

Sistema PowerMessage - Especificação de Requisitos e Projeto

Versão 2.0
01 de abril de 2004

AVISO

**Trabalho desenvolvido por Rodrigo Alvares de Souza
(r_souza01@yahoo.com.br)**

Ao utilizar o documento, citar a fonte.

1. ESPECIFICAÇÃO DE REQUISITOS.....	6
1.1. CONTEXTO	6
1.2. PROCESSO ATUAL	6
1.3. FUNCIONALIDADES	6
1.4. RESTRIÇÕES TÉCNICAS.....	6
1.5. INTERAÇÃO COM OUTROS SISTEMAS.....	6
1.6. TIPOS DE USUÁRIOS E OPERADORES	7
1.7. VIABILIDADE ECONÔMICA	7
1.8. VIABILIDADE LEGAL.....	7
1.9. VIABILIDADE TÉCNICA.....	7
2. DESCRIÇÃO DA ARQUITETURA DO SISTEMA.....	8
2.1. ARQUITETURA DE HARDWARE DO SISTEMA.....	8
2.2. ARQUITETURA DE SOFTWARE DO SISTEMA	8
2.3. ARQUITETURA DE SEGURANÇA	8
3. ALOCAÇÃO DE FUNÇÕES AOS ELEMENTOS DO SISTEMA	9
3.1. IDENTIFICAÇÃO DOS ELEMENTOS DE HARDWARE	9
3.1.1. <i>Servidor</i>	9
3.1.2. <i>Cliente</i>	9
3.2. IDENTIFICAÇÃO DOS ELEMENTOS DE SOFTWARE	9
3.2.1. <i>Servidor</i>	9
3.2.2. <i>Cliente</i>	9
4. MODELAGEM DO SISTEMA.....	10
4.1. IDENTIFICAÇÃO DOS ELEMENTOS EXTERNOS E DOS DADOS DE ENTRADA E/OU SAÍDA	10
4.2. DETALHAMENTO DOS CASOS DE USO.....	10
4.2.1. <i>CASO DE USO 1</i>	10
4.2.2. <i>CASO DE USO 2</i>	10
4.2.3. <i>CASO DE USO N</i>	10
4.3. MODELO OBJETO	11
4.4. DIAGRAMAS DE COLABORAÇÃO.....	12
4.4.1. <i>Cenário 1 : Tentativa de Login Bem sucedida</i>	12
4.4.2. <i>Cenário 2 : Envio de Mensagem Bem Sucedido</i>	12
4.4.3. <i>Cenário 3 : Cadastro de novo usuário bem sucedido</i>	13
4.5. IDENTIFICAÇÃO DE COMPONENTES	13
4.6. DESCRIÇÃO DE COMPONENTES	14
4.6.1. <i>Componente InetLib</i>	14
4.6.2. <i>Componente ATLMsgServer</i>	14
4.7. DESCRIÇÃO DOS DADOS	14
4.7.1. <i>Modelo Entidade Relacionamento</i>	14
4.7.2. <i>Descrição dos Comandos de Comunicação</i>	16
5. MÓDULOS	18
5.1. MÓDULO POWERMESSAGE	18
5.1.1. <i>Narrativa de Processamento</i>	18
5.1.2. <i>Descrição da Interface</i>	18
5.1.3. <i>Módulos e Componentes Utilizados</i>	18
5.2. MÓDULO POWERSERVER	18
5.2.1. <i>Narrativa de Processamento</i>	18
5.2.2. <i>Descrição da Interface</i>	18
5.2.3. <i>Módulos e Componentes Utilizados</i>	18
6. NAVEGAÇÃO DO SISTEMA - DIAGRAMA DE TELAS	19
6.1. TELA DE LOGIN	19

6.2.	TELA DE CADASTRAMENTO DE NOVO USUÁRIO.....	19
6.3.	TELA PRINCIPAL DO SISTEMA	20
6.4.	TELA DE SETUP DO SISTEMA	20
6.5.	MENU DE CONTEXTO PARA INTERAÇÃO COM USUÁRIOS	21
6.6.	TELA DE ENVIO DE MENSAGEM	21
6.7.	TELA DE RECEBIMENTO DE MENSAGEM.....	22
6.8.	DIAGRAMA DE NAVEGAÇÃO	22
7.	PLANO DE TESTES	23
7.1.	CONFIGURAÇÃO	23
7.2.	DOMÍNIO	23
7.3.	LIMITES	24
7.4.	DESEMPENHO	24
7.5.	VOLUME.....	24
8.	OBTENÇÃO DO SOFTWARE	26
8.1.	PARTICIONAMENTO DO SISTEMA.....	26
8.2.	ROTEIRIZAÇÃO DAS PARTIÇÕES	27
8.3.	SUMÁRIO GERENCIAL DO PLANO DE OBTENÇÃO DE SOFTWARE	27
9.	REFERÊNCIAS BIBLIOGRÁFICAS.....	28

Índice de Figuras

Figura 1 - Diagrama de Contexto - Sistema PowerMessage	10
Figura 2 - Diagrama de Objetos	11
Figura 3 - Cenário 1	12
Figura 4 - Cenário 2	12
Figura 5 - Cenário 3	13
Figura 6 - Diagrama Entidade-Relacionamento do sistema PowerMessage	15
Figura 7 - Particionamento do Sistema.....	26

1. Especificação de Requisitos

1.1. Contexto

O sistema a ser desenvolvido tem por finalidade disponibilizar através da *Intranet* um sistema de mensagens similar aos *messengers* disponíveis hoje na Internet

1.2. Processo Atual

Não há.

