



**Guia para Modelagem de Interações  
Metodologia CELEPAR**

**Agosto 2009**

## Sumário de Informações do Documento

Documento: guiaModelagemInteracoes.odt

Número de páginas: 17

Versão	Data	Mudanças	Autor
1.0		<i>criação</i>	<i>Danielle Mayer</i>
1.0	21/08/09	<i>Revisão</i>	<i>Danielle Mayer, Marcos Chiarello, Cleverson Budel, Willian Medeiros</i>
1.0	16/11/2009	<i>Alteração:</i>  <i>1 – Indicação de detalhamento de campo na Fase de Análise, conforme solicitado durante a primeira turma de treinamento.</i>  <i>2 - Alterando exemplos com a correção após a primeira turma de treinamento.</i>	<i>Danielle Mayer</i>

## Sumário

1	Introdução.....	4
1.1	Visão Geral .....	4
2	Diagrama de seqüência.....	5
2.1	Notação Básica.....	5
3	Contexto e nível de abstração - Fase de Análise.....	7
3.1	Classes de Tela.....	8
3.2	Diagrama de Seqüência – Caso de Uso.....	10
4	Contexto e nível de abstração - Fase de Projeto .....	13
4.1	Classes de Tela .....	13
4.2	Diagrama de Seqüência - Caso de Uso.....	14

# 1 INTRODUÇÃO

Este guia tem por objetivo orientar a atuação do Analista de Sistemas ao realizar a modelagem dos aspectos dinâmicos do sistema, considerando sua necessidade sinalizada pela metodologia CELEPAR.

## 1.1 Visão Geral

A modelagem de aspectos dinâmicos do sistema é realizada pela utilização de interações. Uma interação é um comportamento que compreende um conjunto de mensagens trocadas entre um conjunto de objetos em determinado contexto para a realização de um propósito.

Os diagramas de seqüência e os diagramas de comunicação – chamados de diagramas de interação – são diagramas definidos na UML (*Unified Modeling Language*) para a modelagem dos aspectos dinâmicos de sistema.

A modelagem de cada interação pode ser feita de duas maneiras: dando-se ênfase à ordem temporal das mensagens (diagrama de seqüência) ou dando-se ênfase à seqüência das mensagens no contexto de alguma organização estrutural de objetos (diagrama de comunicação). Como ambos são derivados das mesmas informações de um metamodelo da UML, o diagrama de seqüência e o diagrama de comunicação são semanticamente equivalentes e podem ser convertidos um no outro sem qualquer perda de informação. Entretanto, isso não significa que os dois diagramas permitem a visualização das mesmas informações explicitadamente, cada qual fornece uma abstração distinta do contexto do cenário que está representando.

Diagramas de interação podem ser utilizados para fazer a modelagem de um determinado fluxo de controle de um caso de uso ou podem aparecer sozinhos para visualizar, especificar, construir e documentar a dinâmica de uma determinada sociedade de objetos. Estes diagramas não são importantes somente para a modelagem de aspectos dinâmicos do sistema, mas também para a construção de sistemas executáveis por meio de engenharia de produção.

No contexto do Processo de Desenvolvimento da CELEPAR será considerado o diagrama de seqüência para a representação dos aspectos dinâmicos do sistema, seguindo a notação definida pela UML (*Unified Modeling Language*) e a padronização do contexto e nível de abstração a ser

atingido em cada fase, objeto deste guia.

## 2 DIAGRAMA DE SEQUÊNCIA

Diagramas de sequência serão utilizados para a confecção da interface do sistema, para descrever como o sistema irá responder aos estímulos do usuário e para a confecção de plano de testes.

### 2.1 Notação Básica

A notação UML(*Unified Modeling Language*) para diagrama de sequência envolve um conjunto de objetos envolvidos em um processo (cenário) e a especificação de mensagens trocadas entre eles ao longo da linha de vida. Esta notação é composta pelos seguintes elementos:

#### 1. Atores

Instância dos atores já representados na modelagem de casos de uso.

#### 2. Objetos

Representam as instância das classes envolvidas no fluxo de controle ilustrado pelo Diagrama de Sequência.

Um objeto pode existir no início do processo, e aparecerá na parte superior do diagrama, ou criado em tempo de execução do mesmo, e aparecerá na mesma altura que a mensagem que o criar for chamada.

#### 3. Linha de vida

É representada por linhas finas verticais tracejadas partindo do objeto e indica o tempo em que um objeto existe durante um processo,

#### 4. Ocorrência de execução

São representados dentro da linha de vida de um objeto por uma linha mais espessa. Sua finalidade é indicar os períodos em que um objeto está participando ativamente do processo.

#### 5. Mensagens ou estímulos

Nos diagramas de sequência, uma mensagem é um elemento de modelo (representado por uma seta) que define um tipo específico de comunicação entre participantes de uma

interação.

