

# Creativa 2.0

Rafael Alcaide Torralvo  
Verónica Calzada Álvarez  
Jaime Madriñán Fernández

TRABAJO DE FIN DE GRADO. FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

---



Trabajo Fin de Grado en Ingeniería del Software/ Ingeniería Informática

Fecha: 30-05-2022

Directores:

Cora Requena Hidalgo  
Adrián Riesco Rodríguez

# Resumen en castellano

El objetivo de este Trabajo de Fin de Grado es extender una aplicación web ya creada que permita la realización de desafíos y escritos literarios, añadiendo funcionalidades para ampliar el campo de aspectos que esta puede cubrir. En esta extensión hemos implementado el manejo de peticiones, un sistema de versiones, un nuevo sistema de colecciones para los estudiantes y nuevas vistas. Además también hemos creado nuevas funcionalidades para mejorar la experiencia de usuario como una guía, comentarios y conversiones de los escritos a PDF. Este ha sido evaluado con usuarios de otras facultades para comprobar la experiencia de usuario y ampliar la lista de funcionalidades futuras que se pueden implementar.

## Palabras clave

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

# Abstract

The objective of this Final Degree Project is to extend an already created web application that allows the creation of challenges and writings, adding functionalities to widen the field of aspects that it can cover. In this extension we have implemented request management, a versioning system, a new collection system for students and new views. In addition, we have also created new features to improve the user experience such as a guide, comments and conversions of the writings to PDF. This has been evaluated with users from other faculties to test the user experience and expand the list of future functionalities that can be implemented.

## Keywords

Keywords: Creative Writing, Web Application, ReactJS, NodeJS, ExpressJS, Javascript, Axios, Challenge, SQL

# Índice general

<b>Índice</b>	<b>I</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	1
1.3. Estructura de la memoria . . . . .	2
1.4. Repositorio . . . . .	3
<b>2. Introduction</b>	<b>4</b>
2.1. Motivation . . . . .	4
2.2. Objectives . . . . .	4
2.3. Document structure . . . . .	5
2.4. Repository . . . . .	6
<b>3. Preliminares</b>	<b>7</b>
3.1. Creativa 1.0 . . . . .	7
3.2. Herramientas usadas del proyecto anterior . . . . .	8
3.2.1. Node.js . . . . .	8
3.2.2. ReactJS . . . . .	8
3.2.3. Bases de datos . . . . .	9
3.2.4. Gestión de versiones . . . . .	9
3.3. Bibliotecas . . . . .	10
3.3.1. Multer . . . . .	10
3.3.2. React Draft Wysiwyg . . . . .	10
3.3.3. Axios . . . . .	10
3.3.4. jsPDF . . . . .	10
3.4. Herramientas utilizadas en el presente proyecto . . . . .	11
3.4.1. Visual Studio Code . . . . .	11
3.4.2. Gestión de versiones . . . . .	11
3.4.3. Github Desktop . . . . .	11
3.5. Discord . . . . .	11
3.6. Memoria overleaf . . . . .	12

<b>4. Trabajo relacionado</b>	<b>13</b>
4.1. Reedsy Book Editor . . . . .	13
4.2. Scrivener . . . . .	13
4.3. Ulysses . . . . .	14
4.4. Evernote . . . . .	14
4.5. Conclusión . . . . .	15
<b>5. Funcionamiento de la nueva versión</b>	<b>16</b>
5.1. Funcionalidades adicionales . . . . .	16
5.2. Versiones de escritos . . . . .	17
5.3. Petición de unión a un grupo . . . . .	20
5.4. Solicitudes de unión a un grupo . . . . .	22
5.5. Colecciones (profesor) . . . . .	23
5.5.1. Vista general . . . . .	23
5.5.2. Vista de una colección . . . . .	26
5.6. Colecciones (estudiante) . . . . .	28
5.7. Página de inicio . . . . .	30
5.8. Finalizar escrito . . . . .	31
5.9. PDF . . . . .	33
5.10. Script para la actualización de textos colaborativos . . . . .	35
<b>6. Arquitectura de las extensiones realizadas</b>	<b>36</b>
6.1. Patrón de diseño . . . . .	36
6.2. Estructura de la aplicación . . . . .	37
6.2.1. Back-end . . . . .	37
6.2.2. Front-end . . . . .	38
6.3. Base de datos . . . . .	39
<b>7. Diseño de Creactiva 2.0</b>	<b>41</b>
7.1. Entidades y atributos nuevos o modificados . . . . .	41
7.2. Base de datos SQL . . . . .	43
7.2.1. Tabla versionescrito . . . . .	44
7.2.2. Atributo activo (tabla grupoestudiante) . . . . .	46
7.2.3. Atributo idCreador (tabla mensajería) . . . . .	46
7.2.4. Tabla coleccion . . . . .	47
7.2.5. Tabla colecciondesafio . . . . .	47
7.2.6. Atributo finalizado (tabla escrito) . . . . .	48
<b>8. Contribuciones individuales</b>	<b>49</b>
8.1. Rafael Alcaide Torralvo . . . . .	49
8.1.1. Aprendizaje . . . . .	49
8.1.2. Investigación . . . . .	50
8.1.3. Conexión funcional con el servidor . . . . .	50

8.1.4.	Implementación de las funciones . . . . .	50
8.1.5.	Despliegue de la aplicación . . . . .	52
8.1.6.	Memoria . . . . .	52
8.2.	Verónica Calzada Álvarez . . . . .	53
8.2.1.	Primera reunión . . . . .	53
8.2.2.	Etapas de investigación . . . . .	54
8.2.3.	Base de datos . . . . .	54
8.2.4.	Código . . . . .	54
8.2.5.	Memoria . . . . .	56
8.2.6.	Demo de Creativa 2.0 . . . . .	57
8.2.7.	Despliegue de la aplicación . . . . .	57
8.3.	Jaime Madriñán Fernández . . . . .	58
8.3.1.	Etapas iniciales . . . . .	58
8.3.2.	Etapas de investigación . . . . .	58
8.3.3.	Cambios de la base de datos . . . . .	59
8.3.4.	Código . . . . .	60
8.3.5.	Memoria . . . . .	62
8.3.6.	Demo de Creativa 2.0 . . . . .	62
8.3.7.	Despliegue de la aplicación . . . . .	63
<b>9.</b>	<b>Demo de Creativa 2.0</b>	<b>64</b>
<b>10.</b>	<b>Conclusiones y trabajo futuro</b>	<b>71</b>
10.1.	Conclusiones . . . . .	71
10.2.	Trabajo futuro . . . . .	73
<b>11.</b>	<b>Conclusions and Further work</b>	<b>75</b>
11.1.	Conclusions . . . . .	75
11.2.	Further work on next versions . . . . .	76
	<b>Bibliography</b>	<b>79</b>

# Agradecimientos

## **Rafael**

Primero de todo agradecer tanto a Adrián Riesco como a Cora Requena por su constante interés y apoyo a lo largo del curso. También agradezco el gran desempeño que mis compañeros han realizado por sacar este proyecto adelante. Agradezco también a mi familia y todos los compañeros y profesores con los que podido avanzar a lo largo de estos años dado que han sido un pilar muy fundamental para mí.

## **Verónica**

Quisiera agradecer a mis tutores, Adrián Riesco y Cora Requena, por su disponibilidad, ayuda y gran trabajo a lo largo del desarrollo de proyecto. Asimismo, agradecer también a mis compañeros por su participación. Quiero dar las gracias también a mi entorno, familia y amigos, por su apoyo durante todos estos años en los cuales han estado a mi lado.

## **Jaime**

Para empezar, agradezco a mis tutores Cora y Adrián por su involucración y comprensión durante todo el proyecto. También quiero agradecer a mi compañera Verónica, con la que he trabajado codo con codo en todo momento, lo que nos ha permitido progresar continuamente en el desarrollo del TFG. Por último quiero dar las gracias a mi familia, y en especial a mi madre, por su apoyo incondicional y su continuo interés por mí y mi futuro.

# Capítulo 1

## Introducción

Durante el primer capítulo son tratados los objetivos a cumplir de la aplicación Creativa 2.0, así como la motivación a través de la cual surgen.

### 1.1. Motivación

Los objetivos, que más tarde serán nombrados y explicados, parten de una aplicación desarrollada previamente. Por lo tanto, se trata no solo de mejoras, sino de las sugerencias y dificultades que encontraron sus primeros usuarios, y más tarde, el público hacia el que fue dirigido. Por ello, al encontrarnos con un enfoque que no se adaptaba a las necesidades de los usuarios el proyecto fue variando y reorientándose para la comodidad de todos.

### 1.2. Objetivos

El objetivo sigue siendo el mismo, desarrollar una aplicación donde los alumnos puedan ejercer de forma dinámica la escritura creativa; pero, ahora, ofreciéndoles más recursos. Para ello, previamente se hizo funcionar Creativa 1.0, se comprendieron los objetivos decididos en su momento y se arreglaron los fallos con los que contaba. Los nuevos objetivos para esta ampliación son:

- Ampliar las posibilidades de escritura colectiva, es decir, realizar una ampliación donde se observen los cambios en tiempo real.

- Crear bloques temáticos para plantear la nueva sección de desafíos: ej. un bloque de poesía, relatos cortos, otro de imagen y palabra, etc. Todos estos cambios están enfocados a la BBDD.
- Mejorar la usabilidad de Creactiva 2.0.
- Hacerlo más amigable, es decir, una actualización que mejore la UX.

Dado que el objetivo final sería obtener una aplicación que gustase al público, se decidió al final del proyecto realizar una demo de Creactiva 2.0, para ver su acogida por los usuarios e implementar las nuevas funcionalidades que necesitasen los alumnos.

### 1.3. Estructura de la memoria

Esta memoria está agrupada en diferentes capítulos. Cada uno de ellos abarca un área o una etapa distinta del proyecto:

- En el capítulo 2 se encuentra la versión en inglés de esta introducción.
- En el capítulo 3 se encuentra un resumen de cómo funciona Creactiva 1.0 y se explican las herramientas empleadas para la realización del proyecto.
- En el capítulo 4 se encuentran las aplicaciones dirigidas a la escritura creativa.
- En el capítulo 5 se recoge una explicación de cómo usar la aplicación y nuevas funcionalidades.
- En el capítulo 6 se explica la arquitectura de la aplicación.
- En el capítulo 7 se encuentra el diseño de Creactiva 2.0.
- En el capítulo 8 se recogen la aportaciones individuales.
- En el capítulo 9 se encuentra el análisis de la demo realizada.

- En el capítulo 10 se encuentran las conclusiones y el trabajo futuro.
- En el capítulo 11 se encuentra la versión en inglés del capítulo 10.

## 1.4. Repositorio

El código de la aplicación web Creativa 2.0 se encuentra en el repositorio

[https://github.com/JaimeMadFer/EscrituraCreativa\\_TFG](https://github.com/JaimeMadFer/EscrituraCreativa_TFG) bajo licencia MIT.

# Capítulo 2

## Introduction

In our first chapter, the objectives to accomplish from the app Creactiva 2.0 are treated, as well as the motivation through which they emerge.

### 2.1. Motivation

The objectives, that later on will be named and explained, emerge from an app already started. That is why, this is not only about updates, but suggestions, difficulties and obstacles that the beta users found. That is why we found ourselves with an overview of the app that did not match the necessities that was supposed to cover, so the project kept evolving to engage everyone.

### 2.2. Objectives

The objective is still the same, to develop an application where the students can put into practice the creative writing, but now offering them more features. For that purpose, we previously run Creactiva 1.0, in order to understand the objectives decided at the beginning and to fix the failures that it had. The new objectives for this update are:

- Update the possibilities of group writing, that is, to make an extension where changes can be seen in real time.

- To create different topics in order to being able to make a brand new section of challenges: e.g. poetry topic, short tales, etc. All these changes will be made in the database.
- Update the usability of Creactiva 2.0.
- Make a better UX, make it more user-friendly.

Due to the fact that the final goal would be to obtain an app that everybody liked, we decided to realize a demo of Creactiva 2.0, so as to see the users response and to implement the new funcionality that the students wuold may need.

## 2.3. Document structure

The document is group into different chapters. Each one covers separate areas or phases from the project:

- Chapter 3: abstract showing how Creactiva 1.0 works and explanation of the tools used during the development.
- Chapter 4: apps focused on creative writing.
- Chapter 5: on how to use the application and new funcionality.
- Chapter 6: architecture of the application.
- Chapter 7: design of Creactiva 2.0.
- Chapter 8: individual work.
- Chapter 9: demo analytics.
- Chapter 10: concluding remarks and further work.
- Chapter 11: English version of chapter 10.

## 2.4. Repository

The code of this application can be found in the following url

[https://github.com/JaimeMadFer/EscrituraCreativa\\_TFG](https://github.com/JaimeMadFer/EscrituraCreativa_TFG) under MIT license.

# Capítulo 3

## Preliminares

En este apartado presentamos un breve resumen sobre Creativa 1.0 (proyecto desarrollado el año pasado) y sobre las herramientas utilizadas a lo largo del proyecto.

### 3.1. Creativa 1.0

El objetivo de Creativa 1.0 era crear una aplicación la cual promoviera la escritura creativa. Para esto Creativa 1.0 cuenta con un sistema de usuarios y un sistema de administración de usuarios que permite la creación de escritos los cuales permiten almacenar multimedia.

El proyecto promueve la actividad creativa, y cuenta con varias funcionalidades según el tema que se trate. Posee un sistema de registro de usuarios, a los cuales se les puede editar el perfil.

Se pueden crear desafíos que son textos narrativos propuestos para los alumnos, aparte se pueden editar, listar y eliminar.

También tiene implementado un apartado de Escritos, que son los relatos los cuales siguen las normas de los desafíos que el profesor crea, puede ser individual o colaborativo. Estos también se pueden editar, listar, modificar y eliminar.

Tiene un apartado donde agrupar a los alumnos según convenio del profesor y un apartado de Equipos donde aquellos alumnos que formen parte del mismo equipo realizarán el mismo escrito. Y por último un sistema de mensajería para poder enviar solicitudes de unión

de grupo.

## 3.2. Herramientas usadas del proyecto anterior

Al trabajar sobre un proyecto ya empezado, para no complicar el trabajo hemos vuelto a usar varias de las herramientas, de tal forma que muchas de ellas las hemos tomado para poder ampliar el proyecto sin que tengamos problemas de compatibilidad. Estas son las siguientes.

### 3.2.1. Node.js

*Node.js*<sup>1</sup> es un entorno de tiempo de ejecución de *JavaScript* (de ahí su terminación en .js, haciendo alusión al lenguaje *JavaScript*). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en *JavaScript*.

Hemos optado por esta tecnología porque, en primer lugar y como gran ventaja, nos permite tener *JavaScript* incorporado en la plataforma *Node.js*, siendo un lenguaje fácil de aprender y que puede ser manejado por programadores de Java. Posee un modelo de entrada y salida impulsado por eventos, ayuda mucho en el manejo simultáneo de peticiones. También lo elegimos porque está orientado a eventos asíncronos, de esta forma la aplicación es más eficaz dado que el número de bloqueos es menor aumentando así su rendimiento.

### 3.2.2. ReactJS

*ReactJS*<sup>1</sup> es una biblioteca de *JavaScript* para construir interfaces de usuario. Esta biblioteca ayuda a crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en tu aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien. Ya que la lógica de los componentes está escrita en *JavaScript* y no en plantillas, es posible pasar datos de forma sencilla a través de la aplicación y mantener el estado fuera del DOM. Crea componentes encapsulados que manejen su propio estado en interfaces de usuario complejas, de esta forma

el trabajo en equipo es mucho fácil dado que los componentes se pueden copiar y pegar en otras secciones que tengan la misma y reutilizar código.

### 3.2.3. Bases de datos

Para la gestión de la base de datos hemos usado las siguientes tecnologías.

#### Xampp

Para la base de datos hemos utilizado *Xampp*<sup>2</sup> que provee de manera local un servidor web en el cual podemos almacenar los datos, este incluye *MySql*<sup>3</sup> que es un gestor de bases de datos. Permite trabajar de forma rápida al no requerir tiempo de ejecución.

Xampp también funciona de manera independiente y es de código libre que incluye bases de datos MySQL.

#### MySql

MySQL es un sistema de gestión de bases de datos relacionales, funciona prácticamente en todas las plataforma que se basa en un modelo cliente-servidor. El núcleo de MySQL es el servidor MySQL, que maneja todas las instrucciones (o comandos) de la base de datos.

### 3.2.4. Gestión de versiones

Para el control de versiones hemos usado las siguientes herramientas.

#### Git

Git<sup>4</sup> es un sistema de control de versiones. Esto significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota fácilmente. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan su copia del repositorio con la copia en el servidor.

## Github

Github<sup>5</sup> es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador. El uso que nosotros le hemos dado es de repositorio. Dado que muchas aplicaciones como Visual Studio Code son compatibles con Github el manejo y alojamiento de datos en estos repositorios son muy fáciles y eficaces, pudiendo controlar las versiones y cambios en el proyecto.

## 3.3. Bibliotecas

Las bibliotecas que hemos usado en el proyecto se detallan a continuación.

### 3.3.1. Multer

Multer<sup>6</sup> es un *middleware* para Express y Node.js, es un módulo con la funcionalidad de gestionar de los archivos multimedia en el lado del servidor ante, añade ficheros a un desafío o a un escrito se suben a un directorio `./public/multimedia` del servidor.

