

## Aula 30

Professor:

Geraldo Xexéo  
UFRJ

Conteúdo:

**Modelo de Interface**

# Como Projetar a Interface

- ➡ Compreender o ser humano
- ➡ Compreender as ações do ser humano ao usar o sistema
- ➡ Compreender as reações do ser humano a interface que o sistema apresenta

# Porque a qualidade da IU é importante

- ➡ Mais pessoas podem utilizar o computador.
- ➡ Ensinar computadores sobre as pessoas, ao invés de ensinar as pessoas sobre computadores
- ➡ Mais fácil de utilizar, mais formas de utilizar
- ➡ Economia de tempo, economia de dinheiro
- ➡ Mas...
- ➡ O que é Qualidade ?

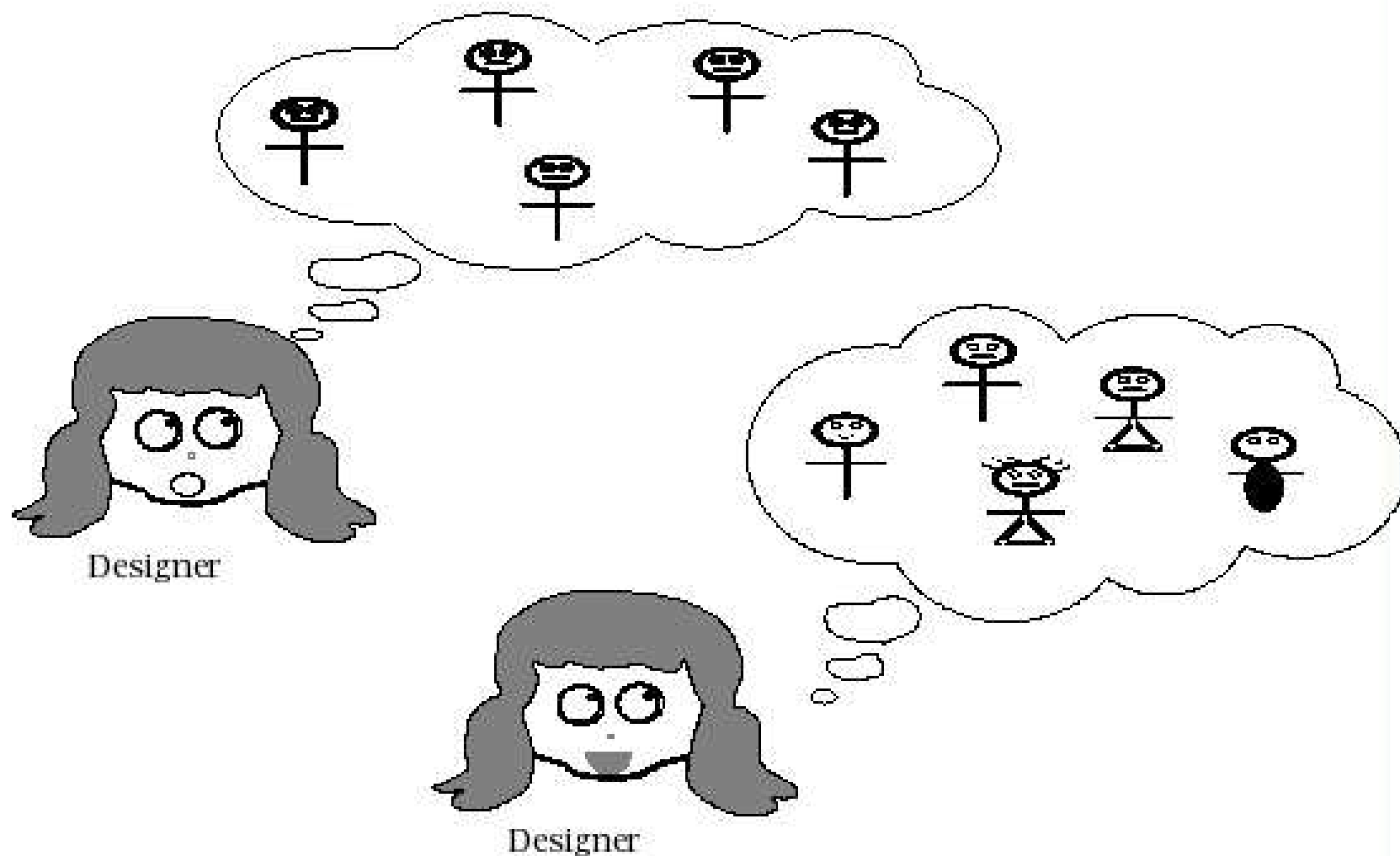
Professor:

Geraldo Xexéo  
UFRJ

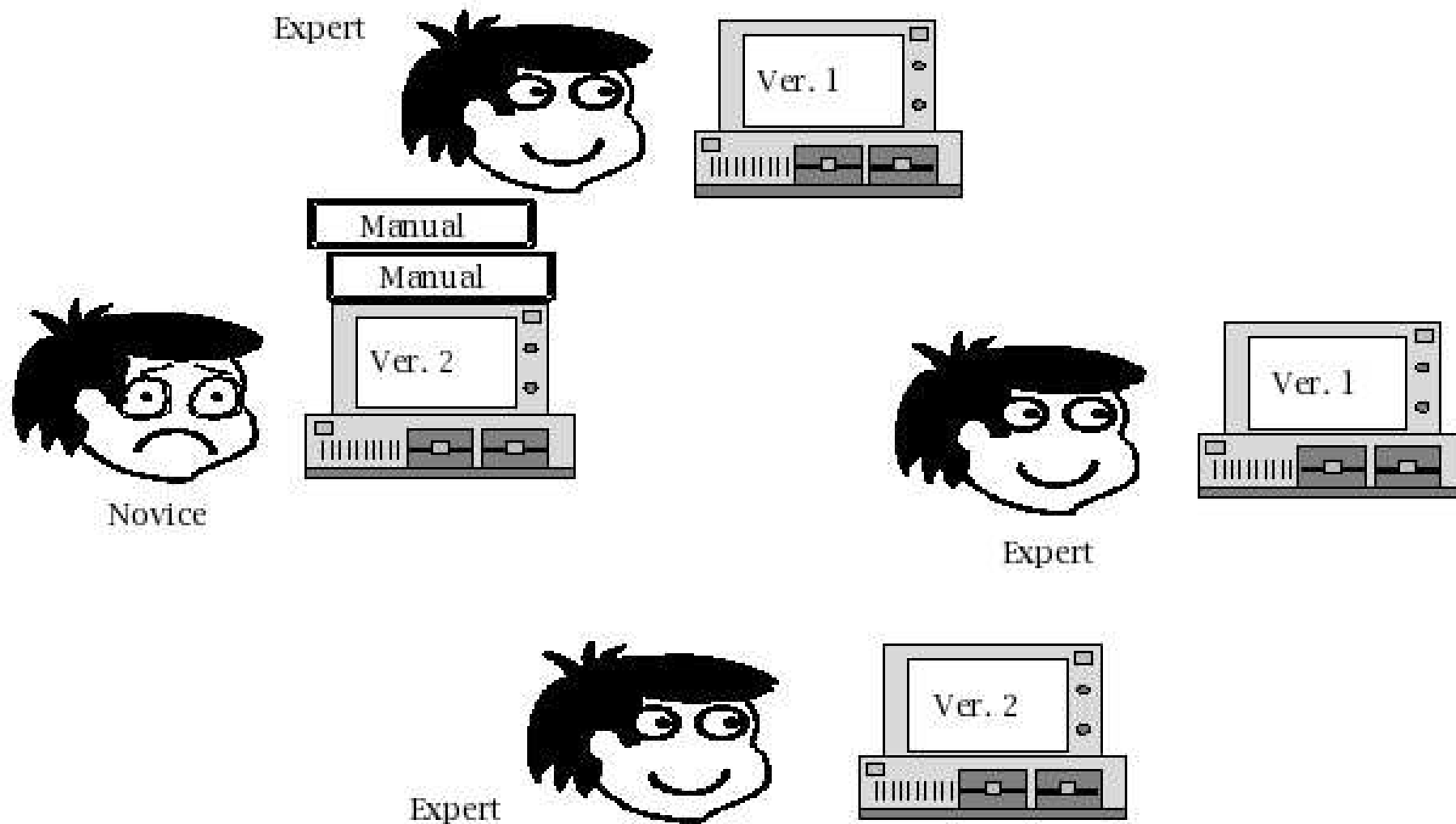
Conteúdo:

**Qualidade**

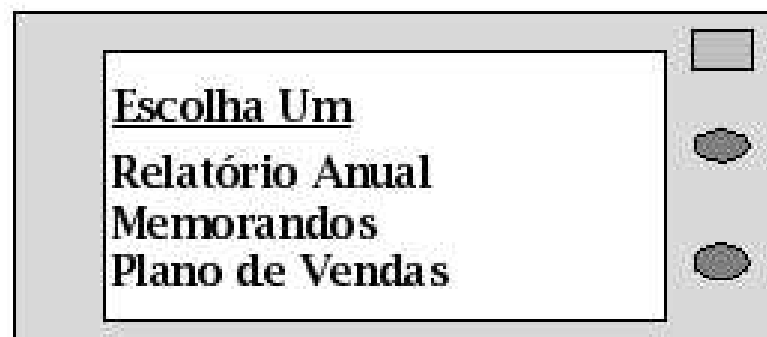
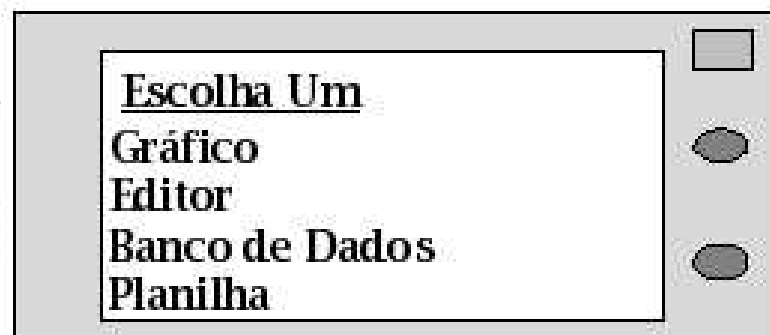
# Compatibilidade com o Usuário



