

18 de Maio de 2011

*Instituto Superior de Engenharia de Lisboa*

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Sistemas Distribuídos 2010/2011

Relatório do 2º Trabalho Prático

**Discentes:**  
Ana Correia, 31831

João Silvestre, 32466

**Discentes:**  
A

**Docente:**  
Luís Assunção

.

Índice

[1. Introdução 5](#_Toc297655838)

[1.1 Objectivos e Descrição do Trabalho 5](#_Toc297655839)

[1.2 Organização do documento 5](#_Toc297655840)

[2. Estrutura do Trabalho 6](#_Toc297655841)

[2.1 Introdução 6](#_Toc297655842)

[2.2 Estrutura 6](#_Toc297655843)

[2.2.1 Serviço Central 6](#_Toc297655844)

[2.2.2 Serviço de Chat 6](#_Toc297655845)

[2.3 Problemas Identificados 6](#_Toc297655846)

[3. Trabalho Desenvolvido 7](#_Toc297655847)

[3.1 Introdução 7](#_Toc297655848)

[3.2 Implementação do Trabalho 7](#_Toc297655849)

[3.2.1 Serviço Central 7](#_Toc297655850)

[3.2.1.1 Tipos Expostos 8](#_Toc297655851)

[3.2.2 Serviço Chat 8](#_Toc297655852)

[3.2.2.1 Tipos Expostos 8](#_Toc297655853)

[3.3 Tratamento de Erro 9](#_Toc297655854)

[3.4 Configurações dos Serviços 9](#_Toc297655855)

[4. Conclusões 10](#_Toc297655856)

[Bibliografia 11](#_Toc297655857)

# 1. Introdução

Este documento consiste no relatório do primeiro trabalho da disciplina de Sistemas Distribuídos. Neste relatório aborda-se a arquitectura da aplicação, detalhes da sua implementação e os problemas encontrados.

## Objectivos e Descrição do Trabalho

Este trabalho consiste na realização de um sistema de conversação online (chat) que contem salas de chat identificadas por temas e permite a tradução da conversa para o idioma que cada cliente escolheu.

Para suportar esta funcionalidade devem existir dois serviços, implementados recorrendo à tecnologia WCF, sendo que um deles deverá ser um serviço central ao qual o outro serviço recorre para se registar numa sala de chat.

O segundo serviço é específico à aplicação sendo que é a partir de aqui que as aplicações podem enviar mensagens para as outras aplicações registadas no mesmo tema, é também este que é responsável por traduzir as mensagens que chegam dos outros membros para a língua pretendida.

## 1.2 Organização do documento

Este documento está dividido em 4 secções.

Na secção 2 consta a descrição do planeamento do trabalho, nomeadamente a sua estrutura, problemas detectados e as soluções apresentadas.

Na secção 3 apresenta-se as técnicas usadas durante a implementação e os problemas encontrados.

Por fim a secção 4 contém as conclusões do trabalho realizado.

# 2. Estrutura do Trabalho

## 2.1 Introdução

O trabalho, como pedido, encontra-se dividido em três partes. O serviço central, o serviço de chat e um cliente.

## 2.2 Estrutura

### 2.2.1 Serviço Central

Este serviço é o responsável pelo registo dos utilizadores nas “salas de chat” dos temas, é aqui que é guardada a informação sobre quem se encontra online num determinado tema. Um serviço de chat quando se regista no serviço central indica o nome de utilizador, o tema em que se pretende registar e o seu Uri, para que os outros possam estabelecer uma ligação para este. O serviço central no momento de registo obtém um callback, para o serviço de chat, para mais tarde o poder notificar quando um novo utilizador se liga. O registo retorna um array com todos os utilizadores que já se encontram ligados.

### 2.2.2 Serviço de Chat

Este serviço é o responsável por estabelecer ligações com outros serviços de chat para “conversar” é também este que comunica com o serviço central de forma a conseguir se registar nas salas de chat. Quando uma aplicação cliente se subscreve no serviço este obtém um callback para este, de forma a poder avisar quando uma nova mensagem é entregue. Mas antes de entregar a mensagem este pede primeiro ao serviço de tradução para que traduza a mensagem para o idioma escolhido pelo cliente.

## 2.3 Problemas Identificados

O problema identificado passa por dois clientes tentarem registar-se sobre o mesmo nome na mesma sala, é preciso conseguir avisar o cliente que deverá escolher um outro nome.

# 3. Trabalho Desenvolvido

## 3.1 Introdução

Nesta secção irá ser explicado a implementação dos dois serviços realizados para este trabalho(Serviço Central e Serviço Chat) e o cliente.

Também será explicado os problemas detectados durante a implementação e as suas resoluções.

