



Desenvolvimento de Sistemas Software

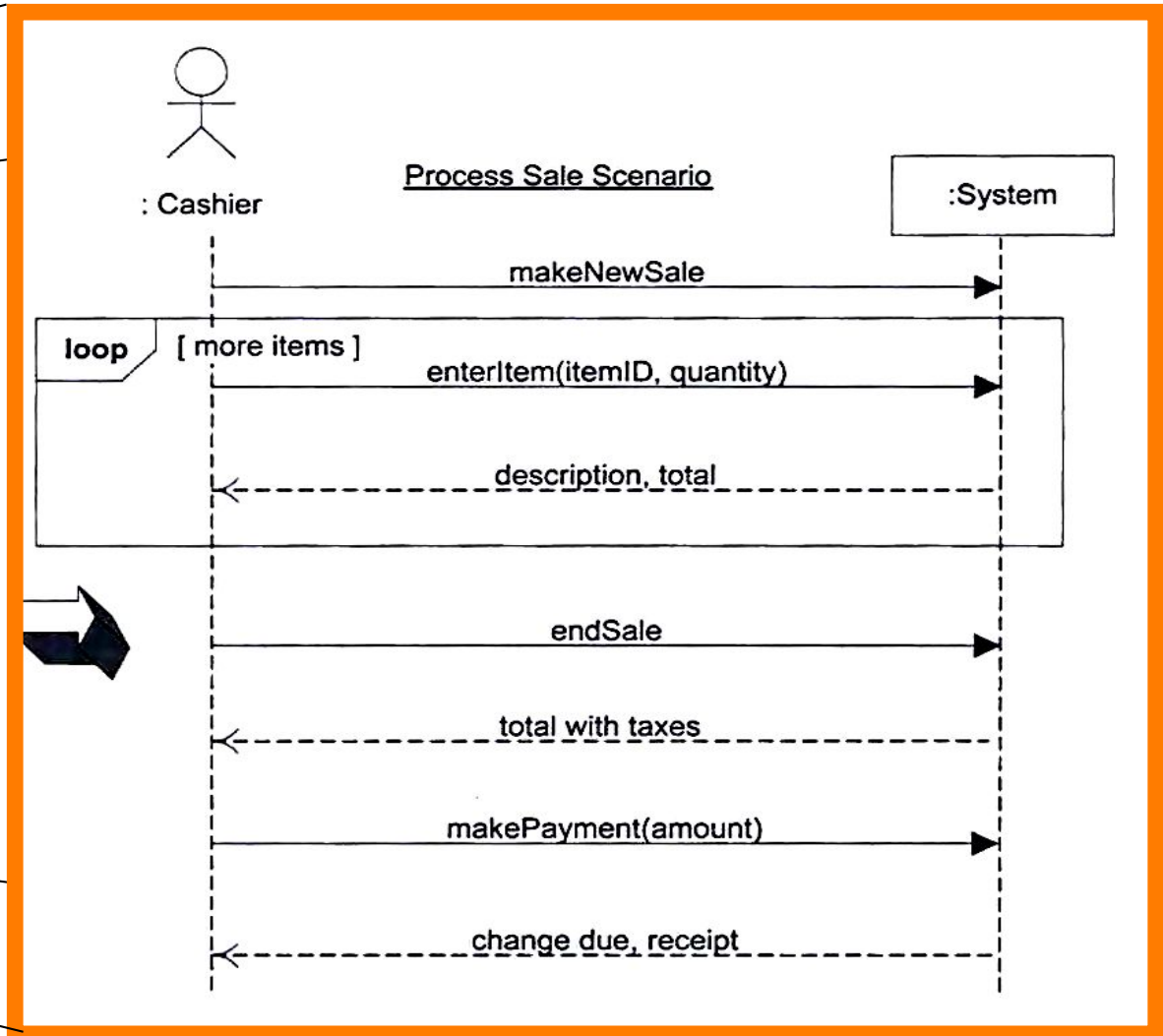
Aula Teórica 13: Especificação de Use Cases com Diagramas de Sequência



REPRESENTAÇÃO DE USE CASE EM DIAGRAMAS DE SEQUÊNCIA

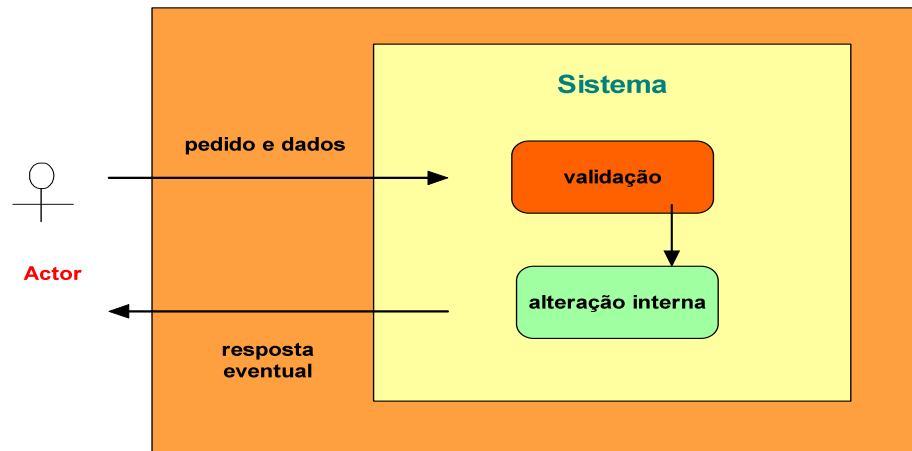
Simple cash-only Process Sale scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
- ...



MÉTODO: UCs → DS

- Use Cases não são operações do sistema mas incluem operações do sistema;
- Jacobson diz que: cada **use case** constitui uma série completa de eventos iniciados por um actor e especifica a interacção que tem lugar entre o actor e o sistema. Um use case é, portanto, uma sequência especial de transacções relacionadas, executadas por um actor e pelo sistema em diálogo.



(para o actor é atómica !)

