows. Each input field contains the placeholder text 'Nom Aliment'.

Fig. 6.1.1.26 Disseny pantalla d'aliments. Font: Elaboració pròpia

### Pantalla de les característiques d'un aliment:

Com es pot veure a la figura 6.1.1.27, en aquesta pantalla es visualitzen totes les característiques d'un aliment determinat, així com valor calòric, carbohidrats, proteïnes, grasses, vitamines, minerals...

The screenshot shows a web interface with a navigation bar at the top containing the following links: Dietes Mèdiques, Inici, Dietes, Aliments, Administracio, and Login. Below the navigation bar, the main content area is titled 'Nom del producte'. It displays nutritional information in a structured layout. At the top, there are two boxes: 'Valor Caloric X' and 'Carbohidrats X'. Below these are two more boxes: 'Proteïnes X' and 'Grases X'. At the bottom, there are three separate boxes: 'Informacio del valor nutricional', 'Informacio de les vitamines', and 'Informacio dels minerals'.

Fig. 6.1.1.27 Disseny pantalla amb les característiques d'un aliment. Font: Elaboració pròpia

### **Pantalla de login:**

Com es pot apreciar a la imatge 6.1.1.28, aquesta pantalla permet identificar usuaris i així poder iniciar sessió. Tots els usuaris hi tenen accés, però només l'administrador podrà veure la pantalla d'administració.

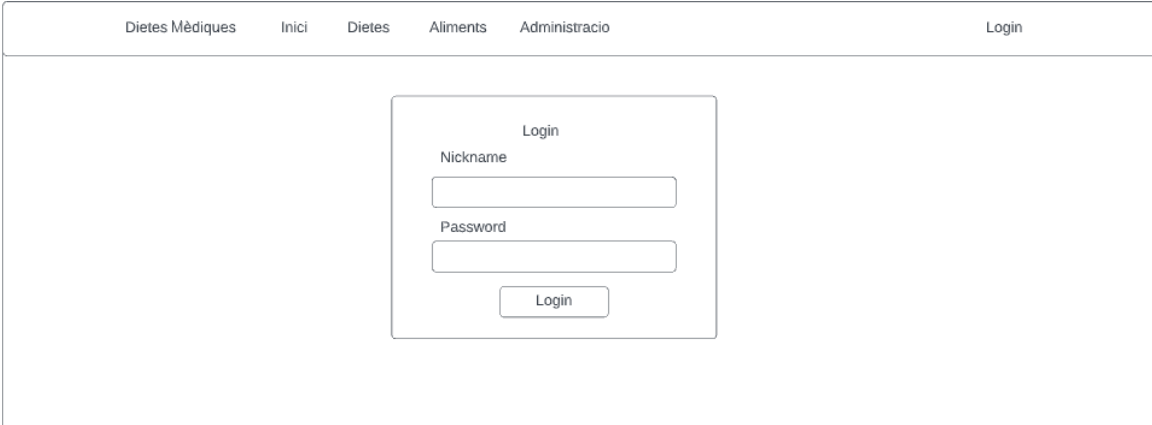


Diagrama de la pantalla de login. A la part superior hi ha un menú amb les opcions: Dietes Mèdiques, Inici, Dietes, Aliments, Administració i Login. Al centre hi ha un formulari amb el títol "Login", un camp de text per a "Nickname", un camp de text per a "Password" i un botó "Login".

Fig. 6.1.1.28 Disseny pantalla de login. Font: Elaboració pròpia

### **Pantalla d'administració:**

Com es pot observar a la figura 6.1.1.29, aquesta és la pantalla per poder administrar tot el contingut de la plataforma. Es pot administrar usuaris, malalties, dietes, grups o subgrups alimentaris, aliments... A l'esquerra de la pantalla hi ha una taula amb els ítems corresponents per poder seleccionar. A la dreta hi ha un formulari per poder visualitzar, afegir o editar l'ítem seleccionat.

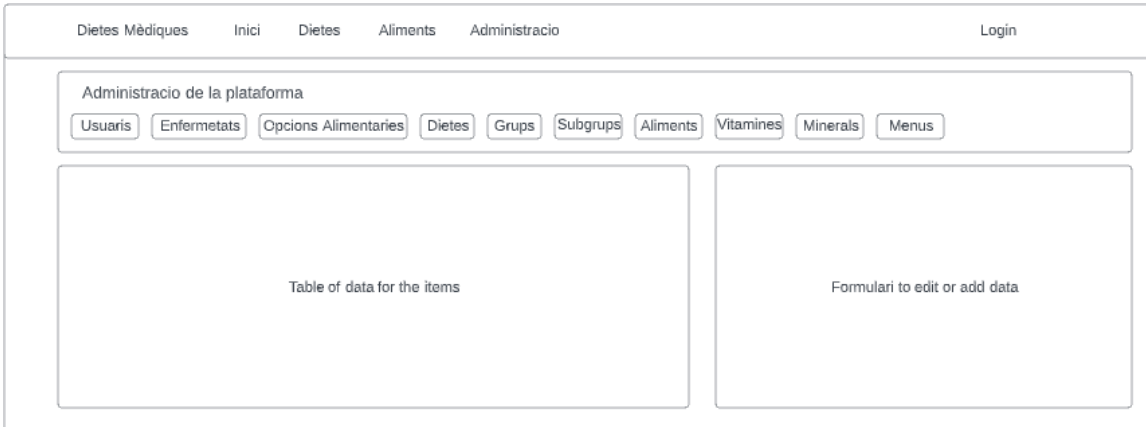


Diagrama de la pantalla d'administració. A la part superior hi ha un menú amb les opcions: Dietes Mèdiques, Inici, Dietes, Aliments, Administració i Login. A sota hi ha un títol "Administració de la plataforma" i una sèrie de botons: Usuaris, Enfermetats, Opcions Alimentaries, Dietes, Grups, Subgrups, Aliments, Vitamines, Minerals i Menus. A la part inferior hi ha dues zones: "Table of data for the items" a l'esquerra i "Formulari to edit or add data" a la dreta.

Fig. 6.1.1.29 Disseny pantalla d'administració. Font: Elaboració pròpia

Un cop els dissenys ja s'han fet les pàgines tenen el següents aspectes:

### **Pantalla inicial:**



Fig. 6.1.1.30 Pantalla inicial. Font: Elaboració pròpia

### **Pantalla de dietes:**

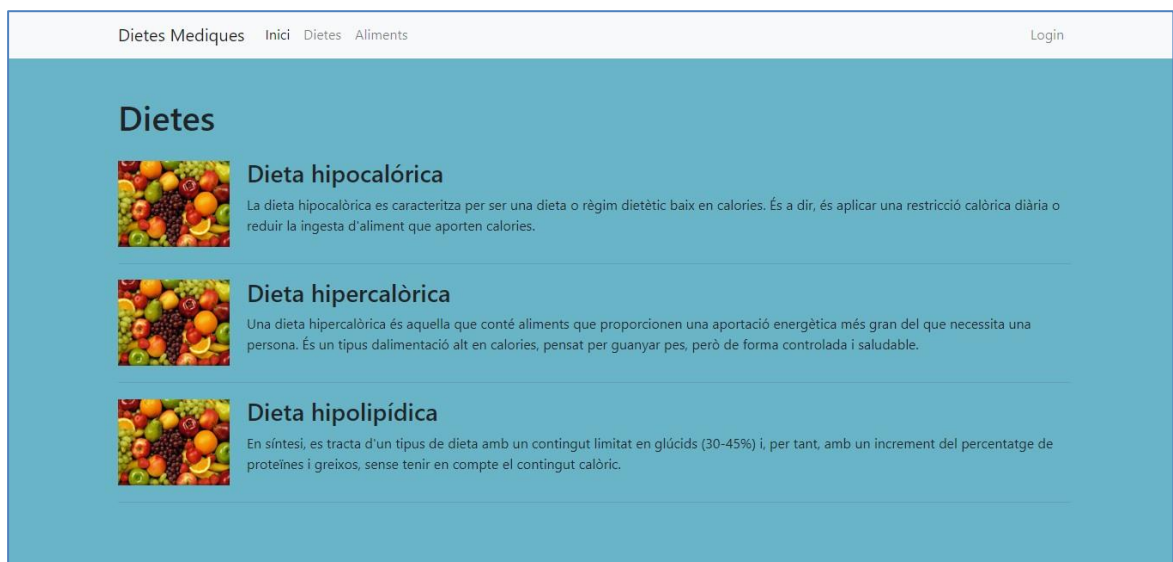


Fig. 6.1.1.31 Pantalla de dietes. Font: Elaboració pròpia



**Pantalla Aliments:**

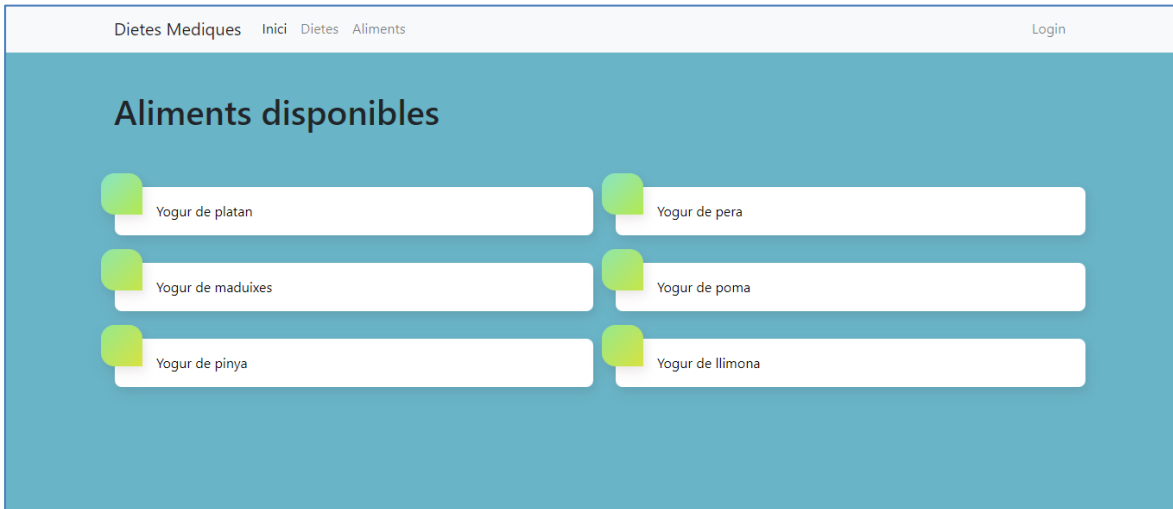


Fig. 6.1.1.32 Pantalla d'aliments. Font: Elaboració pròpia

**Pantalla de les característiques d'un aliment:**

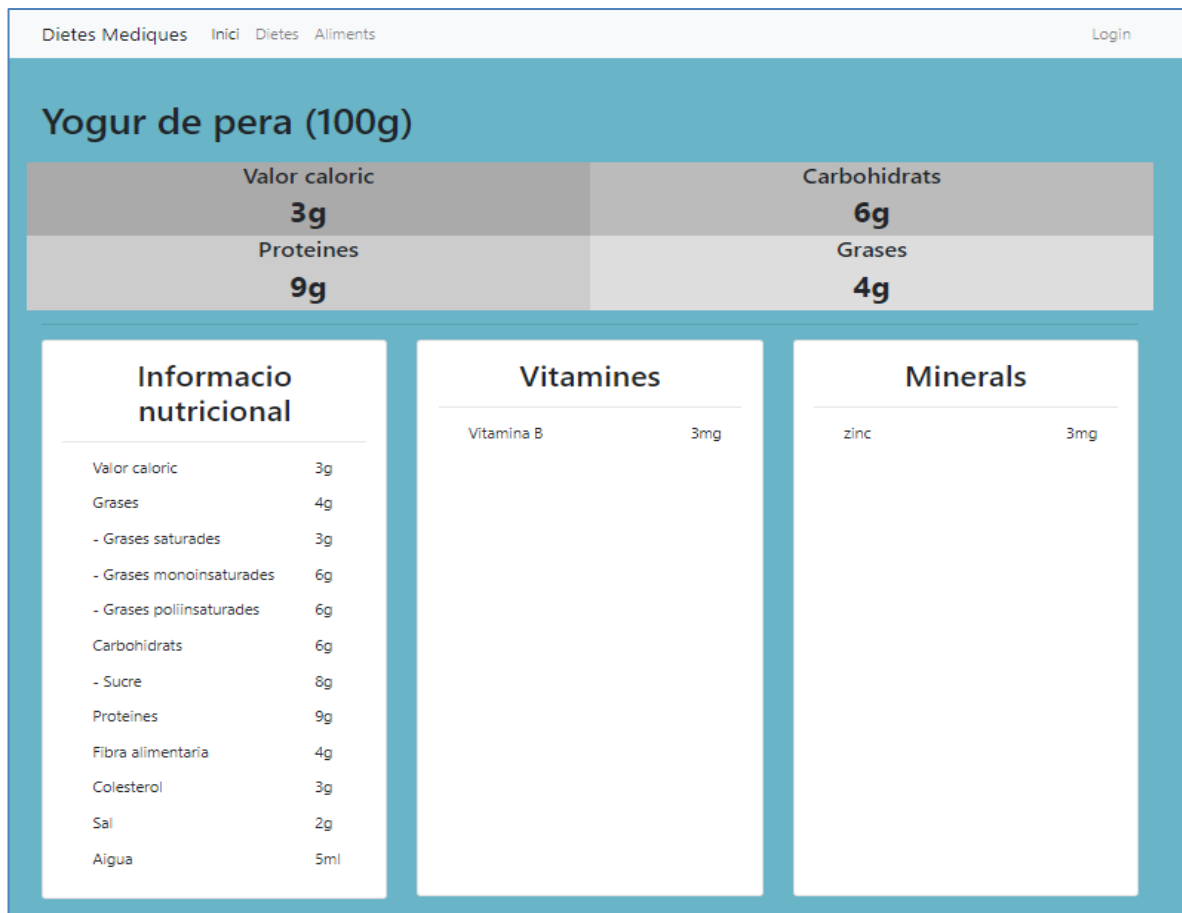


Fig. 6.1.1.33 Pantalla de les característiques d'un aliment. Font: Elaboració pròpia

**Pantalla administració:**

Dietes Mediques Inici Dietes Aliments Login

## Administracio de la plataforma

Selecciona l'area en que vols visualitzar o modificar el contingut

Usuaris Enfermetats Opcions alimentaries Dietes Grups Subgrups Aliments Vitaminas Minerals Menus

### Usuaris Disponibles

NICKNAME	NOM	COGNOMS	ROL	EDIT	BORRAR
jgonzalezgo	Jose	Gonzalez Gomez	PACIENT		
cvillarroyaol	Cesar	Villarroya Ollero	ADMIN		

### Informacio d'usuari

Nom  Primer Cognom

Segon Cognom  Nickname

Contrasenya  Data naixament  dd/mm/aaaa

Fig. 6.1.1.34 Pantalla d'administració. Font: Elaboració pròpia

**Pantalla de login:**

Dietes Mediques Inici Dietes Aliments Login

### Sign In

Nickname

Password

Fig. 6.1.1.35 Pantalla de login. Font: Elaboració pròpia

## 6.1.2 Comentaris sobre la implementació:

A continuació es pot veure més exemples dels continguts, només es mostren els més significatius. Per consultar amb més detall mirar les classes del projecte.

### RestController:

El RestController és el que permet connectar les peticions HTTP externes amb el backend. Tal com es veu a la figura 6.1.2.1, es fa servir mètodes per consultar les vitamines d'un aliment, així com actualitzar-les o esborrar-les. Des d'aquest arxiu connecta el RestController amb el Controller.

```
@RestController
public class AlimentRestController {

    private AlimentController alimentController;

    public AlimentRestController(A AlimentController alimentController) { this.alimentController = alimentContro

    //metodes de vitamines
    @PostMapping("/vitamina")
    public VitaminaDTO addVitamina(@RequestBody VitaminaDTO vitamina){
        return this.alimentController.addVitamina(vitamina);
    }
    @GetMapping("/vitamina")
    public List<VitaminaDTO> getVitamines() { return this.alimentController.getVitamines(); }

    @GetMapping("/vitamina/{nom}")
    public VitaminaDTO getVitamina(@PathVariable String nom) { return this.alimentController.getVitamina(nom);

    @PutMapping("/vitamina/{nom}")
    public VitaminaDTO updateVitamina(@PathVariable String nom,@RequestBody VitaminaDTO vitaminaNova ){
        return this.alimentController.updateVitamina(nom, vitaminaNova);
    }

    @DeleteMapping("/vitamina/{nom}")
    public VitaminaDTO deleteVitamina(@PathVariable String nom) { return this.alimentController.deleteVitamina
```

Fig. 6.1.2.1 Part de contingut de la classe RestController. Font: Elaboració pròpia

## Controller:

El Controller és qui s'encarrega de connectar la capa d'api amb la capa de la persistència. Com es pot apreciar a la figura 6.1.2.2, dins del Controller té els atributs de totes les interfícies que usa per connectar amb la base de dades, en aquest cas un aliment es connecta amb l'aliment, la informació nutricional, les vitamines i els minerals.

```
@Component
public class AlimentController {
    private AlimentDAO alimentDAO;
    private InformacioNutricionalDAO informacioNutricionalDAO;
    private VitaminaDAO vitaminaDAO;
    private MineralDAO mineralDAO;

    public AlimentController(AlimentDAO alimentDAO, InformacioNutricionalDAO informacioNutricionalDAO, Vitamina
        this.alimentDAO = alimentDAO;
        this.informacioNutricionalDAO = informacioNutricionalDAO;
        this.vitaminaDAO = vitaminaDAO;
        this.mineralDAO = mineralDAO;
    }

    public AlimentDAO getAlimentDAO() { return alimentDAO; }

    public InformacioNutricionalDAO getInformacioNutricionalDAO() { return informacioNutricionalDAO; }

    public VitaminaDAO getVitaminaDAO() { return vitaminaDAO; }

    public MineralDAO getMineralDAO() { return mineralDAO; }

    // Metodes aliment
    public AlimentDTO getAliment(String nom) { return new AlimentDTO(this.alimentDAO.getAliment(nom)); }

    public List<AlimentDTO> getAliments(){
```

Fig. 6.1.2.2 Part de contingut de la classe Controller 1. Font: Elaboració pròpia

A la figura 6.1.2.3 s'observa el contingut d'alguns mètodes usats, com obtenir aliments, o afegir un aliment. També es veu la transformació utilitzada de *AlimentDTO* a *Aliment*.

```

public AlimentDTO getAliment(String nom) { return new AlimentDTO(this.alimentDAO.getAliment(nom)); }

public List<AlimentDTO> getAliments(){
    List<AlimentDTO> aliments=new ArrayList<>();

    for (Aliment a: this.alimentDAO.getAliments()){
        aliments.add(new AlimentDTO(a));
    }

    return aliments;
}

public AlimentDTO addAliment(AlimentDTO alimentDTO, String subgrup){

    return new AlimentDTO(this.alimentDAO.addAliment(AlimentDTO2Aliment(alimentDTO), subgrup));
}

private Aliment AlimentDTO2Aliment(AlimentDTO alimentDTO){
    InformacioNutricional informacioNutricional=new InformacioNutricional(alimentDTO.getInformacioNutricio
        alimentDTO.getInformacioNutricional().getGrasesMonoInsaturades(), alimentDTO.getInformacioNutra
        alimentDTO.getInformacioNutricional().getSucre(), alimentDTO.getInformacioNutricional().getPro
        alimentDTO.getInformacioNutricional().getSodi(), alimentDTO.getInformacioNutricional().getAigu
    List<Vitamina> vitamines=new ArrayList<>();
    List<Mineral> minerals=new ArrayList<>();
}

```

Fig. 6.1.2.3 Part de contingut de la classe Controller 2. Font: Elaboració pròpia

