



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

**Web per consultar la compatibilitat dels
principis actius dels medicaments amb
la capacitat de filtratge del ronyó**

Xavier Robert Finestra

Director: Laura Igual Muñoz
Realitzat a: Departament de Matemàtica
Aplicada i Anàlisi. UB

Barcelona, 27 de juny de 2015

Abstract

Nowadays the usage of web applications in order to manage and look up information is on the rise. These applications are known as data-driven applications because they use information as the base of their operation. They are mainly used by companies to make their workers jobs easier, thus minimizing the time wasted looking for information.

During this project, an application programming interface (API) and a web application that will use said API will be implemented. The information that will be shown is from a table that was created as a result of a study done by the faculty of pharmacy of the University of Barcelona. This study was done with the collaboration of some doctors specialized in renal illnesses. The aims of this study were to determine the prevalence of dosing inadequacy as a consequence of renal impairment in patients over 65 that were taking 3 or more drug products and who were being attended in community pharmacies, and to evaluate the effectiveness of the community pharmacist intervention in addressing the problem of dosing inadequacy as a consequence of renal impairment in patients over 65 years that were taking 3 or more drugs when compared with usual care.

This web application will allow doctors and pharmacists to access this table with ease.

This document explains how the project has been developed using the framework Spring MVC, Java, AngularJS and JPA with Hibernate. It will detail its different parts: study of the possibilities, the configuration of the different technologies, the web and API development process and all the tests that have been made with the professors from the faculty of pharmacy of the University of Barcelona.

The goal was to work with technologies, architectures and techniques used in the world of work in order to create a quality product as well as studying said technologies.

Resum

Actualment, esta en augment l'ús d'aplicacions web per tal de resoldre les necessitats de consulta i tractament de dades. Aquesta aplicacions, conegudes com a *data-driven applications*, tracten les dades com a la base del seu funcionament. Són utilitzades principalment per empreses per tal de facilitar la feina dels seus treballadors i així minimitzar el temps gastat en consultar informació.

En aquest projecte s'implementarà una application programming interface (API) i una aplicació web que utilitzarà aquesta API per a mostrar informació. La informació que es mostrarà és una taula que es va crear a partir d'un estudi de la Facultat de Farmàcia de la Universitat de Barcelona, juntament amb diversos metges especialistes en malalties renals. Es va realitzar per determinar si gent gran amb problemes de ronyó estaven rebent medicines amb principis actius que els hi podien causar efectes secundaris. L'aplicació web permetrà a metges i farmacèutics consultar amb facilitat si els principis actius dels medicaments que estan prenent els pacients poden causar problemes.

Aquesta memòria explica com s'ha desenvolupat el projecte, fent servir el framework Spring MVC, Java, AngularJS i JPA amb Hibernate. Detallant el procés d'estudi previ realitzat, en el que es comparen les múltiples tecnologies que es podrien haver utilitzat. La preparació prèvia i la configuració del projecte que s'ha realitzat abans d'iniciar a crear el codi del projecte. La implementació del web i la API, que s'ha fet utilitzant totes les tecnologies prèviament mencionades. I finalment les proves realitzades amb els professors de la facultat de Farmàcia de la Universitat de Barcelona.

S'ha intentat treballar amb tecnologies, arquitectures i tècniques utilitzades en el món empresarial per tal de crear un producte de qualitat, aprofitant la feina per estudiar el seu funcionament.

Índex

1	Introducció.....	3
1.1	Context i motivació.....	3
1.2	Objectius.....	4
1.3	Estructura de la memòria.....	4
2	Estudi.....	6
2.1	Avaluació tecnològica.....	6
2.2	Llenguatge de programació i tecnologies utilitzades en el desenvolupament del projecte.....	8
2.2.1	Java.....	8
2.2.2	Spring Model-View-Controller (Spring MVC).....	8
2.2.3	JavaScript (AngularJS).....	9
2.2.4	Bootstrap.....	15
2.2.5	Gestió i creació del projecte.....	15
2.3	Sistema gestor de bases de dades.....	17
2.4	Servidor web.....	18
2.5	Diagrama de Gantt.....	19
2.6	Estudi econòmic.....	22
3	Anàlisis.....	23
3.1	Especificacions del problema per part del client.....	23
3.1.1	Web.....	23
3.1.2	Mòdul d'administració.....	24
3.1.3	API.....	24
3.2	Requisits d'usuari.....	24
3.3	Requisits del sistema.....	25
3.4	Diagrama de casos d'ús.....	26
3.4.1	Casos d'ús textuals.....	27
4	Desenvolupament.....	35
4.1	Arquitectura general del projecte.....	35
4.2	Relació de les classes.....	40
5	Proves i resultats.....	42
5.1	Proves realitzades.....	42

5.2 Descripció de l'aspecte de l'aplicació.....	44
5.3 Feedback del client.....	45
5.4 Concordança de resultats i objectius.....	46
6. Conclusions i treball futur.....	47
6.1 Conclusions.....	47
6.2 Possibles millores de l'aplicació.....	47
Bibliografia.....	49
Annexos.....	50
Annex A: Manual d'usuari.....	50
1. Pantalla Inicial.....	50
1.1. Inici de sessió.....	50
1.2. Crear compte.....	51
2. Consultes.....	53
3. Administrar compte d'usuari.....	55
Annex B: Manual d'administrador.....	57
4. Administració.....	57
4.1 Modificar la base de dades.....	57
4.2 Modificar el llistat de correus electrònics.....	58
4.3 Modificar el llistat de correus electrònics.....	58
4.4 Suggestiments.....	59
Annex C: Guia de funcionament de l'API.....	60
AccountController.....	60
MessageController.....	62
EmailDomainController.....	63
ActiveIngredientController.....	64
ActiveIngredientRecommendationController.....	67
Annex D: Guia per a futurs desenvolupadors.....	68

1 Introducció

1.1 Context i motivació

Aquest projecte va sorgir com una proposta de la facultat de farmàcia de la Universitat de Barcelona. La base del qual és una taula que es va crear a partir d'un estudi de la Facultat de Farmàcia de la UB [\[1\]](#), juntament amb diversos metges especialistes en malalties renals. Es va realitzar per determinar si gent gran amb problemes de ronyó estaven rebent medicines amb principis actius que els hi podien causar efectes secundaris. La taula indica una sèrie de recomanacions a tenir en compte, segons una capacitat de filtrat i un principi actiu.

El resultat de l'estudi va ser que un 17,5% dels casos revisats estaven rebent medicaments o dosis inadequades. Per tal de intentar evitar els casos es va crear una taula on es podia consultar en funció de la capacitat de filtratge del pacient i el principi actiu, si aquest li podia causar efectes secundaris. Aquest treball es construeix sobre aquesta taula, permetent a metges i farmacèutics consultar-la de forma telemàtica.

El ronyó és un òrgan encarregat de filtrar la sang de l'aparell circulatori. En particular, el ronyó elimina les deixalles (diversos residus metabòlics de l'organisme) que altrament podrien ser tòxics per a l'organisme. La capacitat de filtratge del ronyó es pot mesurar de forma senzilla a partir d'una anàlisi de sang.

A mesura que augmenta l'edat d'una persona, és habitual que aquest prengui més medicaments. A més, a mesura que una persona es fa gran, la capacitat de filtratge del ronyó acostuma a disminuir. El principi actiu d'un medicament pot o no acumular-se a la sang depenent de la capacitat de filtratge del ronyó. En cas que el ronyó no tingui capacitat d'eliminar el principi actiu, aquest pot acabar essent tòxic per al malalt.

1.2 Objectius

L'objectiu d'aquest TFG és dissenyar un web en què els metges i farmacèutics puguin consultar la compatibilitat de la capacitat de filtratge del ronyó amb els principis actius que el malalt pren tant des d'un ordinador com de d'un dispositiu mòbil. Aquest web no ha de ser d'accés públic, sinó que només hi han de poder accedir els metges col·legiats.

Com que el llistat en el que està treballant la Facultat de Farmàcia de la UB s'anirà modificant i actualitzant, també és necessari afegir al web un apartat d'administració que permeti a la gent de la Facultat de Farmàcia de la UB controlar el contingut del llistat de principis actius.

Tenint en ment que és possible que en un futur es desenvolupin aplicacions per a Android i iOS, s'ha volgut unificar la font d'informació de les tres aplicacions (web, Android i iOS) i es desenvoluparà una API (Application Program Interface) que gestionarà l'accés a la informació que hi haurà a la base de dades, evitant així haver de mantenir una base de dades diferent per a cada aplicació.

En concret, els objectius d'aquest projecte són:

- Crear un web d'accés restringit on metges i farmacèutics puguin consultar un llistat de principis actius i els problemes que poden ocasionar amb diferents capacitats de filtratge del ronyó.
- Crear un mòdul d'administració per al web i la base de dades que contindrà tota la informació.
- Crear una API que permeti a altres aplicacions consultar aquest llistat de principis actius i recomanacions.

1.3 Estructura de la memòria

Aquesta memòria esta dividida en els següents capítols:

1. Introducció

En el capítol 1 es presenta la idea i el concepte que han generat la creació d'aquest projecte.

2. Estudi

En el capítol 2 s'expliquen les eines i tecnologies que s'han fet servir per a dur a terme el projecte, el perquè de la seva elecció i com s'ha estructurat la feina.

3. Anàlisis

En el capítol 3 s'explica el comportament de l'aplicació en funció del tipus d'usuari.

4. Desenvolupament

En el capítol 4 s'explica el disseny del codi, l'estructura del projecte i de la base de dades i com es relaciona el web amb la API.

5. Proves i resultats

En el capítol 5 s'expliquen les proves realitzades per a assegurar el correcte funcionament del projecte i el perquè de l'aspecte de l'aplicació.

6. Conclusions i treball futur

En el capítol 6 conté les conclusions extretes del projecte així com possibles millores que es poden fer a l'aplicació.

7. Bibliografia

Recull de textos i documents que s'han utilitat mentre es treballava en aquest projecte.

8. Annex

En l'annex hi ha tres manuals:

- Manual d'usuari, tant per a usuaris normals com per a administradors.
- Manual de funcionament de la API, per a desenvolupadors que necessitin extreure la informació de la base de dades.
- Manual per a futurs desenvolupadors que explica les parts tècniques del projecte i com continuar desenvolupant-lo.

2 Estudi

Per a dur a terme aquest projecte s'ha fet servir utilitzar tecnologies que s'estan utilitzant actualment en el desenvolupament d'aplicacions web en el món empresarial, per dos motius. Per una banda, crear una aplicació funcional i robusta i, per l'altre, aprendre a fer funcionar totes aquestes tecnologies.

L'aplicació que s'ha desenvolupat és un estil d'aplicació web coneguda com a data-driven single-page application (data-driven SPA). Aquest estil d'aplicació es caracteritza per carregar en la màquina de l'usuari tots els recursos HTML, JavaScript i CSS que necessita la pàgina web quan entrem per primera vegada al web per tal de donar una navegació més fluida a l'usuari. Això és possible ja que només s'ha de connectar amb el servidor per a aconseguir la informació, mentre que el formateig de la mateixa es fa segons els fitxers que ja estan descarregats, reduint així el temps d'espera. El fet de decidir crear una aplicació d'aquest estil ha influenciat molt les decisions preses a l'hora d'escollir les tecnologies utilitzades.

2.1 Avaluació tecnològica

El projecte està molt clarament dividit en tres parts tal i com es pot veure a la Figura 2.1. Per un cantó tenim la part que s'encarregarà de tractar directament amb la base de dades i contindrà tota la lògica que s'executarà (servidor). Per l'altre, tenim la part que s'executarà a la màquina de l'usuari i amb la que ell interactuarà (client). Finalment, tenim la base de dades, que és l'encarregada de guardar tota la informació (base de dades).

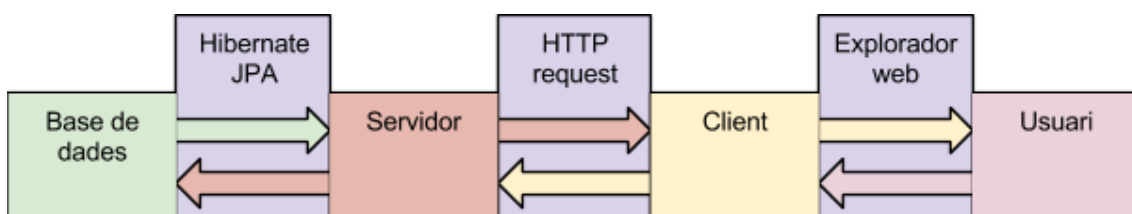


Figura 2.1 Relacions entre les diferents parts de l'aplicació i l'usuari

Aquestes són les tecnologies que s'han escollit per a realitzar el projecte i que seran explicades i justificades en l'apartat següent:

- **Servidor:**
El codi del servidor esta escrit amb el llenguatge de programació **JAVA**[\[6\]](#), utilitzant el framework **Spring MVC**[\[7\]](#). Per a treballar amb la base de dades s'ha utilitzat la implementació de la Java Persistence API (JPA) feta amb **Hibernate**[\[8\]](#). Tot això esta gestionat per un **Apache Tomcat**[\[10\]](#).
- **Client web:**
El codi del client web esta desenvolupat amb **AngularJS**[\[2\]](#), un framework de JavaScript. El *look and feel* de la web esta s'ha creat utilitzant els fitxers d'estil (CSS) de **Bootstrap**[\[11\]](#).
- **Base de dades:**
Per a la base de dades s'ha escollit **MySQL**.
- **Gestió:**
Per a desenvolupar el projecte s'ha utilitzat el **Netbeans IDE**[\[5\]](#) (Integrated Development Environment), amb la ajuda del qual s'ha desenvolupat tot el codi del projecte.
La compilació dels fitxers js s'ha fet amb **Grunt**[\[12\]](#), un *taskrunner* de JavaScript.
Per a gestionar les dependències del servidor s'ha utilitzat **Maven**, una eina de gestió de dependències per a projectes JAVA inclosa a Netbeans.
Per a les dependències del client web s'ha utilitzat **NPM**[\[4\]](#), un gestor de paquets de **NodeJS**[\[3\]](#).

