

TournamentMaker: creador de lligues esportives amb Laravel i Vue.js

Pablo Hervás Santos

Resum— En aquest document es presenta el desenvolupament d'una aplicació web creadora de lligues esportives enfocada per a empreses, clubs i usuaris comuns. La funcionalitat principal del projecte es donar un suport informàtic a tots els usuaris i administradors que participen en el torneig de forma que tinguin tota la informació necessària d'una manera senzilla, interactiva i intuïtiva. També ofereix la possibilitat de veure l'estat del partit, les estadístiques dels equips i dels usuaris; així com les diferents competicions actives. L'aplicació utilitza la tecnologia *Server Side Rendering* de forma que la part del servidor està desenvolupada amb el *framework* Laravel, un dels més utilitzats i més punters per a la programació lliure en la creació de pàgines web. Mentre que de forma paral·lela, tot integrat dins del mateix entorn, la part del client està desenvolupada amb Vue.js, Bootstrap i HTML5.

Paraules clau — Competició, usuari, administrador, integració, sprint

Abstract— This present document shows the development of a sportive leagues maker web application that is focused on companies, clubs and common users. The main functionality of the project is giving IT support to all the users and administrators that participate in the tournament so that all the information they need is easy, interactive and intuitively available. The web application also offers the possibility of knowing at the match status, team and users' statistics, as well as the different active competitions. The Server Side Rendering technology has been used so the server part has been developed with the framework Laravel, one of the most used frameworks, and cutting-edge technology in open source code for developing web pages. While in parallel, everything integrated in the same setting, the client part has been developed with Vue.js, Bootstrap i HTML5.

Index Terms — Competition, user, admin, Integration, sprint



1 INTRODUCCIÓ

EL treball de fi de grau ha d'assolir les següents competències de la menció de tecnologies de la informació: un programa que satisfaci les necessitats de l'usuari, valorar les necessitats dels clients, observar l'existència de sistemes ja desenvolupats i implementar solucions a aquestes sistemes en l'aplicació.

Tenint en compte la quantitat de competicions que es disputen al llarg de l'any n'hi ha moltes que no gaudeixen d'una tecnologia adient al seu nivell.

Actualment, fora de les competicions oficials com són les diferents federacions o grans tornejos de futbol com el MIC, en la resta no s'implementa l'ús de cap tecnologia per manipular o informar dades als usuaris. Aquests opten per l'ús del paper per mostrar els resultats i classificacions, guardar les dades dels participants i informar de la normativa.

- E-mail de contacte: 96.hervas.p@gmail.com
- Menció realitzada: *Tecnologies de la Informació*.
- Treball tutoritzat per: Joan Giner
- Curs 2019/20

De la manca de tecnologia i de la meua experiència com a jugador, entrenador i col·laborador en diferents esdeveniments esportius a nivell local, va sorgir la idea de crear una aplicació web capaç de gestionar les competicions i en especial, les lligues.

TournamentMaker és una aplicació que ara mateix permet gestionar lligues i en futur tota mena de competicions. L'aplicació està pensada per utilitzar-se de manera local a clubs que durant una setmana o cap de setmana a l'any realitzen el seu torneig anual.

Per dur a terme aquesta gestió, es necessita un compte d'administrador amb llicència. Amb la llicència l'administrador podrà crear lligues amb els equips que s'adhereixin a ella, podrà gestionar els calendaris dels equips, crear els clubs participants, crear usuaris, categories, etc. D'aquesta manera tindrà total control sobre la lliga que ha creat i accés a totes les dades i estadístiques.

Per últim, una de les funcionalitats destacades de l'administrador i que en un futur podrà delegar aquesta responsabilitat a àrbitres o administradors de partits, és la simulació de partits, des del seu inici i edició fins al

tancament. En un futur, també es podrà posar sancions tant a jugadors com clubs o veure un històric de les competicions, entre altres funcions.

D'altra banda, si ets un usuari, és a dir, ets un jugador o un entrenador, tindràs accés a totes les estadístiques de les competicions que s'estan disputant. De la mateixa manera, podràs editar tant el teu perfil com els dels jugadors, en cas de ser entrenador. I d'una forma còmoda, l'aplicació t'indicarà la teva competició per obtenir un accés més ràpid i senzill.

2 OBJECTIUS

En aquesta secció es definiran els objectius de l'aplicació segons la seva importància.

2.1 Objectius principals

Un dels objectius més importants que em vaig proposar a l'iniciar el projecte va ser que la creació de la pàgina web tingués un disseny senzill, intuïtiu i amb les menors complicacions possibles. Això permetria que els usuaris estiguessin còmodes amb la seva utilització i, per tant, sigués una eina útil i fàcilment adaptable.

Una de les funcions més importants de l'aplicació és la de mostrar tota la informació principal de la competició a tots els usuaris que la necessitin o vulguin conèixer-la.

L'administrador ha de ser capaç de simular una lliga des de la seva creació fins a la finalització, passant per la creació d'equips, usuaris i altres administradors prèviament.

Primer de tot en la creació de la lliga, és important que l'administrador de la competició pugui crear-la segons els equips que hi participen amb partits d'anada i tornada, depenent de la categoria que es vulgui implementar.

