

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO

**“Desenvolvimento de Componentes para  
Sistemas Cooperativos”**

ORIENTADOR: Prof. Dr. Wanderley Lopes de Souza  
ALUNA: Helen de Freitas Santos

**São Carlos  
Julho/2001**

HELEN DE FREITAS SANTOS

**Desenvolvimento de  
Componentes  
para  
Sistemas Cooperativos**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

**São Carlos**

**Julho/2001**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

S237dc

Santos, Helen de Freitas.

Desenvolvimento de componentes para sistemas cooperativos / Helen de Freitas Santos. -- São Carlos : UFSCar, 2004.  
282 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2001.

1. Engenharia de software. 2. CSCW. 3. Group Ware. 4. Metodologia UML. 5. Componentes de software. 6. Conferência multimídia. I. Título.

CDD: 005.1(20<sup>a</sup>)

# Agradecimentos

À Deus, por ser a origem da nossa existência.

Ao Professor Dr. Wanderley Lopes de Souza, pela orientação e incentivo durante o desenvolvimento deste trabalho.

Ao colega Cléver Ricardo Guareis de Farias, pela colaboração dispensada.

Aos amigos e colegas do PPG-CC, pelos momentos de prazer e dificuldades compartilhados.

Aos funcionários e professores do PPG-CC, pela atenção dedicada.

A todos os meus familiares, sem os quais este trabalho não teria se concretizado.

# Resumo

Trabalho Cooperativo Suportado por Computador (CSCW - Computer Supported Cooperative Work) é um campo multidisciplinar que busca o estudo de formas adequadas para promover a cooperação entre grupos de pessoas envolvidas na realização de uma tarefa comum, utilizando-se de um suporte computacional para facilitar essa cooperação. Além de estudar as diversas maneiras da computação facilitar as atividades cooperativas, o projeto, a implementação e a implantação do suporte computacional para auxiliar as várias formas de cooperação também são objeto de estudo de CSCW.

O suporte à atividade cooperativa é de grande importância uma vez que, para o cumprimento de um objetivo comum, as atividades envolvendo um grupo de pessoas são em maior quantidade que as atividades individuais.

Neste sentido, este trabalho apresenta um conjunto de componentes cooperativos identificados a partir da utilização de uma metodologia para desenvolvimento de software baseado em componentes.

# Abstract

Computer Supported Cooperative Work (CSCW) is a multidisciplinary field that looks for appropriate studies to promote cooperation among groups of people involved in a common task to make use of a computational support to facilitate this cooperation. Besides studying several computation ways to facilitate the cooperative activities, the design, software development and implementation of the support computational are also object of study of CSCW.

The cooperative activity support has a great importance because these activities involving a group of people are larger amount than individual ones to the complement of a common purpose.

Within this sense, this research introduces a collection of identified cooperative components through the utilization of a component-based software development methodology.

# Sumário

|                           |     |
|---------------------------|-----|
| Agradecimentos.....       | iii |
| Resumo.....               | iv  |
| Abstract.....             | v   |
| Sumário.....              | vi  |
| Lista de Figuras.....     | ix  |
| Lista de Tabelas.....     | xi  |
| Abreviações e Siglas..... | xii |

## Capítulo I

|                                  |           |
|----------------------------------|-----------|
| <b>Introdução.....</b>           | <b>01</b> |
| 1.1. Apresentação.....           | 01        |
| 1.2. Motivação.....              | 03        |
| 1.3. Objetivo e Escopo.....      | 04        |
| 1.4. Estrutura do Documento..... | 04        |

## Capítulo II

|                                      |           |
|--------------------------------------|-----------|
| <b>Groupware.....</b>                | <b>06</b> |
| 2.1. Definição de Groupware.....     | 06        |
| 2.2. Classificação de Groupware..... | 08        |
| 2.3. Toolkits Cooperativos.....      | 12        |

## Capítulo III

|   |           |
|---|-----------|
| <b>Desenvolvimento de Software baseado em Componentes.....</b>                | <b>20</b> |
| 3.1. Componentes de Software.....   | 20        |
| 3.2. Unified Modeling Language (UML – Linguagem de Modelagem Unificada).....  | 22        |
| 3.3. Metodologia para o Desenvolvimento de Software Baseado em Componentes... | 24        |
| 3.4. Modelo Conceitual para o Desenvolvimento de Sistemas CSCW.....           | 29        |

|   |    |
|---|----|
| 3.5. Plataforma de Desenvolvimento..... | 32 |
| 3.5.1. Enterprise Java Beans.....       | 32 |
| 3.5.2. Java 2 Enterprise Edition.....   | 33 |
| 3.5.3. Java Media Framework.....        | 34 |

## Capítulo IV

|  |           |
|--|-----------|
| <b>Aplicação da Metodologia – Sistema de Conferência Multimídia.....</b> | <b>37</b> |
| 4.1. Descrição Informal da Aplicação a ser Modelada.....                 | 37        |
| 4.2. Modelagem – Nível de Empresa.....                                   | 44        |
| 4.3. Modelagem – Nível de Sistema.....                                   | 46        |
| 4.4. Modelagem – Nível de Componente.....                                | 48        |
| 4.5. Modelagem – Nível de Objeto.....                                    | 56        |

## Capítulo V

|  |           |
|--|-----------|
| <b>Toolkit para Aplicações Cooperativas (TAC).....</b> | <b>59</b> |
| 5.1. Origem do TAC.....                                | 59        |
| 5.2. Arquitetura do TAC.....                           | 61        |
| 5.3. Componentes Cliente.....                          | 65        |
| 5.4. Componentes Servidor.....                         | 66        |
| 5.4.1. Gerente de Aplicação Servidor (GApS).....       | 66        |
| 5.4.2. Gerente de Acesso Servidor (GAcS).....          | 66        |
| 5.4.3. Gerente de Sessão Servidor (GSS).....           | 68        |
| 5.4.4. Gerente de Participação Servidor (GPS).....     | 70        |
| a) Gerente de Inscrição Servidor (GIS).....            | 70        |
| b) Gerente de Convite Servidor (GCS).....              | 72        |
| c) Gerente de Associação Servidor (GAsS).....          | 74        |
| d) Gerente de Correio Assíncrono Servidor (GCAS).....  | 75        |
| 5.4.5. Gerente de Cooperação Servidor (GCpS).....      | 76        |
| a) Gerente de Comunicação Servidor (GCmS).....         | 77        |
| b) Gerente de Mídia Servidor (GMS).....                | 78        |
| c) Gerente de Correio Síncrono Servidor (GCSS).....    | 79        |



|  |    |
|--|----|
| d) Gerente de Recepção Servidor (GRS).....                       | 80 |
| e) Gerente de Transmissão Servidor (GTS).....                    | 81 |
| 5.4.6. Gerente de Votação Servidor (GVtS).....                   | 82 |
| 5.4.7. Gerente de Anotação Servidor (GAnS).....                  | 84 |
| 5.4.8. Gerente de Transferência de Arquivos Servidor (GTAS)..... | 85 |
| 5.4.9. Gerente de Gravação de Conferência Servidor (GGCS).....   | 85 |
| 5.4.10. Gerente de Visualização Servidor (GVS).....              | 86 |
| 5.5. Notificação de Eventos.....                                 | 87 |

## **Capítulo VI**

|                       |           |
|-----------------------|-----------|
| <b>Conclusão.....</b> | <b>89</b> |
|-----------------------|-----------|

|  |           |
|--|-----------|
| <b>Referências Bibliográficas.....</b> | <b>95</b> |
|--|-----------|

|   |           |
|---|-----------|
| <b>Apêndice 1</b>                                   | <b>99</b> |
| Sistema de Conferência Multimídia – Requisitos..... | 99        |

|   |            |
|---|------------|
| <b>Apêndice 2</b>   | <b>106</b> |
| Sistema de Conferência Multimídia – Nível de Empresa..... | 106        |

|   |            |
|---|------------|
| <b>Apêndice 3</b>   | <b>128</b> |
| Sistema de Conferência Multimídia – Nível de Sistema..... | 128        |

|  |            |
|--|------------|
| <b>Apêndice 4</b>  | <b>180</b> |
| Sistema de Conferência Multimídia – Nível de Componente..... | 180        |

|  |            |
|--|------------|
| <b>Apêndice 5</b>  | <b>251</b> |
| Sistema de Conferência Multimídia – Nível de Objeto..... | 251        |

# Lista de Figuras

|  |    |
|--|----|
| Figura 2.1. Duas Dimensões de Alcance do Groupware.....  | 08 |
| Figura 2.2. Classificação de Groupware de acordo com Tempo e Espaço.....   | 08 |
| Figura 3.1. Níveis de Abstração.....   | 25 |
| Figura 3.2. O uso das visões em um processo de desenvolvimento.....  | 26 |
| Figura 3.3. Modelo CSCW Genérico.....  | 30 |
| Figura 3.4. Arquitetura da Plataforma J2EE.....  | 34 |
| Figura 3.5. Arquitetura JMF de alto nível.....   | 35 |
| Figura 4.1. Nível de Empresa – Diagrama de Objeto – Atividade Controlar<br>Participante por Sessão.....                | 45 |
| Figura 4.2. Nível de Empresa – Diagrama de Atividades – Atividade Controlar<br>Participante por Sessão.....            | 46 |
| Figura 4.3. Nível de Sistema – Visão Estrutural – Diagrama de Casos de Uso –<br>Controlar Participante por Sessão..... | 47 |
| Figura 4.4. Nível de Sistema – Visão Interacional – Diagrama de Seqüência.....   | 48 |
| Figura 4.5. Nível de Componente – Visão Estrutural – Diagrama de Pacotes –<br>Cliente.....                             | 51 |
| Figura 4.6. Nível de Componente – Visão Estrutural – Diagrama de Pacotes –<br>Servidor.....                            | 52 |
| Figura 4.7. Nível de Componente – Visão Estrutural – Interface – Componente<br>Gerente de Acesso Cliente.....          | 53 |
| Figura 4.8. Nível de Componente – Visão Interacional – Diagrama de Seqüência –<br>Cliente .....                        | 54 |
| Figura 4.9. Nível de Componente – Visão Interacional – Diagrama de Seqüência –<br>Servidor.....                        | 55 |
| Figura 4.10. Nível de Componente – Visão Comportamental – Diagrama de<br>Atividade – Servidor.....                     | 56 |
| Figura 4.11. Nível de Objeto – Visão Estrutural – Diagrama de Caso de Uso –<br>Gerente de Acesso Servidor.....         | 57 |
| Figura 4.12. Nível de Objeto – Visão Estrutural – Diagrama de Classe – Gerente de<br>Acesso Servidor.....              | 57 |

|   |    |
|---|----|
| Figura 4.13. Nível de Objeto – Visão Interacional – Diagrama de Seqüência   |    |
| Servidor – Requisição de Registro.....                                      | 58 |
| Figura 4.14. Nível de Objeto – Visão Comportamental – Diagrama de Atividade |    |
| Servidor – Requisição de Validar Username.....                              | 58 |
| Figura 5.1. Estrutura do TAC Cliente.....                                   | 62 |
| Figura 5.2. Estrutura do TAC Servidor.....                                  | 63 |
| Figura 5.3. Estrutura de Banco de Dados.....                                | 64 |
| Figura 5.4. Classes e Métodos do Componente GAcS.....                       | 67 |
| Figura 5.5. Classes e Métodos do Componente GSS.....                        | 70 |
| Figura 5.6. Classes e Métodos do Componente GIS.....                        | 72 |
| Figura 5.7. Classes e Métodos do Componente GCS.....                        | 74 |
| Figura 5.8. Classes e Métodos do Componente GAsS.....                       | 75 |
| Figura 5.9. Classes e Métodos do Componente GCAS.....                       | 76 |
| Figura 5.10. Classes e Métodos do Componente GCmS.....                      | 78 |
| Figura 5.11. Classes e Métodos do Componente GMS.....                       | 79 |
| Figura 5.12. Classes e Métodos do Componente GCSS.....                      | 80 |
| Figura 5.13 Classes e Métodos do Componente GRS.....                        | 81 |
| Figura 5.14. Classes e Métodos do Componente GTS.....                       | 82 |
| Figura 5.15. Classes e Métodos do Componente GVtS.....                      | 83 |
| Figura 5.16. Classes e Métodos do Componente GAnS.....                      | 84 |
| Figura 5.17. Classes e Métodos do Componente GTAS.....                      | 85 |
| Figura 5.18. Classes e Métodos do Componente GGCS.....                      | 86 |
| Figura 5.19. Classes e Métodos do Componente GVS.....                       | 87 |

## Lista de Tabelas

|  |    |
|--|----|
| Tabela 3.1. Técnicas para Modelagem.....   | 28 |
| Tabela 4.1. Atividades Oferecidas pelo SCM sob a Funcionalidade de<br>Participação.....      | 40 |
| Tabela 4.2. Serviços Oferecidos pelo SCM sob a Funcionalidade de<br>Cooperação.....          | 41 |
| Tabela 4.3. Funcionalidades/Atividades Oferecidos pelo SCM.....                              | 42 |
| Tabela 4.4. Componentes Identificados para o Sistema de Conferência Multimídia<br>(SCM)..... | 50 |
| Tabela 5.1. Mensagens a serem enviadas pelo componente GCAS                                  | 76 |

# Abreviações e Siglas

|       |  |
|-------|--|
| AIFF  | Audio Interchange File Format                |
| ALV   | Abstraction-Link-View                        |
| API   | Application Programming Interface            |
| AVI   | Audio Video Interleaved                      |
| CLOS  | Common Lisp Object System                    |
| CORBA | Common Object Request Broker Architecture    |
| CSCW  | Computer Supported Cooperative Work          |
| EJB   | Enterprise Java Beans                        |
| GA    | Gerente de Aplicação                         |
| GAcC  | Gerente de Acesso Cliente                    |
| GAcS  | Gerente de Acesso Servidor                   |
| GAnS  | Gerente de Anotação Servidor                 |
| GAaS  | Gerente de Associação Servidor               |
| GApC  | Gerente de Aplicação Cliente                 |
| GApS  | Gerente de Aplicação Servidor                |
| GCAS  | Gerente de Correio Assíncrono Servidor       |
| GCmS  | Gerente de Comunicação Servidor              |
| GCS   | Gerente de Convite Servidor                  |
| GCSS  | Gerente de Correio Síncrono Servidor         |
| GCpC  | Gerente de Cooperação Cliente                |
| GCpS  | Gerente de Cooperação Servidor               |
| GGCS  | Gerente de Gravação de Conferência Servidor  |
| GIS   | Gerente de Inscrição Servidor                |
| GMS   | Gerente de Mídia Servidor                    |
| GPC   | Gerente de Participação Cliente              |
| GPS   | Gerente de Participação Servidor             |
| GRS   | Gerente de Recepção Servidor                 |
| GSC   | Gerente de Sessão Cliente                    |
| GSS   | Gerente de Sessão Servidor                   |
| GTAS  | Gerente de Transferência de Arquivo Servidor |
| GTS   | Gerente de Transmissão Servidor              |

|       |  |
|-------|--|
| GVC   | Gerente de Visualização Cliente        |
| GVS   | Gerente de Visualização Servidor       |
| GVtS  | Gerente de Votação Servidor            |
| HTML  | Hyper Text Markup Language             |
| http  | HyperText Transfer Protocol            |
| IIOP  | Internet Inter-Orb Protocol            |
| J2EE  | Java 2 Enterprise Edition              |
| JMF   | Java Media Framework                   |
| JNDI  | Java Naming and Directory Interface    |
| JPG   | Joint Photographic Experts Groups      |
| MIDI  | Musical Instrument Digital Interface   |
| MIT   | Massachussets Institute of Technology  |
| MPEG  | Motion Picture Expert Group            |
| NSTP  | Notification Service Transfer Protocol |
| OMG   | Object Management Group                |
| OMT   | Object Modeling Technique              |
| OOSE  | Object-Oriented Software Engineering   |
| ORB   | Object Request Broker                  |
| OVAL  | Objects, Views, Agents and Links       |
| PCM   | Pulse Code Modulation                  |
| RMF   | Rich Music Format                      |
| RMI   | Remote Method Invocation               |
| RPC   | Remote Procedure Call                  |
| RTCAL | Real Time Calendar                     |
| SCM   | Sistema de Conferência Multimídia      |
| SSL   | Secure Sockets Layer                   |
| TAC   | Toolkit para Aplicações Cooperativas   |
| UML   | Unified Modeling Language              |
| WAV   | Wave                                   |
| XML   | Extensible Markup Language             |

---

## Capítulo I

### **Introdução**

#### **1.1. Apresentação**

Na maioria dos softwares existentes atualmente a interação entre o computador e as pessoas é quase a mesma que há algumas décadas. Sob a perspectiva do ambiente de trabalho, este não é formado por pessoas trabalhando isoladamente. Há uma série de atividades, como a comunicação entre os departamentos, a circulação dos documentos e a ordem de execução das tarefas, que implica numa interação entre as pessoas envolvidas na realização do trabalho em grupo. Além disso, as organizações expandiram seus mercados e novas formas de comunicação, como o correio eletrônico e a sala de reunião eletrônica, passaram a ser utilizadas. Numa tentativa de unir a tecnologia ao trabalho em grupo, teve início um novo campo multidisciplinar chamado Trabalho Cooperativo Suportado por Computador (CSCW – Computer Supported Cooperative Work). A área de CSCW surgiu em 1984 num workshop realizado no MIT (Massachusetts Institute of Technology) e os responsáveis pelo termo foram Irene Grief e Paul Cashman [3].

A comunicação entre os indivíduos que trabalham em grupo envolve a utilização de diversas ferramentas tecnológicas para realização de suas tarefas, tais como computador, telefone, fax, correio eletrônico, câmera de vídeo, sendo que através da integração das tecnologias de telecomunicação e de computação é que será possível tornar as atividades cooperativas amplamente auxiliadas por computador.

CSCW é uma área de pesquisa voltada para o estudo de formas adequadas para promover a cooperação entre grupos de pessoas envolvidas na realização de uma tarefa comum, utilizando-se de um suporte computacional para facilitar essa cooperação [5]. CSCW utiliza ferramentas baseadas em computador para aumentar e facilitar a atuação do trabalho cooperativo, o qual geralmente está distribuído no tempo e espaço. De acordo com a opinião e colaboração de muitos especialistas, incluindo cientistas sociais e de computador, CSCW se preocupa em como os grupos trabalham e procura descobrir como a tecnologia, principalmente o computador, pode ajudá-los na realização de suas tarefas. O desenvolvimento de sistemas cooperativos envolve, portanto, tanto questões

---

de ordem social, como comunicação, colaboração e coordenação decorrentes do relacionamento entre os indivíduos, quanto de ordem computacional, como a decisão sobre a utilização de uma arquitetura centralizada ou distribuída, o controle de concorrência e sincronização para garantia da consistência da informação e o gerenciamento do acesso à informação.

Além de estudar as diversas maneiras da computação facilitar as atividades cooperativas, o projeto, a implementação e a implantação do suporte computacional para auxiliar as várias formas de cooperação também são objeto de estudo de CSCW. Os produtos que provêm um suporte computacional para um grupo de pessoas realizar uma mesma tarefa são chamados de groupware [5].

O desenvolvimento de groupware não é uma tarefa simples. Além de problemas sociais característicos das relações entre os indivíduos, esta tarefa é dificultada por problemas pertinentes tanto a sistemas distribuídos quanto ao trabalho cooperativo de maneira geral. Quanto aos problemas relacionados à distribuição, os groupwares devem oferecer serviços que garantam a confiabilidade da informação e devem possibilitar sua utilização em uma multiplicidade de plataformas operacionais. Quanto aos problemas relacionados à cooperação entre grupos de indivíduos, os groupwares devem se preocupar com a integração e adequação dos participantes aos sistemas e com a flexibilização da aplicação.

A utilização de um conjunto de componentes pré-definidos e reutilizáveis constitui uma das abordagens mais populares para auxiliar na construção de aplicações cooperativas [3]. O desenvolvimento de tal conjunto de componentes, também conhecido por toolkit, tem sido alvo de muita pesquisa na área de sistemas cooperativos. Um bom toolkit deve oferecer construções de alto nível, contendo suporte para aspectos pertinentes à arquitetura distribuída [8], minimizando o trabalho dos projetistas. Além de oferecer facilidades para a criação do groupware, os toolkits também devem prover uma arquitetura para gerenciar a aplicação em tempo de execução, podendo até existir várias visualizações para um mesmo conjunto de dados.

O desenvolvimento de software através de componentes baseia-se na construção e utilização de blocos funcionais, que podem ser combinados para o desenvolvimento de



---

uma aplicação, ao invés de desenvolver-se a aplicação de uma forma monolítica. Cada bloco, chamado de componente de software, provê um serviço que é disponível através de sua interface. Um componente de software, ou simplesmente componente, é uma unidade de software capaz de ser instalada e executada por si só, independente, reutilizável, com interfaces bem definidas e que pode ser utilizada em diversas aplicações.

## **1.2. Motivação**

A tendência atual para o desenvolvimento de software não é mais a de se construir sistemas monolíticos, isolados e independentes dos demais, mas sim a de utilizar os componentes de software e agregá-los, como na montagem de um quebra-cabeça, para desenvolver as aplicações.

A tecnologia de componentes de software possibilita aos desenvolvedores uma economia de tempo de desenvolvimento, uma vez que se baseia em conceitos de reutilização e independência de plataforma através da utilização de estruturas gerais e portáteis. É uma tecnologia atraente já que há redução dos custos de desenvolvimento, pois sendo os componentes “pedaços” de programa previamente desenvolvidos, que podem ser utilizados no desenvolvimento de várias aplicações, seus custos serão diluídos entre os vários projetos que os utilizarão.

A maioria dos softwares existentes atualmente suporta somente a interação entre um usuário e o sistema. Neste sentido, em tarefas como a edição de um documento ou a pesquisa em um banco de dados somente um usuário interage com o computador. O suporte à atividade cooperativa é necessário uma vez que, para o cumprimento de um objetivo comum, as atividades envolvendo um grupo de pessoas são em maior quantidade que as atividades individuais.

Além disso, os conjuntos de componentes (toolkits) existentes atualmente enfrentam um grande problema que é a falta de generalidade. Não sendo genéricos, estes refletem as características da área específica para a qual foram desenvolvidos, limitando as aplicações que poderiam ser construídas a partir da combinação e extensão destes e limitando a reutilização dos mesmos.

---

Neste sentido, criando-se um conjunto de componentes genéricos e que poderiam ser combinados e estendidos de diversas formas, liberando os desenvolvedores e projetistas de certas atividades complexas e rotineiras, estes poderão dedicar seu tempo exclusivamente para solução dos problemas característicos da aplicação cooperativa a ser construída.

### **1.3. Objetivo e Escopo**

Este projeto visa à identificação e ao desenvolvimento de um conjunto de componentes de software para aplicações cooperativas, mais precisamente para um ambiente cooperativo para conferência multimídia, e está inserido em um contexto mais amplo, do qual faz parte um trabalho de doutorado [7]. Este trabalho está sendo desenvolvido na Universidade de Twente na área de CSCW e propõe uma abordagem baseada em componentes para o desenvolvimento de aplicações cooperativas. Esta abordagem é estudada e aplicada neste projeto, para a identificação e desenvolvimento dos componentes de CSCW.

### **1.4. Estrutura do Documento**

Este documento está dividido em cinco capítulos:

- O Capítulo II apresenta um estudo mais aprofundado sobre groupwares. É apresentada uma definição de groupware, como os groupwares são classificados, quais são as formas de se desenvolver um groupware e, finalmente, são definidos e mostrados alguns toolkits cooperativos.
- O Capítulo III define e exemplifica alguns componentes de software, faz uma breve introdução à Unified Modeling Language (UML – Linguagem de Modelagem Unificada) [2, 13] e apresenta a metodologia a ser utilizada no desenvolvimento de sistemas baseado em componentes [6, 7], que utiliza diagramas UML. Também são mostradas as tecnologias a serem utilizadas neste projeto, a saber: Enterprise Java Beans (EJB) [21], plataforma Java 2 Enterprise Edition (J2EE) [22] e Java Media Framework (JMF) [23].
- O Capítulo IV descreve a aplicação da metodologia para desenvolvimento de sistema baseado em Componente [6, 7]. Para tanto, é apresentada uma descrição

---

informal da aplicação a ser modelada – sistema de conferência multimídia - , partes da modelagem desta aplicação conforme os níveis de abstração propostos pela metodologia e identificados alguns componentes genéricos para aplicações cooperativas.

- O Capítulo V apresenta a arquitetura do toolkit para aplicações cooperativas e os componentes cooperativos que fazem parte deste, identificados neste projeto.
- Finalmente, o Capítulo VI apresenta algumas conclusões e menciona alguns trabalhos futuros.

---

## Capítulo II

### **Groupware**

Este capítulo define o termo Groupware e apresenta uma classificação para o mesmo como uma tentativa de explicar esta tecnologia. É, também, introduzido neste capítulo, o conceito de toolkits cooperativos, juntamente com a descrição de alguns exemplos destes toolkits.

#### **2.1. Definição de Groupware**

Enquanto o termo CSCW tem sido usado para representar a pesquisa na área do trabalho cooperativo, e como os computadores podem auxiliar os grupos na execução de suas atividades, o termo Groupware tem sido usado para designar a tecnologia (hardware/ou software) gerada pela pesquisa em CSCW [5]. Apesar disso, algumas vezes, ambos são usados como sinônimos de tecnologia CSCW. Assim, os produtos comerciais e outros softwares baseados em computador para auxiliar no trabalho cooperativo são normalmente referenciados como exemplos de groupware [5]. Um groupware é um sistema baseado em computador que suporta um grupo de pessoas engajadas na realização de uma tarefa comum e que provê uma interface para o ambiente compartilhado [5]. O objetivo de um groupware é auxiliar os grupos no que diz respeito à realização das tarefas cooperativas através da comunicação, colaboração e coordenação de suas atividades, quando utilizando hardware e tecnologia de computação. Assim, sistemas de correio eletrônico, teleconferências, salas de reuniões eletrônicas, suporte à decisão e editores de texto cooperativos são exemplos de groupware.

A noção de tarefa comum e de ambiente compartilhado são cruciais para a definição de groupware. Isto exclui sistemas multi-usuários, assim como sistemas de tempo compartilhado, onde os usuários podem não realizar uma tarefa comum. A definição não especifica que os usuários devem estar em atividade simultaneamente. Groupwares que suportam atividade simultânea são chamados groupwares de tempo real.

Não há uma linha rígida de divisão entre os sistemas que são considerados groupware e os que não são, pois um sistema pode suportar a realização de tarefas comuns e

---

ambiente compartilhado em graus variados. Pode-se, então, pensar numa área de alcance do groupware, com diferentes sistemas posicionados em diferentes locais nesta área de alcance, como mostrado na figura 2.1. Assim, de acordo com a definição de realização de tarefa comum, pode-se observar os dois extremos dessa área de alcance dos groupwares da seguinte forma:

- a) por um lado, tem-se um sistema convencional que suporta muitos usuários concorrentemente realizando suas tarefas separada e independentemente. Uma vez que eles não estão trabalhando em uma tarefa comum de uma forma fortemente acoplada, este sistema tem um baixo alcance de atividade cooperativa quanto à realização da tarefa comum;
- b) por outro lado, encontra-se um software para sistema de revisão que, eletronicamente, permite que um grupo de projetistas avalie um módulo do projeto durante uma interação em tempo real. Este sistema auxilia as pessoas que estão concentrando-se em uma tarefa específica ao mesmo tempo e interagindo de forma fechada. Este sistema possui um alto grau de alcance da atividade cooperativa quanto à realização de uma tarefa comum.

Agora, de acordo com a dimensão de ambiente compartilhado e como estes se comportam na área de alcance do groupware, pode-se observar os seguintes extremos:

- a) um sistema típico de correio eletrônico transmite mensagens, mas provê pouca interação. É, então, considerado um groupware de baixo alcance quanto à dimensão de ambiente compartilhado;
- b) um sistema de sala de aula eletrônica usa múltiplas janelas para enviar informações sobre a matéria que está sendo ensinada. Emulando uma sala de aula tradicional, este sistema permite ao instrutor apresentar aos estudantes uma leitura simultânea em estações de trabalho (workstations) individuais remotas. Além disso, o quadro negro controlado pelo professor mostra uma janela com a lista de presença dos alunos, perguntas e comentários dos estudantes. Alguns comandos facilitam a distribuição de leituras e interação da classe. Este sistema representa um groupware com um alto alcance quanto à dimensão de ambiente compartilhado.

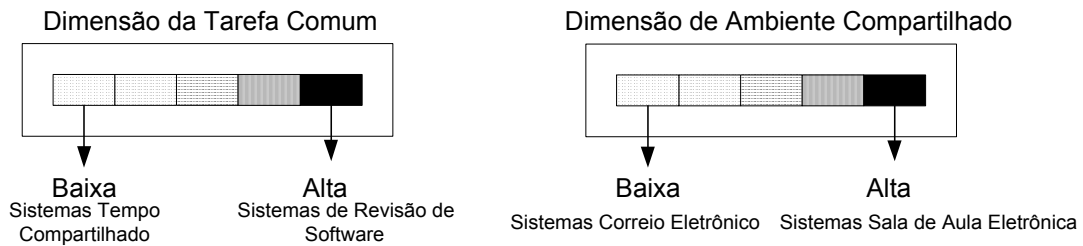


Figura 2.1 Duas Dimensões de Alcance do Groupware

## 2.2. Classificação de Groupware

Para visualizar a variedade de groupwares e tentar entender o que vem a ser esta tecnologia são analisadas duas classificações de groupware. A primeira classificação é baseada em noções de tempo e espaço, enquanto a segunda é baseada na funcionalidade da aplicação.

### 2.2.1. Tempo versus Espaço

Um groupware pode ser concebido para auxiliar os grupos que trabalham face-a-face ou grupos que trabalham de forma distribuída, em diferentes locais. Além disso, um groupware pode ser desenvolvido para aumentar a comunicação e colaboração dentro de um ambiente com interações em tempo real (forma síncrona) ou para trabalhar de forma assíncrona. Estas considerações sobre tempo e espaço sugerem quatro categorias de groupwares, representados por uma matriz de 2x2 mostrada pela figura 2.2.

|                   | Mesmo Tempo   | Tempos Diferentes   |
|-------------------|---|---|
| Mesmo Local       | <p><b>Interação face-a-dace</b><br/>Sala de Reunião</p>                             | <p><b>Interação Assíncrona</b><br/>Quadro de Avisos</p>               |
| Locais Diferentes | <p><b>Interação Distribuída Síncrona</b><br/>Editor de Documentos em Tempo Real</p> | <p><b>Interação Distribuída Assíncrona</b><br/>Correio Eletrônico</p> |

Figura 2.2. Classificação de Groupware de acordo com Tempo e Espaço

### 2.2.2. Funcionalidade da Aplicação

A intenção desta classificação é dar uma idéia geral da extensão do domínio dos groupwares.

- 
- a) Sistemas de Mensagens: é um exemplo muito familiar de groupware que suporta troca assíncrona de mensagens textuais entre grupos de usuários. São exemplos deste tipo de aplicação o correio eletrônico e as listas de discussão. Com a proliferação do uso da Internet a quantidade de mensagens trocadas é tão grande que fez surgir alguns sistemas de mensagens “inteligentes”, agindo de forma a classificar automaticamente as mensagens de acordo com seu conteúdo. Um exemplo é o Information Lens [appud 5], onde o usuário pode arquivar ou redirecionar as mensagens automaticamente, através da especificação de regras.
- b) Sistemas de Comunicação Estruturados: estão relacionados com a organização das informações derivadas da comunicação. Estes sistemas procuram, de forma automática, priorizar as mensagens de texto e voz e prover um mecanismo para ajudar a determinar quais artigos são interessantes ou não em uma lista de discussão, por exemplo, através de uma filtragem baseada em valores atribuídos a cada artigo. São exemplos de sistemas de comunicação estruturados GroupLens [18] e CLUES [11].
- c) Editores Multiusuários: membros de um grupo podem utilizar editores multiusuários ou sistemas de co-autoria para, juntamente, compor e editar um documento, seja um gráfico ou um texto. Isso significa que há uma área de trabalho comum onde todos atuam e podem visualizar a atuação dos outros. Alguns destes editores são para uso assíncrono e separam o texto fornecido pelo autor dos comentários dos vários revisores, como o ForComment [appud 5]. Para que um editor multi-usuário possa ser considerado síncrono, este deve oferecer controle de concorrência e mecanismos de atualização automáticos. Isto é, deve ser possível que um usuário possa, por exemplo, editar uma frase de um parágrafo do texto, enquanto outro está atualizando a frase seguinte, sendo possível que ambos visualizem em tempo real o que o outro está fazendo. São exemplos de editores multiusuários Collaborative Editing System (CES) [appud 5], Shared Book [appud 5], Quilt [appud 5], Sepia [9] e Duplex [15]. Alguns editores fornecem notificação explícita aos usuários, como o Mercury [appud 5] usado por programadores, que informa aos usuários quando seus códigos precisam ser alterados em consequência de modificações realizadas por outras pessoas.
- d) Sistemas para Suporte de Decisão de Grupo e Sala de Reunião Eletrônica: estes sistemas provêm facilidades baseadas em computador para a solução de

---

problemas não estruturados em um ambiente de grupo, cujo objetivo é melhorar a produtividade do processo de tomada de decisão, agilizando o processo decisório ou melhorando a qualidade dos resultados da decisão. Muitos sistemas para suporte de decisão de grupo são implementados como sala de reunião eletrônica, que contém vários computadores conectados em rede; telões e equipamentos de áudio e vídeo para apoiar reuniões face-a-face. Estes sistemas incluem assistência à preparação da pauta de reunião, à geração de idéias, à organização e priorização das idéias propostas e à tomada de decisão. A Universidade do Arizona possui um sistema deste tipo, o PlexCenter Planning and Decision Support Laboratory [appud 5], cujo ambiente é formado por uma mesa de conferência em forma de U com estações de trabalho interligadas entre si e conectadas a um telão.

- e) Conferência: o computador pode ser utilizado como um meio de comunicação de várias formas. Uma delas é auxiliar as pessoas na realização de conferências. Sistemas de conferência utilizam-se de texto, áudio e vídeo para prover o suporte para a realização de uma atividade cooperativa. São três as abordagens através das quais as pessoas podem realizar uma conferência: conferência assíncrona, conferência em tempo real e teleconferência. Conferência assíncrona, em geral, permite a realização de encontros onde os participantes “comparecem” de acordo com sua própria disponibilidade. Desta forma, os participantes têm tempo para examinar as informações que necessitam e podem contribuir para a discussão elaborando textos e disponibilizando-os no sistema. Conferência em tempo real permite a um grupo de usuários, presentes em uma sala de reunião eletrônica ou fisicamente distribuídos, interagir de forma síncrona através de seus terminais ou estações de trabalho, mas não oferecem capacidades para visualização de imagens. São exemplos de conferência em tempo real: Real Time Calendar (RTCAL) [appud 5] e Cognoter [appud 5]. Teleconferência é a abordagem que utiliza algum suporte de telecomunicação para permitir a interação em grupo. As teleconferências tendem a ser complicadas, pois precisam de salas especiais e operadores treinados e têm a desvantagem de não permitir aos participantes compartilhar textos e gráficos. Como exemplo de teleconferência tem-se CRUISER [appud 5]. Uma combinação das vantagens da conferência em tempo real e da teleconferência dá origem à conferência desktop. Este tipo de



---

conferência suporta visões dinâmicas da informação e imagens dinâmicas dos participantes. Alguns exemplo são MMConf [appud 5] e Rapport [appud 5].

- f) Quadro Branco Compartilhado (Shared-Whiteboards): estes sistemas permitem compartilhar informações entre as pessoas que utilizam um quadro branco e computadores. O ambiente gerado pela comunicação entre os vários participantes/equipamentos permite apresentar e compartilhar objetos capturados de uma câmera de vídeo, arquivos gráficos ou textos escritos no quadro branco, ou até mesmo as anotações realizadas individualmente em cada computador. Pode-se, então, reproduzir a situação de uma sala de aula onde o professor compartilha com os alunos as informações colocadas no quadro. A qualquer tempo o controle dos objetos a serem manipulados pode ser transferido de um participante para outro, como por exemplo o professor passar o controle para qualquer aluno da sala de aula.
- g) Sistemas de Coordenação: estes sistemas permitem que os indivíduos visualizem suas ações, assim como as ações relevantes de outras pessoas, visando a atingir um objetivo global [5]. Estes sistemas também podem provocar “disparos” ligados às ações dos usuários informando a cada usuário, que participa da realização de uma tarefa comum, os estados de suas ações e suas condições de espera, ou gerando lembretes e alertas automáticos. Existem quatro abordagens para este tipo de sistema: formulário, procedimento, conversação e estrutura de comunicação. Os sistemas orientados a formulário são chamados de sistemas de workflow ou de controle de fluxo de trabalho. Basicamente este tipo de sistema preocupa-se com o direcionamento dado aos documentos (formulários) em uma organização. Os sistemas orientados a procedimentos os vêem como uma seqüência de tarefas programáveis, de forma que o resultado de uma tarefa alimente a tarefa seguinte, até que todos os procedimentos necessários tenham sido executados. Os sistemas orientados à conversação controlam os pedidos e compromissos dos indivíduos, através da monitoração de sua correspondência eletrônica; e os sistemas orientados à comunicação estruturada controlam as atividades de uma organização com base no relacionamento entre as funções dos membros desta.
- h) Sistemas de Gerenciamento de Workflow: o fluxo de trabalho em uma organização compreende uma seqüência de ações realizadas por indivíduos, de forma que a empresa venha a atingir os objetivos a que propôs realizar. Os

---

sistemas para gerenciamento de workflow envolvem a definição, execução e gerenciamento de fluxos de trabalho visando a controlar os dados e os procedimentos diários dentro de uma organização de forma a automatizar a rotina dos indivíduos quando trabalhando em conjunto. São exemplos de sistemas de gerenciamento de workflow Action Workflow [12] e Mobile [10].

### **2.3. Toolkits Cooperativos**

O desenvolvimento de groupware não é uma tarefa simples, pois além de problemas característicos das relações entre indivíduos esta tarefa é dificultada por problemas pertinentes a sistemas distribuídos.

Numa tentativa de reduzir a complexidade de implementação de aplicações de groupware, vários pesquisadores têm explorado toolkits. Os toolkits são um conjunto de componentes pré-definidos e reutilizáveis, com a finalidade de oferecer poderosas ferramentas e infraestrutura suficientes para permitir que o programador desenvolva groupware de alta qualidade com razoável esforço.

Um toolkit cooperativo provê um conjunto de objetos, mecanismos e comportamentos que podem ser combinados para criar aplicações cooperativas [4], permitindo que estas sejam desenvolvidas mais rapidamente por possibilitarem o reuso dos componentes. O reuso é, sob o ponto de vista do software, um dos principais objetivos do projeto de um toolkit. Quando embutidos em um toolkit, os componentes, os algoritmos e as implementações devem ser combinados e reusados em diferentes aplicações.

Um toolkit cooperativo deve, também, ser flexível, permitindo sua adaptabilidade a cada situação. A flexibilidade é a capacidade de um toolkit suportar vários tipos de aplicações. A flexibilidade pode ser vista sob quatro aspectos diferentes [3]:

- um sistema flexível, que provê comportamentos e objetos genéricos os quais podem ser combinados e usados de várias formas por diferentes pessoas usuárias do sistema;
- um sistema parametrizado, que provê objetos com várias alternativas de comportamento;
- um sistema integrável, que pode ser conectado a outros componentes do sistema;

- um sistema adaptável, que permite aos usuários alterarem o próprio sistema, adicionarem funcionalidades ou comportamentos específicos.

Flexibilidade e adaptação são o coração do projeto de toolkit, visto que isto determina o reuso. Por um lado os toolkits devem esforçar-se para maximizar a generalidade, de forma a tornar a estrutura dos componentes comuns, podendo ser utilizadas em várias aplicações. Por outro lado, devem ser instanciados e manipulados; assim, uma grande quantidade de aplicações pode ser derivada dos componentes.

Analisando os toolkits cooperativos desenvolvidos até os dias de hoje, observa-se que os mesmos contribuíram para o surgimento de duas gerações, cada uma delas decorrente da forma pela qual o toolkit realiza sua funcionalidade. A primeira geração, como MMConf [appud 5] e Rendezvous [appud 3, appud 8], preocupou-se, principalmente, em encapsular comportamentos comuns de várias aplicações cooperativas (aspectos da distribuição, por exemplo) a fim de facilitar o desenvolvimento das novas aplicações e reduzir o tempo de desenvolvimento [4]. A segunda geração de toolkits CSCW tem se preocupado com a flexibilidade [4]. Os toolkits desenvolvidos recentemente têm oferecido estruturas abertas para o programador criar vários tipos de comportamento dentro de uma infra-estrutura pré-definida. As próximas sub-seções apresentam alguns exemplos de toolkits cooperativos.

### 2.3.1. Live

Live [1] consiste de um conjunto de componentes para o desenvolvimento de aplicações cooperativas síncronas. Os componentes deste toolkit foram desenvolvidos na linguagem Java, sob uma arquitetura cliente/servidor centralizada e possibilitam a integração com ferramentas e com outros componentes oferecidos por outros vendedores. Dentre os componentes desenvolvidos encontram-se componentes para o estabelecimento e manutenção de conferências (conexão e desconexão), o convite de participantes para integrar uma atividade cooperativa, para o compartilhamento de dados e eventos, para a sincronização de atividades, para a gravação, envio e recepção de áudio e vídeo e para o armazenamento e a recuperação de uma seqüência de eventos. Estes componentes oferecem suporte de alto nível para integrar aplicações cooperativas com componentes multimídia através de construtores visuais e podem ser utilizados dentro de um ambiente visual, para rapidamente construir uma grande variedade de

---

aplicações cooperativas síncronas. Como características destes componentes encontra-se a possibilidade de personalização das propriedades, de utilização de um conjunto de métodos e de ativação de eventos.

### 2.3.2. GroupKit

GroupKit [appud 8, 19] é um toolkit cooperativo para o desenvolvimento de aplicações síncronas. Oferece flexibilidade através do uso de uma facilidade chamada protocolo aberto, permitindo a interação entre os diferentes componentes do sistema. O protocolo aberto possibilita que os desenvolvedores encapsulem alguns aspectos (que podem ser controlados) do comportamento do sistema através da definição de políticas. Os clientes podem adotar políticas diferentes de forma que comportamentos diferentes possam ser exibidos, tornando o sistema mais flexível e adaptável. GroupKit emprega uma arquitetura semi distribuída contendo três tipos de componentes: um processo central, chamado Registrar, um gerente de sessão e uma conferência. O Registrar é o único processo centralizado do sistema, agindo como um ponto de conexão entre uma comunidade de conferências. O gerente de sessão é um processo replicado (um por participante) que provê tanto uma interface com o participante quanto uma política de como as conferências são criadas e destruídas e de como os participantes entram e saem de uma conferência. Uma conferência é uma aplicação invocada por um participante através do gerente de sessão. Várias conferências podem ser criadas por um mesmo participante. Groupkit também contém componentes para o controle de acesso a elementos compartilhados, para a propagação e notificação de eventos, para a manutenção do status de um participante, conscientização das ações dos outros participantes, entre outros.

### 2.3.3. Rendezvous

Rendezvous [appud 3, appud 8] é um toolkit cooperativo projetado para construir aplicações síncronas e gráficas. Possui arquitetura centralizada e foi escrito em Common Lisp. A idéia básica é montar um único modelo de dados abstrato compartilhados por todos os clientes. Este toolkit coloca os dados e as visões do modelo em um único processador, contendo sempre o estado correto da aplicação; mas, em contrapartida é lento, pois todas as aplicações rodam sobre o mesmo processador. O mecanismo básico deste toolkit é a estrutura Abstraction-Link-View (ALV – Abstração-Ligação-Visão). ALV captura aspectos compartilhados da apresentação da aplicação. A

---

abstração refere-se ao modelo de dados. As visões representam os usuários individuais com mecanismos de interação e de interface próprios. Diferentes usuários podem ver diferentes representações dos dados. As ligações (links) relacionam abstrações às visões. Os links são responsáveis por garantir que as visões sejam apropriadamente atualizadas quando alterações forem realizadas nas abstrações comuns. Seus projetistas sugerem que uma arquitetura mista seria interessante: dados compartilhados em um site central e as visões duplicadas nos clientes.

#### 2.3.4. COAST

COAST (Cooperative Application Systems Toolkit) [20] é um toolkit baseado em objetos para o desenvolvimento de aplicações síncronas e foi implementado em VisualWorks Smalltalk. COAST utiliza uma arquitetura distribuída replicada, na qual cada aplicação mantém uma cópia do documento sendo compartilhado. Cada aplicação manipula um documento compartilhado, que pode ser subdividido em várias partes. O acesso ao documento ou às suas partes é coordenado através de um objeto sessão. Os usuários da aplicação interagem com o documento compartilhado através de objetos-visões, para visualizar o documento em uma janela, e objetos controladores, para processar a entrada de dados dos usuários. Tanto objetos-visões quanto objetos-controladores têm acesso ao documento através do objeto sessão. Um gerente de sessão permite que os usuários criem, juntem-se e deixem sessões. Objetos-usuários representam os vários usuários de um documento. Um gerente de transação assegura a integridade dos objetos compartilhados, tais como (partes de) documentos, objetos-sessões e objetos-usuários. Um gerente de replicação é responsável pela sincronização dos objetos replicados.

#### 2.3.5. PROSPERO

Prospero [3, 4] é um toolkit destinado ao desenvolvimento de aplicações tanto síncronas quanto assíncronas e pertence à segunda geração de toolkits cooperativos. Prospero oferece basicamente dois conjuntos de abstrações para o desenvolvimento de aplicações cooperativas: a divergência e a sincronização de sequências de atividades e o gerenciamento de consistência através da troca de promessas e garantias. Uma sequência de atividades é um conjunto de operações relacionadas sequencialmente que são coerentes no tempo. O uso de promessas e garantias é uma alternativa mais flexível para se obter a consistência de uma informação, se comparado aos tradicionais

---

mecanismos de exclusão adotados em vários groupwares. Uma garantia de consistência é obtida em retorno a uma promessa sobre uma atividade futura a ser realizada. Diferentes níveis de consistência podem ser obtidos baseando-se nas garantias emitidas pelos diversos participantes ou componentes da aplicação. Este toolkit foi implementado em CLOS (Common Lisp Object System), a comunicação entre os clientes PROSPERO foi implementada utilizando RPC (Remote Procedure Call – chamada remota a procedimentos) e não provê funcionalidade para interfaces do usuário.

### 2.3.6. Visual Obliq

Visual Obliq [appud 1] é um toolkit que foi escrito na linguagem Obliq. Este toolkit provê componentes visuais para a criação de interfaces gráficas para aplicações cooperativas e as funcionalidades cooperativas devem ser especificadas textualmente pelo programador.

### 2.3.7. ViewSoft

Viewsoft [appud 1] é um ambiente para desenvolvimento rápido de aplicação cliente-servidor, gerando aplicações C++ no servidor e Java no cliente, necessitando uma pré-configuração do servidor. É bem adaptado para trabalhar com aplicações simétricas, onde todas as instâncias cliente das aplicações cooperativas têm capacidades e interfaces gráficas equivalentes.

### 2.3.8. Suite

Suite [appud 3] é um toolkit para aplicação multi-usuário, derivado de um toolkit para aplicação mono-usuário. Suite adota uma abordagem baseada em editor para atividades cooperativas em um ambiente compartilhado, definindo clientes como editores para dados estruturados. Provê um grande número de níveis de flexibilidade dando ao programador o controle sobre vários projetos. Oferece um framework para prover um controle variável sobre as atualizações/visualizações nos ambientes compartilhados.

### 2.3.9. Intermezzo

Intermezzo [appud 4] é um toolkit cooperativo que provê suporte para flexibilidade através de dois mecanismos: primeiro, as aplicações podem transferir seus códigos para os toolkits. Estes códigos são escritos em uma linguagem interpretada simples, são

---

interpretados pelo toolkit em tempo de execução e sua ativação pode ser controlada por parâmetros que o programador pode ajustar de modo que o sistema responda às mudanças particulares na atividade colaborativa. O segundo mecanismo de suporte é o uso de regras dinâmicas como um mecanismo para configuração do contexto. O resultado é que visões dos dados armazenados tornam-se independentes, permitindo um alto nível de flexibilidade.

#### 2.3.10. Oval

Oval (Objects, Views, Agents and Links – Objetos, Visões, Agentes e Ligações) [appud 3] é um toolkit que oferece quatro componentes básicos, de onde deriva-se seu nome: objetos, visões, agentes e links. Uma definição de objeto provê uma lista de campos que o objeto deve conter. Visões são visualizações dos objetos, permitindo que os mesmos sejam apresentados de várias formas. Agentes são regras de produção associadas com objetos, as quais são disparadas quando suas condições são satisfeitas. Os links relacionam os objetos, permitindo a criação de redes, hierarquias, etc. Este toolkit provê, essencialmente, uma linguagem de programação visual, expressa em termos de objetos e derivações criadas pelos usuários. Assim, uma aplicação pode ser construída através da definição de um conjunto de objetos, agentes para ativar comportamentos de entrada ou de cálculo e fornecer um conjunto de visualizações apropriadas.

#### 2.3.11. Sepia

Sepia [9, appud 3] é um sistema de hipertexto cooperativo que suporta uma grande variedade de aplicações baseadas no modelo de co-autoria e interação sobre documentos hipermídia. Consiste em modelo de armazenamento persistente que provê suporte para colaboração síncrona e assíncrona, aceitando três formas de interação: individual, fracamente acoplada e fortemente acoplada. Trabalho individual compreende a atividade sobre um nó hipertexto que é o foco da atenção para somente um usuário. Trabalho fracamente e fortemente acoplado compreendem atividades simultâneas sobre alguns nós por um grupo de usuários, e diferem entre si quanto às formas nas quais a atividade de qualquer usuário reflete as atividades dos outros. Quando dois usuários estão fracamente acoplados, cada um é livre para alterar o documento e trabalhar separadamente, ainda que tenham informação sobre a presença dos demais usuários, através de um mecanismo implícito de bloqueio para evitar conflitos. Quando eles estão fortemente acoplados, compartilham a mesma visão do documento. A transição entre os

---

modos individual e fracamente acoplado acontece automaticamente, quando dois usuários acessam o mesmo nó; a transição para o trabalho fortemente acoplado pode ser controlada pelo usuário.

### 2.3.12. Clock

Clock [appud 8] é um toolkit implementado na linguagem clock. Possui uma arquitetura flexível onde os programadores não se preocupam com detalhes de distribuição, redes e controle de concorrência. Possui uma arquitetura abstrata que captura a estrutura dos programas, mas não especifica como o programa será implementado quanto a um contexto distribuído, podendo ser obtida uma arquitetura centralizada, distribuída ou mista. Consiste em separar o modelo abstrato das suas visões, como o Rendezvous [appud 3, appud 8], por isso o desenvolvimento das aplicações cooperativas é rápido e de fácil manutenção. O sistema é responsável, em tempo de execução, em implementar a comunicação da rede e o controle de concorrência.

Analisando os diversos toolkits apresentados neste capítulo observa-se uma grande quantidade destes devido à ampla abrangência do escopo de CSCW. Suas características variam de acordo com a sua funcionalidade: toolkits voltados para o desenvolvimento de groupware referentes a editores de texto, hipertexto, conferências e aplicações gráficas, dentre outras. Observa-se, também, que alguns apresentam uma arquitetura flexível, enquanto outros apresentam uma arquitetura rígida. Alguns possuem uma arquitetura centralizada, outros uma arquitetura distribuída e alguns, ainda, combinam essas duas.

Dentre as várias ramificações existentes sob o escopo do trabalho cooperativo escolheu-se, para este projeto de mestrado, o desenvolvimento de componentes que auxiliem na construção de groupware relacionados à conferência multimídia. Assim, características como o gerenciamento de sessão e de participantes, o estabelecimento de uma sessão e a cooperação entre os participantes da sessão com suporte à comunicação através do envio de mensagens e da transmissão/recepção de áudio e vídeo devem estar presentes nestes componentes. Além disso, a flexibilidade, que caracteriza a segunda geração dos toolkits, também deve fazer parte destes componentes. Características quanto à arquitetura centralizada, distribuída ou mista a ser adotada para os componentes identificados e desenvolvidos por este projeto, são definidas no decorrer deste estudo.



O estudo inicial sobre CSCW e groupware, assim como uma pequena introdução aos toolkits cooperativos, tiveram como finalidade apresentar de que forma a comunicação, através da utilização de computadores, permite que pessoas de uma mesma organização ou de diferentes organizações troquem informações rapidamente, eliminando distâncias e aumentando a produtividade, consolidando a necessidade de softwares para auxiliar a realização de trabalhos em grupo.

Os próximos capítulos devem: definir componentes de software, explicar como identificar e desenvolver os componentes de software para aplicações cooperativas através da utilização de uma metodologia proposta em [6, 7], mostrar a tecnologia a ser utilizada, descrever um sistema de conferência multimídia para se aplicar a metodologia e propor componentes genéricos para aplicações cooperativas a partir do estudo de caso realizado sobre o sistema de conferência multimídia.

## Capítulo III

### ***Desenvolvimento de Software Baseado em Componentes***

Este capítulo apresenta o desenvolvimento de software baseado em componentes. Aqui o conceito de componentes de software é definido e alguns componentes necessários em aplicações cooperativas são apresentados. Também é objeto de estudo deste capítulo a Linguagem de Modelagem Unificada (UML – Unified Modeling Language) [2, 13] utilizada para modelar a aplicação a ser desenvolvida e a metodologia para desenvolvimento de software baseado em componentes [6, 7]. É apresentada a plataforma de desenvolvimento dos componentes composta por Enterprise JavaBeans [21], Java 2 Enterprise Edition (J2EE) [22] e Java Media Framework (JMF) [23].

#### **3.1. Componentes de Software**

O reuso é um assunto chave para a Engenharia de Software, cujos benefícios incluem a redução de custo e do tempo para desenvolver o software. O desenvolvimento de software orientado a objeto tem por objetivo prover o reuso do objeto no nível de projeto e implementação. Já o desenvolvimento baseado em componentes de software tem por objetivo prover o reuso dos componentes no nível de implementação e aumentar a interoperabilidade entre as partes do software. Assim, os componentes de software representam parte da funcionalidade que está pronta para ser instalada e executada em vários ambientes [7].

Componentes de software, doravante denominados somente componentes, são blocos de construção prontos para serem instalados e executados, independentes e reusáveis que provêm um único serviço e podem ser usados individualmente ou em composição com outros componentes que oferecem outros serviços [24]. Um componente provê acesso ao seu serviço através de uma ou mais interfaces. As interfaces do componente são um conjunto de operações que especificam os serviços oferecidos pelos componentes. Estes serviços podem, normalmente, ser personalizados, ajustando algumas propriedades do componente.

Em uma abordagem voltada para o desenvolvimento de software baseado em componentes, os seguintes princípios são desejáveis:

- a) cada componente deve prover uma funcionalidade específica (serviço único);
- b) dependências entre componentes devem ser minimizadas;
- c) os componentes devem ser genéricos, a fim de serem reutilizados em um maior número possível de aplicações;
- d) os componentes devem ser facilmente combináveis a fim de facilitar a montagem da aplicação;
- e) os componentes devem ser projetados de tal forma que possam ser futuramente estendidos, dado que não é possível prever todas as necessidades de um serviço oferecido por um componente em um primeiro momento;
- f) os componentes devem ser utilizados em uma grande variedade de plataformas operacionais (interoperabilidade);
- g) os componentes devem possuir propriedades as quais podem ser modificadas para adaptar o serviço provido pelo componente a fim de atender uma necessidade ou situação específica.

Como exemplos de componentes utilizados em aplicações cooperativas tem-se:

a) componentes para o gerenciamento de uma sessão

O gerenciamento de uma sessão cooperativa é uma atividade fundamental em um groupware. As atribuições deste conjunto de componentes incluem o estabelecimento de uma nova sessão, a sincronização e o contato com outras sessões, o controle do acesso de participantes a uma sessão, o controle da entrada e saída de participantes de uma sessão, o suporte para participantes atrasados, o suporte para a persistência ou não de uma sessão quando da ausência de participantes etc.

b) componentes para o convite de participantes

Para que um participante una-se a uma sessão colaborativa, o mesmo deve ser notificado da existência desta sessão. O convite para unir-se a uma sessão pode incluir vários tipos de informações, tais como: o nome da sessão, quem é o responsável por esta sessão, como e quando a atividade cooperativa será realizada, as políticas de acesso adotadas durante a sessão etc. O convidado pode confirmar sua presença, recusar ou unir-se a esta sessão.

c) componentes para o suporte de multimídia

A presença de multimídia (áudio e vídeo) nas aplicações cooperativas é reconhecidamente uma necessidade. Contudo, poucos são os toolkits que provêem este tipo de suporte. Frequentemente, meios adicionais, como telefones e sistemas de videoconferência, são empregados para suprir a ausência integrada de multimídia numa aplicação cooperativa. Pelo menos o envio e recepção de áudio e vídeo devem ser suportados. O armazenamento e a recuperação dessas mídias é desejável, mas não é fundamental.

d) componentes para o compartilhamento de dados e eventos

O compartilhamento de objetos, tais como texto, números, buffers e tabelas, deve ser suportado. Cuidados especiais devem ser tomados para manter-se a consistência dos dados, principalmente quando estes forem replicados. Mecanismos de acesso e controle, tais como o bloqueio do acesso aos dados e a utilização de tokens para garantir o acesso aos mesmos, devem estar presentes. Os eventos realizados em uma interface devem ser rapidamente processados e compartilhados com os demais participantes. O armazenamento e a recuperação de eventos é desejável à medida que participantes atrasados possam adquirir consciência do que foi feito até então.

e) componentes para a sincronização de atividades

Deve ser possível realizar uma atividade com diferentes níveis de consistência e sincronização. O tratamento de conflitos, nestes casos, pode tanto ser feito automaticamente, quando os níveis de consistência e sincronização são elevados, ou pelos próprios participantes, quando os níveis de consistência e sincronização são baixos ou inexistentes (atividades assíncronas).

### **3.2. Unified Modeling Language (UML – Linguagem de Modelagem Unificada)**

Unified Modeling Language (UML – Linguagem de Modelagem Unificada) [2, 13] é uma linguagem de modelagem padronizada pela OMG (Object Management Group – Grupo de Gerência de Objetos) para a especificação, visualização, construção e documentação de software. UML é o resultado da combinação de, basicamente, três

---

métodos de modelagem de software: Booch, Object Modeling Technique (OMT) e Object-Oriented Software Engineering (OOSE) [2].

UML é independente de linguagens de programação e de processos de desenvolvimento. Esta linguagem consiste basicamente de uma coleção de diagramas que são utilizados para modelar um sistema a partir de diferentes perspectivas. Os diagramas de UML podem ser agrupados em duas categorias: diagramas estruturais e diagramas comportamentais.

São quatro diagramas estruturais utilizados para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema:

- *diagrama de classes*, o qual descreve os tipos de objetos (classes, tipos de dados e interfaces) e os diferentes tipos de relacionamentos estáticos que os conectam;
- *diagrama de objetos*, o qual representa uma instância em particular de um diagrama de classes;
- *diagrama de componentes*, o qual descreve os componentes de software do sistema, tais como código fonte, código binário e executáveis, bem como suas dependências;
- *diagrama de instalação (desenvolvimento)*, o qual descreve a configuração em tempo de execução dos nós de processamento e seus respectivos componentes.

Diagramas comportamentais são utilizados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema. Cinco são os diagramas comportamentais definidos pela Linguagem de Modelagem Unificada (UML):

- *diagrama de casos de uso*, o qual define os atores, os casos de uso do sistema e os seus relacionamentos. Um caso de uso é uma unidade de funcionalidade que representa uma sequência de ações que o sistema executa e que produz um resultado observável para os atores desse sistema;
- *diagrama de estados*, o qual descreve o comportamento de um sistema através da representação de um conjunto de estados e transições, conectando estes estados;

- *diagrama de atividades*, o qual é uma variação do diagrama de estados e estes correspondem a atividades representando a execução de operações, sendo que as transições são disparadas pelo término dessas operações;
- *diagrama de seqüência*, o qual descreve uma interação através das mensagens trocadas pelos objetos envolvidos nesta interação, organizadas em uma seqüência temporal;
- *diagrama de colaboração*, o qual descreve uma interação através de objetos, suas conexões e fluxos de mensagens associadas às conexões.

Além dos diagramas, UML também faz uso de pacotes (packages). Um pacote é um mecanismo que pode agrupar classes, componentes, seqüências ou outros pacotes.

### **3.3. Metodologia para o Desenvolvimento de Software baseado em Componentes**

Os componentes desenvolvidos neste projeto são modelados utilizando uma metodologia para desenvolvimento de software baseado em componentes. Esta metodologia, a qual é apresentada nesta seção, foi proposta por Cléver Ricardo Guareis de Farias et. al. [7].

A metodologia baseia-se na utilização de UML para a modelagem de uma aplicação de acordo com quatro níveis de abstração: níveis de empresa, de sistema, de componente e de objeto. Os níveis de componente e de objeto compõem a estrutura interna do sistema.

O *nível de empresa* ou de negócio tem por objetivo capturar o vocabulário e outras informações sobre o ambiente do sistema a ser desenvolvido, fornecendo uma vasta descrição abstrata deste. Este nível define a finalidade, o alvo, os atores (pessoas envolvidas), as atividades, as regras, as políticas, os serviços de suporte e outras características desse sistema. A informação obtida neste nível é usada para a comunicação com os seus usuários e para servir como base para delimitar o ambiente do sistema. Sendo estas informações específicas do ambiente organizacional, elas são comuns a várias aplicações.

O *nível de sistema* define o limite entre o sistema e o seu ambiente, e esta separação é conseguida através da obtenção dos requisitos do sistema. Serviços externos que lhe dão apoio também são identificados neste nível.

O *nível de componente* representa o sistema a ser desenvolvido em termos de um conjunto de componentes combinados e suas interfaces.

O *nível de objeto* corresponde à estrutura interna dos componentes. Este nível representa um componente em termos de um conjunto de objetos relacionados e corresponde ao tradicional desenvolvimento de software orientado a objeto. Um componente não precisa ser, necessariamente, implementado utilizando-se uma tecnologia orientada a objeto, mas essa tecnologia é reconhecida como a forma mais conveniente para sua implementação.

A figura 3.1. descreve a estrutura em camadas do processo de desenvolvimento de software.

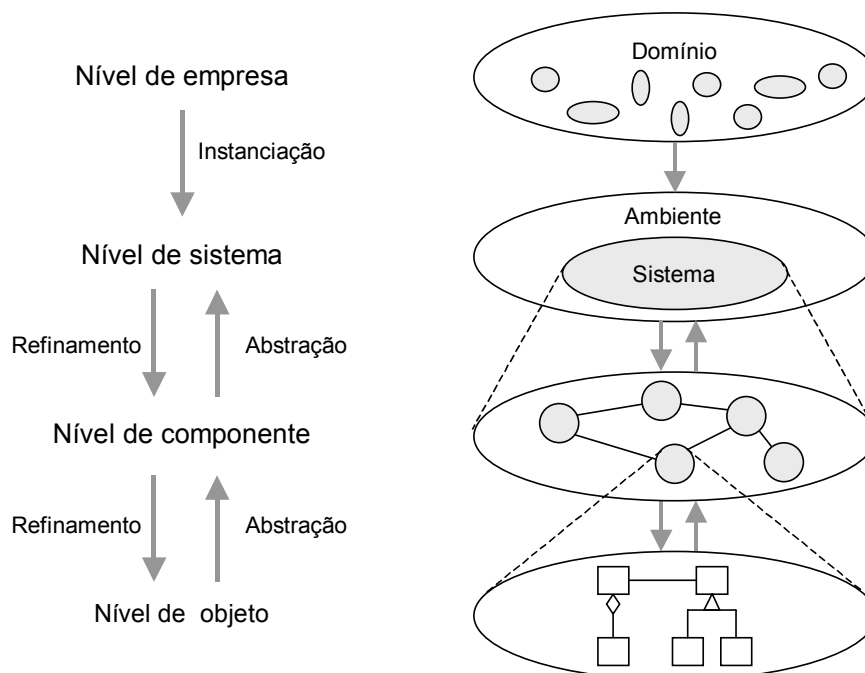


Figura 3.1. Níveis de Abstração

Os quatro níveis de abstração se relacionam, entre si, de diferentes formas. O nível de sistema corresponde a um sub-conjunto dos conceitos apresentados no nível de empresa. Portanto, diferentes sistemas podem ser produzidos a partir de um mesmo conjunto de

conceitos definidos no nível de empresa. O nível de componente é um refinamento do nível de sistema, e o nível de objeto é um refinamento do nível de componente. Isto quer dizer que o sistema pode ser decomposto em componentes e os componentes, por sua vez, em objetos. Inversamente, pode-se abstrair de um conjunto de objetos para formar um componente e de um conjunto de componentes para formar um sistema. Contudo, não é sempre possível abstrair-se do sistema e chegar à descrição completa das informações presentes no nível de empresa, uma vez que os conceitos presentes no nível de sistema podem corresponder a um sub-conjunto daquele.

Em cada nível de abstração, três visões são consideradas pela metodologia: visão estrutural, visão comportamental e visão interacional. A visão estrutural provê informações sobre a estrutura estática das entidades que compõem cada nível de abstração. A visão comportamental provê informações sobre o comportamento de cada entidade isoladamente, enquanto a visão interacional provê informações sobre o comportamento conjunto das entidades ao interagirem umas com as outras. As visões comportamental e interacional podem ser vistas como duas faces de um mesmo aspecto: comportamento. A figura 3.2. mostra como as diferentes visões se apresentam nos níveis de abstração. Devido ao fato do nível de empresa ser, antes de mais nada, um nível conceitual, não há uma clara divisão entre as visões, sendo demonstrado como uma única representação ao longo das diferentes visões no nível.

|                     | Visão Estrutural | Visão Comportamental | Visão Interacional |
|---------------------|------------------|----------------------|--------------------|
| Nível de Empresa    |                  |                      |                    |
| Nível de Sistema    |                  |                      |                    |
| Nível de Componente |                  |                      |                    |
| Nível de Objeto     |                  |                      |                    |

Figura 3.2. O Uso das visões em um processo de desenvolvimento



---

No *nível de empresa* a metodologia descreve diferentes técnicas para obtenção da informação, tais como o glossário de termos e o diagrama de conceitos. O uso de um glossário visa a manter uma documentação dos termos encontrados no domínio do sistema. O uso deste tipo de documentação é comum em engenharia de software e muitas vezes aparece com os nomes de dicionário de dados e dicionário de modelos. Este glossário deve ser mantido e atualizado enquanto o desenvolvimento do sistema continuar. Conseqüentemente, o nível de abstração no qual o termo foi definido também deve ser mencionado, pois o termo pode ser de diferentes tipos, à medida que o sistema se desenvolve. Um diagrama de conceitos consiste em um diagrama de classes UML onde as classes representam os conceitos e as associações entre as classes representam os relacionamentos entre os conceitos. Diagrama de objetos UML pode complementar o uso do diagrama de classes, representando uma possível instanciação do conceito capturado. O diagrama de conceitos é apresentado na seção 3.4.

No *nível de sistema* as diferenças entre as três visões tornam-se claras, passando a ter um importante papel no desenvolvimento do processo. A visão estrutural de uma aplicação cooperativa no nível de sistema é obtida principalmente através dos diagramas UML de casos de uso e de pacotes (componente ou classe). Diagramas casos de uso estão voltados para os requisitos do sistema, enquanto os diagramas de pacotes referem-se ao relacionamento entre o sistema e serviços/sistemas externos. Os conceitos de ator e atividade no nível de empresa são diretamente mapeados para os conceitos de Ator e Caso de Uso, respectivamente, no diagrama de casos de uso. Os relacionamentos estáticos entre um serviço externo que apóia uma atividade e o próprio sistema podem ser representados pela presença de um ator como uma entidade externa associada com um caso de uso, em um diagrama de caso de uso. Uma descrição textual sobre os casos de uso também deve ser apresentada. A visão comportamental deve prover informações sobre cada entidade isoladamente. Assim, diagramas de atividade ou statecharts UML devem ser elaborados para capturar as ações internas do sistema em resposta à invocação de uma operação; e diagramas de colaboração ou de seqüência devem ser elaborados para capturar as operações e notificações das entidades. A visão interacional de uma aplicação no nível de sistema captura as possíveis interações entre o sistema e seu ambiente, atores ou sistemas/serviços de apoio, usando diagramas de seqüência ou diagramas de colaboração UML. Uma descrição das mensagens trocadas contendo o

nome da mensagem, parâmetros, valor de retorno, ator associado e uma breve finalidade deve ser realizada.

A visão estrutural de uma aplicação no *nível de componente* pode ser representada usando-se o diagrama de pacotes, objetivando capturar o relacionamento estático e as dependências entre os componentes internos da aplicação e os sistemas externos. Um diagrama de instalação também pode ser usado para obter a distribuição física dos componentes nos nós do processo. Esta visão também compreende a representação das interfaces dos componentes. Uma interface do componente é uma coleção de operações que especifica o serviço provido por um componente. Esta interface pode ser representada por uma classe de interface, para mostrar suas operações, sem atributos. A visão comportamental de uma aplicação no nível de componente pode ser representada através do diagrama de atividades, enquanto a visão interacional pode ser representada pelos diagramas de seqüência e colaboração, visando a capturar as possíveis interações entre os componentes internos do sistema e entre estes componentes e sistemas externos.

No *nível de objeto*, a visão estrutural pode ser representada pelos diagramas de classe e de objetos; a visão comportamental, pelo diagrama de atividade ou statechart e a visão interacional pelo diagrama de colaboração ou seqüência.

A tabela 3.1. resume as várias técnicas utilizadas em cada nível e para cada uma das visões consideradas pela metodologia.

|                            | <b>Visão Estrutural</b>  | <b>Visão Comportamental</b>   | <b>Visão Interacional</b>  |
|----------------------------|--|---|--|
| <b>Nível de Empresa</b>    | <ul style="list-style-type: none"> <li>• glossário ou dicionário de dados</li> <li>• diagrama de conceitos, representado pelo diagrama de classes e sua instanciação pelo diagrama de objetos</li> </ul> |   |  |
| <b>Nível de Sistema</b>    | <ul style="list-style-type: none"> <li>• casos de uso</li> <li>• pacotes</li> </ul>  | <ul style="list-style-type: none"> <li>• seqüência ou colaboração</li> <li>• atividade ou statechart</li> </ul> | <ul style="list-style-type: none"> <li>• seqüência ou colaboração</li> </ul> |
| <b>Nível de Componente</b> | <ul style="list-style-type: none"> <li>• pacotes</li> <li>• instalação</li> <li>• classes de interface</li> </ul>  | <ul style="list-style-type: none"> <li>• atividades</li> </ul>  | <ul style="list-style-type: none"> <li>• seqüência ou colaboração</li> </ul> |
| <b>Nível de Objeto</b>     | <ul style="list-style-type: none"> <li>• classe</li> <li>• objeto</li> </ul>   | <ul style="list-style-type: none"> <li>• atividades ou statechart</li> </ul>                                    | <ul style="list-style-type: none"> <li>• seqüência ou colaboração</li> </ul> |

Tabela 3.1. Técnicas para modelagem

### **3.4. Modelo Conceitual para o Desenvolvimento de Sistemas CSCW**

Teorias e modelos cooperativos têm sido usados no desenvolvimento de sistemas CSCW. Estes modelos e teorias constituem um framework conceitual para que se possa obter e representar o conhecimento sobre o domínio da aplicação CSCW a ser investigada. Tal conhecimento representa os aspectos mais importantes referentes à atividade cooperativa a ser analisada. Alguns exemplos de frameworks conceituais utilizados no projeto de aplicações CSCW com ênfase em aspectos organizacionais e de coordenação da prática de trabalho são: Teoria da Coordenação [appud 6], Teoria da Atividade [appud 6], Gerenciamento de Tarefas [appud 6], Teoria da Ação/Interação [appud 6] e Suporte à Atividade Orientado a Objeto [appud 6]. A partir do estudo destes frameworks foi proposto um modelo conceitual genérico para projeto de sistemas CSCW, o qual é utilizado na aplicação da metodologia para desenvolvimento de software baseado em componentes, para identificar o vocabulário e informações sobre o sistema a ser desenvolvido e modelar o primeiro nível de abstração, o nível de empresa, desta metodologia.

O modelo conceitual é um framework baseado em quatro componentes: atividade, ator, informação e serviço e em um conjunto de relacionamentos entre estes conceitos. A figura 3.3. representa este modelo conceitual.

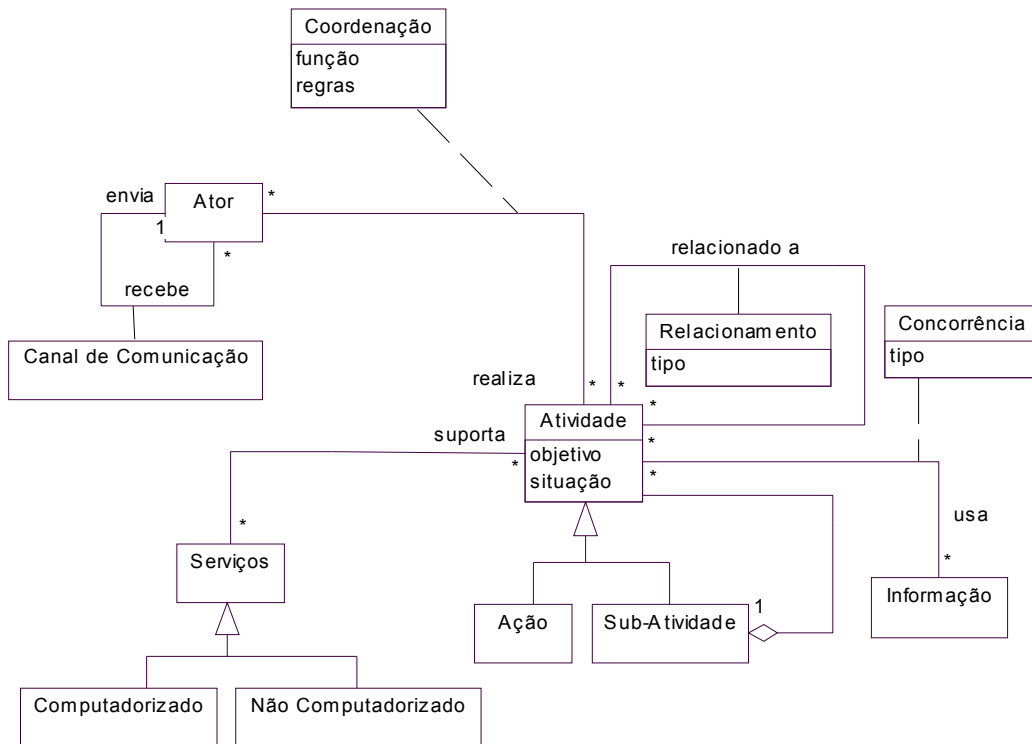


Figura 3.3. Modelo CSCW Genérico

Segundo o modelo conceitual, uma atividade representa um procedimento cooperativo e pode ser subdividida em subatividades e ações. As diferenças entre as subatividades e as ações são de dois tipos: (1) uma subatividade pode ser decomposta, enquanto uma ação é atômica; e (2) uma subatividade é considerada de vida longa, isto é, uma subatividade pode ser realizada em minutos, horas, dias e até meses, enquanto uma ação é imediata, sendo considerado somente o momento no qual a ação termina.

O modelo conceitual também informa que um relacionamento de associação pode ser estabelecido entre duas atividades, mas é restrito às pertencentes a um mesmo nível de agregação. Uma atividade também pode se relacionar com sua subatividade. Exemplos de relacionamentos são, dentre outros, a desativação de uma atividade por outra, a execução seqüencial de duas atividades e a execução sincronizada de duas atividades. Normalmente as atividades são executadas concorrentemente, a menos que forem restringidas, por exemplo, por um relacionamento seqüencial.

---

Um ator é uma entidade ativa, que é responsável por realizar uma atividade. Os atores podem se comunicar com outros para realizar as atividades corretamente. Para que a comunicação possa acontecer, um canal deve ser estabelecido entre os atores responsáveis pela realização de uma atividade. O canal de comunicação pode representar um canal de áudio, como uma linha telefônica, um canal de vídeo conferência, uma ferramenta de correio eletrônico ou até um canal ao vivo, no qual os atores são colocados em uma mesma sala. Desta forma, mensagens de comunicação são trocadas entre dois ou mais atores, quando um ator desempenha um papel de emissor e, possivelmente, vários atores desempenham papel de receptor.

Um ator não é parte de uma atividade, mas está associado a uma ou mais atividades. Uma função e um conjunto de regras coordenadas são os atributos da classe coordenação, estabelecida por um relacionamento de associação entre trabalhadores e atividades. A função é usada para descrever o papel de um ator em uma associação. Desta forma, um ator envolvido em várias atividades pode desempenhar diferentes papéis em cada uma delas. As regras coordenadas, tais como as políticas e mecanismos de controle básicos, regulam os relacionamentos entre diferentes atores realizando a mesma atividade.

O modelo conceitual também define um componente chamado informação. Uma informação representa qualquer tipo de dado eletrônico usado por uma atividade, tais como mensagens, documentos ou registros de banco de dados. As informações podem ser produzidas ou consumidas por uma atividade. Normalmente, uma mesma informação é compartilhada por várias atividades. Para permitir acessos simultâneos a uma informação compartilhada, o modelo conceitual identifica duas alternativas: (1) mecanismos de bloqueio na própria informação ou (2) mecanismos de controle da concorrência para monitorar o acesso à informação. Mecanismos de controle de concorrência são providos através da classe concorrência, estabelecida pela associação entre a atividade e a informação compartilhada.

Finalmente, o conceito de serviço representa qualquer tipo de serviço, computadorizado ou não, que dê suporte à execução de uma atividade. Um serviço compreende ferramentas tais como sistema de groupware específico como um editor para co-autoria ou um sistema de vídeo conferência; sistema não groupware como um editor de texto ou

---

sistema de gerenciamento de banco de dados, ou um serviço executado por uma secretária enquanto escrevendo uma carta.

A representação estrutural deste modelo conceitual está baseada em diagramas de classe e de objeto UML (Unified Modeling Language – Linguagem de Modelagem Unificada) e é aplicada neste trabalho de mestrado para modelar a aplicação de conferência multimídia.

### **3.5. Plataforma de Desenvolvimento**

Para implementação dos componentes desenvolvidos por este projeto será utilizada a linguagem Java, por ter esta a vantagem de ser utilizada em uma multiplicidade de plataformas operacionais. Para facilitar o desenvolvimento, outras facilidades oferecidas pela Sun Microsystems serão utilizadas, a saber: Enterprise Java Beans (EJB) [21], Java 2 Enterprise Edition (J2EE ) [22] e Java Media Framework (JMF) [23]. Estas facilidades são apresentadas a seguir.

#### **3.5.1. Enterprise Java Beans**

Enterprise Java Beans (EJB) [21] é uma arquitetura para computação distribuída baseada em componentes. EJB é uma especificação, e não um produto, que define uma arquitetura de componentes e as interfaces entre estes e um servidor EJB.

Enterprise beans são componentes de aplicações empresariais distribuídas, orientados à transação e funcionam como um servidor de camada intermediária em um sistema cliente/servidor de três camadas, escondendo a complexidade das aplicações distribuídas. As principais características de um enterprise bean são:

- um enterprise bean tipicamente contém a lógica empresarial da aplicação;
- as instâncias de um enterprise bean são criadas e gerenciadas em tempo de execução por um container; o acesso dos clientes é gerenciado através deste container;
- vários serviços, tais como transação e segurança, são providos pela plataforma, independentemente de um enterprise bean;

- um enterprise bean pode ser incluído em uma aplicação sem que sejam necessárias mudanças em seu código fonte ou mesmo sua recompilação.

EJB é a arquitetura de componentes utilizada na plataforma Java 2, Enterprise Edition (J2EE) [22], conforme representado na figura 3.4.

### 3.5.2. Java 2 Enterprise Edition

A plataforma Java 2 Enterprise Edition (J2EE) [22] define um padrão para desenvolvimento de aplicações comerciais multi-camadas. É uma arquitetura baseada em componentes para a criação, desenvolvimento e execução de aplicações no nível empresarial, orientada a objeto, em um ambiente distribuído. A figura 3.4 ilustra a arquitetura da plataforma J2EE.

J2EE simplifica o desenvolvimento das aplicações empresariais tratando-as como componentes padronizados e modulares, oferecendo um conjunto de serviços para estes componentes e definindo muitos detalhes do comportamento da aplicação automaticamente, sem a necessidade de uma programação complexa. Isto é, as camadas da funcionalidade são encapsuladas em tipos específicos de componentes: a lógica empresarial é encapsulada em componentes Enterprise Java Beans (EJB), as interações com o cliente podem ser apresentadas através de páginas web HTML (Hyper Text Markup Language) e a comunicação entre os componentes acontece transparentemente através do uso de vários padrões, tais como HTML, XML (Extensible Markup Language), http (HyperText Transfer Protocol), SSL (Secure Sockets Layer), RMI (Remote Method Invocation) e IIOP (Internet Inter-Orb Protocol), dentre outros.

O modelo J2EE divide as aplicações empresariais em três partes principais: componentes, containers e conectores. Componentes são de grande interesse para os desenvolvedores de aplicações, enquanto os containers e conectores são implementados pelos fornecedores de tecnologia para esconder a complexidade e promover a portabilidade. Os containers agem entre os clientes e os componentes, provendo serviços transparentemente para ambos, incluindo suporte à transação. Os conectores localizam-se abaixo da plataforma J2EE, definindo uma API (Application Programming Interface – Interface para Programação de Aplicação) que permite utilizar os vários

serviços oferecidos por diversos fornecedores, promovendo a flexibilidade por possibilitar a implementação de uma variedade de serviços específicos.

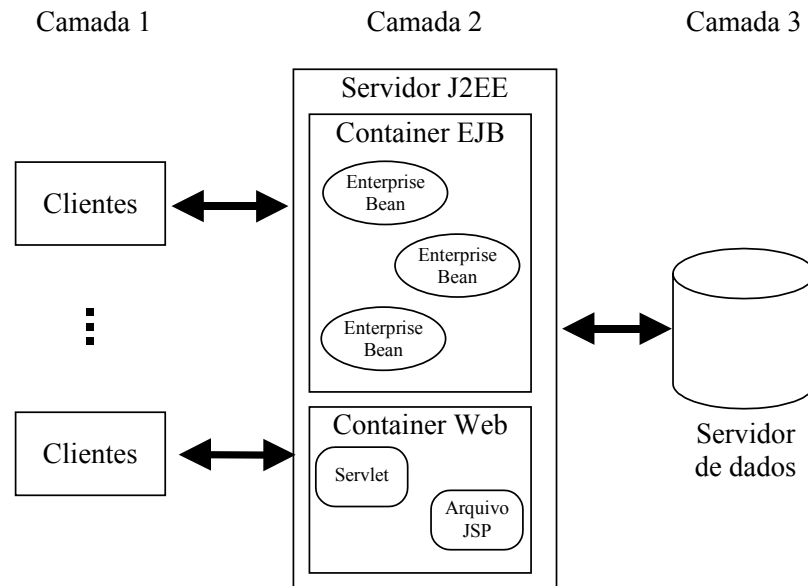


Figura 3.4. Arquitetura da plataforma J2EE.

Esta arquitetura está dividida de acordo com três camadas: cliente (camada 1), servidor (camada 2) e banco de dados (camada 3). O servidor J2EE oferece serviços, tais como *naming* e *directory*, os quais permitem localizar serviços e componentes através da interface de programação de aplicações *Java Naming e Directory Interface™ (JNDI)*; autenticação, que garante a segurança da aplicação por exigir identificação dos clientes, e serviços para a chamada dos métodos de um enterprise bean.

O container é um ambiente de execução que controla o enterprise bean e fornece um conjunto de serviços de sistema para este bean. Instâncias de um enterprise bean são executadas dentro de um container EJB, cujos serviços oferecidos incluem o gerenciamento de transações, segurança, a conexão com clientes remotos, o gerenciamento do ciclo de vida de um componente bean, a conexão com bases de dados, entre outros.

### 3.5.3. Java Media Framework

Java Mídia Framework (JMF) [23] é uma API (Application Programming Interface – Interface para Programação de Aplicação) para incorporar mídia temporal em applets e aplicações Java. JMF provê uma arquitetura unificada e interface de programação para gerenciar a captura, processamento, sincronização e transmissão de dados de mídia



temporal (áudio e vídeo) através da tecnologia Java, possibilitando, portanto, sua utilização em uma grande variedade de plataformas. JMF deve suportar a maioria dos tipos de mídia padrão, tais como AIFF (Audio Interchange File Format), AVI (Audio Vídeo Interleaved), JPG (Joint Photographic Experts Groups), MIDI (Musical Instrument Digital Interface), MPEG (Motion Picture Expert Group), PCM (Pulse Code Modulation), QuickTime, RMF (Rich Music Format), WAV (Wave), etc. A figura 3.5 ilustra a arquitetura em alto nível de JMF.

A API JMF consiste, principalmente, de interfaces que definem o comportamento e a interação dos objetos e são usadas para capturar, processar e apresentar a mídia temporal. Implementações destas interfaces operam em uma estrutura de framework. São características desta API:

- permitir o desenvolvimento de programas em Java que utilizam mídia temporal;
- oferecer suporte para capturar, armazenar e transmitir mídia;
- controlar o tipo de processamento a ser realizado durante a transmissão;
- realizar processamento personalizado sobre mídias contínuas;
- manter a flexibilidade e extensão necessárias para suportar aplicações de mídia avançadas e futuras tecnologias de mídia, pois define uma arquitetura de plug-in que permite acesso direto aos dados de mídia.

JMF provê uma API Java para acessar frameworks de mídia básicos. Assim, a utilização desta API facilita a individualização e extensão das funcionalidades de JMF para suportar formatos e tipos adicionais, otimizar o tratamento aos formatos suportados e criar novos mecanismos de apresentação.

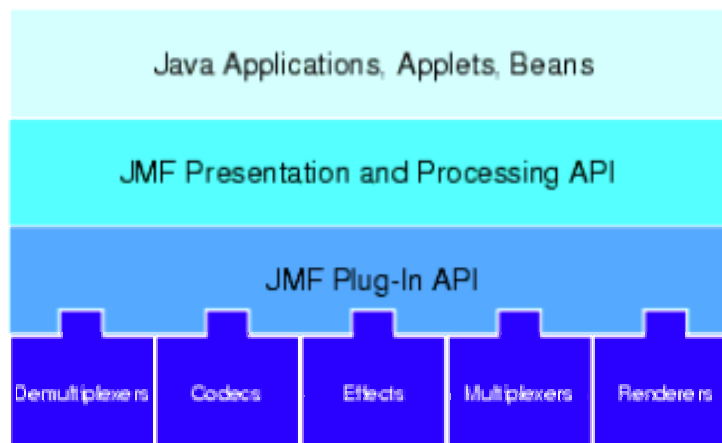


Figura 3.5. Arquitetura JMF de alto nível

---

É através da utilização da metodologia para desenvolvimento de software [6, 7], apresentada neste capítulo, que se pretende identificar e desenvolver os componentes para o sistema de conferência multimídia, de forma que estes componentes apresentem como características fundamentais a generalidade, o reuso e a flexibilidade. Além disso, com a utilização de Java como a linguagem para desenvolvimento, Enterprise Java Beans [21] como a arquitetura para computação distribuída baseada em componentes, Java 2 Enterprise Edition [22] como uma plataforma para desenvolvimento de aplicações comerciais multi-camadas e Java Media Framework [23] como uma API que permite a utilização de mídia temporal em aplicações Java, espera-se que a implementação dos componentes identificados por este trabalho possa ter sua complexidade e seu tempo de desenvolvimento reduzidos.

De acordo com a metodologia para desenvolvimento de software baseado em componentes, cada domínio das aplicações CSCW deve ser estudado a fim de se conhecer o vocabulário dessas aplicações – nível de empresa – prosseguindo, para cada aplicação, a modelagem nos demais níveis – de sistema, de componentes e de objetos, a fim de se identificar componentes genéricos para aplicações cooperativas. Por ser o escopo de CSCW tão amplo, seria um trabalho exaustivo analisar cada uma das aplicações cooperativas e modelá-las em todos os níveis de abstração propostos pela metodologia para desenvolvimento de software baseado em componentes. Assim, uma variedade de aplicações CSCW foi analisada e somente uma foi escolhida para ser detalhada: um sistema de conferência multimídia. Após identificados os componentes da aplicação cooperativa específica, uma tarefa de abstração é realizada para se determinar quais destes são comuns a várias aplicações cooperativas. Estes componentes compõem o toolkit para aplicações cooperativas resultante deste trabalho e são genéricos e flexíveis, permitindo sua reutilização em várias aplicações CSCW.

O próximo capítulo apresenta a utilização da metodologia para desenvolvimento de software baseado em componentes [6, 7] em um estudo de caso envolvendo o sistema de conferência multimídia. Primeiramente, uma descrição informal da aplicação cooperativa será apresentada e, num segundo momento, serão apresentadas partes da modelagem dessa aplicação conforme os níveis de abstração propostos pela metodologia.