Llistat de versions:

Gestió

Maven: 2.2
Grunt: 0.4.1
Npm: 2.11.3

Base de dades:

MySQL: 5.6.21

Server:

Java: JDK 1.8.
Spring MVC: 4.0.5
Hibernate: 4.3.1.Final

Client:

AngularJS: 1.3.15
Bootstrap: 3.1.1

Eina

Netbeans IDE: 8.0.2

2.2 Llenguatge de programació i tecnologies utilitzades en el desenvolupament del projecte

2.2.1 Java

La decisió que s'ha pres al escollir Java com a llenguatge de programació ha estat fortament basada en el fet de que Java és un llenguatge orientat a objectes, amb moltes llibreries i dependències gratuïtes que permeten agilitzar molt la feina.

Un altre factor que va decantar la decisió a favor de Java va ser l'existència del framework Spring MVC.

2.2.2 Spring Model-View-Controller (Spring MVC)

Aquest és un framework en Java per a aplicacions web basat en peticions, les quals gestiona mitjançant un servlet (petit programa java que s'executa en un servidor web i que s'encarrega de gestionar les peticions i les respostes d'altres clients o servidors).

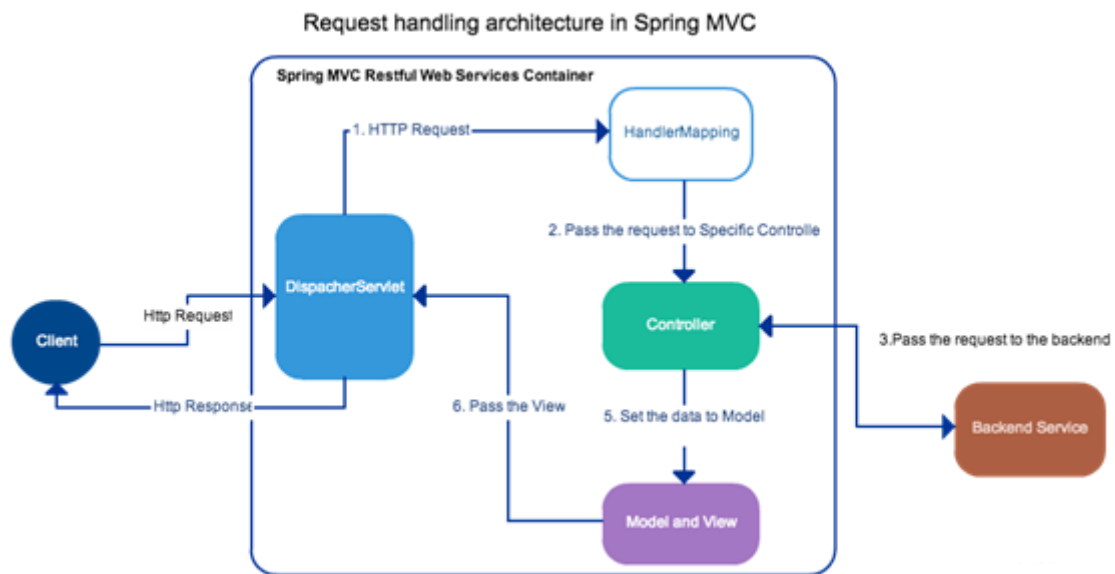


Figura 2.2.2: Arquitectura de Spring MVC

El funcionament del framework es pot veure a la figura 2.2.2. En el cas d'aquest projecte el client és una web feta amb AngularJS, la qual envia peticions al servidor que utilitza Spring MVC per a gestionar-les. Un cop una petició arriba al framework, aquesta entra al servlet, el qual la passa al HandlerMapping, el qual s'encarrega de veure a quin controlador ha de passar la petició segons la URL i els paràmetres de la HTTP request. En cas de no trobar el controlador adient o haver-hi algun error en el format de la petició, es retornarà una resposta amb un codi d'error.

Un cop ha arribat la petició al controlador, aquest s'encarrega d'executar la lògica que s'ha programat per a aquesta petició concreta. Es pot tractar la informació que ha arribat amb la petició o connectar-se al backend i extreure i/o afegir informació a la base de dades. Un cop la informació ha estat extreta es prepara el model que es retornarà al client i s'envia aquest al DispatcherServlet. El servlet s'encarrega de retornar el model amb la informació que ha preparat el controlador.

En el cas del nostre projecte tota la informació que no està a les capçaleres de la petició estarà formatada com a JavaScript Object Notation (JSON). JSON és un format lleuger d'intercanvi de dades simple d'interpretar i generar per part de les màquines. El fet d'enviar i rebre tota la informació amb aquest format facilita molt la feina tant d'enviament com de recepció, ja que Spring MVC pot transformar aquestes dades en classes de java amb facilitat i AngularJS pot convertir-les fàcilment en variables, fent de JSON un format molt pràctic a l'hora de transmetre informació.

2.2.3 JavaScript (AngularJS)

Per a crear el client web que utilitzaran els usuaris per a consultar el llistat de principis actius s'ha decidit treballar amb JavaScript, un llenguatge script orientat a objectes que s'utilitza per a crear efectes interactius dintre de webs. Al ser un llenguatge orientat a objectes com Java, l'enviament d'informació mitjançant JSON és molt simple, ja que JavaScript envia la informació dels seus objectes al servidor, on Spring MVC la rep i la converteix en classes Java. Aquesta facilitat a l'hora de transmetre informació va ser un dels principals motius pels quals es va decidir utilitzar aquests dos llenguatges.

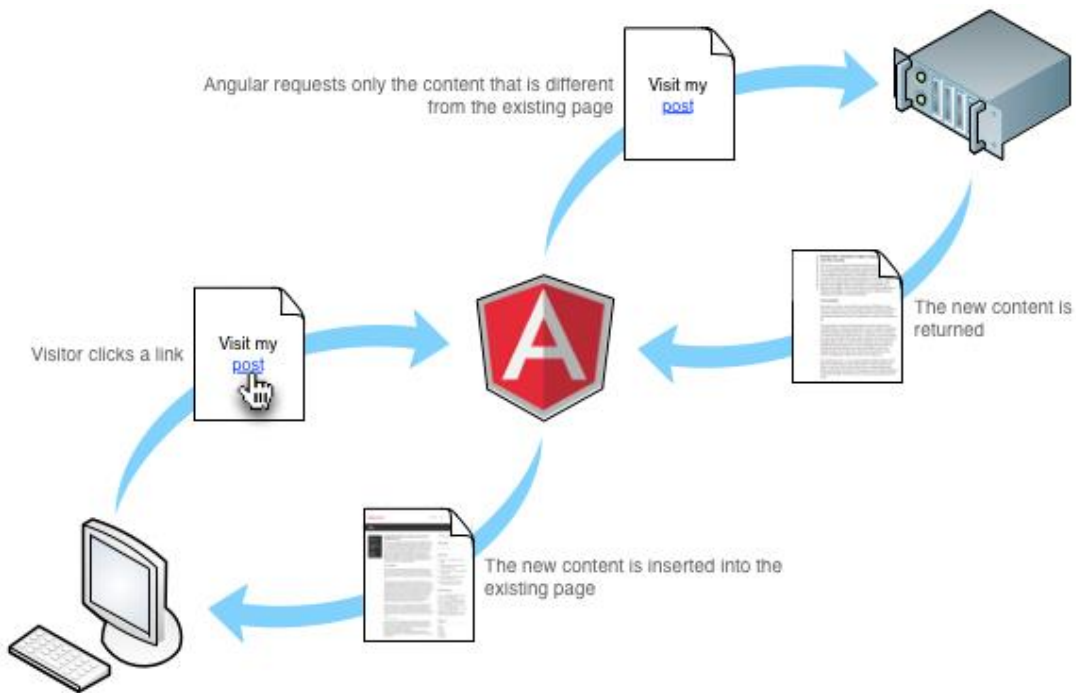


Figura 2.2.3.1: Diagrama de flux del client

Un cop presa la decisió de treballar amb JavaScript va sorgir el dubte de si es volia treballar directament amb JavaScript o es volia utilitzar alguna mena de llibreria o framework. Es va decidir optar per no utilitzar directament JavaScript ja que els frameworks i llibreries existents ofereixen moltes facilitats a l'hora de gestionar la pàgina web.



Figura 2.2.3.2: Logos d'AngularJS, Backbone i ember.js respectivament

Les diferents opcions que es van estudiar a l'hora de decidir com es treballaria el JavaScript de la pàgina web han estat (Figura 2.2.3.2):

- AngularJS
- Ember.js
- Backbone

Característica	AngularJS	Ember.js	Backbone
Mida	39.5 kb	90 kb	6.5 kb
Tipus	Framework	Framework	Llibreria
Llibertat	Mitja	Baixa	Alta
Templating	Si	Si	No
Dificultat inicial	Alta	Molt alta	Mitja
Two way data-binding inclòs per defecte	Si	No	No
Codi js en els fitxers HTML	Si	Si	No
Comunitat	Molt activa	Activa	Activa

Taula 2.2.3.3: Característiques que s'han estudiat per escollir amb quina opció treballar

A la taula 2.2.3.3 es poden veure les principals característiques que es van tenir en compte a l'hora d'escollir.

- **Mida**

Tots tres tenen una mida bastant reduïda, encara que es pot veure clarament que Backbone és el més lleuger dels tres. Això és bo per una part, ja que fa que les peticions al servidor siguin molt més lleugeres, però per l'altre és un indicador de que Backbone ens aporta menys funcionalitats que els altres dos, fent-nos fer més feina per posar en marxa la pàgina web. AngularJS i Ember.js són més pesats, però ens donen molta més part de la feina feta.

- **Tipus**

Mentre que AngularJS i Ember.js són frameworks, Backbone és una llibreria. Això implica que Backbone és més fàcil d'afegir en un lloc concret d'una aplicació web sense necessitat de modificar gaire codi. Com que el que es buscava era una eina per a crear tota la web, aquest punt no s'ha tingut en compte.

- **Llibertat**

- **AngularJS:**

Aquest framework té un funcionament i una estructura molt concreta que no deixa lloc a molta llibertat si es vol crear tot el projecte utilitzant-lo. És possible afegir AngularJS a només una part o pàgina de la web, cosa que ens donaria més llibertat, però això no és el que es vol fer en aquest projecte, per tant es considera que el nivell de llibertat no és molt alt.

- **Backbone:**

Al ser una llibreria molt lleugera que principalment ens facilita les eines per a que nosaltres mateixos gestionem tota la web de la

manera que creiem més convenient, Backbone ens dona un nivell de llibertat altíssim en comparació a les altres dues.

- **Ember.js:**

Ember.js té una estructura molt concreta que no ens permet gairebé gens llibertat a l'hora de prendre decisions de disseny i que obliga al desenvolupador a seguir uns patrons fortament marcats.

- **Templating**

Tant AngularJS com Ember.js ofereixen solucions incorporades per al templating (creació de plantilles) de fitxers HTML, cosa ajuda a evitar la duplicitat de codi, al permetre utilitzar la mateixa plantilla amb diferents controladors (codi js que permet controlar el template). Backbone no té aquesta facilitat incorporada per defecte, s'ha de desenvolupar.

- **Dificultat inicial**

Tant AngularJS com Ember.js tenen una dificultat inicial més alta que Backbone, però un cop assolit un nivell més alt de coneixement sobre l'eina Backbone segueix mantenint la mateixa dificultat ja s'han de desenvolupar moltes facilitats que en els altres casos estan incloses. La representació gràfica d'aquesta dificultat es pot observar a la figura 2.2.3.4.

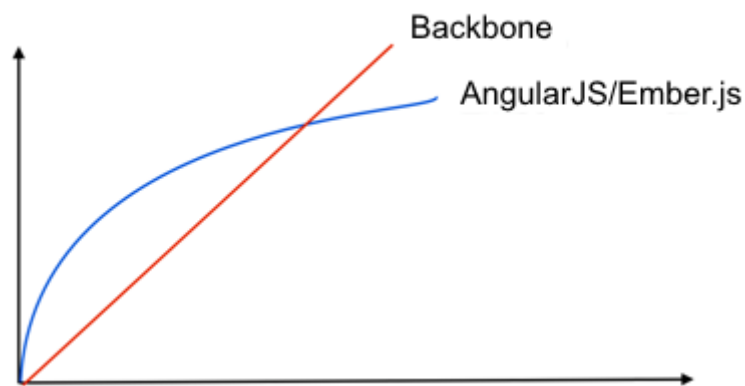


Figura 2.2.3.4: Corbes d'aprenentatge de Backbone, Ember.js i AngularJS.

- **Enllaçament bidireccional**

AngularJS funciona amb enllaçament de dades bidireccional, mentre que Backbone i Ember.js funcionen amb un enllaçament unidireccional.

L'enllaçament de bidireccional (figura 2.2.3.6) fa que en tot moment la informació que veu l'usuari (vista) i la informació que té el client web (model) sigui la mateixa, sense necessitat de refrescar la pàgina ni fer cap connexió addicional al servidor. Per altre banda, l'enllaçament unidireccional fa que sigui el desenvolupador l'encarregat de mantenir la sincronia entre les dades del model i de la vista. Pel projecte actual l'enllaçament bidireccional ens és útil ja que ens estalvia feina, evitant-nos haver de gestionar la sincronia de les dades.

