

1º Trabalho Prático de Avaliação

Nuno Veloso 42181
Steven Brito 42798
Daniela Gomes 42799

3 de Maio de 2018

Resumo

Com o intuito de realizar o 1º trabalho prático da unidade curricular Sistemas Distribuídos pretende-se implementar um sistema distribuído para suportar a troca de mensagens textuais e instantâneas entre pessoas ou grupos num cenário de uma grande área geográfica dividida por regiões. Ao longo deste trabalho iremos aplicar os conceitos aprendidos nas aulas. Iremos descrever e discutir as vantagens, os problemas e os desafios que se colocam no desenvolvimento deste sistema distribuído.

Conteúdo

1	Introdução	3
2	Descrição do Problema	4
3	Requisitos	5
3.1	Funcionais	5
3.2	Não Funcionais	5
4	Arquitetura	6
4.1	Interação entre as partes	6
4.2	Funcionamento	6
4.2.1	Funcionamento entre utilizadores	7
4.2.2	Funcionamento em grupo	7
5	Implementação	8
5.1	Central Manager	8
6	Tolerância a Falhas	10
7	Manual de utilização	11

Capítulo 1

Introdução

Neste trabalho irá ser desenvolvido um sistema distribuído capaz de trocar mensagens entre utilizadores dentro da mesma região ou entre regiões. Desta forma são precisos vários servidores regionais - um servidor para cada região - e um servidor central.

Capítulo 2

Descrição do Problema

O problema consiste em desenvolver um sistema distribuído capaz de trocar mensagens instantâneas, ou seja, sem haver o armazenamento das mesmas, entre utilizadores dentro da mesma região ou entre regiões. É necessário evitar sobrecarga nos servidores. É também necessário garantir que os vários acessos simultâneos aos servidores que mantenham a consistência dos dados. O sistema desenvolvido deve ter em conta tolerância a falhas.

Capítulo 3

Requisitos

Neste capítulo é descrito os requisitos funcionais do sistema, assim como os não funcionais.

3.1 Funcionais

Os requisitos funcionais deste sistema são os seguintes apresentados:

- Evitar sobrecarga no servidor central;
- O cliente comunica apenas com o servidor da sua região;
- O cliente tem um identificador único;
- O cliente conhece todos os servidores regionais;
- O cliente regista-se num servidor regional;
- O cliente pode enviar mensagens para um único cliente ou para um grupo onde este pertença;
- O cliente pode mudar de região;
- O cliente pode criar grupos;
- O grupo pode ter clientes pertencentes a várias regiões;
- Os servidores regionais e centrais devem conhecer a estrutura dos grupos;
- Os clientes que não estejam conectados não recebem as mensagens.

3.2 Não Funcionais

Os requisitos não funcionais deste sistema são os seguintes apresentados:

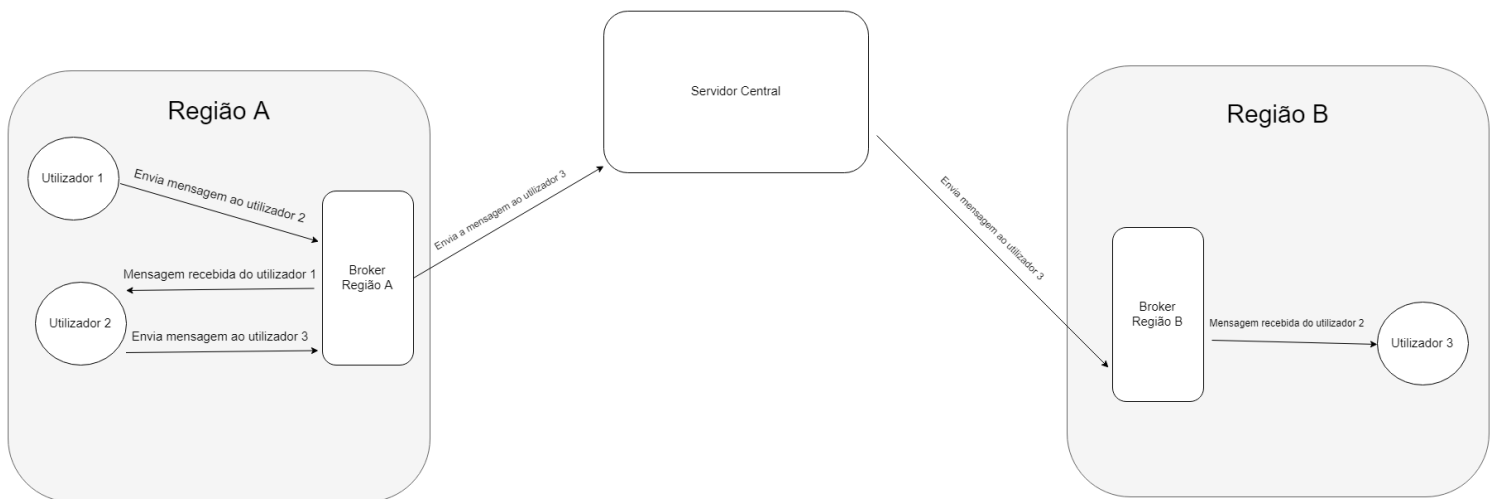
- A aplicação cliente realizada em WinForm;
- Transparência à concorrência no acesso dos servidores.

Capítulo 4

Arquitetura

4.1 Interação entre as partes

Um utilizador apenas interage com o servidor regional onde este está registado. O servidor regional comunica-se com o utilizador e com o servidor central. O servidor central apenas interage com os servidores regionais.



O utilizador é cliente do servidor regional. O servidor regional é servidor do utilizador e cliente do servidor central. O servidor central é servidor do servidor regional.

