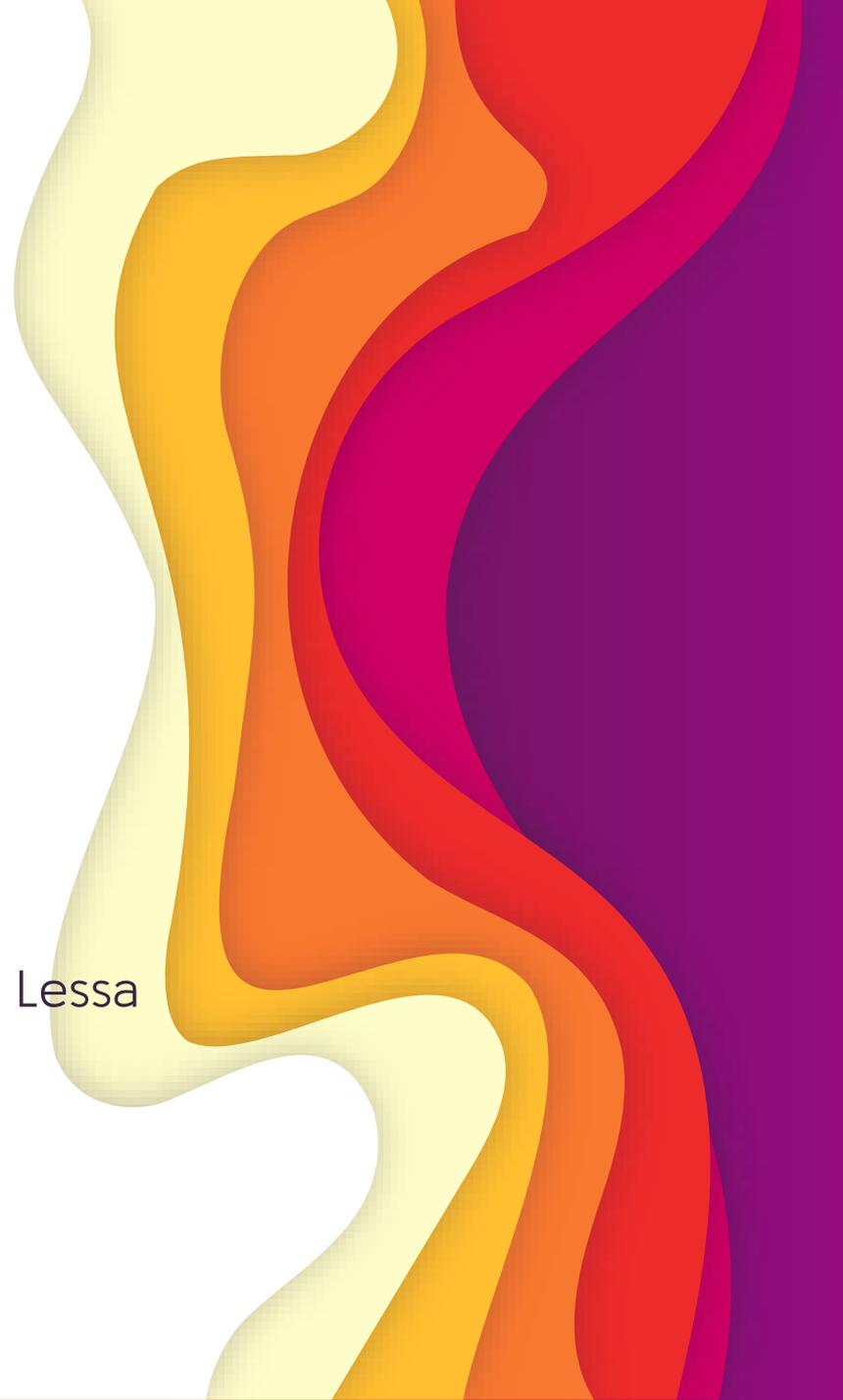


MODELAGEM DE SOFTWARE

Prof. Ricardo Ribeiro Assink
ricardo.assink@unisul.br

Prof. Rafael Lessa

Prof. Richard Henrique de Souza



Agenda

Modelagem UML e a Orientação a Objetos

- *Unified Modeling Language* – UML
- Digrama de sequencia

Unified Modeling Language - UML

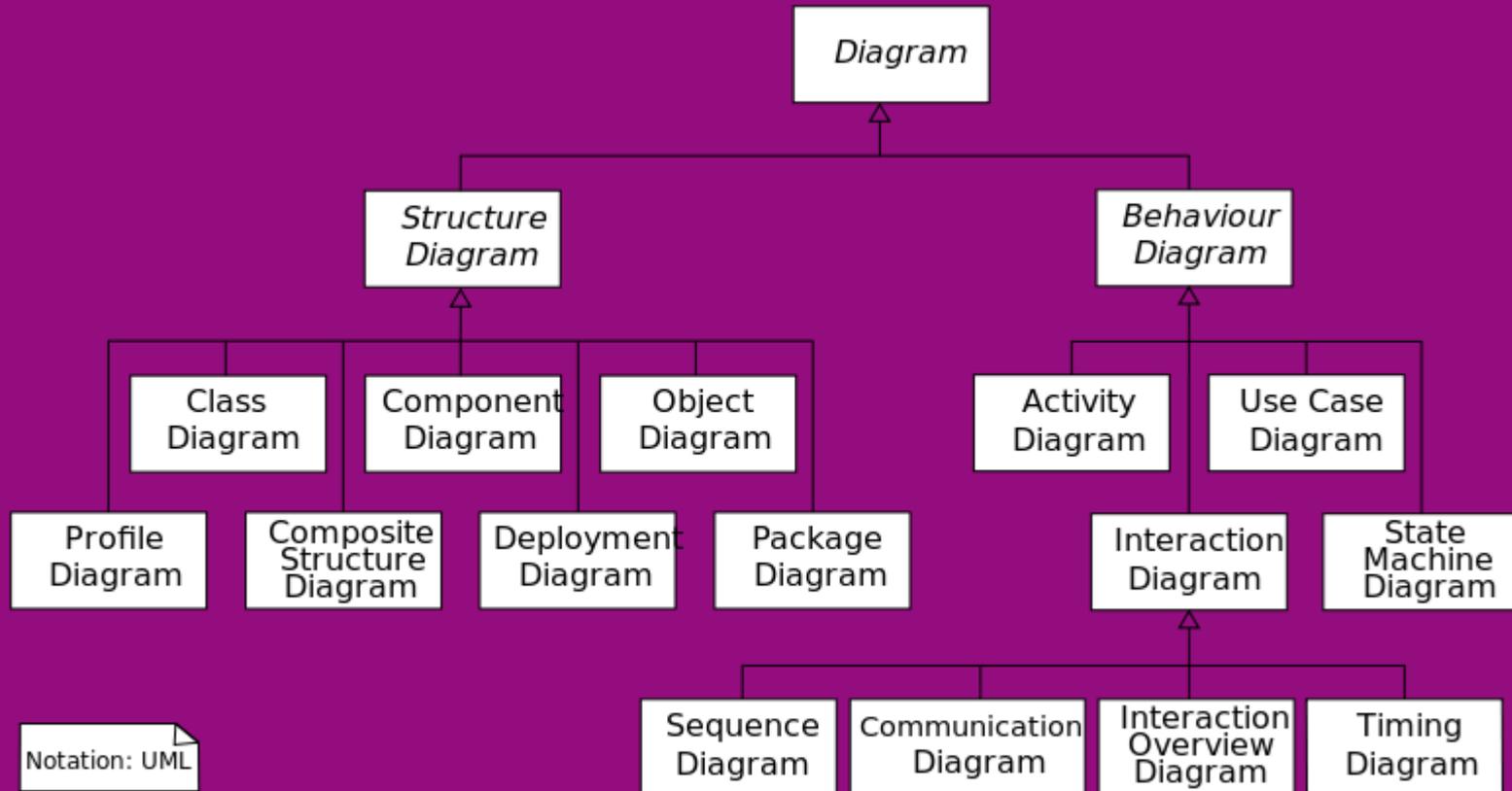
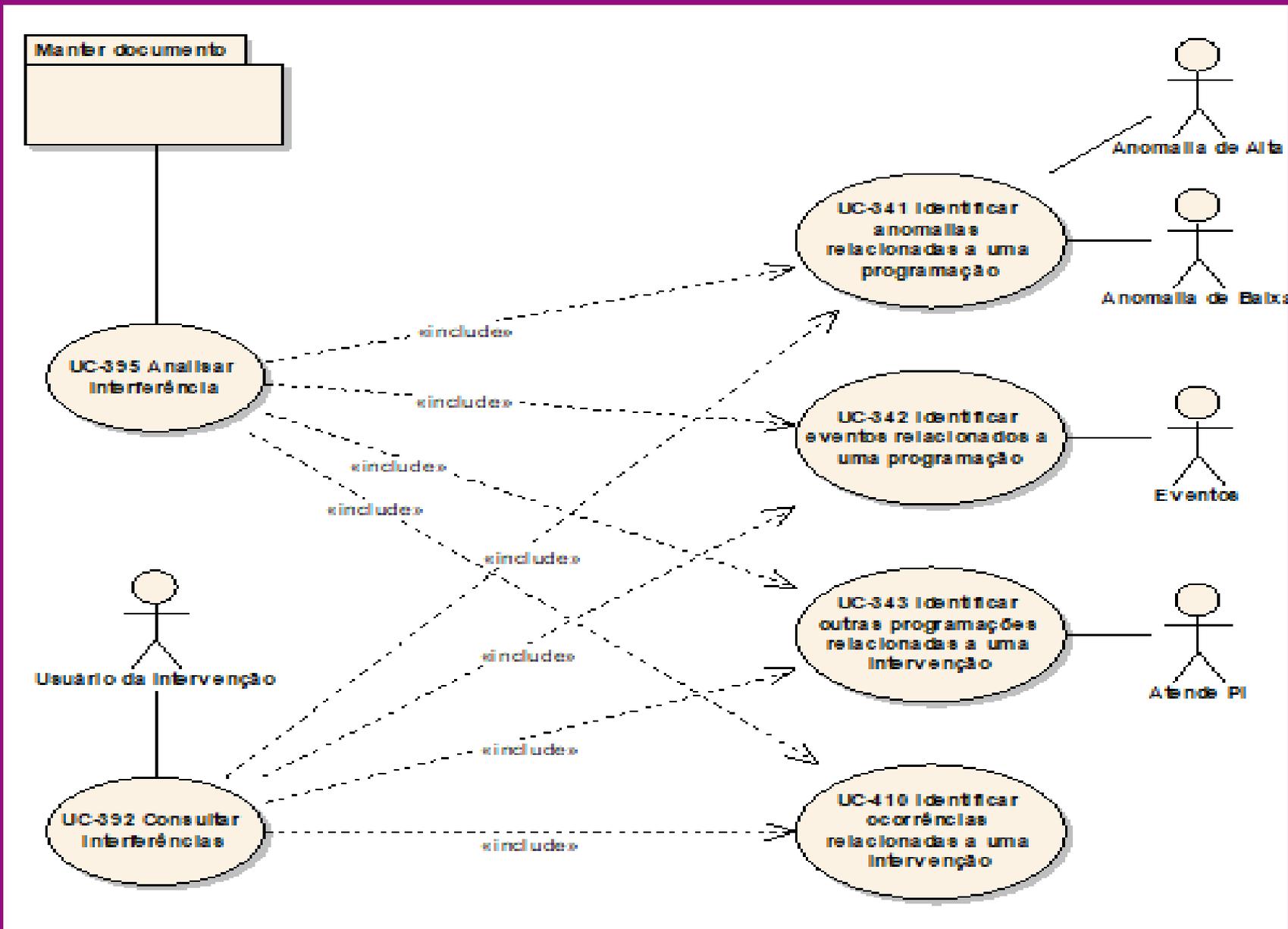


