

Extensión de “Lecto entreno”, con una visión en la recopilación y visualización de datos.

Memoria de título

Autor: Michael Ignacio Villanueva Torres

Profesor patrocinante: Geoffrey Hecht

Profesoras co-patrocinantes: Pamela Guevara y Mabel Urrutia Martinez

© 2024 Michael Villanueva Torres

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que me apoyaron durante este proceso y que, de una forma u otra, hicieron posible la realización de este proyecto.

En primer lugar, mi gratitud a Geoffrey Hecht por la guía y paciencia que me brindó a lo largo de todo el camino. Sus sugerencias y observaciones fueron esenciales para la concreción de esta memoria.

A mi familia, especialmente a mi hermana, por su apoyo incondicional y su constante aliento en esta etapa final. Un agradecimiento especial a mi pareja, quien, en los momentos más difíciles, supo levantarme y devolverme la confianza en mí mismo y en lo que estaba realizando.

Quiero también expresar mi gratitud a los profesores del proyecto FONDECYT, cuyas reuniones y aportes fueron vitales para mejorar el software de manera efectiva. Sin su apoyo, este proyecto no habría sido posible.

Finalmente, a mis amigos y compañeros de estudio, quienes con su alegría y compañía hicieron que este camino fuera más llevadero y gratificante.

Dedico este trabajo a mi abuelita, quien, a pesar de la distancia, siempre me brinda su cariño y motivación para seguir adelante.

Agradecimientos al proyecto

Agradecimientos al Proyecto Fondecyt Regular N°1241145, titulado: “Entrenando la comprensión lectora online mediante una plataforma virtual. Evidencias empíricas desde la Neurociencia” por el financiamiento de esta investigación, brindarme la oportunidad de implementar mejoras a la aplicación LECTO ENTRENO y ayudarme en cada reunión para desarrollar el software.

Índice

1. Introducción.....	4
2. Problemáticas y propuestas.....	7
3. Objetivo general.....	10
4. Objetivo específicos.....	10
5. Diagrama de casos de uso.....	11
6. Modelo de Entidad-Relación (MER).....	14
7. Arquitectura.....	16
8. Metodología.....	17
9. Tecnologías Utilizadas:.....	19
10. Implementación:.....	21
10.1 Vista de Administrador:.....	21
10.2. Panel de control de Actividades:.....	23
10.3. Función para que los profesores compartan asignaturas:.....	28
10.4. Mejoramiento del proceso de actividades para que sea más flexible:.....	29
10.5. Nuevas estrategias implementadas:.....	31
10.6. Creación de Diagnósticos y muestra de datos:.....	35
10.7. Mejoras de diseño y funcionalidades en el software:.....	38
11. Despliegue.....	41
11.1. Preparación y despliegue del Frontend:.....	41
11.2. Preparación y despliegue de la API:.....	41
12. Test y validaciones.....	42
12.1. Test Unitarios.....	42
12.2. Validación de usabilidad.....	43
13. Conclusiones.....	48
14. Recomendaciones para el futuro.....	49
14.1. Funcionalidad para compartir textos:.....	49
14.2. Incorporación de IA para corrección automática:.....	49
14.3. Actualización y optimización de tecnologías:.....	50
15. Referencias.....	51

Resumen

Esta memoria se enfoca en el desarrollo y mejoramiento de "LectoEntreno", un software orientado a mejorar la comprensión lectora en entornos educativos. El proyecto, que surge como una evolución de un software previo, tiene como objetivo optimizar la recopilación y visualización de datos, además de mejorar la experiencia de usuario tanto para estudiantes como para profesores.

Entre las principales mejoras implementadas destacan la creación de un panel de control de actividades que incorpora gráficos y datos visuales, así como la opción de exportar estos datos a Excel. Se añadieron nuevas funcionalidades, como la edición y aplazamiento de actividades, y se integró un perfil de administrador que permite la creación y gestión de usuarios.

Adicionalmente, se desarrollaron estrategias pedagógicas, incluyendo la asociación de minijuegos a las actividades, un minijuego de imágenes o de ordenar párrafos, funciones para que las palabras del texto tengan una imágenes relacionada, y también se mejoró el apartado de los diagnósticos para evaluar el progreso de los estudiantes de manera más precisa.

Se llevaron a cabo test unitarios y una validación mediante el Sistema de Usabilidad de Software (SUS), en la cual el software obtuvo un puntaje favorable, confirmando su eficacia y usabilidad.

Comparada con la versión anterior, esta iteración del software presenta una experiencia de usuario más completa, robusta y eficiente, alcanzando los objetivos planteados al inicio del proyecto y proporcionando una base sólida para futuras mejoras y expansiones.

Summary

This report focuses on the development and improvement of “LectoEntreno”, a software oriented to improve reading comprehension in educational environments. The project, which arises as an evolution of a previous software, aims to optimize data collection and visualization, in addition to improving the user experience for both students and teachers.

The main improvements implemented include the creation of an activity control panel that incorporates graphs and visual data, as well as the option to export this data to Excel. New functionalities were added, such as editing and postponing activities, and an administrator profile was integrated to allow the creation and management of users.

In addition, pedagogical strategies were developed, including the association of mini-games to the activities, a mini-game of images or of ordering paragraphs, functions so that the words in the text have a related image, and the diagnostic section was also improved to evaluate the progress of the students more accurately.

Unit tests and a validation through the Software Usability System (SUS) were carried out, in which the software obtained a favorable score, confirming its effectiveness and usability.

Compared to the previous version, this iteration of the software presents a more complete, robust and efficient user experience, achieving the objectives set at the beginning of the project and providing a solid foundation for future improvements and expansions.

1. Introducción

El analfabetismo funcional se refiere a la incapacidad de comprender y analizar lo leído adecuadamente. Quienes padecen esta condición pueden leer fluidamente en voz alta, pero enfrentan dificultades para interpretar el significado de lo que acaban de leer. Esta limitación afecta tareas diarias, tales como seguir instrucciones o leer noticias en redes sociales.

En Chile, la comprensión lectora ha representado un desafío constante en el ámbito educativo. Aunque se han logrado avances significativos que sitúan al país entre los mejores de América Latina, aún nos encontramos por debajo del promedio de la OCDE (Organización para la Cooperación y el Desarrollo Económicos)[1]. Según los resultados de la prueba PISA de 2018, en una escala del nivel 1 al 6, el 31.7% de los estudiantes chilenos se ubicaba por debajo del nivel 2, nivel deficiente, y solo un 2.6% alcanzaba los niveles 5 y 6 (Ver figura 1). El 31,7% que obtuvo un bajo rendimiento, según estudios de la OCDE, tendrá dificultades para desarrollarse, enfrentará dificultades significativas para continuar sus estudios o desempeñarse eficazmente en el mundo laboral.

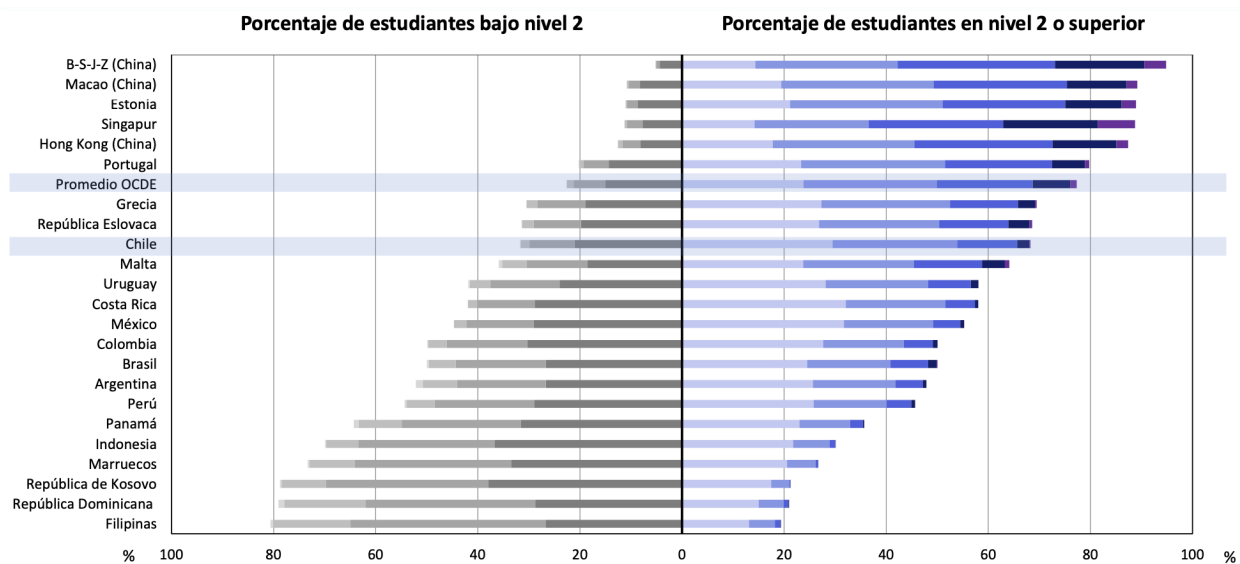


Figura 1: Niveles de Desempeño de Lectura - Comparación Internacional

Esta memoria se deriva "LECTOENTRENO: SOFTWARE PARA APOYO A LA COMPRENSIÓN LECTORA" [2], desarrollado por Ignacio Sanhueza. Este trabajo se llevó a cabo durante el año 2021 y se continuó desarrollando después de haber concluido la memoria.

Lecto Entreno es una aplicación web diseñada con el objetivo de ayudar a estudiantes a mejorar su comprensión lectora y combatir el analfabetismo funcional, esto se realiza mediante textos medianos, cortos y actividades evaluativas. Los estudiantes desarrollan actividades a través de una asignatura y en estas actividades tienen que llevar a cabo la lectura de estos textos incorporando funciones que se han diseñado para fomentar una mayor interacción con el texto y facilitar así una mejor comprensión, después de leer para ver si comprenden lo leído se le realiza una evaluación, funcionalidades como la opción de guardar palabras clave o realizar preguntas en los párrafos que no comprendan. El estudiante realiza todo esto por medio de su pantalla de inicio (Ver Figura 2), aquí se le ven las asignaturas y con sus actividades y encuestas asociadas.

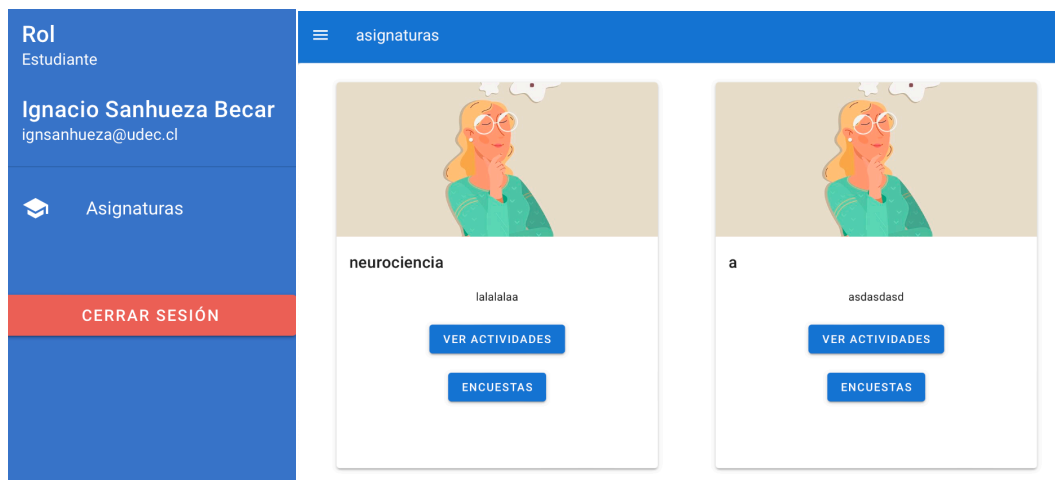


Figura 2: SideBar del estudiante y vista principal del menú de asignaturas

El docente es el actor más importante ya que puede realizar más funciones en el software, puesto que pueden crear asignaturas, textos, y diagnósticos por medio de su SideBar (Ver figura 3), con el objetivo de que dentro de estos ramos se generen actividades para que los alumnos puedan mejorar y desarrollar su comprensión lectora. En una asignatura se pueden crear actividades, asociar una encuesta creada y, además, cada docente tiene la capacidad de obtener un archivo Excel con los datos y resultados de las evaluaciones de un alumno, lo que les permite realizar un avistamiento del progreso del estudiante.

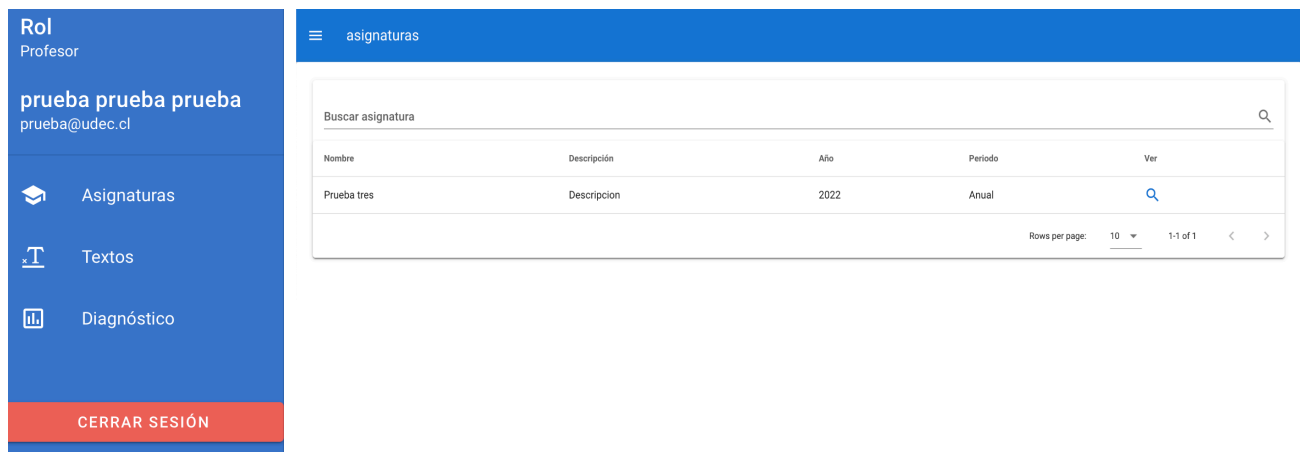


Figura 3: SideBar de Navegación del profesor y vista del menú de asignaturas

2. Problemáticas y propuestas

Para recopilar todas las problemáticas, se realizaron reuniones con las profesoras copatrocinantes durante todo el semestre, al finalizar este proceso, se unieron a las reuniones profesores que forman parte del proyecto FONDECYT, titulado ‘Entrenando la comprensión lectora online mediante una plataforma virtual. Evidencias empíricas desde la Neurociencia’ el cual busca evaluar los efectos de un entrenamiento mediante el software [3].

Durante las primeras reuniones se identificó una problemática clave relacionada con la obtención y manejo de los datos dentro de la aplicación, una función esencial para los profesores. En las actividades, los puntajes de los estudiantes podían ser recuperados de dos maneras: a través de un listado con todas las respuestas de cada estudiante, mostrada por medio de una página individual por alumno, o mediante un archivo Excel que organizaba toda la información por secciones con cada pregunta y una sección final que mostraba los puntajes. Las profesoras solían optar por la segunda opción, ya que revisar cada respuesta de forma individual demandaba demasiado tiempo y era propenso a errores. Sin embargo, incluso el formato del Excel presentaba inconvenientes, como por ejemplo la falta del tiempo global o el uso de un separador “/” en los puntajes, lo que dificulta la manipulación y análisis de los datos. La obtención de datos se convirtió en un proceso ineficiente que complicaba la evaluación de las actividades realizadas.

Hasta ahora, el software "Lecto Entreno" mantiene actividades para los estudiantes donde se leen textos, los cuales tiene opciones como, manejar el tamaño de la letra, buscar el significado de una palabra en el diccionario de la Real Academia Española (RAE) y, en caso de que el texto sea

de tipo “minijuego”, se les aplica un minijuego para ordenar párrafos en la lectura a todas las actividades relacionadas con el texto correspondiente.

Los profesores desean obtener mejores datos de estas actividades sin tener la necesidad de descargar el excel y, a su vez, desean manejar de mejor manera la aplicación, ya que ellos no podían crear diagnósticos, ni perfiles de usuario profesor. En este sentido, el sistema representaba un desafío, ya que la obtención de datos representa un eje fundamental para comprobar la eficacia de los minijuegos y extras en la lectura, y así poder evaluar si los estudiantes mejoran la comprensión lectora.

La mayoría del frontend en Vue requería retoques de accesibilidad, ya que muchos componentes no tenían títulos y no se intuía a qué se asociaba el contenido o qué se debía realizar, por lo que resultaba complicado utilizar la aplicación, tanto a profesores como estudiantes. La problemática principal que se abordó en esta memoria fue la obtención de buenos datos de las actividades y la posibilidad de manejar el sistema a gusto y que no se ingresen datos por código, lo que se realizaba antiguamente para ingresar diagnósticos y generar perfiles de profesor.

Ahora se recapitulan todas las problemáticas encontradas y discutidas dentro de las reuniones con las profesoras.

Problemas dentro del Software:

1. Gestión de asignaturas y compartición de información entre docentes:

- **Problema:** Más de un profesor no puede formar parte del mismo curso.
 - Discusión: Dentro del software, había ocasiones donde 2 profesores formaban parte de un ramo, llegaba algún estudiante en práctica o un profesor de ayuda para la asignatura, y tenían que compartir el perfil de usuario para poder obtener los datos de una actividad, crear textos y todo lo relacionado con la asignatura, lo que es muy poco eficiente.
- **Propuesta:** Permitir que los profesores puedan compartir una lección, de manera que dos o más profesores puedan ver y gestionar la misma asignatura, creando actividades, asociando sus textos agregados en su perfil y compartiendo los datos de todas las actividades de la asignatura.

2. Flexibilidad en la programación de actividades:

- **Problema:** Rigidez en la programación de actividades, afectando a estudiantes que por motivos justificados no pueden cumplir con las fechas establecidas.
 - Discusión: Al momento de crear una actividad común, o una de diagnóstico, se le ingresa la fecha límite que tienen los estudiantes para

poder realizarla. Es así, que cuando los profesores se enfrentan al caso donde los estudiantes por x o y motivo no pueden realizar la actividad, no se puede aplazar la actividad para el estudiante o para la clase entera.

- **Propuesta:** Implementar una funcionalidad que permita reprogramar las actividades para los estudiantes que no pueden realizar la actividad en la fecha inicialmente asignada, esto es para un estudiante en específico y adicionalmente se podrá editar los parámetros globales de una actividad permitiendo que se pueda retrasar la fecha límite para toda la asignatura.

