



**INSTITUTO SUPERIOR  
TECNOLÓGICO QUITO**  
Formamos tu PROPÓSITO DE VIDA

**YO  
SOY  
ITQ**



PROYECTO FINAL DE ASIGNATURA

# Desarrollo de Software

[WWW.ITQ.EDU.EC](http://WWW.ITQ.EDU.EC)

2024

**CUADERNO DEL  
ESTUDIANTE**



**INSTITUTO SUPERIOR  
TECNOLÓGICO QUITO**  
Formamos tu PROPÓSITO DE VIDA

**PROYECTO FINAL DE ASIGNATURA**

**BASES DE DATOS NO RELACIONALES**

**MGTR. MÓNICA ALEXANDRA RAMÍREZ**

**FRANCIS EDUARDO MENDOZA/ANGELO GEOVANY  
NAZARENO**

**ABRIL 2025 – AGOSTO 2025**



## CONTENIDO

OBJETIVO DE LA ASIGNATURA .....	5
INDICACIONES GENERALES .....	<b>¡Error! Marcador no definido.</b>
I INTRODUCCIÓN.....	5
1. CAPÍTULO I .....	6
1.1. NOMBRE DEL PROYECTO .....	6
1.2. PLANTEAMIENTO DEL PROBLEMA.....	6
1.3. OBJETIVOS GENERALES .....	9
1.4. OBJETIVOS ESPECÍFICOS.....	9
1.5. JUSTIFICACIÓN .....	9
2. CAPÍTULO II .....	11
2.1. MARCO TEÓRICO .....	11
3. CAPÍTULO III .....	14
3.1. DESARROLLO Y/O IMPLEMENTACIÓN .....	14
4. CAPÍTULO IV .....	17
4.1. ANÁLISIS DE RESULTADOS .....	17
5. CAPÍTULO V .....	19
5.1. CONCLUSIONES.....	19
5.2. RECOMENDACIONES.....	19
6. ANEXOS .....	21
7. BIBLIOGRAFÍA.....	22

## Índice de Figuras

Ilustración 1 Usuarios.....	14
Ilustración 2 Publicaciones.....	15
Ilustración 3 Feed de publicaciones.....	21
Ilustración 4 Feed de bienvenida.....	21
Ilustración 5 Feed de solicitudes.....	22



## OBJETIVO DE LA ASIGNATURA

La asignatura de Base de Datos No Relacionales tiene como objetivo el uso y aplicación de bases de datos orientadas a documentos, como MongoDB, permitiendo comprender su estructura, ventajas, y usos reales frente a las bases de datos relacionales. A través del desarrollo de un proyecto práctico, se espera adquirir la capacidad de diseñar, construir y manipular bases de datos NoSQL aplicadas a contextos actuales como redes sociales, sistemas de recomendación y otras aplicaciones web modernas.

## I INTRODUCCIÓN

Las redes sociales se han convertido en uno de los pilares de la comunicación moderna, conectando a personas alrededor del mundo y generando enormes cantidades de datos que deben ser almacenados y procesados de forma eficiente. En este contexto, las bases de datos NoSQL, y en particular MongoDB, han demostrado ser una herramienta poderosa para gestionar grandes volúmenes de información no estructurada, lo que las convierte en la elección ideal para este tipo de plataformas.

El presente proyecto tiene como objetivo desarrollar una aplicación web que simule una red social, donde los usuarios puedan crear perfiles, agregar amigos, hacer publicaciones, dar likes, y recibir notificaciones, todo esto con una arquitectura basada en MongoDB. Esta propuesta busca no solo afianzar los conocimientos adquiridos en la asignatura, sino también poner en práctica habilidades de desarrollo frontend y backend, diseño de interfaces, y conexión entre capas tecnológicas.