## Capítulo IV

### ***Aplicação da Metodologia – Sistema de Conferência Multimídia***

Este capítulo descreve como aplicar a metodologia para desenvolvimento de software baseado em componentes [6, 7] para identificação e desenvolvimento de componentes. É apresentada uma descrição sobre a aplicação a ser modelada e partes da modelagem conforme os níveis de abstração propostos pela metodologia.

#### **4.1. Descrição Informal da Aplicação a ser Modelada**

As aplicações cooperativas procuram suprir as reais necessidades de um trabalho em grupo, transformando as possíveis interações entre os indivíduos amplamente auxiliadas por computador. Conseqüentemente, aspectos multimídia devem estar presentes nestas aplicações. Deste modo, nesta seção é apresentado um sistema para conferência multimídia (SCM), sobre o qual é aplicada a metodologia para a modelagem e identificação dos componentes a serem desenvolvidos. Este sistema permite que sessões sejam criadas possibilitando a comunicação e a cooperação através de uma combinação de áudio, vídeo e texto. Além disso, deve prover mecanismos para permitir o convite e a inscrição de participantes, para que participantes atrasados tenham acesso à sessão, para torná-la persistente e para efetuar o controle de frequência dos participantes.

O sistema de conferência multimídia proposto está baseado em quatro tarefas básicas: acessar o sistema, disponibilizar conferência, controlar os participantes da sessão e participar da conferência. Estas tarefas representam as funcionalidades do sistema e serão apresentadas a seguir. Para compreensão deste sistema, três conceitos são fundamentais:

- sessão: espaço de tempo durante o qual durará a atividade de um grupo de pessoas. Uma reunião, uma conferência, uma aula ou até mesmo um possível encontro entre pessoas com o objetivo informal de manter uma comunicação caracteriza uma sessão, pois possui um intervalo de tempo para sua realização e permite a comunicação entre as pessoas que fazem parte da sessão.
- participante: pessoa que manifesta interesse em fazer parte, ou que já participe, de alguma sessão.

- início da cooperação ou estabelecimento de uma sessão: é o ato de um participante se conectar a uma sessão, desde que faça parte dela. Quando duas ou mais pessoas, localizadas em um mesmo local ou distantes geograficamente, se reúnem com a finalidade de realizar uma comunicação, utilizando equipamentos e ferramentas baseadas em computador, transmitindo e recebendo todo e qualquer tipo de informação, elas estão cooperando ou estabelecendo uma sessão.

A funcionalidade de acesso está representada pelas operações que permitem a utilização do sistema pelo participante. Essas operações são: o registro, o login, a remoção de um registro e o logoff. O *registro* permite que o usuário registre-se no sistema, informando um username, uma password, nome completo e email. O *login* possibilita que o participante se conecte ao sistema através da informação de seu username e password, previamente registrados. A *remoção de registro* consiste no cancelamento de um registro, impedindo o acesso do participante ao sistema. E, finalmente, o *logoff* consiste em permitir que o participante se desconecte do sistema.

A funcionalidade de tornar disponível uma conferência é formada pelas operações: criar sessão, atualizar sessão, transferir direitos de controlador e oferecer suporte para votação. *Criar uma sessão* consiste em dar origem a uma sessão através da informação do seu nome, sua finalidade, das mídias que poderão ser utilizadas e do horário para início da cooperação. Após criar uma sessão o participante torna-se o controlador desta. *Atualizar sessão* é uma atividade que somente pode ser executada pelos controladores das sessões e consiste na atualização do nome e/ou finalidade da sessão; na inclusão/alteração/remoção dos horários (dia e hora) em que ocorrerá a comunicação entre os membros da sessão; na inclusão/exclusão das mídias que estarão disponíveis para serem utilizadas pelos participantes de uma sessão ou no bloqueio (encerramento ou remoção) da sessão. O encerramento representa a desativação desta. *Transferir direitos de controlador* consiste em permitir que outro participante seja o responsável pela sessão e esta atividade somente pode ser executada pelo controlador desta. O *suporte à votação* deve permitir que perguntas sejam elaboradas para serem posteriormente respondidas pelos participantes de uma sessão quando estes iniciarem uma cooperação.

---

A funcionalidade de participação compreende as atividades necessárias para gerenciar o direito de acesso dos participantes ao sistema. Essas atividades estão divididas em três grupos, a saber: atividades a serem realizadas pelos participantes/controladores de uma sessão e pelos participantes que não fazem parte de uma sessão. As atividades a serem realizadas por aqueles que fazem parte de uma sessão são divididas em genéricas, para qualquer participante, e específicas, somente para o controlador.

As atividades genéricas são: o convite e o pedido de remoção da participação em uma sessão. O *convite* consiste em enviar mensagens a quaisquer pessoas, registradas ou não no sistema, convidando-as para participarem de uma sessão. Os convidados devem, posteriormente, aceitar ou recusar o convite. Uma mensagem deve ser enviada ao convidado informando-o sobre o convite. O pedido de *remoção da participação em uma sessão* deve ser feito pelo próprio participante desde que não esteja em cooperação. Uma mensagem deve ser enviada ao controlador da sessão informando-o sobre a remoção da participação.

*Analisar inscrições e visualizar inscrições pendentes* são atividades específicas que consistem em aceitar o solicitante como um membro da sessão ou rejeitar sua inscrição e apresentar ao controlador quais inscrições ainda devem ser analisadas, respectivamente. Tanto na aceitação quanto na recusa, uma mensagem deve ser enviada ao solicitante informando-o sobre o parecer do controlador da sessão.

As atividades a serem realizadas por aqueles que não fazem parte de uma sessão são: *solicitar inscrição na sessão, visualizar convites recebidos, confirmar e recusar convite*. As atividades de confirmar ou recusar convite também devem se encarregar do envio de mensagens sobre a intenção do participante.

A tabela 4.1. apresenta as atividades agrupadas sob a funcionalidade de participação.

| Para aqueles que fazem parte da sessão           |  | Para aqueles que NÃO fazem parte da sessão   |
|--|--|--|
| <b>Genéricas (Participante e/ou Controlador)</b> | <ul style="list-style-type: none"> <li>• Convidar</li> <li>• Remover Participação em uma sessão</li> </ul>         | <ul style="list-style-type: none"> <li>• Solicitar Inscrição</li> <li>• Confirmar Convite</li> <li>• Recusar Convite</li> <li>• Visualizar Convites Recebidos</li> </ul> |
| <b>Específicas (controlador)</b>                 | <ul style="list-style-type: none"> <li>• Analisar Inscrições</li> <li>• Visualizar Inscrições Pendentes</li> </ul> |  |

Tabela 4.1. Atividades Oferecidas pelo SCM sob a Funcionalidade de Participação

A funcionalidade de cooperação refere-se a todas as operações oferecidas aos participantes do sistema quando estes desejarem iniciar a cooperação. Aqui os serviços também podem ser divididos em grupos, a saber: o grupo dos serviços específicos, oferecidos somente aos controladores e o grupo dos serviços genéricos, oferecidos aos participantes e controladores.

Como serviços genéricos oferecidos durante uma cooperação estão *estabelecer sessão, votar, selecionar mídia a ser utilizada durante a cooperação, abandonar cooperação, deixar cooperação, transmitir/receber áudio, transmitir/receber vídeo, gravar trechos da conferência, transferir arquivos, realizar anotações, trocar mensagens síncronas, visualizar os participantes durante uma cooperação, visualizar quais sessões estão em cooperação e visualizar participantes transmissores*; e como serviço oferecido exclusivamente aos controladores encontra-se *terminar a cooperação*. A frequência do participante e quaisquer informações sobre a sessão, como horário de início e término, devem ser controladas pelas tarefas iniciar, abandonar, deixar e terminar cooperação. A tabela 4.2. apresenta os serviços agrupados sob a funcionalidade de cooperação.

| <b>Funcionalidade – Cooperação</b>   |   |
|--|---|
| <b>Genéricos</b>   | <b>Específicos</b>  |
| <ul style="list-style-type: none"> <li>• Estabelecer Sessão (ou iniciar cooperação)</li> <li>• Votar</li> <li>• Selecionar Mídia</li> <li>• Abandonar Cooperação</li> <li>• Deixar Cooperação</li> <li>• Transmitir/Receber Áudio</li> <li>• Transmitir/Receber Vídeo</li> <li>• Gravar Trechos da Conferência</li> <li>• Transferir Arquivos</li> <li>• Realizar Anotações</li> <li>• Trocar Mensagens Síncronas</li> <li>• Visualizar Participantes em Cooperação</li> <li>• Visualizar Sessões em Cooperação</li> <li>• Visualizar Participantes Transmissores</li> </ul> | <ul style="list-style-type: none"> <li>• Terminar Cooperação</li> </ul> |

Tabela 4.2. Serviços Oferecidos pelo SCM sob a Funcionalidade de Cooperação

O SCM também deve permitir que algumas operações de visualização da informação sejam realizadas tais como: *visualizar* quais são as *sessões controladas* por um certo participante, *visualizar* quais são as *sessões em que o participante pode cooperar*, *visualizar* quais são as *sessões em que o participante não pertence*, *visualizar* quais são as *sessões que estão em cooperação* sob a perspectiva de um participante, *visualizar* quem são os *participantes de uma determinada sessão*, *visualizar* quem são os *participantes em cooperação* e *visualizar* quem são os *participantes que estão transmitindo áudio e os que estão transmitindo vídeo*. A tabela 4.3. apresenta todas as funcionalidades e atividades oferecidas pelo SCM.

| <b>Acessar Sistema</b>   | <b>Disponibilizar Conferência</b>   | <b>Controlar Participantes</b>   | <b>Participar da Conferência</b>  | <b>Visualizar Informações</b>  |
|--|---|--|---|--|
| <ul style="list-style-type: none"> <li>• Registro</li> <li>• Login</li> <li>• Remoção de registro</li> <li>• Logoff</li> </ul> | <ul style="list-style-type: none"> <li>• Criar sessão</li> <li>• Atualizar sessão</li> <li>• Transferir direitos de controlador</li> <li>• Suporte à votação</li> </ul> | <ul style="list-style-type: none"> <li>• Para aqueles que fazem parte da sessão               <ul style="list-style-type: none"> <li>- Genéricas                   <ul style="list-style-type: none"> <li>a) convidar</li> <li>b) remover participação em uma</li> </ul> </li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Genéricas               <ul style="list-style-type: none"> <li>- estabelecer sessão</li> <li>- votar</li> <li>- selecionar mídia</li> <li>- abandonar cooperação</li> <li>- deixar cooperação</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Sessões controladas por um participante</li> <li>• Sessões em que o participante pode cooperar</li> </ul> |

|  |  |   |  |   |
|--|--|---|--|---|
|  |  | <p>sessão</p> <ul style="list-style-type: none"> <li>- Específicas</li> <li>• Para aqueles que não fazem parte da sessão</li> </ul> | <ul style="list-style-type: none"> <li>- transmitir/receber áudio</li> <li>- transmitir/receber vídeo</li> <li>- gravar trechos da conferência</li> <li>- transferir arquivos</li> <li>- realizar anotações</li> <li>- trocar mensagens síncronas</li> <li>- visualizar participantes em cooperação</li> <li>- visualizar sessões em cooperação</li> <li>- visualizar participantes transmissores</li> <li>• Específicas <ul style="list-style-type: none"> <li>- Terminar cooperação</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Sessões em que o participante não pertence</li> <li>• Sessões em cooperação sob a perspectiva de um participante</li> <li>• Participantes de uma determinada sessão</li> <li>• Participantes em cooperação</li> <li>• Participantes transmitindo áudio ou vídeo</li> </ul> |
|--|--|---|--|---|

Tabela 4.3. Funcionalidades/Atividades Oferecidas pelo SCM

Este sistema de conferência multimídia deve prover mecanismos para controlar quais são as pessoas registradas no sistema, os participantes conectados ao sistema, os participantes por sessão, os controladores das sessões, as inscrições pendentes, os convidados, os participantes em cooperação, as sessões em cooperação, as mídias selecionadas por cada participante em cooperação, os transmissores e os receptores de áudio/vídeo. Neste sentido, todo o controle do sistema a ser modelado está baseado na manutenção de listas. Uma descrição mais detalhada da aplicação envolvendo suas funcionalidades e as listas a serem controladas facilitarão a compreensão do mecanismo de manutenção das informações e está apresentada no apêndice 1. As regras que devem ser obedecidas para garantir que os requisitos do sistema sejam atendidos também estão apresentadas neste apêndice.



Tratando-se de uma aplicação de conferência multimídia, o sistema de comunicação e administração de imagens pela rede deve ser definido. A comunicação entre dois dispositivos conectados na rede pode ser feita de três formas distintas:

- a) transmissão *unicast*, que representa a comunicação ponto-a-ponto;
- b) transmissão *broadcast*, que representa a comunicação de ponto-a-todos os pontos;
- c) transmissão *multicast*, que representa a comunicação ponto-a-multiponto.

A abordagem utilizada neste projeto é a transmissão *multicast*. Tal forma é utilizada quando um nó deseja enviar a mesma informação a vários nós destinos que formam um grupo. Estes nós são especificados por um único endereço, denominado endereço de grupo. IP *multicast* representa a transmissão de datagramas IP a um grupo composto de zero ou mais *hosts* identificado por um único endereço IP destino. Os membros do grupo podem ser dinâmicos, isto é, podem se juntar ou deixar os grupos a qualquer instante, não havendo limitação em relação à localização ou número de membros em um grupo. Os grupos também podem ser permanentes ou transitórios. Os grupos permanentes têm endereços IPs fixos estabelecidos pelo administrador da rede; enquanto os grupos transitórios recebem os seus endereços dinamicamente [ ].

Para resolver o problema de dinamismo dos membros do grupo, onde cada membro pode se juntar ou deixar o grupo a qualquer instante, a mídia deve ser transmitida sempre em duas situações:

- a) quando o participante iniciar a transmissão. Diante desta situação deve-se localizar todos os participantes em cooperação e iniciar a transmissão multicast para o endereço IP de cada um dos participantes localizados.
- b) quando o participante iniciar a cooperação. Diante desta situação deve-se localizar todos os participantes que estão em transmissão e iniciar a transmissão multicast dos transmissores localizados para o endereço IP do participantes que iniciou a cooperação.

Para a transmissão de áudio, o controle da palavra deve ser realizado através da obtenção de um sinal. Quando os participantes solicitam o controle da palavra um número seqüencial é atribuído à solicitação e o direito de transmitir o áudio é concedido assim que chegar a vez. O controle da vez é concedido por ordem de número de

solicitação através da manipulação de uma variável chamada de token – movimentação de um carácter “\*” para a variável, por exemplo para o participante que tem o controle da palavra. Assim que o participante desistir do controle da palavra sua solicitação é removida.

A partir da descrição informal apresentada nesta seção, a seção seguinte mostra a modelagem do sistema de conferência multimídia utilizando-se a metodologia para desenvolvimento de software baseado em componentes [6, 7] para os quatro níveis de abstração propostos.

#### **4.2. Modelagem – Nível de Empresa**

Este nível deve capturar o vocabulário e outras informações sobre o domínio da aplicação cooperativa a ser investigada. Os conceitos que devem ser identificados podem variar de acordo com o domínio da aplicação em desenvolvimento. Contudo, conceitos comuns à maioria das aplicações cooperativas são atividades, atores, regras (políticas), informação (recursos) e serviços de suporte [6].

Neste nível de abstração não há uma clara divisão entre as visões estrutural, comportamental e interacional, pois este é basicamente um nível conceitual. Assim, este nível é apresentado sob uma única perspectiva ao longo das três visões.

As técnicas utilizadas para representar o conhecimento sobre o domínio da aplicação cooperativa são diagrama de conceitos/objetos, diagrama de atividades e glossário de termos.

A figura 4.1. mostra o diagrama de objeto para a funcionalidade controlar participante da sessão. Esta funcionalidade corresponde a uma atividade do sistema e é composta de três outras atividades, a saber: inscrição, pelo participante, para fazer parte da sessão; convite ao participante, incluindo a atividade de análise (aceitação ou recusa) do convite e remoção da participação em uma sessão. Essas atividades são identificadas pela figura 4.1, assim como sua hierarquia, os atores que as executam, as informações produzidas ou consumidas e os serviços suportados pelas mesmas.

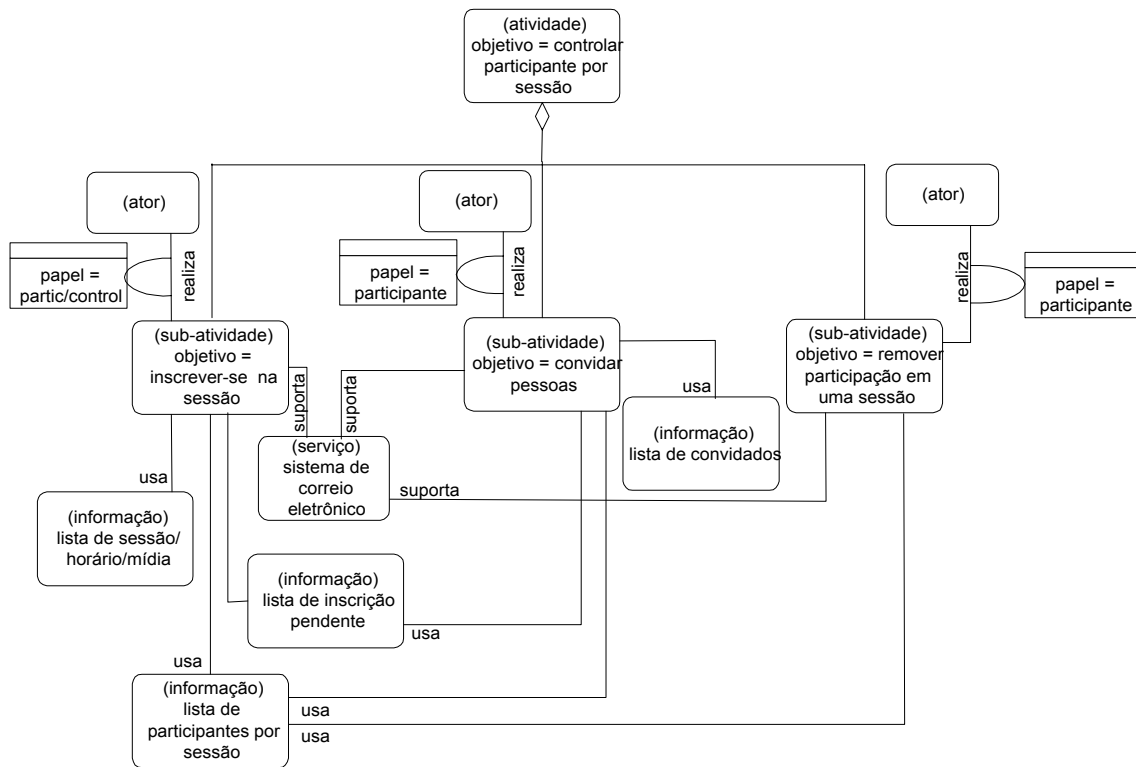


Figura 4.1. Nível de Empresa – Diagrama de Objeto – Atividade Controlar Participante por Sessão

A figura 4.2. apresenta os relacionamentos entre as atividades identificadas no diagrama de objeto apresentado pela figura 4.1. Observa-se que as atividades da funcionalidade controlar participante de uma sessão são independentes e, após a sua realização, uma atividade interna de envio de mensagem deve ser realizada para notificar os interessados sobre cada tarefa executada. A modelagem completa deste nível se encontra no apêndice 2.

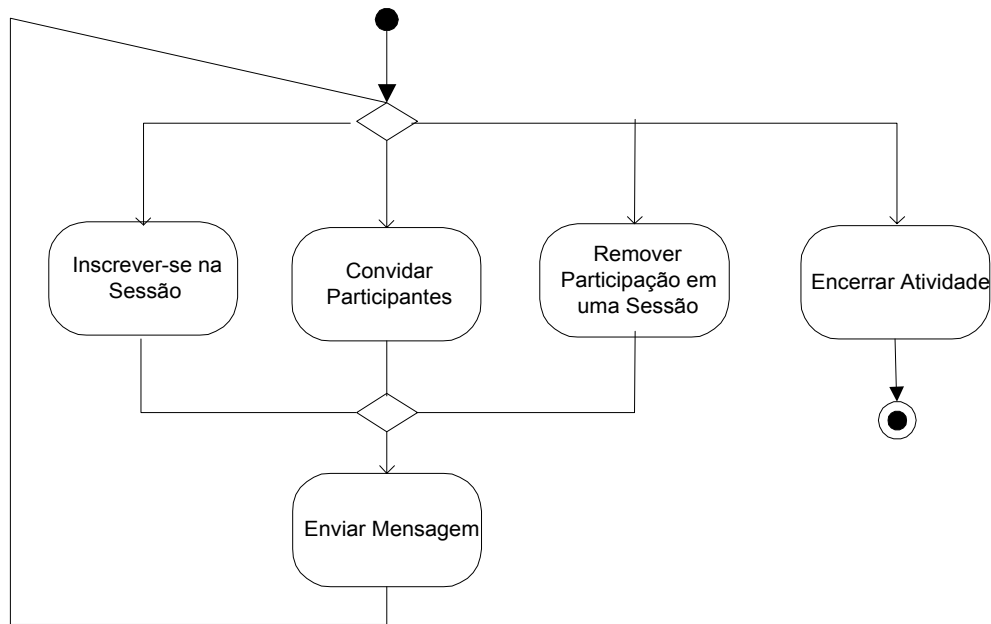


Figura 4.2. Nível de Empresa – Diagrama de Atividades – Atividade Controlar Participante por Sessão

### 4.3. Modelagem – Nível de Sistema

Este nível deve definir o limite entre o sistema e o seu ambiente através da identificação e descrição dos requisitos do sistema e dos serviços externos suportados pelo sistema. Neste nível as diferenças entre as três visões tornam-se aparentes e são discutidas separadamente.

A visão estrutural é capturada através da elaboração do diagrama de casos de uso de UML – para representar os requisitos do sistema – e através da elaboração do diagrama de pacotes UML (classes ou componentes) – para representar o relacionamento estático (dependências) entre o sistema e serviços/sistemas externos. A figura 4.3. apresenta parte do diagrama de casos de uso do sistema de conferência multimídia e refere-se à atividade controlar participante por sessão. Esta figura apresenta os agentes externos – participante, controlador e serviço de correio eletrônico –, as atividades realizadas por eles e o controle a ser realizado por cada atividade. Os relacionamentos entre os casos de uso são do tipo <<extends>> ou <<includes>>. Um relacionamento <<extends>> de um caso de uso A para um caso de uso B indica que uma instância do caso de uso B pode ser adicionada (sujeito a condições específicas especificadas na extensão) ao comportamento especificado por A. Um relacionamento <<includes>> de um caso de



A visão interacional está representada pela figura 4.4. através das mensagens trocadas entre o sistema de conferência multimídia e o seu ambiente quando o participante se conecta ao sistema, solicita inscrição em uma sessão e desconecta-se do sistema. Diante desta situação o participante envia ao SCM um pedido de registro e este lhe retorna uma resposta. O participante solicita ao SCM que lhe sejam mostradas as sessões que ele não pode cooperar e este lhe retorna uma lista de sessões. O participante escolhe uma sessão e envia um pedido de inscrição na sessão ao SCM. Este, por sua vez, retorna uma resposta ao participante e envia uma mensagem ao controlador da sessão informando-o sobre a inscrição que deve ser analisada, através da utilização de um serviço externo de correio eletrônico. Finalmente o participante envia ao SCM um pedido de desconexão do sistema. O apêndice 3 apresenta toda a modelagem do SCM para o nível de sistema.

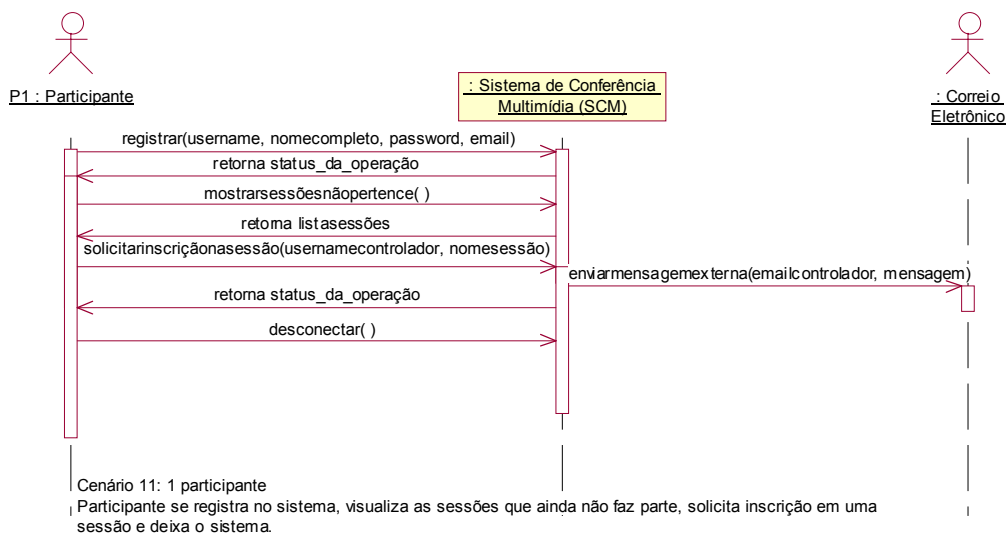


Figura 4.4. Nível de Sistema – Visão Interacional – Diagrama de Seqüência

#### 4.4. Modelagem – Nível de Componente

O nível de componente representa o sistema a ser desenvolvido em termos de um conjunto de componentes combinados e suas interfaces. Sendo a conferência multimídia uma aplicação distribuída, optou-se por modelá-la utilizando a arquitetura cliente-servidor. Dentre as vantagens que justificam esta escolha estão: facilidade para garantia da consistência dos dados devido ao controle centralizado, efetiva utilização dos recursos da máquina cliente e da máquina servidora, redução do problema de manutenção de código da aplicação nas estações clientes e aplicação cliente mais simples por evitar complexos algoritmos distribuídos, dentre outras.

Assim, a aplicação cooperativa deve ser analisada sob a perspectiva da arquitetura escolhida, ou seja, sob a perspectiva do cliente e do servidor. Num primeiro momento as tarefas realizadas para a modelagem no nível de sistema, para as três visões, devem ser repetidas neste nível a fim de identificar o limite do sistema para a aplicação cliente e para a aplicação servidora para que, posteriormente, possam ser identificados os componentes a serem desenvolvidos. O Apêndice 4 apresenta a modelagem do SCM sob a perspectiva do cliente e do servidor.

Os requisitos da aplicação cliente e servidora, capturados pelo diagrama de casos de uso, são analisados para que sejam identificados os componentes do sistema de conferência multimídia. Buscando alcançar as características desejáveis dos componentes tais como oferecimento de um serviço único, generalidade, interoperabilidade e flexibilidade, dentre outras, tentou-se identificar proximidades entre casos de uso e atribuir a implementação desses casos de uso para um mesmo componente. Neste sentido, observou-se que os serviços oferecidos tanto pela aplicação cliente quanto pela aplicação servidora são agrupados em serviços de gerenciamento de acesso, gerenciamento de sessão, gerenciamento de participação, gerenciamento de cooperação e visualização das informações. Esta análise possibilitou a identificação dos componentes da aplicação cliente e da aplicação servidora, os quais são chamados de componentes cliente e componentes servidor.

A tabela 4.4. apresenta os vários componentes identificados por este trabalho para o sistema de conferência multimídia. Alguns componentes servidor foram agrupados para formar componentes servidor compostos. Um componente composto é resultado da agregação de vários componentes tal que o comportamento externo do componente combinado corresponda a um único componente. Os componentes servidor compostos são o gerente de participação e o gerente de cooperação. O primeiro é formado pelos componentes gerente de inscrição, gerente de convite, gerente de correio assíncrono e gerente de associação; enquanto o segundo, pelos componentes gerente de comunicação, gerente de mídia, gerente de correio síncrono, gerente de recepção, gerente de transmissão, gerente de votação, gerente de anotações, gerente de transferência de arquivos e gerente de gravação da conferência.

Além dos componentes que representam a funcionalidade da conferência multimídia três outros componentes foram identificados. Dois pertencem à aplicação cliente e têm a finalidade de unir a interface da aplicação aos componentes cliente (Gerente de SCM) e unir a aplicação cliente à aplicação servidora (Gerente de SCM Cliente). O último pertence à aplicação servidora e deve unir os componentes servidor aos componentes cliente (Gerente de SCM Servidor).

| <b>Classificação</b>               | <b>Nome</b>                            | <b>Grau</b>                   | <b>Funcionalidade</b> |
|------------------------------------|--|-------------------------------|-----------------------|
| Cliente                            | Gerente de SCM                         | Simples                       |                       |
|                                    | Gerente de SCM Cliente                 | Simples                       |                       |
|                                    | Gerente de Acesso Cliente              | Simples                       | Acessar               |
|                                    | Gerente de Sessão Cliente              | Simples                       | Controlar Sessão      |
|                                    | Gerente de Participação Cliente        | Simples                       | Participar            |
|                                    | Gerente de Cooperação Cliente          | Simples                       | Cooperar              |
|                                    | Gerente de Visualização Cliente        | Simples                       | Visualizar            |
| Servidor                           | Gerente de SCM Servidor                | Simples                       |                       |
|                                    | Gerente de Acesso Servidor             | Simples                       | Acessar               |
|                                    | Gerente de Sessão Servidor             | Simples                       | Controlar Sessão      |
|                                    | Gerente de Participação Servidor       | Composto                      | Participar            |
|                                    | Gerente de Cooperação Servidor         | Composto                      | Cooperar              |
|                                    | Gerente de Inscrição Servidor          | Simples                       | Participar            |
|                                    | Gerente de Convite Servidor            | Simples                       | Participar            |
|                                    | Gerente de Correio Assíncrono Servidor | Simples                       | Participar            |
|                                    | Gerente de Associação Servidor         | Simples                       | Participar            |
|                                    | Gerente de Comunicação Servidor        | Simples                       | Cooperar              |
|                                    | Gerente de Mídia Servidor              | Simples                       | Cooperar              |
|                                    | Gerente de Correio Síncrono Servidor   | Simples                       | Cooperar              |
|                                    | Gerente de Recepção Servidor           | Simples                       | Cooperar              |
|                                    | Gerente de Transmissão Servidor        | Simples                       | Cooperar              |
|                                    | Gerente de Votação                     | Simples                       | Votar                 |
|                                    | Gerente de Anotações                   | Simples                       | Anotar                |
|                                    | Gerente de Transferência de Arquivos   | Simples                       | Transferir Arquivos   |
| Gerente de Gravação da Conferência | Simples                                | Gravar Trechos da Conferência |                       |
| Gerente de Visualização Servidor   | Simples                                | Visualizar                    |                       |

Tabela 4.4. Componentes Identificados para o Sistema de Conferência Multimídia (SCM)



A visão estrutural no nível de componentes deve capturar o relacionamento estático entre os componentes internos do sistema e entre estes componentes e sistemas externos, capturar a distribuição física dos componentes em nodos e apresentar as interfaces dos componentes. O primeiro pode ser representado pelo diagrama de pacotes UML (classes ou componentes), enquanto a distribuição física, pelo diagrama de desenvolvimento UML. A interface do componente é um conjunto de operações que especifica o serviço provido pelo componente e pode ser representada através do diagrama de classes UML. Os métodos da classe correspondem às operações da interface. A figura 4.5. mostra o diagrama de pacotes para os componentes cliente e a figura 4.6., para os componentes servidor; ambos representando o relacionamento estático entre os componentes identificados.

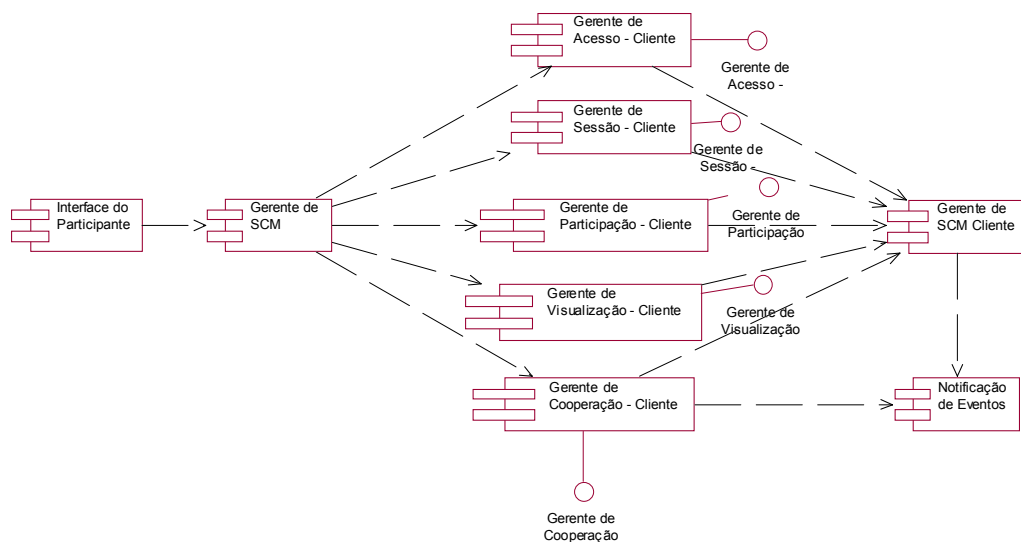


Figura 4.5. Nível de Componente – Visão Estrutural – Diagrama de Pacotes – Cliente

Os componentes se apresentam distribuídos fisicamente em duas estruturas – cliente e servidor–, em consequência da arquitetura utilizada. Neste sentido não é necessária a elaboração do diagrama de desenvolvimento para representar sua distribuição física.

A figura 4.7. apresenta as interfaces do componente Gerente de Acesso Cliente. Este componente possui operações que permitem realizar o serviço de acesso ao sistema provido pelo componente. São estas: registrar o participante, conectar o participante ao sistema, desconectá-lo do sistema e cancelar o registro de um participante.

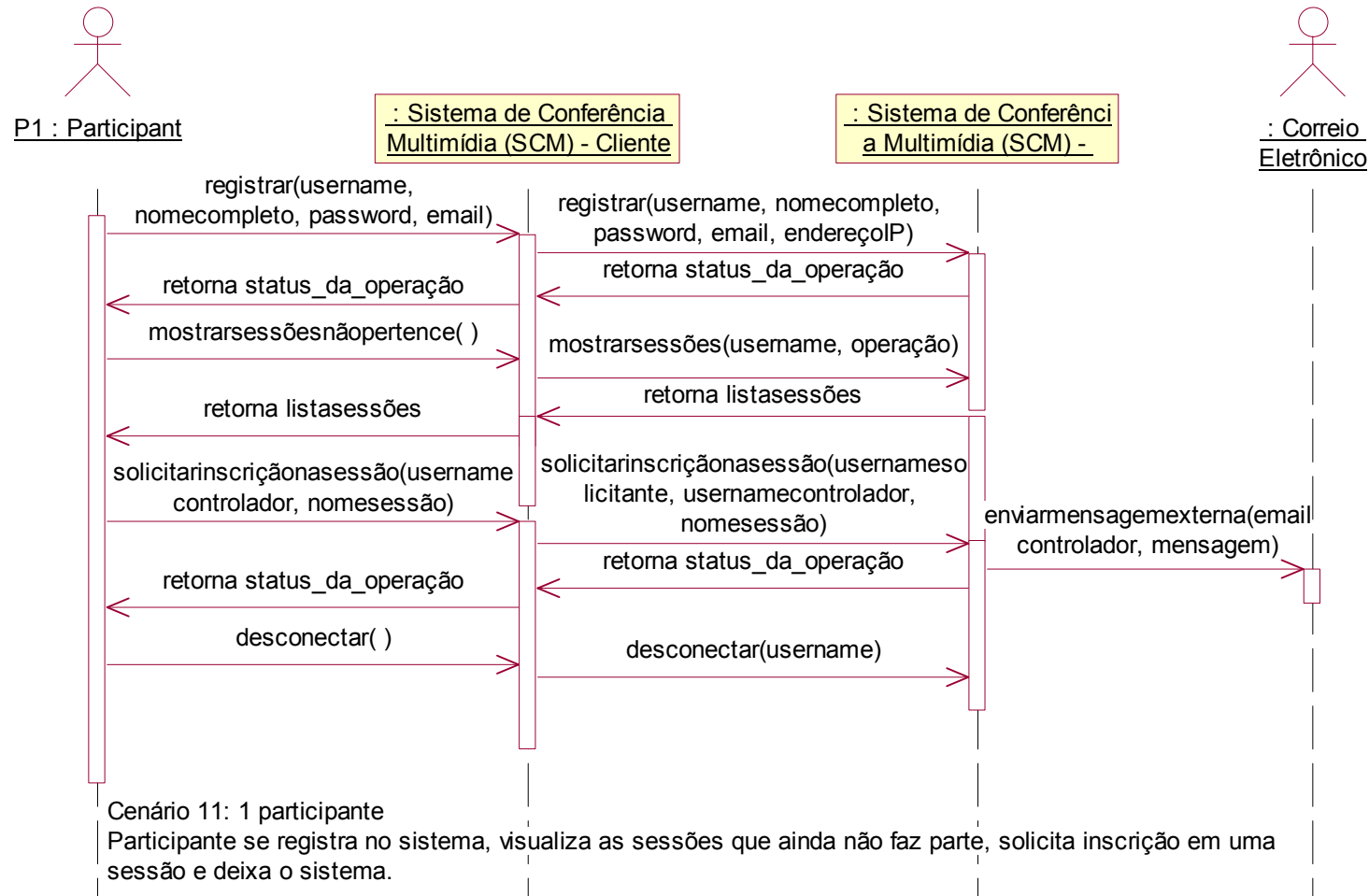


Figura 4.6. Nível de Componente – Visão Estrutural – Diagrama de Pacotes – Servidor

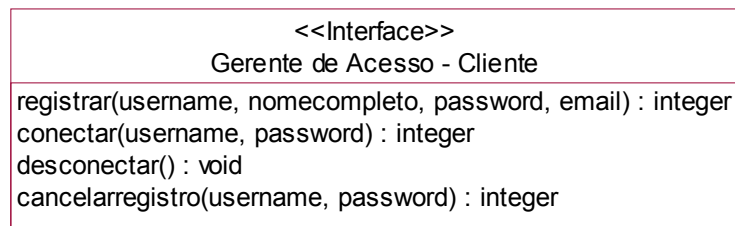


Figura 4.7. Nível de Componente – Visão Estrutural – Interface – Componente Gerente de Acesso Cliente

A visão comportamental de uma aplicação no nível de componente pode ser representada através do diagrama de atividades UML. A visão interacional da aplicação neste nível é capturada através do diagrama de colaboração ou seqüência UML. O uso destes diagramas – visão comportamental e interacional – tem o objetivo de modelar o comportamento do sistema frente à invocação de uma operação e de capturar as possíveis interações entre os componentes do sistema de conferência multimídia e entre estes componentes e sistemas/serviços externos, respectivamente. As figuras 4.8 e 4.9 mostram as mensagens trocadas entre o sistema de conferência multimídia cliente e servidor, respectivamente, e o seu ambiente quando o participante se conecta ao sistema, solicita inscrição em uma sessão e desconecta-se do sistema. A figura 4.10 apresenta o diagrama de atividade – servidor – quando uma requisição de solicitação de inscrição é enviada pelo participante.

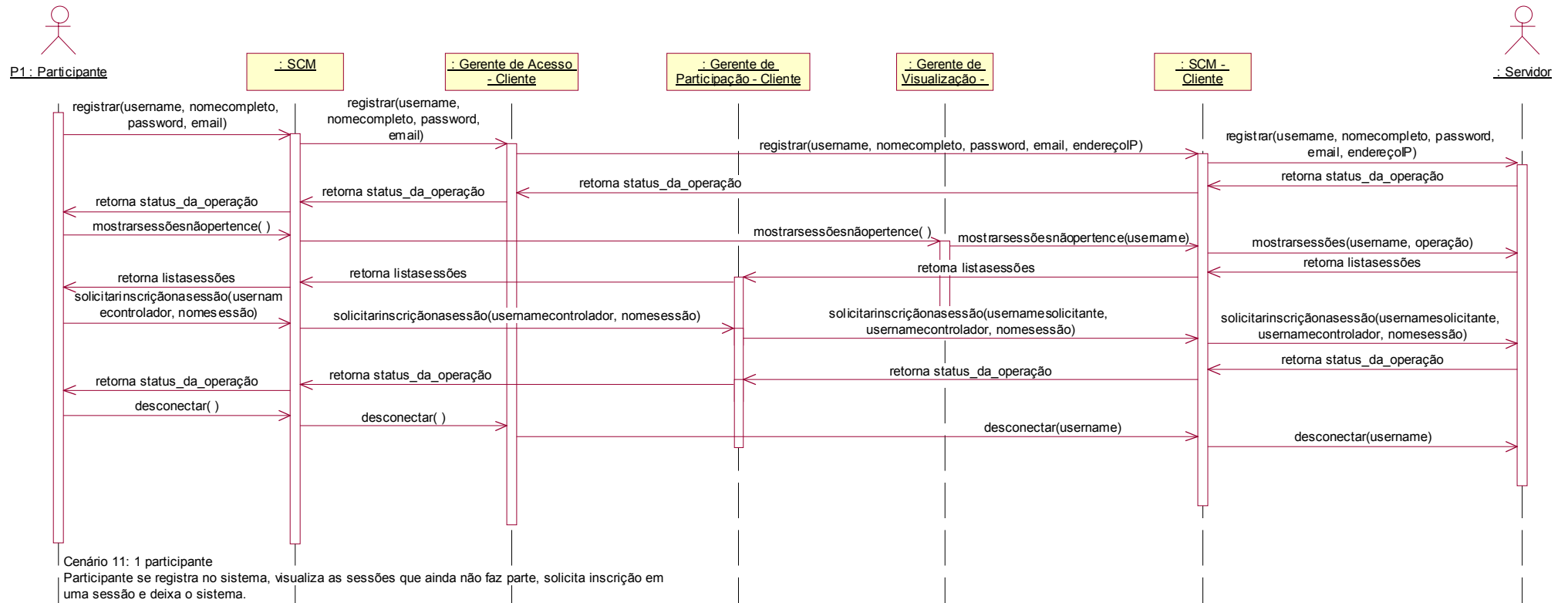


Figura 4.8. Nível de Componente – Visão Interacional – Diagrama de Seqüência – Cliente

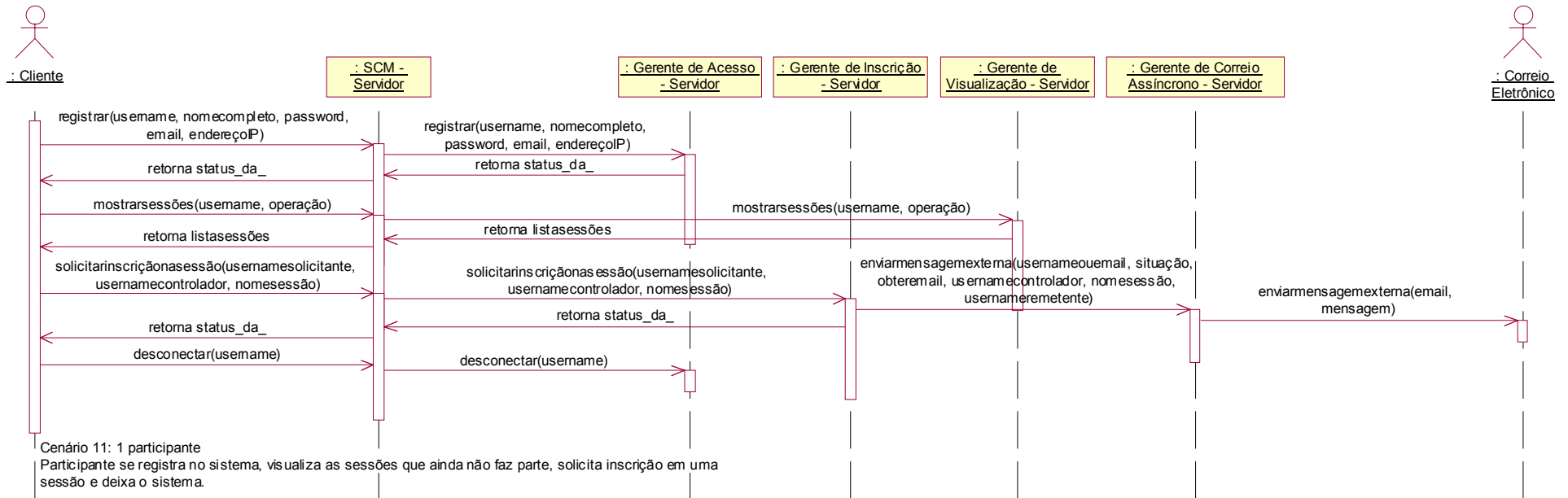


Figura 4.9. Nível de Componente – Visão Interacional – Diagrama de Sequência – Servidor

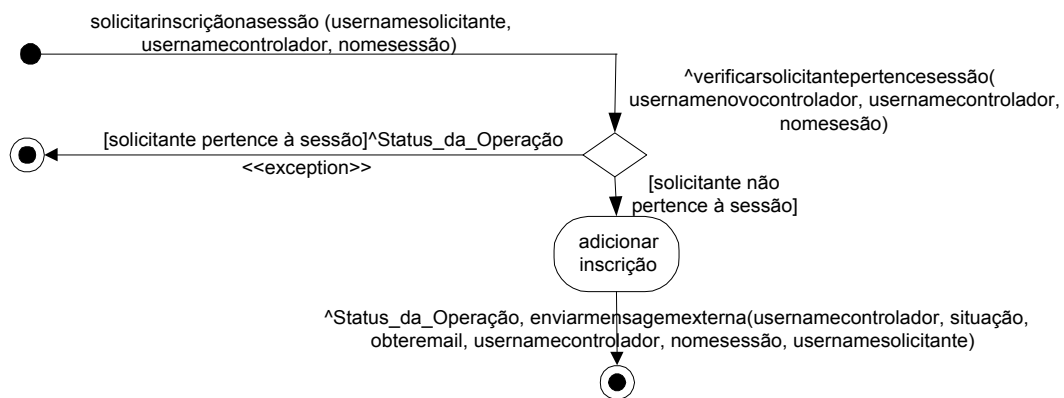


Figura 4.10. Nível de Componente – Visão Comportamental – Diagrama de Atividade – Servidor

#### 4.5. Modelagem – Nível de Objeto

O nível de objeto corresponde à estrutura interna dos componentes. Um componente é estruturado usando um conjunto de objetos relacionados implementados em uma linguagem de programação.

A visão estrutural de uma aplicação no nível de objeto pode ser representada usando diagrama de caso de uso, diagramas de classe ou de objeto. A visão comportamental pode ser representada usando diagramas de atividades ou de estados; enquanto a visão interacional pode ser representada usando diagramas de seqüência ou colaboração.

O processo de desenvolvimento de um componente no nível de objeto corresponde ao tradicional desenvolvimento de software orientado a objeto e não requer maiores discussões.

As figuras 4.11, 4.12, 4.13 e 4.14 representam, as visões estrutural (4.11 e 4.12), interacional e comportamental de uma parte do nível de objeto. O apêndice 5 apresenta a modelagem do SCM para o nível de objeto.

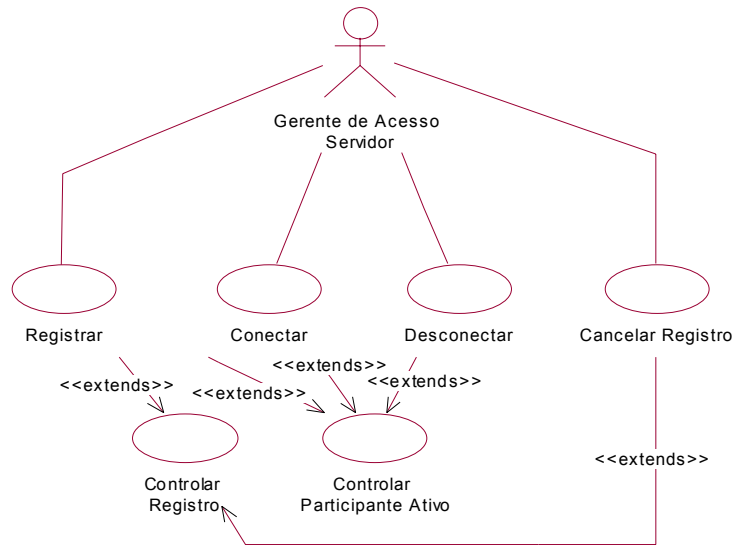


Figura 4.11. Nível de Objeto – Visão Estrutural – Diagrama de Caso de Uso – Gerente de Acesso Servidor

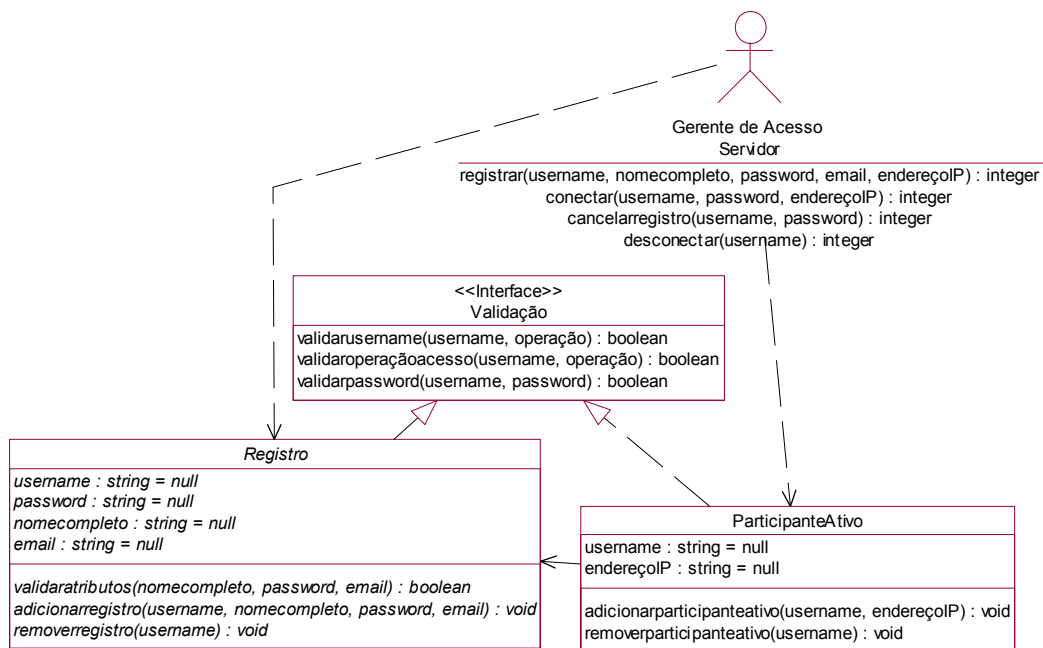


Figura 4.12. Nível de Objeto – Visão Estrutural – Diagrama de Classe – Gerente de Acesso Servidor

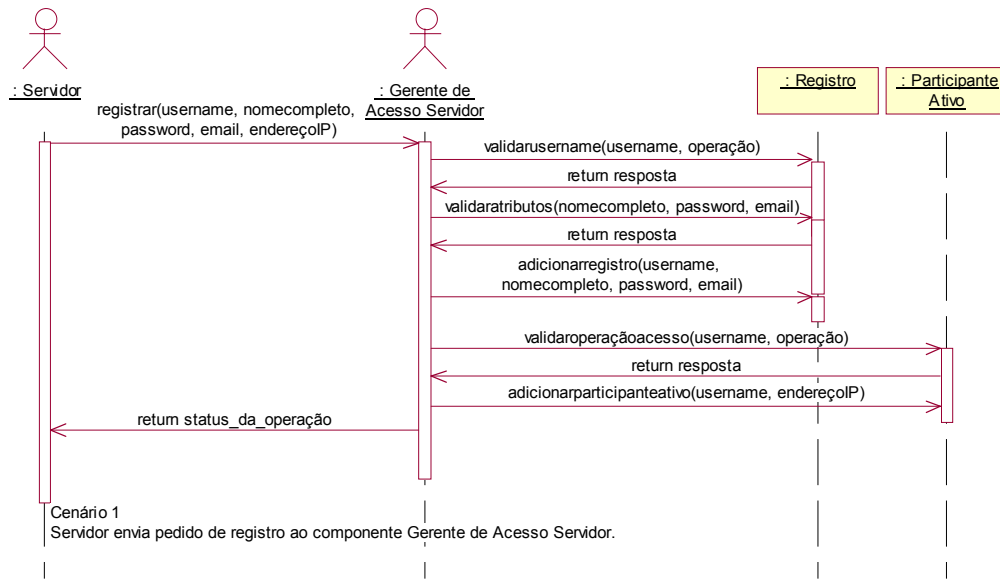


Figura 4.13. Nível de Objeto – Visão Interacional – Diagrama de Seqüência Servidor – Requisição de Registro

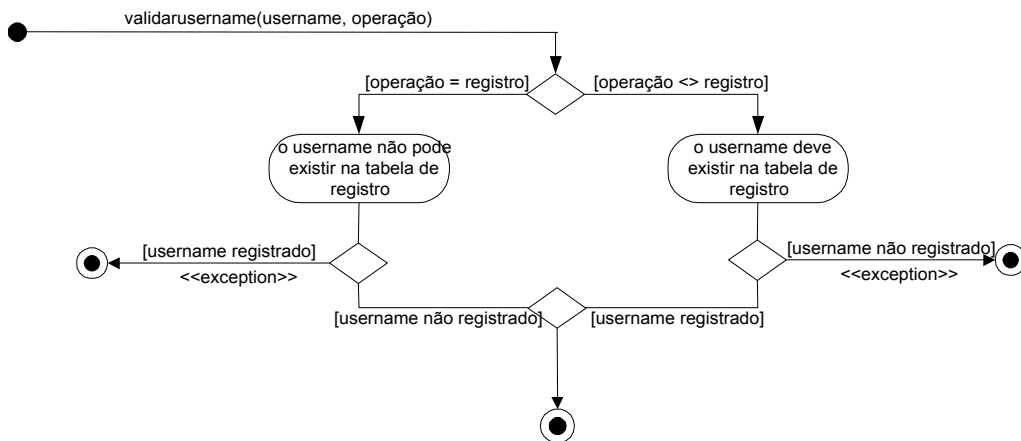


Figura 4.14. Nível de Objeto – Comportamental – Diagrama de Atividade Servidor – Requisição de Validar Username



## Capítulo V

### ***Toolkit para Aplicações Cooperativas (TAC)***

Sob o ponto de vista do processo de desenvolvimento de software, construir aplicações groupware é uma tarefa complexa. Embora avanços tenham sido realizados, nos últimos tempos, com relação ao surgimento de toolkits, arquitetura e componentes básicos para o projeto e implementação de aplicações groupware ainda são desejáveis. Neste sentido, este capítulo apresenta os componentes identificados neste trabalho e uma descrição dos mesmos.

#### **5.1. Origem do TAC**

A partir da modelagem realizada para o Sistema de Conferência Multimídia observou-se que, embora os componentes identificados baseiem-se em uma aplicação cooperativa específica, estes são genéricos. Os serviços providos por estes são utilizados por várias aplicações cooperativas e as justificativas para essa generalidade são apresentadas a seguir.

O gerenciamento de acesso é uma atividade cooperativa fundamental. Toda aplicação cooperativa, independentemente de sua natureza, deve prover mecanismos para controlar quais participantes podem utilizar os serviços oferecidos por esta. Neste sentido, são atividades oferecidas por esta funcionalidade: registrar o participante no sistema, conectar o participante ao sistema, desconectar o participante do sistema e remover o registro do participante. Conseqüentemente, o componente gerente de acesso pode ser utilizado no desenvolvimento de diversas aplicações cooperativas tais como educação à distância, sala de aula eletrônica, sala de reunião eletrônica, conferência, sistemas de gerenciamento de workflow, editores multiusuários, dentre outros, a fim de prover o controle de acesso a essas atividades. Assim, o projetista não precisa se preocupar com o desenvolvimento da lógica necessária para se controlar o acesso do participante, pois a mesma está encapsulada no componente gerente de acesso; sendo necessário apenas utilizar os serviços oferecidos através das interfaces disponíveis para o componente. A mesma idéia se aplica para os demais componentes.

---

O gerenciamento de sessão deve manter informações sobre as sessões cooperativas que permitirão uma posterior interação entre seus membros. Desta forma, os horários para realização de uma reunião cooperativa podem ser antecipadamente marcados. Neste sentido, uma aula de ensino a distância, por exemplo, é considerada uma sessão tendo suas informações armazenadas através da utilização dos serviços oferecidos para o gerenciamento de sessão. As atividades agrupadas por este serviço são: criar sessão, atualizar sessão, transferir direitos de controlador para outro participante e elaborar pesquisas a serem respondidas pelos participantes.

O gerenciamento de participação deve prover mecanismos para permitir que um participante una-se a uma sessão cooperativa. Para tanto o mesmo deve saber da existência da sessão. Informações tais como o nome da sessão, quem é o responsável por esta, sua finalidade e quando a atividade cooperativa será realizada devem ser divulgadas. Assim, são atribuições deste serviço as tarefas de permitir a inscrição do participante no sistema, incluindo as atividades de solicitar inscrição na sessão e analisar inscrições (aceitar ou recusar); de convidar pessoas para participarem de uma sessão, com as atividades de convidar e analisar convites recebidos (aceitar ou recusar); de enviar mensagem utilizando um serviço externo de correio eletrônico e de remover participação na sessão.

O gerenciamento de cooperação deve permitir a administração da cooperação entre os membros de uma sessão tais como o início de uma cooperação, o controle de entrada e saída de participantes, o controle da frequência dos participantes e o envio e recepção de áudio/vídeo/texto entre os participantes durante uma cooperação. Neste sentido, são atribuições deste serviço a comunicação, incluindo as atividades de estabelecer ou finalizar uma cooperação; a escolha da mídia a ser utilizada durante a cooperação, permitindo a inclusão ou remoção de mídias; a troca de mensagem síncrona entre os participantes em cooperação, o envio e a recepção de áudio e vídeo, a votação, a transferência de arquivos entre os participantes de uma sessão, a realização de anotações pelos participantes e a gravação de trechos da conferência.

Finalmente a visualização das informações deve permitir que sejam apresentadas informações aos participantes da aplicação cooperativa tais como as sessões controladas

---

por um participante, os participantes que fazem parte de uma determinada sessão e as sessões em que um dado participante pode estabelecer cooperação.

Por permitir sua utilização em diversas aplicações cooperativas, como explicado anteriormente, a generalidade dos componentes pode ser observada. Além da generalidade, os componentes podem ser estendidos a qualquer momento. Sendo os componentes blocos de programa que podem ser adicionados à aplicação cooperativa a ser desenvolvida, qualquer modificação pode ser realizada nestes blocos para adequá-los a situações específicas, tornando-os adaptáveis a toda aplicação cooperativa.

## **5.2. Arquitetura do TAC**

O TAC usa uma arquitetura cliente-servidor. Foram identificados componentes para a aplicação cliente e aplicação servidora, os quais são chamados de componentes cliente e componentes servidor.

A aplicação cooperativa executada em diferentes locais consiste de várias instâncias da aplicação que, por sua vez, instancia cada componente cliente. Uma instância do componente cliente se comunica com componente servidor através da troca de mensagens. Componentes servidor são componentes centralizados que manipulam as informações necessárias para garantia da execução da aplicação cooperativa, enquanto componentes cliente são componentes distribuídos em cada plataforma de execução da aplicação para preparar as requisições a serem enviadas ao componente servidor.

Há uma relação de dependência entre a aplicação cooperativa e os componentes cliente, assim como entre os componentes servidor. Essa dependência está representada pelas figuras 5.1 e 5.2, respectivamente, e é explicada no decorrer deste capítulo.

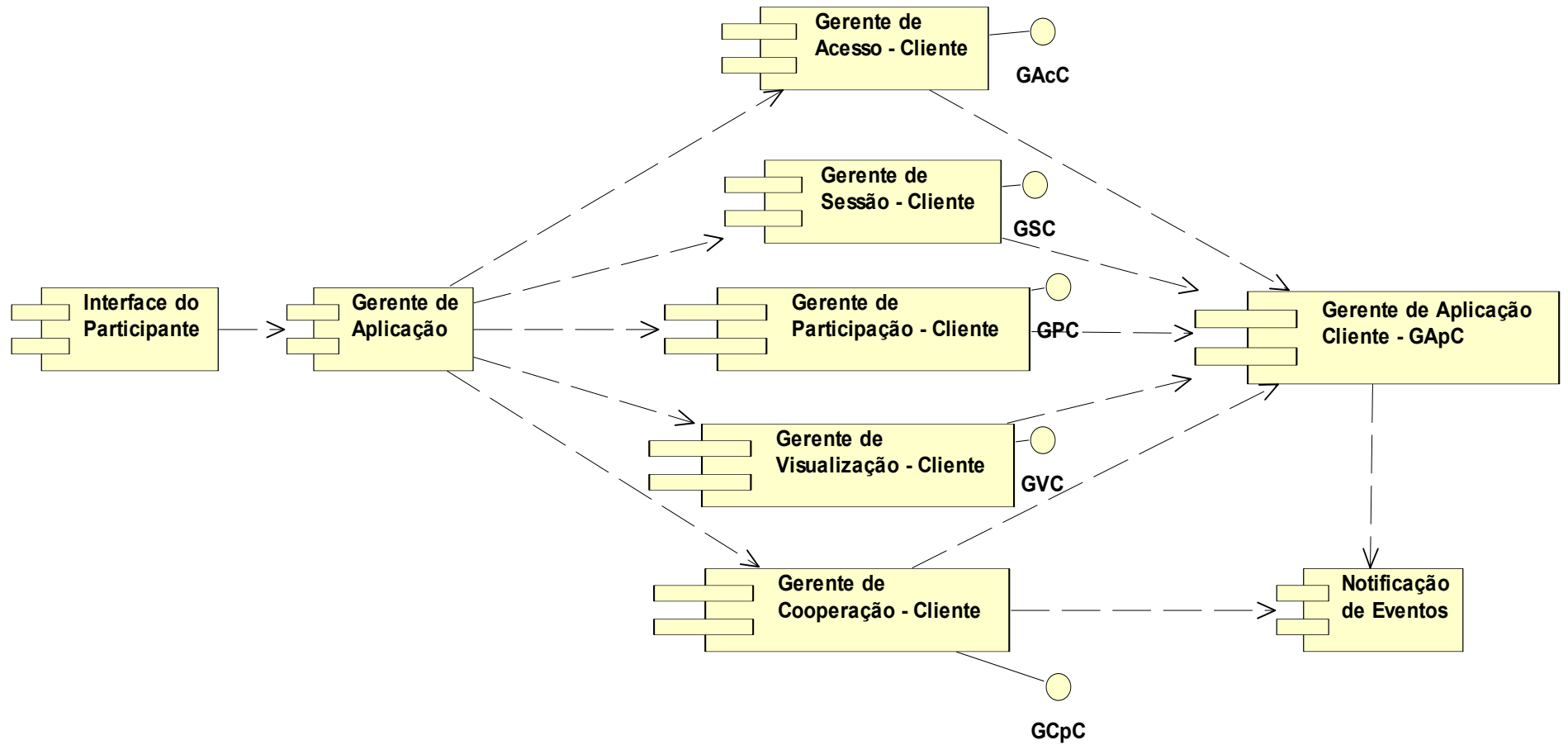


Figura 5.1 Estrutura do TAC Cliente

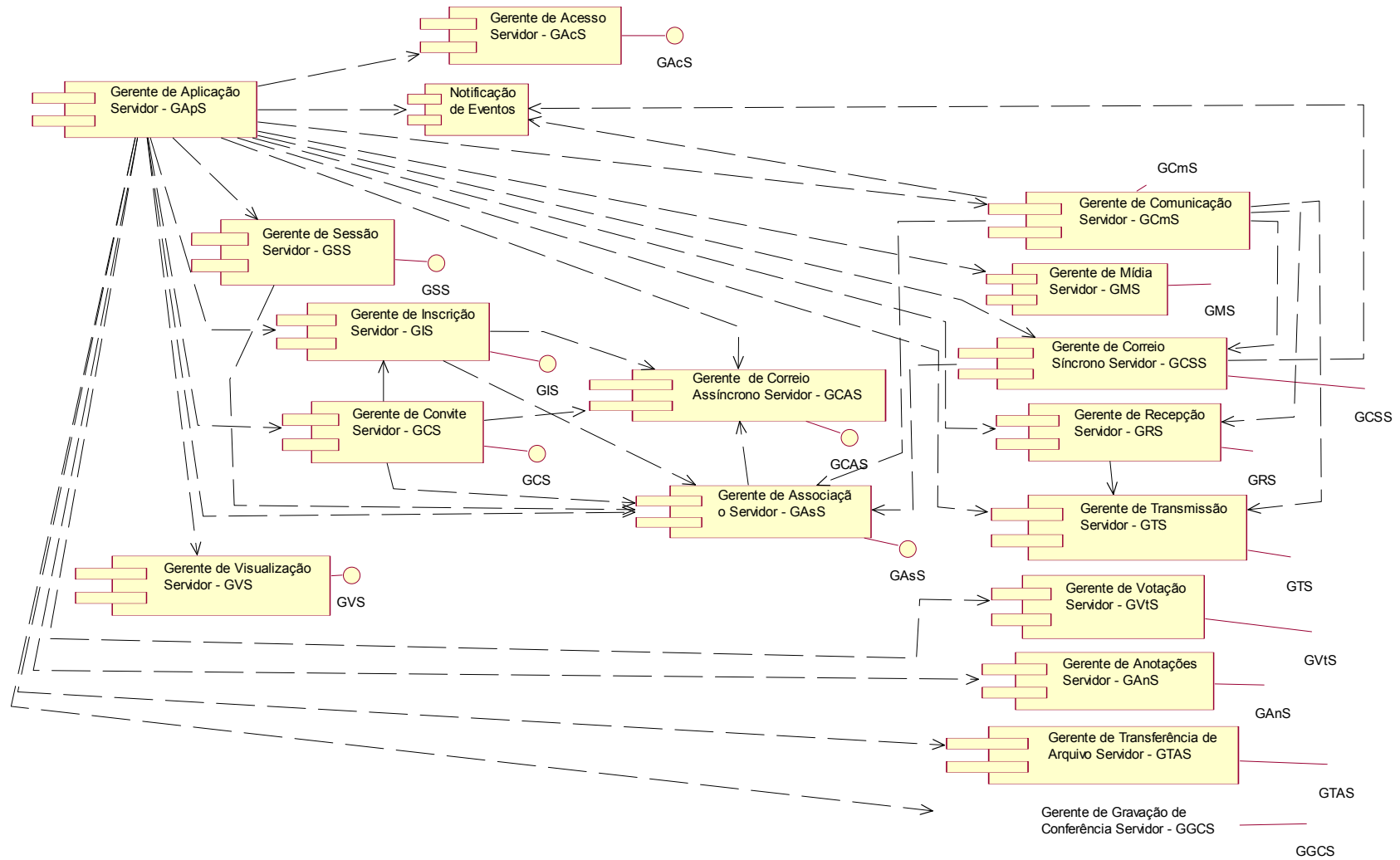


Figura 5.2 Estrutura do TAC Servidor

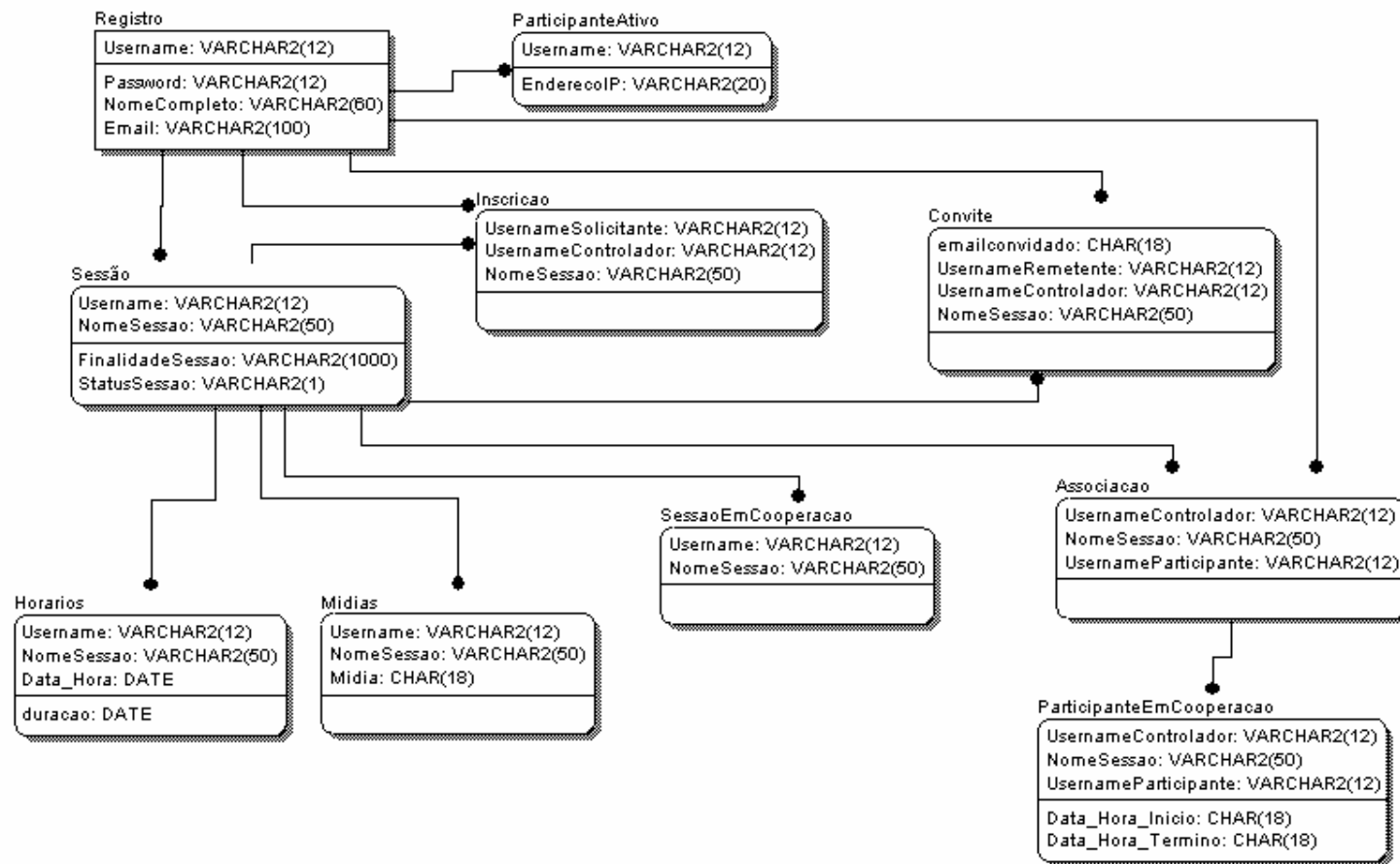


Figura 5.3 Estrutura do Banco de Dados

---

A persistência das informações relevantes para a garantia das funcionalidades deste toolkit está mantida em um banco de dados, cuja modelagem está apresentada pela figura 5.3.

### **5.3. Componentes Cliente**

Os componentes cliente podem ser combinados para compor uma aplicação cooperativa cliente e estão subdivididos em três categorias: (1) componente responsável pela comunicação entre a aplicação específica e componentes cliente, (2) componentes cliente propriamente ditos e (3) componente cliente responsável pela comunicação entre componentes cliente e componentes servidor.

Como componente pertencente à primeira categoria encontra-se apenas o Gerente de Aplicação (GA). Este componente deve receber as requisições da aplicação cooperativa específica e repassá-las a cada componente cliente responsável por manipular a requisição.

Os componentes pertencentes à segunda categoria são os responsáveis por manipular as requisições recebidas pelo componente GA e transformá-las em requisições a serem enviadas aos componentes servidor. Essas novas requisições devem conter informações suficientes e precisas para permitir que o componente servidor realize o serviço para o qual foi projetado. Fazem parte dessa categoria os seguintes componentes: Gerente de Acesso Cliente (GAcC), Gerente de Sessão Cliente (GSC), Gerente de Participação Cliente (GPC), Gerente de Cooperação Cliente (GCC) e Gerente de Visualização Cliente (GVC).

Finalmente a última categoria possui somente um componente: Gerente de Aplicação Cliente (GApC), que é o responsável pela comunicação entre os componentes cliente e os componentes servidor. As interfaces deste componente correspondem a todas as possíveis requisições enviadas por todos os componentes cliente

## 5.4. Componentes Servidor

Os componentes servidor constituem um conjunto de componentes que podem ser combinados para compor uma aplicação cooperativa servidora. Estes componentes provêm funcionalidades comuns às aplicações cooperativas, liberando o projetista e os desenvolvedores para se preocuparem somente com a lógica específica da aplicação cooperativa servidora.

### 5.4.1. Gerente de Aplicação Servidor (GApS)

Este componente é o responsável pela comunicação entre os componentes cliente (através do componente GApC) e os componentes servidor, recebendo as requisições do cliente e encaminhando-as a cada componente servidor.

As interfaces deste componente correspondem a todas as possíveis requisições enviadas pelos componentes cliente aos componentes servidor.

### 5.4.2. Gerente de Acesso Servidor (GAcS)

O componente GAcS permite administrar o acesso do participante a uma aplicação cooperativa. As atividades a serem controladas por este componente são: registro, cancelamento do registro, conexão ou desconexão.

Diante de um pedido de registro, o componente deve validar o username fornecido e os demais atributos (nome completo, password e email). Somente um registro deve ser aceito para cada username; e os três atributos que caracterizam o username são requeridos. Não ocorrendo erros, um registro deve ser adicionado ao banco de dados para controlar as pessoas registradas no sistema. O componente deve, então, conectar o participante ao sistema. Para tanto, o pedido de conexão deve ser validado, não sendo aceita mais de uma conexão por username registrado. Após a validação do pedido de conexão o componente deve adicionar o participante ao banco de dados, visando a controlar os participantes conectados ao sistema.

O pedido de cancelamento do registro tem a finalidade de retirar as informações sobre o participante que se encontram armazenadas no banco de dados. O comportamento do componente diante deste pedido deve ser o de validar o username e a password do



solicitante e verificar se o solicitante não está conectado ao sistema. Não ocorrendo erros, o solicitante deve ser removido do banco de dados.

Quando uma requisição de conexão for enviada ao componente GAcS, o username e a password do solicitante devem ser validados. O componente deve aceitar este pedido somente se ainda não existir uma conexão para o solicitante. Estando válidas essas informações, o solicitante deve ser adicionado ao banco de dados a fim de que sejam controlados os participantes conectados ao sistema.

Finalmente, diante de um pedido de desconexão, o componente deve remover o participante do banco de dados, notificando o sistema que o participante não mais se encontra conectado a este. Para que este pedido seja aceito, o participante não deve estar executando atividades de cooperação.

São métodos deste componente: registrar, conectar, cancelarregistro, desconectar, validarusername, validaratributos, validaroperaçãoacesso, validarpassword, adicionarregistro, adicionarparticipanteativo, removerregistro e removerparticipanteativo. A figura 5.4 mostra as classes e os métodos deste componente.

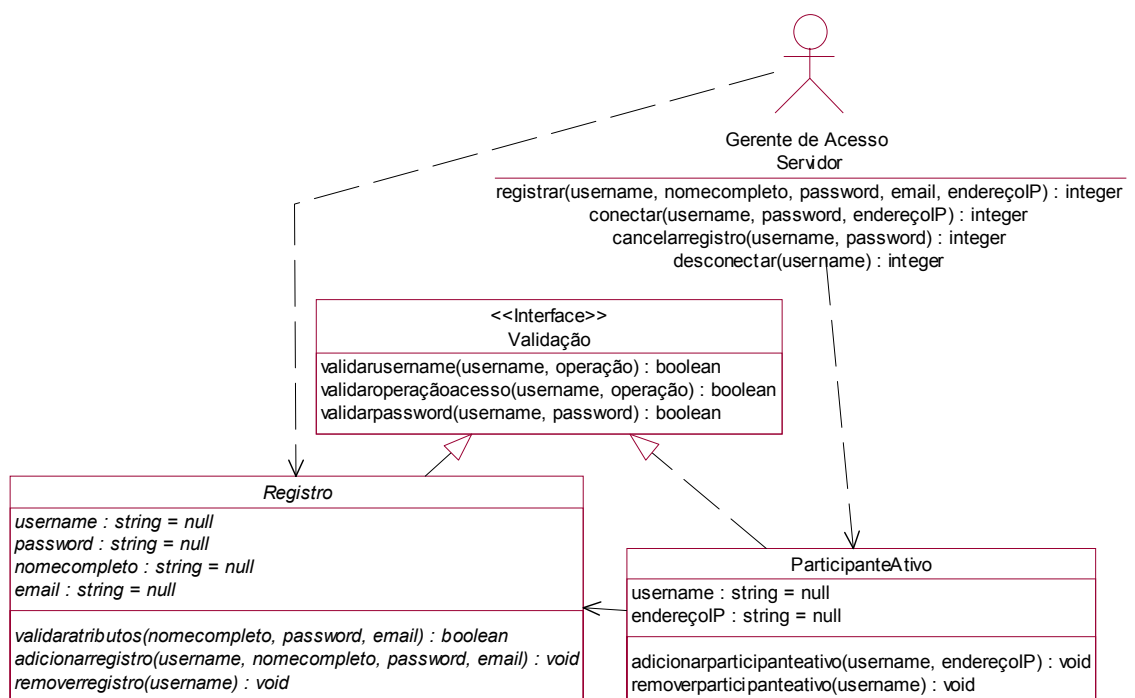


Figura 5.4. Classes e Métodos do Componente GAcS

### 5.4.3. Gerente de Sessão Servidor (GSS)

Este componente permite manter sob controle as sessões que permitirão a cooperação entre seus membros, incluindo informações sobre data e horário das cooperações, finalidade da sessão, mídias disponíveis para serem utilizadas pelos participantes da sessão e os responsáveis pela sessão. As atividades sob controle deste componente são: criar uma sessão, atualizar nome ou finalidade da sessão, atualizar horários para cooperação, atualizar mídias a serem utilizadas, bloquear sessão e transferir direitos de controlador.

Diante de um pedido de criação de uma sessão, o componente deve validar o nome desta. Não será permitida a criação de duas sessões com o mesmo nome por um mesmo participante. As informações sobre a sessão, a saber: nome, finalidade, horários, mídias e status devem ser armazenadas no banco de dados; assim como o solicitante deste pedido, a fim de controlar os responsáveis e o participantes pela/por sessão. O status da sessão define se esta está ativa ou se suas atividades foram encerradas pelo seu controlador.

A atualização do nome e/ou finalidade de uma sessão, atualização dos horários de cooperação e atualização das mídias disponíveis para a sessão só devem ser permitidas pelo componente caso a sessão não esteja em cooperação e o solicitante da atualização seja o controlador desta. Atualizar horário consiste na inclusão, remoção ou alteração de horários para cooperação; enquanto atualizar mídias consiste na inclusão ou remoção de mídias.

Bloquear uma sessão consiste no encerramento das atividades ou na remoção da sessão. Quando uma requisição de bloquear a sessão for enviada ao componente GSS, este somente deve aceitar o pedido caso a sessão não esteja em cooperação e o solicitante do bloqueio seja o controlador da sessão. Sendo a requisição de bloqueio um pedido de encerramento das atividades, o componente deve alterar o status da sessão para encerrada, somente se esta não tem horários futuros para cooperação. Sendo a requisição de bloqueio um pedido de remoção da sessão, este somente deve ser aceito caso não existam cooperações anteriores para a sessão. Sendo possível a remoção da sessão, também devem ser removidos os horários, as mídias e os participantes por sessão.

---

O Componente GSS deve aceitar um pedido para transferir direitos de controlador da sessão para outro participante somente se: a sessão não estiver em cooperação, o solicitante seja o controlador da sessão, o username do novo controlador estiver registrado no sistema e o username do novo controlador fizer parte da sessão que irá controlar. Estando satisfeitas essas condições, o componente deverá alterar no banco de dados todas as informações sobre sessão e controlador antigo.

A figura 5.2 mostra uma dependência entre o componente GSS e GAsS. Essa dependência refere-se à atividade de adicionar ou remover participante por sessão, verificar se o controlador informado é realmente o controlador da sessão, verificar se o novo controlador pertence à sessão e verificar se a sessão está em cooperação, que são controladas pelo componente GAsS.

São métodos deste componente: criarsessão, atualizarnomefinalidade, atualizarhorário, atualizarmídia, bloquearsessão, transferirdireitoscontrolador, validarnomesessão, adicionarsessão, adicionarhorário, adicionarmídia, alterarnomesessãomídia, alterarnomesessãohorário, removersessão, validarhorário, removerhorário, alterarhorário, validarmídia, removermídia, alterarcontroladormídia, alterarcontroladorhorário, alterarfinalidadesessão, verificarhoráriosfuturos, verificarinformaçãoarmazenada, encerraratividadesessão e validarusername novocontrolador. A figura 5.5 mostra as classes e os métodos deste componente.

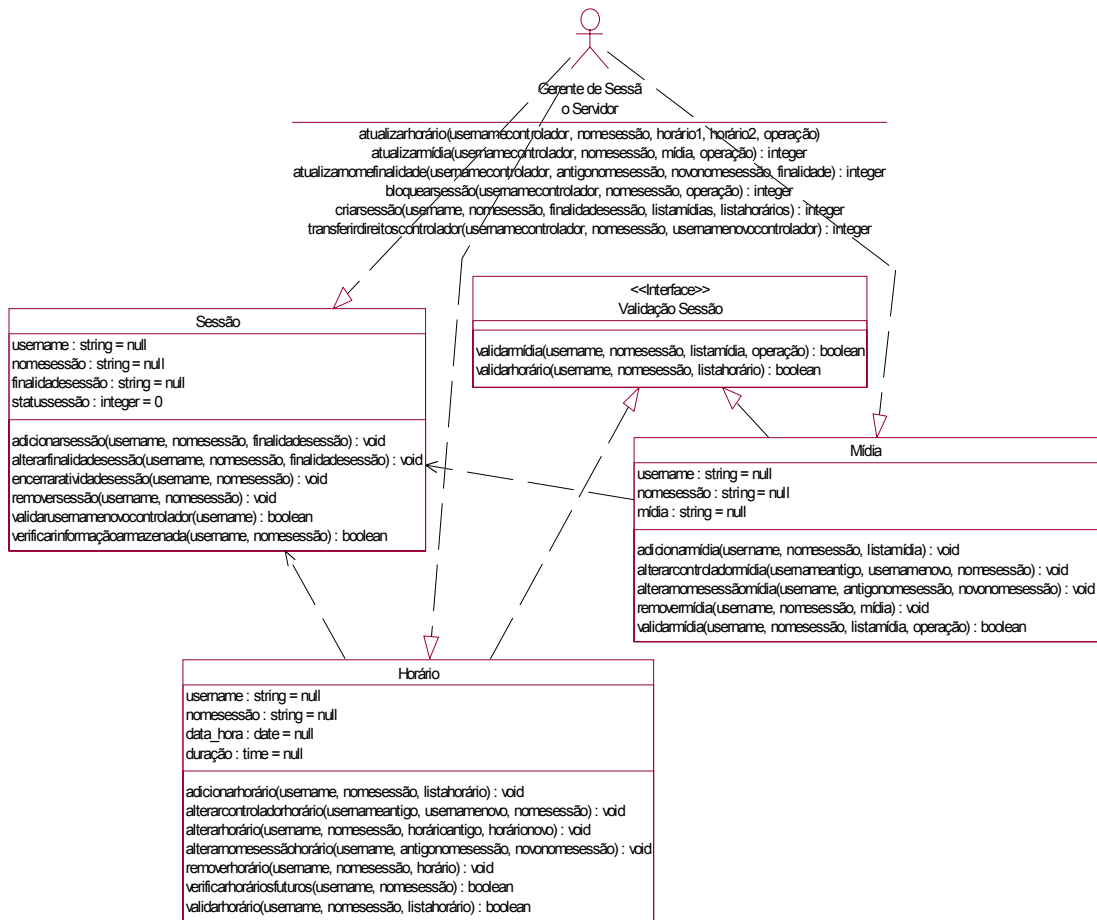


Figura 5.5. Classes e Métodos do Componente GSS

#### 5.4.4. Gerente de Participação Servidor (GPS)

O componente GPS é um componente composto. Um componente composto é resultado da agregação de vários componentes tal que o comportamento externo do componente combinado corresponda a um único componente. A finalidade deste componente é administrar a participação das pessoas nas sessões armazenadas no banco de dados. Para formação deste foram agrupados os seguintes componentes: Gerente de Inscrição Servidor (GIS), Gerente de Convite Servidor (GCS), Gerente de Correio Assíncrono Servidor (GCAS) e Gerente de Associação Servidor (GAsS), que serão descritos a seguir.

##### a) Gerente de Inscrição Servidor (GIS)

---

Tendo conhecimento da existência de uma sessão, o participante pode solicitar sua inscrição nesta, a fim de que possa participar das cooperações a serem realizadas. O controle das inscrições é realizado pelo componente GIS e consiste na inscrição do participante, na análise da inscrição pelo controlador da sessão, em apresentar ao participante as sessões às quais ele não pertence a fim de que possa solicitar sua inscrição na mesma e em apresentar ao controlador quais inscrições devem ser analisadas.

O componente deve aceitar um pedido de inscrição na sessão somente se o solicitante não fizer parte da sessão. Satisfeita a condição o componente deve adicionar o solicitante ao banco de dados a fim de controlar quais são as inscrições pendentes que devem, posteriormente, ser analisadas pelo controlador e enviar uma mensagem ao controlador da sessão utilizando um serviço externo de correio eletrônico, informando-o sobre a inscrição que deve ser analisada.

Diante de um pedido para analisar as inscrições o componente deve permitir o acesso somente se o participante analista for realmente o controlador da sessão e o solicitante da inscrição tenha realmente uma inscrição a ser analisada. Como resposta do controlador da sessão, este poderá aceitar ou recusar a inscrição do participante. Em ambos os casos o componente deve remover a inscrição pendente do banco de dados e enviar uma mensagem ao solicitante da inscrição utilizando um serviço externo de correio eletrônico, informando-o sobre a resposta do controlador ao seu pedido de inscrição. Somente para o caso de aceitação da inscrição o componente deve adicionar o solicitante da inscrição ao banco de dados como um participante da sessão.

A figura 5.2 mostra uma dependência entre este componente e o GAsS. Essa dependência refere-se às atividades de verificar se o controlador informado é realmente o controlador da sessão, verificar se o solicitante da inscrição ainda não faz parte da sessão e de adicionar participante à sessão, que são controladas pelo componente GAsS. Também é representada por esta figura a dependência entre este componente e o GCAS, referindo-se à atividade de enviar mensagem utilizando um serviço externo de correio eletrônico invocada por GIS.

São métodos deste componente: solicitarinscriçãonasessão, mostrarinscriçõespendentes, analisarinscrição, adicionarinscrição, verificarinscriçãoependente, removerinscrição e obterinscrições. A figura 5.6 mostra as classes e os métodos deste componente.

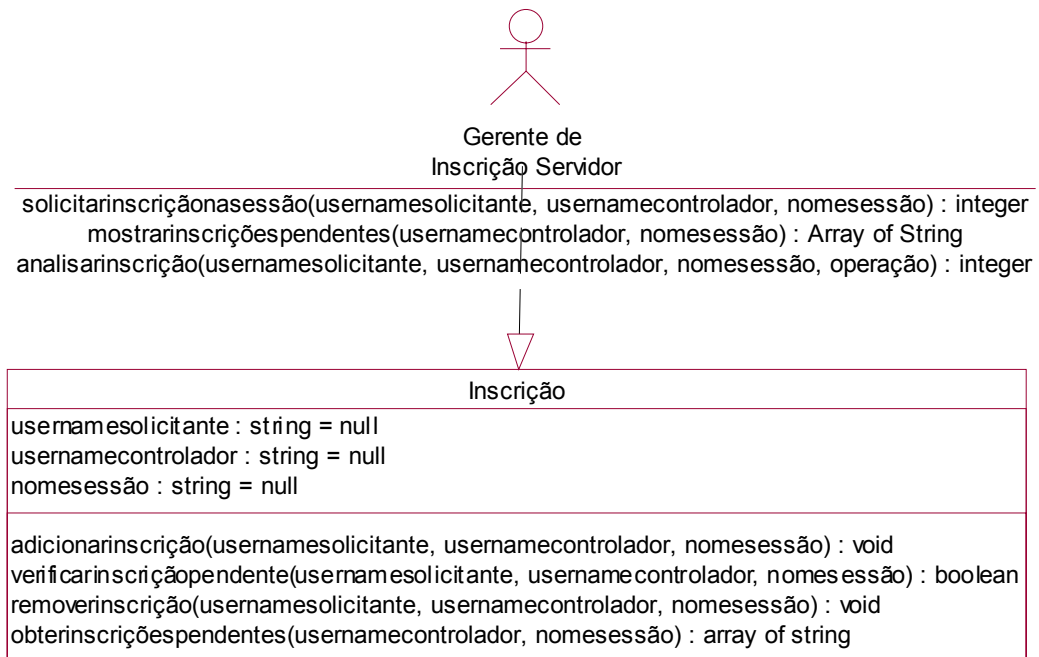


Figura 5.6. Classes e Métodos do Componente GIS

b) Gerente de Convite Servidor (GCS)

Os participantes de uma sessão também podem convidar outras pessoas, registradas ou não no sistema, para participarem da sessão. O controle de convites é realizado pelo componente GCS e consiste no convite, na análise do convite e em apresentar aos participantes quais convites foram recebidos.