## 3.2 Implementação do Trabalho

## 3.2.1 Serviço Central

Na implementação do serviço decidimos expor as seguintes operações:

* Theme[] GetThemes()
* User[] LogOn(string themeName, string userName, Uri address);
* void LogOff(string themeName, long id);

O método LogOn tem como funcionalidade criação de um novo *user*, utilizando o *userName* recebido como parâmetro e retornar um array de Users que se encontram, nesse momento, a discutir sobre o tema escolhido pelo o utilizador. É neste método que se obtêm um callback, para o serviço que chamou o método, para que seja possível, mais tarde, notifica-lo da entrada ou saída de um utilizador. Ao criar este novo user, todos os utilizadores que se encontram a discutir o tema pretendido, são notificados da entrada do novo utilizador criado. Este método lança a excepção *InvalidOperationException*, quando o utilizador que se pretende criar já se encontra no serviço.

O método LogOff tem como funcionalidade eliminar o registo do utilizador que contem o id recebido como parâmetro. Ao eliminar o registo do user, todos os utilizadores que se encontram a discutir o tema em que, este se encontrava, são notificados da sua saída da sala de chat. Este método lança a excepção *InvalidOperationException*, quando o utilizador que se pretende eliminar já não se encontra no serviço.

Cada chamada ao serviço é sempre utilizada a mesma instancia para todos o clientes, porque o serviço guarda estado dos users ligados.

### 3.2.1.1 Tipos Expostos

O CentralService expõe 2 tipos de dados e 1 de callback, os tipos de dados são o User e o Theme que descrevem os utilizadores e os temas, respectivamente. O callback obriga a implementação de dois métodos de notificação de quando um utilizador entra e outro quando o utilizador sai.

## 3.2.2 Serviço Chat

Na implementação do serviço decidimos expor as seguintes operações:

* void SendMessage(string message)
* void ReceiveMessage(Message message)
* void Subscribe(string username, string theme, string language)
* void Unsubscribe()

O método SendMessage tem como funcionalidade enviar a mensagem para todos os utilizadores que se encontram no chat. Caso não consiga enviar a mensagem para algum do cliente, é removido da lista do chat. Este método encontra-se com o atributo OneWay = true, porque este método não retorna uma resposta e não lança uma excepção.

O método ReceiveMessage tem como funcionalidade receber uma mensagem de um utilizador que tenha escrito no chat. No momento da recepção é feito um pedido ao serviço de tradução para traduzir a mensagem para a língua pretendida pelo cliente. Este método encontra-se também com o atributo OneWay=true, pelas mesmas razões que o método SendMessage.

O método Subscribe tem como funcionalidade de registar o utilizador no serviço Central, caso este lance excepção, porque o utilizador encontra-se já registado no serviço, então a excepção é propagada para o cliente.

O método Unsubcribe tem como funcionalidade notificar o serviço central que o utilizador já não tem interesse que continuar no chat.

### 3.2.2.1 Tipos Expostos

O ChatService expõe 1 tipos de dados e 1 de callback, o tipo de dados é o Message que descreve as mensagens do chat. O callback obriga a implementação de um método para receber as mensagens dos outros utilizadores.

## 3.3 Tratamento de Erro

Todos os métodos passíveis de mandar excepção têm um FaultContract de forma a avisar o cliente que este método poderá lançar excepção. Neste trabalho apenas foi utilizada uma excepção (InvalidOperationException) que por ser do runtime já é serializável, no entanto, se fosse criada por nós teríamos que garantir que esta é serializável.

Apenas o cliente trata algum tipo de erro e trata apenas quando o utilizador já existe, pois este poderá acontecer por input do utilizador. A única outra situação em que o serviço poderá lançar excepção é quando se tenta remover um utilizador que não existe mas isto apenas irá acontecer caso exista algum erro na implementação. Caso já exista o utilizador o tratamento dado é pedir ao utilizador para inserir um novo utilizador.

## 3.4 Configurações dos Serviços

Ambos os serviços têm bindings wsDualHttpBinding pois precisam de utilizar callbacks para os clientes.

# 4. Conclusões

Consideramos ter conseguido implementar todas as funcionalidades pedidas do enunciado, no entanto, no decorrer dos testes efectuados à nossa solução detectamos que se um cliente cair e de seguida outro se registar no mesmo endpoint, quando este se regista no serviço central ele irá obter um uri para si mesmo. O que irá resultar na criação de um cliente para si mesmo fazendo com que este receba as mensagens que envia até que outro utilizador se ligue, provocando que o serviço central detecte que o cliente que caiu já não se encontra activo, resultando na divulgação da sua eliminação da rede de contactos activos, corrigindo assim o problema.

Tirando este problema, pensamos ter uma implementação robusta e livre de problemas que possam afectar o seu funcionamento.

# Bibliografia

Assunção, L. (2011). Slides de Apoio à Cadeira de Sistemas Distribuídos.

Microsoft. (s.d.). *.NET Remoting Overview*. Obtido em 05 de 2011, de MSDN: http://msdn.microsoft.com/en-us/library/kwdt6w2k(v=vs.71).aspx

# 