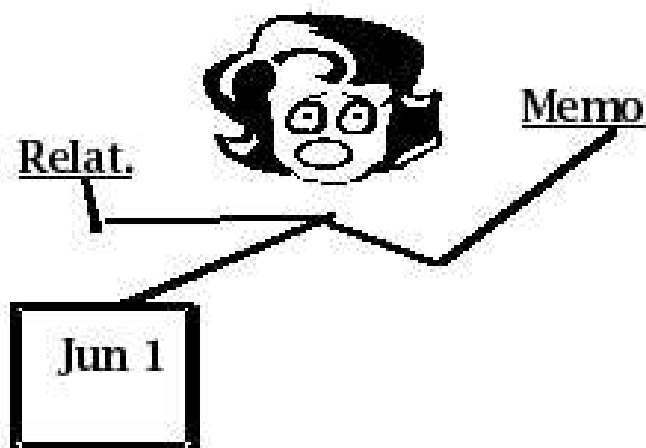
# Compatibilidade entre Produtos



# Compatibilidade com as Tarefas

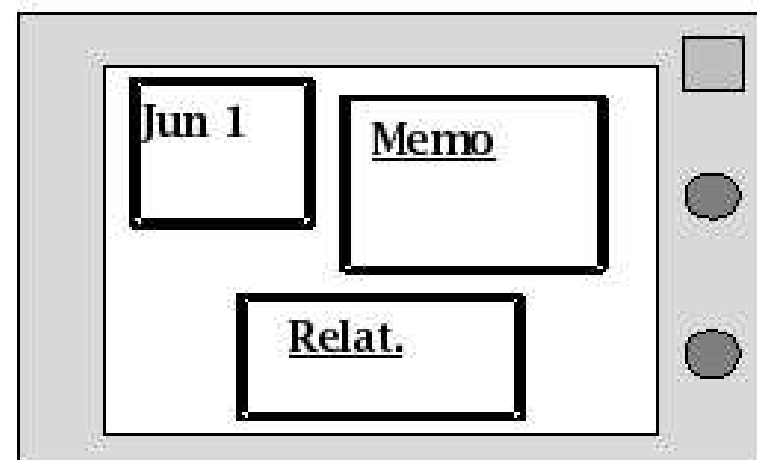


# Compatibilidade com o Fluxo de Trabalho



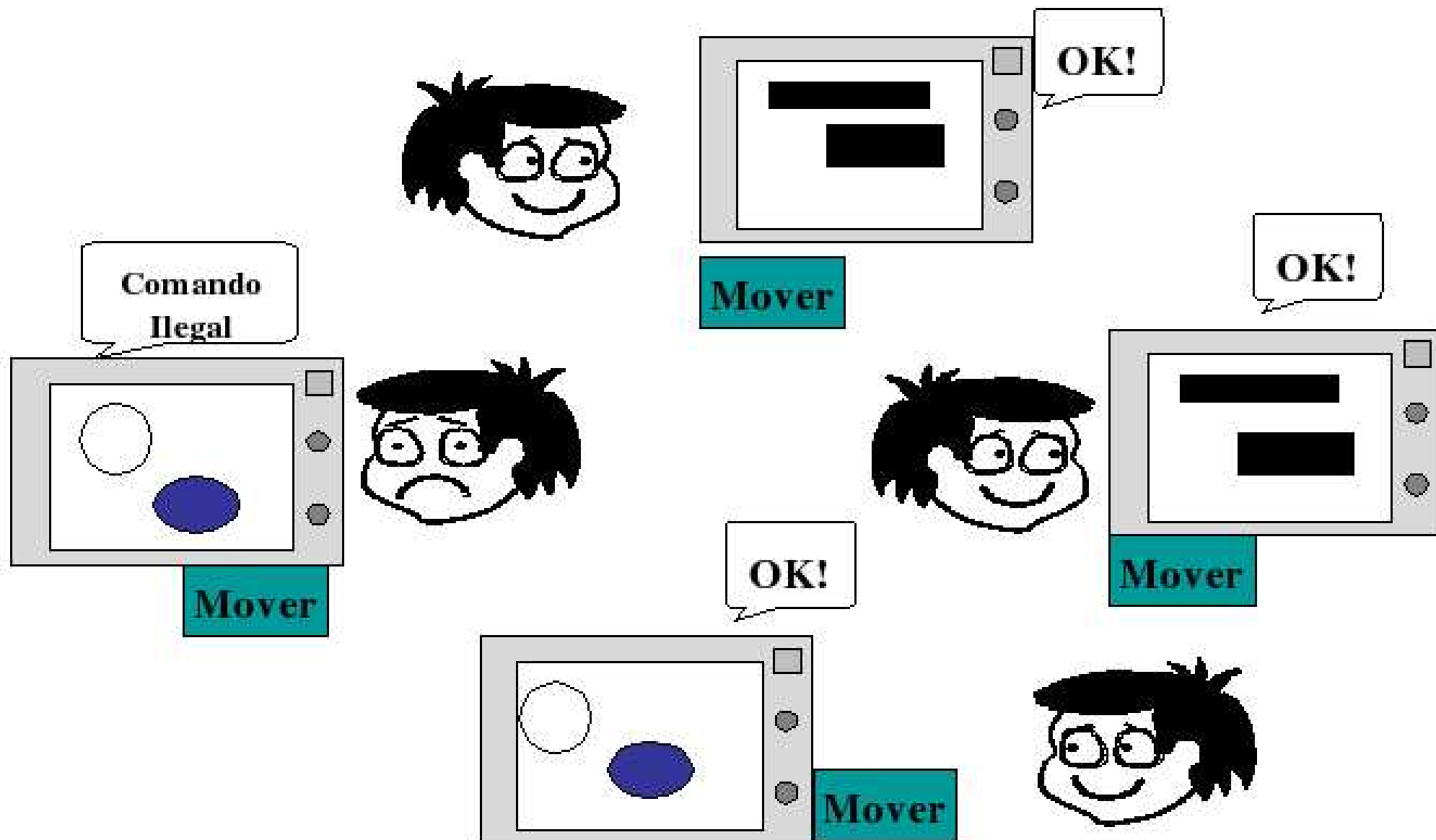
Escolha Um!!

Relatório Anual  
Memorandos  
Agenda  
Telefones



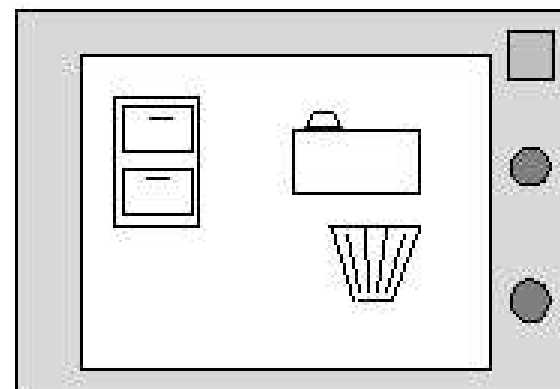
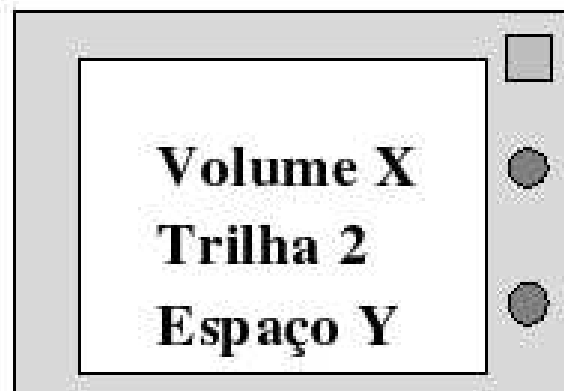


# Consistência

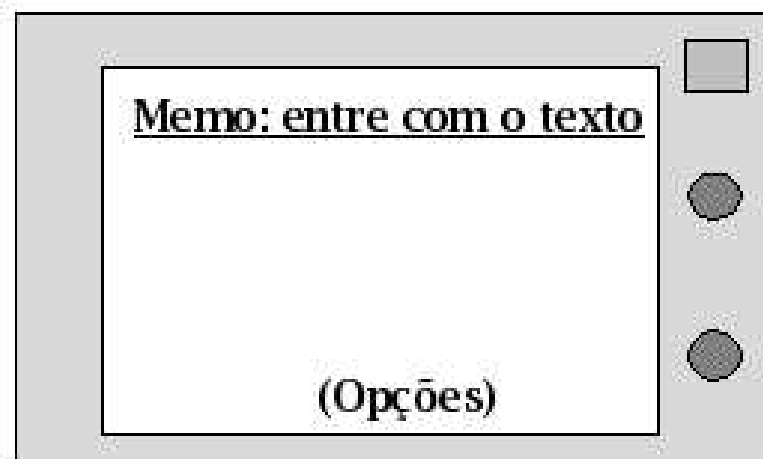
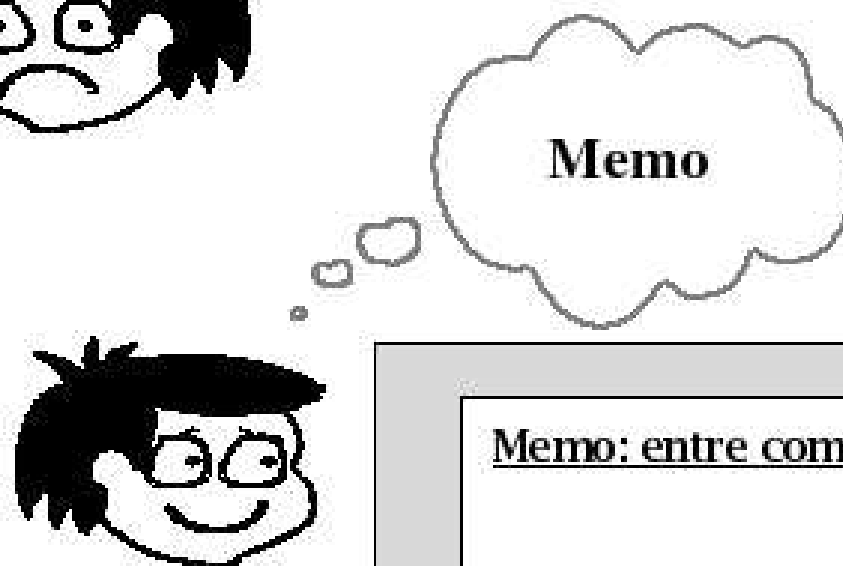
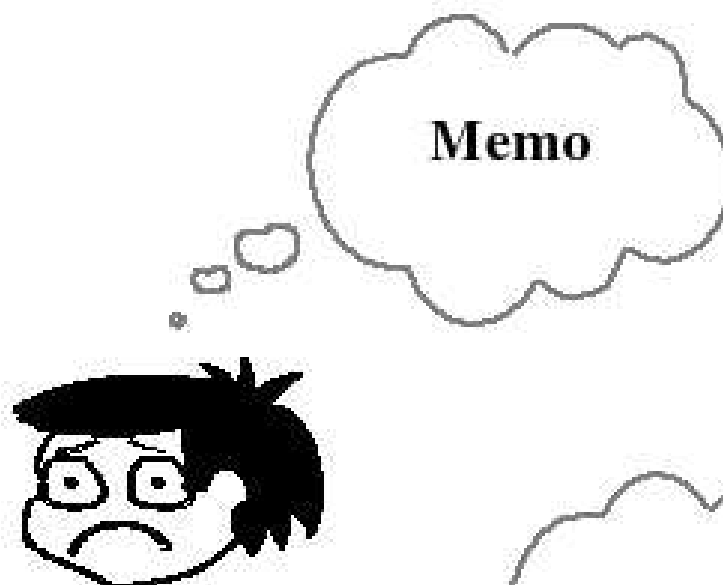
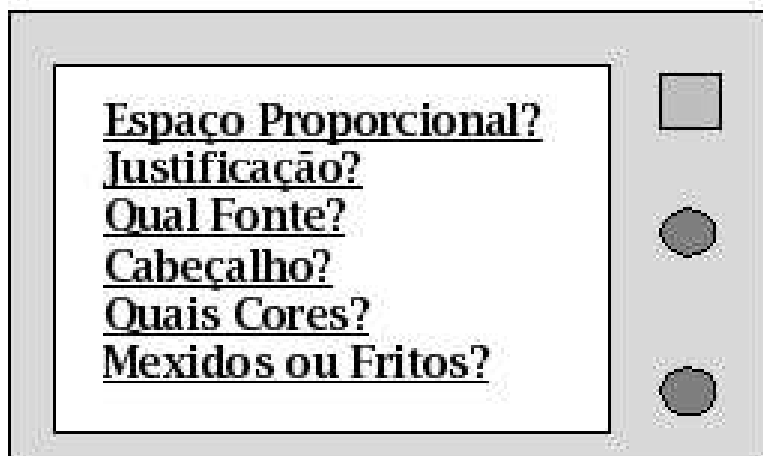


# Familiaridade

??????

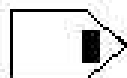


# Simplicidade



# Manipulação Direta

**Troque  
“Cadeia”  
na linha 10  
por “Prisão”**



# Controle

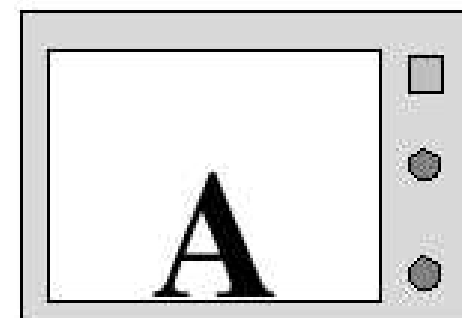
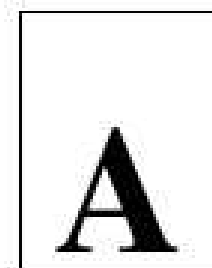
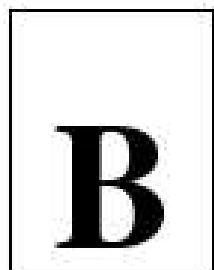
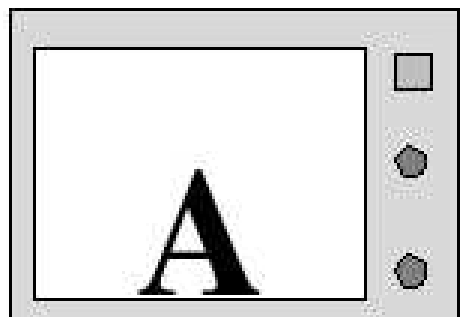
**Entre com o  
comando!  
Comando  
Ilegal!**



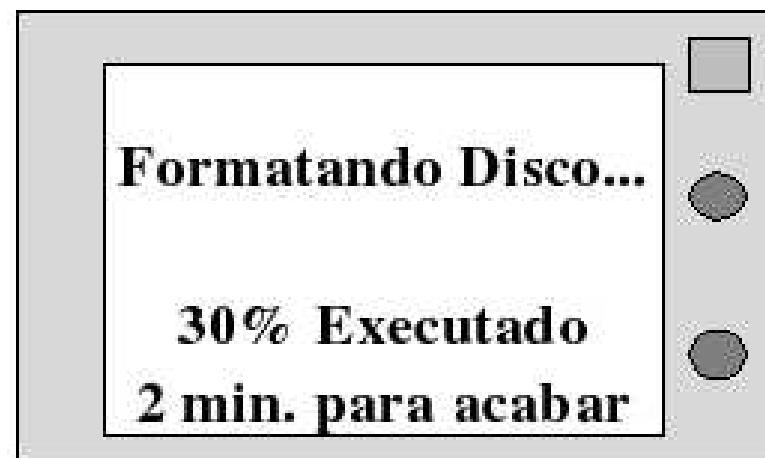
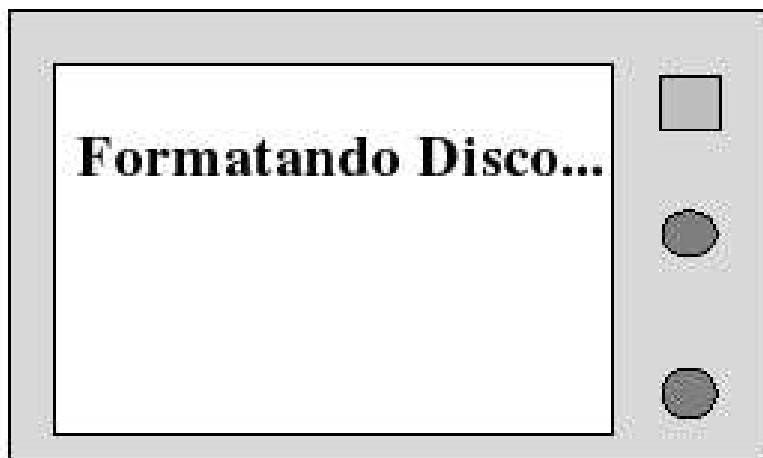
**Pronto...**

**Comando não  
reconhecido...**

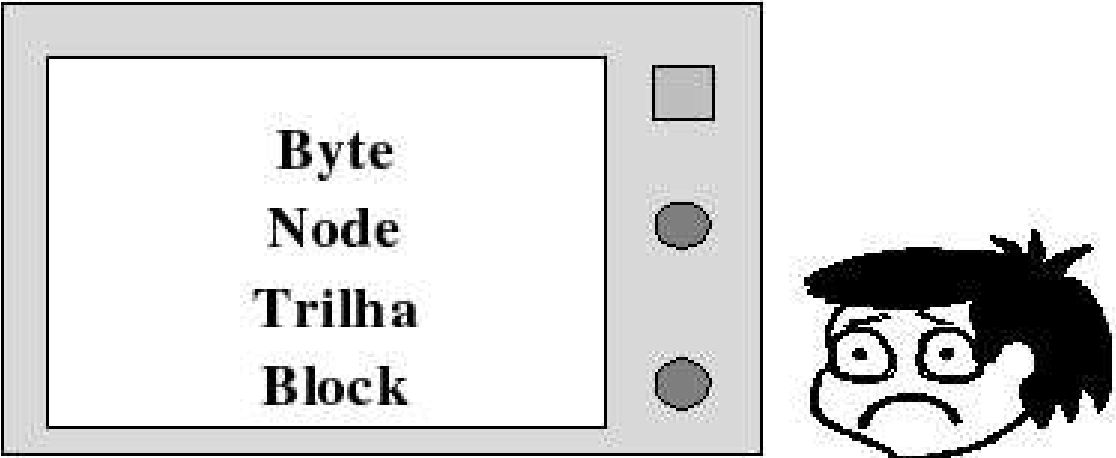
# WYSIWYG



# Respostas e Feedback



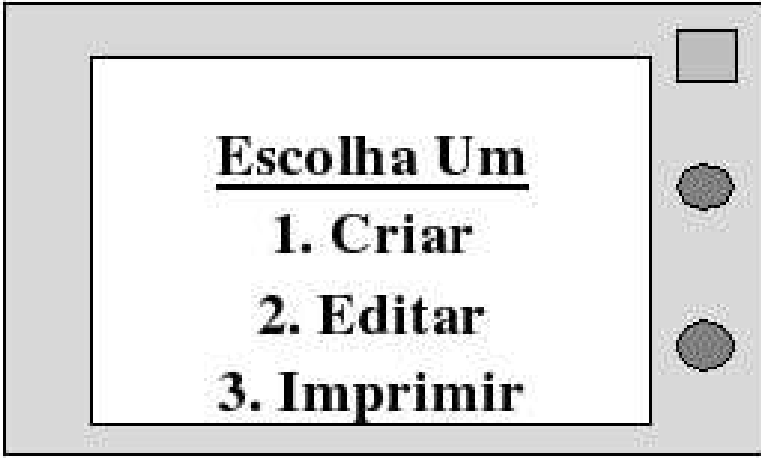
# Encapsulamento da Tecnologia



A window with a light gray border and a white interior. On the right side of the window, there are three controls: a small square button at the top and two circular buttons below it. The window contains a list of terms stacked vertically.

**Byte**  
**Node**  
**Trilha**  
**Block**

A cartoon character with black hair and glasses, looking sad with a downturned mouth.