### **Interfície:**

Les interfícies DAO permeten ser implementades per les classes de la persistència i fer que totes tinguin un model de mètodes semblants. Com s'aprecia a la figura 6.1.2.4, dins de la interfície *AlimentDTO* es poden veure tots els mètodes que implementa la interfície que després la classe DAO conté.

```

public interface AlimentDAO {
    public Aliment getAliment(String nom);
    public List<Aliment> getAliments();
    public Aliment addAliment(Aliment aliment, String subgrup);
    public Aliment updateAliment(String nom, Aliment alimentNou);
    public Aliment deleteAliment(String nom);
    public Aliment addAlimentAlternatiu(String nom, String nomAlternatiu);
    public List<Aliment> getAlimentsAlternatius(String nom);
    public Aliment deleteAlimentAlternatiu(String nom, String nomAlternatiu);
}

```

Fig. 6.1.2.4 Contingut d'una interfície. Font: Elaboració pròpia

### **Domini:**

Les classes del domini permeten guardar la informació de cada objecte. Com es pot observar a la figura 6.1.2.5, hi ha el contingut de la classe *Aliment*, es pot veure alguns atributs que emmagatzemen la informació de la base de dades.

```
public class Aliment {  
  
    private String codi;  
    private String nom;  
    private InformacioNutricional informacioNutricional;  
    private List<Mineral> minerals;  
    private List<Vitamina> vitamines;  
  
    public Aliment(){}  
  
    public Aliment(String id, String nom, InformacioNutricional informacioNutricional, List<Mineral> minerals,  
        List<Vitamina> vitamines) {  
  
        this.codi = id;  
        this.nom = nom;  
        this.informacioNutricional = informacioNutricional;  
        this.minerals = minerals;  
        this.vitamines = vitamines;  
    }  
  
    public Aliment(String codi, String nom) {  
  
        this.codi = codi;  
        this.nom = nom;  
        this.informacioNutricional = new InformacioNutricional();  
        this.minerals = new ArrayList<Mineral>();  
        this.vitamines = new ArrayList<Vitamina>();  
    }  
}
```

Fig. 6.1.2.5 Contingut d'un element del domini. Font: Elaboració pròpia

**DTO:**

En aquests tipus de classes es guarden tota la informació de cada classe del domini a manera de còpia. Com es pot observar a la figura 6.1.2.6, dins la classe del *AlimentDTO* conté els mateixos atributs que la seva classe del domini, l'única diferència és el constructor, ja que el DTO es crea a partir de la classe del domini.

```
public class AlimentDTO {  
  
    private String codi;  
    private String nom;  
    private InformacioNutricionalDTO informacioNutricional;  
    private List<MineralDTO> minerals;  
    private List<VitaminaDTO> vitamines;  
  
    public AlimentDTO(){}  
  
    public AlimentDTO(Aliment aliment){  
        this.codi=aliment.getCodi();  
        this.nom= aliment.getNom();  
        if(aliment.getInformacioNutricional()==null)this.informacioNutricional=null;  
        else this.informacioNutricional= new InformacioNutricionalDTO(aliment.getInformacioNutricional());  
        if(aliment.getMinerals()==null)this.minerals=null;  
        else this.minerals=this.mineralsDTOS(aliment.getMinerals());  
        if(aliment.getVitamines()==null)this.vitamines=null;  
        else this.vitamines=this.vitaminesDTOS(aliment.getVitamines());  
    }  
  
    private List<MineralDTO> mineralsDTOS(List<Mineral> minerals){  
        List<MineralDTO> mineralsDTOS=new ArrayList<>();  
        for(Mineral m: minerals){  
            mineralsDTOS.add(new MineralDTO(m));  
        }  
    }  
}
```

Fig. 6.1.2.6 Contingut d'exemple d'un DTO. Font: Elaboració pròpia

**DAO:**

Les classes DAO són les que connecten directament amb la base de dades mitjançant consultes SQL. A la imatge 6.1.2.7 es poden veure que els atributs dels DAOs sempre hi ha un JdbcTemplate, que és la classe Java que permet la connexió amb la base de dades. Els altres atributs són altres classes DAO que es necessiten per fer algunes peticions. Per recórrer cada fila de les taules de la base de dades s'utilitzen Rowmappers.

```
@Repository
public class AlimentDAO implements cat.tecnocampus.application.daosInterface.AlimentDAO{

    private JdbcTemplate jdbcTemplate;
    private SubgrupAlimentariDAO subgrupAlimentariDAO;
    private InformacioNutricionalDAO informacioNutricionalDAO;
    private VitaminaDAO vitaminaDAO;
    private MineralDAO mineralDAO;

    public AlimentDAO(JdbcTemplate jdbcTemplate, SubgrupAlimentariDAO subgrupAlimentariDAO, InformacioNutricionalDAO informacioNutricionalDAO, VitaminaDAO vitaminaDAO, MineralDAO mineralDAO) {
        this.jdbcTemplate = jdbcTemplate;
        this.subgrupAlimentariDAO = subgrupAlimentariDAO;
        this.informacioNutricionalDAO = informacioNutricionalDAO;
        this.vitaminaDAO = vitaminaDAO;
        this.mineralDAO = mineralDAO;
    }

    ResultSetExtractorImpl<Aliment> alimentsRowMapper =
        JdbcTemplateMapperFactory
            .newInstance()
            .addKeys("id")
            .newResultSetExtractor(Aliment.class);

    RowMapperImpl<Aliment> alimentRowMapper =
        JdbcTemplateMapperFactory
            .newInstance()
            .addKeys("id")
```

Fig. 6.1.2.7 Contingut d'exemple d'un DAO 1. Font: Elaboració pròpia



A la figura 6.1.2.8 es pot observar alguns mètodes utilitzats per fer les peticions, com obtenir un aliment, tots els aliments i tots els aliments d'un subgrup alimentari. Dins de cada mètode hi ha una sentència SQL per poder accedir a les dades desitjades.

```
@Override
public Aliment getAliment(String nom) {
    final var query = "select * from aliment where nom = ?";

    return jdbcTemplate.queryForObject(query, alimentRowMapperLazy, nom);
}

@Override
public List<Aliment> getAliments() {
    final var query = "select * from aliment";

    return jdbcTemplate.query(query, alimentRowMapperLazy);
}

@Override
public List<Aliment> getAlimentsSubgrup(String subgrup) {
    final var query = "select * from aliment where Subgrup_alimentari_id=?";
    final var query2 = "select id from subgrup_alimentari where nom=?";
    String id=jdbcTemplate.queryForObject(query2,String.class, subgrup);

    return jdbcTemplate.query(query, alimentRowMapperLazy, id);
}
```

Fig. 6.1.2.8 Contingut d'exemple d'un DAO 2. Font: Elaboració pròpia

Dins del projecte hi ha una carpeta que es diu pom.xml en el qual es guarden totes les dependències del projecte. Cada dependència permet fer funcions diferents dins de Spring. A la figura 6.1.2.9 es poden veure el contingut del fitxer pom.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.3</version>
    <relativePath/>
  </parent>
  <groupId>cat.tecnocampus</groupId>
  <artifactId>tfg</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>tfg</name>
  <description>tfg</description>
  <properties>
    <java.version>11</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
```

Fig. 6.1.2.9 Contingut del fitxer pom. Font: Elaboració pròpia

A la figura 6.1.2.10 es pot apreciar la dependència amb la que Spring permet connectar-se amb MYSQL:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.19</version>
</dependency>
```

Fig. 6.1.2.10 Dependència per poder connectar el mysql amb el Spring. Font: Elaboració pròpia

A la figura 6.1.2.11 es veu la dependència que permet interactuar amb la base de dades mitjançant *jdbcTemplate*.

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-jdbc</artifactId>  
</dependency>
```

Fig. 6.1.2.11 Dependència per poder connectar a la base de dades. Font: Elaboració pròpia

Es fa ús del fitxer *application.properties* dins la carpeta *resources*. Aquest fitxer conté totes les dades necessàries per poder connectar Spring amb les dades de la base de dades. A la figura 6.1.2.12 es pot observar el contingut d'aquest fitxer amb algunes credencials usades.

```
spring.jpa.hibernate.ddl-auto=update  
spring.datasource.url=jdbc:mysql://localhost:3306/mydb?useUnicode=true&useJDBCCompliant  
spring.datasource.username=root  
spring.datasource.password=cesar
```

Fig. 6.1.2.12 Propietats per poder connectar amb mysql. Font: Elaboració pròpia

Tota l'aplicació està connectada en local, per tant, per poder connectar el backend amb el frontend s'han utilitzat peticions amb localhost. Al frontend s'ha fet ús de la funció *fetch*. Aquesta funció permet fer peticions HTTP a una url determinada i es pot indicar el mètode d'aquesta. Hi ha els següents mètodes: *GET*, *POST*, *PUT* i *DELETE*. S'ha usat cadascuna d'aquestes per obtenir, canviar o esborrar informació de la base de dades. L'únic que pot actualitzar, esborrar o afegir la informació és el o la dietista.

Alguns exemples de codificació d'ús de la funció *fetch*:

El *fetch* és una funció de Javascript que permet enviar peticions HTTP i rebre respostes. S'ha usat per poder obtenir informació del backend des del frontend o transmetre informació cap allà. A la figura 6.1.2.13 es pot veure l'estructura d'una petició *GET*, en aquest cas aconseguir totes les dietes disponibles i a la figura 6.1.2.14 es pot veure la petició *DELETE* per poder esborrar qualsevol informació.