Diagrama de use case

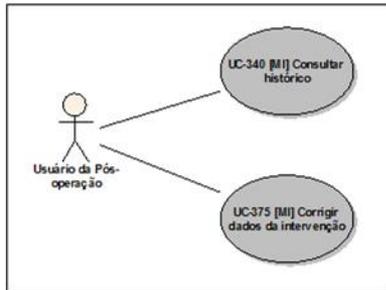


Exemplo de cenários

1. Casos de uso

Essa seção contém os casos de uso para gestão do histórico de programações do Sistema de Intervenções Operativas que serão implementados no Atende MI.

O diagrama a seguir mostra os casos de uso do sistema Atende MI que serão implementados em virtude do Sistema de Intervenções Operativas.



Atende MI

2. UC-340 - [MI] Consultar histórico

O usuário pesquisa as intervenções registradas no histórico segundo critérios de filtros. O usuário pode consultar os detalhes das intervenções, visualizando os dados dos documentos, arquivos anexados e programações de serviço.

2.1 Atores

Usuário da Pós-operação

Pessoa responsável por executar as funções de pós-operação.

2.2 Pós-condições

Lista de intervenções do histórico: O sistema apresenta a lista de intervenções do histórico.

2.3 Fluxo Básico

[B1] Consultar relação das intervenções

1. O usuário solicita consultar as intervenções do histórico;
2. O usuário fornece a combinação de um ou mais filtros (RQ-001-UC-340);
3. O sistema efetua a pesquisa das intervenções no histórico de acordo com o filtro fornecido;
4. O sistema apresenta a lista com as intervenções localizadas.

2.4 Fluxo Alternativo

[A3] Consultar programações de serviço

Caso no passo 4 do fluxo básico [B1] o usuário deseje consultar a relação das programações de serviço:

- 4.1 O usuário seleciona uma intervenção da lista de intervenções;
- 4.2 O usuário seleciona um documento dessa intervenção;
- 4.3 O usuário solicita consultar as programações de serviço;
- 4.4 O sistema mostra a relação das programações de serviço dessa intervenção relacionado ao documento selecionado.

[A4] Consultar trechos interrompidos

Caso no passo 4 do fluxo básico [B1] o usuário deseje consultar a lista de trechos interrompidos:

- 4.1 O usuário seleciona uma intervenção da lista de intervenções;
- 4.2 O usuário solicita consultar os trechos interrompidos;

Exemplo de cenários

4.3 O sistema mostra a relação dos trechos interrompidos.

[A5] Imprimir documento

Caso no passo 4.3 do fluxo alternativo [A1] o usuário desejar imprimir o documento:

4.4 O usuário solicita a impressão do documento;

4.5 O sistema envia o documento para a impressora.

[A1] Consulta dos detalhes de documento

Caso no passo 4 do fluxo básico [B1] o usuário deseje consultar detalhes de um documento específico:

4.1 O usuário seleciona uma intervenção da lista de intervenções;

4.2 O usuário seleciona um documento da intervenção selecionada;

4.3 O sistema mostra os dados do documento selecionado.

[A2] Execução do download de arquivo em anexo

Caso no passo 4 do fluxo básico [B1] o usuário desejar fazer o download de um arquivo específico:

4.1 O usuário seleciona uma intervenção da lista de intervenções

4.2 O usuário seleciona o anexo para executar o download;

4.3 O sistema efetua o download do arquivo selecionado na máquina local do usuário.

2.5 Fluxo de Exceção

[E1] Erro ao executar o download do arquivo

Caso no passo 4.3 do fluxo alternativo [A2] o sistema não conseguir gravar o arquivo na máquina local do usuário:

4.3.1 O sistema apresenta uma mensagem informando que não foi possível executar o download do arquivo;

4.3.2 Encerra o caso de uso.

2.6 Requisitos específicos

RQ-001-UC-340 Filtros para consulta

(Validação)

1. Os filtros para consulta das intervenções no histórico são os descritos no item 1.1 do documento "ATECH.611.01.00018 - Especificação de campos das telas".

RQ-002-UC-340 Limitação da consulta

(Relatório)

1. A consulta de histórico é limitada a um período máximo de 1 mês

2.7 Requisitos realizados

RQ-176 - Terminal de histórico

(Restrição de Design)

1. Devem ser incorporadas ao terminal do Atende MI as funcionalidades de consulta de histórico das intervenções da distribuição. (PropTec. 5.3)

RQ-180 - Consulta detalhes de intervenção

(Funcional)

1. O sistema deve permitir consultar os detalhes de uma intervenção (documento e anexos). (PropTec. 5.6.1)

RQ-205 - Consulta de histórico

(Funcional)

1. O sistema deve permitir executar consultas dos dados de intervenções na base histórica.

1.1. A consulta poderá ser efetuada a partir de filtros por campos discretos.

(PropTec. 5.7)

Modelo conceitual

cd Modelo Conceitual

Logradouros

- + Bairro
- + Localidade Comercial
- + Localidade Técnica
- + Logradouro
- + Município
- + Segmento Logradouro
- + Sub-bairro
- + Tipo logradouro
- + Título logradouro

Gestão de Turmas

- + Ativação
- + Atividade
- + Atividade da turma
- + Eletricista
- + Habilidade
- + Habilidade da turma
- + Habilidade do eletricista
- + Tipo de veículo
- + Turma
- + Veículo

Ordem de Serviço

- + Deslocamento
- + Endereço
- + Inspeção de Fraude
- + Natureza de Serviço
- + Ocorrência de Emergência
- + Ocorrência de IP
- + Ordem de Serviço
- + OS Comercial
- + Reclamação
- + Reclamação de Emergência
- + Reclamação de IP
- + Tipo de Serviço

Controle de acesso

- + Funcionalidade
- + Perfil
- + Sistema
- + Usuário Operação

Comunicação entre objetos

Conceitualmente, objetos se comunicam através da troca de mensagens.

Mensagens definem:

- O nome do serviço requisitado
- A informação necessária para a execução do serviço
- O nome do requisitante.

Comunicação entre objetos

Na prática, mensagens são implementadas como chamadas de função.

Nome = o nome do procedimento.

Informação = a lista de parâmetros.

Requisitante = o procedimento que realizou a chamada.

