Augmented Reality tool for field trips

Final Degree Project

Escuela de Ingeniería Informática – Universidad de Oviedo

Director: Víctor Manuel Álvarez García
Co-director: Juan Ramón Pérez Pérez

Author: Roberto Pérez Sánchez

2019

Gratitudes

To my parents, for always been there supporting me during this 4 year journey and for giving me the possibility to study what I always wanted to. Thank you for letting me organize myself and always support my decisions, as well as knowing when to leave me space for solving my problems and not stressing me out when I was already stressed for my own.

To my supervisors of this project, Víctor and Juan Ramón, for quickly understand my point of view and approach about Ariane and letting me take important decisions about the project. Thank you for all the support and knowledge that you both have given to me during the development of this project.

To my fellow workers in GooApps, where I have learned all the basics needed in order to develop this system, for all the help, support and flexibility given to me in order to finish this project.

To all my friends in Banda del Geriátrico for making the degree classes and moments more enjoyable, and for all the experiences lived together during these 4 years. Thank you.

Abstract

During the past few years, the use of augmented reality as an educational vehicle has increased considerably. This modern technology, combined with the realization of field trips in school in order to explore the nature and to disconnect from the studies, can handle a lot of advantages in both the preparation for the field trip and its execution.

Ariane is a system composed by a web application for the field trip management by teachers, and a mobile application where students can perform the different tasks contained in the field trip.

Ariane provides different activities and tasks for the students. Some of them might remember of the conventional field trip activities like test or explanations about a topic, but by digitalizing this process, and incorporating augmented reality features to the mix, loads of new possibilities emerge in order to prepare the activities, as well as storing student's feedback about the tasks in order to make an analysis about its execution.

Keywords

Augmented reality, field trip, web application, Symfony, React Native, student, mobile application, API

Table of contents

Illustration table

1. Introduction

1.1. System description

The idea behind this system (**Ariane** from now on) is to change the traditional way of performing the different tasks of a field trip in high school with a modern technology such as **Augmented Reality (AR)** in order to make the trip more interesting and engaging for the students, as well as collecting all the answers and feedback of the activity in the web application for the teachers that organize the field trip.

This tool for field trips will consist of two different modules: a **mobile application** where students will perform and answer different activities or tasks related to the field trip using augmented reality as the educational vehicle, and a **web application** where teachers will be able to create the different field trips and activities, as well as list the different answers and logs created form the mobile app. These two applications will communicate via an **API Rest**, that will be used in order to consume the data provided by the web application in the mobile application.

1.2. Technologies

The technologies used for the development of the system are the following:

- **Symfony**: It is a PHP framework designed for the development of web applications using the Model-View-Controller (MVC) design pattern. It will be used for the design and development of the web application in order to manage all the data of Ariane.
- **MySQL**: It is an open source relational database management system (RDBMS). It will be used to store all the data related to users and field trips.
- **React Native**: It is a framework for building native apps using React and JavaScript. It allows the developer to build iOS and Android apps with the same code. Nowadays, thousands of famous apps are using React Native, such as Facebook, Instagram or Skype.
- **React**: Also known as ReactJS or React.js, it is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies.
- **Amazon Web Services (AWS)**: It is a subsidiary of Amazon that provides on-demand cloud computing platforms to individuals, companies and governments. It will be used in order to host the web application and the database.

These technologies will be further explained in chapter 3: Tools and Technologies.

1.3. Alternatives

During the first sprint of this project (later explained in chapter 4 “Planning and budget”), an alternatives study has been made in order to select the technologies used in the development of the system.

1.3.1. Mobile application

1.3.1.1. Ionic

Ionic is an Angular framework for developing hybrid applications. It uses Angular with visual native elements, wrapped with Cordova, and classes for easily integrating Cordova plugins.

Ionic was selected as one of the alternatives for the development of the mobile application of Ariane because of their main advantages:

- Multi-platform development
- The use of Angular creates a powerful SDK, for building rich and robust applications
- The framework focuses on HTML, CSS and Javascript, which enables quick development, low cost and minimal maintenance
- Personal experience using this framework for developing mobile applications

Nevertheless, this development framework has also some disadvantages:

- In-app performance is not as fluid as native applications
- Ionic does not build a native app from the code, it renders the app using a WebView in the device, so its integration with augmented reality features is very limited, making nearly impossible the development of Ariane using Ionic

1.3.2. Web application and database

1.3.2.1. NodeJS and MongoDB

NodeJS is an open-source, cross-platform Javascript run-time environment that executes Javascript outside of a browser. Due to the previous experience developing a web application and an API REST using this technology in the course “Sistemas Distribuidos e Internet” of the degree, NodeJS was the first alternative proposed in order to develop the part of the web application of Ariane.

The use of thie technology would have as a consequence the use of MongoDB as the database of the system. MongoDB is a NoSQL document-oriented database that uses JSON-like documents as a schema. Being a JSON based, non-relational database will offer us very simple operations in order to develop the API REST, but it will make it more difficult to design a complex database than using a relational database.

The option of NodeJS and MongoDB as the pair of web application and database was rejected because I started working in a mobile app development company, GooApps, which uses Symfony and an SQL-like database for all the web applications and back offices of their mobile apps. In GooApps I learned the basics of its use and I decided to take NodeJS and MongoDB to a side, choosing Symfony and an SQL database instead. The different advantages and usage of Symfony will be explained later in chapter 3 “Tools and Technologies”.

1.3.3. Solution

As explained previously, for the web application development, Symfony was the chosen technology. For the development of the mobile application, I finally chose React Native, more or less for the same reason. I started using this technology in GooApps and I came across different libraries that supported augmented reality using this technology, so the development of this features in the final mobile app would be easier.

The more detailed reasons, functionality and advantages of this technologies are explained in chapter 3, “Tools and Technologies”

1.4. Chapters

1. **Introduction**: Initial chapter with a brief description of the system and the technologies used, as well as the alternatives found and an explanation of the different chapters of this memory.

2. **Theoretical aspects**: This chapter will contain information about Augmented Reality and its uses for educational purposes, as well as the explanation of the ‘field trip’ concept and the advantages of digitalizing all the process of it.

3. **Tools and technologies**: In this chapter, the different technologies and tools used for the development of the system will be explained, as well as why they have been chosen and how to use them.

4. **Planning and budget**: This chapter will explain the project management, including the methodology used and the different tasks performed. It will also contain a summary budget of the system.

5. **Analysis**: This chapter contains the system requirements, as well as the identified use cases and the different scenarios related to the use cases.

6. **Architecture and design**: This chapter will explain the architecture used in this project and its associated diagrams, as well as the database and interface design.

7. **Implementation**: In this chapter, the project organization will be explained, including its classes and components.

8. **Tests**: The different test cases, results and techniques used for the testing of the system are explained in this chapter.

9. **System manual**: Handbook about the functionalities of the system.

10. **Detailed budget**: A more detailed budget is explained in this chapter.

11. **Extensions and conclusions**: Possible extensions for the system and the conclusions made after the development of the project.

12. **References**: This final chapter contains the different sources of information used for the development of the system and the memory.

2. Theoretical aspects

2.1. Augmented Reality

Augmented Reality (AR) is an interactive experience of the real-world environment where the objects that reside in the real-world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory. The primary attraction of AR is how it brings components of the digital world into a person's perception of the real world, not as a simple display of data, but through the integration of immersive sensations that are perceived as natural parts of an environment.

The beginning of AR can be traced back to 1968, when Ivan Sutherland developed the first head-mounted display system. The first functional system that provided mixed reality experiences for users were invented in 1992 by Louis Rosenborg at the U.S. Air Force's Armstrong Laboratory, and the first commercial augmented reality experiences were used largely in the entertainment and gaming businesses, but nowadays it has loads of application fields, getting more and more relevance in the modern technology world. AR has transformed the world of education, where content may be accessed by scanning or viewing an image with a mobile device or by bringing immersive, markerless AR experience to the classroom.

AR and the internet finally joined in the early 2000s, but it was not since 2009, with the 'smartphone revolution' that interest in this technology raised. Nevertheless, probably the event that put AR into everybody's mouth was the launch of Pokemon Go in 2016, as it became a more friendly and well-known technology for everyone. During 2016, the total investment in augmented reality and virtual reality reached €925 Million, and shows no signs of slowing down.

But what is its difference with Virtual Reality (VR)? They sound similar, and as the technologies develop, they bleed over into each other. However, they are two very different concepts, with characteristics that really distinguish them from one another.

VR headsets completely take over your vision to give you the impression that you are somewhere else. It can be a game, a 360-degree video, or just the virtual space of the platform's interfaces. Visually, you are taken wherever the headset wants you to go: the outside world is replaced with a virtual one. Two of the most popular items that provide Virtual Reality are the Oculus Rift and the PlayStation VR headsets.

Whereas VR replaces your vision, Augmented Reality adds to it. The technology is designed for completely free movement while projecting images or other visual or auditive components over whatever you look at, mainly smartphones or AR glasses. These components that are displayed can be as simple as a data overlay that shows the time, to something as complicated as holograms floating in the middle of a room.

2.2. Field trip

According to formal definitions, a field trip is “journey by a group of people to a place away from their normal environment”. They are widely used in education as a way of providing students with experiences different from everyday activities, such as going camping with teachers and their classmates in order to disconnect from lessons and exams, while doing group activities and research of the natural environment.

This trips are structured in 3 phases:

- **Preparation**: Both teachers and students learn about the destination before hand
- **Activities**: They are performed during the trip and they could include lectures, tours, worksheets, videos and demonstrations
- **Follow-up activities**: They are activities and discussions occurred in the classroom after the field trip in order to obtain feedback from the students

Field trips also have loads of advantages for the students. They offer students the opportunity to explore new places and environment, taking a break from classroom lectures and awaking student’s interest to learn new things. They also improve the personal relationships between classmates and also with teachers. Students get the opportunity to interact with one another in a more informal, natural and relaxed way. Finally, and one of the most important advantages of this type of activities, is that students learn by actually doing a hands-on experience. They learn effectively since their senses are totally focused in what they are doing, more than it functions in a classroom setting. Field trips offer the opportunity to learn in a practical way, which is really difficult to perform in a school classroom.

3. Tools and technologies

3.1. Web Application and API

3.1.1. Symfony

As briefly explained in chapter 1, **Symfony** is a PHP framework designed for the development of web applications using the Model-View-Controller (MVC) design pattern. Its main goal is to speed up the creation and maintenance of web applications, trying to replace repetitive code, for example the listing, creating or updating actions. It offers loads of templates for an easier development, so you only have to code functionality outside the box. It was created in 2005 by Fabien Potencier and It has been updated regularly since then.

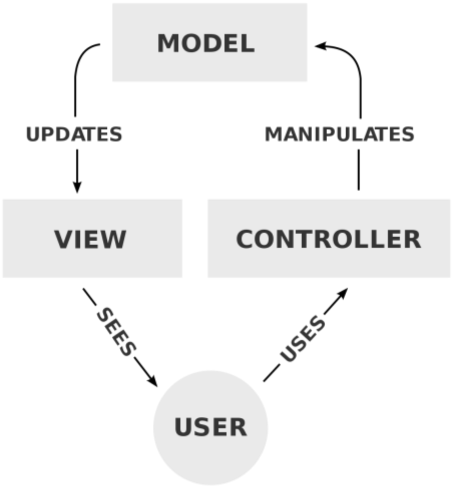


Illustration 1. Model View Controller (MVC).

The main advantages of Symfony and the reasons because it was chosen for this development are:

- **Extremely flexible**. You only have to install what you need and not all framework.
- **Expandable**. According to Gihub, it is the most active PHP project.
- **Easy to use**. There exists a huge amount of information and online tutorials about Symfony, so any advanced user can learn quickly the basic concepts about it.
- **Fast and low-memory consuming**. It has been developed for offering high performance applications, being one of the fastest frameworks while consuming less memory than others, like Ruby On Rails or Django.

Symfony is based in the Bundle System. A bundle is similar to a plugin in other software: it encapsulates a single feature in a set of files. Each of these parts contains everything related to that feature: PHP files and classes, templates, stylesheets, tests, Javascript files and everything that is needed. This structure gives you the flexibility to use prebuilt features packaged in third-party bundles or to distribute your own bundles. In this case, there exists 3 bundles in this application:

- **AppBundle**: where all the entities and base functionality of the web application are created and treated by the user.
- **ApiBundle**: where the API REST methods are created in the different controllers
- **Application**: where the user, groups and media entities are created and treated

3.1.2. Amazon Web Services

Amazon Web Services (AWS) is designed to host quickly and securely our applications in a pay-as-you-go way. This means that every hosting of an application has a free tier, which is 750 hours of use per month for a year. In this case, the web application and the database are deployed in an EC2 and RDS instance respectively.

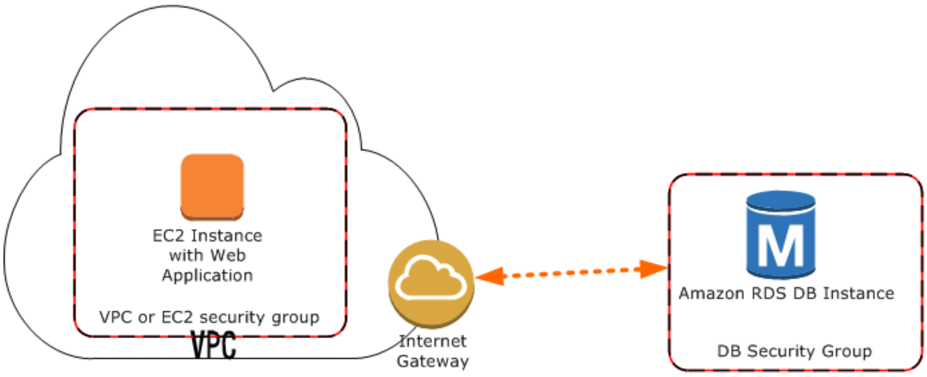


Illustration 2. EC2 and RDS connection.

- **Amazon Elastic Compute Cloud (Amazon EC2):** It is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Its main advantages are the reliability of the system, its security and its pricing, as it is free as we have seen before.
- **Amazon Relational Database Service (RDS):** It makes it easy to set up, operate, and scale a relational database in the cloud. RDS is available on several database instance types and provides six different databases management systems to choose from: Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database or SQL Server. In this case, we will choose MySQL.

The connection to these two systems will be made with an SSH connection for the EC2, and using the program MySQLServer for the RDS.

3.2. Mobile Application

3.2.1. React Native

React Native is a framework that allows developers to implement native apps for mobile devices using Javascript. The developed apps have direct access to all the native APIs and views that operative systems offer, so the user experience and performance are equal to a native app. It was launched in March 2015 by Facebook. It is a relatively new framework, but it is being used by important applications like Facebook, Instagram, Pinterest and Skype.

React Native is a set of React components, where each of them has its equivalent in views and native components. For example, the TextInput native component has its corresponding component in React Native that can be imported in the Javascript code and used as a React component. This process is composed by three main parts:

1. Native modules/code: They are the necessary modules (iOS and Android) in order to not write native code while programming a React Native app.
2. Javascript VM: It is a Javascript Virtual Machine that will execute our Javascript code.
3. React Native Bridge: It is written in C++/java and is responsible of communicating the native and Javascript threads.

The final React Native architecture would be something like this:

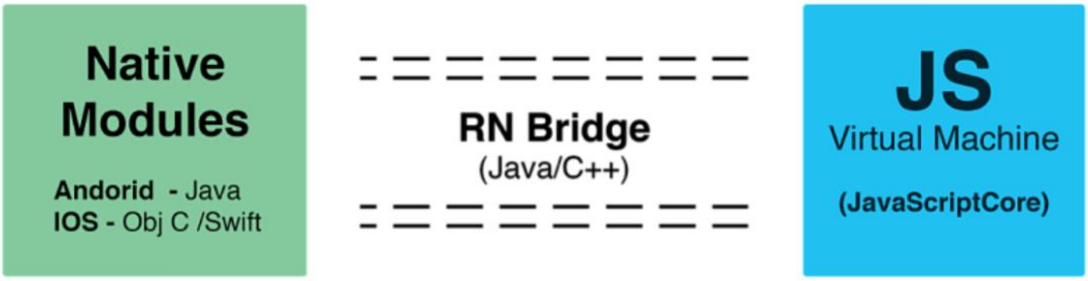


Illustration 3. React Native architecture.

The main advantages of developing a mobile application using React Native are:

- React Native helps in building multi-platform mobile apps
- Saves time and cost comparing it with a native development
- A React Native app ensures speed and agility like if it was a native application

On the other hand, some minor disadvantages of this framework are that Facebook's long term commitment to it is not clear or the Javascript no support for decimals. However, these characteristics do not affect to the development of Ariane.

3.3. Development

3.3.1. JetBrains (WebStorm and PhpStorm)

JetBrains is a software development company whose tools are used by software developers and project managers. It offers a large amount of development environments (IDEs) for the different programming languages like Java, PHP, Javascript or C++. In this case, only two of them will be used:

- WebStorm: IDE for Javascript in order to develop the mobile application
- PhpStom: IDE for PHP in order to develop the web application

Both of them offer loads of functionality and shortcuts for the development and execution of the applications, as well as Git integration, being the developer able to commit, push and update the code directly from the IDE, without having to interact with Git via the computer terminal or via other applications like GithubStudio or GitKraken.

3.3.2. GitHub

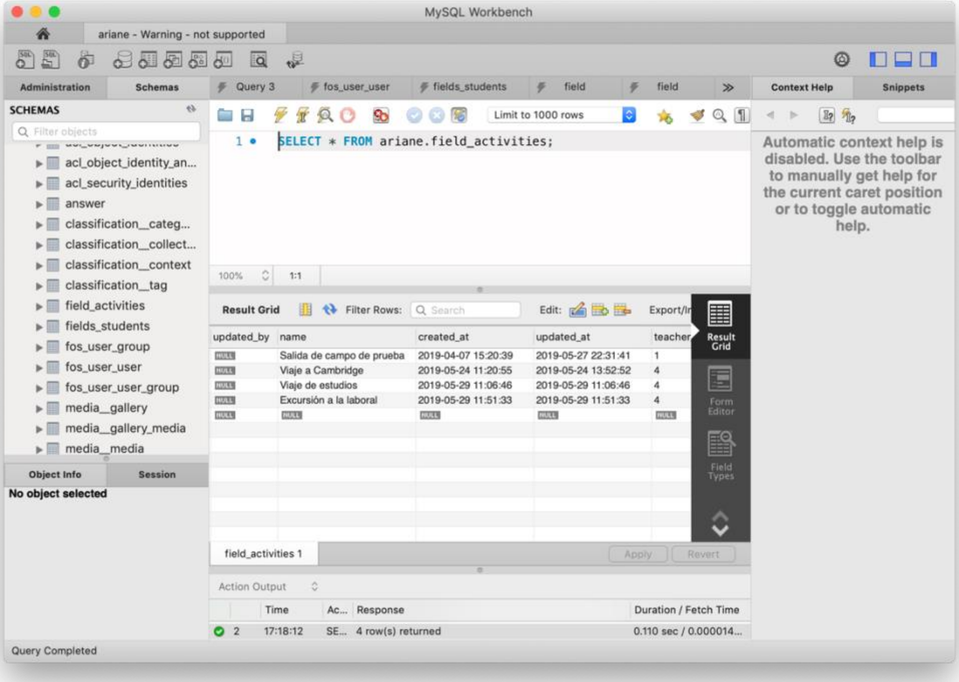
GitHub is an online service used to host our projects using the version control of Git. It offers all the distributed version control and source code management of Git, as well as other features like issues tracking, a Kanban view of the tasks to be made or third-party plugins for continuous integration or test code coverage, for example.

This service will be used to host the code of the web and mobile applications of Ariane in order to keep track of the changes and versions that are produced during the development time.

- Web Application: https://github.com/robertops18/Ariane_TFG_Backend
- Mobile Application: https://github.com/robertops18/ariane

3.3.3. MySQLServer

MySQLServer is a desktop application for the remote or local database management. It allows the administrator of the system to query data from the different tables of the database, as well as executing all the SQL procedures in order to create, update or remove data from the application.



```
1 • SELECT * FROM ariane.field_activities;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

updated_by	name	created_at	updated_at	teacher
NULL	Salida de campo de prueba	2019-04-07 15:20:39	2019-05-27 22:31:41	1
NULL	Viaje a Cambridge	2019-05-24 11:20:55	2019-05-24 13:52:52	4
NULL	Viaje de estudios	2019-05-29 11:06:46	2019-05-29 11:06:46	4
NULL	Excursión a la laboral	2019-05-29 11:51:33	2019-05-29 11:51:33	4
NULL	NULL	NULL	NULL	NULL

