



# Desenvolvimento de Sistemas Software

## Aula Teórica 8: Concepção de Interfaces / Prototipagem



## E depois dos Use Cases?



Use Cases permitem definir o que o sistema deverá fazer.

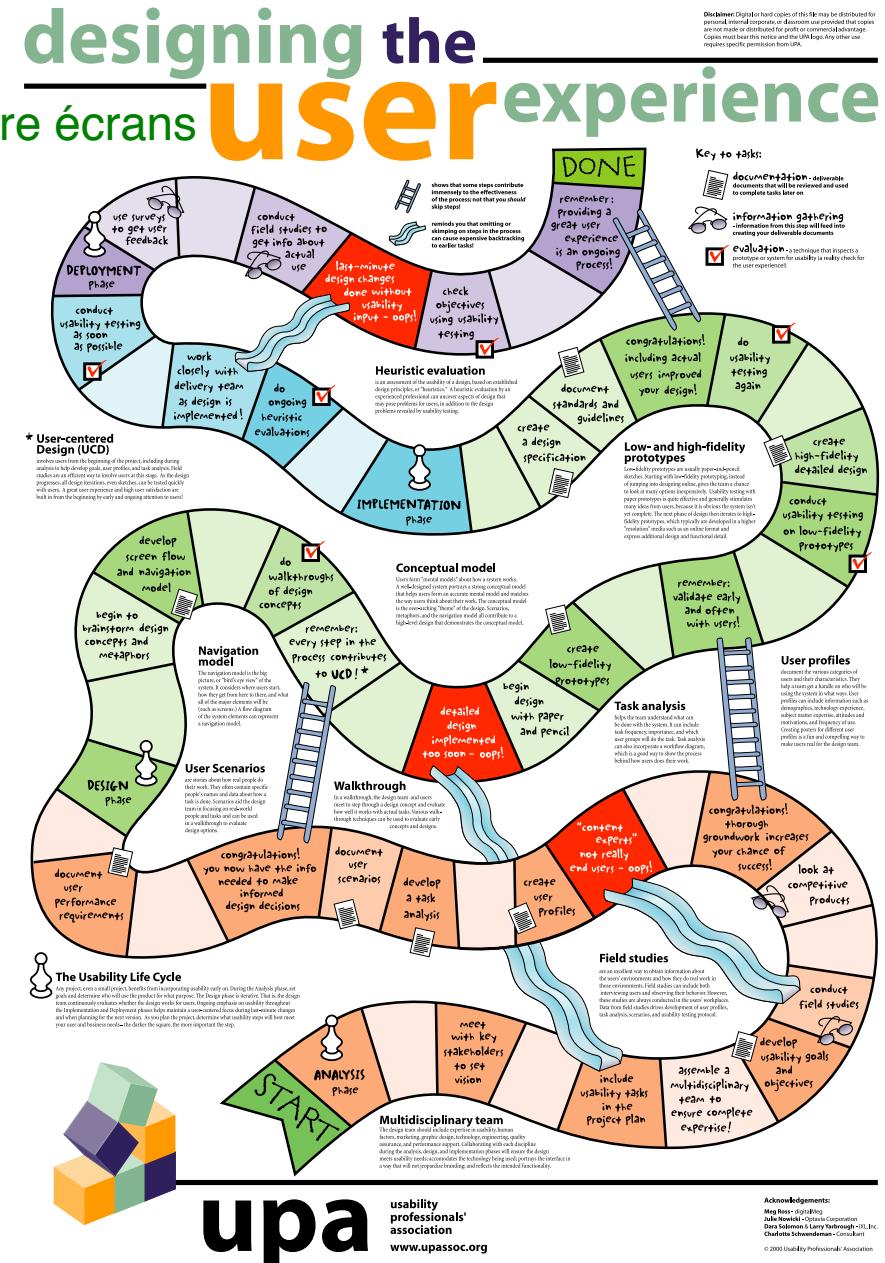
Mas... como, em concreto, o deverá fazer?! (do ponto de vista de quem o vai user)

Falta a interface!



