Desenvolvimento de Sistemas Software

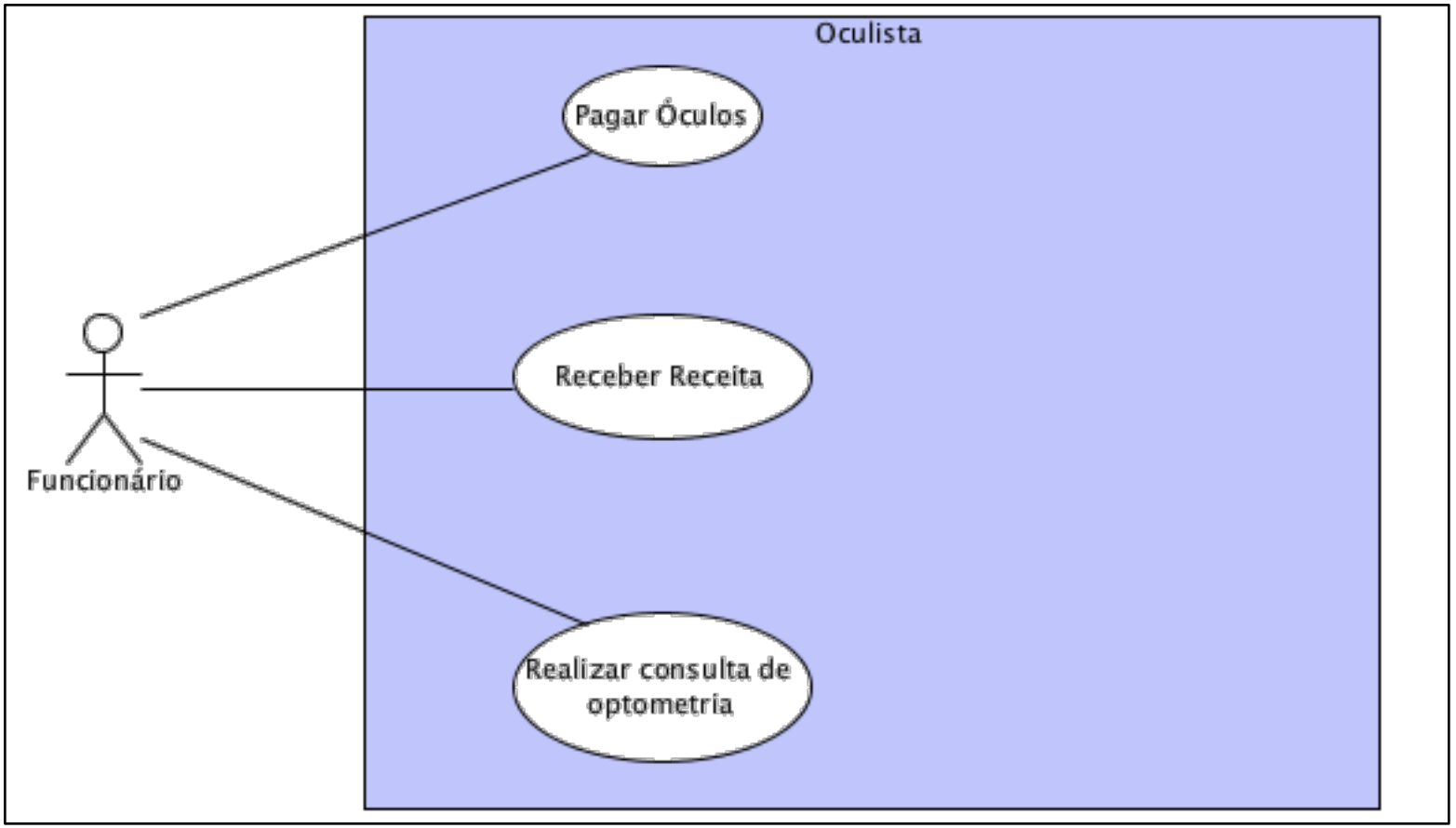
Aula Teórica 7: Diagramas de Use Case (cont.)

"Good use cases are balanced, describing essential system behavior while providing only the necessary details about the interactions between a system and its users"

Patterns for Effective Use Cases

Dependências revisitadas (<<include>> / <<extend>>)

- Mecanismos de estruturação dos modelos
- Adoptar uma abordagem de *refactoring*.



Use Case: Receber receita		
Descrição: Funcionário processa a receita de um cliente		
Pré-condição: Existe papel para imprimir talões		
Pós-condição: Pedido de óculos fica registado		
	Actor	**Sistema**
Comportamento Normal	1. indica nome e/ou data de nascimento do cliente	
		2. apresenta lista de clientes correspondentes
	3. selecciona cliente	
		4. apresenta detalhes do cliente
	5. confirma dados	
	6. indica código de armação e de lentes	
		7. procura produto e apresenta detalhes
	8. confirma produto	
		9. regista reserva e imprime talão
Comp. Alternativo 1 [lista de clientes corresponentes tem tamanho 1] (passo 2)		2.1. apresenta detalhes do único cliente da lista
		2.2 regressa a 5
Comp. Alternativo 2 (passo 3)	3.1. escolhe criar novo cliente	
	3.2. introduz dados do cliente	
		3.3. regista cliente
		3.4 regressa a 6
Excepção 1 (passo 8)	8.1. rejeita produto	
		8.2. cancela reserva

