World Journal of Information Systems 2(1) 10-29
DOI: https://doi.org/10.17013/wjis.v2i1.28

Article

A Collaborative Project between Academia and IT Company for the Application of Software Engineering Concepts

José Carlos Metrôlho[1,2,*], Fernando Ribeiro[1,2], Diogo Marçal[1], Daniela Rei[1] and Diogo Pacheco[3]

[1] Polytechnic Institute of Castelo Branco, Castelo Branco, Portugal

[2] CISeD-Research Centre in Digital Services, Viseu, Portugal

[3] Do iT Lean, Leiria, Portugal

***** Correspondence: metrolho@ipcb.pt

(Article history: Received January 02, 2025; Received in revised form January 29, 2025; Accepted February 20, 2025; Available online March 15, 2025)

Abstract: Fantasy Sport Games have gained popularity among sports enthusiasts, offering not only increased engagement with the sports they follow but also a diverse range of platforms. These applications enable users to form fantasy teams by selecting real-life athletes whose performances in actual competitions earn points for the team. This paper details the development of the Fantasy Sport League, a web-based Fantasy Sport Games created within an academic setting that leverages web scraping techniques to gather real-time data necessary for its functionality. The project represented a collaboration between academic institutions and the industry, providing students the opportunity to apply software development principles from design to product testing. This initiative enriched the students' learning experience and resulted in both a backoffice web application and a mobile app built with OutSystems. The application focuses on soccer, specifically the Portuguese Primeira Liga, allowing users to select athletes, manage leagues, and access a variety of additional features. This paper outlines the people, process and product involved in the project and shares the findings from user acceptance tests conducted to evaluate the satisfaction of potential Fantasy Sport League users with the final application.

Keywords: Fantasy Sport; OutSystems; education; web scraping**.**

© 2025 ITMA. All rights reserved.

1. Introduction

Fantasy Sports Games (FSGs) are popular applications among fans of various sports disciplines. These games allow users to engage in one or more sports by creating a fantasy team composed of real-life athletes, considering a set of rules and restrictions that vary depending on the application and the type of sport. At the end of a period predetermined by the application, the athletes in each fantasy team earn points based on their actual performance. Thus, the sports data generated during the competitions serve as the foundation of FSGs—not only because they provide the necessary data for the applications to function, but also because users can leverage this data to predict which athletes will yield the most points for their fantasy teams.

This type of game enjoys widespread popularity for two main reasons: first, it allows sports fans to engage with sports content in ways beyond traditional consumption - i.e., following live matches - by adopting a managerial role, albeit fictitious; and second, these applications are available across multiple platforms [1].

This paper describes the development process of an FSG using the OutSystems platform, carried out within the framework of a partnership between a higher education institution and an IT company, Do iT Lean, which closely followed the entire process. The goal was to provide students with hands-on experience in real-world software development, thereby reinforcing the practices and concepts learned. The final product aimed to deliver a web and mobile application using the OutSystems platform [2], structured as an FSG in which each user assembles a fantasy team, constrained by a budget and a set of base rules, with the aim of scoring as many points as possible to outperform other users/players.

The application developed, named Fantasy Sport League (FSL), focused on Portugal's top-tier soccer league. In FSL, authenticated users can build a team of athletes who, at the end of each matchday, generate points that accumulate over the season and determine the league winner.

This paper presents the development process and shares the results of this academia-industry collaboration, including the people involved (Section 2), the process (Section 3), the project (Section 4), the product (Section 5), and concludes with the final considerations (Section 6).

2. People

To implement the application, a team of six individuals was assembled: four from the higher education institution and two from the company. The team adopted a customized version of the Scrum framework [3] as its working methodology, as this is the agile development approach used by the company's members. The development was carried out using the OutSystems low-code development platform, which is also the platform used by the company. The use of this platform brought agility to the software development process, allowing the functional requirements defined in the project objectives to be effectively met.

© 2025 ITMA. All rights reserved.

In addition to the knowledge acquired through training and use of the OutSystems platform, the students also gained valuable experience in the use of web scraping technology to collect real-world data for the FSL.

3. Process

This section describes the overall process, including planning and its main phases, the analysis of similar applications that served as a reference for defining the FSL application's requirements, and the methodology adopted.

3.1. Planning

The project was scheduled to last for 10 months, corresponding to a full academic year, starting in October and ending in July. Due to specific constraints, the project was divided into two parts: the first running from October to February, and the second from March to July. The planning of the first phase included six stages (three of which were Sprints), while the second phase comprised ten stages (seven of which were Sprints). Figure 1 presents the project phase timeline. Some adjustments were made throughout the project in response to its natural progression.

