

# Desenvolupament web d'una aplicació per a la gestió d'una lliga virtual de bàsquet: “*Liga Virtual de Baloncesto*”

**Oriol Aldea Elvira**

Desenvolupament web

Àrea d'Informàtica, Multimèdia y Telecomunicació

**Consultor: Pablo Pineda Ruipérez**

**Professor responsable: Santi Caballé Llobet**

Gener 2021



Aquesta obra està subjecta a la llicència Creative Commons.  
**Reconeixement-NoComercial-SenseObraDerivada**

## **Agraïments**

Al Pau, consultor del treball per les seves aportacions i guiatge durant l'elaboració d'aquesta memòria

Al Marc, per proposar-me aquesta idea de projecte i ajudar-me amb els aspectes gràfics.

I sobretot, a la meva parella Jenn, per tot el seu suport, ajuda i ànims en els moments que més ho necessitava.

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Desenvolupament web d'una aplicació per a la gestió d'una lliga virtual de bàsquet: "Liga Virtual de Baloncesto"</i>
<b>Nom de l'autor:</b>	<i>Oriol Aldea Elvira</i>
<b>Nom del consultor/a:</b>	<i>Pablo Pineda Ruipérez</i>
<b>Nom del PRA:</b>	<i>Santi Caballé Llobet</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>01/2021</i>
<b>Titulació o programa:</b>	<i>Desenvolupament web</i>
<b>Àrea del Treball Final:</b>	<i>Àrea d'Informàtica, Multimèdia y Telecomunicació</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Laravel, BBDD, bàsquetbol</i>
<p><b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>L'objectiu d'aquest projecte és el de desenvolupar una aplicació web que permeti la creació, gestió i ús d'una lliga virtual de bàsquet.</p> <p>Aquesta aplicació disposa de dues parts. La primera és l'accés dels usuaris, des d'on poden crear un equip, comprar i vendre jugadors a partir d'un pressupost i consultar la classificació general.</p> <p>La segona part és l'accés de l'administrador que li permet gestionar tot l'entorn de la competició, incorporar nous jugadors, actualitzar les dades estadístiques i generar les classificacions.</p> <p>La puntuació dels equips dels usuaris es genera a partir de l'actuació real dels jugadors donats d'alta a l'aplicació en els seus partits de competició oficial. A més, durant l'evolució de la temporada el preu dels jugadors fluctua en funció d'aquesta actuació.</p> <p>L'aplicació està desenvolupada principalment amb la llibreria Laravel (per al back-end dels usuaris i de l'administrador). A més a més, s'ha fet servir la llibreria Bootstrap 4 (basat en HTML, CSS i Javascript) per perfilar i ajustar l'estil de tota l'aplicació.</p>	

**Abstract (in English, 250 words or less):**

The goal of this project is to develop a web application that allows the creation, management and use of a virtual basketball league.

This application has two parts. The first is user access, from where they can create a team, buy and sell players from a budget and consult the general classification.

The second part is Administrator access that allows you to manage the whole competition environment, incorporate new players, update statistical data and generate rankings.

User teams score is generated from the actual performance rating of the given players registered to the app in their official competition matches. In addition, during the season the price of players fluctuates according to this performance.

The app is primarily developed with the Laravel library (for user back-end and administrator back-end). In addition, Bootstrap 4 library (based on HTML, CSS, and Javascript) has been used to profile and adjust the design of all the application.

# Índex

1. Introducció.....	1
<b>1.1 Context i justificació del Treball.....</b>	<b>1</b>
<b>1.4 Planificació del Treball.....</b>	<b>3</b>
<b>1.5 Breu sumari de productes obtinguts.....</b>	<b>6</b>
<b>1.6 Breu descripció dels altres capítols de la memòria.....</b>	<b>7</b>
2. Anàlisi.....	8
<b>2.1 Anàlisi del mercat.....</b>	<b>8</b>
<b>2.2 Públic objectiu i perfils d'usuari.....</b>	<b>9</b>
<b>2.3 Identificació dels riscos i pla de contingència.....</b>	<b>9</b>
<b>2.4 LVB: cas Bàsquet Maresme Boet Mataró 3Viles.....</b>	<b>10</b>
3. Requisits del projecte.....	11
<b>3.1 Requisits funcionals.....</b>	<b>11</b>
<b>3.2 Requisits tecnològics.....</b>	<b>12</b>
<b>3.3 Casos d'ús.....</b>	<b>15</b>
<b>3.4 Diagrama de la base de dades.....</b>	<b>33</b>
<b>3.5 Diagrama de classes.....</b>	<b>34</b>
<b>3.6 Diagrama de navegació.....</b>	<b>35</b>
4. Disseny i implementació.....	36
<b>4.1 Principis generals.....</b>	<b>36</b>
<b>4.2 Estructura de les pàgines.....</b>	<b>38</b>
<b>4.3 Estil general de l'aplicació.....</b>	<b>39</b>
<b>4.4 Prototips.....</b>	<b>41</b>
<b>4.5 Creació dels models.....</b>	<b>45</b>
<b>4.6 Creació dels controladors.....</b>	<b>46</b>
<b>4.7 Creació de les vistes.....</b>	<b>46</b>
<b>4.8 Creació de les rutes.....</b>	<b>48</b>
<b>4.9 Importació de dades estadístiques.....</b>	<b>49</b>
<b>4.10 Càlcul del preu dels jugadors.....</b>	<b>50</b>
<b>4.11 Càlcul de la puntuació i valor dels equips.....</b>	<b>51</b>
<b>4.12 Generació de les classificacions.....</b>	<b>52</b>
5. Instal·lació.....	53
<b>5.1 Consideracions inicials.....</b>	<b>54</b>
<b>5.2 Requisits d'instal·lació.....</b>	<b>54</b>
<b>5.3 Creació i configuració del projecte.....</b>	<b>55</b>
<b>5.4 Creació i configuració de la base de dades.....</b>	<b>56</b>
<b>5.5 Instal·lació i configuració dels paquets complementaris.....</b>	<b>56</b>
<b>5.6 Relació amb la base de dades.....</b>	<b>58</b>
<b>5.7 Publicació al servidor web públic.....</b>	<b>60</b>
6. Resultat final.....	61
7. Conclusions.....	65
<b>7.1 Conclusions del treball.....</b>	<b>65</b>
<b>7.2 Treball futur.....</b>	<b>66</b>
8. Glossari.....	69
9. Bibliografia.....	72
10. Annexos.....	73
<b>A. Resum de les principals competicions de bàsquet.....</b>	<b>73</b>
<b>B. Lliurables del projecte.....</b>	<b>75</b>

## Llista de figures

Il·lustració 1: Planificació del projecte .....	4
Il·lustració 2: taulell del projecte .....	6
Il·lustració 3: Diagrama casos d'ús Usuari .....	16
Il·lustració 4: Diagrama casos d'ús Administrador.....	24
Il·lustració 5: Diagrama casos d'ús Sistema.....	29
Il·lustració 6: Diagrama de la base de dades .....	33
Il·lustració 7: Diagrama de classes.....	34
Il·lustració 8: Diagrama de navegació dels usuaris .....	35
Il·lustració 9: Diagrama de navegació de l'administrador .....	35
Il·lustració 10: Detall d'un equip d'usuari .....	37
Il·lustració 11: Detall d'un jugador .....	37
Il·lustració 12: Estructura de les pàgines dels usuaris.....	38
Il·lustració 13: Estructura de les pàgines de l'administrador.....	39
Il·lustració 14: Gamma de colors.....	40
Il·lustració 15: Resum tipografies emprades .....	41
Il·lustració 16: Prototips inici sessió i pantalla inicial .....	42
Il·lustració 17: Prototip de la plantilla de l'equip.....	42
Il·lustració 18: Prototip del mercat de jugadors .....	43
Il·lustració 19: Prototip classificació general i de la jornada .....	43
Il·lustració 20: Pantalla de login i pantalla inicial .....	43
Il·lustració 21: Prototips de les pantalles CRUD.....	44
Il·lustració 22: Prototip importació estadístiques i gestió del perfil .....	44
Il·lustració 23: Exemple del model Lliga .....	45
Il·lustració 24: Exemple del controlador TeamController.....	46
Il·lustració 25: Exemple d'arxiu blade de crear un equip des de l'administrador .....	47
Il·lustració 26: Exemple de l'arxiu routes.php de l'aplicació.....	48
Il·lustració 27: Exemple de l'arxi routes del projecte .....	48
Il·lustració 28: Exemple del fitxer amb dades per importar.....	49
Il·lustració 29: Procediment per a obtenir el preu del jugador .....	51
Il·lustració 30: Càlcul de la puntuació de l'equip en una jornada.....	52
Il·lustració 31: Procediment per obtenir la classificació d'una jornada (passada com a paràmetre) .....	53
Il·lustració 32: Procediment per desar les dades de la classificació de la jornada a la base de dades .....	53
Il·lustració 33: Fitxer vhost.....	56
Il·lustració 34: Configuració de la base de dades.....	56
Il·lustració 35: Exemple de migració de la taula "teams" .....	59
Il·lustració 36: Exemple d'arxiu Seeder per a la creació de rols i usuaris.....	59
Il·lustració 37: Pantalla d'INICIO abans i després de crear un equip .....	61
Il·lustració 38: Pantalla d'EQUIPS abans i després de crear un equip.....	61
Il·lustració 39: Pantalla EQUIPS. Detall de l'equip abans i després de comprar jugadors. A la tercera imatge s'ha seleccionat MVP .....	62
Il·lustració 40: pantalles MERCADO i CLASIFICACIÓN .....	62
Il·lustració 41: Dashboard en la vista d'usuari administrador .....	63
Il·lustració 42: Pantalla IMPORTACIÓ D'ESTADÍSTIQUES en la vista d'usuari administrador.....	63

Il·lustració 43: Pantalla EQUIPOS en la vista d'usuari administrador, amb el detall de VER EQUIPO i EDITAR EQUIPO.....	63
Il·lustració 44: Pantalla EQUIPOS/CREAR NUEVO EQUIPO en la vista d'usuari administrador.....	63
Il·lustració 45: Pantalla CLASIFICACIÓN de la vista d'usuari administrador sense i amb estadístiques importades .....	64
Il·lustració 46: Pantalla JUGADORES i detall de jugador en la vista d'usuari administrador.....	64
Il·lustració 47: Pantalla EQUIPOS OFICIALES i detall d'un equip en la vista d'usuari administrador .....	64
Il·lustració 49: Resum de les principals competicions de bàsquet a l'Estat .....	74



# 1. Introducció

## 1.1 Context i justificació del Treball

“Oriol, pots crear alguna cosa semblant al ‘Supermanager’ per als primers equips del club?” Aquesta va ser la pregunta que el responsable de comunicació del club de bàsquet en el qual sóc entrenador em va formular fa uns mesos i que ha esdevingut l’origen d’aquest projecte.

En aquesta conversa informal, estàvem buscant la forma d’oferir activitats diferents i innovadores a la pràctica esportiva per a tots els integrants d’aquest club per tal d’aconseguir una major alienació i donar més valor afegir al club.

El “Supermanager” al qual es referia es tracta del **Supermanager ACB**[1], originàriament una aplicació web i actualment una *app* per a dispositius mòbils, desenvolupada per la **Lliga ACB de bàsquet**. L’objectiu d’aquesta aplicació és que els usuaris registrats puguin convertir-se en gestors d’un equip professional.

El funcionament és senzill: Qualsevol usuari registrat pot crear un equip (o diversos) i amb un pressupost tancat comprar els jugadors de la Lliga ACB (fins a un màxim de 12, que és el nombre màxim de jugadors que es poden inscriure per disputar un partit) per tal de disposar de l’equip que doni el màxim de punts possibles. Aquesta puntuació s’obté de la valoració real que fan els jugadors durant les diferents jornades de la competició. A més a més el preu dels jugadors també varia segons la seva actuació. Els usuaris registrats competeixen entre ells per tal d’acabar al final de la temporada en la primera posició i obtenir premis.

Donant-li voltes a la pregunta vaig arribar a la conclusió que, és cert que existeixen aquests tipus de “jocs” de gestió d’equips, però són exclusius de competicions professionals (alguns exemples són: *Supermanager ACB* per a la Lliga ACB, el *Fantasy Challenge*[2] per a la Euroleague Basketball o la *Liga Fantasy*[3] de Marca per a la LaLiga Santander de futbol) i no hi ha cap eina o cap aplicació en la qual un club amateur o semiprofessional (en aquest cas de bàsquet) pugui fer quelcom semblant al que fan aquestes aplicacions.

Aquí és on apareix la “**Liga Virtual de Baloncesto**” (nom de l’aplicació), una aplicació web que permeti als clubs de bàsquet amateurs (o qui sap si potser alguna federació) crear les seves pròpies competicions virtuals amb els jugadors de la mateixa entitat a partir de les dades estadístiques oficials.

---

1 <https://supermanager.acb.com>

2 <https://fantasychallenge.euroleague.net/>

3 <https://www.laligafantasymarca.com/>

## 1.2 Objectius del Treball

### OBJECTIU PRINCIPAL

- Implementar una aplicació web que permeti la gestió d'una lliga virtual de bàsquet amb els jugadors d'un club. Aquesta aplicació va dirigida, principalment, als clubs que participen en competicions organitzades per la Federació Española de Baloncesto (FEB).

### OBJECTIUS SECUNDARIS

- Dotar a aquestes entitats d'una eina de valor afegit per als seus membres.
- Augmentar la integració per part dels membres d'un club de bàsquet amb l'entitat.
- Assegurar que la informació continguda en la mateixa compleix els estàndards de seguretat i privacitat.
- Assegurar que l'aplicació compleix els requisits d'usabilitat.
- Permetre l'autenticació dels usuaris així com el seu registre.
- Permetre als usuaris "gestionar" els integrants d'un equip: compra-venda de jugadors en funció de les seves actuacions reals.
- Disposar d'una base de dades dels jugadors del club amb la seva informació estadística més rellevant.
- Implementar un sistema de recollida automàtica de les dades estadístiques de les competicions EBA (Lliga Espanyola de Bàsquet Aficionat) i LF2 (Lliga Femenina-2) de la Federació Espanyola de Bàsquet.
- Establir un algoritme per poder generar de manera automàtica el valor dels jugadors.
- Controlar els accessos a l'aplicació en funció del rol de l'usuari (usuari o administrador).
- Permetre a l'administrador poder afegir, editar, eliminar i visualitzar tots aquells elements necessari per al correcte funcionament de l'aplicació (usuaris, equips, competicions, jugadors...).
- Oferir a l'usuari administrador la possibilitat de visualitzar en temps real informació sobre el sistema, com ara: nombre d'equips, nombre d'usuaris, jugador/es més-menys comprats i venuts, propera jornada...

### **1.3 Enfocament i mètode seguit**

Tal com s'indica en la justificació del treball, no existeix cap plataforma o eina de codi obert que permeti la creació i gestió d'una lliga virtual per a clubs amateurs i/o semiprofessionals. A més a més, les aplicacions existents (Supermanager, Fantasy Challenge...) pertanyen a empreses privades, estan adaptades a les especificacions de les lligues i compten amb els recursos necessaris (dades estadístiques, dades dels jugadors, llicències de distribució...).

Per tant, la solució a aquesta necessitat és la de desenvolupar-la de nou, d'aquesta forma es pot adaptar el principi de funcionament d'aquestes *apps* a una eina a l'abast de tothom que compleixi els objectius descrits en l'apartat anterior.

### **1.4 Planificació del Treball**

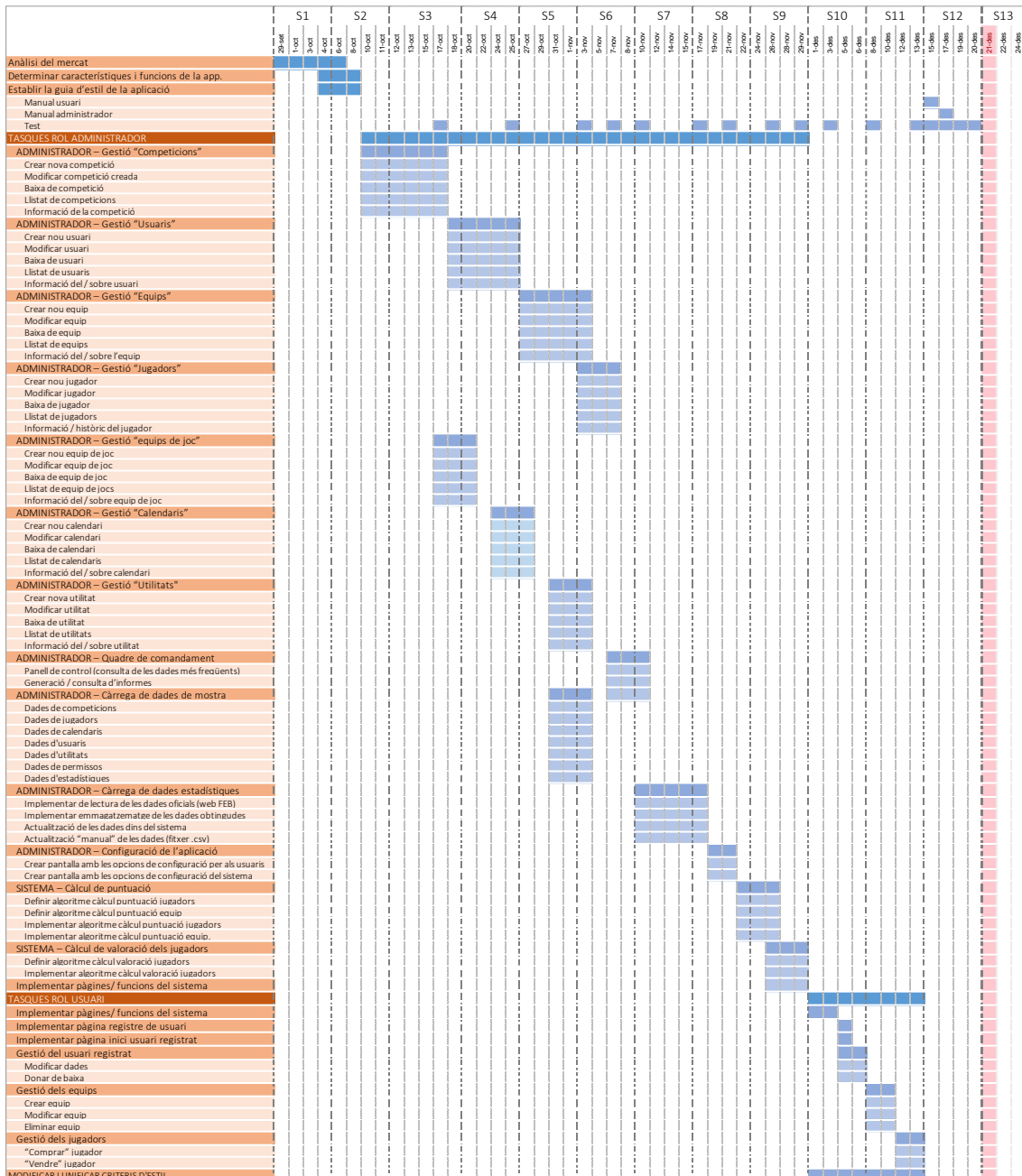
La planificació del treball s'ha realitzat tenint en compte la temporalització de les diferents entregues de les PACs de seguiment, la quantitat de temps disponible per l'autor i d'acord amb la metodologia obtinguda.

Pel que fa al temps material disponible per l'autor, crec que també és un factor a tenir en compte, ja que a causa dels meus compromisos (feines, altres activitats) dels set dies d'una setmana només li podia dedicar al desenvolupament del projecte la tarda de dos dies i els caps de setmana. Això, en un projecte d'unes 13 setmanes de duració acaben sent moltes hores que no es poden destinar al projecte. És per això que la planificació temporal ja s'ha realitzat tenint en compte només els dies disponibles.

## **PLANIFICACIÓ**

Tal com es menciona en el paràgraf anterior, la planificació està basada en els dies disponibles per dedicar al projecte i en les tasques a desenvolupar. Respecte a les tasques, la planificació s'ha realitzat tenint en compte els 2 grans rols (veure l'apartat ROLS) que interactuen en el sistema i agrupant les tasques en unitats relacionals (aquelles tasques que pertanyen a la mateixa entitat).

A continuació es mostra el diagrama de Gantt amb la planificació.



II-lustració 1: Planificació del projecte

## METODOLOGIA

Pel que fa a la metodologia, actualment existeixen dos grans corrents sobre les quals es basen la gran totalitat de desenvolupament de software: la **"clàssica"** (o cascada) i la **"agile"**. En la primera tot el cicle de desenvolupament es fa de manera seqüencial, una fase darrera de l'altra, de tal manera que una fase no pot començar si l'anterior no s'ha finalitzat. En la segona, el desenvolupament es fa de forma iterativa, on cada iteració és com un petit projecte en si mateix, d'aquesta manera es pot anar fent petites funcionalitats i de mica en mica es veu com l'aplicació va prenent forma.

Ara bé, dins de la metodologia *agile* existeixen diferents metodologies, en aquest cas he decidit fer servir la metodologia **Kanban**[4]. Aquesta metodologia, bàsicament es caracteritza per distribuir totes les tasques (anomenades targetes) dins d'un espai anomenat "taulell" de manera que en un cop d'ull l'usuari pugui identificar totes les que hi ha i saber quina ha de realitzar. A més a més aquestes tasques es poden prioritzar segons la importància, la urgència i l'estat en el qual estiguin, d'aquesta forma permet identificar els colls d'ampolla i optimitzar el temps i resultats.

Per tal d'implementar aquesta metodologia faig servir l'aplicació web **Trello**[5]. Trello, entre altres opcions, permet la gestió de projectes de metodologia *agile*. Es basa en taulells virtuals i dins de cada taulell es creen columnes amb les diferents fites o estats (segons l'usuari). A cada columna s'insereixen les "targetes" que contenen tota la informació relativa a una tasca, així com la data de realització, els usuaris que intervenen, si tenen tasques relacionades...

Per la realització del projecte he triat un taulell compost de 7 columnes:

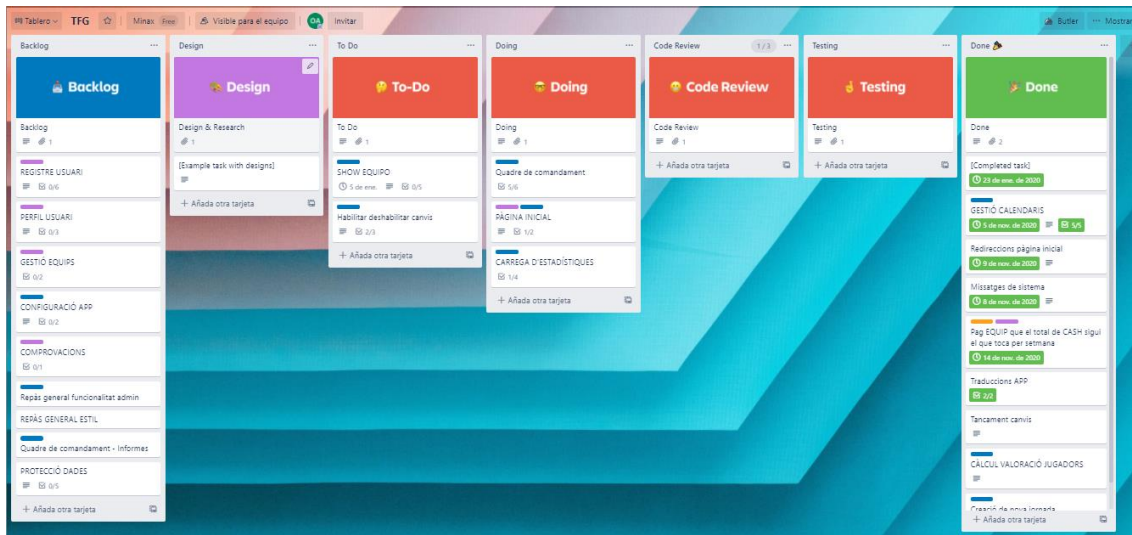
- Backlog: o entrada general de totes les tasques (siguin noves tasques, errors, peticions de clients, millores...).
- Design: on s'agrupen les targetes relatives al disseny.
- TO-DO: On es mostren les tasques a realitzar.
- Doing: On es mostren les tasques que actualment s'estan fent.
- Code review: On es mostren les tasques pendents de revisió per terceres persones.
- Testing: On es mostren les tasques que estan en fase de test.
- Done: On van a parar totes les targetes finalitzades i que han superat les fases anteriors.

