



An online multiplayer sports management game using Laravel

Final Degree Work
Bachelor's Degree in Video Game Design
and Development
Universitat Jaume I
July 2019, Valencia

Author: Alejandro López Balderas
Supervisor: Antonio Morales Escrig

ACKNOWLEDGEMENTS

My first words of thanks must be for Juan Carlos Amengual, surely, without his facility to see things and his support whenever I have needed it, I could not have planned my time to finally carry out this project.
Also, to Antonio Morales, who has always been there when I needed him for any setback, doubt or question.
And of course, to my family, you are everything.

ABSTRACT

The following document presents the final project of the Bachelor's Degree in Video Game Design and Development taught at the Jaume I University of Castellón. The main idea of the project is the development of a sports management video game for the web: BBCoach.

BBCoach is a 2D sports manager game developed in HTML, CSS, JavaScript and Laravel, a famous PHP language framework widely used in web programming.

CONTENTS

Contents

1. Introduction	
1.1. Work Motivation.....	6
1.2. Objectives.....	7
1.3. Environment.....	7
1.3.1. Tools of Development.....	8
1.3.1.1. HTML/CSS.....	8
1.3.1.2. JavaScript/Ajax.....	8
1.3.1.3. PHP/Laravel.....	8
1.3.1.4. MySQL/Heidi.....	8
1.3.1.5. XAMPP.....	9
1.3.2. Why Laravel?.....	9
1.4. Related Subjects.....	9
2. Planning	
2.1. Initial Planning.....	10
2.2. Final Planning.....	10
2.3. Resources evaluation.....	12
3. System analysis and design	
3.1. Overview of the game and mechanics.....	13
3.2. Characters.....	13
3.3. Requirement Analysis.....	13
3.3.1. Functional Requirements.....	13
3.3.2. Non-functional Requirements.....	14
3.4. System Design.....	14
3.4.1. Classes Diagram.....	15
3.4.2. Use case Diagram.....	17
3.5. Interface Design.....	17
4. Work Development and Results	
4.1. Work Development.....	22
4.1.1. Player creation and skill assignment.....	23
4.1.2. Tactics.....	23
4.1.3. Match simulator.....	24
4.1.4. Training.....	28
4.1.5. Calendar.....	29
4.2. Results, testing and validation.....	30
5. Conclusions and future work	
5.1. Conclusions.....	31
5.2. Future work.....	31

Bibliography

LIST OF FIGURES

Figure 1. Tools of development

Figure 2. Gantt diagram installation

Figure 3. Gantt diagram database plan

Figure 4. Gantt diagram about functionality tasks

Figure 5. Gantt diagram design

Figure 6. Price of hosting

Figure 7. Outline of a BBCoach season

Figure 8. Model code

Figure 9. Database schema

Figure 10. Use case diagram

Figure 11. FIFA MANAGER 14

Figure 12. Color pallete

Figure 13. Dashboard page

Figure 14. Calendar page

Figure 15. Roster page

Figure 16. Match page

Figure 17. Tactics page

Figure 18. Training page

Figure 19. Configuration part of .env file

Figure 20. Starting default Laravel Project

Figure 21. Minutes distributor

Figure 22. Team data

CHAPTER 1

INTRODUCTION

Many studies show that, in relation to the loving and affective world, there is always that person in the world that you will love. Be handsome, ugly, tall, short, skinny, fat, friendly, gruff... there will always be someone who you would like.

Well, in video games happens exactly the same, each of them has their own target, their own people to whom their personality and their way of living will make each of these games fit like a finger to them, and make them spend hours and hours in front of the screen.

In my case, from a very young age I like everything around management, either sports, economic, political or military, which made me interested in this kind of video games. Games like *Hattrick* [ExtraLives AB, 1997], *FIFA MANAGER* Saga [EA Sports, 2005-2013], *FarmVille* [Zynga, 2009], *Imperial Glory* [Pyro Studios, 2005] or *Pro Cycling Manager* Saga [Cyanide Studio, 2005-2020] marked my childhood and influenced my creativity and my personality, to the extent that I am currently working into the creation and management of ERP (Enterprise Resource Planning) programs for big companies.

Since I started the degree, I always had in mind to make some video game of this type, but naturally, I did not have the knowledge or experience necessary to carry it out. Now, thanks to my studies, I have learned to manage databases, to program in a web environment, to handle complex data structures or to design user interfaces, which has made me in the perfect situation to start a project for which I was destined.

BBCoach unites this great hobby described above, with my other great passion, basketball. We are talking about a sports management game for a basketball club in which both, the leagues and the players involved in them, will be fictitious. A sort of virtual world in which each coach will have his management space in which he can take charge, among other things, of the training sessions and the way of playing of his team, or the minute charge of his players.

1.1 WORK MOTIVATION

The reasons for carrying out this project are not only academic, in addition to what was mentioned in the previous point, I have carried out this project with the idea of commercializing it in the not too distant future, since there are very few games of this type with the features and design of BBCoach.

From International Basket Manager, Pro Basket Manager 2019 or FIFA Manager, until Pro Cycling Manager, F1 Manager or Online Soccer Manager... likely 90% of this kind of games are developed around real players and clubs, having to pay licenses to leagues or federations for the use of their personal data. My work is based on another perspective, an idea of random no-real international players, that creates a really hard personal relationship between the user, their team and their players, because none of them is being used by any other player in the world at the same moment.

1.2 OBJECTIVES

The main objectives of this project will be listed below, divided into technical and conceptual objectives.

Technical Objectives

- Run a project on a local environment for the development of a web project.
- Get an online user login and register system.
- Obtaining a rewarding user interaction system by introducing AJAX as the main programming tool.
- Create a modern and dynamic user interface, with the latest UX / UI design innovations.

Conceptual Objectives

- Manage to convey to the user a sense of reality, making the players, the tactics, the games or the training sessions being based on situations that can usually occur in basketball.
- Offer the player the possibility of controlling the game system of his team.
- Make the user to want to see his team grow, by creating a training system that will improve physical and mental qualities.
- Development of a multifactorial match simulator adapted to reality in terms of results and statistics.
- Guarantee the user a unique and personal gameplay, in which the team and the players belong exclusively to him.

1.3 ENVIRONMENT

1.3.1 TOOLS OF DEVELOPMENT

The main programs and languages (Figure 1) used during the development of the project are listed below, divided into types of tools:



Figure 1. Tools of development

- IDE: Sublime Text 3. [\[1\]](#)
- Art: Photoshop [\[2\]](#), Adobe Comp.
- Programming languages: HTML, CSS, Ajax, Javascript, JQuery, PHP (+Laravel Framework).
- Database Management System: MySQL.
- Web Server Solution Package: XAMPP.
- Database administrator: Heidi.
- Browsers: Google Chrome.

1.3.1.1. HTML / CSS [\[3\]](#)

HTML and CSS are the basic languages to make any website. On the one hand, HTML is the language that gives a visual structure to the web pages. Headings, paragraphs, images, inputs of text or buttons are some of the possible contents that can be added to a website. On the other hand, CSS is used over HTML classes, defining their style properties. They are different, but as a web developer, you need to work both at the same time if you want a clean job.

1.3.1.2. JAVASCRIPT / JQUERY / AJAX [\[4\]](#) [\[5\]](#) [\[6\]](#)

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. It enables interactive web pages and is an essential part of web applications. In this project, I mixed this language with his main multiplatform library, JQuery. It allows to simplify the way to interact with HTML documents, handle events or develop animations. In addition to all this, JQuery contributes to develop an interaction with AJAX, a web development technique to create interactive applications, which gives the user the ability to make changes on pages without reloading them, greatly improving the interactivity, speed and usability of websites.

1.3.1.3. PHP / LARAVEL [\[7\]](#) [\[8\]](#)

PHP is a widely-used open source scripting language that is especially suited for web development and can be embedded into HTML. In this project, I used one of the most famous and used frameworks based on PHP, Laravel.

Laravel is used to develop PHP in an elegant and simple way. It provides many positive functionalities:

- Blade: Template system to create dynamic views in Laravel, very useful for not reusing code, as a developer can use the same view for some pages. Besides, naming and accessing data in blade views is easier than ordinary PHP.
- Eloquent: Provides a beautiful and simple model system to interact with databases, which makes the code less stressful and easier to read.
- Routing: One of the most important concepts in web developing is the routing system, that will allow the user decide which part of the page to visit, depending on the part of web clicked, or the URL typed. Laravel provides a very useful, clean and dynamic service of routes, giving developer the possibility to build simple or grouped routes, based on permissions.
- Community and Documentation: It is known that Laravel has one of the best developer documentations in the framework of programming. Many people, including me, have learned Laravel based solely in these reports. In addition, its community is huge, allowing constant feedback and a faster and wider learning capacity.

1.3.1.4. MYSQL / HEIDI [\[9\]](#)

I have used MySQL for the management of the game database. MySQL is an open source relational database management system (RDBMS) with a client-server model, a software used to create and manage databases based on a relational model. Also, I needed a program to be able to view and manage these databases in a more comfortable way, and that's where Heidi comes. Heidi is a free and open-source administration tool for MySQL, that allows an easy and fast way to make

connections with MySQL servers, manage tables, rows and data from the same view or change the relational settings of our database.

