
RED SOCIAL ZARPAS



Proyecto de FIn de Curso
2024-2025

Alumno implicado
Nixon Acosta Contreras

Tutor del proyecto
Carlos González

Desarrollo de aplicaciones Web
2º Curso

IES FRANCISCO DE QUEVEDO



Índice

1. Contenido del documento:	3
1.1. Introducción	3
1.2. Justificación del proyecto y objetivos	3
1.3. Planificación	4
1.4. Parte experimental	5
1.4.1. Análisis. ¿Qué hará la aplicación?	5
1.4.2. Diseño. ¿Cómo se hará la aplicación?	13
1.4.2.1 Diseño de la Base de Datos (Mysql)	13
1.4.2.2 Diseño del Backend (Sprint Boot)	14
1.4.2.3 Diseño del Frontend (Angular)	15
1.4.3. Implementación y pruebas	15
1.4.4. Implantación y documentación	15
1.4.5. Resultados y discusión	15
1.5. Conclusiones	15
1.6. Bibliografía y referencias	15
1.7. Anexos	15

1. Contenido del documento:

1.1. Introducción

Este documento trata sobre el desarrollo del proyecto de fin de ciclo ,que consiste en la creación de una red social web de subida de contenidos, similar o mejor dicho inspirada en la aplicación de Reddit. El proyecto tiene como objetivo implementar un conjunto de funcionalidades ,que permita a los usuarios subir publicaciones para compartir con los demás usuarios e interactuar entre ellos mediante comentarios de los mismos o por un chat de mensajería, usando como enfoque de desarrollo full stack con angular para el frontend, spring boot para la lógica del servidor y MYSQL para la conservación de los datos.

1.2. Justificación del proyecto y objetivos.

Justificación: He elegido este proyecto por el interés personal que tenía en el desarrollo de una aplicación o web similar a una red social debido a que en el pasado intente hacer una algo similar con tecnologías como java, firebase, etc... ,sobre recetas de cocinas, también me interesaba tocar Angular y profundizar en tecnologías que ya he usado como Sprint Boot y MYSQL para construir una aplicación funcional aunque no fuera desarrollado para ser un producto comercial, el proyecto me servirá para afianzar y aprender a controlar mejor las tecnologías usadas.

Objetivos: El objetivo principal del proyecto es desarrollar una red social web funcional con características similares a Reddit, usando las tecnologías de Angular, Sprint Boot y MYSQL. Ahora diré los objetivos específicos buscados en el proyecto:

- Implementación de un sistema de autenticación de usuarios(registro e inicio de sesión) usando Spring Boot.
- Permitir a los usuarios la creación de publicaciones de contenidos y chat de mensajería entre usuarios gestionada por MYSQL.
- Implementar un sistema de likes/dislikes para las publicaciones y comentarios.
- Desarrollar una interfaz de usuario con Angular que sea simple ,intuitiva y adaptable para los usuarios y sus dispositivos
- La creación de un API RESTFUL con spring boot para permitir la interacción entre el frontend, el backend y la base de datos.



1.3. Planificación

Cronograma general del proyecto(4 reuniones/fases):

- **1º fase(toma de contacto Angular):**
 - Análisis del proyecto .
 - Prototipado de las pantallas .
 - Prototipado de la base de datos.

- **2º fase:**
 - Desarrollo del backend: Creación del método de autenticación, gestión de usuarios.
 - Desarrollo del frontend: Creación de página principal, el registro/login.
 - Base de datos casi en su totalidad

- **3º fase:**
 - Desarrollo del backend: Creación de las publicaciones . comentarios , likes/dislikes y su gestión.
 - Desarrollo del frontend: Creación de la plantilla de publicaciones y del apartado de comentarios en las publicaciones
 - Base de datos terminada.

- **4º fase:**
 - Mejoras de backend y frontend necesarias relacionados con la anterior fase.
 - Pruebas y corrección de errores visuales y funcionales
 - Documentación Finalizada

1.4. Parte experimental

1.4.1. Análisis. ¿Qué hará la aplicación?

En la aplicación web los usuarios podrán subir sus publicaciones con imágenes y videos(intentara), además se podrá buscar contenido o usuarios por el nombre. Podrán comentar las publicaciones así como darles like/dislike a las mismas permitiendo la interacción entre usuarios creando una comunidad más dinámica y participativa. También podrá el usuario guardar publicaciones y verlas por separado así como abrir un chat de mensajería con distintos usuarios.