TRANSAÇÃO

Cada transacção destas pode muito bem ser analisada, individualizada e transposta de forma sistemática para o Diagrama de Sequência



MÉTODO: UCs → DS

- ▣ Aceites todas estas definições “standard” do que se deve entender ser um UC, o que se deve entender por <<include>> e por <<extend>>, como se devem escrever UC textuais, etc., torna-se agora importante passar dos diagramas de UC e das descrições textuais de Ucs, para especificações mais detalhadas do ponto de vista comportamental.
- ▣ **Problema:** Como passar sistematicamente, ou seja com método, dos UC para os Diagramas de Sequência que especificam, com mais detalhe, como o Sistema em desenvolvimento responde às mais diversas solicitações dos actores, tal como anteriormente identificado no Diagrama de Use Cases ? **Resposta:** Não há qualquer metodologia no RUP para o fazer.
- ▣ **Vamos assim introduzir o nosso próprio método**, desenvolvido visando uma coerente e correcta resolução desta fase de transição do RUP, de forma a que, seguindo certas regras, se atinjam certos objectivos de forma coerente e, portanto, sendo os passos reversivelmente identificáveis.



MÉTODO: UCs → DS

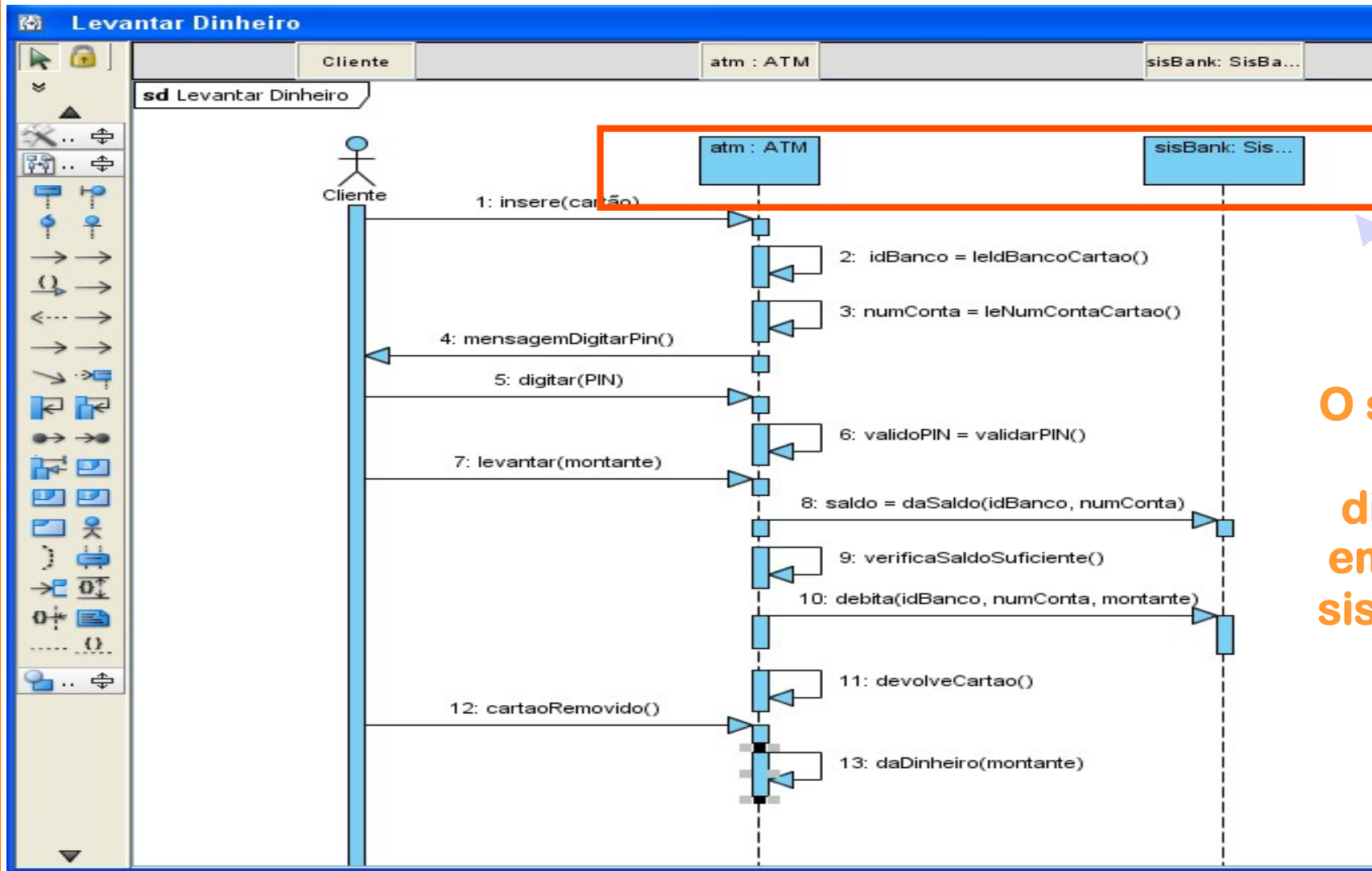
PASSO 1 : Identificar as transacções no UC (por exemplo com cores)

Use Case: Levantar Dinheiro		
Cenário de Sucesso		
	Actor	Sistema
1	Cliente identifica-se introduzindo o seu cartão	
2		Lê do cartão identificação do Banco e número da Conta
3		Sistema pede o PIN ao Cliente
4	Cliente introduz o PIN	
5		Sistema verifica que oPIN é válido
6	O Cliente solicita o levantamento de uma dada importância	
7		Sistema verifica que a conta tem saldo suficiente
8		Sistema subtrai a importância que vai disponibilizar ao saldo da Conta
9		Sistema devolve cartão
10	Cliente retira Cartão	
11		Sistema disponibiliza a importância em notas

Algumas são do tipo Acção -> Reacção (ie. possuem apenas 2 passos)



MÉTODO: UCs → DS



O sistema
foi já
dividido
em 2 sub-
sistemas !