Una vegada ja creada, és necessari que es puguin simular els partits per obtenir un campió. Per simular-los s'han de poder crear els equips amb els seus usuaris. Una vegada estan creats, l'administrador o l'àrbitre ha de poder indicar les accions més rellevants i tancar el partit.

Després d'haver simulat o finalitzat un partit aquestes dades de les accions i resultats han de poder veure's. Per això, és important en el desenvolupament de l'aplicació, que existeixin les classificacions.

Per últim, l'aplicació havia de diferenciar entre usuaris i administradors de forma que els usuaris no tinguessin els mateixos accessos a recursos que un administrador podia

tenir, per exemple l'edició de la competició o de clubs.

2.2 Objectius secundaris

Com a objectius secundaris de l'aplicació era la possibilitat d'obtenir més d'una categoria en els equips i més d'un rol en els usuaris encara que només es treballa en dos rols.

Com a segon objectiu, es troba la possibilitat de poder veure tots els partits del campionat en format de jornada i la possibilitat de veure un d'ells en detall.

L'últim objectiu secundari era la possibilitat de recopilar i mostrar la informació del usuaris respecte a la competició que disputen. En el cas del futbol serien les targetes, els gols, etc., tot això amb la possibilitat d'ordenar-ho segons el criteri que es desitja.

2.2 Objectius futurs

Actualment, s'utilitzen molt els dispositius mòbils per navegar a través d'internet. Una futura millora seria perfeccionar el disseny per a aquests i implementar una aplicació multi plataforma. Aquest fet, facilitaria l'ús de l'aplicació fent-la més còmoda pels usuaris que la puguin consumir.

La segona funcionalitat que m'agradaria implementar en un futur, seria la possibilitat d'escollir diferents modalitats ja siguin esportives com el basquet, tennis, etc., o d'esports electrònics com FIFA, League of Legends, etc. Això donaria un ampli marge de possibilitats de mercat.

Una altra funcionalitat relacionada amb la anterior, seria implementar diferents tipus de tornejos com eliminatòries i lligues més eliminatòries. Una vegada fet això, també es podria implementar una rutina que segons les preferències de l'administrador la competició s'adapti.

També hi havia la possibilitat de comptar amb temps real la duració d'un partit actiu així com el seu estat per a qualsevol consulta d'un altre usuari que no estigués participant en l'esdeveniment.

Per últim, tot i no ser prioritari, es podrien implementar dos rols més com són l'àrbitre i l'administrador de partits de forma que les estadístiques s'actualitzin al moment i els usuaris puguin estar informats de tots els esdeveniments en directe i de com afectarien als resultats.

3 ESTAT DE L'ART

En aquesta secció s'explicarà com es pot integrar

TournamentMaker en el context actual de tornejos i que pot aportar; i com a part tècnica el seu renderitzat i comparacions amb l'altre tipus de renderitzat que es pot utilitzar.

3.1 Introducció

L'ús de la tecnologia en l'esport és cada vegada més prioritari ja sigui per obtenir uns millors resultats o per tenir una millor infraestructura sobre les dades amb les que es treballen.

El principal problema que es troben les entitats esportives a l'hora d'utilitzar d'implementar aquesta tecnologia, és el seu cost. TournamentMaker, en canvi, està pensat per totes les entitats locals que disputen el seu torneig anual o gestionen lligues setmanals.

Com he esmentat en la introducció, hi ha clubs importants com Jàbac Terrassa, Sant Cugat, Rubi, etc., que quan disputen el seu torneig la informació que arriba tant a pares com a participants és a través de papers. Aquest mètode canviaria totalment amb l'ús de l'aplicació ja que tota aquesta informació es trobaria en un domini del club amb les competicions adients.

En la meua experiència com a esportista que ha participat en moltes competicions, quan s'ha disposat d'eines informàtiques com FCF, Valldor7 o MIC, tothom podia accedir en qualsevol moment a obtenir la informació necessària per ells. En canvi, quan disputava o estava d'organitzador en tornejos que anaven amb paper i fulls d'Excel, tant els participants com jo, necessitàvem estar a prop d'una persona del torneig per saber com anava el partit rival, on es disputava el següent partit, quin dia es jugava, etc.

Per tant, trobo que en les entitats locals sense grans pressupostos anuals falta integrar un mínim de tecnologia que els permeti donar un petit salt de qualitat. I encara més important, el fet de que tots els participants puguin quedar satisfets amb la informació rebuda i el tracte personalitzat.

3.2 Server Side Rendering vs Client Side Rendering

Server Side Rendering

Quan parlem de *Server Side Rendering* o *SSR* ens referim a la renderització de pàgines en el navegador des d'un servidor. Per dur a terme aquesta renderització, s'han de seguir uns passos determinats:

- L'usuari envia una petició a una pàgina web a través d'un navegador.
- El servidor comprova que existeix el recurs, el compila i prepara el contingut HTML.
- Una vegada compilada, la envia a l'usuari i perquè es pugui mostrar per pantalla.
- El navegador descarrega la pàgina i la fa visible a

l'usuari.

- El navegador descarrega els arxius js i els executa per fer la pàgina interactiva.