```
    Time      Ac...  Response           Duration / Fetch Time
✓ 2 17:18:12  SE...  4 row(s) returned  0.110 sec / 0.000014...
```

Illustration 4. General overview of MySQLServer.

4. Planning and budget

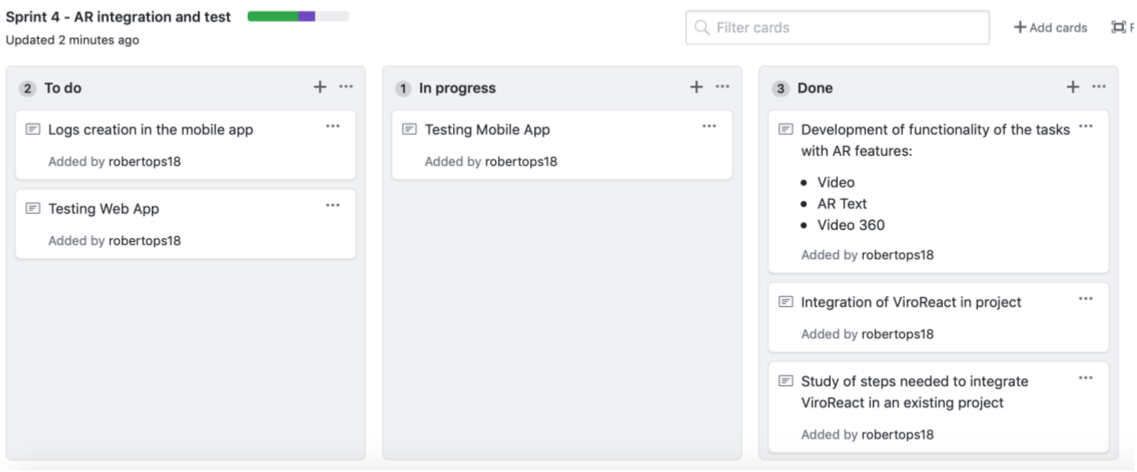
4.1. Planning

The planning of this project has been done following an agile methodology, but none in particular. Some features of the planning of this project might remember to the methodology followed by Scrum, like the division of the development of the project in sprints, but no product backlog or sprints reviews have been made.

The development of this project has been divided in 4 Sprints, each of them with a duration of approximately 4 weeks. The content of each sprint was:

1. Technologies and alternatives study
2. Web Application development
3. Mobile Application development
4. Integration of augmented reality features and testing

The different tasks to be performed in each sprint have been identified and configured using the Project feature of GitHub, which gives the developer a Kanban view of the pending to do, in progress and done tasks.



Sprint 4 - AR integration and test

Updated 2 minutes ago

2 To do

- Logs creation in the mobile app
  Added by robertops18
- Testing Web App
  Added by robertops18

1 In progress

- Testing Mobile App
  Added by robertops18

3 Done

- Development of functionality of the tasks with AR features:
  - Video
  - AR Text
  - Video 360

  Added by robertops18
- Integration of ViroReact in project
  Added by robertops18
- Study of steps needed to integrate ViroReact in an existing project
  Added by robertops18

Illustration 5. GitHub Project view.

Now, each of the sprints will be explained individually, giving a list of the different tasks performed in each sprint and the link to the Kanban view in GitHub.

4.1.1. Sprint 1: Planning

This sprint is dedicated to the study of the different alternatives and technologies that are going to be considered for the development of the project, as well as the design of the database of the system, taking into account the necessary entities and relationships. The result of this sprint is the decision to use React Native for the development of the mobile app and Symfony (with MySQL) for the web application.

The Kanban of this sprint is available here:

- https://github.com/robertops18/ariane/projects/1

Tasks performed in this sprint:

Task	Dedicated hours	Task type
Technologies study and research	50	Analysis
Ionic	10	Analysis
NodeJS	10	Analysis
MongoDB	5	Analysis
React Native	10	Analysis
Symfony	10	Analysis
MySQL	5	Analysis
Database design	15	Design

TOTAL	65

4.1.2. Sprint 2: Web App development

This sprint is dedicated to the development of the web application using Symfony, a PHP framework.

The Kanban of this sprint is available here:

- https://github.com/robertops18/ariane/projects/2

Tasks performed in this sprint:

Task	Dedicated hours	Task type
Amazon Web Services configuration	5	Development
Symfony Project setup	10	Development
Entity creation	10	Development
CRUD of entities	5	Development
API methods	10	Development
Project deploy in AWS	10	Development
CRUD testing	5	Testing

API testing	5	Testing

TOTAL	60

4.1.3. Sprint 3: Mobile app development

This sprint is dedicated to the development of the mobile app and its integration with the data and services provided by the API of the web application.

The Kanban of this sprint is available here:

- https://github.com/robertops18/ariane/projects/3

Tasks performed in this sprint:

Task	Dedicated hours	Task type
Adapt React Native template	7	Development
Interface and flow design	10	Design
API integration	10	Development
Login	5	Development
MapView	8	Development
ListView	5	Development
Field trip detail	10	Development
Task detail	30	Development
Answer to the tasks	10	Development

TOTAL	95

4.1.4. Sprint 4: AR integration and test

This sprint is dedicated to the integration of ViroReact (AR library for React Native) in the application in order to have the augmented reality features, as well as including the creation of all the logs about the actions performed by the user in the app. The final testing of the web and mobile application will also be performed.

The Kanban of this sprint is available here:

- https://github.com/robertops18/ariane/projects/4

Tasks performed in this sprint:

Task	Dedicated hours	Task type
Study of integration of ViroReact	5	Analysis
Integration of ViroReact	15	Development
AR functionality	8	Development
User logs creation	5	Development
Testing mobile app	10	Testing
Testing web app	10	Testing

TOTAL	53

4.1.5. Hours summary

Sprint	Analysis	Design	Development	Testing
Sprint 1	50	15	-	-
Sprint 2	-	-	50	10
Sprint 3	-	10	85	-
Sprint 4	5	-	28	20

Total by type	55	25	163	30

TOTAL	273 horas

4.2. Budget client summary

A summary of the budget is included in this part. The more detailed calculations are included in chapter 11 “Detailed budget”

Product	Total
Analysis	2950, 17 €
Design	957,85 €
Development	7494,20 €
Testing	1379,30 €

Total	12.781,52 €
IVA (21%)	2.684,12 €
TOTAL (with IVA)	15.456,64 €

5. Analysis

5.1. System analysis

The system will consist of two different but connected applications: a mobile application for the students to perform and answer the different activities of the field trip, and a web application where teachers can create the different field trips and tasks, as well as seeing and evaluating the answers and feedback of the students.

The students will use the mobile application to see the general information of the field trip and its tasks, as well as a map with the different locations of the tasks that they have to perform. Depending of the task, they would be able to answer it or not. These answers will be sent to the web application, where teachers can take a look at them and download them in order to evaluate them or to take feedback from the students, like for example their opinion of the trip.

The app will also store all the actions performed by the students while using it, for example, it will store when and where did the student enter a task or visualize a video, in order to track the use that students give to the system. This 'logs' will be available in the web application in order for teachers to export them and keep a registry of the use of Ariane.

In each school will be an administrator of the system. He or she will enter the web application in order to create the different users of the system with their data. There will be two main roles: teachers, that would use the web application to create the field trip and the tasks, and students, that would be the ones that use the mobile application during the trip.

Teachers would have the capability to list, create, remove, export and see the detail of every field trip organized by the school. The web application will provide all of this in an easy and intuitive way, because not all teachers have the skills to manage a complex web application. For each trip, they would assign the different students that participates in it, as well as the initial and final dates and some more information that would be detailed in the system requirements. Teachers will also be able to list, export and see the detail of the answers sent to the web application through the student's mobiles and the logs generated by it. Moreover, each task will tell the teachers how many answers it has received, in order to track the use and interest of the students in it.

5.2. System requirements

5.2.1. Register requirements

RReg_1: An unauthenticated user can be registered to the system by the administrator

- **RReg_1.1**: The administrator of the system can register users providing the following data:
  - **RReg_1.1.1**: Name and surname

RReg_1.1.1.1: These fields are optional

RReg_1.1.2: Username

RReg_1.1.2.1: This field is compulsory

RReg_1.1.3: Email

RReg_1.1.3.1: This field is compulsory

RReg_1.1.4: Password

RReg_1.1.4.1: This field is compulsory

RReg_1.1.4.2: This system will generate a random password for the user that can be changed in this process

RReg_1.1.5: Enabled

RReg_1.1.5.1: If this field is set to false, the user will not be able to log into the system

RReg_1.1.5.2: This field is optional, default to false

RReg_1.1.6: Group

RReg_1.1.6.1: Depending of the group selected, the user will have different roles and permissions

RReg_1.1.6.1.1: The users belonging to group ADMIN will be able to create, update, list and remove users, field activities, tasks and schools, as well as listing and exporting the answers registered in the system

RReg_1.1.6.1.2: The users belonging to group TEACHERS will have the same permissions of the ADMIN group, except from the capability of treating users

RReg_1.1.6.1.3: The users belonging to group STUDENTS will only be able to log into the mobile application and send answers to the system

RReg_1.1.6.1.3.1: The students can also be assigned to a student group previously created

RReg_1.1.6.2: This field is optional

RReg_1.2: The system will display an error in the creation of the user if:

RReg_1.2.1: The email format input is not correct

RReg_1.2.1.1: The correct format of the email is [*@*.*]

RReg_1.2.2: The password length is less than 6 characters

RReg_1.2.3: The username or the email are already registered in the system for another user

RReg_2: All the fields of the user profile will be stored in the database of the system

RReg_3: All the fields of the user profile can be edited by the administrator

5.2.2. Log in requirements

RLog_1: A registered user of group ADMIN or TEACHERS can log into the web application providing their username and password

RLog_1.1: If the combination of username and password is correct, the user will be logged in and able to perform the tasks available for its group

RLog_1.2: If the combination of username and password is incorrect, the system will show an error to the user

RLog_1.3: If the user trying to log into the system is not enabled, the system will warn the user that its account has not been enabled by the administrator

RLog_1.4: Any registered and logged in user can sign out of the web application whenever they want

RLog_2: A registered user of group STUDENTS can log into the mobile application providing their username and password

RLog_2.1: The error messages are the same as described in **RLog_1.1**, **RLog_1.2** and **RLog_1.3**

RLog_2.2: Any registered and logged in user can sign out of the mobile application whenever they want

RLog_3: The system will allow any authenticated user to change its password

RLog_3.1: The password can be changed from the mobile application providing the old and the new one

RLog_3.2: The password can be changed from the mobile application providing a recovery email

RLog_3.2.1: The system will send an email to the one provided by the student with a new random password

RLog_3.3: The password can be changed from the web application providing your username or email.

RLog_3.3.1: The system will send a link to the user to a page where he can introduce the new password

5.2.3. Mobile app requirements

RMA_1: A logged in user (student) will be able to see the different activities and tasks that he or she is assigned to

RMA_1.1: The mobile application will provide a map with the locations and title of the different tasks to be performed by the student

RMA_1.2: The mobile application will provide to the user a list of the field trips that he or she is assigned to

RMA_1.2.1: Each field trip will contain:

RMA_1.2.1.1: Map of the area where it is performed

RMA_1.2.1.2: Title and school associated

RMA_1.2.1.3: Initial and final date

RMA_1.2.1.4: List of tasks available in the field trip

RMA_1.2.2: Each task of the field trip will contain

RMA_1.2.2.1: Title of the task

RMA_1.2.2.3: Description and question

RMA_1.2.3: Each task of the field trip will have a type, that will be one of the following:

RMA_1.2.3.1: Video

RMA_1.2.3.1.1: The student will see a video in augmented reality related to the trip

RMA_1.2.3.1.2: This task requires an ARCore compatible device

RMA_1.2.3.2: Augmented Reality

RMA_1.2.3.2.1: The application will provide the user an Augmented Reality experience related to the trip

RMA_1.2.3.2.2: This task requires an ARCore compatible device

RMA_1.2.3.3: Rating and opinion

RMA_1.2.3.3.1: The student will be able to submit the rating of the trip in a scale from 1 to 5

RMA_1.2.3.3.2: The student will be able to submit their thoughts and suggestions about the trip

RMA_1.2.3.3.3: Every answer of the student will be stored in the database of the system

RMA_1.2.3.5: Description

RMA_1.2.3.5.1: The system will provide the student a global description of the field trip

RMA_1.2.3.6: Test

RMA_1.2.3.6.1: The student will be able to answer a question related to the field trip, choosing one of the options

RMA_1.2.3.6.1.1: if the selected option is the correct one, the app will show a success message

RMA_1.2.3.6.1.2: if the selected option is an incorrect one, the app will show an error message

RMA_1.2.3.6.2: Every answer of the student will be stored in the database of the system

RMA_1.2.3.7: Audio

RMA_1.2.3.7.1: The system will provide an audio player which will play the audio selected by the teacher when creating the task

RMA_1.2.3.8: Youtube

RMA_1.2.3.8.1: The system will provide a video player of a Youtube video

RMA_1.2.3.9: Video 360

RMA_1.2.3.9.1: The system will provide a virtual reality experience in the form of a 360 degree video

RMA_1.2.3.9.2: This task requires an ARCore compatible device

RMA_1.3: The student will only be able to perform the task if their location is in a range of 100 meters from the location of the task

RMA_2: The mobile application will store all the actions performed by the student in the system

RMA_2.1: These logs will be stored in the database of the system

RMA_2.2: The application will store

RMA_2.2.1: When and where a task is answered or performed

RMA_2.2.2: The user that performs it

RMA_2.2.3: The actions done by the student while performing the task

5.2.4. Web Application requirements

RWA_1: An authenticated user of group ADMIN will be able to create users as explained in **RReg_1**

- **RWA_1.1**: When creating the user, the admin can assign a student group to it if it is a student
- **RWA_1.2**: This student groups have to be created in advanced, providing a name to it
- **RWA_1.3**: If the admin does not assign any group to the student created, it can be assigned later editing the user

RWA_2: An authenticated user of group ADMIN or TEACHER will be able to create the following entities of the system

- **RWA_2.1**: Field activities, providing
    - **RWA_2.1.1**: Field activity title
        - **RWA_2.1.1.1**: This field is compulsory
    - **RWA_2.1.2**: Initial and finish date
        - **RWA_2.1.2.1**: These fields are compulsory
    - **RWA_2.1.3**: Area where the field trip is performed
        - **RWA_2.1.3.1**: This field is compulsory
    - **RWA_2.1.4**: Student group to which it is assigned
        - **RWA_2.1.4.1**: This field is optional
    - **RWA_2.1.5**: Teacher in charge of the trip
        - **RWA_2.1.5.1**: This field is optional
- **RWA_2.2**: Tasks, providing
    - **RWA_2.2.1**: Name of the task
        - **RWA_2.2.1.1**: This field is compulsory
    - **RWA_2.2.2**: Type
        - **RWA_2.2.2.1**: This field is compulsory
    - **RWA_2.2.3**: Description
        - **RWA_2.2.3.1**: This field is compulsory
    - **RWA_2.2.4**: Question or URL, depending of the type
        - **RWA_2.2.4.1**: This field is compulsory
        - **RWA_2.2.4.2**: If the type of the task is video or audio, this field will contain the URL of the media
    - **RWA_2.2.5**: Options of the test question
        - **RWA_2.2.5.1**: This field is optional
    - **RWA_2.2.6**: Correct option
        - **RWA_2.2.6.1**: This field is optional
    - **RWA_2.2.7**: The field activity to which the task is associated
        - **RWA_2.2.7.1**: This field is optional
    - **RWA_2.2.8**: Location of the task
        - **RWA_2.2.8.1**: This field is compulsory
    - **RWA_2.2.9**: Image URL for the task
        - **RWA_2.2.9.1**: This field is optional

RWA_3: An authenticated user of group ADMIN or TEACHER will be able to edit any of the fields explained in **RWA_2** of the field activities and tasks

RWA_4: An authenticated user of group ADMIN or TEACHER will be able to see the detail of the field activities and tasks stored in the system

> **RWA_4.1**: For the field trips and tasks, the detail will show the fields explained in **RWA_2**, as well as
>
> > **RWA_4.1.1**: In the detail of the field activities, a list of the tasks belonging to it will be shown
> >
> > **RWA_4.1.2**: In the detail of the tasks, the number of answers received by the students will be displayed

RWA_5: An authenticated user of group ADMIN or TEACHER will be able to list all the field activities, tasks, answers and logs in the system

> **RWA_5.1**: For the field activities and tasks, all the previously explained fields will be shown to the user
>
> **RWA_5.2**: For the answers and logs, the web application will show
>
> > **RWA_5.2.1**: Student that creates the answer
> >
> > **RWA_5.2.2**: Task answered
> >
> > **RWA_5.2.3**: Field activity where the task is located
> >
> > **RWA_5.2.4**: Answer or action of the student
> >
> > **RWA_5.2.5**: These fields will be also the ones displayed in the detail of every answer and log
>
> **RWA_5.3:** For the logs, the location where it was created will be also listed in the web application

RWA_6: An authenticated user of group ADMIN or TEACHERS will be able to delete any of the field activities, tasks, answers or logs created in the system

RWA_7: The web application will provide filters for every field in every list of field activities, tasks, answers and logs

RWA_8: An authenticated user of group ADMIN or TEACHERS will be able to export and download the list of field activities, tasks, answers or logs created in the system

5.2.5. Non-functional requirements

RNF_1: The web application will run in the main web browsers: Google Chrome, Safari, Mozilla Firefox, Opera and Microsoft Edge

RNF_2: The system must comply with the accessible content guidelines AA of W3C

RNF_3: The web application will be hosted in the domain http://ariane-uniovi.es

RNF_4: The mobile application will be available for Android devices

> **RNF_4.1**: The augmented reality features will only be available to devices with ARCore support

5.3. Actors identification

An actor specifies a role played by a user or any other system that interacts with the subject. These actors will later be used in the use case identification.

In this system, the following actors are identified:

- Registered user (teacher or student, not authenticated)
- Admin (authenticated user of group ADMIN)
- Teacher (authenticated user of group TEACHERS)
- Student (authenticated user of group STUDENTS)

5.4. Use case diagrams

5.4.1. User use cases

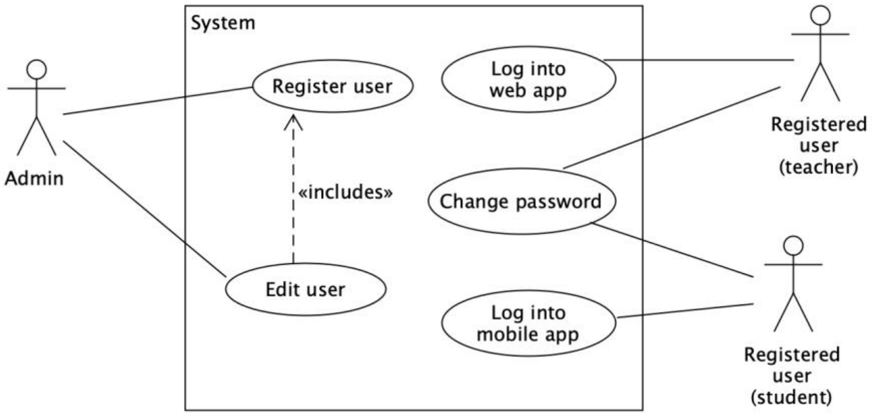


Illustration 6. User use case.

5.4.2. Web application use cases

5.4.2.1. Field trips

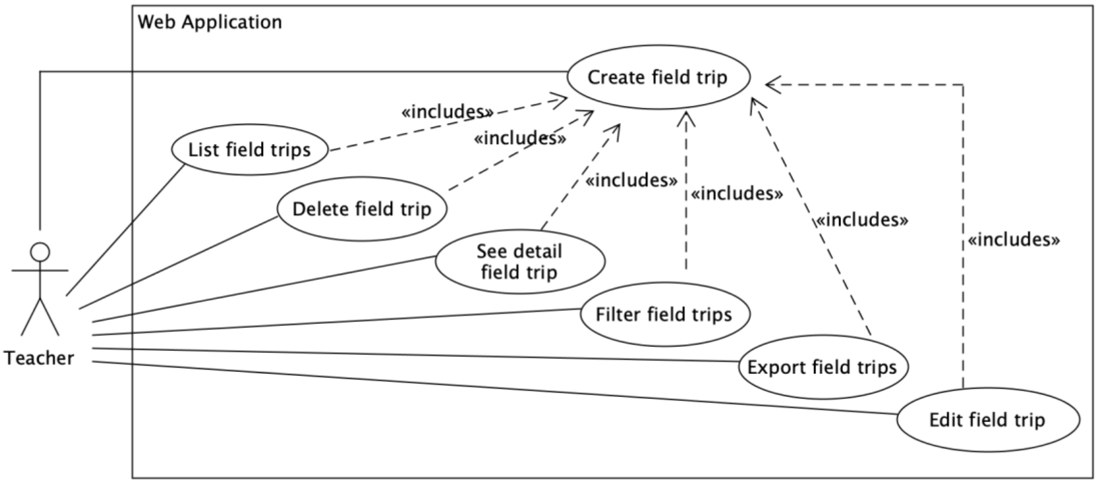


Illustration 7. Field trip use case.

5.4.2.2. Tasks

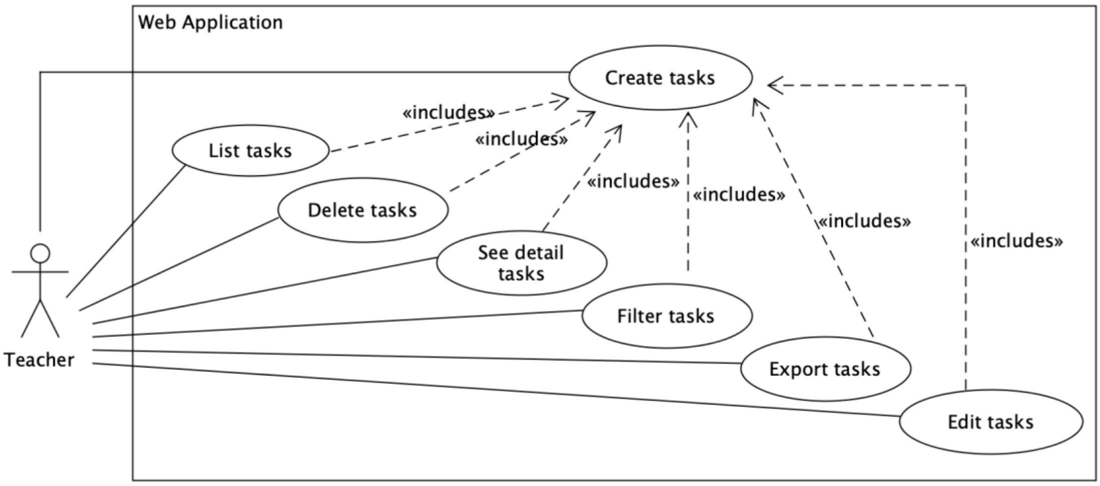


Illustration 8. Tasks use case.

5.4.2.3. Answers and logs

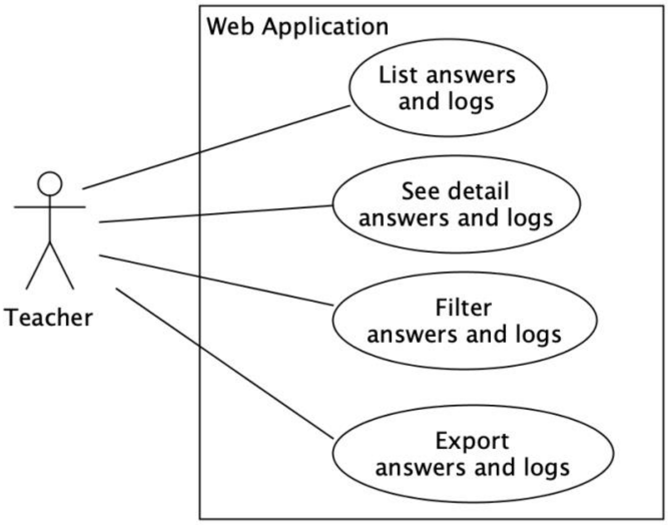


Illustration 9. Answers use case.

5.4.3. Mobile application use cases

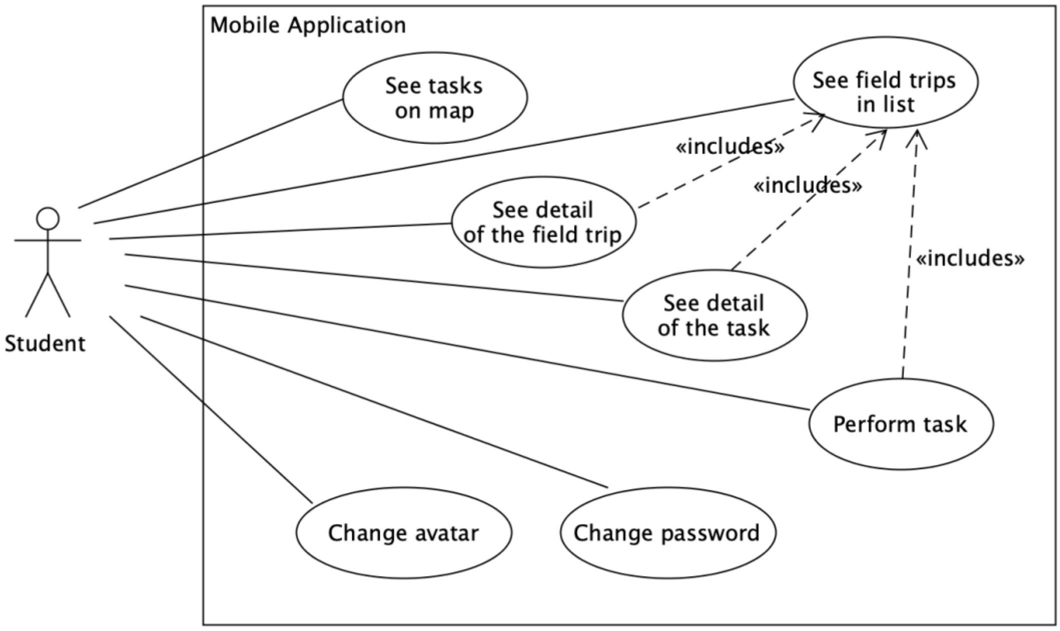


Illustration 10. Mobile app use case.

5.5. Use cases

Use case	
ID	UC-1
Name	Register user
Actors	Admin
Started by	Admin
Type	Primary
References	RReg_1, RReg_2, RWA_1
Preconditions	The user to be created is not registered in the system
Postconditions	The user will be registered in the system
Purpose	
Register a user in the application	
Detailed description	
The teachers and students must be registered in the system in order to use the application. In order to register a user, the admin will input the necessary data. The system will check this data, and if they are correct, the user will have been registered succesfully. Otherwise, the system will show an error.	
Basic path	
Actor	System
1 - The admin introduce the required data in order to register a new user: name and surname (optional), username, email, password and group	
	2 - The system will validate the data, looking if there is another users already registered with the provided username or emai
	3 - The system takes the provided data and creates a new user
	4 - The system redirects the admin to the user listing page
5 - The admin enables the user in order to let him log into the system	
Alternative paths	
1a - The user already exists in the system. The application will show an error to the admin and redirect him to 1	

Use case	
ID	UC-2
Name	Edit user
Actors	Admin
Started by	Admin
Type	Primary
References	RReg_3
Preconditions	The user has to be registered in the system
Postconditions	The user info will be changed
Purpose	
Change some info of the registered user	
Detailed description	
Some teacher or student notice that their data is incorrect or that they want to change their username or email, so they contact the admin of the system in order to edit those fields. In order to edit it, the admin will go to the user listing and edit the fields of the user that are wrong or wanted to be chenged.	
Basic path	
Actor	System
1 – Admin udpates the required fields of the user	
	2 - The system will validate the data, looking if there is another users already registered with the provided username or email
	3 - The system takes the provided data and updates the user
	4 - The system redirects the admin to the user listing page
Alternative paths	
1a - The newly introduced fields are the same as those of another user. The application will show an error to the admin and redirect him to 1	

Use case	
ID	UC-3
Name	Login and logout from the apps
Actors	Admin, teacher and student
Started by	Admin, teacher or student
Type	Primary
References	RLog_1, RLog_2
Preconditions	The user has to be registered in the system
Postconditions	The user will be logged into/out the web/mobile app
Purpose	
Log in or log out of the web or mobile app	
Detailed description	
A registered user will log in to the system in order to perform the activities or tasks that he or she wants. Whenever this user wants, he can log out of the system in order to close his session and protect the operations that have been performed.	
Basic path	
Actor	System
1 – A registered user introduces its username and password in the system	
	2 – The system validates the data and redirects the user to the main page
3 – The user performs different operations	
4 – The user logs out	
	5 – The system closes the session of the user and redirects to the login page
Alternative paths	
1a - The login credentials do not match any registered user of the system, so an informative message is shown to the user and the flow will go back to 1	

Use case	
ID	UC-4
Name	Change password
Actors	Admin, teacher and student
Started by	Admin, teacher or student
Type	Primary
References	RLog_3
Preconditions	The user has to be registered in the system
Postconditions	The user password will be changed
Purpose	
Change the password in order to log in to the system	
Detailed description	
A registered user wants to change his password in order to log in yo the system. The students should fill a form with the old password and with the new one in the mobile application, while the admin and teachers have to do this process through the web app, which will send them an email with the new password.	
Basic path	
Actor	System
1 – A registered user wants to change the password. A teacher request a new password in the web app and the student introduces the old and new password (2 times) in the mobile app	
	2 – The system sends an email to the teacher with the link to the change password screen. The mobile app checks that the newly introduced password is different from the old one and changes it.
3 – The teacher introduces the new password in the change password page.	
4 – The users can now log in to the system with their new passwords	
Alternative paths	
1a – If the new password introduced by the student is the same as the old one, an informative error will be displayed and the flow will return to 1	
1b – The student has to introduce the new password two times. If they are not the same, an informative error will be displayed and the flow will return to 1	
1c – In order to change their password, teachers have to provide their email. If that email does not belong to any user of the system, an informative error will be displayed and the flow will return to 1	

Use case	
ID	UC-5
Name	Field trips and tasks visualization and performance
Actors	Student
Started by	Student
Type	Primary
References	RMA_1
Preconditions	The student has to be registered and authenticated in the system
Postconditions	-
Purpose	
Visualize the field trips to which the student is assigned and perform the tasks of each one	
Detailed description	
A student can use the mobile application in order to see the different field trips and tasks that he is assigned to. The field trips will be displayed in a list, where every detail of the field trip will be accesible, as well as a list of the tasks of each trip. All the tasks of every field trip will also be available in a map, representing each task location with a marker. For some of the tasks, for example the opinión, test and rating ones, the student will be able to answer them. These answers will be stored in the database of the system	
Basic path	
Actor	System
1 – A student logs in to the mobile app and sees a field trip which he is assigned to. In the detail of the field trip, there is a test task	
2 – The students choose one of the options and answers the question	
	3 – The system will check if the answer is the correct one and notify the user if he has answered the question correctly
	4 – The system will store the answer of the question in the database
5 – The user can keep listing the field trips, answering and performing tasks, or seeing the location of the tasks in the map	
Alternative paths	
2a – If the location of the user is not in range with the location of the task, he will not be able to answer the task, and the flow will return to 1	

Use case	
ID	UC-6
Name	Logs storage
Actors	Student
Started by	System
Type	Primary
References	RMA_2
Preconditions	The student has to be registered and authenticated in the system
Postconditions	The actions performed by the student are stored in the database
Purpose	
Keep track of the actions that the students make in the mobile application	
Detailed description	
The system will store all the actions that are performed by a student, keeping a relation to the field trip and task in which the actions are performed. The location of the user in each one of them will also be stored. Some actions stored would be: the visualization of a video, if the students see the description of the trip, a student tried to answer a task but he was not in range of the location of the task…	
Basic path	
Actor	System
1 – A student watches the video related to a field trip	
	2 – The system will create a registry of this action and store it in the database
3 – The student can keep making operations in the mobile app	
Alternative paths	

Use case	
ID	UC-7
Name	Field trips and tasks creation
Actors	Admin, teacher
Started by	Admin, teacher
Type	Primary
References	RWA_2
Preconditions	The user has to be authenticated in the system
Postconditions	The field trip or tasks will be created
Purpose	
Creation of field trips and tasks in the web application in order to consume them in the mobile application	
Detailed description	
The teacher will create the field trip and tasks belonging to it providing the fields explained in RWA_2: • Title • Location • Area • Description and question • Students list assigned to the trip • Etc	
Basic path	
Actor	System
1 – A teacher creates a field trip for 20 students providing the required data	
	2 – The system will check if the data provided is correct and create the field trip, storing it in the database
3 – The teacher creates 3 tasks (video, opinion and rating) and assigned them to the field trip previously created	
	4 – The system will check if the data provided is correct and create the tasks, storing them in the database
5 – The 20 students assigned to the trip can now see and perform the different tasks of the trip in the mobile application	
Alternative paths	
1a – If the data provided for the creation of the field trip is not correct, an error will be displayed by the system and the flow will return to 1	
3a – If the data provided for the creation of the field trip is not correct, an error will be displayed by the system and the flow will return to 3	

Use case	
ID	UC-8
Name	Field trips and tasks editing
Actors	Admin, teacher
Started by	Admin, teacher
Type	Primary
References	RWA_3
Preconditions	The user has to be authenticated in the system and the field trip or tasks to be edited has to be created already.
Postconditions	The field trip or tasks will be edited
Purpose	
Editing of field trips and tasks in the web application	
Detailed description	
The teacher can edit any of the fields of the field trips and tasks in order to change some data or adding another student of the list of assigned students, for example.	
Basic path	
Actor	System
1 – A teacher edit some field of an already created field trip or task	
	2 – The system will check if the data provided is correct and edit the field trip, saving the changes in the database
3 – The changes of the trip or tasks will be displayed in the mobile application	
Alternative paths	
1a – If the data provided while editing the field trip is not correct, an error will be displayed by the system and the flow will return to 1	

Use case	
ID	UC-9
Name	Entity deletion
Actors	Admin, teacher
Started by	Admin, teacher
Type	Primary
References	RWA_6
Preconditions	The user has to be authenticated in the system
Postconditions	The entity will be deleted
Purpose	
Deletion of any field trip, task, answer or log stored in the system	
Detailed description	
The teacher will delete the entity and it will be deleted from the database of the system. In order to delete answers and logs, the teacher will only need to delete the entity in the web application. In order to delete a task, the teacher will previously need to delete all the answers and logs related to that task. And finally, for the deletion of a field trip, all the tasks related to it will also need to be deleted already.	
Basic path	
Actor	System
1 – A teacher wants to delete an already performed field trip	
	2 – The system will check if there is data related to that field trip in the system and delete the field trip
3 – The teacher will be able to continue performing the operations he wants	
Alternative paths	
1a – If the field trip has some assigned tasks to it, or those tasks have some answers or logs related to it, an informative error will be displayed and the flow will return to 1	

Use case	
ID	UC-10
Name	See the detail and list entities
Actors	Admin, teacher
Started by	Admin, teacher
Type	Primary
References	RWA_4, RWA_5
Preconditions	The user has to be authenticated in the system
Postconditions	The entity data will be displayed in the web application
Purpose	
Listing of the data of the entities of the system to give a general look of them	
Detailed description	
All the fields of the entities created in the system (field trips, tasks, answers and logs) will be displayed as a table with all the data of each entity, like name, list of students assigned or type of task. Moreover, if the user wants to, he can see the detail of every entity, taking an individual look at each of the field of the selected item.	
Basic path	
Actor	System
1 – A teacher wants to list all the tasks created in the system	
	2 – The system will query the tasks entities and displayed them in a table
3 – The teacher enters in the detail of one of them	
	4 – The system will display a list of all the fields of the task
Alternative paths	

<table>
<tr><th colspan="2">Use case</th></tr>
<tr><td>ID</td><td>UC-11</td></tr>
<tr><td>Name</td><td>Filter and export entities</td></tr>
<tr><td>Actors</td><td>Admin, teacher</td></tr>
<tr><td>Started by</td><td>Admin, teacher</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>RWA_7, RWA_8</td></tr>
<tr><td>Preconditions</td><td>The user has to be authenticated in the system</td></tr>
<tr><td>Postconditions</td><td>The entity will be filtered and the data exported and downloaded</td></tr>
<tr><td colspan="2">Purpose</td></tr>
<tr><td colspan="2">Filter and export the entities stored in the system</td></tr>
<tr><td colspan="2">Detailed description</td></tr>
<tr><td colspan="2">For every entity of the system, there exist filters for every field of the entity. For example, if there are a lot of tasks created in the system, and the teacher only wants to see the tasks related to one field trip, the data can be queried in order to display only the data required by the teacher. Moreover, this data can be exported to an Excel file and donwloaded into the teachers device.</td></tr>
<tr><td colspan="2">Basic path</td></tr>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1 – A teacher filter the tasks in order to see only the ones related to the specified field trip</td><td></td></tr>
<tr><td></td><td>2 – The system will query the data of the task entity and return to the user only the related tasks</td></tr>
<tr><td>3 – The teacher export the list of tasks to an Excel file and download it to his device</td><td></td></tr>
<tr><td></td><td>4 – The system will create an Excel file with all the data of the previously filtered tasks and provide the Excel to the teacher</td></tr>
<tr><td colspan="2">Alternative paths</td></tr>
</table>

5.6. Traceability matrix

	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11
RReg_1	x										
RReg_2	x										
RReg_3		x									
RLog_1			x								
RLog_2			x								
RLog_3				x							
RMA_1					x						
RMA_2						x					
RWA_1	x										
RWA_2							x				
RWA_3								x			
RWA_4										x	
RWA_5										x	
RWA_6									x		
RWA_7											x
RWA_8											x

5.7. User interface analysis

In order to make a previous analysis of the different user interfaces that Ariane will have, the paper prototyping methodology is going to be used. The methodology is a technique that allows you to create and test user interfaces quickly and cheaply, just drawing in a paper what it could be the graphic interface design of your final application. It also makes easier to perform the changes to be made to the interface. Using this technique, the different screens thought at the beginning of Ariane's development were sketched, as well as the interactions and navigability between one another.

From the start, the two main pages of Ariane's mobile application were really clear: a home page with a map containing all the field trips locations, and another screen with all the field trips as a list. Then, the general screens like the login, change password or profile were deigned.

The last screens to be designed were the field trip and task details. In the first design, which is slightly different from the final one, but really similar at the same time, not all types of tasks were thought and developed, so only the video in augmented reality one was sketched.

In order to keep it clearer and more understandable, I have resketched the first paper prototypes in a clean paper, but the design and flow of the application is the same as the ones done previously to the development of Ariane. These are the final paper prototypes of the system:

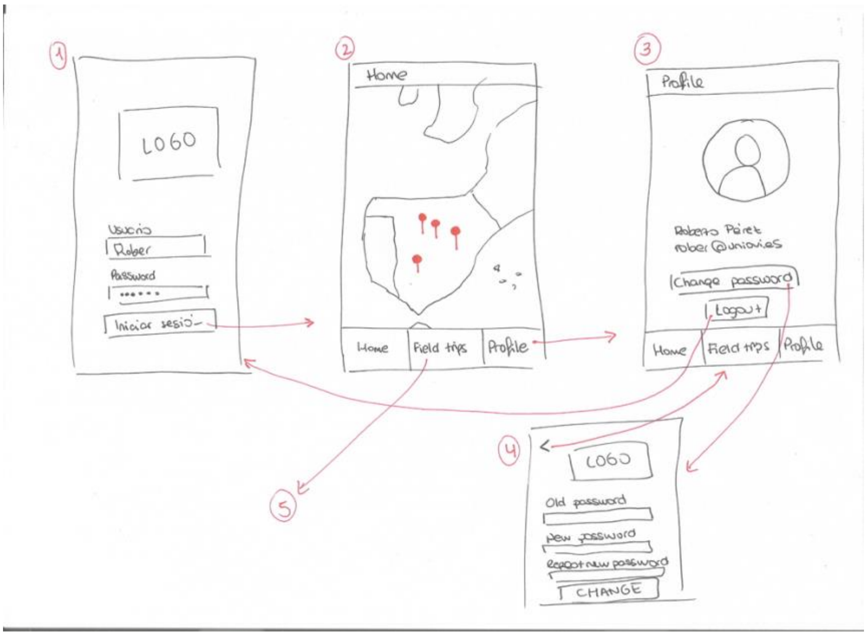


Illustration 11. Paper prototyping (1).

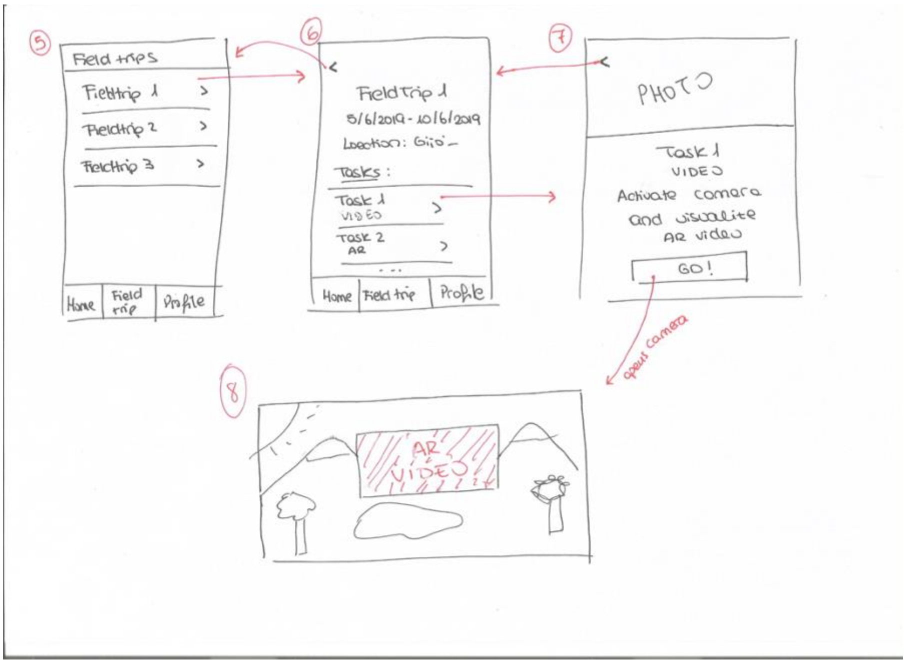


Illustration 12. Paper prototyping (2).

The different screens designed in the prototype are the following:

1. Login page
2. Home map page
3. Profile page
4. Change password page
5. Field trip list page
6. Field trip detail page
7. Task detail page
8. AR Video task performance page

5.8. Test plan

During the development of this project, the following test will be executed:

- System testing: These tests provide the system developer an overall view about the correct functionality of the system. They are used in order to test that all modules and connections with external systems work in the desired way.
- Unit testing: It is the way to prove the correct functionality of a piece of code in order to test if the individual components work as expected.
- Usability testing: The app will be tested by final users in order to see how usable it is.

6. Architecture and design

6.1. Packages diagram

6.1.1. Web Application

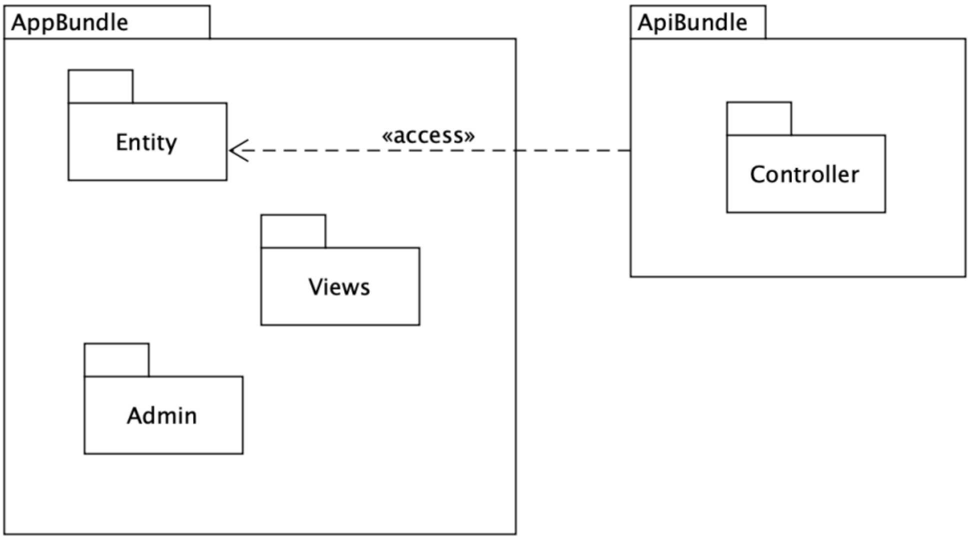


Illustration 13. Packages diagram Web Application.

6.1.1.1. AppBundle

It is the main package of the web application, containing all the components necessary for entities creation and manipulation. It is divided in three subpackages:

- Entity: It contains all the entities and their fields and methods of the application, for example, field trips, tasks or answers
- Views: It contain the custom views used in the Symfony template
- Admin: In this package, there is located one admin class per entity. In each of them, there are contained the configurations of the fields required for the listing, creating, filtering and showing of every entity of the system

6.1.1.2. ApiBundle

It contains all the API REST methods in the Controller subpackages. The API mehots are divided by the entities that are related to. For example, the TaskController will contain the different API methods related to the entity Task.

6.1.2. Mobile application

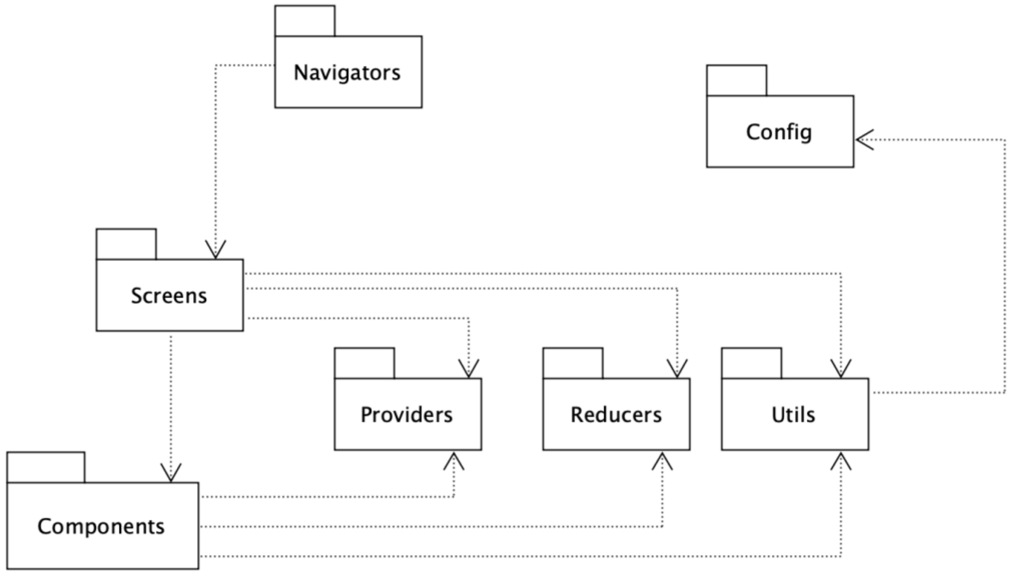


Illustration 14. Packages diagram Mobile App.

6.1.2.1. Navigators

This package and its files are responsible to managing the navigation between Screens of all the React Native mobile application. Inside them, there are several definitions of the flow of the application, as well as the tabs and pages displayed in every part of it.

- AppNavigator: Default and general navigator of the app. It controls the rest of navigators used in the app
- RootNavigator: Defines the tab navigation as well as the groups of screens used in the app

6.1.2.2. Screens

The files contained in this package are the different Screens of the application. Every screen file contains the Javascript code in order to render the screen, the different methods and the styles applied to each of them. Every screen has to be configured and declared in the navigators of the app.

- AR
    - AR: Screen for the AR experiences of the app. It renders a different scene depending on the type of task
    - Scenes
        - TextSceneAR: Scene for the standard text in augmented reality
        - VideoAR: Scene for the visualization of a video in augmented reality

        - Video360AR: Scene for the visualization of a 360 degree video in virtual reality
- Change-password: Screen for changing the password of the student
- Details:
    - Details: Screen for the detail of the field trip
    - Details-Task: Screen for the detail of the task
- Home: Home screen of the app containing the general map
- List: Screen containing the list of field trips
- Login: Login page
- Profile: Screen with the data of the user

6.1.2.3. Components

A component is a set of visual items that can be displayed inside screens and used in several parts of the application.

- Back-icon: Icon for going back using the navigation
- Button: Custom button
- Dismiss-keyboard-wrapper: Component for exiting the keyboard of the device touching any part of the screen
- Horizontal-separator: Line for separating the elements of lists
- List: Custom list
- List-item: Custom list item
- Map: Google Maps component in order to show the location of the user and the locations of the tasks to be performed
- Player: Audio player for the task of type audio
- Profile-icon: Icon for displaying the avatar of the user
- Text-input: Custom text input
- Text-link: Text wrapper to place a link to a website

6.1.2.4. Providers

In this package there are located the Javascript files in charge of communicating the app with another services, such as the API or Firebase (for example, for implementing push notifications).

- Api: Used to declare the API methods used in the app
- Firebase: Provider in order to implement push notifications
- Image-data-form: Used to create a DataForm to send images to the API
- Token-validation: Provider for validating the user authentication token

6.1.2.5. Reducers

These are the files in charge of having a global state to store and retrieve data from all the screens and components of the application

- Index: General reducer where the rest are declared
- Data: Reducer for storing the field trips gotten from the API
- Navigation: Reducer for the navigation
- Profile: Used to store the profile of the authenticated user
- User: Reducer to store the session of the user

6.1.2.6. Utils

Some utilities and functionalities that screen and components will use, for example, the translate util, in order to internationalize the application.

- Language: Translate util

6.1.2.7. Config

Config files that the utilities will use, for example, the translations files for changing the texts of the application

- EN: English translations file
- ES: Spanish translations file

6.2. Deploy diagram

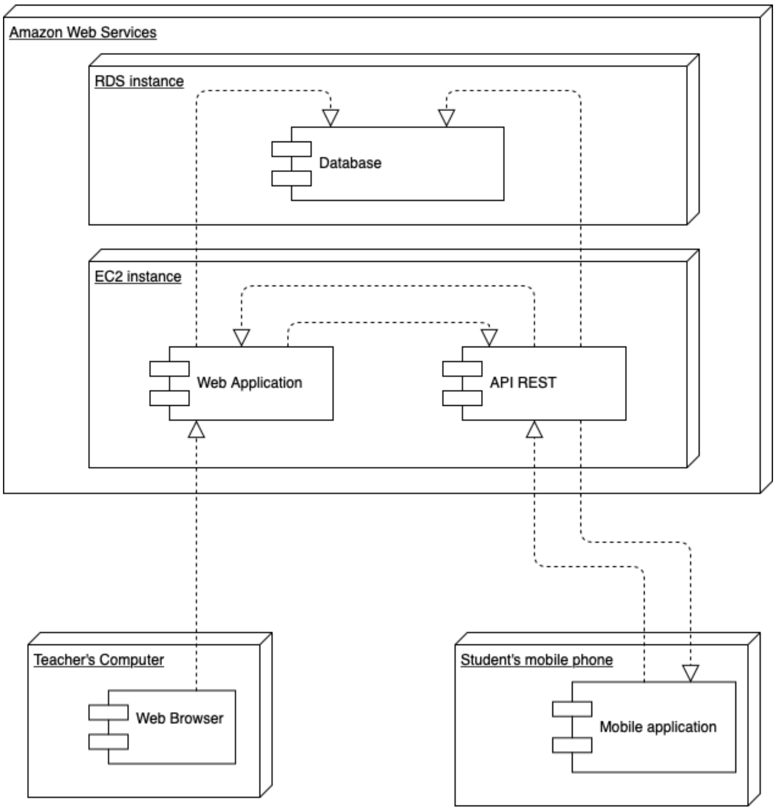


Illustration 15. Deploy diagram.

In this diagram, we can see different elements and modules:

- Amazon Web Services: It is the container of the web application and database. It is divided in two parts:
  - EC2 instance: Where the web app and API are hosted
  - RDS instance: Where the database is hosted
- Teacher's computer: Device from where the teachers connect to the web application using a web browser
- Student's mobile phone: Using the mobile application of Ariane, the students are able to communicate with the web app via the API REST.

A summary of this deploy diagram would be: the teachers access the web application using a web browser in order to manage the entities of the database. The web application is communicated with the database and the API REST, which provides a service that students consume using the mobile application, that is also communicated with the API in order to save the answers and logs created in the mobile app.

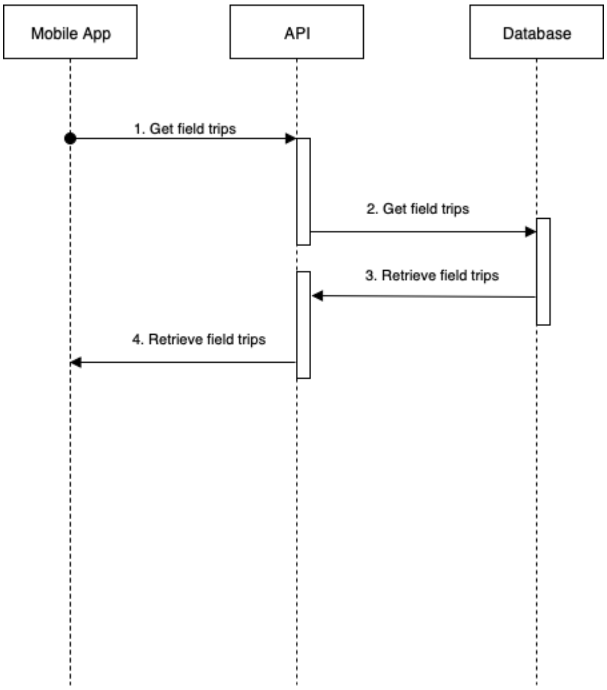


Illustration 17. Sequence diagram of the system flow of the mobile application.

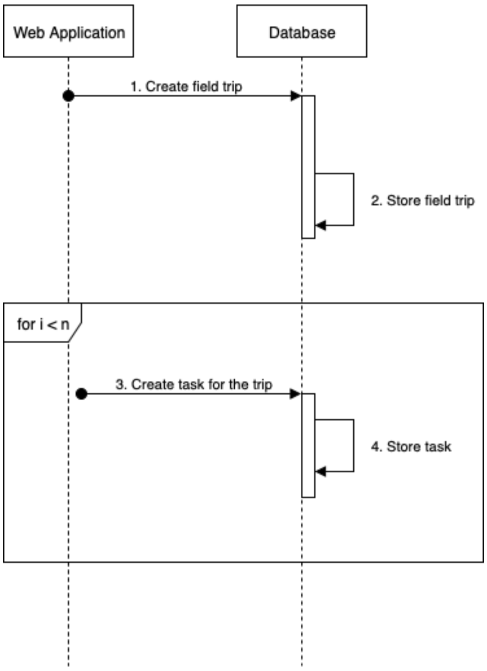


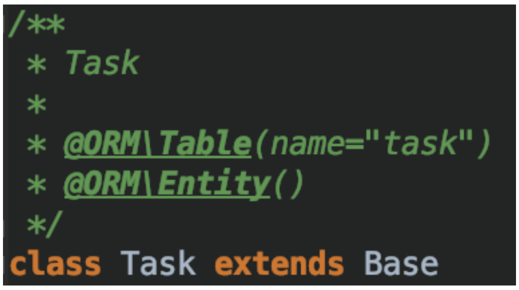
Illustration 16. Sequence diagram of the system flow of the web application.

6.3. Database design

6.3.1. DBMS Description and Integration

Because of the use of the framework Symfony for the development of the web application, the DBMS used for the design of the database of Ariane will be SQL-like, in particular MySQL, in order to manage the data stored using the program MySQLServer.

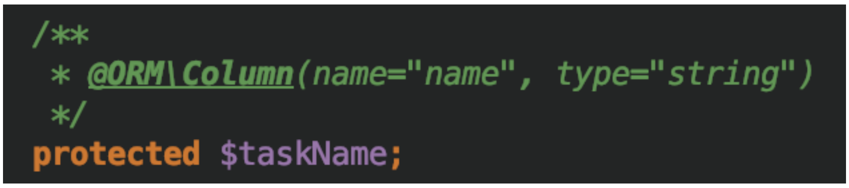
The integration of this DBMS with the system is very straight-forward, as Symfony provides a series of annotations in order to map the entities created in the different classes of the codes to tables of the database. Then, the management of this entities in the code will be related to an Entity Manager, provided in the implementation of Symfony. For every entity, we will annotate them with @Table and @Entity, in order to create the table related to this entity.



```
/**
 * Task
 *
 * @ORM\Table(name="task")
 * @ORM\Entity()
 */
class Task extends Base
```

Illustration 18. Task entity annotation.

For example, in the entity Task, the annotations @ORM\Table with the name of the desired table and @ORM\Entity will be used in order to create the table 'task' in the database. Then, for each field of the entity, something similar will be done, adding an annotation to each of them containing the name of the column in the table and the type of the field (text, number, etc.). For the name of the task, it would be something like the following, using the annotation @ORM\Column

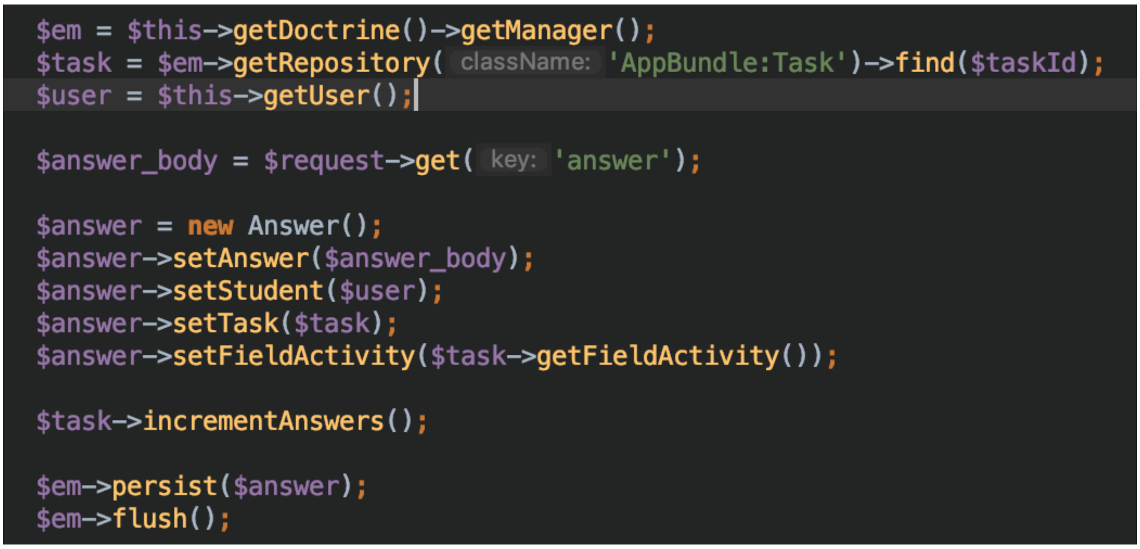


```
/**
 * @ORM\Column(name="name", type="string")
 */
protected $taskName;
```

Illustration 19. Field annotation.

With these annotations we can create all the entities we want with the required fields in a very simple way. The relations between entities are also done with annotations like @ORM\ManyToOne, @ORM\OneToMany or @ORM\OneToOne.

In order to create, edit or delete entities by hand in the code, for example, in the API methods, the entity manager and its main methods are used. For example, if we want to find a task with id equal to 2, the method 'find' or the entity manager will be used, passing the desired id as a parameter. For storing and persisting all the changes and creations of entities, the methods flush and persist will be used.

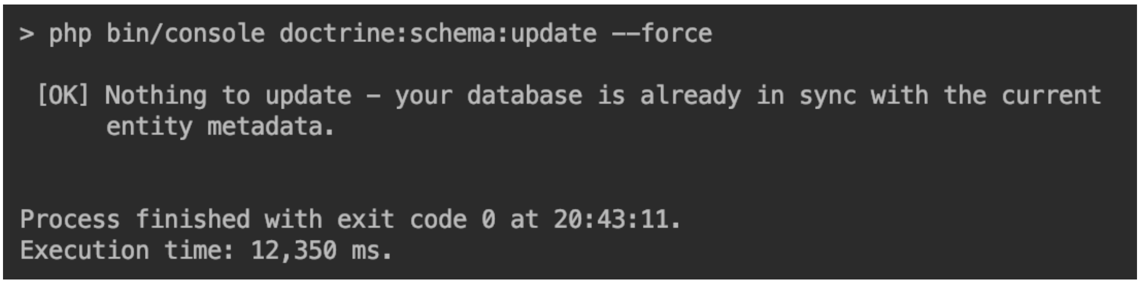


```
$em = $this->getDoctrine()->getManager();
$task = $em->getRepository( className: 'AppBundle:Task')->find($taskId);
$user = $this->getUser();

$answer_body = $request->get( key: 'answer');

$answer = new Answer();
$answer->setAnswer($answer_body);
$answer->setStudent($user);
$answer->setTask($task);
$answer->setFieldActivity($task->getFieldActivity());

$task->incrementAnswers();

$em->persist($answer);
$em->flush();
```

Illustration 20. Example of use of the entity manager.

This example is from the API method for creating answers, that receives the task id and the answers as parameters of the method. First, the entity manager searches the related task for the answer with the find method. Then, the answer is created and its fields are initialized. Finally, the two previously mentioned methods, persist and flush, are called and the answer is saved to the database.

In order to initialize all the tables and fields in the database, there is something more to do. Symfony provides the user a parameters file where the data of the database is configured (url, user, password, name, etc.). Then, and making use of the Symfony executer (console), the command doctrine:schema:update needs to be executed in the beginning and every time that the design of the database changes. This command will update the database schema and provide the user with information about the number of queries executed and the result of the operation.