1.3. Funcionalidades

O sistema deve permitir a troca de mensagens instantâneas pela intranet da corporação, para facilitar a troca de mensagens em momentos onde possam haver telefones ocupados, e também ser mais rápido que o correio eletrônico corporativo. O sistema também deve permitir a autenticação de usuários em qualquer computador da corporação, e a inclusão de usuários da mesma maneira.

1.4. Restrições Técnicas

Em termos de desempenho, o sistema deve emitir e receber mensagens de usuários em tempo menor ou igual a 15 segundos.

O volume de informações a serem armazenadas pode ser considerado baixo, levando-se em consideração que haverá somente a persistência de usuários e mensagens.

Com relação à Intranet, ela deve suportar o protocolo TCP/IP.

1.5. Interação com Outros Sistemas

Prevê-se que será necessária a conexão com a Intranet corporativa.

1.6. Tipos de Usuários e Operadores

Usuário Padrão – Usuários responsáveis pelo cadastramento deles próprios, alteração de cadastro e envio de mensagens;

Administrador - Pessoas responsáveis pela manutenção e instalação do sistema.

1.7. Viabilidade Econômica

Não se aplica.

1.8. Viabilidade Legal

Não se aplica.

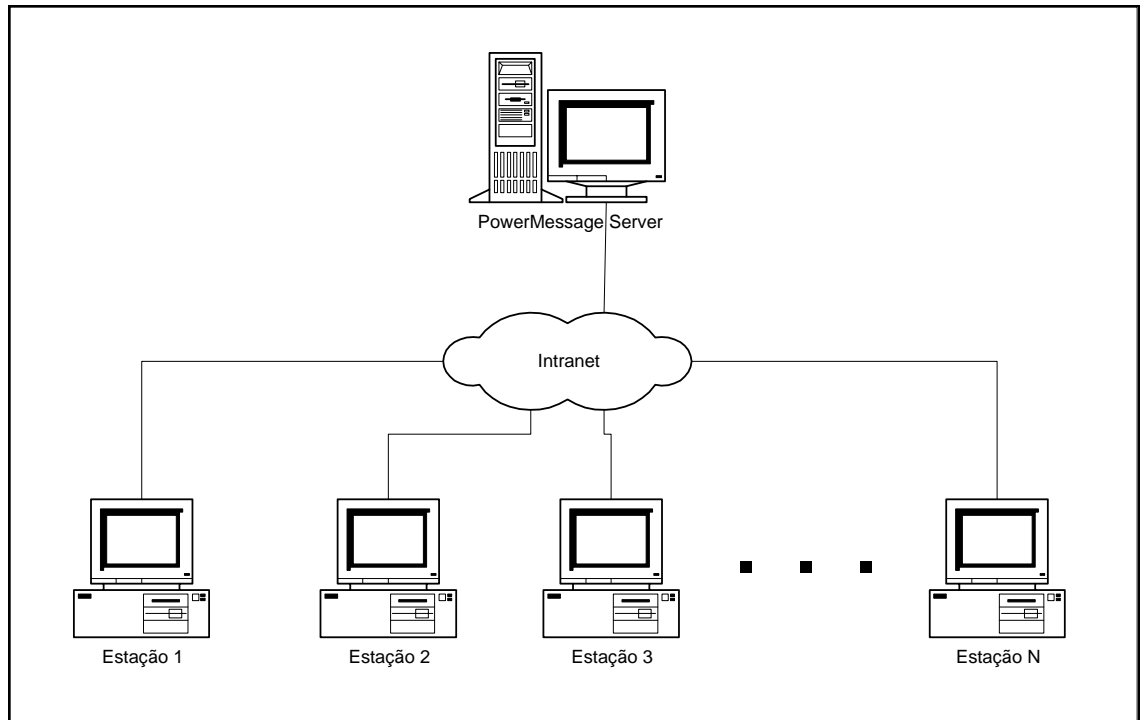
1.9. Viabilidade Técnica

Serão utilizadas as seguintes ferramentas de desenvolvimento:

- Platinum BpWin 2.5
- Platinum ErWin 3.5.2
- Rational Rose 98i
- Visio 97
- Microsoft Visual C++ 6.0, tanto para a parte do servidor, como para a parte do cliente.
- Biblioteca Dundas Ultimate Toolkit
- Biblioteca Dundas Ultimate TCP/IP
- Componentes ActiveX, utilizando a biblioteca Active Template Library (ATL).

2. Descrição da Arquitetura do Sistema

2.1. Arquitetura de Hardware do Sistema



2.2. Arquitetura de Software do Sistema

A arquitetura de software do sistema será baseada na arquitetura 3-tier baseada em componentes ActiveX.

2.3. Arquitetura de Segurança

A segurança do sistema será efetuada através de encriptação de senha e mensagem no cliente, e os dados trafegarão encriptados pela rede. O protocolo de mensagem que será criado para a comunicação não será encriptado, somente a senha e as mensagens.

3. Alocação de Funções aos Elementos do Sistema

3.1. Identificação dos Elementos de Hardware

3.1.1. Servidor

Pentium 166 MHZ ou superior

64 MB de Memória RAM

Disco Rígido de 4 GB

Monitor SVGA

3.1.2. Cliente

Padrão (Mínimo: Pentium 166 MHZ, 32 MB de Memória RAM)

3.2. Identificação dos Elementos de Software

3.2.1. Servidor

Software Básico: Microsoft Windows 95/98/NT

Suporte a TCP/IP

Bibliotecas de runtime do Visual C++ (MFC)

Bibliotecas de runtime do Data Access Objects (DAO)

3.2.2. Cliente

Software Básico: Microsoft Windows 95/98/NT

Suporte a TCP/IP

Bibliotecas de runtime do Visual C++ (MFC)

4. Modelagem do Sistema

4.1. Identificação dos Elementos Externos e dos dados de entrada e/ou Saída

Na figura abaixo temos o diagrama de use case do sistema PowerMessage, ressaltando os elementos externos ao sistema e o fluxo de dados entre eles.

