

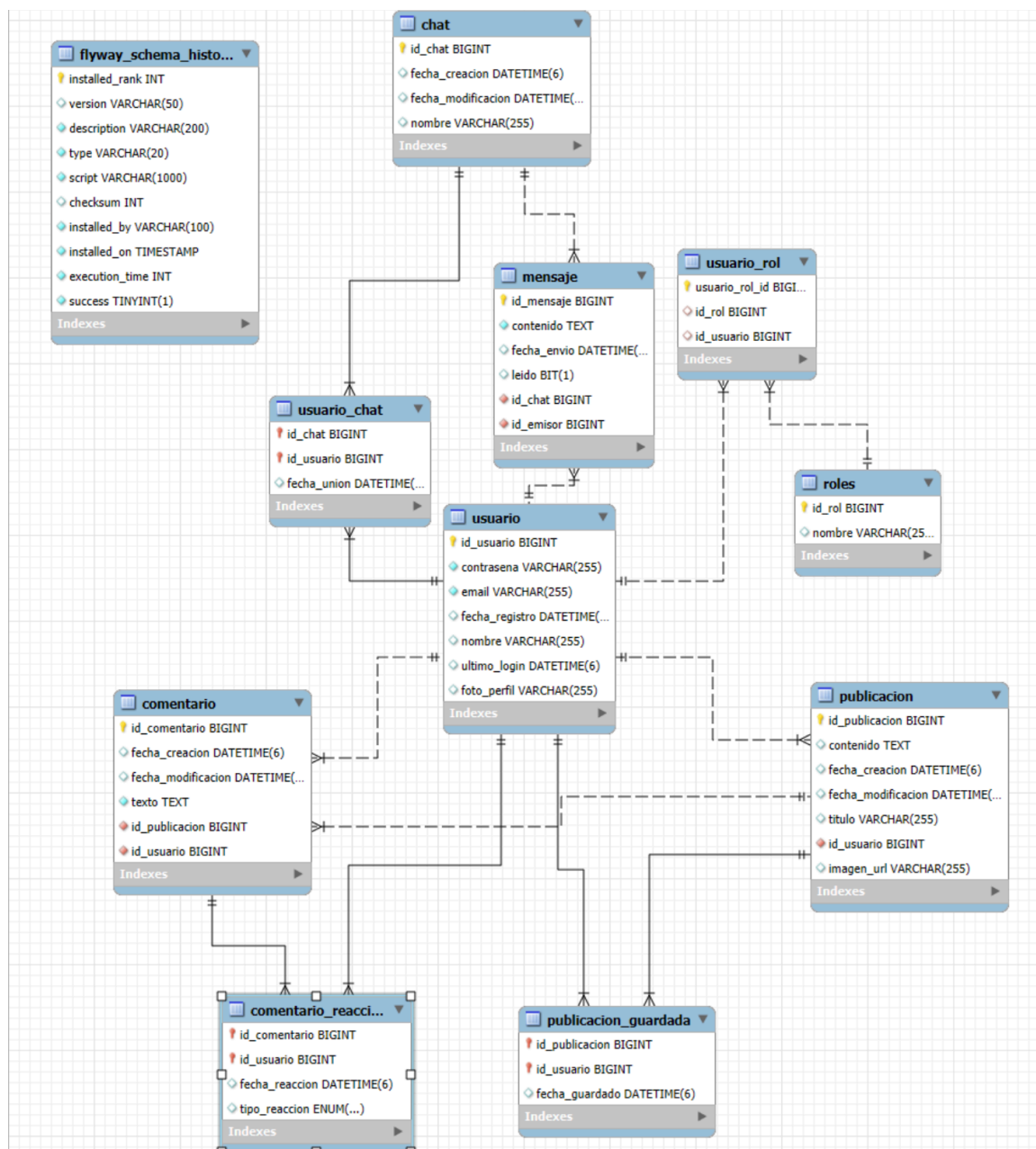
Planificación (22 de mayo)

Según la planificación la documentación debería estar terminada pero debido a que para finalizarla necesito terminar apartados que ahora diré. Otra cosa que como fecha límite para todo me he puesto el 10 de Junio.

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

Desarrollo BBDD

La última actualización de la base de datos es esta (se añadió un apartado para la foto de perfil de los usuarios);



Desarrollo del backend

En principio el backend ya está desarrollado casi en su totalidad. Tiene casi todos los requisitos del API REST y solo faltaría terminar la implementación del sistema de autenticación que la llegue a inhabilitar por motivos de modificación del proyecto y el añadido de roles a la base de datos, tengo que volver a implementarlo.

