



Desenvolvimento de Sistemas Software

Aula Teórica 8

Concepção de Interfaces / Prototipagem

Requisitos funcionais

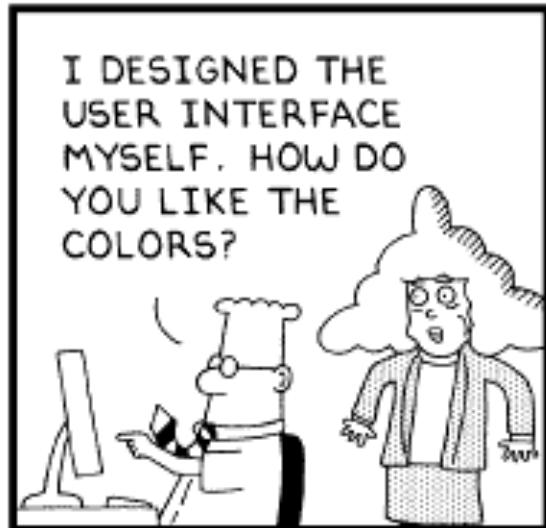


Requisitos funcionais permitem definir o que o sistema deverá fazer.

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

O que falta!?

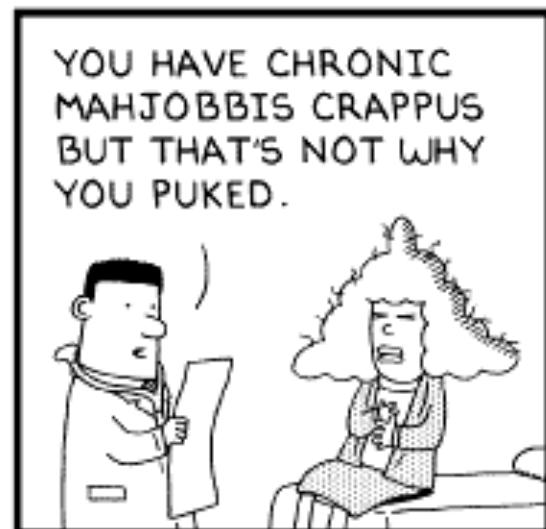
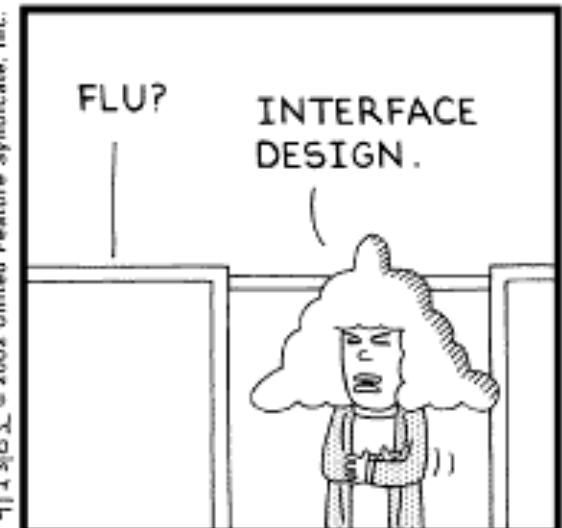
Interface com o utilizador?



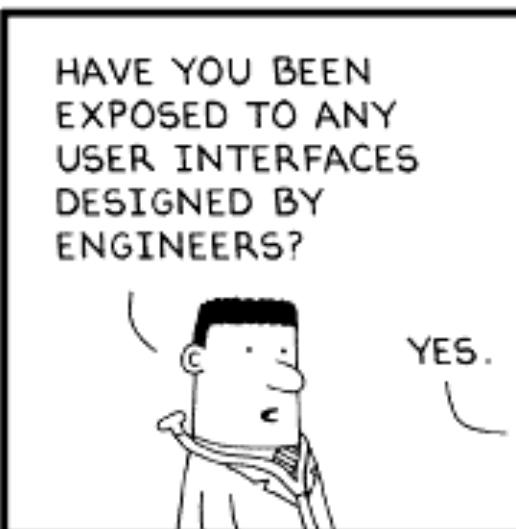
www.dilbert.com scottadams@aol.com



9/13/02 © 2002 United Feature Syndicate, Inc.



www.dilbert.com scottadams@aol.com



9/14/02 © 2002 United Feature Syndicate, Inc.