AQUI VAI O DIAGRAMA DE USE CASES

Figura 1 - Diagrama de Contexto - Sistema PowerMessage

4.2. Detalhamento dos casos de uso

4.2.1. CASO DE USO 1

DIAGRAMA DO CASO DE USO DESCRITO, com os ATORES

Descrição Conforme vista em sala de aula

4.2.2. CASO DE USO 2

DIAGRAMA DO CASO DE USO DESCRITO, com os ATORES

Descrição Conforme vista em sala de aula

4.2.3. CASO DE USO N

DIAGRAMA DO CASO DE USO DESCRITO, com os ATORES

Descrição Conforme vista em sala de aula

4.3. Modelo Objeto

Abeixo, o modelo objeto do sistema incluindo as classes do sistema Cliente e do sistema Servidor, na notação UML.

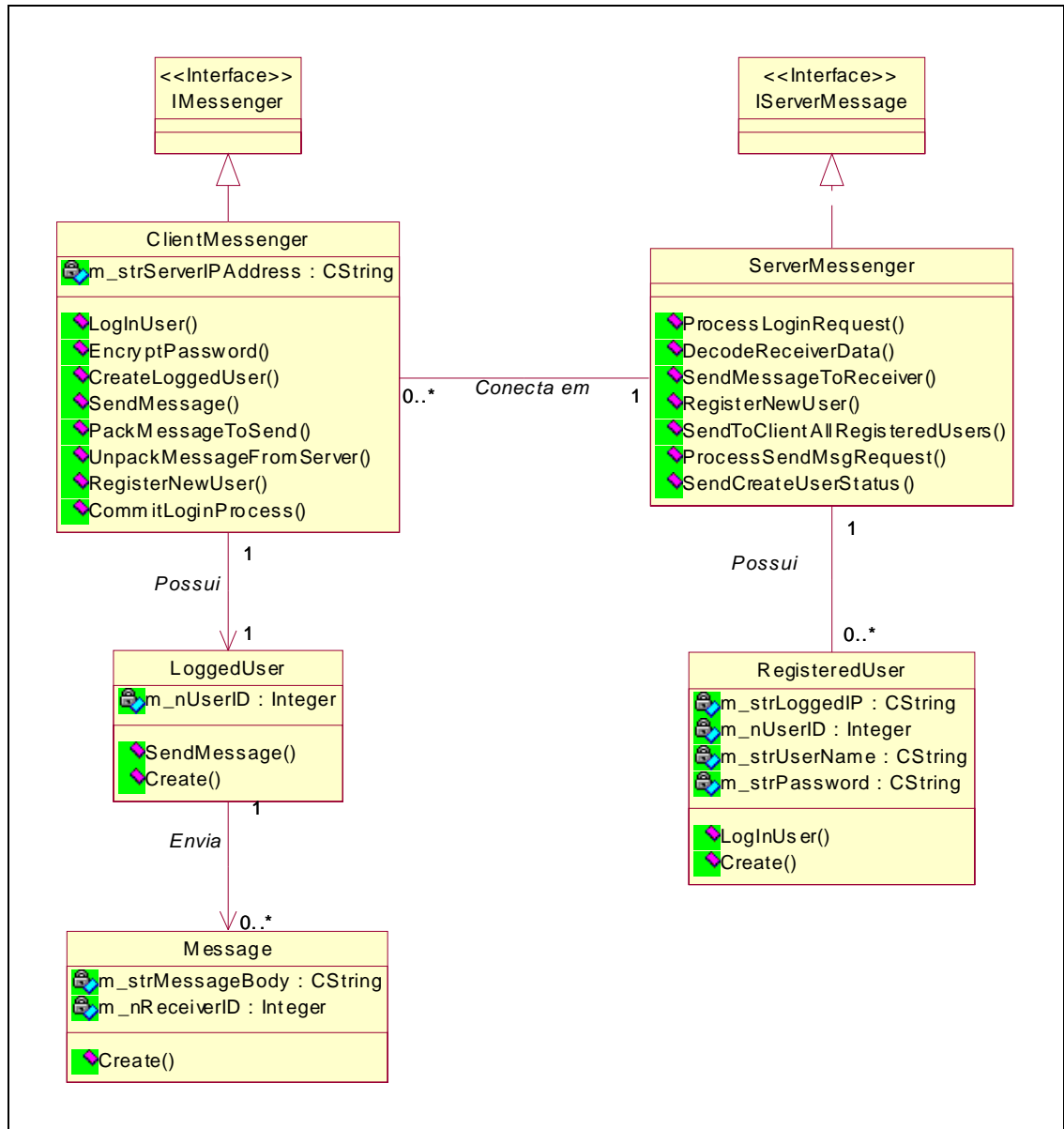


Figura 2 - Diagrama de Objetos

4.4. Diagramas de Colaboração

Neste item estão os diagramas de colaboração mais relevantes ao sistema.