Requisitos funcionales:

- Gestion de Usuarios:
 - Sistema de registro: Sistema donde el usuario debe poder registrarse por medio de correos o cuentas de Google, etc... que permitirá la creación de los perfiles.
 - Sistema de inicio de sesión: Sistema que permite al usuario el inicio de sesión o el cierre de la misma.
 - Sistema de gestión de perfiles: Sistema que permitirá al usuario la gestión así como la eliminación de su perfil .
 - Sistema de mensajería: Sistema que permite el envío de mensajes entre usuarios a través de un chat individual.
- Gestión de Publicaciones:
 - Creación de publicación: Sistema que permitirá al usuario la creación de una publicación.
 - Eliminación de publicaciones: Sistema que permitirá al usuario a borrar sus propias publicaciones.
 - Búsqueda de publicación: Sistema que permitirá al usuario la búsqueda de publicaciones mediante un texto identificativo o temas relacionados a la publicación.
 - Guardado de publicaciones: Sistema que permitirá al usuario el guardado de publicaciones propias y no propias para tenerlas el.



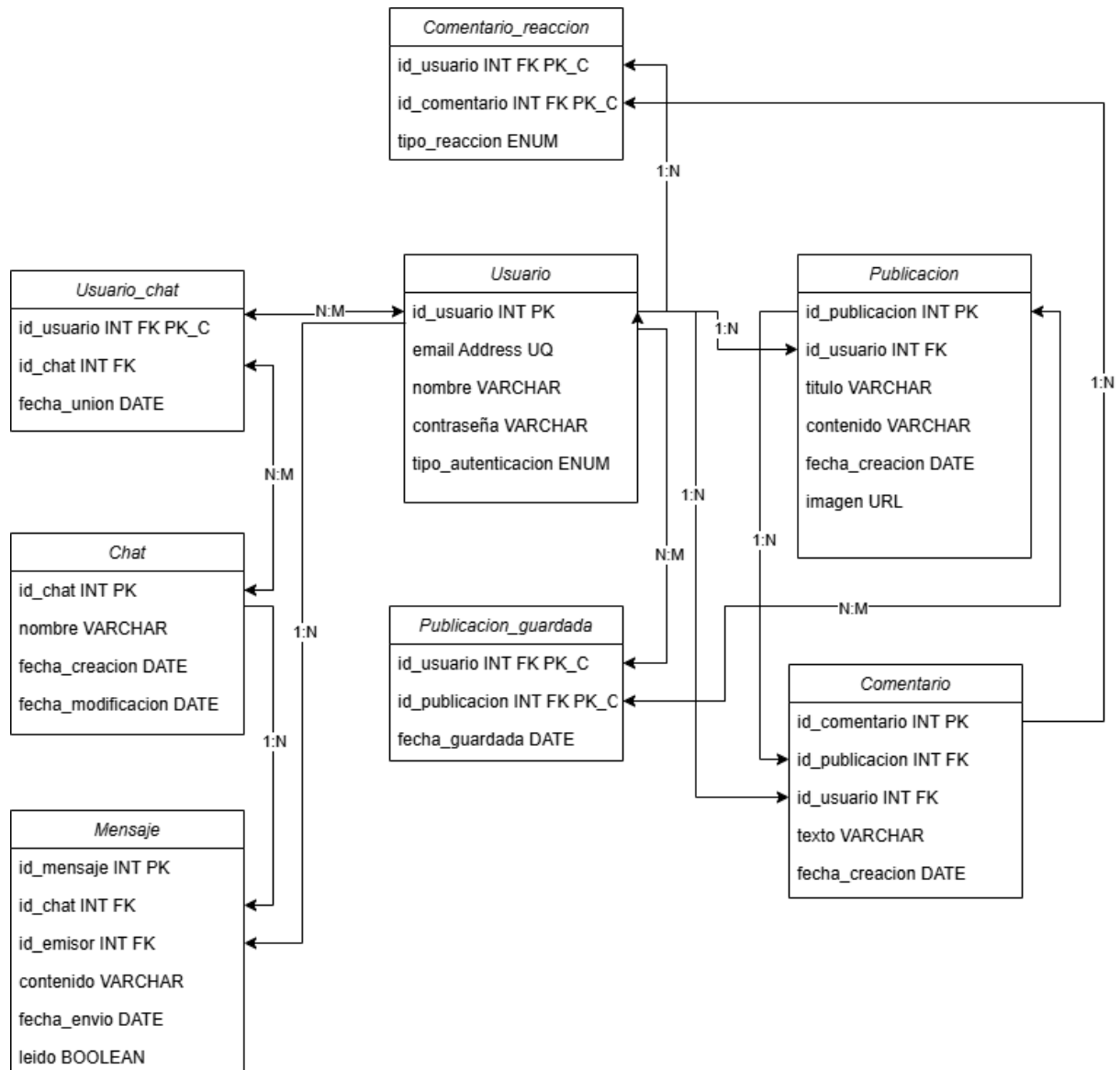
2º DAW 2024/2025

- Gestión de Comentarios:
 - Creación de comentarios: Sistema que permitirá al usuario crear un comentario sobre las publicaciones ya subidas
 - Eliminación de comentarios: Sistema que permitirá al usuario la eliminación de comentarios.

Requisito no funcionales:





- Rapidez de respuesta: Asegurar una rápida transición entre las vistas de la aplicación.
- Carga y tratamiento de imágenes: Las imágenes que se subirán a la aplicación pasarán por un tratamiento que reducirá su tamaño sin la reducción de la calidad de las imágenes, permitiendo el ahorro de espacio de la nube.
- Escalabilidad: Adaptación a distintas resoluciones de pantalla.
- Cerrar sesión: Funcionalidad que permite al usuario cerrar su sesión en cualquier momento.

Base de datos(Diagrama de modelo de datos):



Prototipado(Diseño de pantallas no final):

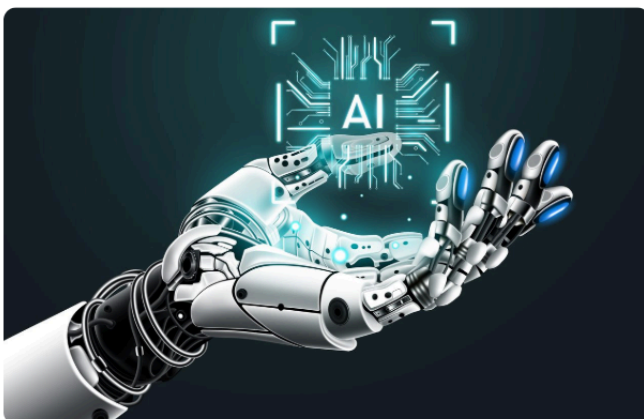