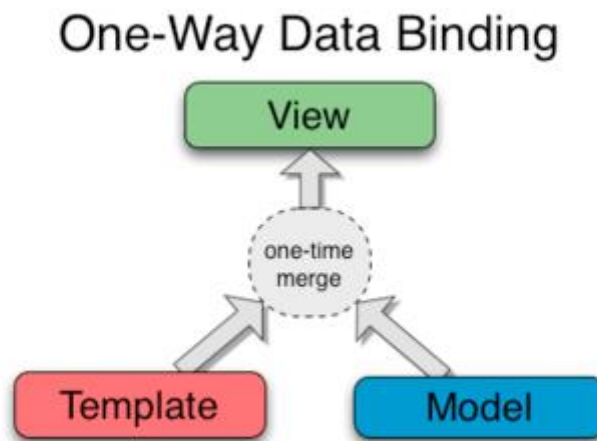


Figura 2.2.3.5: Funcionament de l'enllaçament de dades unidireccional

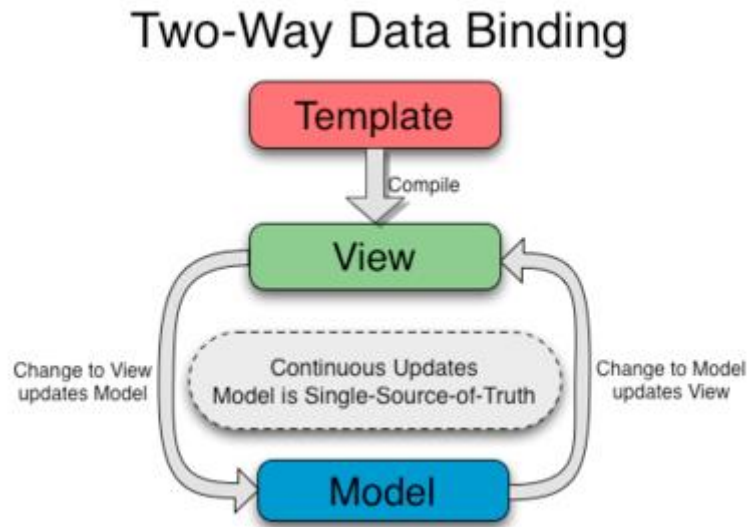


Figura 2.2.3.6: Funcionament de l'enllaçament de dades bidireccional

- **Codi js en les plantilles HTML**

Ember.js i AngularJS utilitzen codi js en les plantilles HTML, el que fa que aquestes no es pugui utilitzar sense tenir el framework carregat a tota la web. En canvi, Backbone no té necessitat d'utilitzar codi js a les plantilles, cosa que permet la seva reutilització sense cap mena de problema. Com que tot el projecte estarà fet amb la mateixa eina, aquesta limitació d'AngularJS i Ember.js no ens afecta, fent que aquest punt no sigui decisiu en l'elecció.

- **Comunitat**

Com es pot veure clarament a la figura 2.2.3.7, AngularJS és el framework que s'està utilitzant més i té una comunitat més activa. Això ens indica que trobarem més possibles solucions per Internet quan ens trobem amb algun problema. També és un punt a favor ja que podrem trobar més mòduls creats per membres de la comunitat que podem utilitzar per solucionar problemes que altre gent s'hagi trobat.

Aquest va ser el punt que va decantar la balança a favor de AngularJS, ja que es volia utilitzar aquest projecte per treballar amb una tecnologia actual que estigues en creixement.

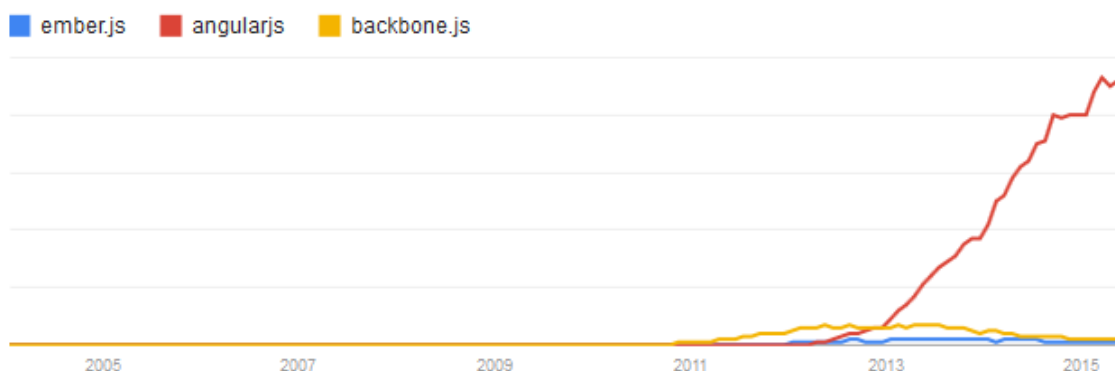


Figura 2.2.3.7: Interès en les tres tecnologies al llarg del temps segons cerques realitzades.

2.2.4 Bootstrap

Per tal de fer que el web tingui un disseny adaptatiu (responsive design) de manera senzilla i que no requereixi molta feina s'ha decidit utilitzar els CSS de bootstrap. Els quals permeten modificar l'aspecte del web utilitzant classes ja preparades de HTML.

Bootstrap utilitza un sistema de quadrícules que permeten estructurar els elements del web amb facilitat, permetent que aquest es vegi correctament en dispositius amb tota mena de mides de pantalla. També és possible indicar elements diferents en funció de la mida de la pantalla, permetent adaptar l'aspecte i la presentació als diferents dispositius.

2.2.5 Gestió i creació del projecte

2.2.5.1 Netbeans IDE

Es un programa de codi obert pensat per a treballar principalment amb Java ja que suporta tot tipus d'aplicacions Java. S'ha escollit aquesta IDE ja que és gratuïta i té total compatibilitat amb les tecnologies amb les que s'ha escollit treballar, sense necessitat d'instal·lar extensions addicionals.

Netbeans pot treballar amb Apache Tomcat, cosa que facilita molt el desenvolupament i la prova de la aplicació, permetent compilar i executar tot el projecte en la pròpia maquina prement només un botó.

2.2.5.2 Maven (Servidor)

Maven és una eina que permet gestionar i construir projectes basats en Java de forma ràpida i eficaç. Funciona a partir d'un fitxer de configuració anomenat POM (Project Object Model) en el que s'indiquen totes les dependències i versions de les mateixes que es fan servir en el projecte. Aquest fitxer de

configuració li permet saber a Maven quines llibreries descarregar sense necessitat de que un desenvolupador es vegi obligat a descarregar-les totes manualment.

El fet que Maven pugui descarregar totes els dependències a partir d'un únic fitxer de configuració permet que en equips de treball grans, no hi hagi problemes a l'hora de gestionar totes les llibreries, ja que si tothom utilitza el mateix fitxer de configuració, tothom tindrà les mateixes dependències a l'hora de construir el projecte.

2.2.5.3 NPM (Client)

És un programa que, mitjançant mòduls fets amb Node.js, permet carregar mòduls o paquets de JavaScript i descarregar amb facilitat les actualitzacions que es vagin fent sobre els mateixos. En el cas d'aquest projecte, NPM s'ha utilitzat per a carregar mòduls addicionals d'AngularJS. El fet de carregar aquests mòduls ja existents enlloc de crear les funcionalitats que ofereixen des de zero ha permès estalviar molt temps de programació.

2.2.5.4 Grunt (Client)

Grunt és un executador de tasques de JavaScript que permet automatitzar tasques repetitives com el compilat, minificació o tests que s'han d'executar el la part del client web. També disposa de plugins que permeten afegir-li funcionalitats. Grunt ha estat un dels mòduls que s'han afegit al projecte utilitzant NPM.

La principal funcionalitat que ha aportat ha estat la recompilació en viu de les plantilles HTML, els fitxers js i els fitxers CSS mitjançant una tasca anomenada Grunt Watch, la qual detecta modificacions realitzades als fitxers del projecte i els recompila.

Tenir aquest mòdul ha evitat haver de recompilar tot el projecte cada cop que s'ha fet una modificació, convertint així una tasca que podria gastar uns 5 minuts a uns 10 segons. Per a poder utilitzar Grunt per a automatitzar totes aquestes tasques, ha estat necessària la utilització de múltiples plugins de Grunt.

2.3 Sistema gestor de bases de dades

Al decidir quin sistema es faria servir per a gestionar la base de dades es van estudiar dues possibilitats, MySQL i Oracle. Es van detectar les següents diferències:

	Oracle	MySQL
Preu	Només el producte més bàsic és gratuït (fins a 4 GB).	La versió gratuïta és molt completa.
Administració i gestió	Requereix molta configuració per a entorns grans. Pot arribar a ser molt complex però també molt potent.	Molt senzill de posar en marxa i fer servir.
Potència	Pot arribar a ser molt potent si es configura bé.	Velocitat de lectura molt alta. Té problemes per a gestionar bases de dades molt grans amb eficiència.

Taula 2.3.1: Comparativa d'Oracle i MySQL

Degut a la mida bastant reduïda de la base de dades de la aplicació i a la simplicitat d'ús, s'ha decidit utilitzar MySQL. Oracle, tot i ser més potent té una barrera econòmica que només ens sortiria a compte travessar en cas de necessitar molta més potencia en la gestió de la base de dades, donada la simplicitat i mida reduïda de les dades que tenim, no és justificable l'esforç addicional que suposaria fer funcionar l'aplicació amb oracle.

La connexió entre Java i la base de dades s'ha realitzat amb una tècnica coneguda com a Object-relational mapping (ORM) utilitzant JPA/Hibernate (Figura 2.3.2). El que s'ha fet és utilitzar la implementació JPA d'Hibernate.

JPA no és més que una especificació que permet utilitzar anotacions en el codi Java per a gestionar la conversió entre classes de Java i taules de la base de dades. Per tal de que la connexió funcioni és necessari crear una implementació de les especificacions JPA, qui és on entra Hibernate.

Hibernate és una de les possibles implementacions de JPA, la qual s'encarrega de tractar amb la base de dades.

Treballar amb una capa ORM ens permet abstroure el codi Java de la base de dades, facilitant els canvis entre els diferents tipus de bases de dades. En un moment donat podríem parar d'utilitzar MySQL i començar a treballar amb Oracle i només hauríem de modificar el controlador que utilitza Hibernate, no seria necessari modificar el codi Java del servidor. Una altre avantatge de treballar amb una ORM és que la pròpia capa és capaç de fabricar la base de

dades mitjançant la informació de disseny de les classes Java que tenen anotacions JPA, fent possible que la base de dades s'hagi fabricat sense la necessitat de utilitzar codi SQL.

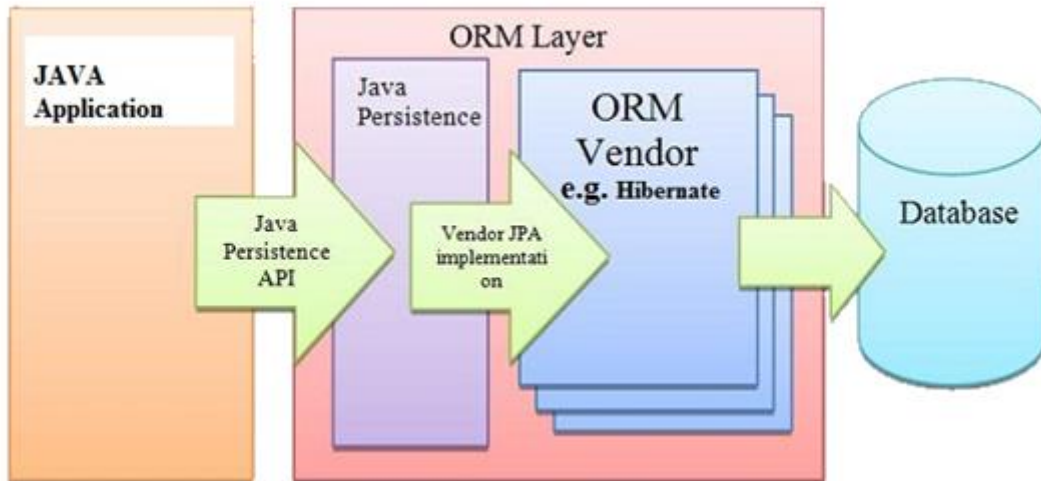


Figura 2.3.2: Funcionament Hibernate i JPA

2.4 Servidor web

S'ha fet servir la versió 8.0.23 d'Apache Tomcat per a gestionar l'aplicació web i fer la feina de servidor. Apache Tomcat és un gestor de servlets de JAVA de programari lliure. Una altra opció que s'ha tingut en compte a l'hora d'escollir el software del servidor és GlassFish. Al ser l'Apache Tomcat un programa més lleuger, simple i amb el qual ja s'havia treballat, no es va considerar justificable la complexitat addicional que afegia al projecte utilitzar GlassFish.

Durant el procés de desenvolupament s'ha utilitzat el servidor d'Apache Tomcat que ve integrat amb Netbeans, mentre que per a que els usuaris de la Facultat de Farmàcia de la UB realitzessin proves s'ha mantingut un ordinador encès contínuament amb els següents programes instal·lats:

- Apache Tomcat versió 8.0.23.
- XAMPP [\[9\]](#) com a base de dades MySQL.
- Dynamic DNS Update Client (DUC) v4.1.0.

Aquest programa serveix per a poder connectar al web que esta hostatjat en un servidor amb IP dinàmica (es modifica cada cop que es reinicia el router) utilitzant una URL que es redirigida automàticament des dels servidors de l'empresa que ofereix aquest servei.

La URL que s'ha d'utilitzar per accedir al web és la següent:

<http://apptfg.ddns.net/basic-web-app/app/index.html>

2.5 Diagrama de Gantt

En aquest apartat es detalla el diagrama de Gantt, figura 2.5. El projecte s'ha dividit en 5 apartats clarament diferenciats:

1. Estudi de requeriments i tecnologies

El primer mes i mig va estar dedicat a triar les tecnologies necessàries per a crear l'aplicació que es demanava. L'estudi de les tecnologies inclou tant el temps dedicat a fer la selecció com el temps d'estudi necessari per a familiaritzar-se amb elles abans de començar a treballar.

Apartats:

- Estudi AngularJS
- Estudi Spring MVC
- Estudi Hibernate/JPA
- Estudi Maven/Grunt/Npm

2. Preparació de l'entorn productiu

En una aplicació d'aquesta mida és necessari escollir una arquitectura que ens permeti organitzar tot el codi i els fitxers de configuració d'una manera que faciliti la feina.

