

DESARROLLO DE UN SOFTWARE COLABORATIVO PARA COMPARTIR RECURSOS ACADÉMICOS DIGITALES

DEVELOPMENT OF COLLABORATIVE SOFTWARE FOR SHARING DIGITAL ACADEMIC RESOURCES

Recibido: 29 de octubre de 2025

Aceptado: 31 de diciembre de 2025

Andrea Yamileth Martínez Muñoz

Tecnológico Nacional de México / IT de
Ciudad Valles

<https://orcid.org/0009-0003-7754-2342>

Autor de correspondencia:

22690413@tecvalles.mx

Miguel Angel Montoya Cruz

Tecnológico Nacional de México / IT de
Ciudad Valles

<https://orcid.org/0009-0007-8043-8297>

Fernando Domingo Espinoza

Tecnológico Nacional de México / IT de
Ciudad Valles

<https://orcid.org/0009-0005-4735-4644>

Marcelo Uribe Davila

Tecnológico Nacional de México / IT de
Ciudad Valles

<https://orcid.org/0009-0007-7901-8930>

RESUMEN

El presente artículo describe el proceso de desarrollo de un software que tiene como objetivo fomentar la colaboración entre estudiantes mediante recursos académicos. El proyecto se implementó utilizando la metodología ágil Kanban la cual resulta ideal para equipos con pocos integrantes al permitir un flujo constante del trabajo mediante la gestión visual de tareas en cinco fases: *backlog*, *ready*, *in progress*, *in review*, y *done*. A través de esta metodología fue posible llevar a cabo desde el análisis de requerimientos planteados, la definición de las etapas de desarrollo tanto para el *frontend* como para el *backend*, así como la asignación de actividades y el establecimiento de una comunicación continua entre los integrantes del equipo. El software se desarrolló con tecnologías como React para la capa visual e interactiva, Node.js y Express para el manejo del servidor y las peticiones, y PostgreSQL como gestor de base de datos. Asimismo, se utilizó Docker para el despliegue y las pruebas en entornos controlados. El resultado es una aplicación *web* completa y funcional para la colaboración académica mediante recursos digitales entre estudiantes.

Palabras clave: programación, kanban, tecnologías, biblioteca, *web*

Abstract

This article describes the development process of software designed to foster collaboration among students using academic resources. The project was implemented using the Kanban agile methodology, which is ideal for small teams as it allows for a continuous workflow through the visual management of tasks in five phases: *backlog*, *ready*, *in progress*, *under review*, and *done*. This methodology enabled the analysis of requirements, the definition of development stages for both the *frontend* and *backend*, the assignment of tasks, and the establishment of continuous communication among team members. The software was developed using technologies such as React for the visual and interactive layer, Node.js and Express for server management and requests, and PostgreSQL as the database management system. Docker was used for deployment and testing in controlled environments. The result is a complete and functional *web* application for academic collaboration among students using digital resources.

Keywords: programming, Kanban, technologies, library, *web*

INTRODUCCIÓN

El internet actualmente es una de las herramientas más utilizadas en cuestiones académicas o la vida diaria, el acceso a recursos académicos digitales se ha incrementado y se han posicionado como plataformas indispensables que permiten la consulta y el aprovechamiento de la información.

Existen diversas plataformas con el concepto de biblioteca compartida, por ejemplo:

NCore plataforma de software y arquitectura de código abierto para la creación de bibliotecas digitales flexibles y colaborativas. Desarrollada por el proyecto Biblioteca Digital Nacional de Ciencias (NSDL), constituye su infraestructura técnica central. Consta de un repositorio digital central basado en Fedora, un modelo de datos específico, una API y un conjunto de servicios de *backend* y herramientas de *frontend* que crean un nuevo modelo para bibliotecas digitales colaborativas y contributivas. (Krafft , Birkland y Cramer, 2018, p.1)

Según lo señalado por Underwood, Black, Auvil y Capitanu (2013):

Las bibliotecas abarcan varios siglos, los géneros identificados pueden cambiar gradualmente a lo largo del tiempo. Debido a que los volúmenes son mucho más extensos que los artículos, tienden a ser internamente heterogéneos, y la clasificación debe comenzar con la segmentación. Describimos una solución multicapa que entrena modelos ocultos de Markov para segmentar volúmenes y utiliza conjuntos de clasificadores superpuestos para abordar el cambio histórico. Probamos este enfoque en una colección de 469,200 volúmenes de la Biblioteca Digital HathiTrust. Para demostrar el valor humanístico de estos métodos, extraemos 32,209 volúmenes de ficción de la biblioteca digital y analizamos la evolución de la proporción de narración en primera y tercera persona en el corpus (p.1-3).

