

---

---

# Creativa: Aplicación para el fomento de la Escritura Creativa

Creativa: Application for the promotion of Creative  
Writing

---

---

Por

Genesis Brigitti Duque Escobar

Luis Alberto Jaramillo Pulido

Gonzalo Rivas Fernández



Grado en Ingeniería Informática.  
Grado en Ingeniería del Software.  
FACULTAD DE INFORMÁTICA

*Dirigido por*

Cora Requena Hidalgo  
Adrián Riesco Rodríguez

MADRID, 2020–2021



# Agradecimientos

En primer lugar, queremos agradecer a Adrián Riesco Rodríguez por sus consejos y continua ayuda durante el proyecto. A continuación, a Cora Requena Hidalgo por confiarnos esta idea y aportar sus conocimientos sobre la materia. También queríamos dar las gracias a Esteban Armas por darnos acceso a las herramienta ShareLatex para realizar este documento y a Maria Jhenay Escobar González por proveer las ilustraciones empleadas en la aplicación.

## Gonzalo

Quiero agradecer a mis amigos Rocío, Marina, Alberto, Javier, Enrique y ambos Carlos por su apoyo en los momentos complicados, y a mis compañeros de la Universidad por acompañarme, no habría sido lo mismo sin ellos.

También quiero agradecer a mi familia por todo el apoyo y el ánimo que me han dado durante la carrera, es gracias a ellos que haya llegado hasta aquí.

## Genesis

Quiero agradecer a mi madre por brindarme la oportunidad de conseguir mis sueños y por ser la mujer que siempre admiraré y amaré en esta vida. A mi padre, por enviarme todos los días su amor y apoyo. A toda mi familia, porque 7258 km no impidieron que sus consejos y cariños llegaran hasta mi, gracias.

También dar las gracias a mis amigos Héctor, Marisabel, Adriana, Kristel y Brenda por estar a mi lado y brindarme la ayuda cuando pasaba por momentos muy difíciles. Por ultimo, a todas mujeres que luchan por romper las barreras y ser felices consiguiendo sus metas.

## Luis

Quiero agradecer a amigos y compañeros por los buenos momentos que he pasado junto a ellos, haciendo que el paso por la universidad haya sido una muy buena experiencia.

Y también quiero agradecer a mi familia, en especial a mi madre y a mi hermana por sus buenos consejos, comprensión y apoyo incondicional que me han brindado durante todo este tiempo y que gran parte de lo conseguido también es de ellos.



# Resumen

El objetivo de este Trabajo de Fin de Grado es crear una aplicación web que permita la realización de desafíos y escritos literarios, promoviendo de esta manera la escritura creativa.

Para su uso, se ha añadido un sistema de cuentas de usuario basado en roles, así como herramientas para permitir a los usuarios organizarse y participar en la creación mediante un editor de texto y un sistema de almacenamiento multimedia.

Este proyecto contempla la posibilidad de coordinar la actividad creativa con la comunidad educativa, dando la posibilidad a profesores a establecer unas pautas que impulsen a los estudiantes interesados a realizar nuevas creaciones como respuesta a retos literarios de diversas temáticas. Se tiene en cuenta la existencia de diferentes intereses creativos y se ha diseñado un sistema que agrupe a los usuarios con intereses afines.

A la hora de usar la aplicación se debe usar una cuenta de usuario a la que se le haya concedido acceso por un responsable de la aplicación.

**Palabras clave:** Escritura creativa, Aplicación web, ReactJS, NodeJS, ExpressJS, Javascript, Axios, Escrito, Desafío, SQL.

# Abstract

The main goal of this Degree Thesis is to create a web application that eases the creation of literary challenges and writings, supporting creative writing.

In order to aid in its usage, a role-based account system has been added, as well as tools that enables student organization and content creation through a text editor and a media storage system.

This project takes into account the possibility of joining creative and educative activities, allowing teachers to create a set of rules in order to push interested students into developing new literary creations of diverse themes. A system that groups users with similar tastes has been designed in order to acknowledge the existence of different creative interests.

An account with administrator permissions is required in order to use the app.

**Keywords:** Creative writing, Web app, ReactJS, NodeJS, ExpressJS, Javascript, Axios, Writing, Challenge, SQL.

# Sobre TEF<sub>L</sub>ON<sup>N</sup>X

TEFLON X(CC0 1.0(DOCUMENTACIÓN) MIT(CÓDIGO))ES UNA PLANTILLA DE L<sup>A</sup>T<sub>E</sub>X CREADA POR DAVID PACIOS IZQUIERDO CON FECHA DE ENERO DE 2018. CON ATRIBUCIONES DE USO CC0.

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado, Trabajos de Fin de Máster o Doctorados. La versión usada es la X

V:X OVERLEAF V2 WITH XE<sub>L</sub>A<sub>T</sub>E<sub>X</sub>, MARGIN 1IN, BIB

## Contacto

**Autor:** DAVID PACIOS IZQUIERO

**Correo:** dpacios@ucm.es

**ASCII:** ascii@ucm.es

DESPACHO 110 - FACULTAD DE INFORMÁTICA

# Índice general

	Página
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	1
1.3. Metodología . . . . .	2
1.4. Arte de la aplicación . . . . .	2
1.5. Estructura de la memoria . . . . .	3
1.6. Repositorio . . . . .	3
<b>2. Introduction</b>	<b>4</b>
2.1. Motivation . . . . .	4
2.2. Objectives . . . . .	4
2.3. Methodology . . . . .	5
2.4. Art used on the platform . . . . .	5
2.5. Document Structure . . . . .	5
2.6. Repository . . . . .	6
<b>3. Tecnologías</b>	<b>7</b>
3.1. Servidor Node.js . . . . .	7
3.2. Cliente ReactJS . . . . .	7
3.3. Bases de datos . . . . .	8
3.3.1. XAMPP . . . . .	8
3.3.2. MySQL . . . . .	8
3.4. Gestión de proyectos . . . . .	8
3.4.1. Trello . . . . .	8
3.4.2. Google Drive . . . . .	9
3.5. Control de versiones . . . . .	9
3.5.1. Github . . . . .	9
3.5.2. Git . . . . .	9
3.5.3. Gitkraken . . . . .	9
3.6. Prototipo Balsamiq Cloud . . . . .	9
3.7. Comunicación Google Meet . . . . .	10
3.8. Memoria Sharelatex . . . . .	10
3.9. Bibliotecas . . . . .	10
3.9.1. Multer . . . . .	10
3.9.2. React Draft Wysiwyg . . . . .	10
3.9.3. Axios . . . . .	10
3.10. Diseño web . . . . .	10



3.10.1. CSS . . . . .	11
3.10.2. React-Bootstrap . . . . .	11
<b>4. Arquitectura</b>	<b>12</b>
4.1. Patrón de diseño . . . . .	12
4.2. Estructura de la Arquitectura de la aplicación . . . . .	13
4.2.1. Backend . . . . .	13
4.2.2. Frontend . . . . .	13
4.3. Base de datos . . . . .	14
4.4. Autenticación basada en tokens . . . . .	14
4.4.1. Funcionamiento del <i>Token JWT</i> en la aplicación . . . . .	15
4.5. Funcionalidades Principales . . . . .	15
4.5.1. Usuarios . . . . .	15
4.5.2. Desafíos . . . . .	16
4.5.3. Escritos . . . . .	16
4.5.4. Grupos . . . . .	17
4.5.5. Equipos . . . . .	17
4.5.6. Mensajería . . . . .	18
4.6. Prototipo . . . . .	18
<b>5. Modelo de datos</b>	<b>20</b>
5.1. Modelo entidad-relación . . . . .	20
5.2. Base de datos SQL . . . . .	22
5.2.1. Usuario . . . . .	23
5.2.2. Grupo . . . . .	23
5.2.3. Desafío . . . . .	24
5.2.4. Escrito . . . . .	24
5.2.5. Equipo . . . . .	25
5.2.6. Categoría . . . . .	26
5.2.7. Equipoestudiante . . . . .	26
5.2.8. Grupoestudiante . . . . .	26
5.2.9. Mensajería . . . . .	26
5.2.10. Multimediadesafío . . . . .	27
5.2.11. Multimediadesafío . . . . .	27
<b>6. Guía de uso</b>	<b>29</b>
6.1. General . . . . .	29
6.1.1. Módulo iniciar y registrar usuario . . . . .	29
6.1.2. Módulo perfil del usuario . . . . .	31
6.1.3. Módulo modificar datos . . . . .	31
6.1.4. Módulo tablero . . . . .	32
6.1.5. Módulo alertas sobre errores . . . . .	33
6.2. Usuario profesor . . . . .	34
6.2.1. Módulo grupos profesor . . . . .	34
6.2.2. Módulo estudiantes . . . . .	36
6.2.3. Módulo perfil del estudiante . . . . .	37
6.2.4. Módulo solicitantes . . . . .	39
6.2.5. Módulo crear grupo . . . . .	40
6.2.6. Módulo componer y editar un desafío . . . . .	40

---

6.2.7.	Módulo editar un escrito . . . . .	41
6.2.8.	Módulo información del equipo . . . . .	43
6.3.	Usuario estudiante . . . . .	43
6.3.1.	Módulo grupos estudiante . . . . .	43
6.3.2.	Módulo crear escrito . . . . .	48
6.3.3.	Módulo Escritos . . . . .	48
6.3.4.	Módulo equipos . . . . .	49
6.3.5.	Módulo componer y editar un escrito . . . . .	49
6.3.6.	Módulo mensajería . . . . .	50
6.4.	Usuario Administrador . . . . .	51
6.4.1.	Módulo grupos administrador . . . . .	51
6.4.2.	Módulo usuarios . . . . .	51
6.4.3.	Módulo solicitantes . . . . .	52
6.4.4.	Módulo información del grupo . . . . .	53
6.4.5.	Módulo perfil del estudiante . . . . .	54
6.4.6.	Módulo perfil del profesor . . . . .	55
<b>7.</b>	<b>Contribuciones individuales</b>	<b>57</b>
7.1.	Luis Alberto Jaramillo Pulido . . . . .	57
7.1.1.	Primera toma de contacto con el cliente . . . . .	57
7.1.2.	Etapa de Investigación . . . . .	57
7.1.3.	Diseño del Prototipo . . . . .	58
7.1.4.	Gestión del proyecto . . . . .	58
7.1.5.	Colaboración en el Repositorio . . . . .	58
7.1.6.	Diseño de la Base de datos . . . . .	59
7.1.7.	Implementación . . . . .	59
7.1.8.	Presentación de la aplicación al cliente . . . . .	61
7.1.9.	Aportaciones en la memoria . . . . .	62
7.2.	Génesis Brigitti Duque Escobar . . . . .	62
7.2.1.	Etapa de investigación . . . . .	62
7.2.2.	Diseño de prototipo . . . . .	63
7.2.3.	Gestión de proyecto . . . . .	63
7.2.4.	Repositorio . . . . .	63
7.2.5.	Conexión del motor lógico y la interfaz de usuario . . . . .	63
7.2.6.	Implementación de autenticación del usuario . . . . .	64
7.2.7.	Implementación del estilos . . . . .	64
7.2.8.	Presentación de la aplicación al cliente . . . . .	64
7.2.9.	Aportación a la memoria . . . . .	65
7.3.	Gonzalo Rivas Fernández . . . . .	65
7.3.1.	Funciones del rol de profesor . . . . .	65
7.3.2.	Funciones del rol de administrador . . . . .	66
7.3.3.	Mecanismos generales de la aplicación: . . . . .	67
7.3.4.	Aportaciones a la memoria: . . . . .	67
7.3.5.	Cronología . . . . .	68
<b>8.</b>	<b>Evaluación de la encuesta realizada sobre <i>Creativa</i></b>	<b>69</b>
<b>9.</b>	<b>Conclusiones y trabajo futuro</b>	<b>72</b>
9.1.	Conclusiones . . . . .	72

---

---

9.2. Trabajo futuro . . . . .	73
<b>10. Concluding remarks and ongoing work</b>	<b>74</b>
10.1. Concluding remarks . . . . .	74
10.2. Future work . . . . .	75
<b>A. Glosario</b>	<b>78</b>

# Índice de figuras

4.1. Esquema de autenticación basada en token . . . . .	15
4.2. Crear desafío (primer boceto) . . . . .	19
4.3. Listar estudiantes (primer boceto) . . . . .	19
5.1. Modelo entidad relación . . . . .	20
6.1. Iniciar sesión . . . . .	29
6.2. Alerta de error . . . . .	30
6.3. Error usuario no válido . . . . .	30
6.4. registrar usuario . . . . .	31
6.5. Perfil del usuario . . . . .	31
6.6. Modificar datos . . . . .	32
6.7. Confirmar modificación . . . . .	32
6.8. Barra lateral de opciones . . . . .	33
6.9. Error 404. . . . .	33
6.10. Error 500. . . . .	34
6.11. Selección de grupos y renombrar . . . . .	34
6.12. Desafíos de un grupo . . . . .	35
6.13. Escritos de un grupo . . . . .	35
6.14. Equipos de un grupo . . . . .	36
6.15. Lista de estudiantes . . . . .	36
6.16. Lista de estudiantes . . . . .	37
6.17. Perfil del Estudiante . . . . .	37
6.18. Perfil del Estudiante . . . . .	38
6.19. Perfil del Estudiante . . . . .	38
6.20. Perfil del Estudiante . . . . .	39
6.21. Lista de alumnado solicitante . . . . .	39
6.22. Aceptar solicitud . . . . .	40
6.23. Crear grupo . . . . .	40
6.24. Crear un desafío . . . . .	41
6.25. Validar desafío creado . . . . .	41
6.26. Editar escrito “Detalles del desafío” . . . . .	42
6.27. Editar escrito “Escrito del estudiante 1” . . . . .	42
6.28. Editar escrito “Escrito del estudiante 2” . . . . .	42
6.29. información del equipo . . . . .	43
6.30. Selección de grupos y gestión. . . . .	43
6.31. Listado de desafíos. . . . .	44
6.32. Listado de desafíos en equipo. . . . .	44

---

6.33. Listado de escritos. . . . .	45
6.34. Listado de escritos en equipo. . . . .	45
6.35. Enviar solicitud a equipos . . . . .	46
6.36. Crear equipo. . . . .	46
6.37. Gestionar equipo. . . . .	47
6.38. Abandonar el equipo. . . . .	47
6.39. Eliminar el equipo. . . . .	48
6.40. Abandonar el equipo un estudiante no líder. . . . .	48
6.41. Gestión de equipos. . . . .	49
6.42. Requisitos y detalles que debe cumplir el escrito. . . . .	49
6.43. Redacción de un escrito. . . . .	50
6.44. Opciones de gestión de una redacción. . . . .	50
6.45. Mensajería. . . . .	51
6.46. Gestionar grupos. . . . .	51
6.47. Lista de usuarios dados de alta. . . . .	52
6.48. Datos del grupo. . . . .	52
6.49. Datos del grupo. . . . .	53
6.50. Datos del grupo. . . . .	53
6.51. Listado de estudiantes. . . . .	54
6.52. Perfil del estudiante. . . . .	54
6.53. Gestionar perfil del estudiante. . . . .	54
6.54. Escritos del estudiante. . . . .	55
6.55. Perfil del profesor. . . . .	55
6.56. Grupos del estudiante. . . . .	56
8.1. Resolución sobre la noción explícita de <i>Creativa</i> . . . . .	69
8.2. Respuestas sobre la experiencia del usuario. . . . .	70
8.3. Resolución sobre el control de acceso a los estudiantes. . . . .	70
8.4. Agregación de ideas extras para <i>Creativa</i> . . . . .	71
8.5. Perfil de usuarios para <i>Creativa</i> . . . . .	71

# Capítulo 1

## Introducción

En este capítulo se desarrollan los objetivos que trata de cumplir la aplicación , así como la motivación que lleva a su desarrollo.

### 1.1. Motivación

Tradicionalmente, la escritura creativa ha empleado como medios principales el papel y bolígrafo para desarrollar su actividad. Aunque estos medios proporcionan una serie de libertades, también cuenta con importantes carencias con respecto a los medios digitales, principalmente la falta de interactividad, así como la necesidad de un espacio y tiempo dedicados a la propia actividad. Por otra parte, existen ciertas herramientas digitales que permiten ejercer la escritura creativa, sin embargo, muchas de ellas ponen límites a la creatividad del escritor, ya sea por formatos limitados o por no ser fácilmente accesibles. De la misma manera, se ha observado que las modalidades antes mencionadas, tanto digitales como físicas carecen en la mayor parte de una fuente de inspiración que pueda estimular la misma actividad creativa. Por todo ello, en colaboración con el personal de la Facultad de Ciencias de la Información de la Universidad Complutense de Madrid se ha desarrollado *Creativa*, una aplicación web que pretende proporcionar una plataforma donde se pueda crear con libertad gracias a la inclusión de formatos tanto escritos como multimedia. Se incluye también un sistema por el que los profesores pueden interactuar con los alumnos proponiéndoles desafíos literarios que sirvan de inspiración a la hora del proceso creativo. Debido a la naturaleza de *Creativa* como aplicación en línea, puede ser accedida desde cualquier dispositivo con conexión a Internet, y está optimizada para soportar distintos formatos de pantalla para la comodidad del usuario, aprovechando la prevalencia que tienen los dispositivos móviles en la actualidad. Esta herramienta también permite a los profesores dar una respuesta constructiva a los distintos tipos de creaciones que puedan crear los estudiantes en relación con los desafíos.

### 1.2. Objetivos

El objetivo de este proyecto es crear un punto de encuentro donde los profesores puedan inspirar a los estudiantes a ejercer la escritura creativa mediante una serie de desafíos de

diferentes temáticas.

- Crear un espacio donde los estudiantes puedan colaborar para escribir de forma colaborativa.
- Permitir el uso de contenidos multimedia para apoyar esa misma actividad.
- Fomentar la inspiración mediante los desafíos creados por los profesores.
- Hacer más cómoda y accesible la actividad creativa utilizando la tecnología web.
- Facilitar la organización al permitir a los profesores agrupar a los estudiantes según sus intereses y proponerles desafíos de acuerdo a los mismos.

### 1.3. Metodología

Durante el desarrollo de la aplicación, se ha seguido una metodología ágil *Scrumban*, elegida por dar una mayor flexibilidad a la hora de planificar las tareas del desarrollo, aportando una mayor eficiencia. Esta metodología otorga transparencia a todo el proceso de trabajo al mantener todas las tareas visibles y da libertad a la hora de realizarlas.

A su vez, esta forma de trabajo permite que cada integrante del equipo de desarrollo pueda seleccionar las tareas que desea completar, cada miembro del equipo puede controlar el progreso de las mismas al disponer de una plataforma que organiza las tareas según su grado de desarrollo.

Al principio hubo reuniones con el cliente cada quince días, que consistieron en la obtención de requisitos. Este proceso se hizo a través de las historias de usuario, recogiénolas en el *Product Backlog*.

El trabajo en equipo se organizó en pequeñas iteraciones, siendo controladas mediante un tablero visual creado en la herramienta en línea *Trello*, para mantener el flujo del desarrollo se utilizaron las etiquetas de trabajo en progreso (*WIP/Work in Progress*) que consistió en dividir el tablero en tres secciones *To Do, Doing*, y *Done*, fijando cuántas tareas como máximo pueden realizarse en cada fase del ciclo de trabajo, impidiendo empezar otras tareas hasta que no se hayan cerrados aquellas en las que se están trabajando.

Las reuniones de planificación se han llevado a cabo semanalmente cada martes. En estas reuniones se presentaban nuevas funcionalidades correspondientes a cada iteración, también se planificaba qué historias de usuario y qué tareas se completarían en la siguiente iteración.

### 1.4. Arte de la aplicación

Todos los componentes visuales como las imágenes empleados en la plataforma han sido desarrollados en exclusiva para la misma por el equipo de desarrollo y sus colaboradores.

## 1.5. Estructura de la memoria

Este documento se compone de los siguientes apartados:

- En el capítulo 2 se recoge la versión en inglés de este capítulo.
- En el capítulo 3 se exponen las tecnologías empleadas para la realización de *Creativa*.
- En el capítulo 4 se explica la arquitectura escogida para dar forma a la aplicación, así como un primer prototipo y sus requisitos.
- En el capítulo 5 se desarrolla la capa de integración de la plataforma, entrando en detalle en la estructura y qué utilidad tiene cada entidad del modelo de datos.
- En el capítulo 6 se incluye una guía que explica paso por paso cómo interactuar con *Creativa*.
- En el capítulo 7 se anotan las diferentes aportaciones que ha hecho cada miembro del equipo al proyecto.
- En el capítulo 8 se exponen los resultados de una puesta en contacto con futuros usuarios de la aplicación.
- En el capítulo 9 se describen las conclusiones finales obtenidas tras la realización de este trabajo, así como las posibles nuevas funcionalidades que se podrían proponer en el futuro para *Creativa*.
- En el capítulo 10 se recoge la versión en inglés del capítulo 9, conclusiones y trabajo futuro.
- En el apartado A del anexo, el glosario, se encuentra una pequeña lista de términos donde se definen varios conceptos relacionados con el funcionamiento de la aplicación.

## 1.6. Repositorio

EL código de la *Creativa* está disponible en el siguiente repositorio: <https://github.com/GenesisDuque/TFG-EscrituraCreativa.git>



# Capítulo 2

## Introduction

This chapter details the objectives of the app, as well as the motivation behind its development.

### 2.1. Motivation

Creative writing has traditionally used pen and paper as its main platform for its activity. Even though these mediums provide certain freedom, they also come with several important inconveniences when faced against digital mediums, like lack of interactivity or the need of a dedicated time and environment in which the act of writing takes place. On the other hand, although some digital creative writing tools do exist, most of them limit the writer's creativity by their lack of accesibility or poor format options. Both mediums—traditional and digital—have shown that they lack of a inspirational component that pushes the writer to create. Because of this, the *Creactiva* app has been developed with the colaboration of staff from the Facultad de Ciencias de la Información at the Universidad Complutense de Madrid, with the goal of providing a platform that supports creativity by accepting both text and media submissions. Teachers can interact with students by challenging them to create new pieces. Given the app nature as a web platform it can be accessed by any device with access to the Internet, taking advantage of the prevalence of the use of the smartphone. This tool allows teachers to give feedback to their student's creations.

### 2.2. Objectives

The main goal of this project is to create a meeting point where teachers can inspire student creativity through challenges of different themes.

- To create a space where students can create by cooperating.
- To allow the use of media in order to further support this activity.
- To foment student inspiration via teacher made challenges.
- To make content creation more accessible and comfortable.

- To ease user organization as allowing teachers create groups of students with similar creative tastes.

## 2.3. Methodology

The agile methodology *Scrumban* has been chosen for this project because it provides flexibility while planning tasks as well as efficiency in their handling. This methodology provides transparency as it keeps ongoing tasks visible and gives freedom on how they are completed.

Because of this each member in the development team can choose which tasks can they work on from the available ones, as well as monitor their progress by accessing a platform that sorts these tasks by their completion status.

At the beginning there was a meeting held every fifteen days with the purpose of obtaining the project requisites. This was made through the creation of user stories that are collected within the *Product Backlog*.

The resulting teamwork has consisted in several small iterations that were organized through a board created using the web tool *Trello* that maintained the progress of the workflow using tags to mark each task progress (*WIP/Work in progress*) that resulted in dividing the board in three sections; *To Do*, *Doing*, and *Done*, keeping a maximum number of tasks in each category that made impossible to create a new task until those in development were closed.

Meetings were held each Tuesday. In these meetings new functionalities from the current iteration were presented, as well as planning which user stories and tasks would be worked on next.

## 2.4. Art used on the platform

All visual assets have been developed in exclusive for the use in the platform by the team members and their collaborators.

## 2.5. Document Structure

The rest of the document is organized as follows:

- Chapter 3 exposes the technologies used in the development of *Creativa*.
- Chapter 4 explains the platform structure as well as its first prototype and requirements.
- Chapter 5 explains in detail the app data structure, as well as how each of its parts work within the platform.
- Chapter 6 includes a user guide for *Creativa*.

- Chapter 7 lists the different additions to the project made by each of the team members.
- Chapter 8 shows the results of a meeting with future users of the app.
- Chapter 9 describes the concluding remarks on the development of the platform, as well as the ways that *Creativa* could be expanded.
- Chapter 10 holds an english translation of Chapter 9, concluding remarks and future work.
- Appendix A of the annexed documents, the glossary, holds a short list of terms related to the inner workings of the app as well as their definitions.

## 2.6. Repository

*Creativa*'s code is available in the following repository: <https://github.com/GenesisDuque/TFG-EscrituraCreativa.git>

# Capítulo 3

## Tecnologías

En esta sección presentamos las tecnologías y herramientas utilizadas para el desarrollo del proyecto.

### 3.1. Servidor Node.js

*Node.JS* [1] es un entorno de desarrollo multiplataforma para *JavaScript* de código abierto dirigido principalmente a servidores web. Al tratarse el proyecto de una aplicación web se ha optado por esta tecnología para la implementación del servidor.

Se ha elegido esta tecnología porque permite manejar varios eventos y alto tráfico de usuarios como profesores y estudiantes. Asimismo, soporta numerosas conexiones concurrentes, permitiendo disminuir el coste de infraestructura si en un futuro se decidiese continuar con el proyecto con un enfoque más dirigido al negocio.

Otras de las razones es la eficiencia que brinda en tiempo real de uso de datos; las operaciones de entrada/salida son sin bloqueo cuando un servidor web hace una petición y espera una respuesta, también porque está orientado a eventos asíncronos, y al estar basado en eventos y sin bloqueos hace que sea liviano, eficiente y posea un gran rendimiento.