- Instal·lació de dependències
- Creació del projecte i estructuració
- Configuració de Maven, Grunt i Npm.

3. Implementació

Aquest apartat engloba tota la feina de creació de codi que s'ha fet per a donar forma a l'aplicació, tant a la part del client web com del servidor.

- Servidor:
 - Connexió amb la base de dades
 - Configuració dels controladors Restful
- Client:
 - Gestió de css
 - Gestió de la càrrega dels fitxers js i de les plantilles (templates) HTML

- Client/Servidor:
 - Login
 - Registre
 - Consulta del llistat de principis actius
 - Permisos dels diferents tipus d'usuari
 - Afegir mòdul d'administració
 - Control del llistat des del mòdul d'administració
 - Gestió d'Usuaris des del mòdul d'administració
 - Gestió de missatges des del mòdul d'administració
 - Gestió del llistat de dominis web des del mòdul d'administració

4. Proves

Aquí s'engloben totes les proves que s'han realitzat per a assegurar que l'aplicació funciona correctament realitzant proves amb múltiples navegadors i amb múltiples usuaris. Durant aquest període de proves, la gent de la Facultat de Farmàcia ha testejat l'aplicació i ha donat feedback, donant lloc a modificacions y ajustos que han allargat el temps d'implementació.

- Proves amb múltiples navegadors
- Proves amb múltiples usuaris

5. Preparació del servidor Físic

En aquest apartat està tota la feina que es va realitzar per penjar l'aplicació en un servidor. Es van provar diferents serveis d'hostatgeria web per realitzar les proves amb els usuaris. Després de dedicar una setmana sense èxit, es va decidir hostatjar el projecte en un ordinador i configurar-lo per a que els usuaris hi poguessin accedir.

- Intentar configurar un servidor per a fer funcionar la aplicació
- Configuració d'un ordinador per a que actuï com a servidor

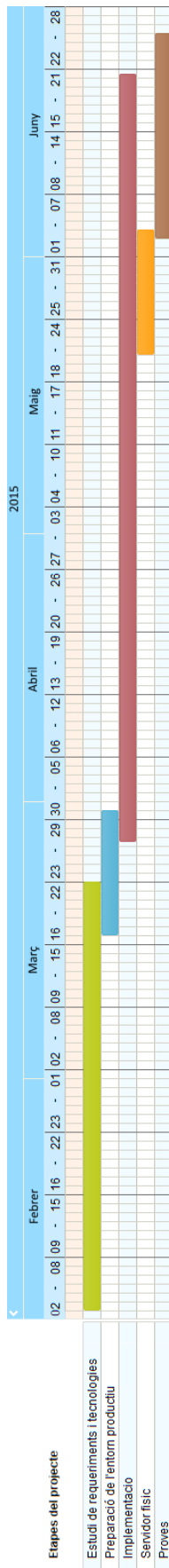


Figura 2.5: Diagrama de Gantt

2.6 Estudi econòmic

El cost total de l'aplicació, desglossat segons els apartats indicats al diagrama de Gantt és el següent:

Tasca	Temps (hores)	Preu (€/h)	Cost (€)
Estudi de requeriments i tecnologies	206	5	1.030
Preparació de l'entorn productiu	48	20	960
Implementació	311	20	6.220
Preparació del servidor físic	38	5	190
Proves	51	5	255
Total	654		8.655

Taula 2.6: Cost dels diferents apartats del projecte

En cas de tractar-se d'un projecte realitzat per algú que tingues coneixements previs sobre les tecnologies utilitzades i el funcionament dels servidors, el cost dels apartats estudi de requeriments i tecnologies, preparació de l'entorn productiu i servidor físic serien molt més baixos, ja que gran part del temps ha estat dedicat a l'aprenentatge. També indicar que gran part del temps dedicat a la implementació ha estat degut a la falta de familiaritat amb les tecnologies.

3 Anàlisis

3.1 Especificacions del problema per part del client

La base d'aquest treball és una taula que es va crear com a resultat d'un estudi [1] que es va realitzar per determinar si gent gran amb problemes de ronyó estaven rebent medicines amb principis actius que els hi podien causar efectes secundaris. La taula indica una sèrie de recomanacions a tenir en compte, segons una capacitat de filtrat i un principi actiu.

Es va crear una taula on es podia consultar, en funció de la capacitat de filtratge del pacient i el principi actiu, si el principi actiu li podia causar efectes secundaris. Aquest treball es construeix sobre aquesta taula, permetent a metges i farmacèutics consultar-la de forma telemàtica.

Com s'ha indicat a l'apartat 1.2 d'aquesta memòria, els objectius són:

- Crear un web d'accés restringit on metges i farmacèutics puguin consultar un llistat de principis actius.
- Crear un mòdul d'administració.
- Crear una API.

3.1.1 Web

El web ha de permetre als usuaris (metges i farmacèutics) consultar la taula de principis actius i veure les diferents recomanacions que es fan en funció de la capacitat de filtratge del pacient. Per tal d'evitar que pugui entrar gent que no siguin farmacèutics o metges s'ha creat un sistema de registre que requereix d'una direcció web d'un domini pertanyent a un col·legi de metges o farmacèutics, en la qual l'usuari rebrà la seva contrasenya un cop registrat.

Un cop l'usuari s'hagi registrat i tingui la seva contrasenya, aquest tindrà accés al web, on podrà fer tres coses:

- Consultar el llistat de principis actius i les seves recomanacions.
- Enviar un missatge amb recomanacions o apunts sobre el contingut del web.
- Modificar la seva contrasenya.

3.1.2 Mòdul d'administració

El web també ha de permetre que els usuaris administradors puguin gestionar els continguts de la base de dades.

- Podent carregar una taula nova de principis actius quan es faci una modificació sobre l'existent.
- Poder carregar de nou el llistat de dominis web que es poden fer servir per a registrar-se a l'aplicació.
- Veure el llistat d'usuaris registrats a l'aplicació.
- Veure els missatges que han enviat la resta d'usuaris.

3.1.3 API

L'API servirà per a que altres aplicacions tinguin accés a les dades de la base de dades sense necessitat de gestionar una base de dades pròpia i així poder centralitzar la informació, evitant els problemes que generaria mantenir múltiples bases de dades amb la mateixa informació. Aquesta API la utilitza el lloc web per a mostrar tota la informació.

La API és la que permet totes les funcionalitats indicades als apartats anteriors (3.1.1 i 3.1.2). Mentre que el web només s'encarrega de mostrar les dades que rep de la API d'una manera que resulti còmode i pràctica per als usuaris.

3.2 Requisits d'usuari

Qualsevol persona que es connecti al web podrà veure el formulari de registre i el formulari de login. Un cop l'usuari s'hagi registrat (si té un domini web vàlid) i hagi rebut la seva contrasenya al seu mail, podrà connectar-se i entrar a la aplicació.

Un cop dins de l'aplicació hi ha tres tipus d'usuaris:

- Administrador
És un usuari que té accés al mòdul d'administració que hi ha al web.
- Usuari
És un usuari que ha utilitzat el procés normal de registre (per tal de poder fer el procés de registre és necessari que l'usuari disposi d'una direcció de correu electrònic d'un dels col·legis de metges o farmacèutics d'Espanya).
- Usuari temporals
És un usuari que tindrà accés durant un temps limitat a l'aplicació, s'han de donar d'alta directament a la base de dades. Aquests usuaris seran alumnes de la Facultat de Farmàcia que utilitzaran el web per motius acadèmics.

Un cop logejats, els usuaris poden realitzar les següents accions, depenen del seu tipus:

- Usuaris, usuaris temporals i administradors:
 - Veure consultes
 - Filtrar consultes per nom
 - Filtrar consultes per rang
- Usuaris i administradors:
 - Enviar suggeriments
- Administradors:
 - Modificar contrasenya
 - Carregar principis actius
 - Carregar recomanacions
 - Esborrar principis actius i suggeriments
 - Veure llistat d'usuaris
 - Carregar dominis web
 - Veure dominis web
 - Esborrar dominis web
 - Veure suggeriments
 - Esborrar suggeriments

3.3 Requisits del sistema

Per tal de funcionar, l'aplicació que s'ha desenvolupat requereix d'un servidor web on allotjar la base de dades i des del qual respondre i gestionar totes les peticions dels usuaris que utilitzaran el client web.

El servidor que hostatjarà l'aplicació necessita el següent per funcionar:

- Apache Tomcat : 8.0.15.0
- MySQL: versió 5.6.21

També és útil tenir un gestor de bases de dades com phpMyAdmin, servirà sempre que permeti connectar amb MySQL.

Es recomana fer servir XAMPP^[9] (versió 5.6.8 o superior) ja que incorpora MySQL i el gestor de bases de dades.

Els usuaris requeriran d'un navegador web per tal d'accedir a l'aplicació web. Les aplicacions o persones que vulguin utilitzar la API només necessitar poder realitzar peticions HTTP. Servirà una versió actual de qualsevol navegador. L'aplicació no assegura compatibilitat amb versions antigues d'Internet Explorer.

3.4 Diagrama de casos d'ús

A la figura 3.3 es mostren els casos d'ús que es detallen a continuació en la secció 3.3.1.

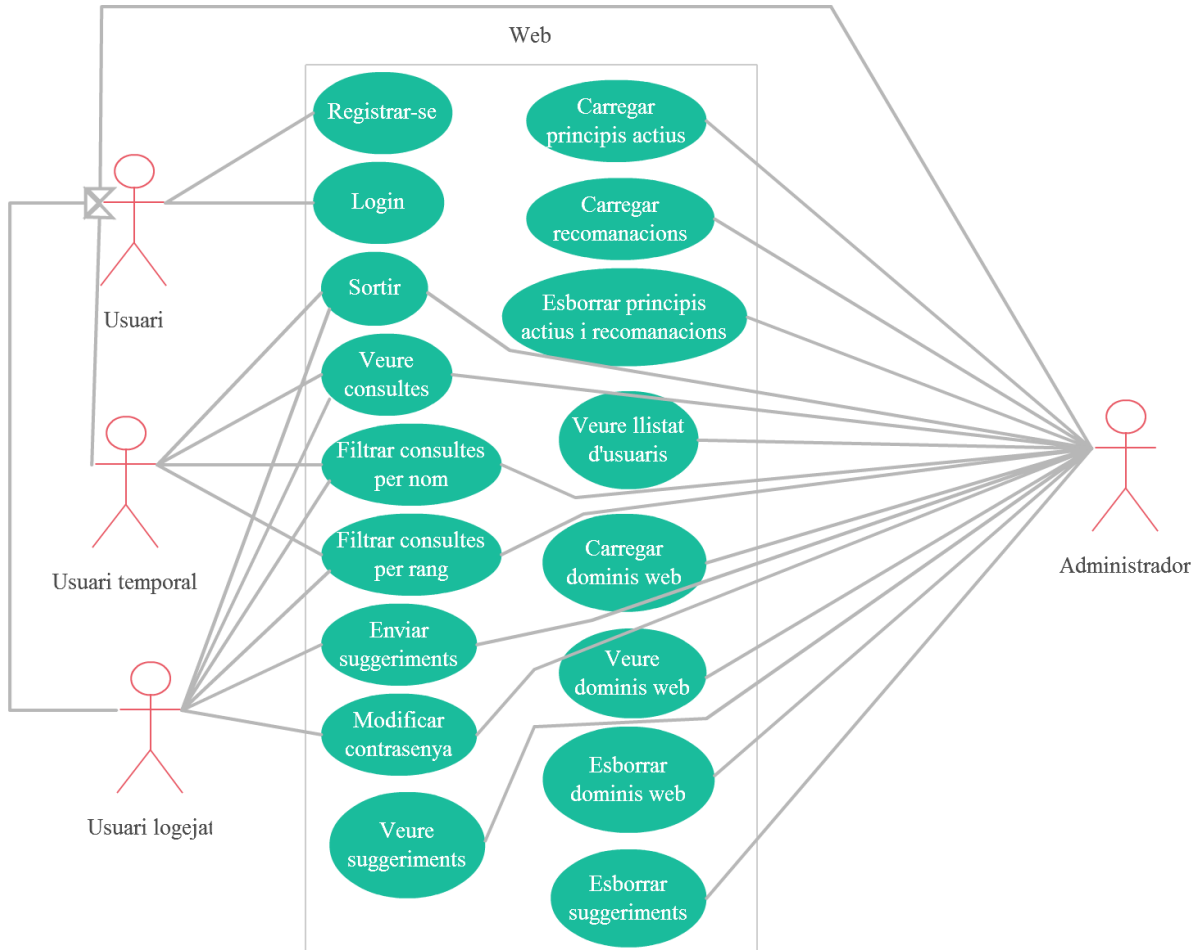


Figura 3.3 Diagrama de casos d'ús de l'aplicació

3.4.1 Casos d'ús textuais

Registrar-se

Descripció: uc01 – Registrar-se
Actors: Usuari
Precondicions: Entrar a la pàgina principal de l'aplicació i fer clic al botó de registrar-se.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per registrar-se2. L'usuari introdueix les dades3. El servidor comprova que el domini del mail tingui permís per a registrar-se i que no existeixi cap usuari amb aquest mail.4. El sistema mostra un missatge indicant que arribarà un mail amb la contrasenya.5. Es redirigeix l'usuari a la pàgina inicial.6. Arriba un correu amb la contrasenya al mail de l'usuari.
Flux Alternatiu: <ol style="list-style-type: none">3. El servidor indica que hi ha un problema amb les dades.4. El sistema mostra un missatge indicant que s'ha trobat un problema.5. El web torna a mostrar el formulari.
Postcondicions: L'usuari podrà logar-se a l'aplicació.

Login

Descripció: uc02 – Login
Actors: Usuari
Precondicions: Entrar a la pàgina principal de l'aplicació i fer clic al botó de logar-se.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per logar-se2. L'usuari introdueix les dades3. El servidor comprova que el mail i la contrasenya siguin correctes.4. L'usuari es logeja i passa a la pàgina de consulta.
Flux Alternatiu: <ol style="list-style-type: none">4. El servidor indica que hi ha un problema amb les dades amb un missatge d'error.5. Es torna a mostrar el formulari de login.
Postcondicions: L'usuari, usuari temporal o administrador esta logat al sistema.

