

FURB – Universidade Regional de Blumenau
Centro de Ciências Exatas e Naturais
Curso de Ciências da Computação - Noturno

Comparativo da Tecnologia
DCOM em vários ambientes de
desenvolvimento

Acadêmico: Egard Charles Ullrich
Orientador: Mauricio Capobianco Lopes

Roteiro

- **Introdução**
- **DCOM—Distributed Component Object Model**
- **Ferramentas que suportam a tecnologia DCOM**
- **Desenvolvimento**
- **Conclusões / Sugestões**

Introdução

- O que foi desenvolvido com o trabalho
- Objetivo do desenvolvimento do trabalho
- Importância do trabalho

O que foi desenvolvido com o trabalho

- Estudo comparativo da tecnologia DCOM nos ambientes propostos
- Desenvolvimento de aplicações clientes no estilo *chat* para troca de informações entre os usuários através da tecnologia DCOM
- Desenvolvimento de um servidor de informações capaz de gerenciar o fluxo de informações dos clientes

Objetivo do desenvolvimento do trabalho

- Estudar a interoperabilidade da tecnologia DCOM em diversos ambientes de desenvolvimento
- Disponibilizar um protótipo de comunicação pessoal capaz de provar a funcionalidade da tecnologia DCOM nos ambientes Visual C++, Visual J++, Visual Basic e Delphi

Importância do trabalho

- Reutilização de código
- Redução do tempo e custo de desenvolvimento de aplicações distribuídas

DCOM – Distributed Component Object Model

- COM – Component Object Model
- Interfaces COM
- DCOM – Distributed Component Object Model

