



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina de Arquitetura e Projeto de Sistemas II**  
**Gabarito – AD1 2º semestre de 2011.**

Nome –

---

Observações:

1. Prova com consulta.

Atenção: Como a avaliação à distância é individual, caso sejam constatadas semelhanças entre provas de alunos distintos, **será atribuída a nota ZERO** a TODAS as provas envolvidas. As soluções para as questões podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser distinta. ALÉM DISSO, às questões desta AD respondidas de maneira muito semelhantes às respostas oriundas dos gabaritos já publicados de ADs e APs de períodos anteriores, **será atribuída a nota ZERO**, incluindo também cópias diretas e sem sentido de tópicos dos slides das aulas.

---

Questão 1 [9 pontos]

Considere a situação em que você tenha que gerar documentação para viabilizar a construção de um portal web para gestão de conteúdos e comunidades, ou seja, gerenciamento de uma rede social composta por *membros* (i.e., estudante, pesquisador ou professor), que podem se comunicar por mensagens pessoais ou públicas (mural) e que são agrupados por laboratórios. Os laboratórios, que representam as *comunidades*, compartilham *conteúdos*, tais como artigos científicos, teses e dissertações, livros, vídeos, imagens e entrevistas em áudio. Cada membro pertence apenas a um laboratório e possui um perfil com seus dados e um conjunto de parceiros de pesquisa de outros laboratórios (membros com os quais ele trabalha). Discussões podem acontecer dentro de cada laboratório na forma de fóruns. No entanto, existe uma comunidade global (do portal) que agrupa todos os laboratórios, onde podem ocorrer discussões gerais. Dessa forma, depois da análise destas informações retiradas de uma entrevista, construa os artefatos a seguir, considerando a Análise e o Projeto Orientados a Objetos:

- a) [1 ponto] Realize a descrição de um sistema para o portal a ser desenvolvido;
- b) [1 ponto] Especifique uma lista de requisitos funcionais e não funcionais que atenda às necessidades do cliente com relação ao sistema a ser desenvolvido;
- c) [1 ponto] Desenhe um diagrama de casos de uso que satisfaça a esta descrição;

- d) [1 ponto] Escolha um dos casos de uso identificados e realize a sua descrição, conforme o *template* da Tabela 1;
- e) [2 pontos] A partir do caso de uso escolhido, construa o(s) Diagrama(s) de Sequência e o(s) Diagrama(s) de Estado correspondentes;
- f) [1 ponto] Construa o Modelo de Classes conceitual (artefato resultante da fase *Análise Orientada a Objetos*), mantendo-o coerente com as entidades utilizadas.
- g) [2 pontos] Explique detalhadamente o que é grau de dependência, bem como suas variações, e relacione este conceito com os tipos de domínios de classes de um sistema. Além disso, calcule o grau de dependência de cada uma das classes apresentadas no Diagrama de Classes conceitual construído em (f).

**Tabela 1 – *Template* para Descrição de Casos de Uso**

<b>Nome:</b>	<definir o nome do caso de uso>
<b>Objetivo:</b>	<descrever o objetivo do caso de uso>
<b>Atores:</b>	<descrever os atores que interagem com o caso de uso>
<b>Pré-condições:</b>	<descrever as pré-condições a serem atendidas para que o caso de uso possa ser executado>
<b>Trigger:</b>	<definir que evento dispara a execução desse caso de uso>
<b>Fluxo Principal:</b>	<descrever o fluxo principal do caso de uso>
<b>Fluxo Alternativo:</b>	<descrever os fluxos alternativos do caso de uso, indicando que evento dispara cada um deles. Cada fluxo deve ser nomeado A1, A2 etc.>
<b>Extensões:</b>	<definir que extensões podem ser executadas>
<b>Pós-condições:</b>	<definir que produto ou resultado concreto o ator principal obterá ao final da execução do fluxo básico>
<b>Regras de negócio:</b>	<listar as regras de negócios que devem ser respeitadas na execução do caso de uso. Cada regra deve ser nomeada RN1, RN2 etc., e ser referenciada em algum fluxo do caso de uso (básico ou alternativo)>

