

# Museum Development Project Report

- Tech4CH -

**Advanced Topics in Computer Science**

Version 1.0 (Pre-Alpha / Pre-MVP)

**Alejandro Mallón Buitrago  
(Spanish Erasmus Student)**  
Roma Tre University / Universidad de Alcalá  
Madrid, Spain  
[ale.mallonbuitrago@stud.uniroma3.it](mailto:ale.mallonbuitrago@stud.uniroma3.it)

# **Abstract**

The development of software applied to museums is a considerable field to explore, since the conclusions and results of the analyses carried out with algorithms, provide us with the basis for the solutions that may be proposed in the future to improve the tours and exhibitions in museums in person. It is also an area to explore and explore the purpose of wanting to advance along the path of virtual reality for the simulation of visits in museums and collect more data in less time, by having people visiting the museum in person (physically) and virtually, as this area is booming and people need to explore more knowledge every day, and have these virtual visits at hand to provide them with the knowledge and concerns they have. Not only the physical and virtual reality improvement, but also the physical-virtual reality hybrid visit is worth mentioning, as it increases the positive experience of the visitors and consolidates the knowledge acquired. With these three aspects we can gather enough information in less time and completely improve the experience of the individuals, as well as test interaction experiments with visitors during their visit through these technologies, as they allow it in a closer way. Finally, to start a project it is necessary to take it step by step, getting to know the visitors of your own museum and implementing the technology gradually.

**Keywords:** *PHP, project development, Arquitectura de sistemas, Arquitectura de Software, PlayBack, SVG, Benchmark, BBDD, MVC, Sketch, Hosting, JS, JavaScript, JQuery, CSS, HTML, GitFlow, GitHub.*

# **Index**

<b>Introduction</b>	<b>3</b>
<b>Description of the project</b>	<b>3</b>
Context	3
Project objectives	3
Requirements	4
<b>Methodological description</b>	<b>5</b>
Working methodologies applied	6
Applied development methodologies	6
<b>Benchmark and Analysis</b>	<b>6</b>
Benchmarking of websites with similar functions and objectives	7
Analysis of playback technologies	7
<b>Technical description</b>	<b>7</b>
Programming language and paradigm	7
Software architecture	8
System architecture	8
Database architecture	9
Frameworks and libraries used	9
Technologies and tools used	10
<b>Software development</b>	<b>10</b>
Implemented classes	10
Implemented views	10
<b>Tests</b>	<b>12</b>
Unit testing	12
Integrated testing	12
Testing frameworks	12
<b>Implementation</b>	<b>13</b>
Training and manual	13
Hosting	13
<b>Maintenance</b>	<b>13</b>
<b>Conclusions</b>	<b>14</b>
<b>Bibliography</b>	<b>14</b>

## **Introduction**

The following document will explain the implementation of a web development project with different technologies, starting from some requirements and previous data. Therefore, we will start by talking about information prior to the purpose of development, previous analysis of competences, technologies and sketch. We will continue by explaining what repository and development methodology is planned and being implemented, the initial and pre-alpha status of the project, together with the next objectives to be met. Consequently, the chosen technology, benefits and proposed tests are explained. And finally, the implementation of the project, maintenance and conclusions are explained.

All this information is presented as a project to be further developed, since the current state of the project is at a very early stage and it is necessary to do much more work on the analysis of requirements, sketch, user experience, analysis of the best technologies, deepening the understanding of specific users and users who use the technology to be developed (archetypes) and to know many more fundamental aspects for the proper and progressive implementation of the project, to achieve a high level product adapted to the market and giving accurate and precise data/solutions to what is required.

The proposed solutions and objectives are ideals and results of analysis carried out in a short period of time, but may vary as the project progresses.

## **1. Description of the project**

For this project we were given a document of functional and technological requirements, on which we had to base the project to be developed and implemented.

### **1.1. Context**

This assignment has been requested as a university project, by Professor Tsvi Kuflik, for the subject of advanced topics in computer science, at the University of Roma Tre in Rome, Italy.

For this purpose, we have been given a series of consecutive lessons for a few weeks corresponding to the above-mentioned subject and subsequently we have been asked for this assignment.

As functional and technical requirements we were asked for several things, but as a main requirement we have been asked to develop a web-app software that addresses different analytical needs for a museum, in all its aspects, related to visits and tours: "You are requested to develop a Web-based system that given a museum floor plan and a set of visit logs of groups of different sizes". This web application, in general terms, is required to be able to animate visitor visits on a floor plan and to contain different analytics on these visits. Considering the scope of the project, the focus has been put on the detailed analysis of the requirements in order to choose the best solution and to propose interesting ideas.

### **1.2. Project objectives**

The aim of the project is to be able to meet all the requirements of the project by the deadline. But taking into account the factors of feasibility, time and size: combining and analysing them we have come to the conclusion to diversify the parts of the project and focus on the most important parts, which are to make a good base of analysis of requirements and technologies to be applied and implemented in the development of the project, as well as to

analyse and think functionally all parts of the project to make it as interactive and contain the best user experience. Even so, a minimal version of the project has been made as a mock-up and with small detailed functionalities such as the playback, as it can be easily used due to its advanced state.

The proposed objectives, following a review of the three factors mentioned above, are as follows:

- Benchmark similar web-apps with similar functional requirements to draw ideas and conclusions, analyse them and come up with a quality solution.
- Perform an analysis of technologies to get the best of them and get an optimisation on the playback at a high level.
- Prototyping/sketching and analysing the user experience for all views and functionalities.
- Prepare a web layout development with the main functionalities well defined, in short, prepare an MVP to be able to present it as the basis of the project. This would be a Pre-Alpha version.

The next goals to be achieved would be:

- To make the database with the users and improve it for use within the project.
- Analyse and improve the technology used.
- Implement more functionalities.
- Implement more analysis cases on individual and collective routes.
- Implement more global analysis cases in the analytics section.
- Centralise screen navigation.
- Improve the MVC model.
- With the "playback" technology already implemented, map the museum floor plan by coordinates, to create a trajectory algorithm according to the data read from each visitor's visit case.

In addition to these points, the methodological objectives of testing, implementation, versioning and hotfixing, for the following phases and versions developed, should be highlighted.

### 1.3. Requirements

The requirements we have been asked to meet by the teaching staff are as follows:

- **Functional requirements:** "*To analyze the logs and present information to the museum curator and create a "playback" system that will allow re-constructing of a museum visit using the time-tagged data in the log.*" We are required to provide a simulation of visits to the museum with an analytical detail of the visit, both individual and group, as well as a separate section on general analysis of all data with conclusions on everything to do with visits to the museum.
- **Technical requirements:** "*Consider following the Model-View-Controller (MVC) design/architectural pattern. Language/development environment – whatever you like.*" We are required to use any type of technology and programming language, with an MVC model and an associated relational database.

- **Additional requirements:** “I need a link to get a project report + a web-based system (for playback) + a user guide (better to have online help).” In addition, we are asked for a technical document (project report) to present the technological details and methodologies applied, as well as a manual on the web-app and the hosting of the website itself.

The details, based on what you want the web-app to contain, are as follows:

1. “Play back” a visit of a selected group (or individual) – see how they moved around in the museum between exhibits, when it is possible to speed up the visit.
2. Create a visit summary for a selected group/individual.
  - a. Where they visited.
  - b. How much time they spent.
  - c. What presentations they watched.
  - d. Did they enjoy:
    - i. Stayed more than the average group.
    - ii. Watched presentations more than average.
    - iii. Rated presentation higher than the average.
    - iv. Did not interrupt the presentations.
3. To present (visualize) statistics about the museum visit:
  - a. Analysis of how many visitors per hour in general
  - b. How many visitors per room per hour?
  - c. Attraction power (how many/proportion of visitors to the site visited a specific POI?) and holding power (how much time/relative time did visitors spend at a POI once they got there) of exhibits.

To this end, we have been given a series of documents, logs, mass data, images and other data of interest:

- a. A museum floor plan with names of POIs (enlarge to see) + an excel file with some corrections.
- b. Visit logs – timestamped positions visited, timestamped presentations viewed with an indication if they ended normally (by the system) or terminated by the user and a list of time stamped events (less relevant).
- c. An excel file that associates the visitor number (“A” column, log file number) with the visitor’s group (“B” column) other columns are not relevant.

Finally, as a summary of the requested requirements, the following are specified:

- a. Create a database of individuals, their time-stamped events, the groups they belong to.
- b. Enable to “replay a visit”.
- c. Create a visit summary for a group or an individual.
- d. Enable the presentation of visitors’ behavior statistics (as noted above and any creative idea you may have).
- e. Write a summary report.

## 2. Methodological description

This section explains the methodologies, both work and development, applied or that should be implemented for the development of the project, since they are an obvious way of carrying out, in a common way (as a team), any activity that requires collective objectives.

## **2.1. Working methodologies applied**

The methodologies that have been applied and are to be implemented throughout the project are, for the pre-development part, to carry out consecutive development and new requirements or clarification dailies. For the design part, always carry out the prototyping and sketch phase, taking into account the user experience and functionalities already developed in order to carry out everything according to the lines already in use.

Throughout the development of the project, the documentation phases for Functional Designs and Technical Designs will be implemented.

A task management will be necessary, which is linked to the next point, as the functionalities are landed in development resources landed in Trello cards.

A constant review of technologies will be carried out, for optimisation and evolution, to avoid depreciation and security flaws.

Finally, versioning guidelines will be taken into account for each deployment version, both in UAT, pre-production and production. Following the basic guidelines for deployment and sharing with testing users and end users.

## **2.2. Applied development methodologies**

The development methodologies that have been implemented and that we want to continue using are more technical, at the level of development, version control in repositories and interactive development tasks.

For this, we will take into account the leading technologies, such as git, in order to have an exhaustive and professional branch control, working in this case with Github, but being able to be evolved or migrated to any technology. But always with the base of Git. In addition to using the technology, since it obviously helps to have a controlled methodology, we will use the GitFlow branch control methodology, this way we know that anyone who enters new will know how to work in the repository. Since GitFlow is a standard in code control methodologies.

Finally, for a methodological control of the core development, the Scrum methodology will be used, with dailys, task control by cards, reviews, requirements, spring plannings and retrospectives.

All these mentioned methodologies, some of them already implemented or with the vision to be implemented, help in a great way that a group/work team, perform all the tasks in an orderly, complementary and parallel way or without any kind of stumbling or disorientation, among others.

## **3. Benchmark and Analysis**

This section explains two very important phases carried out for this project and any project that aims to compete functionally or technologically in the market and wants to provide an effective and efficient final solution.

### **3.1. Benchmarking of websites with similar functions and objectives**

In this phase of the project, a Benchmark has been carried out, this means that all the websites / web-apps that have been found similar to the functionalities that we want to implement have been analysed on the internet. This has been done with the purpose of extracting the best ideas already implemented in the market and analyse them deeply, for their subsequent collection, taking them out of context and landing a solution and optimal results. Thus being competent and avoiding reinventing what others have already analysed and implemented a long time ago with solid foundations.

Thanks to this analysis, we have been able to draw technological, functional and experiential ideas for the entire website and we have drawn conclusions about the most important and least important things to implement in our web-app.

### **3.2. Analysis of playback technologies**

By carrying out an exhaustive analysis, it has been possible to identify and find different libraries to be able to create animations on plans. These technologies have been discarded due to the complexity of implementation, de-versioning, outdated and dubiousness. In addition to this, no specific solutions have been found to the problems posed, so it has been decided to use an extended, conventional, well-known and reusable animation system.

→ **Scalable Vector Graphic:**

- ◆ Markup Language.
- ◆ Independent resolution.
- ◆ Support for all modern browsers.
- ◆ A w3c standard.
- ◆ Easy to create and edit.
- ◆ We can manipulate it with CSS and JS.

Thanks to this technology, we can create representative objects of people or groups of people, which help us to simulate their visits on the museum floor plan.

In order to create animations using this technology, multiple examples of implementation have been reviewed, until the most feasible complementary technology for animating these SVG compositions has been found.

## **4. Technical description**

In this section, all the technical parts used, based on and implemented in the project are detailed, with the aim of making everything used known in detail so that the project can evolve in the hands of anyone.

### **4.1. Programming language and paradigm**

The language used, initially, has been a combination of different languages and mark-up or tagging languages:

- PHP
- HTML
- CSS
- SCSS
- JavaScript

## → JQuery

The decision to use this main programming language, PHP from the server side, has been made because it is easy to learn, implement and collaborate with the other technologies that accompany it. In addition to the fact that actions can be carried out from visual to databases.

It is worth highlighting the programming paradigms it knows, as it is Multiparadigm: imperative, functional, object-oriented, procedural, reflexive. This has helped us a lot to implement the required software architecture, MVC.

On the other hand, the implementation of JavaScript and JQuery from the front end has helped us to implement the required visual functionalities more easily such as playback. This is an interpreted programming language, a dialect of the ECMAScript standard. It is defined as object-oriented, prototype-based, imperative, weakly typed and dynamic. So, being of this kind of paradigm and close to object-oriented, it has helped us in its implementation/combination with PHP and between the tagged ones.

