

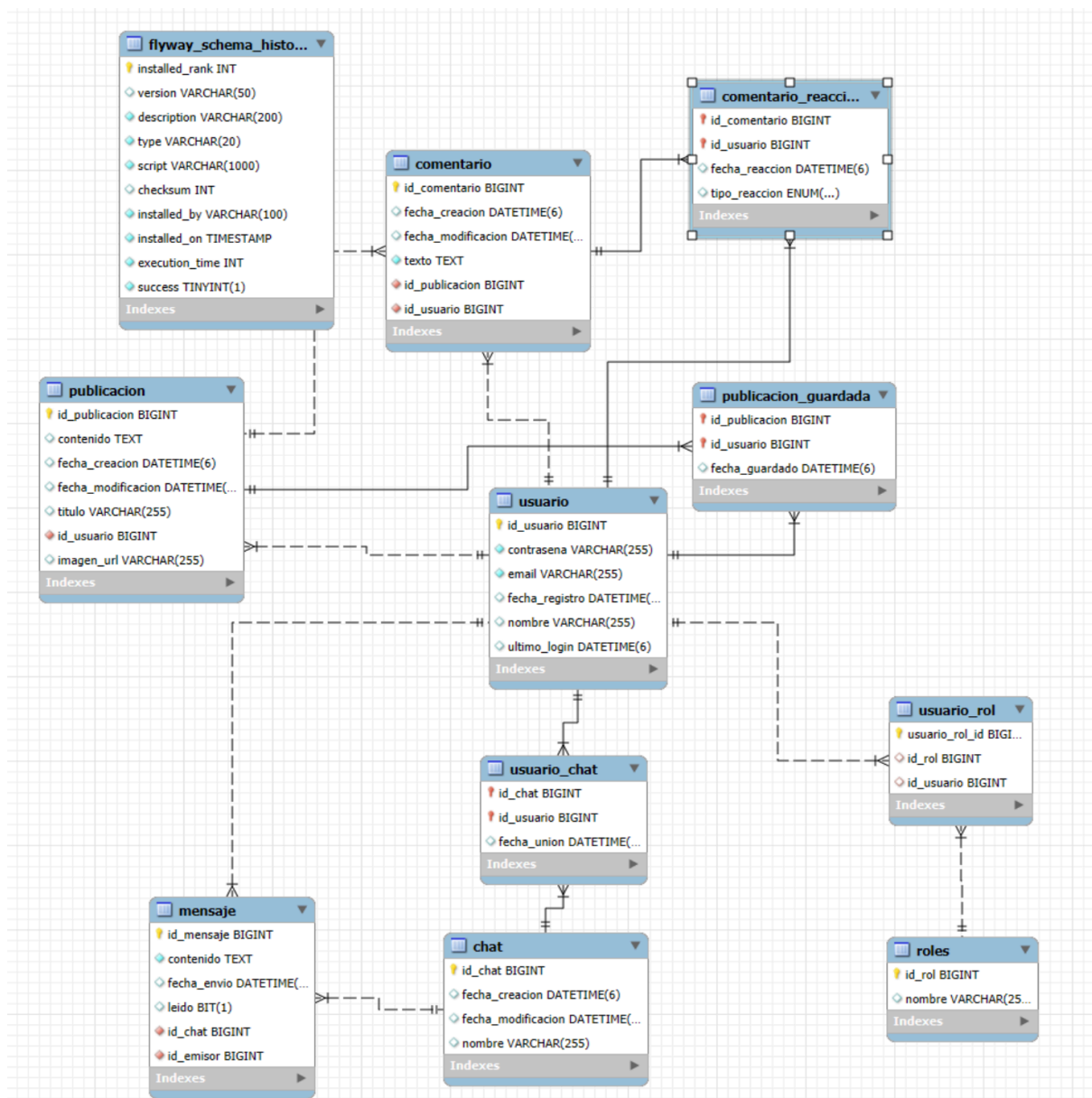
Planificación (8 de mayo)

Según la planificación gran parte del Backend ya está terminada en principio pero le faltan algunos retoques debido a lo siguiente

Está en la documentación al principio del punto 1.4.2 el enlace al repositorio de [GITHUB](#).

Y esta modificada la documentación en el 1.4.2 y hecha parte en la 1.4.3

La base de datos se ha modificado ya que he metido implementación de roles debido a que me he dado cuenta que para eliminar cuentas o publicaciones solo lo podrá hacer el mismo usuario creador o administradores que filtren contenidos maliciosos por así decirlo, debido a este pensamiento es que he modificado la base de datos . Además me he dado cuenta tarde pero tengo que añadir imágenes en perfiles de usuario y chats.



El desarrollo del Backend con Spring Boot ya se conecta casi en su totalidad con la base de datos y me falta el desarrollo de los comentarios y del chat.

A la clase application.properties he añadido varias cosas.

```

1  spring.application.name=Zarpas_Backend
2  server.port=9000
3
4  #Conexion BBDD
5  spring.datasource.url=jdbc:mysql://localhost/zarpas
6  spring.datasource.username=root
7  spring.datasource.password=admin
8
9  #Hibernate
10 spring.jpa.hibernate.ddl-auto=update
11 #spring.jpa.show-sql=true
12 #spring.jpa.properties.hibernate.format_sql=true
13
14 #Url Image
15 spring.flyway.baseline-on-migrate=true

```

El desarrollo del frontend esta diría en su punto medio ,me falta el desarrollo visual casi en su mayoría. Aquí se puede ver el desarrollo de las entidades en el proyecto de angular.

```

config.json  TS TipoReaccion.ts  TS Usuario.ts  TS Rol.ts  TS UsuarioRol.ts  TS
Zarpas_Frontend > gestion-zarpas-frontend > src > app > modelos > TS Usuario.ts > Usuario > usuario
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 }

```

También he podido realizar pruebas con postman para mostrar datos de la API creada como se puede ver en la documentación en el apartado 1.4.3. Hay que recalcar que los datos mostrados en las pruebas son totalmente ficticios

