

---

This is the **published version** of the bachelor thesis:

Cruz Gargallo, Arnau; Martí Godia, Enric, dir. YourLeagues : Aplicació web per a crear i gestionar lligues i tornejos. 2022. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/264169>

under the terms of the  license

# YourLeagues: Aplicació web per a crear i gestionar lligues i tornejos

Arnau Cruz Gargallo

**Resum**— L'objectiu d'aquest projecte es dissenyar i implementar una plataforma online anomenada YourLeagues per a poder gestionar competicions de diferents esports, permetent així que grups d'amics o centres esportius amb menys recursos puguin gestionar les seves pròpies competicions de manera ràpida i senzilla. Amb YourLeagues es pot ser l'administrador d'una competició registrant-se a la aplicació web i creant un torneig o una lliga. També és possible gestionar els calendaris, els equips i els resultats, per tal de garantir un bon funcionament de la competició. Per altra banda, un usuari també pot ser participant d'aquestes lligues, on pot registrar el seu equip, invitar-hi jugadors i unir-se a les diferents lligues ja creades enviant una sol·licitud de participació. En aquest article s'explica com va sorgir la idea de YourLeagues, els objectius de la plataforma, el seu desenvolupament, les conclusions i les millores del projecte en un futur.

**Paraules clau**—Angular, aplicació web, classificació, competició, equips, Firebase, resultats.

**Abstract**— The main goal of this project is to design and implement an online platform called YourLeagues, this platform allows you to manage different kind of sports competitions, so groups of friends or sports centers with less budget can manage their own competitions in an easier and faster way. YourLeagues allows you to be the manager of a competition by just registering in the application and creating a tournament or a league related to the most practiced sports in the world, also allows you to manage the calendar, the teams and the results in order to guarantee a proper functioning of the competition. On the other hand, YourLeagues also gives you the possibility to be a competitor from those leagues, where you can register your team, invite players and join different leagues by just sending a joining request. In this report I will explain the motivation of YourLeagues, the goals, the design and the development of the platform. Furthermore, I will give the conclusion of the project and the improvements for the future.

**Index Terms**—Angular, competition, Firebase, results, standings, teams, web application.

## 1 INTRODUCCIÓ

El projecte sorgeix pensant en les diferents competicions d'esports en l'àmbit amateur, on usualment en aquest nivell no es tenen els recursos tecnològics ni econòmics suficients per a poder disposar d'una aplicació que gestioni la seva competició. És per això que sorgeix YourLeagues, una aplicació web que permet la creació i gestió de competicions de forma gratuïta.

YourLeagues, a part de proporcionar una eina a les petites organitzacions d'esport on poder crear i gestionar competicions de manera totalment gratuïta, també permet a grups de persones o grups d'amics, fer lligues entre ells de manera intuïtiva.

Abans de definir els objectius de l'aplicació, s'ha realitzat un estudi de les diferents aplicacions relacionades amb YourLeagues ja existents en el mercat, concretament s'han triat 3: DoLeague [1], GestorLigas [2] i Xporty [3].

Les tres aplicacions són aplicacions web que permeten la creació i gestió de competicions de qualsevol tipus d'esport. Aquestes aplicacions permeten el registre d'un usuari, on aquest podrà ser organitzador o jugador. El procés per a crear una competició és molt semblant en les tres aplicacions, consta de 3 fases: la primera és introduir la informació general de la competició, la segona és el període d'inscripció, on s'inscriuen els diferents equips que formaran part de la competició i l'última fase és l'inici de la competició, on es realitzen els encreuaments i calendaris de manera automàtica.

Per a realitzar un estudi més detallat a la taula 1 es mostren diferents característiques de les aplicacions per a poder comparar-les entre elles.

	DoLeagues	GestorLigas	Xporty
Crear Lligues	✓	✓	✓
Crear Tornejos	✓	✓	✓
Aplicació actualment activa	X	X	✓
El jugador introdueix les dades del seu equip manualment	X	X	✓
Rols Jugador i Organitzador separats	X	X	✓
Totes les funcionalitats són gratuïtes	✓	X	X
Disposa d'un xat per a cada partit	X	X	X

Taula 1 - Comparativa de característiques.

- E-mail de contacte: [arnau.cruz.gargallo@gmail.com](mailto:arnau.cruz.gargallo@gmail.com)
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Enric Martí Gòdia (Departament Ciències de la Computació)
- Curs 2021/22

Com es pot veure a la taula l'aplicació més completa és Xporthy, la qual té pràcticament totes les funcionalitats principals que proposa YourLeagues, les quals es presenten a continuació.

El principal objectiu del projecte és el disseny, la implementació i el test d'una aplicació web que permet la creació i gestió de diferents competicions esportives, siguin lligues o tornejos dels següents esports: Futbol 7, Futbol Sala, Futbol 11, Bàsquet, Pàdel, Tennis, Handbol i Voleibol.

Les funcionalitats principals que cobreix el projecte són:

- Registre i connexió dels usuaris.
- Crear equips.
- Gestionar els equips (modificar l'escut de l'equip, convidar altres usuaris als equips, eliminar usuaris de l'equip, etc).
- Crear tornejos i lligues amb certs paràmetres de personalització com ara indicar si serà a doble volta o a volta única.
- Unir-se a les diferents lligues i tornejos.
- Consultar classificacions, resultats, estadístiques, emparellaments o equips de les diferents competicions.
- Gestionar les lligues i els tornejos que l'usuari hagi creat (començar la lliga, introduir els resultats, introduir les dates per a cada partit, ajornar partits, etc).
- Generar de forma automàtica els emparellaments i els calendaris de la competició.
- Processar els resultats introduïts per l'administrador de la competició i actualitzar la classificació de manera automàtica.

Les eines de desenvolupament que s'utilitzen en el projecte són les següents:

- **Angular** [4] és un framework opensource fet per google, per al desenvolupament d'aplicacions web. Aquest framework permet crear una interfície web SPA *Single Page Application*, és a dir una aplicació d'una sola pàgina dinàmica on el que canvia són solament les vistes, fent així l'efecte d'una pàgina web tradicional. Angular també ens permet realitzar el front-end de la nostra aplicació web, és a dir, la creació d'interfícies i components visuals que permeten a l'usuari interactuar amb l'aplicació web. Els principals llenguatges de programació que es fan servir són Typescript, Html i Css.
- **Firestore** [5] és un "back-end as a service" desenvolupat per google. Firestore ens proporciona l'eina *Firestore Database*. Aquesta eina ens proporciona la infraestructura necessària per a crear una base de dades no relacional sense la necessitat de desenvolupar el codi back-end estàndard. A més, també permet allotjar la nostra aplicació web a un domini de forma totalment gratuïta gràcies a l'eina *Firestore Hosting*.