O componente deve aceitar um pedido de convite somente se a sessão informada estiver armazenada no banco de dados, o remetente do convite fizer parte da sessão, o convidado não pertencer à sessão e este ainda não foi convidado para participar da sessão. O componente deve, então, adicionar o convidado ao banco de dados a fim de controlar os convites a serem analisados e enviar uma mensagem ao convidado utilizando um serviço externo de correio eletrônico, informando-o sobre a sessão

---

para a qual foi convidado (nome, finalidade, responsável) e quem foi o remetente do convite.

Analisar um convite consiste em aceitar ou recusar um convite recebido. Diante de uma requisição para analisar um convite recebido o componente GCS deve aceitar o pedido somente se o solicitante realmente recebeu o convite e a sessão informada estiver armazenada no banco de dados. Tanto na aceitação quanto na recusa, o componente deve remover o convidado do banco de dados. Diante da recusa do convite, uma mensagem deve ser enviada ao remetente do convite informando-o sobre a recusa do mesmo. Diante da aceitação do convite, o componente deve seguir dois rumos dependendo de quem é o remetente do convite. Caso o remetente seja o controlador da sessão, o componente deve adicionar o convidado ao banco de dados como um participante da sessão e enviar uma mensagem ao controlador da mesma, informando-o sobre a aceitação do convite. Caso contrário, o componente deve adicionar uma inscrição do convidado ao banco de dados para controlar as inscrições a serem analisadas pelo controlador da sessão e enviar uma mensagem ao controlador da sessão informando-o sobre o pedido de inscrição.

A figura 5.2 mostra uma dependência entre este e GAsS; e este e GIS. Essas dependências referem-se às atividades de validar o controlador da sessão, verificar se o remetente pertence à sessão e adicionar o participante por sessão, que são controladas pelo componente GAsS; e adicionar inscrição a ser analisada, que é controlada pelo componente GIS. Também é representada por esta figura a dependência entre este componente e GCAS, referindo-se à atividade de enviar mensagem utilizando um serviço externo de correio eletrônico invocada por GCS.

São métodos deste componente: `convidar pessoas`, `mostrar convites recebidos`, `analisar convites`, `adicionar convidado`, `verificar existe convite`, `remover convidado`, `obter user name remetente`, `obter convites` e `verificar convidado pertence sessão`. A figura 5.7 mostra as classes e os métodos deste componente.

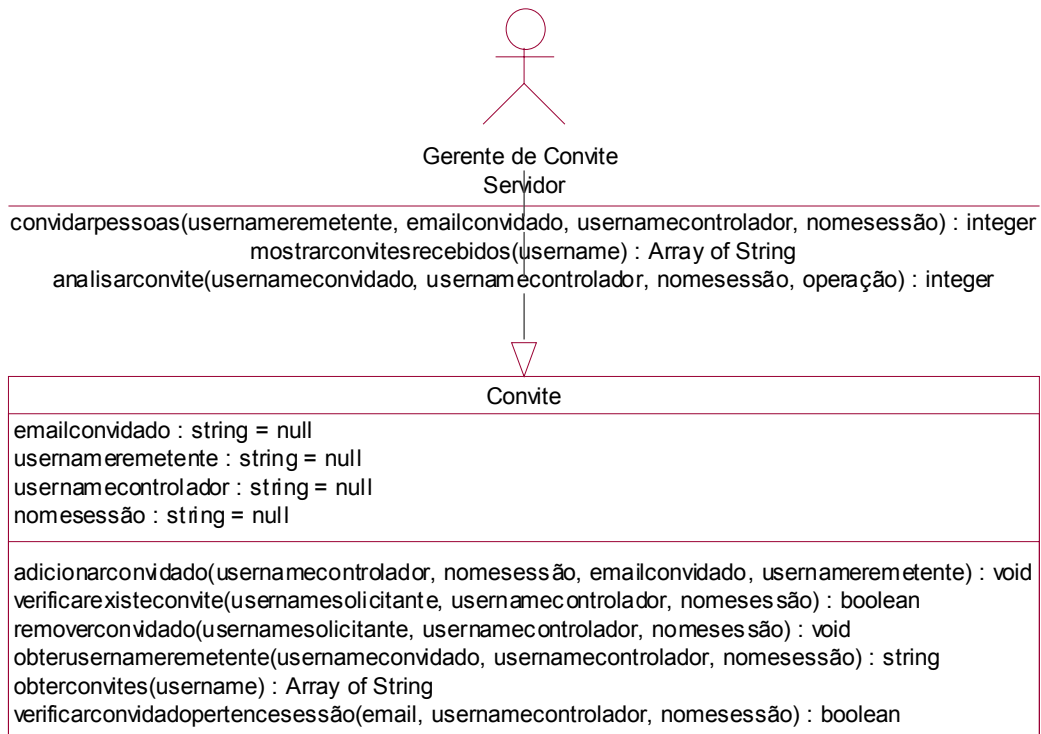


Figura 5.7. Classes e Métodos do Componente GCS

c) Gerente de Associação Servidor (GAsS)

Este componente permite controlar os participantes que se associaram a uma sessão através da remoção destes do banco de dados. Este componente deve aceitar o pedido de remoção do participante da sessão somente se o participante pertencer à sessão e se este não se encontrar em cooperação. O componente deve remover o solicitante da sessão e enviar uma mensagem ao controlador desta, utilizando um serviço externo de correio eletrônico, informando-o sobre a desistência do participante em continuar fazendo parte da sessão.

A figura 5.2 representa uma dependência entre este componente e GCAS, referindo-se à atividade de enviar mensagem utilizando um serviço externo de correio eletrônico invocada por GAsS.

Alguns métodos utilizados por outros componentes foram inseridos neste por apresentarem certa semelhança quanto à funcionalidade. Como método utilizado somente por este componente, encontra-se solicitarremoçãonasessão. Os demais



métodos são, também, invocados por outros componentes, justificando a dependência entre os componentes conforme representação na figura 5.2. São eles: verificarsolicitantemcooperação, verificarsolicitantepertenteasessão, adicionarparticipantenasessão, removerparticipantenasessão, validarcontroladorsessão e verificarsessãoemcooperação. A figura 5.8 mostra as classes e os métodos deste componente.

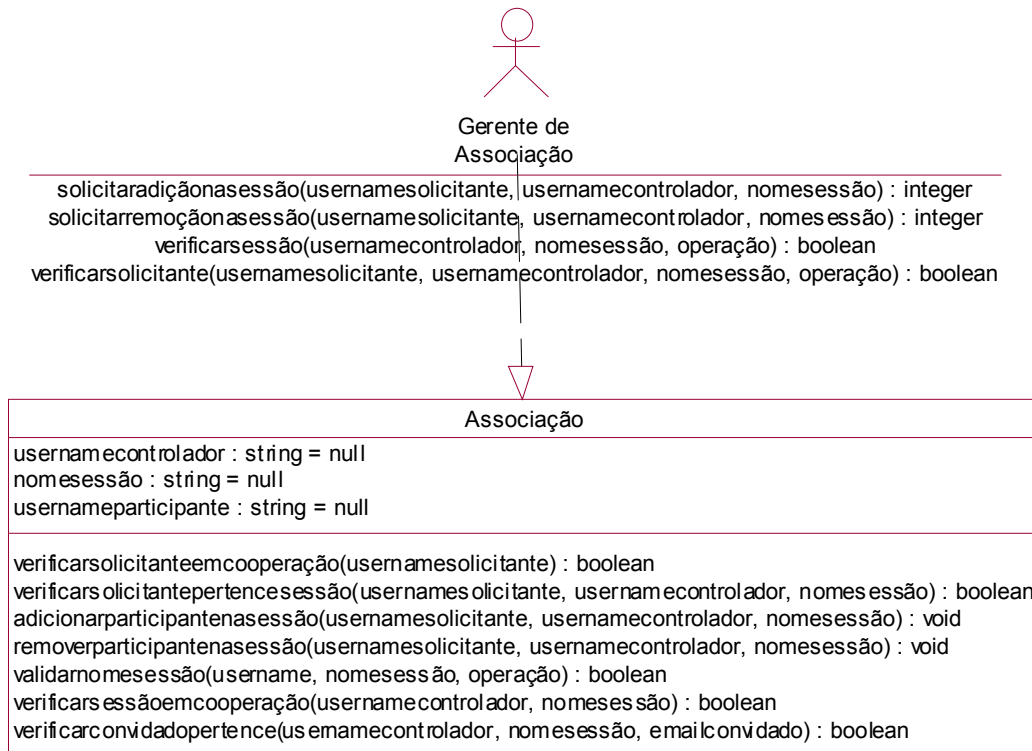


Figura 5.8. Classes e Métodos do Componente GAS

d) Gerente de Correio Assíncrono Servidor (GCAS)

O componente GCAS tem a finalidade de enviar uma mensagem aos participantes, através da utilização de um serviço externo de correio eletrônico. O serviço oferecido por este componente pode ser invocado a partir de diferentes situações, como representadas na tabela 5.1. Neste sentido, várias mensagens podem ser remetidas e este componente deve elaborá-las de acordo com cada situação. Em algumas situações o email do destinatário também deve ser obtido por este componente, situações estas que se encontram definidas na tabela 5.1.

| Situação                                    | Mensagem  | Destinatário             | Obter Email |
|---|---|--------------------------|-------------|
| Solicitar Inscrição na Sessão               | Informar sobre a nova inscrição recebida.   | Controlador da Sessão    | Sim         |
| Analisar Inscrição Recebida – Aceitar       | Informar sobre a aceitação da inscrição.  | Solicitante da Inscrição | Sim         |
| Analisar Inscrição Recebida – Recusar       | Informar sobre a não aceitação da inscrição.  | Solicitante da Inscrição | Sim         |
| Convidar Pessoas                            | Informar sobre o convite  | Convidado                | Não         |
| Analisar Convite Recebido – Aceitar         | Informar sobre a aceitação do convite (caso o remetente seja o controlador da sessão) ou Informar sobre a solicitação de inscrição (caso o remetente não seja o controlador da sessão). | Controlador da Sessão    | Sim         |
| Analisar Convite Recebido – Recusar         | Informar sobre a não aceitação do convite.  | Remetente do Convite     | Sim         |
| Solicitar Remoção da Participação na Sessão | Informar sobre a desistência do participante.   | Controlador da Sessão    | Sim         |

Tabela 5.1. Mensagens a serem enviadas pelo componente GCAS

São métodos deste componente: `enviarmensagemexterna`, `elaborarmensagem`, `obteremail` e `enviarmensagem`. A figura 5.9 mostra as classes e os métodos deste componente.

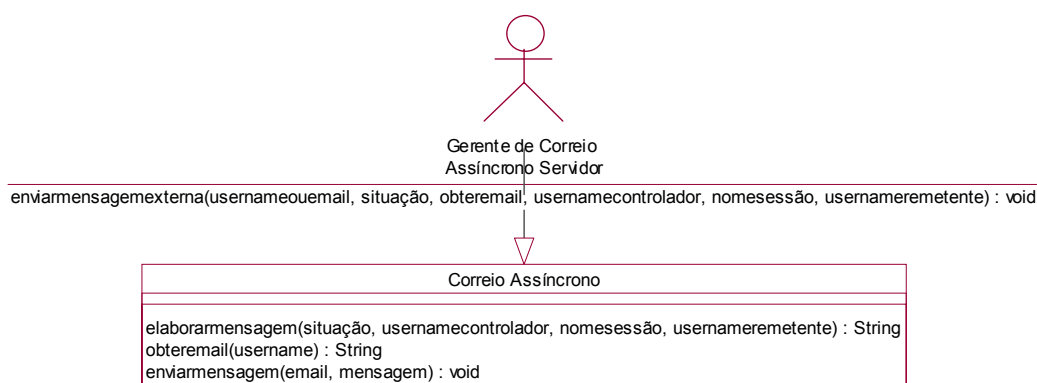


Figura 5.9. Classes e Métodos do Componente GCAS

#### 5.4.5. Gerente de Cooperação Servidor (GCpS)

O componente GCpS, assim como o GPS, é um componente composto. Sua finalidade é administrar a cooperação entre os participantes de uma sessão. Para formação deste componente foram agrupados os seguintes: Gerente de Comunicação Servidor (GCmS),

---

Gerente de Mídia Servidor (GMS), Gerente de Correio Síncrono Servidor (GCSS), Gerente de Recepção Servidor (GRS), Gerente de Transmissão Servidor (GTS), Gerente de Votação Servidor (GVtS), Gerente de Anotação Servidor (GANs), Gerente de Transferência de Arquivo Servidor (GTAS) e Gerente de Gravação de Conferência Servidor (GGCS), que serão descritos a seguir.

a) Gerente de Comunicação Servidor (GCmS)

Os membros de uma sessão podem estabelecer ou encerrar uma cooperação. Estas atividades compreendem as tarefas de iniciar, abandonar, terminar ou deixar a cooperação; e são administradas pelo componente GCmS.

Diante de um pedido para iniciar a cooperação, este componente deve analisá-lo sob duas perspectivas: a do controlador e a do participante da sessão. Sob a visão do controlador da sessão, o componente deve permitir o início da cooperação caso o horário esteja previsto, devendo adicionar a informação sobre a sessão que está em cooperação no banco de dados. Sob a visão do participante, este somente poderá iniciar a cooperação se o controlador da sessão estiver em cooperação. Em ambos os casos deve ser adicionada ao banco de dados, uma informação sobre os participantes que estão em cooperação.

A atividade de encerrar uma cooperação foi subdividida em abandonar, deixar e terminar a cooperação. Abandonar significa encerrar a cooperação sem concluir todas as atividades previstas. Terminar corresponde a uma atividade permitida somente ao controlador da sessão e deve ser executada sempre que este concluir todas as atividades previstas para a cooperação. Deixar corresponde à atividade permitida aos demais participantes da cooperação, e deve ser executada após o término da cooperação pelo controlador, sempre que o participante desejar encerrar as atividades de cooperação. O encerramento de uma cooperação deve terminar as transmissões e recepções de áudio/vídeo, caso estas ocorram, remover informações sobre os participantes que estão em cooperação do banco de dados e controlar a frequência dos participantes. Quando estas atividades forem executadas pelo controlador da sessão, o componente deve remover a informação sobre a sessão que está em cooperação; assim como enviar uma mensagem síncrona a todos os participantes em cooperação informando-os sobre o encerramento da cooperação.

Tanto no estabelecimento quanto no encerramento da cooperação o componente deve notificar os participantes sobre a presença ou ausência do solicitante desta atividade.

São métodos deste componente: iniciarcooperação, abandonarcooperação, terminarcooperação, deixarcooperação, verificarhoráriocooperação, adicionarsessãoemcooperação, adicionarparticipanteemcooperação, removersessãoemcooperação, removerparticipanteemcooperação, armazenarinformaçãocooperação, controlarfrequência e verificarparticipanteemrecepção. A figura 5.10 mostra as classes e os métodos deste componente.

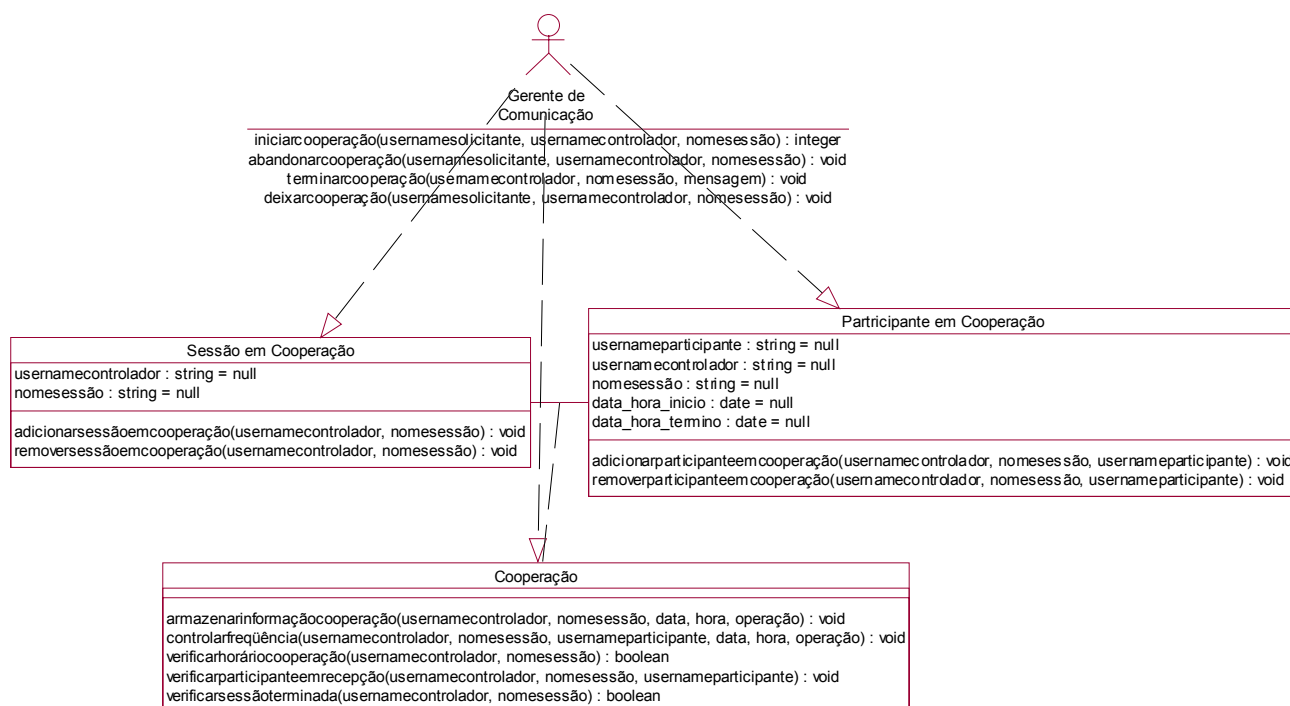


Figura 5.10. Classes e Métodos do Componente GCmS

b) Gerente de Mídia Servidor (GMS)

Este componente deve controlar quais mídias, dentre as disponíveis para cada sessão, serão utilizadas pelos participantes durante a cooperação. O componente GMS deve adicionar ou remover a informação sobre as mídias por participante/sessão no banco de dados.

São métodos deste componente: atualizarmídiacooperação, verificarmídiadisponível, verificarmídiaselecionada, adicionarmídiaporparticipante e removermídiaporparticipante. A figura 5.11 mostra as classes e os métodos deste componente.

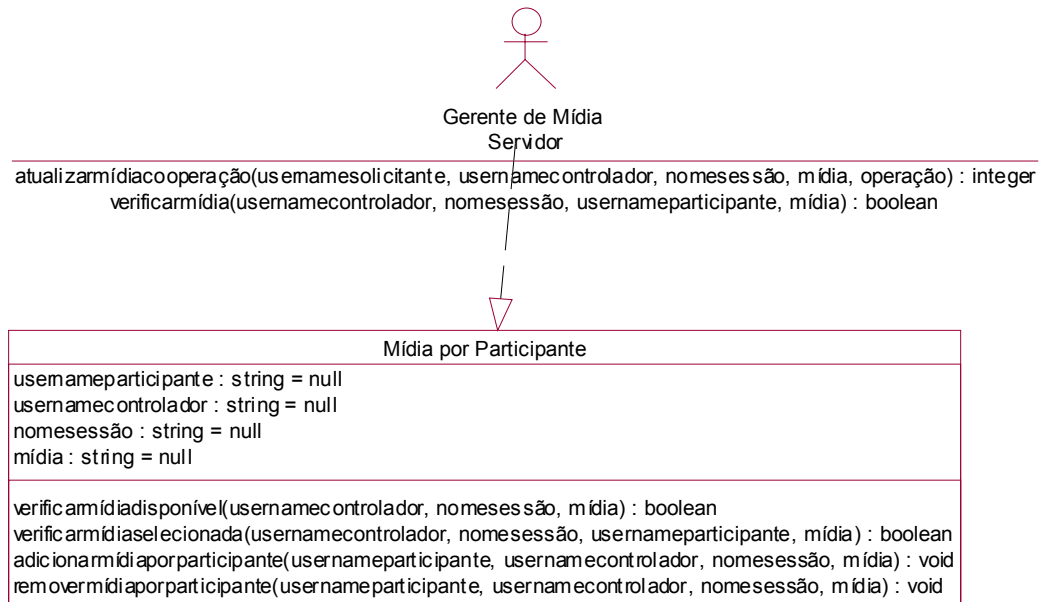


Figura 5.11. Classes e Métodos do Componente GMS

### c) Gerente de Correio Síncrono Servidor (GCSS)

Durante uma cooperação os participantes podem se comunicar através do envio e recepção de mensagens síncronas. Neste sentido, foi identificado o componente GCSS para administrar a atividade de comunicação síncrona entre os participantes em cooperação.

Diante da requisição de envio de uma mensagem síncrona, o componente deve armazenar a mesma no banco de dados, aguardando que esta seja, posteriormente, recuperada, e notificar o destinatário sobre a existência da mensagem.

Diante da requisição de recebimento de uma mensagem, este componente deve verificar se existe uma mensagem para o solicitante desta atividade e, em caso afirmativo, remetê-la ao solicitante e removê-la do banco de dados.

A figura 5.2 mostra uma dependência entre este componente e os componentes notificação de eventos e GAsS. Essas dependências justificam-se devido às atividades de notificar mensagem; verificar sessão em cooperação e verificar participante em cooperação, executadas pelo componente GCSS e controladas pelos componentes notificação de eventos e GAsS, respectivamente.

São métodos deste componente: enviarmensagem, recebermensagem, armazenarmensagem, verificarmensagem e removermensagem. A figura 5.12 mostra as classes e os métodos deste componente.

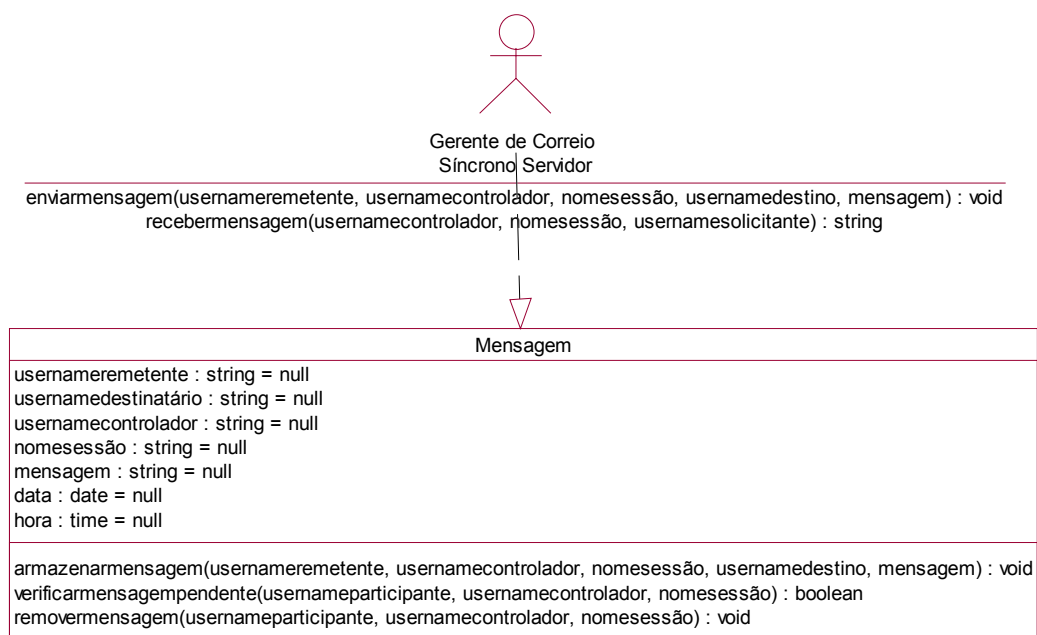


Figura 5.12. Classes e Métodos do Componente GCSS

#### d) Gerente de Recepção Servidor (GRS)

A recepção de áudio e vídeo deve ser administrada pelo componente GRS. Este componente deve permitir a execução de atividades de iniciar, suspender, reiniciar e terminar a recepção de mídia contínua. O componente deve aceitar esta requisição somente se o solicitante estiver em cooperação e o participante de quem se deseja receber a mídia contínua estiver transmitindo-a.

São métodos deste componente: solicitaroperação, recepção, verificaroperação, iniciarrecepção, reiniciarrecepção, suspenderrecepção e encerrarrecepção. A figura 5.13 mostra as classes e os métodos deste componente.

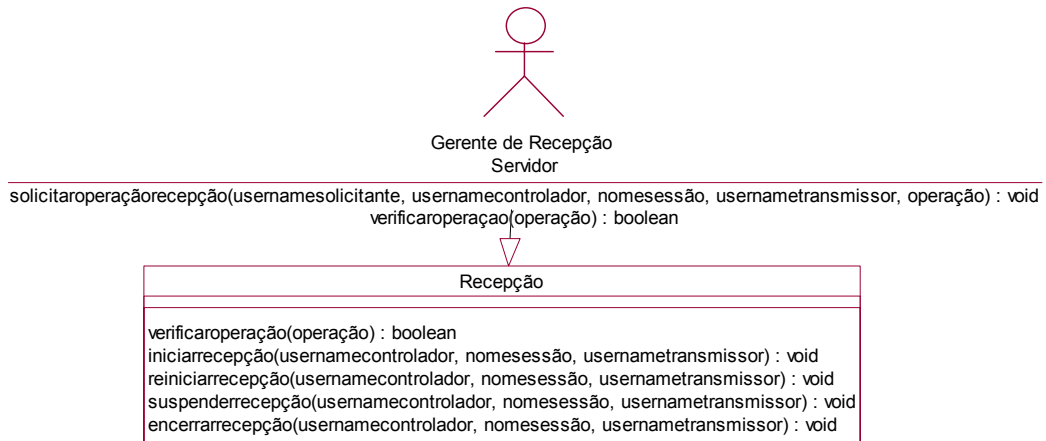


Figura 5.13. Classes e Métodos do Componente GRS

#### e) Gerente de Transmissão Servidor (GTS)

Durante a cooperação é desejável que, além da recepção de áudio e vídeo, o participante também possa transmitir mídia contínua. Esta tarefa deve ser administrada pelo componente GTS e deve permitir que a transmissão possa ser iniciada, suspensa, reiniciada e terminada.

O componente, diante de uma requisição de iniciar a transmissão, deve adicionar informações sobre o participante no banco de dados, a fim de controlar quem são aqueles que estão transmitindo uma mídia contínua. Também deve ser tarefa deste componente remover informações sobre o participante quando receber um pedido de término da transmissão da mídia contínua. Sempre que um participante iniciar a transmissão de vídeo, a mídia deve ser enviada para os endereços IP de todos os participantes que estiverem em cooperação. No caso da transmissão de áudio, o controle da palavra deve ser realizado por este componente. À medida que os participantes solicitam o controle da palavra um número sequencial lhes é atribuído. Tal seqüência define a ordem na qual um token deve ser dado ao participante, sendo transferido ao próximo participante da seqüência assim que o detentor do token desistir da palavra.

São métodos deste componente: começartransmissão, interrompertransmissão, verificarparticipanteemcooperação, verificarparticipanteemtransmissão, verificaroperação, adicionarparticipantetransmissor, removerparticipantetransmissor, iniciartransmissão, reiniciartransmissão, suspendertransmissão e encerrartransmissão. A figura 5.14 mostra as classes e os métodos deste componente.

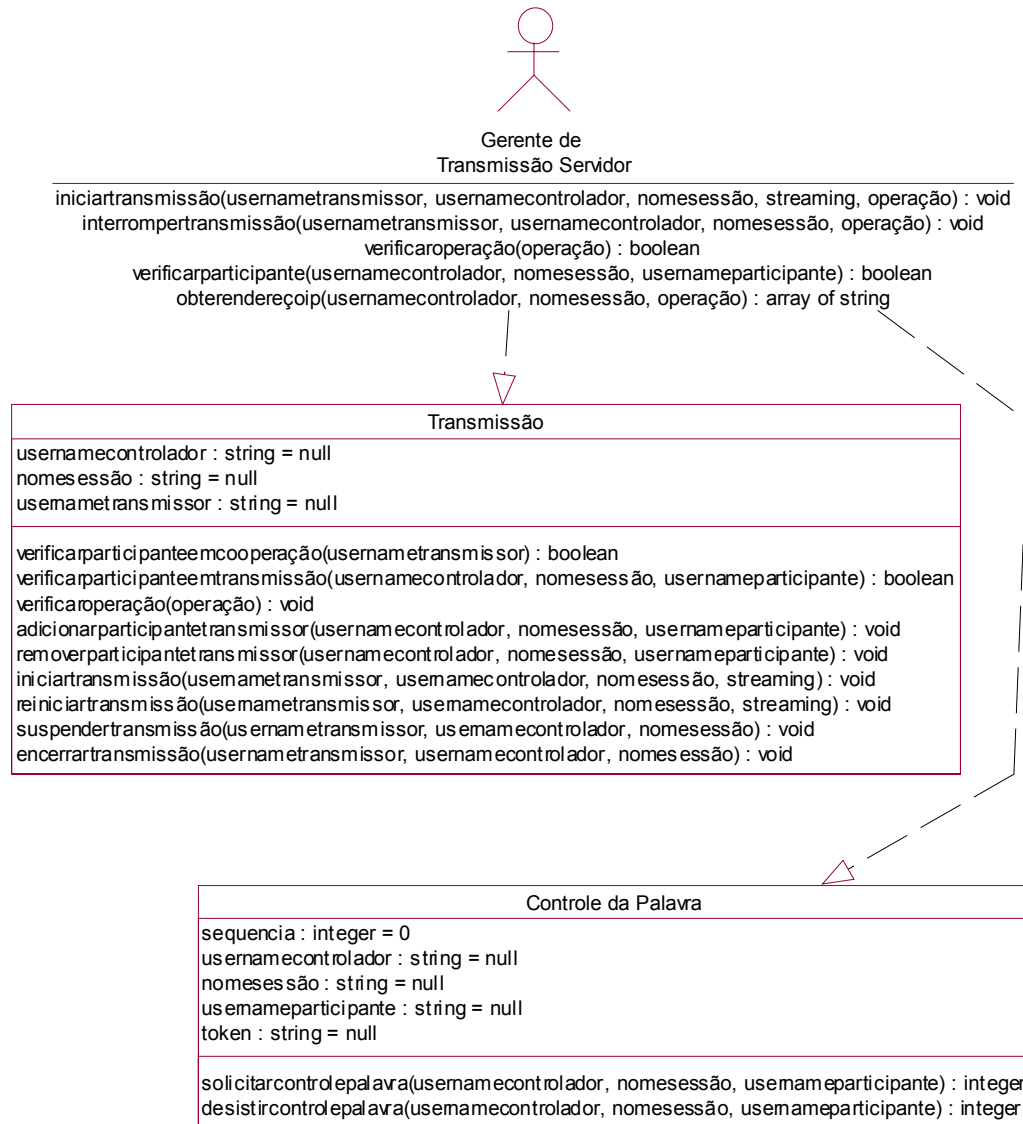


Figura 5.14. Classes e Métodos do Componente GTS

#### 5.4.6. Gerente de Votação Servidor (GVtS)

Diversas aplicações para internet têm utilizado a votação para auxiliar tanto as pesquisas de opinião como o conhecimento do perfil dos usuários. Neste sentido, um componente



que minimize a tarefa de projeto e desenvolvimento dessas atividades seria de grande utilidade aos projetistas. Assim, este componente é responsável por computar o voto dos participantes.

A tarefa deste componente consiste em permitir a elaboração de perguntas para serem posteriormente votadas, validar o voto e totalizar a quantidade de votos recebidos. A elaboração das perguntas é tarefa do controlador da sessão. Sendo de grande importância o resultado da pesquisa obtido, a validação é de grande necessidade e consiste em permitir que um participante vote somente uma única vez. Nesta validação o voto não será armazenado, mas sim a informação de que o participante já votou.

São métodos desse componente: `validarvoto`, `adicionarvoto`, `totalizarvoto`, além de `elaborarpesquisa`, `removerpesquisa` e `mostrar pesquisas elaboradas` que devem ser utilizados pelo controlador da sessão. A figura 5.15 mostra as classes e os métodos desse componente.

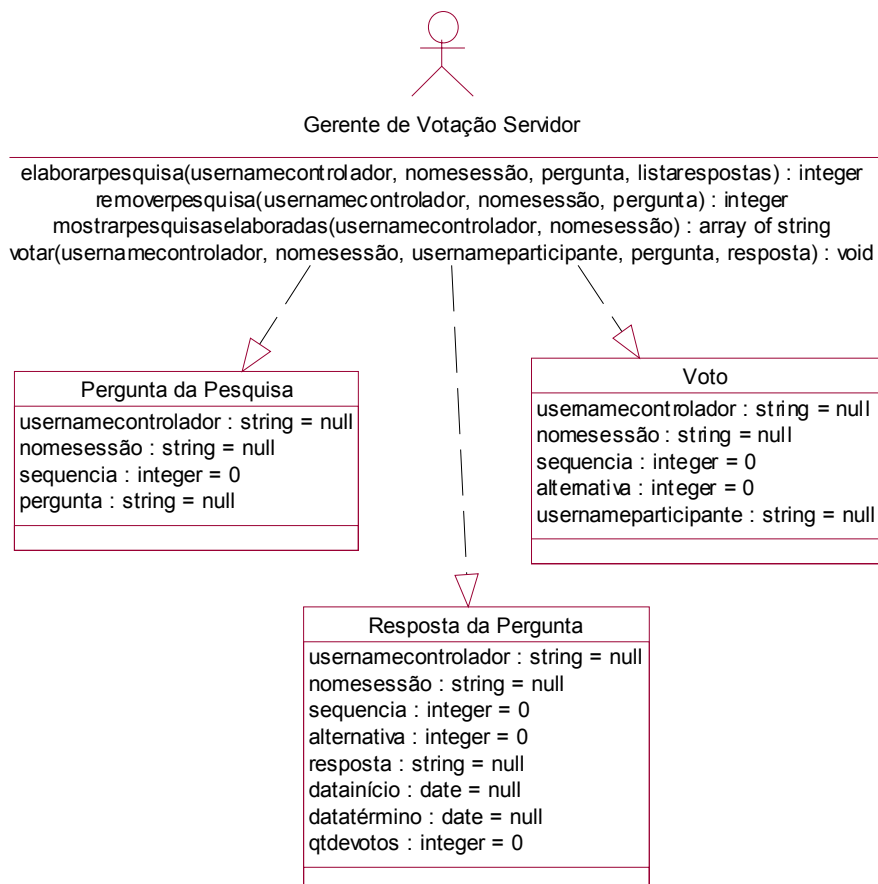


Figura 5.15. Classes e Métodos do Componente GVtS

#### 5.4.7. Gerente de Anotação Servidor (GAnS)

Atualmente grande quantidade das atividades dos indivíduos para execução de uma tarefa envolve um grupo de pessoas. Entretanto, na maioria das vezes, as pessoas se reúnem, mas a tecnologia, tanto hardware quanto software, não satisfaz as necessidades de interação entre as pessoas envolvidas na realização de tal tarefa. Até mesmo uma simples aula tem as anotações dos alunos compartilhadas entre os demais. Visando auxiliar essa situação, um componente que permita realizar anotações durante uma cooperação, para que a mesma possa ser compartilhada ou, posteriormente, enviada aos membros, seria um grande avanço no sentido de reduzir e facilitar a tarefa dos projetistas e desenvolvedores.

A complexidade de desenvolvimento de tal componente, que teria a mesma função de editores multiusuários, é extremamente elevada e muitos trabalhos foram realizados nessa área. Assim, a função do componente Gerente de Anotação Servidor é ativar um editor multiusuário qualquer já desenvolvido e que possua o melhor desempenho, grande facilidade de utilização e compatibilidade com os componentes deste projeto. O estudo para escolha de tal editor não foi realizado neste trabalho.

Dentre os métodos desse componente encontra-se: `ativارانotação` e `desativارانotação`, apresentados na figura 5.16.

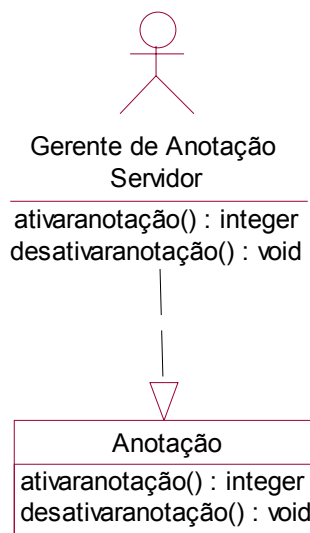


Figura 5.16. Classes e Métodos do Componente GAnS

#### 5.4.8. Gerente de Transferência de Arquivos Servidor (GTAS)

Dentre as atividades constantemente realizadas pelos usuários localizados em locais geograficamente separados encontra-se a transferência de arquivos, o que justifica a importância de um componente que realize tal atividade.

O Gerente de Transferência de Arquivos Servidor deve permitir que os arquivos sejam visualizados, selecionados e transferidos para outros participantes. Assim, são métodos desse componente: `mostrararquivos` e `transferirarquivos`. A figura 5.17 apresenta as classes e métodos desse componente.

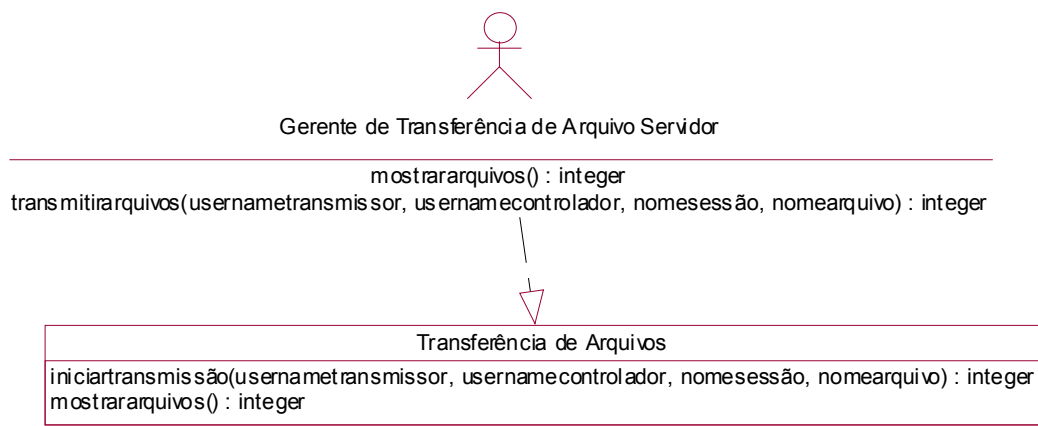


Figura 5.17. Classes e Métodos do Componente GTAS

#### 5.4.9. Gerente de Gravação de Conferência Servidor (GGCS)

Deseja-se que as conferências, além de transmitidas, sejam também gravadas. Tal é a função do Gerente de Gravação de Conferência Servidor. Este componente possui mecanismos para iniciar e encerrar a gravação da conferência. A figura 5.18 mostra os métodos e classes desse componente.

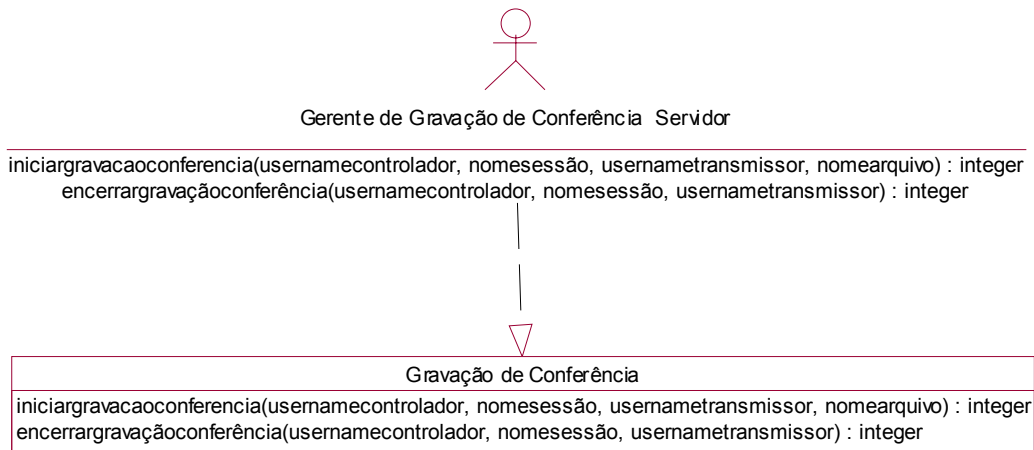


Figura 5.18. Classes e Métodos do Componente GGCS

#### 5.4.10. Gerente de Visualização Servidor (GVS)

O componente Gerente de Visualização Servidor é o responsável por apresentar todas as informações armazenadas no banco de dados aos participantes. As informações a serem apresentadas pertencem a duas categorias: (1) informações sobre as sessões e (2) informações sobre os participantes.

A categoria de informações sobre as sessões deve apresentar uma lista de sessões relacionadas a um participante. Neste sentido, esta categoria está dividida em quatro classes, a saber: sessões de que o participante faz parte, sessões de que o participante não faz parte, sessões controladas pelo participante e sessões que o participante faz parte e que estão em cooperação.

A categoria de informações sobre os participantes foi subdividida em três classes. Uma deve apresentar todos os participantes que são membros de uma sessão; outra deve apresentar todos os participantes que estão em cooperação e a última deve apresentar quem são os participantes que estão transmitindo áudio/vídeo durante uma cooperação.

São métodos deste componente: `mostrarsessões`, `obtersessõesporparticipante`, `obtersessõesnãopertence`, `obtersessõescontroladas`, `obtersessõessemcooperação`, `mostrarparticipantes`, `obterparticipantesporsessão`, `obterparticipantessemcooperação` e `obterparticipantesemtransmissão`. A figura 5.19 mostra as classes e os métodos deste componente.

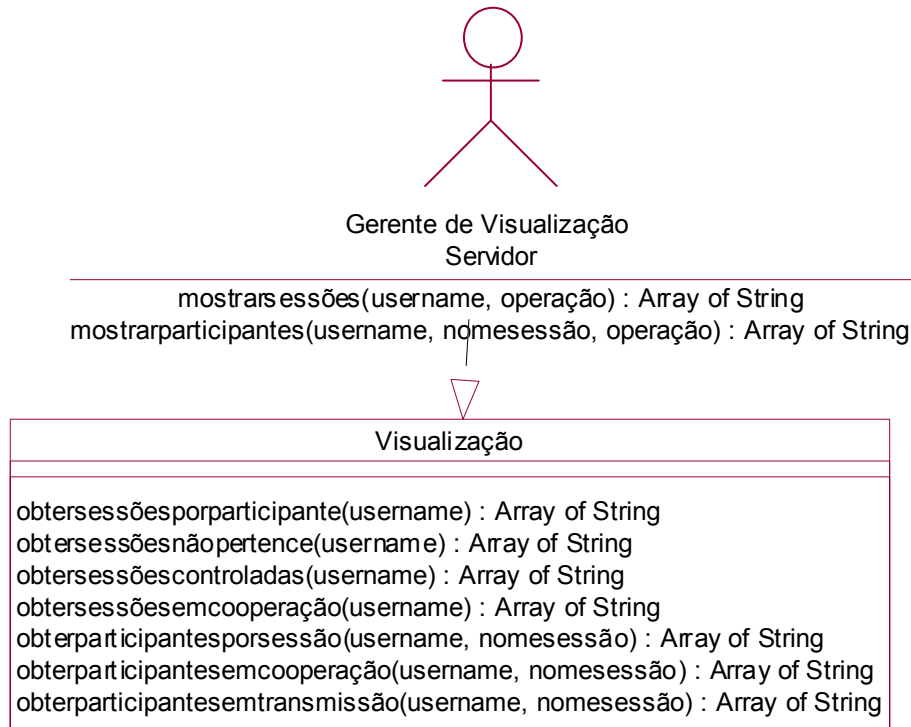


Figura 5.19. Classes e Métodos do Componente GVS

### 5.5. Notificação de Eventos

Uma característica essencial das aplicações cooperativas é a habilidade de uma pessoa ver os efeitos das ações de outras. Nestes sistemas, as interfaces dos usuários devem ser mantidas consistentes para parecer que estes estão trabalhando conjuntamente. Por exemplo, usuários de editores compartilhados devem ser capazes de visualizar as alterações e ações realizadas pelos co-editores. A importância dessa habilidade se justifica pelos seguintes aspectos:

- a) funcional, pois cada pessoa acessa uma versão atualizada dos trabalhos realizados pelos envolvidos na tarefa cooperativa, facilitando e agilizando o trabalho em grupo;
- b) coordenativo, pois previne atualizações inconsistentes das informações, articulando e ordenando as relações dos indivíduos com grande exatidão;
- c) informativo, pois suporta notificações dos indivíduos durante a realização do trabalho em grupo.

---

Cada aplicação que atualiza dados ou ambientes compartilhados é responsável pela notificação sobre a ocorrência da alteração e, conseqüentemente, pela transmissão da alteração a todas as partes interessadas. Normalmente a técnica para manutenção da consistência é específica à aplicação a ser implementada. Assim, cada nova aplicação cria alguns mecanismos para alcançar a consistência da informação e do ambiente compartilhado.

Neste sentido, a notificação é alvo de muitos estudos e interessantes pesquisas e revela uma quantidade de modelos que provêm a notificação sobre as atividades dos participantes em um ambiente cooperativo. Dois tipos de abordagens para suportar a notificação em ambientes cooperativos podem ser encontrados: notificação social e orientada à tarefa. Notificação orientada à tarefa é voltada para as atividades realizadas para obter uma tarefa compartilhada. Este tipo pode ser obtido através da informação sobre o estado de certos documentos ou ambiente de trabalho compartilhados, permitindo aos usuários coordenar suas atividades. Notificação social inclui informações sobre a presença e atividades de pessoas em um ambiente compartilhado, informando aos usuários quando os dados foram atualizados ou quais pessoas estão envolvidas na realização de um trabalho em grupo.

Vários trabalhos foram desenvolvidos, visando a prover serviços de notificação tais como Notification Service Transfer Protocol (NSTP) [16] e NESSIE [17]. CORBA (Common Object Request Broker Architecture) [14] também oferece serviço de notificação de eventos. CORBA é a tecnologia ORB (Object Request Broker) adotada pelo OMG, e define uma estrutura para que diferentes implementações de ORB possam prover serviços e interfaces comuns para suportar clientes e implementações de objetos portáteis. ORB, de acordo com a definição do OMG, provê mecanismos através dos quais os objetos fazem pedidos e recebem respostas de forma transparente e provê interoperabilidade entre aplicações em diferentes máquinas em ambientes distribuídos e heterogêneos.

Um componente para a notificação de eventos não será desenvolvido neste projeto. É tarefa de implementação avaliar os serviços de notificação de eventos existentes a fim de se escolher aquele que apresentar um melhor desempenho e maior facilidade de uso.

## Capítulo VI

### Conclusão

Diante do crescimento da área de Trabalho Cooperativo Suportado por Computador (Computer Supported Cooperative Work – CSCW) é de grande importância que características como agilidade, generalidade, flexibilidade, reuso e confiabilidade estejam presentes no desenvolvimento das aplicações cooperativas. Essas características podem ser obtidas através da utilização de toolkits e de uma metodologia para o desenvolvimento de software baseado em componentes. Estes oferecem funcionalidades únicas e pré-projetadas e proporcionam certa comodidade para que os projetistas se preocupem somente com aspectos específicos da aplicação cooperativa.

Este trabalho apresentou um conjunto de componentes para aplicações cooperativas, identificado a partir da utilização de uma metodologia para desenvolvimento de software baseado em componentes. Esta metodologia foi aplicada sobre um sistema de conferência multimídia com o objetivo de descobrir quais partes da aplicação cooperativa poderiam ser transformadas em blocos de programa pré-projetados, auxiliando no desenvolvimento de uma grande variedade de aplicações. A metodologia orientou o processo de desenvolvimento de software baseado em componentes, pois dividiu esse processo em quatro níveis e definiu passos a serem realizados em cada nível. Neste sentido, uma visão estrutural e comportamental da aplicação cooperativa foi obtida e representada graficamente, possibilitando que futuras extensões das funcionalidades dos serviços providos pelos componentes sejam realizadas de forma clara e simples.

Como exemplo de outras metodologias para o desenvolvimento de software baseado em componentes destaca-se Unified Process [appud 7] e Catalysis [appud 7, 3]. O Unified Process não é realmente um processo para desenvolvimento de software e sim um framework baseado em UML que descreve as melhores práticas para o desenvolvimento de software. Unified Process identifica duas dimensões: tempo (ciclo, fases e repetições) e fluxo de trabalho, onde cada fluxo de trabalho captura um conjunto de atividades e instrumentos (domínio, caso de uso, análise, projeto etc) necessários para realização das atividades. Catalysis é outro processo para desenvolvimento de software

baseado em UML. Esta abordagem se aproxima mais de um modelo de processo que pode ser adaptável a cada situação, baseando-se em três conceitos (tipo, colaboração e refinamento) e em frameworks. Catalysis também divide o processo de desenvolvimento de software em três níveis: negócio, componente ou especificação do sistema e implementação do componente. A especificação do componente descreve o comportamento visível externamente, enquanto a implementação do componente descreve seu comportamento e suas estruturas internas. A metodologia estudada neste projeto não é tão genérica e completa quanto Unified Process e Catalysis; porém, a simplicidade relativa a esta metodologia constitui seu principal benefício. A metodologia na qual se baseia este estudo é a escolha mais razoável e pragmática para um processo de desenvolvimento de groupware baseado em componentes, pois considera o uso de componentes explicitamente.

Alguns benefícios e deficiências da utilização desta metodologia foram observados e estão apresentados a seguir:

1) Análise Global da Metodologia:

a) Benefícios

- A divisão da metodologia nos quatro níveis de abstração (empresa, sistema, componente e objeto) representa suficientemente as várias etapas para identificação e desenvolvimento de componentes de software para aplicações cooperativas. Tal representação abrange desde a obtenção do conhecimento sobre diversas aplicações CSCW (permitindo a identificação de elementos genéricos) até a implementação dos componentes;
- A criação das visões (estrutural, comportamental e interacional) proporciona tanto a representação/visualização da estrutura dos componentes quanto o comportamento dos mesmos em cada fase de abstração;
- A utilização da metodologia UML facilita a representação dos níveis/visões propostos pela metodologia;
- A partir do nível de sistema a modelagem define que cada nível seguinte pode ser obtido a partir de refinamentos sucessivos do nível anterior, permitindo o detalhamento até a implementação do componente;
- A metodologia se apresenta extremamente simples quando comparada com outras;



---

b) Deficiências

- Com a utilização da metodologia não foi possível permitir a utilização de um protocolo aberto para geração do código dos componentes;
- Dificuldade em se separar a aplicação groupware final do componente genérico a ser identificado por se basear em um estudo de caso como forma de obtenção da generalidade.

2) Nível de Empresa

a) Benefícios

- Os elementos básicos propostos pelo modelo conceitual [6], a saber: atividade, ator, informação e serviço representam de forma clara e objetiva o conhecimento sobre o domínio da aplicação CSCW a ser desenvolvida;
- O modelo conceitual também permite a criação de outros elementos considerados necessários;

b) Deficiências

- A representação apresenta uma deficiência visual por não haver uma diferença entre atividade, sub-atividade, ação, informação e serviço, tornando a representação um tanto que confusa;
- Uma ferramenta automatizada para montagem do modelo conceitual ajudaria consideravelmente a elaboração da representação. Tal ferramenta deveria interligar diagrama de objeto, diagrama de atividade e glossário;
- Na representação do diagrama de atividades a falta de uma diferenciação na representação da atividade a ser escolhida pelo participante e da atividade a ser disparada internamente pelo sistema dificulta o entendimento do diagrama.

3) Nível de Sistema

a) Benefícios

- O mapeamento dos elementos identificados no nível de empresa facilita a elaboração da visão estrutural do diagrama de casos de uso, no nível de sistema;

#### 4) Nível de Componente

##### a) Deficiências

- Infelizmente a identificação dos componentes depende da subjetividade e experiência do projetista, o qual deve reconhecê-los através da busca de características desejáveis aos componentes e das proximidades dos casos de uso representados pelo diagrama de casos de uso;
- Utilizando-se a arquitetura cliente servidor, falta de uma representação para apresentar quais métodos do componente cliente se comunicam com os métodos do componente servidor.

#### 5) Nível de Objeto

##### a) Benefícios

- Este nível muito de aproxima da implementação, bastando a transformação da representação em comandos da linguagem de programação a ser utilizada;

O toolkit resultante deste projeto utiliza uma arquitetura cliente-servidor e oferece serviços genéricos que podem ser estendidos para implementar novas funcionalidades. É formado por componentes cliente (gerente de aplicação, gerente de aplicação cliente, gerente de acesso, gerente de sessão, gerente de participação, gerente de cooperação e gerente de visualização) e componentes servidor (gerente de aplicação servidor, gerente de acesso, gerente de sessão, gerente de inscrição, gerente de correio assíncrono, gerente de associação, gerente de comunicação, gerente de mídia, gerente de correio síncrono, gerente de recepção, gerente de transmissão e gerente de visualização).

A implementação dos componentes identificados para o sistema de conferência multimídia está em andamento. Esta implementação baseia-se na utilização da plataforma Java 2 Enterprise Edition (J2EE), que define um padrão para desenvolvimento de aplicações comerciais multi-camadas [19]; Enterprise Java Beans (EJB), que é uma arquitetura para computação distribuída baseada em componentes [18] e Java Media Framework (JMF), que é uma API (Application Programming Interface – Interface para Programação de Aplicação) para incorporar mídia temporal em applets e aplicações Java [20]. A implementação dos componentes para o envio e recepção de mídia faz parte de um projeto de iniciação científica.

---

O toolkit proposto neste trabalho é atraente por utilizar uma plataforma de implementação portátil a vários sistemas operacionais. Também, abrange uma quantidade de funcionalidade desejável para aplicações cooperativas quando comparado com outros toolkits. Analisando-se alguns toolkits observa-se:

- COAST (Cooperative Application Systems Toolkit) [17] é um toolkit que oferece aos desenvolvedores arquitetura e classes para aplicações cooperativas que podem ser usadas para implementar tais aplicações. COAST utiliza arquitetura replicada, dificultando a manutenção das aplicações; provê componentes genéricos (gerente de sessão, gerente de replicação e gerente de transação), classes abstratas (visualização, controladores, objetos replicados e sessões) que podem ser refinadas para implementar aplicações específicas e é implementado em VisualWorks Smalltalk. Este toolkit não possui componentes para o gerenciamento de envio e recepção de mídia, assim como não permite a troca de mensagens entre os participantes da sessão;
- Rendezvous [appud 3, appud 8] e Suite [appud 3] permitem um fácil projeto de aplicações cooperativas, mas não oferecem informações sobre as ações dos participantes durante uma cooperação;
- GroupKit [appud8, 16] requer um grande esforço para especificar o comportamento do sistema, que é feito através da definição de políticas; enquanto TAC apresenta características pré-definidas e projetadas nos seus componentes.

Neste sentido, o TAC apresenta vantagens sobre os demais por concentrar várias funcionalidades de aplicações cooperativas em um só toolkit, oferecendo componentes para tornar a sessão persistente, para o convite de participantes, para o envio de mensagens assíncronas (utilizando-se de um serviço de correio eletrônico externo), para suporte a multimídia (áudio e vídeo), para a notificação das ações dos participantes e para a comunicação síncrona entre os participantes durante uma cooperação.

Como trabalhos futuros deseja-se que uma nova geração desses toolkits seja criada para permitir a utilização de um protocolo aberto. Tal protocolo permitirá que aspectos comportamentais dos componentes sejam encapsulados através da definição de políticas. Um novo componente para permitir a realização de anotações durante a cooperação também enriquecerá o toolkit; assim como uma nova versão dos

---

componentes sessão e cooperação a fim de acrescentar dois novos serviços, respectivamente: agendar futuros compromissos cooperativos e armazenar/recuperar mídias para que participantes ausentes tenham acesso à cooperação. Ao agendar os compromissos uma mensagem deve ser enviada aos participantes da sessão informando-os sobre a próxima reunião. Após a implementação desses componentes, espera-se que aplicações cooperativas sejam projetadas utilizando-se os mesmos para se validar sua generalidade. Dentre as possíveis aplicações que poderão ser desenvolvidas destaca-se uma aplicação de ensino à distância.

## Referências Bibliográficas

1. Banavar, G., Doddapaneti, S., Miller, K. and Mukherjee, B.: Rapidly Building Synchronous Collaborative Applications by Direct Manipulation. *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work (CSCW'98)*, pp. 139-148, 1998.
2. Booch, G., Rumbaugh, J. and Jacobson, I.: *Unified Modeling Language User Guide*, MA: Addison-Wesley, 1998.
3. Dourish, P.: *Open Implementation and Flexibility in CSCW Toolkits*. PhD Thesis, Department of Computer Science, University College London, London, UK, 1996.
4. Dourish, P.: Using Metalevel Techniques in a Flexible Toolkit for CSCW Applications. *ACM Transactions on Computer-Human Interaction*, Vol. 5, N° 2, pp. 109-155, 1998.
5. Ellis, C.A., Gibbs, S.J. and Rein, G.L.: Groupware: Some issues and experiences. *Communications of the ACM*, 34 (1), pp. 38-58, 1991.
6. Farias, C. R. G., Ferreira Pires, L., and van Sinderen, M.: A conceptual model for the development of CSCW systems. *Fourth International Conference on the Design of Cooperative Systems (COOP'2000)*, Sophia Antipolis (France), May/2000.
7. Farias, C. R. G., van Sinderen, M. and Ferreira Pires. A systematic approach for component-based software development. *Proceedings of the Seventh European Concurrent Engineering Conference (ECEC'2000)*, Leicester (United Kingdom), pp. 127-131, April/2000.

8. Greenberg, S. and Roseman, M.: Groupware Toolkits for Synchronous Work. In M. Beaudouin-Lafon (editor), *Computer-Supported Cooperative Work*, Trends in Software Series, Chapter 6, p135-168, John Wiley & Sons Ltd, 1998.
9. Haake, J.M. and Wilson, B.: Supporting collaborative writing of hyperdocuments in SEPIA. *Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, pp. 138-146, 1992.
10. Jablonski, S.: MOBILE: A Modular Workflow Model and Architecture. *Proceedings of the International Working Conference on Dynamic Modelling and Information Systems*, pp. 1-30, 1994.
11. Marx, M. and Schmandt, C.: CLUES: dynamic personalized message filtering. *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work (CSCW'96)*, pp. 113-121, 1996.
12. Medina-Mora, R., Winograd, T., Flores, R. and Flores, F.: The action workflow approach to workflow management technology. *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW'92)*, pp. 281-288. 1992
13. OMG. *Unified Modeling Language 1.3 Specification*. Available at <http://cgi.omg.org/cgi-bin/doc?ad/99-06-08>.
14. OMG. CORBA. Available at <http://www.omg.org/technology/documents/formal/>.
15. Pacull, F., Sandoz, A. and Schiper, A.: Duplex: a distributed collaborative editing environment in large scale. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, pp. 165-173, 1994.

16. Patterson, J.F., Day, M. and Kucan, J., Notification servers for synchronous groupware. *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work (CSCW'96)*, pp. 122-129, 1996.
17. Prinz, W., NESSIE: An Awareness Environment for Cooperative Settings. *Proceedings of the Sixtn European Conference on Computer Supported Cooperative Work (ECSCW'99)*, pp. 391-410, 1999.
18. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, pp. 175-186, 1994.
19. Roseman, M. amd Greenberg, S.: Building Real Time Groupware with GroupKit, A Groupware Toolkit, *ACM Transactions on Computer Human Interaction*, 3(1), pp. 99-106, 1996.
20. Schuckmann, C., Kirchner, L., Schümmer, J. and Haake, J. M.: Designing object-oriented synchronous groupware with COAST. *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, pp. 30-38, 1996.
21. Sun Microsystems. *Enterprise Java Beans Specification*. Available at <http://java.sun.com/products/ejb/docs.html>.
22. Sun Microsystems. *Java 2 Enterprise Edition (J2EE)*. Available at <http://java.sun.com/j2ee/j2sdkee/techdocs/guides/ejb/html/TOC.html>.
23. Sun Microsystems. *Java™ Media Framework API Guide*. Available at <http://java.sun.com/products/java-media/jmf/2.0/jmf20-fcs-guide/JMFTOC.html>

24. Szyperski, C.: *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley, 1998.



## Apêndice 1

### Sistema de Conferência Multimídia – Requisitos

Este apêndice apresenta as listas a serem atualizadas conforme as funcionalidades do Sistema de Conferência Multimídia (SCM) e as regras a serem obedecidas para garantir essas funcionalidades – tabela 1 e 2 respectivamente.

| <b>Lista</b>                   | <b>Funcionalidade</b>   | <b>Operação</b>   |
|--------------------------------|---|---|
| Pessoas Registradas no Sistema | Registrar   | Adicionar à Lista   |
|                                | Remover de Registro   | Remover da Lista  |
| Participantes Ativos           | Registrar   | Adicionar à Lista   |
|                                | Login   | Adicionar à Lista   |
|                                | Logoff  | Remover da Lista  |
| Sessão / Horários / Mídias     | Criar Sessão  | Adicionar à Lista   |
|                                | Bloquear Sessão (Remover)   | Remover da Lista  |
|                                | Bloquear Sessão (Encerrar)  | Atualizar Lista   |
|                                | Atualizar Sessão (Atualizar)  | Atualizar Lista   |
| Participantes por Sessão       | Criar Sessão  | Adicionar à Lista   |
|                                | Bloquear Sessão (Remover)   | Remover da Lista  |
|                                | Remover Participação em uma Sessão  | Remover da Lista  |
|                                | Analisar Inscrições (aceite)  | Adicionar à Lista   |
|                                | Confirmar Convite (quando o convite foi feito pelo controlador da sessão)     | Adicionar à Lista   |
| Controlador                    | Criar Sessão  | Adicionar à Lista   |
|                                | Transferir Direitos de Controlador  | Remover controlador antigo e Adicionar novo controlador à lista |
|                                | Bloquear Sessão (Remover)   | Remover da Lista  |
| Inscrições Pendentes           | Analisar Inscrições   | Remover da Lista  |
|                                | Solicitar Inscrição   | Adicionar à Lista   |
|                                | Confirmar Convite (quando o convite não foi feito pelo controlador da sessão) | Adicionar à Lista   |
|                                |   |   |
| Convidados                     | Convidar  | Adicionar à Lista   |
|                                | Confirmar Convite   | Remover da Lista  |
|                                | Recusar Convite   | Remover da Lista  |
| Participantes em Cooperação    | Estabelecer Sessão  | Adicionar à Lista   |
|                                | Abandonar Cooperação  | Remover da Lista  |
|                                | Terminar Cooperação   | Remover da Lista  |
|                                | Deixar Cooperação   | Remover da Lista  |
| Sessões em Cooperação          | Estabelecer Sessão (controlador)  | Adicionar à Lista   |

|   |  |                   |
|---|--|-------------------|
|   | Terminar Cooperação (controlador)              | Remover da Lista  |
| Mídia por Participante                  | Selecionar Mídia (escolher)                    | Adicionar à Lista |
|   | Selecionar Mídia (remover)                     | Remover da Lista  |
| Participantes Transmissores             | Transmitir Áudio/Vídeo (iniciar transmissão)   | Adicionar à Lista |
|   | Transmitir Áudio/Vídeo (terminar transmissão)  | Remover da Lista  |
|   | Transmitir Áudio/Vídeo (suspender transmissão) | Remover da Lista  |
|   | Transmitir Áudio/Vídeo (reiniciar transmissão) | Adicionar à Lista |
| Receptores por Participante Transmissor | Receber Áudio/Vídeo (iniciar recepção)         | Adicionar à Lista |
|   | Receber Áudio/Vídeo (terminar recepção)        | Remover da Lista  |
|   | Receber Áudio/Vídeo (suspender recepção)       | Remover da Lista  |
|   | Receber Áudio/Vídeo (reiniciar recepção)       | Adicionar à Lista |
| Frequência                              | Iniciar a Cooperação                           | Adicionar à Lista |
|   | Finalizar Cooperação                           | Atualizar a Lista |
| Informações sobre a Sessão              | Iniciar a Cooperação                           | Adicionar à Lista |
|   | Finalizar a Cooperação                         | Atualizar a Lista |
| Pesquisa                                | Elaborar Pesquisa                              | Adicionar à Lista |
|   | Votar  | Atualizar a Lista |

Tabela 1. Tabela de Listas a serem Utilizadas – SCM

| <b>Regras</b>         |  |  |  |
|-----------------------|--|--|--|
| <b>Funcionalidade</b> | <b>Parâmetros</b>  | <b>Regras</b>  | <b>Mensagens</b>   |
| Registrar             | <ul style="list-style-type: none"> <li>• Username</li> <li>• Password</li> <li>• Nome completo</li> <li>• Email</li> </ul> | <ul style="list-style-type: none"> <li>• Não deve existir um registro para o username informado</li> </ul>   | <ul style="list-style-type: none"> <li>• Username já utilizado por outro participante.</li> </ul>  |
| Remover Registro      | <ul style="list-style-type: none"> <li>• Username</li> <li>• Password</li> </ul>   | <ul style="list-style-type: none"> <li>• O username deve estar registrado no sistema</li> <li>• O username não deve estar conectado ao sistema</li> <li>• A password deve estar correta</li> </ul> | <ul style="list-style-type: none"> <li>• Username não registrado no sistema.</li> <li>• Remoção não permitida: username conectado ao sistema.</li> <li>• Senha Incorreta.</li> </ul> |
| Login                 | <ul style="list-style-type: none"> <li>• Username</li> <li>• Password</li> </ul>   | <ul style="list-style-type: none"> <li>• O username deve estar registrado no sistema</li> <li>• O username não deve estar conectado ao</li> </ul>  | <ul style="list-style-type: none"> <li>• Username não registrado no sistema.</li> <li>• Username já</li> </ul>   |

|                           |  |  |   |
|---------------------------|--|--|---|
|                           |  | <p>sistema</p> <ul style="list-style-type: none"> <li>• A password deve estar correta</li> </ul>   | <p>conectado ao sistema.</p> <ul style="list-style-type: none"> <li>• Senha Incorreta.</li> </ul>   |
| Logoff                    | <ul style="list-style-type: none"> <li>• Username</li> </ul>   |  |   |
| Criar Sessão              | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da Sessão</li> <li>• Finalidade da Sessão</li> <li>• Mídias disponíveis</li> <li>• Horário(s) da Cooperação</li> </ul> | <ul style="list-style-type: none"> <li>• Não pode existir uma sessão criada com o mesmo nome pelo mesmo participante</li> </ul>  | <ul style="list-style-type: none"> <li>• Já existe uma sessão com esse nome criada por este participante.</li> </ul>  |
| Atualizar Sessão          |  |  |   |
| 1. Incluir Horário        | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da Sessão</li> <li>• Horário da Cooperação</li> </ul>  | <ul style="list-style-type: none"> <li>• Não deve existir uma cooperação para esta sessão no mesmo horário que se deseja incluir</li> </ul>  | <ul style="list-style-type: none"> <li>• Horário já cadastrado para esta sessão.</li> </ul>   |
| 2. Alterar Horário        | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da Sessão</li> <li>• Horário Antigo</li> <li>• Novo Horário</li> </ul>   | <ul style="list-style-type: none"> <li>• Não deve estar havendo cooperação para o horário que se deseja alterar</li> <li>• Não deve ser alterado para um horário que já tenha uma cooperação prevista</li> </ul>                                 | <ul style="list-style-type: none"> <li>• Alteração não permitida: sessão em cooperação neste horário.</li> <li>• Alteração não permitida: horário já está cadastrado para esta sessão.</li> </ul>       |
| 3. Remover Horário        | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da Sessão</li> <li>• Horário</li> </ul>  | <ul style="list-style-type: none"> <li>• Não deve estar havendo cooperação para o horário que se deseja remover</li> </ul>   | <ul style="list-style-type: none"> <li>• Remoção não permitida: sessão em cooperação neste horário.</li> </ul>  |
| 4. Alterar Nome da Sessão | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da Sessão Antigo</li> <li>• Novo Nome da Sessão</li> </ul>   | <ul style="list-style-type: none"> <li>• Não deve existir uma sessão com o mesmo nome que já tenha sido criada anteriormente pelo solicitante</li> <li>• Não deve estar havendo cooperação para a sessão que se deseja alterar o nome</li> </ul> | <ul style="list-style-type: none"> <li>• Alteração não permitida: já existe uma sessão com esse nome criada por este participante.</li> <li>• Alteração não permitida: sessão em cooperação.</li> </ul> |
| 5. Alterar Finalidade da  | <ul style="list-style-type: none"> <li>• Username</li> <li>• Nome da</li> </ul>  |  |   |

| Sessão                             | Sessão  |  |   |
|------------------------------------|---|--|---|
| 6. Incluir Mídia Disponível        | <ul style="list-style-type: none"> <li>• Nova Finalidade da Sessão</li> <li>• Username da Sessão</li> <li>• Nome da Sessão</li> <li>• Mídia disponível</li> </ul> | <ul style="list-style-type: none"> <li>• Não deve incluir uma mídia que já esteja disponível</li> <li>• Não deve incluir uma mídia para uma sessão que esteja em cooperação</li> </ul>   | <ul style="list-style-type: none"> <li>• Mídia já está disponível para esta sessão</li> <li>• Inclusão não permitida: sessão em cooperação.</li> </ul>  |
| 7. Remover Mídia Disponível        | <ul style="list-style-type: none"> <li>• Username da Sessão</li> <li>• Nome da Sessão</li> <li>• Mídia</li> </ul>   | <ul style="list-style-type: none"> <li>• Não deve ser removida a mídia para uma sessão que esteja em cooperação</li> <li>• Não deve remover uma mídia que não esteja disponível</li> </ul>   | <ul style="list-style-type: none"> <li>• Remoção não permitida: sessão em cooperação.</li> <li>• Remoção não permitida: mídia não disponível para esta sessão</li> </ul>  |
| 8. Remover Sessão                  | <ul style="list-style-type: none"> <li>• Username da Sessão</li> <li>• Nome da Sessão</li> </ul>  | <ul style="list-style-type: none"> <li>• Não deve ser removida uma sessão que possua informação sobre a cooperação armazenada</li> <li>• Não deve ser removida uma sessão que esteja em cooperação</li> </ul>  | <ul style="list-style-type: none"> <li>• Remoção não permitida: sessão possui informações armazenadas.</li> <li>• Remoção não permitida: sessão em cooperação.</li> </ul>                                       |
| 9. Fechar Sessão                   | <ul style="list-style-type: none"> <li>• Username da Sessão</li> <li>• Nome da Sessão</li> <li>• Data de Fechamento</li> </ul>                                    | <ul style="list-style-type: none"> <li>• Não deve ser fechada uma sessão que possua horários de cooperação futuros</li> <li>• Não deve ser fechada uma sessão que esteja em cooperação</li> </ul>  | <ul style="list-style-type: none"> <li>• Sessão com horário de cooperação futuro já cadastrado.</li> <li>• Sessão em cooperação.</li> </ul>   |
| Transferir Direitos de Controlador | <ul style="list-style-type: none"> <li>• Username do Controlador Atual</li> <li>• Username do Novo Controlador</li> <li>• Nome da Sessão Controlada</li> </ul>    | <ul style="list-style-type: none"> <li>• Os usernames devem ser diferentes</li> <li>• O username do novo controlador deve estar registrado no sistema</li> <li>• O username do novo controlador deve pertencer à sessão controlada</li> <li>• A sessão não deve estar em cooperação</li> </ul> | <ul style="list-style-type: none"> <li>• Os usernames devem ser diferentes.</li> <li>• Username não registrado.</li> <li>• Novo controlador não faz parte da sessão.</li> <li>• Sessão em Cooperação</li> </ul> |
| Solicitar Inscrição                | <ul style="list-style-type: none"> <li>• Username do</li> </ul>   | <ul style="list-style-type: none"> <li>• O username não deve</li> </ul>  | <ul style="list-style-type: none"> <li>• Username já faz</li> </ul>   |

|                             |   |   |   |
|-----------------------------|---|---|---|
| na Sessão                   | <ul style="list-style-type: none"> <li>Solicitante</li> <li>• Username do Controlador</li> <li>• Nome da Sessão à qual deseja se inscrever</li> </ul>                         | participar da sessão à qual deseja se inscrever   | parte da sessão.  |
| Solicitar Remoção na Sessão | <ul style="list-style-type: none"> <li>• Username do Solicitante</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> </ul>                                      | <ul style="list-style-type: none"> <li>• O username não deve estar cooperando na sessão que não mais deseja participar</li> </ul>     | <ul style="list-style-type: none"> <li>• Remoção Inválida. Participante em cooperação.</li> </ul>   |
| Analisar Inscrição          |   |   |   |
| 1. Aceitar Inscrição        | <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Username do Solicitante</li> </ul>                                      | <ul style="list-style-type: none"> <li>• O username do solicitante deve ter uma inscrição</li> </ul>                                  | <ul style="list-style-type: none"> <li>• Solicitante não está inscrito</li> <li>• Correio Eletrônico enviado ao solicitante informando-o sobre a aceitação da inscrição.</li> </ul>       |
| 2. Recusar Inscrição        | <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Username do Solicitante</li> </ul>                                      | <ul style="list-style-type: none"> <li>• O username do solicitante deve ter uma inscrição</li> </ul>                                  | <ul style="list-style-type: none"> <li>• Solicitante não está inscrito</li> <li>• Correio Eletrônico enviado ao solicitante informando-o sobre a recusa da inscrição.</li> </ul>          |
| Convidar                    | <ul style="list-style-type: none"> <li>• Username de quem está convidando</li> <li>• Email do Convidado</li> <li>• Nome da Sessão</li> <li>• Controlador da Sessão</li> </ul> | <ul style="list-style-type: none"> <li>• O convidado não deve fazer parte da sessão para a qual estará recebendo o convite</li> </ul> | <ul style="list-style-type: none"> <li>• Convite Inválido. Participante já faz parte da sessão.</li> <li>• Correio Eletrônico enviado ao convidado informando-o sobre convite.</li> </ul> |
| Confirmar/Recusar Convite   | <ul style="list-style-type: none"> <li>• Username do Convidado</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> </ul>  | <ul style="list-style-type: none"> <li>• O username do convidado deve estar na lista de convidados para a sessão informada</li> </ul> | <ul style="list-style-type: none"> <li>• Participante não foi convidado para fazer parte desta sessão.</li> </ul>   |
| Estabelecer Sessão          | <ul style="list-style-type: none"> <li>• Username do</li> </ul>   | <ul style="list-style-type: none"> <li>• O início da cooperação</li> </ul>  | <ul style="list-style-type: none"> <li>• Cooperação não</li> </ul>  |

|                           |  |   |   |
|---------------------------|--|---|---|
|                           | Solicitante <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Horário de Acesso</li> </ul>   | só será permitido se o horário de acesso coincidir com o horário previsto para a cooperação <ul style="list-style-type: none"> <li>• Se o solicitante não for o controlador, só permitir a conexão se o controlador já estiver estabelecido a sessão</li> </ul> | prevista para este horário. <ul style="list-style-type: none"> <li>• Controlador não está em cooperação.</li> </ul>   |
| Finalizar Cooperação      |  |   |   |
| 1. abandonar a cooperação | <ul style="list-style-type: none"> <li>• Username do Solicitante</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Horário de Saída</li> </ul>   |   | <ul style="list-style-type: none"> <li>• Agradecendo a participação e motivos para o abandono.</li> </ul>   |
| 2. terminar a cooperação  | <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Horário de Término</li> </ul>  | <ul style="list-style-type: none"> <li>• O solicitante deve ser o controlador da sessão</li> </ul>  | <ul style="list-style-type: none"> <li>• Operação Inválida. Somente o controlador da sessão pode terminar a cooperação.</li> <li>• Agradecendo a participação.</li> </ul> |
| 3. deixar a cooperação    | <ul style="list-style-type: none"> <li>• Username do Solicitante</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Horário de Saída</li> </ul>   | <ul style="list-style-type: none"> <li>• Só será permitido se a cooperação foi terminada pelo controlador da sessão</li> </ul>  | <ul style="list-style-type: none"> <li>• Operação Inválida. cooperação não foi terminada pelo controlador da sessão.</li> </ul>   |
| Trocar Mensagens          |  |   |   |
| 1. Enviar Mensagem        | <ul style="list-style-type: none"> <li>• Username do Remetente</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Username do Destinatário</li> <li>• Mensagem</li> <li>• Horário da</li> </ul> | <ul style="list-style-type: none"> <li>• O destinatário deve ser qualquer participante que esteja em cooperação na sessão</li> </ul>  | <ul style="list-style-type: none"> <li>• Operação Inválida. Participante não está em cooperação.</li> </ul>   |

|                        |  |   |   |
|------------------------|--|---|---|
| 2. Receber Mensagem    | <p>Transmissão</p> <ul style="list-style-type: none"> <li>• Username de quem deseja receber a mensagem</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> </ul>   | <ul style="list-style-type: none"> <li>• Só serão recebidas as mensagens referentes à sessão com a qual se está cooperando</li> </ul> | <ul style="list-style-type: none"> <li>• Você tem mensagens novas.</li> </ul>                                     |
| Selecionar Mídia       |  |   |   |
| 1. Incluir             | <ul style="list-style-type: none"> <li>• Username do Solicitante do serviço</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Mídia selecionada</li> </ul> | <ul style="list-style-type: none"> <li>• A mídia selecionada para ser utilizada ainda não deve ter sido escolhida</li> </ul>          | <ul style="list-style-type: none"> <li>• Mídia Inválida. Esta mídia já está selecionada.</li> </ul>               |
| 2. Remover             | <ul style="list-style-type: none"> <li>• Username do Solicitante do serviço</li> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Mídia selecionada</li> </ul> | <ul style="list-style-type: none"> <li>• A mídia selecionada para ser utilizada deve ter sido escolhida anteriormente</li> </ul>      | <ul style="list-style-type: none"> <li>• Mídia Inválida. Esta mídia não foi selecionada.</li> </ul>               |
| Transmitir Áudio/Vídeo | <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Username do Transmissor</li> </ul>   |   |   |
| Receber Áudio/Vídeo    | <ul style="list-style-type: none"> <li>• Username do Controlador</li> <li>• Nome da Sessão</li> <li>• Username do Receptor</li> <li>• Username do Transmissor</li> </ul>         | <ul style="list-style-type: none"> <li>• O transmissor deve estar enviando áudio/vídeo</li> </ul>                                     | <ul style="list-style-type: none"> <li>• O participante selecionado não está transmitindo áudio/vídeo.</li> </ul> |

Tabela 2. Regras do SCM

## **Apêndice 2**

### ***Sistema de Conferência Multimídia – Nível de Empresa***

Este apêndice apresenta a modelagem do Sistema de Conferência Multimídia (SCM) para o nível de empresa. Este nível deve capturar o vocabulário e outras informações sobre o domínio da aplicação cooperativa a ser investigada. Os conceitos que devem ser identificados podem variar de acordo com o domínio da aplicação em desenvolvimento. Contudo, conceitos comuns à maioria das aplicações cooperativas são atividades, atores, regras (políticas), informação (recursos) e serviços de suporte.

Neste nível de abstração não há uma clara divisão entre as visões estrutural, comportamental e interacional, pois este é basicamente um nível conceitual. Assim este nível será apresentado sob uma única perspectiva ao longo das três visões.

A modelagem deste nível consiste, basicamente, na execução das seguintes etapas:

- 1) elaboração do diagrama de conceitos/objetos;
- 2) elaboração do diagrama de atividades;
- 3) elaboração do glossário de termos.

A seguir serão apresentadas cada uma das etapas para modelagem do SCM no nível de empresa.



