

# **MOLIC**

(apêndice)

**Maíra Greco de Paula**

**Simone D.J. Barbosa**

**Rio de Janeiro, maio de 2003.**



## APÊNDICE A – ANÁLISE BASEADA EM CENÁRIOS

Obs.: nos Apêndices A, B e C, são apresentados exemplos baseados no *design* de um sistema interativo de Quadro de Avisos — módulo do projeto ORÉ [ORÉ, 2003]. Em linhas gerais, o Quadro de Avisos permite que membros de uma organização divulguem avisos sobre assuntos diversos. Os avisos são organizados em seções e podem ser divulgados para o público em geral ou somente para os membros da organização, de acordo com as permissões de acesso a cada seção. Além disto, no momento da inserção do aviso, pode-se informar se o aviso deve ser incluído em um quadro geral de avisos ou não.

\*\*\*

Cenários são representados através de narrativas ricas em detalhe contextual, envolvendo usuários, processos e dados que podem ser reais ou potenciais (Figura 1). Apesar de contextualizados, os cenários, quando usados na fase de análise e especificação de requisitos, não devem conter detalhes da interface propriamente dita, como textos e rótulos, seleção de *widgets*, etc. Pretende-se com isto evitar um comprometimento precoce dos *designers* com uma solução de interface a ser adotada, o que dificultaria a exploração de soluções alternativas que emergissem da modelagem de tarefas e do projeto cuidadoso da interação.

### Cenário 3: Consulta ao Quadro de Avisos

Ana Lúcia é colaboradora da Associação Saúde-Criança Renascer há vários anos. Quando tem um tempo livre maior, ajuda em atividades da Organização, trabalhando em alguma oficina, ou fazendo recreação com as crianças. Como está em férias, Ana resolve usar parte de seu tempo para isto. Ela resolve consultar o Quadro de Avisos para ver se estão precisando de algo específico, ou se ela tem alguma idéia diferente. Ana está bastante acostumada com o uso do Quadro. Como seu contato direto é pequeno, ela tem notícias do que está acontecendo a partir dos avisos na Internet...[2] <sup>1</sup> Semanalmente ela dá uma olhadinha para ver o que tem de novo! [6] Mas agora Ana não quer só dar uma espiada. Ela quer saber de que eles podem estar precisando. Em vez de navegar pelas seções do Quadro [7,8], ela resolve buscar os avisos específicos sobre isto [7]. Assim, ela pode conseguir as informações que deseja de forma mais ágil. Ela informa que deseja buscar avisos sobre “doação”, e todos os avisos que contêm esta palavra são mostrados [7]. Ela logo se interessa por um dos pedidos, relacionado a roupas de inverno (“Posso usar meu tempo livre para ajudar minhas amigas a arrumarem seus armários, na condição de doarem aquelas roupas que não estiverem mais usando”, pensa ela).

Figura 1: Cenário *Consulta ao Quadro de Avisos*.

Através dos cenários, pode-se identificar os signos que farão parte da aplicação, desde os signos pertencentes ao domínio até os que surgiram porque as tarefas do usuário estão sendo informatizadas. A seguir apresenta-se uma forma de se organizar os signos que vão sendo identificados durante o processo de *design*, proposta em [Paula, 2003].

### Tabela de Signos

As tarefas descritas nos cenários apresentam ou manipulam informações. As informações apresentadas ao usuário ou modificadas por ele são signos que aparecerão na interface de

<sup>1</sup> As referências no decorrer do cenário (números entre colchetes) serão explicadas mais adiante.

alguma forma. Pode-se classificar estes signos como signos de domínio, transformados ou de aplicação, de acordo com o grau de familiaridade que se espera que os usuários tenham com eles. Signos transportados diretamente do mundo do usuário para a aplicação são representadas por signos do **domínio** (por exemplo, nome e endereço). Signos originados no domínio mas que aparecem na interface através de alguma transformação, tal como analogias ou metáforas, são representados como signos **transformados** (por exemplo, pastas na área de trabalho do Windows). Por último, signos que só fazem sentido dentro do sistema, e não têm prévio significado para os usuários, são chamados de signos da **aplicação** (por exemplo, login e senha).

É importante classificar os signos em tipos (domínio, transformado ou aplicação), pois tipos diferentes requerem diferentes tomadas de decisão por parte do *designer*. Em geral, signos transportados diretamente do mundo dos usuários (de domínio) devem apenas requer informações se limitações forem impostas pela aplicação. Por exemplo, o signo *nome* é claro para os usuários, mas pode requerer a seguinte explicação – “Por favor, forneça seu nome completo. Não é permitido mais que 100 caracteres.”. Signos que são transformações de signos existentes no mundo dos usuários (transformados), requerem informações sobre os limites da analogia ou metáfora realizada para transportá-los para a aplicação. Por exemplo, uma explicação sobre as pastas em um *desktop* seria “Estas pastas funcionam como no mundo real, exceto que nunca ficam cheias. Isto é, você pode manter e colocar muitas coisas nelas. Na realidade, o disco do seu computador é o real local onde as coisas estão sendo guardadas. Então, é ele que controla a quantidade de espaço que pode ser ocupado.” Finalmente, signos existentes apenas na aplicação, que podem ser totalmente desconhecidos pelos usuários, requerem uma explicação completa sobre o que significam e como são utilizados. Por exemplo, o signo *zoom* em uma aplicação gráfica.

Alguns signos podem gerar dúvidas no momento de classificação em um dos tipos. Por exemplo, o signo senha pode ser classificado em aplicação, mas pode aparecer a seguinte dúvida “Não se pode considerar o signo senha nesta aplicação como uma analogia à assinatura, impressão digital ou algo que identifica uma única pessoa? Se sim, então também se pode classificá-lo como signo transformado.” Neste caso, fica a cargo do *designer* definir o tipo do signo, baseado nas características dos usuários e suas tarefas, e a quantidade de informação a ser fornecida. Em todo caso, a classificação auxilia o *designer* a refletir sobre a explicação a ser associada a cada signo.

A Figura 2 apresenta o cenário Pedido de Inscrição no Quadro de Avisos, dando destaque aos signos extraídos deste cenário.

**Cenário 1: Pedido de Inscrição no Quadro de Avisos**

Rita acaba de ser contratada pelo Renascer como funcionária, para trabalhar no atendimento às famílias. Dentre as instruções necessárias para o seu trabalho, Rita é avisada que há um Quadro de Avisos virtual que todos os membros da organização estão começando a utilizar [1]. É uma ótima referência para saber o que está acontecendo, quais os próximos eventos, etc [2,3]. Rita explica que já conhece o Quadro, o qual havia acessado como visitante quando lhe ofereceram o emprego [1]. Então lhe explicaram que, como visitante, ela só poderia acessar os avisos públicos, e agora deveria se inscrever para ter acesso às seções destinadas aos funcionários [5,9,10,11,12]. Rita entra no Quadro e, partindo das instruções que encontra na página, fornece os dados necessários para o pedido de inscrição: seu nome completo, e-mail, e login e senha que deseja utilizar [15,16]. Após finalizar o pedido, novas instruções são apresentadas a Rita, dizendo que agora ela precisa esperar que o responsável pelo Quadro cheque seus dados e efetue sua inscrição [17,18]. Ela não entende direito como saberá quando sua inscrição será efetivada, e ao perguntar sobre isto ao coordenador da sua área, ele lhe diz que ela receberá uma mensagem através do email que forneceu [19].

Figura 2: Cenário Pedido de Inscrição no Quadro de Avisos, com os signos em destaque.

Examinando todos os signos extraídos dos cenários, pode-se perceber que alguns destes signos podem ser agrupados e relacionados a conceitos ou entidades do domínio e/ou da própria aplicação. Estas entidades são representadas como signos compostos. Por exemplo, nome, e-mail, login e senha podem ser agrupados no signo composto usuário.

Os signos podem ser representados em uma tabela contendo campos como nome do signo, descrição e tipo (composto, domínio, transformado ou aplicação), como pode ser visto na Tabela 1. Esta tabela de signos pode ser completada à medida que a modelagem for sendo feita e novos signos forem surgindo.

<b>Signo composto</b>	Usuário	
<b>Nome do signo</b>	<b>Descrição</b>	<b>Tipo</b>
nome	nome do usuário	Domínio
e-mail	e-mail do usuário	Domínio
login	identificação do usuário	Aplicação
senha	senha para conferência da identificação do usuário	Aplicação
<b>Signo Composto</b>	Aviso	
<b>Nome do signo</b>	<b>Definição</b>	<b>Tipo</b>
título	título do aviso	Domínio
autor	autor do aviso	Domínio
data	data de postagem do aviso	Domínio
chamada	texto do aviso (resumido)	Domínio
texto	detalhes do texto do aviso	Domínio
quem_postou	membro que postou o aviso	Aplicação
prazo	prazo de validade da exposição do aviso	Domínio
quadro_geral	indicação se o aviso deve ser apresentado no quadro geral de avisos	Domínio

Tabela 1: Parte da tabela de signos do Quadro de Avisos.

Algumas críticas ao uso de cenários se referem à frequência com que ficam incompletos ou ambíguos. Para tentar solucionar estes problemas, [Paula, 2003] propõe complementar os cenários com perguntas que revelem os intuítos do *designer* ao elaborá-los, isto é, perguntas que identifiquem os pontos que o *designer* almeja descobrir, explorar e/ou ratificar junto aos usuários ao se construir o cenário. Além de apoiar a reflexão do *designer*, estas perguntas podem evitar que os cenários fiquem incompletos ou ambíguos, ou até mesmo revelar novos elementos nos cenários.

O *designer* pode gerar uma lista global de perguntas que seriam referenciadas nos cenários gerados (Figura 3). A referência pode ser feita incluindo-se o número da pergunta entre colchetes, no trecho do cenário onde se descreve o aspecto que a pergunta pretende abordar (Figura 1 e Figura 2).

**Algumas perguntas exploradas nos cenários do Quadro de Aviso:**

1. Dentro/Fora do Renascer, quem utiliza o QA?
2. Do ponto de vista de um funcionário/voluntário/colaborador/interessado do Renascer, para que serve o QA?
3. Que tipos de informação podem ser divulgados no QA?
4. Quem pode divulgar informações no QA?
5. Quem pode consultar o QA?
6. De quanto em quanto tempo deve-se consultar o QA em busca de novidades?
7. Quais são as formas de encontrar um aviso no QA?
8. Como os avisos estão organizados no QA?
9. Quem pode se inscrever no QA?
10. Qual é a vantagem de se inscrever no QA?
11. Existem avisos/seções privativos?
12. Quem pode acessar avisos/seções privativas?
13. Quem define quem pode acessar avisos/seções privativas?
14. Como saber se um aviso/seção é privativo ou público?
15. Quem pode solicitar a inscrição de um novo membro no QA?

Figura 3: Parte das perguntas exploradas nos cenários do Quadro de Aviso.

**Cenários construídos para o Quadro de Avisos:**

***Perguntas exploradas nos cenários***

1. Dentro/Fora do Renascer, quem utiliza o QA?
2. Do ponto de vista de um funcionário/voluntário/colaborador/interessado do Renascer, para que serve o QA?
3. Que tipos de informação podem ser divulgados no QA?
4. Quem pode divulgar informações no QA?
5. Quem pode consultar o QA?
6. De quanto em quanto tempo deve-se consultar o QA em busca de novidades?
7. Quais são as formas de encontrar um aviso no QA?

