

Implementació d'un nou gestor de terminologia: TermUAB

Roger Gomà Roca

Resum—Actualment, a l'assignatura de Terminologia aplicada a la traducció que s'implementa a la facultat de traducció i interpretació de la UAB, s'utilitza una aplicació del TERMCAT per la gestió terminològica, i permet que els alumnes i professors realitzin les pràctiques d'aquesta assignatura. Aquest gestor de terminologia és de gran utilitat per la gestió de terminologia i docència però aquest sistema s'està quedant obsolet i té unes quantes inconsistències que fan que el sistema no sigui gaire competent en l'àmbit de la docència o per als professionals. És per això que, amb aquest projecte, es pretén realitzar un nou sistema que sigui capaç de millorar totes aquestes inconsistències i aplicar millores que serien útils per als usuaris.

Paraules clau—Terminologia, MERN, SPA, núvol, test unitaris, NodeJs, ReactJs

Abstract—Nowadays, in the traduction and interpretation school, they uses an application of TERMCAT for terminology management that allows students and teachers to carry out some projects of the subjects. This terminology management is very useful for managing terminology and teaching but this system is becoming obsolete and has a few inconsistencies that make the system are not very competent in the field of teaching or professionals. That is why it is intended to permorm a new system that can improve all of these inconsistencies and implement improvements that would be useful for users.

Index Terms— Terminology, MERN, SPA, cloud, unit testing, NodeJs, ReactJs



1 INTRODUCCIÓ

AQUEST projecte neix a causa dels problemes que tenen els alumnes i profesors de la facultat de traducció i interpretació per utilitzar l'eina de terminologia del TERMCAT per realitzar algunes classes.

El TERMCAT és el centre de terminologia de la llengüa catalana creat el 1985 amb l'objectiu de coordinar l'activitat terminològica en llengua catalana, mitjançant la prestació de serveis de qualitat, la creació de productes terminològics i la normalització de neologismes, per garantir la disponibilitat de la terminologia catalana en tots els sectors de coneixement d'activitat per afavorir-ne l'ús. El gestor de terminologia GESTEM (creat pel TERMCAT), és de gran utilitat per la gestió de terminologia i la docència però aquest sistema s'està quedant obsolet i té unes quantes inconsistències que fa que el sistema no sigui gaire competent en l'àmbit de la docència o per als professionals. És per això que es pretén realitzar un nou sistema, que anomenarem TermUAB, que sigui capaç de millorar totes aquestes inconsistències i aplicar millores que serien molt útils per als usuaris.

- E-mail de contacte: roger.goma@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Lluís Gesa (Departament DCC) i Anna Aguilar (Departament de traducció i interpretació)
- Curs 2018/19

2 ESTAT DE L'ART

En l'actualitat, no existeixen gaires aplicacions que compleixin les expectatives necessàries per utilitzar-les durant les classes, tot i això, hi ha eines interessants que poden ser d'utilitat. Actualment a l'assignatura de Terminologia aplicada a la traducció, utilitzen dos sistemes per introduir o consultar les diferents terminologies: Gesterm i Bacus.

2.1 Gesterm

Per la introducció de nous termes, els alumnes utilitzen el gestor GESTERM [1], ja que és molt visual i fàcil d'utilitzar. Aquest sistema facilita les principals tasques associades al treball terminològic: creació de fitxers terminològics i de diccionaris que les continguin, manteniment de la informació associada a les fitxes i diccionaris, cerques avançades, i generació de llistats imprimibles.

El problema principal que té aquest sistema és que va ser implementat fa més de 10 anys i té moltes inconsistències, com per exemple que treballa en local i si no exportem la feina realitzada, la perdem.

2.2 Bacus

D'altra banda, antigament en aquesta assignatura també s'utilitzava el sistema BACUS [2], que és un sistema molt similar a l'anterior, però aquest s'utilitza tant per la intro-

ducció com la consulta de terminologies.

Aquest servei, tot i també estar molt obsolet, és molt interessant, ja que permet editar i consultar la informació en qualsevol moment, ja que la plataforma està allotjada al cloud.

Tot i així, es va deixar d'utilitzar per problemes amb els servidors i actualment només s'utilitza com a base de dades per realitzar-hi consultes. D'altra banda, cal dir que aquesta base de dades de BACUS és molt gran i seria interessant poder-la aprofitar.

2.3 WikiTerm

En darrer lloc, també s'ha vist que hi ha altres plataformes en línia com WikiTerm [3] o WikiData [4], que també permeten la consulta i la introducció de noves terminologies però segons la professora Anna Aguilar, aquestes plataformes no permeten introduir les dades necessàries per a cada terminologia i no són útils per l'assignatura.

3 OBJECTIUS

Donada l'alta complexitat del projecte, és importat tenir clars els objectius del projecte.

L'objectiu principal en el desenvolupament de l'aplicació és el de crear un nou sistema basat en el gestor GESTERM. Aquest nou sistema pretén crear una comunitat en l'àmbit de la terminologia per tal de facilitar la consulta de diferents terminologies. A continuació es defineixen i prioritzen els objectius del projecte:

OBJ-01: Aplicar l'enginyeria del software per crear un nou sistema basat en el gestor GESTERM.

