

# Sistemas Distribuídos

## *3ª Serie - Relatório*

**Grupo:**

- G05D

**Alunos:**

- João Pires 31933
- José Casimiro 32713
- António Dente 33168

## Índice

Introdução .....	3
Arquitectura .....	4
Serviços .....	5
ICentralService .....	5
ICService .....	5
Servidor .....	6
Tratamento de falhas .....	6
Cliente .....	7
Conclusão .....	7

## Introdução

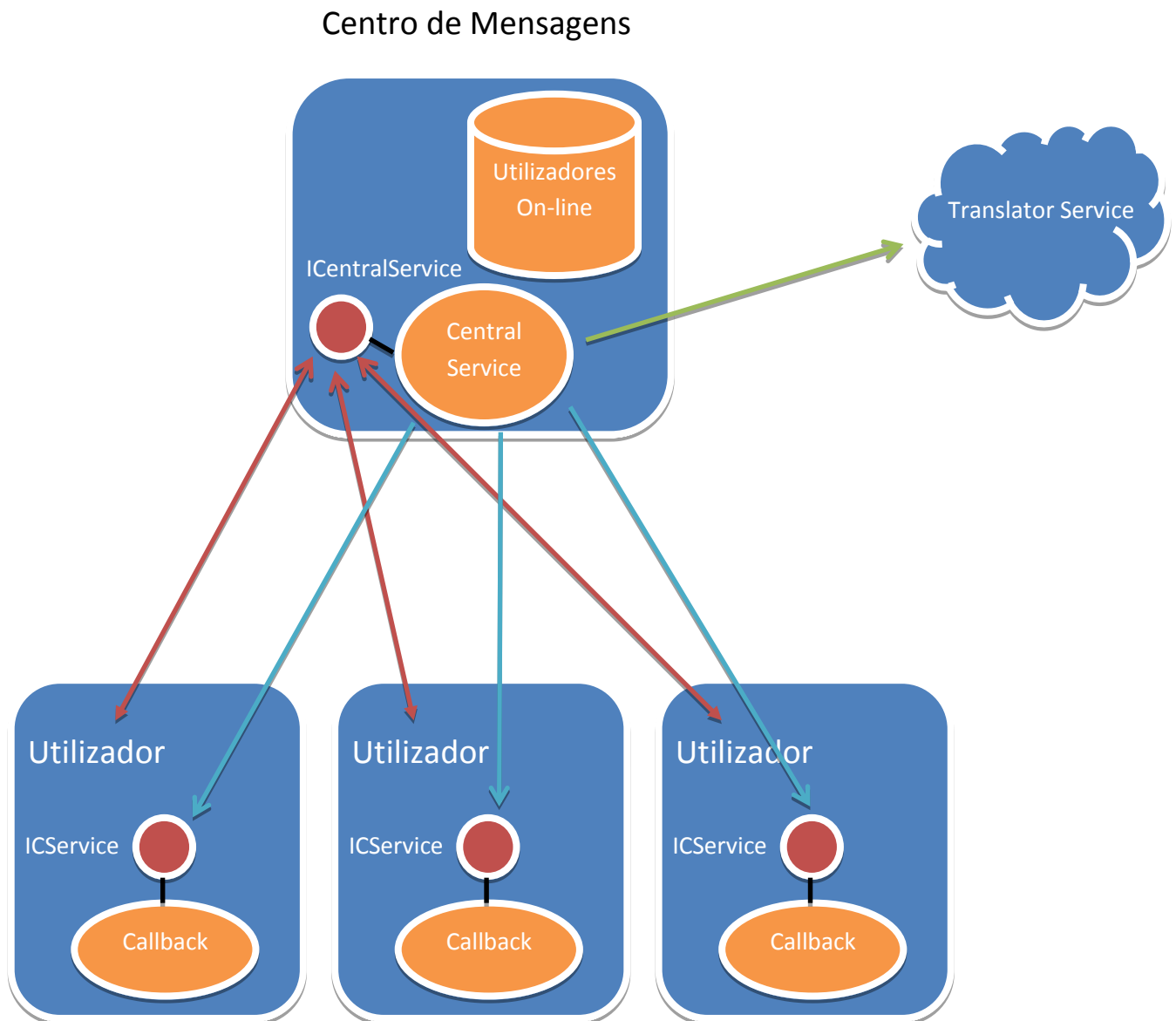
Neste trabalho foi-nos pedido que realizássemos um sistema distribuído baseado em serviços WCF (Windows Communication Foundation).

WCF é a plataforma de programação para criação, configuração e *deployment* de *network-distributed services* da Microsoft.