# Abordagem HCD

- **Concepção**
  1. desenvolva o modelo de navegação entre écrans
  2. avalie o modelo
  3. crie protótipo de baixa fidelidade (papel e lápis)
  4. faça testes de usabilidade (com utilizadores)
  5. crie protótipo de alta fidelidade
  6. faça novamente testes de usabilidade
  7. documente normas e *guidelines*
  8. crie uma especificação detalhada





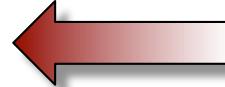
# Prototipagem da UI

- Análise da solução antes do desenvolvimento
  - Esboços do aspecto gráfico (e comportamento)
- Objectivos
  - Testar um conceito
  - Explorar alternativas
  - Explorar tecnologias
  - Comunicar
- Quando
  - Nas fases iniciais (antes de codificar)
  - Face a decisões complexas
  - Soluções inovadoras
- Vantagens
  - Economia de tempos e recursos
  - Validação precoce



# Tipos de protótipos da UI

- Baixa fidelidade



- Rápidos e simples (potencialmente sem recurso a computador)
- Apenas uma ideia/análise aproximada
- Desenhos e capturas de écran
- Usar e deitar fora

- Alta fidelidade

- Permitem animação da interface
- Permitem capturar interacção do utilizador
- Mais próximos da tecnologia de implementação
- HTML, Flash, IDEs...



# Protótipos de baixa fidelidade

- Papel e lápis uma das abordagens que melhor funciona
  1. Abordagem de baixo custo (bons sistemas de prototipagem são normalmente pagos).
  2. Simples de construir e manipular mesmo para os utilizadores (curva de aprendizagem muito baixa)
  3. Permite maior liberdade de expressão (softwares de prototipagem limitam o que pode ser feito)
  4. Aspecto “imperfeito” encoraja contribuições e alterações.
  5. Permite maior controlo sobre nível de abstracção.

Student Information

Student Number: 789-567-234

First Name: Scott

Middle: William

Surname: Ambler

Salutation: Mr. ▾

Date first entry: June 14 2003

Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to C++	Fall 2003	A+	Passed
CSC 200 Intro to AM	Fall 2003	A	Passed
CSC 203 Advanced AM	Spring 2004	-	Enrolled
-	-	-	-

Add ... Drop ... Transcript Close

Student Information

Student Number: 789-567-234

FirstName: Scott

Middle: William

Surname: Ambler

Salutation: Mr. ▾

Date first entry: June 14 2003

Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to C++	Fall 2003	A+	Passed
CSC 200 Intro to AM	Fall 2003	A	Passed
CSC 203 Advanced AM	Spring 2004	-	Enrolled
-	-	-	-

Add ... Drop ... Transcript Close



# Protótipos de alta fidelidade

- Mais próximos do aspecto final da aplicação.
- Idealmente desenvolvidos na tecnologia de desenvolvimento.
- Custo elevado
  - Excel drag and drop: 3 meses para o protótipo vs 1 semana no Excel!
  - Microsoft Bob: perto de 1 ano (sistema foi um fracasso!)
- Preferir abordagem evolucionária, minimizando custos de prototipagem e riscos de desenvolvimento.
- Perigo de “comprometimento prematuro”.
  - Cuidado com divulgação

The screenshot shows a Microsoft Internet Explorer window titled "Edit Student Information - Microsoft Internet Explorer". The window contains a form with fields for Student number, First name, Middle name, Surname, Salutation, and a date field for First enrolled. Below the form is a table titled "Schedule" with columns for Seminar, Term, Mark, and Status. The table lists four seminars: CSC 100 Intro to C#, CSC 200 Intro to Agile Modeling, CSC 203 Advanced Agile Modeling, and CSC 220 Intro to Agile Databases. The status for all seminars is "Enrolled". At the bottom of the window are "Print Transcript" and "Help" buttons.