### 3.3.2. React Draft Wysiwyg

React Draft Wysiwyg<sup>7</sup> es una biblioteca de ReactJs diseñada para el control de estilos de texto y el manejo de estos.

### 3.3.3. Axios

Axios<sup>8</sup> es una biblioteca de Javascript que permite realizar peticiones HTTP a un servidor, siendo su utilidad en el proyecto el manejo de peticiones de cliente-servidor

### 3.3.4. jsPDF

*jsPDF*<sup>9</sup> es una biblioteca de Javascript que permite transformar un escrito a formato PDF y guardarlo para su impresión. Se ha utilizado en el lado del cliente para la transformación de los escritos.

## 3.4. Herramientas utilizadas en el presente proyecto

Estas son las nuevas herramientas que hemos usado para la ampliación del proyecto.

### 3.4.1. Visual Studio Code

*Visual Studio Code*<sup>10</sup> es un editor de código con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su interfaz de usuario es común a varios entornos que usamos durante el curso y resulta útil para el desarrollo de páginas web que implementan distintos lenguajes. A la hora del trabajo en equipo, al tener varias extensiones y al ser compatible con GitHub, se puede trabajar en él de forma simultánea y subir los cambios a nuestros repositorios.

### 3.4.2. Gestión de versiones

Nos hemos basado en el modelo de control de versiones del año pasado [3.2.4](#) dado que las herramientas usadas son las más eficientes, en nuestra ampliación además hemos añadido lo siguiente.

### 3.4.3. Github Desktop

*Github Desktop*<sup>11</sup> es una interfaz gráfica para Git y se puede conectar con Github que permite hacer todas las operaciones de gestión del control de versiones desde una aplicación de escritorio.

## 3.5. Discord

*Discord*<sup>12</sup> es una aplicación que permite crear servidores permanentes dentro de ella donde nos podemos comunicar por voz y donde también nos podemos transferir archivos de poco peso para no tocar el proyecto.

## 3.6. Memoria overleaf

*Overleaf* <sup>13</sup> es un editor en línea que permite trabajar de forma colaborativa y ha sido empleado para la elaboración de la memoria. El cual se escribe en latex que es un lenguaje para editar textos.

# Capítulo 4

## Trabajo relacionado

A continuación compararemos algunas aplicaciones o servicios que comparten parecido con nuestro proyecto.

### 4.1. Reedsy Book Editor

*Reedsy Book Editor*<sup>14</sup> es una herramienta de producción que se encarga del formato y la conversión, incluso antes de que el escrito no haya sido terminado de escribir. Se pueden arrastrar y soltar capítulos, insertar imágenes y hacer un seguimiento de los cambios para ver las versiones anteriores de tu trabajo. Tiene la posibilidad de activar un sistema integrado de recordatorio de objetivos. Los puntos que comparten con nuestra proyecto son principalmente la facilidad y simplicidad a la hora de escribir dado que es fácil aplicar estilos de escritura, la escritura colaborativa, la distribución, ofreciendo descargas en varios formatos, en nuestro caso PDF, posibilidad de comentar los escritos y poseer un sistema de versiones.

### 4.2. Scrivener

*Scrivener*<sup>15</sup> está orientada a novelistas y otros escritores que desean una interfaz elaborada y organizada. Se puede elegir una plantilla para el proyecto (novela, ensayo, guión, etc.) facilitando la organización. La barra lateral izquierda incluye secciones y fichas virtuales pa-

ra todos los elementos, y se pueden organizar como se desee. Otras funciones más avanzadas te permiten hacer un seguimiento de investigaciones, crear materiales especializados como material de portada y de contraportada, e incluso analizar el contenido del propio texto. El parecido que comparte principalmente con nuestro proyecto es el objetivo, la escritura creativa. Además de que posee un sistema de archivación para poder agrupar los escritos según los temas o tópicos en caso de la aplicación. También se pueden descargar los archivos en PDF y subir archivos multimedia a los mismos.

### 4.3. Ulysses

*Ulysses*<sup>16</sup> está orientada principalmente a escritores que quieren mejorar la productividad de su formato. Es una aplicación de escritura bastante estándar; no es tan fácil de usar como Scrivener, dado que usa su propio lenguaje para la creación de apartados pero ofrece mucho más potencial, proporcionando palabras clave, capacidad de vista dividida, seguimiento del progreso y todos los proyectos alineados en la barra lateral. Ulysses está centrado en la escritura creativa orientada a la distribución, es decir, su objetivo es la vista y arquitectura final del escrito para así distribuirlo, para esto tiene una funcionalidad que permite adaptar perfectamente los escritos a PDF sin ninguna pérdida y ninguna modificación.

### 4.4. Evernote

*Evernote*<sup>17</sup> posee un sistema de plantillas para todo, desde la toma de notas en clase hasta la planificación personal o la estructuración de una novela. Además de las abundantes plantillas organizativas, Evernote también permite etiquetar todo en categorías específicas, compartir notas con colaboradores e incluso chatear con ellos en la aplicación. Una de sus características es el Web Clipper, con el que se puede guardar cualquier fragmento de contenido web que interese. Está diseñada para ser adaptable y accesible para todo tipo de escritores y sus proyectos. Esta aplicación fomenta la escritura creativa pero centrada más en la parte colaborativa dado que intenta crear escritos a partir de otros escritos y/o

ideas, aparte de que estos pueden comentar y posee estilos de texto.

### 4.5. Conclusión

A continuación mostraremos una tabla comparando nuestra aplicación con las demás. En esta tabla mostraremos los campos que hemos cubierto en Creactiva 2.0 y los compararemos con las demás aplicaciones ya nombradas antes.

	Estilos	Escritura Colaborativa	Comentarios	Temas	PDF	Versiones
Creativa 2.0	Sí	Sí	Sí	Sí	Sí	Sí
Reedsy Book Editor	Sí	Sí	Sí	No	Sí	Sí
Scrivener	Sí	No	No	Sí	Sí	No
Ulysses	Sí	No	No	Sí	Sí	Sí
Evernote	Sí	Sí	Sí	Sí	No	No

# Capítulo 5

## Funcionamiento de la nueva versión

La expansión de esta página web tiene varias mejoras, muchas de ellas independientes entre sí. Estas mejoras se introducen en el apartado 5.1, y se desarrollan a partir del apartado 5.2.

### 5.1. Funcionalidades adicionales

**Versiones de escritos:** Se originó con la idea de crear un diseño mas amigable, empezando por la modificación de la base de datos para poder almacenar las distintas versiones, así como posteriormente la vista para añadir la posibilidad de usar esta nueva funcionalidad.

**Unión a grupo:** En esta funcionalidad que ya estaba implementada en la versión anterior, añadimos la posibilidad de que el propio estudiante pudiera solicitar la unión a un grupo, así como su vista correspondiente, la cual sigue teniendo que ser aceptada por un profesor para que el estudiante este dentro del grupo.

**Colecciones:** Se origino para mejorar la organización de los desafíos así como para una categorización de los grupos. Es una opción que mejora la usabilidad de la aplicación. Se implementó como una opción en el menú lateral de la vista, el cual te lleva a su propia sección para el visionado de las colecciones que selecciones así como los escritos y desafíos de dicha colección.

**Página de inicio:** La creación de una vista principal así como una pagina que explica las distintas funcionalidades y guía de uso de la aplicación. Se implemento como pagina a la que se entra según haces login en la aplicación y se le añadieron hipervínculos y apartados para un uso mas sencillo.

**Finalizar escrito:** Esta funcionalidad estaba planteada ya en la aplicación y así como la existencia en la base de datos de un campo que marcaba la fecha de finalización de un texto. Por lo tanto nuestra implementación continuo desde ese punto añadiendo en la aplicación así como en la vista de un texto la posibilidad, mediante un botón, de finalizar el texto y que se quedase cerrado para poder ser visionado y valorado por los profesores.

**PDF:** Para implementar la transformación de escritos a formato PDF, se tuvo en cuenta una búsqueda definida que finalizo con la implementación de la biblioteca jsPDF con sus funcionalidades y personalizacion para dicha función.

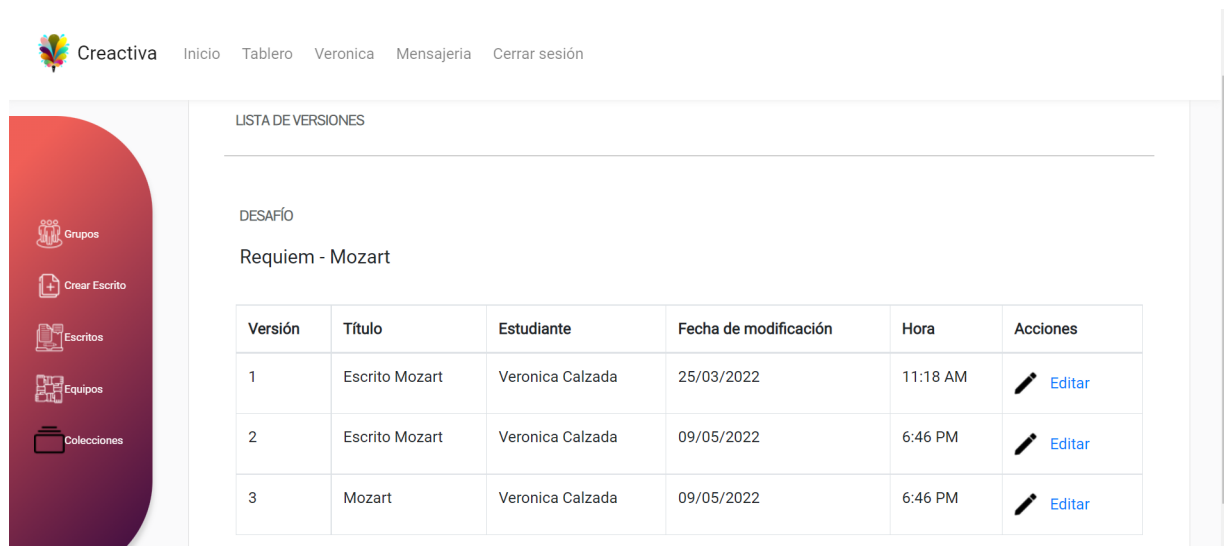
**Script para la actualización de textos colaborativos:** Para un próximo despliegue de la aplicación desarrollamos un script que actualiza en un margen de 1-2 segundos el texto de los escritos conjuntos con todos los usuarios que estén participando al mismo tiempo en dicho escrito, por ahora solo funciona de manera local.

## 5.2. Versiones de escritos

Con el fin de que los usuarios con rol de estudiante de la plataforma puedan encontrar en Creativa 2.0 un diseño más amigable, intentamos que la realización de escritos se asemeje a otras aplicaciones de escritura, no necesariamente creativa, que ya hubiesen utilizado con anterioridad. Para ello, implementamos este cambio que permite el almacenamiento de versiones cada vez que se guarde un escrito. De este modo, pueden acceder a escritos de versiones anteriores si en algún momento los quieren restaurar. Con esto conseguimos que no se pierdan los cambios que los estudiantes descartan al guardar un escrito con un texto




nuevo.

Todas las versiones de un escrito se podrán ver en una página como la que se muestra en la figura 5.1. Como se muestra en la siguiente figura, el usuario debe estar previamente en la vista de editar un escrito, y pulsar a continuación el botón ‘Acceder a versiones anteriores’, situado al final de la página.



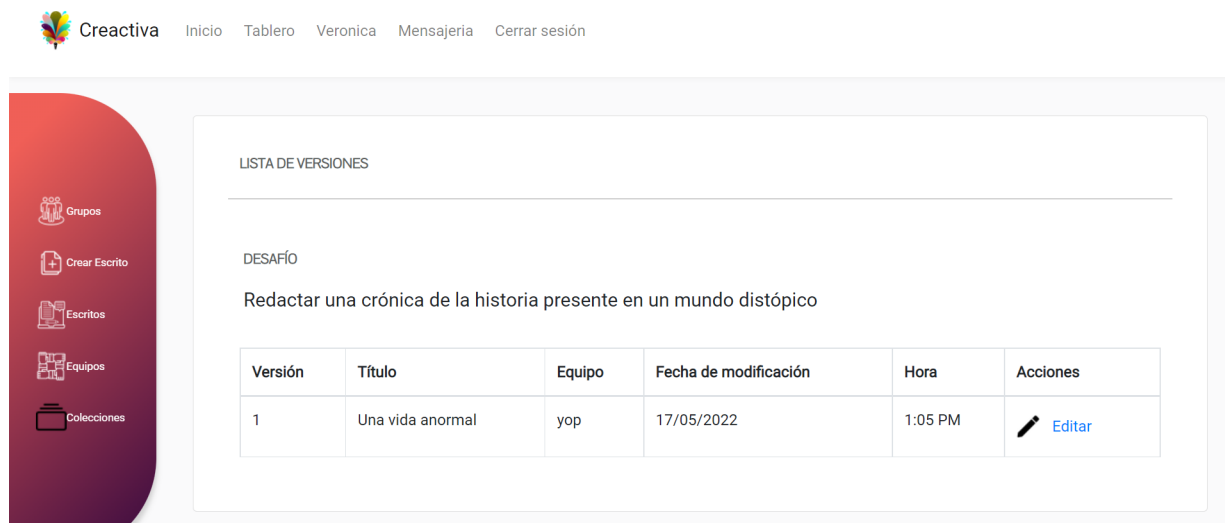
The screenshot shows the 'Creaactiva' interface. At the top, there is a navigation bar with the logo and links for 'Inicio', 'Tablero', 'Veronica', 'Mensajería', and 'Cerrar sesión'. On the left, a sidebar contains icons for 'Grupos', 'Crear Escrito', 'Escritos', 'Equipos', and 'Colecciones'. The main content area is titled 'LISTA DE VERSIONES' and displays the following information:

DESAFÍO  
Requiem - Mozart

Versión	Título	Estudiante	Fecha de modificación	Hora	Acciones
1	Escrito Mozart	Veronica Calzada	25/03/2022	11:18 AM	 <a href="#">Editar</a>
2	Escrito Mozart	Veronica Calzada	09/05/2022	6:46 PM	 <a href="#">Editar</a>
3	Mozart	Veronica Calzada	09/05/2022	6:46 PM	 <a href="#">Editar</a>

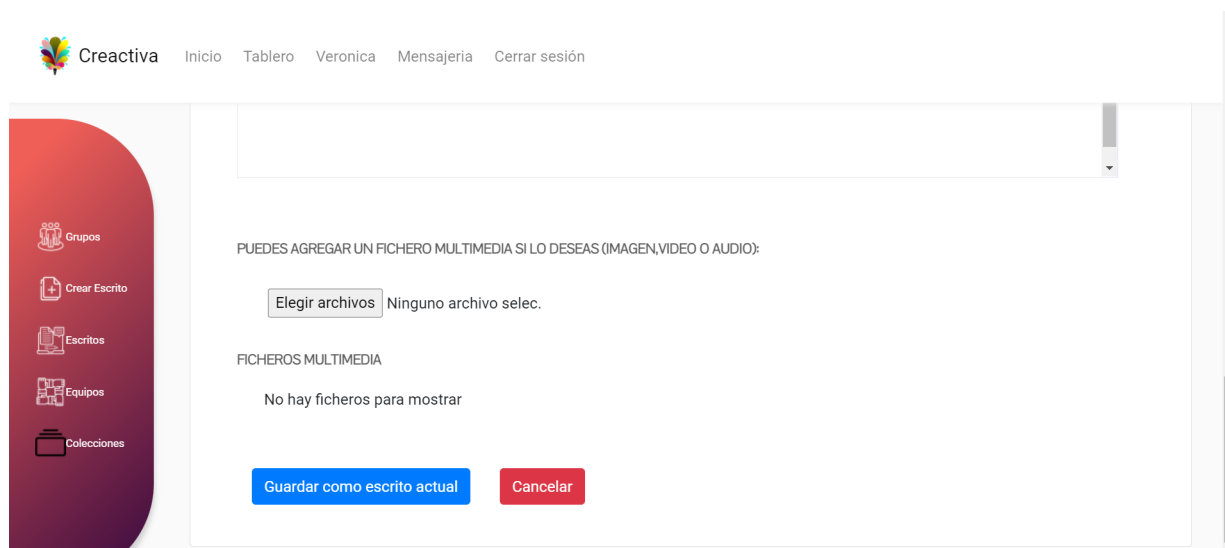
**Figura 5.1:** *Listado de versiones de un escrito individual*

Lo mismo ocurre para los escritos colaborativos, como se puede observar en la figura 5.2.




**Figura 5.2:** *Listado de versiones de un escrito colaborativo*

Cada versión de un escrito se podrá modificar de la misma manera que se hacía cuando solo había una única versión. Con lo cual, si pulsamos en 'Editar', nos llevará a la figura 5.3.







**Figura 5.3:** *Botón para guardar la inserción de la versión de un escrito*

Si guardamos el escrito como el actual, veremos cómo se añade una nueva versión para ese escrito, independientemente de la versión que hayamos editado. El resultado de esta acción se ve reflejado en la figura 5.4, tomando como referencia el escrito de la figura 5.1.


