

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARIA
SEDE CONCEPCION – REY BALDUINO DE BÉLGICA

RED SOCIAL MEAN STACK

Trabajo de Titulación para optar al Título
Profesional Técnico Universitario en
INFORMÁTICA

Alumno:

Andrés Mardones Otárola

Profesor Guía:

Jorge Urrutia Delucchi

2018

DEDICATORIA

A mis padres, Virginia y Juan, abuelos, Darío e Isabel, quienes aportaron en gran parte a mi formación como persona, regalándome sus conocimientos y experiencias, a mis hermanos y los momentos que junto a ellos he forjado. A María, por los conocimientos, la motivación, la alegría y el apoyo brindado como colega, amiga y polola. A Jonathan, por los conocimientos compartidos y la gran amistad formada. Agradecer también a mi profesor, jefe de carrera y profesor guía Don Jorge Urrutia Delucchi, por su ayuda y guía en la universidad. Y a todos mis cercanos y, quienes hayan depositado en mí su confianza y buenas vibras.

RESUMEN

KEYWORDS: API, API Rest, MEAN Stack, Red Social, Prototipo, JavaScript, TypeScript, ECMAScript, ES6, Angular 7, MongoDB, Express, NodeJS, SPA, Front-End, Back-end, Framework, Bootstrap 4, DOM, JSON, BSON, MCV, NoSQL, JQuery, AJAX, CSS3, HTML5, Web Design, WebApps, aplicación web.

El presente trabajo de título consiste en el desarrollo y prototipado de una red social interactiva en base a MEAN Stack.

Para la realización de este sistema, fue necesario estudiar las características, comportamiento de las redes sociales actuales, las modas e intereses de las comunidades que existen actualmente. También requirió conocimiento sobre aplicaciones webs, desarrollo de APIs y diseño web, nivel avanzado del lenguaje JavaScript, TypeScript y ECMAScript.

El desarrollo del proyecto será totalmente enfocado a construir una API, la que será la base de la red social. Esta API, comunicará el Front-end, con la base de datos, mediante el sistema REST, peticiones HTTP y la información en formato JSON.

ÍNDICE

DEDICATORIA..... 1

RESUMEN..... 2

ÍNDICE 3

ÍNDICE DE TABLA..... 5

ÍNDICE DE ILUSTRACIONES..... 5

ÍNDICE DE DIAGRAMAS..... 5

SIGLAS Y SIMBOLOGÍA 6

INTRODUCCIÓN 7

CAPÍTULO 1.- ANÁLISIS 8

1.1- CONTEXTO DEL PROBLEMA..... 9

1.2- OBJETIVOS GENERALES 10

1.3- OBJETIVOS ESPECÍFICOS..... 10

2.- REQUERIMIENTOS..... 11

REQUERIMIENTOS FUNCIONALES 12

REQUERIMIENTOS NO FUNCIONALES 15

3.- CASOS DE USO 16

4.- CASOS DE USO EXTENDIDOS..... 17

CASO DE USO: VALIDACIÓN 18

CASO DE USO: VER PUBLICACIONES 19

CASO DE USO: VER PERFIL DE USUARIO 20

CASO DE USO: GESTIONAR PERFIL..... 21

CASO DE USO: GESTIONAR PUBLICACIONES. 22

CASO DE USO: EXPLORAR GENTE. 23

CASO DE USO: SEGUIR / DEJAR DE SEGUIR..... 24

5.- BASE DE DATOS PROTOTIPO..... 25

DEFINICIÓN 25

TIPOS..... 25

DISEÑO 26

DESCRIPCIÓN DE TABLAS..... 27

6.- VISTA DEL PROYECTO 28

INICIO DE SESIÓN 28

SECCIÓN DE REGISTRO..... 29

SECCIÓN DE PUBLICACIONES..... 30

PERFIL DE USUARIO 31

PERFIL DE OTRO USUARIO 32

7.- CONCLUSIONES Y RECOMENDACIONES 33

8.- BIBLIOGRAFÍA 34

ÍNDICE DE TABLA

- Tabla 2-1. Requerimiento N°1
- Tabla 2-2. Requerimiento N°2
- Tabla 2-3. Requerimiento N°3
- Tabla 2-4. Requerimiento N°4
- Tabla 2-5. Requerimiento N°5
- Tabla 2-6. Requerimiento N°6
- Tabla 2-7. Requerimiento N°7
- Tabla 2-8. Requerimiento N°8
- Tabla 4-1. Caso de uso extendido: Validación.
- Tabla 4-2. Caso de uso extendido: Ver publicaciones.
- Tabla 4-3. Caso de uso extendido: Ver perfil de usuario.
- Tabla 4-4. Caso de uso extendido: Gestionar perfil.
- Tabla 4-5. Caso de uso extendido: Gestionar publicaciones.
- Tabla 4-6. Caso de uso extendido: Explorar gente.
- Tabla 4-7. Caso de uso extendido: Seguir / Dejar de seguir.

ÍNDICE DE ILUSTRACIONES

- Ilustración 6-1. Inicio de Sesión
- Ilustración 6-2. Sección de registro.
- Ilustración 6-3. Sección de publicaciones.
- Ilustración 6-4. Perfil de usuario.
- Ilustración 6-5. Perfil de otro usuario.