```
> php bin/console doctrine:schema:update --force

  [OK] Nothing to update - your database is already in sync with the current
       entity metadata.


Process finished with exit code 0 at 20:43:11.
Execution time: 12,350 ms.
```

Illustration 21. Database schema command.

6.3.2. E-R Diagram

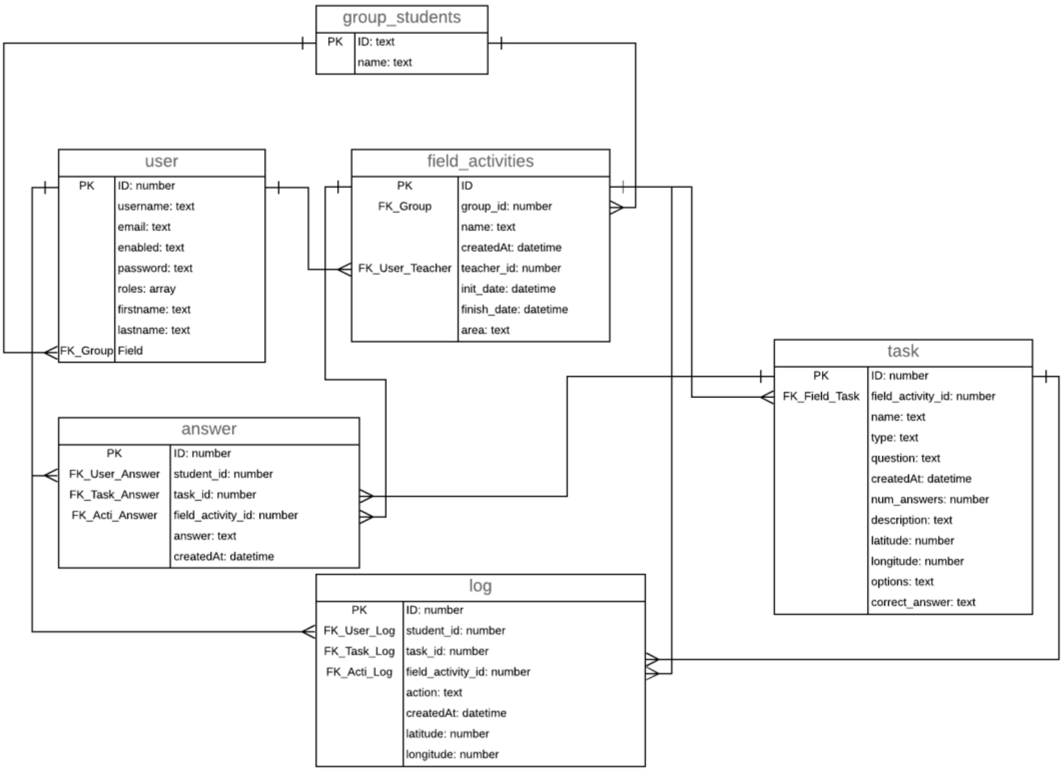


Illustration 22. E-R diagram.

6.4. User interface design

The design of Ariane's user interface has been created following the design described in the analysis part of this document. There, all the interactions between screens are explained, following a drawn interface methodology. Nevertheless, in that design, not all different types of tasks where developed, so the design of every individual component was not included. In this part, all screens and components of the different tasks would be presented using screenshots from the final mobile application.

6.4.1. Login

The login page of the mobile application consists of the logo of Ariane and two inputs for the username and password of the user. It also has a link to the recovery password page, where the student can introduce his email in order to change this access password.

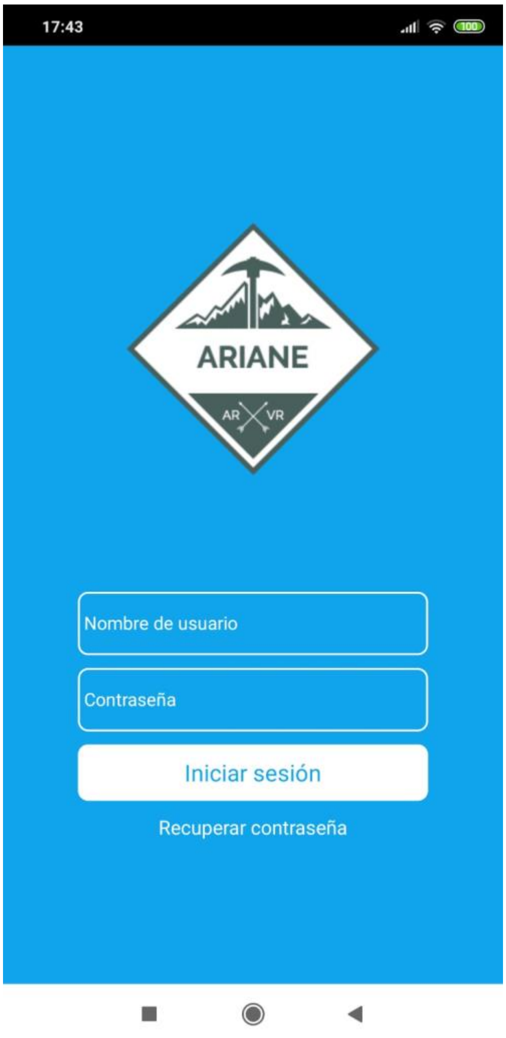


Illustration 24. Login page.

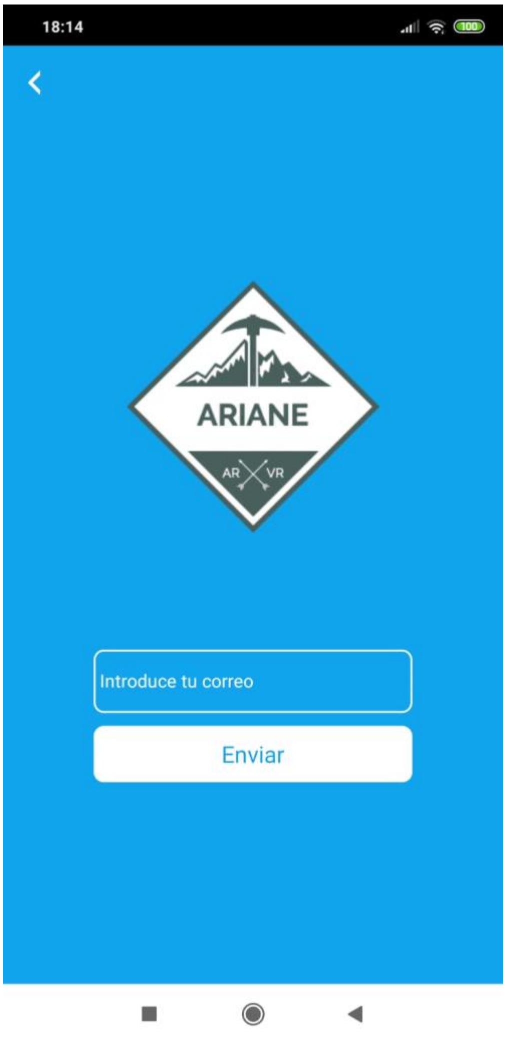


Illustration 23. Password recovery page.

In the login page we found the following elements:

- Input text for the username of the student
- Input text for the password of the student
- Button for logging in the app
- Link to the recovery password page

The recovery password page has the following items:

- Input text for introducing the email where the new password has to be sent
- Button for submitting the form

6.4.2. Home screen

The first page that the student sees when he logs in, is the map view containing all the locations of the tasks where he is associated with, as well as another marker in a different colour displaying the current location of the user.

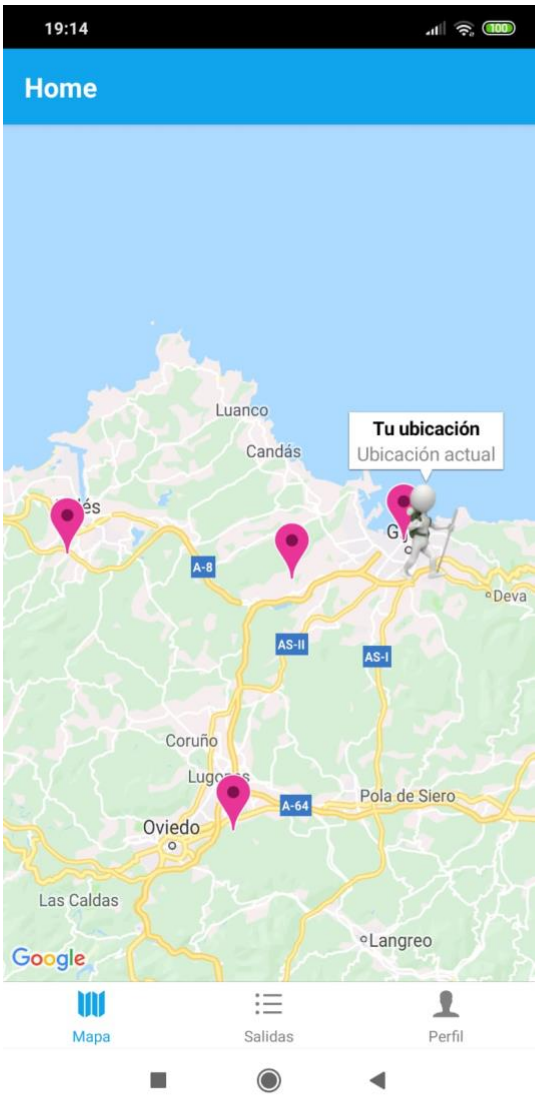


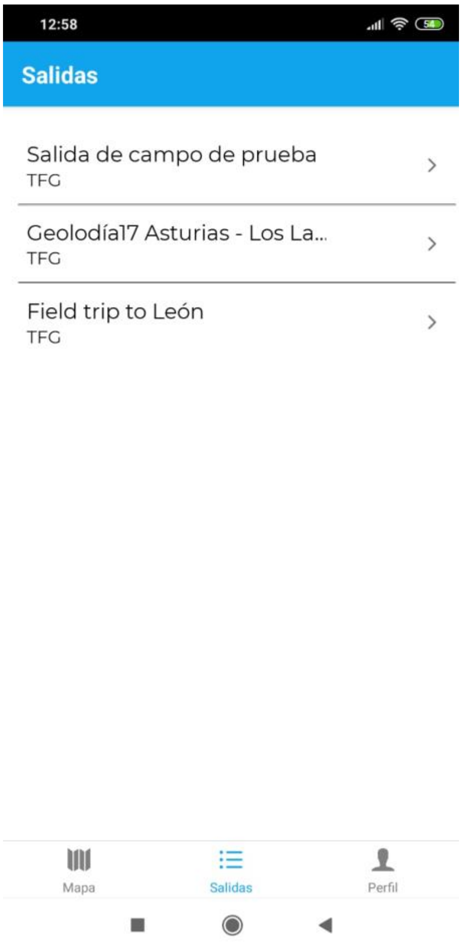
Illustration 25. Home page.

The home page contains:

- The map view containing the markers of all the locations of the task of the student
- Marker with the current location of the user
- Tab navigation to go to the field trips list and the profile screen

6.4.3. Field trips screen

The field trips screen shows the list of the field trips to which the student is assigned to. In the list, the name of the field trip and the group of students that is assigned to it is displayed. When an item of the list is clicked, the app navigates to the screen containing the detail of the field trip.



Salida de campo de prueba
TFG

Geolodía17 Asturias - Los La...
TFG

Field trip to León
TFG

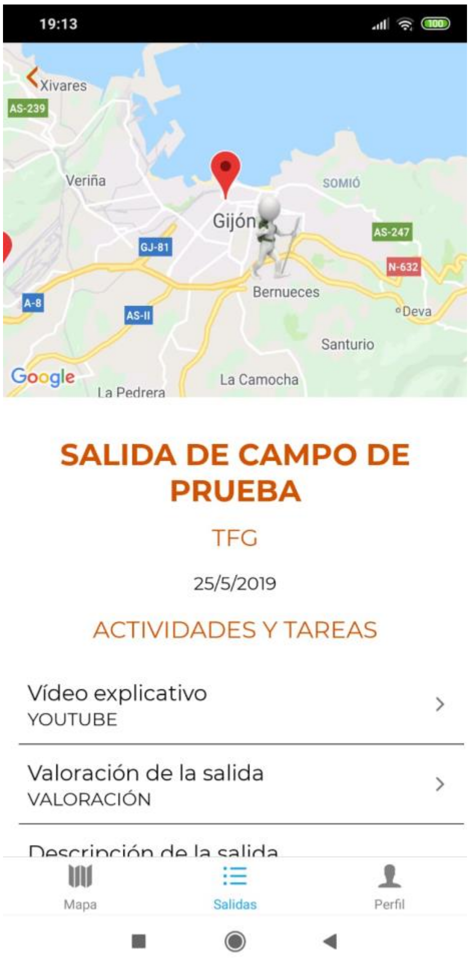
Illustration 26. Field trips list screen.

6.4.4. Field trip details screen

The page containing the field trip details have the following components:

- A reduced map view focused on the area where the field trip is located with the markers of the task locations and the current location of the student
- The name of the field trip
- The group to which it is assigned
- The initial and final date of the field trip. If the field trip is performed in one day, only that date is rendered in screen.
- The list of the tasks contained in the field trip
    - Each item of the list displays the title and the type of the task

- If the user clicks on the item, the app will get the user to the task detail page



SALIDA DE CAMPO DE PRUEBA

TFG

25/5/2019

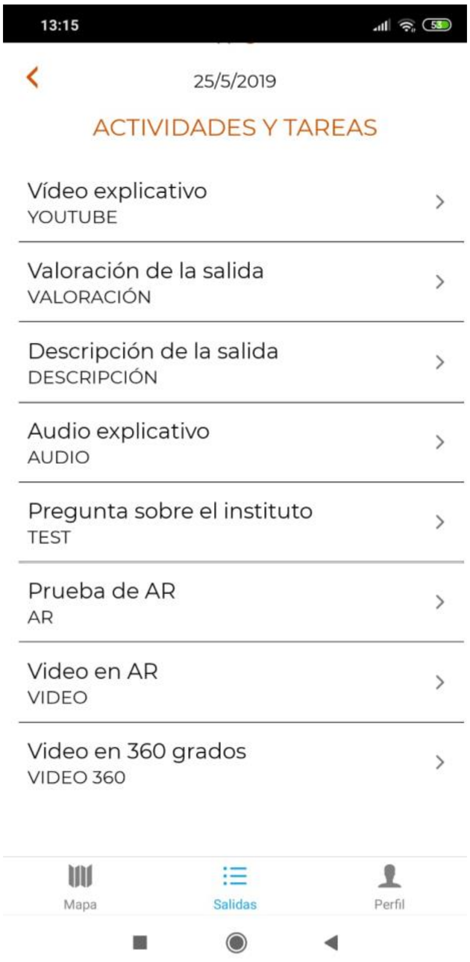
ACTIVIDADES Y TAREAS

Vídeo explicativo
YOUTUBE

Valoración de la salida
VALORACIÓN

Descripción de la salida

Illustration 28. Field trip detail.



25/5/2019

ACTIVIDADES Y TAREAS

Vídeo explicativo
YOUTUBE

Valoración de la salida
VALORACIÓN

Descripción de la salida
DESCRIPCIÓN

Audio explicativo
AUDIO

Pregunta sobre el instituto
TEST

Prueba de AR
AR

Video en AR
VIDEO

Video en 360 grados
VIDEO 360

Illustration 27. Tasks list in field trip detail.

6.4.5. Task detail screen

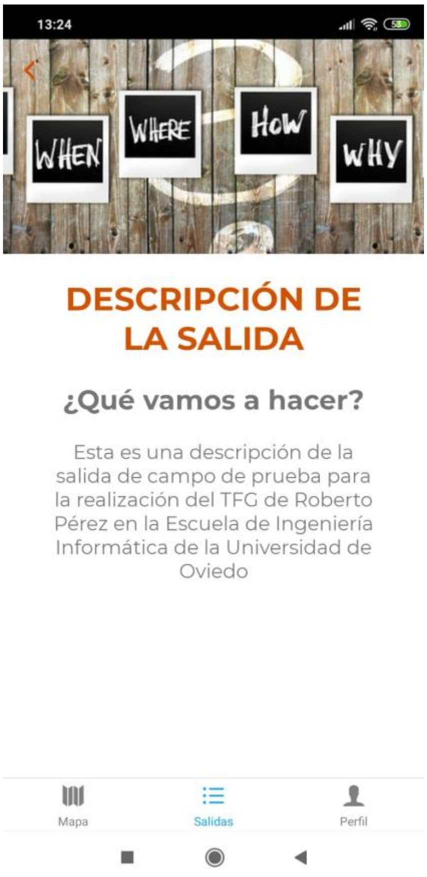
When one of the tasks in the list is clicked, the task detail page will be opened. Depending on the type of the task, a different activity will be rendered. However, all tasks have a common schema:

- Each task displays a related photo. Teachers can custom this photo including an image URL when creating the task. If there is not image URL in the task, a common photo will be displayed (the photos in the next screenshots of the task detail are the default ones)
- The title of the task, the description and the question related are also displayed in all tasks.

Then, depending on the type of each task, different components and behaviours are rendered in the device screen. If the user clicks on the back button, the app will navigate to the field trip detail screen.

6.4.5.1. Description task

This type of tasks does not add any other information to the task detail screen. It is used in order to add a global description to the field trip.



DESCRIPCIÓN DE LA SALIDA

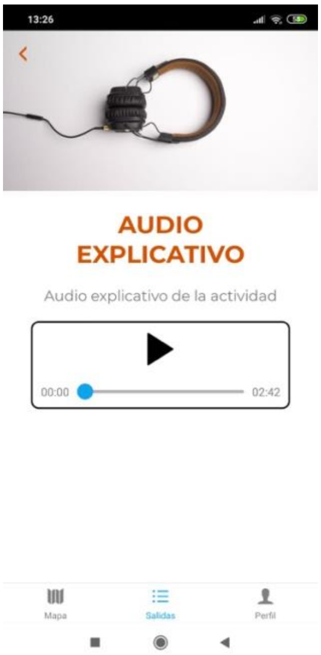
¿Qué vamos a hacer?

Esta es una descripción de la salida de campo de prueba para la realización del TFG de Roberto Pérez en la Escuela de Ingeniería Informática de la Universidad de Oviedo

Illustration 29. Description task.

6.4.5.2. Audio task

This type of tasks uses an audio player in order to give the student information about the task. The audio URL comes from the task form in the web application, and can be played and paused, as well as moved forward or backwards using the slider provided in the component.



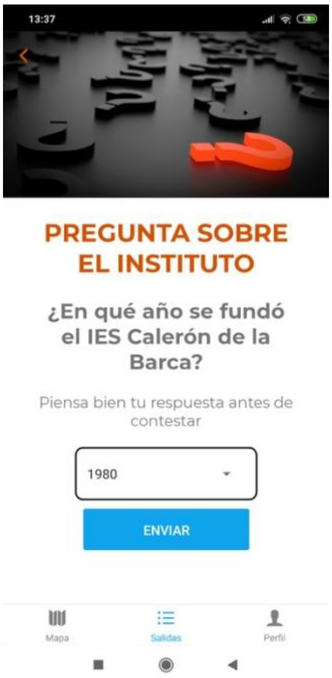
AUDIO EXPLICATIVO

Audio explicativo de la actividad

Illustration 30. Audio task.

6.4.5.3. Test task

For the test task type, the teacher will have to introduce the different options for the related question in the web application, as well as the correct one. Then, in the mobile app, these options will be rendered inside a select item, where the user can see all the options and select the one that fits the question better. If he answers correctly by clicking the submit button, the app will show a success message. Otherwise, it will show an error message with the correct answer.



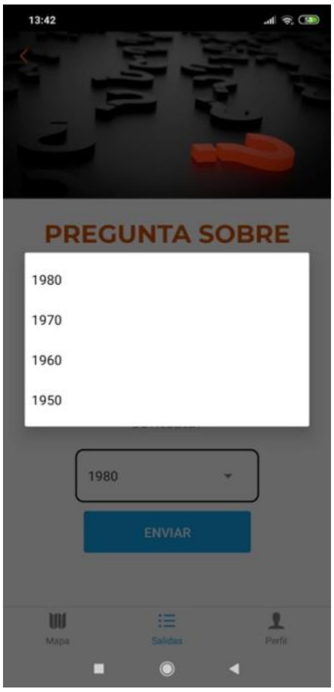
PREGUNTA SOBRE EL INSTITUTO

¿En qué año se fundó el IES Calerón de la Barca?

Piensa bien tu respuesta antes de contestar

1980

Illustration 32. Test task.



PREGUNTA SOBRE

1980

1970

1960

1950

1980

Illustration 31. Test task options.

6.4.5.4. Youtube task

This task is used in order to reproduce Youtube videos related to the field trip that is taking place. The teacher will introduce the video URL in the task form and the mobile app will display it in full screen.

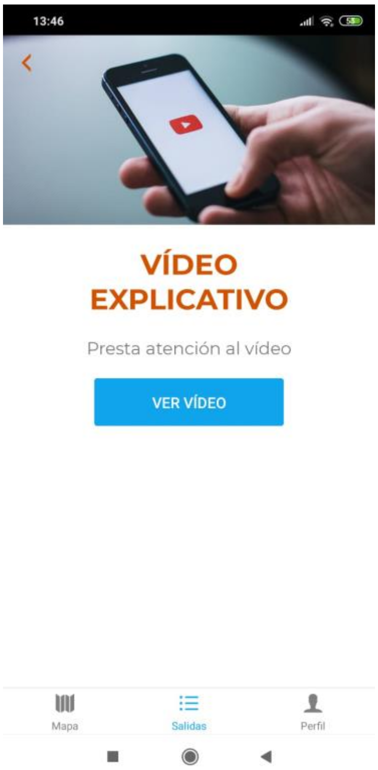


Illustration 33. Youtube task.

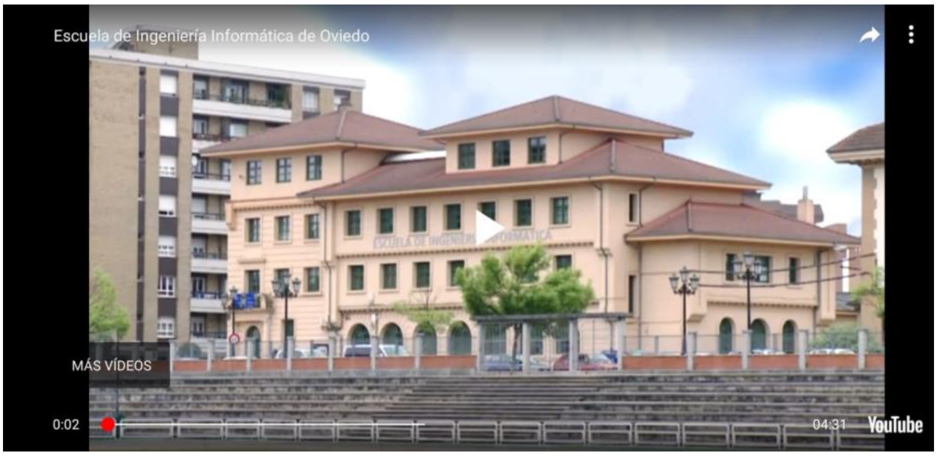
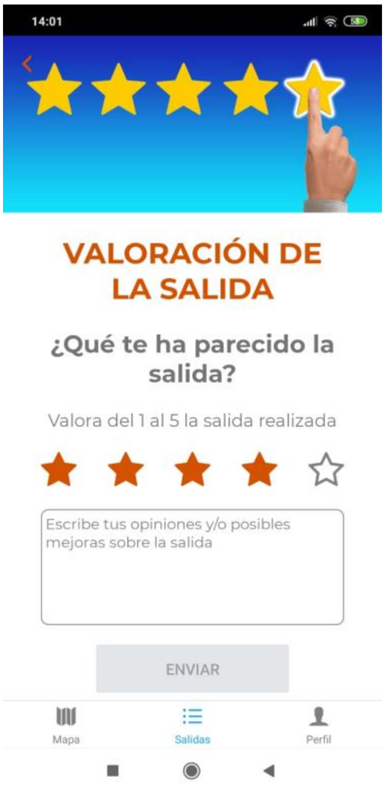


Illustration 34. Youtube video visualization.

6.4.5.5. *Rating and opinion task*

Using this task, the students can provide feedback about the field trip to their teachers. Students can rate the field trip from 1 to 5 using the stars, as well as send their opinion or suggestions about the field trip clicking on the submit button, which will be disabled until the students fills up the rating and opinion form. This information is sent to Ariane's database and teachers can treat this feedback in the web application.



VALORACIÓN DE LA SALIDA

¿Qué te ha parecido la salida?

Valora del 1 al 5 la salida realizada

★ ★ ★ ★ ☆

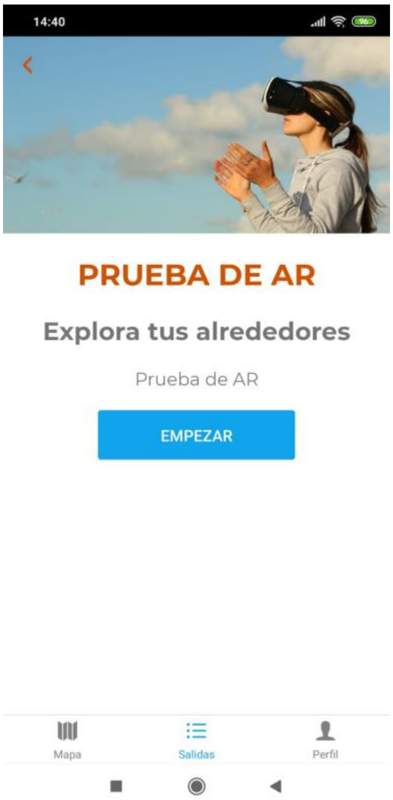
Escribe tus opiniones y/o posibles mejoras sobre la salida

Illustration 35. Rating task.

6.4.5.6. *Augmented reality tasks*

As the other tasks, the augmented reality ones will have its photo, title, description and the button to start the task. Ariane has three different augmented reality tasks:

- Text: the description of the task will be displayed as an augmented reality text



14:40

PRUEBA DE AR

Explora tus alrededores

Prueba de AR

Illustration 37. AR task.

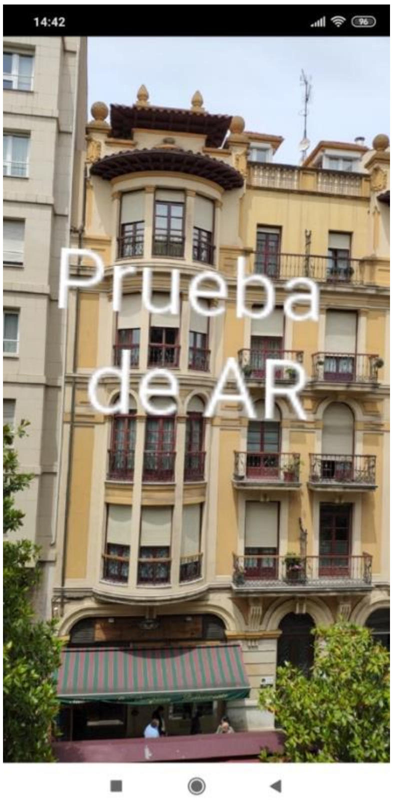
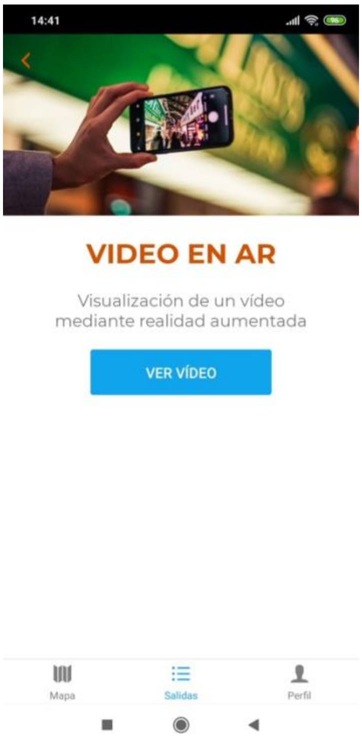


Illustration 36. AR task example.

- Video: the video provided by the teacher in the creation of the task will be displayed using AR



14:41

VIDEO EN AR

Visualización de un vídeo mediante realidad aumentada

Illustration 38. Video task.

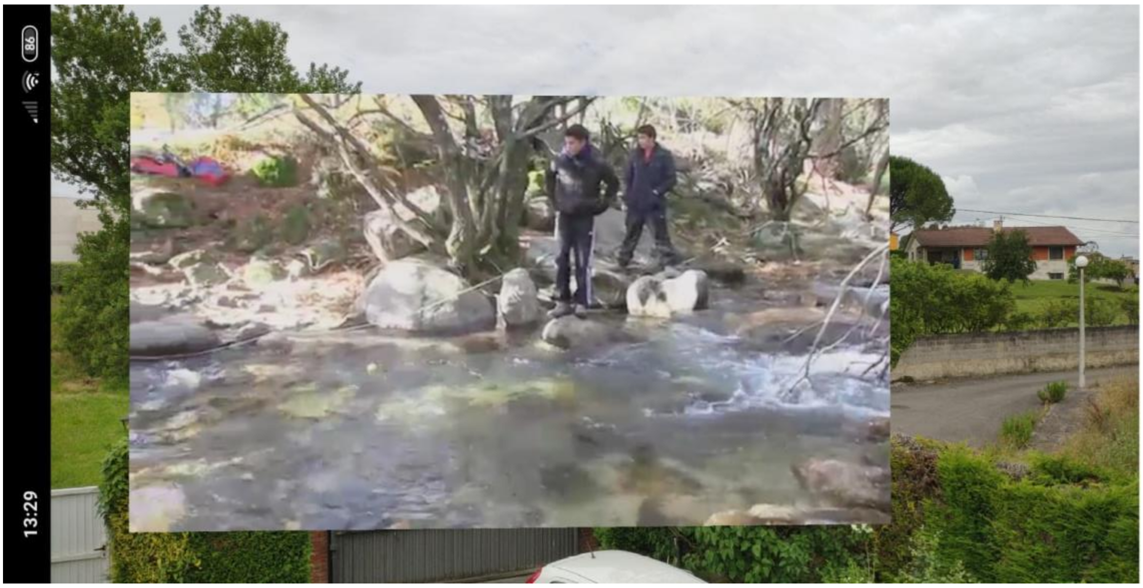
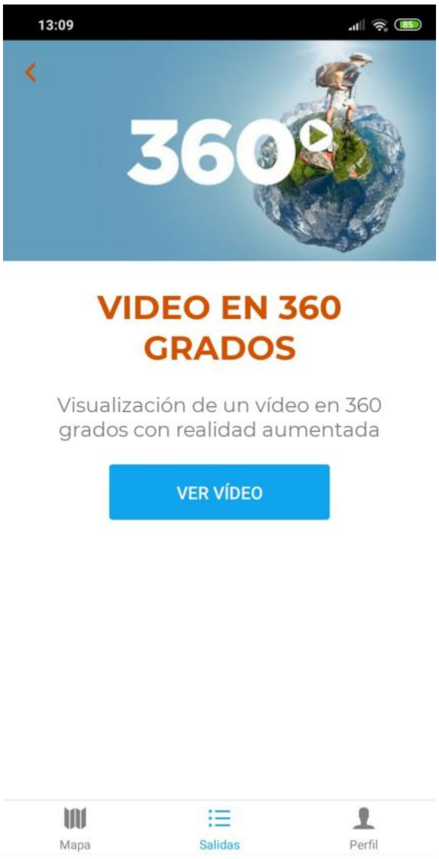


Illustration 39. Video in AR example.

- Video 360: a 360 degrees video will be displayed when this task is performed



VIDEO EN 360 GRADOS

Visualización de un vídeo en 360 grados con realidad aumentada

Illustration 42. Video 360 task.

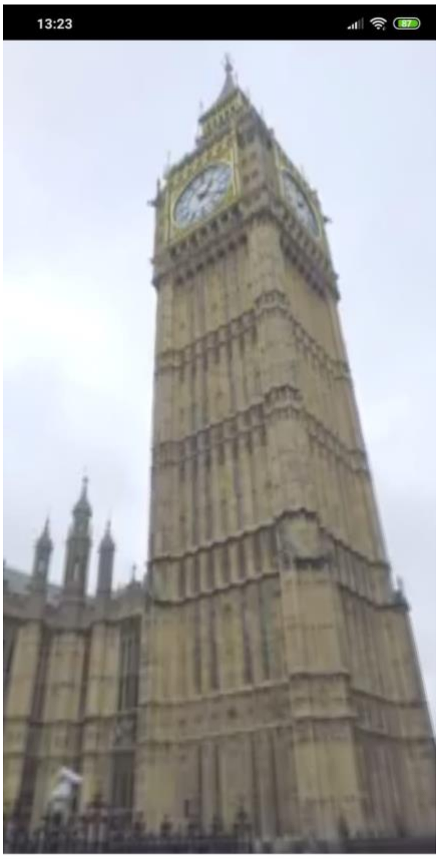


Illustration 41. Angle from the 360 video (London tour).

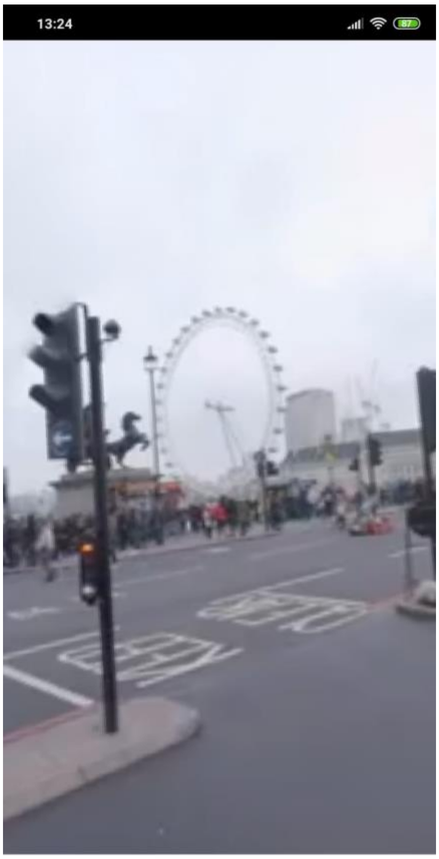


Illustration 40. Angle from the 360 video (London tour).

6.4.6. Profile screen

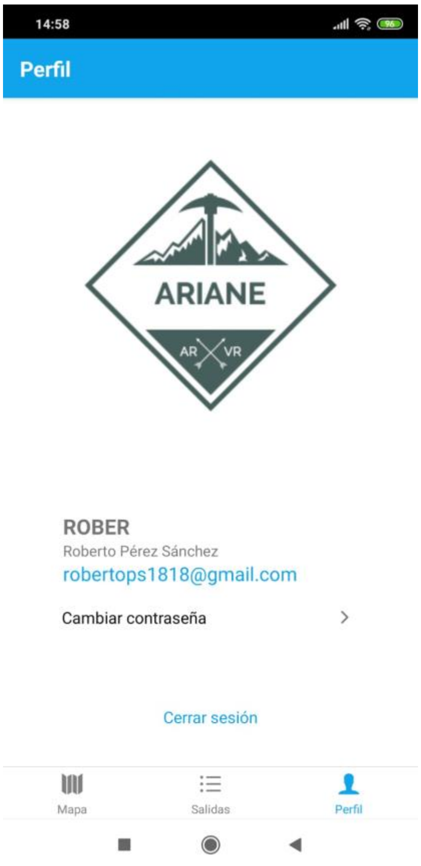
The profile screen contains the following components:

- Logo of Ariane
- Student’s username in capital letters
- Student’s complete name
- Student’s email
- Link to the change password screen
- Label in order to log out from the app

6.4.7. Change password screen

This screen is used in order to change the password of the current logged in user. It has the following components:

- Logo of Ariane
- Input for the old password
- Input for the new password
- Input for repeating the new password
- Submit button for changing the password
- Back button for going back to the profile screen



ROBER
Roberto Pérez Sánchez
robertops1818@gmail.com

Illustration 44. Profile screen.

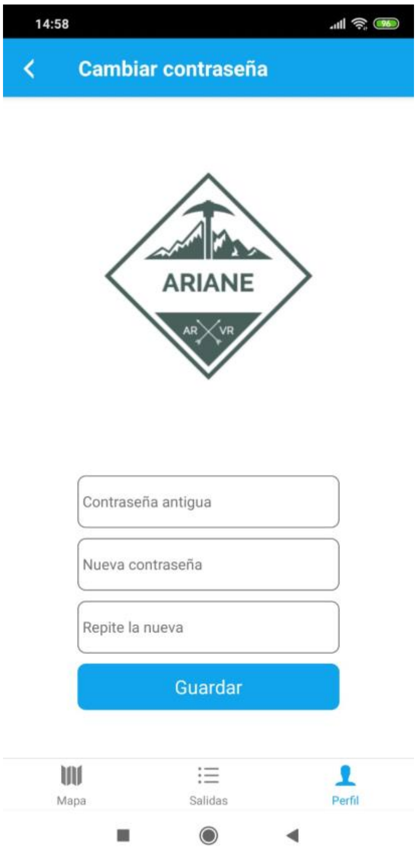


Illustration 43. Change password screen.

6.4.8. Web Application

The user interfaces of the web application are the same for all entities, the only change would be the different fields that each of them has. For that reason, only a general overview of the different views will be explained in this chapter instead of going entity per entity and showing the minimal differences between each view.

The first page that a teacher or admin sees is the login page, which contains:

- Two inputs for the username and password of the user
- The submit button for logging in
- A ‘Forgotten password?’ label in order to go to the recovery password page

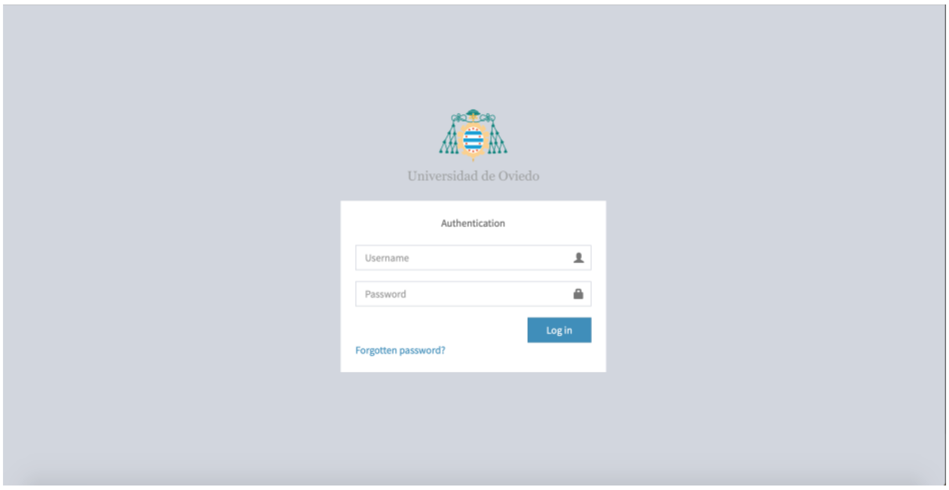


Illustration 45. Web app login page.

The recovery page of the web application has:

- Text input to introduce the email
- Reset password button in order to receive the new password at the email previously introduced

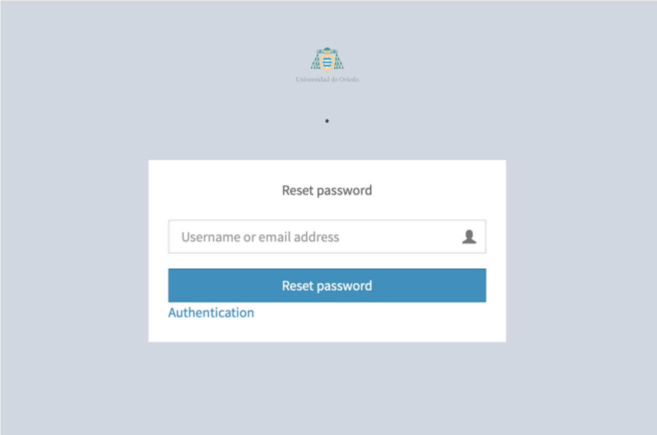


Illustration 46. Web app password recovery page.

Once the user has logged in correctly to the web application, the home page will be rendered.

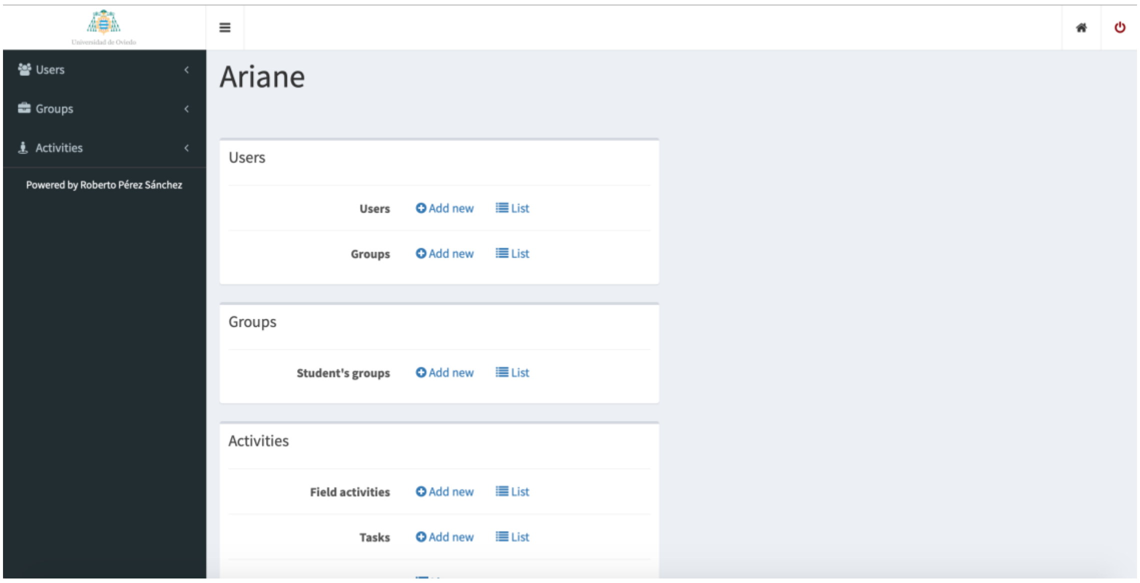


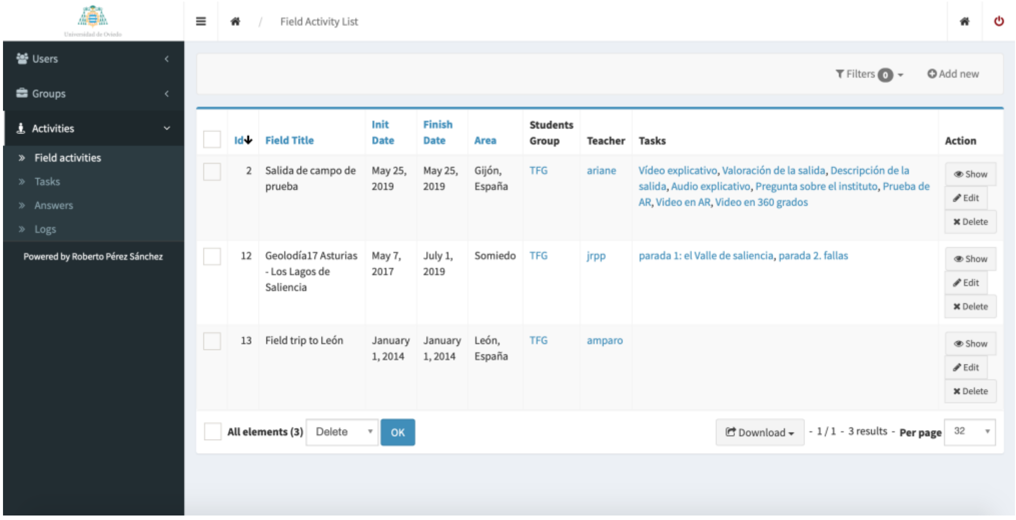
Illustration 47. Web app home page.

In the home page we can see two main sections:

- Side menu, with the links to the different parts of the web separated by entities
- The main dashboard, where the different entities can be accessed too

The entities that are displayed differ depending on the role of the authenticated user. For admins, all the entities will be visible, but for teachers, only the ones related to the field trips (field activity, task, answer and log) will be available.

The entity list page will be something like this:



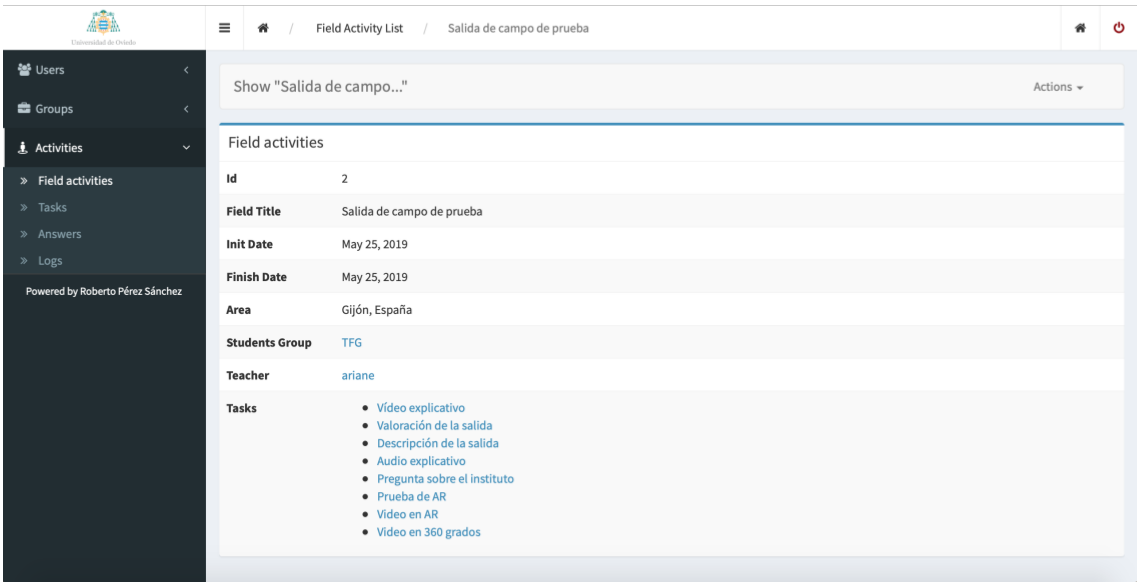
□	Id↓	Field Title	Init Date	Finish Date	Area	Students Group	Teacher	Tasks	Action
□	2	Salida de campo de prueba	May 25, 2019	May 25, 2019	Gijón, España	TFG	ariane	Vídeo explicativo, Valoración de la salida, Descripción de la salida, Audio explicativo, Pregunta sobre el instituto, Prueba de AR, Video en AR, Video en 360 grados	Show Edit Delete
□	12	Geolodía17 Asturias - Los Lagos de Saliencia	May 7, 2017	July 1, 2019	Somiedo	TFG	jrpp	parada 1: el Valle de saliencia, parada 2. fallas	Show Edit Delete
□	13	Field trip to León	January 1, 2014	January 1, 2014	León, España	TFG	amparo		Show Edit Delete

□ All elements (3) Delete OK
Download - 1 / 1 - 3 results - Per page 32

Illustration 48. Web app list page.

In this page, the web application will display the different fields in that entity. For example, for the field trips, all the created ones are displayed, including their fields: title, init and finish date, area, student group which is associated to it, the teacher responsible of the trip and the different tasks that compose the field trip.

If the user wants, to see the detail of an individual field trip, clicking in the 'Show' button, this details page will appear, displaying all the fields of the entity in a more visual and hierarchical way.



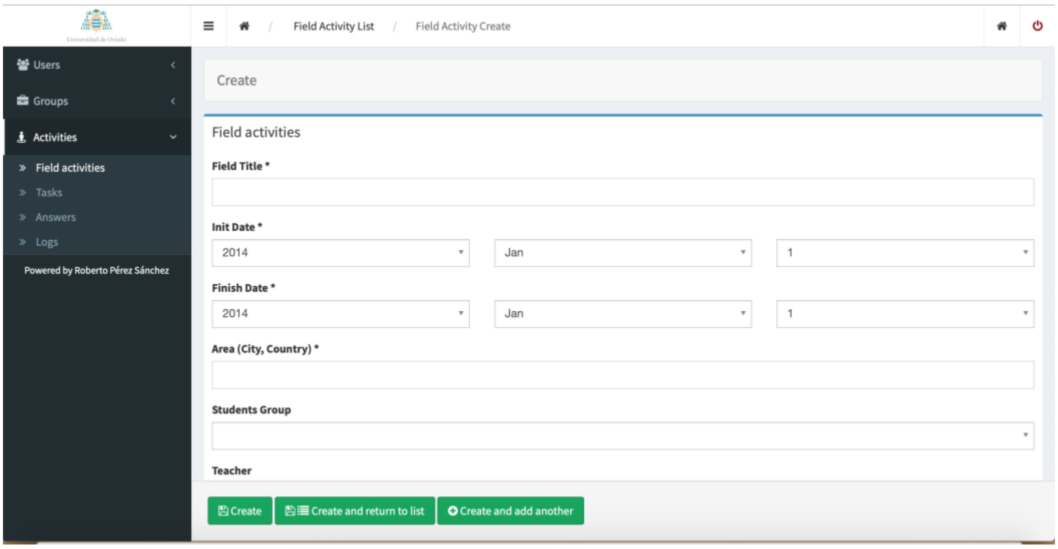
Show "Salida de campo..."

Field activities

Id	2
Field Title	Salida de campo de prueba
Init Date	May 25, 2019
Finish Date	May 25, 2019
Area	Gijón, España
Students Group	TFG
Teacher	ariane
Tasks	Vídeo explicativo Valoración de la salida Descripción de la salida Audio explicativo Pregunta sobre el instituto Prueba de AR Video en AR Video en 360 grados

Illustration 49. Web app detail page.

If the user wants to add a new field trip, he has to press the 'Add new' label in the list page. The form will contain the necessary fields in order to create the entity, and it will look like this:



Create

Field activities

Field Title *

Init Date *
2014 | Jan | 1

Finish Date *
2014 | Jan | 1

Area (City, Country) *

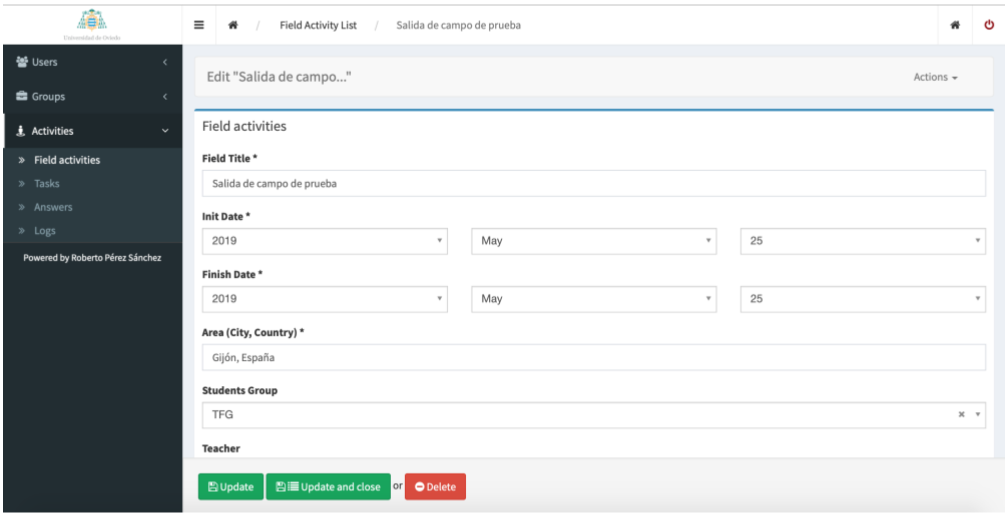
Students Group

Teacher

Illustration 50. Web app form page.

In the form, the fields with an asterisk after the label are the ones that are compulsory in order to create the new entity. When the user wants to save the entity, he clicks on any of the three bottom buttons and the entity will be saved. Depending on the button clicked, the app will stay on the form page ('Create'), returns to the list page ('Create and return to list') or initialize another empty form ('Create and add another').

Then, if the user wants to edit some of the fields of a field trip by clicking on 'Edit' in the list page, the app will show the form page, but filled with the initial data. Doing this, the user only has to edit the field which he is interested in.



Edit "Salida de campo..."

Field activities

Field Title *: Salida de campo de prueba

Init Date *: 2019 | May | 25

Finish Date *: 2019 | May | 25

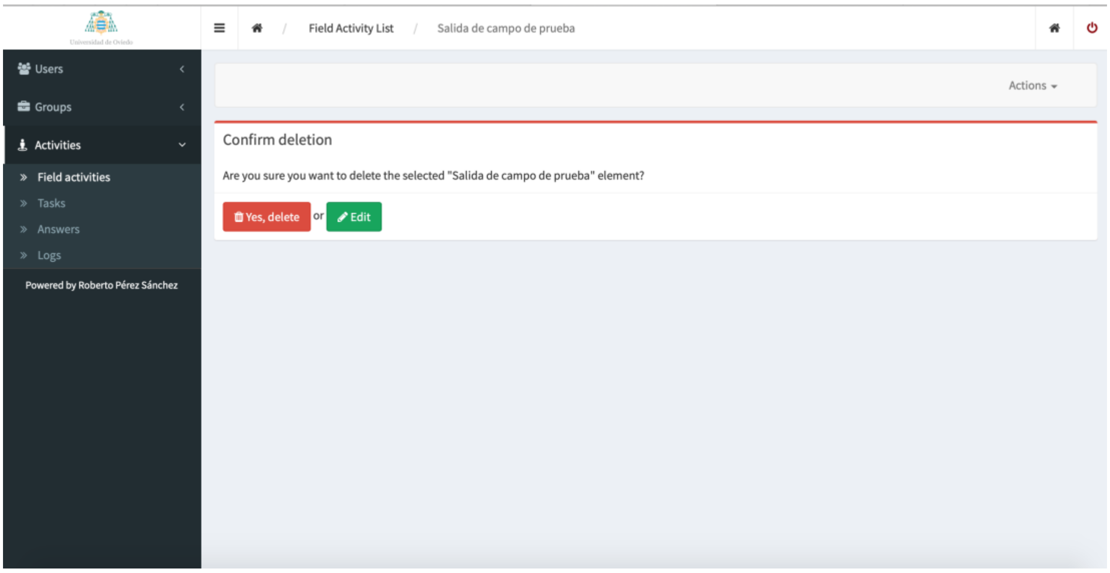
Area (City, Country) *: Gijón, España

Students Group: TFG

Teacher

Illustration 51. Web app edit page.

The bottom buttons change when the form is editing an entity. The user can choose between updating it and stay in the same page, return to the list or even delete the entity. If the user clicks on delete from here or from the list page, a confirm page will be shown in case the user has clicked the delete button unintentionally.

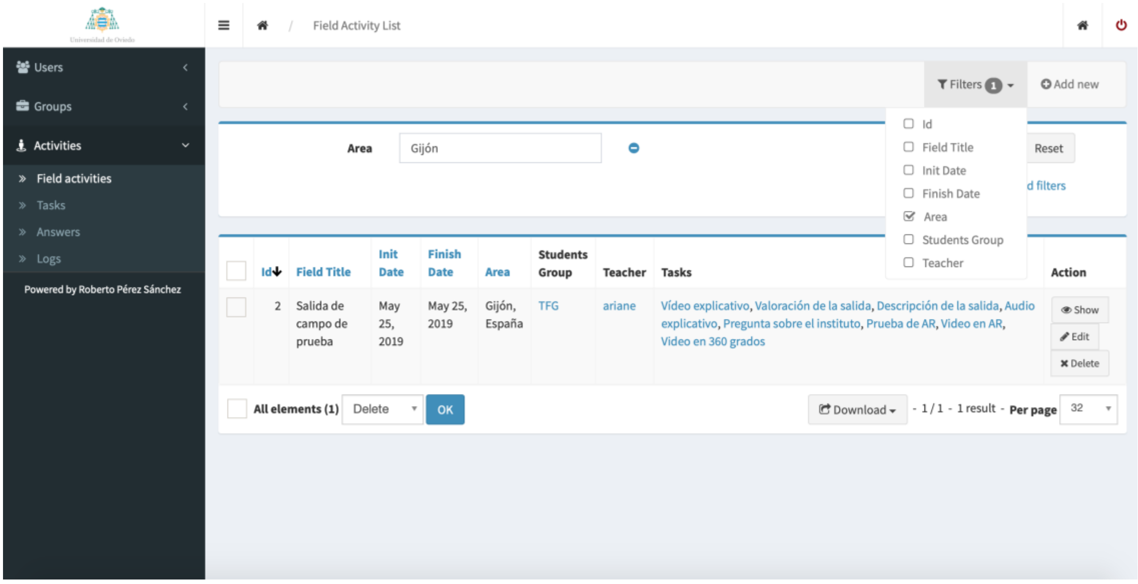


Confirm deletion

Are you sure you want to delete the selected "Salida de campo de prueba" element?

Illustration 52. Web app confirm delete page.

In every list page, if the user clicks on the 'Filters' label, a list of the available filters for the entity will appear, where the user can choose to activate one or another.



Field Activity List

Users
Groups
Activities
- Field activities
- Tasks
- Answers
- Logs

Powered by Roberto Pérez Sánchez

Filters 1 | Add new

□ Id
□ Field Title
□ Init Date
□ Finish Date
☑ Area
□ Students Group
□ Teacher

Area: Gijón

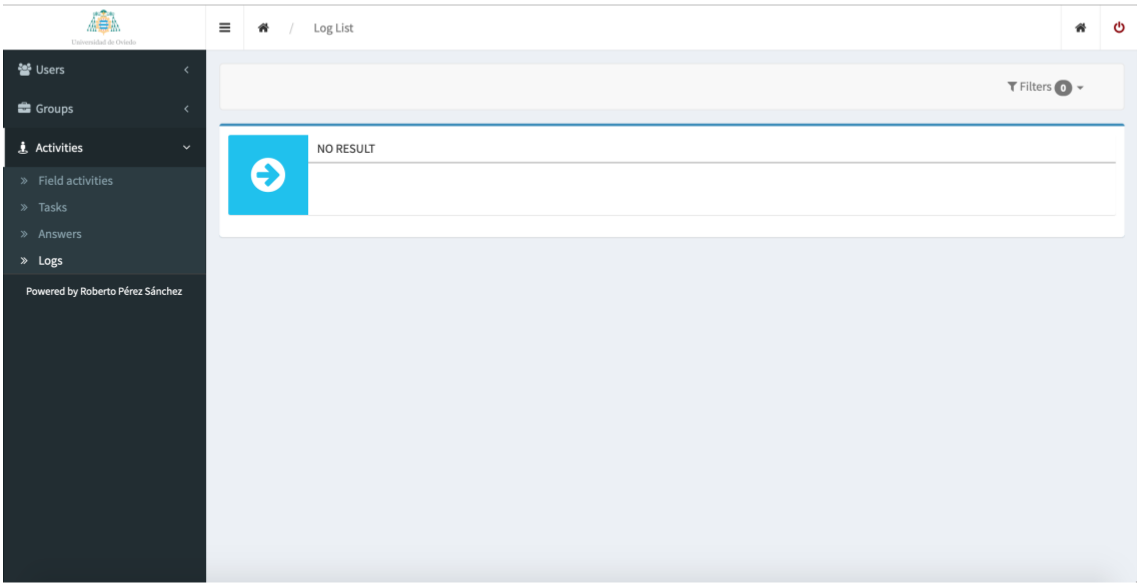
Reset
d filters

□	Id	Field Title	Init Date	Finish Date	Area	Students Group	Teacher	Tasks	Action
□	2	Salida de campo de prueba	May 25, 2019	May 25, 2019	Gijón, España	TFG	ariane	Vídeo explicativo, Valoración de la salida, Descripción de la salida, Audio explicativo, Pregunta sobre el instituto, Prueba de AR, Video en AR, Video en 360 grados	Show Edit Delete

□ All elements (1) Delete OK | Download - 1 / 1 - 1 result - Per page 32

Illustration 53. Web app filters.

Finally, if some of the entity lists is empty, the app will show a page with a 'No result' label.



Log List

Users
Groups
Activities
- Field activities
- Tasks
- Answers
- Logs

Powered by Roberto Pérez Sánchez

Filters 0

NO RESULT

Illustration 54. Web app empty list page.

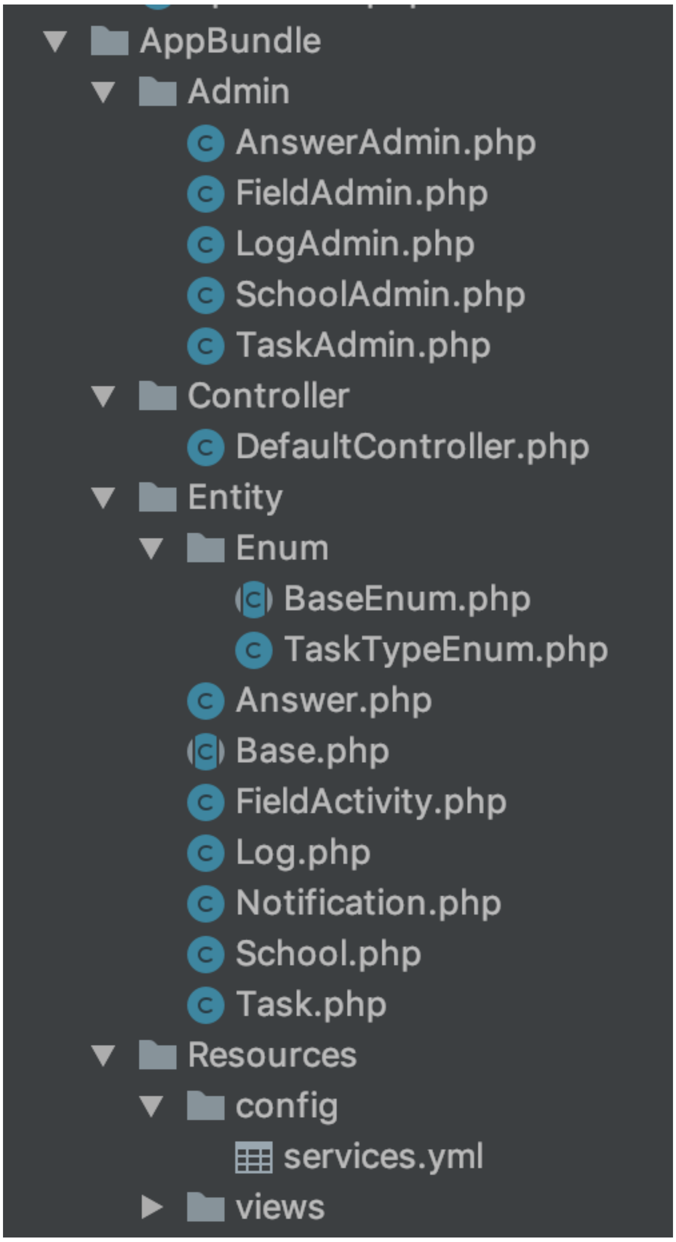
7. Implementation

7.1. System structure

7.1.1. Web Application

As explained in the package diagram of the web application, it is separated in two different modules: AppBundle for the entities and actions of the web application, and ApiBundle, for the creation of the API methods used by the mobile app.

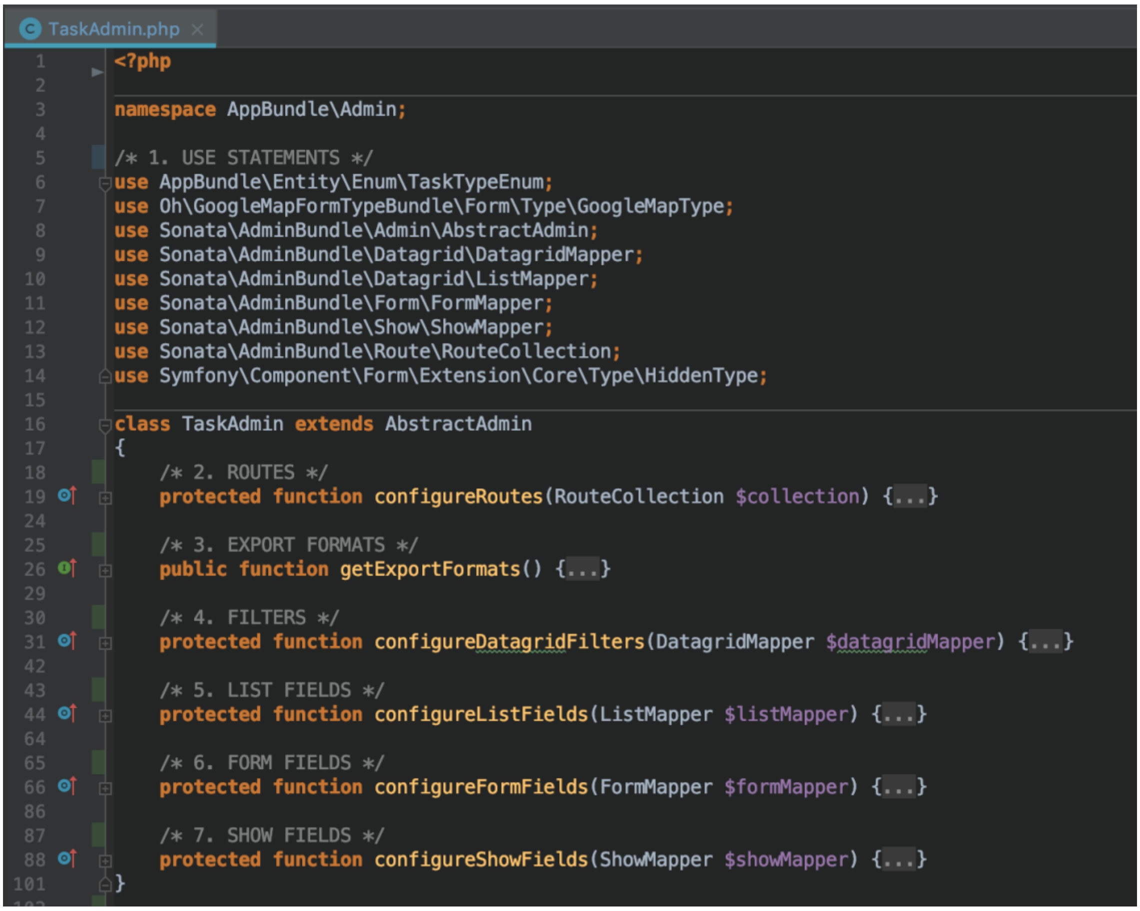
7.1.1.1. AppBundle