4.2 Funcionamento

Para poder utilizar o sistema, o utilizador deve escolher em qual região pretende-se conectar. Estando este conectado, é possível usufruir das seguintes funcionalidades:

- Enviar mensagens para o utilizador;
- Criar grupos;
- Enviar mensagens para um grupo;
- Trocar de região;
- Sair de uma região.

4.2.1 Funcionamento entre utilizadores

Para um utilizador enviar uma mensagem para outro utilizador da mesma região o processo ocorre apenas no servidor dessa região. Caso o utilizador deseje enviar uma mensagem para outro utilizador fora da sua região, o servidor da região irá comunicar-se com o servidor central dizendo para este tratar de enviar a mensagem para um determinado utilizador que o servidor regional não tem conhecimento. O servidor central irá enviar a mensagem a todos os servidores regionais que conhece e estes irão verificar se o utilizador encontra-se registado na respetiva região. Em caso afirmativo tratarão de enviar a mensagem para o utilizador pretendente. Em caso negativo, o servidor regional não comunica com nenhum utilizador.

4.2.2 Funcionamento em grupo

É possível ao utilizador criar grupos dentro de uma região e adicionar outros utilizadores quer estejam na mesma ou em diferentes regiões. A existência do grupo é replicada por todos os outros servidores regionais. Ao adicionar um utilizador de outra região, o servidor regional envia a informação ao servidor central, que por sua vez envia aos outros servidores regionais até que a região onde se encontra o utilizador receba a mensagem e o adiciona ao grupo. Aos utilizadores que pertençam a um grupo, estes têm permissão para adicionar outros utilizadores. Os utilizadores que pertencem a um grupo nunca podem sair desse grupo, só sairão quando o criador apagar o grupo.

Capítulo 5

Implementação

A solução encontra-se dividida em vários projetos de bibliotecas, aplicações de consola e uma aplicação WinForm.

5.1 Central Manager

Para ser implementado o servidor central, foi necessário definir a interface `ICentralManager`, apresentada na figura 5.1.

```
public interface ICentralManager
{
    void RegisterGroup(Group group, IBroker callerBroker);

    void AddUserToGroup(string groupName, int destNumber, IBroker callerBroker);

    void SendMessageToBrokers(int receiver, Message message, IBroker callerBroker);

    void SendMessageToGroup(string groupName, Message message, IBroker callerBroker);

    void UnregisterGroup(string groupName, IBroker callerBroker);
}
```

Figura 5.1: Interface `ICentralManager`

O *central manager* conhece todos os servidores regionais existentes, pois é injetado no construtor uma lista com todos os *proxies* dos servidores regionais desse sistema. Desta forma, o *central manager* é *stateless* visto que não armazena mais nenhuma informação.

Para a implementação do *central manager*, foi usado o padrão *singleton*. O padrão *singleton* adequa-se melhor às necessidades, pois assume-se que o servidor central estará sempre a receber e a enviar pedidos. Desta forma o padrão *single call* teria muito maior *overhead* em relação ao *singleton* pois teria de estar sempre a criar novas instâncias para responder a cada pedido, sendo que não se sabe o número de pedidos de antemão.

De modo a que não é possível determinar o uso deste sistema distribuído, não existe forma de saber se é usado com bastante regularidade e com espaçamento no intervalo de tempo entre as mensagens; No pior dos casos tem o objeto em memória com tempo infinito e os pedidos nunca passam pelo *manager*, e no melhor dos casos os pedidos são atendidos sempre pela

mesma instância, sem esta estar a ocupar recursos em memória desnecessariamente.

O *lease time* escolhido para a instância do *central manager* foi o tempo por omissão do CLR do .NET que é de 5 minutos e o tempo de renovação por cada chamada é também de 5 minutos. Foi tomada essa decisão porque não é possível prever o intervalo de tempo entre cada mensagem consecutiva. O problema com esta decisão é se o intervalo entre cada mensagem for superior a 5 minutos. Isto irá causar o *overhead* da criação do *manager*. A vantagem desta solução é que se o intervalo entre as mensagens for superior a

Capítulo 6

Tolerância a Falhas

Com esta arquitetura existem cenários em que o servidor central possa falhar e as trocas de mensagens entre regiões não aconteça, mas que dentro de cada região continue a funcionar. Se o servidor central voltar a conectar-se, irá realizar uma nova conexão com os servidores regionais que conhece. Esta solução permite a troca de mensagens entre utilizadores dentro da mesma região enquanto não estiver conectado um servidor central.

Um outro cenário seria um servidor regional a falhar. Neste caso só a região desse servidor é que não conseguiria trocar mensagens. Mensagens de outras regiões também não chegariam à região afetada. Caso o servidor central envie mensagens para este servidor regional, tal não é possível, pois este servidor foi desconectado. Esta solução tem um inconveniente. O servidor central tem guardado informação de servidores regionais que podem já não estar conectados. O servidor central não tem maneira de identificar se o servidor regional foi desconectado ou se é um problema de comunicação. Se um servidor regional falhar e ficar ativo novamente, irá conseguir conectar-se com o servidor central. O problema desta solução é perder a informação dos utilizadores e dos grupos existentes.

Se um cliente desconectar-se sem informar o servidor regional, os dados ficam guardados nesse servidor, pois não há forma de saber que o cliente tem problemas de comunicação ou se está mesmo desconectado. Uma solução alternativa passaria por contar o número de chamadas consecutivas que o servidor regional faz ao cliente e este não responde com sucesso. Teria de ser atribuído um número máximo de chamadas consecutivas falhadas, e eliminar a informação deste cliente no servidor regional após o número de chamadas ultrapassar o limite. O problema com esta solução é que o cliente poderia estar novamente disponível após o limite de tentativas, e assim o servidor regional iria apagar informação de um cliente ativo.

Capítulo 7

Manual de utilização