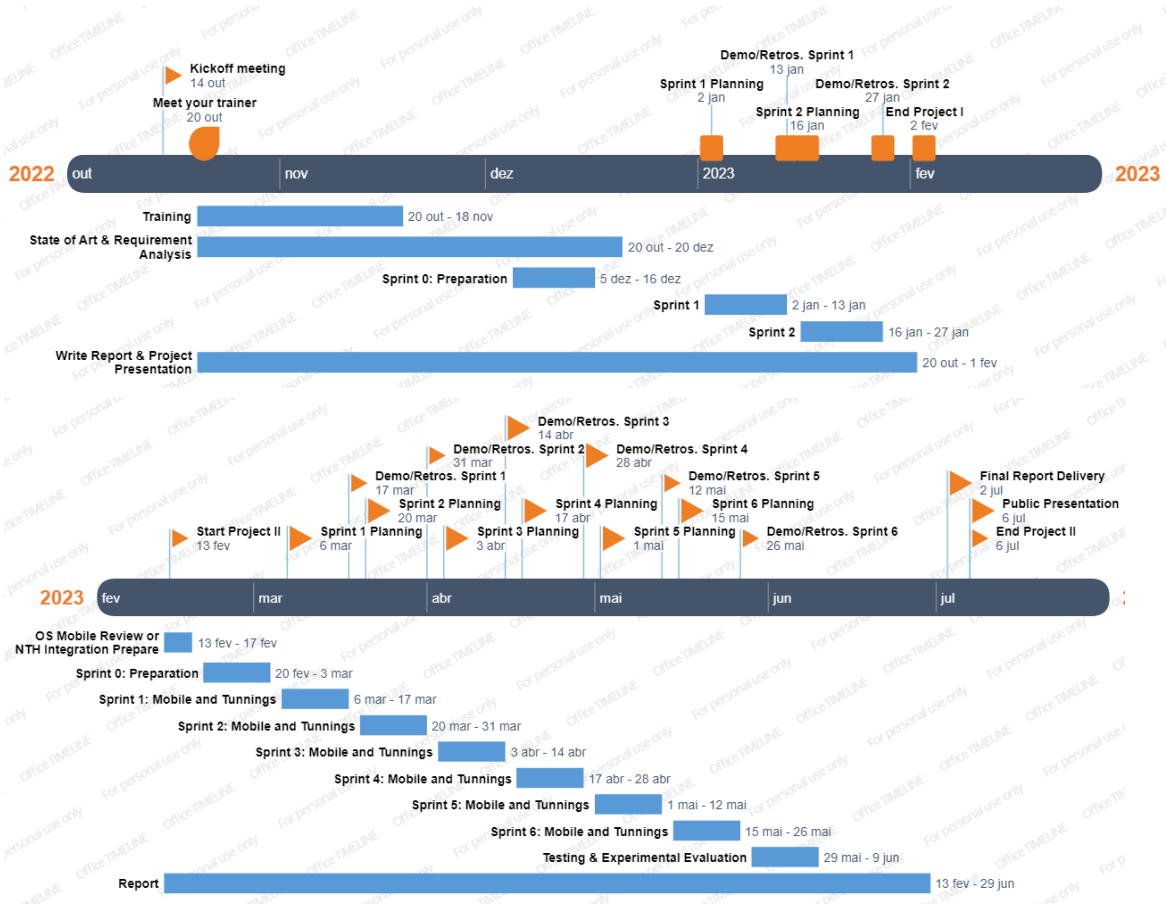


Figure 1. Timeline of the Project Phases: (a) First part; (b) Second part.

© 2025 ITMA. All rights reserved.

3.2. State of the Art

In the initial phase of the process, an analysis of several articles addressing the intended topic was conducted. The approach focused on reviewing literature surveys related to FSG applications, as well as exploring applications available on app stores and across the internet. The following guiding questions were used to support the analysis of the scientific literature:

- Who is the target audience of FSGs?
- What are the potential sports disciplines targeted by FSGs?
- What motivations and strategies are used to keep users engaged in the game?
- What are the most common methods for gathering the information needed to "feed" the game?
- What are the typical rules of FSGs?
- Are forecasting or prediction mechanisms used in FSGs?

Scopus was used as the data source. The search string applied was "fantasy AND sport*". The search was conducted across document title, abstract, and keywords, and restricted to the fields of Engineering and Computer Science. A total of 96 results were obtained. By filtering for publications from the last 10 years (since 2012—older works were excluded due to technological evolution and current relevance), 23 documents were excluded, resulting in 73 articles. After screening the titles and abstracts, 63 articles were found to be unrelated to the topic, leaving 10 articles. Following a more detailed review, 6 of these were further excluded due to insufficient information or misalignment with the research focus, resulting in a final selection of 4 articles for analysis.

Upon analysis, it was found that most of these articles focus on the development and evaluation of prediction systems that enable users to make informed decisions when selecting teams and players, aiming to maximize points based on historical data. Only one article provided details on the architecture and technologies used in implementing the proposed model - specifically, article [4], which used Python and Jupyter Notebook.

Regarding game mechanics, all articles mention the objective of FSGs and how they rely on real-world athlete performance to award points to users. However, only article [5] provides detailed information on the game rules and identifies the target audience for such applications.

As for the sports modalities covered by the FSGs in these studies, although not all refer to a specific application, each article addresses a particular sport, ranging from cycling [5] to cricket [4], and both American football [6] and English Premier League of soccer [7].

Given that information is essential to the functioning of FSGs, all reviewed articles present different strategies for obtaining such data. These include: an online ranking system [5]; a "League Commissioner" or platforms that periodically collect the necessary data [6]; an API constantly updated by the Fantasy Premier League (FPL) [7]; and an open dataset on the Indian Premier League (2009–2019) [4].

© 2025 ITMA. All rights reserved.

Some articles also discuss strategies to enhance user engagement. For instance, article [5] mentions the provision of daily and seasonal rewards as a motivation mechanism, as well as offering insights into the best possible score at a given point in the season. Article [7] emphasizes that the multi-platform availability of the application contributes to its growing user base.

In conclusion, most FSGs are built upon real-world athlete performance data from actual competitions. As such, the collection and handling of this data play a crucial role in game functionality. For less experienced or new users, several methods of analysis and prediction are presented, including systems for generating optimal or "ideal" teams, based on user inputs or predefined algorithms. All reviewed articles reported positive outcomes regarding the implementation of their respective team prediction systems.

Table 1. Comparison of related works.

Work	Year of Publication	Motivation Strategy for Players	Target Sport	Strategy to Obtain Game "Feed" Information	Existence of a Prediction System for the Best Team
Optimization modelling for analyzing fantasy sport games [5]	2017	Motivating information for players joining mid-season and competition between players	Cycling across various categories	Cyclist rankings available on the web	MIP model that can be applied to other sports
Bidding Strategies for Fantasy-Sports Auctions [6]	2016	Not defined	American football	"League commissioner" or online platforms for collecting match data	Auto-draft mechanism that selects athletes if the team manager is absent during the draft or to assist less experienced players
Player Recommendation System for Fantasy Premier League using Machine Learning [7]	2022	Applications are cross-platform	English Premier League soccer	API provided by FPL	A system is developed to predict ideal athletes using various metrics available via the API or generated by the authors
Dream11 IPL Team	2022	Not defined	Cricket	Dataset on Indian Premier	Prediction method for the

© 2025 ITMA. All rights reserved.

Recommendation using Machine Learning and Skill-Based Ranking of Players [4]				League 2008-2019	best team based on inputs provided by users, such as real teams that the user wants to base their team

In addition to these scientific studies, applications available on the Google and Apple app stores were also analyzed using the keywords “fantasy sport”. Four applications were selected, [8], [9], [10], [11], whose features closely resemble those intended for this project. Furthermore, following a search on web search engines using strings such as "fantasy sport game" and "fantasy sport," two additional relevant applications were identified: [12], [13]. This analysis revealed that FSGs span various sports and represent a growing market. They also incorporate a range of prediction techniques to identify the optimal team under specific constraints, such as the user’s available budget.

The analysis of these six applications further showed that, although each app offered different features, they shared several common traits. This helped identify the functionalities provided by leading applications of this kind available in the market and enabled the detection of recurring patterns in the rules across different platforms. It can be inferred that there is a set of rules and characteristics that developers generally consider best suited for this type of application.