**A2.1) Diagramas de Conceito – Diagramas de Objeto**

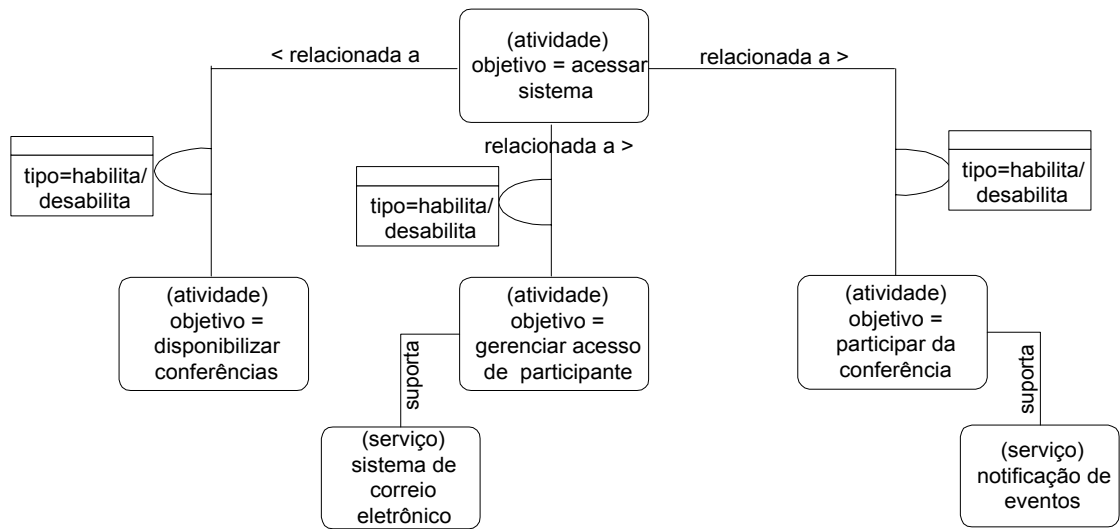


Figura 1. Diagrama de Objeto do Sistema de Conferência Multimídia

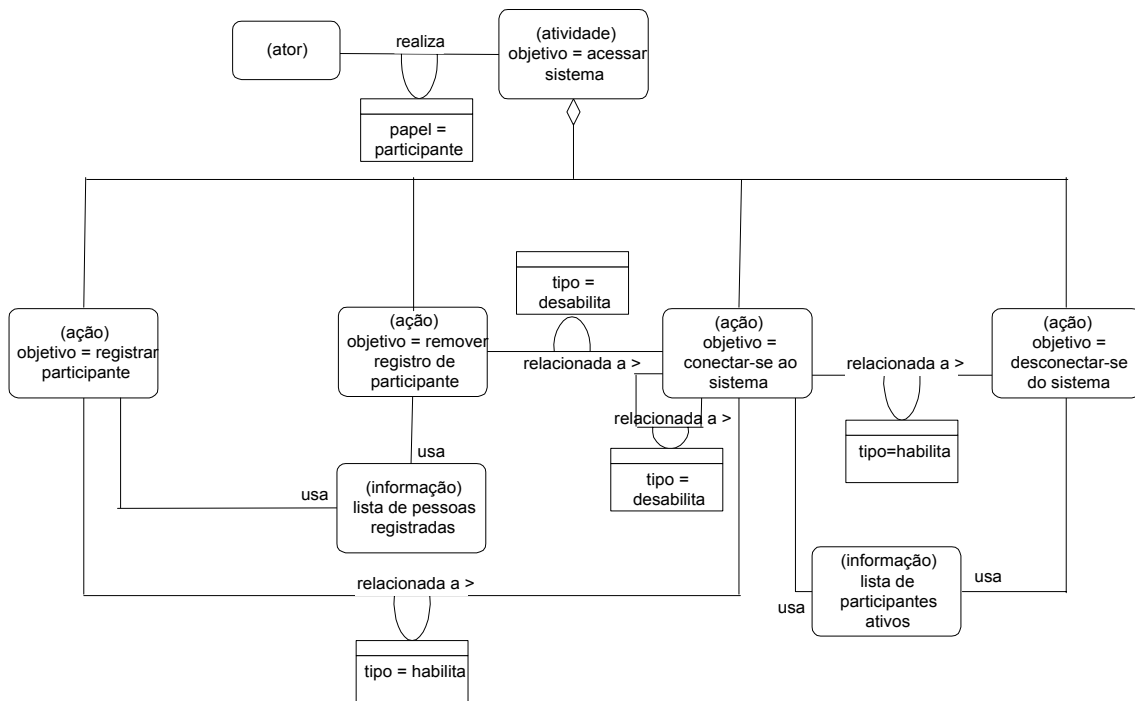


Figura 2. Diagrama de Objeto da Atividade Acessar Sistema

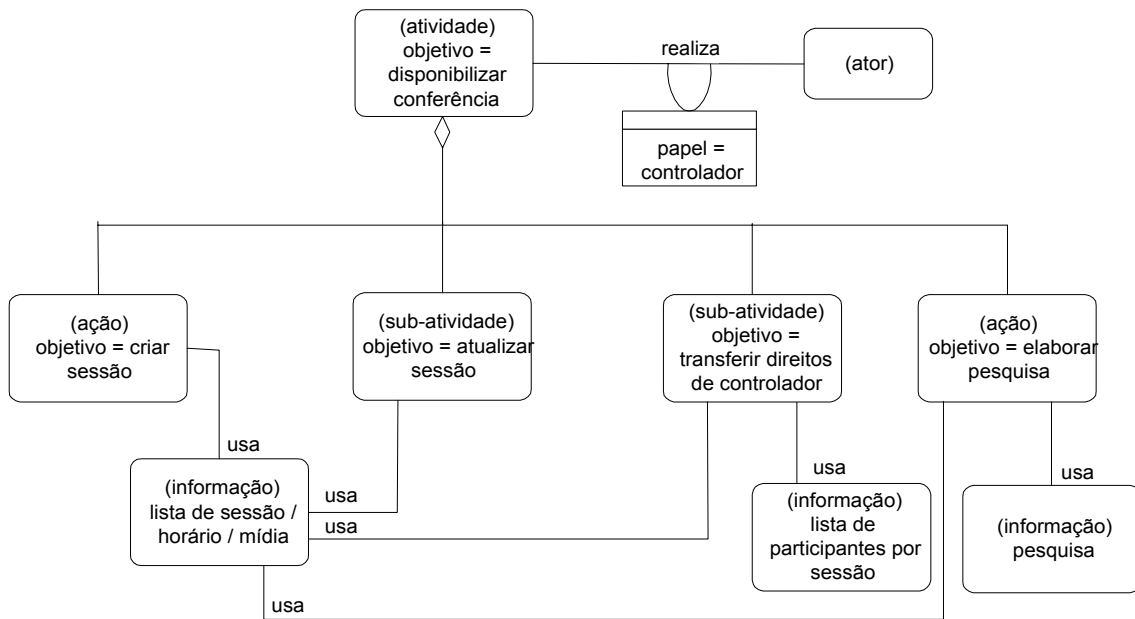


Figura 3. Diagrama de Objeto da Atividade Disponibilizar Conferência

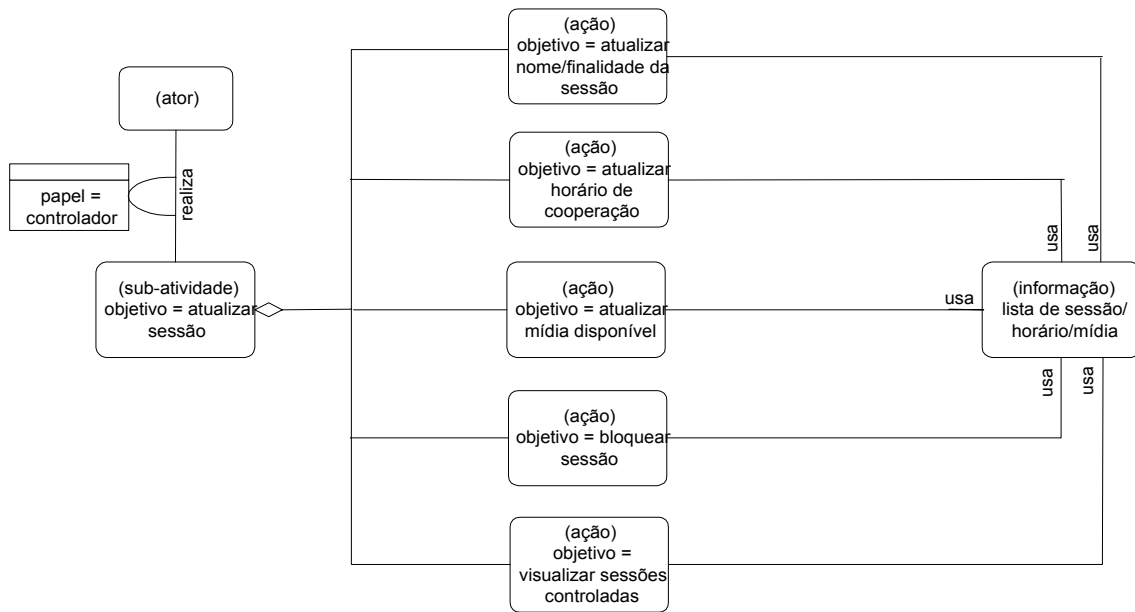


Figura 4. Diagrama de Objeto da Sub-Atividade Atualizar Sessão

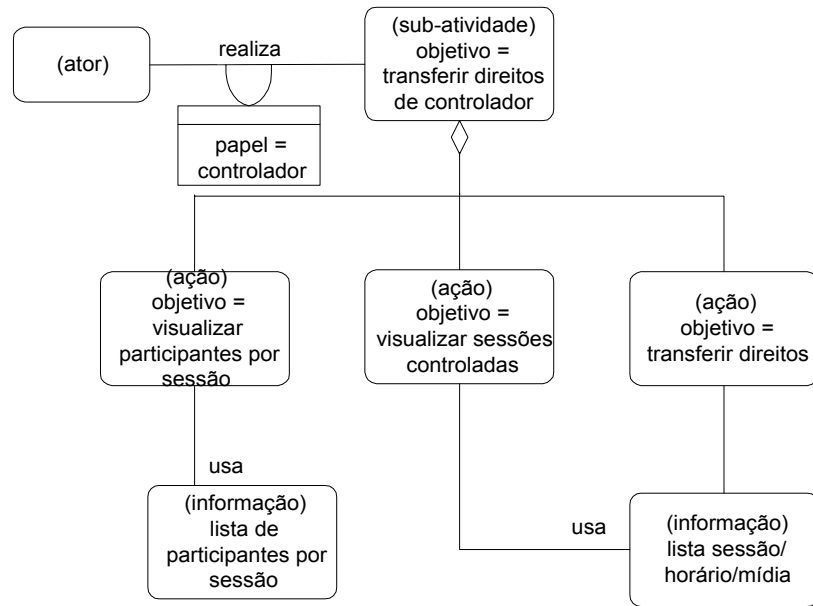


Figura 5. Diagrama de Objeto da Sub-Atividade Transferir Direitos de Controlador

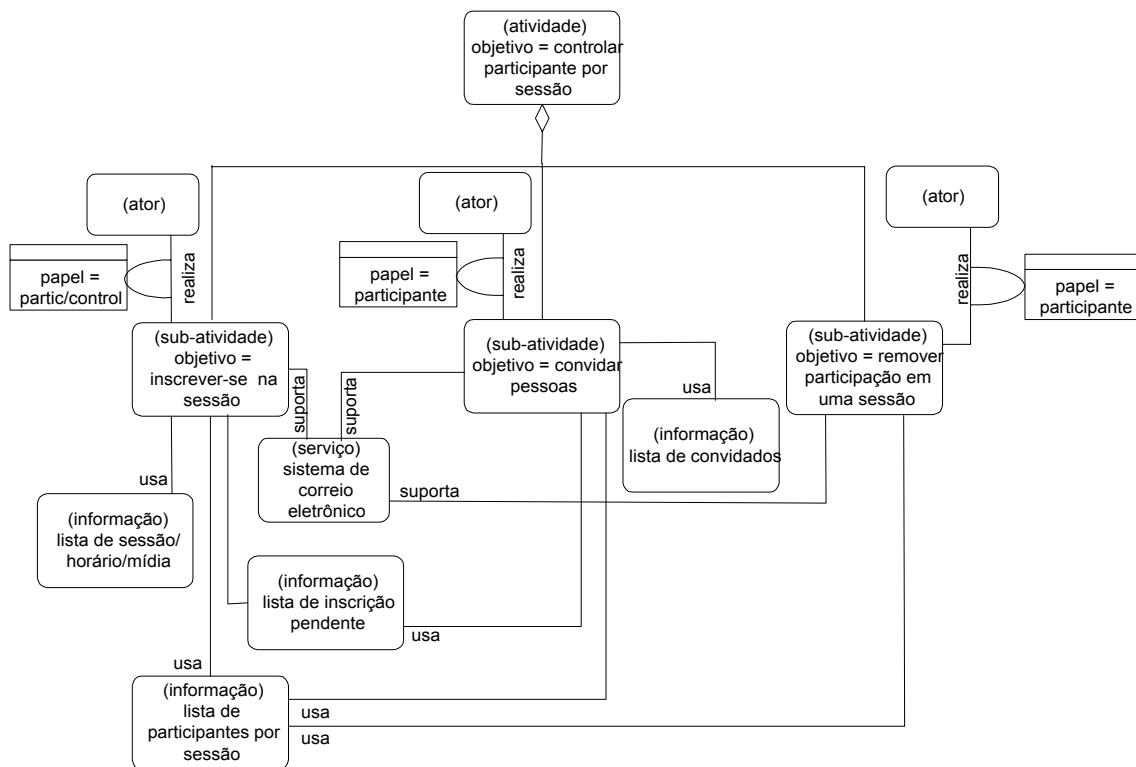


Figura 6. Diagrama de Objeto da Atividade Controlar Participante por Sessão

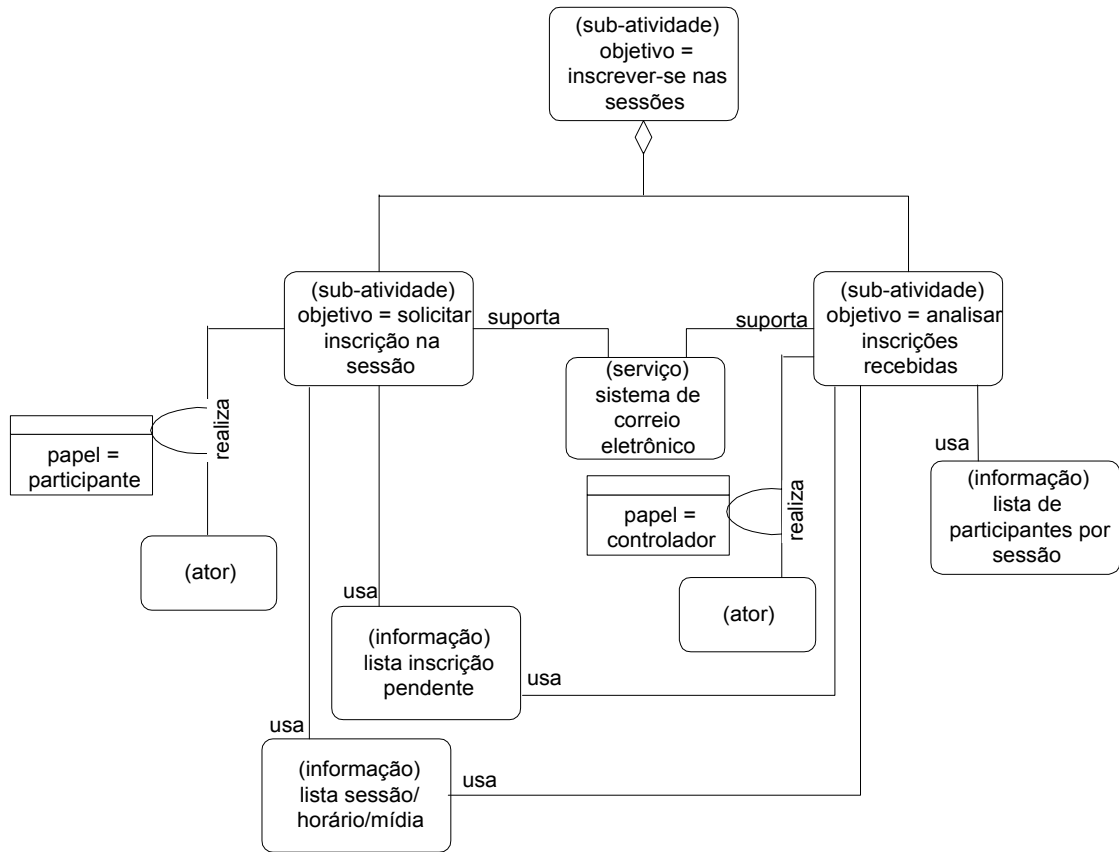


Figura 7. Diagrama de Objeto da Sub-Atividade Inscrever-se na Sessão

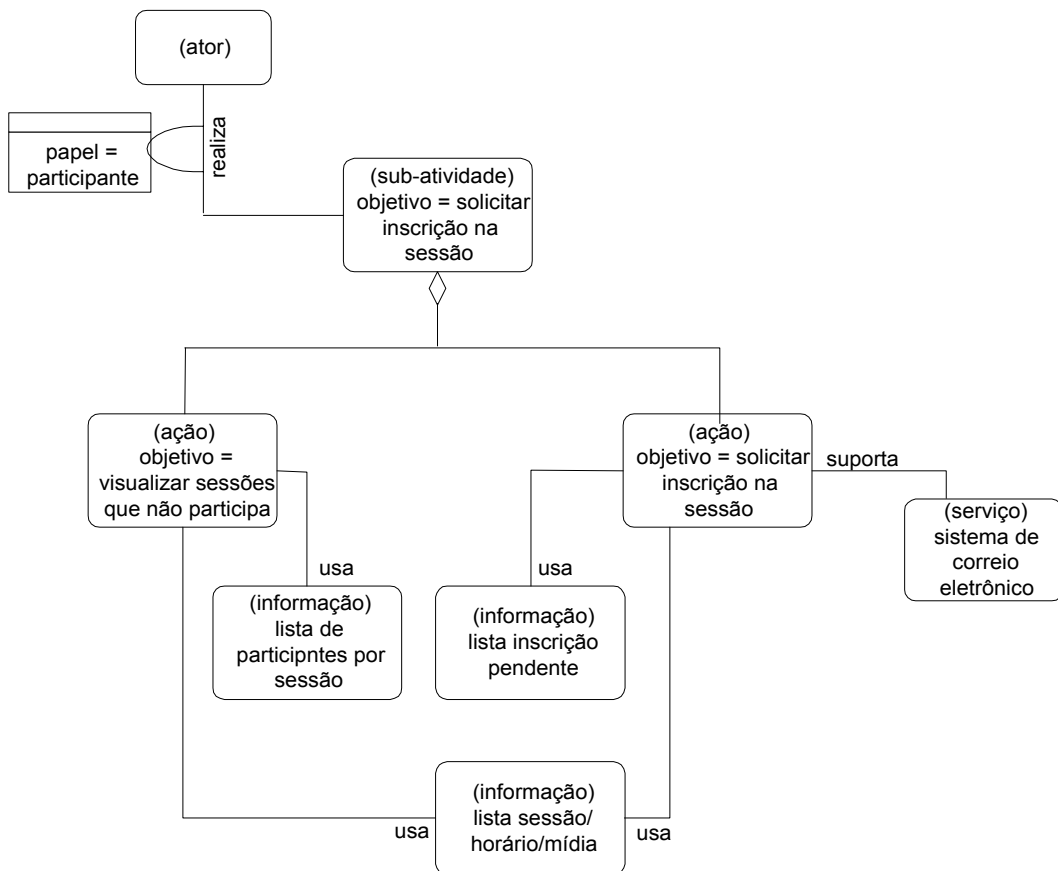


Figura 8. Diagrama de Objeto da Sub-Atividade Solicitar Inscrição na Sessão

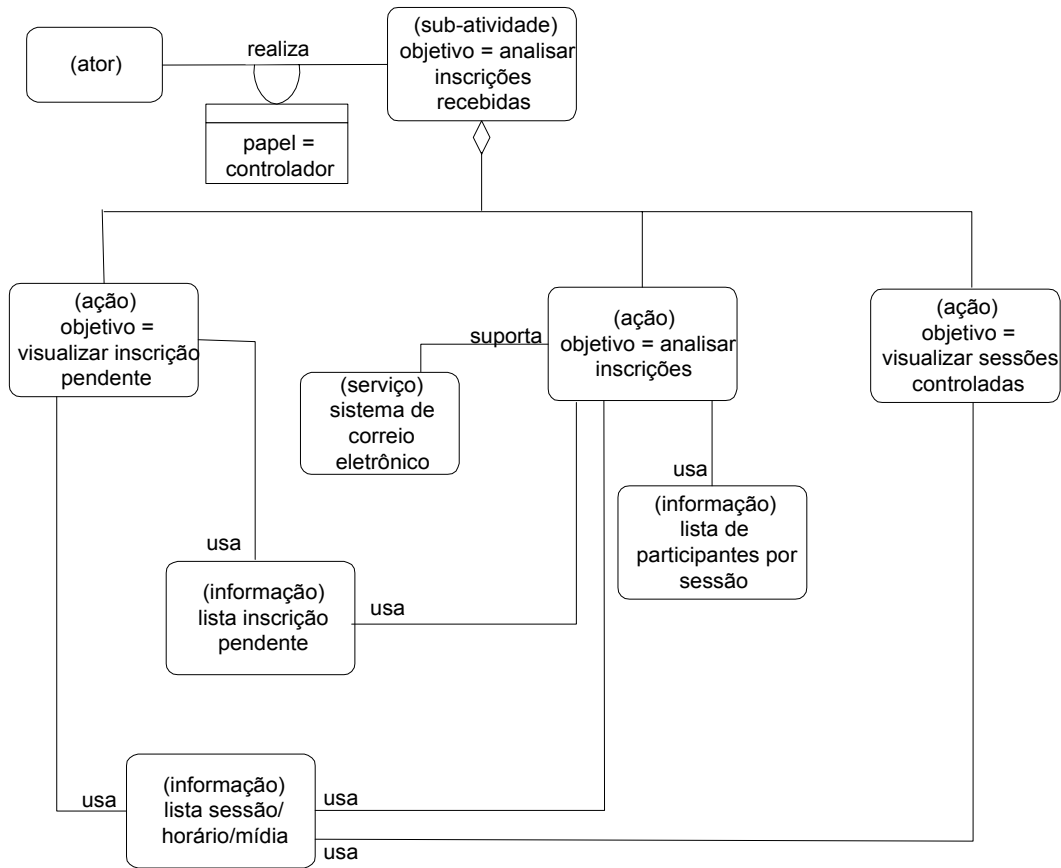


Figura 9. Diagrama de Objeto da Sub-Atividade Analisar Inscrições Recebidas

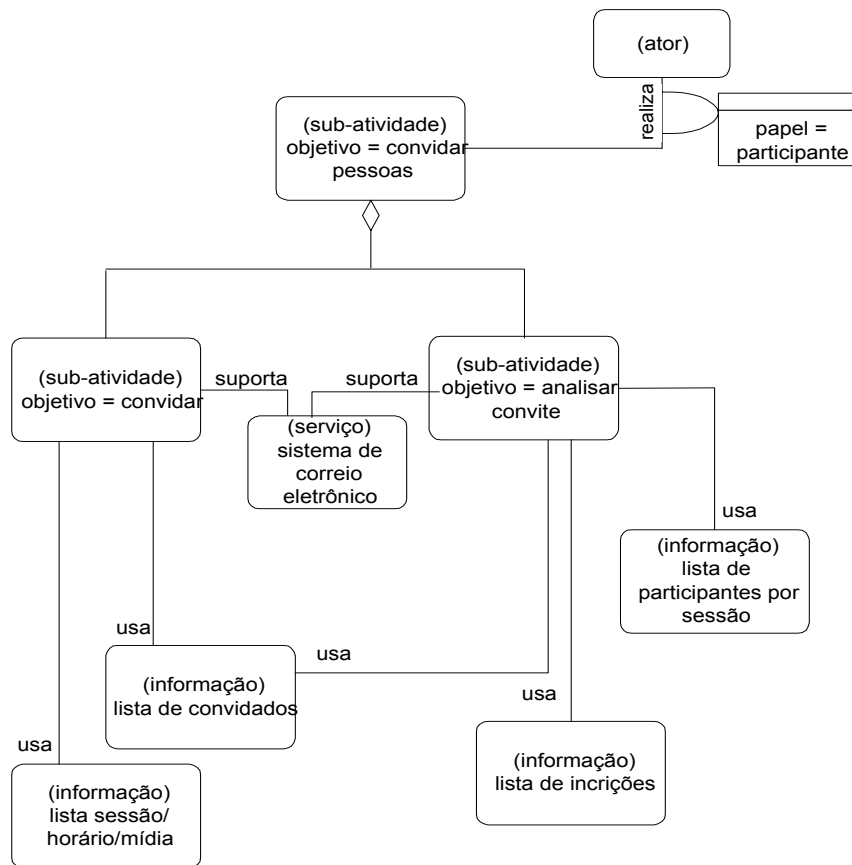


Figura 10. Diagrama de Objeto da Sub-Atividade Convidar Pessoas

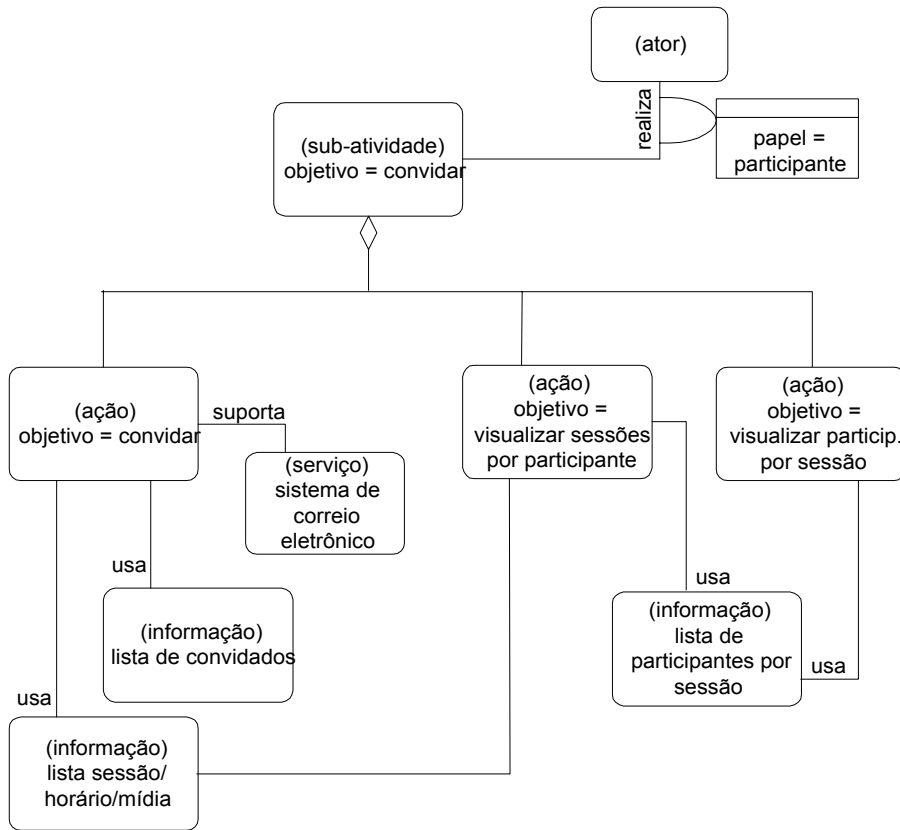


Figura 11. Diagrama de Objeto da Sub-Atividade Convidar

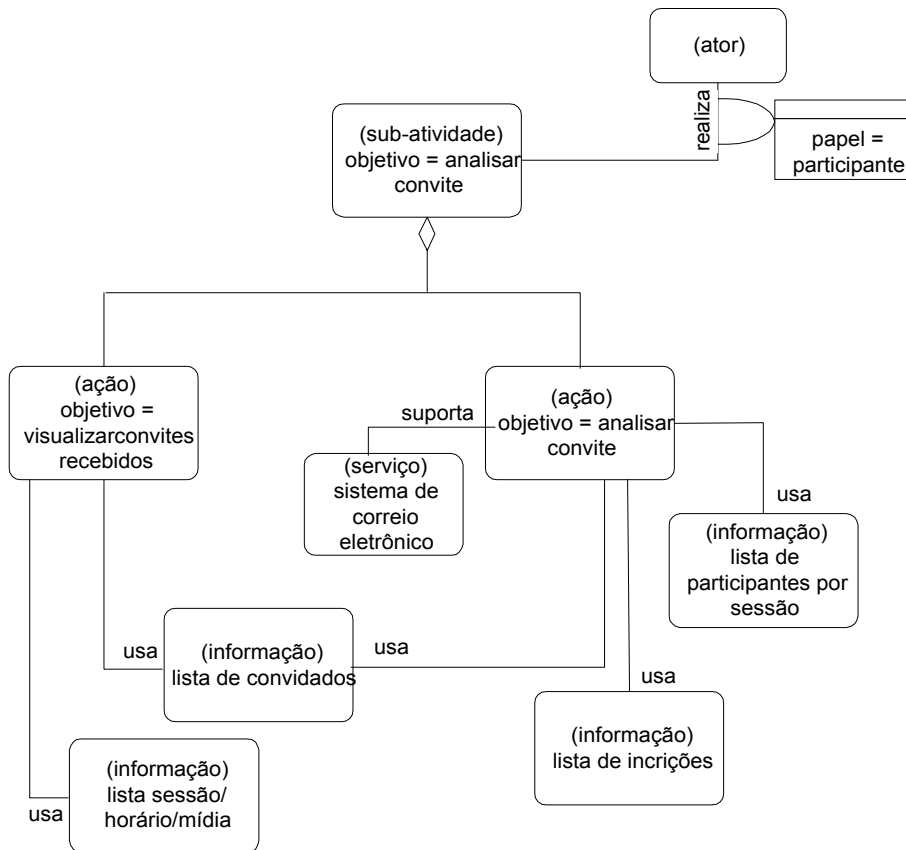


Figura 12. Diagrama de Objeto da Sub-Atividade Analisar Convite

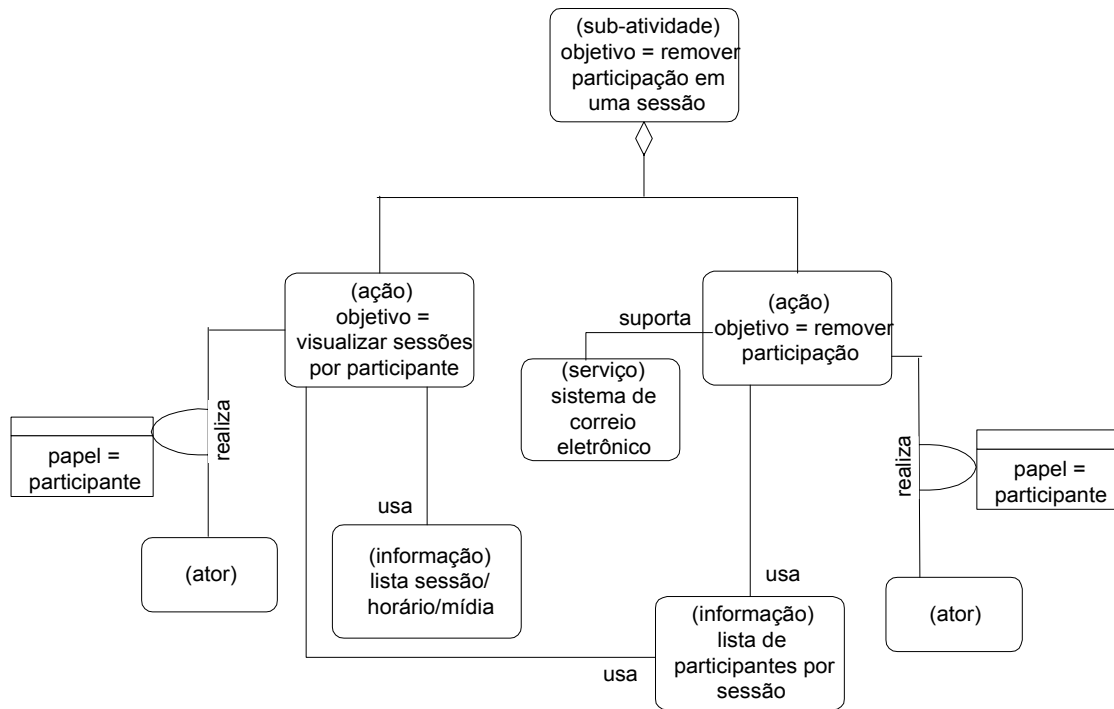


Figura 13. Diagrama de Objeto da Sub-Atividade Remover Participação em uma Sessão

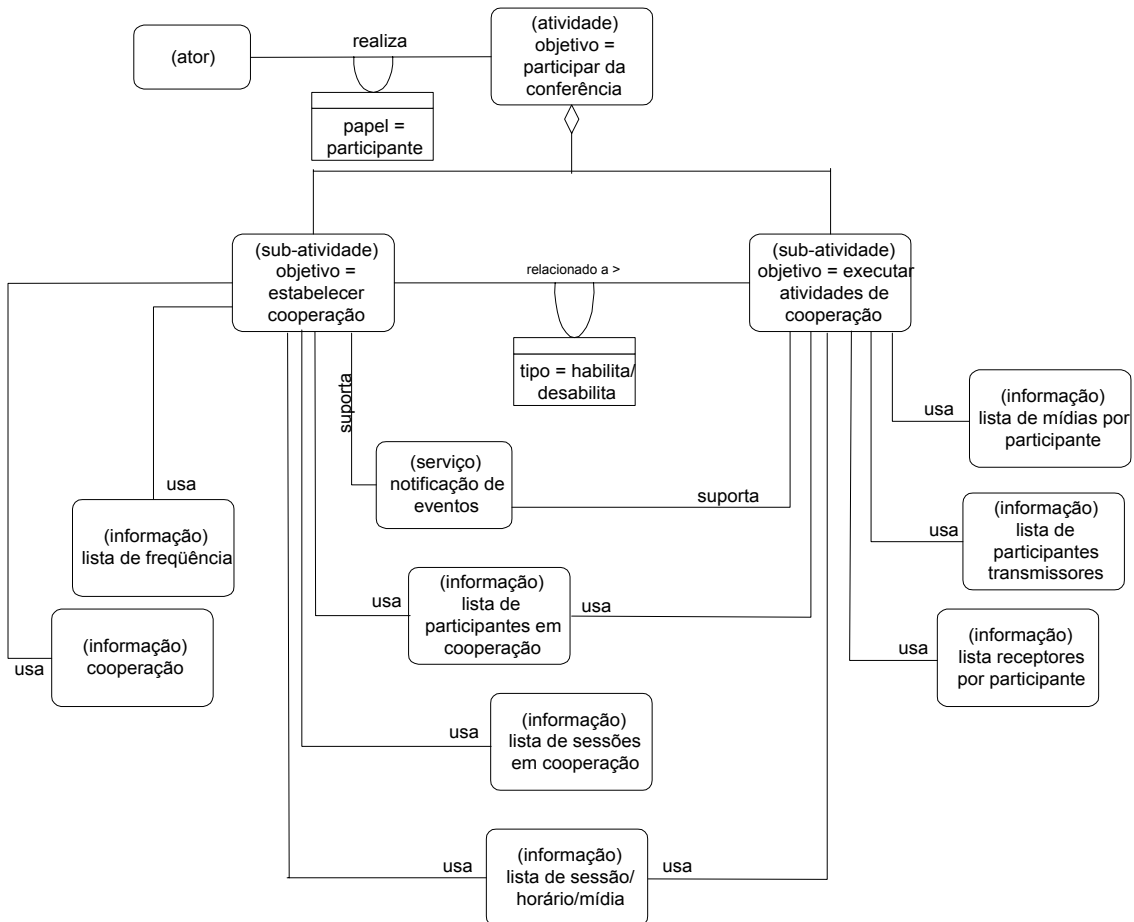


Figura 14. Diagrama de Objeto da Atividade Participar da Conferência

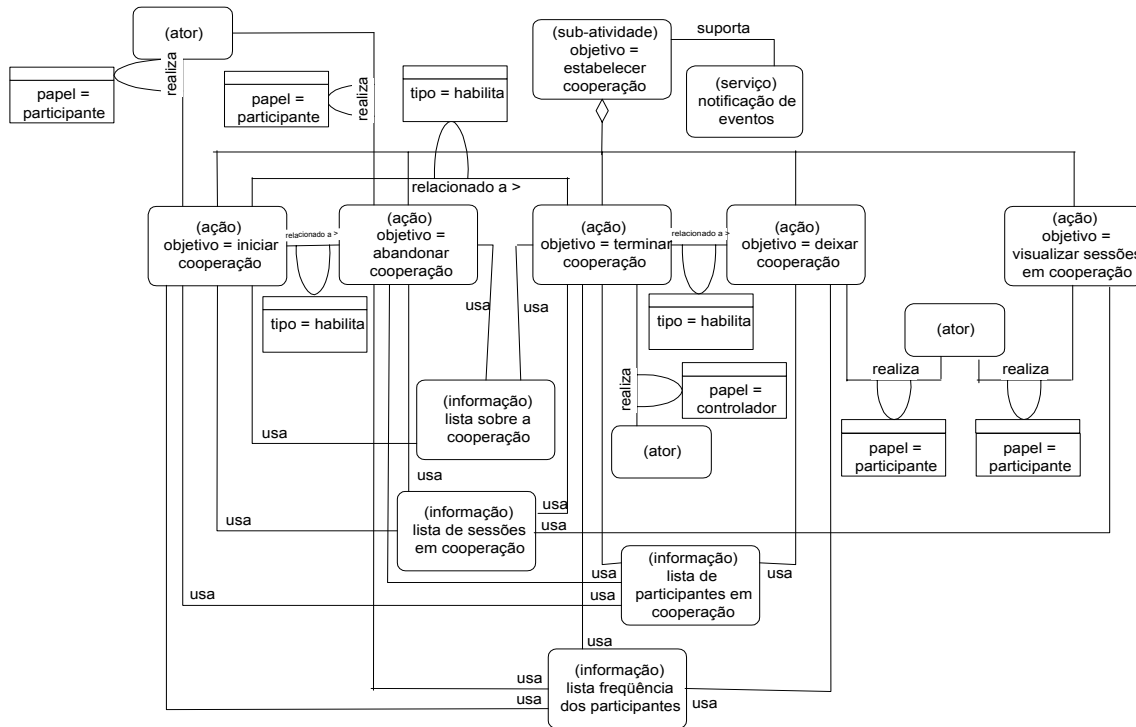


Figura 15. Diagrama de Objeto da Atividade Estabelecer Cooperação

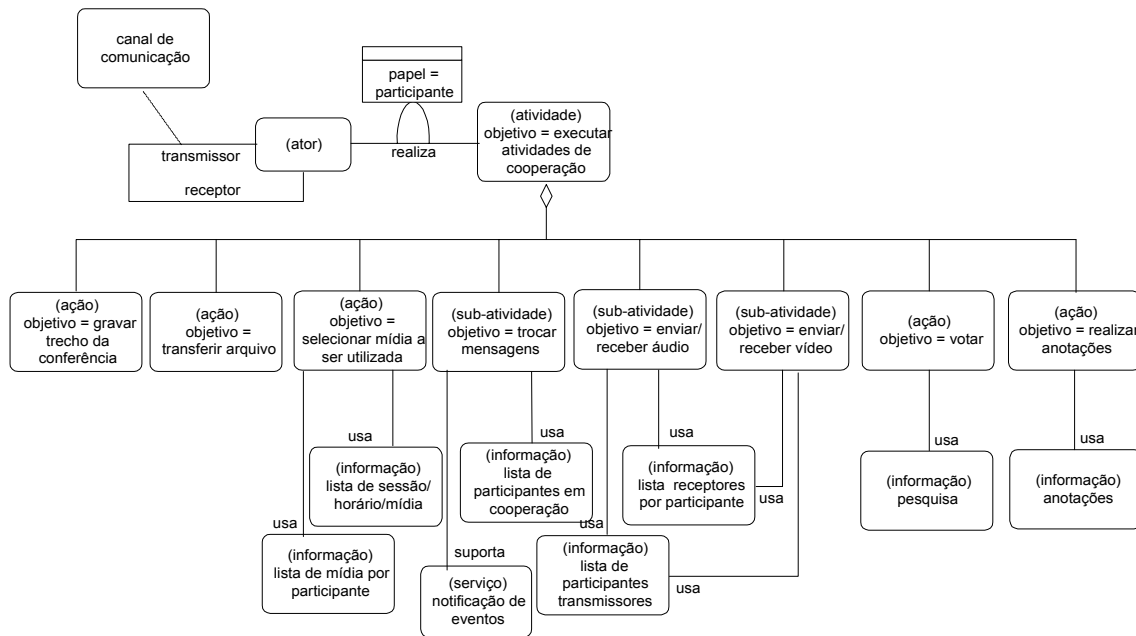


Figura 16. Diagrama de Objeto da Sub-Atividade Executar Atividades de Cooperação



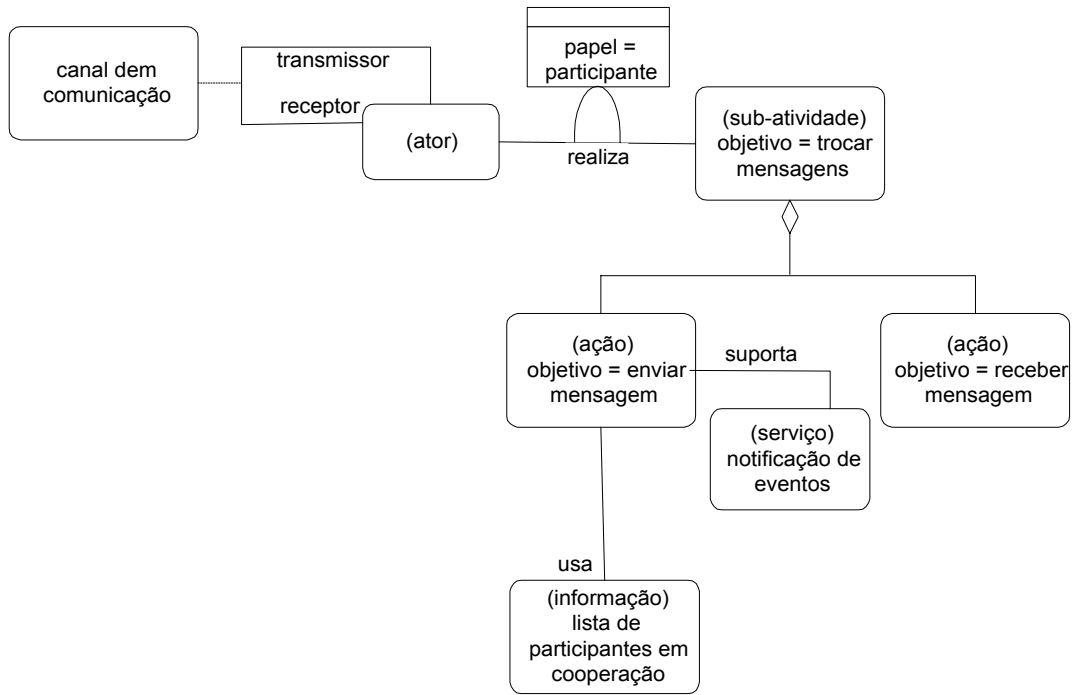


Figura 17. Diagrama de Objeto da Sub-Atividade Trocar Mensagens

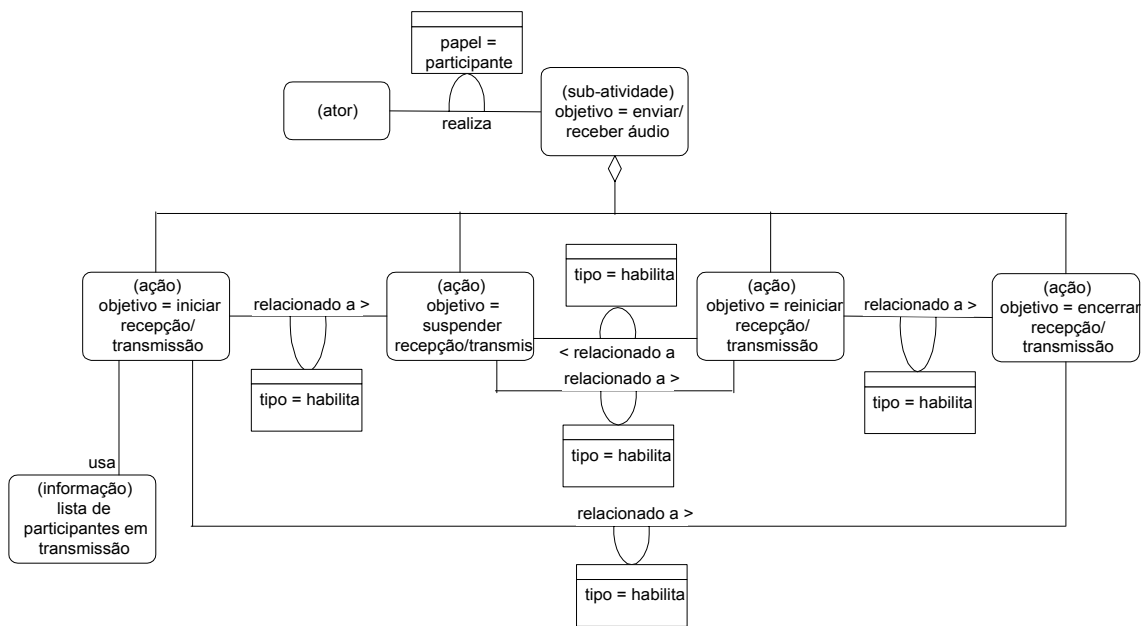


Figura 18. Diagrama de Objeto da Sub-Atividade Enviar/Receber Áudio

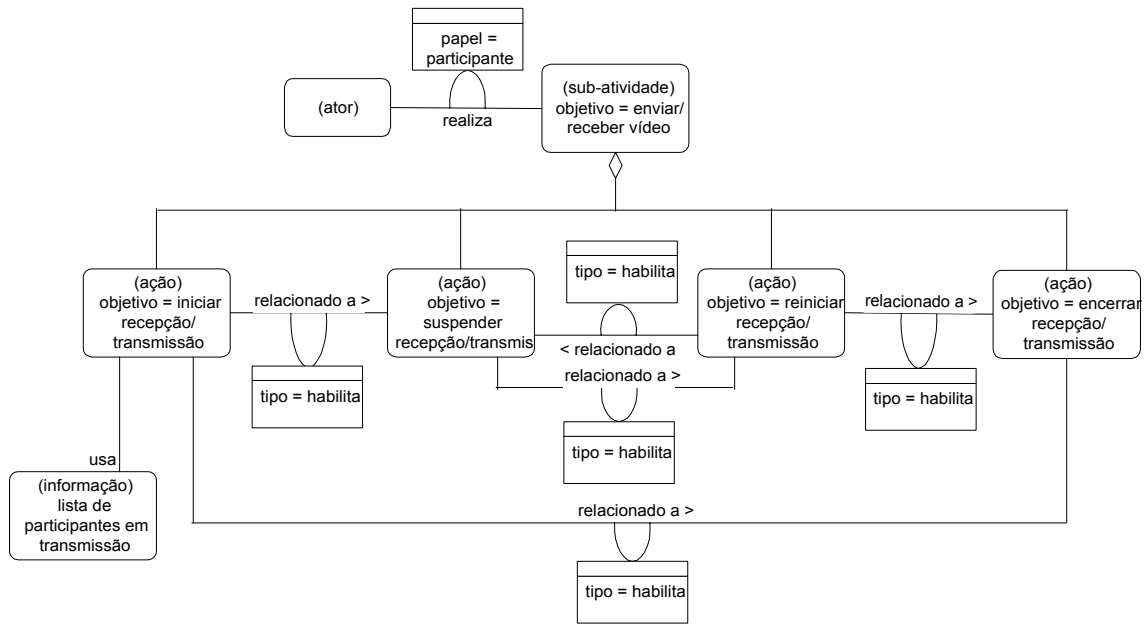


Figura 19. Diagrama de Objeto da Sub-Atividade Enviar/Receber Vídeo

### A2.2) Diagramas de Atividade

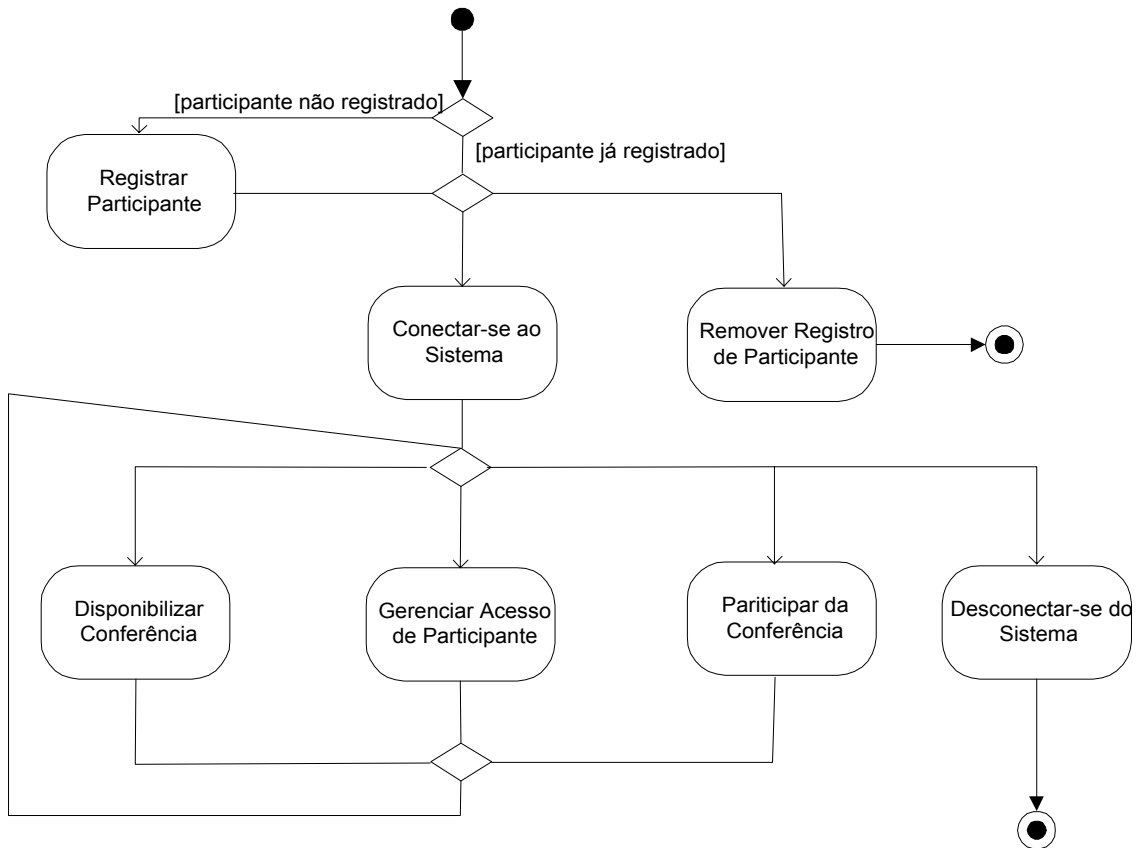


Figura 20. Diagrama de Atividade do Sistema de Conferência Multimídia

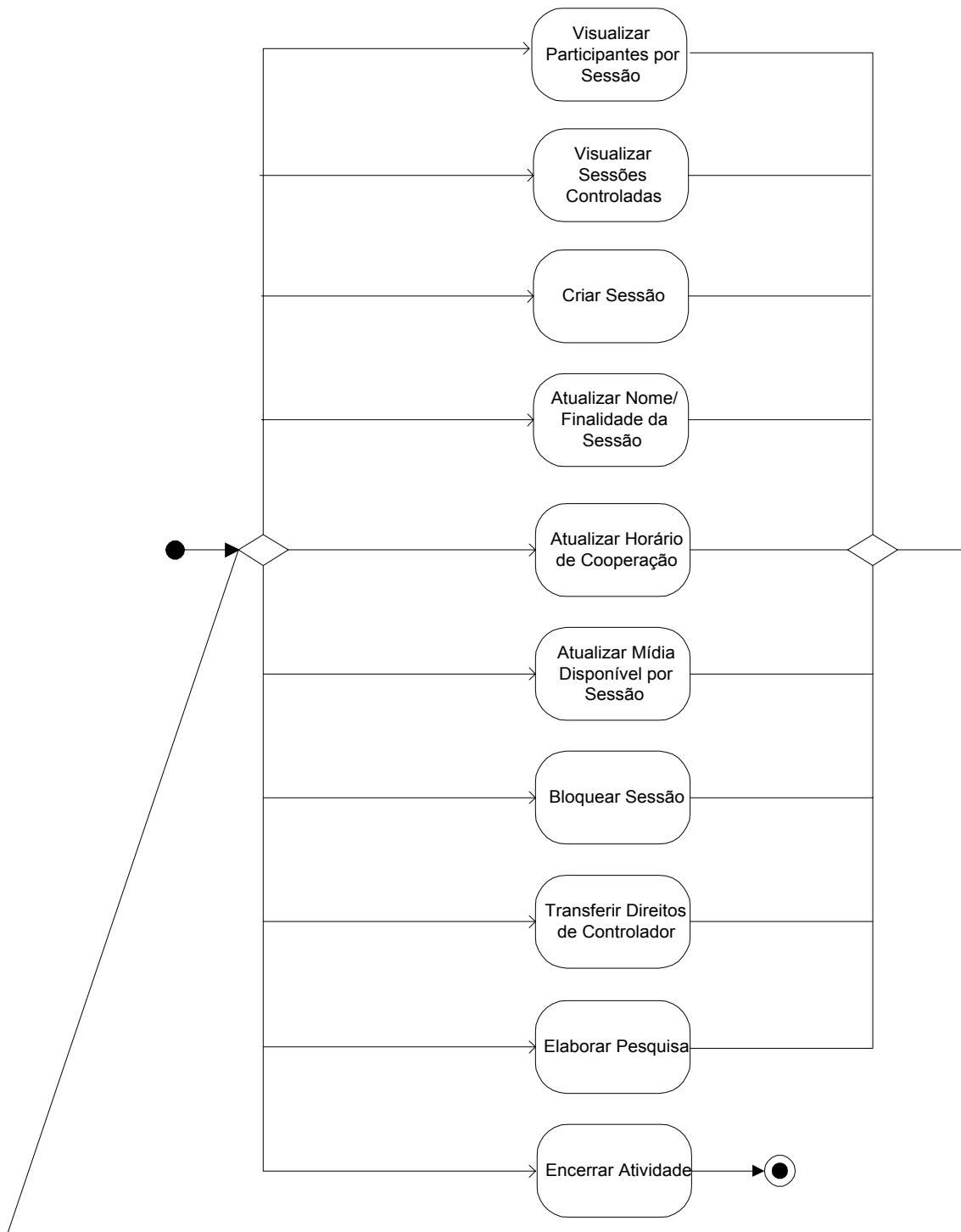


Figura 21. Diagrama de Atividade – Disponibilizar Conferência

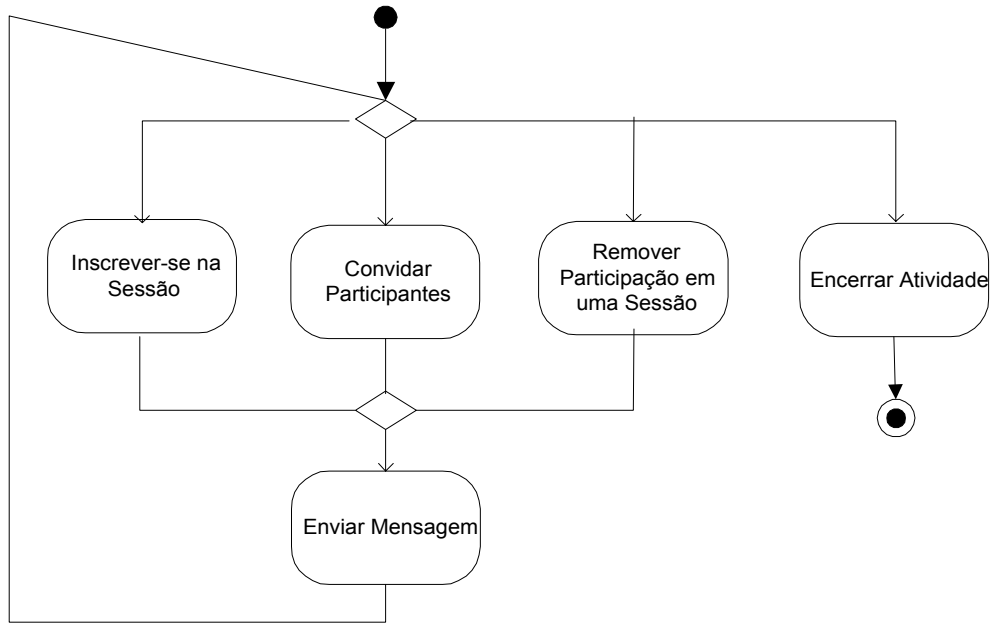


Figura 22. Diagrama de Atividade – Gerenciar Acesso de Participante

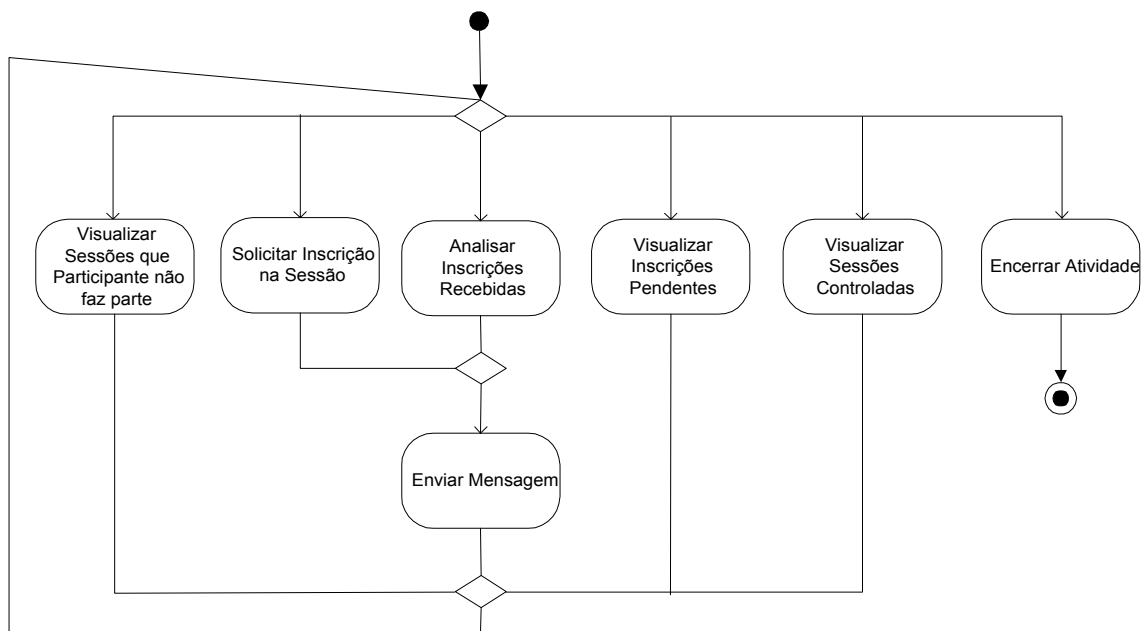


Figura 23. Diagrama de Atividade – Inscrever-se na Sessão

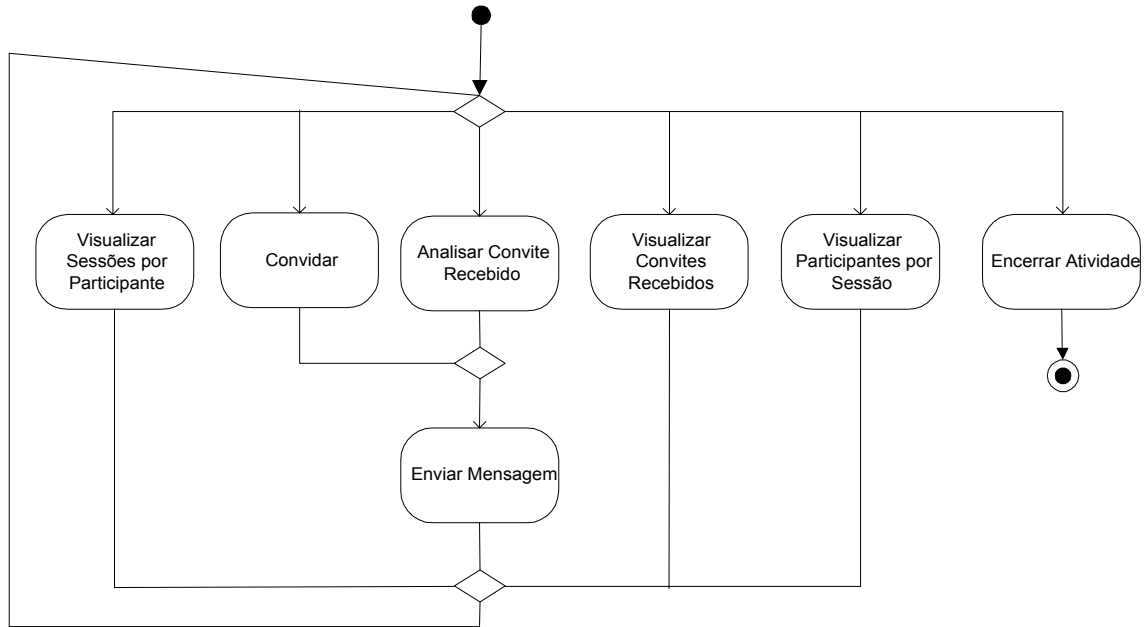


Figura 24. Diagrama de Atividade – Convidar Pessoas

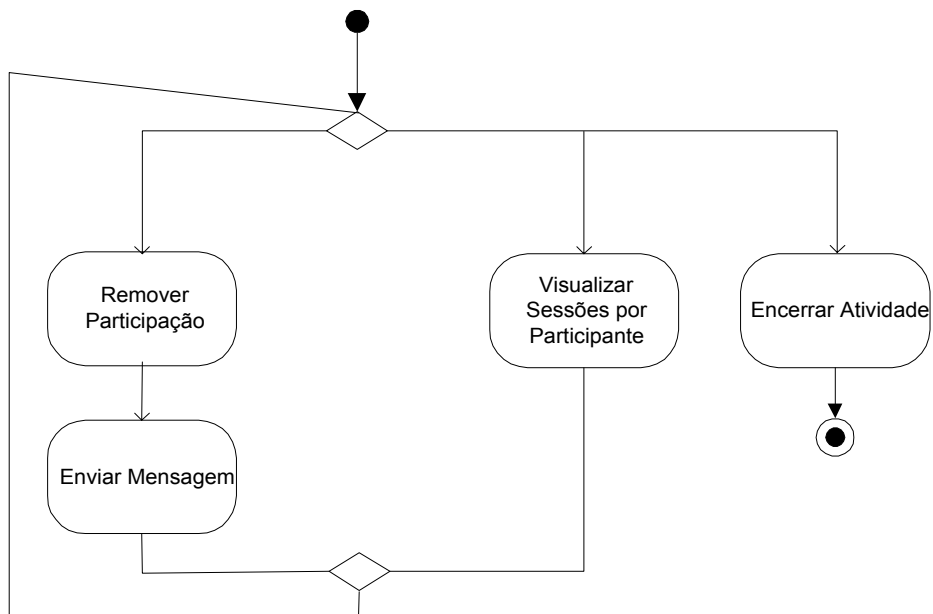


Figura 25. Diagrama de Atividade – Remove Participação na Sessão

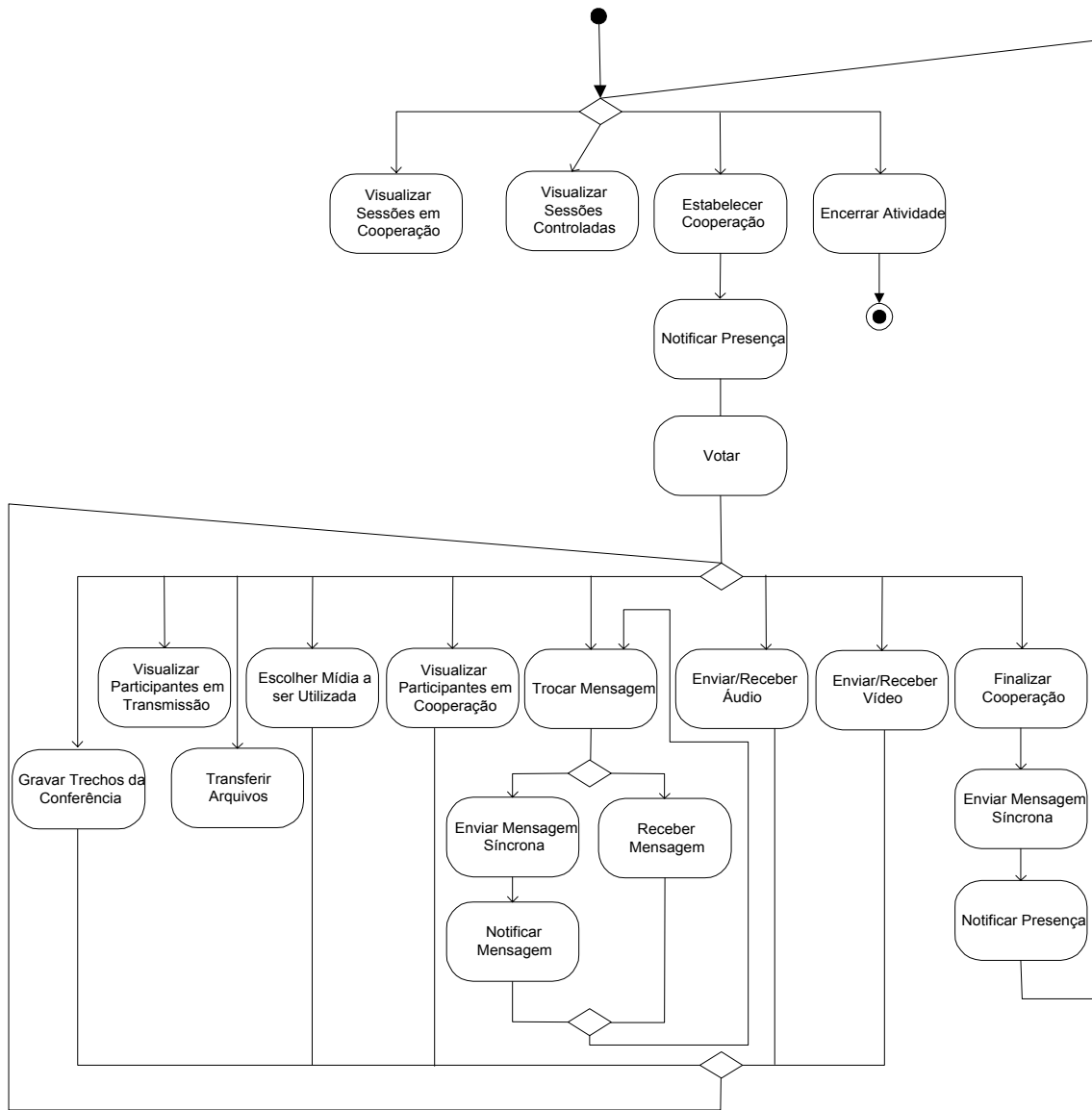


Figura 26. Diagrama de Atividade – Participar da Conferência

**A2.3) Glossário**

| <b>Nome</b>                      | <b>Nível</b> | <b>Tipo</b> | <b>Descrição</b>   |
|----------------------------------|--------------|-------------|--|
| Participante                     | Empresa      | Ator        | Pessoa que se realiza as atividades de acessar o sistema, disponibilizar conferências (criar), gerenciar acesso de participante (solicitar inscrição em uma sessão, remover participação em uma sessão, convidar pessoas para participarem de uma sessão e confirmar/recusar convite) e participar da conferência. |
| Controlador                      | Empresa      | Ator        | Pessoa que cria uma conferência, gerencia o acesso de participantes e dá início à cooperação.  |
| Acessar Sistema                  | Empresa      | Atividade   | Atividade realizada por qualquer participante para utilizar os serviços oferecidos pelo sistema.   |
| Disponibilizar Conferência       | Empresa      | Atividade   | Atividade realizada pelos participantes (criar conferência) ou pelos controladores das sessões.  |
| Gerenciar Acesso de Participante | Empresa      | Atividade   | Atividade realizada pelos participantes do sistema (solicitar inscrição, remover participação em uma sessão, convidar pessoas para participarem de uma sessão e confirmar/recusar convite) e pelos controladores das sessões (analisar inscrições).  |
| Participar da Conferência        | Empresa      | Atividade   | Atividade realizada pelos participantes de uma sessão com a finalidade de iniciar uma cooperação entre os membros.   |
| Registrar Participante           | Empresa      | Ação        | Atividade realizada por qualquer pessoa que tenha interesse em participar do sistema de conferência multimídia. Esta atividade registra um novo participante no sistema.   |
| Remover Registro de Participante | Empresa      | Ação        | Atividade realizada por qualquer participante que tenha o interesse em remover sua participação de uma sessão.   |
| Conectar-se ao Sistema           | Empresa      | Ação        | Atividade realizada por qualquer participante registrado no sistema com a finalidade de possibilitar que o participante utilize os serviços oferecidos pelo sistema.   |
| Desconectar-se do Sistema        | Empresa      | Ação        | Atividade realizada por qualquer participante, conectado ao sistema, quando este desejar deixar o sistema.   |

|                                    |         |               |   |
|------------------------------------|---------|---------------|---|
|                                    |         |               |   |
| Criar Sessão                       | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema permitindo a criação de uma sessão.   |
| Atualizar Sessão                   | Empresa | Sub-Atividade | Atividade realizada pelos controladores da sessão quando estes desejarem realizar qualquer atualização nas informações das sessões ou quando necessitar transferir direitos de controladores. |
| Remover Sessão                     | Empresa | Ação          | Atividade realizada pelos controladores da sessão para remoção de uma sessão controlada por estes.  |
| Transferir Direitos de Controlador | Empresa | Ação          | Atividade realizada pelos controladores da sessão quando estes desejarem transferir os direitos de controlador para um outro participante do sistema.   |
| Elaborar Pesquisa                  | Empresa | Ação          | Atividade realizada pelos controladores quando estes desejarem elaborar as perguntas para realização de pesquisa.   |
| Atualizar Nome da Sessão           | Empresa | Ação          | Atividade realizada pelos controladores da sessão para atualização do nome da sessão controlada por estes.  |
| Atualizar Finalidade da Sessão     | Empresa | Ação          | Atividade realizada pelos controladores da sessão para atualização da finalidade da sessão controlada por estes.  |
| Incluir Horário de Cooperação      | Empresa | Ação          | Atividade realizada pelos controladores da sessão para inclusão de horários de cooperação para a sessão controlada por estes.   |
| Alterar Horário de Cooperação      | Empresa | Ação          | Atividade realizada pelos controladores da sessão para alteração do horário de cooperação para a sessão controlada por estes.   |
| Remover Horário de Cooperação      | Empresa | Ação          | Atividade realizada pelos controladores da sessão para remoção do horário de cooperação para a sessão controlada por estes.   |
| Incluir Mídia Disponível           | Empresa | Ação          | Atividade realizada pelos controladores da sessão para inclusão de mídias disponíveis para a sessão controlada por estes.   |
| Remover Mídia Disponível           | Empresa | Ação          | Atividade realizada pelos controladores da sessão para remoção de mídias disponíveis para a sessão controlada por estes.  |
| Encerrar Atividade da Sessão       | Empresa | Ação          | Atividade realizada pelos controladores da sessão para encerrar as atividades da  |



|                                    |         |               |  |
|------------------------------------|---------|---------------|--|
|                                    |         |               | sessão controlada por estes, quando não houver mais a necessidade de cooperação entre os membros da sessão.  |
| Solicitar Inscrição na Sessão      | Empresa | Ação          | Atividade realizada por qualquer participante que tenha desejo fazer parte de uma sessão. A solicitação deve ser analisada pelo controlador da sessão.   |
| Remover Participação em uma Sessão | Empresa | Ação          | Atividade realizada por qualquer participante que não tenha mais interesse em participar de uma sessão.  |
| Analisar Inscrições Recebidas      | Empresa | Sub-Atividade | Atividade realizada pelo controlador da sessão para analisar as inscrições recebidas. O parecer do controlador deve ser aceitar ou recusar a inscrição do participante.  |
| Convidar Participantes             | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema. Sua função a de convidar qualquer pessoa, registrada ou não no sistema, para participar de uma sessão. Este convite deverá ser confirmado ou recusado por aquele que o recebeu.   |
| Confirmar Convite Recebido         | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema. Tem a finalidade de confirmar o convite recebido transformando o participante em membro da sessão.  |
| Recusar Convite Recebido           | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema cuja finalidade é recusar o convite recebido.  |
| Aceitar Inscrição                  | Empresa | Ação          | Atividade realizada pelo controlador da sessão cuja finalidade é aceitar a inscrição de um participante na sessão controlada por este.   |
| Recusar Inscrição                  | Empresa | Ação          | Atividade realizada pelo controlador da sessão cuja finalidade é recusar a inscrição de uma participante na sessão controlada por este.  |
| Iniciar Cooperação                 | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema e pertencente à sessão. Sua finalidade é iniciar a cooperação entre os membros da sessão no horário previsto para comunicação. Quando realizada pelo controlador, as informações sobre o início da cooperação devem ser armazenadas. |

|  |         |               |  |
|--|---------|---------------|--|
| Finalizar Cooperação                   | Empresa | Sub-Atividade | Atividade realizada pelos participantes (abandonar ou deixar cooperação) ou pelo controlador da sessão (abandonar ou terminar cooperação). Sua finalidade é encerrar a cooperação com os membros da sessão.  |
| Executar Atividades de Cooperação      | Empresa | Sub-Atividade | Atividade realizada pelos participantes e controlador da sessão em cooperação. Essas atividades consistem em finalizar a cooperação, selecionar mídia a ser utilizada durante a cooperação, trocar mensagens e transmitir / receber áudio/vídeo.                     |
| Abandonar Cooperação                   | Empresa | Ação          | Atividade realizada pelos participantes ou pelo controlador da sessão. Tem a finalidade de encerrar a cooperação. Quando executada pelo controlador da sessão, a cooperação deve ser encerrada para todos os demais participantes.                                   |
| Terminar Cooperação                    | Empresa | Ação          | Atividade realizada pelo controlador da sessão quando a cooperação chegar ao final. Uma mensagem de agradecimento deve ser enviada a todos os participantes, a frequência deve ser controlada e as informações sobre   |
| Deixar Cooperação                      | Empresa | Ação          | Atividade realizada por qualquer participante registrado no sistema, exceto pelo controlador da sessão. Sua finalidade é finalizar a cooperação dos participantes após esta ter sido terminada pelo controlador. A frequência dos participantes deve ser controlada. |
| Votar                                  | Empresa | Ação          | Atividade realizada por qualquer participante referente à votação (resposta à perguntas de uma pesquisa).  |
| Selecionar Mídia a ser Utilizada       | Empresa | Sub-Atividade | Atividade realizada por qualquer participante registrado no sistema durante a cooperação. Sua finalidade é selecionar qual(is) mídia(s) será(ão) utilizada(s) pelo participante, dentre as disponíveis para a sessão, durante a cooperação.                          |
| Trocar Mensagens                       | Empresa | Sub-Atividade | Atividade realizada entre dois participantes para trocar mensagens.  |
| Visualizar Participantes em Cooperação | Empresa | Ação          | Atividade realizada por qualquer participante em cooperação para informá-lo quem são os demais participantes que estão em cooperação.  |

|                               |         |            |  |
|-------------------------------|---------|------------|--|
| Transmitir/Receber Áudio      | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação desde que a mídia esteja disponível para a sessão e para o participante. Tem a finalidade de transmitir/receber áudio. No caso da recepção o participante deve selecionar de qual participante em cooperação e que esteja transmitindo deseja receber. |
| Transmitir/Receber Vídeo      | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação desde que a mídia esteja disponível para a sessão e para o participante. Tem a finalidade de transmitir/receber vídeo. No caso da recepção o participante deve selecionar de qual participante em cooperação e que esteja transmitindo deseja receber. |
| Gravar Trechos da Conferência | Empresa | Ação       | Atividade realizada por qualquer participante e tem a finalidade de permitir a gravação de trechos da conferência.   |
| Transferir Arquivos           | Empresa | Ação       | Atividade realizada por qualquer participante e tem a finalidade de permitir a transferência de arquivos entre os participantes em cooperação.   |
| Realizar Anotações            | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação que deseja realizar anotações durante a cooperação.  |
| Incluir Mídia a ser Utilizada | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação para selecionar qual mídia, dentre as disponíveis para a sessão, será utilizada por ele.   |
| Remover Mídia a ser Utilizada | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação para remover, dentre as mídias selecionadas, aquela que não mais utilizará.  |
| Enviar Mensagem               | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação para enviar mensagem para outro participante durante a cooperação.   |
| Receber Mensagem              | Empresa | Ação       | Atividade realizada por qualquer participante em cooperação para receber mensagem de outro participante durante a cooperação.  |
| Lista de Pessoas Registradas  | Empresa | Informação | Informação sobre quais pessoas estão registradas no sistema. Utilizada em registrar participante e remover registro de participante  |

|   |         |            |   |
|---|---------|------------|---|
| Lista de Participantes Ativos           | Empresa | Informação | Informação sobre quais participantes estão conectados ao sistema. Utilizada em conectar-se ao sistema e desconectar-se do sistema.  |
| Lista de Sessão/Horário/Mídia           | Empresa | Informação | Informação sobre quais são as sessões existentes no sistema, quais mídias estarão disponíveis para serem utilizadas e quais serão os horário para cooperação entre seus membros. Utilizada por criar sessão, atualizar sessão, remover sessão, solicitar inscrição na sessão, convidar participantes, remover participação em uma sessão, iniciar cooperação, executar atividades de cooperação, selecionar mídia a ser utilizada, incluir mídia a ser utilizada e remover mídia. |
| Pesquisa                                | Empresa | Informação | Informação sobre as perguntas a serem apresentadas aos participantes para estes realizarem a votação.   |
| Lista de Participantes por Sessão       | Empresa | Informação | Informação sobre quem são os membros de uma sessão. Utilizada por criar sessão, remover sessão, solicitar inscrição na sessão, analisar inscrições recebidas, convidar participantes, confirmar convite recebido, remover participação em uma sessão e aceitar inscrição.   |
| Lista de Controladores                  | Empresa | Informação | Informação sobre quem são os controladores de cada sessão. Utilizada por criar sessão, remover sessão e transferir direitos de controlador.   |
| Lista de Informações sobre a Cooperação | Empresa | Informação | Informação sobre a cooperação como data e hora de início e término da cooperação. Utilizada por iniciar cooperação, finalizar cooperação, abandonar cooperação e terminar cooperação.   |
| Lista de Inscrição Pendentes            | Empresa | Informação | Informação sobre as inscrições que ainda não foram analisadas. Utilizada em solicitar inscrição, analisar inscrições recebidas, aceitar inscrição e recusar inscrição.  |
| Lista de Convidados                     | Empresa | Informação | Informação sobre quais pessoas foram convidadas a participar da sessão e quais ainda não analisaram o convite, isto é, ainda não aceitaram nem recusaram o mesmo. Utilizada por convidar participantes, recusar convite recebido e confirmar convite recebido.  |

|                                       |         |            |   |
|---------------------------------------|---------|------------|---|
| Lista de Frequência                   | Empresa | Informação | Informação sobre a frequência dos participantes quando estes estabelecerem uma sessão. Utilizado por iniciar cooperação, finalizar cooperação, abandonar cooperação, terminar cooperação e deixar cooperação.   |
| Lista de Participantes em Cooperação  | Empresa | Informação | Informação sobre quem são os participantes em cooperação num dado instante. Utilizada por iniciar cooperação, finalizar cooperação, executar atividades de cooperação, trocar mensagens, visualizar participantes em cooperação, abandonar cooperação, terminar cooperação e deixar cooperação. |
| Anotações                             | Empresa | Informação | Informação sobre as anotações realizadas pelos participantes durante uma cooperação. Utilizada por executar atividades de cooperação e realizar anotações.  |
| Lista de Mídias por Participante      | Empresa | Informação | Informação sobre as mídias selecionadas pelo participante para serem utilizadas durante uma cooperação. Utilizada por executar atividades de cooperação, selecionar mídia a ser utilizada, incluir mídia a ser utilizada e remover mídia.   |
| Lista de Participantes Transmissores  | Empresa | Informação | Informação sobre quem são os participantes em cooperação que estão transmitindo áudio/vídeo. Utilizada por executar atividades de cooperação e transmitir áudio/vídeo.  |
| Lista de Receptores por Transmissores | Empresa | Informação | Informação sobre quem são os participantes em cooperação que estão recebendo áudio/vídeo e de quem estão recebendo. Utilizada por executar atividades de cooperação e receber áudio/vídeo.  |
| Sistema de Correio Eletrônico         | Empresa | Serviço    | Serviço de correio eletrônico utilizado para enviar mensagem para aqueles que estão ou não registrados no sistema de conferência multimídia.  |
| Notificação                           | Empresa | Serviço    | Serviço de notificação de eventos utilizado para notificar aos participantes sobre a chegada de novas mensagens ou entrada/saída dos participantes durante a cooperação.  |

## **Apêndice 3**

### ***Sistema de Conferência Multimídia – Nível de Sistema***

Este apêndice apresenta a modelagem do Sistema de Conferência Multimídia (SCM) para o nível de sistema. Este nível deve definir o limite entre o sistema e o seu ambiente através da identificação e descrição dos requisitos do sistema e dos serviços externos suportados pelo sistema. Neste nível as diferenças entre as três visões tornam-se aparentes e são discutidas separadamente.

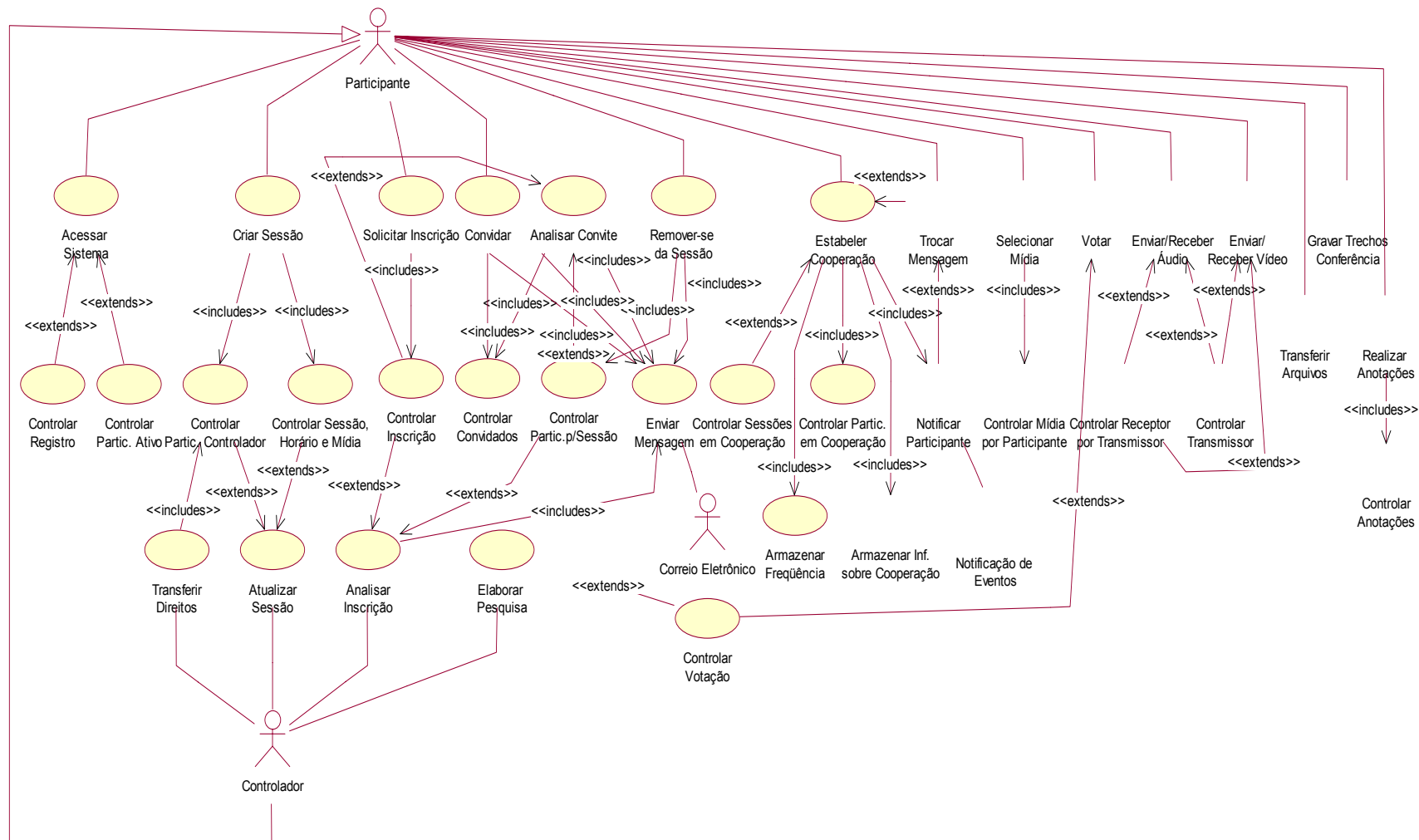
A modelagem deste nível consiste, basicamente, na execução das seguintes etapas:

- 1) Visão Estrutural
  - a) elaboração do diagrama de casos de uso
  - b) descrição textual dos casos de uso
  - c) elaboração do diagrama de pacotes (componentes)
- 2) Visão Comportamental
  - a) elaboração do diagrama de colaboração ou seqüência
- 3) Visão Interacional
  - a) elaboração do diagrama de colaboração ou seqüência
  - b) descrição das mensagens
  - c) relação das interfaces
- 4) Atualização do Glossário

A seguir serão apresentadas cada uma das etapas para modelagem do Sistema de Conferência Multimídia no nível de sistema.

#### **A3.1) Visão Estrutural**

- a) Diagrama de Casos de Uso



## b) Descrição Textual dos Casos de Uso

**Caso de Uso:** Acessar Sistema.

**Finalidade:** Este caso de uso tem a finalidade de demonstrar como se dá o acesso ao sistema.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o participante tenta acessar o sistema através de um pedido de <u>registro</u> . Deve ser informado um username (nome para o participante se identificar), nome do participante, uma senha para permitir seu acesso e seu email. | 2. O sistema deve verificar se o username fornecido está sendo usado.<br>2.1. Se já estiver, deve retornar uma mensagem de erro e rejeitar o pedido de registro.<br>2.2. Se não estiver:<br>a) ponto de extensão: utilizar caso de uso controlar registro;<br>b) ponto de extensão: utilizar caso de uso controlar participante ativo;<br>c) o acesso ao sistema será concedido;   |
| 1. Este caso de uso tem início quando o participante tenta acessar o sistema através de um pedido de <u>conexão</u> ( <u>login</u> ). Deve ser informado o username e a senha.   | 1. O sistema deve verificar algumas condições e permitir o acesso se:<br>1.1. o participante (username) está registrado no sistema;<br>1.2. a senha para o username é válida;<br>1.3. o participante ainda não está conectado (se não faz parte da lista de participantes ativos).<br>2. Se o acesso não for liberado devido a algumas das condições anteriores não ter sido satisfeita, uma mensagem de erro deve ser retornada e o pedido de conexão deve ser negado.<br>3. Se o acesso for liberado:<br>3.1. ponto de extensão: utilizar caso de uso controlar participante ativo;<br>3.2. a entrada ao sistema será permitida; |
| 1. Este caso de uso tem início quando o participante solicita a <u>remoção de seu registro</u> no sistema. Deve ser informado o username e a senha do participante.  | 2. O sistema deve verificar algumas condições e permitir a remoção se:<br>2.1. o participante está registrado no sistema (se faz parte da lista de pessoas registradas);<br>2.2. a senha para o username é válida;<br>2.3. se o participante não está conectado (se não faz parte da lista de participantes ativos).   |



|   |  |
|---|--|
|   | <ol style="list-style-type: none"> <li>3. Se pelo menos uma das condições anteriores não for satisfeita, retornar uma mensagem de erro e recusar o pedido de remoção do registro.</li> <li>4. Se a remoção for permitida, ponto de extensão: utilizar o caso de uso controlar registro.</li> </ol> |
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o participante deseja <i>deixar o sistema (logoff)</i>. Este pedido deve conter o username do participante.</li> </ol> | <ol style="list-style-type: none"> <li>2. Ponto de extensão: utilizar o caso de uso controlar participante ativo.</li> </ol>   |

**Caso de Uso:** Controlar Registro.

**Finalidade:** Este caso de uso tem a finalidade de adicionar ou remover as pessoas que se registram (participantes) à uma lista de registros.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o sistema recebe um pedido de <i>registro</i>. Este pedido deve conter o username, nome, senha e email.</li> </ol>          | <ol style="list-style-type: none"> <li>2. O sistema deve adicionar o participante (username e seus dados) a uma lista de pessoas registradas.</li> </ol> |
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o sistema recebe um pedido de <i>remoção do registro</i> de um participante. Este pedido deve conter o username.</li> </ol> | <ol style="list-style-type: none"> <li>2. O sistema deve marcar o participante (username) como removido na lista de pessoas registradas.</li> </ol>      |

**Caso de Uso:** Controlar Particip. Ativo.

**Finalidade:** Este caso de uso tem a finalidade de adicionar ou remover os participantes conectados ao sistema à uma lista de participantes ativos.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o sistema recebe um pedido de <i>registro</i>. Este pedido deve conter o username.</li> </ol>            | <ol style="list-style-type: none"> <li>2. O sistema deve adicionar o participante (username) a uma lista de participantes ativos.</li> </ol> |
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o sistema recebe um pedido de <i>conexão (login)</i>. Este pedido deve conter o username.</li> </ol>     | <ol style="list-style-type: none"> <li>2. O sistema deve adicionar o participante (username) a uma lista de participantes ativos.</li> </ol> |
| <ol style="list-style-type: none"> <li>1. Este caso de uso tem início quando o sistema recebe um pedido de <i>desconexão (logoff)</i>. Este pedido deve conter o username.</li> </ol> | <ol style="list-style-type: none"> <li>2. O sistema deve remover o participante (username) da lista de participantes ativos.</li> </ol>      |

**Caso de Uso:** Criar Sessão.

**Finalidade:** Este caso de uso tem a finalidade de criar uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o participante solicita a <u>criação de uma sessão</u> . Este pedido deve conter o nome da sessão, a finalidade da existência da sessão, quais mídias poderão ser utilizadas, data e horário de início da cooperação, ou seja, do estabelecimento da sessão. | 2. O sistema deve verificar se já foi criada uma sessão com este nome e por este participante (username):<br>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;<br>2.2. em caso negativo,<br>2.2.1. incluir caso de uso controlar sessão, horários e mídias.<br>2.2.2. incluir caso de uso controlar particip. controlador. |

**Caso de Uso:** Atualizar Sessão.

**Finalidade:** Este caso de uso tem a finalidade de alterar informações sobre a sessão ou remover uma sessão.

**Ator:** Controlador.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>  |
|---|---|
| 1. Este caso de uso tem início quando o controlador solicita uma <u>inclusão de horário</u> no qual os participantes da sessão devem iniciar uma cooperação, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controladas por este solicitante.<br>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja incluir horário. Este pedido deve conter o nome da sessão e a data e o horário do início da cooperação. | 2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante da inclusão.<br><br>2. O sistema deve verificar se já existe um horário para a nova data e hora solicitados.<br>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;<br>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias. |
| 1. Este caso de uso tem início quando o controlador solicita uma <u>alteração de horário</u> no qual os participantes da sessão devem iniciar uma cooperação, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controladas por este solicitante.  | 2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante da alteração.  |

|  |   |
|--|---|
| <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja alterar o horário. Este pedido deve conter o nome da sessão, a antiga data e horário e a nova data e horário.</p>  | <p>2. O sistema deve verificar se já existe um horário para a nova data e hora solicitados ou se já está havendo uma comunicação na antiga data e hora.</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p>   |
| <p>1. Este caso de uso tem início quando o controlador solicita a <u>remoção de horário</u> no qual os participantes da sessão devem iniciar uma cooperação, fazendo sua escolha a partir de uma lista de sessões controladas por este solicitante.</p> <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja remover o horário. Este pedido deve conter o nome da sessão e o horário que se deseja remover.</p> | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante.</p> <p>2. O sistema deve verificar se está havendo cooperação entre os membros da sessão.</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p>  |
| <p>1. Este caso de uso tem início quando o controlador solicita uma <u>alteração do nome da sessão</u>, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controladas por este solicitante.</p> <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja alterar o nome. Este pedido deve conter o antigo nome da sessão e o novo nome da sessão.</p>  | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante da alteração.</p> <p>2. O sistema deve verificar se já existe uma sessão criada por este solicitante com o novo nome ou se existe alguma comunicação para a sessão antiga.</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias e utilizar caso de uso controlar particip. Controlador.</p> |
| <p>1. Este caso de uso tem início quando o</p>   | <p>2. O sistema deve mostrar uma lista das</p>  |

|   |  |
|---|--|
| <p>controlador solicita uma <u>alteração da finalidade da sessão</u>, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controladas por este solicitante.</p> <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja alterar a finalidade. Este pedido deve conter o nome da sessão e a nova finalidade desta.</p>  | <p>sessões controladas pelo solicitante da alteração.</p> <p>4. Ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p>   |
| <p>1. Este caso de uso tem início quando o controlador solicita a <u>inclusão de mídia</u> disponível para cada sessão, fazendo a escolha da sessão para a qual deseja incluir a mídia, a partir de uma lista de sessões controladas por este solicitante.</p> <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja escolher a(s) mídia(s) que estará(o) disponível(is). Este pedido deve conter o nome da sessão e a mídia a ser disponibilizada.</p>           | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante deste serviço.</p> <p>2. O sistema deve verificar se a mídia desejada já está disponibilizada para a sessão e se não está havendo cooperação para esta sessão.</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p>                   |
| <p>1. Este caso de uso tem início quando o controlador solicita a <u>remoção de mídia</u> disponível para cada sessão, fazendo a escolha da sessão para a qual deseja excluir a mídia, a partir de uma lista de sessões controladas por este solicitante.</p> <p>3. O controlador deve selecionar a sessão, dentre as apresentadas na lista, para a qual deseja escolher a(s) mídia(s) que não mais estará(o) disponível(is). Este pedido deve conter o nome da sessão e a mídia a ser indisponibilizada.</p> | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante deste serviço.</p> <p>2. O sistema deve verificar se a mídia desejada está disponibilizada ou se não está havendo cooperação para esta sessão.</p> <p>2.1. caso uma das alterantivas anteriores seja verdadeira, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p> |

|  |  |
|--|--|
| <p>1. Este caso de uso tem início quando o controlador solicita a <u>remoção de uma sessão</u>, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controlados por este solicitante.</p> <p>3. O controlador deve selecionar uma ou mais sessões para remoção. Este pedido deve conter o(s) nome(s) da(s) sessão(ões) a ser(em) excluída(s).</p>   | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante da remoção.</p> <p>2. O sistema deve verificar se existe algum participante cooperando na(s) sessão(ões) a ser(em) removida(s) ou se existe alguma informação armazenada sobre a(s) mesma(s).</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo:</p> <p>2.2.1. ponto de extensão: utilizar caso de uso controlar particip. controlador;</p> <p>2.2.2. ponto de extensão: utilizar caso de uso controlar sessão, horários e mídias.</p> |
| <p>1. Este caso de uso tem início quando o controlador solicita o <u>fechamento de uma sessão</u>, a qual não mais terá comunicação entre seus membros, fazendo sua escolha a partir de uma lista de sessões ainda ativas, controladas por este solicitante e que não possuam qualquer comunicação futura existente na lista de horários.</p> <p>3. O controlador deve selecionar uma ou mais sessões para fechamento. Este pedido deve conter o(s) nome(s) da(s) sessão(ões) a ser(em) fechada(s) e a data de fechamento.</p> | <p>2. O sistema deve mostrar uma lista das sessões controladas pelo solicitante do fechamento.</p> <p>2. O sistema deve verificar se existe alguma cooperação para a(s) sessão(ões) ou se existe algum horário futuro previsto para início da cooperação para cada sessão.</p> <p>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;</p> <p>2.2. em caso negativo, ponto de extensão: utilizar caso de uso controlar sessões, horários e mídias.</p>  |

**Caso de Uso:** Transferir Direitos Controlador.

**Finalidade:** Este caso de uso tem a finalidade de trocar o controlador de uma sessão.

**Ator:** Controlador.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o controlador deseja <i>transferir direitos de controlador</i> a um outro participante da sessão.                             | 2. O sistema deverá apresentar uma lista de quais sessões são controladas pelo solicitante.  |
| 3. O controlador deve selecionar a sessão que deseja transferir direitos de controlador. Este pedido deve conter o nome da sessão e o username do novo controlador. | 4. O sistema deverá verificar se os usernames são diferentes, se o username do novo controlador é válido, isto é, está registrado no sistema e se o novo controlador pertence à sessão a qual deverá controlar.<br>4.1. em caso afirmativo, incluir caso de uso controlar particip. controlador.<br>4.2. em caso negativo, enviar uma mensagem de erro e rejeitar o pedido de transferência de direitos. |

**Caso de Uso:** Elaborar Pesquisa.

**Finalidade:** Este caso de uso tem a finalidade de permitir ao controlador elaborar perguntas a serem respondidas pelos participantes quando estes iniciarem uma cooperação.