Els entorns de desenvolupament que s'utilitzen en el projecte són els següents:

- **Visual Studio Code** [6]: Editor de codi font, el qual facilita el desenvolupament del codi gràcies a les seves extensions de Firebase i Angular.
- **GitHub Desktop** [7]: Versió d'escriptori de GitHub. Aquest entorn permet tenir una traçabilitat del codi desenvolupat, ja que al tenir un control de les diferents versions es pot veure quines modificacions s'han realitzat en cada versió.
- **Trello** [8]: Aplicació web que s'utilitza per a l'administració de projectes i fa la funció de taulell, permetent així tenir una traçabilitat de les tasques pendents, les tasques en procés i les tasques finalitzades.
- **MarvelApp** [9]: Aplicació web per a dur a terme dissenys i prototips per a aplicacions web i mòbil.

Per a la realització del projecte hem triat la metodologia waterfall o en cascada [10], que consisteix en dividir el treball en un conjunt de fases que es realitzen de forma seqüencial.

El nostre projecte consta de 5 fases:

1. Disseny de l'aplicació.
2. Disseny de la interfície.
3. Implementació i test.
4. User testing i manteniment.
5. Memòria i presentació.

Al final de cada fase es crea un lliurament que permet fer un seguiment de les tasques realitzades. Abans del començament de cada fase es realitza un estudi de les diferents tasques a realitzar i s'afegeixen al taulell virtual de Trello, facilitant així el seguiment de les tasques.

Hi ha una fase del projecte que es fa de manera iterativa: la fase d'implementació i test. En aquesta fase es fa un desenvolupament incremental, és a dir es segmenta l'aplicació en diferents versions, on cada versió implementa un conjunt de funcionalitats de l'aplicació. Finalitzada una versió es realitza el test d'aquesta per així poder validar-la, i per tant, poder passar a la següent versió. D'aquesta manera, si es produeixen errors, serà més fàcil detectar-los i corregir-los que no pas al final del desenvolupament.

En el capítol 2 s'explica en detall el funcionament del framework angular, els diferents mòduls que conté i per a què serveix cadascun d'ells. En el capítol 3 s'explica el disseny de la base de dades i quines col·leccions s'han triat. En el capítol 4 es detalla com s'ha desenvolupat l'aplicació i es mostren els diferents mòduls implementats. En el capítol 5 es mostren els resultats obtinguts durant el desenvolupament del projecte, i s'explica com ha realitzat l'user testing de l'aplicació. Finalment, el capítol 6 conté les principals conclusions del projecte i les possibles millores.

## 2 COM FUNCIONA ANGULAR

Un projecte d'Angular generalment està format pels següents mòduls:

- Els **components** contenen totes les vistes i funcionalitats de l'aplicació, cada component està format per quatre fitxers:
  - Un fitxer *.html*, el qual conté la interfície del component que serà visible a l'aplicació.
  - Un fitxer *.css*, que conté els estils que s'apliquen al fitxer html per tal de donar-li l'aspecte visual al component.
  - Un fitxer *.ts*, que conté la lògica de la vista on s'inclouen totes les funcions que té un component.
  - Un fitxer *.spec.ts*, que conté totes les proves unitàries relacionades amb el component.
- Els **models** són estructures de dades que permeten guardar les dades en objectes, facilitant així la manipulació i l'accés.
- Els **serveis** estan formats per funcions i s'encarreguen d'obtenir dades de la base de dades i entregar-les als components.
- Els **Guards** s'utilitzen per a vigilar que un usuari no pugui accedir a determinades rutes de l'aplicació com per exemple que un jugador no pugui accedir a la ruta de crear una competició.
- Els **Resolvers** tenen com a funció que un component no es renderitzi fins que no s'han rebut les dades d'un servei, evitant així errors de disseny dins de l'aplicació, com per exemple que no es carreguin les dades d'un usuari i, per tant, no es puguin mostrar dades com el nom o la imatge d'aquest.

Per a explicar millor el funcionament d'angular es mostra a la figura 1 l'arquitectura de l'aplicació:

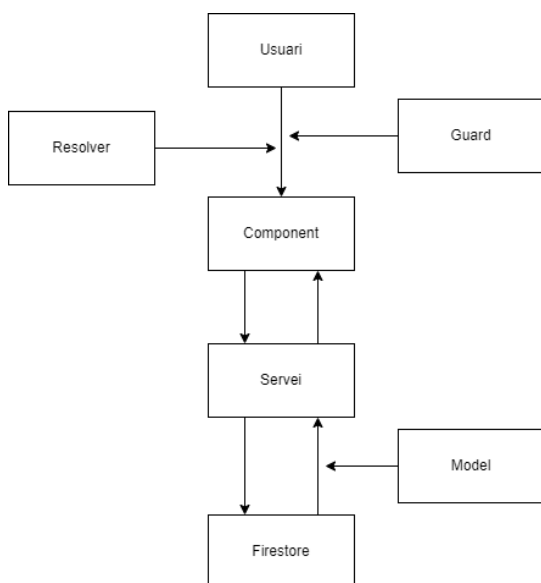


Figura 1 - Arquitectura de l'aplicació

1. L'usuari accedeix a una ruta de l'aplicació.
2. Si la ruta té un Guard, es comprova que l'usuari que intenta accedir al component tingui l'accés permès.
3. El Resolver bloqueja la vista a l'usuari fins que no es rebin les dades.
4. El Component accedeix al servei i crida a la funció que realitza la consulta per a rebre les dades necessàries.
5. El Servei fa una lectura a la base de dades de Firebase anomenada *Firestore Database* i rep les dades.
6. Les dades rebudes es modelen i es guarden en el Model corresponent.
7. El Servei envia l'objecte on s'han guardat les dades al component.
8. El Resolver desbloqueja la vista a l'usuari.

## 3 DISSENY DE LA BASE DE DADES

Com que el projecte està en fase de prototipatge i, per tant, el tràfic de dades és baix, s'ha triat la quota gratuïta de Firestore Database, la qual permet emmagatzemar fins a 1 gigabyte de dades i realitzar un màxim de 50.000 lectures per dia [11], el qual serà suficient per a les primeres versions de l'aplicació.

Per a realitzar el disseny de la base de dades, en primer lloc, s'han recollit els requisits de dades de l'aplicació en un document, que es troba a l'apèndix A.1. Per a dur a terme el disseny d'una base de dades no relacional s'ha seguit el disseny proposat per dataversity [12]. En la figura 2 es mostra el diagrama de col·leccions que formen la base de dades de l'aplicació.