Per complir amb aquest objectiu, és necessari complir amb els subobjectius següents:

SUB-OBJ-01-01: Realitzar un document per la captura de requeriments.

SUB-OBJ-01-02: Anàlisi detallat de la situació actual del projecte.

SUB-OBJ-01-03: Disseny de la interfície d'usuari i de l'estructura del codi.

SUB-OBJ-01-04: Realització de tests per comprovar el correcte funcionament del software.

OBJ-02: Realització d'una eina user-friendly, de fàcil ús i multiplataforma.

OBJ-03: Creació d'una àmplia documentació que permeti millorar el sistema en un futur i on es puguin consultar manuals per un posterior manteniment.

OBJ-04: L'aplicació ha de permetre gestionar continguts

públics (per a qualsevol usuari) i privats (per l'assignatura de Terminologia aplicada a la traducció).

4 PLANIFICACIÓ

Per tal de seguir un ordre a l'hora de realitzar el treball, és molt important fer una bona planificació, i així poder organitzar les tasques i detectar possibles desviacions durant el transcurs del treball.

El projecte s'ha dividit en diferent fases:

Definició del projecte: Definició de l'abast, els objectius i els requisits del projecte per tal de poder fer una bona planificació.

Disseny i arquitectura del sistema: Definició dels diferents casos d'ús i disseny de la interfície d'usuari i de l'estructura del codi i la base de dades.

Implementació: Desenvolupament de les funcionalitats definides.

Proves: Comprovació que es compleixen tots els criteris d'acceptació definits i que el codi és de qualitat.

Documentació: Generació de diferents documents i manuals per futures implementacions.

A continuació es mostra una taula amb la planificació general del projecte:

Tasca	Data inicial	Data final
Definició del projecte		
Generació de l'informe inicial	20/02/19	10/03/19
Generació del document de requisits	06/03/19	24/03/19
Disseny i arquitectura del sistema		
Definició dels casos d'ús	25/03/19	07/04/19
Disseny de la interfície d'usuari	08/04/19	21/04/19
Disseny de l'estructura del codi i la base de dades	14/04/19	31/04/19
Implementació de l'aplicació	01/05/19	16/06/19
Tests	01/06/19	23/06/19

Documentació		
Generació de manuals tècnics	16/06/19	23/06/19
Generació de l'informe final	16/06/19	30/06/19

Fig. 1: Planificació general del projecte

5 METODOLOGIA

Per aquest projecte s'ha decidit per optar per una de les metodologies més utilitzades en el sector TIC, el desenvolupament Agile.

Aquesta metodologia consisteix en una sèrie de processos i best pràctiques que permeten al programador desenvolupar un codi de major qualitat i amb un menor temps.

El seu cicle de vida està basat en els sprints que són petites iteracions que permeten entregar petites versions ja finalitzades. Cada tasca d'un sprint, té diferents fases:

Planificació: Cada tasca ha d'estar planificada i definida per tal de tenir clar què és el que s'ha d'implementar.

Desenvolupament: Amb tots els requisits definits és procedeix a implementar la funcionalitat.

Test: Després de cada implementació el codi a de passar una sèrie de tests, tant pel que fa al codi com a nivell funcional.

Integració: Un cop es dona per bona la nova funcionalitat, s'ha d'integrar amb la resta de funcionalitats que ja han passat tot el cicle amb anterioritat.

Un altra factor important per el correcte desenvolupament del projecte són les reunions amb el client. S'han realitzat reunions periòdiques amb la professora Anna Aguilar i el personal informàtic de la facultat de traducció i Interpretació (SERIM). Aquestes reunions han servit per definir quins són els requisits que ha de complir el sistema i també han sigut útils per conèixer quin és l'estat actual dels sistemes que tenen en el SERIM i poder adaptar aquest projecte dins d'aquests sistemes.

Per tal de tenir un control de les tasques a realitzar, s'ha utilitzat Git per tenir un control de les diferents versions del codi. Per cada nou desenvolupament, és necessari crear una nova branca, passar tot el procés de desenvolupament i un cop finalitzat, integrar aquesta branca a la branca pare.

6 TECNOLOGIES UTILITZADES

A continuació, es mostren les tecnologies utilitzades pel desenvolupament del sistema. S'ha decidit utilitzar 4 tecnologies JavaScript, ja que és el rei dels llenguatges per al desenvolupament web. En el projecte s'utilitza l'stack

MERN [5][6][7] que utilitza JS tant en el client com en el servidor. Està compost per quatre tecnologies molt importants a la indústria actualment: MongoDB, Express, ReactJs i Nodejs.

Inicialment es pretenia utilitzar una base de dades relacional com SQL ja que la base de dades de BACUS utilitza aquest tipus de tecnologia, però per culpa de la complexitat extra que podria suposar fer una migració i aprofitar-la, es va decidir realitzar una nova base de dades no relacional (MongoDB), i en un futur, si el projecte té èxit, ja es buscarà la manera de migrar aquestes dades.

A continuació, es pot veure una petita descripció de cada una d'aquestes tecnologies:

MongoDB: Base de dades orientada a documents i d'esquema lliure que proporciona una alta capacitat d'emmagatzement, velocitat en les consultes i és fàcilment escalable.