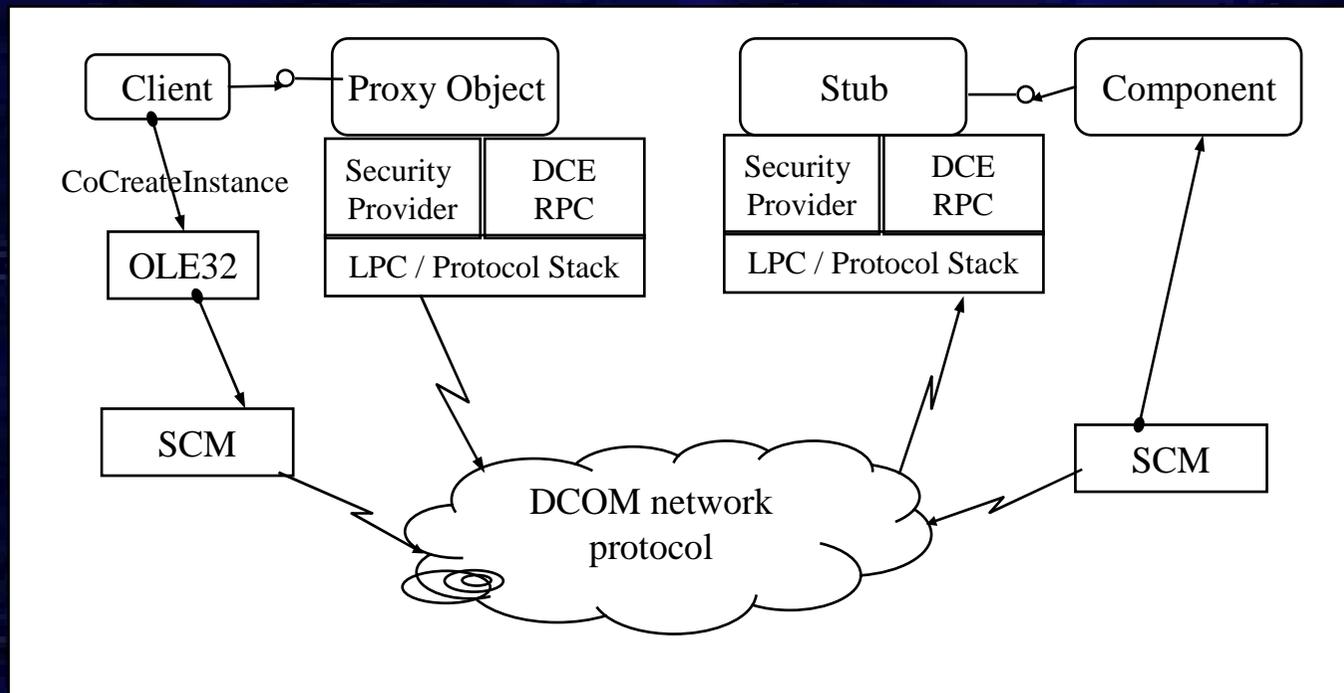
COM – Component Object Model

- Surgimento
- O que é ?
 - Arquitetura de software componentizada
 - Biblioteca COM como API do Windows
 - Empacotados como .exe ou .dll
- Pode ser suportada por diversas linguagens

Interfaces COM

- Linguagem independente e possui IDL própria
- Componentes COM tem acesso restrito por interfaces
- Suporte a tipos de interfaces básicas
 - Interface *IUnknown*
 - Interface *IDispatch*

O que é DCOM e como ele funciona



Ferramentas que suportam a tecnologia DCOM

Os ambientes utilizados para demonstração da tecnologia DCOM neste trabalho são:

- Visual C++
- Visual J++
- Visual Basic
- Delphi

Desenvolvimento

- Requisitos do Sistema
- Diagrama de Casos de Uso
- Diagrama de Seqüência
- Diagrama de Classes
- Protótipos Desenvolvidos

Requisitos do Sistema

- Sistema permite ao usuário conectar-se a uma rede de computadores gerenciada por um servidor.
- Sistema permite que um usuário envie mensagens para um usuário específico ou todos os usuários conectados na rede via COM/DCOM.
- Sistema permite que um usuário receba as mensagens destinadas a ele via COM/DCOM.
- Sistema atende as necessidades do problema através de uma LAN (*Local Area Network*).