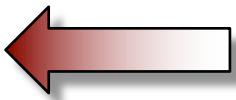


Prototipagem da UI

- 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 ecrã
- 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 lapis: 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.





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 permaturo”.
 - Cuidado com divulgação

The screenshot shows a Microsoft Internet Explorer window titled "Edit Student Information - Microsoft Internet Explorer". The window displays a form for editing student information and a table of course transcripts.

Student Information Form:

First name:	Scott
Middle name:	William
Surname:	Ambler
Salutation:	Mr.
First enrolled:	June 14 2003

Schedule: Add Seminar... Drop Seminar...

Transcript Table:

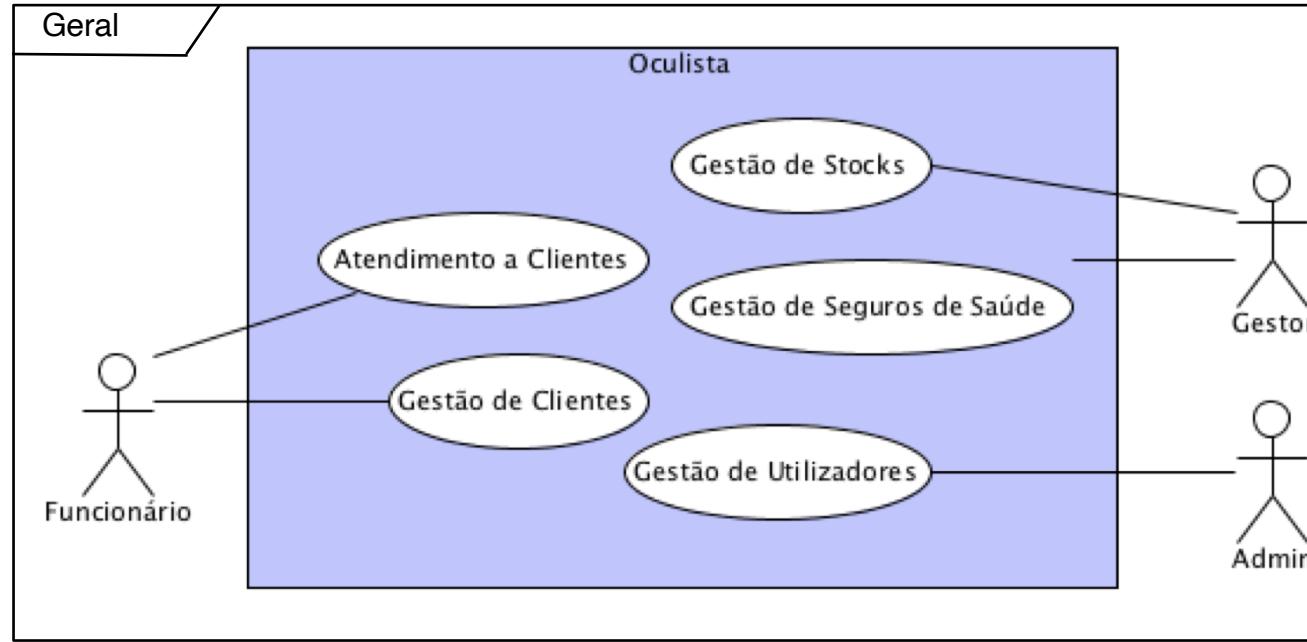
Seminar	Term	Mark	Status
CSC 100 Intro to C#	Fall 2003	A+	Passed
CSC 200 Intro to Agile Modeling	Fall 2003	B-	Passed
CSC 203 Advanced Agile Modeling	Spring 2004	-	Enrolled
CSC 220 Intro to Agile Databases	Spring 2004	-	Enrolled

Print Transcript... Help...