Seminar	Term	Mark	Status
CSC 100 Intro to C#	Fall 2003	A+	Passed
CSC 200 Intro to Agile Modeling	Fall 2003	B-	Passed
CSC 203 Advanced Agile Modeling	Spring 2004	-	Enrolled
CSC 220 Intro to Agile Databases	Spring 2004	-	Enrolled

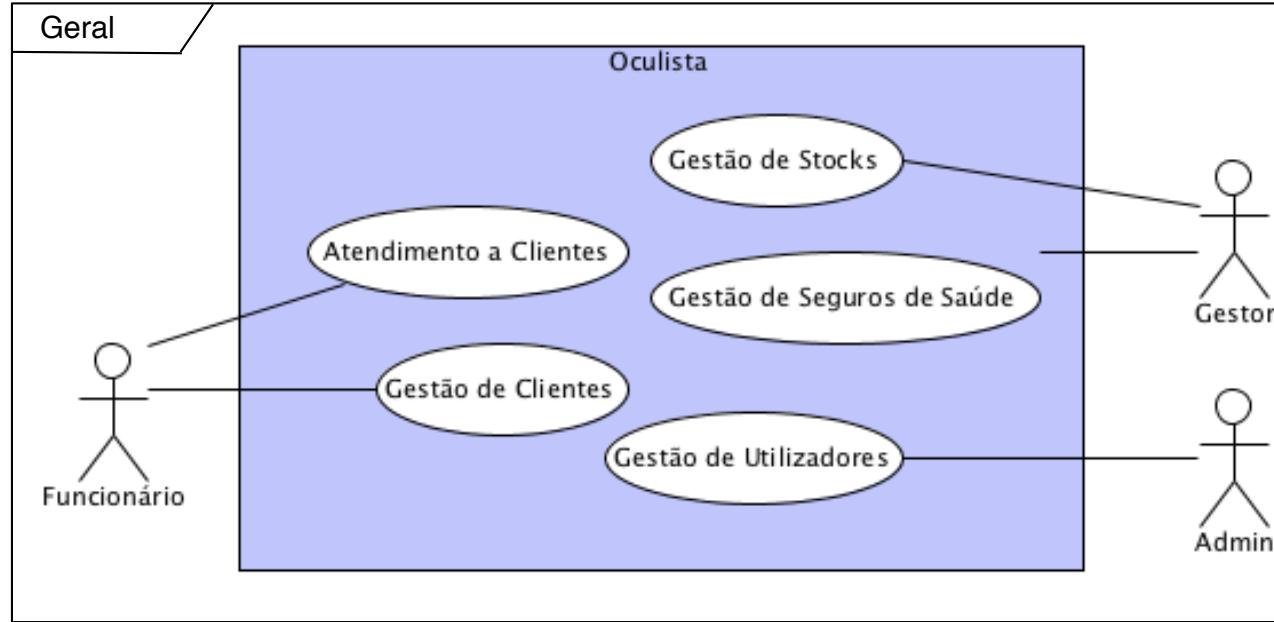


# Abordagem

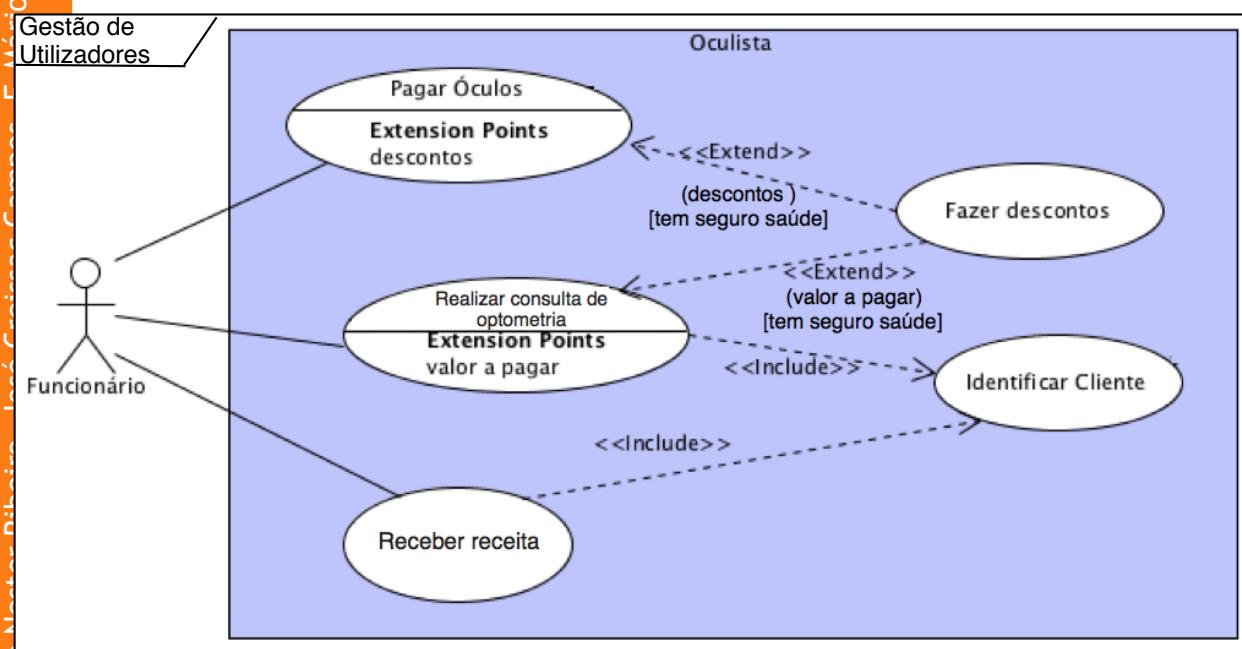
- Vamos utilizar prototipagem de baixa fidelidade e desenvolver os protótipos a partir do modelo de Use Case:
  1. Identificar conjuntos de Use Case sobre os mesmos dados
  2. Decidir estilo de interface / estrutura de menus
  3. Para cada Use Case (ou conjunto de Use Case)
    - a) Identificar informação que o actor deve fornecer
    - b) Identificar informação que o actor deve conhecer
    - c) Identificar acções possíveis (do actor)
    - d) Desenhar a janela para o(s) Use Case(s)
  4. Definir navegação entre janelas



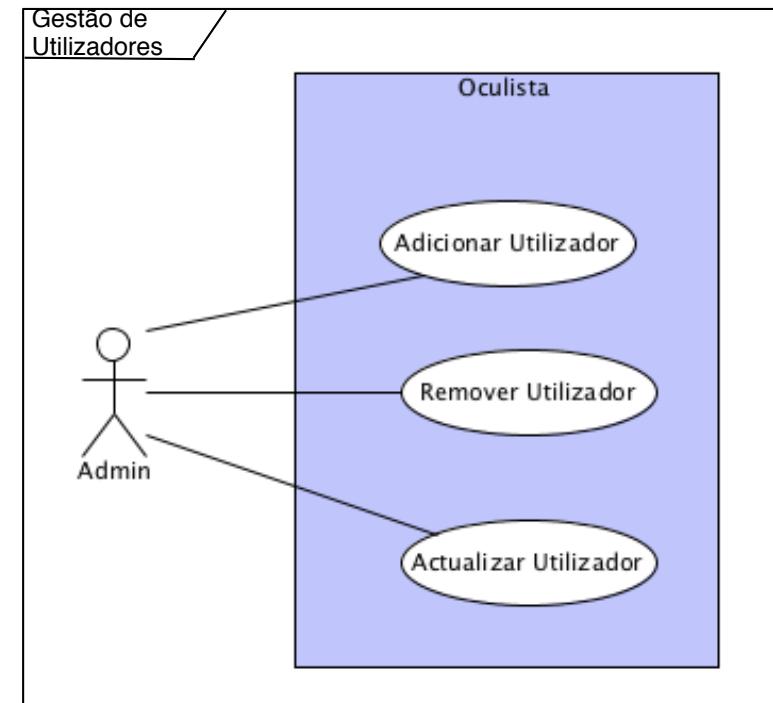
Geral



Um exemplo!