[Inicio](#)
[Tablero](#)
[Veronica](#)
[Mensajería](#)
[Cerrar sesión](#)

DESAFÍO

Requiem - Mozart

Versión	Título	Estudiante	Fecha de modificación	Hora	Acciones
1	Escrito Mozart	Veronica Calzada	25/03/2022	11:18 AM	 <a href="#">Editar</a>
2	Escrito Mozart	Veronica Calzada	09/05/2022	6:46 PM	 <a href="#">Editar</a>
3	Mozart	Veronica Calzada	09/05/2022	6:46 PM	 <a href="#">Editar</a>
4	Escrito Mozart	Veronica Calzada	17/05/2022	1:03 PM	 <a href="#">Editar</a>

**Figura 5.4:** Listado de versiones de escritos después de la inserción de una nueva versión

### 5.3. Petición de unión a un grupo

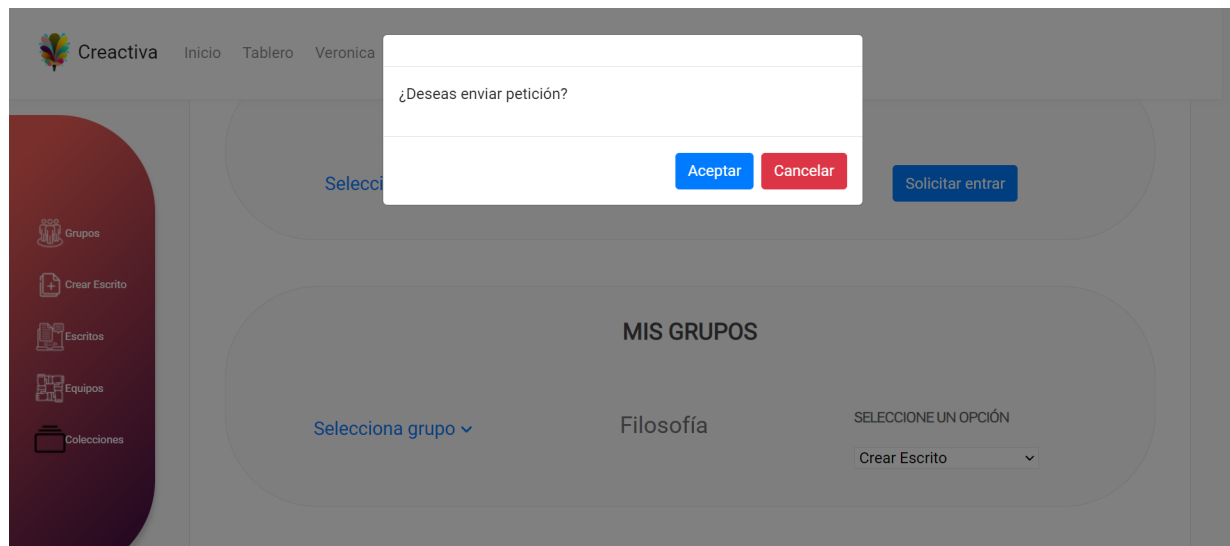
Los estudiantes antes tenían que esperar a ser invitados por el profesor para entrar en un grupo, que cumplía la función de asignatura dentro de la plataforma Creativa 2.0, de ahí la potestad del profesor. En cambio, ahora, con esta nueva funcionalidad, el propio estudiante puede directamente enviarle su solicitud de unión a un grupo al profesor.

El proceso para realizar la petición sería el siguiente: En el menú lateral ‘Grupos’, primero se selecciona el grupo al que se quiere entrar en el desplegable ‘Selecciona grupo’. En este desplegable se mostrarán los grupos a los que no pertenece el estudiante, pero no aparecerán los grupos en los que se esté procesando una petición de unión. Posteriormente, se pulsa el botón ‘Solicitar entrar’, lo cual nos lleva a la figura 5.6. En la figura 5.5 se puede observar como se seguiría el proceso de petición.



**Figura 5.5:** *Desplegable para seleccionar grupo en el proceso de enviar petición a un grupo*

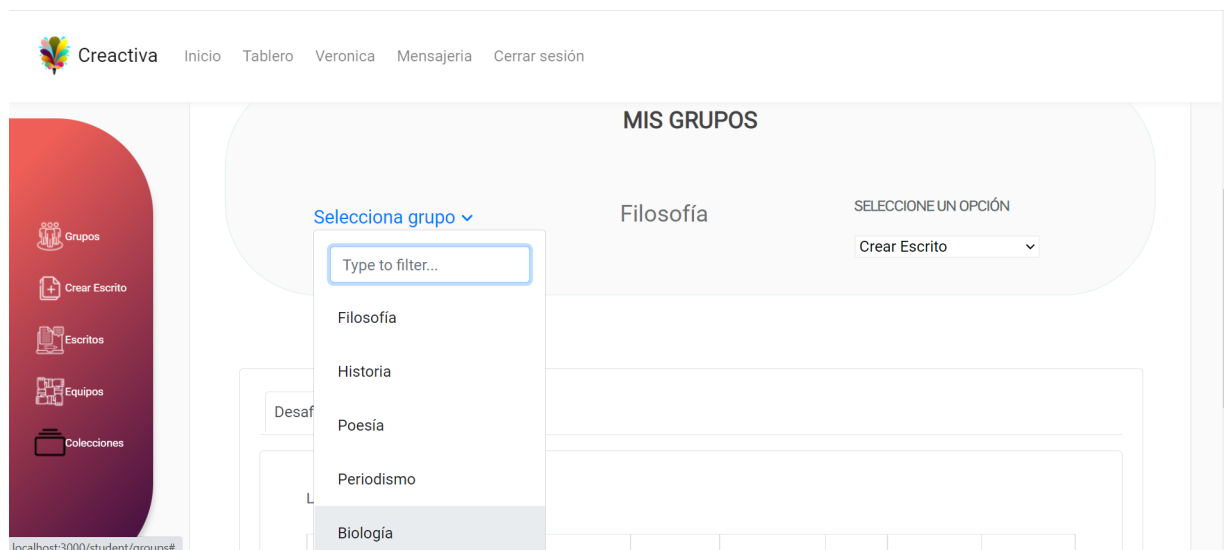
Al pulsar el botón ‘Solicitar entrar’ aparecerá un modal como el que se ve en la figura 5.6. Cuando se pulse en ‘Aceptar’, se enviará la solicitud al profesor.



**Figura 5.6:** *Confirmación para enviar petición de unión a un grupo*

Como ya mencionamos, al enviar la petición, el grupo solicitado ya no aparecerá en el desplegable de ‘ENTRAR A NUEVOS GRUPOS’, pero tampoco saldrá en el listado del desplegable de ‘MIS GRUPOS’, ya que falta la confirmación del profesor para admitir al

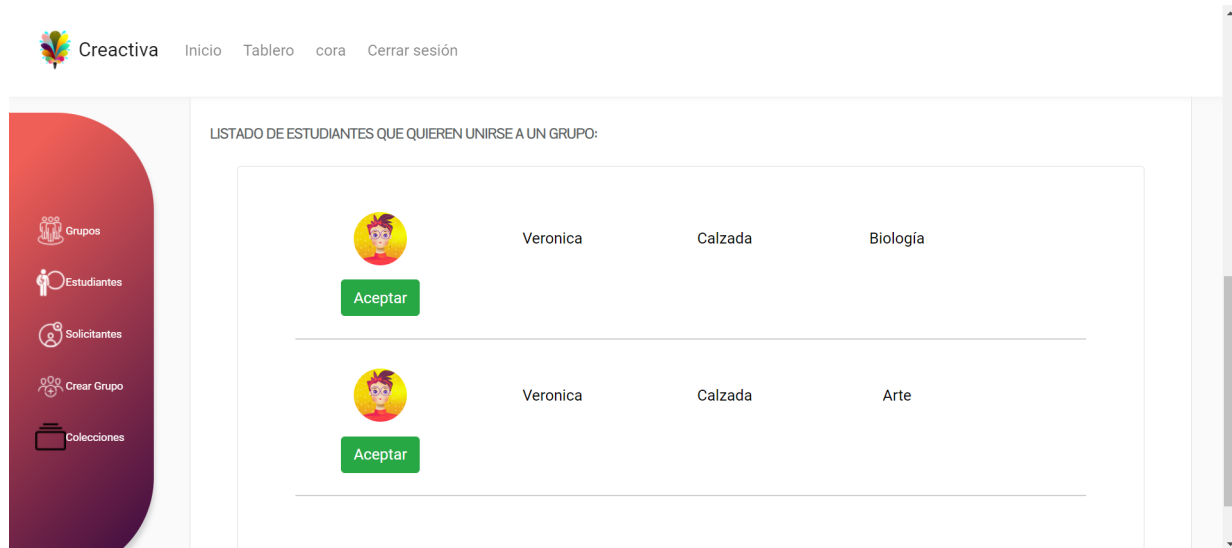
estudiante en el grupo. En la figura 5.8 el profesor aceptaría la petición, y entonces el alumno ya estaría dentro del grupo, tal y como se muestra en la figura 5.7.



**Figura 5.7:** *Comprobación de la correcta incorporación a un grupo*

## 5.4. Solicitudes de unión a un grupo

Como usuario con rol de profesor, llegarán por parte de los alumnos un número de peticiones para entrar en un grupo que se tendrán que gestionar. Esta gestión se realizará en el menú lateral 'Solicitantes'. Por tanto, ahora en vez de recibir en este menú solo las solicitudes de las personas que quieran unirse a la plataforma, también se recibirán las solicitudes de los estudiantes para unirse a algún grupo. Como se aprecia en la figura 5.8, esto viene especificado por una frase que lo explica. En caso de tener solicitudes de ambos tipos, la página quedará dividida en dos para que sea más inteligible.



**Figura 5.8:** *Listado de solicitudes de unión a un grupo*

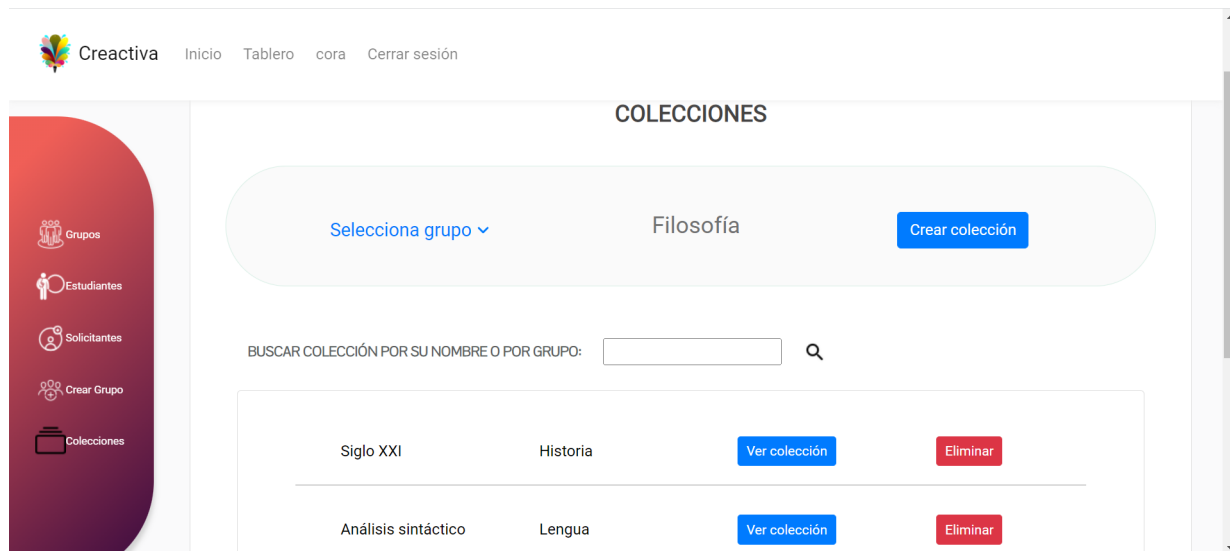
Si el profesor acepta una petición, el estudiante formará ya parte del grupo como se muestra en la figura 5.7. Y los grupos aparecerán en la sección ‘Mis grupos’ del estudiante aceptado.

## 5.5. Colecciones (profesor)

Con el objetivo de organizar mejor los desafíos se creó un nuevo menú lateral, las ‘Colecciones’. Se pensó en esta nueva utilidad para así categorizar los grupos y llevar un mejor control de los distintos tipos de desafíos que hay pudiendo observar también los escritos de estos, finalizados o no.

### 5.5.1. Vista general

Cada categoría es lo que denominamos colección, y su listado está en la pestaña de ‘Colecciones’, como se observa en la figura 5.9.

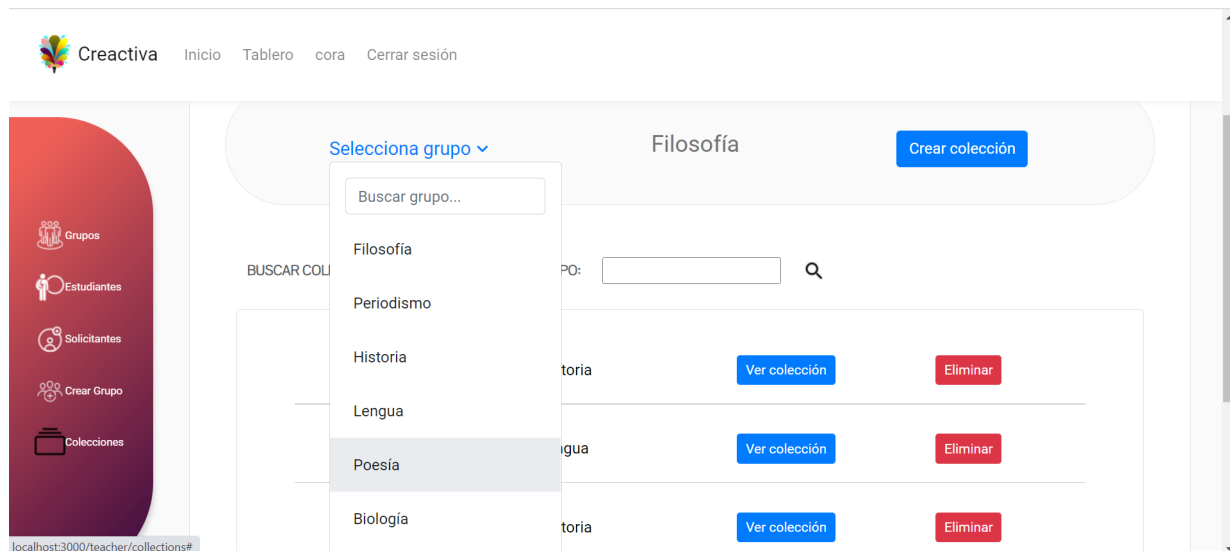


**Figura 5.9:** *Listado de colecciones de un profesor*

Para cada colección que se lista en la figura 5.9 se permiten dos opciones:

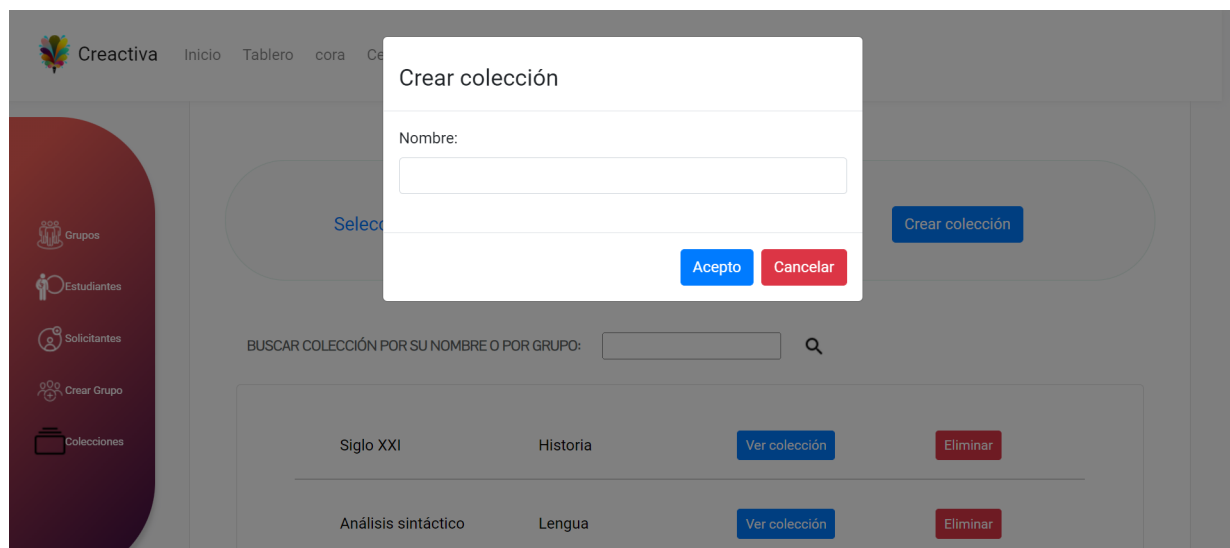
- Eliminarla y que desaparezca de la lista, pulsando el botón ‘Eliminar’.
- Ver en detalle el contenido de la colección, que nos llevará a la vista que indica la imagen 5.13, pulsando el botón ‘Ver colección’. En la colección se encuentran los desafíos que elige el profesor dueño de dicha colección.

Además, se proporciona un buscador que permite al usuario encontrar una colección introduciendo el nombre de la propia colección o el grupo al que pertenece (ver figura 5.9). También existe la opción de crear una nueva colección. Para crearla, se comienza seleccionando el grupo al que pertenecerá dicha colección, de la cual el profesor tiene que impartir la materia. Esto se puede observar en la figura 5.10.