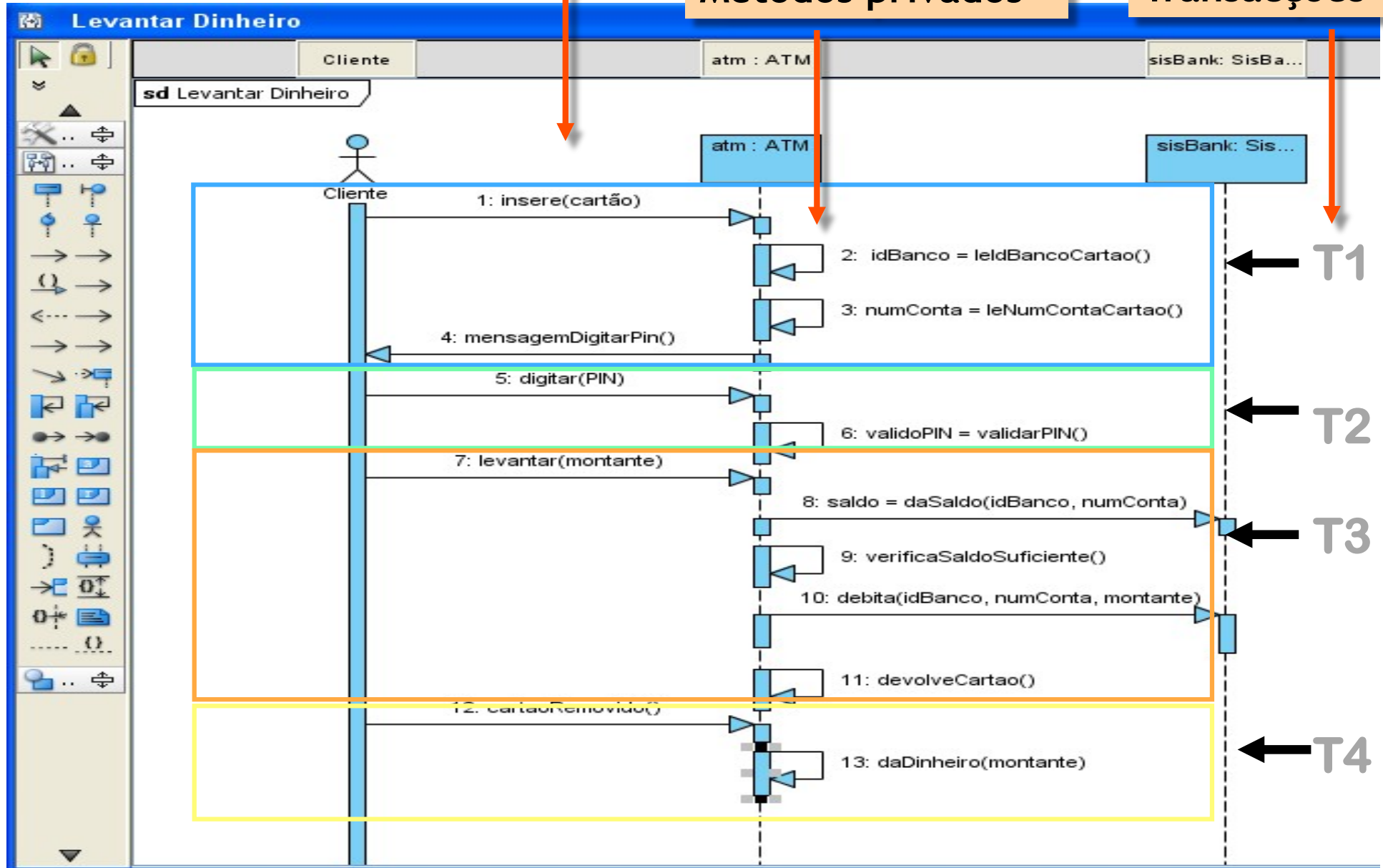
269

API pública

MÉTODO: UCs → DS

Métodos privados

Transacções





MÉTODO: UCs → DS

LEVANTAMENTO DE UMA ATM: Alternativas e Excepções. Como fazer ?

Levantar Dinheiro Details			
Name: <u>Levantar Dinheiro</u>			
Info Description Diagrams			
Agency FB 8			
Main			
Alternatives	1	Actor Input	System Response
	2		5a PIN inválido
	3		O PIN é inválido (menos de 3 vezes)
	4		Sistema actualiza número de tentativas
			Voltar ao passo 3 do Main Flow
Exception: 2	1	Actor Input	System Response
	2		2a Sistema não consegue ler informação do Cartão (Cartão ilegível)
	3		Sistema informa Cliente do erro
			Sistema devolve o Cartão
Exception: 5	1	Actor Input	System Response
	2		5a O PIN é inválido pela 3ª vez
			Sistema recolhe o Cartão
Exception: 6	1	Actor Input	System Response
	2		6a Sistema não consegue ler informação bancária do Cartão (Cartão ilegível)
	3		Sistema informa Cliente do erro
			Sistema devolve o Cartão
Exception: 8	1	Actor Input	System Response
	2		8a Sistema não possui dinheiro suficiente ou combinação de notas para tal Levantamento
	3		Sistema informa o utilizador do total máximo ou da combinação de notas
			Sistema devolve o Cartão
Exception: 9	1	Actor Input	System Response
	2		9a A Conta bancária não tem saldo suficiente
	3		Sistema informa o utilizador de que a Conta não possui saldo suficiente
			Sistema devolve o Cartão



MÉTODO: UCs → DS

- ▣ Para o **Main Flow** (Cenário de Sucesso) sabemos já identificar as transacções definidas no UC textual e passar cada uma delas para o Diagrama de Sequência.
- ▣ Precisamos agora de ter uma forma sistemática de representar no Diagrama de Sequência os outros dois tipos de fluxo que podemos ter num UC: **Alternativas** e **Excepções**.
- ▣ Precisamos ainda de saber tratar sistematicamente as situações em que o UC **<<include>>** outros UCs ou “**é estendido**” por outros UCs.
- ▣ Se criarmos um “**template**” geral do que pode ser um UC textual, talvez tal nos ajude a identificar todos os casos possíveis de transformação de um UC num DS.



