



ESCOLA SUPERIOR DE  
TECNOLOGIA E GESTÃO DE LEIRIA  
INSTITUTO POLITÉCNICO DE LEIRIA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

**Sistemas Distribuídos e Paralelos**

**2005/2006**

**Engenharia Informática**

2.º ANO regime Diurno

3.º ANO regime Nocturno

**Engenharia Informática e Comunicações**

2.º ANO

---

## **CuscoRMI**



Relatório da implementação do serviço  
*CuscoRMI* do sistema de *Buscas*  
*Distribuídas*.

**Documento elaborado por:**

Sérgio Miguel Neves Lopes – aluno n.º 10635

**Versão 1.0**

Janeiro/2006

---

# ÍNDICE

---

---

<b>1</b>	<b>Introdução.....</b>	<b>2</b>
<b>1.1</b>	<b>Descrição Geral do Trabalho .....</b>	<b>2</b>
1.1.1	Descrição da Quarta Etapa – Implementação do <i>CuscoRMI</i> .....	2
1.1.2	Nota às Etapas Anteriores .....	2

---

<b>2</b>	<b>Descrição Classes/Interfaces que Compõem a Solução RMI</b>	
<b>3</b>		
2.1	Descrição da classe <i>Cusco</i> .....	3
2.2	Descrição da interface <i>Notifiable</i> .....	3
2.3	Descrição da classe <i>ClientRMI</i> .....	3
2.4	Descrição da classe <i>ServerRMI</i> .....	4
2.5	Descrição da interface <i>CuscoServices</i> .....	4

---

<b>3</b>	<b>Diagrama de Classes .....</b>	<b>5</b>
<b>4</b>	<b>Justificação de Opções Tomadas.....</b>	<b>5</b>
<b>5</b>	<b>Bibliografia .....</b>	<b>7</b>

# 1 Introdução

## 1.1 Descrição Geral do Trabalho

O BUD – Buscas Distribuídas, pretende ser um sistema distribuído com a capacidade para, regularmente, efectuar a pesquisa de ficheiros no parque informático de uma instituição.

### 1.1.1 Descrição da Quarta Etapa – Implementação do *CuscoRMI*

Esta segunda etapa tem por objectivo a implementação do serviço de pesquisa denominado *Buscas*.

O objectivo é a implementação de um servidor híbrido permita a pesquisa de ficheiros e pastas nas diferentes máquinas.

### 1.1.2 Nota às Etapas Anteriores

Não foram corrigidos os erros das etapas anteriores, nomeadamente das classes *Cusco* e *CuscoGUI*.

A classe *FileInfo* passou a implementar a interface *Serializable*.

## 2 Descrição Classes/Interfaces que Compõem a Solução RMI

### 2.1 Descrição da classe *Cusco*

Em relação à etapa anterior a classe *Cusco* viu ser removida a implementação da interface *ClientHandler* e a necessidade de uma instância de *GUIHandler* no seu construtor.

O método **search** sofreu uma alteração na visibilidade passando a ser um método privado. Todas as referências feitas à componente gráfica foram substituídas por referências a objectos do tipo *Notifiable*.

Foi criado um novo método **search** de visibilidade pública, consequência da implementação da interface *CuscoServices* por parte da classe *Cusco*, que permite continuar a disponibilizar o serviço de pesquisa, desta forma, também como serviço remoto.

Esta classe é portanto o objecto remoto que disponibiliza os métodos de forma a ser possível efectuar e controlar pesquisas.

### 2.2 Descrição da interface *Notifiable*

A interface *Notifiable* define o comportamento de todos os elementos que pretendem ser notificados do progresso da pesquisa permitindo desta forma implementar um mecanismo de callback.

### 2.3 Descrição da classe *ClientRMI*

Implementação de um cliente que permite efectuar pedidos de pesquisa.

Esta implementação pretende definir um cliente que possibilite a execução de pesquisas através da linha de comandos. Todos os parâmetros necessários à execução das pesquisas são passados por linhas de comandos e todas as mensagens geradas durante a pesquisa, bem como os resultados da mesma, são dirigidos para o *stdout*.

