Sistemas Distribuídos

Relatório da Trabalho Prático 2

2022/2023

Trabalho realizado por:

- Rodrigo Alves, nº48681
- Diogo Carreiro, nº48729

Introdução

O objetivo do trabalho é implementar um sistema com arquitetura distribuída, neste caso o REST, que inclua um cliente geral, um cliente de administração e um servidor.
O **cliente geral** tem a possibilidade de:

- Registar novos anúncios (oferta ou procura), sendo que o anúncio fica em estado inativo até ser aprovado pelo gestor.
- Listar anúncios (com estado ativo) de ambos os tipos.
- Procurar anúncios, enviando texto a pesquisar na descrição, e opcionalmente uma localização
- Obter todos os detalhes de um anúncio, dado o seu identificador (aid).
- Enviar nova mensagem ao anunciante, pelo identificador do anúncio (aid). E consultar as mensagens inseridas para um determinado anúncio (aid).

O **cliente de gestão** pode:

- Listar anúncios por estado.
- Aprovar um anúncio, ou alterar o estado de um anúncio (ativo/inativo).

Por outro lado o **servidor** é responsável por:

- Serviço para as operações a disponibilizar a cada cliente, sendo que nas listagens do cliente geral mostra apenas anúncios em estado ativo.
- Armazenamento persistente de dados.

Para isso utilizámos o spring para o desenvolvimento do projeto, uma vez que achámos que estamos mais familiarizados com esta ferramenta.

Estrutura

Neste trabalho, foram implementados 3 módulos, um representa o cliente geral, outro o de administração e o servidor. O módulo servidor contém todas as implementações que são necessárias para o bom funcionamento do sistema.

- Módulo Cliente Geral:
    - **public void add_Anuncio(String localizacao, Integer preco, String data, String genero, anunciante, String tipo_alojamento, String tipo_anuncio):**
        - Este método irá adicionar um anúncio à base de dados.
    - **public Optional<Anuncio_Entity> get_Anuncio(Integer aid):**
        - Este método retorna o anúncio que tem o id igual ao do parâmetro aid (se existir).
    - **public List<Anuncio_Entity> getAllAnuncios():**
        - Este método retorna todos os anúncios.
    - **public List<Anuncio Entity> getAllAnunciosByTipoanuncio (String tipo_anuncio):**
        - Este método retorna todos os anúncios de um certo tipo(oferta ou de procura).
    - **public List<Anuncio_Entity> getAllAnunciosByAnunciante (String anunciante):**
        - Este método retorna todos os anúncios de um anunciante.
    - **public void addMessage(String message, String emissor, Integer aid):**
        - Este método adiciona uma mensagem ao um anúncio com id = aid.

    - **public Optional<List<Message_Entity>> getAllMessagesByAid(Integer aid):**
        - Este método retorna todas as mensagens de um dado anúncio.

- Módulo Cliente de Administração;
    - **public Optional<Anuncio_Entity> get_Anuncio(Integer aid):**
        - Este método retorna o anúncio que tem o id igual ao do parâmetro aid (se existir).
    - **public List<Anuncio_Entity> getAllAnunciosByEstado(String estado):**
        - Este método retorna todos os anúncios de um certo estado (ativo ou inativo).
    - **public void change_state(Integer aid):**
        - Este método altera o estado de um anuncio na base de dados
    - **public void aprove_state(Integer aid):**
        - Este método serve para aprovar um anúncio. Basicamente se o anúncio em causa não tiver o seu estado definido como “ativo” este método vai alterá-lo. Caso contrário o método não altera o seu estado.

Problemas encontrados e observações

Durante a realização deste trabalho, tivemos dificuldades na utilização do JSON, na ligação das tabelas com o nosso sistema e nas operações em que era necessário a consulta da base de dados. No módulo cliente geral a operação de visualização das mensagens de um anúncio não se realiza, infelizmente não foi possível resolver este problema.

Conclusão

No geral, quase todos os objetivos definidos pelo professor foram alcançados com sucesso. Em suma, neste segundo trabalho foi utilizada uma arquitetura diferente comparando com a que foi utilizada na primeira versão do mesmo, o que nos permitiu desenvolver as nossas competências, tanto na linguagem Java, como na cadeira de Sistemas Distribuídos.