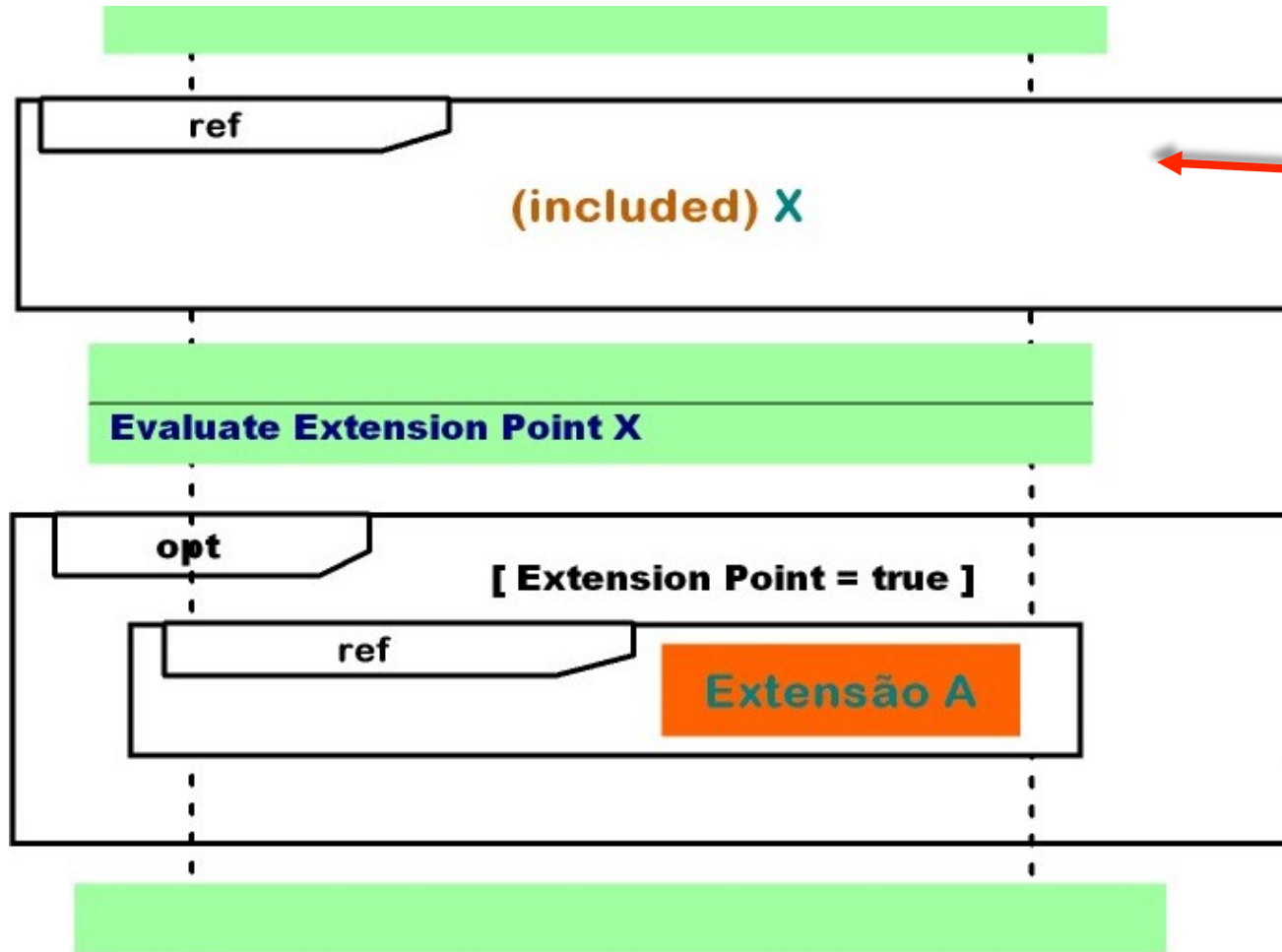
TEMPLATE GERAL UCs

TEMPLATE GERAL COMPLETO DE UM USE CASE TEXTUAL





PASSAGEM PARA DS

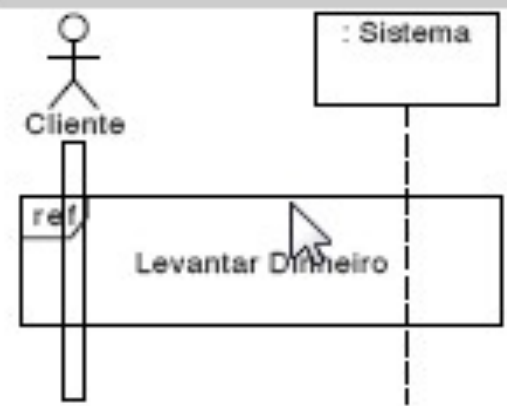


Tratamento dos
<<include>>

Tratamento dos
<<extend>>


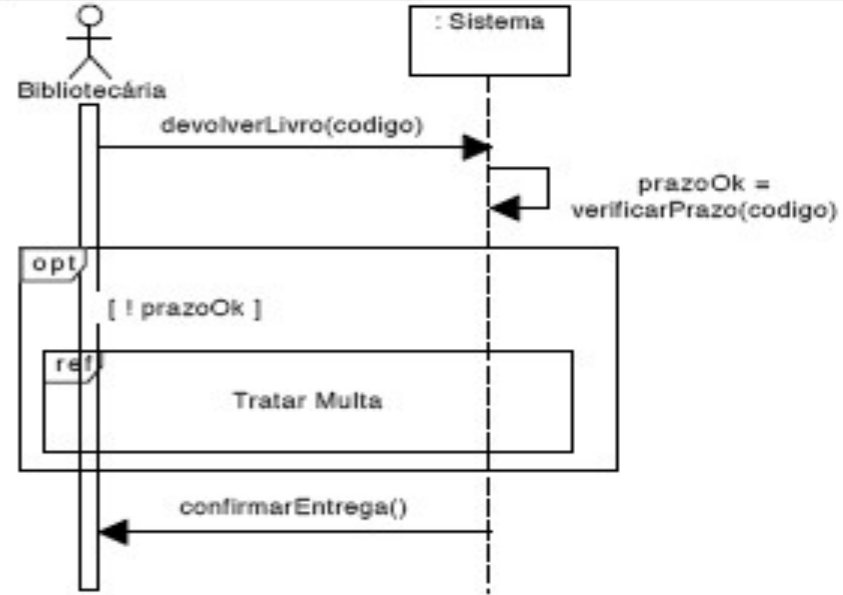
PASSAGEM PARA DS

Tabela 6.12: Regra de transformação de passos de inclusão para DSS

Construção UC	Construção DSS
Passo de inclusão de um UC.	Frame <i>Ref</i>
Exemplo	
1. <i>Include</i> : Levantar Dinheiro	 <pre> sequenceDiagram actor Cliente participant Sistema as : Sistema activate Cliente ref Cliente->>Sistema: Levantar Dinheiro andref Sistema->>Cliente: endref deactivate Cliente </pre>

