Graduado en Ingeniería Informática
Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Desarrollo de una aplicación web para la programación de laboratorios de biología portátiles

Autor: Ahmed Sefraoui
Director: Alfonso Rodríguez-Patón Aradas

MADRID, JUNIO 2019
Table of contents

RESUMEN ................................................................................................................... 1
ABSTRACT ................................................................................................................. 2
1. INTRODUCTION .................................................................................................... 3
  1.1 Approach.......................................................................................................... 3
  1.2 Objectives......................................................................................................... 4
    1.2.1 Back-End .................................................................................................... 4
    1.2.2 Front-End .................................................................................................... 4
    1.2.3 Functionality ............................................................................................... 5
  1.3 Motivation.......................................................................................................... 5
  1.4 Used tools......................................................................................................... 5
    1.4.1 Hardware .................................................................................................... 5
    1.4.2 Software ..................................................................................................... 5
2. STATE OF ART ....................................................................................................... 8
  2.1 Back-End (David) .......................................................................................... 10
  2.2 Front-End (Alejandro) .................................................................................... 10
3. DEVELOPMENT ................................................................................................... 11
  3.1 Work methodology ......................................................................................... 11
    3.1.1 Requirements analysis ............................................................................. 12
    3.1.2 System design ........................................................................................... 13
    3.1.3 Implementation and tests ......................................................................... 14
  3.2 Problem Statement ......................................................................................... 15
  3.3 Design of the solution .................................................................................... 16
    3.4.1 Registration ............................................................................................... 25
    3.4.2 Login .......................................................................................................... 26
    3.4.3 Creation of a project .................................................................................. 27
    3.4.4 Detail of a project ....................................................................................... 28
    3.4.5 Creation of a group .................................................................................... 29
  3.5 Future lines of work ....................................................................................... 30
3.5.1 Hardware ................................................................. 30
3.5.2 Software .............................................................. 31
3.5.3 Design ................................................................. 31
3.5.4 Logic and security .................................................. 31
4. CONCLUSIONS .......................................................... 33
BIBLIOGRAPHY ............................................................. 34
Illustrations

Figure 1 Blockly interface ................................................................. 8
Figure 2 MakeCode interface ............................................................. 9
Figure 3 Bitbloq interface ................................................................. 9
Figure 4 Gantt diagram .................................................................... 11
Figure 5 Waterfall model ................................................................. 12
Figure 6 System mindmap ............................................................... 13
Figure 7 Current architecture ......................................................... 15
Figure 8 Database scheme .............................................................. 17
Figure 9 File hierarchy ................................................................... 20
Figure 10 Home design ................................................................. 21
Figure 11 Experiment design ......................................................... 22
Figure 12 Forms design ................................................................. 23
Figure 13 User registration ............................................................. 25
Figure 14 Login ............................................................................. 26
Figure 15 New project ................................................................. 27
Figure 16 Detail of a project ........................................................... 28
Figure 17 New group ................................................................. 29
Figure 18 ReteJS framework ......................................................... 30
RESUMEN

El objetivo de BioBlocks 3.0 es la creación de una plataforma web que permita la creación de experimentos biológicos que puedan ser compartidos, permitiendo el uso a un rango amplio de usuarios. Se pretende que la plataforma diseñada sea de versión abierta y que permita especificar los experimentos de manera sencilla, para su posterior ejecución en laboratorios portátiles abiertos controlados por placas Arduino. Este proyecto es la continuación de tres anteriores trabajos realizados por estudiantes de la facultad, con el objetivo principal de la integración de las partes desarrolladas, además del desarrollo de nuevas funcionalidades.

El desarrollo del proyecto se dividió en tres partes principales:

1. La primera parte del trabajo estuvo relacionada con el estudio de los anteriores trabajos y métodos usados, además del análisis de costes en posibles cambios de herramientas.
2. La segunda parte se centró en el desarrollo de la integración de las partes realizadas, mediante el uso de las herramientas Vue.JS en el front-end y Node.JS en el back-end, utilizando MySQL como base de datos persistente.
3. La tercera parte abarcó el añadido de nuevas funcionalidades, centradas en el back-end de la plataforma, y el despliegue de esta en un servidor web.

Durante las primeras semanas, el objetivo fue principalmente el análisis de los trabajos y entornos y la investigación de las herramientas a utilizar. Tales herramientas fueron HTML, JavaScript, CSS, MySQL, Node.JS y Vue.JS.

A partir de la primera semana, las tareas se centraron en el funcionamiento básico de la plataforma. Esto es, inicio de sesión, registro de un nuevo usuario, creación de un menú de inicio, etc. A pesar de funcionar correctamente, se realizó una modificación en la estructura y funcionalidad del proyecto, como, por ejemplo, la lógica de algunas funciones para permitir mayor escalabilidad.