En el presente artículo se describe el uso de la metodología Kanban y sus principales características para la implementación de un software colaborativo. Según Castellano (2019) Esta metodología busca conseguir un proceso productivo, organizado y eficiente. Se creó en Toyota (Japón) y se utiliza para controlar el avance del trabajo en una cadena de producción. Forma parte de la metodología Lean Manufacturing basada en la utilización de técnicas just-in-time (JIT). El principal objetivo del sistema Kanban es asegurar una tasa de producción sostenible para evitar exceso de producto terminado, cuellos de botella y retrasos en la entrega de pedidos. Los trabajos en curso deben organizarse en función de la capacidad del centro de trabajo y equipos. Requiere una comunicación en tiempo real sobre la capacidad y una transparencia del trabajo total.

Este proyecto de innovación e investigación pretende comprender cuáles son las prácticas colaborativas que pueden ayudar a mejorar el aprendizaje, generando así mayor capacidad colectiva mediante una herramienta tecnológica.

Ante la identificación de las áreas de oportunidad, se vuelve necesario desarrollar una herramienta que facilite la gestión y el intercambio de recursos educativos, promoviendo la colaboración. El objetivo principal de este proyecto es desarrollar un software colaborativo que permita a los estudiantes compartir, comentar y organizar recursos académicos por materia. El software integra un sistema de registro de usuarios, clasificación de materias por categorías y un módulo de interacción mediante comentarios y votos.

METODOLOGÍA

Problemática

Actualmente no existe ningún software especial en línea en el que se pueda compartir recursos de interés académico entre los estudiantes de las diversas carreras del Instituto Tecnológico de Ciudad Valles.

Debido a lo anterior surgió nuestra propuesta de innovación, la cual se explica en el presente artículo, utilizando una investigación que se clasifica como aplicada y con un enfoque descriptivo-técnico.

Metodología de desarrollo

La metodología utilizada durante el desarrollo del software se enfocó en el uso de prácticas ágiles y con la incorporación de diversas tecnologías actuales para asegurar que el procedimiento de implementación se llevará a cabo de manera organizada y eficiente. En este apartado se describe el enfoque metodológico así como las herramientas empleadas para el desarrollo del sistema.

La metodología ágil Kanban fue la seleccionada para el desarrollo del software porque es apropiada para equipos pequeños que tienen ciclos de trabajo continuos y se adaptó bien a nuestro caso. Kanban posibilita una visualización clara del avance a través de un tablero fraccionado en fases que simbolizan la situación de cada tarea dentro del proceso de desarrollo.

El tablero Kanban del proyecto se estructuró en las siguientes cinco fases, utilizando la herramienta de gestión de proyectos Github Projects:

- **Backlog:** para la recopilación inicial de los requerimientos, ideas y funcionalidades propuestas.
- **Ready:** tareas priorizadas y listas para ser desarrolladas.
- **In Progress:** especifica las actividades que se encuentran en proceso de implementación.
- **In Review:** tareas completadas y en revisión por parte del equipo.
- **Done:** actividades ya finalizadas y con aprobación del equipo.

Kanban se basa en principios Lean y se implementa mediante tableros visuales que permiten observar el estado de las tareas en columnas como “Backlog”, “In progress” y “Done” (Castillo y Guaña, 2024). Esta visualización facilita la comprensión global del avance del proyecto y permite identificar de manera temprana los cuellos de botella.

Como mencionan Castillo y Guaña (2024), Kanban se caracteriza por promover la eficiencia, la transparencia del flujo de trabajo y la colaboración dentro del equipo. Su origen en el Sistema de Producción de Toyota le proporciona una orientación hacia la mejora continua, limitando el trabajo en progreso y permitiendo que las tareas fluyan de manera constante a través del sistema.