3.3. The Methodology

The development methodology was defined by the team members of the company. Thus, as is customary within the company, this team also followed a methodology based on the Scrum framework, although with some adaptations compared to the original Scrum [14]. The adaptations were because it was an academic - or semi-corporate - project, meaning that, for instance, it was not possible to hold the Daily Scrums on a daily basis. However, biweekly meetings were maintained, and the frequency was adjusted whenever necessary. On the other hand, Sprint Reviews and Sprint Retrospectives took place at the end of each Sprint. These Sprints varied in length from 2 to 3 weeks.

Regarding artifacts, a product backlog was created, for which the product owner (company representative) and the Scrum Master (academic advisors) played an essential role, particularly in defining the requirements that were prioritized for the next sprint. For each User Story in the backlog, acceptance criteria were also defined, which would help to verify the functionalities during the testing phase. It is important to mention that in the initial phase, the students had to learn the methodology through training suggested by the company members. This training included the following topics:

© 2025 ITMA. All rights reserved.

- Web Fundamentals – In this guided path, several key concepts were reinforced, including:
    a) HTTP Essential Training – about the Hypertext Transfer Protocol, particularly the differences between versions of the HTTP protocol, how requests and responses are made, how HTTP headers are structured, as well as the concept of cookies and caching [15].
    b) HTML Essential Training – about the role of HTML in web programming, text formatting methods, creating links, and using tables in HTML. It also covered how to manage images, graphics, media files (video and audio files), and other related topics [16].
    c) Learning REST APIs – about what a REST API is, the difference between URL and URI, RESTful libraries, as well as how REST requests and responses work, specifically the methods used in those requests and responses, such as GET, POST, among others [17].
    d) Programming Foundations: Web Security – in this module, the concept of hacking, some principles of cybersecurity, methods to improve security during web programming, as well as common cybersecurity attack methods were covered [18].
    e) Learning ECMAScript – about what ECMAScript is, how variables, data structures, arrays, and array methods are declared, objects, some other important functions, as well as strategies for incorporating JavaScript [19].
    f) Learning React.js – about the concept of React programming and key elements, components, and improvements in React programming using JavaScript [20].
- SQL Introduction – in this guided path, the following topics were addressed:
    a) Programming Foundations: Databases – about the concept of databases. This course also reinforced some database fundamentals, such as the concept of keys, unique values, and SQL syntax [21].
    b) SQL Essential Training – about SQL programming in greater depth, particularly performing SELECTs, creating, editing, and inserting records, among other table operations, such as how to relate multiple tables, using data types like strings, numbers, dates and times, aggregated data, etc. The concept of transactions, triggers, views, and subselects was also presented, along with performing CRUD operations [22].
- Agile Methodology OutSystems – about what agile methodology is, covering concepts related to agile methodologies [23].
- Scrum: Foundations (for developers) – for better adaptation to the framework that would be used:
    a) Scrum: The Basics – about the Scrum framework, its advantages, specifications, and the various concepts required for its implementation [14].

© 2025 ITMA. All rights reserved.

b) Agile Software Development: Scrum for Developers – about using Scrum, particularly delving into concepts such as roles in Scrum, artifacts, events like sprints, daily scrum, daily retrospective, among others, and product backlog, with examples of applying each concept in a tool like Jira [24].

- The Do iT Lean Methodology – about the methodology used by the partner company in the development of this work [25].
- Becoming a Reactive Web Developer – a guided path consisting of 20 courses that cover the basics of reactive web programming in OutSystems [26].
- Becoming a Mobile Developer – a guided path consisting of 26 courses covering the basics of mobile programming in OutSystems [27].
- Integrating with Web Services – an OutSystems course explaining how to integrate external web services into code developed on the OutSystems platform [28].
- Best Practices on Architecture Design – a course addressing best practices related to the architecture of applications developed using the OutSystems platform [29].

During this period, in addition to the training, a state-of-the-art analysis was conducted, requirements analysis and modeling of the FSL application were performed, the tools used were described, and research was done on low-code platforms and web scraping.

In terms of implementation, the system development began with the implementation of the core modules as well as the back office. Additionally, documentation and reports were also written. During this first phase, 19 tasks were identified, which represent work units for the implementation to be executed. In the first phase, the tasks related to the back office were completed, except for those involving the FSL rules, which, due to time constraints, were moved to the second phase.

In the first part, Figure 1 (top), there were 3 sprints (1 for preparation and 2 for implementation). For each sprint, tasks were selected from the backlog (the set of all identified tasks) based on their priority. Sprint 0 was used to consolidate the requirements analysis. The tool used to monitor the progress of task development throughout the project was Trello [30]. Figure 2 shows a screenshot of Trello at the beginning of the second phase.

© 2025 ITMA. All rights reserved.

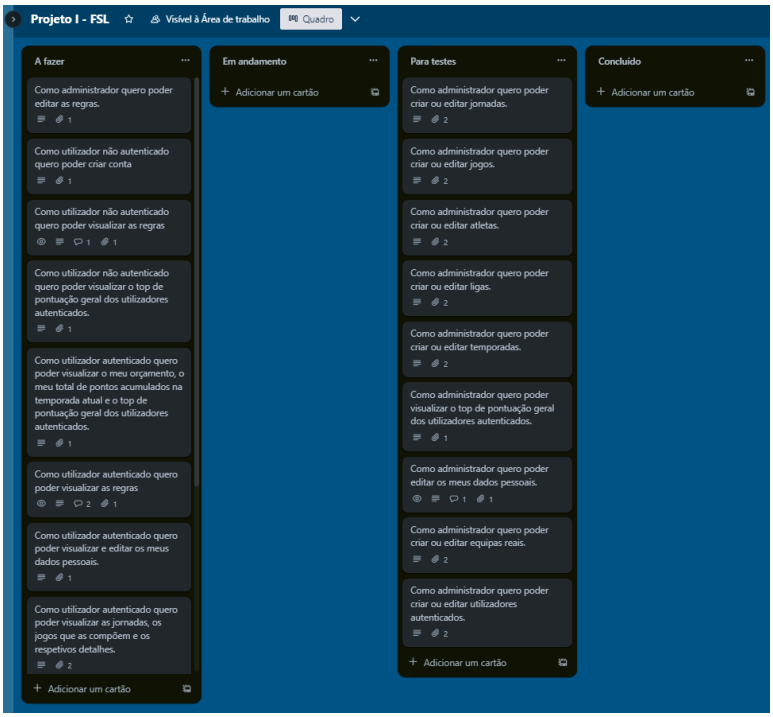


Figure 2. Tasks on Trello at the beginning of the implementation of the second phase.