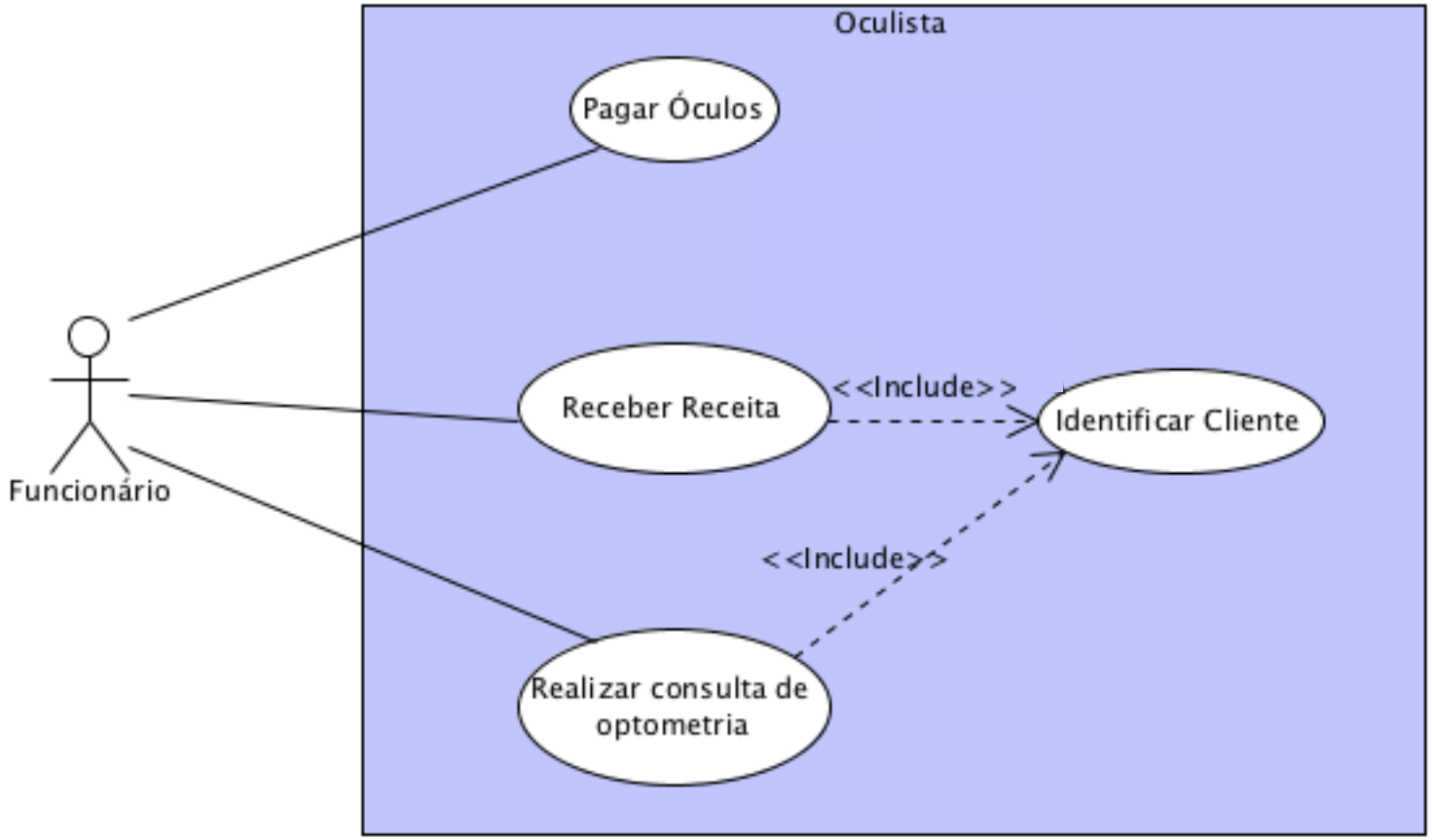
Use Case: Receber receita		
Descrição: Funcionário processa a receita de um cliente		
Pré-condição: Existe papel para imprimir talões		
Pós-condição: Pedido de óculos fica registado		
	Actor	**Sistema**
Comportamento Normal	1. indica nome e/ou data de nascimento do cliente	
		2. apresenta lista de clientes correspondentes
	3. selecciona cliente	
		4. apresenta detalhes do cliente
	5. confirma dados	
	6. indica codigo de armacao e de lentes	
	8. C	
Comp. Alternativo 1 [lista de clientes corresponentes tem tamanho 1] (passo 2)		
Comp. Alternativo 2 (passo 3)	3.1.	
	3.2.	
Excepção 1 (passo 8)	8.1.	

Use Case: Realizar consulta de optometria		
Descrição: Funcionário processa a receita de um cliente		
Pré-condição: Existe papel para imprimir talões		
Pós-condição: Pedido de óculos fica registado		
	Actor	**Sistema**
Comportamento Normal	1. indica nome e/ou data de nascimento do cliente	
		2. apresenta lista de clientes correspondentes
	3. selecciona cliente	
		4. apresenta detalhes do cliente
	5. confirma dados	
		6. determina dívidas do cliente
		7. indica valor total a cobrar (dívidas + consulta)
	8. confirma pagamento	
		9. imprime talão
Comp. Alternativo 1 [lista de clientes corresponentes tem tamanho 1] (passo 2)		2.1. apresenta detalhes do único cliente da lista
		2.2 regressa a 5
Comp. Alternativo 2 (passo 3)	3.1. escolhe criar novo cliente	
	3.2. introduz dados do cliente	
		3.3. regista cliente
		3.4. regressa a 5
Excepção 1 (passo 8)	8.1. indica não pagamento	
		8.2. regista dívida na ficha de cliente

Use Case: Identificar cliente		
Descrição: Identificação de um cliente por nome e/ou data de nascimento		
Pré-condição:		
Pós-condição: Cliente pretendido fica seleccionado		
	Actor	**Sistema**
Comportamento Normal	1. indica nome e/ou data de nascimento do cliente	
		2. apresenta lista de clientes correspondentes
	3. selecciona cliente	
		4. apresenta detalhes do cliente
	5. confirma dados	
Comp. Alternativo 1 [lista de clientes correspondentes tem tamanho 1] (passo 2)	2.1. apresenta detalhes do único cliente da lista	
		2.2. regressa a 5
Comp. Alternativo 2 (passo 3)	3.1. escolhe criar novo cliente	
	3.2. introduz dados do cliente	
		3.3. regista cliente

Use Case: Receber receita	
Descrição: Funcionário processa a receita de um cliente	
Pré-condição: Existe papel para imprimir talões	
Pós-condição: Pedido de óculos fica registado	
	Actor
Comportamento Normal	1. <<i
	2. ind
	4. Co
Excepção 1 (passo 4)	4.1. n