As mensagens são utilizadas para demonstrar a comunicação entre:

- Um ator e outro ator;
- Um ator e um objeto, onde um ator produz um evento que dispara um método em um objeto;
- Um objeto e um ator, ocorre quando um objeto envia uma mensagem em resposta à um método solicitado;
- Um objeto e um objeto, onde um objeto transmite uma mensagem para outro objeto em geral solicitando a execução de um método.
- Um objeto e si próprio, onde um objeto envia uma mensagem para si mesmo o que é conhecido como autochamada;

As mensagens são classificadas em síncronas e assíncronas:

- **mensagem síncrona:** representa que o objeto que enviou a mensagem aguarde a conclusão do processamento da mensagem realizada pelo objeto destino antes de continuar seu fluxo de execução.
- **mensagem assíncrona:** representa que o objeto que enviou a mensagem prossegue sua execução independentemente do tratamento da mensagem realizada pelo objeto destino.
- **mensagem de criação:** representa a criação de um participante em uma interação.
- **mensagem de destruição:** representa a destruição de um participante em uma interação.

## 6. Fragmentos de Interação e Ocorrência de Interação

Fragmento de interação são noções abstratas de unidades de interação geral, ou seja, é o quadro que representa a própria interação.

Ocorrência de interação é uma interação referenciada em outro quadro de interação. É possível utilizar as ocorrências de interação para obter o conteúdo comum de uma interação e reutilizá-lo em outro diagrama de sequência.

## 7. Fragmentos Combinados e Operador de Interação

Fragmento Combinado é representado por um retângulo que determina a área de abrangência do diagrama permitindo uma modelagem semi-independente da parte do diagrama que permite expressar questões de testes se-então, laços ou processamentos paralelos através dos operadores de interação incluídos a ele.

O fragmento combinado conterá operadores de interação que identificam o tipo do fragmento que está sendo modelado.

Operadores mais utilizados:

- **Alt – Alternativas:** Este operador define que o fragmento representa uma escolha entre dois ou mais comportamentos;
- **Opt – Opção:** Este operador define que o fragmento representa uma escolha de comportamento onde este comportamento será ou não executado;
- **Par – Paralelo:** Este operador define que o fragmento representa uma execução paralela de dois ou mais comportamentos;
- **Loop – Laço:** Este operador define que o fragmento representa um laço que poderá ser repetido diversas vezes;
- **Break – Quebra:** Este operador define que o fragmento indica uma “quebra” na execução normal do processo. É utilizado para modelar tratamento de exceções;
- **Critical - Região Crítica:** Este operador define que o fragmento indica uma operação atômica que não pode ser interrompida por outro processo até ser totalmente concluída.

Operadores menos utilizados: **Neg - Negativo, Assertion – Afirmação, Ignore – Ignorar, Consider – Considerar, Seq – Seqüência Fraca, Strict – Seqüência Estrita.**

### 3 CONTEXTO E NÍVEL DE ABSTRAÇÃO - FASE DE ANÁLISE

Durante a fase de análise os diagramas de seqüência serão utilizados para modelar a definição do sistema focalizando o desdobramento das intenções do usuário, já detalhados pela especificação de caso de uso realizada no detalhamento dos requisitos, em relação às interfaces

necessárias (UI – user interface).

Neste contexto os diagramas de seqüência serão compostos por objetos atores (instâncias de atores) e objetos tela (instância de classes de telas).

### 3.1 Classes de Tela

Tem por finalidade representar telas (UI – user interface) utilizadas para operacionalizar o funcionamento de cada caso de uso através dos diagramas de seqüência.

Padronização Sugerida:

1. **Classe Tela:** Cada tela deverá ser representada por uma classe com o estereótipo *<<Frenteira>>* do perfil (RUPAnalysis).
2. **Representação de Campo:** Cada campo deverá ser representado como atributo da classe.
  - **Nome do Atributo:** O nome do atributo deve ser o nome desejado para apresentação na tela do sistema, inclusive com espaçamento, caixa alta, caixa baixa, acentos e caracteres especiais desejados.
  - **Tipo do Atributo:** tipo definido de acordo com os seguintes valores:

combobox  
password  
link  
textArea  
text  
checkbox  
multibox  
radio  
plaintext  
button  
image  
file

### 3. Tela com conteúdo do tipo Tabela/Árvores/Abas:

Poderão ser representados como uma nova classe, com estereótipo *<<Frenteira>>*,



associada à classe de tela pai (associação de composição).

#### 4. Detalhes de cada campo: Preencher na propriedade documentação de cada campo os detalhes que **forem relevantes** à confecção da interface gráfica.