In the second phase, the timeline presented in Figure 1 (bottom) was followed. During this phase, in terms of implementation, the FSL back office was completed, including the implementation of the rule functionalities in the back office (view, create, edit, and remove). Next, the development of the FSL_BL module, responsible for the business rules of the FSL, was started. Once that module was completed, the development of the mobile app, which constitutes the FSL game for the authenticated user, began. After finalizing the mobile app, the web scraping mechanism was developed to populate the FSL database. Documentation was produced throughout the work process.

Out of the 19 tasks to be implemented, 9 were completed in the first phase, and the remaining 10 tasks were carried over to the second phase. During this entire period, some changes were made both in the implementation and in the documentation produced, and wireframes for the mobile app were created.

4. Project

The following provides a summary of some of the artifacts created during the two phases of the project, corresponding to the different stages of the development cycle.

4.1. *Requirements*

In the initial requirements analysis phase, the actors were identified and described, i.e., the different roles that interact with the application, as well as detailed descriptions of the functionalities. The three actors considered were: Unauthenticated User, Authenticated User, and Administrator. For each of these actors, the following functionalities were designed:

© 2025 ITMA. All rights reserved.

- Unauthenticated User: has the option to create an account or log in with an existing one. This type of user has limited functionality available as they must authenticate and log in to be able to execute the main functionalities of the application.
- Authenticated User: a user with a previously created account and an active session. This user can perform all actions related to the gameplay provided by FSL, such as creating and joining leagues, choosing a team, and buying or selling players within the available budget at any given moment.
- Administrator: this is the type of user with the most privileges in FSL. The administrator is responsible for overseeing the consistency and currency of the data needed for the game. They perform operations related to managing players, such as defining when a player is injured, managing authenticated users, controlling who participates in each league, and modifying the information regarding FSL rules available to authenticated users, among other data management functionalities.

4.2. Modeling

After identifying the actors and their desired functionalities for the FSL application, a modeling phase followed in which a use case diagram was created. The Use Case diagram is visually easy to understand and discuss with the other team members. Subsequently, wireframes were designed for both the web back-office component and the FSL mobile component. The wireframes illustrate the functionalities included in the back-office and mobile app, as shown in the use case diagram. They also helped identify the fields required for the entities in the application database. These wireframes, in some cases, did not exactly match the interfaces created during the implementation phase because, during this phase, the development team deemed it necessary to adjust some of the interfaces represented by the wireframes for the improvement of the application. Figure 3 presents a set of navigation diagrams for both components, showing some of the wireframes and how they relate to each other.

After completing the modeling phase, a set of 19 tasks (User Stories) was created, which made up the product backlog and served as the basis for planning the Sprints executed in both phases of the project.

In terms of the architecture of the FSL application, the Four Layer Canvas model was followed. This model consists of two applications: FSL BackOffice and FSL Core. The FSL BackOffice application includes a back-office front-end module; the FSL Core application includes modules related to the data model, FSL_CS (Core Service), and business logic represented in the FSL_BL module. Additionally, the FSL Web Scraping application is responsible for collecting data on players, games, and other information. The FSL application (or FSL mobile) includes the code for the mobile application. This architecture can be observed in Figure 4.

© 2025 ITMA. All rights reserved.

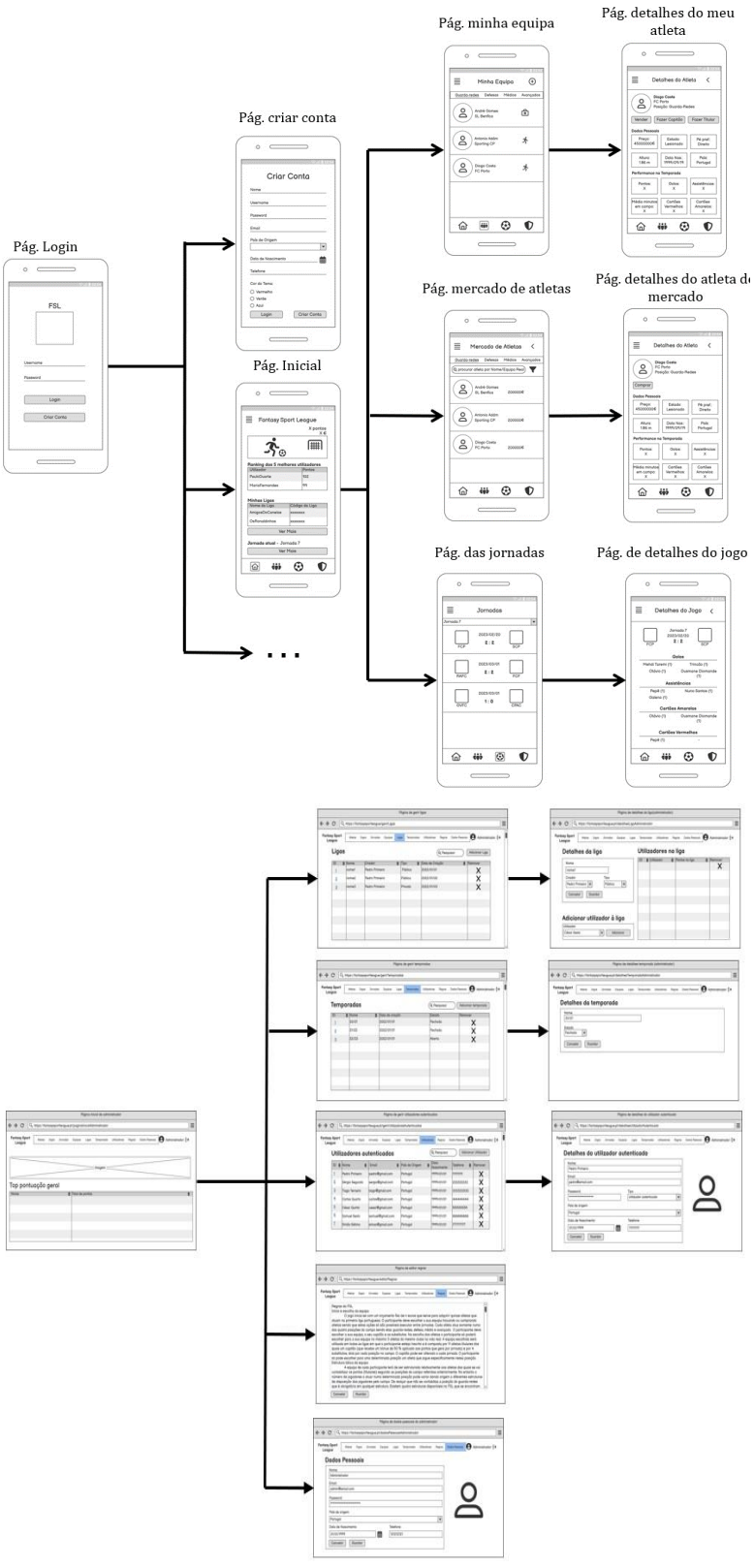


Figure 3. Navigation diagrams of: (a) mobile app (top); (b) back office (bottom).