4.4.1. Cenário 1 : Tentativa de Login Bem sucedida

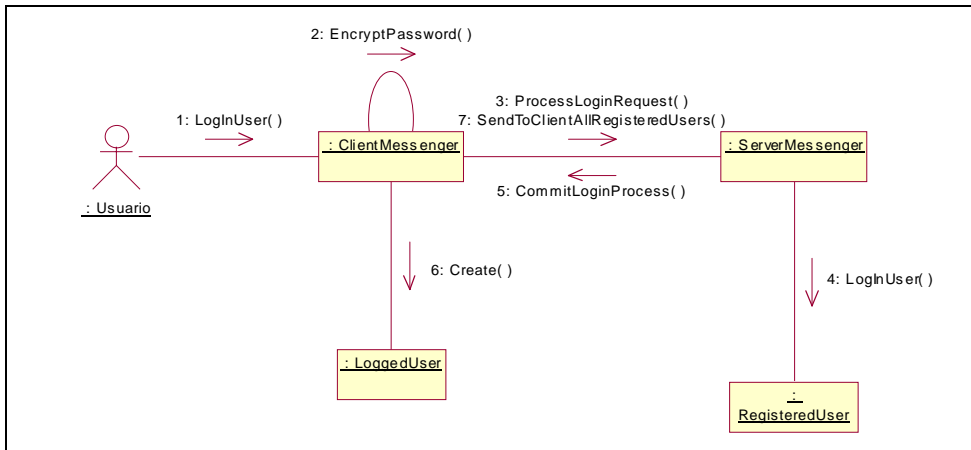


Figura 3 - Cenário 1

4.4.2. Cenário 2 : Envio de Mensagem Bem Sucedido

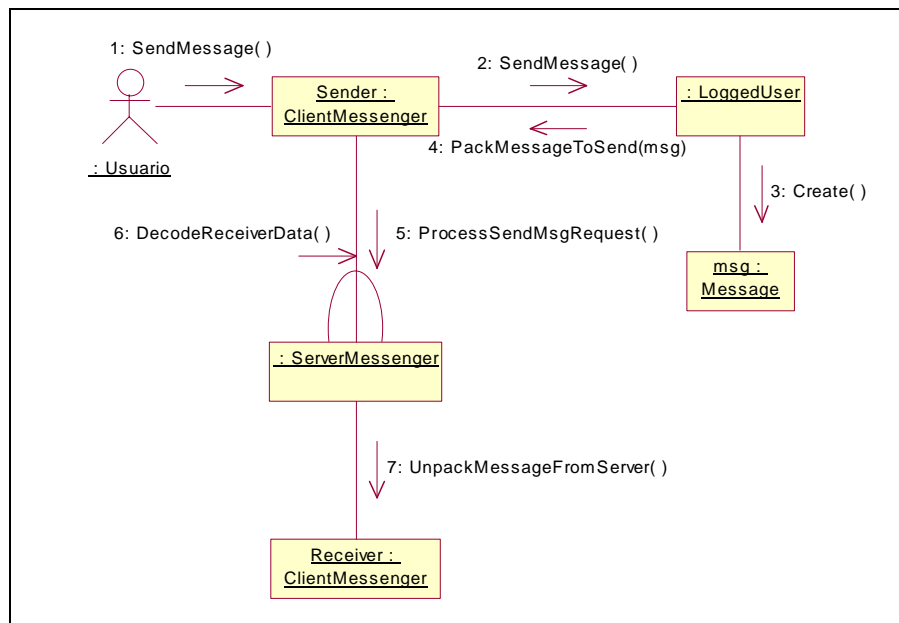


Figura 4 - Cenário 2

4.4.3. Cenário 3 : Cadastro de novo usuário bem sucedido

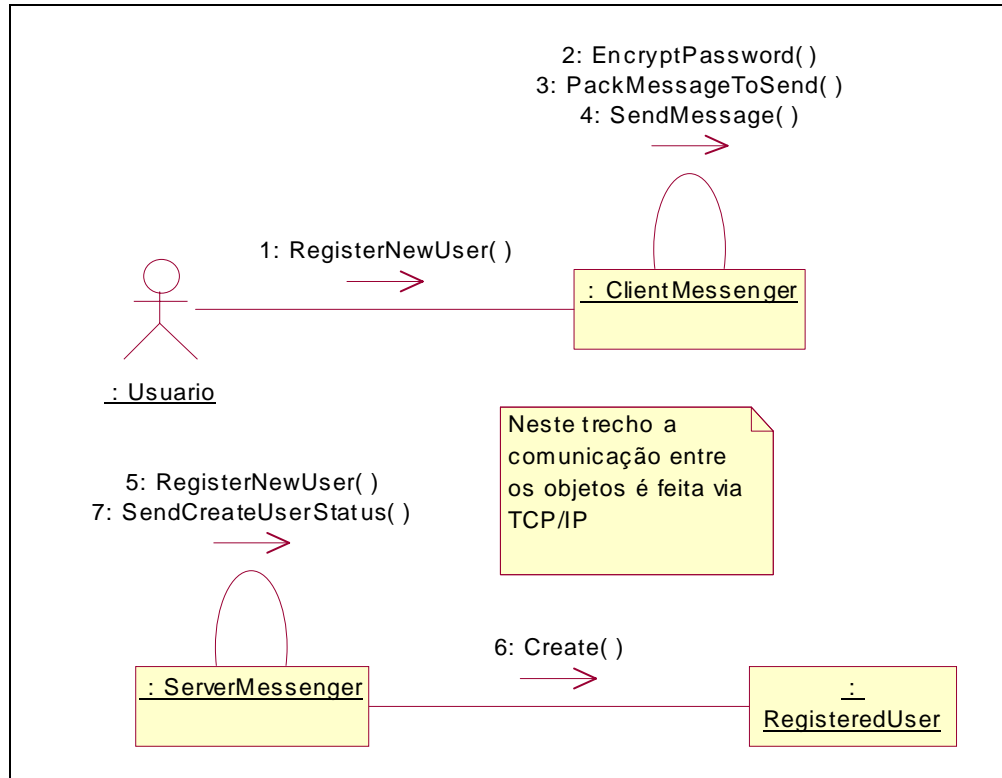
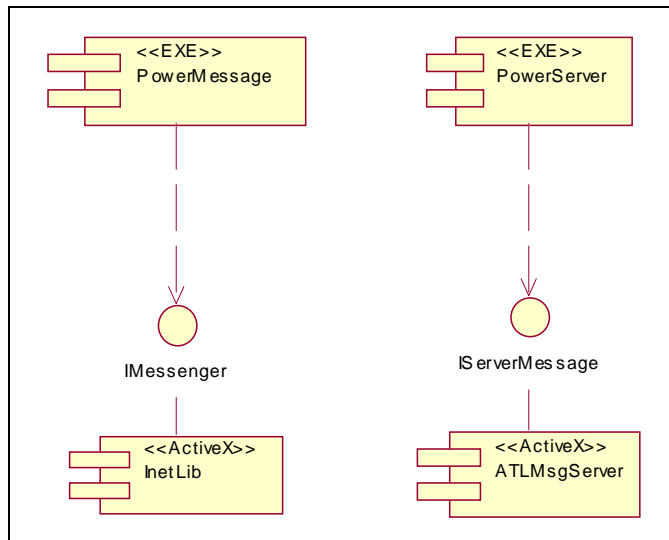