**Obrigatório:** Sim / Não

**Situação:** Habilitado/Desabilitado

**Tamanho:** 50 ( número de caracteres,)

**Máscara:** 999999999-99

**Validação:** Utilizar validação para CPF. Caso CPF inválido apresentar a mensagem:

AVISO - “CPF inválido”.

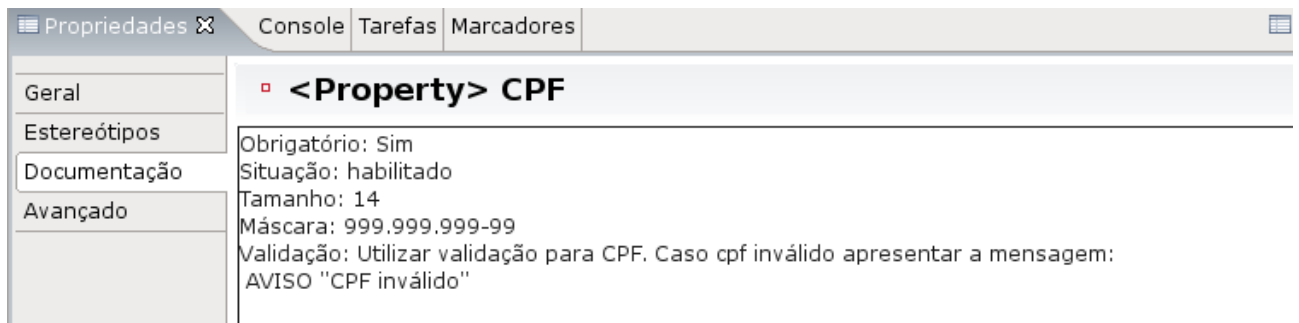


Figura 1: Documentação de Campo

#### 5. Exemplo – Diagrama de Classes de Tela:

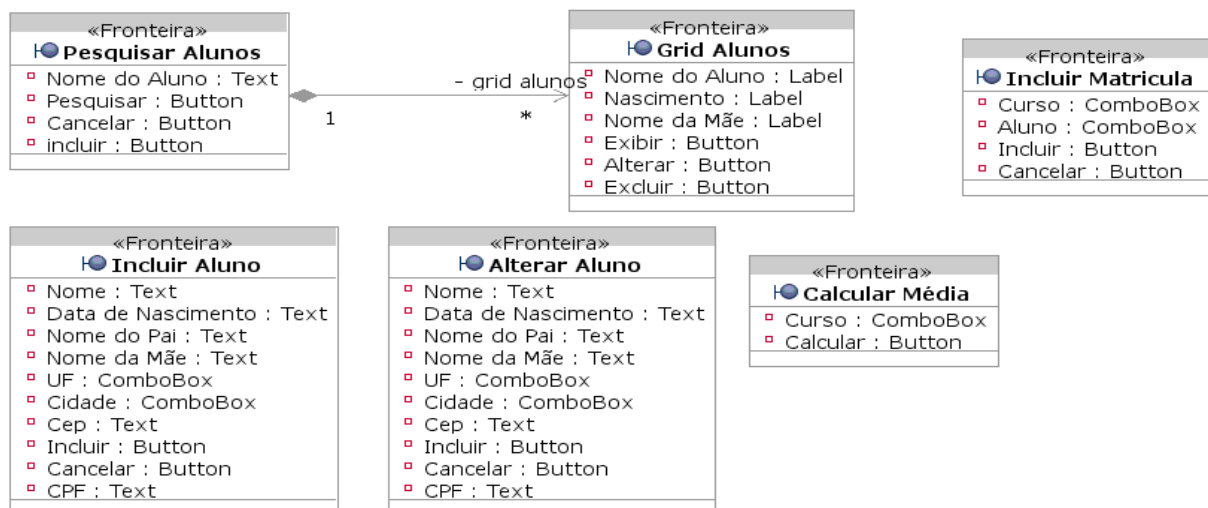


Figura 2: Classes de Tela

### 3.2 Diagrama de Seqüência – Caso de Uso

O objetivo do diagrama de seqüência neste momento será demonstrar a interação entre o(s) atore(s) e o sistema e promover a prospecção da interface do sistema (protótipo).

Padronização sugerida para cada cenário de caso de uso:

- 1. Representação da comunicação entre ator e o sistema:** A comunicação dar-se-á entre a instância do ator e a instância da respectiva interface (tela identificada) e vice versa, através de mensagens que representem eventos, ou seja, mensagens não vinculadas à métodos.  
  