Zarpas**Temas Populares**

-  [r/Noticias](#)
-  [r/Tecnología](#)
-  [r/Memes](#)
-  [r/Videojuegos](#)

Título del post 1: El Futuro de la Inteligencia Artificial

En los últimos años, la inteligencia artificial ha experimentado un progreso impresionante. Desde la automatización de tareas simples hasta la creación de sistemas complejos capaces de aprender y mejorar por sí mismos. ¿Qué podemos esperar del futuro de la IA? Algunos predicen avances en áreas como la medicina personalizada, la automatización en el trabajo, y nuevas formas de entretenimiento interactivo.

#Tecnologia #Novedad

 100 20**COMENTARIOS**

Usuario1: ¡Interesante! ¿Alguien sabe más sobre esto?

Usuario2: Yo he leído algo, ¡deberíamos investigarlo más!



Página perfil (versión escritorio):

Zarpas





Juango123

¡Bienvenido a mi perfil! Aquí encontrarás mis publicaciones y más. Conectemos y compartamos ideas.

[Seguir](#)[Mensaje](#)[Editar](#)

150

Publicaciones

230

Seguidores

180

Siguiendo

Seguidores

Siguiendo

Guardados

Publicaciones

Post Reciente: La evolución de los videojuegos

Los videojuegos están cambiando constantemente, y este año se están destacando con tecnologías como la realidad virtual y aumentada. Los desarrolladores están creando experiencias inmersivas que realmente transportan a los jugadores a nuevos mundos.

 Imagen relacionada con videojuegos



2º DAW 2024/2025

Página perfil (versión móvil):

Zarpas



Juango123

¡Bienvenido a mi perfil! Aquí encontrarás mis publicaciones y más. Conectemos y compartamos ideas.

Seguir

Mensaje

Editar

150

Publicaciones

230

Seguidores

180


Siguiendo

Seguidores **Siguiendo** **Guardados**

Publicaciones

Post Reciente: La evolución de los videojuegos

Los videojuegos están cambiando constantemente, y este año se están destacando con tecnologías como la realidad virtual y aumentada. Los desarrolladores están creando experiencias inmersivas que realmente transportan a los jugadores a nuevos mundos.