Veure consultes

Descripció: uc03 – Veure consultes
Actors: Usuari logat, administrador, usuari temporal
Precondicions: L'usuari s'ha d'haver logat i estar a la pantalla de consultes.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari veu el llistat de principis actius a la pàgina.2. L'usuari fa clic el nom del principi actiu que vol consultar.
Flux Alternatiu: <ol style="list-style-type: none">2. L'usuari no troba el principi actiu que vol consultar.
Postcondicions: L'usuari, usuari temporal o administrador veu la informació de la consulta.

Filtrar consultes per nom

Descripció: uc03.1 – Filtrar consultes per nom
Actors: Usuari logat, administrador, usuari temporal
Precondicions: L'usuari s'ha d'haver logat i estar a la pantalla de consultes.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari veu el llistat de principis actius a la pàgina.2. L'usuari escriu el nom del principi actiu que vol buscar.3. L'usuari fa clic el nom del principi actiu que vol consultar.
Flux Alternatiu: <ol style="list-style-type: none">3. L'usuari no troba el principi actiu que vol consultar.
Postcondicions: L'usuari, usuari temporal o administrador veu la informació de la consulta.

Filtrar consultes per rang

Descripció: uc03.2 – Filtrar consultes per rang
Actors: Usuari logat, administrador, usuari temporal
Precondicions: L'usuari s'ha d'haver logat i estar a la pantalla de consultes.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari veu el llistat de principis actius a la pàgina.2. L'usuari escriu el nom del principi actiu que vol buscar.3. L'usuari fa clic el nom del principi actiu que vol consultar.
Flux Alternatiu:
Postcondicions: L'usuari, usuari temporal o administrador veu la informació de la consulta.

Enviar suggeriments

Descripció: uc03.3 – Enviar suggeriments
Actors: Usuari logat, administrador
Precondicions: L'usuari s'ha d'haver logat i estar a la pantalla de consultes.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari veu el llistat de principis actius a la pàgina.2. L'usuari fa clic a la opció d'enviar un missatge.3. L'usuari escriu el missatge i fa clic al botó d'enviar.4. El sistema mostra un missatge indicant que s'ha enviat correctament
Flux Alternatiu: <ol style="list-style-type: none">4. El sistema mostra un missatge dient que s'ha trobat un problema en l'enviament.
Postcondicions: L'usuari, usuari temporal o administrador veu la informació de la consulta.

Modificar contrasenya

Descripció: uc04 – Modificar contrasenya
Actors: Usuari logat, administrador
Precondicions: L'usuari s'ha d'haver logat i estar a la pantalla de modificar contrasenya.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per modificar la contrasenya2. L'usuari introdueix les dades3. El servidor comprova que la contrasenya siguin correcte.4. El sistema mostra un missatge dient que s'ha modificat la contrasenya correctament.
Flux Alternatiu: <ol style="list-style-type: none">4. El sistema mostra un missatge dient que s'ha trobat un problema en la modificació.
Postcondicions:

Carregar principis actius

Descripció: uc05.1 – Carregar principis actius
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla de modificar base de dades i la taula de principis actius de la base de dades ha d'estar buida.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari fa clic al botó de carregar principis actius2. L'usuari selecciona el fitxer csv amb la informació.3. El servidor comprova que la informació sigui correcte4. El servidor carrega la informació a la base de dades.5. El sistema mostra un missatge dient que s'ha carregat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">4. El sistema mostra un missatge dient que s'ha trobat un problema en la càrrega.
Postcondicions: Es carreguen els principis actius a la base de dades.

Carregar recomanacions

Descripció: uc05.2 – Carregar recomanacions
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla de modificar base de dades i la taula de principis actius de la base de dades ha tenir carregats els principis actius.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari fa clic al botó de carregar principis actius2. L'usuari selecciona el fitxer csv amb la informació.3. El servidor comprova que la informació sigui correcte i la carrega a la base de dades.4. El sistema mostra un missatge dient que s'ha carregat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">4. El sistema mostra un missatge dient que s'ha trobat un problema en la càrrega.
Postcondicions: Es carreguen les recomanacions a la base de dades.

Esborrar principis actius i suggeriments

Descripció: uc05.2 – Esborrar principis actius i suggeriments
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla de modificar base de dades i la taula de principis actius de la base de dades ha d'estar plena.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per esborrar els principis actius2. L'administrador escriu la contrasenya d'administrador3. L'administrador fa clic al botó d'esborrar les taules de principis actius i recomanacions.4. El servidor comprova que la informació sigui correcte5. El servidor esborra les dades de la base de dades.6. El sistema mostra un missatge dient que s'ha esborrat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">5. El sistema mostra un missatge dient que s'ha trobat un problema amb la contrasenya.
Postcondicions: S'esborren les taules de principis actius i recomanacions.

Veure llistat d'usuaris

Descripció: uc06 – Veure llistat d'usuaris
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat i estar a la pantalla d'administrar usuaris.
Flux bàsic: <ol style="list-style-type: none">1. L'administrador veu el llistat d'usuaris que estan registrats a l'aplicació..
Flux Alternatiu:
Postcondicions:

Carregar dominis web

Descripció: uc06.1 – Carregar dominis web
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla de modificar base de dades i la taula de principis actius de la base de dades ha tenir carregats els principis actius.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari fa clic al botó de carregar principis actius2. L'usuari selecciona el fitxer csv amb la informació.3. El servidor comprova que la informació sigui correcte4. El servidor carrega la informació a la base de dades.5. El sistema mostra un missatge dient que s'ha carregat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">4. El sistema mostra un missatge dient que s'ha trobat un problema en la càrrega.
Postcondicions: Es carreguen les recomanacions a la base de dades.

Veure dominis web

Descripció: uc06.2 – Veure dominis web
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat i estar a la pantalla d'administrar dominis web.
Flux bàsic: <ol style="list-style-type: none">1. L'administrador veu el llistat de dominis web que es poden registrar a l'aplicació..
Flux Alternatiu:
Postcondicions:

Esborrar dominis web

Descripció: uc06.3 – Esborrar dominis web
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla d'administrar dominis web.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per esborrar els dominis web2. L'administrador escriu la contrasenya d'administrador3. L'administrador fa clic al botó d'esborrar la taula de dominis web.4. El servidor comprova que la informació sigui correcte5. El servidor esborra la base de dades.6. El sistema mostra un missatge dient que s'ha esborrat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">5. El sistema mostra un missatge dient que s'ha trobat un problema amb la contrasenya.
Postcondicions: S'esborren les taules de principis actius i recomanacions.

Veure suggeriments

Descripció: uc07.1 – Veure suggeriments
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat i estar a la pantalla d'administrar suggeriments.
Flux bàsic: <ol style="list-style-type: none">1. L'usuari veu el llistat de suggeriments que han enviat els usuaris.
Flux Alternatiu:
Postcondicions:

Esborrar suggeriments

Descripció: uc07.2 – Esborrar suggeriments
Actors: Administrador
Precondicions: L'administrador s'ha d'haver logat, estar a la pantalla d'administrar suggeriments i la taula de suggeriments de la base de dades ha d'estar plena.
Flux bàsic: <ol style="list-style-type: none">1. El web mostra un formulari amb les dades necessàries per esborrar els suggeriments2. L'administrador escriu la contrasenya d'administrador3. L'administrador fa clic al botó d'esborrar la taula de suggeriments.4. El servidor comprova que la informació sigui correcte5. El servidor esborra la base de dades.6. El sistema mostra un missatge dient que s'ha esborrat la informació correctament.
Flux Alternatiu: <ol style="list-style-type: none">5. El sistema mostra un missatge dient que s'ha trobat un problema amb la contrasenya.
Postcondicions: S'esborra la taula de suggeriments.

4. Desenvolupament.

4.1 Arquitectura general del projecte

Aquest projecte es divideix en dues parts, client i servidor. La part del servidor és on es troba la base de dades i el codi java mentre que el client és el que engloba el lloc web fet amb AngularJS. L'únic punt de connexió entre aquestes dues parts és la API desenvolupada en Java. Aquesta separació entre client i servidor permet modificar el client sense afectar el funcionament del servidor, cosa que facilita la creació i funcionament de múltiples clients de forma simultània, per exemple, una web i múltiples aplicacions mòbils treballant al mateix temps contra un únic servidor.

4.1.1 Servidor

El servidor és la part més important que s'ha desenvolupat en aquest treball, ja que és l'encarregat d'emmagatzemar i gestionar les dades. El framework que s'ha utilitzat per al servidor és Spring MVC, el funcionament del qual ha estat explicat a l'apartat 2.2.2 d'aquest escrit.

L'estructura de classes del servidor pot veure a la figura 4.1.1.1 , 4.1.1.2 , . Aquesta estructura de classes és idèntica per a totes les classes principals de Java utilitzades, les quals es poden veure a la figura 4.2.2.

La classe Account és on està emmagatzemada tota la informació d'un usuari registrat. Tota la resta de classes serveixen per a convertir la informació que ens arriba del client a una classe Java o per treure la informació de la base de dades i crear una classe java.

Account

Aquesta és la classe que agrupa tota la informació i amb la que treballa el codi Java del servidor.

AccountController

A aquesta classe és on arriben les peticions que es fan a l'apartat de gestió de comptes de la API.

AccountResource

Aquesta classe es fa servir per a rebre o enviar la informació de la classe Account al client.

AccountResourceAsm

És l'encarregada de codificar i descodificar la classe AccountResource per a que el servidor la pugui utilitzar. Transforma els bytes que li arriben al controlador en una classe de Java que el servidor pot utilitzar.

AccountService i AccountServiceImpl

Aquesta interfície i la seva implementació s'encarreguen de realitzar tota la lògica del servidor. La informació entra a través del controlador, és preparada i passa al AccountServiceImpl el qual la tracta, comprova que sigui correcte i decideix per a que la farà servir. Ja sigui crear una nova instància de la classe o modificar/eliminar una instància existent. Aquí també és on s'executen totes les accions addicionals al tractament d'una classe, com per exemple l'enviament del mail amb la contrasenya.

AccountRepo i JpaAccountRepo

Aquesta interfície i la seva implementació s'encarreguen de fer de pont entre el codi Java i la base de dades.