En los siguientes capítulos se detallarán el problema que motiva este desarrollo, los objetivos que guían la propuesta, la justificación de su importancia, el marco teórico que respalda su estructura, el desarrollo práctico de la aplicación, el análisis de los resultados obtenidos, y finalmente, las conclusiones y recomendaciones derivadas del trabajo realizado.

## 1. CAPÍTULO I

### 1.1. NOMBRE DEL PROYECTO

**Interacción Social Dinámica: Desarrollo de una Aplicación Web con MongoDB para Simular una Red Social Moderna.**

### 1.2. PLANTEAMIENTO DEL PROBLEMA

**Situación existente:** Hoy en día, el desarrollo de aplicaciones web está en constante evolución y una de las funcionalidades más comunes y demandadas son las redes sociales. Sin embargo, muchos proyectos académicos o educativos suelen quedarse en ideas básicas y no profundizan en la lógica real que implica el manejo de datos dinámicos como usuarios, publicaciones, relaciones y notificaciones.

**Consecuencias de no actuar:** Si no se generan experiencias prácticas que simulen sistemas reales como una red social, corremos el riesgo de no entender cómo se aplican los conceptos de bases de datos NoSQL en el mundo real. Esto puede generar una brecha entre el conocimiento teórico y su aplicación práctica, lo que afecta su preparación profesional.

¿Cómo diseñar y desarrollar una aplicación web tipo red social que aproveche la estructura flexible y escalable de MongoDB para gestionar usuarios, interacciones, contenido y notificaciones en tiempo real?

En el contexto actual, el uso intensivo de las redes sociales ha transformado la manera en que las personas se comunican, comparten información y establecen relaciones. Esto ha

generado la necesidad de plataformas que sean robustas, escalables y seguras, capaces de gestionar grandes volúmenes de datos y de interacciones en tiempo real. Sin embargo, el desarrollo de dichas aplicaciones implica enfrentar múltiples desafíos técnicos y de usabilidad que deben ser resueltos para ofrecer una experiencia de usuario satisfactoria.

### **Situación Actual y Necesidades:**

- **Diversidad de Interacciones:** Las redes sociales modernas permiten una amplia variedad de interacciones: registro y autenticación de usuarios, creación y edición de perfiles, establecimiento de relaciones de amistad, publicación de contenidos (texto e imágenes), comentarios y reacciones en forma de "likes". Cada uno de estos elementos implica la necesidad de gestionar datos de manera eficaz y con integridad.
- **Gestión de Datos Complejos:** La información a manejar no se reduce a simples textos; se requiere almacenar multimedia, relaciones complejas entre usuarios y un historial de interacciones que debe estar disponible de forma rápida y eficiente. Las bases de datos tradicionales pueden presentar limitaciones en términos de escalabilidad y flexibilidad frente a las demandas de una red social.
- **Rendimiento y Escalabilidad:** Con el crecimiento constante de la cantidad de usuarios, es imperativo que el sistema mantenga un rendimiento óptimo. La capacidad de escalar horizontal y verticalmente se vuelve crítica para soportar el aumento de tráfico y la interacción en tiempo real sin que se generen cuellos de botella en la experiencia del usuario.
- **Seguridad y Privacidad:** La gestión de la información personal y sensible de los usuarios exige altos estándares de seguridad. Desde el proceso de autenticación hasta la protección de datos almacenados, el sistema debe contar con mecanismos de



encriptación y control de accesos que garanticen la integridad y la confidencialidad de la información.

### **Definición del Problema:**

El problema principal radica en desarrollar una aplicación web de red social que permita la interacción dinámica y segura entre usuarios, gestionando de forma eficiente y escalable la información asociada a perfiles, publicaciones, comentarios, reacciones y relaciones de amistad. Esto implica la necesidad de:

- Diseñar una arquitectura que integre un back-end robusto, basado en una base de datos NoSQL (MongoDB), y un front-end intuitivo y responsive.
- Garantizar que la aplicación maneje de manera óptima y segura grandes volúmenes de datos y transacciones en tiempo real.
- Implementar mecanismos de autenticación y autorización que resguarden la privacidad de los usuarios, asegurando la protección contra accesos no autorizados y vulnerabilidades.
- Incorporar funcionalidades adicionales (como geolocalización y mensajería en tiempo real) que respondan a las expectativas de los usuarios modernos y potencien la interacción en la red social.



