



Instituto Superior de Engenharia

Politécnico de Coimbra

Interação Pessoa-Máquina
**CTeSP Tecnologias e Programação de Sistemas de Informação
(Cantanhede)**

Professor: João Leal

joao.leal@isec.pt



**Instituto Superior
de Engenharia**
Politécnico de Coimbra



GUIAS PARA WEBDESIGN

Desenho da interacção

- Os princípios de desenho da interacção aplicam-se genericamente a diferentes sistemas
 - ▣ Sítios Web
 - ▣ Sistemas Multimédia interactivos
 - ▣ Aplicações para telemóveis
 - ▣ Aplicações de computadores
 - ▣ Aplicações para quiosques multimédia
 - ▣ ...
 - ▣ Cockpits de aviões!

Desenho da interacção

- Os designers da interacção precisam de:
 - ▣ Ajuda para reconhecerem problemas e mau design
 - ▣ Direcções para melhorarem
 - **Princípios genéricos sobre Usabilidade**
 - Normas sobre Usabilidade
 - Regras de Desenho Gráfico
 - ▣ Guias práticos, e não soluções “empacotadas”
 - ▣ Princípios concretos, não psicologia cognitiva!
 - Guias de Boas práticas na Web

Princípios Genéricos

- Usabilidade
 - ▣ Princípios de Desenho de Norman
 - ▣ Heurísticas de Jakob Nielsen
 - ▣ Princípios de Desenho de Constantine & Lockwood
 - ▣ Princípios de Togs
 - ▣ Regras de ouro de Schneiderman

Princípios de Desenho de Norman

- Fornecer um bom modelo conceptual
- Tornar as coisas **visíveis**
 - O utilizador não deve ter problemas em perceber o estado do sistema e as alternativas para realizar determinada acção.
- Princípio do **Mapeamento**
 - Os utilizadores devem determinar claramente a relação entre as ações e os resultados, os controlos e os seus efeitos, e entre o estado do sistema e o que é visível.
- Princípio do **Feedback**
 - Os utilizadores devem receber feedback contínuo e informativo sobre o resultado das suas acções.

Princípios de Desenho de Norman

- Falar a linguagem do utilizador



Heurísticas de Jakob Nielsen

□ **Visibilidade** do sistema (H1)

- O sistema deve manter sempre o utilizador informado sobre o que está a acontecer, através de respostas apropriadas e dentro de um tempo razoável

□ **Correspondência** entre o sistema e o mundo real (H2)

- O sistema deve “falar” a linguagem do utilizador, com palavras, frases e conceitos que lhe são familiares, em vez de usar a linguagem do sistema, e de uma forma lógica e natural que evite termos técnicos.

□ **Controlo** e liberdade do utilizador (H3)

- O sistema deve fornecer formas do utilizador sair de estados não desejados sem ter que passar por diálogos extensos.

Heurísticas de Jakob Nielsen

- **Consistência** e aderência a normas (H4)
 - Evitar termos, situações e ações diferentes com o mesmo significado. Devem seguir-se as convenções das plataformas, normas e regras.
- **Prevenção de erros** (H5)
 - Desenhar o sistema de forma cuidadosa prevenindo os problemas antes de acontecerem.
- **Reconhecer** em vez de lembrar (H6)
 - Tornar os objectos, acções e opções visíveis.
 - O utilizador não deve ter que recordar informação de uma parte do sistema para outra.
 - As instruções de utilização do sistema devem ser visíveis e acessíveis quando necessárias.

Heurísticas de Jakob Nielsen

- **Flexibilidade** e eficiência na utilização (H7)
 - ▣ Acomodar diferentes níveis de experiência fornecendo “aceleradores” para os utilizadores experientes que são invisíveis para os novatos.
 - ▣ Permitir aos utilizadores adotarem ações frequentes.
- **Desenho** estético e minimalista (H8)
 - ▣ Os diálogos não devem conter informação, que é irrelevante ou raramente necessária. Cada unidade extra de informação num diálogo compete com unidades relevantes de informação e diminui a sua visibilidade relativa.
 - ▣ Simplicidade – Less is more
 - ▣ Apurar o desenho gráfico
 - ▣ ...