Este enfoque resultó especialmente adecuado para nuestro proyecto, ya que permitió organizar, priorizar y distribuir las actividades tanto del *frontend* como del *backend* de forma flexible.

Además, Castillo y Guaña, (2024) destacan que Kanban es una metodología adaptable a diversos lenguajes, entornos y arquitecturas de software, característica que coincidió con las necesidades del proyecto. Esto permitió la integración de tecnologías como React para la interfaz de usuario, Node.js y Express para el servidor, PostgreSQL como base de datos y Docker para la ejecución en entornos controlados.

En resumen, la selección de la metodología Kanban nos permitió gestionar el proyecto de forma ordenada, y visualmente clara, lo que contribuyó a un desarrollo más eficiente.

Tecnologías utilizadas

Para desarrollar el software, se utilizó un conjunto de herramientas tecnológicas que facilitaron la creación de una aplicación moderna y escalable. Las tecnologías más relevantes que se implementaron fueron:

Frontend

El Diseño de Interfaz de Usuario adquiere una elevada importancia en la usabilidad, la experiencia de usuario y la calidad del software, de esta manera se hace énfasis en la relevancia de un buen desarrollo de la parte visual del sistema (Labrada, 2020).

Así que se empleó React, una biblioteca de JavaScript que simplifica la creación de componentes reutilizables, el manejo del estado y la actualización correcta de la interfaz. React permitió que se desarrollara un entorno dinámico en el cual los usuarios tienen la posibilidad de navegar y visitar los diferentes recursos digitales académicos con facilidad.

Backend

La API REST fue implementada gracias al uso de Node.js y del framework Express, lo que permitió al servidor gestionar las peticiones, verificará los datos, administra las sesiones de usuario y de esta forma manejar la lógica del sistema.

Base de datos

Para esta sección del proyecto se empleó PostgreSQL, ya que es un gestor de bases de datos relacional sólido, eficiente y con mucha documentación, para, de esta manera, lograr almacenar la información. La base de datos fue creada con tablas principales como recursos académicos y usuarios, de igual manera la definición de campos y relaciones entre ellas lo cual ha posibilitado la organización estructurada de los datos.

Contenedores y despliegue

En base a las pruebas y con el objetivo final de asegurar un entorno de ejecución controlado y replicable, se utilizó Docker. Esta herramienta nos posibilita crear contenedores independientes para el servidor, la base de datos y otros servicios necesarios para el funcionamiento del software. Su uso permitió realizar las tareas de prueba y despliegue.

Arquitectura del sistema

El software se implementó como una arquitectura cliente–servidor, compuesta por las dos capas principales:

- **Frontend:** esta es la interfaz diseñada para el cliente, que permite la interacción visual con el usuario y su desarrollo fue con React.
- **Backend:** el servidor es responsable del procesamiento de datos, lógica del sistema y gestión de peticiones, su implementación se llevó a cabo con Node.js y Express.

Lizama, Kindle, Jeira, y Gonzalez (2016) definen a la parte del cliente como programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito, una vez que son servidas sus solicitudes, termina el trabajo.

Por otro lado, los mismos autores describen a la parte del servidor como un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente.

Finalmente, el sistema se desplegó en contenedores haciendo uso de la tecnología Docker, lo que garantizó entornos de prueba aislados y replicables para cada servicio: servidor, base de datos y herramientas auxiliares.

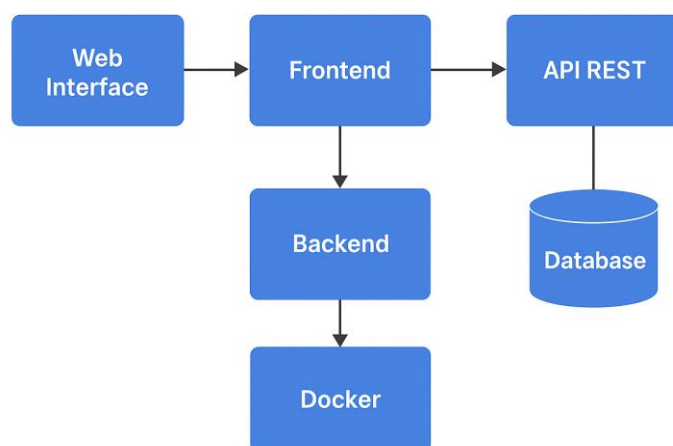


Figura 1

Arquitectura general del sistema

Nota: La Figura 1 muestra de manera general la arquitectura cliente-servidor empleada en el