Diagramas de interação

Uma colaboração é um nome dado à interação entre duas ou mais classes.

Um colaboração é representada por um conjunto de interações entre objetos selecionados em um contexto limitado, focando a relação entre os objetos.

Esta colaboração pode mostrar a implementação de uma operação ou a realização de um use case.

Diagramas de interação

Uma colaboração pode ser descrita com mais de um diagrama de interação, com cada um mostrando uma visão diferente.

Normalmente pode-se escolher entre utilizar o diagrama de **colaboração** ou o diagrama de **sequência**.

Basicamente, estes diagramas mostram o mesmo fato em perspectivas diferentes.

Diagrama de colaboração

O **diagrama de colaboração** enfatiza o objeto e sua cooperação com os outros objetos, entre eles, as mensagens são apresentadas.

No diagrama de colaboração, além de mostrar a troca de mensagens entre os objetos, percebe-se também os objetos com os seus relacionamentos.

Este diagrama mostra a cronologia das mensagens, seus nomes e respostas, e seus possíveis argumentos.

Diagrama de colaboração

O diagrama de colaboração é desenhado como um diagrama de objeto, onde os diversos objetos são mostrados juntamente com seus relacionamentos.

A cronologia é apresentada através da numeração das mensagens. A relação do objeto pode existir temporariamente ou permanentemente.

Diagrama de colaboração

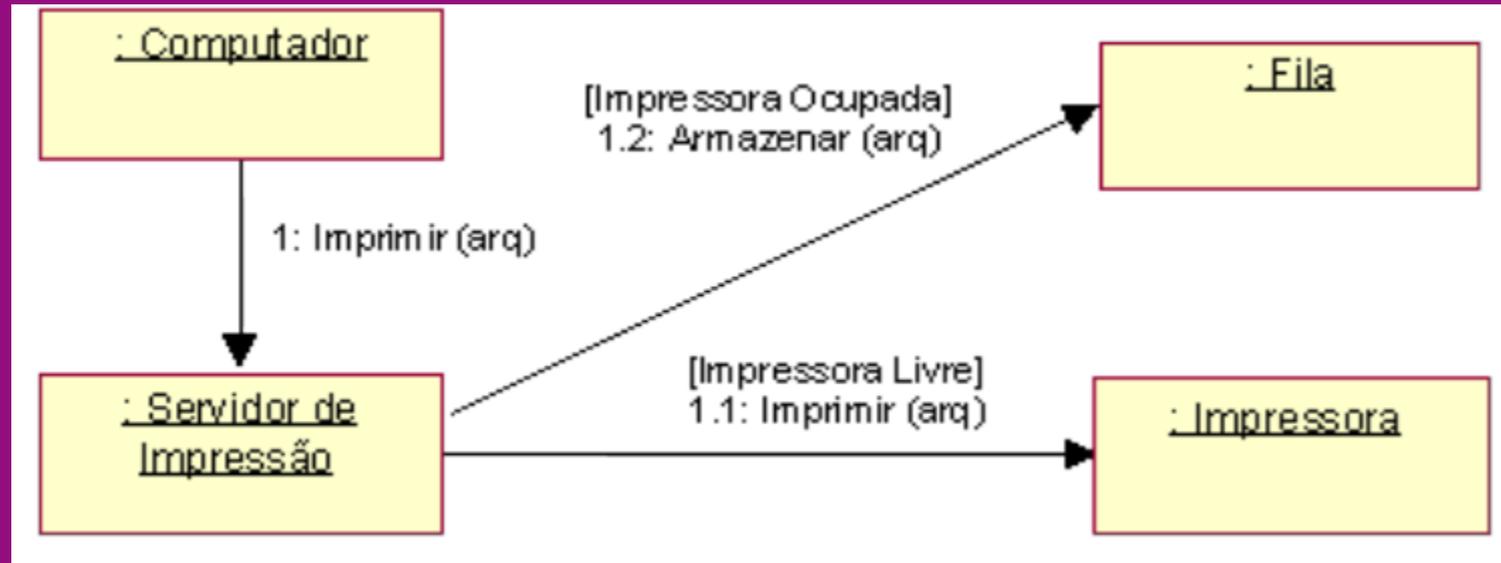


Diagrama de colaboração

Sintaxe:

```
PredecessorCondition SequenceExpression
                               ReturnValue      :=      MessageName
(ArgumentList)
```

PredecessorCondition [opcional]: numeração da mensagem que deve ter sido enviada antes da mensagem atual ser executada. Ex: 1.1, 1.2

Sequence Expression: ordem ascendente de numeração.

Exemplos:

```
1.2.* [i:=1..n]
```

```
1.2.* || [i:=1..n]
```

```
1.2.* [x>5]
```

Diagrama de colaboração

Os objetos que são criados em determinado cenário recebem <<new>>, objetos que são destruídos recebem <<destroy>>.