NodeJs: És un runtime, és a dir, és un programa que executa JavaScript sobre el motor de JavaScript que utilitza Chrome. Node és l'entorn de desenvolupament de la capa del servidor i permet desenvolupar ràpidament aplicacions escalables a nivell de servidor.

Express: Framework per a NodeJS per realitzar aplicacions web. S'adapta molt bé a la filosofia de treballar amb Node al costat del servidor, ja que és ràpid, fàcil d'utilitzar i minimalista. Aquest framework permet organitzar les aplicacions web, des de l'assignació de rutes fins al control de sol·licituds o vistes. En una arquitectura MVC (Model – View – Controller), Express seria el controlador, ja que permet connectar tots els components, gestionar i processar les dades i enviar-les a les capes superiors que permeten a l'usuari tenir accés a la informació.

ReactJs: Llibreria de JavaScript creada i mantinguda per Facebook que permet desenvolupar el Front End d'una aplicació web de manera neta, organitzada i ajuda a que la interacció entre usuaris i l'aplicació sigui molt més efectiva.

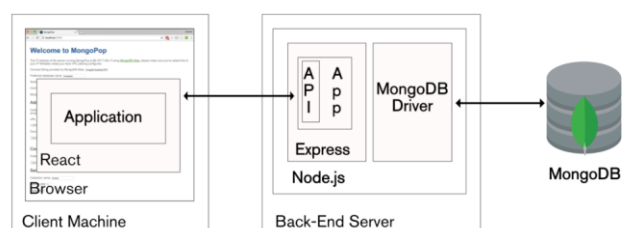


Fig. 2: Arquitectura d'una aplicació MERN

7 REQUISITS I CASOS D'ÚS

Una de les primeres tasques a realitzar en l'enginyeria del software és fer una captura de requeriments i de casos d'ús. Aquest punt podríem dir que és un dels més importants alhora de planificar un projecte, ja que defineix de manera escrita quin serà el funcionament de l'aplicació.

És per això que, seguint el SUB-OBJ-01-01, s'ha generat un document SRS per tal de documentar de manera extensa tant els requisits com els diferents casos d'ús del sistema [8]. A continuació es detallen de manera breu en els següents dos apartats.

7.1 Requisits del sistema

Els requisits identificats s'han classificat de dues formes diferents, els requisits funcionals (REQ-F-X) i els no funcionals (REQ-NF-X). Podem veure una breu descripció de cada un d'ells a continuació:

REQ-F-1: El sistema ha de permetre el registre d'un usuari des de l'aplicació.

REQ-F-2: El sistema ha de tenir un usuari administrador, capaç de tenir un control complet de l'aplicació.

REQ-F-3: El sistema ha de permetre l'usuari ja registrat, poder iniciar i tancar la sessió amb el seu usuari i contrasenya.

REQ-F-4: El sistema ha de permetre a l'usuari ja registrat, recuperar la seva contrasenya.

REQ-F-5: El sistema ha de permetre a l'usuari registrat visualitzar les seves dades i editar-les.

REQ-F-6: El sistema ha de permetre a l'usuari iniciar un nou projecte.

REQ-F-7: El sistema ha de permetre publicar un nou projecte.

REQ-F-8: El sistema ha de permetre a l'usuari editar els seus projectes. Els projectes que no estiguin encara publicats s'han de guardar al compte de l'usuari.

REQ-F-9: El sistema ha de permetre a l'usuari eliminar un projecte, ja publicat o no.

REQ-F-10: El sistema ha de tenir un buscador que permeti cercar qualsevol terminologia publicada.

REQ-F-11: El sistema ha de permetre visualitzar totes les entrades disponibles a la base de dades.

REQ-F-12: El sistema ha de permetre a l'usuari administrador afegir nous camps d'índex i camps d'atribut.

REQ-F-13: El sistema ha de permetre a l'usuari administrador editar i eliminar qualsevol projecte.

REQ-F-14: El sistema ha de permetre a l'usuari administrador conèixer de quin usuari és propietari cada projecte.

REQ-F-15: El sistema ha de permetre establir quines són les llengües del projecte.

REQ-F-16: El sistema ha de permetre introduir 3 tipus de camps d'entrada.

REQ-F-17: El sistema ha de permetre fer un arbre de camp.

REQ-F-18: El sistema ha de permetre una visualització prèvia abans de publicar el projecte.

REQ-NF-1: El sistema ha de ser user friendly.

REQ-NF-2: El sistema ha de ser responsive, per què s'adapti a qualsevol dispositiu.

REQ-NF-3: El sistema ha de respondre de manera ràpida a les consultes dels usuaris.

REQ-NF-4: El sistema ha d'estar sempre operatiu.

REQ-NF-5: El sistema estarà allotjat a un servidor del SERIM.

REQ-NF-6: La màquina ha de permetre allotjar una base de dades MongoDB i executar codi NodeJS.

REQ-NF-7: La màquina on s'allotgi el sistema ha de ser capaç de realitzar backups diàries o setmanals.

REQ-NF-8: La informació personal dels usuaris ha d'estar encriptada i protegida d'usuaris malignes.