 Imagen relacionada con videojuegos



50




10


Chat(Versión Escritorio):


Zarpas




Contactos

 Jose35

 PerigcoFuentes_

 La_Bestia13

 DonOmar

¡Hola! ¿Cómo estás?

15:30, 2 de abril

¡Hola! Estoy muy bien, ¿y tú?

15:31, 2 de abril

Todo bien, gracias. ¿Qué tal el día?

15:32, 2 de abril

Todo tranquilo, he estado trabajando en algunos proyectos. ¿Tú qué tal?

15:33, 2 de abril

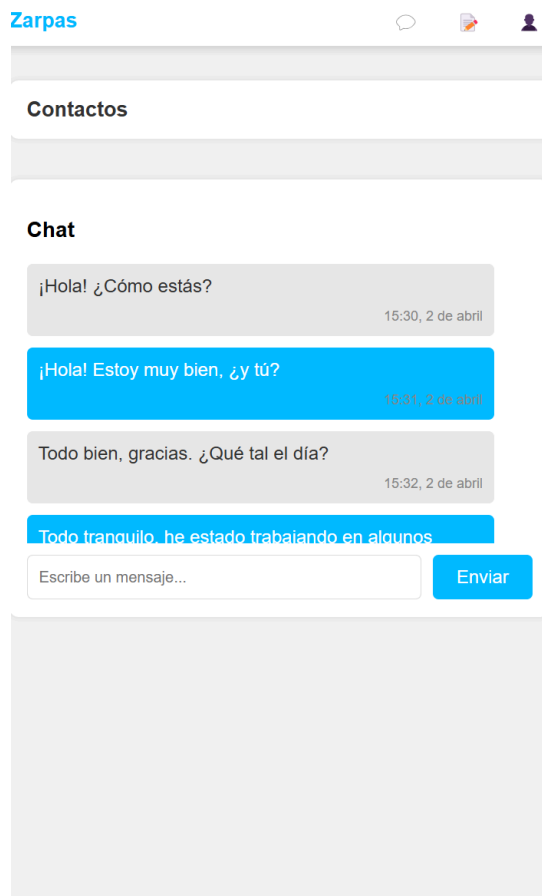
Escribe un mensaje...

Enviar



2º DAW 2024/2025

Chat(Versión Móvil):



1.4.2. Diseño. ¿Cómo se hará la aplicación?

Enlace del repositorio con lo que llevo de proyecto en GITHUB:

https://github.com/Nixo350/TFG_DAW2/tree/main

1.4.2.1 Diseño de la Base de Datos (Mysql)

Para la aplicación web (Zarpas) se utilizará un modelo Entidad-Relacion usando una base de datos de Mysql que estará desplegada a partir de un contenedor de Docker Compose usando la última imagen disponible de Mysql. La base de datos sigue sin estar todavia le faltan algunos retoques pero esta ya casi completada -----

Muestra de diagrama entidad-relación usando Mysql Workbench:



Hecho por Nixon Acosta



2º DAW 2024/2025

1.4.2.2 Diseño del Backend (Spring Boot)

Para el desarrollo del backend se está usando un proyecto de Spring Boot donde se utiliza una arquitectura basada en capas donde se separan todos los componentes (controladores, servicios, modelo).

Ejemplo de clase entidad finalizada `usuaria.java`:

```

11 @Entity
12 @Table(name = "usuario")
13 @Getter
14 @Setter
15 @Builder
16 @NoArgsConstructor
17 @AllArgsConstructor
18 public class Usuario {
19     @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     @Column(name = "id_usuario")
22     private Long idUsuario;
23
24     @Column(unique = true, nullable = false)
25     private String email;
26
27     private String nombre;
28
29     @Column(nullable = false)
30     private String contrasena;
31
32     @Column(name = "fecha_registro")
33     private Timestamp fechaRegistro;
34
35     @Column(name = "ultimo_login")
36     private Timestamp ultimoLogin;
37
38     @OneToMany(mappedBy = "usuario", cascade = CascadeType.ALL, orphanRemoval = true)
39     @JsonManagedReference
40     private List<Publicacion> publicaciones;
41
42     @OneToMany(mappedBy = "usuario", cascade = CascadeType.ALL, orphanRemoval = true)
43     @JsonManagedReference
44     private List<Comentario> comentarios;
45
46     @ManyToMany
47     @JoinTable(
48         name = "usuario_chat",
49         joinColumns = @JoinColumn(name = "id_usuario"),
50         inverseJoinColumns = @JoinColumn(name = "id_chat")
51     )
52     @JsonManagedReference
53     private List<Chat> chats;
54
55     @OneToMany(mappedBy = "emisor", cascade = CascadeType.ALL, orphanRemoval = true)
56     @JsonManagedReference
57     private List<Mensaje> mensajesEnviados;
58
59     @ManyToMany
60     @JoinTable(
61         name = "publicacion_guardada",
62         joinColumns = @JoinColumn(name = "id_usuario"),
63         inverseJoinColumns = @JoinColumn(name = "id_publicacion")
64     )
65     @JsonManagedReference
66     private List<Publicacion> publicacionesGuardadas;
67
68     @ManyToMany
69     @JoinTable(
70         name = "comentario_reaccion",
71         joinColumns = @JoinColumn(name = "id_usuario"),
72         inverseJoinColumns = @JoinColumn(name = "id_comentario")
73     )
74     @JsonManagedReference
75     private List<Comentario> comentariosReaccionados;
76
77     @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER, mappedBy = "usuario")
78     @JsonBackReference
79     private Set<UsuarioRol> usuarioRoles = new HashSet<>();
80 }