A continuació es mostra una imatge del taulell de Trello emprat per al projecte amb algunes de les tasques a realitzar i el seu estat dins de l'aplicació.

---

4 <https://kanbantool.com/es/metodologia-kanban>

5 <https://trello.com/>



**II-lustració 2: taulell del projecte**

## 1.5 Breu sumari de productes obtinguts

Com a resultat final del treball s'obindrà una aplicació web que permetrà:

- A l'usuari administrador: Crear i gestionar una lliga virtual de bàsquet, així com supervisar el seu correcte funcionament i actualitzar les dades.
- Als usuaris del joc: Poder crear els seus equips, fitxar i vendre jugadors i veure la seva posició en la classificació general.

El producte final generat constarà de:

- Els arxius de codi que conformen el projecte (en format .zip).
- L'arxiu amb la base de dades inicial (en format .sql).
- L'enllaç al servidor públic: <https://ligavirtual.maresmeboet3v.com>
- Arxius amb les dades estadístiques dels jugadors fins al 31/12/2020 (per a incorporar les dades i simular les diferents jornades fins a aquesta data).

Les dades d'accés a l'aplicació són:

- Administrador: [aldea86@gmail.com](mailto:aldea86@gmail.com) / password
- Usuari: [test@example.com](mailto:test@example.com) / password

## 1.6 Breu descripció dels altres capítols de la memòria

Els següents apartats d'aquesta memòria estan formats pels següents punts:

2. Anàlisi: S'analitza amb més profunditat la situació actual de les aplicacions similars, les necessitats que es busca cobrir, el públic objectiu i el context particular.
3. Requisits del projecte: Es detallen els diferents requisits identificats a partir dels objectius seleccionats així com quines tecnologies s'empraran per a dur a terme el projecte i quins són els actors que intervenen.
4. Disseny: S'exposa una visió general de quin serà l'aspecte final que es vol que l'aplicació tingui.
5. Implementació: Es detalla tota la part de creació i configuració de l'entorn de treball i base de dades. També es detallen quines són aquelles rutines necessàries per al correcte funcionament general de l'aplicació.
6. Proves / Demostració: S'exposen les proves realitzades així com una demostració del funcionament conjunt del projecte.
7. Conclusions: Es descriuen les conclusions finals a les quals s'ha arribat així com les línies a seguir de treball futur.

## 2. Anàlisi

### 2.1 Anàlisi del mercat

Tal com es descriu en l'apartat primer d'aquesta memòria, “**Liga Virtual de Baloncesto**” (LVB) neix de la idea de voler disposar d'una aplicació on els integrants (joves jugadors en la seva majoria) d'un club de bàsquet poguessin sentir-se com si fossin els entrenadors dels primers equips del club i poguessin comprar i vendre jugadors per tal d'aconseguir la màxima puntuació.

**LVB** agafa la idea d'una aplicació/joc que fa molt de temps que existeix i que és molt coneguda dins dels món del basquetbol, primer com a aplicació web i actualment com a *app* per a dispositius mòbils que és el **Supermanager ACB**. El funcionament bàsic d'aquesta eina ja s'ha descrit en la justificació d'aquesta memòria, però fent un resum ràpid, permet a un usuari crear un equip amb els jugadors de la **Lliga ACB** de bàsquet (actualment anomenada Liga Endesa) a partir d'un pressupost i competir amb altres usuaris per tal d'aconseguir premis.

El potencial que té el Supermanager, resideix en el fet que és una eina de la competició és a dir, compta amb tots els equips i jugadors que participen en aquesta competició així com amb totes les dades estadístiques disponibles amb la qual cosa, per qualsevol amant i seguidor d'aquest esport és molt atractiu poder tenir un equip amb les principals estrelles d'aquesta competició esportiva.

El problema està quan es vol adaptar aquesta idea de competició a altres categories d'aquest esport siguin semiprofessionals i/o amateurs (tant masculines com femenines): no existeix cap aplicació oficial més d'aquest nivell per a la resta de les principals competicions[6]. Aquí és on **LVB** té el seu protagonisme: vol intentar cobrir aquest espai vacant i donar l'opció a aquells clubs i federacions que vulguin emular el funcionament del Supermanager.

**LVB** ofereix la possibilitat que un club on algun dels seus primers equips participi en alguna de les competicions sènior gestionades per la Federació Espanyola de Baloncesto pugui oferir als seus integrants que es converteixin en gestors i emulin a l'entrenador del primer fitxant i venent jugadors.

Tot i el punt anterior, **LVB** no és exclusiu d'aquesta competició, ja que **LVB** permet a un club registrar qualsevol jugador sempre que se li subministri un flux de dades estadístiques per tal de poder actualitzar la puntuació i valoració d'aquest jugador. És a dir, el que **LVB** no ofereix és la generació aleatòria de dades estadístiques, sinó que aquestes han de provenir d'una font real vàlida.

---

6 Consultar l'annex: “Resum de les principals competicions de basquet a l'Estat”.



## **2.2 Públic objectiu i perfils d'usuari**

D'una banda, el públic al qual s'adreça aquesta aplicació és a qualsevol club de bàsquet amateur o semiprofessional on el seu primer equip competeixi en una competició d'àmbit estatal o autonòmic on es registrin, de manera fiable, les dades estadístiques dels jugadors durant el transcurs d'una temporada esportiva.

D'altra banda, el públic objectiu final és, per extensió, tots els membres integrants d'aquest club (jugadors, entrenadors, personal de suport, famílies, aficionats) que vulguin tenir l'oportunitat de sentir-se un entrenador d'un primer equip, poder comprar i vendre, de manera virtual, els jugadors del primer equip del club i competir contra companys d'equip i altres membres del club per veure qui és millor gestor.

Com que el públic objectiu final acaba sent els integrants d'una entitat, la gran majoria dels membres d'aquesta són jugadors i jugadores menors de 16 anys. Tot i que aquest perfil d'usuari està molt avançat i té molt domini de les noves tecnologies, un dels objectius del projecte és que l'aplicació resultant sigui senzilla, intuïtiva i fàcil de fer servir per arribar al màxim d'usuaris possibles.

## **2.3 Identificació dels riscos i pla de contingència**

Un dels punts importants a tenir en compte en la planificació del projecte són els possibles riscos que puguin aparèixer i alterin de forma substancial el correcte desenvolupament de tot el projecte. A l'hora de realitzar la planificació de les tasques ja he tingut en compte d'una banda allargar una mica la durada establerta dels diferents blocs de tasques i, de l'altra, reservar l'última setmana abans de l'entrega sense moltes tasques tot plegat encaminat a disposar d'un petit matalàs per poder dedicar per acabar algunes tasques de la implementació.

Tot i això, un dels possibles punts crítics del projecte és l'apartat a realitzar la integració de forma automàtica de les dades estadístiques oficials dins de l'aplicació, ja que no depèn en exclusiva de l'aplicació sinó que està supeditat a com obtenir i processar aquestes dades. Com a contingència a aquest punt, es decideix implementar dues formes addicionals d'incloure les dades estadístiques: a partir de la incorporació d'un arxiu d'Excel (càrrega múltiple de registres) i a través d'un formulari dins de l'aplicació (càrrega d'un sol registre cada cop), d'aquesta forma es podran seguir incorporant les dades estadístiques a la aplicació i aquesta funcionarà sense entrebancs.

Un altre possible punt crític és el fet d'arribar a la data límit de l'entrega i no poder haver complert la planificació inicial descrita. En aquest cas, la decisió presa és la de no fer les tasques corresponents a la documentació i manuals d'ús, d'aquesta forma s'aprofitaria aquest temps per acabar altres funcionalitats més importants i, com que un dels objectius és que sigui usable i intuïtiva no crec que hi hagi cap problema en què els usuaris acabin descobrint el funcionament ràpidament.

## 2.4 LVB: cas Bàsquet Maresme Boet Mataró 3Viles

Cal recordar, com es menciona en la justificació d'aquesta memòria, que “**Liga Virtual de Baloncesto**” va sorgir com una idea del departament de comunicació del club en el que estic col·laborant com a entrenador de diversos equips de base. Doncs bé, aquest club és el Bàsquet Maresme Boet Mataró 3 Viles[7] (BM3V).



Aquest club del Maresme neix de la fusió l'any 2020 de dos club: l'AE Boet Mataró i el Bàsquet 3 Viles i gràcies a aquesta unió s'ha convertit en un dels principals club de Catalunya i de l'Estat en nombre d'equips inscrits en la Federació Catalana de Bàsquet amb un total de 50 equips (31 masculins, 16 femenins i 3 equips d'escola) i amb un total de gairebé 600 jugadors i jugadores. Donada aquesta base de jugadors i el fet de ser la unió de dues entitats, el nou departament de comunicació i xarxes socials volia trobar alguna eina per intentar buscar alguna cohesió i identificació comuna.

A més a més, es dóna també la casualitat que el BM3V és també un dels pocs clubs de Catalunya on els seus 2 primers equips (tant masculí com femení) estan competint en competicions estatals de la FEB. L'equip masculí competeix a la Liga EBA (4a en importància), sota el nom de Mataró Parc Maresme i el femení a Liga Femenina 2 (2a en importància) amb el nom d'Advisoria Maresme.

Aquest fet, el de tenir els primers equips disputant competició estatal amb dades estadístiques oficials i tots els potencials usuaris (prop de 600 només comptant jugadors) és el que va donar origen a **LVB**.

És per això que en aquesta memòria i projecte totes les dades emprades com ara els noms de jugadors i jugadores, les imatges i estadístiques són dels equips Mataró Parc Maresme i Advisoria Maresme del club Bàsquet Maresme Boet Mataró 3 Viles de la temporada 2020/21 que competeixen a la Liga EBA (Grup C5) i Liga Femenina 2 (Grup C).

---

7 <https://maresmeboet3v.com>

## 3. Requisits del projecte

### 3.1 Requisits funcionals

Gràcies als aparats anteriorment descrits, ja queda palès quina és la situació inicial de la qual parteix el projecte així com, a grans trets, quines funcionalitats hauria de cobrir per tal de poder funcionar sense cap contratemps. És per això que ja es poden determinar quins requisits de funcionalitat hauria de complir l'aplicació.

D'una banda, els requisits que ha de complir són:

- Permetre el registre d'usuaris i, posteriorment modificar les dades del perfil.
- Poder crear equips als usuaris i, un cop creats, poder comprar – vendre jugadors segons el pressupost disponible.
- Que els usuaris registrats puguin consultar la classificació dels seus equips.
- Que el usuaris registrats puguin consultar tots els jugadors disponibles així com la valoració que han fet en totes les jornades.

D'altra banda, l'aplicació ha de complir una altra sèrie de requisits que permetin tot els requisits anteriors.

- Poder gestionar (veure, editar, crear, eliminar) tant usuaris com equips i jugadors.
- Poder gestionar totes les dades complementàries, com ara llistat de posicions, nombre d'equips...
- Poder importar dades estadístiques per actualitzar el valor dels jugadors.
- Poder generar les classificacions dels equips inscrits.
- Poder crear les noves jornades perquè els usuaris puguin realitzar canvis en els seus equips.
- Poder activar o desactivar que els usuaris puguin fer canvis en els seus equips.

## 3.2 Requisits tecnològics

Un cop definits els requisits de funcionalitat que ha de complir el projecte, el següent pas és definir quines són les tecnologies necessàries per a dur a terme el desenvolupament del projecte així com determinar quines eines caldran per a poder implementar aquestes tecnologies.

### RECURSOS EMPRATS

A nivell d'eines de programari s'han emprat les següents:

- IDE de desenvolupament: Sublime Text 3
- Desenvolupament local: XAMPP (Apache, MariaDB i PHP), Node.js i Composer
- Gestió de Base de Dades: TablePlus i Dbeaver.
- Còpies de seguretat: Github.
- Planificació: Trello, Microsoft Excel.
- Altres: Filezilla, Microsoft Word, Google Chrome i Mozilla Firefox.

### TECNOLOGIES EMPRADES



**PHP:** (acrònim recursiu de *Hypertext Preprocessor*) és un llenguatge de codi obert adequat per al desenvolupament web i com pot ser incrustat en HTML.

Una de les característiques de PHP és que el codi és executat en el servidor, tot generant un arxiu HTML de resposta que és el que s'envia al client. D'aquesta manera el client rep el resultat, però no sabrà quin era el codi que s'estava executant.

Una altra dels avantatges és la seva simplicitat per a usuaris principiants, però alhora també disposa de característiques avançades per als programadors experts. A més a més, PHP pot ser emprat per a múltiples finalitats: connexió amb Bases de Dades, generació d'arxius PDF i imatges, etc.



**Javascript:** (tot sovint abreujat com a JS) és un llenguatge orientat a objectes, lleuger i interpretat, amb funcions de primera classe, més conegut per ser el llenguatge d'scripting per a pàgines Web, però també utilitzat en molts àmbits fora d'un navegador web com ara node.js o Apache CouchDB.

Es tracta d'un llenguatge d'scripting multiparadigma, basat en prototipus, que és dinàmic i suporta estils de programació funcional, imperativa i orientada a objectes.

JavaScript s'executa al costat del client del web, que es pot utilitzar per dissenyar / programar el comportament de les pàgines web en ocórrer un esdeveniment. JavaScript és un llenguatge de seqüència fàcil d'aprendre i també potent, molt utilitzat per controlar el comportament de les pàgines web.



**CSS** (Cascading Style Sheets o Fulls d'Estil en Cascada): és un llenguatge de fulls d'estil utilitzat per descriure l'aspecte i el format d'un document escrit en un llenguatge de marques com ara HTML.

CSS està dissenyat principalment per permetre la separació de contingut del document (escrit en HTML o un llenguatge de marques similar) de la presentació del document, incloent-hi elements com la disposició, colors, i fonts. Aquesta separació pot millorar l'accessibilitat al contingut, proporcionar més flexibilitat i control en l'especificació de característiques de presentació, permetre que múltiples pàgines comparteixin un format comú, i redueix complexitat i repetició en el contingut estructural (com per exemple en permetre disseny web sense taules).



**MariaDB** (base de dades): MariaDB Server és una de les bases de dades relacionals de codi obert més populars. Està fet pels desenvolupadors originals de MySQL i garanteix que es mantindrà de codi obert. Forma part de la majoria d'ofertes al núvol i és el valor per defecte de la majoria de distribucions de Linux



**Bootstrap** (Versió 4): és una biblioteca multiplataforma o conjunt d'eines de codi obert per al fàcil disseny de llocs i aplicacions web. Conté plantilles de disseny amb tipografia, formularis, botons, quadres, menús de navegació i altres elements de disseny basat en HTML i CSS, així com extensions de Javascript addicionals. A diferència de molts frameworks web, només s'ocupa del desenvolupament "front-end", no té res a veure amb el "back-end".



**Laravel** (versió 8): és una llibreria de codi obert per a desenvolupar aplicacions i serveis web amb PHP 5 i PHP 7. La seva filosofia és desenvolupar codi PHP de forma elegant i simple, evitant el "codi espagueti". Va ser creat en 2011 i té una gran influència de frameworks com Ruby on Rails, Sinatra i ASP.NET MVC[8]

---

8 MVC: Model-Vista-Controlador. és un patró de disseny utilitzat per a la implementació d'interfícies d'usuari. Aquest patró de desenvolupament de programari divideix l'aplicació en tres parts interconnectades: el model de dades, la interfície usuari i la lògica de control. El patró MVC es veu freqüentment en aplicacions web, on es pot visualitzar una pàgina HTML i el codi que proveïx de dades dinàmiques a la pàgina, el controlador és el sistema de gestió de bases de dades i el model és el model de dades.(Font: <https://ca.wikipedia.org/wiki/Model-Vista-Controlador> )

Laravel té com a objectiu ser un framework que permeti l'ús d'una sintaxi elegant i expressiva per crear codi de forma senzilla i permetent multitud de funcionalitats. Intenta aprofitar el millor d'altres frameworks i aprofitar les característiques de les últimes versions de PHP. El lema de Laravel és “ser el framework de PHP per als artesans”.

A part d'implementar el patró MVC Laravel també inclou una altra sèrie de característiques per tal d'oferir més seguretat, robustesa i fiabilitat als usuaris. Aquestes característiques són:

- Eloquent ORM: Laravel inclou Eloquent, un mapejador relacional d'objectes (ORM) que fa que sigui possible interactuar amb la vostra base de dades. Quan s'utilitza Eloquent, cada taula de base de dades té un "Model" corresponent que s'utilitza per interactuar amb aquesta taula. A més de recuperar registres de la taula de base de dades, els models eloqüents també permeten inserir, actualitzar i suprimir registres de la taula[9].
- Routing: Laravel incorpora un sistema gestor molt complet que permet gestionar totes les rutes de l'aplicació. A més, també incorpora una protecció de seguretat CSRF[10] en tots els formularis que s'envien dins de l'aplicació.
- Middleware: El middleware de Laravel “proporciona un mecanisme convenient per inspeccionar i filtrar les sol·licituds HTTP que entren a la vostra aplicació. Per exemple, Laravel inclou un middleware que verifica que l'autenticació de l'usuari de la vostra aplicació. Si l'usuari no està autenticat, el middleware redirigirà l'usuari a la pantalla d'inici de sessió de la vostra aplicació. No obstant això, si l'usuari està autenticat, el middleware permetrà que la sol·licitud continuï cap a l'aplicació”[11].
- Artisan: Artisan és la interfície de línia d'ordres inclosa amb Laravel. Artisan existeix a l'arrel de la vostra aplicació com a script artesanal i proporciona una sèrie d'ordres útils que us poden ajudar mentre creeu l'aplicació[12].
- Blade: Blade és el motor de plantilla senzill, però potent, que s'inclou amb Laravel. A diferència d'alguns motors de plantilla de PHP, Blade no us restringeix l'ús de codi PHP normal a les vostres plantilles. De fet, totes les plantilles Blade es compilen en codi PHP normal i es guarden a la memòria cau fins que es modifiquen, cosa que significa que Blade afegeix essencialment zero despeses generals a la vostra aplicació.[13]

---

9 Font: <https://laravel.com/docs/8.x/eloquent>

10 CSRF: És l'acrònim anglès de Cross-Site Request Forgeries. O bé “falsificacions de sol·licituds entre llocs” són un tipus d'explotació malintencionada mitjançant la qual es realitzen ordres no autoritzades en nom d'un usuari autenticat. (Font: <https://laravel.com/docs/8.x/csrf>)

11 Font: <https://laravel.com/docs/8.x/middleware>

12 Font: <https://laravel.com/docs/8.x/artisan>

13 Font: <https://laravel.com/docs/8.x/blade>

- Autenticació: Laravel ofereix les eines necessàries per implementar l'autenticació de forma ràpida, segura i senzilla. Laravel disposa de diversos sistemes d'autenticació i dona l'opció a l'usuari d'implementar la que cregui més convenient.
- Packages: Igual que en el cas de l'autenticació on Laravel disposa de diferents "paquets" per poder implementar de forma ràpida tot aquest entorn, existeixen gran multitud de "paquets" amb funcionalitats concretes per tal de donar un impuls a l'aplicació a desenvolupar
- Documentació i suport: Laravel compta amb una gran comunitat de desenvolupadors que s'encarreguen del seu manteniment, de donar suport als usuaris i de desenvolupar *packages* i complements.

El projecte està desenvolupat principalment amb HTML, CSS, Javascript i PHP. Les 3 primeres tecnologies, juntament amb el framework Bootstrap seran els emprats per a la part de maquetació de les pàgines web i la informació a mostrar a l'usuari tant per als usuaris com a l'administrador.

Pel que respecta a la part de programació (que inclou la gestió de la informació, lògica de l'aplicació i les connexions amb la base de dades entre altres) s'emprarà el llenguatge de programació PHP (versió 7.4.10) sota la llibreria Laravel (versió 8).

L'elecció d'aquestes tecnologies respon, simplement al fet que són tecnologies àmpliament conegudes i emprades en el desenvolupament web i que són les tecnologies amb les quals treballa normalment, per tant són amb les que estic més còmode per desenvolupar aquest projecte.

Tot i el paràgraf anterior, amb l'única eina que no estic gaire familiaritzat és amb el *framework* Laravel, ara bé, l'estandardització, robustesa, seguretat i totes les dependències que disposa així com tota la documentació disponible fa que la corba d'aprenentatge sigui força assequible i permeti desenvolupar tot el projecte de manera més àgil, estable i segura.

Finalment, pel que respecta la Base de Dades, l'elecció de MariaDB simplement és perquè és la continuació de MySQL, és de codi obert, està present en gairebé totes les distribucions de Linux i allotjaments web a més que s'integra perfectament amb Laravel.

### 3.3 Casos d'ús

Un cop definides les funcionalitats que haurà de complir l'aplicació i tal com ja s'ha insinuat en alguns dels apartats anteriors, caldrà determinar quins són els actors principals (rols) que intervindran en l'aplicació així com quins permisos i accions podran dur a terme cadascun d'ells.

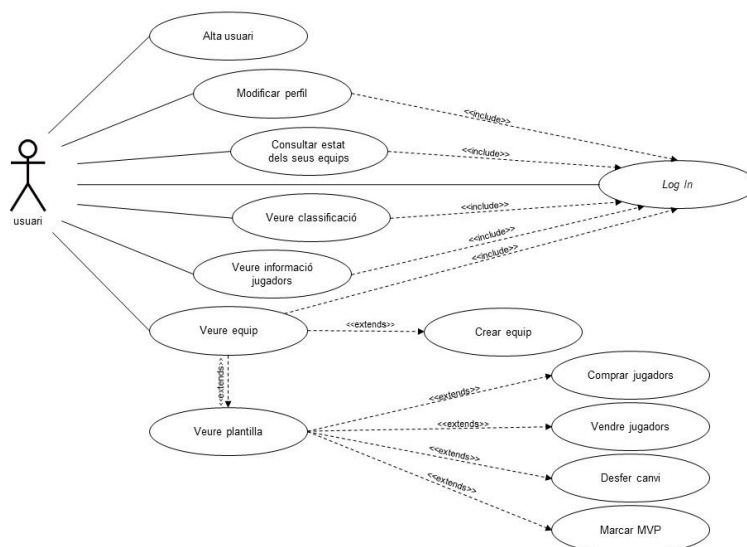
En l'aplicació intervindran tres actors principals: administrador, usuari i sistema.