Use Case: Realizar consulta de optometria		
Descrição: Funcionário cobra a realização de uma consulta		
Pré-condição: Existe papel para imprimir talões		
Pós-condição: Consulta fica paga		
	Actor	**Sistema**
Comportamento Normal	1. <<include>> identificar cliente	
		2. determina dívidas do cliente
		3. indica valor total a cobrar (dívidas + consulta)
	4. confirma pagamento	
		5. regista reserva e imprime talão
Excepção 1	4.1. não confirma pagamento	
		4.2. regista dívida na ficha de cliente


Oculista
Pagar Óculos
Receber Receita
<<Include>>
Identificar Cliente
<<Include>>
Realizar consulta de optometria
Funcionário

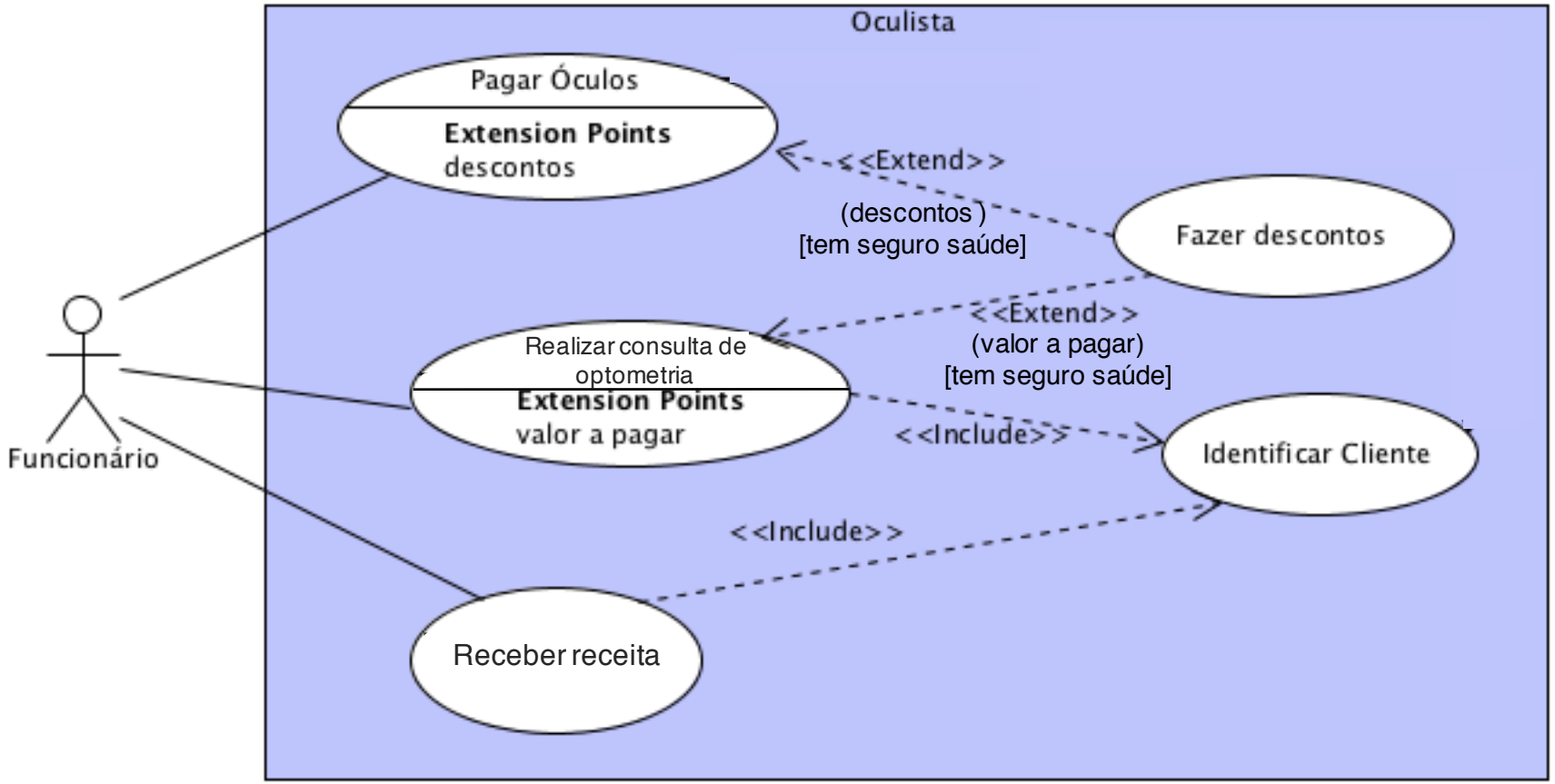
Use Case: Pagar óculos		
...		
	Actor	**Sistema**
Comportamento Normal	1. indica número de talão de óculos a pagar	
		2. determina dívidas do cliente
		3. indica valor total a cobrar (dívidas + óculos)
	4. confirma pagamento	
		5. regista pagamento e imprime talão
Excepção 1 (passo 4)	4.1. indica não pagamento	
		4.2. anula entrega

Use Case: Realizar consulta de optometria		
...		
	Actor	**Sistema**
Comportamento Normal	1. <<include>> identificar cliente	
		2. determina dívidas do cliente
		3. indica valor total a cobrar (dívidas + consulta)
	4. confirma pagamento	
		5. regista reserva e imprime talão
Excepção 1 (passo 4)	4.1. indica não pagamento	
		4.2. regista dívida na ficha de cliente