**Ator:** Controlador.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>  |
|---|---|
| 1. Este caso de uso tem início quando o controlador deseja <i>elaborar pesquisa</i> para uma sessão.  | 2. O sistema deverá apresentar uma lista de quais sessões são controladas pelo solicitante. |
| 3. O controlador deve selecionar a sessão e elaborar as perguntas. Este pedido deve conter o username do controlador e o nome da sessão com suas respectivas perguntas (questões e alternativas). | 4. Ponto de extensão: controlar votação.  |

**Caso de Uso:** Controlar Particip. Controlador.

**Finalidade:** Este caso de uso tem a finalidade de controlar quem são os controladores das sessões.

**Ator:** Participante/Controlador (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido de um participante para <u>criar uma sessão</u> . Este pedido deve conter o username do participante e o nome da sessão.  | 2. O sistema deverá atualizar uma lista de controladores contendo o username do participante, que passa a ser controlador, e o nome da sessão controlada por ele. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido do controlador para a <u>remoção da sessão</u> . Este pedido deve conter o username do solicitante (controlador) e o(s) nome(s) da(s) sessão(ões).  | 2. O sistema deverá remover o username do controlador da lista de controladores da sessão.  |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>transferir os direitos de controlador</u> para um outro participante. Este pedido deve conter o username do controlador atual, o nome da sessão e o username do participante que será o novo controlador da sessão. | 2. O sistema deverá remover o controlador atual da lista de controladores da sessão e adicionar o username do novo controlador à esta lista.                      |

**Caso de Uso:** Controlar Sessão, Horários e Mídias.

**Finalidade:** Este caso de uso tem a finalidade de controlar quais são as sessões existentes e quais as suas características.

**Ator:** Participante/Controlador (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido de um participante para <u>criar uma sessão</u> . Este pedido deve conter o username do participante, o nome da sessão, a finalidade da existência da sessão, as mídias a serem utilizadas, data e horário de início da cooperação. | 2. O sistema deverá atualizar uma lista contendo o username do participante, o nome da sessão, a finalidade e o(s) horário(s) de início da cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de um participante para <u>inclusão no horário</u> no qual os participantes da sessão deverão iniciar a cooperação. Este pedido deve conter o username do controlador, o nome da sessão e a nova data e horário.                    | 2. O sistema deverá atualizar uma lista contendo o username do participante, o nome da sessão, a finalidade e o(s) horário(s) de início da cooperação. |

|   |  |
|---|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>alteração no horário</u> no qual os participantes da sessão deverão iniciar a cooperação. Este pedido deve conter o username do controlador, o nome da sessão, a antiga data e horário e a nova data e horário início da cooperação. | 2. O sistema deverá atualizar uma lista contendo o username do participante, o nome da sessão, a finalidade e o(s) horário(s) de início da cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>alteração no nome da sessão</u> . Este pedido deve conter o username do controlador, o nome antigo da sessão e o novo nome da sessão.  | 2. O sistema deverá atualizar uma lista contendo o username do participante, o nome da sessão, a finalidade e o(s) horário(s) de início da cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>alteração na finalidade da sessão</u> . Este pedido deve conter o username do controlador, o nome da sessão e a nova finalidade da sessão.   | 2. O sistema deverá atualizar uma lista contendo o username do participante, o nome da sessão, a finalidade e o(s) horário(s) de início da cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido do controlador para a <u>remoção da sessão</u> . Este pedido deve conter o username do solicitante (controlador) e o(s) nome(s) da(s) sessão(ões).   | 2. O sistema deverá marcar a(s) sessão(ões) para o usuário solicitante como removida(s).   |

**Caso de Uso:** Solicitar Inscrição na Sessão.

**Finalidade:** Este caso de uso tem a finalidade de permitir que as pessoas registradas no sistema, os participantes, solicitem sua inscrição em alguma(s) sessão(ões), da(s) qual(is) ainda não façam parte.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante solicita sua <u>inscrição em uma sessão</u> . Este pedido deve conter o o(s) username(s) do(s) controlador(es), o(s) nome(s) da(s) sessão(ões) a(s) qual(is) o solicitante deseja fazer parte. | 2. O sistema deve verificar se o participante (username) faz parte ou não da sessão: <ol style="list-style-type: none"> <li>2.1. caso faça parte, deve retornar uma mensagem de erro e rejeitar o pedido de inscrição;</li> <li>2.2. caso não faça parte:               <ol style="list-style-type: none"> <li>2.2.1. incluir caso de uso controlar inscrição;</li> <li>2.2.2. incluir caso de uso enviar mensagem.</li> </ol> </li> </ol> |



**Caso de Uso:** Solicitar Remoção da Sessão.

**Finalidade:** Este caso de uso tem a finalidade de permitir que as pessoas registradas no sistema, os participantes, solicitem sua remoção de alguma(s) sessão(ões) da(s) qual(is) façam parte.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante solicita a <u>remoção de sua participação em uma sessão</u> . Devem ser informados o username do controlador da sessão e o nome da sessão a qual o solicitante deseja deixar de fazer parte. | 2. O sistema deve verificar se o solicitante está em cooperação na sessão.<br>2.1. em caso afirmativo, retornar uma mensagem de erro e rejeitar o pedido;<br>2.2. em caso negativo, incluir caso de uso controlar participante por sessão e incluir caso de uso enviar mensagem. |

**Caso de Uso:** Analisar Inscrição.

**Finalidade:** Este caso de uso tem a finalidade de analisar (aceitar ou recusar) as inscrições recebidas

**Ator:** Controlador.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o controlador deseja <u>recusar as inscrições</u> que foram solicitadas pelos participantes para fazerem parte de uma sessão.                      | 2. O sistema deverá mostrar uma lista contendo todas as sessões controladas pelo solicitante.            |
| 3. O controlador deverá escolher qual sessão deseja analisar as inscrições.  | 4. O sistema deverá, então, mostrar uma lista de todas as inscrições existentes para a sessão escolhida. |
| 5. O controlador deverá informar qual participante terá sua inscrição recusada. Este pedido deve conter o username do participante que realizou a inscrição.                             | 2. Incluir caso de uso enviar mensagem.<br>3. Incluir caso de uso controlar inscrição.                   |
| 1. Este caso de uso tem início quando o controlador deseja <u>aceitar as inscrições</u> que foram solicitadas pelos participantes para que estes possam vir a fazer parte de uma sessão. | 2. O sistema deverá mostrar uma lista contendo todas as sessões controladas pelo solicitante.            |
| 3. O controlador deverá escolher qual sessão deseja analisar as inscrições.  | 4. O sistema deverá, então, mostrar uma lista de todas as inscrições existentes para a sessão escolhida. |

|  |  |
|--|--|
| 6. O controlador deverá informar qual participante terá sua inscrição aceita. Este pedido deve conter o username do participante que realizou a inscrição. | 7. Incluir caso de uso enviar mensagem<br>8. Incluir caso de uso controlar participante por sessão.<br>9. Incluir caso de uso controlar inscrição. |
|--|--|

**Caso de Uso:** Controlar Inscrição.

**Finalidade:** Este caso de uso tem a finalidade de controlar as inscrições para fazer parte de uma sessão que ainda não foram analisadas.

**Ator:** Participante/Controlador (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para aceitar a <u>inscrição de um participante em uma sessão</u> . Este pedido deve conter o username daquele que deseja fazer parte da sessão, o username do controlador da sessão e o nome da sessão. | 2. O sistema deve adicionar o participante à uma lista de inscrições pendentes . |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>aprovação da inscrição</u> de um participante. Este pedido deve conter o username do controlador, o nome da sessão e o username do solicitante.   | 2. O sistema deve remover o solicitante da lista de inscrições pendentes.        |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>recusa da inscrição</u> de um participante. Este pedido deve conter o username do controlador, o nome da sessão e o username do solicitante da inscrição.   | 2. O sistema deve remover o solicitante da lista de inscrições pendentes.        |

**Caso de Uso:** Convidar Pessoas.

**Finalidade:** Este caso de uso tem a finalidade de convidar pessoas, registradas ou não no sistema, para participar de uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante <u>convida alguma(s) pessoa(s) registrada(s) ou não</u> para participar(em) de uma sessão. Este pedido deve conter o(s) email(s) do(s) convidado(s), o nome da sessão e o username do controlador da sessão. | 2. O sistema deverá verificar se as pessoas convidadas (seus email) já fazem parte da sessão ou não.<br>2.1. em caso afirmativo, enviar mensagem de erro e rejeitar o convite;<br>2.2. caso contrário:<br>2.2.1. incluir caso de uso controlar |

|  |  |
|--|--|
|  | convidado;<br>2.2.2. incluir caso de uso enviar mensagem |
|--|--|

**Caso de Uso:** Analisar Convite.

**Finalidade:** Este caso de uso tem a finalidade de analisar convite recebido, permitindo que o participante aceite ou recuse o convite recebido para participar de uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante solicita que lhe sejam mostrados todos os convites que foram feitos a ele.<br><br>3. O participante deve escolher quais convites quer <i>confirmar</i> . Este pedido deve conter o username do controlador e o nome da sessão.           | 2. O sistema deverá mostrar uma lista contendo todas as sessões para as quais o participante foi convidado.<br><br>2. O sistema deverá fazer a confirmação do convite da seguinte forma:<br>2.1. Se o remetente do convite for o controlador da sessão:<br>2.1.1. Ponto de extensão: utilizar caso de uso controlar particip. por sessão;<br>2.2. Se o remetente do convite não for o controlador da sessão:<br>2.2.1. Ponto de extensão: utilizar caso de uso controlar inscrição;<br>2.3. Incluir caso de uso controlar convidados;<br>2.4. Incluir caso de uso enviar mensagem. |
| 1. Este caso de uso tem início quando o participante solicita que lhe sejam mostrados todos os convites que foram feitos a ele.<br><br>3. O participante deve escolher quais convites deseja <i>recusar</i> . Este pedido deve conter o username do controlador da sessão e o nome da sessão. | 2. O sistema deve mostrar uma lista contendo todas as sessões para as quais o participante foi convidado.<br><br>2. O sistema deverá fazer a recusa do convite da seguinte forma:<br>2.1. Incluir caso de uso controlar convidados.  |

**Caso de Uso:** Controlar Convidados.

**Finalidade:** Este caso de uso tem a finalidade de atualizar a lista de convidados, marcando quais convites foram aceitos e quais foram recusados.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>convidar</u> pessoas para participar de alguma sessão. As pessoas a serem convidadas podem estar registradas no sistema ou não. Este pedido deve conter o(s) email do(s) convidado(s) e o nome da sessão para a qual foram convidados. | 2. O sistema deverá incluir cada participante que receberá o convite em uma lista de convidados.       |
| 1. Este caso de uso tem início quando o sistema recebe uma <u>confirmação de um convite</u> por parte de um participante que foi previamente convidado. Este pedido deve conter o email do participante convidado e o nome da sessão.   | 2. O sistema deve manter o participante na lista de convidados, marcando que o convite foi confirmado. |
| 1. Este caso de uso tem início quando o sistema recebe um solicitação de <u>recusa do convite</u> . Este pedido deve conter o email do participante convidado e o nome da sessão.   | 2. O sistema deve manter o participante na lista de convidados, marcando que o convite foi recusado.   |

**Caso de Uso:** Controlar Participante por Sessão.

**Finalidade:** Este caso de uso tem a finalidade de controlar quem são os participantes inscritos em cada sessão.

**Ator:** Participante/Controlador (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para aceitar a <u>criar uma sessão</u> . Este pedido deve conter o username do solicitante, o username do controlador da sessão e o nome da sessão. | 2. O sistema deve adicionar o solicitante à uma lista de participantes por sessão. |
| 1. Este caso de uso tem início quando o sistema receber um pedido para <u>remover uma sessão</u> . Este pedido deve conter o username do solicitante, o username do controlador da sessão e o nome da sessão.        | 2. O sistema deve remover o solicitante  |

|  |   |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para aceitar a <u>remover a participação em uma sessão</u> . Este pedido deve conter o username do solicitante, o username do controlador da sessão e o nome da sessão.             | 2. O sistema deve remover o participante de uma lista de participante por sessão. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>aprovação da inscrição</u> de um participante. Este pedido deve conter o username do controlador, o nome da sessão e o username do solicitante da inscrição na sessão.        | 2. O sistema deve adicionar o solicitante a uma lista de participante por sessão. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>confirmação de convite</u> de um participante. Este pedido deve conter o username do controlador, o nome da sessão e o username do participante que está aceitando o convite. | 2. O sistema deve adicionar o solicitante à uma lista de participante por sessão. |

**Caso de Uso:** Estabelecer Cooperação.

**Finalidade:** Este caso de uso tem a finalidade iniciar ou encerrar a cooperação entre os membros de uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>  |
|---|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>início de uma cooperação</u> . Este pedido deve conter o username do controlador e o nome da sessão. | 2. O sistema deve verificar se a sessão escolhida pelo participante é uma sessão com horário e data prevista para comunicação compatíveis com a hora e data de acesso do participante.<br>2.1. Em caso negativo, o sistema deve retornar uma mensagem de erro ao participante e recusar o pedido de início da cooperação.<br>2.2. Em caso afirmativo,<br>2.2.1. se o participante não for o controlador da sessão, só permitir a conexão se o controlador já estiver conectado à mesma;<br>2.2.2. se o solicitante for o controlador da sessão ponto de extensão: utilizar caso de uso controlar sessões em cooperação; |

|   |   |
|---|---|
|   | <p>2.2.3. incluir caso de uso controlar participantes em cooperação;</p> <p>2.2.4. incluir caso de uso notificar presença;</p> <p>2.2.5. incluir caso de uso armazenar informações sobre a sessão;</p> <p>2.2.6. incluir caso de uso armazenar frequência.</p>  |
| <p>1. Este caso de uso tem início quando o participante deseja abandonar a comunicação com a sessão (<i>abandonar a cooperação</i>), isto é, deseja deixar de participar de uma cooperação antes do término da mesma. Este pedido deve conter o username do controlador da sessão e o nome da sessão.</p> | <p>2. Se o participante for um controlador, o sistema deverá:</p> <p>2.1. ponto de extensão: utilizar caso de uso controlar sessões em cooperação;</p> <p>2.2. ponto de extensão: utilizar caso de uso trocar mensagem;</p> <p>3. incluir caso de uso controlar participantes em cooperação.</p> <p>4. incluir caso de uso notificar presença;</p> <p>5. incluir caso de uso armazenar informações sobre a cooperação.</p> <p>6. incluir caso de uso armazenar frequência.</p>  |
| <p>1. Este use case tem início quando o controlador formaliza o <i>término da cooperação</i> (somente o controlador pode solicitar o término de uma sessão). Este pedido deve conter o username do controlador e o nome da sessão.</p>  | <p>2. O sistema deve verificar se o participante é realmente um controlador. Caso não seja, o sistema deve retornar uma mensagem de erro ao participante. Caso a solicitação de término tenha vindo de um controlador, o sistema deverá, então:</p> <p>2.1. ponto de extensão: utilizar caso de uso controlar sessões em cooperação;</p> <p>2.2. incluir caso de uso controlar participantes em cooperação;</p> <p>2.3. ponto de extensão: utilizar caso de uso trocar mensagem;</p> <p>2.4. incluir caso de uso notificar presença;</p> <p>2.5. incluir caso de uso armazenar informações sobre a cooperação;</p> <p>2.6. incluir caso de uso armazenar frequência .</p> |
| <p>1. Este caso de uso tem início quando o participante deseja <i>deixar a cooperação</i>, após o término da mesma. Este pedido deve conter o</p>   | <p>2. O sistema só poderá aceitar este serviço se a sessão foi anteriormente terminada pelo controlador. Em caso afirmativo, o sistema deverá :</p>   |

|   |  |
|---|--|
| username do controlador da sessão e o nome da sessão. | 2.1. incluir caso de uso controlar participante em cooperação;<br>2.2. incluir caso de uso notificar presença;<br>2.3. incluir caso de uso armazenar informações sobre a cooperação;<br>2.4. incluir caso de uso armazenar frequência. |
|---|--|

**Caso de Uso:** Controlar Participantes em Cooperação.

**Finalidade:** Este caso de uso tem a finalidade de manter uma lista contendo o username de todos os participantes que estão conectados a uma sessão.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para controlar os participantes de uma sessão que estão conectados à mesma; e isto ocorre quando houver o <i>início da cooperação</i> . Este pedido deve conter o username do participante, o username do controlador da sessão e o nome da sessão ao qual o participante se conectou.   | 2. O sistema deverá adicionar o username do participante à uma lista de participantes em cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para remover o participante de uma sessão de uma lista de participantes ativos na sessão; e isto ocorre quando o participante <i>abandonar a cooperação</i> ou quando houver o <i>término da mesma</i> ou <i>quando o participante desejar deixar a cooperação</i> . Este pedido deve conter o username do participante que está deixando a sessão, o username do controlador da sessão e o nome da sessão da qual o participante está se desconectando. | 2. O sistema deverá remover o username do participante da lista de participantes em cooperação.      |

**Caso de Uso:** Votar.

**Finalidade:** Este caso de uso tem a finalidade de realizar pesquisas a serem respondidas pelos participantes da sessão.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido do sistema para iniciar a <u>votação</u> . Este pedido deve conter o username do participante, o username do controlador e o nome da sessão ao qual o participante se conectou. | 2. O sistema deverá mostrar ao participante a pergunta com suas alternativas para que o participante escolha sua resposta. |
| 3. O participante deve escolher a resposta. Este pedido deve conter a resposta do participante.  | 4. Ponto de extensão: controlar votação  |

**Caso de Uso:** Controlar Votação.

**Finalidade:** Este caso de uso tem a finalidade de controlar a votação, isto é, armazenar as perguntas com suas respectivas respostas e receber as respostas dos participantes durante a votação.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para iniciar a <u>elaborar perguntas para votação</u> . Este pedido deve conter o username do participante, o username do controlador, o nome da sessão ao qual o participante se conectou, a pergunta e suas respectivas alternativas. | 2. O sistema deverá armazenar no sistema a pergunta com suas respectivas alternativas. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido de <u>votação</u> . Este pedido deve conter o username do controlador, o nome da sessão, o username do participante, a pergunta e a resposta escolhida pelo participante.   | 2. O sistema deve totalizar a resposta dada pelo participante.                         |



**Caso de Uso:** Controlar Sessões em Cooperação.

**Finalidade:** Este caso de uso tem a finalidade de manter uma lista contendo as sessões que estão em cooperação.

**Ator:** Controlador (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para controlar as sessões que estão em cooperação; e isto ocorre quando houver o <i>início da cooperação</i> . Este pedido deve conter o username do controlador da sessão e o nome da sessão a qual o participante se conectou.                                | 2. O sistema deverá adicionar o username do controlador e o nome da sessão a uma lista de sessões em cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para remover uma sessão de uma lista de sessões em cooperação; e isto ocorre quando o participante <i>abandonar a cooperação</i> . Este pedido deve conter o username do controlador da sessão e o nome da sessão da qual o participante está se desconectando. | 2. O sistema deverá remover o username do controlador e o nome da sessão da lista de sessões em cooperação.      |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para remover uma sessão de uma lista de sessões em cooperação; e isto ocorre quando o participante <i>terminar a cooperação</i> . Este pedido deve conter o username do controlador da sessão e o nome da sessão da qual o participante está se desconectando.  | 2. O sistema deverá remover o username do controlador e o nome da sessão da lista de sessões em cooperação.      |

**Caso de Uso:** Armazenar Informações sobre a Cooperação.

**Finalidade:** Este caso de uso tem a finalidade de armazenar informações sobre a cooperação tais como participantes que iniciaram a cooperação, hora de início e hora de saída.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>                    |
|---|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>iniciar a cooperação</u> . Este pedido deve conter o o username do controlador da sessão, o nome da sessão, a data e a hora de início da cooperação.   | 2. O sistema deverá armazenar as informações. |
| 1. Este use case tem início quando o sistema recebe um pedido para <u>finalizar a cooperação</u> ( <u>abandonar a cooperação</u> , quando o solicitante for o controlador da sessão ou <u>terminar a cooperação</u> ). Este pedido deve conter o o username do controlador, o nome da sessão, a data e hora de saída da cooperação. | 2. O sistema deverá armazenar as informações. |

**Caso de Uso:** Armazenar Frequência.

**Finalidade:** Este caso de uso tem a finalidade de controlar a frequência dos participantes de uma sessão.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>controlar a frequência do participante</u> (quando houver o início, término, abandono ou quando o participante deixar a cooperação). Este pedido deve conter o username, o username do controlador da sessão, o nome da sessão, a data e a hora de entrada e saída do participante. | 2. O sistema deve armazenar informações sobre a presença do participante em uma sessão, a data e hora de início e de saída da cooperação. |

**Caso de Uso:** Selecionar Mídia.

**Finalidade:** Este caso de uso tem a finalidade de mostrar as mídias disponíveis para cada sessão, permitir que o participante selecione quais mídias serão utilizadas por ele, iniciar, suspender, reiniciar ou terminar a transmissão/recepção de áudio/vídeo.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante deseja que lhe sejam mostradas quais mídias podem ser utilizadas na sessão.<br><br>3. O participante deve <i>selecionar</i> uma ou mais <i>mídias que deseja utilizar</i> , a partir de uma relação que lhe é apresentada. Este pedido deve conter a(s) mídia(s) selecionada(s).                 | 2. O sistema deve mostrar quais são as mídias que poderão ser utilizadas pelos participantes da sessão desejada.<br><br>4. Incluir caso de uso controlar mídia por participante. |
| 1. Este caso de uso tem início quando o participante deseja que lhe sejam mostradas quais mídias foram selecionadas para serem utilizadas durante a cooperação.<br><br>3. Este caso de uso tem início quando o participante deseja <i>selecionar</i> uma ou mais <i>mídias que deseja deixar de utilizar</i> . Este pedido deve conter a(s) mídia(s). | 2. O sistema deve mostrar quais mídias estão disponíveis para serem utilizadas pelos participantes.<br><br>4. Incluir caso de uso controlar mídia por participante.              |

**Caso de Uso:** Controlar Mídia por Participante.

**Finalidade:** Este caso de uso tem a finalidade de informar o sistema sobre quais mídias foram selecionadas pelo participante para serem utilizadas por este durante uma cooperação.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o sistema recebe um pedido para armazenar quais <i>mídias</i> foram <i>selecionadas por um participante por sessão</i> . Este pedido deve conter o username do participante, o username do controlador da sessão, o nome da sessão à qual o participante está conectado e as mídias que foram selecionadas por ele. | 2. O sistema deverá adicionar as mídias selecionadas à uma lista de mídias por participante. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <i>deixar de utilizar mídias durante uma</i>  | 2. O sistema deverá remover as mídias selecionadas à uma lista de mídias por participante.   |

|  |  |
|--|--|
| <i>cooperação</i> . Este pedido deve conter o username do participante, o username do controlador, o nome da sessão e a(s) mídia(s) a serem removidas. |  |
|--|--|

**Caso de Uso:** Trocar Mensagem.

**Finalidade:** Este caso de uso tem a finalidade de transmitir/receber mensagens de texto à/de outros participantes conectados simultaneamente a uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>   |
|--|--|
| <ol style="list-style-type: none"> <li>Este caso de uso tem início quando o participante deseja <i>enviar mensagem</i> de texto para outros participantes conectado simultaneamente a uma sessão.</li> <li></li> <li>O participante escolhe para qual(is) participante(s) deseja transmitir a mensagem e digita a mensagem a ser transmitida na área apropriada. Esta requisição deve conter o(s) username(s) do(s) destinatário(s) e a mensagem a ser enviada.</li> </ol> | <ol style="list-style-type: none"> <li>O sistema deve mostrar quais são os participantes que estão cooperando (ponto de extensão: utilizar caso de uso visualizar participantes em cooperação).</li> <li></li> <li></li> <li>O sistema inicia a transmissão da mensagem.</li> <li>Ponto de extensão: utilizar caso de uso notificar mensagem.</li> </ol> |
| <ol style="list-style-type: none"> <li>Este caso de uso tem início quando o participante está cooperando com outros participantes e deve <i>receber</i> qualquer <i>mensagem</i> que lhe tenha sido enviada por algum dos outros participantes que estão cooperando na mesma sessão.</li> </ol>  | <ol style="list-style-type: none"> <li>O sistema deve verificar se existe(m) mensagem(ns) que foi(ram) enviada(s) ao participante para esta sessão.</li> <li></li> <li>Em caso afirmativo, o sistema inicia a recepção da mensagem.</li> </ol>   |

**Caso de Uso:** Gravar Trechos da Conferência.

**Finalidade:** Este caso de uso tem a finalidade de permitir a gravação de trechos da conferência pelos participantes.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| <ol style="list-style-type: none"> <li>Este caso de uso tem início quando o participante deseja <i>gravar trechos da conferência</i>. Esta requisição deve conter o username do controlado e o nome da sessão.</li> <li></li> <li>O participante escolhe de qual(is) participante(s) deseja gravar. Esta requisição deve conter o(s) username(s) do(s) transmissor(es).</li> </ol> | <ol style="list-style-type: none"> <li>O sistema deve mostrar quais são os participantes que estão transmitindo (ponto de extensão: utilizar caso de uso visualizar participantes em transmissão).</li> <li></li> <li></li> <li>O sistema inicia a gravação da mídia contínua.</li> </ol> |

**Caso de Uso:** Transferir Arquivos.

**Finalidade:** Este caso de uso tem a finalidade de permitir a transferência de arquivos entre os participantes de uma sessão.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante deseja <i>transferir arquivos</i> . Esta requisição deve conter o username do controlador e o nome da sessão.          | 2. O sistema deve mostrar quais são os participantes que estão em cooperação (ponto de extensão: utilizar caso de uso visualizar participantes em cooperação). |
| 3. O participante escolhe para quem deseja enviar o arquivo. Esta requisição deve conter o(s) username(s) do(s) destinatário(s) e o(s) arquivo(s) a ser(em) transferido(s). | 4. O sistema inicia a transferência do(s) arquivo(s).  |

**Caso de Uso:** Realizar Anotações.

**Finalidade:** Este caso de uso tem a finalidade de permitir a realização de anotações pelos participantes durante uma cooperação.

**Ator:** Participante.

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>   |
|---|--|
| 1. Este caso de uso tem início quando o participante deseja <i>realizar anotações</i> durante a cooperação. Esta requisição deve conter o username do controlador, o nome da sessão, o username do participante e o texto referente à anotação. | 2. O sistema deve disponibilizar uma área de edição onde o texto possa ser digitado, alterado e gravado. |

**Caso de Uso:** Enviar/Receber Áudio.

**Finalidade:** Este caso de uso tem a finalidade de iniciar, terminar, suspender ou reiniciar o envio e a recepção de áudio durante uma cooperação.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o participante deseja <u>iniciar a recepção de áudio</u> . Este pedido deve conter o username do participante de quem se deseja receber (emissor).                   | 2. O sistema deve estabelecer um canal de comunicação entre o receptor e o emissor e iniciar a recepção.<br>3. Incluir caso de uso controlar receptor por participante. |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>terminar a recepção de áudio</u> . Este pedido deve conter o username de quem se deseja terminar (emissor).                      | 2. O sistema deve encerrar a recepção de áudio.<br>3. Incluir caso de uso controlar receptor por participante.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>suspender a recepção de áudio</u> . Este pedido deve conter o username de quem se deseja suspender (emissor).                    | 2. O sistema deve suspender a recepção de áudio.<br>3. Incluir caso de uso controlar receptor por participante.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>reiniciar a recepção de áudio</u> . Este pedido deve conter o username do participante de quem se deseja reiniciar (emissor). | 2. O sistema deve reiniciar a recepção de áudio.<br>3. Incluir caso de uso controlar receptor por participante.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>iniciar a transmissão de áudio</u> . Este pedido deve conter a mídia contínua a ser transmitida.                              | 2. O sistema deve iniciar a transmissão de áudio.<br>3. Incluir caso de uso controlar transmissor.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>terminar a transmissão de áudio</u> .  | 2. O sistema deve encerrar a transmissão de áudio.<br>3. Incluir caso de uso controlar transmissor.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>suspender a transmissão de áudio</u> .   | 2. O sistema deve suspender a transmissão de áudio.<br>3. Incluir caso de uso controlar transmissor.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>reiniciar a transmissão de áudio</u> . Este pedido deve conter a mídia contínua a ser transmitida.                            | 2. O sistema deve reiniciar a transmissão de áudio.<br>3. Incluir caso de uso controlar transmissor.  |

**Caso de Uso:** Enviar/Receber Vídeo.

**Finalidade:** Este caso de uso tem a finalidade de iniciar, terminar, suspender ou reiniciar o envio e a recepção de vídeo durante uma cooperação.

**Ator:** Participante.

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o participante deseja <u>iniciar a recepção de vídeo</u> . Este pedido deve conter o username do participante de quem se deseja receber (emissor).                   | 2. O sistema deve estabelecer um canal de comunicação entre o receptor e o emissor e iniciar a recepção.<br>3. Incluir caso de uso controlar receptor por participante. |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>terminar a recepção de vídeo</u> . Este pedido deve conter o username de quem se deseja terminar (emissor).                      | 2. O sistema deve encerrar a recepção de vídeo.<br>3. Incluir caso de uso controlar receptor por participante.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>suspender a recepção de vídeo</u> . Este pedido deve conter o username de quem se deseja suspender (emissor).                    | 2. O sistema deve suspender a recepção de vídeo.<br>3. Incluir caso de uso controlar receptor por participante.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>reiniciar a recepção de vídeo</u> . Este pedido deve conter o username do participante de quem se deseja reiniciar (emissor). | 2. O sistema deve reiniciar a recepção de vídeo.<br>3. Incluir caso de uso controlar receptor por participante.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>iniciar a transmissão de vídeo</u> . Este pedido deve conter a mídia contínua a ser transmitida.                              | 2. O sistema deve iniciar a transmissão de vídeo.<br>3. Incluir caso de uso controlar transmissor.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>terminar a transmissão de vídeo</u> .  | 2. O sistema deve encerrar a transmissão de vídeo.<br>3. Incluir caso de uso controlar transmissor.   |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação <u>suspender a transmissão de vídeo</u> .   | 2. O sistema deve suspender a transmissão de vídeo.<br>3. Incluir caso de uso controlar transmissor.  |
| 1. Este caso de uso tem início quando o sistema recebe uma solicitação de <u>reiniciar a transmissão de vídeo</u> . Este pedido deve conter a mídia contínua a ser transmitida.                            | 2. O sistema deve reiniciar a transmissão de vídeo.<br>3. Incluir caso de uso controlar transmissor.  |

**Caso de Uso:** Controlar Receptor por Transmissor.

**Finalidade:** Este caso de uso tem a finalidade de manter informações sobre quais participantes estão recebendo informações de áudio ou vídeo durante uma cooperação e de quem estão recebendo essas informações.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>   | <i>Resposta do Sistema</i>  |
|---|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>iniciar a recepção de áudio</i> ou <i>iniciar a recepção de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do participante que deseja receber (receptor) áudio ou vídeo e o username do participante de quem se deseja receber (emissor).                  | 2. O sistema deverá adicionar o participante que está recebendo áudio ou vídeo a uma lista de receptores. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>suspender a recepção de áudio</i> ou <i>suspender a recepção de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do participante que deseja receber (receptor) áudio ou vídeo e o username do participante de quem se deseja suspender a recepção (emissor). | 2. O sistema deverá remover o participante que está recebendo áudio ou vídeo da lista de receptores.      |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>terminar a recepção de áudio</i> ou <i>terminar a recepção de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do participante que deseja receber (receptor) áudio ou vídeo e o username do participante de quem se deseja terminar a recepção (emissor).    | 2. O sistema deverá remover o participante que está recebendo áudio ou vídeo da lista de receptores.      |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>reiniciar a recepção de áudio</i> ou <i>reiniciar a recepção de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do participante que deseja receber (receptor) áudio ou vídeo e o username do participante de quem se deseja reiniciar a recepção (emissor). | 2. O sistema deverá adicionar o participante que está recebendo áudio ou vídeo a uma lista de receptores. |



**Caso de Uso:** Controlar Transmissor.

**Finalidade:** Este caso de uso tem a finalidade de manter informações sobre quais participantes estão aptos a transmitir áudio ou vídeo durante uma cooperação.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>iniciar a transmissão de áudio</i> ou <i>iniciar a transmissão de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão e o username do transmissor.     | 2. O sistema deverá adicionar o participante que está transmitindo áudio ou vídeo a uma lista de transmissores. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>suspender a transmissão de áudio</i> ou <i>suspender a transmissão de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do transmissor.  | 2. O sistema deverá remover o participante que está transmitindo áudio ou vídeo da lista de transmissores.      |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>terminar a transmissão de áudio</i> ou <i>terminar a transmissão de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão, o username do transmissor.    | 2. O sistema deverá remover o participante que está transmitindo áudio ou vídeo da lista de transmissores.      |
| 1. Este caso de uso tem início quando o sistema recebe um pedido <i>reiniciar a transmissão de áudio</i> ou <i>reiniciar a transmissão de vídeo</i> . Este pedido deve conter o username do controlador, o nome da sessão e o username do transmissor. | 2. O sistema deverá adicionar o participante que está transmitindo áudio ou vídeo a uma lista de transmissores. |

**Caso de Uso:** Enviar Mensagem Externa.

**Finalidade:** Este caso de uso tem a finalidade de enviar mensagem aos participantes, utilizando um serviço externo de correio eletrônico.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe uma <i>solicitação de inscrição na sessão</i> .               | 2. O sistema deve enviar uma mensagem ao controlador utilizando um serviço externo de correio eletrônico. |
| 1. Este caso de uso tem início quando o sistema recebe uma <i>solicitação de remoção da participação na sessão</i> . | 2. O sistema deve enviar uma mensagem ao controlador utilizando um serviço externo de correio eletrônico. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para  | 2. O sistema deve enviar uma mensagem aos convidados utilizando um serviço                                |

|  |   |
|--|---|
| <i>convidar pessoas para participarem de uma sessão.</i>   | externo de correio eletrônico.  |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>aceitar convite</u> recebido. | 2. O sistema deve enviar uma mensagem ao controlador utilizando um serviço externo de correio eletrônico. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>recusar convite</u> recebido. | 2. O sistema deve enviar uma mensagem ao controlador utilizando um serviço externo de correio eletrônico. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>aceitar inscrição</u> .       | 2. O sistema deve enviar uma mensagem ao solicitante utilizando um serviço externo de correio eletrônico. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>recusar inscrição</u> .       | 2. O sistema deve enviar uma mensagem ao solicitante utilizando um serviço externo de correio eletrônico. |

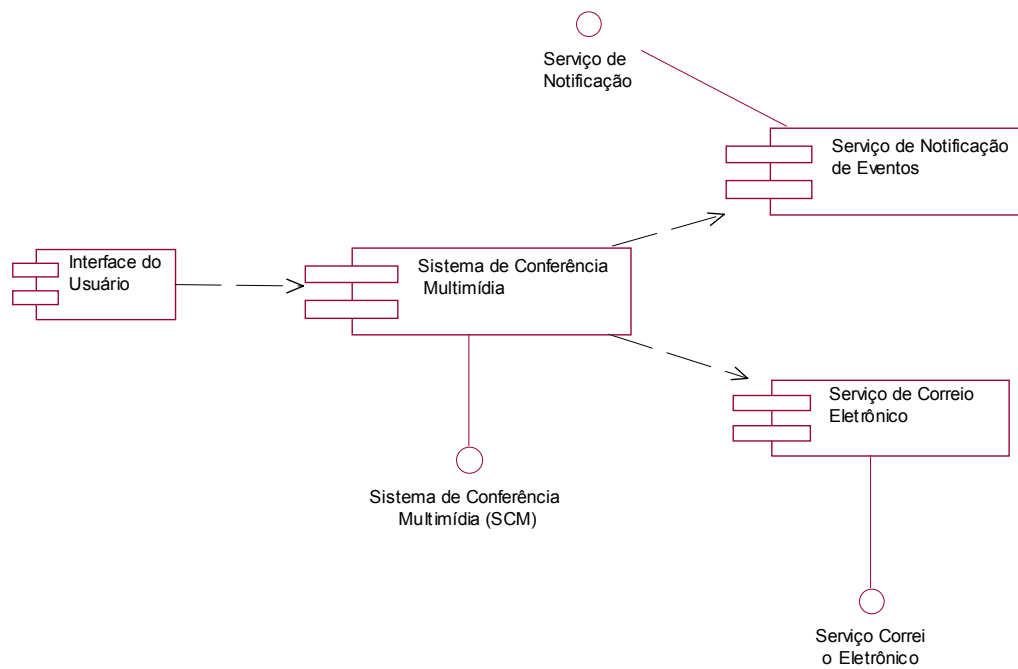
### Caso de Uso: Notificar Participante

**Finalidade:** Este caso de uso tem a finalidade de notificar os participantes sobre a presença ou saída de participantes de uma cooperação e sobre a chegada de mensagens síncronas durante a cooperação.

**Ator:** Participante (indiretamente).

| <i>Ação do Ator</i>  | <i>Resposta do Sistema</i>  |
|--|---|
| 1. Este caso de uso tem início quando o sistema recebe um pedido do participante para <u>iniciar a cooperação</u> .                                  | 2. O sistema deve notificar o(s) participante(s) sobre a presença de um participante na cooperação. |
| 1. Este caso de uso tem início quando o sistema recebe um pedido do participante para <u>abandonar a cooperação</u> .                                | 2. O sistema deve notificar o(s) participante(s) sobre a saída do participante da cooperação.       |
| 1. Este caso de uso tem início quando o sistema recebe um pedido do controlador para <u>terminar a cooperação</u> .                                  | 2. O sistema deve notificar o(s) participante(s) sobre a saída do participante da cooperação.       |
| 1. Este caso de uso tem início quando o sistema recebe um pedido do participante para <u>deixar a cooperação</u> .                                   | 2. O sistema deve notificar o(s) participante(s) sobre a saída do participante da cooperação.       |
| 1. Este caso de uso tem início quando o sistema recebe um pedido para <u>enviar uma mensagem síncrona</u> a um participante, durante uma cooperação. | 2. O sistema deve notificar o participante sobre a chegada de uma nova mensagem.                    |

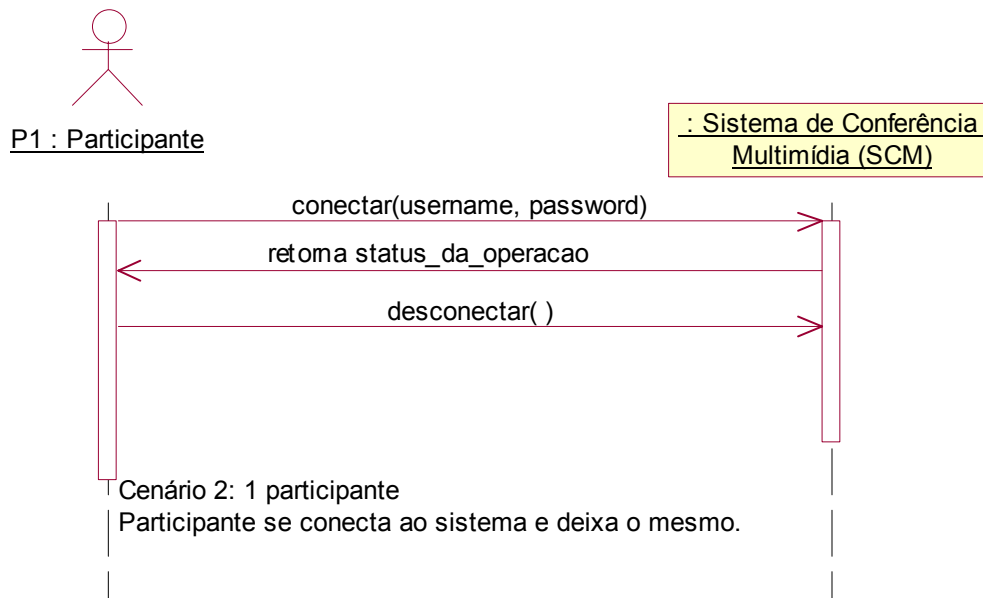
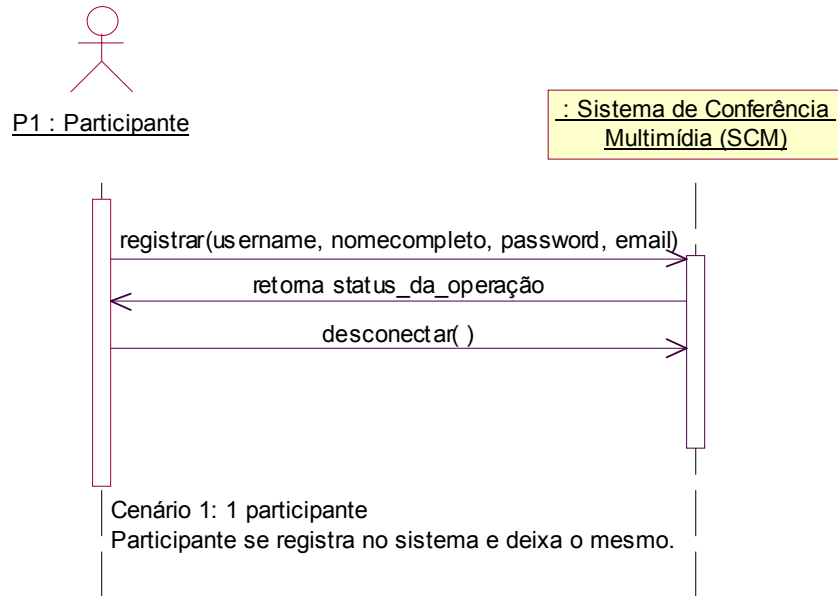
## b) Diagrama de Pacotes

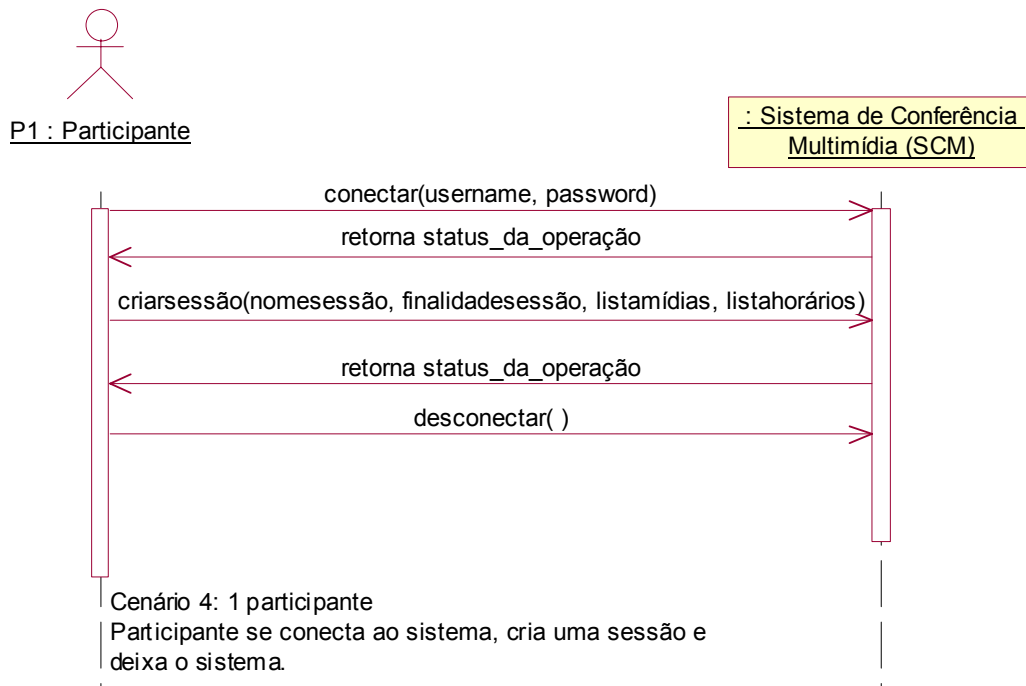
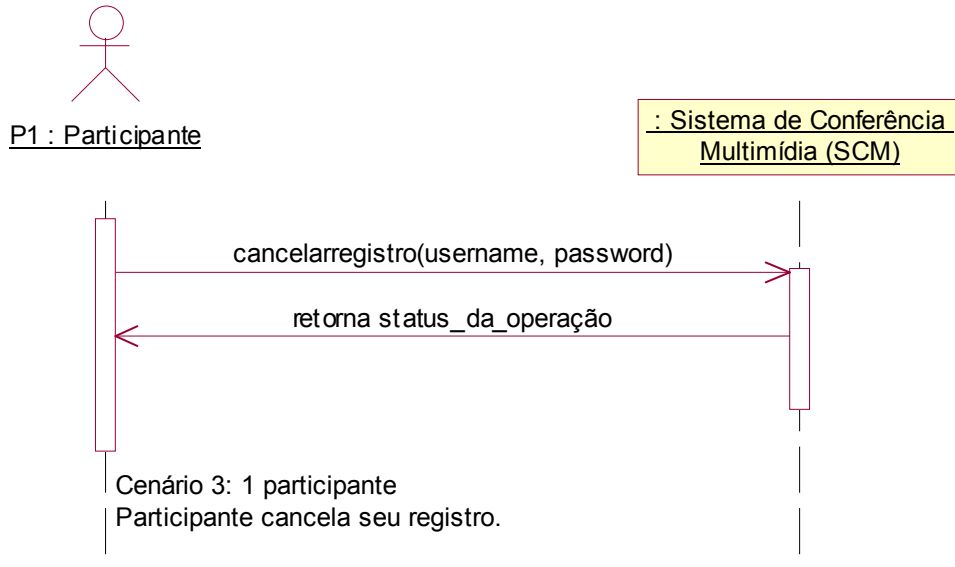
**A3.2) Visão Comportamental**

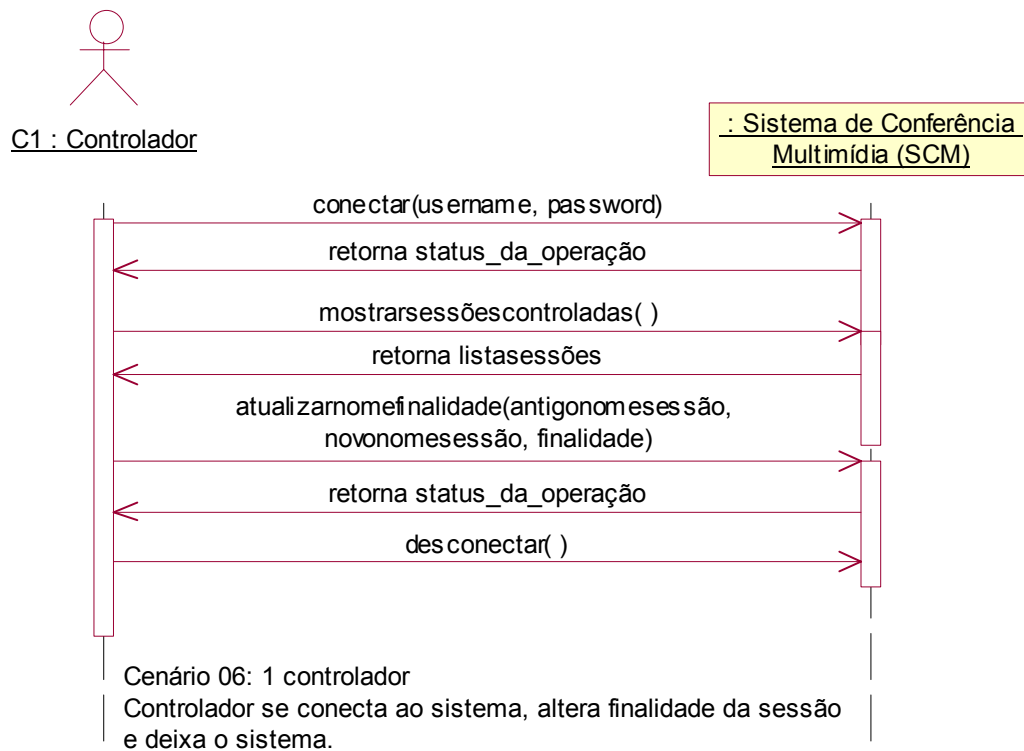
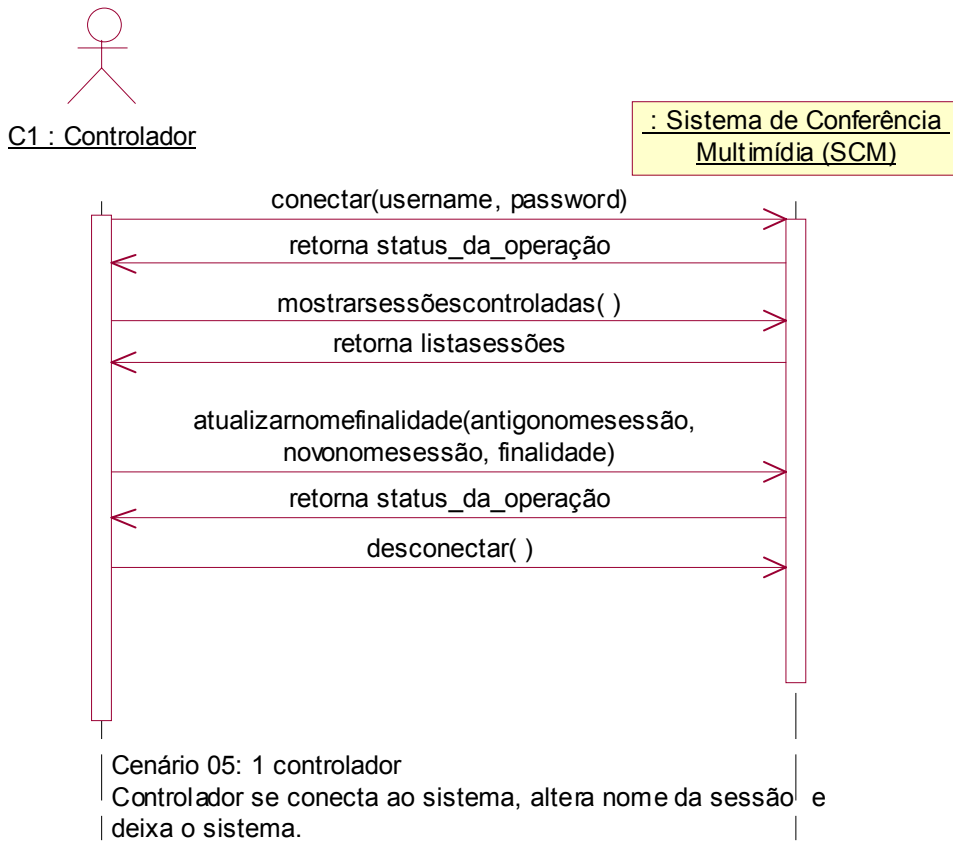
- a) Será detalhada somente no nível de componente

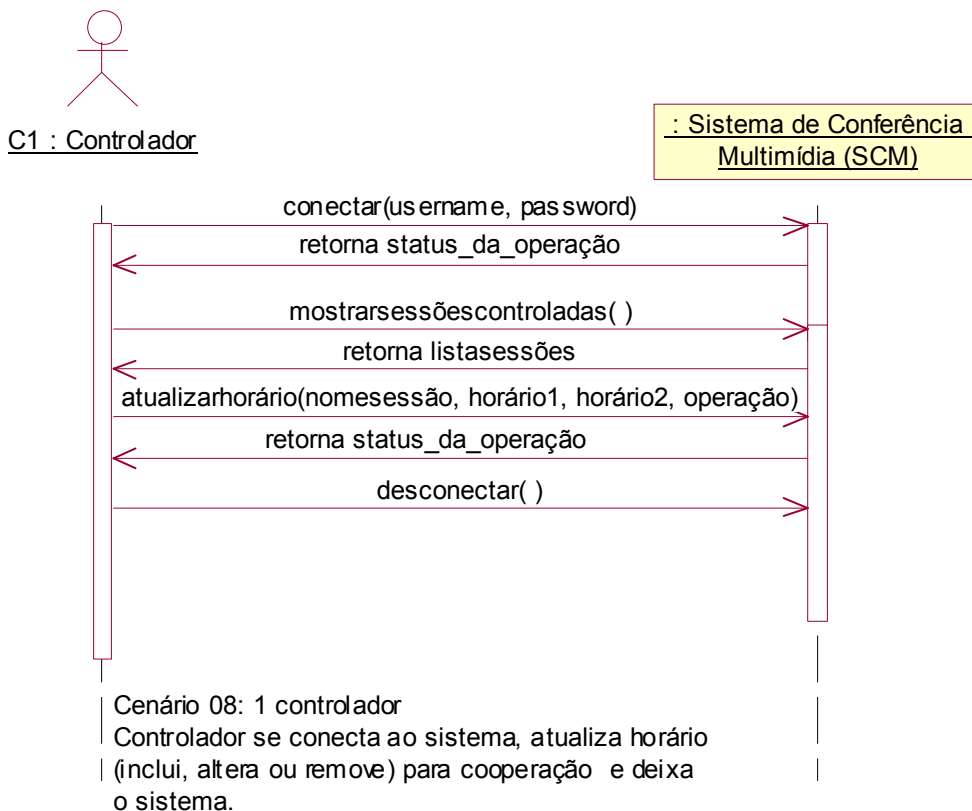
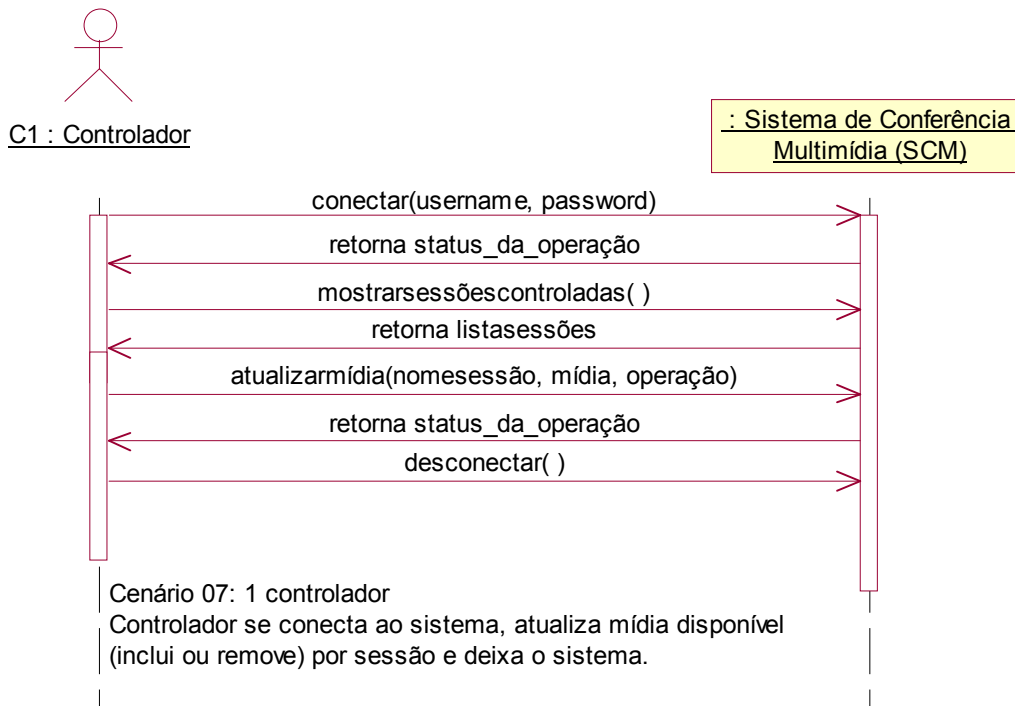
**A3.3) Visão Interacional**

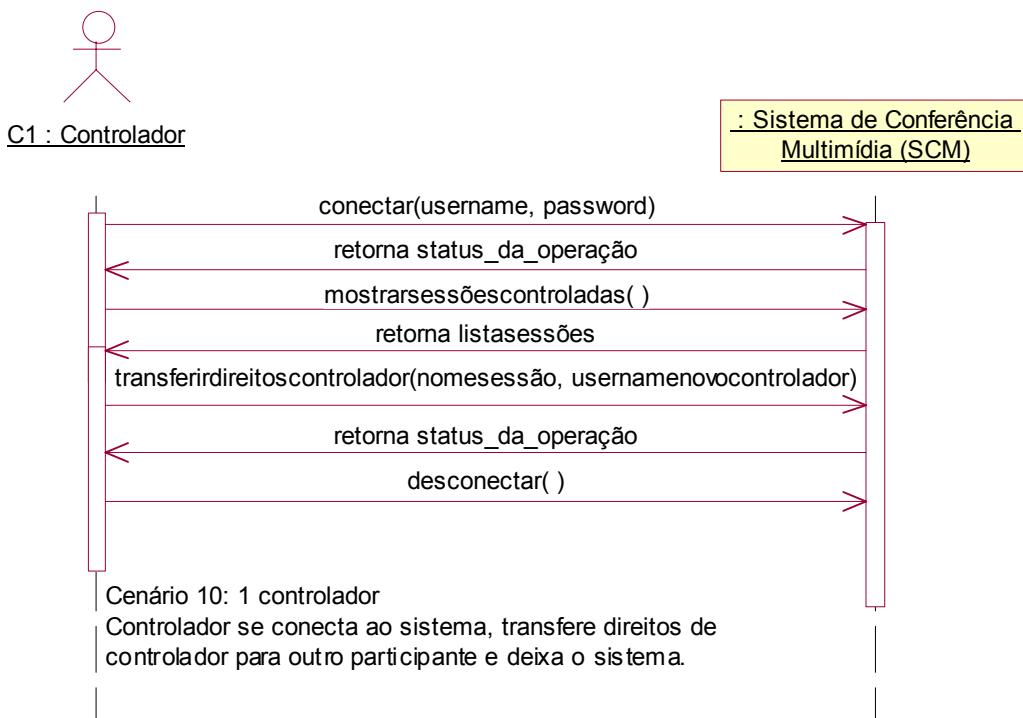
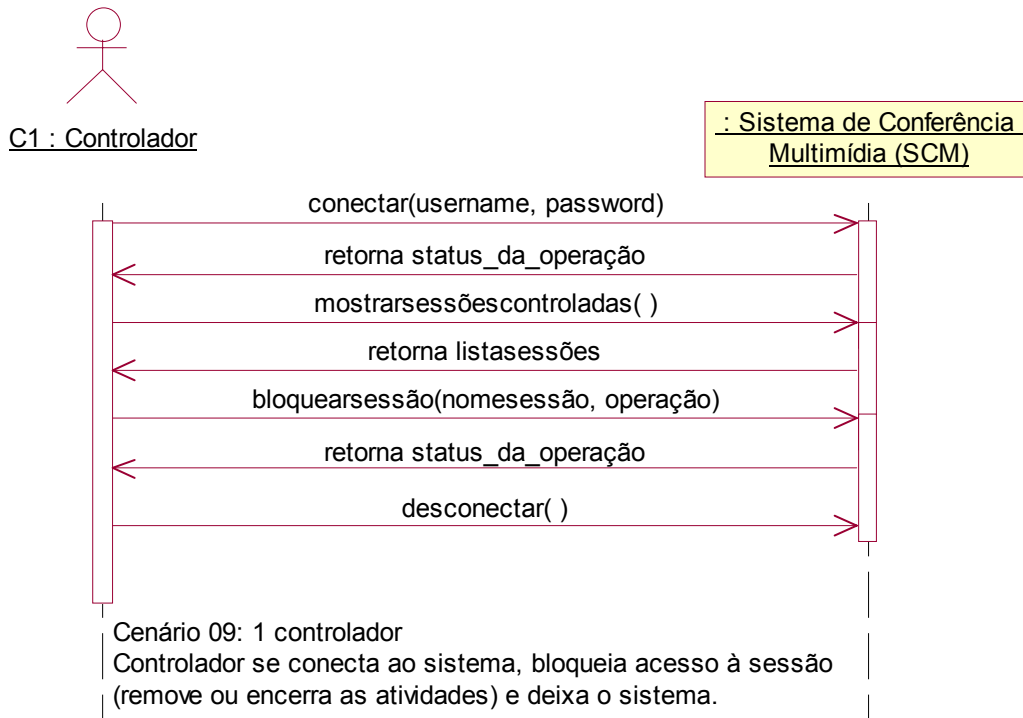
- a) Diagrama de Sequência



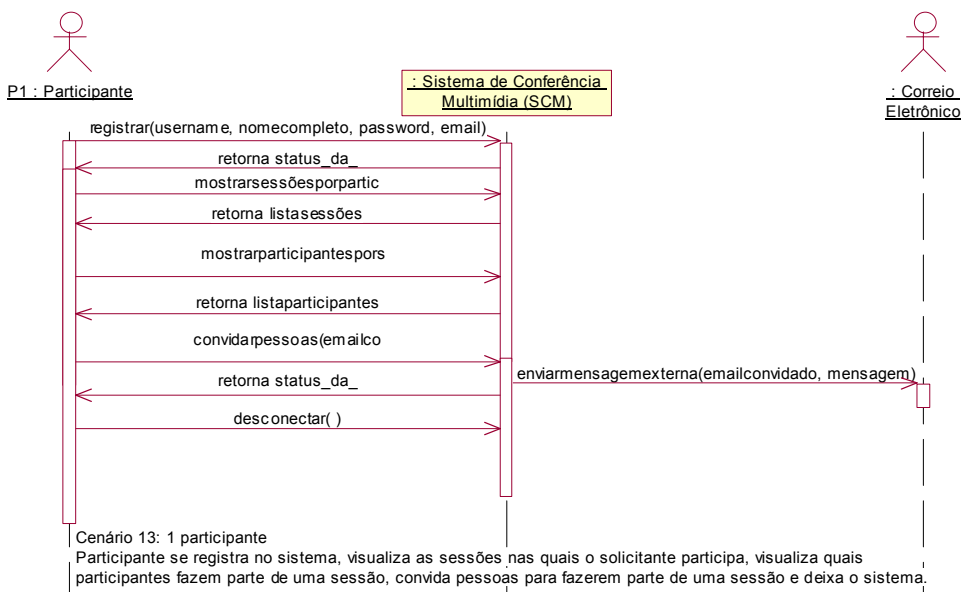
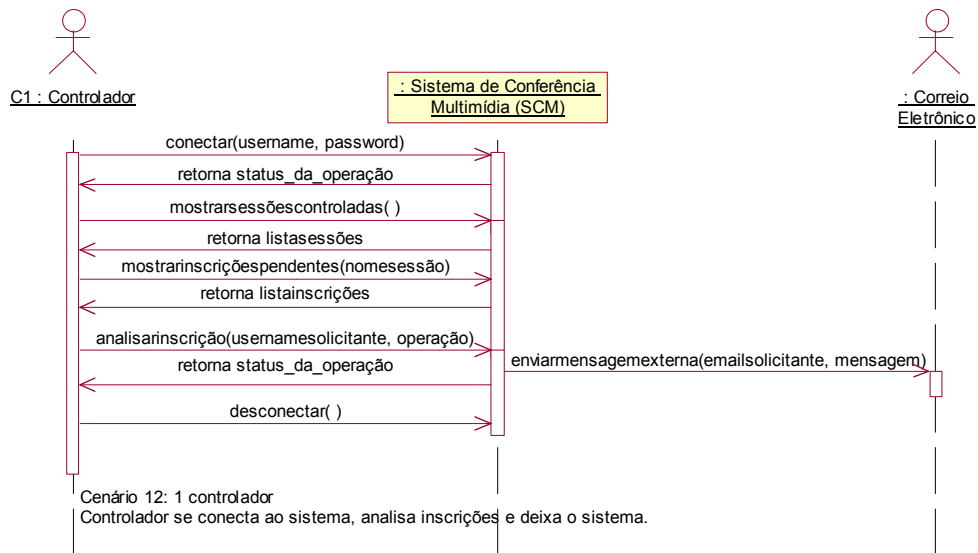
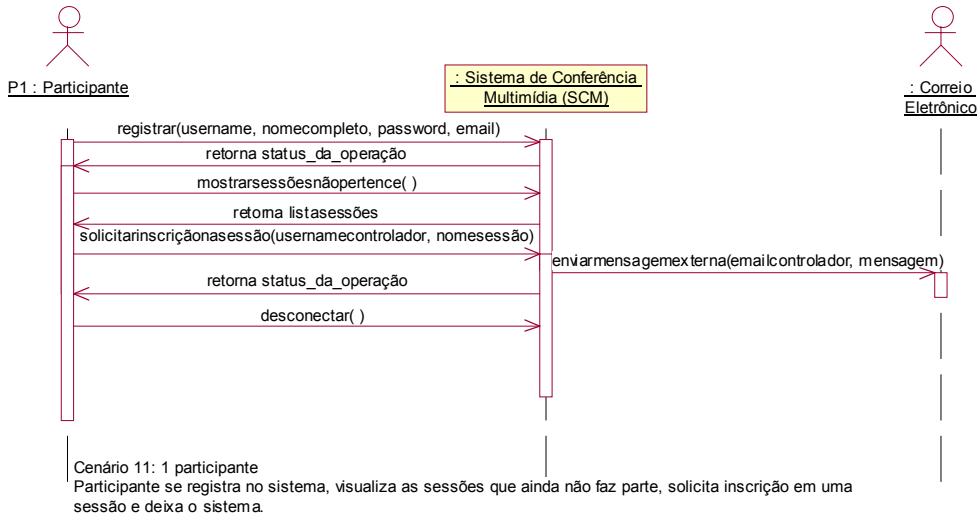


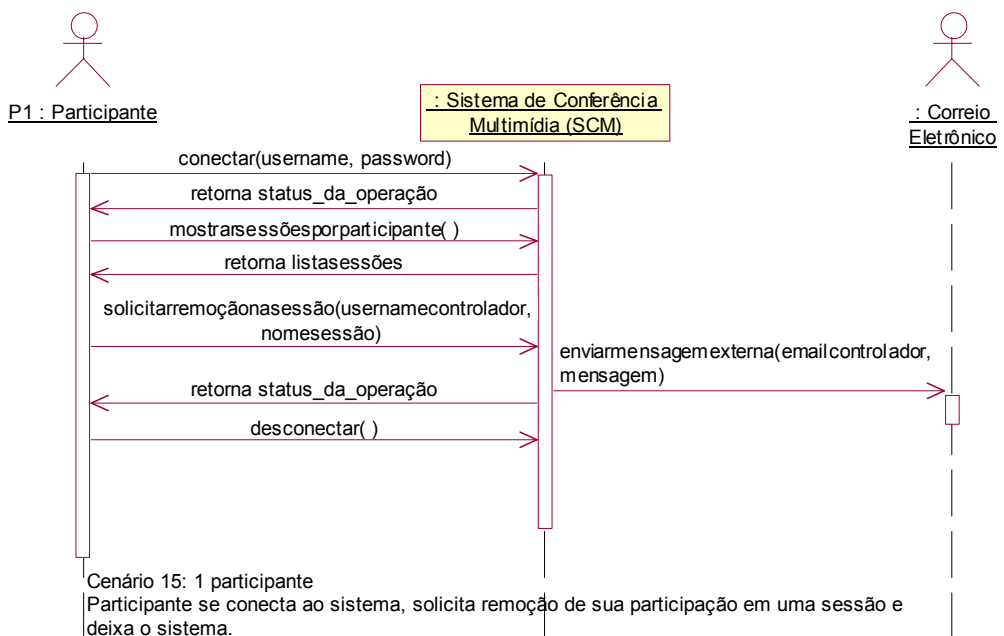
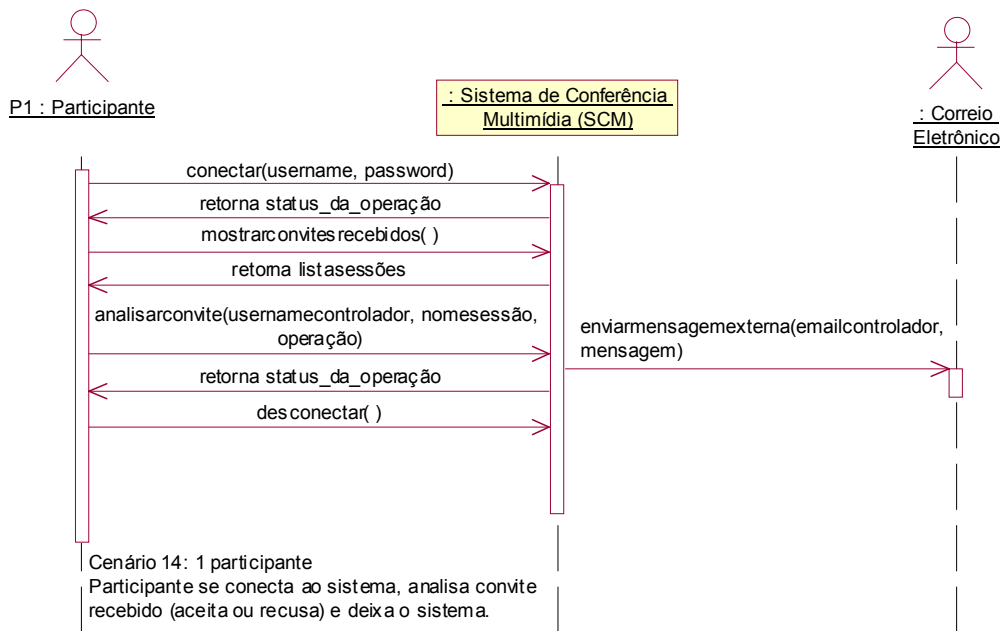


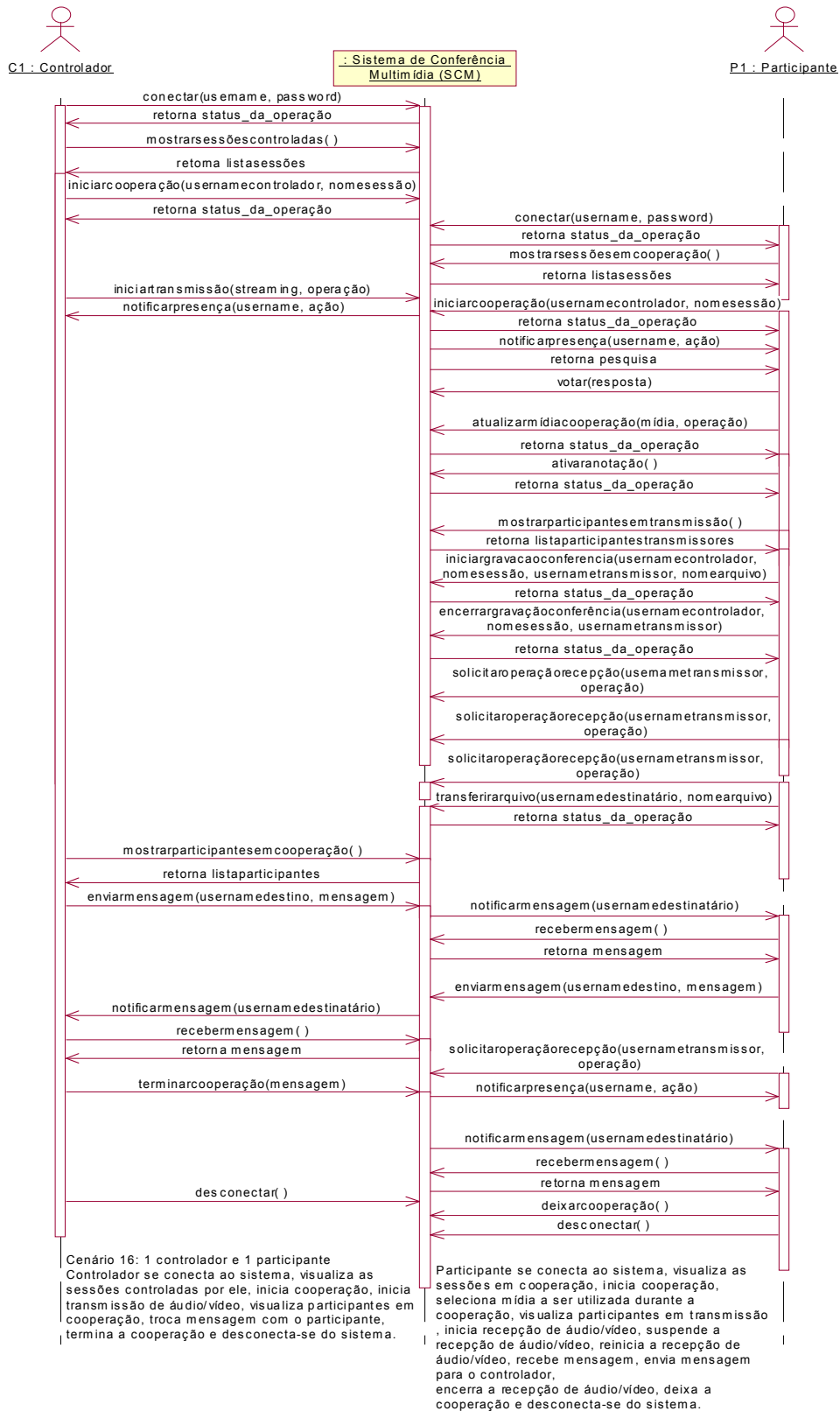


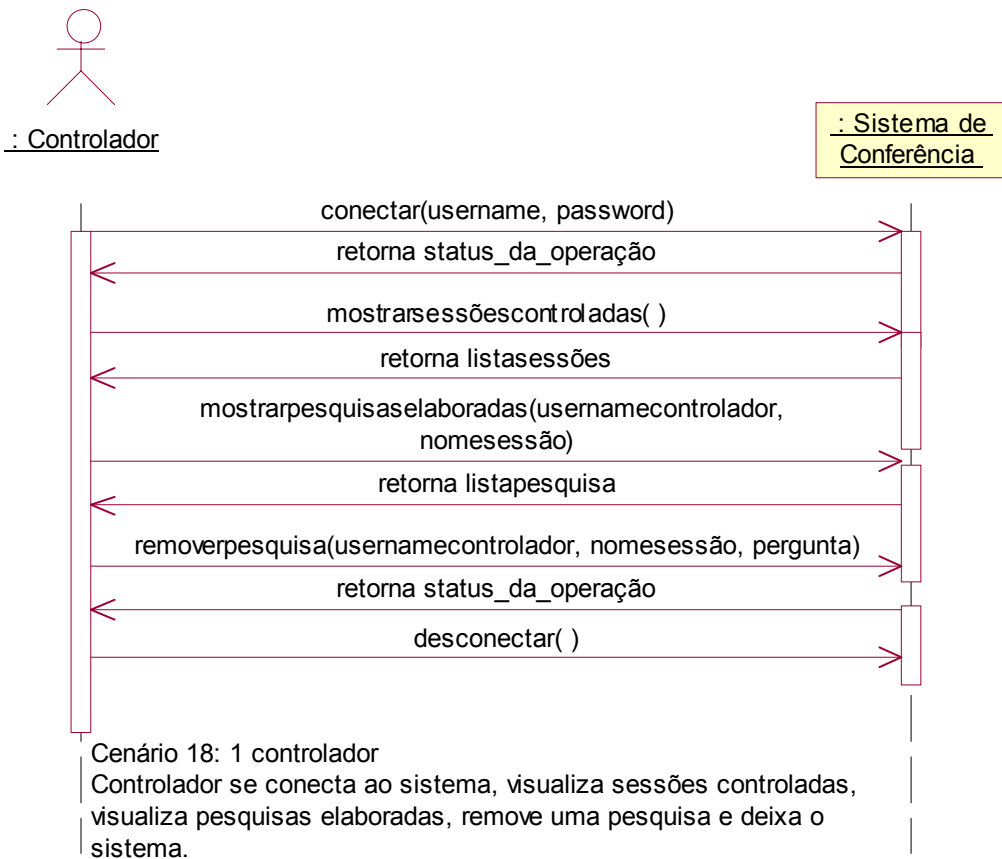
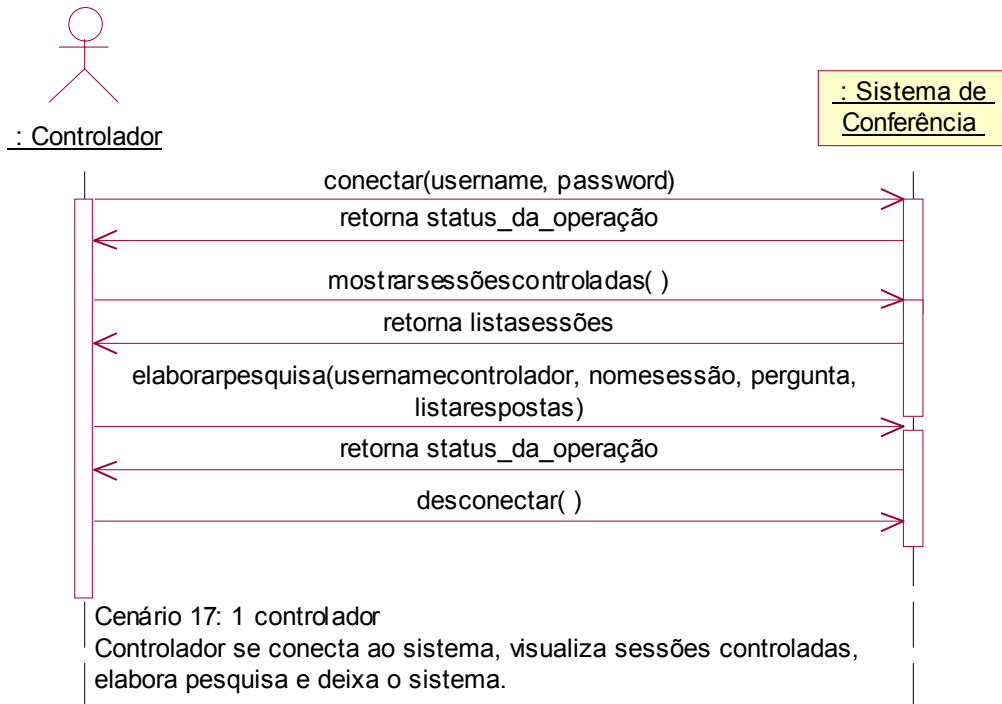












---

b) Descrição das Mensagens

*:Acessar Sistema*

**Nome:** registrar (username: String, nomecompleto: String, password: String, email: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Registrar um novo participante no sistema.

**Nome:** conectar (username: String, password: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Conectar um participante registrado ao sistema.

**Nome:** cancelarregistro (username: String, password: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Cancelar o registro de um participante no sistema.

**Nome:** desconectar ()

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Desconectar o participante do sistema.

*:Criar Sessão*

**Nome:** criarsessão (nomesessão: String, finalidadesessão: String, listamidias: Array of String, listahorários: Array of String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Criar uma nova sessão para cooperação.

---

*:Atualizar Sessão*

**Nome:** mostrarsessõescontroladas ()

**Valor de Retorno:** listasessão: Array of String

**Ator Associado:** Controlador

**Finalidade:** Mostrar quais são as sessões controladas pelo participante solicitante.

**Nome:** atualizarnomefinalidade (antigonomesessão: String, novonomesessão: String, novafilanildadesessão: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Informar o novo nome e/ou finalidade de uma sessão.

**Nome:** atualizarhorário (nomesessão: String, horário: String, novohorário: String, operação: Int)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Informar um horário, para cooperação entre os participantes da sessão a ser incluído, alterado ou removido.

**Nome:** atualizarmídia (nomesessão: String, mídia: String, operação: Int)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Informar mídia a ser incluída ou removida para uma sessão.

**Nome:** bloquearsessão (nomesessão: Array of String, operação: Int)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Informar a sessão que será removida ou fechada por um controlador

---

*:Transferir Direitos Controlador*

**Nome:** mostrarsessãocontroladas ()

**Valor de Retorno:** listasessões: Array of String

**Ator Associado:** Controlador

**Finalidade:** Mostrar quais sessões são controladas por um participante.

**Nome:** mostrarparticipantesporsessão (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** listaparticipantes: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais participantes fazem parte de uma sessões.

**Nome:** transferirdireitoscontrolador (nomesessão: String, novousername: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Transferir direitos de controlador de uma sessão para um outro participante, o qual passará a ser o novo controlador da sessão.

*:Suporte à Votação*

**Nome:** elaborarpesquisa (usernamecontrolador: String, nomesessão: String , pergunta: String, listarespostas: Array of String)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Armazenar pesquisa a ser votada posteriormente pelo participante.

**Nome:** mostrarpesquisaselaboradas (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** listapesquisas: Array of String

**Ator Associado:** Controlador

**Finalidade:** Mostrar as pesquisas que foram elaboradas pelo controlador da sessão.

---

**Nome:** removerpesquisa (usernamecontrolador: String, nomesessão: String, pergunta: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Remover uma pesquisa armazenada.

*:Solicitar Inscrição*

**Nome:** mostrarsessõesnãoopertence ()

**Valor de Retorno:** listasessões: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais são as sessões as quais o participante não pertence.

**Nome:** solicitarinscriçãonasessão (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Aceitar inscrição do participante para pertencer à uma sessão.

*:Analisar Inscrição*

**Nome:** mostrarsessõescontroladas ()

**Valor de Retorno:** listasessão: Array of String

**Ator Associado:** Controlador

**Finalidade:** Mostrar quais são as sessões controladas pelo participante solicitante.

**Nome:** mostrarinscriçõespendentes (nomesessão: String)

**Valor de Retorno:** listainscrição: Array of String

**Ator Associado:** Controlador

**Finalidade:** Apresentar as inscrições solicitadas pelos participantes que ainda não foram aceitas ou recusadas pelo controlador da sessão.



---

**Nome:** analisarinscrição (username solicitante: String, operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Controlador

**Finalidade:** Aceitar ou recusar a inscrição solicitada por um participante para fazer parte de uma sessão.

*:Convidar Pessoas*

**Nome:** mostrarsessõesporparticipante ()

**Valor de Retorno:** listasessões: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais são as sessões as quais o participante pertence.

**Nome:** mostrarparticipantesporsessão (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** listaparticipantes: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais participantes fazem parte de uma sessão.

**Nome:** convidarpessoas (emailconvidado: String, usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** status

**Ator Associado:** Participante

**Finalidade:** Convidar pessoas, registradas ou não no sistema, para participar de uma sessão.

*:Analisar Convite*

**Nome:** mostrarconvitesrecebidos ()

**Valor de Retorno:** listasessões: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais são as sessões para as quais o participante foi convidado.

---

**Nome:** analisarconvite (usernamecontrolador: String, nomesessão: String, operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Aceitar ou recusar convite para participar de uma sessão.

*:Solicitar Remoção na Sessão*

**Nome:** mostrarsessõesporparticipante ()

**Valor de Retorno:** listasessões: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais são as sessões as quais o participante pertence.

**Nome:** solicitarremoçãonasessão (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Remover participante de uma lista de participantes por sessão.

*:Enviar Mensagem*

**Nome:** enviarmensagemexterna (email: String, mensagem: String)

**Valor de Retorno:** void

**Ator Associado:** Participante (indiretamente)

**Finalidade:** Enviar mensagem através de um serviço externo de correio eletrônico.

*:Estabelecer Cooperação*

**Nome:** mostrarsessõessemcooperação ()

**Valor de Retorno:** listasessõessemcooperação: Array of String

**Ator Associado:** Participante

---

**Finalidade:** Mostrar quais são as sessões o participante pertence e que estão em cooperação.

**Nome:** mostrarsessõescontroladas ()

**Valor de Retorno:** listasessão: Array of String

**Ator Associado:** Controlador

**Finalidade:** Mostrar quais são as sessões controladas pelo participante solicitante.

**Nome:** iniciarcooperação (usernamecontrolador: String, nomesessão: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Iniciar a cooperação entre os participantes de uma sessão.

**Nome:** abandonarcooperação (mensagem: String)

**Valor de Retorno:** void

**Ator Associado:** Participante/Controlador

**Finalidade:** Abandonar a cooperação pelo participante da sessão antes do término da mesma.

**Nome:** terminarcooperação (mensagem: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Controlador

**Finalidade:** Terminar a cooperação entre os participante da sessão.

**Nome:** deixarcooperação ()

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Deixar a cooperação entre os participante da sessão após o término da mesma.

---

*:Trocar Mensagem*

**Nome:** mostrarparticipantesemcooperação ()

**Valor de Retorno:** listaparticipantes: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais são os participantes que estão em cooperação em uma determinada sessão.

**Nome:** enviarmensagem(usernamedestino: String, mensagem: String)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Enviar mensagem para qualquer participante que esteja em cooperação.

**Nome:** recebermensagem()

**Valor de Retorno:** mensagem: String

**Ator Associado:** Participante

**Finalidade:** Receber mensagem durante uma cooperação.

*:Votar*

**Nome:** votar (resposta: String)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Obter resposta do participante à uma pesquisa.

*:Selecionar Mídia Disponível a ser Utilizada*

**Nome:** mostrarmídiadisponível ()

**Valor de Retorno:** status: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quais mídias estão disponíveis para serem utilizadas durante uma cooperação.

---

**Nome:** atualizarmídiacooperação (mídia: String, operação: Int)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Informar mídia a ser incluída ou removida por participante durante uma cooperação.

*:Enviar/Receber Áudio*

**Nome:** iniciartransmissão (áudio: Stream, operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Iniciar ou reiniciar a transmissão de áudio.

**Nome:** suspendertransmissão (operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Suspender ou encerrar a transmissão de áudio.

**Nome:** solicitaroperaçãoorecepção (username transmissor: String, operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Iniciar, suspender, reiniciar ou encerrar a recepção de áudio.

**Nome:** mostrarparticipantesemtransmissão ()

**Valor de Retorno:** listaparticipantestransmissores: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quem são os participantes quem estão transmitindo áudio.

*:Enviar/Receber Vídeo*

**Nome:** iniciartransmissão (vídeo: Stream, operação: Int)

**Valor de Retorno:** void

---

**Ator Associado:** Participante

**Finalidade:** Iniciar ou reiniciar a transmissão de vídeo.

**Nome:** suspendertransmissão (operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Suspender ou encerrar a transmissão de vídeo.

**Nome:** solicitaroperaçãorecepção (usernameemissor: String, operação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante

**Finalidade:** Iniciar, suspender, reiniciar ou encerrar a recepção de vídeo.

**Nome:** mostrarparticipantesemtransmissão ()

**Valor de Retorno:** listaparticipantestransmissores: Array of String

**Ator Associado:** Participante

**Finalidade:** Mostrar quem são os participantes quem estão transmitindo vídeo.

*:Transferir Arquivo*

**Nome:** transferirarquivo (usernamedestinatário: string, nomearquivo: string)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Transferir arquivo para outro participante.

*:Realizar Anotação*

**Nome:** ativaranotação ()

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Ativar um editor de texto para permitir a gravação de anotações durante a realização da conferência.

---

*:Gravar Trechos da Conferência*

**Nome:** iniciargravaçãoconferência (usernamecontrolador: string, nomesessão: string, usernametransmissor: string, nomearquivo: string)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Iniciar a gravação da conferência.

**Nome:** encerrargravaçãoconferência (usernamecontrolador: string, nomesessão: string, usernametransmissor: string)

**Valor de Retorno:** status: Int

**Ator Associado:** Participante

**Finalidade:** Encerrar a gravação da conferência.

*:Notificar Participante*

**Nome:** notificarmensagem(usernamedestinatário: String)

**Valor de Retorno:** void

**Ator Associado:** Participante (indiretamente)

**Finalidade:** Informar participante sobre uma mensagem recebida.

**Nome:** notificarpresença(username: String, ação: Int)

**Valor de Retorno:** void

**Ator Associado:** Participante (indiretamente)

**Finalidade:** Informar participante sobre a presença de um participante na cooperação.

## c) Relação das Interfaces

*Sistema:*

registrar(username, nomecompleto, password, email)  
conectar(username, password)  
cancelarregistro(username, password)  
desconectar()  
criarsessão(nomesessão, finalidadeessão, listamídias, listahorários)  
mostrarsessõescontroladas()  
atualizarnomefinalidade(antigonomesessão, novonomesessão, finalidade  
atualizarhorário(nomesessão, horário1, horário2, operação)  
atualizarmídia(nomesessão, mídia, operação)  
bloquearsessão(nomesessão, operação)  
mostrarparticipantesporessão(usernamecontrolador, nomesessão)  
transferirdireitoscontrolador(nomesessão, nomenovocontrolador)  
elaborarpesquisa (usernamecontrolador, nomesessão, pergunta, listarespostas)  
mostrarpesquisaselaboradas (usernamecontrolador, nomesessão)  
removerpesquisa (usernamecontrolador, nomesessão, pergunta)  
mostrarsessõesnãoopertence()  
solicitarinscriçãonasessão(usernamecontrolador, nomesessão)  
mostrarinscriçõespendentes(nomesessão)  
analisarinscrição(usuariosolicitante, operação)  
mostrarsessõesporparticipante()  
convidarpessoas(emailconvidado, usernamecontrolador, nomesessão)  
mostrarconvitesrecebidos()  
analisarconvite(usernamecontrolador, nomesessão, operação)  
solicitarremoçãonasessão(usernamecontrolador, nomesessão)  
mostrarsessõesemcooperação()  
iniciarcooperação(usernamecontrolador, nomesessão)  
abandonarcooperação(mensagem)  
terminarcooperação(mensagem)  
deixarcooperação()  
mostrarparticipantesemcooperação()  
enviarmensagem(usernamedestino, mensagem)  
recebermensagem()



---

votar (resposta)

transferirarquivo (usernamedestinatário, nomearquivo)

ativaranotação()

iniciargravaçãoconferência (usernamecontrolador, nomesessão, usernametransmissor, nomearquivo)

encerrargravaçãoconferência (usernamecontrolador, nomesessão, usernametransmissor)

atualizarmídiacooperação(mídia, operação)

iniciartransmissão(áudiovídeo, operação)

suspendertransmissão(operação)

mostrarparticipantesemtransmissão()

solicitaroperaçãorecepção(usernameetransmissor, operação)

*Participante:*

notificarpresença(username, ação)

notificarmensagem(mensagem, usernamedestinatário)

*Serviço Correio Eletrônico:*

enviarmensagemexterna(email, mensagem)

## **Apêndice 4**

### ***Sistema de Conferência Multimídia – Nível de Componente***

Este apêndice apresenta a modelagem do Sistema de Conferência Multimídia (SCM) para o nível de componente. O nível de componente representa o sistema a ser desenvolvido em termos de um conjunto de componentes combinados e suas interfaces.