Diagrama de Casos de Uso

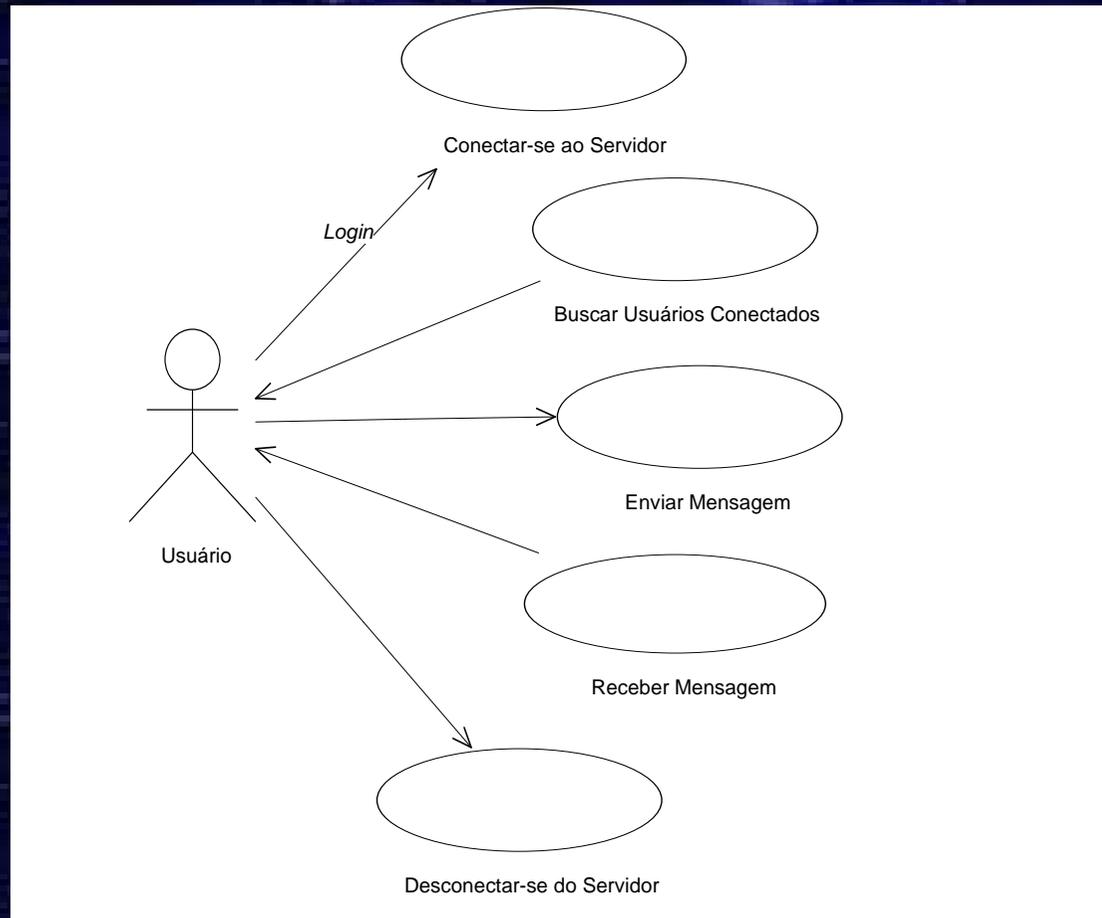
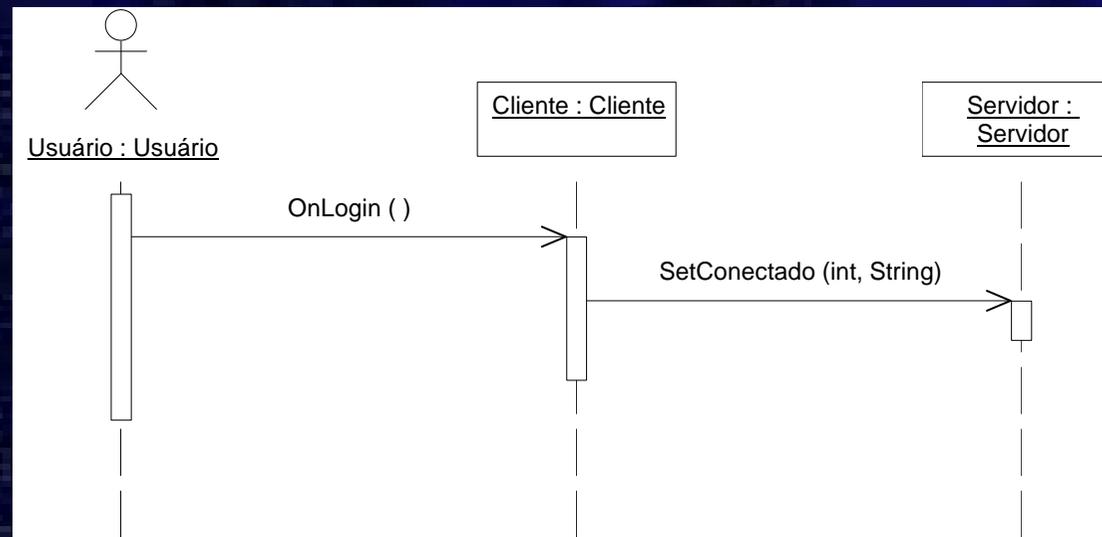


