



Projecte de gestió d'entitats

MyEntities

Xavier Ribes Caelles
Grau d'Enginyeria Informàtica

Albert Grau Perisé
Santi Caballe Llobet

Gener del 2021



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

Llicències alternatives

A) Creative Commons:



Aquesta obra està subjecta a una llicència de [Reconeixement-NonComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NonComercial-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NonComercial 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2020 Xavier Ribes Caelles.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© Xavier Ribes Caelles

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Projecte de gestió d'entitats - My Entities</i>
Nom de l'autor:	<i>Xavier Ribes Caelles</i>
Nom del consultor:	<i>Albert Grau Perisé</i>
Data de lliurament (mm/aaaa):	<i>01/2021</i>
Àrea del Treball Final:	<i>Java EE</i>
Titulació:	<i>Grau d'Enginyeria Informàtica</i>
Resum del Treball	
<p>Analizant la situació actual recent, on lamentablement, s'han produït globalment milers de canvis en la situació contractual dels treballadors, es creu la possibilitat de desenvolupar una aplicació, que permeti en certa mesura, facilitar la gestió d'altres i baixes d'usuaris en les seves respectives empreses.</p> <p>Avui en dia encara existeixen petits i/o mitjans negocis, els quals no disposen de les eines informàtiques adequades per realitzar tasques que es podrien agilitzar mitjançant processos informàtics, per aquest motiu, es construirà una aplicació encarregada de cobrir aquest cicle de vida. Per fer-ho, caldrà estudiar quines seran les funcionalitats i/o requeriments bàsics que el sistema haurà de proveir, confeccionar un model de dades que permeti emmagatzemar la informació a tractar i implementar la lògica interna del sistema.</p> <p>Així doncs, l'elaboració d'aquest TFG pretén tan la confecció d'una aplicació basada en les funcionalitats anteriorment esmentades i que serà accessible mitjançant l'ús d'un navegador web, com l'estudi de tot el procés intrínsec que comporta dissenyar, implementar i documentar una aplicació.</p>	

Abstract

Analyzing the current world situation, where, unfortunately, there have been thousands of changes in the job position of workers, it is believed the possibility of developing an application, which, in some measure, facilitates the management of affiliations and terminations of users in their respective companies.

Today there are still small and medium sized businesses which do not have the appropriate computer tools to perform tasks that could be streamlined through computer processes. For this reason, an application that covers the lifecycle involved in this process will be built. To do this, it will be necessary to study the basic functionalities and requirements that the system must provide, draw up a data model to store the required information and develop the internal logic of the system.

Therefore, the elaboration of this 'TFG' aims to build an application (accessible through a web browser) based on the aforementioned functionalities and to study the process involved in designing, deploying and documenting an application.

Paraules clau:

JavaEE, Docker, Microservices, Spring, SpringBoot, Hibernate

Índex

1. Introducció	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball	2
1.2.1 Objectius personals	2
1.2.2 Objectius específics.....	3
1.3 Enfocament i mètode seguit	3
1.4 Planificació del Treball	5
2. Anàlisi	6
2.1 Model de casos d'ús	6
2.1.1 Login/Logout.....	6
2.1.2 Gestió d'usuaris.....	6
2.1.4 Gestió de permisos.....	8
2.1.5 Gestió d'aplicacions.....	8
2.1.5.1 Càrrega de comptes per aplicació	9
2.1.5.2 Càrrega de permisos per aplicació	10
2.2 Diagrama de classes	10
2.2 Diagrama de seqüència	11
3. Disseny	12
3.1 Arquitectura	12
3.1.1 Patró de disseny	12
3.2 Tecnologia aplicació web	13
3.2.1 Vue.js.....	13
3.3 Tecnologia servidor.....	15
3.3.1 JWT	15
3.3.2 Representational State Transfer (REST)	16
3.3.3 SpringBoot	17
3.3.4 JPA.....	18
3.4 Docker	18
3.5 Diagrama d'arquitectura	19
4. Prototip	20

5. Implementació	25
5.1 Instal·lació vue-cli	25
5.2 Creació de les vistes	26
5.3 Creació de les rutes	26
5.4 Publicació de l'aplicació.....	28
5.5 Creació servei autenticació	28
6. Conclusions	30
Bibliografia	32
Annex	33

Llista de figures

Figura 1. Diagrama de Gantt	5
Figura 2. Model conceptual	10
Figura 3. Exemple diagrama seqüència	11
Figura 4. Patró MVC	12
Figura 5. Estructura projecte Vue.js	14
Figura 6. JWT token request	15
Figura 7. JWT token resource access	16
Figura 8. REST interaction flow	16
Figura 9. Arxiu configuració SpringBoot	17
Figura 10. Exemple classe SpringBoot	18
Figura 11. Exemple DockerFile	19
Figura 13. Prototip pantalla general	20
Figura 14. Prototip menú lateral	20
Figura 15. Prototip pantalla usuaris.....	21
Figura 16. Prototip pantalla nou usuari.....	21
Figura 17. Prototip pantalla llistat accounts.....	22
Figura 18. Prototip pantalla nova compta.....	22
Figura 19. Prototip pantalla llistat de permisos.....	23
Figura 20. Prototip pantalla afegir nou permís	23
Figura 21. Prototip pantalla llistat aplicacions	24
Figura 22. Prototip pantalla nova aplicació.....	24
Figura 23. Prototip pantalla login.....	25
Figura 24. Exemple codificació vistes	26
Figura 25. Exemple fitxer routes.....	27
Figura 26. Exemple route guard	27

1. Introducció

1.1 Context i justificació del Treball

La informàtica ha estat des de els seus inicis una tecnologia la qual neix per modernitzar, automatitzar i/o facilitar processos els quals requereixen càlculs complexos, repetir tasques, o bé comunicar-se amb altres sistemes. Per aquest motiu, es creu la necessitat de desenvolupar una aplicació i/o sistema informàtic que permeti millorar processos i tasques que es realitzen de forma sistemàtica o manual.

Les empreses tan de l'índole que directament ens ocupa (IT), com d'altres sectors, tenen la necessitat de gestionar el conjunt d'informació dels seus usuaris, comptes, rols i aplicacions a les que tenen accés. Aquest manteniment, en molts casos, sol resultar difícil, no només pel fet de donar d'alta i/o baixa al personal de l'empresa, sinó per les diferents eines en que l'equip de recursos humans ha d'estar familiaritzat.

Al cercar en la web programes de gestió d'entitats, sovint ens apareixen solucions les quals no s'adapten totalment a les nostres necessitats, són complexes, requereixen d'una formació i per tan s'ha de superar una corba d'aprenentatge o bé, per tal de solucionar la nostra problemàtica acabem pagant per mòduls i/o aplicacions excessivament costosos. Aquesta situació, ens obra la porta a desenvolupar una aplicació totalment gratuïta, es a dir, programari lliure. La aplicació permetrà gestionar de forma efectiva i mitjançant una interfície gràfica molt senzilla, les entitats 'User', 'Account', 'Permission' i 'Application'. Cadascuna d'elles tindrà una funcionalitat ben diferenciada i en els propers apartats s'en detallaran les seves propietats.

La confecció d'aquest projecte, en el marc personal i professional, resulta especialment atractiu, primerament per l'auto-disciplina que un treball d'aquestes dimensions requereix i en el qual un mateix ha de gestionar-se horaris, rutines d'estudi, recerca d'informació, etc... Així com també i tenint en compte l'àmbit acadèmic per l'oportunitat que l'elaboració d'aquests tipus de treballs et permeten aprofundir en àrees i/o assignatures que t'hagin despertat curiositat, ja sigui pel contingut exposat, com per l'entusiasme en que el professorat ha desenvolupat la seva professió.

En segon lloc, en l'àmbit professional, tampoc he treballat mai en projectes d'aquesta índole, ja que gairebé el cent per cent de la meva experiència laboral s'ha centrat en desenvolupar i/o mantenir aplicacions les quals ja estaven confeccionades i gairebé sempre, construïdes sobre tecnologies més antigues, com JSP i Struts, etc...

No obstant, en aquest TFG el punt de partida es totalment diferent, ja que en no disposar d'un software ja provat i desplegat, ens trobem el repte d'enfrontar-nos a preguntes i/o problemes els quals ara mateix encara no sabem com respondre.

1.2 Objectius del Treball

Seguint les pautes marcades en la documentació de l'assignatura, els objectius generals d'aquesta, són:

1. Capacitat de comunicació escrita en l'àmbit acadèmic i professional.
2. Ús i aplicació de les TIC en l'àmbit acadèmic i professional.
3. Capacitat de comunicació en llengua estrangera.
4. Treball en equip.
5. Capacitat per adaptar-se a les tecnologies i als futurs entorns actualitzant les competències professionals.
6. Capacitat per innovar i generar noves idees.

Un cop realitzat el treball, aquestes, en més o menys mesura haurien de ser assolides. No obstant, tot i que ens serviran per establir una base sobre el que s'espera, passem a determinar els objectius personals i específics d'aquest TFG.

1.2.1 Objectius personals

1. Documentar i gestionar un projecte de la mida d'un TFG.
2. Estudiar i analitzar en profunditat aplicacions desenvolupades en la tecnologia Java.
3. Obtenir experiència en un camp altament sol·licitat.
4. Aprendre a treballar amb diferents eines i/o tecnologies i saber com enllaçar-les.
5. Poder avaluar amb objectivitat el resultat obtingut.

1.2.2 Objectius específics

1. Poder analitzar i dissenyar de forma completa el desenvolupament d'un sistema informàtic.
2. Planificar i distribuir la càrrega de treball envers al temps disponible.
3. Obtenir un document de qualitat que recopil·li la metodologia emprada al realitzar aquest treball.
4. Obtenir una aplicació i/o producte que satisfaci els requeriments plantejats.

1.3 Enfocament i mètode seguit

Per tal de realitzar aquest TFG s'opta per desenvolupar un producte totalment nou, ja que tot i que en la Web existeix molta documentació i/o projectes que segueixen arquitectures i/o tecnologies similars a les que es volen utilitzar, no es l'objectiu d'aquest treball ampliar i/o adaptar un producte, sinó com ja s'ha comentat anteriorment, dissenyar i documentar tot el procés intrínsec que suposa la creació d'una aplicació, investigació i desplegament d'un sistema d'informació d'aquestes característiques.

En qualsevol cas, puntualitzar que tot aquell coneixement obtingut mitjançant la cerca e investigació d'informació ja sigui provinent de pàgines web, com de projectes existents sobre aplicacions construïdes en el llenguatge de programació Java, serviran per establir una guia d'actuació en aquest treball.

En línies generals, no s'entrarà en tecnicismes en aquest apartat, tot i això podem discernir una serie de tasques per tal d'estructurar i finalment realitzar el treball.