Luego de haber conseguido la integración, las tareas siguientes se enfocaron en el añadido de funcionalidades y vistas con estilos. En esta etapa se desarrolló el buscador, la creación de proyectos, grupos y el diseño base de la aplicación.

A partir de ese momento, opté por investigar sobre la mejora en el pintado de los componentes de la herramienta BioBlocks, pero, tras un periodo de estimación, consideré que suponía un tiempo muy elevado, por lo que se descartó el objetivo.

Por último, las tareas fueron el añadido de lógica a la plataforma, el despliegue de la aplicación en un servidor y la investigación sobre el pintado de componentes, que permita un desarrollo más claro para el siguiente alumno.
ABSTRACT
The main goal of BioBlocks 3.0 is the development of a web platform that allows the creation of biological experiments that can be shared, allowing the use of a wide type of users. It is intended that the designed platform be an open source that allows to create new experiments in a simple way, for its later execution in different portable laboratories that would be controlled with Arduino boards. This project is the continuation of three previous projects carried out by students of the faculty, and its main objective is the integration of the previously developed parts, as well as the development of new functionalities.

The development of the project was divided into three main parts:

1. The first part of the work was related to the study of the previous works and methods used, as well as the cost analysis in possible tool changes.
2. The second part focused on the development of the integration of the performed parts, through the use of the tools Vue.JS in the Front-End and Node.JS in the Back-End, using MySQL as a persistent database.
3. The third part covered the addition of new functionalities focused on the Back-End of the platform and the deployment of the platform in a web server.

During the first weeks, the objective was mainly the analysis of the works and environments and the investigation of the tools to be used. Such tools were HTML, JavaScript, CSS, MySQL, Node.JS and Vue.JS.

From the first week, the tasks focused on the basic operation of the platform. This is, login, registration of a new user, creation of a start menu, etc. Despite working correctly, a modification was made in the structure and functionality of the project, such as, for example, the logic of some functions to allow greater scalability.

After having achieved the integration, the following tasks focused on the addition of functionalities and views with styles. In this stage the search engine was developed, the creation of projects, groups and the basic design of the application.

From that moment, I opted to investigate the improvement in the painting of the components of the BioBlocks tool, but, after a period of estimation, I considered that it was a very high time, so the goal was discarded.

Finally, the tasks were the addition of logic to the platform, the deployment of the application on a server and the research on the painting of components, which allows a clearer development for the next student.
1. INTRODUCTION

During the last years there was an increase in the use and interest of platforms and technologies that allow the construction of elements without the direct help of experts or professionals. Due to this, the platforms that allow a simple use of new technologies or advanced aspects of a field have suffered a great increase of users.

From this concept come the programming platforms based on blocks that help students or people without basic knowledge of programming to the creation of technological projects such as robots or electronic games. That is why the idea of the development of Bioblocks was conceived, a platform focused on the simple use of the field of programming for the definition of biological experiments applied in a portable laboratory.

The first stages of the project were carried out during the three previous years, in which the main objectives were related to the analysis of requirements and the specification of the tools to be used. It should be noted that the current stage is based on the codification of the platform, so it is essential that the developments carried out and the possible problems related to the future requirements development be specified in an appropriate manner.

1.1 Approach

The elaboration of this project is being carried out in the Department of Artificial Intelligence (DIA) of the Informatics Faculty of the Polytechnic University of Madrid. The working group consists of a student and the responsible tutor, focusing the development on web technologies.

Throughout the development of the work, the main objective has been to implement a platform that offers its services on a web server, counting as a main task the connection of the Front-End with the Back-End made in previous projects. This platform will allow the definition of biological experiments through the use of the Blockly block programming environment. With the development of this project, it is proposed to establish a stable platform, which allows the release of new versions in a clear way for the future participants in the project. In addition to this, the fact of deploying the current version of the platform on a server is proposed, so that the changes can be viewed in a more efficient manner, allowing the use of the platform to external users.

To carry out the solution, a study was made of the tools used in the previous works, with the aim of making a stable and rapid integration. The frameworks used will be detailed in the following section. During the realization of the development, the idea of a development for mobile platforms was discarded, since the main tasks will be carried out in personal or work computers.
1.2 Objectives
After reading the work of my colleagues and using the tool for a few days, these were the objectives I saw most necessary:

1.2.1 Back-End
- Back-End integration with the Front-End: this was the main objective of the work, investing most of the time in integrating the Back-End functionality developed in NodeJS with the Front-End developed in VueJS that were carried out by previous project partners. The goal will be the creation of an architecture that offers from the client all the functionalities that already exist on the server side, with the addition of new functionalities. In spite of having functional versions, extensive modifications of these had to be made in order to allow the operation between both parts.
- Creation of a permit system: to be able to add logic to the platform, it was proposed to add certain functionalities to the users, such as allowing project creators to assign other administrators, who have the same permissions as the creator of the project, or give users read and/or write permissions to a project according to the confidence that the user has. In addition to this, the creation of administrator-type users, who have full permissions on the platform, is contemplated.
- Hierarchy of users: since the platform is created so it can be used by a wide range of profiles, three types of users were established: student, teacher and standard, which refers to casual users of the platform or researchers. In the future, the addition of a greater differentiation between each type of user is contemplated, being the current one the impossibility of creating a group as a student.
- Social organization: to allow a dynamic and open use for the users of the platform, the system should allow the creation of a project saved in the cloud that is private or public, as well as being part of a group related to a project.