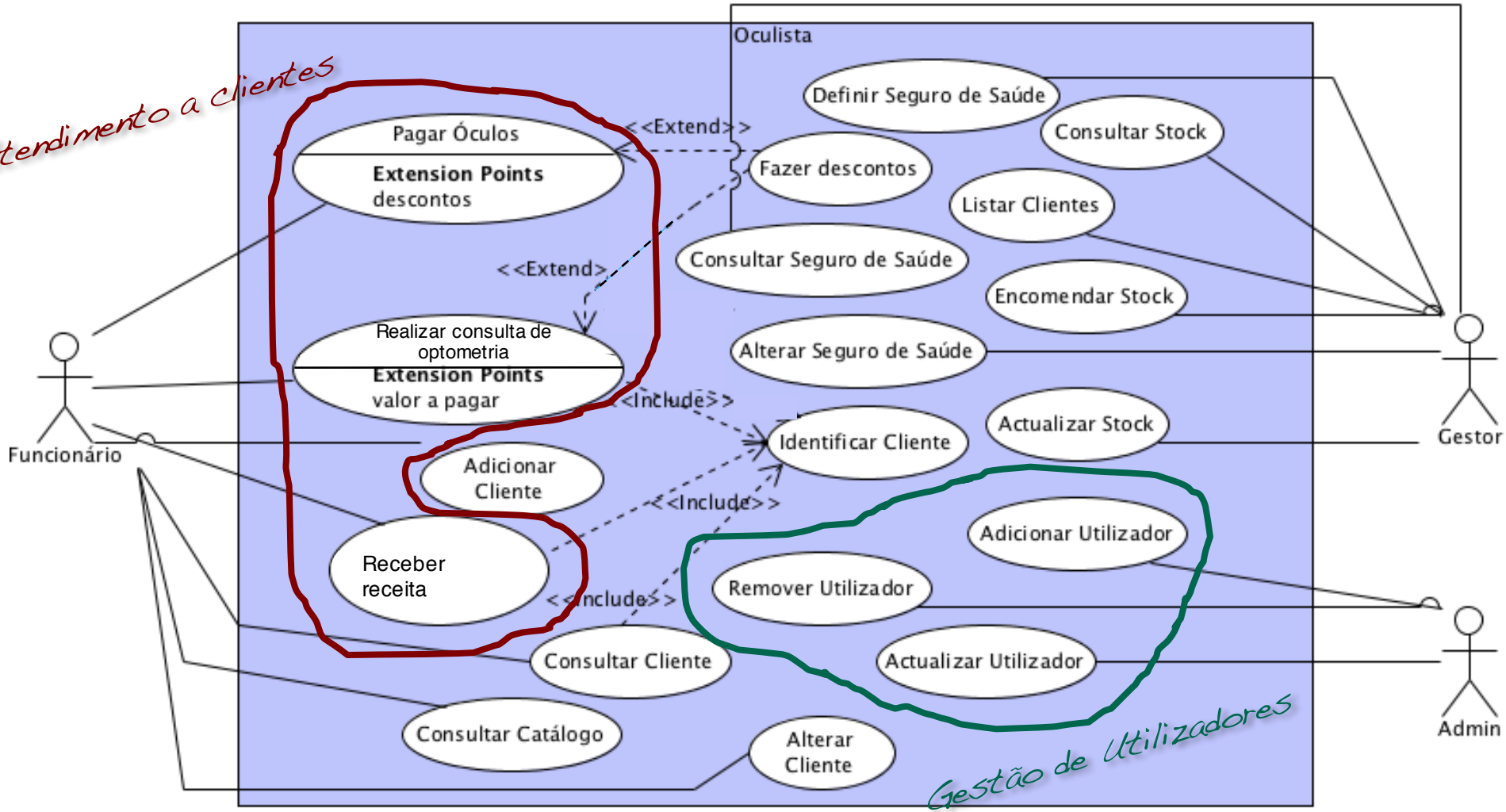
Use Case: Pagar óculos		
...		
	Actor	**Sistema**
Comportamento Normal	1. indica número de talão de óculos a pagar	
		2. determina dívidas do cliente
		3. indica valor total a cobrar (dívidas + óculos) [ponto de extensão: descontos]
	4. confirma pagamento	
		5. regista pagamento e imprime talão
Excepção 1 (passo 4)	4.1. indica não pagamento	
		4.2. anula entrega

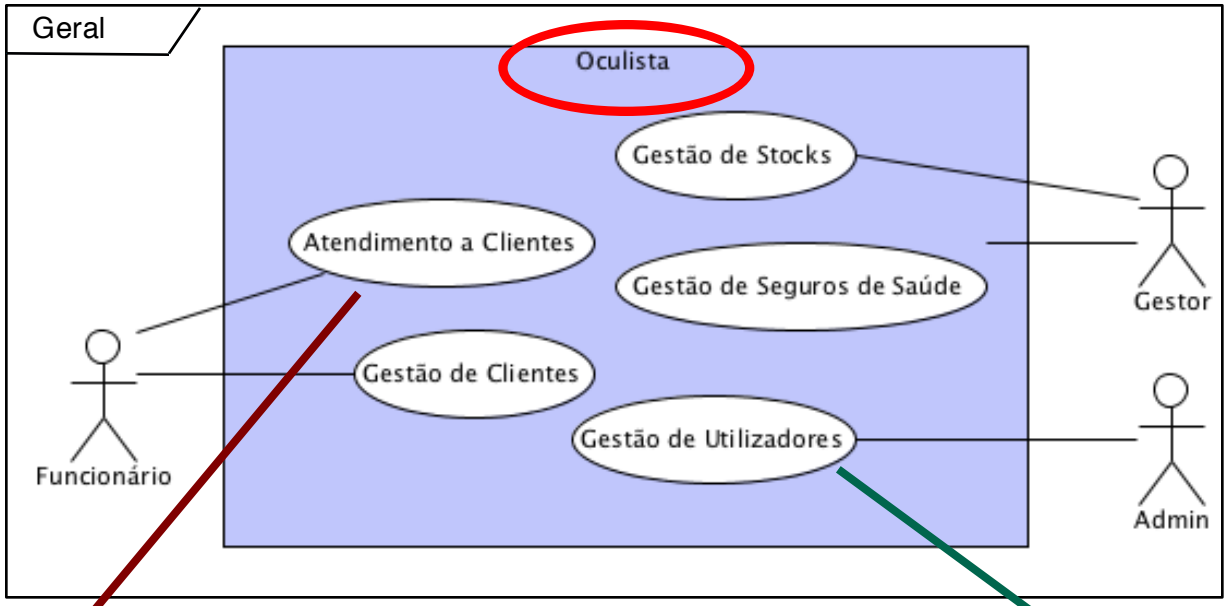
Use Case: Realizar consulta de optometria		
...		
	Actor	**Sistema**
Comportamento Normal	1. <<include>> identificar cliente	
		2. determina dívidas do cliente
		3. indica valor total a cobrar (dívidas + consulta) [ponto de extensão: valor a pagar]
	4. confirma pagamento	
		5. regista reserva e imprime talão
Excepção 1 (passo 4)	4.1. indica não pagamento	
		4.2. regista dívida na ficha de cliente

