

# Sistemas Distribuídos — Trabalho Prático

## Troca de ficheiros

Engenharia Informática  
Universidade do Minho

### Informações gerais

- Cada grupo deve ser constituído por até 4 elementos.
- Deve ser entregue o código fonte e um relatório de até 6 páginas (A4, 11pt) no formato PDF.
- A apresentação do trabalho ocorrerá em datas a anunciar e com marcação prévia de hora.

### Resumo

As plataformas de troca de ficheiros como o SoundCloud permitem que músicos partilhem as suas criações diretamente com os seus fãs. Para o efeito, podem carregar ficheiros de música acompanhados de meta-informação variada (título, autor, interprete, género, ...). A meta-informação serve para que os ouvintes tomem conhecimento dos ficheiros partilhados e possam efetuar pesquisas. Tendo encontrado os ficheiros desejados, podem então descarregá-los para uso posterior.

Tendo em conta que os ficheiros a serem trocados são de dimensão considerável, normalmente com vários MB, a concretização destes sistemas tem que prestar particular atenção aos recursos consumidos com o armazenamento, manipulação e transmissão destes ficheiros. Em particular, é importante a limitação do número de operações simultâneas que podem ser efetuadas para não sobrecarregar o sistema e a manutenção de uma justiça relativa entre os diferentes utilizadores.

Neste projeto pede-se a implementação de uma plataforma para partilha de ficheiros de música sob a forma de cliente/servidor em Java utilizando *sockets* e *threads*.

### Funcionalidade Básica

Este serviço deverá suportar a seguinte funcionalidade básica:

- Autenticação e registo de utilizador, dado o seu nome e palavra-passe. Sempre que um utilizador desejar interagir com o serviço deverá estabelecer uma conexão e ser autenticado pelo servidor.
- Publicar um ficheiro de música, fornecendo o seu conteúdo e alguns meta-dados (título, intérprete, ano e um número variável de etiquetas) e recebendo um identificador único.
- Efetuar uma procura de música, enviando uma etiqueta a procurar e recebendo de volta uma lista das músicas que lhe correspondem. Para cada música devem ser também recebidos os seguintes seus meta-dados:

- identificador único;
  - título, intérprete, ano e um número variável de etiquetas, fornecidos quando a música é carregada;
  - número de vezes que a música foi descarregada.
- Descarregar um ficheiro de música, fornecendo o seu identificador único.

## Funcionalidades Adicionais

A resposta será valorizada pelas seguintes funcionalidades adicionais:

### Limite de descargas:

- O servidor deve garantir que em cada momento não há mais do que MAXDOWN descargas de ficheiros a ocorrer em simultâneo.
- Pedidos de descarga que ultrapassem o limite devem esperar até obterem vez.
- Em caso de espera, deve ser estabelecida uma política que combine a ordem de chegada com a necessidade de todos os utilizadores obterem ficheiros.

### Notificação de novas músicas:

- Durante o período em que um cliente está ligado ao servidor, deve receber notificações de novas músicas que sejam carregadas.
- Cada notificação deve conter o título e autor da música.
- Deve ser possível continuar a receber e mostrar notificações mesmo durante outras operações demoradas de carga ou descarga de ficheiros.

### Tamanho dos ficheiros ilimitado:

- A solução apresentada não deve assumir que cada ficheiro de música cabe completamente em memória.
- Pelo contrário, deve garantir que a transferência de cada ficheiro nunca ocupa mais do que MAX-SIZE bytes em memória, tanto do cliente como do servidor.

## Cliente

Deverá ser disponibilizado um cliente que ofereça uma interface com o utilizador que permita suportar a funcionalidade descrita acima. Este cliente deverá ser escrito em Java usando *threads* e *sockets* TCP,

## Servidor

O servidor deverá ser escrito também em Java, usando *threads* e *sockets* TCP, mantendo em memória a informação relevante para suportar as funcionalidades acima descritas (exceto o conteúdo dos próprios ficheiros), receber conexões e input dos clientes, bem como fazer chegar a estes a informação pretendida. O protocolo entre cliente e servidor deverá ser baseado em texto, orientado à linha (exceto o conteúdo dos próprios ficheiros). Para o servidor não ficar vulnerável a clientes lentos, cada thread não deverá escrever em mais do que um socket.