proyecto. Fuente: Propia

Diseño de la base de datos

El diseño y desarrollo de la base de datos, fue implementada usando PostgreSQL, y cuenta con una estructura relacional para de esta manera lograr la integridad de los datos. Las tablas diseñadas fueron la siguientes:

- **Usuarios:** almacena los datos de identificación básicos como el nombre, correo, contraseña, y fecha de registro, de tal manera estos mismos datos sirven de autenticación.
- **Recursos:** incluye diferentes campos, entre los principales están el identificador, título, descripción, enlaces, además de su autor, fecha en que fue subido y materia asociada.
- **Materias:** en esta tabla se almacenan principalmente los nombres de las materias a las que los recursos estarán asociados.
- **Votos:** guarda la cantidad de votos asociados a un recurso en tiempo real, además clasifica el voto como positivo o negativo, para de esta manera aumentar o disminuir la cantidad al instante.
- **Comentarios:** esta tabla contiene los campos que almacenan el contenido de las diferentes interacciones asociadas con cada recurso.

Por ende, las tablas se asocian a través de llaves foráneas y primarias, lo que posibilita un flujo óptimo y organizado de los datos.

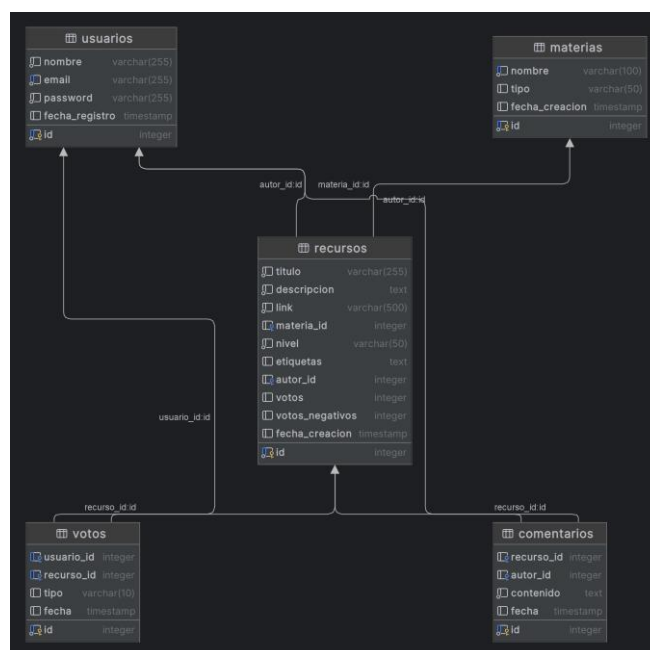


Figura 2

Diagrama entidad relación

Nota: En la figura 2 se muestra el diagrama entidad relación en el que se rige la base de datos.

Fuente: Propia

Flujo de usuario

El funcionamiento en general del sistema se resume en un proceso sencillo:

1. El usuario ingresa primeramente al sistema y puede observar los recursos académicos disponibles ordenados de manera que los más recientes se muestran primero.
2. Una vez dentro puede buscar, navegar y entrar a consultar los recursos publicados.
3. Si el usuario está registrado tiene la opción de subir nuevos recursos agregando título, descripción y demás, y si no se encuentra registrado, existe un apartado de registro.
4. Si el usuario se encuentra firmado en el sistema puede interactuar comentando y votando diferentes recursos académicos según lo considere.
5. En su panel personal puede administrar sus aportaciones y consultar la cantidad de comentarios y votos en sus recursos publicados.



Figura 3

Diagrama del flujo de usuario

Nota: En la presente figura se observa de manera clara el flujo de usuarios en el sistema.