Este sistema deve conter um Centro de mensagens onde vários utilizadores se podem registar e trocar mensagens de vários temas. Cada utilizador, no ato de registo, indica a sua linguagem preferencial e o tema onde se pretende registar, quando uma nova mensagem é submetida sobre aquele tema o serviço central remete essa mensagem para todos os utilizadores correntemente registados no tema. A mensagem é traduzida para a linguagem definida por cada utilizador (caso necessário) através de um serviço de tradução. Neste caso estamos a utilizar o serviço disponibilizado pela Microsoft em [www.microsofttranslator.com](http://www.microsofttranslator.com). As mensagens são fornecidas pelos próprios clientes para o servidor, ou seja, qualquer mensagem enviada pelo utilizador vai ser submetida para o tema em que o utilizador em questão se registou e na linguagem definida por este.

## Arquitectura

De forma a implementar o sistema descrito no enunciado, foi definido um serviço de registo de clientes e um serviço para receção de mensagens. Foram também criadas duas aplicações consola, uma para ser o servidor que faz hosting do serviço de registo, e outra que faz hosting do serviço de receção de mensagens. A imagem seguinte mostra o esquema geral da solução:



— Chamada do servidor ao serviço de tradução

— Utilizador chama métodos de registo e “desregisto” da interface ICentralService

— Servidor chama método de callback exposto pela interface IService

## Serviços

Para a definição dos contractos dos serviços foi criada uma biblioteca que é partilhada tanto pela aplicação cliente como pelo servidor. Ai definimos o serviço `ICentralService` e o serviço `ICService`.

### `ICentralService`

Este serviço é o serviço principal, hosted por uma aplicação servidora. Este serviço tem associado a ele um contracto de callback, `ICService`, que está definido e explicado seguidamente.

Neste serviço estão definidas as seguintes operações:

```
Register(int theme, int language): int;  
UnRegister(int clientId);  
GetSupportedLanguages(): string [];  
GetSupportedThemes(): string [];  
SubmitMessage(string msg);
```

### `ICService`

Este serviço, sendo um contracto de callback para `IcentralService`, define apenas uma operação:

```
Receive(string msg);
```

Esta operação está definida como `IsOneWay`, visto não ser esperado qualquer tipo de feedback após a sua chamada.

## Servidor

A nossa aplicação que faz hosting do serviço ICentralService tem uma implementação desse serviço, onde é definido o comportamento das operações.

Essa implementação foi definida com um modo de concorrência múltiplo, dando assim a possibilidade de vários clientes acederem a este serviço concorrentemente. O modo de instanciação é PerCall, e visto os campos que são partilhados pelas instâncias serem campos de tipo não existe a necessidade de definir o modo de instanciação do serviço como singleton.

Para realizar o registo de clientes, utilizámos um repositório estático com uma estrutura de dados concorrente, ConcurrentDictionary, desta forma os acessos feitos por múltiplas instâncias do serviço a este repositório estão controladas. No registo de um novo cliente, é guardada a informação de tema e linguagem desse utilizador, é criado um ID único para ele e guardada uma referência para o serviço de callback associado.

Além da subscrição e revogação de clientes a este servidor, existe também a possibilidade de um cliente fazer um envio de uma mensagem ao servidor que por sua vez irá enviar a todos os clientes interessados. Essa mensagem terá de ser traduzida, utilizando o serviço de tradução do Bing, e enviada a todos os clientes interessados. O envio de mensagens é feito de forma assíncrona. Para suportar a utilização de um serviço externo foi criada uma WebReference para esse serviço.

## Tratamento de falhas

No servidor a falha principal que podemos observar é não existir conexão para com o serviço que estamos a tentar comunicar, essas situações ocorrem na tentativa de envio de mensagem por callback a um cliente, e a chamar o serviço de tradução. Para tal, nesses locais, estamos atento á possibilidade de falha de conexão removendo o cliente do repositório, ou não traduzindo a mensagem, caso os serviços associados estejam *offline*.

## Cliente

Como contracto no lado cliente colocámos uma interface `IService` que contém um único método (*Receive(string msg)*) que serve de callback para receber as mensagens que definimos com [*OperationContract(IsOneWay = true)*].

Para efeitos de testes realizámos a concretização de cliente em consola. Antes do registo no servidor pedimos o tema e a linguagem, ambos obtidos do servidor, e seguidamente registamos o cliente com as escolhas. Após o registo o cliente fica à espera de *input* por parte do utilizador, este input, caso exista, é enviado para o servidor como uma submissão de mensagem sobre o tema. Quando o cliente obtém uma mensagem através do callback mostra-a na consola. Caso a tentativa de ligação ao servidor falhe, o utilizador é informado que o servidor se encontra *offline*.

## Conclusão

A realização deste trabalho permitiu-nos compreender e aprofundar alguns conceitos relacionados com arquiteturas orientadas a serviços bem a utilização de web services. Permitiu ainda a introdução ao grupo da infraestrutura WCF.