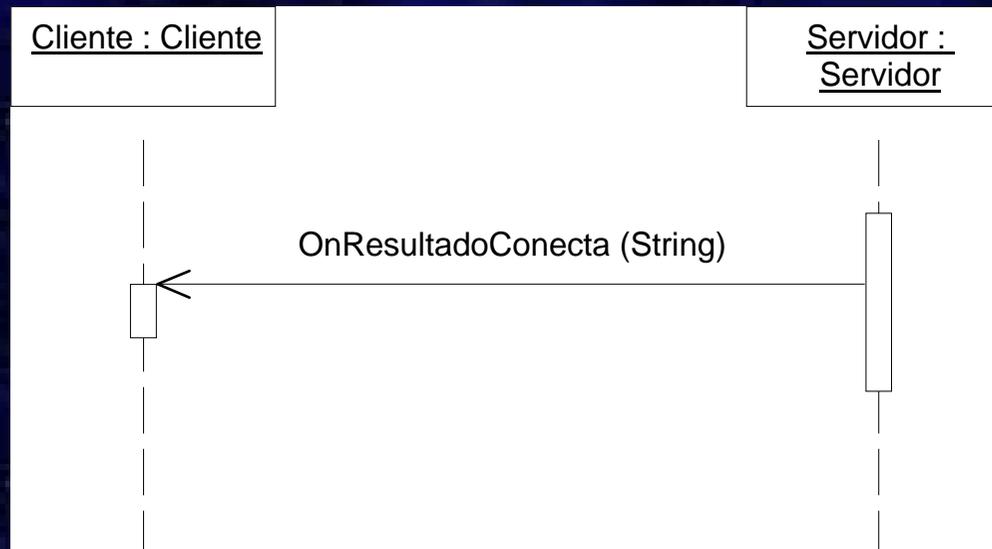
Diagrama de Seqüência

- Conectar-se ao Servidor
- Buscar usuários conectados
- Enviar Mensagem
- Receber Mensagem
- Desconectar-se do Servidor