1.3.1.5. XAMPP [\[10\]](#)

And here is the tool that handles everything and makes our website work. XAMPP is an acronym, its letters mean: X: Cross-platform; A: Apache; M: MariaDB (MySQL); P: PHP; P: PERL.

This free and open-source web server solution has allowed me to start a local Apache server with PHP and MySQL built in, to work locally on the project.

1.3.2 WHY LARAVEL?

I decided to use this framework first of all because of its popularity, as its used in many companies for projects with large amounts of data. This made me interested both personally and professionally. Moreover, as I wanted to develop my idea over AJAX to improve the user experience with the game, not having to wait for the page being reloaded, Blade templates gave me the possibility of reusing views of same page and be able to load them by this programming method more easily. Finally, the simple treatment of databases would be particularly helpful for me, because I had to make a lot of calls to project tables.

1.4 RELATED SUBJECTS

VJ1217 – Web game design and development. The concepts of MVC and development in JavaScript that I learned in the subject helped me.

VJ1208 – Programming II. The base of everything, basic programming concepts necessary.

VJ1215 – Algorithms and data structures. A very important part of my development as a programmer, I learned to develop complex and fast algorithms and to choose the correct data structures for it.

VJ1209 – 2D Design. Concepts of web layout and use of Photoshop.

VJ1220 – Databases. Another of the subjects that most helped me for the realization of the project. The creation and manipulation of databases is essential in my development.

CHAPTER 2

PLANNING

As all this kind of projects, I had an initial planning to develop it. In this section I'm going to explain and compare it with the final work, showing it with a Gantt Diagram.

2.1 INITIAL PLANNING

I designed the initial planning at the end of October (2019), and I did not know exactly all tasks that the project was going to require. Because of that I planned a general sketch based on the weights of each parts that I was going to have to develop.

- Installation of Laravel and configuration of Apache and MySQL - 3h
- Creation and implementation of the relational database - 60h
- Functionality of the game - 150h
- Creation of Design, images and icons - 50h
- Preparation of the project presentation - 7h
- Drafting of the project - 30h

2.2 FINAL PLANNING

At this point I display in tables from 1 until 4 a list of different General tasks divided in subtasks, and time spent on each one. Figures from 2 until 5 shows Gantt Diagrams about times of development.

Installation of Laravel and configuration of Apache and MySQL	Time
XAMPP installation (Apache, PHP and MySQL)	40 Minutes
Composer installation	10 Minutes
Laravel installation	10 Minutes
Configuration and installation of packages (JQuery, bootstrap, Fontawesome...) and Model relationship settings.	1 Hour
Creation of MySQL database and local server connection with Heidi.	1 Hour

Table 1. Installation and configuration planning

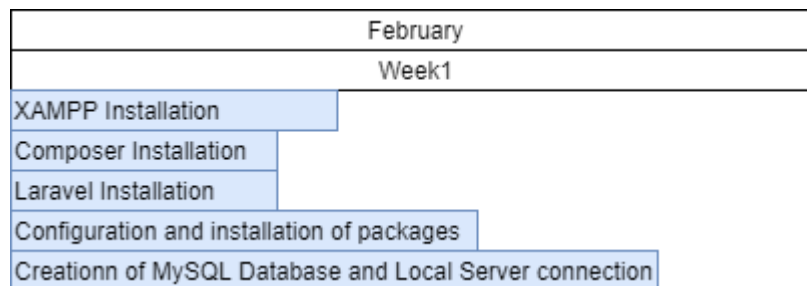


Figure 2. Gantt diagram Installation

Creation and implementation of the relational database	Time
Sketch of the initial state of the database (users, leagues, tactics, teams, players and matches)	14 hours
Implementation of initial state	10 hours
Sketch of the changes and additions around the initial state (player's nationality and positions, match events or players stats of each game)	10 hours
Implementation of middle state	8 hours
Sketch of the changes and additions around the middle (calendar, training, column values)	10 hours
Implementation of final state	8 hours

Table 2. Database creation planning

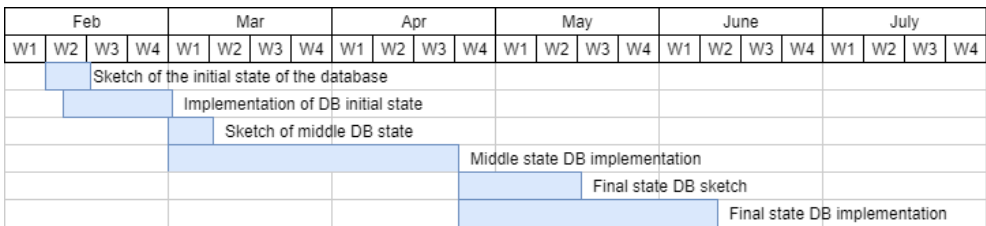


Figure 3. Gantt Diagram Database Plan

Functionality of the game	Time
Implementation of views (AJAX programming and relationship of routes with button or inputs, distribution of minutes in tactics or training)	40 hours
Implementation of match calendar	35 hours
Develop of trainings	40 hours
Games simulator	80 hours

Table 3. Functionality planning



Figure 4. Gantt Diagram about functionality tasks

Creation of Design, images and icons	Time
Design of views	30 hours
Layout of designs	30 hours

Table 4. Design planning

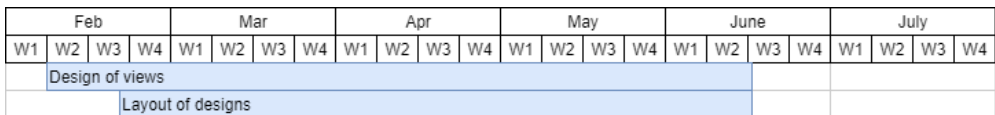


Figure 5. Gantt Diagram Design

2.3 RESOURCES EVALUATION

PERSONAL BUDGET

The cost of the project (Table 5) is divided by departments, and its not included the time for learning. The following table shows an estimate of costs:

Department	Field	Cost/time	Hours	Cost
Design	Art	10€/h	60	600€
Programming	Program	15€/h	200	3000€
Documentation	Other	10€/h	40	400€
			Total:	4000 €

Table 5. Personal budget

The information about salary is based on average estimation in job offers found on Stratos [11] website.

SOFTWARE LICENSES BUDGET

The price of the tools user for this project are cheap or even free to use. The cost for each license can be found on its respective website.

Licenses and cost:

- HTML, CSS and JavaScript librarys (free)
- JQuery library (free)
- Photoshop CSS6 (10€/month)
- Sublime Text (free)
- Heidi SQL (free)
- Laravel Framework (free)

HOSTING

The Project has been carried out locally, but if it had to be commercialized it would be necessary to hire some type of hosting that would allow us to have it online. In addition, a non-existent domain would have to be hired to give it body and name. In a common place of accommodation (Figure 6), we find that, for a demanding web project like ours, with domain included, the price would amount to € 6.99 / month.

Alojamiento web con todo lo que necesitas

HOSTING HOSTING ASP.NET

Plan	Duration	Price	Features
Hosting Standard	3 meses	GRATIS	La solución para sitios web profesionales
Hosting Unlimited	6 meses	GRATIS	Aloja proyectos web exigentes
Hosting Unlimited Pro	3 meses	GRATIS	Alto rendimiento para tus proyectos web

dominios incluidos gratis (.es, .com, .com.es, .net, .org, .org.es, .biz, .info, .name, .eu, .cat) **

1 2 3

Figure 6. Price of hosting

FINAL PRICE

The final amount of money up to 4000€ + (5 months * 10 €/month) = 4050 € to which should be added the cost of the hosting monthly fees.

SYSTEM ANALISIS AND DESIGN

3.1 OVERVIEW OF THE GAME AND MECHANICS

BBCoach will be a management simulation videogame that will allow the player to become fully involved on the decision-making around a basketball team. This is only a small part of what the game will offer in its final version, since it will comprise, in addition to the developed sports block, an economic block, a social block, and several other mechanics.

This project works over the sports block, in which the final objective is that the user's team is at the highest part of the league system created, and this will be achieved by having a weekly control about training, tactics and matches, which are explained in more depth in the section 4.1 of the project.

3.2 CHARACTERS

Manager

Main role managed by the user, who will be the owner of a club, over which he will have full control. Its function, as explained before, will be to make your team grow as much as possible in the sporting aspect.

Player

Will be a simulated role, which will be created with pseudo-random name, surnames, skills and social characteristics.

Their skills will influence the course of the matches, each one of them will have 100 points of rank, with 0 being number with the least value, and 100 being the highest.

In one hand, the skills that can be improved in trainings will be: Defense (individual and collective), free throw, 2pt shot, 3pt shot, game view, physical condition, jump, speed, strength, rebound, pass, footwork and game direction. On the other, there will be skills that will advance according to certain factors, such experience or leadership.

3.3 REQUIREMENT ANALYSIS

The main application requirements, both functional and non-functional (Tables 6 and 7), will be listed below. At the end of the document, we will analyze which of these requirements have been completed and which have not.