Tot aquest procés d'obtenir el recurs, convertir-lo i enviar-ho, es fa des del servidor per això rep aquest nom. Això implica que com millor sigui el servidor i la pàgina que ha de renderitzar, menor serà l'espera i el temps de resposta.

Client Side Rendering

Quan parlem de *Client Side Rendering* o *CSR* ens referim a la renderització de pàgines en el navegador des del client. En aquest cas, l'encarregat de fer les tasques que realitzava el servidor les fa el navegador del client. Per dur a terme això, són necessàries llibreries com ReactJs, AngularJs, NodeJs o VueJs. En aquest cas, els passos que realitza per renderitzar són els següents:

- L'usuari fa una petició a la pàgina web a través del navegador.
- L'usuari utilitza un servidor estàtic per suportar l'arxiu Html i els fitxers de l'usuari.
- El navegador descarrega l'Html i després els arxius Js mentre que l'usuari veu el símbol de carregant.
- Una vegada el navegador obté els arxius Js, realitza una petició API per obtenir el contingut dinàmic i ho processa.
- Després de que el servidor doni una resposta, el contingut és renderitzat utilitzant la DOM en el navegador del client.

Comparativa

Una vegada definides les diferents formes d'operar per obtenir el contingut d'una pàgina web, toca comparar-les.

Càrrega d'una pàgina web

Quan parlo de la càrrega d'una pàgina és des del moment que es fa la petició fins que es mostra el contingut de la pàgina en el navegador.

- Primera vegada

Si s'utilitza CSR s'han de descarregar tots els arxius i posteriorment compilar la pàgina ja que quan es fa la cerca l'usuari només té un document en blanc. En canvi, en SSR s'ha de recompilar tot ja que hi existeixen arxius i això triga més que no pas una compilació.

- Segona i posteriors

Si s'utilitza SSR la pàgina ja està en la caché i per tant no s'ha de precompilar, mentre que en CSR s'hauria de compilar la pàgina de nou.

- Pàgina a través d'enllaç

Si és la primera vegada que s'accedeix a aquest enllaç, si s'utilitza CSR al haver descarregat els arxius una compilació és més ràpida que no la recompilació que formula SSR. Per tant, quan s'accedeix per primera vegada a un recurs si

s'utilitza CSR es triga menys que utilitzant SSR. D'altra banda, si la pàgina web utilitza la caché es triga molt menys utilitzant SSR que CSR.

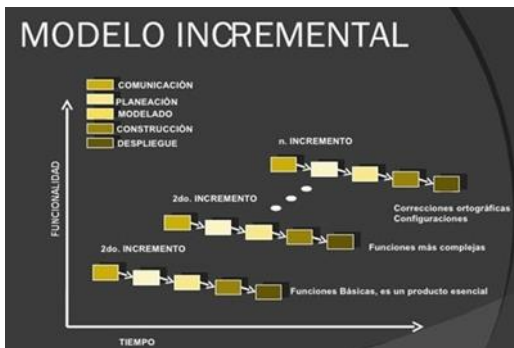
Impacte en SEO (buscadors)

Quan es parla de buscadors es parla de gran quantitat de dades o de metadada. Aquest punt és molt important per a webs d'empreses o webs amb cerques.

Amb CSR, el contingut de la pàgina web es genera utilitzant JavaScript. Això significa que els canvis d'una pàgina a una altra poden estar modificats per l'execució de JavaScript.

Amb SSR, la pàgina sencera està compilada amb la metadada correcta i s'envia després de generar el contingut HTML. Això assegura que la pàgina sempre és correcta sense els possibles canvis que pot provocar JavaScript. Per tant, això implica que SSR sigui una millor solució quan es parla de pàgines de cerca optimitzades.

4 METODOLOGIA



1. Model incremental

La metodologia emprada en aquest treball ha estat la del model incremental, un mètode que es va explicar en l'assignatura de Sistemes d'Informació.

Aquesta metodologia és la més adient pel treball i a la meua manera de treballar, ja que em permet anar tasca a tasca d'una manera organitzada. Les característiques principals que hi destaquen en aquesta metodologia són les següents:

- Els canvis que es produeixen entre versions són petits.
- Les versions s'actualitzen cada poc temps i permet una bona estructura de codi.
- És fàcilment adaptable a les necessitats que es presenten.
- Permet una bona gestió de les tasques en cada etapa.

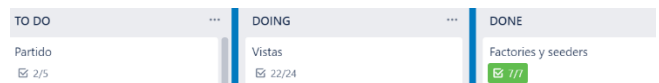
Per dur a terme tot aquest procés i implementar la

metodologia correctament, he utilitzat les eines de Trello i BitBucket.

4.1 Trello

Trello gràcies a les seves funcionalitats ha sigut l'eina que més s'ha utilitzat en aquest treball per fer un seguiment de les tasques. Aquesta eina utilitza la metodologia Kanban, una metodologia àgil, que consisteix en utilitzar etiquetes per obtenir un seguiment de forma visual del projecte i les tasques.

L'aplicació de Trello en els projectes aporta qualitat, una reducció de malbaratament, millora continua i flexibilitat. Per dur a terme això, aquesta metodologia es basa en un taulell amb 3 columnes principals que són: TO DO, DOING i DONE.