1. Analitzar el conjunt de funcionalitats i requeriments que l'aplicació ha de satisfer. A grans trets, l'aplicació haurà de permetre:
 1. Realitzar l'alta, modificació i baixa d'usuaris.
 2. Realitzar l'alta i baixa de comptes.
 1. Relacionar una compta amb el seu usuari.
 3. Realitzar l'alta i baixa de permisos.
 1. Assignar permisos a un usuari/aplicació
 4. Realitzar l'alta, modificació i baixa d'aplicacions.
 1. Obtenir un llistat de les comptes existents en la aplicació configurada.
 2. Crear noves comptes desde MyEntities a l'aplicació configurada.

2. Determinar el conjunt d'eines (software) més adients per confeccionar el producte.
 1. **SpringToolSuite4**: entorn de desenvolupament per realitzar les tasques de programació amb Java i enfocat específicament a Spring.
 2. **Lombok**: plugin que permet estalviar-se la codificació dels getters i setters per aquelles classes més senzilles mitjançant l'ús d'anotacions.
 3. **JPA**: framework per la persistència de dades en Java sobre bases de dades relacionals.
 4. **Spring**: framework de desenvolupament d'aplicacions Java que segueix el model IoC (injecció de dependències).
 5. **Apache Maven**: software per gestionar llibreries, dependències i realitzar el compilament i empaquetatge de l'aplicació.
 6. **VisualStudioCode**: entorn de desenvolupament de la part client.
 7. **Vue.js**: framework de JavaScript per construir ràpidament interfícies d'usuari, es basa en la re-utilització de components i la reactivitat.
 8. **Vuetify**: llibreria utilitzada sobre Vue.js per facilitar el 'prototipatge' de pantalles.
 9. **MySQL**: sistema gestor de base de dades relacional.
 10. **Docker**: eina per al desplegament d'aplicacions dins dels anomenats contenidors de software.
 11. **GitHub**: eina de control de versions que s'utilitzarà per emmagatzemar tot el programari desenvolupat.
3. Implementar els diferents components Java que conformaran el nucli de l'aplicació.
4. Dissenyar una interfície d'usuari que permeti consumir els serveis exposats.
5. Provar i/o testejar individualment i globalment tots els elements desenvolupats.
6. Documentar transversalment cadascuna de les fases seguides.

Tot i el fet de treballar individualment, al llarg de tot el projecte es seguirà una metodologia de desenvolupament àgil, on cadascuna de les etapes/tasques anteriorment descrites, es construiran mitjançant iteracions i refinament.

1.4 Planificació del Treball

La planificació d'aquest treball seguirà les dates claus emmarcades en l'àmbit de l'assignatura que en realitza el seguiment.

Per tal de distribuir la feina acordament a la durada, es divideix el projecte en cinc fases, que queden detalls a continuació:

1. Pla de projecte
 1. Resum de la proposta
 2. Justificació de l'interès i rellevància de la proposta
 3. Explicació de la motivació personal
2. Anàlisi i disseny
 1. Model de casos d'ús
 2. Disseny relacional de la BBDD
 3. Diagrama de classes principals
 4. Diagrama d'arquitectura
3. Implementació
 1. Implementació del producte
4. Memòria
 1. Preparació de la memòria
 2. Confecció de la presentació
5. Defensa Virtual - Tribunal TFG
 1. Defensa davant el tribunal

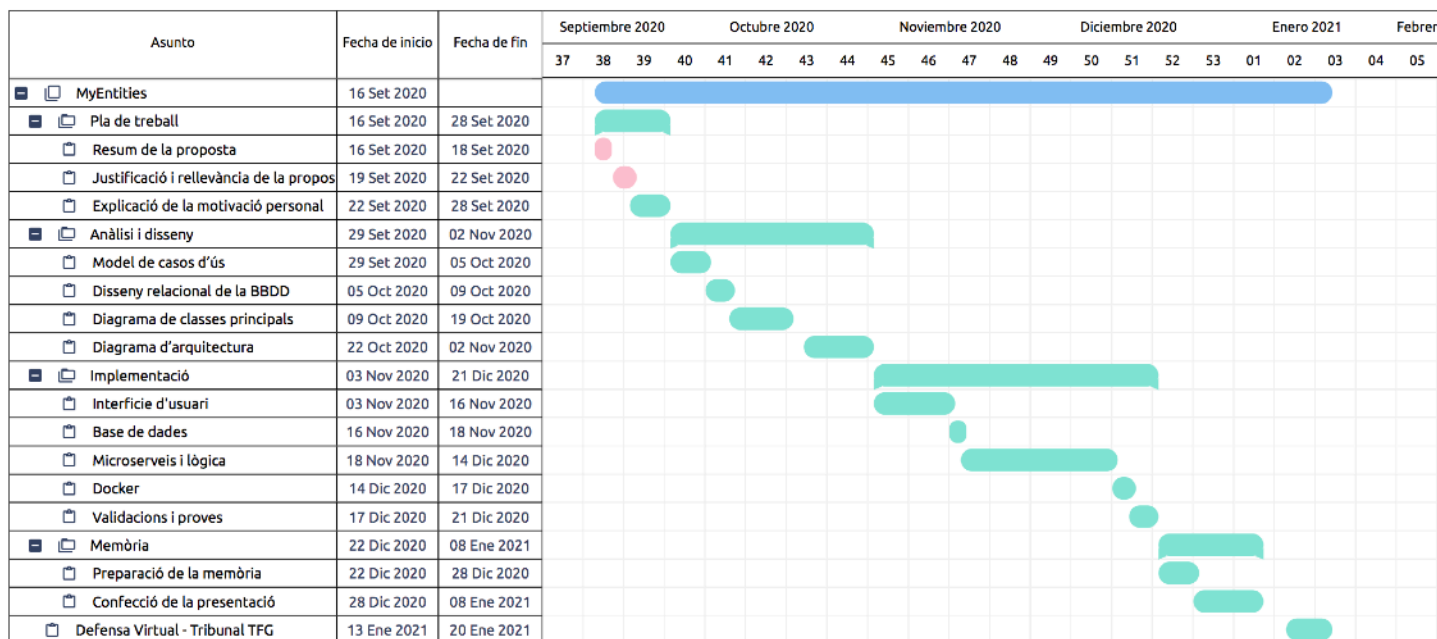


Figura 1. Diagrama de Gantt

2. Anàlisi

En tot projecte de desenvolupament de programari existeix una fase d'anàlisi i disseny, on detallar, tan els casos d'ús dels actors principals, com també tots aquells diagrames i/o seqüències que puguin ser d'utilitat alhora d'implementar l'aplicació.

En aquest apartat es tractarà l'anteriorment comentat, així com s'aprofundirà en l'arquitectura de l'aplicació.

2.1 Model de casos d'ús

2.1.1 Login/Logout

Login/Logout	
Descripció	<ul style="list-style-type: none">· Permet autenticar-se al sistema mitjançant l'us de credencials (usuari/contrasenya)· Permet finalitzar la sessió actual de l'usuari connectat
Actors	Administrador,Usuari registrat
Casos d'ús relacionats	Cap
Pre-condició	Es disposa d'un usuari per accedir al sistema
Post-condició	<ul style="list-style-type: none">· El sistema autentica l'usuari· El sistema finalitza la sessió d'usuari

2.1.2 Gestió d'usuaris

Gestió d'usuaris	
Descripció	Alta, modificació, baixa d'usuaris
Actors	Administrador,Usuari registrat
Casos d'ús relacionats	2.1.1 Login
Pre-condició	Es disposa d'un usuari per accedir al sistema
Post-condició	El procés d'alta, modificació i/o baixa d'usuaris finalitza correctament.

Gestió d'usuaris	
Procés	<p>Alta d'usuaris</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'usuaris i mitjançant el botó d'afegir (+) es mostra la fitxa d'alta d'usuaris · S'introdueixen les dades obligatòries (nom, cognoms, contrasenya) i es guarda el formulari · El sistema guarda l'usuari introduït <p>Modificació d'usuaris</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'usuaris i mitjançant el botó d'edició (llapis) es mostra la fitxa amb la informació de l'usuari · S'introdueixen les dades a modificar i es guarda el formulari · El sistema modifica les dades de l'usuari <p>Baixa d'usuaris</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'usuaris i es prem el botó d'esborrar (paperera) · Es mostra un missatge de confirmació sobre l'eliminació · El sistema esborra el registre seleccionat
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.1.3 Gestió de comptes

Gestió de comptes	
Descripció	Alta i baixa de comptes
Actors	Administrador,Usuari registrat
Casos d'ús relacionats	2.1.1 Login 2.1.5 Gestió d'aplicacions
Pre-condició	Es disposa d'un usuari per accedir al sistema Es disposa d'una aplicació creada en el sistema
Post-condició	El procés d'alta i/o baixa de comptes finalitza correctament
Procés	<p>Alta de comptes</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla de comptes i mitjançant el botó d'afegir (+) es mostra la fitxa de comptes · S'introdueixen les dades obligatòries (nom, descripció, usuari i aplicació) i es guarda el formulari · El sistema guarda la compta introduïda · L'usuari sincronitza la compta i aquesta es crea en el sistema extern <p>Baixa de comptes</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla de comptes i es prem el botó d'esborrar (paperera) · Es mostra un missatge de confirmació sobre l'eliminació · El sistema esborra el registre seleccionat
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.1.4 Gestió de permisos

Gestió de permisos	
Descripció	Assignació/Anul·lació de permisos
Actors	Administrador, Usuari registrat
Casos d'ús relacionats	2.1.1 Login 2.1.5 Gestió d'aplicacions
Pre-condició	Es disposa d'un usuari per accedir al sistema Es disposa d'una aplicació creada en el sistema
Post-condició	El procés d'assignació/anul·lació de permisos finalitza correctament
Procés	Assignació de permisos <ul style="list-style-type: none">· S'accedeix a la pantalla de permisos· Es consulta el rol a assignar· Es selecciona l'usuari que disposarà del permís· Es confirmen els canvis Anul·lació de permisos <ul style="list-style-type: none">· S'accedeix a la pantalla de permisos· Es consulta el rol a revocar· S'elimina l'usuari el qual ja no disposarà d'aquest permís· Es confirmen els canvis
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.1.5 Gestió d'aplicacions

Gestió d'aplicacions	
Descripció	Alta, modificació i baixa d'aplicacions
Actors	Administrador
Casos d'ús relacionats	2.1.1 Login
Pre-condició	Es disposa d'un usuari administrador per accedir al sistema
Post-condició	El procés d'alta, modificació i/o baixa d'aplicacions finalitza correctament

