

TRABALHO PRÁTICO

Meta 2 - 2 valores

O objetivo desta segunda meta consiste em adicionar, à aplicação Servidor desenvolvida na meta anterior, um **serviço remoto Java RMI** e um **web service (API REST)**.

O **serviço RMI** pretendido deve permitir:

- Listar os servidores ativos, incluindo o servidor que aloja o serviço RMI invocado, com a seguinte informação para cada um: endereço IP, portos de escuta TCP e UDP, carga e tempo decorrido desde a receção do último *heartbeat* (exceto para o próprio servidor);
- Registrar e apagar *listeners* para efeitos de notificações assíncronas (mecanismo de *callback*) sobre: (1) receção de contactos de clientes via UDP, com indicação do respetivo endereço IP e porto de escuta UDP; (2) aceitação de ligações TCP solicitadas por clientes, com indicação do respetivo endereço IP e porto de escuta TCP; (3) perda de ligações TCP com clientes, com indicação do respetivo endereço IP e porto de escuta TCP, bem como do nome de utilizador se for conhecido (clientes autenticados); e (4) *logins* e *logouts* de clientes, com indicação do nome de utilizador.

O serviço remoto associado a cada servidor deve ser registado num *RMI Registry* lançado na máquina onde este se encontra a correr. Para permitir a coexistência de vários servidores na mesma máquina, sugere-se que o nome associado ao serviço RMI de um determinado servidor seja “SHOW_SERVICE_” acrescido do respetivo número do porto de escuta UDP.

Deve também ser desenvolvida uma **aplicação autónoma** que, através da interação com o serviço remoto pretendido, permita utilizar todas as funcionalidades oferecidas por este, incluindo a receção das notificações assíncronas através de um mecanismo de *callback*. A única informação que deve ser fornecida a esta aplicação é o endereço IP da máquina onde são postos a correr o RMI Registry e o serviço remoto, bem como o porto de escuta UDP.

A **API REST (web service)** a desenvolver deve permitir, acedendo diretamente à base de dados local, que um utilizador:

- Se autentique (o estado do utilizador não deve ser alterado na base de dados);
- Obtenha a lista de todos os espetáculos existentes, podendo filtrar o resultado através das datas de início e de fim (fornecidos opcionalmente na URI);
- Obtenha a lista das suas reservas pagas;
- Obtenha a lista das suas reservas que aguardam pagamento;
- Se estiver autenticado como administrador:
 - Obtenha a lista de utilizadores registados;

- Solicite a inserção/registo de um utilizador;
- Solicite a eliminação de um utilizador registado.

Todas as funcionalidades, com exceção da consulta dos espetáculos e da autenticação, podem apenas ser executadas por utilizadores previamente autenticados. Este requisito é garantido através da inclusão de um *token* JWT (devolvido pelo pedido de autenticação) no campo *Authorization* do cabeçalho dos pedidos HTTP (*Authorization: Bearer <token JWT>*).

Na autenticação, deve ser usado o mecanismo básico de autenticação HTTP, em que as credenciais são transmitidas, com codificação base64, no campo *Authorization* do cabeçalho do pedido (*Authorization: Basic <credentials>*). A resposta, caso a autenticação seja bem-sucedida, contém o *token* JWT que deve ser incluído nos pedidos subsequentes.

Como o *token* JWT identifica univocamente um utilizador autenticado, a identidade de quem emite um pedido não deve constar da URI nem do corpo das mensagens. Caso não exista *token*, este seja inválido ou tenha sido emitido há mais de 2 minutos, os pedidos HTTP devem devolver um código de resposta 401 (*Unauthorized*).

No caso de inserção e eliminação de utilizadores, deve ter-se em atenção as condições em que as operações não devem ter sucesso (e.g., utilizador existente / não existente, utilizador identificado como autenticado na base de dados, utilizador com reservas associadas, etc.).

Observações:

- Não é solicitado o desenvolvimento de qualquer tipo de aplicação cliente para a API REST pretendida, aconselhando-se, para efeitos de interação com esta, o recurso à ferramenta Postman.
- Grupos que não tenham a primeira meta minimamente operacional ou que não a tenham entregue devem, em alternativa, desenvolver um serviço RMI que permita que *listeners* (aplicações autónomas) sejam notificados, através de *callbacks*, sempre que um determinado pedido é feito ao *web service* REST desenvolvido;
- Para efeitos de desenvolvimento, teste e apresentação da API REST, a base de dados pode ser povoada de forma manual (ou seja, sem ser através da interação com a aplicação distribuída desenvolvida na primeira meta).

Considerações Gerais

Deve ter-se em consideração que esta segunda meta do trabalho:

- Encontra-se sujeita às mesmas condições gerais da primeira meta;
- Deve ser um acrescento ao que já foi implementado na primeira meta, sendo que o trabalho final a entregar para avaliação deve incluir as duas metas;

- Esta segunda meta do trabalho prático deverá ser entregue até ao dia 3 de janeiro de 2023, às 8h00, através da plataforma InforEstudante, num ficheiro ZIP com a designação PD-22-23-F2-TP-Gx.zip, sendo x o número do grupo.