Fuente: Propia

RESULTADOS Y DISCUSIÓN

El desarrollo del software dio como resultado una aplicación *web* colaborativa totalmente funcional en su primera versión, fue creada para simplificar la colaboración mediante recursos académicos digitales entre estudiantes. El sistema integra un *frontend* dinámico, un *backend* robusto y una base de datos con una estructura que funciona en conjunto mediante API REST.

Los siguientes son los resultados más importantes obtenidos:

Módulos funcionales del sistema

Como resultado del desarrollo del sistema, se lograron los objetivos propuestos en cuanto a la funcionalidad de todos los módulos, logrando un sistema completamente funcional. Los principales módulos son los siguientes.

Módulo de autenticación de usuarios

Este módulo permite el inicio de sesión o registro mediante credenciales básicas. Incluye las validaciones, y manejo de sesiones. Su principal función consiste en garantizar acceso seguro a las funcionalidades completas del sistema.

La imagen muestra dos interfaces de usuario lado a lado, separadas por una línea vertical. La interfaz de la izquierda es para 'Iniciar Sesión' y la de la derecha es para 'Crear Cuenta'. Ambas tienen un diseño limpio con campos de entrada blancos y botones azules.

Interfaz de Inicio de Sesión:

- Título: **Iniciar Sesión** (con ícono de llave).
- Subtítulo: *Accede a tu biblioteca digital*.
- Campo: Correo Electrónico (con ícono de correo) con el valor 'tu@email.com'.
- Campo: Contraseña (con ícono de candado) con caracteres ocultos por asteriscos.
- Botón: **Iniciar Sesión**.
- Enlaces: [¿No tienes cuenta? Regístrate](#) y [Volver al inicio](#).

Interfaz de Crear Cuenta:

- Título: **Crear Cuenta** (con ícono de persona).
- Subtítulo: *Únete a la comunidad estudiantil*.
- Campo: Nombre Completo (con ícono de persona) con el valor 'Juan Pérez'.
- Campo: Correo Electrónico (con ícono de correo) con el valor 'tu@email.com'.
- Campo: Contraseña (con ícono de candado) con caracteres ocultos por asteriscos.
- Nota: *Mínimo 6 caracteres*.
- Botón: **Crear Cuenta**.
- Enlaces: [¿Ya tienes cuenta? Inicia sesión](#) y [Volver al inicio](#).

Figura 4

Módulos de inicio de sesión y registro

Nota: En la Figura 4 se observan las interfaces de inicio de sesión y registro. Fuente: Propia

Módulo de búsqueda y filtrado

Esta funcionalidad contiene un sistema de filtros por fecha de creación, materia, y nivel, además de contar con un buscador que identifica palabras clave asociadas a recursos académicos.

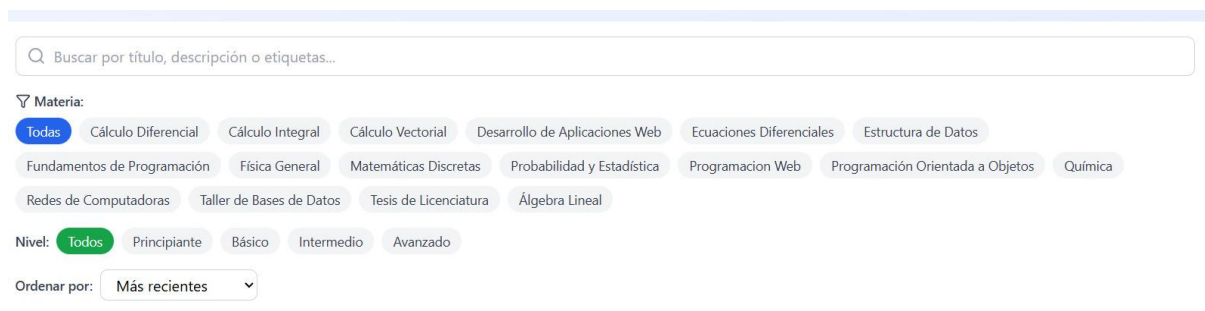


Figura 5

Módulo de búsqueda y filtrado

Nota: La Figura 5 muestra la interfaz del módulo de búsqueda y filtrado. Fuente: Propia

Módulo del panel de usuario

Para este apartado cada usuario tiene un área en la que puede administrar y listar todos sus recursos publicados, subir uno nuevo, modificar la información, o borrar material que haya compartido anteriormente. Además, puede observar el número total de interacciones en sus recursos publicados.

