



FACULTY OF ENGINEERING

Degree in Computer Engineering

## **Fantasy**

Course 2018-2019

Luis Gutiérrez Flores  
Nicolás Ruiz Requejo  
Jesús Rodríguez Heras  
Arantzazu Otal Alberro  
Alejandro Segovia Gallardo  
Alejandro José Caraballo García  
Gabriel Fernando Sánchez Reina

Puerto Real, May 18, 2019





FACULTY OF ENGINEERING

Degree in Computer Engineering

**Fantasy**

DEPARTMENT: Computer Engineering.

PROYECT DIRECTOR: Alicia Ale.

PROJECT AUTHOR: Team Fantasy.

Puerto Real, May 18, 2019

Signed: Team Fantasy



### **Summary**

Web application to promote learning through the imagination and creativity of children between 10 and 13 years old in scientific-technological subjects in collaboration with the European project STIMEY.

As a game, children can create interactive stories and teachers can evaluate them.

**Keywords:** Fantasy, learning, development, illusion, entertainment, creativity, questionnaire, evaluation, teaching, science, European Union.



# Contents

<b>I</b>	<b>Prolegomenon</b>	<b>11</b>
<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Motivation . . . . .	13
1.2	Description of the current system . . . . .	13
1.3	Objectives and scope of the project . . . . .	13
1.3.1	Objetives . . . . .	13
1.3.2	Scope . . . . .	13
1.4	Organization of the document . . . . .	14
<b>2</b>	<b>Planning</b>	<b>15</b>
2.1	Development methodology . . . . .	15
2.2	Project planning . . . . .	15
2.3	Organization . . . . .	16
2.3.1	Roles . . . . .	16
2.3.2	Hardware and software resources . . . . .	16
2.4	Costs . . . . .	16
2.4.1	Human costs . . . . .	16
2.4.2	Material costs . . . . .	17
2.5	Risk management . . . . .	17
2.6	Team policy . . . . .	17
2.7	Hits . . . . .	17
2.7.1	Sprint 0 (27 <sup>th</sup> February to 6 <sup>th</sup> March) . . . . .	17
2.7.2	Sprint 1 (6 <sup>th</sup> March to 27 <sup>th</sup> March) . . . . .	17
2.7.3	Sprint 2 (27 <sup>th</sup> March to 3 <sup>rd</sup> April) . . . . .	17

2.7.4	Sprint 3 (3 <sup>rd</sup> April to 10 <sup>th</sup> April)	17
2.7.5	Sprint 4 (10 <sup>th</sup> April to 24 <sup>th</sup> April)	18
2.7.6	Sprint 5 (24 <sup>th</sup> April to 1 <sup>st</sup> May)	18
2.8	Meetings	18
2.8.1	Meeting 0 (27 <sup>th</sup> February)	18
2.8.2	Meeting 1 (6 <sup>th</sup> March)	18
2.8.3	Meeting 2 (27 <sup>th</sup> March)	18
2.8.4	Meeting 3 (3 <sup>rd</sup> April)	18
2.8.5	Meeting 4 (10 <sup>th</sup> April)	18
2.8.6	Meeting 5 (24 <sup>th</sup> April)	18
<b>II</b>	<b>Developing</b>	<b>19</b>
<b>3</b>	<b>Analisis of requirements</b>	<b>21</b>
3.1	Workspace	21
3.1.1	Teachers	21
3.1.2	Students	21
3.2	Characteristics of active points	22
3.3	Characteristics of fantasies	22
<b>4</b>	<b>System design</b>	<b>25</b>
4.1	Design of use cases	25
4.1.1	CRUD fantasy	25
4.1.2	Use case: Choose language	28
4.1.3	Use case: Copy fantasy	28
4.1.4	CRUD background	29
4.1.5	CRUD active point	29
4.1.6	CRUD image	31
4.1.7	CRUD video	33
4.1.8	CRUD text	35
4.1.9	CRUD quiz	36
4.1.10	CRUD audio effect	38
4.1.11	CRUD additional information	40
4.1.12	Use case: Organize active points	41
4.1.13	CRUD final quiz	41
4.2	User interface desing	44
4.3	Data design	44
4.4	Design of components	44
4.5	Parameterization of the base software	44