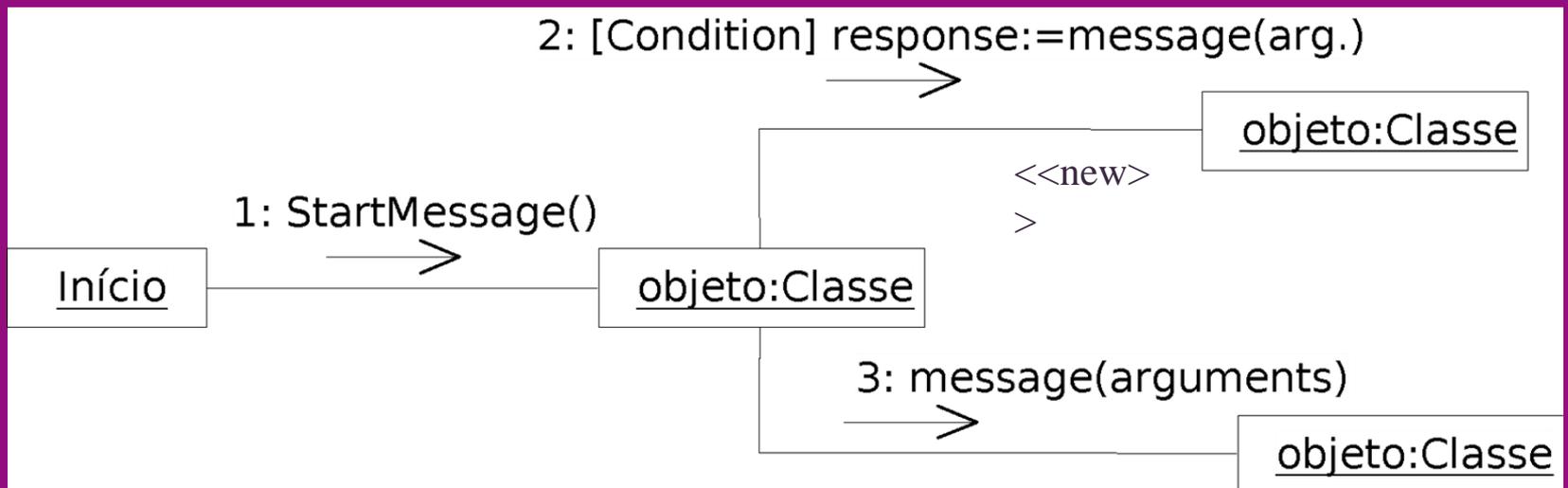


Diagrama de colaboração

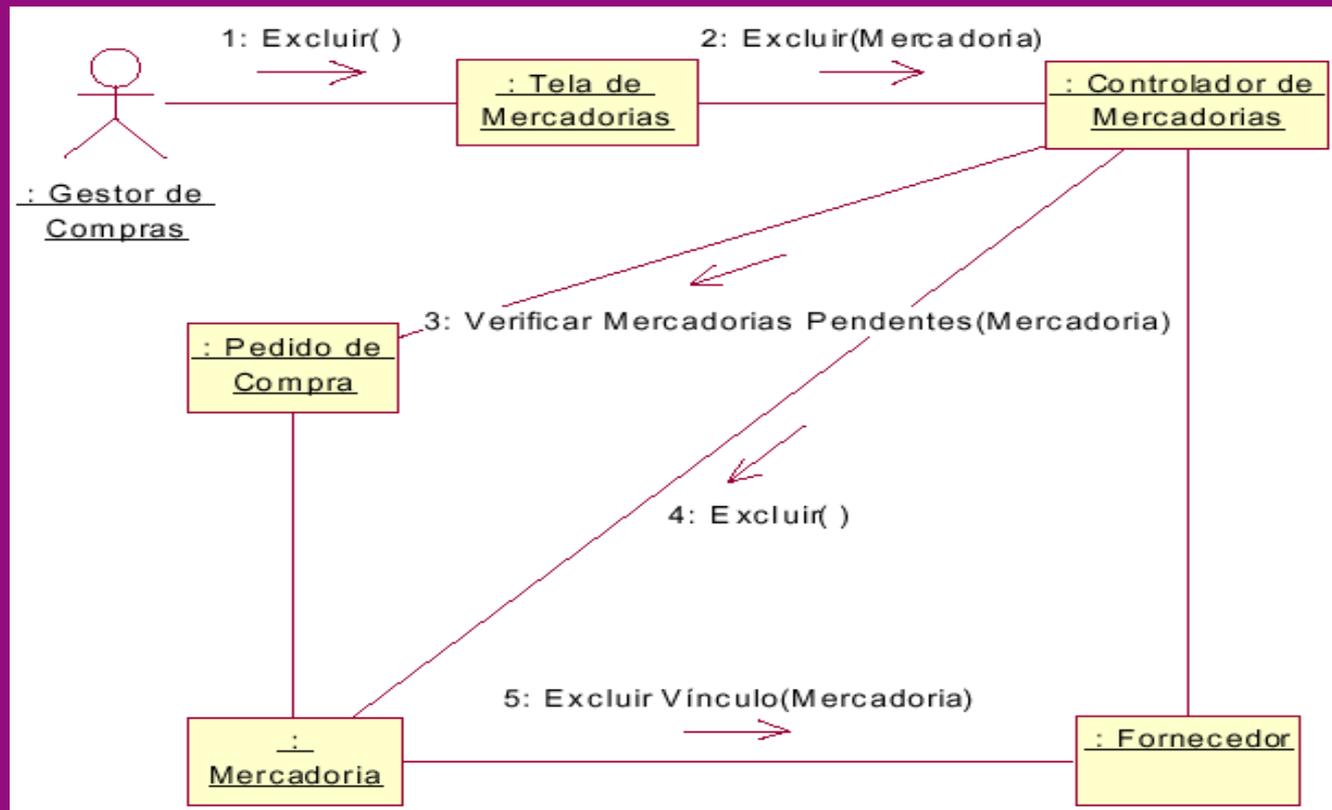


Diagrama de colaboração

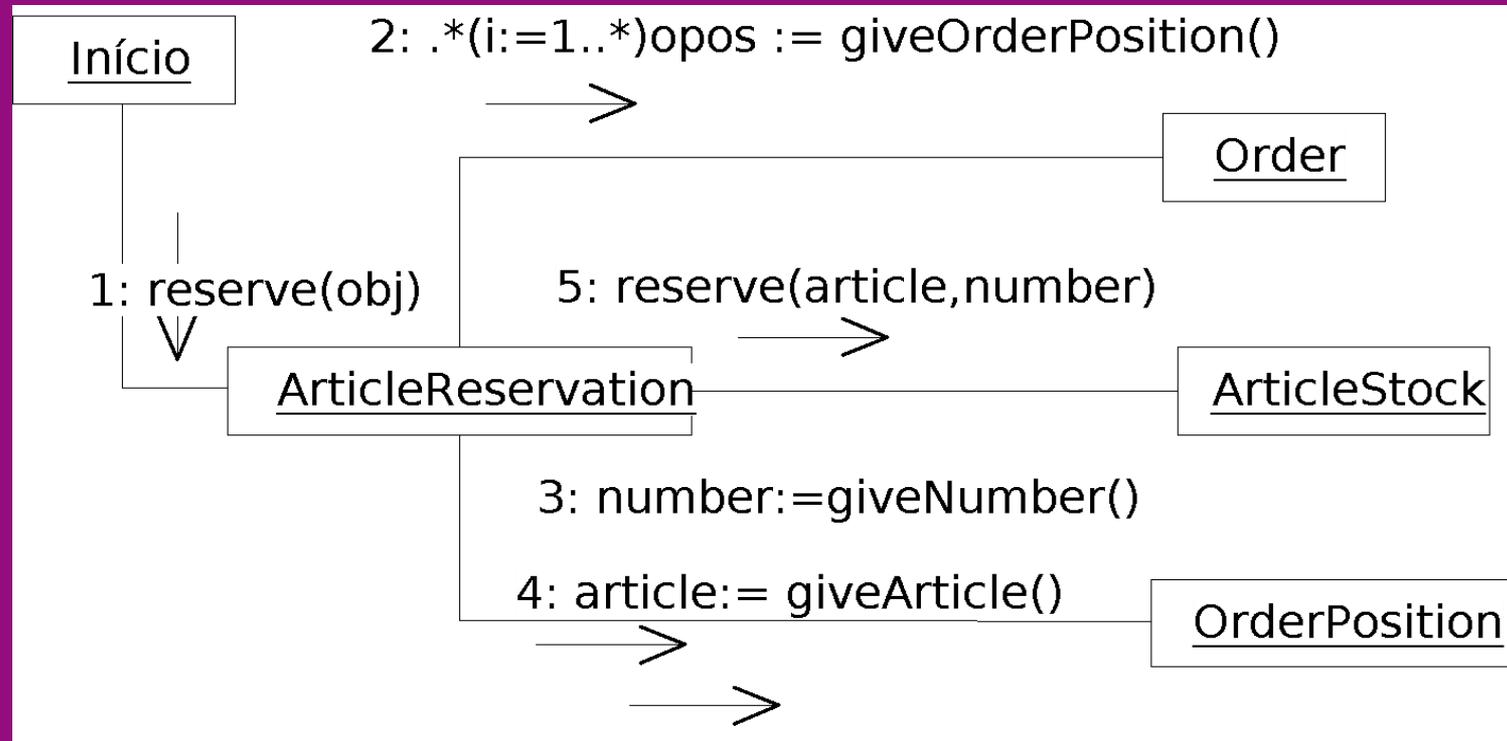
As setas de mensagens são desenhadas entre os objetos para mostrar o fluxo de mensagens entre eles.

Nos diagramas de colaboração a ordenação das mensagens é mostrada apenas pela numeração delas.

Pode conter objetos ativos que executam os métodos paralelamente com outros.

Este diagrama é útil sempre que você quer se referir a uma interação particular.

Diagrama de colaboração



```
private $host;
private $username;
private $password;
private $database;
private $charset;

static public $link = null;

public function connect()
{
    $link = mysql_connect($this::$host, $this::$username, $this::$password);
    if (!$link) {
        throw new MySQLException("MySQL connection failed: " . mysql_error());
    }
    mysql_select_db($this::$database, $link);
    mysql_set_charset($this::$charset, $link);
}
```

Diagrama de sequencia

Diagrama de sequência

Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento. O diagrama de sequência representa essa informação de uma forma simples e lógica.

Ele registra o comportamento de um único caso de uso e exibe os objetos e as mensagens passadas entre esses objetos no caso de uso.

Diagrama de sequência

Representa interações entre objetos de um cenário, realizadas através de operações ou métodos (procedimentos ou funções).

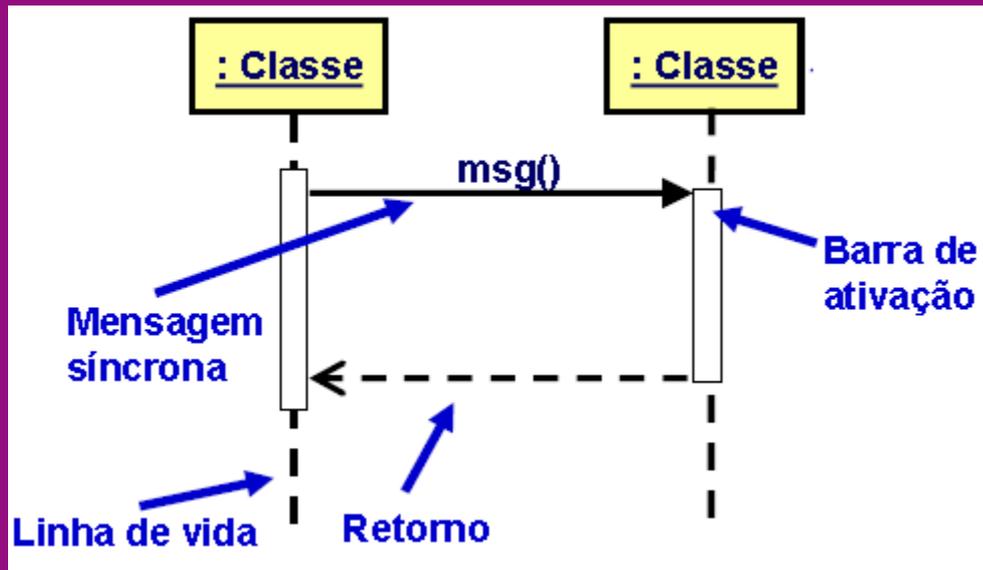


Diagrama de Seqüência

- Interação entre os objetos
- Determina a seqüência de eventos que ocorrem em um determinado processo
 - Quais condições devem ser satisfeitas ...
 - Quais métodos devem ser disparados ...
 - E em qual ordem ...
- Baseia-se no Diagrama de Casos de Uso
 - 1 Caso de Uso ⇔ N Diagramas de Seqüência
- Baseia-se, também, no Diagrama de Classes
 - Fornecem as classes e os métodos associados

Diagrama de Seqüência

- **Componentes - ATORES**
- Exatamente os mesmos dos Casos de Uso
- Interagem , Solicitam serviços , Eventos , Processos
- Não são obrigatórios no Diagrama de Seqüência