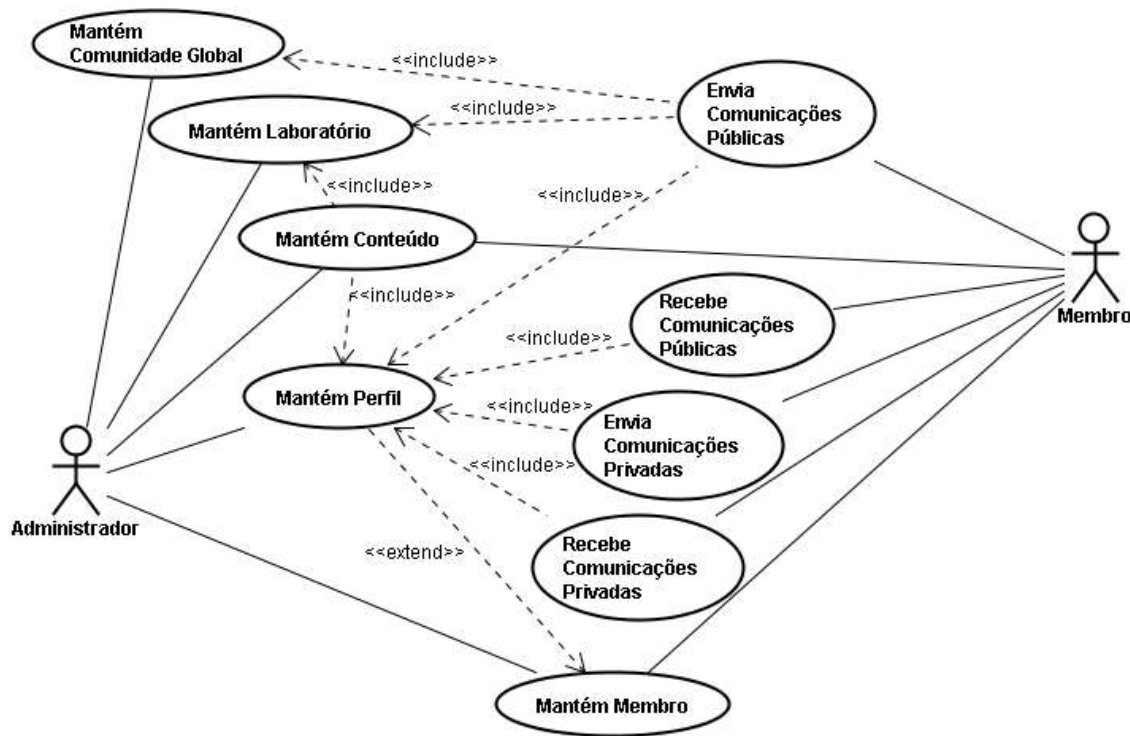
### ***Resposta:***

- a) O Sistema de Gestão de Conteúdos e Comunidades (SGCC) consiste de um portal web que viabiliza, na forma de uma rede social, as comunicações realizadas por um conjunto de laboratórios envolvidos em uma pesquisa, cada um deles descritos por nome, sigla, endereço físico, *website* e telefone. Um laboratório consiste de uma comunidade composta por um conjunto de membros, que são os usuários, por um conjunto de conteúdos e por um espaço para comunicações públicas (mural) entre seus membros. Um membro de um laboratório possui papel de coordenador deste. Um membro pertence apenas a um laboratório e é descrito por nome, *e-mail* (usado para *login*), senha e telefone. Um membro pode ser estudante (com informação de curso, período e se é bolsista ou não), pesquisador (com informação de grupo de pesquisa e se é bolsista ou não) ou professor (com informação de instituição e área). Um membro possui um perfil (página) com foto, mini currículo e *link* para o Currículo Lattes, que apresenta um conjunto de comunicações públicas (mural) e privadas (mensagem) trocadas entre os membros que possuem contato, isto é, parceiros de pesquisa (membros do próprio laboratório e de outros laboratórios). Por sua vez, um conteúdo pertence a um laboratório e pode ser artigo científico, trabalho científico, livro, vídeo, imagem ou entrevista em áudio, cadastrados por um usuário. Todos os membros, através de seus laboratórios, pertencem a uma comunidade global que representa, por exemplo, um projeto de pesquisa de alvo da motivação do SGCC.