2. Trello

Degut a la flexibilitat que ofereix, s'han pogut diferenciar les tasques de *front-end* i *back-end*, a més a més de les diferents funcionalitats de les classes.

4.2 BitBucket

BitBucket és una eina que permet un manteniment de codi de la mateixa manera que ho fa GitHub. La seva principal funció és anar pujant versions de codi a un repositori online cada cop que s'acaba una tasca o quan el programador ho decideix.

Aquesta eina proporciona que els canvis es quedin reflectits en l'aplicació de forma que sempre queda un historial del codi. D'aquesta forma, sempre es pot fer *rollback*, és a dir, tornar a un codi anterior funcional, si en la etapa actual s'han fet tantes modificacions, que és millor tornar al punt inicial, que anar eliminant.

El fet de que sigui un repositori online, permet que el codi sigui accessible des de qualsevol punt amb una connexió a internet. Encara que la modificació de codi ha de ser *offline*. Per fer modificacions, s'ha de descarregar el codi, modificar-lo i posteriorment pujar-ho i sincronitzar-ho.



3. BitBucket

5 PLANIFICACIÓ

La planificació del projecte va quedar definida en 5 *sprints* degut a la metodologia emprada. Aquests *sprints* normalment coincidien amb les entregues de seguiment.

El primer *sprint* consistia en la planificació del treball, la configuració de l'entorn, visualització de tutorials i proves bàsiques sobre els llenguatges que s'han utilitzat i creació de la base de dades i els usuaris.

El segon, consistia en la separació entre usuari i administrador de forma que no tinguessin els mateixos accessos des d'un principi. Dins d'aquest *sprint* també es va incloure la creació de clubs i equips i la afiliació d'usuaris a equips.



4. Segon *sprint*

En el tercer *sprint* es va realitzar la creació de lligues segons les preferències dels administradors, així com la visualització de la classificació. Això influïa en que s'havien de crear els partits de forma que es poguessin veure tots els de la competició.

El quart, tractava de simular els partits i de guardar la informació sobre les accions rellevants dels partits així com gols, targetes, etc., i poder mostrar-les en les estadístiques de jugadors.

L'últim *sprint* consisteix en fer que tot això que era visible i funcional per l'administrador també ho fos per l'usuari segons els seus privilegis. Per tant, havia de poder veure tot el que estigui relacionat amb la seva competició i les diferents que hi existeixen.

6 DISSENY I EXECUCIÓ DE L'APLICACIÓ WEB

En la següent secció es descriuran les principals característiques relacionades amb el desenvolupament del projecte.

6.1 Requeriments funcionals

Registre a la pàgina web (Risc Crític): qualsevol persona pot registrar-se en l'aplicació sempre i quan disposi d'un compte de correu electrònic i pertanyi a un equip. El correu ha de ser únic.

Accés personal a la pàgina web (Risc Crític): l'usuari ha de poder accedir a l'aplicació mitjançant el seu usuari únic i una contrasenya.

Navegar per diferents apartats (Risc Crític): l'usuari, un cop iniciada la sessió, ha de poder navegar pels diferents apartats visibles en el menú superior segons els seus privilegis.

Crear competicions (Risc Crític): l'administrador ha de poder crear competicions segons el nombre d'equips que hi participen.

Crear clubs i equips (Risc Crític): l'administrador ha de poder crear clubs i equips.

Eliminar clubs, equips i usuaris (Risc Crític): l'administrador ha de poder eliminar clubs, equips i usuaris.

Eliminar competicions (Risc Crític): l'administrador ha de poder eliminar competicions, és a dir fer-les no visibles pels usuaris.

Crear rols i categories (Risc Alt): l'administrador ha de poder crear rols d'usuaris i categories en funció del tipus de competicions que es disputin.

Eliminar rols i categories (Risc Alt): l'administrador ha de poder eliminar rols d'usuaris i categories en funció del tipus de competicions que es disputin.

Iniciar i tancar partits (Risc Alt): l'administrador ha de poder iniciar i tancar partits de les competicions que es disputen.

Veure partits i estadístiques (Risc Baix): l'usuari ha de poder veure estadístiques i partits, i els seus detalls, de les competicions que es disputin.

Editar perfil (Risc Baix): l'usuari ha de poder editar les seves dades com la imatge de perfil, contrasenya de manera segura.

6.2 Requeriments no funcionals

Usabilitat: la pàgina web ha de ser segura, ha d'estar protegida davant d'atacs maliciosos, i les contrasenyes dels usuaris han d'estar encriptades en la base de dades.

Visibilitat: la pàgina web ha de ser intuïtiva, tots els requeriments han de ser visibles i fàcils d'entendre per part de l'usuari.