Gestão de Utilizadores



### 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

### 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: 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: 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



# Qualidade da Interface - Usabilidade

Os Use Case!!

Não Actores, mas Utilizadores concretos!

??!

- “Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

(ISO DIS 9241-11).

- effectiveness - Accuracy and completeness with which users achieve specified goals;
  - Percentagem de cobertura dos Use Cases?
  - Modo como são suportados?
- efficiency - Resources expended in relation to the accuracy and completeness with which users achieve goals (time, learning effort, etc.)
  - O utilizador consegue utilizar?
- satisfaction - Freedom from discomfort, and positive attitudes towards the use of the product
  - O utilizador gosta de utilizar?

} Respeitar Use Cases

} Pensar nos utilizadores



# Conhecer os Utilizadores

- Quem são?
  - influencia objectivos de usabilidade e desenho.
- Provavelmente um grupo heterogéneo...
  - Não vale a pena definir o utilizador médio
  - Não vale a pena utilizar o mínimo múltiplo comum
  - Nunca nós próprios como modelo!
- Possíveis dimensões de análise:
  - Classes de utilizadores
  - Níveis de pericia
- Criar categorias facilita o trabalho



# Classes de Utilizadores

- Sub-conjunto homogéneo de utilizadores
  - Tipo de utilização do sistema
  - Características pessoais
- Quatro classes típicas
  - Utilizadores directos
  - Utilizadores indirectos
  - Utilizadores remotos
  - Utilizadores de suporte
- ou ainda,
  - obrigados a utilizar o sistema, ou podem escolher?
  - intermitentes ou continuados?

Diferente de actores!



# Níveis de Perícia

- Influencia o tipo de sistema, suporte e treino.
- Tipicamente
  - Inexperientes
  - Intermédios
  - Peritos
- Mas...
  - Mais que uma dimensão (tecnologia vs. domínio)
  - Quão perito é um perito? (especialização)



# Utilizadores Inexperientes

- Pouca ou nenhuma experiência com computadores
- Podem recuar utilizar o sistema
  - sistemas de ajuda simples e eficazes
- Necessitam de feedback frequente
- Preferem ser 'guiados' na interacção
  - 'wizards', noção de tarefa.
- Têm de se sentir seguros que não vão 'estragar' o sistema
  - facilidades de 'undo'.