```
fetch('http://localhost:8080/dieta',)
.then(res => {
  if (res.ok) { console.log("HTTP request successful") }
  else { console.log("HTTP request unsuccessful") }
  return res
})
.then(res => res.json())
.then(dataDietes => {
```

Fig. 6.1.2.13 Exemple d'ús de *fetch* amb *GET*. Font: Elaboració pròpia

```
fetch('http://localhost:8080/'+categoria+'/' +img.parentNode.parentNode.id, {
method: "DELETE",
headers: {
  'Content-type': 'application/json'
}
```

Fig. 6.1.2.14 Exemple d'ús de *fetch* amb *DELETE*. Font: Elaboració pròpia

## 6.2 Aplicació mòbil

En aquesta part del projecte s'ha dissenyat i implementat l'aplicació mòbil per l'ús exclusiu de consultes per part dels usuaris finals. En aquesta aplicació poden consultar menús dissenyats per la dietista o simplement generar automàticament un menú dinàmic en el qual poden veure els menjars i aliments de cada menú generat amb una selecció de dies determinats que ells poden escollir. La implementació de l'aplicació s'ha fet mitjançant Microsoft Power Apps. Per poder accedir a l'aplicació és necessari tenir un compte Microsoft Outlook i descarregar l'aplicació de Power Apps. La dietista comparteix aquest programari amb els seus pacients. Per poder connectar les dades de la base de dades amb la Power App s'ha fet ús d'un gateway per poder passar les dades d'extrem a extrem.

### 6.2.1 Disseny de l'aplicació mòbil

Seguint el mateix procés de disseny que amb l'aplicació web, s'han dissenyat els casos d'ús, diagrames BPMN de cada procés de l'aplicació i s'ha fet el disseny de cada pantalla.

A continuació es detallen els casos d'ús més representatius juntament amb el seu diagrama BPMN. S'han dissenyat els diagrames BPMN dels casos d'ús mitjançant l'eina BPMN.io. Alguns d'aquests diagrames BPMN s'han mostrat separats per la gran quantitat de processos del cas d'ús.

#### Casos d'ús

S'han dissenyat alguns casos d'ús que són més significatius:

##### **Cas N°1:**

- **Nom:** Iniciar sessió
- **Descripció:** L'usuari vol iniciar sessió
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina d'inici de sessió
- **Postcondicions:** L'usuari ha iniciat sessió

**Flux Normal:**

1. L'usuari accedeix a l'aplicació mòbil
2. El sistema mostra a l'usuari un formulari per posar el seu nom d'usuari i la seva contrasenya
3. L'usuari introdueix les seves dades
4. L'usuari polsa el botó de login
5. El sistema comprova si és un usuari registrat
6. El sistema mostra la pàgina d'inici

**Flux Alternatiu:**

7. L'usuari no omple tots els camps del formulari i fa que no es mostri la pàgina d'inici
8. L'usuari no està registrat, per tant, no va a la pàgina d'inici

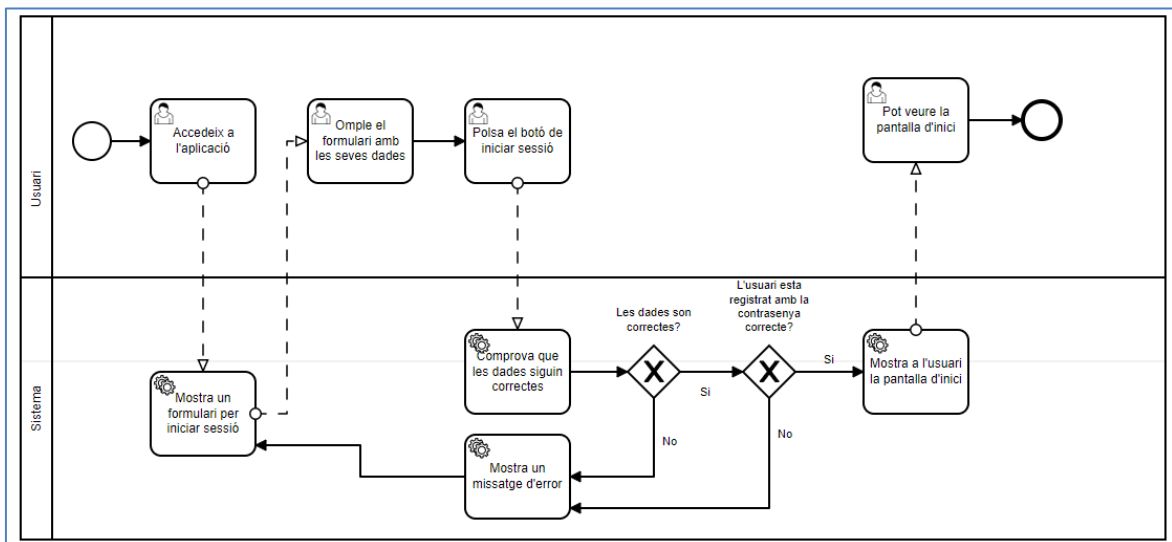
**Diagrama BPMN:**

Fig. 6.2.1.1 Digrama BPMN sobre iniciar sessió. Font: Elaboració pròpia

**Cas N°2:**

- **Nom:** Canviar les dades de l'usuari
- **Descripció:** L'usuari vol canviar les seves dades
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina d'inici
- **Postcondicions:** L'usuari ha modificat les seves dades

**Flux Normal:**

1. L'usuari polsa el botó de la icona de l'usuari
2. El sistema mostra a l'usuari un formulari amb totes les dades de l'usuari
3. L'usuari polsa el botó de la icona del llapis
4. El sistema fa modificable els camps perquè l'usuari els pugui canviar i la icona del llapis passa a ser un check
5. L'usuari modifica les dades que vulgui canviar i polsa sobre el botó de check
6. El sistema modifica les dades de l'usuari i fa que el formulari no sigui editable. També canvia la icona del check per un llapis

**Flux Alternatiu:**

1. L'usuari no polsa el botó per modificar i no pot canviar els camps

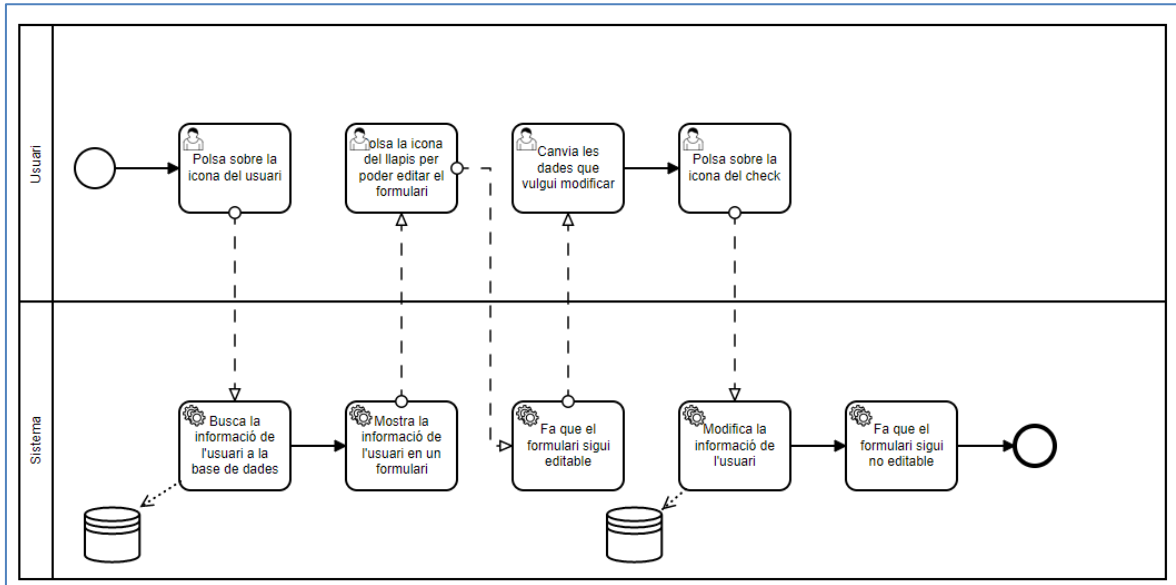
**Diagrama BPMN:**

Fig. 6.2.1.2 Diagrama BPMN sobre canviar les dades de l'usuari. Font: Elaboració pròpia



**Cas N°3:**

- **Nom:** Consultar les característiques d'un aliment mitjançant el buscador