<b>5</b>	<b>Implementation of the system</b>	<b>45</b>
5.1	Technological environment . . . . .	45
5.2	Source code . . . . .	45
5.3	Code quality . . . . .	45
<b>6</b>	<b>System tests</b>	<b>47</b>
6.1	Unit tests . . . . .	47
6.2	Integration testing . . . . .	47
6.3	System tests . . . . .	47
6.3.1	Functional testing . . . . .	47
6.3.2	Non-functional tests . . . . .	47
6.4	Acceptance Tests . . . . .	47
<b>III</b>	<b>Epilogue</b>	<b>49</b>
<b>7</b>	<b>User manual</b>	<b>51</b>
7.1	Introduction . . . . .	51
7.2	Features . . . . .	51
7.3	Previous requirements . . . . .	51
7.4	Utilization . . . . .	51
<b>8</b>	<b>Installation and operation manual</b>	<b>53</b>
8.1	Introduction . . . . .	53
8.2	Previous requirements . . . . .	53
8.3	Inventory of components . . . . .	53
8.4	Installation procedures . . . . .	53
8.5	Operating procedures and service level . . . . .	53
8.6	Implantation tests . . . . .	53
<b>9</b>	<b>Conclusions</b>	<b>55</b>
9.1	Objetives . . . . .	55
9.2	Learned lessons . . . . .	55
9.3	Future work . . . . .	55



## **Part I**

# **Prolegomenon**



# Chapter 1

## Introduction

### 1.1 Motivation

It is a work of the subject “ Computer Projects ” which, at a professional level, helps us to gain work experience and face real situations facing a demanding clientele.

### 1.2 Description of the current system

Initially, our client had an application that showed information about a topic on a page and the students did not focus on learning, but went directly to the final questionnaire in order to finish earlier. This means that students do not learn properly or encourage their imagination or creativity.

### 1.3 Objectives and scope of the project

#### 1.3.1 Objectives

Motivation of creativity and promotion of imagination in children.

To fulfill the general objective, we will have to cover the following points:

- Interactive learning resources.
- Can be evaluated by a teacher.
- You can share fantasies between users.
- It is simple and manageable by primary school students.
- Foster STEM skills and teachings (science, technology, engineering and maths).

#### 1.3.2 Scope

The students will be able to create fantasies, share them and they will be able to be evaluated by the professors, who will be able to send as a task the making of fantasies.

## **1.4 Organization of the document**

This document is organized according to the specifications set out for the presentation of an end-of-degree project following the following sections:

1. Introduction.
2. Project plan.
3. Analysis of requirements.
4. System design.
5. Implementation of the system.
6. System tests.
7. User manual.
8. Installation manual.
9. Conclusions.

## Chapter 2

# Planning

This chapter includes the planning, the approach and the principle of a project that we have named “**Fantasy**”, a web portal where teachers can perform a series of tasks (fantasies) with the objective that students can play and learn in a creative way.

The students will also have the possibility of creating the fantasies that the teacher sends them as work and then they will be evaluated by said professor.

### 2.1 Development methodology

The methodology used will be textbf Scrum: Agile development method characterized by having an incremental development and basing the quality of the result on knowledge rather than on the processes used.

### 2.2 Project planning

The project will last for three months and weekly meetings with the client will be held for a maximum of one hour.

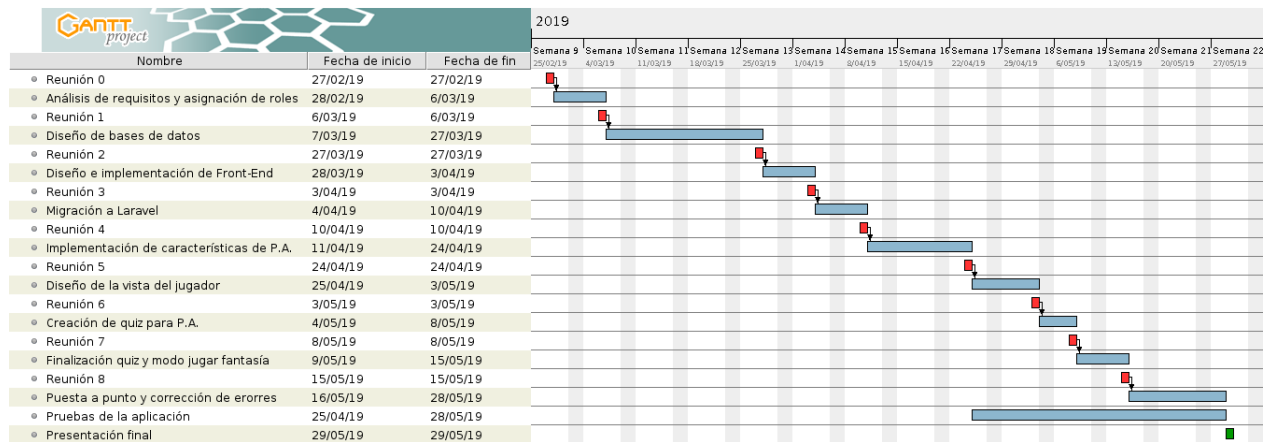


Figure 2.1: Gantt diagram

## 2.3 Organization

### 2.3.1 Roles

- **Administrator:** Luis Gutiérrez Flores.
- **Analysts:** Jesús Rodríguez Heras and Nicolás Ruiz Requejo.
- **Designers:** Arantzazu Otal Alberro, Gabriel Fernando Sánchez Reina and Nicolás Ruiz Requejo.
- **Developers:** Luis Gutiérrez Flores, Alejandro Segovia Gallardo and Alejandro José Caraballo García.
- **Test Engineers:** Jesús Rodríguez Heras and Luis Gutiérrez Flores.