Nesta fase, não deverão ser representados regras de negócio e operações com elementos de tela como Habilitação/Desabilitação (Ocultaç o/Apresentaç o).
- 2. Cen rio de caso de uso representado por mais de uma tela:** A tela que ativar  a pr xima deve enviar uma mensagem (evento)   tela destino.
- 3. Nomenclatura das mensagens:** O nome das mensagens que representam eventos devem com car com o caracter de barra invertida “\”.
- 4. Chamadas   outros casos de uso:** A chamada   dar-se-  pela adi  o do fragmento combinado **ocorr ncia de intera  o** na respectiva tela do caso de uso invocador.
- 5. Exemplo:**

### Manter Alunos : Pesquisar

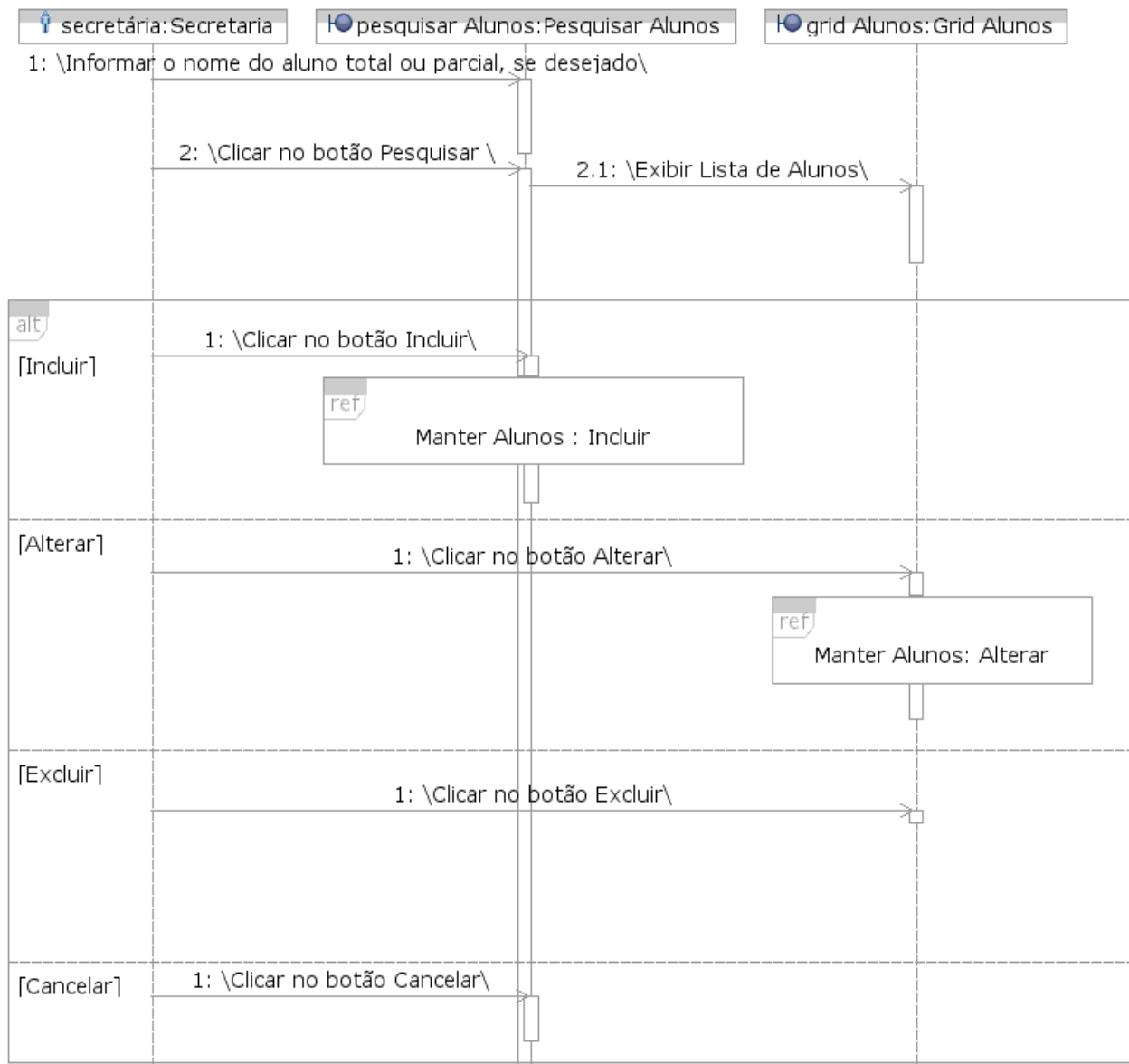


Figura 3: Diagrama de Seqüência / Cenário Pesquisar - Fase Análise

## 4 CONTEXTO E NÍVEL DE ABSTRAÇÃO - FASE DE PROJETO

Durante a fase de projeto os diagramas de sequência, elaborados na fase de análise, deverão ser refinados para descrever como o sistema irá responder aos estímulos do usuário levando em consideração a **arquitetura lógica adotada**. Para maiores detalhes sobre a arquitetura consultar o **Guia para Modelagem de Classes de Projeto**.