A classe é composta apenas por um método **main**, onde são validados os parâmetros de entrada, e pela implementação dos métodos definidos pela interface *Notifiable*

## 2.4 Descrição da classe **ServerRMI**

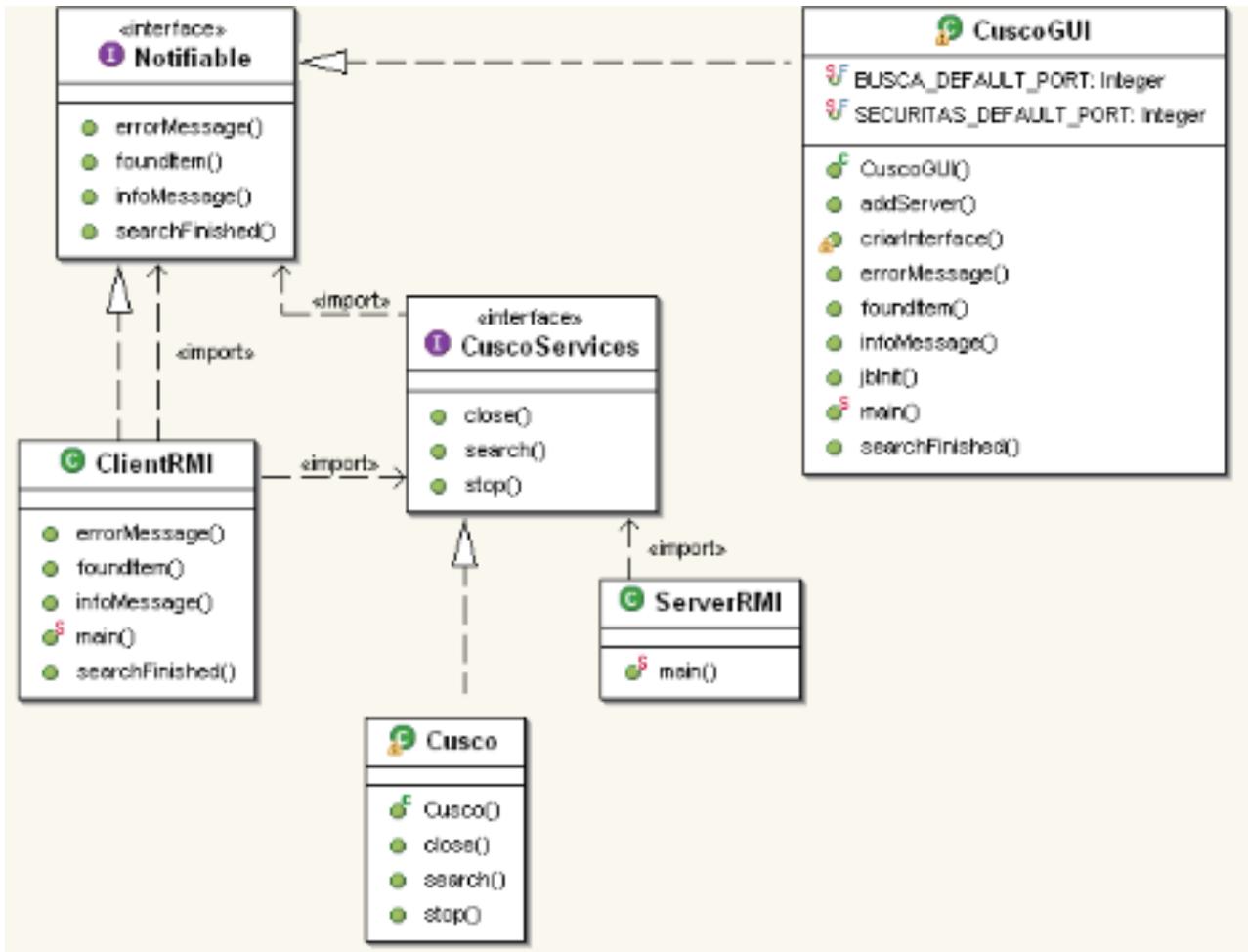
Classe que permite tornar disponível o objecto remoto *Cusco*

## 2.5 Descrição da interface **CuscoServices**

Interface remota que define os serviços que o *Cusco* disponibiliza remotamente.

Estão agrupados nesta interface todos os serviços que se consideraram importantes para a execução e manutenção de uma pesquisa.

### 3 Diagrama de Classes



### 4 Justificação de Opções Tomadas

Na implementação da solução RMI foi considerado que a interface criada na etapa anterior passaria também a aceder ao método de pesquisa através de métodos remotos, assim foi considerado que o método de pesquisa é apenas disponibilizado remotamente e qualquer cliente implementado tem forçosamente de usar o mecanismo de RMI. Como resultado desta implementação foi criada uma interface comum a todos os clientes que pretendam utilizar o serviço de pesquisa que o *cusco* disponibiliza.

Ao ser iniciada, a interface gráfica, tenta criar um servidor RMI, caso não seja possível por já existir um servidor para o mesmo objecto remoto, é obtido o objecto, desta forma é possível

iniciar a aplicação sem iniciar o *ServerRMI* e permite que, se a interface gráfica e a aplicação servidor se encontrarem na mesma máquina apenas uma instância do objecto remoto é criada e usada ao longo de todos os pedidos.

## 5 Bibliografia

Apontamentos teóricos da disciplina de Sistemas Distribuídos e Paralelos.

“The Java Tutorial”, actualização de 15 de Abril de 2005, versão de *download*.