© 2025 ITMA. All rights reserved.

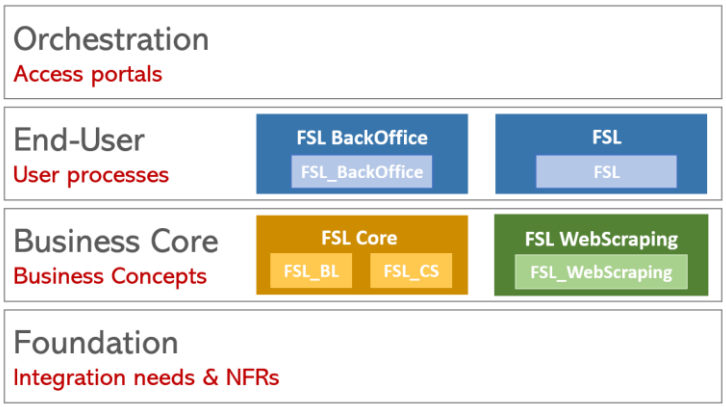


Figure 4. FSL Architecture.

Another important artifact was the data model. That is, the database modeling that supports both the web application, back office, and the mobile application. The database modeling was done before starting the implementation of the application. The identification of entities and their respective fields resulted from the analysis of the previously created wireframes, which allowed the team to determine the data necessary for the application's functionality. As the development team was using an adaptation of an agile methodology, the fields required in the database underwent some changes during the implementation phase compared to what was identified from the wireframes. During the implementation phase, it was also necessary to alter some of the entities and/or attributes of certain tables. Figure 5 shows the ER diagram of the database, the designed version.

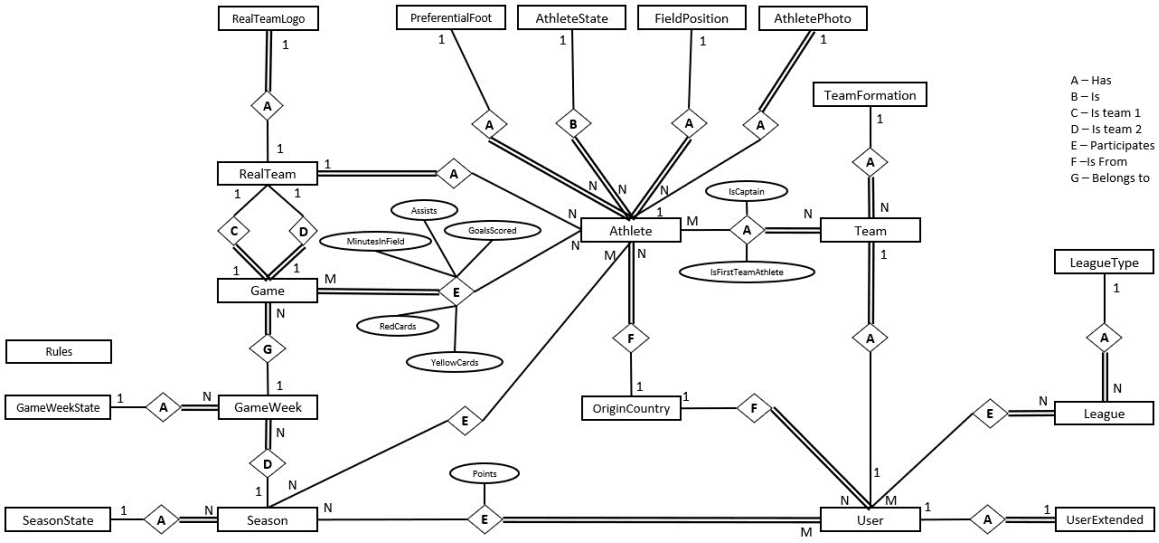


Figure 5. ER Diagram of FSL.

4.3. Tools

Various technologies and tools were used for the development of this work. Below are the most relevant ones. The Web Scraping technology [31] was employed to gather the data necessary for the operation of the application to be developed. The data was obtained using

© 2025 ITMA. All rights reserved.

this technique and came from a website dedicated to presenting sports statistics and events. The application was implemented using the low-code platform OutSystems [2]. Trello [30] was used to manage the project's development and monitor the status of tasks that needed to be completed by the team members. The Balsamiq Wireframes tool [32] was used for creating wireframes.

The FSL application uses real data, specifically from sports events (games), teams, and athletes. Thus, the data required for the application to function was gathered through web scraping from a website dedicated to presenting sports statistics and events. The web scraping process was made possible using OutSystems, with the company Do iT Lean providing a set of information [31] that helps students/developers understand how to perform web scraping using OutSystems.

The data is presented on the website in an unstructured format. By collecting data from this page, the web scraping process converts it into a structured format. The data collected by web scraping can be of various types, such as videos, images, or text [33].

For most web scraping processes, two components are required: the web crawler and the web scraper. The web crawler is an artificial intelligence algorithm that navigates the web searching for the data to be collected, passing the URLs of relevant pages to the web scraper. On the other hand, the web scraper is responsible for quickly and efficiently extracting the data from the website [34][33].

There are different types of web scrapers depending on their origin and characteristics. Regarding the origin of web scrapers, they can either be self-built, developed by the user to meet more complex and specific web scraping needs. However, this requires a higher degree of expertise, particularly in programming skills. For less specific needs, pre-built web scrapers can be used, which are downloaded from the web and already developed, saving time for the user and allowing some level of configuration of the tool. There are also web scraping extensions for browsers, though they have many limitations compared to dedicated software web scraper solutions [33].

The web scraper can extract either the entire webpage or just a portion of it, with the latter option generally requiring less time for the process. The web scraping process begins when the web scraper receives the URL of the target page from which it should collect data and access the page via an HTTP request. The web scraper proceeds by collecting the HTML code and may even retrieve the CSS and JavaScript of the page in cases of more complex web scrapers. The web scraper analyzes the retrieved documents and uses locators to find the relevant data. A parsing process follows where the scraper divides the document by elements, such as tags or text pre-defined by the person configuring the scraper. During the development or configuration of the scraper, actions must be performed, such as defining the elements to be saved, which belong to the target page, and converting the obtained data into valid data types for the chosen storage solution. Once the data is collected, it is loaded into databases or JSON files, for example [34][33][35].