### 2.3.2 Hardware and software resources

As hardware resources we have the laptops of the 7 members of the group and the Stimey server.

As software resources we have the framework Laravel, Atom, Visual Studio Code, TeXStudio, PhpMyAdmin, MySQL, GitHub.

## 2.4 Costs

### 2.4.1 Human costs

- Hours in the learning of Laravel.
- PHP and MySQL training hours.
- GitHub training hours.
- Documentation hours.



## 2.4.2 Material costs

- Our computers.
- Transportation to school.
- Stimey server expenses.

## 2.5 Risk management

- Do not meet deadlines for trying to cover too much and leaving incomplete functionalities.

## 2.6 Team policy

The team has decided to hold weekly meetings with the client, throughout the week, the team members will try to establish meetings between them with the necessary duration to continue advancing in the project (estimated time: two hours).

## 2.7 Hits

### 2.7.1 Sprint 0 (27<sup>th</sup> February to 6<sup>th</sup> March)

1. Creation of work platforms and version control (GitHub).
2. Creating the requirements sketch.
3. Creation of use cases and their descriptions.

### 2.7.2 Sprint 1 (6<sup>th</sup> March to 27<sup>th</sup> March)

1. Creation of platform mockups.
2. Implementation of the database.

### 2.7.3 Sprint 2 (27<sup>th</sup> March to 3<sup>rd</sup> April)

1. Implementation of front-end.
2. Migrations of the database.
3. Adaptation of the project to the Laravel framework.
4. Creation of fantasies.

### 2.7.4 Sprint 3 (3<sup>rd</sup> April to 10<sup>th</sup> April)

1. End of front-end.
2. Final migrations of the database.
3. Completion of fantasy creation.

### **2.7.5 Sprint 4 (10<sup>th</sup> April to 24<sup>th</sup> April)**

1. End of database migrations.
2. Creation of active points with their basic characteristics.

### **2.7.6 Sprint 5 (24<sup>th</sup> April to 1<sup>st</sup> May)**

1. Creation of active points with all their characteristics.
2. Start of the view to be able to play the fantasy.

## **2.8 Meetings**

### **2.8.1 Meeting 0 (27<sup>th</sup> February)**

1. Analysis of system requirements.
2. Creation of use cases.
3. Approach of the system database.
4. Distribution of sprint tasks 0.

### **2.8.2 Meeting 1 (6<sup>th</sup> March)**

1. End of sprint 0.
2. Distribution of information to search.
3. Start of sprint 1.

### **2.8.3 Meeting 2 (27<sup>th</sup> March)**

1. End of sprint 1.
2. Start of sprint 2.

### **2.8.4 Meeting 3 (3<sup>rd</sup> April)**

1. End of sprint 2.
2. Start of sprint 3.

### **2.8.5 Meeting 4 (10<sup>th</sup> April)**

1. End of sprint 3.
2. Start of sprint 4.

### **2.8.6 Meeting 5 (24<sup>th</sup> April)**

1. End of sprint 4.
2. Start of sprint 5.

## **Part II**

# **Developing**



## Chapter 3

# Analisis of requirements

### 3.1 Workspace

The graphic design used should be the same as the page on [Stimey](#) on all the icons used.

We can highlight two user roles: teachers and students.

The teacher will have more permissions and privileges than the student, so that he can create fantasies to evaluate his students and they will have to complete them or create the ones that the teacher puts them as work.

In the workspace we will have a series of options that will be available for both the teacher and student roles depending on the permissions of each role:

#### 3.1.1 Teachers

- **Background:** Opens a window where you can select an image from the Internet, the computer or an image previously used. This image will cover all the workspace.
- **Active Point:** They can establish active points in the workspace and modify them conveniently by adding images, text, video, audio, etc. They can also establish a score for each active point of the fantasy to evaluate the students.

#### 3.1.2 Students

- **Background:** You can do the same as the teachers.
- **Active Point:** They can establish active points in the workspace and modify them conveniently by adding images, text, video, audio, etc. You can not set a score for active points.