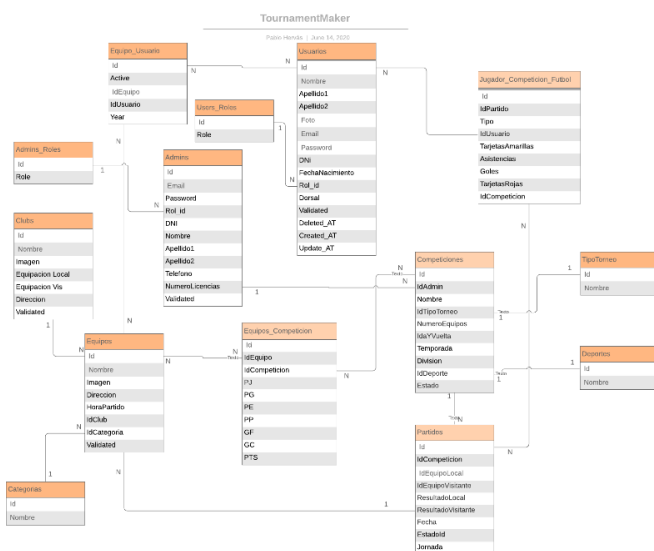
Rapidesa: la pàgina web ha de ser ràpida i, per tant, ha de respondre correctament a les peticions de l'usuari.

Eficiència: la pàgina web ha de ser eficient i ha de retornar la informació desitjada en tot moment.

6.3 Disseny de la base de dades

Una de les parts més importants del desenvolupament, ha sigut el disseny de la base de dades. Un disseny que ha anat canviant a mesura que avançava en el projecte. S'ha desenvolupat amb una base de dades relacional SQL

Estructura de la Base de dades



5. Model Base de Dades

Definició de les taules

- **Usuarios:** entitat ja que conté tota la informació d'un usuari, com el seu nom, l'usuari, l'e-mail, la contrasenya, el rol que indica si és entrenador o jugador.
- **Administradors:** entitat ja que conté tota la informació d'un administrador: el seu nom, l'usuari, l'e-mail, la contrasenya, el rol que indica si és súper administrador o administrador de competicions.
- **Clubs:** entitat que conté tota la informació relacionada amb els clubs com les imatges de les equipacions, adreça, nom, etc.
- **Equipos:** entitat que conté tota la informació relacionada amb els equips com nom, club i categoria que pertanyen, etc.
- **Categorías:** entitat que conté les diferents categories per les competicions.
- **Esports:** entitat que conté els diferents esports que es poden realitzar en les competicions.
- **Competiciones:** entitat que conté la informació relacionada a les competicions.
- **Partits:** entitat que conté la informació relacionada als partits com: marcador, equips, dia i hora.

- **Equipos-Usuarios:** entitat que conté la relació entre els usuaris i els equips.
- **Equipos-Competiciones:** entitat que conté la relació entre els equips assignats a una competició i les seves estadístiques com punts, victòries, etc.
- **Usuari-Competició-Futbol:** entitat que conté la relació entre un usuari i la competició que disputa segons l'esport, en aquest cas: gols, targetes, etc.

6.4 Disseny de l'aplicació web

La part visible de l'aplicació ha estat desenvolupada amb Vue.js que utilitza html5 i classes de css i Bootstrap. La meua intenció principal era que el disseny fos intuïtiu de forma que no fos necessari ser un expert en l'aplicació per saber com funciona i adaptar-se a ella. Això, la faria una aplicació més agradable i l'usuari no sentiria cap tipus de frustració sobre ella.

TournamentMaker Usuarios ▾ Equipos ▾ Competiciones ▾ Otros ▾

6. Menú superior

El disseny està basat en altres aplicacions similars que tracten els esdeveniments esportius ja que són intuïtives i coherents. Per tant, quan es mostren els partits d'una jornada, si es fa clic a sobre d'un, es poden veure els seus detalls. D'igual manera que si es fa clic sobre l'icona de l'equip es mostren els detalls d'ell mateix.

Jornada 1					
Jornada 1 ▾					
	FC Barcelona B	0	0	Madrid A	
	Selección Catalana B	0	0	Rubi A	
	Guadalupe A	0	0	Alavés A	

7. Jornada

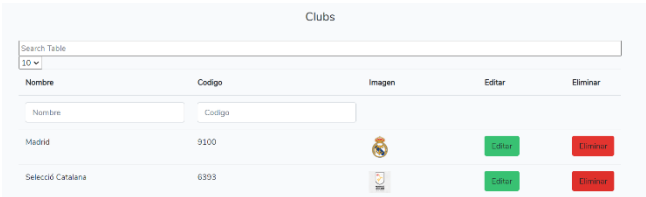
Sobre el disseny des de la part de l'administrador, es tracta de que sigui senzill i ràpid ja que manipularà una gran quantitat d'informació. Per tant, les vistes principals són molt semblants i actuen de la mateixa manera.

Competiciones						
Search Table						
10 ▾						
Nombre	Deporte	Categoría	Jornada	Clasificación	Estadísticas	Eliminar
Nombre	Deporte	Categoría				
prueba	Fútbol	Juveniles		Simulac	Clasificación	Estadísticas Eliminar

1-1 of 1 PrevNext

8. Competicions

Quan es tracta d'usuaris, equips, rols o categories apareixen filtres de cerca i columnes que es poden ordenar a petició de l'administrador de forma que si es vol trobar un valor en concret per eliminar, mostrar o editar es faci ràpidament. Aquestes funcionalitats estan desenvolupades amb Vue.js.