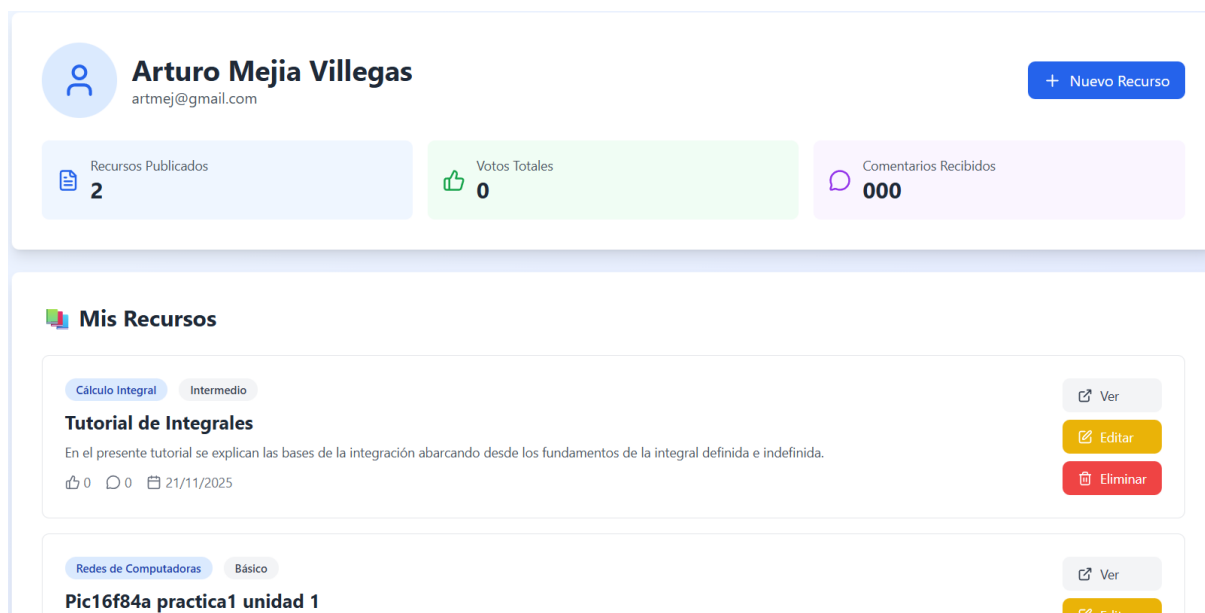


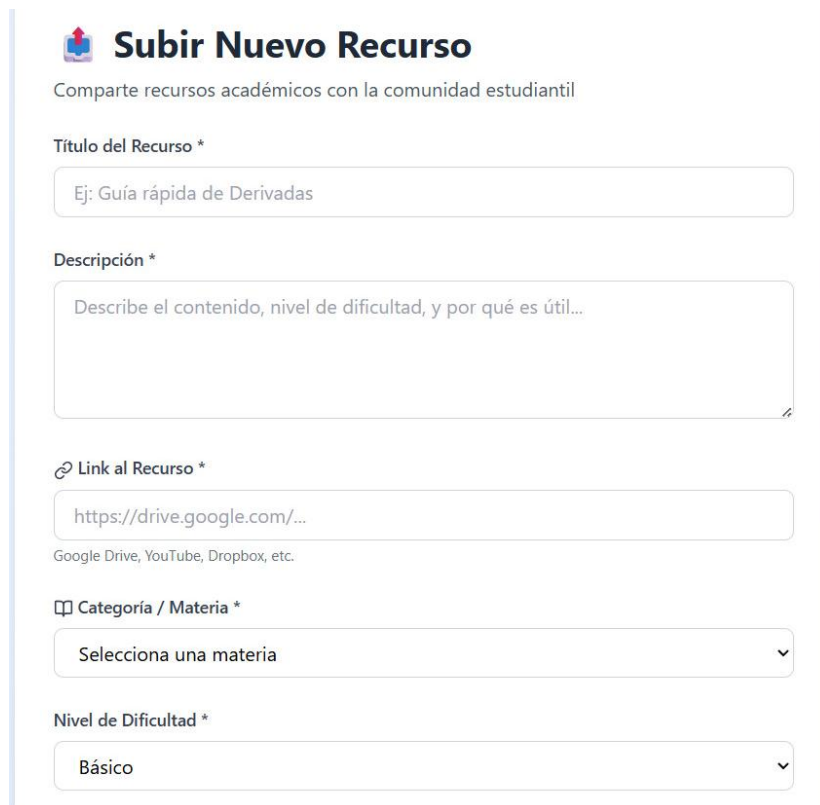
Figura 6

Panel de usuario

Nota: En la Figura 6 se muestra la interfaz del panel de usuario en donde se observan diferentes funcionalidades. Fuente: Propia

Módulo para subir recursos académicos

Este apartado del sistema permite a los usuarios publicar recursos académicos, añadiendo descripciones, organizarlo por materia o categoría y además asignarle un nivel de dificultad.



Subir Nuevo Recurso

Comparte recursos académicos con la comunidad estudiantil

Título del Recurso *

Ej: Guía rápida de Derivadas

Descripción *

Describe el contenido, nivel de dificultad, y por qué es útil...

Link al Recurso *

<https://drive.google.com/...>

Google Drive, YouTube, Dropbox, etc.

Categoría / Materia *

Selecciona una materia

Nivel de Dificultad *

Básico

Figura 7

Modulo para subir recursos académicos

Nota: La Figura 7 muestra la interfaz que permite a los usuarios subir nuevos recursos académicos. Fuente: Propia



Mostrando 3 recursos

Matemáticas Discretas Intmedio

Pic16f84a practica1 unidad 1

practica en ensamblador

codigo simulacion

Mikel Angel Montoya Cruz 14/11/2025

0 0 0 Abrir

Cálculo Diferencial Intmedio

Guía de Derivadas

Apuntes completos con ejercicios

derivadas ejercicios matemáticas

Juan Pérez 14/11/2025

1 0 +1 2 Abrir

Cálculo Diferencial Intmedio

Guía completa de Cálculo Diferencial

Apuntes completos con ejercicios resueltos paso a paso

derivadas limites ejercicios

Juan Pérez 13/11/2025

1 0 +1 0 Abrir

Figura 8

Interfaz de los recursos publicados

Nota: La Figura 8 muestra cómo se observan todos los recursos publicados en el sistema. Fuente: Propia

Ambiente de ejecución seguro

Con Docker, logramos dividir el servidor y la base de datos en contenedores autónomos. Esto facilitó las pruebas y los despliegues, evitando dificultades de configuración entre distintas computadoras.