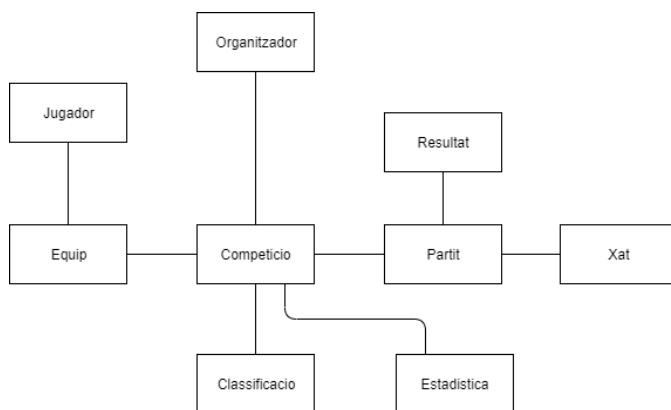


Figura 2 - Diagrama de col·leccions

En total s'han triat 9 col·leccions:

- **Jugador:** guarda totes les dades d'un usuari com el nom, cognom, mail, contrasenya, població, edat i altres dades pròpies d'un jugador com són l'edat i el pes.
- **Organitzador:** emmagatzema totes les dades d'un usuari i a part també guarda dades pròpies d'un organitzador com ara el nom de l'organització, l'email i el número de telèfon.
- **Equip:** aquesta col·lecció inclou dades com el nom de l'equip, l'usuari capità, els jugadors, la imatge de l'equip i la seva abreviació.

- **Competició:** conté camps com ara els equips, l'organitzador de la competició, el nom, la imatge, les sol·licituds el tipus de competició i el tipus d'esport.
- **Classificació:** emmagatzema la puntuació d'un equip, els gols a favor, els gols en contra, els partits guanyats, empatats, perduts i jugats.
- **Estadística:** inclou dades pròpies de cada jugador dins d'una competició com són les targetes grogues, les targetes vermelles i els gols marcats.
- **Partit:** aquesta col·lecció guarda l'horari del partit, l'equip local, l'equip visitant i el número de jornada a la qual pertany el partit.
- **Resultat:** aquesta col·lecció emmagatzema el guanyador del partit, els events que han succeït durant aquest, com ara les targetes o els gols i els resultats de cada part disputada.
- **Xat:** guarda la id dels dos capitans, la id de l'organitzador i tots els missatges enviats.

#### 4 DESENVOLUPAMENT DE L'APLICACIÓ

Previ al desenvolupament de l'aplicació s'han recollit tots els requisits funcionals que l'aplicació haurà d'incorporar per a poder garantir un bon funcionament. A més, s'ha assignat un grau de prioritat en funció de la importància de cada requisit, assignant "A" com a màxima prioritat i "C" com a mínima prioritat. Aquests requisits es poden veure a l'apèndix a l'apartat A.2.

Recollits tots els requisits funcionals de l'aplicació, s'han descrit 20 casos d'ús que permeten descriure el comportament de l'aplicació assegurant-nos així el compliment de tots els requisits.

Coneguts tots els casos d'ús i tots els requisits funcionals, s'ha realitzat un disseny de 35 pàgines [13]. Aquest disseny s'ha portat a terme gràcies a l'eina *MarvelApp*.

Una vegada finalitzats els dissenys previs, s'ha començat la implementació de l'aplicació web, on s'han seguit els següents procediments per a crear el projecte:

1. S'ha creat un repositori a github [14], que permet tenir un control i seguiment del codi.
2. Amb l'eina Visual Studio Code s'ha creat el projecte d'angular el qual conté tot el codi font de l'aplicació.
3. S'ha creat l'estructura de carpetes del projecte seguint aquesta estructura:
  - app
  - Components
  - Models
  - Serveis
  - Utils
  - Guards
  - Resolvers

4. Creació del projecte a la plataforma Firebase la qual permet crear la base de dades utilitzant *Firestore Database*, i allotjar la aplicació web al núvol utilitzant l'eina *Firebase Hosting*.
5. Finalment, s'ha vinculat el projecte d'angular amb el projecte de Firebase, per tal de facilitar les interaccions amb la base de dades.

Com s'ha esmentat prèviament en la metodologia del projecte, la fase de desenvolupament es divideix en sprints. Per tant tots els elements de l'aplicació s'han desenvolupat en dos sprints, és a dir dues iteracions.

Cada iteració consisteix en el desenvolupament de diferents funcionalitats per tal de cobrir certs requisits funcionals de l'aplicació. Abans de cada iteració s'han definit quines funcionalitats s'hauran de realitzar durant el sprint i s'han afegit al taulell virtual del Trello.

Finalitzat el desenvolupament d'aquestes funcionalitats, s'han realitzat proves unitàries, que consisteixen en passar diferents checklist que verifiquen el bon funcionament de l'aplicació web. Aquestes checklist són diferents condicions que ha de complir el component, les quals permeten detectar errors greus en el desenvolupament.

Una vegada finalitzat un sprint, abans de començar el següent, s'han corregit tots els errors en el desenvolupament que ens marquen les checklist, per així garantir un bon funcionament de l'aplicació.

A la taula 2 es mostra com a exemple la checklist del component *home-page*:

Prova realitzada	Estat
La pàgina home-page canvia de forma si un usuari es connecta o desconnecta.	✓
La pàgina home-page canvia de forma depenent del rol de l'usuari.	✓
La informació recuperada en la home-page correspon a l'usuari connectat a l'aplicació	✓
La informació de la home-page s'actualitza automàticament si s'afegeix un nou equip o una nova competició.	✓
Les rutes de veure detalls de competició o equips funcionen correctament.	✓

Taula 2 - Checklist home-page

En total s'han desenvolupat 27 components, 11 models, 6 serveis, 2 resolvers i 2 guards, els quals podem observar en el diagrama de mòduls a la figura 3.

A continuació s'expliquen tots els elements desenvolupats: els Components, els Models, els Serveis, els Resolvers i els Guards.