### 3.2. Cliente ReactJS

*ReactJS* [1] es un *framework* basado en *JavaScript* que permite crear interfaces de usuario.

Esta tecnología se ha seleccionado en la implementación del lado del cliente por muchas razones, una de ellas es el buen rendimiento que proporciona al ejecutar la aplicación. Al usar virtual *DOM*, permite cambiar partes de la interfaz sin renderizar toda la aplicación en comparación de otras aplicaciones web, al estar vinculada una componente a una *url* no hay necesidad de recargar la página, siendo las componentes las que se recargan, esto hace que la interacción con la aplicación resulte más dinámica.

Otra razón es la separación completa entre el lado del servidor y el lado del cliente, al tenerlo separado nos brinda la posibilidad de poder trabajar en equipo, permite dividir

el trabajo, como tener personas que trabajan en el servidor y otros en el cliente.

Otro motivo es porque está orientado a componentes, es modular mediante componentes, nos permite reutilizar código, cada parte de la página se puede dividir en componentes, evitando duplicar código innecesariamente.

### 3.3. Bases de datos

Para la base de datos se han empleado las siguientes herramientas.

#### 3.3.1. XAMPP

Como servidor web se ha utilizado XAMPP [2], que incluye MySQL[3] para la gestión de la base de datos, se ha encontrado ideal porque ha permitido trabajar de forma rápida desde un primer momento sin requerir tiempo en su ejecución o en configuraciones a comparación de otros servidores web. Es un servidor independiente de plataforma de código libre, es gratuito e incluye bases de datos MySQL.

#### 3.3.2. MySQL

MySQL[3] es un gestor de bases de datos que habitualmente viene integrado con servidores web, comportándose como un sistema relacional que guarda datos en tablas separadas, es decir, trabaja como un modelo cliente-servidor, almacenando un conjunto de datos que son organizados en forma de tablas y cada una de estas tablas están relacionadas entre si.

### 3.4. Gestión de proyectos

Para la organización del equipo de trabajo se han empleado las siguientes herramientas.

#### 3.4.1. Trello

Como gestor de proyectos en línea se ha usado *Trello* [4]. Se ha considerado que esta herramienta se adecúa a la metodología empleada *Scrumban*.

Este gestor de proyectos posee un sistema de tarjetas que ha permitido la representación de los artefactos de la metodología ágil como son *Product Backlog*, *Sprint Backlog*, entre otros.

Otro de los usos ha sido para la organización de tareas, listas, eventos, y un registro de actividades. Todo lo proporcionado por esta herramienta ha sido de gran utilidad para poner en práctica la metodología ágil *Scrumban*.

### 3.4.2. Google Drive

Como servicio de alojamiento de archivos se ha optado por *Google drive* al estar todo el equipo más familiarizado con la herramienta, se ha utilizado para alojar ficheros como documentos de investigación de alguna tecnología nueva, para la organización de la memoria, para los esquemas de representación de la base de datos y la presentación del proyecto, *mockups* entre otros.

## 3.5. Control de versiones

Para la gestión del proyecto se han empleado las siguientes herramientas.

### 3.5.1. Github

*GitHub* [5] se ha empleado como repositorio remoto para la gestión del proyecto, usar esta herramienta nos ha permitido trabajar de forma colaborativa, mediante el seguimiento y compartición de un repositorio de versiones en común.

### 3.5.2. Git

*Git* [6] es un sistema de control de versiones de código abierto y gratuito, que ha sido de utilidad para administrar y realizar un seguimiento del historial de código fuente, registrando cambios realizados en un archivo o conjuntos de archivos en el proyecto, y la posibilidad de poder revisar o restaurar versiones anteriores.

### 3.5.3. Gitkraken

*Gitkraken* [7] es una interfaz gráfica multiplataforma para *Git*, que ha permitido hacer todas las operaciones de gestión del control de versiones desde una aplicación de escritorio.

## 3.6. Prototipo Balsamiq Cloud

*Balsamiq Cloud* [8] es una herramienta de diseño de interfaz de usuario utilizado para crear prototipos o diseños para una aplicación web, se ha empleado en la elaboración del primer diseño de la aplicación presentada al cliente, y que ha ayudado para entender de forma más detallada los requisitos del cliente.

## 3.7. Comunicación Google Meet

*Google Meet* es una aplicación o servicio de comunicación, se ha utilizado esta herramienta para las reuniones de equipo y presentaciones del proyecto al cliente.

## 3.8. Memoria Sharelatex

*ShareLatex* es un editor en línea de Látex que permite trabajar de forma colaborativa y sido empleado para la elaboración de la memoria.

## 3.9. Bibliotecas

Aquí se presentan las bibliotecas empleadas en la aplicación.

### 3.9.1. Multer

*Multer* [9] es un *middleware* para Express y Node.js, es un módulo empleado en el lado del servidor, su uso es la gestión de ficheros multimedia de desafíos y escritos. Cuando un usuario de la aplicación (profesor o estudiante), agrega ficheros multimedia a un desafío o a un escrito, se suben a un directorio `./public/multimedia` del servidor.

### 3.9.2. React Draft Wysiwyg

*React Draft Wysiwyg* [10] es una biblioteca de *ReactJs* empleado para la creación de un editor de texto cuya finalidad es elaborar textos narrativos en los desafíos y escritos. Este editor permite controlar la apariencia del texto y desarrollar una amplia gama de composición de texto enriquecido que van desde estilos de textos básicos hasta funciones más complejas.

### 3.9.3. Axios

Axios [11] es una biblioteca de *Javascript* que permite realizar peticiones *HTTP* a un servidor y realizar operaciones *CRUD*, se ha utilizado en el lado del cliente para enviar peticiones desde el navegador al servidor.

## 3.10. Diseño web

Los siguientes instrumentos son herramientas, lenguajes y bibliotecas para el diseño gráfico, estructuración y agregación de componentes visuales.

### 3.10.1. CSS

Es una hoja de estilos en cascada aplicada en ficheros de extensión *HTML*, afectando a los objetos definidos para mejorar la visualización y estilo tal como su color o tamaño.

### 3.10.2. React-Bootstrap

*React Bootstrap* [12] es una herramienta de código abierto para la usabilidad en el diseño web a través de la importación de plantillas o componentes basados en *HTML* y *CSS*, reemplazando el *Bootstrap*, *javascript* y dependencias con *JQUERY*, lo que la convierte en una buena biblioteca externa como opción a la hora de desarrollar las interfaces.



# Capítulo 4

## Arquitectura

En esta sección se explica la arquitectura de la aplicación web, del patrón de diseño empleado, bases de datos, funcionalidades, prototipo y guía de uso.

### 4.1. Patrón de diseño

El patrón de diseño utilizado en el proyecto es el patrón Modelo-Vista-Controlador, se ha optado por este patrón de diseño porque permite que el proyecto sea escalable, modular, y de fácil mantenimiento, permitiendo la reutilización de los componentes.

Otro motivo es que proporciona una separación clara de donde tiene que ir cada tipo de lógica, separando la aplicación en tres componentes lógicos: modelo, vista y controlador.

**Capa del modelo:** En la aplicación es la lógica de negocio en la aplicación, relacionando los datos que la aplicación va utilizar, como consultas, actualizaciones, creación de datos y eliminación de la misma. El modelo notifica los cambios de estado para que la vista se actualice.

**Capa de la vista:** La capa donde el usuario interactúa con la aplicación mediante una interfaz de usuario, enviando las peticiones al controlador.

**Capa del controlador:** Es la lógica de la aplicación, recibe solicitudes de la vista y ejecuta las acciones en el modelo.

En el proyecto la separación se realiza mediante dos carpetas *server* y *client*, donde el servidor contiene la capa del controlador y el modelo, mientras que el cliente contiene toda la implementación de la interfaz de la aplicación.

Además se ha implementado una capa de servicio en el cliente, con la finalidad de facilitar la lógica entre la vista y el dominio de la aplicación, y que este dominio sea reutilizado por múltiples vistas, implementándose los servicios del estudiante, profesor y usuario general.

En el proyecto tanto los servicios del estudiante como del profesor contiene las funcionalidades propias según el rol, mientras que el servicio de usuario contiene las funcionalidades comunes entre el estudiante y el profesor tomando como ejemplo iniciar sesión y registro.

## 4.2. Estructura de la Arquitectura de la aplicación

En esta sección se tratará la estructura de la aplicación, haciendo una división clara entre la capa del cliente y el servidor.

### 4.2.1. Backend

A continuación, se muestra la organización del servidor en la aplicación.

**Routers:** La carpeta *routers* contiene las rutas del servidor, donde reciben las solicitudes *http* que se envían desde la capa del cliente.

**Controllers:** El directorio *controllers* contiene los controladores de la aplicación para cada rol. Cuando un cliente ejecuta una acción en la aplicación, el controlador recibe la solicitud del cliente a través de *routers*.

Una vez que el controlador haya recibido la petición, es la encargada de enviarla al modelo. Esta capa es la encargada de enviar una respuesta al servicio del cliente, si la petición se ha ejecutado correctamente o en caso de que haya ocurrido un error.

**Models:** La carpeta *model* es el modelo de la aplicación, aquí se ejecutará todas las operaciones lógicas de la base de datos.

Una vez realizada la acción, y ejecutada las operaciones en la base de datos, será la encargada de devolver una respuesta al controlador. Si ocurriese un error al manipular los datos en *SQL*, mediante las funciones de *Callback* se informaría del error al controlador.

**Middleware:** El directorio *Middleware* contiene el fichero *verifySingup*, cuya función es de comprobar si existe un *email* duplicado al registrarse un usuario por primera vez en la aplicación, y *Auth-jwt.js* verifica el token del usuario en la base de datos, cuando el usuario inicia sesión.

**Database:** Contiene las credenciales que se emplean para obtener acceso a la base de datos *SQL*.

**Utils:** El directorio *utils* contiene el *middleware multer*, aquí se hace la gestión del almacenamiento de las carpetas de los usuarios de la plataforma.

### 4.2.2. Frontend

Aquí se muestra la organización del lado del cliente en la aplicación.

**Components:** La carpeta *componentes* contiene las componentes de la aplicación como las páginas de cada vista según el rol del usuario administrador, profesor y estudiante. También contiene otros tipos de componentes como el *sidebar* y *apis* integradas en la aplicación.

**Links:** El directorio *links* contiene las redirecciones del *sidebar* a vistas de la plataforma.

**Routes:** La carpeta *routes* contiene las rutas asociadas a componentes, cuando el cliente navega por la interfaz para acceder a las vistas de la aplicación.

**Services:** El directorio *services* es la encargada de gestionar las peticiones del usuario del lado del cliente. Cuando un usuario ejecuta una solicitud desde la vista de una componente, *services* recibe los parámetros necesarios para ejecutar la acción, y luego a través de Axios se envía al servidor a través de solicitudes *http*.

**Styles:** El directorio *Styles* contiene todas las hojas de estilo usadas en la aplicación.

**Axios:** *Axios* es un *middleware* que se encarga de que las peticiones que se hacen en el cliente lleguen al servidor.

### 4.3. Base de datos

Para el diseño de la base de datos se ha elegido un modelo de base de datos relacional, se ha utilizado *MySQL* como gestor de base de datos, que viene incluido dentro del *XAMPP* como interfaz de usuario. Este entorno ha permitido una mejor gestión de los datos de la aplicación durante el proceso de desarrollo.

Para la configuración el nombre de la base de datos es “EscrituraCreativa”. Se ha optado por un modelo de datos relacional en el proyecto, para una mejor estructuración y consistencia de los datos, dado que no se espera que el volumen de datos sea masivo, al tratarse en un marco entre un grupo de profesores y estudiantes.

Otra razón es que las operaciones de la base de datos relacionales son atómicas, es decir que en la aplicación existen varias tablas conectadas entre si, y cualquier operación que se quiera ejecutar, y no cumpla con ciertos criterios de información, no se realizará, y si existen nuevos cambios, estos no se aceptan hasta que una transacción no termine.

En el siguiente capítulo 5 Modelo de datos de la memoria, se explica más detalladamente el proceso de desarrollo de la base de datos aplicados en el proyecto.

### 4.4. Autenticación basada en tokens

En el proyecto la autenticación está basada en el uso de *tokens*, empleando la biblioteca *JWT Token*, que es un protocolo que permite a los usuarios de la aplicación verificar su identidad mediante un token de acceso único.

Se ha elegido este protocolo de autenticación para reducir la carga de trabajo en el servidor, ya que no es necesario guardar la información de los usuarios conectados en el servidor porque la verificación de autenticidad de la solicitud se hace a través del *token*.

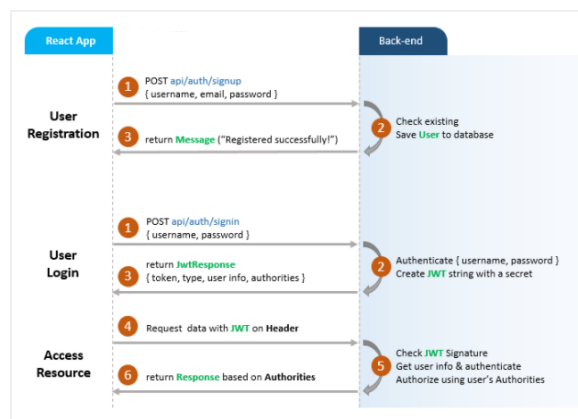


Figura 4.1: Esquema de autenticación basada en token

En la siguiente figura 4.1 se muestra un ejemplo del proceso de autenticación entre el navegador y el servidor basado en *token*

#### 4.4.1. Funcionamiento del *Token JWT* en la aplicación

El profesor o estudiante ingresa sus credenciales en la aplicación, después el servidor comprueba que la información enviada del usuario sean válidas.

Si las credenciales son válidas devuelve un *token* firmado, después el *token* es guardado en el *local storage* del lado del cliente.

Cabe decir que las peticiones que se hacen al servidor incluyen este *token* (a través de un *Authorization header*). Después el servidor decodifica el *JWT* y si el *token* es válido procesa la solicitud. Cuando el usuario finaliza su sesión, el *token* es destruido en el lado del cliente.

## 4.5. Funcionalidades Principales

En este apartado se muestra las principales funcionalidades de la aplicación.

### 4.5.1. Usuarios

En esta sección se definen las funciones comunes que tienen los usuarios.

- Iniciar sesión: Un usuario puede iniciar sesión para entrar en la plataforma web siempre que los datos sean válidos, y esto se hace rellenando los campos del formulario de usuario y contraseña.
- Registro: Los estudiantes pueden registrarse en la aplicación rellenando los siguientes campos del formulario: nombre,apellidos, correo y contraseña. Una vez que el estudiante

se haya registrado en la plataforma, pasara a una lista de solicitantes que permanecerá hasta que profesor le dé de alta en su grupo.

- Consultar Perfil: Los usuarios tanto profesores como estudiantes pueden consultar su perfil.
- Editar Perfil: Los usuarios pueden modificar cualquier campo como nombre,apellidos, correo y contraseña y también pueden gestionar la foto de su perfil.

### 4.5.2. Desafíos

En el contexto de la aplicación los desafíos se definen como retos narrativos que invitan a los estudiantes de un grupo a crear nuevos escritos que cumplan ciertas características o requisitos. Las funciones principales del desafío son:

- Crear desafío: Un profesor puede crear un desafío, que consiste en la elaboración de un reto narrativo dirigido a estudiantes, compuesto por los siguientes campos: título, descripción, categoría, tipo de desafío (individual o colaborativo), ficheros multimedia(imagen, audio o vídeo), y una fecha.
- Editar desafío: Un profesor puede editar un desafío,todos los campos del desafío pueden ser modificados: título, descripción, categoría, tipo de desafío, fecha, al igual que los ficheros multimedia se pueden agregar o eliminar ficheros anteriores
- Listar Desafíos: El profesor o estudiante pueden listar cualquier tipo de desafíos (individuales o colaborativos). Se pueden mostrar todos los desafíos de todos los grupos o los desafíos de un grupo en concreto. Al listar los desafíos mostrará los siguientes campos: título, grupo, categoría, tipo, fecha, hora.
- Eliminar Desafío: El profesor puede eliminar un desafío si es el propietario del desafío, se ha considerado la desactivación del desafío para la persistencia de datos

### 4.5.3. Escritos

Un escrito consiste en un relato en texto que sigue las normas o reglas propuestas del desafío. Las funciones principales del escrito son:

- Crear Escrito: Un estudiante puede crear un escrito individual o colaborativo dependiendo del tipo de desafío, que consiste en la elaboración de un relato narrativo. Se distinguen dos tipos de escritos:
- Escrito individual: un estudiante de un grupo puede crear un escrito individual correspondiente a ese grupo
- Escrito colaborativo: es necesario que el estudiante disponga de un equipo para crear un escrito colaborativo de un grupo, una vez que haya sido creado, todos los miembros del equipo pueden ver y acceder el escrito creado.

Un escrito está compuesto de dos secciones, una correspondiente al desafío y la otra sección al escrito.

La sección del desafío contiene título, categoría, descripción y ficheros multimedia del desafío, y la sección del escrito incluye título, descripción, y ficheros multimedia del escrito. El estudiante puede editar los campos correspondientes a la sección del escrito.

- **Editar Escrito:** El estudiante puede editar el escrito, cuando el escrito es colaborativo, solo los estudiantes que pertenezcan al equipo pueden editar el escrito; todos los campos pueden ser modificados correspondientes a la sección del escrito: título, descripción, y ficheros multimedia.

El profesor también puede editar todos los campos de la sección del escrito y también puede agregar comentarios.

- **Listar Escrito:** Los estudiantes pueden listar todos sus escritos correspondiente a un desafío
- **Consultar Escritos:** Un estudiante puede consultar los escritos para editar o cuando los escritos están finalizados, pudiendo ver los comentarios del profesor.

#### 4.5.4. Grupos

Los grupos son creados por el profesor y se define en la aplicación como el conjunto de estudiantes que pertenecen a un grupo.

- **Crear grupo:** El profesor puede crear un grupo asignándole un nombre.
- **Editar grupo:** El profesor puede editar el grupo modificando el nombre del grupo.
- **Listar grupo:** El profesor puede listar todos sus grupos creados.
- **Agregar estudiante a un grupo:** El profesor puede agregar estudiante a un grupo.
- **Quitar estudiante a un grupo:** El profesor puede quitar estudiante de un grupo.
- **Consultar estudiante:** El profesor puede consultar los datos de información del estudiante, así como los escritos hechos por el estudiante.

#### 4.5.5. Equipos

La idea principal de la aplicación es la participación de los estudiantes de forma colaborativa en la escritura, por ello unos de los requisitos es la formación de equipos para que participen conjuntamente de forma activa en el desarrollo del escrito, las funcionalidades principales son las siguientes:

- **Crear Equipo:** Los estudiantes pueden crear un equipo si no pertenecen a otro equipo en un grupo, asignando un nombre al equipo.
- **Listar Equipos:** Lista los equipos de un grupo, mostrando nombre, integrantes y propietario del equipo
- **Eliminar Equipo:** Si el estudiante es el propietario del equipo puede eliminar el equipo.

- Expulsar estudiante de un equipo: Si el estudiante es propietario del equipo puede expulsar a un integrante del equipo, pudiendo seleccionar a uno de los integrantes a expulsar.
- Listar Equipos: Se puede listar todos los equipos de un grupo.

#### 4.5.6. Mensajería

La mensajería se ha incluido en la aplicación en los estudiantes, con el objetivo de facilitar la gestión de los equipos y quitar carga de trabajo al profesor, siendo empleados los mensajes por parte de los estudiantes para invitar o enviar solicitudes para formar parte de un equipo.

- Enviar solicitud: Si el estudiante no dispone de equipo, puede enviar una solicitud al propietario del equipo para unirse al equipo
- Invitar estudiante a un equipo: Si el estudiante es el propietario del equipo siempre que no exceda de un máximo de cinco integrantes puede invitar a otro estudiante sin equipo, el estudiante propietario puede seleccionar de una lista de estudiantes sin equipo y enviar una invitación.
- Consultar mensaje: El estudiante puede consultar los mensajes.
- Eliminar mensaje: El estudiante puede eliminar los mensajes recibidos.

#### 4.6. Prototipo

Al principio de las primeras reuniones surge la exploración de ideas y requisitos principales a desarrollar, logrando una visión de un primer boceto como apoyo a la estructura visual de la aplicación. Durante las semanas posteriores se inicia un proyecto en Balsamiq Cloud como herramienta de apoyo y de forma colaborativa donde se plasma el primer diseño, ya que, su objetivo es crear una imagen visual a partir de bocetos rápidos como si se tratará de un dibujo donde los diseñadores pueden remover objetos sin tachar o crear un nuevo lienzo.

Asimismo, se trazan los esquemas de las ventanas principales de cada tipo de actor y su rol para concretar así el comportamiento individual de los usuarios. Dichas ventanas, contienen botones de redirección a otra páginas donde se carga la funcionalidad deseada. Este flujo de conducta se agrega con la ayuda de la herramienta para observar la relaciones entre distintos eventos. Por ejemplo, si el profesor deseaba crear un desafío seleccionaba el botón y aparecía una vista con un editor de texto (ver la figura 4.2) o si deseaba observar una lista de estudiantes elegía el botón correspondiente y se dirigía una ventana donde se mostraba un listado de estudiantes (ver la figura 4.3).

De forma análoga, los esquemas era presentados y debatidos con el cliente para obtener retroalimentación. No obstante, existía para este primera estructura una falta de usabilidad eficaz sobre la experiencia del usuario, un alto coste de movilidad entre las distintas ventanas por la carencia jerárquica de las funcionalidades, el desarrollo en paralelo de la base causaba cambios constantes y una interfaz sencilla y pobre de estética.

Por lo tanto, para mejorar nuestro primer prototipo se opta por diseñar componentes visuales reutilizables para todos roles de usuarios. A continuación entender si cada cambio o agregación era satisfactorio y cómodo, y reorganizar los elementos principales mediante cuadrículas para adaptarse a diferentes pantallas.

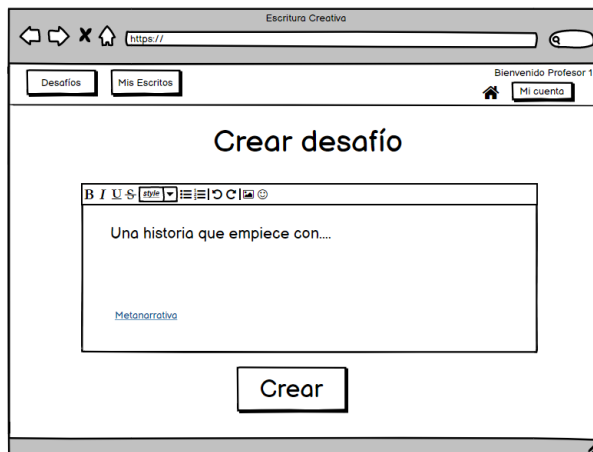


Figura 4.2: Crear desafío (primer boceto)

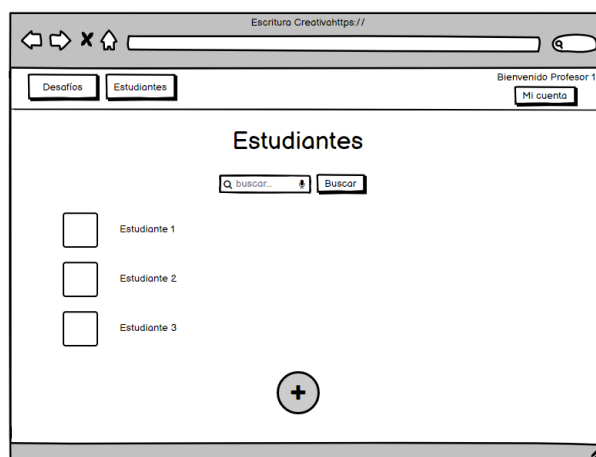


Figura 4.3: Listar estudiantes (primer boceto)



# Capítulo 5

## Modelo de datos

En este capítulo se detalla la estructura del modelo de datos que se emplea en la capa de integración de la aplicación *Creativa*.

### 5.1. Modelo entidad-relación

En la figura 5.1 se muestra el diagrama Entidad Relación del modelo de datos de *Creativa*, mostrando las relaciones que existen entre las diferentes entidades que lo forman y sus atributos.