3. Creación de perfiles de docentes y manejo de usuarios:

- **Problema:** Limitaciones en la creación y gestión autónoma de perfiles de docentes.
 - **Discusión:** En el sistema hasta ahora, se asignaba manualmente el rol de docente dentro del sistema a los usuarios ya creados y así se generaban perfiles de profesor, los docentes no podían crear perfiles para otros profesores ni editar o eliminar un usuario.
- **Propuesta:** Desarrollar un perfil de administración para los profesores, donde aquí se permita la creación y gestión de perfiles de usuario de manera eficiente, permitiendo realizar correcciones y actualizaciones cuando sea necesario.

4. Obtención de datos de una actividad:

- **Problema:** Dificultades en la recolección y análisis de datos de las actividades estudiantiles.
 - **Discusión:** Los profesores necesitan obtener datos de las actividades, ver los puntajes obtenidos y el tiempo que se demoraron los estudiantes, para saber a cuales les dificulta más la lectura. Este proceso es tedioso e ineficiente, el sistema entrega un listado con todas las respuestas de cada estudiante, mostrada por medio de una página individual, o un excel con todas de las respuestas del estudiante a una actividad. Los profesores siempre elegían la opción del excel, ya que esta opción no tenía posibilidad de errores al contabilizar los puntajes, pero el excel mantenía un formato no deseado por los docentes, entregando las preguntas por separado y el puntaje con "/" que no les dejaba manejar los datos de manera simple en el excel.
 - **Propuesta:** Crear un panel de control de actividades como parte de una asignatura, que permita ver los datos de esta, ver las respuestas de cada estudiante y descargar un excel que contenga todas las respuestas de los estudiantes a esa actividad y un resumen de esta.

5. Error al momento de responder una actividad:

- **Problema:** Pérdida de información cuando los estudiantes necesitan reiniciar o actualizar la página durante una actividad.
 - **Discusión:** Al realizar una actividad, inicia la toma de tiempo, en donde los estudiantes primero tienen que leer el texto y posteriormente deben responder preguntas para terminar la actividad. Sin embargo, si el estudiante actualiza la página, o vuelve atrás durante su actividad, el tiempo vuelve a 0 al apartado de lectura sin guardar los datos.
- **Propuesta:** Guardar los datos en el almacenamiento local del navegador del usuario, para así no perder información, se restablecerá los datos al generar el componente y se eliminarán al terminar la actividad.

6. Seguimiento de actividades completadas:

- **Problema:** Los docentes no tienen la capacidad de verificar cuántos de los estudiantes han completado las actividades de una asignatura.
 - **Discusión:** Cuando los profesores realizaban una actividad no sabían cuáles estudiantes habían terminado o no la actividad, para así preguntar a los estudiantes que les falta terminar y sus dificultades al realizar la actividad o la lectura.
- **Propuesta:** Por medio del dashboard de una actividad, mostrar un gráfico de los estudiantes que han terminado o no de realizar la actividad.

7. Creación de diagnósticos en el sistema:

- **Problema:** No se puede crear un diagnóstico por parte de los profesores.
 - **Discusión:** Dentro del software existen los diagnósticos, los cuales son encuestas que pueden ser aplicadas en alguna asignatura por medio de una actividad de diagnósticos, estas son contestadas por los estudiantes y los profesores pueden ver sus respuestas al igual que las actividades normales por medio del excel. Pero los diagnósticos que están en el sistema fueron agregados manualmente, por lo que los profesores no tienen el poder de crear un diagnóstico.
- **Propuesta:** Se creará un apartado donde se puedan crear los diagnósticos, agregando las pregunta y respuestas asociadas.

3. Objetivo general

Desarrollar un sistema que facilite a profesores y estudiantes el logro de sus objetivos educativos, ya sea enseñar o mejorar la comprensión lectora, mediante herramientas y datos necesarios para una gestión efectiva del software.

4. Objetivo específicos

- Desarrollar e implementar un perfil de administración que facilite la creación y gestión de usuarios.
- Implementar una funcionalidad para que los profesores compartan información de las asignaturas.
- Desarrollar un panel de control de las actividades por asignatura que facilite la obtención de datos, mediante gráficos de resultados obtenidos y de término de una actividad. Integrando una funcionalidad para reprogramar actividades.
- Desarrollar una sección para la creación de diagnósticos.
- Integrar nuevas estrategias destinadas a mejorar la comprensión lectora.
- Realizar mejoras en el diseño y código de las vistas de docente y estudiante, mejorando apartados y funcionalidades del sistema actual.
- Evaluar la usabilidad y efectividad del sistema mediante pruebas con usuarios finales en entornos académicos reales.

5. Diagrama de casos de uso

- Se actualizó el diagrama de casos de uso para así documentar los flujos de interacción entre los usuarios (estudiantes y profesores) y el sistema. Utilizando herramientas como draw.io.
- Dentro de los diagramas para representar el software, el color amarillo indica que se ha modificado ese caso de uso en el proceso de esta memoria y el color rojo es la creación de uno.
- Los diagramas proporcionaron una representación visual detallada de los requisitos funcionales, mejorando la comprensión del sistema y facilitando el desarrollo de nuevas características.

Ambos diagramas se abstienen de los casos de uso del login y autenticación, donde también se agrega el caso de uso de recuperación de contraseña, esto se aplica para todos los actores.

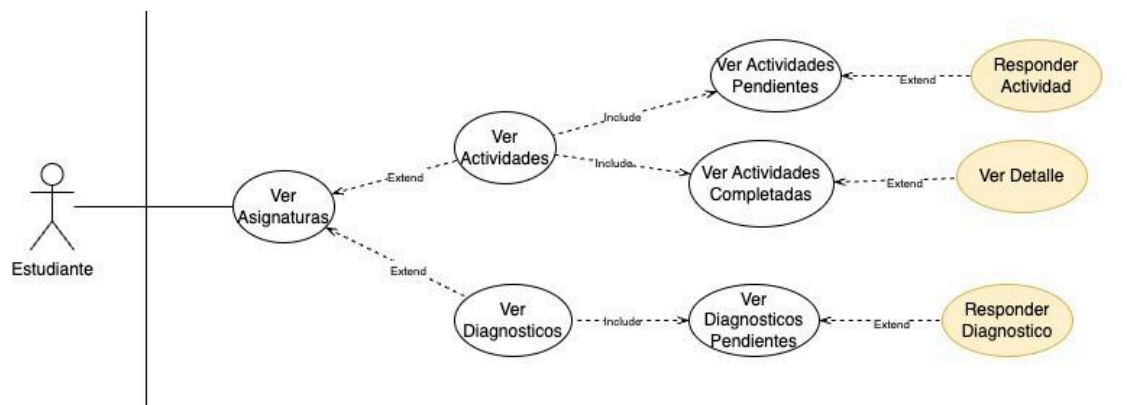


Figura 6 : Diagrama de casos de uso actualizado del actor estudiante

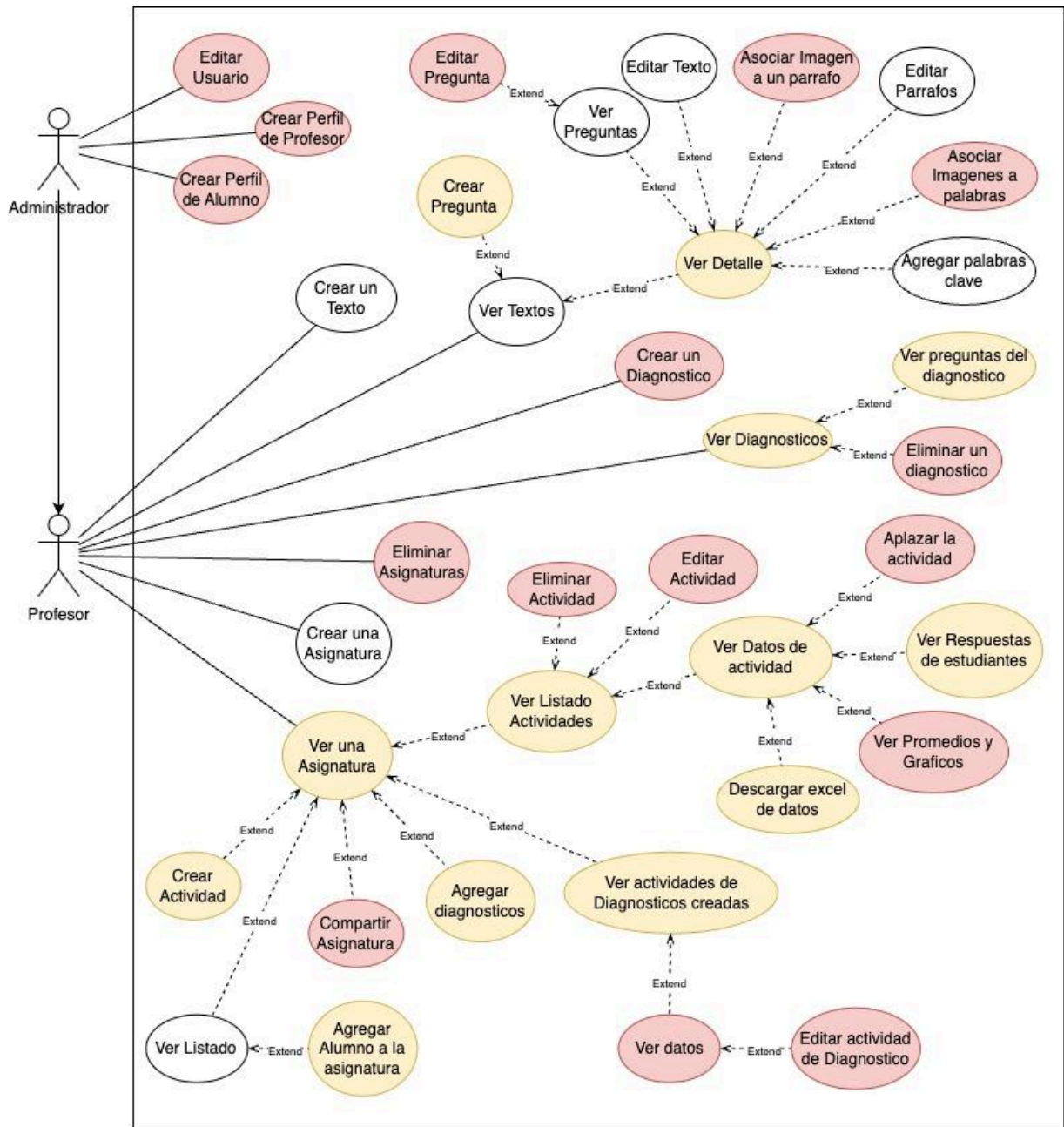


Figura 7 : Diagrama de casos de uso¹ actualizado del actor estudiante profesor y administrador

¹ Diagramas: https://drive.google.com/file/d/1mVhe0GB_h8EsqVnti42-1IDK0ROAPem9/view?usp=sharing

6 . Modelo de Entidad-Relación (MER)

- Se utilizó draw.io para realizar el MER, reflejando tanto datos como relaciones existentes en el software. Este proceso se llevó a cabo en paralelo con la revisión del código para garantizar la coherencia entre la base de datos y las funciones del software.
- El MER actualizado proporcionó una visión clara de la organización de los datos y facilitó la planificación de nuevas funcionalidades y optimizaciones de las existentes.
- En el diagrama (Ver Figura 8) las tablas con color amarillo representa las tablas donde se realizaron modificaciones.
- Para acortar espacio, en vez de utilizar los círculos para representar los atributos, se ocuparon tablas y la Priority Key (PK) es el identificador. La entidad Types no tiene relaciones directas con otras entidades en el MER, se utiliza únicamente para almacenar información de referencia sobre los tipos que pueden estar asociados a otras entidades. Aquí se describen algunas relaciones importantes.
 - Users y Lesson: Cada usuario (profesor o estudiante) puede estar asociado a múltiples lecciones así como una lección puede estar asociado tanto a varios profesores como varios estudiantes, ya que los profesores pueden compartir una asignatura.
 - Lesson y Activity: Cada asignatura puede contener varias actividades y una actividad pertenece únicamente a una asignatura.
 - Activity y Questions: Las actividades pueden incluir múltiples preguntas de un texto y como las Questions están asociadas a un texto se pueden repetir en múltiples actividades.
 - Questions y Text: Un texto contiene varias preguntas y estas preguntas están asociadas a un texto único.
 - Text y Paragraph: Un texto puede contener múltiples párrafos. La entidad paragraphs almacena estos párrafos y su relación con el texto correspondiente.

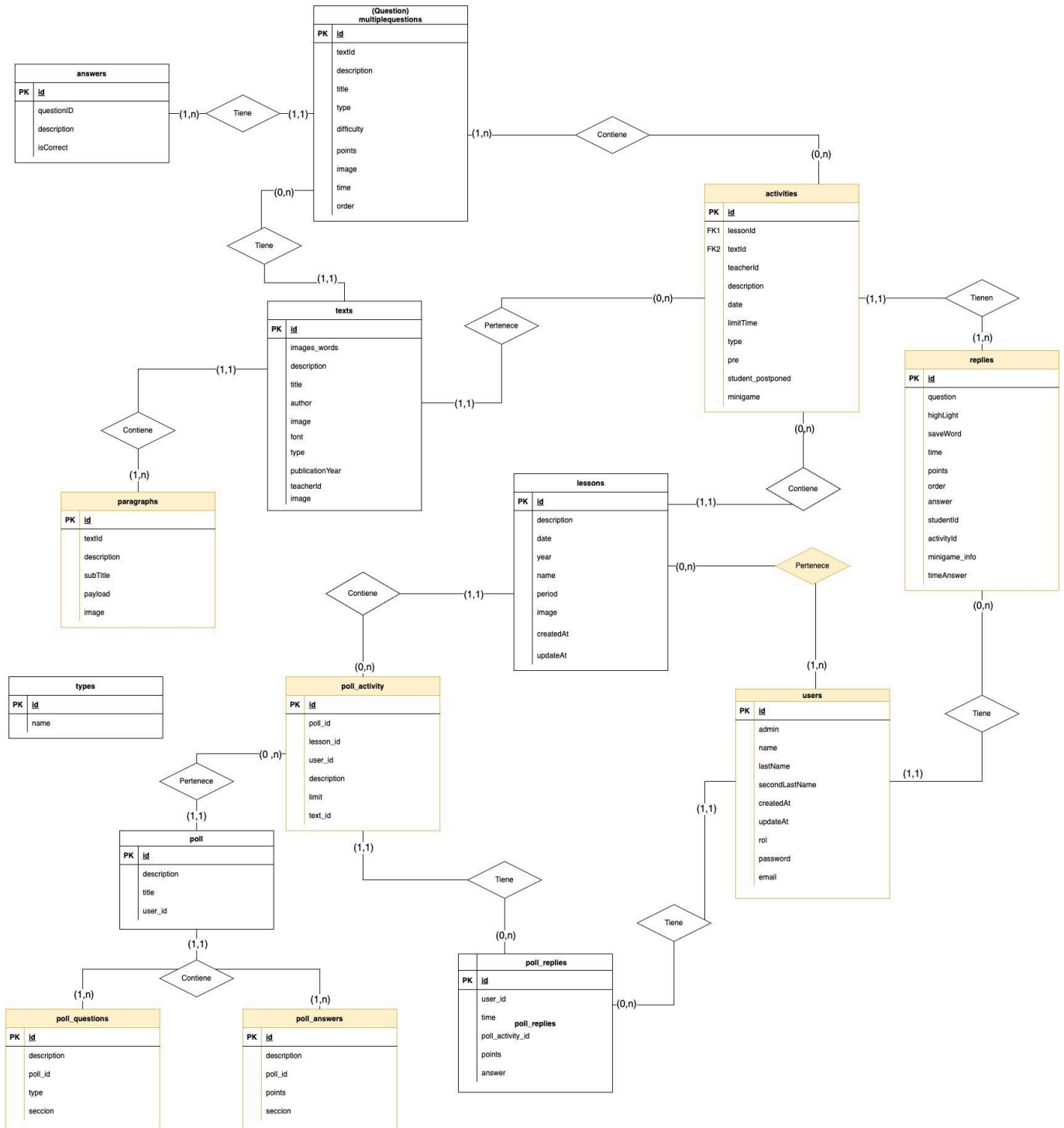


Figura 8 : Modelo de entidad relación² actualizado de base de datos del software Lecto Entrenó

² MER: <https://drive.google.com/file/d/1533z57Gcl17IHvagvePTi59zvglBEckI/view?usp=sharing>

7. Arquitectura

Modelo C4:

Para demostrar una vista general a la arquitectura del software se realizó un modelo C4 [7], realizado hasta nivel 2, mostrando el contexto y los contenedores.

Nivel de contexto: Dentro del sistema (Ver Figura 4) los 2 únicos actores son los profesores y los estudiantes, los profesores tiene la facultad de obtener el poder de administrador, ambos utilizan el software para desempeñar sus tareas y el único sistema externo al software es el sistema de email el cual se utiliza para enviar correos a los usuarios.

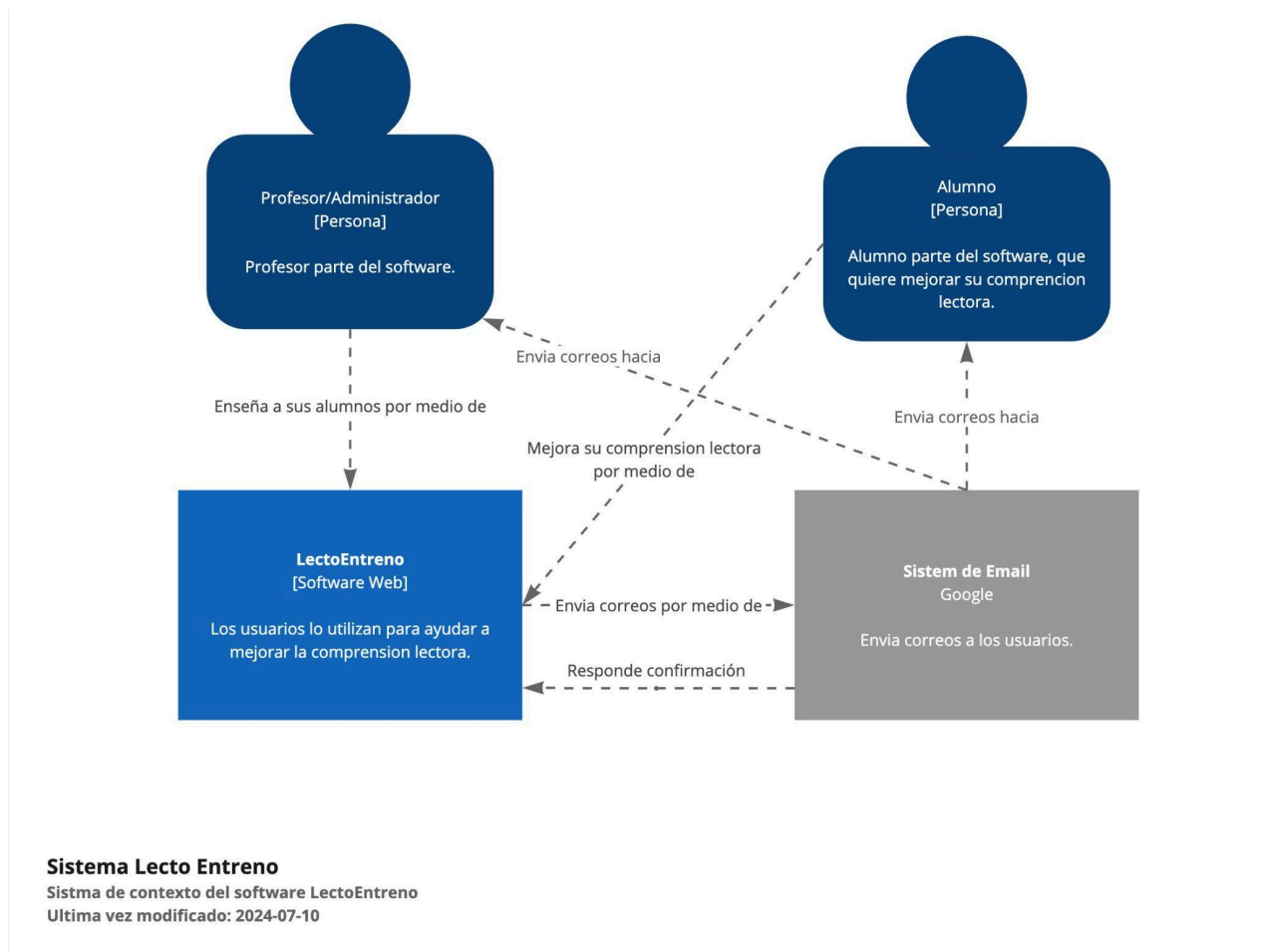


Figura 4: Nivel de contexto, parte del modelo C4 en el software Lecto Entreno

Nivel de contenedores: Aquí se realiza un zoom al software de Lecto Entreno (Ver figura 5) donde encontramos los contenedores de la api, la aplicación web y la base de datos, indicando cómo interactúan con el fin de administrar los datos y que los usuarios puedan utilizar el software.