7.2 Casos d'ús

Els casos d'ús serveixen per descriure les accions o activitats que poden realitzar els diferents actors dins del sistema. En el context de l'enginyeria del software, un diagrama de casos d'ús representa un sistema com un seguit d'interaccions que es desenvoluparan entre el sistema i els diferents actors com a resposta a una acció que inicia un actor principal.

A continuació es pot veure aquest diagrama de casos d'ús i una breu explicació dels actors i casos d'ús:

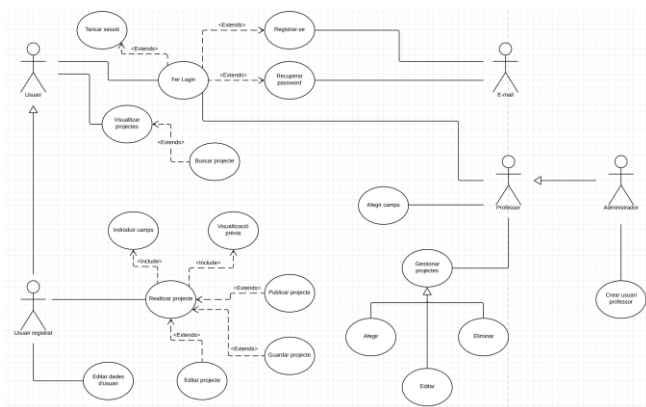


Fig. 3: Diagrama de casos d'ús de l'aplicació

Com es pot veure en el diagrama, el sistema té 5 actors diferents:

Usuari: Persona que no està registrada en el sistema i que utilitza l'aplicació de manera gratuïta però només pot realitzar consultes.

Usuari registrat: Persona registrada al sistema i que utilitza l'aplicació amb totes les funcionalitats disponibles.

Administrador: Usuari extern a l'aplicació capaç de gestionar el contingut de l'aplicació i crear usuaris professors.

Professors: Usuaris amb permisos especials que permeten publicar, editar o eliminar les publicacions dels usuaris registrats.

E-mail: Aquest és un actor extern a l'aplicació que permet fer la validació dels usuaris registrats.

Els casos d'ús identificats en el sistema són els següents:

CU-1 Registrar-se: Si s'accedeix al menú de l'aplicació, hi apareix una secció per registrar-se on es demana un e-mail, nom i cognoms de l'usuari i una contrasenya.

CU-2 Fer login: Si s'accedeix al menú de l'aplicació, hi apareix una secció per fer el login a l'aplicació on es demana l'e-mail i la contrasenya, si són correctes el sistema mostrarà la pàgina principal amb l'usuari ja registrat, en cas contrari, es mostrarà l'error corresponent.

CU-3 Recuperar contrasenya: Si l'usuari ha oblidat la contrasenya, pot accedir a la pàgina de login i allà demanar-ne una de nova. El sistema demanarà el correu electrònic i li enviarà un enllaç on poder introduir una nova contrasenya.

CU-4 Visualitzar projectes: A la pantalla principal de l'aplicació el sistema mostrarà els projectes ja publicats.

CU-5 Buscar projectes: L'usuari pot accedir al buscador de projectes o terminologies i realitzar la cerca desitjada. Si la cerca no té èxit, es mostrarà un missatge d'error.

CU-6 Tancar sessió: Si l'usuari ja ha iniciat sessió i accedeix al menú, li apareixerà una secció per finalitzar la sessió actual.

CU-7 Realitzar projecte: L'usuari pot accedir al menú de l'aplicació i escollir l'opció de realitzar nou projecte. El sistema mostrarà una nova pantalla on es pot començar a introduir els camps desitjats.

CU-8 Introduir camps: El sistema mostra tots els camps que pot introduir l'usuari i aquest els pot omplir de manera obligatòria o opcional.

CU-9 Visualització prèvia: L'usuari abans de publicar el projecte se li permet fer una visualització prèvia de com quedarà el seu projecte.

CU-10 Publicar projecte: L'usuari pot publicar el projecte de manera oficial a l'aplicació en el moment que tingui tots els camps necessaris introduïts. Un cop realitzada la visualització prèvia, li apareixerà un botó per publicar el treball.

CU-11 Guardar projecte: L'usuari pot guardar el projecte en qualsevol moment des del botó de "Guardar".

CU-12 Editar projecte: Un usuari que ja tingui un projecte començat, ja sigui guardat o publicat, el pot editar. Accedint a la secció del menú "Els meus projectes", podrà visualitzar els seus projectes realitzats i editar-los.

CU-13 Editar dades d'usuaris: L'usuari pot accedir al menú de l'aplicació i escollir "Les meves dades". En aquesta secció el sistema permet editar les dades personals (E-mail, contrasenya, nom o cognoms).

CU-14 Afegir nous camps: Un cop l'usuari administrador ha fet login, el sistema mostra una pantalla principal on es poden gestionar diferents aspectes de l'aplicació. Si l'usuari selecciona l'opció d'afegir nous camps, el sistema mostra un formulari on es poden modificar els diferents camps (llenguatge, etc.).

CU-15 Gestionar projectes: L'usuari pot accedir a la secció de gestionar projectes, el sistema mostra una llista de tots els projectes publicats i veure qui és el propietari.

CU-16 Afegir projecte: L'administrador pot afegir un nou projecte de la mateixa forma que els usuaris registrats.