© 2025 ITMA. All rights reserved.

4.4. Implementation

The main blocks implemented were the FSL Core application (including the FSL_CS and FSL_BL modules), the back office within the FSL_BackOffice application, the mobile app named FSL mobile, and the web scraping mechanism developed within the FSL Web Scraping application.

The implementation carried out in the first part corresponded to the entire FSL_CS module and the beginning of the FSL back office. In this back office, the administrator can manage the entities relevant to the application. Although the data for the application is automatically obtained through web scraping, the existence of the back office was deemed essential, for example, to allow the administrator to correct any errors in the data. The implementation of both the necessary code and the application interfaces was done in English, as recommended by the company, justifying that in the business world, the overwhelming majority of applications use the English language, which is considered a good programming practice.

The implementation of the second part consisted of completing the FSL back office, developing the FSL_BL (module that applies the FSL business rules), developing the mobile app, and implementing the web scraping process.

To build all these modules within the application architecture, artifacts such as database entities, server and client actions, and pages were developed.

5. The Product

This section presents the final product, the prototype, resulting from the development process. The main implemented elements were the FSL Core application, the FSL back office, and the mobile app named FSL mobil. To illustrate the final product, the interfaces of some functionalities are presented. This section concludes with a description of the acceptance tests conducted.

5.1. Prototype

In the FSL back office, the administrator can manage the relevant entities for the application. Although the data for the application is automatically obtained through web scraping, the existence of the back office was deemed essential so that, for example, the administrator can correct any data errors. The FSL back office allows for the consultation and management of information related to: Athletes, Games, Rounds, Teams, Leagues, Seasons, and application users. For each of these entities, the application allows searching and viewing information, deleting/adding entities, and viewing details of each instance.
Figure 6 illustrates the interface for querying information about the athletes loaded into the system.

© 2025 ITMA. All rights reserved.

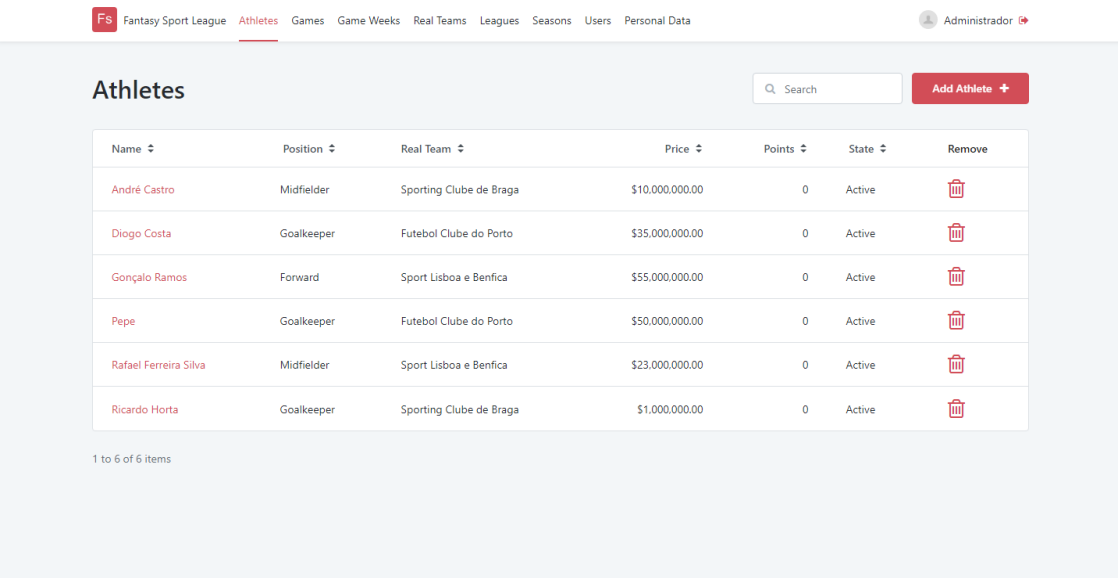


Figure 6. Athlete data query UI.

An important feature relates to the definition of the rules that determine the scoring to be assigned (Figure 7).

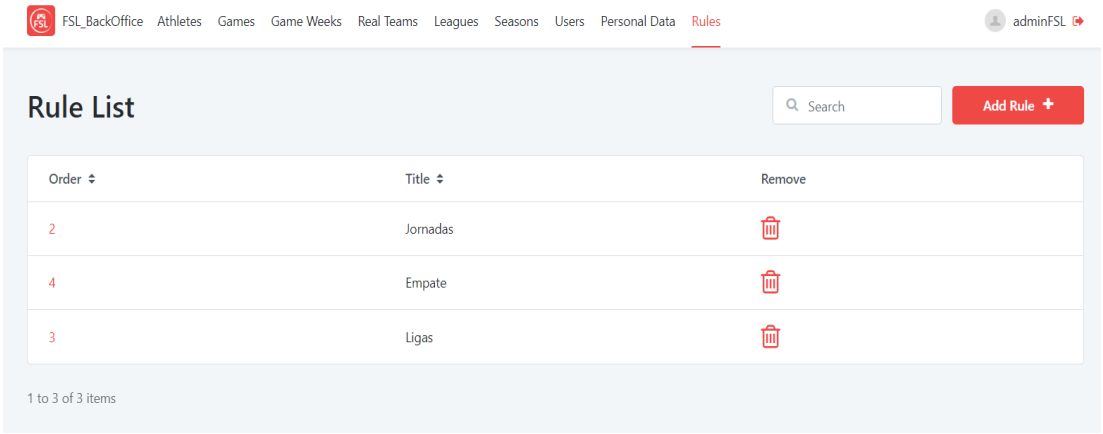


Figure 7. Create and configure rules.

The player uses the FSL mobile application to participate in the game. The menu (Figure 8, left) provides access to 5 main areas: my team; the leagues; the athlete market; the rounds; the game rules (Figure 8, center); and access to personal data. The player can see all the players in their team and view details of each one (Figure 8, right). The constitution of each team follows a specific tactical structure. Each player has an initial budget, which can be adjusted according to the appreciation/depreciation of the real athletes in the team and the transactions made in the athlete market. The player's team points depend on the real performance of each athlete according to their performance in each round, including factors such as: whether they were starters, their playing time, whether they received disciplinary cards, whether they scored goals or provided assists, etc. Only the points generated by the starting players are counted in each round.