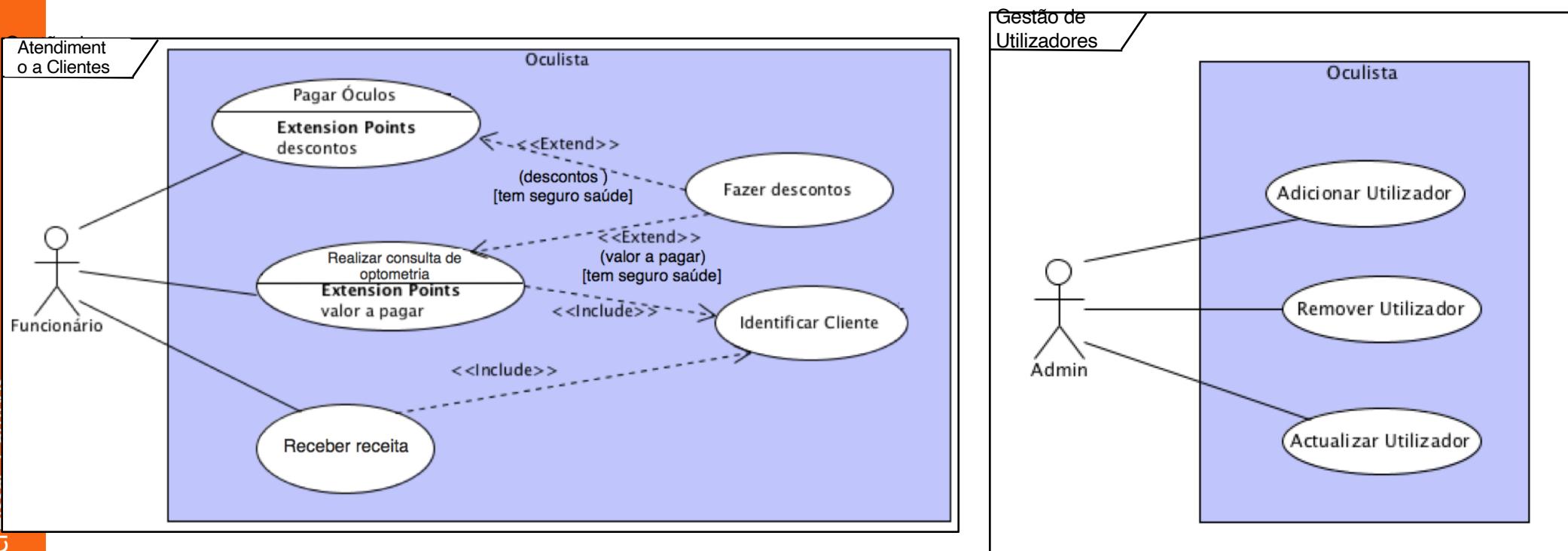
Abordagem

- 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!



Estilos de Interacção

- **Linha de comando**
 - Vantagens: Flexível; Agrada a utilizadores peritos; Configurável (scripts).
 - Desvantagens: Má facilidade de aprendizagem; taxas de erro altas; Desapropriado para não-peritos.
- ***Form filling***
 - Vantagens: Simplifica entrada de dados; Facilita aprendizagem; Guia o utilizador.
 - Desvantagens: Consome espaço de ecrã; Interfaces *rígidas*.



Estilos de Interacção

- **Menus**
 - Vantagens: Ideal para noviços ou utilizadores ocasionais; Pode agradar a peritos se existirem mecanismos de aceleração (“*shortcuts*”); Possibilita exploração; Guia o utilizador.
 - Desvantagens: Demasiados menus leva a *information overload* e a interfaces complexas; Pode ser lento para utilizadores frequentes; Necessita de espaço de ecran.
- **Manipulação directa**
 - Vantagens: Facilita aprendizagem. Erros podem ser evitados mais facilmente. Encoraja exploração. Alto nível de satisfação subjectiva.
 - Desvantagens: Mais complicada de programar. Menos apropriada a ecrans pequenos. Nem sempre uma representação espacial é a mais adequada.