CU-17 Editar projecte: L'usuari administrador pot editar qualsevol projecte que ja estigui publicat.

CU-18 Eliminar projecte: L'usuari administrador pot eliminar qualsevol projecte que ja estigui publicat sense el permís de l'usuari propietari del projecte.

CU-19 Recuperar contrasenya: Envia un e-mail amb un enllaç que permet accedir a una secció de la web on es pot introduir una nova contrasenya.

8 DISSENY

Seguin l'objectiu SUB-OBJ-01-03, en aquest apartat es pretén mostrar els diferents dissenys realitzats, tant pel que fa a la interfície d'usuari, la base de dades i l'estructura del repositori.

8.1 Base de dades

El sistema de base de dades es força senzill, ja que a l'utilitzar una base de dades no relacional, no es necessari crear taules ni relacions entre elles.

El sistema està dividit en dues parts, els usuaris i els projectes. En l'apartat d'usuaris, és on es guarda la informació respecte a l'usuari: identificador únic, nom, cognoms, e-mail, contrasenya i el rol de l'usuari. Aquestes dades ens serviran per a l'autenticació de l'usuari a l'hora d'iniciar sessió a la nostra aplicació.

Pel que fa als projectes, es guarden en un altre fitxer on apareix tota la informació del projecte en qüestió: Nom, descripció, identificador de l'usuari que l'ha creat, si el projecte està guardat o ja publicat, i totes les altres dades que tingui el projecte.

8.2 Estructura del repositori

Pel que fa a l'estructura del repositori ja no és tant senzilla. Tot el codi està situat en el mateix repositori però s'ha de diferenciar clarament quin és el codi del servidor i quin és el del client.

A l'arrel del projecte s'hi pot veure tot un seguit d'arxius de configuració:

Jest.config.js: Serveix per configurar els tests unitaris.

Package.json: És on es troben totes les dependències del projecte i la configuració principal.

Webpack.config.js: És un arxiu de configuració que permet compilar el javascript i els scss del front-end, i unificar-los en un sol fitxer de sortida.

També s'hi pot trobar la carpeta /src, que és on hi ha tant el codi del servidor com el del front-end. Aquesta carpeta està estructurada de la següent forma:

/app: És on es troba tota la lògica de ReactJs, és a dir, del front end. Conté els arxius js, el css i els tests unitaris.

/config: Hi ha les traduccions del sistema, configuracions per la base de dades i el registre d'usuaris.

/models: Esquemes per emmagatzemar les dades a la base de dades.

/public: Arxius generats per webpack després de compilar el codi React. Són els fitxers que utilitzarà el navegador per renderitzar les pàgines.

/routes: Accions que s'executen quan es crida a una determinada ruta.

8.3 Interfície d'usuari

Abans de començar a desenvolupar el codi, és important tenir clar com és l'interfície d'usuari. Com que no es tenia coneixement de cap eina per fer dissenys, es va optar per l'opció tradicional i es van fer els dissenys amb paper i llapis.

El disseny ha estat basat principalment en l'interfície del GESTERM, ja que era senzilla i fàcil d'utilitzar.

Com que el disseny no ha estat realitzat a partir de cap plantilla de css, s'ha hagut de fer tot el disseny des de zero, tenint en compte les característiques i components que proporciona Materialize.

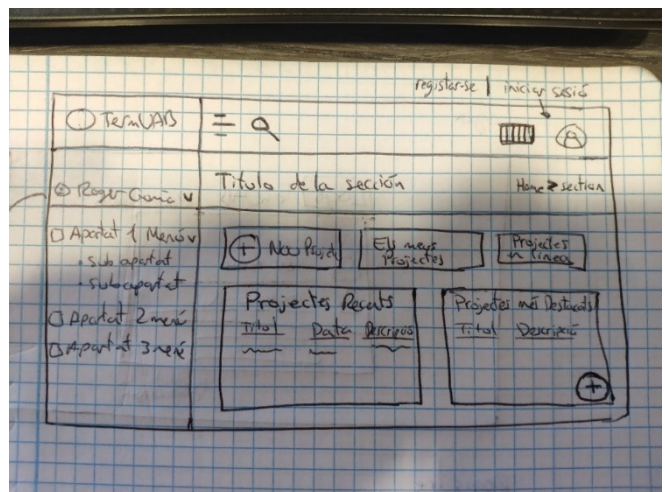


Fig. 4: Disseny de la pàgina inicial de l'aplicació

9 IMPLEMENTACIÓ

Per realitzar el projecte s'ha treballat en un entorn en local creat amb node i que consisteix en tenir un arxiu index.js on es configura aquest entorn, s'indica el port on s'executa la nostra aplicació, el path dels nostres fitxers, la connexió amb la base de dades, etc.

Per tal simplificar l'explicació, dividirem la implementació en 2 apartats: El backend i el frontend.

9.1 Backend

Pel que fa a la implementació del backend del nostre sistema, va ser necessari realitzar una bona cerca per tal de buscar la millor solució pel projecte. Es van consultar alguns posts com "Building a React CRUD Application from start to finish" [9] o "Build a Login/Auth App with the MERN Stack" [10] i a partir d'aquí va començar la implementació.