© 2025 ITMA. All rights reserved.



Figure 8. FSL mobile: (a) Menu with main options (left); (b) League rules (center); (c) Athlete details (right).

Access to the athlete market (Figure 9, left) allows the player to buy or sell athletes. The player can also select a specific round and view the results of the games (Figure 9, right), as well as the details of each game.

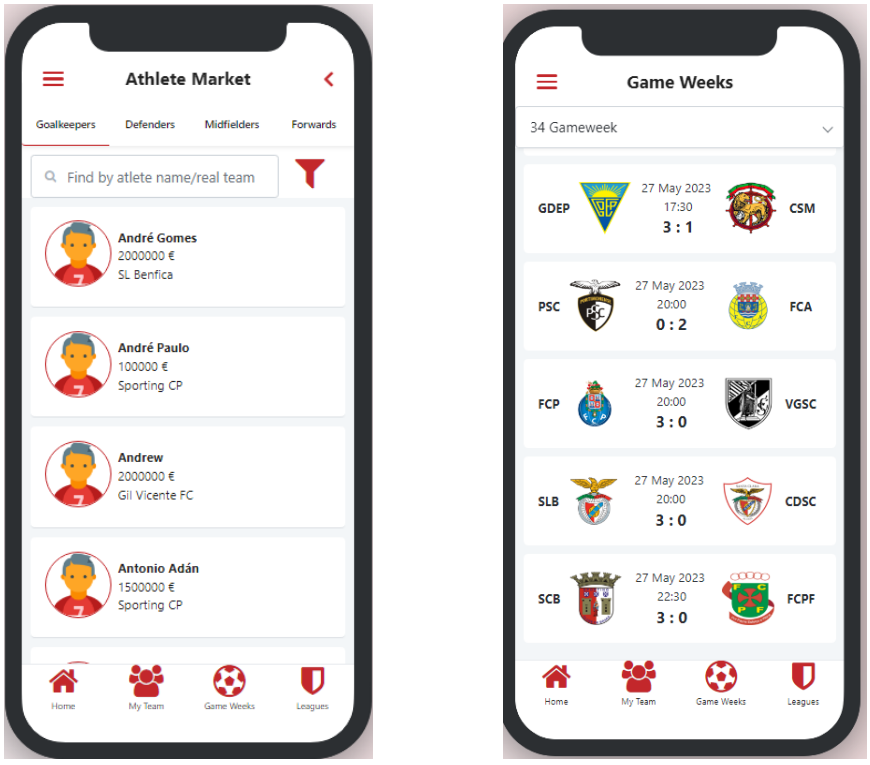


Figure 9. FSL mobile: (a)Athlete market (left); (b) Round results (right).

© 2025 ITMA. All rights reserved.

The operation of the application is powered by real data obtained through web scraping. Periodically, the FSL Web Scraping application performs this operation on the Transfermarkt website (www.transfermarkt.pt/) to obtain statistical information about athletes and weekly games. This includes the athletes who played, the goals they scored, etc. These data are essential to maintain the mechanics and attractiveness of the game. The Transfermarkt website was chosen as the source for all data for two main reasons: the fact that the website contains all the necessary data, and the fact that it was verified that it is possible to obtain this data using web scraping. About 500 athlete records were collected, as this is approximately the number of athletes competing in the Portuguese top soccer league. For each athlete, it was necessary to collect the webpage containing their participation and performance data for each game, which represents about 7,000 records.

5.2. Acceptance Tests

Acceptance tests were conducted to evaluate the acceptance and usability of the developed applications. Several potential FSL users were involved, and after using the FSL, they were asked to fill out a set of questions related to the application's usability and user satisfaction with the provided features. For usability evaluation, several questions adapted from statements suggested by the Public Administration Digital Competence Center were presented, based on the System Usability Scale (SUS) created in 1986 by John Brooke. Users expressed their degree of agreement with these statements [36]. To assess acceptance, users were presented with several statements created by the development team to evaluate their level of satisfaction regarding functional requirements using the same scale as the usability statements. For each question, users had to indicate how much they agreed with each statement on a scale from 1 to 5, where 1 means "Strongly Disagree" and 5 means "Strongly Agree".

Nine users participated in this evaluation, with an average age of 35. Four of the participants had previously used FSL applications. The results obtained are presented below, where for each question (a total of 17), respondents were able to rate from 1 (strongly disagree) to 5 (strongly agree). As seen in the results expressed in percentage terms Table 2 and Table 3, there was general satisfaction with the FSL.

Table 2. Usability results.

Questions	1	2	3	4	5
Would you use the application?			11,1	33,3	55,6
Is using the application complex?	44,4	33,3	11,1	11,1	
"I find the application easy to use."				77,8	22,2
"I think I would need help from someone with technical knowledge to use the application."	77,8	22,2			
"I think the various options in the application are very well integrated."			11,1	66,7	22,2
"I think the application presents a lot of inconsistencies."	77,8	22,2			

© 2025 ITMA. All rights reserved.

Table 3. Feature results.

Questions	1	2	3	4	5
"I imagine people will learn how to use the application quickly."				55,6	44,4
"I find the application easy to use."	66,7	33,3			
"I felt confident using the application."				33,3	66,7
"I needed to learn new things before I could use the application."	77,8	22,2			
"I find the athlete data useful for forming teams."				66,7	33,3
"I think the functionality that allows me to change the color of the application's elements is relevant."			11,1	55,6	33,3
"I think it's good that I can create and join private leagues."			22,2	66,7	11,1
"I think it's good that an athlete can play multiple roles in my team (starter, captain, or substitute)."				33,3	66,7
"I consider it good that the application provides data on the rounds and games of the Portuguese top soccer league."			11,1	33,3	55,6
"I easily found the athletes I was looking for."				22,2	77,8
"I think it's important that the real performance of the athletes in my team is converted into points."				44,4	55,6

In addition to the tests described here, throughout the implementation period, tests were conducted on each of the FSL functionalities by the development team to ensure that the functional requirements implemented could be successfully executed.

6. Conclusion

This article presents the software development practices and concepts applied in an academic project carried out in collaboration with a software development company. Over the course of an academic year, an FSG was implemented, involving the creation of an IT solution comprising a back-office system and a mobile application, the latter consuming real data from the Portuguese Soccer League through web scraping. The OutSystems platform, used for implementing the solution, as well as the web scraping technique, were learned and applied by the students during the project.