PASSAGEM PARA DS

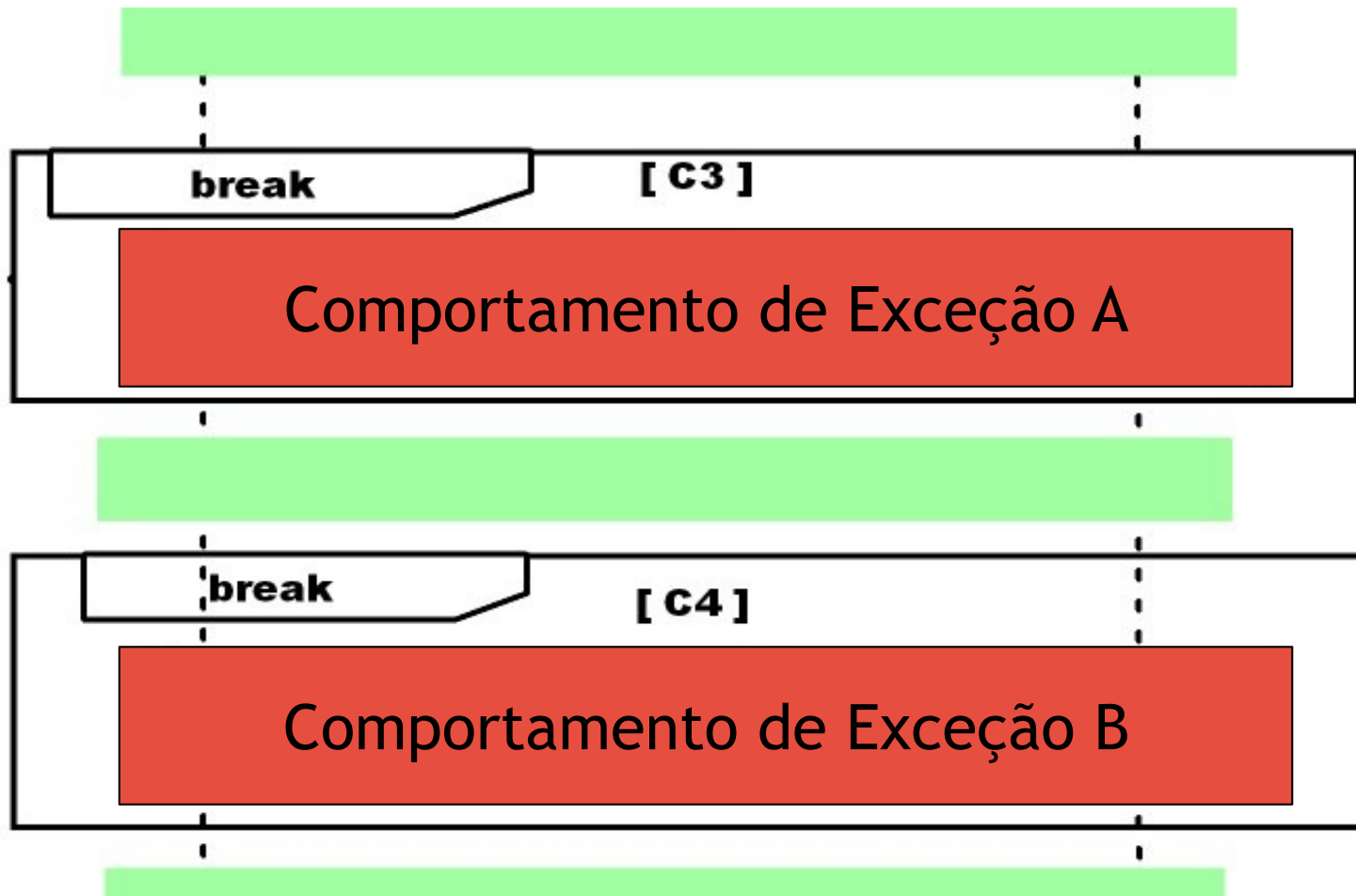
Tabela 6.6: Regra de transformação de extensões para DSS

Construção do UC	Construção do DSS
Extensão	Frame <i>Ref</i> dentro de um frame <i>Alt</i> ou <i>Opt</i> .
<div>  Exemplo </div>	
<ol style="list-style-type: none"> 1. Bibliotecária indica a devolução de um livro. 2. Sistema verifica se a entrega foi feita dentro do prazo. 3. Sistema confirma a entrega. 4. <i>Success</i> <p>2a. Entrega fora de prazo:</p> <p>2a1. <i>Extended by:</i> Tratar Multa</p>	 <pre> sequenceDiagram participant B as Bibliotecária participant S as : Sistema B->>S: devolverLivro(codigo) S->>S: verificarPrazo(codigo) S-->>B: prazoOk = alt [! prazoOk] ref Tratar Multa end S->>B: confirmarEntrega() </pre>