Gestió d'aplicacions	
Procés	<p>Alta d'aplicacions</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'aplicacions i mitjançant el botó d'afegir (+) es mostra la fitxa d'alta d'usuaris · S'introdueixen les dades obligatòries (dades de connexió, configuració de camps) i es guarda el formulari · El sistema guarda l'aplicació introduïda <p>Modificació d'aplicacions</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'aplicacions i mitjançant el botó d'edició (llapis) es mostra la fitxa amb la informació de l'aplicació · S'introdueixen les dades a modificar i es guarda el formulari · El sistema modifica les dades de l'aplicació <p>Baixa d'aplicacions</p> <ul style="list-style-type: none"> · S'accedeix a la pantalla d'aplicacions i es prem el botó d'esborrar (paperera) · Es mostra un missatge de confirmació sobre l'eliminació · El sistema esborra el registre seleccionat
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.1.5.1 Càrrega de comptes per aplicació

Càrrega d'accounts per aplicació	
Descripció	Un cop configurada una aplicació, s'ha de poder obtenir el llistat d'usuaris existents en l'aplicació destí
Actors	Administrador
Casos d'ús relacionats	2.1.1 Login 2.1.5 Gestió d'aplicacions
Pre-condició	Es disposa d'un usuari administrador per accedir al sistema
Post-condició	El procés finalitza correctament i es pot comprovar que els usuaris s'han carregat
Procés	<ul style="list-style-type: none"> · S'accedeix a la pantalla d'aplicacions · Es prem el botó 'Load application accounts' · El procés confirma l'execució del mateix
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.1.5.2 Càrrega de permisos per aplicació

Càrrega de permisos per aplicació	
Descripció	Un cop configurada una aplicació, s'ha de poder obtenir el llistat de perfils existents en l'aplicació destí
Actors	Administrador
Casos d'ús relacionats	2.1.1 Login 2.1.5 Gestió d'aplicacions
Pre-condició	Es disposa d'un usuari administrador per accedir al sistema
Post-condició	El procés finalitza correctament i es pot comprovar que els permisos s'han carregat
Procés	<ul style="list-style-type: none"> · S'accedeix a la pantalla d'aplicacions · Es prem el botó 'Load application roles' · El procés confirma l'execució del mateix
Alternatives i excepcions	El sistema informa sobre qualsevol error/excepció que hagi pogut succeir

2.2 Diagrama de classes

En aquest punt es mostra el diagrama UML corresponent al model conceptual de l'aplicació. En el diagrama es representen les entitats de dades que hauran de ser gestionades en l'aplicació.

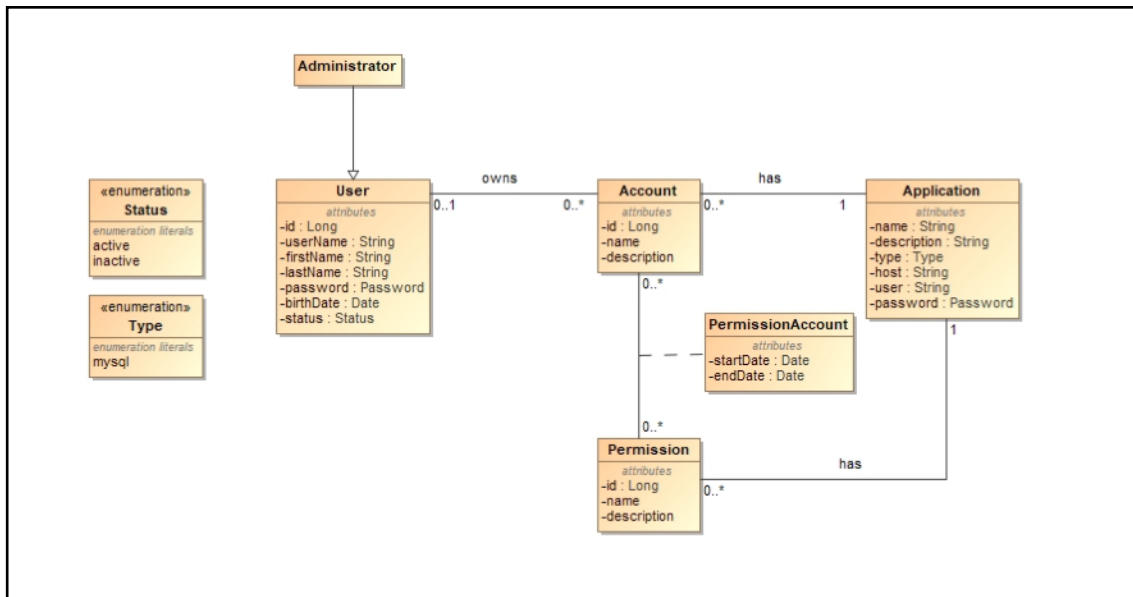


Figura 2. Model conceptual

2.2 Diagrama de seqüència

A continuació s'exposa un dels diagrames de seqüència, on a nivell de lògica, més processos hi participen.

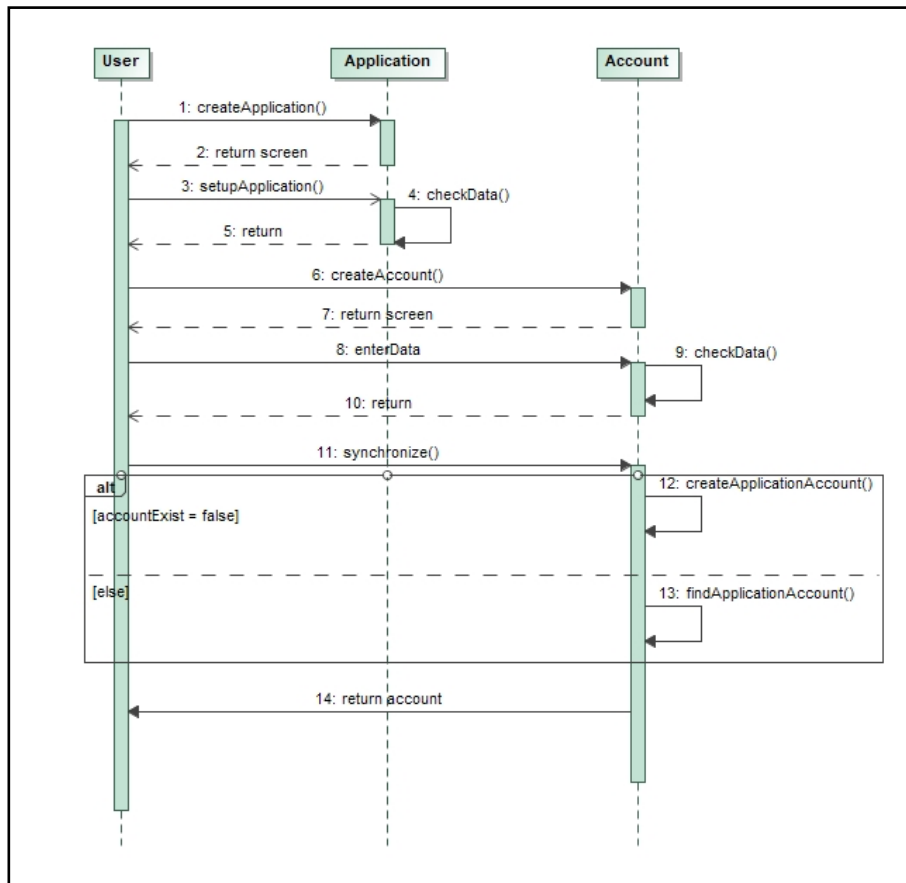


Figura 3. Exemple diagrama seqüència

3. Disseny

3.1 Arquitectura

Per realitzar el desenvolupament de l'aplicació s'ha decidit seguir una arquitectura per capes, on cadascuna d'elles tindrà una funcionalitat ben diferenciada. El patró per excel·lència que encapsula aquest comportament és l'anomenat Model-Vista-Controlador (MVC); aquest patró es caracteritza per delegar en tres components principals, la responsabilitat de controlar les diferents capes de l'aplicació.

3.1.1 Patró de disseny

A continuació s'en resumeixen les característiques de cadascuna.

- **Model:** Gestiona tots els accessos a la informació amb la que el sistema opera (consulta, actualitzacions, insercions) i envia a la vista la informació sol·licitada.
- **Controlador:** Actua com a intermediari entre la vista i el model, ja que és l'encarregat de respondre als events (normalment invocats per l'usuari) i realitzar peticions al model quan es sol·licita informació.
- **Vista:** Defineix els components visuals necessaris ('User Interface') per a que l'usuari pugui accedir a la informació continguda en el model.

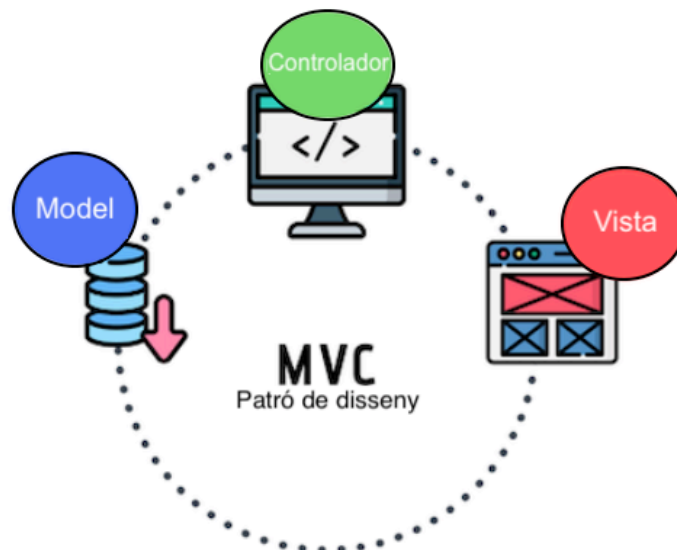


Figura 4. Patró MVC

Per implementar cadascuna de les capes exposades anteriorment tal com ja s'havia exposat al pla de treball, s'ha decidit utilitzar les següents tecnologies.

3.2 Tecnologia aplicació web

Actualment existeixen una varietat de tecnologies enfocades exclusivament al desenvolupament d'aplicacions web, el que es coneix com *frontend*, aquí podem trobar un conjunt de llibreríes i *frameworks* per realitzar qualsevol tipus d'aplicació i/o funcionalitat que es necessiti.

Aquesta diversitat, ha dificultat en certa mesura seleccionar-n'he la més adequada; per aquest motiu, el criteri per decantar-se per una envers totes les altres ha estat, per una banda, el desig d'estudiar tecnologies que s'escapin del model adoptat en els darrers anys (JSP, Java ServerFaces), que tot i resultar solucions molt vàlides, potser es van quedant enrere; per l'altra, l'objectiu alhora de dissenyar aquest 'frontal' web serà aconseguir el desacoblament de la part servidor, per tal d'evitar el 'render' de les pàgines; el que tècnicament s'anomena aplicacions SPA (*Single Page Application*).

Per aquest motiu, finalment, s'ha acabat escollint la llibreria *Vue.js* ja que es tracta, d'una de les llibreries basades en *JavaScript* i de codi lliure més noves, la qual es caracteritza per presentar una corba d'aprenentatge progressiva, basada en la definició i re-utilització de components i que alhora consta d'una creixent comunitat de suport.

