

## 1º Trabalho Prático de Avaliação

**Objectivo:** Desenvolvimento de sistemas distribuídos usando objectos distribuídos na plataforma .NET

**Nota:** O trabalho deve ser realizado até **14 de Maio de 2015**, incluindo um relatório que descreva o trabalho com as opções tomadas ao longo da sua realização. (Enviar, os projetos em Zip file e relatório, para [lass@isel.ipl.pt](mailto:lass@isel.ipl.pt)). Note que, como foi referido na apresentação da disciplina, a qualidade do relatório terá peso na avaliação do trabalho realizado. O relatório deverá permitir ao leitor entender, qual o objetivo, os requisitos funcionais e não funcionais (note que os requisitos não são a transcrição deste enunciado), a arquitetura do sistema, as interações entre as partes, bem como os aspetos relevantes da implementação dessa arquitetura realçando os pontos fortes e fracos da solução que proponha. Evite descrever código a menos que se justifique nalguma situação, por exemplo, as interfaces dos objetos envolvidos.

Considere um cenário de um sistema distribuído com os seguintes requisitos:

- Cada utilizador (Peer), através de uma aplicação gere informação que consiste numa colecção de referências musicais {Artista, Álbum, título, ano, formato: CD, MP3, ...};
- A coleção de referências musicais pode ser armazenada num ficheiro XML com a estrutura que ache mais adequada e fácil de carregar/guardar, por exemplo, através da classe *System.Xml.Serialization.XmlSerializer*;
- Cada *Peer* pode estar conectado a um conjunto de outros *Peers* online (Figura 1);

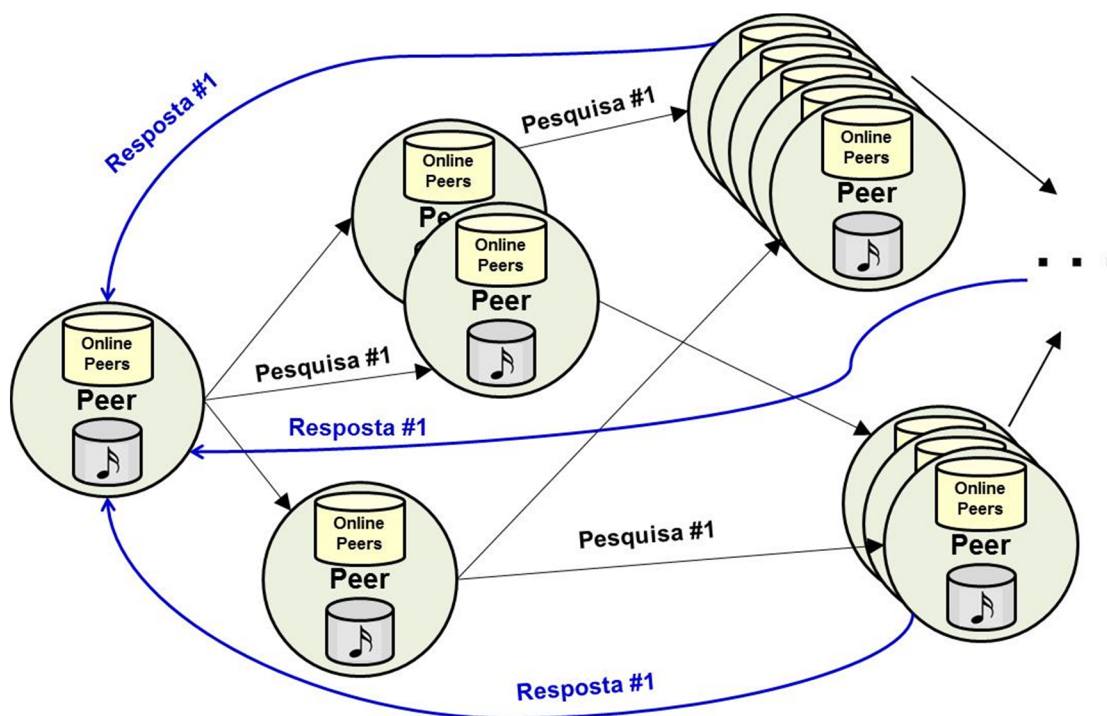


Figura 1 – Sistema distribuído de procura de referências musicais

- Há medida que cada utilizador (*Peer*) vai conhecendo a localização (URL) de outros *Peers* vai criando um registo de *Peers online* de forma a aumentar a sua rede de contactos para futuras pesquisas;
- Quando um utilizador fizer uma pesquisa para uma determinada música que não existe localmente, é enviado um pedido para todos os *Peers* que são conhecidos;
- Os sucessivos *Peers* vão reenviando os pedidos através da rede *Peer-to-Peer* que pode atingir dimensões razoáveis. Assim deve existir um mecanismo que evite que um pedido sem resposta possível (música inexistente) continue a circular na rede *Peer-to-Peer* eternamente;
- A resposta a um pedido pode vir de qualquer *Peer* que pode opcionalmente enviar informações, por exemplo um URL, para posteriormente o utilizador poder obter a música em *mp3*;
- Devem ser tratadas situações de falha, principalmente:
  - ✓ Um determinado *Peer* pode desligar-se ou falhar independentemente dos outros *Peers* que mantêm eventualmente referências para o mesmo;
  - ✓ Dado que um pedido pode demorar tempo até atingir um *Peer* que tem a referência musical, pode acontecer que quando vai ser entregue a resposta o *Peer* que fez o pedido já não está conectado;
- Enquanto um utilizador está a interagir com a aplicação deve ser possível visualizar os pedidos que outros *Peers* estão a fazer ao *Peer* local. Sugere-se a utilização de modo gráfico (*WinForms*) com duas partes, uma para a interação com o utilizador e outra para visualização/monitorização da atividade local do *Peer*;

**Sugestões:**

1. Qualquer questão ou dúvida sobre requisitos, deve ser discutida com o professor;
2. Antes de começar a escrever código, defina os requisitos funcionais e não funcionais, desenhe a arquitetura do sistema, as interfaces dos objetos envolvidos bem como os diagramas de interação mais importantes;
3. Deve utilizar ficheiros de configuração, simplificando assim a demonstração de um cenário com pelo menos 5 *Peers*;
4. Tenha em atenção o tratamento e propagação de exceções para assim o sistema ser mais fiável e permitir tratar as falhas;
5. Tenha em atenção o tempo de vida de objectos remotos;
6. No relatório discuta e justifique as opções tomadas, por exemplo, o modo de ativação de objetos (*SingleCall* ou *Singleton*) bem como as técnicas de manutenção de estado e de controlo de concorrência utilizadas;
7. Não utilize valores *hardcoded* que possam ser parâmetros de configuração, por exemplo URLs, portos etc.;
8. Quando tiver questões sobre os requisitos, verifique no site *Moodle* se existem “*Frequently Asked Questions*” com esclarecimentos sobre o trabalho.