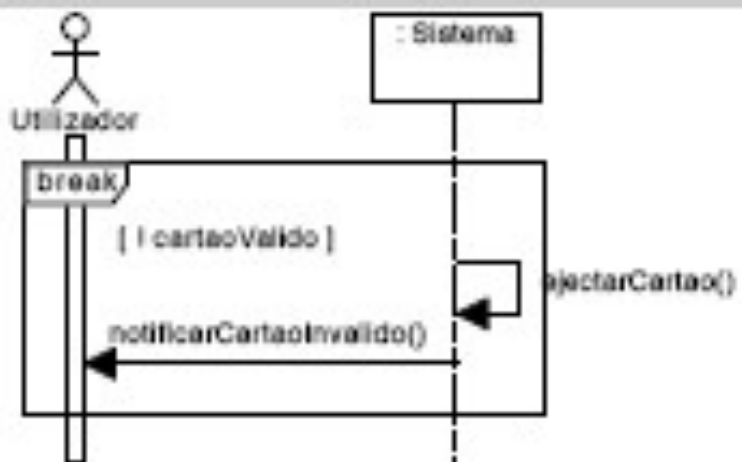
Tratamento dos Fluxos de Exceção

PASSAGEM PARA DS



PASSAGEM PARA DS

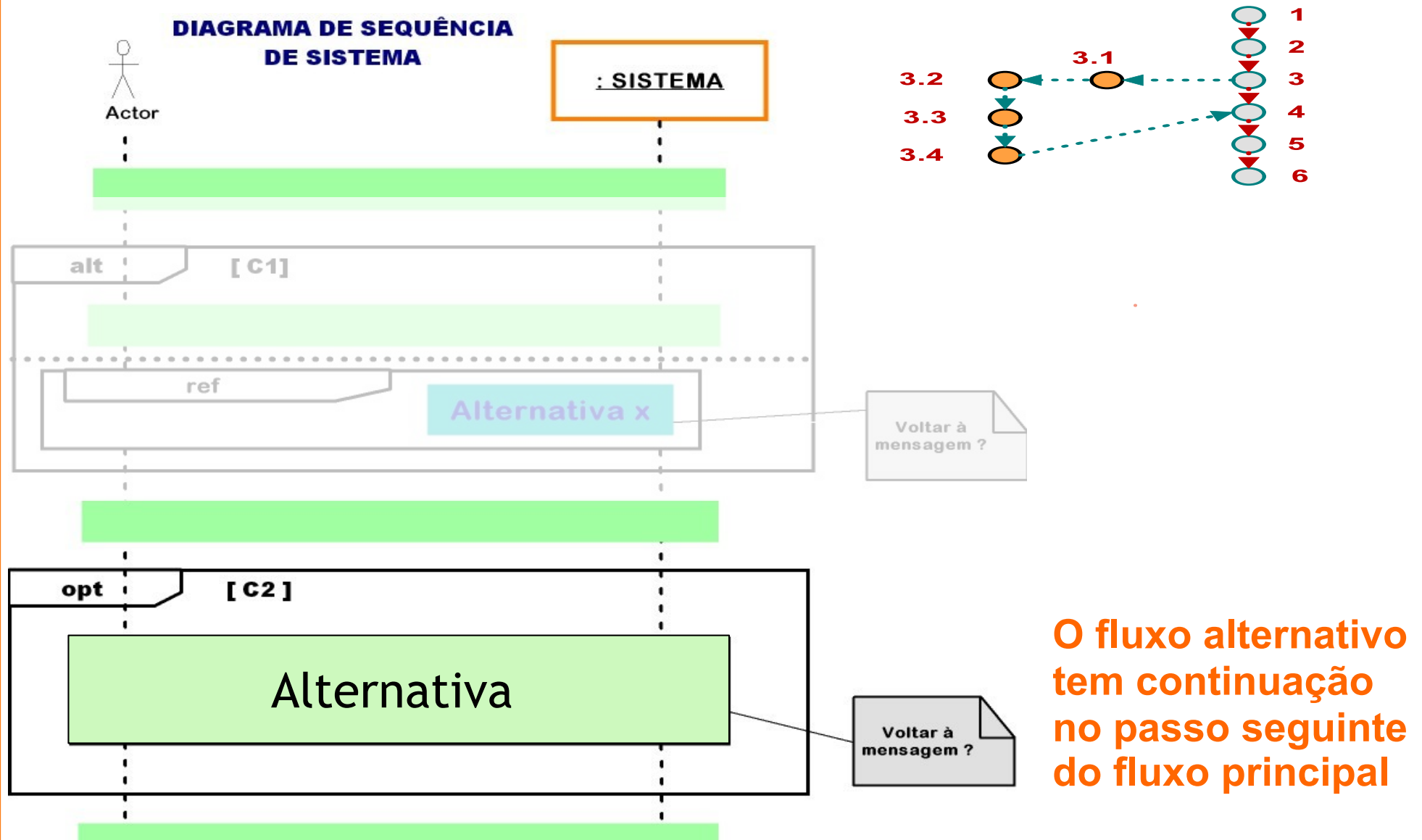
Tabela 6.10: Regra de transformação de fluxos de excepção para DSS

Construção do UC	Construção do DSS
Fluxo de excepção	Frame <i>break</i> contendo os passos do fluxo de excepção.
Exemplo	
<p>1a. Cartão inválido.</p> <p>1a1. Sistema ejecta o cartão.</p> <p>1a2. Sistema notifica o utilizador.</p> <p>1a3. <i>Failure</i></p>	 <pre> sequenceDiagram actor Utilizador participant Sistema as : Sistema activate Utilizador break Sistema->>Sistema: ejectarCartao() Sistema->>Utilizador: notificarCartaoInvalido() end deactivate Utilizador </pre>



Tratamento dos Fluxos Alternativos

PASSAGEM PARA DS



PASSAGEM PARA DS

Tabela 6.7: Regra de transformação de fluxos *goto next/resume* para DSS

Construção do UC

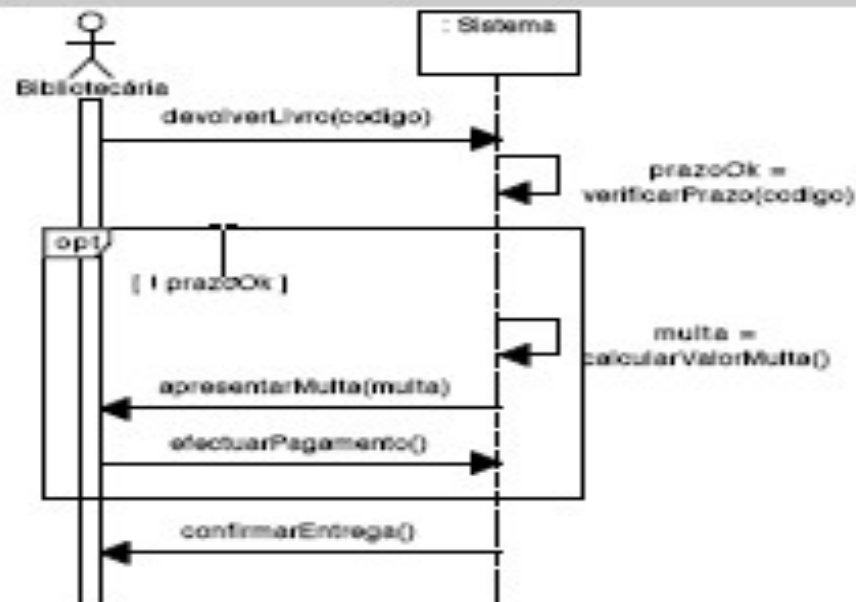
Passo *Goto* que regressa ao passo seguinte ao que gerou a alternativa.