## 3.2 Characteristics of active points

- It will be possible to move it within the background and modify its contents.
- If an image is added to the active point, that point is adapted to the shape of the image.
- You can also assign a video, which will open a window to play it, or an audio. In case there is no audio or video, the respective button will not be displayed.
- Active points may have background music that will be muted if the audio or video playback assigned to that point by the faculty begins. The music will be restored when the corresponding audio or video ends.
- The active points can be reorganized by the teachers and students so that they emerge in the desired order.
- Through the realization (and not creation) of a fantasy, a student can not continue with the next active point without finishing the current one.
- The active points quiz should be fun and intuitive.
- The questions raised in the quiz of the active points should be 2 and not too difficult (multiple answer, write a word, quiz with images and questions about it, join items).
- The quiz of the active point will appear on the screen when that active point is closed.
- Once the quiz is finished, the next active point appears in the order established by the teachers/students in the workspace.
- Each active point will have a score to add (to all) a maximum of 100 points.
- When assigning a score to an active point, it will be subtracted from the total we carry (maximum 100 points). If an active point is eliminated, the general counter recovers the score assigned to that active point.
- The student does not know the total number of active points in the fantasy.
- When the student obtains a score when completing an active point, this amount is added to the global counter.
- Finally, we can have a statistical summary with the right/failed questions of each active point.
- Only the score obtained will be saved the first time a quiz is done, then it can be done more times, but the note will not be recorded in the system.
- The student will have the option to save his progress with a save button manually or through the auto-save option.

## 3.3 Characteristics of fantasies

- At the end of all active points there will be a button at the bottom right of “**more information**” and in the center a new quiz that will be the final exam. This exam will have an independent score to all active points and will not have the statistical summary. If this quiz is repeated, the note would be updated with a percentage of the new grade, plus the previous note with the objective that a student who repeats a quiz can not get the best grade per repetition of it.
- The teachers can send the students to make fantasies to learn as homework. These tasks can be done in groups of students based on two ideas:
  1. **Compulsory:** A student performs fantasy and the rest seeks additional information.
  2. **Optional but ideal:** Concurrent fantasy edition among all the members of the group.
- Each fantasy will have a code to be shared.

- We will have two types of permissions in the fantasies: “**see**” and “**see and edit**”.
- The platform will notify the faculty when the students have finished their respective jobs.
- The fantasies may be private, shared or public. By default, they will always be public and can be accessed by everyone who uses the platform.
- Shared fantasies can be accessed by other people with a password.
- Fantasies can be cloned.





# Chapter 4

## System design

### 4.1 Design of use cases

All the use cases described below have the following implicit precondition to be able to use said use cases in the final application:

- The user (teachers/students) must have an account on the Stimey platform and have logged in with that account.

The proposed use cases are the following:

#### 4.1.1 CRUD fantasy

##### Use case: Create fantasy

- **Description:** Create a new fantasy.
- **Actors:** Creator-editor (user).
- **Preconditions:** The user must have permission to create a new fantasy.
- **Postconditions:** Fantasy is stored in the system.
- **Main stage:**
  1. The user selects the option “Create new fantasy”.
  2. The system requests the name of the fantasy.
  3. The user enters the name of the fantasy.
  4. The system requests the fantasy code.
  5. The user enters the fantasy code.
  6. The system gives to choose if the fantasy will be public (by default), shared or private.
  7. The user selects “ Public ”.
  8. The user creates the fantasy.
  9. The fantasy is stored in the system.

- **Extensions:**

- 7. a) The user selects that the fantasy will be shared.

- 1. The system allows to insert in a list the identifiers of other users with which the fantasy will be shared.
    - 2. The user enters the identifiers of the users who will share the fantasy.
    - 3. Step 8.

- 7. b) The user selects that the fantasy will be private.

- 1. The system marks the fantasy as private for that user without giving the possibility of sharing.
    - 2. Step 8.

- \*a) At any time, the user can go back to the main menu.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Visualize fantasy

- **Description:** Read an existing fantasy.

- **Actors:** Creator-editor (user).

- **Preconditions:** The fantasy must exist in the system and the user must have modification permissions.

- **Postconditions:** There are no changes in the fantasy.

- **Main stage:**

- 1. The user selects the option “ My fantasies ”
  - 2. The system displays a list of fantasies accessible by the user.
  - 3. The user selects the fantasy that he wants to visualize.
  - 4. The system displays a pop-up window with the fantasy information and its options.
  - 5. The user selects the option “ Visualize fantasy ”.
  - 6. The system shows the fantasy.
  - 7. The user reads the fantasy without making any changes and, when it is over, closes the fantasy.
  - 8. The fantasy remains unchanged.