```
▼ AppBundle
    ▼ Admin
          AnswerAdmin.php
          FieldAdmin.php
          LogAdmin.php
          SchoolAdmin.php
          TaskAdmin.php
    ▼ Controller
          DefaultController.php
    ▼ Entity
        ▼ Enum
              BaseEnum.php
              TaskTypeEnum.php
          Answer.php
          Base.php
          FieldActivity.php
          Log.php
          Notification.php
          School.php
          Task.php
    ▼ Resources
        ▼ config
              services.yml
        ▶ views
```

Illustration 55. AppBundle hierarchy.

As explained in chapter 6.3.1, the entities of the system are commented with the ORM annotations in order to store their structure into tables in the database. Then, an Admin file is created for each of them in order to manage all the operations related to that entity. In order to explain the structure of the admin files, we are going to tackle the Task entity.



```php
<?php

namespace AppBundle\Admin;

/* 1. USE STATEMENTS */
use AppBundle\Entity\Enum\TaskTypeEnum;
use Oh\GoogleMapFormTypeBundle\Form\Type\GoogleMapType;
use Sonata\AdminBundle\Admin\AbstractAdmin;
use Sonata\AdminBundle\Datagrid\DatagridMapper;
use Sonata\AdminBundle\Datagrid\ListMapper;
use Sonata\AdminBundle\Form\FormMapper;
use Sonata\AdminBundle\Show\ShowMapper;
use Sonata\AdminBundle\Route\RouteCollection;
use Symfony\Component\Form\Extension\Core\Type\HiddenType;

class TaskAdmin extends AbstractAdmin
{
    /* 2. ROUTES */
    protected function configureRoutes(RouteCollection $collection) {...}

    /* 3. EXPORT FORMATS */
    public function getExportFormats() {...}

    /* 4. FILTERS */
    protected function configureDatagridFilters(DatagridMapper $datagridMapper) {...}

    /* 5. LIST FIELDS */
    protected function configureListFields(ListMapper $listMapper) {...}

    /* 6. FORM FIELDS */
    protected function configureFormFields(FormMapper $formMapper) {...}

    /* 7. SHOW FIELDS */
    protected function configureShowFields(ShowMapper $showMapper) {...}
}
```

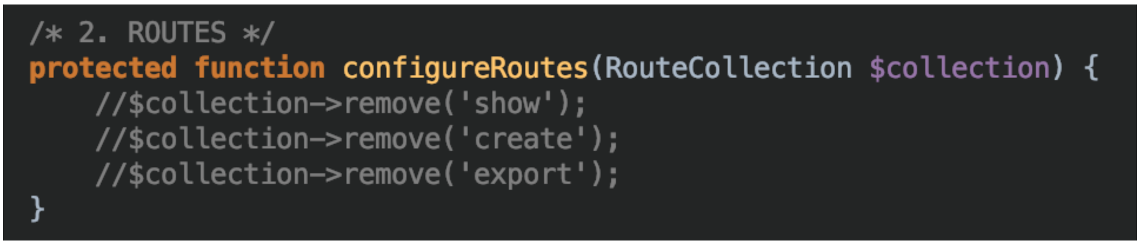
Illustration 56. Admin structure.

7.1.1.1.1. Use statements

This part of the PHP file is similar to the imports part of any programming language, where the providers and services used by the class are declared for its use.

7.1.1.1.2. Routes

In the method configureRoutes, the developer can restrict the action that can be made to the entity (delete, show, edit…). In this case, for the tasks, none of them are restricted, but in the answers, they cannot be edited nor created. As it can be seen below, for restricting any of the action, the developer just needs to uncomment the line of the action he wants to restrict.

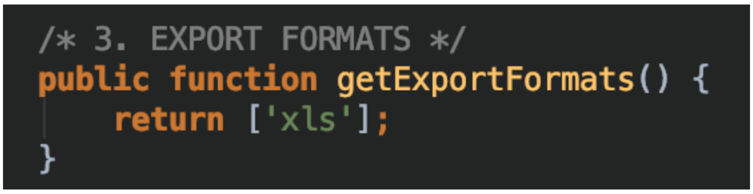


```
/* 2. ROUTES */
protected function configureRoutes(RouteCollection $collection) {
    //$collection->remove('show');
    //$collection->remove('create');
    //$collection->remove('export');
}
```

Illustration 57. ConfigureRoutes admin.

7.1.1.1.3. Export formats

In the getExportFormats method, the type of files available for the download of the entity list is configured. In this case, the only available format is Excel (xls).

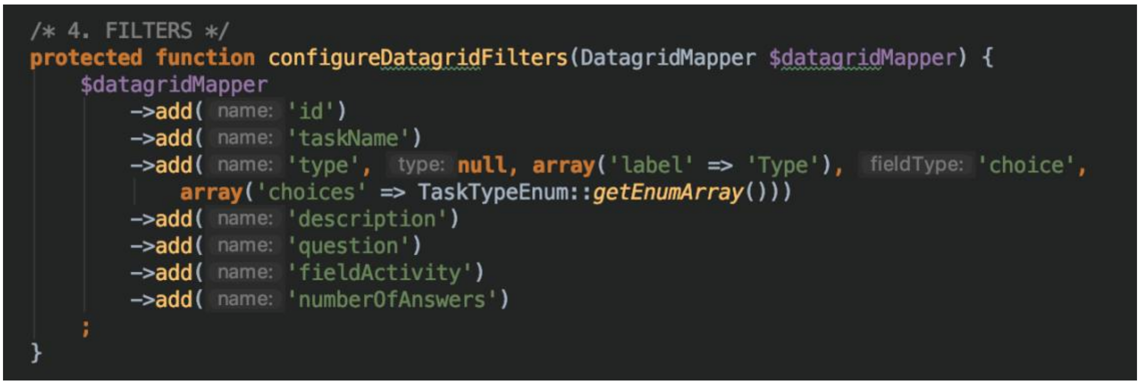


```
/* 3. EXPORT FORMATS */
public function getExportFormats() {
    return ['xls'];
}
```

Illustration 58. Admin export formats.

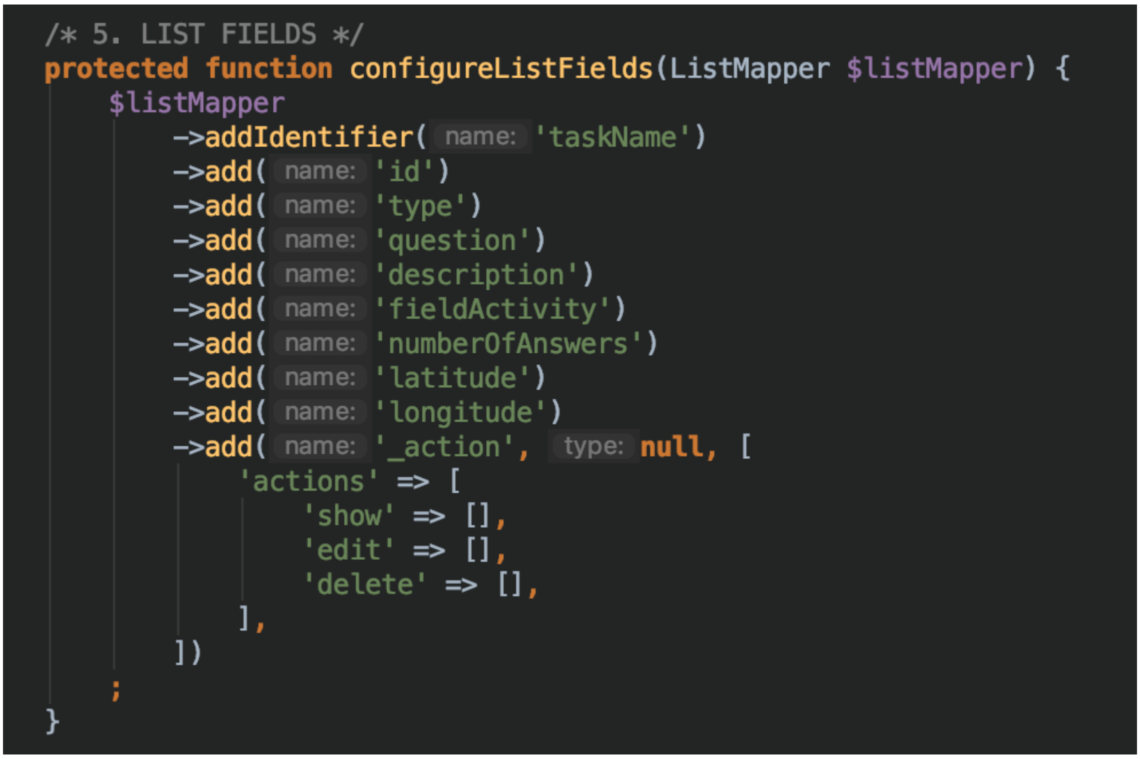
7.1.1.1.4. Filters, list, form and show

These four actions related to the entities work in a very similar way. Each of them has a related method that adds the fields of the entity that are going to take part in the operations, for example, the form fields for the creation or the filters available for filtering the data stored in the database. In every method, there is a mapper that stores the fields required and other optional properties as the label of the field or its type. All the names added to the mapper have to be the same as the name of the fields declared in the entity.



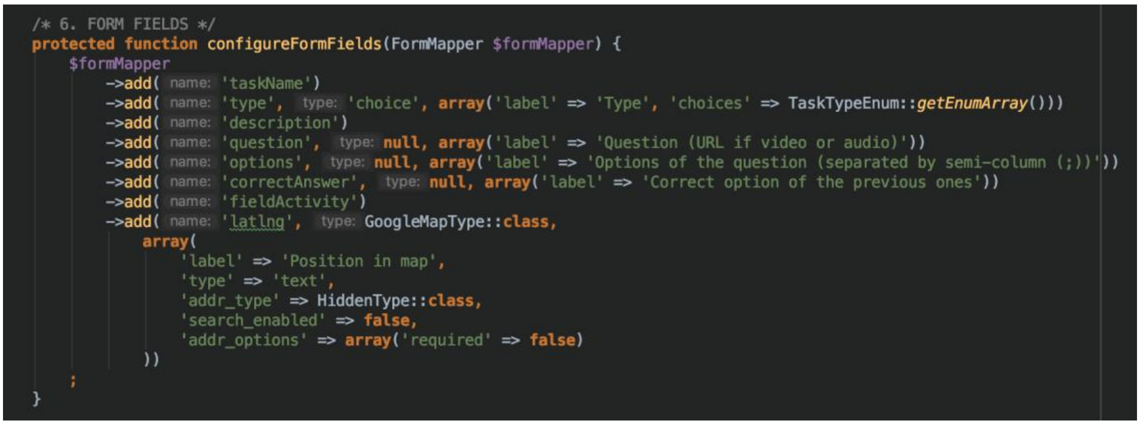
```
/* 4. FILTERS */
protected function configureDatagridFilters(DatagridMapper $datagridMapper) {
    $datagridMapper
        ->add( name: 'id')
        ->add( name: 'taskName')
        ->add( name: 'type',  type: null, array('label' => 'Type'),  fieldType: 'choice',
            array('choices' => TaskTypeEnum::getEnumArray()))
        ->add( name: 'description')
        ->add( name: 'question')
        ->add( name: 'fieldActivity')
        ->add( name: 'numberOfAnswers')
    ;
}
```

Illustration 59. Admin filters.



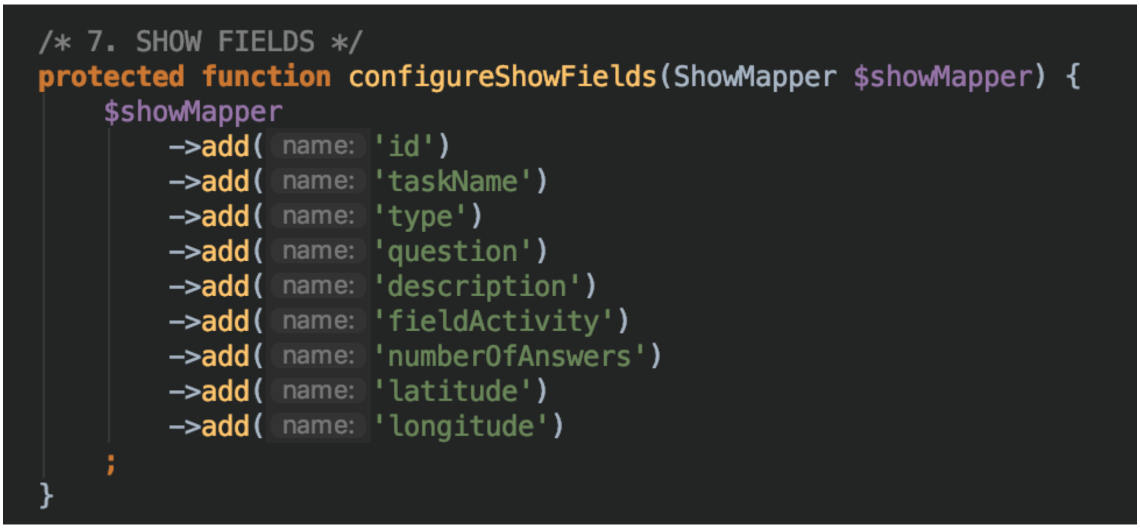
```
/* 5. LIST FIELDS */
protected function configureListFields(ListMapper $listMapper) {
    $listMapper
        ->addIdentifier( name: 'taskName')
        ->add( name: 'id')
        ->add( name: 'type')
        ->add( name: 'question')
        ->add( name: 'description')
        ->add( name: 'fieldActivity')
        ->add( name: 'numberOfAnswers')
        ->add( name: 'latitude')
        ->add( name: 'longitude')
        ->add( name: '_action', type: null, [
            'actions' => [
                'show' => [],
                'edit' => [],
                'delete' => [],
            ],
        ])
    ;
}
```

Illustration 60. Admin listing.