- Usuari: Els usuaris principals de l'aplicació, els que "jugaran" és a dir, són els usuaris que podran crear els equips, comprar i vendre jugadors, veure la informació dels jugadors i consultar les classificacions.
- Administrador: Serà l'usuari administrador, l'únic que tindrà accés al *back-end* (zona de gestió) de l'aplicació per tal de poder gestionar el correcte funcionament de tot el conjunt.
- Sistema: No és cap persona física sinó que es tracta de la mateixa aplicació. Les funcions que té aquest rol són les de comprovar que els usuaris puguin comprar/vendre segons els paràmetres establerts, comprovar que no s'accedeixin a zones no autoritzades, generar les classificacions i actualitzar el valor dels jugadors i la puntuació dels equips.

Un cop definits els principals actors, passem a detallar els principals casos d'ús de cadascun d'ells.

### 3.3.1 Casos d'ús de l'usuari

Per tal de tenir una visió global de tots els casos d'ús de l'usuari es mostra a continuació el corresponent diagrama d'ús.



II-Il·lustració 3: Diagrama casos d'ús Usuari



Cas d'ús	<i>Alta usuari</i>
Funcionalitat	Permetre el registre d'un usuari al sistema
Actor principal	Usuari no registrat
Nivell d'objectiu	Usuari
Stakeholder i interessos	El potencial usuari vol registrar-se al sistema
Cas d'ús relacionat	---
Pre-condició	L'usuari no està registrat
Garanties en cas d'èxit	El sistema redirigirà a l'usuari a la pantalla d'inici de l'aplicació.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema demana les dades nom d'usuari, correu electrònic i contrasenya a l'usuari</li> <li>2. El sistema comprova que l'usuari no existeix</li> <li>3. El sistema dona d'alta a l'usuari</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. En cas que faltin camps per emplenar mostra missatge d'error.</li> <li>2. En cas que la contrasenya no coincideixi mostra missatge d'error.</li> <li>3. En cas que l'usuari ja existeixi mostra missatge d'error.</li> </ol>

Cas d'ús	<i>Log in</i>
Funcionalitat	Permet l'entrada d'un usuari a l'aplicació
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol accedir a la aplicació
Cas d'ús relacionat	---
Pre-condició	L'usuari ja està registrat
Garanties en cas d'èxit	El sistema redirigeix a l'usuari a la pantalla inicial de l'aplicació.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema demana el correu electrònic i la contrasenya.</li> <li>2. El sistema comprova que les credencials són correctes</li> <li>3. El sistema mostra la pantalla d'inici al sistema.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. En cas que les credencials no siguin les correctes mostra missatge d'error i permet tornar-ho a intentar.</li> <li>2. En cas que l'usuari no existeixi mostra missatge d'error i permet l'alta al sistema.</li> </ol>

Cas d'ús	<i>Modificar perfil</i>
Funcionalitat	Permet a l'usuari poder modificar la seva informació personal i canviar la imatge del perfil.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol modificar la seva informació personal, com ara: nom, correu electrònic, contrasenya i imatge de perfil
Cas d'ús relacionat	---
Pre-condició	L'usuari està autenticat al sistema
Garanties en cas d'èxit	El sistema mostra missatge indicant els canvis desats i l'usuari els pot consultar.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema mostra les dades actual a l'usuari.</li> <li>2. L'usuari modifica les dades que li mostra el sistema.</li> <li>3. El sistema valida les dades.</li> <li>4. El sistema mostra els canvis realitzats</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error al usuari.</li> </ol>

Cas d'ús	<i>Consultar estat dels equips</i>
Funcionalitat	Mostra a l'usuari el resultat obtingut pels seus equips en la última jornada disputada.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol saber el resultat dels seus equips en l'última jornada disputada
Cas d'ús relacionat	---
Pre-condició	<p>L'usuari està autenticat al sistema.  L'usuari té equips creats.  L'usuari té jugadors assignats a l'equip.  S'ha disputat la jornada.  S'han generat les classificacions de la jornada.</p>
Garanties en cas d'èxit	L'usuari veu l'estat dels seus equips proporcionada pel sistema.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema comprova que l'usuari té equips</li> <li>2. El sistema comprova que existeix la classificació de la jornada actual</li> <li>3. El sistema mostra a l'usuari l'estat dels seus equips</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. a) L'usuari no disposa d'equips</li> <li>2. a) El sistema mostra un missatge al usuari.</li> <li>1. b) No s'ha generat la classificació de la jornada actual</li> <li>2. b) El sistema mostra les dades dels equips de la jornada prèvia.</li> </ol>

Cas d'ús	<i>Veure classificació</i>
Funcionalitat	Permet a l'usuari veure la classificació general
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol conèixer la classificació de la competició
Cas d'ús relacionat	--
Pre-condició	L'usuari està autènticat al sistema. L'usuari té equips creats. L'usuari té jugadors assignats als equips. S'ha disputat la jornada. S'han generat les classificacions de la jornada.
Garanties en cas d'èxit	El sistema mostra la classificació general endreçada per punts (ascendent) i el resum dels equips de l'usuari
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema comprova que l'usuari té equips</li> <li>2. El sistema comprova que existeix la classificació de la jornada actual.</li> <li>3. El sistema mostra a l'usuari la classificació general.</li> <li>4. El sistema mostra a l'usuari l'estat dels seus equips</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. a) L'usuari no disposa d'equips</li> <li>2. a) El sistema mostra la classificació general.</li> <li>1. b) No s'ha generat la classificació de la jornada actual</li> <li>2. b) El sistema mostra la classificació general fins la última jornada computada.</li> <li>1. c) No s'ha disputat cap jornada.</li> <li>2. c) El sistema mostra un missatge a l'usuari.</li> </ol>

Cas d'ús	<i>Veure informació dels jugadors</i>
Funcionalitat	Permet a l'usuari consultar un fitxa amb la informació bàsica de cada jugador
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol saber més informació d'un jugador, com ara l'històric de punt que ha realitzat per tal de decidir si el ven o el compra per al seu equip.
Cas d'ús relacionat	---
Pre-condició	L'usuari està autènticat al sistema. El sistema disposa de les dades dels jugadors emmagatzemades.
Garanties en cas d'èxit	El sistema mostra un llistat amb els jugadors i la seva informació relativa.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema comprova que existeixen jugadors creats.</li> <li>2. El sistema retorna a l'usuari un llistat amb la informació de cada jugador.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema no disposa de jugadors registrats</li> <li>2. El sistema mostra un missatge a l'usuari.</li> </ol>

Cas d'ús	<i>Veure equips</i>
Funcionalitat	Permet a l'usuari consultar els equips creats amb la puntuació acumulada i el valor total del mateix
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol accedir als equips creats per poder accedir a la plantilla de cada equip per fer canvis.
Cas d'ús relacionat	Crear equip
Pre-condició	L'usuari està autenticat al sistema L'usuari disposa d'equips creats
Garanties en cas d'èxit	El sistema retorna un llistat amb els equips de l'usuari amb la puntuació acumulada i el valor total d'aquest.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema busca si l'usuari té equips creats</li> <li>2. El sistema retorna un llistat amb els equips de l'usuari</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. L'usuari no té equips creats.</li> <li>2. El sistema mostra a l'usuari la opció de crear un equip.</li> </ol>

Cas d'ús	<i>Crear equip</i>
Funcionalitat	Permet a l'usuari crear un nou equip per poder incloure jugadors i competir contra els altres usuaris.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol crear un equip per comprar/vendre jugadors i competir contra els altres usuaris
Cas d'ús relacionat	---
Pre-condició	L'usuari no té cap equip creat. L'usuari té menys de 3 equips creats El nom de l'equip no existeix.
Garanties en cas d'èxit	El sistema dona d'alta el nou equip.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari introdueix el nom de l'equip</li> <li>2. El sistema comprova que el nom no existeix</li> <li>3. El sistema dona d'alta l'equip</li> <li>4. El sistema mostra a l'usuari els equips creats (<i>veure equips</i>)</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El nom de l'equip ja existeix</li> <li>2. El sistema mostra un missatge d'error a l'usuari.</li> </ol>

Cas d'ús	<i>Veure plantilla</i>
Funcionalitat	Permet a l'usuari gestionar quins jugadors té disponibles.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol gestionar els jugadors que té en el seu equip
Cas d'ús relacionat	Veure equips, Crear equip
Pre-condició	L'usuari està autenticat al sistema. L'usuari té equips creats
Garanties en cas d'èxit	El sistema mostra a l'usuari un llistat amb els jugadors que pertanyen a l'equip seleccionat.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema busca els jugadors associats a l'equip indicat</li> <li>2. El sistema retorna un llistat amb els jugadors associats.</li> <li>3. Si el sistema <u>permet fer canvis</u> dona l'opció a l'usuari de <u> vendre jugador, comprar jugador, marcar l'MVP.</u></li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. L'usuari no té cap jugador associat (o no arriba al màxim de jugadors possible per equip)</li> <li>2. El sistema retorna a l'usuari un missatge d'informació.</li> <li>3. Si el sistema <u>permet fer canvis</u>, mostra a l'usuari la opció de <u>comprar jugador.</u></li> </ol>

Cas d'ús	<i>Comprar jugador</i>
Funcionalitat	Permet a l'usuari poder comprar un jugador per al seu equip
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	El jugador vol comprar un jugador per al seu equip per tal de sumar punts quan es generi la classificació
Cas d'ús relacionat	Veure plantilla, Vendre jugador
Pre-condició	L'usuari està autenticat al sistema. L'usuari disposa d'un equip creat. L'usuari disposa d'espai lliure en el seu equip per comprar un jugador. El sistema permet realitzar canvis als usuaris.
Garanties en cas d'èxit	El nou jugador comprat per l'usuari s'associa a l'equip
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema mostra a l'usuari el llistat dels jugadors disponible que no superen el valor econòmic de l'usuari per aquell equip.</li> <li>2. L'usuari escull el jugador que vol comprar</li> <li>3. El sistema associa el jugador comprat a l'equip de l'usuari.</li> <li>4. El sistema actualitza els diners disponibles per a l'equip de l'usuari restant el cost del jugador comprat.</li> <li>5. El sistema dona l'opció a l'usuari de <u>desfer el canvi</u></li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema no troba cap usuari per sota del total econòmic disponible de l'equip</li> <li>2. El sistema retorna una llista buida a l'usuari.</li> </ol>

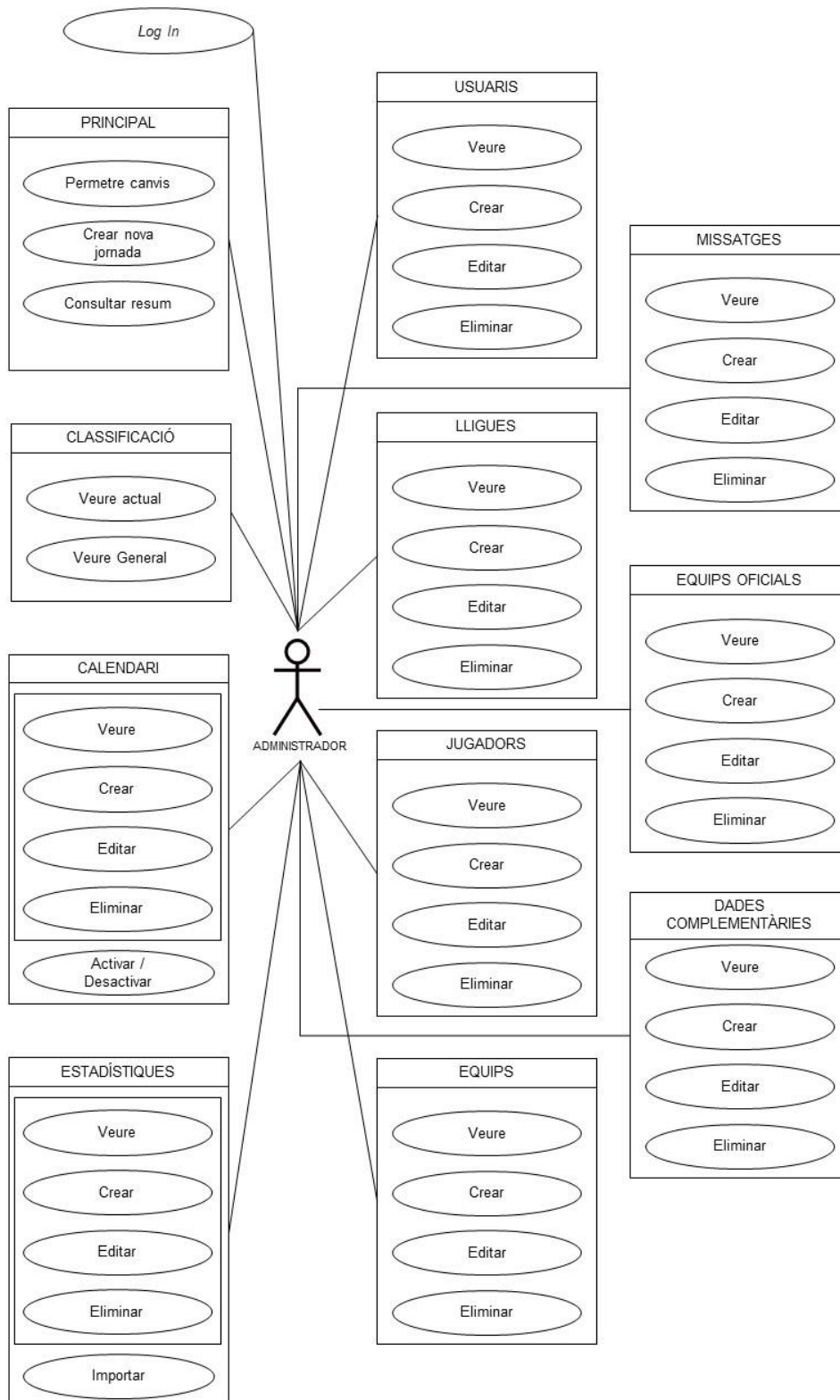
Cas d'ús	<i>Vendre jugador</i>
Funcionalitat	Permet a l'usuari vendre un dels jugadors disponibles en el seu equip
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol vendre un jugador del seu equip per tal de poder adquirir un altre jugador i millorar el seu equip.
Cas d'ús relacionat	Veure plantilla
Pre-condició	L'usuari està autenticat al sistema. L'usuari disposa d'un equip creat. L'usuari disposa de jugadors en el seu equip. El sistema permet realitzar canvis als usuaris. No vol vendre un jugador que acaba de comprar (en la jornada actual).
Garanties en cas d'èxit	El sistema ven el jugador seleccionat i permet a l'usuari poder comprar de nou.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari selecciona el jugador que vol vendre</li> <li>2. El sistema desassocia el jugador indicat.</li> <li>3. El sistema actualitza els diners disponibles per a l'equip de l'usuari sumant el preu del jugador venut.</li> <li>4. El sistema dóna l'opció a l'usuari de <u>desfer el canvi</u></li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. L'usuari no té jugadors en la plantilla</li> <li>2. El sistema no permet l'opció de vendre jugadors.</li> </ol>

Cas d'ús	<i>Desfer canvi</i>
Funcionalitat	Permet a l'usuari desfer la compra i venda que ha fet en una posició de la plantilla d'un equip.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol desfer un "canvi" que ha realitzat en la seva plantilla.
Cas d'ús relacionat	Veure plantilla
Pre-condició	L'usuari està autenticat al sistema El sistema permet fer canvis als usuaris El usuari ha venut un jugador i ha comprat un jugador en la seva posició
Garanties en cas d'èxit	El sistema torna a mostrar la plantilla amb el jugador que estava prèviament en aquella posició de l'equip.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari selecciona el jugador a desfer el canvi.</li> <li>2. El sistema desassocia el jugador seleccionat i torna a associar el que estava prèviament.</li> <li>3. El sistema actualitza el total econòmic de l'equip a l'estat anterior a la transacció.</li> </ol>
Escenari alternatiu	

Cas d'ús	<i>Marcar MVP</i>
Funcionalitat	Permet a l'usuari escollir quin creu que serà el jugador que farà millor actuació en el seu partit, cosa que permetrà a l'usuari sumar un extra de punts per al seu equip.
Actor principal	Usuari
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'usuari vol disposar d'un extra de punts per escalar posicions en la classificació general.
Cas d'ús relacionat	Veure plantilla
Pre-condició	L'usuari està autenticat L'usuari té un equip creat. L'usuari té jugadors associats a l'equip El sistema permet fer canvis
Garanties en cas d'èxit	El sistema destaca en la plantilla el jugador marcat per l'usuari com a MVP
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari selecciona el jugador que vol que sigui l'MVP.</li> <li>2. El sistema actualitza l'estat i mostra a l'usuari el canvi</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. a) L'usuari encara no ha seleccionat cap MVP</li> <li>2. a) El sistema notifica aquest fet al usuari.</li> <li>1. b) L'usuari ja te un jugador marcar i marca un altre diferent</li> <li>2. b) El sistema actualitza la selecció</li> <li>1. L'usuari ja té un jugador seleccionat i selecciona el mateix</li> <li>2. El sistema desmarca el jugador seleccionat per l'usuari i actualitza l'estat</li> <li>3. El sistema notifica a l'usuari que no té cap jugador marcat.</li> </ol>

### 3.3.2 Casos d'ús de l'administrador

Per tal de tenir una visió global de tots els casos d'ús de l'administrador es mostra a continuació el corresponent diagrama d'ús.



II-lustració 4: Diagrama casos d'ús Administrador



Cas d'ús	<i>Log in</i>
Funcionalitat	Permet l'entrada de l'usuari administrador
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	L'administrador vol accedir a l'aplicació
Cas d'ús relacionat	---
Pre-condició	L'administrador ja està registrat
Garanties en cas d'èxit	El sistema redirigeix a l'administrador a la pantalla inicial de l'aplicació.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema demana el correu electrònic i la contrasenya.</li> <li>2. El sistema comprova que les credencials són correctes</li> <li>3. El sistema mostra la pantalla d'inici al sistema.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. En cas que les credencials no siguin les correctes, mostra missatge d'error i permet tornar-ho a intentar.</li> </ol>

Cas d'ús	<i>Permetre canvis</i>
Funcionalitat	Permet a l'administrador activar o no el període de canvis per als usuaris.
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permetre als usuaris fer canvis per tal de seguir amb la competició
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat. És una jornada no disputada
Garanties en cas d'èxit	Els usuaris poden fer canvis
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'administrador escull la data i hora màximes que tenen els usuaris per a fer canvis</li> <li>2. El sistema informa als usuaris que poden fer canvis.</li> </ol>
Escenari alternatiu	

Cas d'ús	<i>Veure classificació general</i>
Funcionalitat	Permet mostrar a l'administrador la classificació general
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet a l'usuari saber quin són els equips i la puntuació que porten acumulada.
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat. S'ha generat la classificació de la jornada actual
Garanties en cas d'èxit	L'administrador pot consultar la classificació general
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema mostra a l'usuari la classificació general de tots els equips endreçats per la puntuació en ordre descendent.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error a l'administrador</li> </ol>

Cas d'ús	<i>Veure classificació de la jornada</i>
Funcionalitat	Permet mostrar a l'administrador la classificació de la jornada actual
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet a l'usuari saber quins són els equips i la puntuació que han aconseguit en la jornada actual
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat. S'ha generat la classificació de la jornada actual
Garanties en cas d'èxit	L'administrador pot consultar la classificació de la jornada
Escenari principal d'èxit	2. El sistema mostra a l'usuari la classificació general de la jornada de tots els equips endreçats per la puntuació en ordre descendent.
Escenari alternatiu	2. El sistema mostra un missatge d'error a l'administrador

Cas d'ús	<i>Crear nova jornada</i>
Funcionalitat	Permet a l'administrador iniciar el procediment per crear una nova jornada.
Actor principal	Administrador / Sistema
Nivell d'objectiu	Usuari
Stakeholder i interessos	Crear una nova jornada per a continuar amb la competició.
Cas d'ús relacionat	Sistema: Crear nova jornada
Pre-condició	L'administrador està autenticat. L'administrador a importat les dades estadístiques
Garanties en cas d'èxit	El sistema mostra la nova jornada creada
Escenari principal d'èxit	1. L'administrador escull quina és la nova jornada a crear 2. El sistema inicia el procediment per <u>generar una nova jornada</u> 3. El sistema mostra a l'administrador la nova jornada creada.
Escenari alternatiu	1. El sistema cancel·la el procediment per generar una nova jornada i mostra un missatge d'error a l'usuari.

Cas d'ús	<i>Importar estadístiques</i>
Funcionalitat	Permet a l'usuari importar dades estadístiques dels jugadors.
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet importar un arxiu .xlsx amb dades estadístiques per tal de poder calcular la puntuació dels equips, la valoració dels jugadors i la classificació.
Cas d'ús relacionat	Sistema: importar estadístiques
Pre-condició	L'administrador està autenticat. Hi ha una jornada activa que encara no s'ha importat les estadístiques
Garanties en cas d'èxit	El sistema mostra un missatge de confirmació
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'administrador selecciona el fitxer .xlsx vàlid per importar</li> <li>2. El sistema executa el cas d'ús <u>importar estadístiques</u>.</li> <li>3. El sistema mostra un missatge de confirmació</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error a l'administrador.</li> </ol>

Cas d'ús	<i>Activar / Desactivar jornada</i>
Funcionalitat	Permet establir quina és la jornada activa
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet a l'administrador establir quina és la jornada actual activa de tal forma que, per exemple, es pugui processar jornades de partits suspesos.
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat. La jornada del calendari està donada d'alta en el sistema
Garanties en cas d'èxit	El sistema mostra un missatge de confirmació
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'administrador selecciona quina és la jornada a activar</li> <li>2. El sistema marca la jornada seleccionada com a activa.</li> <li>3. El sistema mostra la nova jornada a l'administrador.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error a l'administrador</li> </ol>

Els casos d'ús que es detallen a continuació són els que fan referències a les accions de Llistar, Crear, Editar i Eliminar dels diferents models als quals l'administrador té accés. Com que l'estructura i funcionament d'aquests casos és la mateixa per a tots ells, només es mostra 1 de cada per tal de simplificar el document, sent "XXX" el nom del model en qüestió.