- **Descripció:** L'usuari vol buscar si pot menjar un aliment i saber quines característiques té.
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina d'inici
- **Postcondicions:** L'usuari ha consultat l'aliment desitjat

**Flux Normal:**

1. L'usuari polsa el botó de la icona del buscador
2. El sistema mostra a l'usuari la pàgina del buscador
3. L'usuari escriu al buscador l'aliment que vol consultar
4. El sistema busca l'aliment desitjat, si l'usuari el pot menjar surt de color verd, si no en vermell
5. L'usuari polsa sobre el botó de l'ull al costat de l'aliment
6. El sistema mostra la pàgina de les característiques de l'aliment.

**Flux Alternatiu:**

1. L'usuari no troba l'aliment desitjat

## Diagrama BPMN:

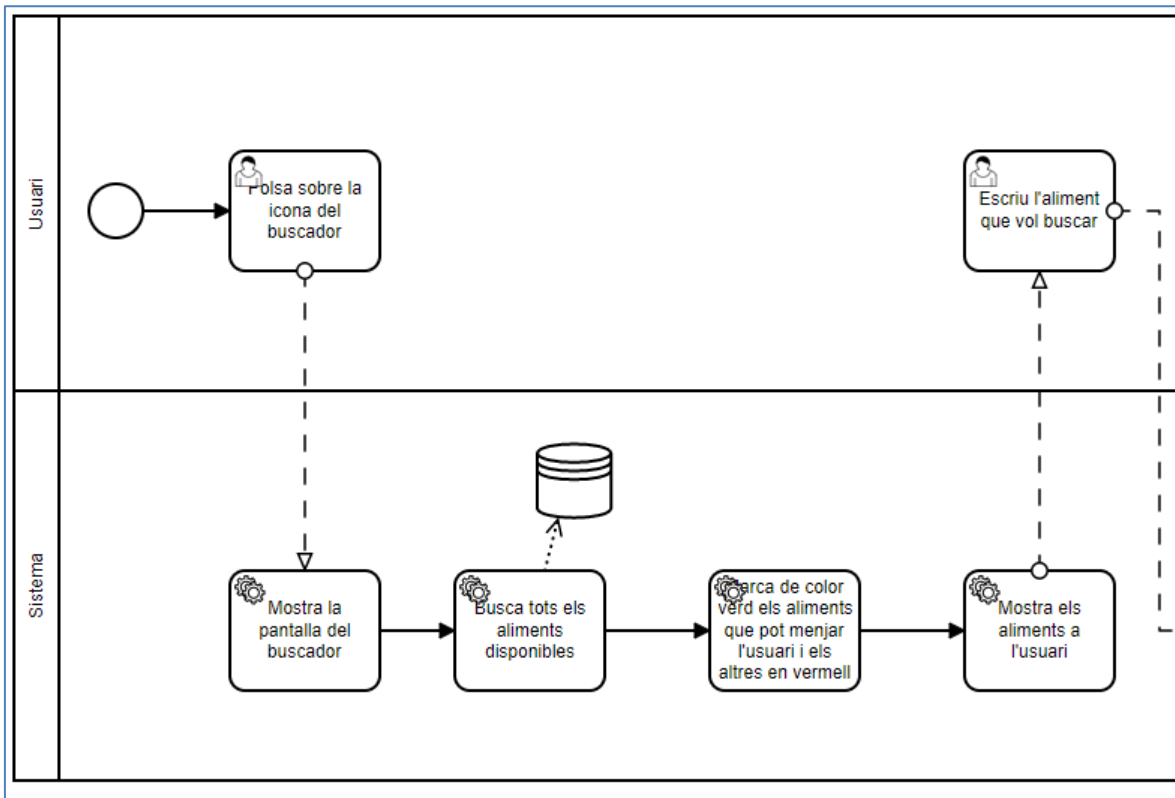


Fig. 6.2.1.3 Diagrama BPMN sobre consultar les característiques d'un aliment mitjançant el buscador 1. Font: Elaboració pròpia

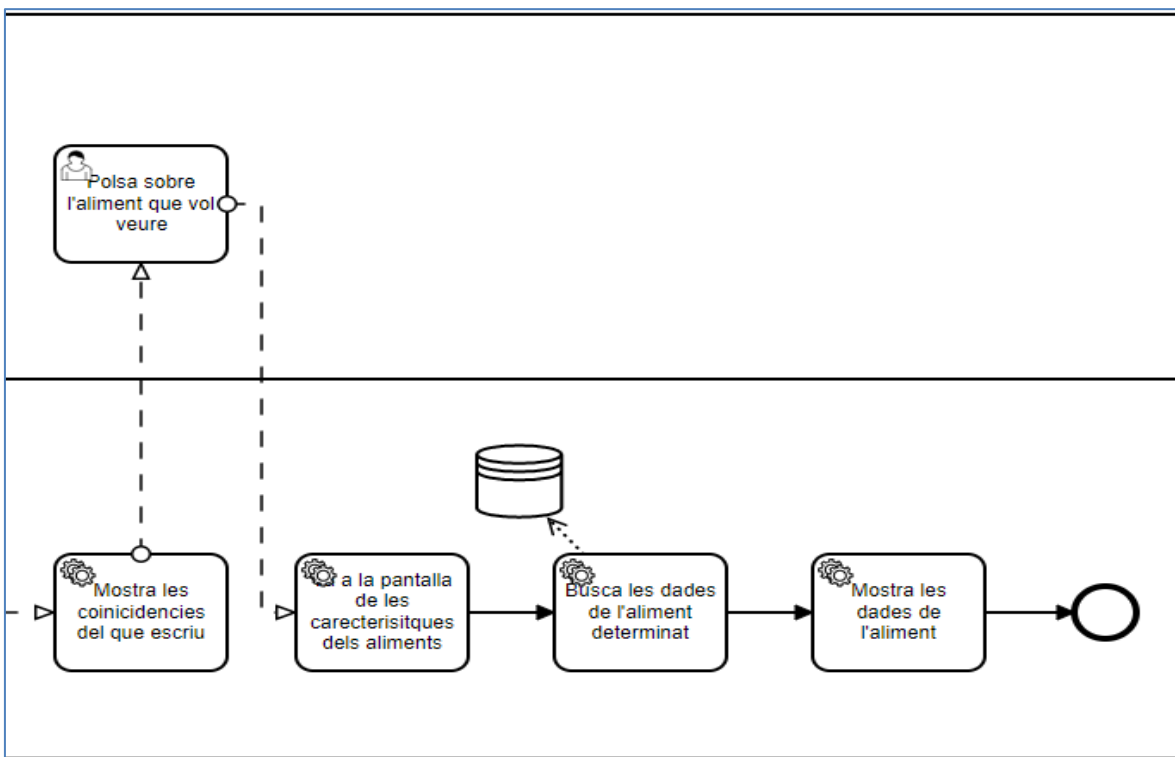


Fig. 6.2.1.4 Diagrama BPMN sobre consultar les característiques d'un aliment mitjançant el buscador 2. Font: Elaboració pròpia

**Cas N°4:**

- **Nom:** Consultar aliments d'una variant del menú
- **Descripció:** L'usuari vol consultar els aliments que es necessiten per seguir el seu menú
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina d'inici
- **Postcondicions:** L'usuari ha consultat els aliments del seu menú

**Flux Normal:**

1. L'usuari polsa el botó de consultar menú
2. El sistema mostra a l'usuari la pantalla de consulta del seu menú assignat
3. L'usuari escull entre si vol esmorzar, dinar, berenar o sopar.
4. El sistema mostra les diferents variants del moment de menjar escollit per l'usuari
5. L'usuari polsa sobre la variant que vulgui consultar
6. El sistema mostra els plats assignats de la variant
7. L'usuari polsa el botó de l'aliment
8. El sistema mostra una llista dels aliments per aquella variant

**Flux Alternatiu:**

9. L'usuari no té variants assignades
10. Una variant no té aliments assignats

### Digrama BPMN:

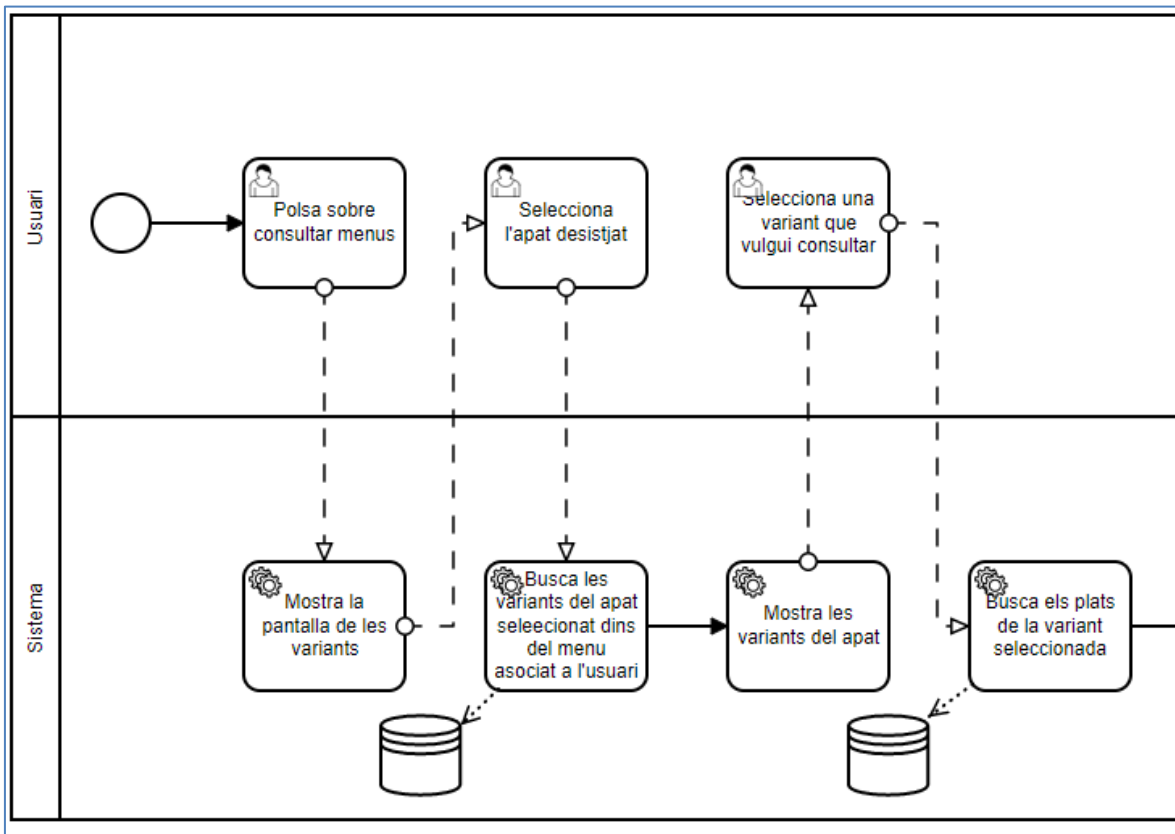


Fig. 6.2.1.5 Digrama BPMN sobre consultar aliments d'una variant del menú 1. Font: Elaboració pròpia

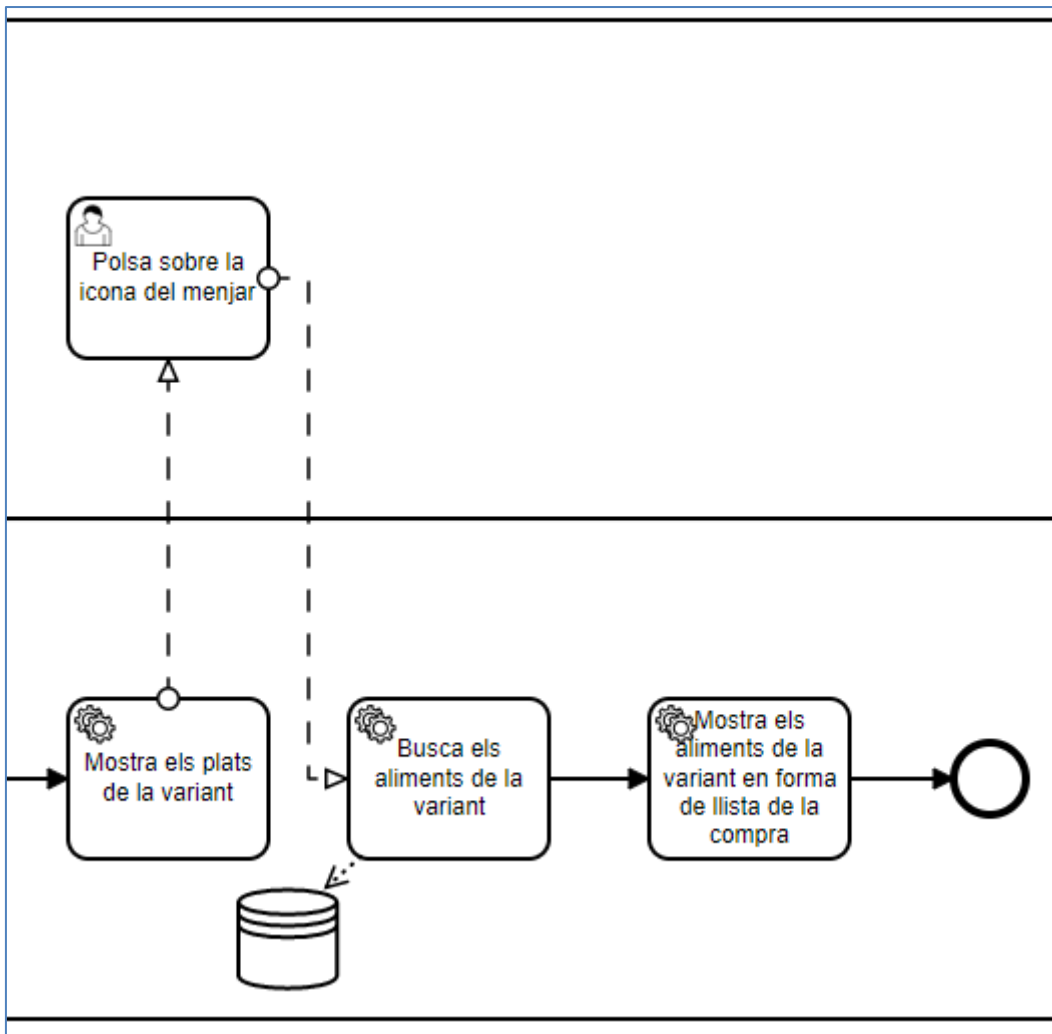


Fig. 6.2.1.6 Digrama BPMN sobre consultar aliments d'una variant del menú 2. Font: Elaboració pròpia

**Cas N°5:**

- **Nom:** Generar un menú dinàmic
- **Descripció:** L'usuari vol generar un exemple de menú dinàmic
- **Actor:** Usuari
- **Precondicions:** L'usuari es troba a la pàgina d'inici
- **Postcondicions:** L'usuari ha consultat el resultat del menú dinàmic

**Flux Normal:**

1. L'usuari polsa el botó de generar menú dinàmic
2. El sistema mostra a l'usuari la pantalla per escollir de quants dies vol el menú
3. L'usuari escull fins a quants dies vol el menú i polsa a generar
4. El sistema genera un menú amb les variants assignades del menú de l'usuari
5. L'usuari consulta per cada dia els plats que ha de menjar

**Flux Alternatiu:**

6. L'usuari no té variants assignades
7. Una variant no té aliments assignats
8. L'usuari no té menús assignats

**Diagrama BPMN:**

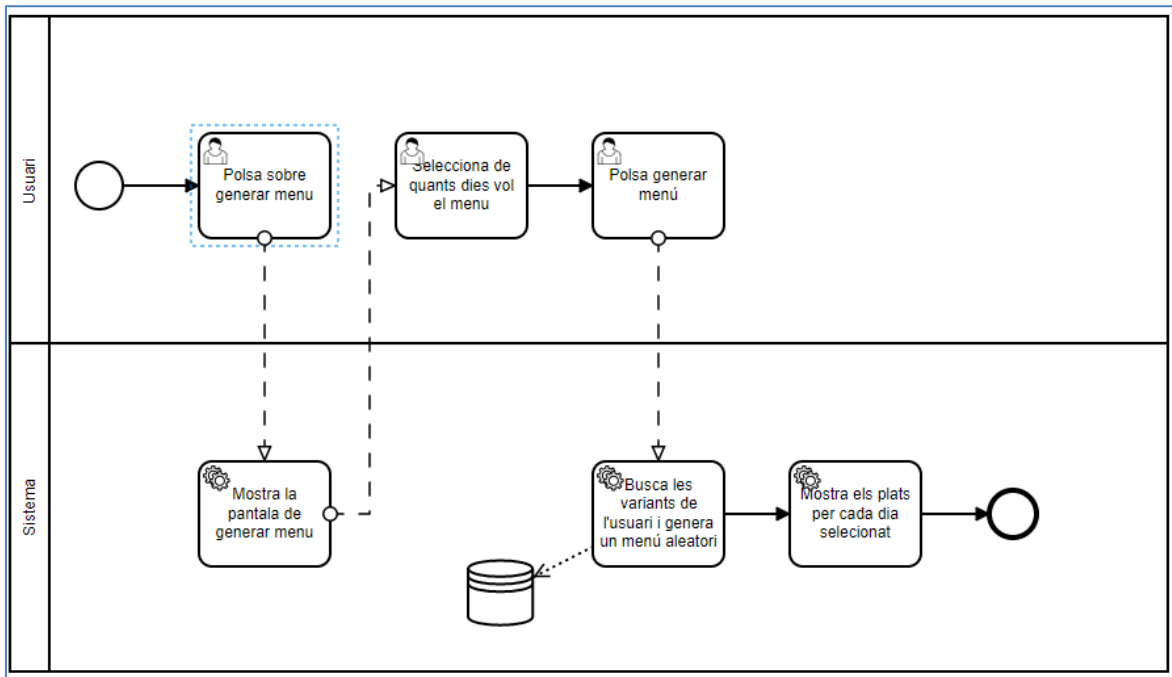


Fig. 6.2.1.7 Diagrama BPMN sobre generar un menú dinàmic. Font: Elaboració pròpia

### Disseny de les pantalles:

S'ha dissenyat mitjançant l'eina Lucidchart totes les pantalles usades a l'aplicació mòbil:

#### Pantalla d'inici de sessió:

Aquí l'usuari és capaç d'iniciar sessió amb les credencials del o la dietista i accedir al programari. Es pot veure els dissenys a la figura 6.2.1.8.



Fig. 6.2.1.8 Disseny de la pantalla d'inici de sessió. Font: Elaboració pròpia

#### Pantalla inicial:

A continuació a la figura 6.2.1.9 es pot veure la pàgina inicial de l'aplicació on es pot accedir al buscador, a les dades de l'usuari o a la generació o consulta de menús. També hi ha l'opció per poder analitzar el codi de barres d'un aliment i poder veure les seves característiques.

