

Room Rent sd02

A associação de estudantes precisa rever o sistema para gerir os anúncios de quartos para arrendamento. O seu trabalho é implementar este novo sistema com arquitetura distribuída, que inclua:

1. Cliente geral
 - a) Registrar novos anúncios (oferta ou procura), sendo que o anúncio fica em estado inativo até ser aprovado pelo gestor. O sistema atribui um novo código único, que será devolvido como resultado da operação. Cada anúncio terá localização, preço, género, data, anunciante, tipologia (quarto, T0, T1...), e poderá ser de oferta de alojamento, ou de procura. Tem ainda um estado (inativo, ativo), preço e descrição. Os dados devem ser guardados numa BD, no servidor.
 - b) Listar anúncios (com estado ativo) de ambos os tipos.
 - c) Procurar anúncios, enviando texto a pesquisar na descrição, e opcionalmente uma localização.
 - d) Obter todos os detalhes de um anúncio, dado o seu identificador (aid).
 - e) Enviar nova mensagem ao anunciante, pelo identificador do anúncio (aid). E consultar as mensagens inseridas para um determinado anúncio (aid).
2. Cliente de Gestão
 - a) Listar anúncios por estado.
 - b) Aprovar um anúncio, ou alterar o estado de um anúncio (ativo/inativo).
3. Serviço em backend (servidor)
 - a) Serviço para as operações a disponibilizar a cada cliente.
(nas listagens do cliente geral, mostrar apenas anúncio em estado ativo)
 - b) Armazenamento persistente de dados.

Valorização superior (5 valores):

- redundância no backend/servidor, para reforçar a disponibilidade do serviço, tolerando falhas;
- mecanismo *publish/subscribe* onde a aplicação cliente pode pedir para ser notificada sobre mensagens de anúncios que tenha criado;
- segurança (comunicação protegida; acesso ao serviço de gestão só para origem autorizada).

Procure, tanto quanto possível, abstrair-se dos detalhes de comunicação, usando formas de comunicação mencionadas nas aulas e não dependentes de uma tecnologia específica. O armazenamento persistente deve usar BD em Postgres. A interação pode fazer-se com uma aplicação de linha de comandos. Aspectos em aberto são decididos pelo grupo e descritos no relatório.

Quaisquer parâmetros de configuração devem estar fora do código, sendo passados como argumento à aplicação ou lidos de um ficheiro de propriedades (ver `java.util.Properties`). A solução implementada deve ser compatível com a plataforma de `alunos.di.uevora.pt`.

Entrega

Os trabalhos devem ser entregues dentro do prazo estabelecido, através do *upload* de um ficheiro .zip no espaço apropriado, no *Moodle*. Esse ficheiro incluirá uma pasta `sd-t02-YYYYY-ZZZZZ` (YYYYY e ZZZZZ são os números de aluno de cada elemento do grupo). Essa pasta deve conter o código fonte e eventuais ficheiros de configuração (incluindo scripts para executar as aplicações necessárias) e um **relatório** com identificação e observações dos alunos.

Os alunos podem trabalhar individualmente ou em grupos de dois elementos.

Prazo: ver moodle