1.2.2 Front-End
- Algorithm for the components placement: at the time of establishing the objectives, one of the aspects that I considered important was the improvement of the painting and/or placement of the components of the experiment. Despite having an initial development approach, the complete realization of this section meant a very high cost of time, so we decided not to continue with this improvement.
- Implementation of the views: in addition to the integration to be carried out, one of the important aspects was the development of several views on the platform, which shows the essential information for the user. Despite having an acceptable design, this aspect may need to be improved.
1.2.3 Functionality

- Execution of the program: that is, to allow a simple deployment of the program, which allows to work efficiently in the launch of the new versions. To do this, the project hierarchy was modified along with the addition of new parameters when executing the program.
- Deployment of the platform: to allow project participants quick access to the application's visualization, one of the objectives is to publish the current version on a public server maintained by CeSViMa.

1.3 Motivation

The main motivation when participating in the project was the opportunity to work with several web development tools linked to the field of biology, which, despite not having been explored in excess, has allowed me to investigate related aspects.

The subjects with which I have been able to work will be quite positive for my future professional career. The main technologies with which I have worked were Node.js and Vue.js. Despite not being able to work with more globally used tools such as .NET or Java, I think I have been able to learn a lot about the web development environment, being one of the most demanded fields at present.

On the other hand, I believe that the idea will allow a large number of people who want to enter the world of biology and programming to do it in a simple and fun way. The field of computer science is one of those that experiment a greater speed of transformation, reason why platforms like Bioblocks are the ideal solution to this circumstance.

1.4 Used tools

In this section I will detail all the tools that have been used for the development of the project, both in hardware, software and the languages involved in the development of the project.

1.4.1 Hardware

To make this project the use of specialized tools has been not necessary, because the work was based solely on software development. Therefore, the main tool has been the personal computer from which I work. It should be noted that in order to carry out the project, a very advanced team is not required, so the tasks could have been carried out with a low-performance team.

1.4.2 Software

In this section I will name all the software and programming languages used for the realization of the project, allowing the continuation of the work in a simpler way. All the software used is available for free, except Microsoft Word, which requires the purchase of a license.
**NodeJS**
It is an open-source environment based on JavaScript on the side of the server, designed to generate web applications in a fast and optimized way. It is a tool widely used in the present and is used by some wide network platforms such as LinkedIn or PayPal. Being inspired by JavaScript, it allows an easy learning and application for any programmer.

**VueJS**
VueJS is a tool that, like NodeJS, is based on JavaScript. It is a progressive framework that allows you to build user interfaces. This means that it allows to develop simple applications, with the possibility of expanding the functionalities through other plugins.

**MySQL**
MySQL is a relational database management system (RDBMS) developed under a dual license: General public license/commercial license by Oracle Corporation and is considered one of the most popular databases along Oracle and Microsoft SQL Server, especially for web development environments. Thus, MySQL is nothing more than an application that allows managing files called databases.

**HTML (HyperText Markup Language)**
It is the standard markup language that allows the development of pages and web applications on the internet. It is a standard that serves as a reference to connect with the development of web pages in different versions, defining a basic structure and a code for the definition of the content.

**CSS (Cascade Style Sheet)**
It is a style sheet language used together with HTML that allows to give visual and aesthetic qualities to a web page. It is mainly used to present in a more pleasant way the contents in a web page.

**JavaScript**
It is a high level language that allows complex activities on a web page. Along with HTML and CSS, it is the third layer of standards in web development technologies, which allows the addition of dynamic content and interaction.

**Visual Studio Code**
It is a multiplatform code editor that allows several practical functions when working with the code. It supports a considerable amount of programming languages and web programming with HTML, CSS and JavaScript.

**Notepad++**
Notepad ++ is a free source code editor and a Notepad replacement that supports several programming languages. When running in the MS Windows environment, its use is governed by the GPL license. In addition to the previous code editor, Notepad ++ was used for the code editing of the PXT-Blockly part.
**GitHub**
GitHub was used to allow the copy of the project in the cloud. It is a web-based hosting service for version control using Git that allows recovering the previous versions of the project in case of any error in the current functionality.