8. Como os avisos estão organizados no QA?
9. Quem pode se inscrever no QA?
10. Qual é a vantagem de se inscrever no QA?
11. Existem avisos/seções privativos?
12. Quem pode acessar avisos/seções privativas?
13. Quem define quem pode acessar avisos/seções privativas?
14. Como saber se um aviso/seção é privativo ou público?
15. Quem pode solicitar a inscrição de um novo membro no QA?
16. Que informações são necessárias para uma pessoa solicitar sua inscrição no QA?
17. Quando a inscrição de um novo membro no QA é efetivada?
18. Quem efetiva a inscrição de um novo membro no QA?
19. Como a pessoa que solicitou uma inscrição saberá que ela já foi efetuada?
20. É possível imprimir um aviso?
21. Existe um destaque para os avisos mais importantes?
22. Quais são os dados de um aviso?
23. Quem pode alterar um aviso?
24. Quais dados de um aviso podem ser alterados?
25. Quem pode remover um aviso?
26. Quais são os dados de uma seção?
27. Quem pode alterar uma seção?
28. Quais dados de uma seção podem ser alterados?
29. Quem pode remover uma seção?
30. Quem/Quando pode remover um membro do QA?

### **Cenário 1 – Pedido de Inscrição no Quadro de Avisos**

Rita acaba de ser contratada pelo Renascer como funcionária, para trabalhar no atendimento às famílias. Dentre as instruções necessárias para o seu trabalho, Rita é avisada que há um Quadro de Avisos virtual que todos os membros da organização estão começando a utilizar [1]. É uma ótima referência para saber o que está acontecendo, quais os próximos eventos, etc [2,3]. Rita explica que já conhece o Quadro, o qual havia acessado como visitante quando lhe ofereceram o emprego [1]. Então lhe explicaram que, como visitante, ela só poderia acessar os avisos públicos, e agora deveria se inscrever para ter acesso às seções destinadas aos funcionários [5,9,10,11,12]. Rita entra no Quadro e, partindo das instruções que encontra na página, fornece os dados necessários para o pedido de inscrição: seu nome completo, e-mail, e login e senha que deseja utilizar [15,16]. Após finalizar o pedido, novas instruções são apresentadas a Rita, dizendo que agora ela precisa esperar que o responsável pelo Quadro cheque seus dados e efetue sua inscrição [17,18]. Ela não entende direito como saberá quando sua inscrição será efetivada, e ao perguntar sobre isto ao coordenador da sua área, ele lhe diz que ela receberá uma mensagem através do email que forneceu [19].

## **Cenário 2 – Visita a uma seção específica**

A família de André, que esteve internado no Hospital da Lagoa, está sendo atendida pelo Renascer. Sua mãe, Bárbara, se inscreveu para participar de uma das oficinas da organização, mas não sabe muito bem onde deve ir para assistir ao curso escolhido. Ao encontrar-se no centro da cidade com uma das voluntárias que conheceu no hospital, soube que poderia conseguir estas informações pela Internet. Bárbara não tem a mínima idéia de como fazer isso, mas sabe que no colégio de seus filhos tem esta tal de Internet. Ela pede à voluntária que anote num papel tudo que ela precisa para poder chegar a estes dados, para poder pedir ajuda para a professora de seus filhos. Ao falar com a professora que cuida do computador, no colégio das crianças, Bárbara disse o que queria: “Olha só, a senhora procure para mim, por favor, neste lugar (mostrando o papelzinho), que dia e onde que vai ser o curso de costura.” A professora informa o endereço do Quadro de Avisos. Vai direto à seção Oficinas [8] e ordena os avisos por data para ficar mais fácil de encontrar o último aviso sobre os cursos de corte e costura [7]. Ela encontra o aviso e lê as primeiras informações que aparecem: a data de início e o local da Oficina, mas percebe que há mais detalhes sobre este aviso [3,22]. Ele pede para ver estes detalhes, e o Quadro apresenta um mapinha indicando como chegar até o local e fotos do material que os alunos aprendem a fazer. A professora, então, se oferece para imprimir os detalhes do aviso e o mapa, para que Bárbara consiga chegar direitinho no local [20].

## **Cenário 3 – Consultas ao quadro de avisos**

Ana Lúcia é colaboradora da Associação Saúde-Criança Renascer há vários anos. Quando tem um tempo livre maior, ajuda em atividades da Organização, trabalhando em alguma oficina, ou fazendo recreação com as crianças. Como está em férias, Ana resolve usar parte de seu tempo para isto. Ela resolve consultar o Quadro de Avisos para ver se estão precisando de algo específico, ou se ela tem alguma idéia diferente. Ana está bastante acostumada com o uso do Quadro. Como seu contato direto é pequeno, ela tem notícias do que está acontecendo a partir dos avisos na Internet...[2] Semanalmente ela dá uma olhadinha para ver o que tem de novo! [6] Mas agora Ana não quer só dar uma espiada. Ela quer saber de que eles podem estar precisando. Em vez de navegar pelas seções do Quadro [7,8], ela resolve buscar os avisos específicos sobre isto [7]. Assim, ela pode conseguir as informações que deseja de forma mais ágil. Ela informa que deseja buscar avisos sobre “doação”, e todos os avisos que contêm esta palavra são mostrados [7]. Ela logo se interessa por um dos pedidos, relacionado a roupas de inverno (“Posso usar meu tempo livre para ajudar minhas amigas a arrumarem seus armários, na condição de doarem aquelas roupas que não estiverem mais usando”, pensa ela).

## **Cenário 4 – Passeio de um usuário cadastrado para “olhar vitrine”**

Joana, voluntária do Renascer, chega em casa do serviço e vai dar uma verificada rápida nas últimas notícias desta ONG, através de seu Quadro de Avisos virtual. O trabalho de voluntária de Joana é na sede do Renascer, e ela tem acesso às seções referentes a Administração, Atendimento, Oficina, Sede, Eventos e Cursos Profissionalizantes [11]. Os avisos mais importantes, relacionados a estas seções, já aparecem no quadro geral, logo que ela acessa o Quadro [21]. Com isso, Joana descobre que tem reunião no dia seguinte, sobre assuntos administrativos. “Que bom que este aviso está bem destacado... Com a pressa que estou talvez nem chegasse a vê-lo”, pensa ela. Joana aproveita e



verifica o texto completo do aviso, obtendo a pauta da reunião, para, desta forma, melhor se preparar para ela [22].

### **Cenário 5 – Publicação de um aviso**

Ana, que tinha resolvido arrumar os armários das amigas para poder arrecadar mais roupas, descobre que muitas delas até já teriam doado antes, mas tinham dificuldade de levar as roupas até a Sede e acabavam doando para algum conhecido. Ana tem uma idéia para poder ajudar mais um pouquinho. Ela resolve colocar um aviso no Quadro, informando que estará recolhendo roupas de inverno e que pode passar na casa das pessoas que não têm como levar até a Sede, para buscá-las. Após efetuar login do Quadro, ela verifica que pode criar novos avisos [4]. “Que bom que posso colocar este aviso direto, sem maiores burocracias”, diz ela a uma amiga que ia ajudá-la na busca das roupas. Ela escolhe a seção de “Doações” para colocar seu aviso [22]. Ela coloca um título e uma chamada legal, assim as pessoas não pensarão duas vezes antes de abri-lo [22]. Como a autora é ela mesma, ela não precisa informar este dado (“Oba, menos trabalho!”) [22]. No texto do aviso ela informa o que pretende fazer, seu telefone e os horários nos quais está disponível. Além disso, ela coloca um prazo máximo para que o aviso fique disponível para as pessoas lerem...[22] Não adianta ter o aviso depois que ela voltar das férias, pois ela não poderá buscar mais as roupas. Então ela pede que o aviso seja mostrado só até o final do mês. Ana sai da Internet e vai ao cinema com suas amigas. Ela volta empolgadíssima do cinema, pois mais uma amiga se animou a ajudá-las. Ela corre para o computador para alterar seu aviso, colocando os dias e horários que sua amiga Júlia também pode [24]. Ela nunca alterou um aviso, mas sempre percebeu que na chamada dos seus próprios avisos havia uma indicação da possibilidade de alterá-lo [23]. Ela coloca os dados novos e manda postar novamente. “Ana pensa bem e acha que ainda pode melhorar o aviso. Como é pouco o tempo que ela tem disponível até que suas férias terminem, ela volta a edição do aviso e indica que deseja que o mesmo seja mostrado logo na primeira página do Quadro [21,24].

### **Cenário 6 – Remoção de um aviso**

Rita está nos seus primeiros dias de trabalho no Renascer. Dois membros diferentes da organização pediram que ela inserisse avisos sobre uma reunião que ocorreria na semana seguinte. Após inserir os dois avisos, cada um com um autor diferente [4], Rita se dá conta que ambos tratavam da mesma reunião. Ela corre para apagar um deles. Como foi ela que inseriu os avisos, ela tem permissão para mexer neles [23,25]. “Ufa”, pensa Rita, “ainda bem que me dei conta bem rápido... ia pegar muito mal!”. Alguns dias depois, quando um colega pede para ela alterar um aviso dele, ela lhe informa que não pode, pois ela leu no sistema de ajuda do Quadro que só quem é o autor ou quem inseriu é que tem permissão para fazer isto [23].

### **Cenário 7 – Manutenção do quadro de avisos**

Renato é um dos administradores do Quadro de Avisos. Dentre as tarefas que eles executam, está a checagem dos dados das pessoas que querem se tornar membros do Quadro [18]. Eles precisam verificar se os dados destas pessoas estão corretos e quais seções estas podem visualizar [13]. Além disso, quando a administração resolve inserir uma seção nova ou alterar o nome, o responsável ou quem tem acesso às seções existentes

no Quadro, são eles também que realizam estas atividades [26,27,28]. Carolina está checando suas tarefas de administradora do Quadro de Avisos e vê no seu e-mail três pendências. Ela precisa remover um membro do Quadro, pois a pessoa se desligou do Renascer [30], alterar o responsável de uma seção [27,28] e, também, remover uma seção [29]. “Eu sabia que não podia ser tudo fácil.”, pensa ela, “Remover uma seção sempre é mais chatinho, pois tenho que remanejar os avisos que não podem ser removidos juntamente com a seção, para outras seções” [23,24]. Então Carolina altera a seção dos avisos e remove a seção. Depois disto, ela procura o membro a ser removido, confere os dados para ver se é ele mesmo e o remove do Quadro. Ela pensa “Então este membro não era autor de avisos postados no quadro, pois se ele fosse, eu não teria conseguido removê-lo [30].” Carolina altera o responsável de uma seção e pensa “Até que foi fácil.”

### Tabela de signos do Quadro de Avisos

Signo Composto: Usuário		
Signo	Definição	Tipo
nome	nome do usuário	domínio
email	email do usuário	domínio
login	login do usuário no sistema	aplicação
senha	senha do usuário no sistema	aplicação
Signo Composto: Aviso		
Signo	Definição	Tipo
título	título do aviso	domínio
autor	autor do aviso	domínio
data	data de postagem do aviso	domínio
chamada	texto do aviso (resumido)	domínio
texto	detalhes do texto do aviso	domínio
quem_postou	membro que postou o aviso	aplicação
quadro_geral	indicação se o aviso deve ser postado no quadro geral	domínio
prazo	prazo de validade da exposição do aviso	domínio
Signo Composto: Seção		
Signo	Definição	Tipo
nome	nome da seção	domínio
responsável	responsável pela seção	domínio
pública	indicação se a seção deve ser pública	domínio
Signo Composto: Pedido		
Signo	Definição	Tipo
texto	texto do pedido de revisão de dados	aplicação
Signo	Definição	Tipo
busca_personalizada	opções com as formas de busca personalizada	aplicação
opção_ordenação	opções com as formas de ordenação possíveis	aplicação
critério	critério de busca	aplicação

## APÊNDICE B – CONSTRUÇÃO DOS MODELOS DE TAREFAS

A partir dos cenários construídos pelos projetistas e ratificados pelos clientes e usuários (como foi visto no Apêndice A), a próxima etapa no projeto de IHC é a modelagem de tarefas. Neste Apêndice será descrita a modelagem de tarefas proposta em [Paula, 2003], que é fundamentada na teoria da Engenharia Semiótica.