```


1.4.2.3 Diseño del Frontend (Angular)

Para la aplicación web (Zarpas) en el desarrollo del Frontend se va a desarrollar con Angular en su versión más actual, la 19. Se usará una estructura basada en componentes y será igual a la base de datos en el aspecto de que se usará un servidor en un contenedor local usando Docker Compose para interactuar con todos los componentes. (Todavía no está empezado este apartado a Seguir)

Ejemplo de entidad usuario.ts:

```

config.json  TS TipoReaccion.ts  TS Usuario.ts X  TS Rol.ts  TS UsuarioRo

Zarpas_Frontend > gestion-zarpas-frontend > src > app > modelos > TS Usuario.ts > •-O Usuar

1  import { Chat } from "../Chat";
2  import { Comentario } from "../Comentario";
3  import { ComentarioReaccion } from "../ComentarioReaccion";
4  import { Mensaje } from "../Mensaje";
5  import { Publicacion } from "../Publicacion";
6  import { PublicacionGuardada } from "../PublicacionGuardada";
7  import { UsuarioRol } from "../UsuarioRol";
8
9  export interface Usuario {
10     idUsuario: number;
11     email: string;
12     nombre: string;
13     contrasena: string;
14     fechaRegistro: Date;
15     ultimoLogin: Date;
16     publicaciones: Publicacion[];
17
18     mensajesEnviados: Mensaje[];
19     publicacionesGuardadas: PublicacionGuardada[];
20     comentariosReaccionados: ComentarioReaccion[];
21     usuarioRoles: UsuarioRol[];
22 }
23