### 4.1 Classes de Tela

Na fase de projeto o Diagrama de Classes de Tela pode ser complementado conforme a padronização abaixo sugerida:

1. **Mensagens de obrigatoriedade de preenchimento:** Para documentar este tipo de mensagem utilize a propriedade **Documentação** da referida classe de tela.

Exemplo:

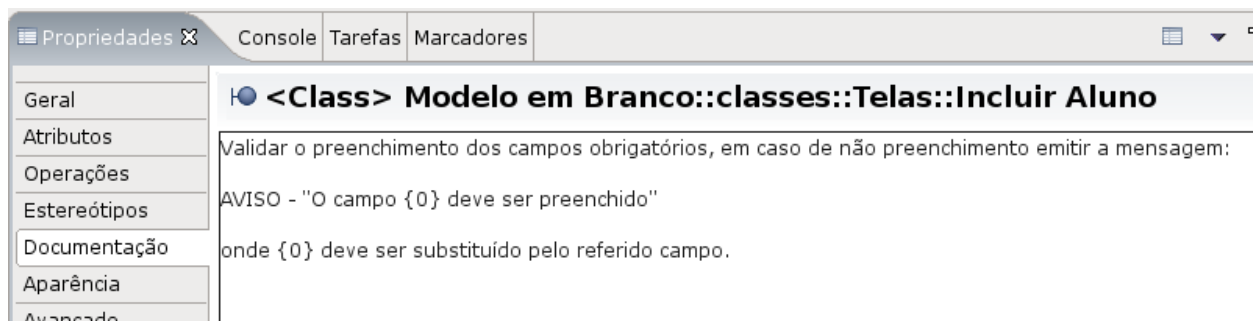


Figura 4: Documentação da Classe de Tela

2. **Detalhes de cada campo:** Preencher na propriedade documentação de cada campo os detalhes que forem relevantes, inclusive mensagens de validação específicas do campo, como por exemplo:

**Obrigatório:** Sim / Não

**Situação:** Habilitado/Desabilitado

**Tamanho:** 50 ( número de caracteres,)

**Máscara:** 999999999-99

**Validação:** Utilizar validação para CPF. Caso CPF inválido apresentar a mensagem:

AVISO - “CPF inválido”.

Exemplo:

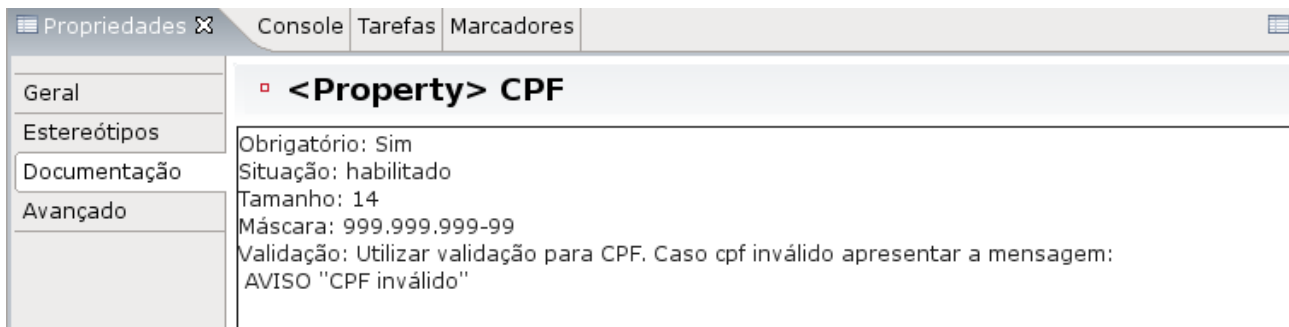


Figura 5: Documentação de Campo

## 4.2 Diagrama de Seqüência - Caso de Uso

O objetivo do diagrama de seqüência neste momento será demonstrar como o sistema irá responder aos estímulos do usuário levando em consideração a **arquitetura lógica adotada**.

Padronização sugerida para cada cenário de caso de uso:

1. Para cada diagrama deve ser adicionado a respectiva instância da classe <<Controle>> e a instância da classe < Serviço>>.
2. **Inicialização:** Normalmente a primeira mensagem do diagrama do respectivo cenário poderá ser uma mensagem auto-chamada na instância da classe controle para chamar a operação que será responsável pela ativação da tela inicial. Para fins de padronização pode-se nomear a operação como *carregarNome da Tela*.
3. **Representação de chamada de operações do sistema:**

A ativação de validações deverão ser representadas como mensagens (chamada de operações) enviadas da instância da **tela** para a instância da **classe de controle**, da instância da classe de **controle** para a instância da classe de **serviço**, da instância da classe de serviço para a instância das classes de domínio e persistência.