b) Lista de requisitos extraídos da Descrição do SGCC:

REQUISITOS FUNCIONAIS	
Nº	DESCRIÇÃO
RF1:	O sistema deve manter um laboratório (comunidade). Por manter, entende-se cadastrar, alterar, consultar ou remover um laboratório.
RF2:	O sistema deve manter um membro (usuário).
RF3:	O sistema deve manter um conteúdo.
RF4:	O sistema deve permitir que um laboratório tenha membros associados, e que um membro exista apenas quando for associado a um, e somente um, laboratório.
RF5:	O sistema deve permitir que um membro de um laboratório publique um conteúdo apenas no laboratório em que está associado.
RF6:	O sistema deve permitir que um, e somente um, membro de um laboratório tenha papel de coordenador deste, e que este laboratório não exista sem um coordenador.
RF7:	O sistema deve manter um perfil para cada membro.
RF8:	O sistema deve permitir que um membro envie e receba comunicações públicas (mural) de membros parceiros de pesquisa através de seu perfil.
RF9:	O sistema deve permitir que um membro envie e receba comunicações privadas (mensagem) de membros parceiros de pesquisa através de seu perfil.
RF10:	O sistema deve permitir que um membro envie e receba comunicações públicas (mural) do laboratório ao qual pertence através de seu perfil.
RF11:	O sistema deve manter uma comunidade global.
RF12:	O sistema deve permitir que um membro envie e receba comunicações públicas (mural) da comunidade global através de seu perfil.
REQUISITOS NÃO FUNCIONAIS	
Nº	DESCRIÇÃO
RNF1:	O sistema deve ter tempo de resposta menor ou igual a 30 segundos.
RNF2:	O sistema deve estar disponível 24 (vinte e quatro) horas por dia, 7 (sete) dias por semana (24x7), com a possibilidade de paradas pré-programadas (que impeçam o acesso pelos usuários do sistema) para manutenção no mesmo ou na infraestrutura de apoio (servidores, sistemas operacionais, SGBD, servidor web etc.), preferencialmente entre 22h e 6h.
RNF3:	O sistema deve realizar backup diário automaticamente entre 22h e 6h.
RNF4:	Nas funcionalidades que demandarem algum tipo de interação entre o sistema e o usuário (ou seja, na interface humano-computador), todas as falhas do sistema devem ser apresentadas seguindo o mesmo padrão de interface do próprio sistema através da exibição de mensagens apropriadas ao erro ocorrido.
RNF5:	A interface humano-computador deve apresentar ícones e textos sem se tornar cansativa aos usuários mais experientes, apresentando módulos de ajuda aos usuários associados às funcionalidades do sistema.

c) Diagrama de Casos de Uso do SGCC:



d) Descrição do Caso de Uso *Mantém Membro*:

<b>Nome:</b>	Mantém Membro
<b>Objetivo:</b>	Realizar cadastro, alteração, consulta ou remoção de membros na base de dados do sistema.
<b>Atores:</b>	Administrador e Membro
<b>Pré-condições:</b>	O Administrador deve fazer <i>login</i> em uma área restrita do sistema.
<b>Trigger:</b>	Uma solicitação para manutenção de um membro é realizada por um candidato a membro, por e-mail ou pessoalmente.
<b>Fluxo Principal:</b>	<ol style="list-style-type: none"> <li>1. O Administrador informa ao sistema o nome do membro e/ou e-mail.</li> <li>2. O sistema verifica a existência do membro.</li> <li>3. O sistema exibe as opções "cadastrar" (apenas no caso em que o membro não esteja cadastrado no sistema), "alterar", "consultar" e "remover" membro.</li> <li>4. O Administrador seleciona uma das opções.</li> </ol>
	<p>Sub-fluxo: Cadastrar membro</p> <ol style="list-style-type: none"> <li>4.1 O Administrador informa ao sistema os dados do membro: <ul style="list-style-type: none"> <li>• Nome completo</li> <li>• E-mail (<i>login</i>)</li> <li>• Senha</li> <li>• Foto</li> <li>• Telefone fixo</li> <li>• Telefone celular – <i>opcional</i>.</li> <li>• Minicurrículo</li> <li>• link para o Currículo Lattes</li> </ul> </li> <li>4.2 O Administrador confirma o cadastro das novas informações na</li> </ol>

base de dados do sistema.