Heurísticas de Jakob Nielsen

- **Ajudar os utilizadores a reconhecer, diagnosticar e recuperar erros (H9)**
 - ▣ Mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicando de forma precisa o problema, e sugerindo uma solução construtiva.

- **Documentação e Ajuda (H10)**
 - ▣ A ajuda e documentação, quando necessária, deve estar concentrada nas tarefas dos utilizadores, e deve ser fácil, concisa, focada nas tarefas e acessível.

Princípios de Constantine & Lockwood

□ Princípio da **Simplicidade**

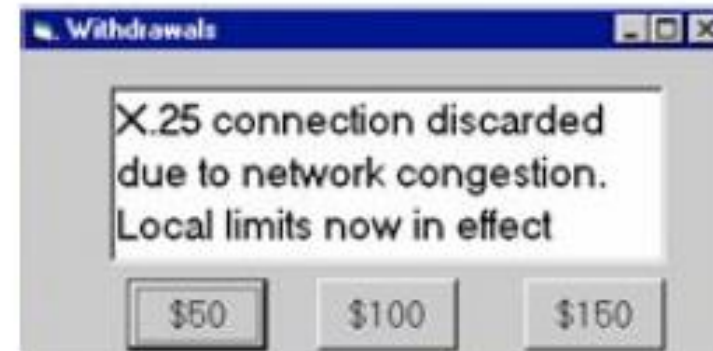
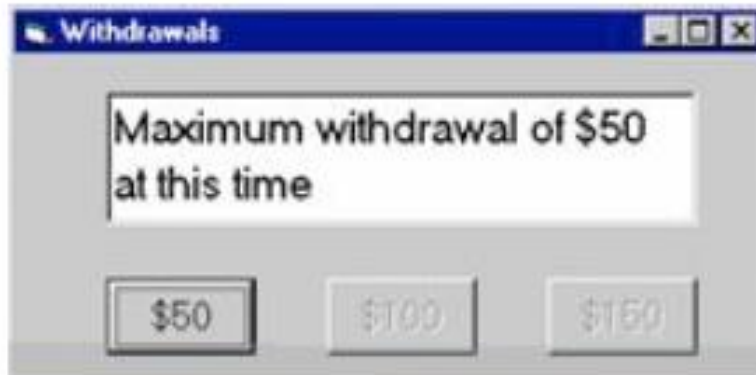
- ▣ Tornar as ações simples e comuns fáceis de executar, comunicando de forma clara, simples e na linguagem do utilizador
- ▣ Manter simples as tarefas simples e frequentes
- ▣ Concentrar-se no essencial e nas necessidades reais
- ▣ Oferecer atalhos razoáveis, e não arbitrários
- ▣ Reduzir os “passos tecnológicos”
 - Ex: botões “Actualizar quantidade” de alguns formulários web

Princípios de Constantine & Lockwood

- Princípio da **Visibilidade (H1)**
 - ▣ Manter visíveis todas as opções e materiais necessários para uma determinada tarefa, sem distrair os utilizadores com informação redundante.
- Princípio do **Feedback**
 - ▣ Manter os utilizadores informados das acções ou interpretações, das mudanças de estado ou condição, e dos erros e excepções que são relevantes ou de interesse para o utilizador. Oferecer essa informação de forma clara, concisa e não ambígua numa linguagem familiar aos utilizadores.

Princípios de Constantine & Lockwood

- Princípio do **Feedback**
 - ▣ Falar a linguagem do utilizador



Princípios de Constantine & Lockwood

□ Princípio do **Feedback**

- Ser claro, conciso, não ambíguo
- Transmitir condições, interpretações, mudanças de estado, progresso, erros e exceções
- Boas mensagens de erro
- Apenas quando for relevante ao utilizador/tarefa

Princípios de Constantine & Lockwood

□ Princípio da **Tolerância**

- ▣ As operações devem ser flexíveis e tolerantes a erros e/ou má utilização
- ▣ Fornecer um vasto leque de opções e acções
- ▣ Interpretar razoavelmente qualquer acção razoável do utilizador
- ▣ Prevenir erros do utilizador sempre que possível
- ▣ Minimizar o prejuízo dos erros
 - Tornar as acções reversíveis (Ex: Undo/Redo)