### 1.3.OBJETIVOS GENERALES

Desarrollar una aplicación web funcional que simule una red social, donde los usuarios puedan registrarse, crear perfiles, establecer relaciones de amistad, realizar publicaciones, comentar, dar likes y recibir notificaciones, utilizando MongoDB como base de datos NoSQL para el almacenamiento eficiente y estructurado de la información.

### 1.4.OBJETIVOS ESPECÍFICOS

- Implementar un sistema de registro y autenticación de usuarios con validación de datos.
- Diseñar perfiles personalizados que permitan a los usuarios editar su información y subir fotos.
- Crear un sistema de solicitudes de amistad con opciones para aceptar o rechazar.
- Desarrollar funcionalidades para que los usuarios puedan realizar publicaciones con texto e imágenes.
- Integrar la posibilidad de comentar en publicaciones y reaccionar con “me gusta”.
- Configurar MongoDB para almacenar y recuperar eficientemente datos como usuarios, relaciones, publicaciones y reacciones.
- Añadir un sistema de notificaciones para informar a los usuarios sobre interacciones relevantes.

### 1.5.JUSTIFICACIÓN

Este proyecto representa una oportunidad ideal para aplicar los conocimientos adquiridos en la asignatura de Base de Datos No Relacionales en un contexto práctico y realista. Desarrollar una red social desde cero me permite sumergirme en un entorno donde se requiere comprender estructuras de datos dinámicas, relaciones complejas entre entidades y necesidades reales de escalabilidad, como las que enfrentan muchas aplicaciones actuales.

Además, trabajar con MongoDB nos permite aprovechar las ventajas de las bases de datos NoSQL: flexibilidad, rapidez en el acceso a los datos y una gran capacidad de adaptación a nuevos requerimientos sin necesidad de redefinir toda la estructura.

Esta iniciativa se vuelve relevante por los siguientes motivos:

- **Innovación y Actualización Tecnológica:** El proyecto propone la integración de tecnologías modernas como MongoDB para el manejo de bases de datos NoSQL y frameworks avanzados en el desarrollo del front-end, lo que garantiza una arquitectura robusta y una experiencia de usuario optimizada. Esto no solo representa un reto técnico, sino también una oportunidad para explorar soluciones innovadoras en el diseño de aplicaciones escalables.
- **Seguridad y Confianza:** En un contexto donde la protección de datos personales es primordial, la aplicación aborda la implementación de sistemas de autenticación y autorización que cumplen con altos estándares de seguridad. Esto es esencial para fomentar la confianza entre los usuarios y prevenir vulnerabilidades que puedan comprometer la información sensible.
- **Mejora de la Comunicación y Conectividad:** Al proporcionar herramientas para la creación de perfiles, relaciones de amistad, publicaciones, comentarios y notificaciones en tiempo real, la aplicación facilita la comunicación y la conexión entre personas de forma más interactiva y dinámica, respondiendo a las necesidades actuales de interacción social en entornos digitales.
- **Escalabilidad y Rendimiento:** La arquitectura propuesta contempla el crecimiento continuo en la cantidad de usuarios y datos, asegurando que el sistema mantenga un rendimiento óptimo a pesar de la carga creciente. Esto es crucial para ofrecer una experiencia consistente y confiable en el largo plazo.

En resumen, este proyecto no solo se orienta a resolver desafíos técnicos y funcionales en el desarrollo de redes sociales, sino que también contribuye a la formación y el avance en la integración de tecnologías emergentes. Además, promueve la creación de entornos digitales seguros y eficientes, fundamentales en la actualidad para fomentar una comunicación efectiva y protegida en la red.