4.3 O sistema gera um código de confirmação do processamento. [a]

Sub-fluxo: Alterar membro

4.1 O Administrador altera os dados desejados, dentre aqueles listados no passo 3 do sub-fluxo *Cadastrar membro*.

4.2 O Administrador confirma a alteração das informações na base de dados do sistema. [a]

Sub-fluxo: Consultar membro

4.1 O sistema exibe os dados do membro.

Sub-fluxo: Remover membro

4.1 O Administrador confirma a remoção das informações na base de dados do sistema.

4.2 O sistema verifica as dependências de sistema relacionadas ao membro. [a]

5. O sistema gera um relatório de atualização.

**Fluxo Alternativo:**

[a] Uma falha acontece no processamento da solicitação

1 O sistema envia um *e-mail* ao Administrador informando a falha.

**Extensões:**

Mantém Perfil.

**Pós-condições:**

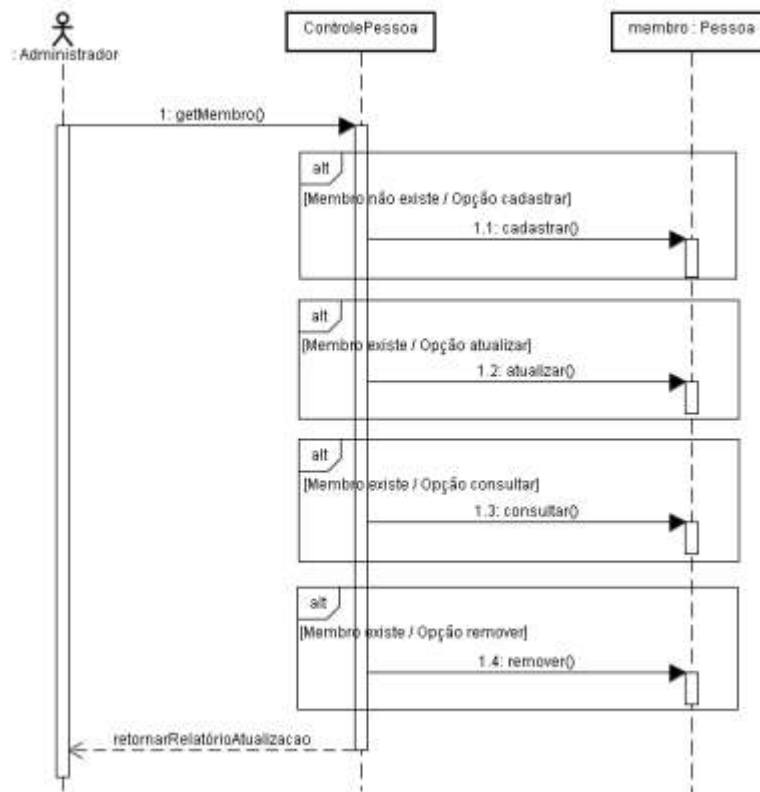
O membro é mantido no sistema com sucesso.