Resumidamente:

- Tela → Controle;
- Controle → Serviço;
- Serviço → Domínio e Persistência.

**Observação:** É aconselhável que todas as chamadas a métodos estejam explícitas no diagrama de sequência. Desta forma, alterações na sua assinatura são efetivadas automaticamente em todos os diagramas do projeto.

- 4. Representação do resultado de operações do sistema:** Poderá ser expresso com o envio de mensagens ao ator ou à (s) tela(s) conforme a necessidade de manipular componentes, como Habilitar/Desabilitar (Ocultar/Apresentar) e preenchimento de componentes em tempo de execução.

Normalmente estas mensagens estarão agregadas à fragmentos combinados e operadores de interação(alternativas, opção, paralelo, laço, quebra, região crítica, negativo, afirmação, ignorar, considerar, sequência fraca, sequência estrita) segundo a definição da UML.

- **Envio de mensagens ao ator:** A mensagem que representar o resultado partirá da instância da classe de controle para a referida instância da tela que irá apresentar esta mensagem ou diretamente à instância do ator se for uma mensagem *popup*.
  - **Habilitar/Desabilitar (Ocultar/Apresentar) de componentes de tela:** Representada por mensagens disparadas da instância da classe controle para a instância da classe de tela.
  - **Preenchimento de Campos em tempo de execução:** Representada por mensagens disparadas da instância da classe controle para a instância da classe de tela.
- 5. Cenário de caso de uso representado por mais de uma tela:** A mensagem (evento) à tela destino, que na fase de análise era disparada por uma outra tela, deve ser disparada pela instância da classe de controle. Para fins de padronização sugere-se preencher o texto “Exibir” como nome da mensagem.
  - 6. Nomenclatura das mensagens:** O nome das mensagens que representam eventos devem estar entre barras invertidas “\”.
  - 7. Chamadas à outros casos de uso:** A chamada à outros casos de uso, que na fase de análise era representada pela adição do fragmento combinado **ocorrência de interação** na respectiva tela do caso de uso invocador, será representada pela adição na respectiva classe controle. O retorno do caso de uso chamado deve ser representado por uma nota vinculada

ao fragmento.

**8. Gerenciamento de Transações:** É aconselhável sinalizar, através do fragmento combinado Região Crítica, os trechos do diagrama de sequência onde é necessário o controle de transação. Desta forma, os métodos que exigem o controle de transação deverão ser invocados de dentro do fragmento.