“ Software should at least not do something stupid when confronted with unexpected input or actions. ”

Constantine and Lockwood

Princípios de Constantine & Lockwood

□ Princípio da **Reutilização**

- ▣ Reutilizar componentes e comportamentos, mantendo a consistência (em aspeto/aparência e em comportamento/dinâmica)
 - Redução da necessidade dos utilizadores repensarem e recordarem informação.
- ▣ A reutilização ajuda os utilizadores E os programadores!
 - Facilidade de aprender e usar!
- ▣ Reutilização deve ser racional, evitar consistências absurdas ou designs consistentemente maus

Princípios de Constantine & Lockwood

- Princípio da **Reutilização** - Exemplos
 - ▣ Operações e elementos similares para tarefas similares
 - Tecla Ctrl + inicial do comando (ctrl+C, ctrl+S, ctrl+O, ...)
 - Como proceder para save, spelling, select, style?
 - ctrl+S, ctrl+Sp, ctrl+shift+L (Quebra da consistência)
 - ▣ Bons exemplos de Reutilização
 - Ctrl +C, Ctrl + V (Copy e Paste) é mantido em todas as aplicações!

Princípios de Togs

- 1. Princípio da **Antecipação**
 - ▣ Antecipar o que os utilizadores precisam
- 2. Princípio da **Autonomia** (=H3)
 - ▣ O utilizador deve ter o controlo do sistema
- 3. Princípio da **Color Blindness**
 - ▣ Não depender apenas da cor para codificar informação
- 4. Princípio da **Consistência** (=H4)
 - ▣ Principalmente, consistência com as expetativas do utilizador

Princípios de Togs

- 5. Princípio dos **Defaults**
 - Fáceis de alterar
- 6. Princípio da **Eficiência do utilizador**
 - Focar na produtividade do utilizador e não do sistema
- 7. Princípio das **Interfaces exploráveis**
 - “Give users well marked roads and landmarks, then let them shift to four-wheels drive”
- 8. Princípio da **Lei de Fitt**
 - Botões maiores são mais rápidos

Princípios de Togs

- 9. Princípio dos **Human-Interface Objects**
 - ▣ Familiares, consistentes, estáveis, auto-explicáveis
 - ▣ Ex: directorias, ficheiros, *recycle_bin*
- 10. Princípio da **redução da latência**
 - ▣ Distrair o utilizador minimizando tempos de espera
- 11. Princípio da **aprendizagem**
 - ▣ Idealmente: sem curva de aprendizagem
 - ▣ Limitar as exigências
 - Usabilidade e aprendizagem não são mutuamente exclusivas
 - Decidir qual a mais importante e atacar as duas

Princípios de Togs

- 12. Princípio das **Metáforas**
 - Escolher metáforas que permitam entender instantaneamente os detalhes do modelo conceptual
 - Criar imagens na mente do utilizador
- 13. Princípio da **protecção do trabalho do utilizador**
 - Assegurar que o utilizador nunca perde o trabalho
- 14. Princípio da **legibilidade**
 - Contraste das cores
 - Tamanho da fonte