**Microsoft Word**
Microsoft Word is a computer licensed program for word processing. It was created by the Microsoft company, and comes pre-integrated in the office suite called Microsoft Office. This was the program used for the creation of memory and other documents.
2. STATE OF ART

The Do It Yourself community, or DIY, experienced a great growth during the last years due to the interest to build elements of daily use without the need of advanced tools. This movement has also expanded in the more specialized fields, and there are already spaces on the network where information is shared about the creation of devices such as robots or other types of electronic devices.

In certain areas of the industry, due to the cheapening of the tools and the existence of public platforms that allow sharing content, there has been a great increase in the creation of own products, and something similar begins to occur in the scientific field. Thanks to these advances, both professionals and less experienced users can carry out scientific projects that initially required large resources and were limited to research departments of large companies.

That is why there are already certain web platforms that allow experiments to be carried out in a visual way and without the need for other knowledge such as programming. As for these tools that allow to carry out own experiments, we can name three platforms that have served as reference for the project.

- **Blockly**: being a standard in block programming and the tool that drives languages such as Scratch, Blockly adds an interface that allows programming through graphic blocks, generating the selected code automatically. This eliminates the need to focus on the syntax, thus allowing the use of the tool to less expert users.

![Figure 1 Blockly interface](image)
• **MakeCode**: it is an open source platform that helps progress in programming, providing a block editor, similar to Scratch, and a JavaScript editor for more advanced users. It also provides a web-based simulator, which allows to visualize the operation of the programmed code.

![Figure 2 MakeCode interface](image1)

• **Bitbloq**: another of the tools to take into account is Bitbloq, since the concept of the tabbed division between the Software and Hardware part was based on it. Unlike MakeCode, Bitbloq offers a simpler interface and less prior knowledge required.

![Figure 3 Bitbloq interface](image2)
Seeing the lack of alternatives available on the web, it was decided to develop Bioblocks, motivated by the lack of open source platforms that allows experiments in portable laboratories without the need of programming knowledge.

After a bad development experience in the Bitbloq environment, during the previous semester the investigation consisted of an alternative solution and the design and programming of the different functionalities of the application. After a period of testing different platforms, the use of PXT-Blockly was proposed, a fork created from the PXT component by the Microsoft MakeCode team, which allows, in addition to other functionalities, to generate Arduino code in a simple way.

Therefore, at the beginning of the work there were two main branches of development of the project, which, despite not being integrated between them, already had an initial skeleton of operation. In the next section, I will summarize the existing work of my previous colleagues to obtain a more complete vision of this project.

2.1 Back-End (David)

David's work was mainly focused on the development of the application's Back-End, developed with NodeJS and the ExpressJS framework and interacting with a MySQL database. The main functionalities were those related to the logic of the program, such as:

1. Registration of a new user
2. Log in to the system
3. Create a new project/group
4. Search of public projects
5. Add a user to a project
6. Add a project to a group
7. List of projects/groups

To allow a correct operation with the Front-End part, during the project I made certain modifications in the logic of the NodeJS code, such as the login, since there were certain problems when making the API calls from the part of the Front-End. In addition to this, I also included certain changes in the database, which will be detailed in the following sections.

2.2 Front-End (Alejandro)

Alejandro's part was focused on the development of the Front-End composed of PXT-Blockly, which would provide programming logic with blocks, PaperJS, a framework responsible for the rendering of the hardware part and a simple interface developed with VueJS. The main part of the project was related to the design of the hardware blocks that form the experiment and the connections of these to the Arduino board. In addition, a secure environment was created that would allow the addition of extensions by users.

Regarding the Front-End part, I made slight modifications in the PXT-Blockly part to allow the saving of the projects and implemented the design of the VueJS interface.
3. DEVELOPMENT

3.1 Work methodology

During the first weeks of work, the work plan to be followed was established according to the objectives set. The first task consisted in the documentation of the objectives and the general description of the work to be carried out, established during the first working sessions with the tutor.

The list of tasks to perform was:

- Learn the programming environment PXT-Blockly and the Blockly language.
- Learn Node.JS, Vue.JS, JavaScript and MySQL.
- Connect the Back-End with the Front-End.
- Develop new functionalities in the Back-end.
- Software testing and validation.
- Writing of the memory and preparation of the oral presentation.

In addition, a high-level Gantt Diagram was designed that served as a guide to follow for the development of the project. In general, the diagram was followed quite accurately.

As can be seen, the planning consisted of four main phases. Each of these can be broken down into several parts but since not having an exact estimate of the workloads in each part, it was decided to make an approximation of the development time, establishing a high-level diagram and dividing the tasks after a contact with the environment.
In order to organize the work efficiently, the work was carried out following the waterfall development, which orders the stages of the development process in such a way that the beginning of each stage must await the completion of the previous stage. At the end of each stage, a review was conducted to determine if the project was ready to advance to the next phase.