Cas d'ús	<i>Llistar XXX</i>
Funcionalitat	Mostra un llistat amb tot els elements del model seleccionat
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet a l'administrador veure tots els registres que existeixen del model seleccionat
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat al sistema
Garanties en cas d'èxit	El sistema mostra un llistat amb les dades requerides
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'administrador demana al sistema les dades</li> <li>2. El sistema retorna un llistat amb les dades</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error a l'administrador</li> </ol>

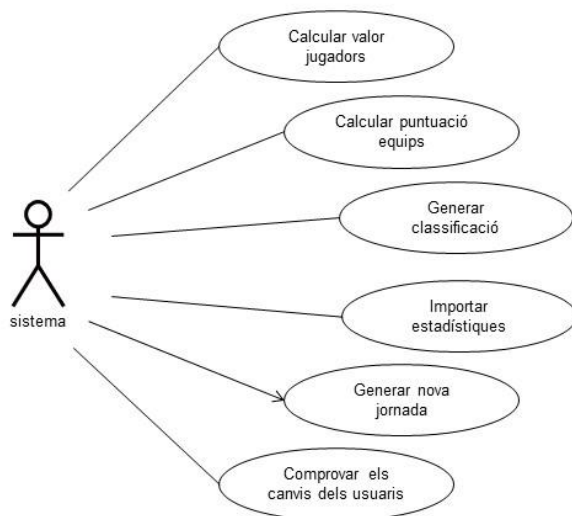
Cas d'ús	<i>Crear XXX</i>
Funcionalitat	Permet la creació d'un registre del model especificat
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet la creació d'un registre del model especificat
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat al sistema
Garanties en cas d'èxit	El sistema crea el registre
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari emplena les dades sol·licitades</li> <li>2. El sistema comprova que les dades siguin vàlides</li> <li>3. El sistema dona d'alta el registre</li> <li>4. El sistema mostra un missatge de confirmació a l'usuari</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. En cas que les dades no siguin vàlides, mostra missatge d'error</li> <li>2. En cas que es produeixin un error, el sistema mostra un missatge a l'administrador.</li> </ol>

Cas d'ús	<i>Editar XXX</i>
Funcionalitat	Permet l'edició d'un registre del model especificat
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet l'edició d'un registre del model especificat
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat al sistema El registre ja està donat d'alta en el sistema
Garanties en cas d'èxit	El sistema actualitza el registre especificat
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari emplena les dades sol·licitades</li> <li>2. El sistema comprova que les dades siguin vàlides</li> <li>3. El sistema actualitza el registre</li> <li>4. El sistema mostra un missatge de confirmació a l'usuari</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. En cas que les dades no siguin vàlides, mostra missatge d'error</li> <li>2. En cas que es produeixin un error, el sistema mostra un missatge a l'administrador.</li> </ol>

Cas d'ús	<i>Eliminar XXX</i>
Funcionalitat	Permet eliminar un registre
Actor principal	Administrador
Nivell d'objectiu	Usuari
Stakeholder i interessos	Permet a l'administrador eliminar un registre del sistema
Cas d'ús relacionat	---
Pre-condició	L'administrador està autenticat al sistema. El registre existeix en el sistema
Garanties en cas d'èxit	El sistema elimina el registre i notifica a l'administrador
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'administrador selecciona el registre a eliminar</li> <li>2. El sistema elimina el registre</li> <li>3. El sistema envia un missatge de confirmació a l'administrador.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema mostra un missatge d'error a l'administrador.</li> </ol>

### 3.3.3 Casos d'ús del sistema

Per tal de tenir una visió global de tots els casos d'ús del sistema es mostra a continuació el corresponent diagrama d'ús.



**II-lustració 5: Diagrama casos d'ús Sistema**

Cas d'ús	<i>Calcular valor jugadors</i>
Funcionalitat	Actualitzar el preu dels jugadors en funció de les dades estadístiques introduïdes
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	Actualitzar el valor dels jugadors per tal que els usuaris puguis decidir si venen o compren els jugadors
Cas d'ús relacionat	Importar estadístiques
Pre-condició	S'està executant el cas d'ús <u>importar estadístiques</u>
Garanties en cas d'èxit	El sistema actualitza el valor dels jugadors
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. Quan el sistema està desant les dades estadístiques importades.</li> <li>2. Per cada registre de jugador</li> <li>3. El sistema calcula la variació de preu a partir dels paràmetres establerts</li> <li>4. El sistema desa la variació del valor registrada.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema cancel·la la importació i mostra un missatge d'error al usuari.</li> </ol>

Cas d'ús	<i>Importar estadístiques</i>
Funcionalitat	Importa un arxiu .xlsx amb dades estadístiques
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	El sistema importa les dades estadístiques per poder generar les classificacions i actualitzar el preu i valoració dels jugadors.
Cas d'ús relacionat	Calcular valor jugadors
Pre-condició	S'ha escollit un arxiu .xlsx amb el format vàlid
Garanties en cas d'èxit	El sistema mostra un missatge notificant la correcta importació de les dades.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'arxiu a importar</li> <li>2. El sistema llegeix l'arxiu facilitat</li> <li>3. Per cada línia el sistema mira si existeix el jugador <ol style="list-style-type: none"> <li>3.1. Si el jugador existeix crea un nou registre amb les dades estadístiques</li> <li>3.2. Si el jugador no existeix, el crea i crea un nou registre amb les dades estadístiques</li> <li>3.3. Si el jugador i el registre existeix, actualitza les dades.</li> </ol> </li> <li>4. El sistema calcula el nou valor econòmic del jugador i actualitza el registre.</li> <li>5. El sistema notifica a l'usuari la correcta importació de les dades.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema cancel·la la importació de les dades i mostra un missatge d'error al usuari.</li> </ol>

Cas d'ús	<i>Calcular puntuació equips</i>
Funcionalitat	Calcula la puntuació de cada equip al generar la classificació
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	Calcular la puntuació total dels equips per tal de saber quin és l'ordre de la classificació general i de la jornada actual.
Cas d'ús relacionat	Generar classificació
Pre-condició	S'està generant una nova jornada Hi ha dades estadístiques de la jornada actual
Garanties en cas d'èxit	El sistema calcula la puntuació i permet generar la classificació.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema calcula per cada equip el total de punts obtinguts pels jugadors associats a l'equip així com el valor total econòmic del mateix (que és la suma del preu dels jugadors més el total disponible de l'equip).</li> <li>2. El sistema continua amb el procés de <u>creació de nova jornada</u>.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema cancel·la la generació de la classificació i mostra un missatge d'error a l'usuari.</li> </ol>

Cas d'ús	<i>Generar classificació</i>
Funcionalitat	Genera la classificació corresponent a la jornada actual
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	Obtenir la classificació de la jornada actual i actualitzar la classificació general per tal de saber quin és l'equip situat en primer posició.
Cas d'ús relacionat	Generar nova jornada
Pre-condició	S'ha executat el cas <u>calcular puntuació dels equips</u>
Garanties en cas d'èxit	El sistema genera la classificació i mostra la nova classificació a l'usuari administrador
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema obté la puntuació dels equips per a la jornada actual, el endreça descendentment i guarda l'ordre resultant.</li> <li>2. El sistema mostra als usuaris la nova classificació</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. El sistema cancel·la la generació de la classificació i mostra un missatge d'error a l'usuari.</li> </ol>

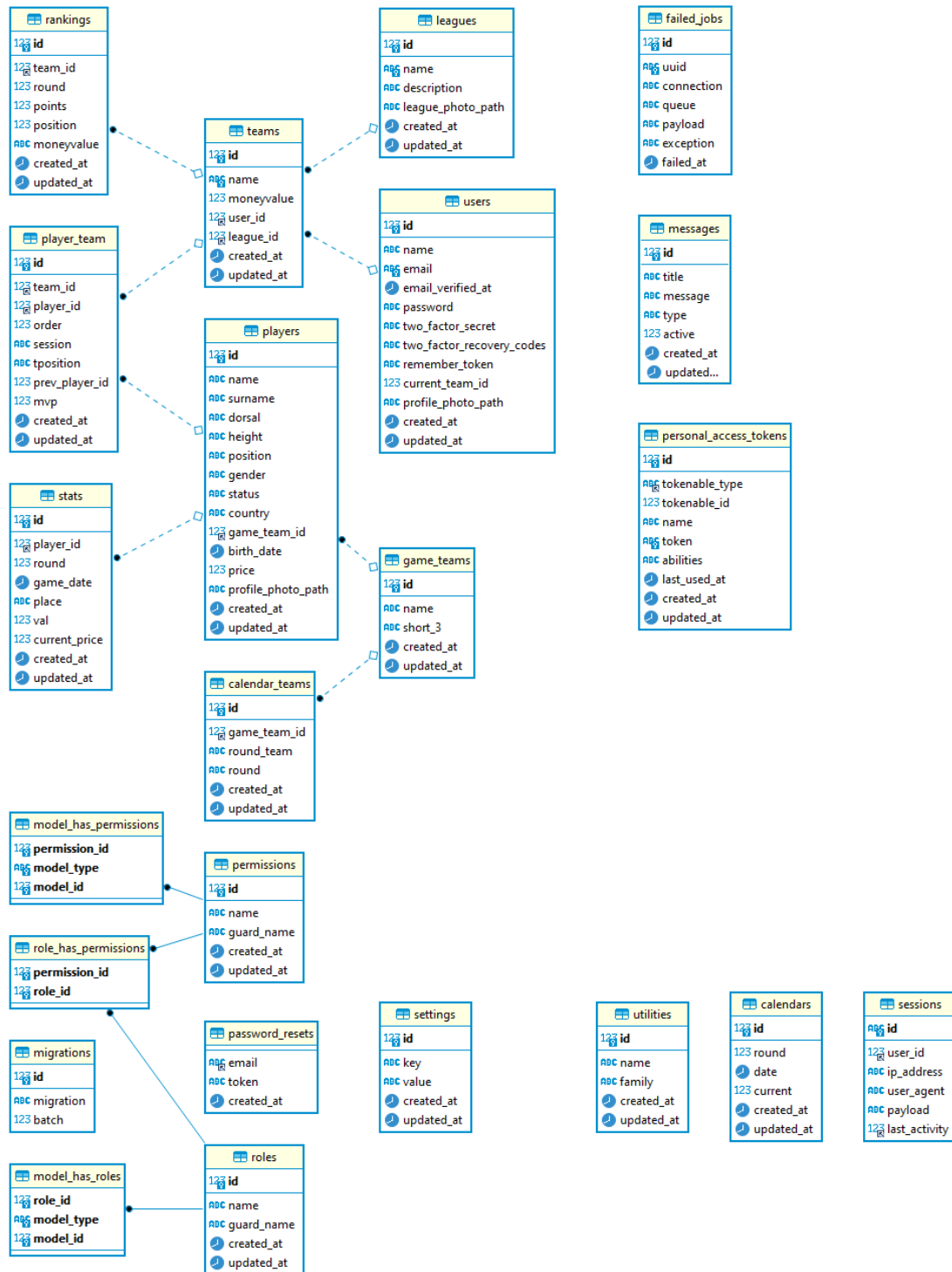
Cas d'ús	<i>Generar nova jornada</i>
Funcionalitat	Genera una nova jornada per tal que els usuaris puguin realitzar canvis i seguir amb la competició
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	Generar una nova jornada per continuar la competició
Cas d'ús relacionat	Generar classificació
Pre-condició	S'ha disputat la jornada actual. S'ha escollit la nova jornada. S'ha importat les estadístiques corresponents a la jornada actual. S'ha calculat el nou valor dels jugadors
Garanties en cas d'èxit	Es crea la nova jornada
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. El sistema <u>genera la classificació</u>.</li> <li>2. El sistema còpia les plantilles actuals amb la nova jornada.</li> <li>3. El sistema actualitza la jornada actual amb el valor de la nova jornada.</li> </ol>
Escenari alternatiu	<ol style="list-style-type: none"> <li>1. Si la jornada ja existeix, actualitza totes les dades.</li> <li>2. Si es produeix un error, cancel·la el procés i mostra un missatge d'error a l'usuari.</li> </ol>

Cas d'ús	<i>Comprovar els canvis dels usuaris</i>
Funcionalitat	Comprova que els canvis que volen fer els usuaris siguin vàlids
Actor principal	Sistema
Nivell d'objectiu	Sistema
Stakeholder i interessos	Evitar adulterar la competició i el correcte funcionament de l'aplicació
Cas d'ús relacionat	
Pre-condició	--
Garanties en cas d'èxit	Es permet fer el canvi a l'usuari o es mostra missatge d'error.
Escenari principal d'èxit	<ol style="list-style-type: none"> <li>1. Quan l'usuari vol fer un canvi (comprar o vendre un jugador)</li> <li>2. El sistema comprova que el període de canvis estigui habilitat</li> <li>3. El sistema comprova que l'usuari tingui prou diners per fer la compra.</li> <li>4. En cas afirmatiu, permet fer el canvi a l'usuari</li> <li>5. En cas negatiu, no permet fer el canvi a l'usuari.</li> </ol>
Escenari alternatiu	



### 3.4 Diagrama de la base de dades

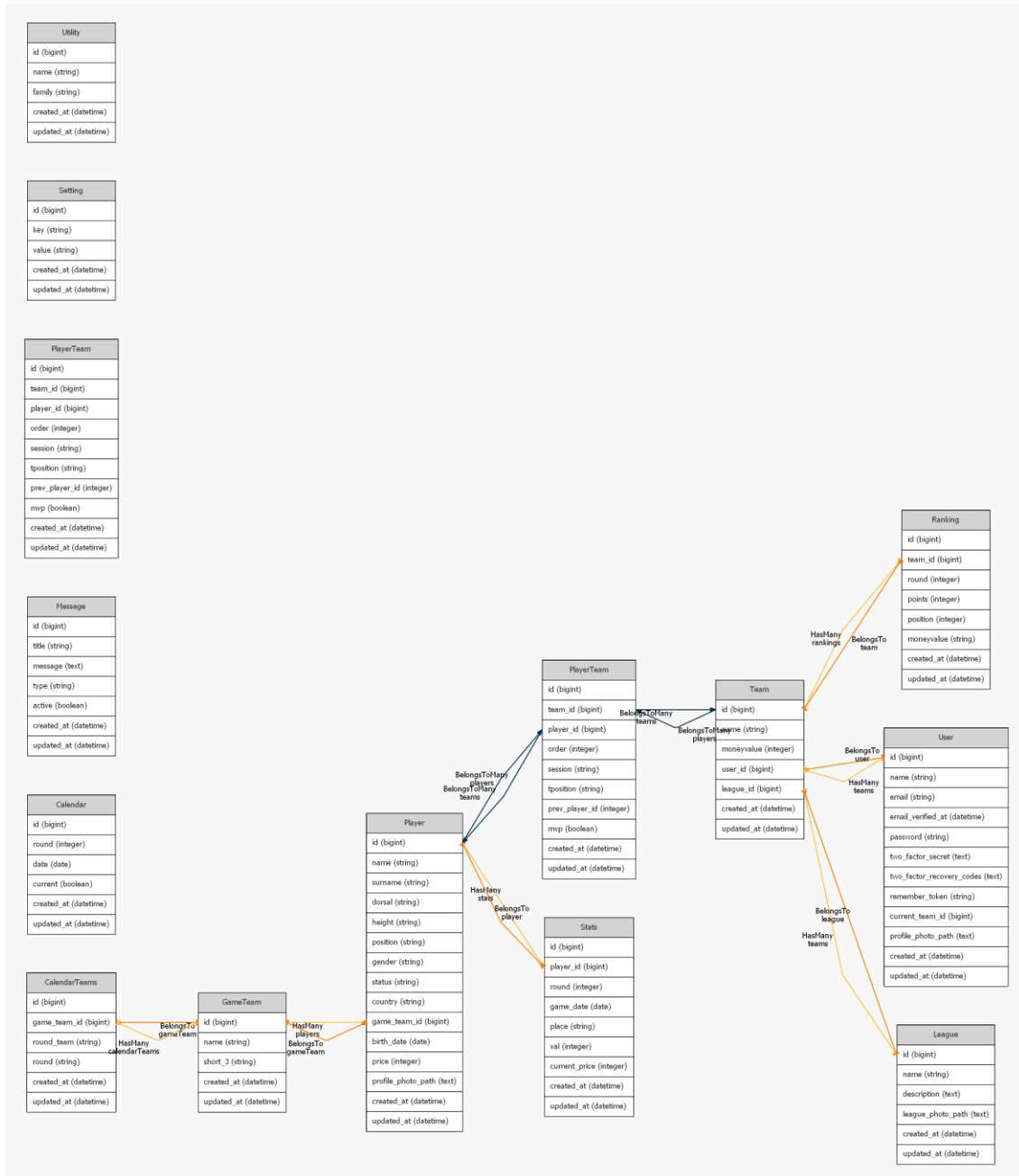
A continuació es mostra com queda definida tota l'estructura de la base de dades mostrant quines taules hi ha així com quins camps conformen cada taula i les diferents relacions que existeixen entre les diferents taules i camps de la base de dades.



II-lustració 6: Diagrama de la base de dades

### 3.5 Diagrama de classes

En la següent imatge es mostra quina és l'estructura interna de les diferents classes que conformen el conjunt de l'aplicació. D'igual forma, també es mostren quins són els atributs de cada classe i quina relació hi ha entre les diferents classes.

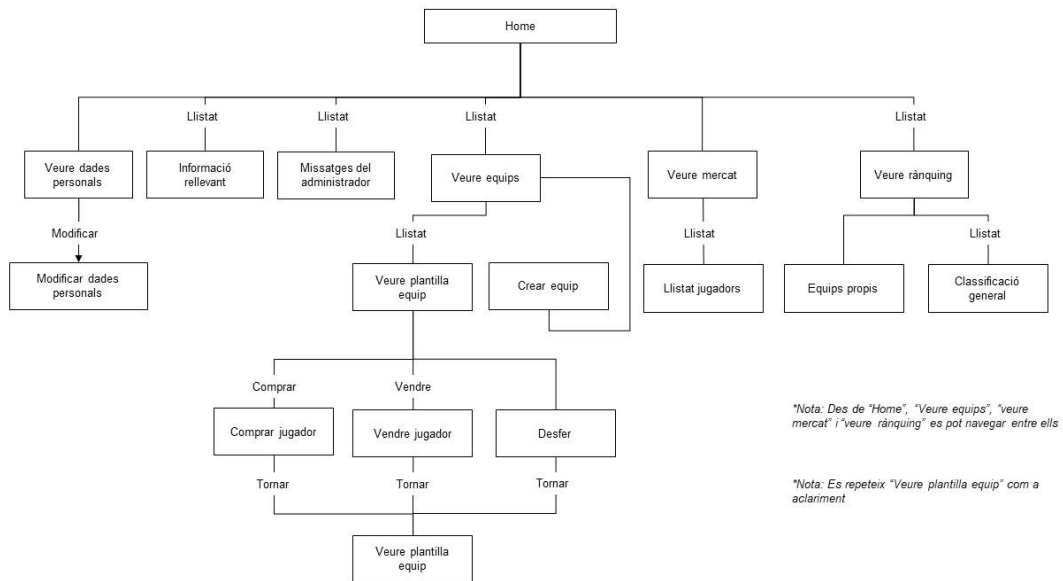


II-lustració 7: Diagrama de classes

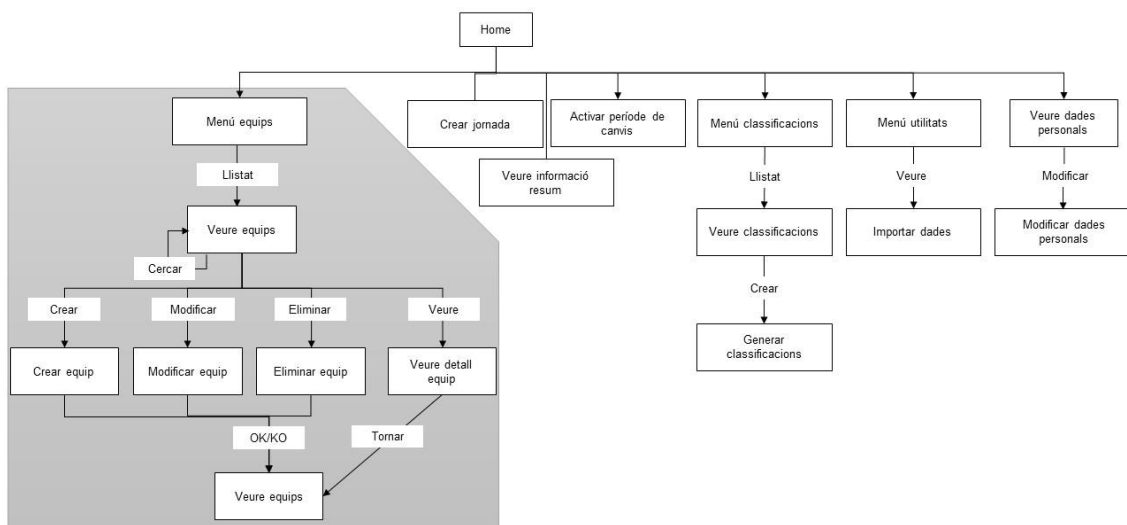
### 3.6 Diagrama de navegació

En els apartats anteriors s'ha definit quines són les funcionalitats que ha de satisfer l'aplicació i també s'ha determinat quins són els rols que hi han d'interactuar i quines accions poden realitzar. D'igual forma, s'ha exposat quina és l'estructura interna de la base de dades i la relació de les diferents classes que conformen el projecte.

Ara bé, a continuació s'exposa el diagrama de navegació tant de l'usuari com de l'administrador. L'objectiu no és altre que el de poder tenir una visió global de l'estructura de l'aplicació i veure quin és el flux bàsic de navegació entre opcions.



II-lustració 8: Diagrama de navegació dels usuaris



*\*Nota: La estructura de la navegació "Menú equips" és idèntica a la dels següents elements: Lligues, Usuaris, Jugadors, Estadístiques, Equips de joc, Utilitats. Per simplificar, només ho dibuixo una vegada*

II-lustració 9: Diagrama de navegació de l'administrador

## 4. Disseny i implementació

### 4.1 Principis generals

#### 4.1.1 Usuaris

“Liga Virtual de Baloncesto” està dissenyada pensant en els usuaris potencials que són els usuaris que han de crear els equips, comprar i vendre jugadors i, en definitiva, competir contra els altres usuaris per veure qui és el millor gestor durant una temporada de joc.

Per aquest motiu, el disseny de la part dels usuaris s’ha plantejat de forma que sigui **simple, fàcil, intuïtiva** i visualment, **atractiva** per als potencials usuaris amb l’objectiu que la corba d’aprenentatge de l’aplicació sigui molt petit i ràpidament la puguin fer servir (evitant així que els usuaris abandonin l’aplicació per complexa).

En el disseny de la part dels usuaris s’ha dissenyat de manera que la navegació per accedir al contingut sigui de forma més estàndard i simple possible. Per això s’ha decidit agrupar aquestes funcions dins de quatre grans àrees. Aquestes 4 àrees són:

- “Home”: Aquí l’usuari pot consultar informació rellevant de l’aplicació i modificar la informació del perfil.
- “Equipos”: Agrupa totes les funcions referents a la creació i llistat dels equips. Dins de cada equip estan totes les accions referents a la compra/venda de jugadors.
- “Mercado”: Permet a l’usuari consultar dades de tots els jugadors.
- “Clasificación”: Permet a l’usuari consultar la classificació general.

Amb aquesta agrupació s’aconsegueix dues coses: d’una banda que l’usuari pugui accedir a pràcticament totes les funcionalitats en un parell de passos. De l’altra, automatitzar el comportament de l’usuari per aquelles accions que requereixen més de dos passos.

Un exemple d’aquest últim punt és el fet d’afegir un jugador a la plantilla de l’equip (suposem que l’equip ja està creat). Els passos a realitzar (un cop autenticats al sistema) serien:

EQUIPOS > *escollir l’equip* > veuríem la plantilla > *comprar* > escollir jugador.

Un cop escollit el jugador el sistema ens retornaria a la pàgina de la plantilla de l’equip des d’on es podria tornar a comprar o vendre un jugador. Si, per contra l’usuari vol canviar d’equip (en el supòsit que tingués més d’un) hauria de fer el recorregut anterior.

#### 4.1.2 Administrador

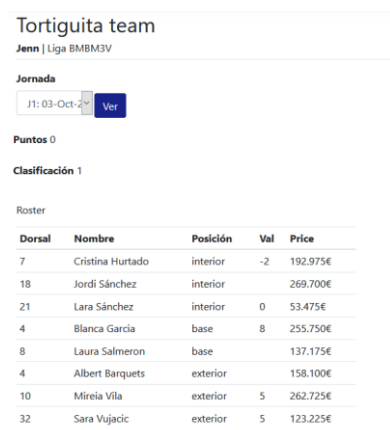
Pel que respecta al disseny de l'administrador s'ha optat per buscar més **l'eficiència** que un disseny atractiu. Per aquest motiu s'ha optat per buscar un **disseny net i simple**, però alhora que també sigui **intuïtiu**. Per això l'estructura és pràcticament igual en els diferents apartats del *back-end* de l'administrador.

Aquesta estructura general segueix el següent esquema:

Des del menú general s'accedeix al llistat general amb tots els registres referents a aquell model. Per tal de facilitar la lectura i estructuració es mostra en format taula, aquesta taula disposa de funcionalitats auxiliars proporcionades pel *plugin* de Javascript *DataTable*[14] per tal que aquesta visualització sigui la més còmoda possible.

Des d'aquest llistat l'administrador té l'opció d'eliminar, editar o visualitzar cadascun dels registres del llistat així com crear un nou registre. En el cas de veure un registre en concret, el sistema redirigeix a l'administrador a una nova pàgina amb tota la informació relacionada del registre seleccionat (aquesta pàgina de detall si que és la més diferent segons el tipus de model que s'està gestionant).

Així, per exemple, a grans trets, la pàgina de consulta d'un registre d'un equip també inclourà altra informació relacionada com ara l'usuari al que pertany, la lliga a la què està adscrit, l'ordre de la classificació (punts i valor econòmic) i la plantilla de la què disposa. En canvi, la pàgina de consulta d'un jugador mostrarà només la informació personal del jugador (amb la seva imatge) i la valoració que ha realitzat en les diferents jornades disputades.



Tortiguita team  
Jenn | Liga BMBM3V

Jornada  
J1: 03-Oct-2020 Ver

Puntos 0

Clasificación 1

Roster

Dorsal	Nombre	Posición	Val	Price
7	Cristina Hurtado	interior	-2	192.975€
18	Jordi Sánchez	interior		269.700€
21	Lara Sánchez	interior	0	53.475€
4	Blanca García	base	8	255.750€
8	Laura Salmeron	base		137.175€
4	Albert Barquets	exterior		158.100€
10	Mireia Vila	exterior	5	262.725€
32	Sara Vujacic	exterior	5	123.225€

Il·lustració 10: Detall d'un equip d'usuari



Olga Ruano



Fecha de nacimiento: 1988-09-22  
País: ESP  
Sexo: female  
Altura: 191  
Fecha creación: 21st Dec 2020

Equipo: Advisoria Maresme  
Posición: interior  
Dorsal: 14  
Estado: active

Jornada	Fecha	Place	Val	price
1	15-Nov	→	16	+42.896€

Il·lustració 11: Detall d'un jugador

14 Datatable: És un plugin fet amb jQuery que ofereix una iteració avançada a una taula d'HTML. Més informació: <https://datatables.net/>

## 4.2 Estructura de les pàgines

Un cop definits en l'apartat anterior quins són els principis que regiran tant el *back-end* dels usuaris com el *back-end* de l'administrador, el següent pas és definir de forma molt simple quina serà l'estructura que tindran aquests *back-ends* per tal de poder partir d'una base per al posterior disseny dels prototipus i implementar el disseny final amb l'aspecte final que tindrà l'aplicació.

Cal destacar que a continuació es mostra aquesta estructura inicial **un cop els diferents usuaris (i administrador) estan autenticats dins de l'aplicació.**

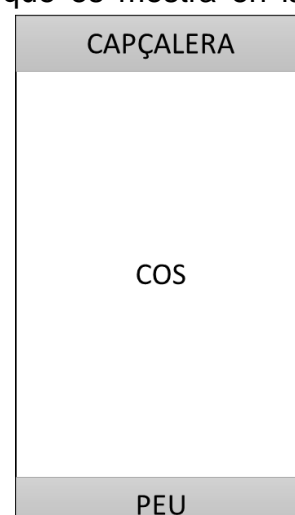
### 4.2.1 Usuaris

L'estructura principal de la part dels usuaris segueix la que es mostra en la imatge de la dreta. En aquesta imatge es diferencia clarament tres zones: **capçalera, cos i peu.**

A la **capçalera** anirà el logotip i nom de l'aplicació, l'opció de tancar sessió, el nom de la pàgina/acció en la què està l'usuari actualment i, finalment alguna altra informació necessària. Estarà fixat a la part superior de la pantalla.

Al **peu** només anirà el menú amb les diferents opcions (Home, Equips, Mercat i Classificació), indicant a l'usuari en tot moment dins de quina opció està. Estarà fixat a la part inferior de la pantalla.

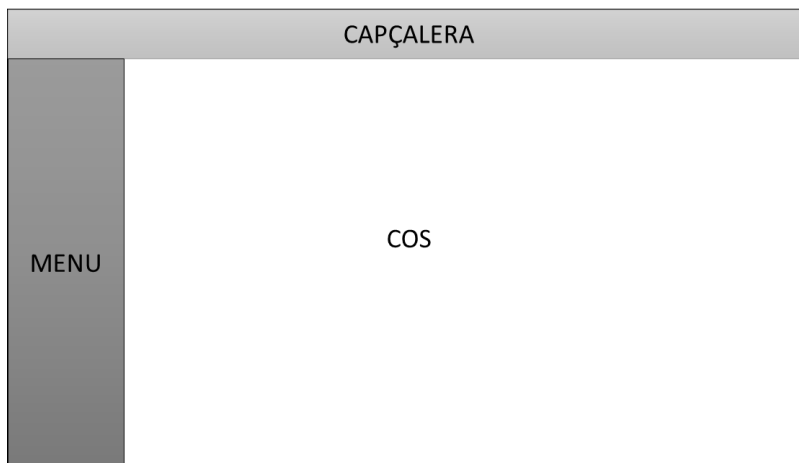
Finalment, en el **cos** anirà tota la resta del contingut amb el qual ha d'interactuar l'usuari.



**Il·lustració 12:**  
**Estructura de les pàgines dels usuaris**

### 4.2.2 Administrador

Pel que fa a l'estructura general de les pàgines del *back-end* de l'administrador, segueixen l'estil següent.



**Il·lustració 13: Estructura de les pàgines de l'administrador**

Com es pot observar en la imatge anterior, aquesta estructura segueix la mateixa estructura que molts altres dissenys de *back-end*. Tornen a diferenciar-se tres zones que són: **capçalera**, **menú** i **cos**. No hi ha la zona de peu (com a la part dels usuaris), ja que no s'havia d'ubicar cap informació rellevant.

A la **capçalera** es mostrarà el logotip de l'aplicació i l'usuari registrat (amb opció de tancar la sessió i modificar les seves dades). Aquest menú estarà situat a la part superior de la pantalla, però no de manera fixa.

Al **menú**, es mostraran els enllaços de les principals accions que pot realitzar l'administrador per a la gestió de les dades. Tal com s'ha comentat en punts anteriors, primer es mostra el llistat de tots els registres del model seleccionat i a partir d'aquí l'administrador pot escollir qualsevol de les altres opcions CRUD[15]. Aquest menú estarà fixat a la banda esquerra de la pantalla.

Finalment, en el **cos**, que ocupa la resta d'espai de la pantalla, és la "caixa" on es mostrarà tota la informació de cada pàgina així com les altres possibles accions que l'administrador pugui realitzar des de la pàgina en qüestió.

Tot plegat, tal com ja s'ha comentat anteriorment, encaminat a què sigui una estructura senzilla però a l'hora funcional.

També destacar que aquesta estructura és la que té incorporada el tema base en generar un projecte amb Laravel.

### 4.3 Estil general de l'aplicació

Com ja s'ha mencionat amb anterioritat, el projecte està orientat, principalment a un públic jove i adolescent, per tant s'ha buscat disposar d'un disseny visual que sigui, d'una banda "net", és a dir, no recarregar en excés les pantalles i

---

15 CRUD: <https://es.wikipedia.org/wiki/CRUD> És l'acrònim de "Create, Read, Update and Delete" (Crear, Llegir, Actualitzar i Eliminar), que es fa servir per a referir-se a les funcions bàsiques n bases de dades o per gestionar registres en un programari.

d'altra, atractiu, tot buscant una gamma de colors i tipografia que cridin l'atenció als usuaris que l'han de fer servir.

En aquest apartat només es mostrarà la gamma de colors i tipografies emprades per a la part dels usuaris que és la que s'ha hagut de dissenyar des de zero. De la part de l'administrador no es detalla la gamma de colors i tipografia, ja que com s'ha mencionat en el punt anterior, s'ha fet servir la que incorpora Laravel per defecte.

#### 4.3.1 Gamma de colors

Pel que fa als colors de l'aplicació s'ha decidit per emprar 2 colors principals: un color taronja i un color blau fosc juntament amb un fons blanc general. S'ha mirat de buscar un equilibri entre aquests dos colors per evitar que l'aplicació visualment estigui molt carregada.

A més a més el color blau escollit també es fa servir com color base per altres elements com ara el títol de la pàgina, botons i separació entre blocs. Per un altre costat, s'ha fet servir una tonalitat del color taronja principal per a la capçalera dels llistats així com alguns elements d'informació relacionada (no s'ha emprat el taronja principal, ja que l'efecte resultant no era prou net).

Com a altres colors auxiliars, s'ha emprat un gris per separar els registres dels llistats així com un color verd per indicar missatges i accions positius i un color vermell per a indicar les accions "perilloses".

Pel que respecta a l'estil de la resta de text, s'ha optat per un color negre (si es tracta sobre un color clar) o blanc (si es tracta d'un color fosc).



Il·lustració 14: Gamma de colors

#### 4.3.2 Tipografia

Quant a la tipografia del tema s'ha optat per l'elecció de la font **Teko**[16]. Es tracta d'una font de codi obert proporcionada per Google en el seu repositori

16 <https://fonts.google.com/specimen/Teko>



Google Fonts. Aquesta tipografia s'escapa de les tipografies més usuals que se solen trobar en aplicacions web. D'aquesta forma també s'aconsegueix un aspecte més trencador a l'aplicació. S'ha emprat aquesta tipografia en dos dels pesos disponibles: *bold* i *normal*.

A continuació és mostra els diferents estils en els que es fa servir tot indicant la mida i el color emprat.

Títol	<b>Bold, 19.2px</b>	#19248B
Subtítol	Normal, 16px	#212529
Pre títol (nom aplicació)	Normal, 13px	#212529
Títol cos	<b>Bold, 16px</b>	#212529
Subtítol cos	Normal, 13px	#212529
Títol de llistat	Normal, 16px	#212529
Subtítol de llistat	<b>Bold, 10px</b>	#212529
Botons	Normal, 12.4px	#FFFFFF
Menú inferior	Normal, 11px	#FFFFFF
Menú superior (mercado)	<b>Bold, 14px</b>	#FFFFFF
Menú superior seleccionat (mercado)	<b>Bold, 14px</b>	#19248B
Text separador (plantilla)	<b>Bold, 13px</b>	#19248B
“Badge” (posició jugador)	Normal, 12px	#212529
“Badge” (altres)	Normal, 12px	#FFFFFF
Altres	Normal, 16px	#212529

II-lustració 15: Resum tipografies emprades

#### 4.3.3 Icones i altres elements gràfics

En l'aplicació també s'ha emprat l'ús d'icones per tal d'ajudar i afavorir la comprensió per part dels usuaris i fer que l'experiència d'usuari sigui més atractiva. S'ha buscat icones tan significatives com sigui possible.

Les icones emprades provenen de dues fonts: La primera és la llibreria d'icones FontAwesome17 que incorpora Bootstrap. L'altra font d'icones és Flaticon18. Les icones emprades de Flaticon són aquelles que tenen una estètica més fina i amb elements relacionats amb el món del bàsquet (com ara les icones del menú inferior).

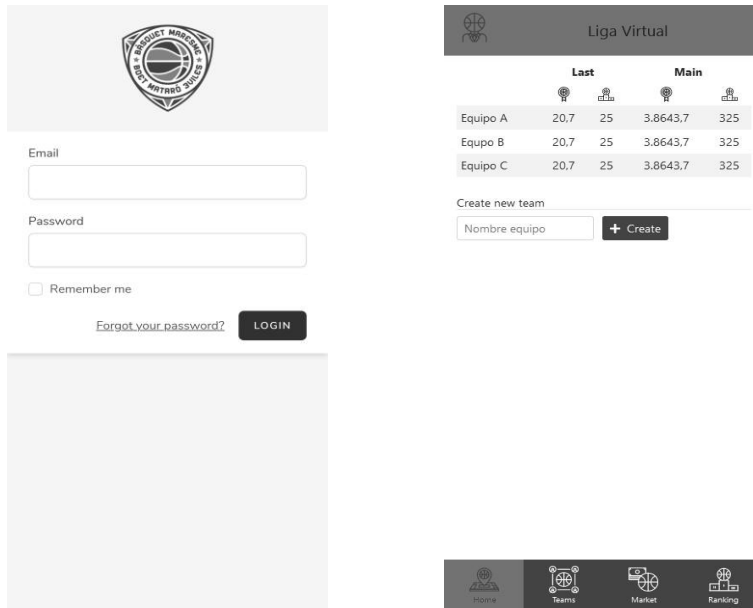
#### 4.4 Prototips

17 <https://fontawesome.com/>

18 <https://www.flaticon.com/>

A partir dels principis a tenir en compte a l'hora de dissenyar l'aplicació, sobretot el *back-end* dels usuaris a continuació es mostren els prototips inicials que es van realitzar per a posteriorment implementar el disseny de l'aplicació. Com es pot observar, aquests prototips inicials, un cop implementats, van patir algunes millores i adaptacions de disseny fins a aconseguir el resultat final.

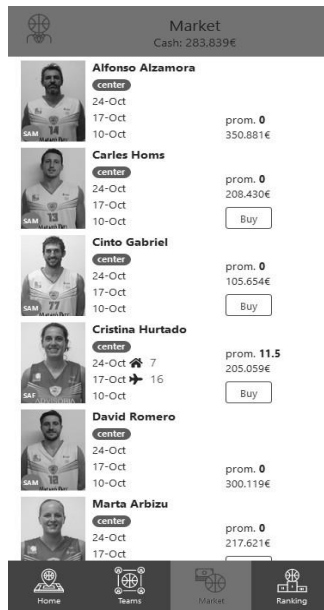
#### 4.4.1 Prototips usuari



II-lustració 16: Prototips inici sessió i pantalla inicial



II-lustració 17: Prototip de la plantilla de l'equip

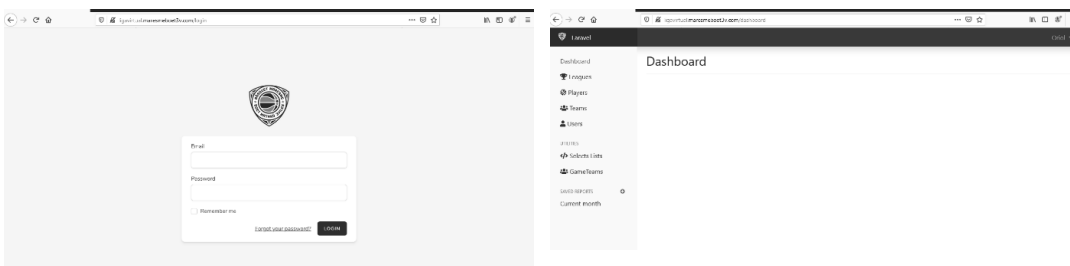


II-lustració 18: Prototip del mercat de jugadors

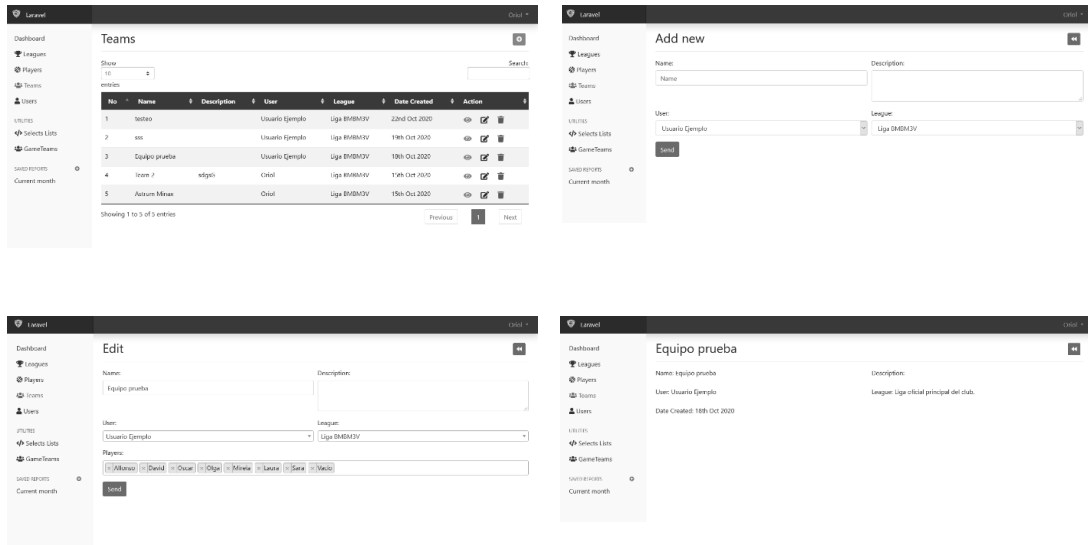


II-lustració 19: Prototip classificació general i de la jornada

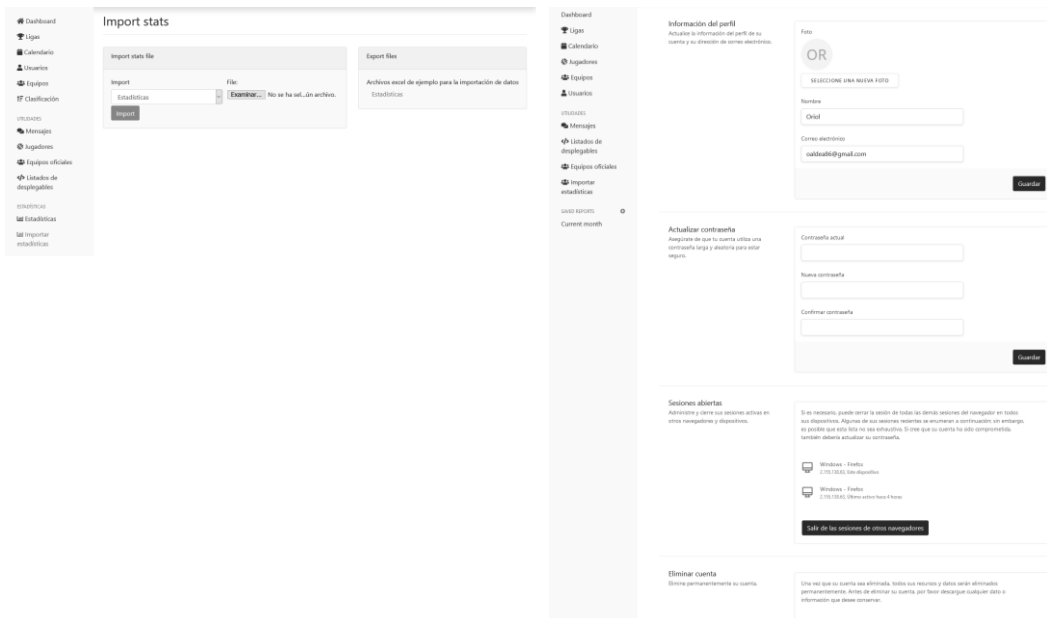
#### 4.4.2 Prototips administrador



II-lustració 20: Pantalla de login i pantalla inicial



II-lustració 21: Prototips de les pantalles CRUD



II-lustració 22: Prototip importació estadístiques i gestió del perfil

## 4.5 Creació dels models

Els models són les classes de Laravel que s'encarreguen de tot el relacionat amb la gestió de la base de dades és a dir, ens permeten consultar les dades de la base de dades així com poder escriure i/o eliminar dades. Cada model fa referència a una taula de la base de dades que és amb la qual interactua. Els diferents models es troben emmagatzemats dins de la carpeta `app/Models`.

Per crear un model es fa mitjançant la instrucció:

```
php artisan make:model League
```

o si també volem crear un arxiu *migrations* associat:

```
php artisan make:model League --m
```

L'arxiu generat fa servir la classe `Illuminate\Database\Eloquent\Model` que conté totes les funcions bàsiques per treballar amb la base de dades. En aquest arxiu del model és on es defineixen els atributs de la taula corresponent, quins mètodes pot disposar el model i també quines són les relacions entre els diferents models (i taules de la base de dades).

En el cas de la imatge inferior, es pot observar que el `LeagueModel` té dos camps (*name* i *description*) i està relacionat amb la taula *teams* (un `League` té diversos `Teams`).

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class League extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'description',
    ];

    /**
     * The accessors to append to the model's array form.
     *
     * @var array
     */
    protected $appends = [
        'profile_photo_url',
    ];
    public function teams()
    {
        return $this->hasMany('App\Models\Team');
    }
}
```

Il·lustració 23: Exemple del model Lliga

## 4.6 Creació dels controladors

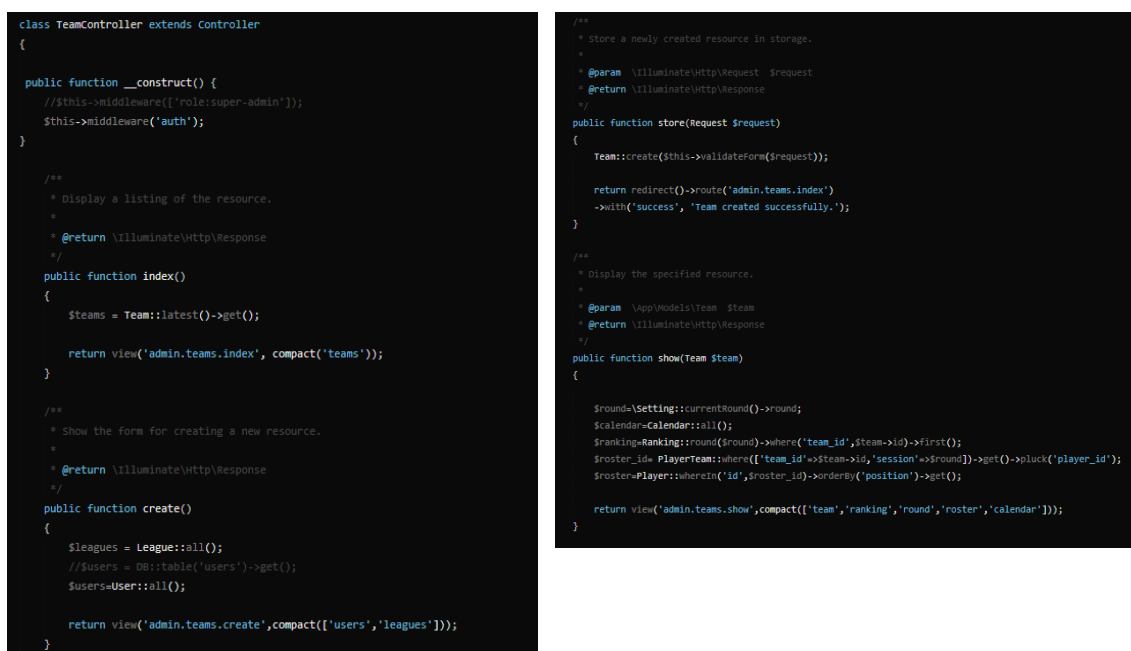
Els controladors són els encarregats d'agrupar la lògica de gestió de sol·licituds relacionada en una sola classe. Per exemple, la classe *TeamController* permet gestionar totes les sol·licituds entrants relacionades amb els equips, incloses la visualització, la creació, l'actualització i la supressió d'usuaris. Per defecte, els controladors s'emmagatzemen al directori *app/Http/Controllers*.

Per crear un controlador es fa amb la comanda:

```
php artisan make:controller TeamController
```

Ara bé, Laravel també permet crear un controlador amb els mètodes bàsics (index, create, store, edit, update, show, destroy) mitjançant la comanda:

```
php artisan make:controller TeamController --r
```



```
class TeamController extends Controller
{
    public function __construct() {
        // $this->middleware(['role:super-admin']);
        $this->middleware('auth');
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $teams = Team::latest()->get();

        return view('admin.teams.index', compact('teams'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $leagues = League::all();
        // $users = DB::table('users')->get();
        $users=User::all();

        return view('admin.teams.create', compact(['users', 'leagues']));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        Team::create($this->validateForm($request));

        return redirect()->route('admin.teams.index')
            ->with('success', 'Team created successfully. ');
    }

    /**
     * Display the specified resource.
     *
     * @param \App\Models\Team $team
     * @return \Illuminate\Http\Response
     */
    public function show(Team $team)
    {
        $round=Setting::currentRound()->round;
        $calendar=Calendar::all();
        $ranking=Ranking::round($round)->where('team_id',$team->id)->first();
        $roster_id=PlayerTeam::where(['team_id'=>$team->id,'session'=>$round])->get()->pluck('player_id');
        $roster=Player::whereIn('id',$roster_id)->orderBy('position')->get();

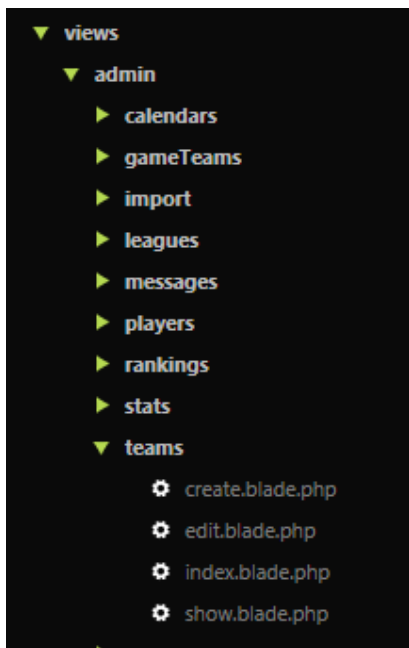
        return view('admin.teams.show',compact(['team','ranking','round','roster','calendar']));
    }
}
```

Il·lustració 24: Exemple del controlador TeamController

## 4.7 Creació de les vistes

Les vistes són bàsicament arxius amb codi HTML i són els arxius encarregats de mostrar de forma estructura tota la informació recuperada pel model i processada (si cal) pel controlador. Tots els arxius de vistes s'emmagatzemen a la carpeta *resources/views*.

Ara bé, cada vista ha de tenir per a una millor comprensió i estructuració de les vistes, aquestes es poden agrupar en carpetes.



Per exemple, en la imatge del costat es mostra que tots els arxius relacionats amb les vistes dels equips (teams) estan dins de *views/admin/teams*. La lectura i comprensió és molt més clara que si totes les vistes de tots els controladors estiguessin dins de la carpeta *views*.

Les vistes estan realitzades mitjançant un sistema de plantilles anomenat Blade. Un dels punts forts de Blade és que totes les instruccions blade (entre `{{ }}`) es tradueixen amb la funció *htmlspecialchars* de PHP.

Un altre punt fort són les “directrius” (*@section* i *@yield*), mitjançant la directriu *@section* es pot definir seccions dins de l'arxiu, cosa que permet estructurar-lo. La directriu *@yield* en canvi, permet incloure codi dins d'una secció ja

definida.

```
@section('content')
<div class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center pt-3 pb-2 mb-3 border-bottom">
  <h1 class="h2">{{__('Add new')}}</h1>
  <div class="btn-toolbar mb-2 mb-md-0">
    <div class="btn-group mr-2">
      <a class="btn btn-primary btn-sm" href="{{ route('admin.teams.index') }}" title="{{ __('Go back')}}"><i class="fas fa-backward"></i>
    </a>
    </div>
  </div>
</div>

<form action="{{ route('admin.teams.store') }}" method="POST" >
  @csrf
  <div class="form-row">
    <div class="form-group col">
      <label>{{ __('Name') }}</label>
      <input type="text" name="name" class="form-control @error('name') is-invalid @enderror" placeholder="{{ __('Name') }}" value="{{ old('name') }}">
      @error('name')
      <div class="invalid-feedback">
        {{ $message }}
      </div>
      @enderror
    </div>
    <div class="form-group col">
      <label>{{ __('Description') }}</label>
      <textarea name="description" class="form-control @error('description') is-invalid @enderror" placeholder="{{ __('Description') }}">
        {{ old('description') }}
      </textarea>
      @error('description')
      <div class="invalid-feedback">
        {{ $message }}
      </div>
    </div>
  </div>
</form>
</div>
</div>
```

**Il·lustració 25: Exemple d'arxiu blade de crear un equip des de l'administrador**

## 4.8 Creació de les rutes

Fins ara hem exposat de quina manera es creen per al projecte els diferents element que conformen el patró MVC implementat per Laravel. Ara bé, falta un punt important dins de tot l'ecosistema de Laravel: les rutes.

Les rutes, definides dins de l'arxiu *routes/web.php* són les encarregades de redirigir les diferents peticions HTTP rebudes cap al controlador corresponent. Dins de l'arxiu de rutes es pot definir quins usuaris tenen o no accés o bé quins mètodes estan definits i com ha de reaccionar l'aplicació en cas que no ho estiguin.

```
Route::get('/', function () {
    // return view('welcome');
    return view('auth.login');
    // return redirect()->route('login');
});

Route::group(['middleware' => ['role:super-admin']], function () {
    Route::prefix('admin')->name('admin.')->group(function () {

        Route::get('importExportView', [ImportExportController::class, 'importExportView']->name('importexport.index'));
        Route::get('export', [ImportExportController::class, 'export']->name('export'));
        Route::post('import', [ImportExportController::class, 'import']->name('import'));
        Route::get('dashboard/allowChanges', [DashboardController::class, 'allowChanges']->name('dashboard.allowchanges'));
        Route::put('dashboard/setChanges', [DashboardController::class, 'setChanges']->name('dashboard.setchanges'));
        Route::post('dashboard/createTeamsRound', [DashboardController::class, 'createNewRound']->name('dashboard.newround'));
        //Route::get('dashboard/createRanking', [DashboardController::class, 'createRanking']->name('dashboard.newranking'));
        Route::resource('dashboard', DashboardController::class);
        Route::resource('leagues', LeagueController::class)->shallow();
        Route::put('gameTeams/{gameTeam}/calendar', [GameTeamController::class, 'calendar']->name('gameTeams.calendar'));
        Route::resource('gameTeams', GameTeamController::class)->shallow();
        Route::resource('players', PlayerController::class)->shallow();
        Route::put('teams/{team}/round', [TeamController::class, 'round']->name('teams.round');
        Route::resource('teams', TeamController::class);
        Route::resource('users', UserController::class);
        Route::resource('utilities', UtilityController::class);
        Route::resource('stats', StatsController::class);
        Route::put('calendars/{calendar}/current', [CalendarController::class, 'current']->name('calendars.current'));
        Route::resource('calendars', CalendarController::class)->except(['show']);
        Route::put('messages/{message}/active', [MessageController::class, 'active']->name('messages.active');
        Route::resource('messages', MessageController::class);
        Route::put('ranking/round', [RankingController::class, 'round']->name('ranking.round');
        Route::resource('ranking', RankingController::class)->only(['index', 'show', 'round']);
    });
});

Route::group(['middleware' => ['role:usuario']], function () {
    Route::prefix('user')->name('user.')->group(function () {
        Route::resource('dashboard', HomeUserController::class);
        Route::resource('market', MarketController::class)->only(['index']);
        Route::get('ranking/current', [RankingUserController::class, 'current']->name('ranking.current'));
        Route::resource('ranking', RankingUserController::class)->only(['index', 'show']);
        Route::put('teams/{team}/market', [TeamUserController::class, 'market']->name('teams.market');
        Route::put('teams/buy', [TeamUserController::class, 'buy']->name('teams.buy');
        Route::post('teams/mvp', [TeamUserController::class, 'mvp']->name('teams.mvp');
        Route::get('teams/stats', [TeamUserController::class, 'stats']->name('teams.stats');
        Route::put('teams/{team}/sell', [TeamUserController::class, 'sell']->name('teams.sell');
        Route::put('teams/{team}/undo', [TeamUserController::class, 'undo']->name('teams.undo');
        Route::resource('teams', TeamUserController::class);
        //Route::post('teams/store', [TeamController::class, 'newTeam']->name('newTeam');
        //Route::get('teams/{id}/show', [TeamController::class, 'showTeam']->name('showTeam');
        // Route::get('teams/{id}/show', [TeamController::class, 'showTeam'], function () {})->name('showTeam');

    });
});

Route::middleware(['auth:sanctum', 'verified']->get('/dashboard', function () {
    $user = Auth::user();
    if ($user->hasRole('super-admin')) {
        $view = 'admin.dashboard.index';
    } else {
        $view = 'user.dashboard.index';
    }
    return redirect()->route($view);
});
```

II-lustració 26: Exemple de l'arxiu routes.php de l'aplicació



## 4.9 Importació de dades estadístiques

Un dels punts crítics per al correcte funcionament de tota l'aplicació és el fet que les estadístiques que realitzen els jugadors en els seus corresponents partits cada jornada s'han d'incorporar al sistema per tal d'actualitzar el valor dels jugadors i que els usuaris puguin fer canvis la propera jornada.

Aquesta importació de les dades es pot realitzar de dues formes tal com s'ha explicat en l'aparat d'Anàlisi d'aquest document. La primera és individualment mitjançant el formulari d'alta d'estadística de l'usuari administrador i la segona és l'alta en massa mitjançant un arxiu .xlsx. Aquest apartat es basa en aquest segon punt.

Per poder incorporar les dades mitjançant el fitxer Excel, cal un fitxer amb un format vàlid que es pot descarregar dins de l'aparat "Importar estadístiques" de l'administrador.

	A	B	C	D	E	F
1	NAME	TEAM	ROUND	GAME_DATE	HOME	VAL
2	GARCIA RIERA, BLANCA	2	9	28/11/2020	1	15
3	SERRANO RANZ, LUCIA	2	9	28/11/2020	1	7
4	HURTADO OBIOLS, CRISTINA	2	9	28/11/2020	1	9
5	SALMERON SEGU, LAURA	2	9	28/11/2020	1	4
6	VILA CABIRO, MIREIA	2	9	28/11/2020	1	13
7	ARBIZU FELIPE, MARTA	2	9	28/11/2020	1	7
8	RUANO ORIOLA, OLGA	2	9	28/11/2020	1	5
9	SORIA GARCIA, LAURA	2	9	28/11/2020	1	4
10	SANCHEZ GARCIA, LARA	2	9	28/11/2020	1	2
11	ARIAS CAPITAN, LAURA	2	9	28/11/2020	1	-6
12	VUJACIC, SARA	2	9	28/11/2020	1	8
13	GARCIA RIERA, BLANCA	2	8	21/11/2020	0	5
14	SERRANO RANZ, LUCIA	2	8	21/11/2020	0	1
15	HURTADO OBIOLS, CRISTINA	2	8	21/11/2020	0	13
16	SALMERON SEGU, LAURA	2	8	21/11/2020	0	12
17	VILA DELICADO, NEYLAN	2	8	21/11/2020	0	0
18	VILA CABIRO, MIREIA	2	8	21/11/2020	0	20
19	ARBIZU FELIPE, MARTA	2	8	21/11/2020	0	6
20	RUANO ORIOLA, OLGA	2	8	21/11/2020	0	17
21	SORIA GARCIA, LAURA	2	8	21/11/2020	0	2
22	SANCHEZ GARCIA, LARA	2	8	21/11/2020	0	0
23	ARIAS CAPITAN, LAURA	2	8	21/11/2020	0	7
24	VUJACIC, SARA	2	8	21/11/2020	0	0

II-lustració 28: Exemple del fitxer amb dades per importar

Com es pot observar en la imatge anterior, el fitxer consta de 6 columnes:

- **NAME:** Nom del jugador/a en majúscules i en format COGNOM 1 COGNOM 2, NOM
- **TEAM:** Codi de l'equip de joc donat d'alta al sistema.
- **ROUND:** Número de la jornada del calendari de l'equip (segons el calendari oficial de joc).
- **GAME\_DATE:** Data de disputa del partit.
- **HOME:** Per indicar si l'equip ha jugat el partit com a local (1) o visitant (0)
- **VAL:** Valoració estadística del jugador en el partit (proporcionat per les estadístiques oficials de la competició).

El procediment per a importar les dades estadístiques anteriors un cop l'administrador ha seleccionat i pujat el fitxer és el següent.

Per cada fila del fitxer Excel, el sistema:

- 1- Comprova quina jornada de l'aplicació es correspon la jornada indicada en el fitxer per l'equip seleccionat[19].
- 2- Seguidament el sistema busca si el jugador indicat existeix en el sistema.
  - a. Si el jugador existeix, el sistema retorna una instància *Player* que conté tota la informació del jugador.
  - b. Si el jugador no existeix, el sistema crea un nou registre de jugador mitjançant la següent instrucció:

```
$newpl=Player::create(['name'=>$nom, 'surname'=>$cognom, 'game_team_id'=>1, 'status'=>'active', 'price'=>$default_price, 'country'=>'ESP']);
```

El valor *\$default\_price* s'obté d'un valor estàndard desat a la base de dades al crear el projecte i és el preu inicial d'un jugador que no existeix en el sistema (posteriorment es pot canviar des de l'administrador si calgués).

- 3- Un cop el sistema té la instància del jugador, el següent pas és calcular el nou preu del jugador (veure següent apartat).
- 4- Finalment, el sistema guarda els valors en la base de dades. Ara bé, com que es pot donar la situació de què es pugin dades que ja han estat introduïdes prèviament (per exemple que les dades inicials estiguin malament) el sistema primer comprova si existeixen, en cas afirmatiu fa una actualització i si no, les insereix.

#### 4.10 Càlcul del preu dels jugadors

Dins del procés d'importació de les estadístiques descrit en l'apartat anterior, hi ha un punt que cal determinar quin és el nou preu que tindrà el jugador en qüestió (sigui existent o de nova creació). De fet, no es tracta de calcular el nou preu, sinó que es tracta de determinar quina és la variació (positiva o negativa) total respecte el preu base (*default\_price*) per a la jornada en qüestió.

Per tal de calcular aquesta variació el sistema simplement recupera la valoració realitzada pel jugador i la multiplica per un valor per defecte anomenat *preu\_per\_punt*[20]. Ara bé, aquest no és la variació definitiva perquè per un tema de competitivitat amb la resta d'usuaris i per evitar pujades i baixades massa exagerades en la classificació, aquesta variació respecte al preu està

---

19 Cal tenir en compte, per exemple, que en el cas de la implementació de l'aplicació amb el Bàsquet Maresme es tracta de 2 equips (masculí i femení) que els seus calendaris no són els mateixos, és a dir, pot ser que una jornada només jugui un equip i/o en una altra juguin els dos.

20 Per exemple si el *preu\_per\_punt* és 10.000€ i un jugador ha fet una valoració de 3, tindrà un increment de +30.000€ en el seu preu.

limitada sempre per un +/-15% del preu actual del jugador[21]. És a dir que si el preu del jugador supera el valor del llindar, el preu màxim del jugador serà igual al del llindar, si no es queda el preu obtingut.

```
public function calculatePrice($val=0) {
    $default_price_val=\Setting::price_per_point();

    //Obtenir el preu actual
    $current_price=abs($this->currentPrice());
    $price_to_save=abs($default_price_val*$val);

    //Calcular el increment max i min respecte el preu (15%)
    $max_price= round($current_price*0.15);

    //Comprovar que el preu a sumar/restar sigui menor que els increments
    if ( abs($price_to_save)>$max_price ) {
        $price_to_save=$max_price;
    }

    if ($val<0) {
        $price_to_save=$price_to_save*-1;
    }

    return $price_to_save;
}
```

**Il·lustració 29: Procediment per a obtenir el preu del jugador**

#### 4.11 Càlcul de la puntuació i valor dels equips

El càlcul de la puntuació aconseguida pels jugadors d'un equip en la jornada en qüestió és important per tal que es pugui generar correctament la classificació.

El procediment per calcular aquesta puntuació i el valor total de l'equip en la jornada és el següent:

- 1- El sistema obté els jugadors associats a l'equip en la jornada en qüestió.
- 2- El sistema suma la valoració obtinguda per aquests jugadors a la jornada.
- 3- El sistema comprova si l'equip té el jugador MVP[22], en cas afirmatiu suma l'extra de punts associat.
- 4- El sistema suma el valor del preu dels jugadors associats en la jornada actual més el valor romanent de l'equip[23].

---

21 És a dir, un jugador que el seu preu actual sigui 100.000€, el seu preu mínim serà de 85.000€ i el màxim de 115.000€.

22 MVP: acrònim de Most Valuable Player o jugador més valorat del partit.

23 Aquest romanent és el total que queda del pressupost inicial després de comprar tots els jugadors que conformen la plantilla.

```

public function createRanking($new_round=0) {
    $actual=\Setting::currentRound()->round;

    $teams=Team::all();
    foreach ($teams as $team) {
        //Sumar els punts dels jugadors
        $pls=DB::table('player_team')->where(['team_id'=>$team->id,'session'=>$actual])->pluck('player_id');
        $suma_val=DB::table('stats')
        ->where('round',$actual)
        ->whereIn('player_id',$pls)
        ->pluck('val')->sum();

        //Saber si el equip te el MVP
        $max_val=Stats::where('round',$actual)->orderBy('val','desc')->pluck('val')->first();
        $mvp=Stats::where(['round'=>$actual,'val'=>$max_val])->pluck('player_id');

        $pl_mvp=DB::table('player_team')->where(['team_id'=>$team->id,'session'=>$actual,'mvp'=>true])->pluck('player_id')->first();
        if($mvp->contains($pl_mvp)) {
            $puntos_mvp=\Setting::get('mvp_points');
        } else {
            $puntos_mvp=0;
        }

        $suma_val=$suma_val+$puntos_mvp;
    }
}

```

```

$suma_money_val=DB::table('stats')
->where('round',$actual)
->whereIn('player_id',$pls)
->pluck('current_price')->sum();

$suma_money_players=DB::table('players')
->whereIn('id',$pls)
->pluck('price')->sum();

$cash=$team->moneyvalue;
$suma_money=$suma_money_val+$cash+$suma_money_players;

```

II-lustració 30: Càlcul de la puntuació de l'equip en una jornada

## 4.12 Generació de les classificacions

Un altre dels punts crítics per al correcte funcionament és el fet de calcular la classificació de la jornada i la classificació general en finalitzar cada jornada, ja que l'objectiu últim pels usuaris és que quan acabi la competició real estiguin en primera posició.

El procediment per a calcular la classificació de la jornada és relativament senzill. Dins del procediment de generar la classificació s'executa l'algoritme de càlcul de la puntuació de l'equip. Un cop el sistema té aquest valor, simplement el desa a la base de dades.

Ara bé, com que es pot donar la situació de què es pugin dades que ja han estat introduïdes prèviament (per exemple que les dades inicials estiguin malament) el sistema primer comprova si existeixen, en cas afirmatiu fa una actualització i, si no existeixen, les insereix.

Posteriorment, quan es necessari mostrar la classificació, el sistema recupera tots aquests valors per la jornada en qüestió i els endreça per ordre descendent per la puntuació obtinguda.

Per mostrar la classificació general el procediment és el mateix que el del paràgraf anterior, però tenint en compte totes les jornades que ja s'han disputat.

```
Ranking::updateOrCreate(  
    ['team_id'=>$team->id,  
    'round'=>$actual],  
    ['points'=>$suma_val,  
    'moneyvalue'=>$suma_money],  
);
```

**Il·lustració 32: Procediment per desar les dades de la classificació de la jornada a la base de dades**

```
public static function round($round=0) {  
    if($round==0) {  
        //Si no pasan round, agafar la anterior a la activa  
        $round=(is_null(\Setting::prevRound())) ? 0 : \Setting::prevRound()->round;  
    }  
  
    $result=DB::table('rankings')  
        ->select(DB::raw('sum(points) as total_points,moneyvalue as money, team_id'))  
        ->where('round',$round)  
        ->groupBy('team_id','moneyvalue')  
        ->orderByDesc('total_points')  
        ->get();  
    foreach ($result as $k=>$res) {  
        $res->position=$k+1;  
        $res->team=Team::find($res->team_id);  
    }  
    return $result;  
}
```

**Il·lustració 31: Procediment per obtenir la classificació d'una jornada (passada com a paràmetre)**

# 5. Instal·lació

## 5.1 Consideracions inicials

L'objectiu d'aquest capítol és exposar de manera breu tot el procés a seguir per a instal·lar, crear i configurar el projecte amb Laravel així com quines són les principals accions per a implementar-lo correctament.

També s'aprofita aquest capítol per detallar aquelles funcionalitats del projecte necessàries per al correcte funcionament de la competició com ara: el càlcul del preu i puntuació dels jugadors, càlcul de la puntuació i valor dels equips participants, la importació de les dades estadístiques i la generació de les classificacions entre altres.

## 5.2 Requisits d'instal·lació

Per al desenvolupament del projecte és necessari disposar d'un servidor web, un sistema gestor de base de dades i totes les llibreries necessàries del llenguatge seleccionat (en aquest cas PHP). Es pot instal·lar tots els components anteriors individualment, ara bé en el mercat existeixen solucions que integren totes aquestes tecnologies i la seva instal·lació és molt senzilla.

En el cas d'aquest projecte s'ha optat per instal·lar el software XAMPP per a Windows 10. Com diu al seu web: *"XAMPP és una distribució d'Apache completament gratuïta i fàcil d'instal·lar que conté MariaDB, PHP i Perl. El paquet d'instal·lació de XAMPP ha estat dissenyat per a ser increïblement fàcil d'instal·lar i utilitzar."*[24] A més a més, XAMPP també incorpora un gestor de bases de dades per tal de poder gestionar les bases de dades MariaDB.

Un cop es té instal·lat i configurat el que es consideraria l'entorn bàsic, el següent pas és instal·lar una sèrie d'eines que permetin la configuració de tot l'entorn així com poder afegir les dependències necessàries al projecte. Aquestes eines són:

- **Composer:** Composer és una eina per a la gestió de dependències en PHP. Us permet declarar les biblioteques de què depèn el vostre projecte i les gestionarà (instal·larà / actualitzarà).[25]
- **NPM:** NPM és un gestor de paquets per al llenguatge de programació JavaScript[26]. Vindria a ser l'equivalent de Composer però per JS.
- **Sistema de control de versions GIT:** Git és un sistema de control de versions distribuïdes de codi obert i gratuït[27]. En aquest cas s'ha optat per instal·lar el client per a Windows de GitHub[28].

---

24 Font: <https://www.apachefriends.org/es/index.html>

25 Font: <https://getcomposer.org/doc/00-intro.md>

26 Font: <https://www.npmjs.com/>

Les tecnologies i programes acabats de descriure són comuns a molts desenvolupaments web i serveis d'allotjament web. Tots disposen de força documentació a internet i àmplies comunitats de desenvolupador i suport. Per tant, és molt fàcil trobar-hi guies d'instal·lació i configuració d'aquestes. Per aquest motiu no es detallen en aquesta memòria ni en cap annex el procés d'instal·lació ni configuració.

### 5.3 Creació i configuració del projecte

Per a continuar amb la creació del projecte és necessari tenir instal·lat i configurat el que s'indica en l'apartat anterior. Un cop realitzat ja es pot crear el projecte de Laravel per continuar amb el desenvolupament. Primerament cal situar-se dins de la carpeta htdocs del XAMPP (es tracta de la carpeta pública a l'iniciar el servidor web). Un cop en aquest directori, amb l'ajuda de Composer es pot crear el projecte de Laravel amb la següent instrucció:

```
composer create-project laravel/laravel ligavirtual
```

On *ligavirtual* és el nom del projecte. Composer s'encarregarà de buscar totes les dependències necessàries i crear l'estructura de directoris i arxius requerits per al funcionament inicial.

Un cop ha finalitzat, amb el servidor web Apache iniciat si es va a la url: <http://localhost/ligavirtual> s'hauria de veure la pantalla inicial per defecte de Laravel indicant la correcta creació del projecte.

Tot i que amb l'adreça anterior ja es pot accedir a la instal·lació per defecte del projecte generat aquesta adreça no acaba de ser molt usable de cara gestionar les peticions HTTP i menys encara si posteriorment es vol publicar a un servidor web públic. Caldria canviar aquesta url per una de l'estil nomaplicacio.com, d'aquesta forma les urls resultants són més usables i a l'hora de publicar-ho no hi haurà tants problemes de configuració.

En aquest cas i tenim en compte l'url final de publicació s'ha optat per canviar la url: <http://localhost/ligavirtual> per la de <http://ligavirtual.maresmeboet3v.com>. Per a fer aquest canvi es fa mitjançant la creació d'una host virtual dins del servidor web.

El procés de creació d'aquesta host virtual[29] està àmpliament documentat en internet, ja que depenen del sistema operatiu i del servidor web es realitza d'una forma o d'una altra. Un cop s'ha creat en el servidor web, la url de l'aplicació passa a ser: <http://ligavirtual.maresmeboet3v.com> i l'arxiu vhost (que és on s'emmagatzema aquesta configuració) queda de la següent forma:

---

27 Font: <https://git-scm.com/>

28 GitHub és una aplicació web que ofereix hosting i control de versions GIT a través del navegador (Font: <https://ca.wikipedia.org/wiki/GitHub>):

29 Per crear el host virtual s'ha seguit el següent tutorial:

<https://desarrolloweb.com/articulos/configurar-virtual-hosts-apache-windows.html>

```

<VirtualHost *:80>
  ##ServerAdmin login.oaldea86@outlook.com
  DocumentRoot "C:\xampp\htdocs\ligavirtual\public/"
  ServerName ligavirtual.maresmeboet3v.com
  <Directory "C:\xampp\htdocs\ligavirtual\public/">
    Require all granted
  </Directory>
</VirtualHost>

```

**Il·lustració 33: Fitxer vhost**

## 5.4 Creació i configuració de la base de dades

Un cop ja està creat el projecte en Laravel, el següent pas és crear la base de dades, obtenir les credencials i desar aquestes credencials en el fitxer de configuració de Laravel.

Com s'ha mencionat amb anterioritat, XAMPP incorpora un gestor de base de dades anomenat phpMyAdmin[30]. Per al desenvolupament del projecte s'ha creat un nou usuari amb una base de dades associada.

Un cop tenim creada la base de dades el següent pas és incloure aquestes dades en l'arxiu de configuració de Laravel corresponent. Aquest arxiu s'anomena `.env` i està situat a l'arrel de la carpeta del projecte. Bàsicament les dades que es necessiten són: usuari, base de dades i contrasenya. En la següent imatge es mostra aquesta configuració per a l'entorn de desenvolupament.

```

APP_NAME=LigaVirtual
APP_ENV=local
APP_KEY=base64:+bWJyK74eDy7pFFIF7H3H4oaf7UURo1K9240X7JcGEo=
APP_DEBUG=true
APP_URL=http://ligavirtual.maresmeboet3v.com

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ligavirtual
DB_USERNAME=ligavirtual
DB_PASSWORD=Liga2020virtual@

```

**Il·lustració 34: Configuració de la base de dades**

## 5.5 Instal·lació i configuració dels paquets complementaris

30 PhpMyAdmin: <https://www.phpmyadmin.net/>



Després de la creació del projecte i la configuració de la base de dades el següent pas és instal·lar les dependències necessàries per al projecte. Aquestes dependències permetran obtenir funcionalitats extres afegides a part del nucli de Laravel. En aquest cas les dependències necessàries són:

## JETSTREAM:

Jetstream[31]: És un dels sistemes d'autenticació que proporciona Laravel. La seva instal·lació es fa mitjançant Composer amb la següent instrucció:

```
composer require laravel/jetstream
```

Un cop instal·lat el paquet, el següent pas és instal·lar i configurar tot el sistema jetstream i acabar les configuracions necessàries. Les instruccions per a dur-lo a terme són les següents:

```
php artisan jetstream:install livewire
npm install && npm run dev
php artisan migrate
```

D'aquesta manera ja es té instal·lat i configurat per defecte tot el sistema d'autenticació per al projecte.

## LARAVEL PERMISSION

Laravel permission[32] com el seu nom indica és un paquet per a Laravel que permet de manera senzilla i fàcil poder assignar i treure permisos a les diferents rutes i funcions del projecte. Laravel permission també permet l'ús i gestió de "rols" d'usuari. La instal·lació i configuració del paquet es fa realitzant les següents instruccions:

```
composer require spatie/laravel-permission
php artisan vendor:publish --provider="Spatie\Permission\
PermissionServiceProvider"
php artisan migrate
```

A partir d'aquí el següent pas és crear els rols necessaris i la seva configuració. Per a dur-lo a terme s'ha seguit aquesta guia: <https://spatie.be/docs/laravel-permission/v3/basic-usage/basic-usage> tot creant un **rol usuari** i un **rol superadministrador**.

## LARAVEL EXCEL

Laravel Excel[33] és un paquet per Laravel que permet d'una forma molt fàcil i senzilla treballar amb arxius Excel o CSV. Laravel Excel permet importar dades

---

31 Més informació: <https://jetstream.laravel.com>

32 Spatie laravel-permission: <https://github.com/spatie/laravel-permission>

33 Laravel Excel: <https://github.com/Maatwebsite/Laravel-Excel>

a la base de dades del projecte i a la inversa de manera molt fàcil. La instal·lació del paquet es fa realitzant la instrucció:

```
composer require maatwebsite/excel
```

## COUNTRIES

Countries<sup>[34]</sup> és un paquet per Laravel per treballar amb països.

La instal·lació del paquet es fa realitzant la instrucció:

```
composer require pragmarx/countries
```

## 5.6 Relació amb la base de dades

Una de les característiques de les quals disposa Laravel (a través de Artisan) és tot l'ecosistema per poder treballar amb la base de dades del projecte. No només en l'àmbit de fer consultes a la base de dades o d'escriptura de les mateixes sinó pel que fa a la creació i gestió de taules i d'insercions massives de dades.

Aquestes funcionalitats són:

### MIGRATE

La migració és una funcionalitat de Laravel que permet afegir o eliminar taules, columnes, claus o índex a la base de dades. Un dels punts forts de les migracions és que proporcionen al sistema un control de canvis sobre la base de dades. Les migracions estan localitzades dins de la carpeta *database/migrations*.

Una migració no és més que una classe de Laravel amb una estructura fixada i consta de dos mètodes: *up()* i *down()*. El mètode *up()* es fa servir per afegir elements a la base de dades i el mètode *down()* per treure'ls.

Per crear una migració es fa amb la comanda:

```
php artisan make:migration nom_migracio
```

Per executar totes les migracions no executades es fa amb:

```
php artisan migrate
```

---

34 Countries: <https://github.com/antonioribeiro/countries>

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateTeamsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('teams', function (Blueprint $table) {
            $table->id();
            $table->string('name')->unique();
            $table->integer('moneyvalue');
            $table->foreignId('user_id')->constrained()->onDelete('cascade');
            $table->foreignId('league_id')->constrained()->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('teams');
    }
}

```

Il·lustració 35: Exemple de migració de la taula "teams"

## Seeders

Moltes vegades quan es genera un projecte web és necessari disposar de dades bàsiques inicials per al correcte funcionament. Això és precisament el que fan els *Seeder*: incloure informació a la base de dades.

Els arxius *seeder* estan guardats en la carpeta data base/seeders. Tot i que no és obligatori, es recomana que el nom de l'arxiu sigui el "nommodelSeeder". El funcionament és senzill: es crea un arxiu seeder i s'indiquen totes les dades a inserir a la base de dades.

```

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Spatie\Permission\Models\Permission;
use Spatie\Permission\Models\Role;
use Spatie\Permission\PermissionRegistrar;

class PermisosSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        // Reset cached roles and permissions
        app()[PermissionRegistrar::class]->forgetCachedPermissions();

        $role1 = Role::create(['name' => 'usuario']);
        $role2 = Role::create(['name' => 'super-admin']);
        // gets all permissions via Gate::before rule; see AuthServiceProvider

        // create demo users
        $user = \App\Models\User::factory()->create([
            'name' => 'Usuario Ejemplo',
            'email' => 'test@example.com',
        ]);
        $user->assignRole($role1);

        $user = \App\Models\User::factory()->create([
            'name' => 'Oriol',
            'email' => 'oaldeas86@gmail.com',
        ]);
    }
}

```

Per crear un seeder es realitza amb la comanda:

```
php artisan make:seeder
UserSeeder
```

Per a executar tots els seeders creats simplement cal executar la comanda:

```
php artisan db:seed
```

Il·lustració 36: Exemple d'arxiu Seeder per a la creació de rols i usuaris

## 5.7 Publicació al servidor web públic

En els apartats anteriors s'ha estat fent referència sempre al desenvolupament de l'aplicació sempre en el servidor en local. Ara bé, un cop ja està tot el projecte acabat, el següent pas és publicar tot el projecte en un servidor públic per tal que els usuaris hi tinguin accés i es pugui fer servir.

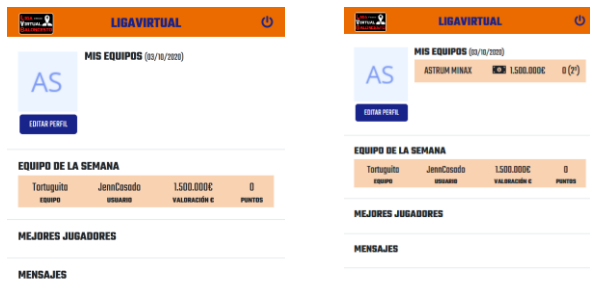
Els passos a seguir són els següents:

- 1- Restaurar la base de dades local per deixar-la amb el contingut inicial
- 2- Exportar el contingut de la base de dades en un arxiu .sql
- 3- Obtenir les dades d'accés del servidor web públic.
- 4- Crear la base de dades en el servidor web públic i accedir-hi
- 5- Importar l'arxiu .sql generat en el punt 2
- 6- Configurar l'accés FTP al servidor públic
- 7- Netejar el cache del projecte (en local)
- 8- Pujar tots els arxius del servidor en local cap al servidor públic
- 9- Un cop pujats tots els arxius, modificar l'arxiu `.env` del servidor públic per posar les noves dades de connexió a la base de dades
- 10- Comprovar que tot estigui pujat correctament.

## 6. Resultat final

Un cop implementat tot el projecte i satisfets els objectius i requisits marcats s'obté el resultat final que es presenta juntament amb aquesta memòria i que a continuació es mostra.

Com es pot observar, el disseny per als usuaris, tot i que l'estructura de les pàgines és la mateixa que descrita en l'apartat 4.2 d'aquest document, es pot apreciar que hi ha canvis d'estil (tipografia, ús de colors) i de posicionament de la informació a mostrar als usuaris.



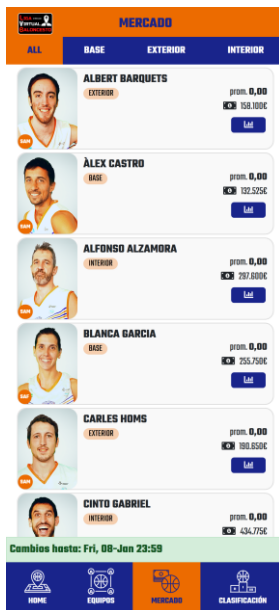
Il·lustració 37: Pantalla d'INICIO abans i després de crear un equip



Il·lustració 38: Pantalla d'EQUIPOS abans i després de crear un equip

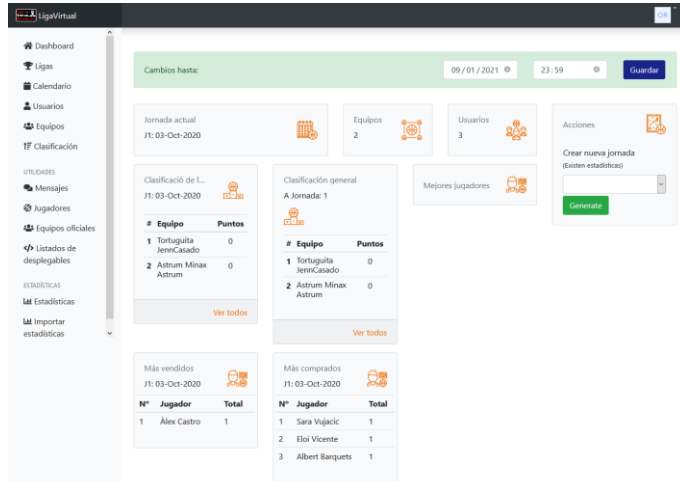


II-il·lustració 39: Pantalla EQUIPS. Detall de l'equip abans i després de comprar jugadors. A la tercera imatge s'ha seleccionat MVP

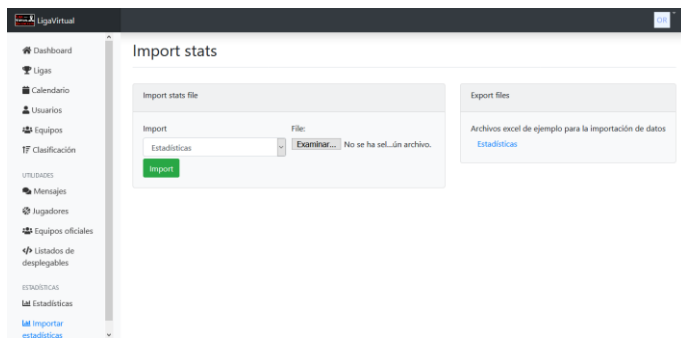


II-il·lustració 40: pantalles MERCADO i CLASIFICACIÓN

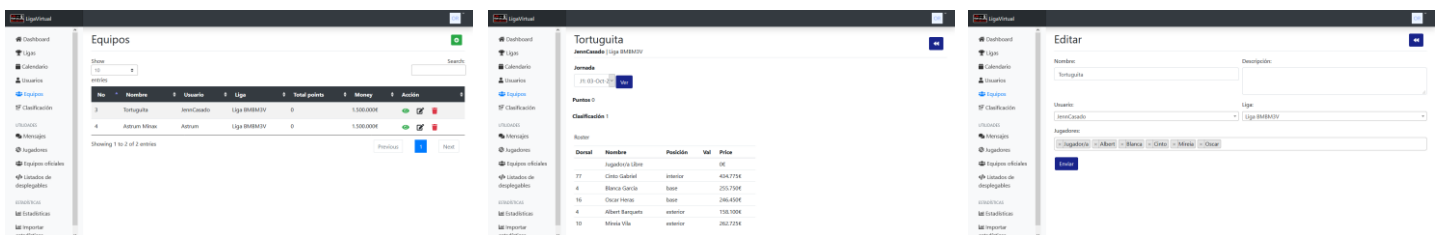
Pel que respecta a l'estil de l'apartat de l'usuari administrador, el resultat final no varia molt del prototip presentat. En les següents imatges es mostra un petit recull de les principals pantalles d'aquesta zona. Respecte a les pantalles de llistar, crear, modificar i eliminar és mostra, a tall d'exemple, les referents a EQUIPS.



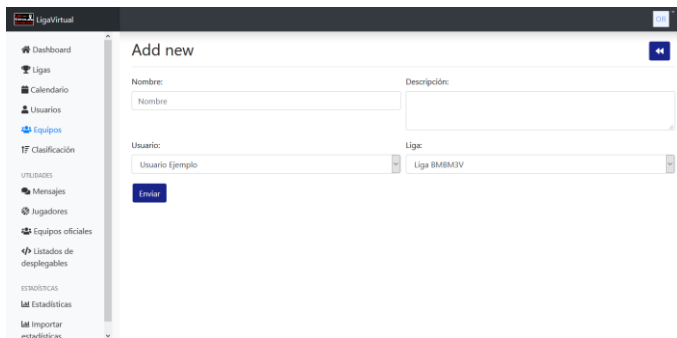
II-il·lustració 41: Dashboard en la vista d'usuari administrador



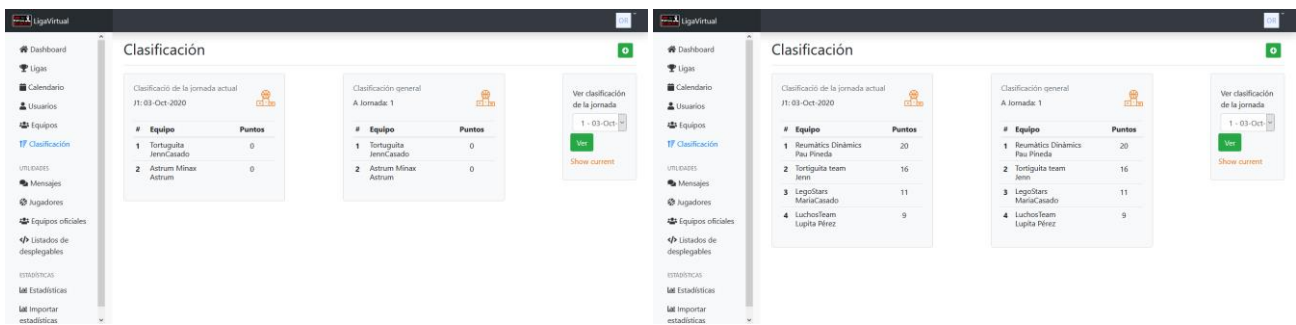
II-il·lustració 42: Pantalla IMPORTACIÓ D'ESTADÍSTIQUES en la vista d'usuari administrador



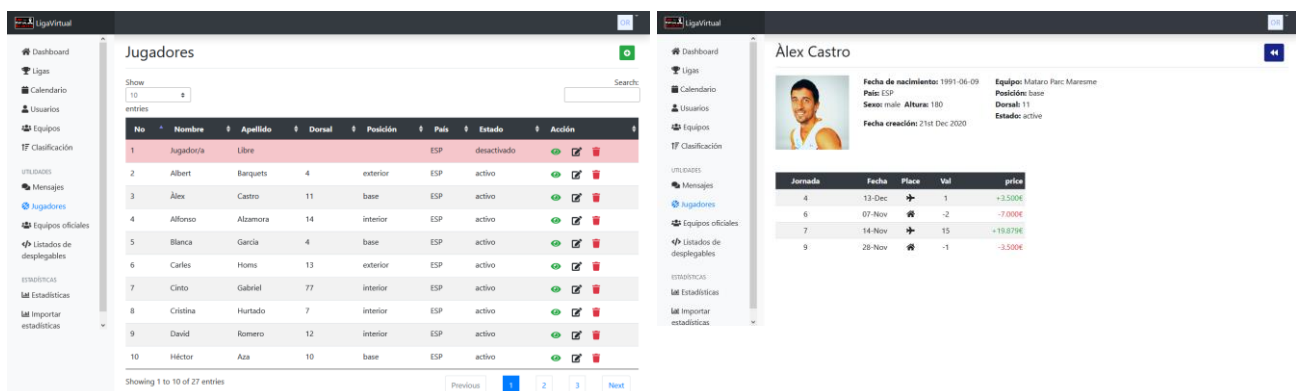
II-il·lustració 43: Pantalla EQUIPOS en la vista d'usuari administrador, amb el detall de VER EQUIPO i EDITAR EQUIPO



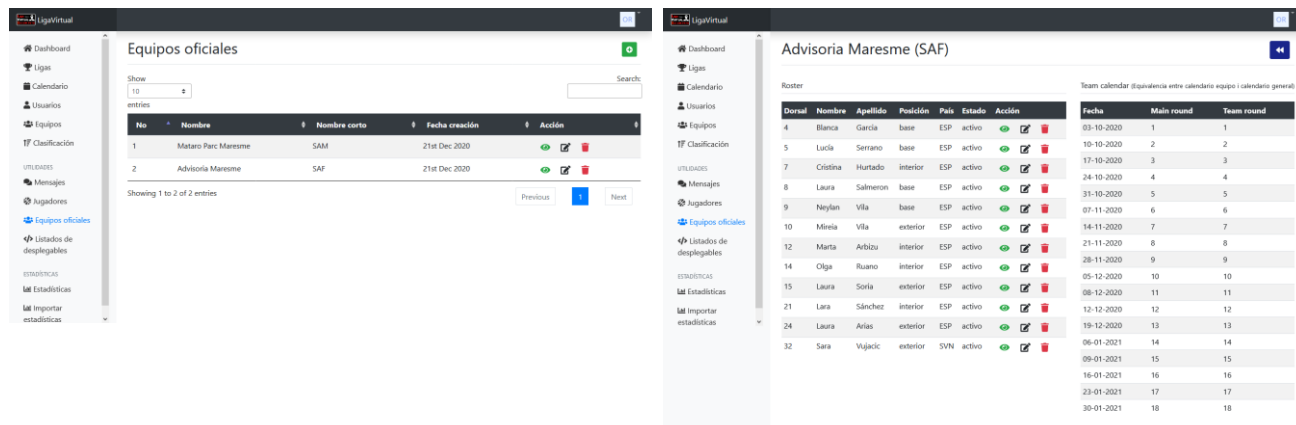
II-il·lustració 44: Pantalla EQUIPOS/CREAR NUEVO EQUIPO en la vista d'usuari administrador



II-il·lustració 45: Pantalla CLASIFICACIÓN de la vista d'usuari administrador sense i amb estadístiques importades



II-il·lustració 46: Pantalla JUGADORES i detall de jugador en la vista d'usuari administrador



II-il·lustració 47: Pantalla EQUIPOS OFICIALES i detall d'un equip en la vista d'usuari administrador



# 7. Conclusions

## 7.1 Conclusions del treball

Arribats a aquest punt del projecte, toca fer repàs a tot plegat i valorar tota la feina realitzada, el projecte resultant i tot el procés per dur-lo a terme. Tot i això, al final s'entrega un producte final funcional que compleix gran part de la planificació inicial. Així i tot, no cal dir que el projecte entregat no és un projecte tancat i està subjecte a millores futures.

A l'inici del treball es va realitzar una planificació i es van establir una sèrie d'objectius. A l'entrega del treball es pot afirmar que s'han complert la majoria dels objectius i els que no s'han complert han estat deguts a problemes aliens al desenvolupament directe del projecte.

Un dels objectius que no s'han pogut implementar ha estat el fet d'implementar un sistema d'integració automàtic de les dades estadístiques directament des del web oficial de la competició. No s'ha pogut dur a terme a causa de diversos motius.

El primer d'ells ha estat que l'organisme en qüestió (la Federación Española de Baloncesto) no disposa de cap API amb la qual aplicacions de tercers puguin nodrir-se d'aquestes dades.

El segon d'ells ha estat que l'estructura de les pàgines tampoc era prou definida per poder fer una lectura i processament de la mateixa i així poder extreure les dades i posteriorment processar-la.

El tercer motiu, és que les hores de programació necessàries per a poder arribar a implementar el descrit en el paràgraf anterior eren prou important com per fer-ho inviable si es té en compte el total d'hores disponibles per a la realització del projecte.

Com que aquest objectiu era un punt crític per al correcte funcionament es va haver de cercar alternatives que permetessin continuar amb el desenvolupament. Per aquest motiu es va optar per implementar un sistema alternatiu per a importar les dades estadístiques basat en la importació de les dades a través d'un arxiu Excel. Si bé aquest sistema no es tant transparent per a l'usuari com l'inicial previst, la solució implementada facilita poder incorporar dades a la base de dades del projecte.

A part del problema descrit en els paràgrafs anterior, mentre es realitzaven les proves i testos amb usuaris es va descobrir un fet que, si bé no era un error, feia que l'aplicació no acabés sent prou atractiva per als usuaris.

El problema trobat era que, com que LVB està pensada per implementar-la per a un club, el llistat final de jugadors disponibles per als usuaris no supera els 15 jugadors (si és un equip) o bé prop de 25 (en cas de comptar amb 2 equips,

com és el cas implementat). Aquest fet fa que al final, pràcticament tots els equips dels usuaris acabarien tenint els mateixos jugadors amb la qual cosa al final la classificació final acabaria sense grans variacions d'una jornada a l'altra.

Per tal de solucionar aquest fet i trobar una forma de dinamitzar i fer més competitiva l'aplicació es va arribar a la conclusió d'implementar un sistema per atorgar un sistema extra de punts als equips dels usuaris. Aquest "extra" ve donat mitjançant l'elecció del que l'usuari creu que serà el MVP (millor jugador) de la jornada de tots els jugadors disponibles. És a dir, l'usuari quan realitza els canvis pertinents, pots seleccionar qui creu que serà el jugador que farà la millor actuació en el seu partit. Un cop s'acaba la jornada, quan es genera la classificació, si l'usuari a encertat aquest MVP s'enduu un extra de 30 punts per al seu equip[35], d'aquesta forma s'aconsegueix que la classificació final pugui variar més substancialment cada jornada.

Un altre fet a destacar és que, si s'observa el disseny inicial dels prototips de la part dels usuaris i es compara amb el resultat final, es pot observar un canvi substancial en el disseny, aquest canvi respon a què en la fase de proves amb els usuaris es va veure que la usabilitat, tot i que era funcional, no acabava de ser prou fàcil i intuïtiva, a més a més, el disseny inicial tampoc acabava de ser prou atractiu, per això, tot escoltant les opinions dels usuaris de test es va optar per modificar-lo. Afortunadament, aquest canvi en el disseny no va suportar un canvi substancial en la planificació del projecte.

Tot i els fets descrits en els anteriors paràgrafs, com es comenta en l'inici d'aquestes conclusions el treball resultant final satisfà les exigències inicials previstes. A més a més, personalment, la realització del mateix m'ha permès aprofundir en els coneixements previs de les tecnologies emprades així com també m'ha permès poder treballar amb un *framework* com Laravel i que feia temps que volia implementar, però no trobava un projecte el suficientment atractiu com per fer-ho.

També cal destacar que tot i que la planificació s'ha seguit en els terminis indicats, la sensació personal és m'hagués agradat poder disposar de més hores per dedicar-les al projecte i així haver pogut implementar altres funcionalitats i millores en l'aplicació que, per falta de temps, no s'ha pogut dur a terme i que s'exposen en les línies de treball futur.

## 7.2 Treball futur

L'aplicació "Liga Virtual de Baloncesto" no és un producte tancat sinó que té marge per poder implementar millores que la facin encara més atractiva per als usuaris o inclús que facilitin a l'administrador realitzar altres processos de gestió. També es detallen en aquest apartat algunes línies de treball futur que m'hagués agradat que estiguessin implementades en data d'entrega d'aquesta

---

35 Aquest valor extra de 30 punts ve donat de que en les proves realitzades, la mitjana de punts aconseguits cada jornada rondava els 70 punts, per tant donar 30 punts extra es va considerar un valor suficient com per permetre un salt substancial en el total de la classificació.

memòria, però que, com s'ha comentat en les conclusions, ni tan sols s'han plantejat per falta de temps.

Així doncs, línies de millora de l'aplicació LVB són, entre altres:

### **PERMETRE LA CREACIÓ DE LLIGUES PRIVADES**

Una de les millores interessant seria la de què els usuaris poguessin crear altres lligues paral·leles a la competició oficial. D'aquesta forma hi hauria una competició oficial on participarien tots els equips creats, però també hi hauria altres lligues (que serien privades) creades pels usuaris, per exemple, per poder competir directament entre els diferents integrants d'un equip del club.

### **PERMETRE LA CONFIGURACIÓ DE LA LLIGA**

Una opció de futur seria que l'administrador pogués configurar des del *back-end* les característiques principals de la competició com ara: el nombre màxim d'equips per usuari, el preu per defecte, si està actiu l'MVP o no (i la seva puntuació) o el preu per punt.

### **INCORPORAR MÉS DADES ESTADÍSTIQUES**

Actualment l'aplicació treballa només amb el concepte estadístic de la valoració, ara bé, estaria bé poder incorporar més dades al sistema per tal que els usuaris poguessin consultar moltes més dades d'un jugador per poder valorar si el compren/venen o el conserven en els seus equips.

### **DESENVOLUPAR LA VERSIÓ ESCRIPTORI DELS USUARIS**

En un principi LVB està pensada perquè els usuaris la facin servir des de dispositius mòbils, ara bé, es pot donar el cas que algun usuari necessiti entrar des d'un ordinador. Actualment la versió d'ordinador és la mateixa que la versió mòbil per tant, caldria fer un disseny (o ajustar l'existent) per tal de tenir una versió per escriptori adaptada.

### **DESENVOLUPAR UNA APP PER A DISPOSITIUS MÒBILS**

Tot i que l'aplicació, en el seu perfil per a usuaris, està pensada per a dispositius mòbils, de cara als usuaris també és més atractiu si es disposa d'una *app* específica. Ara bé, caldria fer un estudi per valorar si caldria realitzar-la des de zero o bé mirar algun sistema d'encapsular LVB dins d'una aplicació.

## **IMPLEMENTAR LA INCORPORACIÓ D'ESTADÍSTIQUES DE FORMA REMOTA**

Es tracta d'implementar l'objectiu que no s'ha pogut dur a terme. Caldria valorar quin sistema i/o llenguatge de programació és el més adient per tal de generar una API que permeti obtenir les dades estadístiques directament de la web oficial de la Federación Española de Baloncesto. D'aquesta manera el procés de creació de jornada seria més transparent per a l'administrador.

## **IMPLEMENTAR UN SISTEMA DE COMUNICACIÓ**

Ara mateix l'aplicació permet la comunicació de l'administrador cap als usuaris mitjançant un sistema de missatges, ara bé, com a possible millora de futur estaria la d'implementar un sistema de missatge bidireccional entre els usuaris i amb l'administrador (de l'estil d'un xat o bé un fòrum).

## 8. Glossari

**Apache (servidor HTTP):** és un servidor HTTP (de pàgines web) de codi obert multiplataforma desenvolupat per Apache Software Foundation. Apache és utilitzat principalment per a servir tant contingut estàtic i pàgines web dinàmiques a la World Wide Web. Moltes aplicacions web s'han dissenyat fent servir l'entorn i les funcionalitats que Apache ofereix[36].

**API:** Una interfície de programació d'aplicacions (en anglès Application Programming Interface, API) és una interfície que especifica com diferents components de programes informàtics haurien d'interaccionar. Dit d'una altra manera, és un conjunt d'indicacions, quant a funcions i procediments, ofert per una biblioteca informàtica o programoteca per ser utilitzat per un altre programa per interaccionar amb el programa en qüestió. O, dit encara d'una altra manera, és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis.[37]

**CRUD:** És l'acrònim de "Create, Read, Update and Delete" (Crear, Llegir, Actualitzar i Eliminar), que es fa servir per a referir-se a les funcions bàsiques n bases de dades o per gestionar registres en un programari. [38]

**Framework:** Un entorn de treball (de l'anglès \*framework), o marc de treball és un conjunt estandarditzat de conceptes, pràctiques i criteris per a enfocar un tipus de problemàtica particular que serveix com a referència, per a enfrontar i resoldre nous problemes d'índole similar.

En el desenvolupament de programari, un entorn de treball és una estructura conceptual i tecnològica d'assistència definida, normalment, amb artefactes o mòduls concrets de programari, que pot servir de base per a l'organització i desenvolupament de programari. Típicament, pot incloure suport de programes, biblioteques, i un llenguatge interpretat, entre altres eines, per a així ajudar a desenvolupar i unir els diferents components d'un projecte.

Representa una arquitectura de programari que modela les relacions generals de les entitats del domini, i proveeix una estructura i una especial metodologia de treball, la qual estén o utilitza les aplicacions del domini. [39]

**Htmlspecialchars:** Funció de PHP que converteix caràcters especials en entitats HTML. [40]

**HTTP:** El protocol de transferència d'hipertext o HTTP (HyperText Transfer Protocol) estableix el protocol per a l'intercanvi de documents d'hipertext i

---

36 Font: [https://ca.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://ca.wikipedia.org/wiki/Servidor_HTTP_Apache)

37 Font:

[https://ca.wikipedia.org/wiki/Interf%C3%ADcie\\_de\\_programaci%C3%B3\\_d%27aplicacions](https://ca.wikipedia.org/wiki/Interf%C3%ADcie_de_programaci%C3%B3_d%27aplicacions)

38 Font: <https://es.wikipedia.org/wiki/CRUD>

39 Font: <https://es.wikipedia.org/wiki/Framework>

40 Font: <https://www.php.net/manual/es/function.htmlspecialchars.php>

multimèdia al web. HTTP disposa d'una variant xifrada mitjançant SSL anomenada HTTPS.

HTTP segueix un model client-servidor on el client (generalment un navegador) inicia la petició d'informació establint una connexió TCP/IP al port 80 d'una màquina remota. El servidor espera una petició (consulta o request) amb el mètode i la versió del protocol utilitzat: p. ex. "GET / HTTP/1.2". A continuació, mitjançant una semàntica específica, s'envien una sèrie de capçaleres amb extensions MIME que donen metainformació sobre el tipus de document multimèdia demanat, estat de connexió, etc. a un recurs d'Internet, la referència al qual es fa mitjançant les URI (Universal Resource Identifier), com a lloc mitjançant URL (Universal Resource Locator) o com a nom mitjançant URN (Universal Resource Name). [41]

**IDE:** Un entorn integrat de desenvolupament o IDE (acrònim en anglès de Integrated development environment), és una eina informàtica per al desenvolupament de programari de manera còmoda i ràpida. Així doncs és un entorn de desenvolupament que agrupa diferents funcions en un sol programa, habitualment: editor de codi, compilador, depurador i un programa de disseny d'interfície gràfica.[42]

**Plugin:** és una aplicació informàtica que interacciona amb una altra aplicació per aportar-li una funció o utilitat addicional, generalment molt específica, com per exemple servir com a controlador en una aplicació, per a fer així funcionar un dispositiu en un altre programa. Aquesta aplicació addicional és executada per part de l'aplicació principal. Els connectors típics tenen la funció de reproduir determinats formats de gràfics, dades multimèdia, codificar/descodificar correus electrònics, filtrar imatges de programes gràfics, etc. [43]

**MVC:** Model-Vista-Controlador. És un patró de disseny utilitzat per a la implementació d'interfícies d'usuari. Aquest patró de desenvolupament de programari divideix l'aplicació en tres parts interconnectades: el model de dades, la interfície usuari i la lògica de control. El patró MVC es veu freqüentment en aplicacions web, on es pot visualitzar una pàgina HTML i el codi que proveeix de dades dinàmiques a la pàgina, el controlador és el sistema de gestió de bases de dades i el model és el model de dades. [44]

**Node.js:** és un entorn de programació dissenyat per escriure aplicacions d'Internet escalables, notablement servidors web. Els programes estan escrits en JavaScript, utilitzant una arquitectura orientada a esdeveniments, i entrada/sortida asíncrona per tal de minimitzar el temps de sistema i maximitzar

---

41 Font: [https://ca.wikipedia.org/wiki/Protocol\\_de\\_transfer%C3%A8ncia\\_d%27hipertext](https://ca.wikipedia.org/wiki/Protocol_de_transfer%C3%A8ncia_d%27hipertext)

42 Font: [https://ca.wikipedia.org/wiki/Entorn\\_integrat\\_de\\_desenvolupament](https://ca.wikipedia.org/wiki/Entorn_integrat_de_desenvolupament)

43 Font: [https://ca.wikipedia.org/wiki/Connector\\_\(aplicaci%C3%B3\\_inform%C3%A0tica\)](https://ca.wikipedia.org/wiki/Connector_(aplicaci%C3%B3_inform%C3%A0tica))

44 Font: <https://ca.wikipedia.org/wiki/Model-Vista-Controlador>

l'escalabilitat. Node.js consisteix en el motor de JavaScript V8 de Google i de diverses llibreries incloses. [45]

**MVP:** és l'acrònim en anglès dels termes Most valuable player (jugador més valuós). Es tracta d'un trofeu atorgat al qui és considerat millor jugador en un partit, una sèrie de partits o una lliga sencera en diferents esports d'equip. Tradicionalment atorgat a la lliga NBA als Estats Units, avui en dia s'aplica aquest terme també en altres països (entre ells Espanya) i esports: hoquei, futbol Americà, o beisbol entre altres. [46]

**Lliga EBA:** La Lliga EBA (Lliga Espanyola de Bàsquet Aficionat) és una competició de bàsquet organitzada per la Federació Espanyola de Bàsquet.

Es tracta de la quarta lliga en importància de les que es disputen a Espanya, després de la Lliga ACB, la LEB Or i LEB Plata.

La lliga es disputa simultàniament en 5 grups (A, B, C, D i E) on s'integren els equips per proximitat geogràfica cobrint tot el territori espanyol. Tradicionalment els equips catalans i balears competeixen al grup C, que sovint se subdivideix en dues lligues. [47]

**Lliga Femenina 2:** La Lliga Femenina 2 és la segona màxima competició de clubs femenins de bàsquet a Espanya. L'organitza la Federació Espanyola de Bàsquet.

Es compon de dos grups de dotze equips en cadascun. Els quatre primers de tots dos quadres disputaran en una mateixa seu la fase d'ascens a Lliga Femenina. Els dos millors de cada grup passessin a les finals. El guanyador de cadascuna d'elles ascendeix esportivament a Lliga Femenina de Bàsquet. [48]

**Valoració estadística:** La valoració al bàsquet és una fórmula estadística que és utilitzada per mesurar el rendiment de jugadors i equips. És similar, però no exactament igual, a l'eficiència utilitzada a l'NBA.

S'acostuma a calcular amb la següent fórmula:  $(Punts + Rebots + Assistències + Pilotes robades + Taps + Faltes rebudes) - (Tirs de Camp Fallats + Tirs Lliures Fallats + Taps rebuts + Pèrdues + Faltes realitzades)$ . [49]

---

45 Font: <https://ca.wikipedia.org/wiki/Node.js>

46 Font: <https://ca.wikipedia.org/wiki/MVP>

47 Font: [https://ca.wikipedia.org/wiki/Lliga\\_EBA](https://ca.wikipedia.org/wiki/Lliga_EBA)

48 Font: [https://es.wikipedia.org/wiki/Liga\\_Femenina\\_2\\_de\\_baloncesto\\_de\\_Espa%C3%B1a](https://es.wikipedia.org/wiki/Liga_Femenina_2_de_baloncesto_de_Espa%C3%B1a)

49 Font: [https://ca.wikipedia.org/wiki/Valoraci%C3%B3\\_\(basquetbol\)](https://ca.wikipedia.org/wiki/Valoraci%C3%B3_(basquetbol))

## 9. Bibliografia

[https://ca.wikipedia.org/wiki/Model\\_de\\_desenvolupament\\_en\\_cascada](https://ca.wikipedia.org/wiki/Model_de_desenvolupament_en_cascada) [04/01/2021]

[https://ca.wikipedia.org/wiki/Metodologia\\_%C3%A0gil](https://ca.wikipedia.org/wiki/Metodologia_%C3%A0gil) [04/01/2021]

<https://www.apd.es/metodologia-kanban/> [04/01/2021]

<https://kanbantool.com/es/metodologia-kanban> [04/01/2021]

<https://www.php.net/manual/es/intro-whatism.php> [04/01/2021]

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript) [04/01/2021]

[https://ca.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://ca.wikipedia.org/wiki/Cascading_Style_Sheets) [04/01/2021]

[https://ca.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://ca.wikipedia.org/wiki/Bootstrap_(framework)) [04/01/2021]

<https://ca.wikipedia.org/wiki/MariaDB> [04/01/2021]

<https://es.wikipedia.org/wiki/Laravel> [04/01/2021]

<https://laravel.com/docs/8.x> [18/12/2020]

<https://mariadb.org/> [04/01/2021]

<https://datatables.net/> [15/12/2020]

<https://es.wikipedia.org/wiki/CRUD> [04/01/2021]

<https://fonts.google.com/specimen/Teko> [04/01/2021]



# 10. Annexos

## A. Resum de les principals competicions de bàsquet

A continuació es realitza un breu resum de les diferents competicions de clubs professionals, semiprofessionals i amateurs tant masculines com femenines d'equips "sènior" (és a dir amb jugadors majors de 18 anys) per tal que tots aquells lectors que no siguin aficionats al basquetbol puguin tenir una idea de quin és el rang i l'espai que "**Liga Virtual de Baloncesto**" pretén cobrir.

El rang d'aquest document va des de la màxima competició estatal (tant masculina com femenina) fins a la primera competició autonòmica (en aquest cas de Catalunya).

### COMPETICIONS MASCULINES

Pel que respecta a les competicions masculines, la primera competició de clubs en l'àmbit estatal és la Liga Endesa, una lliga privada gestionada per l'Asociación de Clubs de Baloncesto50 (ACB).

Per sota de la Liga Endesa estan les competicions de club d'àmbit estatal gestionades per la Federación Española de Baloncesto (FEB)<sup>51</sup>. Aquestes competicions, en ordre descendent d'importància són: Liga LEB52 Oro, Liga LEB Plata y Liga EBA<sup>53</sup>. A partir d'aquí ja apareixerien les competicions de caràcter autonòmic que en el cas de Catalunya seria la Lliga Catalana.

Aquestes 5 competicions serien les principals, 4 d'elles d'àmbit estatal i la última ja de d'àmbit autonòmic. A partir d'aquí ja estarien totes les altres competicions i lligues gestionades per les federacions autonòmiques. D'aquestes 5 competicions, les 2 primeres (Liga Endesa i LEB Oro es consideren professionals, mentre que les altres són considerades semiprofessionals o amateurs). Ara bé, en totes aquestes 5 competicions, en major o menor mesura, les bases de competició estableixen l'obligatorietat de recollir, publicar i facilitar dades estadístiques, tant als clubs rivals com als aficionats a través de les pàgines web de la Liga Endesa (<https://acb.com>) com de la FEB (<https://feb.es>) com del Federació Catalana de Basquetbol (<https://basquetcatala.cat>).

### COMPETICIONS FEMENINES

Pel que respecta a les competicions femenines, el nombre de competicions considerades professionals, semiprofessionals i amateurs és bastant més reduït. En el cas femení, la màxima competició de clubs és la Liga Femenina

---

50 <https://acb.com>

51 <https://feb.es>

52 LEB: Liga Española de Baloncesto

53 EBA: Equipos de Baloncesto Aficionados

(actualment Liga DIA), la següent és la Liga Femenina 2 i després ja apareixen les competicions gestionades per les federacions autonòmiques, que en el cas de Catalunya és la Lliga Catalana. Només la Liga DIA té la consideració de professional.

Igual que en el cas masculí, en aquestes competicions hi ha l'obligatorietat de recopilar les dades estadístiques d'una manera més o menys estandarditzada i homogènia.

#### **TAULA RESUM COMPARATIVA**

<b>Competicions masculines</b>	<b>Competicions femenines</b>
Liga Endesa	Liga DIA
Liga LEB Oro	Liga Femenina 2
Liga LEB Plata	
Liga EBA	
Lliga Catalana	Lliga Catalana

**Il·lustració 48: Resum de les principals competicions de bàsquet a l'Estat**

## B. Lliurables del projecte

Juntament amb aquest document de memòria també s'adjunta els següents documents:

- Codi font: dins de la carpeta SOURCE es troba un arxiu comprimit amb tot el codi de l'aplicació.
- Base de dades: dins de la carpeta BBDD hi ha l'arxiu de backup de la base de dades amb les dades inicials per engegar el projecte
- Arxius d'estadístiques: dins de la carpeta STATS hi ha diversos arxius amb les dades estadístiques dels jugadors dels primers equips (masculí i femení) del club Bàsquet Maresme Boet Mataró 3 Viles corresponents a les primeres jornades de la temporada 2020/21 (fins al 31/12/2020) de les competicions Lliga EBA (grup C-5) i Lliga Femenina 2 (Grup C)

Hi ha un arxiu per jornada, amb el nom stats\_jXX.xlsx, on XX és el número de la jornada. A més a més, l'arxiu stats\_j00.xlsx conté, en un únic arxiu, totes les dades de la resta d'arxius.

- Documentació: dins de la carpeta DOCS hi ha una guia d'instal·lació del projecte i un manual de funcionament bàsic de l'aplicació i del procediment a seguir per part de l'administrador per al correcte funcionament.