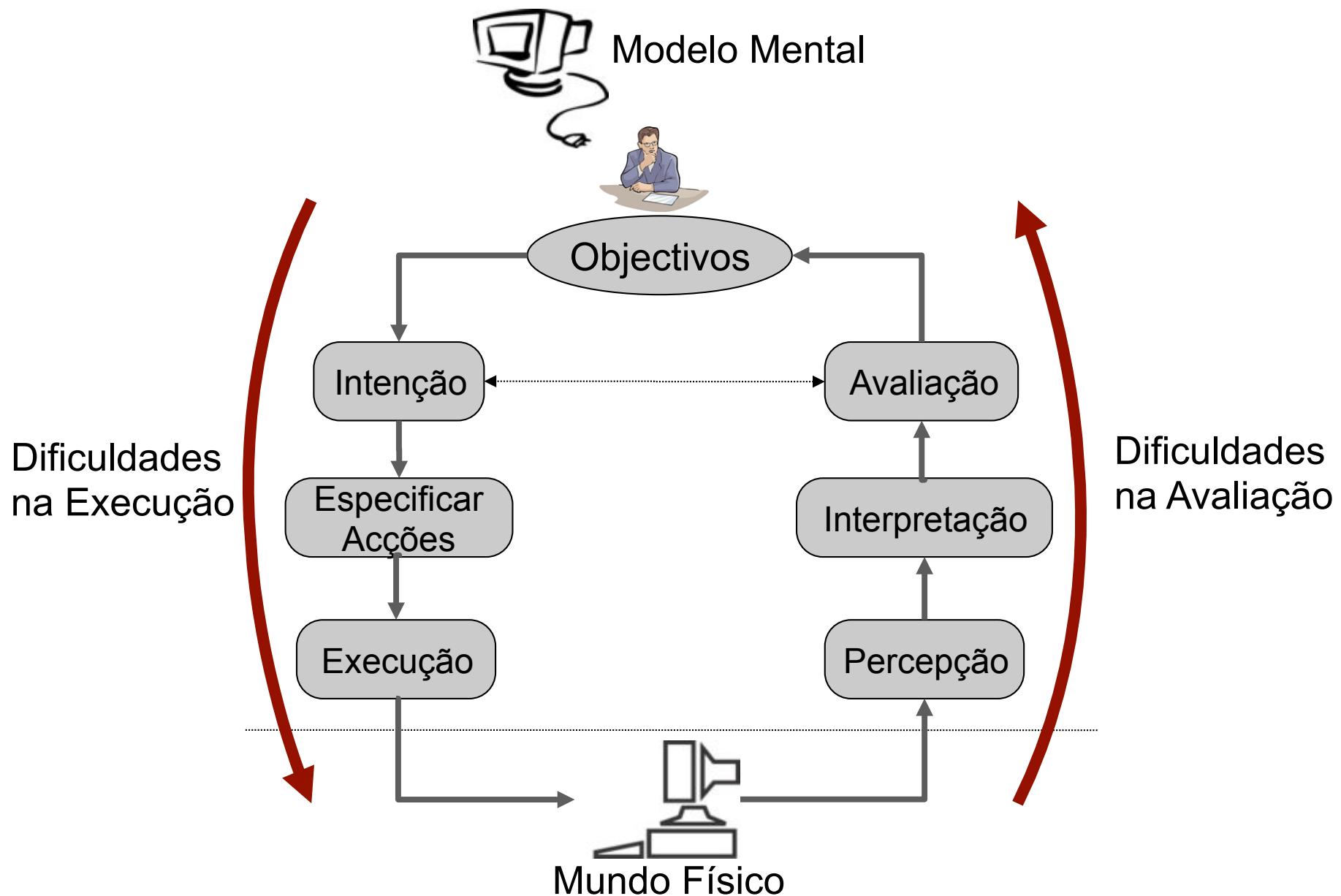


200

# Utilizadores Peritos

- Sentem-se mais confiantes na sua interacção com o sistema
- São capazes de procurar 'feedback' quando necessitam
  - Menor necessidade de feedback e suporte.
- Preferem ter a possibilidade de utilizar 'short-cuts' e comandos abreviados.
- Preferem interfaces que possam configurar
- Gostam de sentir que detém o controlo da interacção.

# Simplificando... - Modelo de Interacção de Norman





# Dificuldades na Execução

- Estamos a facilitar a execução dos Use Case?
- Exemplo:

**Saldos e Movimentos**

Escolha a Conta *	Conta Ordem <input type="button" value="▼"/>
Nº Movimentos	<input type="text"/> ou Indique Período <input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="a"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="dd/mm/aaaa"/>
Tipo Movimentos *	<input type="button" value="Todos"/> <input type="button" value="▼"/>
Canal Operações *	<input type="button" value="Todos"/> <input type="button" value="▼"/>