Princípios de Togs

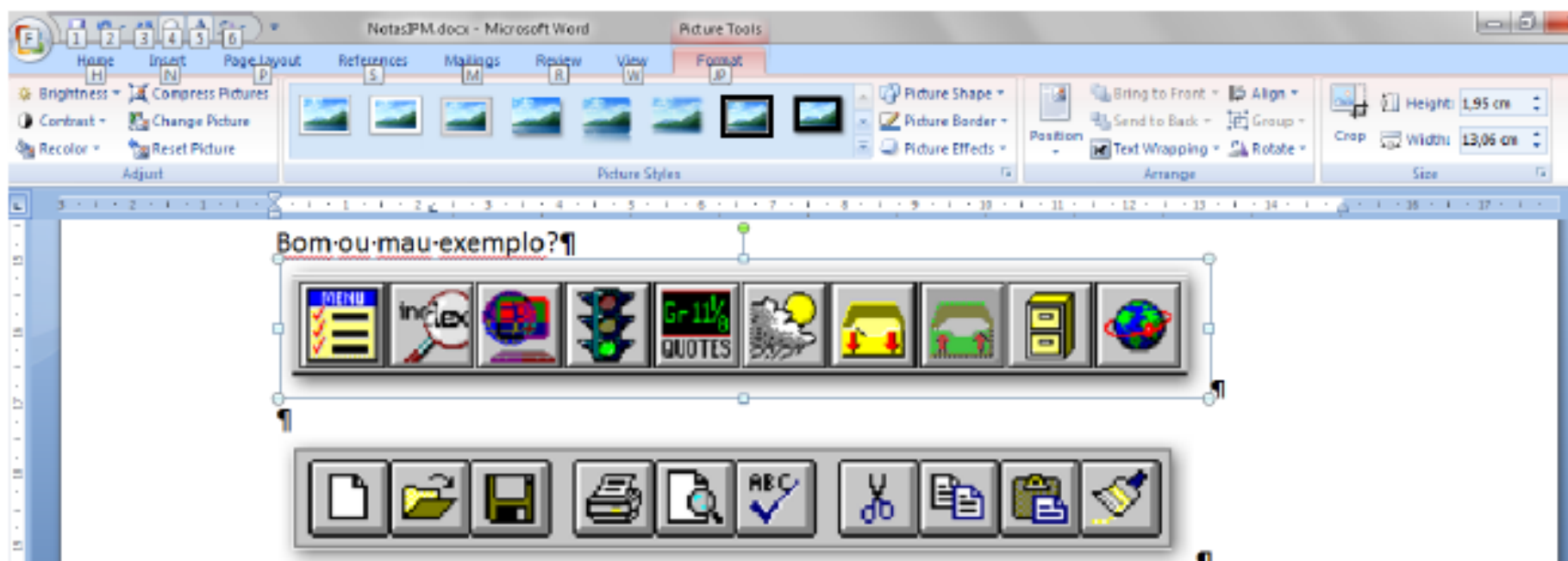
- 15. Princípio do **track state**
 - Onde estava o utilizador na última sessão
 - Cookies
- 16. Princípio da **navegação visível**
 - Reduzir ao mínimo a navegação
 - Navegação clara e natural
- 17. Princípio da **estrutura**
- www.asktog.com

Princípios de Tog

- 1. Princípio da **antecipação**
 - ▣ As aplicações devem tentar antecipar as necessidades e desejos do utilizador.
 - ▣ Não esperar que os utilizadores pesquisem, recolham informação ou evoquem as ferramentas necessárias.
 - Fornecer ao utilizador toda a informação e ferramentas necessárias a cada passo do processo.

Princípios de Tog

□ 1. Princípio da **antecipação**



Princípios de Tog

- 2. Princípio da **autonomia**
 - A autonomia do utilizador não significa que se abandonem regras
 - Usar mecanismos para manter os utilizadores informados
 - Manter a informação de estado atualizada e facilmente visível para o utilizador
 - Permitir sempre uma saída
 - Nunca deixar o utilizador encurralado!

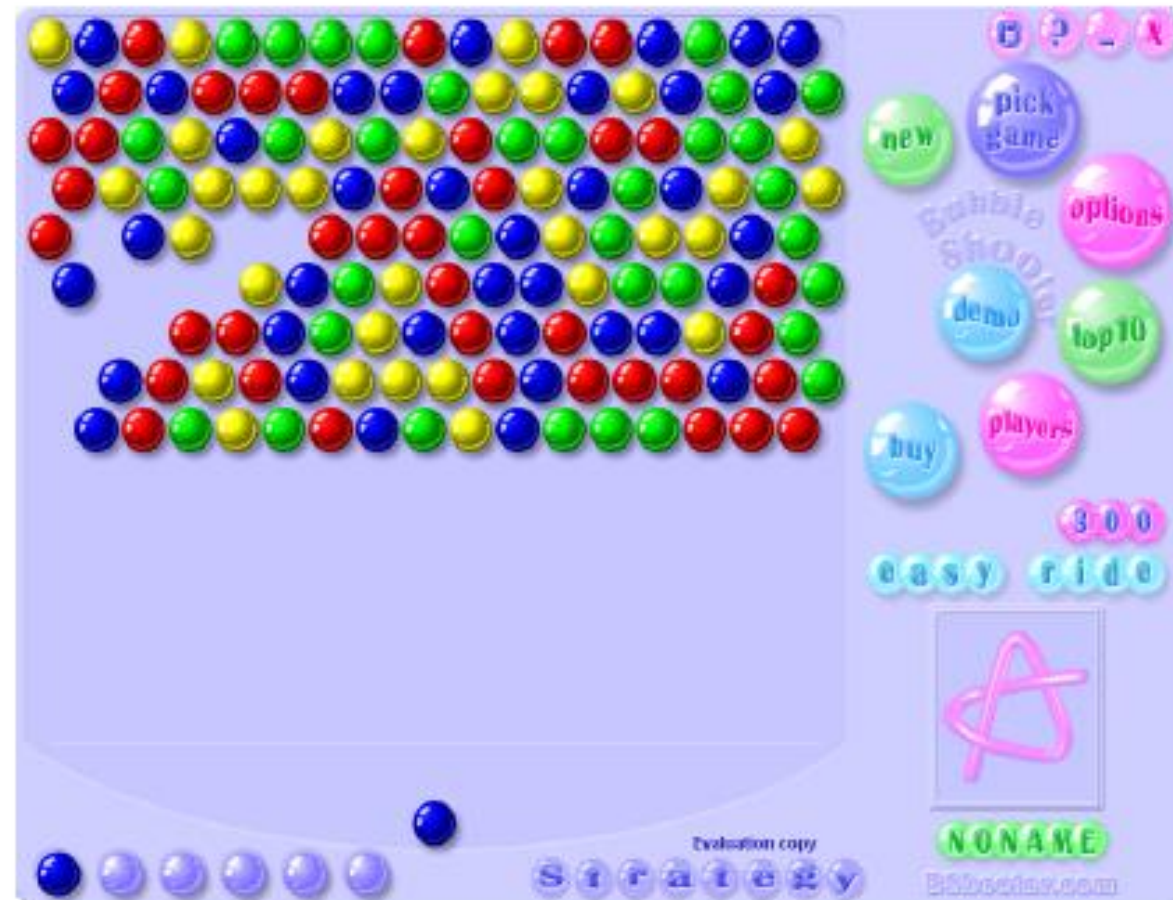
Princípios de Tog

□ 3. Princípio da **Color Blindness**

- Sempre que se usa cor para transmitir informação, devem-se também usar outras pistas alternativas para aqueles com problemas de identificação de cores
- Cerca de 10% dos homens têm problemas de daltonismo
- Usar alternativas às cores
 - Exemplo: símbolos diferentes

Princípios de Tog

□ Color Blindness (Jogo Bubble Shooter)



Princípios de Tog

□ Color Blindness (Bubble Bomb)



Princípios de Tog

□ 4. Princípio da **consistência**

- No padrão gráfico
- Nos *layouts*
- Na hierarquia da informação
- No significado semântico
- Na terminologia

Princípios de Tog

- 6. Princípio da **Eficiência do utilizador**
 - O que interessa é a produtividade do utilizador não a da máquina.
 - Manter o utilizador ocupado.
 - Identificar a ação base comum a um conjunto de tarefas
 - Opções disponíveis
 - Insert page break
 - Add Footnote
 - Insert Table of Contents
 - Opções disponíveis, de acordo com este princípio
 - Insert:
 - Page break
 - Footnote
 - Table of contents

Princípios de Tog

- 7. Princípio das **Interfaces exploráveis**
 - “Give users well-marked roads and landmarks, then let them shift into fourwheel drive.”
 - Proporcionar ao utilizador “deixas” perceptuais que dêem uma sensação de familiaridade
 - Tornar as ações reversíveis
 - Exemplo: possibilitar sempre Undo/Redo
 - Permitir sempre uma saída