The students were immersed in an agile development process, adopting professional practices used by the company, which provided a working environment that can be considered a simulation of a professional setting. Throughout the project, various development and project management tools were employed, and the students worked in teams - a dynamic rarely explored in the academic context. This experience allowed the students to develop several soft skills, such as dedication, rigor, punctuality, communication, and adaptability to new technologies - skills that are essential in the job market.

In addition to the high quality of the product developed, with well-implemented requirements validated by the company's Product Owner, the completeness of the project and the adoption of a technical-scientific approach that contributes to solid preparation for the demands of the job market in the field of Computer Science stand out. The work aimed to integrate the four pillars of Software Engineering - People, Process, Project, and Product - providing a holistic and practical view for students and reinforcing best practices in software product development.

© 2025 ITMA. All rights reserved.

Acknowledgments: We would like to thank the Research Centre in Digital Services (CISeD) for its support.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] 'What Are Fantasy Sports?' Accessed: Feb. 01, 2023. [Online]. Available: https://sites.psu.edu/pfsports/what-are-fantasy-sports/

[2] 'OutSystems'. Accessed: Jan. 18, 2023. [Online]. Available: https://www.outsystems.com/

[3] 'Scrum Guides'. Accessed: Jan. 18, 2023. [Online]. Available: https://scrumguides.org/index.html

[4] S. Choudhari, N. Wagholikar, A. Swaminathan, and S. Kurhade, 'Dream11 IPL Team Recommendation using Machine Learning and Skill-Based Ranking of Players', in 2022 International Conference for Advancement in Technology, ICONAT 2022, 2022. doi: 10.1109/ICONAT53423.2022.9725819.

[5] J. Beliën, D. Goossens, and D. van Reeth, 'Optimization modelling for analyzing fantasy sport games', INFOR, vol. 55, no. 4, pp. 275–294, 2017, doi: 10.1080/03155986.2017.1279899.

[6] A. Anagnostopoulos, R. Cavallo, S. Leonardi, and M. Sviridenko, Bidding strategies for fantasy-sports auctions, vol. 10123 LNCS. 2016. doi: 10.1007/978-3-662-54110-4_8.

[7] V. Rajesh, P. Arjun, K. R. Jagtap, C. M. Suneera, and J. Prakash, 'Player Recommendation System for Fantasy Premier League using Machine Learning', in 2022 19th International Joint Conference on Computer Science and Software Engineering, JCSSE 2022, 2022. doi: 10.1109/JCSSE54890.2022.9836260.

[8] 'UEFA Gaming: Fantasy Football'. Accessed: Nov. 01, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.uefa.eurofantasy

[9] 'Premier League - Official App'. Accessed: Nov. 01, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.pl.premierleague

[10] 'RealFevr - Fantasy Sports 2022'. Accessed: Nov. 02, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.realfevr.fantasy

[11] 'Cycling Fantasy'. Accessed: Nov. 03, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.cycle_fantasy_app&pli=1

[12] 'Fantrax Fantasy Sports'. Accessed: Nov. 02, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.fantrax.www.twa&hl=pt_PT&gl=US

[13] 'F1 Fantasy'. Accessed: Nov. 02, 2022. [Online]. Available: https://web.archive.org/web/20220507093612/https://fantasy.formula1.com/fantasy

[14] 'Scrum: The Basics'. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/scrum-the-basics

[15] 'HTTP Essential Training'. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/http-essential-training

[16] 'HTML Essential Training'. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/html-essential-training-4

[17] 'Learning REST APIs'. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/learning-rest-apis

© 2025 ITMA. All rights reserved.

[18] ‘Programming Foundations: Web Security’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/programming-foundations-web-security-2

[19] ‘Learning ECMAScript 6+ (ES6+)’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/learning-ecmascript-6-plus-es6-plus

[20] ‘Learning React.js’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/learning-react-js-5

[21] ‘Programming Foundations: Databases’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/programming-foundations-databases-2

[22] ‘SQL Essential Training’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/sql-essential-training-3

[23] Project Timebox Overview - Agile Methodology - Training OutSystems. Accessed: Jan. 18, 2023. [Online Video]. Available: https://www.outsystems.com/training/lesson/1294/project-timebox-overview?LearningPathId=17

[24] ‘Agile Software Development: Scrum for Developers’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.linkedin.com/learning/agile-software-development-scrum-for-developers

[25] The Do IT Lean Agile Methodology. Accessed: Jan. 18, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=-iZENUjpnBw

[26] ‘Becoming a Reactive Web Developer Guided Path - Training - OutSystems’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.outsystems.com/training/paths/18/becoming-a-reactive-web-developer/

[27] ‘Becoming a Mobile Developer Guided Path - Training - OutSystems’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.outsystems.com/training/paths/1/becoming-a-mobile-developer/

[28] ‘Integrating with Web Services Course - Training - OutSystems’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.outsystems.com/training/courses/116/integrating-with-web-services/?LearningPathId=2

[29] Best Practices on Architecture Design - Training OutSystems. Accessed: Jan. 18, 2023. [Online Video]. Available: https://www.outsystems.com/training/lesson/2072/best-practices-on-architecture-design

[30] ‘Trello’. Accessed: Jan. 18, 2023. [Online]. Available: https://trello.com/

[31] M. Antunes, ‘Web Scraping Made Easy: The Full Tutorial’. Accessed: Jan. 18, 2023. [Online]. Available: https://www.doitlean.com/news/read/web-scraping-made-easy:-the-full-tutorial

[32] ‘Balsamiq’. Accessed: Jan. 18, 2023. [Online]. Available: https://balsamiq.com/

[33] ‘What is Web Scraping and How to Use It?’ Accessed: Dec. 29, 2022. [Online]. Available: https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/

[34] H. Cura, ‘Web Scraping - saiba o que é e para que serve’. Accessed: Dec. 21, 2022. [Online]. Available: https://pplware.sapo.pt/internet/web-scraping-saiba-o-que-e-e-para-que-serve/

[35] W. Hillier, ‘What Is Web Scraping? A Complete Step-by-Step Guide’. Accessed: Dec. 29, 2022. [Online]. Available: https://careerfoundry.com/en/blog/data-analytics/web-scraping-guide/

[36] Centro de Competências Digitais da Administração Pública, ‘Testes de usabilidade’, Mar. 2021. Accessed: Jun. 26, 2023. [Online]. Available: https://ticapp.gov.pt/wp-content/uploads/2021/03/TicAPP2021-TestesUsabilidade.pdf

© 2025 ITMA. All rights reserved.