![Waterfall model](image)

Figure 5 Waterfall model

One of the main advantages found when working with this model was the clear planning and development that the project had, allowing each phase of development to be clear and not present many difficulties.

Next, the tasks carried out in each phase of the development will be detailed.

3.1.1 Requirements analysis
During this first phase of the project, the study of the proposed problem and the establishment of requirements for the development of the work were carried out. This allowed an adequate planning, allowing the elimination of errors or unnecessary problems.

This phase consisted mainly in the study of the tools with which the previous projects were developed, as well as the discussion of the current state of work and the most important objectives with the tutor of the project.

After the enumeration of the requirements to be developed, the analysis phase was completed in order to proceed to the design phase. However, during an advanced stage of
the work, the initial analysis was revised, eliminating the improvement of the hardware visualization.

3.1.2 System design
During this phase, the solutions to be developed were designed to meet the requirements established during the previous phase. The first task was the fragmentation of each objective in simpler steps, allowing the work to have more concrete and simple objectives to analyze.

Figure 6 System mindmap

The figure shows the steps to be taken to achieve the requirements established during the first phase of work. However, the tasks shown only offer a high-level view of the work done, so it is worth mentioning that these objectives are at the same time divided down into ones of less complexity.

After studying and establishing the specifications required for the requirements of the first phase, the next step was to prepare the design of the system and establish the necessary hardware and system requirements to carry out the development. This phase allowed to set up a general architecture of the project and to obtain a clearer vision of the work flow.
3.1.3 Implementation and tests

From the previous phase, the vision of the project during the next two phases becomes low-level, focusing the tasks on the writing of the code necessary to meet the objectives set and the study of the obtained solutions.

These two phases have been applied during the same interval due to the limitation of development time, so the execution of both parts was not carried out in an extensive manner. However, during the development of both phases, an attempt was made to comply with the quality and reliability standards in the program.

During the implementation phase, one of the points to take into account was the creation of a clean code understandable by other programmers. For this, an attempt was made to maintain a clear project structure and a modular programming, which allows the modification of the parts in a simple manner. In addition, during the creation of the code, the JSHint tool was used, a static code analysis tool that verifies if the JavaScript source code complies with the coding rules and does not present potential errors.

In addition to the integration of the units that were developed in the system, during the testing phase each unit was carried out separately and a final test of the system integrated completely was performed to detect possible failures in the development. On the other hand, a small part was invested in the use of the platform by external users, which allowed to find failures that would have been difficult to recognize for a person who knows the default operation of the application.

3.1.4 Maintenance

To maintain the correct functioning of the platform, it is necessary to establish a period of review and maintenance of the software developed, since, despite establishing a phase of design and implementation of tests, it is most likely that unknown faults arise for the external users.

For this a good practice that can be applied is the fix of certain errors in the current version and the release of a version with the corrected problems. However, this phase has not yet been executed in the development of this project.

During the following sections the different technical aspects that have been taken into account to carry out the project will be detailed, along with the list of full functionalities of the platform and the study of possible future lines of development.
3.2 Problem Statement
As mentioned in previous sections, the objective of the project is based on the
development of a platform that allows any type of user, such as students or scientists, to
define and carry out experiments without the need for advanced equipment. In addition
to this, a tool that generates the Arduino code that results from the experiment and the
protocols necessary to perform such experiment must be offered.

The solution to this problem is born with Bioblocks, an open source platform that allows
to define experiments in a simple way by programming by Scratch type blocks. In the
previous developments, the tools to be used were specified and the architecture of the
system was designed, dividing the work into two main levels. To this architecture was
added the connection element between the Front-End and the Back-End, keeping the other
elements unmodified.

![Diagram](image)

*Figure 7 Current architecture*

In this case, there is a Back-End developed in MySQL, a familiar tool for students during
the course of the degree and which is compatible with the desired design structure. On
the other hand, the connection from the Front-End is made through the ExpressJS
framework, which allows to accept calls to the NodeJS server in a fast and clean way.

The Front-End part is composed of the VueJS framework and the Element component
visualization library. As previously mentioned, the part related to the experiments has
been developed with PXT-Blockly, which allows the logic in block programming, and PaperJS, in charge of rendering the canvas of the experiment. On the other hand, the Axios library is responsible for making HTTP requests to the server.

In this phase of the project, integration must be carried out between the two branches of work developed. The objective is therefore to allow communication between the frameworks of NodeJS and VueJS, in addition to the connection with the MySQL database. To do this, you must find the tools and libraries that allow a connection without errors between Front-End and Back-End, detailed in the next section. On the other hand, the login and registration of the user should be treated, keeping the information of the client's session.