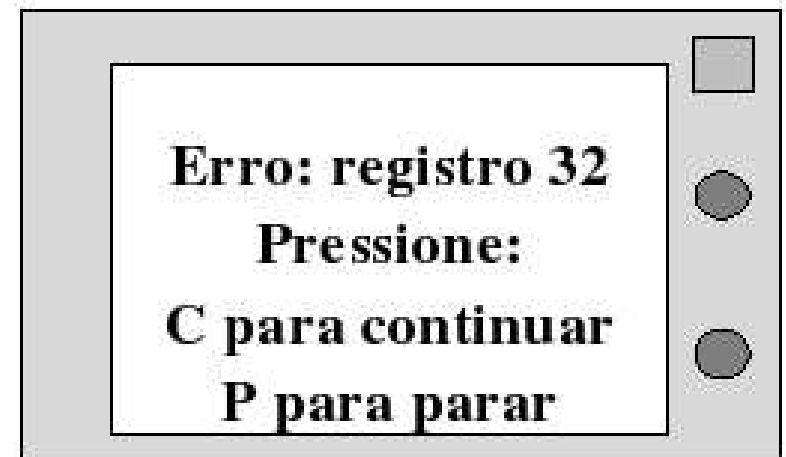
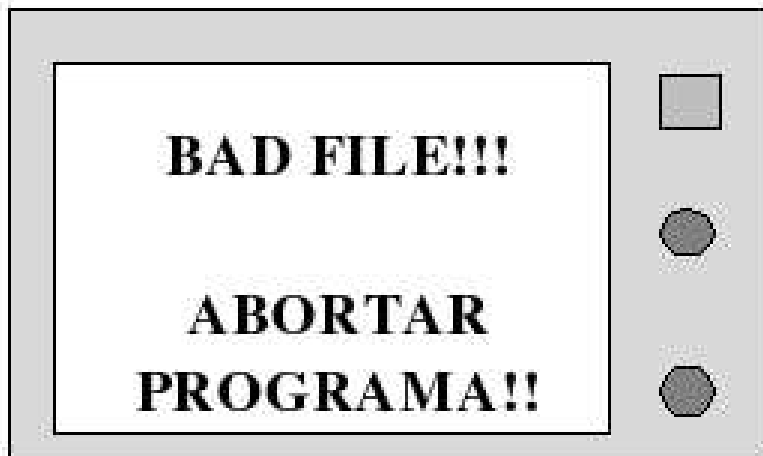
A window with a light gray border and a white interior. On the right side of the window, there are three controls: a small square button at the top and two circular buttons below it. The window contains a menu with a title and three numbered options.

**Escolha Um**  
**1. Criar**  
**2. Editar**  
**3. Imprimir**

A cartoon character with black hair and glasses, smiling with an upward-curved mouth.



# Robustez



# Proteção

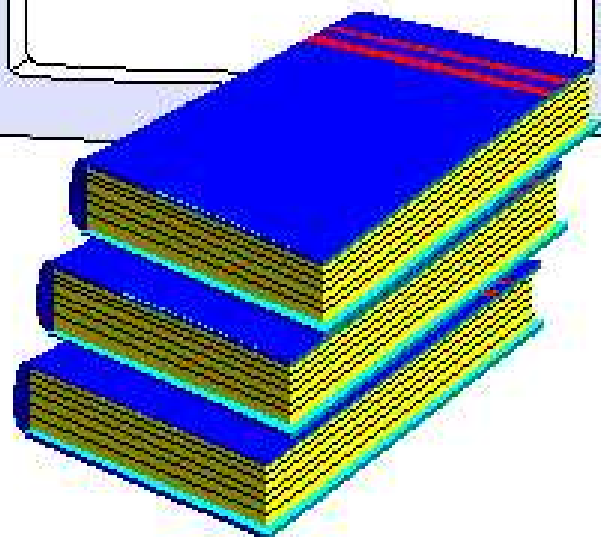
**TODOS  
OS  
ARQUIVOS  
APAGADOS**



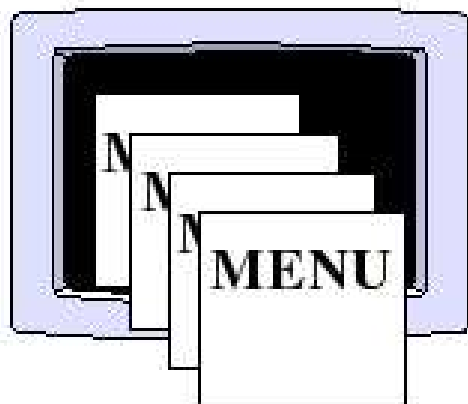
**UNDO**

**Você confirma  
o comando?  
(Pressione S ou N)**

# Facilidade de Aprendizado



# Facilidade de Uso



**EXPERT**



**NOVICE**



**NOVICE**



**EXPERT**



# Como Atingir Qualidade

- ➡ Conhecendo a teoria
- ➡ Desenvolvendo junto com o usuário
- ➡ Conhecendo as melhores práticas e *guidelines*
  - Siga um conjunto de Princípios
- ➡ Analisando o Resultado

Professor:

Geraldo Xexéo  
UFRJ

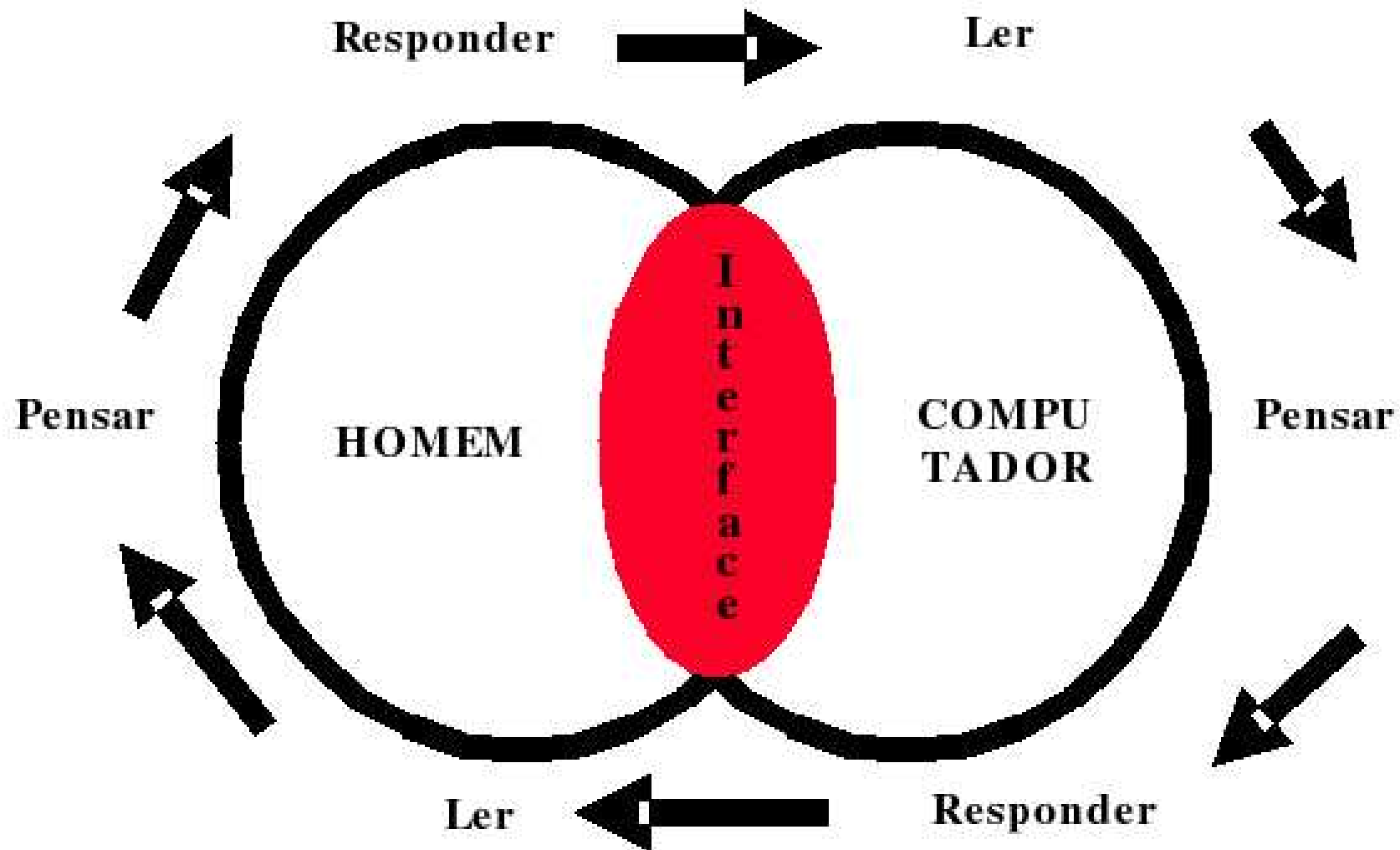
Conteúdo:

**Teoria Básica para  
Interface Homem  
Computador**

# Interação Homem Computador

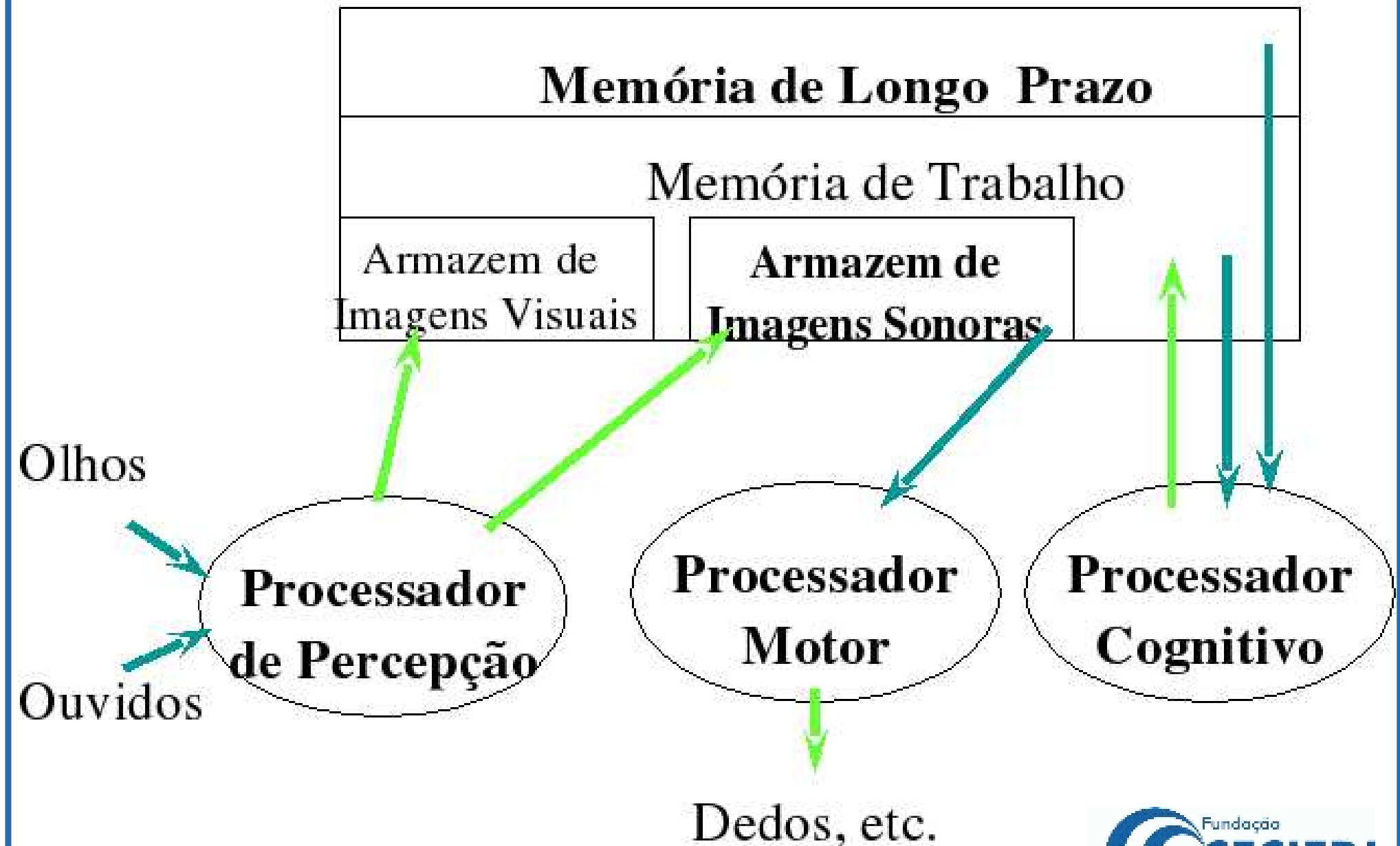
➡ A interação homem computador corresponde a um ciclo, onde cada parte recebe informações, as processa e emite novas informações