## 9. Exemplos:

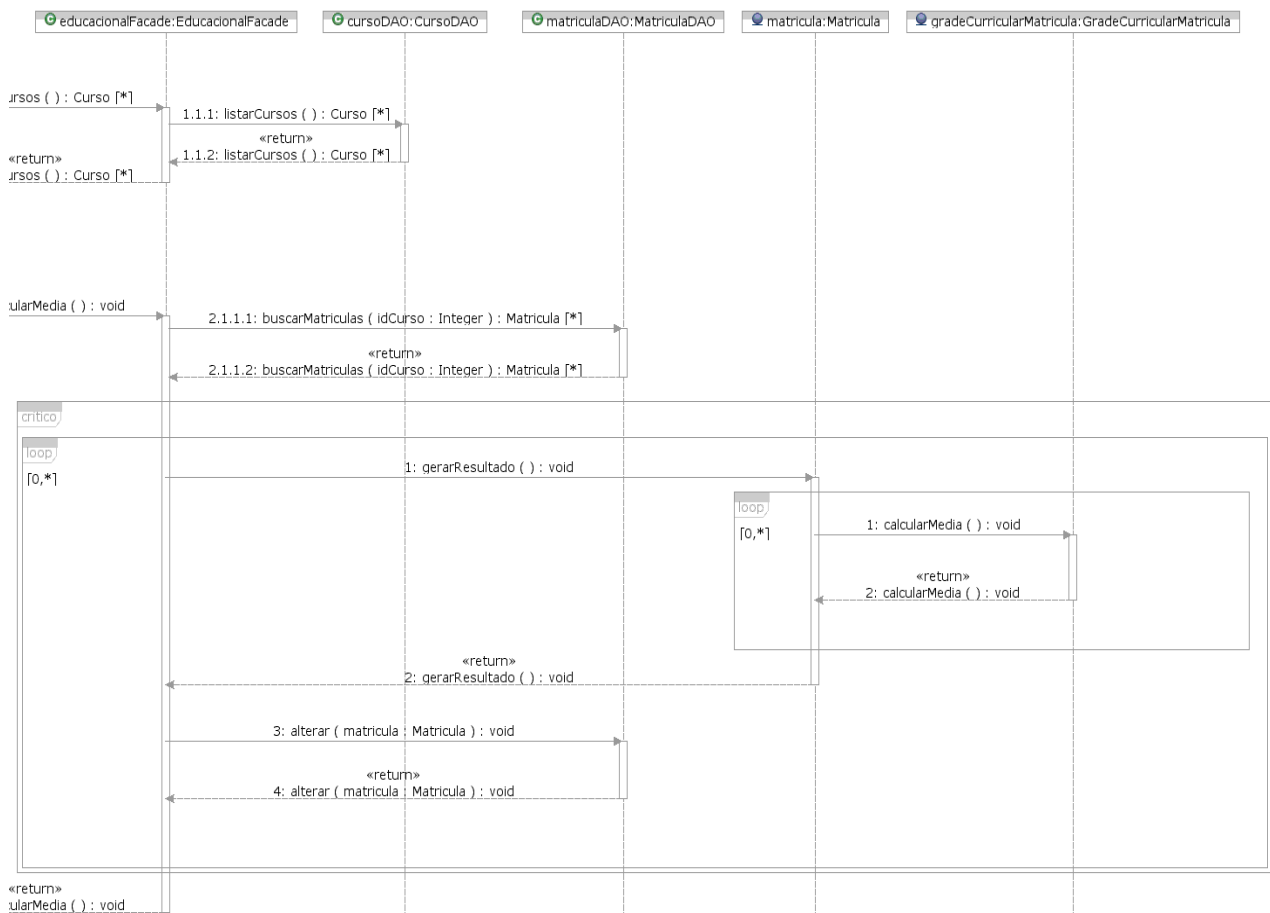


Figura 6: Uso do Fragmento combinado Região Crítica (Critico)

