



SeekArtist 2.0 – Sistema de Gestão de Artistas de Rua

1 Descrição da tarefa

A Associação de Artes e Espectáculos **TáNaForja** quer melhorar o sistema de gestão de artistas **SeekArtist**. Para tal, necessita que o sistema se torne mais robusto, escalável e seguro.

O seu trabalho é melhorar as implementações das aplicações **servidor** (SeekServ), **cliente geral** (SeekER), e **cliente administrador** (SeekAdmin), de acordo com os seguintes parâmetros:

• Modelo de Dados

- A localização deve ser dada por um par de coordenadas GPS (latitude, longitude).
- Um artista deve ter a lista de todas localizações por onde já passou (atuou) **bem como** futuras localizações onde irá atuar.
- Há a possibilidade de enviar donativos a artistas. Os donativos devem ser guardados com data, valor, artista e user (que fez o donativo).
- As aplicações cliente passam a necessitar de autenticação. Para isso, deve ser possível registar **Users**, cada um com username, email, password e tipo (administrador|normal).

• Cliente Geral

- Inscrever-se no sistema, indicando o username, email e password pretendidos. (poderá não ser possível, caso já exista um utilizador com o mesmo username ou email).
- Autenticar-se, usando o seu username e password. Só será permitido efetuar as operações seguintes após uma correta autenticação.
- Pedido de registo de um novo artista (se o utilizador encontrar um artista na rua, pede para este entrar no sistema). Aqui, o artista ficará com um estado de não aprovado até ser aprovado por um administrador. Deve ser considerado o facto de, possivelmente, o artista já ter sido introduzido por outro utilizador. Neste caso, deve ser adicionada apenas a localização (caso ainda não tenha sido anteriormente).
- Listar artistas, com filtros (opcionais – no uso, não na implementação) por localização e arte.
- Listar localizações onde existem artistas a atuar.
- Para um determinado artista, identificado por um artistID, listar as datas e localizações onde este já atuou.
- Para um determinado artista, identificado por um artistID, listar a data e localização da próxima (futura) atuação.

- Enviar um donativo a um artista, indicando o valor monetário.
- Listar os donativos recebidos por um determinado artista.
- (bónus) dar uma classificação (*rating*) a um artista.
- (bónus) ver, na listagem de artistas, o *rating* de cada um.

• **Cliente Administrador**

- Autenticar-se, usando o seu `username` e `password`. Só será permitido efetuar as operações seguintes após uma correta autenticação e confirmação da permissão (ser do tipo `administrador`).
- Dar permissão de administrador a um `User`.
- Listar artistas por estado.
- Aprovar um artista que ainda não tenha sido aprovado.
- Consultar e alterar as informações de um artista.
- Todas as operações permitidas pelo cliente geral¹.

• **Servidor**

- (bónus) Suporte para redundância/replicação, de forma a tolerar falhas.
- (bónus) Mecanismo *publish/subscribe* onde a aplicação cliente pode pedir para ser notificada sobre novos artistas ou atuações inseridos, em tempo real.

2 Implementação

Procure, tanto quanto possível, abstrair-se dos detalhes de comunicação e das diferenças de plataforma entre cliente e servidor, usando uma solução de *Middleware* mencionada nas aulas. É desejável que possam ser usadas tecnologias (e.g., linguagens) diferentes nos clientes e servidores, sendo essa possibilidade um dos critérios de avaliação do trabalho.

Não se preocupe com a interface visual. Pode funcionar tudo pela linha de comandos, através de um menu simples.

Quaisquer parâmetros de configuração devem estar fora do código, podendo ser passados à aplicação na forma de argumentos ou através da leitura de um ficheiro de propriedades (ver `java.util.Properties`).

A solução implementada deve ser compatível com a plataforma do servidor `alunos.di.uevora.pt`, onde deverá ter uma BD em PostgreSQL (cujo proprietário é um dos elementos do grupo) para registar os dados do serviço².

¹Nota/dica: Talvez valha a pena ter apenas uma aplicação cliente e diferenciar as operações pelo tipo de utilizador... Para isso, poderá ser necessário “perguntar” ao servidor o tipo de `User`, após a autenticação.

²Por outras palavras, devem construir o sistema com uma base de dados no vosso PC, para poder funcionar fora da Universidade, mas terá de ser possível funcionar no `alunos.di.uevora.pt`.

3 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-YYYY-ZZZZ (YYYY e ZZZZ são os números de aluno de cada elemento do grupo – sem o 'l'). Essa pasta deve conter o código fonte e eventuais ficheiros de configuração (incluindo scripts para executar as aplicações necessárias) e um **relatório** com a identificação e observações dos alunos.

Os alunos podem trabalhar individualmente ou em grupos de dois elementos, sendo recomendado, mas não obrigatório, manter os grupos do primeiro trabalho.

Prazo: janeiro - época normal (*ver moodle*)