3.3.1 FUNCTIONAL REQUIREMENTS

NUMBER	REQUIREMENT	DESCRIPTION
FR1	REGISTER	The game will allow a user to register on system with certain access codes.
FR2	TEAM	The game will grant the user a specific team and players.
FR3	LEAGUE	The game will create an automatic league system.
FR4	CALENDAR	The game will have a match schedule.
FR7	TACTICS	The game will allow the player to establish how his team should play.
FR8	PLAYERS	The game will have a section to view players and their skills.
FR9	MATCHES	The game will allow the user to play matches against other players.

FR10	TRAINING	The game will give the user the possibility to improve the skills of his players through training.
------	----------	--

Table 6. Functional Requirements

3.3.2 NON-FUNCTIONAL REQUIREMENTS

NUMERO	REQUERIMIENTO	DESCRIPCIÓN
NFR1	UI DESIGN	The game features a graphic design that attracts the player.
NFR2	UX DESIGN	The game has a versatile and easy to use user interface.
NFR3	SECURITY	Access to the game will be restricted under encrypted passwords and defined users.
NFR4	MULTIPLATFORM	The game must work on different types of operating systems and hardware platforms.
NFR5	EFFICIENCY	The game must support the handling of a large amount of information during its execution.
NFR6	TRUTHFULNESS	All aspects of the game must be real within the framework of Basketball.

Table 7. Non-functional requirements

3.4 SYSTEM DESIGN

The initial idea has always been to leave space between the player and the game, that is, to add patience as a key point of the design. I didn't want to make another game in which there were matches every day, because it would not be special or close, the leagues would last a month and a half at most and the player would play down. So, I came to the conclusion that one game or training a week would make the player have to look forward to that moment, and go over and over again during the week if everything is correct.

BBCoach is based on seasons and weeks. Each season (Figure 7) consist of 15 weeks, 14 for the league days, and 1 bridge week, to establish promotions and descents and generate calendars for the new leagues. In each of these weeks a date is established for training sessions and for games, in our initial case are Wednesdays at 3:00 p.m. and Saturdays at 5:00 p.m., with which 14 games will be played and training will be done 15 times each season.



Figure 7. Outline of a BBCoach season

During each of the weeks, the player will have to worry about establish a training session, plan the tactics for his next game, and finally visualize it.

Thus, the strong point of the game is based on the game, a simulation that I have tried to be as real and complete as possible, and which we will discuss in detail in the work development section.

Next, the main technical characteristics and concepts of the game will be described through different diagrams.

3.4.1 CLASS DIAGRAM

As previously mentioned, when installing, Laravel offers us an MVC working model (Model View Controller), a software architecture pattern that separates logic, view and events, and which has very positive basic concepts for development of a project, such as code reuse or concept separation, with which we gain structuring, execution speed and information search speed.

On this basis, the framework we are working on stores the main development objects in the models part, each one related to a table in our database. So, to access a row of any of the tables, you only have to name the class created in the model and make a query about it with the nomenclature offered by the environment.

In summary, each class in Laravel is related to a Model which in turn is related to a table, which means that the class diagram will be identical to our relational database, detailed in Figure 9 on the next page.

As an example (Figure 8), the relationship between the model-class (*PartidoEvento*) and the database table ('partido_eventos') is shown in detail below. We can see then that both the table and the model function to relate matches, events and teams.

```
<?php
class PartidoEvento extends BaseModel
{
    protected $table='partido_eventos';
    public $timestamps = true;

    public function Partido()
    {
        return $this->belongsTo('Partido', 'partido_id');
    }

    public function Evento()
    {
        return $this->belongsTo('Evento', 'evento_id');
    }

    public function Equipo()
    {
        return $this->belongsTo('Equipo', 'equipo_id');
    }
}
```

Figure 8. Model code

With the lane-code `$evento_partido = PartidoEvento::where('partido_id','=,1)->get();` we would access all the events of the match that had 'id = 1,'

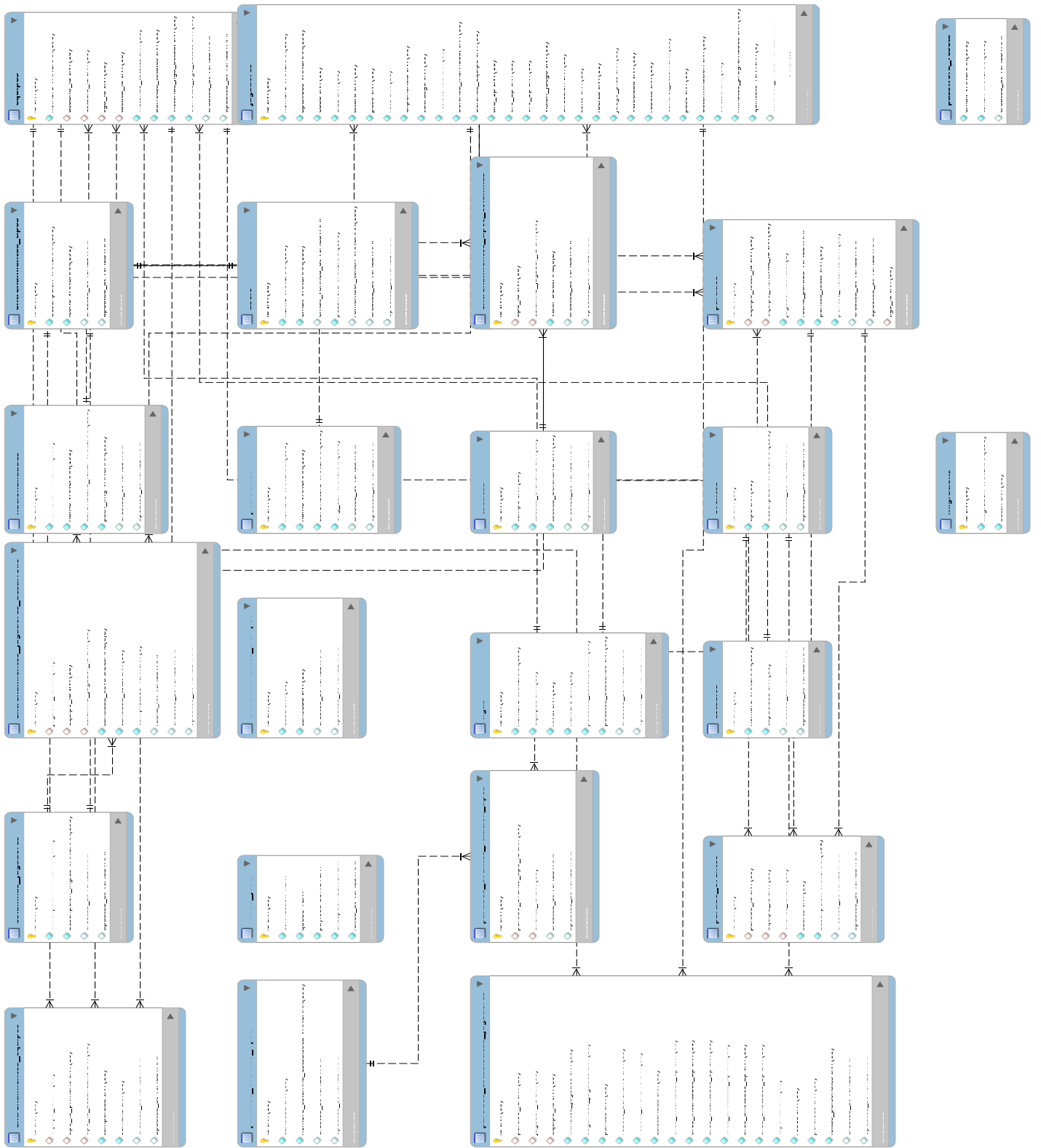


Figure 9. Database Schema

3.4.2 USE CASE DIAGRAM

The use case diagram (Figure 10) is described below, in which we find the two main actors in the development of the game, the user and the system. The system is included as an actor because it is programmed to perform certain functions by itself, at certain specific times. For example, when the time comes for a game, the system will know that the game has to be played, and the same will happen with training. Currently in this version of the project there are buttons to simulate all these situations.