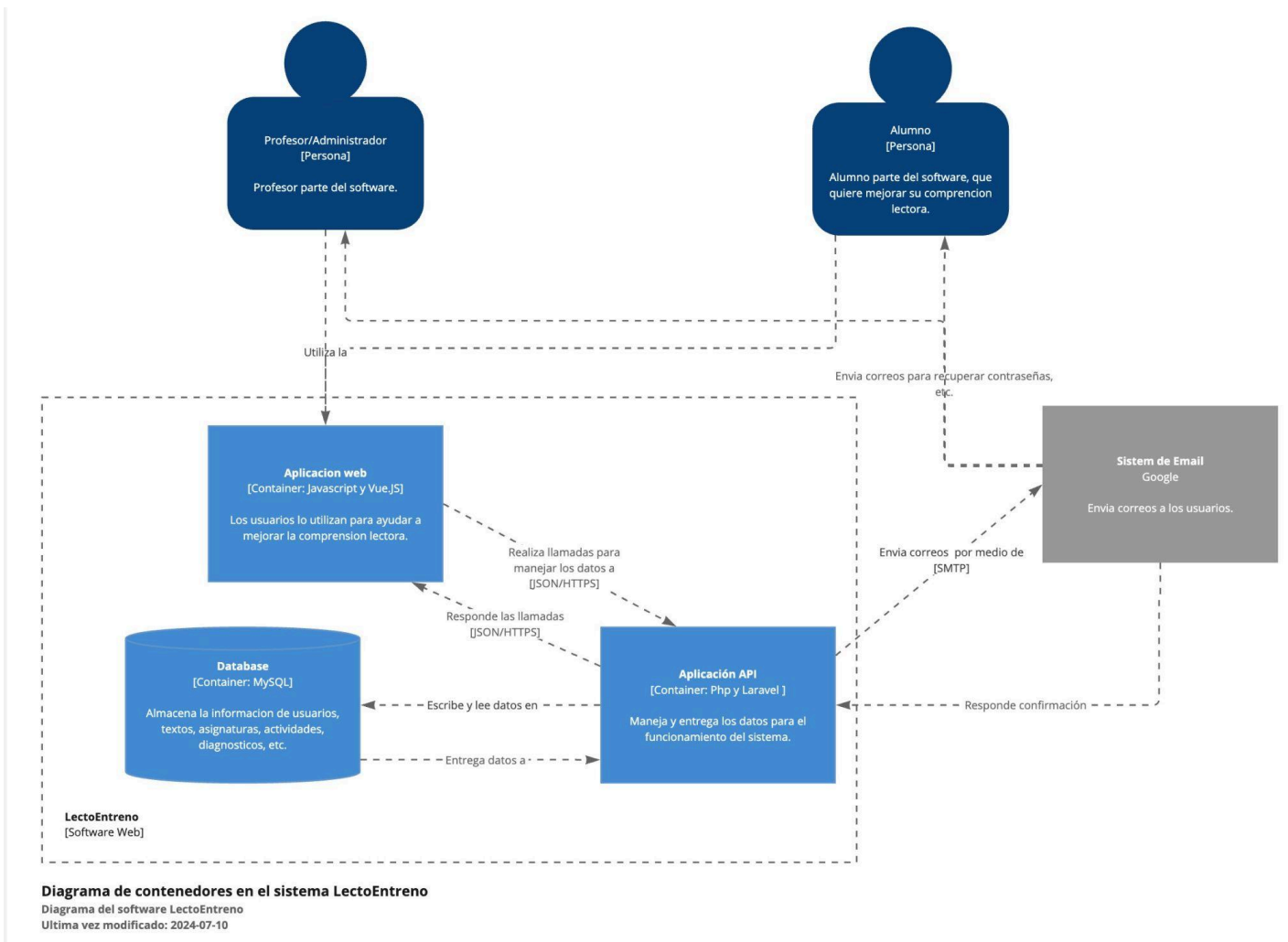


Figura 5: Nivel de contexto, parte del modelo C4 en el software Lecto Entreno

8. Metodología

Para el desarrollo de esta memoria y para asegurar una retroalimentación efectiva, se adoptó una metodología ágil, organizando reuniones semanales con el profesor guía, Geoffrey Hecht. Estos encuentros tenían como finalidad aclarar dudas, revisar los avances realizados y planificar las actividades de la siguiente semana, alineadas con los objetivos específicos del proyecto.

Además, se sostuvieron reuniones regulares con las doctoras Mabel Urrutia y Pamela Guevara, en las cuales se presentaron avances y se discutieron dudas sobre el funcionamiento del sistema, al igual que los errores encontrados. Al final de cada reunión, se redactó una minuta que resumía las funcionalidades discutidas y los cambios sugeridos para el software. Posteriormente, se unieron al grupo los profesores Esteban Pino y Karina Fuente, parte del equipo del proyecto FONDECYT. La inclusión de estos profesores fue muy beneficiosa, ya que estaban familiarizados con el software y proporcionaron valiosa retroalimentación sobre su uso.

Durante el proceso de desarrollo, se generó una copia de la base de datos que se implementó localmente para evitar interrupciones en la operatividad del sistema.

Para el desarrollo del frontend de la aplicación, se realizaron bocetos preliminares que ayudaron a visualizar claramente el desarrollo. Estos bocetos permitían identificar los datos necesarios para cada componente o página y las modificaciones requeridas en la base de datos o las llamadas a la API. Además, antes de implementar cualquier cambio en la base de datos, se revisó el modelo de entidad-relación (MER) actualizado para asegurarse de que los ajustes fueran coherentes con la estructura del sistema.

Además se utilizó una variedad de recursos tecnológicos y plataformas para optimizar el desarrollo del software. Entre estos recursos, se ocupó ChatGPT como apoyo, recurriendo a este modelo de lenguaje avanzado para obtener una opinión a la hora de entender errores de programación, así como para conocer prácticas de diseño.

En el desarrollo del frontend, se utilizó bibliotecas y frameworks modernos para asegurar una interfaz de usuario atractiva. Dentro de la documentación de apoyo, se utilizó tanto Vue.js [4], como Vuetify para realizar componentes estilizados y responsive. Además, para la visualización de datos, se integró vue-chartjs [5], que permitió implementar gráficos dinámicos y comprensibles para los usuarios.

Para aprender a realizar las llamadas y modificaciones en la API, se utilizó la documentación de laravel en la versión 8.

Las tecnologías utilizadas y la documentación técnica fueron de utilidad para asegurar una implementación efectiva y eficiente del proyecto, permitiendo superar desafíos técnicos y la mejora continua del software. A su vez, se mantuvo la misma organización de archivos que en el proyecto anterior, con una estructura de carpetas claramente definida tanto en el frontend como en el backend, lo que facilita la gestión y el mantenimiento del código. Además, se sostuvo la estandarización de las variables en inglés para mantener la consistencia y mejorar la legibilidad del código.

9. Tecnologías Utilizadas:

Backend:

- **PHP con Laravel 8:** Laravel [6] es un framework PHP robusto y popular, utilizado para construir APIs y gestionar la comunicación con bases de datos, en este caso, MySQL. Se mantuvo con estas tecnologías debido a la amplia documentación y la gran comunidad de apoyo que existe, lo que facilita la resolución de problemas y la implementación de nuevas funcionalidades. Laravel 8 proporciona un conjunto de herramientas necesarias para cumplir con los requerimientos del proyecto, como la implementación de rutas, controladores, migraciones, y más, asegurando un desarrollo eficiente.
- **MySQL:** MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado en el desarrollo web. Es conocido por su rendimiento, fiabilidad y facilidad de uso. La elección de MySQL se debe a su compatibilidad con Laravel y su capacidad para manejar grandes volúmenes de datos de manera eficiente.

Frontend:

- **Javascript con VueJS:** Framework de javascript, era una tecnología existente en el código anterior y se mantuvo por su eficiencia en la creación de interfaces dinámicas y su integración con Vuetify, que facilita el diseño responsive y estilizado.
- **Vuex y Vue Router:** Paquetes de vue usados para gestionar el estado de la aplicación y el enrutamiento, respectivamente.
- **SweetAlert2, Vue Chart.js, Vuetify, Vuex, xlsx, axios, Chart.js, y core-js:** Estas bibliotecas proporcionan funcionalidades extendidas para crear alertas, gráficos, manejo de estado, manipulación de datos en Excel, peticiones HTTP y compatibilidad entre navegadores en el frontend.
- Las bibliotecas y paquetes de Vue son tecnologías implementadas en el frontend, tanto para generar Excel, gráficos o componentes como alertas.

Herramientas de Diseño y Pruebas:

- **Bocetos preliminares en tablet³:** Se utilizó una herramienta de dibujo, para realizar un diseño de manera intuitiva y acorde a las necesidades del usuario, permitiendo una visualización clara de las interfaces y componentes, antes de su implementación.
- **Postman:** Postman es una herramienta esencial para probar y desarrollar APIs. Se utilizó Postman para realizar solicitudes HTTP y así verificar respuestas asegurando el correcto funcionamiento de la API, antes de su implementación en el Frontend. Sus características, como la creación de colecciones de pruebas, facilitaron la validación de la API, garantizando que cumplieran con los requisitos.
- **Draw.io:** Software web para generar diagramas, utilizado para diseñar los diagramas de casos de uso y el modelo de entidad relación (MER), facilitando la planificación y comprensión del sistema.
- **Miro:** Software colaborativo utilizado para crear pizarras de trabajo, el cual se utilizó para crear el diagrama C4 de la aplicación. Esta herramienta facilitó el diseño y visualización de la arquitectura del sistema.

Control de Versiones:

- **GitHub:** Esencial para el control de versiones, permitiendo mantener un historial organizado de los cambios.

³ Bocetos : https://drive.google.com/drive/folders/1Ix2J_nH9SQMDNRqH2pdNRjP9tVB_IDrJ?usp=sharing

10. Implementación:

10.1 Vista de Administrador:

Para mejorar la gestión y administración de usuarios en el sistema, se desarrolló un apartado específico en el menú principal que es accesible únicamente para los profesores con el rol de administrador (Ver figura 9). Esta sección, denominada “Administrador”, incluye funcionalidades críticas que no estaban disponibles en la versión anterior de la aplicación (Ver Figura 10).

Acceso y Seguridad: El acceso al menú “Administrador” está bloqueado para los usuarios que no poseen el rol de administrador. Esto se logra mediante un filtro que impide el acceso incluso a través de la URL. Para asegurar la seguridad de las funcionalidades, se generó un middleware de laravel [8] que verifica si el usuario tiene el rol de administrador, restringiendo así el acceso a las llamadas críticas solo a usuarios autorizados.

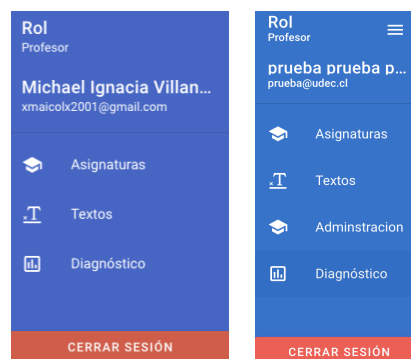


Figura 9: A la izquierda se muestra el sidebar previamente existente. A la derecha el sidebar actual que posee el profesor con rol de Administrador.

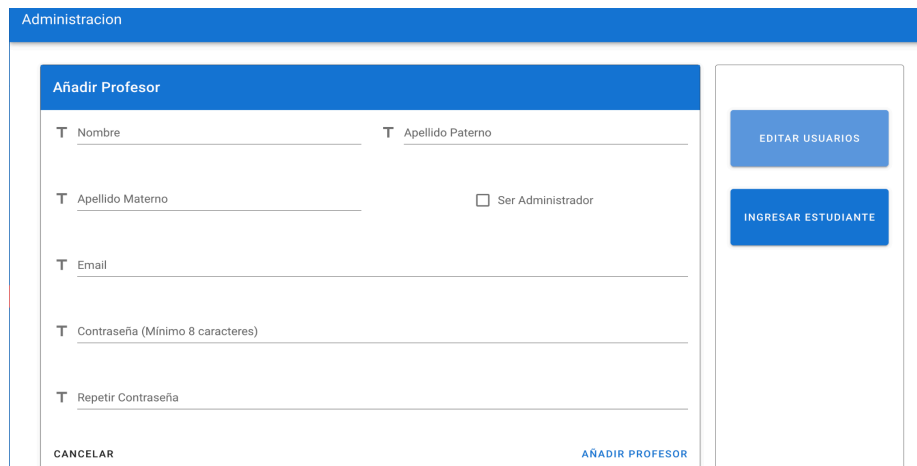
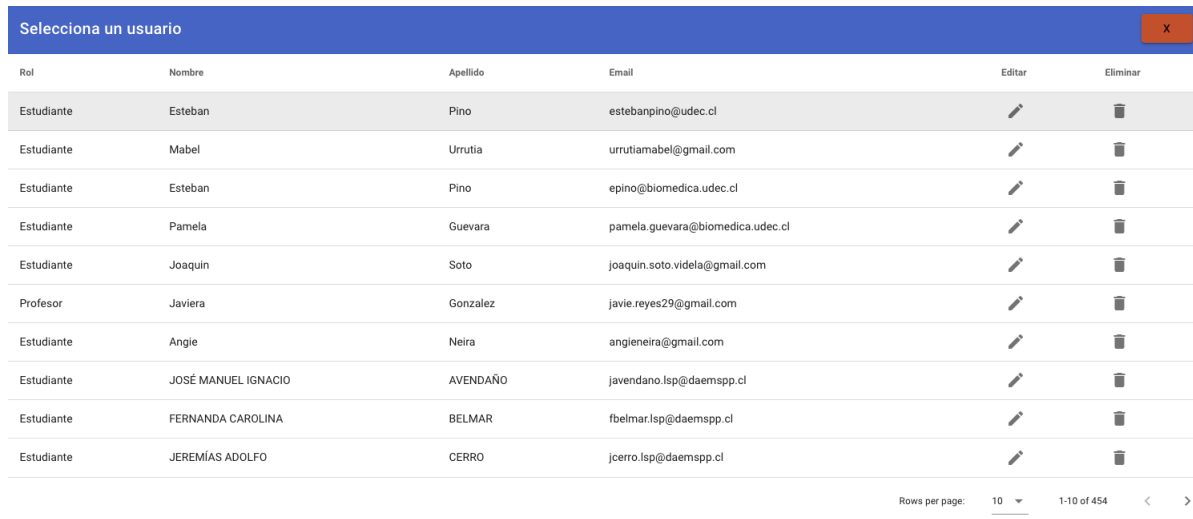


Figura 10: Menú de “Administración” que cuenta con la opción de crear y editar perfiles.

Funcionalidades del Apartado “Administrador”:

1. **Creación de Perfiles de Usuario:** Se desarrolló una funcionalidad para la creación de usuarios con el perfil de docente, una característica que no estaba disponible previamente. También se implementó la opción de crear usuarios con el perfil de estudiante a través de un modal, permitiendo una gestión más centralizada de los usuarios.
2. **Gestión de Usuarios Existentes:** Se creó un modal que permite editar todos los datos de un usuario (Ver figura 11), se muestra una tabla con todos los usuarios existentes, dando la opción de eliminar o editar. A los perfiles de profesor se incluye el rol de administrador como parámetro para editar. Al intentar eliminar o editar un usuario, se solicita confirmación para evitar eliminaciones accidentales.



Rol	Nombre	Apellido	Email	Editar	Eliminar
Estudiante	Esteban	Pino	estebanpino@udec.cl		
Estudiante	Mabel	Urrutia	urrutiamabel@gmail.com		
Estudiante	Esteban	Pino	epino@biomedica.udec.cl		
Estudiante	Pamela	Guevara	pamela.guevara@biomedica.udec.cl		
Estudiante	Joaquin	Soto	joaquin.soto.videla@gmail.com		
Profesor	Javiera	Gonzalez	javie.reyes29@gmail.com		
Estudiante	Angie	Neira	angieneira@gmail.com		
Estudiante	JOSÉ MANUEL IGNACIO	AVENDAÑO	javendano.lsp@daemssp.cl		
Estudiante	FERNANDA CAROLINA	BELMAR	fbelmar.lsp@daemssp.cl		
Estudiante	JEREMÍAS ADOLFO	CERRO	jcerro.lsp@daemssp.cl		

Rows per page: 10 1-10 of 454

Figura 11: Modal de edición, a la derecha de la tabla se encuentran las opciones de edición y eliminar.

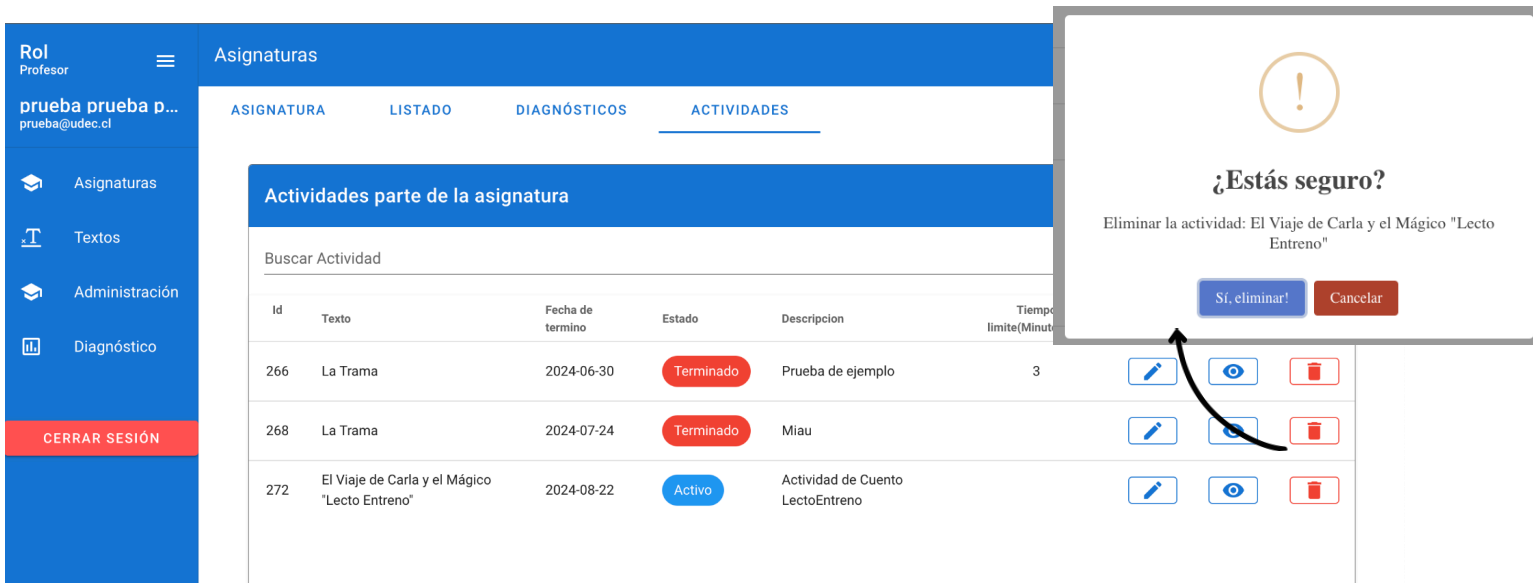
Modificaciones en el Backend: Se añadió una nueva propiedad a la tabla usuario en la base de datos para integrar el permiso de administrador y a su vez se crearon nuevas llamadas en la API para permitir la creación de perfiles con el rol de profesor para la edición y eliminación de usuarios de manera segura y eficiente, restringiendo el acceso a las llamadas críticas solo a usuarios autorizados.

Estas mejoras en la vista de administrador no solo facilitan la gestión de usuarios, sino que también aseguran que solo los profesores con los permisos adecuados puedan realizar cambios críticos en el sistema, mejorando la seguridad y la eficiencia operativa del software.

10.2. Panel de control de Actividades:

El panel de control de actividades se integró dentro del menú de una asignatura (Ver Figura 12) en la pestaña “Actividades”, permitiendo la gestión completa de las actividades relacionadas con dicha asignatura. Esta integración incluye funcionalidades para crear, editar, ver datos y eliminar actividades. El diseño del panel se inspiró en experiencias previas, como mi práctica profesional y el software "Canvas". Este enfoque facilita la comprensión de los datos por parte de las profesoras, familiarizadas con dicho software.

Para el diseño de la vista, se utilizaron bocetos preliminares, permitiendo mayor libertad al tomar notas sobre los datos que debían mostrarse y obtenerse de la base de datos.



The screenshot shows a web interface for a course. On the left is a blue sidebar with navigation options: Asignaturas, Textos, Administración, Diagnóstico, and a CERRAR SESIÓN button. The main content area is titled 'Asignaturas' and has tabs for ASIGNATURA, LISTADO, DIAGNÓSTICOS, and ACTIVIDADES. Under the 'ACTIVIDADES' tab, there is a search bar 'Buscar Actividad' and a table of activities. The table has columns for Id, Texto, Fecha de termino, Estado, Descripción, and Tiempo limite(Minutos). Three activities are listed: ID 266 (La Trama, Terminado, 3 minutes), ID 268 (La Trama, Terminado), and ID 272 (El Viaje de Carla y el Mágico "Lecto Entreno", Activo). Each row has edit, view, and delete icons. A modal dialog is open over the delete icon of the third activity, asking '¿Estás seguro?' and showing the text 'Eliminar la actividad: El Viaje de Carla y el Mágico "Lecto Entreno"'. The modal has 'Sí, eliminar!' and 'Cancelar' buttons. An arrow points from the delete icon to the modal.

Id	Texto	Fecha de termino	Estado	Descripción	Tiempo limite(Minutos)
266	La Trama	2024-06-30	Terminado	Prueba de ejemplo	3
268	La Trama	2024-07-24	Terminado	Miau	
272	El Viaje de Carla y el Mágico "Lecto Entreno"	2024-08-22	Activo	Actividad de Cuento LectoEntreno	

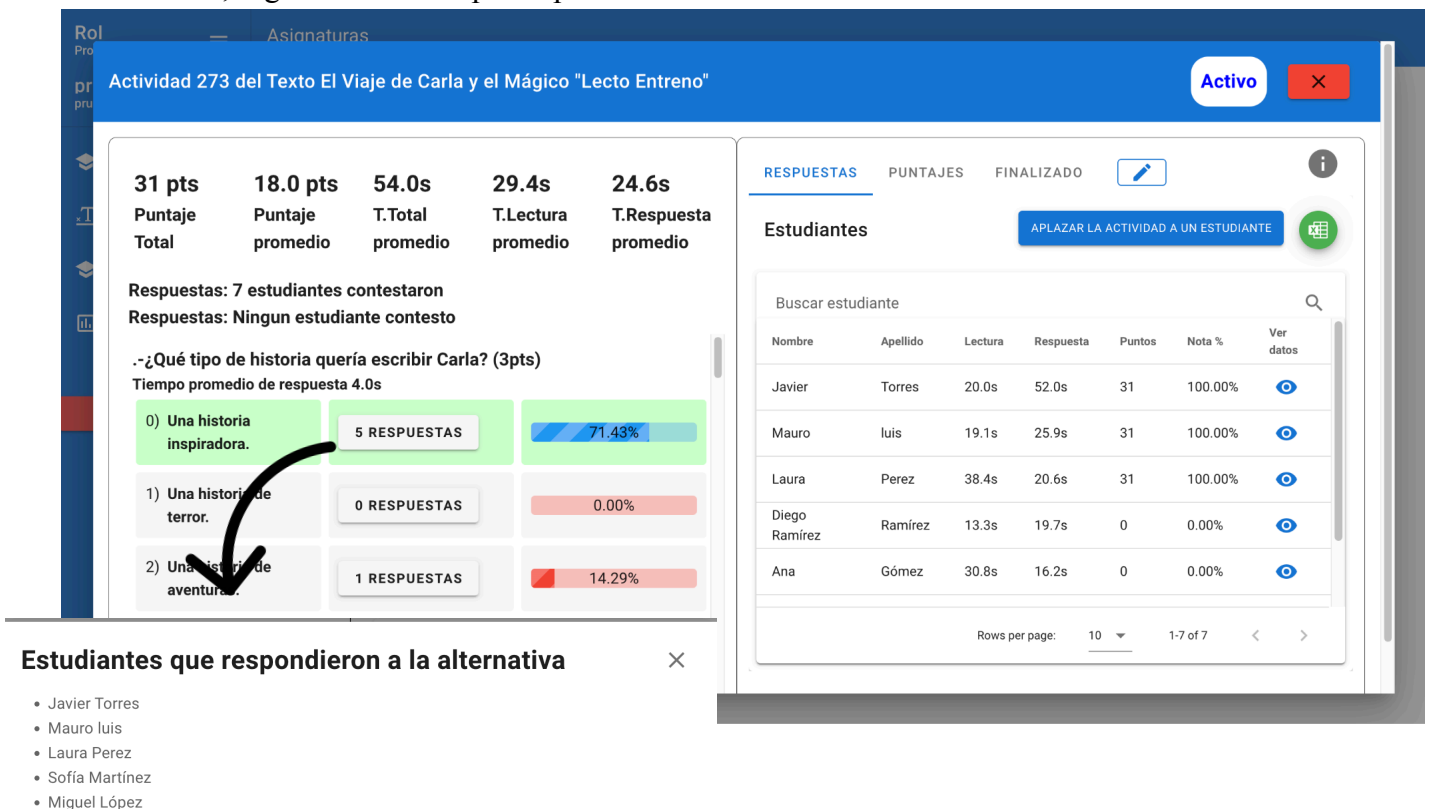
Figura 12: Sección de actividades dentro del menú de asignatura y modal de confirmación al eliminar una actividad.

Dentro de la tabla en la sección de actividades se pueden observar todas las actividades creadas en dicha asignatura, mostrando datos relevantes a primera vista. A su vez, se puede buscar una actividad por medio de la descripción o ID.

Las acciones que se pueden realizar tienen alertas de errores y confirmaciones para evitar modificaciones accidentales. En el dashboard de datos de una actividad (Ver figura 13), el profesor puede ver el título de la actividad, el texto utilizado y su estado (activo o terminado). La información dentro del dashboard está dividida en:

Parte Superior Izquierda: Se muestra un resumen de los puntajes y tiempos promedio de lectura y respuesta, calculados a partir de todos los alumnos que han respondido la actividad.

Parte Inferior: Se presentan los resultados generales de la actividad, con detalles de cada pregunta y sus respuestas por parte de los estudiantes. Al hacer clic en "X RESPUESTAS", se muestran los estudiantes que respondieron esta opción, a la derecha de cada alternativa igual se muestra un porcentaje mediante un gráfico de barra lateral, que representa la cantidad de estudiantes que respondieron dicha alternativa frente otras, de esta manera los profesores pueden saber rápidamente cuál fue la alternativa que más respuestas obtuvo. Para las preguntas de desarrollo, se muestra el nombre y la respuesta del estudiante, este tipo de preguntas no se contabilizan para el puntaje total, ya que solo se consideran las preguntas con respuestas definidas, según lo indicado por el profesor.



Actividad 273 del Texto El Viaje de Carla y el Mágico "Lecto Entreno" Activo

31 pts Puntaje Total | 18.0 pts Puntaje promedio | 54.0s T.Total promedio | 29.4s T.Lectura promedio | 24.6s T.Respuesta promedio

Respuestas: 7 estudiantes contestaron
Respuestas: Ningun estudiante contesto

.-¿Qué tipo de historia quería escribir Carla? (3pts)
Tiempo promedio de respuesta 4.0s

Alternativa	Respuestas	Porcentaje
0) Una historia inspiradora.	5 RESPUESTAS	71.43%
1) Una historia de terror.	0 RESPUESTAS	0.00%
2) Una historia de aventuras.	1 RESPUESTAS	14.29%

Estudiantes que respondieron a la alternativa

- Javier Torres
- Mauro luis
- Laura Perez
- Sofía Martínez
- Miguel López

Nombre	Apellido	Lectura	Respuesta	Puntos	Nota %	Ver datos
Javier	Torres	20.0s	52.0s	31	100.00%	
Mauro	luis	19.1s	25.9s	31	100.00%	
Laura	Perez	38.4s	20.6s	31	100.00%	
Diego	Ramírez	13.3s	19.7s	0	0.00%	
Ana	Gómez	30.8s	16.2s	0	0.00%	

Figura 13 : Dashboard de datos de una actividad, mostrando el modal de respuestas a una alternativa.

Parte Derecha: Aquí se incorporaron 3 secciones para integrar más información, además de un botón para editar la asignatura.

1. **Respuestas:** Sección predeterminada que muestra una tabla con todas las respuestas a la actividad hasta el momento, con información sobre tiempos, puntajes de actividad

realizada por el estudiante y la opción de ver los datos de esta. Esta sección incluye la funcionalidad para flexibilizar la fecha límite para un estudiante específico mediante el botón de “APLAZAR LA ACTIVIDAD PARA UN ESTUDIANTE”, así como la opción de descargar los datos en un Excel el cual muestra un resumen de la actividad, todas las respuestas de cada estudiante por pregunta, al igual que una tabla resumen con los puntajes y tiempos de cada estudiante.

Al ver los datos de una respuesta (Ver figura 14) se muestra todo lo recopilado en la actividad del estudiante, esto por medio de un modal, donde en su primera sesión “Párrafos” se muestra el texto leído por el estudiante, preguntas si es que se realizaron y los resultados del minijuego incluido, donde el color verde indica que la respuesta es correcta y el color rojo errónea.



Asignaturas

Actividad

31 pts
Puntaje
Total

respues
respues

¿Qué t
Tiempo p

0) Una
insp

1) Una
terr

2) Una
aver

3) Una

Asignaturas

Detalles de la Actividad de Javier Torres

PÁRRAFOS RESPUESTAS PALABRAS GUARDADAS

Tiempo de lectura: 20.02 segundos

Había una vez una niña llamada Carla que vivía en un pequeño pueblo al pie de las montañas. A Carla le encantaba jugar con sus amigos y escuchar las historias que su abuela le contaba cada noche. Pero había algo que la preocupaba: a Carla le costaba mucho entender las palabras cuando leía. Un día, su maestro le habló de un software mágico llamado "Lecto Entreno". Este programa era especial porque ayudaba a los niños a comprender mejor las palabras y a disfrutar de la lectura.

× Estudiante no realizó pregunta.

Cuando Carla abrió "Lecto Entreno" por primera vez, se sintió un poco nerviosa. El programa le presentó varios juegos y actividades que parecían difíciles. Sin embargo, el software estaba diseñado de una manera muy amigable. Carla comenzó con juegos simples, donde tenía que unir palabras con imágenes, y poco a poco fue ganando confianza. Cada vez que completaba una actividad, el programa la felicitaba con una

Figura 14 :Detalles de la actividad de un estudiante, sección de “Párrafos” .

En la siguiente sección (Ver Figura 15) se visualiza el apartado de preguntas de la actividad, donde se muestran las alternativas correctas e incorrectas que contestó el estudiante y los puntajes obtenidos por pregunta. Dentro del apartado del tiempo de una actividad, se entrega el tiempo de lectura y de respuesta, así como también el tiempo específico que se demoró en contestar cada pregunta. En la última sección “Palabras

guardadas” se muestran tanto las palabras guardadas como subrayadas del texto leído por el estudiante al realizar la actividad.



Detalles de la Actividad de Javier Torres

PÁRRAFOS RESPUESTAS PALABRAS GUARDADAS

Tiempo de respuesta: 51.98 segundos

1) ¿Qué tipo de historia quería escribir Carla? (3/3pts)

Tiempo que se demora en la pregunta : 7.76 segundos

Una historia inspiradora. (Respuesta correcta)

Una historia de terror.

Una historia de aventuras.

Una historia cómica.

2) ¿Cómo se sintió Carla cuando su historia fue publicada? (3/3pts)

Tiempo que se demora en la pregunta : 4.27 segundos

Figura 15 :Detalles de la actividad de un estudiante, sección de “Respuestas” .

2. **Puntajes:** La sección de Puntajes (Ver figura 16) muestra un gráfico resumen con los porcentajes obtenidos en la actividad, indicando la cantidad de estudiantes en cada rango de porcentaje. Al hacer click en una barra, se enlista a los estudiantes que obtuvieron ese porcentaje de nota.



Figura 16 : Sección de Puntajes, mostrando lista de estudiantes en ese porcentaje de nota.

3. **Finalizado:** La sección (Ver figura 17) muestra un gráfico en forma de dona que indica los estudiantes que terminaron la actividad y los que no. Los estudiantes que terminaron se representan con el color azul y los que no con el color rojo. Al hacer click en el gráfico, se muestra un listado de los estudiantes que han terminado o no la actividad con su respectivo puntaje y porcentaje.



Figura 17 : A la derecha Sección de Finalizado, mostrando lista de estudiantes que terminaron la actividad.

Modificaciones en el Backend: se realizaron migraciones con PHP para agregar nuevas propiedades a la base de datos. Se añadió la columna 'student_postponed' a la tabla de actividades para manejar los aplazamientos, y la columna 'timeAnswer' a la tabla de respuestas para guardar el tiempo de respuesta de cada estudiante.

Además, se modificaron y crearon nuevas llamadas a la API para obtener datos tanto de completitud de una actividad como de información específica de cada actividad, mejorando así los datos entregados al profesor para un análisis más detallado. Estas llamadas incluyen medidas de seguridad y validación mediante middleware y tokens JWT.

10.3. Función para que los profesores compartan asignaturas:

La nueva funcionalidad permite a los profesores compartir una asignatura con otros educadores, posibilitando que más de un profesor pueda crear actividades y ver los datos de toda la asignatura. Esta función se implementó mediante un botón en la sección principal del menú de la asignatura. Al hacer click en este botón, se abre un modal donde se muestran todos los perfiles de profesores creados en el sistema (Ver figura 18). Se puede seleccionar uno o más profesores para compartir la asignatura y se pide confirmación antes de completar el proceso.

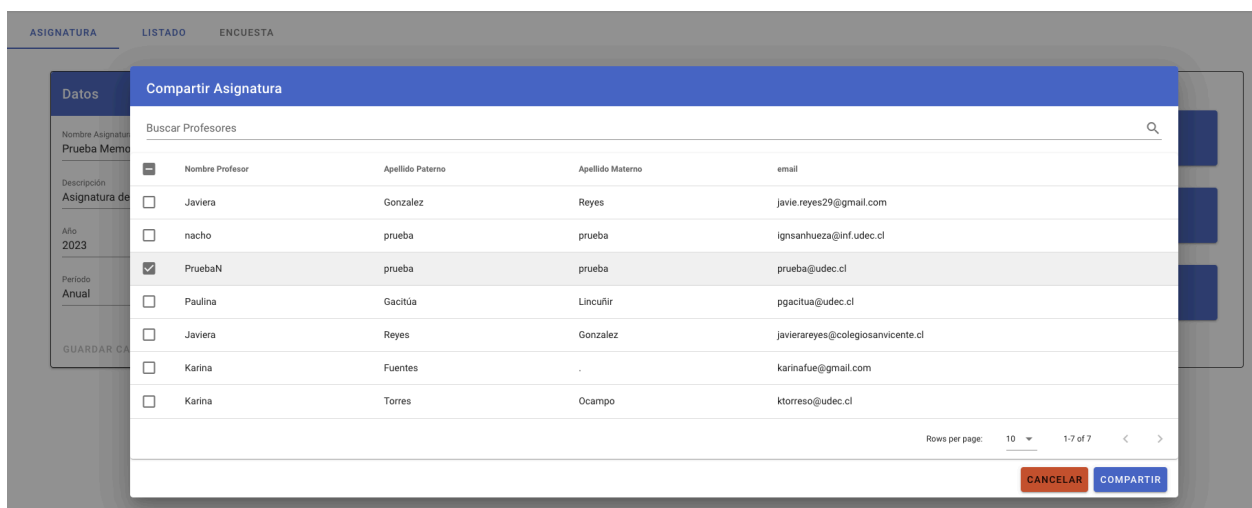


Figura 18: Modal para compartir una asignatura.