Use Case: Fazer descontos		
...		
	Actor	**Sistema**
Comportamento Normal		1. pede confirmação de dados do seguro
	2. confirma dados	
Alternativa 1 [dados inválidos] (passo 2)	2.1. não confirma dados	
		2.2. propõe preço normal
	2.3. aceita preço normal	
Excepção 1 (passo 2.3)	2.3.1. não aceita peço normal	

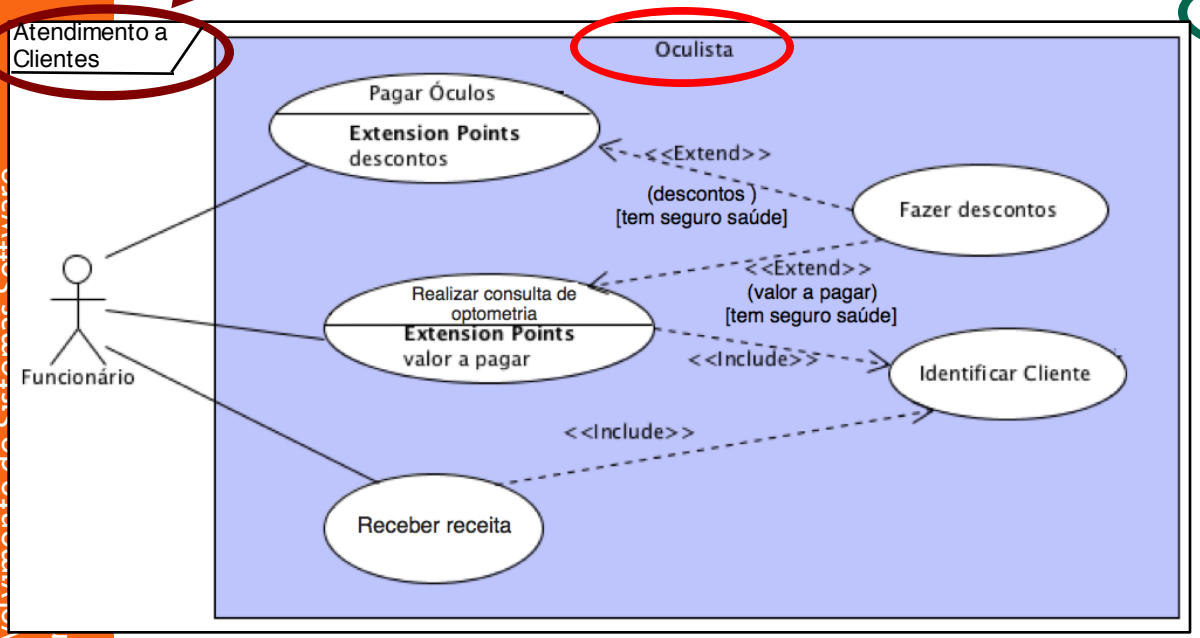

Oculista
Pagar Óculos
Extension Points
descontos
<<Extend>>
(descontos)
[tem seguro saúde]
Fazer descontos
<<Extend>>
(valor a pagar)
[tem seguro saúde]
Realizar consulta de optometria
Extension Points
valor a pagar
<<Include>>
Identificar Cliente
<<Include>>
Receber receita
Funcionário

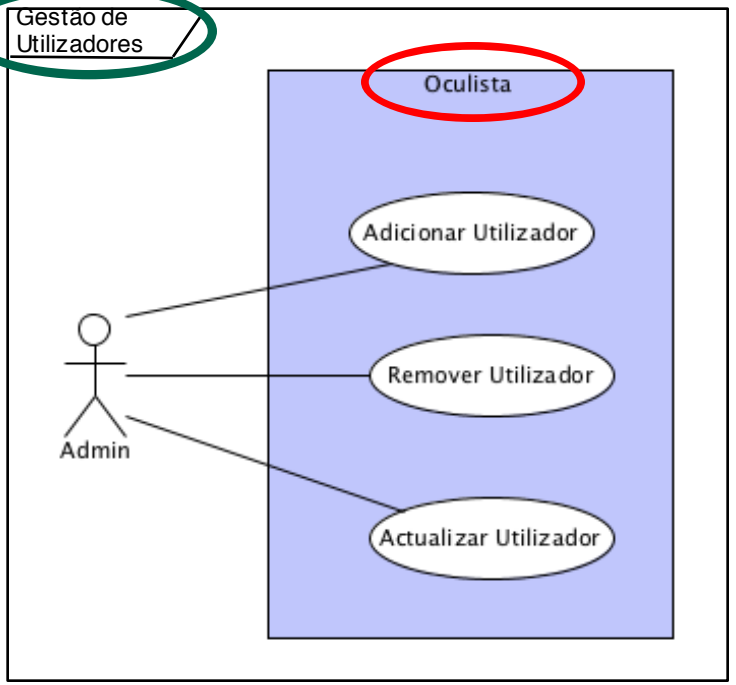
Estruturação de modelos?