3.2.1 Vue.js

Per dissenyar l'aplicació web, es seguirà un dels estàndards utilitzats en projectes que utilitzen *frameworks* orientats a l'utilització de components web. Bàsicament, es dividirà el projecte segons els recursos, fomentant una estructura per seccions.

Seguint la terminologia anglesa, aquestes seran:

- Components: Encarregats d'encapsular funcionalitats completes de l'aplicació, es a dir, contenen tan la vista, com la interacció i lògica.

- Mixins: Parts de codi javascript que es re-utilitzat en diferents components.

·Router: Actua com a controlador enllaçant la vista a una URL, alhora defineix l'accés a les diferents pàgines envers els permisos de l'usuari.

·Store: Comunment utilitzat en aplicacions reactives per gestionar la informació que s'emmagatzema en una caché global.

·Utils: Funcions genèriques que s'utilitzen en certs components, com per exemple formatejos de data, filtres, etc...

·Views: Conté els components que son accessibles des de el router, es a dir, les pàgines que seran visitades per consultar el model de dades.

En l'aplicació dissenyada, aquesta podria ser una distribució dels components:

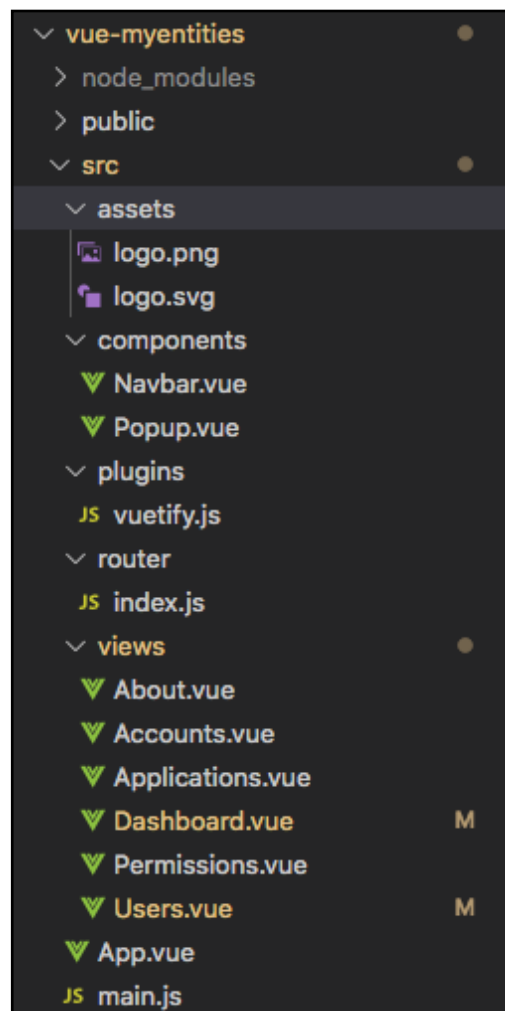


Figura 5. Estructura projecte Vue.js

3.3 Tecnologia servidor

Un cop s'ha vist una de les parts de la nostra aplicació client-servidor, faltaria per definir com s'implantarà la part servidor i que serà l'encarregada tan de gestionar les peticions d'entrada com de sortida.

En aquesta definició s'haurà de tindre en compte que, donat que es vol aconseguir un alt nivell de desacoblament, s'evitarà l'ús de sessions així com de variables emmagatzemades en el context d'usuari que puguin comportar fallides i/o comportaments inesperats.

Una de les tecnologies per donar solució a aquesta problemàtica, passa per fer ús de les anomenades aplicacions *stateless*. D'aquesta forma cada petició que es realitza, conté la informació necessària per poder accedir a les dades de manera inequívoca i independent d'altres crides. Això s'aconsegueix mitjançant un token, obtingut al realitzar el login a l'aplicació i que serà l'encarregat de determinar si l'usuari té accés o no a un recurs determinat.

3.3.1 JWT

En l'aplicació MyEntities utilitzarem la implementació proporcionada per JWT per cobrir aquesta necessitat.

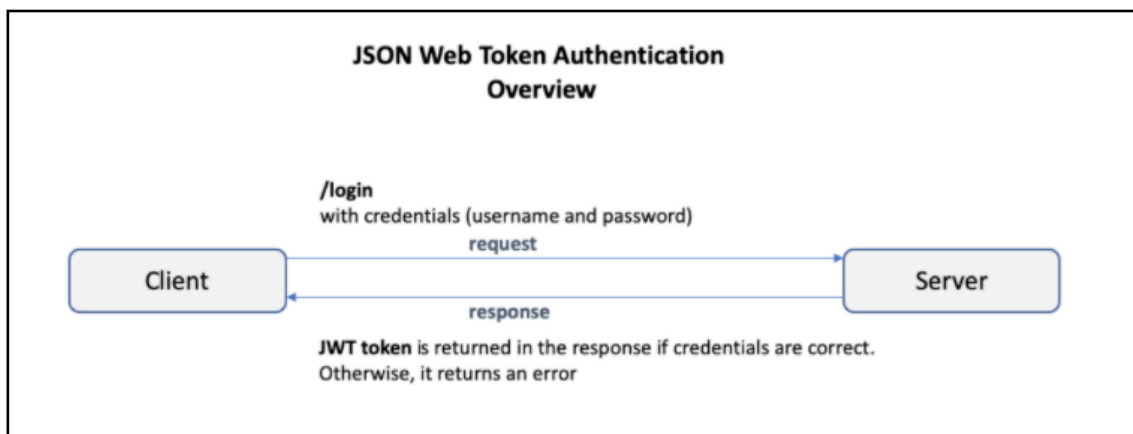


Figura 6. JWT token request

Com s'observa, es realitza una petició amb el nom d'usuari i contrasenya, si aquesta es correcta es genera un token que s'utilitzà en futures peticions per poder accedir als recursos protegits, en cas contrari es retornarà un error i no se'n permetrà l'accés.

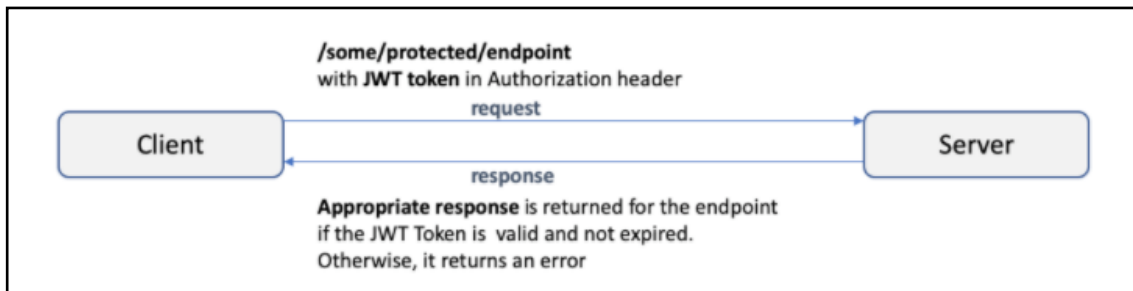


Figura 7. JWT token resource access

3.3.2 Representational State Transfer (REST)

En aquest punt s'ha esmentat breument, com es manipularan les diferents entitats que conformen l'aplicació, aquestes, seran accessibles en l'arquitectura client-servidor, mitjançant la definició de recursos. Cada recurs, farà referència a una entitat en concret i es podrà interactuar amb cadascun d'ells mitjançant l'estàndard REST; aquest, defineix un conjunt d'operacions GET, POST, PATCH, DELETE sobre el protocol de comunicació HTTP.

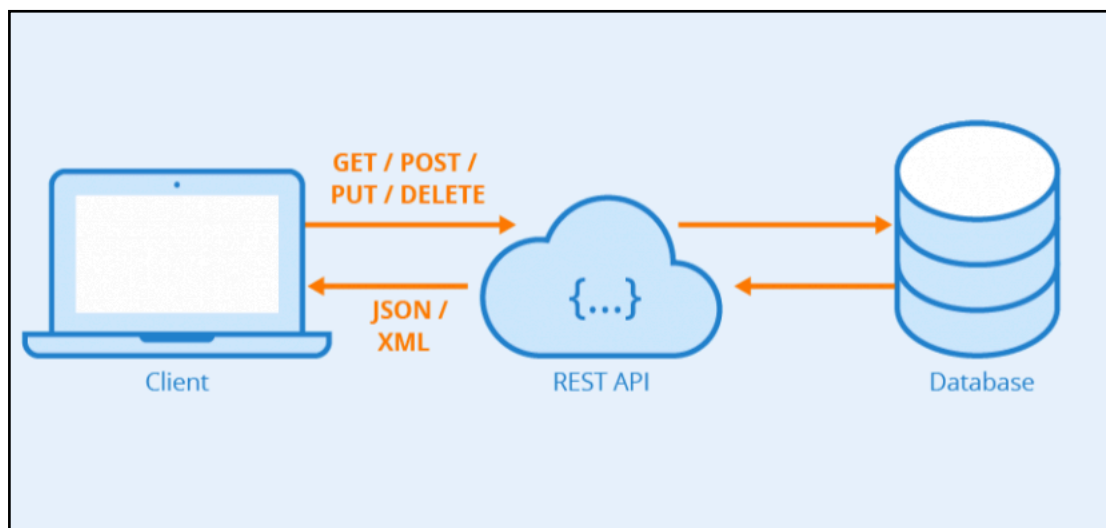


Figura 8. REST interaction flow

3.3.3 SpringBoot

L'encarregat de capturar totes les peticions HTTP que arribin al servidor, serà una aplicació dissenyada en el framework SpringBoot; aquest ens facilitarà l'operativa alhora d'implementar aplicacions basades en Spring, ja que seleccionant un conjunt de dependències ja serà possible disposar d'una aplicació totalment configurada i preparada per ser desplegada en un servidor d'aplicacions, i en aquest darrer cas, fins i tot no farà falta definir-n'he cap, ja que SpringBoot també disposa d'un servidor d'aplicacions integrat (embedded).

Per tant, a nivell de configuració s'indicarà únicament les dades de connexió a la base de dades i el port on es desplegarà el servei.

```
spring.application.name=myentities
spring.datasource.url=jdbc:mysql://docker-mysql-myentities:3306/MyEntities
spring.datasource.username=root
spring.datasource.password=root
server.port=10222
lombok.equalsAndHashCode.callSuper = call

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
# Keep the connection alive if idle for a long time (needed in production)
spring.datasource.testWhileIdle = true
spring.datasource.validationQuery = SELECT 1

# Show or not log for each sql query
spring.jpa.show-sql = true

# Hibernate ddl auto (create, create-drop, update)
spring.jpa.hibernate.ddl-auto = update

# Naming strategy
spring.jpa.hibernate.naming-strategy = org.hibernate.cfg.ImprovedNamingStrategy
```

Figura 9. Arxiu configuració SpringBoot

Com s'ha comentat, per manipular les dades contingudes en la base de dades, es definiran un conjunt de funcions que respondran depenent del mètode HTTP invocat. El framework Spring defineix un seguit d'operacions que serviran per traduir cada petició HTTP en els corresponents mètodes JAVA; en la documentació del framework es pot trobar tota la informació relativa a aquestes.