A modelagem deste nível consiste, basicamente, na execução das seguintes etapas:

- 1) Identificação da arquitetura a ser utilizada;
- 2) Modelagem do nível de sistema sob a perspectiva da arquitetura escolhida;
  - 2.1) Visão Estrutural
    - a) Cliente
      - a1) elaboração do diagrama de casos de uso;
      - a2) elaboração do diagrama de pacotes (componentes)
    - b) Servidor
      - b1) elaboração do diagrama de casos de uso;
      - b2) elaboração do diagrama de pacotes (componentes)
  - 2.2) Visão Comportamental
    - a) Cliente
      - a1) elaboração do diagrama de colaboração ou seqüência (as ações internas não serão detalhadas neste ponto);
    - b) Servidor
      - b1) elaboração do diagrama de colaboração ou seqüência (as ações internas não serão detalhadas neste ponto);
  - 2.3) Visão Interacional
    - a) elaboração do diagrama de colaboração ou seqüência
    - b) descrição das mensagens
    - c) relação das Interfaces
      - c1) cliente
      - c2) servidor
      - c3) entidades externas
- 3) Identificação dos Componentes a serem Desenvolvidos
- 4) Nível de Componente

#### 4.1) Visão Estrutural

##### a) Cliente

a1) elaboração do diagrama de pacotes (componentes) ou diagrama de desenvolvimento;

a2) elaboração das classes de interface;

##### b) Servidor

b1) elaboração do diagrama de pacotes (componentes) ou diagrama de desenvolvimento;

b2) elaboração das classes de interface;

#### 4.2) Visão Interacional

##### a) Cliente

a1) elaboração do diagrama de colaboração ou seqüência;

##### b) Servidor

b1) elaboração do diagrama de colaboração ou seqüência;

#### 4.3) Visão Comportamental

##### a) Cliente

a1) elaboração do diagrama de atividade;

##### b) Servidor

b1) elaboração do diagrama de atividade;

A seguir, serão apresentadas as etapas para modelagem do Sistema de Conferência Multimídia no nível de componente.

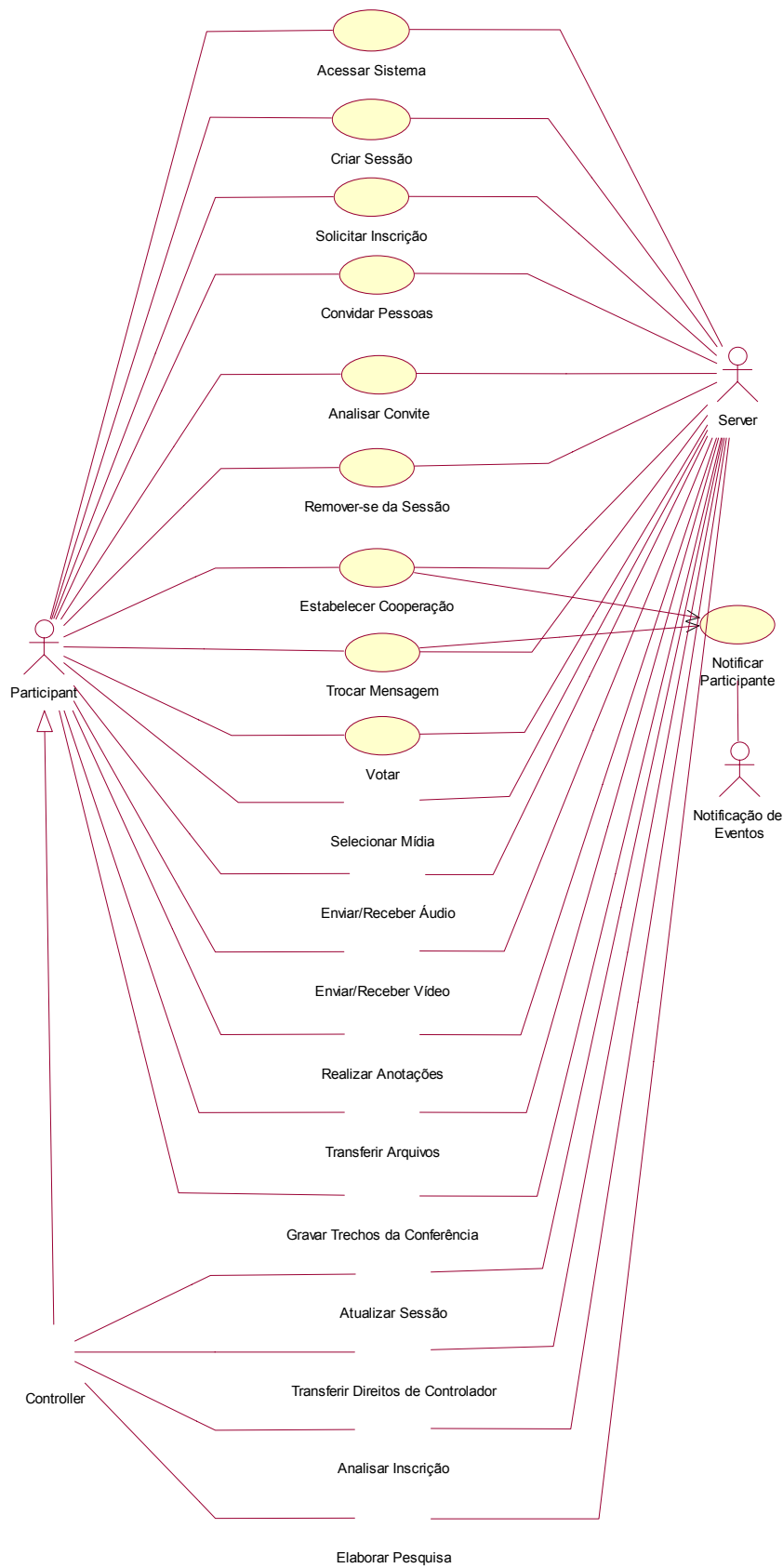
#### **A4.1) Identificação da Arquitetura a ser Utilizada**

Sendo a conferência multimídia uma aplicação distribuída, optou-se por modelá-la utilizando a arquitetura cliente-servidor. Dentre as vantagens que justificam esta escolha estão: facilidade para garantia da consistência dos dados devido ao controle centralizado, efetiva utilização dos recursos da máquina cliente e da máquina servidora, redução do problema de manutenção de código da aplicação nas estações clientes e aplicação cliente mais simples por evitar complexos algoritmos distribuídos, dentre outras.

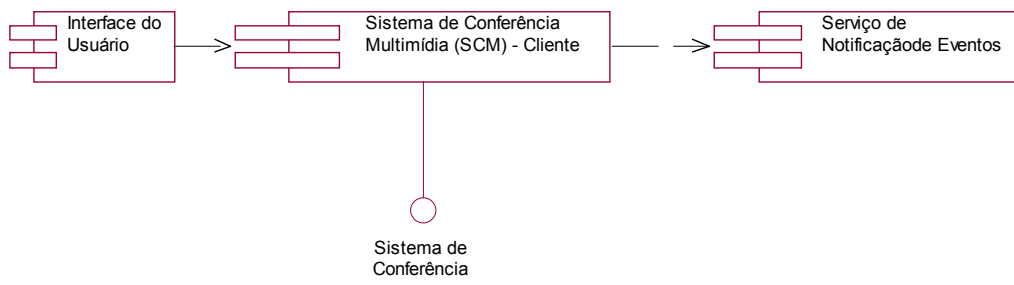
**A4.2) Nível de Sistema**

A aplicação cooperativa deve, então, ser analisada sob a perspectiva da arquitetura escolhida, ou seja sob a perspectiva do cliente e do servidor. Num primeiro momento as tarefas realizadas para a modelagem no nível de sistema, para as três visões, devem ser repetidas neste nível a fim de identificar o limite do sistema para a aplicação cliente e para a aplicação servidora para que, posteriormente, possam ser identificados os componentes a serem desenvolvidos.

**A4.2.1) Visão Estrutural****a) Cliente****a1) Diagrama de Casos de Uso**

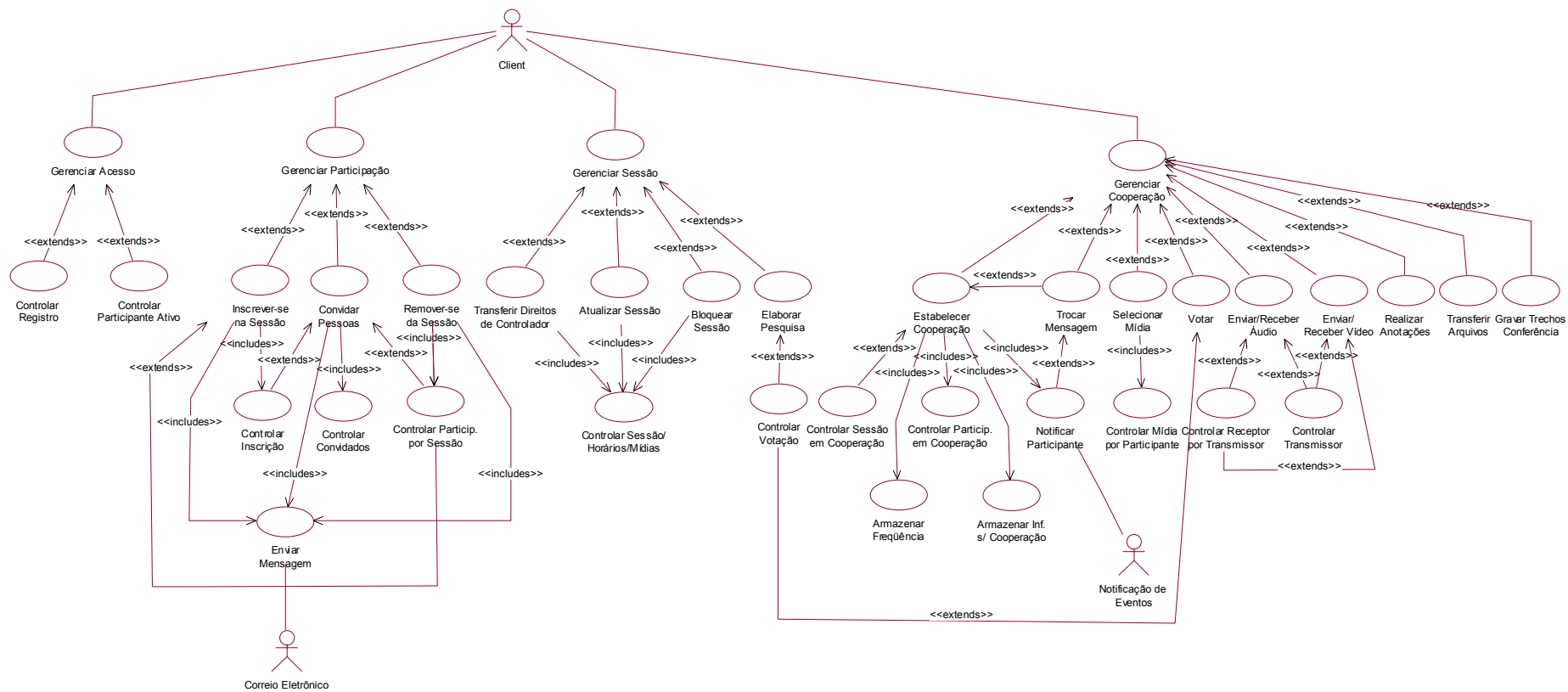


## a2) Diagrama de Pacotes (Componentes)

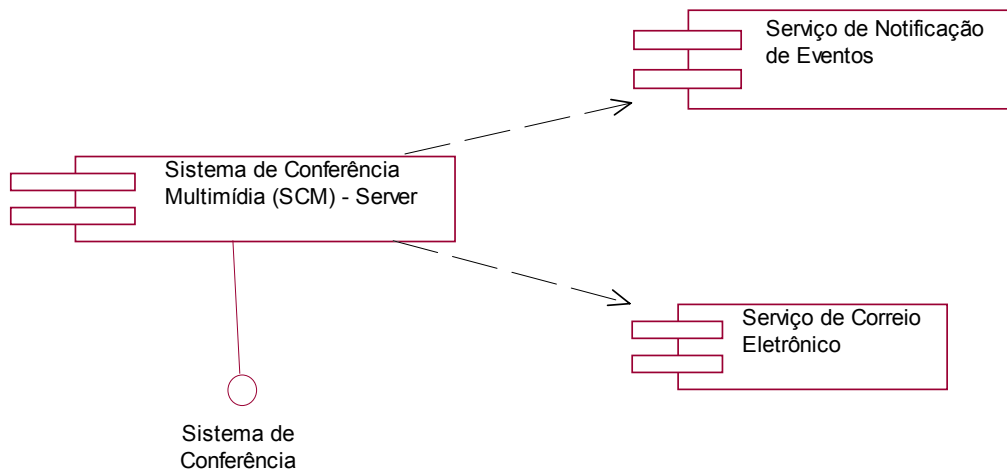


## b) Servidor

## b1) Diagrama de Casos de Uso



b2) Diagrama de Pacotes (Componentes)

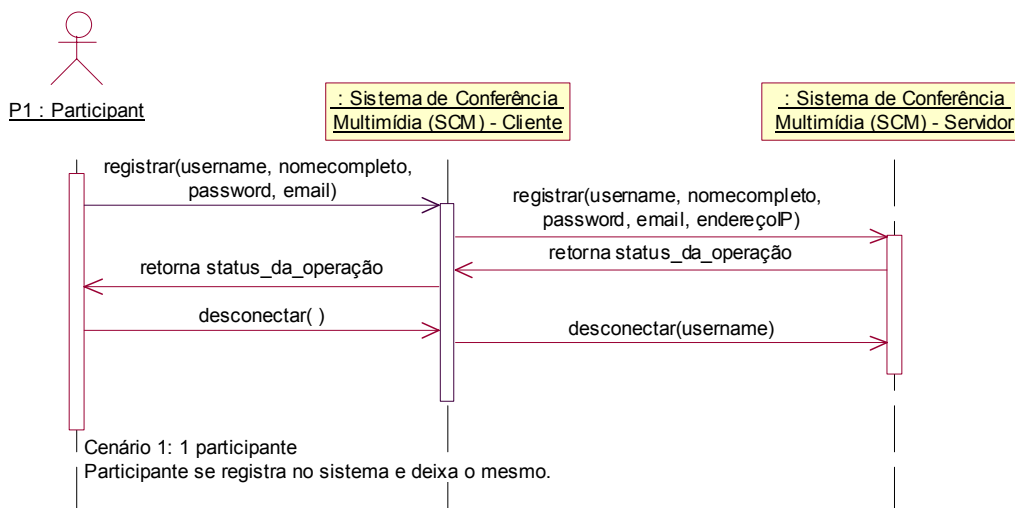


A4.2.2) Visão Comportamental

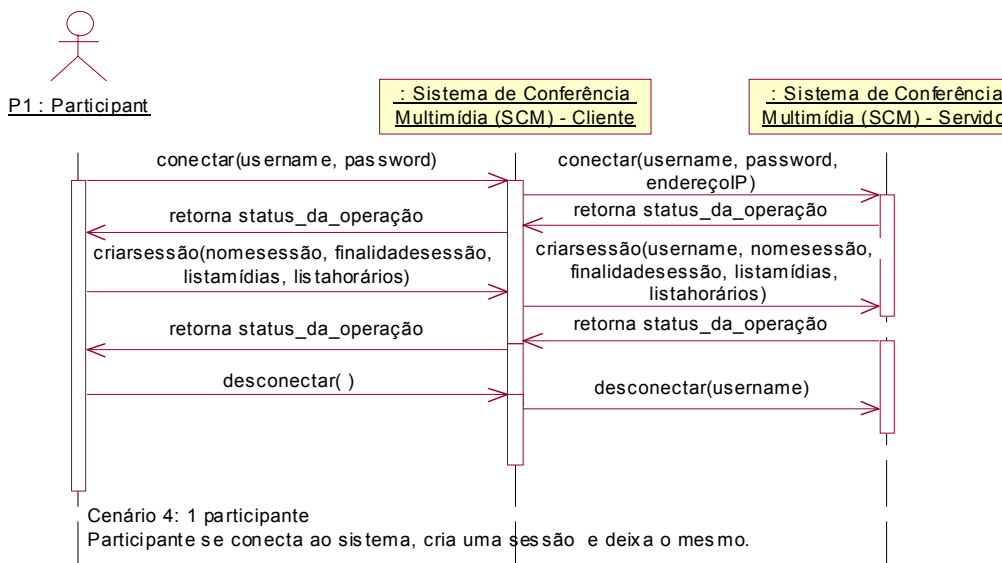
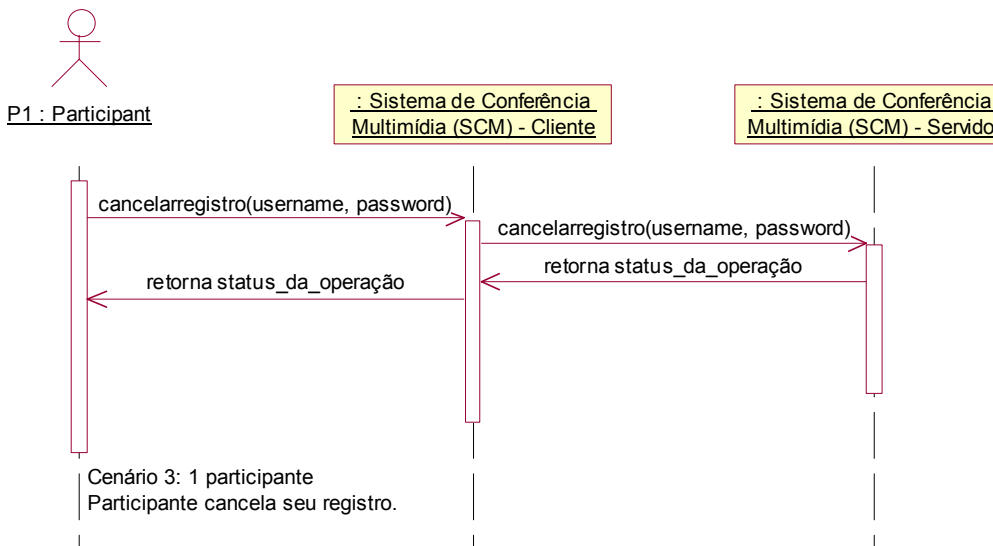
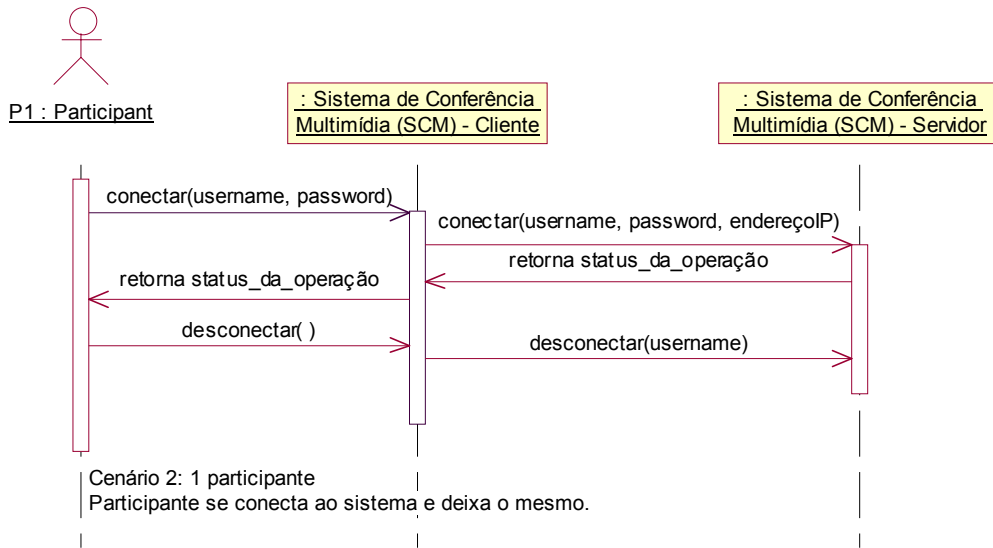
- a) Cliente (as ações internas não serão detalhadas neste ponto)
- b) Servidor (as ações internas não serão detalhadas neste ponto)

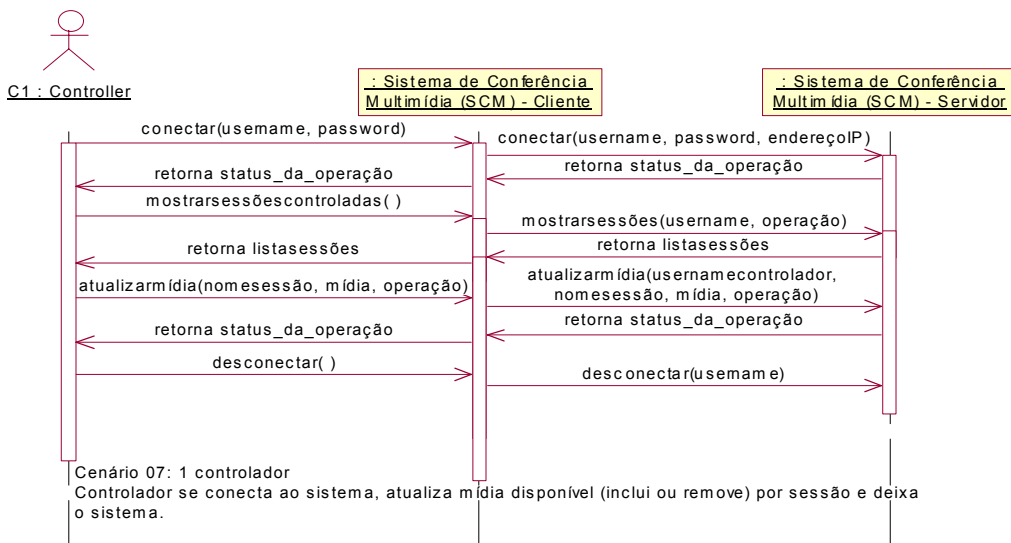
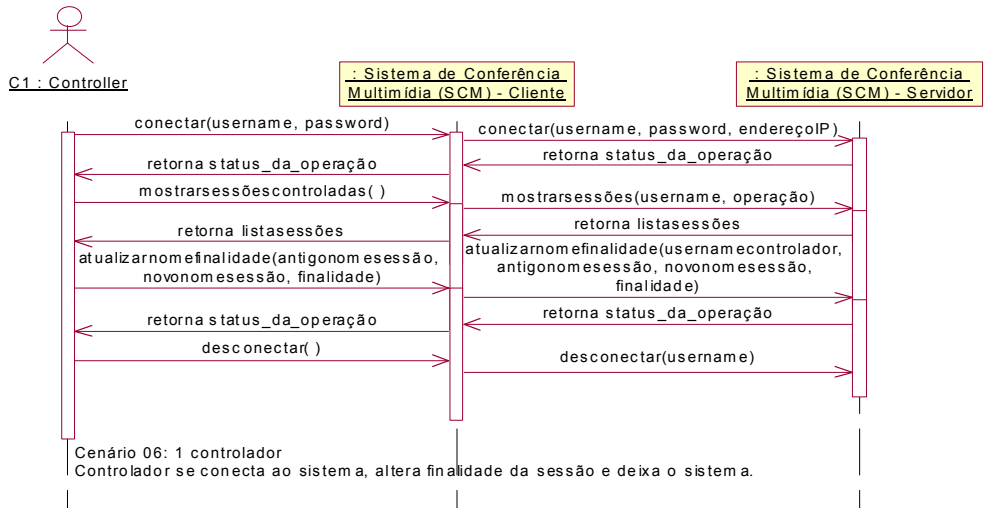
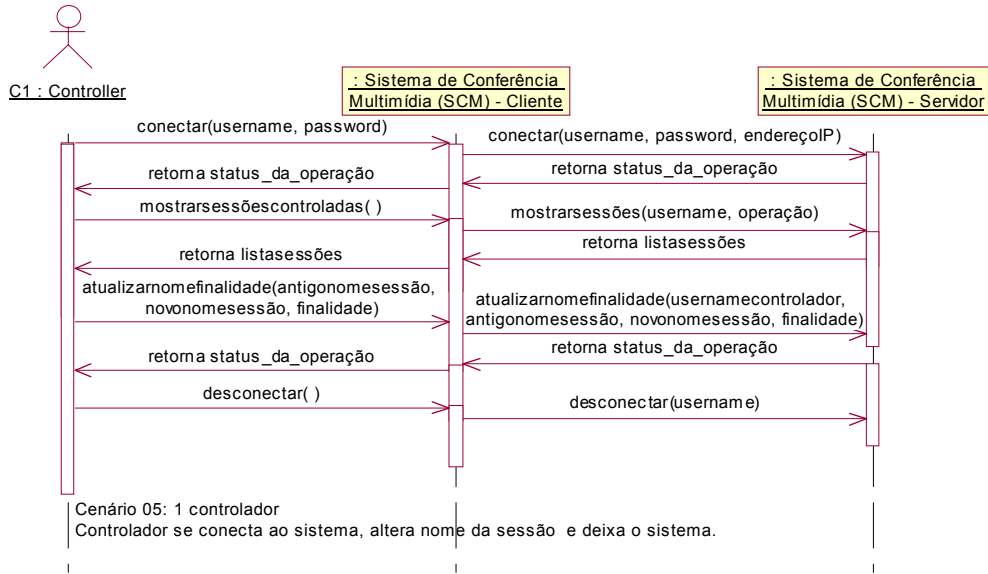
A4.2.3) Visão Interacional

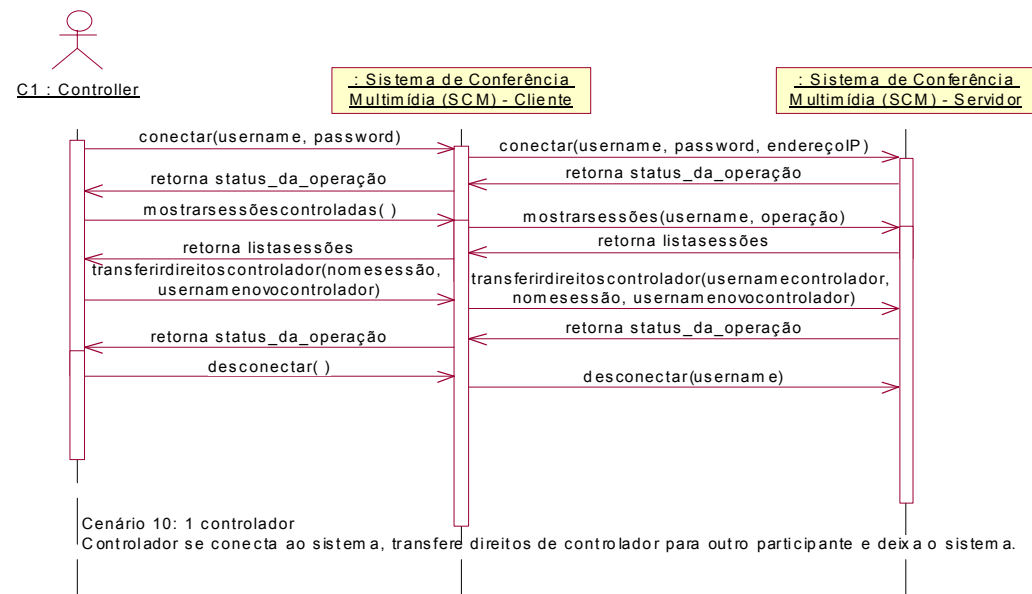
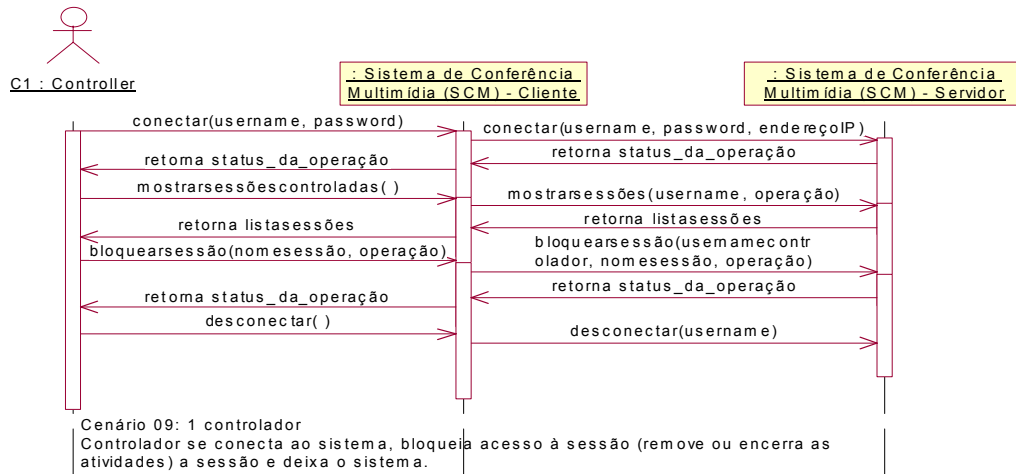
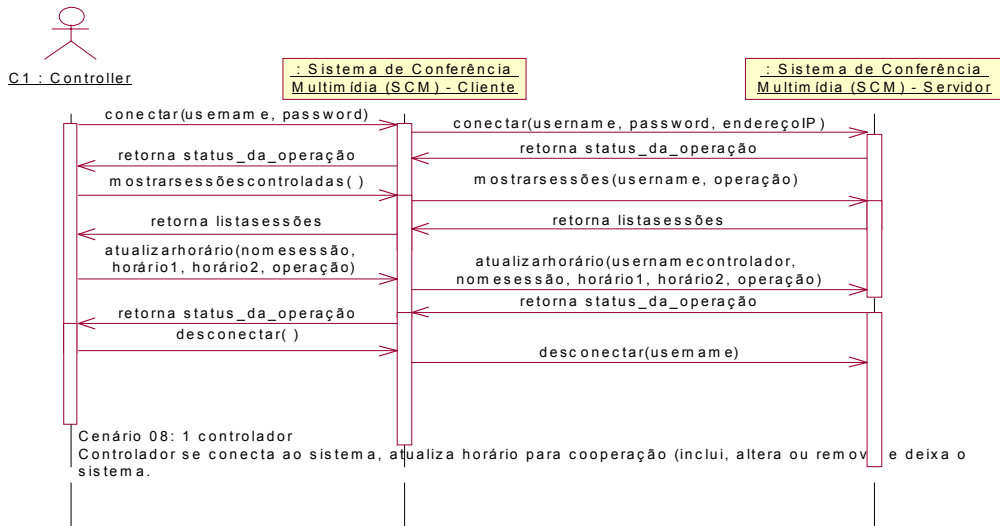
- a) Diagrama de Seqüência

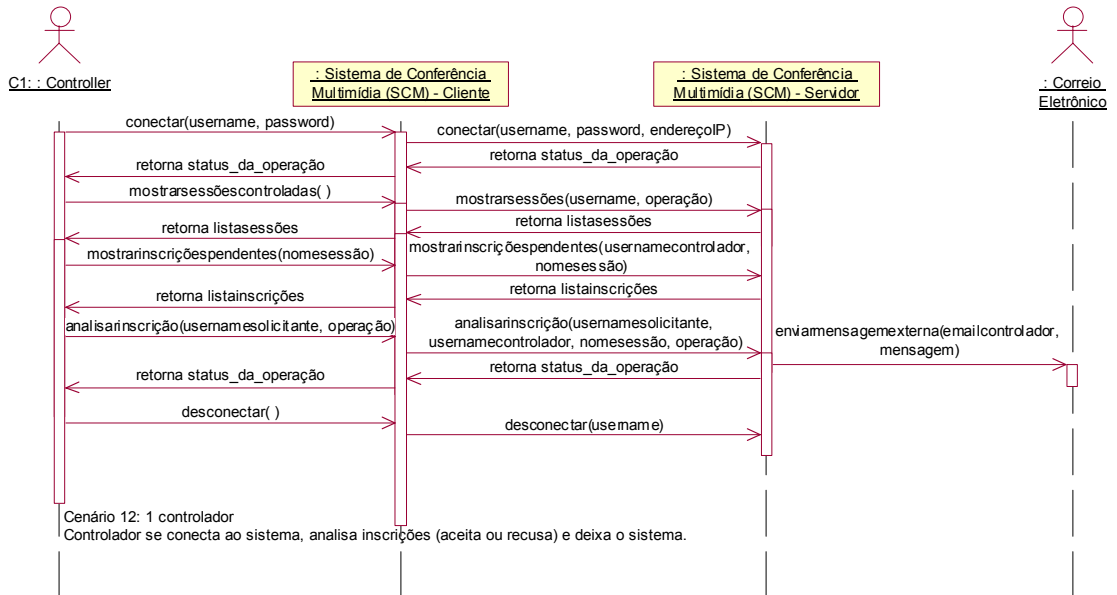
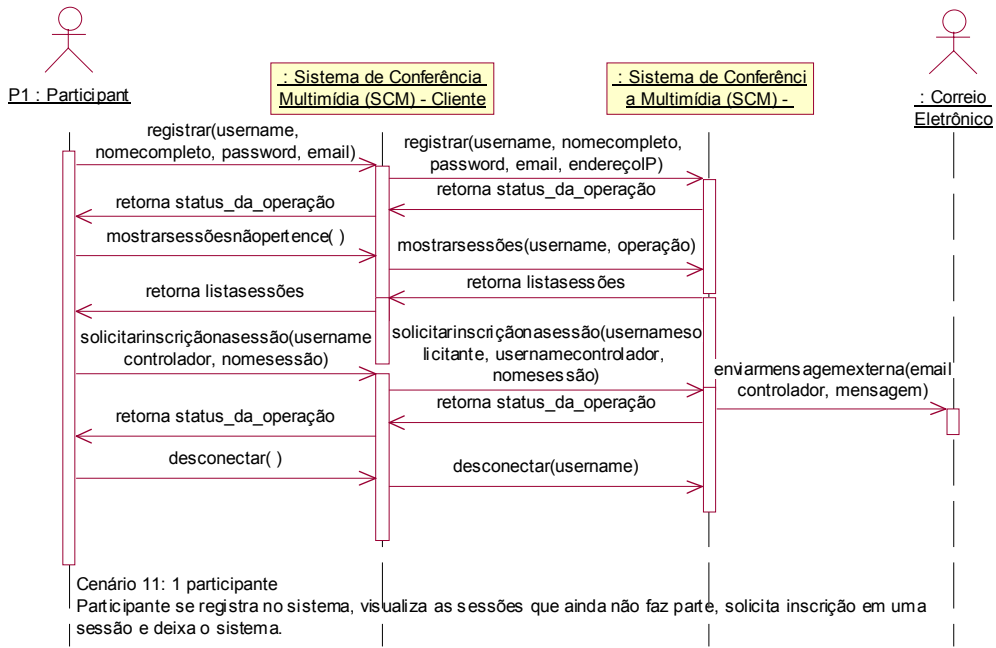


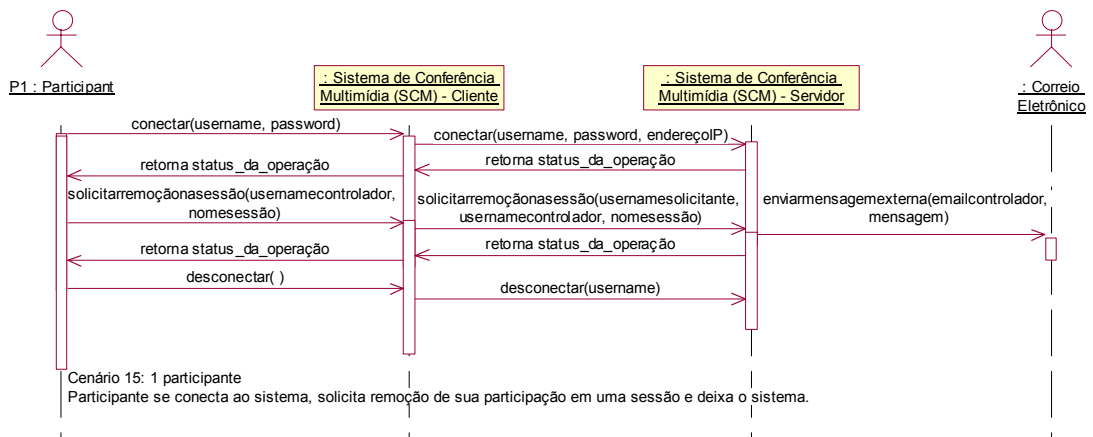
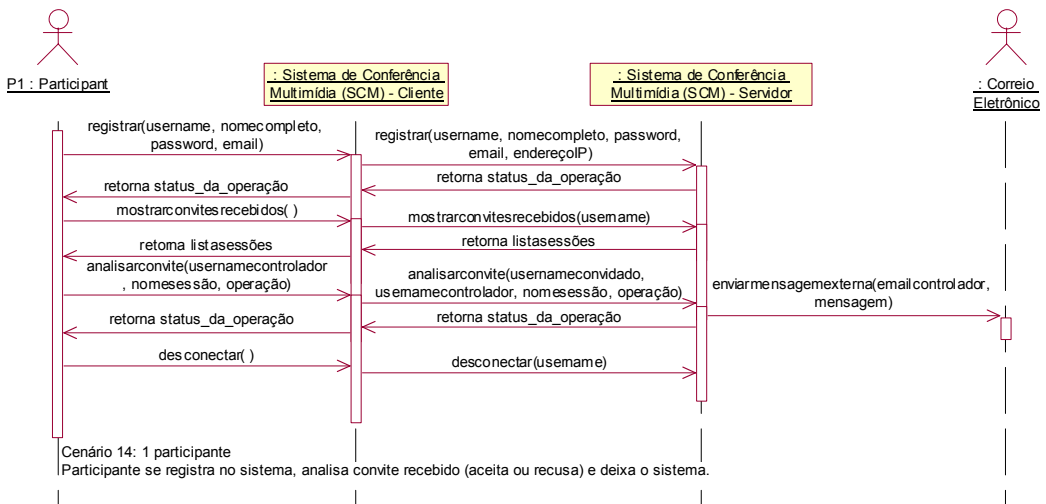
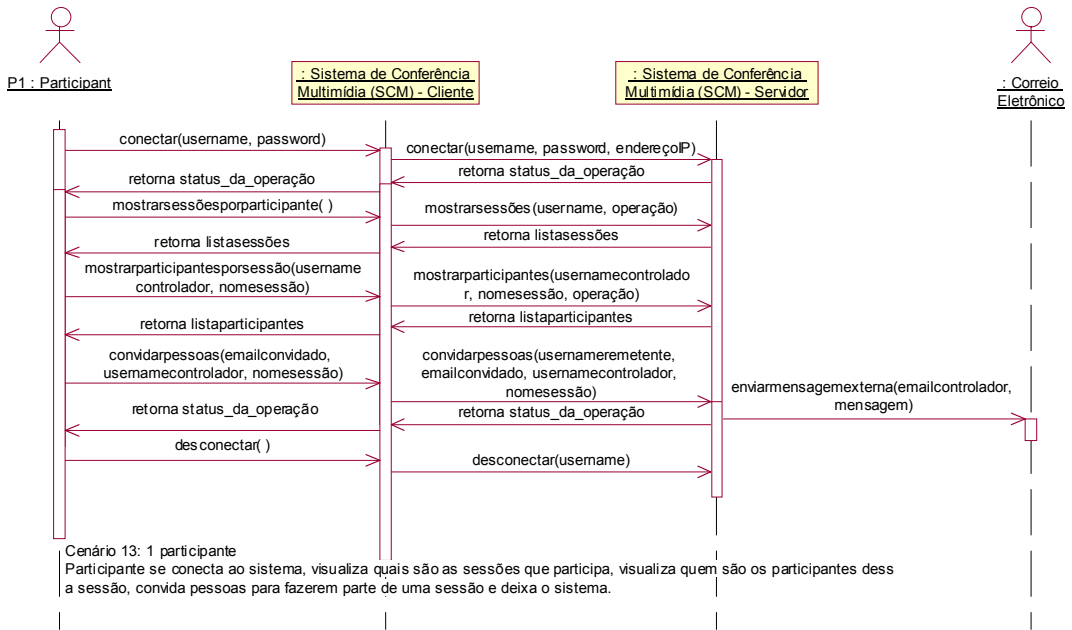




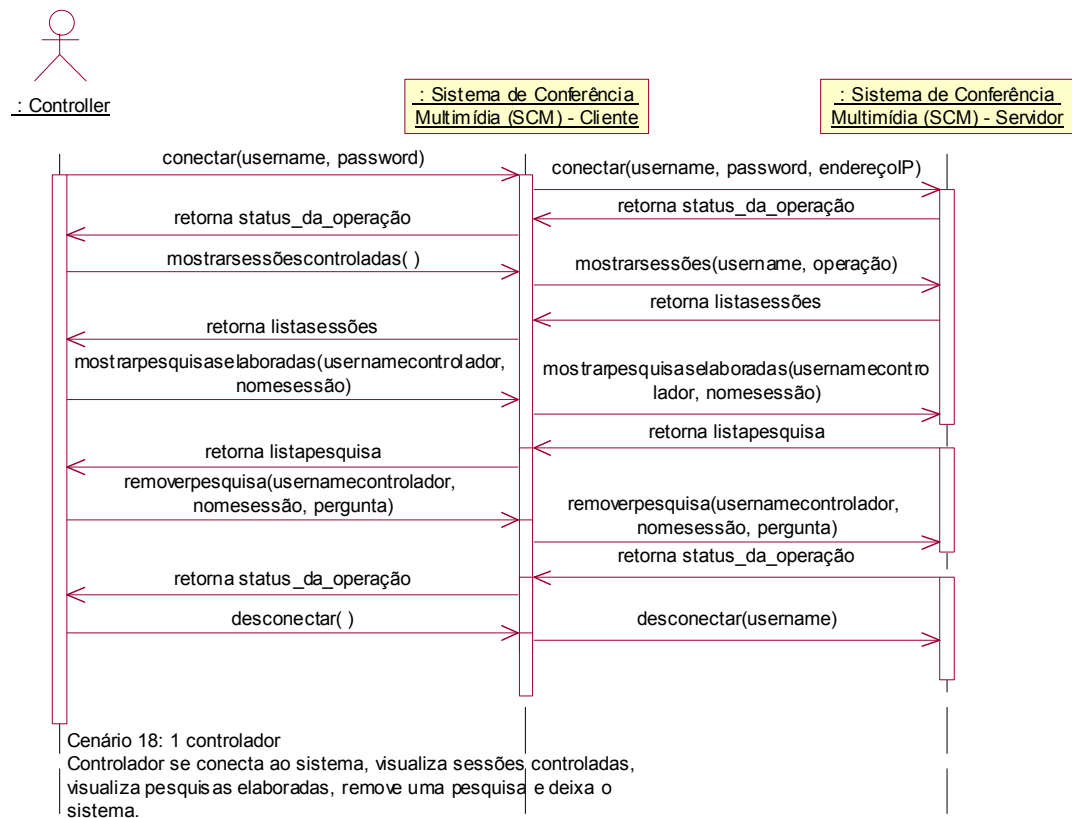
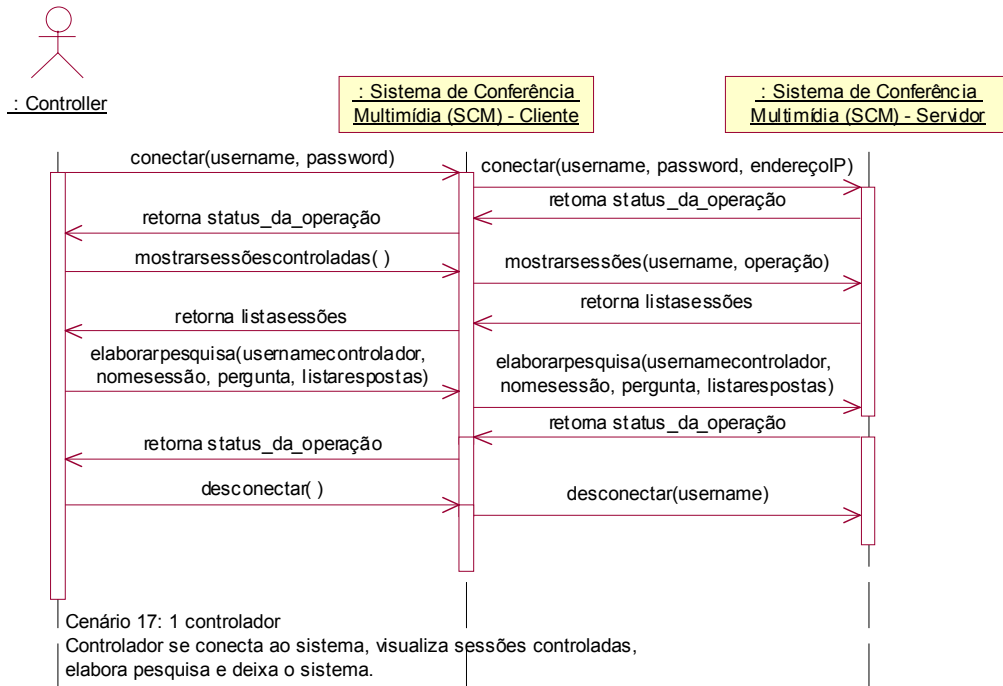












- 
- b) Descrição das Mensagens (por ter sido apresentada uma descrição das mensagens no apêndice 3, as mensagens para a aplicação cliente e para a aplicação servidora não serão descritas)
  
  - c) Relação das Interfaces (as interfaces também não serão relacionadas pelo mesmo motivo citado no item b)



### **A4.3) Identificação dos Componentes a serem Desenvolvidos**

Os requisitos da aplicação cliente, capturados pelo diagrama de casos de uso, são analisados para que sejam identificados os componentes do sistema de conferência multimídia. Observou-se que os serviços oferecidos pela aplicação cliente podem ser agrupados em serviços de gerenciamento de acesso, gerenciamento de sessão, gerenciamento de participação, gerenciamento de cooperação e visualização das informações.

O serviço de gerenciamento de acesso é uma atividade cooperativa fundamental. As atribuições deste serviço correspondem à tarefa de acessar o sistema de conferência multimídia e incluem as atividades de registrar o participante no sistema, conectar o participante ao sistema, desconectar o participante do sistema e remover o registro do participante.

O serviço de gerenciamento de sessão deve manter informações sobre as sessões cooperativas que permitirão uma posterior interação entre seus membros. As atividades agrupadas por este serviço são: criar sessão, atualizar sessão, transferir direitos de controlador para outro participante e elaborar pesquisa.

O serviço de gerenciamento de participação deve prover mecanismos para permitir que um participante una-se a uma sessão cooperativa. Para tanto o mesmo deve saber da existência da sessão. Informações tais como o nome da sessão, quem é o responsável por esta, sua finalidade e quando a atividade cooperativa será realizada devem ser divulgadas. Assim são atribuições deste serviço as tarefas de permitir a inscrição do participante no sistema, incluindo as atividades de solicitar inscrição na sessão e analisar inscrições (aceitar ou recusar); de convidar pessoas para participarem de uma sessão, com as atividades de convidar e analisar convites recebidos (aceitar ou recusar); de enviar mensagem utilizando um serviço externo de correio eletrônico e de remover participação na sessão.

O serviço de gerenciamento de cooperação deve permitir a administração da cooperação entre os membros de uma sessão tais como o estabelecimento de uma cooperação, o controle de entrada e saída de participantes, o controle da frequência dos participantes e suporte para persistência de uma atividade cooperativa. Neste sentido, são atribuições

deste serviço a comunicação, incluindo as atividades de estabelecer ou finalizar uma cooperação; a escolha da mídia a ser utilizada durante a cooperação, permitindo a inclusão ou remoção de mídias; a troca de mensagem síncrona entre os participantes em cooperação, o envio e a recepção de áudio e vídeo, a transferência de arquivos, a gravação de trechos da conferência e o suporte à votação assim que o participante se conectar à sessão.

Finalmente o serviço de visualização das informações deve permitir que sejam apresentadas informações tais como as sessões controladas por um participante, os participantes que fazem parte de uma determinada sessão e as sessões que um dado participante pode estabelecer cooperação.

Esta análise possibilitou a identificação dos componentes da aplicação cliente e da aplicação servidora, os quais serão chamados de componentes cliente e componentes servidor. Buscando alcançar as características desejáveis dos componentes tais como oferecimento de um serviço único, generalidade, interoperabilidade e flexibilidade, dentre outras, tentou-se identificar proximidades entre casos de uso e atribuir a implementação desses casos de uso para um mesmo componente.

A tabela 1 apresenta os vários componentes identificados por este trabalho para o sistema de conferência multimídia. Alguns componentes servidor foram agrupados para formar componentes servidor compostos. Um componente composto é resultado da agregação de vários componentes tal que o comportamento externo do componente combinado corresponda a um único componente. Os componentes servidor agrupados são o gerente de participação e o gerente de cooperação. O primeiro é formado pelos componentes gerente de inscrição, gerente de convite, gerente de correio assíncrono e gerente de associação; enquanto o segundo, pelos componentes gerente de comunicação, gerente de mídia, gerente de correio síncrono, gerente de recepção e gerente de transmissão.

Além dos componentes que representam a funcionalidade da conferência multimídia três outros componentes foram identificados. Dois pertencem à aplicação cliente e têm a finalidade de unir a interface da aplicação aos componentes cliente (Gerente de SCM) e unir a aplicação cliente à aplicação servidora (Gerente de SCM Cliente). O último

pertence à aplicação servidora e deve unir os componentes servidor aos componentes cliente (Gerente de SCM Servidor).

| <b>Classificação</b> | <b>Nome</b>                   | <b>Grau</b> | <b>Funcionalidade</b> |
|----------------------|-------------------------------|-------------|-----------------------|
| Cliente              | Gerente de SCM                | Simples     |                       |
|                      | Gerente de SCM Cliente        | Simples     |                       |
|                      | Gerente de Acesso             | Simples     | Acessar               |
|                      | Gerente de Sessão             | Simples     | Controlar Sessão      |
|                      | Gerente de Participação       | Simples     | Participar            |
|                      | Gerente de Cooperação         | Simples     | Cooperar              |
|                      | Gerente de Visualização       | Simples     | Visualizar            |
| Servidor             | Gerente de SCM Servidor       | Simples     |                       |
|                      | Gerente de Acesso             | Simples     | Acessar               |
|                      | Gerente de Sessão             | Simples     | Controlar Sessão      |
|                      | Gerente de Participação       | Composto    | Participar            |
|                      | Gerente de Cooperação         | Composto    | Cooperar              |
|                      | Gerente de Inscrição          | Simples     | Participar            |
|                      | Gerente de Convite            | Simples     | Participar            |
|                      | Gerente de Correio Assíncrono | Simples     | Participar            |
|                      | Gerente de Associação         | Simples     | Participar            |
|                      | Gerente de Comunicação        | Simples     | Cooperar              |
|                      | Gerente de Mídia              | Simples     | Cooperar              |
|                      | Gerente de Correio Síncrono   | Simples     | Cooperar              |
|                      | Gerente de Recepção           | Simples     | Cooperar              |
|                      | Gerente de Transmissão        | Simples     | Cooperar              |
|                      | Gerente de Visualização       | Simples     | Visualizar            |

Tabela 1. Componentes Identificados para o Sistema de Conferência Multimídia (SCM)

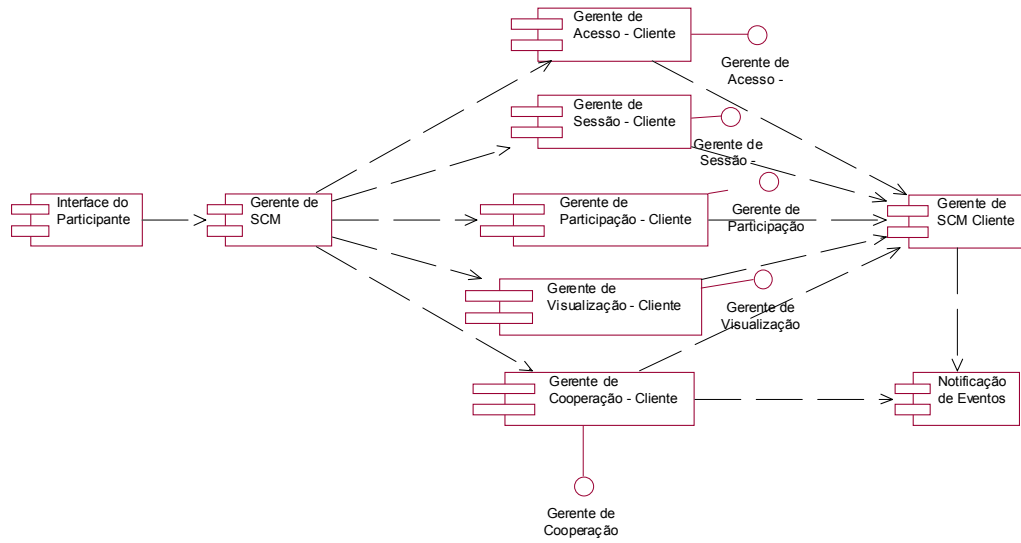
Embora os componentes identificados tenham se baseado em uma aplicação cooperativa específica – sistema de conferência multimídia –, a generalidade pode ser observada. Os componentes sob a funcionalidade acessar o sistema, por exemplo, podem ser reutilizados por qualquer aplicação cooperativa.

#### **A4.4) Nível de Componente**

##### **A4.4.1) Visão Estrutural**

###### **a) Cliente**

###### **a1) Diagrama de Pacotes (Componentes)**



a2) Classes de Interface

| <<Interface>><br>SCM   |
|--|
| registrar(username, nomecompleto, password, email) : integer<br>conectar(username, password) : integer<br>desconectar() : void<br>cancelarregistro(username, password) : integer<br>criarsessão(nomesessão, finalidadesessão, listamídias, listahorários) : integer<br>mostrarsessõescontroladas() : Array of String<br>atualizarnomefinalidade(antigonomesessão, novonomesessão, finalidadesessão) : integer<br>atualizarhorário(nomesessão, horário1, horário2, operação) : integer<br>atualizarmídia(nomesessão, mídia, operação) : integer<br>bloquearsessão(nomesessão, operação) : integer<br>transferirdireitoscontrolador(nomesessão, novousername) : integer<br>elaborarpesquisa(usernamecontrolador, nomesessão, pergunta, listarespostas) : integer<br>mostrapesquisaselaboradas(usernamecontrolador, nomesessão) : array of string<br>removerpesquisa(usernamecontrolador, nomesessão, pergunta) : integer<br>mostrarsessõesnãoopertence() : Array of String<br>solicitarinscriçãonasessão(usernamecontrolador, nomesessão) : integer<br>mostrarinscriçõespendentes(nomesessão) : Array of String<br>analisarinscrição(usernamecontrolador, operação) : integer<br>mostrarsessõesporparticipante() : Array of String<br>mostrarparticipantespossessão(usernamecontrolador, nomesessão) : Array of String<br>convidarpeessoas(emailconvidado, usernamecontrolador, nomesessão) : integer<br>mostrarconvitesrecebidos() : IArray of String<br>analisarconvite(usernamecontrolador, nomesessão, operação) : integer<br>solicitarremoçãoonasessão(usernamecontrolador, nomesessão) : integer<br>mostrarsessõesemcooperação() : Array of String<br>iniciarcooperação(usernamecontrolador, nomesessão) : integer<br>abandonarcooperação(mensagem) : void<br>terminarcooperação(mensagem) : void<br>deixarcooperação() : void<br>mostrarparticipantesemcooperação() : Array of String<br>enviarmensagem(usernamedestino, mensagem) : void<br>recebermensagem() : string<br>atualizarmídiacooperação(mídia, operação) : integer<br>iniciargravacaoconferencia(usernamecontrolador, nomesessão, usernametransmissor, nomearquivo) : integer<br>encerrargravacaoconferencia(usernamecontrolador, nomesessão, usernametransmissor) : integer<br>votar(resposta) : integer<br>enviarpesquisa(usernamedestinatário, pergunta, listarespostas) : string<br>transferirarquivo(usernamedestinatário, nomearquivo) : integer<br>ativaranotação() : integer<br>iniciartransmissão(streaming, operação) : void<br>interrompertransmissão(operação) : void<br>mostrarparticipantesemtransmissão() : Array of String<br>solicitaroperaçãorecepção(usernamecontrolador, operação) : void<br>notificarpresença(username, ação) : void<br>notificarmensagem(usernamedestinatário) : void |

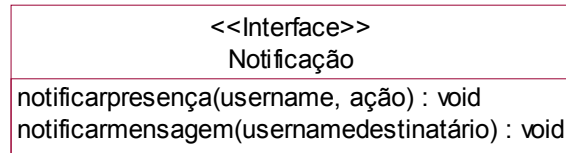
| <<Interface>><br>Gerente de Acesso - Cliente   |
|--|
| registrar(username, nomecompleto, password, email) : integer<br>conectar(username, password) : integer<br>desconectar() : void<br>cancelarregistro(username, password) : integer |

| <<Interface>><br>Gerente de Sessão - Cliente   |
|--|
| criarsessão(nomesessão, finalidadesessão, listamídias, listahorários) : integer<br>atualizarnomefinalidade(antigonomesessão, novonomesessão, finalidade) : integer<br>atualizamídia(nomesessão, mídia, operação) : integer<br>atualizarhorário(nomesessão, horário1, horário2, operação) : integer<br>bloquearsessão(nomesessão, operação) : integer<br>transferirdireitoscontrolador(nomesessão, usemamenovocontrolador) : integer<br>elaborarpesquisa(usemamecontrolador, nomesessão, pergunta, listarespostas) : integer<br>mostrarpesquisaselaboradas(usernamecontrolador, nomesessão) : array of string<br>removerpesquisa(usernamecontrolador, nomesessão, pergunta) : integer |

| <<Interface>><br>Gerente de Participação - Cliente   |
|--|
| solicitarinscriçãonasessão(usernamecontrolador, nomesessão) : integer<br>mostrarinscriçõespendentes(nomesessão) : Array of String<br>analisarinscrição(usernamesolicitante, operação) : integer<br>convidarpessoas(emailconvidado, usernamecontrolador, nomesessão) : integer<br>mostrarconvitesrecebidos() : Array of String<br>analisarconvite(usernamecontrolador, nomesessão, operação) : integer<br>solicitarremoçãonasessão(usernamecontrolador, nomesessão) : integer |

| <<Interface>><br>Gerente de Visualização - Cliente   |
|--|
| mostrarsessõescontroladas() : Array of String<br>mostrarsessõesporparticipante() : Array of String<br>mostrarsessõesnãoopertence() : Array of String<br>mostrarsessõesemcooperação() : Array of String<br>mostrarparticipantesemcooperação(usernamecontrolador, nomesessão) : Array of String<br>mostrarparticipantesporemissão(usernamecontrolador, nomesessão) : Array of String<br>mostrarparticipantesemtransmissão(usernamecontrolador, nomesessão) : Array of String |

| <<Interface>><br>Gerente de Cooperação - Cliente  |
|---|
| iniciarcooperação(usernamecontrolador, nomesessão) : integer<br>deixarcooperação() : void<br>abandonarcooperação(mensagem) : void<br>terminarcooperação(mensagem) : void<br>enviarmensagem(usernamedestino, mensagem) : void<br>recebermensagem() : String<br>atualizamídiacooperação(mídia, operação) : integer<br>iniciargravacaoconferencia(usernamecontrolador, nomesessão, usernametransmissor, nomearquivo) : integer<br>encerrargravaçãoconferência(usernamecontrolador, nomesessão, usernametransmissor) : integer<br>votar(resposta) : integer<br>enviarpesquisa(usernamedestinatário, pergunta, listarespostas) : string<br>transferirarquivo(usernamedestinatário, nomearquivo) : integer<br>ativaranotação() : integer<br>iniciartransmissão(áudiovídeo, operação) : void<br>suspendertransmissão(operação) : void<br>solicitaroperaçãorecepção(usernametransmissor, operação) : void<br>notificarpresença(username, ação) : void<br>notificarmensagem(usernamedestinatário) : void |



Participante

|  |
|--|
| notificarpresença(username, ação) : void<br>notificarmensagem(usernamedestinatário) : void |
|--|



Servidor

---

```

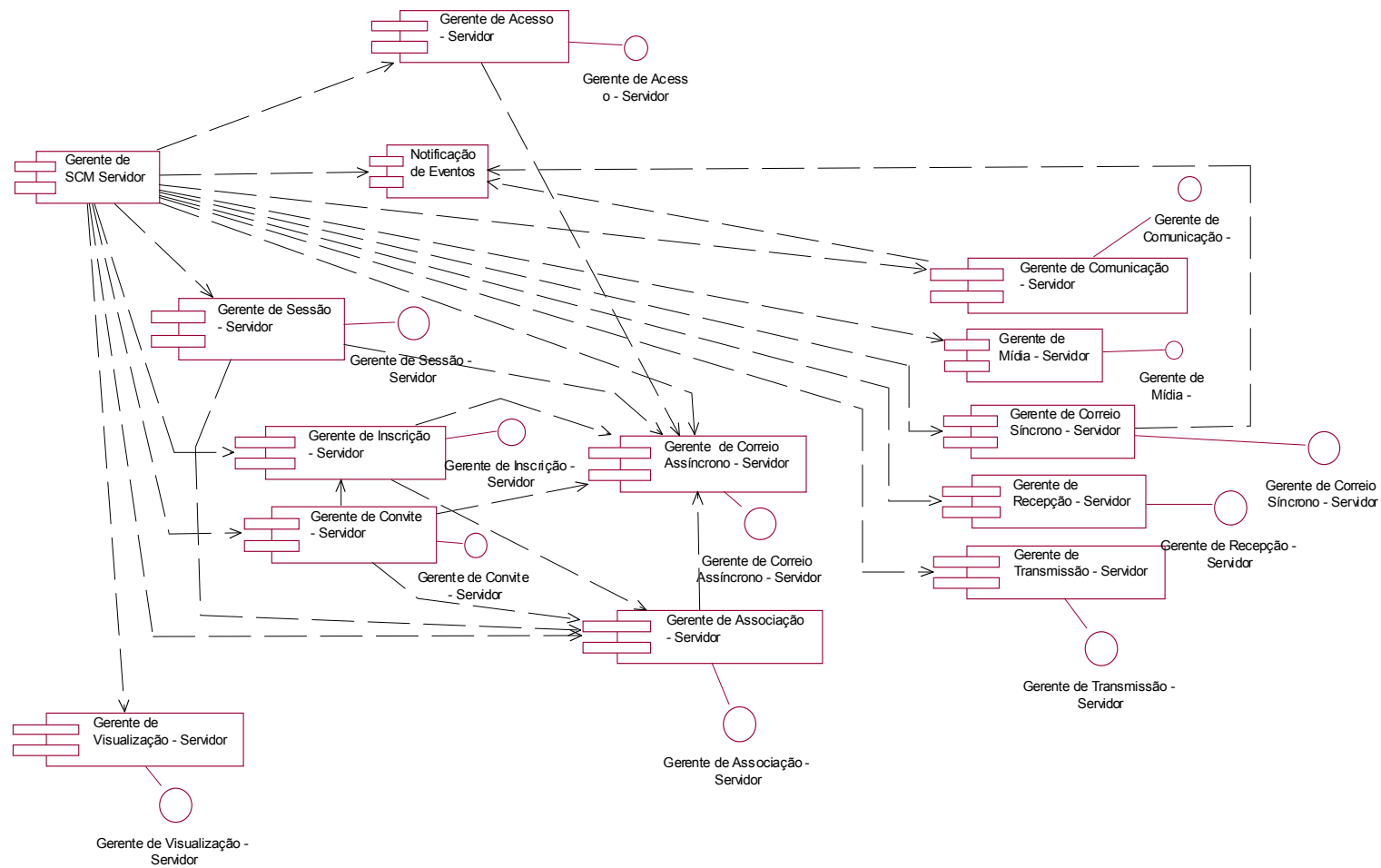
registrar(username, nomecompleto, password, email, endereçoIP) : integer
conectar(username, password, endereçoIP) : integer
cancelarregistro(username, password) : integer
desconectar(username) : integer
criarsessão(username, nomesessão, finalidadeessão, listamídias, listahorários) : integer
atualizarnomefinalidade(usernamecontrolador, antigonomesessão, novonomesessão, finalidade) : integer
atualizarhorário(usernamecontrolador, nomesessão, horário1)
atualizarmídia(usernamecontrolador, nomesessão, mídia, operação) : integer
bloquearsessão(usernamecontrolador, nomesessão, operação) : integer
transferirdireitoscontrolador(usernamecontrolador, nomesessão, usenamenocontrolador) : integer
elaborarpesquisa(usernamecontrolador, nomesessão, pergunta, listarespostas) : integer
mostrarpesquisaselaboradas(usernamecontrolador, nomesessão) : array of string
removerpesquisa(usernamecontrolador, nomesessão, pergunta) : integer
solicitarinscriçãoonasessão(usenamesolicitante, usernamecontrolador, nomesessão) : integer
mostrarinscriçõespendentes(usernamecontrolador, nomesessão) : Array of String
analisarinscrição(usenamesolicitante, usernamecontrolador, nomesessão, operação) : integer
mostrarsessões(username, operação) : Array of String
mostrarparticipantes(usernamecontrolador, nomesessão, operação) : Array of String
convidarpessoas(usenameremetente, emailconvidado, usernamecontrolador, nomesessão) : integer
mostrarconvitesrecebidos(username) : Array of String
analisarconvite(usernameconvidado, usernamecontrolador, nomesessão, operação) : integer
solicitarremoçãoonasessão(usenamesolicitante, usernamecontrolador, nomesessão) : integer
iniciarcooperação(usenamesolicitante, usernamecontrolador, nomesessão) : integer
abandonarcooperação() : void
terminarcooperação(usernamecontrolador, nomesessão, mensagem) : void
deixarcooperação(usenamesolicitante, usernamecontrolador, nomesessão) : void
enviarmensagem(usenameremetente, usernamecontrolador, nomesessão, usenamedestino, mensagem) : void
recebermensagem(usernamecontrolador, nomesessão, usenamesolicitante) : string
atualizarmídiacooperação(usenamesolicitante, usernamecontrolador, nomesessão, mídia, operação) : integer
iniciargravacaoconferência(usernamecontrolador, nomesessão, usenametransmissor, nomearquivo) : integer
encerrargravacaoconferência(usernamecontrolador, nomesessão, usenametransmissor) : integer
enviarpesquisa(usenamedestinatário, pergunta, listarespostas) : string
votar(usernamecontrolador, nomesessão, pergunta, resposta) : integer
transferirarquivo(usenamedestinatário, nomearquivo) : integer
ativaranotação() : integer
iniciartransmissão(usenametransmissor, usernamecontrolador, nomesessão, streaming, operação) : void
suspendertransmissão(usenametransmissor, operação) : void
solicitaroperaçãorecepção(usenamesolicitante, usernamecontrolador, nomesessão, usenametransmissor, operação) : void

```

## b) Servidor

### b1) Diagrama de Pacotes (Componentes)





## b2) Classes de Interface

| <<Interface>><br>SCM - Servidor   |
|---|
| registrar(username, nomecompleto, password, email, endereçoIP) : integer<br>conectar(username, password, endereçoIP) : integer<br>cancelarregistro(username, password) : integer<br>desconectar(username) : integer<br>criarsessão(username, nomesessão, finalidadesessão, listamídias, listahorários) : integer<br>mostrarsessões(username, operação) : Array of String<br>atualizarnomefinalidade(usernamecontrolador, antigonomesessão, novonomesessão, finalidade) : integer<br>atualizarhorário(usernamecontrolador, nomesessão, horário1, horário2, operação) : integer<br>atualizarmídia(usernamecontrolador, nomesessão, mídia, operação) : integer<br>bloquearsessão(usernamecontrolador, nomesessão, operação) : integer<br>transferirdireitoscontrolador(usernamecontrolador, nomesessão, usernameovocontrolador) : integer<br>elaborarpesquisa(usernamecontrolador, nomesessão, pergunta, listarespostas) : integer<br>mostrarpesquisaselaboradas(usernamecontrolador, nomesessão) : array of string<br>removerpesquisa(usernamecontrolador, nomesessão, pergunta) : integer<br>solicitarinscrição(nomesessão, usernamecontrolador, nomesessão) : integer<br>mostrarinscriçõespendentes(usernamecontrolador, nomesessão) : Array of String<br>analisarinscrição(usernamecontrolador, usernamecontrolador, nomesessão, operação) : integer<br>convidarpessoas(usernameremetente, emailconvidado, usernamecontrolador, nomesessão) : integer<br>mostrarconvitesrecebidos(username) : Array of String<br>analisarconvite(usernameconvidado, usernamecontrolador, nomesessão, operação) : integer<br>solicitaradição(nomesessão, usernamecontrolador, nomesessão) : integer<br>solicitarremoção(nomesessão, usernamecontrolador, nomesessão) : integer<br>verificarsessão(usernamecontrolador, nomesessão) : boolean<br>verificarsolicitante(usernamecontrolador, usernamecontrolador, nomesessão, operação) : boolean<br>iniciarcooperação(usernamecontrolador, usernamecontrolador, nomesessão) : integer<br>abandonarcooperação() : void<br>terminarcooperação(usernamecontrolador, nomesessão, mensagem) : void<br>deixarcooperação(usernamecontrolador, usernamecontrolador, nomesessão) : void<br>mostrarparticipantes(username, nomesessão, operação) : Array of String<br>enviarmensagem(usernameremetente, usernamecontrolador, nomesessão, usernamedestino, mensagem) : void<br>recebermensagem(usernamecontrolador, nomesessão, usernamecontrolador) : string<br>atualizarmídiacooperação(usernamecontrolador, usernamecontrolador, nomesessão, mídia, operação) : integer<br>iniciargravaçãoconferência(usernamecontrolador, nomesessão, usernametransmissor, nomearquivo) : integer<br>encerrargravaçãoconferência(usernamecontrolador, nomesessão, usernametransmissor) : integer<br>enviarpesquisa(usernamedestinatário, pergunta, listarespostas) : string<br>votar(usernamecontrolador, nomesessão, pergunta, resposta) : integer<br>transferirarquivo(usernamedestinatário, nomearquivo) : integer<br>ativaranotação() : integer<br>iniciartransmissão(usernametransmissor, usernamecontrolador, nomesessão, streaming, operação) : void<br>interrompertransmissão(usernametransmissor, operação) : void<br>solicitaroperação(recepção, usernamecontrolador, nomesessão, usernametransmissor, operação) : void<br>verificaroperação(operação) : boolean<br>verificarparticipante(usernamecontrolador, nomesessão, usernameparticipante) : boolean |

| <<Interface>><br>Gerente de Acesso - Servidor   |
|---|
| registrar(username, nomecompleto, password, email, endereçoIP) : integer<br>conectar(username, password, endereçoIP) : integer<br>cancelarregistro(username, password) : integer<br>desconectar(username) : integer |

| <<Interface>><br>Gerente de Sessão - Servidor   |
|---|
| criarsessão(username, nomesessão, finalidadesessão, listamídias, listahorários) : integer<br>atualizarnomefinalidade(usernamecontrolador, antigonomesessão, novonomesessão, finalidade) : integer<br>atualizarhorário(usernamecontrolador, nomesessão, horário1, horário2, operação) : integer<br>atualizarmídia(usernamecontrolador, nomesessão, mídia, operação) : integer<br>bloquearsessão(usernamecontrolador, nomesessão, operação) : integer<br>transferirdireitoscontrolador(usernamecontrolador, nomesessão, usernameovocontrolador) : integer<br>elaborarpesquisa(usernamecontrolador, nomesessão, pergunta, listarespostas) : integer<br>mostrarpesquisaselaboradas(usernamecontrolador, nomesessão) : array of string<br>removerpesquisa(usernamecontrolador, nomesessão, pergunta) : integer |

|   |
|---|
| <<Interface>><br>Gerente de Inscrição - Servidor  |
| solicitarinscrição(nomesessão, usernamecontrolador, nomesolicitante) : integer<br>mostrarinscriçõespendentes(usernamecontrolador, nomesessão) : Array of String<br>analisarinscrição(usernamecontrolador, nomesolicitante, nomesessão, operação) : integer<br>obterinscriçõespendentes(usernamecontrolador, nomesessão) : array of string |

|   |
|---|
| <<Interface>><br>Gerente de Convite - Servidor  |
| convidarpessoas(usernameremetente, emailconvitado, usernamecontrolador, nomesessão) : integer<br>mostrarconvitesrecebidos(username) : Array of String<br>analisarconvite(usernameconvitado, usernamecontrolador, nomesessão, operação) : integer<br>obterconvites(username) : Array of String |

|   |
|---|
| <<Interface>><br>Gerente de Associação - Servidor   |
| solicitaradição(nomesessão, usernamecontrolador, nomesolicitante) : integer<br>solicitarremoção(nomesessão, usernamecontrolador, nomesolicitante) : integer<br>verificarsessão(usernamecontrolador, nomesessão) : boolean<br>verificarsolicitante(usernamecontrolador, nomesolicitante, nomesessão, operação) : boolean |

|   |
|---|
| <<Interface>><br>Gerente de Correio Assíncrono - Servidor   |
| enviarmensagemexterna(usernameouemail, situação, obteremail, usernamecontrolador, nomesessão, usernameremetente) : void |

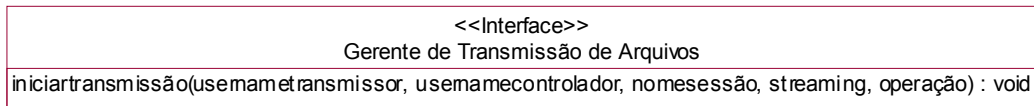
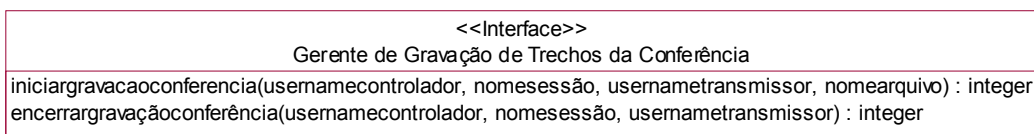
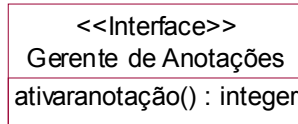
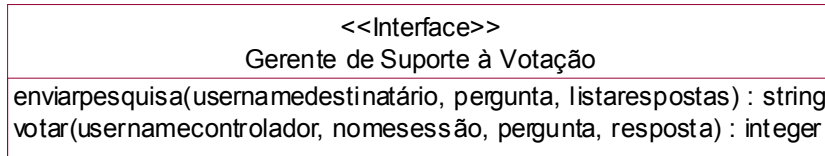
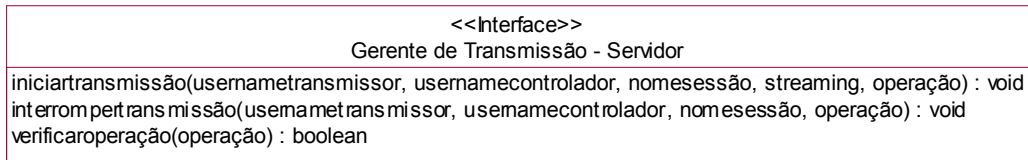
|  |
|--|
| <<Interface>><br>Gerente de Visualização - Servidor  |
| mostrarsessões(username, operação) : Array of String<br>mostrarparticipantes(username, nomesessão, operação) : Array of String |

|  |
|--|
| <<Interface>><br>Gerente de Comunicação - Servidor   |
| iniciarcooperação(usernamecontrolador, nomesolicitante) : integer<br>abandonarcooperação() : void<br>terminarcooperação(usernamecontrolador, nomesessão, mensagem) : void<br>deixarcooperação(usernamecontrolador, nomesolicitante, nomesessão) : void |

|   |
|---|
| <<Interface>><br>Gerente de Mídia - Servidor  |
| atualizarmídiacooperação(usernamecontrolador, nomesolicitante, nomesessão, mídia, operação) : integer<br>verificarmídia(usernamecontrolador, nomesessão, usernameparticipante, mídia) : boolean |

|  |
|--|
| <<Interface>><br>Gerente de Correio Síncrono - Servidor  |
| enviarmensagem(usernameremetente, usernamecontrolador, nomesessão, usernamedestino, mensagem) : void<br>recebermensagem(usernamecontrolador, nomesessão, usernamecontrolador) : string |

|  |
|--|
| <<Interface>><br>Gerente de Recepção - Servidor  |
| solicitaroperaçãorecepção(usernamecontrolador, nomesolicitante, nomesessão, usernametransmissor, operação) : void<br>verificaroperação(operação) : boolean |



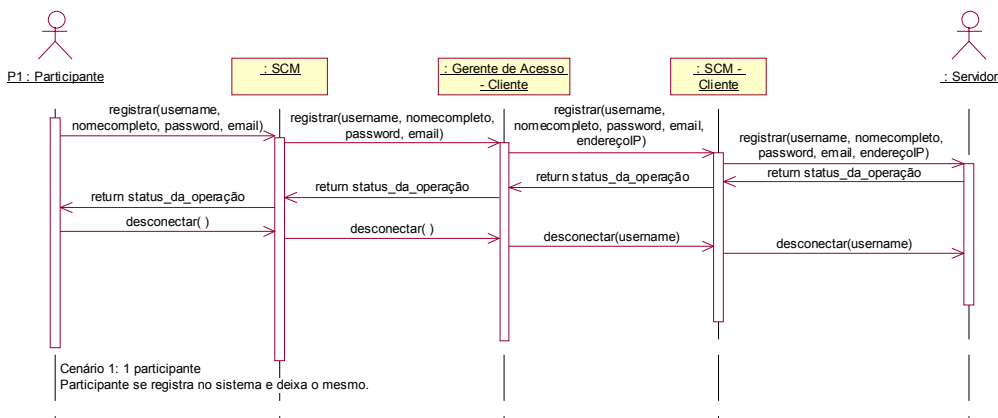
Correio Eletrônico

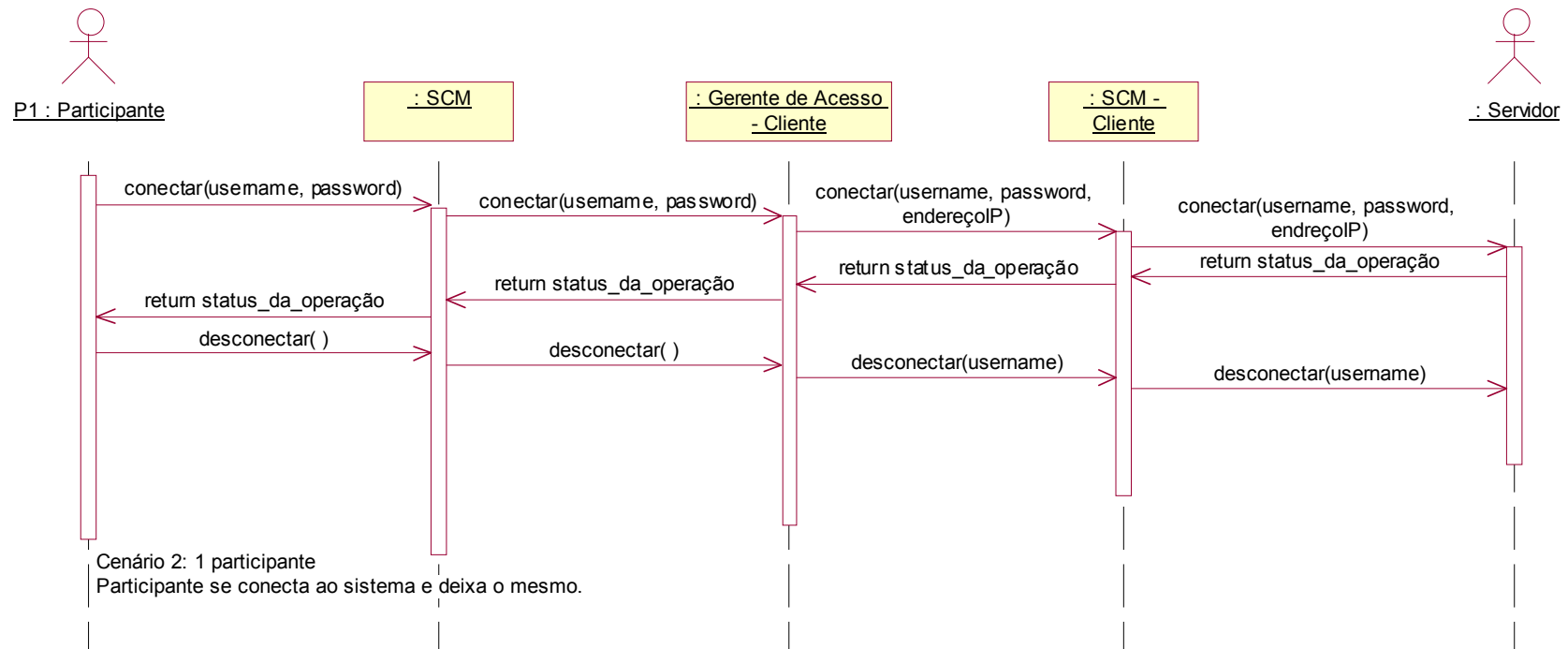
enviarmensagemexterna(email, mensagem) : void