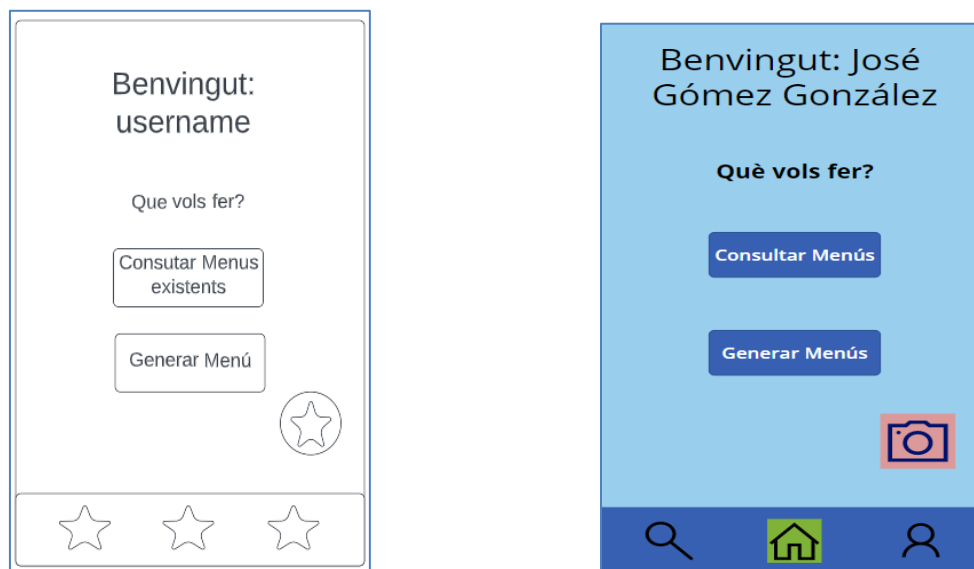


Fig. 6.2.1.9 Disseny de la pantalla inicial. Font: Elaboració pròpia



**Buscador:**

A continuació a la figura 6.2.1.10 es pot apreciar la pàgina del cercador d'aliments. Aquí es pot cercar els aliments que es vulguin consultar . Es mostra de color vermell quan l'usuari el pot menjar o vermell en cas contrari.

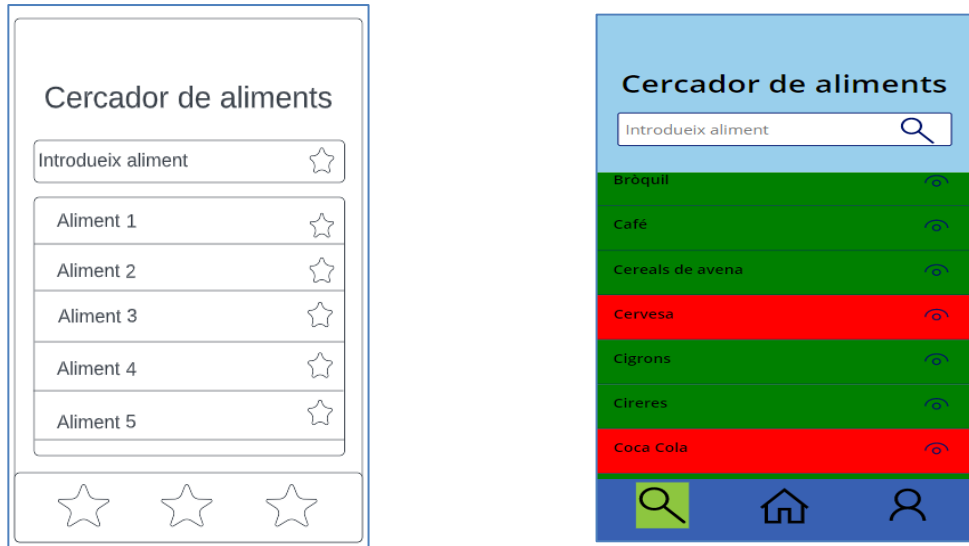


Fig. 6.2.1.10 Disseny del buscador. Font: Elaboració pròpia

**Dades personals:**

A continuació a la figura 6.2.1.11 es pot veure la pàgina amb informació de l'usuari. Aquí pot visualitzar la seva informació i editar-la en cas de desitjar-ho.

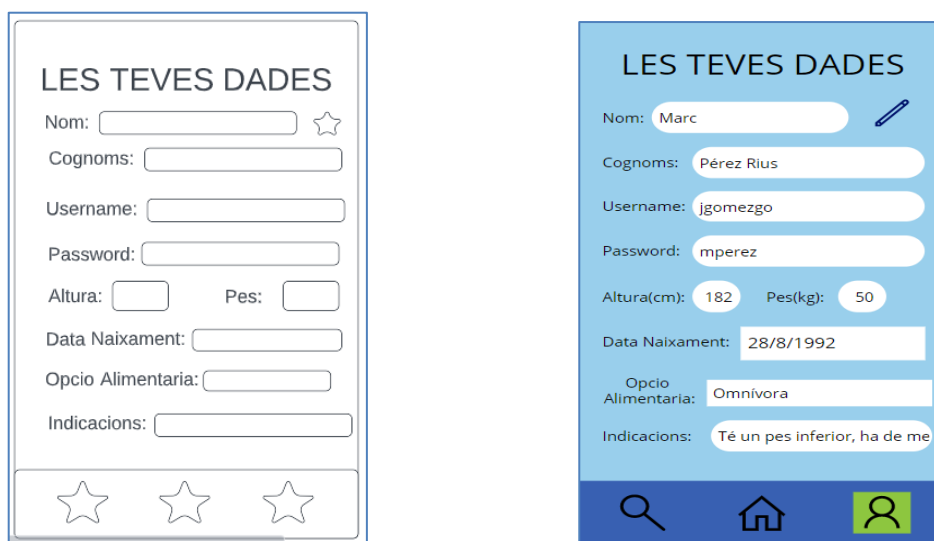


Fig. 6.2.1.11 Disseny de la pantalla de les dades personals Font: Elaboració pròpia

### Creació de menús:

A continuació a la figura 6.2.1.12 es pot visualitzar la pàgina de la creació de menús, aquí se selecciona els dies dels quals es vol el menú.

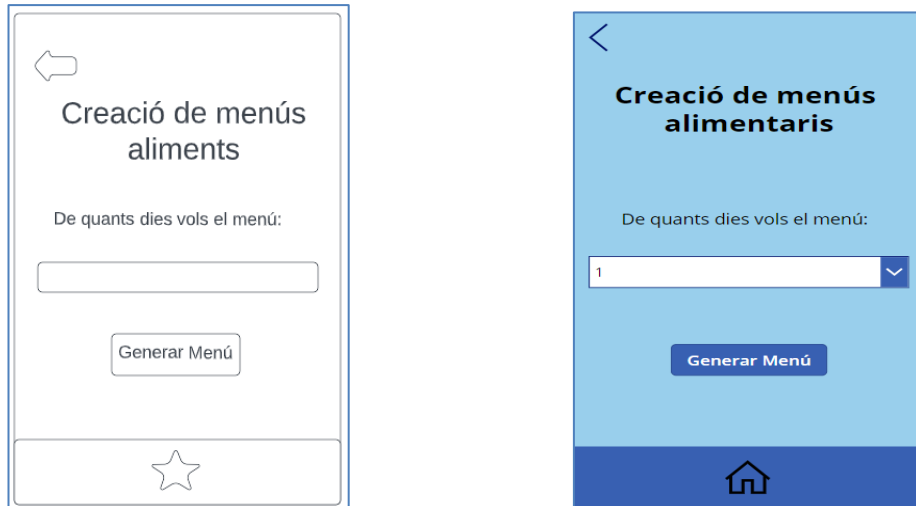


Fig. 6.2.1.12 Disseny de la pantalla de creació de menús Font: Elaboració pròpia

### Menú generat per dies:

A continuació a la figura 6.2.1.13 es pot visualitzar la pàgina del menú generat per dies amb els plats per l'esmorzar, el dinar, el berenar i el sopar.

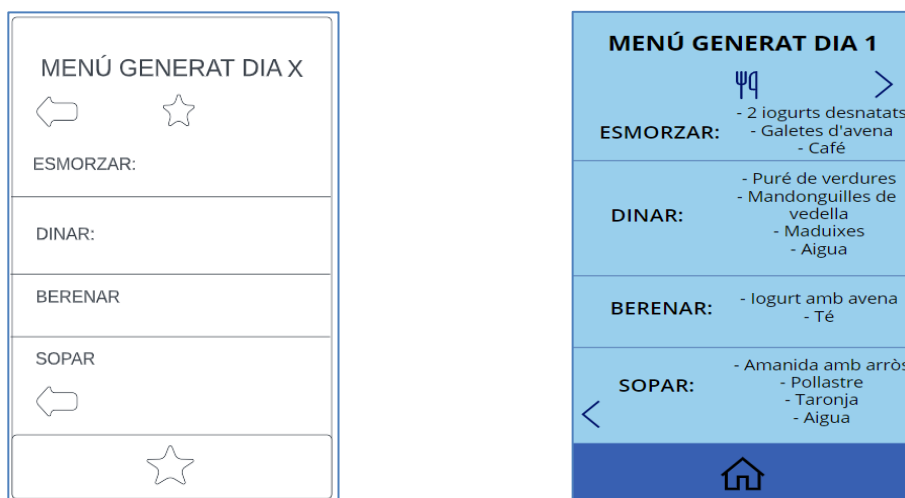


Fig. 6.2.1.13 Disseny de la pantalla del menú generat per dies. Font: Elaboració pròpia

**Variants del menú de la dietista:**

A continuació a la figura 6.2.1.14 es pot visualitzar la pàgina de les variants pel menú assignat al pacient.

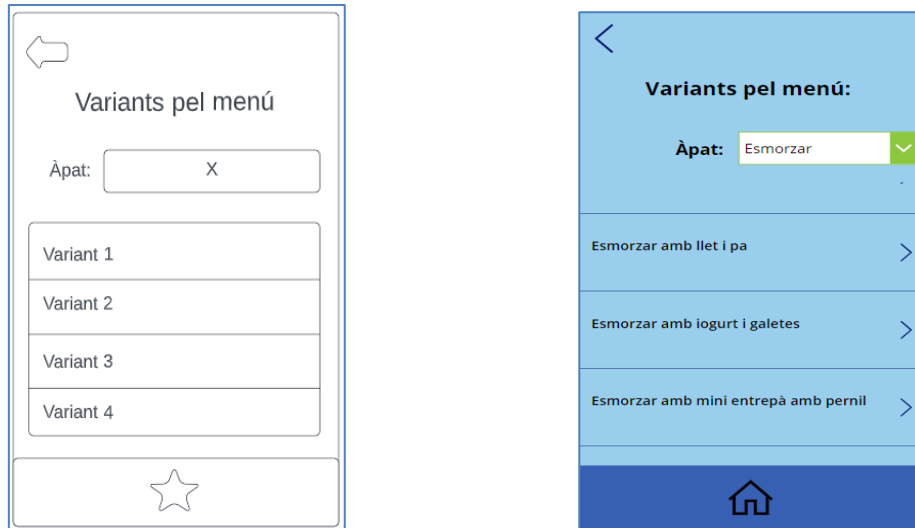


Fig. 6.2.1.14 Disseny de la pantalla de les variants de la dietista. Font: Elaboració pròpia

**Plats de la variant:**

A continuació a la figura 6.2.1.15 es pot visualitzar la pàgina dels plats assignats a la variant determinada



Fig. 6.2.1.15 Disseny de la pantalla dels plats de cada variant. Font: Elaboració pròpia

### **Llista de la compra:**

A continuació a la figura 6.2.1.16 es pot visualitzar la pàgina de la llista de la compra d'una variant



Fig. 6.2.1.16 Disseny de la pantalla de la llista de la compra. Font: Elaboració pròpia

### **Llista de la compra d'un menú:**

A continuació a la figura 6.2.1.17 es pot visualitzar la pàgina de la llista de la compra d'un menú

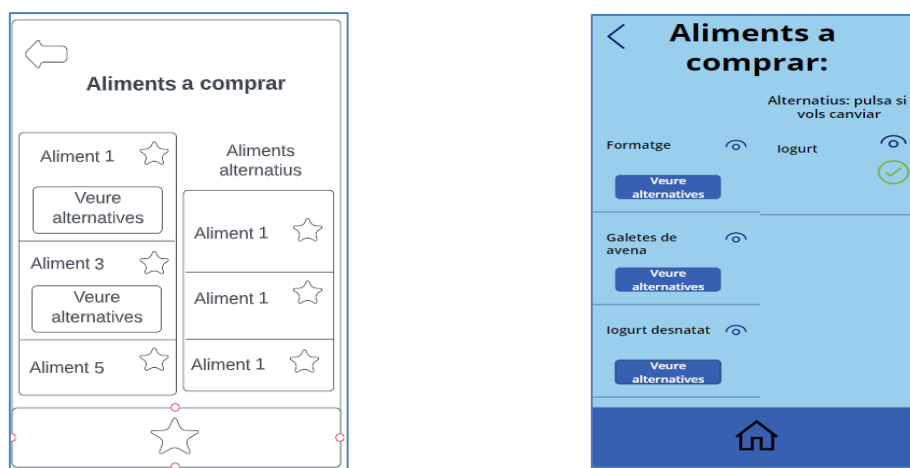


Fig. 6.2.1.17 Disseny de la pantalla de la llista de la compra d'un menú. Font: Elaboració pròpia

### Característiques d'un aliment:

A continuació a la figura 6.2.1.18 es pot visualitzar la pàgina de les característiques d'un aliment, així com la informació nutricional, les vitamines i els minerals.

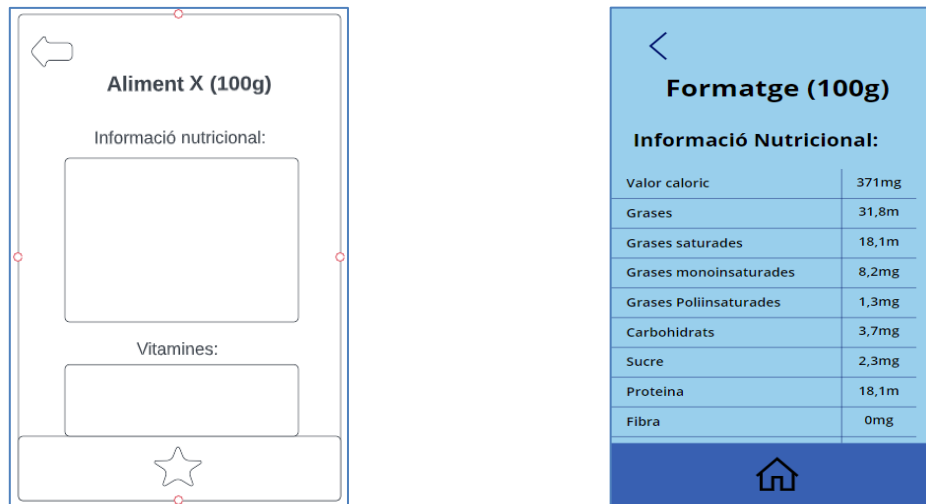


Fig. 6.2.1.18 Disseny de la pantalla de les característiques d'un aliment. Font: Elaboració pròpia

### 6.2.2 Comentaris sobre la implementació:

Per el desenvolupament de la part mòbil s'ha usat l'eina Power Apps, donat que ofereix funcionalitats que faciliten la programació d'aplicacions mòbils. Power Apps permet que les seves aplicacions funcionin tant en Android com IOS.

L'aplicació Mòbil s'ha connectat a la base de dades de MYSQL mitjançant un connector de MYSQL. Aquest permet connectar-se a la base de dades en local de MYSQL i importar les taules dins la Power App. Per poder connectar-ho correctament s'ha d'introduir una sèrie de credencials com el server i el username.

S'ha utilitzat alguns exemples de funcions com Filter, en què filtra les taules de la base de dades, Collect ,per poder guardar dades en llistes internes de la Power App o Search per buscar coincidències d'un text.

A continuació es descriuen algunes d'aquetes funcions:

#### **Funció Filter:**

Filter és una funció que filtra una taula segons les condicions que se li indiquin.

Com es pot veure a la figura 6.2.2.1, en aquest cas està filtrant les variants per l'id del menú, perquè l'usuari pugui visualitzar només les variants assignades al seu menú.