Figura 5 - Cenário 3

4.5. Identificação de Componentes

O sistema possuirá a estrutura de camadas, como foi citado no tópico de arquitetura. Basicamente serão 2 executáveis associados a um componente ActiveX cada um. O sistema cliente (PowerMessage.exe) estará atachado a um componente InetLib, que possuirá uma interface IMessenger. Já o sistema servidor (PowerServer.exe), estará atachado a um componente ATLMsgServer, que terá como interface pública IServerMessage. O sistema será implementado utilizando ATL/COM devido a familiaridade com o padrão, mas nada impede que no futuro ele seja feito em Java ou CORBA.



4.6. Descrição de Componentes

4.6.1. Componente InetLib

Interface IMessenger : Responsável pelo controle da parte de comunicação e envio das mensagens pela rede.

4.6.2. Componente ATLMsgServer

Interface IServerMessage : Responsável pelo controle da parte de comunicação e envio das mensagens pela rede. Sua única operação é:

HRESULT Start([out] long * pStartStatus),

que efetua a inicialização da interface e entra em um laço de escuta na porta 9000.

4.7. Descrição dos Dados

4.7.1. Modelo Entidade Relacionamento

O modelo entidade-relacionamento preliminar inclui a persistência dos objetos RegisteredUser e Message, podendo ser incrementados posteriormente.

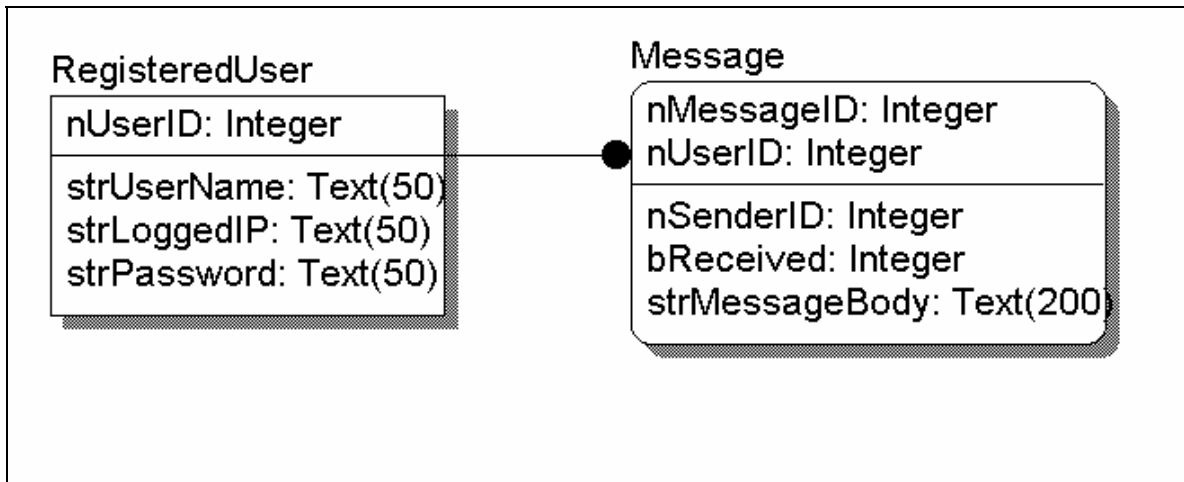


Figura 6 - Diagrama Entidade-Relacionamento do sistema PowerMessage

4.7.2. Descrição dos Comandos de Comunicação

Os comandos de comunicação serão enviados em um formato pseudo-XML. A sintaxe dos comandos estão especificados abaixo:

- o **Criação de Usuário**

```
<CreateUser>
  <UserName> User Name </UserName>
  <UserPassword> Password </UserPassword>
</CreateUser>
```

- o **Solicitação de Login**

```
<LoginRequest>
  <UserName> User Name </UserName>
  <UserPassword> Password </UserPassword>
</LoginRequest>
```

- o **Confirmação de Login**

```
<LoginResponse>
  <UserID> UserID </UserID>
  <StatusLogin> LoginOK | LoginDenied </StatusLogin>
</LoginResponse>
```