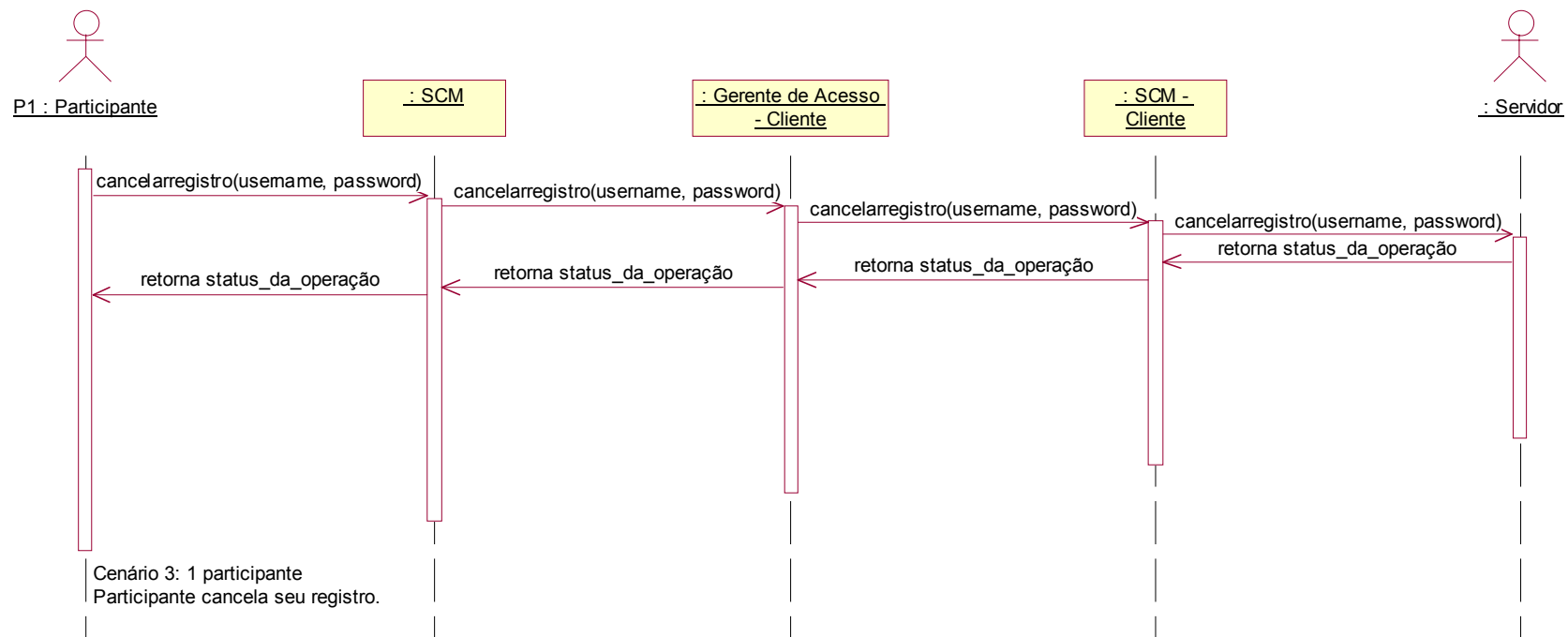
#### A4.4.2) Visão Interacional

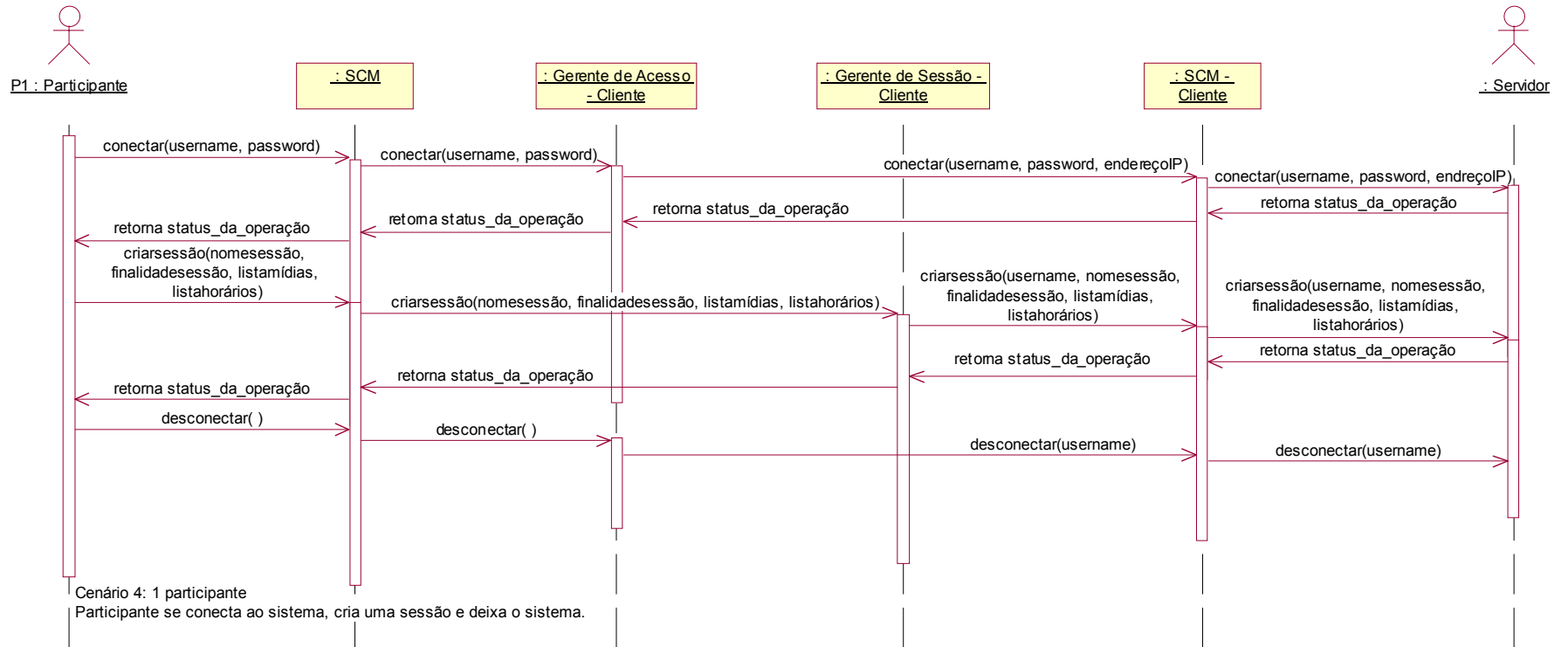
##### a) Cliente

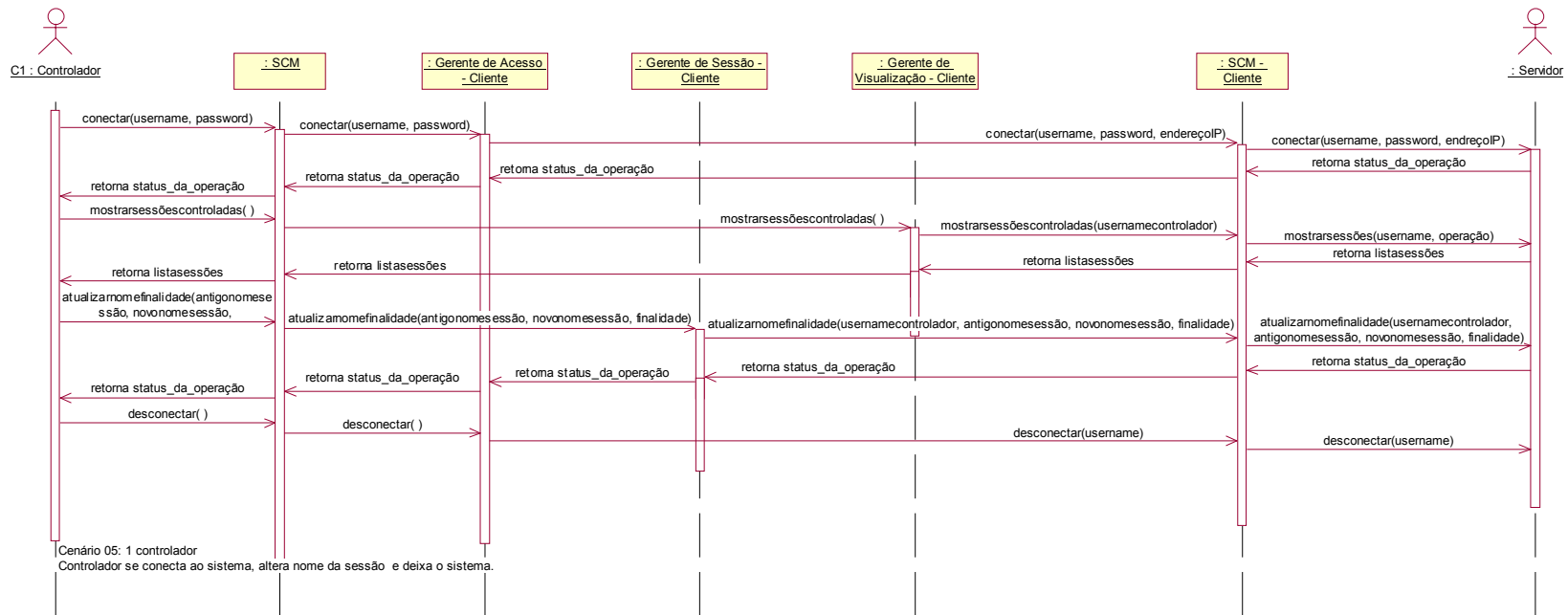
##### a1) Diagrama de Seqüência



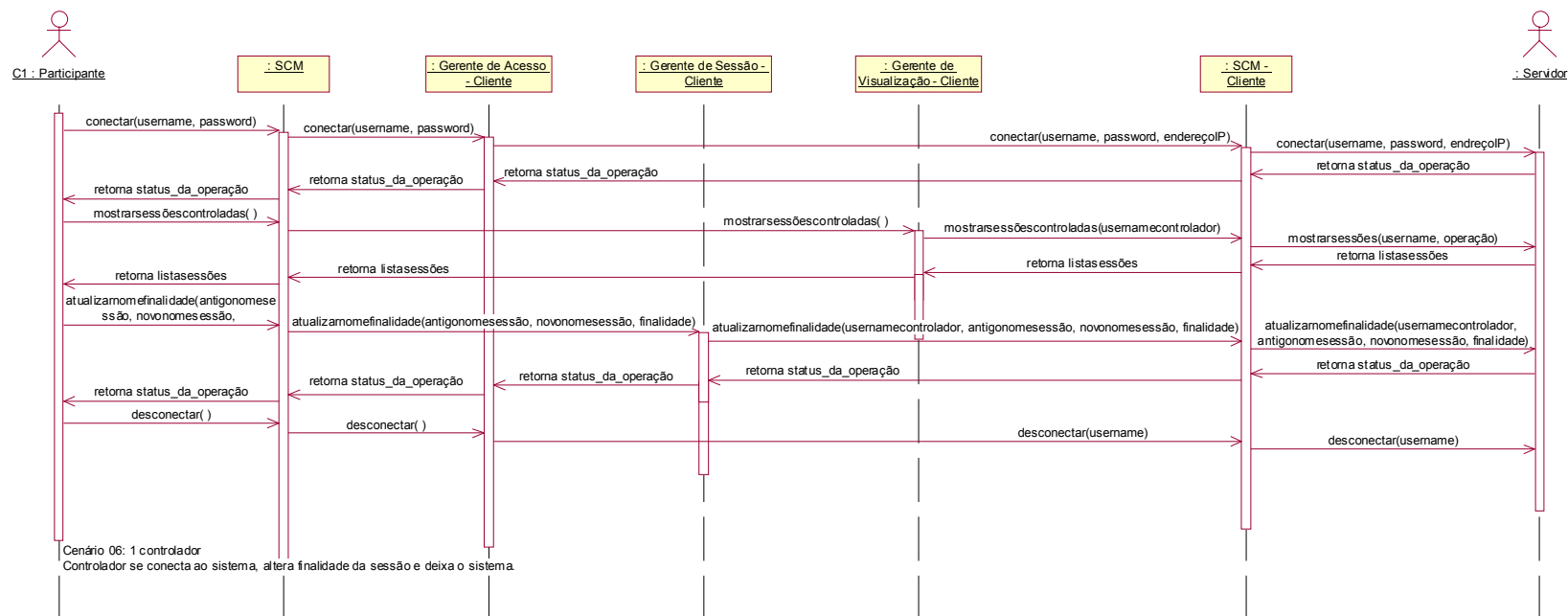


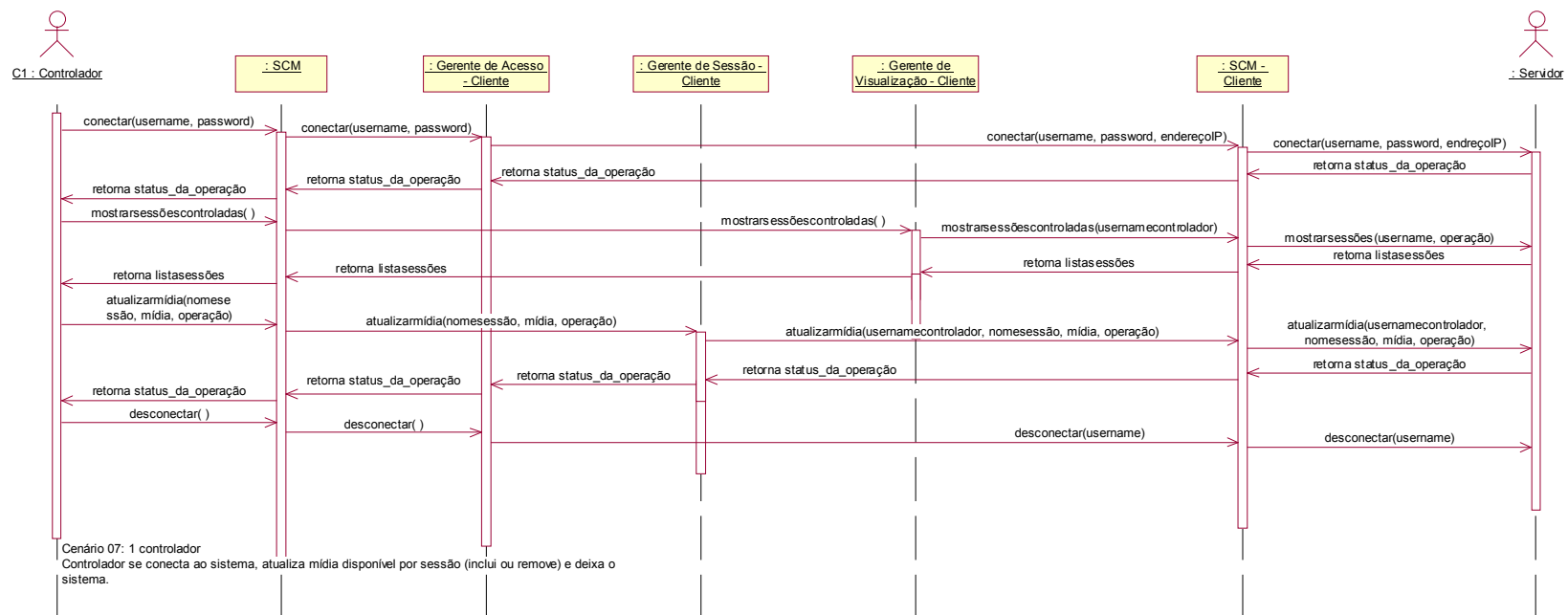


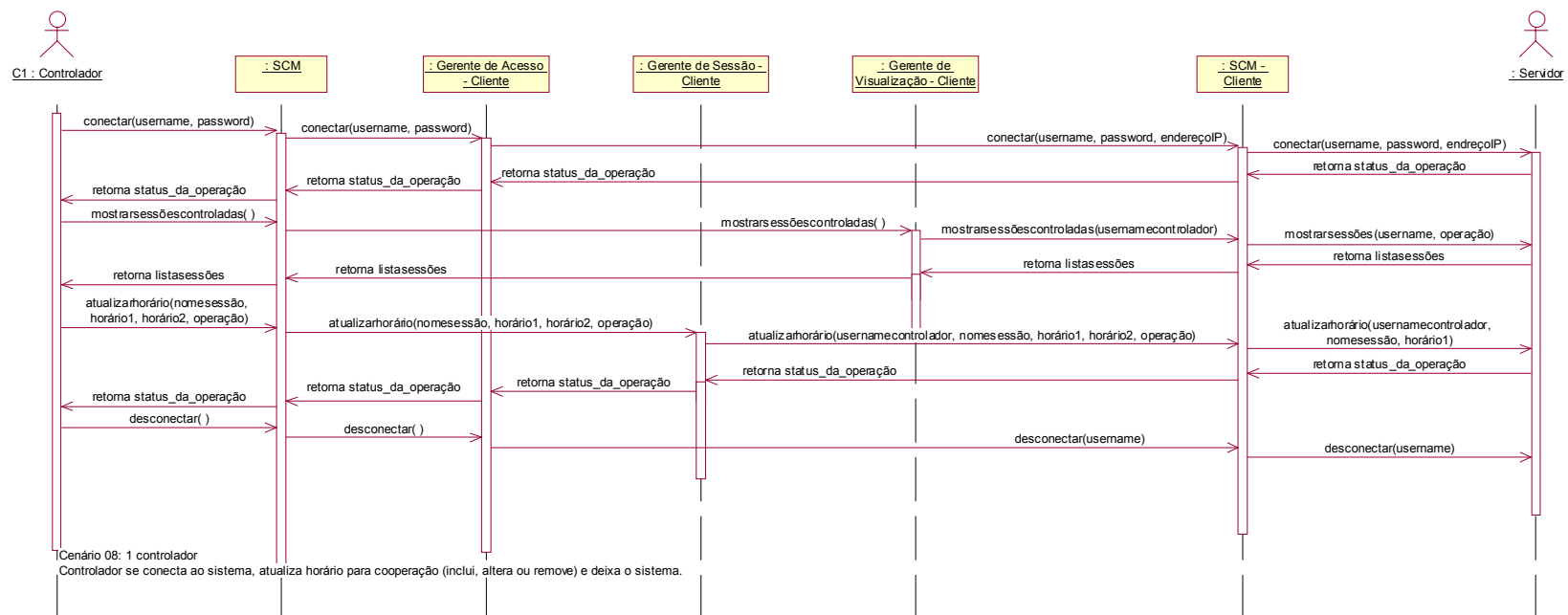


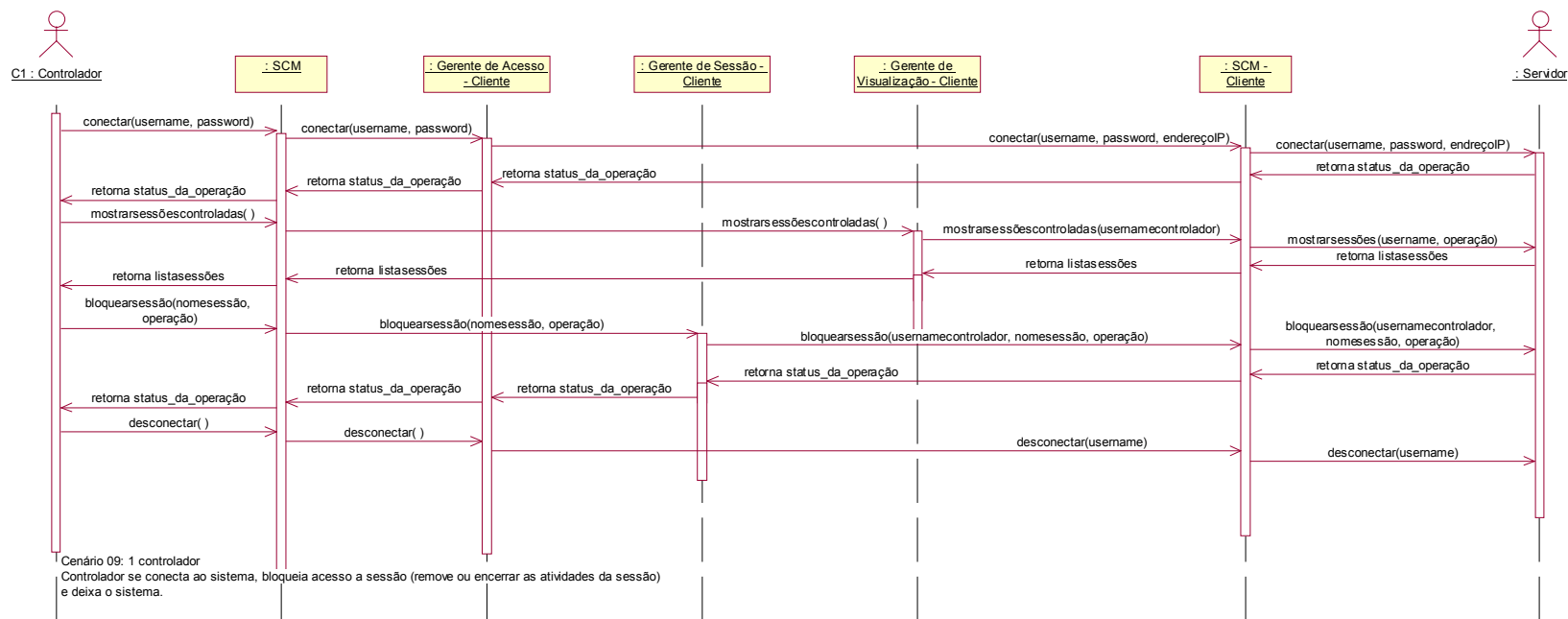


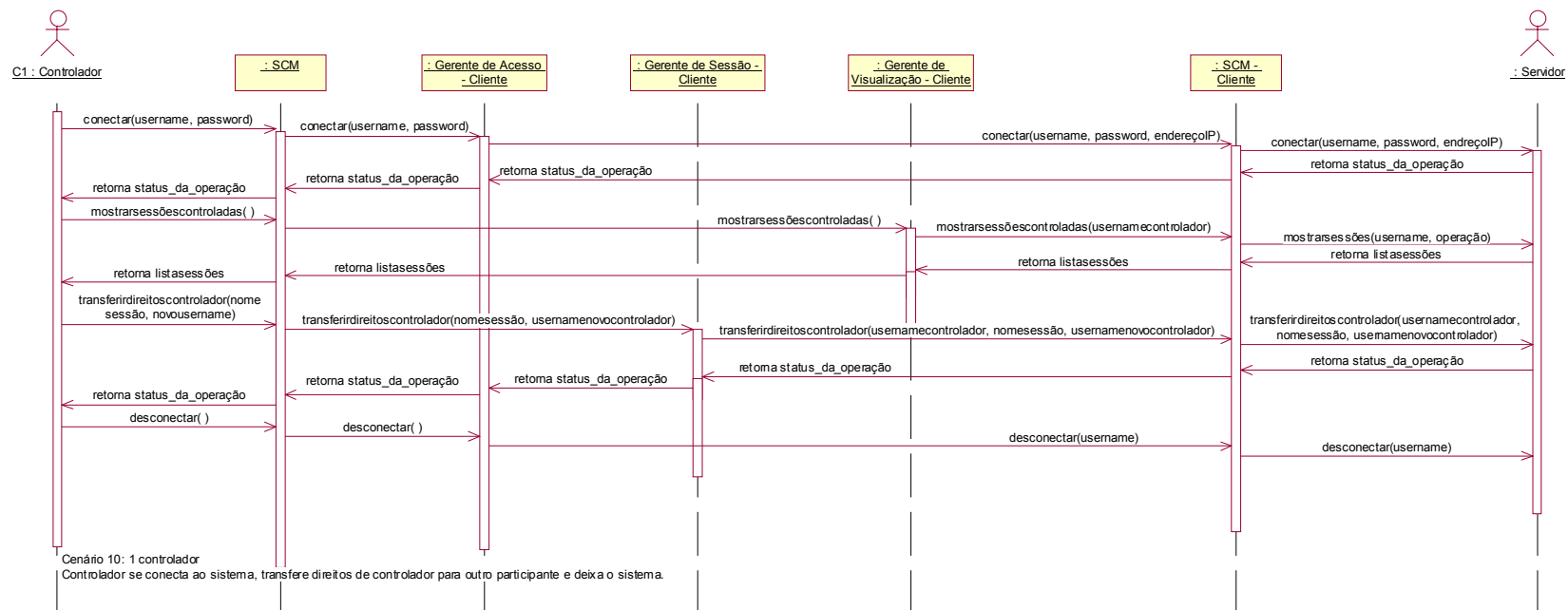


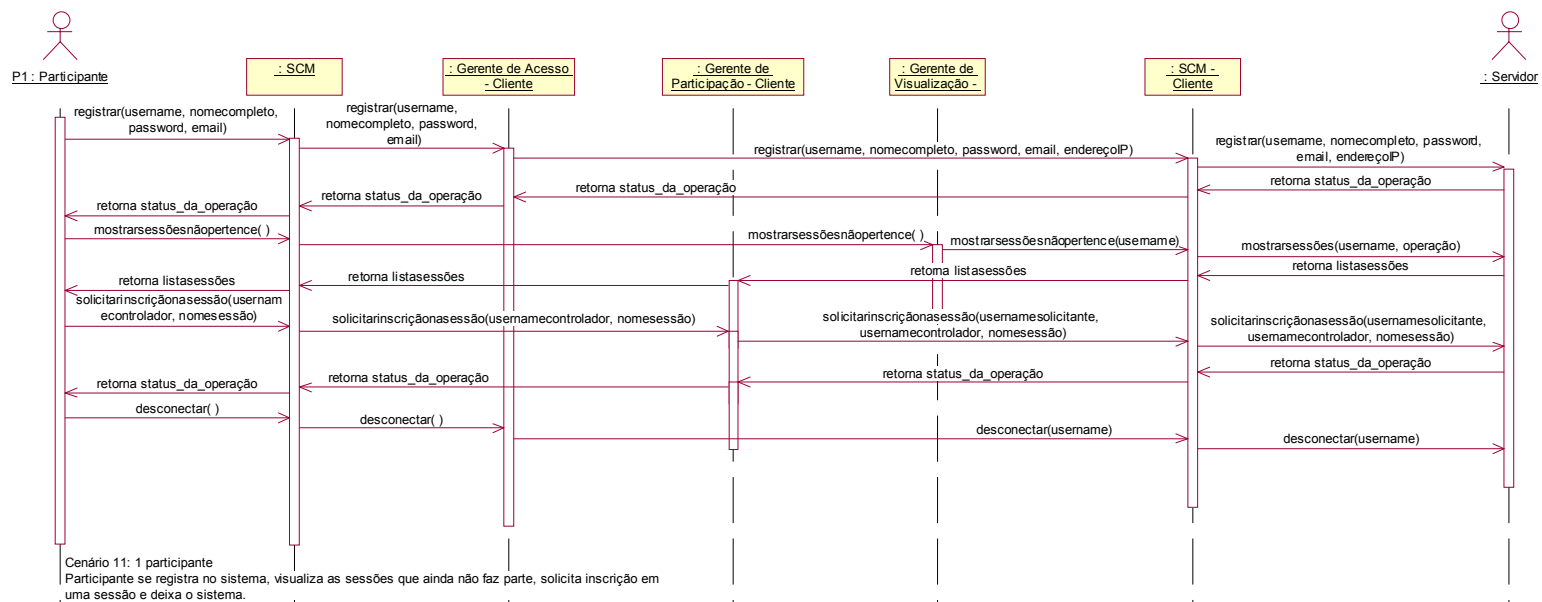


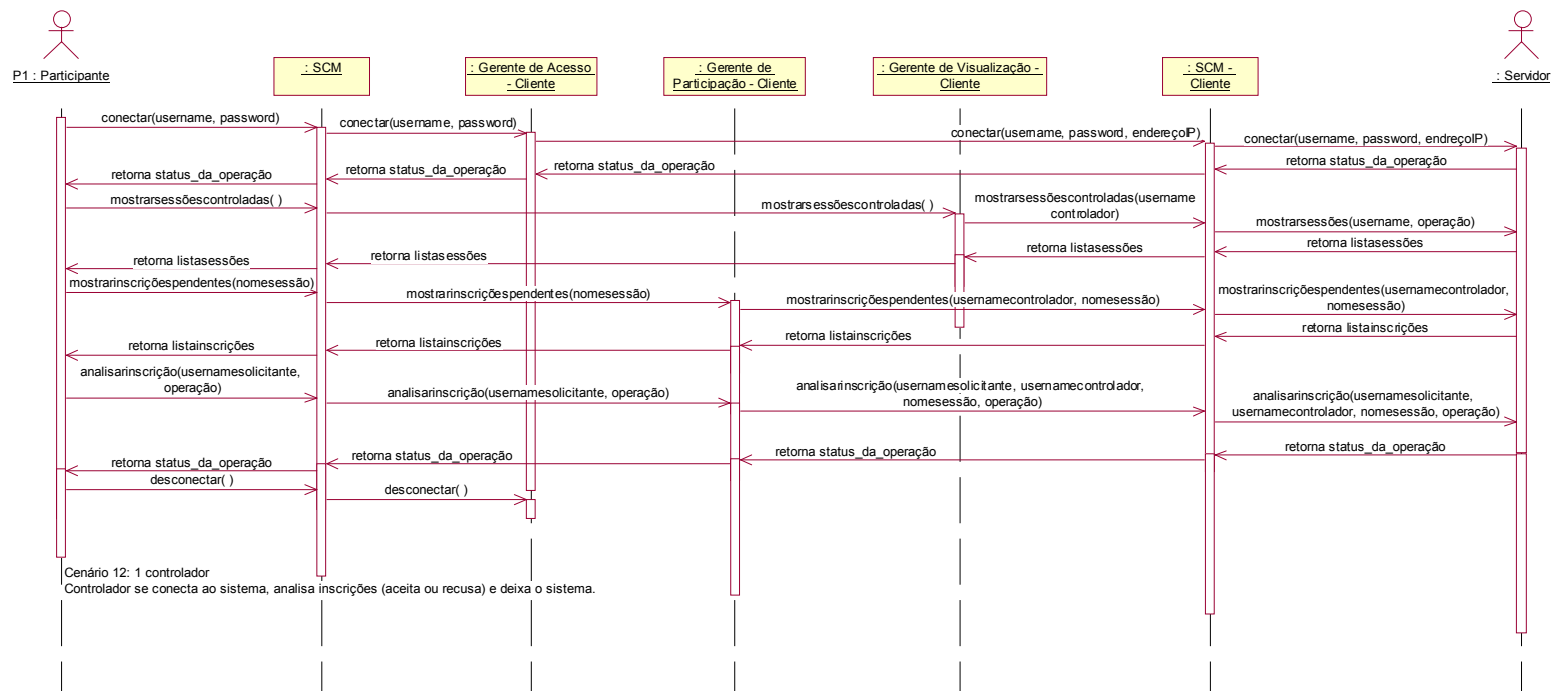


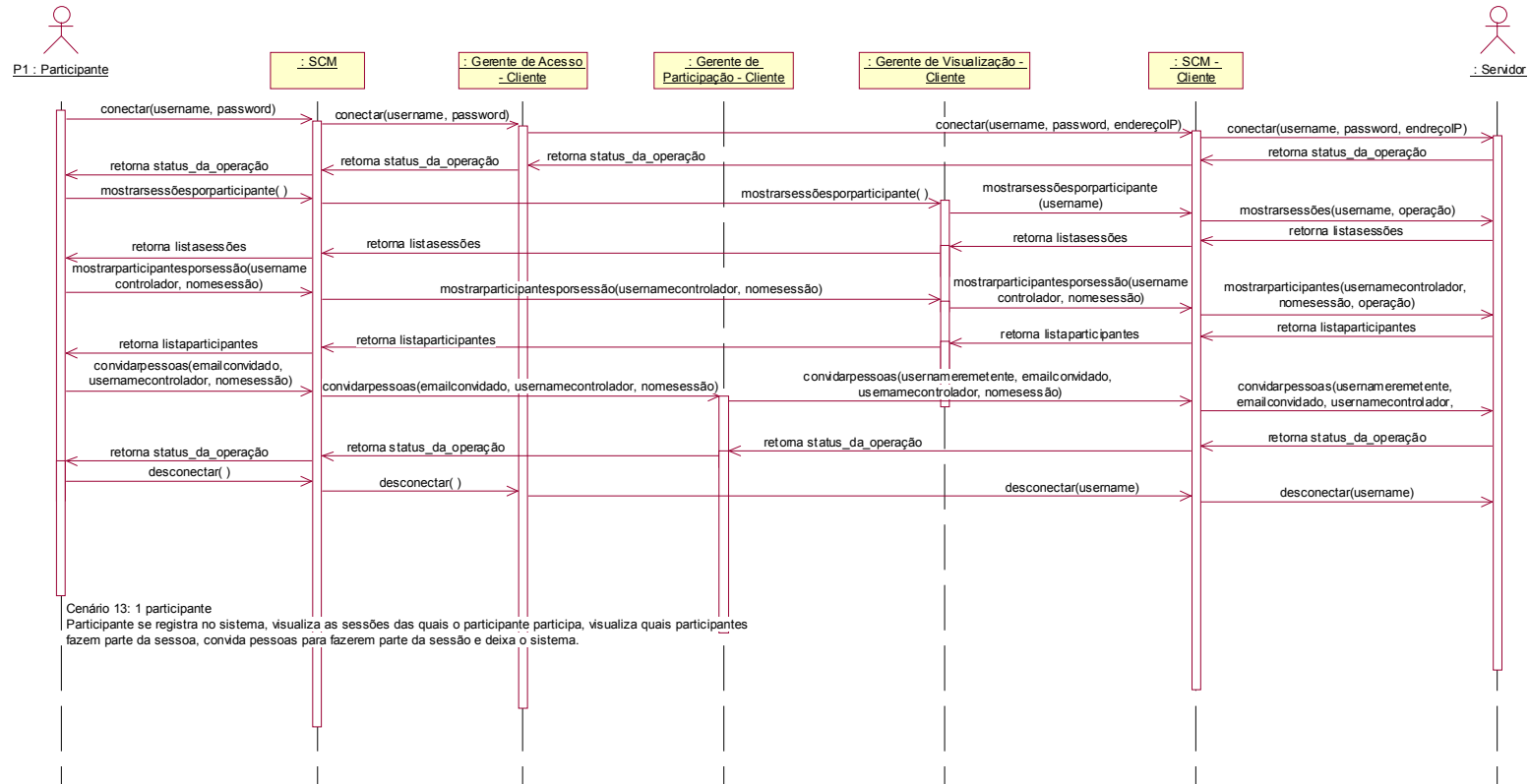




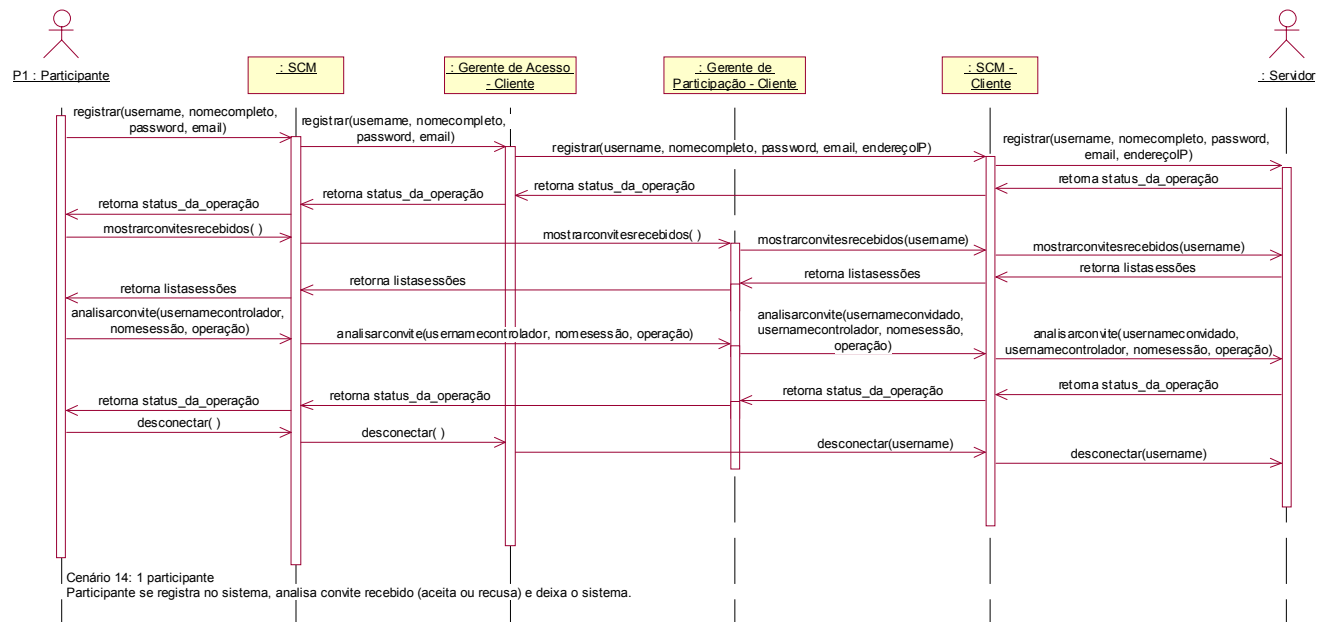


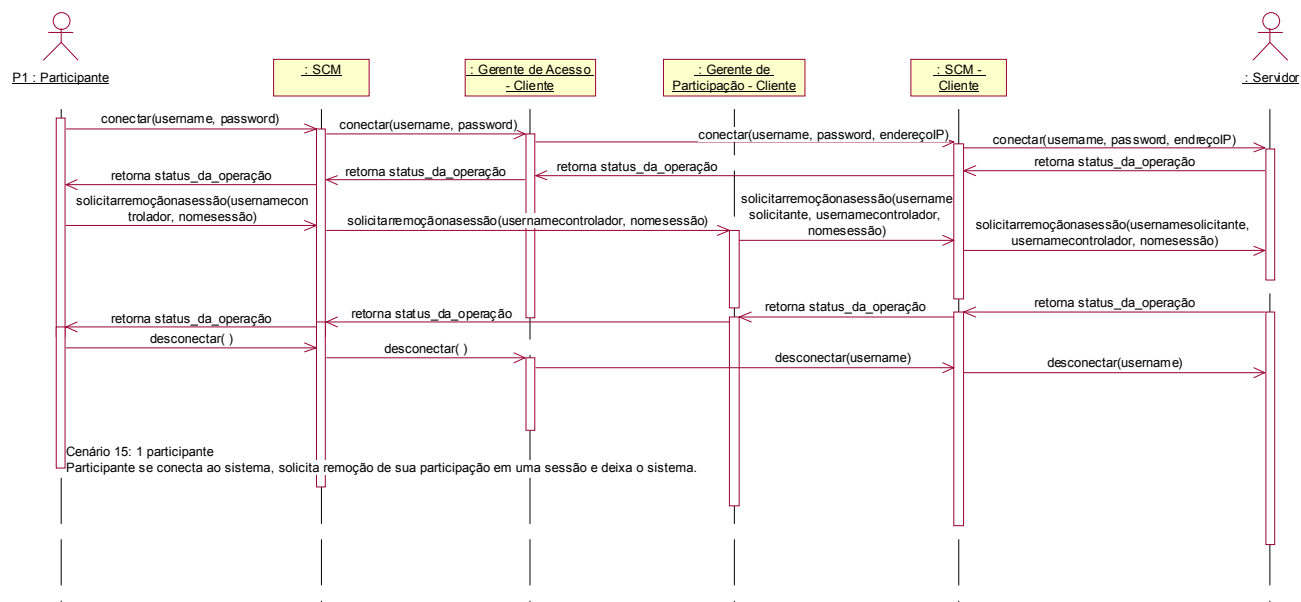


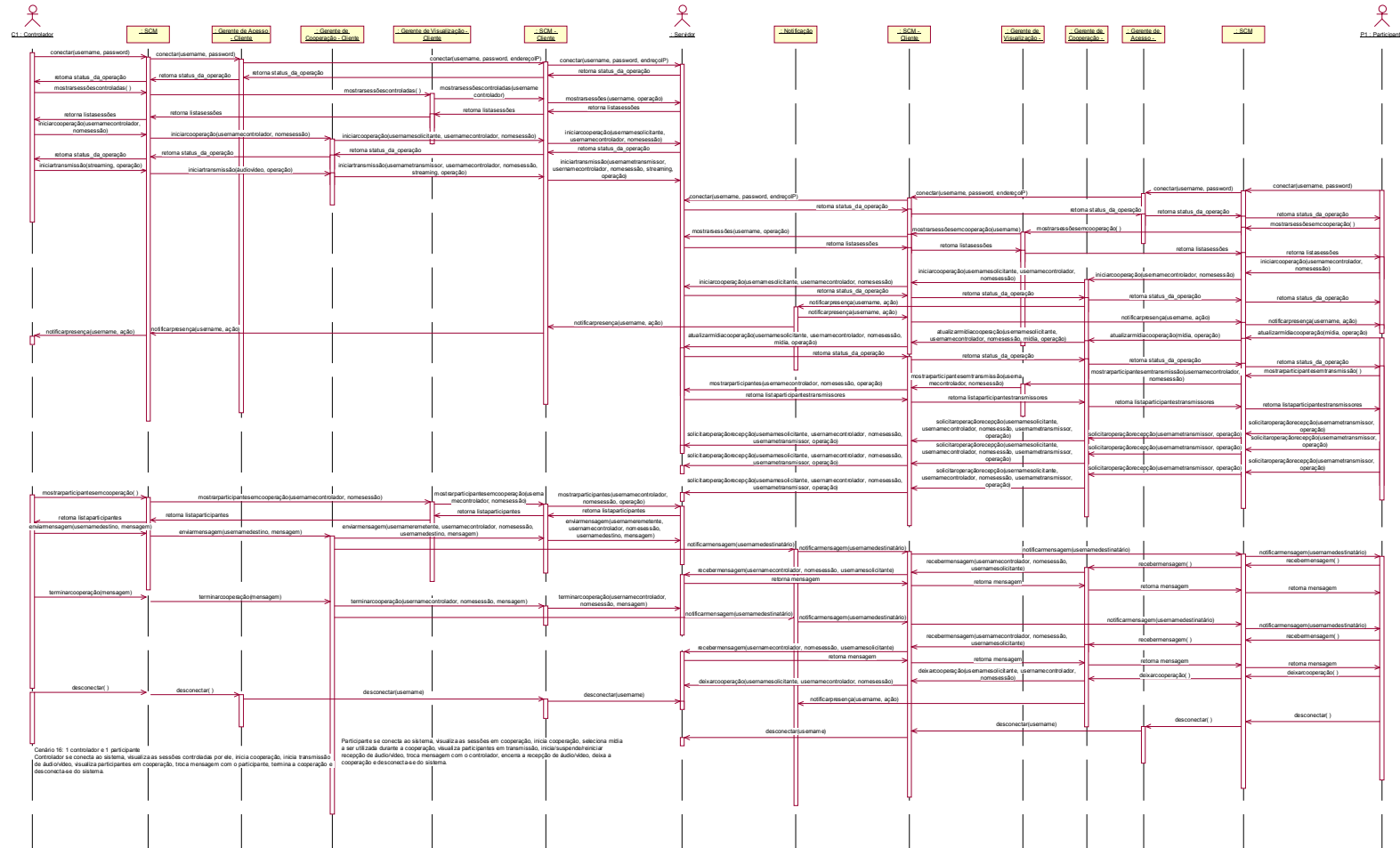






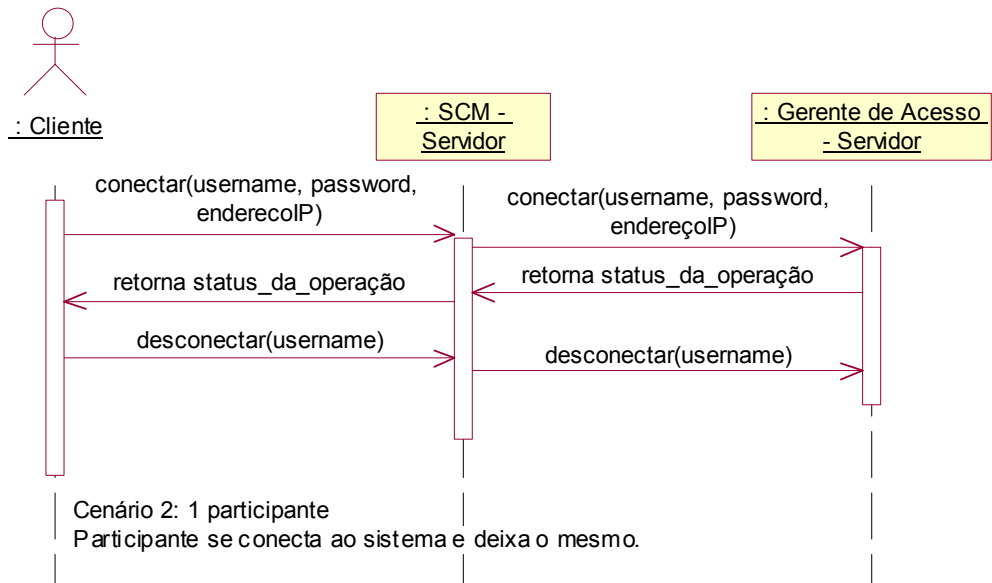
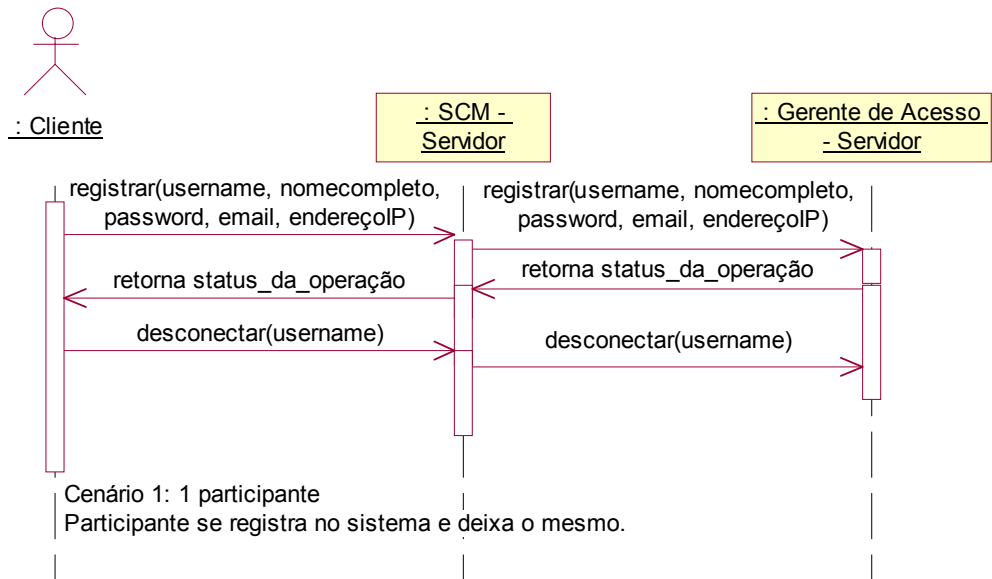


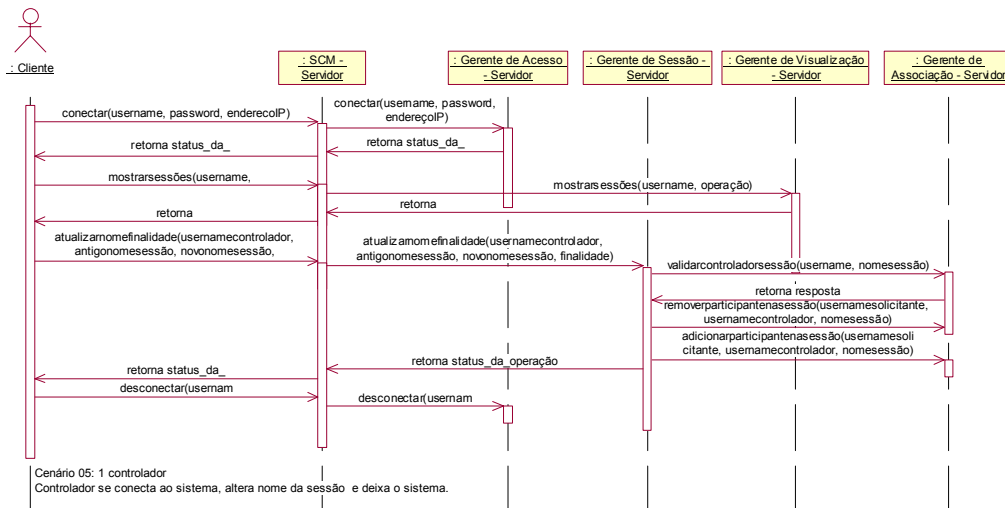
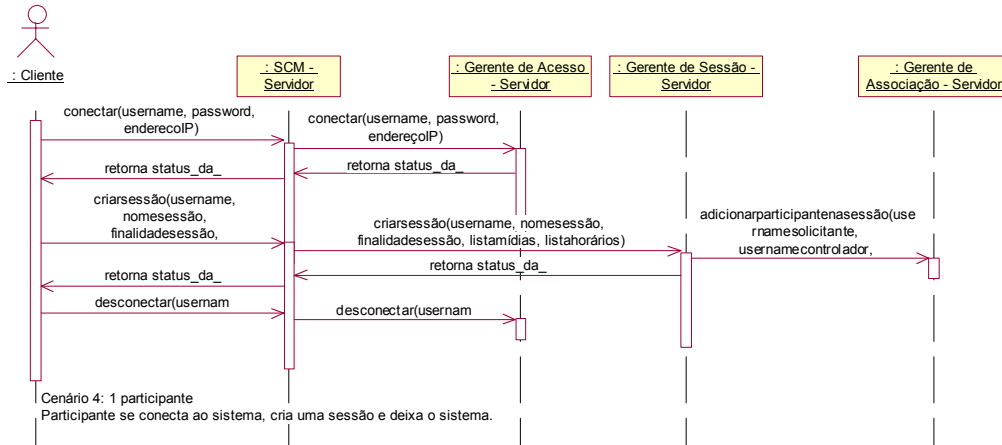
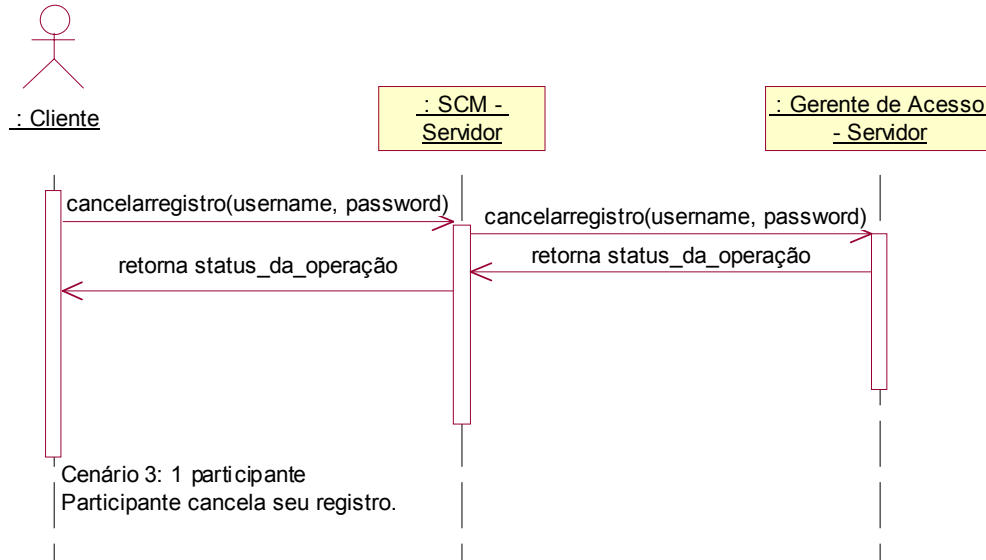


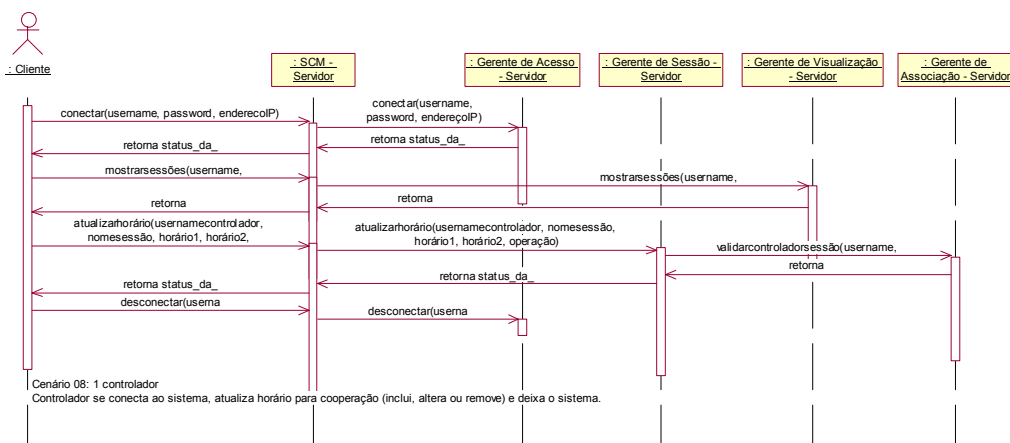
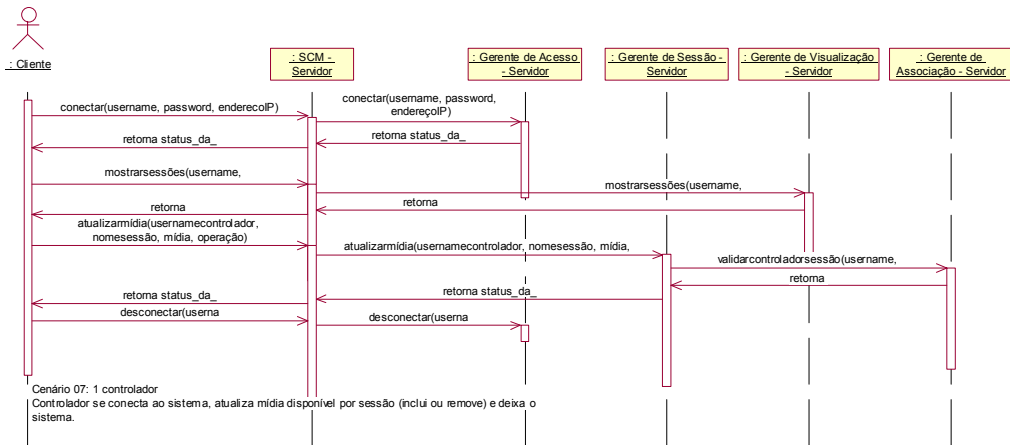
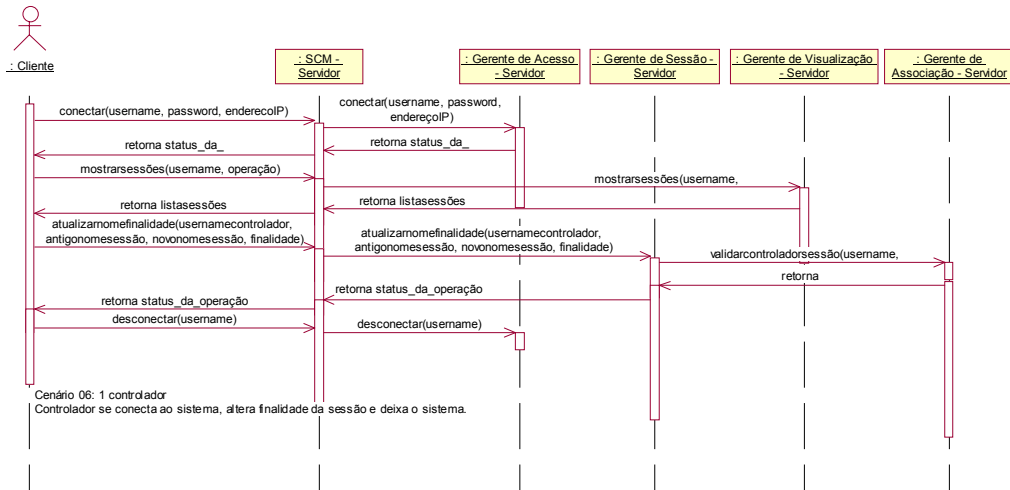


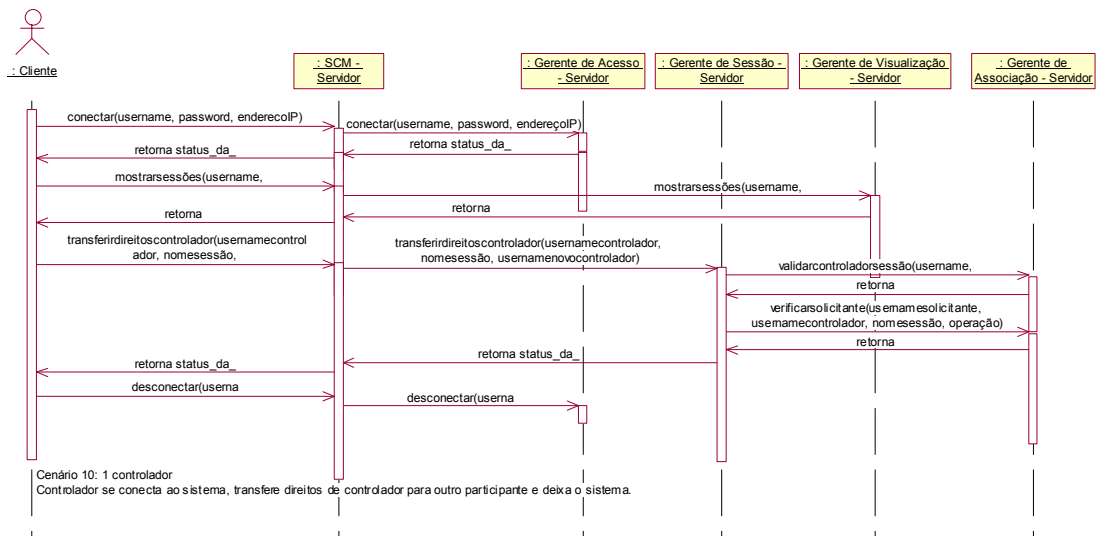
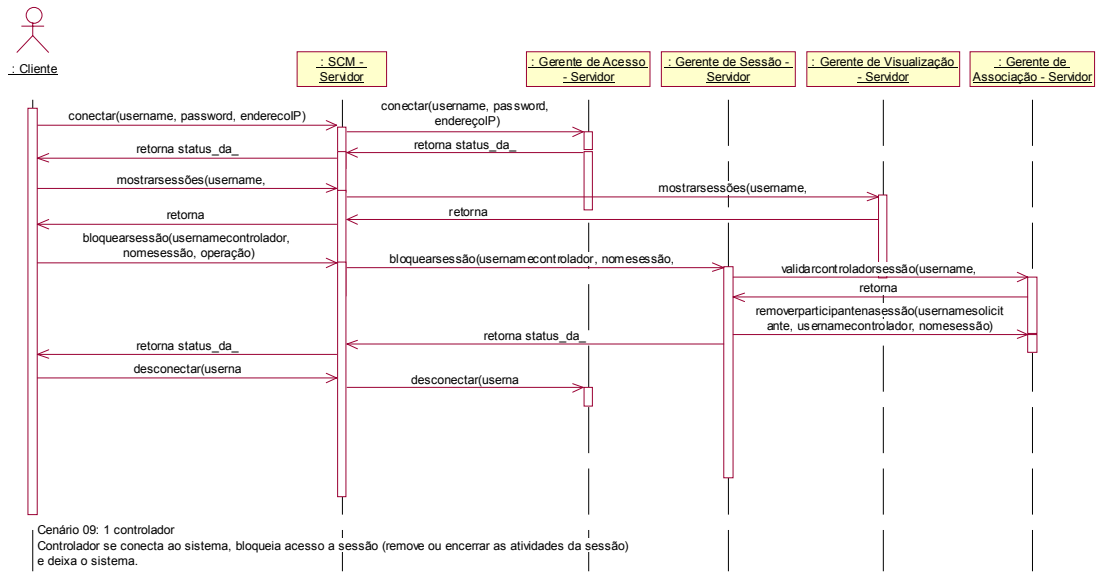
b) Servidor

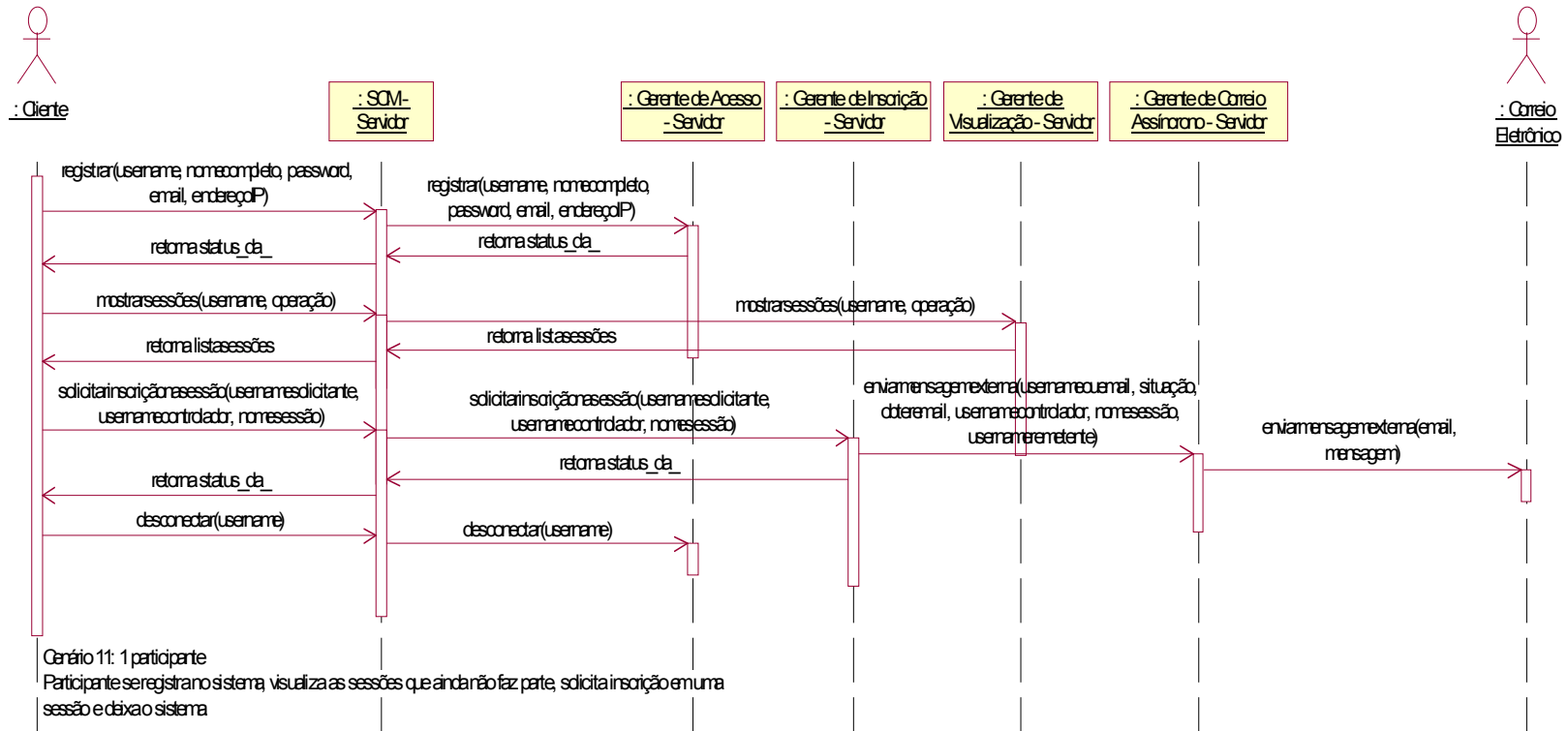
b1) Diagrama de Seqüência



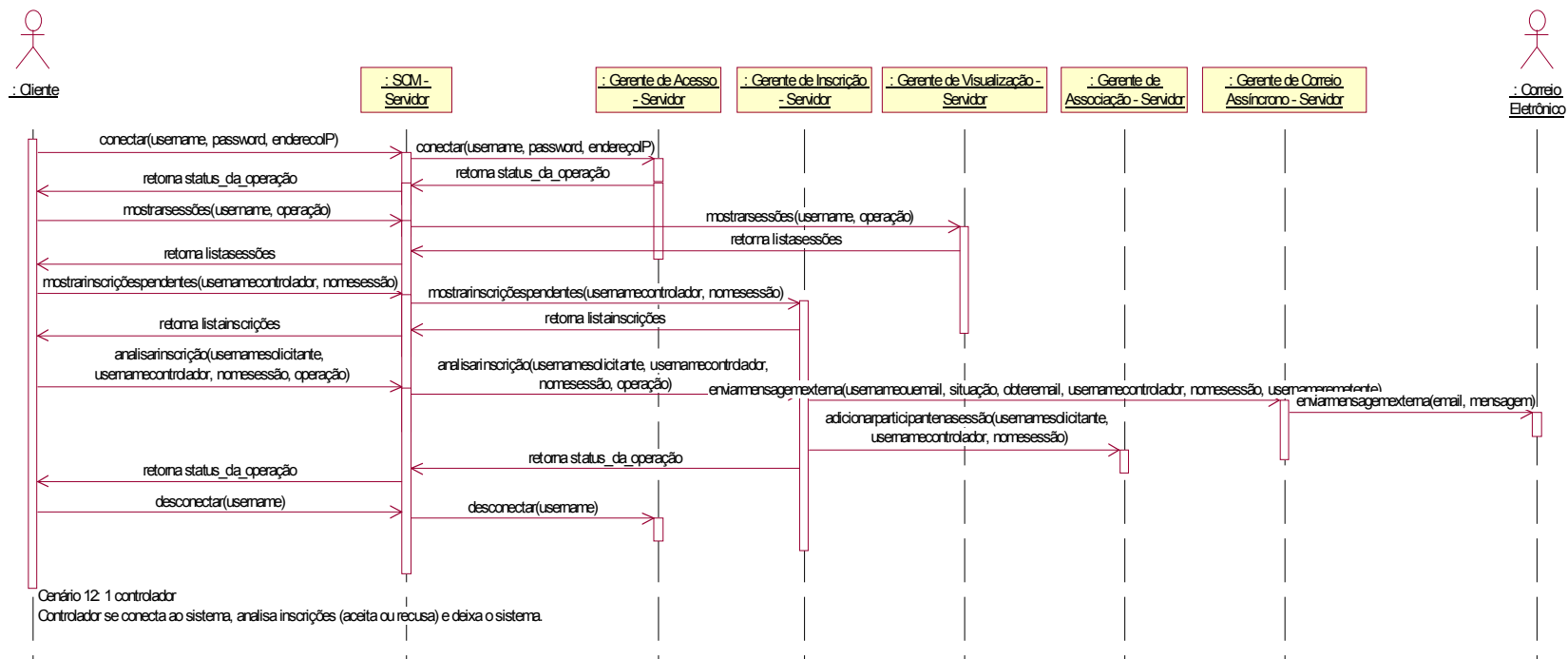


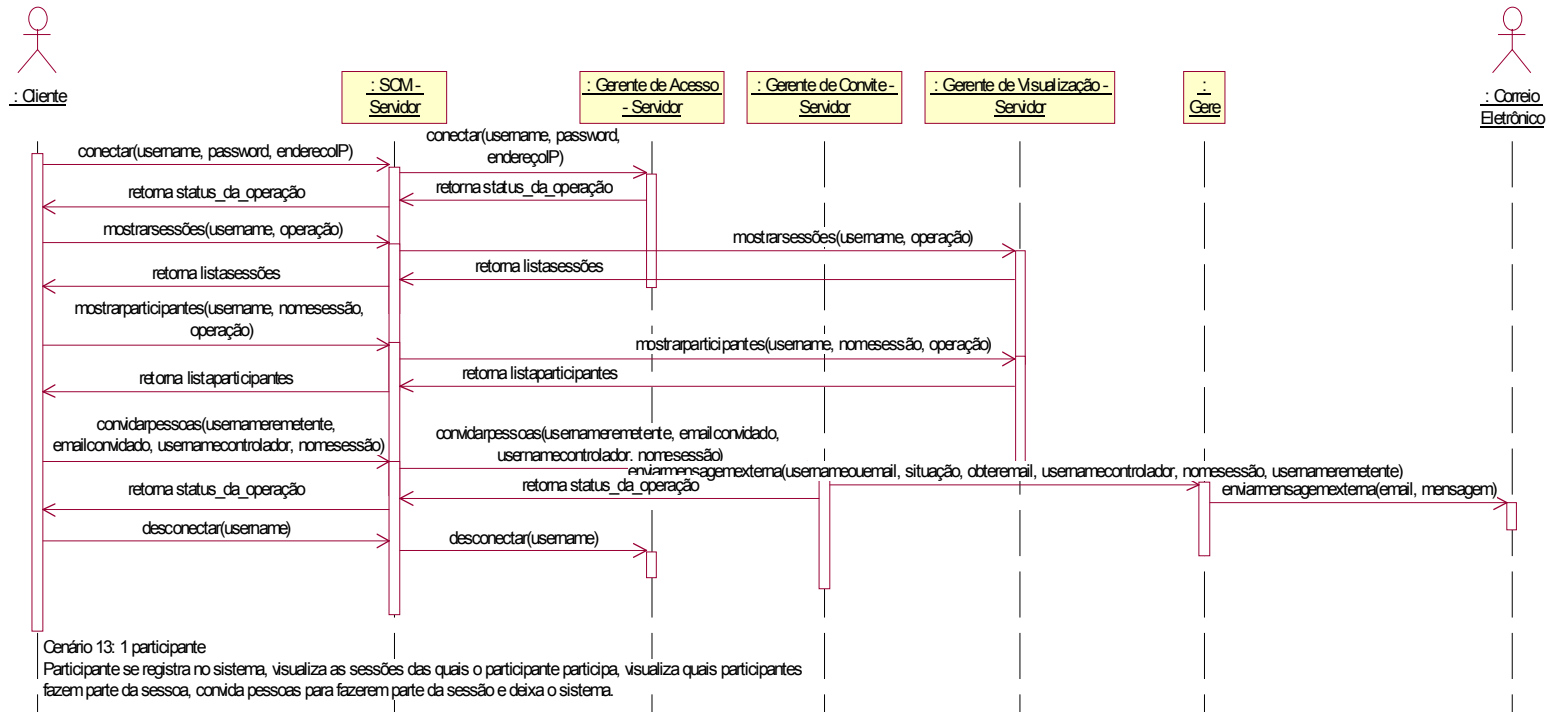


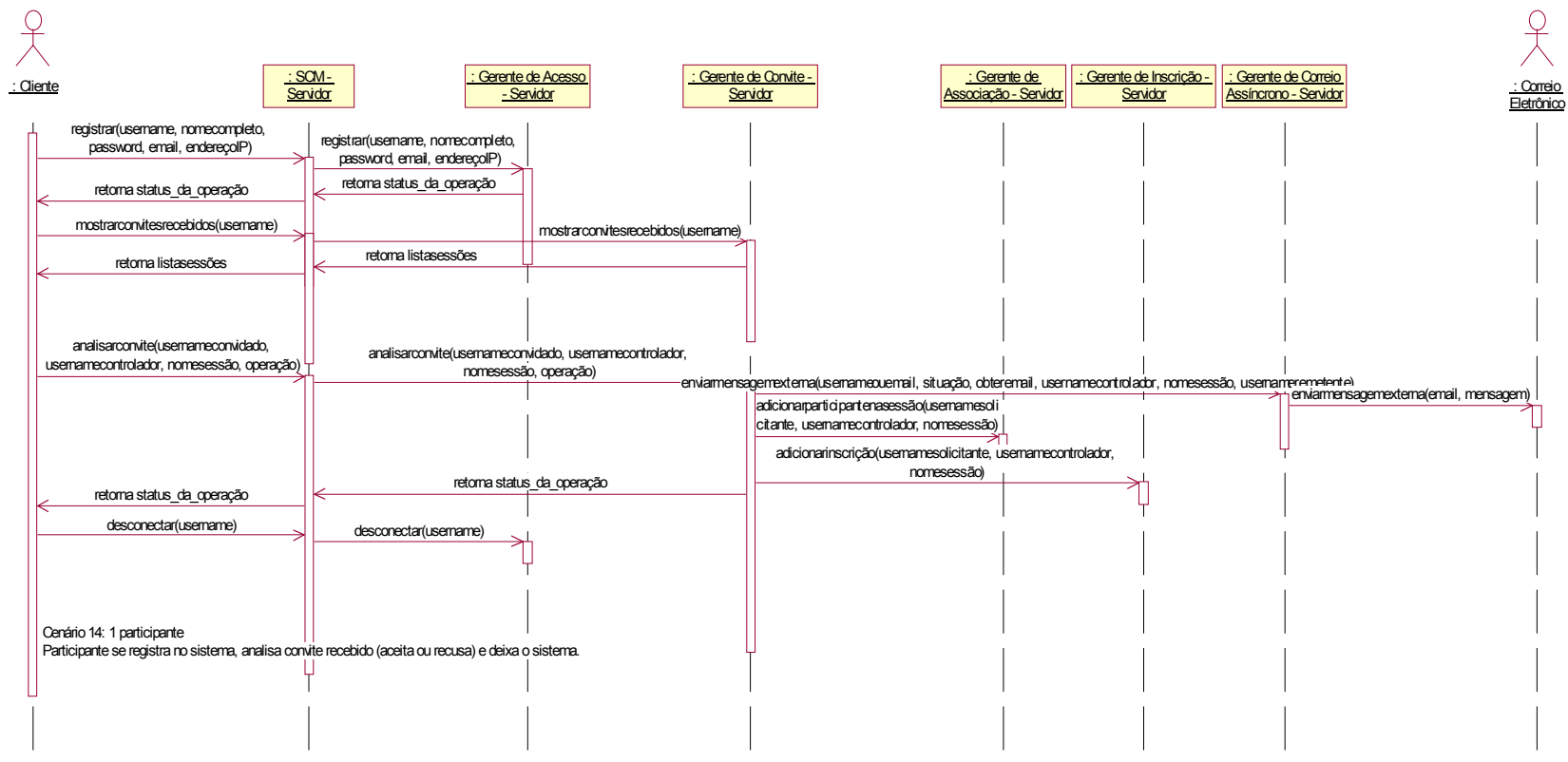


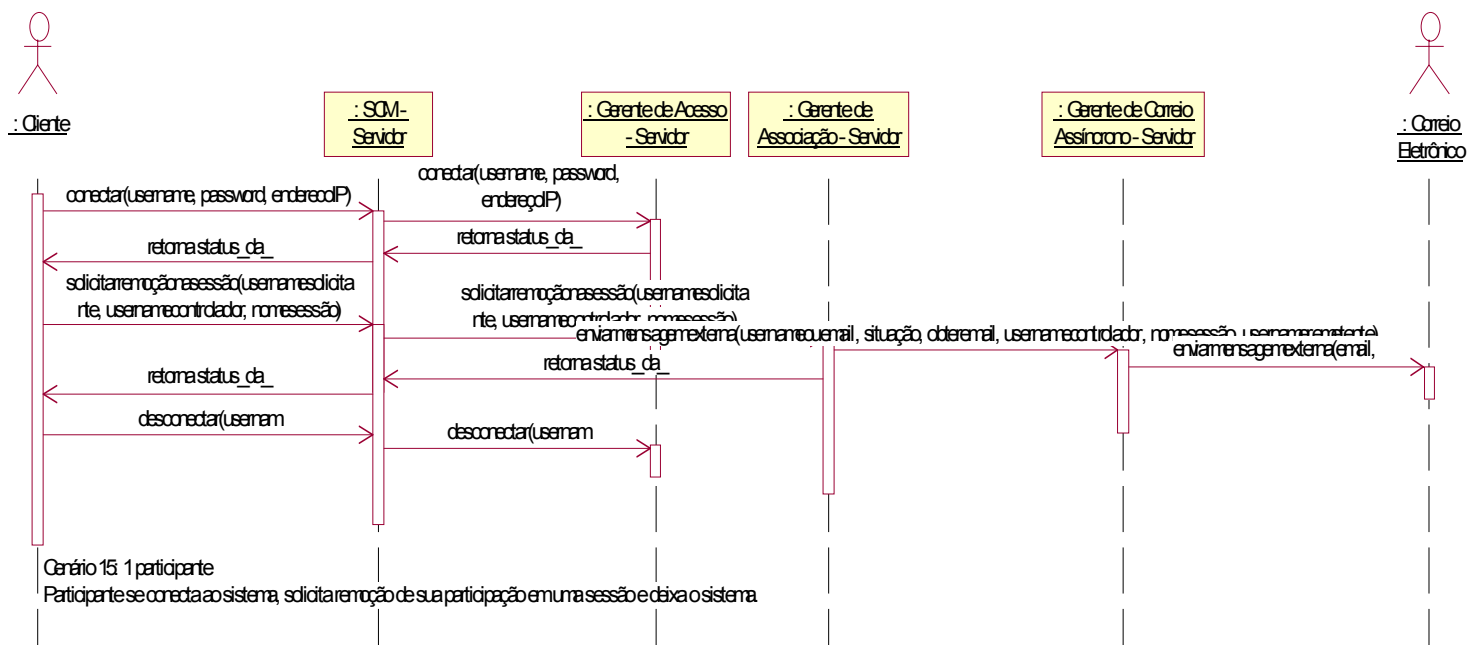


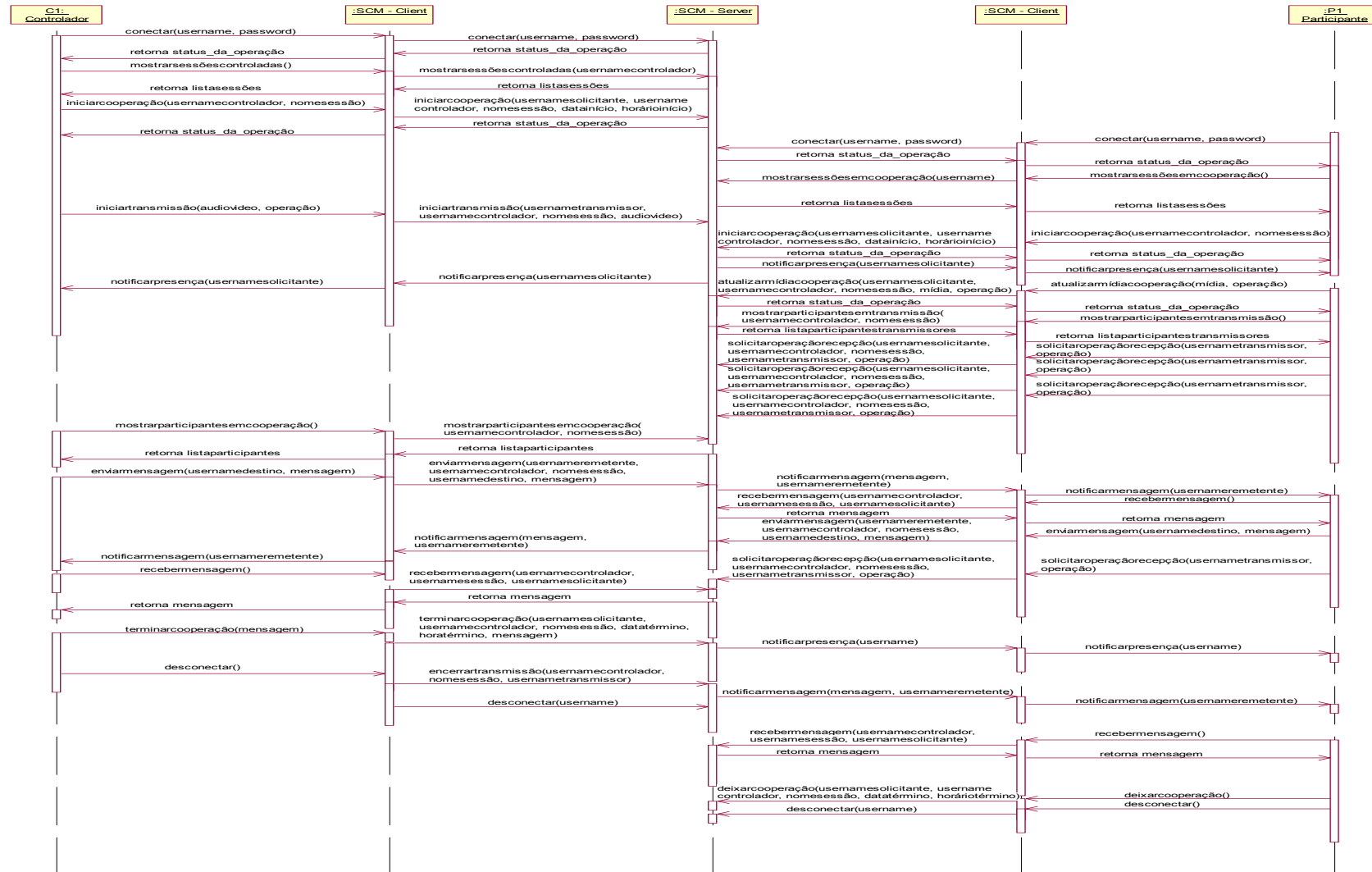












Cenário 16: 1 controlador e 1 participante  
 Controlador se conecta ao sistema, visualiza as sessões controladas por ele, inicia cooperação, inicia transmissão de áudio/vídeo, visualiza participantes em cooperação, troca mensagem com o participante, termina a cooperação e desconecta-se do sistema.

Participante se conecta ao sistema, visualiza as sessões em cooperação, inicia cooperação, seleciona mídia a ser utilizada durante a cooperação, visualiza participantes em transmissão, inicia/suspende/reinicia recepção de áudio/vídeo, troca mensagem com o controlador, encerra a recepção de áudio/vídeo, deixa a cooperação e desconecta-se do sistema.

A4.4.3) Visão Comportamental

a) Cliente

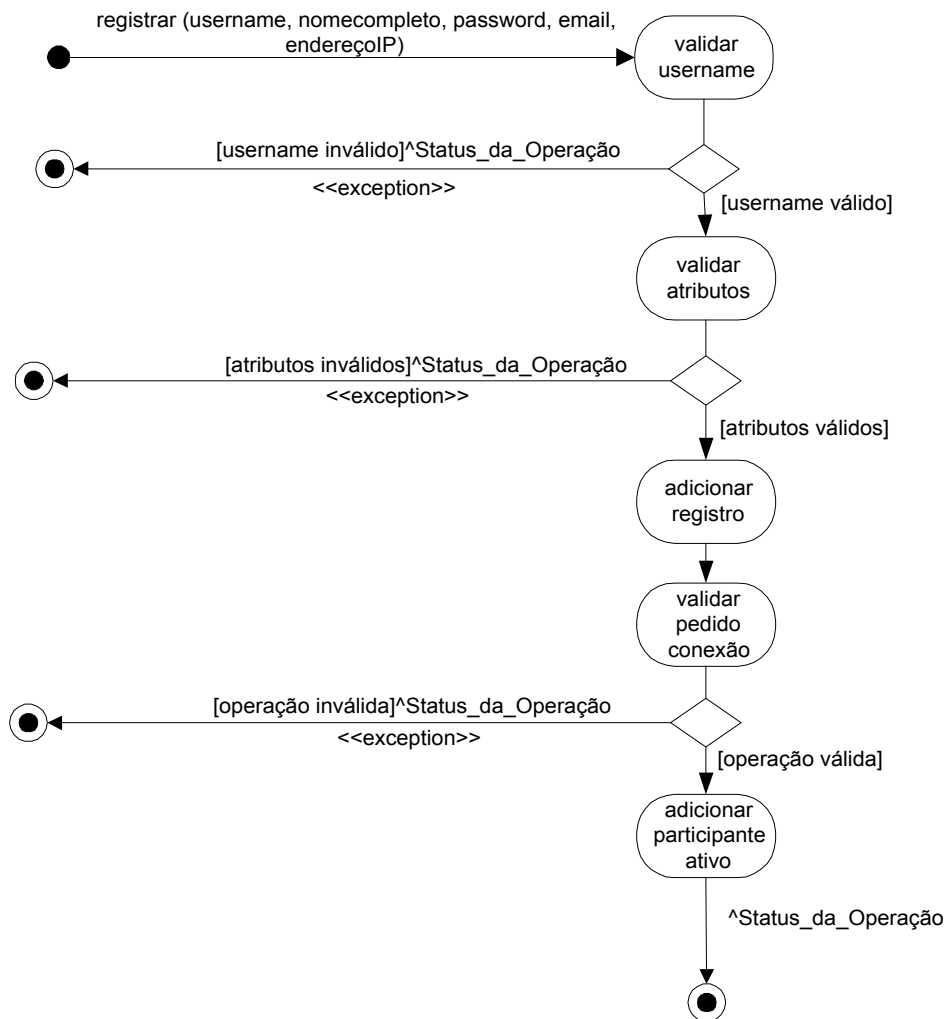
a1) Diagrama de Atividades

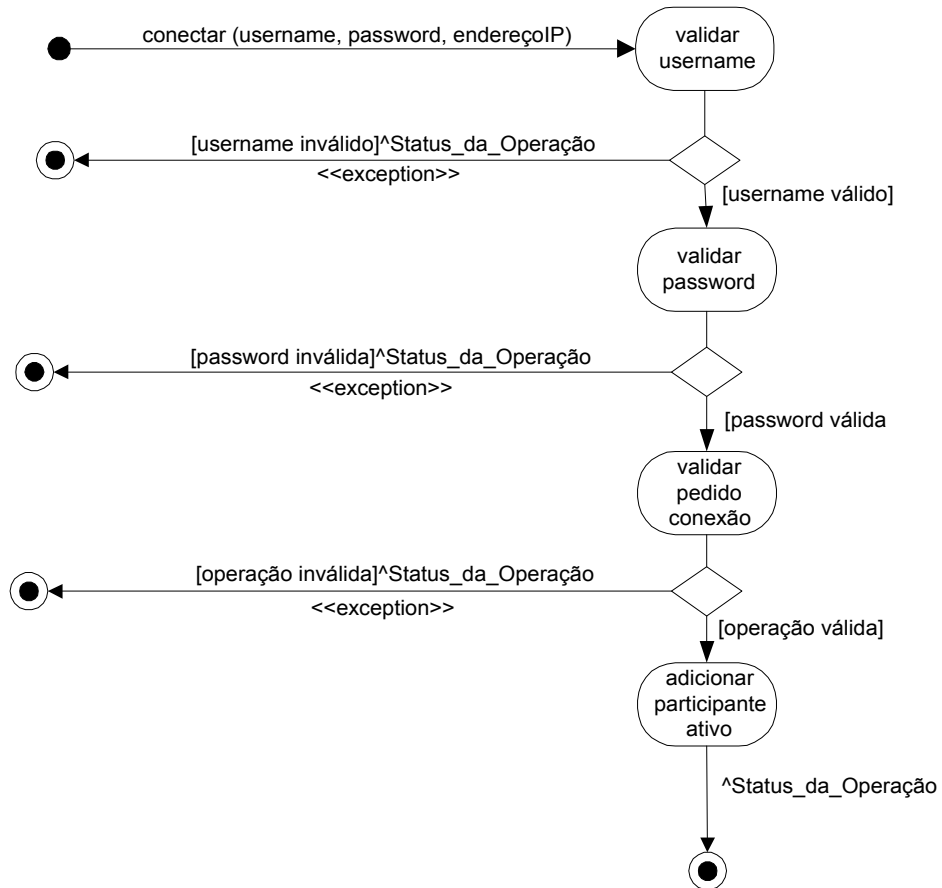
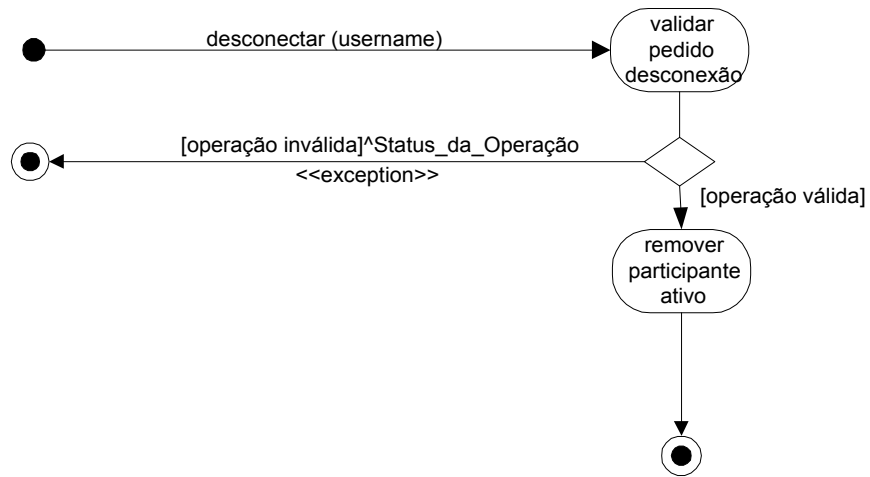
b) Servidor

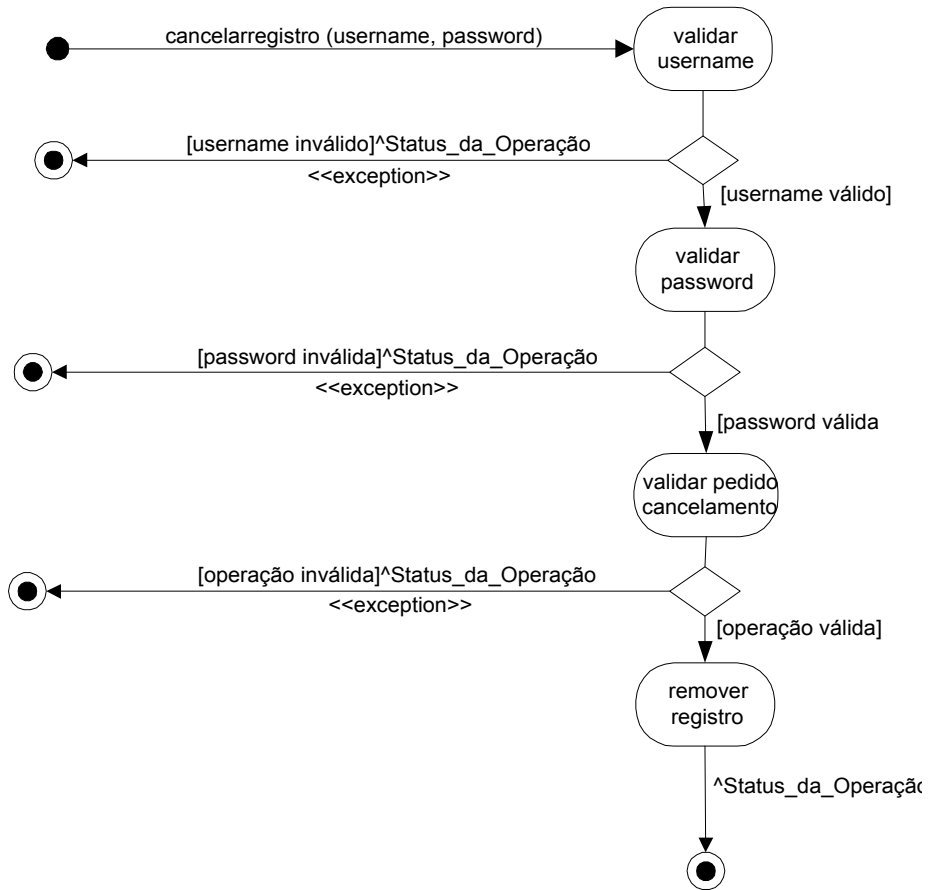
b1) Diagrama de Atividades

Gerente de Acesso

- Registrar
- Desconectar
- Conectar
- Cancelar Registro



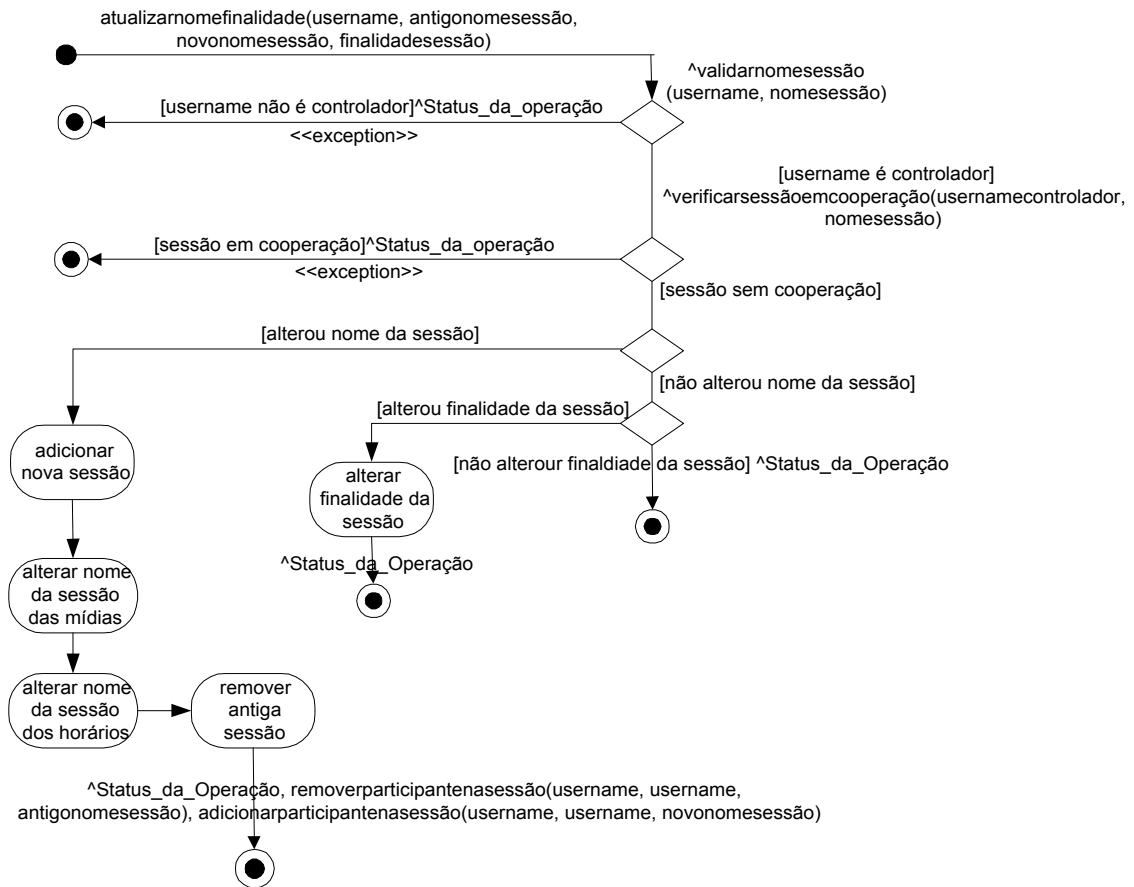
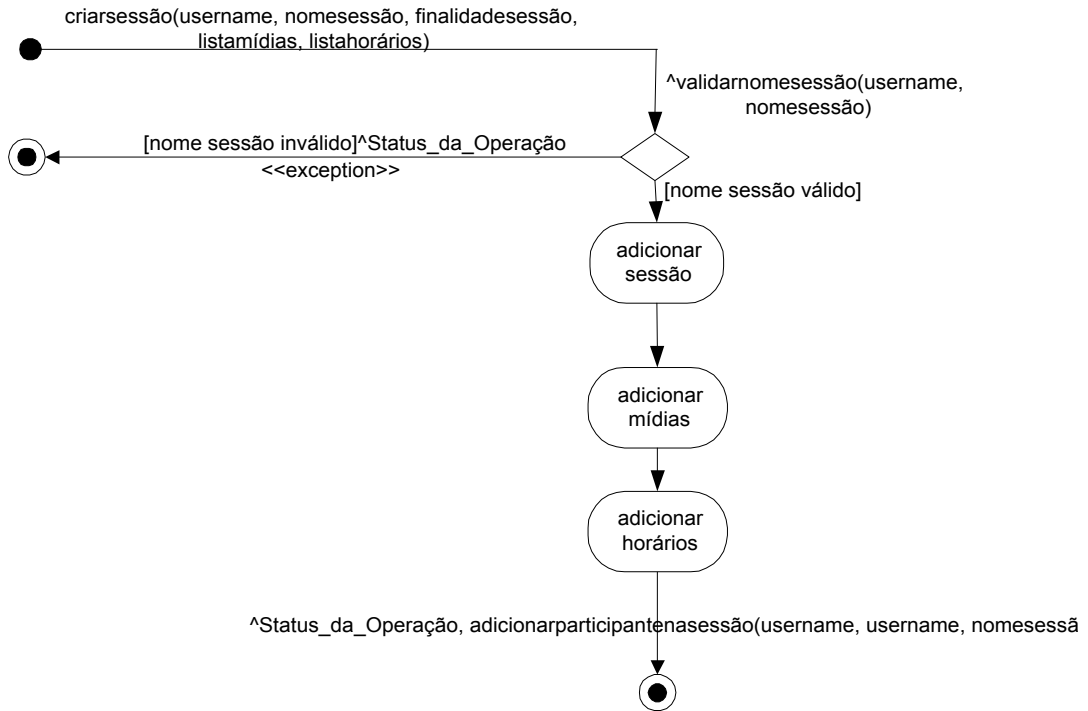