3.3 Design of the solution
In this section, the solutions found to cover the needs of the project will be defined, in addition to the high level designs of the application proposed in the early phase of development. This stage has been the most important during the project and in which more time has been invested to be able to guarantee a base of stable development in which to be able to add new changes in a simpler way. Despite not having to carry out research on the tools to use because they are familiar in my field of work, a previous study was necessary on the viability of the current and future components in order to ensure the completion of the project without the need to apply considerable changes.

In general, both the tools of NodeJS, VueJS, and MySQL offer the fundamental functionalities to be able to work with relative simplicity around the objectives that are proposed. It is true that there are certain problems that still need to be studied, such as the simplification in the hierarchy of files and libraries, an aspect in which I tried to work without finding specific solutions, or the communication between the environment of PXT-Blockly and VueJS, which was solved, but in which it is possible that limitations are found.

Once the problems and the tools that will be used in the project are listed, the established objectives were designed and programmed. As explained in previous sections, the main objective is the connection between the Back-End and the Front-End, so this was the starting point of the development. After having the aspect of the connection resolved by using the Axios library, the next step was the design of the pages that the application will have. To do this, I started making sketches of the main views that would be shown, in order to have a reference on which to start programming. Next, the main screens that were designed will be shown.
3.3.1 Database

In this section the current structure of the MySQL database will be shown, detailing the changes applied in the new version. During the development some sections had to be redesigned to allow the addition of new functionalities. However, the architecture developed in the previous project has not been modified, the only changes being those related to the creation of new fields. The scheme that the current database has is quite simple, so in the future it is possible that new objects must be added or even redesign the structure to allow the application of new functionalities.

The changes applied were around the users, projects and groups created, due to the need to differentiate the types of existing users, to be able to save a project in the cloud and the possibility of identifying the creator of a group, so that the user only visualizes his own groups. For this, the user_type field was added in the user object, the owner field in the projects_group object and the xml_file field in the project object.

Figure 8 Database scheme
As can be seen, there have been no major changes in the database except the fields mentioned. One of the main reasons why it was decided not to add more elements to the database was the processing of requests in the controller of the server, trying to make the minimum number of calls to the database possible. Next, the existing objects in the database and the purpose of these will be defined.

- **user**: in this object the main information of the existing users will be saved. In the current phase of the project, only the information regarding the credentials and the type of user is saved, but ideally in the future other fields should be added to personalize the experience to the users, such as the research department, the development region, etc.

- **user_belongs_project**: this object serves as a link between the object that stores the information about a user and the object that stores the different existing projects.

- **project**: in this object the essential information about a developed project will be saved. Currently the object has the identifier of the creator of the project and the code that generates the experiment, being sent to the PXT-Blockly environment to visualize the components. This object also has possible additions, such as the identifier of the project image (saved in the server) or the type of motherboard to run the experiment.

- **project_belongs_group**: like **user_belongs_project**, this object relates the existing projects to the development group in which they are located.

- **projects_group**: this last object saves the main information about the groups that are created.
3.3.2 Middleware

This part of the development is the one that has more time invested, due to the connection between the two existing levels of development. In this section I will describe the main functionalities created, in addition to the problems encountered throughout the development.

In terms of authentication, the first tool with which I tried to get the session started was JWT, a tool that allows the generation of a token assigned to a user. This allows the user's session to be saved for a set period of time, after which the user is deleted and is asked to authenticate once more. Despite working in certain circumstances, this solution caused quite a few problems when integrating it with the Front-End and the Back-End. The second tool with which I tried to develop the login was PassportJS. This library was used in the previous work, but due to certain problems in the beginning of the project, it was left as a second option to investigate. PassportJS works in a similar way to JWT, storing sessions created by users during a set period. After the user has been inactive for more than the time limit, authentication will have to be re-enabled to access the system. This library is integrated with MySQL, so that the search of the users is done in the database.

As discussed in previous sections, calls from the Front-End were made with the Axios tool. This allows to execute HTTP calls to a server, returning the response object that is generated after the call. This library has been used to execute all the dynamic functionalities in the platform, such as the login, the registration of a user, the list of projects, etc. After reaching the server, this request was sent to the corresponding module, either data access, creation, modification or deletion of a certain data.

Finally, the connection and calls to the database were made through the MySQL module, which allows executing requests to a database by providing server credentials. The requests usually return to the client two different values depending on the result that is returned by the database, differentiating between a correct call and another with errors in the parameters.
Finally, the hierarchy of files that was finally implemented to integrate the two levels of development will be shown.