Construção do DSS

Frame *Ref* dentro de um frame *Alt* ou *Opt*.

Exemplo

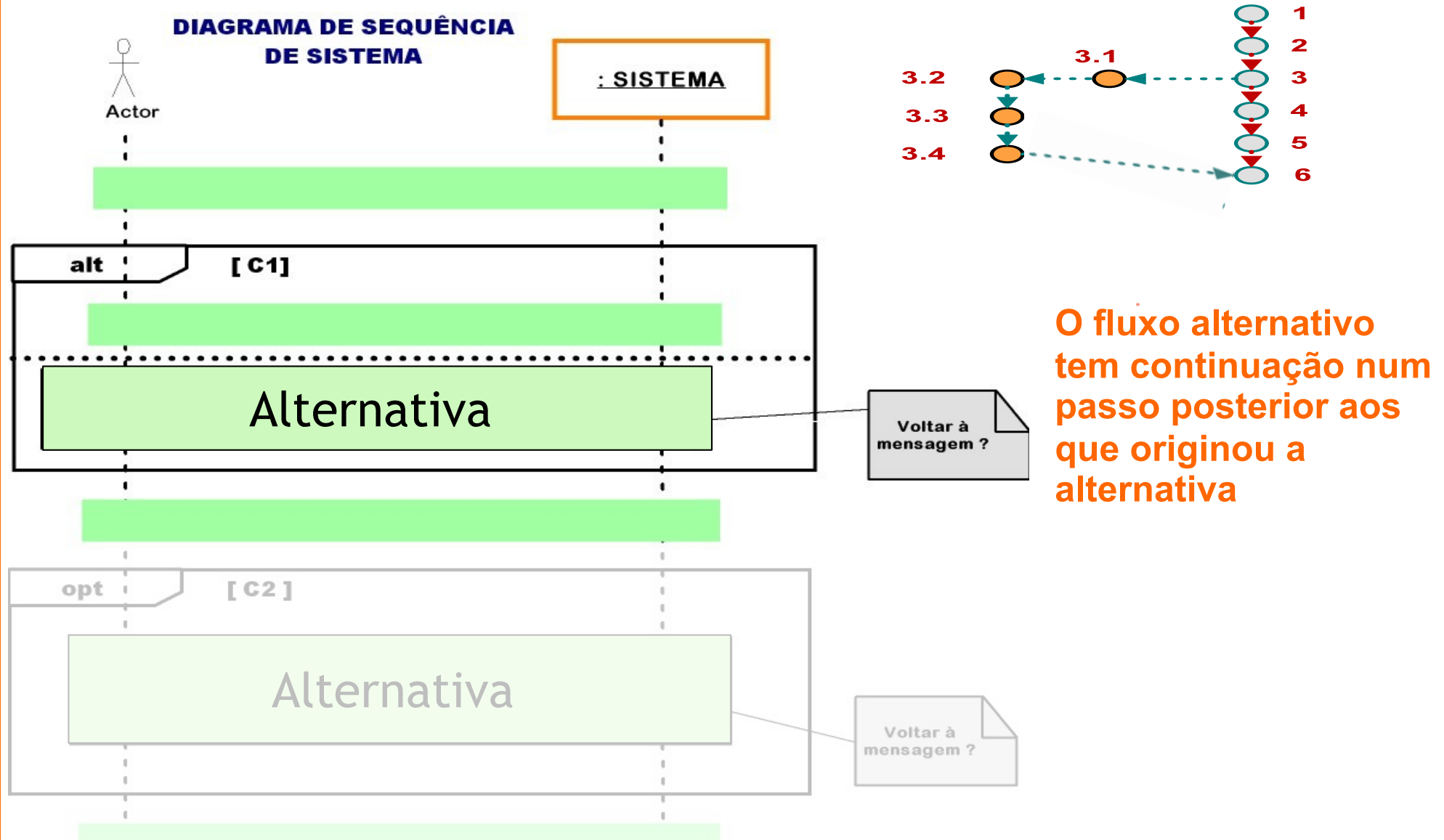
1. Bibliotecária indica a devolução de um livro.
 2. Sistema verifica se a entrega foi feita dentro do prazo.
 3. Sistema confirma a entrega.
 4. *Success*
- 2a. Entrega fora de prazo:**
- 2a1. Sistema calcula montante a pagar.
 - 2a2. Sistema apresenta valor da multa.
 - 2a3. Bibliotecária efectua pagamento.
 - 2a4. *Goto 3* (ou *Resume*)





Tratamento dos Fluxos Alternativos

PASSAGEM PARA DS



PASSAGEM PARA DS

Tabela 6.11: Regra de transformação de fluxos paralelos para DSS

Construção do UC	Construção do DSS
Passo <i>Goto</i> que aponta para um passo posterior ao que originou a alternativa onde se encontra, saltando alguns passos do fluxo principal.	frame <i>Alt</i> com dois operandos. Um deles conterá o fluxo alternativo. O outro conterá a parte saltada.
Exemplo	
<ol style="list-style-type: none"> 1. Cliente pretender encomendar produtos seleccionados. 2. Sistema verifica que se trata de um novo cliente. 3. Sistema requisita os dados necessários para o envio da encomenda. 4. Cliente introduz os dados solicitados. 5. Sistema requisita informação do cartão de crédito. 6. Cliente insere dados do cartão de crédito. <p>2a. É cliente habitual:</p> <ol style="list-style-type: none"> 2a1. Sistema apresenta informações de preço, factura e envio. 2a4. <i>Goto</i> 5 	<pre> sequenceDiagram actor Cliente participant Sistema as : Sistema Cliente->>Sistema: 1: realizarEncomenda(prods) Sistema->>Sistema: 2: novoCliente := verificarCliente(cliente) alt [novoCliente] Sistema->>Cliente: 3: requisitarDados() Cliente->>Sistema: 4: introduzirDados(dados) else [! novoCliente] Sistema->>Cliente: 5: apresentaInfo() end Sistema->>Cliente: 6: requisitaInfoCartaoCredito() Cliente->>Sistema: 7: insereCartaoCredito(info) </pre>