```
/* 6. FORM FIELDS */
protected function configureFormFields(FormMapper $formMapper) {
    $formMapper
        ->add( name: 'taskName')
        ->add( name: 'type', type: 'choice', array('label' => 'Type', 'choices' => TaskTypeEnum::getEnumArray()))
        ->add( name: 'description')
        ->add( name: 'question', type: null, array('label' => 'Question (URL if video or audio)'))
        ->add( name: 'options', type: null, array('label' => 'Options of the question (separated by semi-column (;))'))
        ->add( name: 'correctAnswer', type: null, array('label' => 'Correct option of the previous ones'))
        ->add( name: 'fieldActivity')
        ->add( name: 'latlng', type: GoogleMapType::class,
            array(
                'label' => 'Position in map',
                'type' => 'text',
                'addr_type' => HiddenType::class,
                'search_enabled' => false,
                'addr_options' => array('required' => false)
            ))
    ;
}
```

Illustration 61. Admin form.

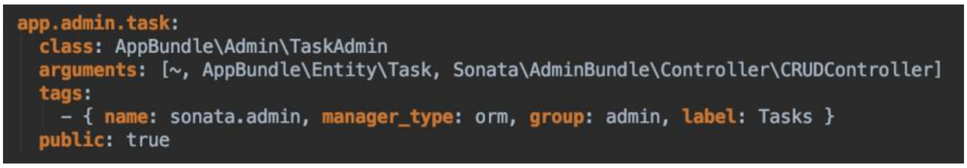


```
/* 7. SHOW FIELDS */
protected function configureShowFields(ShowMapper $showMapper) {
    $showMapper
        ->add( name: 'id')
        ->add( name: 'taskName')
        ->add( name: 'type')
        ->add( name: 'question')
        ->add( name: 'description')
        ->add( name: 'fieldActivity')
        ->add( name: 'numberOfAnswers')
        ->add( name: 'latitude')
        ->add( name: 'longitude')
    ;
}
```

Illustration 62. Admin show.

7.1.1.1.5. Use of the admins in the final web application

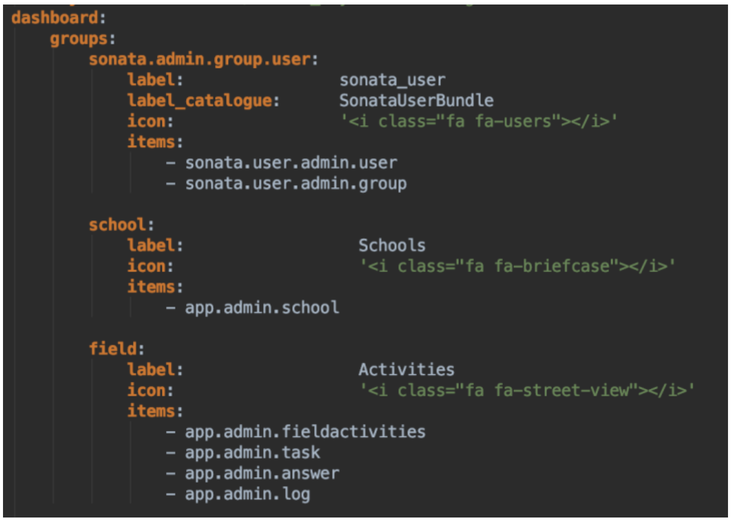
In order to make use of these admins, they need to be declared as a services.yml file of the Symfony project and added to the application in the config.yml. For the Tasks, it would be something like this regarding the services.yml file:



```
app.admin.task:
  class: AppBundle\Admin\TaskAdmin
  arguments: [~, AppBundle\Entity\Task, Sonata\AdminBundle\Controller\CRUDController]
  tags:
    - { name: sonata.admin, manager_type: orm, group: admin, label: Tasks }
  public: true
```

Illustration 63. TaskAdmin in services.yml.

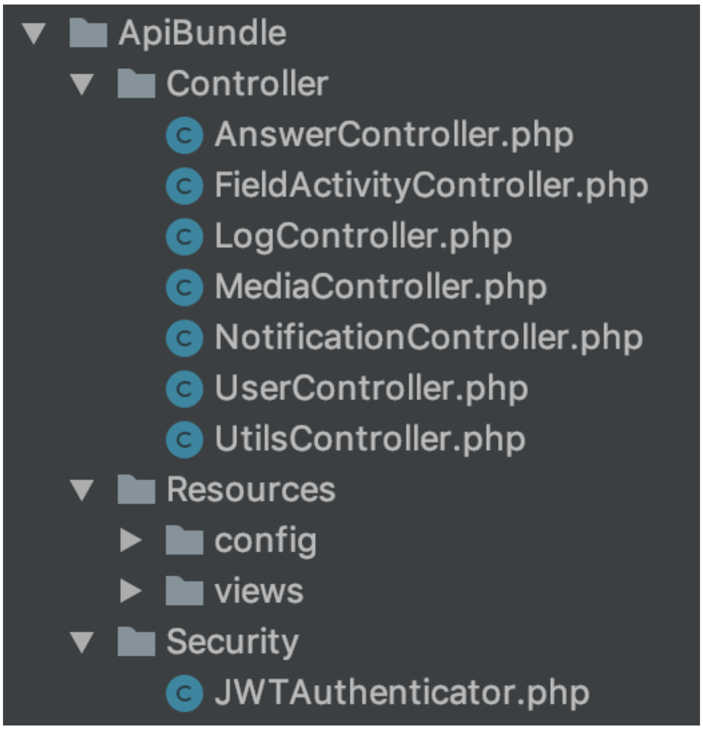
Then, in the config.yml, all the services declared are structured in what the final lateral menu of the application would be.



```
dashboard:
    groups:
        sonata.admin.group.user:
            label:                  sonata_user
            label_catalogue:        SonataUserBundle
            icon:                   '<i class="fa fa-users"></i>'
            items:
                - sonata.user.admin.user
                - sonata.user.admin.group

        school:
            label:                    Schools
            icon:                     '<i class="fa fa-briefcase"></i>'
            items:
                - app.admin.school

        field:
            label:                    Activities
            icon:                     '<i class="fa fa-street-view"></i>'
            items:
                - app.admin.fieldactivities
                - app.admin.task
                - app.admin.answer
                - app.admin.log
```

Illustration 64. Admins in config.yml.

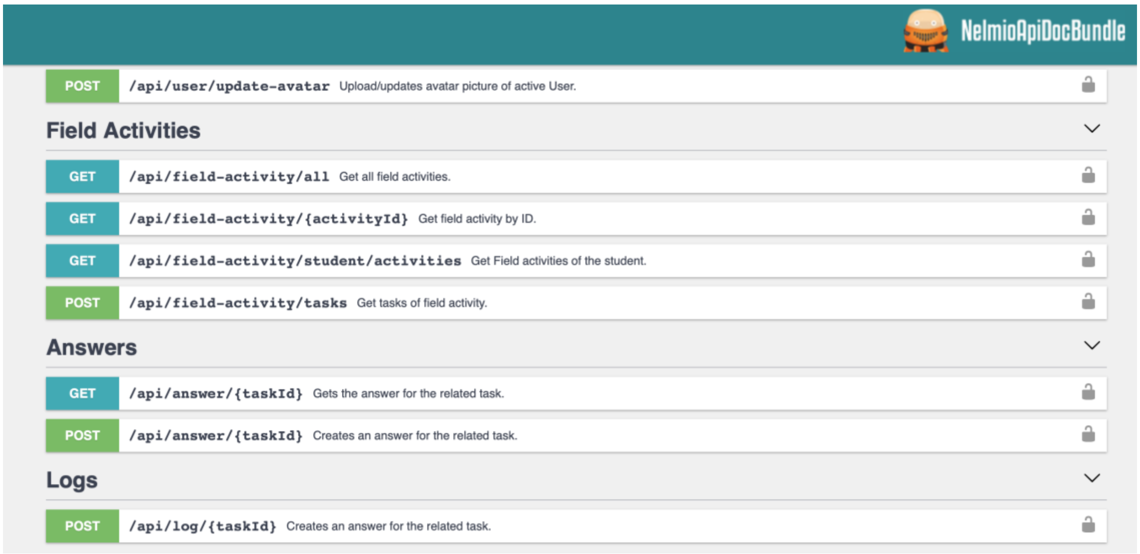
7.1.1.2. ApiBundle



```
▼ ApiBundle
  ▼ Controller
      AnswerController.php
      FieldActivityController.php
      LogController.php
      MediaController.php
      NotificationController.php
      UserController.php
      UtilsController.php
  ▼ Resources
    ▶ config
    ▶ views
  ▼ Security
      JWTAuthenticator.php
```

Illustration 65. ApiBundle hierarchy.

In the ApiBundle, there exists one Controller per entity that gets involved in the API REST. For example, all the methods of the API declared related to the answers would be declared in the AnswerController. All of the API methods can be tested and listed in http://ariane-uniovi.es/api/doc.



POST `/api/user/update-avatar` Upload/updates avatar picture of active User.

Field Activities

GET `/api/field-activity/all` Get all field activities.

GET `/api/field-activity/{activityId}` Get field activity by ID.

GET `/api/field-activity/student/activities` Get Field activities of the student.

POST `/api/field-activity/tasks` Get tasks of field activity.

Answers

GET `/api/answer/{taskId}` Gets the answer for the related task.

POST `/api/answer/{taskId}` Creates an answer for the related task.

Logs

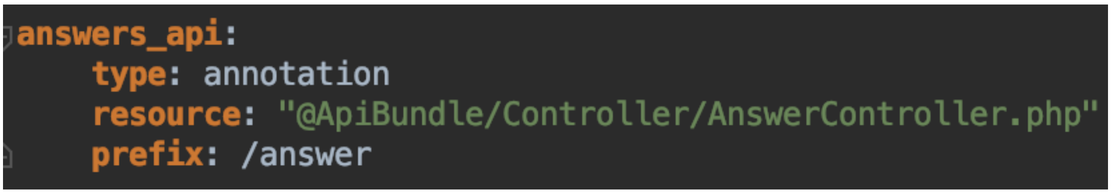
POST `/api/log/{taskId}` Creates an answer for the related task.

Illustration 66. API methods doc.

The methods inside each controller as declared as actions and have two main parts:

1. Annotations: In the annotations of the method it is declared the route of the method, the type of method (GET, POST, etc.), the different responses of the method (200, 400, 404), the parameters that receives, the tag where is located in the documentation and the security level.
2. Method: The functionality to be executed when the method is called. In the parameters of the function are also declared the URL parameters of the API method.

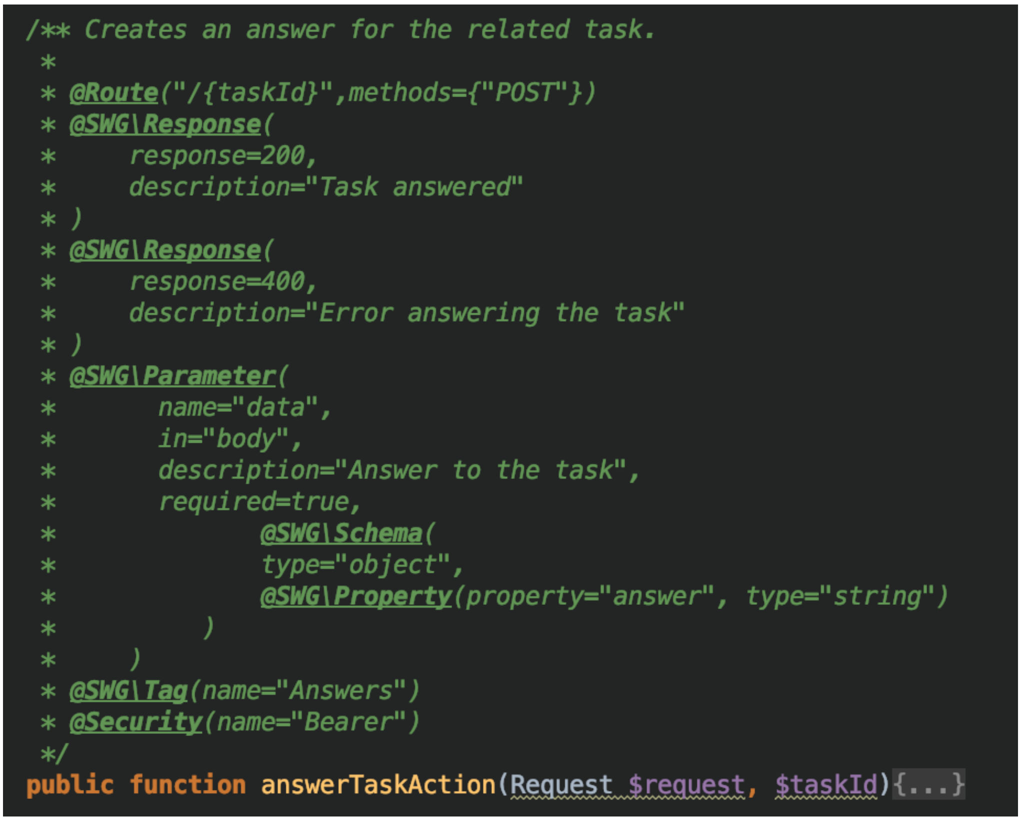
Each controller is declated in the routing.yml file of the project, where the endpoints of each controller are declared.



```
answers_api:
    type: annotation
    resource: "@ApiBundle/Controller/AnswerController.php"
    prefix: /answer
```

Illustration 67. AnswerController in routing.yml.

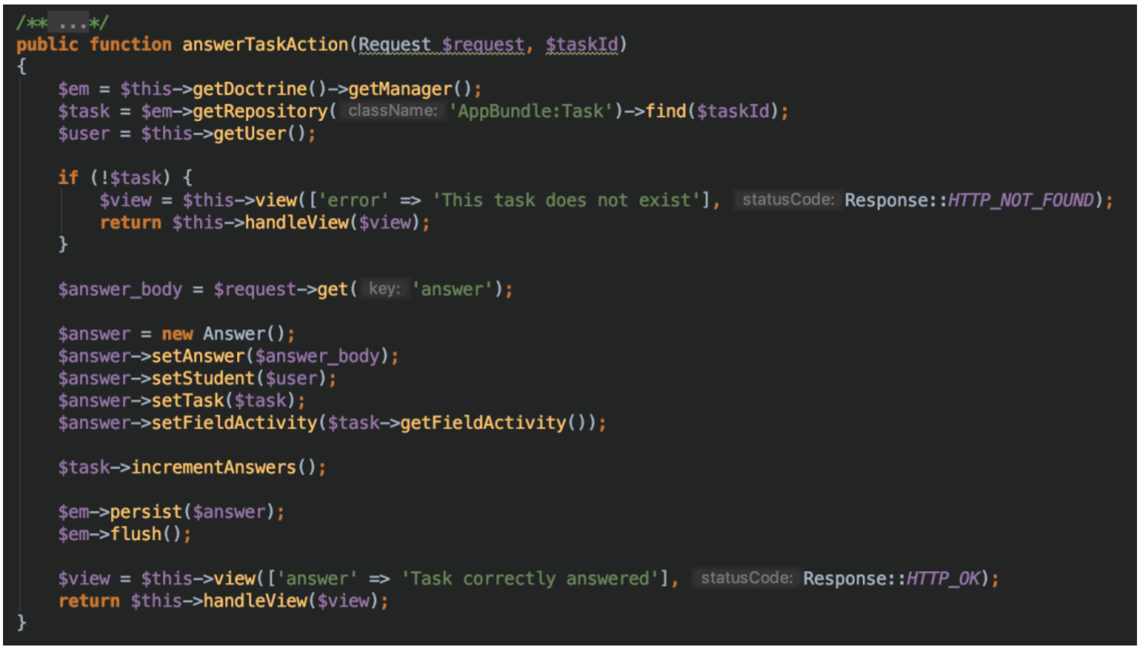
For example, in the AnswerController, there is a method in order to answer to the task in the mobile application.



```
/** Creates an answer for the related task.
 *
 * @Route("/{taskId}",methods={"POST"})
 * @SWG\Response(
 *     response=200,
 *     description="Task answered"
 * )
 * @SWG\Response(
 *     response=400,
 *     description="Error answering the task"
 * )
 * @SWG\Parameter(
 *       name="data",
 *       in="body",
 *       description="Answer to the task",
 *       required=true,
 *              @SWG\Schema(
 *              type="object",
 *              @SWG\Property(property="answer", type="string")
 *          )
 *     )
 * @SWG\Tag(name="Answers")
 * @Security(name="Bearer")
 */
public function answerTaskAction(Request $request, $taskId){...}
```

Illustration 68. Annotation of the API method.

Taking into account the routing.yml and the @Route annotation, the final path of this method would be **ariane-uniovi.es/api/answer/{taskId}**, being the taskId the ID of the task to be answered. This method is a POST, and it receives one body parameter, that is the answer to the task.



```
/** ...*/
public function answerTaskAction(Request $request, $taskId)
{
    $em = $this->getDoctrine()->getManager();
    $task = $em->getRepository( className: 'AppBundle:Task')->find($taskId);
    $user = $this->getUser();

    if (!$task) {
        $view = $this->view(['error' => 'This task does not exist'],  statusCode: Response::HTTP_NOT_FOUND);
        return $this->handleView($view);
    }

    $answer_body = $request->get( key: 'answer');

    $answer = new Answer();
    $answer->setAnswer($answer_body);
    $answer->setStudent($user);
    $answer->setTask($task);
    $answer->setFieldActivity($task->getFieldActivity());

    $task->incrementAnswers();

    $em->persist($answer);
    $em->flush();

    $view = $this->view(['answer' => 'Task correctly answered'],  statusCode: Response::HTTP_OK);
    return $this->handleView($view);
}
```

Illustration 69. API method for answering tasks.

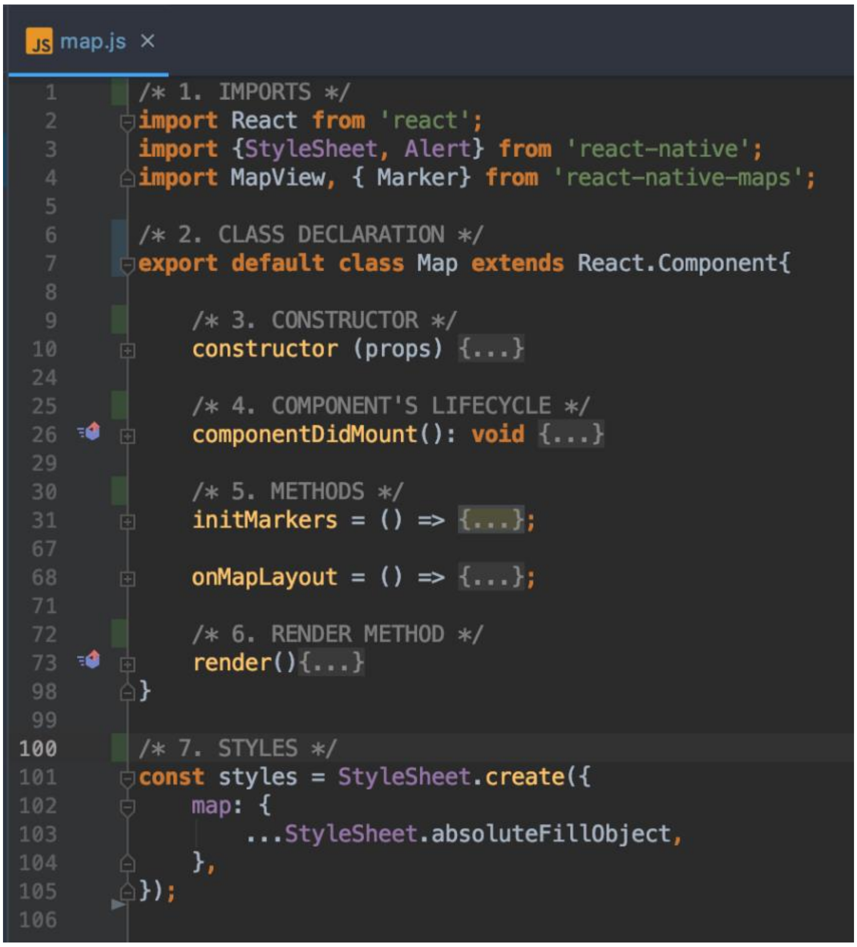
Then, in the method itself, we can see that using the entity manager, the current user and related tasks are taken from the database and the answer and its fields are created and persisted to the corresponding table in the database.

7.1.2. Mobile Application

7.1.2.1. Screens and components

As explained in chapter 6.1.2 (Packages diagram), the code for the mobile application is divided in seven packages, each of them containing different Javascript files with classes, screens, components or configuration and providers files.

The screens and components, which are the visual elements of the application, all have the same structure. We will see in detail this structure taking a look at the map component of the home page of the mobile application.



```
/* 1. IMPORTS */
import React from 'react';
import {StyleSheet, Alert} from 'react-native';
import MapView, { Marker} from 'react-native-maps';

/* 2. CLASS DECLARATION */
export default class Map extends React.Component{

    /* 3. CONSTRUCTOR */
    constructor (props) {...}

    /* 4. COMPONENT'S LIFECYCLE */
    componentDidMount(): void {...}

    /* 5. METHODS */
    initMarkers = () => {...};

    onMapLayout = () => {...};

    /* 6. RENDER METHOD */
    render(){...}
}

/* 7. STYLES */
const styles = StyleSheet.create({
    map: {
        ...StyleSheet.absoluteFillObject,
    },
});
```

Illustration 70. Basic class structure of React Native class.

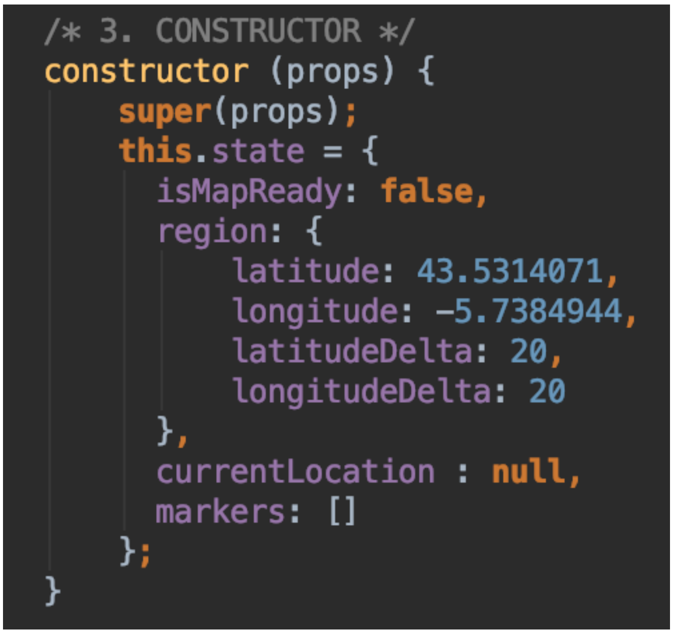
7.1.2.1.1. Imports

As in any other programming language, in this part of the file all the components and services used by the class are imported in order to use their functionality.

7.1.2.1.2. Class declaration

The name of the class declared here, as well as the classes that extends it or the classes which implements, as any other class oriented programming language.

7.1.2.1.3. Constructor



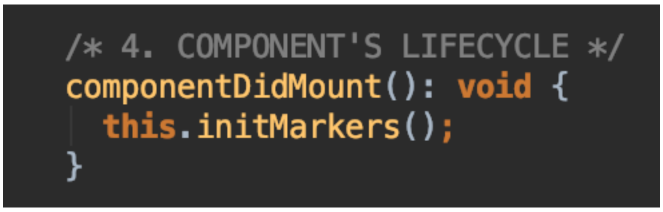
```
/* 3. CONSTRUCTOR */
constructor (props) {
    super(props);
    this.state = {
      isMapReady: false,
      region: {
          latitude: 43.5314071,
          longitude: -5.7384944,
          latitudeDelta: 20,
          longitudeDelta: 20
      },
      currentLocation : null,
      markers: []
    };
}
```

Illustration 71. Constructor of the map component.

In the constructor of the component, the main properties of the component are initialized and declared. In this case, in the state of the component is initialized the region where the map is located, the current location of the user and the list of markers of the map, as well as a boolean flag (isMapReady) to control de rendering of the map.

7.1.2.1.4. Lifecycle methods

These methods are triggered depending on the lifecycle of the component. There are different ones like componentWillMount, componentDidMount, componentWillUnmount, etc. Their name is very descriptive and they are in charge of calling the different methods of the component depending the state of this one, if the developer wants to use them (it is optional). In this case, the only one used is the componentDidMount, that is called when the component finish its creation execution in the app. In this specific case, this method can be omitted and call to the methods inside the contructor, but there are other examples where the use of this lifecycle methods is really powerful.

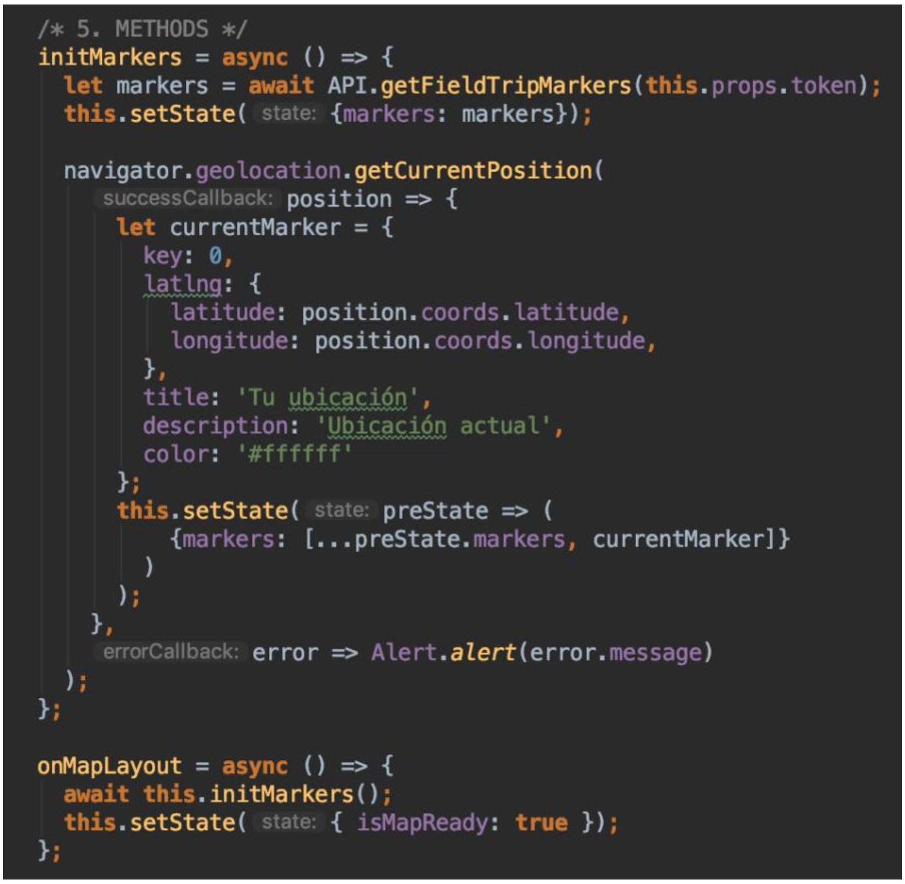


```
/* 4. COMPONENT'S LIFECYCLE */
componentDidMount(): void {
    this.initMarkers();
}
```

Illustration 72. Lifecycle methods of map component.

7.1.2.1.5. Methods

In this part of the component, the needed methods for the execution of it are declared. In this case, there are two methods: one in order to initialize the markers of the map and another one in order to change the boolean flag declared in the constructor.

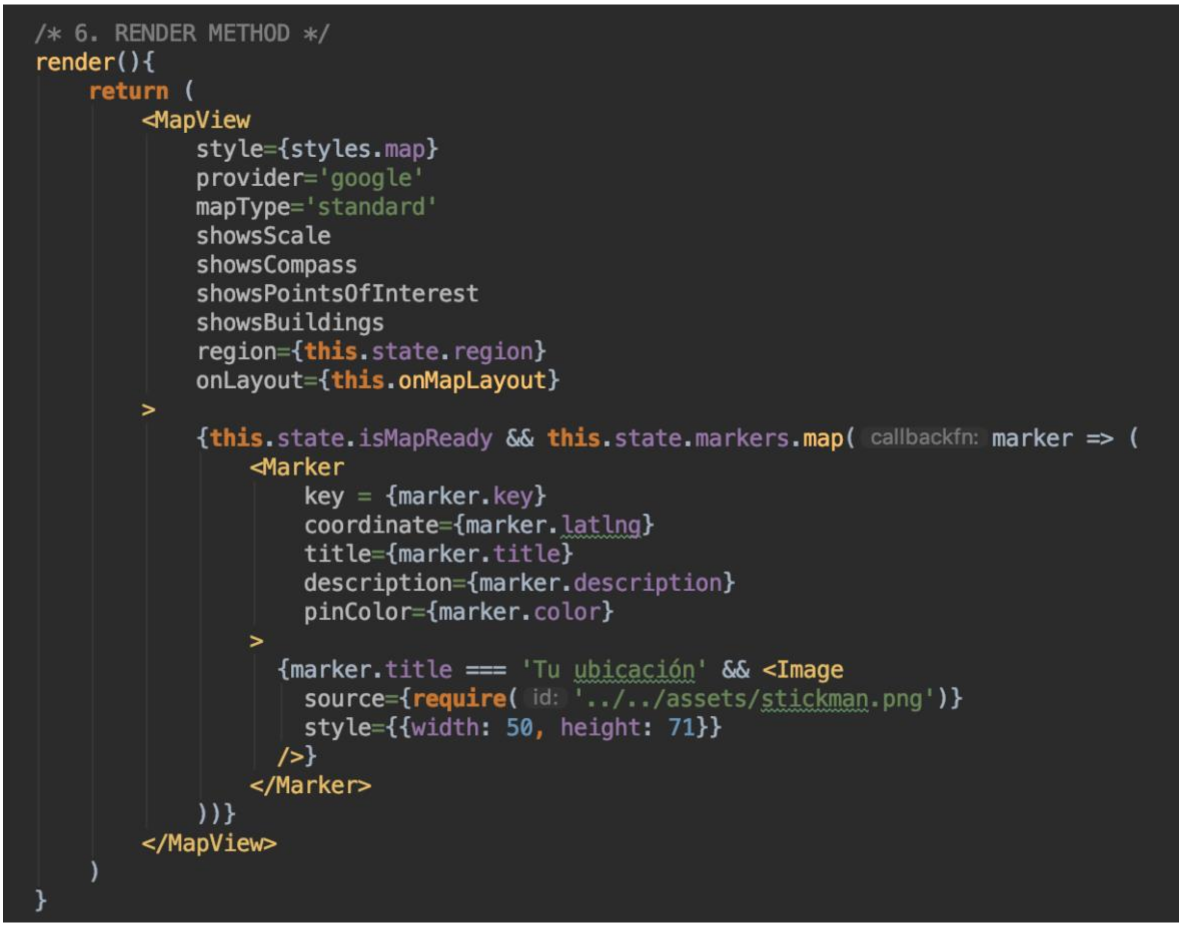


```
/* 5. METHODS */
initMarkers = async () => {
  let markers = await API.getFieldTripMarkers(this.props.token);
  this.setState( state: {markers: markers});

  navigator.geolocation.getCurrentPosition(
    successCallback: position => {
      let currentMarker = {
        key: 0,
        latlng: {
          latitude: position.coords.latitude,
          longitude: position.coords.longitude,
        },
        title: 'Tu ubicación',
        description: 'Ubicación actual',
        color: '#ffffff'
      };
      this.setState( state: preState => (
          {markers: [...preState.markers, currentMarker]}
        )
      );
    },
    errorCallback: error => Alert.alert(error.message)
  );
};

onMapLayout = async () => {
  await this.initMarkers();
  this.setState( state: { isMapReady: true });
};
```

Illustration 73. Methods of the map component.

7.1.2.1.6. Render method

This is the method that renders all the visual elements of the components and screens of the application. It uses HTML-like elements in order to render native elements as texts, views, buttons, spinners and many other components. All the elements used are imported in the first part of the file from the different libraries installed in the project.



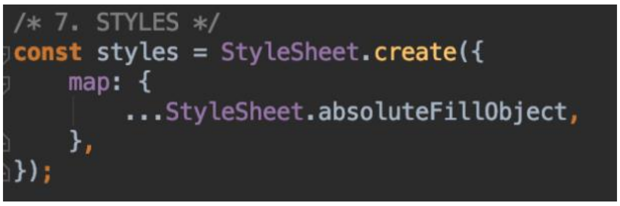
```
/* 6. RENDER METHOD */
render(){
    return (
        <MapView
            style={styles.map}
            provider='google'
            mapType='standard'
            showsScale
            showsCompass
            showsPointsOfInterest
            showsBuildings
            region={this.state.region}
            onLayout={this.onMapLayout}
        >
            {this.state.isMapReady && this.state.markers.map( callbackfn: marker => (
                <Marker
                    key = {marker.key}
                    coordinate={marker.latlng}
                    title={marker.title}
                    description={marker.description}
                    pinColor={marker.color}
                >
                  {marker.title === 'Tu ubicación' && <Image
                    source={require( id: '../../assets/stickman.png')}
                    style={{width: 50, height: 71}}
                  />}
                </Marker>
            ))}
        </MapView>
    )
}
```

Illustration 74. Render method of the map component.

In this case, we can see that this components uses a MapView, which has several properties like the style used or the region where it is centered. Inside the MapView, the markers of the different locations are rendered using the Marker element.

7.1.2.1.7. Styles

Finally, the styles part of the component. In this region of the file, the different stylesheets are declared in order to be used by the components that are rendered on screen. The properties, values and way that they are used is very similar to CSS (color, textAlign, font, background…), so it is a very simple to get used to its use and declaration. The only difference is that in React Native, each style is declared as a variable, which is later assign to the component.

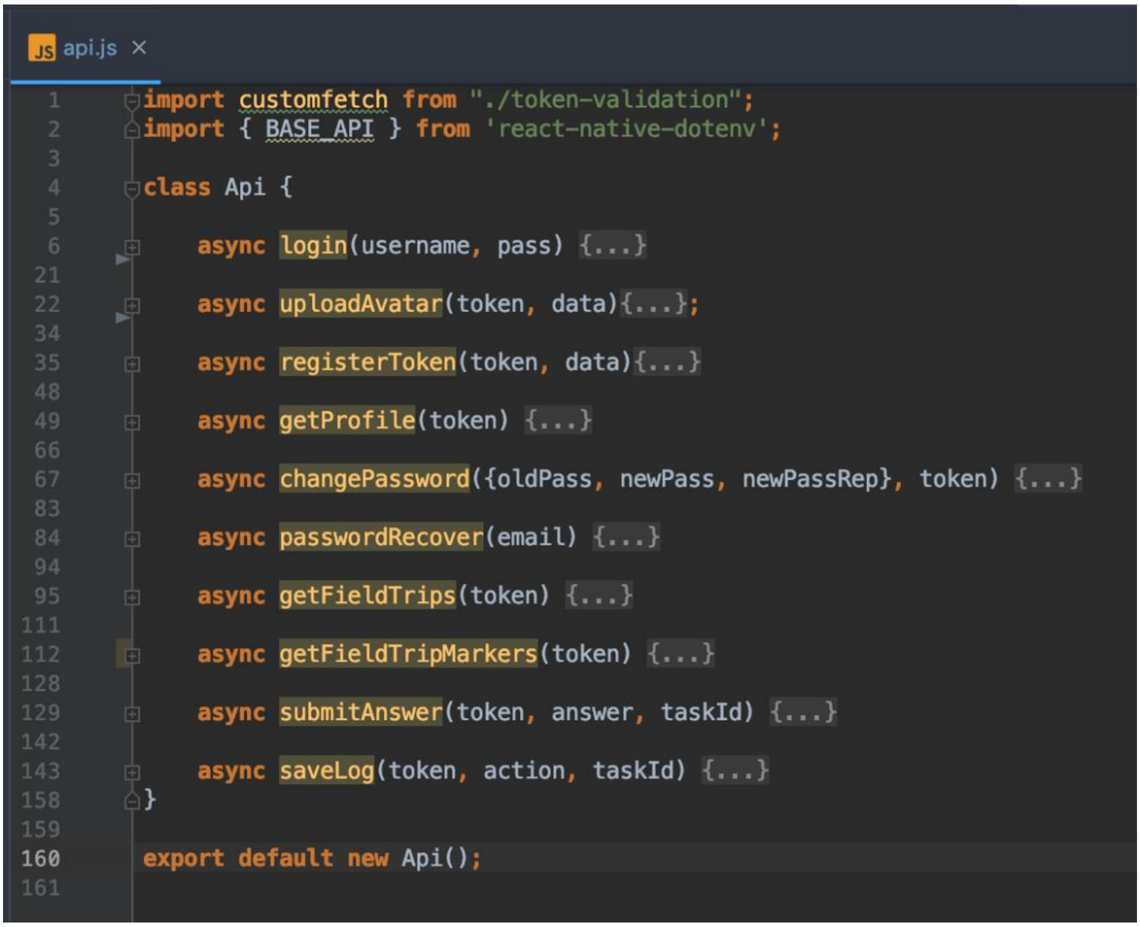


```
/* 7. STYLES */
const styles = StyleSheet.create({
    map: {
        ...StyleSheet.absoluteFillObject,
    },
});
```

Illustration 75. Styles of the map component.

7.1.2.2. Providers

The providers are classes containing methods in order to communicate with external systems. One example of provider would be the api.js file, where all the methods for accessing the API REST of the system are declared.



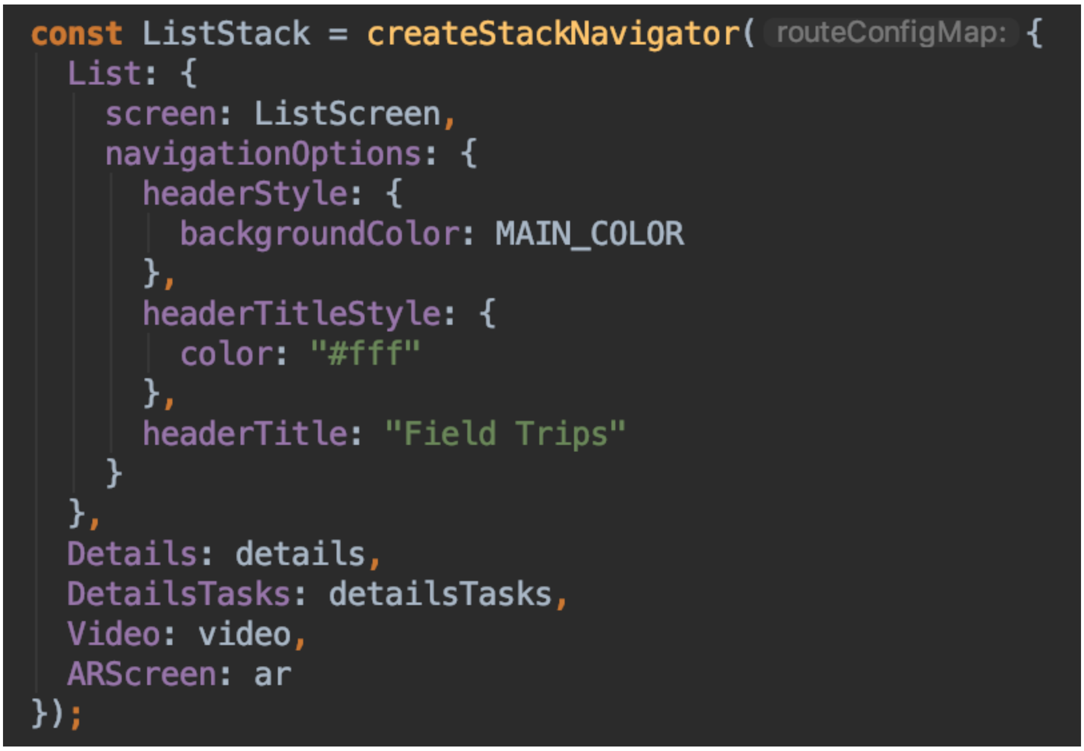
```
import customfetch from "./token-validation";
import { BASE_API } from 'react-native-dotenv';

class Api {

    async login(username, pass) {...}

    async uploadAvatar(token, data){...};

    async registerToken(token, data){...}

    async getProfile(token) {...}

    async changePassword({oldPass, newPass, newPassRep}, token) {...}

    async passwordRecover(email) {...}

    async getFieldTrips(token) {...}

    async getFieldTripMarkers(token) {...}

    async submitAnswer(token, answer, taskId) {...}

    async saveLog(token, action, taskId) {...}
}

export default new Api();
```

Illustration 76. API Provider.

This provider is imported in the screens or components where is needed in order to call to the methods declared inside it. Using this methodology, the same method can be called from multiple locations inside the project, but inly declaring it once.

7.1.2.3. Navigators

As explained in chapter 6, the navigators are the config files of the navigation between screens in the app. Inside the RootNavigator, all the flows of the application are separated in what it is call ‘stacks’. For example, in the List Stack, there will be included the screen of the list of field trips, the details of the trip and the tasks, the video screen and the AR Scene manager.



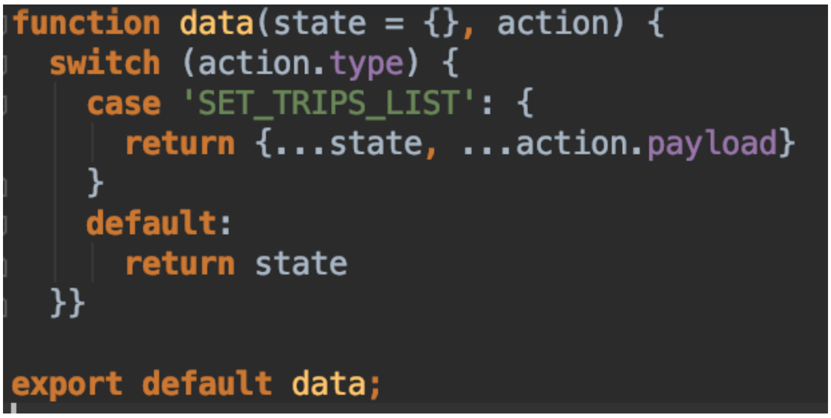
```
const ListStack = createStackNavigator( routeConfigMap: {
  List: {
    screen: ListScreen,
    navigationOptions: {
      headerStyle: {
        backgroundColor: MAIN_COLOR
      },
      headerTitleStyle: {
        color: "#fff"
      },
      headerTitle: "Field Trips"
    }
  },
  Details: details,
  DetailsTasks: detailsTasks,
  Video: video,
  ARScreen: ar
});
```

Illustration 77. ListStack from RootNavigator.

7.1.2.4. Reducers

The reducers are in charge of keeping a global state of the application. There can be several reducers defined in an application, each of them responsible of an action.

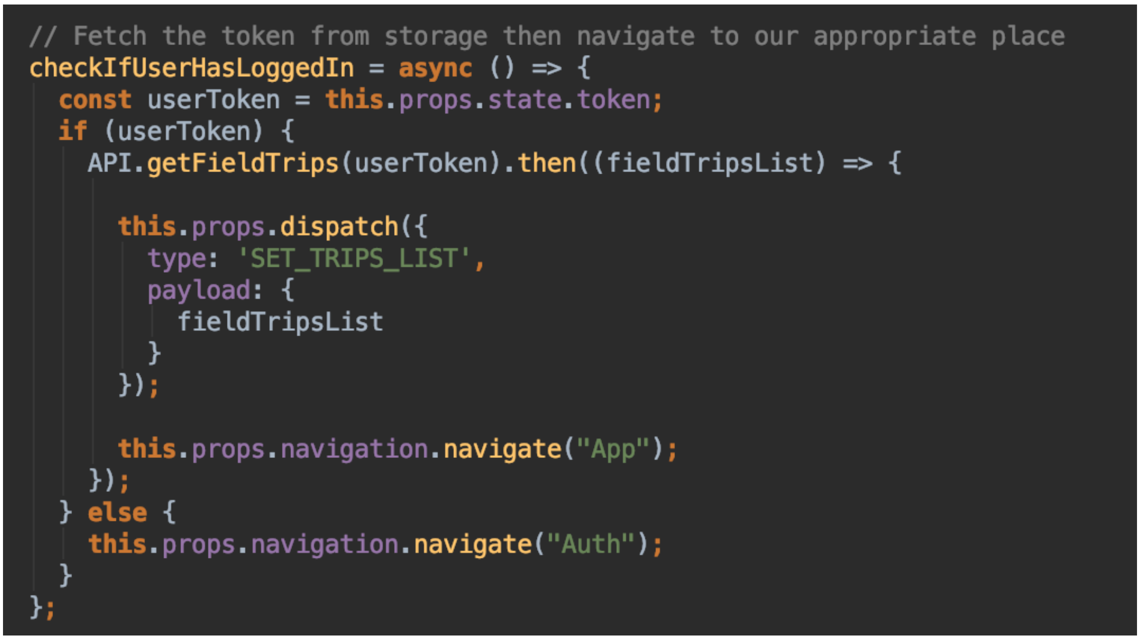
For example, at the start of the application, the reducer for getting the field trips (data.js) of the user is called, setting the list of field trips in order to retrieve it in the rest of the app.