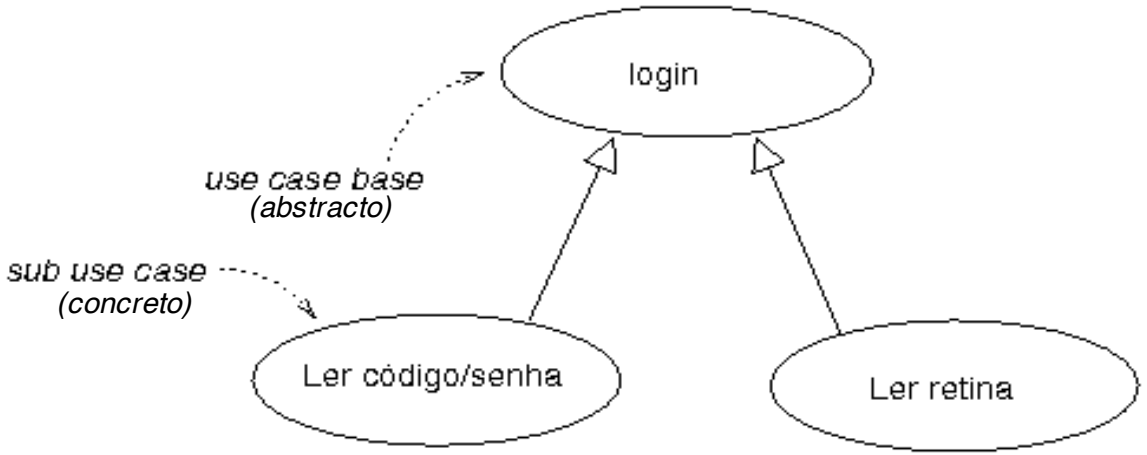


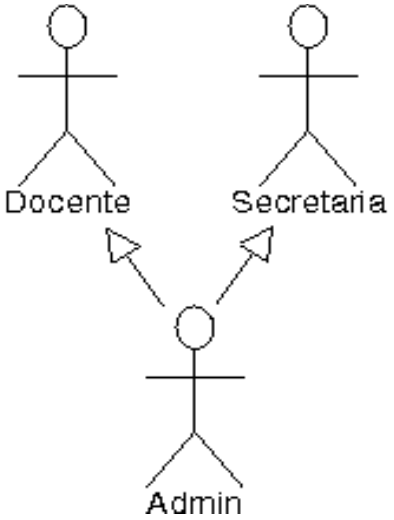
Três diagramas, um modelo!





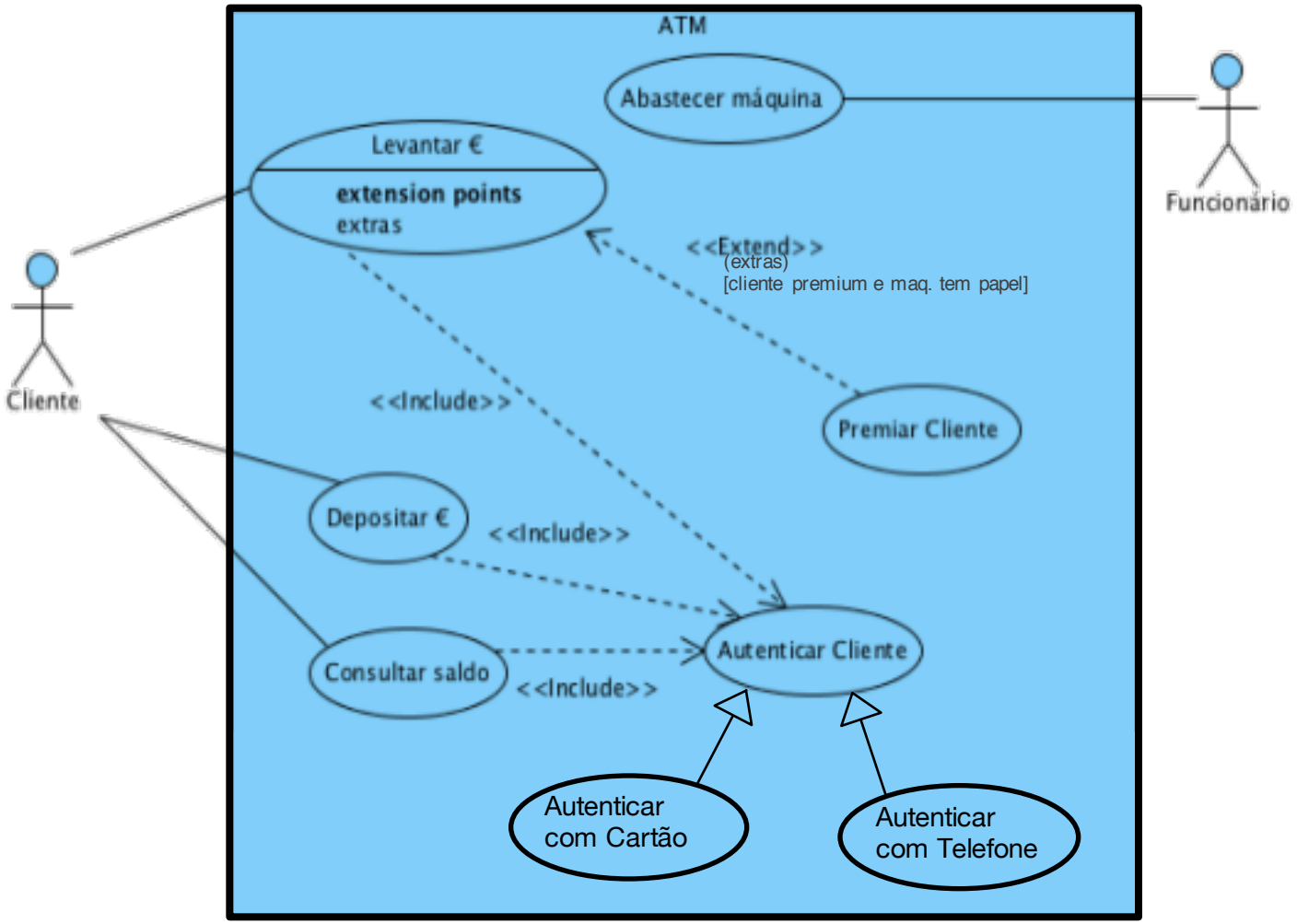
Diagramas de UC - Generalização/Especialização





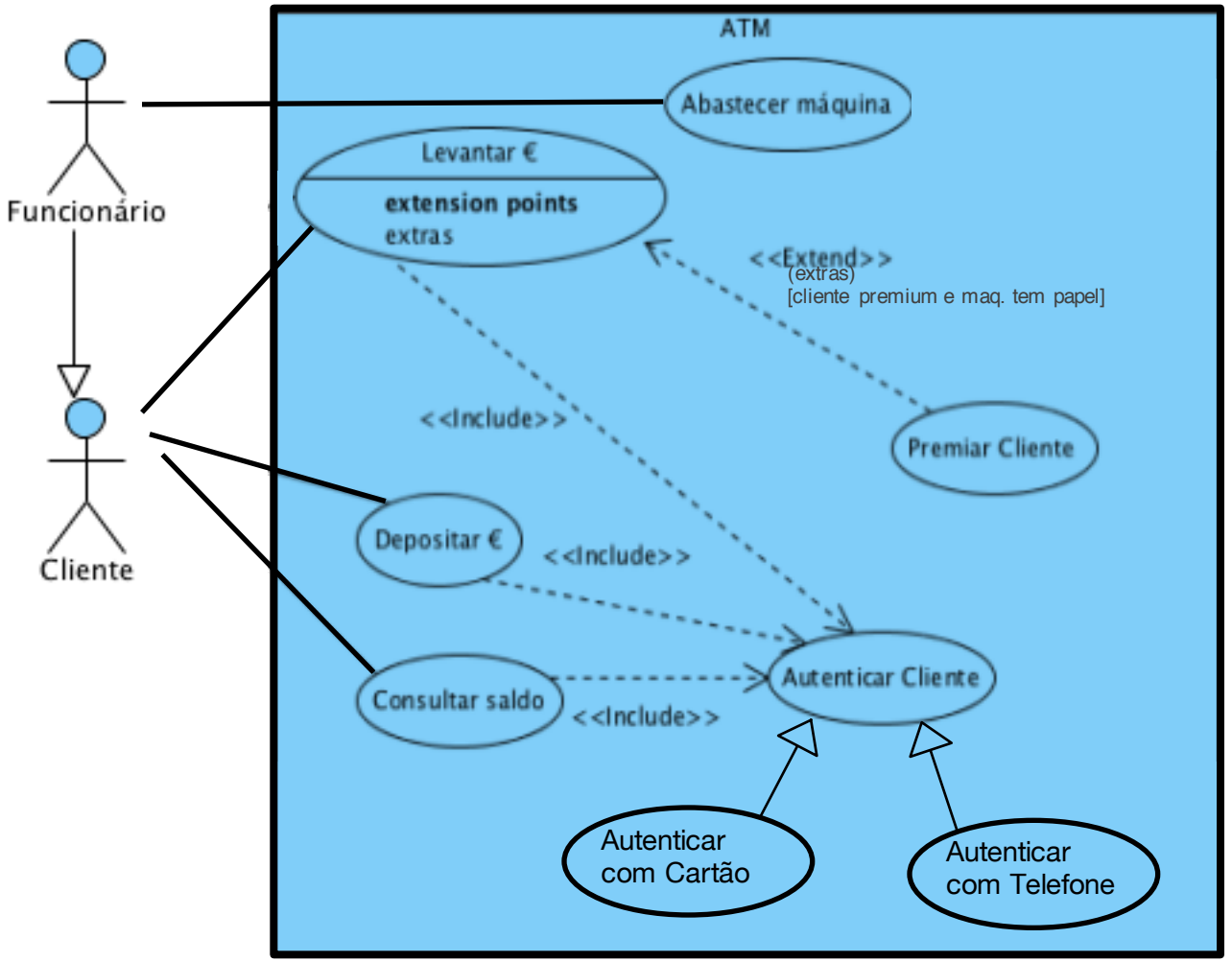
- Sub-elementos são casos particulares de super-elementos.
- Um sub-elemento pode ser utilizado onde quer que o super-elemento possa.
- No case dos Use Case, útil para organizar Use Cases por tipo.
- No caso dos Actores, útil para *user profiling* (definição de níveis de acesso).
- Nos exemplos apresentados:
  - Existem duas formas de fazer login.
  - O actor Admin pode realizar todos os *use cases* de Docente e Secretaria.

Generalização de Use Cases



- O cliente pode realizar qualquer uma das formas de autenticação.

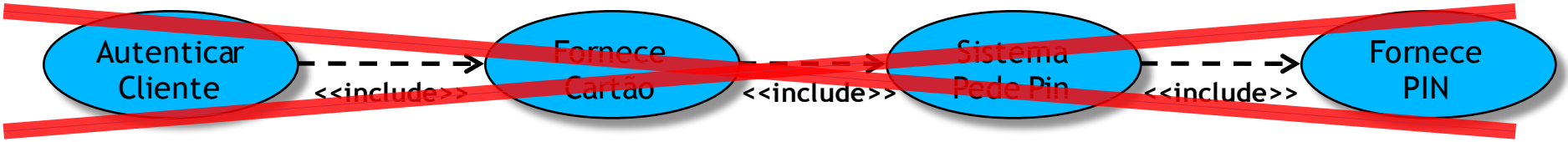
Generalização de Actores



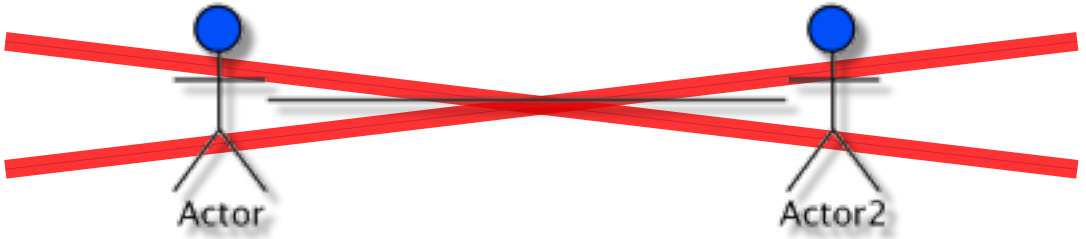
- Tudo o que o cliente pode fazer, o Funcionário também pode.
- **Fará sentido?** O Funcionário acede pelo interior (quando vem cá fora, é Cliente?)

Alguns aspectos a ter em atenção…

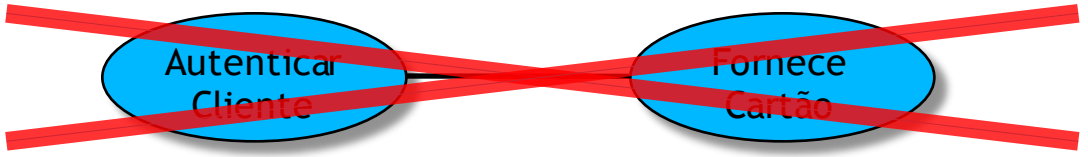
- Modelo de Use Cases não representa fluxo de dados



- Modelo de Use Cases não representa comunicação entre actores



- Modelo de Use Cases não representa comunicação entre use cases



- include, extends e generalização devem ser utilizados com moderação!