O primeiro passo da modelagem de tarefas consiste em extrair dos cenários as **metas** ou objetivos que os usuários terão ao utilizar o sistema. Cada cenário pode descrever mais de uma meta, e uma meta pode ser descrita em mais de um cenário. Sendo assim, deve-se ler todos os cenários para identificar e organizar as metas neles descritas. Para cada meta deve-se identificar os cenários que lhe deram origem, pois esta identificação facilita a rastreabilidade entre as diferentes representações utilizadas ao longo do projeto.

O conjunto de metas encontradas no cenário deve ser organizado em um ou mais diagramas hierárquicos de metas. Este diagrama provê uma visão macro das metas que cada classe de usuários pode realizar, organizadas de acordo com algum critério que o projetista ache relevante. Uma meta é representada por um retângulo com bordas arredondadas contendo o nome da meta, expresso do ponto de vista do usuário. Para isto, pode-se considerar como expressão da meta a lacuna indicada em uma fala do usuário tal como “Eu quero utilizar o sistema para <meta>”. Cada meta é identificada por uma letra, e sua representação inclui também o(s) papel(papéis) de usuários que poderá atingi-la através do sistema.

No diagrama do sistema de Quadro de Avisos da Figura 4, as metas foram agrupadas e identificou-se quatro tipos de situação de uso importantes: a solicitação de inscrição no quadro por visitantes (representada pela meta Solicitar inscrição), a consulta de avisos por visitantes ou membros (representada pela meta Consultar os avisos), a manipulação de avisos por membros (representada pela árvore Manipular avisos) e a administração do quadro de avisos pelos administradores do Quadro (representada pela árvore Administrar o quadro). Esta última situação não faz parte da *atividade-fim* do Quadro, mas compreende metas necessárias para o *suporte da aplicação*.

Atividade-fim e suporte da aplicação foram os critérios de classificação das metas adotados pelo projetista do Quadro de Avisos. Para cada aplicação, as metas podem ser classificadas e organizadas de forma diferente, de acordo com critérios adotados pelo projetista do sistema. Critérios como, por exemplo, quais são as metas a serem priorizadas no projeto da aplicação, quais metas gastam mais tempo para serem modeladas, a frequência na qual se espera que cada meta seja atingida, quais se relacionam “fortemente” e precisam ficar “conectadas” no modelo de interação, ou até mesmo quais metas podem ser atingidas por determinadas classes de usuários.

Os papéis de uma meta (as classes de usuários que poderão realizar a meta) em um nível hierarquicamente superior no diagrama de metas são propagados para os níveis inferiores. Por exemplo, o papel “Adm” representado na meta Administrar o quadro será também um dos papéis de todas as metas desta sub-árvore. Caso haja papéis adicionais para uma determinada meta, estes devem ser representados na própria representação da meta, como

em Alterar uma seção, cujas tarefas podem ser realizadas tanto por administradores, “Adm”, quanto por responsáveis pela seção, “RS”.

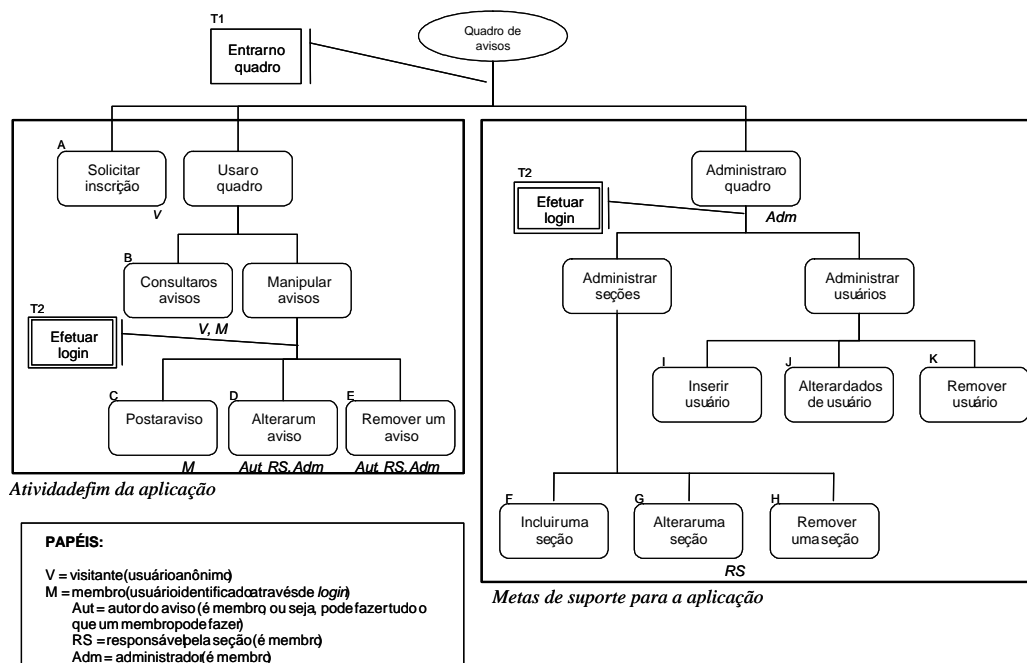


Figura 4: Diagrama hierárquico de metas.

Após a criação do diagrama, para cada meta identificada (metas no último nível - folhas - do diagrama de metas) será associado um modelo de tarefas que consiste em uma decomposição hierárquica dos passos necessários para se atingir a meta correspondente, do ponto de vista do usuário que visa atingi-la. Em outras palavras, cada tarefa pode ser decomposta em subtarefas, e cada subtarefa pode ser novamente decomposta em novas subtarefas, e assim sucessivamente. Ao representar cada meta, deve-se indicar os cenários que a descrevem (mesmo que parcialmente) (Figura 5).

## Meta B: Consultar avisos

Cenários Associados: 2, 3, 4 e 5

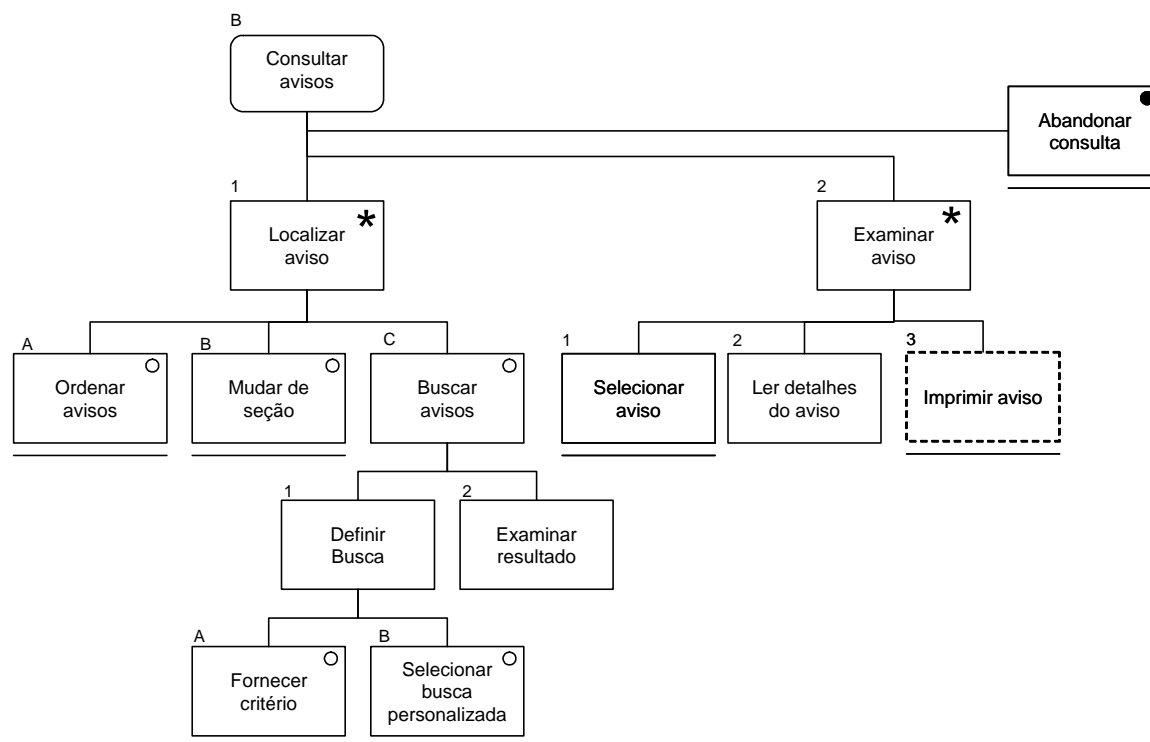


Figura 5: Modelo de Tarefas para a meta Consultar avisos.

As **tarefas** são representadas por retângulos (por exemplo, Ler detalhes do aviso), com marcações especiais para indicar a que tipo de estrutura estão associadas, como será visto adiante. Deve-se ressaltar que o modelo de tarefas deve refletir como o usuário trabalha, evitando-se focar em um ambiente ou plataforma tecnológica específica. Esta consideração não apenas facilita o re-uso de modelos de tarefas, como também evita que decisões sobre a solução de interação ou de interface sejam tomadas prematuramente, dificultando a exploração de soluções alternativas por parte dos projetistas.

Deve-se observar que as tarefas que o usuário vai realizar de fato são as que estão representadas nas folhas (último nível) da estrutura hierárquica. A decomposição das tarefas em subtarefas deve parar antes que o modelo inclua detalhes operacionais de interface, tais como “digitar X”, “pressionar botão Y”, etc. Dependendo do ponto em que se pára a decomposição das tarefas, existem tarefas tão simples que poderão – em fases posteriores no processo de *design* – ser mapeadas diretamente em um elemento de interface ou de interação. Estas tarefas são representadas no modelo como **operadores**, os quais são representados por uma linha abaixo do retângulo. Como pode-se verificar na Figura 5, o projetista optou por modelar a tarefa Imprimir aviso como um operador. Então, na modelagem da interação, o projetista deve, provavelmente, estabelecer valores *default* para os signos envolvidos nesta tarefa e o usuário, ao executar esta tarefa, não precisará configurar parâmetros de impressão. Isto torna a solução mais simples e rápida para os usuários, porém menos flexível.

## Estruturas de Tarefas

As tarefas podem ser organizadas nos seguintes tipos de estruturas: sequenciais, independentes de ordem, alternativas e iterativas.

Em uma estrutura **sequencial**, existe uma ordem em que as tarefas devem necessariamente ser efetuadas pelo usuário. As tarefas, nesta estrutura, são representadas por retângulos contendo o nome da tarefa, expresso do ponto de vista do usuário, e um número indicando sua posição na sequência (Figura 6).

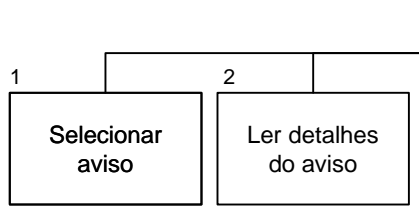


Figura 6: Estrutura sequencial.