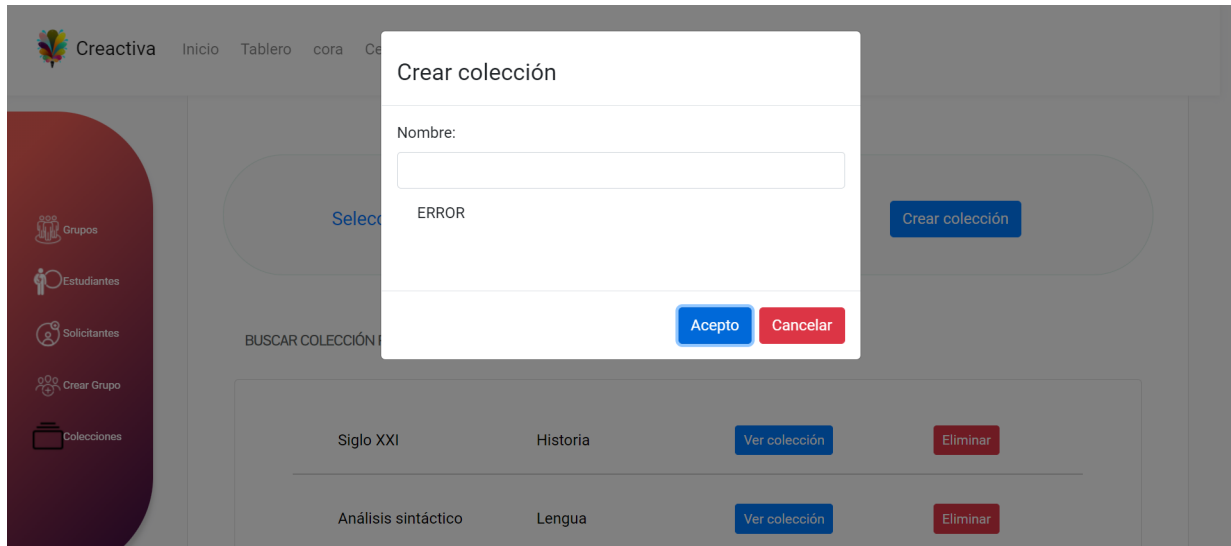
**Figura 5.10:** *Desplegable para seleccionar grupo en el proceso de creación de una colección*

Después se pulsa en el botón ‘Crear colección’, y aparecerá el modal de la figura 5.11.



**Figura 5.11:** *Campo de texto para introducir el nombre de una nueva colección*

Si no se introduce ningún nombre y se pulsa el botón ‘Acepto’, saldrá un mensaje de error en el propio modal como se muestra en la figura 5.12. En cualquier otro caso, el nombre de la colección será válido.



**Figura 5.12:** Mensaje de error al introducir un nombre vacío para crear una colección

### 5.5.2. Vista de una colección

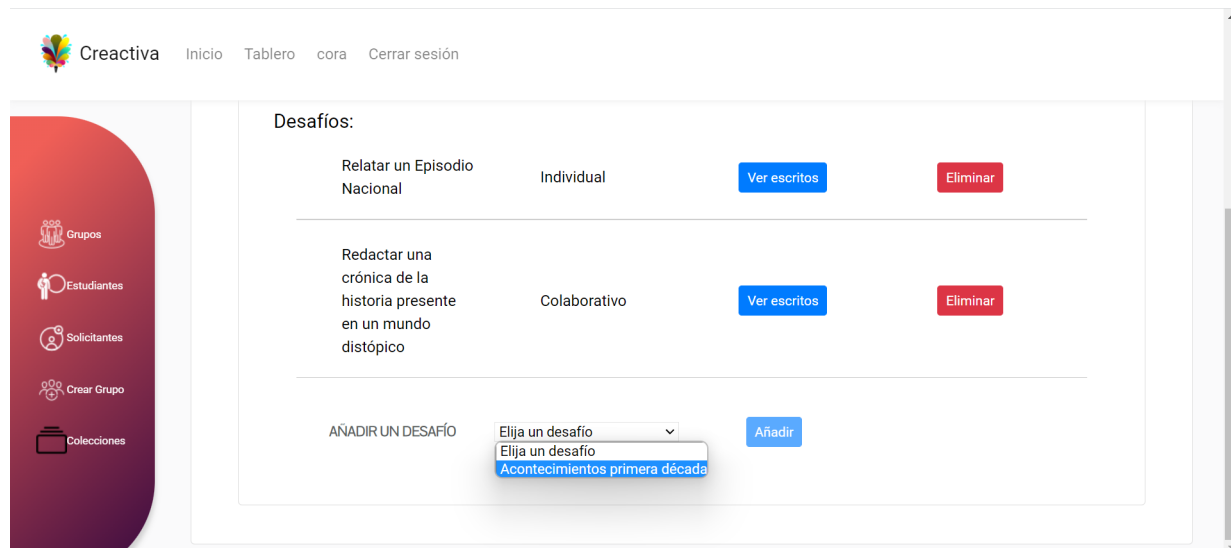
Desde la figura 5.9 se puede acceder a la vista de una colección concreta, donde su aspecto es el que muestra la figura 5.13.



**Figura 5.13:** Listado de desafíos de una colección

La colección dispone de un listado de desafíos, que pueden ser añadidos y eliminados. Para eliminar un desafío hay que pulsar el botón 'Eliminar' sobre el desafío de la lista

que queramos. Por otro lado, para añadir un desafío, hay que seleccionar un desafío en el desplegable de la figura 5.14, y después pulsar el botón 'Añadir'.



**Figura 5.14:** Desplegable para añadir desafíos a una colección

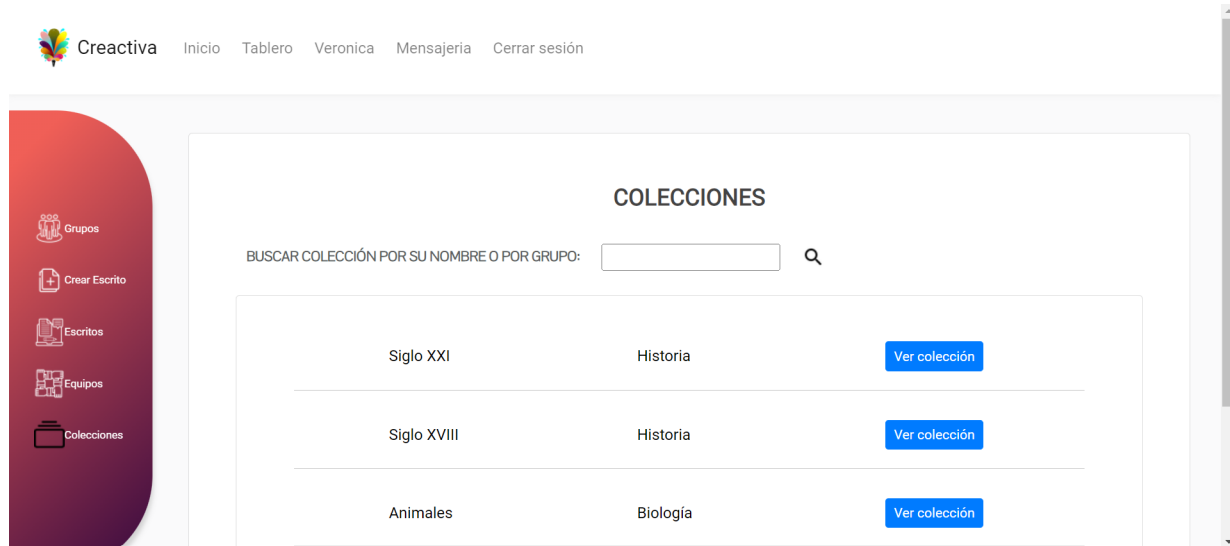
Por último, se pueden ver todos los escritos de los estudiantes que responden a un desafío y están vinculados a la colección creada por el profesor pulsando el botón 'Ver escritos'. Esta acción nos llevará a la figura 5.15.



**Figura 5.15:** Listado de escritos de un desafío perteneciente a una colección

## 5.6. Colecciones (estudiante)

Esta vista permite al estudiante ver las colecciones que crea el profesor. La página tiene el mismo formato que las colecciones del profesor, con la excepción de que los estudiantes no pueden modificar las colecciones ni los desafíos que hay en ellas. En la figura 5.16 se muestra la página principal de las colecciones.



**Figura 5.16:** *Listado de colecciones de un estudiante*

Pulsando el botón ‘Ver colección’ accederíamos a la colección para verla en detalle, y puede ser que no haya desafíos activos para un grupo, y por lo tanto tampoco habría desafíos para la colección (ver figura 5.17). En caso de haber desafíos, estos le ilustrarían como en la figura 5.18.



**Figura 5.17:** *Mensaje informativo de una colección sin desafíos*



**Figura 5.18:** *Listado de una colección con desafíos*

Por último, la vista de los escritos para cada desafío (a la que se accede pulsando el botón 'Ver escritos') es igual que la de las colecciones del profesor, ya que esta vista no tiene la opción de hacer ninguna modificación, es meramente informativa (ver figura 5.15).

## 5.7. Página de inicio

Tras realizar la demo con un grupo de prueba para ver cómo de útil encontraban la aplicación, se nos propuso la implementación de una página de inicio que explicase cómo navegar por la aplicación, dónde encontrar cada funcionalidad y cuál era el objetivo de Creativa 2.0.

Inicio se encuentra en la barra superior, y es lo primero que encontraremos nada más acceder a la aplicación. Esta página se muestra en la figura 5.19.



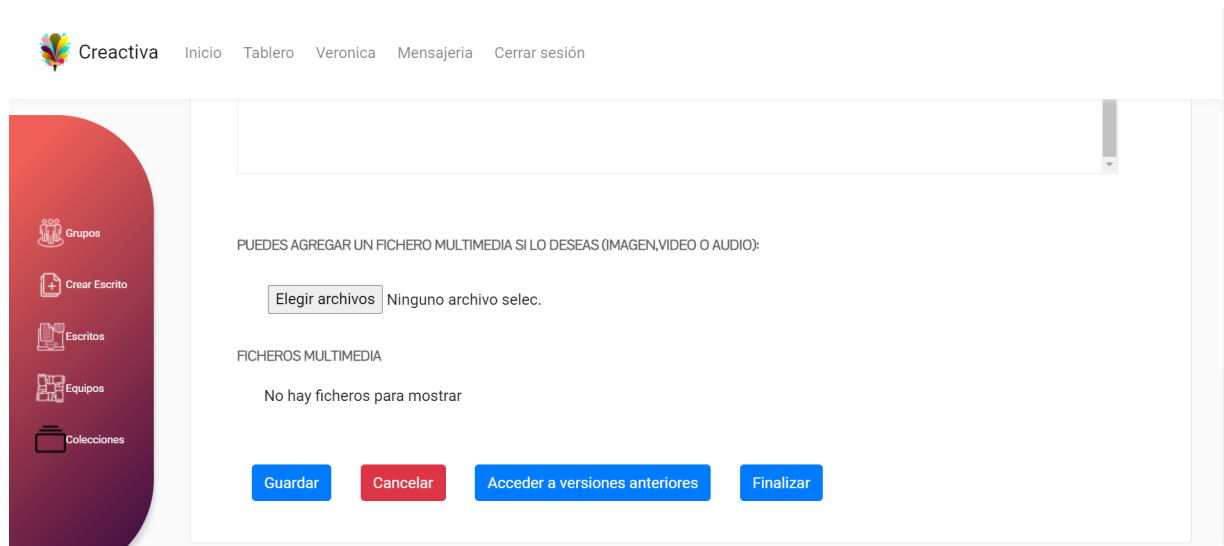
**Figura 5.19:** *Página de inicio*

Los títulos cuentan con hipervínculos que redirigen al usuario a la sección de la que se ofrece la explicación para que sea más fácil y dinámico. Cabe destacar que cada rol de usuario tendrá una página de inicio algo distinta. La sección ‘Tablero’ es la principal diferencia, ya que cada usuario tiene distintos subapartados dentro de esta sección. Por ejemplo, los dos primeros subapartados de los estudiantes son ‘Grupos’ y ‘Crear escrito’, por lo que aquí se explicará cómo funcionan estos subapartados, mientras que los de los profesores son ‘Grupos’ y ‘Estudiantes’, que tendrán una explicación muy distinta. Además, el subapartado ‘Grupos’ en este caso tiene un funcionamiento completamente distinto para los estudiantes y para los profesores. La otra diferencia es que solo los estudiantes tienen la sección de ‘Mensajería’,

ya que esta se usa para enviar mensajes entre estudiantes que pueden ser añadidos a un equipo.

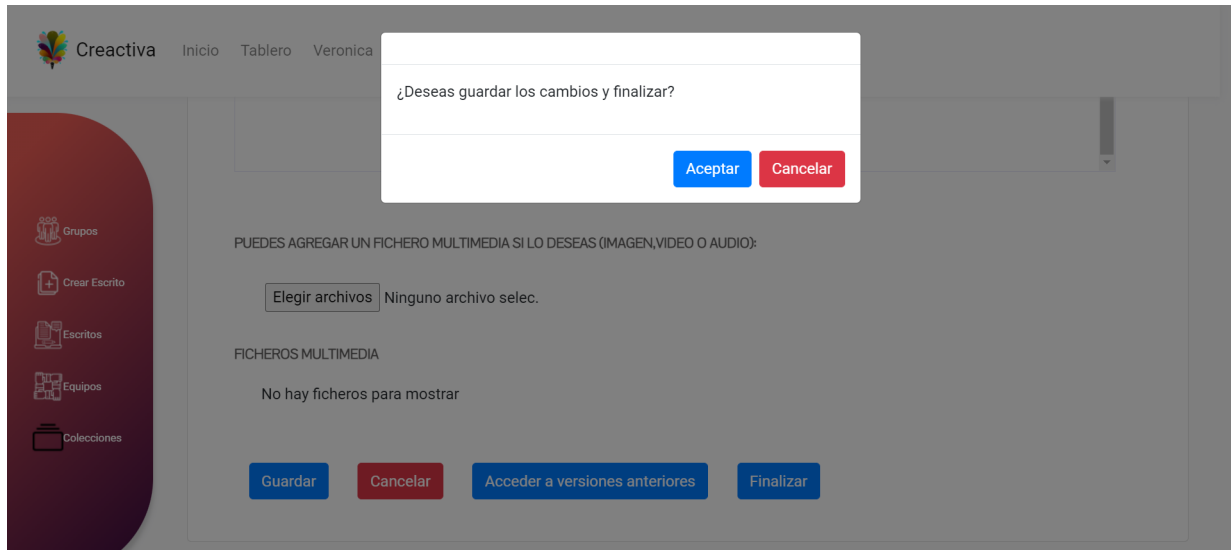
## 5.8. Finalizar escrito

Antes, los profesores debían esperar a la fecha en que terminase un desafío para saber que los estudiantes no iban a modificar más sus respectivos escritos. Ahora, en CREATIVA 2.0, cuando un alumno termine un escrito puede entregarlo de manera definitiva antes de la fecha de entrega. Con esta opción, los profesores pueden evaluar los escritos sin tener que esperar a que llegue la fecha de entrega. Para finalizar un escrito simplemente hay que pulsar el botón ‘Finalizar’ que se muestra en la figura 5.20.



**Figura 5.20:** *Botón para finalizar un escrito*

Después de haber pulsado el botón ‘Finalizar’, aparecerá el modal de la figura 5.21. Al pulsar en ‘Aceptar’, se finalizará el escrito.



**Figura 5.21:** *Confirmación de la finalización de un escrito*

Para saber si un escrito de un estudiante ha finalizado basta con ir a la lista de escritos y fijarnos en la columna 'Acciones'. Si la acción que se indica es 'Editar', el escrito no se ha finalizado todavía y se puede seguir editando. En cambio, si la acción a realizar es 'Ver', eso significa que el estudiante ha finalizado el escrito, o que la fecha límite de entrega del escrito ha llegado a su fin, por lo que solo se puede ver la realización del mismo, y la corrección del profesor si está disponible. En la figura 5.22 se muestra un ejemplo con escritos, donde se pueden ver ambas acciones.

