

# Service Exchange Platform

## Treball Final de Grau

Lorena Tortajada Roperó

**Resum**– En aquest treball es presenta el projecte relacionat amb la creació d'una plataforma d'anuncis de serveis mitjançant eines de desenvolupament populars. Concretament, es tracta del disseny d'una web que té com a objectiu principal l'intercanvi de serveis entre particulars. Aquest treball neix amb l'objectiu d'actuar com a plataforma de serveis per donar resposta a necessitats sorgides davant crisis sanitàries com la COVID-19. Així doncs, el projecte, amb nom **Service Exchange Platform**, aplica tecnologies com ara Node.js per a la creació del *back-end*, mitjançant el framework per defecte d'Express. També, n'aplica d'altres més innovadores com ara els framework Vue.js, ElasticSearch i Bootstrap pel *front-end* amb la finalitat d'aconseguir una experiència d'usuari satisfactòria. Així mateix, la base de dades escollida per guardar tota la informació a processar pel *back-end* és PostgreSQL. Per dur a terme el projecte, s'han definit uns requisits i posteriorment s'ha planificat un seguit de tasques a l'Eina de Planificació d'Asana. Mitjançant una metodologia en format cascada, s'ha procedit a la creació en primer lloc del *front-end*, posteriorment el *back-end* i la integració amb la BD i per últim la unió del *front-end* i *back-end* mitjançant Axios.

**Paraules clau**– web, framework, *front-end*, *back-end*, Vue.js, Bootstrap, Nodejs, Express.js, PostgreSQL, ElasticSearch, Axios

**Abstract**–This paper presents the whole project related to the creation of a service announcement platform using popular development tools. Specifically, it contains the design of a website whose main objective is the exchange of services between individuals. This work was born with the aim of acting as a service platform to respond to needs arising from health crises such as COVID-19. Thus, the project, named **Service Exchange Platform**, applies technologies such as Node.js for *back-end* creation, using the default framework of Express. Regarding the *front-end*, the project makes use of more innovative technologies such as Vue.js framework, ElasticSearch and Bootstrap, in order to achieve a satisfactory user experience. In addition, the chosen database that stores all the information to be processed by the *back-end* is PostgreSQL. To carry out the project, some requirements have been defined and subsequently a series of tasks have been planned in Asana Planning Tool. Using a cascade methodology, the *front-end* was created in first place, while later the *back-end* jointly the integration of DB. Finally, the *front-end* and *back-end* were merged through Axios.

**Keywords**– web, framework, *front-end*, *back-end*, Vue.js, Bootstrap, Nodejs, Express.js, PostgreSQL, ElasticSearch, Axios



## 1 INTRODUCCIÓ

Amb més de 1700 milions de webs l'any passat [1], recentment s'han creat de noves molt populars com el cas de Lets-Bonus, Groupalia o inclús Glovo, que l'any passat va facturar 52 milions d'euros [2], on s'ofereixen serveis entre una empresa i particulars. No obstant, cap d'elles té com a base

l'intercanvi entre particulars i per tant, el xat personal entre ells. A més a més, disposen d'una cartellera de serveis limitats als proposats per l'empresa, que molts cops dista de les necessitats reals d'un usuari. L'objectiu del projecte és el desenvolupament d'una pàgina web que pretén actuar com a plataforma de serveis per donar resposta a necessitats sorgides davant crisis sanitàries com la COVID-19.

Doncs, un cop explicada la motivació, el problema a resoldre es pot resumir a grans trets com la creació d'una pàgina web. Aquesta tasca pot semblar molt simplista però alberga una gran complexitat.

En un món actual on el creixement d'aplicacions web és exponencial [1], l'abandonament per part de l'usuari es

- E-mail de contacte: lorena.tortajada@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Sergi Robles (DEIC)
- Curs 2019/20

situa al 53 % si la pàgina web triga més de 3 segons en carregar-se [3].

En la creació d'aquesta web és molt important la selecció adequada de les eines per a la seva construcció (i.e frameworks) així com el propi disseny, ja que seran els que proporcionin les funcionalitats desitjades pels clients amb un resultat de bona experiència. Service Exchange Platform té coma principal objectiu ésser una web dinàmica on qualsevol ciutadà que es trobi confinat o en altre situació de no tenir accés directe a serveis, pot publicar un servei que ofereix o demanda i establir conversa amb els interessats.

La resta del document s'estructura de la següent manera. La subsecció 1.1 detallarà els objectius generals que es volen assolir amb aquest projecte.

La següent secció detalla requisits propis del disseny i funcionament, Sec. . La Sec.2 para atenció a l'estat de l'art i per conseqüència, a les tecnologies que s'han utilitzat tant en disseny com en implementació de la pàgina web justificant-ne la seva adequació al temps i els requisits de la secció prèvia per garantir una experiència d'usuari satisfactòria. La metodologia proposada per realitzar amb èxit el projecte es detalla la Sec.4. El camí marcat per la metodologia i els avenços assolits reflecteixen el contingut de l'apartat Sec.5. Per últim, s'exposen els resultats obtinguts per la prova pilot i el testeig així com les conclusions finals, Sec.6 i Sec.7 respectivament.

## 1.1 Objectius i abast del projecte

Aquest primer apartat defineix l'objectiu principal d'aquest projecte i fixe l'abast d'aquest tot establint una sèrie de sub-objectius.

Com que es tracta d'un treball de final de grau d'enginyeria informàtica i per tant, s'ha d'assolir com a objectiu principal **la capacitat d'un estudiant de gestionar i resoldre un projecte de complexitat similar als de la vida real**, se'n proposa com a objectiu principal **el desenvolupament de cada una de les parts que s'engloben dins la creació d'una pàgina web: front-end, back-end, base de dades i altres afegits per millorar la qualitat que oferiran un servei per publicar anuncis de particulars**

D'aquest objectiu global se'n deriva d'altres ordenats segons el grau de pes que tenen al projecte en ordre descendent. És a dir, el primer objectiu té una prioritat màxima i l'últim llistat té una prioritat mínima;

1. Aprendre noves tecnologies emergents que serveixin pel futur laboral.
2. Analitzar els requisits funcionals i no funcionals del projecte
3. Seleccionar les tecnologies necessàries:
  - (a) Seleccionar la tecnologia del *front-end*.
  - (b) Seleccionar la tecnologia del *back-end*.
  - (c) Seleccionar el tipus de Base de Dades.
  - (d) Seleccionar tecnologia addicional per millorar l'experiència de l'usuari.
4. Realitzar el disseny de la pàgina web.

5. Assegurar el dinamisme i l'adaptabilitat de la pàgina.
6. Implementar aquest disseny i les funcionalitats associades tant en el *front-end* com en el *back-end*.
7. Provar el disseny mitjançant una prova pil·lot.
8. Corregir errors o introduir millores sorgides de l'etapa de proves.
9. Analitzar aspectes del futur de la pàgina web. (e.g millores que es poden introduir a posteriori).