## **2. CAPÍTULO II**

### **2.1.MARCO TEÓRICO**

#### **Bases de Datos Relacionales y No Relacionales.**

Las bases de datos relacionales están basadas en un modelo estructurado de tablas que contienen filas y columnas. Cada fila representa un registro, y cada columna representa un campo de ese registro. Estas bases, como MySQL o PostgreSQL, se fundamentan en el uso del lenguaje SQL para realizar consultas. Su principal fortaleza es la integridad referencial entre tablas, pero su rigidez en la estructura puede dificultar adaptarse a proyectos con requerimientos cambiantes.

Por otro lado, las bases de datos no relacionales, como MongoDB, no utilizan tablas, sino que almacenan la información en documentos en formato JSON o BSON. Esto permite mayor flexibilidad, ya que cada documento puede tener su propia estructura, algo que es muy útil cuando se manejan datos heterogéneos, como perfiles de usuario, publicaciones, imágenes, etc. Además, son más escalables horizontalmente y permiten trabajar con grandes volúmenes de datos de manera eficiente.

#### **Operadores de comparación y operadores lógicos en MongoDB.**

MongoDB ofrece operadores que permiten realizar consultas específicas sobre los documentos:

- **Operadores de comparación:** \$eq (igual), \$ne (distinto), \$gt (mayor que), \$lt (menor que), \$gte (mayor o igual), \$lte (menor o igual).
- **Operadores lógicos:** \$and, \$or, \$not, \$nor. Permiten combinar múltiples condiciones dentro de una misma consulta.

### Operaciones CRUD en MongoDB

CRUD significa: Create, Read, Update y Delete, es decir, crear, leer, actualizar y eliminar documentos. En MongoDB se ejecutan así:

- **insertOne() o insertMany()** para crear documentos.
- **find()** para leer.
- **updateOne(), updateMany() o \$set** para modificar.
- **deleteOne() o deleteMany()** para eliminar.

### Agregaciones en MongoDB

Las agregaciones permiten procesar datos de manera avanzada. Por ejemplo, se pueden contar likes por publicación, agrupar comentarios por usuario, o generar estadísticas de actividad.

Se usa el método `.aggregate()` junto con diferentes etapas como:

- **\$match:** filtra documentos por condiciones.
- **\$group:** agrupa documentos y permite sumar, contar o promediar valores.
- **\$sort:** ordena resultados.

- **\$limit:** limita la cantidad de resultados.
- **\$project:** muestra o esconde campos específicos de la salida.

**Función \$cond en MongoDB:** \$cond es un operador condicional que funciona como un if-else. Permite evaluar condiciones dentro de una agregación.

**Índices en MongoDB:** Un índice en MongoDB mejora la velocidad de las consultas sobre campos específicos. Son parecidos a los índices en libros: permiten encontrar más rápido lo que se busca sin tener que leer todo.

**Backups y restauraciones en MongoDB:** Es importante poder respaldar la base de datos para evitar pérdidas de información.

MongoDB permite hacer copias de seguridad con mongodump y restaurarlas con mongorestore.

Estas herramientas son fundamentales para mantener la integridad del sistema y recuperarse de fallos.

## MongoDB y Atlas

MongoDB Atlas es la plataforma en la nube de MongoDB. Permite crear, administrar y escalar bases de datos en línea de forma sencilla. Ofrece paneles gráficos, alertas, y backups automáticos. Para este proyecto, se puede usar Atlas para simular un entorno de producción real y permitir que la aplicación esté disponible desde cualquier lugar.

## Gestión de usuarios en MongoDB

MongoDB permite manejar la seguridad mediante la creación de usuarios con diferentes roles. Esto es clave en proyectos colaborativos o donde se necesita restringir el acceso.