Algumas tarefas podem ser realizadas em qualquer ordem. Uma estrutura de tarefas **independente de ordem** representa um conjunto (e não uma sequência) de tarefas a serem efetuadas pelo usuário. Tipicamente, o projetista sugere uma ordem de execução, mas é o usuário quem determina, de fato, em que ordem as tarefas serão efetuadas. Neste tipo de estrutura, as tarefas são representadas como as tarefas sequenciais mas, como a ordem é apenas sugerida, incluímos um ponto de interrogação após o número que indica a posição relativa da tarefa na estrutura. No exemplo, identificamos algumas tarefas independentes de ordem (Figura 7).

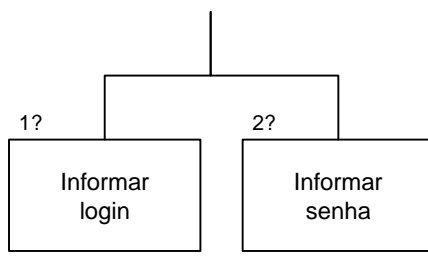


Figura 7: Estrutura independente de ordem.

Para o atingimento de uma meta, há momentos em que diversos cursos de ação são possíveis. Tais cursos de ação são representados por uma estrutura **alternativa**, onde o usuário deverá selecionar qual das tarefas da estrutura será efetuada. Nesta estrutura, utilizam-se pequenos círculos no canto superior direito do retângulo de cada tarefa alternativa, e letras como identificadores em vez de números (Figura 8).

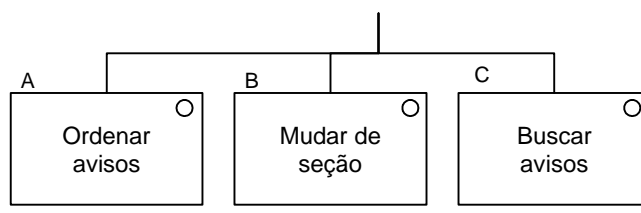


Figura 8: Estrutura alternativa.

Quando uma tarefa pode ser realizada diversas vezes, utiliza-se uma estrutura **iterativa**. Um asterisco (\*) no canto superior direito do retângulo é utilizado para indicar a iteração (Figura 9). Geralmente, uma tarefa iterativa representa tarefas que podem ser efetuadas zero ou mais vezes. Caso seja necessário definir um número mínimo ou máximo de repetições, pode-se incluir, acima do retângulo e alinhada à direita, uma expressão que indica a cardinalidade da iteração. A expressão [n+] indica que a tarefa deve ser realizada pelo menos n vezes; [m..n] indica que a tarefa deve ser realizada no mínimo m e no máximo n vezes.

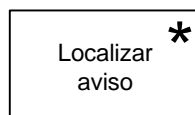


Figura 9: Estrutura iterativa.

A Figura 10 apresenta uma tarefa iterativa com suas sub-tarefas alternativas. Esta estrutura projetada pelo projetista indica que o usuário poderá realizar a tarefa Localizar aviso n vezes consecutivas. Em cada vez que ele realizar esta tarefa, ele terá que optar por uma forma de localização, ou seja, através da ordenação dos avisos, da mudança da seção ou da busca por avisos.

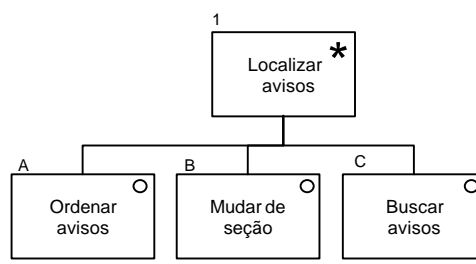


Figura 10: Tarefa Localizar avisos e suas sub-tarefas.

### Tarefas Opcionais

Quando o usuário pode optar por realizar ou não uma tarefa, ela é dita **opcional**, e é representada com uma borda tracejada (Figura 11).

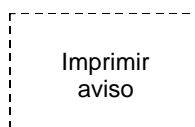


Figura 11: Tarefa opcional.

### Tarefas Ubíquas

Algumas tarefas podem ser feitas em qualquer ponto da realização da meta. Estas tarefas são denominadas **ubíquas**, e são representadas por um círculo preenchido no canto superior direito do retângulo (Figura 12). Nesta figura, pode-se verificar que o projetista modelou como tarefa ubíqua a tarefa Abandonar consulta. Isto significa que o projetista deve prover explicitamente meios para o usuário realizar esta tarefa dentro do sistema, de forma independente do ambiente em que o sistema executar (sem contar, por exemplo, com o botão “fechar janela” no Windows ou “voltar” num navegador *web*).

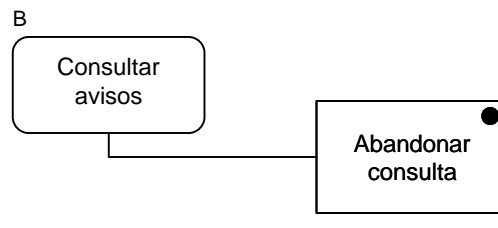


Figura 12: Tarefa ubíqua.

### Pré-condições

É comum existirem tarefas que são **pré-condições** para a realização de uma determinada meta ou tarefa. Estas pré-condições podem ser representadas através de um *callout* ligado a uma meta ou tarefa (Figura 13). No diagrama de metas da Figura 4, pode-se verificar que a tarefa Efetuar login é pré-condição para a realização de determinadas metas, como por exemplo a Postar aviso.

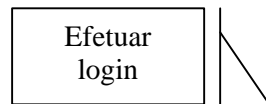


Figura 13: Representação de pré-condição para a realização de uma tarefa ou meta.

### Re-uso de Tarefas e Metas

Há tarefas que fazem parte da estrutura de diversas metas. Neste caso, para facilitar a representação e manutenção do modelo, estas tarefas são definidas como **estereótipos**. Um estereótipo de tarefa pode ou não receber parâmetros. Graficamente, é representado por um retângulo com borda dupla, contendo uma expressão da forma <<nome\_do\_estereotipo(parâmetro1, parâmetro2, ...)>> ou simplesmente <<nome\_do\_estereótipo>> (no caso de não haver parâmetros). Como pode-se verificar no diagrama hierárquico de metas da Figura 4, este estereótipo foi utilizado como pré-condição para a execução de algumas metas.



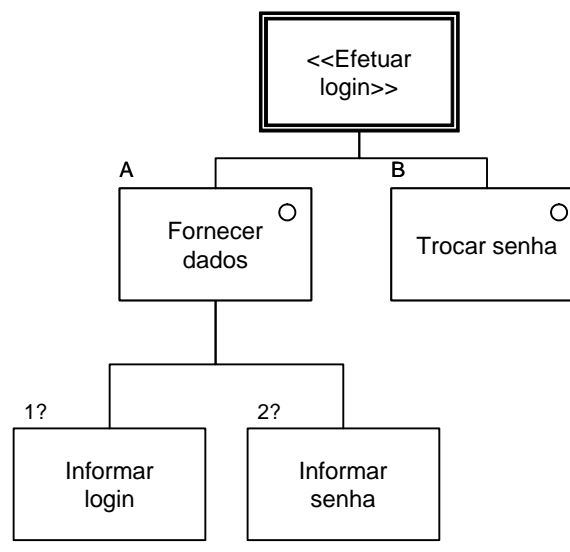


Figura 14: Definição do estereótipo <<Efetuar login>>.

Além do estereótipo, outra forma de re-uso é fazer referência direta a metas ou tarefas já utilizadas em outras partes de uma mesma modelagem, mas que não são genéricas o suficiente para se tornarem estereótipos. No caso de referência a uma meta isto é feito incluindo-se a expressão Nome da Tarefa = META X (onde X indica a meta em questão). No caso de referência a tarefas, é utilizada a expressão Nome da Tarefa = X.Y (onde X representa a meta e Y a tarefa em questão). E, no caso de referência a subtarefas, Nome da Tarefa = X.Y.Z (onde X representa a meta, Y a tarefa e Z a subtarefa) e assim sucessivamente, dependendo do nível onde encontra-se a subtarefa. Na Figura 15 vê-se um exemplo de reaproveitamento vê-se um exemplo de reaproveitamento da tarefa Localizar avisos que foi especificada na meta Consultar avisos (Figura 5).

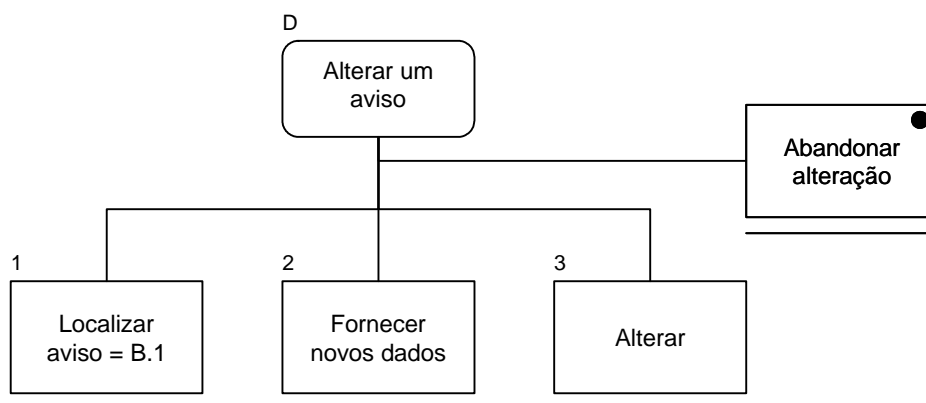


Figura 15: Reaproveitamento direto de tarefas.

A Engenharia Semiótica ressalta a importância de se representar os signos e as rupturas na conversa preposto do *designer*-usuário que possam ocorrer durante uma interação. Sendo assim, a representação diagramática, que tem como objetivo dar uma visão geral das metas e da hierarquia de tarefas, é complementada com uma especificação textual, que inclui, para cada tarefa, os signos a ela associados e as decisões do projetista sobre prevenção e tratamento de erros a ela vinculados.

## Signos

Os cenários, além de auxiliarem na construção da tabela de signos, também fornecem informações sobre quando, como e onde os signos são utilizados. Através da análise dos cenários, pode-se detectar alguns signos apresentados ou manipulados em cada tarefa (como foi visto no Apêndice A). Sendo assim, após a modelagem diagramática das tarefas, associa-se às tarefas os signos correspondentes. Deve-se representar textualmente a identificação da tarefa, a palavra-chave SIGNOS, e incluir os signos correspondentes.

Para se fazer referência a um dos signos que compõem um signo composto, deve-se utilizar o formato nome\_signo\_composto.nome\_signo. Se, por outro lado, for necessário fazer referência a todos os signos de uma determinada composição, pode-se utilizar o formato nome\_signo\_composto.\*. Por exemplo, o signo correspondente ao login do usuário pode ser expresso como usuário.login, enquanto em uma situação em que são apresentados e/ou manipulados todos os dados de um usuário, o conjunto destes signos pode ser representado por usuário\*. Caso os signos não possam ser definidos nesta fase de modelagem, ou dependam da configuração do sistema em tempo de execução, pode-se utilizar a notação nome\_signo\_composto.X.

Cada signo que for utilizado em apresentação (dado de *output*) é representado por seu nome seguido de ponto de exclamação (signo1!). Já um signo correspondente a um dado que será fornecido ou manipulado pelo usuário (dado de *input*) é representado por seu nome seguido de ponto de interrogação (signo2?). Caso haja diversas instâncias de um mesmo tipo, pode-se utilizar o construto conjunto(nome\_do\_signo, cardinalidade) para representá-las (Figura 16).