-  Grupos
-  Crear Escrito
-  Escritos
-  Equipos
-  Colecciones

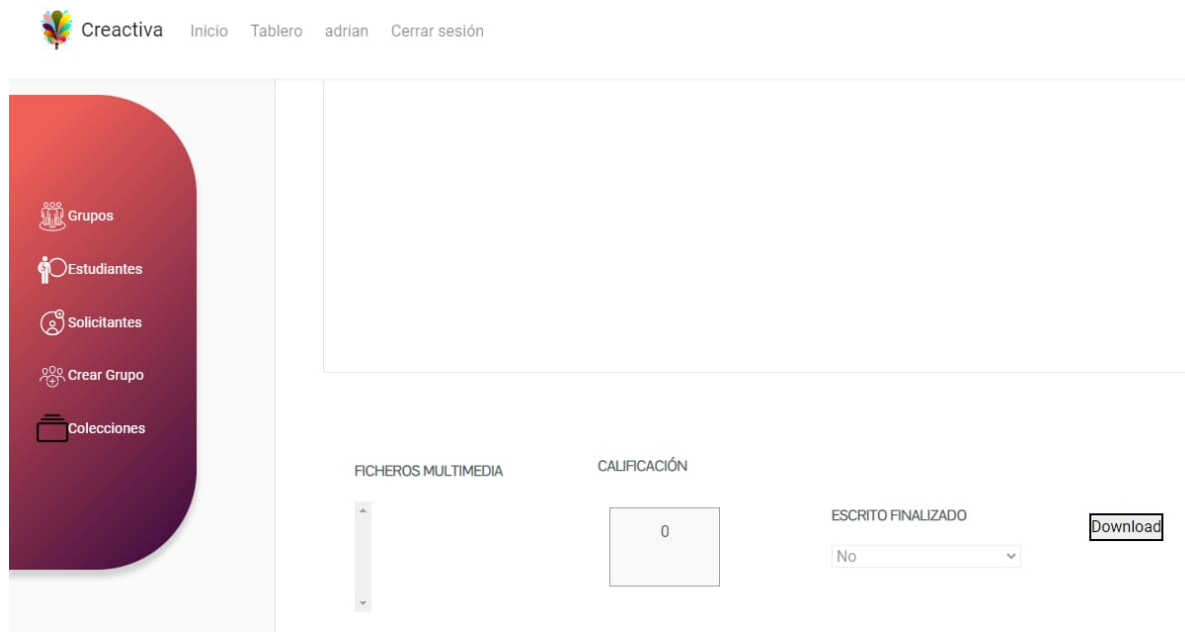
LISTA DE ESCRITOS

Escrito	Grupo	Desafío	Desafío Finalizado	Estudiante	Fecha	Hora	Acciones
1	Filosofía	Ponte en la piel de John Galt y redacta tu propio diario	No	Veronica Calzada	26/05/2022	4:34 PM	 <a href="#">Editar</a>
Prueba 4.4	Historia	Relatar un Episodio Nacional	Si	Veronica Calzada	14/05/2022	4:04 PM	 <a href="#">Ver</a>
Escrito Mozart	Arte	Requiem - Mozart	Si	Veronica Calzada	13/09/2022	5:50 PM	 <a href="#">Ver</a>

**Figura 5.22:** *Listado de escritos tras finalizar un escrito*

## 5.9. PDF

Para poder transcribir o mostrar un escrito de forma fija una vez esté finalizado existe la posibilidad de transformarlo a PDF a través del botón mostrado en el la figura 5.23. Al realizar esta acción se muestra un desplegable que te enseña las opciones para guardar o imprimir dicho escrito como PDF.



**Figura 5.23:** *Vista botón PDF*

Aquí se muestra como el archivo se almacena directamente en nuestro disco después de pulsar el botón 5.24. Siendo el sistema de descargas propio del explorador.

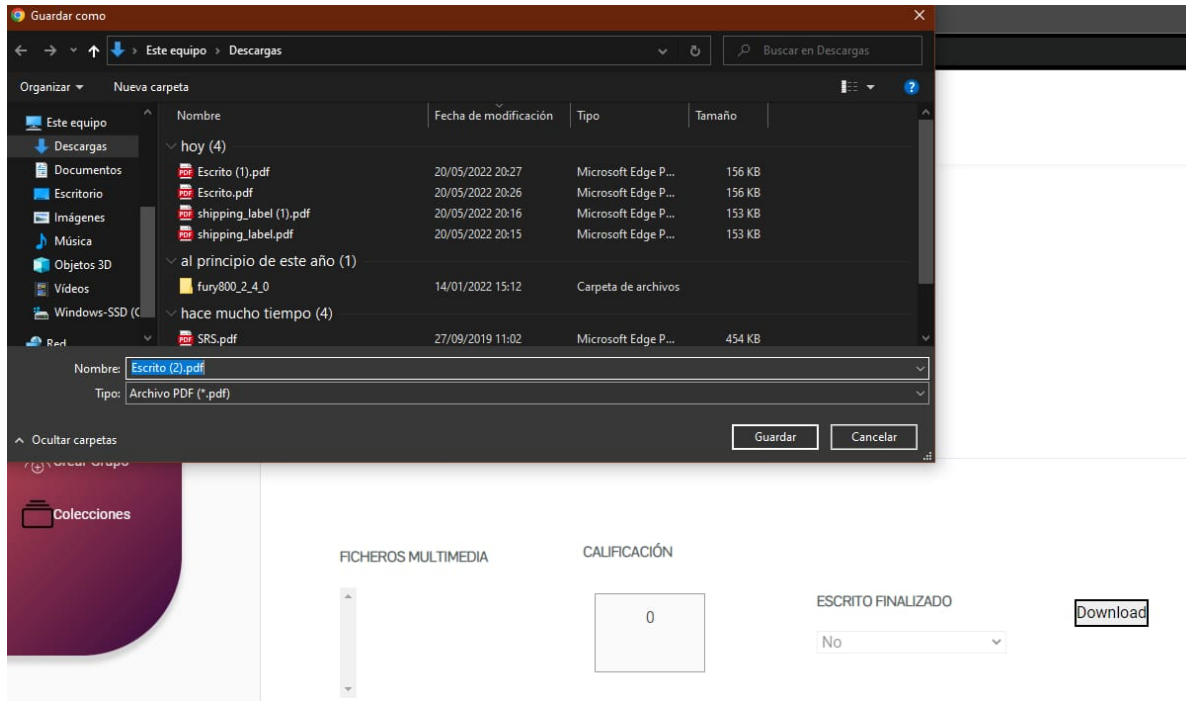


Figura 5.24: Vista botón PDF

## 5.10. Script para la actualización de textos colaborativos

Para que, en el modo colaborativo de un texto, distintos usuarios puedan editar un mismo texto y vean estas ediciones a tiempo real sin afectar en tiempos de recarga hemos creado un script que refresca el campo necesario así como revisa lo que se escribe para una correcta actualización. Este script actualiza de manera simultánea los escritos de los usuarios que lo estén realizando, esto debido al no estar desplegada la aplicación solo se puede comprobar la funcionalidad de nuevo montado un sistema de máquinas virtuales que por desconocimiento sobre este tema no hemos podido volver a montar este sistema.

# Capítulo 6

## Arquitectura de las extensiones realizadas

En esta sección se explica la arquitectura de la aplicación web, del patrón de diseño empleado, las bases de datos y las funcionalidades adicionales realizadas.

### 6.1. Patrón de diseño

Para la arquitectura del proyecto utilizamos principalmente el patrón de diseño Modelo-Vista-Controlador. Este patrón de diseño permite que el proyecto sea escalable, modular, y de fácil mantenimiento, permitiendo la reutilización de los componentes.

**Capa del modelo.** La capa donde reside la lógica de negocio de la aplicación, donde se relacionan los datos que la aplicación va a utilizar, como la inserción de datos, la creación de nuevos usuarios, consultas... etc. Cuando se realiza una acción, el modelo notifica del cambio de estado para que la vista se actualice

**Capa de la vista.** La capa visual, donde el usuario interactúa con la aplicación y sus funcionalidades mediante las interfaces de usuario, desde donde se envían las peticiones al controlador para su procesamiento posterior.

**Capa del controlador.** La capa donde reside la lógica de la aplicación, donde se reciben y manejan las solicitudes que vienen de las vistas y ejecuta las acciones del modelo,

según el procesamiento de la solicitud.

Respecto a la organización del proyecto, se ha distribuido el código en dos carpetas principales *server* y *client*, donde la carpeta del servidor contiene la capa del controlador y del modelo, mientras que la carpeta cliente contiene toda la implementación de la capa de la vista. Implementando 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 se diferencian los perfiles de estudiante y profesor, siendo distintos entre si, conteniendo distintos servicios y funcionalidades propias de su rol. Así mismo el servicio de usuario contiene las funcionalidades comunes entre el estudiante y el profesor tomando como ejemplo iniciar sesión y registro.

## 6.2. Estructura 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.

### 6.2.1. Back-end

El back-end se refiere al lado del servidor en la aplicación, refiriéndose así a la parte en la cual se desarrolla todos los procesos de negocio y con las operaciones de la base de datos de la aplicación con las que se ejecutan las distintas funcionalidades desarrolladas. La organización por directorios del lado del servidor es la siguiente:

- **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.
- **Scripts:** El directorio scripts contiene los scripts creados para las funcionalidades añadidas de la aplicación.

### 6.2.2. Front-end

El front-end se refiere al lado del cliente en la aplicación, refiriéndose así a la parte visual de la aplicación con la que interactúan los usuarios para acceder a las distintas funcionalidades desarrolladas. La organización por directorios del lado del cliente es la siguiente:

- **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 menú lateral y APIs integradas en la aplicación.

- **Links:** El directorio links contiene las redirecciones del menú lateral 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.
- **Scripts:** El directorio scripts contiene los scripts creados para las funcionalidades añadidas de la aplicación.

### 6.3. Base de datos

Para el diseño de la base de datos se ha elegido un modelo de base de datos relacional, utilizando MySQL como gestor de base de datos y XAMPP como interfaz de usuario para su conexión con la aplicación. Este entorno ha permitido una mejor gestión de los datos de la aplicación durante el proceso de desarrollo y testing. 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. Aun con las extensiones realizadas se mantiene el mismo razonamiento, siendo

un volumen de datos y tablas manejable para los desarrolladores. 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 [7.2](#), se explica más detalladamente el proceso de desarrollo de la base de datos aplicados en el proyecto.

# Capítulo 7

## Diseño de Creativa 2.0

En este capítulo se explican las modificaciones realizadas sobre la base de datos de Creativa 2.0, así como los cambios sobre el modelo entidad-relación que ello conlleva.

### 7.1. Entidades y atributos nuevos o modificados

La creación de este proyecto supuso la aparición de varias novedades en el modelo entidad-relación. Estas novedades se introdujeron mediante la creación de nuevas entidades, pero también mediante la modificación de algunas entidades que ya existían anteriormente. Todas las modificaciones que se han realizado sobre el modelo se explican a continuación:

**Versiónescrito:** anteriormente los escritos se guardaban en única versión, y por obviedad no se usaba el atributo versión. Esto significa que cada vez que se actualizaba un escrito, se ‘borraban’ los cambios del guardado anterior. Con la inserción de la nueva entidad Versiónescrito esto ya no ocurre, ya que se guardan todos los cambios que antes se perdían. Además, no sólo se puede acceder a la última versión modificada, sino que es posible consultar o modificar un escrito cuya versión se ha creado en un pasado lejano, en caso de ser necesario conocer cierta información de un cambio de un escrito realizado hace tiempo. Sus atributos más relevantes son el identificador del escrito y de la versión, ya que ambos valores en conjunto deben ser diferentes para cada versión de un escrito. Esto es porque los escritos se repetirán para cada versión, pero

cada versión debe ser diferente para cada escrito. El resto de datos, como el nombre del escrito o el texto que contiene el escrito, son iguales que los de la entidad escrito, ya que no se necesita almacenar ninguna información adicional.

**Grupoestudiante - atributo activo:** la entidad Grupoestudiante sufrió un cambio en su estructura, al introducirse en ella el atributo activo. El uso de este atributo en la entidad sirve para procesar las peticiones de unión a un grupo de estudiantes a profesores.

**Mensajería - atributo idCreador:** sobre esta entidad se resolvió un conflicto que provocaba varios errores en la aplicación. La modificación consistió en modificar el valor del atributo idCreador, que recibía dos identificadores distintos cuando lógicamente debía recibir siempre el mismo. Por tanto, el valor de idCreador de la entidad Mensajería en Creactiva 2.0 se corresponde con el identificador del equipo cuyo líder creó el mensaje.

**Mensajería - atributo activo:** el cambio realizado sobre esta entidad simplemente consiste en hacer uso del atributo activo, que estaba en desuso. Se utiliza igual que en muchas otras entidades, cuando se elimine un mensaje de la vista, en la lógica de la aplicación seguirá estando disponible, con el atributo activo a 0, simbolizando el borrado de dicho mensaje de la entidad.

**Colección:** esta entidad representa una forma de clasificar ciertos desafíos que forman parte de un grupo. Al ser un sistema de clasificación, sus atributos son mayormente identificadores que relacionan esta entidad con otras. Las relaciones por tanto son el identificador del profesor creador de la colección, y el identificador del grupo que contiene los desafíos que forman parte de la colección. Dicho esto, una colección solo puede tener asignado un profesor, pero un profesor puede tener muchas colecciones. En cuanto a la entidad grupo, una colección solo puede estar vinculada a un grupo, pero un grupo puede pertenecer a muchas colecciones de un mismo profesor. Una colección también dispone de un nombre, que representará en la interfaz a la colección. Por

último, se hace uso del atributo activo para eliminar las colecciones simplemente en la parte del cliente de la aplicación, mientras que en la lógica seguirá existiendo el atributo.

**Colección desafío:** esta entidad actúa de intermediario entre colecciones y desafíos. Por ese mismo motivo cabe destacar sus relaciones, donde una colección puede tener múltiples desafíos, pero un desafío solo puede pertenecer a una colección. Los atributos que conforman la entidad son, por tanto, el identificador de la propia entidad, el identificador de una colección, y el identificador de un desafío.

**Escrito - atributo finalizado:** la entidad escrito no sufrió un cambio en su estructura en Creactiva 2.0, pero sí un cambio respecto al uso de su atributo finalizado. Esto es básicamente porque antes este atributo existía pero no se utilizaba. Ahora su uso consiste en finalizar un escrito, como bien indica su nombre. Esto implica que no se pueda modificar de nuevo un escrito una vez haya sido marcado como finalizado. Sólo se puede finalizar un escrito si no ha llegado ya la fecha límite para seguir editando dicho escrito.

## 7.2. Base de datos SQL

Se ha utilizado la misma base de datos que en el proyecto anterior, que se gestiona a través del lenguaje SQL. Con SQL podemos almacenar las tablas de nuestro modelo con sus atributos, y crear relaciones entre las tablas sobre los atributos que identifican a cada una de ellas. Este lenguaje dispone de múltiples herramientas que nos han facilitado su uso:

**Claves :** consiste en la posibilidad de poder definir atributos según la función que desempeñen. Nosotros hemos hecho uso de la clave primaria, que se le asigna al atributo que identifica, y que además con su valor distingue, a cada entrada de la tabla. Esta clave, además, en muchas ocasiones se acompaña de una opción que ofrece SQL que consiste en incrementar automáticamente el valor del atributo en uno. Esto por tanto

es muy útil para los números enteros que identifican una entidad. También se hace uso en muchas tablas de la clave externa, que se basa en hacer referencia a un atributo de otra tabla, y de esta forma crear una relación.

**Restricciones** : su funcionamiento consiste en realizar una acción cuando se altera algún atributo de otra tabla. Es decir, si en otra tabla se modifica o elimina un atributo, y en la tabla actual existe una restricción sobre un atributo que es clave externa del atributo de la otra tabla, se realizará la acción que se indique en la configuración de la restricción. Es bastante común que la acción que se indique sea ‘cascade’, que lo que hará será replicar la acción sobre el atributo indicado en la tabla donde se origina el cambio.

**Tipos para los atributos** : existen múltiples tipos posibles para asignar a un atributo, lo que permite una fácil gestión, e incluso un mejor entendimiento del uso que se hace de cada atributo.

**Importación y exportación** : el sistema de importación y exportación de la base de datos en formato sql es intuitivo y muy sencillo de usar, lo que nos ha supuesto una cómoda vía para comparar cambios sobre una versión más antigua o más nueva de la base de datos.

**Consultas** : la opción de phpMyadmin de poder realizar consultas sobre las tablas existentes en la base de datos nos ha supuesto una herramienta útil para realizar pruebas. Especialmente hemos hecho uso de ella para revisar consultas extensas o enrevesadas.

A través del uso de las herramientas que acabamos de nombrar, hemos implementado las mejoras sobre la base de datos, que se comentan a continuación.

### 7.2.1. Tabla versionescritos

Esta entidad hace referencia a todas las versiones que se crean de los escritos. Las versiones no se pueden eliminar, ya que existen precisamente para poder acceder a ellas en

cualquier momento. Cada versión de un escrito tiene atributos que también se usan en la entidad ‘escrito’, ya que al fin y al cabo es una copia del mismo. Estos atributos son los siguientes:

- **idEscrito**: número entero que hace referencia al identificador de un escrito. Junto con el atributo ‘idVersion’, forma la clave primaria de la entidad. Además, es también una clave externa que hace referencia al atributo ‘id’ de la tabla ‘escrito’.
- **idVersion**: número entero que representa el identificador de una versión. Forma, junto con el atributo ‘idEscrito’, la clave primaria de esta entidad. También representa el número de versión de un escrito que se está editando.
- **idDesafío**: número entero que representa el identificador de un desafío. Es una clave externa que hace referencia al atributo ‘id’ de la tabla ‘desafío’.
- **idEscritor**: número entero que representa el identificador del estudiante creador del escrito. Es una clave externa que hace referencia al atributo ‘id’ de la tabla ‘usuario’.
- **nombre**: cadena de texto que representa el título de la versión de un escrito. Se puede repetir el nombre en distintas versiones. Sigue el formato de codificación de caracteres UTF-8.
- **texto**: campo de texto que representa el cuerpo del escrito. Sigue el formato de codificación de caracteres UTF-8.
- **colaborativo**: número entero cuyo valor solo puede ser 1 o 2. Si vale 1, indica que la versión del escrito se realiza de manera individual, en cambio, si su valor es 2, esto quiere decir que la versión del escrito se realiza de manera colaborativa.
- **fecha**: representa la fecha de realización de la versión del escrito. Su valor siempre es la fecha real en que se guarda la versión.