## 2 ESTAT DE L'ART - ELECCIÓ DE TECNOLOGIES

Aquesta segona secció presenta l'estat de l'art dels principals elements que s'han identificat en la creació d'una pàgina web: *front-end*, *back-end*, base de dades i tecnologies addicionals per millorar la web. Gràcies a l'estat de l'art es determinaran quines són les tecnologies més adients per aquest projecte tenint en compte el llistat d'objectius i l'abast presentats anteriorment.

Així doncs, tal com expressa el punt 2 de la Sec.1.1, l'elecció de les tecnologies emprades es pot classificar en quatre tipus: elecció de la tecnologia del *front-end*, elecció de la tecnologia del *back-end*, elecció del tipus de base de dades i elecció de tecnologia addicional per millorar l'experiència de l'usuari.

### 2.1 Elecció de Framework pel *front-end*

Aquesta subsecció revisa els principals *front-ends* Javascript -ja que és el que s'imparteix com a base de desenvolupament web a la universitat- existents al mercat i n'escull el més adient tenint en compte els objectius i el temps disponible.

Existeixen diversos frameworks open-source que permeten una millora en la gestió i més facilitat a l'hora de crear *front-ends* (vistes per l'usuari). Entre ells, en destaquen tres que treballen amb eines JavaScript: **Angular, React i Vue.js**.

La següent taula mostra una comparativa dels tres frameworks segons diferents criteris:

Criteri	Angular	React	Vue.js
Learning Curve	Diffícil	Mitjà	Fàcil
Single Page Apps	Idoni	No gaire útil	No gaire útil
Multi-page apps	No gaire útil	Idoni	Idoni
Performance Benchmarks	Bo	Bo	Bo
Deployment	Bo: Angular CLI	Bo: create-react-app	Bo: Vue CLI
Industry Adoption	Alt	Alt	Mitjà

TAULA 1: COMPARATIVA ANGULAR, REACT I VUE.JS EXTRET DE [4].

Davant aquesta comparativa, **Angular** -en concret Angular 8 (la versió més nova)- és un framework creat per Google **tot-en-un**: té eines potents per la construcció, routing, estat i per fer validacions de formularis. És un framework molt adient per webs d'una sola pàgina però poc per aquelles que tinguin un disseny clàssic de fitxers HTML, CSS i Javascript.

Ja que només dispo de coneixement en matèria de *Multi-page apps* i Angular suposa una corba d'aprenentatge molt forta a causa de l'ús de sintaxi Typescript, queda **descartat** com a opció.

Pel que fa a React i Vue, ambdós *front-ends* són molt bones opcions. No obstant, React té una corba d'aprenentatge més complicada, ja que tot s'escriu en JavaScript però sol fer ús de **JSX**, una extensió sintàctica de java que descriu com s'ha de veure la interfície UI. Com es disposa de poc temps, és un obstacle pel projecte a tenir present.

D'altra banda, Vue està menys estès en l'àmbit comercial segons [4]. No obstant, en l'enquesta d'Stack Overflow per a desenvolupadors, va quedar en segon lloc, superant a Angular segons [4]. Així mateix, en la seva web s'ofereixen cursos gratuïts i una gran quantitat de documentació. També disposa d'una comunitat molt participativa [5].

A més a més, podem trobar nombrosos llibres com [6] i [7] que introdueixen a Vue pas per pas i senyalen que també és compatible amb l'última especificació de Javascript ES6 i amb Bootstrap per facilitar la creació dels arxius d'estil .CSS.

Articles més recents com [8], mostren estadístiques de la influència creixent d'aquest framework on inclús el creador **Evan You** publicava un Twitter senyalant que Vue era el framework més satisfactori en un 91.2% al reconegut rànquing *State of JS 2018*.

Així mateix, Vue presenta tots els elements desitjables per a que una web sigui flexible, escalable i **adaptada a dispositius mòbils** amb un component Weex o NativeScript del disponible al repositori d'elements i plugins.

Després de tota la informació donada, la tria es decanta per l'ús de **Vue.js**, degut al seu **aprenentatge progressiu i el seu creixement positiu com a tecnologia**.

## 2.2 Elecció de Framework pel back-end

Aquesta subsecció presenta l'elecció del segon element present al disseny d'una pàgina web: el *back-end*.

Per temes de simplicitat i temps limitat, l'elecció del *back-end* està estretament lligada a l'elecció feta pel cas del *front-end*.

Com no es disposa de gaire temps, és aconsellable que el *back-end* es sustenti sobre el mateix llenguatge de programació que el *front-end*, JavaScript. D'aquesta manera, es simplificarà la seva programació pels mateixos coneixements de JavaScript apresos en assignatures impartides al grau.

Així doncs, es descarta Django -una opció que al principi semblava molt potent-, ja que opera amb Python, una tecnologia també vista en el Grau.

Una bona opció per a centrar el projecte en el *front-end*, part més visual i per tant més valorada pels usuaris, serà l'ús del conegut i familiar entorn **Node.js**.

Amb l'ajut del framework open-source d'**Express.js**, la part del *back-end* serà creada. Express.js és el 'de facto' framework per fer aplicacions web sobre Node.js [9].

Express ajudarà amb les peticions HTTP procedents del client, el view-rendering de les interfícies d'usuari i la connexió amb les dades emmagatzemades a la base de dades.

## 2.3 Elecció de la Base de Dades

Aquesta subsecció presenta l'elecció del tercer element present al disseny d'una pàgina web: la base de dades.

Per tal d'escollir quina BD és la millor, s'ha buscat estadístiques a DBEngines on hi consta un rànquing.

El model de BD serà **relacional** perquè és aquest model en el qual dispo de coneixements impartits amb anterioritat per la universitat.

Un altre aspecte important és que s'ha decidit, en la mesura del possible, utilitzar programari lliure. Per tant, el servidor de gestió de la base de dades ha de tenir com a requisit ser *Open Source*.

Així doncs, el rànquing publicat per DBEngines, és el següent:

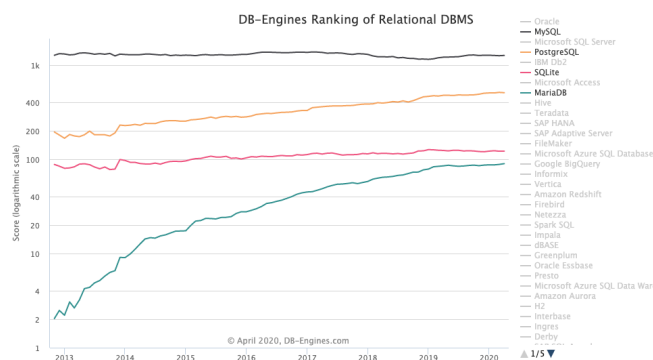


Fig. 1: Ranking de popularitat BD relacionals i amb SGBD Open Source segons [10].