# IHC





# Model Human Processor



# Sistema Cognitivo



**MLP infinita**  
**MLP complexa**  
**Aprendizado**  
**Reconhecimento**  
**de Padrões**

**MCP limitada**  
**MCP de pequena**  
**duração**  
**Dado a erros**  
**Acesso a MLP não é**  
**confiável**  
**Processamento Lento**

# A Memória de Curto Prazo

- ➡ Rimas são difíceis de diferenciar
- ➡ Perda com a menor distração
- ➡ Taxa de esquecimento aumenta
  - ▬ com a complexidade
  - ▬ com a quantidade de informação
  - ▬ com a semelhança (confusão?)
- ➡ Nomes são melhor lembrados quando associados a imagens do que a palavras
- ➡ Idade diminui quantidade de informação

# Ainda as Memórias



## Memória de Curto prazo

- = 6174591765 vs. (617) 459-1765
- = DECIBM GMC vs. DEC IBM GMC
- = Acesso rápido (~ 70ms)



## Memória de Longo Prazo

- = Episódica e semântica
- =  **muito grande**  (se não for ilimitada)
- = Acesso lento (>100ms)



## Memória Sensorial!!

- = Ao toque, por exemplo

# Percepção

➡ Estímulos que ocorrem muito rapidamente podem se fundir

— cinema

➡ Causalidade (por percepção)

— dois estímulos distintos podem se fundir se o primeiro parece causar o segundo

# Passos da Execução de Uma Ação

- ➡ Formar o objetivo, o que se deseja no sentido mais amplo (por exemplo, "matar a sede").
- ➡ A Execução, dividida em 3 passos:
  - ▢ Formar a intenção, o que se fará (por exemplo, "beber água" ou "beber um suco").
  - ▢ Especificar a ação, (algo como "ir a geladeira, pegar uma garrafa de água, ir ao armário, pegar um copo, colocar água no copo e beber")[1].
  - ▢ Executar a ação,
- ➡ A Avaliação, dividida também em 3 passos
  - ▢ Perceber o estado do mundo,
  - ▢ Interpretar o estado do mundo e
  - ▢ Avaliar o resultado (em relação aos objetivos originais).

Professor:

Geraldo Xexéo  
UFRJ

Conteúdo:

**Desenvolvendo com  
o Usuário**

# Desenvolvendo com o Usuário

- ➡ A única maneira de discutir como funcionará a interface com o usuário de um sistema é mostrar essa interface
- ➡ Prototipagem & Mock-ups
  - ▬ Baixa Fidelidade
  - ▬ Alta Fidelidade
- ➡ Storyboards
- ➡ Mapas de Tela/Navegação



# Protótipo

- ➡ Implementação simplificada do sistema
- ➡ Pode inclusive ser descartável
- ➡ Deve ter uma finalidade
  - ▬ validar um modelo de interface
  - ▬ um modelo de funcionamento ou ainda um algoritmo.
- ➡ Normalmente protótipos são construídos utilizando a ferramenta em que o sistema está ou será desenvolvido (podendo servir inclusive para validar essa ferramenta) e apresentam algum comportamento, mesmo que simulado.

# Vantagens de Protótipos

- ➡ Tem um bom potencial de redução do tempo do projeto,
- ➡ Diminui o risco de insucesso
- ➡ Excelente fator para o sucesso a curto (primeira-vez) e longo prazo de um sistema.

# Vantagens de Protótipos

- ➡ O uso de protótipos pode ser um motivador para a participação dos usuários no sistema, diminuindo o antagonismo e aumentando a cooperação entre desenvolvedores e futuros usuários.
- ➡ A visão de um algo já realizado melhora a moral do projeto como um todo

# Vantagens de Protótipos

- ➡ Facilitam em muito a validação de sistemas
  - = principalmente de novos sistemas, onde há certo grau de exploração da solução mais adequada.
- ➡ Além disso, podem facilitar a encontrar funções desnecessárias ou funções esquecidas, principalmente com usuários já acostumados com sistemas semelhantes.

# Riscos do Protótipo

- ➡ A possibilidade de perder tempo fazendo melhorias de baixa importância no protótipo
- ➡ Pode trazer como risco a confiança demasiada e um otimismo exagerado em relação a prazos, como uma decepção proporcional a seguir

# Riscos do Protótipo

- ➡ Riscos de qualidade historicamente associados aos protótipos, como a utilização no código final de código que foi criado para ser jogado fora, e conseqüentemente sem seguir regras e padrões de desenvolvimento

# Mock-up

- ➡ Uma representação da interface que não cumpre nenhuma finalidade a não ser demonstrar a uma proposta para a aparência final do sistema, sem a capacidade de simular seu comportamento (a não ser, possivelmente, a navegação entre telas).
- ➡ Um mock-up não precisa ser feito com uma ferramenta de programação, podendo ser feito com ferramentas de desenho, como os softwares Visio® ou SmartDraw®.
- ➡ Em computação, é normal usar o termo protótipo mesmo quando se trata de um mock-up.

# Protótipo de Baixa Fidelidade

➡ É um *mock-up* feito a mão da interface, basicamente um conjunto de desenhos, com a finalidade de demonstrar a aparência básica e simular, manualmente, o comportamento do sistema.



# Protótipo de Baixa Fidelidade



# Protótipo de Baixa Fidelidade

- ➡ A construção de protótipos de baixa fidelidade é algo que muitos fazemos sem nem mesmo nos darmos conta.
- ➡ Esse tipo de protótipo é totalmente diferente do protótipo tradicional proposto normalmente e que inclui a construção de um software.

# Protótipo de Baixa Fidelidade

- ➡ PBF podem ser desenhados em quadros negros, quadros brancos, sobre papel, sobre transparência, em "tablet PCs", ou qualquer outra forma, incluindo a união das citadas.
- ➡ Podem ser feitos a mão livre ou com auxílio de réguas, gabaritos ou outros materiais de desenho.
- ➡ Podem ser preto e branco ou coloridos.
- ➡ Podem usar materiais pré-desenhados, como *frames* de janelas, ou serem feitos a partir de colagens.

# Protótipo de Baixa Fidelidade

- ➡ Poucos são as habilidades necessárias para desenhar um PBF.
- ➡ Seu custo também é baixo e o ciclo de interação com o usuário muito rápido.
- ➡ PBFs podem ser desenhados junto com o usuário, inclusive em uma reunião de JAD.

# Protótipo de Baixa Fidelidade

➡ Um dos efeitos psicológicos interessantes é que o usuário, não vendo uma implementação, fica mais disposto a propor mudanças, pois não vê nenhum custo ou esforço associado às mesmas, o que, na prática, é verdade.

# PBF e Storyboards

- ➡ A principal característica de PBF é que eles são explicados e "executados", ou melhor, "encenados", manualmente.
- ➡ Partes do protótipo podem ser desenhadas com detalhes, enquanto outras podem ser só indicadas.
- ➡ Alguns objetos podem ser cortados em um tamanho apropriado, como caixas mensagens e menus.
- ➡ A construção dessa encenação é semelhante à técnica de storyboarding usada no cinema.

# Protótipo de Baixa Fidelidade

Arquivo Editar Clientes
Ajuda

S	Nome do Cliente	CNPJ	Média Mensal
<input type="checkbox"/>	Loja XYZ	XXX998821-97	R\$ 7000,00
<input type="checkbox"/>	Casa Varanda	998245231-77	R\$ 80.009,00

INSERIR
ATUALIZAR
REMOVER

### Detalhe de Cliente

CNPJ:   
NOME:   
RUA:   
NÚM:  COMPL:   
CEP:  BAIRRO:   
CIDADE:  ESTADO:   
TELEFONES:    
FAX:   
EMAIL:

SALVAR
CANCELAR

### Login

USUÁRIO   
PASSWORD

OK

CADASTRO

ESQUECI MINHA SENHA

### Confirmação

Apagar Cliente 127

SIM
NÃO

### Deseja Salvar antes de fechar?

SIM
NÃO
CANCELAR

# PBF e Storyboards

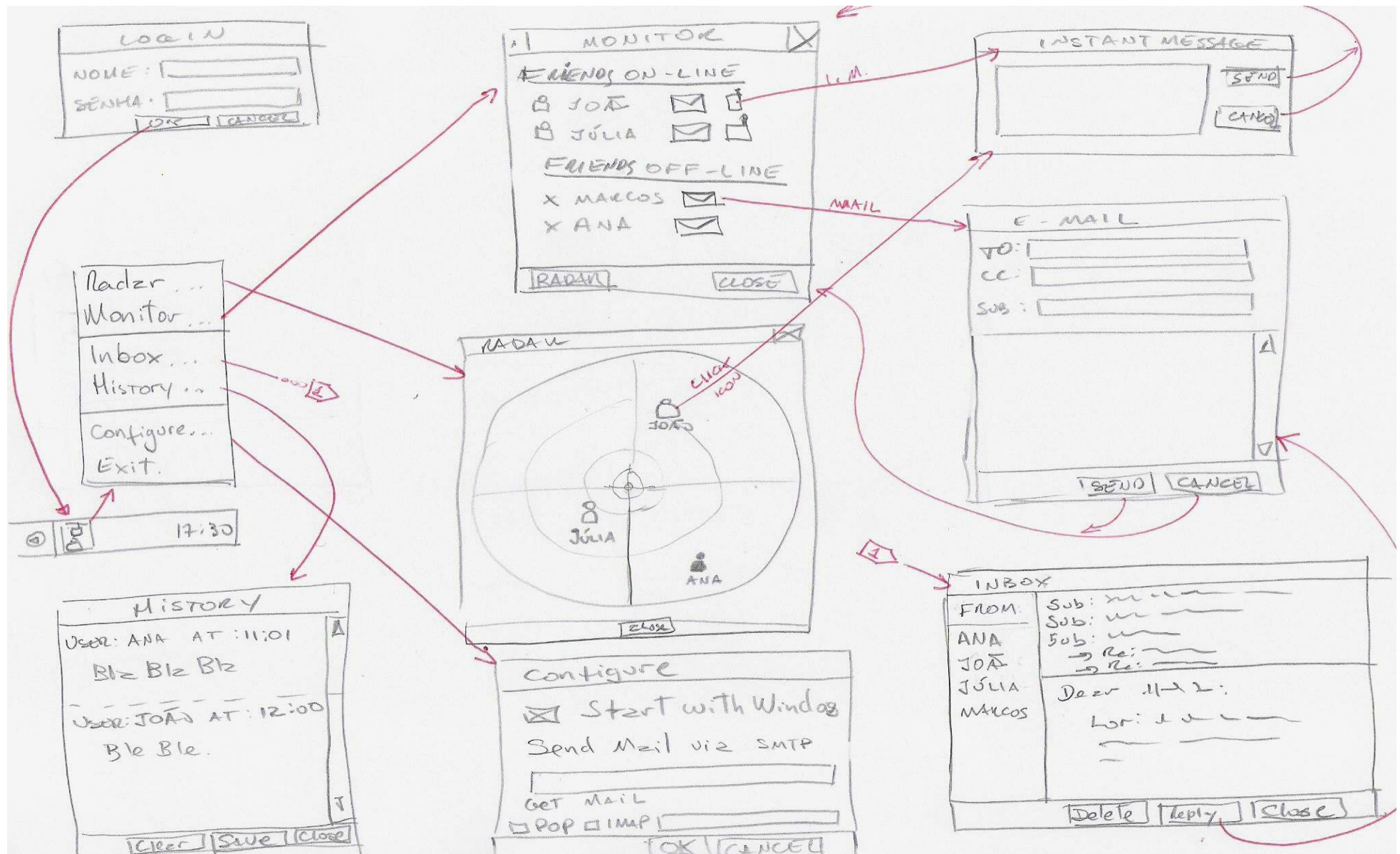
- ➡ comportamento de um PBF é executado por uma pessoa, em uma estratégia conhecida como "Mágico de Oz"
  - ➡ sua documentação não é tão simples.
- ➡ A partir da encenação do uso da interface é feita uma avaliação, que pode levar a necessidade de aceitá-la, melhorá-la ou até mesmo iniciar tudo do início.



# Storyboard

- ➡ Um projeto de uma sequência de cenas cinematográficas muito utilizado na publicidade, animação e em cinema em geral. À primeira vista um storyboard parece uma história em quadrinhos.
- ➡ Apesar do storyboard não ser uma HQ propriamente dita, por não possuir balões nem se destinar à reprodução, preserva as características de divisão de ação em quadros.

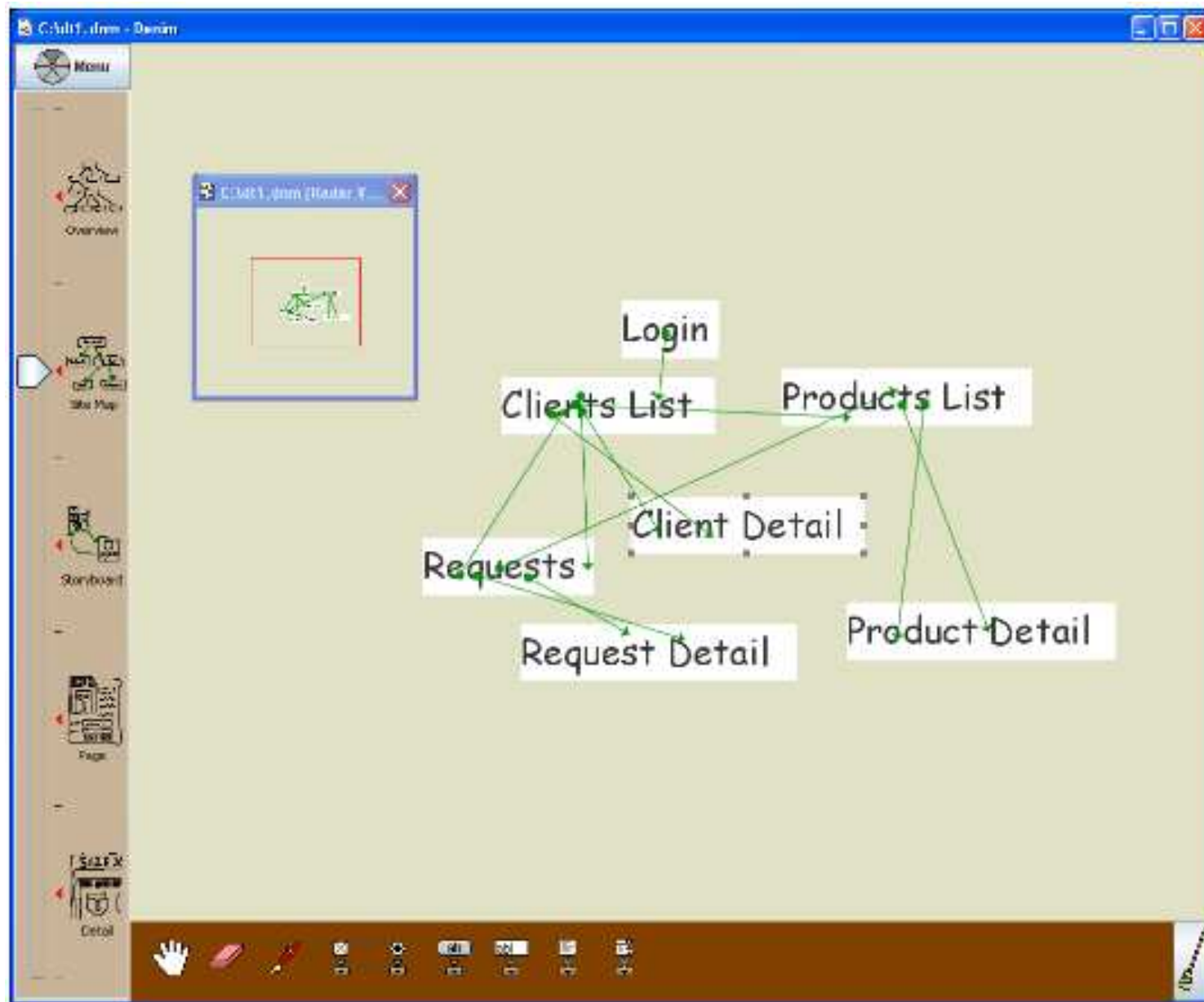
# Storyboard



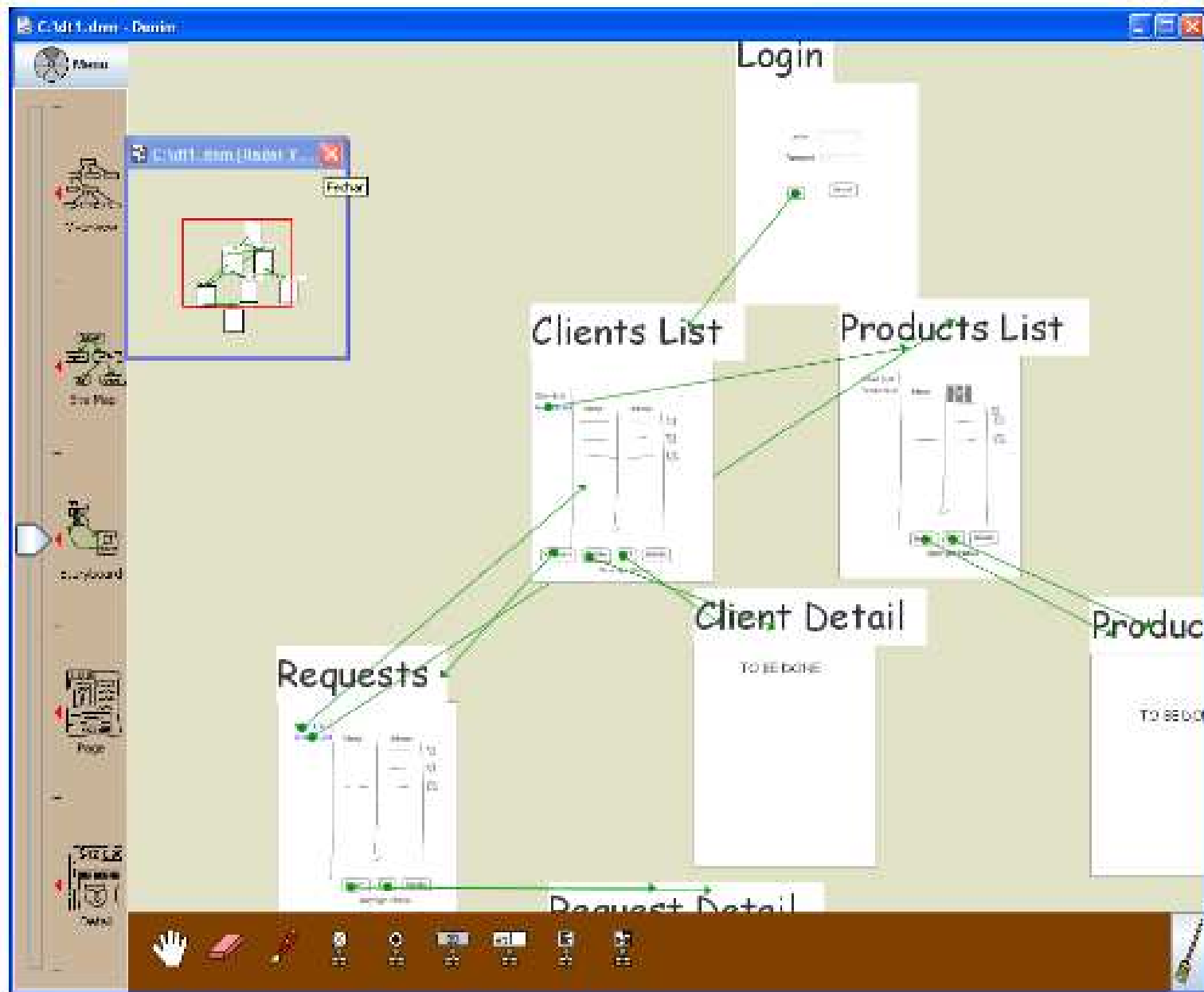
# PBF com Software

- ➡ Também é possível desenvolver PBF com software
- ➡ Um exemplo disso é o software DENIM, projetado para ser utilizado com computadores "tablete"
  - ▬ Pode ser utilizado com mouse