- **Extensions:**

- \*a) At any time, the user can go back to the main menu.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

**Use case: Update fantasy**

- **Description:** Modify an existing fantasy.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist in the system and the user must have modification permissions.
- **Postconditions:** The fantasy is modified.
- **Main stage:**
  1. The user selects the option “My fantasies”.
  2. The system displays a list of fantasies accessible by the user.
  3. The user selects the fantasy that he wants to modify.
  4. The system displays a pop-up window with the fantasy information and its options.
  5. The user selects the option “Modify fantasy”.
  6. The system displays the fantasy creation screen for modification.
- **Extensions:**
  - \*a) At any time, the user can go back to the main menu.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Delete fantasía**

- **Description:** Erase an existing fantasy.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist in the system and the user must have removal permissions.
- **Postconditions:** Fantasy is eliminated from the system.
- **Main stage:**
  1. The user selects the option “My fantasies”.
  2. The system displays a list of fantasies accessible by the user.
  3. The user selects the fantasy that he wants to modify.
  4. The system displays a pop-up window with the fantasy information and its options.
  5. The user selects the option “Clear fantasy”.
  6. The system displays a confirmation message.
  7. The user selects “Accept”.
  8. The system erases the fantasy.
- **Extensions:**
  7. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.
  - \*a) At any time, the user can go back to the main menu.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

### 4.1.2 Use case: Choose language

- **Description:** Change the language of the application.
- **Actors:** Teacher or student (user).
- **Preconditions:** None.
- **Postconditions:** The application changes to the language selected by the user.
- **Main stage:**
  1. The user presses the language change button.
  2. The system displays a list of available languages.
  3. The user selects a language from those that are available in the system.
  4. The application changes the language.
- **Extensions:** None.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

### 4.1.3 Use case: Copy fantasy

- **Description:** Clone a fantasy.
- **Actores:** Creator-editor (user).
- **Precondiciones:** The fantasy must exist in the system and the user must have modification permissions.
- **Postcondiciones:** Create a copy of the selected fantasy.
- **Main stage:**
  1. The user selects the option “My fantasies”.
  2. The system displays a list of fantasies accessible by the user.
  3. The user selects the fantasy that he wants to copy.
  4. The system displays a pop-up window with the fantasy information and its options.
  5. The user selects the option “Copy fantasy”.
  6. The system creates a copy of the selected fantasy.
- **Extensions:**
  - \*a) At any time, the user can go back to the main menu.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

#### 4.1.4 CRUD background

- **Description:** Allows selecting, modifying and deleting the background.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist in the system and the user must have modification permissions.
- **Postconditions:** The fund that the user has chosen is established.
- **Main stage:**
  1. The user selects the “Background” option.
  2. The system displays a window to add a background to the workspace.
  3. The user selects an image.
  4. The system sets the background selected by the user.
- **Extensions:**
  - \*a) At any time, the user can go back to the main menu.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

#### 4.1.5 CRUD active point

##### Use case: Create active point

- **Description:** Create a new active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist in the system and the user must have modification permissions.
- **Postconditions:** An empty active point is created in the workspace.
- **Main stage:**
  1. The user selects the option “New active point”.
  2. The system creates a new active point in the workspace.
  3. The user can move the active point to the area of the workspace that he wants.
  4. The system will save the active point in the fantasy.
- **Extensiones:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Visualize active point**

- **Description:** Shows an existing active point for reading.
- **Actors:** Creator-editor (user).
- **Preconditions:** Fantasy must exist in the system and the active point must exist in fantasy. In addition, the user must have modification permissions.
- **Postconditions:** The active point is shown for reading.
- **Main stage:**
  1. The user selects the active point that he wants to view.
  2. The system displays a window with the information of the active point and its options.
  3. The user selects the option “Visualize”.
  4. The system displays a window with the summary of that active point.
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Update active point**

- **Description:** Modifies an existing active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** Fantasy must exist in the system and the active point must exist in fantasy. In addition, the user must have modification permissions.
- **Postconditions:** Modifies the selected active point.
- **Main stage:**
  1. The user selects the active point that he wants to modify.
  2. The system displays a window with the information of the active point and its options.
  3. The user selects the option “Modify”.
  4. The system displays the creation window of the active point.
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

### Use case: Delete active point

- **Description:** Deletes an existing active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** Fantasy must exist in the system and the active point must exist in fantasy. In addition, the user must have modification permissions.
- **Postconditions:** Deletes the selected active point.
- **Main stage:**
  1. The user selects the active point that he wants to delete.
  2. The system displays a window with the information of the active point and its options.
  3. The user selects the option “Delete”.
  4. The system displays a confirmation message.
  5. The user selects “Accept”.
  6. The system deletes the active point.
- **Extensions:**
  5. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

## 4.1.6 CRUD image

### Use case: Create image

- **Description:** Inserta una imagen en un punto activo.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist and the fantasy must be being edited.
- **Postconditions:** Insert an image in the selected active point.
- **Main stage:**
  1. The user selects the corresponding active point within the fantasy.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Insert image”.
  4. The system shows a window in which the user chooses where to choose the image (Internet, local, image already used in fantasy).
  5. The user chooses the option “Internet” to include an image of the Internet.
  6. The system asks the user for the url of the image.