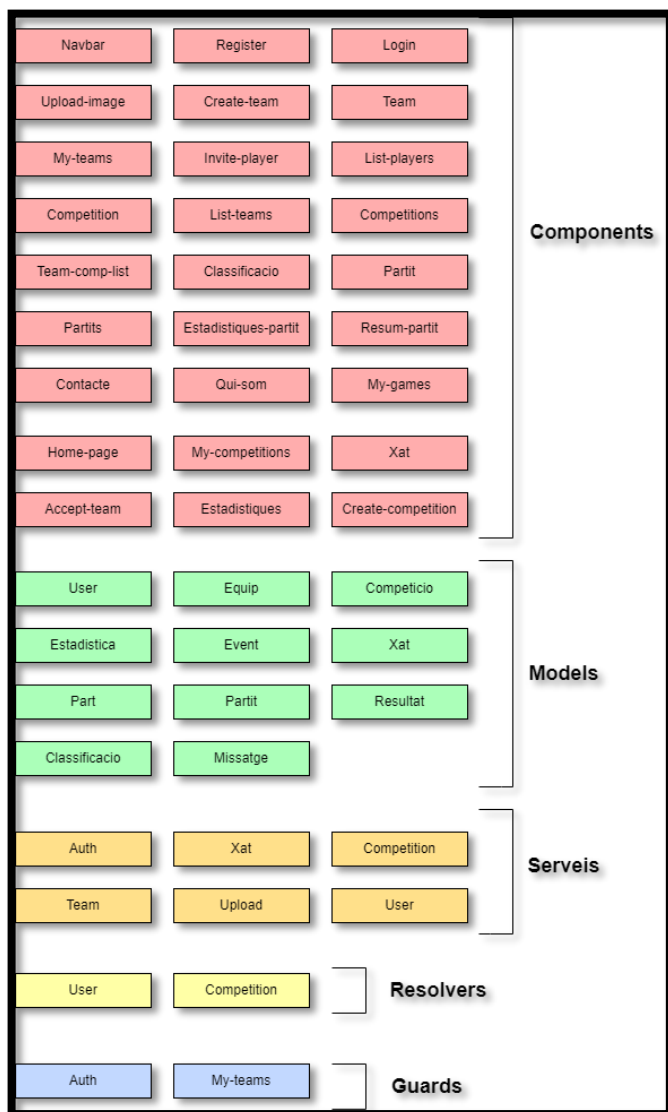


Figura 3 - Diagrama de mòduls de l'aplicació

#### 4.1 COMPONENTS

En aquesta secció s'expliquen els components més importants que s'han desenvolupat:

- La **navbar** consisteix en una barra de navegació que serà visible en tot moment a la part superior de la pàgina web. Aquesta navbar pot adoptar diferents formes depenent del rol de l'usuari connectat.
- El component **register** permet la creació d'un usuari per a que posteriorment pugui identificar-se dins de la pàgina web.
- El **login** consisteix en un formulari de login que permet autenticar un usuari dins l'aplicació, permetent així mostrar diferent informació en funció de les dades emmagatzemades a la base de dades.
- **Home-page** és la pàgina principal de l'aplicació web. Aquest component pot adoptar diferents formes depenent si el visitant de la pàgina web està connectat o no, i depenent del rol d'usuari, organitzador o jugador.

- **Upload-image** permet pujar una imatge al núvol de Firebase, per a després poder mostrar-la dins de l'aplicació web. Aquest component s'utilitza posteriorment en altres components.
- **Create-team** consisteix en un formulari, on s'introdueixen dades com ara el nom, l'abreviació i l'escut de l'equip que es vol crear.
- **Team** consisteix en la pàgina principal d'un equip. Aquest component està format per 3 components més: **invite-player**, que ens permet convidar un jugador a l'equip introduint l'email del jugador, el component **list-players** que permet llistar tots els jugadors que formen part d'un equip i finalment el component **team-comp-list** mostra totes les competicions a les quals forma part un equip.
- El component **my-teams** consisteix en una llista dels diferents equips que forma part l'usuari connectat. Aquest component també conté totes les invitacions als diferents equips.
- **Accept-team** és el component que permet acceptar la invitació a un equip. Per a acceptar la invitació cal omplir un formulari on el jugador indica quin dorsal vol utilitzar dins l'equip.
- **Create-competition** tracta d'un formulari que permet crear una competició. Té diferents camps com ara el nom de la competició, el tipus d'esport o el tipus de competició.
- **Classificació** mostra la classificació d'una lliga, on per a cada equip mostra la puntuació obtinguda durant la competició, els partits guanyats, perduts, jugats i els gols/cistelles/sets a favor i en contra. A part també ordena de forma descendent els equips en funció de la seva puntuació, deixant així a la part superior de la classificació a l'equip amb més punts, i a la part inferior de la classificació a l'equip amb menys punts.
- **Estadístiques** mostra el número de targetes grogues, targetes vermelles i el número de gols de cada jugador, aquest component únicament serà visible per a competicions de futbol, és a dir Futbol 11, Futbol 7 o Futbol Sala.
- **Resum-partit** és un dels components més importants de l'aplicació ja que tracta de la pàgina dels partits on s'introduiran els diferents events obtinguts

#### 4.2 MODELS

Els models són estructures de dades que emmagatzemen les dades obtingudes de la base de dades en objectes, això permet que posteriorment quan un component necessiti accedir a les dades ho faci de manera ràpida i senzilla.

En aquesta secció s'explicaran alguns dels models desenvolupats:

- El model **Usuari**, s'encarrega de guardar les dades obtingudes a la base de dades d'un usuari en objectes. Aquest model pot adoptar diferents formes, la forma principal és *User*.

User té aquesta estructura:

```
User {
  id: string;      localitat: string;
  email: string;   rol: string;
  nom: string;     img: string;
  cognoms: string; sexe: string;
  dni: string;     date: Date;
}
```

Jugador hereta de la classe User i té els següents atributs extres:

```
Jugador extends User{
  altura: number;
  pes: number;
  equips: string [];
  invitacions: string [];
}
```

Organitzador també hereta de la classe User i té els següents atributs extres:

```
Organitzador extends User{
  orgName: string;
  orgEmail: string;
  orgTelefon: string;
  orgDesc: string;
  competicions: string [];
}
```

**Competició** té la següent forma:

```
Competicio {
  id: string;      tipusSport: string;
  organitzador: string; tipusCompeticio: string;
  nom: string;     img: string;
  email: string;   sollicituds: string[];
  privacitat: boolean; equips: string[];
  numEquips: number; nomOrganizacio: string;
  estat: string;
}
```

### 4.3 SERVEIS

En aquesta secció s'explicaran alguns dels serveis més importants i més difícils d'implementar de l'aplicació així com algunes de les seves funcions.

El primer servei que s'ha implementat ha estat el servei **auth-service**, que conté bàsicament totes les funcions relacionades amb la creació i gestió d'usuaris. Algunes funcions destacades que s'ha implementat dins d'aquest servei són: *InsertUser()* que s'encarrega d'inserir les dades del formulari de registre a la base de dades, *isLoggedIn()* la qual retorna si el visitant de la pàgina web està connectat amb un usuari o no o *login()* que s'encarrega d'autenticar un usuari dins de la pàgina web.