Per exemplificar el funcionament d'aquest conjunt de llibreries es pot observar la següent classe 'UserController.java', on mitjançant les anotacions `@RequestMapping("/user")` i `@GetMapping` s'indica que totes les peticions *HTTP* realitzades al recurs 'user' i sempre i quan l'operació sigui *GET* cridin el mètode `getUsers()`.

```
@RestController
@RequestMapping("/user")
public class UserController {

    private static final Logger log = LoggerFactory.getLogger(UserController.class);

    @Autowired
    private UserService userService;

    @Autowired
    private UserMapper userMapper;

    @GetMapping
    public List<User> getUsers() throws ResourceNotFoundException {
        log.info("Getting all users");
        List<User> users = userService.getAll().stream().map(userEntity -> userMapper.userEntityToUser(userEntity))
            .collect(Collectors.toList());

        return users;
    }
}
```

Figura 10. Exemple classe SpringBoot

El fet de separar l'accés i/o manipulació de les diferents entitats i/o recursos en serveis molt especialitzats, ens encamina cap a una arquitectura de micro-serveis. Tot i no disposar de diferents màquines per tal de distribuir els diferents serveis, s'estudiarà com aquest tipus de desenvolupament facilita l'escalament horitzontal.

3.3.4 JPA

El *framework* que interaccionarà amb el sistema gestor de base de dades *MySQL* serà l'API estàndard de Java, JPA (Java Persistence Api). No es contempla l'ús d'*Hibernate* ja que a nivell de consultes, no es preveu una complexitat elevada.

3.4 Docker

Docker és una eina de programari lliure que permet automatitzar el desplegament d'aplicacions, dins dels anomenats contenidors de software. A diferència de les antigues màquines virtuals, no es necessari instal·lar un sistema operatiu, ja que mitjançant els recursos que ofereix el propi kernel, es capaç de crear individualment processos completament aïllats d'altres serveis del sistema.

Per desplegar l'aplicació dins un contenidor, serà necessari crear una imatge de la mateixa, partint d'un fitxer Dockerfile; en ella s'indicarà l'imatge oficial principal, encarregada d'executar el .jar de la REST API i el port on escoltarà el contenidor en temps d'execució.

```
FROM java:8
VOLUME /tmp
EXPOSE 8080
ADD /target/myentities-1.0.0-SNAPSHOT.jar myentities-1.0.0-SNAPSHOT.jar
ENTRYPOINT ["java", "-jar", "myentities-1.0.0-SNAPSHOT.jar"]
```

Figura 11. Exemple DockerFile

A banda d'aquesta imatge i donada la facilitat que ofereix Docker, el servei que oferirà accés a una base de dades MySQL també es crearà d'aquesta forma. D'aquesta manera, en cas que es vulgui migrar l'aplicació a un entorn cloud, l'arquitectura serà capaç de suportar-ho.

3.5 Diagrama d'arquitectura

Un cop explicats els components principals de l'arquitectura de l'aplicació, es detalla com queden relacionats.

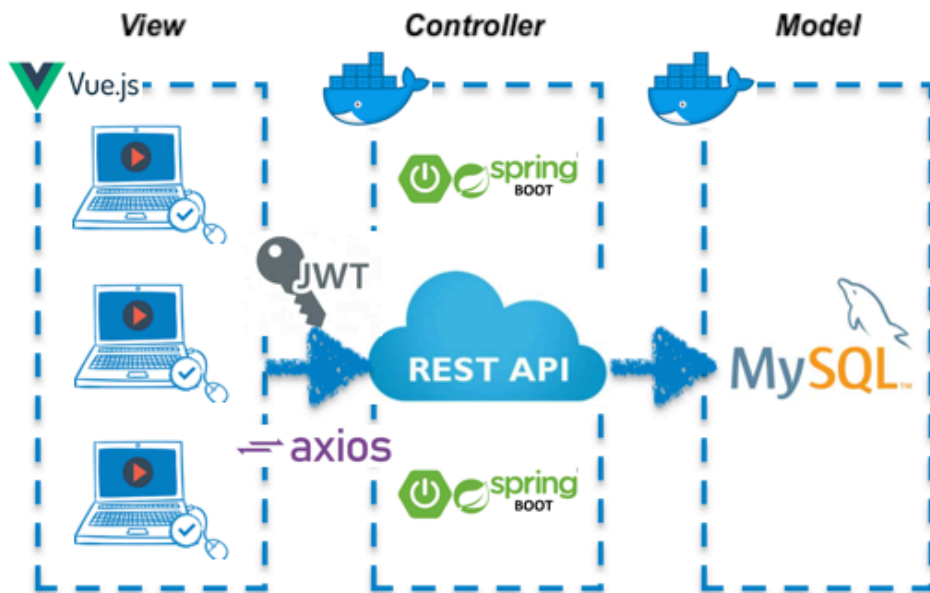


Figura 11. Diagrama d'arquitectura

4. Prototip

La interfície d'usuari de l'aplicació presentarà un menú lateral, on l'usuari podrà accedir a les diferents pantalles a les que hi té accés.

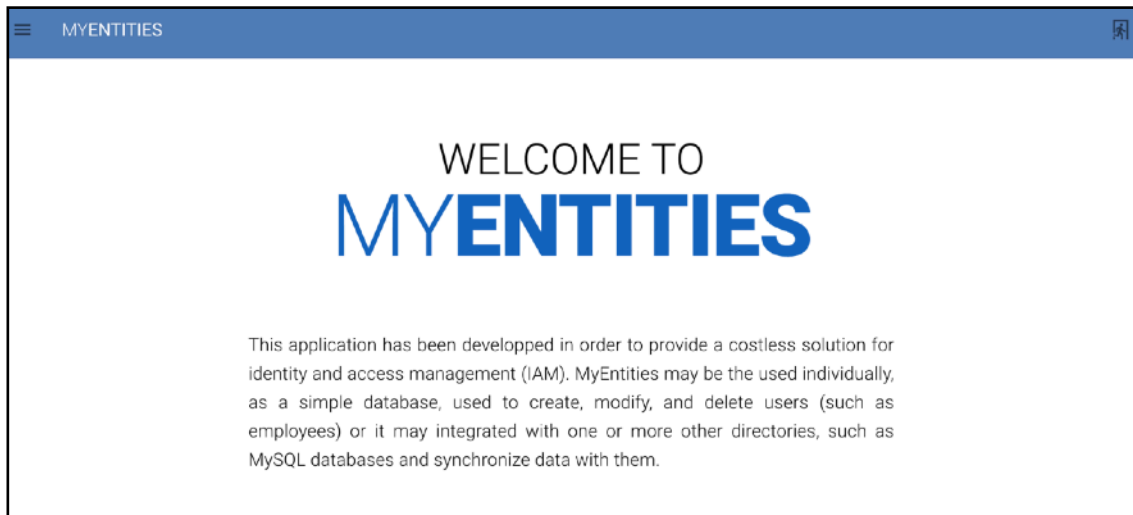


Figura 13. Prototip pantalla general

En clicar sobre l'ícona superior esquerra es desplegaran les opcions de menú.

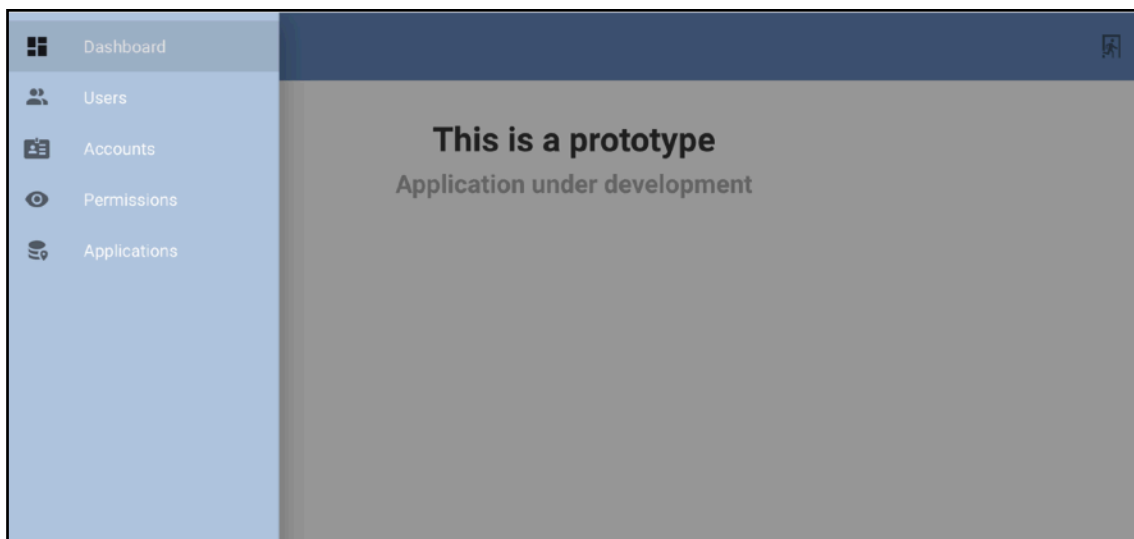


Figura 14. Prototip menú lateral

El disseny de cadascuna de les pantalles de mestres seguirà el mateix format, es presentarà un llistat central amb els registres existents i sobre els mateixos, es podrà seleccionar l'acció a realitzar.

En aquest cas, s'exemplifica les pantalles per les identitats usuari, compta, permís i aplicació.

UserName	FirstName	LastName	BirthDate	Status	Action
XRIBES	XAVIER	RIBES	09/06/1990	ACTIVE	
GRIBES	GEMMA	RIBES	19/08/1987	ACTIVE	
JRIBES	JERONI	RIBES	06/09/1962	ACTIVE	
MCAELLES	MAR	CAELLES	01/03/1964	ACTIVE	

Figura 15. Prototip pantalla usuaris

Per afegir qualsevol nou registre, independentment del objecte (user, account, application, permission), apareixerà un popup mostrant les dades a emplenar.

Add new user

Username _____

Password _____

Firstname _____ Lastname _____

Birthdate
01/11/2020 _____

CANCEL SAVE

Figura 16. Prototip pantalla nou usuari

Llistat comptes (accounts)

The screenshot shows a web interface for 'MYENTITIES'. At the top, there is a blue header with a menu icon and the text 'MYENTITIES'. Below the header, there is a section titled 'Accounts' with a '+ Add' button on the right. The main content is a table with the following data:

Name	Description	User	Application	Action
ACCOUNT_TEST	Account for testing	UNKNOWN	MyEntities Database	

At the bottom of the table, there is a pagination control showing 'Rows per page: 10' and '1-1 of 1' with navigation arrows.