### 3. CAPÍTULO III

#### 3.1. DESARROLLO Y/O IMPLEMENTACIÓN

Para esta red social se utilizó MongoDB como sistema de base de datos no relacional. Mongo permite el almacenamiento flexible de documentos JSON, ideal para manejar estructuras dinámicas como usuarios con notificaciones, publicaciones con comentarios y chócale.

##### Modelo de datos con Mongo:

##### Los principales modelos creados fueron:

- **Usuario:**
  - Incluye campos como nombre, correo, contraseña, biografía, fecha creación y foto de perfil.
  - Campos amigos, solicitudes y notificaciones.
  - Permite estructurar la red social en forma de nodos conectados

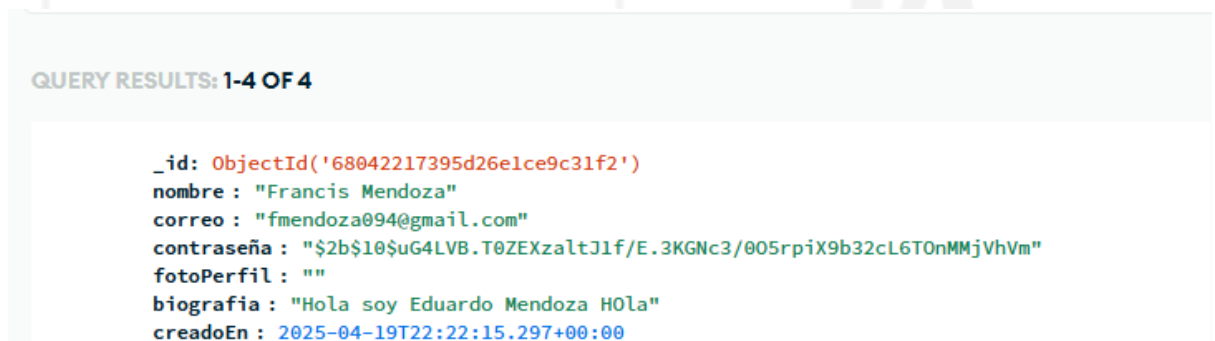


Ilustración 1 Usuarios

- **Publicaciones:**

- Contiene texto, imagen, autor, likes y un arreglo de comentarios.
- Los comentarios están como subdocumentos y almacena textos, usuario autor fecha y contador de likes.



Ilustración 2 Publicaciones



## Funciones CRUD utilizadas.

### CREATE (Crear)

- Registrar usuarios.
- Publicar imágenes y textos.
- Agregar comentarios a publicaciones.
- Enviar solicitudes de amistad.
- Crear notificaciones.

### READ (Leer o Consultar)

- Mostrar publicaciones de todos los usuarios.
- Consultar lista de amigos y solicitudes pendientes.
- Buscar usuarios por nombre o correo electrónico.
- Ver notificaciones del usuario.

### UPDATE (Actualizar)

- Dar like a publicaciones y comentarios.
- Aceptar o rechazar solicitudes de amistad.
- Editar el perfil del usuario (nombre, foto, biografía).
- Marcar notificaciones como vistas.

## Agregación de datos en MongoDB

En este proyecto se aplicó la agregación de datos principalmente mediante el uso del método `.populate()` de Mongoose, una herramienta clave cuando se manejan relaciones entre colecciones en MongoDB.

### ¿Qué se logró con `.populate()`?

Se logró enriquecer la información que se mostraba al usuario al vincular automáticamente referencias entre documentos. Por ejemplo:

- Cuando se muestra una publicación, se usó `.populate('autor', 'nombre fotoPerfil')` para que en vez de mostrar solo el ID del autor, aparezca directamente su nombre y su foto.
- En los comentarios de las publicaciones, también se aplicó `.populate('comentarios.usuario', 'nombre')`, lo cual permite visualizar quién comentó sin necesidad de hacer múltiples consultas por separado.

#### **Beneficios de esta implementación:**

- Mejor experiencia para el usuario final: se muestran nombres reales y fotos en vez de identificadores técnicos.
- Menos código en el frontend: ya se reciben los datos completos, listos para ser renderizados.
- Optimización de consultas: al evitar múltiples llamadas a la base de datos por cada usuario relacionado

## **4. CAPÍTULO IV**

### **4.1. ANÁLISIS DE RESULTADOS**

Durante el desarrollo de este proyecto, se logró crear una red social funcional en la que las personas pueden registrarse, subir su foto, escribir publicaciones, dar "chócale" (like), comentar, tener amigos y recibir notificaciones. Todo esto fue posible gracias al uso de **MongoDB**, que nos permitió manejar los datos de una manera muy flexible, sin necesidad de complicarnos con estructuras demasiado rígidas como en otras bases de datos.

Una de las cosas más importantes fue que logramos organizar los datos de una manera que realmente se ajusta al funcionamiento natural de una red social. Por ejemplo, los comentarios están guardados dentro de cada publicación, lo cual hace que sea más fácil mostrarlos y mantener todo relacionado.

### **Usuarios:**

Desde el inicio los usuarios pueden registrarse, iniciar sesión, editar su perfil y subir una imagen de perfil. Todo esto se guarda de manera sencilla en MongoDB. Además, se protegieron los datos para que solo el dueño pueda hacer cambios.

### **Publicaciones:**

Los usuarios pueden compartir lo que piensan o sienten, escribiendo publicaciones y agregando imágenes si desean. Estas publicaciones aparecen en orden, mostrando también quién la escribió, su nombre y su foto.

### **Publicaciones:**

Los usuarios pueden compartir lo que piensan o sienten, escribiendo publicaciones y agregando imágenes si desean. Estas publicaciones aparecen en orden, mostrando también quién la escribió, su nombre y su foto.

### **Notificaciones:**

Cada vez que alguien comenta o le da like a tu publicación, te llega una notificación. Estas aparecen en una lista especial, y también se muestra cuántas tienes sin leer, como lo hacen las redes sociales reales. Al entrar a revisarlas, se marcan como "vistas".

### **Seguridad:**

Solo los usuarios que han iniciado sesión pueden hacer publicaciones, comentar o ver su perfil. Esto ayuda a mantener un buen control de lo que se hace en la red y evitar que personas no autorizadas usen funciones importantes.

## 5. CAPÍTULO V

### 5.1. CONCLUSIONES

- **Aprendí mucho sobre cómo funciona una red social por dentro.** Antes pensaba que era solo tener un botón para dar "me gusta" o dejar un comentario, pero ahora entiendo todo lo que pasa detrás: cómo se guardan los datos, cómo se muestran, cómo se conectan entre sí y cómo se protegen.
- **MongoDB me pareció muy práctico y flexible.** No tuve que complicarme creando muchas tablas o relaciones raras. Pude guardar la información como yo quería, de forma directa y clara, y eso me ayudó a avanzar más rápido.
- **Hacer este proyecto me enseñó a pensar como usuario y como desarrollador al mismo tiempo.** Me puse en los zapatos de quien va a usar la red social: quería que todo se vea bien, que funcione sin errores y que sea fácil de usar. Eso me ayudó a mejorar los detalles y a organizarme mejor.
- **Aprendí que no todo es solo programar, también hay que planificar.** Fue clave ir paso a paso, primero pensando qué quería que hiciera la aplicación, luego cómo hacerlo y por último ir probando que todo funcione. Me equivoqué varias veces, pero eso me ayudó a entender mejor cada parte.
- **Pude aplicar varias cosas que vi en clase y también investigar por mi cuenta.** No todo estaba en los apuntes, así que tuve que buscar soluciones, probar códigos y corregir errores. Eso me dio más confianza en mí mismo para seguir aprendiendo.

### 5.2. RECOMENDACIONES

- A partir de todo lo que pude construir y probar en este proyecto, recomendaría seguir

desarrollando este tipo de plataformas sociales como espacios de práctica para estudiantes. Usar herramientas como MongoDB y Node.js ayuda bastante a entender cómo funcionan las aplicaciones reales por dentro. Además, incluir más funcionalidades como mensajes privados o reacciones diferentes podría hacer el proyecto aún más completo en futuras versiones.

- Una mejora importante sería usar una base de datos separada exclusivamente para las notificaciones. Esto haría que el sistema esté más ordenado y que escale mejor si se usara con muchos usuarios. También sería bueno implementar validaciones más estrictas en los formularios y mejorar el diseño visual para hacerlo más amigable, sobre todo si se pensara usar esta app con personas externas al entorno académico.
- Este proyecto no solo sirve como entrega de una materia. También podría usarse como base para enseñar cómo funciona el backend de una red social en talleres o clases. Incluso se podría adaptar como una red interna para instituciones educativas, donde los estudiantes publiquen novedades, compartan ideas o reciban notificaciones de sus profesores. Es una buena forma de aplicar conocimientos de MongoDB, Express y autenticación en proyectos reales.

## 6. ANEXOS

Link Repositorio GitHub:

[https://github.com/Eduardomendoza31/proyecto\\_redsocial](https://github.com/Eduardomendoza31/proyecto_redsocial)

Link red social:

<http://localhost:3000/>

### Imágenes

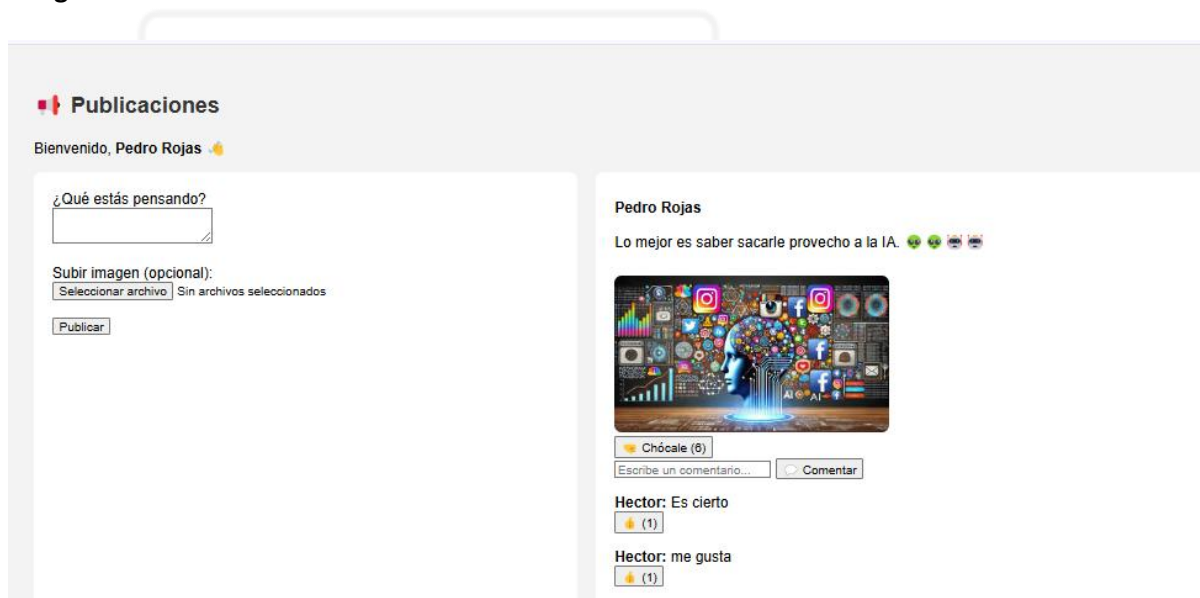


Ilustración 3 Feed de publicaciones

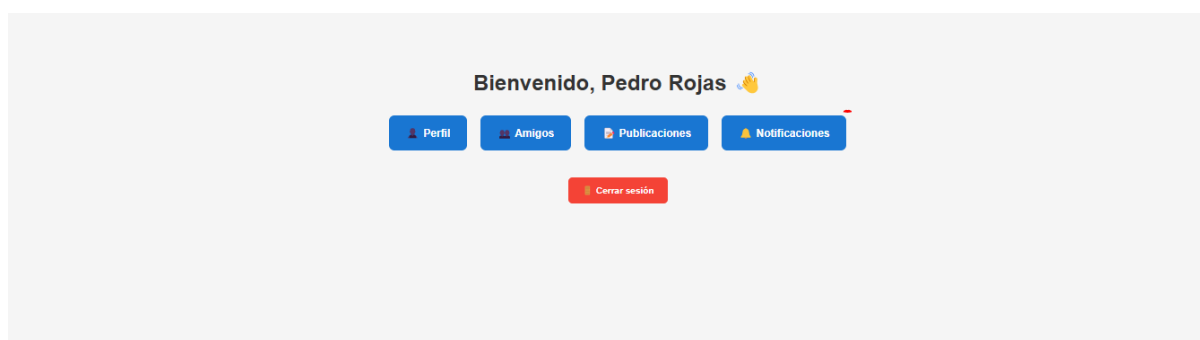


Ilustración 4 Feed de bienvenida

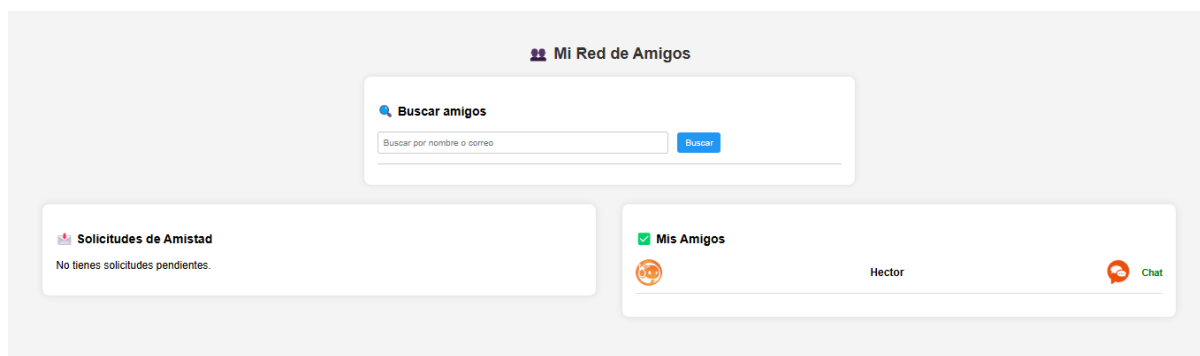


Ilustración 5 Feed de solicitudes

## 7. BIBLIOGRAFÍA

MongoDB, Inc. (s.f.). *¿Qué es MongoDB?* Recuperado el 19 de abril de 2025, de <https://www.mongodb.com/es/company/what-is-mongodb>

Microsoft. (s.f.). *Datos no relacionales y NoSQL*. Recuperado el 21 de abril de 2025, de <https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>

Platzi. (s.f.). *Cosas básicas de un CRUD en MongoDB*. Recuperado el 22 de abril de 2025, de <https://platzi.com/tutoriales/1533-mongodb-basico/4102-cosas-basicas-de-un-crud-en-mongodb/>

MongoDB, Inc. (s.f.). *MongoDB Atlas Database*. Recuperado el 17 de abril de 2025, de <https://www.mongodb.com/products/platform/atlas-database>

Desarrollador Web. (s.f.). *Cómo conectar MongoDB con Visual Studio Code* [Video]. YouTube. <https://www.youtube.com/watch?v=IEV7gNJeNJI>