**Regras de negócio:**

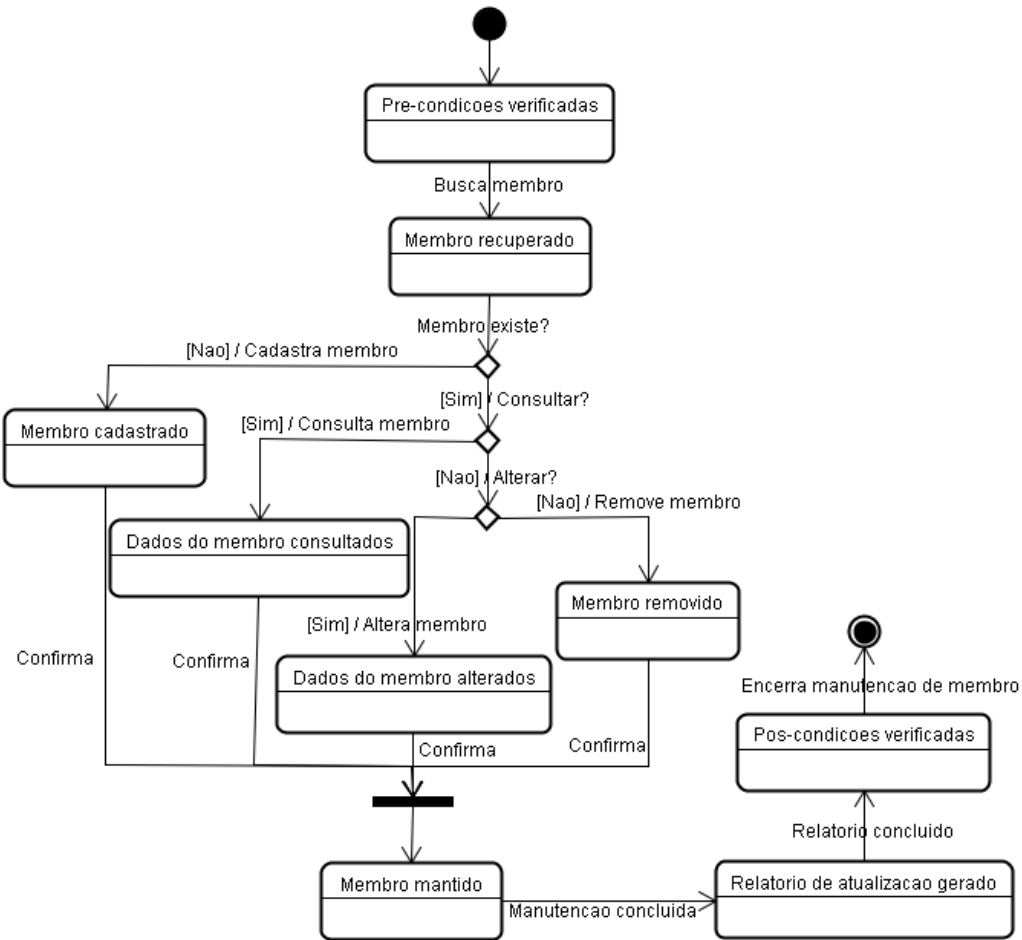
RN01 – O sistema deve ser capaz de realizar a manutenção de um membro na base de dados do sistema.

e) Diagramas do SGCC relativos ao Caso de Uso *Mantém Membro*:

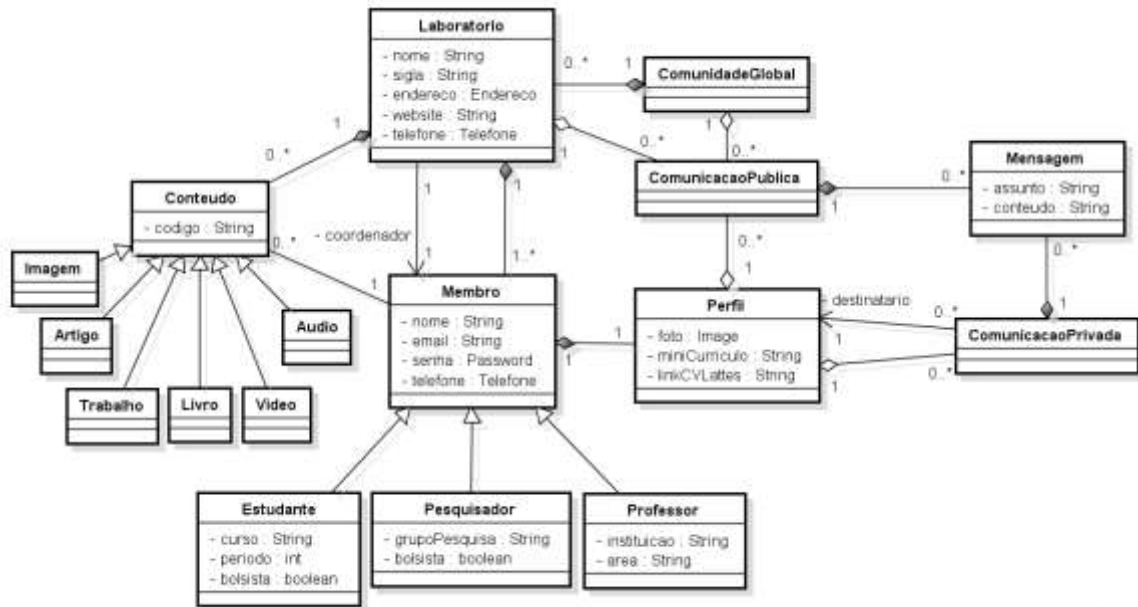
a. Sequência:



b. Transição de Estados:



f) Modelo de Classes Conceitual do SGCC:

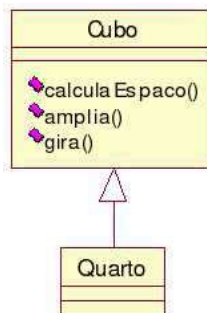


- g) Grau de Dependência (GD) é uma métrica semelhante a *Fan-out* (i.e., indica quantos módulos são acessados por um dado módulo) de projeto estruturado e serve para verificar projetos orientados a objetos. Pode ser classificado em dois tipos: (i) *GD direto*, que indica quantas classes são referenciadas diretamente por uma determinada classe; e (ii) *GD indireto*, que indica quantas classes são referenciadas direta ou indiretamente (recursivamente) por uma determinada classe. Para identificar o GD, deve-se verificar se uma classe *A* referencia diretamente outra classe *B*, de maneira que *A* seja subclasse (ou tenha atributo, ou tenha parâmetro do método do tipo, ou tenha variáveis em métodos do tipo, ou ainda chame métodos que retornem valores do tipo) de *B*. Analisando os diferentes tipos de domínios de classes de um sistema, temos que classes de domínios mais altos (negócio e aplicação) tendem a ter alto GD indireto e as classes de domínios mais baixos (arquitetura e base) tendem a ter baixo GD indireto. Para o cálculo de GD na ocorrência de agregação e composição, estamos considerando a navegação no sentido todo → parte. Cálculo do grau de dependência:

CLASSES	GD Direto	GD Indireto
<i>ComunidadeGlobal</i>	2	7
<i>Laboratorio</i>	3	6
<i>Membro</i>	2	5
<i>Estudante</i>	1	6
<i>Pesquisador</i>	1	6
<i>Professor</i>	1	6
<i>Conteudo</i>	1	5
<i>Imagem</i>	1	6
<i>Artigo</i>	1	6
<i>Trabalho</i>	1	6
<i>Livro</i>	1	6
<i>Video</i>	1	6
<i>Audio</i>	1	6
<i>Perfil</i>	2	6
<i>ComunicacaoPublica</i>	1	1
<i>ComunicacaoPrivada</i>	2	3
<i>Mensagem</i>	0	0

#### Questão 2 [1 ponto]

Para o modelo do projeto orientado a objetos desenhado abaixo, explique qual é o perigo detectado (**ver aula 8**) e associe-o diretamente com a heurística que apoia o projetista a evitá-lo (**ver aula 9**). Após selecionar estes dois itens, **justifique-os detalhadamente**:



**Resposta:**

*Perigo detectado nos modelos:* utilização inadequada da herança (herança forçada).

*Heurística que apoia o projetista a evitar este perigo:* “Se duas ou mais classes compartilham somente características, não deve ser utilizada herança”. *Raciocínio:* herança está fortemente relacionada com o compartilhamento de comportamento (um quarto não deve ter comportamentos como *ampliar* e *girar*) e, neste caso, uma nova classe `Formato3D` contendo as características deve ser criada e associada às demais, conforme ilustrado abaixo:

