



# Desenvolvimento de Sistemas Software

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



### ▶ Hunting for use-case scenarios

#### Part I: Analyzing customer psychology

by [Pan-Wei Ng](#)

Independent Consultant

*Use cases, an increasingly popular technique, are often the basis for successful projects. One reason the technique is so powerful is that use cases force both user representatives and analysts to explore system usage scenarios -- and reach consensus on how to handle them -- early in the project lifecycle. This is crucial to project success, as these same scenarios also drive the analysis, design, and test processes. However, hunting down every possible user or system behavior is not simple; it requires lateral thinking and venturing out of the box. Frequently, practitioners give up too easily and simply let use cases terminate.*



(The Rational edge, Out 2003)



# Especificação de *Use Cases* - excepções

Use Case: Levantar €

Descrição: Cliente levanta quantia da máquina

Pré-condição: Sistema tem notas

Pós-condição: Cliente tem quantia desejada e saldo da conta foi actualizado

Comportamento normal:

1. Actor fornece cartão
2. Sistema pede PIN
3. Actor indica PIN
4. Sistema valida acesso e apresenta opções
5. Cliente indica que pretende levantar dada quantia
6. Sistema processa levantamento da quantia
7. Sistema fornece quantia, talão e devolve cartão
8. Cliente retira notas, talão e cartão

Comportamento Alternativo [sem papel]:

- 4.1. Sistema avisa de impossibilidade de emitir talão e p
- 4.2. Cliente diz que sim
- 4.3. Sistema processa levantamento da quantia
- 4.4. Sistema fornece quantia e devolve cartão
- 4.5. Cliente retira notas e cartão

Excepção [saldo de conta insuficiente]:

- 4.1. Sistema avisa de impossibilidade de levantar quanti
- 4.2. Sistema fornece cartão
- 4.3. Cliente retira cartão

O que falta?

- PIN inválido?
- Cartão ilegível?
- € insuficiente na máquina?
- Valor diário excedido?
- Cliente não aceita não receber talão?
- Quantia impossível com notas existentes?
- Cliente pode desistir?
- Ligação ao servidor cai? (!)
- Dispensador de notas encrava? (!)
- ...



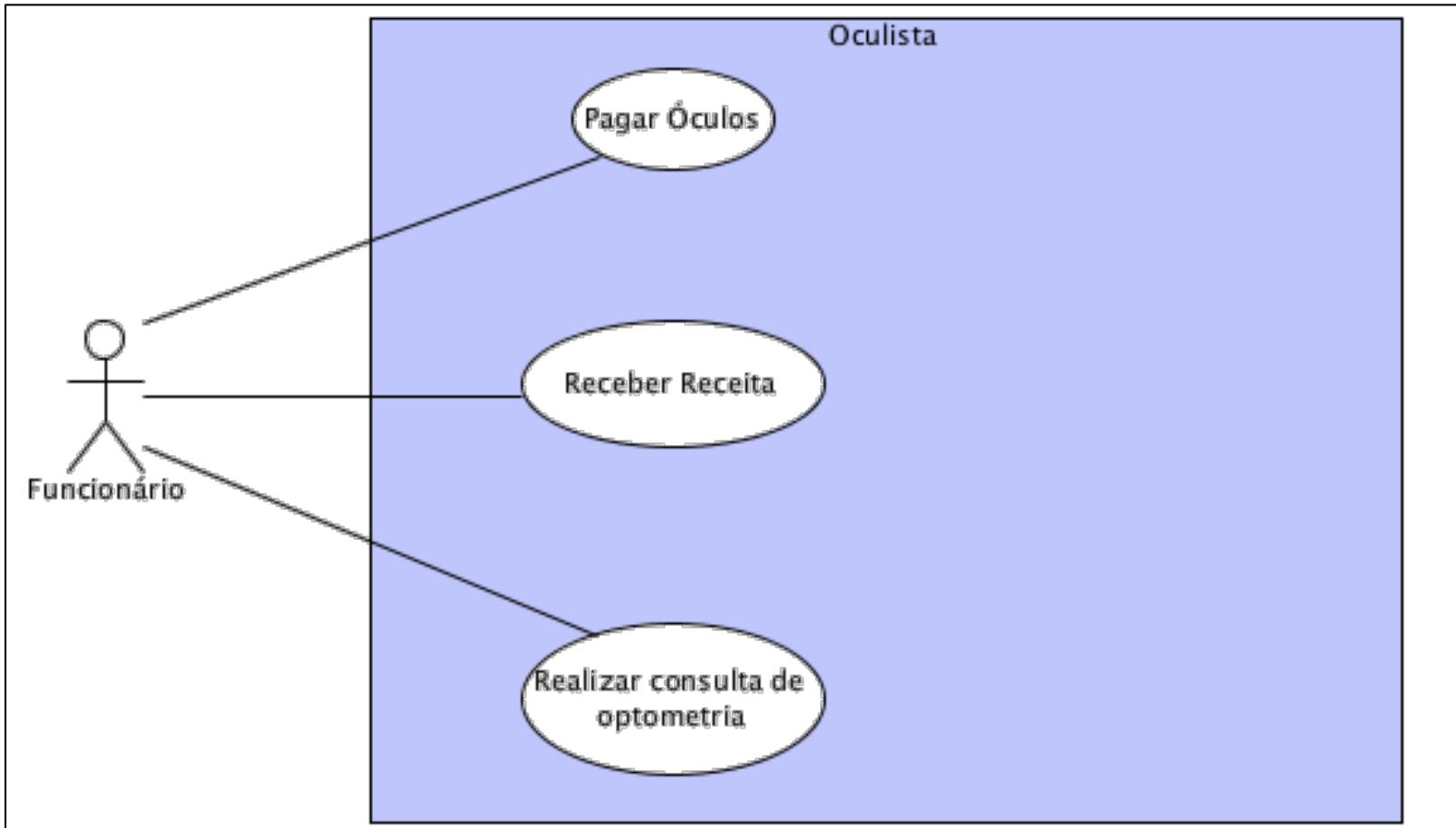
“Good use cases are balanced, describing essential system behavior while providing only the necessary details about the interactions between a system and its users”

*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

Comportamento normal:

1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema apresenta lista de clientes correspondentes
3. Funcionário selecciona cliente
4. Sistema apresenta detalhes do cliente
5. Funcionário confirma dados
6. Funcionário indica código de armação e de lente
7. Sistema procura produto e apresenta detalhes
8. Funcionário confirma
9. Sistema regista reserva e imprime talão

Comportamento Alternativo [lista de clientes correspondentes tem tamanho 1]

- 2.1. Sistema apresenta detalhes do único cliente
- 2.2. regressa a 5

Comportamento Alternativo

- 3.1. Funcionário escolhe criar novo cliente
- 3.2. Funcionário introduz dados do cliente
- 3.3. Sistema regista cliente
- 3.4. regressa a 6

Excepção

- 8.1. Funcionário não confirma produto
- 8.2. Sistema cancela reserva

### Use Case: Realizar consulta de optometria

Descrição: Funcionário cobra a realização de uma consulta

Pré-condição: Existe papel para imprimir talões

Pós-condição: Consulta é feita e paga

Comportamento normal:

1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema apresenta lista de clientes correspondentes
3. Funcionário selecciona cliente
4. Sistema apresenta detalhes do cliente
5. Funcionário confirma dados
6. Sistema determina dívidas do cliente
7. Sistema indica valor total a cobrar (dívidas + consulta)
8. Funcionário confirma pagamento
9. Sistema regista reserva e imprime talão

Comportamento Alternativo [lista de clientes correspondentes tem tamanho 1]

- 2.1. Sistema apresenta detalhes do único cliente da lista
- 2.2. regressa a 5

Comportamento Alternativo

- 3.1. Funcionário escolhe criar novo cliente
- 3.2. Funcionário introduz dados do cliente
- 3.3. Sistema regista cliente
- 3.4. regressa a 6

Excepção

- 8.1. Funcionário não confirma pagamento
- 8.2. Sistema regista dívida na ficha de 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

Comportamento normal:

1. <<include>> identificar cliente
2. Funcionário indica código de armação e de lentes
3. Sistema procura produto e apresenta detalhes
4. Funcionário confirma
5. Sistema regista reserva e imprime talão

Excepção

- 4.1. Funcionário não confirma produto
- 4.2. Sistema cancela reserva

### 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

Comportamento normal:

1. <<include>> identificar cliente
2. Sistema determina dívidas do cliente
3. Sistema indica valor total a cobrar (dívidas + consulta)
4. Funcionário confirma pagamento
5. Sistema regista reserva e imprime talão

Excepção

- 4.1. Funcionário não confirma pagamento
- 4.2. Sistema regista dívida na ficha de cliente

### Use Case: Identificar cliente

Descrição: identificação de um cliente por nome

Pré-condição:

Pós-condição: Cliente pretendido fica seleccionado

Comportamento normal:

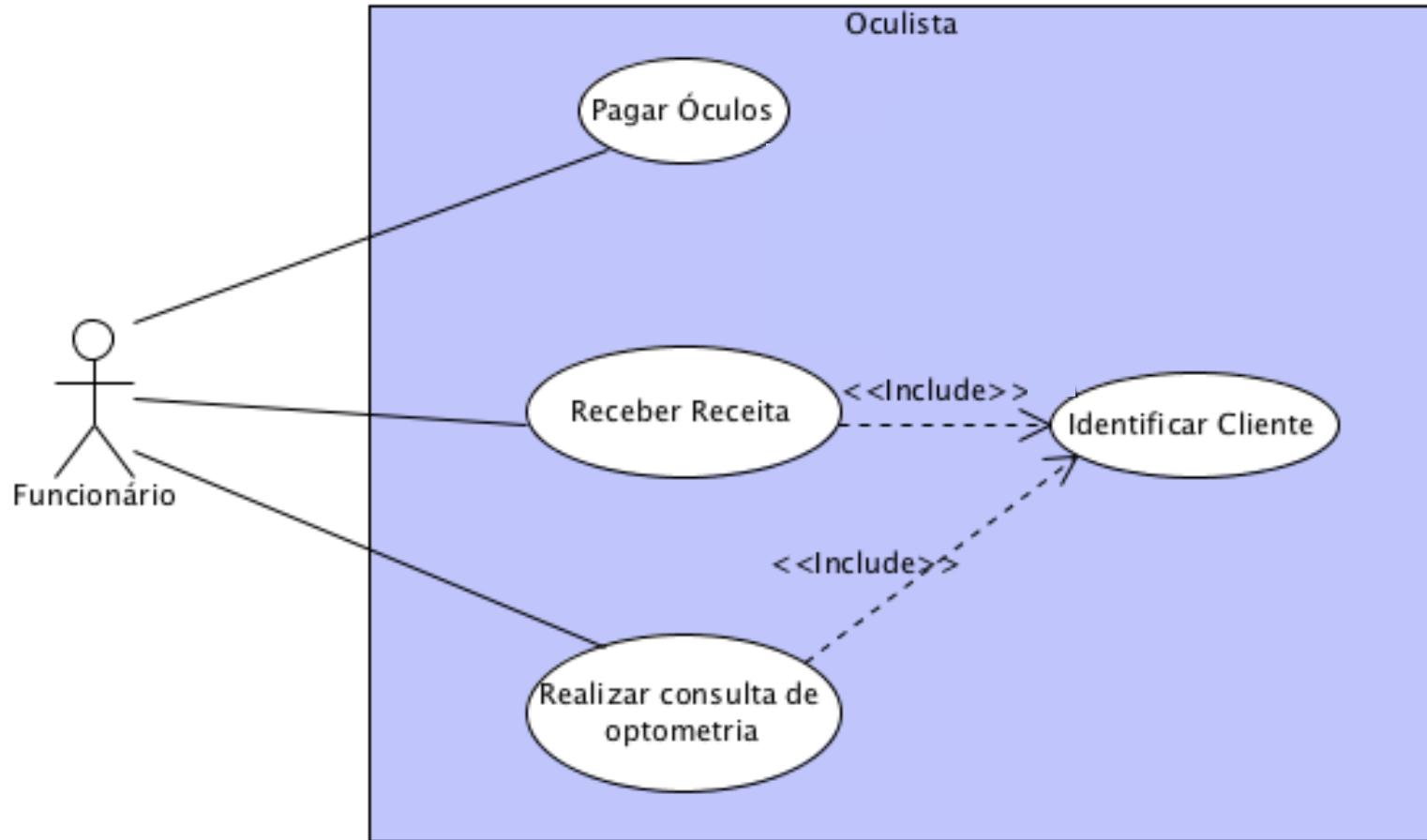
1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema apresenta lista de clientes correspondentes
3. Funcionário selecciona cliente
4. Sistema apresenta detalhes do cliente
5. Funcionário confirma dados

Comportamento Alternativo [lista de clientes correspondentes tem tamanho 1]

- 2.1. Sistema apresenta detalhes do único cliente da lista
- 2.2. regressa a 5

Comportamento Alternativo

- 3.1. Funcionário escolhe criar novo cliente
- 3.2. Funcionário introduz dados do cliente
- 3.3. Sistema regista cliente



## 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

Comportamento normal:

1. <<include>> identificar cliente
2. Sistema determina dívidas do cliente
3. Sistema indica valor total a cobrar (dívidas + consulta)
4. Funcionário confirma pagamento
5. Sistema regista reserva e imprime talão

Excepção

- 4.1. Funcionário não confirma pagamento
- 4.2. Sistema regista dívida na ficha de cliente

Alternativa [cliente tem seguro de saúde]

- 3.1. Sistema pede confirmação de dados do seguro
- 3.2. Funcionário confirma dados
- 3.3. regressa a 3

Alternativa [dados inválidos]

- 3.2.1. Funcionário não confirma dados
- 3.2.2. Sistema propõe preço normal
- 3.2.3. Funcionário aceita
- 3.2.4. regressa a 3.3

Excepção [não aceita preço normal]

- 3.2.3.1. Funcionário não aceita preço normal

## Use Case: Pagar óculos

Descrição: Funcionário cobra um par de óculos

Pré-condição: Existe papel para imprimir talões

Pós-condição: Óculos ficam pagos

Comportamento normal:

1. Funcionários indica número de talão de óculos a pagar
2. Sistema determina dívidas do cliente
3. Sistema indica valor total a cobrar (dívidas + óculos)
4. Funcionário confirma pagamento
5. Sistema regista pagamento e imprime talão

Excepção

- 4.1. Funcionário não confirma pagamento
- 4.2. Sistema regista dívida na ficha de cliente

Alternativa [cliente tem seguro de saúde]

- 3.1. Sistema pede confirmação de dados do seguro
- 3.2. Funcionário confirma dados
- 3.3. regressa a 3

Alternativa [dados inválidos]

- 3.2.1. Funcionário não confirma dados
- 3.2.2. Sistema propõe preço normal
- 3.2.3. Funcionário aceita
- 3.2.4. regressa a 3.3

Excepção [não aceita preço normal]

- 3.2.3.1. Funcionário não aceita preço normal

## 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

Comportamento normal:

1. <<include>> identificar cliente
2. Sistema determina dívidas do cliente
3. Sistema indica valor total a cobrar (dívidas + consulta)  
[ponto de extensão: valor a pagar]
4. Funcionário confirma pagamento
5. Sistema regista reserva e imprime talão

Excepção

- 4.1. Funcionário não confirma pagamento
- 4.2. Sistema regista dívida na ficha de cliente

Ponto de Extensão: valor a pagar [cliente tem seguro de saúde]

3. Extended by: Fazer desconto

## Use Case: Pagar óculos

Descrição: Funcionário cobra um par de óculos

Pré-condição: Existe papel para imprimir talões

Pós-condição: Óculos ficam pagos

Comportamento normal:

1. Funcionários indica número de talão de óculos a pagar
2. Sistema determina dívidas do cliente
3. Sistema indica valor total a cobrar (dívidas + óculos)  
[ponto de extensão: descontos]
4. Funcionário confirma pagamento
5. Sistema regista pagamento e imprime talão

Excepção

- 4.1. Funcionário não confirma pagamento
- 4.2. Sistema regista dívida na ficha de cliente

Ponto de Extensão: descontos [cliente tem seguro de saúde]

4. Extended by: Fazer desconto

## Use Case: Fazer desconto

Descrição: Funcionário cobra a realização de uma consulta

Pré-condição:

Pós-condição: preço fica calculado com desconto

Comportamento normal:

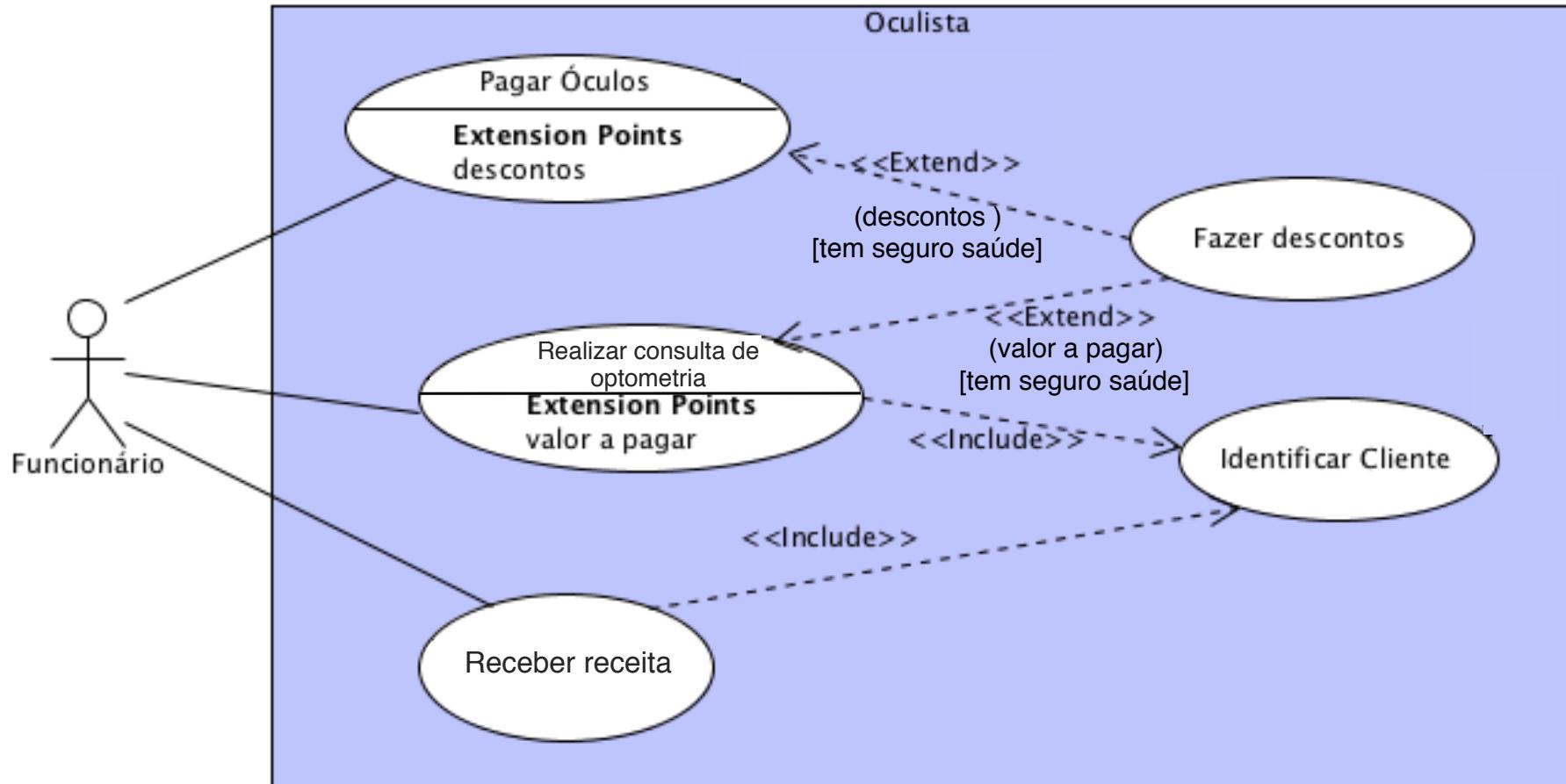
1. Sistema pede confirmação de dados do seguro
2. Funcionário confirma dados

Alternativa [dados inválidos]

- 2.1. Funcionário não confirma dados
- 2.2. Sistema propõe preço normal
- 2.3. Funcionário aceita

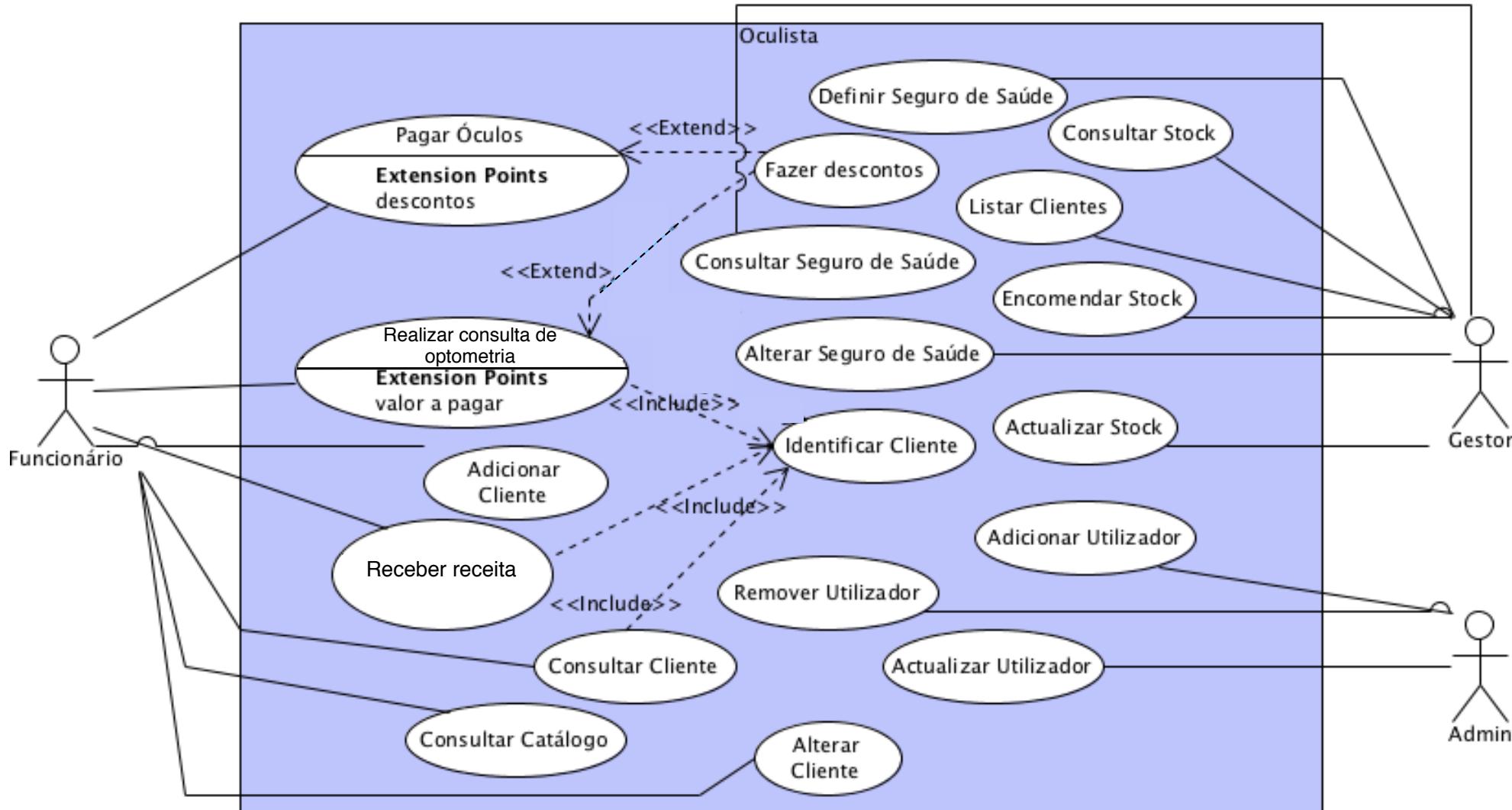
Excepção [não aceita preço normal]

- 2.3.1. Funcionário não aceita preço normal





# Estruturação de modelos?





177

Martins

Gestão de Utilizadores

Funcionário

Desenv  
António

Geral

Oculista

Gestão de Stocks

Atendimento a Clientes

Gestão de Seguros de Saúde

Gestão de Clientes

Gestão de Utilizadores

Funcionário

Gestor

Admin

Três diagramas,  
um modelo!

Oculista

Adicionar Utilizador

Remover Utilizador

Actualizar Utilizador

Gestão de Utilizadores

Admin

Oculista

Pagar Óculos

Extension Points  
descontos

Fazer descontos

Realizar consulta de  
optometriaExtension Points  
valor a pagar<<Extend>>  
(descontos)  
[tem seguro saúde]<<Extend>>  
(valor a pagar)  
[tem seguro saúde]

&lt;&lt;Include&gt;&gt; &gt;

Identificar Cliente

&lt;&lt;Include&gt;&gt;

Receber receita



- a) 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.
- b) Devem ser especificados num use case quer os cenários de sucesso (tarefa completada com êxito) quer os de insucesso (tarefa não completada);
- c) 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);
- d) 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 alternativos* ou *cenários alternativos*  
bem ainda como fluxos que, sendo alternativos, conduzem a situações de insucesso, a que chamaremos *fluxos de excepção* ou apenas *excepções*;
- e) Use cases são multi-nível, ou seja, use cases podem ser especificados usando a funcionalidade especificada noutras use cases através de relações definidas em UML, designadamente, inclusão, extensão e generalização;
- f) A generalização é também aplicável aos actores, desta forma sendo possível representar o relacionamento entre actores/papéis perante o sistema;
- g) 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);
- h) Use cases relacionados com actores devem ser identificados por verbos no infinitivo e sujeitos, deixando claro qual a tarefa que o sistema deve fornecer ao actor.



# Requisitos funcionais

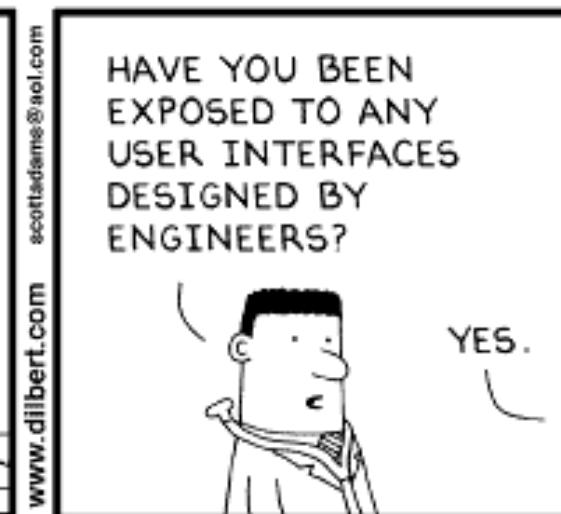
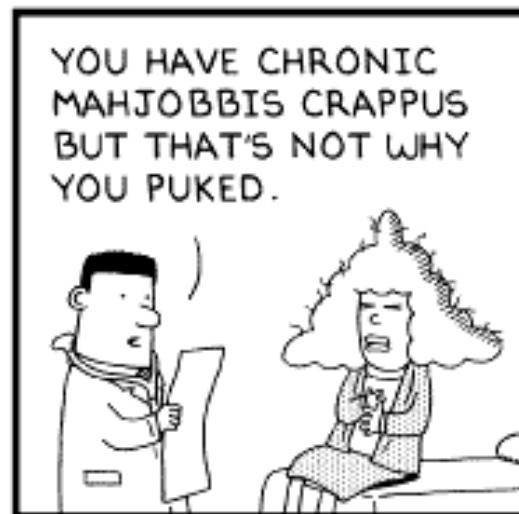
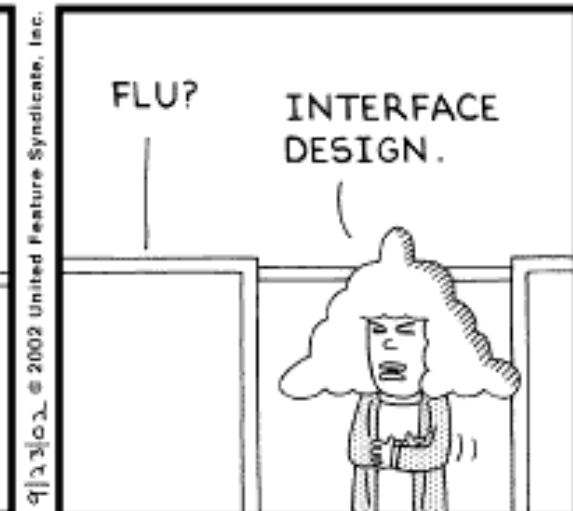
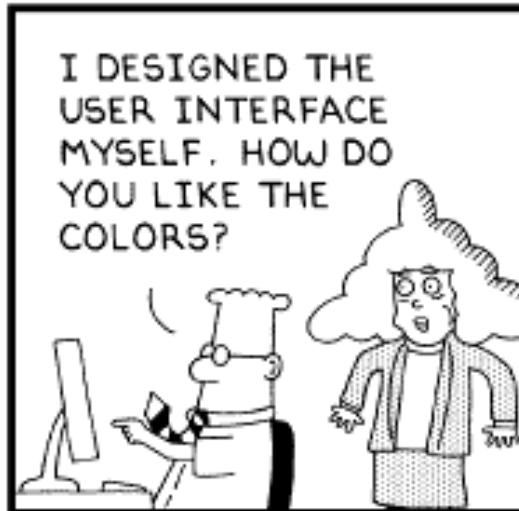


Requisitos funcionais permitem definir o que o sistema deverá fazer.

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

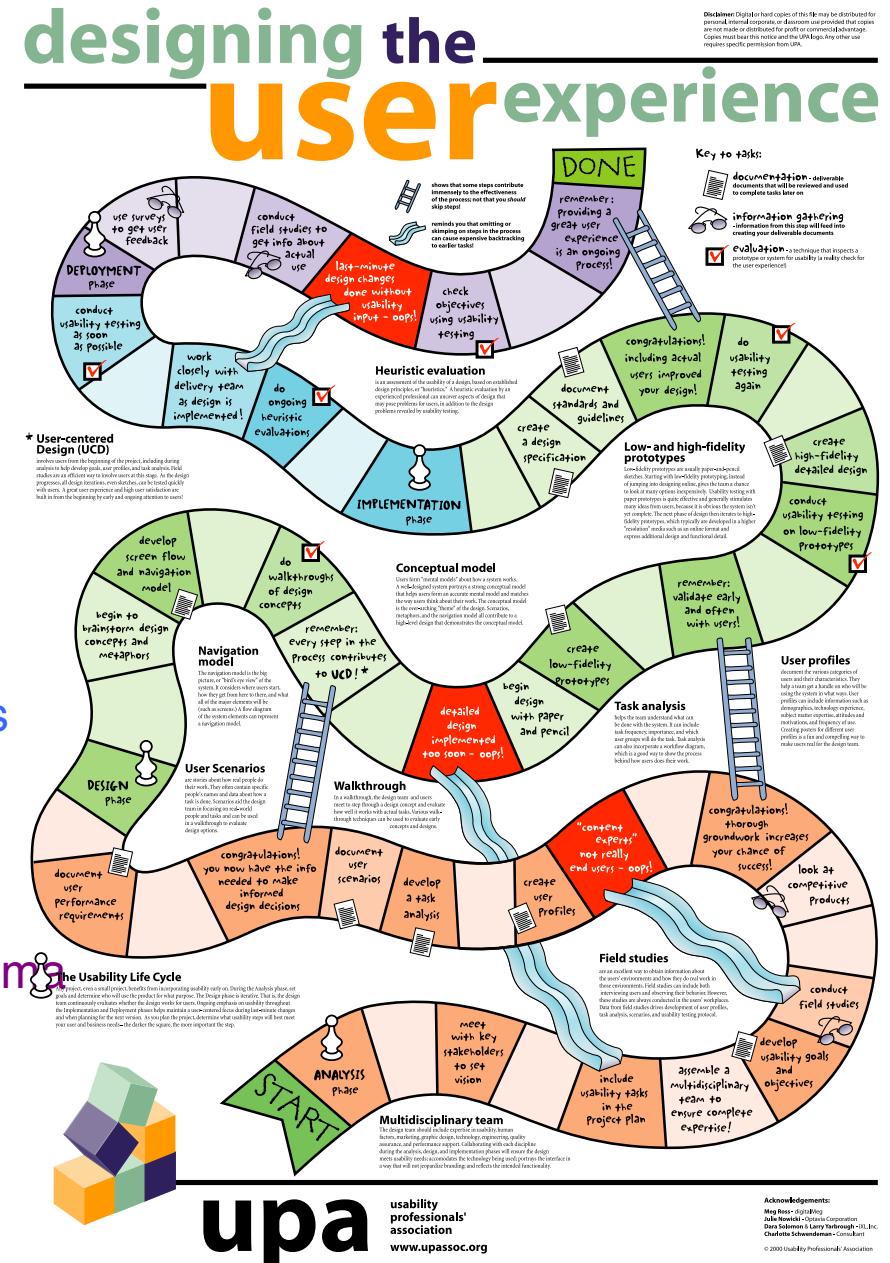
O que falta!

# Interface com o utilizador?



# Concepção centrada no utilizador

- Análise de requisitos
    - análise de tarefas
  - Concepção
    - prototipagem
    - padrões de desenho
    - técnicas de avaliação
  - Desenvolvimento
    - arquitecturas software
    - tecnologias de programação de interfaces
    - técnicas de avaliação
  - Deployment
    - obter informação sobre utilização do sistema
    - avaliar resultado face a objectivos





# Trabalho

## Enunciado do Trabalho Prático

Desenvolvimento de Sistemas Software / Bases de Dados

Licenciatura em Engenharia Informática

2011/2012

## Contexto

O Medicamento de Uso Humano está sujeito a padrões de qualidade, segurança e eficácia, alicerçados na atuação conjunta dos responsáveis pela sua colocação no mercado, das autoridades competentes nacionais e comunitárias. Os diferentes intervenientes (Fabricantes, Distribuidores, Prescritores, Farmácias, outros locais de venda e Utilizadores) estão sujeitos a um conjunto de obrigações e procedimentos competindo ao INFARMED acompanhar e garantir a sua aplicação [1].

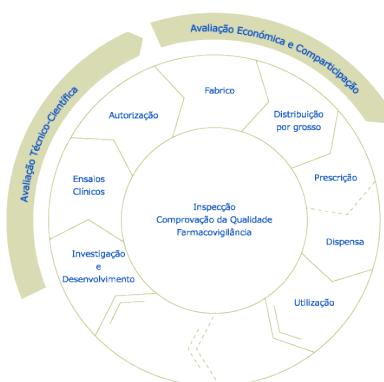


Figura 1 - Círculo Interativo do Medicamento de Uso Humano [1]

A Figura 1 descreve o circuito interativo dos medicamentos para uso humano, desde a investigação e desenvolvimento de novas substâncias ativas e novos medicamentos, até à utilização desses medicamentos. O presente enunciado foca-se nas duas fases imediatamente anteriores à utilização: a prescrição e dispensa. A prescrição, ou receita, consiste na indicação, por parte do médico, dos medicamentos que um paciente deve tomar. A dispensa consiste no fornecimento, pela farmácia, dos medicamentos prescritos pelo médico.

Nos últimos anos tem vindo a ser levada a cabo uma política de informatização dos processos de prescrição e dispensa de medicamentos, tendo em vista a sua desmaterialização. Ou seja, a realização dessas atividades sem recurso a documentos em suporte físico, em papel, recorrendo apenas a documentos electrónicos. Nesse contexto, a Portaria número 198/2011 de 18 de Maio (em anexo) [2] regulamenta a desmaterialização da prescrição médica e da dispensa pelas farmácias dos medicamentos prescritos, estabelecendo os requisitos a que deve obedecer, quer a prescrição electrónica (i.e., a prescrição de medicamentos efectuada com recurso às tecnologias de informação e de comunicação), quer o seu controlo através de meios informáticos. Em articular, são definidos os modelos de receita médica e as regras de prescrição.

Ao mesmo tempo, tem vindo a ser encorajada a prescrição de medicamentos genéricos como forma de gerar economias de escala. São considerados medicamentos genéricos aqueles que sejam essencialmente similares de um medicamento de referência, em que tenha caducado os direitos de propriedade industrial relativos às respectivas substâncias ativas ou processo de fabrico e que não invoquem a seu favor indicações terapêuticas diferentes relativamente ao medicamento de referência já autorizado [3].

## Objectivos

Pretende-se com este trabalho desenvolver duas aplicações informáticas que suportem a desmaterialização das atividades de prescrição e dispensa de



medicamentos, segundo as regras legalmente definidas. Uma das aplicações deverá permitir ao médico prescrever medicamentos, idealmente fornecendo suporte à seleção de medicamentos com base nas patologias a tratar. A outra aplicação deverá suportar a dispensa dos medicamentos receitados. Em ambos os casos deverá ser suportada e encorajada a prescrição e dispensa de medicamentos genéricos.

Tal como anteriormente referido, os requisitos para a prescrição e dispensa electrónicos estão descritos na Portaria número 198/2011 de 18 de Maio (em anexo). Como complemento à portaria, sugere-se a consulta do portal do Infarmed [4]. Em especial a secção de perguntas frequentes sobre medicamentos de uso humano [5].

As soluções metodológicas e tecnológicas a adoptar no desenvolvimento das aplicações terão que obedecer a um conjunto de requisitos, em função da(s) Unidade(s) Curricular(es) (UC) a que forem submetidos para avaliação. Listam-se de seguida esses requisitos para cada uma das UC:

- **Base de Dados:**

- a primeira aplicação deverá ser desenvolvida para funcionar em ambiente WINDOWS usando Visual Basic, Visual C# ou Java.
- a segunda aplicação deverá ser desenvolvida utilizando tecnologias ASP, ASPX, Perl ou PHP
- A base de dados deverá ser criada usando o motor ORACLE.
- A informação a representar, armazenar e processar deverá estar, pelo menos, na terceira forma normal.
- As chaves primárias, candidatas e estrangeiras deverão estar claramente identificadas.
- As tabelas deverão estar indexadas pelos atributos usados em seleções, ordenações e agrupamentos.
- Os possíveis valores nulos deverão estar claramente identificados.
- As restrições de integridade estrutural e referencial deverão ser objecto de estudo e de declaração.

- As aplicações deverão presumivelmente funcionar de forma independente do motor de base de dados escolhidos.
- Para implementar o que é pedido no ponto anterior, o utilizador da aplicação deverá identificar-se, autenticar-se e selecionar a base de dados à qual se pretende ligar.
- É obrigatória a utilização de SEQUENCES, VIEWS, FUNCTIONS, PROCEDURES e TRIGGERS.

- **Desenvolvimento de Sistemas Software:**

- Uma das aplicações deverá ser desenvolvida utilizando tecnologias orientadas a objetos (idealmente Java).
- A análise de requisitos deverá produzir um Modelo de Domínio e um modelo de *Use Case* e deverá considerar as duas aplicações.
- A aplicação desenvolvida em tecnologias orientadas a objetos será a única avaliada e deverá ser alvo de um processo de modelação suportado por UML, de acordo com o processo descrito nas aulas.

## Avaliação

O trabalho será alvo de processos de acompanhamento e avaliação independentes por parte de cada uma das UC envolvidas. Assim,

- **Desenvolvimento de Sistemas Software:**

- O trabalho será realizado em duas fases. Em cada fase deverá ser entregue um relatório detalhando os objectivos da fase, a abordagem seguida para os atingir, os resultados obtidos (os modelos) e uma análise critica dos mesmos.
  - Fase 1: relatório de análise de requisitos, a entregar até 4 de Novembro
  - Fase 2: relatório final de modelação e desenvolvimento e software produzido, a entregar até 30 de Dezembro



- Cada fase terá uma avaliação independente. Os pesos de cada fase são indicados abaixo.
  - Fase 1: 35% da classificação final
  - Fase 2: 65% da classificação final
- A apresentação e discussão final do trabalho será realizada na semana de 02 de Janeiro 2012.
- **Bases de Dados:**
  - Os resultados do trabalho prático serão apresentados em sessão pública a marcar no final do semestre, sendo obrigatória a entrega de relatório em papel uma semana antes da apresentação do trabalho, assim como uma versão do trabalho e do relatório em formato electrónico.
  - Serão elaboradas sessões intermédias para apresentação parcial do trabalho. Essas sessões serão marcadas na semana anterior e realizadas durante as aulas práticas. Os grupos com alunos que pertencerem a turnos diferentes farão as apresentações em ambos os turnos.

### Grupos de Trabalho

A definição dos grupos de trabalho será efectuada com recurso ao sistema de *elearning*, em moldes a definir nas aulas.

### Referências

- [1] Infarmed. *Medicamentos Uso Humano*.  
[http://www.infarmed.pt/portal/page/portal/INFARMED/MEDICAMENTOS\\_USO\\_HUMANO](http://www.infarmed.pt/portal/page/portal/INFARMED/MEDICAMENTOS_USO_HUMANO) (acedido em 12/10/2011).
- [2] Portaria nº 198/2011 de 18 de Maio
- [3] Decreto Lei nº 242/2000 de 26 de Setembro, Artigo 19º.

[4] Infarmed. Página inicial.

<http://www.infarmed.pt/portal/page/portal/INFARMED> (acedido em 12/10/2011).

[5] Infarmed. Perguntas Gerais de Medicamentos de Uso Humano.

[http://www.infarmed.pt/portal/page/portal/INFARMED/PERGUNTAS\\_FREQUENTES/MEDICAMENTOS\\_USO\\_HUMANO/](http://www.infarmed.pt/portal/page/portal/INFARMED/PERGUNTAS_FREQUENTES/MEDICAMENTOS_USO_HUMANO/) (acedido em 12/10/2011).

José Creissac Campos

José Machado

António Abelha

António Nestor Ribeiro



# 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
- UP vs HCSD
- Apresentação do trabalho