9. Clubs

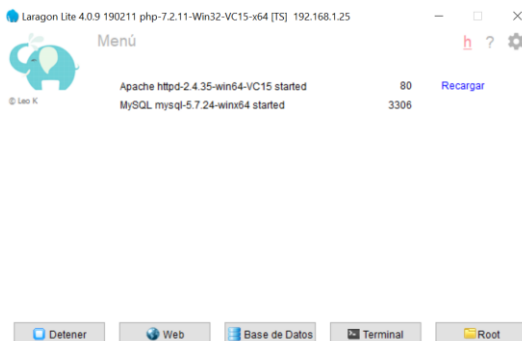
7 SOFTWARE EMPRAT

En aquesta secció explicaré el programari que he utilitzat per desenvolupar l'aplicació.

7.1 Servidor Local – Laragon

Laragon és un entorn de proves que ens permet desenvolupar aplicacions web amb un servidor de proves Apache, el qual està enfocat en el llenguatge de programació PHP.

Utilitza la base de dades PhpMyAdmin i proporciona una terminal per dur a terme comandes de Laravel com migracions o creacions d'entitats, un botó que ens porta a la carpeta arrel dins de l'explorador de carpetes i un accés directe a la nostra base de dades i a la nostra pantalla principal de l'aplicació.



10. Laragon

7.2 Back-end – Laravel

Laravel és un dels *frameworks* més populars actualment de PHP ja que permet el desenvolupament de pàgines web amb gran qualitat. Les seves principals característiques

són:

- Sistemes de rutes i d'abstracció de la base de dades.
- Sistema d'autenticació i gestió de les sessions.
- Manteniment i escalabilitat per a tot tipus de projectes.
- Separació de codi i estil MVC.

Per últim, cal destacar que utilitza un sistema de comandes anomenat Artisan per resumir moltes tasques tedioses i repetitives.

Vaig escollir aquest *framework* sobre altres com Symfony perquè vaig estar mirant la documentació i múltiples opinions, ja fossin tutorials, blogs o papers. Sobretot, després a diferents tutorials vaig veure que tenia una corba d'aprenentatge inferior al principi i que l'estructura que utilitzava s'adaptava a les meves necessitats pel projecte. A més a més, des de la versió 5 està integrat dins del mateix entorn.



11. Laravel

7.3 Front-end – Vue.js

Vue.js és un *framework* on el nucli principal està format per una llibreria encarregada de renderitzar vistes al navegador. La seva forma d'organitzar el codi és per mitjà de petits components que contenen tot l'HTML, CSS i JavaScript necessari per funcionar com a peça independent. Les seves principals característiques són:

- Model MVVM.
- Adaptació a diferents eines com PhpStorm, entorn que he utilitzat per desenvolupar codi gràcies a la llicència de la universitat.

Vaig escollir aquest *framework* ja que després de fer comparatives amb React.js i Angular.js, el vaig trobar més escalable i més senzill de practicar sense coneixements previs. Tot això i l'avantatge de tenir una bona integració amb Laravel van fer que em decantés per ell.

7.4 Integració

La integració que proporciona Laravel amb Vue.js es realitza incloent el fitxer 'app.js' que inclou tots els *components* de Vue.js en la capçalera de les nostres vistes.

Una vegada està inclòs el fitxer, dins de la vista haurem de cridar a l'etiqueta igual que fem amb una de HTML i dins haurem de definir paràmetres i rutes als controladors adients.

8 IMPLEMENTACIÓ

En aquesta secció es parlarà de diferents factors importants respecte a la implementació de la pàgina web.

8.1 Preparació de l'entorn de treball

Per la realització del projecte, s'ha treballat en un entorn local, instal·lant Laragon a l'ordinador. S'ha utilitzat el servidor prèviament esmentat Apache amb la base de dades MySQL i conjuntament amb Laravel versió 7.1.3 i Vue amb la versió 2.5.17.

Per instal·lar Laragon a l'ordinador vaig fer-ho amb l'ajuda de *composer*, que és un gestor de dependències per poder utilitzar de forma més senzilla Laravel. D'aquesta forma totes les característiques que es vulguin instal·lar pel projecte, ja siguin test, plugins, etc.

Una vegada instal·lat, vaig haver de configurar l'arxiu *hosts* que es troba dins de C: users, de forma que definia la *url* d'accés a la pàgina principal i no fos rebutjada.

Per últim, he utilitzat PhpStorm com a editor de codi ja que hi ha un plugin associat a Vue i té una bona integració amb Git per poder utilitzar l'eina de BitBucket. Dins del codi vaig haver de modificar la *url* principal a la definida en el arxius *hosts* i la connexió a la base de dades.

8.2 Casos d'ús

En aquest apartat definiré les principals diferències entre el rol d'usuari i administrador.

Els permisos dels usuaris són els següents:

- L'usuari podrà ser entrenador o jugador.
- L'usuari podrà veure totes les dades relacionades amb les competicions actives.
- L'usuari només podrà editar i eliminar les seves dades.