The *api* folder contains all the functionality related to the Back-End developed in NodeJS, being `server.js` the file to be executed to deploy the server. On the other hand, the *src* folder contains the part related to the Front-End, containing the libraries, the Back-End connection module and the displayed templates. It should be noted that the directory does not have the folders related to the PXT-Blockly environment, which are located at the same level as the project folder.
3.3.3 Views

- Start: the main objective of this page is the listing of the most recent projects and groups of the user. In addition, the template has a basic navigation menu that allows the end of session, navigation to other views and a search engine that serves to find public projects created in the community. This menu is present in every view that the application has, to provide an easier use of the platform.

*Figure 10 Home design*
- Detail of the project: the goal of this view is showing the PXT-Blockly component that allows the creation and modification of the experiment components. In addition to the view of the experiment, a menu was added to allow basic functionalities regarding the project, such as saving or sharing the project with another user.

*Figure 11 Experiment design*
• Forms: the views to be able to create new projects and groups are very similar, consisting of a descriptive title and the fields to be filled out. In the beginning these fields are only text fields and selection boxes, but in the future the addition of files related to the experiment or reference images could be considered.

*Figure 12 Forms design*
3.4 Functionalities

In this section the functionalities developed throughout the work will be explained. These are mainly the information listing from the database and the creation of new instances in the server. This section will also provide some possible improvements that could be made to the components which do not involve excessive time spent and provide a better user experience.

The platform includes certain features that are not included in the following illustrations, because they do not have the most importance within the views. These functionalities are:

- **Save a project:** this functionality appears in the top navigation menu that contains a project. In order to save a project, you must be the creator of that project or have administrator permissions.
- **Settings of a project:** within this section you can modify the characteristics of the project, such as the name or visibility. Like the project saving, the user must have administrator permissions or have created the project to be able to access the settings.
- **Cloning a project:** in case of not having administrator permissions in an external project, the user will have the option to clone the project in their local directory. This will allow the user to apply the appropriate changes to the experiment without modifying the original project. After selecting this option, the application will redirect the user to the form of creation of a project, with the difference that the experiment will count from the beginning with the components of the cloned experiment.
- **Sharing a project:** in case that the user decides to add a user as administrator of a project, he will have to enter the user name. Initially the user names were not unique, so it was changed to allow the operation of this feature with no possible errors.
3.4.1 Registration
To be able to register a new user in the system, the functionality implemented in the previous work was used, which performs the insertion in the user object with the collected fields. The data sending is made by inserting in a variable the different elements of the form and sending it to the server, but not before checking the correct confirmation of the password. The fields to fill in are the name, username, email, the type of user and password.
3.4.2 Login
To authenticate in the platform, the tool mentioned in previous paragraphs PassportJS was used. With it, the user's credentials are collected and searched in the database. If there is a user that matches the given one, the user will be authenticated. To be able to access the platform, it is necessary to authenticate in the system, so from the beginning of the session all the application's functionalities are offered. As mentioned above, this procedure is repeated every time the user does not connect for more than 24 hours. One of the possible future functionalities around the beginning of session is the possibility of recovering the account in case of forgetting password. For this, the user would have to verify his account with a message sent to his email address, and insert his new password in a form.

![Figure 14 Login](image-url)
3.4.3 Creation of a project
When creating a project, you must specify the project name and the visibility for other users. In addition, in case of cloning an existing project of another user, the data of the associated experiment will be copied, allowing the modification of the cloned project in a personal environment. However, the functionality offered by cloning other user's projects is only available for those that are set as public.

![New project](image)

*Figure 15 New project*
3.4.4 Detail of a project
After the creation of a project, a view is offered to visualize the details. First, the screen of the experiment associated with the project is displayed, in which you can make modifications to the existing experiment or delete the selected components. In the upper level there is a navigation menu that allows you to save the experiment that is being edited, change the project settings, both the name and visibility, share the project with another user given its user name or delete that project. Regarding the saving, the option of an auto-save was proposed a considerable number of times, but since there is the possibility that the user only wants to modify the experiment without the intention of saving, the option of manual saving was chosen. One of the possible improvements around the experiments that I thought was a version control, which allows to recover to previous version of the experiment in case of failures in the current one.

Figure 16 Detail of a project
3.4.5 Creation of a group
When creating a group, no more fields are offered than the group name. As reviewed in previous sections, the change of the hierarchical structure and the addition of new features will bring new fields to fill in this form. One of the aspects that were not detailed, was whether the group should be associated with a project or a user, since, in case of opting for the second option, such fields as the participants or the visibility of the group towards other users could be defined.

Figure 17 New group
3.5 Future lines of work
During the realization of the project improvements and implementations of the program have been emerging, both in meetings with the tutor and during the development of the application. As it has been observed, my contribution focused mainly on the development of a stable platform, which allows work in both branches in a simpler way, improving efficiency and allowing a better integration of changes.

In this section I will explain the possible improvements for this project based on the work done and the work of my previous colleagues.