```
function data(state = {}, action) {
  switch (action.type) {
    case 'SET_TRIPS_LIST': {
      return {...state, ...action.payload}
    }
    default:
      return state
  }}

export default data;
```

Illustration 78. Content of field trips reducer.

The call of the reducer is done by the dispatch method.



```
// Fetch the token from storage then navigate to our appropriate place
checkIfUserHasLoggedIn = async () => {
  const userToken = this.props.state.token;
  if (userToken) {
    API.getFieldTrips(userToken).then((fieldTripsList) => {

      this.props.dispatch({
        type: 'SET_TRIPS_LIST',
        payload: {
          fieldTripsList
        }
      });

      this.props.navigation.navigate("App");
    });
  } else {
    this.props.navigation.navigate("Auth");
  }
};
```

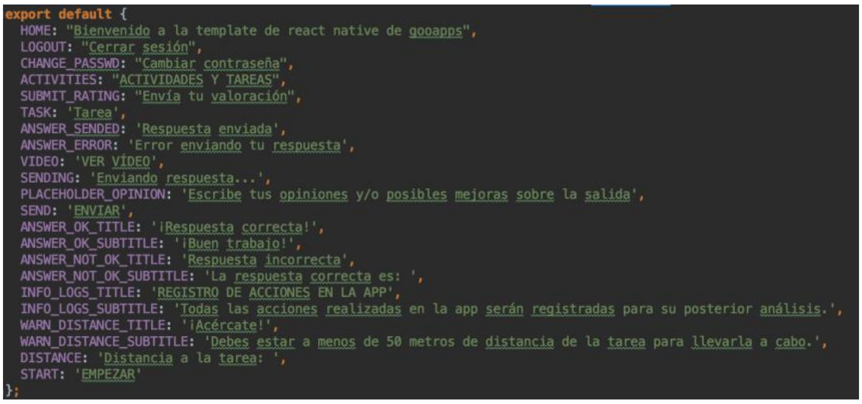
Illustration 79. Call to the data reducer.

In the above method, the app checks if the user is logged in. If it is logged, the app makes an API call to get the field trips related to that user. When the API retrieves the field trips, the app calls to the dispatch method and the trips are stored in the reducer.

7.1.2.5. Utils and config

The only util used in Ariane is the one that dials with the internationalization of the app, the translate util. It can be used in all the visual components of the app in order to display text in different languages depending on the locale of the device. The two available languages are Spanish and English.

Every text that is fixed in the app is stored in two config files, one for Spanish and one for English, each of the files containing a key-value structure. So, if the locale of the device in Spanish, the translate util will take the translations of the text from the Spanish config file, and the other way around.



```
export default {
  HOME: "Bienvenido a la template de react native de gooapps",
  LOGOUT: "Cerrar sesión",
  CHANGE_PASSWD: "Cambiar contraseña",
  ACTIVITIES: "ACTIVIDADES Y TAREAS",
  SUBMIT_RATING: "Envía tu valoración",
  TASK: 'Tarea',
  ANSWER_SENDED: 'Respuesta enviada',
  ANSWER_ERROR: 'Error enviando tu respuesta',
  VIDEO: 'VER VÍDEO',
  SENDING: 'Enviando respuesta...',
  PLACEHOLDER_OPINION: 'Escribe tus opiniones y/o posibles mejoras sobre la salida',
  SEND: 'ENVIAR',
  ANSWER_OK_TITLE: '¡Respuesta correcta!',
  ANSWER_OK_SUBTITLE: '¡Buen trabajo!',
  ANSWER_NOT_OK_TITLE: 'Respuesta incorrecta',
  ANSWER_NOT_OK_SUBTITLE: 'La respuesta correcta es: ',
  INFO_LOGS_TITLE: 'REGISTRO DE ACCIONES EN LA APP',
  INFO_LOGS_SUBTITLE: 'Todas las acciones realizadas en la app serán registradas para su posterior análisis.',
  WARN_DISTANCE_TITLE: '¡Acércate!',
  WARN_DISTANCE_SUBTITLE: 'Debes estar a menos de 50 metros de distancia de la tarea para llevarla a cabo.',
  DISTANCE: 'Distancia a la tarea: ',
  START: 'EMPEZAR'
};
```

Illustration 80. Spanish config file.

7.2. Augmented Reality in React Native

ViroReact is the name of the library used to implement Ariane's augmented reality features. ViroReact is a platform for developers to rapidly build native cross platform AR/VR applications using React Native. It supports both ARCore and ARKit, which are the Android and iOS tools for developing native augmented reality applications. This platform is based on two main components:

1. A high performance native 3D rendering engine
2. A custom extension of React for AR and VR development

In order to add augmented reality using ViroReact to a React Native application, there are two different ways. The first one is to use the own ViroReact in order to create the initial project, so all the configurations and libraries will be imported and automatically configured. This would have been really helpful for the development of Ariane, but I decided to use ViroReact as the AR library once the project was already created and functioning. The second way, and the most difficult of the two to carry out, was to add to the Android part of the already created project the necessary configuration in order to run Ariane with the augmented reality features working as expected.

So, for developing the AR part of the Ariane's mobile application, the following configurations had to be added to the base project:

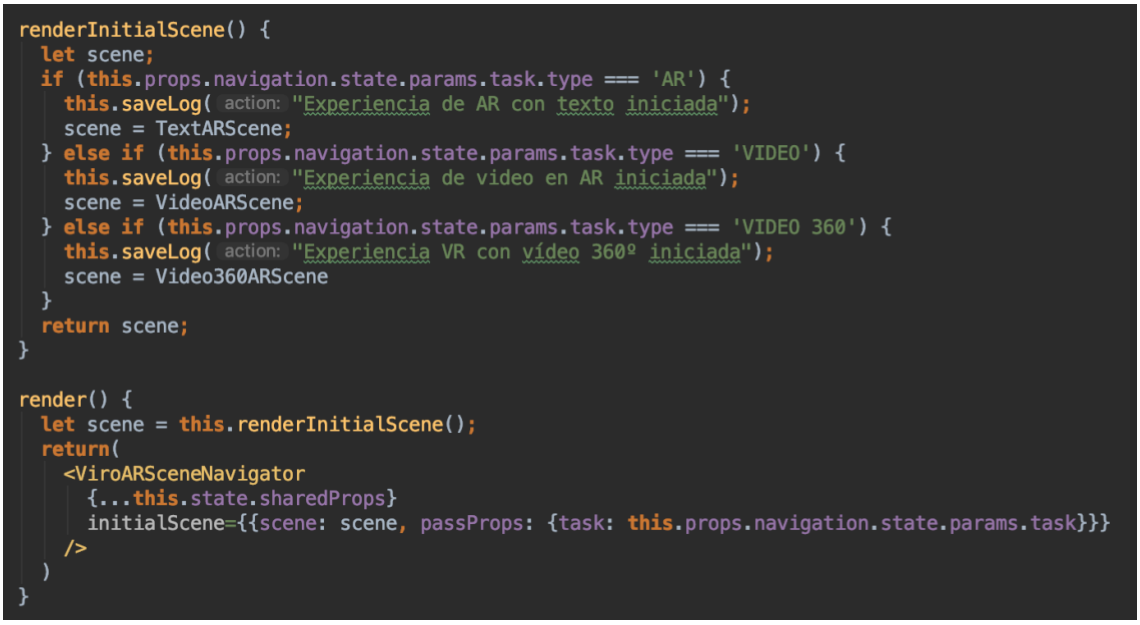
- Installation of the react-viro library via npm, the package manager of Ariane
- AndroidManifest.xml: This file provides the configuration of the app resulting from the android project, including permissions, main screen, app name and icon and API keys. In order to run ViroReact in my project, some tags had to be added, like the ARCore dependency and camera permissions
- Settings.gradle: In this file, external packages Android dependencies are declared in order to use them in the final app, like geolocation, geocoder, internationalization or Firebase integration. At the end of this file, the configuration of react-viro, viro-renderer and arcore had to be added
- MainApplication.java: This file is in charge of loading the different libraries and components declared in settings.gradle for their use in the final app. Only one line and its corresponding import had to be added to this file:

    -
      ```
      new ReactViroPackage(
              ReactViroPackage.
              ViroPlatform.valueOf(BuildConfig.VR_PLATFORM)
      ),
      ```
- Build.gradle (app): There are two build.gradle files per Android project. In the app one (located inside app folder), product-flavors had to be added, in order to compile the application making use of the ViroReact configuration. Moreover, different dependencies had to be added to the dependencies section of the file, like ARCore or react-viro ones.

These approaches were not discovered in the first try. ViroReact offers a section in its website about exactly this topic: how to integrate its augmented reality features in an already existing React Native project. However, this documentation is really confusing, as it does not specify exactly in which file or where to insert the new configurations. So, following a test and error methodology, I started looking in ViroReact forums, in Github issues similar to my needs and in any possible related website how to integrate the library in my project. Finally, I came across a ViroReact example project in Github, which helped me in order to see in which files and where to put the configurations, and how to run the mobile application in my device using the new productFlavors introduced in build.gradle.

These were all configurations in order to be able to use the augmented reality features of ViroReact in our project. Now, the implementation part will be explained. First of all, you need to select which of the functionalities and features of ViroReact you want to use. In my case, I implemented 3 types of augmented reality experiences: ARText, ARVideo and ARVideo360. The general use of this library in a React Native project is the following:

- First, a general 'controller' of the different augmented reality features has to be configured. In Ariane, this controller is the ARScreen, which contains a ViroARSceneNavigator component which renders the different AR scenes depending on the type of task to be performed.

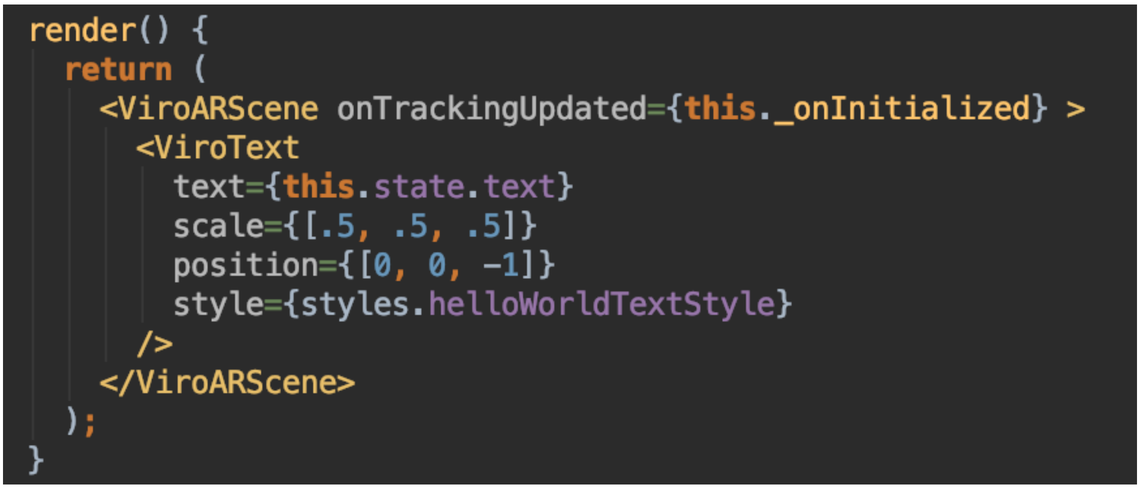


```
renderInitialScene() {
  let scene;
  if (this.props.navigation.state.params.task.type === 'AR') {
    this.saveLog( action: "Experiencia de AR con texto iniciada");
    scene = TextARScene;
  } else if (this.props.navigation.state.params.task.type === 'VIDEO') {
    this.saveLog( action: "Experiencia de video en AR iniciada");
    scene = VideoARScene;
  } else if (this.props.navigation.state.params.task.type === 'VIDEO 360') {
    this.saveLog( action: "Experiencia VR con vídeo 360º iniciada");
    scene = Video360ARScene
  }
  return scene;
}

render() {
  let scene = this.renderInitialScene();
  return(
    <ViroARSceneNavigator
      {...this.state.sharedProps}
      initialScene={{scene: scene, passProps: {task: this.props.navigation.state.params.task}}}
    />
  )
}
```

Illustration 81. ARScreen scene selection.

- Then, the necessary scenes depending of the functionality required have to be implemented as other components of the app. They will have the same structure as other components, explained previously in this chapter. In the render method, the ViroReact component required would be rendered inside a ViroARScene component, which is the general component where all the ViroReact features have to be included.
  - TextSceneAR: This AR scene will render the description of the task in the real world, providing an augmented reality experience to the user. This

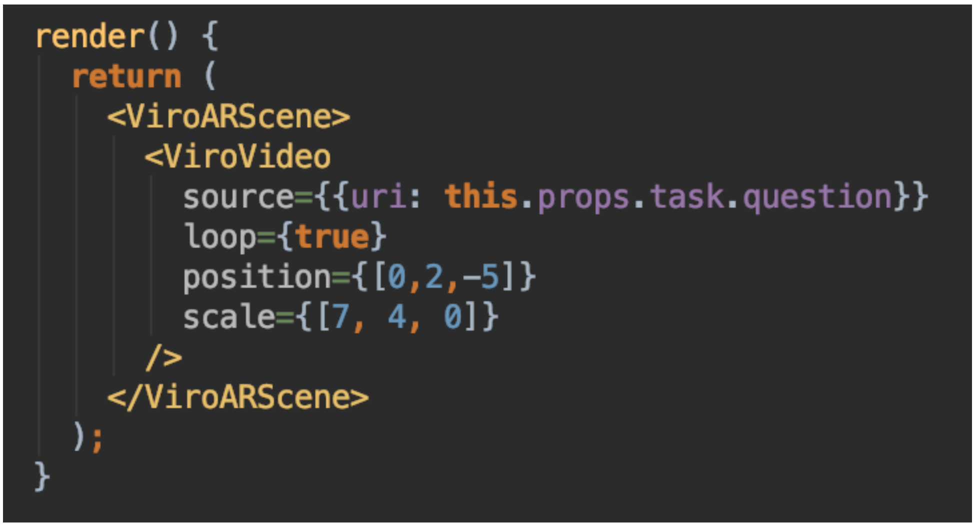
will be made using the ViroText component, which receives three properties: the text to be rendered, the scale of the text and the position relative to the camera of the device.



```
render() {
  return (
    <ViroARScene onTrackingUpdated={this._onInitialized} >
      <ViroText
        text={this.state.text}
        scale={[.5, .5, .5]}
        position={[0, 0, -1]}
        style={styles.helloWorldTextStyle}
      />
    </ViroARScene>
  );
}
```

Illustration 82. ViroText usage.

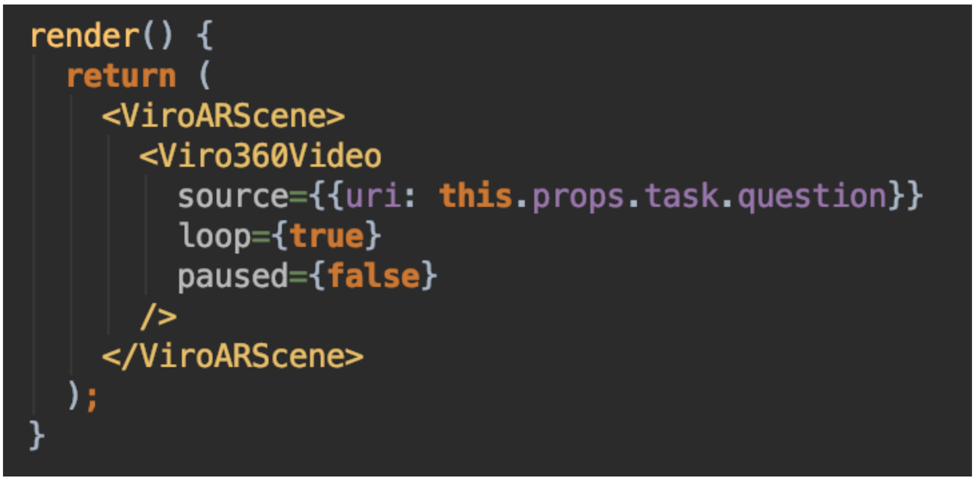
- VideoSceneAR: The video provided in the property source will be displayed in a plane using augmented reality. This is fulfilled using the ViroVideo component of ViroReact. This component also receives the scale and position properties, as well as the source of the video, in this case, coming from the API as the task's question that is going to be performed, and the loop flag, in order to replay the video when it has finished.



```
render() {
  return (
    <ViroARScene>
      <ViroVideo
        source={{uri: this.props.task.question}}
        loop={true}
        position={[0,2,-5]}
        scale={[7, 4, 0]}
      />
    </ViroARScene>
  );
}
```

Illustration 83. ViroVideo usage.

- Video360SceneAR: A 360 video consists on a video that changes depending on the orientation of the device which you are using for watching the video. Users can choose to turn their device in order to see different perspectives of the video. This is the basic functionality of virtual reality glasses, which immerse you in a virtual world where you can explore all of your surroundings. The experience is developed in Ariane using the Viro360Video component, which receives a 360 degrees video URL and displays the video in the device screen, where the user is able to turn his mobile in order to watch the different angles of the video



```
render() {
  return (
    <ViroARScene>
      <Viro360Video
        source={{uri: this.props.task.question}}
        loop={true}
        paused={false}
      />
    </ViroARScene>
  );
}
```

Illustration 84. Viro360Video usage.

ViroReact offers many more augmented reality features, like 3D object rendering, animated images, buttons, tracking objects or image targets and image recognition. Most of these features are static and could not be implemented in the creation and realization of the tasks inside a field trip because of the differences between some of the types of task available in Ariane. For example, it would have been really interesting to add to the system a 360 image of the field trip with some points of interest or some 3D objects rendering. But, taking all the problems while integrating ViroReact in my project, the available features of the library and the flexibility of the field trips and tasks, the augmented reality features added to Ariane give an enjoyable experience to the user.

7.3. System creation

7.3.1. Dynamic task detail depending on type

7.3.1.1. Problem introduction

One of the biggest problems I faced when developing the mobile application was that depending on the type of the task that was being rendered, the contents of the screen had to change. One approach would have been creating a detail screen for each of the types and then show one or the other depending on the type of the task. This approach would have been a mess, because for each screen, it has to be added to the navigator of the app, making the navigation between screens mush less ineffective.

7.3.1.2. Problem resolution

In order to resolve this problem, I have only created one DetailTask screen, and taking advantages of the render methods of React Native, I have been able to render different components depending on the type of the task using a simple switch-case implementation.

In the global render method of the screen, the shared components and elements are rendered, but, in the zones where the screen depends on the task, another render method is called, which has a switch case clause inside with the components to be rendered for each type of task.

Nevertheless, this is not the optimal solution as it still has the switch-case clause to change the visualization of the screen. The optimal solution would be to create a component for each task and a generic component, that internally selects the Javascript file that has to be used, but this approach might consume a lot of time and costs that the project might not be capable of dealing with.

7.3.2. Navigations and reducers behaviour

7.3.2.1. Problem introduction

Due to my inexperience developing apps with React Native, the functionality of the navigators and reducers was completely new for me. Moreover, it did not look like nothing I had worked in my short programming 'career'.

7.3.2.2. Problem resolution

I started a course about the basics of working with React Native using navigation, reducers and the basics and most common features of the framework in Platzi, a video-based tutorial platform which offers a great variety of courses. With the help of this course and the support and help from my colleagues in GooApps, I quickly learned the basics of working with React Native and started developing a more complex system for Ariane.

7.3.3. Google Maps in Symfony form for the tasks

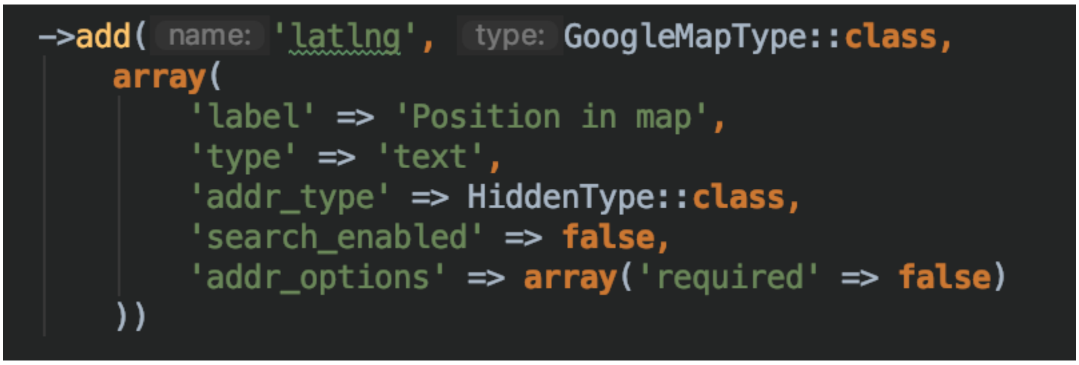
7.3.3.1. Problem introduction

For the creation of the tasks, a location was needed. It needed to have two coordinates: latitude and longitude. But putting two input fields in the form for manually typing the latitude and longitude of the task would be very tedious and maybe difficult to understand for the final user. So I needed to incorporate to the Symfony project another type for the form field, something like a map in order to place a marker where the task was supposed to be located.

7.3.3.2. Problem resolution

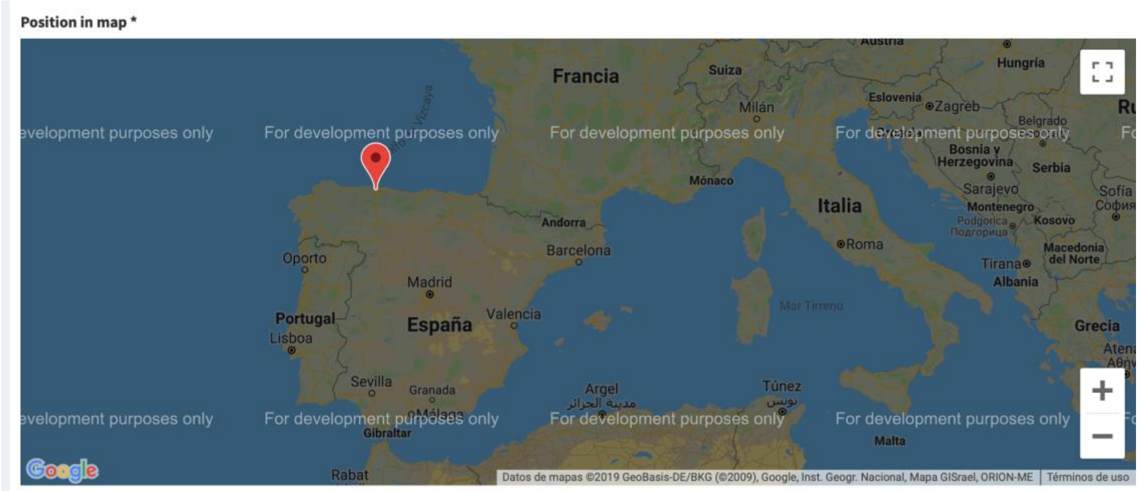
After some researches, I ended up finding a library that suited my problem perfectly. It was a custom GoogleMaps type for Symfony forms, which displays a map where the user can place a marker, and the coordinates of that point were persisted to the entity that was being created.

After making the needed configurations of the project (installing, importing, config files), I finally achieve to introduce this new type of field in the TaskAdmin. But there were some functionalities of it, like the geolocation to go to your position, the searchbar and the geocoder that were not working. I discovered that, because the GoogleMapType was using an API key from Google, that api key had to be activated in order to these functionalities to work. Nowadays, if you want to fully activate an api key from Google, you have to input your credit card data. It is very unlikely, taking into account the traffic and the number of calls to the API, that they would charge something to my credit card, but I did not want to risk it. So, unfortunately, those functionalities are not supported in Ariane, but the map works fine and the teacher is going to be able to locate the tasks in the map without any problem.



```
->add( name: 'latlng',  type: GoogleMapType::class,
    array(
        'label' => 'Position in map',
        'type' => 'text',
        'addr_type' => HiddenType::class,
        'search_enabled' => false,
        'addr_options' => array('required' => false)
    ))
```

Illustration 85. Use of the GoogleMapType in TaskAdmin.



Position in map *

Illustration 86. Visualization of the map without activating the API key.

7.3.4. ViroReact integration

7.3.4.1. Problem introduction

As explained in chapter 7.2, dedicated exclusively to this topic, the documentation about integrating ViroReact with an existing React Native project was very confusing. It says what the developer has to add in order to add augmented reality features to the project, but this information is incomplete and not really clear.

7.3.4.2. Problem resolution

After a long time searching for information about this issue over the Internet, taking a look to every Github issue of the library and looking at different sample projects, I finally came across the solution explained in chapter 7.2 in order to integrate ViroReact in my React Native project.

8. Tests

In this chapter, the different type of tests being carried in Ariane will be explained, taking into account the procedures used, the expected result and the actual result of each of them.

8.1. System testing

One the application has been finished, an exhaustive testing of all the functionalities and views of Ariane has been made in order to test whether the system works as expected or not.

The different tests that have been carried are the following.

8.1.1. Web Application tests

ID	Description	Expected result	Actual result	Passed
T_WA_1	The teacher navigates to ariane-uniovi.es	The login page will be shown	The login page is shown	YES
T_WA_2	The teacher tries to log in to the web application with incorrect credentials	An error ('Invalid credentials') will be shown on screen	An error ('Invalid credentials') is shown on screen	YES
T_WA_3	A teacher with a disabled account tries to log in to the system	An error ('Account is disabled') will be shown on screen	An error ('Account disabled') is shown on screen	YES
T_WA_4	The teacher clicks on 'Forgotten password?' label	A new page with an email input is rendered on screen	A new page with an email input is rendered on screen	YES
T_WA_5	The teacher introduces an email in order to change the password	An informative text is displayed and an email is received with the instructions to change the password	An informative text is displayed and an email is received with the instructions to change the password	YES
T_WA_6	The teacher tries to log in to the web application with correct credentials	The user will be redirected to the home page	The home page is rendered	YES

T_WA_7	The teacher goes to the field trip page in order to create a new one clicking on the 'Field trips' label of the side menu	The list of the existing field trips, if any, is displayed on screen	The list of the existing field trips, if any, is displayed on screen	YES
T_WA_8	The teacher clicks on 'Add new' and introduces the requested fields	The field trip is created	The field trip is created	YES
T_WA_9	The teacher wants to see the detail of a field trip and clicks on the 'Show' button	The detail of the field trip is rendered on screen, with all its fields in a list	The detail of the field trip is rendered on screen, with all its fields in a list	YES
T_WA_10	The teacher introduces a filter by teacher assigned in the filter list	A filtered list of all the field trips with that associated teacher is shown on screen	A filtered list of all the field trips with that associated teacher is shown on screen	YES
T_WA_11	The teacher tries to delete a field trip with associated tasks, answers or logs	The system will not allow the teacher to delete this field trip	The system does not allow the teacher to delete this field trip	YES
T_WA_12	The teacher tries to delete a field trip without associated tasks, answers nor logs	The field trip is deleted	The field trip is deleted	YES
T_WA_13	The teacher edits a field of the field trip, first clicking on the edit button and then changing its name	The field trip field is changed	The field trip field is changed	YES
T_WA_14	The teacher downloads the list of field trips	An excel file containing the field trips is downloaded to the teacher's computer	An excel file containing the field trips is downloaded to the teacher's computer	YES
T_WA_15	The teacher goes to the field trip page in order to create a new one clicking on the 'Tasks' label of the side menu	The list of the existing tasks, if any, is displayed on screen	The list of the existing tasks, if any, is displayed on screen	YES
T_WA_16	The teacher clicks on 'Add new' and introduces the requested fields	The task is created	The task is created	YES

T_WA_17	The teacher wants to see the detail of a task and clicks on the 'Show' button	The detail of the task is rendered on screen, with all its fields in a list	The detail of the task is rendered on screen, with all its fields in a list	YES
T_WA_18	The teacher introduces a filter by field trip in the filter list	A filtered list of all the task contained in the specified field trip is shown on screen	A filtered list of all the task contained in the specified field trip is shown on screen	YES
T_WA_19	The teacher tries to delete a task with associated answers or logs	The system will not allow the teacher to delete this task	The system will not allow the teacher to delete this task	YES
T_WA_20	The teacher tries to delete a field trip without associated answers nor logs	The task is deleted	The task is deleted	YES
T_WA_21	The teacher edits a field of the task, first clicking on the edit button and then changing its name	The task's field is changed	The task's field is changed	YES
T_WA_22	The teacher downloads the task list	An excel file containing the tasks is downloaded to the teacher's computer	An excel file containing the tasks is downloaded to the teacher's computer	YES
T_WA_23	The teacher wants to see the different answers to the tasks, and clicks on the 'Answers' label in the side menu	The answer list page is shown on screen	The answer list page is shown on screen	YES
T_WA_24	The teacher wants to see the detail of an answer and clicks on the 'Show' button	The detail of the answer is rendered on screen, with all its fields in a list	The detail of the answer is rendered on screen, with all its fields in a list	YES
T_WA_25	The teacher introduces a filter by student in the filter list	A filtered list of all the answers made by that student is shown on screen	A filtered list of all the answers made by that student is shown on screen	YES
T_WA_26	The teacher tries to delete an answer	The answer is deleted	The answer is deleted	YES

T_WA_27	The teacher downloads the answer list	An excel file containing the answers is downloaded to the teacher's computer	An excel file containing the answers is downloaded to the teacher's computer	YES
T_WA_28	The teacher wants to see the different logs generated by the mobile app, and clicks on the 'Logs' label in the side menu	The logs list page is shown on screen	The logs list page is shown on screen	YES
T_WA_29	The teacher wants to see the detail of a log and clicks on the 'Show' button	The detail of the log is rendered on screen, with all its fields in a list	The detail of the log is rendered on screen, with all its fields in a list	YES
T_WA_30	The teacher introduces a filter by student in the filter list	A filtered list of all the logs generated by that student is shown on screen	A filtered list of all the logs generated by that student is shown on screen	YES
T_WA_31	The teacher tries to delete a log	The log is deleted	The log is deleted	YES
T_WA_32	The teacher downloads the log list	An excel file containing the logs is downloaded to the teacher's computer	An excel file containing the logs is downloaded to the teacher's computer	YES
T_WA_33	The admin wants to create a user and clicks on the 'Users' label in the side menu	The user list page is shown on screen	The user list page is shown on screen	YES
T_WA_34	The admin clicks in 'Add new' and introduces the requested fields	The user is created	The user is created	YES
T_WA_35	The admin want to edit some field of the user, cliks on its username and introduces the new data	The user will be edited with the new fields	The user is edited with the new fields	YES
T_WA_36	The admin tries to delete a user that belongs to a group, or that has any answer or log related	The system does not allow the admin to delete this user and an error	The system does not allow the admin to delete this user and an error	YES

		message is displayed	message is displayed	
T_WA_37	The admin tries to delete a user that does not belong to a group, nor that has any answer or log related	The user is deleted	The user is deleted	YES
T_WA_38	The admin filters the user list depending on the students group they belong to	A filtered list with only the users of that group is displayed	A filtered list with only the users of that group is displayed	YES
T_WA_39	The admin donwloads the user list	An excel file containing the users is downloaded to the admin's computer	An excel file containing the users is downloaded to the admin's computer	YES
T_WA_40	The admin wants to create a new students group and clicks on the 'students group' label on the side menu	The students group list page is shown on scree	The students group list page is shown on scree	YES
T_WA_41	The admin clicks in 'Add new' and introduces the requested fields	The new students group is created	The new students group is created	YES
T_WA_42	The admin want to edit some field of the students group, cliks on 'Edit' and introduces the new data	The students group will be edited with the new data	The students group is edited with the new data	YES
T_WA_43	The admin tries to delete a students group with associated students or field trips	The system does not allow the admin to delete this students group and an error message is displayed	The system does not allow the admin to delete this students group and an error message is displayed	YES
T_WA_44	The admin filters the students group list depending on its name	A filtered list with only the group with that name is displayed	A filtered list with only the group with that name is displayed	YES

T_WA_45	The admin donwloads the students group list	An excel file containing the students groups is downloaded to the admin's computer	An excel file containing the students groups is downloaded to the admin's computer	YES
T_WA_46	The admin wants to see the detail of a students group and clicks on 'Show'	The detail of the students group is rendered on screen, with all its fields in a list	The detail of the students group is rendered on screen, with all its fields in a list	YES
T_WA_47	The admin or teacher clicks on the logout icon in the right hand corner	The user logs out and the system navigates to the login page	The user logs out and the system navigates to the login page	YES

8.1.2. Mobile app tests

ID	Description	Expected result	Actual result	Passed
T_MA_1	The student tries to log in to the app with invalid credentials	A message is shown to the student and the app stays in the login screen	A message is shown to the student and the app stays in the login screen	YES
T_MA_2	The student clicks on 'Recuperar contraseña' in order to change his password	The app navigates to the recovery password screen	The app navigates to the recovery password screen	YES
T_MA_3	The students introduces his email and clicks on 'Enviar'	The app will show an info message and send an email with the intructions to change the password	The app will show an info message and send an email with the intructions to change the password	YES
T_MA_4	The student logs in with valid credentials	The app navigates to the home screen with the map and the tasks for the user, if any	The app navigates to the home screen with the map and the tasks for the user, if any	YES
T_MA_5	The student clicks on a marker displayed on the map	The title of the task and the title of the field trip which it is associated to are displayed as an info box on the marker	The title of the task and the title of the field trip which it is associated to are displayed as an info box on the marker	YES
T_MA_6	The students clicks on the profile tab in the navigation bar	The app navigates to the profile screen	The app navigates to the profile screen	YES
T_MA_7	The student clicks on the 'Cambiar contraseña' label	The app navigates to the change password screen	The app navigates to the change password screen	YES

T_MA_8	The student introduces his old password and the new one, but the repetition of the new is not correct	An error message is shown to the user and the password is not changed	An error message is shown to the user and the password is not changed	YES
T_MA_9	The student introduces his old password, the new one and the repetition of the new one correctly	The student's password is changed	The student's password is changed	YES
T_MA_10	The student clicks on the back button	The app navigates back to the profile screen	The app navigates back to the profile screen	YES
T_MA_11	The student clicks on the central tab 'Salidas' of the tab navigator	The app navigates to the page containing the field trip list	The app navigates to the page containing the field trip list	YES
T_MA_12	The student clicks on a field trip item of the list	The app navigates to the field trip detail screen	The app navigates to the field trip detail screen	YES
T_MA_13	The student clicks on a marker of the map view	The title of the task is displayed in an info box	The title of the task is displayed in an info box	YES
T_MA_14	The student clicks on a task item of the task list	The app navigates to the task detail page	The app navigates to the task detail page	YES
T_MA_15	The type of the clicked task is YOUTUBE	The app shows the photo of the task and a button to go the video visualization	The app shows the photo of the task and a button to go the video visualization	YES
T_MA_16	The student clicks the button in order to see the video	The Youtube video is played	The Youtube video is played	YES
T_MA_17	The type of the task is DESCRIPTION	The app shows the photo of the task and a general description of the field trip	The app shows the photo of the task and a general description of the field trip	YES
T_MA_18	The type of the task is AUDIO	The app shows the photo of the task and an audio player for the reproduction of the audio	The app shows the photo of the task and an audio player for the reproduction of the audio	YES

T_MA_19	The students click on the play button	The audio starts and the icon changes to the pause one	The audio starts and the icon changes to the pause one	YES
T_MA_20	The student clicks on the pause button	The audio pauses and the icon changes on to the play one	The audio pauses and the icon changes on to the play one	YES
T_MA_21	The student uses the seek bar in order to advance the audio	The audio pauses and advances to the point where the student stops the slider	The audio pauses and advances to the point where the student stops the slider	YES
T_MA_22	The type of the task is TEST	The app shows the photo of the task, an options selection and a submit button	The app shows the photo of the task, an options selection and a submit button	YES
T_MA_23	The student select an uncorrect option and clicks on the submit button	The app stores the answer of the student in the database and shows an error message to the student with the correct answer	The app stores the answer of the student in the database and shows an error message to the student with the correct answer	YES
T_MA_24	The student select the correct option and clicks on the submit button	The app stores the answer of the student in the database and shows a success message to the student	The app stores the answer of the student in the database and shows a success message to the student	YES
T_MA_25	The type of the task is RATING	The app shows the photo of the task, a rating bar, a text box and a submit button	The app shows the photo of the task, a rating bar, a suggestions text box and a submit button	YES
T_MA_26	The student rates the field trip, writes his opinion and clicks the submit button	The app stores the rating and the opinion of the user in the database and shows a success	The app stores the rating and the opinion of the user in the database and shows a success message to the student	YES

		message to the student		
T_MA_27	The type of the task is AR	The app shows the photo of the task and a button to access the AR experience	The app shows the photo of the task and a button to access the AR experience	YES
T_MA_28	The student clicks on the button to perform the AR task, but his device does not support ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	YES
T_MA_29	The student clicks on the button to perform the AR task, and his device supports ARCore	The app opens the camera and displays the description of the task in augmented reality	The app opens the camera and displays the description of the task in augmented reality	YES
T_MA_30	The type of the task is VIDEO	The app shows the photo of the task and a button to play the video in augmented reality	The app shows the photo of the task and a button to play the video in augmented reality	YES
T_MA_31	The student clicks on the button to see the video, but his device does not support ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	YES
T_MA_32	The student clicks on the button to see the video, and his device supports ARCore	The app opens the camera and displays the video in augmented reality	The app opens the camera and displays the video in augmented reality	YES
T_MA_33	The type of the task is VIDEO 360	The app shows the photo of the task and a button to play the 360	The app shows the photo of the task and a button to play	YES

		video in virtual reality	the 360 video in virtual reality	
T_MA_34	The student clicks on the button to see the video, but his device does not support ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	The app opens the camera, but displays a message to the user saying that this task needs ARCore	YES
T_MA_35	The student clicks on the button to see the video, and his device supports ARCore	The app opens the camera and displays the 360 video in virtual reality	The app opens the camera and displays the 360 video in virtual reality	YES
T_MA_36	The student goes to the profile page and clicks on 'Cerrar sesión'	The student logs out from the app and it navigates to the login page	The student logs out from the app and it navigates to the login page	YES

8.1.3. API tests

ID	Description	Expected result	Actual result	Passed
T_API_1	Navigate to ariane-uniovi.es/api/doc	Get the API documentation	Get the API documentation	YES
T_API_2	Test the login method, introducing the username and password of an enabled user	Get a 200 response with the token of the user	Get a 200 response with the token of the user	YES
T_API_3	Register the token of the user in the 'Authorize' part of the documentation, typing 'Bearer ' and the token got in the login method	Authorization enabled	Authorization enabled	YES
T_API_4	Test the profile API method	Get the profile of the authenticated user	Get the profile of the authenticated user	YES
T_API_5	Test the field-activity/all method	Get all the field trips	Get all the field trips	YES
T_API_6	Test the field-activity/student/activities method	Get the field trips associtated to the authenticated user	Get the field trips associtated to the authenticated user	YES
T_API_7	Test the answer/{taskId} POST method in order to answer a task, introducing the id of the task and the answer in the parameters	Get a 200 response saying that the task has been answered	Get a 200 response saying that the task has been answered	YES
T_API_8	Test the log/{taskId} POST method in order to create a log for the task, introducing the id of the task, the action and a location in the parameters	Get a 200 response saying that the log has been created	Get a 200 response saying that the log has been created	YES

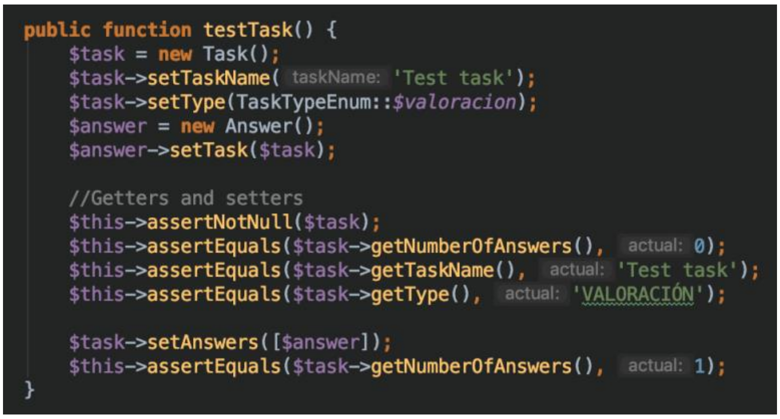
8.2. Unit testing

Unit testing is a level of software testing where individual components of the software are tested. Its purpose is to validate that each of this units of code performs as expected. In Ariane, some unit testing has been made in both the web and mobile application. In the web application, the behaviour of the different entities has been tested, while in the mobile application, the rendering of the components has been tested in order to see if they render correctly without any error or warning.

8.2.1. Web Application

In Symfony, unit testing is developed using PHPUnit. Its syntaz is very similar to the worldwide known JUnit from Java, so it was really simple to write some tests in order to prove if the different entities of the system were working correctly or not.

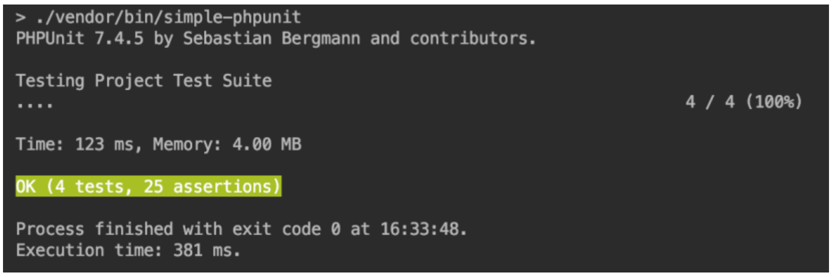
The different tests are stored in the 'test' package of the Symfony project. Inside it, in the AppBundle/Controller subpackage, is the EntityControllerTest, the PHP class containing the different entity tests designed for Ariane. In it, the field activity, task, answer, log and student group are tested, creating some mock entities and fields. To run these tests, the command vendor/bin/simple-phpunit has to be run. This will print in the console the output of the different test written in the Controller.