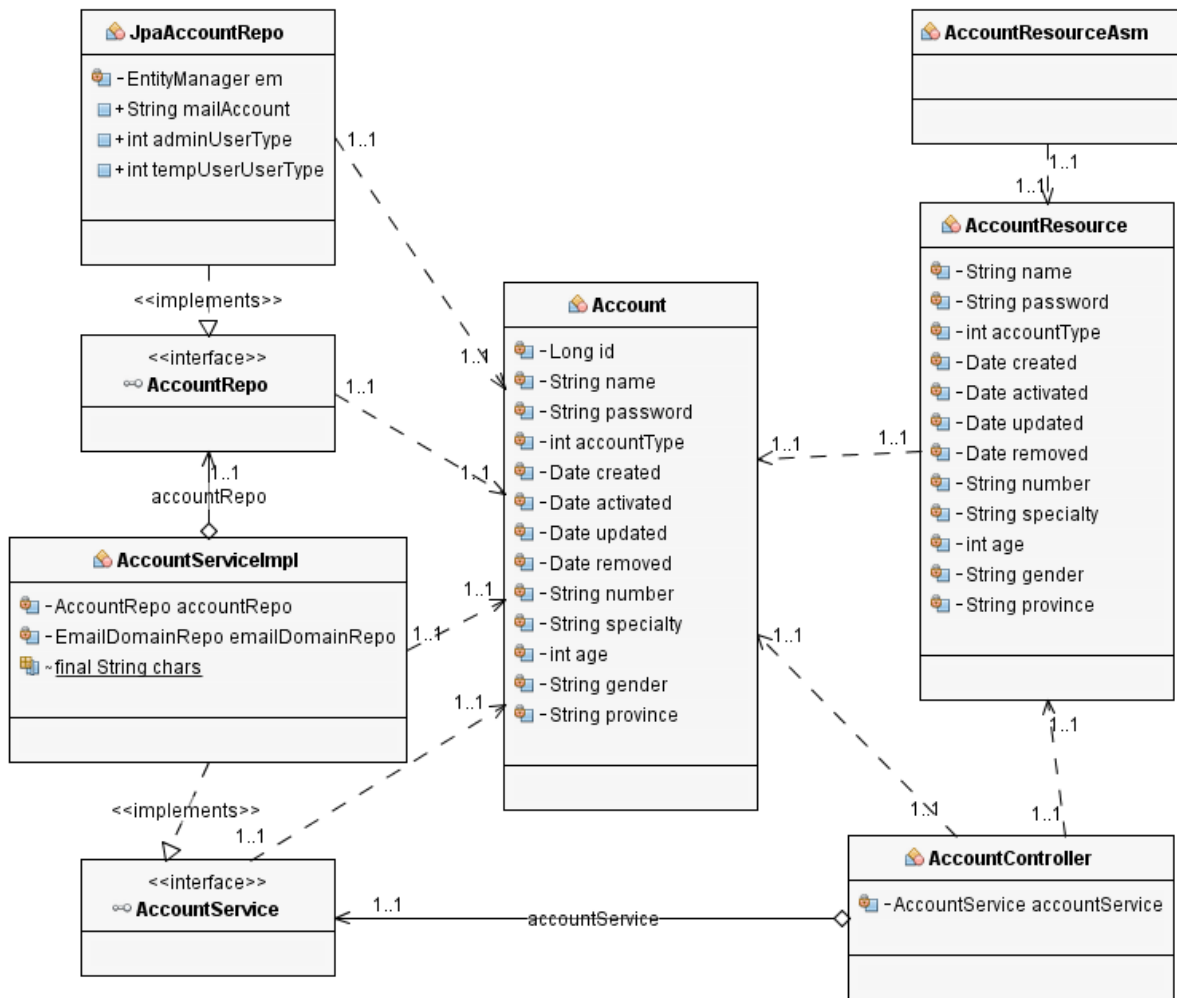


Figura 4.1.1.1 Estructura de gestió de la classe Account en el servidor

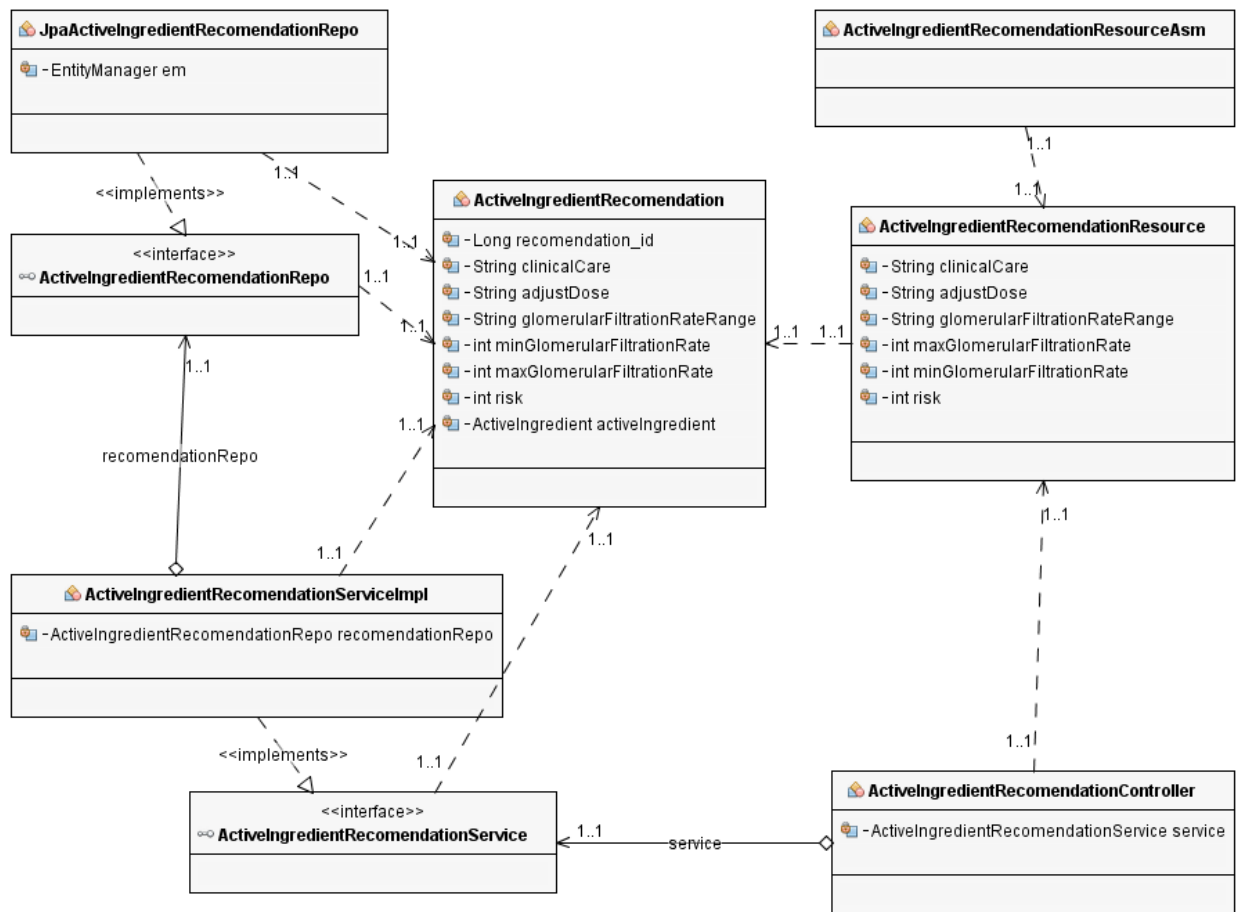


Figura 4.1.1.2 Estructura de gestió de la classe ActiveIngredient en el servidor

4.1.2 Client

El client està dissenyat com a una single-page application (SPA), ja que només té una única pàgina HTML anomenada index.html (com es pot veure a la Figura 4.1.2). En aquesta pàgina HTML és on es carreguen totes les dependències del web (fitxers js i css). Tota la navegació del web es fa mitjançant AngularJS, el qual s'encarrega d'anar carregant els diferents templates per a cada estat del web (un estat és una pàgina diferent carregada sense necessitat de cridar al servidor). A la figura 4.1.2 es poden veure els diferents templates que s'utilitzen a casa estat.

Per tal de carregar dades del servidor als diferents estats, s'utilitzen els serveis. Aquests serveis (indicat a la figura 4.1.2 quins serveis s'utilitzen a cada estat) són els encarregats de demanar la informació al servidor, el qual l'envia en format JSON, i passar-la als estats per a que ajuntin la informació amb els templates per tal de mostrar a l'usuari la pàgina completa.

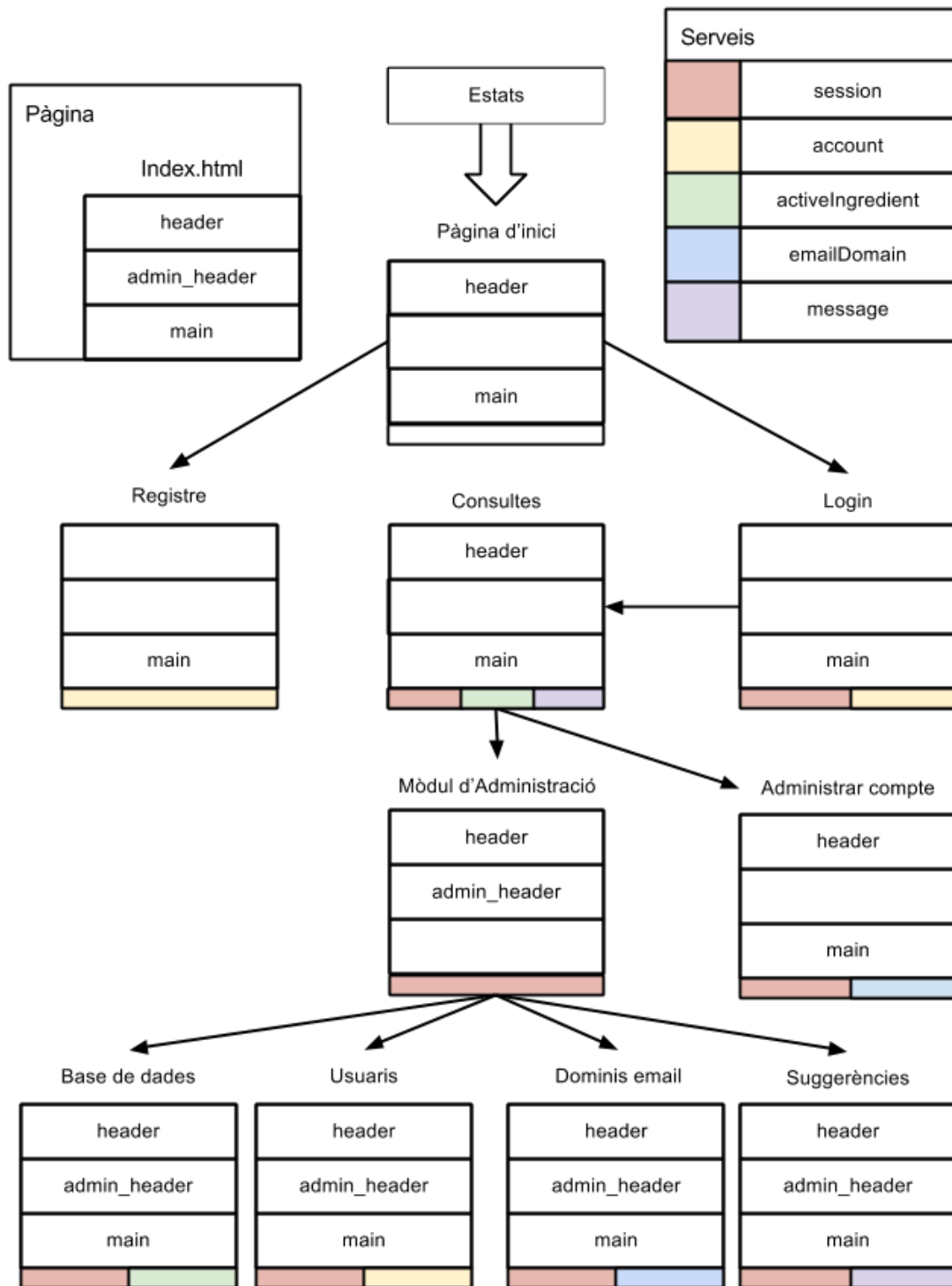


Figura 4.1.2: Estructura del web

4.2 Relació de les classes

La base de dades guarda tota la informació que necessita l'aplicació per funcionar. Les dades s'han dividit en 5 taules per tal d'evitar la repetició d'informació, estalviant espai i facilitant el manteniment.

La informació de la base de dades només és accedida per el codi Java, ja que el client web només serveix per a connectar amb la API.

Les classes que s'han utilitzat en el codi Java (figura 4.2.2) coincideixen amb les taules de la base de dades (Figura 4.2.1) ja que la capa ORM (Hibernate amb JPA) ha creat les taules a partir de les classes, afegint també les relacions entre elles.

Totes les classes utilitzades tenen un id per facilitar les consultes complexes a la base de dades:

EmailDomain

Conté els dominis web que es poden registrar a l'aplicació.

ActiveIngredient

Conté tota la informació d'un principi actiu que no depèn de la capacitat de filtratge del pacient.

ActiveIngredientRecommendation

Conté tota la informació d'un principi actiu que depèn de la capacitat de filtratge del pacient.

Account

Conté tota la informació que ens facilita un usuari al registrar-se al web més el tipus de compte que té (usuari, administrador o usuari temporal).

Message

Conté els suggeriments que han enviat els usuaris del web, guardant el text del missatge, qui l'ha enviat i la data d'enviament.

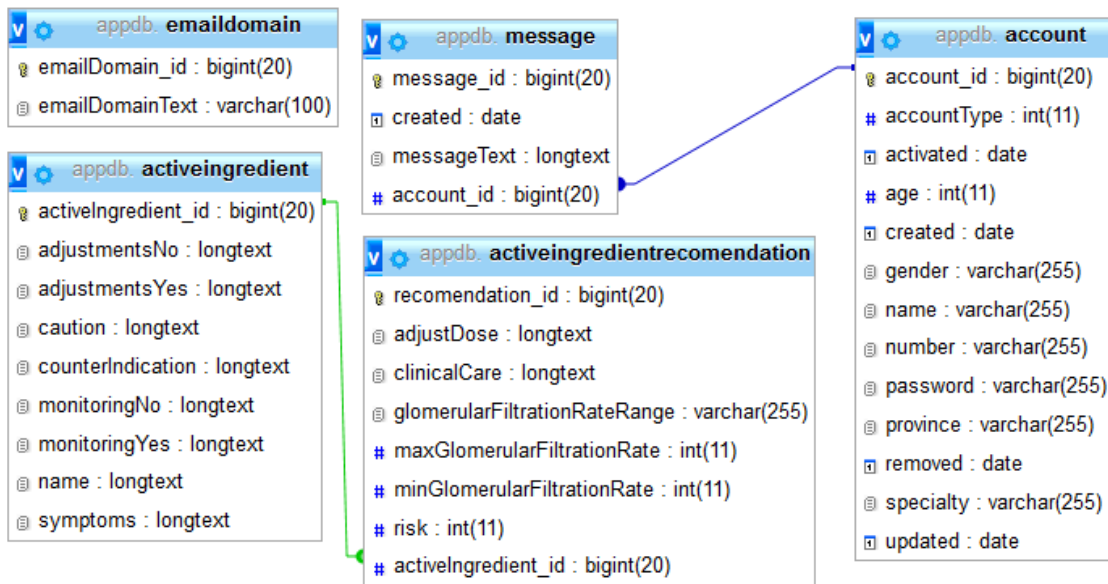


Figura 4.2.1: Diagrama de taules de la base de dades

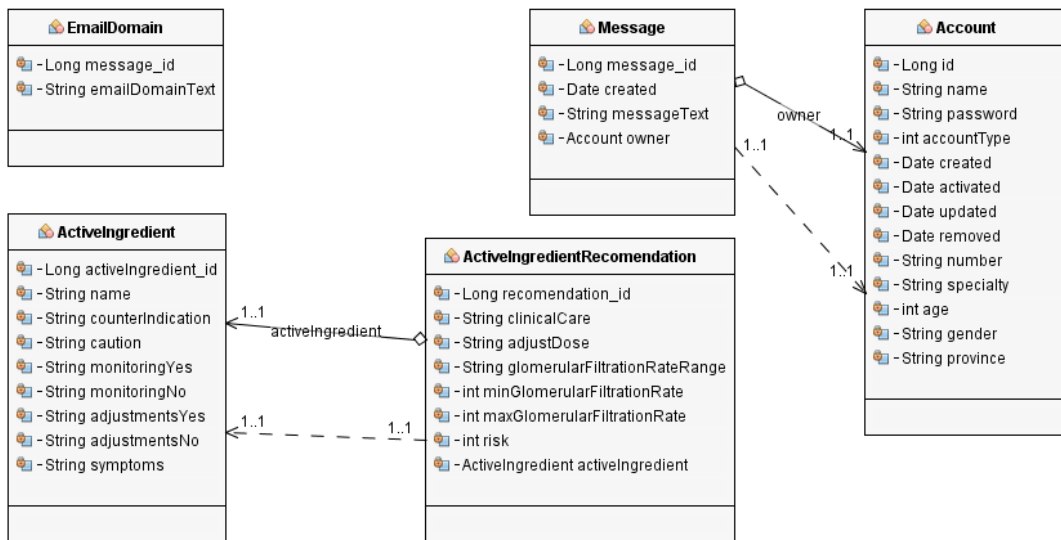


Figura 4.2.2: Diagrama de classes de Java

5. Proves i resultats

En aquest apartat es resumeixen els resultats finals del lloc web. Per veure el resultat final es pot accedir al web que esta hostatjat actualment a la url:

<http://apptfg.ddns.net/basic-web-app/app/index.html>

Per tal de registrar-se al web és necessari utilitzar un compte de correu de gmail o hotmail. Per tal de veure les opcions d'administració es pot accedir amb l'usuari "Admin" que té la contrasenya "1234".