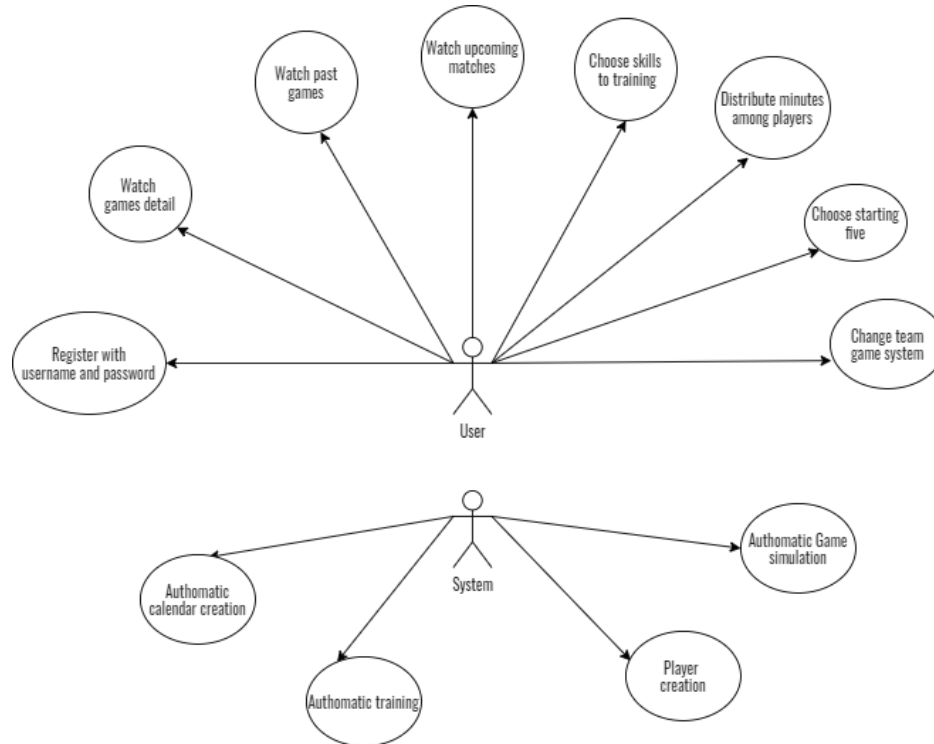


Figure 10. Use case Diagram

3.5 INTERFACE DESIGN

As I mentioned in the introduction, I always loved management games, and the one that most impressed me with its design, both visually and interactively, was the FIFA MANAGER (Figure 11), with some great dashboards, in which practically does not miss any detail. I wanted my interface to have those two characteristics, to please the view of the player, and to offer him on a screen all the content about what he was seeing.

But my game was not going to be an executable, it was going to work on a web page, so I had to adapt the menu and container layouts to web-oriented management layouts. I was very inspired by following different Instagram accounts (@uxuistack, @ui_lab or @farzan90.design, among others), and so I began to design my proposal.

I wanted bright colors and a black background, very fashionable in management programs today. Finally (Figure 12), I mixed the classic black and white with a palette of yellow, blue and red that I found online.

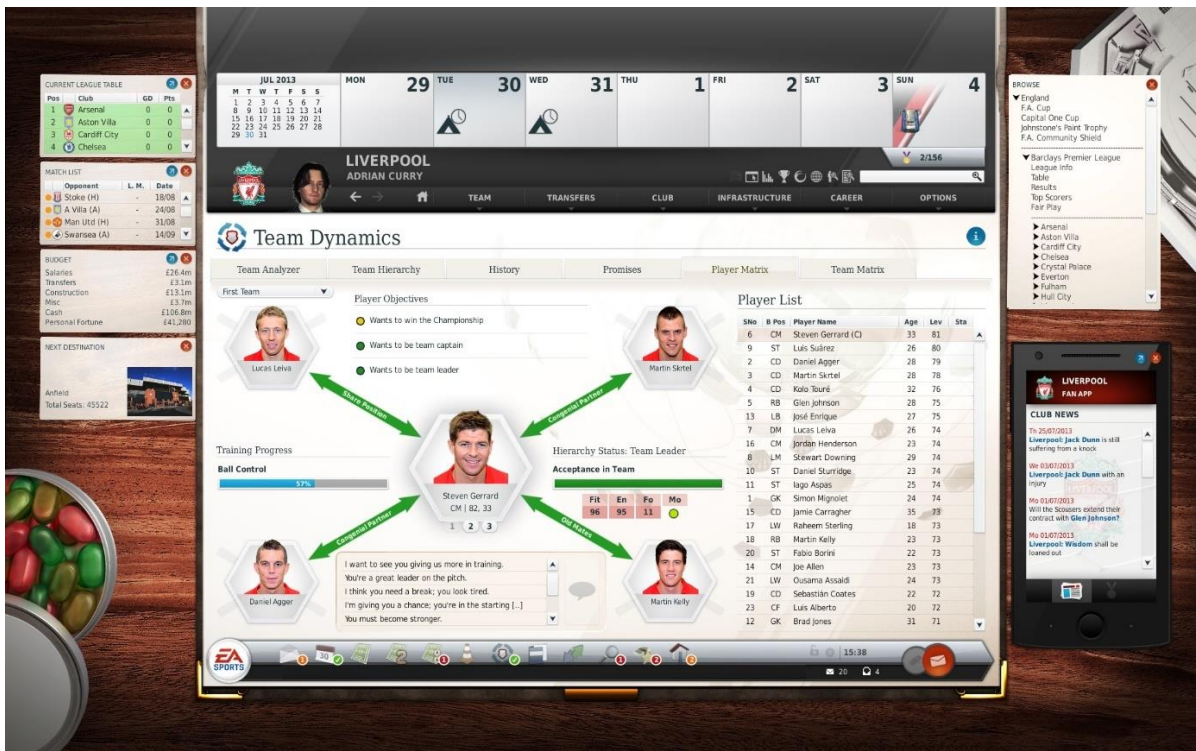


Figure 11. FIFA MANAGER 14

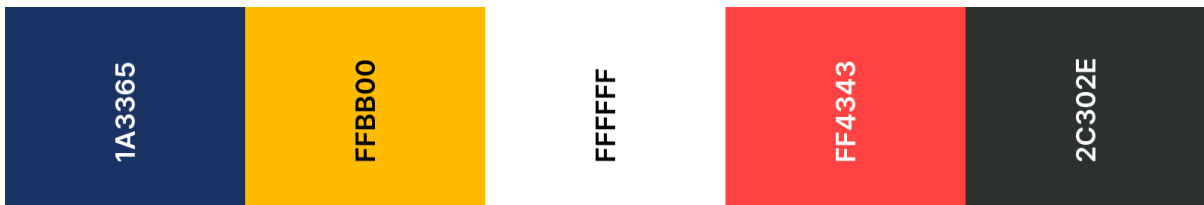


Figure 12. Color Palette

The design of the game is based on a side menu through which to navigate between screens, having a control panel as the main page, in which the user can have an overview of the development of his team. The game screens are shown below, with a brief description of each one.

HOME / DASHBOARD

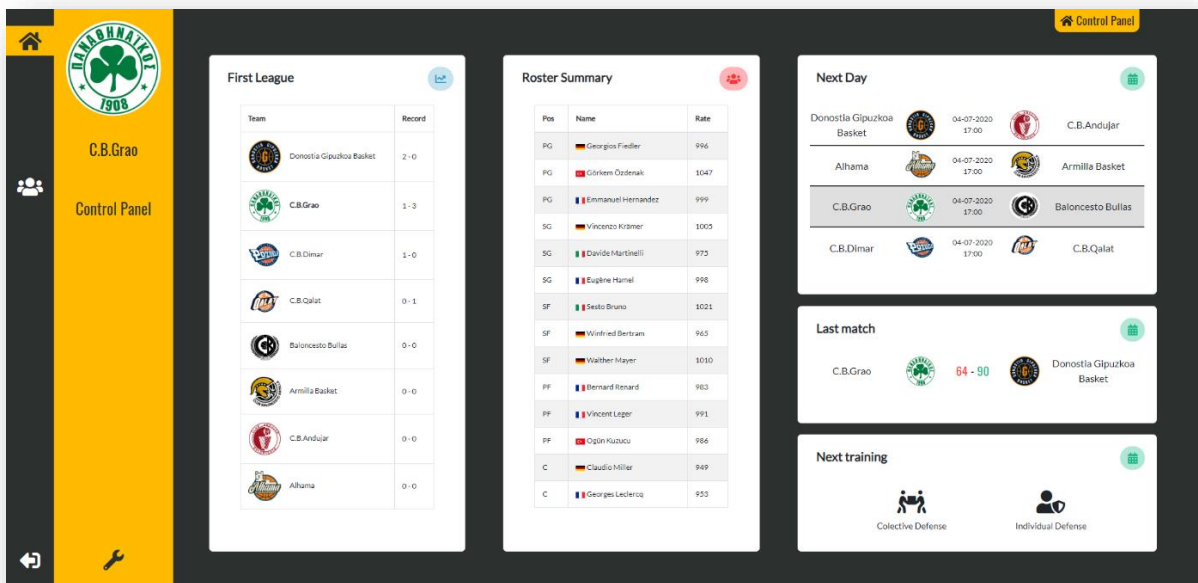


Figure 13. Dashboard page

Lace is the main concept that I have used both in this and the rest of the pages for the player's good first impression. In this case (Figure 13), all the containers are equal in height and, except for the third, in width. This gives the user a little pleasure to see that everything fits, that everything is aligned, and that the distances between parts with the same characteristics are equal, it is a very important concept in the world of web layout. Another important part is the identification of the current screen, indicated in the upper right on each of the pages.