Exemplo

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

Fluxo 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

Fluxo de Excepção [funcionário não pretende confirmar]

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

Use Case: Identificar cliente

Descrição: identificação de um cliente por nome

Pré-condição:

Pós-condição: Cliente pretendido fica seleccionado

Fluxo 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

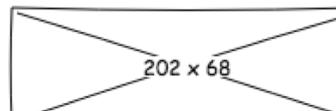
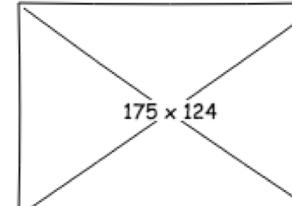
Fluxo Alternativo 1 [lista de clientes correspondentes tem tamanho 1]

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

Fluxo Alternativo 2 [funcionário quer criar novo cliente]

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

Registo de receitas

Nome Cliente:	indique o nome...	Data Nasc.:	DD/MM/AAAA	Procurar >
Cliente 1 Cliente 2				Novo Cliente
Armações... Armação 1 Armação 2 Armação 3		Lentes Esq./Dir. Lente Lente		Cancelar Registar

Qualidade da Interface - Usabilidade

Os Use Case!!

Não Actores, mas
Utilizadores concretos!

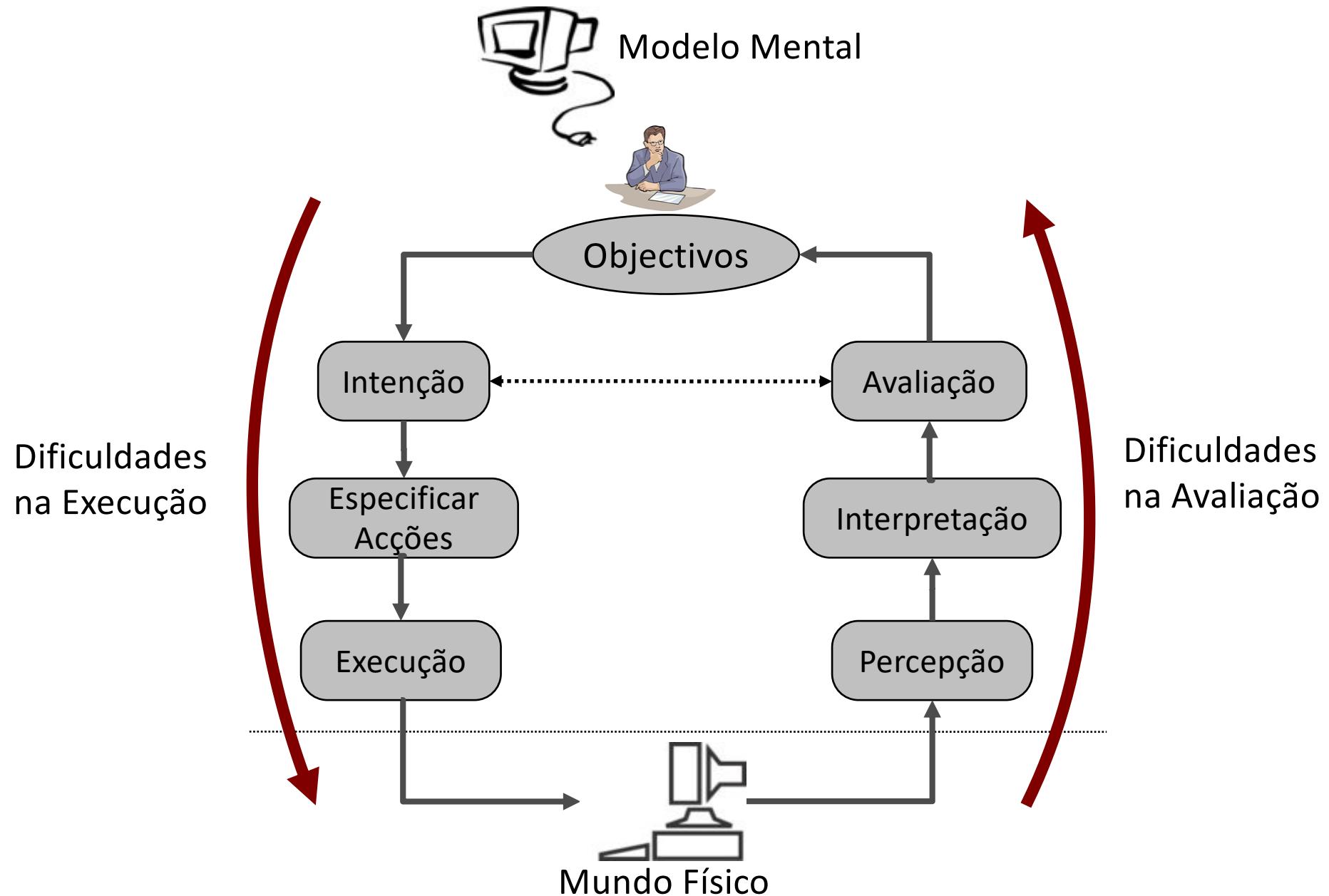
Onde/como vai ser utilizado?!