Inicialment, es va crear la base de dades amb MongoDB on s'emmagatzarien totes les dades i que ja s'ha explicat la seva estructura en l'apartat anterior.

El segon pas va ser crear l'autenticació d'usuaris. Per fer-ho de manera senzilla i segura, es va utilitzar un middleware molt popular anomenat Passport [11], i que ens permetrà tenir tan usuaris registrats des de la mateixa aplicació, com a través de xarxes socials com Gmail, facebook, twitter, etc. Actualment el sistema només permet el registre des de la mateixa pàgina web i el sistema emmagatzema les següents dades de l'usuari: nom i cognoms, e-mail i una contrasenya encriptada a través d'un hash.

Passport executa l'acció d'autenticació necessària en funció de la petició que rebí al servidor, és per això que també és necessari crear un sistema de rutes que ens permeti gestionar aquestes peticions.

Un cop implementació de l'autenticació dels usuaris, es va procedir a crear les funcionalitats pròpies del sistema. Aquestes funcionalitats són més senzilles de realitzar, ja que s'encarreguen d'enviar la informació necessària del servidor al navegador o del navegador al servidor. Cada funcionalitat té associada una URL i el tipus de petició (GET, POST, PUT, DELETE) i en funció de quina petició llenca el navegador, el nostre servidor guardarà, enviarà o esborrarà les dades necessàries de la nostra base de dades.

9.2 Frontend

Pel que fa a la part visual de l'aplicació, tal com s'ha esmentat anteriorment, s'ha utilitzat el framework anomenat ReactJs. De l'encapsulació i compilació dels fitxers, se n'ha encarregat webpack 4, que ha sigut configurat dins el projecte en un arxiu de configuració. Webpack a permès gestionar les dependències del projecte de manera senzilla i permet minificar el codi compilat i així, reduir el pes dels fitxers.

Per tal d'aconseguir un millor rendiment, el sistema funciona amb una SPA (Single Page Application) [12], és a dir, la web mostra totes les pantalles a la mateixa pàgina, sense necessitat de recarregar el navegador. És per això que s'ha creat un sistema de rutes que permet renderitzar les diferents vistes en funció de la url del navegador, però sempre recarregant els components que encara no estan en el DOM.

Per tal de facilitar la implementació, s'ha utilitzat un framework de CSS anomenat Materialize [13]. Aquest framework ha facilitat la tasca de fer un disseny responsive, unificat i elegant. Com que només amb aquest framework no era possible la implementació exacta dels dissenys realitzats, també va ser necessari introduir css personalitzat.

Aquesta tasca es va realitzar amb l'ajuda d'un paquet de npm [14] anomenat Css-Modules [15] i que transforma el nom les classes de css amb un identificador únic. D'aquesta forma es podrà utilitzar qualsevol nom de classe, encara que Materialize ja l'estigui utilitzant.

L'aplicació té 4 tipus de pàgines, les quals totes comparteixen tant el menú lateral com la barra superior. Quan entrem a l'arrel de la nostra aplicació hi veiem la pàgina principal on se'ns presenten les diferents opcions a realitzar.

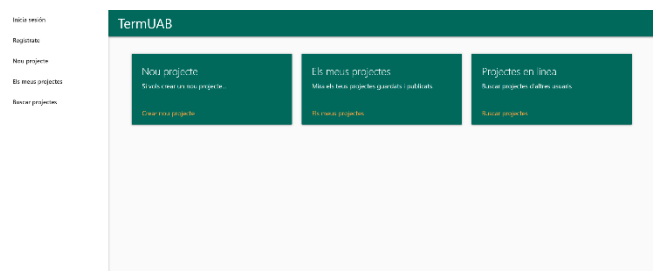


Fig. 5: Pàgina inicial

També tenim les dues pàgines d'autenticació (login i registre), que simplement tenen un formulari on introduir les dades personals.

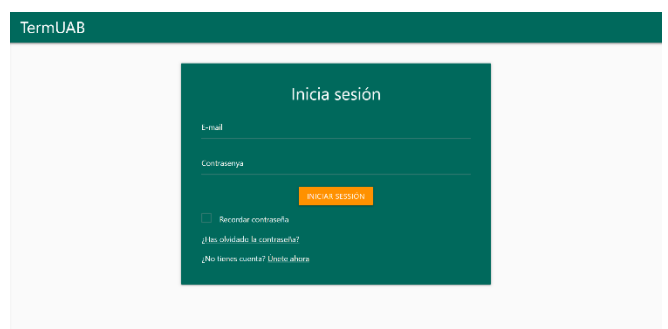


Fig. 6: Pàgina de login

Una de les pàgines més importants és la d'introducció d'un nou projecte. Aquesta té molts possibles camps per introduir, un botó per guardar el projecte.

Fig. 7: Pàgina on es pot crear un nou projecte

Finalment, tenim les pàgines per consultar els projectes. Tant els projectes públics com els personals es mostren de la mateixa forma, però en diferents apartats.

Fig. 8: Pàgina amb el projectes del usuari

10 PROVES

Les proves són un apartat molt important quan es desenvolupa una aplicació, ja que d'aquesta manera s'assegura una major qualitat del sistema. Per tal de complir amb l'objectiu SUB-OBJ-01-4, s'han realitzat dos tipus de proves: unitàries i funcionals.

10.1 Proves unitàries

Per certs components del codi, és necessari realitzar proves exhaustives per tal de comprovar el seu correcte funcionament. És important realitzar aquestes proves al mateix moment que es va desenvolupant el sistema, d'aquesta manera es pot assegurar que el nou desenvolupament funciona i que les funcionalitats ja implementades segueixen tenint un correcte funcionament.

Per realitzar aquestes proves, s'ha utilitzat Jest [16] i l'objectiu principal era realitzar un seguit de proves per tal de cobrir el màxim coverage possible.

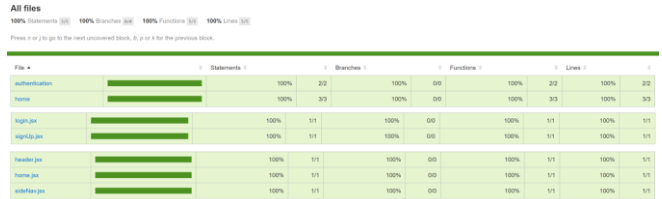


Fig. 9: Coverage dels tests unitàris

10.2 Proves funcionals

Un cop realitzades amb èxit les proves unitàries, és important validar que l'aplicació es comporta com s'esperava. És per això que es fa una validació a nivell funcional sobre dispositius reals. Amb aquestes proves no només es comproven les funcionalitats, sinó que també es comprova la performance i la usabilitat.

11 RESULTATS

Després de tot el treball realitzat s'ha aconseguit tenir una aplicació responsive que permet realitzar totes les funcionalitats des de diferents dispositius. Aquestes funcionalitats permeten a l'usuari registrar-se al sistema per tal de realitzar nous projectes, ja siguin per l'assignatura de Terminologia aplicada a la traducció o per ús personal. Aquests projectes es guarden a la base de dades i cada usuari els pot recuperar quan vulgui. A més a més, el sistema permet visualitzar qualsevol dels projectes publicats de manera pública.

Desgraciadament, no s'ha pogut finalitzar tot el projecte i complir tots els requisits ja que actualment els projectes només permeten introduir els camps bàsics com el nom, descripció i les llengües del projecte.

Actualment, tampoc existeixen els usuaris administradors i per tant l'OBJ-04 no s'ha pogut realitzar amb èxit.

Aquest objectiu no s'ha pogut complir degut a la falta de coneixement de les tecnologies del backend i que a penalitzat molt en el temps d'implementació, sobretot al principi del projecte.

Tot i així s'ha de dir que, tal i com s'ha vist durant l'article, els altres objectius s'han pogut assolir amb èxit, ja que s'ha intentat seguir les pautes marcades per l'enginyeria del software.

Per finalitzar, com que aquest projecte pretén ser Open Source, s'ha publicat el codi en un repositori públic de Github: <https://github.com/rogergoma96/TermUAB-Code> on qualsevol persona pot veure i descarregar el projecte.

Com que l'objectiu OBJ-03 del projecte era el de realitzar una documentació que facilites comprendre el projecte, s'ha realitzat un petit manual que permet entendre com funciona el sistema i instal·lar-lo en local.

12 CONCLUSIONS

Finalment, podem dir que actualment tenim un programa funcional, una primera versió utilitzable però que s'hi poden aplicar moltes millores. Tenint en compte que el software s'ha desenvolupat completament des de zero amb un temps limitat, es pot dir que s'han assolit la major part dels objectius encara que el sistema no tingui totes les funcionalitats marcades en els requisits i els casos d'ús.

De cara a un treball futur es podrien afegir les funcionalitats relacionades amb els administradors. Uns usuaris que siguin capaços de gestionar el contingut del sistema i els seus usuaris. També seria interessant fer una migració de la base de dades que té BACUS, ja que conté molta informació que ajudaria a enriquir la nova aplicació. D'altra banda, també seria necessari realitzar algunes millores a en l'àmbit de l'experiència de l'usuari, afegint alguns textos informatius per exemple.

AGRAÏMENTS

Per acabar, agrair el suport que m'ha donat el meu tutor acadèmic Lluís Gesa per tirar endavant el projecte. També vull donar les gràcies a la professora Anna Aguilar, que va ser qui va fer la proposta del treball i sempre ha estat a disposició per ajudar-me.

BIBLIOGRAFIA

[1] TERMCAT – GesTerm, Data d'accés: 20 Febrer 2019, Disponible a: <http://www.termcat.cat/ca/GesTerm/>

[2] BACUS – Base de datos de conocimiento universitatio UAB, Data d'accés: 20 Febrer 2019, Disponible a: http://www.fti.uab.es/departament/BACUS/index_es.htm

[3] WIKITERM, Data d'accés: 20 Febrer 2019, Disponible a: <https://www.clanget.com/wikiterm/>

[4] WikiData – Data d'accés: 26 Maig 2019, Disponibilitat a: https://www.wikidata.org/wiki/Wikidata:Main_Page

[5] PLATZI- Qué es el stack MERN de JavaScript, Data d'accés: 30 Març 2019, Disponible a: <https://platzi.com/blog/que-es-mern-stack-javascript/>

[6] Youtube - MERN Stack Curso - MongoDB, Express, React y Nodejs, Data d'accés: 26 Maig 2019, Disponible a: <https://www.youtube.com/watch?v=DqpL5UtIHus>

[7] Youtube – The MERN Stack Tutorial, Data d'accés: 26 Maig 2019, Disponible a: <https://www.youtube.com/watch?v=qvBZevK1HPo>

[8] GitHub – Document de Requisites, Data d'accés: 14 Abril 2019, Disponible a: <https://github.com/rogergoma96/TermUAB/blob/master/Documentacion/Deliveries/Document%20de%20requisits.pdf>

[9] Building A React CRUD Application From Start To Finish, Data d'accés: 26 Maig 2019, Disponible a: <https://medium.com/codingthesmartway-com-blog/the-mern-stack->

[tutorial-building-a-react-crud-application-from-start-to-finish-part-2-637f337e5d61](https://medium.com/codingthesmartway-com-blog/the-mern-stack-tutorial-building-a-react-crud-application-from-start-to-finish-part-2-637f337e5d61)

[10] Build a Login/Auth App with the MERN Stack, Data d'accés: 26 Maig 2019, Disponible a: <https://blog.bitsrc.io/build-a-login-auth-app-with-mern-stack-part-1-c405048e3669>

[11] Passport JS, Data d'accés: 30 Maig 2019, Disponible a: <http://www.passportjs.org/docs/>

[12] Single Page Application, Data d'accés: 30 Maig 2019, Disponible a: https://es.wikipedia.org/wiki/Single-page_application

[13] MaterializeCSS, Data d'accés: 30 Maig 2019, Disponible a: <https://materializecss.com/>

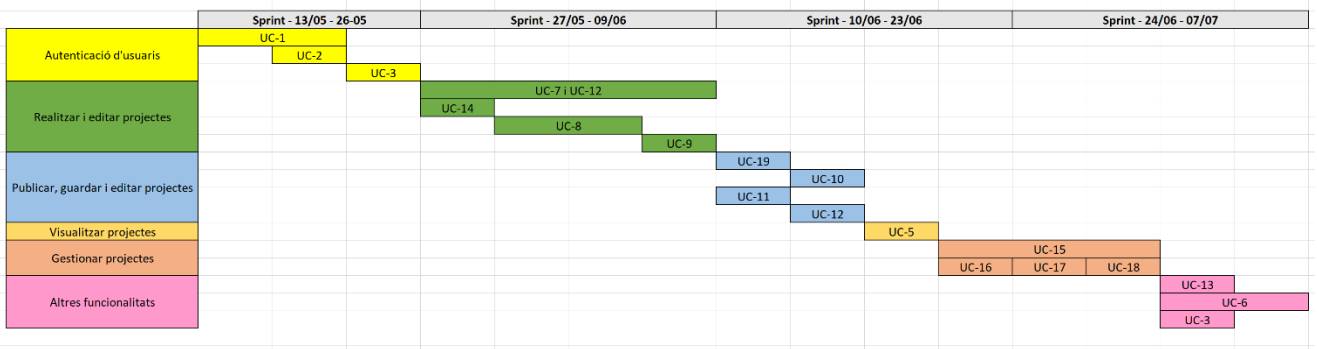
[14] Npm js, Data d'accés: 30 Maig 2019, Disponible a: <https://www.npmjs.com/>

[15] Css modules, Data d'accés: 30 Maig 2019, Disponible a: <https://www.npmjs.com/package/react-css-modules>

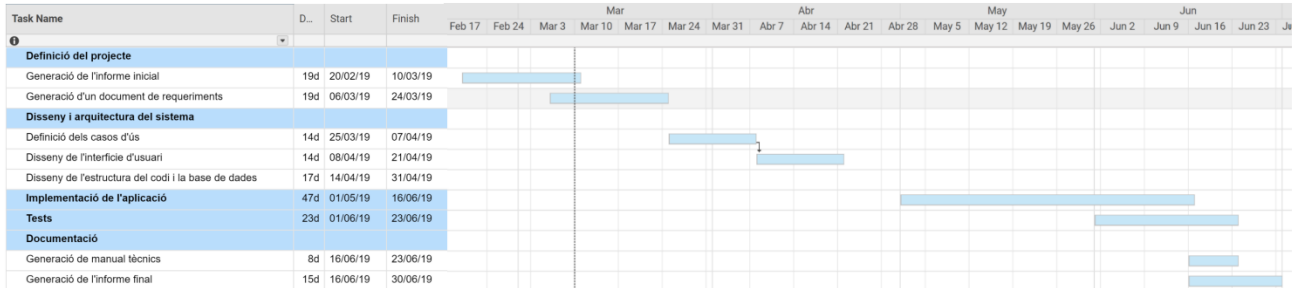
[16] Jest js, Data d'accés: 30 Maig 2019, Disponible a: <https://jestjs.io/docs/en/tutorial-react>

APÈNDIX

A.1 DIAGRAMA DE GANTT AMB ELS CASOS D'ÚS



A.2 DIAGRAMA DE GANTT DEL PROJECTE



A.3 DIAGRAMA DE CASOS D'ÚS