```

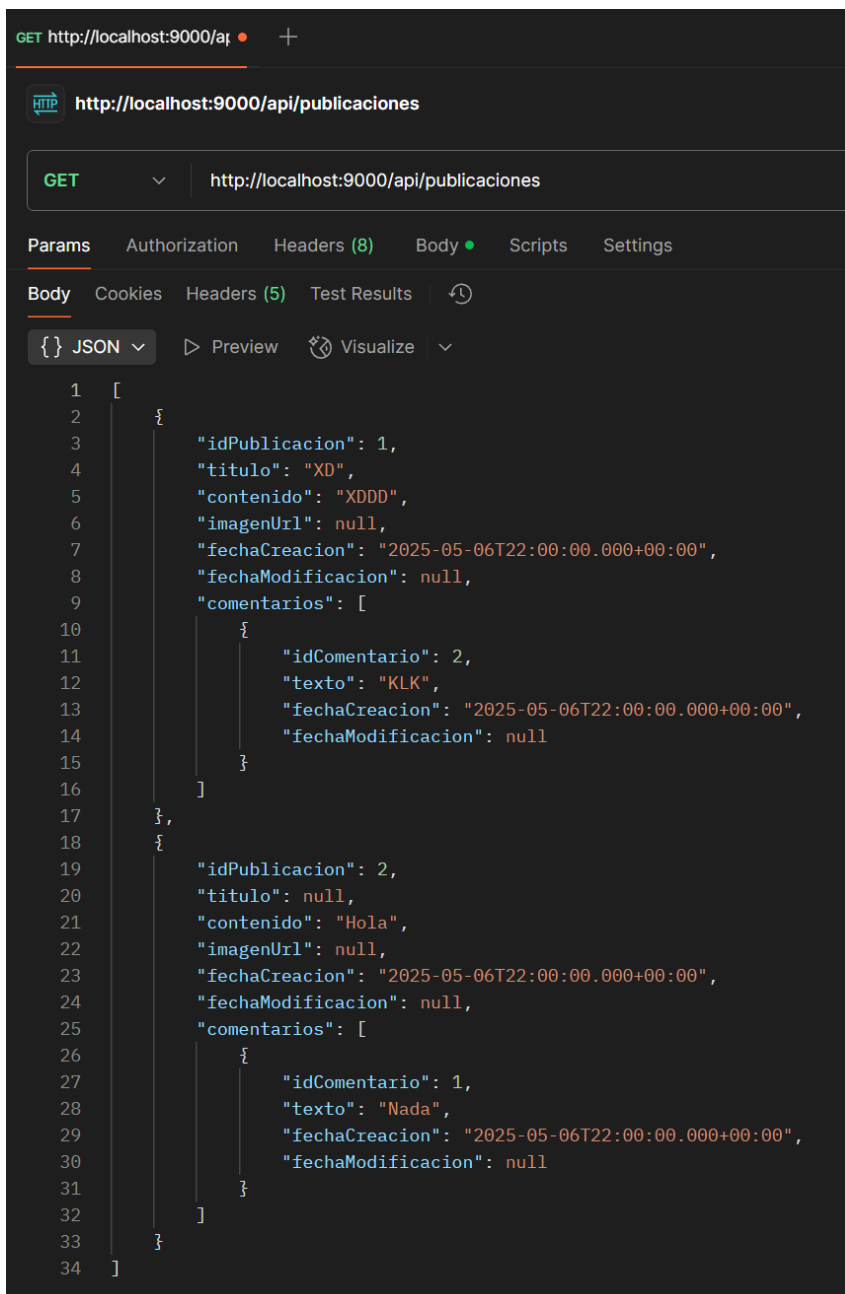
El diseño del icono de la aplicación web Zarpas es el siguiente;



1.4.3. Implementación y pruebas

Para la implementación de la aplicación web Zarpas aun no esta disponible ya que como tal la implementación de los contenedor para cada uno de los servicios no está desarrollado todavía porque estoy haciendo tanto backend como frontend están desplegadas en local.

Las pruebas las puedo hacer mediante postman para el muestreo de datos mediante la API creada como voy a mostrar en la siguiente diapositiva(solo mostraré los datos de las publicaciones mediante postman. Hay que recalcar que los datos mostrados en las pruebas son totalmente ficticios):



```
GET http://localhost:9000/api/publicaciones

GET http://localhost:9000/api/publicaciones

Params  Authorization  Headers (8)  Body  Scripts  Settings
Body  Cookies  Headers (5)  Test Results  ↻
{} JSON  Preview  Visualize  v

1  [
2    {
3      "idPublicacion": 1,
4      "titulo": "XD",
5      "contenido": "XDDD",
6      "imagenUrl": null,
7      "fechaCreacion": "2025-05-06T22:00:00.000+00:00",
8      "fechaModificacion": null,
9      "comentarios": [
10       {
11         "idComentario": 2,
12         "texto": "KLK",
13         "fechaCreacion": "2025-05-06T22:00:00.000+00:00",
14         "fechaModificacion": null
15       }
16     ]
17   },
18   {
19     "idPublicacion": 2,
20     "titulo": null,
21     "contenido": "Hola",
22     "imagenUrl": null,
23     "fechaCreacion": "2025-05-06T22:00:00.000+00:00",
24     "fechaModificacion": null,
25     "comentarios": [
26       {
27         "idComentario": 1,
28         "texto": "Nada",
29         "fechaCreacion": "2025-05-06T22:00:00.000+00:00",
30         "fechaModificacion": null
31       }
32     ]
33   }
34 ]
```



1.4.4. Implantación y documentación

Todavía me falta la implementación ya que el proyecto aun lo tengo en local 100% y lo unico que esta con un servicio en un contenedor es la Base de datos, pero en principio todo se implementara o mejor dicho configurara con un docker-compose.yml que definirá los servicios y variables de entorno a utilizar en tres contenedores uno ya mencionado y uno para el frontend(Angular) y el backen(Spring Boot).

Por el apartado de la documentación estará subido en el documento README.MD en el repositorio que compartiré y que está el enlace en este documento para su visibilidad.

1.4.5. Resultados y discusión

1.5. Conclusiones

1.6. Bibliografía y referencias

1.7. Anexos