```
Filter(Filter('mydb.moment_de_menjar';id in Filter('mydb.menu_has_moment_de_menjar';Menu_id=idMenu)
.Moment_de_menjar_id);moment=Dropdown3.SelectedText.Value)
```

Fig. 6.2.2.1 Exemple funció amb Filter. Font: Elaboració pròpia

#### **Funció Collect:**

La funció Collect es una funció que permet emmagatzemar dades en taules virtuals dins de Power Apps.

Com es pot observar a la figura 6.2.2.2, en aquest cas està recorrent totes les variants filtrades pel menú de l'usuari i emmagatzema la variant dins una col·lecció interna anomenada esmorzars.

```
ForAll(Filter('mydb.moment_de_menjar';moment="Esmorzar";id in Filter('mydb.menu_has_moment_de_menjar';
Menu_id=idMenu).Moment_de_menjar_id);Collect(esmorzars;{id:id;nom:nom;primer_plat:primer_plat;
segon_plat:segon_plat;postre:postre;beguda:snack;moment:moment}));;
```

Fig. 6.2.2.2 Exemple funció amb Collect. Font: Elaboració pròpia

**Funció Search:**

La funció Search busca coincidències dins d'una galeria, una taula o un conjunt de texts i els mostra segons un paràmetre.

Com es mostra a la figura 6.2.2.3, en aquest cas mostra tots els aliments que continguin la paraula especificada en l'entrada de text.