Modificaciones en el Backend: se creó una nueva tabla en la base de datos para gestionar la relación de muchos a muchos entre las tablas usuarios (con el rol de profesor) y asignaturas. Esta tabla almacena los registros de las asignaturas compartidas entre profesores. La estructura de esta tabla incluye referencias a las claves primarias de las tablas usuarios y asignaturas, estableciendo una relación directa entre ellas.

Se desarrolló una nueva llamada a la API que permite compartir una asignatura con los profesores seleccionados, esta llamada se encarga de registrar en la tabla relacional las asociaciones correspondientes, asegurando que los profesores seleccionados tengan acceso a las actividades y datos de la asignatura compartida.

Esta funcionalidad no solo facilita la colaboración entre profesores, sino que también optimiza la utilización del sistema.

10.4. Mejoramiento del proceso de actividades para que sean más flexibles:

Para hacer el proceso de actividades más flexible, se añadieron nuevas funcionalidades que permiten a los profesores ajustar las actividades según las necesidades individuales de los estudiantes y de la clase en general.

Funcionalidades Implementadas:

1. Aplazamiento de actividades para estudiantes individuales:

- Se añadió la capacidad de aplazar la fecha límite de una actividad para un estudiante en particular. Esta funcionalidad incluye un formulario donde el profesor puede ingresar una descripción de la justificación y la nueva fecha límite para el estudiante seleccionado.
- Esta opción se encuentra integrada por medio de un botón en el dashboard de una actividad, el cual abre un modal proporcionando el formulario para aplazar la actividad a un estudiante en específico (Ver figura 19).

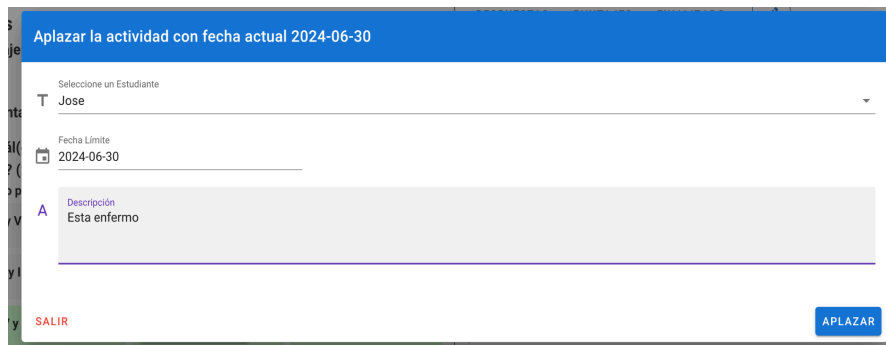


Figura 19: Modal para aplazar la actividad a un estudiante.

2. Edición de actividades:

- Se implementó un botón que permite a los profesores editar las actividades existentes. Los cambios incluyen la modificación de variables generales, como el tiempo límite de la actividad, aplicable a toda la asignatura.
- Estas modificaciones aseguran que los profesores puedan ajustar las actividades en respuesta a cambios en el calendario académico o situaciones imprevistas.

Modificaciones en el Backend:

- Dentro de la base de datos se añadió la columna `student_postponed` a la tabla de actividades, la cual almacena toda la información relacionada con el aplazamiento de actividades para estudiantes específicos.
- Cuando se guarda la información de aplazamiento, se actualiza esta columna con los nuevos datos relevantes, permitiendo así gestionar adecuadamente las nuevas fechas límite para los estudiantes.
- Se crearon llamadas en la API para la edición de la actividad y para aplazar a un estudiante en específico, donde el aplazamiento se agrega a los otros existentes en una actividad. Si el alumno ya existía en `student_postponed` se actualiza solamente la fecha límite.

Actualización de la vista del estudiante:

- Verificación de Aplazamientos: Al mostrar las actividades pendientes para un estudiante, el sistema verifica primero si la propiedad `student_postponed` es nula. Si es así, se muestra la fecha límite normal. Si contiene datos, el sistema verifica si el usuario está en la lista de aplazamientos.
 - **Para estudiantes aplazados:** Si el estudiante está en la lista, se muestra la nueva fecha límite especificada en el aplazamiento.
 - **Para estudiantes no aplazados:** Si el estudiante no está en la lista, se muestra la fecha límite original de la actividad.

Estas mejoras aseguran que los profesores tengan las herramientas necesarias para gestionar actividades de manera flexible y personalizada, optimizando la experiencia educativa tanto para ellos como para los estudiantes.

10.5. Nuevas estrategias implementadas:

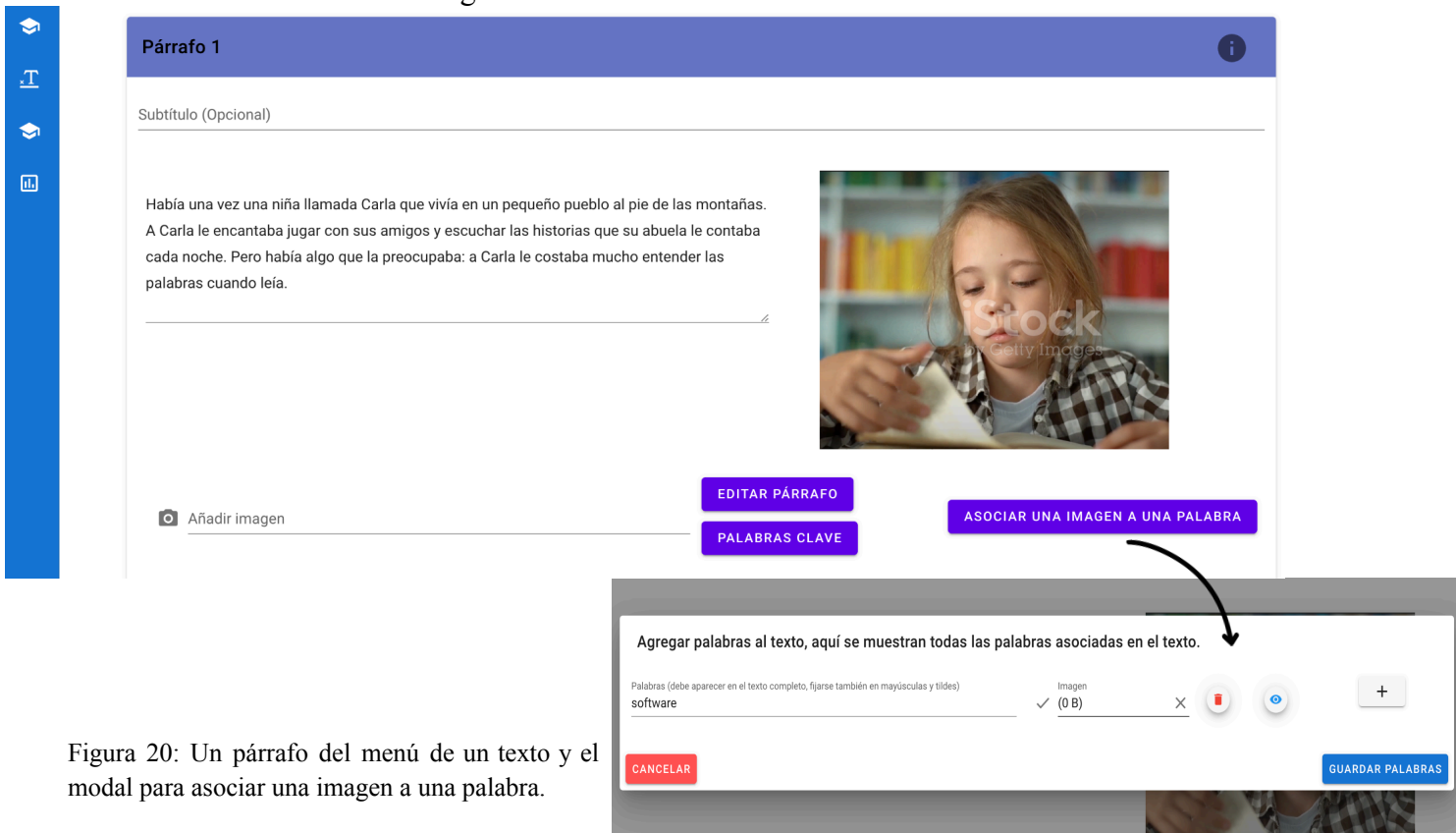
Imagen Relacionada a una Palabra:

Esta estrategia fue propuesta por las profesoras co-patrocinantes. Se basa en la idea de que los estudiantes entienden mejor un texto si las palabras difíciles están acompañadas de una imagen que las represente, facilitando así la comprensión del texto y el significado de las palabras.

Implementación:

1. Componente de asociación de palabras e imágenes:

- Se generó un componente en el menú de un texto (Ver figura 20) junto a las otras opciones disponibles. Este componente permite a los profesores ingresar una palabra dentro del texto y subir una imagen para asociarla (Ver figura 20).
- Se muestra una lista de todas las palabras con imágenes en el texto, permitiendo agregar, ver la imagen y eliminar las asociaciones cuando sea necesario.
- La palabra solo se puede guardar si está dentro del texto y se permiten todos los formatos de imagen.

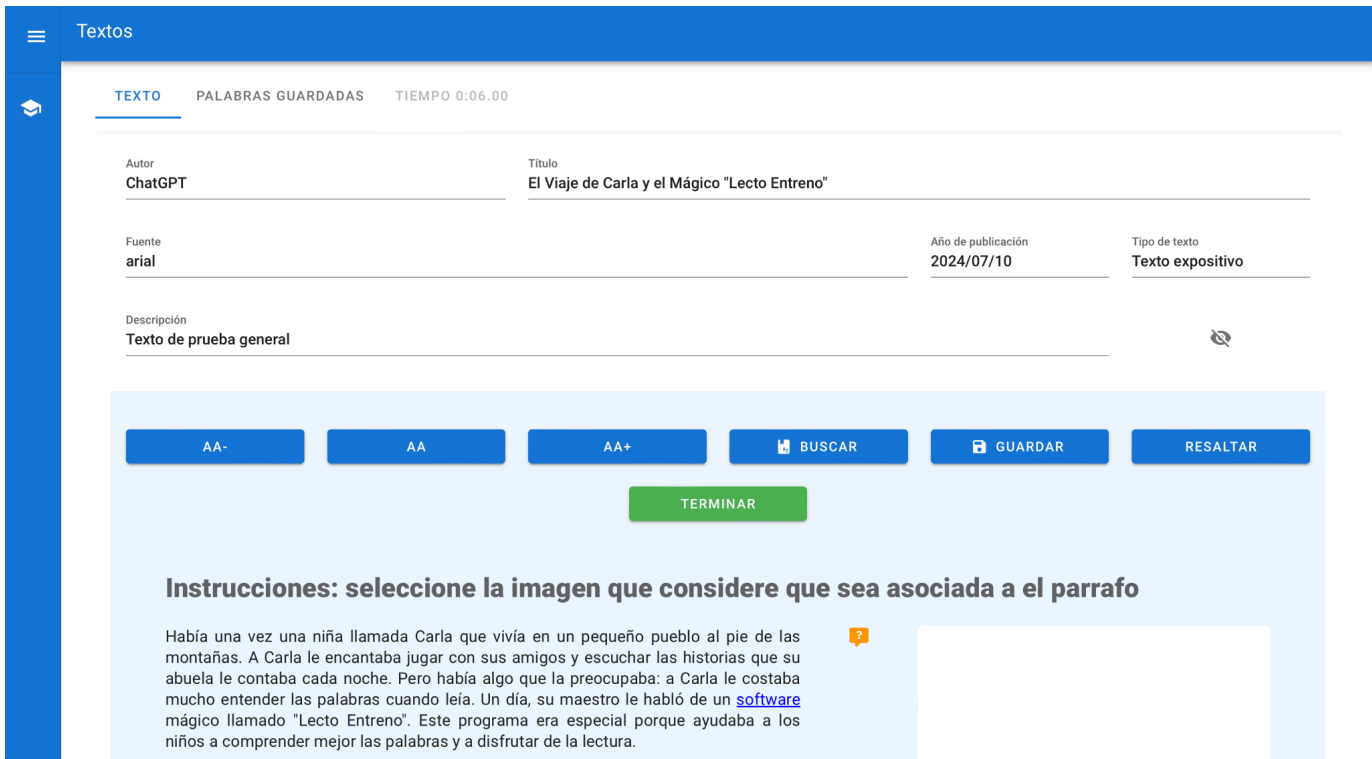


The screenshot displays a text editor interface. On the left, a vertical blue sidebar contains icons for home, text, edit, and image. The main editor area shows a paragraph titled "Párrafo 1" with the following text: "Había una vez una niña llamada Carla que vivía en un pequeño pueblo al pie de las montañas. A Carla le encantaba jugar con sus amigos y escuchar las historias que su abuela le contaba cada noche. Pero había algo que la preocupaba: a Carla le costaba mucho entender las palabras cuando leía." To the right of the text is an image of a young girl reading a book. Below the text, there are three buttons: "EDITAR PÁRRAFO", "PALABRAS CLAVE", and "ASOCIAR UNA IMAGEN A UNA PALABRA". An arrow points from the "ASOCIAR UNA IMAGEN A UNA PALABRA" button to a modal window. The modal window has the title "Agregar palabras al texto, aquí se muestran todas las palabras asociadas en el texto." It contains a text input field with "software" and a checkbox for "Imagen" which is checked. Below the input field are "CANCELAR" and "GUARDAR PALABRAS" buttons.

Figura 20: Un párrafo del menú de un texto y el modal para asociar una imagen a una palabra.

2. Interfaz para Estudiantes:

- En este apartado, cuando el estudiante realiza una actividad (Ver figura 21) en la sección de lectura, las palabras con imágenes asociadas se muestran como hipervínculos de color azul.
- Al hacer clic en la palabra, se abre un modal con la imagen correspondiente.



Textos

TEXTO PALABRAS GUARDADAS TIEMPO 0:06.00

Autor: ChatGPT Título: El Viaje de Carla y el Mágico "Lecto Entreno"

Fuente: arial Año de publicación: 2024/07/10 Tipo de texto: Texto expositivo

Descripción: Texto de prueba general

AA- AA AA+ BUSCAR GUARDAR RESALTAR TERMINAR

Instrucciones: seleccione la imagen que considere que sea asociada a el parrafo

Había una vez una niña llamada Carla que vivía en un pequeño pueblo al pie de las montañas. A Carla le encantaba jugar con sus amigos y escuchar las historias que su abuela le contaba cada noche. Pero había algo que la preocupaba: a Carla le costaba mucho entender las palabras cuando leía. Un día, su maestro le habló de un [software](#) mágico llamado "Lecto Entreno". Este programa era especial porque ayudaba a los niños a comprender mejor las palabras y a disfrutar de la lectura.

Figura 21: Actividad de un estudiante en la sección de lectura, mostrando la imagen relacionada a la palabra.

3. Modificaciones en el Backend:

- En la base de datos se añadió la columna `images_words` a la tabla de textos, que guarda la información del nombre de cada imagen y su palabra asociada en formato JSON. De esta manera al entregar los datos al usuario, se le da el enlace a la ubicación, con el nombre de la imagen.
- Las imágenes se almacenan en el servidor del hosting, mediante el storage de php [9], de este modo el nombre se guarda solamente en la base de datos y al momento de entregar la información se da la URL o dirección de la imagen al frontend, lo que facilita su acceso y mantiene la eficiencia del sistema.

- Se crearon nuevas llamadas en la API para guardar y obtener la información de las imágenes. En caso de eliminación de la asociación la imagen igual se elimina del servidor.
- Se modificaron las llamadas de obtención de un texto en la API para incluir las imágenes correspondientes.

Minijuego de las Imágenes:

Esta estrategia, fue una propuesta propia y bien recibida por parte de los profesores, la cual se basa en la relación entre un párrafo del texto y una imagen. El estudiante debe seleccionar una de las imágenes, incorporadas por el profesor al texto, que represente mejor el párrafo leído. Esta función se incorpora como un minijuego dentro de la lectura en una actividad, similar al minijuego de ordenar párrafos ya existente.

Implementación:

1. Componente de subida de imágenes para párrafos:

- En el menú de un texto, se añadió un input en cada párrafo para subir una imagen relacionada. Al subir la imagen, esta se muestra a la derecha del párrafo (Ver figura 22).

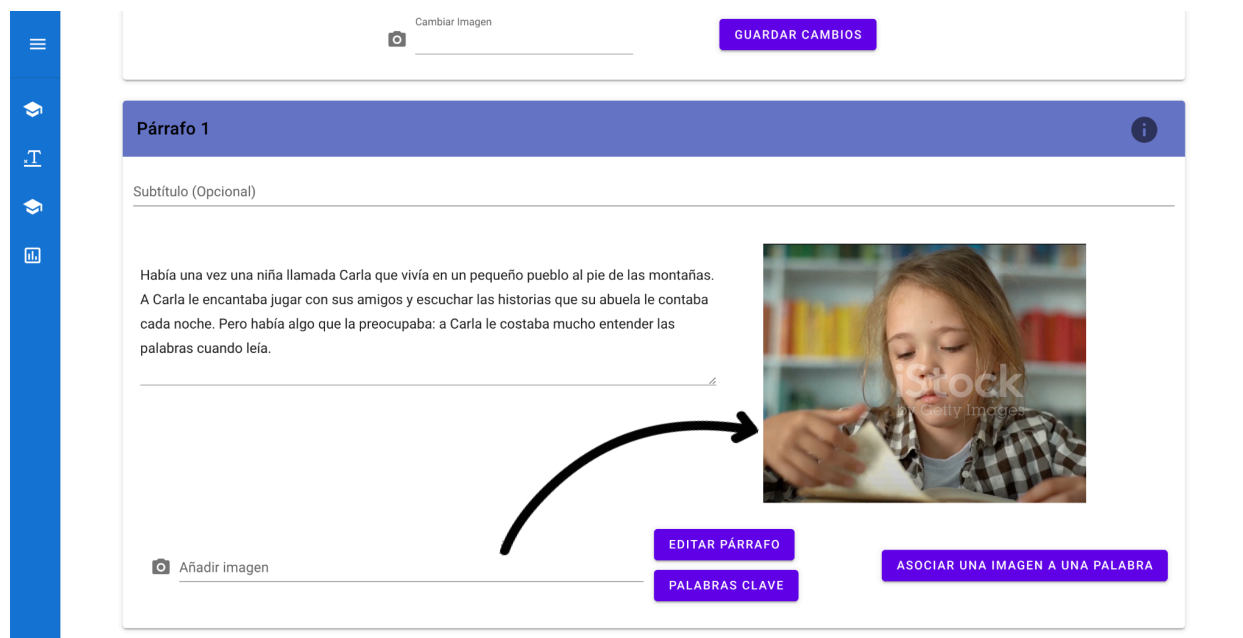


Figura 22: Un párrafo del menú de un texto indicando el input para subir una imagen y su resultado al subirla.

2. Interfaz para Estudiantes:

- Durante la lectura, se presenta un carrusel de imágenes a la derecha de cada párrafo (Ver imagen 23). Los estudiantes pueden seleccionar una imagen o elegir la opción "Ninguna Imagen" si consideran que ninguna corresponde al párrafo leído.
- Al finalizar la actividad, se proporciona retroalimentación sobre las respuestas del minijuego, indicando si las selecciones fueron correctas o incorrectas.



Figura 23: Actividad con el minijuego de ordenar imágenes incluido, en la sección de lectura.

3. Modificaciones en el Backend:

- En la base de datos se añadió la columna minigame a la tabla de actividades, que guarda el tipo de minijuego seleccionado al crear la actividad. Por su parte, en la tabla de respuestas se añadió la columna minigame_info para almacenar la información de las respuestas de los estudiantes en el minijuego.
- Se crearon y modificaron llamadas en la API para gestionar las imágenes de los párrafos y la información del minijuego, las imágenes se almacenan en el servidor y solamente se guarda el nombre en la base de datos para agilizar las llamadas y que así sean más eficientes.
- Las llamadas para obtener la información de un texto se actualizaron para incluir las imágenes correspondientes con el minijuego.

Estas estrategias implementadas mejoran la interacción de los estudiantes con el texto y ofrecen nuevas herramientas a los profesores para evaluar y mejorar la comprensión lectora.

10.6. Creación de diagnósticos y muestra de datos:

Dentro del Sidebar principal del profesor, se utilizó el antiguo apartado de los diagnósticos. En la sección del sidebar de “Diagnósticos”, que muestra todos los diagnósticos, se agregó una pestaña para añadir nuevos diagnósticos (Ver figura 24).

Una solicitud de los profesores era que los diagnósticos tuvieran más de un tipo de respuestas, como por ejemplo que pudieran colocar una pregunta con una escala de respuestas y otra con una escala diferente, ya que antiguamente los diagnósticos solamente tenían un tipo de respuestas globales para todas las preguntas de alternativas a las que se le asignaba un puntaje, las otras opciones, preguntas de “Si y No” o de desarrollo no tenían puntaje. Para poder mostrar varias preguntas con diferentes alternativas, se crearon secciones las cuales contendrán un grupo de preguntas asociadas, que pueden ser de tres tipos:

1. **Preguntas de Respuestas Globales:** Estas son preguntas tipo encuesta con respuestas como por ejemplo: "Muy mal", "Mal", "Bien" y "Muy bien", respuestas que serán para todas las preguntas de este tipo en la sección y tendrán asociado un puntaje.
2. **Preguntas de Sí o No:** Preguntas que se responden con un simple "Sí" o "No", que ahora tendrá asociado un puntaje que será global para todas las preguntas de la sección.
3. **Preguntas de Desarrollo:** Preguntas abiertas donde se pueden escribir respuestas largas.

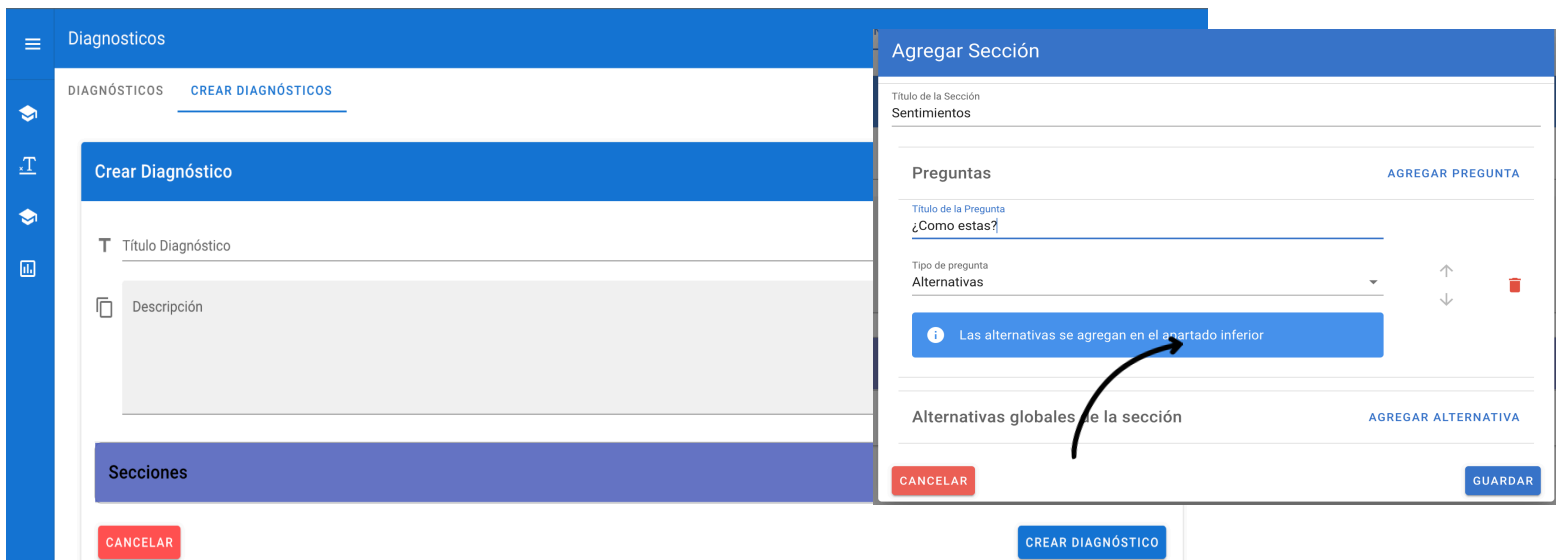


Figura 24 : Sección para crear diagnósticos, integrando una sección.

Estos diagnósticos pueden ser utilizados en más de una asignatura a través de una actividad de diagnóstico, lo que justifica su manejo en un menú aparte para su creación, a diferencia de las actividades normales creadas directamente en una asignatura.

Adicionalmente, y a petición de los docentes, en el apartado de diagnósticos ahora se muestra a cualquiera de los profesores la totalidad de diagnóstico creados en el software, para que todos puedan ocupar cada uno de los diagnósticos y no tengan que crearlos nuevamente. Adicionalmente, se agregó un botón para eliminar un diagnóstico pero con la condición de que seas el profesor creador del mismo, por lo que si no creaste el diagnóstico saldrá una alerta avisando que no puedes eliminarlo por falta de autorización (Ver figura 25).

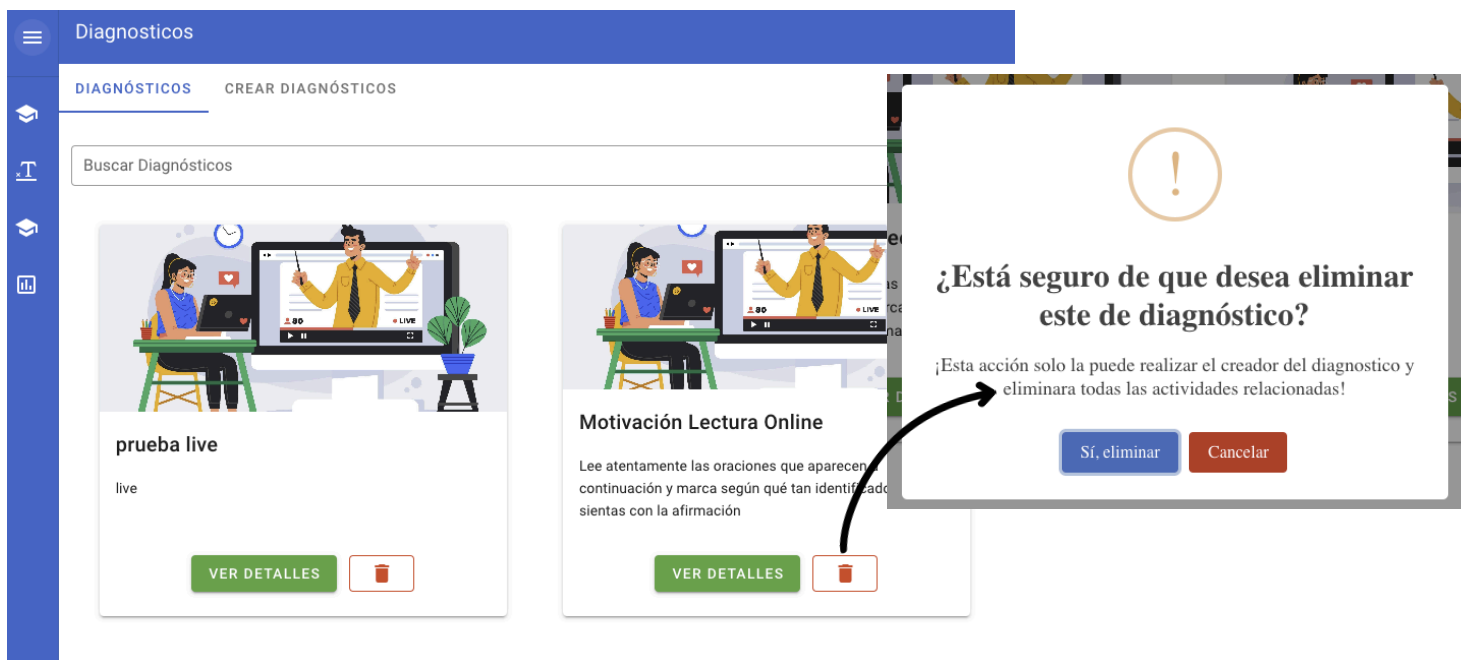
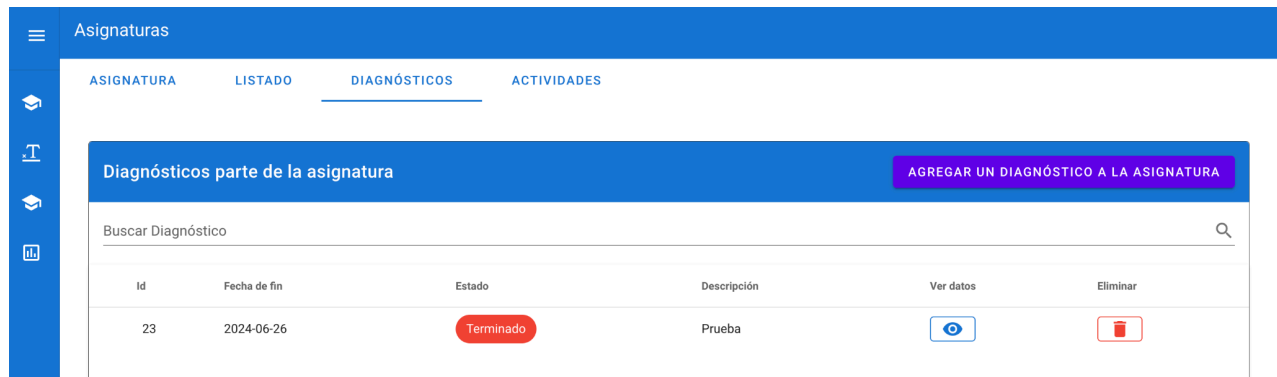


Figura 25 : Menú o apartado de diagnósticos, mostrando la confirmación de eliminación.

Incorporación de Lecturas: Se añadió la funcionalidad para integrar la lectura de un texto a la actividad de diagnóstico, una solicitud por parte de las profesoras para mejorar el contexto de las encuestas. Esta funcionalidad se integró al momento de crear la actividad de diagnóstico.

Muestra de datos: En el menú de una asignatura se muestra una tabla con todas las actividades de diagnósticos creadas en la asignatura (Ver figura 26). Para mantener la coherencia del software enfocado en los datos recopilados, se creó un apartado para ver los datos de estas actividades. Este panel es más sencillo en comparación con los de las actividades normales, mostrando principalmente datos generales del diagnóstico.



Asignaturas

ASIGNATURA LISTADO **DIAGNÓSTICOS** ACTIVIDADES

Diagnósticos parte de la asignatura AGREGAR UN DIAGNÓSTICO A LA ASIGNATURA

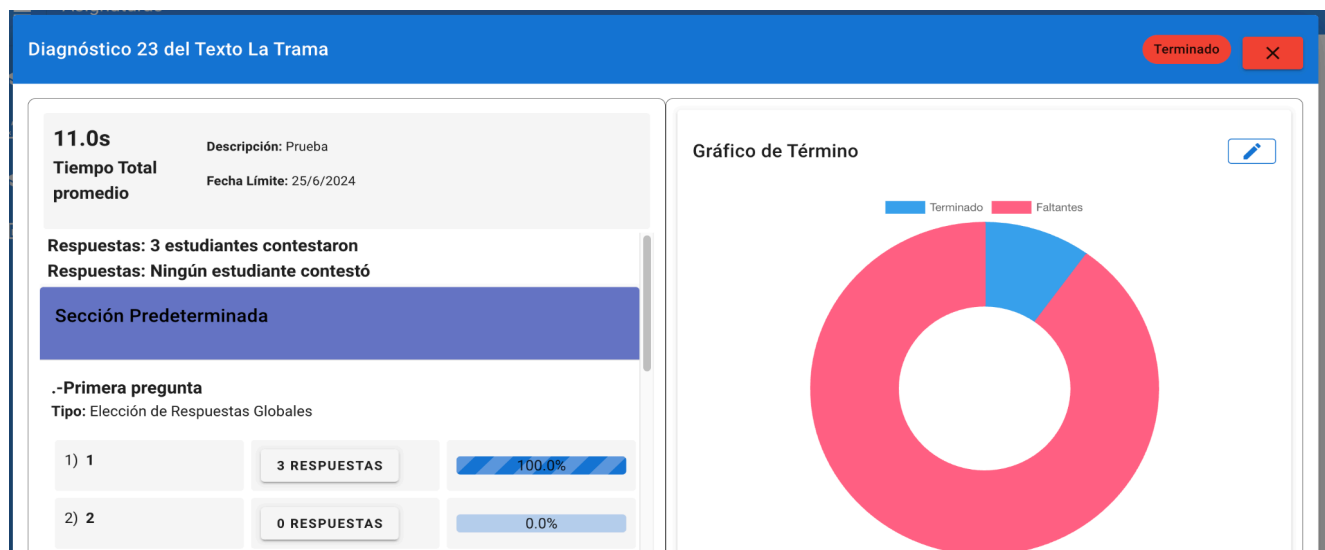
Buscar Diagnóstico

Id	Fecha de fin	Estado	Descripción	Ver datos	Eliminar
23	2024-06-26	Terminado	Prueba		

Figura 26: Sección de diagnósticos mostrando la tabla de con todas las actividades de diagnóstico.

El dashboard de una actividad de diagnóstico (Ver figura 27) muestra:

- **Resumen de tiempo:** Tiempo promedio dedicado por los estudiantes a completar la encuesta.
- **Preguntas y respuestas:** Detalles de cada pregunta y las respuestas proporcionadas por los estudiantes. Al hacer click en una respuesta, se muestran los estudiantes que eligieron esa opción. En el caso de preguntas de desarrollo, se muestra el nombre y la respuesta del estudiante.
- **Gráfico de completitud:** Un gráfico que indica rápidamente qué estudiantes han completado la encuesta. Al hacer click en el gráfico se muestra el nombre y tiempo de cada estudiante.



Diagnóstico 23 del Texto La Trama Terminado ✕

11.0s
Tiempo Total promedio
Descripción: Prueba
Fecha Límite: 25/6/2024

Respuestas: 3 estudiantes contestaron
Respuestas: Ningún estudiante contestó

Sección Predeterminada

.-Primera pregunta
Tipo: Elección de Respuestas Globales

1) 1	3 RESPUESTAS	100.0%
2) 2	0 RESPUESTAS	0.0%

Gráfico de Término

Terminado Faltantes

Figura 27: Dashboard de datos de una actividad de diagnóstico.

Modificaciones en el Backend: Dentro de la base de datos, a la tabla de las actividades de diagnóstico se le añadió la columna `text_id` para relacionar un texto con la actividad de diagnóstico. A su vez, en las tablas de preguntas del diagnóstico (PollQuestion) y de respuestas del diagnóstico (PollAnswer) se incorporó una columna de “section” para poder crear secciones de preguntas y respuestas en un diagnóstico. La tabla de las respuestas a los diagnósticos se modificó para incluir la propiedad `time`, permitiendo medir y mostrar el tiempo tomado por cada estudiante para completar la encuesta.

Dentro de la API:

- Se implementaron llamadas para la creación de diagnósticos y para obtener la completitud de la actividad, proporcionando los datos específicos de cada estudiante.
- Se modificaron las llamadas de creación de una Actividad de Diagnóstico para incluir la asociación con un texto.

Estas mejoras permiten a los profesores crear diagnósticos personalizados y obtener una visión clara y detallada de las respuestas de los estudiantes, facilitando un mejor seguimiento y análisis de las encuestas realizadas.

10.7. Mejoras de diseño y funcionalidades en el software:

Asociación de un minijuego a una actividad y no al texto:

Anteriormente, el software incluía un minijuego de ordenar párrafos que se agregaba automáticamente a lectura de un texto si el tipo de texto era “minijuego”, en vez del tipo de texto asociativo o expositivo. Esto limitaba a los profesores, ya que si querían realizar una actividad con el mismo texto a diferentes cursos, uno con el minijuego y otro sin él, debían duplicar el texto uno con el tipo minijuego y otro con cualquier otro tipo. Para resolver este inconveniente, se modificó el sistema para que el minijuego se asocie directamente a la actividad y no al texto.