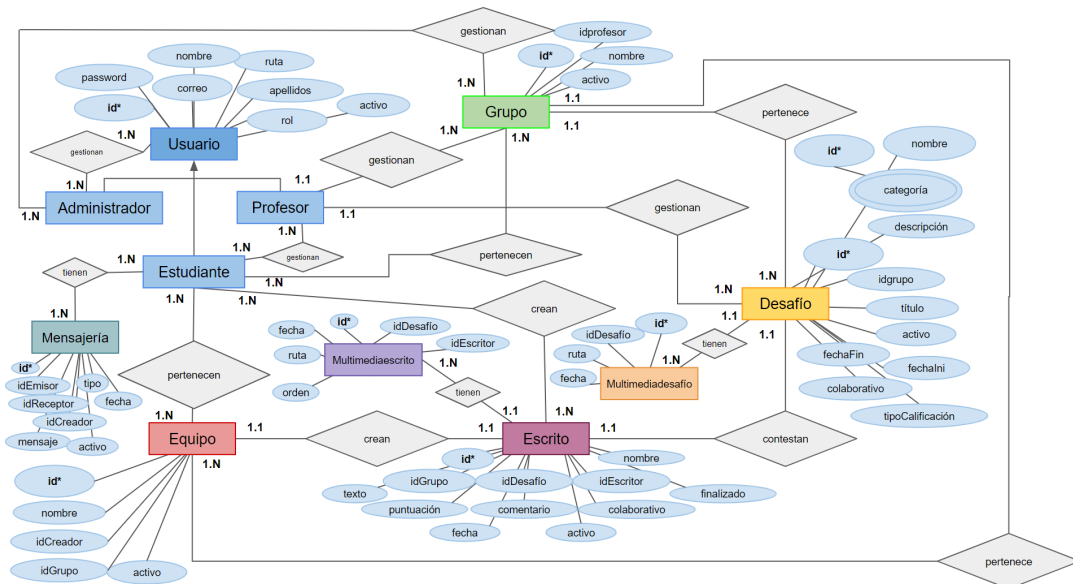


Figura 5.1: Modelo entidad relación

A continuación, se desarrolla el diagrama anterior así como las relaciones que representa:

**Usuario:** Consiste en los usuarios de la aplicación. Se divide en tres roles, identificados por el atributo “rol”, siendo estos profesor, estudiante y administrador. Esta entidad contiene datos personales tales como nombre, apellidos y correo electrónico usado

para el registro, y de la misma forma los atributos empleados para el funcionamiento de la aplicación, como la contraseña, el rol y la ruta que dirige a la imagen de perfil del usuario. El correo de registro es único y no se puede crear más de un usuario por dirección de correo. El atributo “activo” permite identificar si un usuario ha sido aceptado por un administrador y por tanto, si se le permite acceso a la plataforma.

**Grupo:** Esta entidad recoge los datos de los grupos. Creados por los profesores, agrupan a diferentes estudiantes dependiendo de sus intereses creativos. Además de un identificador, cuentan con el identificador del profesor que lo creó, así como el nombre del grupo y el campo activo, que identifica si el grupo está aún en uso o no.

**Desafío:** Contiene toda la información referente a los desafíos literarios. Creados por los profesores, pertenecen a un grupo concreto. Pueden ser accedidos por los estudiantes, quienes pueden crear escritos para enfrentarse al reto presentado. Cuentan con su propio identificador, así como el identificador del grupo al que pertenecen. Tienen la fecha inicio y la fecha del fin del intervalo en el que el desafío está activo. Contienen un título, así como una descripción que detalla las normas del reto y además si el desafío es colaborativo, es decir, si puede ser enfrentado por un equipo o por estudiantes individuales. Especifica la forma de calificación designada por el profesor, que sirve para fijar el formato en el que se dará respuesta a los escritos de ese desafío. Añade un identificador que indica si es visible y accesible o no.

**Escrito:** Son los textos literarios creados por los estudiantes como contestación a un desafío. Contiene su propio identificador, así como aquellos de su escritor o equipo creador, el grupo en el que fue creado y el desafío al que se enfrenta. Cuenta con un nombre a modo de título, así como el texto en el que consiste el escrito. Contiene la información sobre si es colaborativo o no. Añade también la puntuación otorgada por el profesor, así como los posibles comentarios que haya podido hacer el mismo sobre el contenido del escrito así como los datos sobre la última fecha de edición y de un identificador que indica si es activo y visible o no. Un estudiante o equipo solo puede crear un escrito por cada desafío.

**Equipo:** Formados por los estudiantes, los equipos son formaciones que se enfrentan a desafíos colaborativos, creando escritos en conjunto. Pertenecen a un único grupo y como tal, solo pueden contener estudiantes de ese mismo grupo. Incluye un identificador unívoco, así como el identificador del grupo al que pertenecen y el identificador del creador, o líder del equipo. Contiene también información sobre el nombre del equipo y de si se encuentra activo y utilizable o desactivado.

**Mensajería:** Esta entidad representa los mensajes que pueden enviar los estudiantes. Su función principal es la de asistir en la creación de equipos colaborativos e incluye un sistema de invitaciones para el mismo.

**Multimediaesafío:** Esta entidad identifica los enlaces a archivos multimedia que pertenecen a un desafío. Los archivos que son referenciados por estos enlaces residen en el almacenaje del servidor.

**Multimediaescrito:** Esta entidad representa los enlaces a archivos multimedia que pertenecen a un escrito. Contiene una manera de identificar el orden en el que estos archivos se añaden para facilitar la creación de obras que consisten en más de un archivo, como las novelas visuales.

Entre estas entidades existen relaciones que tienen ciertas limitaciones como parte de su diseño:

- Un profesor puede gestionar varios grupos, pero un grupo solo puede ser gestionado por un profesor.
- Un profesor puede gestionar múltiples desafíos, pero un desafío solo puede ser gestionado por un profesor.
- Cada estudiante puede pertenecer a varios grupos y cada grupo puede contener a varios estudiantes.
- Cada estudiante puede crear varios escritos, pero cada escrito solo puede ser creado por un estudiante.
- Cada estudiante puede pertenecer a más de un equipo y un equipo puede contener más de un estudiante.
- Cada escrito solo puede contestar a un desafío. Cada desafío puede ser contestado por múltiples escritos.
- Varios administradores pueden gestionar a varios usuarios y varios usuarios pueden ser gestionados por varios administradores.
- Varios profesores pueden gestionar a varios estudiantes y varios estudiantes pueden ser gestionados por varios profesores.
- Un desafío puede tener varios archivos multimedia relacionados. Un archivo multimedia puede estar relacionado solo con un desafío a la vez.
- Un escrito puede tener varios archivos multimedia relacionados. Un archivo multimedia puede estar relacionado solo con un escrito a la vez.
- Un estudiante puede tener varios mensajes a la vez. Cada mensaje solo está relacionado con su receptor y su remitente.

## 5.2. Base de datos SQL

Se ha empleado SQL para la creación de la base de datos. Este sistema nos permite crear bases relacionales sujetas a ciertas normas, así como crear relaciones entre las diferentes tablas. Se pueden poner limitaciones a los atributos de las mismas, como establecer los campos identificadores como únicos. SQL permite también establecer comportamientos automáticos dentro de la base de datos, por ejemplo, eliminar todos los datos referidos a un usuario si este es eliminado, de manera que no queda información perdida e inaccesible en la base de datos al realizar operaciones sobre la misma. De esta manera también se pueden generar restricciones, como detectar automáticamente que no existan repeticiones en campos únicos o incluso generar identificadores nuevos de forma automática al crear una fila en una tabla. El lenguaje también posibilita crear tablas intermedias, estructuras auxiliares que encarnan ciertas relaciones entre tablas que tengan atributos propios. En la mayor parte de las tablas se ha empleado un atributo llamado “activo” que representa el estado de un elemento. Si se desea eliminar un elemento, en lugar de borrarlo de la base de datos, se desactiva, haciendo que se le ignore por parte de la lógica de

la aplicación, no obstante, esta decisión se ha tomado para tener una persistencia de los datos y que en ningún caso se pueda perder información valiosa. Se desarrolla a continuación la estructura de cada una de las tablas, así como la utilidad de cada uno de sus atributos.

### 5.2.1. Usuario

Esta entidad representa a todos los usuarios registrados. Estos tienen disponibles una serie de funciones u otras dependiendo de su rol. Un usuario solo puede acceder a la plataforma si ha sido aceptado en la aplicación. Esta tabla contiene los siguientes atributos:

**id:** Campo numérico que identifica de manera unívoca a un usuario. Se genera automáticamente cuando se crea el perfil y no puede haber dos usuarios con el mismo identificador. Es la clave primaria de la entidad.

**correo:** Email provisto a la hora de crear el perfil. Es único y no puede haber dos usuarios con el mismo correo. Se emplea para iniciar sesión como usuario y puede ser utilizado como clave de búsqueda para profesores y administradores.

**password:** Contraseña de la cuenta del usuario. Se encuentra encriptada para mayor seguridad. Se necesita introducir junto con el correo electrónico para poder iniciar sesión.

**nombre:** Nombre del usuario. Puede haber más de un usuario con el mismo nombre. En ciertos listados puede usarse para filtrar los distintos usuarios.

**apellidos:** Apellidos del usuario. Puede haber más de un usuario con los mismos apellidos. En ciertos listados pueden usarse para filtrar y buscar entre los distintos usuarios.

**activo:** Campo que indica si un usuario está aceptado en la plataforma o no. Si no lo está, no podrá iniciar sesión en la aplicación. Este campo puede ser manipulado por profesores y administradores.

**rol:** Este campo sirve para identificar qué privilegios tiene un usuario. Puede designar a un usuario como estudiante, profesor o administrador.

**ruta:** Contiene la ruta de la imagen de perfil del usuario, que se guarda en el almacenamiento del servidor.

### 5.2.2. Grupo

Esta entidad representa a los conjuntos de estudiantes organizados por profesores en torno al interés en un tema en común. Cada grupo es creado y gestionado por un único profesor, que puede añadir a estudiantes al mismo. Una vez el grupo tiene estudiantes, el profesor puede proponer desafíos exclusivamente a los integrantes de ese grupo. Los administradores pueden manipular cualquier grupo sin importar su creador, así como los estudiantes que pertenecen al mismo. Esta tabla contiene los siguientes atributos:

**id:** Este campo identifica de forma unívoca a cada grupo y se genera automáticamente cuando se crea un grupo nuevo. No pueden existir dos grupos con el mismo identificador. Es la clave primaria de la entidad.

**idprofesor:** Identificador del profesor que creó el grupo.

**nombre:** Nombre que identifica al grupo. Pueden existir varios grupos con el mismo nombre. Puede ser cambiado en cualquier momento por el profesor que lo organiza.

**activo:** Este campo identifica si el grupo sigue vigente o si ha sido desactivado y por tanto no puede utilizarse.

### 5.2.3. Desafío

Esta entidad representa los retos que puede proponer un profesor a los estudiantes de un grupo. Su contenido puede ser visto y editado por el profesor que los creó y solo puede ser accedido por los estudiantes del grupo al que pertenece. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco empleado para discernir entre desafíos. Dos desafíos no pueden tener el mismo identificador y este se genera automáticamente al crear el desafío en la base de datos. Es la clave primaria de esta entidad.

**idGrupo:** Es el identificador del grupo al que pertenece. Un desafío solo puede pertenecer a un grupo.

**título:** Contiene el título del reto. Varios desafíos pueden tener el mismo título.

**descripción:** Texto que contiene todas las pautas que han de seguirse para enfrentarse al reto. Dos desafíos pueden tener la misma descripción.

**tipoCalificación:** Define el tipo de respuesta que puede dar el profesor a la hora de calificar un escrito que conteste al desafío. Puede ser numérica o en forma de comentarios.

**idCategoría:** Identificador de la categoría a la que pertenece el desafío. Las categorías permiten saber a qué género literario pertenece el resto y pueden ayudar a guiar al estudiante a crear un escrito adecuado.

**colaborativo:** Este atributo permite saber si un desafío es colaborativo y por tanto puede ser enfrentado por equipos, o individual, y solo puede ser enfrentado por estudiantes en solitario.

**fechaini:** Contiene la fecha de inicio del desafío. Ayuda a ver qué intervalo de tiempo ha estado abierto el desafío a respuestas.

**fechafin:** Contiene la fecha de cierre del desafío. una vez pasado el plazo de tiempo, el desafío se considera cerrado.

**activo:** Esta variable indica si el desafío está activo y disponible, o desactivado y por lo tanto inaccesible.

### 5.2.4. Escrito

Esta entidad representa los textos literarios que pueden crear los estudiantes para enfrentarse a los desafíos propuestos por los profesores. Pueden ser editados múltiples veces por

los estudiantes. De la misma manera, los profesores pueden revisar sus contenidos y dar una respuesta en formato de calificaciones o comentarios. Al igual de los desafíos, pueden ser tanto individuales, creados por estudiantes, o colectivos, creados por equipos de estudiantes. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la entidad. Se genera automáticamente al crear un nuevo escrito en la base de datos. Dos escritos no pueden tener el mismo identificador. Es la clave primaria de la entidad.

**idGrupo:** Identificador del grupo al que pertenece el escrito. Un escrito solo puede pertenecer a un grupo.

**idDesafío:** Identificador del desafío al que contesta. Un escrito solo puede enfrentarse a un desafío y a la vez, un escrito siempre debe ser creado para un desafío.

**idEscritor:** Identificador del estudiante que creó el escrito. Si se establece que es un escrito colaborativo, este identificador representa el del equipo que lo creó. Un escrito solo puede pertenecer a un estudiante o equipo.

**nombre:** Título del escrito. Puede haber múltiples escritos con el mismo nombre. Puede ser usado para filtrar en búsquedas.

**texto:** Contenido del escrito. Puede haber escritos con el mismo texto.

**puntuación:** Calificación otorgada por un profesor sobre el escrito.

**comentario:** Respuesta y críticas hechas por un profesor que corrige el escrito.

**colaborativo:** Este atributo identifica si el escrito pertenece a un desarrollo colaborativo, y por tanto pertenece a un equipo, o si es individual y pertenece a un estudiante en solitario.

**finalizado:** Este atributo indica si el escrito está terminado y listo para corregir.

**fecha:** Este atributo indica la fecha de actualización del escrito.

**activo:** Este atributo indica si el escrito desafío está activo y disponible, o desactivado y por lo tanto inaccesible.

### 5.2.5. Equipo

Esta entidad representa los equipos organizados y formados por estudiantes para enfrentarse a desafíos en un grupo dado. Cada estudiante sólo puede estar en un equipo de cada grupo y este equipo siempre ha de tener un líder. Si el líder del equipo desea abandonar el equipo y aún quedan miembros en él, debe elegir un sucesor antes de abandonarlo. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la entidad. Se genera de forma automática cuando un nuevo equipo es creado en la base de datos. Dos equipos no pueden tener el mismo identificador. Es la clave primaria de la entidad.

**nombre:** Es el nombre del equipo. Dos equipos pueden tener el mismo nombre sin importar el grupo al que pertenecen.

**activo:** Es el atributo que representa si un equipo es accesible y visible o no.

**idCreador:** Es el identificador del líder actual del equipo. Cada equipo solo puede tener un líder y siempre ha de tener uno. Si el líder de un equipo desea abandonarlo y quedan más miembros en él, debe escogerse un nuevo líder primero.

**idGrupo:** Este atributo identifica al grupo al que pertenece el equipo. Un equipo solo puede pertenecer a un grupo.

### 5.2.6. Categoría

Esta entidad representa un enumerado de categorías a las que puede pertenecer un desafío de manera que pueda ser clasificado según género literario. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la categoría. No pueden existir dos categorías con el mismo nombre.

**nombre:** Contiene el nombre de la categoría.

### 5.2.7. Equipoestudiante

Esta entidad representa la relación que existe entre los estudiantes y los equipos a los que pertenecen. Un estudiante solo puede formar parte de un equipo en cada grupo. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de relación. Es la clave primaria de la entidad.

**idEquipo:** Este atributo contiene el identificador del equipo de la relación.

**idEstudiante:** Este atributo contiene el identificador del estudiante de la relación.

### 5.2.8. Grupoestudiante

Esta entidad representa la relación que existe entre los estudiantes y los grupos a los que pertenecen. Los estudiantes son añadidos a los grupos por los profesores o administradores. Múltiples alumnos pueden formar parte de múltiples grupos. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la relación. Es la clave primaria de la entidad.

**idGrupo:** Este atributo contiene el identificador del grupo de la relación.

**idEstudiante:** Este atributo contiene el identificador del estudiante de la relación.

### 5.2.9. Mensajería

Esta entidad representa la funcionalidad de los mensajes y los datos implícitos en ellos. Estos mensajes se emplean principalmente entre estudiantes para la organización de los equipos de manera que forman un sistema de invitaciones. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de entidad. No puede haber dos mensajes con un mismo identificador. Es la clave primaria de la entidad.

**idEmisor:** Identificador del usuario emisor del mensaje.

**idReceptor:** Identificador del usuario receptor del mensaje.

**idCreador:** Identificador del usuario creador del grupo en caso de que el mensaje consista en la invitación a un equipo.

**mensaje:** Este atributo contiene el texto del mensaje. Dos mensajes pueden tener el mismo mensaje.

**tipo:** Este campo concreta si el mensaje es una invitación o una notificación general.

**fecha:** Este campo contiene cuándo se envió el mensaje.

**activo:** Este campo permite saber si el mensaje es visible o no.

### 5.2.10. Multimediadesafío

Esta entidad representa todos los archivos multimedia de los que dispone un desafío. Puede existir más de un solo archivo por desafío, pero un archivo solo puede pertenecer a un desafío a la vez. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la entidad. No puede haber dos enlaces de multimedia de un desafío iguales. Es la clave primaria de la entidad.

**idDesafío:** Identificador del desafío al que pertenece este archivo multimedia. Un archivo solo puede pertenecer a un desafío cada vez.

**ruta:** Este atributo contiene la dirección exacta del servidor donde se aloja el archivo multimedia.

**fecha:** Este atributo contiene el momento en el que fue creada la asociación entre el archivo y el desafío.

### 5.2.11. Multimediadesafío

Esta entidad representa todos los archivos multimedia de los que dispone un escrito. Puede existir más de un solo archivo por desafío, pero un archivo solo puede pertenecer a un escrito a la vez. Se ha tenido en cuenta que en el caso de la elaboración de elementos narrativos como las novelas visuales es importante mantener un orden entre los diferentes archivos que pertenezcan a un escrito. Esta tabla contiene los siguientes atributos:

**id:** Identificador unívoco de la entidad. No puede haber dos enlaces de multimedia de un escrito iguales. Es la clave primaria de la entidad.

**idEscritor:** Identificador del escritor que ha creado el escrito al que pertenece el archivo multimedia. No puede estar relacionado con más de un escritor.

**idDesafío:** Identificador del desafío al que contesta el escrito. Junto con idEscritor, permite identificar el escrito al que pertenece formando una clave compuesta.



**ruta:** Este atributo contiene la dirección exacta del servidor donde se aloja el archivo multimedia.

**fecha:** Este atributo contiene el momento en el que fue creada la asociación entre el archivo y el escrito.

**orden:** Este atributo numérico permite establecer un orden entre los distintos archivos que pertenezcan a un mismo escrito, que permite crear secuencias de forma intencionada. Esta funcionalidad es muy útil a la hora de crear novelas visuales o distintas viñetas relacionadas de forma cohesiva.

# Capítulo 6

## Guía de uso

Esta sección pretende ser una guía de uso rápido para que los usuarios se familiaricen con las opciones, vistas y el flujo de movilidad dentro de la aplicación tras la navegación por las diversas funcionalidades disponibles de cada usuario.

### 6.1. General

En este apartado se describen los módulos comunes para todos los roles de usuarios así como una descripción sobre el uso de cada uno de ellos.

#### 6.1.1. Módulo iniciar y registrar usuario

La primera vista que tenemos de la aplicación es la ventana de iniciar sesión, como se muestra en la figura 6.1. Esta ventana muestra un campo de texto donde el usuario introducirá su correo y contraseña, elegida previamente durante el alta.

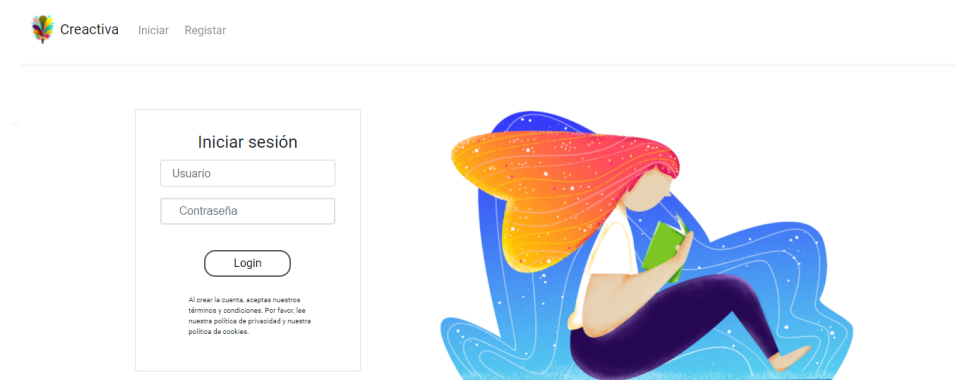


Figura 6.1: Iniciar sesión

En caso de faltar datos en el formulario, se alertará al usuario del error como se ve en la figura 6.2.

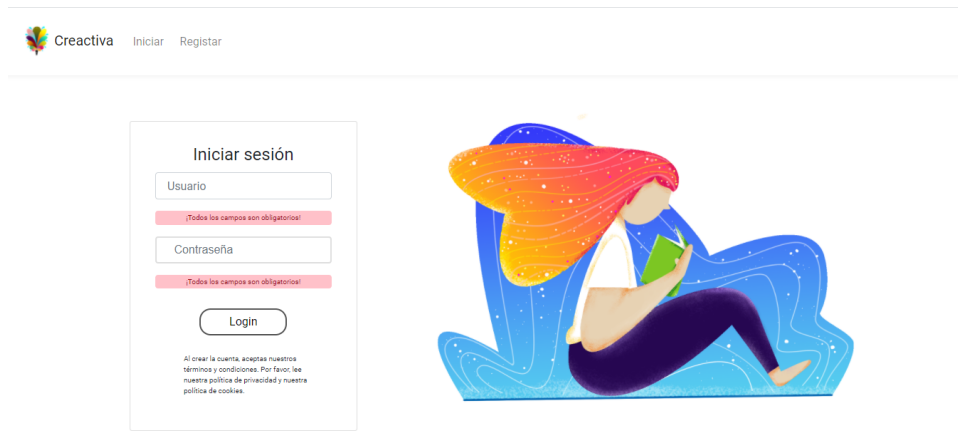


Figura 6.2: Alerta de error

Si bien, puede saltar una alerta en caso de que la cuenta ingresada o contraseña sean inválidos.

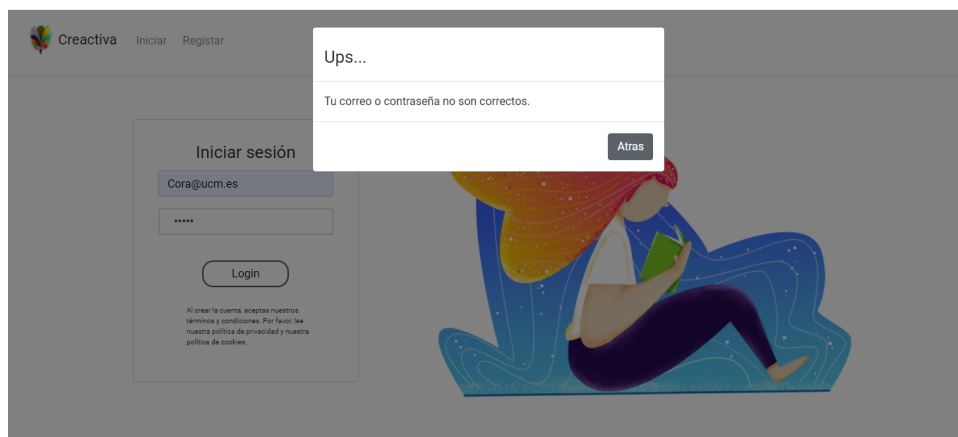
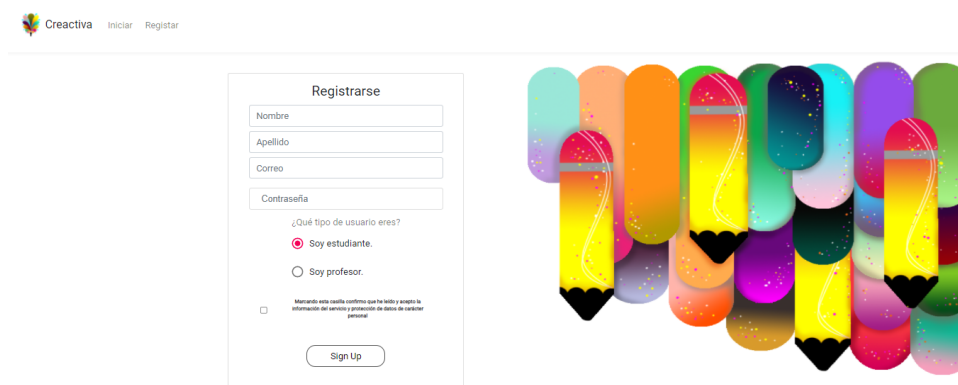


Figura 6.3: Error usuario no válido

Sobre la barra de navegación encontraremos la opción “Registrar”. Una vez seleccionado nos dirigimos a la ventana de registro, como se muestra en la figura 6.4, donde encontramos un formulario con los campos necesarios para ser dado de alta en la aplicación de *Creativa*.

Una vez rellenos el usuario deberá aceptar las condiciones de uso sobre sus datos, esperar a que su cuenta sea confirmada y dada de alta en el sistema.

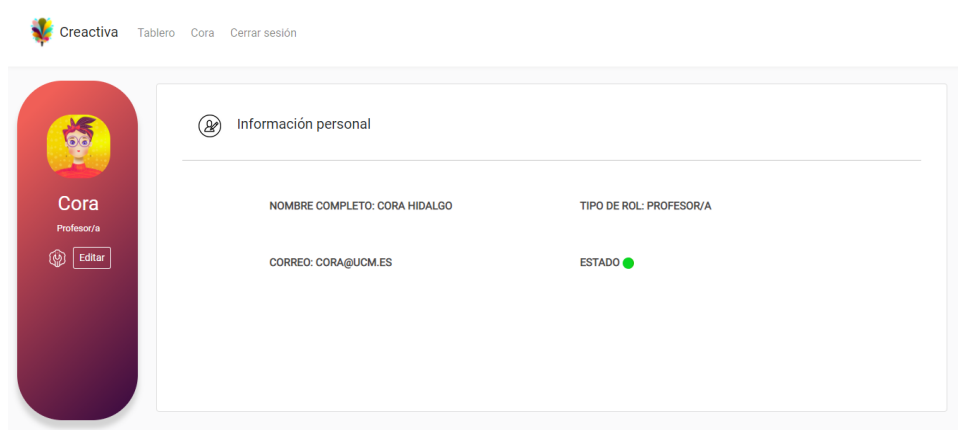


The screenshot shows the registration page of the Creativa application. At the top left, there is a logo and navigation links for 'Inicio' and 'Registrar'. The main content area is titled 'Registrarse' and contains a form with the following fields: 'Nombre', 'Apellido', 'Correo', and 'Contraseña'. Below these fields, there is a question '¿Qué tipo de usuario eres?' with two radio button options: 'Soy estudiante.' (selected) and 'Soy profesor.'. There is also a checkbox for 'Mantener este correo confirmado que he leído y acepto la información del servicio y protección de datos de cualquier personal'. A 'Sign Up' button is located at the bottom of the form. To the right of the form is a decorative graphic of several colorful pencils with glowing tips.

Figura 6.4: registrar usuario

### 6.1.2. Módulo perfil del usuario

Una vez iniciada la sesión nos redirige a la vista “Perfil del usuario”, tal y como se muestra en la figura 6.5, donde podemos observar los datos personales en un contenedor. También se muestra una barra lateral con una fotografía y un botón como opción disponible para modificar dichos datos.



The screenshot shows the user profile page of the Creativa application. At the top left, there is a logo and navigation links for 'Tablero', 'Cora', and 'Cerrar sesión'. The main content area is titled 'Información personal' and contains the following information: 'NOMBRE COMPLETO: CORA HIDALGO', 'TIPO DE ROL: PROFESOR/A', 'CORREO: CORA@UCM.ES', and 'ESTADO' with a green dot. On the left side, there is a vertical sidebar with a profile picture of a woman, the name 'Cora', the role 'Profesora/a', and an 'Editar' button.

Figura 6.5: Perfil del usuario

### 6.1.3. Módulo modificar datos

Después de seleccionar el botón “Editar” en la ventana de “Perfil”, nos redirigimos a una nueva ventana con un formulario. En primer lugar, el usuario puede rellenar aquellos campos que desee modificar y los campos no rellenados se quedarán con los datos ya predeterminados en la base de datos (paso 1 en la figura 6.6).

En segundo lugar, el usuario tendrá las opciones de guardar y cancelar (paso 2 en la figura 6.6). Si selecciona “Guardar” entonces se mostrará un aviso para confirmar sus cambios, y si la opción es “Cancelar” entonces nos redirigimos a la ventana del perfil.

1

MODIFICAR DATOS

NOMBRE: Genesis

APELLIDOS: Duque

CORREO: genesis@ucm.es

FOTO DE PERFIL: Ningún archivo seleccionado

CONTRASEÑA: Nueva contraseña

CONFIRMAR CONTRASEÑA: Confirmar nueva contraseña

2

Guardar Cancelar

Figura 6.6: Modificar datos

En tercer lugar, si el usuario confirma sus cambios (paso 3 en la figura 6.7) la sesión se cerrará y se redirigirá a la ventana de iniciar sesión para nuevamente identificarse. Esto ha sido implementado como medida de seguridad sobre la actualización de los datos.

MODIFICAR DATOS

Aviso

¿Esta seguro de aplicar estos cambios? Deberá iniciar nuevamente sesión de su cuenta

3

Aceptar Cancelar

Figura 6.7: Confirmar modificación

#### 6.1.4. Módulo tablero

Desde la barra de navegación el usuario puede seleccionar la opción “Tablero” (paso 1 en la figura 6.8) el cual llamará a una vista con una barra lateral sobre el lado izquierdo, conteniendo las funcionalidades específicas de rol identificado.

En la parte derecha se cargarán los componentes relacionados con la opción seleccionada desde la barra lateral (paso 2 en la figura 6.8).

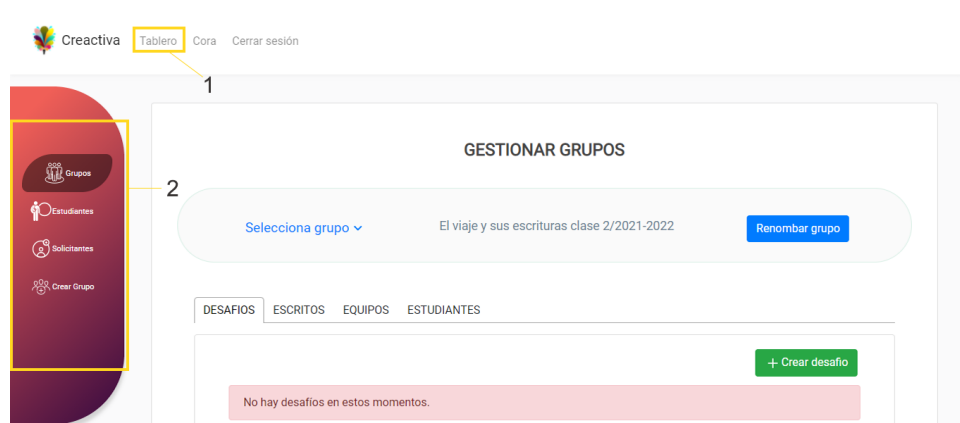


Figura 6.8: Barra lateral de opciones

### 6.1.5. Módulo alertas sobre errores

Estos errores aparecen durante la navegación en la aplicación cuando las solicitudes de los usuarios no son satisfactorias.

**Error 404:** El usuario intenta acceder a un enlace erróneo o una página que no existe, ver la figura 6.9.

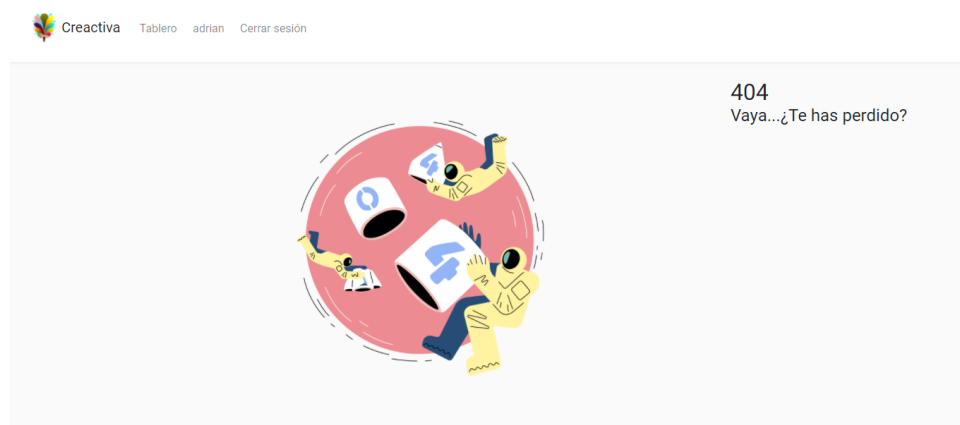


Figura 6.9: Error 404.

**Error 500:** Esto significa que ha sucedido un error interno del servidor, pero no se puede dar mas detalles sobre lo que ha ocurrido (ver la figura 6.10).

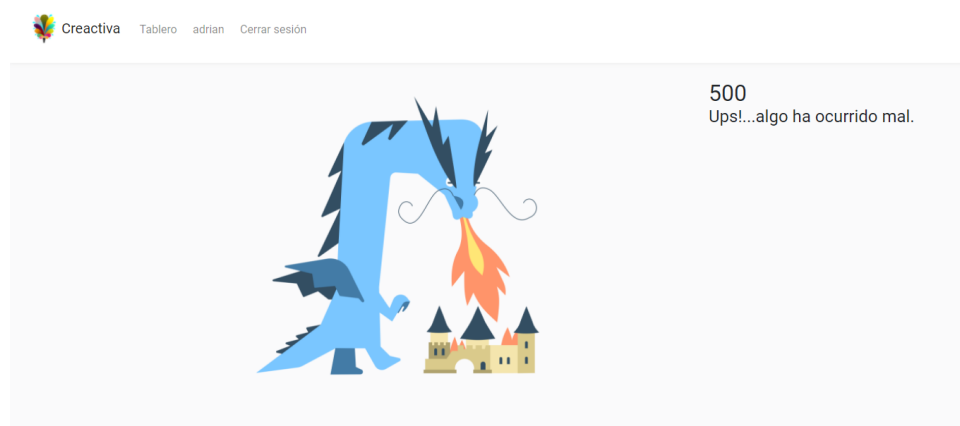


Figura 6.10: Error 500.

## 6.2. Usuario profesor

Uno de los roles más importantes dentro de la aplicación es el del profesor. Entre sus principales funciones se encuentra la gestión de grupos, la corrección y evaluación de desafíos, además de la gestión de las solicitudes de participación de los estudiantes.

### 6.2.1. Módulo grupos profesor

El propósito principal de este módulo es el de ayudar al profesor a gestionar sus grupos con facilidad y rapidez. Al inicio tenemos un espacio de selección donde puede escoger un grupo determinado (paso 1 en la figura 6.11) o un botón para renombrar dicho grupo (paso 2 en la figura 6.11).



Figura 6.11: Selección de grupos y renombrar

Después de elegir un grupo se puede observar una tabla de pestañas desde la cual podemos realizar distintas funcionalidades tales como:

**Desafíos:** Al seleccionar la pestaña “Desafíos”(paso 1 en la figura 6.12) se lista los desafíos ya creados de un único grupo con opciones de editar su contenido o borrar

(paso 2 en la figura 6.12). Además se puede observar un botón “Crear desafío” en la parte superior derecha de la pestaña (paso 3 en la figura 6.12), que al ser pulsado nos permite crear un nuevo desafío.

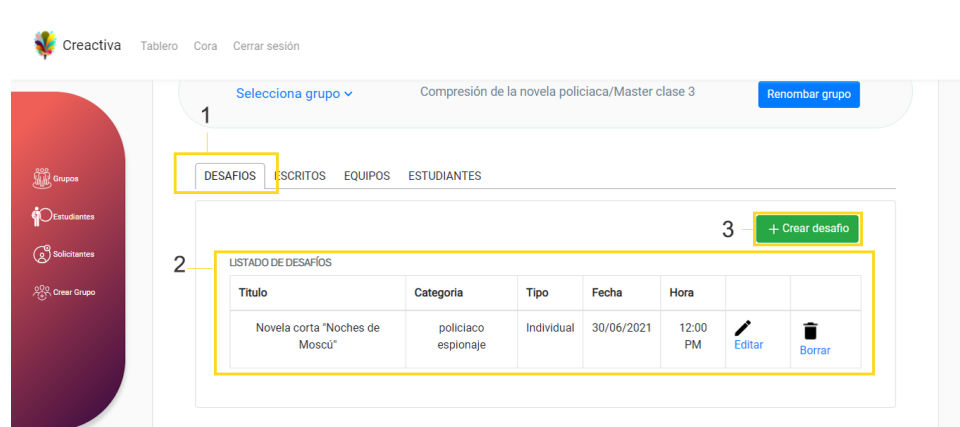


Figura 6.12: Desafíos de un grupo

**Escritos:** Al seleccionar la pestaña “Escritos” (paso 1 en la figura 6.13), el profesor puede elegir un desafío (paso 2 en la figura 6.13) desde el cual se filtrará una lista con todos los escritos realizados por los estudiantes para dicho grupo.

Si el profesor desea agregar comentarios de corrección antes de la fecha fin del desafío puede accionar el botón “Editar Escrito” (paso 3 en la figura 6.13), y luego ser redirigido a la ventana de edición.

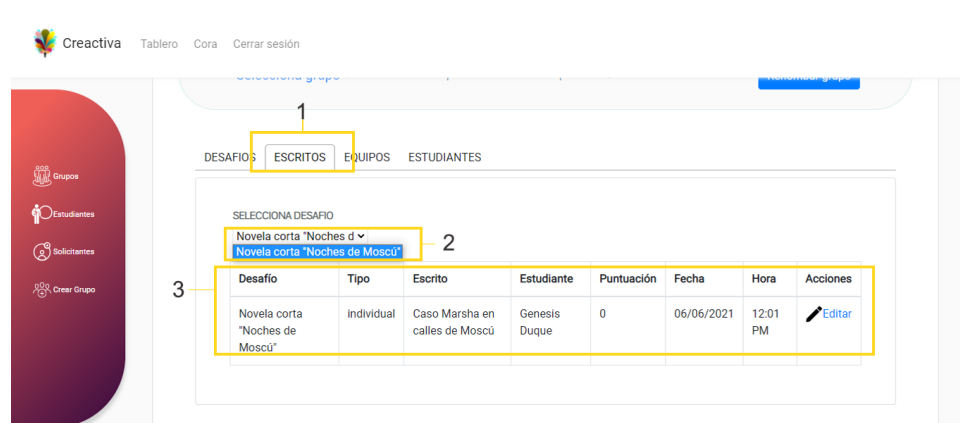


Figura 6.13: Escritos de un grupo

**Equipos:** Al seleccionar la pestaña “Equipos” (paso 1 en la figura 6.14) se lista los equipos del grupo elegido, facilitando la búsqueda a través de un filtro (paso 2 en la figura 6.14). El botón “Ver equipo” una vez accionado redirigirá a una nueva ventana con la información y detalles del equipo (paso 3 en la figura 6.14).



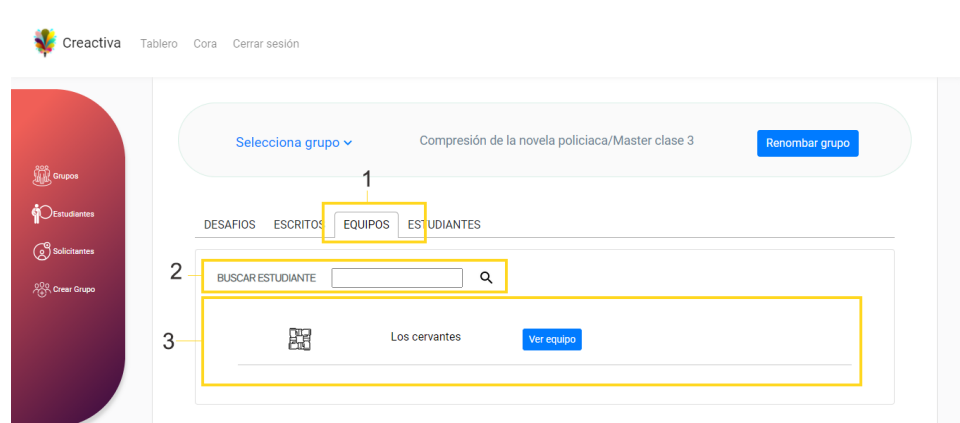


Figura 6.14: Equipos de un grupo

**Estudiantes:** Al seleccionar la pestaña “Estudiantes” (paso 1 en la figura 6.15) se lista los estudiantes del grupo, lográndose filtrar una búsqueda por el nombre o correo de los estudiantes (paso 2 en la figura 6.15). El botón de “Ver perfil” (paso 3 en la figura 6.15) una vez accionado redirigirá a una nueva ventana del perfil del estudiante seleccionado.

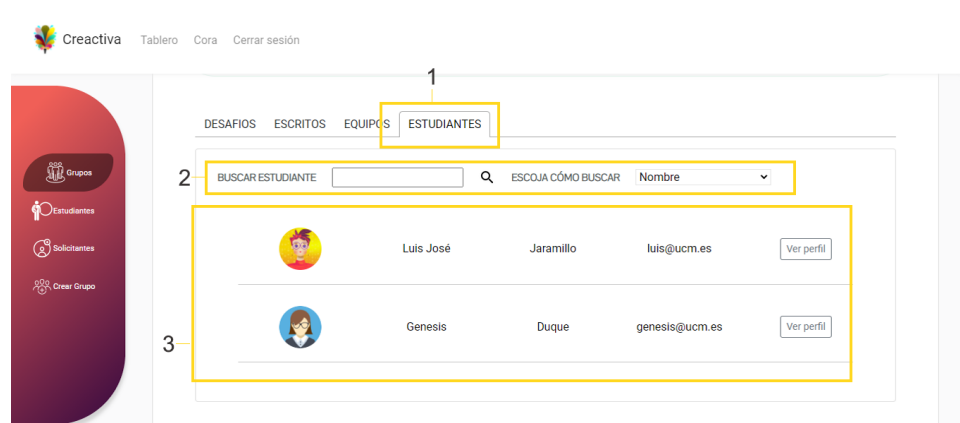


Figura 6.15: Lista de estudiantes

### 6.2.2. Módulo estudiantes

Desde esta sección se muestra una lista con todos los estudiantes dados de alta en la plataforma (paso 1 en la figura 6.51). La búsqueda de un determinado estudiante puede ser ligera a través del filtro agregado, ya sea por nombre o correo electrónico (paso 2 en la figura 6.51). Además, contará con la opción “Ver perfil”, un enlace el cual redirigirá a la página de “perfil del estudiante” (ver figura 6.17 )

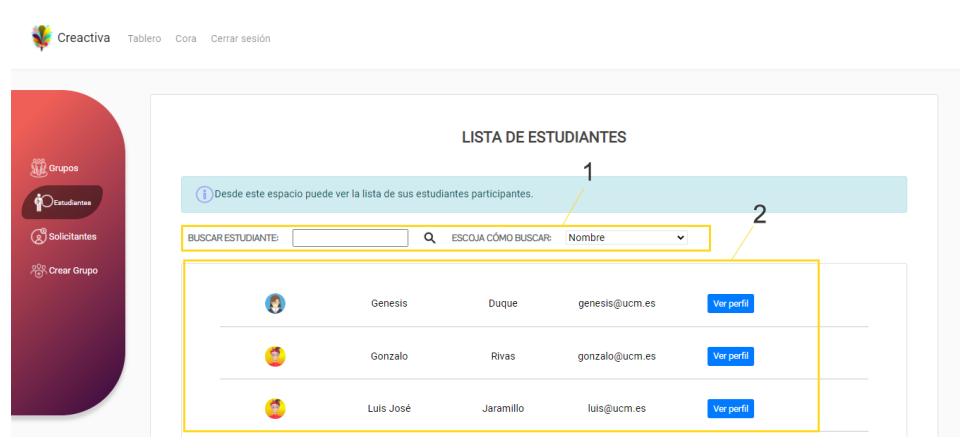


Figura 6.16: Lista de estudiantes

### 6.2.3. Módulo perfil del estudiante

Sobre esta vista el profesor tiene dos pestañas con las siguientes funcionalidades:

**Datos:** Al pulsar en esta pestaña, figura 6.17 (paso 1), se muestran los datos del estudiante seleccionado.

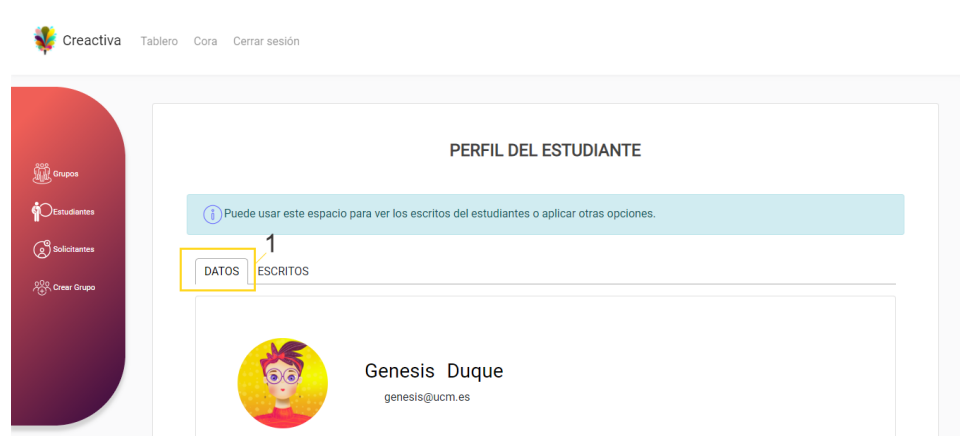


Figura 6.17: Perfil del Estudiante

Luego, desplazando la página se encuentra la opción de “Invitar a un grupo”, donde el profesor puede invitar a dicho estudiante a cualquiera de sus grupos seleccionando uno de ellos, ver figura 6.18 (paso 2).

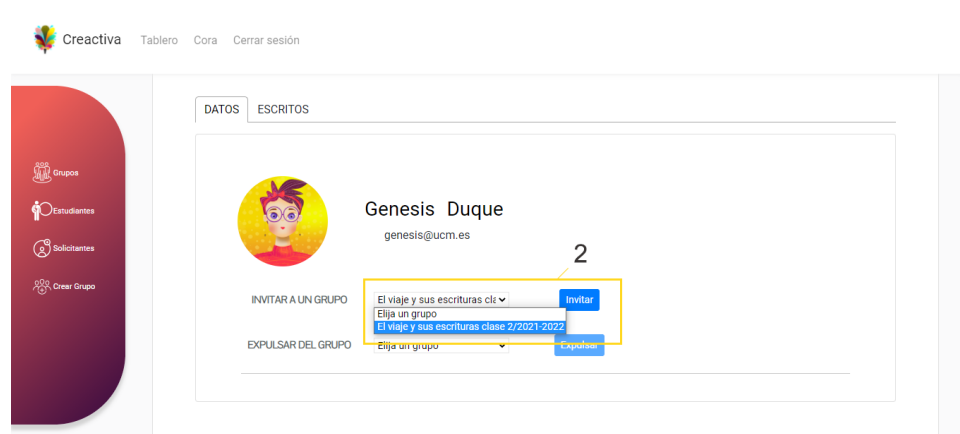


Figura 6.18: Perfil del Estudiante

A continuación, otra de las opciones es “Expulsar del grupo”, cuya acción del profesor es seleccionar uno de los grupos y pulsar “Expulsar” como se puede ver en la figura 6.19 (paso 3).

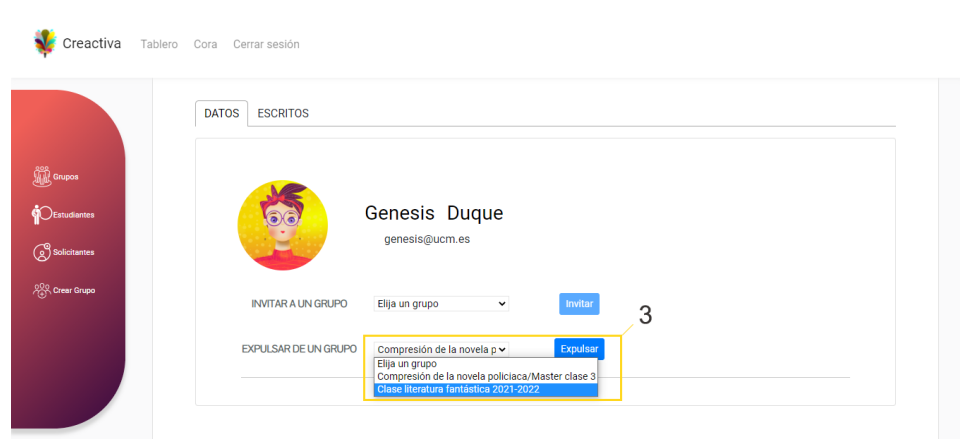


Figura 6.19: Perfil del Estudiante

**Escritos:** Esta pestaña tras ser seleccionada (paso 1 y 3 en la figura 6.20) se muestran unas tarjetas las cuales contienen con el nombre del desafío y del escrito, todas ellas correspondiente al estudiante elegido (paso 2 en la figura 6.20).

Pueden ser filtradas por el nombre del desafío o por título del escrito (paso 2 en la figura 6.20) y al pulsar sobre ellas nos lleva a la ventana de “Editar escrito” como se muestra en la figura 6.26

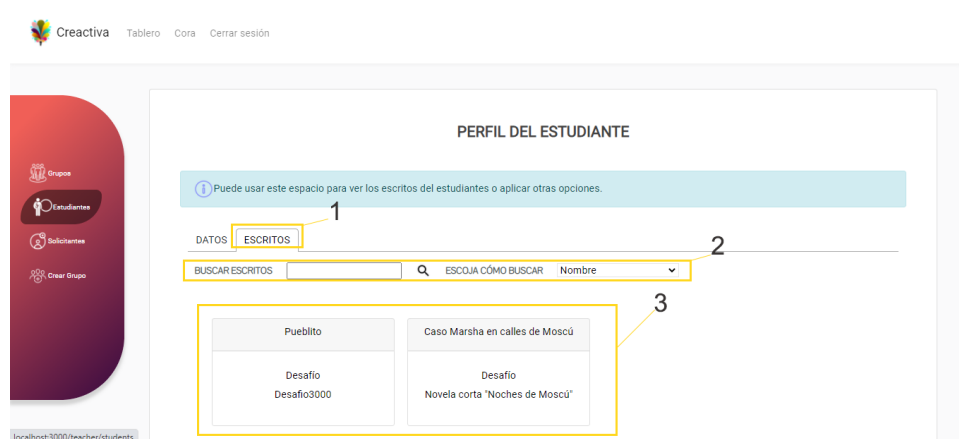


Figura 6.20: Perfil del Estudiante

### 6.2.4. Módulo solicitantes

En esta sección se lista a los estudiantes que han decidido registrarse en la aplicación y cuyas cuentas están a la espera de ser activadas por un profesor. Si algún profesor desea activar a uno de esos estudiantes, solo tiene que buscar usando el filtrado o no para facilitar el evento (pasos 1 y 2 en la figura 6.51). Luego, tras pulsar el botón “Ver perfil”, será dirigido a una nueva vista desde la cual puede aceptar la solicitud tal como se muestra en el paso 3 de la figura 6.51.