```
Search('mydb.aliment'; TextInput3.Text; "nom")
```

Fig. 6.2.2.3 Exemple funció amb Search. Font: Elaboració pròpia

**Lectura de codis de barres:**

S'ha implementat la lectura dels codis de barres mitjançant un ítem de lectura de codis de barres a Power Apps. Aquest ítem només amb pulsar-lo ja analitza automàticament el codi mostrat i retorna el codi d'aquest. S'agafa el codi i es comprova si hi és a la base de dades, en cas afirmatiu l'aplicació mostra les característiques generals de l'aliment desitjat.

A la figura 6.2.2.4 es pot veure un exemple usant aquest ítem.

```
If(BarcodeScanner1.Value in 'mydb.codis_aliments_reals'.codi_real; Navigate(Characteristiques_aliment; Fade; {idAliment: First(Filter('mydb.codis_aliments_reals'; codi_real=BarcodeScanner1.Value)).aliment_id; nomAliment: LookUp('mydb.aliment'; id=First(Filter('mydb.codis_aliments_reals'; codi_real=BarcodeScanner1.Value)).aliment_id; nom)}))
```

Fig. 6.2.2.4 Exemple funció per llegir el codi de barres d'un aliment. Font: Elaboració pròpia





## 7. Conclusions

Totes les funcionalitats requerides s'han implementat durant tot el projecte. S'ha creat un prototip per tal de donar suport als dietistes a la seva feina així com per ajudar als seus pacients a decidir que poden o no poden menjar o a tenir un coneixement més ampli sobre el que implica seguir una dieta mèdica i estar informats sobre el que han de menjar. Per tal de complir els objectius s'ha creat una plataforma web perquè el o la dietista pugui administrar tots els continguts que necessiti i una aplicació mòbil pels pacients en què els poden ajudar a saber què menjar i què no menjar. També pot escanejar codis de barres per poder saber les característiques genèriques de l'aliment desitjat.

Un cop finalitzat el projecte es conclou que el projecte és viable econòmicament i tecnològicament. No s'ha necessitat un gran pressupost i s'ha fet amb eines tecnològiques que es tenien a l'abast.

Seguint la planificació inicial del projecte, s'ha executat el projecte amb certa demora respecte al que estava previst, però s'ha aconseguit de continuar endavant sense dificultats. La principal demora ha sigut durant l'etapa de desenvolupament de la plataforma web, ja que s'havia de programar molta quantitat de menús i funcionalitats per cada ítem. Els manuals d'ús de l'aplicació per usuaris no ha donat temps a fer-se.

Aquest projecte ha estat un projecte llarg i de molta complexitat que ha requerit dur a terme un ampli estudi del software necessari per poder dur-lo a terme.

En l'àmbit personal, en finalitzar el projecte s'ha millorat els coneixements de frontend , backend i programació d'aplicacions per tal d'obtenir a complir aquest projecte.

El projecte ha sigut desenvolupat perquè en un futur s'hi puguin afegir només funcionalitats si s'escau.



## **8. Ampliacions**

Un cop el projecte ha finalitzat, es plantegen algunes ampliacions de cara al futur.

A continuació es detallen algunes de les ampliacions plantejades:

- Afegir nous idiomes a part del català
- Implementar seguretat al backend i al frontend
- Afegir més dietes amb més menús de les existents
- Ampliar el nombre de codis de barres registrats
- Posar les dades de la base de dades en un servidor



## 9. Bibliografia

- [1] Tabla de alimentos para diabéticos [en línea] [consulta: 7 de gener de 2022].  
Disponible a <https://www.tuasaude.com/es/que-puede-comer-un-diabetico/>
- [2] DIETA HIPOLIPÍDICA Dieta Baixa en colesterol i greixos saturats [en línea] [consulta: 7 de gener de 2022].  
Disponible a [https://www.institutovascularfuentes.com/dieta\\_hipolidica.pdf](https://www.institutovascularfuentes.com/dieta_hipolidica.pdf)
- [3] Colesterol: Los principales alimentos que mejorarán tus niveles diabéticos [en línea] [consulta: 7 de gener de 2022].  
Disponible a <https://www.mayoclinic.org/es-es/diseases-conditions/high-blood-cholesterol/in-depth/cholesterol/art-20045192>
- [4] Yazio [en línea] [consulta: 3 de gener de 2022].  
Disponible a <https://www.yazio.com/es/>
- [5] Yuka [en línea] [consulta: 3 de gener de 2022].  
Disponible a <https://yuka.io/es/>
- [6] MyRealFood [en línea] [consulta: 3 de gener de 2022].  
Disponible a <https://myrealfood.app/>
- [7] MyFitnessPal [en línea] [consulta: 3 de gener de 2022].  
Disponible a <https://www.myfitnesspal.com/>
- [8] HealtyifyMe [en línea] [consulta: 3 de gener de 2022].  
Disponible a <https://www.healthifyme.com/in/>
- [9] Llei Orgànica 3/2018, de 5 de desembre [en línea] [consulta: 16 de maig de 2022].  
Disponible a [https://www.boe.es/boe\\_catalan/dias/2018/12/06/pdfs/BOE-A-2018-16673-C.pdf](https://www.boe.es/boe_catalan/dias/2018/12/06/pdfs/BOE-A-2018-16673-C.pdf)
- [10] Base de dades d'aliments de Yazio [en línea] [consulta: 3 de març de 2022].  
Disponible a <https://www.yazio.com/es/alimentos>
- [11] Menús hipocalòrics [en línea] [consulta: 20 de maig de 2022].  
Disponible a <https://www.fisterra.com/m/ficha.asp?idFicha=2975>
- [12] Menús hipercalòrics [en línea] [consulta: 20 de maig de 2022].  
Disponible a <https://www.fisterra.com/m/ficha.asp?idFicha=2985>

[13] Lucidchart [en línia] [consulta: 3 de març de 2022].

Disponible a <https://www.lucidchart.com/pages/>

[14] BPMN.io [en línia] [consulta: 3 de març de 2022].

Disponible a <https://bpmn.io/>

[15] Capas, cebollas y colmenas: arquitecturas en el backend [en línia] [consulta: 3 de març de 2022].

Disponible a <https://www.adictosaltrabajo.com/2019/07/02/capas-cebollas-y-colmenas-arquitecturas-en-el-backend/>