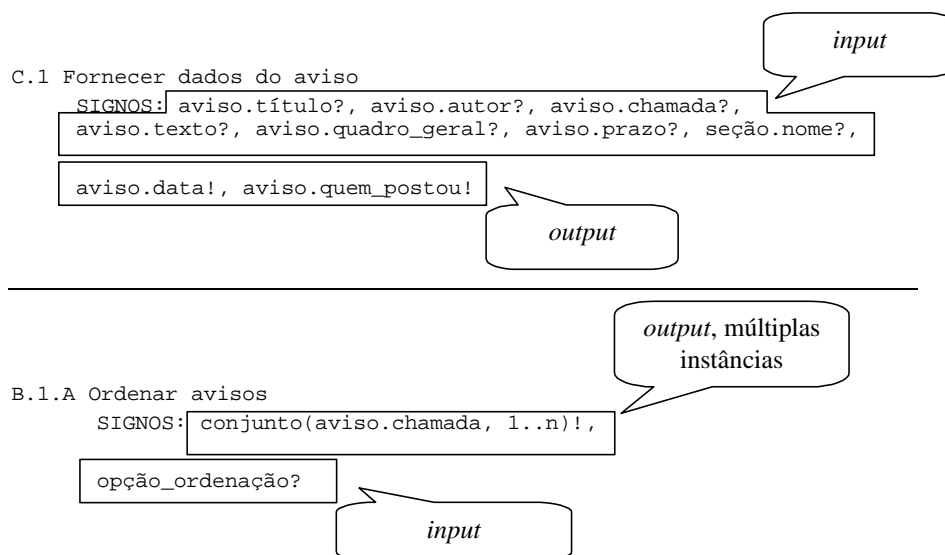


Figura 16: Identificação dos signos no modelo de tarefas.

## Prevenção e Tratamento de Erro

Como dito anteriormente, a Engenharia Semiótica ressalta a importância de se representar os signos e rupturas que possam ocorrer durante uma interação. Por isto, além da especificação dos signos relacionados a cada tarefa, deve-se identificar as possíveis rupturas na comunicação preposto do *designer*-usuário que podem ocorrer durante uma

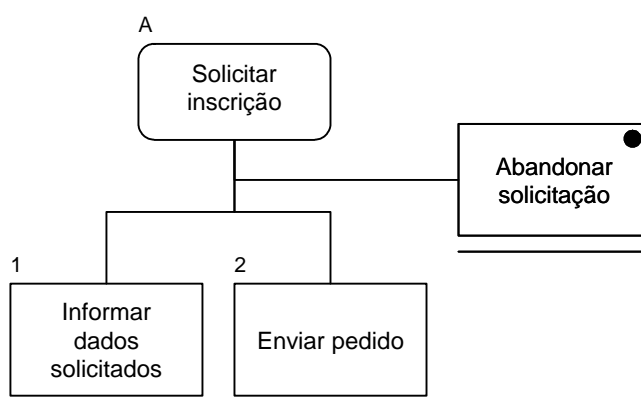
interação, e que já podem ser previstas e registradas no modelo de tarefas. Sendo assim, para cada tarefa, deve-se especificar os tipos de apoio a prevenção e tratamento de erro que o *designer* pretende prover, classificando-os em uma das seguintes categorias:

- **prevenção passiva:** Erros que devem ser prevenidos por documentação ou instruções *online* (por exemplo, “O usuário não possui acesso ao sistema”).
- **prevenção ativa:** Erros que devem ser prevenidos ativamente pelo sistema (por exemplo, tarefas que devem ou não devem estar disponíveis em certas situações. No modelo de interface, isto poderá ser mapeado em, por exemplo, habilitar ou desabilitar botões de acordo com o *status* atual da aplicação; impedir que o usuário digite letras ou símbolos em campos numéricos; e assim por diante).
- **prevenção apoiada:** Situações que o sistema consegue detectar como sendo erros em potencial, mas cuja decisão recai sobre o usuário. Geralmente são realizados na interface por mensagens de confirmação (por exemplo, “Arquivo já existe, deseja sobrescrever?”).
- **tratamento apoiado:** Erros que devem ser tratados pelo usuário com apoio do preposto do *designer* (por exemplo, apresentar uma mensagem de erro e uma oportunidade para o usuário corrigi-lo).
- **captura de erro:** Erros que são identificados pelo sistema e devem ser notificados ao usuário, sem que haja qualquer passo corretivo possível dentro do sistema (por exemplo, “O arquivo está corrompido.” ou “Espaço em disco insuficiente.”).

Os tipos de prevenção e tratamento são apresentados, no modelo de tarefas, de forma textual, logo após a indicação dos signos (por exemplo, TRATAMENTO APOIADO, na Figura 17).

### Meta A: Solicitar inscrição

Cenários relacionados: 1



#### A.1 Informar dados solicitados

SIGNOS: usuário.nome?, usuário.email?, usuário.login?, usuário.senha?

TRATAMENTO APOIADO: falta de informações obrigatórias

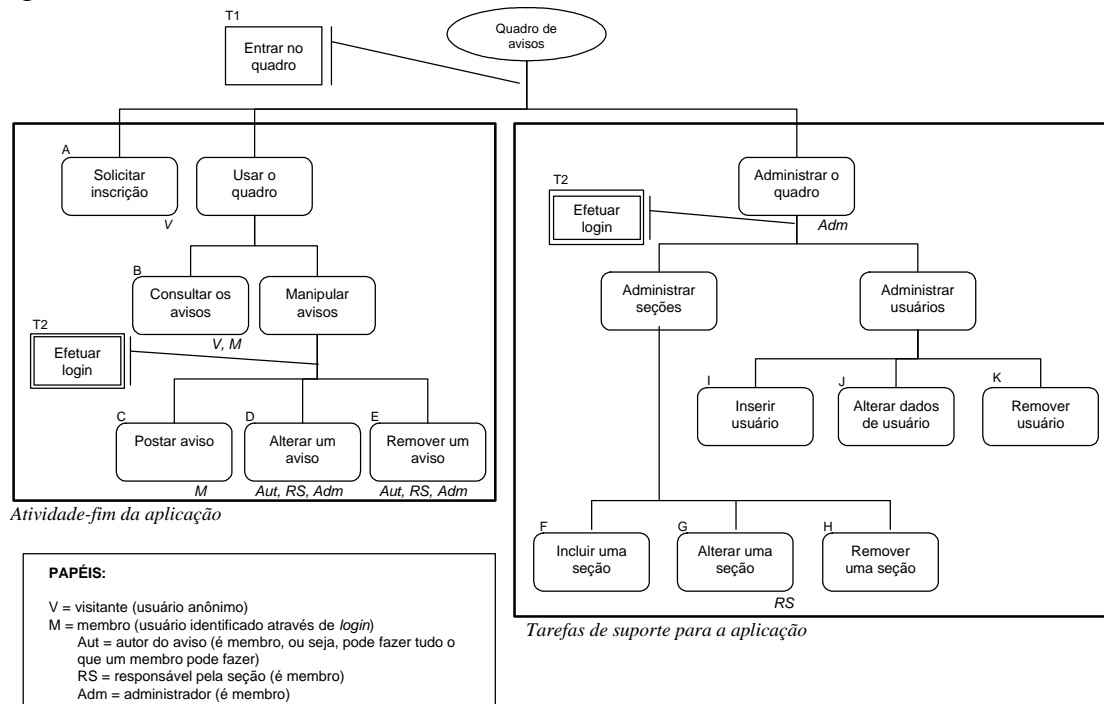
TRATAMENTO APOIADO: login existente

TRATAMENTO APOIADO: e-mail inválido

Figura 17: Modelo de tarefas completo da meta Solicitar inscrição.

## Modelos de Tarefas do Quadro de Avisos

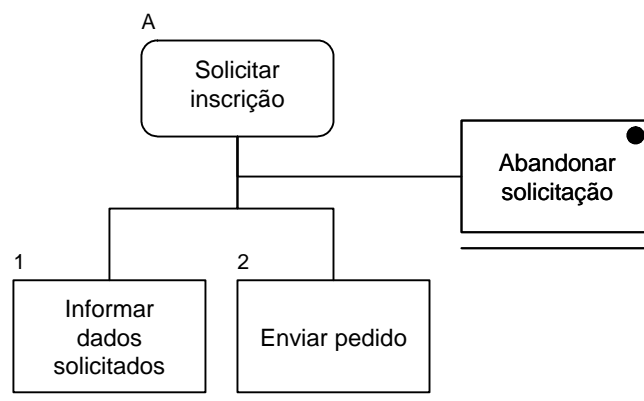
### Diagrama de Metas



### Meta A: Solicitar inscrição

Cenários relacionados: 1

Papéis: visitante



### A.1 Informar datos solicitados

```
SIGNOS: usuário.nome?, usuário.email?, usuário.login?,
usuário.senha?
```

TRATAMENTO APOIADO: falta de informações obrigatórias

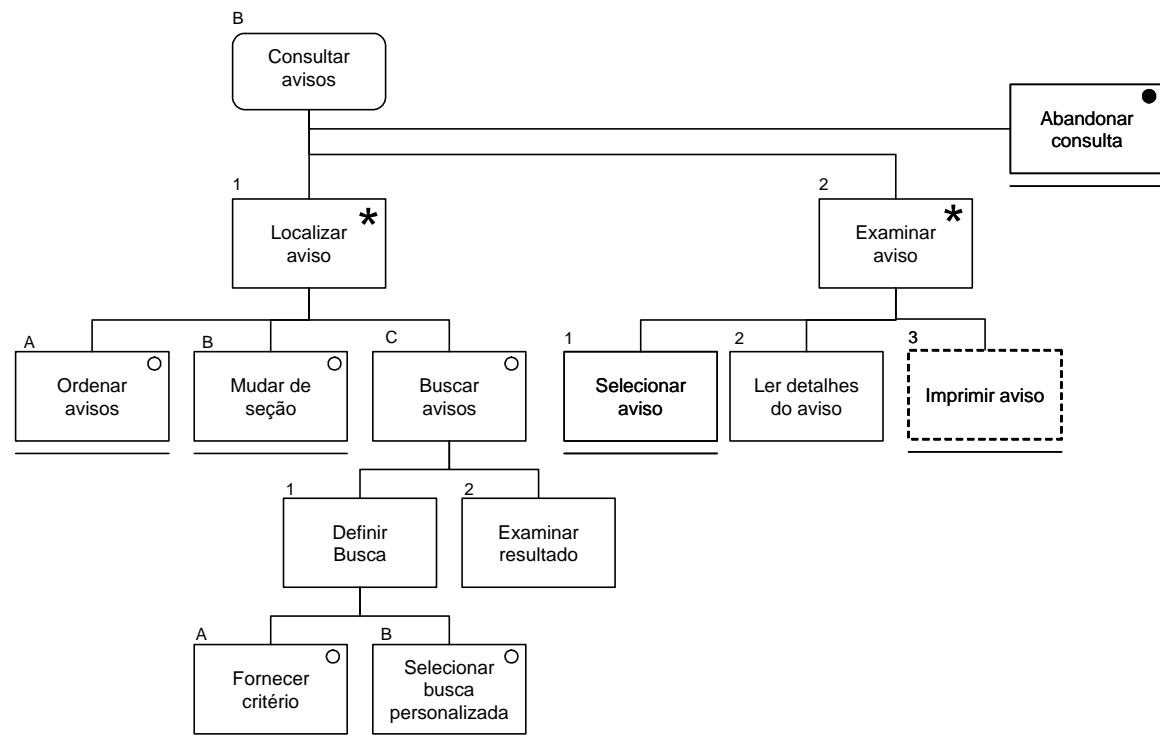
TRATAMENTO APOIADO: login existente

TRATAMENTO APOIADO: e-mail inválido

## Meta B: Consultar avisos

Cenários Associados: 2, 3 e 4

Papéis: visitante, membro



### B.1.A Ordenar avisos

SIGNOS: conjunto(aviso.chamada, 1..n)!,  
opção\_ordenação?