Figura 17. Prototip pantalla llistat accounts

Afegir una nova compta

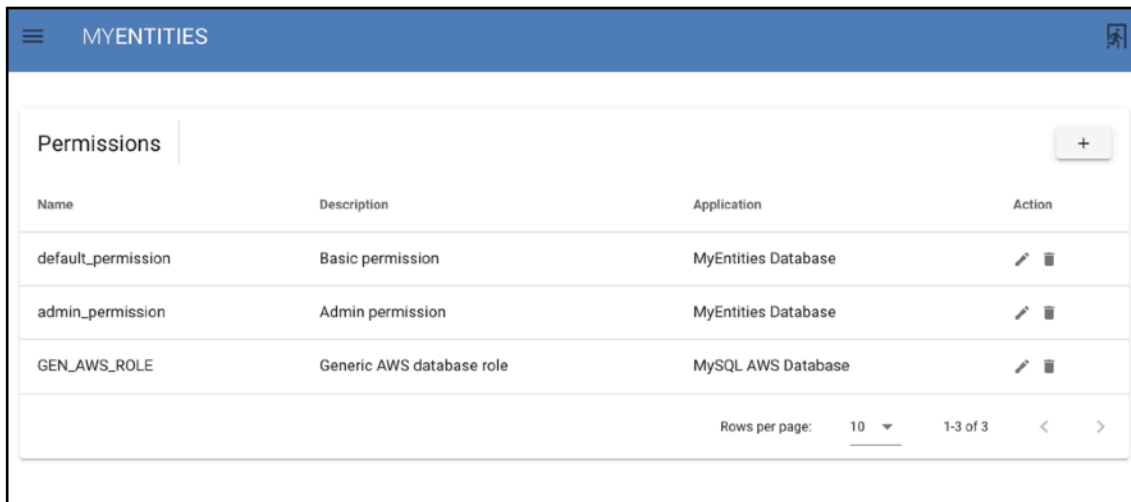
The screenshot shows a modal form titled 'Add new account' overlaid on the 'Accounts' list page. The form has a 'BASIC' tab selected. The fields are:







- Name:
- Description:
- Application:
- User:

At the bottom right of the form, there are two buttons: 'CANCEL' and 'SAVE'.

Figura 18. Prototip pantalla nova compta

Llistat permisos

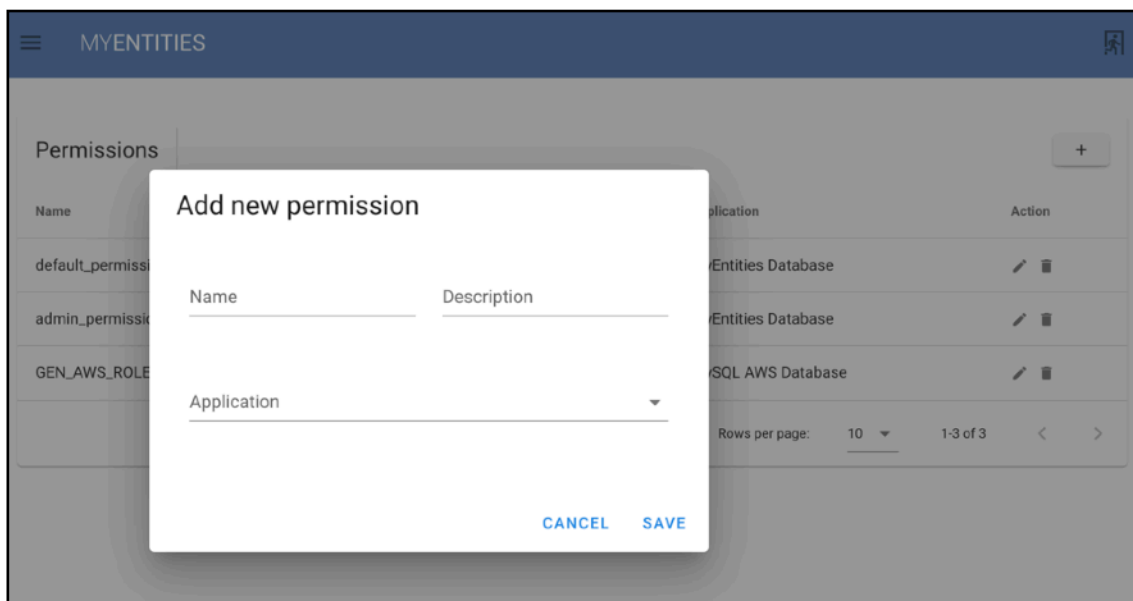


Name	Description	Application	Action
default_permission	Basic permission	MyEntities Database	 
admin_permission	Admin permission	MyEntities Database	 
GEN_AWS_ROLE	Generic AWS database role	MySQL AWS Database	 

Rows per page: 10 1-3 of 3

Figura 19. Prototip pantalla llistat de permisos

Afegir un nou permís



Add new permission

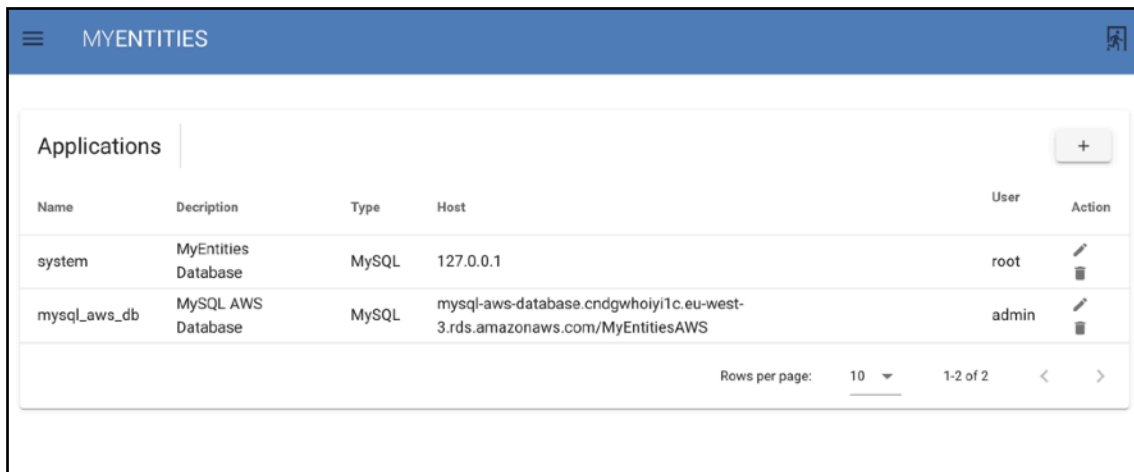
Name _____ Description _____

Application _____

CANCEL SAVE

Figura 20. Prototip pantalla afegir nou permís

Llistat aplicacions



The screenshot shows a web interface for managing applications. At the top, there is a blue header with the text 'MYENTITIES' and a hamburger menu icon on the left, and a user profile icon on the right. Below the header, the main content area is titled 'Applications' and features a table with columns for Name, Description, Type, Host, User, and Action. There are two entries in the table. The first entry is 'system' with description 'MyEntities Database', type 'MySQL', host '127.0.0.1', and user 'root'. The second entry is 'mysql_aws_db' with description 'MySQL AWS Database', type 'MySQL', host 'mysql-aws-database.cndgwhoiyi1c.eu-west-3.rds.amazonaws.com/MyEntitiesAWS', and user 'admin'. Each entry has edit and delete icons in the Action column. At the bottom right of the table, there is a pagination control showing 'Rows per page: 10' and '1-2 of 2' with navigation arrows.





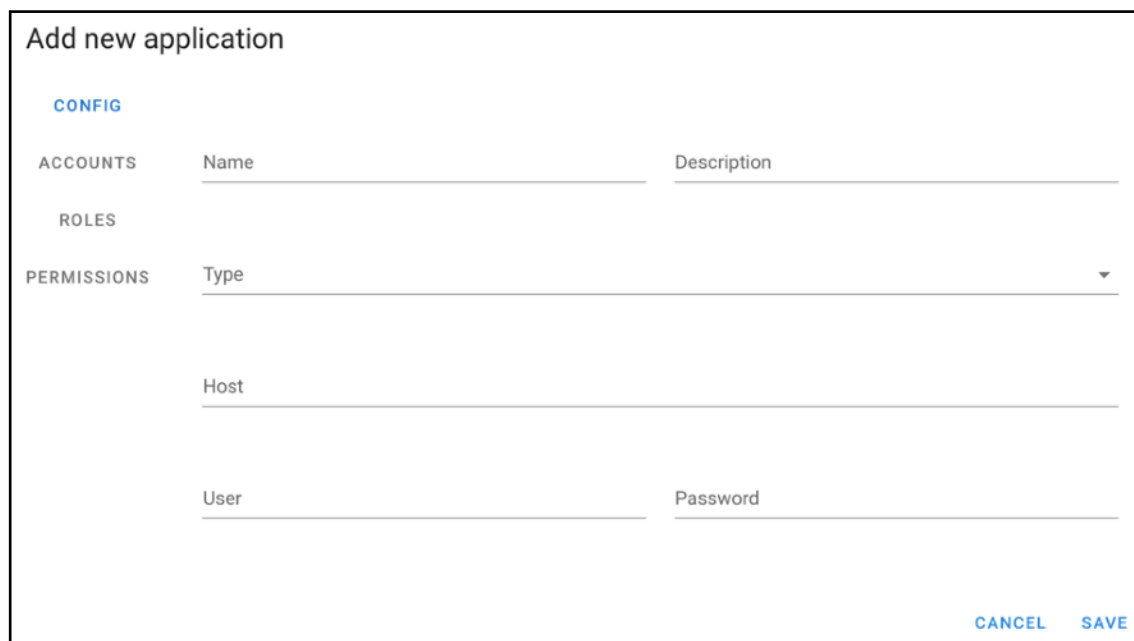
Name	Description	Type	Host	User	Action
system	MyEntities Database	MySQL	127.0.0.1	root	 
mysql_aws_db	MySQL AWS Database	MySQL	mysql-aws-database.cndgwhoiyi1c.eu-west-3.rds.amazonaws.com/MyEntitiesAWS	admin	 

Figura 21. Prototip pantalla llistat aplicacions

Afegir una nova aplicació



The screenshot shows a form titled 'Add new application'. The form is organized into sections: 'CONFIG' (highlighted in blue), 'ACCOUNTS', 'ROLES', and 'PERMISSIONS'. Under 'ACCOUNTS', there are two input fields: 'Name' and 'Description'. Under 'ROLES', there is a dropdown menu for 'Type'. Under 'PERMISSIONS', there are three input fields: 'Host', 'User', and 'Password'. At the bottom right of the form, there are two buttons: 'CANCEL' and 'SAVE'.