5.1 Proves realitzades

S'ha comprovat que el web funciona correctament en els següents navegadors i dispositius (s'han detectat alguns problemes d'estils molt menors a Internet Explorer i Safari):

- **PC:**
 - Chrome
 - Firefox
 - Internet Explorer
- **Mac:**
 - Chrome
 - Firefox
 - Safari
- **Ipad**
- **Android tablet**

Amb aquestes proves es considera que qualsevol usuari pot accedir amb facilitat al web des d'algun dispositiu.

La principal diferencia d'aspecte és en el Ipad i la tableta Android, ja que al ser més estretes que una pantalla d'ordinador el web es veu diferent, com es pot apreciar a les Figures 5.1.1 i 5.1.2.

Guia AP-IR

Q Consultas

Administrar cuenta

Administración

Salir

Principio activo

FGe

En caso de no seleccionar un FGe, saldrá todo

Muy bajo Bajo Medio Alto

AAS

ACENOCUMAROL

Figura 5.1.1: Aspecte de la pàgina de consultes en tablet

Guia AP-IR

Administración

Modificar Base de datos

Gestión de usuarios

Listado de mails

Sugerencias

Cargar principios activos desde excel

Browse... No file selected.

Cargar Recomendaciones desde excel

Browse... No file selected.

Figura 5.1.2: Aspecte de la pàgina d'administració en tablet

5.2 Descripció de l'aspecte de l'aplicació

L'aspecte final de l'aplicació està altament influenciat per la decisió d'utilitzar els css de Bootstrap. A l'utilitzar aquests css, tota la aplicació segueix l'estil que aquests marquen. L'aspecte final és bastant minimalista, utilitzant només colors clars (com es pot apreciar a la figura 5.2) i línies simples.

The screenshot displays the user interface of the web application. At the top, there is a navigation bar with the following elements: 'Guia AP-IR', 'Consultas' (with a magnifying glass icon), 'Administrar cuenta' (with a gear icon), 'Administración' (with a gear icon), and 'Salir' (with an arrow icon). Below the navigation bar, there is a search filter section. It includes a text input field labeled 'Principio activo', a dropdown menu labeled 'FGe' with a selection icon, and a note: 'En caso de no seleccionar un FGe, saldrá todo'. Below these are four colored buttons representing filter levels: 'Muy bajo' (green), 'Bajo' (orange), 'Medio' (red), and 'Alto' (dark red). Underneath the buttons is a list of medication names in a light gray box: AAS, ACENOCUMAROL, ALMAGATO, AMIODARONA, and AMLODIPINO.

Figura 5.2: Aspecte de la pàgina de consultes del web

L'aspecte de l'aplicació es pot observar als Annexos 1 i 2. En el manual d'usuari i el manual d'administrador es fa un recorregut per tot el web indicant tot el que poden fer els usuaris i els administradors.

5.3 Feedback del client

Al llarg del desenvolupament del projecte s'han anat realitzant reunions amb els professors de la Facultat de Farmàcia de la UB per anar definint conjuntament l'aspecte del web i que poguessin indicar les modificacions volien realitzar. Això s'ha fet així per tal d'evitar tots els problemes que hauria causat mostrar el producte final ja acabat i preguntar llavors quines modificacions volien realitzar. Involucrant al client en el desenvolupament del producte, s'aconsegueix un resultat molt més pròxim al que ells volen, minimitzant la quantitat de vegades que s'ha de reescriure o modificar el codi.

Apart d'aquestes reunions que s'han anat realitzant aproximadament cada 3 setmanes, durant la part final del projecte s'ha mantingut el web hostatjat en un servidor(<http://apptfg.ddns.net/basic-web-app/app/index.html>) per tal de que poguessin anar veient els canvis que es realitzaven de forma instantània. Això ha permès que el feedback fos més constant i ha ajudat bastant en el esprint final.

La majoria de les peticions que han realitzat posteriorment a la definició de requeriments inicial han estat purament cosmètiques, essent la estructura de la pàgina de consultes la única que s'ha demanat modificar múltiples vegades.

Els professors de farmàcia han mostrat en tot moment un gran interès pel projecte i la web desenvolupada.

5.4 Concordança de resultats i objectius

S'ha aconseguit acabar la major part dels objectius que es van marcar al principi del treball.

Els objectius aconseguits han estat els següents:

- Servidor:
 - Connexió amb la base de dades
 - Configuració dels controladors Restful
- Client:
 - Gestió de css
 - Gestió de la càrrega dels fitxers js i de les plantilles (templates) HTML
- Client/Servidor:
 - Login
 - Registre
 - Consulta del llistat de principis actius
 - Permisos dels diferents tipus d'usuari
 - Afegir mòdul d'administració
 - Control del llistat des del mòdul d'administració
 - Gestió d'Usuaris des del mòdul d'administració
 - Gestió de missatges des del mòdul d'administració
 - Gestió del llistat de dominis web des del mòdul d'administració

Els dos objectius que no s'han pogut tancar han estat: la gestió d'usuaris temporals des del mòdul d'administració del web i la recuperació de contrasenyes perdudes per part dels usuaris.

Els usuaris temporals es poden crear directament a la base de dades i són completament funcionals, la part que falta és que els administradors els puguin crear i eliminar des del mòdul d'administració. Per tal d'aconseguir-ho s'hauria de modificar el client per tal d'afegir els formularis necessaris i en el servidor s'haurien d'afegir els controladors que gestionarien les peticions del client.

La funcionalitat de recuperar la contrasenya no ha estat desenvolupada per restriccions temporals, però seria molt simple afegir aquesta opció a la pàgina de login del web.

6. Conclusions i treball futur

6.1 Conclusions

S'ha desenvolupat una API i un lloc web que rep i envia informació d'ella. Aquest lloc web és d'accés restringit, per a que només gent que pertanyi a un col·legi de farmàcia o medicina d'Espanya hi pugui accedir. Aquest web també incorpora un mòdul d'administració que permet gestionar la base de dades del servidor.

El desenvolupament d'aquest Treball de Final de Grau m'ha permès utilitzar moltes tecnologies que no havia utilitzat mai. Això m'ha fet estudiar i investigar sobre les moltes possibilitats que tenen les aplicacions web, en concret les SPA i el gran potencial que tenen les APIs.

També m'ha permès seguir el procés de creació d'una aplicació d'una dimensió molt més gran que cap que mai havia fet, cosa que m'ha fet adonar-me'n de la complexitat que pot arribar a tenir utilitzar múltiples tecnologies al mateix temps.

6.2 Possibles millores de l'aplicació

Aquest projecte s'ha creat com a base a possibles desenvolupament futurs. El fet d'haver creat una API facilita molt la creació d'aplicacions de consulta per a Android i/o iOS. Les aplicacions només s'haurien d'encarregar de demanar i mostrar la informació, no haurien de tractar-la ja que tota aquesta feina ja esta feta en el servidor.

També hi ha feina a fer en la part ja creada del servidor i del client. Les principals millores que necessita l'aplicació són les següents:

- **Seguretat**
Tot i haver configurar el mòdul de seguretat de Spring MVC, no s'ha arribat a utilitzar amb tot el seu potencial. Configurant correctament aquest mòdul es podria evitar que usuaris no registrats no poguessin fer peticions a la API, augmentant molt la seguretat.
També es podria fer un sistema de tokens encriptats utilitzant una llibreria com JavaScript Object Signing and Encryption (JOSE) per tal de controlar qui esta fent les peticions.
- **Google Analytics**
Aquest plugin de js es podria afegir i configurar per tal de que ens permetés obtenir informació sobre els usuaris que utilitzen l'aplicació i

sobre com la utilitzen. Tota aquesta informació es podria veure des d'un web de Google.

- **Optimitzar consultes a la base de dades**

La consulta que s'encarrega d'extreure el llistat de principis actius i recomanacions de la base de dades i enviar-lo al client web esta molt poc optimitzada, donant lloc a esperes per part de l'usuari. Aquesta consulta es podria millorar si es configurés JPA adequadament, convertint una consulta que genera centenars de peticions a base de dades en només dues peticions.

Bibliografia.

[1] *Effectiveness of a drug dosing service provided by community pharmacists in polymedicated elderly patients with renal impairment — a comparative study*

<http://biomedcentral.com/1471-2296/14/96>

[2] AngularJS. <https://angularjs.org/>

[3] Nodejs. <https://nodejs.org/>

[4] NPM. <https://www.npmjs.com/>

[5] Netbeans. <https://netbeans.org/>

[6] Java. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

[7] Spring MVC. <https://spring.io/guides/gs/serving-web-content/>

[8] Hibernate/JPA. <http://hibernate.org/orm/>

[9] XAMPP. <https://www.apachefriends.org/index.html>

[10] Apache Tomcat 8.0.23: <http://tomcat.apache.org/download-80.cgi>

[11] Bootstrap CSS. <http://getbootstrap.com/css/>

[12] Grunt. <http://gruntjs.com/>

- *Figura 2.2.2:*

<https://malalanayake.wordpress.com/2014/07/27/spring-mvc-with-spring-hibernate/>

- *Figura 2.2.3.1:*

<http://www.adevedo.com/content/why-use-angularjs-very-short-introduction>

- *Figura 2.2.3.5 i Figura 2.2.3.6:*

<http://www.creativebloq.com/web-design/angularjs-51411807>

- *Figura 2.2.3.7: s'ha utilitzat l'eina trends de Google.*

<https://www.google.com/trends/explore?hl=en-US#q=ember.js,+angularjs,+backbone.js&cmpt=q>

Annexos

Annex A: Manual d'usuari

1. Pantalla Inicial

Aquesta és la pàgina d'inici que es mostra al obrir l'aplicació web.



1. **Inici Sessió:** permet entrar a l'aplicació com a usuari logat per poder accedir als serveis que ofereix la Guía AP-IR-
2. **Crear compte:** en cas de que no es disposi d'un compte d'usuari, amb aquesta opció, l'usuari podrà registrar-se.

1.1. Inici de sessió

Aquesta és la pantalla d'inici de sessió, on haurem de introduir les nostres dades.

Iniciar sesión

Email del colegio:

1

Contraseña:

2

Iniciar sesión

1. **Correu Electrònic:** aquí introduïrem el correu electrònic amb el que ens hem registrat prèviament a l'aplicació.

2. **Contrasenya:** paraula de pas proporcionada per l'aplicació en el moment del registre. Més endavant podrem modificar-la en les opcions del comte d'usuari. Pantalla 3.

1.2. Crear compte

Aquest és el formulari de registre d'un nou compte d'usuari. Accessible des de la pantalla inicial.

Registrar un usuario nuevo

Email del colegio 1
El email del colegio es obligatorio

Número de colegiado 2
El número de colegiado es obligatorio

Edad 3
La edad es obligatoria

Provincia 4

Sexo 5

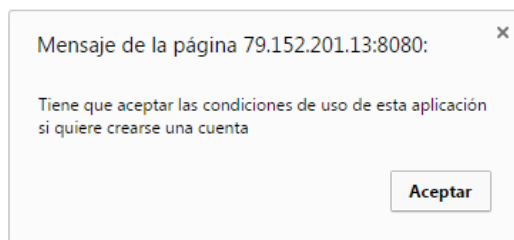
Especialidad 6

Al registrarse en esta aplicación acepta los terminos y condiciones de uso. 7

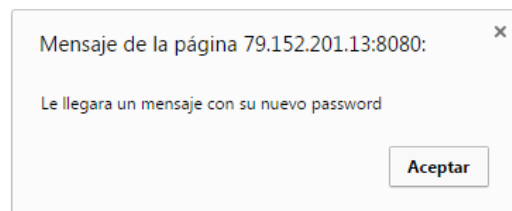
8 9

1. **Correu Electrònic:** aquí introduïrem el correu electrònic amb el que ens hem registrat prèviament a l'aplicació.
2. **Nombre de col·legiat:** nombre de col·legiat personal de l'usuari.
3. **Edat**
4. **Província**
5. **Sexe**
6. **Especialitat:** especialitat mèdica de l'usuari.
7. **Termes i condicions d'ús:** si acceptem o no les condicions d'ús de l'aplicació. És obligatori acceptar-les per poder acabar el registre.
8. **Crear usuari:** finalitzar el registre, enviant les dades al servidor
9. **Reset:** per començar de nou en cas de que sigui necessari.

En cas de que ens falti algun camp per completar o hi hagi alguna errada es mostrarà el següent missatge:



Si el registre ha estat un èxit es notificarà a l'usuari amb el següent missatge i haurà de rebre un correu amb la contrasenya d'accés a l'aplicació.



2. Consultes

Aquesta és la pantalla mostrada als usuaris un cop logats a l'aplicació.

1 2 6

Guía AP-IR Q Consultas * Administrar cuenta Salir

3 4 5

Principio activo FGe En caso de no seleccionar un FGe, saldrá todo

Muy bajo Bajo Medio Alto

ALMAGATO
CARBONATO CÁLCICO / COLECALCIFEROL
CINITAPRIDA
CLORURO SODICO
DOMPERIDONA
ESOMEPRAZOL
GLIBENCLAMIDA

1. **Consultes:** és un tab de navegació que ens permet accedir a la pàgina de consultes.
2. **Administrar compte:** és un tab de navegació que ens permet gestionar opcions del compte d'usuari. Ho veurem més endavant.
3. **Cerca per principi actiu:** és un camp on podem escriure un principi actiu, i l'aplicació sols ens mostrarà les coincidències.
4. **Capacitat de filtratge:** és un camp on podem escriure un nombre positiu, i l'aplicació ens mostrarà les recomanacions corresponents dintre de cada principi actiu. En cas de deixar aquest camp en blanc, apareixeran les recomanacions per a tots els filtratges possibles.
5. **Llegenda de colors:** ahora mostrar la informació el color de fons apareixerà segons la capacitat de filtratge. Més endavant veurem un exemple.
6. **Sortir:** és un botó que ens permet tancar la sessió d'usuari actual.