- o **Carga de Usuários**

```
<LoadUsers>
  <User>
    <UserID> UserID 1</UserID>
    <UserName> User Name 1</UserName>
    <Status> Online | Offline </Status>
  </User>
  <User>
    <UserID> UserID 2</UserID>
    <UserName> User Name 2</UserName>
    <Status> Online | Offline </Status>
  </User>
  .
  .
  .
  <User>
    <UserID> UserID n</UserID>
    <UserName> User Name n</UserName>
    <Status> Online | Offline </Status>
```

```
</User>  
</LoadUsers>
```

- **Envio de Mensagem Cliente → Servidor**

```
<ClientToServer>  
<ReceiverID> ReceiverID </ReceiverID>  
<MessageBody> MessageBody </MessageBody>  
</ClientToServer>
```

- **Envio de Mensagem Servidor → Cliente**

```
<ServerToClient>  
<SenderID> ReceiverID </SenderID>  
<MessageBody> MessageBody </MessageBody>  
</ServerToClient>
```

5.1. Módulo PowerMessage

5.1.1. Narrativa de Processamento

O módulo PowerMessage é um executável que implementará apenas a interface com o usuário.

5.1.2. Descrição da Interface

O módulo em questão não terá nenhuma interface pública, isto é, ele é apenas um aplicativo comum, não um OLE server.

5.1.3. Módulos e Componentes Utilizados

Ele se comunicará com o componente InetLib, que será o responsável pela comunicação e identificação dos elementos XML que chegam pela rede.

5.2. Módulo PowerServer

5.2.1. Narrativa de Processamento

O módulo PowerServer é o executável que instanciará o componente ATLMsgServer, e permitirá executar as rotinas de administração.

5.2.2. Descrição da Interface

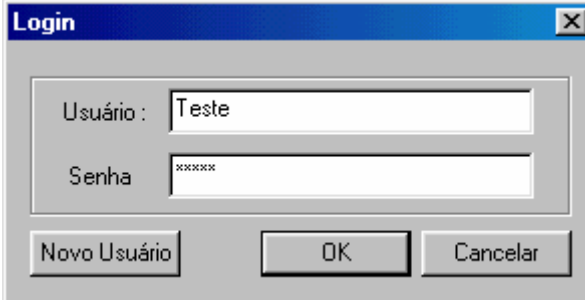
O módulo em questão não terá nenhuma interface pública, isto é, ele é apenas um aplicativo comum, não um OLE server.

5.2.3. Módulos e Componentes Utilizados

Ele se comunicará com o componente ATLMsgServer, que será o responsável pela comunicação, identificação dos elementos XML e processamento das solicitações que chegam pela rede.

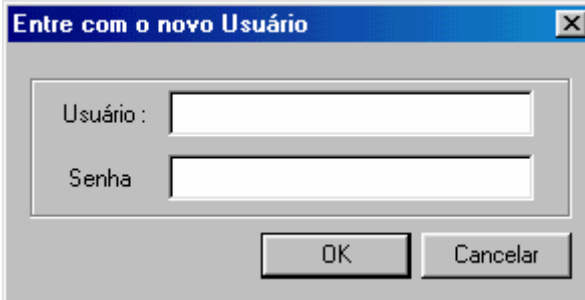
6. Navegação do Sistema - Diagrama de Telas

6.1. Tela de Login



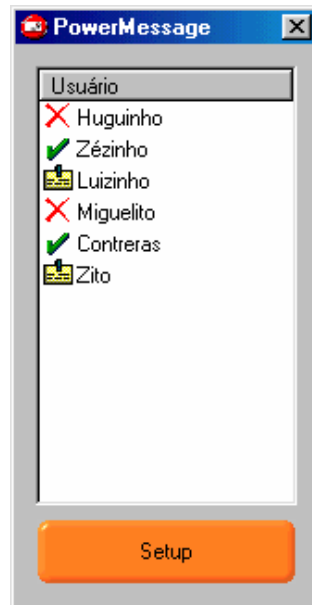
The screenshot shows a dialog box titled "Login" with a close button (X) in the top right corner. It contains two input fields: "Usuário:" with the text "Teste" and "Senha:" with six asterisks. Below the fields are three buttons: "Novo Usuário", "OK", and "Cancelar".

6.2. Tela de Cadastramento de Novo Usuário



The screenshot shows a dialog box titled "Entre com o novo Usuário" with a close button (X) in the top right corner. It contains two empty input fields: "Usuário:" and "Senha:". Below the fields are two buttons: "OK" and "Cancelar".

