**Como eu fiz: geração de tokens**

Primeiramente, você precisará adicionar a biblioteca ***Auth0 java-jwt*** no projeto, incluindo essa dependência no pom.xml:

<dependency>

<groupId>com.auth0</groupId>

<artifactId>java-jwt</artifactId>

<version>4.2.1</version>

</dependency>

Na sequência, será necessário criar a classe responsável pela geração dos *tokens*:

@Service

public class TokenService {

@Value("${api.security.token.secret}")

private String secret;

public String gerarToken(Usuario usuario) {

try {

var algoritmo = Algorithm.HMAC256(secret);

return JWT.create()

.withIssuer("API Voll.med")

.withSubject(usuario.getLogin())

.withExpiresAt(dataExpiracao())

.sign(algoritmo);

} catch (JWTCreationException exception){

throw new RuntimeException("erro ao gerar token jwt", exception);

}

}

private Instant dataExpiracao() {

return LocalDateTime.now().plusHours(2).toInstant(ZoneOffset.of("-03:00"));

}

}

Você também vai precisar adicionar a seguinte propriedade no arquivo application.properties:

api.security.token.secret=${JWT\_SECRET:12345678}

Por fim, será necessário criar o DTO DadosTokenJWT e alterar a classe AutenticacaoController:

public record DadosTokenJWT(String token) {}

@RestController

@RequestMapping("/login")

public class AutenticacaoController {

@Autowired

private AuthenticationManager manager;

@Autowired

private TokenService tokenService;

@PostMapping

public ResponseEntity efetuarLogin(@RequestBody @Valid DadosAutenticacao dados) {

var authenticationToken = new UsernamePasswordAuthenticationToken(dados.login(), dados.senha());

var authentication = manager.authenticate(authenticationToken);

var tokenJWT = tokenService.gerarToken((Usuario) authentication.getPrincipal());

return ResponseEntity.ok(new DadosTokenJWT(tokenJWT));

}

}