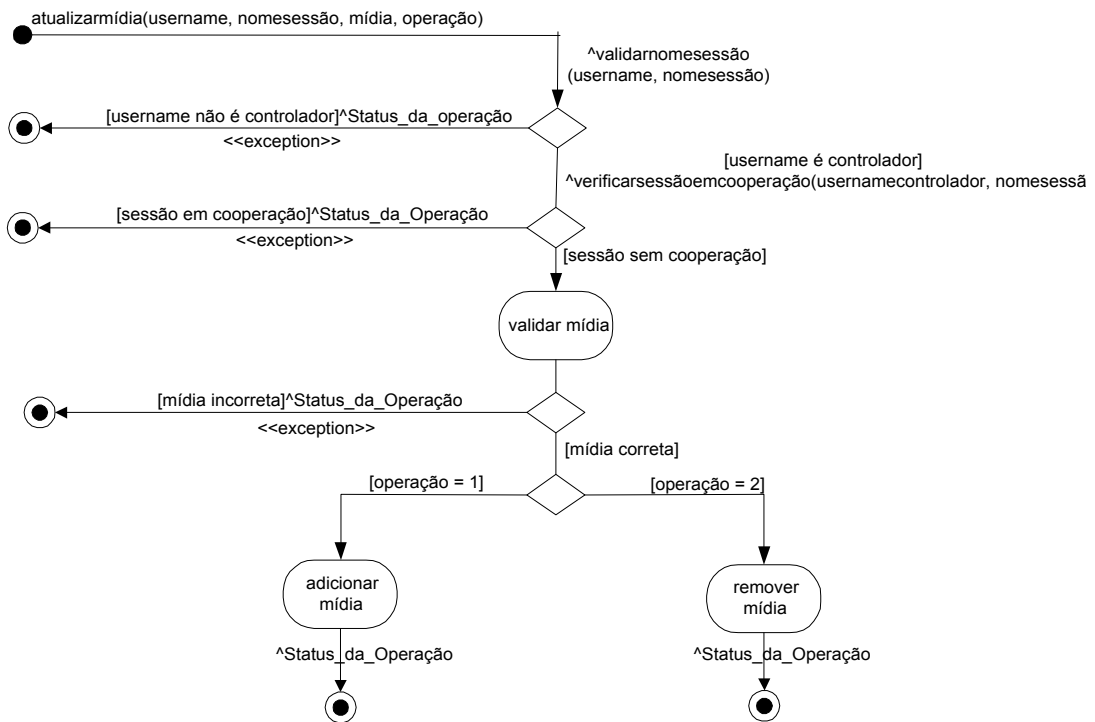
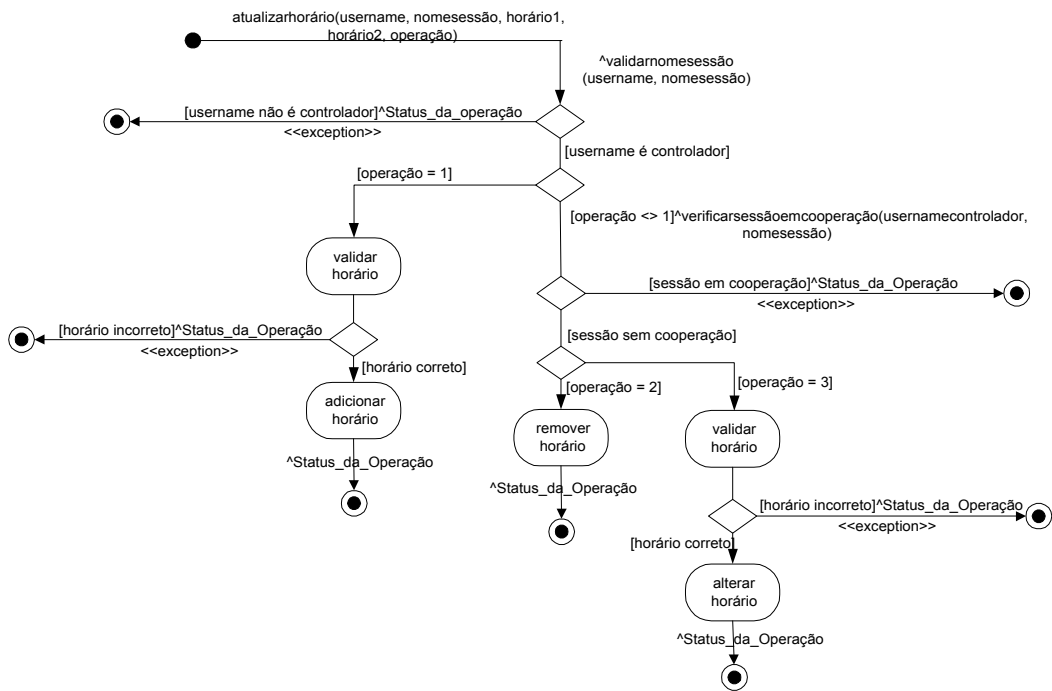


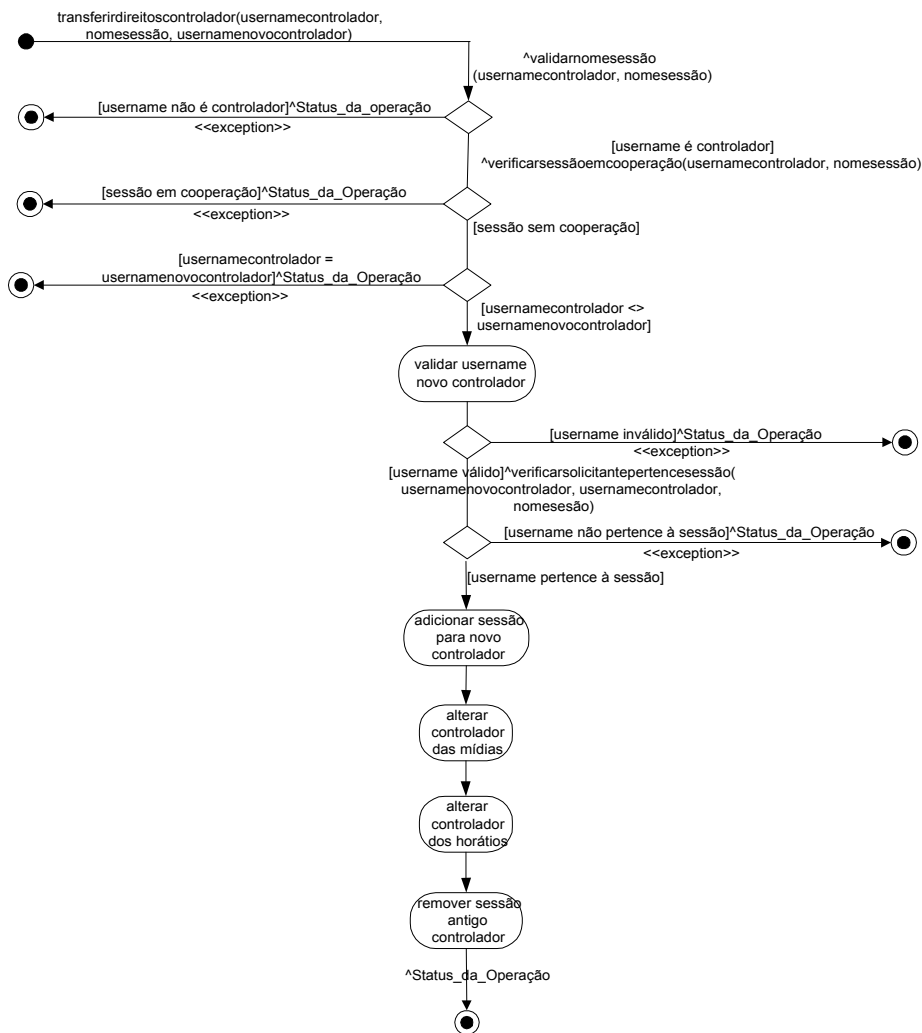
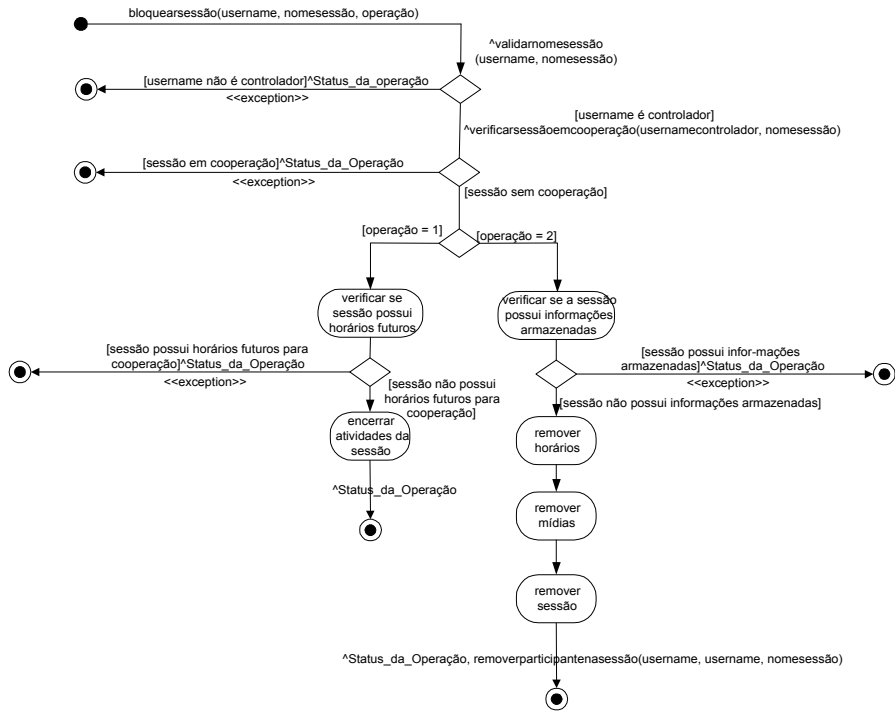
Gerente de Sessão

- Criar Sessão
- Atualizar Nome Finalidade
- Atualizar Horário
- Atualizar Mídia
- Bloquear Sessão
- Transferir Direitos Controlador
- Elaborar Pesquisa
- Mostrar Pesquisas Elaboradas
- Remover Pesquisa



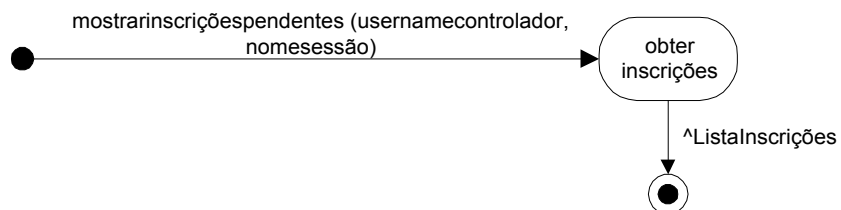
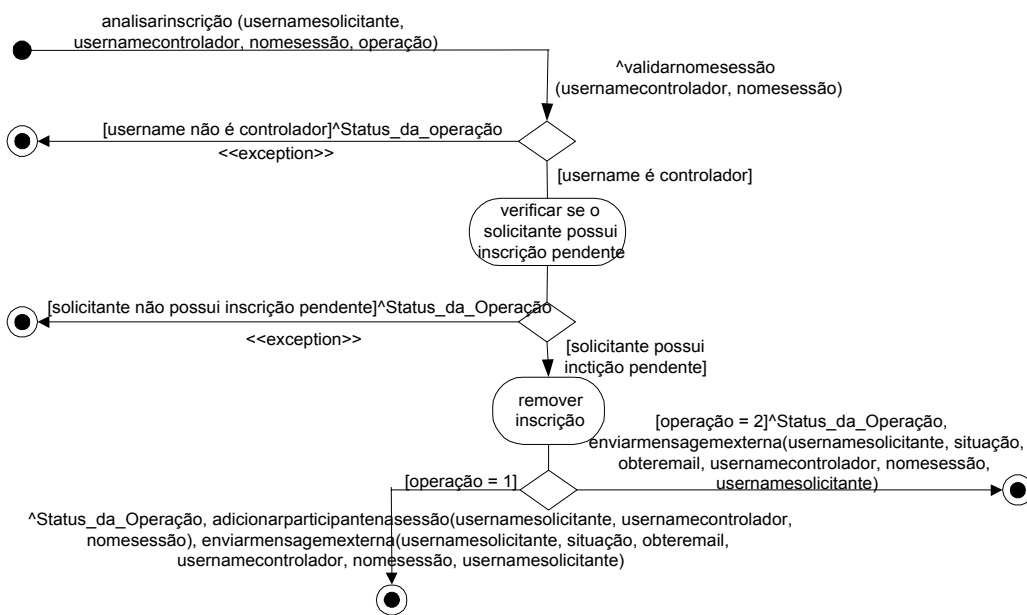
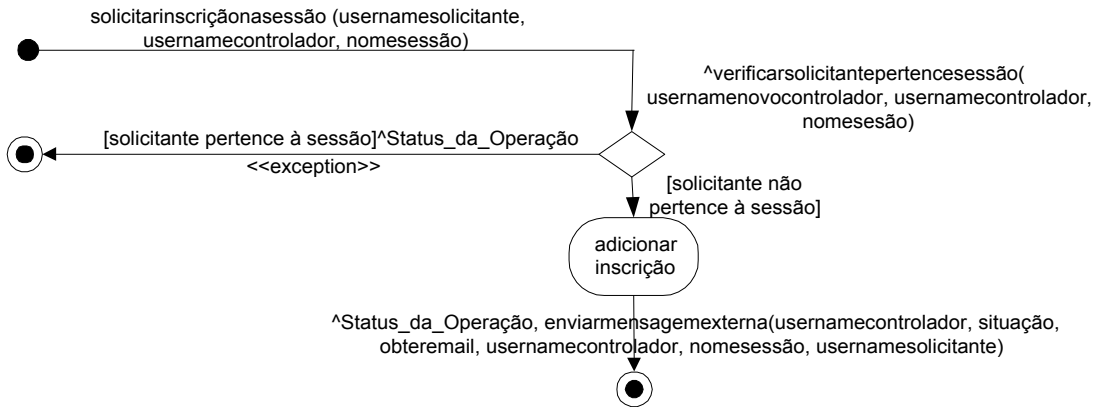






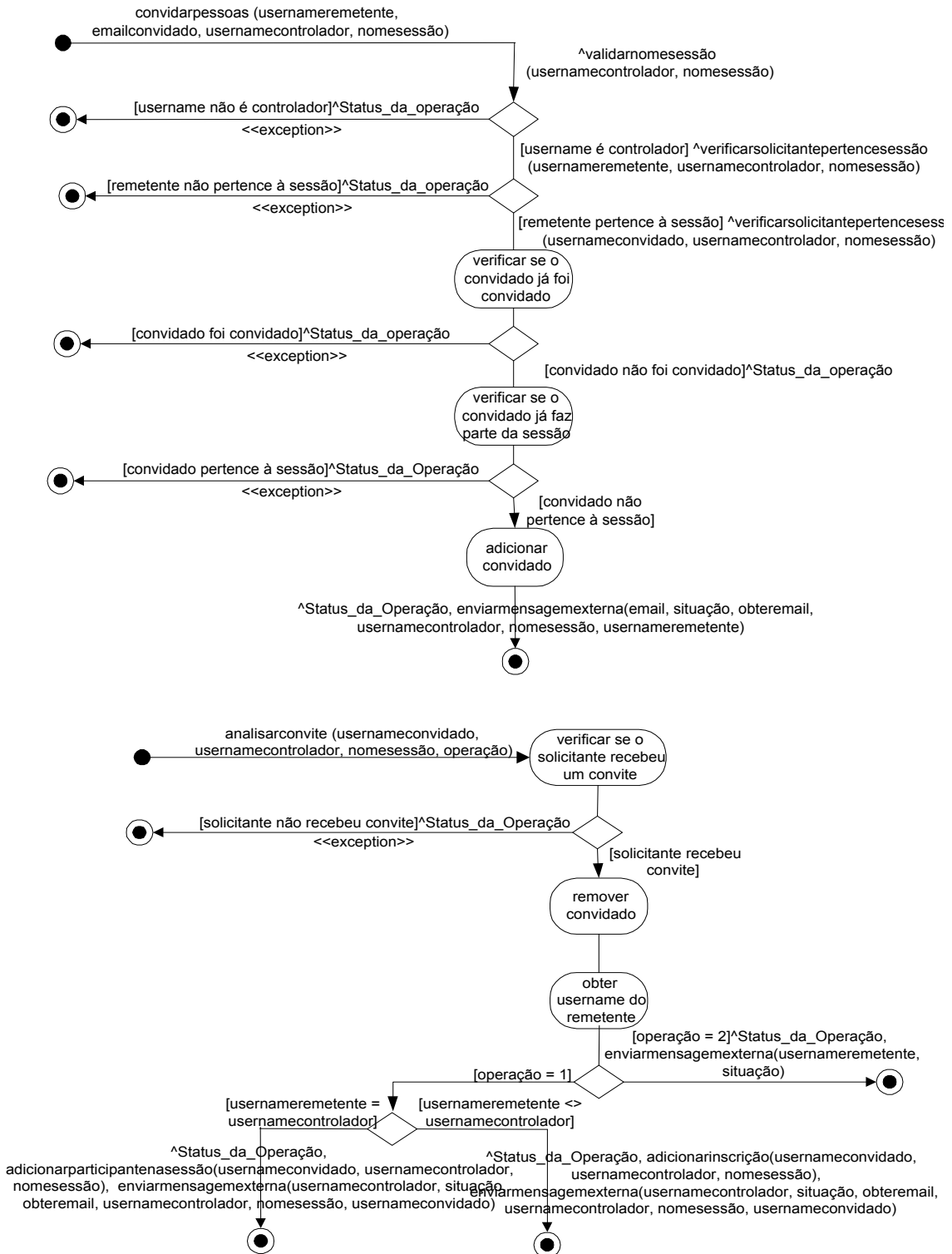
### Gerente de Inscrição

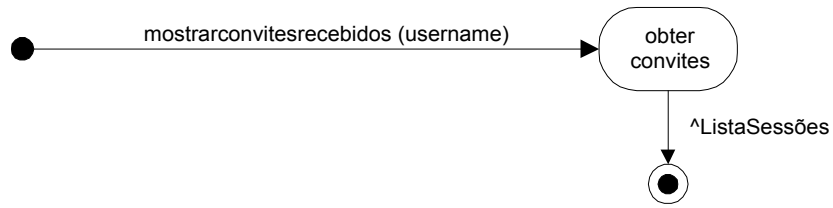
- Solicitar Inscrição na Sessão
- Analisar Inscrição
- Mostrar Inscrições Pendentes



### Gerente de Convite

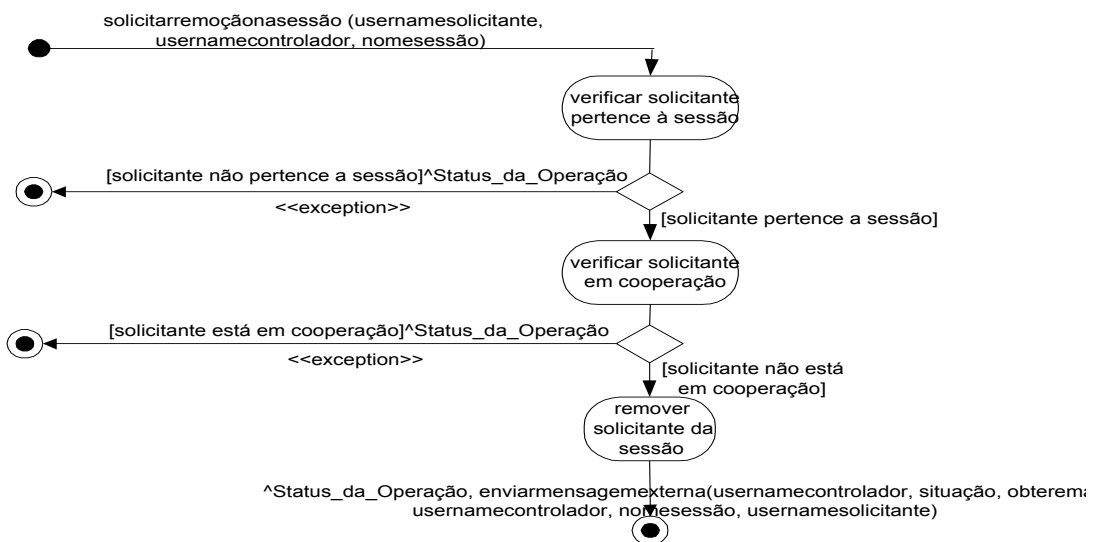
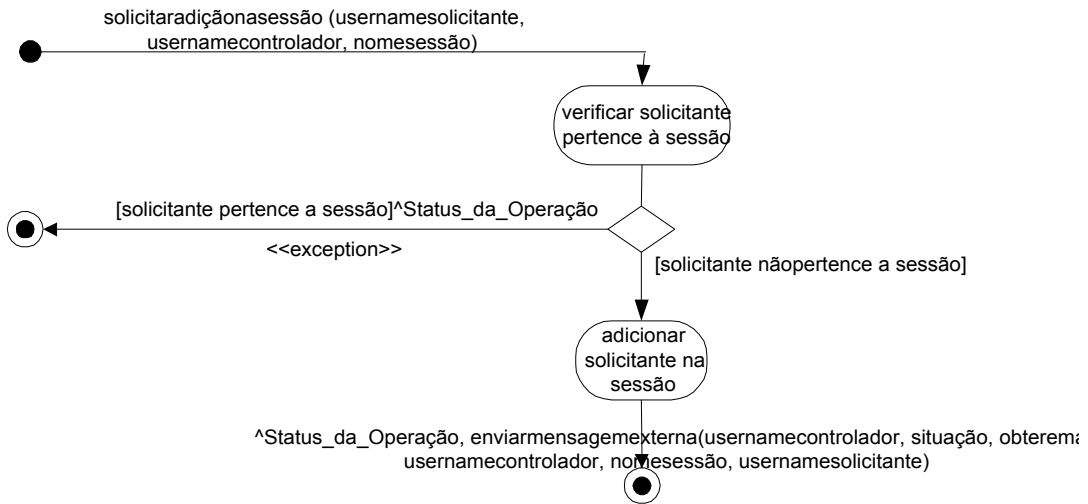
- Convidar Pessoas
- Analisar Convite
- Mostrar Convites Recebidos

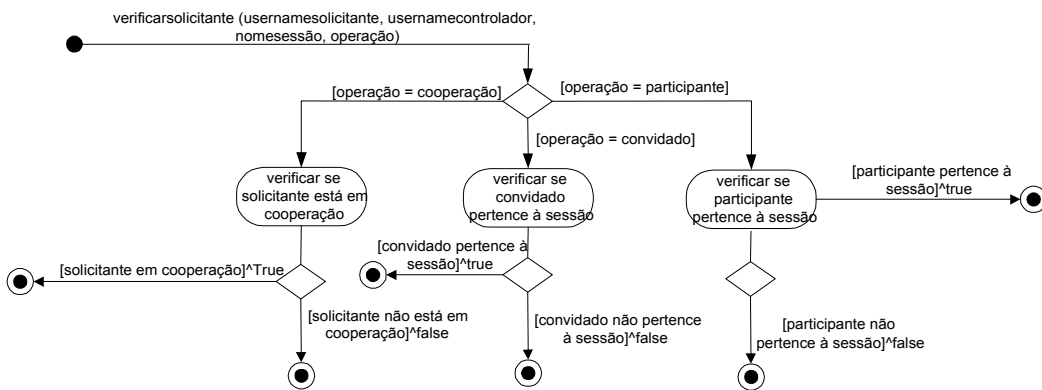
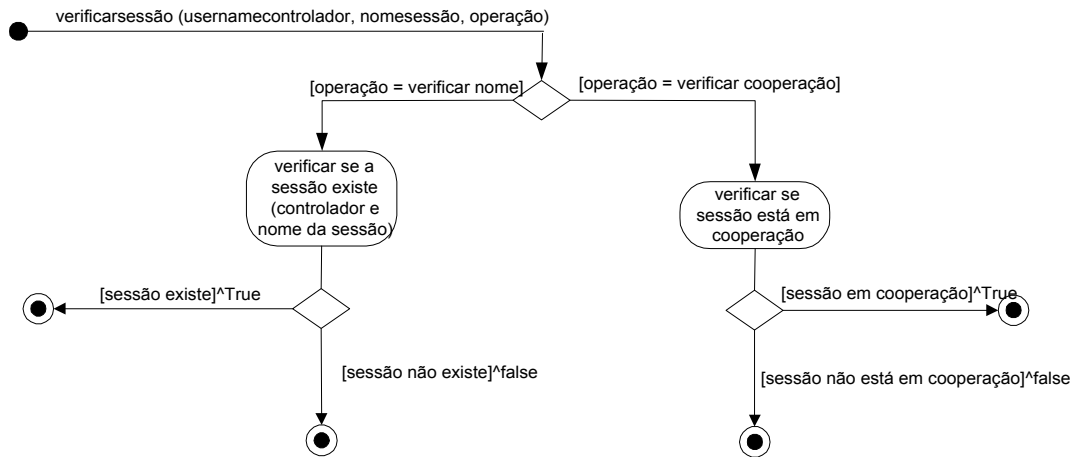




Gerente de Associação

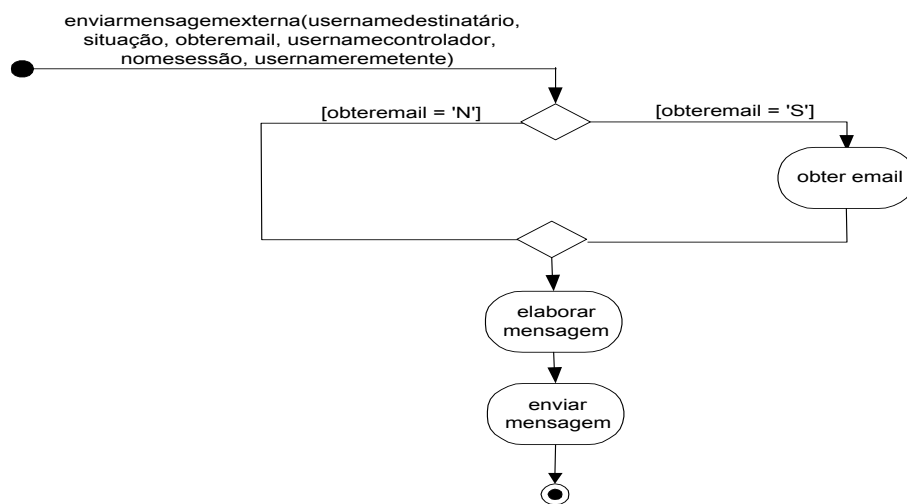
- Solicitar Adição na Sessão
- Solicitar Remoção na Sessão
- Verificar Sessão
- Verificar Solicitante





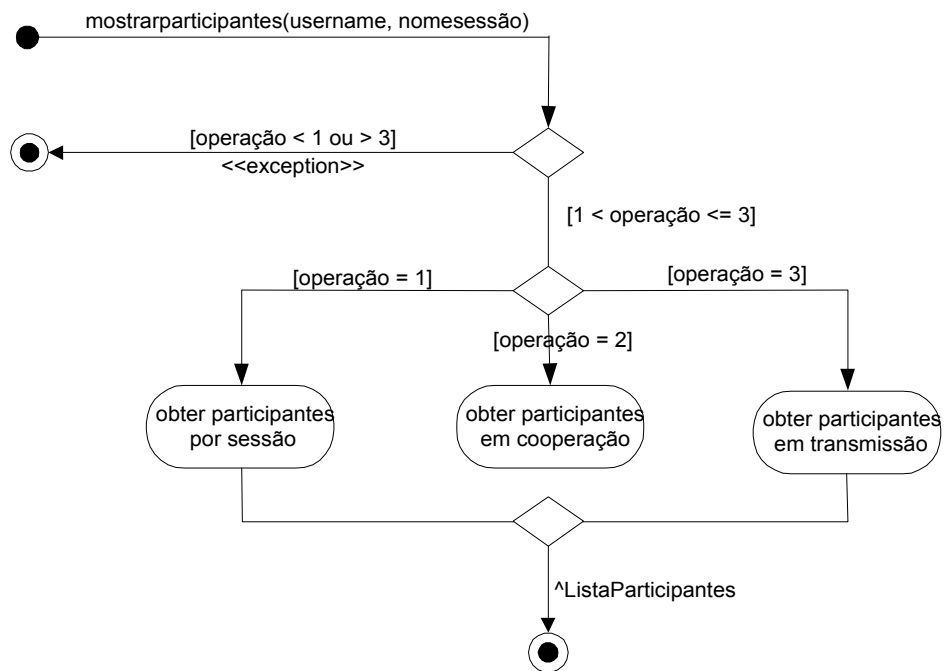
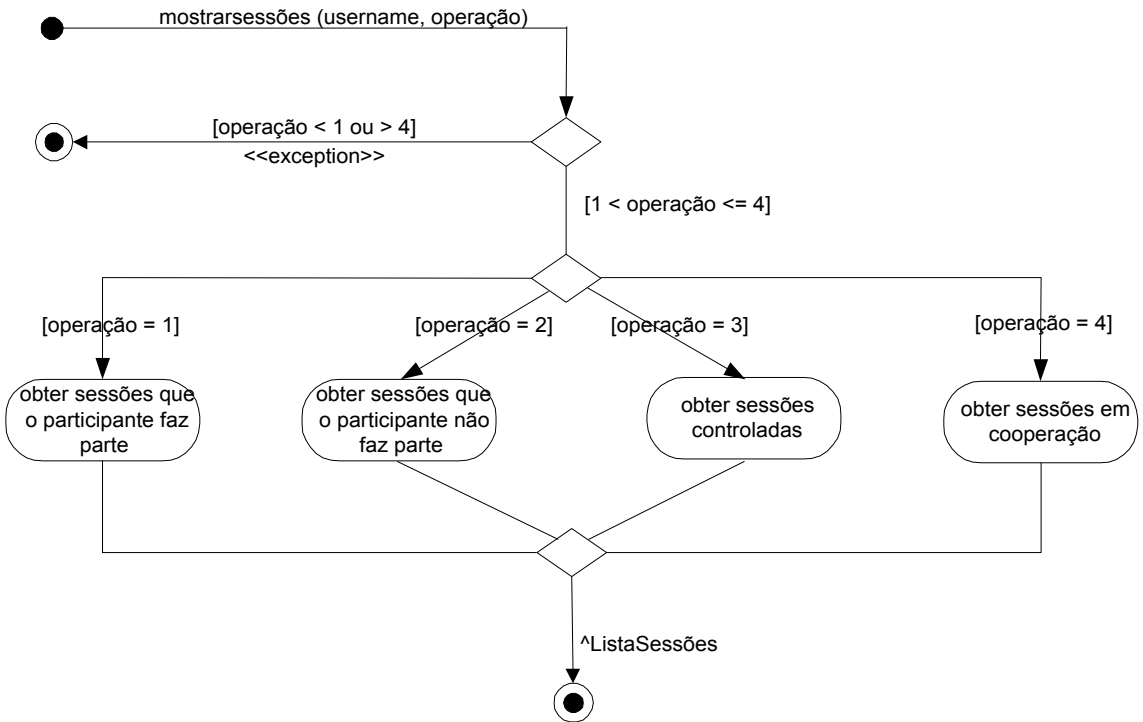
### Gerente de Correio Síncrono

- Enviar Mensagem Externa



### Gerente de Visualização

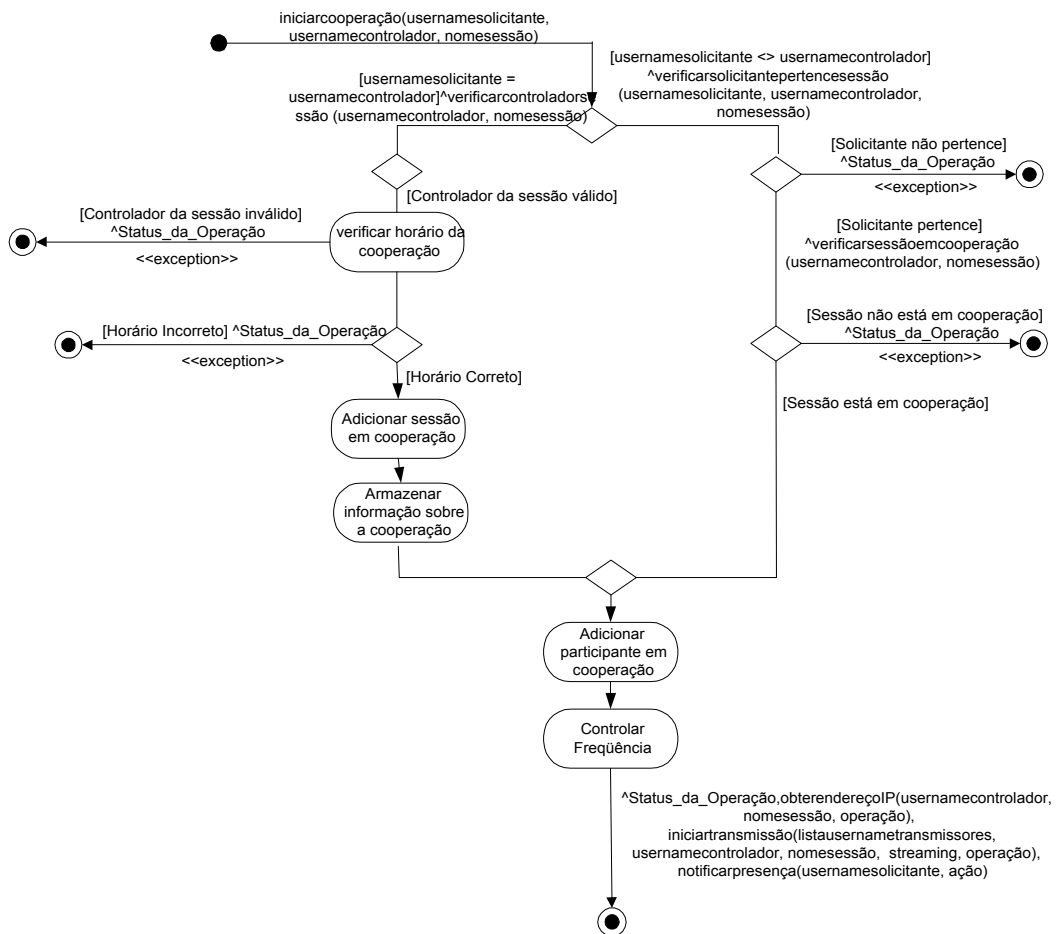
- Mostrar Sessões
- Mostrar Participantes

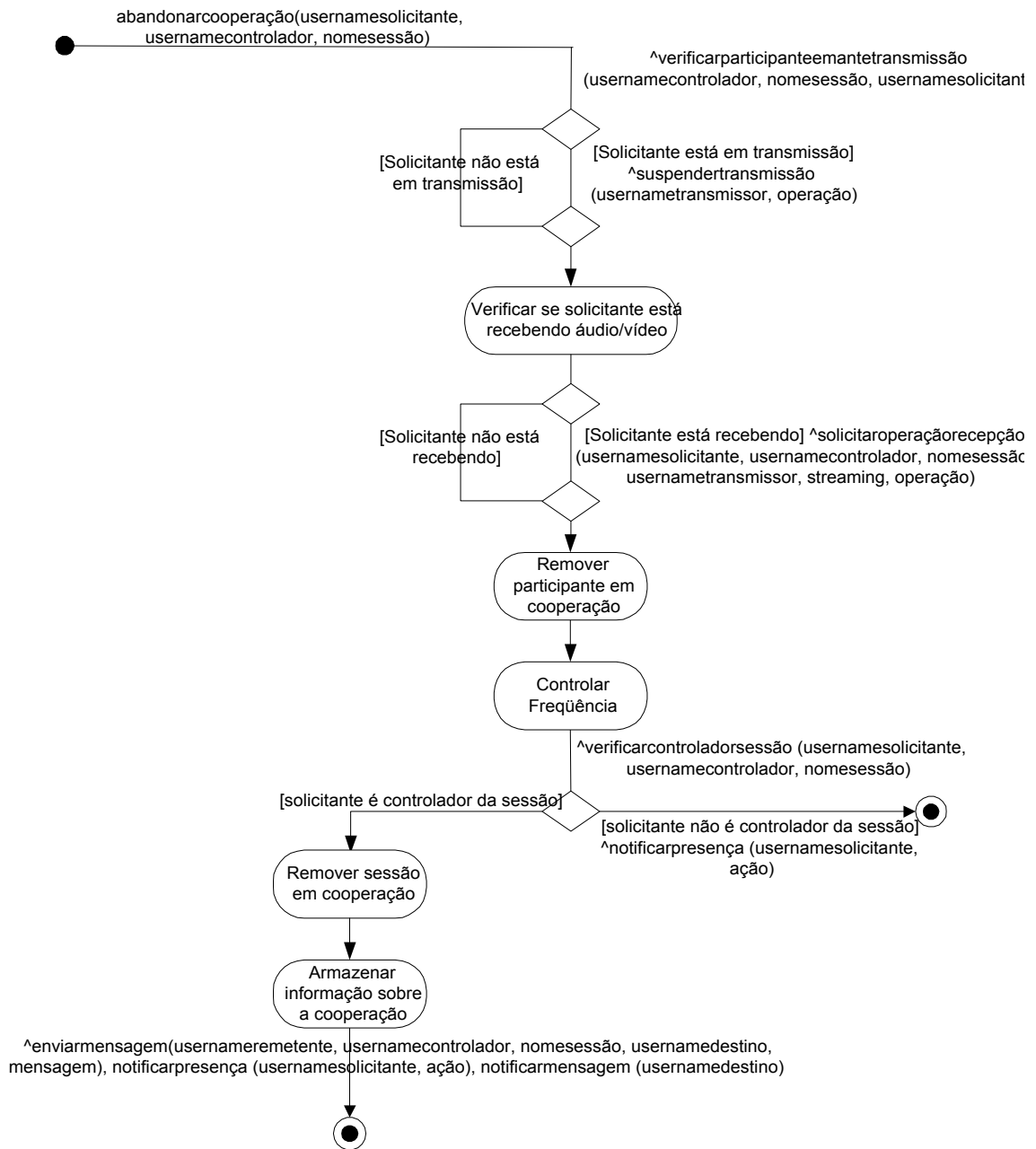


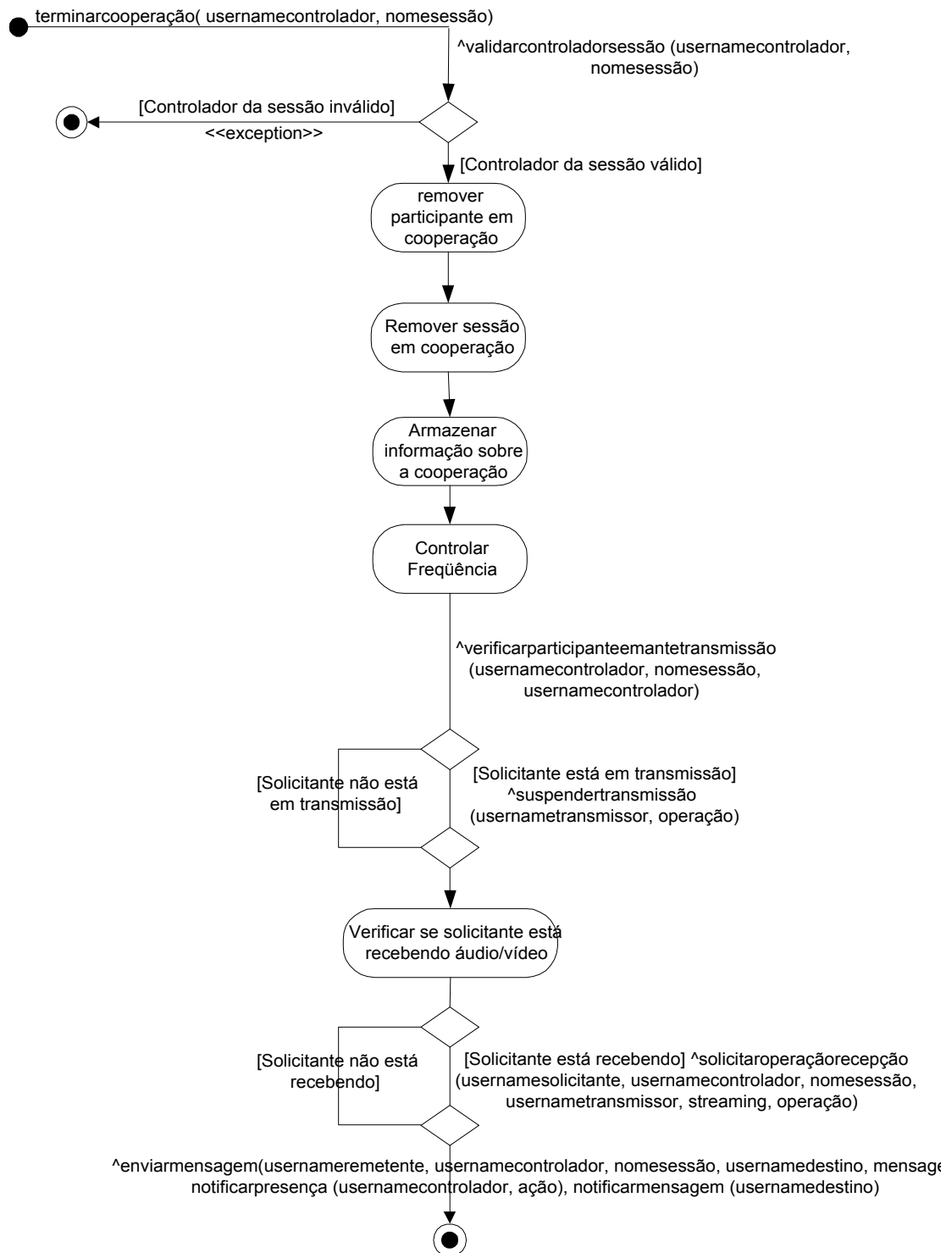


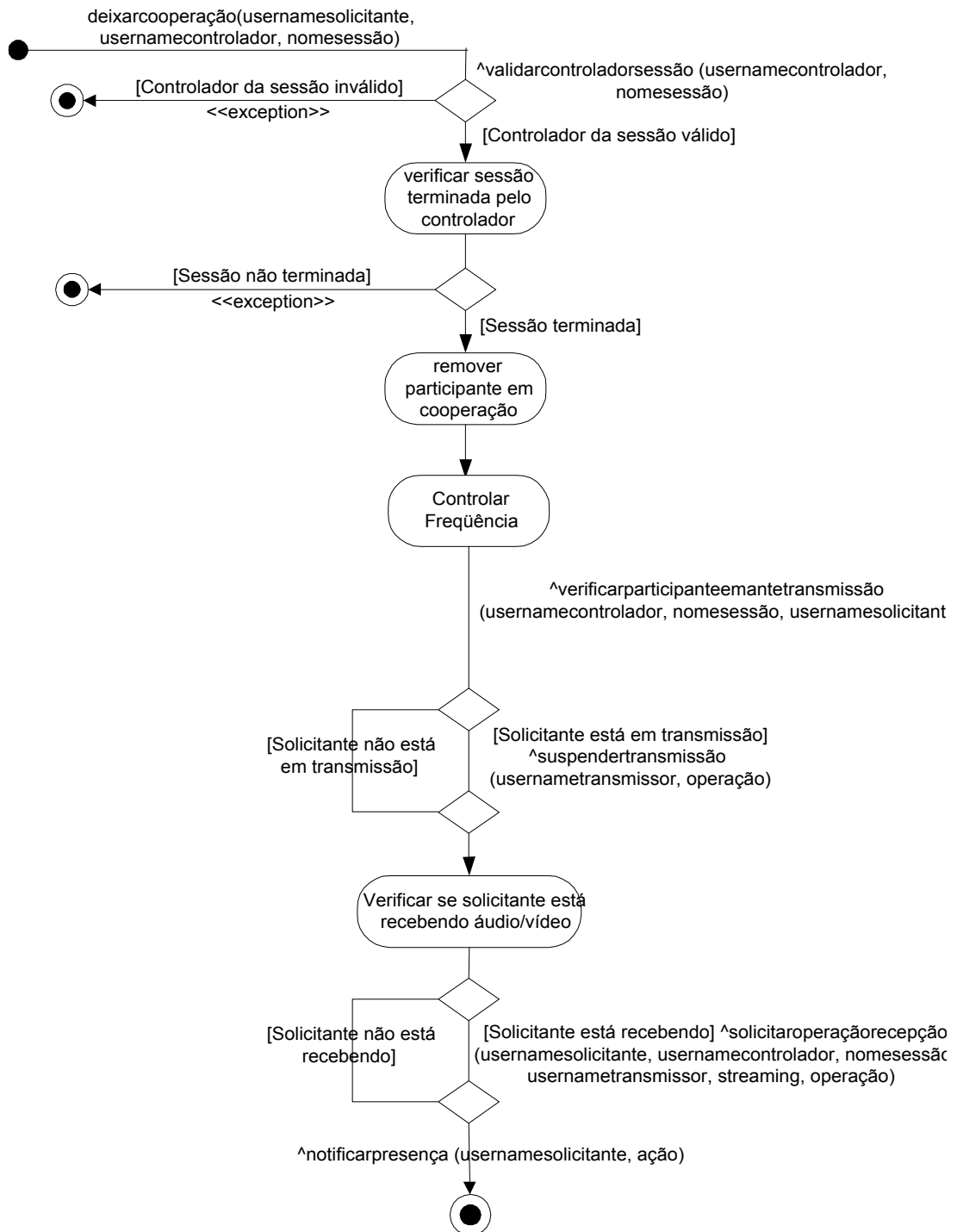
### Gerente de Comunicação

- Iniciar Cooperação
- Abandonar Cooperação
- Terminar Cooperação
- Deixar Cooperação



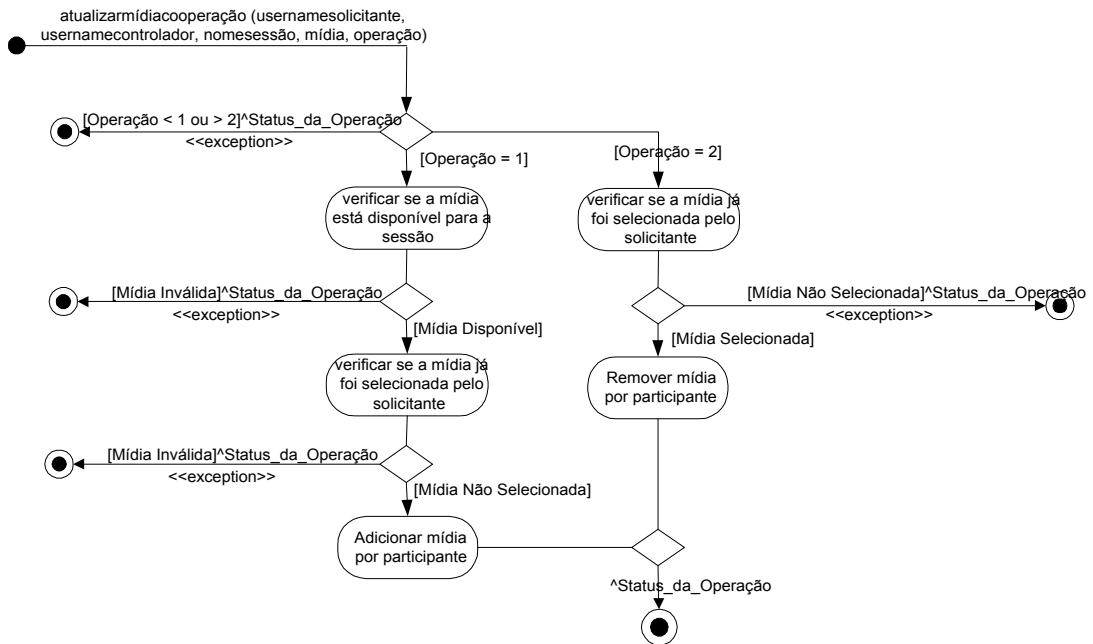






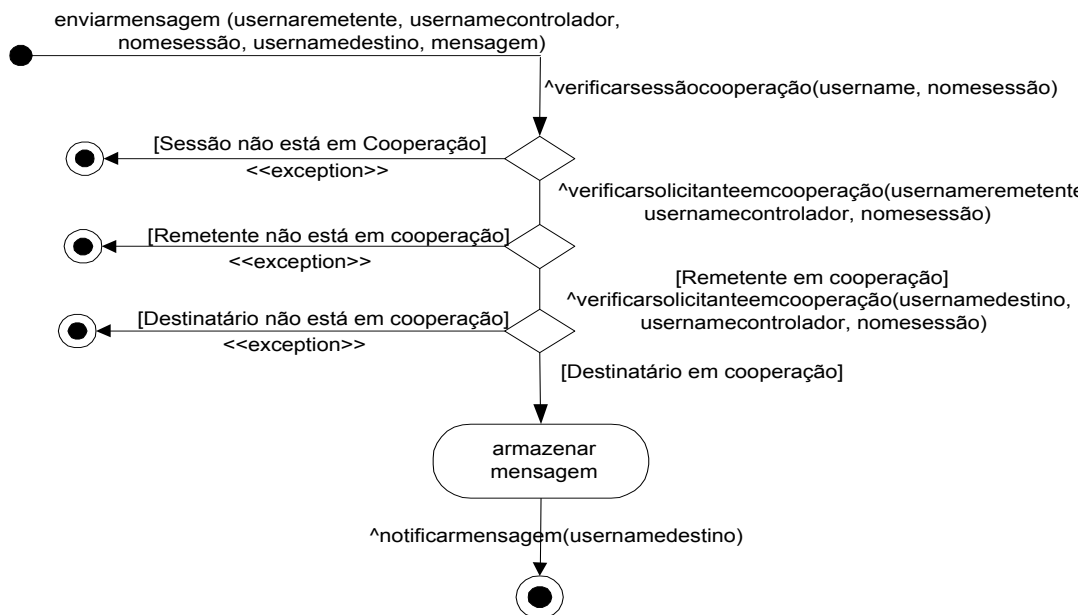
### Gerente de Comunicação

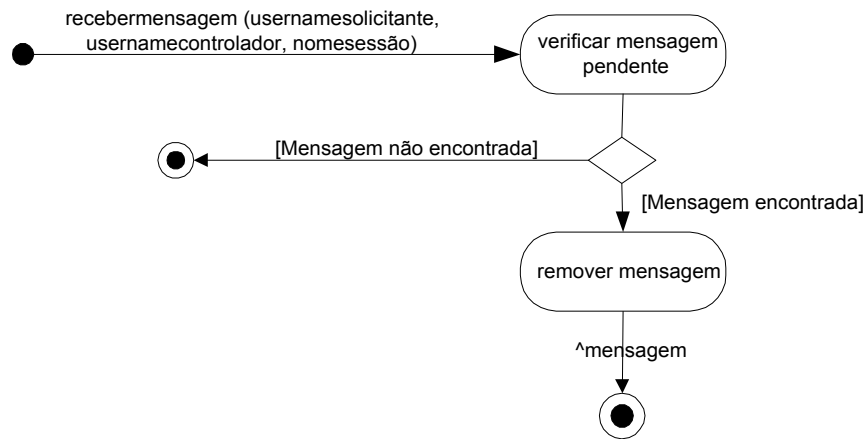
- Atualizar Mídia



### Gerente de Correio Síncrono

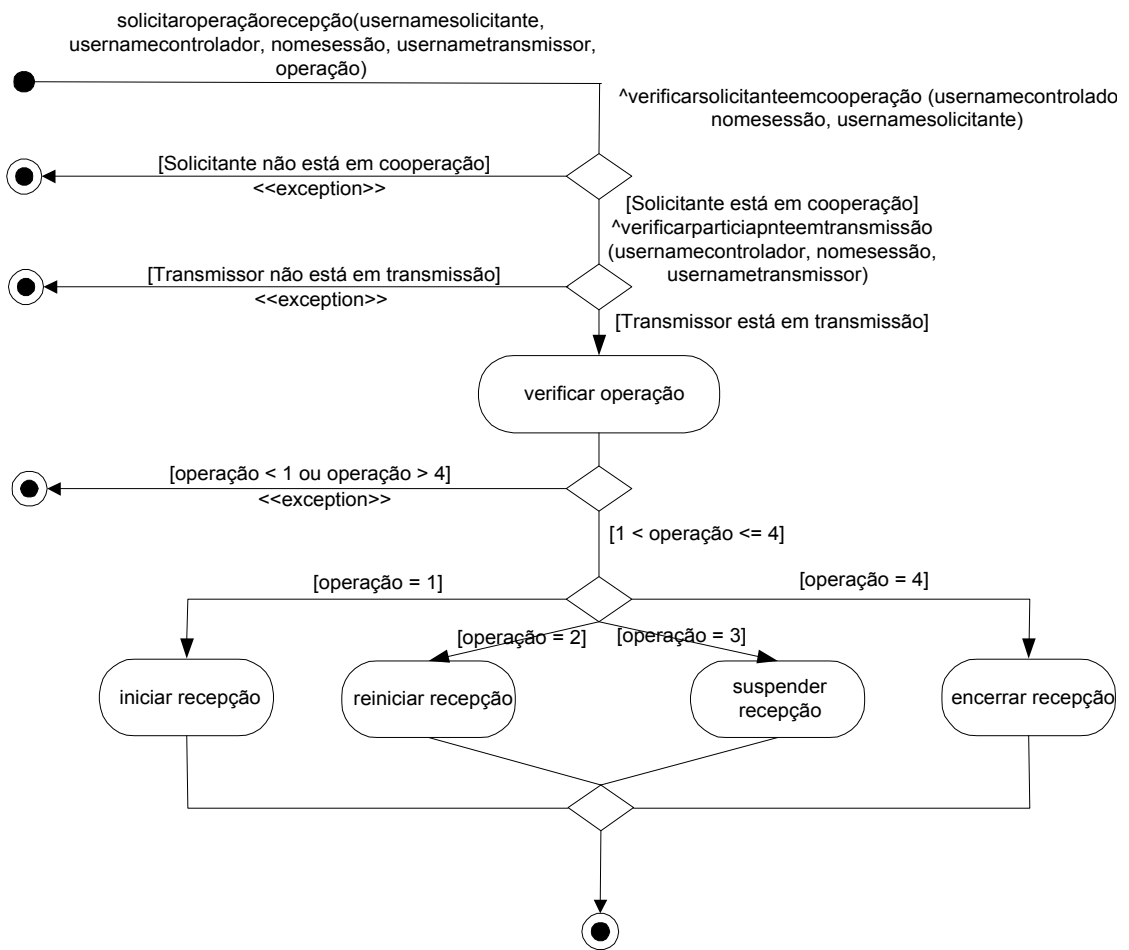
- Enviar Mensagem
- Receber Mensagem





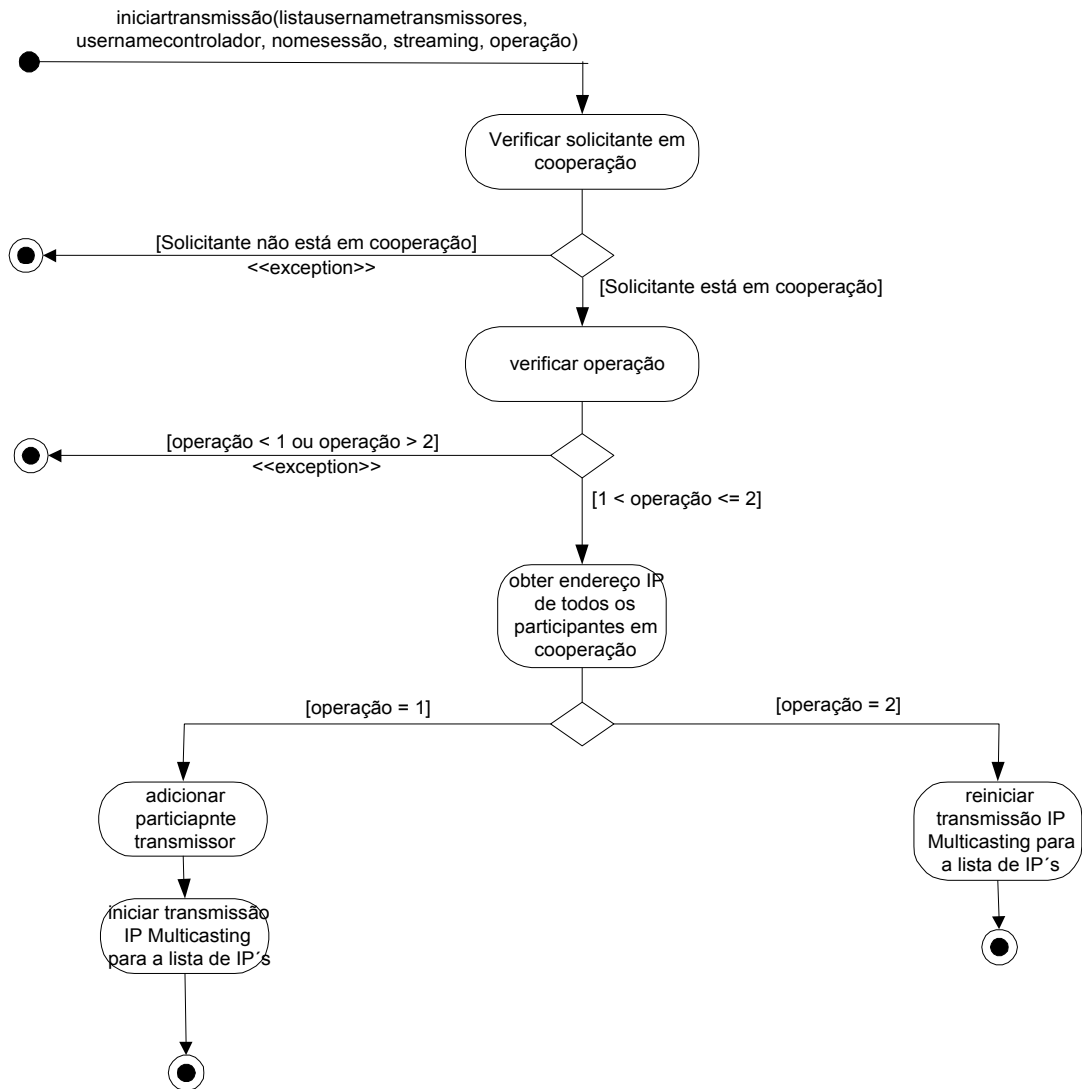
### Gerente de Recepção

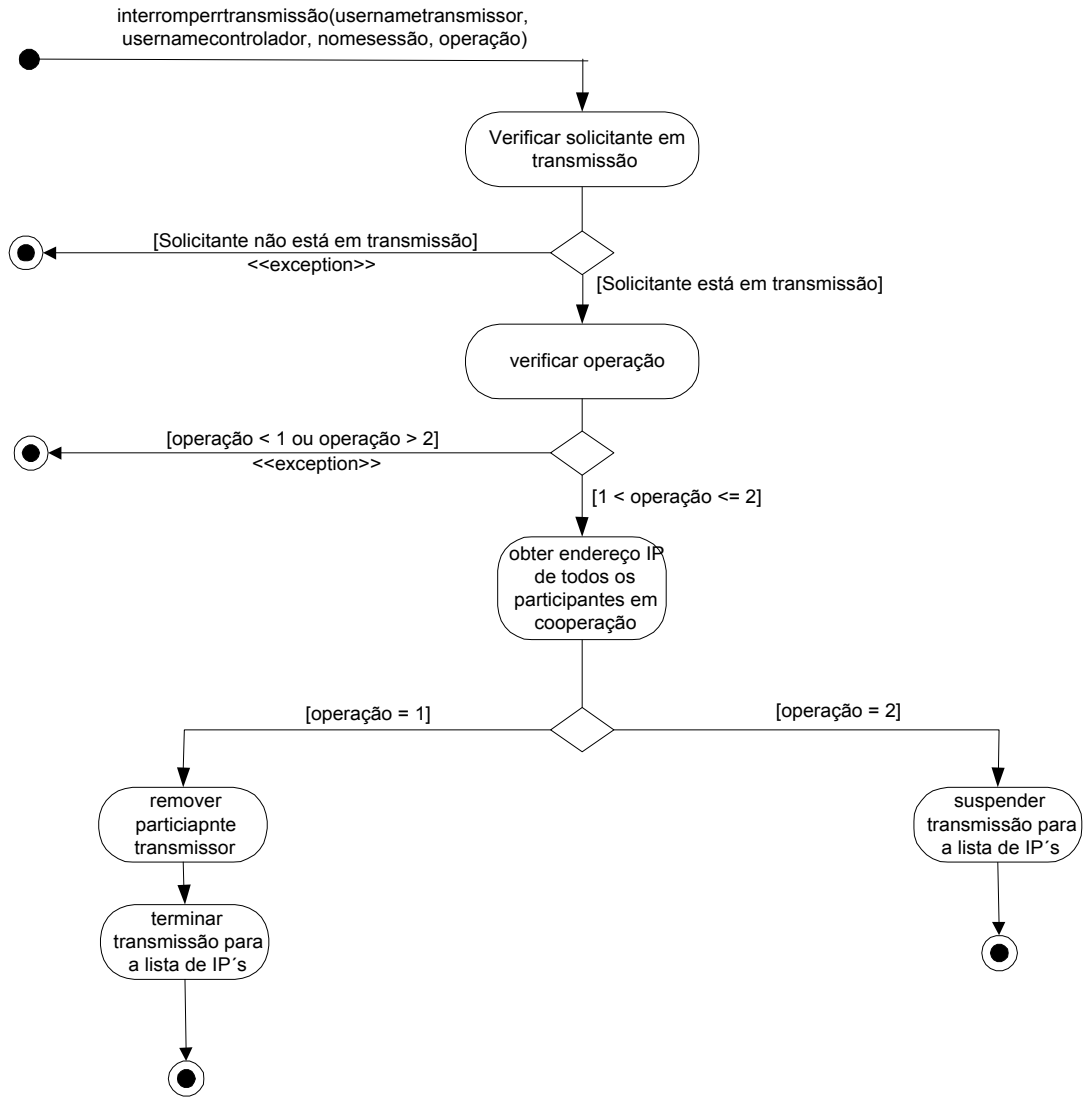
- Solicitar Operação de Recepção



### Gerente de Recepção

- Iniciar Transmissão
- Interromper Transmissão







## Apêndice 5

### Sistema de Conferência Multimídia – Nível de Objeto

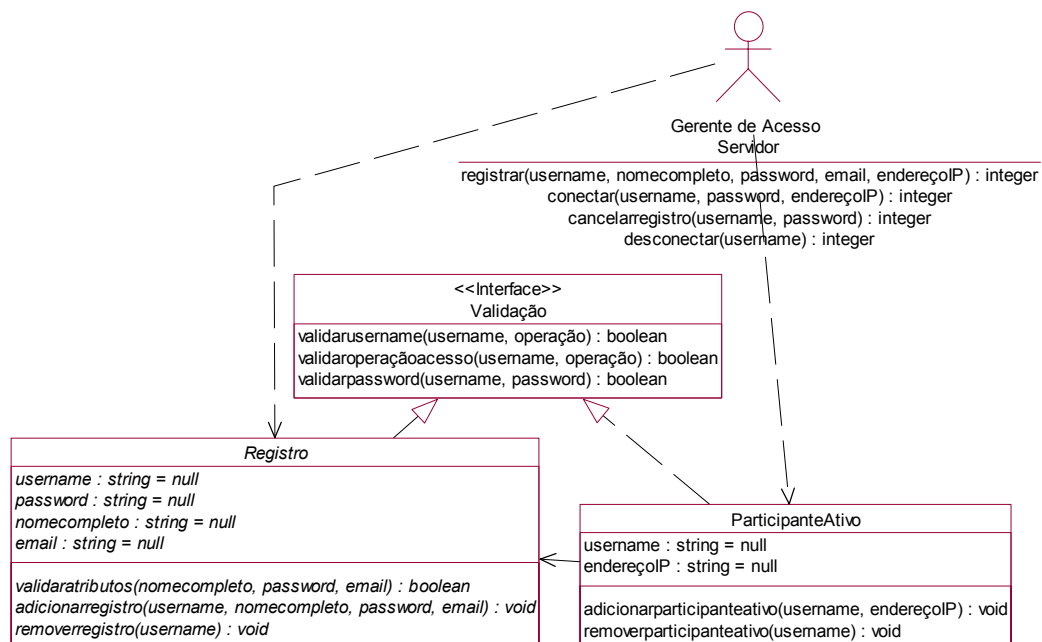
Este apêndice apresenta a modelagem do Sistema de Conferência Multimídia (SCM) para o nível de objeto. O nível de objeto corresponde ao tradicional desenvolvimento de software orientado a objeto.

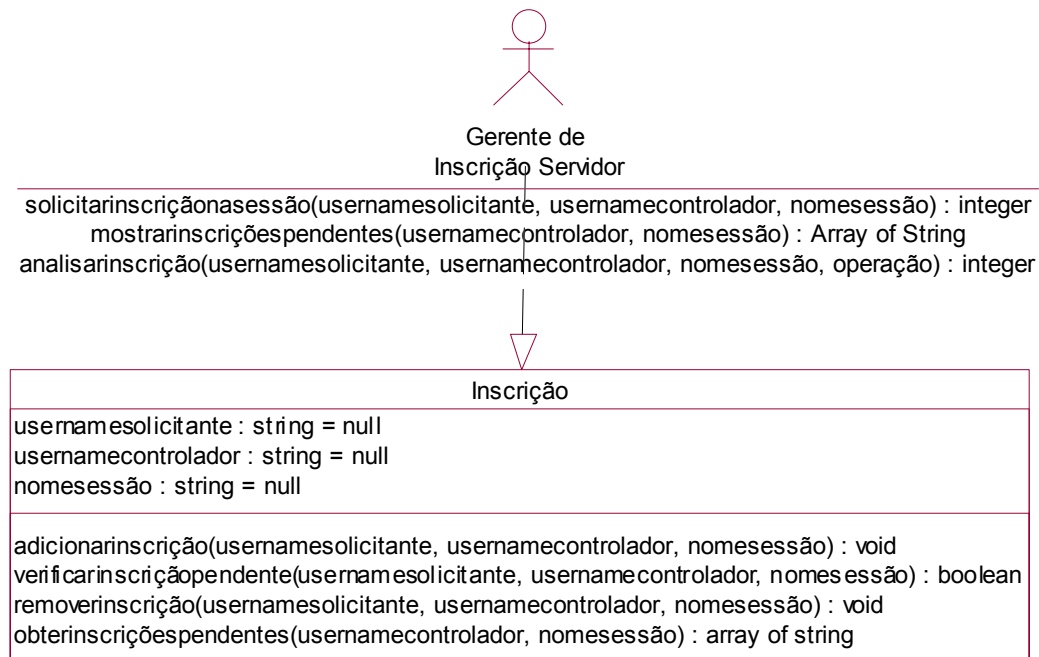
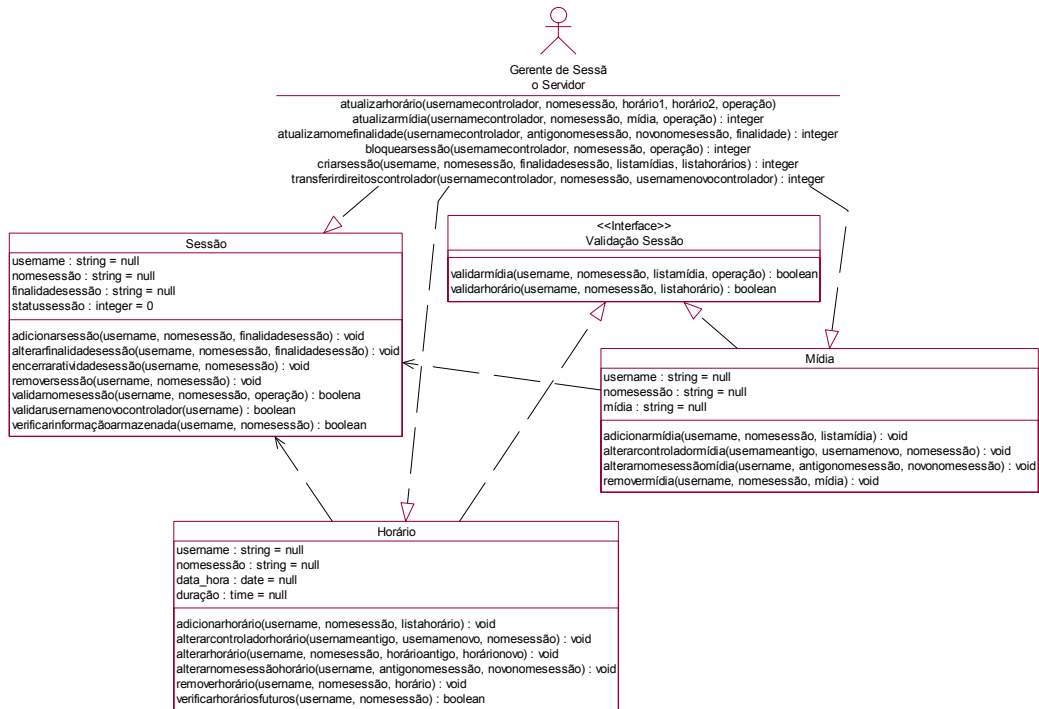
A modelagem deste nível consiste, basicamente, na execução das seguintes etapas:

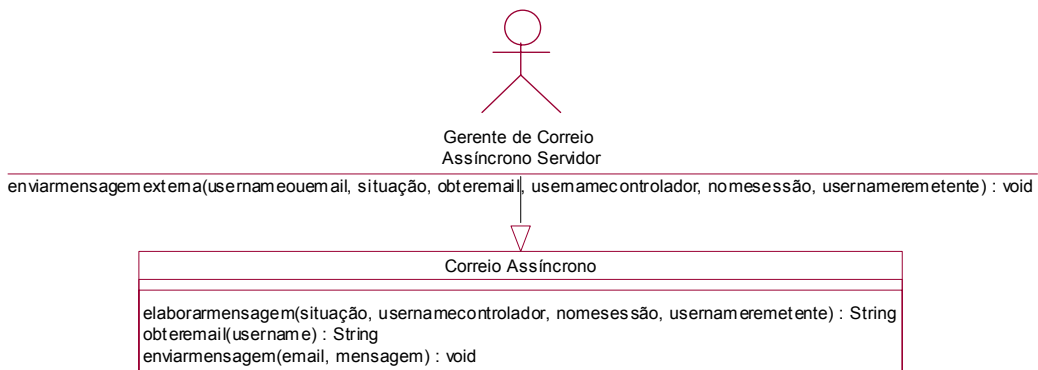
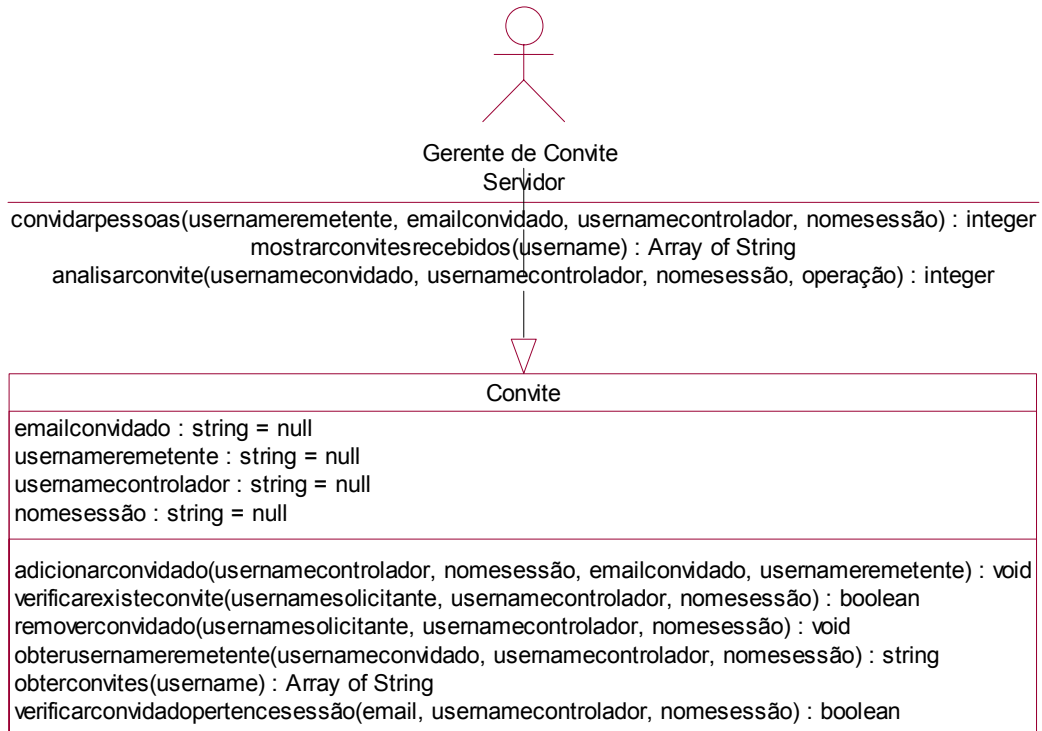
- 1) Visão Estrutural (diagrama de casos de uso, classe ou objeto);
- 2) Visão Interacional (diagrama de seqüência ou colaboração);
- 3) Visão Comportamental (diagrama de atividades ou estado).

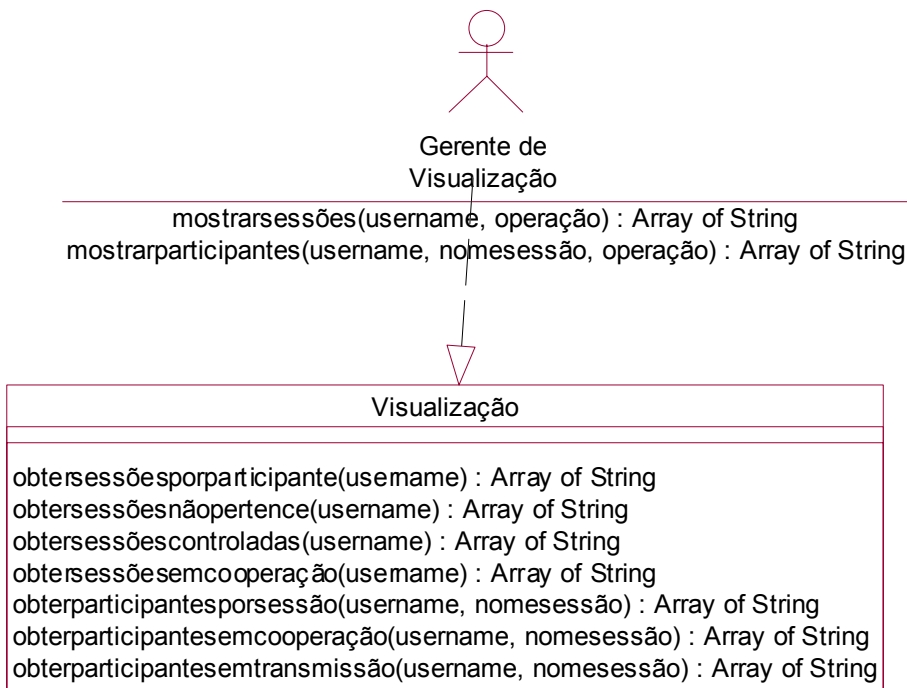
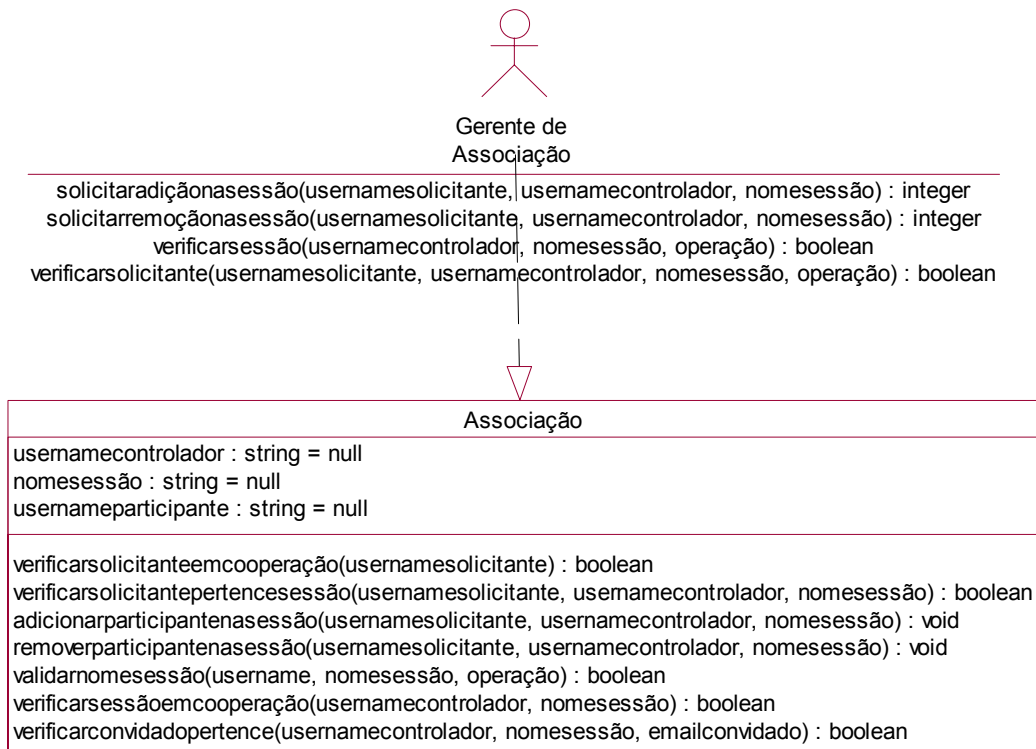
Numa tentativa de não estender em muito este trabalho, somente uma parte da visão estrutural e interacional será apresentada.

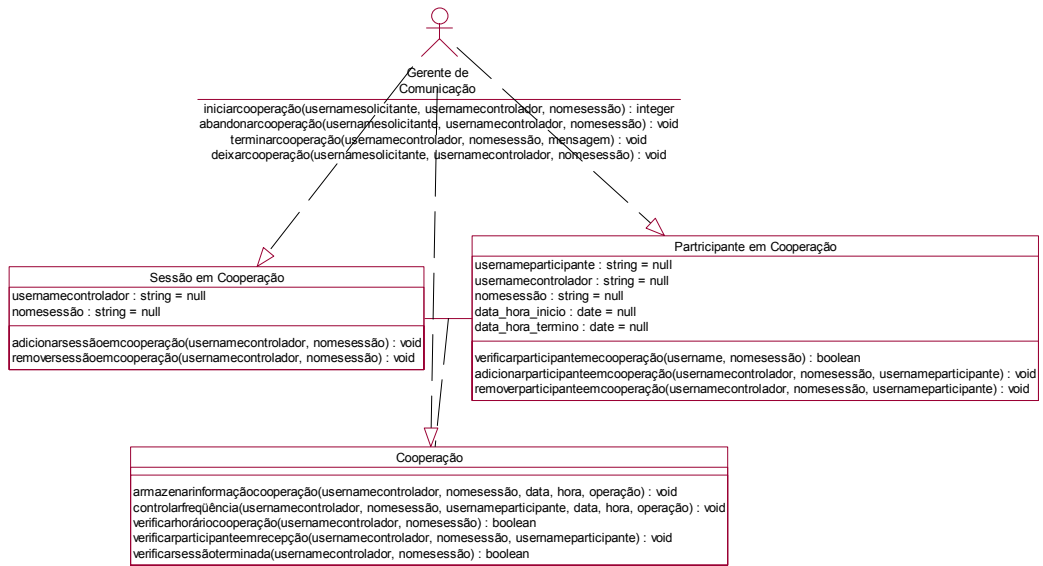
#### A5.1) Visão Estrutural

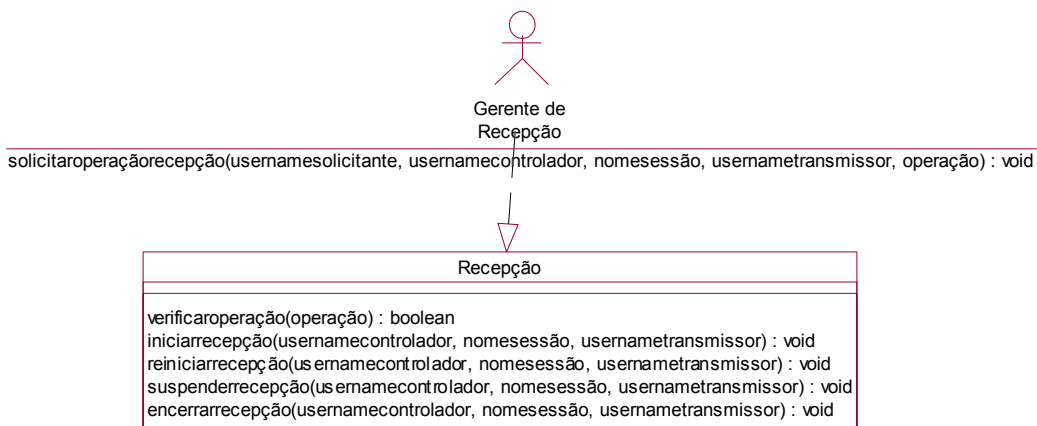
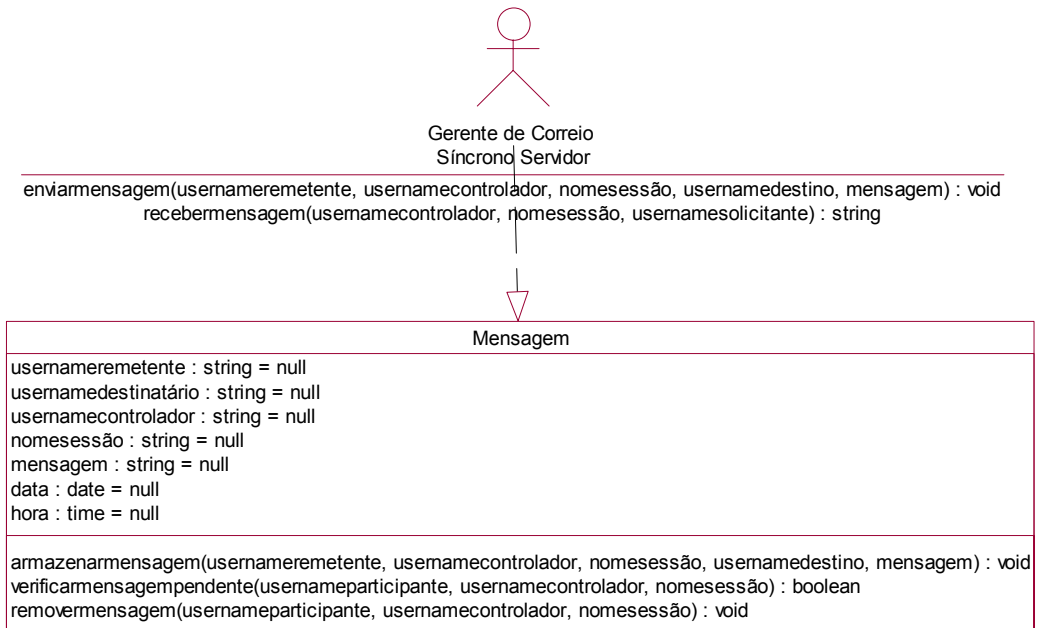


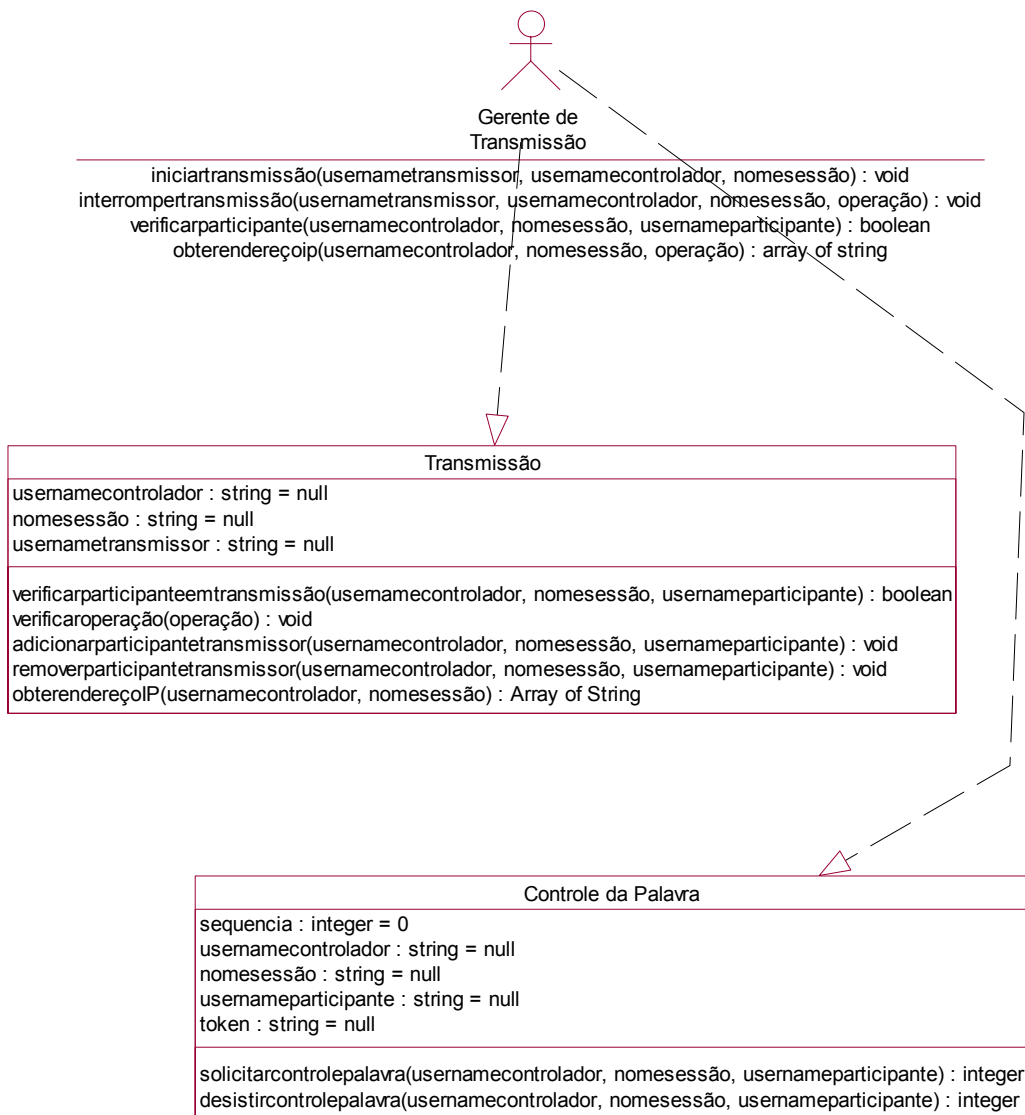


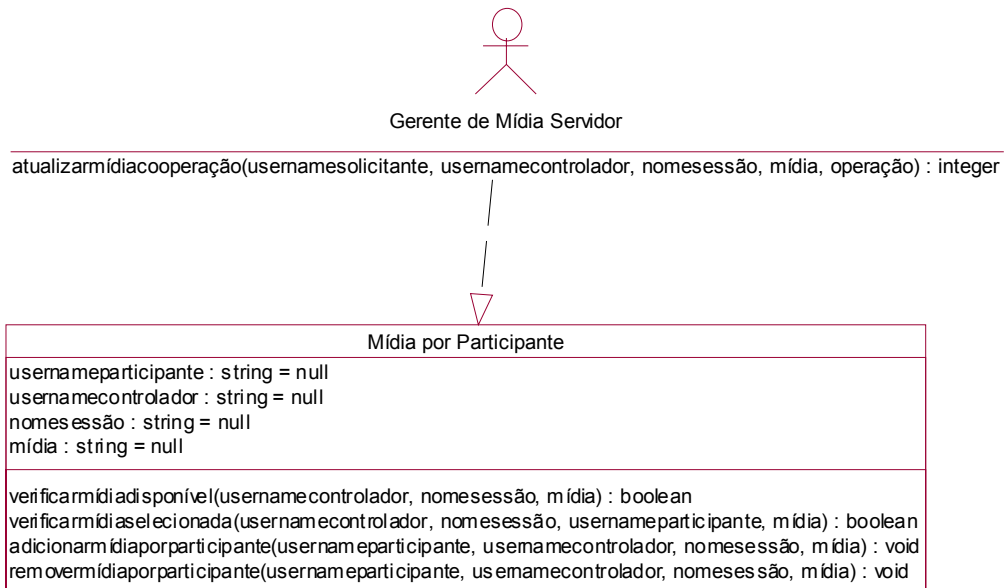












**A5.2) Visão Interacional**