7. The user inserts the correct url of the image.
8. The active point takes the shape of the image.

- **Extensions:**

5. a) The user chooses the “Local” option to include an image from his computer.
  1. The system opens a file browser window.
  2. The user selects the desired image and press “Accept”.
  3. The system closes the file browser window.
  4. Step 8.
5. b) The user chooses the option “Image previously used” to include an image already used.
  1. The system opens a window with the images previously used.
  2. The user selects the desired image and press “Accept”.
  3. The system closes the pop-up window.
  4. Step 8.
7. a) The url is not correct.
  1. The system displays an error message.
  2. Step 6.

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Update image

- **Description:** Update an image.

- **Actors:** Creator-editor (user).

- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be an image.

- **Postconditions:** The image is modified.

- **Main stage:**

1. The user selects the corresponding active point within the fantasy.
2. The system opens a popup window with the information of the active point.
3. Step 4 of **Create image**.

- **Extensiones:**

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.



### Use case: Delete image

- **Description:** Deletes an image of an active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be an image.
- **Postconditions:** Delete the image and leave the active point in its default state.
- **Main stage:**
  1. The user selects the corresponding active point within the fantasy.
  2. The system opens a popup window with the information of the active point.
  3. The user selects the image and presses the “Delete” button.
  4. The system displays a confirmation message.
  5. The user selects “Accept”.
  6. The system deletes the image of the active point.
- **Extensions:**
  5. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

## 4.1.7 CRUD video

### Use case: Create video

- **Description:** Insert a video within an active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist and the fantasy must be being edited.
- **Postconditions:** Insert a video in the selected active point.
- **Main stage:**
  1. The user selects the corresponding active point within the fantasy.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Insert video”.
  4. The system shows a window in which the user chooses where to choose the image (Internet, local, video already used in fantasy).
  5. The user chooses the option “Internet” to include an Internet video.
  6. The system asks the user for the url of the video.

7. The user enters the correct url of the video.
8. The system saves the video in the active point.

- **Extensions:**

5. a) The user chooses the “Local” option to include a video from his computer.
  1. The system shows a file browser sale.
  2. The user selects the desired image and press “Accept”.
  3. The system closes the file browser window.
  4. Step 8.
5. b) The user chooses the option “Video previously used” to include a video already used.
  1. The system opens a window with the videos previously used.
  2. The user selects the desired video and press “Accept”.
  3. The system closes the pop-up window.
  4. Step 4.
7. a) The url is not correct.
  1. The system displays an error message.
  2. Step 6.

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Update video

- **Description:** Update a video.

- **Actors:** Creator-editor (user).

- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be a video.

- **Postconditions:** The video is modified.

- **Main stage:**

1. The user selects the corresponding active point within the fantasy.
2. The system opens a popup window with the information of the active point.
3. Step 4 of **Create video**.

- **Extensions:**

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Delete video

- **Description:** Delete a video of an active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be a video.
- **Postconditions:** Delete the video of the active point.
- **Main stage:**
  1. The user selects the corresponding active point within the fantasy.
  2. The system opens a popup window with the information of the active point.
  3. The user selects the video and presses the “Delete” button.
  4. The system displays a confirmation message.
  5. The user selects “Accept”.
  6. The system deletes the video from the active point.
- **Extensions:**
  5. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

### 4.1.8 CRUD text

- **Description:** Insert a text in an active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist and the fantasy must be being edited.
- **Postconditions:** Insert a text in the selected active point.
- **Main stage:**
  1. The user selects the active point to which he wants to add-edit the text.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects enter the desired text in the “Text” field with the formatting options that you want.
  4. The user clicks on the “Accept” button.
  5. The system saves the text in the corresponding active point.
- **Extensions:**

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

## 4.1.9 CRUD quiz

### Use case: Create quiz

- **Description:** Create a small questionnaire about the subject that the active point deals with.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist and the fantasy must be being edited.
- **Postconditions:** Create a small questionnaire in relation to the corresponding active point.
- **Main stage:**
  1. The user selects the corresponding active point.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Create quiz”.
  4. The system shows the possible options.
  5. The user selects “Simple answer”.
  6. The system displays a pop-up window to create the question with its possible answers.
  7. The user populates the pop-up window with the question and the appropriate answers and press “Accept” when it finishes.
  8. The system closes the pop-up window.
  9. The questionnaire is registered in the selected active point.
- **Extensions:**
  3. a) The user chooses the option “Word”.
    1. The system opens a pop-up window to create the question and its answer.
    2. The user populates the pop-up window with the question and the appropriate answer and press “Accept” when it finishes.
    3. Step 8.
  3. b) The user chooses the option “Quiz with images”.
    1. The system opens a pop-up window to create the question with the image and its response.
    2. The user fills in the pop-up window with the question, the image and the appropriate answer, and press “Accept” when it finishes.
    3. Step 8.
  3. c) The user chooses the “Join” option.
    1. The system opens a pop-up window to create the join quiz.
    2. The user populates the pop-up window with the possible answers and their correct answer and press “Accept” when it finishes.
    3. Step 8.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Visualize quiz**

- **Description:** Shows the status of quiz.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist, the fantasy must be edited and there must be a quiz.
- **Postconditions:** Displays the status of quiz at the corresponding active point.
- **Main stage:**
  1. The user selects the corresponding active point.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Read quiz”.
  4. The system displays a pop-up window with the final view of quiz.
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Update quiz**

- **Description:** It allows modifying the quiz.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist, the fantasy must be edited and there must be a quiz.
- **Postconditions:** Modify the quiz of an active point.
- **Main stage:**
  1. The user selects the corresponding active point.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Modify quiz”.
  4. Step 4 of **Create quiz**
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Delete quiz**

- **Description:** Delete the quiz of the selected active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist, the fantasy must be edited and there must be a quiz.
- **Postconditions:** Delete the quiz of the selected active point.
- **Main stage:**
  1. The user selects the corresponding active point.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Delete quiz”.
  4. The system displays a confirmation message.
  5. The user selects “Accept”.
  6. The system deletes the quiz of the active point.
- **Extensions:**
  5. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**4.1.10 CRUD audio effect****Use case: Create audio effect**

- **Description:** Sets a background audio effect on the active point.
- **Actors:** Creator-editor (user).
- **Preconditions:** The corresponding active point must exist and the fantasy must be being edited.
- **Postconditions:** Sets the background audio effect.
- **Main stage:**
  1. The user selects the corresponding active point.
  2. The system displays a pop-up window with the information of the active point.
  3. The user selects the option “Add audio effect”.
  4. The system displays a pop-up window in which you select the user from where you want to select the audio (Internet, local, audio already used in the fantasy).
  5. The user chooses the option “Internet” to include an Internet audio.
  6. The system asks the user for the audio url.
  7. The user inserts the audio url.

8. The system saves the audio in the active point.

- **Extensions:**

5. a) The user chooses the option “Local” to include an audio from his computer.

1. The system opens a file browser window.
2. The user selects the desired audio and press “Accept”.
3. The system closes the file browser window.
4. Step 8.

5. b) The user chooses the “Audio previously used” option to include an already used audio.

1. The system opens a window with the previously used audios.
2. The user selects the desired audio and press accept.
3. The system closes the pop-up window.
4. Step 8.

7. a) The url is not correct.

1. The system displays an error message.
2. Step 6.

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Update audio effect

- **Description:** Allows you to modify an audio effect.

- **Actors:** Creator-editor (user).

- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be an audio.

- **Postconditions:** Update the audio effect.

- **Main stage:**

1. The user selects the active point.
2. The system opens a popup window with the information of the active point.
3. Step 4 of **Create audio effect**

- **Extensions:**

\*a) En cualquier momento, el usuario puede volver atrás.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

**Use case: Delete audio effect**

- **Description:** Delete an audio effect from an active point.
- **Actors:** Creator-editor (usuario).
- **Preconditions:** There must be the corresponding active point, you must be editing the fantasy and there must be an audio.
- **Postconditions:** Deletes an audio effect from an active point.
- **Main stage:**
  1. The user selects the active point.
  2. The system opens a popup window with the information of the active point.
  3. The user selects the audio and presses the “Delete” button.
  4. The system displays a confirmation message.
  5. The user selects “Accept”.
  6. The system deletes the audio from the active point.
- **Extensions:**
  5. a) The user selects “Cancel”.
    1. The system closes the pop-up window.
    2. Step 1.

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**4.1.11 CRUD additional information**

- **Description:** Insert a text as additional fantasy information.
- **Actors:** Creator-editor (user).
- **Preconditions:** You must be editing the corresponding fantasy.
- **Postconditions:** Insert a text as additional fantasy information.
- **Main stage:**
  1. The user selects the option “Additional information”.
  2. The system displays a pop-up window with a text box.
  3. The user enters the desired text in the text box with the desired formatting options.
  4. The user clicks on the “Accept” button.
  5. The system saves the additional information in the corresponding fantasy.
- **Extensions:**

\*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.