Per tant, les dues alternatives guanyadores a considerar segons els requisits esmentats prèviament són **MySQL i PostgreSQL**. Tot i que MySQL té un índex de popularitat major, es veu com aquesta tecnologia decau mentre que PostgreSQL continua en creixement.

A més a més, una tesis feta a la Universitat Autònoma de Mèxic de l'any 2016 on es compara els dos SGBD anteriors conclou que, és millor utilitzar **PostgreSQL** ja que MySQL experimenta problemes de concurrència [11].

## 2.4 Elecció tecnologies addicionals

En aquest subapartat s'esmenten les tecnologies addicionals que formen part de la web i que, tot i no ser un element bàsic en la seva construcció, complementen i milloren la qualitat de la web.

Doncs, per millorar aspectes de disseny com l'**accessibilitat** i aconseguir una web **user-friendly**, s'utilitzarà la llibreria **Bootstrap**. L'objectiu d'aquesta tecnologia és aconseguir que la web s'ajusti al concepte de *responsive web-app*.

Bootstrap és un conjunt d'eines que inclouen components integrats preconstruïts i potents complements construïts per complementar el codi HTML, CSS o JS.

## 3 REQUISITS I DISSENY

En aquesta secció es presenten els requisits funcionals i no funcionals propis de la pàgina web i la base del disseny d'aquesta, és a dir, la idea sobre la qual ha estat concebuda.

Així doncs, per determinar els requisits funcionals de la pàgina, s'ha de tenir en compte el següent: un usuari pot ésser a la vegada tant 'prestador' d'un servei com consumidor d'un altre. L'usuari podrà publicar anuncis oferint un servei o demanant-ne. Tal com s'ha mencionat a la introducció, **els serveis s'estructuraran en cinc classes:**

- Esports
- Educació
- Oci
- Salut
- Assistència a persones

D'altra banda, l'usuari podrà escollir impartir aquest servei de forma gratuïta o bé remunerada. Des de l'aplicació, no s'intentarà regular formes de pagament. Més aviat la idea és habilitar un xat per posar-s'hi en contacte. Serà tasca dels usuaris que s'acordin sessions, durades i remuneració del servei en qüestió si s'escau. Es guardarà un historial de les converses entre dos usuaris amb relació a un anunci. Així mateix, els serveis poden oferir-se de manera Virtual o associats a una determinada localitat. D'aquesta manera, si l'usuari introdueix al cercador de serveis una localització desitjada, es desplegaran dins les mateixes classes, serveis afegits on la seva existència depengui de la localitat. Un exemple d'aquest servei podria ser el servei d'atendre una persona gran.

L'aplicació web no regularà el compliment de la llei decretada pel confinament en relació a desplaçaments. Així mateix, també pretén ser una web on el seu ús no es restringeixi només al període de confinament sinó que, com a particulars, es puguin anunciar serveis d'esport, restauració a domicili, etc. Un exemple d'aquesta casuística seria el lloguer d'una pista de pàdel per un determinat preu.

Els requisits funcionals derivats de l'anterior explicació del projecte són:

1. Publicar per part de l'usuari anuncis tant sigui d'oferta com de demanda d'un determinat servei.
2. Mostrar un taulell d'anuncis configurable per l'usuari amb filtres i categories per buscar serveis desitjats.
3. Contactar per xat amb el proveïdor d'aquest servei anunciat.
4. Trobar serveis específics per una localització desitjada.
5. Guardar la informació requerida a una base de dades.

Mentre que un requisit no funcional és:

1. Seleccionar les tecnologies adients pel desenvolupament de l'aplicació.

## 4 METODOLOGIA I PLANIFICACIÓ

El següent apartat mostra la metodologia seguida i la planificació del projecte, ja que aquesta garanteix una probabilitat d'èxit superior.

El model de desenvolupament escollit serà **en cascada**, un disseny seqüencial que inclourà les següents fases:

- **Anàlisi i especificació de requisits**
- **Disseny resultant de la web i BD**
- **Implementació de les funcionalitats del *front-end* i *back-end* i comunicació entre aquest**
- **Prova pilot i testejos**
- **Desplegament final de la web i millores proposades pel futur**

El core de la planificació serà la implementació de les funcionalitats de *front-end* i *back-end* i la seva relació que desplega una sèrie de fites enumerades a a continuació:

1. Fi etapa aprenentatge: es destina una part del projecte per l'aprenentatge dels frameworks escollits.
2. Fi etapa disseny: aquesta fita inclou el disseny de les taules de la Base de Dades necessàries i el plantejament del disseny global de la web.
3. Fi disseny vistes: aquesta fita té com a resultat un Mockup amb el disseny de la pàgina web que després s'implementarà.
4. Fi implementació d'usuari: aquesta fita inclou tota la implementació derivada de la interacció de l'usuari amb la pàgina web: des del *front-end*, *back-end* i connexió amb la BD.
5. Fi implementació anuncis: aquesta fita inclou tota la implementació relacionada amb la publicació d'anuncis i la modificació d'aquests.
6. Fi etapa testeig: Aquesta fita conclou quan es proven els escenaris possibles d'utilització de la web amb exemples.

En la fase prova pilot i testeig que finalitza amb la fita *Fi etapa testeig* es comprovarà cada un dels requisits funcionals introduint clients i anuncis de prova. També es farà una prova pilot amb familiars i amics per detectar limitacions o possibles millores de cara al futur.

El límit del projecte és el **28 de juny** amb el lliurament del dossier. És per això que s'utilitzarà una eina externa de planificació, l'**Asana**, per poder portar el compte de tasques que s'han de fer cada setmana, estat d'aquestes i comentaris.

La planificació en cascada sorgida s'ha plasmat a l'eina de planificació d'Asana. En ella, també s'han configurat les fites mencionades que s'aniran complint a mesura que s'han assolit els objectius.

Mitjançant el següent enllaç es poden veure totes les tasques programades i les fites a l'Asana: Service Exchange Platform-TFG.

## 5 DESENVOLUPAMENT

Aquesta secció resumeix el desenvolupament que es va dur a terme seguint la planificació esmentada en l'anterior secció i els documents adjunts derivats.

Així doncs, el primer pas del desenvolupament va ser fer tota la planificació de tasques i fites sorgides dels requisits

funcionals de la pàgina web. Un cop totes les tasques estaven programades, es va configurar un Gantt inicial per a una visió global del projecte. El desenvolupament s'ha fet seguint aquest Gantt adjuntat a l'apèndix.

Els primers passos que marcaven la primera fita *Fi etapa aprenentatge* van ser aprendre les noves tecnologies que s'utilitzarien com Vue.js i Express.js.

Un cop assolida l'etapa d'aprenentatge, la següent etapa era el disseny de la Base de Dades. El disseny resultant es troba a Fig. 5.

La Base de dades consta de quatre taules: *Anuncio*, *Chat*, *Usuario* i *Mensaje*.

La diferència entre una Oferta de servei i una Demanda de servei ve indicada pel camp "tipo" de la taula Anuncio. A més a més el camp Preu d'anunci admet valors nuls per tenir en compte la possibilitat esmentada d'oferir un servei gratuït. Per altra banda, la identificació de l'usuari es fa a través de l'email i un userid intern. El fet d'incloure l'email és perquè es vol tenir la possibilitat de realitzar l'autenticació d'un usuari a la pàgina web mitjançant l'API de Google (més còmode) i atorgar a la vegada seguretat. Per últim, cal considerar que un xat és referent a un únic anunci i compartit entre dos usuaris. Així mateix, un anunci només pot pertànyer a un usuari. Cada xat està identificat de forma única per un *chatid*. Cada missatge de la taula *Mensaje* tindrà un identificador únic, *mensajeid*. No obstant això, tots els missatges referents a una mateixa conversa compartiran la clau forània de l'identificador de xat. D'aquesta manera, un xat pot contenir diversos missatges però un missatge només pertany a un xat.

Un cop dissenyada la base de dades, s'havia de fer el disseny global de la pàgina web. Per això, es fa fer ús del Software de pagament **Axure RP** que permet realitzar un mockup de la pàgina web molt realístic amb les principals funcionalitats. Seguint la premisa d'utilitzar programari lliure, Axure emet una llicència especial per a estudiants [12].

D'aquesta manera, es van identificar i dissenyar quatre vistes d'usuari: **About**, **Home**, **Main** i **UserView**. El disseny inicial de cada una de les vistes i les funcionalitats dissenyades es troben al següent link: Mockup-Video.

Gràcies al disseny de Mockup, es va detectar un dels primers problemes. Per canviar de pàgina, era necessari en molts casos tirar cap amunt, fent l'experiència menys **user-friendly**, un inconvenient tractat a l'apartat contigu.

## 5.1 Adaptabilitat

Aquest subapartat presenta alguns aspectes considerats en forma de complement o com a tecnologia addicional per millorar la qualitat i l'experiència de l'usuari mentre navega per la pàgina web, donant prioritat al *front-end* reflecteixen s'ha comentat abans.

Així doncs, amb la reutilització de codi en forma de components que permet Vue, va ésser molt fàcil incorporar en cada vista un botó dinàmic que tirés cap amunt en el moment desitjat per l'usuari millorant d'aquesta manera la seva experiència. A més a més, la pàgina està dissenyada amb uns colors aptes per a persones que pateixen daltonisme, sense barrejar o superposar colors que confonen [13]. Així mateix, totes les vistes contenen un enllaç a les vistes de Home i About visible a l'inici per facilitar la navegació.

Aquest element també es va fer reaprofitant components de Vue.

### 5.1.1 Inici de Sessió i Registre amb Google

Tenint en compte les dades proporcionades a la introducció sobre l'abandonament per part de l'usuari de la pàgina web i volent millorar l'experiència de l'usuari donant més pes al *front-end*, es va afegir la autenticació d'un usuari (tant registre com login) per una entitat externa, en aquest cas Google.

Aquest fet permet que un usuari es registri i es doni d'alta a la web fent servir només la compta ja creada del seu correu Gmail personal. Doncs, s'accelera el procés d'autenticació, ja que no s'ha d'introduir contrasenya ni s'ha de fer el registre prèviament.

Tanmateix, aquesta autenticació aporta seguretat a la pàgina web tot i no ésser un objectiu considerat.

## 5.2 Aspectes d'implementació del *front-end*

En aquest subapartat es mostren els aspectes clau del desenvolupament de la pàgina web relacionat amb el *front-end*, sobretot la inclusió d'una nova tecnologia per garantir el disseny fet prèviament.

Així doncs, a causa d'utilitzar una eina tant potent com Axure, l'etapa de disseny global de la pàgina web es va allargar. No obstant, quedava molt més definida la pàgina de manera que la implementació seria més curta.

Com ja s'ha esmentat en nombrosos casos, aquest projecte es centra en el *front-end* i doncs, per poder implementar la vista de *Main* com al disseny present al Mockup amb múltiples seleccionables, s'ha hagut de fer ús d'una nova tecnologia: **ElasticSearch**. Existeixen components dissenyats a Vue sota el nom del paquet *ReactiveSearch* que ofereixen les funcionalitats desitjades [14] a l'apartat 3.

Alguns dels avantatges més interessants que atorga aquesta nova tecnologia per al projecte són:

- Proporcionar accessoris declaratius per als escenaris de cerca més habituals: com utilitzar suggeriments automàtics, com gestionar els tipus de cerca, com ajustar mitjançant filtres, aplicar la rellevància.
- Estalviar una gran part de consultes específiques a la BD que s'havien de programar al *back-end* per par cada opció seleccionable de l'usuari.

Així doncs, es va haver d'incloure noves tasques d'aprendre aquesta nova tecnologia i adaptar-la a la BD. De manera que la BD estigui a temps real sincronitzada amb el *Dataset* creat per ElasticSearch proporcionat per la web **appBase.io** [15].

### 5.2.1 Reutilització de components

Aquest subapartat detalla una eina disponible del *front-end* amb la qual es van implementar les vistes de l'usuari.

Gràcies a la utilització del framework Vue.js, la reutilització de codi va ser un element clau en la implementació del *front-end* per mostrar els botons comuns d'inici de sessió a les vistes, un botó flotant per pujar cap amunt present a totes elles així com el peu de pàgina amb la meua direcció

de contacte. Es va crear un HTML un sol cop que s'ha declarat com a **Component** en un altre component més gran que dona lloc als fitxers HTML de les vistes. D'aquesta manera, només calia els tags `<ComponentName ></ComponentName>` per fer ús de tota l'estructura.

Així doncs, les quatre vistes presents: `UserView`, `Main`, `About` i `Home`, cadascuna dona lloc a un component de Vue que a la vegada està format per components de vue més petits.

Per entendre'ns, un component de vue és una instància reutilitzable de Vue que inclou part de codi HTML, i el Javascript i CSS associats a aquell codi.

És comú que una aplicació web s'organitzi en un arbre de components niats [5], en aquest cas, dels quatre components que formaven les vistes se'n deriven un total de deu components niats.

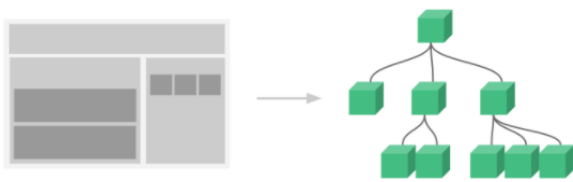


Fig. 2: Aplicació web com a conjunt de components extret de [5].

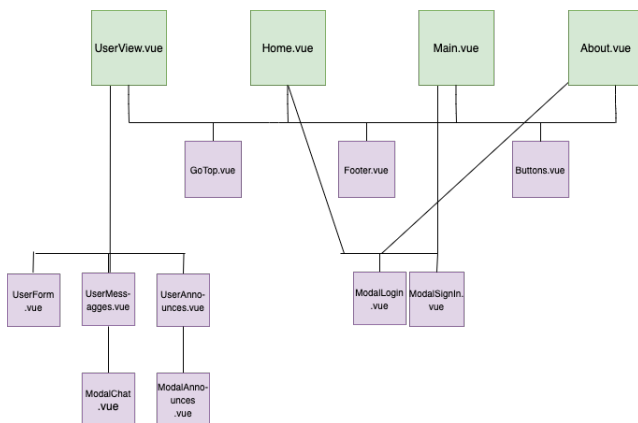


Fig. 3: Conjunt de components de SEP.

### 5.2.2 Eines de Vue utilitzades

Aquest subapartat introdueix les eines del framework utilitzades a la implementació del *front-end*.

- **Vuex:** Vuex és un patró de gestió d'estats + llibreria per a les aplicacions de Vue.js. Serveix com a botiga centralitzada per a tots els components d'una aplicació, amb regles que assegurin que l'estat només es pot mutar d'una manera previsible [16]. Per aquest projecte, Vuex ha suposat poder definir components amb un estat global determinat i accessible des de tots els components per actuar de manera reactiva a inputs de l'usuari variant l'estat del component.
- **Vue-router:** Permet l'assignació dels components a les rutes desitjades. D'aquesta manera, Vue Router sap on i quan presentar-los. Per aquest projecte ha suposat la simplificació de l'encaminament entre vistes.

- **Vue-devtools:** Plugin disponible per la majoria de navegadors que permet debugar aplicacions creades per Vue.js. Per aquest projecte ha suposat poder detectar en quin component eren els errors i visualitzar l'estat d'aquest.

## 5.3 Aspectes d'implementació del *back-end*

Aquesta subsecció tracta els principals punts d'implementació del *back-end*.

Per una banda, s'ha hagut de crear de forma modular una API amb un seguit de funcions associades a una petició HTTP que el client llança contra el servidor. Aquestes peticions són contestades amb respostes en format JSON.

D'altra banda, la programació del *back-end* s'ha fet en Node.js mitjançant el framework Express [9].

Express permet de forma senzilla la integració d'una base de dades. Afegir la capacitat de connectar bases de dades a aplicacions Express només és carregar un controlador Node.js adequat per a la base de dades de l'aplicació disponible a la pròpia web del framework.

## 5.4 Comunicació del *front-end* amb el *back-end*

Aquesta subsecció detalla la comunicació entre *front-end* i *back-end* introduïda en part en l'apartat anterior.

Per tal de passar els paràmetres del *front-end* escollits per l'usuari mentre navega cap al *back-end* i poder fer la consulta a la base de dades es fa servir la *promise Axios*. Una *promise* basada en un client HTTP per node.js [17].

Aquest paquet s'ha d'instal·lar de manera extra tant al *front-end* com al *back-end* i permet dos funcionalitats bàsiques per a la interconnexió:

1. Fer peticions XMLHttpRequests des del navegador cap al *back-end*.
2. Fer peticions HTTP des de node.js
3. Interceptar peticions i respostes.
4. Transformar les dades de peticions i respostes al format desitjat, en aquest cas, JSON.

D'aquesta manera, el *front-end* demana una determinada URL on al *back-end* és interpretada com una crida a una API prèviament implementada.

Les peticions HTTP es poden fer pel mètode de GET o POST. En aquest cas, s'ha decidit fer les consultes pel segon mètode. Un exemple d'aquestes consultes es troben quan es vol llistar els anuncis d'un determinat usuari, eliminar o inserir de nous a la vista *UserView*.

A més a més, aquesta *promise* té un 94% segons [17] de compatibilitats, és a dir, és compatible gairebé amb tots els navegadors i sistemes operatius que s'utilitzen actualment:

Latest ✓	Latest ✓	Latest ✓	Latest ✓	Latest ✓	11 ✓
72 7 ✓	79 7 ✓	11 8.1 ✓	79 10 ✓	9 10.11 ✓	
				10 10.12 ✓	
				11 10.13 ✓	

SAUCELABS

Fig. 4: Navegadors i Sistemes Operatius compatibles amb Axios [17].

## 6 RESULTATS

Aquesta secció agrupa els resultats obtinguts durant el desenvolupament i l'aconseguit de les fites.

### 6.1 Resultats propis de la implementació

Aquest resultats fan referència als documents que es generen durant el disseny i la implementació de l'aplicatiu web.

Així doncs, el primer a mostrar és el del disseny de la Base de Dades que s'utilitza:

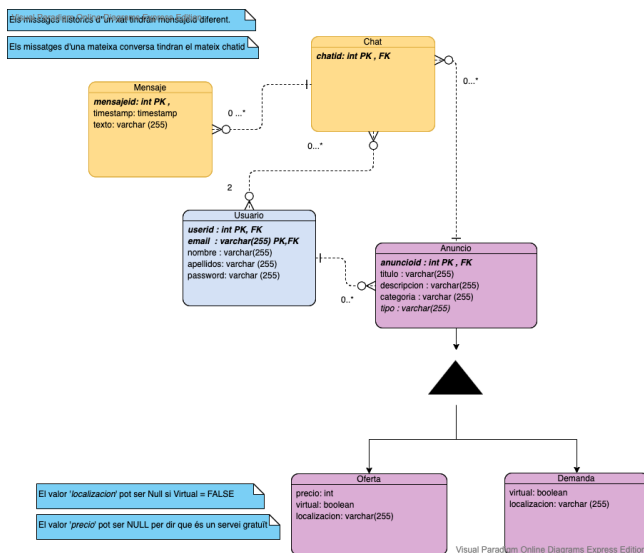


Fig. 5: Diagrama resultant de la Base de Dades.

Un cop fet tot el desenvolupament, és important tenir clar l'esquema global de la pàgina web. Per això s'ha realitzat el següent esquema que resumeix els diferents components integrants d'un disseny de pàgina web i les tecnologies associades que s'han utilitzat així com la situació d'aquestes.

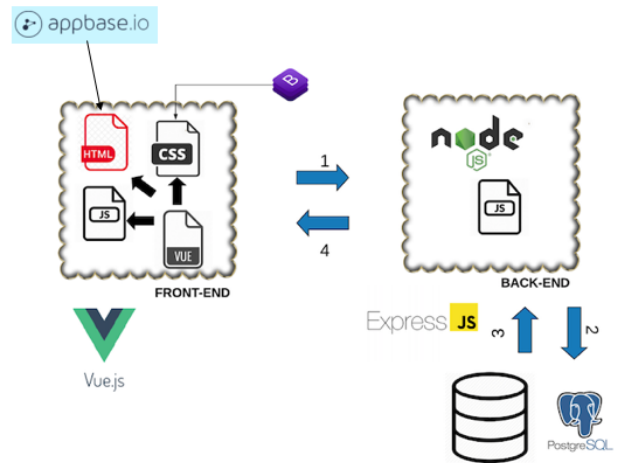


Fig. 6: Esquema global de l'entorn web.

Així doncs, una interacció d'un usuari és recollida pel *front-end* que emet una petició http contra el *back-end* mitjançant Axios (pas 1). Un cop al *back-end*, es redirigeix aquest URL demanada contra la funcionalitat que li pertoca. Dins la funció que executa el *back-end* per donar resposta, s'interacciona amb la base de dades per inserir, recollir o modificar les dades adients (pas 2). La base de dades retorna les dades corresponents al *back-end* (pas 3) on se li donaran el format de resposta mitjançant axios. Per últim, la resposta s'envia al *front-end* per ser presentada a l'usuari (pas 4).

### 6.2 Resultats de la prova pilot i testeig

Aquest subapartat explica els casos d'ús que s'han plantejat per comprovar la funcionalitat especificada a requisits de la Sec 3. Per això, s'ha fet una prova pilot d'agafar un familiar i demanar-li un seguit de funcions a realitzar.

Els casos d'ús es mostren a l'apèndix amb el resultat que s'ha obtingut i la funcionalitat que es vol comprovar a la Taula 2.

### 6.3 Resultats del projecte

Aquest segon subapartat fa referència als resultats referents al projecte en general pel que fa al seguiment de la planificació i possibles modificacions d'aquesta i/o endarreriment de tasques.

El progrés que s'ha seguit en referència a les tasques és el següent:

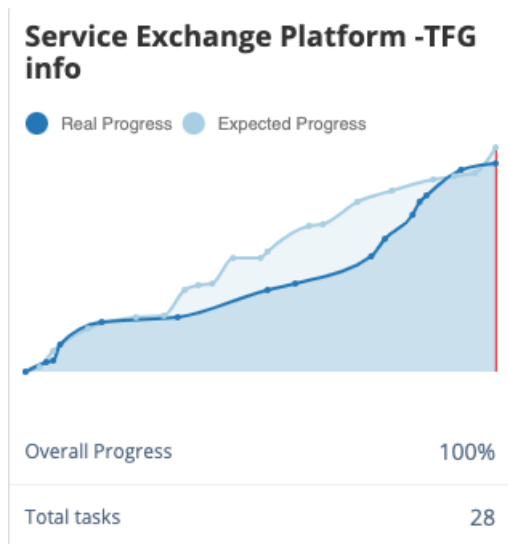


Fig. 7: Progrés del projecte.

Es pot veure un increment del procés més o menys lineal amb un total de 28 tasques on hi ha hagut una petita variació del progrés real versus l'esperat a mitjans de projecte a causa de la introducció de la nova tecnologia ElasticSearch, totalment desconeguda per mi que va suposar una nova etapa d'aprenentatge i una demora en la implementació de la vista *Main* que és on s'utilitza.

## 6.4 Resultats de la prova d'estrès

Es sotmet l'aplicació a una prova d'estrès per tal de detectar un possible coll d'ampolla. La prova d'estrès consisteix en fer inicis de sessions consecutius contra la BD i determinar el temps invertit.

Per dur a terme la prova, es crea un bucle for d'una consulta del tipus "SELECT from table where .." amb diferents quantitats de vegades. Es mesura el temps en mil·lisegons abans i després del bucle. Els resultats es mostren a la següent figura:

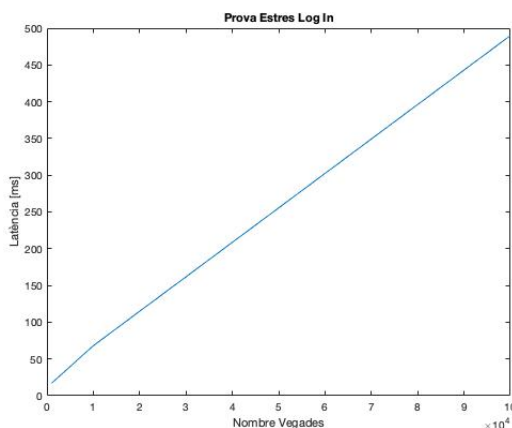


Fig. 8: Prova d'estrès de Log In.

L'eix d'abscisses mostra el nombre de vegades que s'ha fet la consulta mentre que l'eix d'ordenades mostra en mil·lisegons el temps trigat.

Tot i que la latència és baixa (no es supera els 500 ms per 100.000 consultes) es pot veure com el resultat creix lineal-

ment. Aquest fet s'ha de tenir en compte si s'augmenta de manera exponencial el nombre d'usuaris que inicien sessió a l'aplicatiu. Es pot arribar al cas de què el nombre d'usuaris simultanis sigui tan gran que la latència sigui molt alta i es produeixi una caiguda de la BD. Dependent del flux de dades que hi hagi, s'hauria de plantejar l'opció de migrar la BD a un **entorn escalable**.

## 7 CONCLUSIONS

Per tal d'exposar les conclusions, repassarem els objectius i l'abast del projecte de la Sec. 1.1.

Per una banda, s'ha superat satisfactòriament l'objectiu principal de desenvolupar, gestionar i resoldre un projecte de complexitat similar als de la vida real, doncs la selecció de les tecnologies fetes per la pàgina web s'han fet exhaustivament tenint en compte cada detall d'aquesta. Així mateix, les funcionalitats que inclou l'aplicatiu són les típiques funcionalitats que ofereixen plataformes similars.

Pel que fa als subobjectius derivats, S'ha après dues tecnologies que estan en ple creixement: Vue.js i PostgreSQL, d'altres molt utilitzades a les pàgines web actuals com Bootstrap, Express.js i ElasticSearch.

S'ha analitzat, llistat i implementat cada requisit necessari de la pàgina web sempre tenint en compte el temps disponible i la motivació d'aquesta: crear una xarxa virtual de serveis en situacions excepcionals de la societat com la pandèmia global que vivim.

Les tecnologies involucrades s'han triat d'acord amb uns criteris sustentats per informació d'estudis, rànquings, llibres i tesis. Així ho reflecteix la Sec 2. A més a més, per l'interès de centrar-se en *front-end*, s'ha explorat funcions pròpies del framework Vue.js que han pogut agilitzar la implementació present a Sec 5

El disseny de la pàgina web s'ha pogut fer de manera satisfactòria i realista produint el resultat del Mockup de la Sec. 5.2.

S'ha tingut en compte criteris importants a l'hora de crear una pàgina web centrant-se en el *front-end* com l'adaptabilitat dels colors per a daltònics o el dinamisme que proporciona l'autenticació de Google comentats a la Sec5.1.

Un cop presentats els objectius i l'abast, s'ha establert una planificació detallada a Sec 4 on, donada una metodologia, s'ha seguit una planificació de tasques concretes.

Les diferents parts d'implementació amb les seves claus principals han estat comentades a les subseccions de 5.

Per últim, els resultats s'han presentat en quatre nivells: resultats propis de la implementació, resultats del projecte, resultats de la prova pilot i testeig i resultats de la prova d'estrès.

## 8 LIMITACIONS I ASPECTES DEL FUTUR

Aquest apartat tracta sobre les limitacions que presenta l'aplicatiu web i com podrien solucionar-se en un futur.

En integrar l'autenticació amb Google, s'ha utilitzat l'API de Google que permet un màxim de 10.000 peticions al dia de lectura de forma gratuïta, una limitació de 500 peticions de lectura en 100 segons i una limitació de 500 peticions de lectura en 100 segons per usuari.



Així mateix, l'eina d'ElasticSearch proporcionada permet cridar a l'API 100.000 vegades al mes i enregistrar un màxim de 10.000 entrades d'informació de forma gratuïta.

En un futur, si l'aplicació tingués una demanda d'usuaris superiors als límits imposats, es té la opció d'ampliar qual-sevol de les dues APIs per un preu. S'hauria de realitzar un estudi de mercat i mirar quin pla de dades és el que s'ajusta a la demanda.

Per altra banda, com la seguretat no era un objectiu d'aquest projecte, aquesta pàgina web manca de mecanismes de seguretat com ara: encriptació de les dades sensibles i protecció contra *SQL Injeccion*. De cara el futur, es podria parar més atenció i assignar recursos a aquests mecanismes de protecció per tal de garantir la seguretat.

Per últim, la base de dades que guarda les dades de l'aplicació és el meu ordinador, amb recursos de Memòria i CPU limitats. Si es volgués publicar de cara al públic, el meu ordinador hauria de suportar totes les entrades de dades a totes hores i mantenir la base de dades aixecada, algo bastant impossible. S'hauria de buscar una alternativa o bé tenint una base de dades a un servidor o bé buscar algun Cloud amb base de dades virtual escalable.

## AGRAÏMENTS

Aquest projecte no hagués estat possible sense el suport i els ànims incondicionals dels meus pares que, davant una etapa fosca de la meua vida, m'han oferit tota l'ajuda possible. També cal agrair a tots els meus companys el suport rebut i en especial a tu, Dhyagho, per donar-me del teu coneixement sense esperar res a canvi. Per últim, cal agrair al coordinador d'Enginyeria, al Departament d'Enginyeria de la Informació i a les Comunicacions (dEIC) i al meu tutor Sergi l'oportunitat i la confiança que van dipositar en mi per fer el projecte que m'ha fet créixer com a estudiant i persona.

## REFERÈNCIES

- [1] G. Moreno, "¿cuántos sitios web existen?" *Statista.com*, 2018.
- [2] A. Díaz. Ventas y resultados de glovo 2017-2018. [Online]. Available: <https://es.statista.com/estadisticas/811361/>
- [3] E. P. Retina, "Think with google elige: velocidad o abandono," *El Pais*, 2018.
- [4] M. Schwarzmüller, "React vs. angular vs. vue: Which is the best javascript framework?" [Online] (Accedit: 17.04.2020).
- [5] Vue.js. The progressive javascript framework. [Online]. Available: <https://vuejs.org/>
- [6] A. Kyriakidis and K. Maniatis, *The Majesty of Vue. js*. Packt Publishing Ltd, 2016.
- [7] O. Filipova, *Learning Vue. js*. Packt Publishing Ltd, 2016.
- [8] M. Nowak, "Reasons why vue.js is getting more traction every month," *MonteRail.com*, 2018.
- [9] P. SERBY, "Case study: How and why to build a consumer app with node.js," *VentureBeat.com*, 2020.
- [10] D. Engines. Db-engines ranking - trend of relational dbms popularity. [Online]. Available: [https://db-engines.com/en/ranking\\_trend/relational+dbms](https://db-engines.com/en/ranking_trend/relational+dbms)
- [11] P. López Herrera, "Comparación del desempeño de los sistemas gestores de bases de datos mysql y postgresql," *Universidad Autónoma del Estado de México*, 2016.
- [12] Axure. Free licenses for students and teachers. [Online]. Available: <https://www.axure.com/edu>
- [13] A. H. Keeper, "Qué colores confunden los daltónicos," [Online] (Accedit: 27.05.2020).
- [14] S. Kothari, "ReactiveSearch vue 1.0 ui components for search," 2019.
- [15] A. Ayran, "Cli for indexing data from postgres to elasticSearch," [Online] (Accedit: 27.05.2020).
- [16] Vue.js. What is Vuex. [Online]. Available: <https://vuex.vuejs.org/>
- [17] GitHub. Promise based http client for the browser and node.js. [Online]. Available: <https://github.com/axios/axios>

## **APÈNDIX**

### **A.1 Llistat de Versions**

A continuació es llisten les versions utilitzades de SW:

1. Vue.js: 2.6.11 ( Last Estable Version )
2. Express.js : 4.17.1
3. Node.js: 12.16.2 (LTS)
4. PostgreSQL : 12.
5. Bootstrap: v4.4.1
6. Axure RP: 9
7. ElasticSeatch 5.4

### **A.2 Proves de testeig**

Les proves es troben descrites a la Taula 2.

### **A.3 Diagrama inicial de Gantt**

El diagrama de Gantt inicial es pot veure a la Fig. 9.

### **A.4 Diagrama final de Gantt**

El diagrama de Gantt final es pot veure a la Fig. 10.

Requisit a comprovar	Cas d'ús presentat	Resultat
Publicació d'anuncis	S'ha provat com un usuari amb sessió no iniciada no pot publicar un anunci	OK
Publicació d'anuncis	S'ha provat com un usuari amb sessió iniciada pot publicar un anunci seleccionant el tipus, categoria, preu i modalitat (Virtual o Municipi)	OK
Publicació d'anuncis	S'ha provat com un usuari amb sessió iniciada pot publicar un anunci seleccionant adjuntant una imatge des del seu ordinador	OK
Mostrar un tauler d'anuncis	S'ha provat que tant un usuari no registrat com un de registrat poden cada un dels seleccionables per filtrar anuncis	OK
Contactar per xat amb el proveïdor d'un servei	S'ha provat que un usuari amb sessió no iniciada no pugui obrir un xat	OK
Contactar per xat amb el proveïdor d'un servei	S'ha provat que un usuari amb sessió iniciada pugui obrir un xat	OK
Guardar informació a la BD	S'ha provat que el missatge que s'envia referent a un anunci es guarda a la BD	OK
Guardar informació a la BD	S'ha provat que es pot modificar les dades d'un usuari	OK
Guardar informació a la BD	S'ha provat que un usuari registrat pot afegir un anunci a la BD	OK
Guardar informació a la BD	S'ha provat que un usuari registrat pot eliminar un anunci a la BD	OK
Trobar serveis associats a localització	S'ha provat a crear anuncis en una localització i que es mostren a la vista Main	OK
Contactar per xat amb el proveïdor d'un servei	S'ha provat que si s'envia un missatge, el destinatari rep un correu al e-mail personal	OK
Contactar per xat amb el proveïdor d'un servei	S'ha provat que un usuari pot mostrar els seus xats segons l'anunci i contestar	OK

TAULA 2: CASOS D'ÚS PER TESTEIG DE FUNCIONALITATS

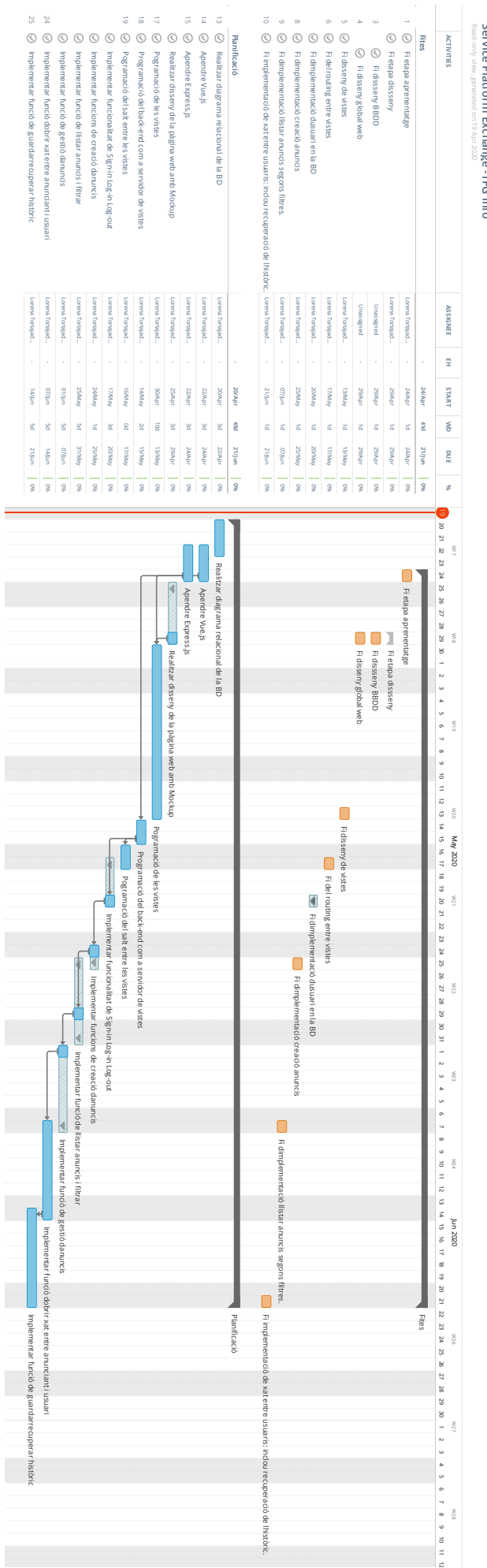


Fig. 9: Diagrama de Gantt inicial.

Service Exchange Platform - TFG info

Resource view generated on 24 Jun 2020

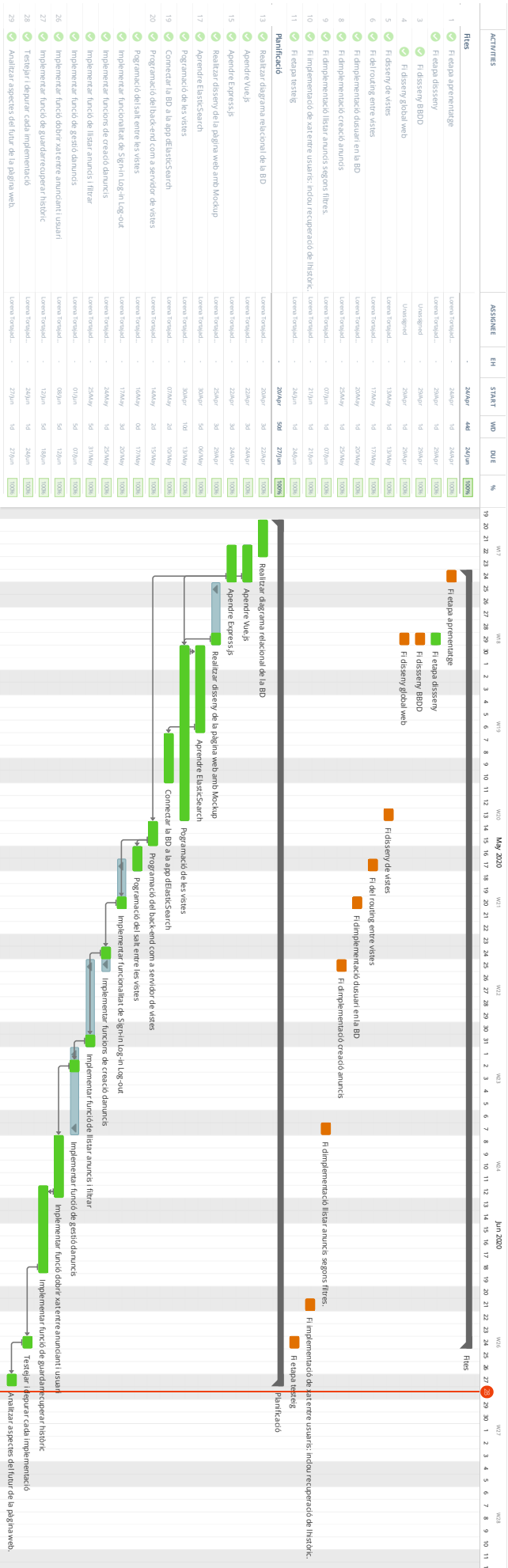


Fig. 10: Diagrama de Gantt final.