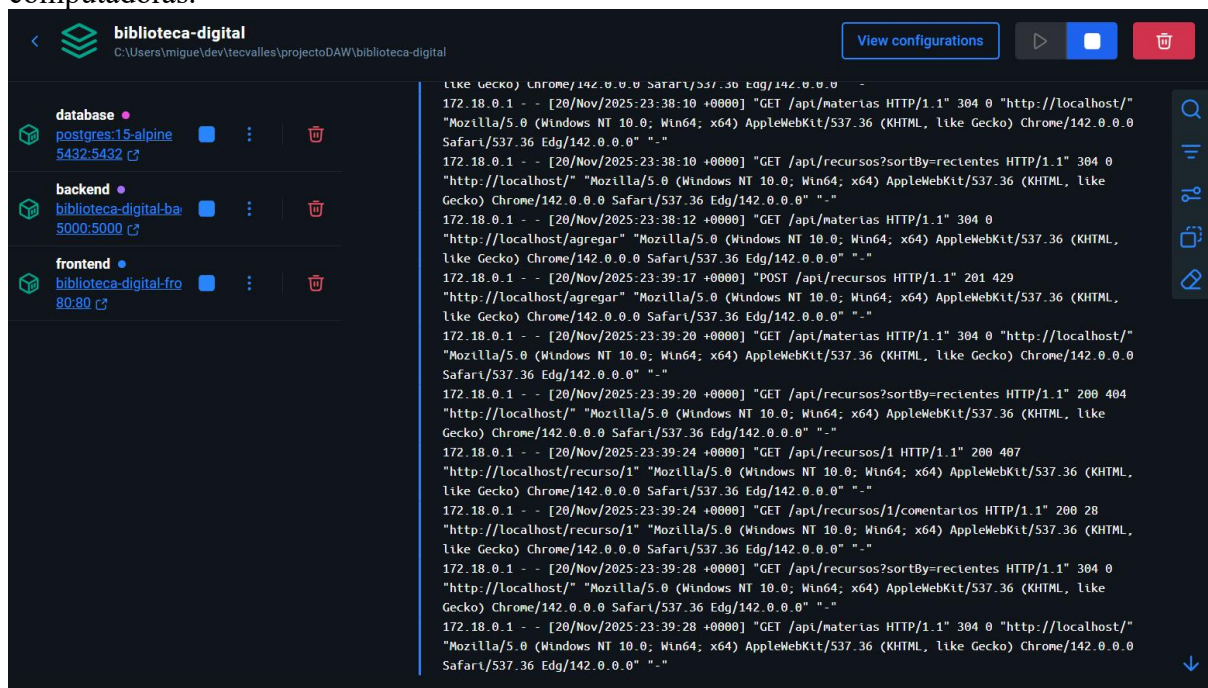


Figura 9

Estructura de contenedores docker

Nota: La figura 9 muestra la estructura implementada y corriendo en docker para los diversos contenedores de la parte del *frontend*, *backend* y la base de datos. Fuente: Propia

DISCUSIÓN Y CONCLUSIÓN

Desde hace varios años las plataformas digitales se han convertido en una herramienta trascendental. Por ello las tecnologías con enfoque académico ayudan a mejorar el aprendizaje de los estudiantes.

En base a la implementación de una metodología ágil como la Kanban, el desarrollo del sistema propuesto fue concluido con éxito.

En esta ocasión se desarrolló un sistema enfocado principalmente para estudiantes con la intención de ser una opción que ayude a fomentar el trabajo colaborativo a través de recursos académicos digitales con contenido de interés para alumnos de las diversas carreras del instituto.

Hasta el momento se ha desarrollado la primera versión de este sistema con éxito en base a los objetivos propuestos. Por supuesto que el equipo de desarrollo ha considerado mejoras a futuro como el implementar un apartado para subir trabajos de fin de grado como tesis o investigaciones, así como también permitir publicar archivos desde el dispositivo de preferencia del usuario, los cuales se quedarán almacenados en un servidor y a disposición de todos los usuarios, siempre contemplando las medidas pertinentes de seguridad.

REFERENCIAS

- Castellano Lendínez, L. (2019). Kanban. Metodología para aumentar la eficiencia de los procesos. *3C Tecnología_Glosas de innovación aplicadas a la pyme*, 29(1), 30–41. <https://doi.org/10.17993/3ctecno/2019.v8n1e29/30-41>
- Castillo Anzules, M., & Guaña Moya, E. J. (2024). Kanban: Una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software, una revisión sistemática. *Revista Ingenio global*, 3(1), 17–28. <https://doi.org/10.62943/rig.v3n1.2024.68>
- Krafft, D. B., Birkland, A., & Cramer, E. J. (2008). *NCore: Architecture and Implementation of a Flexible, Collaborative Digital Library* (No. arXiv:0803.1500). arXiv. <https://doi.org/10.48550/arXiv.0803.1500>
- Labrada, S. M. (2020). Principios del proceso de diseño de interfaz de usuario. *Revista Cubana de Transformación Digital*, 1(3), 143–155.
- Lizama, O., Kindley, G., Morales, J. I. J., & Gonzales, A. (s/f). *Redes de computadores Arquitectura Cliente—Servidor*.
- Underwood, T., Black, M. L., Auvil, L., & Capitanu, B. (2013). Mapping Mutable Genres in Structurally Complex Volumes. *2013 IEEE International Conference on Big Data*, 95–103. <https://doi.org/10.1109/BigData.2013.6691676>