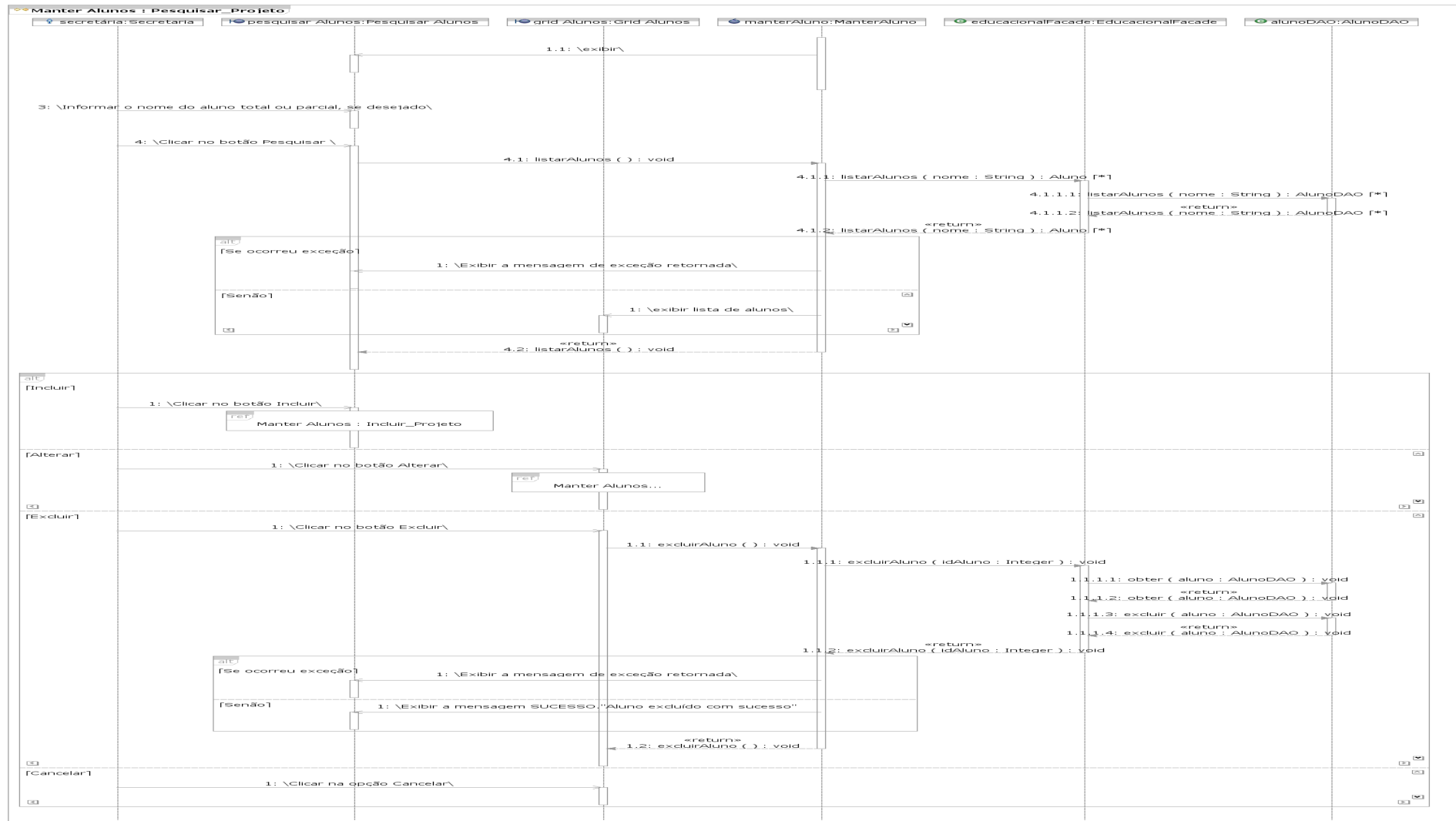


Figura 7: Diagrama de Seqüência / Cenário Pesquisar - Fase Projeto



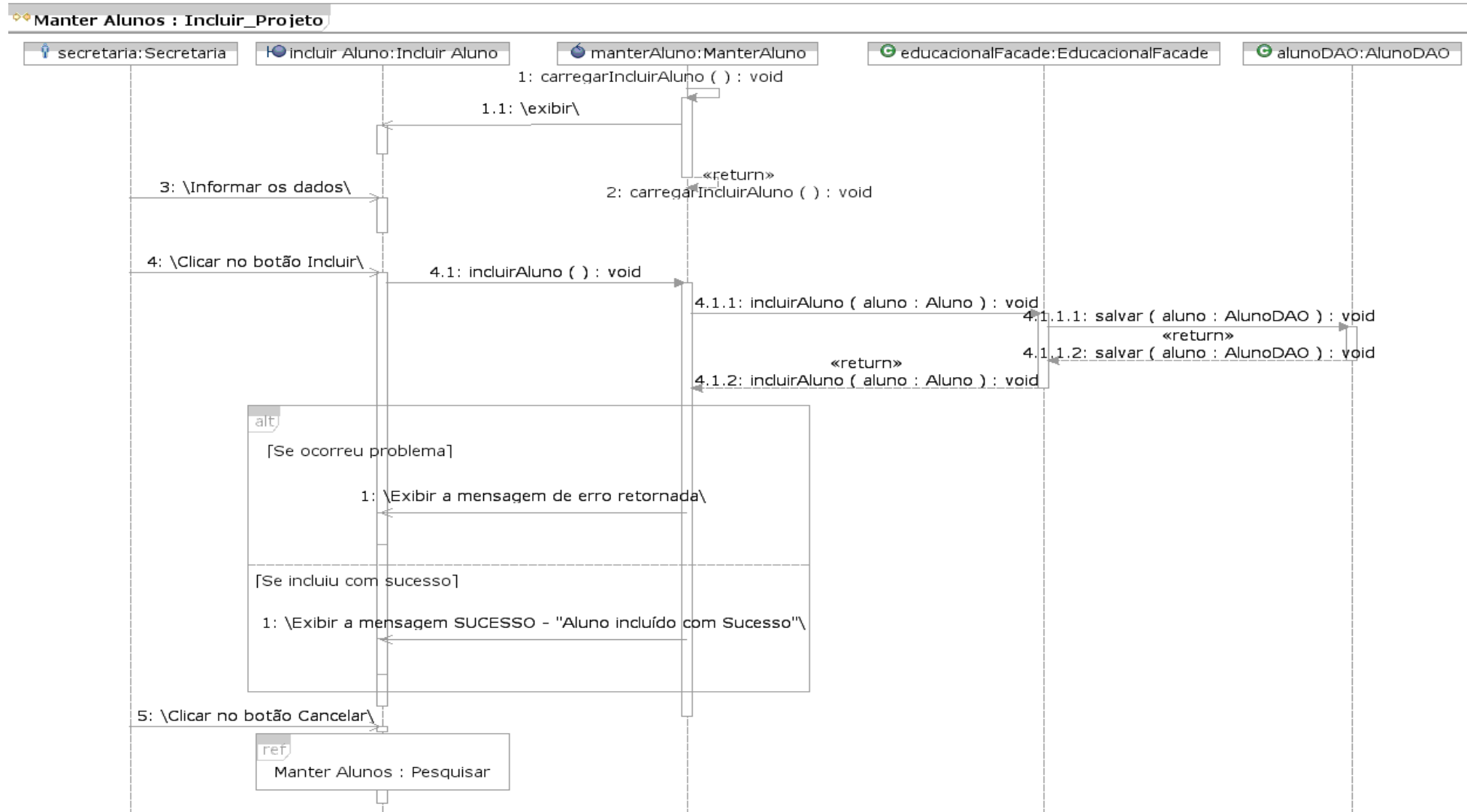


Figura 8: Diagrama de Seqüência / Cenário Incluir - Fase Projeto