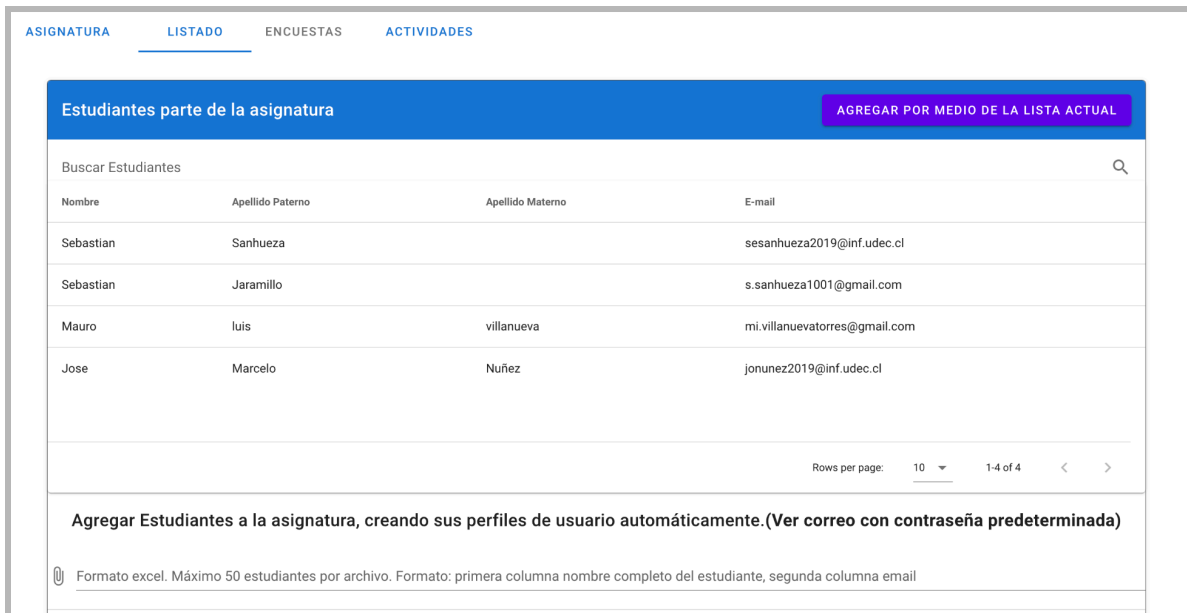
En la base de datos, se añadió la propiedad `minijuego` a la tabla de actividades, permitiendo así una mayor flexibilidad en la evaluación de los estudiantes. Esta misma propiedad es la misma que se utilizó para el minijuego de imágenes.

De esta forma, los profesores tendrán más libertad a la hora de crear actividades y en un futuro será más sencillo integrar nuevos minijuegos dentro de la aplicación.

Cambios de diseño en las vistas de docentes y estudiantes:

Se realizaron mejoras significativas en el diseño del software para mejorar la usabilidad tanto para docentes como para estudiantes:

- **Títulos y Modales de Información:** Se añadieron títulos a los componentes y modales de información en algunas vistas para facilitar la comprensión de las funcionalidades que se pueden realizar.
- **Sidebar Oculto:** Se rediseñó el sidebar, con la posibilidad de ocultarlo y mostrar solo iconos, proporcionando más espacio en pantalla.
- **Estética General:** Se hicieron ajustes de diseño para hacer el sistema más atractivo y fácil de usar.



ASIGNATURA LISTADO ENCUESTAS ACTIVIDADES

Estudiantes parte de la asignatura AGREGAR POR MEDIO DE LA LISTA ACTUAL

Buscar Estudiantes

Nombre	Apellido Paterno	Apellido Materno	E-mail
Sebastian	Sanhueza		sesanhueza2019@inf.udec.cl
Sebastian	Jaramillo		s.sanhueza1001@gmail.com
Mauro	luis	villanueva	mi.villanuevatorres@gmail.com
Jose	Marcelo	Núñez	jonunez2019@inf.udec.cl

Rows per page: 10 1-4 of 4

Agregar Estudiantes a la asignatura, creando sus perfiles de usuario automáticamente. (Ver correo con contraseña predeterminada)

Formato excel. Máximo 50 estudiantes por archivo. Formato: primera columna nombre completo del estudiante, segunda columna email



ASIGNATURA AÑADIR LISTADO CREAR ACTIVIDAD ENCUESTA ACTIVIDADES CREADAS

Formato excel. Máximo 50 estudiantes por archivo. Formato: primera columna nombre estudiante, segund columna email

Buscar Estudiantes

Nombre	Apellido Paterno	Apellido Materno	E-mail
Ignacio	Sanhueza	Becar	ignsanhueza@udec.cl

Rows per page: 10 1-1 of 1

Figura 28: Arriba Sección de Listado del software actual y abajo la sección de listado del software antiguo.

En la figura 28 se muestra el listado de una asignatura tanto en el software antiguo como en el actual acá se puede diferenciar la falta de títulos e indicaciones dentro de esta vista antigua frente a la actual con un formato más estético.

Guardado de información en localStorage:

Uno de los problemas críticos en el software era la pérdida de datos al realizar una actividad, cuando un estudiante recargaba la página o salía del sistema era como si volviera a realizar la actividad desde 0. Esto afectaba el tiempo registrado y el estado de la actividad. Para solucionar esto, se implementó el guardado de datos en localStorage [10], el almacenamiento local del navegador, utilizando el ID de la actividad. Esto asegura que, al recargar la página o regresar al sistema, los datos se recuperen y la actividad continúe desde el punto donde se dejó.

Retroalimentación para estudiantes al finalizar una actividad:

A solicitud de los profesores, se añadió una función de retroalimentación inmediata para los estudiantes al terminar una actividad (Ver figura 29). Al finalizar, los estudiantes son redirigidos a la vista donde pueden ver los resultados de la actividad mostrando respuestas correctas e incorrectas, así como su desempeño en el minijuego, si es que la actividad incluía uno. Esto les proporciona una comprensión clara de sus errores y aciertos, facilitando un aprendizaje más efectivo.

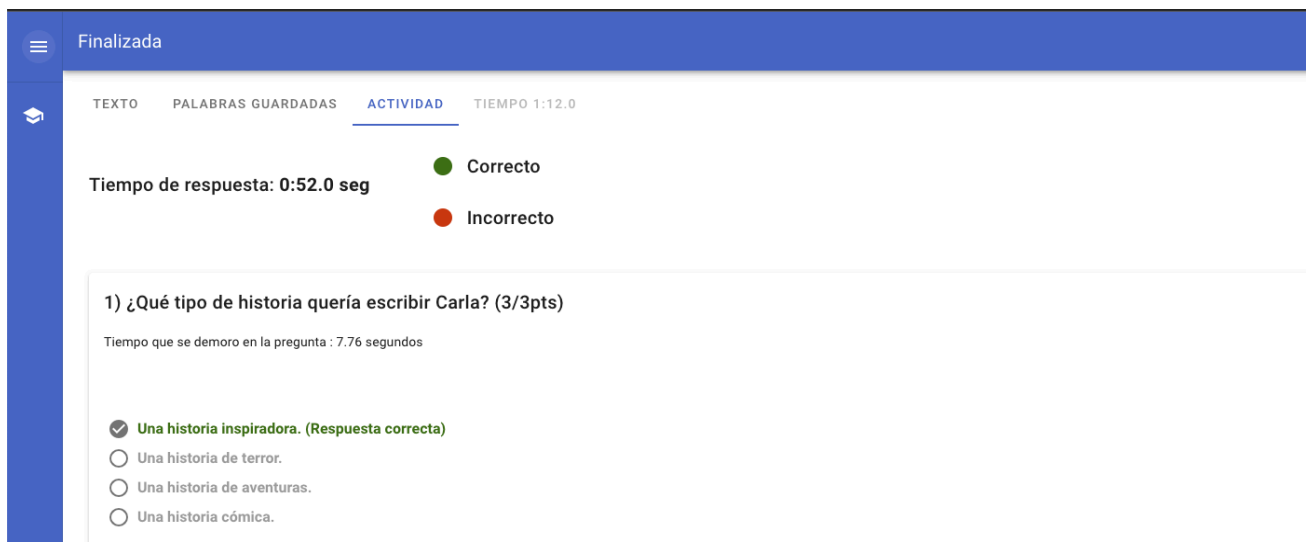


Figura 29: Vista de la retroalimentación mostrada al estudiante al terminar la actividad.

Estas mejoras han optimizado la funcionalidad y usabilidad del software, proporcionando a los usuarios una experiencia más intuitiva y eficiente.

11. Despliegue

El proceso de despliegue de la aplicación se realizó en la plataforma BlueHost, contratada por los profesores. Aunque fue un proceso engorroso, finalmente se logró con éxito. Aquí se detallan los pasos y consideraciones tomadas para el despliegue:

11.1. Preparación y despliegue del Frontend:

1. **Creación de una nueva página en BlueHost:** Se creó una nueva página denominada "lectoentrenodev" dentro de BlueHost. Esta página apuntaba a una carpeta específica en el servidor.
2. **Generación de archivos para despliegue:** Se utilizó el comando `npm build` para generar los archivos de distribución (dist). Estos archivos contienen el frontend optimizado para producción.
3. **Subida de archivos al servidor:** Los archivos generados en la carpeta dist se subieron a la carpeta designada en el servidor de BlueHost. Inicialmente, se encontraron algunos errores menores que fueron corregidos rápidamente.
4. **Configuración inicial:** Se realizaron las configuraciones necesarias para asegurarse de que el frontend pudiera comunicarse con la API correctamente.

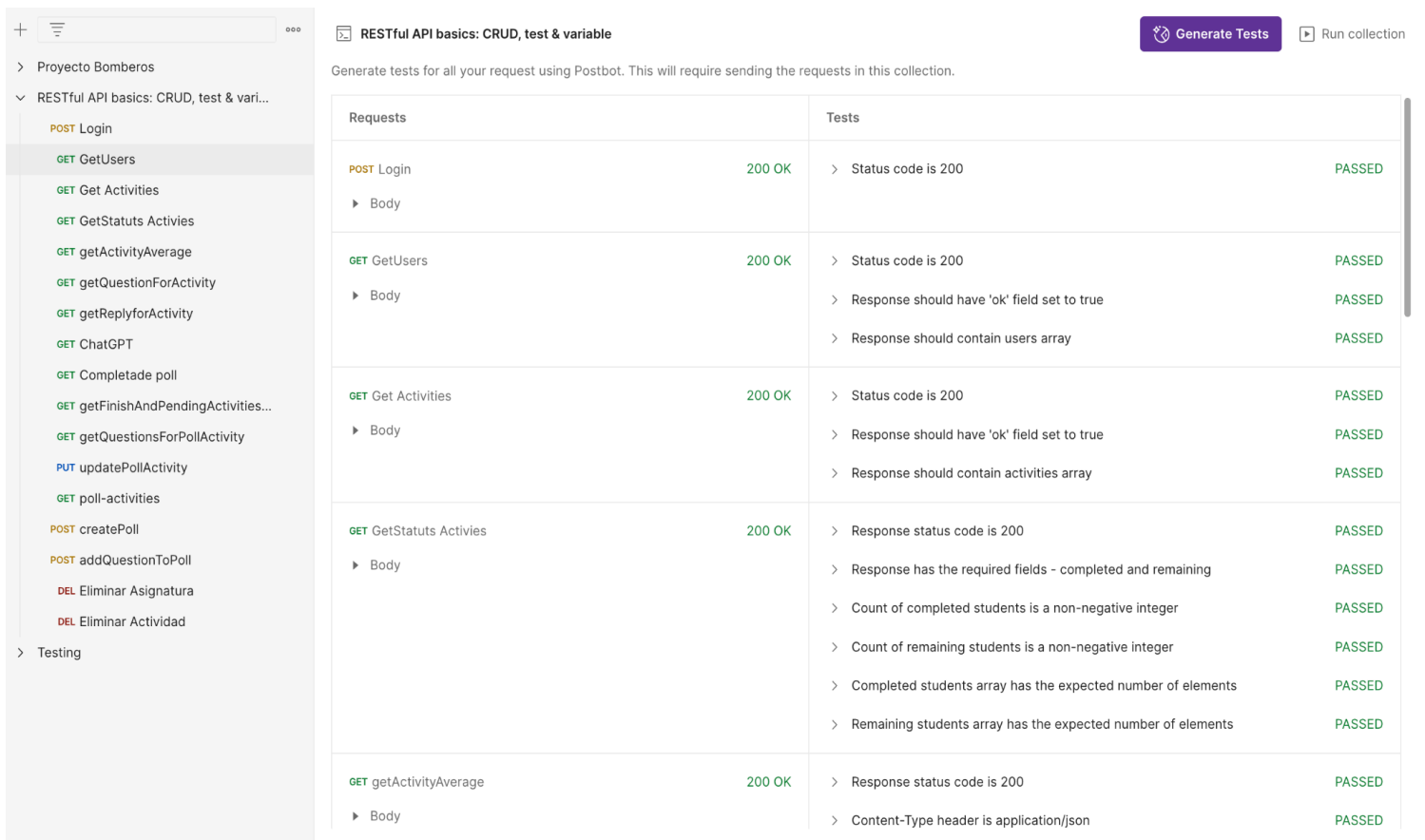
11.2. Preparación y despliegue de la API:

1. **Copia de la base de datos:**
 - Utilizando phpMyAdmin en la cuenta del servidor, se realizó una copia de la base de datos actual. Esta copia se utilizó para la API.
2. **Despliegue de la API en Laravel:** El despliegue de la API fue más complejo ya que requirió ayuda externa. Bluehost, el hosting entregado por las profesoras, no deja implementar una api, ya que está destinado para páginas creadas con wordpress, pero deja implementar aplicaciones laravel, por lo que para lograr subir la API se tuvo que generar una página web, integrando ahí el proyecto laravel con la API. Los pasos incluyen:
 - Generación de una aplicación Laravel en el servidor.
 - Subida de los archivos de la API a esta aplicación Laravel.
 - Configuración de las rutas y controladores para asegurar el correcto funcionamiento de la API.
3. **Verificación y pruebas:** Una vez configurada la API, se realizaron pruebas utilizando Postman para asegurarse de que las llamadas a la API respondieron correctamente.
4. **Integración del frontend con la API:** Con la URL de la API obtenida y funcionando, se integró esta URL en el frontend, asegurando la correcta comunicación entre el frontend y la API.

12. Test y validaciones

12.1. Test unitarios

Para corroborar que funcionara bien el sistema, se implementaron test unitarios para asegurar que las llamadas a la API respondan de manera rápida y correcta. Estos test se realizaron utilizando **Postman** [11], una herramienta ampliamente utilizada para el desarrollo y prueba de APIs. Con Postman, se configuraron test automáticos para validar las respuestas de las diferentes endpoints de la API (Ver figura 30), verificando tanto la integridad de los datos como la eficiencia en los tiempos de respuesta. No se realizaron test a todas las llamadas de la API, por temas de tiempo pero si la mayoría donde se obtienen datos.



RESTful API basics: CRUD, test & variable

Generate Tests Run collection

Generate tests for all your request using Postbot. This will require sending the requests in this collection.

Requests	Tests
<p>POST Login 200 OK</p> <p>▶ Body</p>	<p>> Status code is 200 PASSED</p>
<p>GET GetUsers 200 OK</p> <p>▶ Body</p>	<p>> Status code is 200 PASSED</p> <p>> Response should have 'ok' field set to true PASSED</p> <p>> Response should contain users array PASSED</p>
<p>GET Get Activities 200 OK</p> <p>▶ Body</p>	<p>> Status code is 200 PASSED</p> <p>> Response should have 'ok' field set to true PASSED</p> <p>> Response should contain activities array PASSED</p>
<p>GET GetStatuts Activies 200 OK</p> <p>▶ Body</p>	<p>> Response status code is 200 PASSED</p> <p>> Response has the required fields - completed and remaining PASSED</p> <p>> Count of completed students is a non-negative integer PASSED</p> <p>> Count of remaining students is a non-negative integer PASSED</p> <p>> Completed students array has the expected number of elements PASSED</p> <p>> Remaining students array has the expected number of elements PASSED</p>
<p>GET getActivityAverage 200 OK</p> <p>▶ Body</p>	<p>> Response status code is 200 PASSED</p> <p>> Content-Type header is application/json PASSED</p>

Figura 30 : Test unitarios en la aplicación de postman.

12.2. Validación de usabilidad

Además de los test unitarios, se llevó a cabo una validación de usabilidad por parte de los usuarios mediante el **Sistema SUS (System Usability Scale)** [12], una herramienta de evaluación rápida y confiable para medir la usabilidad percibida de un sistema. Esto se realizó tanto para el usuario profesor como para el estudiante.

El sistema SUS mantiene 10 preguntas y respuestas predefinidas, preguntas de tipo encuesta donde las respuestas con una escala de Likert:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Neutro
4. De acuerdo
5. Totalmente de acuerdo

Para aplicar este método a los usuarios se diseñó un formulario de Google Forms, dirigido a los profesores. Cada formulario contenía las 10 preguntas estándar del sistema SUS, seguidas de algunas preguntas abiertas para obtener un feedback más detallado.

Preguntas del sistema SUS integrando el software:

1. Creo que me gustaría utilizar este el LectoEntreno con frecuencia para apoyarme en clases.
2. Encontré el software de LectoEntreno innecesariamente complejo.
3. Pensé que el software para utilizarlo era fácil de usar.
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este software.
5. Descubrí que las diversas funciones y funcionalidades dentro del software estaban bien integradas.
6. Pensé que había demasiada inconsistencia al momento de utilizar el software.
7. Imagino que la mayoría de la gente aprendería a utilizar el LectoEntreno rápidamente.
8. Encontré el sistema muy complicado de usar.
9. Me sentí muy seguro al utilizar el sistema, sin miedo al utilizar la aplicación.
10. Necesitaba aprender muchas cosas antes de poder empezar a usar el sistema ya que es muy complejo.

De los 10 enunciados disponibles en el sistema SUS, cinco son positivos (1, 3, 5, 7 y 9) y cinco son negativos (2, 4, 6, 8 y 10). Esto no es casual ya que para obtener la puntuación total se

obtiene mediante una serie de cálculos simples, que permiten determinar la satisfacción general de los usuarios con la aplicación.

Preguntas abiertas para el usuario profesor:

1. ¿Encontró errores dentro del software?, si es que es así, ojala adjuntar pantallazos de donde ocurrió el error o los errores, esto sería de mucha ayuda.
2. ¿Encontró que existía una saturación de información de una actividad?, si es así, ¿que le gustaría sacar del dashboard?.
3. ¿Encontró que era fácil o difícil el proceso completo para realizar una actividad a los estudiantes y ver sus resultados?, si es que es así, en qué apartado le costó más.
4. ¿Hay alguna funcionalidad que le gustaría ver en un futuro en el software?

Antes de que se realice el formulario los usuarios deben realizar tareas dentro del software para dar su validación frente a este, por lo que en la descripción del google forms de cada usuario se le indican los pasos a seguir para que puedan probar toda la aplicación.

Descripción del formulario para profesores:

Basado en el sistema SUS, se realizarán preguntas asociadas al software LectoEntreno. Estas preguntas deben ser respondidas después de realizar tareas dentro del software, y se contestarán en una escala del 1 al 5, donde 1 representa una experiencia deficiente y 5 una experiencia excelente. Además, se incluirán preguntas adicionales de desarrollo para obtener una respuesta más detallada y constructiva. Agradezco de antemano tu atención y colaboración.