### **4.2. Software architecture**

For the software architecture (set of coherent patterns and abstractions that provide a clear and defined framework for interacting with the software source code), the one specified in the requirements has been implemented (Model View Controller View (MVC) is a style of software architecture that separates an application's data, user interface, and control logic into three distinct components. The View, or user interface, which composes the information that is sent to the client and the mechanisms for interaction with the client). There have been no implementation problems since it is fully extended and standardised. Thanks to the languages chosen with their accompanying paradigms, it has been possible to implement it in an optimal way and with a view to its continued use and extension.

The use of MVC has been accepted, and an improved proposal has not been made, because it is a proven software design pattern and is known to work. With MVC the application can be developed quickly, in a modular and maintainable way. Separating the application functions into models, views and controllers makes the application very lightweight.

The Model-View-Controller (MVC) paradigm is a design pattern that separates the code into three layers. Using a design pattern is often highly recommended, as it is a way to standardise our code, optimise it and make it more readable.

Within the project, the Model, View and Controller implementation can be seen clearly and distinctly separated by individual folders and interconnected with each other. In addition, it is worth highlighting a proposed extension for navigation and knowledge between MVC classes, which is the figure of the Rooter, this implementation has not yet been done, but it would be advisable that the next steps are made.

### **4.3. System architecture**

For the architecture of the system used during development, various plugins have been implemented, in addition to using XAMPP to set up a server to provide PHP support for integration and unit testing. In addition to using, at the time of hosting, an architecture provided by a third party provider.

#### **4.4. Database architecture**

For the database architecture, it is proposed to continue using MySQL, due to its easy implementation with the proposed languages and technologies.

MySQL is a database management system with a dual license. It is currently the most famous and widely used open source database in the world. Like it, we can find others such as Oracle itself or Microsoft SQL Server, which may be proposals for future migration if at some point the amount of data we collect exceeds the limits of MySQL technology.

The creation of the database should be done through an already implemented file and all queries should be made directly to it. As in the previous point, a third party provider is used for the architecture and hosting of the database.

#### **4.5. Frameworks and libraries used**

For this project, different style and animation libraries have been used, as well as JavaScript logic libraries. This is due to the requirements for the reproduction and simulation of museum visits and the required analytics graphics or SVG animation. In addition to styles, fonts and icons for quick implementation of reused and already formatted styles. Among the frameworks and libraries have been implemented:

- nucleo-icons
- nucleo-svg
- soft-ui-dashboard
- soft-ui-dashboard.css.map
- soft-ui-dashboard.min
- nucleo.eot
- nucleo
- nucleo.woff
- nucleo.woff2
- nucleo-icons.eot
- nucleo-icons
- nucleo-icons
- nucleo-icons.woff
- nucleo-icons.woff2
- bootstrap.min
- popper.min
- bootstrap-notify
- Chart.extension
- chartjs.min
- perfect-scrollbar.min
- smooth-scrollbar.min
- soft-ui-dashboard
- soft-ui-dashboard.js.map
- soft-ui-dashboard.min
- soft-ui-dashboard.scss

All these libraries are not in use in their entirety, but they have been implemented with the aim of making use of them for the wide range of resources they provide and the centralisation of resources needed for the future of the project.

Other libraries not literally indicated, have been implemented directly in the classes through universal urls and collected as APIS. For example, Bootstrap, JQuery, Fonts Google, SCSS or JavaScript Libraries, among others.

#### **4.6. Technologies and tools used**

The technologies used are, above all, the implementation of resources and libraries mentioned above for the management and animation of the required functionalities.

Visual Studio Code was used to develop the software, XAMPP was used to set up the server that provides the web visualisation, SourceTree was used to manage the repository, and MySQL Workbench was used for the MySQL database. Trello was used for methodological management.

In addition, the internet was used to search for resources, more specifically StackOverflow and Codepen.

For hosting, a third party provider has been used: 000webhost.

### **5. Software development**

This section specifies the classes and views of the MVC model that have been implemented so far, in order to follow the same steps or to better recognise the order of the project and to give an orientation of the project so that anyone can find any functionality or section of it. In addition to being able to collaborate more quickly. The main views of the project are also specified.

#### **5.1. Implemented classes**

The implemented classes are:

**Model:**

- modelo.php

**View:**

- Partials Folder
  - ◆ header.php
  - ◆ footer.php
- home.php
- playback.php
- analytics.php

**Controller:**

- controlador.php

#### **5.2. Implemented views**

The implemented graphical GUI views, initial and basic, with:

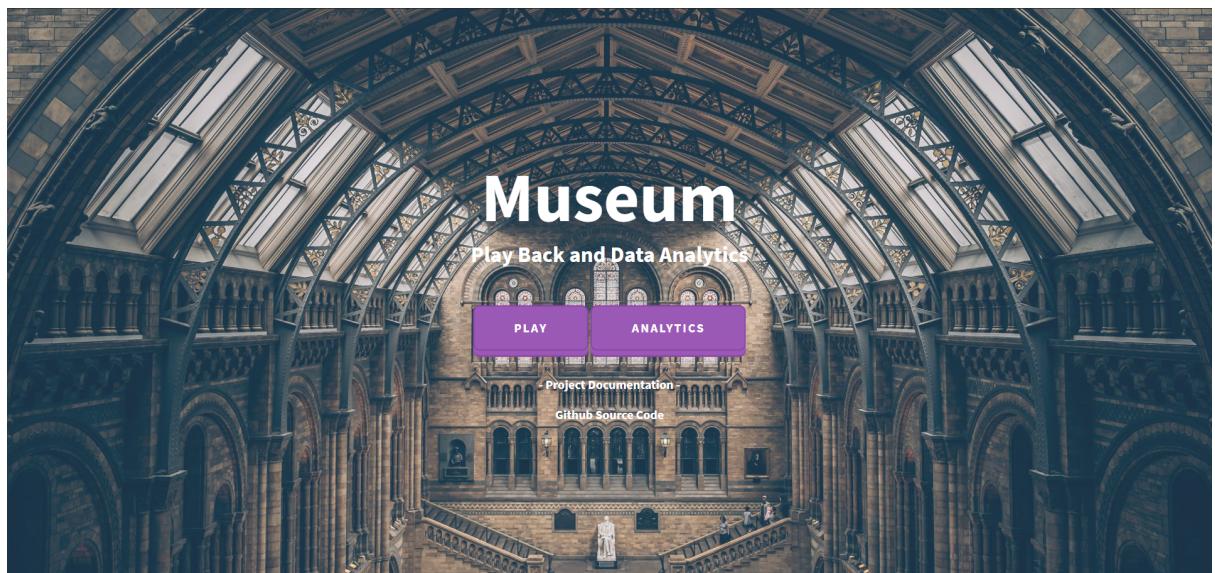
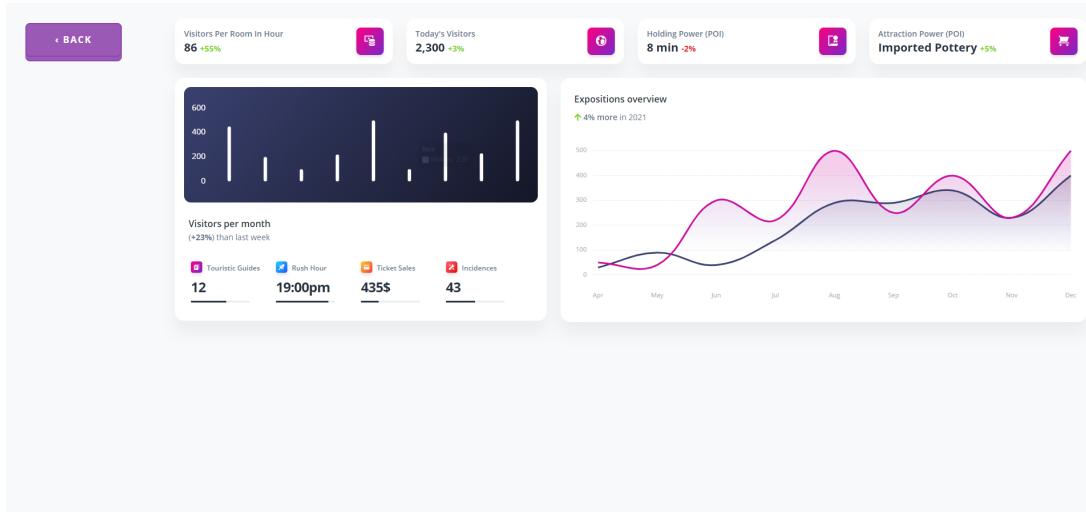


Image I. Index view. Own elaboration.

< BACK

Day	Visitor	Positions	Presentations	Events	Actions
1	visitor_195_146	14:12:05,14:12:19,JerusalemPhoto 14:12:19,14:13:23,MaterialCultures	14:12:19,14:13:23,MaterialCultures,137,System,5 14:14:11,14:14:47,MaterialCultures,165,User,5	14:12:14,chosenExhibit,44 14:12:14,return from screen	<b>PLAYBACK</b>
2	visitor_196_146	14:12:32,14:13:54,MaterialCultures	14:12:32,14:13:01,MaterialCultures,165,User	14:14:38,return from	

Image II. PlayBack View. Own elaboration.



**Image III. Analytics View. Own elaboration.**

## 6. Tests

This section details the different types of tests carried out and to be carried out during the course of the development of the project. It is necessary to comply with this phase, as it is essential to maintain quality and safety standards in order to ensure that the project is on track.

### 6.1. Unit testing

Unit tests are the testing of a single method of a class. The scope is very small and perfectly bounded. Any dependency of the module under test must be replaced by a mock, or a stub.

### 6.2. Integrated testing

Integration testing is the testing of interaction between two or more elements, which can be classes, modules, packages, subsystems, etc. including the interaction of the system with the production environment.

### 6.3. Testing frameworks

Testings are as important or more important than the development itself, because even if you have made a nice development, if it is not correct, if it does not work well, if it fails, if it has security holes, if it is not optimised, if it is not optimised towards the user experience, it can end up in disuse. For this there are a variety of frameworks that help automate this process and make it more efficient, which is why it has been considered and is strongly recommended to use and implement any of these frameworks:

#### → JavaScript:

- ◆ Mocha
- ◆ Fraudster
- ◆ Jest
- ◆ Jasmine
- ◆ Tape

→ **PHP:**

- ◆ PHPUnit
- ◆ Codeception
- ◆ Behat
- ◆ PHPSec
- ◆ Storyplayer

## 7. Implementation

This section explains the process of teaching users who use the platform, the platform's user manual and the hosting process.

### 7.1. Training and manual

For any development project, it is necessary to consider training/teaching of the system to any user who will interact with or use the system. This can be a manual on the interactive website itself or a physical training with classes for the system's administrator users.

For this reason, we have taken into account the creation of a specific manual for the web and what has been developed so far. Even if it is a Pre-Alpha / Pre-MVP version. Since it is being shown as a template / idea of what the project can be in the future and what paths it can take.

This manual can be found complementary to this technical document of development memory and also complementary to the developed web-app-model.

### 7.2. Hosting

For hosting, a third party system has been used, where free hosting is provided through a subdomain (000webhost.com). Simply by registering, choosing a domain name and uploading the files manually, the website is in use and available on the internet.

This website has been implemented:

- A basic Pre-Alpha & Pre-MVP prototype version of what is required.
- The download of these same documents: Technical memory document and user manual.
- Access to the repository to view and download the source code.

The url where the web-app is hosted is: <https://museum-playback.000webhostapp.com/>

## 8. Maintenance

In this section, reference is made to the importance of the maintenance process of the project throughout its development, implementation, acceptance, deployment and implementation of improvements with evolutions and hotfixes.

This is why the project has a long-term objective and must take into account both the implementation of new functionalities and the fixing of all the problems that arise. This can be managed in the same way with the methodologies mentioned above.

## **9. Conclusions**

As a conclusion of the work, I have been able to learn and to establish a lot of theoretical knowledge in practice. I was able to put into practice the work methodologies that I had been taught, the technologies and logic that I had learnt.

It is important to emphasise the importance of organisation for this type of project and that it must be carried out at a short pace in order to achieve the maximum quality in the development of the project.

I have been able to make a retrospective at the end of the work, knowing that the best thing to do is to make versions that are perhaps not so advanced in programming, but with an established theoretical base and deeply analysed. In this way, it is possible to draw the best and most reliable conclusions from the work done.

In addition to all this, I have learned new technologies and faced a requirement that has had me involved more than necessary: "playback". In order to be able to perform in the best way and to be able to break through the limitations of knowledge.

## **10. Bibliography**

Stack Overflow. (s. f.). *Development of specific and generic functionalities*. Retrieved 12 June 2021, from: <https://stackoverflow.com/>

Codepen. (s. f.). *Specific functionalities and animations*. Retreived 11 June 2021, from: <https://codepen.io/>

Advanced Topics in Computer Science. (2021, May). *Museum Development Project Report*. Tsvi Kuflik. University of Roma Tre.