Resumindo

- Um use case descreve as sequências de interacções entre actores externos e o sistema em projecto, sendo este visto como uma *black box*, para que um dado objectivo ou tarefa se realize.
- Devem ser especificados num use case quer os cenários de sucesso (tarefa completada com êxito) quer os de insucesso (tarefa não completada);
- Cada passo de interacção actor-sistema descrito num use case designa-se por *evento*, *acção* ou *operação*, e devem distinguir-se quanto à sua origem (actor ou sistema);
- Um use case descreve um fluxo principal de eventos/operações, designado *fluxo principal* ou *cenário principal*, bem como outros possíveis fluxos ou caminhos designados *fluxos allernativos* ou *cenários alternativos*, bem ainda como fluxos que, sendo alternativos, conduzem a situações de insucesso, a que chamaremos *fluxos de excepção* ou apenas *excepções*;
- Use cases são multi-nível, ou seja, use cases podem ser especificados usando a funcionalidade especificada noutros use cases através de relações definidas em Ume., designadamente, inclusão, extensão e generalização;
- A generalização é tambétn aplicável aos actores, desta forma sendo possível representar o relacionamento entre actores/papéis perante o sistema;
- Use cases devem ser simples e legíveis, não devem conter detalhes sobre a interface com o utilizador e devem ter o nível de detalhe necessário a cada iteração de requisitos (são refináveis);
- Use cases relacionados com actores devem ser identificados por verbos no infinitivo, deixando claro qual a tarefa que o sistema deve fornecer ao actor.

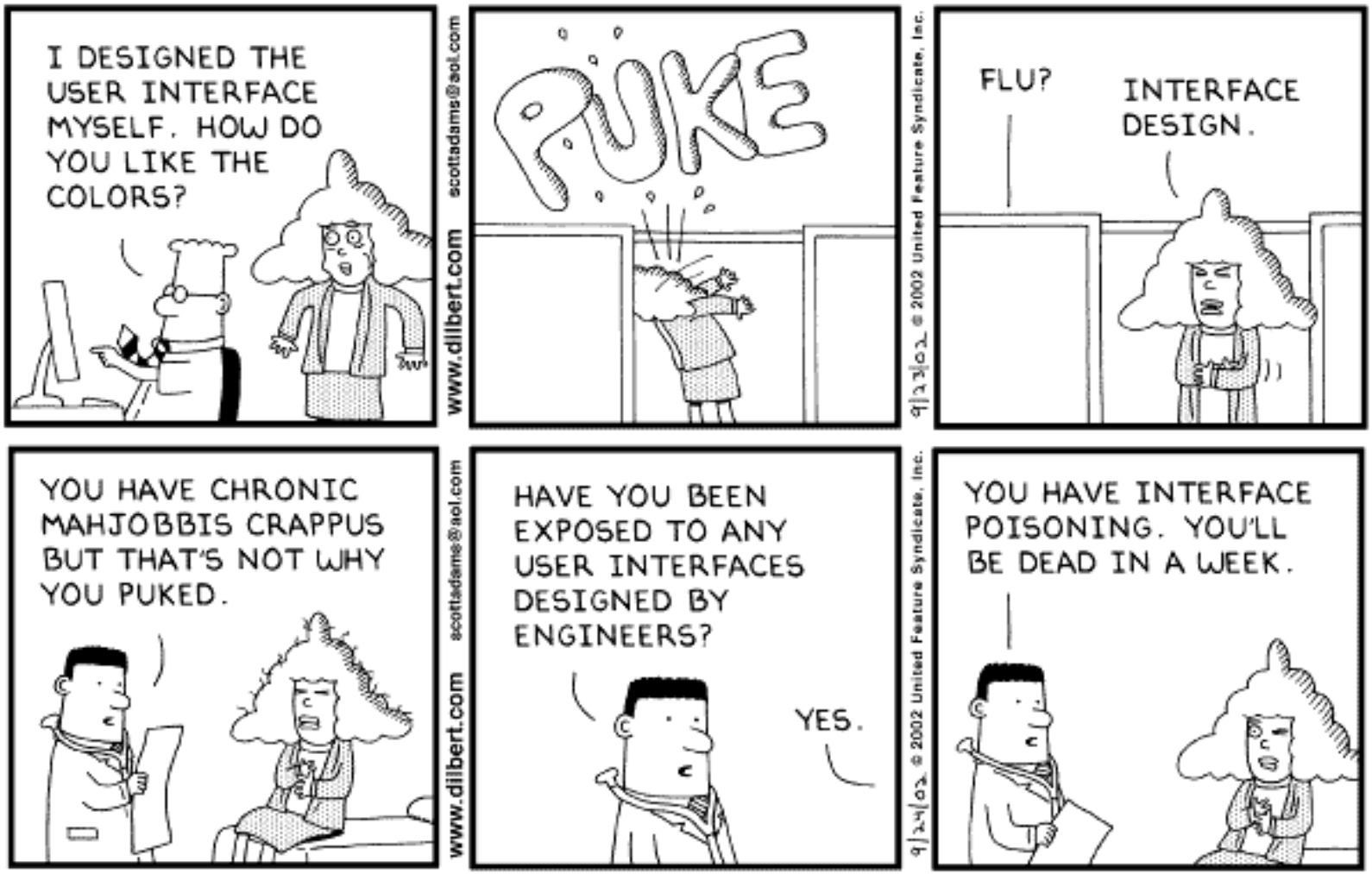
Requisitos funcionais

Requisitos funcionais permitem definir o que o sistema deverá fazer.

Após análise de requisitos, UP propõe a concepção do sistema começando pela arquitectura.

O que falta!?

Interface com o utilizador?



Diagramas de Use Case

Sumário:

- Valor dos Use Case no processo de desenvolvimento
- Dependências entre Use Cases (<<include>> / <<extend>>)
- Estruturação de modelos de Use Case
- Notas finais