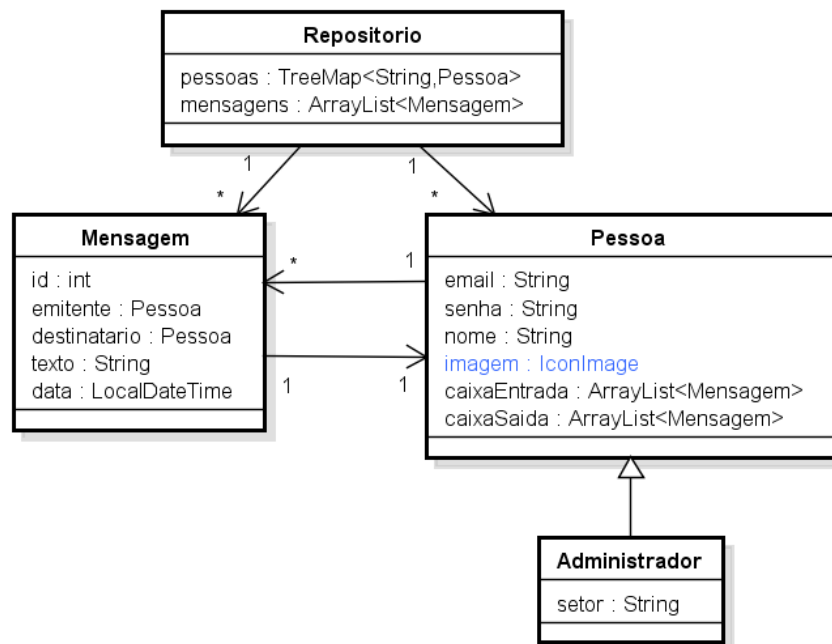


Projeto 2

Objetivo: Criar um Sistema de Mensagens Rápidas (SMR).

Cenário Real: Pessoas cadastradas e identificadas pelo email podem enviar mensagens entre si, desde que estejam logadas no sistema. Após um *login* com a senha, uma pessoa permanece logada no sistema até o seu *logoff* e durante este tempo ela poderá enviar/receber/listar mensagens. Só existirá uma pessoa logada no sistema de cada vez. Quando uma pessoa A envia uma mensagem para outra pessoa B, a mensagem é adicionada na caixa de entrada de B e na de saída de A. Uma pessoa pode enviar mensagem para ela mesma e, neste caso, a mensagem constará em ambas as caixas de entrada e de saída.

Diagrama UML: Repositório e Modelo de Negócio



Regras de negócio:

1. Uma pessoa é identificada pelo e-mail e senha.
2. Uma mensagem é identificada por um número sequencial auto incrementado (1, 2, 3, ...) gerado internamente pelo sistema e que é único em todo o sistema.
3. Uma mensagem é datada pelo horário do computador no momento de seu envio.
4. Uma pessoa só poderá ter acesso às mensagens que ela enviou ou recebeu.
5. O administrador pode espionar mensagens do sistema e ver as estatísticas.
6. Uma mensagem nunca poderá ser removida do repositório.
7. O texto da mensagem não pode ser vazio e deve ter no máximo 200 caracteres.
8. Só poderá existir uma pessoa logada de cada vez.

Classe Fachada (métodos estáticos)

public static Pessoa	login(email,senha) – localiza a pessoa no repositório, a torna pessoa logada e retorna esta pessoa
public static void	logoff() – descarta a pessoa logada
public static Pessoa	obterLogada() – retorna a pessoa logada
public static Pessoa	cadastrarPessoa(email, senha, nome, iconimage) – cadastra uma pessoa no sistema
public static Pessoa	cadastrarAdministrador(email, senha, nome, iconimage, setor) – cadastra um administrador no sistema
public static ArrayList<Pessoa>	listarPessoas(termo) - retorna as pessoas cujo nome contém o termo fornecido (termo vazio retorna todas)
public static Mensagem	enviarMensagem(emaildestinatário, texto) – cria uma nova mensagem,

	considerando que o emissor é a pessoa logada.
public static ArrayList<Mensagem>	listarCaixaEntrada() – retorna as mensagens recebidas pela pessoa logada
public static ArrayList<Mensagem>	listarCaixaSaida() – retorna as mensagens enviadas pela pessoa logada
public static Mensagem	apagarMensagem(id) – exclui a mensagem da caixa de entrada e/ou de saída da pessoa logada e retorna a mensagem excluída
public static ArrayList<Mensagem>	espionarMensagens(termino) – retorna as mensagens cujo texto contém o termino fornecido (termino vazio retorna todas)
public static ArrayList<Pessoa>	relatorio1() – retorna as pessoas que não enviaram mensagens
public static ArrayList<Mensagem>	relatorio2() – retorna as mensagens com emissor igual ao destinatário

Estes métodos devem lançar exceção quando necessário.

A Fachada deverá declarar o atributo *private static Pessoa logado*.

Classe de Aplicação Swing com as funcionalidades (Menu):

1. Login	Dados: email e senha Resultados: confirmação
2. Logoff	Resultados: confirmação
3. Cadastrar Pessoa	Dados: email, senha, nome, bufferedimage Resultados: confirmação de cadastro
4. Cadastrar Administrador	Dados: email, senha, nome, setor, nome do arquivo de imagem Resultados: confirmação de cadastro
5. Listar Pessoas	Dados: termino Resultados: as pessoas que contenham o termino no nome (se termino vazio retorna todas)
6. Enviar Mensagem	Dados: email do destinatário e texto da mensagem Resultados: ID da mensagem enviada
7. Listar caixa de entrada	Resultados: as mensagens da caixa de entrada da pessoa logada
8. Listar caixa de saída	Resultados: as mensagens da caixa de saída da pessoa logada
9. Apagar Mensagem	Dados: id da mensagem (que consta na cx de entrada/saída da pessoa logada) Resultados: confirmação
10. Espionar Mensagens (*)	Dados: termino Resultados: id, email do emissor, email do destinatário, data e texto de todas as mensagens do sistema que contenham o termino (se termino vazio retorna todas)
11. Relatório 1 (*)	Resultados: as pessoas que não enviaram mensagens
12. Relatório 2 (*)	Resultados: as mensagens com emissor igual ao destinatário

As opções 6 a 9 só poderão ser executadas se a pessoa estiver logada.

(*) As opções 4 e 10 a 12 só poderão ser exibidas e executadas quando a pessoa logada for um administrador.

Considerações finais:

- Desenvolver o sistema usando as camadas (pacotes): aplicacao, fachada, repositório e modelo
- A Fachada será testada por uma aplicação console fornecida pelo professor
- As datas devem ser visualizadas formatadas “dd/MM/yyyy hh:mm:ss”
- Utilize somente coleções indicadas no diagrama
- Utilizar expressões de regex ou campos com máscaras, para validação de email
- Mensagens de exceções devem ser exibidas num Label específico
- Várias funcionalidades podem ser implementadas numa única tela.
- A aplicação swing deverá pré-cadastrar 3 pessoas, 6 mensagens e 1 administrador com os dados: email=admin@ifpb, senha=123, nome=admin1, setor=DTI, sem imagem.

Trabalho em dupla – avaliação individual no laboratório na data 12/12/2019

Enviar para fausto.ifpb.poo@gmail (no dia da avaliação) o projeto eclipse zipado, contendo seu nome dentro dos arquivos fontes. Projetos contendo conteúdos idênticos serão zerados!