TEAM / CALENDAR

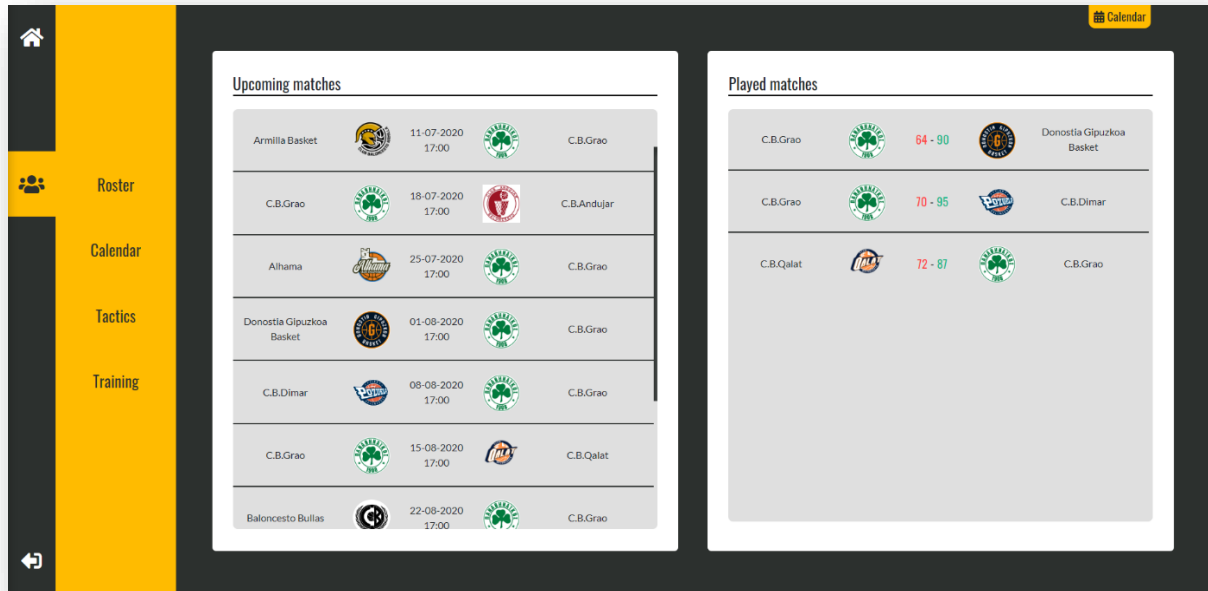


Figure 14. Calendar Page

This page (Figure 14) has sought to highlight the listings above the standard container type with a light gray background. In addition, custom CSS scrolls have been designed with colors belonging to the project, such as the one seen in the list of upcoming matches.

TEAM / ROSTER

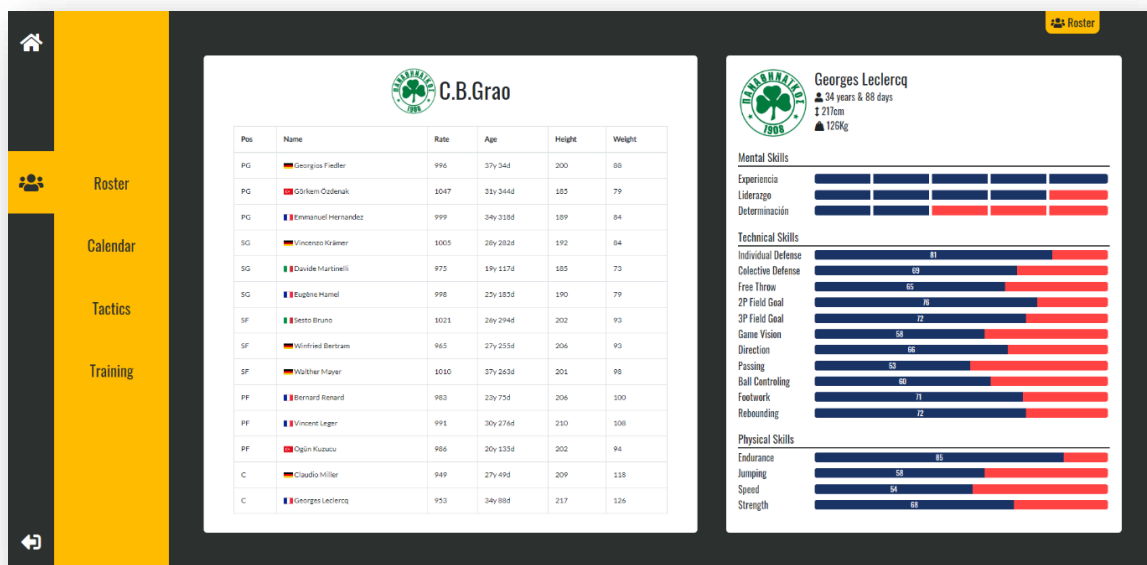


Figure 15. Roster Page

On this screen (Figure 15), some concepts that gratify the user experience are mixed. The AJAX programming technique allows us, by clicking on any of the rows of our players table, the container on the right loads the information of that player, changing the progress bars and the personal information of the player in less than 0.1 seconds. This, mixed with the contrast of colors and the use of icons, create an immersive page, versatile and easy to use.

MATCH (Figure 16)

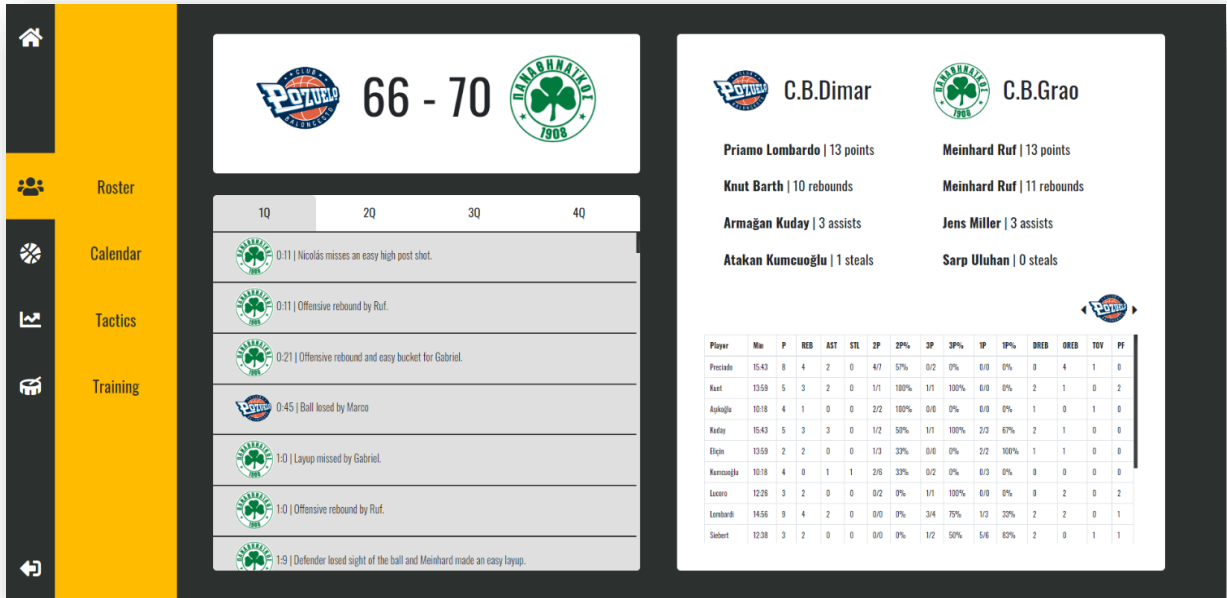


Figure 16. Match Page

TEAM / TACTICS

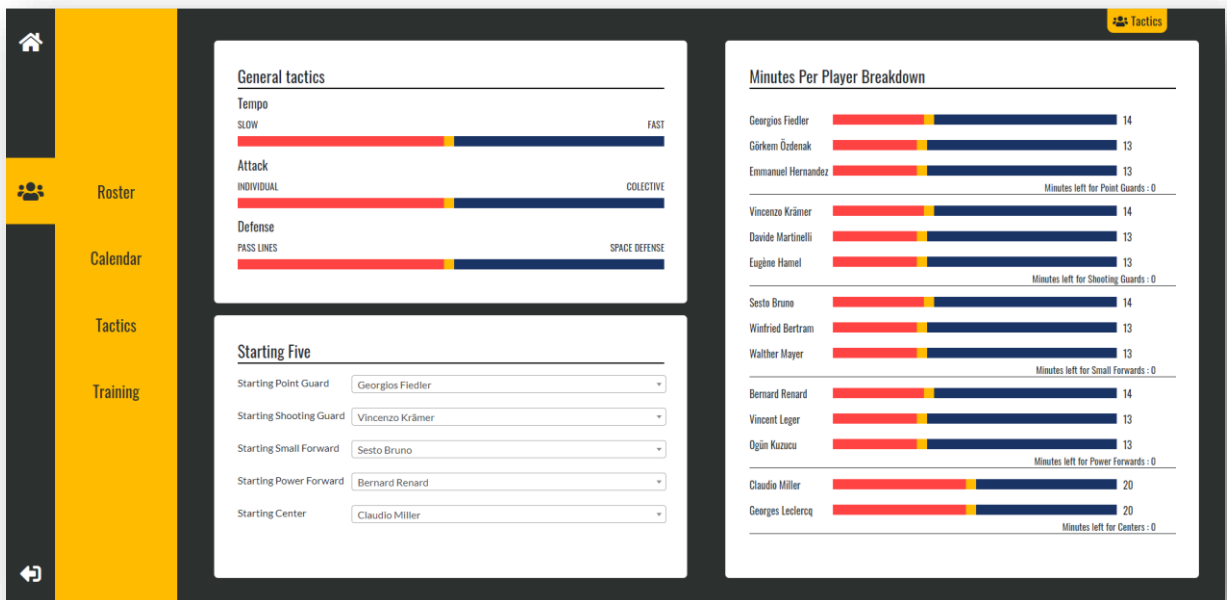


Figure 17. Tactics Page

Also custom-designed, we highlight on this page (Figure 17) the value markers, which save their values when you drop after also dragging through AJAX, which allows the player to change values of their tactics

or distribute the minutes of their players at a high speed, greater than if the page was reloaded each time.

TEAM / TRAINING

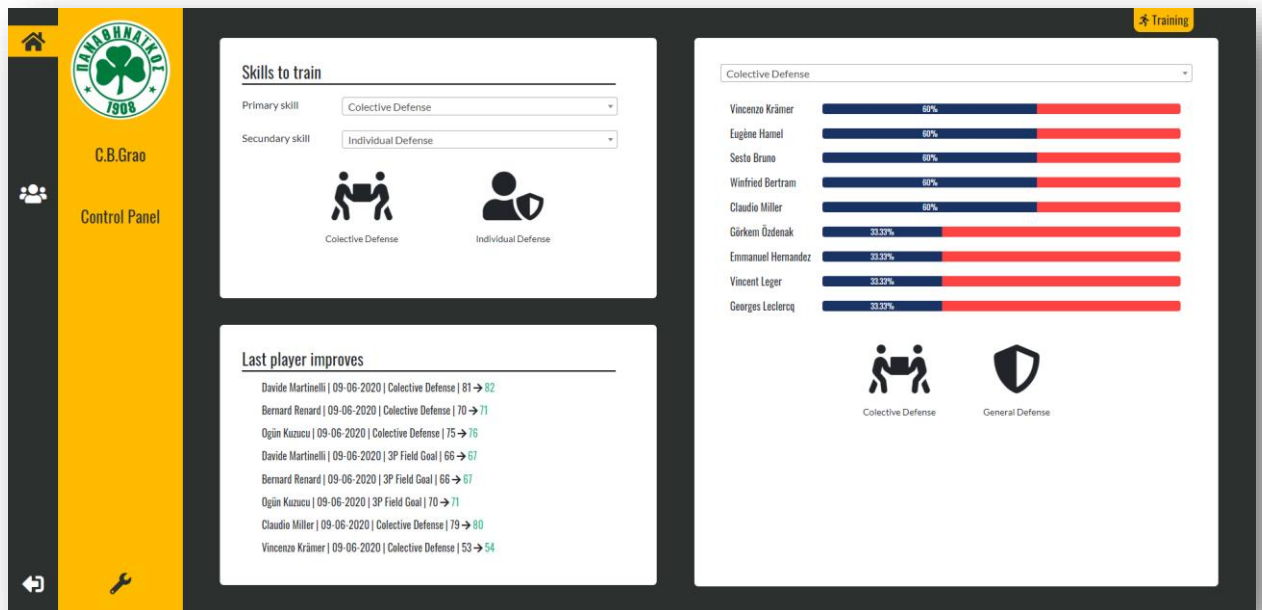


Figure 18. Training Page

On this page (Figure 18) I wanted to play with the relationship between each workout and an icon. It is much more visual to be able to see the icon and know what you are training than to read it every time. Also, I have marked the improvements in green in the lower left container to give a positive reinforcement to the training.

WORK DEVELOPMENT AND RESULTS

4.1 WORK DEVELOPMENT

4.1.1. INITIAL STATE

The project started with the installation and creation of a Laravel project within our local server through Composer, an installer of PHP packages, among which is the framework that has been used.

```
composer create-project --prefer-dist laravel / laravel blog
```

This command, executed on the folder chosen for our development, created an MVC structure (Model View Controller) to work on, which we will explain after.

Later, a MySQL database was created with Heidi and her credentials were configured.

To finish, before I could run first test, I had to edit `.env` (Figure 19) file (simple configuration) created in the project, to connect with the local server (`APP_URL`) and with the created database (`DB_CONNECTION`, `DB_HOST`, `DB_PORT`, `DB_DATABASE`, `DB_USERNAME`, `DB_PASSWORD`).