- Exemple d'ús de la cerca per principi actiu:
 - Al escriure el text "ALM", només sens mostren aquells principis actius que coincideixen amb la cerca.

The screenshot shows a search interface with two input fields: 'ALM' and 'FGe'. To the right of the 'FGe' field is the text 'En caso de no seleccionar un FGe, saldrá todo'. Below the input fields is a horizontal bar with a green segment labeled 'Muy bajo' and an orange segment labeled 'Ba'. Below this bar is a list of results, with 'ALMAGATO' visible. At the bottom, there is a section labeled 'Sugerencias'.

- Exemple d'ús del camp de capacitat de filtratge
 - FGe = 10: Mostra les recomanacions per a un FGe inferior a 30.

The screenshot shows the search interface with 'ALM' in the first input field and '10' in the 'FGe' field. The text 'En caso de no seleccionar un FGe, saldrá todo' is present. The horizontal bar shows 'Muy bajo' in green and 'Bajo' in orange. The results section for 'ALMAGATO' includes the following information:

Recomendaciones según la capacidad de filtraje:

FGe < 30ml/min/1,73 m²

Nivel de riesgo
2

Ajuste de dosis
En caso de ERC, evitar tratamientos frecuentes o prolongados, aunque no se han realizado recomendaciones posológicas ya que se pue

Seguimiento clínico
"Durante tratamientos prolongados se aconsejan controles periódicos de los niveles de fosfato y calcio. La hipofosfatemia puede incremen general). Debido al contenido de magnesio en el almagato, se puede producir también una hipermagnesemia."

- FGe = 33: Mostra les recomanacions per a un FGe superior a 30

ALM 33 En caso de no seleccionar un FGe, saldrá todo

Muy bajo Bajo

ALMAGATO

Recomendaciones según la capacidad de filtraje:

FGe > 30ml/min/1,73 m2

Nivel de riesgo
1

Ajuste de dosis
En caso de ERC, evitar tratamientos frecuentes o prolongados, aunque no se han realizado recomendaciones posológicas ya que se pue

Seguimiento clínico
"Durante tratamientos prolongados se aconsejan controles periódicos de los niveles de fosfato y calcio. La hipofosfatemia puede incremen general). Debido al contenido de magnesio en el almagato, se puede producir también una hipermagnesemia."

3. Administrar compte d'usuari

Aquesta és la pantalla que ens permet accedir a totes les opcions del compte d'usuari.

1

Guia AP-IR Consultas Administrar cuenta Administración 5 6 Salir

Contraseña actual 2
Campo obligatorio

Nueva contraseña
Campo obligatorio 3

Confirmar nueva contraseña
Campo obligatorio

Modificar Contraseña

Nombre	Número de colegiado	Especialidad	Edad	Sexo	Provincia	Creada
4 admin	1234	Médico de atención primaria	3	Hombre	Alicante	2015-06-17

1. **Administrar compte:** és un tab de navegació que ens permet gestionar opcions del compte d'usuari.
2. **Contrasenya actual:** en cas de que vulguem canviar la contrasenya haurem d'introduir l'actual.

3. **Nova contrasenya:** son dos camps on haurem de escriure la nova contrasenya per duplicat.
4. **Informació del compte:** aquí es mostra informació del compte d'usuari.
5. **Administració:** és una tab de navegació sols visible per usuaris amb el rol d'administradors, on podrà gestionar parts de l'aplicació web, com la base de dades, altres usuaris, suggeriments... Ho veurem a continuació

Annex B: Manual d'administrador

4. Administració

És una pantalla sols accessible per a usuaris amb el rol d'administrador. A continuació descriurem els seus apartats.

4.1 Modificar la base de dades

Administración | Modificar Base de datos | Gestión de usuarios | Listado de mails | Sugerencias

1 Cargar principios activos desde excel
Seleccionar archivo | Ningún archivo seleccionado

2 Cargar Recomendaciones desde excel
Seleccionar archivo | Ningún archivo seleccionado

3 Borrar listado de principios activos y recomendaciones
Password de la cuenta de admi
Borrar

Principio activo

4 Nombre
ALMAGATO
CARBONATO CÁLCICO / COLECALCIFEROL
CINITAPRIDA

1. **Carregar principis actius:** l'administrador ha de seleccionar el fitxer de principis actius en format csv per tal de que l'aplicació el carregui a la base de dades. Sempre esborrar els continguts de la base de dades abans de fer una càrrega.
2. **Carregar recomanacions:** l'administrador ha de seleccionar el fitxer de recomanacions en format csv per tal de que l'aplicació el carregui a la base de dades. Sempre esborrar els continguts de la base de dades abans de fer una càrrega.
3. **Esborrar:** Esborra les taules de principis actius i recomanacions de la base de dades.
4. **Llistat de principis actius:** Aquest llistat ens permet veure si els principis actius estan carregats correctament a la base de dades.

4.2 Modificar el llistat de correus electrònics

Administración Modificar Base de datos Gestión de usuarios Listado de mails Sugerencias

Nombre de la cuenta

1

Nombre	Número de colegiado	Especialidad	Edad	Sexo	Provincia	Creada	Activada	Eliminada
admin	1234	Médico de atención primaria	3	Hombre	Alicante	2015-06-17	2015-06-17	
xavier_123@test.com	123	Médico de atención primaria	26	Hombre	Álava	2015-06-17	2015-06-17	

1. **Llistat d'usuaris:** Aquest llistat mostra tots els usuaris registrats a l'eina.

4.3 Modificar el llistat de correus electrònics

Administración Modificar Base de datos Gestión de usuarios Listado de mails Sugerencias

1 Cargar listado de dominios de mail desde excel
 Ningún archivo seleccionado

2 Borrar listado de dominios de mail
Password de la cuenta de admi

3 Texto a buscar

Dominio





@cgcom.es

@cgcom.org

@cmourense.org

1. **Carregar recomanacions:** l'administrador ha de seleccionar el fitxer de dominis web en format csv per tal de que l'aplicació el carregui a la base de dades. Sempre esborrar els continguts de la base de dades abans de fer una càrrega.
2. **Esborrar:** Esborra les taules de dominis web de la base de dades.
3. **Llistat de principis actius:** Aquest llistat ens permet veure tots els dominis web amb els que l'aplicació permet registrar als usuaris.

4.4 Suggestiments

Administración  Modificar Base de datos  Gestión de usuarios  Listado de mails  Sugerencias

Borrar listado de sugerencias

1 Password de la cuenta de admin

Borrar sugerencias

Texto a buscar

2 Envío Texto del mensaje

2015-06-21	test le10
------------	-----------

1. **Esborrar:** Esborra les taules de principis suggeriments de la base de dades.
2. **Llistat de suggeriments:** Aquest llistat ens permet veure tots els suggeriments que han enviat els usuaris.

Annex C: Guia de funcionament de l'API

AccountController

Receives all requests related to the accounts.

The possible inputs for requests are:

/basic-web-app/rest/accounts

GET

Returns a List of all accounts

@param **name** if this parameter isn't empty, return the account the matches this name

@return a List of accounts or a single account

/basic-web-app/rest/accounts

POST

Adds one account to the database

@param **sentAccount** the values that the created account will have

@return returns the values added to the created account

/basic-web-app/rest/accounts/{accountId}

GET

Returns the account with the given Id

@param **accountId** the given Id

@return the account with the given Id or a NOT_FOUND error

/basic-web-app/rest/accounts/name

POST

Returns the account that matches the given name

@param **accountName** the given name

@return the account with the given name or a NOT_FOUND error

/basic-web-app/rest/accounts/activated

GET

Returns a List of all activated accounts

@param **name** if this parameter isn't empty, return the activated account that matches this name

@return a List of activated accounts or a single activated account

/basic-web-app/rest/accounts/modifyPassword

POST

Modifies the given user password

@param **name** the user that will have its password modified

@param **password** the old password in MD5

@param **newPassword** the new password in MD5

@return returns OK or CONFLICT

MessageController

Receives all requests related to the messages the users can send as suggestions.

The possible inputs for requests are:

/basic-web-app/rest/message

POST

Adds one message to the database

@param **payload** the values that the created message will have

@return returns the created message

/basic-web-app /rest/message

GET

Returns a List of all the messages

@return a List of all the messages

/basic-web-app/rest/message/clearTable

POST

Clears all the email domains from the database

@param **name** the user name

@param **password** the user password in MD5

@return returns a message with OK or CONFLICT

EmailDomainController

Receives all requests related to the email domains the users can use to register.

The possible inputs for requests are:

/basic-web-app/rest/emailDomain

POST

Adds one emailDomain to the database

@param **payload** the values that the created emailDomain will have

@return returns the created emailDomain

/basic-web-app/rest/emailDomain

GET

Returns a List of all emailDomains

@return a List of all emailDomains

/basic-web-app/rest/emailDomain/excel

POST

Given the content of a csv file, creates all the emailDomains sent in the data.

@param **payload** the content of a csv file

@return returns HttpStatus.CONFLICT if there is an Error or HttpStatus.CREATED if there was no problem and everything was created

/basic-web-app/rest/emailDomain/clearTable

POST

If the user and password are in the database, and the user is an administrator, deletes all data from the emailDomain table

@param **name** user email

@param **password** user password in MD5

@return HttpStatus.CONFLICT if the password doesn't match,
HttpStatus.OK if everything went OK

ActiveIngredientController

Receives all requests related to the active ingredients the users can look recommendations for.

The possible inputs for requests are:

/basic-web-app/rest/activeIngredients

POST

Adds one activeIngredient to the database

@param **res** the values that the created activeIngredient will have

@return returns the created activeIngredient

/basic-web-app/rest/activeIngredients

GET

Returns a List of all activeIngredients

@return a List of activeIngredients

/basic-web-app/rest/activeIngredients/recomendations

GET

Returns a List of all activeIngredients with their recommendations

@return List of all activeIngredients with their recommendations

/basic-web-app/rest/activeIngredients/{activeIngredientId}

GET

Returns the activeIngredient with the given Id

@param **activeIngredientId** the given Id

@return the activeIngredient with the given Id or a NOT_FOUND error

/basic-web-app/rest/activeIngredients/ {activeIngredientId}/activeIngredient-recomendations

POST

Adds a recommendation to the activeIngredient with the given Id

@param **activeIngredientId** the given Id

@param **sentActiveIngredientRecomendation** all the information of the recommendation

@return the activeIngredientRecommendation created or a NOT_FOUND exception

/basic-web-app/rest/activeIngredients/ {activeIngredientId}/activeIngredient-recomendations

GET

Returns a list of the recommendations of the activeIngredient with the given Id

@param **activeIngredientId** the given Id

@return the list of recommendations or a NOT_FOUND exception

/basic-web-app/rest/activeIngredients/excel

POST

Given the content of a csv file, creates all the activeIngredients sent in the data.

@param **payload** the content of a csv file

@return HttpStatus.CONFLICT if there is an Error or HttpStatus.CREATED if there was no problem and everything was created

/basic-web-app/rest/activeIngredients/excelRecomendations

POST

Given the content of a csv file, creates all the recommendations sent in the data.

@param **payload** the content of a csv file

@return HttpStatus.CONFLICT if there is an Error or HttpStatus.CREATED if there was no problem and everything was created

/basic-web-app/rest/activeIngredients/clearTable

POST

If the user and password are in the database, and the user is an administrator, deletes all data from the activeIngredient and activeIngredientRecomendation table

@param **name** user email

@param **password** user password in MD5

@return HttpStatus.CONFLICT if the password doesn't match, HttpStatus.OK if everything went OK

ActiveIngredientRecomendationController

Receives all requests related to the recommendations for each active ingredient.

The possible inputs for requests are:

**/basic-web-app/rest/activeIngredient-recomendations/
{activeIngredientRecomendationId}**

GET

Returns the activeIngredientRecomendation with the given Id

@param **activeIngredientRecomendationId** the given Id

@return an activeIngredientRecomendation or HttpStatus.NOT_FOUND

**/basic-web-app/rest/activeIngredient-recomendations/
{activeIngredientRecomendationId}**

PUT

Updates a activeIngredientRecomendation with the new information sent in the request

@param **activeIngredientRecomendationId** the given Id

@param **sentActiveIngredientRecomendation** the new information for the activeIngredientRecomendation that has to be updated

@return the updated information for this activeIngredientRecomendation or HttpStatus.NOT_FOUND if there is no activeIngredientRecomendation for the given Id

Annex D: Guia per a futurs desenvolupadors

Per tal de poder executar aquest projecte és necessari disposar del codi font i haver instal·lat node.js. Es pot descarregar el executable a instal·lar de la pròpia web de node.js: <https://nodejs.org/download/> i seguir les instruccions que trobem allà mateix.

Un cop obert el projecte amb Netbeans, és necessari anar a la carpeta:

```
\src\main\webapp\app
```

I executar la comanda:

```
npm install
```

Aquesta comanda instal·larà totes les dependències del d'AngularJS.

També és necessari assegurar-nos de que Maven ha descarregat totes les dependències de Java, les quals es trobaran a la carpeta Dependències del projecte de Netbeans.

Un cop estiguin totes les dependències descarregades és necessari executar la comanda:

```
grunt watch
```

a la carpeta:

```
\src\main\webapp\app
```

Quan Grunt hagi acabat de preparar el web, serà necessari Construir el projecte des de Netbeans i executar-lo seleccionant la opció Run o Debug. Per tal de tenir accés a la base de dades serà necessari tenir MySQL funcionant (es recomana fer servir XAMPP [\[9\]](#)) al port localhost:3306 amb una base de dades de nom appdb.