### B.1.B Mudar de seção

SIGNOS: conjunto(aviso.chamada, 1..n)!,  
seção.nome?

#### B.1.C.1.A Fornecer critério

SIGNOS: critério?

#### B.1.C.1.B Selecionar busca personalizada

SIGNOS: busca\_personalizada?

#### B.1.C.2 Examinar resultado

SIGNOS: conjunto(aviso.\*, 0..n)!

### B.2.1 Selecionar aviso

SIGNOS: conjunto(aviso.\*, 1..n)!, aviso.\*?

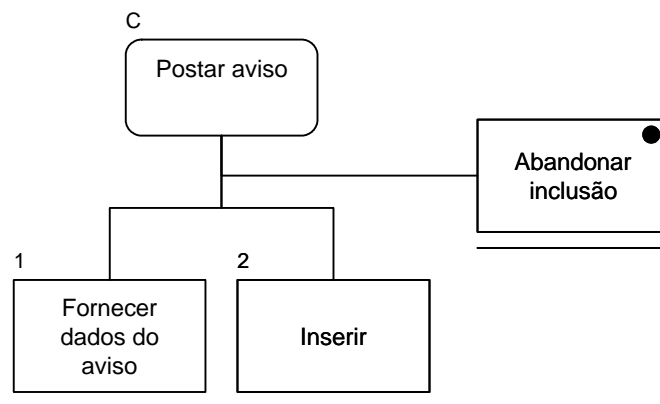
### B.2.2 Ler detalhes do aviso

SIGNOS: aviso.\*!

**Meta C: Postar aviso**

Cenários Associados: 5,6

Papéis: membro

**C.1 Fornecer dados do aviso**

SIGNOS: aviso.título?, aviso.data!, aviso.autor?, aviso.chamada?,  
aviso.texto?, aviso.quem\_postou!, aviso.quadro\_geral?, aviso.prazo?,  
seção.nome?

TRATAMENTO APOIADO: falta de informações obrigatórias

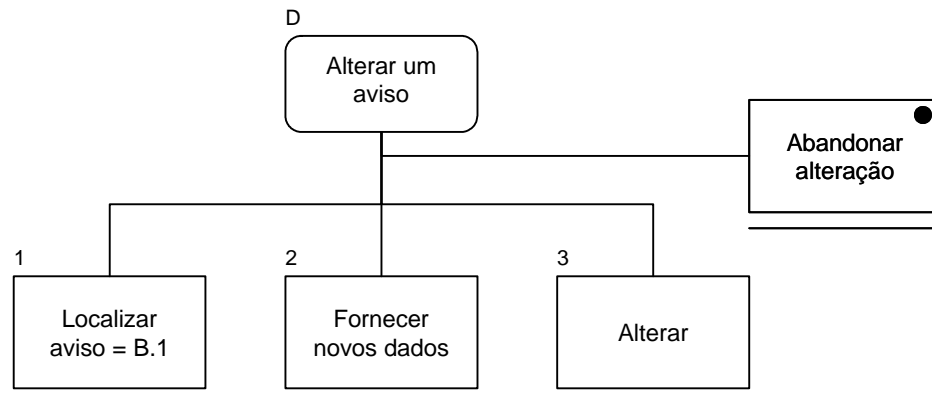
PREVENÇÃO ATIVA: somente membro pode postar aviso

PREVENÇÃO ATIVA: membro só pode postar aviso em seções a que tenha  
permissão

## Meta D: Alterar um aviso

Cenários Associados: 5

Papéis: membro



### D.2 Fornecer novos dados

SIGNOS: aviso.título?, aviso.data!, aviso.autor?, aviso.chamada?, aviso.texto?, aviso.quem\_postou!, aviso.quadro\_geral?, aviso.prazo?, seção.nome?

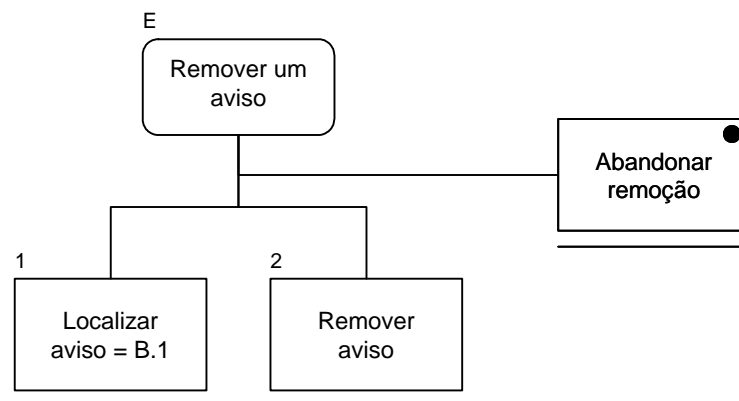
TRATAMENTO APOIADO: falta de informações obrigatórias

PREVENÇÃO ATIVA: somente o autor, quem inseriu o aviso ou o responsável pela seção podem alterar o aviso

## Meta E: Remover um aviso

Cenários Associados: 6

Papéis: membro



### E.2 Remover aviso

SIGNOS: aviso.\*!

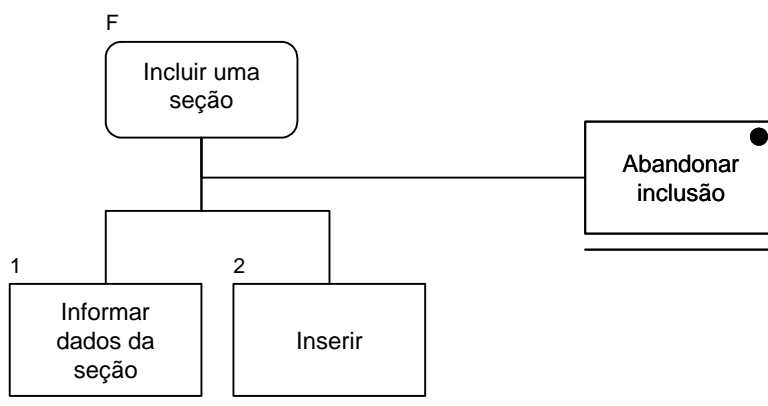
PREVENÇÃO ATIVA: somente o autor, quem inseriu o aviso ou o responsável pela seção podem remover o aviso

PREVENÇÃO APOIADA: o aviso será permanentemente removido do Quadro de Avisos

## Meta F: Incluir uma seção

Cenários Associados: 7

Papéis: administrador



### F.1 Informar dados da seção

SIGNOS: seção.nome?, seção.responsável?, seção.pública?, conjunto(usuário.nome, 1..n)?

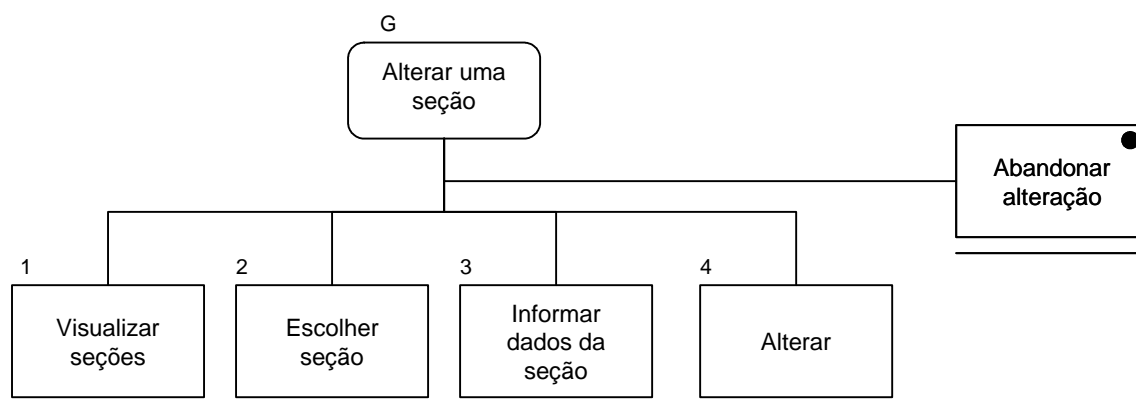
TRATAMENTO APOIADO: falta de informações obrigatórias

TRATAMENTO APOIADO: nome de seção já existe

## Meta G: Alterar uma seção

Cenários Associados: 7

Papéis: administrador, responsável pela seção



### G.1 Visualizar seções

SIGNOS: conjunto(seção.\*, 1..n)!

PREVENÇÃO ATIVA: somente administradores ou responsáveis por seção podem manipular seções

### G.2 Escolher seção

SIGNOS: seção.\*?

### G.3 Informar dados da seção

SIGNOS: seção.nome?, seção.responsável?, conjunto(usuário.nome, 1..n)?, seção.pública?

TRATAMENTO APOIADO: falta de informações obrigatórias

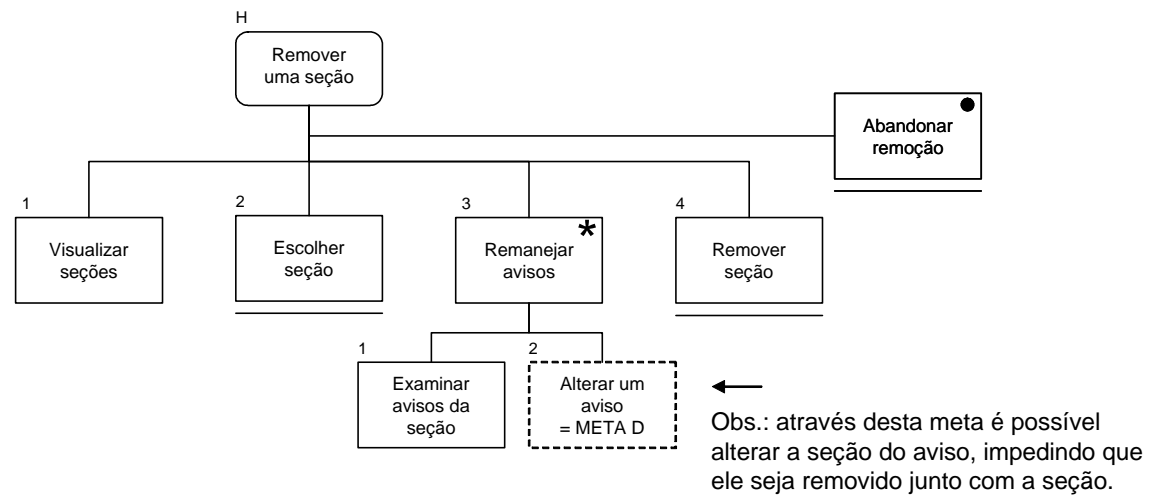
TRATAMENTO APOIADO: nome de seção já existe



## Meta H: Remover uma seção

Cenários Associados: 7

Papéis: administrador, responsável pela seção



### H.1 Visualizar seções

SIGNOS: conjunto(seção.\*, 1..n)!

PREVENÇÃO ATIVA: somente administradores ou responsáveis por seção podem manipular seções

### H.2 Escolher seção

SIGNOS: seção.\*?

### H.3.1 Examinar avisos da seção

SIGNOS: conjunto(aviso.\*, 1..n)!