ÍNDICE DE DIAGRAMAS

- Diagrama 3-1. Casos de uso

SIGLAS Y SIMBOLOGÍA

- **Mean Stack:** Conjunto de tecnologías basadas en JavaScript (**M**ongoDB, **E**xpress, **A**ngularJS & **N**odeJS).
- **API:** Herramienta de desarrollo en la que se facilita un conjunto de subrutinas, funciones y procedimientos usados para desarrollar un sistema, reutilizar código y facilitar la comunicación de cierto sistema, ya sea entre sus componentes o con agentes externos.
- **API Rest:** Rest es una forma simple y funcional de organizar las interacciones Front-end y Back-end dentro de un sistema y fuera de este. Es un método inspirado en HTTP, permitiendo interacciones con la web, aunque no está vinculado a esta.
- **TypeScript:** Es un lenguaje de programación libre y de código abierto, un superconjunto de JavaScript.
- **ECMAScript:** Una especificación estándar de lenguaje de programación empleado en JavaScript.
- **Front-end:** El desarrollo del frontend, se centra en el usuario, desarrollar el ambiente con el cual el usuario interactúa directamente.
- **Back-end:** El desarrollo de backend, es el manejo y procesamiento de los datos del sistema o aplicación, de igual manera se encarga de comunicar el sistema con la base de datos.

INTRODUCCIÓN

Los seres humanos tenemos necesidades generales básicas, tanto **fisiológicas** como **psicológicas**. Algunas como la afiliación, la que agrupa necesidades como la estima, amistad, intimidad, entre otras. Estas necesidades manifiestan lo social que es el ser humano naturalmente, en su esencia.

Muchos psicólogos y especialistas han postulado su visión acerca de lo que a su parecer es la necesidad social o de afiliación, algunos como **Murray, Geen, A. Maslow**, entre otros. **Todos concuerdan en lo necesario** que es para el ser humano, el estar comunicado, sentirse parte de una comunidad, y ser reconocido por **ser y pertenecer**.

Respecto a lo anteriormente mencionado y con ayuda del avance tecnológico surgieron herramientas para satisfacer estas necesidades, herramientas como **las redes sociales**, las cuales son un mundo virtual en la que cada persona es dueña de su espacio, mostrando lo que quiere de sí, compartiendo su vida y la de otros.

Existen tantas cosas buenas, como cosas malas respecto a las redes sociales. En el presente informe, se desarrollarán ideas personales, ya que la gran variedad de opiniones, relatos, estudios y temas, nos dan distintos focos respecto a las redes sociales. Algunos temas giran en torno a ejes centrales, como son la seguridad, la psicología, la comunicación, educación, entre otros.

CAPÍTULO 1.- ANÁLISIS

1.1- CONTEXTO DEL PROBLEMA

En el mundo de las redes sociales, el comportamiento de cada individuo varía según su personalidad, cada uno tiene la opción de valorar lo que expresan los otros en forma de admiración o crítica.

Las redes sociales no controlan el cómo hacerlo, no premian la buena participación, y los deseos de aportar a esta, por otro lado, tampoco suelen haber métodos de “sancionar” los aportes destructivos de cada usuario, ya sea críticas ofensivas, cyberbullying, daño a la imagen, entre otros.

Así también las redes sociales no están desarrolladas para incluir toda la comunidad, ya que se centran en el aporte a la juventud, dejando de lado a la tercera edad, que no se adapta fácilmente a los cambios tecnológicos ni a interfaces complejas.

1.2- OBJETIVOS GENERALES

- Desarrollar un prototipo de una red social, con sistema modulado, al que se puedan agregar nuevos módulos a la red social, mejorando o ampliando esta misma.

1.3- OBJETIVOS ESPECÍFICOS

- Desarrollar una red social en la que el usuario pueda crear y compartir textos, imágenes.
- Diseñar y generar una interfaz minimalista y atractiva.

2.- REQUERIMIENTOS

REQUERIMIENTOS FUNCIONALES

Tabla 2-1. Requerimiento N°1.

N° Requerimiento	1
Importancia	Alta
Interfaz	Inicio de sesión
Tipo	Acceso
Descripción	El usuario ingresa sus datos, para que el sistema lo valide.
Actor	Usuario

Fuente: Elaboración propia.

Tabla 2-2. Requerimiento N°2.

N° Requerimiento	2
Importancia	Alta
Interfaz	Registro
Tipo	Acceso
Descripción	El usuario debe poder registrarse en el caso de no tener una cuenta.
Actor	Usuario

Fuente: Elaboración propia.

Tabla 2-3. Requerimiento N°3.

N° Requerimiento	3
Importancia	Alta
Interfaz	Perfil de usuario
Tipo	Gestión
Descripción	El usuario dispondrá de una sección de perfil, donde pueda gestionar los datos de su cuenta, así como sus seguidores y seguidos.
Actor	Usuario

Fuente: Elaboración propia.

Tabla 2-4. Requerimiento N°4.

N° Requerimiento	4
Importancia	Alta
Interfaz	Publicaciones
Tipo	Interacción
Descripción	El usuario podrá ver, reaccionar a publicaciones que le interese.
Actor	Usuario

Fuente: Elaboración propia.