# Denim (1/3)



## Denim (2/3)



# Denim (3/3)

C:\Id11.dnm [Run] - Denim

Back Forward Refresh

## Clients List

Client List	Name:	Adress:	
Product List:	_____	_____	<input type="checkbox"/>
	_____	_____	<input type="checkbox"/>
	_____	_____	<input type="checkbox"/>

Requests Delete Edit Delete

Copyright Notice

# O Software Denim

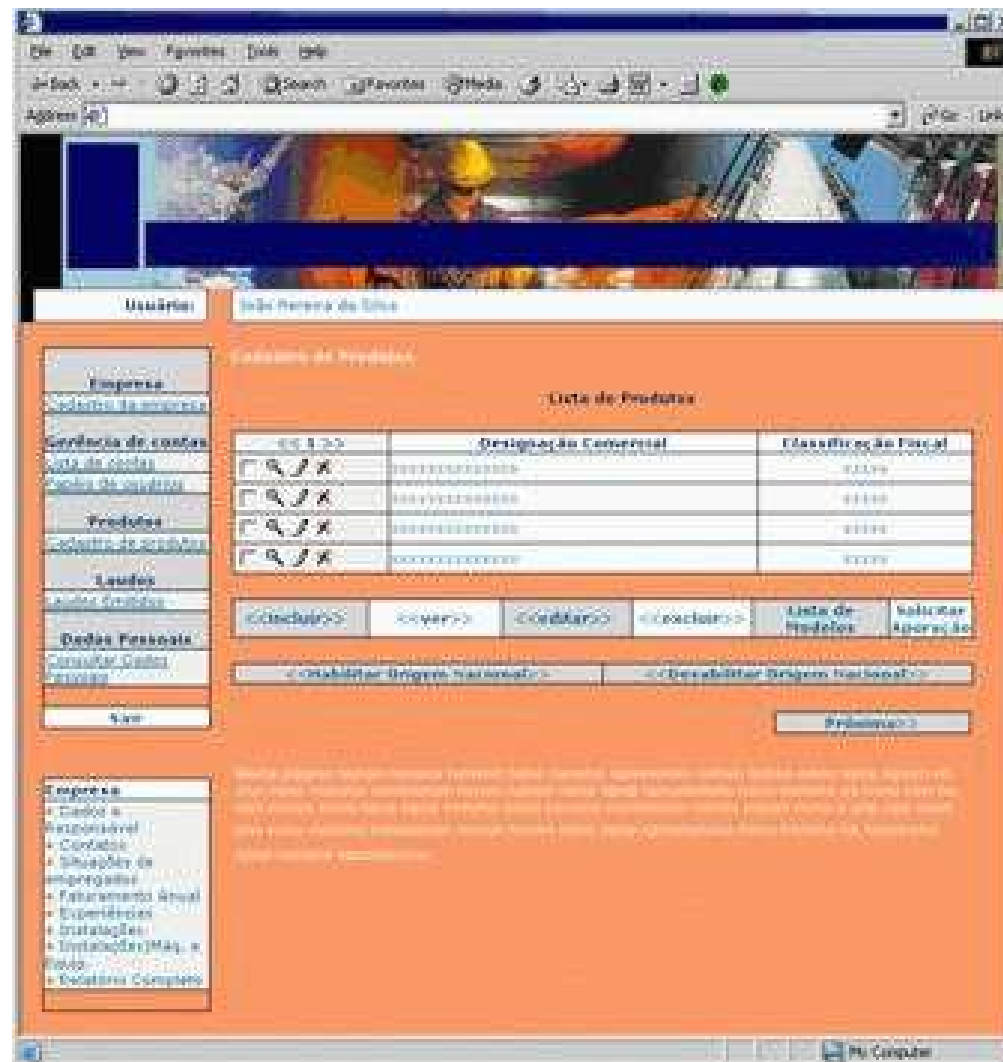
- ➡ Desenvolvido por um grupo especializado em interface
- ➡ Prova de conceito em interface com o usuário
- ➡ Pode ser obtido em <http://dub.washington.edu/denim/>

# Protótipo de Alta Fidelidade

- ➡ É um mock-up feito de forma a se assemelhar ao software final, sendo normalmente construído na linguagem de desenvolvimento ou em uma ferramenta com resultados similares.
- ➡ Protótipos de alta fidelidade são executados pelo computador.



# Protótipo de Alta Fidelidade



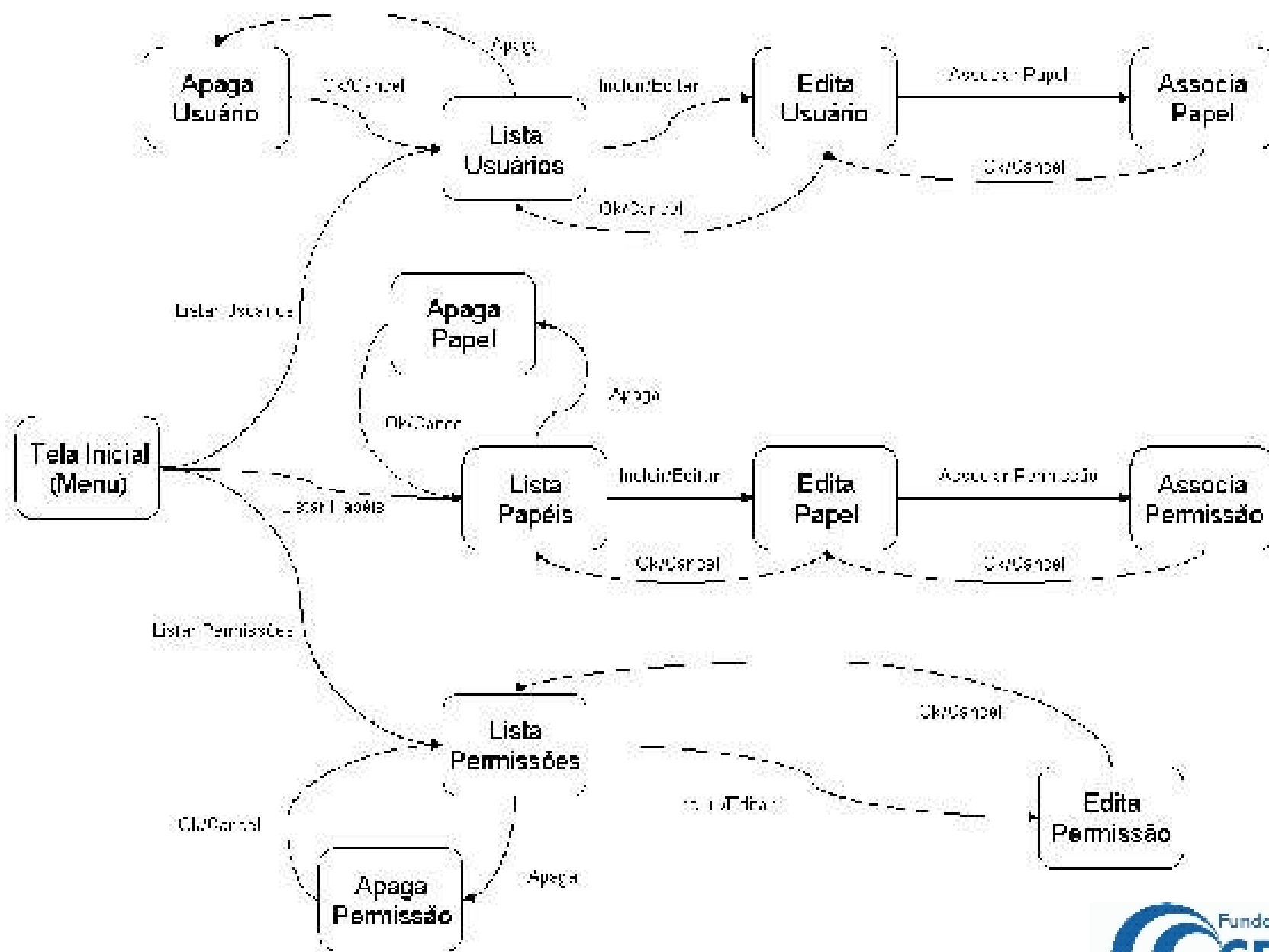
# Software para Alta Fidelidade

- ➡ Visio
- ➡ SmartDraw
- ➡ Ferramentas de Design
- ➡ As próprias ferramentas de desenvolvimento

# Mapa de Telas/Navegação

- ➡ Outra ferramenta, normalmente mais adequada para o desenvolvedor, que permite a compreensão de como é possível navegar nas telas do sistema

# Mapa de Telas/Navegação



Professor:

Geraldo Xexéo  
UFRJ

Conteúdo:

**Modelo Mental**

# Modelo Mental

- ➡ O usuário faz um modelo mental do sistema
- ➡ Faz suposições sobre o que deve ocorrer no sistema
- ➡ Baseado no aprendizado que teve em outra parte do sistema ou em outros sistemas

# O Modelo Mental do Usuário

- ➡ O usuário já possui um modelo mental que descreve a tarefa que seu software vai auxiliar
- ➡ Esse modelo vem da combinação de várias experiências desse usuário, tanto com o mundo real quanto com outros softwares.

# Levantando o Modelo Mental

- ➡ Antes de desenvolver seu projeto, investigue qual o modelo mental da tarefa que sua aplicação vai tomar parte
- ➡ Investigue as metáforas utilizadas.
- ➡ Procure atender as expectativas do usuário quanto aos componentes que devem existir na tarefa e o fluxo de trabalho que ela executa na sua interface
  - Organizando menus e barras de ferramenta
  - Adequando seu uso das janelas e espaços na tela as tarefas



# Exemplo de Modelo Mental

- ➡ Se em uma parte do sistema ao tentar apagar um objeto o usuário tem a oportunidade de desistir ou voltar atrás, vai esperar que o mesmo aconteça em todo o sistema.

# Princípios do Modelo Mental

- ➡ Familiaridade
- ➡ Simplicidade
- ➡ Disponibilidade
- ➡ Explorabilidade

# Familiaridade

- ➡ O modelo mental é baseado na experiência
- ➡ Quando possível, melhore a interface com o usuário com componentes que reflitam o modelo esperado

# Simplicidade

- ➡ O modelo mental de uma tarefa é simplificado
- ➡ Tem o foco nos principais componentes da tarefa
- ➡ Os detalhes, mesmo quando existentes, não devem competir com os componentes básicos pela atenção do usuário

# Disponibilidade

- ➡ A interface deve ser organizada e sem excesso de informação
- ➡ Porém, as necessidades do usuário devem estar prontamente disponíveis
  - ▢ Por exemplo, evite estruturas muito profundas de menu

# Explorabilidade

➡ Os usuários devem ser encorajados a descobrir a funcionalidade

- Por meio de dicas de como usar a interface

- Por exemplo, um botão **deve parecer** um botão

➡ Não esconda funcionalidade

Professor:

Geraldo Xexéo  
UFRJ

Conteúdo:

**Guias e Princípios**

# Princípios da Interface com Usuário

- ➡ Estrutura
- ➡ Simplicidade
- ➡ Visibilidade
- ➡ Feedback
- ➡ Tolerância
- ➡ Reuso



# Princípio da Estrutura



Seu projeto deve organizar a interface com o usuário de forma significativa

- ⇒ Reconhecida pelo usuário
- ⇒ Baseada em modelos consistentes
  - ⇒ Coisas relacionadas juntas, coisas não relacionadas separadas
  - ⇒ Coisas similares devem parecer similares, coisas diferentes devem parecer diferentes

# Princípio da Simplicidade

- ➡ Seu projeto deve fazer tarefas simples serem simples
- ➡ Deve se comunicar claramente e na linguagem do usuário
- ➡ Deve fornecer "aceleradores" que sejam facilmente associados aos procedimentos mais longos

# Princípio da Visibilidade

- ➡ Todas as opções e materiais para uma tarefa devem estar visíveis
- ➡ A atenção do usuário não deve ser atrapalhada com informação não específica a tarefa ou redundante
- ➡ O número de alternativas não pode confundir o usuário

# Princípio da Tolerância

- ➡ O custo de errar deve ser reduzido
- ➡ Deve ser possível desfazer e refazer
- ➡ Erros também devem ser evitados
- ➡ É possível tolerar entradas variadas e interpretá-las da maneira correta, quando razoável

# Princípio do Feedback

- ➡ Os usuários devem ficar informados do que está acontecendo
- ➡ Mudanças de estado
- ➡ Mudanças de condições
- ➡ Erros e exceções relevantes
- ➡ Linguagem clara e concisa
- ➡ Termos não ambíguos
- ➡ Linguagem familiar ao usuário

# Princípio do Reuso

- ➡ Seu projeto deve reusar componentes externos e internos
- ➡ Manter a consistência de comportamento e aparência
- ➡ O usuário não deve precisar repensar ou lembrar.

# Princípios de Interface com Humanos da Apple

- ➡ **Metáforas**
- ➡ Refletir o modelo mental do usuário
- ➡ **Ações explícitas e implícitas**
- ➡ Manipulação direta
- ➡ **Controle do usuário**
- ➡ Feedback e Comunicação

## O que temos de novo?

- ➡ Consistência
- ➡ WYSIWYG
- ➡ Perdão
- ➡ Estabilidade
- ➡ Estética (integrada)
- ➡ **Ausência de Modo**
- ➡ Complexidade Gerenciada

# Metáforas

- ➡ Trazem para a interface conceitos e funcionalidade que o usuário está habituado a usar no mundo real
- ➡ São o bloco básico dos modelos mentais
- ➡ Devem ser óbvias
- ➡ Criam expectativas
- ➡ Sugerem um uso
  - ▬ Não estão limitadas ao uso do mundo real
  - ▬ Exemplo
    - ▬ Um cesta de lixo tem um tamanho máximo, mas isso não deve ser uma propriedade de uma cesta de lixo do sistema operacional



# Ações Explícitas

- ⇒ Determinam claramente o resultado da manipulação de um objeto
- ⇒ Exemplo: comandos em menus
- ⇒ Não exigem que o usuário memorize algo

# Ações Implícitas

- ⇒ Fornecem o resultado por meio de dicas visuais e contextos
- ⇒ Exemplo: Drag and Drop
- ⇒ Exemplo: Puxar para o Lixo
- ⇒ Exige o reconhecimento dos objetos envolvidos, o conhecimento das ações possíveis e as suas consequências

# Explicítas x Implícitas

- ➡ Devemos examinar o modelo mental do usuário para determinar que ações são apropriadas

# Usuário no Controle

- ➡ O usuário deve iniciar as ações
- ➡ Todas as ações devem estar disponíveis para o usuário
  - Usuários novatos podem se beneficiar com alternativas limitadas
  - Deve haver a possibilidade de expandir para as alternativas completas

# Ausência de Modo

- ➡ Modo: um estado do sistema que só permite que algumas ações sejam tomadas
- ➡ Sempre que possível, permita que os usuários façam tudo que eles quiserem, a qualquer momento.
- ➡ Evite o uso de interfaces modais

# Guidelines

- ➡ Normalmente, cada interface possui um guia de implementação, que indica como e de que modo usar os seus componentes
- ➡ Sua empresa também deve ter padrões de qualidade

Professor:







Geraldo Xexéo  
UFRJ

Conteúdo:

**Analisando o Resultado**

# Critique você mesmo

## Observe o usuário usando o sistema

-  Dificuldades
-  Tarefas longas
-  Tarefas feitas erroneamente
-  Trabalho perdido
-  Procure por padrões
-  Não ajude o usuário a usar o sistema

## Escute o usuário falando sobre o sistema



# Treinamento e Manual

- ➡ Usuários não lêem manual
- ➡ Usuários não recebem treinamento formal
  - Pelo menos na maioria dos casos
- ➡ Usuários são treinados para a tarefa, não para o software

# Experimentos de Interface (1/10)

- ➡ Se apresente e descreva o objetivo da observação em termo gerais
  - ➡ Não diga especificamente o que vai ser observado
  - ➡ Deixe claro que o testado é o produto, não o participante

# Experimentos de Interface (2/10)

- ➡ Diga ao participante quanto tempo vai demorar o teste
- ➡ Diga que está OK se ele desistir a qualquer momento, por qualquer razão
  - ▬ O participante nunca deve se sentir pressionado
  - ▬ A desistência pode indicar que a tarefa é muito difícil ou complexa

# Experimentos de Interface (3/10)

- ➡ Escolha uma metodologia
- ➡ Exemplo: Pensando Alto
- ➡ Explique como funciona
  - ▬ O participante deve falar alto durante a observação, falando o que vier a mente enquanto trabalha
    - ▬ Foco em pensar e planejar
  - ▬ Observe
    - ▬ Expectativas
    - ▬ Intenções
    - ▬ Estratégias de solução de problemas
    - ▬ Frustrações

# Experimentos de Interface (4/10)

- ➡ Use toda a informação que puder encontrar na fala do usuário
  - ▬ Detalhes do modelo mental
- ➡ Se necessário, ensine ao usuário como descrever uma tarefa simples, como fazer café

# Fazendo Café

- ➡ Quero fazer café
- ➡ Preciso de pó, água e cafeteira
- ➡ A água está na pia
- ➡ O café está no armário
- ➡ A cafeteira já está na mesa
- ➡ Abro o armário
- ➡ Pego o pote de café
- ➡ Abro o pote de café
- ➡ Preciso de uma colher
- ➡ A colher está na gaveta
- ➡ Abro a gaveta
- ➡ Pego a colher
- ➡ Pego uma colher de café e coloco na cafeteira
- ➡ A cafeteira está fechada
- ➡ Abro a cafeteira...

# Experimentos de Interface (5/10)

- ➡ Explique de modo geral o que o participante vai fazer
- ➡ Explique quais os materiais e qual a sequencia em que deve usá-los
- ➡ Explique o objetivo de cada equipamento na sala e como será usado no teste
  - ▬ Software, hardware, câmeras, etc...
- ➡ Se for necessário, demonstre seu produto, mas **não demonstre o que vai ser testado**

# Experimentos de Interface (6/10)



Não ajude ou interfira com o participante

- Se for necessária sua ajuda, isto deve estar determinado no protocolo do experimento
  - Por exemplo: se o participante tentar por 3 minutos fazer algo sem ajuda e não conseguir, a ajuda será dada.
- Se o participante ficar muito frustrado, pode ser melhor intervir logo para evitar que ele desista



# Experimentos de Interface (7/10)



Conclua explicando o que você estava tentando encontrar e responda as perguntas do participante

# Experimentos de Interface (8/10)

- ➡ Use os resultados
- ➡ Você verá os usuários fazendo coisas que você pode não ter antecipado
- ➡ Não culpe o usuário pelos erros que cometem
- ➡ Lembre que o objetivo do teste é descobrir que partes do produto podem ser difíceis de usar

# Experimentos de Interface (9/10)

- ➡ Procure por padrões
- ➡ O problema de um não representa necessariamente um problema geral
- ➡ Analise detalhadamente porque um usuário isolado teve um problema
  - Pode ser descartado?
  - É um problema real?

# Experimentos de Interface (10/10)

- ➡ Revise os resultados
- ➡ Use equipe com representantes de todas as áreas funcionais da empresa
  - ▬ Use a experiência desses participantes nos seus pontos de vista específicos
  - ▬ Gerência
  - ▬ Marketing
  - ▬ Qualidade
  - ▬ Design
  - ▬ Engenharia...

# Problemas Detectados

- ➡ Problemas com um usuário não significam problemas com todos os usuários
  - Mas indicam uma área de preocupação
- ➡ Problemas com todos os usuários são certamente problemas

## Aula 30

Professor:

Geraldo Xexéo  
UFRJ

Conteúdo:

**FIM: Modelo de Interface**