Un altre servei molt important que s'ha implementat ha estat el de **user-service**, que té totes les funcions relacionades amb les dades de l'usuari, com ara la funció *getUserById()* o *getUserByEmail()* que són dues funcions que s'encarreguen de retornar les dades d'un usuari mitjançant el seu email, o la seva id. Una altra funció molt important d'aquest servei és *getUserInvitation()* que s'encarrega de retornar totes les invitacions d'equips que té un usuari.

El servei més important és **competition-service**, aquest servei conté totes les funcions relacionades amb les competicions. A continuació s'expliquen algunes de les funcions més importants que implementa aquest servei:

- *CreatePartits()*, aquesta funció s'encarrega de crear tots els partits d'una competició un cop s'ha iniciat a partir de la llista dels equips inscrits. Per a crear un partit també ha de crear el xat d'un partit i els resultats.
- *CreateClassificacio()* i *CreateEstadistiques()* tenen com a funció crear la classificació i les estadístiques de la competició. Per a dur a terme aquestes tasques s'inicialitza a 0 totes les dades de la classificació i de les estadístiques de cadascun dels equips.
- *UpdateResultatGeneral()*, s'encarrega d'actualitzar el resultat d'un partit i també actualitzar la classificació i les estadístiques. La dificultat d'aquesta funció ha estat modificar la classificació una vegada es modificava un resultat.
- *RecalcularGuanyador()*, aquesta funció es crida una vegada s'ha actualitzat el resultat d'un partit i s'encarrega de recalcular el guanyador a partir del recompte de gols o punts d'un partit.

### 4.4 GUARDS

En aquesta secció s'expliquen els guards desenvolupats:

- **Auth-guard** s'encarrega de que un usuari connectat no pugui accedir a la ruta `/login` o `/registre`.
- **My-teams** fa possible que un usuari no connectat no pugui accedir a la ruta `/my-teams` doncs no ha de ser possible que un usuari no connectat formi part de cap equip.

### 4.5 RESOLVERS

Els resolvers desenvolupats són els següents:

- **User-resolver** permet que un component no es carregui fins que no s'han rebut les dades d'un usuari i
- **Competition-resolver** fa possible que qualsevol component relacionat amb les competicions no es carregui fins que no s'hagin obtingut les dades necessàries.

## 5 RESULTATS I USER TESTING

En aquest apartat es mostraran els resultats obtinguts en el desenvolupament de l'aplicació i l'experiència dels primers usuaris externs.

### 5.1 RESULTATS

A continuació es mostren els resultats d'alguns dels components desenvolupats: el formulari de registre i el formulari de login, el component dels meus equips, els components que conformen la vista d'una competició els quals són la classificació, els resultats i les estadístiques, el resum i el xat d'un partit i finalment els jugadors d'un equip.

#### 5.1.1 FORMULARI DE REGISTRE I LOGIN

Figura 4 - Formulari de registre (pas 1)

El formulari de registre consta d'un formulari de 3 passos (Figura 4), on s'introdueixen diferents dades personals que seran emmagatzemades a la base de dades per a millorar l'experiència de l'usuari.

El primer i segon pas són dades generals de l'usuari, en canvi, el tercer pas serà diferent en funció del rol triat en el pas previ, en cas d'haver triat el rol de jugador, hauràs de triar la teva altura i el teu pes, en canvi, si has triat el rol d'organitzador, hauràs d'introduir algunes dades de la teva organització.

Figura 5 - Formulari de login

El formulari de login (Figura 5) permet autenticar un usuari dins de l'aplicació, permetent així carregar les dades emmagatzemades a la base de dades corresponents a l'usuari connectat.

#### 5.1.2 ELS MEUS EQUIPS

Figura 6 - Els meus equips

L'usuari amb rol jugador, té un apartat anomenat "Els meus equips" (Figura 6) on pot veure els diferents equips als quals forma part. Per a cada equip es mostra l'escut, el nom i el rol del jugador dins de l'equip, que pot ser (C) capità o (J) jugador. Si l'usuari selecciona un equip, l'aplicació el porta a veure els detalls de l'equip i si prem el botó blau al costat del títol es carrega el formulari per a crear un equip.

#### 5.1.3 CLASSIFICACIÓ, RESULTATS I ESTADÍSTIQUES

Equip	PT	PJ	PG	PP	PE	GF	GC	DG
1 Parets City	6	2	2	0	0	2	0	2
2 United States	3	1	1	0	0	3	1	2
3 Eagles	0	2	0	2	0	0	2	-2
4 Borussia Dortmund	0	1	0	1	0	1	3	-2

Figura 7 - Classificació d'una lliga

El component classificació està format pels diferents equips que participen en la lliga i es mostren els punts que té cada equip (Figura 7), els partits jugats, guanyats, perduts i diferents estadístiques referents als gols, sets o punts depenent de l'esport.

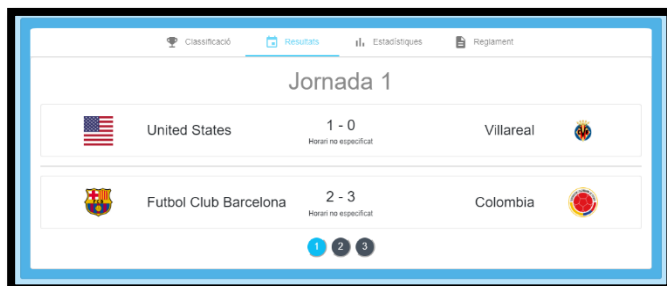


Figura 8 - Resultats

El component resultats (Figura 8) mostra els diferents partits en items, on cada item és un partit, on l'equip local es mostra a l'esquerra i l'equip visitant a la dreta. A part també es pot veure el resultat i l'horari del partit. Si el que es vol és navegar entre diferents jornades es pot fer amb els botons 1, 2 i 3 que es troben a la part inferior.

Id	Imatge	Nom	Equip	Gols	Assistències	Cartes grogues	Cartes vermelles
1		Erling Haaland	Colòmbia	0	0	3	
2		Alexia Putellas	Futbol Club Barcelona	2	1	1	
3		Sergino Dest	Futbol Club Barcelona	0	0	1	

Figura 9 - Estadístiques

Les estadístiques (Figura 9) únicament es mostren si la competició és Futbol Sala, Futbol 7 o Futbol 11 mostrant les targetes grogues, les targetes vermelles i els gols de cada jugador.

