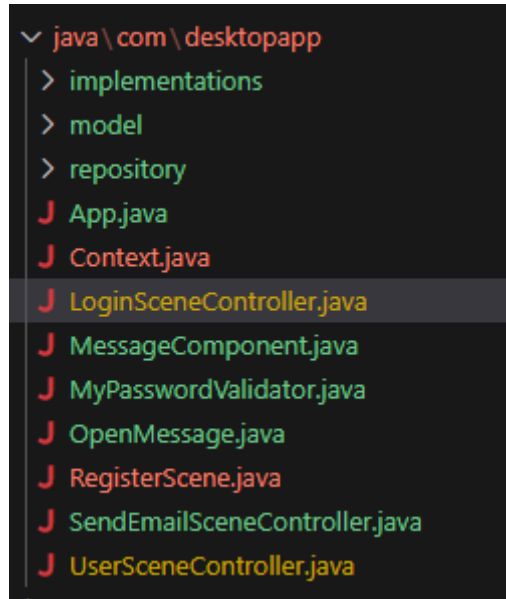


Trabalho muito bem-organizado. Usou conceitos como repositório e implementações nas primeiras semanas onde estávamos apenas aprendendo eles. Muito a elogiar aqui.



Achei seu código bem clean. Você fez bastante *Early Return* e usou bastante streams deixando bem claro.



Os testes ficaram bons e você compreendeu a ideia. Fez uma ótima diversidade de aspectos dos testes e, em geral, o trabalho ficou muito bom.



```
@Test
void validateSmallPass() {
    assertEquals(MyPasswordValidator.Validate("oi12321"), false);
    assertEquals(MyPasswordValidator.Validate("1234!aaaaaa"), true);
    assertEquals(MyPasswordValidator.Validate("oio@q1iaaa"), true);
    assertEquals(MyPasswordValidator.Validate("minh"), false);
}
```

Principal feature a prova bem implementada!



```
Message new_message = new Message();
new_message.setAbout(inputAbout.getText());
new_message.setMessage(inputMessage.getText());
new_message.setId_by(user.getId());
new_message.setId_from(users.get(0).getId());

messageRepository messageRepo = new messageRepository();
messageRepo.create(new_message);
```

Transferência de dados entre as telas bem-feito. Elogio adicional as diversas formas que você trabalhou com as telas. Buscando comportamentos mais específicos.



```
this.user = user;  
  
VBox scene = UserSceneController.CreateScene(user);  
  
box.getChildren().setAll(scene);
```

Boa organização e bom uso de Status code. Bom projeto no geral. A organização interna dos arquivos, espaçamento, comentário, entre outros, poderia estar um pouco melhor.



Jwt bem usado. Spring Security bem configurado.

```
public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
    return http  
        .csrf(csrf -> csrf.disable())  
        .authorizeHttpRequests(authorize -> authorize  
            .requestMatchers("/user", "/auth").permitAll()  
            .anyRequest().authenticated()  
        )  
        .addFilterBefore(new JWTAuthenticationFilter(jwtService), UsernamePasswordAuthenticationFilter.class)  
        .build();  
}
```

Competências	Cristian
Criar Controladores/Endpoints	Apto
Configurar uma rota personalizada	Apto
Configurar um método HTTP personalizado	Apto
Receber dados pela URL	Apto
Receber dados pelo Corpo	Apto
Receber os dados por Query	Apto
Receber os dados pela Header	Apto
Retornar o Status Code	Apto
Reconhecer qual o melhor Status Code a usar	Apto
Retornar um JSON	Apto
Criar um serviço e configurá-lo	Apto
Utilizar um serviço definido	Apto
Configurar um serviço com tempo de vida	Apto
Reconhecer o melhor tempo de vida do serviço	Apto
Criar um modelo JPA	Apto
Criar um repositório JPA	Apto
Definir uma query personalizada em um repositório JPA	Apto
Modificar e salvar dados dos modelos JPA	Apto
Configurar o Cors	Apto
Aplicar o BCrypt no armazenamento de senhas	Apto
Configurar o JWT para autenticação	Apto
Configurar o Spring Security	Apto
Criar um middleware	Apto
Saber estruturar um projeto organizado e com sentido	Apto
Saber resolver problemas usando desenvolvimento Backend	Apto
Aplicar o princípio da Responsabilidade Única (S)	Apto
Aplicar o princípio do Aberto-Fechado (O)	Apto
Aplicar o princípio da Segregação de Interface (I)	Apto
Aplicar o princípio da Inversão de Dependência (D)	Apto
Criar testes com Junit [RETROATIVO]	Apto
Criar testes que façam sentido e sejam úteis [RETROATIVO]	Apto
Representar relações nos modelos JPA [EXTRA]	Em Desenvolvimento
Acessar dados usando relações dos modelos JPA [EXTRA]	Inapto
Média	100%
Média Ponderada	97%

Competências	Cristian
Construir telas usando Scene Builder	Apto
Construir telas com qualidade, organização e sentido.	Apto
Saber vincular um controlador JavaFX a tela construída	Apto
Saber vincular componentes e usar seus dados	Apto
Saber vincular métodos JavaFX a eventos	Apto
Ser capaz de manipular Stages e Scenes	Apto
Ser capaz de passar dados entre diferentes controladores.	Apto
Ser capaz de processar dados simples usando Streams	Apto
Configurar e usar o Maven	Apto
Instalar pacotes usando o Maven	Apto
Escrever testes unitários com Junit	Apto
Criar testes consistentes e variados	Apto
Configurar o persistence.xml	Apto
Criar tabelas usando modelos no Hibernate	Apto
Usar o Hibernate para salvar dados	Apto
Buscar dados e fazer queries com Hibernate	Apto
Ser capaz de modificar dados no banco usando Hibernate	Apto
Ser capaz de construir CRUDs funcionais	Apto
Média	100%

Comentários gerais: Um excelente trabalho. Quase irretocável.

Média geral: 99%