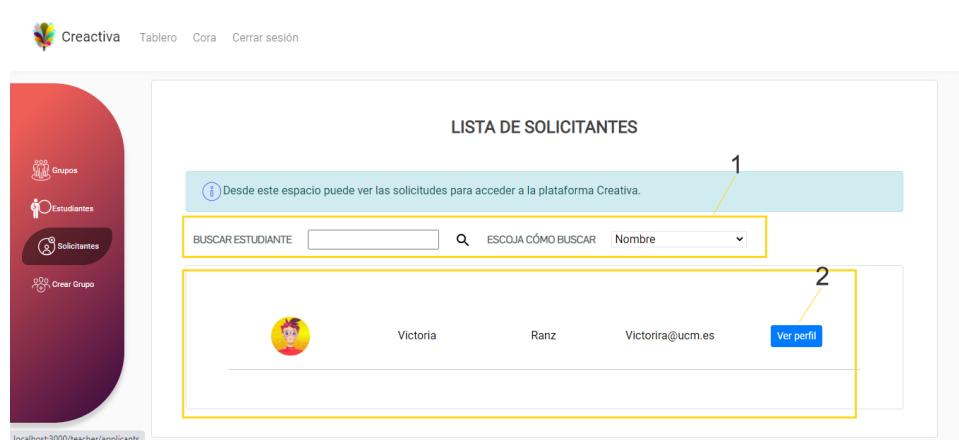


Figura 6.21: Lista de alumnado solicitante

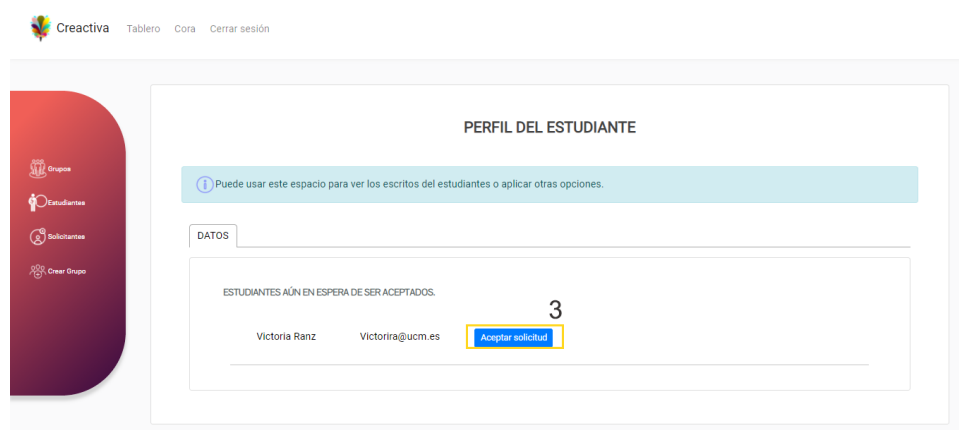


Figura 6.22: Aceptar solicitud

### 6.2.5. Módulo crear grupo

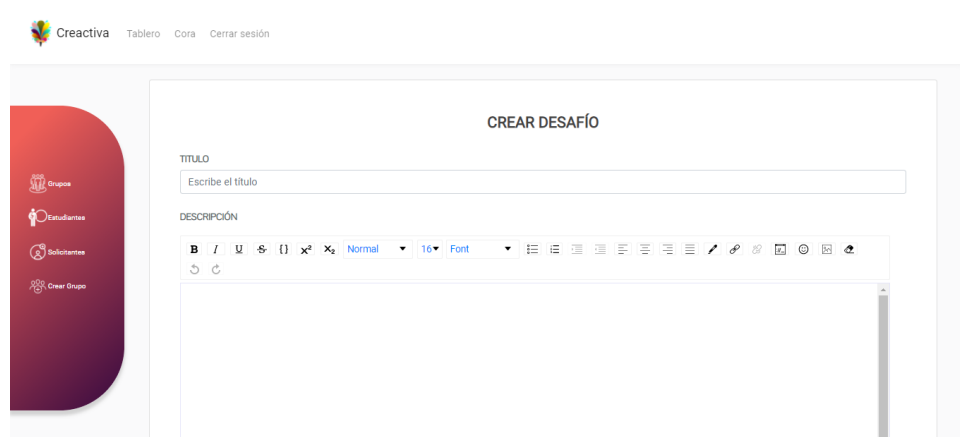
En esta ventana el profesor puede crear un grupo nuevo, para lo cual debe ingresar un nombre (paso 1 en la figura 6.23), pulsar “Aceptar” y confirmar el evento (paso 2 en la figura 6.23).



Figura 6.23: Crear grupo

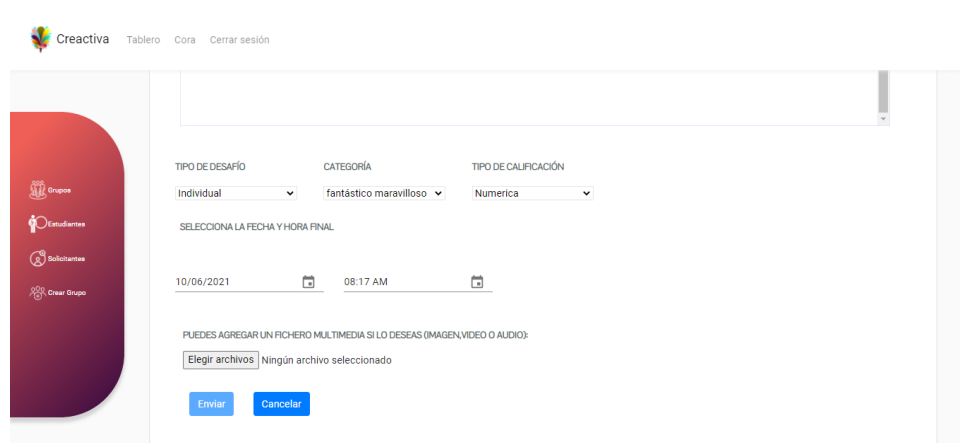
### 6.2.6. Módulo componer y editar un desafío

En este espacio el profesor puede establecer un nuevo desafío ingresando el nombre, una descripción o los requisitos del reto, subir elementos multimedia, colocar hora y fecha fin, elegir una categoría y el tipo de calificación. Una vez rellenado los campos entonces puede pulsar “Enviar” para activar el desafío o “Cancelar” en caso contrario (ver figuras 6.24 y 6.25).



The screenshot shows the 'CREAR DESAFÍO' (Create Challenge) form. At the top, there is a navigation bar with the logo 'Creativa' and links for 'Tablero', 'Cora', and 'Cerrar sesión'. On the left, a sidebar contains icons for 'Grupos', 'Estudiantes', 'Solicitudes', and 'Crear Grupo'. The main form area is titled 'CREAR DESAFÍO' and contains a 'TÍTULO' field with the placeholder 'Escribe el título'. Below it is a 'DESCRIPCIÓN' field with a rich text editor toolbar including options for bold, italic, underline, strikethrough, text color, background color, font size, font family, bulleted list, numbered list, indent, outdent, link, unlink, image, video, and audio. The description field is currently empty.

Figura 6.24: Crear un desafío



The screenshot shows the validation step for creating a challenge. It features three dropdown menus: 'TIPO DE DESAFÍO' (Individual), 'CATEGORÍA' (fantástico maravilloso), and 'TIPO DE CALIFICACIÓN' (Numerica). Below these is a section for selecting the final date and time, with a date picker set to 10/06/2021 and a time picker set to 08:17 AM. There is also a section for adding multimedia files, with a button 'Elegir archivos' and the text 'Ningún archivo seleccionado'. At the bottom, there are 'Enviar' and 'Cancelar' buttons.

Figura 6.25: Validar desafío creado

Análogamente, la vista de editar un desafío es similar a la de crear desafío con la diferencia de que se muestran los datos guardados con anterioridad de su creación o edición.

### 6.2.7. Módulo editar un escrito

En primer lugar, el profesor podrá observar que en el espacio de “Detalles del desafío” se encuentran los requisitos y ficheros multimedia que él o ella han agregado (ver figura 6.26).

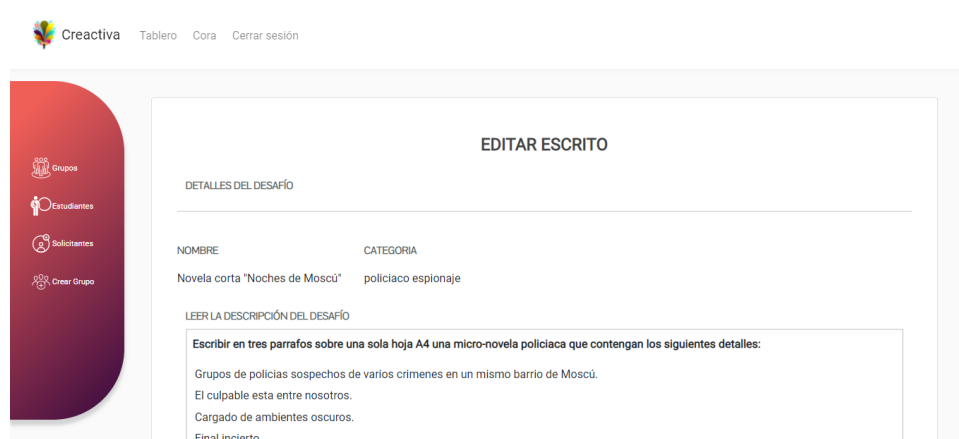


Figura 6.26: Editar escrito “Detalles del desafío”

En segundo lugar, el espacio “Escrito del estudiante” es el texto que el alumno o los alumnos han editado. Desde el cual el profesor puede comentar, corregir, calificar o dar por terminado el escrito (ver figuras 6.27 y 6.28).

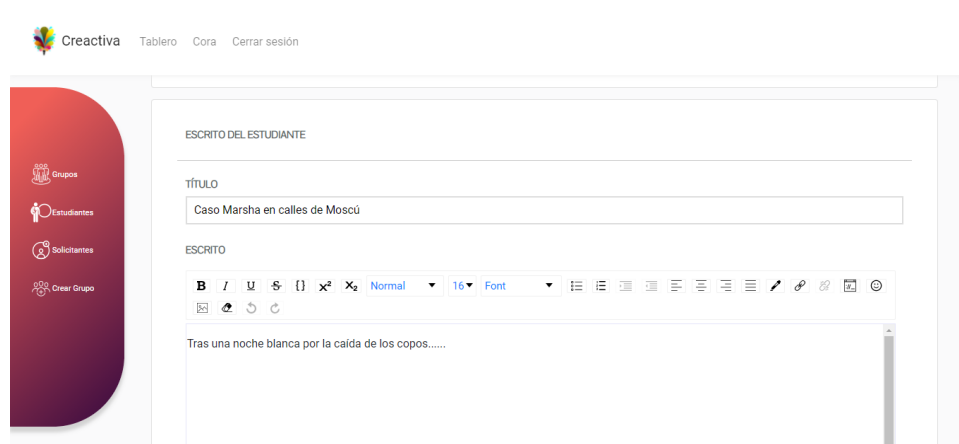


Figura 6.27: Editar escrito “Escrito del estudiante 1”

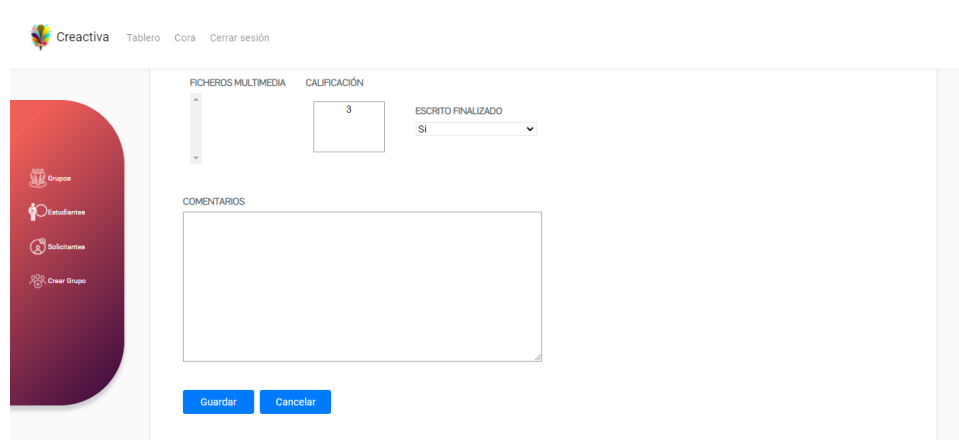


Figura 6.28: Editar escrito “Escrito del estudiante 2”

## 6.2.8. Módulo información del equipo

El profesor podrá ver la información de un equipo, tal como sus miembros (paso 1 en la figura 6.29), sus escritos (paso 2 en la figura 6.29) de la misma forma que se muestra en la figura 6.20 y el perfil de cada miembro (paso 3 en la figura 6.29) como en las figuras 6.19 y 6.18.

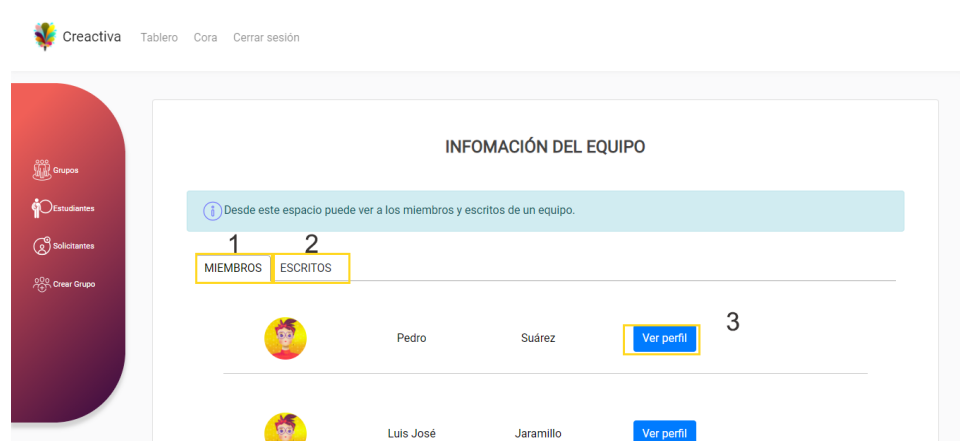


Figura 6.29: información del equipo

## 6.3. Usuario estudiante

Este rol permite a los estudiantes de la aplicación crear los escritos para los desafíos aceptados. También tienen la posibilidad de crear y gestionar equipos de estudiantes.

### 6.3.1. Módulo grupos estudiante

Al iniciar esta ventana el estudiante puede seleccionar un grupo (paso 1 en la figura 6.30) y una opción de gestión sobre el grupo (paso 2 en la figura 6.30), las cuales explicaremos a continuación.



Figura 6.30: Selección de grupos y gestión.



**Crear escrito:** Cuando el estudiante escoge la opción “Crear escrito”, cargará un grupo de pestañas tales como:

- **Desafíos:** Se lista los desafíos activos y disponibles que el estudiante puede aceptar al pulsar “Crear escrito”(paso 3 en la figura 6.31). Si este botón no esta disponible entonces el estudiante ya contiene un escrito activo sobre dicho desafío.

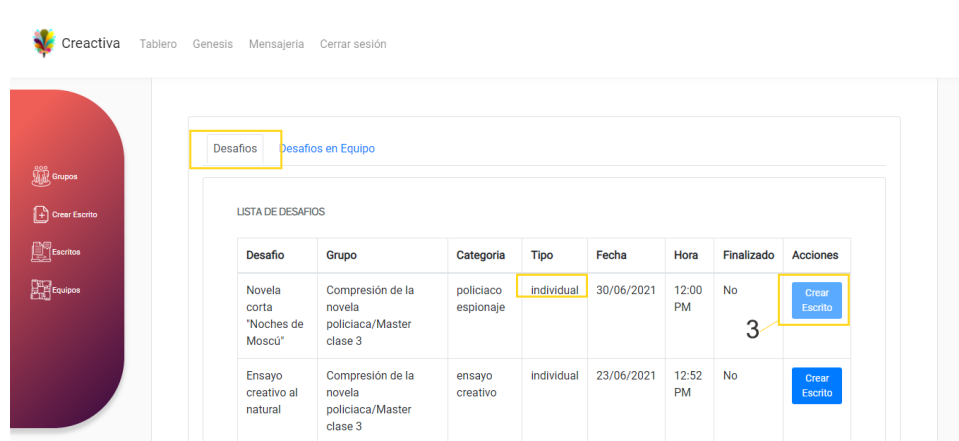


Figura 6.31: Listado de desafíos.

- **Desafíos en equipo:** Se lista los desafíos activos y disponibles para los equipos que pueden ser aceptados al pulsar “Crear escrito” (paso 3 en la figura 6.32). Si este botón no esta disponible entonces el equipo del estudiante ya contiene un escrito activo sobre dicho desafío.

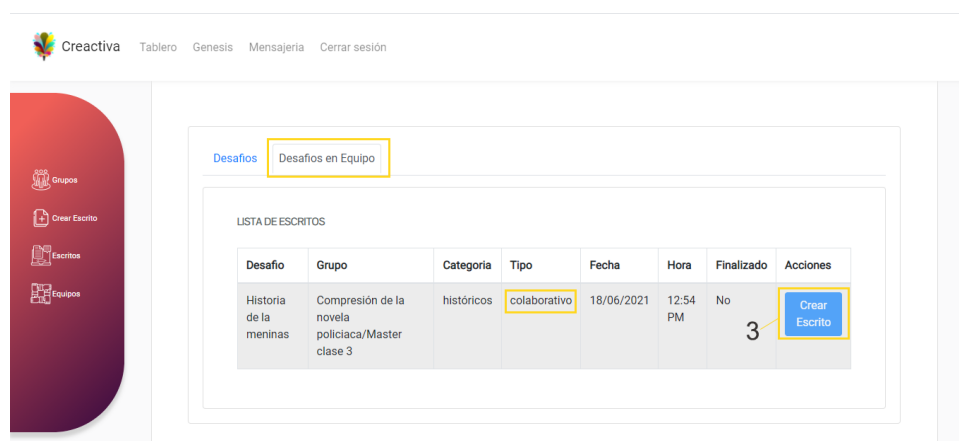


Figura 6.32: Listado de desafíos en equipo.

**Escritos:** Cuando el estudiante escoge la opción “Escritos”, cargará un grupo de pestañas tales como:

- **Escritos:** Se lista los escritos de cada desafío aceptados por el estudiante de modo individual, desde el cual puede acceder al editor de texto pulsando en “Editar” (paso 3).

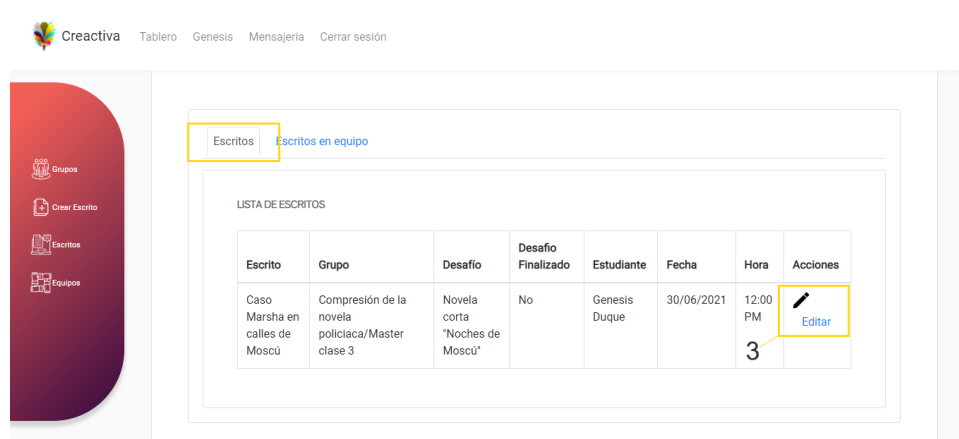


Figura 6.33: Listado de escritos.

- **Escritos en equipo:** Se lista los escritos de cada desafío aceptados por el equipo al que pertenece el estudiante como miembro. Luego, pulsando el enlace “Editar” (paso 3) se redirigirá a la ventana de editar escrito.

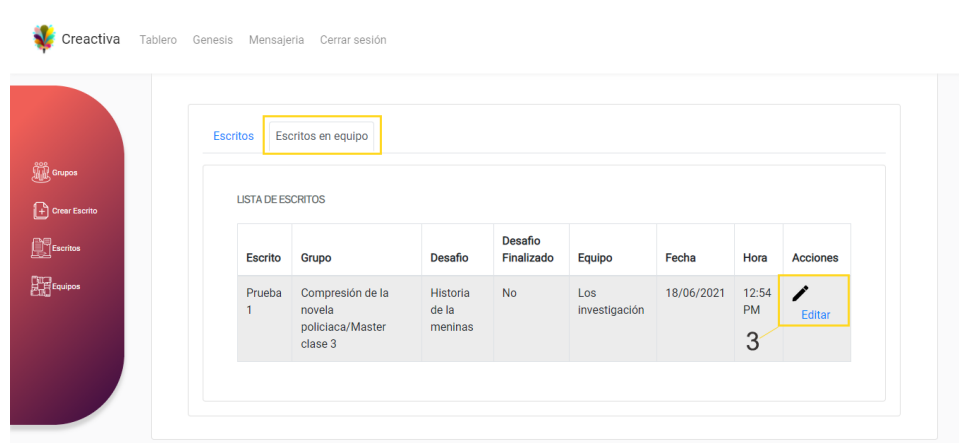


Figura 6.34: Listado de escritos en equipo.

**Equipos:** Cuando el estudiante escoge la opción “Equipos”, cargará un grupo de pestañas tales como:

- **Equipos:** Se lista los equipos activos del grupo seleccionado, de tal manera que si el estudiante aun no pertenece a ningún equipo del grupo entonces puede enviar solicitudes a los líderes de los equipos hasta que alguno de ellos le acepten tal y como podemos ver en la figura 6.35.



Figura 6.35: Enviar solicitud a equipos

- **Mi equipo:** Si el estudiante aún no es miembro de algún equipo entonces puede crear su propio equipo (paso 3 en la figura 6.36) ingresando un nombre y luego confirmando el evento. A continuación se cargará distintas funcionalidades para gestionar su equipo, por ejemplo, invitar a otros compañeros del grupo a su equipo, eliminar miembros ya existentes (paso 4 en la figura 6.37) y observar la información del equipo en la lista (paso 5 en la figura 6.37).

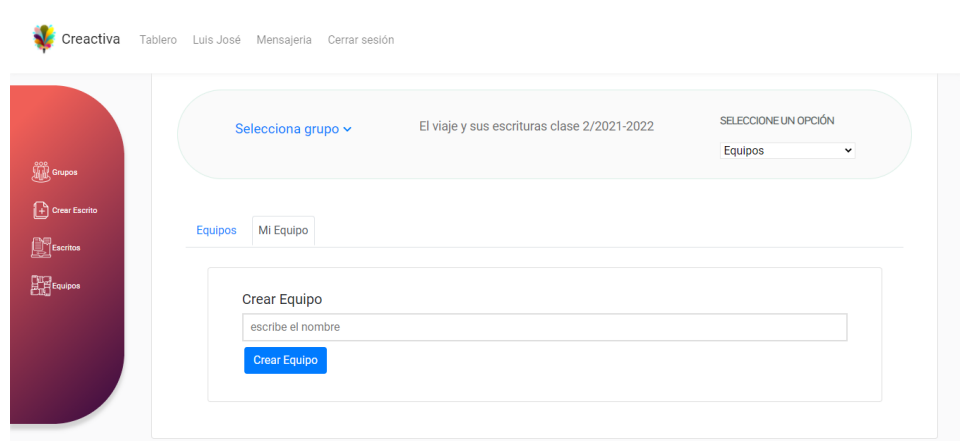


Figura 6.36: Crear equipo.

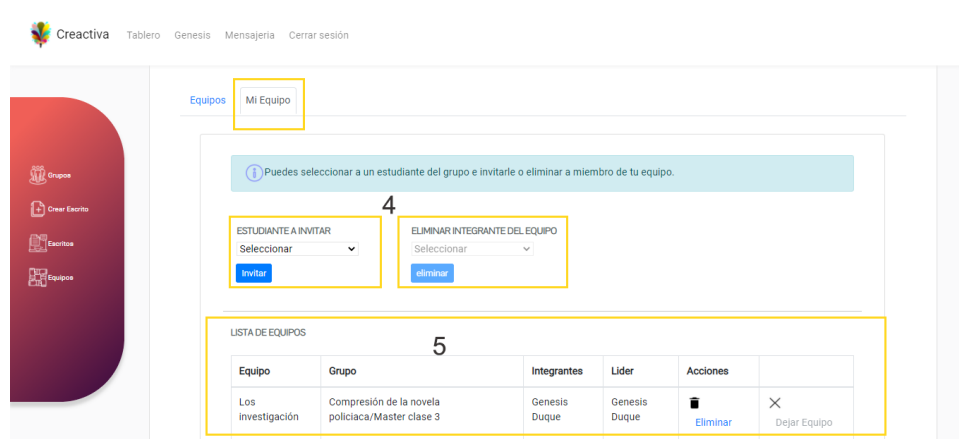


Figura 6.37: Gestionar equipo.

Si el estudiante líder desea abandonar el equipo debe pulsar “Dejar equipo” (paso 6 en la figura 6.38). Tendrá que elegir a uno de los miembros como nuevo líder del equipo (paso 7 en la figura 6.38) o, si no hay miembros, entonces se eliminará el equipo de forma inmediata.

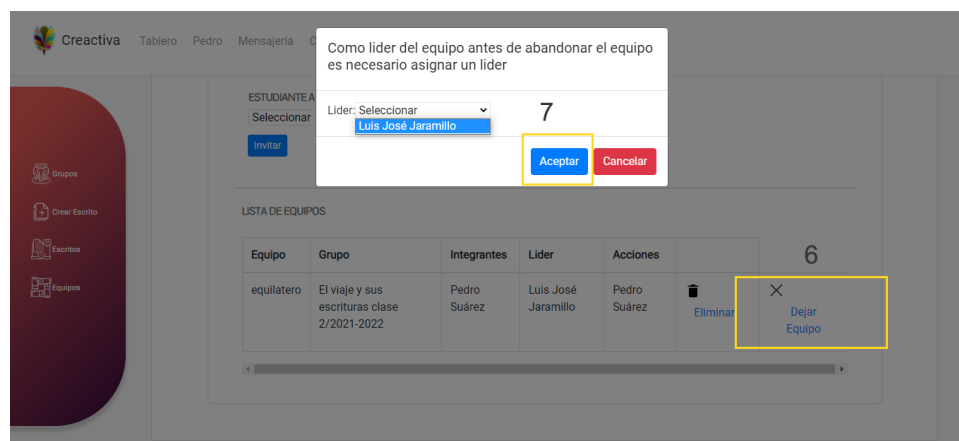


Figura 6.38: Abandonar el equipo.

Si el estudiante líder desea eliminar el equipo solo debe pulsar “Eliminar equipo” (paso 8 en la figura 6.39) y luego aceptar la eliminación total del equipo (paso 9 en la figura 6.39). Esta acción solo es realizada por el líder, en caso de ocurrir los miembros del equipo serán expulsados.

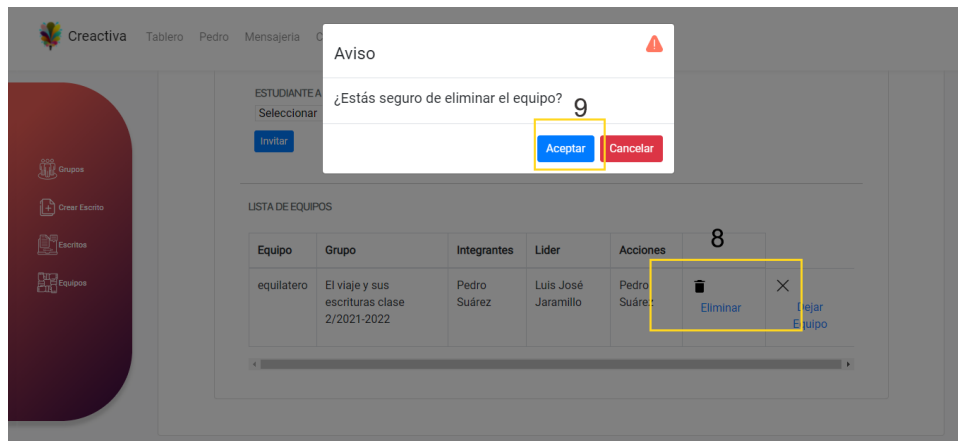


Figura 6.39: Eliminar el equipo.

En otro caso, si el estudiante es miembro de un equipo del cual no es líder solo podrá observar la información del equipo al que pertenece y darse baja del equipo, ver la siguiente figura 6.40

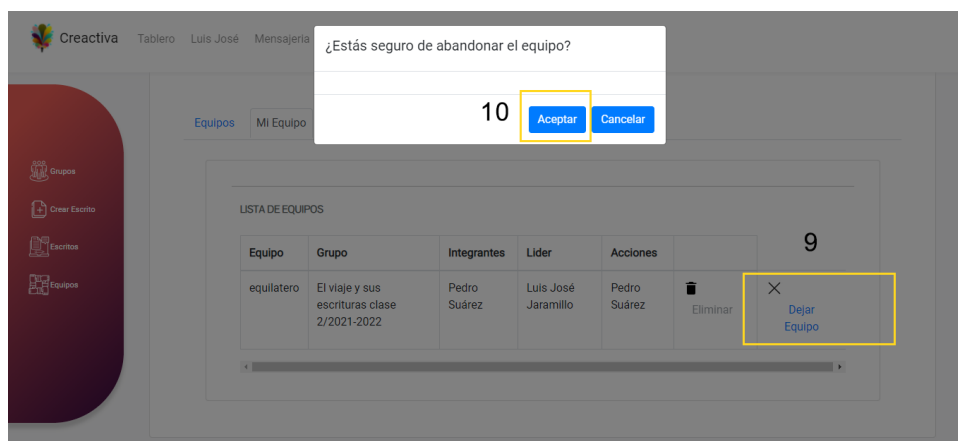


Figura 6.40: Abandonar el equipo un estudiante no líder.

### 6.3.2. Módulo crear escrito

En esta vista el estudiante podrá acceder de forma directa a las pestañas que se muestran las figuras 6.31 y 6.32

### 6.3.3. Módulo Escritos

En esta vista el estudiante podrá acceder de forma directa a las pestañas que se muestran las figuras 6.33 y 6.34

### 6.3.4. Módulo equipos

En esta ventana se listarán los equipos creados por el propio estudiante (ver figura 6.41), donde cada fila expone la información correspondiente al equipo y un enlace con el nombre de “Gestionar” que al ser pulsado nos llevará a la ventana como se muestra en la figura 6.37.

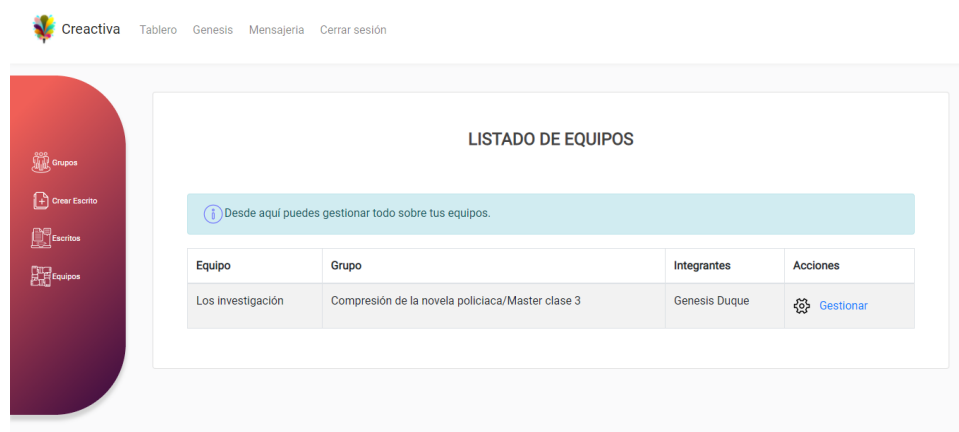


Figura 6.41: Gestión de equipos.

### 6.3.5. Módulo componer y editar un escrito

Sobre este espacio el estudiante puede elaborar de forma individual o en colaborativo su escrito. En primer lugar, el estudiante puede visualizar los detalles, los requisitos y los archivos multimedia que el profesor ha agregado (ver figura 6.42).

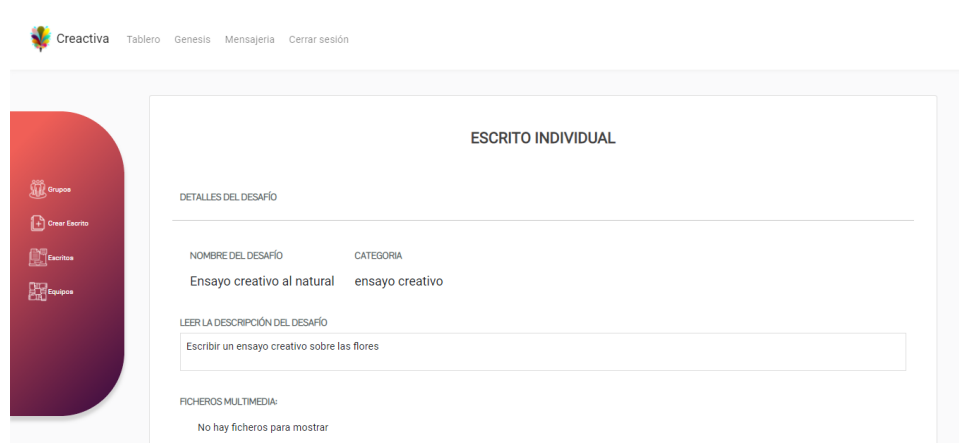


Figura 6.42: Requisitos y detalles que debe cumplir el escrito.

En segundo lugar se cuenta con el espacio de escritura para desarrollar el escrito, el cual cuenta con un espacio para el título, un editor de texto, y ficheros de multimedia (ver figura 6.43).

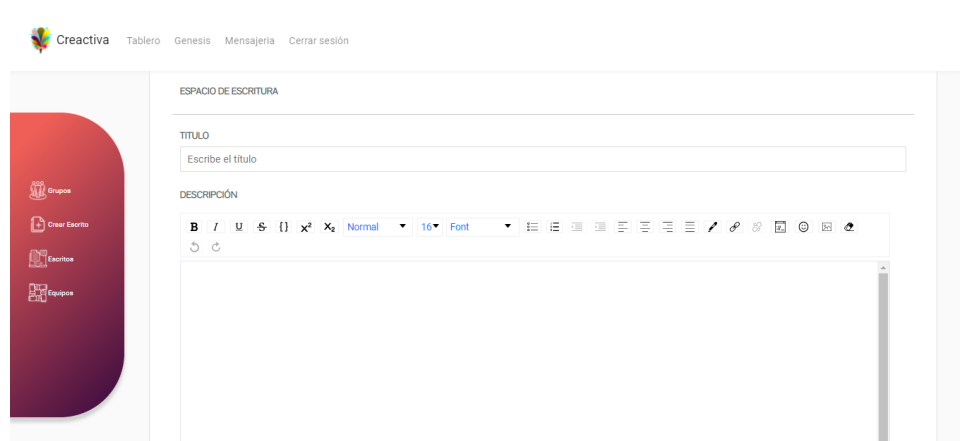


Figura 6.43: Redacción de un escrito.

En tercer lugar, luego de terminar la redacción el estudiante tiene la opciones de “Guardar” y confirmar el evento o “Cancelar” y salir de la ventana sin guardar los cambios previos (ver figura 6.44).

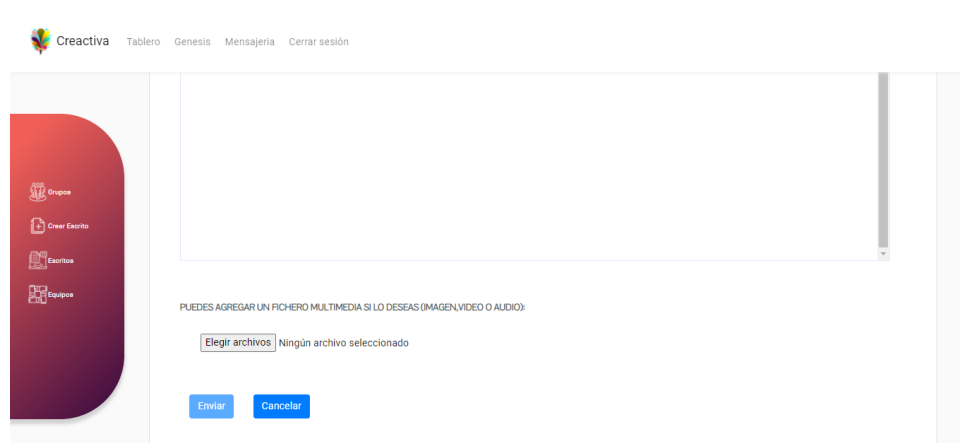


Figura 6.44: Opciones de gestión de una redacción.

### 6.3.6. Módulo mensajería

Inicialmente se listarán los mensajes recibidos por otros estudiantes de la aplicación junto con la fecha, nombre, correo, grupo y hora. Esto como motivo de solicitud de aceptación al equipo creado por el estudiante.

El estudiante también contará con dos opciones tales como “ver” está para visualizar el mensaje como se muestra en la figura figura 6.45 y “Eliminar” el mensaje de la lista.

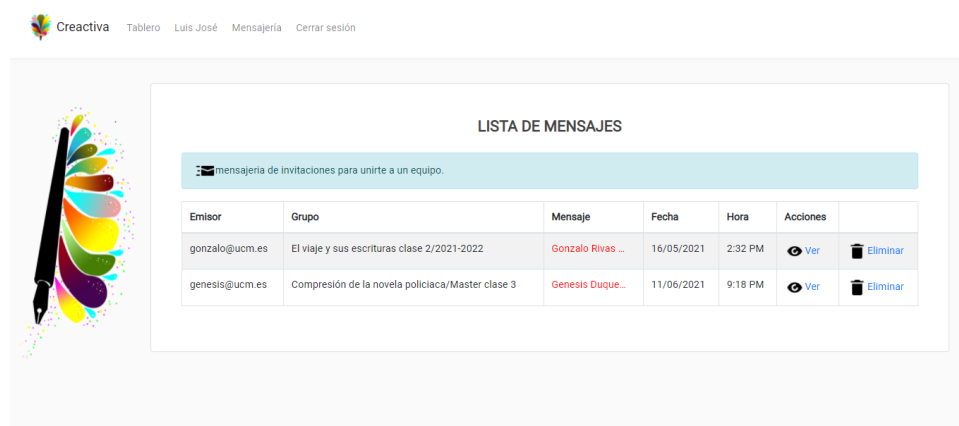


Figura 6.45: Mensajería.

## 6.4. Usuario Administrador

El rol de “Administrador” permite la administración de toda la plataforma, desde la gestión de grupos de los profesores hasta la gestión de solicitudes de nuevos usuarios.

### 6.4.1. Módulo grupos administrador

Desde esta vista se enumera los grupos activos existentes de los profesores, como se observa en la figura 6.46. En la parte superior se muestra un buscador para aplicar un filtrado en la búsqueda de grupos, junto a la información resumida de cada grupo se muestra el botón “Ver Grupo”, que lleva a su vista detallada.

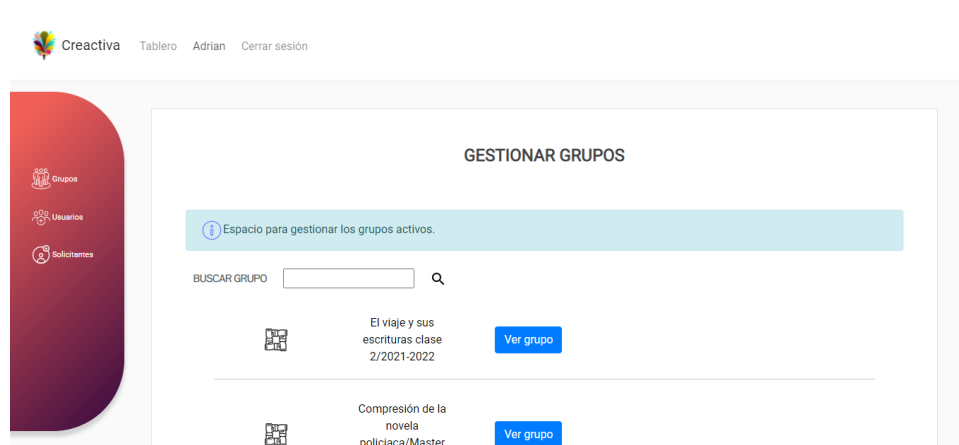


Figura 6.46: Gestionar grupos.

### 6.4.2. Módulo usuarios

Esta ventana muestra la lista de todos los usuarios registrados (ver figura 6.47). Pulsando el botón “Ver perfil” se accede a la gestión del usuario seleccionado.



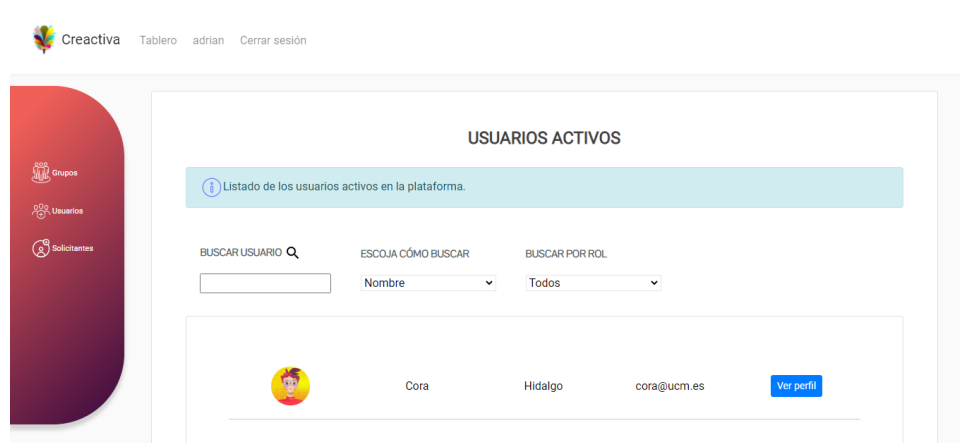


Figura 6.47: Lista de usuarios dados de alta.

### 6.4.3. Módulo solicitantes

Esta vista recopila una lista de usuarios, tanto profesores como alumnos, que han completado el registro y están a la espera de su aprobación (ver figura 6.48). Al acceder desde el botón “Ver perfil” el administrador puede ver la información del nuevo usuario y aprobar la solicitud de registro (ver figura 6.49).

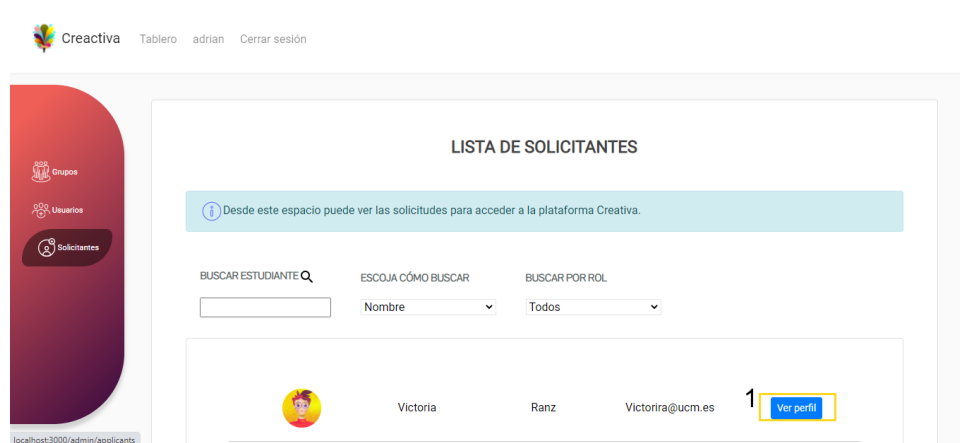


Figura 6.48: Datos del grupo.

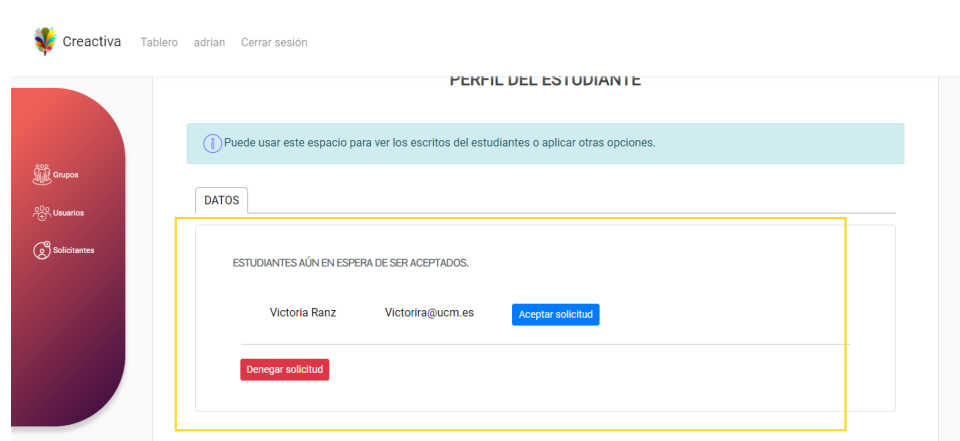


Figura 6.49: Datos del grupo.

#### 6.4.4. Módulo información del grupo

En esta ventana podemos ver la información correspondiente a un grupo en las siguientes pestañas:

**Datos:** Desde este espacio el administrador puede renombrar y desactivar el grupo (paso 1), así como identificar al profesor creador del grupo y ver su perfil (paso 2), observe la figura 6.50.

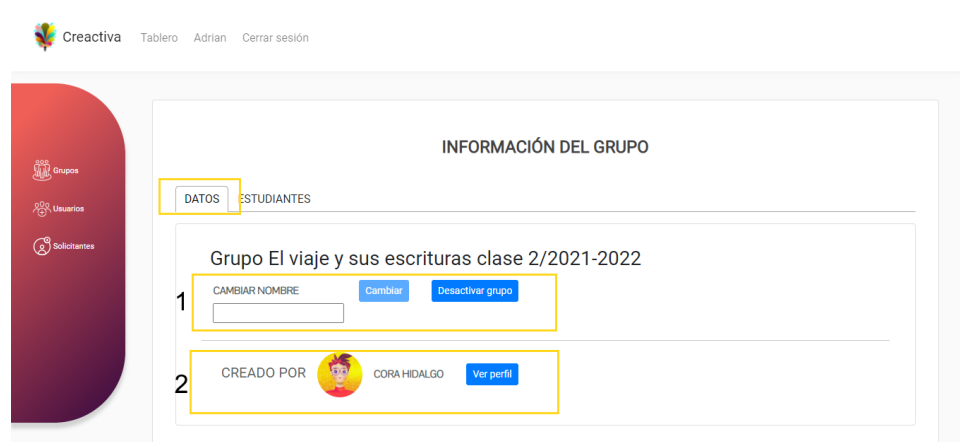


Figura 6.50: Datos del grupo.

**Estudiantes:** Se despliega el listado de los estudiantes miembros del grupo seleccionado, desde el cual podemos acceder al perfil de cada uno pulsando “Ver perfil” (paso 1), observe la figura 6.51.

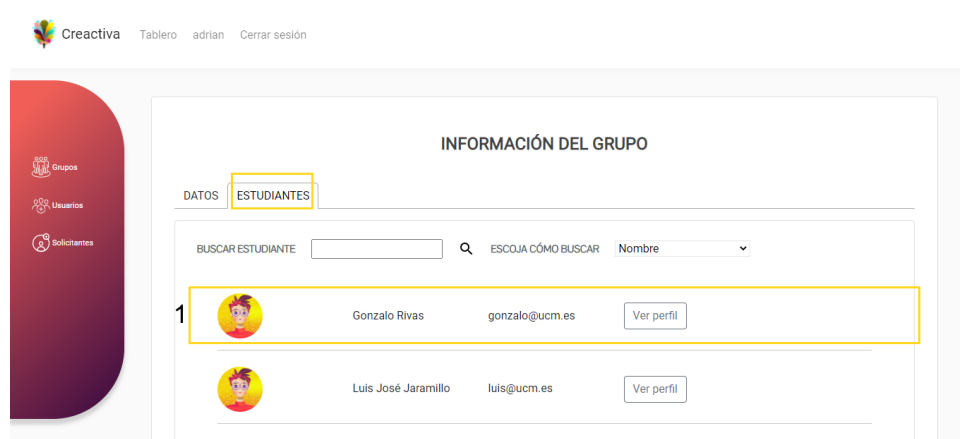


Figura 6.51: Listado de estudiantes.

### 6.4.5. Módulo perfil del estudiante

**Datos:** El administrador desde esta pestaña puede invitar (paso 1) o expulsar a un estudiante de un grupo (paso 2) y desactivar la cuenta (paso 3), ver la figura 6.53 .

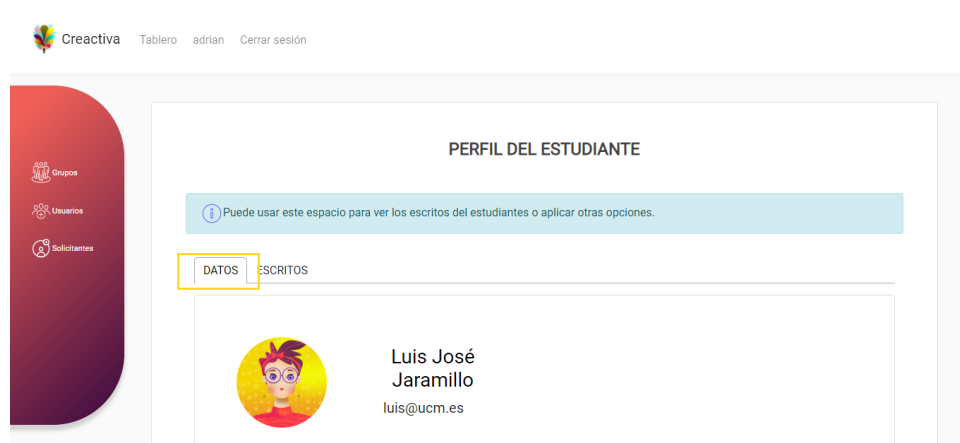


Figura 6.52: Perfil del estudiante.

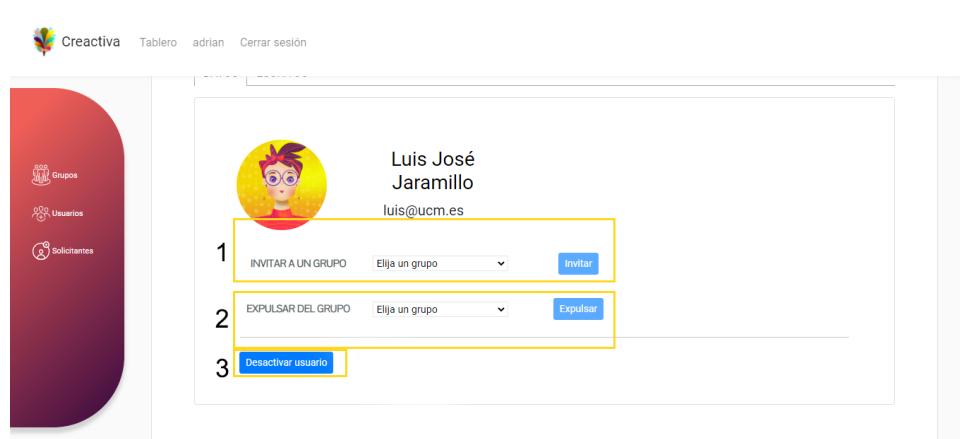


Figura 6.53: Gestionar perfil del estudiante.

**Escritos:** Sobre esta acción el administrador solo puede visualizar los escritos que el estudiante esta realizando o ha finalizado (paso 1), ver la figura 6.54 .

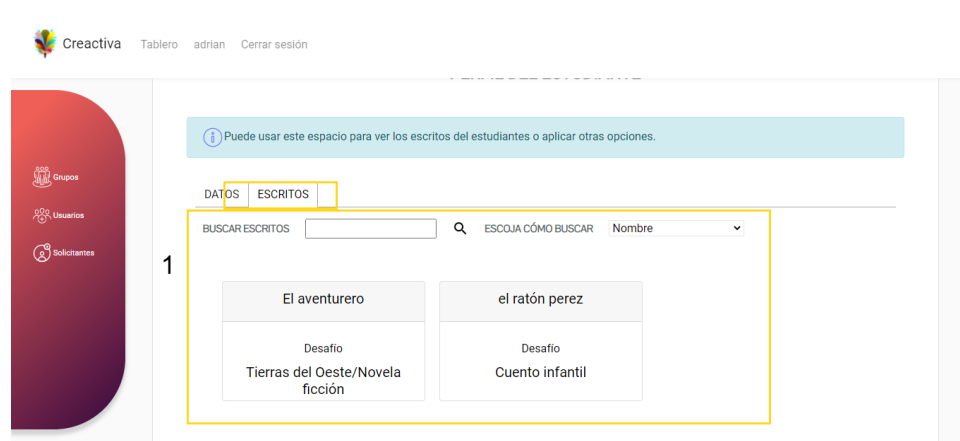


Figura 6.54: Escritos del estudiante.

#### 6.4.6. Módulo perfil del profesor

**Datos:** En esta pestaña se expone los datos del profesor y el administrador pulsando “desactivar usuario” podrá inutilizar la cuenta del profesor como se muestra en la figura figura 6.55.

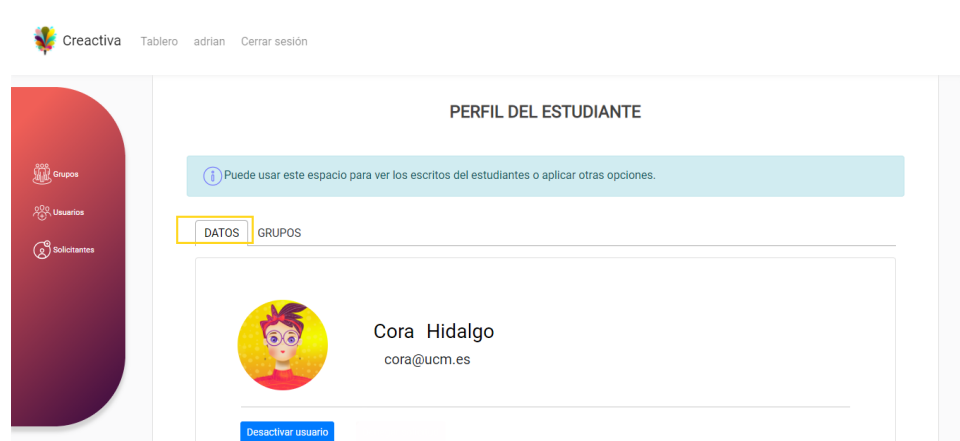


Figura 6.55: Perfil del profesor.

**Grupos:** Se listan los grupos creados por el profesor como se puede ver en la figura 6.56, y el administrador pulsando “Ver grupo” se redirigirá a la figura 6.50.

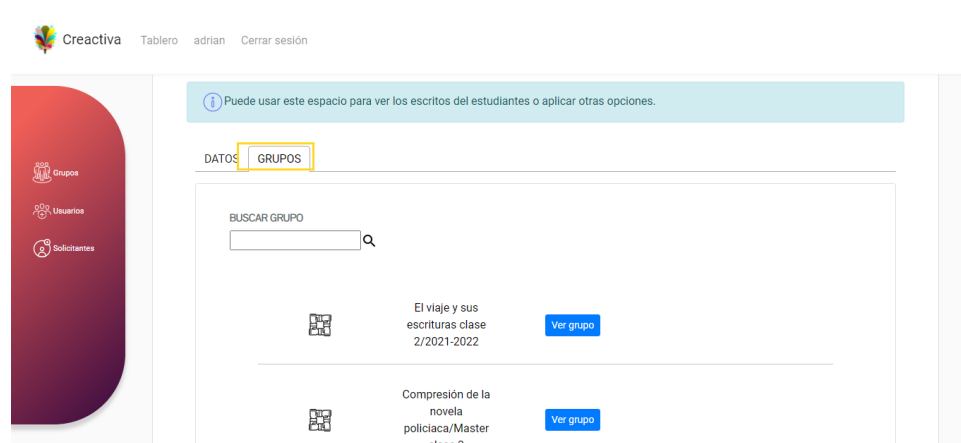


Figura 6.56: Grupos del estudiante.

# Capítulo 7

## Contribuciones individuales

En este capítulo cada uno de los integrantes de *Creativa* explicará sus contribuciones al proyecto, de forma que se pueda valorar el trabajo realizado individualmente. Cabe destacar que este proyecto durante su desarrollo fue compuesto equitativamente entre todos los miembros destacando la buena comunicación, esfuerzo y motivación.

### 7.1. Luis Alberto Jaramillo Pulido

Mi contribución en el proyecto para el desarrollo de la aplicación se ha llevado a cabo a través de varias etapas estructurándose de la siguiente forma.

#### 7.1.1. Primera toma de contacto con el cliente

Nuestra primera reunión se celebró el pasado mes de octubre vía *Google Meet* con nuestro tutor Adrián Riesco y nuestra “cliente” Cora Requena. En esta primera toma de contacto se nos expuso de forma más detallada la temática del trabajo fin de grado y una idea aproximada de la aplicación web a desarrollar. Se me encomendó la tarea de mantener periódicamente la comunicación por correo para celebrar las reuniones entre los profesores y mis compañeros.

#### 7.1.2. Etapa de Investigación

Durante esta etapa después de habernos reunido en una primera instancia con el cliente, me propuse a realizar una labor de búsqueda de aplicaciones similares como *iDeas for Writing*, *Story Planer*, *Literautas* entre otros, con la finalidad de mejorar la comprensión sobre la aplicación.

También en esta etapa colaboré en la elaboración de las funcionalidades que va a requerir la aplicación y en la elección de las tecnologías que se van a utilizar en el proyecto. Como no disponía de conocimiento previo sobre algunas tecnologías como *NodeJS* y *ReactJS*, comencé a investigar de distintas fuentes como videotutoriales de *youtube*, web

o apuntes que me prestaron mis compañeros. Esto me ayudó a adquirir un mayor grado de conocimiento y aptitudes para el desarrollo del proyecto.

### 7.1.3. Diseño del Prototipo

Posteriormente decidimos elaborar el prototipo de la aplicación utilizando la herramienta *Balsamiq Cloud* para visualizar resultados provisionales y poder mostrarlos al cliente, con el fin de determinar una mejor comprensión de los requerimientos del cliente.

Por ello nos repartimos las vistas de los roles de la aplicación: administrador, profesor y estudiante entre los tres miembros del equipo, encargándome del diseño de los *mockups* de la vista del estudiante en el prototipo.

### 7.1.4. Gestión del proyecto

Antes de empezar el proceso de desarrollo del proyecto, era necesario organizar nuestro entorno de trabajo.

En una primera instancia creé un documento *Excel* para organizar las tareas en el *Drive*. Este documento nos serviría para ir apuntando los requisitos de la aplicación como historias de usuario.

El problema que nos encontramos usando una hoja de cálculos de *Excel*, es la dificultad de gestionar el proyecto cuando existen muchos requisitos, siendo difícil hacer un seguimiento de las tareas de mis compañeros, cuando existen muchas historias de usuarios, resultando la colaboración improductiva por la mala gestión del tiempo y del equipo.

Por ello trasladé a mi equipo la necesidad de una herramienta de gestión de proyecto como *Trello*. En esta plataforma de gestión de proyectos me encargué de organizar todos los artefactos de la metodología *Scrum* como el *Product Backlog* y *Sprint Backlog*, y utilizar el límite de trabajo en progreso *To Do*, *Doing*, *Done*, para estar al tanto de todo el flujo de trabajo de las tareas de mis compañeros y hacer una mejor gestión del tiempo y del equipo.

### 7.1.5. Colaboración en el Repositorio

Todo el equipo podía trabajar conjuntamente en un mismo proyecto a través de un repositorio remoto común de *Github*. Mi trabajo en esta parte se basa en la colaboración de la gestión del proyecto mediante el seguimiento del proyecto, realización de cambios, actualización o incorporación de nuevas funcionalidades en la aplicación.

En un primer momento todas las operaciones de gestión del control de versiones se hacían a través de la línea de comandos del *Git*. Más adelante mi compañera Génesis nos sugirió el uso de una aplicación cliente de *Git* llamada *Gitkraken*, que nos permitía hacer todas las gestiones del proyecto desde una interfaz de cliente, haciendo que todo el proceso de la gestión del repositorio resulte más sencillo.

### 7.1.6. Diseño de la Base de datos

Colaboré en el diseño de la base de datos en la etapa inicial de desarrollo, diseño de la base de datos para cada uno de los roles, definición de las tablas de la aplicación, relaciones entre tablas, etc.

Durante todo el proceso de desarrollo de la aplicación a medida que se iba incorporando nuevas funcionalidades agregué nuevas tablas a la base de datos como equipos, mensajería, multimedia de desafío y multimedia de escritos.

### 7.1.7. Implementación

Una vez definido nuestra organización de trabajo, procedimos a la parte de la implementación del proyecto. Conseguí hacer una primera versión de conexión de la base de datos del servidor con el cliente y del inicio de sesión, siendo después mejorada por mi compañera Genesis, y también una versión inicial de las vistas del administrador, profesor y estudiante.

Después nos dividimos el trabajo, en esta parte me encargue de toda la funcionalidad del estudiante y parte de las funcionalidades del profesor.

Las funcionalidades que he implementado lo he estructurado en tres secciones, funciones generales de la aplicación, funciones del rol del estudiante, y funciones del rol del profesor.

#### Funciones generales de la aplicación

Las funciones generales de la aplicación he considerado las funcionalidades comunes en ambos roles de profesores y estudiantes.

**Editor de texto:** Permite al profesor o estudiante escribir los textos de los desafíos o escritos en un editor de texto enriquecido, se ha dado la posibilidad de aportar variedad de estilos enriquecidos a sus textos mediante el acoplamiento de este *API* a la aplicación.

**Calendario:** *API* utilizada para que el profesor pueda agregar una fecha de duración al desafío. Esto lo hace a través de un selector de fecha y hora, mediante un *widget* de interfaz gráfica de usuario.

**Registro de historial de colaboración del escrito:** Trabajé en el desarrollo de una primera versión de registro mediante *Socket.io*, que guardase en un borrador del escrito, el historial de aportaciones de cada estudiante del equipo al hacer un escrito colaborativo. Esta implementación no se terminó de desarrollar, pero se ha tenido en cuenta como trabajo futuro.

**Multimedia en desafíos y escritos:** Los profesores o estudiantes tienen la opción de aportar más recursos a sus desafíos o escritos mediante la incorporación de contenido multimedia como vídeos, ficheros de audio e imágenes, o eliminación de los archivos multimedia en desafíos o escritos.

**Almacenamiento multimedia en desafíos y escritos:** Implementé la funcionalidad de subir los contenidos multimedia a un servidor web utilizando una plataforma de



servicio de alojamiento en la nube *Cloudinary*, el inconveniente que tenía es que era de pago cuando excedías de un límite de capacidad de alojamiento de archivos en la plataforma, por eso decidimos que el almacenamiento multimedia de los desafíos o escritos fuese en el servidor local.

Después de ello me encargué de la implementación de esta funcionalidad, donde cada profesor o estudiante dispone de una carpeta personal en el servidor, en la cual cada desafío o escrito tiene una carpeta propia donde se almacena el contenido multimedia de sus desafíos o escritos.

También me ocupé de la implementación de la creación de una carpeta personal del usuario donde se almacena la foto de perfil del profesor o estudiante.

**Modales de confirmación y modales informativos:** Modales de confirmación y modales informativos cuando un usuario ejecuta una acción en la plataforma.

**Avisos:** Avisos en la plataforma cuando no existen desafíos, grupos, escritos, mensajes y equipos.

**Foto de perfil:** Subida y actualización de la foto de perfil del estudiante al servidor.

## Funciones del rol de Estudiante

Estas son las funcionalidades implementadas en el rol del estudiante.

**Listado de grupos:** Permite al estudiante listar todos los grupos a los que pertenece.

**Listado de desafíos:** Permite al estudiante listar los desafíos según el tipo de desafío: individuales o colaborativos. En la aplicación se puede acceder de dos formas, seleccionando un grupo y listando el desafío del grupo del estudiante, y la otra forma es listar todos los desafíos de distintos grupos.

**Creación de escritos:** Permite al estudiante la creación escritos individuales o colaborativos, mediante el envío y validación de los datos del escrito. En escritos colaborativos permite que cualquier miembro del equipo puede crear un escrito.

**Edición de escritos:** El estudiante puede editar un escrito individual o colaborativo si dispone de un equipo, mediante la actualización y validación de todos los campos del escrito. En escritos colaborativos permite que todos los integrantes del equipo puedan editar un mismo escrito.

**Listado de escritos:** El estudiante puede listar todo tipo de escritos ya sean individuales o colaborativos. En la aplicación se puede acceder de dos maneras, seleccionando un grupo, listando los escritos con su correspondiente desafío del grupo del estudiante, y la otra manera es listar todos los escritos de distintos grupos del estudiante.

**Listado de mensajes:** Los estudiantes tienen la opción de listar todos los mensajes recibidos en su bandeja.

**Envío de mensaje:** En la aplicación existen dos formas de envío de mensajería o solicitudes, la primera forma es cuando un estudiante sin equipo envía una solicitud o mensaje a un propietario para unirse a un equipo, y la otra forma es cuando un propietario

del equipo envía una invitación a un estudiante sin equipo para que forme parte de su equipo.

**Consultar mensaje:** El estudiante puede consultar su mensajería, para aceptar o rechazar una solicitud para formar parte de un equipo o una invitación para unirse a un equipo.

**Eliminar mensaje:** Permite al estudiante eliminar mensaje de la bandeja de mensajería.

**Listado de equipos:** En la aplicación un estudiante puede listar todos los equipos de un grupo seleccionado, y también puede listar todos los equipos que tiene de distintos grupos.

**Consultar equipo:** En la aplicación el estudiante puede consultar directamente su equipo o seleccionar un grupo y consultar su equipo.

**Expulsar integrante del equipo:** Permite al estudiante seleccionar al integrante del equipo que desea expulsar.

**Eliminar equipo:** Permite al propietario del equipo eliminar un equipo.

**Cambiar propietario del equipo:** El estudiante propietario de un equipo tiene la opción de seleccionar a otro miembro para heredar el rol de propietario del equipo.

### **Funciones del rol de profesor**

Estas son las funcionalidades implementadas en el rol del profesor

**Creación de desafíos:** Permite al profesor la creación de desafíos individuales o colaborativos, mediante el envío y validación de los datos del desafío.

**Edición de desafíos:** Los profesores pueden editar un desafío individual o colaborativo, mediante la actualización y validación de todos los campos del desafío.

**Listado de desafíos:** Permite al profesor listar desafíos individuales o colaborativos, seleccionando un grupo y listando el desafío del grupo del profesor.

**Eliminar desafío:** Los profesores pueden eliminar un desafío

**Edición escritos:** El profesor puede editar cualquier campo del escrito y también puede agregar comentarios al escrito.

**Listado de escritos:** El profesor puede visualizar todos los escritos de los estudiantes de un grupo después de haber seleccionado un desafío.

### **7.1.8. Presentación de la aplicación al cliente**

Realizamos una presentación del proyecto de forma telemática hacia un grupo de profesores de Ciencias de la información, donde presentamos mediante ejemplos el funcionamiento de la aplicación.

Mi aportación consistió en la colaboración de las diapositivas y presentación de las funcionalidades de la aplicación. Nuestra intención fue conocer de primera mano las

impresiones sobre la aplicación, y también recoger comentarios de mejora que podíamos incorporar al proyecto.

### 7.1.9. Aportaciones en la memoria

En la memoria los puntos que he desarrollado son los siguientes:

**Tecnologías:** Explicación de todas las tecnologías empleadas en el proyecto.

**Metodologías:** En este apartado se explica de la metodología seguida en el grupo de trabajo y la plataforma utilizada para llevarla a cabo

**Patrón de diseño:** Esta sección está incluida dentro del capítulo de la arquitectura, se hace una breve explicación de patrón de diseño escogido y el motivo de su elección y las ventajas que nos aporta al seleccionar este patrón de diseño.

**Estructura de la Arquitectura de la aplicación:** En esta sección se explica la organización de la arquitectura del proyecto.

**Autenticación basada en tokens:** Explicación del proceso de autenticación utilizando *tokens*

**Funcionalidades:** Apartado que está incluido dentro del capítulo de la arquitectura, se menciona las funcionalidades principales de la aplicación, así como una breve explicación de cada uno.

**Contribución del proyecto:** Explicación de todo el trabajo aportado al proyecto durante el proceso de desarrollo de la aplicación:

## 7.2. Génesis Brigitti Duque Escobar

Durante el desarrollo de la aplicación *Creativa* mis contribuciones se distribuyeron por distintas áreas de trabajo y varias etapas como se detalla a continuación.

### 7.2.1. Etapa de investigación

Inicialmente la investigación partió desde la búsqueda de tecnologías, métodos e instrumentos de maquetación y como crear la arquitectura de una página web.

A continuación, el propósito era comprender cómo llevar a cabo la construcción de una plataforma web compuesta por una interfaz de usuario y un motor de procesos lógicos, ya que es la forma más común y exitosa, permitiendo un buen rendimiento y mantenimiento sin un alto coste sobre los cambios. Si bien, el desafío inicial para nosotros era lograr la conexión e integración de dos motores distintos con los que ninguno había antes trabajado y conocer qué herramientas eran las necesarias para desarrollar este primer objetivo.

Luego inicié una investigación por diversas páginas web y canales de vídeos en *Youtube* dedicados al desarrollo y a partir de esto surge proponerles a mis compañeros *ReactJS*, una herramienta con la anteriormente había trabajado y junto con la propuesta de Gonzalo

Rivas de utilizar *NodeJS* para la parte lógica de la aplicación, concluimos que eran nuestros principales instrumentos con los que dar inicio al desarrollo de la plataforma debido a su compatibilidad y las ventajas que nos ofrecía. Finalmente, di comienzo a la parte *Frontend* instalando todos los programas necesarios en mi computador.

### 7.2.2. Diseño de prototipo

Tras mi incorporación en el proyecto mis compañeros ya habían avanzado con diferentes esquemas de la plataforma, creando un entorno con las ideas que se propusieron en la primera consulta con el cliente. Por tanto, a partir de los bocetos ya construidos inicié una mejora en todos ellos para mejorar el flujo de navegación, el estilo visual de los componentes y proporcionar un diseño ligero y fácil de utilizar por los usuarios. Así mismo, se continuó con la creación de otras vistas que luego fueron presentadas y bien recibidas por el cliente.

### 7.2.3. Gestión de proyecto

Creé y realicé la actualización de un documento *Excel* llamado “Planificador de tareas” desde el espacio de *Drive*, donde se llevaba el control de cada reunión realizada con el tutor y el cliente o reuniones rápidas por los miembros del proyecto, y también se anotaban las tareas futuras a revisar en la siguiente sesión.

### 7.2.4. Repositorio

Mi contribución consistió en crear desde mi cuenta de *Github* un repositorio de versiones con el objetivo guardar todos nuestros avances y trabajar de forma paralela en distintas partes del proyecto. Una vez añadidos mis compañeros, les ayudé a comprender el funcionamiento y comportamiento de la herramienta de forma local y remota.

Además, después de trabajar los primeros meses de desarrollo con *Git* a través de la consola, me recomiendan el uso de *Gitkraken* debido a su ágil gestión de proyectos en repositorios. Tras probarlo unos días expuse a mis compañeros su usabilidad y lo integré en el proyecto.

### 7.2.5. Conexión del motor lógico y la interfaz de usuario

Tras haber creado el proyecto dividido en *Frontend* y *Backend*, faltaba realizar la conexión entre ambas instancias. Desde este punto empieza mi búsqueda de alguna API y su método de implementación cuyo resultado tras varios de investigación fue la biblioteca *Axios* porque puedes realizar solicitudes hacia un servidor, y son completamente configurables, y recibir respuesta de una manera sencilla de procesar enviada desde el motor lógico.

En primer lugar, procedí a la instalación de la API con el sistema de paquetes *npm* y luego en la parte del cliente añadí los ficheros *Javascript* que importan a la biblioteca *Axios* y el código que solicita las peticiones de los usuarios.

En segundo lugar, tras los avances de mis compañeros en las distintas partes, realicé pruebas con peticiones de tipo GET y POST, cuyo primer resultado fue encontrar problemas debido una mala configuración de rutas desde el cliente hasta la base de datos.

En tercer lugar, después de la corrección de fallos conseguí establecer la primera conexión y recibir la respuesta de una petición sencilla tal como extraer los datos personales de la tabla de usuario.

### 7.2.6. Implementación de autenticación del usuario

Al inicio, cuando se implementó el inicio de sesión y el registro de usuario, no se consideró la autenticación de la identidad y la seguridad de la contraseña del usuario. Por ello, mi contribución fue integrar a *React* la biblioteca de *JWT Token autenticación* junto con las validaciones de los formularios y *Axios*.

Si bien, primero he añadido los ficheros necesarios para ejecutar el servicio en la parte del *Frontend* y luego la configuración para la autenticación del cliente con la API desde el *Backend*.

### 7.2.7. Implementación del estilos

Durante la primera fase de esta etapa utilicé hoja de estilos *CSS* para desarrollar una rejilla con la que ofrecer el control de las dimensiones y el desplazamiento de los componentes como por ejemplo la barra de navegación, un contenedor y una barra lateral como se muestran en el capítulo 6. Después, tras terminar cada funcionalidad fui añadiendo al código *HTML* estilos y colores así como la organización en las vistas.

A continuación, para minimizar el tiempo de desarrollo de la interfaz del usuario, instale bibliotecas externas de *ReactJS* cuya funcionalidad era importar componentes visuales con estilos *CSS* ya implementados, y con la posibilidad de ser modificados tales como “react-bootstrap” y “material-ui”.

Por ultimo, es de mencionar la implementación de adaptaciones a pantalla de diversos dispositivos, es decir, un diseño web *responsive* donde cada componente se ajusta en tamaño y ubicación dependiendo de la resoluciones de cada pantalla.

### 7.2.8. Presentación de la aplicación al cliente

Para realizar esta presentación nos dividimos por segmentos las diapositivas presentadas al cliente, de las cuales elaboré aquellas necesarias para explicar la misión y la visión de este proyecto y los usuarios participantes. El día de la presentación di la bienvenida y comienzo de la exposición, y al finalizar ayudé a responder las preguntas e inquietudes de los profesores participantes.

### 7.2.9. Aportación a la memoria

La memoria ha sido realizada de forma equitativa por todos los miembros, y parte de mi trabajo desarrollado se muestra a continuación en los siguientes apartados:

**Prototipo:** Explicación sobre la elaboración de nuestro primer prototipo.

**Guía de uso:** Especificación de cada una de las vistas y sus funcionalidades para cada tipo de usuario, donde se detallan los pasos seguir y cómo debe realizar el evento.

**Evaluación de la encuesta realizada sobre *Creativa*** Argumentación sobre los resultados arrojados en la encuesta realizada a un grupo de profesores del máster sobre la escritura creativa impartido en la universidad complutense de Madrid.

## 7.3. Gonzalo Rivas Fernández

Las aportaciones de Gonzalo Rivas Fernández se centran en cuatro grupos, funciones del rol del profesor, funciones del rol del administrador, mecanismos generales de la aplicación y aportaciones a la memoria.

### 7.3.1. Funciones del rol de profesor

**Listado de estudiantes:** Permite buscar a todos los estudiantes que tengan acceso a la plataforma. Contiene un sistema para filtrar por nombre y apellidos o por email. Permite acceder a los perfiles de estos estudiantes.

**Listado de solicitantes:** Permite buscar a todos los estudiantes que aún no tienen acceso a la plataforma. Añade un sistema para filtrar por nombre y apellidos o por email. Permite acceder a los perfiles de estos estudiantes.

**Ver estudiantes de un grupo:** Pestaña que pertenece a la vista de un grupo permite ver a todos los estudiantes que pertenecen al mismo. Cuenta con un sistema de filtrado por nombre y apellido o correo electrónico.

**Ver perfil de estudiante:** Permite ver los datos de un estudiante y ejecutar ciertas acciones sobre el mismo. Puede acceder a datos de tanto estudiantes como solicitantes.

**Añadir estudiante a un grupo:** Mediante un selector, el profesor puede añadir a un estudiante a uno de sus grupos.

**Expulsar a un estudiante de un grupo:** El profesor puede seleccionar a uno de sus grupos para que el estudiante deje de formar parte del mismo.

**Aceptar solicitante:** Da permisos a un solicitante para poder emplear la plataforma como un estudiante de pleno derecho.

**Ver escritos de estudiante:** Muestra una lista de los escritos elaborados por un estudiante. Esta lista puede ser filtrada por nombre del escrito o título del desafío relacionado.

**Consultar escritos de un equipo:** Busca una lista de los escritos elaborados por un equipo. Contiene una herramienta de búsqueda por nombre de escrito o título del desafío.

**Crear grupo:** Genera un grupo nuevo con el nombre dado por el profesor.

**Renombrar grupo:** Otorga un nombre nuevo al grupo seleccionado.

### 7.3.2. Funciones del rol de administrador

Muchas de estas funciones son similares a las que posee el profesor, pero el rol de administrador tiene una mayor libertad al aplicarlas al tener más privilegios.

**Ver grupos:** Muestra una lista de todos los grupos existentes en la aplicación que puede ser filtrada por nombre.

**Ver información de grupo:** Permite visualizar datos sobre el grupo, como su nombre y el profesor que lo gestiona.

**Desactivar grupo:** Hace que un grupo ya no sea visible por los usuarios.

**Renombrar grupo:** Da un nuevo nombre al grupo escogido.

**Ver estudiantes del grupo:** Muestra una lista con todos los estudiantes que pertenecen al grupo seleccionado. Puede ser filtrada por nombre y apellidos o por email.

**Ver usuarios:** Expone todos los usuarios con acceso a la aplicación. Puede ser filtrada por nombre y apellidos o correo electrónico. Permite buscar por rol.

**Ver solicitantes:** Muestra todos los usuarios que aún no han sido aceptados en la aplicación. Esta selección se puede filtrar.

**Ver perfil de usuario:** Muestra la información básica de un usuario y permite realizar ciertas acciones sobre él dependiendo de su rol y su estado de privilegios en la aplicación.

**Añadir estudiante a un grupo:** Mediante un selector, el administrador puede añadir a un estudiante a cualquier grupo disponible.

**Expulsar a un estudiante de un grupo:** El administrador puede seleccionar cualquier grupo a los que pertenece un estudiante para expulsarlo del mismo.

**Desactivar usuario:** Revoca el permiso de un usuario a acceder a la plataforma. Este usuario queda de nuevo como un solicitante. No se puede ejecutar en un administrador.

**Eliminar usuario:** Borra la cuenta de un usuario, así como su contenido creado. No se puede ejecutar en un administrador.

**Ver escritos de estudiante:** Muestra una lista con todos los escritos del estudiante que se pueden filtrar.

**Ver grupos de profesor:** Expone todos los grupos que gestiona un profesor, que pueden ser filtrados por nombre.

### 7.3.3. Mecanismos generales de la aplicación:

**Sistema de filtrado por búsqueda dinámica:** Empleado en múltiples lugares de la aplicación, esta herramienta de búsqueda filtra al introducir una clave de búsqueda al instante sin necesidad de pulsar un botón de buscar. Permite utilizar distintos campos de los datos para filtrar la información.

**Normalización del sistema de tarjetas en la vista:** Ordena los distintos tipos de información en distintas tarjetas a las que se puede acceder desde un índice superior.

**Sistema de trato de errores:** Se ha implementado un sistema de redirecciones que muestran cuándo ha habido un error interno en el servidor o si el usuario ha solicitado recursos inexistentes, asegurando así que nunca pierda el control sobre el flujo de la aplicación.

**Sistema de sustitución de imágenes de perfil:** Coloca una imagen por defecto si un usuario no dispone de imagen de perfil en diferentes listados y vistas de perfil.

**Sistema de adaptación de ventanas al rol:** Algunas funcionalidades, como los listados emplean las mismas clases independientemente de qué rol las acceda para mejorar su mantenibilidad. Este sistema permite a estas clases adaptarse a las necesidades de cada tipo de usuario y mostrar la información más relevante dependiendo de cuál sea el usuario conectado.

**Estructurado y diseño del servidor de datos:** Una separación clara entre los diferentes componentes del funcionamiento del servidor, de manera que se mantenga el encapsulamiento y se pueda identificar de una manera clara y estructurada cada una de las funciones del mismo. Separa las funciones que pertenecen a cada uno de los roles de los usuarios, las funciones empleadas en común y la funcionalidad de la autenticación e ingreso.

### 7.3.4. Aportaciones a la memoria:

La memoria ha sido realizada de forma colaborativa por los miembros del equipo. Por lo tanto, se muestran a continuación los apartados realizados completamente por la persona o en los que haya contribuido.

**Palabras clave:** Enumerado de palabras clave para facilitar la búsqueda y categorización del proyecto.

**Introducción:** Realización y traducción del capítulo en el que incluyen los apartados de la motivación de la creación del proyecto y los objetivos que se persiguen con su desarrollo.

**Resumen:** Creación y traducción del capítulo introductorio que da una breve descripción del proyecto de *Creativa*.

**Modelo de datos:** Desarrolla el esquema del diseño de la información empleada por la aplicación. Añade el modelo entidad-relación como una manera de visualizar toda la estructura de un vistazo, así como el desarrollo de la base de datos *SQL*, donde se detalla cada entidad que forma parte de la misma.



**Conclusiones y trabajo futuro:** Creación y traducción del capítulo que contiene los pensamientos finales y reflexiones de *Creativa*, así como las posibilidades de expandirla en el futuro.

**Glosario:** Este apartado define varios conceptos en los que se basa el funcionamiento de la aplicación.

### 7.3.5. Cronología

Durante la época inicial del desarrollo, mientras planificábamos cómo estructurar la aplicación, propuse emplear las tecnologías de *NodeJS* y *ExpressJS* para crear la parte del *backend* de la aplicación, ya que son tecnologías que pueden adaptarse a diversos sistemas de manejo de la vista. Proveí de material e información al equipo sobre cómo emplear estas tecnologías según nuestras necesidades en el proyecto. Contribuí en la realización de *mockups* para generar un prototipo de la parte visual de la aplicación, principalmente de la parte de las ventanas profesor. Organisé un espacio colaborativo en *Google Drive* para que el equipo pudiese tener siempre disponible un espacio donde compartir cualquier archivo o diagrama que empleásemos durante el desarrollo.

Al desconocer gran parte de cómo funcionaba *ReactJS*, estuve consultando diverso material, en parte proporcionado por el equipo para aprender sobre el mismo y realicé un pequeño proyecto de prueba para ver cómo empezar de cero a usarla. De forma similar estuve trabajando sobre cómo emplear la herramienta de control de versiones *Git* para poder trabajar en tándem con el resto del equipo.

Me encargué de diseñar la parte del esquema de datos que se refiere a los usuarios de tipo profesor. Participé en la sesión conjunta donde todo el equipo diseñó en detalle la implementación de la base de datos. Una vez logrados estos avances en común, el equipo procedió a generar la estructura básica sobre la que se basa *Creativa*, y cuando ya estaba creado el esqueleto que conecta la parte de la vista con el servidor de datos, se distribuyeron las tareas entre los miembros del equipo. Durante este desarrollo me encargué de las partes de la funcionalidad del administrador y del profesor, así como algunas mejoras generales.

Durante la preparación de la presentación que se realizó con el personal de la Facultad de Ciencias de la Información, creé la encuesta que se repartió a los asistentes y contribuí a la presentación de diapositivas agregando la parte de los profesores, también exponiéndola durante la reunión.

Por último colaboré a realizar y traducir diversos capítulos de la documentación del proyecto.

## Capítulo 8

# Evaluación de la encuesta realizada sobre *Creativa*

Una vez realizadas las funcionalidades más importantes de nuestra aplicación y mejoras en la calidad de estilo y diseño, se decidió por presentar una demostración de la plataforma a un grupo de dieciséis profesores de la Universidad Complutense de Madrid, el máster de escritura creativa.

El objetivo al hacer esta presentación[13] era recoger diversas opiniones si *Creativa* cumplía la expectativas de uno de los usuarios futuros a utilizar la aplicación.

Al finalizar la exposición se dio inicio a una ronda de preguntas donde la mayoría de los profesores se vieron interesados pronunciando cuestiones y de las cuales surgieron ideas de implementar a futuro. Si bien, dada por terminada la presentación se envía una invitación a los correos de todos los participantes para realizar una encuesta sobre *Creativa*.

La siguiente información representa las preguntas y las respuestas de la encuesta [14].

**Respuesta 1** Respecto a la noción sobre los objetivos de *Creativa* el 100% de los encuestados opinaron que ha sido explícito la descripción de las funcionalidades, otorgando la máxima nota con un “Muy de acuerdo” (ver el figura 8.1) .

¿Le ha resultado claro el concepto de la aplicación? (Por ejemplo, qué hace cada tipo de usuario, o qué son los desafíos o los grupos)

2 respuestas



Figura 8.1: Resolución sobre la noción explícita de *Creativa* .

**Respuesta 2** De acuerdo a los datos obtenidos el 50 % de los encuestados opinaron que ha sido muy sencillo e intuitivo, otorgando la máxima un “Muy de acuerdo”, mientras que el otro 50 % le ha asignado con un “De acuerdo” (ver el gráfico 8.2).



Figura 8.2: Respuestas sobre la experiencia del usuario.

**Respuesta 3** Las opiniones obtenidas acerca de la solicitud de acceso de los estudiantes sea controlada por los profesores, el 100 % de los encuestados afirman un “sí” sobre mantener dichas restricciones en la aplicación (ver el gráfico 8.3).

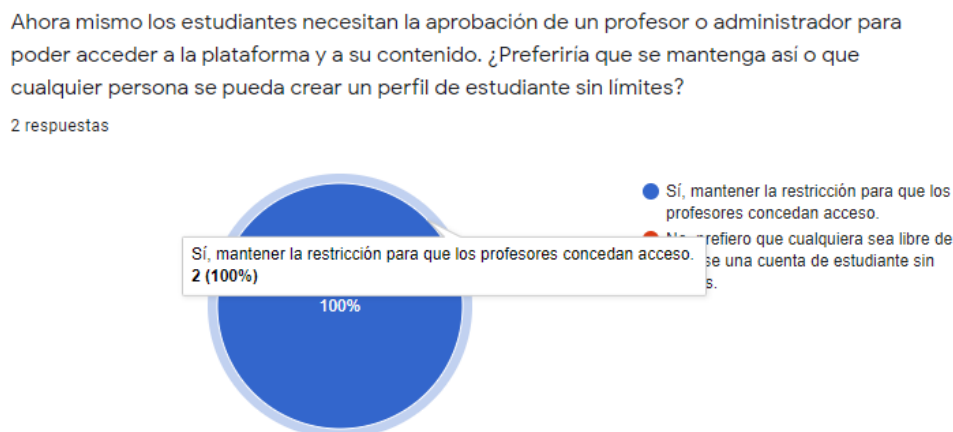


Figura 8.3: Resolución sobre el control de acceso a los estudiantes.

**Respuesta 4** Respecto en agregar funcionalidades de los encuestados, en este caso los profesores echaran en falta, el 100 % contestaron con una opinión personal sobre lo que a ellos les gustaría ver en la aplicación, véase en el gráfico 8.4

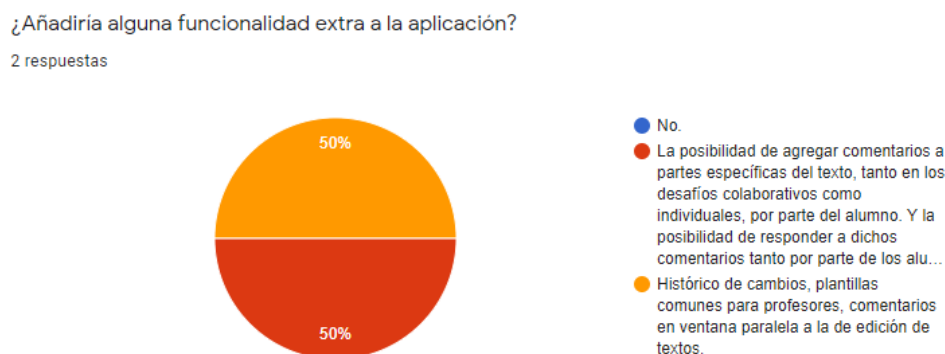
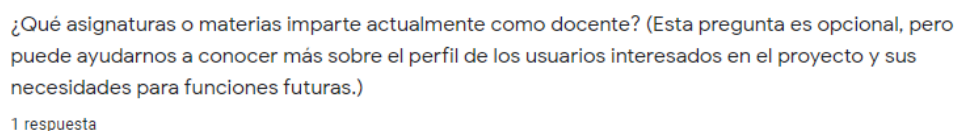


Figura 8.4: Agregación de ideas extras para *Creativa* .

**Respuesta 5** Esta última cuestión de la encuesta era de forma opcional, donde se pedía a los encuestados conocer sus áreas de trabajo, ya que, con estos datos se conseguiría analizar el perfil de los usuarios *Creativa*, uno de los encuestados participantes respondió a esta pregunta como podemos observar en el gráfico 8.5



Literatura y Medios de Comunicación (grados en Comunicación Audiovisual y Periodismo), Literatura y Prensa periódica (Grado en Periodismo), Novela policiaca (Máster universitario en Escritura Creativa), Teoría y práctica de la escritura de no ficción y periodística (Máster universitario en Escritura Creativa).

Figura 8.5: Perfil de usuarios para *Creativa* .

En conclusión, la cantidad de los datos no han sido suficientes para realizar un análisis deductivo de cada una de las preguntas y poder contrastar en detalle los porcentajes debido a que el alcance de los profesores encuestados solo fueron dos de dieciséis participantes en la presentación de la aplicación. Aún así, durante la presentación se recibieron una serie de opiniones muy positivas por parte de varios de los asistentes, expresando que deseaban emplear esta herramienta en su labor educativa.

# Capítulo 9

## Conclusiones y trabajo futuro

Este capítulo contiene las reflexiones finales tras el trabajo realizado y las posibilidades que tiene de ser expandido en el futuro.

### 9.1. Conclusiones

En este proyecto se ha desarrollado una plataforma web para promover la escritura creativa. Para ello se ha organizado la comunidad de usuarios en profesores que proponen desafíos y alumnos que siguen estas pautas para crear nuevas obras literarias. Se dispone de un sistema de grupos para poder organizar a los usuarios según sus intereses a la hora de escribir, así como un sistema para aportar comentarios y sugerencias sobre los escritos existentes. De igual manera se ha dado una mayor libertad creativa al usuario desarrollando un sistema que permita aceptar archivos multimedia como audio o vídeo para complementar sus escritos.

Se han empleado nuevas tecnologías para el desarrollo de una interfaz sencilla y dinámica que sea cómoda de utilizar y, al mismo tiempo, no dificulte la tarea creativa. En consecuencia, el equipo ha investigado tecnologías web como *ReactJS* para su uso en la parte visual del proyecto, así como el uso de *NodeJS* y *ExpressJS* para la implementación del servidor, estructurando el modelo de integración con una base de datos relacional en SQL. Conforme el desarrollo ha ido avanzando, se ha expandido el proyecto al integrar diversos componentes como tecnologías de autenticación y encriptado para aumentar la seguridad de la aplicación o bibliotecas como *MaterialIU* o *React Bootstrap* para mejorar su aspecto visual. Se han empleado diversas herramientas para realizar el trabajo en grupo de forma remota, se ha empleado *Github* para mantener un control de versiones. Para gestionar el flujo del proyecto se ha creado un tablero en *Trello*, donde se organizan las tareas y su estado de desarrollo. Se ha mantenido la comunicación mediante reuniones rutinarias empleando herramientas como *Google Meet* y *Discord*.

Al ser *Creactiva* una plataforma ideada para que sirva de complemento a las clases de la Facultad de Ciencias de la Información se ha otorgado el derecho de admisión de nuevos estudiantes a los usuarios que ostenten el rol de profesores, permitiendo así mantener una comunidad creativa dentro del entorno académico.

## 9.2. Trabajo futuro

*Creativa* tiene la intención de ser una plataforma viva usada para fomentar la creatividad. Por tanto existen algunas funciones que podrían añadirse a la misma para hacer aún más completa la experiencia de usuario. Con esta intención se mantuvo una reunión con miembros de la comunidad educativa de la Facultad de Ciencias de la Información, buscando sus opiniones sobre la aplicación y futuras funcionalidades que podrían añadirse. Las ideas principales sobre líneas de trabajo futuras serían las siguientes:

- Posibilidad de **exportar los escritos en formato PDF** imprimible.
- Mantener un **sistema de versiones de los escritos**, permitiendo su lectura y revisión. Esta función también permitiría al estudiante volver a versiones anteriores para retomar su trabajo desde otro punto de vista.
- Posibilidad de que varios **estudiantes trabajen simultáneamente** sobre el mismo escrito cuando este sea colaborativo. Inclusión de un sistema de mensajería simultáneo para estas situaciones.
- Posibilidad de que los profesores puedan **incluir comentarios sobre partes específicas del texto** de un escrito para dar más detalle a sus opiniones y sugerencias.
- Posibilidad de los estudiantes a **contestar dichos comentarios**.
- Un **sistema de plantillas para los profesores**, de manera que puedan organizarse desafíos similares con una mayor comodidad.
- Posibilidad de **crear una biblioteca que recoja los escritos** de los estudiantes para consumo general de los usuarios.
- Ampliación del sistema de mensajería para que los **estudiantes puedan solicitar a los profesores ser incluidos en sus grupos**.
- Posibilidad de **enlazar unos escritos con otros**, permitiendo así la creación de sagas literarias y otros trabajos relacionados. Ampliación similar del sistema de desafíos para crear una modalidad encadenada que permita la creación de escritos sucesivos.

# Capítulo 10

## Concluding remarks and ongoing work

This chapter contains the final thoughts on the work done as well as the ways it could be expanded in the future.

### 10.1. Concluding remarks

This project consists in the development of an online platform that supports creative writing. In order to achieve this goal, the platform's user community has been split between teachers that create literary challenges and students that face these challenges by making new artistic work. A group system has been implemented that sorts users based on their interests as well as a system that facilitates providing feedback on existing writings. A higher level of creative freedom has been granted to the user by adding a media storage system that allows the usage of images, audio, and video files to complement their writings.

The team has utilized new technologies in the creation of a dynamic and simple interface that is comfortable to use and, at the same time, does not hamper the creative experience. Because of this, the team has investigated technologies like *ReactJS* for the visual part of the project, as well as *NodeJS* and *ExpressJS* in the server implementation, using a relational *SQL* database in order to sort the data infrastructure. As development on the project advanced, it was expanded by integrating new components like authentication and encryption technologies, used to raise the platform security, and libraries like *MaterialIU* and *ReactBootstrap*, used to enhance the visual aspects of the app. As the team has worked on the project by remote means, diverse tools like *Github* have been used for version control. A *Trello* board has been used to keep the workflow organized by sorting the different tasks by their level of development. Online tools like *Google Meets* and *Discord* have been used to host weekly meetings.

Since *Creactivais* meant for use as a support tool on Facultad de Ciencias de la Información's courses, teachers have the right to admit new students on the platform, giving them the tools to keep a creative community in a learning environment.

## 10.2. Future work

*Creativa* has the goal to be a live platform that is used to foment creativity. Because of this, there are some features that could be added to improve the user experience. With this in mind the team hosted a meeting with members of the educative community from the Facultad de Ciencias de la Información in order to gather feedback on the app and possible future additions. The main lines of future work are the following:

- Possibility of **exporting writings as printable PDF files**.
- Keeping a **version control system for writings**, that supports both reading and checking them. This feature would allow students to go back to previous work and resuming their writing from a different point of view.
- Possibility of **students working in tandem** on the same writing if it is collaborative. It could include a chat system supporting communication between writers.
- Possibility of teachers **putting comments on specific parts of the text** of a writing that would let them make more precise remarks.
- Possibility for the students to **answer these comments**.
- A **template system for teachers**, that could ease the creation of similar challenges.
- Possibility of **creating a library that houses all writings** from students for everyone to enjoy.
- Expanding the messaging system in order to allow **students to ask teachers to invite them to their groups**.
- Possibility of **linking writings to one another**, allowing the creation of literary sagas as well as other related works. Expanding the challenge system in the same manner in order to create a chain challenge mode that enables the creation of successive writings.



# Bibliografía

- [1] A. Nandaa, *Beginning API Development with Node.js*. Packt Publishing, Limited, 2008. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=5477671>.
- [2] A. Friends, “Documentation: Xampp.” [https://www.apachefriends.org/es/faq\\_windows.html](https://www.apachefriends.org/es/faq_windows.html).
- [3] Y. L. S. J. S. M. K. G. Elizabeth Naramore, Jason Gerner, *Beginning PHP5, Apache, and MySQL Web Development*. John Wiley Sons, Incorporated, 2005. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=225829>.
- [4] R. Shah, “How to use trello.” <https://blog.trello.com/how-to-use-trello-to-write-a-book>.
- [5] A. P. Umesh R Sharma, *GitHub Essentials*. Packt Publishing, Limited, 2018. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=5446049>.
- [6] R. Somasundaram, *Git : Version Control for Everyone*. Packt Publishing, Limited, 2013. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=1119761&query=>.
- [7] S. G. Pérez, “Control de versiones: Gitkraken.” <https://medium.com/@sergupe6/control-de-versiones-por-qu%C3%A9-gitkraken-ee1f30b4a18f>.
- [8] B. Team, “Documentation: Balsamiq.” <https://balsamiq.com/wireframes/cloud/docs/intro/>.
- [9] D. Megida, “Uploading files using node.js and multer.” <https://blog.logrocket.com/uploading-files-using-multer-and-node-js/>.
- [10] J. Puri, “React draft wysiwyg.” <https://jpuri.github.io/react-draft-wysiwyg/#/docs>.
- [11] T. A. A. Reference, “Matt zabriskie.” [https://axios-http.com/docs/api\\_intro](https://axios-http.com/docs/api_intro).
- [12] T. React, “React bootstrap.” <https://react-bootstrap.github.io/getting-started/introduction/>.
- [13] Creactiva, “Presentación de creactiva.” [https://docs.google.com/presentation/d/1-zirMi\\_NX-Z50iz3mSwgiauFPIXFYh6IDFujiLcg43o/edit#slide=id.gd448e9a636\\_0\\_49](https://docs.google.com/presentation/d/1-zirMi_NX-Z50iz3mSwgiauFPIXFYh6IDFujiLcg43o/edit#slide=id.gd448e9a636_0_49).

- 
- [14] Creativa, “Encuesta sobre creativa.” [https://docs.google.com/forms/d/e/1FAIpQLSdWoyehO-6bik-o2RFSLK8phiuBf4nys\\_WZg7gZYqImK4FLrQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdWoyehO-6bik-o2RFSLK8phiuBf4nys_WZg7gZYqImK4FLrQ/viewform?usp=sf_link).
- [15] L. M. C. . . Material-UI, “Material-ui.” <https://material-ui.com/es/>.
- [16] E. R. Martinez, *React : Build 4 real-world apps with React Native*. Packt Publishing, Limited, 2018. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=5322211>.

# Apéndice A

## Glosario

Este anexo define los conceptos empleados para explicar el funcionamiento de la aplicación *Creativa*.

**Profesor:** Tipo de usuario capaz de crear y organizar grupos, proponer desafíos literarios y revisar escritos creados por estudiantes. Tiene la autoridad para aceptar a nuevos estudiantes en la aplicación.

**Estudiante:** Tipo de usuario capaz de enfrentarse a desafíos propuestos por los profesores creando escritos literarios. Puede organizarse junto con otros estudiantes formando equipos. Forman parte de grupos organizados por profesores.

**Administrador:** Tipo de usuario capaz de gestionar la comunidad de la aplicación y sus datos. Gestionan al resto de usuarios, así como sus grupos. Son capaces de aceptar nuevos estudiantes y profesores a la aplicación.

**Solicitante:** Sean profesores o estudiantes, son usuarios que se han creado una cuenta y están a la espera de ser aceptados en la aplicación por un responsable de la misma.

**Grupo:** Conjunto de estudiantes gestionados por un profesor. Permite controlar qué usuario puede acceder a determinados desafíos.

**Desafío:** Propuesto por un profesor, reta a los estudiantes a crear un escrito siguiendo una serie de pautas. Puede contener archivos multimedia para inspirar a aquellos que decidan enfrentarse al desafío. Está ligado al grupo para el que fue creado. Pueden ser propuestos de forma individual o colaborativa.

**Escrito:** Creación literaria realizada por un estudiante siguiendo las pautas de un desafío. Puede contener archivos multimedia para ampliar las posibilidades artísticas.

**Equipo:** Creado y formado por estudiantes de un grupo, se organiza para enfrentarse de forma conjunta a un desafío colaborativo. Cuenta con un líder que lo forma y una serie de miembros.

Frase  
Autor

Autor TFG

FECHA

Ult. actualización 15 de junio de 2021

L<sup>A</sup>T<sub>E</sub>X lic. LPPL & powered by **TEFLON** CC-ZERO

Este documento esta realizado bajo licencia Creative Commons “CC0 1.0 Universal”.