```
public function testTask() {
    $task = new Task();
    $task->setTaskName( taskName: 'Test task');
    $task->setType(TaskTypeEnum::$valoracion);
    $answer = new Answer();
    $answer->setTask($task);

    //Getters and setters
    $this->assertNotNull($task);
    $this->assertEquals($task->getNumberOfAnswers(),  actual: 0);
    $this->assertEquals($task->getTaskName(),  actual: 'Test task');
    $this->assertEquals($task->getType(),  actual: 'VALORACIÓN');

    $task->setAnswers([$answer]);
    $this->assertEquals($task->getNumberOfAnswers(),  actual: 1);
}
```

Illustration 87. Example of unit test in Symfony.

The output of the developed tests is the following:



```
> ./vendor/bin/simple-phpunit
PHPUnit 7.4.5 by Sebastian Bergmann and contributors.

Testing Project Test Suite
....                                                              4 / 4 (100%)

Time: 123 ms, Memory: 4.00 MB

OK (4 tests, 25 assertions)

Process finished with exit code 0 at 16:33:48.
Execution time: 381 ms.
```

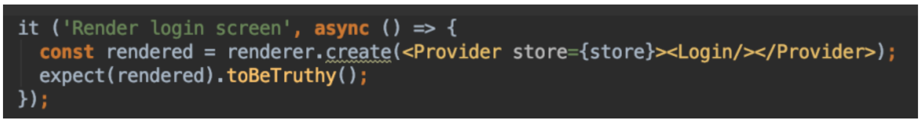
Illustration 88. Symfony unit tests output.

8.2.2. Mobile application

In order to perform unit tests in React Native, the test framework Jest will be used. Jest is a Javascript test framework developed by Facebook, which started in the context of React, but that can be used in order to test any Javascript code (Babel, Typescript, Node, React, Angular, Vue and more). It provides a large amount of methods and functionalities in order to test the behaviour of our app, like assertions to compare functions outputs and class fields, functions mocks in order to give functionality to the tested component and really understandable and friendly exceptions when tests failed. It also offers a great functionality as code coverage, which is the percentage of code of our application that tests are covering.

In Ariane, Jest would be used in order to test the correct rendering of the different screens and components used in its development, such as buttons, text inputs, icons and the different app screens.

The tests are developed in *.test.js files and are run with the command 'jest'. In our case, all the tests being carried are in the App.test.js file. For example, the Login page will be tested like this:

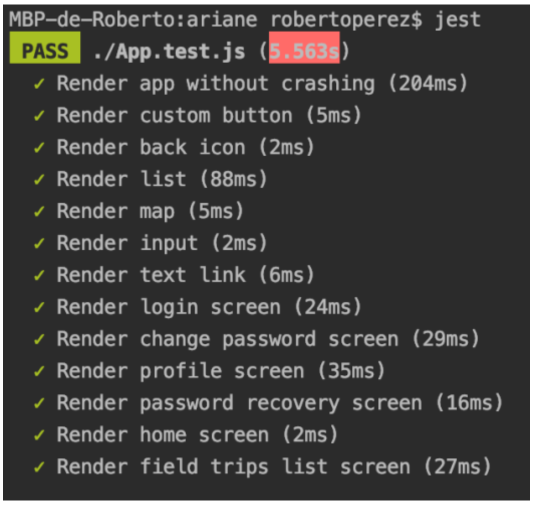


```
it ('Render login screen', async () => {
  const rendered = renderer.create(<Provider store={store}><Login/></Provider>);
  expect(rendered).toBeTruthy();
});
```

Illustration 89. Login page Jest test example.

The test renders a mock Login page and expected to be truthy, that is, that the page has been rendered correctly. The Login page have to be wrapped inside a Provider in order to pass the store to it, because the Login page used the storage of the React Native Application in order to save the user session.

The output of all the tests is:



```
MBP-de-Roberto:ariane robertoperez$ jest
 PASS  ./App.test.js (5.563s)
  ✓ Render app without crashing (204ms)
  ✓ Render custom button (5ms)
  ✓ Render back icon (2ms)
  ✓ Render list (88ms)
  ✓ Render map (5ms)
  ✓ Render input (2ms)
  ✓ Render text link (6ms)
  ✓ Render login screen (24ms)
  ✓ Render change password screen (29ms)
  ✓ Render profile screen (35ms)
  ✓ Render password recovery screen (16ms)
  ✓ Render home screen (2ms)
  ✓ Render field trips list screen (27ms)
```

Illustration 90. Jest tests output.

8.3. Usability testing

In order to test the usability of the app once it has been finished, a standard form has been created to collect opinions from different type of users (potential students and teachers, mobile app developers and non-technological people). Additionally, some guided activities have been developed in order to test if the process to make in order to create and perform the field trip and its task is simple or not. In total, 5 people were handled the application and the results are:

Usability	Always	Frequently	Ocasionally	Never
Do you know where you are inside the app?	100%			
Is any help needed for the app use?	100%			
Is the app usability simple?	100%			
Functionality	Always	Frequently	Ocasionally	Never
Does all the tasks work as expected?	100%			
Is the wait time of the app remarkable?			20%	80%
Did you encounter any error while using the app?				100%
Interface quality				
Graphic aspects	Very appropiate	Appropiate	Inapporpiate	Not appropiate at all
The font and size of the text is	100%			
The icons and images used are	100%			
The used colours are	80%	20%		
Interface design		Yes	No	Some times
Is it easy to use?		100%		
Is the screen design clear and attractive?		80%		20%
Do you think the app is well structured?		100%		
Do you think that the access to every section of the app is convenient?		100%		
General impressions				
User opinion		Yes	No	DK/NA
Do you think the app is useful?		100%		
Do you think that the app will be well received by the students?		80%		20%

Do you think the app will make the organization of the field trips easier for the teachers?	100%		
Do you think that the augmented reality features are suitable for this purpose?	80%		20%

Guided activity	Difficulty	Success
Create a new field trip	Easy: 100%	100%
Create 3 tasks for the new field trip	Easy: 100%	100%
See the field trip list in the app	Easy: 100%	100%
See the detail of the field trip	Easy: 100%	100%
Perform the three tasks created for that field trip	Easy: 80% Medium: 20%	100%
Query the answers and logs created for that field trip and tasks	Easy: 80% Medium: 20%	100%

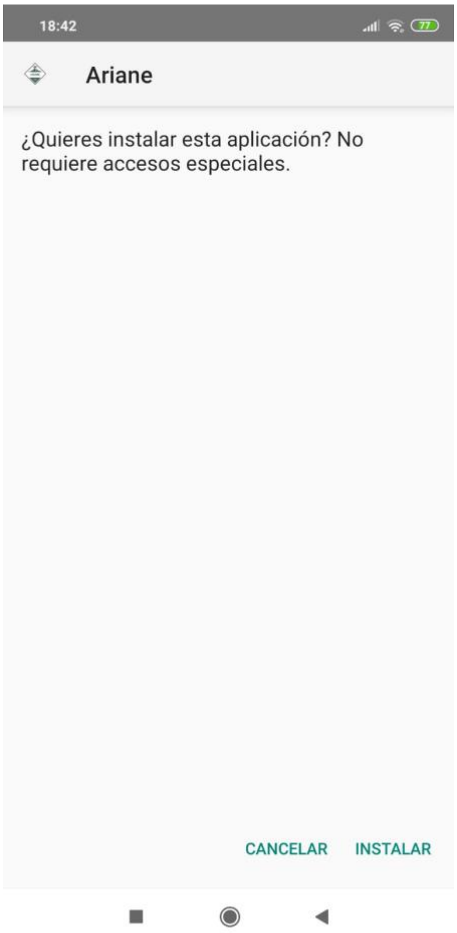
Regarding the results of this usability test, it can be said that the app is really usable, even for non-technological oriented users. It is very simple to use, and the interface design, images and text sizes are the optimal ones. In conclusion, it is a general opinion that this system will be well received in schools, giving the teachers another educational vehicle and less work in order to organize the trip, and giving the students a way to interact with the field trip and their tasks in another and more entertaining way.

9. System manual

9.1. Installation manual

The Ariane app installation will be done by downloading and opening an apk file in the device. In order to be able to install the apk, the option for installing applications from unknown origins has to be activated in the device settings. Depending on the Android version and device, this option is in one place or another, but usually it is under the Privacy or Security setting. Another way to search for this option is to type 'Install' in the search bar of the general settings menu of the device and this option will pop on screen.

After downloading the apk and opening it, this installation screen will pop up on screen. Then, by clicking on install, the installation of Ariane will start.



¿Quieres instalar esta aplicación? No requiere accesos especiales.

Illustration 92. Installation process (1).

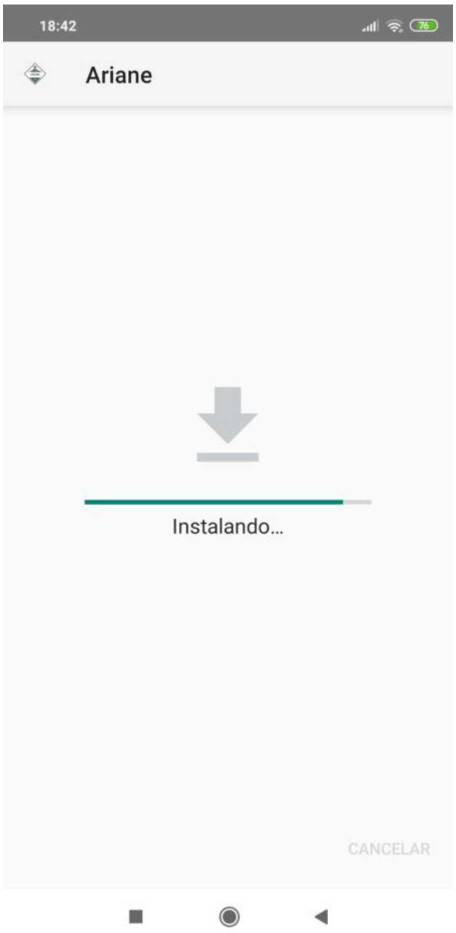
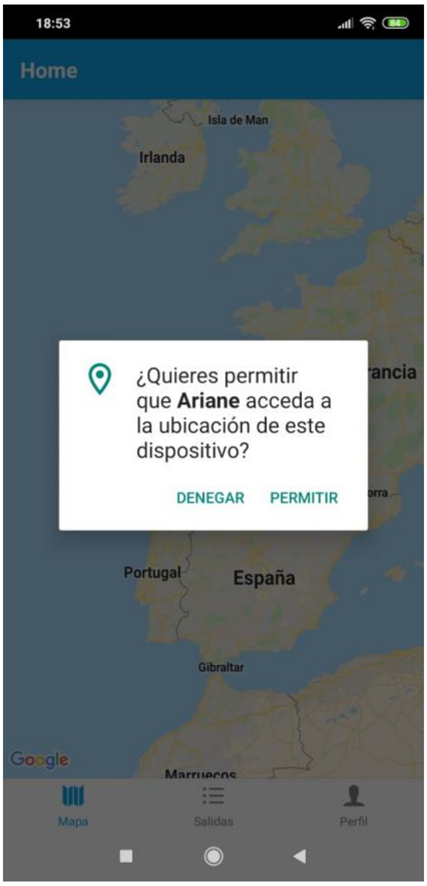


Illustration 91. Installation process (2).

Sometimes and depending on the Android version and the device, a confirmation screen for the installation of this type of applications will pop up on screen, asking the user to cancel the installation, or to continue anyways. This last option is the one which has to be clicked in order to install the app correctly.

Once the application is installed, the first page that the student sees is the login page. When the student logs in successfully to the app with the account that Ariane's admin has previously created for him, an alert will show on screen asking for permission to access the device geolocation in order to use it in the app.



¿Quieres permitir que **Ariane** acceda a la ubicación de este dispositivo?

Illustration 93. Geolocation permissions.

The student needs to allow the app to access it in order to keep track of the actions that he does in the app, like performing or seeing the details of the tasks. If the user denies this access, it can still be granted from the settings of the device, accessing to Ariane's profile and activating the location permissions.

9.2. Web application manual

When a school wants to use Ariane, the system developer creates an admin account for the person in charge of the system in that school. This admin is the one in charge of creating the different groups of students and the student and teacher accounts for the system use. Then, the teachers are the ones in charge of creating the different field trips and tasks to be performed by the students.

9.2.1. Groups creation

In order to create the different groups to later assign users and field trips to them, the admin has to head to the Students group tab of the tab menu.

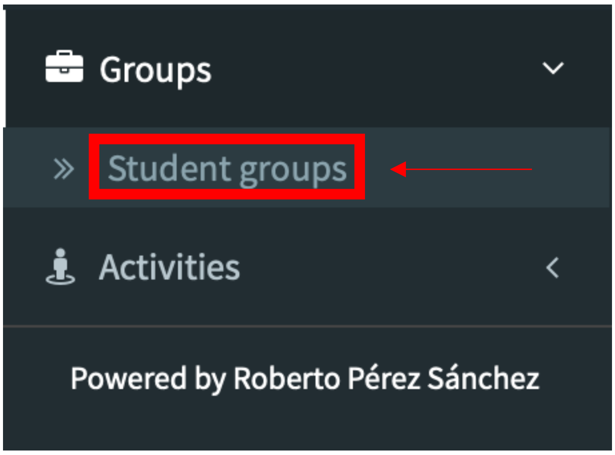
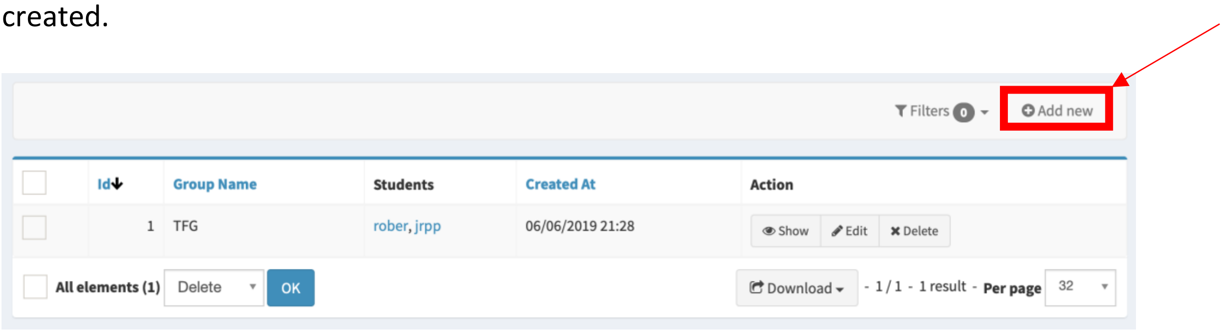


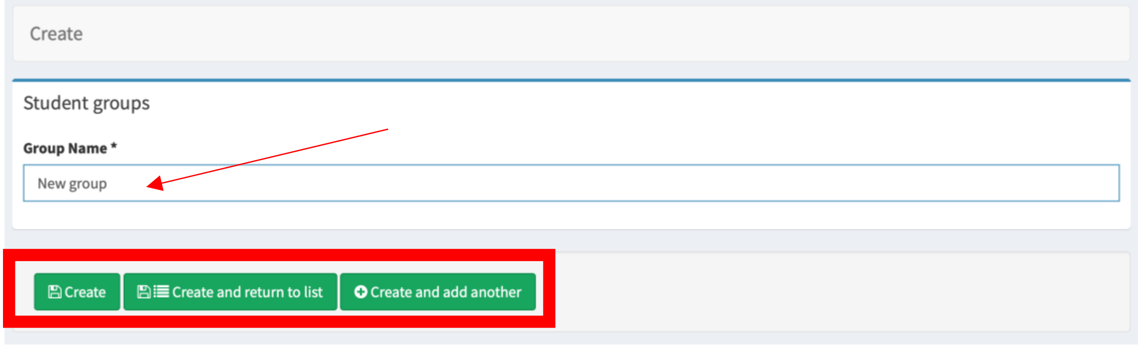
Illustration 94. Group tab in side menu.

Then, in the list view, the admin has to press on 'Add new' in order to add a new group. Then, by filling up the form and submitting the changes, the new group would be created.



	Id	Group Name	Students	Created At	Action
□	1	TFG	rober, jrpp	06/06/2019 21:28	Show Edit Delete

Illustration 95. Add new button.

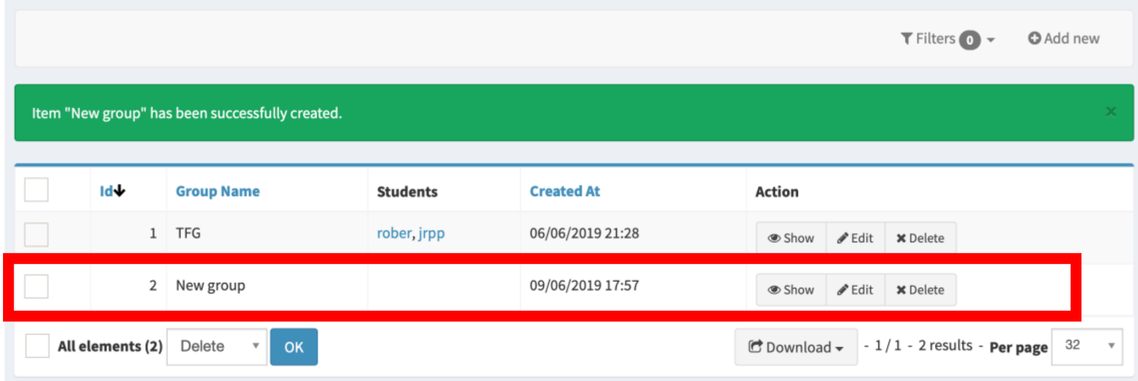


Create

Student groups

Group Name *: New group

Illustration 96. Student group form.



Item "New group" has been successfully created.

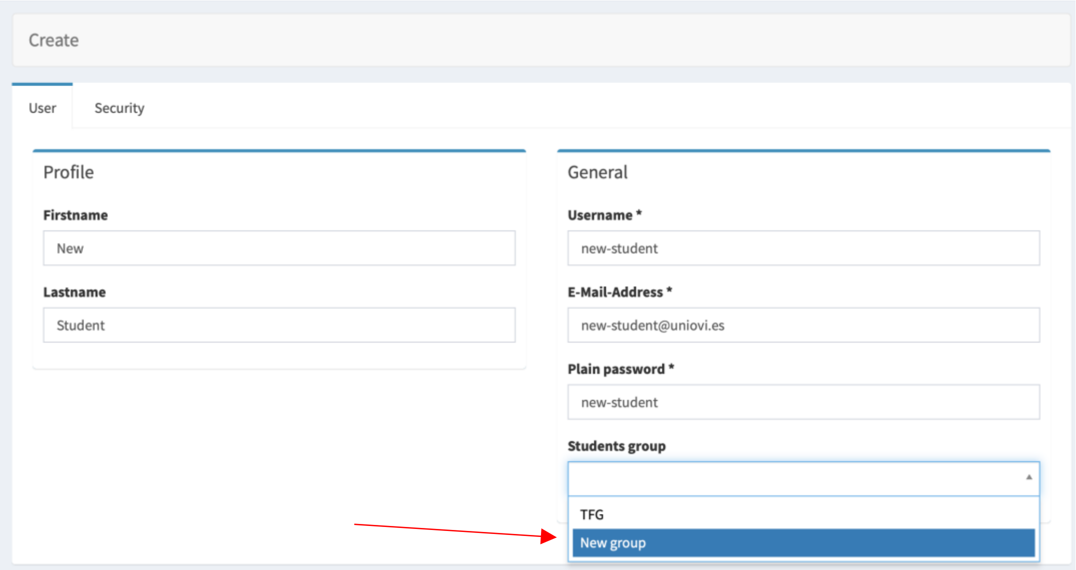
□	Id	Group Name	Students	Created At	Action
□	1	TFG	rober, jrpp	06/06/2019 21:28	Show Edit Delete
□	2	New group		09/06/2019 17:57	Show Edit Delete

□ All elements (2) Delete OK — Download - 1 / 1 - 2 results - Per page 32

Illustration 97. Newly created group.

9.2.2. User creation

Now, when creating a new user, the steps are the same: Go to the user tab, click on 'Add new' and fill the form. In it, the admin can see a select option where the group to which the user belongs can be selected.



Create

Profile

Firstname: New

Lastname: Student

General

Username *: new-student

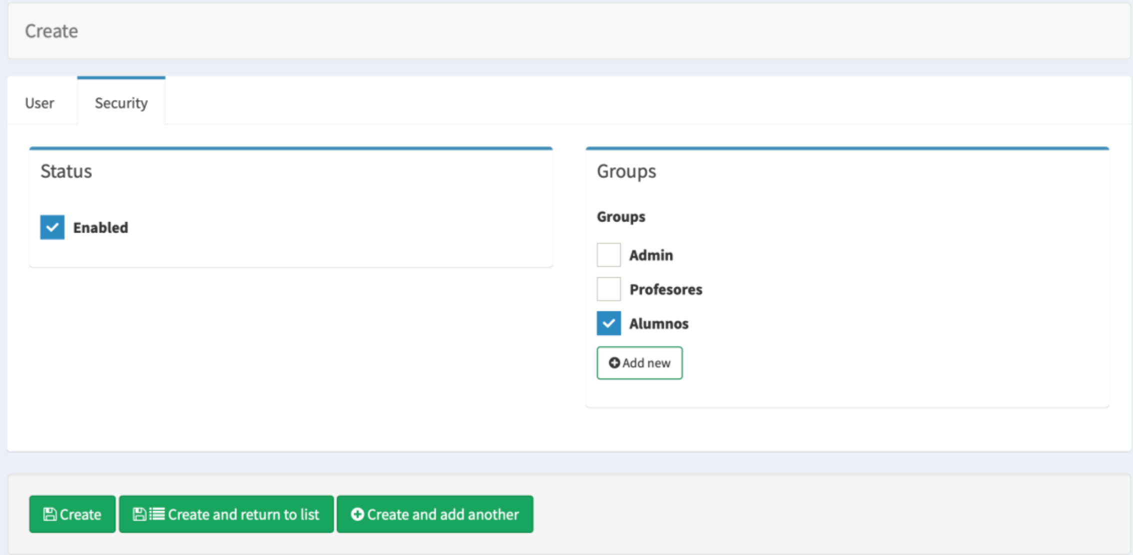
E-Mail-Address *: new-student@uniovi.es

Plain password *: new-student

Students group: TFG / New group

Illustration 98. New user form.

In the security tab, the admin can select the role of the newly created user to be teacher, student or another admin.



Status

☑ Enabled

Groups

Groups

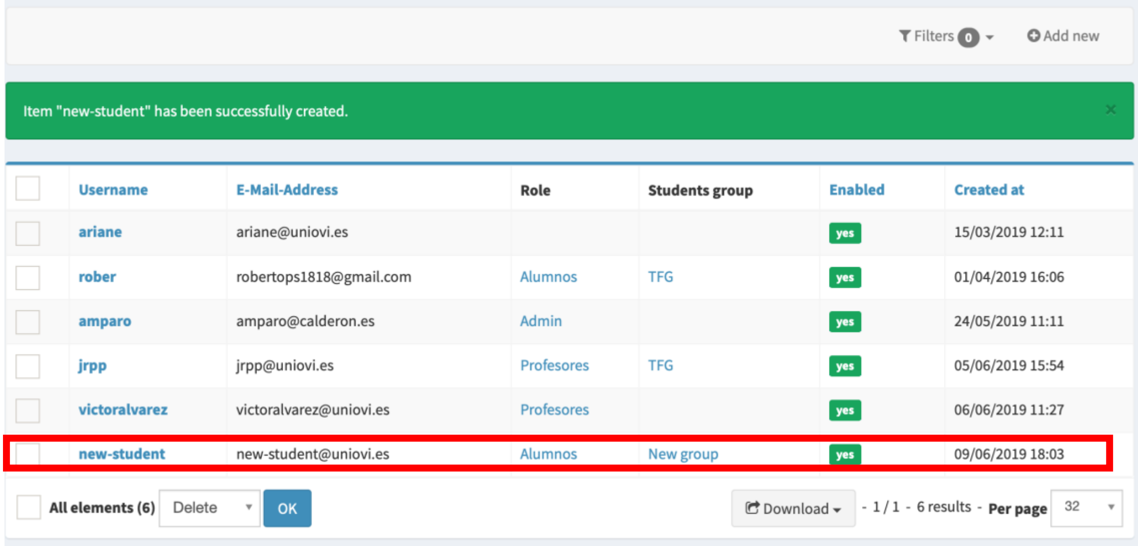
□ Admin

□ Profesores

☑ Alumnos

Illustration 99. User creation, security tab.

When submitting the form, the newly created user will appear in the list page.



Item "new-student" has been successfully created.

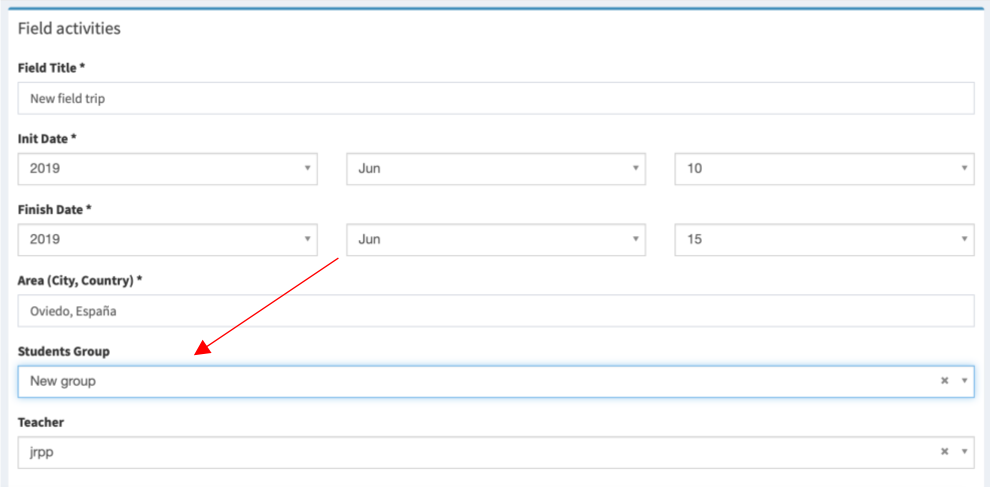
□	Username	E-Mail-Address	Role	Students group	Enabled	Created at
□	ariane	ariane@uniovi.es			yes	15/03/2019 12:11
□	rober	robertops1818@gmail.com	Alumnos	TFG	yes	01/04/2019 16:06
□	amparo	amparo@calderon.es	Admin		yes	24/05/2019 11:11
□	jrpp	jrpp@uniovi.es	Profesores	TFG	yes	05/06/2019 15:54
□	victoralvarez	victoralvarez@uniovi.es	Profesores		yes	06/06/2019 11:27
□	new-student	new-student@uniovi.es	Alumnos	New group	yes	09/06/2019 18:03

□ All elements (6) Delete · 1 / 1 · 6 results · Per page 32

Illustration 100. Newly created user.

9.2.3. Field trip creation

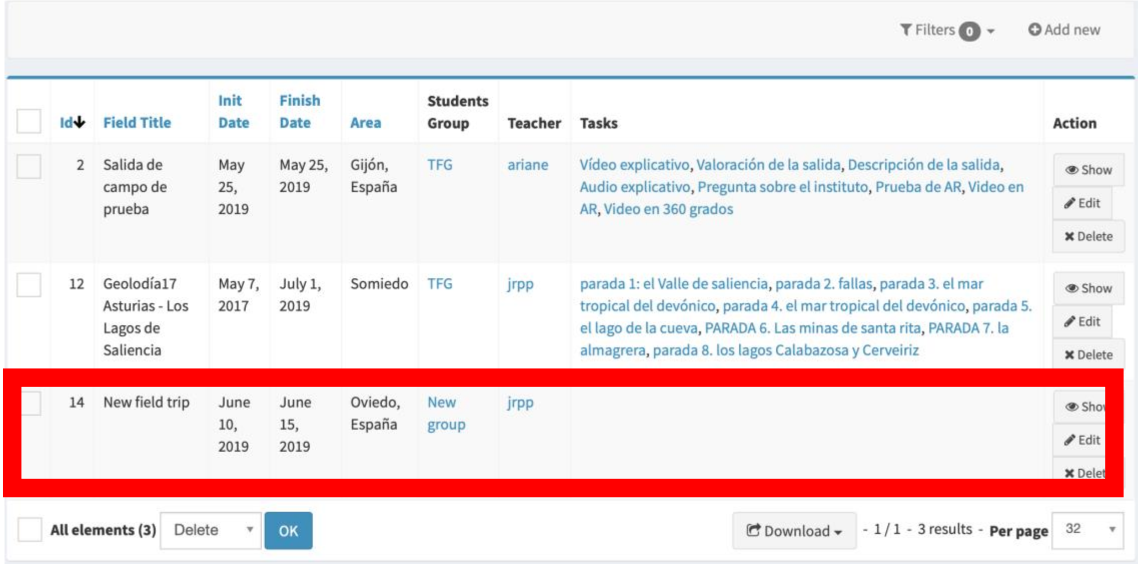
Then, for the creation of the field trip, the teacher will head to the field activity tab in the side menu and click on ‘Add new’. When submitting the form, the group to which this trip is assigned to, in order to the students to visualize the newly created trip in the mobile app.



Field activities

Field Title *	New field trip	
Init Date *	2019	Jun	10
Finish Date *	2019	Jun	15
Area (City, Country) *	Oviedo, España	
Students Group	New group	
Teacher	jrpp	

Illustration 101. New field trip form.

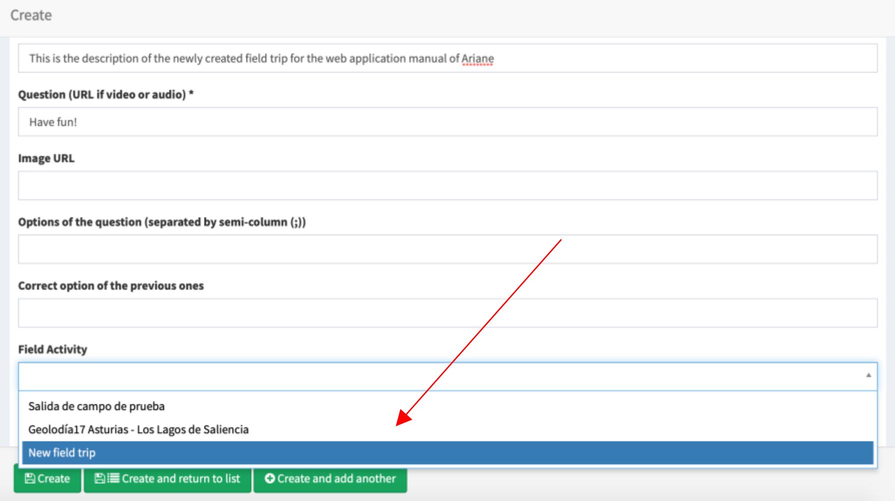


□	Id↓	Field Title	Init Date	Finish Date	Area	Students Group	Teacher	Tasks	Action
□	2	Salida de campo de prueba	May 25, 2019	May 25, 2019	Gijón, España	TFG	ariane	Vídeo explicativo, Valoración de la salida, Descripción de la salida, Audio explicativo, Pregunta sobre el instituto, Prueba de AR, Video en AR, Video en 360 grados	Show Edit Delete
□	12	Geolodía17 Asturias - Los Lagos de Saliencia	May 7, 2017	July 1, 2019	Somiedo	TFG	jrpp	parada 1: el Valle de saliencia, parada 2. fallas, parada 3. el mar tropical del devónico, parada 4. el mar tropical del devónico, parada 5. el lago de la cueva, PARADA 6. Las minas de santa rita, PARADA 7. la almagrera, parada 8. los lagos Calabazosa y Cerveiriz	Show Edit Delete
□	14	New field trip	June 10, 2019	June 15, 2019	Oviedo, España	New group	jrpp		Show Edit Delete

Illustration 102. Newly created field trip.

9.2.4. Tasks creation

Now, let's create a pair of tasks for the newly created field trip. For that, the teacher has to navigate to the tasks page in the side menu and, like always, click on 'Add new'. In the form, the field trip which the task is associated to needs to be referenced.



Create

This is the description of the newly created field trip for the web application manual of Ariane

Question (URL if video or audio) *

Have fun!

Image URL

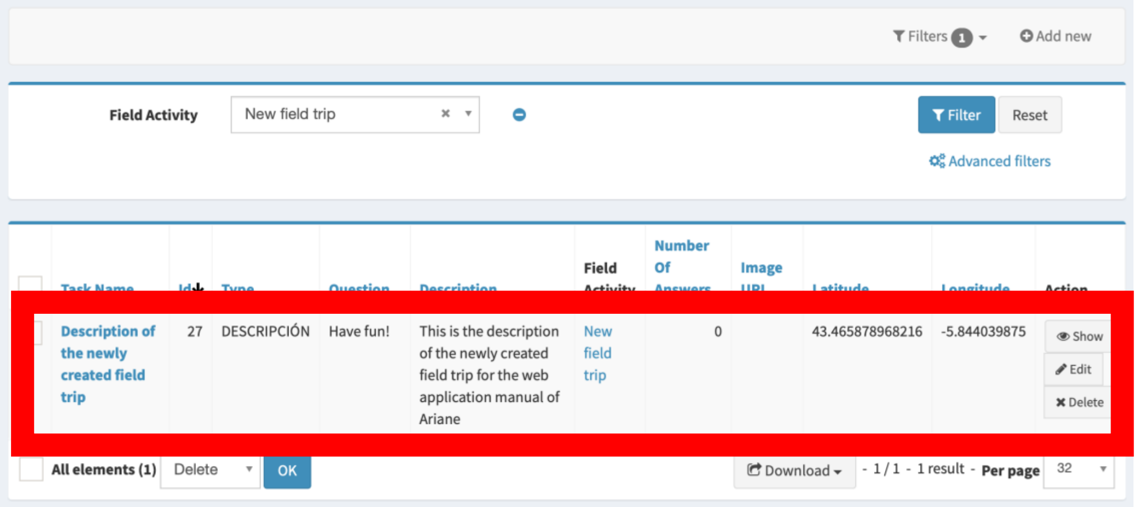
Options of the question (separated by semi-column (;))

Correct option of the previous ones

Field Activity

Salida de campo de prueba
Geolodía17 Asturias - Los Lagos de Saliencia
New field trip

Illustration 103. New task form.



Field Activity: New field trip

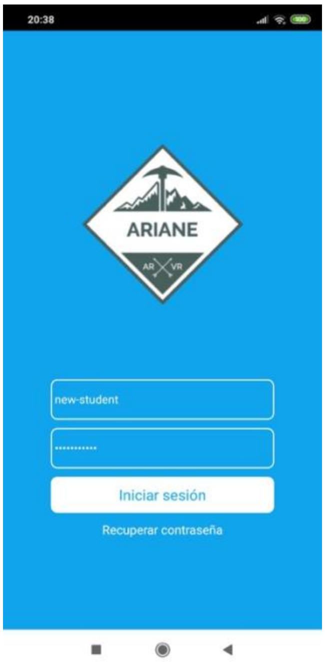
Task Name	Id	Type	Question	Description	Field Activity	Number Of Answers	Image URL	Latitude	Longitude	Action
Description of the newly created field trip	27	DESCRIPCIÓN	Have fun!	This is the description of the newly created field trip for the web application manual of Ariane	New field trip	0		43.465878968216	-5.844039875	Show Edit Delete

Illustration 104. Newly created task.

Teachers can repeat this process all the times they want with different types of tasks, in order to give the student a more complete experience of the field trip.

9.2.5. Mobile app visualization

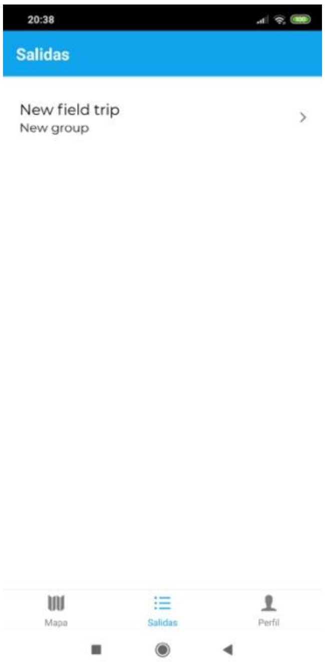
Now, if the newly created student within the newly created group logs in to the mobile application, he will see the new field trip that the teacher has just create and the task related to it.



new-student

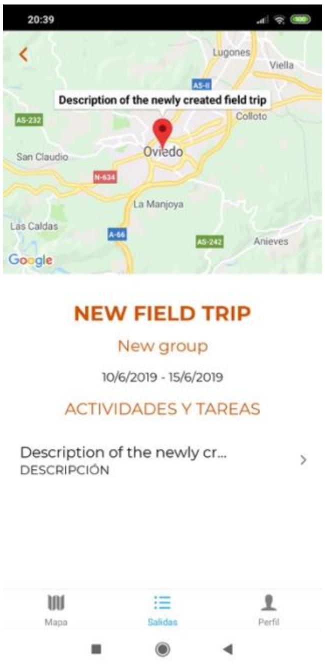
••••••••••

Illustration 106. Login of the new user.



New field trip
New group

Illustration 105. Field trips of the new user.



NEW FIELD TRIP

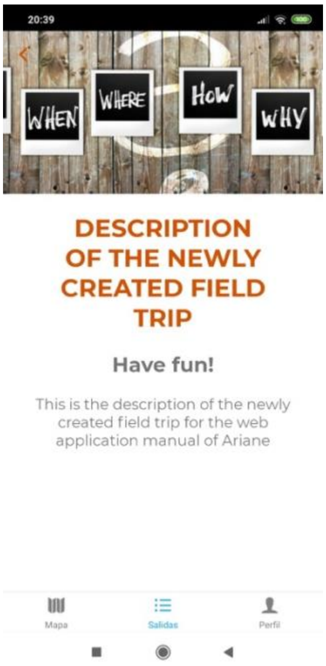
New group

10/6/2019 - 15/6/2019

ACTIVIDADES Y TAREAS

Description of the newly cr...
DESCRIPCIÓN

Illustration 108. Detail of the new field trip.



DESCRIPTION OF THE NEWLY CREATED FIELD TRIP

Have fun!

This is the description of the newly created field trip for the web application manual of Ariane

Illustration 107. Detail of the new task.

9.2.6. Special formats in the web application

In some of the entity forms in the web application, some of its fields needs to have a special format in order to the web application to consume them correctly. They are the following:

- Area of the field trip. The area where the field trip is performed needs to be introduced in the format: City, Country. This is because the mobile app takes this area and transforms it to a location in the map. But, if the name of the city exists in two different countries (for example, Barcelona in Spain and Ecuador), the transformation cannot be done unless the country is specified.
- Options to the question in test task. The options available for answering the question have to be separated by semi-column (;) in order to the mobile application to take the options and put them in the select component.
- URL of the videos and audios. When the task is of type video, 360 video, audio or youtube, the link to the resources has to be introduced in the question form field, as it is explained in the label of the text input.

9.3. Mobile app manual

When the admin of the school creates an account for each user, they can log in to the application using the credentials created. As seen in 9.2, each of the field trips and tasks created for the user group will be displayed in the app.

In order to perform the task, the user only needs to click on the item of the list that he wants to. Depending on the type, the task would be different. Some of them will require a feedback or answer from the student and other are just descriptive or interactive.

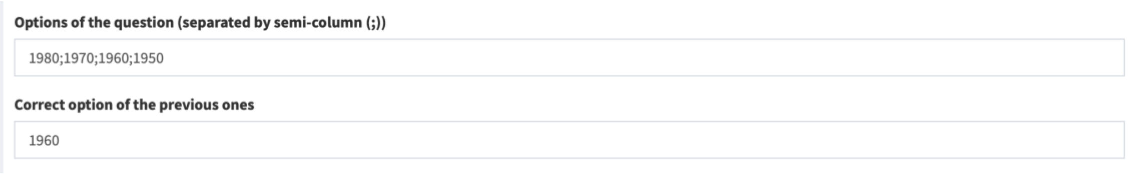
These answers, and all the actions performed by the users in the app, like playing an audio or video, or even seeing the detail of a task, will be stored in the web application in order for the teachers to analyse all this information.

9.3.1. Tasks realization

There exist the following types of tasks:

1. Description: In this task, a general description of the field trip is displayed on screen
2. AR: In this task, a description similar to the previous one is displayed, but using Augmented Reality. This experience can only be performed by students whose mobile device supports ARCore. The list of compatible devices could be found here: https://developers.google.com/ar/discover/supported-devices
3. Test: This task consists on two parts. In the first place, when the teacher creates it, he has to introduce the available options and the correct answer. The format of this inputs can be found in chapter 9.2.6. Then, in the mobile app, the students need to select one of the options and submit the test. If they get the question

right, the app will show a green success message. But, if they answer incorrectly, a red error message will pop on screen telling them the correct answer.



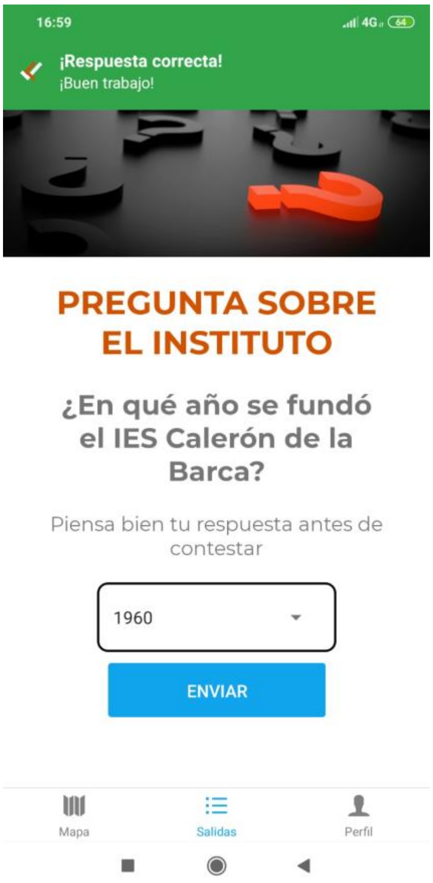
Options of the question (separated by semi-column (;))

1980;1970;1960;1950

Correct option of the previous ones

1960

Illustration 109. Options of the test task.



16:59

¡Respuesta correcta!
¡Buen trabajo!

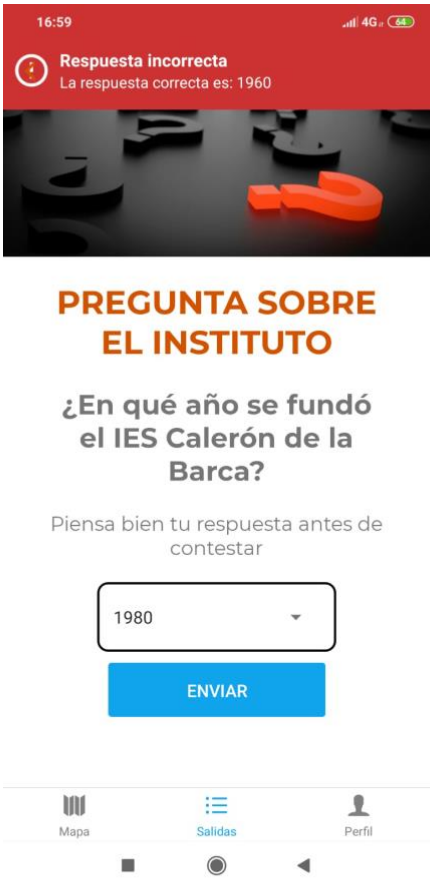
PREGUNTA SOBRE EL INSTITUTO

¿En qué año se fundó el IES Calerón de la Barca?

Piensa bien tu respuesta antes de contestar

1960

Illustration 111. Correct answer to the test task.



16:59

Respuesta incorrecta
La respuesta correcta es: 1960

PREGUNTA SOBRE EL INSTITUTO

¿En qué año se fundó el IES Calerón de la Barca?

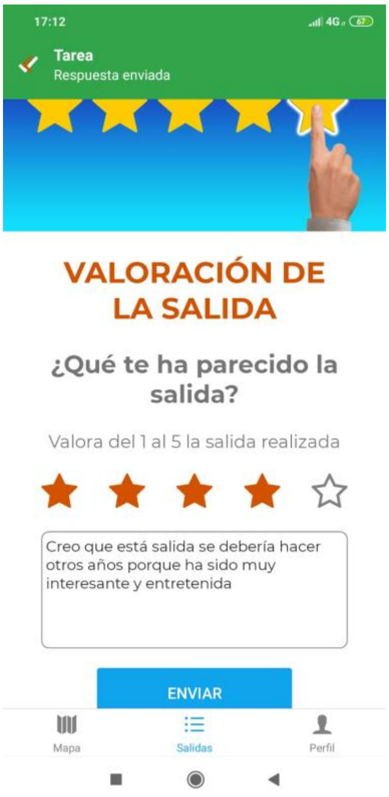
Piensa bien tu respuesta antes de contestar

1980

Illustration 110. Incorrect answer to the test task.

4. Youtube: This task just plays a Youtube video about the field trip in order to hive the student more information about it
5. Audio: This task plays an audio related to the field trip. The audio can be paused and played the times that the student wants to. He can also move around the audio with a slider in order to control its reproduction.
6. Video: This task plays a video from the URL specified by the teacher in Augmented Reality. Like the other AR tasks, these tasks can only be performed by ARCore compatible devices.

7. Video 360: In order to give a more immersive experience to the student, a 360 degrees video task can be created, specifying the URL of the 360 video when creating the task (only ARCore devices)
8. Rating and opinion: This task is used in order to get feedback from the students about their rating and opinion of the field trip, that can be rated from 1 to 5 with a star bar. Then, the students can send their opinions and suggestions, which will be stored in the web application for the teachers to see them.



17:12

Tarea
Respuesta enviada

VALORACIÓN DE LA SALIDA

¿Qué te ha parecido la salida?

Valora del 1 al 5 la salida realizada

★ ★ ★ ★ ☆

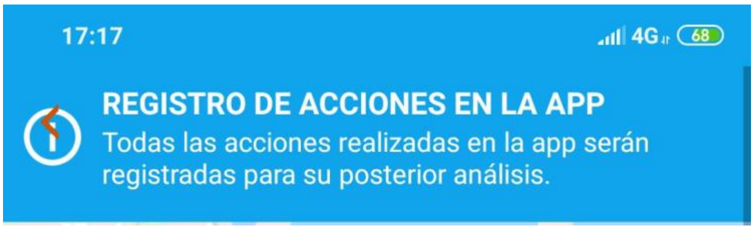
Creo que está salida se debería hacer otros años porque ha sido muy interesante y entretenida

ENVIAR

Illustration 112. Rating and opinion sent.

9.3.2. Answers and logs visualization

Whenever a student enters into the detail of a field trip, the mobile app will give him an information message telling that all the info and actions performed in the app will be stored for its future study and analysis. This way, the student is always concerned about the use that he or she has to give to the app.



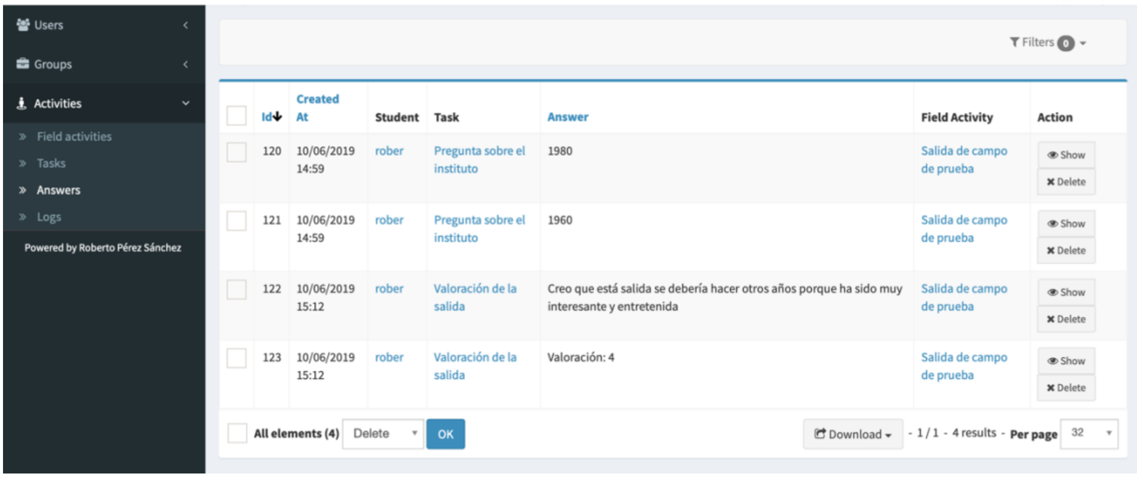
17:17

REGISTRO DE ACCIONES EN LA APP
Todas las acciones realizadas en la app serán registradas para su posterior análisis.

Illustration 113. Info registry message.

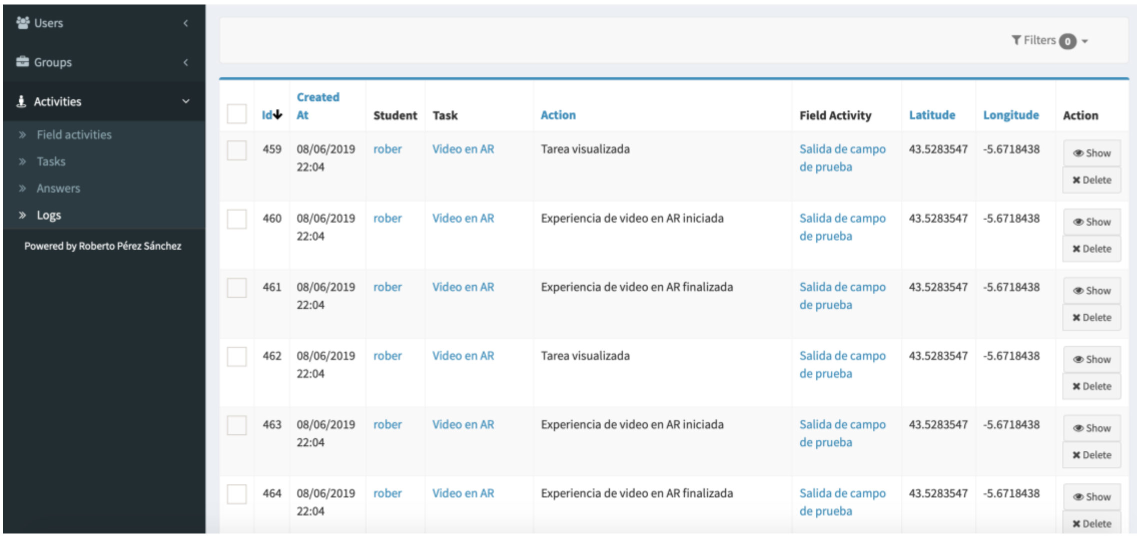
All the answers and logs created by the mobile application can be displayed and filtered by the teachers in the web application going to the answer and logs tabs in the side menu of the web.

When the list page of both is rendered, the teacher can see the detail of every individual registry, as well as filter the general list or even download the lists to an Excel file in their computer.



□	Id	Created At	Student	Task	Answer	Field Activity	Action
□	120	10/06/2019 14:59	rober	Pregunta sobre el instituto	1980	Salida de campo de prueba	Show Delete
□	121	10/06/2019 14:59	rober	Pregunta sobre el instituto	1960	Salida de campo de prueba	Show Delete
□	122	10/06/2019 15:12	rober	Valoración de la salida	Creo que está salida se debería hacer otros años porque ha sido muy interesante y entretenida	Salida de campo de prueba	Show Delete
□	123	10/06/2019 15:12	rober	Valoración de la salida	Valoración: 4	Salida de campo de prueba	Show Delete

Illustration 114. Answers list.



□	Id	Created At	Student	Task	Action	Field Activity	Latitude	Longitude	Action
□	459	08/06/2019 22:04	rober	Video en AR	Tarea visualizada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete
□	460	08/06/2019 22:04	rober	Video en AR	Experiencia de video en AR iniciada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete
□	461	08/06/2019 22:04	rober	Video en AR	Experiencia de video en AR finalizada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete
□	462	08/06/2019 22:04	rober	Video en AR	Tarea visualizada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete
□	463	08/06/2019 22:04	rober	Video en AR	Experiencia de video en AR iniciada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete
□	464	08/06/2019 22:04	rober	Video en AR	Experiencia de video en AR finalizada	Salida de campo de prueba	43.5283547	-5.6718438	Show Delete

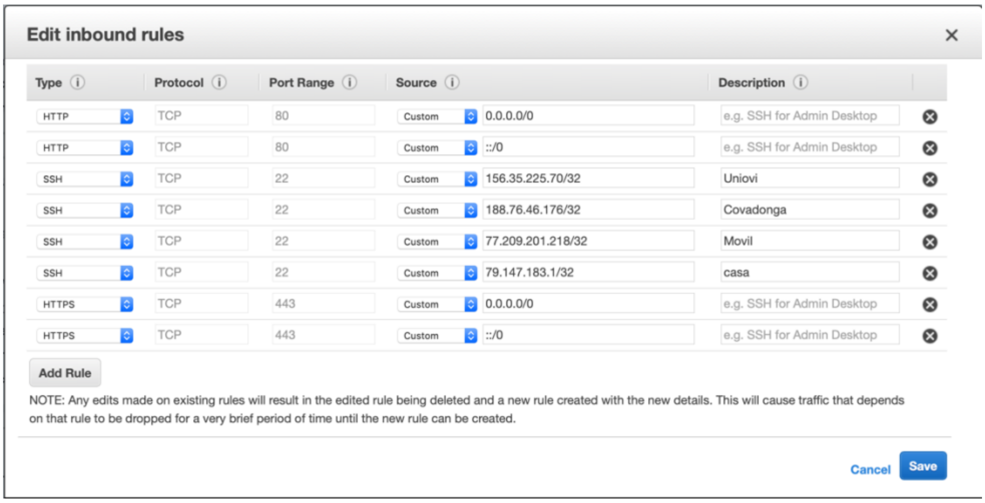
Illustration 115. Logs list.

Because every single action in the mobile app is stored, the logs list will be huge, so the filters available in the right-hand top corner of the list will result really handy in order to search for individual users, tasks or field trips.

9.4. Developer manual

In order to deploy the web application in a new server, the developer will have to follow the next steps:

1. Create an EC2 (Ubuntu Server) instance in Amazon Web Services. The process is pretty straight-forward because no custom configuration will be needed, the default fields for the instance work perfectly in this case. At the end of the process, the user must download the private key (.pem file) provided by AWS in order to use it to connect to the server via SSH
2. Create an RDS instance in Amazon Web Services. In this case, the default configuration will be used as well. The user must introduce the name of the database, as well as a root user and password. The type of DBMS is up to the user, in the case the recommended ones are MySQL or MariaDB in order to the Symfony project to map the entities into tables.
3. Once the two instances are created and running, the user must create the security groups. In the AWS side menu of the EC2 configuration, go to Security Groups. Two security groups have to be created, one for each instance. The connections for the instances will be declared in the Inbound section of each security group
    a. In the EC2 security group, the user will grant SSH connection from the IPs that he wants and HTTP and HTTPS connection from everywhere



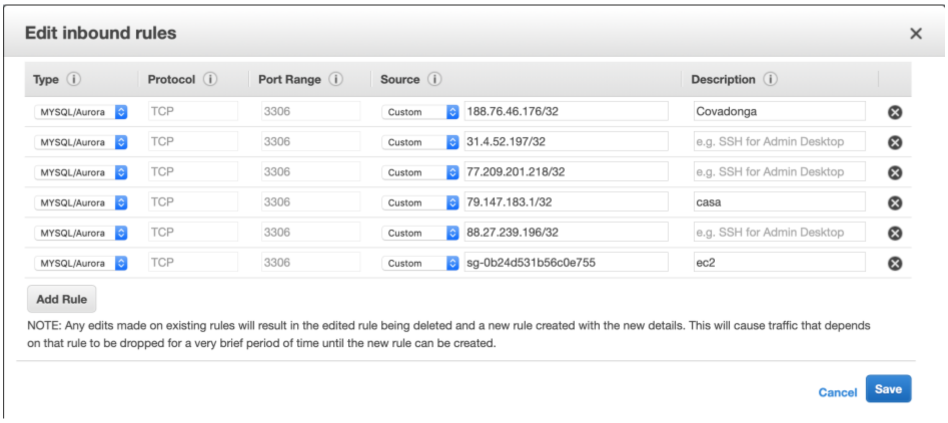
Edit inbound rules

Type	Protocol	Port Range	Source		Description
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	::/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom	156.35.225.70/32	Uniovi
SSH	TCP	22	Custom	188.76.46.176/32	Covadonga
SSH	TCP	22	Custom	77.209.201.218/32	Movil
SSH	TCP	22	Custom	79.147.183.1/32	casa
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	::/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Illustration 116. EC2 security group.

    b. In the RDS security group, access through MySQL/Aurora will be grantes from the IPs selected by the user, as well as from the EC2 instance, introducing the ID of the security group previously created



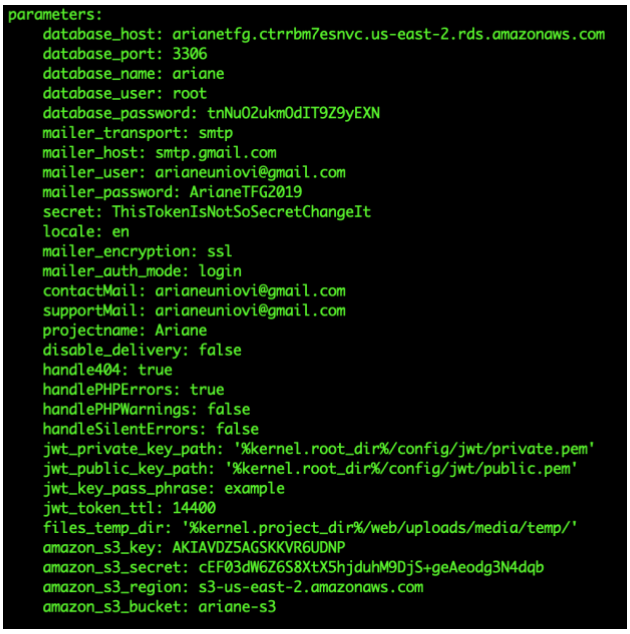
Edit inbound rules

Type	Protocol	Port Range	Source		Description
MYSQL/Aurora	TCP	3306	Custom	188.76.46.176/32	Covadonga
MYSQL/Aurora	TCP	3306	Custom	31.4.52.197/32	e.g. SSH for Admin Desktop
MYSQL/Aurora	TCP	3306	Custom	77.209.201.218/32	e.g. SSH for Admin Desktop
MYSQL/Aurora	TCP	3306	Custom	79.147.183.1/32	casa
MYSQL/Aurora	TCP	3306	Custom	88.27.239.196/32	e.g. SSH for Admin Desktop
MYSQL/Aurora	TCP	3306	Custom	sg-0b24d531b56c0e755	ec2

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Illustration 117. RDS security group.

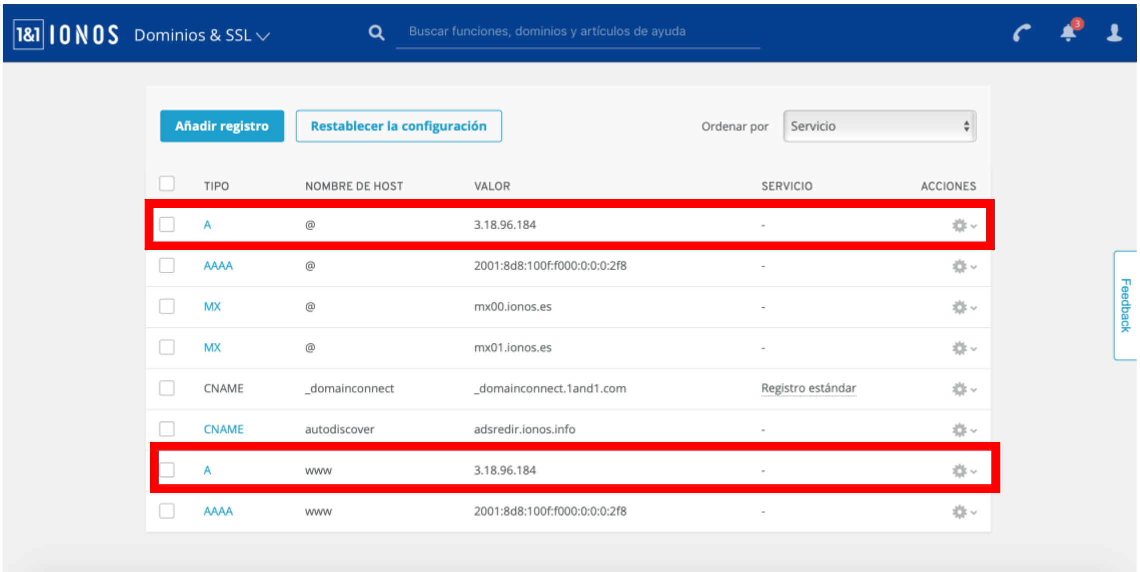
4. After creating the security groups, the developer has to connect to the EC2 instance using an SSH connection using the private key previously downloaded. In Windows, the connection can be made with some program like PuTTY. In MacOS, using the command line and the command **ssh -i PRIVATEKEY.pem ubuntu@EC2_IP**.
5. Once the user is connected to the instance, some basic packages have to be installed: php, git and composer. The following required components will be asked to be installed by the own system.
6. Clone the Ariane's web application code in the directory /var/www or transfer the files using some program like Filezilla.
7. Move to Ariane's root directory and run: **composer install.** If some permissions related error occurs, just run it with **sudo**. The parameters needed in order to configure the project will be asked to the developer. These are the current parameters use for this deploy:



```
parameters:
    database_host: arianetfg.ctrrbm7esnvc.us-east-2.rds.amazonaws.com
    database_port: 3306
    database_name: ariane
    database_user: root
    database_password: tnNuO2ukmOdIT9Z9yEXN
    mailer_transport: smtp
    mailer_host: smtp.gmail.com
    mailer_user: arianeuniovi@gmail.com
    mailer_password: ArianeTFG2019
    secret: ThisTokenIsNotSoSecretChangeIt
    locale: en
    mailer_encryption: ssl
    mailer_auth_mode: login
    contactMail: arianeuniovi@gmail.com
    supportMail: arianeuniovi@gmail.com
    projectname: Ariane
    disable_delivery: false
    handle404: true
    handlePHPErrors: true
    handlePHPWarnings: false
    handleSilentErrors: false
    jwt_private_key_path: '%kernel.root_dir%/config/jwt/private.pem'
    jwt_public_key_path: '%kernel.root_dir%/config/jwt/public.pem'
    jwt_key_pass_phrase: example
    jwt_token_ttl: 14400
    files_temp_dir: '%kernel.project_dir%/web/uploads/media/temp/'
    amazon_s3_key: AKIAVDZ5AGSKKVR6UDNP
    amazon_s3_secret: cEF03dW6Z6S8XtX5hjduhM9DjS+geAeodg3N4dqb
    amazon_s3_region: s3-us-east-2.amazonaws.com
    amazon_s3_bucket: ariane-s3
```

Illustration 118. Parameters.yml file.

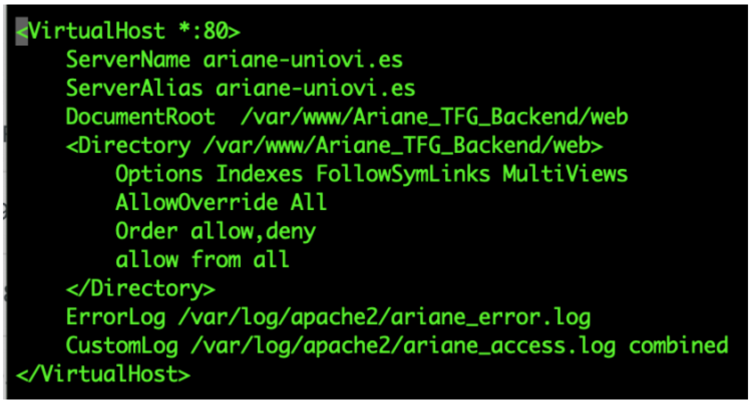
8. The most important parameters are the database related, where you have to enter the RDS instance public URL, the name of the database and the root user and password entered when creating the RDS instance. The rest can be the same as the one used in this deploy.
9. Then, a public domain has to be purchased. In this case, the domain ariane-uniovi.es was purchased at 1&1 IONOS, which offers a free year with .es domains. There, in the domain configuration, the public IP of the EC2 instance has to be configured in the DNS configuration page of IONOS, like the following:


1&1 IONOS
Dominios & SSL
Buscar funciones, dominios y artículos de ayuda
Añadir registro
Restablecer la configuración
Ordenar por
Servicio
Feedback

□	TIPO	NOMBRE DE HOST	VALOR	SERVICIO	ACCIONES
□	A	@	3.18.96.184	-	
□	AAAA	@	2001:8d8:100f:f000:0:0:0:2f8	-	
□	MX	@	mx00.ionos.es	-	
□	MX	@	mx01.ionos.es	-	
□	CNAME	_domainconnect	_domainconnect.1and1.com	Registro estándar	
□	CNAME	autodiscover	adsredir.ionos.info	-	
□	A	www	3.18.96.184	-	
□	AAAA	www	2001:8d8:100f:f000:0:0:0:2f8	-	

Illustration 119. Domain DNS configuration

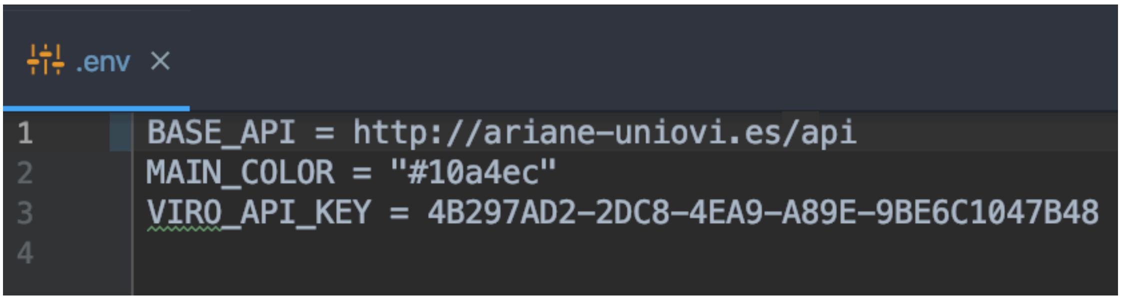
10. Last, but not least, a virtual host has to be created in the EC2 instance in order to redirect to Ariane's code. In the directory /etc/apache2/sites-available, a new conf file will be created with the wanted name. Inside it, the developer will type the document root and destiny of the system. Here it is the conf file for this deploy (the changes would be the path to the code and the domain previously purchased):



```
<VirtualHost *:80>
    ServerName ariane-uniovi.es
    ServerAlias ariane-uniovi.es
    DocumentRoot  /var/www/Ariane_TFG_Backend/web
    <Directory /var/www/Ariane_TFG_Backend/web>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>
    ErrorLog /var/log/apache2/ariane_error.log
    CustomLog /var/log/apache2/ariane_access.log combined
</VirtualHost>
```

Illustration 120. Conf file.

11. In order to enable the site, the developer has to run the following command: **a2ensite path_to_conf_file** and then restart the Apache server with **sudo service apache2 restart**.
12. When the 1&1 IONOS server is finished indexing the system, Ariane would be available in the purchased domain.
13. For the app, in order to use the new API, a new release has to be compiled. The path to the API has to be changed in the .env and .env.production. The path would be http://NEWDOMAIN/api



```
BASE_API = http://ariane-uniovi.es/api
MAIN_COLOR = "#10a4ec"
VIRO_API_KEY = 4B297AD2-2DC8-4EA9-A89E-9BE6C1047B48
```

Illustration 121. .env app file

14. Finally, the new release APK will be created running the command: **react-native run-android –variant=gvrRelease,** with the device attached to the computer.

10. Detailed budget

10.1. Development

Task	Hours	Cost (per hour)	Total
Analysis	55	35€/h	1.925 €
Development	163	30€/h	4.890 €
Design	25	25€/h	625 €
Testing	30	30€/h	900 €

TOTAL	8.340 €

10.2. Server and licenses

Product	Months of use	Cost	Total
Amazon EC2 instance (a1.xlarge)	4	0,091€/h	65,52 €
Amazon RDS instance (db.t3.xlarge)	4	0,24€/h	172,80 €
Github Entreprise	4	6,25€/month	25 €
WebStorm license	4	12,90€/month	51,60 €
PhpStorm license	4	19,90€/month	79,60 €
Ionos domain ariane-uniovi.es	4	10€/year	3,33 €

TOTAL	397,85 €

10.3. Indirect costs

Product	Months of use	Cost (per month)	Total
MacBook Pro 2015	4	21,7€/month	86,8 €
Light	4	150€/month	600 €
Water	4	50€/month	200 €
Internet	4	50€/month	200 €
Insurance	4	100€/month	400 €

TOTAL	1.486,8 €

10.4. Costs budget

Product	Total
Development	8340 €
Server and licenses	397,85 €
Indirect costs	1.486,8 €

Total	10.224,65 €
Fund for risks (5%)	511,23 €
TOTAL (with funds)	10.735,88 €

10.5. Client budget

For the client budget, the benefits wanted from the system development (20%) and the indirect costs have been prorated between the 4 different activities of the development phase.

Product	Total
Analysis	2950, 17 €
Design	957,85 €
Development	7494,20 €
Testing	1379,30 €

Total	12.781,52 €
IVA (21%)	2.684,12 €
TOTAL (with IVA)	15.456,64 €

11. Extensions and conclusions

11.1. Extensions

11.1.1. Augmented Reality features

Being a booming technology, a variety of new features for augmented reality tools may appear in the near future. It would be interesting to add some more features about augmented reality to the app, for example a virtual tutorial about how to use it, instead of having to look in the user manual in order to know the main characteristics of the app.

Moreover, it would be interesting to add image recognition to the application. This is a feature present in other augmented reality libraries, like Vuforia, that recognizes an image and present an AR experience different for every image detected.

11.1.2. System compiling for different schools

Ariane was thought as a system for schools, as it can be very helpful when organising a field trip and to make students more active while doing them. It would be a good idea to have the system to be independent between schools, so every school can only access their own field trips and tasks.

Now, this approach can be made deploying one instance of the system in different servers and domains, but it would be a great extension for Ariane to be deployed in one single instance and to be able to have different modules with their individual entities for each school that uses the system.

11.1.3. Different types of tasks

If the application is used in some school or field trip, it is likely that the teachers themselves think of another type of tasks that would be interesting to have in their field trips, for example, a drawing module or an image uploader. These tasks would take time to program and deploy in the application, but all the considerations and suggestions of teachers and students can be studied and maybe developed in the future.

11.1.4. iOS development

Although React Native offers the possibility of compiling the code to Android and iOS, the process of creating Ariane has been focused on the Android development. I do not have an Apple device neither an Apple Developer Account in order to test all the functionalities of the application in iOS, so I left it apart for the moment. But, taking into account that the code is the same for the iOS development, and very little features would have to be adapted, the iOS development can be the next step of Ariane.

11.2. Conclusions

Since my tutor of this final degree work, Víctor, talked to us in the class of “Realidad y Accesibilidad Aumentadas” about Ariane, I really liked the idea behind it. The possibility to add some kind of technology in the process of the management of the field trips at school, and moreover, the possibility to add augmented reality to it, was something that really attracted me from the beginning.

I started thinking about which technology would be best in order to develop this system. From the beginning I knew that I wanted to develop Ariane to be available for both Android and iOS, so my first thought was Ionic, as I was working with it in GooApps, the company where I am doing my business practices. But this idea was quickly rejected in favour of React Native, which gives a lot more support for multiplatform augmented reality features. Moreover, in GooApps we started using this technology in our development, so I had a lot of support from my colleagues there.

During the development of Ariane, I learned a lot of new things about Augmented Reality and how it is used in science and engineering, as well as how to implement it using different technologies apart from the ones seen in the AR subject in the degree. On the other hand, I have had a lot of problems trying to add different AR libraries to the project, which sometimes lead me to break all the project and having to start a completely new one, but I think that the final result is the one I expected from the beginning.

Additionally, one of the most satisfying things about developing Ariane was the possibility to go to my high school, IES Calderón de la Barca in Gijón, in order to give a speech to some students and teachers of 1st course of Bachillerato in order to present to them the web and mobile application developed and its use, as well as talking to them about the Software Engineering degree and any doubt about university or working life they might have. It was very satisfactory to see the students and teachers really interested in how the system was made and all the process behind it. Moreover, although it is not a field trip for its own, Ariane might be tested in a trip to Dublin that the high school is organising in September, so it would be a great scenario to really know if my system is capable of giving an enjoyable experience to the students.

In conclusion, I have learned many new aspects about Augmented Reality and its uses, as well as having to deal with technologies never used before as React Native and PHP, that gave the experience I was wanting to obtain, combining it with other knowledge previously obtained in the different subjects of the degree, which helped me a lot, especially in the part of analysis, design, testing and documentation.

12. References

[Airbnb19] Airbnb. “Enzyme docs”. https://airbnb.io/enzyme/docs/api/. 2019

[AllianceTeck19] AllianceTeck. “Ionic pros and cons”. https://www.alliancetek.com/blog/post/2016/04/06/the-pros-cons-of-developing-with-ionic.aspx. 2019

[AmazonEC2-19] Amazon Web Services. “EC2”. https://aws.amazon.com/ec2/?nc1=h_ls. 2019

[AmazonRDS19] Amazon Web Services. “RDS”. https://aws.amazon.com/rds/?nc1=h_ls. 2019

[Deloitte19] Deloitte. “Ionic Framework”. https://www2.deloitte.com/es/es/pages/technology/articles/Ionic-principales-framework-visuales.html. 2019

[Diligent19] Diligent. “¿Qué es Symfony?”. https://www.diligent.es/framework-symfony-php/#. 2019

[Jest19] Jest. “Documentation and API”. https://jestjs.io. 2019

[Knoji19] Knoji. “Advantages of school field trips”. https://learning.knoji.com/advantages-of-school-field-trips/. 2019

[MediumJest19] Medium. “Unit testing in React Native”. https://medium.com/@hdsenevi/unit-testing-in-react-native-with-jest-and-enzyme-40cd7dabb6f1. 2019

[MediumRN19] Medium. “Ventajas y desventajas de React Native”. https://medium.com/weblineindia/what-is-react-native-development-advantages-and-disadvantages-e5bf052473a7. 2019

[Paradigma19] Paradigma Digital. “React Native”. https://www.paradigmadigital.com/dev/desarrollando-aplicaciones-moviles-nativas-con-react-native/. 2019

[PCMagARvsVR19] PCMag. “AR vs VR”. https://www.pcmag.com/article/347242/ar-vs-vr-whats-the-difference. 2019

[PCMagEvolution19] PCMag. “Evolution of AR”. https://www.pwc.be/en/news-publications/insights/2017/the-evolution-of-augmented-reality.html. 2019

[ReactNative19] React Native documentation. https://facebook.github.io/react-native/. 2019

[ReactNavigation19] React Navigation. https://reactnavigation.org. 2019

[ReactRedux19] React Redux. https://es.redux.js.org/docs/basico/uso-con-react.html. 2019

[Symfony19] Symfony documentation. https://symfony.com/doc/3.4. 2019

[ViroReact19] ViroReact. https://viromedia.com/viroreact/. 2019

[ViroReactDocs19] ViroReact Docs. https://docs.viromedia.com. 2019

[WikipediaAR19] Wikipedia. "Augmented Reality". https://en.wikipedia.org/wiki/Augmented_reality. 2019

[WikipediaAWS19] Wikipedia. "Amazon Web Services". https://es.wikipedia.org/wiki/Amazon_Web_Services. 2019

[WikipediaFieldTrip19] Wikipedia. "Field Trip". https://en.wikipedia.org/wiki/Field_trip. 2019

[WikipediaGithub19] Wikipedia. "GitHub". https://en.wikipedia.org/wiki/GitHub. 2019

[WikipediaJetbrains19] Wikipedia. "JetBrains". https://en.wikipedia.org/wiki/JetBrains. 2019

[WikipediaMongoDB19] Wikipedia. "MongoDB". https://en.wikipedia.org/wiki/MongoDB. 2019

[WikipediaMySQL19] Wikipedia. "MySQL". https://en.wikipedia.org/wiki/MySQL. 2019

[WikipediaNodeJS19] Wikipedia. "NodeJS". https://en.wikipedia.org/wiki/Node.js. 2019

[WikipediaReact19] Wikipedia. "React". https://en.wikipedia.org/wiki/React_(JavaScript_library). 2019

[WikipediaSymfony19] Wikipedia. "Symfony". https://en.wikipedia.org/wiki/Symfony. 2010

13. Appendices

13.1. Diagram creation tools

In order to create the different diagrams used in this document, like use case diagrams, sequence diagrams or the ER diagram, three different tools were used depending on the type of diagram to be developed. These are the three tools used:

- UMLet: it is a free, open-source UML tool with a very simple user interface. It is used to develop all kind of UML diagrams, as well as use case or state diagrams. It gives the user the ability to export the diagrams in different formats, even to share them using Eclipse. During the developing of this project, it was used in order to create the different use case diagrams.
- Draw.io: It is an online UML diagram tool which gives a better and more usable experience than UMLet, but the functionality is mainly the same. It was used in order to create the sequence and deploy diagrams of Ariane.
- LucidChart: This online tool offers more types of diagrams than UMLet and Draw.io, including not UML diagrams like structure diagrams or organization charts. This tool was used in order to model the ER diagram of Ariane's database.