Diagrama de Seqüência

- **Componentes - OBJETOS**

- Representam as instâncias das classes
- Retângulos contendo um texto
 - Primeira parte, em minúsculo, o nome do objeto
 - Segunda parte, em letras iniciais maiúsculas, o nome da classe
 - Informações separadas por dois pontos (:)
- Linha de vida
 - Linha vertical tracejada

física1:

Física



Diagrama de Seqüência

- **Componentes – LINHAS DE VIDA**
- Representa o tempo que um objeto existiu durante um processo
- Linhas finas verticais tracejadas
 - Iniciam no retângulo que representa o objeto
 - Interrompida por um “X” quando o objeto é destruído

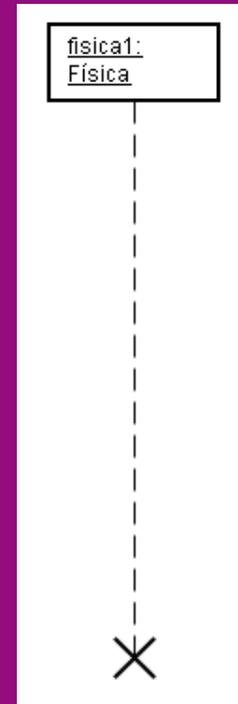


Diagrama de Seqüência

- **Componentes – FOCO DE CONTROLE/ATIVACÃO**
- Indica os períodos em que um determinado objeto está participando ativamente do processo
 - Executando um ou mais métodos do processo
- Representados por extensões mais grossas/largas da Linha de Vida

Diagrama de Seqüência

- Componentes – FOCO DE CONTROLE/ATIVAÇÃO

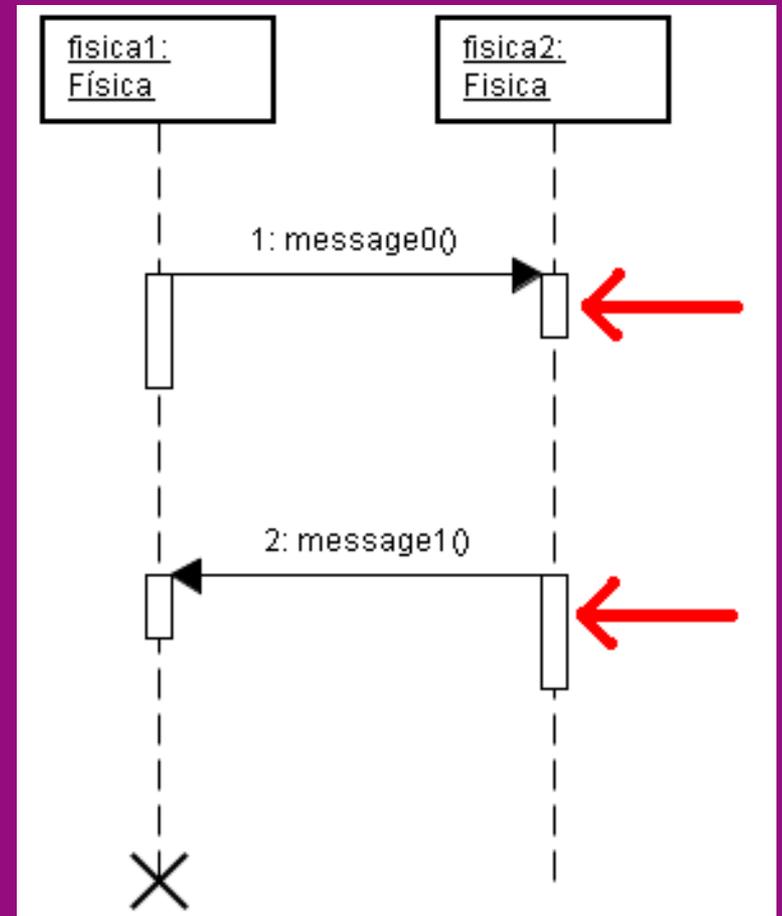


Diagrama de Seqüência

- **Componentes – MENSAGENS/ESTÍMULOS**
- Demonstram a ocorrência de eventos que normalmente forçam a chamada de um método em algum dos objetos envolvidos no processo
- Mensagens entre:
 - Ator e Ator
 - Ator e Objeto
 - Objeto e Objeto
 - Objeto e Ator

Diagrama de Seqüência

- Mensagem com disparo de método entre objetos.

