Trabalho Prático de Avaliação Nº 1

Objectivo: Desenvolvimento de aplicações distribuídas usando objectos distribuídos na plataforma .NET

Nota: O trabalho deve ser realizado até 6 de Maio de 2010, incluindo um relatório que descreva o trabalho com as opções tomadas ao longo da sua realização. (Enviar, os projectos em Zip file e relatório, para 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 realçar os pontos fortes e fracos da solução do problema.

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

- Os utilizadores através de uma aplicação cliente gerem uma colecção de perguntas e respectivas respostas sobre temas em que são especialistas;
- A colecção de perguntas/respostas 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*);
- De forma a evitar sobrecarga, existem vários servidores de zona, ligados em anel, permitindo aos utilizadores de uma zona conectarem-se ao servidor dessa zona (Figura 1);

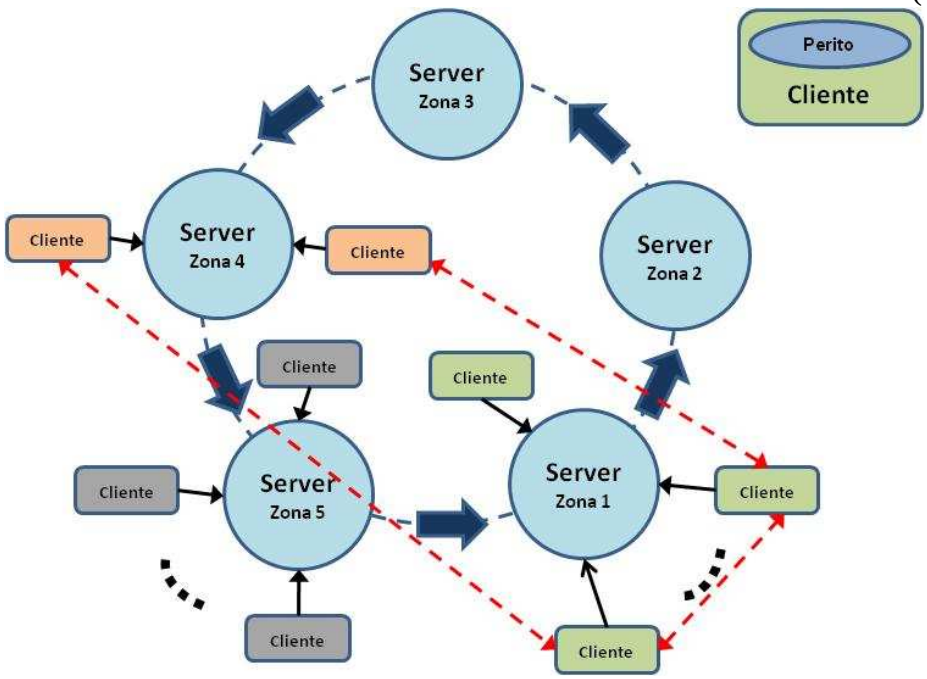

Figura 1 – Sistema distribuído de perguntas/respostas

- De forma a poder ser útil a outros utilizadores, cada aplicação cliente regista-se (*Register*) no servidor da sua zona indicando qual o tema em que é especialista bem como um objecto *Perito* capaz de aceitar posteriormente pedidos remotos sobre perguntas oriundas de outros clientes (linha tracejada a vermelho na Figura 1), que podem estar ligados a qualquer um dos servidores do anel. Note que cada utilizador (aplicação cliente) só conhece um servidor, embora existam diferentes clientes ligados a diferentes servidores;
- Um utilizador deve retirar o seu registo (*Unregister*) antes de ficar indisponível;

- Cada vez que existe um *Register/Unregister* num servidor de zona, este reenvia essa acção para o próximo servidor de zona no anel, provocando, assim, que tendencialmente todos os servidores de zona conhecem todos os especialistas disponíveis;
- Quando um cliente pretende obter respostas sobre um tema, interroga o seu servidor de zona que lhe indica os possíveis especialistas (eventualmente de outras zonas). O cliente pode assim interactuar em modo *Peer-to-Peer* com o objecto *Perito* dos diversos especialistas para obter as respostas pretendidas;
- Devem ser tratadas situações de falha, principalmente:
  - ✓ Assumindo que a estrutura do anel de servidores é conhecida por todos os servidores de zona, a reconfiguração do anel perante a falha dos servidores deve ser dinâmico, isto é, quando o servidor *k* não consegue contactar o próximo no anel *k+1* deve contactar o servidor *k+2*, informando que os especialistas conectados ao servidor que falhou deixaram de estar disponíveis;
  - ✓ A situação em que um cliente falha sem fazer o *Unregister* no servidor de zona, pode ser detectada por qualquer outro cliente que tentou contactá-lo. Nesse caso o cliente deve informar o seu servidor da ocorrência, desencadeando no anel a actualização de não existência do cliente que falhou.
- Enquanto um utilizador está a interactuar com a aplicação cliente deve ser possível visualizar os pedidos que o objecto *Perito* está a satisfazer em cada momento. Sugere-se a utilização de modo gráfico (*WinForms*) com duas partes, uma com a interacção com o utilizador e a segunda com a visualização da actividade do objecto *Perito*;
- **[requisito opcional]** Valoriza-se a possibilidade de o sistema contemplar a possibilidade de o anel poder ser aumentado dinamicamente, isto é, a possibilidade de introduzir um novo servidor de zona ou a reposição de um servidor que anteriormente falhou.

Sugestões:

1. Qualquer questão ou dúvida sobre requisitos, deve ser discutida com o professor;
2. Antes de começar a escrever código desenhe a arquitectura do sistema (servidor e cliente), as interfaces dos objectos envolvidos bem como os diagramas de interacção mais importantes;
3. Utilizar ficheiros de configuração tanto no servidor como no cliente, simplificando assim a construção de um protótipo de demonstração com pelo menos 3 servidores, com clientes conectados a diferentes servidores;
4. Tenha em atenção o tratamento e propagação de excepções 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, modo de activação de objectos (*SingleCall* ou *Singleton*) bem como as técnicas de controlo de concorrência utilizadas.

Luís Assunção