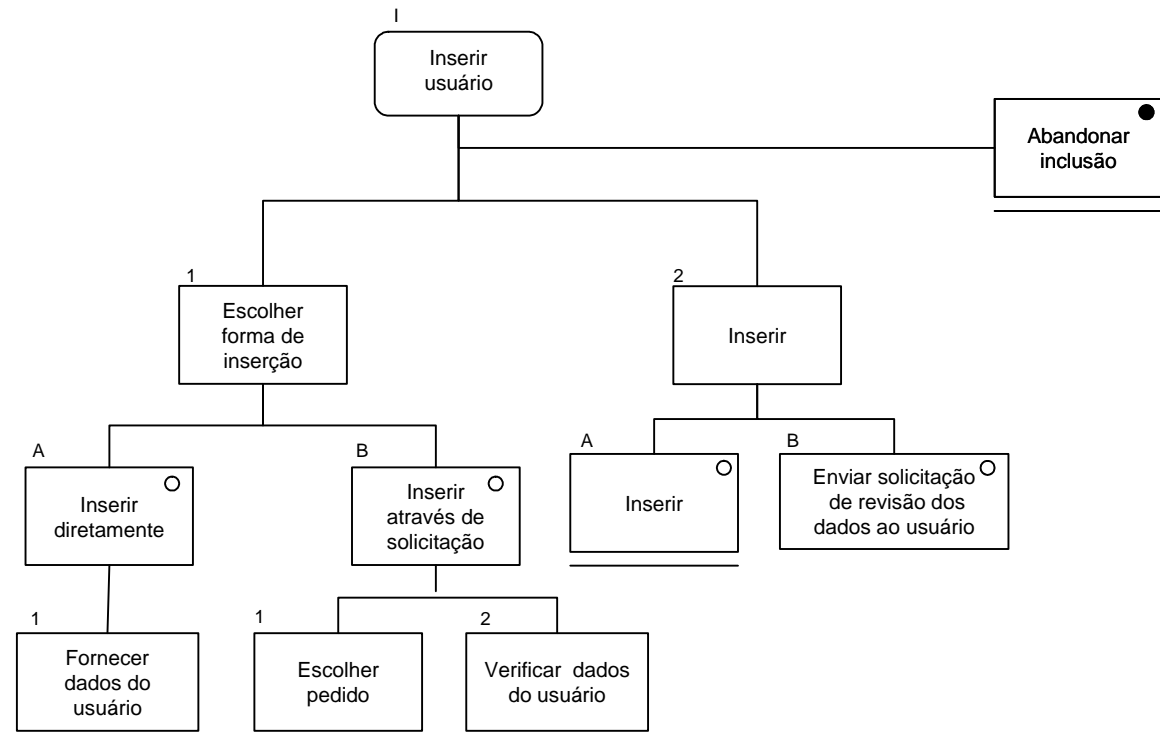
### H.4 Remover seção

ALERTA: a seção e seus avisos serão permanentemente removidos do Quadro de Avisos

**Meta I: Inserir usuário**

Cenários Associados: 7

Papéis: administrador



I.1.A.1 Fornecer dados do usuário

SIGNOS: usuário.nome?, usuário.email?, usuário.login?,  
usuário.senha?

I.1.B.1 Escolher pedido

SIGNOS: usuário.nome?

I.1.B.2 Verificar dados do usuário

SIGNOS: usuário.nome!, usuário.email!, usuário.login!

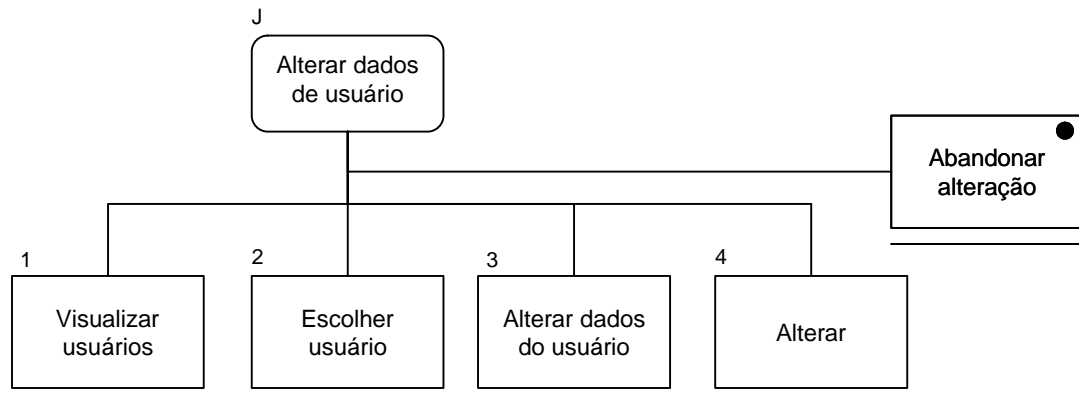
I.2.B Enviar solicitação de revisão dos dados ao usuário

SIGNOS: usuário.\*!, pedido.texto?

## Meta J: Alterar dados de usuário

Cenários Associados: 7

Papéis: administrador



### J.1 Visualizar usuários

SIGNOS: conjunto((usuário.nome, usuário.login), 1..n)!

PREVENÇÃO ATIVA: somente administradores podem alterar os dados de um usuário

### J.2 Escolher usuário

SIGNOS: conjunto((usuário.nome, usuário.login), 1..n)?

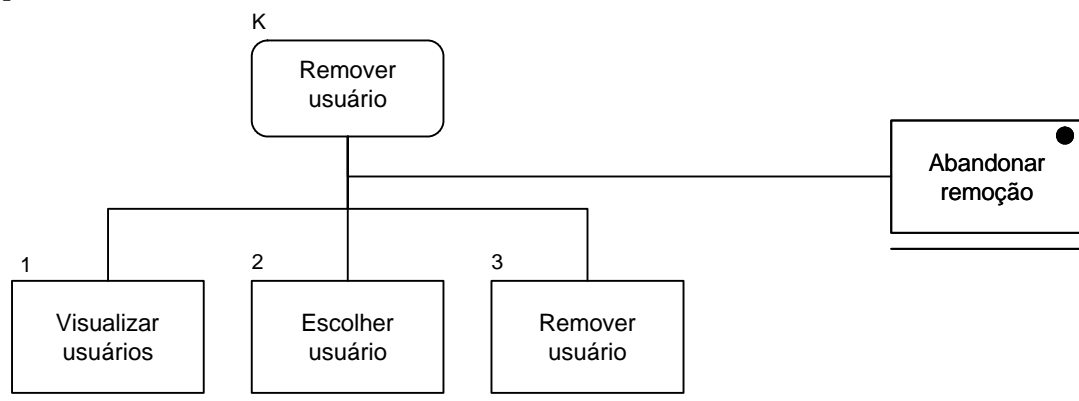
### J.3 Alterar dados do usuário

SIGNOS: usuário.nome?, usuário.email?, usuário.login?, usuário.senha?

## Meta K: Remover usuário

Cenários Associados: 7

Papéis: administrador



### K.1 Visualizar usuários

SIGNOS: conjunto ((usuário.nome, usuário.login), 1..n)!

PREVENÇÃO ATIVA: somente administradores podem remover um usuário

### K.2 Escolher usuário

SIGNOS: conjunto ((usuário.nome, usuário.login), 1..n)?

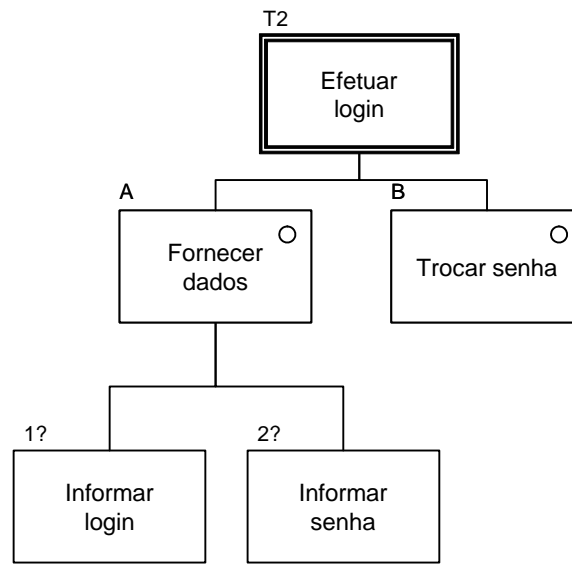
### K.3 Remover usuário

PREVENÇÃO APOIADA: o usuário será permanentemente removido do Quadro de Avisos

TRATAMENTO APOIADO: não é possível remover usuários que sejam autores de avisos

**Tarefa T2: Efetuar login**

Cenários associados: 5



T2.A.1 Informar login

SIGNOS: usuário.login?

TRATAMENTO APOIADO: login incorreto

T2.A.2 Informar senha

SIGNOS: usuário.senha?

TRATAMENTO APOIADO: senha inválida

T2.B.1 Trocar senha

SIGNOS: usuário.senha?

TRATAMENTO APOIADO: senha não confere

## APÊNDICE C – CONSTRUÇÃO DO MODELO DE INTERAÇÃO

O modelo de tarefas descreve a estrutura das tarefas que serão apoiadas pelo sistema. No entanto, é insuficiente para dar forma à solução computacional tal como esta será percebida pelo usuário. Para isto, a partir do modelo de tarefas, deve-se projetar a interação. Na modelagem das tarefas, cada meta e suas tarefas são modeladas separadamente, não incluindo a “relação” entre as metas. Por exemplo, na modelagem de tarefas do sistema de Quadro de Avisos, existem as metas Consultar avisos e Remover avisos. Como foi visto no Apêndice B, a modelagem destas metas e suas tarefas é feita separadamente, ou seja, em diagramas distintos. Mas é importante, em algum momento do processo de *design*, identificar a necessidade destas metas estarem disponíveis em um mesmo momento de interação. A “proximidade” entre estas metas facilita o uso do sistema, pois para um usuário que estiver buscando um aviso para ser removido, é mais prático que ele possa remover o aviso procurado assim que ele seja encontrado. Isto evita de ele ter que sair da busca e ir em outro local na interface, para remover o mesmo aviso encontrado. Portanto, antes de se especificar a interface propriamente dita, a partir do modelo de tarefas deve-se projetar a interação de forma cuidadosa.

O modelo de interação que será apresentado neste Apêndice é fundamentado na teoria da Engenharia Semiótica [Paula, 2003]. Esta teoria vê a interação como uma conversa entre o usuário e um representante do *designer* que está cristalizado na interface (preposto do *designer*). Como a interação é vista como uma conversa, as ferramentas de *design* de IHC devem apoiar o *designer* no projeto de todas as conversas que podem acontecer quando o usuário utiliza a aplicação.

Para modelar estas conversas, [Paula, 2003] propõe um modelo diagramático de interação, complementado por especificações textuais. O objetivo do modelo diagramático de interação é dar uma visão global aos *designers* do discurso interativo como um todo.

Os elementos básicos do diagrama de interação são cenas, processos do sistema e transições. Uma **cena** representa uma conversa sobre um determinado assunto, ou **tópico**. Esta conversa é composta de um ou mais diálogos, que por sua vez são compostos de falas do usuário e do preposto, organizadas como pares conversacionais. A grosso modo, pode ser vista como o palco onde a conversa usuário–preposto ocorre. Uma cena poderá ser mapeada, no modelo de interface, por exemplo, em um componente estruturado tal como uma tela, janela ou página.