Els permisos de l'administrador són els següents:

- L'administrador podrà ser administrador de competicions o súper administrador.
- L'administrador podrà veure tota la informació referent a les competicions actives i inactives.
- L'administrador podrà editar la informació d'usuaris, equips, clubs, categories i rols.
- L'administrador serà l'únic capaç de crear, iniciar i finalitzar partits i competicions.
- L'administrador serà l'únic que podrà accedir a dades sensibles.

9 RESULTATS

En la següent secció es presentaran els resultats extrets del desenvolupament del projecte i es destacaran les funcionalitats més importants.

9.1 Funcionalitats del projecte

TournamentMaker ara mateix és capaç d'assolir la simulació d'una lliga de futbol des de l'inici fins al final. Per això, tant l'usuari com l'administrador desenvolupen les següents tasques:

Usuari

- Registre i login.
- Afiliació a un equip.
- Veure les jornades i partits.
- Veure les competicions actives.
- Veure els diferents clubs i equips segons la categoria.
- Veure les estadístiques de la competició, tant classificació com la dels usuaris.

Administrador

- Podrà realitzar les tasques que fa l'usuari.
- Crear tant usuaris com administradors.
- Donar de baixa tant clubs, equips, usuaris i administradors.
- Crear lligues.
- Iniciar, editar i finalitzar partits.
- Crear categories i crear rols.

9.2 Millores respecte les competicions

En aquest apartat es definiran possibles millores que s'implementaran en les competicions.

Fixar temps als partits: la primera d'elles és la capacitat de fixar una duració de partits segons el tipus de partit que sigui. D'aquesta forma en el cas que fossin eliminatòries, es podria fer que els primers partits tinguessin una duració més breu i les finals més llarga.

Selecció d'equips en la creació: la segona seria millorar la selecció d'equips on hi hagi un buscador de forma que no s'hagi de fer *scroll down* per trobar els equips i sigui tot més ràpid.

Inclusió d'àrbitres: la tercera millora seria la possibilitat d'incloure àrbitres en els partits de forma que ells seleccionessin els jugadors que hi participen com a titulars i suplents, que descrigués les sancions del partit i tanqués l'acta amb els codis dels clubs.

10 CONCLUSIONS

Com a conclusions es pot dir que s'han realitzat tots els objectius principals i gairebé tots els secundaris. L'aplicació és funcional des de la part d'administrador, que és l'encarregat de totes les funcionalitats referents a la competició, mentre que des de la part d'usuari només es poden visualitzar dades.

La realització d'aquest treball ha sigut una bona eina pel meu aprenentatge a l'hora de programar i planificar les tasques. A més a més, la programació ha sigut en un llenguatge diferent amb el que treballa, de forma que he après un altre tipus de programació. Encara que em fa dubtar sobre quin tipus de llenguatge m'agrada més o em resulta més còmode, m'ha fet veure que prefereixo ser desenvolupador de *back-end* que de *front-end*.

10.1 Problemes durant la realització

Al llarg del projecte han sortit diferents problemes en quant a la creació de les diferents funcionalitats.

Durant el tercer *sprint*, que va ser la creació de lligues, va suposar una major duració de la establerta des d'un principi. Això va resultar en què el prototip que es volia tenir abans del quart no estigués disponible. Aquesta extensió de temps es va produir perquè tal i com està plantejada la base de dades van sortir diferents errors en la creació i la visualització del calendari amb els equips corresponents, així com la multi selecció dinàmica d'equips en la creació.

El quart *sprint*, que era la simulació de partits, es va allargar degut a l'anterior i a la simulació dels partits. En aquesta simulació vaig tenir dubtes sobre com mostrar i guardar les dades rellevants del partit, així com el seu estat.

AGRAÏMENTS

Per acabar, vull agrair tota l'ajuda que he rebut per part del meu tutor de treball de fi de grau, Joan Giner, gràcies a ell he après diferents tecnologies que no coneixia.

També vull donar les gràcies a la meua parella, als meus amics i familiars que m'han ajudat a definir el disseny i funcionalitat de l'aplicació a través dels seus comentaris i consells.

BIBLIOGRAFIA

[1] React. [En línia] 22 / Octubre / 2019. [Data: 28 / Febrer / 2020.] <https://es.wikipedia.org/wiki/React>.