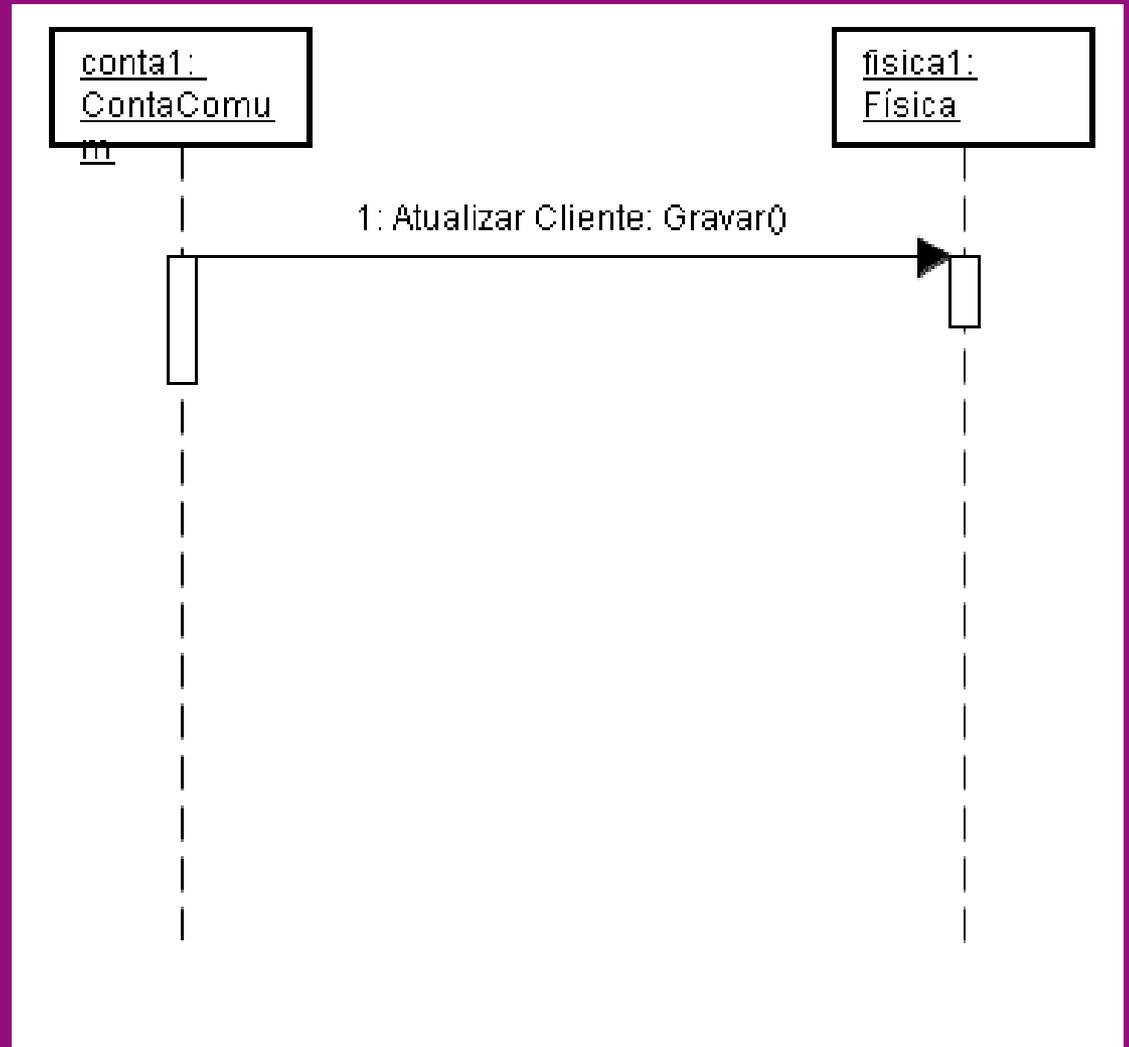


Diagrama de Seqüência

- Mensagem que instancia um novo objeto

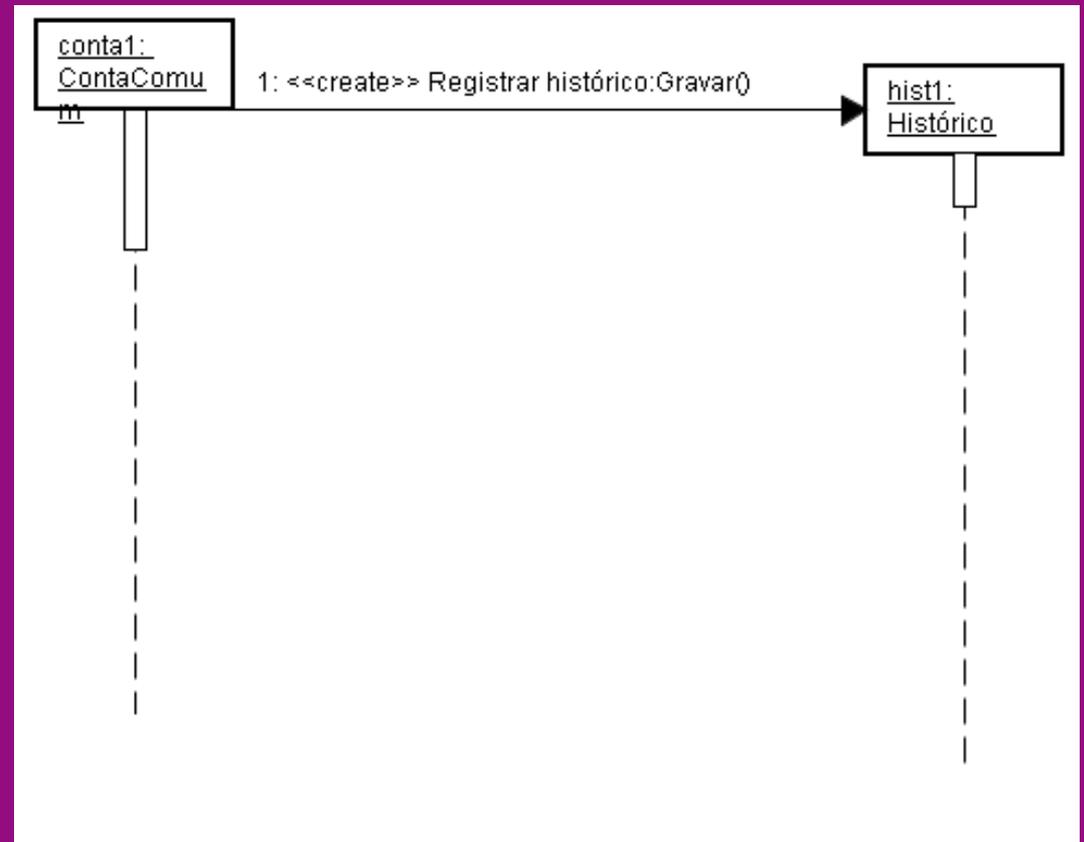


Diagrama de Seqüência

- Mensagem que dispara um método destrutor – elimina um objeto não mais necessário.

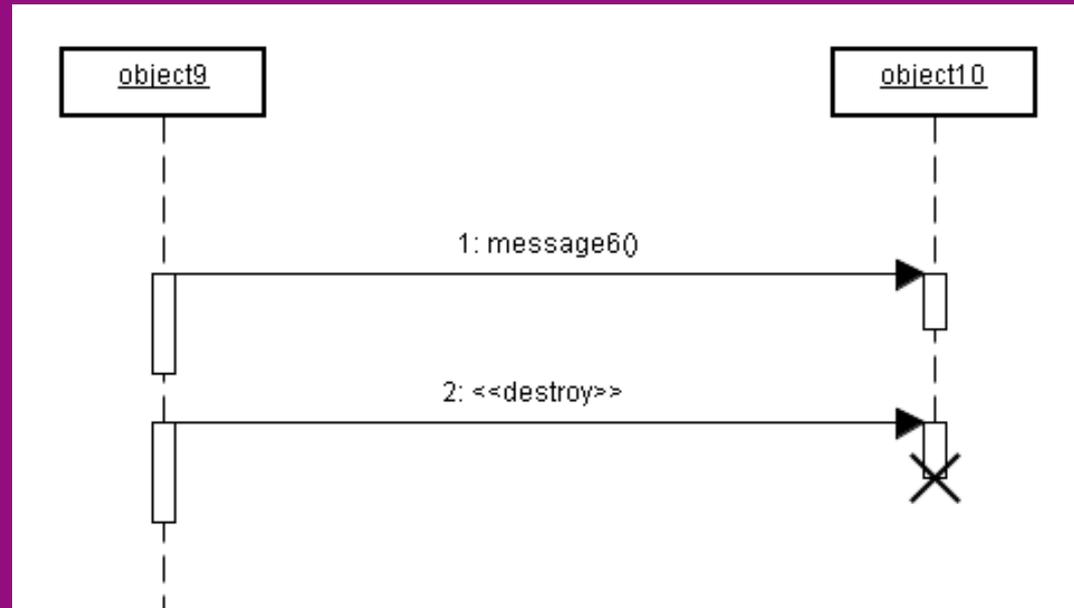


Diagrama de Seqüência

- Mensagem de Retorno ... Linha tracejada.
- Podem retornar valores ou status...

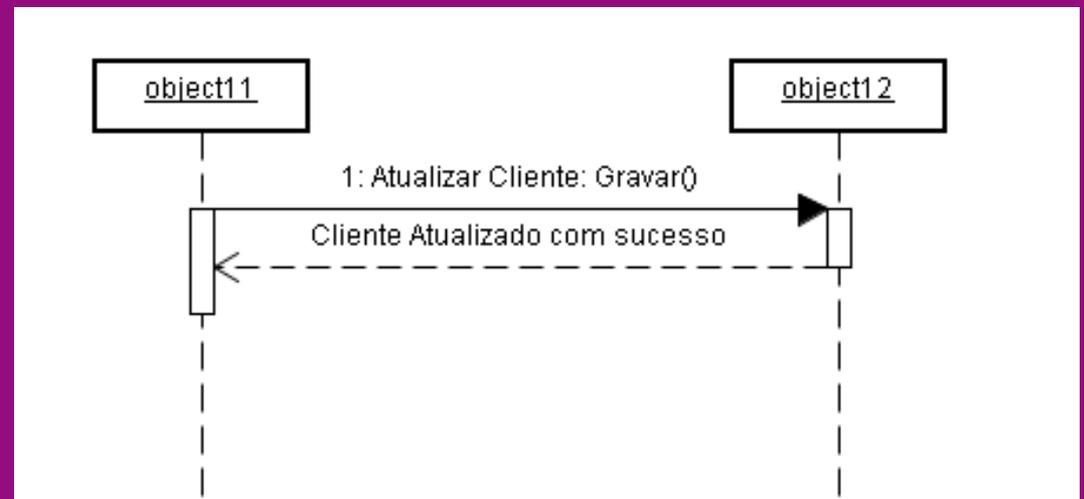


Diagrama de Seqüência

- Mensagem com Condição de Guarda
- Entre colchetes []

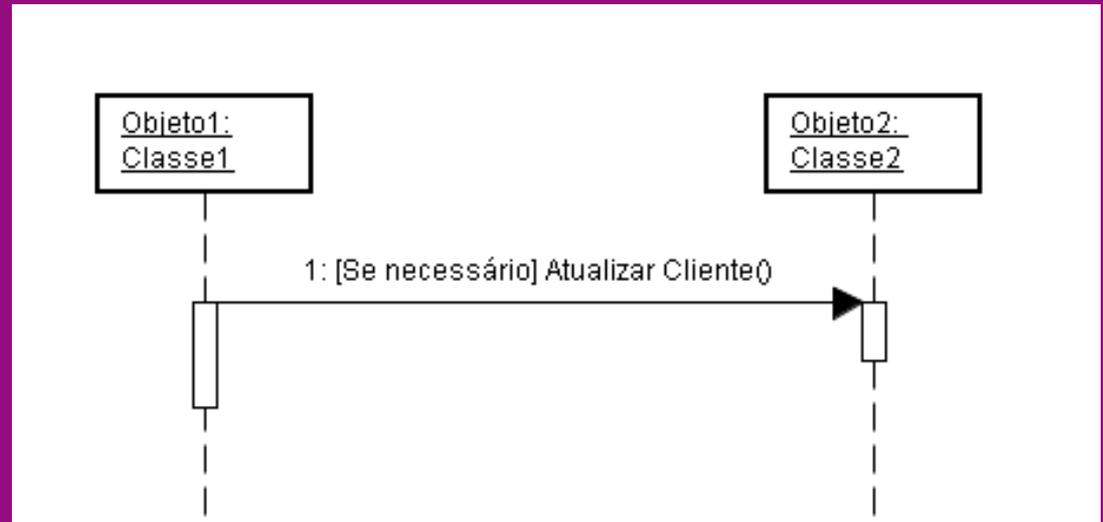
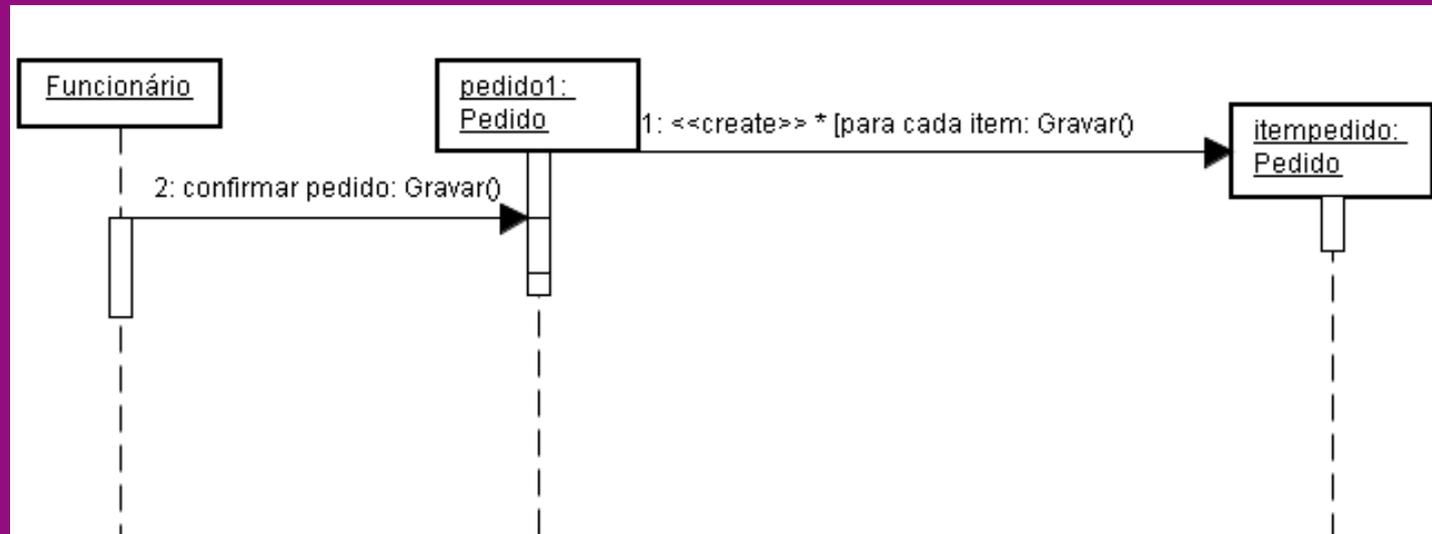