### **7.2.2. Atributo activo (tabla grupoestudiante)**

Este atributo, perteneciente a la entidad ‘grupoestudiante’, no existía antes ya que no era necesario. Pero, ahora, al implementar el sistema de solicitudes de unión a un grupo de estudiantes a profesores, sí se requiere del uso de este atributo. Su utilidad se pone de manifiesto en el momento en que el estudiante le envía la petición al profesor (ver figura 5.6). En ese momento, se añade a la tabla ‘grupoestudiante’ una nueva entrada con los valores de ‘idGrupo’ e ‘idEstudiante’ que correspondan, y el valor de ‘activo’ se introduce como 0, ya que el estudiante todavía no forma parte del grupo. En cuanto el profesor acepte al estudiante en el grupo (ver figura 5.8), el valor de ‘activo’ pasará a ser 1, y el proceso de petición habrá finalizado.

### **7.2.3. Atributo idCreador (tabla mensajería)**

Este atributo ya existía previamente en la tabla ‘mensajería’, lo que ocurre es que no funcionaba correctamente. Por lo tanto no hemos ni insertado ni modificado el atributo como tal, simplemente hemos cambiado su funcionamiento. El problema es que se le asignaban dos valores distintos. Por un lado, su valor podía ser el identificador del usuario que creó el mensaje y, por otro lado, podía hacer referencia al identificador de un equipo, donde el líder de dicho equipo es quien creó el mensaje. Concretamente, cuando un estudiante quería unirse a un equipo, al enviar la solicitud el atributo ‘idCreador’ se correspondía con el identificador del equipo. Por el contrario, cuando era un estudiante quien invitaba a otro a unirse a un equipo, ahí se le asignaba a ‘idCreador’ el identificador del usuario líder del equipo. Además, esta incongruencia provocaba otros fallos en cadena. Estos fallos eran el no envío de mensajes de error, como por ejemplo poder enviar varias peticiones de unión a un equipo y, como consecuencia, la posterior gestión incorrecta de los mensajes. Para finalizar, lo que hemos considerado mejor para solucionar este conflicto es asignarle siempre el identificador del equipo al atributo ‘idCreador’, ya que así se puede saber, de forma más directa, qué equipo está realizando la incorporación de estudiantes.

#### 7.2.4. Tabla coleccion

Esta entidad representa la gran novedad de Creativa 2.0. Consiste en un sistema de organización de desafíos para los grupos disponibles en la plataforma. Tiene el objetivo de proporcionar una cómoda distribución de los desafíos para los estudiantes y los profesores, además de ofrecer una rápida localización de los mismos a través de su clasificación. Los atributos que componen la entidad se explican a continuación:

- **id**: número entero que representa el identificador de una colección. Es la clave primaria de tabla y se genera automáticamente, incrementándose en uno su valor por cada nueva entrada que se crea en la tabla.
- **nombre**: campo de texto que contiene el nombre de la colección. Puede haber dos colecciones con el mismo nombre. Utiliza el formato de codificación de caracteres UTF-8.
- **activo**: número entero cuyo valor solo puede ser 0 o 1. Su valor será 0 cuando un profesor borre la colección. De esta forma hacemos que la colección desaparezca solo de la vista, ya que seguirá existiendo en la base de datos. En cualquier otro caso 'activo' valdrá 1.
- **idProfesor**: número entero que representa el identificador de un usuario con rol de profesor. Es una clave externa que hace referencia al atributo 'id' de la tabla 'usuario'.
- **idGrupo**: número entero que representa el identificador de un grupo. Es una clave externa que hace referencia al atributo 'id' de la tabla 'grupo'.

#### 7.2.5. Tabla colecciondesafio

Esta entidad simboliza la unión entre colecciones y desafíos. Su funcionamiento consiste en almacenar todos los desafíos de cada una de las colecciones. Los atributos que forman la entidad son los siguientes:

- **id**: número entero cuyo valor se autoincrementa en uno por cada entrada nueva que se genere en la tabla. Es la clave primaria.
- **idColeccion**: número entero que representa el identificador de una colección. Es una clave externa que hace referencia al atributo 'id' de la tabla 'coleccion'.
- **idDesafio**: número entero que representa el identificador de un desafío. Es una clave externa que hace referencia al atributo 'id' de la tabla 'desafio'.

### 7.2.6. Atributo finalizado (tabla escrito)

Este atributo sirve para 'entregar' un escrito. Si el atributo vale 0, el escrito se podrá seguir editando, siempre y cuando la fecha límite no se haya cumplido. En cambio, si el atributo vale 1 significa que el estudiante ha finalizado el escrito y por lo tanto ya no se puede seguir editando. Una vez finalizado un escrito, el profesor podrá evaluar dicho escrito antes de que llegue la fecha límite de entrega. Precisamente esta es la utilidad principal que tiene este atributo.

# Capítulo 8

## Contribuciones individuales

En este apartado los integrantes del grupo de Creactiva 2.0 explicarán sus contribuciones al proyecto. Gran parte del proyecto fue compuesto equitativamente entre todos los miembros.

### 8.1. Rafael Alcaide Torralvo

Mi contribución en el proyecto son mejoras que han tomado forma a través de varias etapas, que detallo a continuación.

#### 8.1.1. Aprendizaje

Durante la etapa inicial, al ser el proyecto que planificábamos un proyecto ya basado en otro realizado el año anterior, empezó una etapa de estudio en la cual estuve consultando diverso material para llegar al entendimiento completo del proyecto ya realizado. Realicé un pequeño proyecto para lograr un mayor entendimiento del funcionamiento de la aplicación y para entender sobre los patrones de código y sobre todo entender el comportamiento de ya que era una nueva tecnología para mí. En este proyecto realicé una pequeña página web, en la cual seguiría el mismo patrón de arquitectura pero más básico. De esto resultó un proyecto donde la parte del cliente estaba implementada con ReactJs y la parte del servidor con Node.js; a pesar de ser muy simple ayudó al progreso y entendimiento de estas tecnologías tanto a mí como a mis compañeros.

### 8.1.2. Investigación

La etapa de investigación surgió a base de la búsqueda de tecnologías web y herramientas para la realización de las primeras partes del proyecto.

En esta etapa colaboré con la búsqueda de funcionalidades parecidas a las que se nos pedían, para de este modo poder entenderlas y traducirlas a nuestro proyecto. Dejando una amplia lista a nuestro alcance de funcionalidades ya creadas o parecidas para facilitar el trabajo tanto mío como de mis compañeros. Proporcionando un pequeño proyecto con las funciones no muy comunes que teníamos que replicar dejando claro cómo se comportarían estas funciones en nuestro proyecto a modo de prueba para poder tener claro los resultados al ejecutarlas sin tener que implementarlas en nuestro proyecto y así dejar la base de datos original intacta para las funciones que realmente devuelvan el resultado esperado.

### 8.1.3. Conexión funcional con el servidor

Al principio teníamos problemas de compatibilidad con el proyecto, ya que faltaban bibliotecas y dependencias, tanto mi grupo como yo hemos tenido que arreglar estos problemas y solucionar los errores. Esto se solventó rápido, al colaborar con mis conocimientos en Visual Code importamos las bibliotecas necesarias y ajustamos las dependencias. Siendo el principal problema que algunos paquetes estaban obsoletos y algunas herramientas usadas tenían que ser instaladas de manera individual dado que muchas de estas herramientas eran extensiones de las originales.

### 8.1.4. Implementación de las funciones

Una vez definido todos los aspectos que cubriremos cada integrante y conseguir adaptar el proyecto a nuestros entornos virtuales comencé con la implementación de las funcionalidades del proyecto.

**Finalizar escrito.** En este apartado implementé la funcionalidad de que un escrito pudiera finalizarse de manera manual por el alumno asemejándose a una entrega, en vez de

dejar que este se finalizara solo por la fecha de finalización. Para esto necesitábamos una vista que se adapte a los datos que guarda el alumno dado que ahora él puede finalizar el proyecto. Al realizar esta vista que se actualiza dinámicamente según esté o no finalizado, implementé el botón que permitía al alumno finalizar el escrito desde la vista del escrito. Para esto necesité crear el atributo en la base de datos para que el alumno lo pudiera modificar. Al crear este botón necesité implementar un nuevo modal para que aparezca en la parte superior de la pantalla para que el alumno verifique que realmente quiere finalizar el escrito dado que es irreversible. A nivel de vista implemente el cambio de vista a modo solo lectura por parte del estudiante para que este no pueda modificar más el escrito y el cambio de vista a modo modificar para que el profesor pueda calificar una vez entregado.

**Formato PDF del escrito** Para una mejor portabilidad de los escritos implementé la funcionalidad de convertir un escrito a formato PDF. Para esto necesité entender cómo funcionaba la biblioteca jsPDF la cual nos proporcionaría las funciones para convertir el escrito a PDF. Para llevar a cabo la conversión creamos un script que se activaría cada vez que se necesite crear el PDF, pero a raíz de que el proyecto está creado en ReactJS surgían varios problemas a la hora de implementar esta funcionalidad dado que los scripts no se cargaban correctamente. ReactJS usa Renders para cargar las vistas y no usa vistas creadas directamente, por lo tanto tuvimos que llamar al script externamente para poder corregir este fallo.

**Script para la actualización de textos colaborativos** Desarreñe un script hecho en javascript el cual nos permite actualizar simultáneamente los escritos si más de un usuario esta realizando el mismo escrito con un margen de 2-5 segundos dependiendo del número de peticiones. Al no haber logrado el despliegue para esta versión el script en este momento solo funciona de manera local, la única forma de implementarlo y comprobar su funcionalidad se baso en el montaje de un servidor local con varias maquinas virtuales, usando una como servidor y el resto de clientes, para comprobar el escrito en paralelo con diversos supuestos usuarios. Esto fue comprobado por mi compañero, al no estar aquí no he podido retocar

mas esa funcionalidad dado que no tengo experiencia con máquinas virtuales

### 8.1.5. Despliegue de la aplicación

En este apartado para poder desplegar la aplicación me hice una cuenta en Azure que soportara una base de datos de nuestro estilo, una vez hecho esto intenté subir la base de datos a la nube lo cual lo conseguí con éxito pero a raíz de que nuestro proyecto estaba dividido en cliente-servidor conectar ambas a la vez y sincronizarlas se me hizo cuesta arriba. Tras una investigación descubrí que esto prácticamente era imposible de realizar debido a la arquitectura de nuestro proyecto por lo cual intente realizar este despliegue en otros servicios como Digital Ocean y Amazon Web services, pero estos al no ser gratuitos y solo disponer de un periodo de prueba tan corto no pude terminar con las pruebas sobre estos servicios. Debido a esto hemos pospuesto el despliegue de la aplicación como un proyecto futuro.

### 8.1.6. Memoria

Creamos la memoria en la plataforma Overleaf y buscamos una plantilla para poder realizar el Trabajo de Fin de Grado. Una vez que lo obtuvimos, se lo enviamos a los tutores, Adrián y Cora, para futuras correcciones, y a nuestros compañeros. Los puntos principales desarrollados por mí:

- **Resumen**

He aportado con un breve resumen de la aplicación que vamos a presentar.

- **Capítulo 3: Preliminares**

He contribuido en este apartado creando todo el contenido de este punto así como su desarrollo

- **Capítulo 4: Trabajo relacionado**

He detallado los aspectos comunes y no comunes con aplicaciones que comparten un objetivo con la nuestra.

- **Capítulo 5: Funcionamiento de la nueva versión**

He aportado mi contribución a nivel funcional de la aplicación en este apartado

- **Bibliografía**

He detallado las fuentes de información que hemos usado en las referencias de algunos términos no comunes.

## 8.2. Verónica Calzada Álvarez

Mi contribución en el proyecto para las mejoras de Creactiva se han llevado a cabo en pareja junto a Jaime Madriñán. Estas mejoras han tomado forma a través de varias etapas:

### 8.2.1. Primera reunión

Para comenzar el proyecto Creactiva 2.0, se realiza una primera junta donde se reúnen todos los miembros del equipo junto a nuestros dos tutores. Previamente, se nos había dado la memoria, el código y la documentación del Trabajo de Fin de Grado anterior, del cual partíamos. Con ello, contábamos con los conocimientos suficientes para entender lo que se nos iba a exigir en esta reunión. Adrián Riesco como tutor encargado, principalmente, de la parte informática, determinó que los objetivos a cumplir (que se mencionarán más adelante) eran adecuados para el proyecto. Estos objetivos se obtienen gracias a Cora Requena, tutora que actúa como ‘cliente’ y a través de la cual obtenemos los nuevos requisitos de usuario. Los requisitos que quedan a implementar por mi parte a partir de esta primera reunión son:

- Crear bloques temáticos para plantear la nueva sección de desafíos: ej. un bloque de poesía, relatos cortos, otro de imagen y palabra, etc. Todos estos cambios enfocados a la base de datos.
- Mejorar la usabilidad de la aplicación sobre la que trabajamos.
- Hacerlo más amigable, o sea, una actualización que mejore la UX.

Además, se establecen reuniones cada dos semanas por Meet y también tutorías para realizar nuevas mejoras o resolver dudas.

### 8.2.2. Etapa de investigación

Si bien es cierto que contábamos con el código y la documentación necesaria para comenzar el proyecto, empezamos a encontrar fallos en la primera versión de la aplicación ya que nosotros teníamos asignados comprobar que la aplicación era funcional, por lo cual, nos ponemos en contacto con dos de sus integrantes para resolver dudas que no aparecen en las ‘guías’. Se añade al *server* la ruta que falta y se arregla el *node modules* de *server*. Una vez arreglados estos primeros fallos, pudimos entender toda la funcionalidad de Creactiva. Encontramos la aplicación poco amigable con el usuario y propusimos algunos cambios simples como cambiar ‘grupos’ por ‘asignaturas’, que no sea necesario que la activación de la cuenta del profesor dependa de un administrador, etc. Pero se pensó que era mejor dejar esos retoques para el final y seguir con la siguiente tarea asignada.

### 8.2.3. Base de datos

Lo primero que se nos pidió como pareja (así como posteriormente en varias ocasiones), fue la población de la base de datos, que se encontraba vacía. Más tarde, una adecuación de la información de la base de datos para que fuese lo más realista posible de cara a la demostración. Además, la base de datos ha sufrido múltiples modificaciones por nuestra parte como se puede apreciar en el capítulo 7, ya que solo nuestras funcionalidades han necesitado modificar la base de datos.

### 8.2.4. Código

Aunque se han realizado una gran variedad de cambios, los más significativos serían los siguientes:

**Versiones de escritos** Se crea una nueva funcionalidad que gestiona la posibilidad de

poder acceder a versiones anteriores de escritos con el fin de poder modificarlos si fuese necesario. Las versiones se implementan en individual y colaborativo.

**Alumno pide ser añadido a un grupo** Nueva funcionalidad para que el alumno tenga la posibilidad de pedir a los profesores que sea añadido a un determinado grupo. Antes, el alumno solo tenía la posibilidad de estar en un grupo por decisión del profesor, ahora puede pedir entrar.

**Colecciones** Se pensó en esta funcionalidad primero como una manera de poder combinar escritos de distintas categorías o desafíos, individuales o en grupo. Más tarde, pasó a ser colecciones; enfocado a un “coleccionar” desafíos con la intención de que estos desafíos puedan ser de una misma serie.

**Colecciones del alumno** Se trata de una vista de colecciones para los estudiantes, pero ellos no pueden editar nada.

**Colecciones del profesor** La funcionalidad de colecciones está pensada principalmente para el profesor. Esta propuesta organiza los desafíos de los grupos y mejora el control de la totalidad de los desafíos que hay, pudiendo ver también los escritos de estos, finalizados o no. Cada colección de desafíos por grupo es una colección.

**Arreglo de fallos de la versión anterior** A continuación, se explicarán brevemente los arreglos hechos a Creactiva 1.0:

**Grupos** Se arregla el botón de grupos que no estaba implementado, que consistía en eliminar un desafío de un grupo. También se cambia la tabla del listado de equipos en un grupo (menú lateral ‘Grupos’), que listaba los integrantes en distintas columnas.

**Mensajería** El botón eliminar mensaje de mensajería no estaba implementado en Creactiva. Además, se recibían solicitudes repetidas de unión a equipos.

**Equipos** En la vista de ‘Equipos’ la columna de acciones estaba mal, pues sacaba el nombre de otro integrante del equipo, y en integrantes solo aparecía el líder. También aparecía la opción de eliminar equipo aunque no fueses el creador del equipo, por lo cual también lo corregimos asumiendo que era un error.

**Apartado Inicio** Tras realizar la demostración de Creactiva 2.0, muchas de las sugerencias fueron una sección de ayuda a modo de tutorial para entender como navegar por la aplicación web y cuál era su finalidad. Por lo que implementamos una nueva categoría llamada ‘Inicio’ en la barra superior de Creactiva 2.0.

### 8.2.5. Memoria

Creamos la memoria en la plataforma Overleaf y buscamos una plantilla para poder realizar el Trabajo de Fin de Grado. Una vez que lo obtuvimos, se lo enviamos a los tutores, Adrián y Cora, también les explicamos el funcionamiento de esta. Los puntos principales desarrollados por mí, en colaboración con Jaime fueron:

- **Capítulo 1: Introducción**

Donde se tratan los objetivos y motivación de la realización de Creactiva 2.0.