Tabla 2-5. Requerimiento N°5.

N° Requerimiento	5
Importancia	Alta
Interfaz	Perfiles de otros usuarios
Tipo	Interacción
Descripción	El usuario podrá desmarcar la opción para que las publicaciones de un usuario aparezcan en la sección de publicaciones.
Actor	Usuario

Fuente: Elaboración propia.

Tabla 2-6. Requerimiento N°6.

N° Requerimiento	6
Importancia	Alta
Interfaz	Perfil
Tipo	Interacción
Descripción	El usuario podrá crear y gestionar publicaciones.
Actor	Usuario

Fuente: Elaboración propia.

REQUERIMIENTOS NO FUNCIONALES

Tabla 2-7. Requerimiento N°7.

N° Requerimiento	7
Importancia	Alta
Interfaz	Todas las interfaces
Tipo	Diseño
Descripción	La plataforma debe ser responsive, minimalista e intuitiva
Actor	Desarrollador

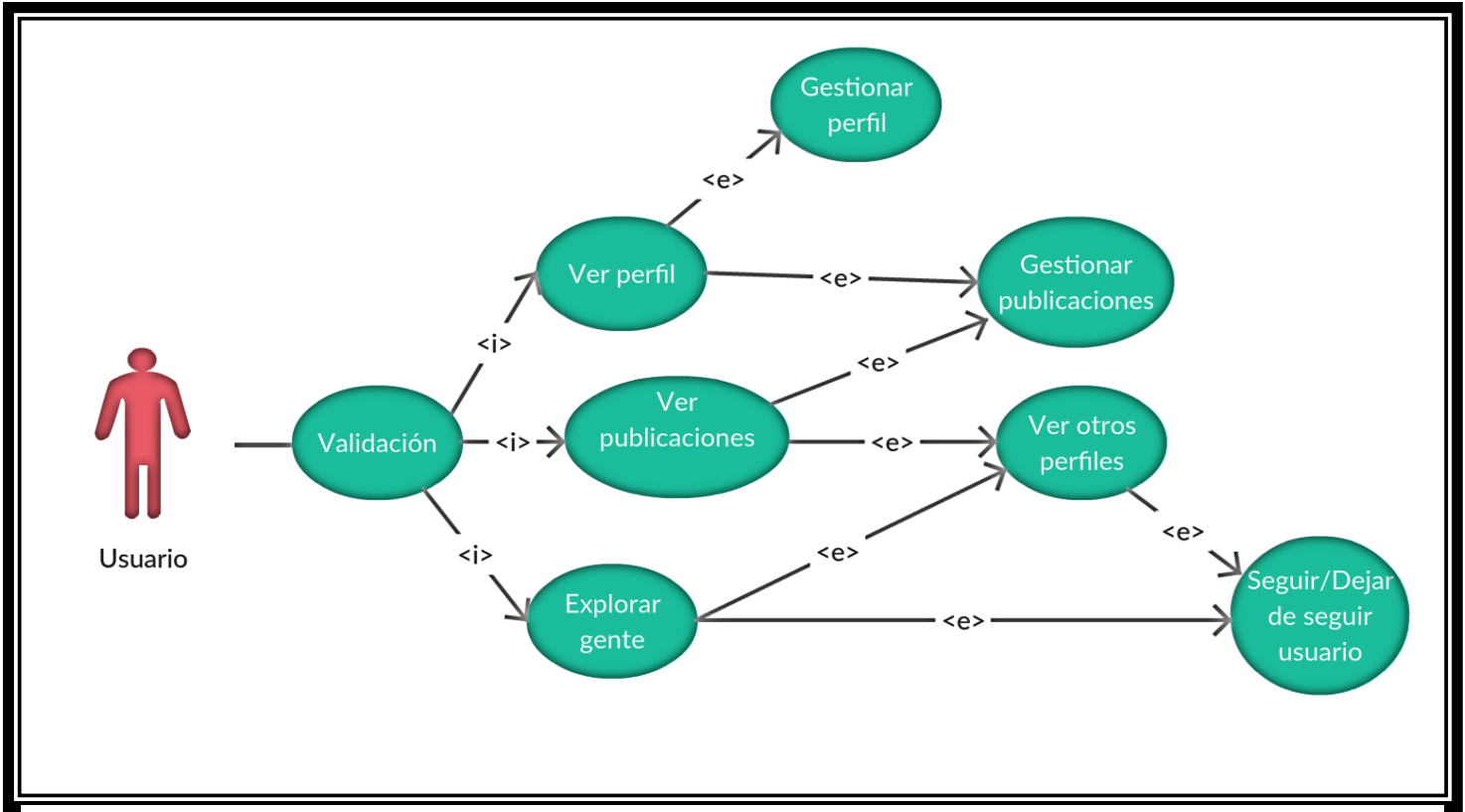
Fuente: Elaboración propia.

Tabla 2-8. Requerimiento N°8.

N° Requerimiento	8
Importancia	Alta
Interfaz	Perfil
Tipo	Alcance
Descripción	Debe actuar de manera eficiente en modo mobil, tablet y escritorio.
Actor	Desarrollador

Fuente: Elaboración propia.

3.- CASOS DE USO



Fuente: Elaboración propia.

Diagrama 3-1. Diagrama de casos de uso.

4.- CASOS DE USO EXTENDIDOS

CASO DE USO: VALIDACIÓN

Actores: Usuario, sistema

Precondiciones: El usuario debe abrir la aplicación o ingresar a la página web.

Descripción del flujo: El sistema comprueba si ha iniciado sesión antes o no, en caso de que no haya registros de sesión en caché, redirigirá al formulario de inicio de sesión.

Postcondición: En el caso de que haya registros de sesión, redirigir al inicio de la aplicación. En caso contrario, el usuario deberá rellenar el formulario de inicio de sesión o registro.

Tabla 4-1. Validación.

USUARIO	SISTEMA
1- El usuario abre la aplicación.	
	2- El sistema comprueba en el caché si existen datos de inicio de sesión.
	3- El sistema redirige al inicio en caso de que existan datos. Si no, redirigirá a la página de login.

Fuente: Elaboración propia.

CASO DE USO: VER PUBLICACIONES

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema.

Descripción del flujo: El sistema muestra la pantalla de inicio, en la que se encuentran listadas las publicaciones y el menú principal.

Postcondición: -

Tabla 4-2. Ver publicaciones.

USUARIO	SISTEMA
1- El usuario selecciona la opción para ver las publicaciones, o bien es redirigido por el sistema.	
	2- El sistema devuelve los posts relacionados a los seguimientos del usuario
	3- El sistema espera la próxima acción del usuario.

Fuente: Elaboración propia.

CASO DE USO: VER PERFIL DE USUARIO

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema.

Descripción del flujo: El usuario debe seleccionar un enlace hacía el perfil de otro o de el mismo.

Postcondición: Si no es el perfil del usuario identificado, debe tener la opción para seguir o dejar de seguir a este usuario.

Tabla 4-3. Ver perfil de usuario.

USUARIO	SISTEMA
1- El usuario puede darle click a un enlace que dirige al perfil de determinado usuario.	
	2- El sistema muestra la página con la información del usuario seleccionado.

Fuente: Elaboración propia.

CASO DE USO: GESTIONAR PERFIL.

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema y debe ser el dueño del perfil.

Descripción del flujo: El usuario debe ingresar a su perfil, una vez dentro, puede gestionar los datos que desee.

Postcondición: El usuario enviará nueva información a la plataforma, actualizándose así su información para cualquier usuario de la red social

Tabla 4-4. Gestionar perfil.

USUARIO	SISTEMA
1- El usuario puede darle click a un enlace que dirige al perfil de determinado usuario.	
	2- El sistema muestra la página con la información del usuario seleccionado.
3- Si lo desea, puede editar la información que se encuentra en su perfil	
	4- Una vez editada la información el sistema guarda estos datos y lo actualiza en la base de datos.

Fuente: Elaboración propia.

CASO DE USO: GESTIONAR PUBLICACIONES.

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema y debe ser dueño de la publicación a gestionar.

Descripción del flujo: seleccionar la opción disponible para la publicación, sea editar o eliminar.

Postcondición: Se actualiza la información en la base de datos.

Tabla 4-5. Gestionar publicaciones.

USUARIO	SISTEMA
1- El usuario selecciona la publicación a gestionar y la acción que realizará.	
	2- El sistema actualiza la información dependiendo de la acción del usuario.

Fuente: Elaboración propia.

CASO DE USO: EXPLORAR GENTE.

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema.

Descripción del flujo: El usuario ingresará a una página, donde se encuentran listados los usuarios del sistema, en esta puede seguir o dejar de seguir a otro usuario.

Postcondición: -

Tabla 4-6. Explorar gente.

USUARIO	SISTEMA
1- El usuario ingresa a la página ‘Gente’, en el menú.	2- El sistema lista los usuarios, mostrándolos de manera ordenada y dando la opción seguir o dejar de seguir, según corresponda.

Fuente: Elaboración propia.

CASO DE USO: SEGUIR / DEJAR DE SEGUIR.

Actores: Usuario, sistema

Precondiciones: El usuario debe estar validado ante el sistema.

Descripción del flujo: El usuario puede seleccionar la opción seguir en el caso de que no siga al usuario seleccionado, de lo contrario, el usuario puede seleccionar la opción dejar de seguir.

Postcondición: En el caso de que haya una opción seleccionada, se deben actualizar los datos en la base de datos.

Tabla 4-7. Seguir/Dejar de seguir.

USUARIO	SISTEMA
	1- Identificar si el usuario seleccionado es seguido por el usuario identificado en la plataforma.
	2- Mostrar las opciones (seguir si el usuario no es seguido, dejar de seguir en caso contrario).
3- El usuario puede seleccionar una opción.	

Fuente: Elaboración propia.

5.- BASE DE DATOS PROTOTIPO

DEFINICIÓN

Desde el punto de vista informático, una base de datos puede ser definida como un conjunto de información estructurada que es almacenada para su posterior uso.

TIPOS

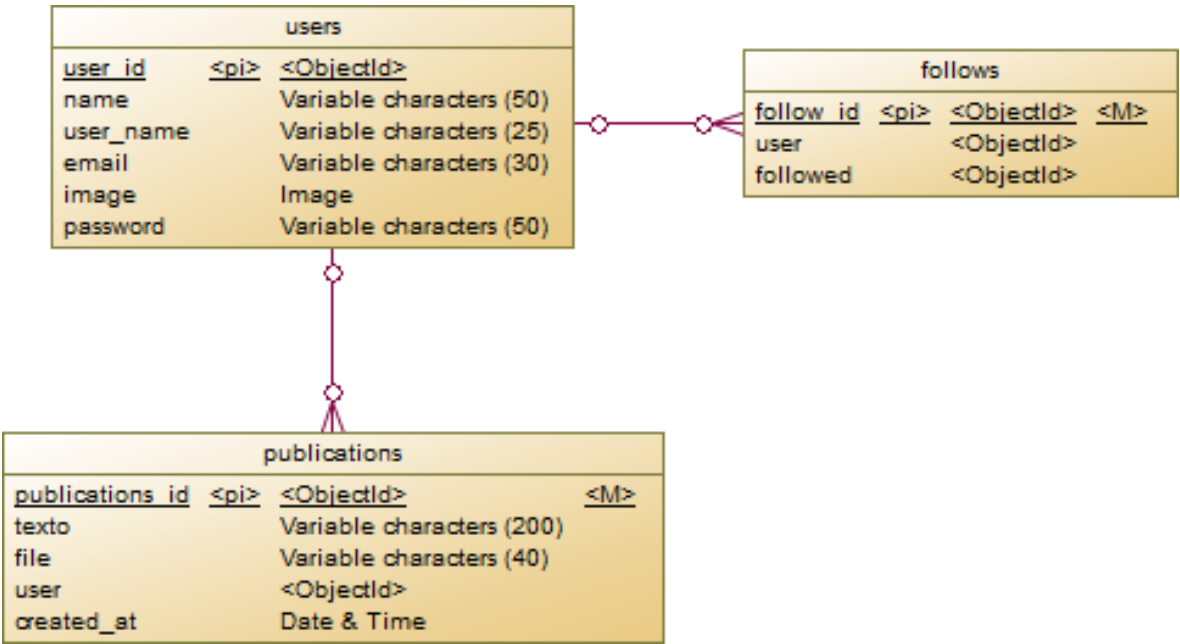
Según la forma en la que se ordena la información se pueden dividir en dos tipos, las bases de datos **relacionales** y **no relacionales**.

- **Bases de datos relacionales:** Se organiza la información en pequeños trozos que se relacionan los unos con los otros mediante la relación de identificadores.
- **Bases de datos no relacionales:** La información se organiza en documentos, y a diferencia de las bases de datos relacionales, en estas, los datos son independientes entre sí.

DISEÑO

La base de datos usada en el proyecto será no relacional. Por su rapidez y poco consumo en recursos, además de la gran cantidad de datos que puede almacenar, y la gran escalabilidad que tendrá este prototipo.

El esquema usado, se mostrará a continuación:



Fuente: elaboración propia.

Ilustración 5-1: Diseño de la base de datos

DESCRIPCIÓN DE TABLAS

Users: Esta colección almacena los principales datos del usuario.

- **user_id:** Representa el ID de usuario en la base de datos. Este lo genera automáticamente la base de datos. Es la clave primaria de esta colección.
- **name:** Este campo representa el nombre del usuario que será dueño de la cuenta.
- **user_name:** El valor user_name, es el nombre que tendrá el usuario en la red social, sea un nombre, un apodo, un nombre creado por el usuario, nombre de empresa, fundación, etc.
- **email:** Almacena el correo electrónico del usuario.
- **image:** Este campo almacena el nombre de la imagen de perfil perteneciente al usuario.
- **password:** Almacena la contraseña encriptada del usuario.

Follows: Esta colección almacena los seguimientos entre usuarios.

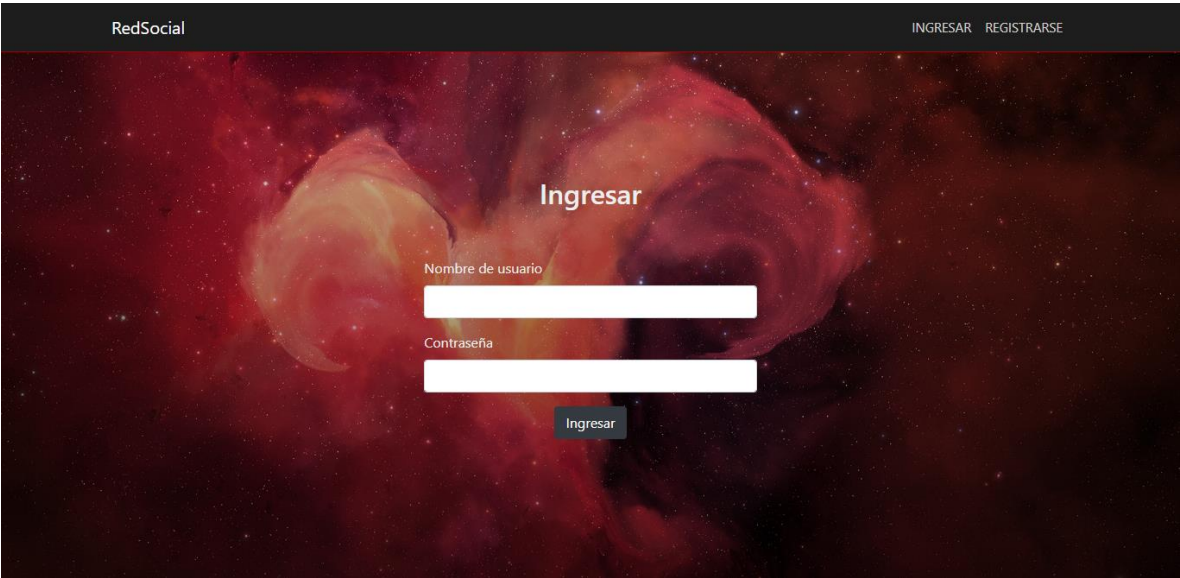
- **follow_id:** Almacena el ID del registro, es generado automáticamente por la base de datos y está asignada como clave primaria.
- **user:** Este campo almacena el ID del usuario que sigue a otro.
- **followed:** Almacena el ID del usuario seguido.

Publications: Esta colección almacena las publicaciones hechas por los usuarios.

- **publications_id:** Almacena el ID del registro y es la clave primaria, es generado por la base de datos.
- **texto:** Almacena el texto o descripción de la publicación.
- **file:** Almacena el nombre del archivo que va adjunto a la publicación.
- **user:** Este campo almacena el ID del usuario al que pertenece la publicación.
- **create_at:** Contiene la fecha y hora perteneciente al momento en el que fue ingresada la publicación.

6.- VISTA DEL PROYECTO

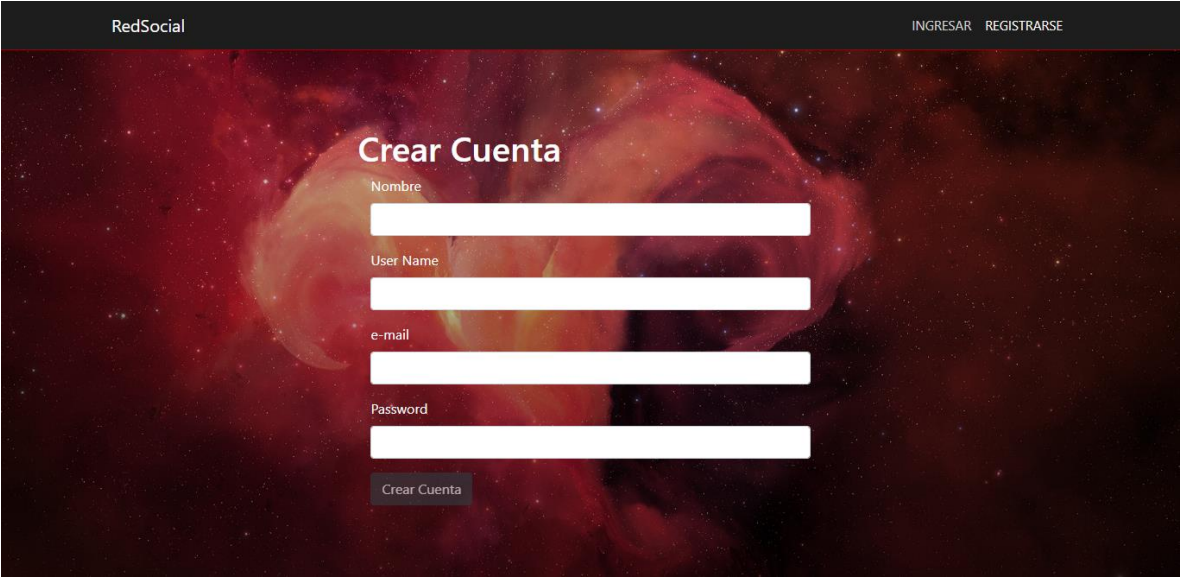
INICIO DE SESIÓN



Fuente: elaboración propia

Ilustración 6-1. Inicio de Sesión

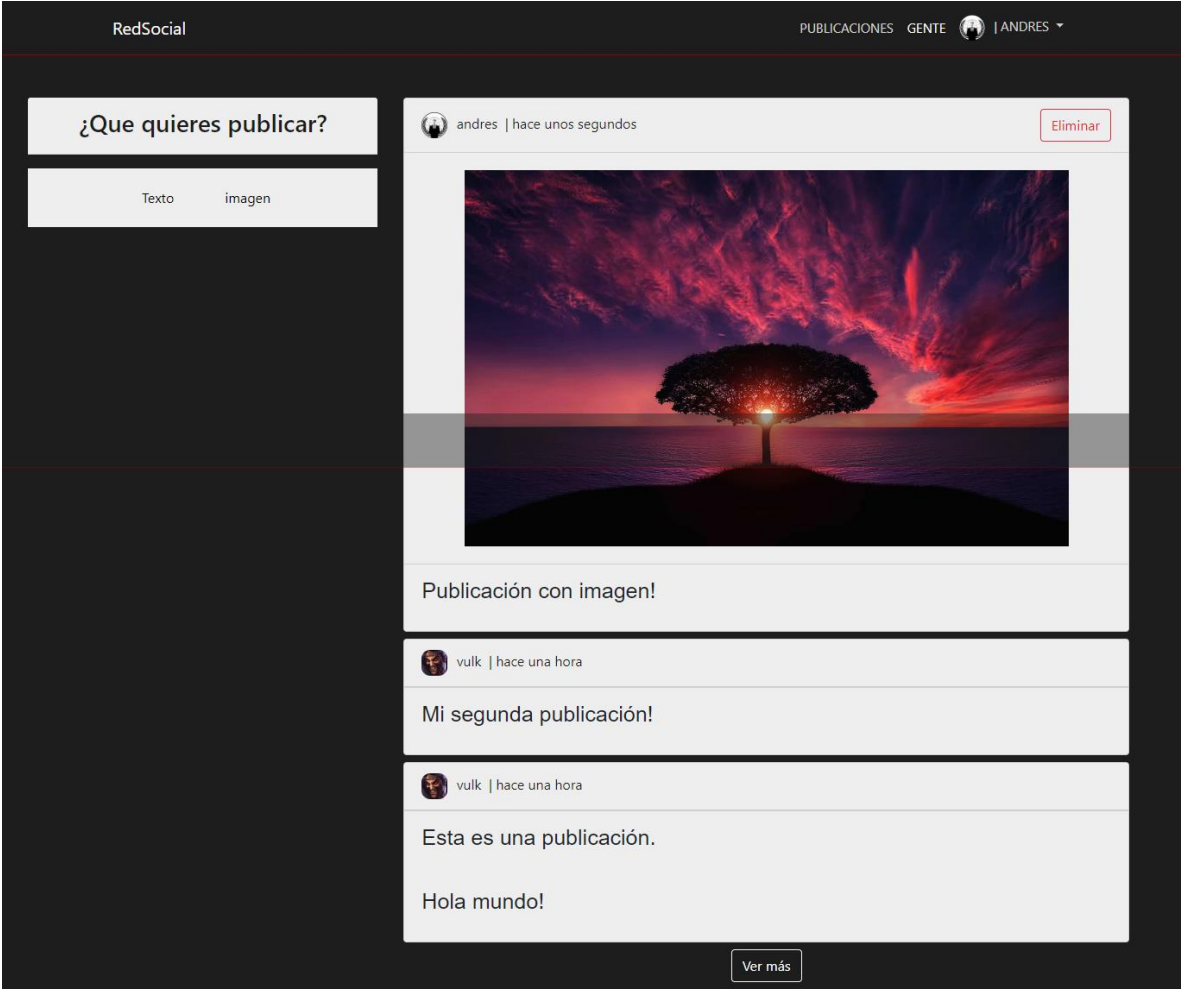
SECCIÓN DE REGISTRO



Fuente: elaboración propia

Ilustración 6-2 Sección de registro

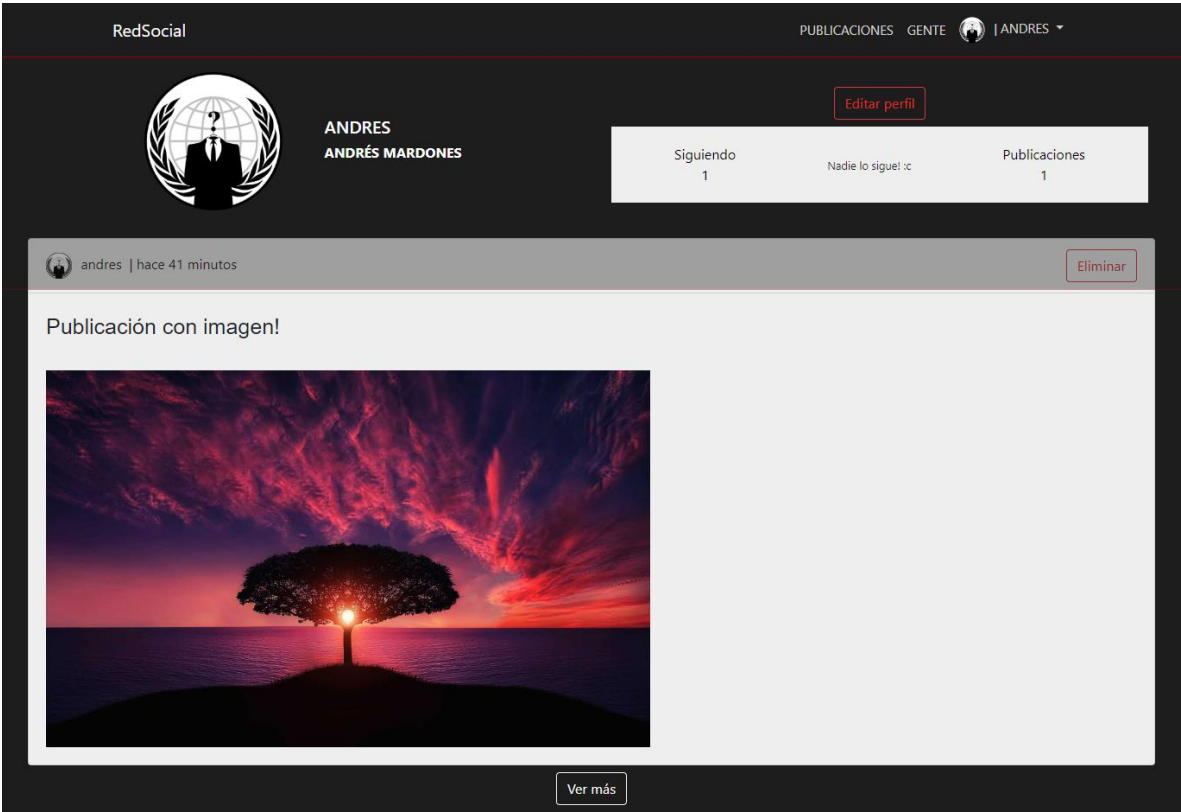
SECCIÓN DE PUBLICACIONES



Fuente: elaboracion propia

Ilustración 6-3. Página principal y publicaciones de usuarios seguidos

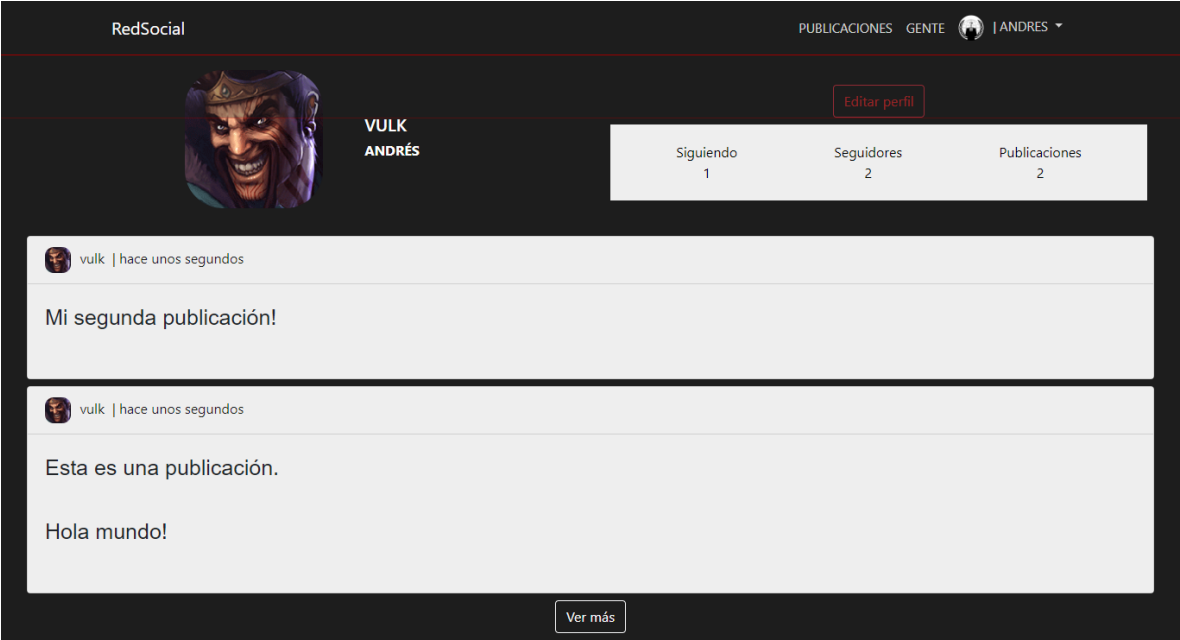
PERFIL DE USUARIO



Fuente: elaboración propia.

Ilustración 6-4. Perfil de usuario

PERFIL DE OTRO USUARIO



Fuente: elaboración propia.

Ilustración 6-5. Perfil de otro usuario

7.- CONCLUSIONES Y RECOMENDACIONES

El eje central del prototipo es la API, la construcción de esta trajo consigo varios desafíos ya que el nivel necesitado para programar fue muy elevado, así mismo el resultado final fue exitoso, el fruto es una API modulada y escalable, que podrá ser usada como base sólida para la construcción de un próximo proyecto, junto a todo lo aprendido, tanto en el área de lógica como en el diseño de las vistas de la aplicación.

La red social cumple con los requisitos mencionados anteriormente en el documento, las publicaciones de las imágenes se guardan de dos formas distintas, el texto en la base de datos junto a los nombres de los archivos, y los archivos se guardan con un nombre generado por el sistema en una carpeta, evitando así que la base de datos crezca más de lo necesario, por ende, haciendo más rápida la respuesta de las consultas y las modificaciones a esta.

8.- BIBLIOGRAFÍA

Páginas web:

- 1- Descarga de NodeJS. Disponible en: <https://nodejs.org/es/>.
- 2- Documentación de NodeJS. Disponible en: <https://nodejs.org/es/docs/>.
- 3- Descarga y documentación de ExpressJS Disponibles en: <https://expressjs.com/es/>.
- 4- Descarga de AngularJS: Disponible en: <https://angularjs.org/>.
- 5- Documentación y guías, recomendaciones, ejemplos e información general de AngularJS. Disponibles en: <https://docs.angularjs.org/guide>.
- 6- Documentación, guías e información de TypeScript Disponibles en: <https://www.typescriptlang.org/docs/home.html>.
- 7- Descarga del framework Bootstrap. Disponible en: <https://getbootstrap.com/>.
- 8- www.youtube.com.
- 9- www.wikipedia.org.