Figura 22. Prototip pantalla nova aplicació

Accés a l'aplicació (Login)

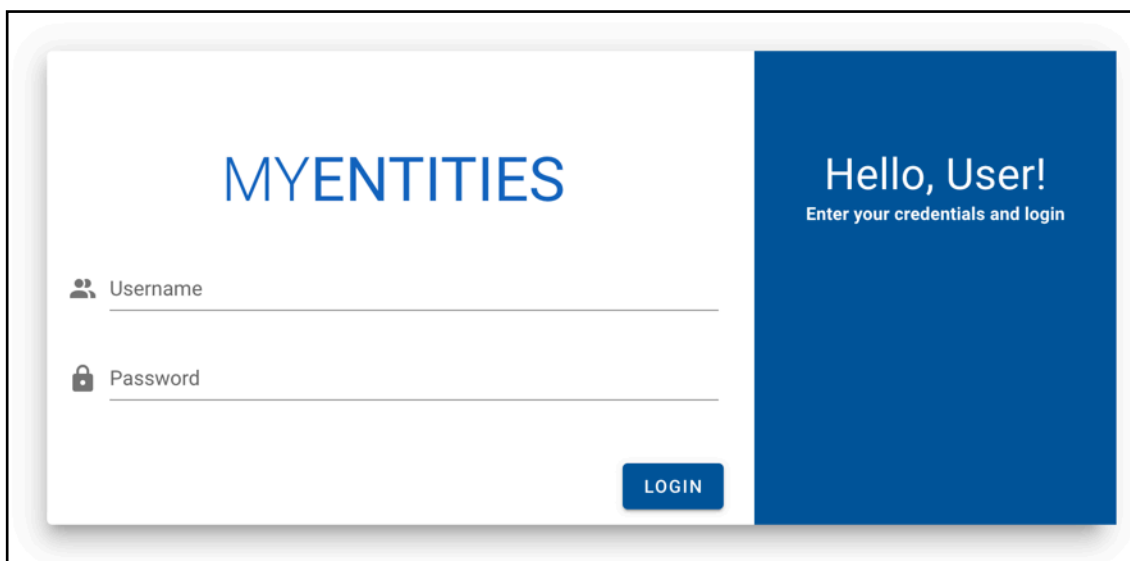


Figura 23. Prototip pantalla login

5. Implementació

5.1 Instal·lació vue-cli

El `vue-cli` (es a dir, la “interfície de línia de comandes Vue”) es una eina creada per l'equip Vue.js per facilitar el ràpid desenvolupament de les aplicacions Vue. Tot i que Vue.js es pot utilitzar directament des d'un navegador, la majoria de projectes, utilitzen aquesta eina per tal d'administrar i gestionar les diferents biblioteques i/o dependències de l'aplicació.

Primerament, ens haurem d'assegurar de disposar de Node.js, ja que serà l'entorn d'execució sobre el qual correrà Vue.js.

Un cop instal·lat, mitjançant la utilitat 'npm' (node package management) s'afegiran els paquets de 'vue-cli'.

```
$ npm install -g @vue/cli
```

A continuació, amb la següent comanda es crearà un projecte i automàticament se'ns mostrarà un conjunt de llibreries per configurar ràpidament l'estructura del projecte.

```
$ vue create name_of_project
```

5.2 Creació de les vistes

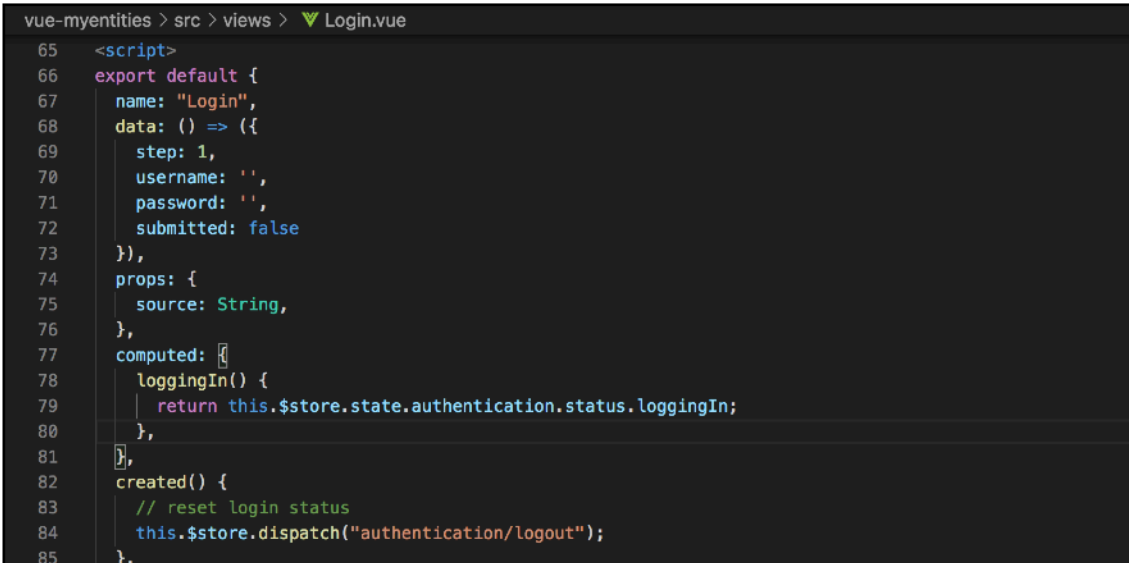
Les vistes contenen el codi HTML de l'aplicació; la particularitat de Vue.js així com d'altres frameworks per al disseny d'aplicacions web, es l'utilització de la sintaxis template basada en HTML i que permet vincular de forma declarativa el DOM renderitzat a les dades de la instància de Vue subjacent.

Per exemple:

```
<span>Message: {{ msg }}</span>
```

En el següent fragment de codi HTML, l'etiqueta del '*mustache*' es substituirà amb el valor de la propietat *msg*.

En el directori `/src/views` es trobaran les diferents vistes les quals contenen la lògica de la capa de presentació. En cadascuna d'elles, es realitzarà la crida corresponent per tal d'obtenir el llistat d'usuaris, comptes, permisos i/o aplicacions així com també per crear, modificar o esborrar dades.



```
vue-myentities > src > views > Login.vue
65 <script>
66 export default {
67   name: "Login",
68   data: () => ({
69     step: 1,
70     username: '',
71     password: '',
72     submitted: false
73   }),
74   props: {
75     source: String,
76   },
77   computed: {
78     loggingIn() {
79       return this.$store.state.authentication.status.loggingIn;
80     },
81   },
82   created() {
83     // reset login status
84     this.$store.dispatch("authentication/logout");
85   },
86 }
```

Figura 24. Exemple codificació vistes

5.3 Creació de les rutes

Les rutes són un dels principals elements en qualsevol projecte Vue.js ja que permeten la navegabilitat entre pàgines segons la crida HTTP processada. Vue-Router es la llibreria oficial que implementa aquestes funcionalitats.

Per afegir aquesta dependència s'executarà la comanda:

```
$ vue add router
```

En el directori /src/_helpers trobem els components definits per la navegabilitat

```
vue-myentities > src > _helpers > JS router.js > router.beforeEach() callback
1  import Vue from 'vue'
2  import VueRouter from 'vue-router'
3  import Dashboard from '../views/Dashboard.vue'
4  import Users from '../views/Users.vue'
5  import Accounts from '../views/Accounts.vue'
6  import Permissions from '../views/Permissions.vue'
7  import Applications from '../views/Applications.vue'
8  import Login from '../views/Login.vue'
9
10 Vue.use(VueRouter)
11
12 const routes = [
13   {
14     path: '/',
15     name: 'Dashboard',
16     component: Dashboard
17   },
18   {
19     path: '/login',
20     name: 'Login',
21     component: Login
22   },
23 ]
```

Figura 25. Exemple fitxer routes

Es important remarcar que donada la necessitat d'autoritzar cadascuna de les peticions/accessos als recursos protegits (*backend*), s'ha implementat l'anomenat route-guard, el qual comprova en cadascuna de les peticions (accés a la pantalla / user, /account/, etc...) si l'usuari disposa d'un token JWT vàlid.

```
router.beforeEach((to, from, next) => {
  // redirect to login page if not logged in and trying to access a restricted page
  const publicPages = ['/login'];
  const authRequired = !publicPages.includes(to.path);
  const loggedIn = localStorage.getItem('user');

  if (authRequired && !loggedIn) {
    return next('/login');
  }

  next();
})
```

Figura 26. Exemple route guard

5.4 Publicació de l'aplicació

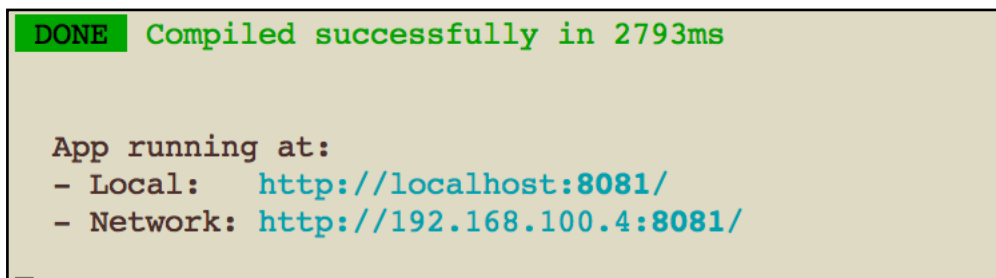
Per instal·lar les dependències del projecte així com per desplegar l'aplicació localment, s'executarà la següent comanda, dins el directori del projecte.

```
$ npm install
```

i seguidament

```
$ npm run serve
```

Si l'execució ha estat satisfactòria, s'observarà el següent missatge



```
DONE Compiled successfully in 2793ms

App running at:
- Local: http://localhost:8081/
- Network: http://192.168.100.4:8081/
```

Figura 27. Exemple desplegament satisfactori

5.5 Creació servei autenticació

Com s'ha definit en el anàlisi i disseny de l'aplicació, els diferents endpoints exposats des de l'aplicació SpringBoot, quedaran protegits mitjançant un token JWT.

Per la implementació d'aquest, s'ha utilitzat el mòdul *spring-security*; el qual mitjançant un conjunt de classes que actuen com a filtres de les peticions HTTP capturades, es capaç d'autenticar i autoritzar l'usuari que sol·licita accés. Dins el paquet 'uoc.project.myentities.security' es troba tota la lògica relativa a aquesta funcionalitat.