Pasos o tareas que completar antes de realizar el test del software:
<https://lectoentrenodev.educarteudec.com>

1. El profesor debe iniciar sesión con su cuenta y verificar, que todo funcione correctamente si es que no se acuerda de la contraseña la puede realizar el cambio de contraseña, después se tiene que realizar el proceso completo para ver los resultados de una actividad.
2. Crear una asignatura
3. Crear un texto
4. Añadir preguntas al texto, para que después se puedan asociar a una actividad.
5. Añadir extras al texto, imágenes a las palabras, imágenes a los párrafos para después agregar un minijuego a la lectura etc.
6. Probar el menú de administrador, si es que no son administradores pueden entrar al perfil de prueba y editar su perfil para otorgarle privilegios de administrador, probar el crear un

- usuario de profesor o de algún estudiante, editar algún perfil o revisar la información sin editar.
7. Añadir estudiantes a la asignatura, se puede realizar por 2 medios, por la lista actual del software con todos los estudiantes registrados o también por medio de un Excel que genera los perfiles automáticamente del alumno y se le asigna una clave aleatoria la cual pueden cambiar por medio del login "olvidé mi contraseña", esta clave predeterminada les llega al correo asociado.
 8. Crear una actividad dentro de una asignatura, agregando un minijuego o si se puede crear las 3 actividades posibles con un mismo texto, 1 sin ningún minijuego, 1 con el minijuego de ordenar párrafos y otra de seleccionar la imagen correcta de cada párrafo (Esta actividad solo se puede realizar si tiene alguna foto dentro del texto asociado a un párrafo).
 9. Que los estudiantes realicen las actividades, tratando de realizarla como si fuera una actividad normal de un colegio o universidad, para probar si es que pueden deducir por medio del software lo que tienen que hacer. Indicarle si donde pueden ver la actividad, esto es dentro de la asignatura creada en actividades pendientes, ahí pueden encontrar la actividad.
 10. Obtener y verificar datos por medio del dashboard de actividades en la asignatura, aquí verificar y observar los gráficos, ver si es que está muy saturada de información, ver si pueden obtener los datos de la actividad, tanto por medio de Excel como viendo las pruebas de cada estudiante y sus puntajes obtenidos.

Descripción del formulario para estudiantes:

Hola, soy Michael Villanueva Torres, memorista de la carrera de Ingeniería Civil en Informática, y estoy realizando mejoras al software LectoEntreno. Basado en el sistema SUS, se realizarán preguntas asociadas al sistema. Estas preguntas deben ser respondidas después de realizar tareas dentro del software, utilizando una escala del 1 al 5 donde 1 representa una experiencia deficiente y 5 una experiencia excelente. Además, se agregará una pregunta adicional de desarrollo para obtener una mejor respuesta por tu parte. Desde ya, muchas gracias por tu atención.

Pasos o tareas que completar antes de realizar el test del software:
<https://lectoentrenodev.educarteudec.com>

1. Como estudiante puede crearse un perfil en el apartado de login de la página o puede que el profesor se encargue de crear el perfil que los integrará al sistema, aquí esta como pueden responder la actividad.

2. Encontrar la asignatura creada por el profesor y realizar click en sus actividades, ver las actividades pendientes que creará el profesor.
3. Realizar la actividad leyendo el texto utilizando los extras incluidos en la lectura y realizar el minijuego si está presente.
4. Terminar la lectura y aquí tendrán que realizar el apartado de las preguntas.
5. Responder las preguntas y finalizar la actividad.
6. Revisar la retroalimentación que se da al finalizar.

Ojalá puedan realizar todas sus tareas con éxito, muchas gracias.

Resultados:

Para calcular con las respuestas la puntuación total, se tiene que obtener de esta manera primero suma las respuestas de los enunciados impares y después resta 5, después suma las respuestas de los enunciados pares y resta ese total a 25, al obtener ese resultado, suma ambos resultados y multiplícalo por 2,5.

- Respuestas enunciados impares: **Ejemplo** $\rightarrow (2 + 3 + 1 + 5 + 2) = 13 - 5 = 8$
- Respuestas enunciados pares: **Ejemplo** $\rightarrow (1 + 2 + 2 + 3 + 1) = 25 - 9 = 16$
- Cálculo del SUS: **Ejemplo** $\rightarrow (8 + 16) * 2,5 = 60$ (**Puntaje Total**)

Primero mostraré los resultados obtenidos en ambas encuestas y realizaré una breve lista de los comentarios/observaciones útiles.

Puntajes Obtenidos en la encuesta del profesor: La encuesta la contestaron 4 profesores parte del proyecto FONDECYT.

- Profesor 1: Puntaje Total $\rightarrow 92.5$
 - Impares $(4+5+4+4+5) \Rightarrow (22 - 5) = 17$
 - Pares $(1+1+1+1+1) \Rightarrow (25 - 5) = 20$
 - $17 + 20 = 37 * 2.5 = 92.5$
- Profesor 2: Puntaje Total $\rightarrow 92.5$
 - Impares $(5+5+4+3+5) \Rightarrow (22 - 5) = 17$
 - Pares $(1+1+1+1+1) \Rightarrow (25 - 5) = 20$
 - $17 + 20 = 37 * 2.5 = 92.5$
- Profesor 3: Puntaje Total $\rightarrow 82.5$

- Impares $(4+5+5+3+4) \Rightarrow (21 - 5) = 16$
- Pares $(1+4+1+1+1) \Rightarrow (25 - 8) = 17$
- $16 + 17 = 33 * 2.5 = 82.5$
- Profesor 4: Puntaje Total \rightarrow
 - Impares $(5+5+4+4+4) \Rightarrow (22 - 5) = 17$
 - Pares $(1+1+2+1+1) \Rightarrow (25 - 6) = 19$
 - $17 + 19 = 36 * 2.5 = 90$

En base a los resultados otorgados por el forms generado se obtuvo puntaje promedio de 89.3 puntos y mediante las preguntas abiertas se obtuvo un buen feedback indicando algunos pequeños errores y obteniendo datos para mejorar la usabilidad del usuario.

En base a los criterios del sistema SUS se obtuvo una puntuación relativamente alta, confirmando así la usabilidad del software. En la Figura 31 se muestran los puntajes en grado de aceptación.

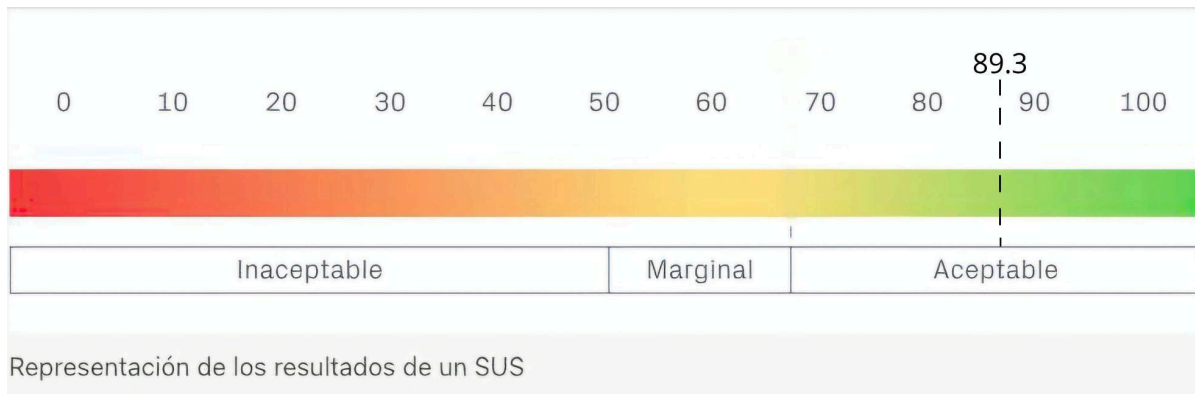


Figura 31 : Representación de los resultados de un SUS

Puntajes Obtenidos en la encuesta del estudiante: Esta encuesta era opcional para a los estudiantes que realizaron las actividades, con el mismo modelo SUS, 8 estudiantes la contestaron.

- Estudiante 1=Impares $(4+3+4+4+5)$ Pares $(3+1+4+2+1) \rightarrow ((20-5)+(25-11)) * 2.5 = 72.5$
- Estudiante 2=Impares $(5+5+5+5+5)$ Pares $(1+1+1+1+1) \rightarrow ((25-5) + (25-5)) * 2.5 = 100$
- Estudiante 3=Impares $(1+5+2+5+4)$ Pares $(1+1+2+1+1) \rightarrow ((17-5) + (25-6)) * 2.5 = 77.5$
- Estudiante 4=Impares $(3+1+5+5+5)$ Pares $(1+1+1+1+1) \rightarrow ((19 -5) + (25-5)) * 2.5 = 80$
- Estudiante 5=Impares $(5+5+5+5+5)$ Pares $(1+1+1+1+1) \rightarrow ((25-5) + (25-5)) * 2.5 = 100$
- Estudiante 6=Impares $(3+4+2+4+4)$ Pares $(1+2+3+1+3) \rightarrow ((17-5) + (25-10))*2.5= 67.5$

- Estudiante 7=Impares (4+5+3+5+5) Pares (1+1+2+1+1) $\rightarrow ((22-5) + (25-6)) * 2.5 = 90$
- Estudiante 8=Impares (3+3+2+4+3) Pares (2+2+2+2+2) $\rightarrow ((15-5) + (25-10)) * 2.5 = 62.5$

Con esto se obtiene un promedio de **81.25 puntos** por parte de los estudiantes, indicando un sistema aceptable.

En base a los resultados y datos entregados, me ayudó a darme cuenta que falta mejorar el apartado de los estudiantes ya que muchos indicaron que no se entiende muy bien cómo realizar la actividad sin ayuda externa.

Como resultado del modelo SUS se obtiene una respuesta positiva por parte de los usuarios, tanto estudiantes como profesores, calificaron de manera positiva la usabilidad del software, comentando en las preguntas abiertas errores y mejoras, logrando obtener un buen feedback por parte de los usuarios para así mejorar la aplicación.

13. Conclusiones

En esta memoria, se implementaron mejoras significativas, tanto como en el diseño de la aplicación como también integrando nuevas funcionalidades para los usuarios en el software web **Lecto Entreno**. Estas mejoras se centraron principalmente para el usuario profesor, en la recopilación y visualización de datos para así brindar una herramienta más robusta que les permita evaluar el progreso de los estudiantes y optimizar sus métodos de enseñanza para mejorar la comprensión lectora.

El software anterior, aunque funcional, presentaba varias deficiencias, particularmente en la calidad y cantidad de datos proporcionados a los usuarios. Estos problemas se abordaron mediante la introducción de nuevas funcionalidades y la corrección de errores existentes, lo que resultó en un sistema más estable y útil.

Se utilizó el **Sistema SUS** para evaluar el software, se obtuvo una puntuación promedio de **89.3 puntos** en las encuestas de los profesores y de **82.5 puntos promedio** por parte de los estudiantes. Esta puntuación refleja la buena aceptación del sistema por parte de los usuarios y sugiere que las modificaciones realizadas lograron su objetivo de mejorar la experiencia del usuario.

Durante todo el proceso de la memoria se realizaron reuniones semanales, tanto con mi profesor patrocinantes o como las profesoras co-patrocinantes, reuniones que sirvieron para tener un

desarrollo constante y poder discutir dudas que se mostraban frente al software o mejoras que sentía que le hacían falta, esta continua comunicación con las personas que eran mi cliente, me permitió mejorar mi confianza y forma de hablar, mejorando así mi comunicación.

Aunque igual se enfrentan a varios desafíos, especialmente en la implementación del frontend ya que al principio, se intentó aplicar conocimientos de **Next.js**, otro framework con javascript, pero pronto me di cuenta de que Vue.js requería un enfoque diferente ya que la forma de generar componentes era distinta, por lo se leyó la documentación del mismo. Los conocimientos previos de cómo estructurar una pantalla o un componente fueron clave para el desarrollo de la memoria, ya que diseñar es fundamental en el desarrollo frontend, para saber que datos necesitas o en la forma que recibirá los datos el usuario.

En resumen, el proyecto no solo logró cumplir con los objetivos planteados, sino que también dejó una base sólida para futuras mejoras y expansiones del software. La positiva recepción por parte de los usuarios y la validación a través del sistema SUS confirman que las decisiones tomadas durante el desarrollo fueron las correctas, y el software ahora está mejor equipado para cumplir su propósito educativo.

14. Recomendaciones para el futuro

14.1. Funcionalidad para compartir textos:

Se recomienda desarrollar una funcionalidad que permita a los profesores compartir textos entre sí. Esta funcionalidad duplica el texto compartido, de modo que cada profesor pueda personalizarlo a su manera según las necesidades de su clase sin repercutir en el contenido original. Esto facilita la colaboración entre docentes mejorando el proceso para obtener un texto.

14.2. Incorporación de IA para corrección automática:

La integración de inteligencia artificial para la corrección automática de respuestas de desarrollo en las actividades. Esta propuesta, que surgió durante las reuniones del proyecto FONDECYT, sugiere que la IA podría asignar una puntuación inicial basada en el contenido del texto y las respuestas de los estudiantes, agilizando el proceso de evaluación. Los profesores tendrían la capacidad de revisar y ajustar estas puntuaciones si consideran que la IA no evaluó correctamente.

A petición de los profesores realice una prueba de concepto integrando llamadas en la API pero todavía es necesario seguir trabajando para evaluar su efectividad y la gestión de costes asociados.

14.3. Actualización y optimización de tecnologías:

Es crucial mantener el software actualizado con tecnologías modernas y optimizar el código para mejorar la velocidad y reducir el tamaño de la aplicación. Durante el despliegue, se observó que los archivos del frontend eran excesivamente grandes debido a las numerosas dependencias. Se recomienda realizar un análisis detallado del contenido de la aplicación utilizando herramientas como webpack-bundle-analyzer [13] para identificar y eliminar dependencias redundantes o innecesarias.

Propongo reconstruir el proyecto desde cero utilizando un framework más moderno como Next.js, que ofrece mejor rendimiento y optimización. Este enfoque permitiría aprovechar las últimas tecnologías y prácticas de desarrollo, garantizando una aplicación más ligera y rápida. La figura 32 muestra el resultado del análisis realizado con webpack-bundle-analyzer, donde se observan, de color azul las librerías integradas instaladas, se reflejan en el tamaño final de la aplicación.

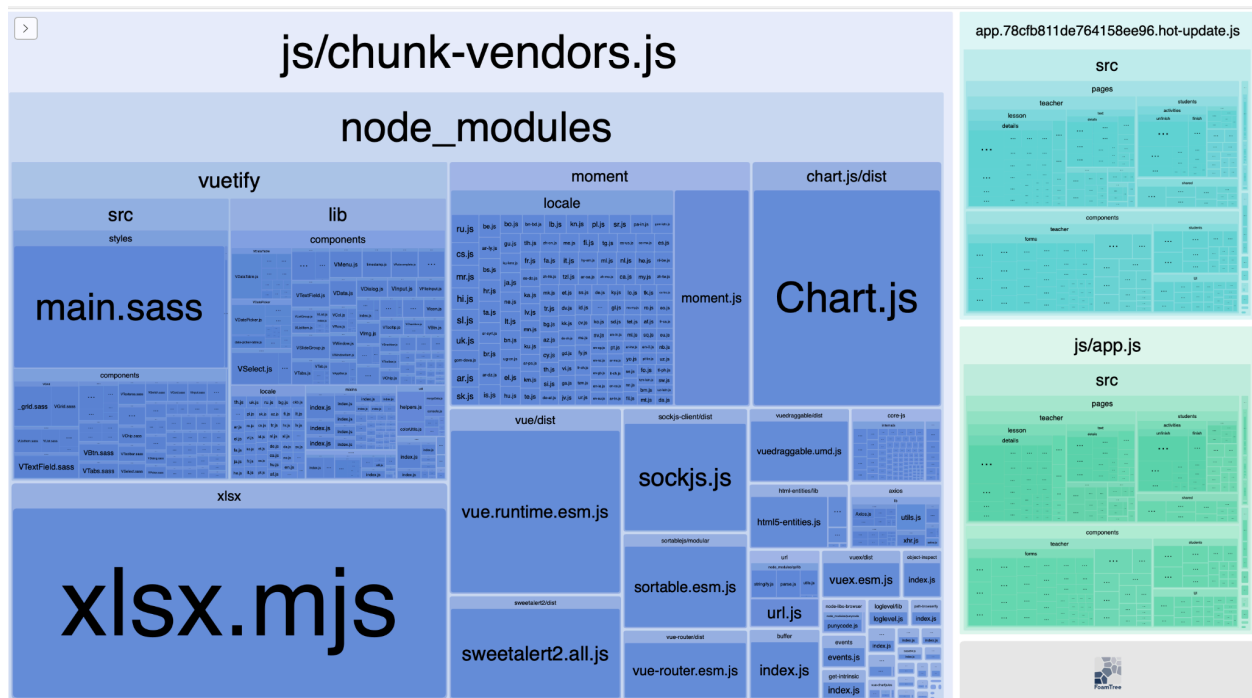


Figura 32: Análisis de tamaño de las librerías integradas en la aplicación usando webpack-bundle-analyzer

En resumen, estas recomendaciones apuntan a mejorar la colaboración entre docentes, agilizar el proceso de evaluación y mantener el software actualizado y optimizado para ofrecer una mejor experiencia de usuario y rendimiento general.

15. Referencias

1. Agencia de Calidad de la Educación. (2019). PISA 2018 Entrega de Resultados. <https://bibliotecadigital.mineduc.cl/handle/20.500.12365/9286>.
2. Sanhueza Becar, I. (2021). LECTOENTRENO, software para apoyo a la comprensión lectora. Tesis de pregrado. Universidad de Concepción.
3. Universidad de Concepción. (2024). Nuevos proyectos NEPSAM apuntarán a la creación de nuevo conocimiento en desafíos científicos actuales. <https://nepsam.udec.cl/fondecyt-regular-2024/>.
4. Vue.js. (n.d.). Vue.js Documentation. <https://v2.vuejs.org/v2/>. Ultima visita: 09/07/24
5. Vuetify.js. (n.d.). Vuetify.js Documentation. <https://v2.vuetifyjs.com/en/>. Ultima visita : 29/06/24
6. Laravel. (n.d.). Laravel Documentation. <https://laravel.com/docs/11.x>. Ultima visita : 26/05/24
7. Vivanco, J. (2019). El modelo C4 de documentación para la arquitectura de software. <https://medium.com/@javiervivanco/el-modelo-c4-de-documentaci%C3%B3n-para-la-arquitectura-de-software-424704528390>. Ultima visita: 04/07/24
8. Laravel. (n.d.). Laravel Middleware. <https://laravel.com/docs/11.x/middleware>. Ultima visita: 10/04/24
9. Laravel. (n.d.). Laravel Filesystem. <https://laravel.com/docs/11.x/filesystem>. Ultima visita: 10/07/24
10. Guvenbas, S. (2022). LocalStorage in Vue.js. <https://medium.com/@sultanguvenbas/localstorage-in-vue-js-1446acc0594>.
11. QAlified.(2023). Postman para API Testing. <https://qalified.com/es/blog/postman-para-api-testing/>. Ultima visita: 10/06/24
12. Busquets, C. (n.d.). Cómo medir usabilidad: ¿Qué es SUS?. <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>. Ultima visita: 11/07/24
13. Npm. (n.d.). webpack-bundle-analyzer. <https://www.npmjs.com/package/webpack-bundle-analyzer>. Ultima visita: 08/06/24