- **Chapter 2: Introduction**

Donde se describe el capítulo 1 en inglés.

- **Capítulo 5: Funcionamiento de la nueva versión**

Se realiza el capítulo 5 del apartado 5.2 al 5.7 (inclusive). Se explican las nuevas implementaciones llevadas a cabo en la aplicación.

- **Capítulo 7: Diseño de Creactiva 2.0**

Se explica detalladamente el nuevo diseño de la aplicación.

- **Capítulo 9: Demo de Creactiva 2.0**

Se analizan los datos extraídos de la demostración de Creactiva 2.0.

También realizamos la comunicación con los tutores para que nos corrigiesen los capítulos pertinentes una vez finalizados, compenetrándonos así con nuestros compañeros.

### 8.2.6. Demo de Creactiva 2.0

Para la demo de Creactiva 2.0, nos encargamos de realizar un formulario de Google donde recogimos información acerca de todos los puntos que nos interesaban sobre la aplicación. También fuimos nosotros quienes comprobamos que la aplicación estuviese completamente funcional para ese día, dado que por problemas en el repositorio solo contábamos con las implementaciones expuestas del apartado 5.2 al 5.7. Tras la demostración, de nuevo nosotros como pareja, y dado que realizamos el capítulo 9 analizamos la información recabada y en conjunto con los tutores, decidimos qué puntos se iban a implementar antes de dar por concluido el proyecto. Los resultados obtenidos tras la demo de Creactiva 2.0 se pueden observar en el capítulo 9.

### 8.2.7. Despliegue de la aplicación

Para comprobar que nuestra aplicación es útil y apta, hicimos una búsqueda de *hosting* gratuito. Investigamos el *hosting* y ojeamos cuál elegir entre las opciones gratuitas, siendo en primera instancia 000webhost. Debido a que trabajamos con Node y, sobretodo por React, esta primera opción quedó descartada.

Decidimos así, optar por cambiarnos a la opción de GitHub Student, lo cual nos daba acceso a una cuenta Premium y cierto presupuesto en algunos sitio de pago. Una vez dispusimos de este presupuesto, comenzamos el despliegue, esta vez con Microsoft Azure y subimos a su servidor la base de datos, pero no fue posible continuar con el despliegue por este método. Posteriormente probamos Digital Ocean y Amazon Web Services, pero no soportaban la aplicación con el créditos del que disponíamos. Por lo cual, se estudia el despliegue de la aplicación para un futuro.

## 8.3. Jaime Madriñán Fernández

Mi trabajo en este proyecto lo he realizado conjunto con Verónica Calzada. El desarrollo del trabajo se distribuye en varias fases, que comentaré a lo largo de esta sección.

### 8.3.1. Etapa inicial

Empezamos el proyecto teniendo una reunión los cuatro integrantes del equipo y nuestros dos tutores, Adrián y Cora. A esta reunión ya veníamos con todo el material que se había realizado en Creactiva 1.0, por lo que ya teníamos conocimientos previos de lo que se iba a hablar en la reunión. Adrián, como especialista de los conocimientos informáticos, nos comentó superficialmente las tareas que debíamos completar a lo largo del TFG. Cora por su parte, que se centra más en las tareas que se deben desarrollar, contribuye también a estas tareas y especifica si es necesario algún detalle sobre cómo deben ser los estilos de alguna funcionalidad concreta de la página. Inicialmente, los objetivos que se proponen en Creactiva 2.0 son los siguientes:

- Crear una opción que permita combinar o agrupar desafíos. Se busca poder clasificar mejor los desafíos ya existentes.
- Mejorar la usabilidad de la aplicación.
- Mejorar la interfaz, consiguiendo un cómodo uso de la aplicación por parte del usuario.

También se llega a la conclusión de que es conveniente realizar reuniones todo el grupo cada dos semanas y como añadido se propone la opción de poder convocar reuniones extraordinarias en caso de ser necesario.

### 8.3.2. Etapa de investigación

Cuando decidimos empezar a hacer cambios en el proyecto, nos encontramos con algunos problemas que no esperábamos. Estos problemas consistían en arrancar la aplicación,

y nos provocaron que tuviésemos que investigar donde estaba el error, que a su vez retrasó un poco el inicio de los objetivos que teníamos previstos. Tras consultar a través de correo electrónico nuestras preocupaciones con los miembros del proyecto del año pasado, llegamos a la conclusión que la ruta que se usa para iniciar el arranque del servidor no está incluido con el nombre que debería, así que lo cambiamos. Después de esto, probamos cómo funciona la página web viendo las operaciones disponibles, y testeamos en general la aplicación. El siguiente paso nos llevó a sugerir algunas modificaciones, como que el profesor no necesitase la autorización del administrador para entrar en la aplicación después de registrarse, y que además pudiese añadir a los alumnos (que también necesitan la autorización del administrador), para así facilitar el proceso de registro y agilizar más el proceso de entrada en la aplicación. Otra idea que pensamos fue cambiar el nombre de ‘Grupos’ a ‘Asignaturas’, ya que ‘Grupos’ y ‘Equipos’ pensamos que son conceptos que se pueden confundir. Finalmente se pospusieron temporalmente estos cambios, en parte porque ambos casos estaban realizados de esta forma por un motivo razonable. En el caso del administrador porque supone tener más seguridad en la aplicación, ya que si cualquier usuario puede entrar sin autorización podrían alternarse los roles de forma voluntaria y con malas intenciones, y en caso de los ‘Grupos’ porque este nombre sugiere una denominación más adecuada para la escritura creativa.

### **8.3.3. Cambios de la base de datos**

Una vez superados los primeros baches, nos pusimos en funcionamiento para realizar la inserción de datos en la base de datos, que era una de las primeras tareas asignadas. Posteriormente, se fueron añadiendo los atributos y las tablas necesarias, al mismo tiempo que se modificaba la funcionalidad de la aplicación. Las modificaciones que se introdujeron en la base de datos se comentan en profundidad en el capítulo 7.

### 8.3.4. Código

Los cambios más trascendentales que se han realizado a lo largo del proyecto se comentan a continuación:

- **Versiones de escritos**

Consiste en añadir un histórico de modificaciones que se realizan sobre un escrito. Estas se denominan versiones, y se guardan de manera que pueden volver a ser modificadas en cualquier momento. Se puede acceder al listado de versiones desde la vista de editar un escrito, pulsando el botón ‘Acceder a versiones anteriores’.

- **Petición de unión a un grupo por parte del alumno**

Con esta nueva funcionalidad, el alumno ahora puede agilizar el proceso para entrar en un grupo, ya que en vez de esperar a que el profesor encargado le invite a un grupo, puede mandar directamente dicha invitación al profesor. Se puede encontrar este sistema de solicitud en la pestaña ‘Grupos’.

- **Colecciones**

Esta nuevo menú lateral surgió con el objetivo de poder categorizar y ordenar los desafíos más claramente. Por tanto una colección representa un sistema de clasificación de desafíos que se encuentran dentro de un mismo grupo. Las colecciones además se diferencian para los alumnos y los profesores de la forma que se indica a continuación.

- **Colecciones del alumno**

Las colecciones que tiene el alumno son las mismas del profesor, pero sin ninguna de las operaciones disponibles. Por ejemplo, dispone del listado de colecciones, pero no puede crear o eliminar una colección.

- **Colecciones del profesor**

El profesor es el usuario a quien va enfocado las colecciones. Contiene el listado y

las operaciones de añadir y crear colecciones, como comenté anteriormente. También dispone del listado de desafíos que forman una colección, pudiendo añadir o prescindir de desafíos que formen parte del grupo sobre el que se creó la colección. Por último se pueden ver los escritos que forman parte de cada desafío, lo que permite tener más información disponible.

## ■ Correcciones sobre Creactiva 1.0

### ● Eliminar desafío

El usuario con rol de profesor anteriormente no podía eliminar desafíos, ya que saltaba un mensaje de error que no permitía realizar esta acción. Esta opción de borrado se encontraba en el menú lateral ‘Grupos’.

### ● Mensajería

La mensajería tenía varios problemas que al principio nos confundieron bastante, pero a los que finalmente acabamos terminamos hallando una relación y gracias a esto conseguimos solventar correctamente. El problema comenzaba cuando se enviaban peticiones entre estudiantes, ya que no estaba bien definido un atributo en la base de datos y esto propiciaba que no se mandasen algunos mensajes de error, o que se enviaran algunos mensajes de error incorrectos. Esto viene explicado en detalle en el apartado [7.2.3](#).

### ● Equipos

Se arregló el listado de las tablas que hacían referencia a un equipo. Estas tablas se encontraban en los menús laterales ‘Equipos’ y ‘Grupos’ (desplegable ‘Equipos’). Esta errata se provocaba por el incorrecto listado de los integrantes de un equipo, ya que se descuadraban las columnas.

## ■ Apartado Inicio

La última implementación realizada fue la página de inicio, cuya idea surgió como sugerencia de los alumnos que realizaron la demo de Creactiva 2.0. Esto se debe a que

encontraron dificultades para entender como funcionaba la aplicación, y pensaron que una página explicativa ayudaría mucho a orientar a los nuevos usuarios. Esta página se encuentra en el encabezado de la aplicación.

### 8.3.5. Memoria

Los capítulos que he desarrollado conjunto con Verónica han sido los siguientes.

- **Capítulo 1: Introducción**

Se realiza una explicación general de Creactiva 2.0

- **Chapter 2: Introduction**

Traducción del capítulo anterior al inglés.

- **Capítulo 5: Funcionamiento de la nueva versión**

Se explican las nuevas implementaciones, incluyendo gráficos para facilitar la lectura (desde el apartado 5.2 hasta el 5.7).

- **Capítulo 7: Diseño de Creactiva 2.0**

Se describen los cambios implementados en cuanto al modelo entidad-relación y la base de datos.

- **Capítulo 9: Demo de Creactiva 2.0**

Se hace un análisis de la información recogida tras la realización de la demostración de Creactiva 2.0.

### 8.3.6. Demo de Creactiva 2.0

La demostración en vivo del funcionamiento de la aplicación se realizó a través de nuestra tutora Cora, con la que concretamos un día para poder hacerla en el aula donde ella imparte clases, y así acceder al público que iba a actuar de cliente, los alumnos de Cora. Unos días antes de la presentación tuvimos que hacer una pausa momentánea de algunos cambios que estábamos realizando sobre la aplicación. Esto nos supuso presentar las funcionalidades

realizadas desde el apartado 5.2 hasta el 5.7. Después de la demo, los alumnos de Cora respondieron unas preguntas a través de los formularios de Google, cuyas respuestas y más detalles se pueden ver en el apartado 9.

### 8.3.7. Despliegue de la aplicación

Para realizar el despliegue, comenzamos buscando algún *hosting* gratuito, y nos centramos en 000webhost. Pero al ser gratuito la realización del despliegue supone más complicaciones, por lo que no fuimos capaces de sacarlo adelante. Continuamos buscando opciones, y nos topamos con Github Student. Esta opción era bastante tentativa, ya que a través de la universidad nos permitía disponer de una cuenta de pago de forma gratuita. A través de Microsoft Azure empezamos a investigar distintas formas de realizar el despliegue, pero sin éxito. Ante la falta de conocimientos y de experiencia sobre el tema, conseguimos arrancar la parte del servidor, pero no la parte del cliente. En general, el principal problema que encontré por mi parte, es que para realizar el despliegue de forma más simple los servicios ofrecen opciones de pago donde, en mi opinión, el presupuesto es desorbitado. Finalmente, se decide proponer esta opción para una implementación futura.

# Capítulo 9

## Demo de Creativa 2.0

Tal y como mencionábamos al principio del documento, queríamos dar a Creativa 2.0 un enfoque dinámico, tanto para profesores como para alumnos, y saber qué mejoras podíamos llevar a cabo y si se trataba de una aplicación realista que podía ayudar y servir a quienes la usaran. Para ello, decidimos llevar a cabo una demostración en la Facultad de Ciencias de la Información de la Universidad Complutense de Madrid, al grupo de 3º del grado de Comunicación Audiovisual, de la asignatura Literatura y Cine (adaptación). Se trataba de un grupo de 20 personas, obtuvimos retroalimentación de dos maneras: 10 a través de nuestro formulario y los restantes nos redactaron en un fichero de texto su opinión. De este modo, no solo pudimos ver de primera mano, gracias al formulario, cuáles eran los problemas o dudas que más surgían, sino también, de manera más extensa, que nos expusieran sus sugerencias en el fichero de texto.

- Puntos mejor valorados de Creativa 2.0:
  - Los trabajos se presentan en forma de desafíos, lo cual hace que el estudiante se tome la escritura de otra manera.
  - Se puede trabajar en equipo.
  - Está bien enfocada al ámbito pedagógico, sobre todo de parte de los profesores para llevar una evaluación continua.

- La aplicación es completa, es decir, cuenta con todas las funcionalidades que un alumno (que quiera utilizar una aplicación que fomente la escritura creativa) necesita.
- Puntos susceptibles de mejora:
  - Las posibilidades de los estudiantes son limitadas, dado que para gran parte de las acciones se necesita permiso del profesor.
  - Hacer una diferenciación entre la pestaña general “Grupos” y las demás.
  - La interfaz es aburrida.
- Nueva implementación: crear un sistema de recompensas que aumente en función de los desafíos que completes. Dependiendo de la recompensa acumulada podrías obtener más privilegios que otros usuarios.

A continuación, se mostrará un listado más amplio y detallado con las propuestas y sugerencias de mejora que aportaron los estudiantes. Ya que gracias a ellos pudimos analizar los puntos anteriores, y realizar cambios desde la realización de la demo hasta dar por concluido el proyecto. En esta ocasión las aportaciones serán solo de cambios a realizar o de nueva funcionalidad.

- Hacer más secciones y que estas estén más diferenciadas, las académicas, que ya están, y otras por ocio. Por ejemplo: para fomentar más la creatividad y crear mayor interacción con la plataforma. Para ello se podría incluir una sección en la que sean los estudiantes los que puedan hacer sus propias actividades y tener la posibilidad de generar comunidades dentro de la aplicación. Así no estará tan orientada a lo didáctico y despertará mayor interés. Y, de este modo, sería posible seguir llevando una evaluación continua desde aquí.
- Abrir más las posibilidades de los usuarios. Lo primero sería abrir la posibilidad de crear desafíos a todos, no únicamente a los profesores, como ocurre actualmente. De

esta manera, pueden llegar a formarse comunidades entre usuarios, donde aparezcan líderes inesperados que puedan aportar contenido a la aplicación, más allá del que pueda crear un profesor. En esta misma dirección, sería interesante poder leer los escritos de otros usuarios e incluso poder valorarlos, eso sí, con una mecánica donde se evite la toxicidad propia de las redes sociales.

- Sobre el diseño de la página debería hacerse una diferenciación visual entre la pestañas de ‘Grupos’; y las otras tres. Como la pestaña de ‘Grupos’ engloba las otras tres, debería remarcararse para que visualmente se entienda que es la pestaña principal general y las otras son subsidiarias y específicas; o bien, incluirse en la parte superior con secciones más importantes como mensajería, tu perfil o tablero, y el área izquierda limitarse a áreas directas como escritos, equipos o colecciones. Así como una mejora de la aplicación para que sea más intuitiva con una interfaz y estética más atractiva, con más colores y más visual.
- Poder ver y buscar los escritos de los demás usuarios para poder aprender y crear un sentimiento de comunidad dentro de la aplicación y entre todos los usuarios. Así se podría añadir una categoría de escritos más populares, mejor valorados (añadiendo la opción de valorar escritos por otros usuarios) y los desafíos en los que ha participado más gente. Leer los escritos de los demás y ver que la gente los valora puede motivar al resto de personas a mejorar para recibir también valoraciones positivas. Además, se podría añadir la posibilidad de suscribirte a una categoría o similar que notifique a los usuarios cada vez que hay un reto (desafío). O bien, recomendaciones de desafíos para realizar y búsqueda de escritos por palabras clave.
- Poder ver y buscar los demás perfiles y añadir las categorías y *hashtags* que les interesan, y así buscar a gente afín. También sería interesante la opción de cambiar los escritos de público a privado y a la inversa.
- Creación de grupos por parte del alumno y posibilidad de crearlos sin profesores y

que todos dispongan de un chat grupal y un foro de dudas. Así se tendría una manera directa para comunicar dudas o pedir información. También, un chat donde poder hablar con otros usuarios mientras se va escribiendo y así poder compartir las ideas.

- Con respecto a la opción colaborativa, es decir, en equipo, se podría añadir a la página de ‘Escritos’ la opción de dibujar: uno dibuja el cómic, otro escribe el guión; uno escribe la novela y otra ilustra. Por último, añadir moderadores en los escritos en colaborativo.
- Distintas maneras de enfocar Creactiva 2.0:
  - Como un juego con sistema de recompensas. Al pasar el nivel básico (cuento), desbloqueas el siguiente nivel (poesía) por ejemplo. Así con varios niveles (teatro, cine, fotografía, haiku, música, etc.). Como todo juego tendría un desafío final con un reto difícil, pero con una buena recompensa en forma de calificación, por ejemplo.
  - Incluir una recompensa o entender los desafíos como un juego podría hacer que los participantes quisieran participar en un mayor número de desafíos. Para ello, establecer un sistema de puntos con los que pudieses acceder a nuevos niveles, podría establecerse que puedas desbloquear con los puntos nuevos niveles artísticos que empezase con desafíos solo literarios, más tarde desbloquee guiones o desafíos cinematográfico, fotográficos o incluso pictóricos. Además, con estos puntos podría hacerse que puedas adquirir nuevos iconos para tu perfil o que puedas cambiar el color de la aplicación o que te dé alguna ventaja.
  - Que todas las personas (o aquellas que tengan un cierto nivel de puntos o de recompensas que les hagan personas PRO dentro de la aplicación) puedan crear desafíos para hacer una interfaz mucho más participativa. De este modo se hará que las personas estén más interesadas en hacer desafíos y en crearlos para que participen sus amigos o incluso gente que no conocen.