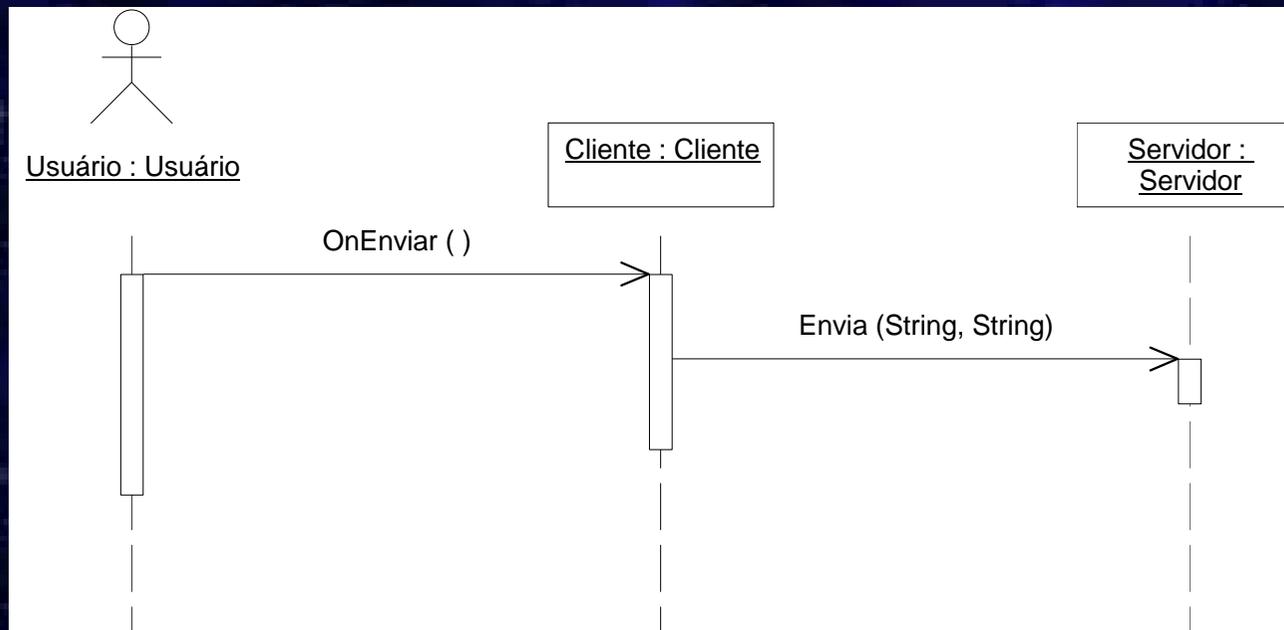
Conectar-se ao servidor



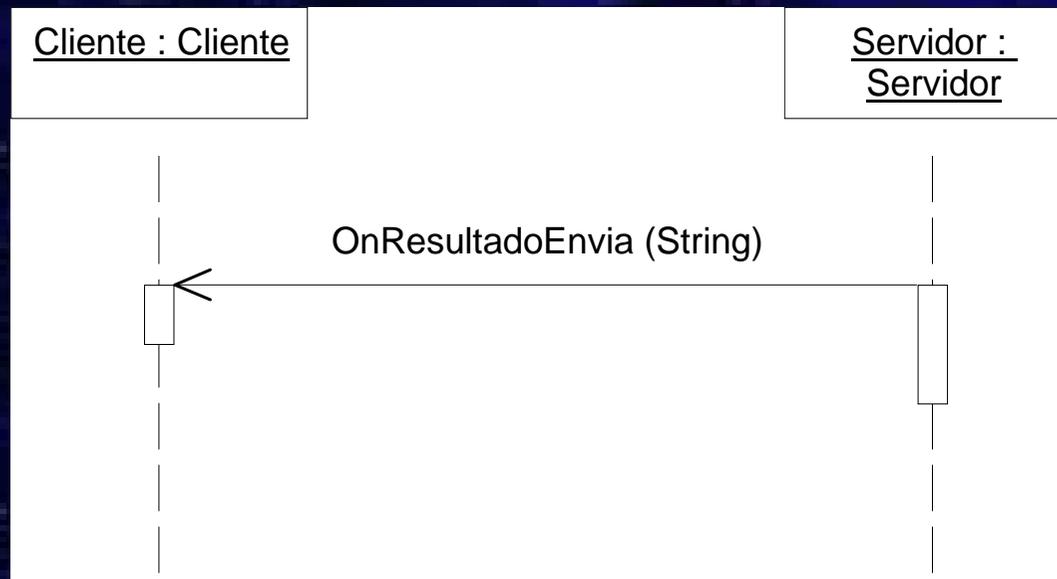
Buscar usuarios conectados



Enviar Mensagem



Receber Mensagem



Desconectar-se do Servidor

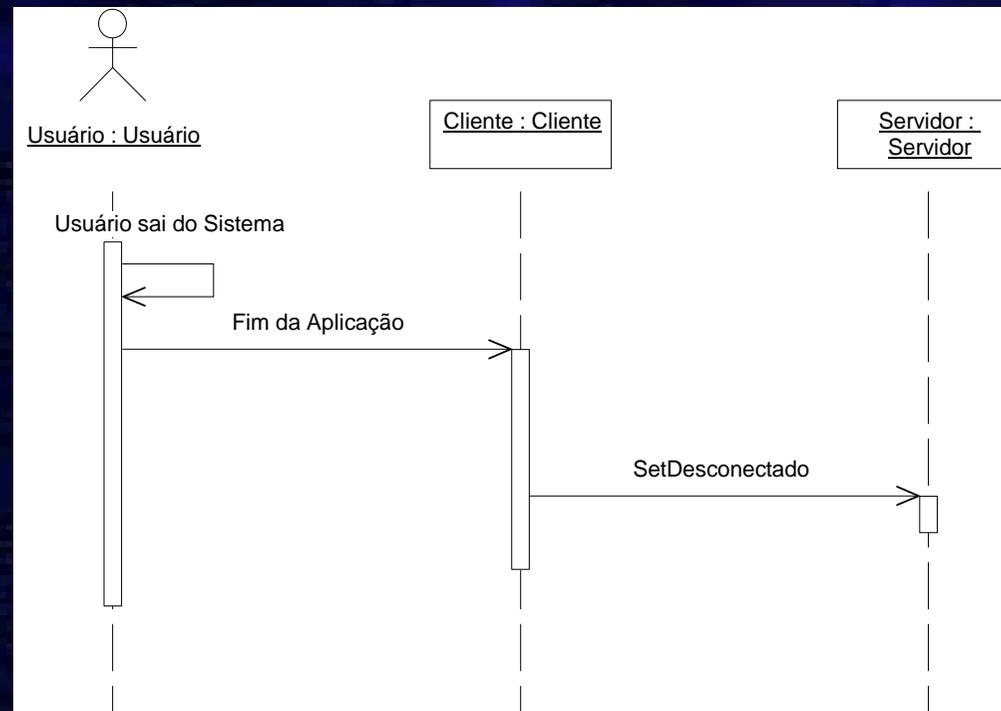
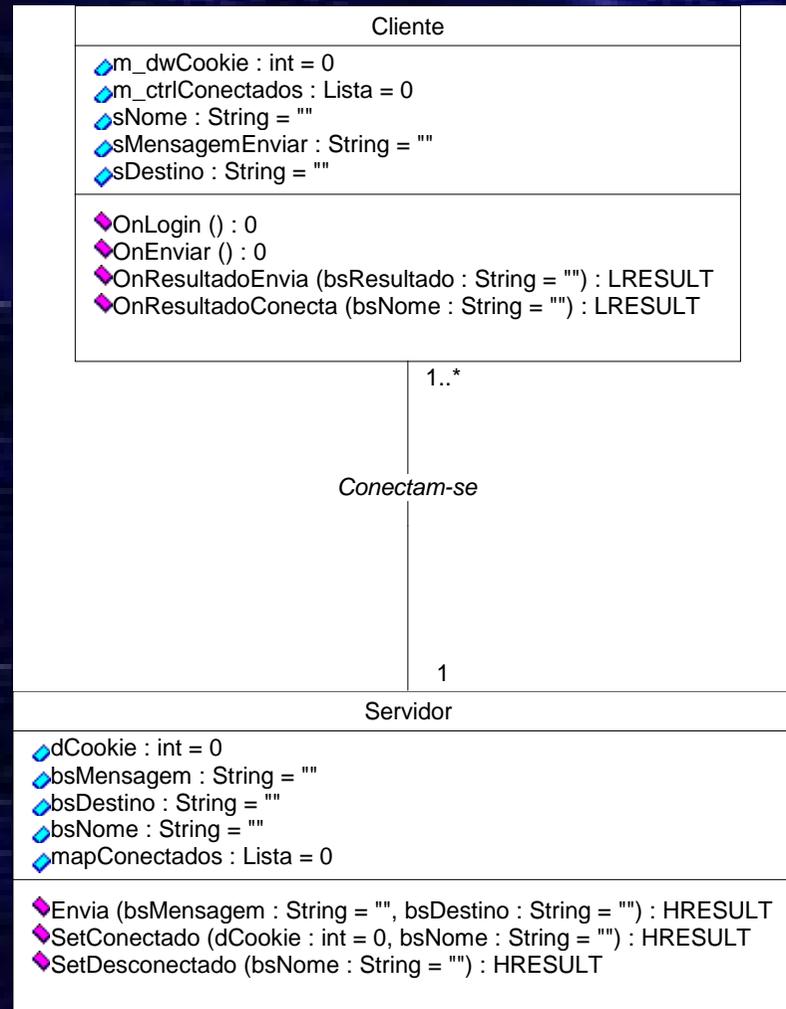
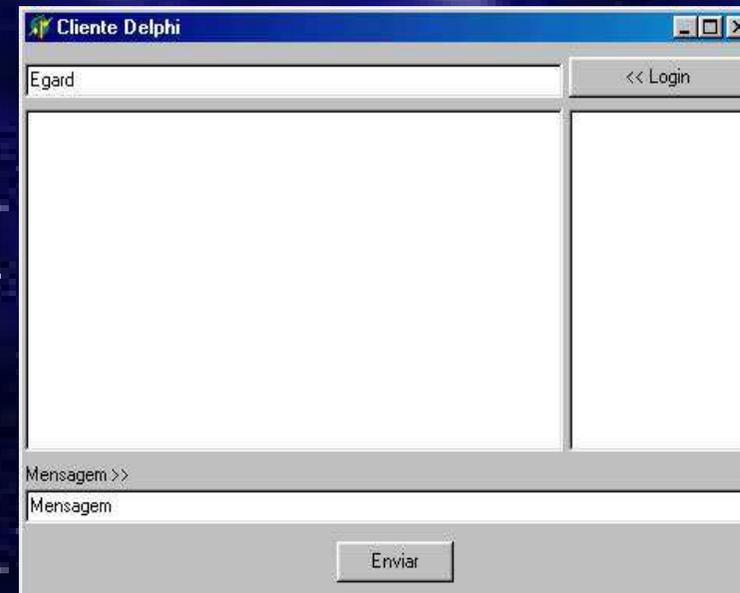
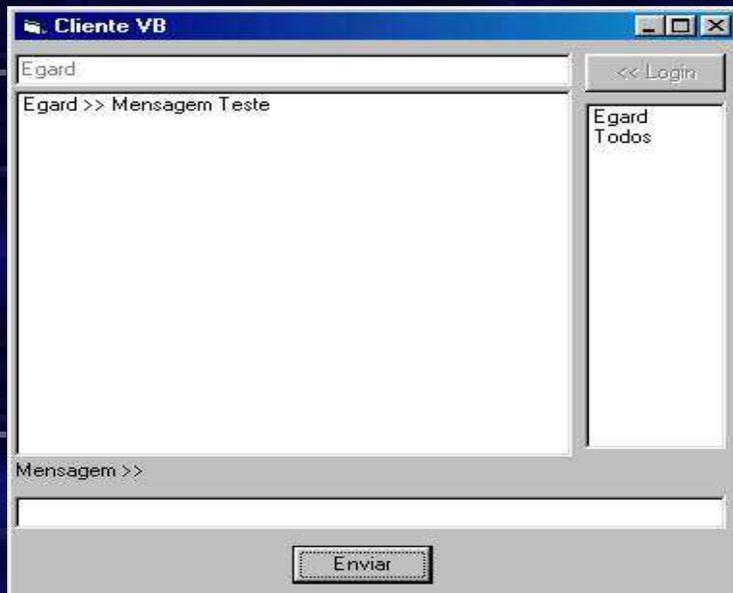
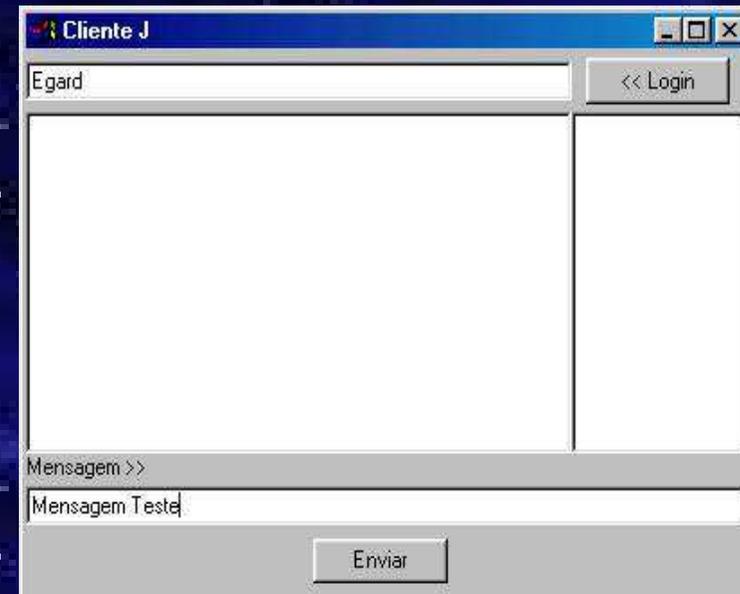
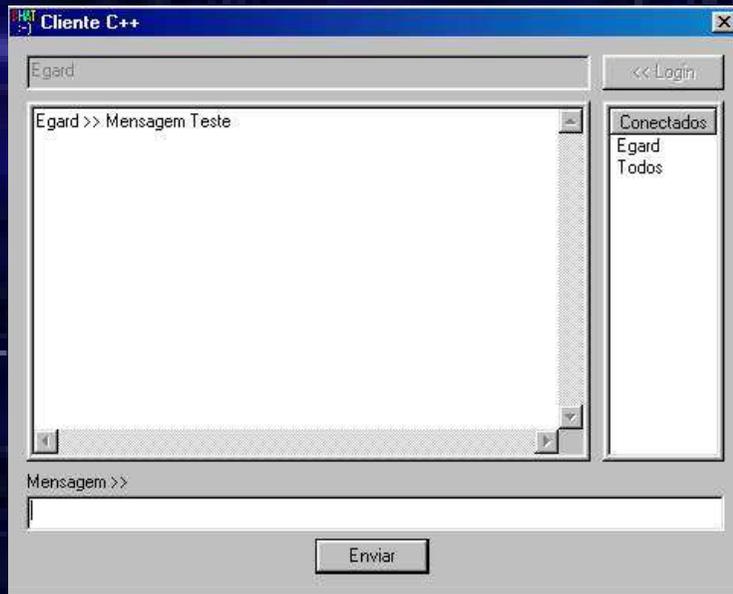


Diagrama de Classes



Protótipos Desenvolvidos



Como Executar o Servidor a partir dos Clientes

- **Em Visual C++**

```
CoCreateInstance(CLSID_InterfaceExemplo, NULL,  
CLSCTX_ALL, IInterfaceExemplo, &m_pIExemplo);
```

- **Em Visual J++**

```
pServidor = new InterfaceExemplo();
```

- **Em Visual Basic**

```
Set objServ = New InterfaceExemplo
```

- **Em Delphi**

```
InterfaceExemplo.Create(nil);
```

Como Conectar-se ao Servidor

- **Em Visual C++**

```
m_pIExemplo->SetConectado(m_dwCookie, bsNome);
```

- **Em Visual J++**

```
pServidor.SetConectado(dwConectado, sLogin);
```

- **Em Visual Basic**

```
Call objServ.SetConectado(dwConectado, sNome)
```

- **Em Delphi**

```
InterfaceExemplo.SetConectado(dwConectado, sNome);
```

Como Enviar uma Mensagem

- **Em Visual C++**

```
m_pIExemplo->Envia(bsMensagem,bsDestino);
```

- **Em Visual J++**

```
pServidor.Envia(sMensagem, "Todos");
```

- **Em Visual Basic**

```
Call objServ.Envia(sMensagemEnviar, sDestino)
```

- **Em Delphi**

```
InterfaceExemplo.Envia(sMensagem, sDestino);
```

Resultado da implementação

- Foi possível implementação da tecnologia em todas os ambientes propostos.
- Maior dificuldade de implementação em Visual C++.
- Semelhança na utilização da tecnologia em Visual Basic e Visual J++.
- Implementação do DCOM é feita de forma mais superficial em Visual Basic e Visual J++ do que em Visual C++.
- As maiores dificuldades de implementação se deram na utilização da técnica de *Callback* principalmente no ambiente Visual J++.

Conclusões

- **Objetivos alcançados**
- **Dificuldades encontradas**
- **Conhecimentos adquiridos em relação as facilidades e vantagens fornecidas pela tecnologia**

Sugestões

- Ampliar as funcionalidades do *chat* com sistema de *log* e armazenamento de mensagens para usuários não conectados
- Estudo comparativo da tecnologia COM+
- Desenvolvimento de aplicações distribuídas através da plataforma .NET via internet