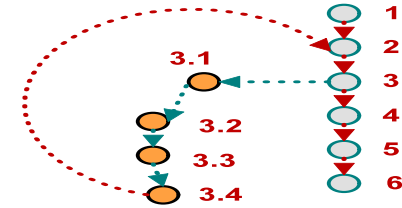
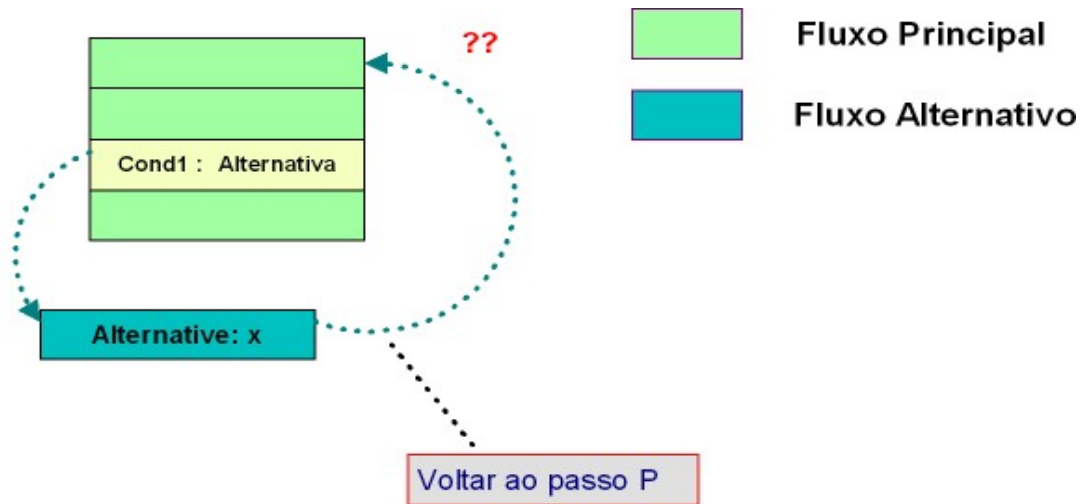
PASSAGEM PARA DS

Entregar obra		
Description		
Diagrams		
Absalom		
8		
F		
Super Use Case		
Author		
Date		
Actors		
Brief Description		
Preconditions		
Post-conditions		
Normal Flow of Events		
Alternative Flow of Events		
Alternative Flow of Events		

loop		
opt		
ref		
Pagar Multa		



PASSAGEM PARA DS



		1	
loop		2	
		3	
opt	[código inválido]		
		4	
		5	
		6	

Fluxo alternativo que
faz GOTO para um
passo anterior

LOOP

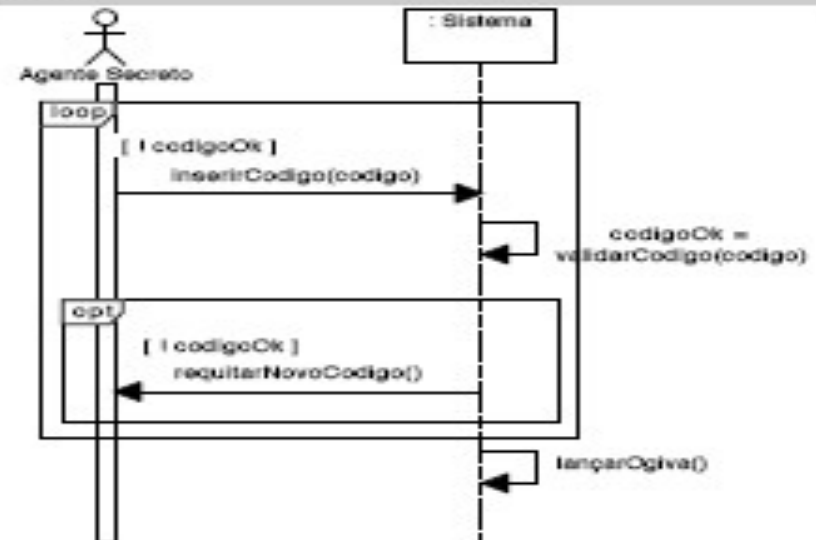
FLUXOS ALTERNATIVOS

Tabela 6.9: Regra de transformação de alternativas cíclicas para DSS

Construção do UC	Construção do DSS
Passo <i>Goto</i> que aponta para um passo anterior ao que originou a alternativa onde se encontra.	Frame <i>Loop</i> que englobe tudo desde o passo referenciado pelo <i>Goto</i> até ao passo da alternativa, inclusivé; estando esta incluída num frame <i>Opt</i> ou <i>Alt</i> (fo-otnote).

Exemplo

1. Agente secreto insere o código.
2. Sistema valida o código.
3. Sistema lança ogiva nuclear.
4. *Success*
- 2a. Código inválido:**
- 2a1. Sistema requisita o código novamente.
- 2a4. *Goto 1*



Situação típica de alternativa que corresponde a um ciclo contendo main flow.

Justificável quando comportamento cíclico faz parte das *regras de negócio*.



285

[illegible]



Especificação de Use Cases

Sumário

- Especificação de Use Cases com Diagramas de Sequência
- Identificação de transações
- Representação do Comportamento base (*Main Flow*)
- Representação de «include» e «extend»
- Representação de Fluxos de Exceção
- Representação de alternativas (com regresso a passos posteriores; com regresso a passos anteriores)