S'ha de tenir en compte la configuració del 'Control d'accés HTTP (CORS)' ja que per defecte, aquest, no exposa en les respostes, totes les capçaleres que s'han generat. De la mateixa forma, també serà important configurar-ho de cara a posar l'aplicació en un entorn productiu sobre diferents dominis.

Request

```
$ curl --location --request POST 'http://localhost:10222/login' \
\
--header 'Content-Type: application/json' \
--header 'Cookie: JSESSIONID=8D7F161ABDA1B7D616BD928845FF54DB' \
```

```
--data-raw '{
  "userName":"admin",
  "password":"123456"
}'
```

Response

```
{
  "Authorization": "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbIsImV4cCI6MTYwOTcxNTc4N30.vtbcApdQy_6DfzTLPF6UHVZZmfUCUpEmIWki_WGkMI7hs5anF8i71SzqIR5ELsE2l28nEKf454awgqeYzRteBg"
}
```

Figura 28. Exemple resposta JWT

5.6 Creació imatge / publicació container

Per crear i desplegar l'aplicació spring-boot dins un docker container, primerament, serà necessari compilar el projecte mitjançant *Maven* amb l'instrucció 'mvn clean install' el qual crearà l'arxiu .jar.

Tot seguit es crearà la imatge docker mitjançant les instruccions especificades al fitxer Dockerfile del projecte.

Figura 29. Creació imatge JAVA docker

```
((base) Xavier-MacBook-Pro:myentities xribes13$ docker build -f Dockerfile -t myentities_app .
Sending build context to Docker daemon 46.66MB
Step 1/5 : FROM java:8
--> d23bdf5b1b1b
Step 2/5 : VOLUME /tmp
--> Using cache
--> 979d4ecb7482
Step 3/5 : EXPOSE 8080
--> Using cache
--> f41e4e012cfa
Step 4/5 : ADD /target/myentities-1.0.0-SNAPSHOT.jar myentities-1.0.0-SNAPSHOT.jar
--> Using cache
--> 1fc8f496ed70
Step 5/5 : ENTRYPOINT ["java","-jar","myentities-1.0.0-SNAPSHOT.jar"]
--> Using cache
--> 215dbdleaecd
Successfully built 215dbdleaecd
Successfully tagged myentities_app:latest
(base) Xavier-MacBook-Pro:myentities xribes13$
```

Per desplegar la base de dades MySQL es descarregarà la imatge oficial desde el Docker Repository i s'executarà un script .sql per definir-h'he les taules i relacions.

```
((base) Xavier-MacBook-Pro:2020-2021 xribes13$ docker run -d -p 6033:3306 --name=docker-mysql-myentities --env="MYSQL_ROOT_PASSWORD=root" --env="MYSQL_PASSWORD=root"
--env="MYSQL_DATABASE=MyEntities" mysql:latest
4a3ab6f48efeb9cc9d262c019e5b524c3e48835b135fa738280b67e737352289
(base) Xavier-MacBook-Pro:2020-2021 xribes13$ docker exec -i docker-mysql-myentities mysql -uroot -proot MyEntities < myentities_db_script.sql
```

Figura 30. Creació imatge MySQL

Finalment es relaciona el contenidor MySQL amb la imatge docker JAVA.

```
(base) Xavier-MacBook-Pro:myentities xribes13$ docker run --name "myentities_app" -t --link docker-mysql-myentities:mysql -p 10222:10222 myentities_app

:: Spring Boot ::
(92.3.4.RELEASE)

2021-01-05 00:14:47.339 INFO 1 --- [main] u.p.myentities.MyEntitiesApplication : Starting MyEntitiesApplication v1.0.0-SNAPSHOT on 12896fa68853 with PID
1 (/myentities-1.0.0-SNAPSHOT.jar started by root in /)
2021-01-05 00:14:47.351 INFO 1 --- [main] u.p.myentities.MyEntitiesApplication : No active profile set, falling back to default profiles: default
2021-01-05 00:14:51.859 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFERRED mode.
2021-01-05 00:14:52.240 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 34ms. Found 6 JPA repository
y interfaces.
2021-01-05 00:14:54.497 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 10222 (http)
2021-01-05 00:14:54.544 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-01-05 00:14:54.545 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.38]
2021-01-05 00:14:54.731 INFO 1 --- [main] o.s.c.e.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
```

Figura 31. Desplegament imatge dins un contenidor

6. Conclusions

En aquest últim apartat i per posar punt final al treball realitzat durant aquests últims mesos, analitzarem de forma objectiva tant els resultats obtinguts en el desenvolupament del projecte, com personalment, l'experiència, en gran part, enriquidora que suposa implementar un treball de final de grau.

Tenint en compte les fites inicials proposades, podem dir que aquestes s'han completat de forma satisfactòria, tot i això, durant l'evolució del projecte així com de cadascuna de les iteracions de programació realitzades i a mesura que els nostres coneixements sobre tecnologies, que fins aquest moment eren desconegudes (Vue.js, Docker, etc...) avançaven, hem anat detectant millores i/o altres funcionalitats que s'haguessin pogut implementar. Per posar exemples sobre aquests casos, quan expira un token d'autenticació JWT, l'aplicació no realitza el 'logout' de l'usuari sinó que el deixa dins l'aplicació però sense poder realitzar accions sobre la base de dades (consultar, escriure, modificar i/o esborrar).

La gestió d'errors també es un aspecte a millorar, ja que tot i que a nivell d'aplicació queda tot enregistrat al log, faltaria notificar a l'usuari, de forma individualitzada cadascun dels diferents errors.

Pel que fa al desplegament de l'aplicació, m'hagués agradat disposar de més temps, per investigar l'automatització de projectes realitzats amb Docker, ja que existeix tot un món en aquest sector; com s'ha dit, ens em limitat a desplegar de forma manual contenidors de l'aplicació, però faltaria veure temes com augmentar-n'he l'escalabilitat, definició de fitxers .yaml per treballar amb Docker Composer, etc...

A títol personal escriure aquestes línies em suposa deixar enrere mesos, setmanes, dies, en definitiva, hores, moltes hores dedicades a la superació d'un mateix. En aquest sentit, valoro molt positivament els coneixements adquirits al llarg d'aquest temps, així com també tota la inversió realitzada durant anys. He de confessar que tal com s'acostuma a dir 'hi ha moments per tot', així ha estat; no obstant, avui ha arribat el dia assenyalat al calendari i tantes emocions no s'haurien de descriure amb paraules; tan sols em queda agrair a la UOC per posar els mitjans i recursos per fer això possible, així com també a tot el personal docent d'aquesta universitat. Gràcies.

Bibliografia

- [1] «Spring» 29 09 2020. [En línia]. Available: <https://spring.io/>
- [2] «Vue.js» 29 09 2020. [En línia]. Available: <https://vuejs.org/>
- [3] «Vuetify.js» 29 09 2020. [En línia]. Available: <https://vuetifyjs.com/>
- [4] «Vue.js» 29 09 2020. [En línia]. Available: <https://en.wikipedia.org/wiki/Vue.js>
- [5] «Maven» 29 09 2020. [En línia]. Available: <https://maven.apache.org/>
- [6] «Node.js» 29 09 2020. [En línia]. Available: <https://nodejs.org/es/>
- [7] «Docker» 29 09 2020. [En línia]. Available: <https://www.docker.com/>
- [8] «STS» 29 09 2020. [En línia]. Available: <https://spring.io/tools>
- [9] «codepen.io» 09 11 2020. [En línia]. Available: <https://codepen.io/>
- [10] «stackoverflow» 09 11 2020. [En línia]. Available: <https://stackoverflow.com/questions/49477230/get-component-to-wait-for-asynchronous-data-before-rendering>
- [11] «jasonwatmore» 10 11 2020. [En línia]. Available: <https://jasonwatmore.com/>
- [12] «Fontawesome» 10 11 2020. [En línia]. Available: <https://fontawesome.com/>
- [13] «Vue-router» 22 11 2020. [En línia]. Available: <https://router.vuejs.org>
- [14] «Deploying a Vue.js app to Heroku» 29 09 2020. [En línia]. Available: <https://medium.com/better-programming/deploying-a-vue-js-app-to-heroku-d16f95c07a04>
- [15] «Unsplash» 22 11 2020. [En línia]. Available: <https://unsplash.com/s/photos/wallpaper?color=white>
- [16] «AmazonRDS» 05 12 2020. [En línia]. Available: <https://eu-west-3.console.aws.amazon.com/rds/home?region=eu-west-3#GettingStarted>

Annex

Annex A : Glosari

A

Apache

És un servidor HTTP de pàgines web de codi obert multiplataforma desenvolupat per Apache Software Foundation.

B

Backend

Part lògica d'una aplicació, executada sobre uns servidor d'aplicacions, encarregada de controlar l'accés i la gestió sobre la base de dades.

C

CORS

Cross-Origin Resource Sharing, és un mecanisme que permet l'intercanvi de recursos protegits des de una pàgina web en un domini diferent del que ha efectuat la petició.

F

Framework

Entorn de treball que proporciona un sistema robust i segur, format per llibreries i/o components estables.

Frontend

Part d'una aplicació encarregada de gestionar la interfície gràfica de l'usuari, fent possible la visualització de les dades i la interacció amb la mateixa.

M

MVC

L'Arquitectura Model-Vista-Controlador és un patró de disseny utilitzat per a la implementació d'interfícies d'usuari, el qual divideix l'aplicació en tres parts: model de dades, interfície usuari i lògica de control.

MySql

És un sistema de gestió de bases de dades relacional desenvolupat sota llicència doble Llicència pública general / Llicència comercial per Oracle Corporation i es considera com la base de dades de codi obert més popular del món.

S

SPA

Single Page Application, fa referència a les aplicacions web dissenyades en una sola pàgina amb el propòsit de donar una experiència més fluïda als usuaris.

V

Visual Studio

És un entorn de desenvolupament integrat (IDE) de Microsoft. S'utilitza per desenvolupar programes informàtics, així com llocs web, aplicacions web, serveis web i aplicacions mòbils.

Vue.js

És un framework JavaScript de codi obert model de vista-vista-model per construir interfícies d'ús i aplicacions d'una sola pàgina.

Annex B : Lliurables del projecte

Memòria del projecte

Document actual on es detalla les motivacions del projecte, planificació del treball, anàlisi, disseny, implementació i conclusions obtingudes.

Codi font

S'entreguen dos fitxers amb extensió .zip, els quals contenen, per una banda, el projecte frontend realitzat amb Vue.js i per l'altra, l'API que actua com a backend, implementada en springboot.

Presentació en PowerPoint

Presentació del projecte amb PowerPoint, resumint les parts més importants de l'aplicació, així com motivacions i resultats obtinguts.

Fulla d'auto-avaluació

Document d'avaluació de la feina realitzada durant l'elaboració del projecte.

Vídeo presentació

Document audio-visual on es presenta el PowerPoint així com l'accés i funcionalitats bàsiques de l'aplicació MyEntities.