- “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 a
 - O utilizador consegue utilizar?
 - satisfaction - Freedom from discomfort, and positive attitudes tow
 - O utilizador gosta de/quer utilizar?

} Respeitar Use Cases

} Pensar nos utilizadores

Simplificando... - Modelo de Interacção de Norman



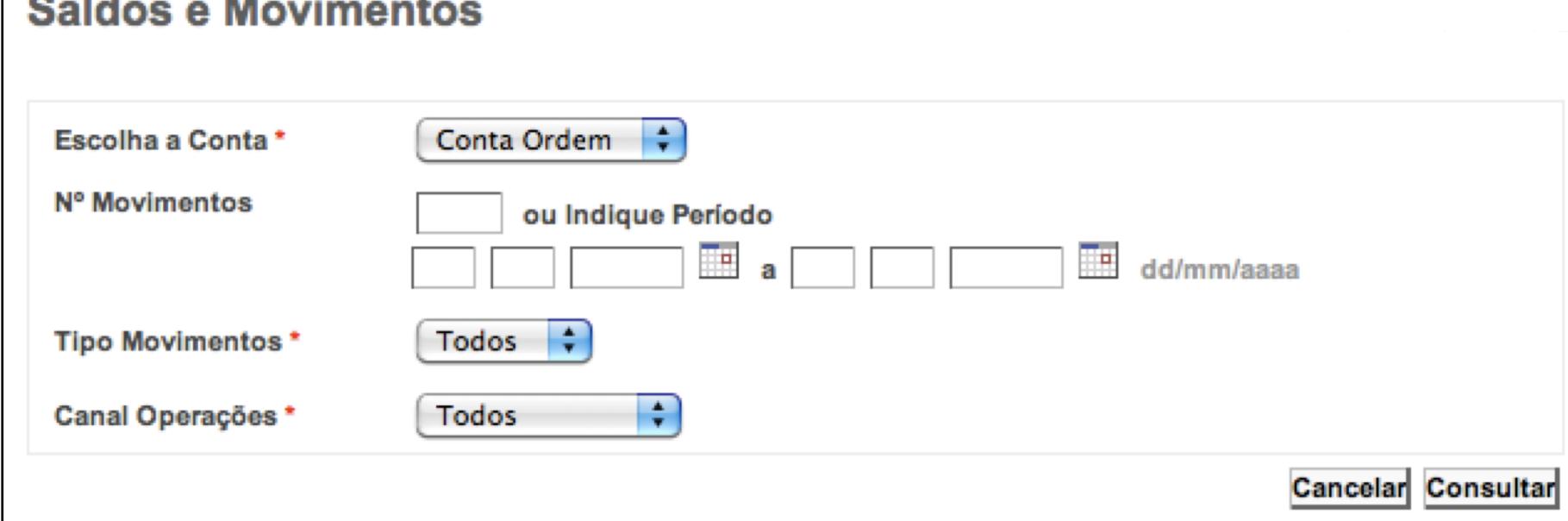
Dificuldades na Execução

- Estamos a facilitar a execução dos *Use Case*?
- Importante: tornar os use case mais comuns mais acessíveis.
- Exemplo:

Saldos e Movimentos

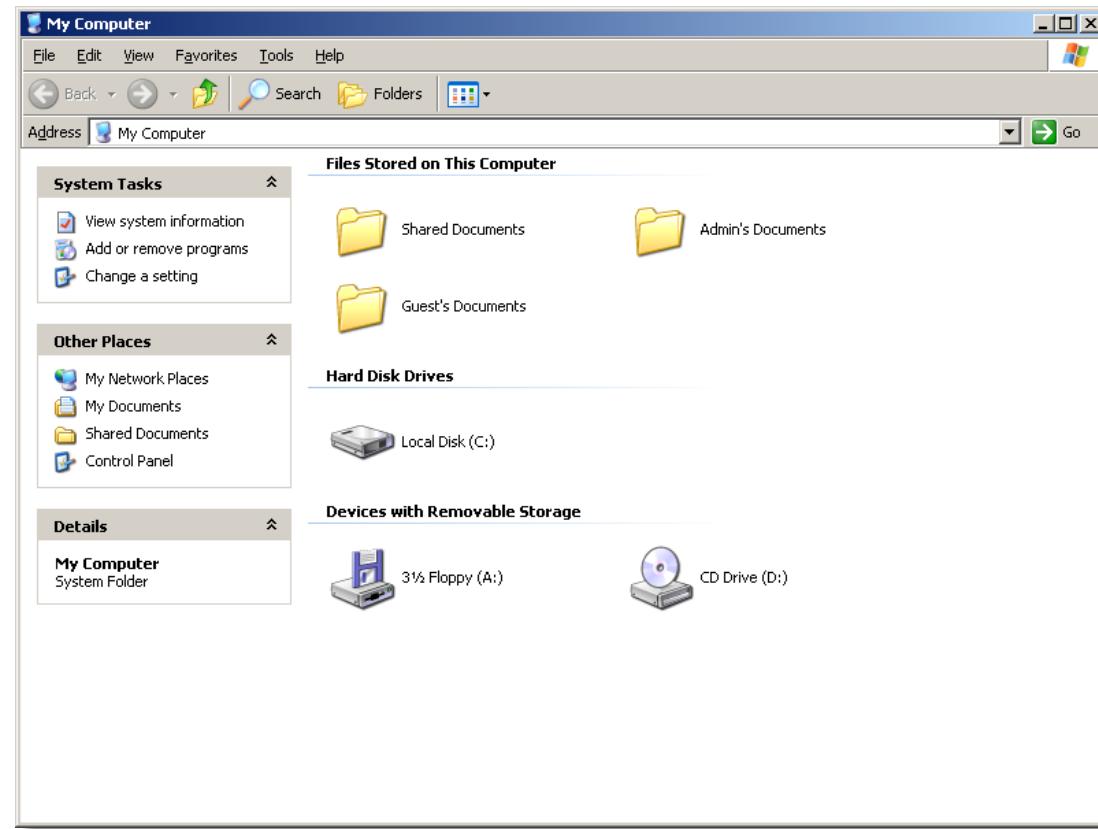
Escolha a Conta *	<input type="button" value="Conta Ordem"/>
Nº Movimentos	<input type="text"/> ou Indique Período
	<input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="a"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="dd/mm/aaaa"/>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

Cancelar **Consultar**



Dificuldades na Avaliação

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



- Importante: sempre que actor tiver que tomar decisões, fornecer a informação necessária.



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