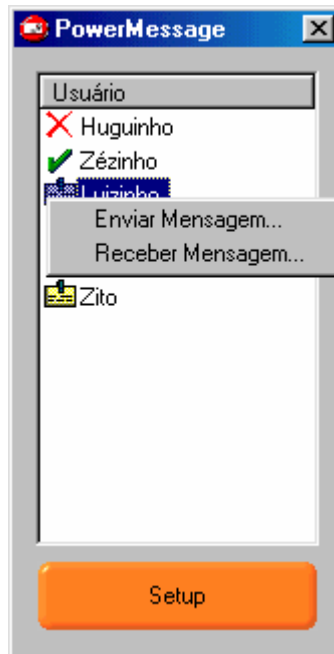
6.3. Tela Principal do Sistema



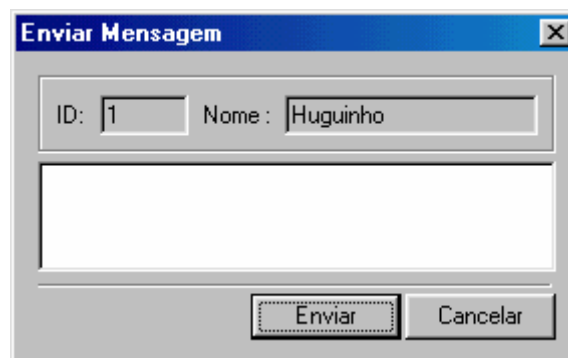
6.4. Tela de Setup do Sistema



6.5. Menu de Contexto para Interação com Usuários



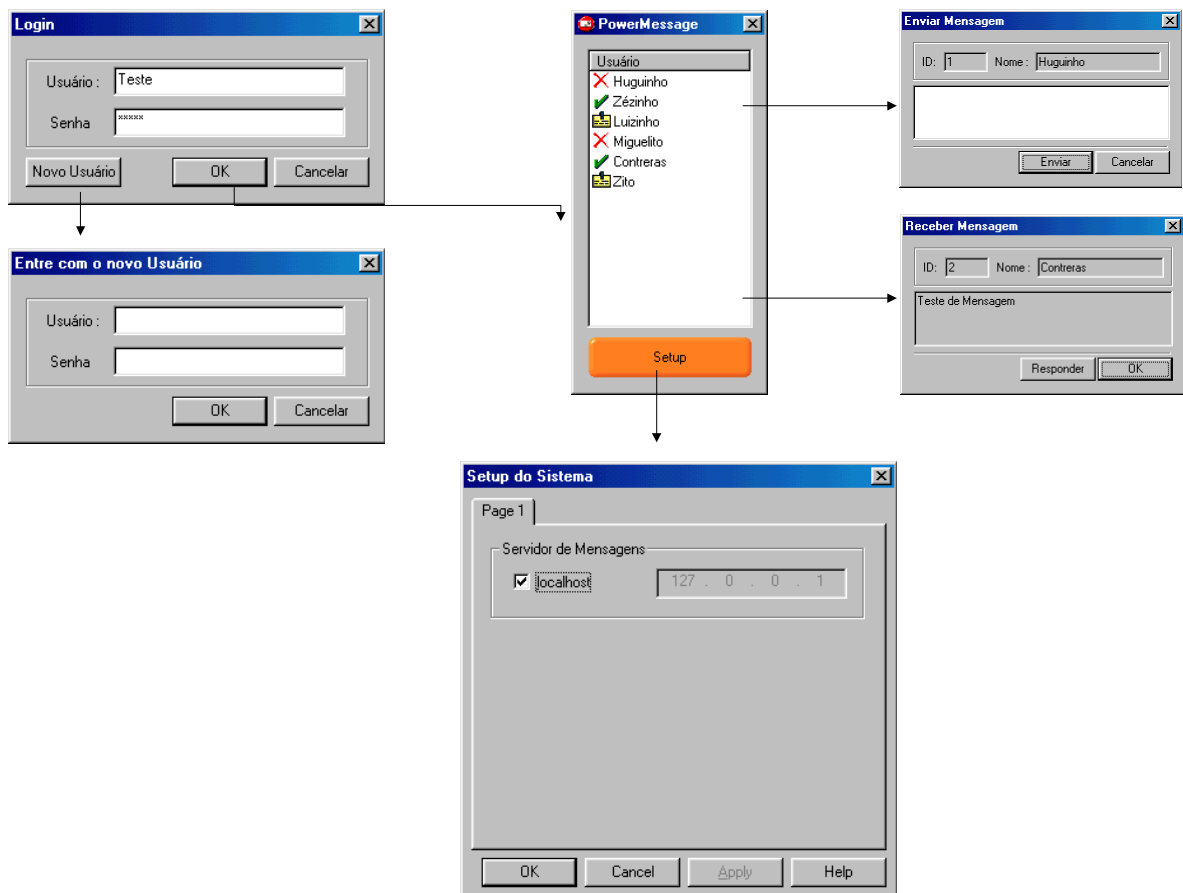
6.6. Tela de Envio de Mensagem



6.7. Tela de Recebimento de Mensagem



6.8. Diagrama de Navegação



7.1. Configuração

Esta categoria contém os testes referentes à configuração do ambiente.

7.1.1. Teste de configuração do endereço do servidor com endereço correto.

- Verificar a conexão do cliente com o servidor, sendo informado o endereço IP correto do servidor.

Resultado Esperado : O sistema deve executar corretamente e apresentar a tela de envio e recepção de mensagem.

7.1.2. Teste de configuração do endereço do servidor com endereço incorreto.

- Verificar a conexão do cliente com o servidor, sendo informado o endereço IP incorreto do servidor.

Resultado Esperado : O sistema deve apresentar uma mensagem de falha na conexão com o servidor após a tentativa de login.

7.2. Domínio

Esta categoria contém os testes referentes ao domínio de um campo ou atributo.

7.2.1. Teste de Tamanho do Atributo de Nome de Usuário (Cliente)

- Verificar se o campo Usuário na tela de Login e de Novo Usuário aceita 50 posições.

Resultado Esperado : O campo não deve aceitar mais caracteres após o 50º caractere ser digitado.

7.2.2. Teste de Tamanho do Atributo de Senha (Cliente)

- Verificar se o campo Senha na tela de Login e de Novo Usuário aceita 50 posições.

Resultado Esperado : O campo não deve aceitar mais caracteres após o 50º caractere ser digitado.

7.2.3. Teste de Validade do Atributo de Nome de Usuário (Cliente)

- Verificar se a tela de cadastramento de novo usuário é possível o cadastramento de um usuário já existente.

Resultado Esperado : Após a confirmação de inclusão de usuário, o sistema deve retornar uma mensagem de erro indicando a existência de que já existe um usuário com o mesmo nome.

7.2.4. Teste de Tamanho do Atributo de Mensagem (Cliente)

- Verificar se o campo Mensagem da Tela de Enviar Mensagem aceita 200 posições

Resultado Esperado : O campo não deve aceitar mais caracteres após o 200º caractere ser digitado.

7.3. Limites

Esta categoria contém os testes referentes ao *stress* do sistema. Por exemplo, abrir o aplicativo várias vezes.

7.3.1. Teste de Múltiplas Instâncias do Programa (Cliente e Servidor)

- Verificar o comportamento do sistema com relação à múltiplas instâncias.

Resultado Esperado : Na execução de uma segunda instância do aplicativo, o sistema operacional deve dar o foco para a aplicação já previamente sendo executada, impedindo a execução de outra.

7.4. Desempenho

Esta categoria contém os testes que se propõe a verificar se o desempenho do sistema está de acordo com o esperado.

7.4.1. Teste de Inclusão de Usuário

- Verificar o tempo para a inclusão de um usuário no sistema

Resultado Esperado : $T < 15s$.

7.4.2. Teste de Login

- Verificar o tempo para efetuar o login no sistema e o sistema se apresentar pronto par uso.

Resultado Esperado : $T < 5s$.

7.4.3. Teste de Envio de Mensagem

- Verificar o tempo para o envio de mensagem entre o envio e a recepção.

Resultado Esperado : $T < 5s$.

7.5. Volume

Esta categoria contém os testes referentes à base de dados..

7.5.1. Teste de Inclusão de usuário com a base sem dados

- Incluir um usuário como primeiro usuário do sistema.

Resultado Esperado : Alta performance do sistema ($T < 5s$).

7.5.2. Teste de Inclusão de usuário com a base com grande volume de dados

- Incluir um usuário com muitos usuários cadastrados.

Resultado Esperado : Alta performance do sistema ($T < 5s$).

7.5.3. Teste de Login de usuário com a base sem dados

- Logar um usuário como primeiro usuário do sistema..

Resultado Esperado : Alta performance do sistema ($T < 5s$).

7.5.4. Teste de Login de usuário com a base com grande volume de dados

- Logar um usuário com muitos usuários cadastrados.

Resultado Esperado : Alta performance do sistema ($T < 5s$).

7.5.5. Teste de Envio de Mensagem com nenhuma mensagem cadastrada

- Enviar uma mensagem sem mensagens cadastradas no sistema.

Resultado Esperado : Alta performance do sistema ($T < 5s$).

7.5.6. Teste de Envio de Mensagem com grande volume de mensagens cadastradas.

- Enviar uma mensagem com muitas mensagens cadastradas no sistema.

Resultado Esperado : Alta performance do sistema ($T < 5s$).

8. Obtenção do Software

8.1. Particionamento do Sistema

Para obter o software, o sistema foi particionado de acordo com os processos definidos pelo diagrama de identificação dos subsistemas. A figura abaixo apresenta as partições e sub-partições do sistema.

**AQUI VAI O DIAGRAMA DE CASOS DE USO, DEVIDAMENTE PARTICIONADO,
CONFORME VISTO EM SALA DE AULA**

Figura 7 - Particionamento do Sistema

8.2. Roteirização das Partições

Como estratégia de implementação as partições serão desenvolvidas sequencialmente, com prioridades conforme apresentado no particionamento do sistema (Figura 14).

8.3. Sumário Gerencial do Plano de Obtenção de Software

Partições/Atividades
Modelagem do Sistema
Partição 1: Cadastramento dos Usuários
Partição 2: Login
Partição 3: Enviar Mensagem
Partição 4: Processa Requisição de Mensagem
Partição 5: Receber Mensagem
Homologação
Implantação
Operação

9. Referências Bibliográficas

Nesta análise bibliográfica, além dos livros e artigos que estão sendo usados efetivamente para estudo do sistema, também estão listados aqueles que foram utilizados como base para escolha da tecnologia que será utilizada para implementação.

Wutka, Mark, *Java : Técnicas Profissionais*, Berkeley, 1997

Pressman, Roger, *Engenharia de Software*, Makron Books, 1995

Anúnciação, Heverton, *Linux para Redes Brasileiras*, Érica, 1997

Soares, Luiz Fernando et alii, *Redes de Computadores : Das LANS, MANS e WANS às Redes ATM*, Editora Campus, 1995

Gregory , Kate, *Special Edition Using Visual C++ 6*, QUE, 1998

Parker, Tim, *Teach Yourself TCP/IP in 14 Days*, Sams Publishing, 1996

Rosenberger, Jeremy, *Teach Yourself CORBA in 14 Days*, Sams Publishing, 1998

Garret, David et al, *Intranets Unleashed*, Sams Publishing, 1998

Schussel, George, *Client/Server: Past, Present and Future*, Artigo disponível em <http://news.dci.com/geos/dbsejava.htm>

Darleen, Sadosky et al, *Three Tier Software Architectures*, Artigo disponível em http://www.sei.cmu.edu/str/descriptions/threetier_body.html

Larman, Craig, *Applying UML and Patterns*, Prentice Hall, 1997

Rumbaugh, James et al, *Modelagem e Projetos Baseados em Objetos*, Editora Campus, 1997

Three Tier Application Development, MSDN Library, 1998

Ken Bergmann, *Client Server Solutions, Implementing the layered paradigm*, MSDN Library, 1996

Sundblad, Per et al, *Scalable design patterns, MSDN Library*, 1997

Kanter, Joel P., *Understanding Thin Client/Server Computing*, Microsoft Press, 1998

Chang, Kai H. et al, *Testing object-oriented programs : from formal specification to test scenario generation*, The Journal of Systems and Software, 1998