- [2] Angular. [En línia] 21 / Gener / 2020. [Data: 28 / Febrer / 2020.] [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework)).
- [3] Garbar, Dmitry. Belisoft React Angular. [En línia] 05 / Gener / 2018. [Data: 28 / Febrer / 2020.] <https://belisoft.com/front-end-development-services/react-vs-angular>.
- [4] Alvarez, Miguel Angel. Desarrollo Web. [En línia] 24 / Octubre / 2018. [Data: 19 / Febrer / 2020.] <https://desarrolloweb.com/articulos/laragon.html>.
- [5] Desarrollo Web Laravel. [En línia] [Data: 19 / Febrer / 2020.] <https://desarrolloweb.com/home/laravel>.
- [6] JetBrains. [En línia] [Data: 19 / Febrer / 2020.] <https://www.jetbrains.com/phpstorm/>.
- [7] Desarrollo Web PHP. [En línia] [Data: 19 / Febrer / 2020.] <https://desarrolloweb.com/home/php>.
- [8] B., Gustavo. Hostinger MySQL. [En línia] [Data: 19 / Febrer / 2020.] <https://www.hostinger.es/tutoriales/que-es-mysql/>.
- [9] Genbeta Vue.js. [En línia] [Data: 19 / Febrer / 2020.] <https://www.genbeta.com/desarrollo/por-que-elegir-vuejs-5-razones-para-considerarlo-nuestro-proximo-framework-de-referencia>.
- [10] Segal, Nathan. HtmlGoodies Bootstrap. [En línia] [Data: 19 / Febrer / 2020.] <https://www.htmlgoodies.com/html5/markup/10-common-uses-of-bootstrap.html>.
- [11] Trello. [En línia] [Data: 19 / 02 / 2020.] <https://trello.com/>.
BitBucket. [En línia] [Data: 19 / Febrer / 2020.] <https://bitbucket.org/>.
- [12] BbvaOpen4u Api Rest. [En línia] BBVA, 23 / Març / 2016. [Data: 19 / Febrer / 2020.] <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-sus-ventajas-en-el-desarrollo-de-proyectos>.
- [13] RestfulApi. [En línia] [Data: 28 / Febrer / 2020.] <https://restfulapi.net/http-methods/>.
- [14] FCF. [En línia] [Data: 02 / 28 / 2020.] <http://fcf.cat/es/competicion>.
- [15] Medium. [En línia] [Data: 13 / Juny / 2020.] <https://blog.solutelabs.com/client-side-vs-server-side-rendering-what-to-choose-when-dd1620fb2808>.
- [16] Hemmys Chacon. Medium. [En línia] Medium, 31 / 07 / 2017. [Data: 16 / 04 / 2020.] <https://medium.com/@hemmys25/creaci%C3%B3n-de-un-select-dependiente-con-laravel-5-4-vue-js-select2-957ac043bbdd>.
- [17] Metodologia Kanban. [En línia] GetBillage. [Data: 18 / 04 / 2020.] <https://www.getbillage.com/es/blog/metodologia-kanban-ventajas-y-caracteristicas>.
- [18] Andrade, Luis. Medium. [En línia] 23 / Juliol / 2018. [Data: 28 / Febrer / 2020.] <https://medium.com/drakezair/reactjs-vs-angularjs-vs-vuejs-la-pelea-de-los-grandes-2018-5f4027e61cef> Etc.

APÈNDIX

En l'apèndix es complementarà la informació de l'informe amb més dades sobre la realització del projecte.

A1. ÍNDEX DE FIGURES

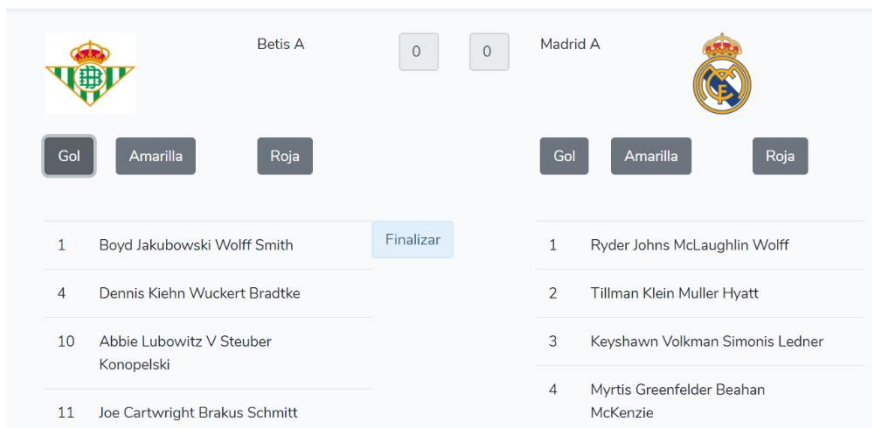
1. Model incremental	4
2. Trello	4
3. Bitbucket	4
4. Segon <i>sprint</i>	5
5. Model base de dades	6
6. Menú superior	6
7. Jornada	6
8. Competicions	6
9. Clubs	7
10. Laragon	7
11. Laravel	7

A2. CAPTURES DE PANTALLA

Partit en actiu des de la part de l'administrador

En aquesta vista s'afegiran les dades més rellevants dels partits. Quan es prem un botó es mostren els jugadors de l'equip per seleccionar-ne un.

Una vegada es prem el botó de finalitzar s'envien les dades i s'actualitzen les estadístiques.



Estadístiques dels participants en la competició

En la imatge de la dreta es mostren les estadístiques dels participants de la competició ordenades per gols. Però si es prem qualsevol altre capçalera s'ordenaran segons aquella preferència.

Estadísticas					
Posición	Jugador	Goles	Tarjetas Amarillas	Tarjetas Rojas	Asistencias
1	Keyshawn Volkman	6	0	0	0
2	Boyd Jakubowski	2	0	0	0
3	Gayle Koelpin PhD	0	0	0	0
4	Emmett Runte IV	0	0	0	0
5	Estella Auer Sr.	0	0	0	0
6	Mrs. Candida Greenfelder	0	0	1	0

Pantalla principal de l'usuari

En aquesta imatge es pot veure la vista principal de l'usuari que participa en una competició indicant quin serà el següent partit a disputar.