Uma cena é representada por um retângulo com bordas arredondadas e uma identificação do seu tópico, expresso preferencialmente por um verbo no modo infinitivo e do ponto de vista do usuário. O(s) diálogo(s) que a compõe(m) podem ser identificados na parte inferior da cena, entre colchetes, e separados do nome da cena por uma linha horizontal (Figura 18). Os diálogos indicam o que o usuário e o preposto podem dizer sobre o tópico da conversa.

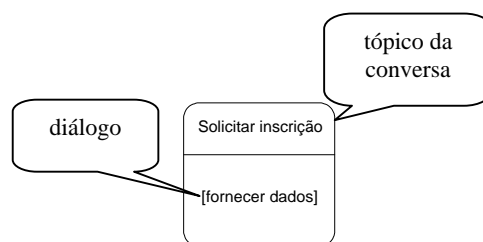


Figura 18: Cena Solicitar inscrição.

Uma cena especial é a que contém apenas uma fala do preposto. Isto é, uma fala resultante de um processamento, que foi emitida em uma cena em resposta a uma fala do usuário (que causou o processamento). Por exemplo, a cena que contém somente a seguinte fala do preposto: “*Aguarde email do administrador.*”, em resposta a um pedido de inscrição do usuário (Figura 19).

Os **processos do sistema** emitem as falas do preposto do *designer* que afetam o rumo que a conversa pode tomar, como resposta a uma ou mais falas do usuário. Em outras palavras, representa-se um processo quando houver mais de um caminho de interação possível a partir de uma cena, e a decisão sobre que caminho tomar depende de um processamento interno do sistema, revelado apenas pela fala do preposto do *designer* que dele resulta. Portanto, os processos só precisam ser representados quando seu resultado precisa ser comunicado ao usuário. Na prática, isto ocorre sempre que há mais de uma transição possível entre a cena corrente e a(s) próxima(s), e a escolha dentre estas transições depende de um processamento, e não da decisão direta do usuário. Um processo é representado por uma caixa preta (Figura 19). Usou-se a noção de “caixa preta” pois o usuário não percebe diretamente o que está ocorrendo em um processamento do sistema, só percebe através de falas do preposto. Esta representação visa motivar o *designer* a projetar cuidadosamente as falas do preposto a serem emitidas após cada processamento, como única forma de comunicar ao usuário o que/como ocorreu, e explicar por quê.

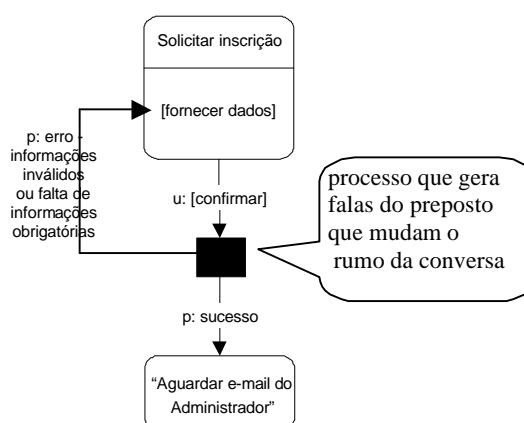


Figura 19: Cenas, processo do sistema e transições.

Uma **transição** representa uma mudança de rumo na conversa, seja causada por uma fala do usuário (a partir de uma cena), ou por uma fala do preposto (a partir de um processamento). Uma transição é expressa por uma seta preta indicando a direção da

transição e um rótulo. Opcionalmente, quando há a possibilidade de se voltar à cena de origem, a transição pode ter uma seta branca indicando esta possibilidade. Isto é, a transição terá em uma extremidade a seta preta indicando a direção da transição e na outra, a seta branca, indicando a possibilidade de se voltar à cena de origem. O rótulo de uma transição é composto de até três partes (Figura 20):

1. **pré-condições:** condições que devem ser satisfeitas para que o usuário ou preposto possa enunciar a fala correspondente. São precedidas pela expressão *pré:*. Podem ser expressas em linguagem natural, através de expressões lógicas ou alguma outra notação de preferência do *designer*. Por exemplo, *pré: o usuário deve ter permissão para alterar o aviso*.
2. **uma ou mais falas do usuário ou do preposto do designer.** No caso do usuário, trata-se do(s) enunciado(s) do usuário que causa(m) a transição. As falas do usuário são expressas no formato *u:[fala do usuário]*. Por exemplo, *u:[confirmar alteração]*. No caso do preposto, trata-se da(s) fala(s) que revela(m) o(s) resultado(s) de algum processamento do sistema que causa(m) a transição. As falas do preposto são expressas no formato *p:fala do preposto*. Por exemplo, *p: erro - login inexistente*.
3. **pós-condições:** condições que passam a ser verdadeiras durante a transição. São precedidas pela expressão *pós:*. Por exemplo, *pós: login efetuado*.

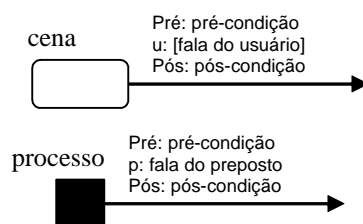


Figura 20: Representação de uma transição.

Algumas cenas podem ser acessadas (a conversa descrita na cena poderá ocorrer) em qualquer ponto da aplicação, ou seja, a partir de qualquer outra cena. O acesso a estas cenas, chamado **ubíquo**, é marcado por uma transição que sai de uma cena preenchida pela cor cinza (por exemplo, o acesso a cena *Efetuar login*, na Figura 21). Se houver pré-condições para se acessar a cena, deve-se indicá-las na transição que dá acesso à cena (por exemplo, a *pré: \_login=F* na transição que dá acesso à cena *Efetuar login*).

Com frequência, as possibilidades de interação, bem como o comportamento de cada cena, variam conforme o estado atual da aplicação. Quando isto ocorre, podemos definir **signos de controle** da interação cujos valores compõem este estado. No exemplo da Figura 21, para poder acessar a cena *Efetuar login*, o usuário não pode estar “logado” no sistema (*\_login=F*). Ao efetuar o login com sucesso, este signo de controle recebe o valor *true* (*\_login ← T*), indicando que houve uma mudança de estado da aplicação. Estes signos de controle podem ser acrescentados à tabela de signos descrita no Apêndice A, com o tipo *controle*. Os signos de controle não necessariamente aparecerão na interface, mas são necessários para o entendimento e tomada de decisão da equipe de *design*.

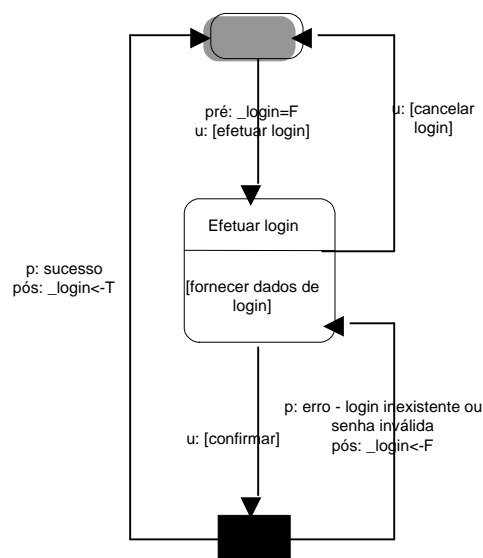


Figura 21: Parte do diagrama de interação referente à tarefa *Efetuar login*.

A correspondência entre cenas do modelo de interação e tarefas e metas do modelo de tarefas pode ser feita utilizando-se um retângulo de cor cinza representando a(s) tarefa(s) ou meta(s) que engloba(m) a(s) cena(s). A identificação desta(s) tarefa(s) ou meta(s) pode ser expressa no canto inferior esquerdo do retângulo (Figura 22). Este elemento é utilizado para aumentar a rastreabilidade entre os modelos, facilitando a avaliação do impacto de mudanças corretivas ou evolutivas que possam ocorrer, bem como a manutenção da consistência entre modelos, a cada nova versão.

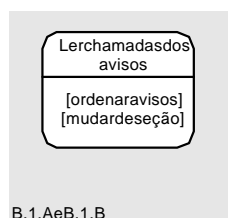


Figura 22: Correspondência entre tarefas/metabol e cenas.

As possíveis rupturas na comunicação usuário-preposto do *designer* detectadas na modelagem de tarefas (Apêndice B) devem aparecer nas conversas projetadas pelo *designer* no modelo de interação. Por exemplo, o *TRATAMENTO APOIADO: o critério deve ser informado* detectado na especificação textual da tarefa *Fornecer critério* para uma busca, deve ser previsto em algum diálogo, como fala do preposto. Esta fala deve ser seguida por uma fala do usuário, que poderá remediar o seu “erro”, informando um critério. Novas rupturas na comunicação podem ser detectadas na modelagem da interação.

As decisões de *design* que são tomadas durante a construção do modelo de interação precisam, com frequência, ser apresentadas aos usuários para sua apreciação. Para isto, pode ser necessário gerar cenários que reflitam as alternativas de solução propostas, para facilitar a discussão com os usuários.

Como complementação do diagrama de interação, é necessário definir em detalhes cada diálogo, incluindo os signos envolvidos, os tipos de elementos de interface



correspondentes (texto livre, escolha simples, escolha múltipla, rótulo, etc.) e informações adicionais como obrigatoriedade de signos e presença de valores *default*. A partir deste detalhamento, torna-se mais direta a construção dos *storyboards* correspondentes a cada cena.

Sendo assim, o diagrama de interação é complementado com uma especificação textual. Os signos que aparecem na especificação textual do modelo de interação são os mesmos que estão representados no modelo de tarefas. Enquanto no modelo de tarefas eles aparecem ligados a uma tarefa, no modelo de interação eles aparecem como falas de usuário ou preposto, contextualizadas pelos diálogos e cenas que compõem.

A Figura 24 apresenta a especificação textual da cena *Buscar Avisos* (Figura 23). No diálogo *[fornecer critério]*, o signo envolvido é *critério*. Como este critério será fornecido pelo usuário, seu tipo é definido como *<texto livre>*. Além disto, é obrigatória a informação deste signo pelo usuário. Na especificação textual isto é representado da seguinte forma: *<texto livre: obrigatório>*.

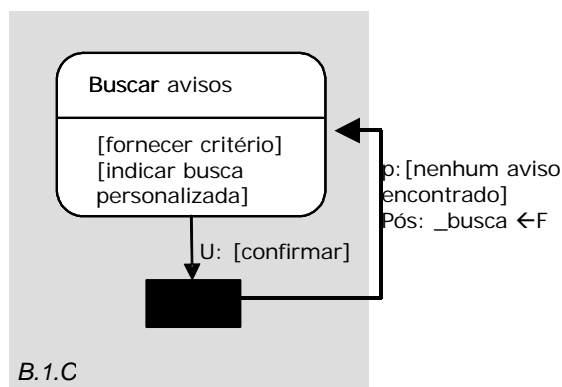


Figura 23: Solução de interação para indicar resultado negativo de busca (nenhum aviso encontrado).

**Cena: Buscar Avisos**

A [fornecer critério]

critério? <texto livre: obrigatório>

B [indicar busca personalizada]

busca\_personalizada? <escolha simples: obrigatório>

Figura 24: Especificação textual da cena *Buscar Avisos*.

O objetivo do modelo diagramático de interação é dar uma visão global aos *designers* do discurso interativo como um todo. A especificação textual fornece os detalhes de cada conversa. O modelo de interação facilita a reflexão do *designer* sobre suas tomadas de decisão, pois ele consegue se imaginar tanto no papel do usuário quanto no do seu preposto, ao projetar ou verificar as falas de cada um, nos atos comunicativos.

### Parte do Modelo de Interação do Quadro de Avisos