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:zZ7htqKYVG0ASVg3TCOJ7PbBAjEKCD+14/QaRdfaYkM=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=proyectoFinal
DB_USERNAME=root
DB_PASSWORD=
```

Figure 19. Configuration part of `.env` file

Once this is done, accessing the IP address of our local server from the browser, you will find the views and functionalities that Laravel brings by default (Figure 20). And here the dance begins.

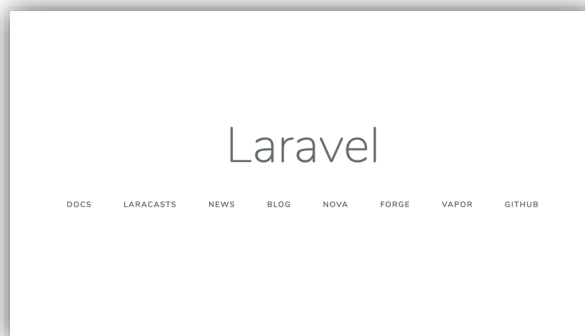


Figure 20. Starting default Laravel project

4.1.1. PLAYER CREATION AND SKILL ASIGNMENT

My idea from the beginning was to create a special bond between the user and his players, that he felt them own and real. To start, a random nationality is assigned to the player, within a list of 5 (Spanish, French, Italian, Turkish and German) and depending on it, applying this nationality to a PHP library for creating random people called faker (<https://github.com/fzaninotto/Faker>), the 'Player' object is created, and different attributes are randomly created over it.

Thus, within the randomness of its creation, the skills and physical attributes of the players are gradually assigned, depending on each other, to make it as transparent and credible as possible.

Next, I go on to detail the order developed when assigning physical features or abilities and some examples.

The first thing to be assigned as I have said before is the nationality, from there the name and surname, in addition to the age, which is nearly (70%), thinking of a real professional basketball team, to a value between 22 and 32 years old, and to a lesser extent youth and maturity (20% from 18 to 21 years old and 10% from 33 to 37 years old).

Then, position is given to player, based on his height, weight and abilities. To finish, experience, leadership and determination are generated, where age has a significant weight in the first two, being high for players over 30 years old, and normal until 26 years old. Younger players will have to play many games in order to increase these social skills. The relationship between the different positions and the skills in which they excel with respect to the rest is detailed below (Table 8).

POSITION	MAIN SKILL
Point Guard	1-point shot, game vision, direction, ball control.
Small Guard	1-point shot, 3-point shot, speed.
Small Forward	3-point shot, jump.
Power Forward	Rebound, strength, footwork, jump.
Center	Rebound, strength, footwork.

Table 8. Main skills by position

4.1.2. TACTICS

This is the section that determines within the team how, who and how much they play. We can divide it into 3 parts: General tactics, starting-five and minutes distribution.

General tactics control the three main aspects of a basketball game, the rhythm of the team, the type of attack and the type of defense. Each of them influences a circumstantial part of the development of the match, we will comment it on in the next point.

The starting-five is done by position, and is usually made up of the best players on your team, who will get you started on the best match.

Finally, we will explain in more detail the development in the project of the minute distributor (Figure 21), a fundamental part of the rotation of the players in the match.

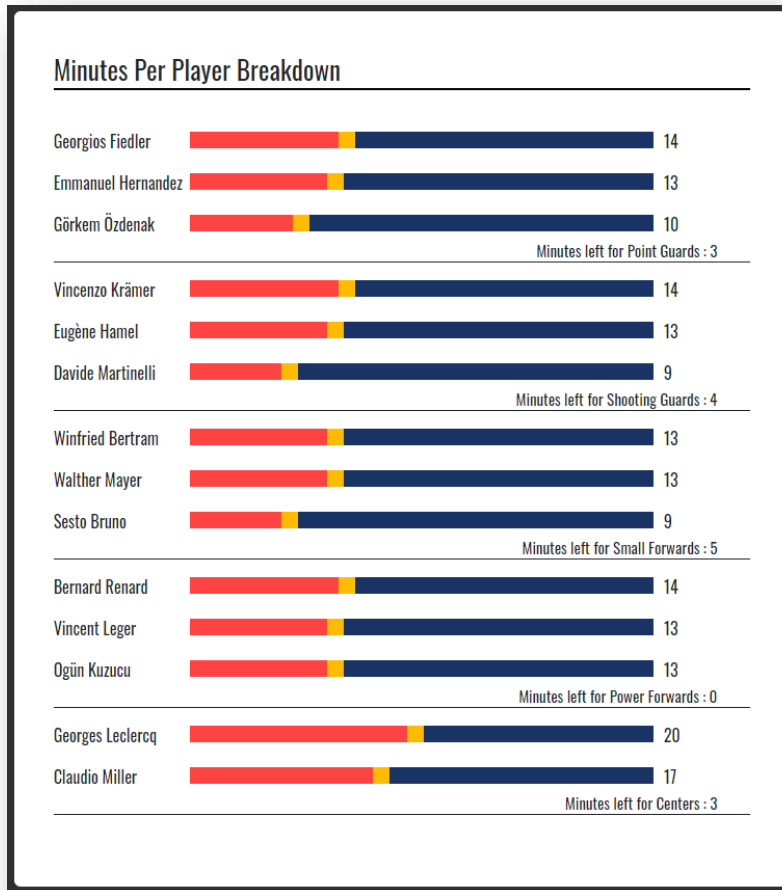


Figure 21. Minutes distributor

The only restriction in game is that each player has to play in his position during the match, and based on it, a tool (Figure 15) that distributes 40 minutes among the players of the same location has been developed. A player's minute marker stops at the maximum number of minutes he can play, depending on the time allotted to his other teammates on the field. In addition, for each of these positions, the time remaining to be allocated will be displayed.

4.1.3 MATCH SIMULATOR

This is the basis on which much of the project is based, the development that I'm most proud of.

To develop this part, I have created a set of functions that conceptually could be summarized in: The clock, the events, and the central data structure of the game.

- **The clock or game time.** 10 minutes divided into four quarters, 40 in total. From this standard, the events continue, until the time of each quarter and the game runs out. When an event ends, the attacking and defending teams may or may not exchange, depending on the type of event itself.
- **Events.** Each of the events that take place about the game is made up of a specific time and type.
 - o Event time: random element that will depend on the game rhythm chosen by the attacking team. Attacks at normal speed always have a 50% percentage to occur, and the other 50% are divided into fast and slow attacks. Thus, if the tempo tactic is somewhere in between, there will be a 25% chance of both a fast attack and a slow attack. On the contrary, if the tempo tactic is at 75%, there will be a 37.5%

probability that the attack will be fast, and 12.5% that it will be slow. The times of each type of attack have been chosen based on real basketball situations.

- Fast attack: 7 to 10 seconds.
 - Normal attack: 10 to 16 seconds.
 - Slow attack: from 16 to 24 seconds.
- Types of events: In addition to a certain duration, an event has a certain action, and here many factors, both individual and collective, compete to win the raffle. Next table (Table 9) summarizes the types of events, and the factors that influence each of them.

TYPE OF EVENT	FACTORS THAT INFLUENCE
Attack 2Pointer	2-point*3, jump, ball control, endurance*2
Attack 3Pointer	3-point*3 (or *4 if skill is higher than 80), jump, ball control, endurance
Defense 2Pointer	Individual Defense*3, jump, Team Defense, endurance*2
Defense 3Pointer	Individual Defense*3, jump, Team Defense, endurance*2
Turnover	Ball control, Pass, Experience, Individual Defense
Assist	Random, but higher percentage on Guards and Small Guards.
Steal	Ball control, Pass, Experience, Individual Defense
First Free Throw in	1-point
Second Free Throw in	1-point
Third Free Throw in	1-point
First Free Throw out	1-point
Second Free Throw out	1-point
Third Free Throw out	1-point
Offensive Foul	Random, 10% percentage inside fouls.
Personal Foul	Individual Defense, Experience
Out of Bounds	Ball control, Pass, Experience, Individual Defense
Substitution	Minutes
Defensive Rebound	Random, but higher percentage on Centers and Power Forwards.
Offensive Rebound	Random, but higher percentage on Centers and Power Forwards.
Time-out	5% Random.

Table 9. Relationship of events and factors

To finish, each one of these events, as they are carried out, keep saving and updating different variables located in a central data structure, which will be explained in the next point. We can see just below, as an example, next table with the statistical values that are updated for each event.

TYPE OF EVENT	UPDATED
Attack 2Pointer	2-point made, 2-point attempts
Attack 3Pointer	3-point made, 3-point attempts
Defense 2Pointer	2-point attempts
Defense 3Pointer	3-point attempts
Turnover	Turnovers
Assist	Assists
Steal	Steals
First Free Throw in	1-point made, 1-point attempts
Second Free Throw in	1-point made, 1-point attempts
Third Free Throw in	1-point made, 1-point attempts
First Free Throw out	1-point attempts
Second Free Throw out	1-point attempts

Third Free Throw out	1-point attempts
Offensive Foul	Fouls
Personal Foul	Fouls
Out of Bounds	Turnovers
Substitution	
Defensive Rebound	Defensive Rebound
Offensive Rebound	Offensive Rebound
Time-out	

Table 10. Relationship about type of event and variable updated.

Events or functions to highlight in terms of development:

- Substitution. One of the parts that has made me think the most about the whole project. Finally, I decided to assign minutes to each player per quarter, and following an event that could generate a substitution, check if any of the players should be substituted or not, since they have exceeded their assigned minutes per room. In addition, logically, a player must enter who has not yet exceeded his minutes per quarter.
 - Free Kick. Events generated only after shooting actions, or if a team has exceeded 4 fouls in a quarter. 2 or 3 events are generated, depending on whether the fault is 2 or 3 points.
 - Change of possession. It does not correspond to any event, but it is key to the development of the game, since apart from the baskets themselves scored, the rebound events, thefts or losses also make the attacking team to defend and vice versa.
- **Central data structure:** The game is designed for many users to interact with it at the same time, so a simulator based on constant calls to a database was not viable from the point of view of speed and memory, since it would collapse and run slower. Therefore, the idea has been to keep a data structure internally updated that works like a dictionary (set of pairs with key and value). JSON has been used to transport this data between the model and the controller, since it can be easily manipulated and transported with its encoding and decoding in a character string.

To begin with, this data structure, once created, has an initialization phase, in which the pertinent calls are made to the database to download the values that will later be used. Subsequently, the data stored in it is used, in addition to keeping them updated. And finally, the details that we want to store from the match (mostly statistics), are sent back to the database using a load function.

Thanks to all this process, a greater speed of code execution and an optimization of resources of our database are achieved.

Here are some images of the type of structure used, and the parts of it.

Basic data of the two teams that are going to play the match (Figure 22).

```
$array_equipo_local = array(
    "id" => $this->EquipoLocal->id,
    "nombre" => $this->EquipoLocal->nombre,
    "tiempos_muertos_1" => 0,
    "tiempos_muertos_2" => 0,
    "puntos" => 0,
    "faltas_cuarto_1" => 0,
    "faltas_cuarto_2" => 0,
    "faltas_cuarto_3" => 0,
    "faltas_cuarto_4" => 0,
);
$this->array_partido["datos_equipo"][$this->EquipoLocal->id][] = $array_equipo_local;

$array_equipo_visitante = array(
    "id" => $this->EquipoVisitante->id,
    "nombre" => $this->EquipoVisitante->nombre,
    "tiempos_muertos_1" => 0,
    "tiempos_muertos_2" => 0,
    "puntos" => 0,
    "faltas_cuarto_1" => 0,
    "faltas_cuarto_2" => 0,
    "faltas_cuarto_3" => 0,
    "faltas_cuarto_4" => 0,
);
$this->array_partido["datos_equipo"][$this->EquipoVisitante->id][] = $array_equipo_visitante;
```

Figure 22. Team data

Data of the players of each team (Figure 23).

```
foreach($this->EquipoLocal->jugadores() as $jugador){
    $array_jugador = array(
        "id" => $jugador->id,
        "nombre" => $jugador->nombre,
        "apellidos" => $jugador->apellidos,
        "edad" => $jugador->edad,
        "dias" => $jugador->dias,
        "altura" => $jugador->altura,
        "peso" => $jugador->peso,
        "rate" => $jugador->rate,
        "experiencia" => $jugador->experiencia,
        "liderazgo" => $jugador->liderazgo,
        "defensa_individual" => $jugador->defensa_individual,
        "defensa_equipo" => $jugador->defensa_equipo,
        "resistencia" => $jugador->resistencia,
        "tiro_1p" => $jugador->tiro_1p,
        "tiro_2p" => $jugador->tiro_2p,
        "tiro_3p" => $jugador->tiro_3p,
        "vision_juego" => $jugador->vision_juego,
        "direccion" => $jugador->direccion,
        "salto" => $jugador->salto,
        "fuerza" => $jugador->fuerza,
        "juego_pies" => $jugador->juego_pies,
        "velocidad" => $jugador->velocidad,
        "rebote" => $jugador->rebote,
        "control_balon" => $jugador->control_balon,
        "pase" => $jugador->pase,
        "determinacion" => $jugador->determinacion,
        "nacionalidad" => $jugador->nacionalidad->name,
        "posicion_id" => $jugador->posicion_id,
        "posicion" => $jugador->Posicion->eng_abr,
        "quinteto" => $jugador->quinteto,
        "minutos" => $jugador->minutos_programados,
        "resistencia" => $jugador->resistencia,
        "resistencia_actual" => $jugador->resistencia,
        "minutos_jugar_1" => 0,
        "minutos_jugar_2" => 0,
        "minutos_jugar_3" => 0,
        "minutos_jugar_4" => 0,
        "minutos_jugados_1" => 0,
        "minutos_jugados_2" => 0,
        "minutos_jugados_3" => 0,
        "minutos_jugados_4" => 0,
        "minutos_jugados" => 0,
        "puntos" => 0,
        "rebotes_ofensivos" => 0,
        "rebotes_defensivos" => 0,
        "asistencias" => 0,
        "tiros_intentados_1p" => 0,
        "tiros_intentados_2p" => 0,
        "tiros_intentados_3p" => 0,
        "tiros_annotados_1p" => 0,
        "tiros_annotados_2p" => 0,
        "tiros_annotados_3p" => 0,
        "tapones" => 0,
        "robos" => 0,
        "perdidas" => 0,
        "faltas_personales" => 0,
    );
    $this->array_partido["equipo"][$this->EquipoLocal->id]["jugadores"][$jugador->id][] = $array_jugador;
}
```

Figure 23. Player data

4.1.4. TRAINING

The training runs weekly, based on how old the player is at the time he receives the first training for a particular skill. As the game always adapts to reality, the younger a player is, the less time it takes to learn a new skill, below the table 11 with the age relationships and weeks of training.

AGE GROUP	WEEKS
18 - 19	2
20 - 24	4
25 - 29	5
30 - 35	6
36	7

Table 11. Age groups and weeks to improve an skill.

The primary skill that is trained will add one week to work, and the secondary skill, half. So, if I want to improve the three-point shot of a 29-year-old, he will need, for example, 4 primary and 2 secondary workouts to increase the value.

Once you match or exceed the required work weeks, the training will be completed, and you will be waiting for a new training of that skill.

The types of training and the skills each improves are detailed below (Table 12).

TRAINING TYPE	SKILLS IMPROVED
Individual Defense	Individual Defense 100%
Colective Defense	Colective Defense 100%
General Defense	Individual Defense 50%, Colective Defense 50%
Ball control	Ball Controlling 100%
Free-Throw	Free Throw 100%
Mid Range Shots	2P Field Goal 100%
Three-Pointers	3P Field Goal 100%
General Shooting	Free Throw 33%, 2P Field Goal 33%, 3P Field Goal 33%
Pass and game concepts	Passing 50%, Game Vision 50%
Low-post movements	Footwork 50%, Jumping 50%
Physical skills	Jumping 33%, Speed 33%, Strength 33%
Rebound	Rebounding 100%

Table 12. Trainings and skills improved

4.1.5. CALENDAR

To finish this section, I want to explain in detail the development of calendar creation.

Once a league is completed, automatically, on the scheduled dates, the calendar with round trip matches will be generated.

To make this assignment I have used the all-against-all system, in which we find 7 rounds and 4 matches per round, as they are leagues of 8 teams, so, for example, we would start from a table that begins writing successively and until the end the numbers of the teams up to the highest odd, that is, from 1 to 7.

ROUND	MATCH 1	MATCH 2	MATCH 3	MATCH 4
1st	1	2	3	4
2nd	5	6	7	1
3rd	2	3	4	5
4th	6	7	1	2
5th	3	4	5	6
6th	7	1	2	3
7th	4	5	6	7

Table 13. First step of calendar generator

The last number, we write it alternately from right to left of the number of the first column.

ROUND	MATCH 1	MATCH 2	MATCH 3	MATCH 4
1st	1-8	2	3	4
2nd	8-5	6	7	1
3rd	2-8	3	4	5
4th	8-6	7	1	2
5th	3-8	4	5	6
6th	8-7	1	2	3
7th	4-8	5	6	7

Table 14. Second step of calendar generator

Finally, row and column pairings are completed by writing down the missing numbers starting with the highest odd number.

ROUND	MATCH 1	MATCH 2	MATCH 3	MATCH 4
1st	1-8	2-7	3-6	4-5
2nd	8-5	6-4	7-3	1-2
3rd	2-8	3-1	4-7	5-6
4th	8-6	7-5	1-4	2-3
5th	3-8	4-2	5-1	6-7
6th	8-7	1-6	2-5	3-4
7th	4-8	5-3	6-2	7-1

Table 15. Final step for calendar generator

This system will always be used, for all leagues, since they will always be 8 teams.

4.2 RESULTS, TESTING AND VALIDATION

The results of the project could not be more satisfactory, since all the requirements and objectives proposed at the beginning of the project have finally been met. The basic objectives that have been met:

- A relational database has been created. Finally, 23 tables, with their corresponding columns, which have allowed creating connections between teams, players, leagues and matches, among other things.
- Colorful UX / UI design, easy on the eyes, easy, handy and intuitive.
- The user has been given the possibility to improve their players and move up to higher leagues by creating training and a league system.
- A game simulator has been developed that adjusts to the reality of a basketball game, and which largely depends on the tactic the player decides for his team.

In one of the several tests that have been carried out during the final results checking process (/app/Console/Commands/test.php) (Figure 22), we can see, for example, that the points of each

team oscillate between 55 and 110 points, the maximum points of the players are around 35 or 40, and the rebounds move between 15 or 17. To this we can add 6 or 7 assists, 2 or 3 steals and 5 fouls as the highest statistical data of a player.

Really positive results that could be seen in any professional basketball game.

```
77 - 91
Davide Martinelli | Points: 16
Georges Leclercq | Rebounds: 10
Davide Martinelli | Assists: 3
Veit Hoppe | Steals: 1
89 - 85
Walther Mayer | Points: 27
Guillermo Hernández | Rebounds: 13
Ahmet Atakol | Assists: 5
Walther Mayer | Steals: 1
78 - 96
Bora Aydan | Points: 16
Veit Hoppe | Rebounds: 9
Marc Lacroix | Assists: 3
Benno Schlegel | Steals: 0
86 - 89
Edward Herzog | Points: 13
Jens-Uwe Voß | Rebounds: 7
Aaron Messina | Assists: 3
Richard Techer | Steals: 1
93 - 85
Cirino Coppola | Points: 16
Danny Fontana | Rebounds: 10
Gerald Bach | Assists: 4
Sebastian Böhme | Steals: 1
106 - 90
Maggiore Ferraro | Points: 17
Thierry Fouquet | Rebounds: 7
Christos Reiter | Assists: 4
Max Rizzo | Steals: 1
84 - 70
Edward Herzog | Points: 13
Rupert Weis | Rebounds: 8
Berkay Okur | Assists: 3
Zacharie Imbert | Steals: 1
78 - 91
Pericle Serra | Points: 17
Fabio Marchetti | Rebounds: 10
Davis Parisi | Assists: 4
Veit Hoppe | Steals: 0
89 - 76
Vincenzo Krämer | Points: 14
Claudio Miller | Rebounds: 13
Samuel Pons | Assists: 5
Georges Leclercq | Steals: 0
87 - 70
Max Romano | Points: 13
Benno Schlegel | Rebounds: 12
Philippe Barthelemy | Assists: 4
Benno Schlegel | Steals: 0
```

Figure 24. Test of matches and stats

The user testing part will be carried out by friends and family, who will test the game in its final version to find errors, small details that may have escaped me and give me feedback for possible improvements, either in functionality or design.

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

This project has really been more complicated than I originally thought. I wanted to take care of every little detail to make the experience unique.

Starts are always complicated, and thinking about the right way on which to build development was a real challenge. Number of players per team, teams per league, division of seasons by weeks... they were not ideas designed from one day to the next, since they involved making development one way or another.

Another really difficult point, but rewarding at the same time, has been the creation of players. Getting the attributes of each player related to their position and their physical attributes such as height and weight was a tedious task. Making, for example, that a Point Guard, which is usually one of the shortest players on the team, does not measure 2.13cm or has a better shot than a Center, obviously as a general rule, has required several hours of work on game concept.

And finally, the job more complicated to think, the match simulator, the real challenge. If the reader of this thinks about all the variables that must be handled to simulate a basketball game, he will surely be missing more than half when he has finished. Global time, playing time per player based on the distribution of minutes arranged in tactics, substitutions, personal fouls, statistics, changes of possession depending on events... are just some of the factors that condition the simulation.

In conclusion, a great challenge, of which I am really proud, and which serves as the first version of what is to come.

5.2 FUTURE WORK

As I just commented, this is the first step towards something much bigger, since the control of the equipment by the user, in a definitive version will be total, with great additions:

- **Financial control.** The player will have many factors to have monetary control of the club. Sponsors, stadium, wages or infrastructure are some of the examples in which there will be losses or gains to grow or disappear.
- **Sports control.** The buying and selling or exchange of players is a fundamental part of my idea, something that will make the player able to maintain the competitiveness of his team as it grows economically and athletically.
- **Social and personal control.** Here would be the personal and narrative part of the game, in which the fans of the club will have to be kept happy and have a simulated social life according to a real manager.
- **Internal leagues.** One of the ideas to monetize the game is to offer the possibility of creating internal leagues between friends, with a specific date of game play time, so that, apart from its main league, any group of friends can compete in a private league.

BIBLIOGRAPHY

LINKS

- [1] Sublime Text 3: <https://www.sublimetext.com/>
- [2] Photoshop: <https://www.adobe.com/es/products/photoshop.html>
- [3] HTML/CSS: <https://www.w3.org/standards/webdesign/htmlcss>
- [4] JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [5] JQuery: <https://jquery.com/>
- [6] AJAX: https://www.w3schools.com/xml/ajax_intro.asp
- [7] PHP: <https://www.php.net/manual/en/intro-what-is.php>
- [8] Laravel: <https://laravel.com/docs/7.x>
- [9] Heidi: <https://www.heidisql.com/>
- [10] XAMPP: <https://www.apachefriends.org/es/index.html>
- [11] Stratos: <https://www.stratos-ad.com/>