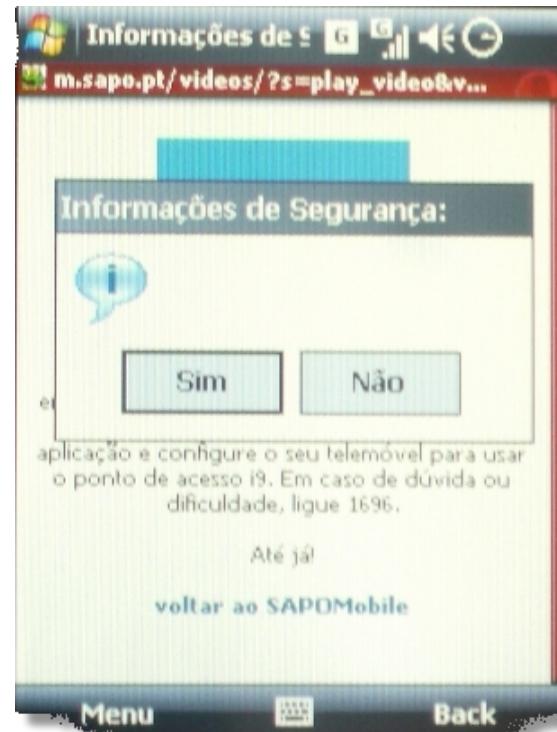
**Cancelar** **Consultar**

- Outro exemplo: criação dos turnos práticos!!
- Importante: tornar os use case mais comuns mais acessíveis.



# Dificuldades na Avaliação

- Estamos a fornecer toda a informação relevante?
- Exemplo:

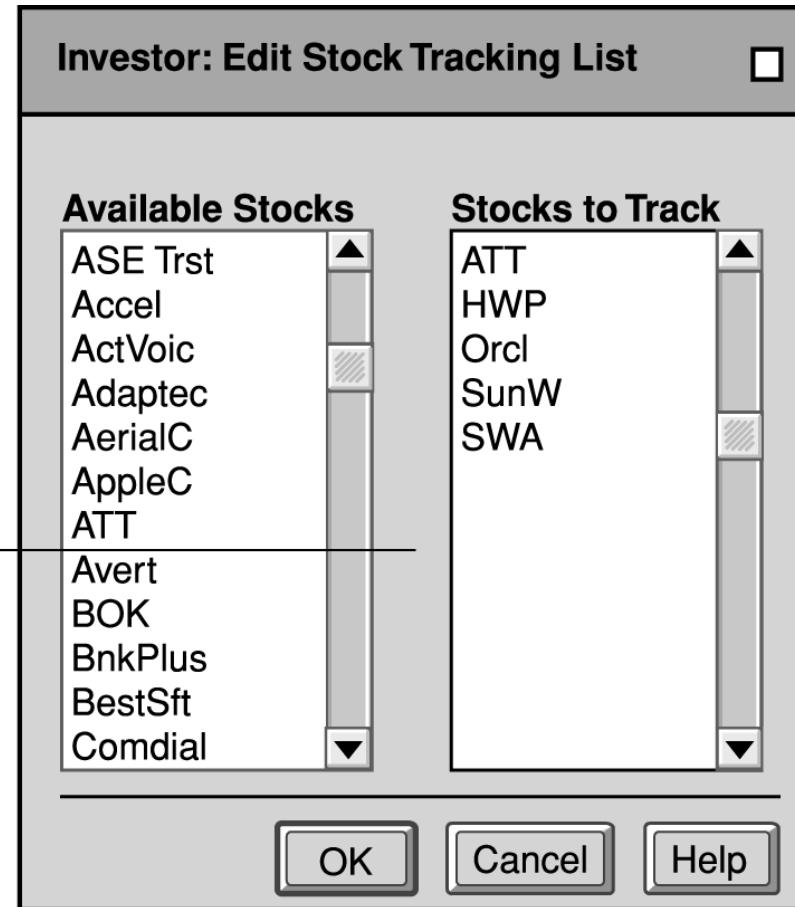


- Outro exemplo: inserção de sumários!!
- Importante: sempre que actor tiver que tomar decisões, fornecer a informação necessária.



## Outro exemplo...

No visible method of transferring items from one list to the other (drag-and-drop is the only method)





## Shneiderman's 8 Golden Rules

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer error prevention and simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load



# Concepção da Interface

## Sumário:

- Prototipagem de Interfaces
  - Protótipos de alta e baixa fidelidade
- Uma abordagem para a prototipagem baseada em Use Case
- Noções de IHC
  - Usabilidade
  - Modelo de Interacção de Norman
  - Shneiderman's 8 Golden Rules