Ejemplo UsuarioController.java

```
ZarpasBackendApplication.java  MensajeController.java  ControladorGeneral.java  ComentarioController.java  ChatController.java  Rol.java  UsuarioController.java x
10
11 import java.util.List;
12 import java.util.Set;
13
14 @RestController  Nix0350
15 @RequestMapping("/api/usuarios")
16 public class UsuarioController {
17
18     private final UsuarioService usuarioService;
19
20     @Autowired  Nix0350
21     public UsuarioController(UsuarioService usuarioService) {
22         this.usuarioService = usuarioService;
23     }
24
25     @PostMapping  Nix0350
26     public ResponseEntity<Usuario> crearUsuario(@RequestBody Usuario usuario) {
27         Usuario nuevoUsuario = usuarioService.guardarUsuario(usuario);
28         return new ResponseEntity<>(nuevoUsuario, HttpStatus.CREATED);
29     }
30
31     @GetMapping  Nix0350
32     public ResponseEntity<List<Usuario>> obtenerTodosLosUsuarios() {
33         List<Usuario> usuarios = usuarioService.obtenerTodosLosUsuarios();
34         return new ResponseEntity<>(usuarios, HttpStatus.OK);
35     }
36
37     @GetMapping("/{id}")  Nix0350
38     public ResponseEntity<Usuario> obtenerUsuarioPorId(@PathVariable("id") Long id) {
39         return usuarioService.obtenerUsuarioPorId(id).map(usuario -> new ResponseEntity<>(usuario, HttpStatus.OK)).orElse(new ResponseEntity<>(HttpStatus.NOT_FOUND));
40     }
41
42     @GetMapping("/{email}")  Nix0350
43     public ResponseEntity<Usuario> obtenerUsuarioPorEmail(@PathVariable("email") String email) {
44         return usuarioService.obtenerUsuarioPorEmail(email).map(usuario -> new ResponseEntity<>(usuario, HttpStatus.OK)).orElse(new ResponseEntity<>(HttpStatus.NOT_FOUND));
45     }
46
47     @PutMapping("/{id}")  Nix0350
```

Ejemplo de UsuarioServiceImpl.java

```
16 import java.util.List;
17 import java.util.Optional;
18 import java.util.Set;
19 import java.util.stream.Collectors;
20
21 @Service
22 public class UsuarioServiceImpl implements UsuarioService {
23
24     private final UsuarioRepository usuarioRepository;
25     private final RolRepository rolRepository;
26     private final UsuarioRolRepository usuarioRolRepository;
27
28     @Autowired
29     public UsuarioServiceImpl(UsuarioRepository usuarioRepository, RolRepository rolRepository, UsuarioRolRepository usuarioRolRepository) {
30         this.usuarioRepository = usuarioRepository;
31         this.rolRepository = rolRepository;
32         this.usuarioRolRepository = usuarioRolRepository;
33     }
34
35     @Override
36     @Transactional
37     public Usuario guardarUsuario(Usuario usuario) {
38         if (usuario.getFechaRegistro() == null) {
39             usuario.setFechaRegistro(Instant.now());
40         }
41         if (usuario.getUltimoLogin() == null) {
42             usuario.setUltimoLogin(Instant.now());
43         }
44         return usuarioRepository.save(usuario);
45     }
46
47     @Override
48     @Transactional(readOnly = true)
49     public Optional<Usuario> obtenerUsuarioPorId(Long id) {
50         return usuarioRepository.findById(id);
51     }
52
53     @Override
54     @Transactional(readOnly = true)
55     public Optional<Usuario> obtenerUsuarioPorEmail(String email) {
56         return usuarioRepository.findByEmail(email);
57     }
58 }
```

Desarrollo del Frontend

En principio el frontend esta algo desarrollado solo faltaria intentar implementar el chat,que me esta dando problemas los mensajes.Y le he añadido componentes por BootStrap y Angular material, ahora estoy viendo si cambio algunos componentes por otros para que sea más fácil o mejor visible tanto el código como la usabilidad de la web.

Implementación

Como comento en la documentación tengo pensada ponerlo todo en un README.MD y como todavía no he terminado la aplicación tampòco me he puesto con ello pero el despliegue sería por un docker-compose.yml con tres servicios la base de datos , el frontend(Angular) y el backend(spring boot)

Se añade icono de la aplicación Zarpas;