#### 4.1.12 Use case: Organize active points

- **Description:** Organize the appearance of active points.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must be created.
- **Postconditions:** It establishes the order of appearance of the active points of the fantasy.
- **Main stage:**
  1. The user selects the corresponding fantasy.
  2. The system displays a window with the fantasy information and available options.
  3. The user selects the option “Organize active points”.
  4. The system displays a pop-up window with the name of the existing active points in the fantasy and a box to establish the order of appearance.
  5. The user establishes the order of appearance in the boxes next to the name of the active points of the fantasy.
  6. The user presses “Accept” to save the changes made.
  7. The system keeps the order of appearance of the active points.
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

#### 4.1.13 CRUD final quiz

##### Use case: Create final quiz

- **Description:** Create a questionnaire about the subject that fantasy deals with.
- **Actors:** Creator-editor (user).
- **Preconditions:** There must be and must be editing the fantasy.
- **Postconditions:** Create a questionnaire related to the theme of fantasy.
- **Main stage:**
  1. The user presses the “Final questionnaire” button.
  2. The system shows the possible options.
  3. The user selects “Create new quiz”.
  4. The system shows the possible creation options.
  5. The user selects “Simple answer”.
  6. The system displays a pop-up window to create the question with its possible answers.
  7. The user populates the pop-up window with the question and the appropriate answers and press “Accept” when it finishes.
  8. The system closes the pop-up window.

9. The questionnaire is recorded in the fantasy.

- **Extensions:**

5. a) The user chooses the option “Word”.

1. The system opens a pop-up window to create the question and its answer.
2. The user populates the pop-up window with the question and the appropriate answer and press “Accept” when it finishes.
3. Step 8.

5. b) The user chooses the option “Quiz with images”.

1. The system opens a pop-up window to create the question with the image and its response.
2. The user fills in the pop-up window with the question, the image and the appropriate answer, and press “Accept” when it finishes.
3. Step 8.

5. c) The user chooses the “Join” option.

1. The system opens a pop-up window to create the join quiz.
2. The user populates the pop-up window with the possible answers and their correct answer and press “Accept” when it finishes.
3. Step 8.

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

### Use case: Visualize final quiz

- **Description:** Shows the status of quiz.

- **Actors:** Creator-editor (user).

- **Preconditions:** The fantasy must exist and must be being edited, and the final quiz must exist.

- **Postconditions:** Shows the status of the final quiz.

- **Main stage:**

1. The user selects the “Final questionnaire” button.
2. The system shows the possible options.
3. The user selects the option “Read quiz final”.
4. The system displays a pop-up window with the final view of quiz.

- **Extensions:**

\*a) At any time, the user can go back.

- **Variations:** None.

- **Not-functional:** None.

- **Issues:** None.

**Use case: Update final quiz**

- **Description:** It allows modifying the final quiz.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist and must be being edited, and the final quiz must exist.
- **Postconditions:** Modify the final quiz of the fantasy.
- **Main stage:**
  1. The user presses the "Final questionnaire" button.
  2. The system shows the possible options.
  3. The user selects the option "Modify quiz final".
  4. Use case **Create final quiz**
- **Extensions:**
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

**Use case: Delete final quiz**

- **Description:** Delete the final quiz of the fantasy.
- **Actors:** Creator-editor (user).
- **Preconditions:** The fantasy must exist and must be being edited, and the final quiz must exist.
- **Postconditions:** Delete the final quiz of the fantasy.
- **Main stage:**
  1. The user presses the "Final questionnaire" button.
  2. The system shows the possible options.
  3. The user selects the final "Delete quiz" option.
  4. The system displays a confirmation message.
  5. The user selects "Accept".
  6. The system deletes the final quiz of the fantasy.
- **Extensions:**
  5. a) The user selects "Cancel".
    1. The system closes the pop-up window.
    2. Step 1.
  - \*a) At any time, the user can go back.
- **Variations:** None.
- **Not-functional:** None.
- **Issues:** None.

## **4.2 User interface desing**

## **4.3 Data design**

### **DATABASE DIAGRAM**

## **4.4 Design of components**

## **4.5 Parameterization of the base software**

## **Chapter 5**

# **Implementation of the system**

### **5.1 Technological environment**

### **5.2 Source code**

### **5.3 Code quality**



# **Chapter 6**

## **System tests**

### **6.1 Unit tests**

### **6.2 Integration testing**

### **6.3 System tests**

#### **6.3.1 Functional testing**

#### **6.3.2 Non-functional tests**

### **6.4 Acceptance Tests**





## **Part III**

# **Epilogue**



## **Chapter 7**

# **User manual**

### **7.1 Introduction**

### **7.2 Features**

### **7.3 Previous requirements**

### **7.4 Utilization**



## **Chapter 8**

# **Installation and operation manual**

### **8.1 Introduction**

### **8.2 Previous requirements**

### **8.3 Inventory of components**

### **8.4 Installation procedures**

### **8.5 Operating procedures and service level**

### **8.6 Implantation tests**



## **Chapter 9**

# **Conclusions**

### **9.1 Objectives**

### **9.2 Learned lessons**

### **9.3 Future work**