Diagrama de Seqüência



- Mensagem com Condição de Guarda
- Disparo de mensagem a vários objetos (*)

Abertura de Conta

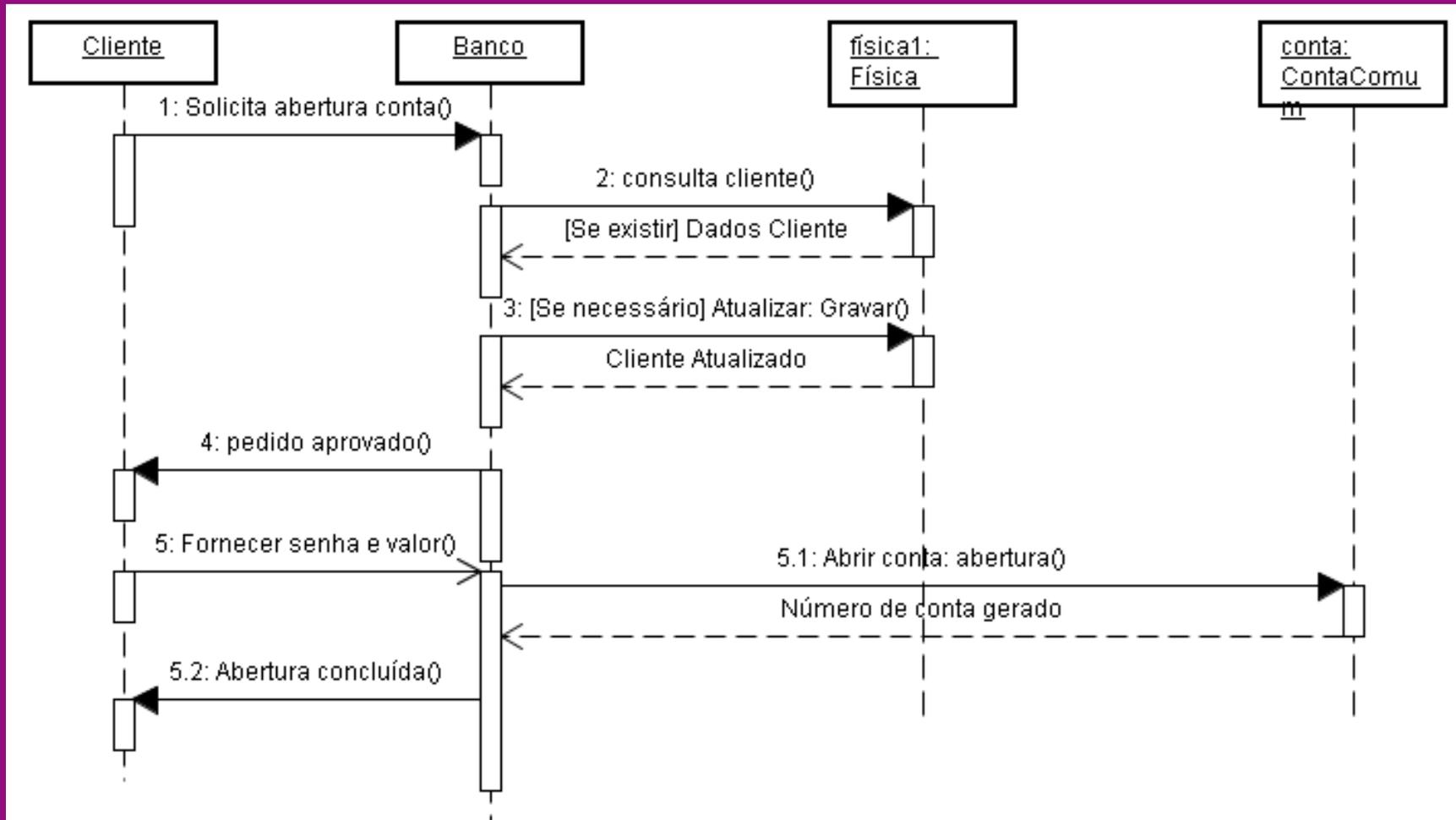


Diagrama de sequência

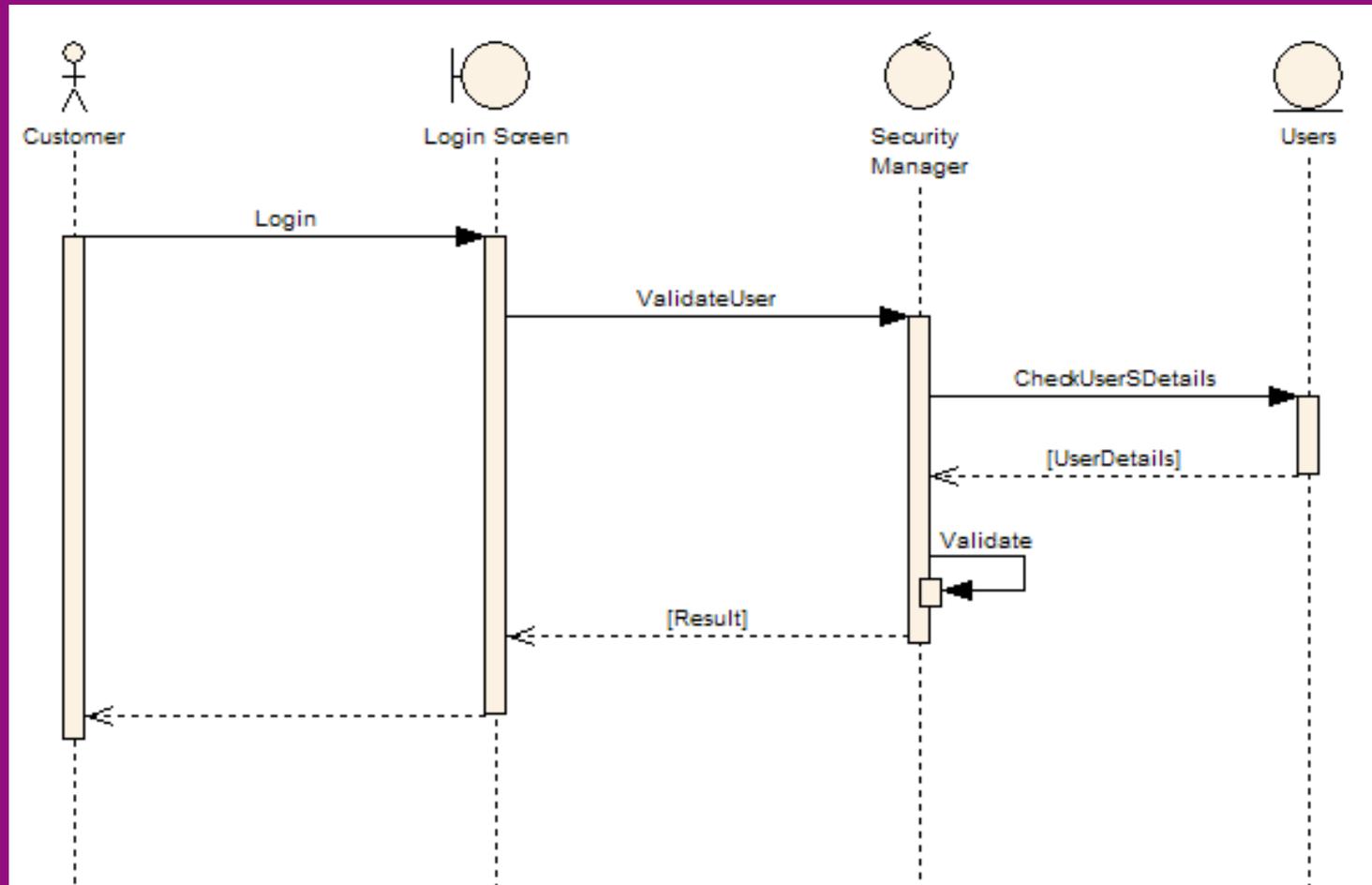
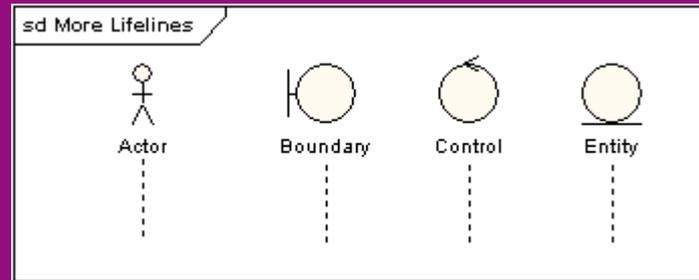
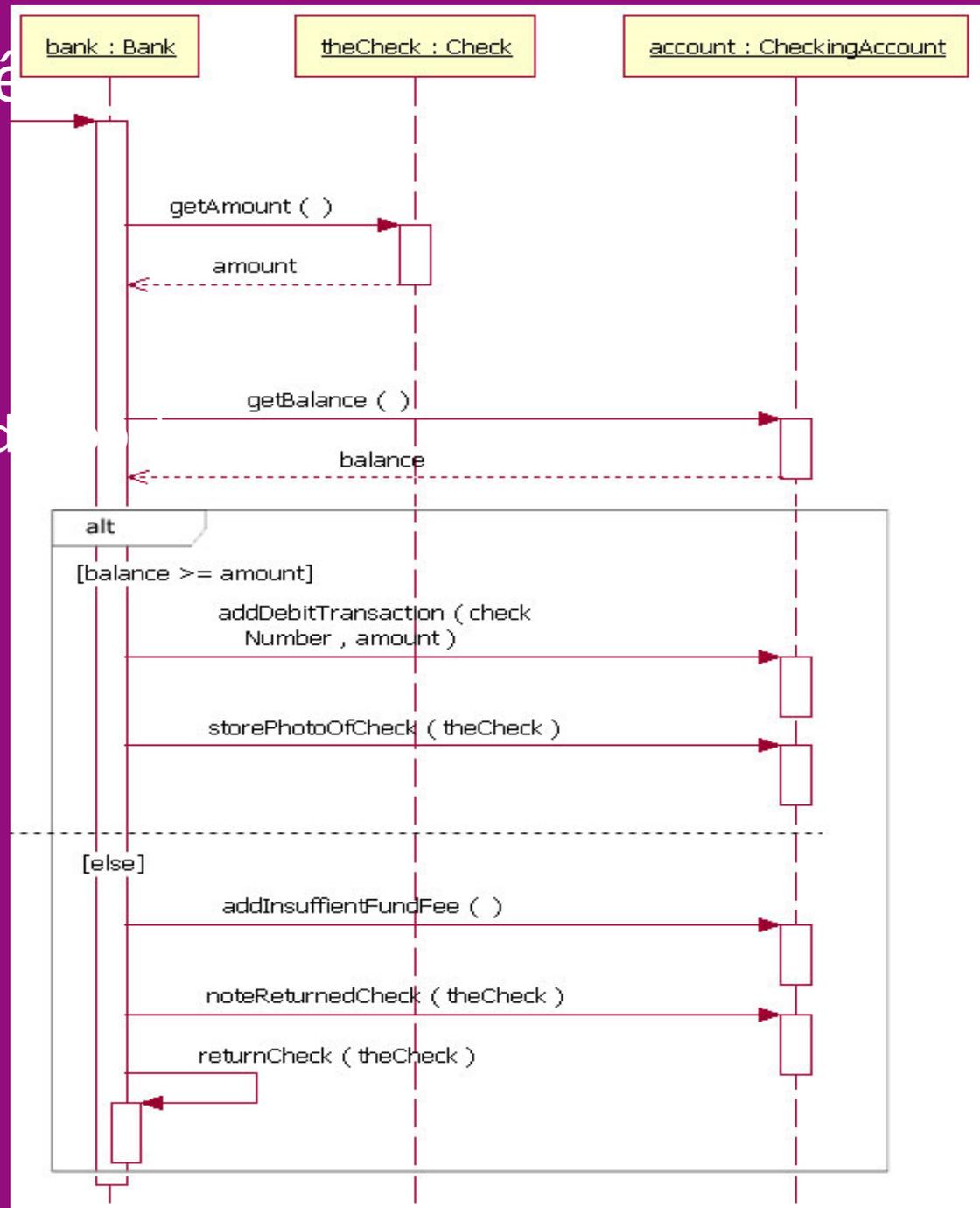


Diagrama de sequência

Fragmentos combinados
(alternatives, options, and



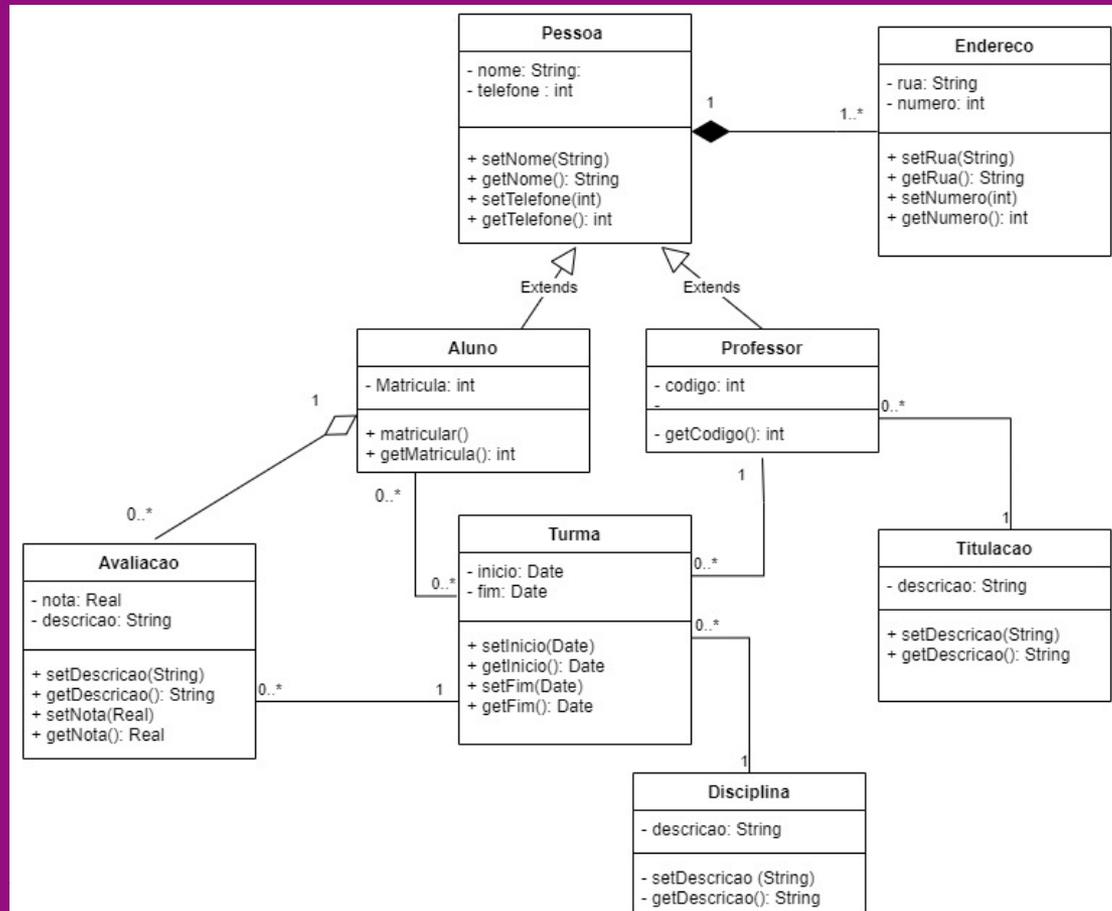
Vamos trabalhar?



Exercício: Registro de notas

Escopo do projeto

Professor de um curso acessa o sistema para registrar as notas da prova de uma turma.



Referência bibliográfica

FOWLER, Martin e SCOTT, Kendall. Uml Essencial. 2a. Edição. Bookman. Porto Alegre, 2000.

SCHNEIDER, Geri. Applying use case: a practical guide. Addison-Wesley, 1998.

OESTEREICH, Bernd. Developing Software with UML. Addison-Wesley, 1999.



CRÉDITOS

COORDENAÇÃO



Vera Rejane Niedersberg
Schuhmacher

PROFESSORES



Rafael Lessa
Daniella Vieira