- También podría existir un apartado de ejercicio que consista en emplear las palabras que proponen para esta semana; la próxima semana otro y así continuamente. Como resultado, estarían ganando puntos posicionándose en el ranking de la semana. Para ello que existan diferentes niveles, desde principiante a experto; según la edad y dificultad como vean los usuarios. Además de reforzar la creatividad, está reforzando a un mejor vocabulario los propios miembros.
- Crear una tabla en la zona de escribir a modo de pizarra. Para poder pegar las ideas que vayan surgiendo, poner *post-its* y esquemas con el arco de la historia, personajes e imágenes que inspiren. También añadir un corrector ortográfico, gramatical y de puntuación y una opción de buscar sinónimos y antónimos. Poner más fuentes e incluso *emojis* para hacerlo más creativo. Por si se diera el desafío de tener que crear un texto con *emojis*. Además, añadir en ‘Escritos’ o en una página adicional un diccionario y una biblioteca virtual en la que consultar los temas de los que vas a escribir y recomendaciones de bibliografía en relación a cada tema.

Tras ver el análisis en detalle, pensamos que también sería de ayuda y nos resultaría más fácil visualmente ver la aceptación final de Creactiva 2.0 mediante los gráficos que se presentarán a continuación.

Primero, nos encontramos con un diagrama de sectores donde se preguntaba a los usuarios si recomendarían la aplicación web a terceros. Como se observa, de los 10 que rellenaron el formulario, solo una persona indica que no, aludiendo, como se puede ver en la figura 9.1, que aunque la propuesta es interesante “algo falla”.

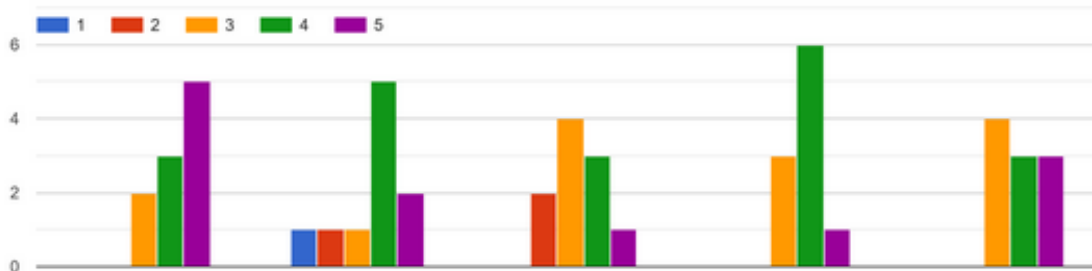
¿Recomendarías Creativa a otras personas? Si no es así, especifica el motivo en "Otro"  
 10 respuestas



**Figura 9.1:** Diagrama de sectores sacado del formulario proporcionado al grupo de la demo.

En el siguiente diagrama de barras (figura 9.2), pedimos a los estudiantes calificar del 1 al 5 cinco aspectos de Creativa 2.0. Recibe siempre el “aprobado” en todos sus puntos a evaluar, destacando la usabilidad, la buena experiencia de usuario y la facilidad para realizar trabajos en grupo.

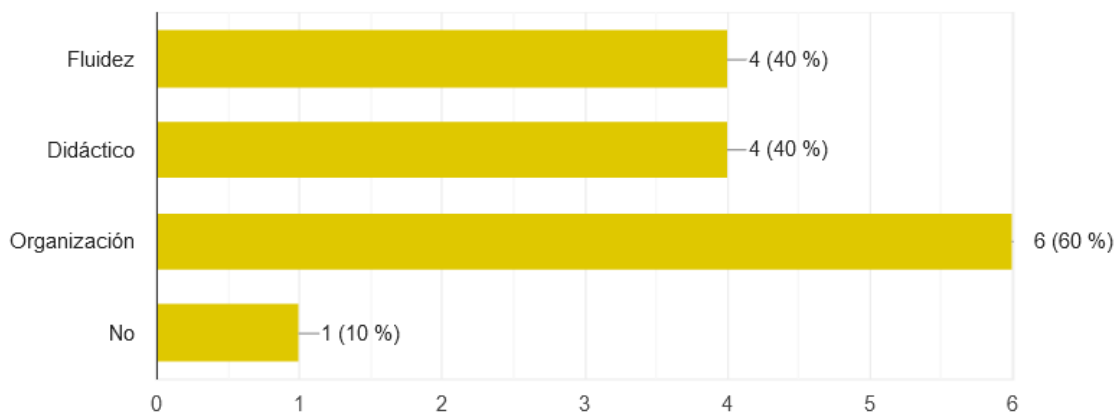
Califica del 1 al 5 los siguientes aspectos



**Figura 9.2:** De izquierda a derecha  
 Me ha resultado fácil de usar  
 La interfaz es clara e intuitiva  
 Me permite obtener mayor retroalimentación que con otros métodos  
 He tenido facilidad para realizar trabajos en grupo  
 He tenido facilidad para encontrar un grupo

De nuevo, se nos presenta un diagrama de barras para medir las respuestas de los usuarios

tras realizar la primera demostración de Creactiva 2.0. Esta vez nos interesaba saber cuál era el punto fuerte de la aplicación para así explotarlo poniendo la vista en el futuro. Por lo que para saber qué ofrece nuestra aplicación que no tengan las demás, decidimos darles ya unas opciones predeterminadas (como se puede observar en figura 9.3): fluidez, didáctico y organización; teniendo también la opción de que no ofrece nada novedoso. Es así como destaca organización, lo cual corrobora su facilidad a la hora de ser usada y su buena experiencia de usuario.



**Figura 9.3:** *Diagrama de barras con las respuestas de los alumnos a: ¿Crees que Creactiva ofrece algo que otras aplicaciones no? Así podemos entender mejor cuáles son los puntos fuertes de la aplicación.*

En general, la aceptación de Creactiva 2.0 ha sido bastante buena. El aspecto más nombrado a corregir ha sido el estético, ya que ha dificultado el uso de la aplicación por no ser del todo intuitiva.

Tras la demo, se analizaron las respuestas obtenidas por los usuarios y se estudió qué puntos de mejora llevar a cabo durante el curso actual. Estos fueron:

- Realización de un apartado de inicio con una explicación detallada del funcionamiento de la totalidad de la aplicación, así como el enfoque de Creactiva 2.0.
- Posibilidad de los estudiantes de poder ver los escritos de otros compañeros.

# Capítulo 10

## Conclusiones y trabajo futuro

En este capítulo desarrollamos nuestras conclusiones e ideas finales, así como las posibilidades futuras en relación a todo el tiempo y esfuerzo dedicado al proyecto. También se ha tenido muy en cuenta el *feedback* proporcionado por los alumnos de la clase de Cora, ya que fueron ellos quienes plantearon muchas de las posibles siguientes implementaciones futuras. Tras todo el trabajo realizado en el proyecto se ha concluido lo siguiente.

### 10.1. Conclusiones

Este trabajo ha tenido como objetivo la inserción de nuevas herramientas que mejorasen la escritura creativa. Una vez tuvimos la aplicación base con sus funcionalidades empezamos a realizar observaciones y modificaciones sobre la misma, las cuales venían indicadas en la propuesta de trabajo de futuro de Creactiva 1.0. Esto provocó que actualizásemos tanto el aspecto visual del front-end como el aspecto técnico del back-end. Para este fin fue muy importante la estructura de la base de datos y todos los cambios que se hicieron sobre ella para soportar los nuevos campos y datos que necesitan las extensiones realizadas. Al hablar de aspectos técnicos se han añadido, modificado y actualizado diversos módulos de la aplicación ya citados con anterioridad para su correcto funcionamiento.

Como mención especial está el despliegue de la aplicación, así como las diversas modificaciones para este fin que no pudo ser realizado para esta versión. Nuestro primer acercamiento fue con *Microsoft Azure*, utilizando el crédito proporcionado por la universidad, en el cual

conseguimos desplegar la base de datos y conseguir una versión local interconectada para la prueba de los textos colaborativos, así como del *script* para la actualización de dichos textos. Con estas pruebas y los diversos intentos de desplegar la aplicación en sí misma acabábamos con el crédito disponible y nos propusimos diversas alternativas de las cuales exploramos:

- **Digital Ocean.** Ofrece un servicio extenso de *Droplets* y *Kubernetes*, así como de base de datos y espacio de almacenamiento en la nube. Tiene un sistema nuevo (APP) que te permite desplegar directamente tu aplicación desde el repositorio de Github. No fue de utilidad dado a errores que surgían al intentar desplegar.
- **Github Pages/Github.io.** Ofrece servicios para hacer *hosting* y crear paginas web directamente desde el repositorio de github. Este servicio no nos fue de utilidad por incompatibilidades con nuestra aplicación y las tecnologías de React.
- **Amazon Web Services (AWS).** Es uno de los servicios mas amplio junto a Azure y que ofrece todo tipo de servicios en la nube. Fue una de nuestras ultimas opciones de despliegue ya que su versión gratuita no abarca lo suficiente para el despliegue total de la aplicación.

Hemos tenido en cuenta que este proyecto, dado que es una aplicación web, requiere de una fuerte capa de presentación, con facilidades para que la experiencia de usuario sea lo mas cómoda posible debido a que el usuario es la prioridad en este modelo de proyecto. Para este fin tuvimos en consideración la retroalimentación obtenida en la presentación a los alumnos de la clase de demostración, así como las encuestas que los estudiantes pudieron rellenar después de la prueba funcional de la propia aplicación, incluyendo en la aplicación las peticiones permisibles y añadiendo todas las posibles funcionalidades futuras para su próxima versión.

## 10.2. Trabajo futuro

Con el trabajo realizado se ha logrado el esqueleto de lo que sería el proyecto completo, pero queda trabajo para conseguir un aplicación totalmente funcional. Los puntos principales que se deberían desarrollar son:

- **Despliegue de la base de datos.** Utilizando un servicio que se encargue de asegurar de que sea eficiente en cuanto a coste/peticiones. La aplicación requiere de un modelo para que tenga actualizaciones constantes en cuanto a los escritos colaborativos, por lo que la elección del proveedor es primordial.
- **Despliegue de la aplicación en la nube.** El objetivo de la aplicación es la promoción de la escritura creativa, y para este fin, y que la aplicación funcione de verdad, uno de los principales pilares es la escritura colaborativa, para la cual es necesaria el despliegue total de la aplicación.
- **Añadir retos y recompensas.** Completar distintos retos mientras realizas escritos puede ser una idea muy interesante que despierte el interés de los estudiantes. Esto se asimila al funcionamiento de muchos juegos, y más aún si se introduce también un sistema de recompensas según el número de retos que completes.
- **Implementación de un chat a tiempo real entre los usuarios.** Una aplicación de esta índole se beneficiaría de las interacciones entre usuarios para con los retos y objetivos propuestos.
- **Desarrollo de una versión móvil/versión lite.** Utilizando una versión mas portable y manejable, se podría conseguir que la experiencia de usuario mejorara y facilitaría a los individuos el uso de esta.

Existen muchas futuras opciones y posibilidades, junto al trabajo que quedaría por delante para finalizar el proyecto. Durante el desarrollo se ha creado un prototipo de aplicación

para su uso local que podría complementarse, de modo que, en un entorno en la nube se podría ampliar y lograr a una escala mayor el objetivo principal del proyecto.

# Capítulo 11

## Conclusions and Further work

In this chapter we develop our conclusions and final ideas, as well as future possibilities in relation to all the time and effort dedicated to the project. The feedback provided by the students in Cora's class has also been taken into account, because they were the ones that introduce us the most part of the following possible implementations that could be made in the future. After all the work done in the project we have concluded the following.

### 11.1. Conclusions

This work has been aimed at creating new tools that could improve creative writing. Once we had the base application with its functionalities we started to make observations and modifications, which were indicated in the section 'Further work' from *Creactiva 1.0*. This made that we had to update both the visual aspect of the front-end and the technical aspect of the back-end. For this purpose it was very important the database structure and all the changes that were made on it to support the new fields and data needed by the extensions made. When talking about technical aspects, we have added, modified and updated several modules of the application already mentioned before for its correct functioning.

As a special mention is the deployment of the application, as well as the various modifications for this purpose that could not be done for this version. Our first approach was with *Microsoft Azure*, using credit provided by the university, in which we managed to deploy the

database and get a local version interfaced for testing of the collaborative texts, as well as the *script* for the update of these texts. With these tests and the various attempts to deploy the application itself we ran out of available credit and we proposed several alternatives which we explored:

- **Digital Ocean.** It offers an extensive service of *Droplets* and *Kubernetes*, as well as database and cloud storage space. It has a new system (APP) that allows you to directly deploy your application from the Github repository. It was not useful due to errors that arose when trying to deploy.
- **Github Pages/Github.io.** Offers services to make *hosting* and create web pages directly from the github repository. This service was not useful for us due to incompatibilities with our application and React technologies.
- **Amazon Web Services (AWS).** It is one of the most comprehensive services next to Azure and offers all kinds of cloud services. It was one of our last deployment options since its free version does not cover enough for the full deployment of the application.

We have taken into account that this project, since it is a web application, requires a strong presentation layer, with facilities to make the user experience as comfortable as possible because the user is the priority in this project model. For this purpose we took into consideration the feedback from the presentation to the students of Cora's class, as well as the surveys that the students were able to fill in after the functional test of the application itself, including in the application the allowable requests and adding all possible future functionalities for its next version.

## 11.2. Further work on next versions

With the work done we have achieved the skeleton of what would be the complete project, but there is still work to be done to achieve a fully functional application. The main points that should be developed are:

- **Database deployment.** Using a service that will ensure that it is cost/request efficient. The application requires a model to have constant updates in terms of collaborative writing, so the choice of vendor is paramount.
- **Deployment of the application in the cloud.** The objective of the application is the promotion of creative writing, and to this end one of the main pillars is collaborative writing, for which the full deployment of the application is necessary.
- **Add dares and rewards.** Completing different dares while you are making writings can be a very interesting idea that could call students attention. This is similar to how many games work, and even more if a reward system that changes according to the number of completed dares is added.
- **Implementation of a real-time chat between users.** An application of this nature would benefit from the interactions between users in order to meet the challenges and objectives proposed.
- **Development of a mobile version/lite.** By using a more portable and manageable version, the user experience could be improved and it would make it easier for individuals to use it.

There are many future options and possibilities, along with the work that would remain ahead to finalize the project. During development, a prototype application has been created for local use that could be complemented, so that in a cloud environment the main objective of the project could be scaled up and achieved on a larger scale.

# Bibliografía

- [1] A. Nandaa. *Beginning API Development with Node.js*. Packt Publishing, 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] Elizabeth Naramore and 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. Somasundaram. *Git: Version control for everyone*. Packt Publishing, Limited, 2013. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=1119761&query=>.
- [5] A. P. Umesh, R Sharma, and Jason Gerner. *Github essentials : Unleash the Power of Collaborative Development Workflows Using GitHub*. Packt Publishing, Limited, 2018. <https://ebookcentral.proquest.com/lib/universidadcomplutense-ebooks/reader.action?docID=5446049>.
- [6] D. Megida. Multer and Node. <https://blog.logrocket.com/uploading-files-using-multer-and-node-js/>.
- [7] J. Puri. React draft wysiwyg. <https://jpuri.github.io/react-draft-wysiwyg/#/docs>.
- [8] Matt Zabriskie. Axios. [https://axios-http.com/docs/api\\_intro](https://axios-http.com/docs/api_intro). T. A. A. Reference.

- [9] jsPDF. <https://www.npmjs.com/package/jspdf>.
- [10] Manuel Cillero. Visual studio code. <https://manuel.cillero.es/doc/apuntes-tic/herramientas/visual-studio-code/>.
- [11] Github desktop. <https://docs.github.com/es/get-started/using-github/github-desktop>.
- [12] Discord. <https://es.wikidat.com/info/discord>.
- [13] Overleaf. <https://www.uc3m.es/sdic/servicios/overleaf>.
- [14] Reedsybookeditor. [https://blog.reedsy.com/writing-apps/#1\\_\\_reedsy\\_book\\_editor](https://blog.reedsy.com/writing-apps/#1__reedsy_book_editor), 2020.
- [15] Scrivener. [https://blog.reedsy.com/writing-apps/#2\\_\\_scrivener](https://blog.reedsy.com/writing-apps/#2__scrivener), 2020.
- [16] Ulysses. [https://blog.reedsy.com/writing-apps/#3\\_\\_ulysses](https://blog.reedsy.com/writing-apps/#3__ulysses), 2020.
- [17] Evernote. [https://blog.reedsy.com/writing-apps/#16\\_\\_evernote](https://blog.reedsy.com/writing-apps/#16__evernote), 2020.