Princípios de Tog

□ 8. Princípio da **Lei de Fitts**

- Tempo para atingir um alvo no ecrã/tempo para selecionar uma opção depende da distância da posição atual e da dimensão do alvo
- Aumentar as dimensões do alvo
 - Usar objetos grandes para funções importantes (botões grandes são mais rápidos)
- Reduzir a distância
 - Colocar os componentes importantes em sítios mais acessíveis

Princípios de Tog

- 10. Princípio da **Redução da Latência**
 - ▣ Sempre que possível, usar *multi-threading* para desviar a latência para *background*.
 - ▣ Mostrar uma ampulheta para qualquer ação que demore de 1/2 a 2 segundos
 - ▣ Animar a ampulheta para mostrar que o sistema não “morreu”
 - ▣ Mostrar uma mensagem indicativa do eventual tempo de espera para qualquer ação que demore mais de 2 segundos
 - ▣ Comunicar o tempo de espera através de um indicador de progresso animado
 - ▣ Assinalar fim de operação (*beep*)

Princípios de Tog

- 10. Princípio da **Redução da latência**
 - ▣ *Feedback* visual ou sonoro dos botões em 50 ms
 - ▣ Conhecimento de todos os clicks de botões através de feedback visual ou auditivo
 - ▣ Proporcionar mensagens que mantenham o utilizador entretido e informado enquanto espera por longos processos
 - ▣ Emitir um som e dar uma forte indicação visual em resultado do retorno de um processo longo (> 10 segundos), de modo a que os utilizadores saibam quando podem voltar a usar o sistema.

Princípios de Tog

□ 12. Princípio das **metáforas**

- Escolher boas metáforas que permitam ao utilizador compreender instantaneamente os mais pequenos detalhes do modelo conceptual.
- Escolher metáforas que facilitem a criação de imagens mentais.
- Escolher metáforas enérgicas que apelem aos sentidos – visão, audição, tácto e cinestésico – e que desencadeiem memórias.
- Usar metáforas apenas quando forem óbvias e claramente instrutivas

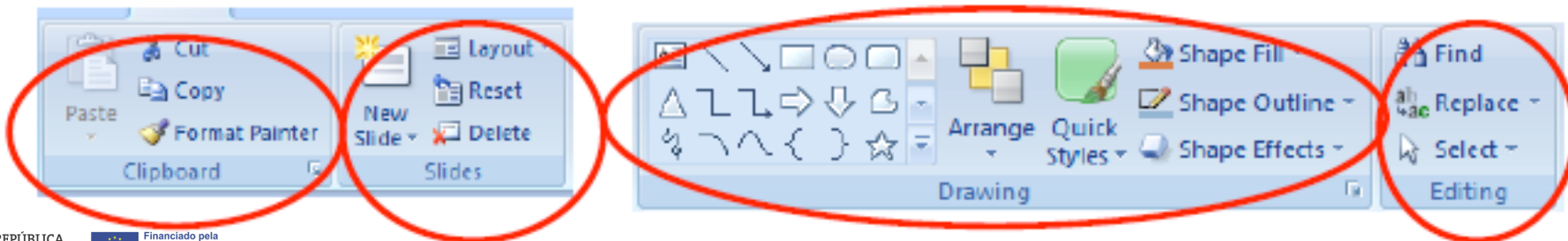
Princípios de Tog

- 17. Princípio da **Estrutura**
 - ▣ Organização com significado, baseado em modelos consistentes e claros, que são reconhecíveis e aparentes

Princípios de Tog

□ 17. Princípio da **Estrutura**

- Objectos relacionáveis devem ficar próximos (ideia de grupo) ou associados ou com aspecto semelhante
- Objectos não relacionáveis devem ficar separados ou com aspeto diferenciado



Regras de ouro (Shneiderman)

- 1. Lutar pela **Consistência (H4)**
- 2. Permitir aos Utilizadores experientes a utilização de **Atalhos (H7)**
- 3. Utilizar **Feedback** Informativo **(H1)**
- 4. Desenhar as Caixas de Diálogo Fechadas
- 5. Utilizar **Prevenção e Tratamento de Erros (H5)**
- 6. Permitir a fácil **Reposição** de **Ações**
- 7. Suportar a Localização Interna de **Controlo**
- 8. **Reduzir** a Carga sobre a memória **STM**