3.5.1 Hardware
Regarding the hardware part, I believe that a redesign of the visualization of the components and the improvement in some functional aspects should be carried out.

In terms of the hardware painting, it is possible that the PaperJS framework offers a low level solution. To achieve a good design and improve the user experience with PaperJS, the implementation of numerous aspects must be carried out, so the ideal solution would be to base the painting of the components in work libraries based in nodes such as ReteJS, a JavaScript framework that allows a visual implementation of interconnected nodes with labels.

Another aspect to take into account when displaying the components is the definition or use of an algorithm that allows its placement. When it comes to displaying numerous pieces of hardware, the platform should make a distribution of the parts around the Arduino board. Regarding this improvement, I think there are different approaches such as the use of any existent algorithm that performs the calculation of the positions of different elements or a simple implementation of the distribution of the pieces. In case of
problems in this aspect, a simple solution would be to allow the user to manually place the components, as can be found in the Bitbloq platform.

One of the problems found by my partner Alejandro was the identification of the components found in the hardware part. This is necessary if you want to use the same component to perform different executions, avoiding the creation of a different component. To arrive at a solution, the separated definition between software and hardware was chosen, associating each block with specific variables of the experiment. To be able to execute this solution, typed variables combined with event-based mirroring could be declared as shown in this demo.

3.5.2 Software
To allow a personalized use for each user, a functionality that allows the addition of custom blocks should be implemented. This aspect is considered in the PXT-Blockly tool, so at first great difficulties to implement the solution should not be found. The only problem that could arise would be the addition of a specialized logic of the component, so the viability of this characteristic should be discussed.

On the other hand, one of the aspects discussed with the tutor was to provide the series of protocols necessary to carry out the experiment. This would allow the user, after loading on the Arduino board the generated code, the correct execution of the experiment.

3.5.3 Design
Despite having an acceptable interface, to ensure a comfortable and intuitive use it is possible that certain aspects of the platform have to be re-designed. In general, I think that many changes should not be applied, but rather to apply certain styles that improve the overall visualization of the page.

For this I believe that the ideal solution is to load external libraries of styles, recommending above all Bootstrap, a free and open source CSS framework aimed at a mobile web development and responsive Front-End. It contains design templates based on CSS and JavaScript for typography, forms, buttons, navigation elements and other interface components.

However, I do not think that the current development should take into account this objective but rather work on it during a more advanced phase of the project.

3.5.4 Logic and security
Unlike the previous points, in these two aspects I have a high level vision so there are no specific improvements to name. However, I think it is a good idea to begin to set objectives regarding the logic followed by the platform and the security in the execution of the program.

Regarding the logic of the application, certain validation rules should be added for the creation of projects and groups, that is, to set a series of limitations for each type of user
when using the platform. Besides that, I think it would be a good idea to propose a new scheme when sharing projects and adding different sharing modes.

On the other hand, in order to guarantee a successful execution without errors, a study of the security of the environment must be carried out, which is why it should be taken into account in the future.

However, and like the previous section, I believe that these requirements should be treated in a stable and future version of the platform, since the main objective should be to offer the user a complete environment for the creation of experiments first and complying with the needs of the majority of users.
4. CONCLUSIONS

In general, the project has been very interesting for me, and it has allowed me to gain some knowledge of the field of biotechnology. Despite not having reached all of the proposed objectives, I consider that the project is in a good development point, so it would be ideal to continue developing it.

As for the planning, I think I have spent an excessive period in the analysis of the objectives, since I had to carry out a study of the previous works to analyze the possible improvements. However, I have not had too many problems during the development of the platform, so I have been able to streamline the process and implement all the functionalities.

Despite this, during the final phase of development I had some problems with the deployment of the platform on a public server, due to a strange failure when accessing the system. This meant that the work did not have such a continuous development due to the work break during the server's fix period.

Regarding the hardware part, I would have liked to invest some time in the visualization of the components, although I think it may be better to work on this aspect within more specific objectives.

Finally, if a solid foundation in the functionality of the experiments is achieved, the platform could be adopted very quickly. I think Bioblocks is a very interesting project with which I have learned a lot and has very good objectives, so I hope that it will continue to provide work.
BIBLIOGRAPHY

<table>
<thead>
<tr>
<th>Firmante</th>
<th>CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fecha/Hora</td>
<td>Sun Jun 02 11:30:06 CEST 2019</td>
</tr>
<tr>
<td>Emisor del Certificado</td>
<td>EMAILADDRESS=<a href="mailto:camanager@fi.upm.es">camanager@fi.upm.es</a>, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES</td>
</tr>
<tr>
<td>Numero de Serie</td>
<td>630</td>
</tr>
<tr>
<td>Metodo</td>
<td>urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.shal (Adobe Signature)</td>
</tr>
</tbody>
</table>