#### 5.1.4 RESUM I XAT D'UN PARTIT

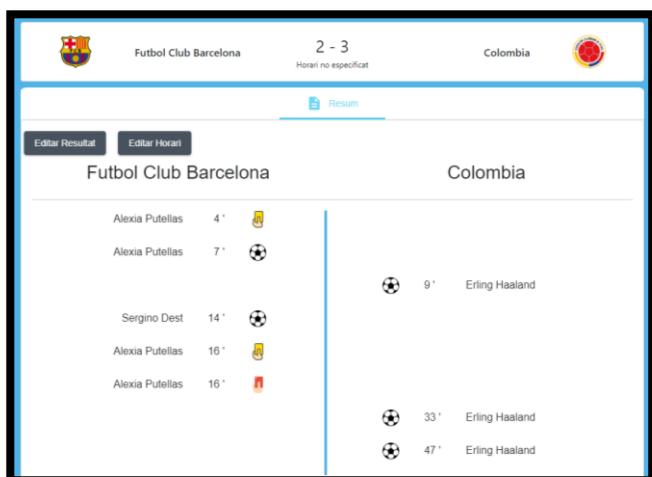


Figura 10 - Resum d'un partit

El resum d'un partit (Figura 10) conté tots els events que succeeixen durant un partit. Cada event està definit per qui l'ha realitzat, quina acció s'ha realitzat i a quin minut s'ha realitzat. Aquests events estan ordenats cronològicament en forma de timeline.

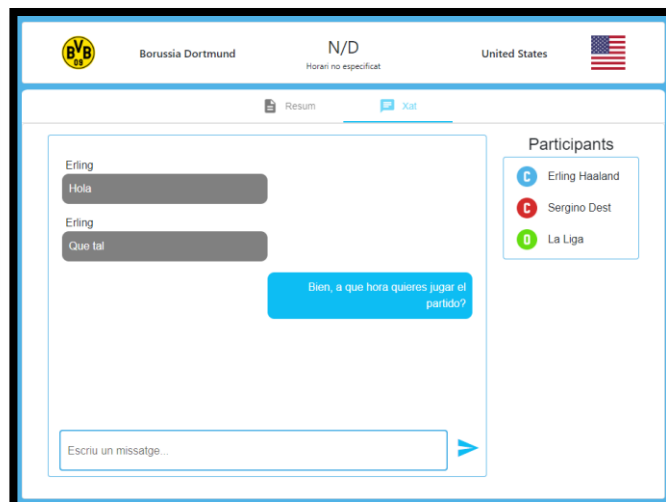


Figura 11 - Xat d'un partit

El xat (Figura 11) creiem que és una eina molt útil, ja que facilita el contacte amb l'equip rival per a poder definir un horari i un lloc de partit. En el xat només poden escriure els capitans dels equips o l'organitzador de la lliga. Els missatges que envia l'usuari connectat surten de color blau, mentre que els missatges que envien els altres participants són de color gris i amb el nom de qui l'ha enviat.

#### 5.1.5 JUGADORS D'UN EQUIP

Imatge	Dorsal	Nom	Cognoms	Rol
	2	Sergino	Dest	Jugador
	11	Alexia	Putellas	Capità

Figura 12 - Jugadors d'un equip

El component llistar-jugadors mostra tots els jugadors que formen part d'un equip. En aquest cas podem veure que el Futbol Club Barcelona està format per dos jugadors que tenen el dorsal 2 i 11 i es diuen Sergiño Dest i Alexia Putellas.

## 5.2 USER TESTING

El principal objectiu de l'usuari testing o proves d'usuari és assegurar que l'aplicació web funcioni com l'usuari espera i, per tant, poder garantir un cert grau de qualitat del software desenvolupat.

Per a realitzar l'usuari testing, s'han preparat 5 processos que l'usuari haurà de dur a terme:

1. Registrar-se com a usuari, crear un equip i convidar a un jugador a l'equip.
2. Registrar-se com a organitzador, crear una competició i acceptar sol·licituds d'inscripció.
3. Connectar-se amb un usuari que tingui rol d'organitzador, donar el començament d'una competició i introduir la data i els resultats de tots els partits de les dues primeres jornades.
4. Visualitzar les estadístiques de la competició.
5. Enviar i rebre diferents missatges pel xat d'un partit.

Per a realitzar aquests processos s'han seleccionat 5 persones amb diferents rangs d'edat, sexe i aficions.

Un cop els usuaris han finalitzat els tres processos, se'ls hi ha enviat un enllaç amb un formulari per a que responguin 5 preguntes sobre els diferents processos realitzats:

1. Puntua de l'1 al 5 si l'aplicació web t'ha semblat intuïtiva, es a dir si t'ha resultat fàcil realitzar els processos plantejats, entenenent 1 com la puntuació més baixa i 5 com la puntuació més alta.
2. Puntua de l'1 al 5 el disseny de la pàgina web, entenenent 1 com la puntuació més baixa i 5 com la puntuació més alta.
3. Puntua de l'1 al 5 si el contingut de la pàgina web és l'adequat per a una aplicació de creació i gestió de tornejos, entenenent 1 com la puntuació més baixa i 5 com la puntuació més alta.
4. T'has trobat algun error ja sigui funcional o de disseny dins de l'aplicació web? Descriu-lo.
5. Podries proposar alguna millora per a l'aplicació web? En cas afirmatiu digues quina.

Per a analitzar millor les dades, s'ha realitzat un gràfic radar. En aquest gràfic (figura 13), podem veure la mitjana aritmètica de les tres primeres preguntes on s'havia d'avaluar de l'1 al 5 certes qualitats de l'aplicació, les quals són: el contingut, el disseny i si l'aplicació és intuïtiva o no.

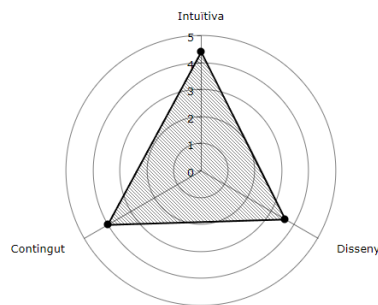


Figura 13 - Gràfic radar sobre les respostes del formulari

Com es pot observar en la Figura 13 el que més ha agradat de l'aplicació, amb una valoració d'un 4,4 sobre 5 és que l'aplicació és molt intuïtiva. En canvi, el que menys ha agradat ha estat el disseny d'aquesta amb una valoració d'un 3,8 sobre 5.

L'únic error que els usuaris han trobat dins de l'aplicació ha estat relacionat amb el component *upload-image*. Aquest component s'encarrega de pujar una imatge a Firebase Storage, que és el núvol on s'emmagatzemen totes les imatges relacionades amb l'aplicació web. Es tracta d'un error en la reusabilitat d'aquest component, quan l'usuari pujava una imatge en el formulari de creació d'usuari, posteriorment es mostrava aquesta mateixa imatge en el formulari de crear equip.

## 6 CONCLUSIONS I MILLORES

- S'ha dissenyat la interfície de l'aplicació web utilitzant l'eina MarvelApp.
- S'ha dissenyat i implementat una base de dades utilitzant el mòdul Firestore de la plataforma Firebase.
- S'ha utilitzat una nova tecnologia anomenada Angular en el desenvolupament de l'aplicació web.
- S'han aplicat diferents coneixements i procediments adquirits durant el grau universitari, els quals són: la metodologia de desenvolupament, la captació de requisits, el disseny de casos d'ús, el disseny de l'aplicació, el desenvolupament i el test del codi.
- Aquest projecte m'ha servit per a aprendre noves tecnologies relacionades amb el món de les aplicacions web. El fet d'haver de desenvolupar un software des de 0 m'ha permès aplicar tots els conceptes adquirits durant la carrera i gràcies al fet que aquests procediments s'han aplicat correctament el projecte s'ha pogut complementar satisfactòriament.

Amb el treball realitzat i el feedback rebut gràcies al usuari testing, hem trobat certes millores per a l'aplicació:

- Millorar el disseny d'algunes interfícies per tal de donar-li un estil més modern.

- Realitzar una vista per a visualitzar tots els emparellaments dels tornejos en comptes de visualitzar únicament els partits de cada ronda.
  - Afegir un sistema d'inscripcions amb pagament per a les competicions on per inscriure't s'ha de pagar una quota.
  - Afegir un algoritme de Machine learning que permeti predir el guanyador d'un partit a partir de les estadístiques emmagatzemades, per així donar-li un aspecte més competitiu.
- [13] <https://marvelapp.com/prototype/60g0856/> projecte de MarvelApp que conté totes les pantalles dissenyades YourLeagues (data últim accés: Maig 2022).
- [14] <https://github.com/arnaucruz1997/YourLeagues> repositori web que conté el codi font del projecte YourLeagues (data últim accés: juny 2022).

## AGRAÏMENTS

Agraeixo al meu tutor del TFG Enric Martí Gòdia per la seva atenció i ajuda durant totes les fases del desenvolupament del projecte.

També agraeixo a la meva mare Angels, al meu pare Joan i als meus dos germans Ivan i Alex els quals m'han donat suport durant tots aquests anys d'universitari.

## BIBLIOGRAFIA

- [1] <https://doleague.com/> web oficial de DoLeague, aplicació web per a gestionar competicions (data últim accés: febrer 2022).
- [2] <https://gestorligas.com/> web oficial de GestorLigas, aplicació web per a gestionar competicions (data últim accés: febrer 2022).
- [3] <https://xporty.com/> web oficial d'Xporty, aplicació web per a gestionar competicions (data últim accés: febrer 2022).
- [4] <https://angular.io/> web oficial del framework Angular, amb documentació i recursos (data últim accés: febrer 2022).
- [5] <https://firebase.google.com/> web oficial de Firebase, amb documentació, productes i recursos (data últim accés: febrer 2022).
- [6] <https://code.visualstudio.com/> web oficial del entorn de desenvolupament Visual Studio Code, amb documentació, extensions, manuals d'ús i enllaç de descarrega (data últim accés: febrer 2022).
- [7] <https://desktop.github.com/> web oficial de GitHub Desktop, amb manual d'ús i enllaç de descarrega (data últim accés: febrer 2022).
- [8] <https://trello.com/es/> web oficial de Trello, amb documentació i accés al portal (data últim accés: febrer 2022).
- [9] <https://marvelapp.com/> web oficial de Marvel app, pàgina web per a realitzar prototips (data últim accés: febrer 2022).
- [10] <https://www.workfront.com/project-management/methodologies/waterfall> web on explica la metodologia Waterfall (data últim accés: març 2022).
- [11] <https://firebase.google.com/docs/firestore/quotas> web on explica les característiques de la quota gratuïta (data últim accés: maig 2022).
- [12] <https://www.dataversity.net/how-to-design-schema-for-your-nosql-database/> pàgina web on explica com dissenyar una base de dades No-Sql (data últim accés: febrer 2022).

## APÈNDIX

### A.1 REQUISITS DE DADES:

Un usuari ha de tenir la següent informació: nom, cognoms, email, contrasenya, DNI, sexe, data de naixement, lloc de residència, la imatge de perfil i el rol d'usuari (organitzador o jugador).

Un usuari amb rol d'organitzador ademés ha de guardar la següent informació: nom, descripció, correu i telèfon de l'organització i competicions que organitza.

El rol de jugador també ha de guardar l'alçada i pes del jugador.

Un equip ha de tenir la següent informació: jugadors que formen l'equip, nom i l'escut de l'equip, abreviació del nom de l'equip (paraula de 3 lletres) i un capità que serà el creador de l'equip.

Un jugador ha de poder estar en un o més equips alhora i també ha de poder crear més d'un equip.

Un organitzador ha de poder crear diverses competicions de qualsevol esport. Una competició ha de guardar la següent informació: l'organitzador que crea la competició, número d'equips que hi podran participar, equips inscrits, privacitat de la competició (pública o privada), tipus d'esport que podrà ser: Futbol 7, Futbol Sala, Futbol 11, Bàsquet, Pàdel, Tennis, Handbol o Voleibol, informació general de la lliga, normativa de la lliga, tipus de competició que podrà ser un torneig o una lliga, en cas de que sigui una lliga, s'haurà d'indicar si la lliga serà a volta única o a doble volta, i també si tindrà o no una fase de play-off.

Una característica pròpia de les lligues és la classificació on s'hauran de guardar les següents dades: nº de partits jugats, nº de partits guanyats, nº de partits perduts i puntuació. Si la competició és una competició que la seva puntuació es calcula amb sets es a dir Voleibol, tennis, pàdel o Handbol la classificació també tindrà com a dades els sets guanyats i els sets perduts, en canvi si l'aplicació és únicament de puntuació i no té sets es a dir Futbol 7, Futbol Sala, Futbol 11, Bàsquet la classificació també tindrà com a dades els partits empatats, els gols o cistelles a favor i en contra i la diferència entre a favor i en contra.

Una vegada la competició comença es generen tots els partits que cal jugar, per tant una competició tindrà un nombre determinat de partits.

Per a cada partit es guarda la següent informació: competició a la que forma part, equip local, equip visitant, lloc on es jugarà el partit, horari de partit, equip guanyador, equip perdedor i resultat o puntuació que serà diferent en funció de l'esport.

- **Futbol 7, futbol Sala, futbol 11, bàsquet, handbol:** la puntuació quedarà dividida en 2 números enters positius, on cada número indica la puntuació de cada equip i guanya l'equip que faci més punts/gols.

- **Voleibol:** la puntuació queda dividida en 5 sets on cada set tindrà 2 números enters positius que indicaran la puntuació de cada equip en cada set, un set el guanya l'equip que arriba a 25 punts amb diferència de 2 abans que l'equip contrari excepte quan és el 5é set que el guanya l'equip que arriba a 15 punts amb diferència de 2 abans que l'equip rival, i un partit el guanya l'equip que guanya 3 sets.

- **Tennis, pàdel:** la puntuació queda dividida en 3 sets on cada set tindrà 2 números enters positius, un set el guanya l'equip que arriba als 6 punts amb diferència de 2, als 7 punts amb diferència de 2 o en el tie-break en cas de que el resultat en el set hagi estat 6 a 6.

Apart cada partit ha de tenir un xat el qual contindrà la següent informació: els capitans dels dos equips que participaran en el partit, l'organitzador de la competició i els missatges. Un xat podrà tenir n missatges i cada missatge tindrà com a informació a quin xat s'ha enviat el missatge, l'usuari que envia el missatge i l'hora en la que s'ha enviat el missatge.

Cada partit ha de tenir un resum on per a cada esport serà diferent:

- **Futbol 7, futbol Sala, futbol 11, handbol:** una timeline, on s'indicaran qui i quan ha marcat un gol o ha rebut una targeta groga o vermella.

- **Bàsquet, voleibol, tennis i pàdel:** Una taula on es mostrarà el resultat de cada part i el resultat final.

També cada competició tindrà un apartat d'estadístiques on apareixerà la següent informació dels jugadors en la competició corresponent:

- **Futbol 7, Futbol Sala, Futbol 11, Handbol:** nom i cognoms, imatge de perfil, partits jugats, gols, promig de gols per partit, equip al que pertany, targetes grogues i targetes vermelles.

- **Bàsquet, voleibol, tennis i pàdel:** no tindran apartat d'estadístiques.

## A.2 REQUISITS FUNCIONALS:

Per a descriure els requisits funcionals que incorporarà la nostra aplicació web, els dividirem en 6 grups.

### 1. Requisits funcionals d'un usuari no registrat.

ID	Descripció	Prioritat
RF-01	L'usuari ha de poder registrar-se a l'aplicació web.	A
RF-02	L'usuari ha de poder iniciar sessió a l'aplicació web.	A
RF-03	L'usuari ha de poder visualitzar totes les competicions públiques.	B
RF-04	L'usuari ha de poder visualitzar l'apartat de "Contacte".	C
RF-05	L'usuari ha de poder visualitzar l'apartat de "Qui som?".	C

### 2. Requisits funcionals d'un usuari registrat.

ID	Descripció	Prioritat
RF-06	L'usuari ha de poder visualitzar totes les competicions públiques.	A
RF-07	L'usuari ha de poder visualitzar la classificació o <u>bracket</u> de les competicions públiques.	A
RF-08	L'usuari ha de poder visualitzar els resultats de la competició.	A
RF-09	L'usuari ha de poder veure els detalls dels equips.	B
RF-10	L'usuari ha de poder veure la informació general de la competició.	B
RF-11	L'usuari ha de poder visualitzar l'apartat de "Contacte".	B
RF-12	L'usuari ha de poder visualitzar l'apartat de "Qui som?".	B
RF-13	L'usuari ha de poder realitzar una cerca de la competició filtrant per esport i per estat de la competició.	B
RF-14	L'usuari ha de poder veure els detalls de cada partit.	B

### 3. Requisits funcionals d'un usuari amb rol jugador.

ID	Descripció	Prioritat
RF-15	El jugador ha de poder modificar el seu perfil.	A
RF-16	El jugador (capità) ha de poder invitar a altres usuaris al seu equip.	A
RF-17	El jugador (capità) ha de poder enviar sol·licitud d'inscripció a diferents competicions.	A
RF-18	El jugador ha de poder visualitzar i escriure en el xat de cada partit per tal de concretar l'horari d'aquest.	A
RF-19	El jugador ha de poder crear un equip.	A
RF-20	El jugador ha de poder visualitzar l'apartat dels "meus equips".	B
RF-21	El jugador ha de poder visualitzar l'apartat de "partits vinents".	B
RF-22	El jugador ha de poder visualitzar l'apartat de les "meves competicions".	B
RF-23	El jugador (capità) ha de poder modificar informació general de l'equip.	B
RF-24	El jugador (capità) ha de poder expulsar a altres usuaris de l'equip.	B
RF-25	El jugador ha de poder enviar i confirmar els resultats de cada partit.	B
RF-26	El jugador ha de poder enviar sol·licitud per a unir-se a un equip	B
RF-27	El jugador (capità) ha de poder acceptar usuaris a l'equip.	B

### 4. Requisits funcionals d'un usuari amb rol organitzador.

ID	Descripció	Prioritat
RF-28	L'organitzador ha de poder crear una competició.	A
RF-29	L'organitzador ha de poder acceptar les inscripcions dels equips.	A
RF-30	L'organitzador ha de poder introduir els resultats dels partits finalitzats.	A
RF-31	L'organitzador ha de poder introduir l'horari de cada partit.	A
RF-32	L'organitzador ha de poder donar començament a la competició.	A
RF-33	L'organitzador ha de poder finalitzar la competició, una vegada hagin acabat tots els partits.	B
RF-34	L'organitzador ha de poder visualitzar i escriure en el xat de cada partit per tal de concretar l'horari d'aquest.	B

### 5. Requisits funcionals del sistema.

ID	Descripció	Prioritat
RF-35	El sistema ha de generar automàticament el calendari de la competició.	A
RF-36	El sistema ha d'actualitzar automàticament la classificació de la lliga en funció dels resultats introduïts.	A
RF-37	El sistema ha d'actualitzar automàticament els brackets del torneig en funció dels resultats introduïts.	A
RF-38	El sistema ha d'actualitzar les estadístiques en funció dels resultats introduïts.	A
RF-39	El sistema ha d'enviar missatges amb els recordatoris i resultats als jugadors de forma automàtica.	B

### 6. Requisits funcionals d'un usuari amb rol administrador.

ID	Descripció	Prioritat
RF-40	L'administrador ha de poder eliminar comptes d'usuari.	A
RF-41	L'administrador ha de poder eliminar competicions.	A
RF-42	L'administrador ha de poder eliminar equips.	A
RF-43	L'administrador ha de poder canviar resultats.	A
RF-44	L'administrador ha de poder tancar competicions.	B