

Avaliação heurística

Ártus Bolzanni

Avaliação por pares

- Uma das ferramentas mais valiosas é a crítica por pares.
- Ela tem grande valor antes dos testes com usuários, pois evita gastar recursos com usuários para identificar erros simples.
- Quando bem estruturada, a crítica pode servir de base para convencer partes interessadas de que algo está errado e necessita ser mudado.

Avaliação heurística

- A avaliação heurística é uma técnica para encontrar problemas de usabilidade.
- Permite receber *feedback* muito rapidamente, a baixo custo e com retorno garantido.

Avaliação heurística

- O funcionamento da avaliação heurística se resume a reunir um grupo de pessoas, geralmente ligadas ao projeto, e entregá-las um conjunto de heurísticas.
- Com as heurísticas em mãos, o grupo procurará, individualmente, por erros no projeto.

Definição

(Heurística) - uma regra que pode ajudar a resolver um dado problema tomando como base o conhecimento sobre a natureza do problema.

Avaliação heurística

- 1 treinamento pré-avaliação;
- 2 avaliação;
- 3 atribuição de graus de severidade:
 - 1 individualmente;
 - 2 em grupo;
- 4 criar um relatório consolidado.

Heurísticas de Nielsen

São 10 as heurísticas de Nielsen:

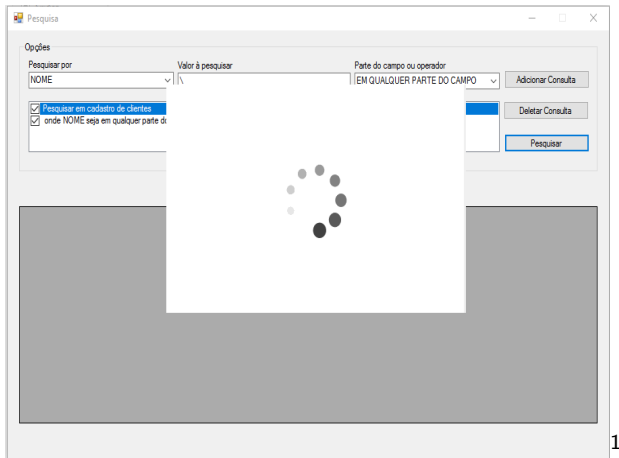
- 1 visibilidade do estado do sistema;
- 2 compatibilidade entre o sistema e o mundo real;
- 3 controle e liberdade para o usuário;
- 4 consistência e padronização;
- 5 prevenção de erros;
- 6 reconhecimento em vez de memorização;
- 7 eficiência e flexibilidade de uso;
- 8 estética e design minimalista;
- 9 ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros;
- 10 ajuda e documentação.

Visibilidade do estado do sistema

- É quase mandatório que o usuário esteja sempre a par do estado do sistema.
- Isso pode ser garantido por elementos de interface pequenos, mas que auxiliam bastante o usuário.
 - a mudança do símbolo de *play* para o *pause* em aplicativos de músicas;
 - tempo estimado para o fim de um *download*;
 - barra indicando quanto do espaço de armazenamento foi utilizado;
 - a seleção da faixa que está tocando.

Visibilidade do estado do sistema

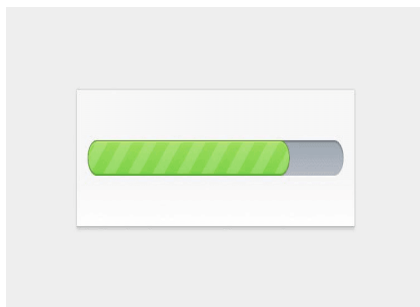
Ampulheta ou roda para indicar que uma ação está sendo realizada.



¹Retirado de: social.msdn.microsoft.com

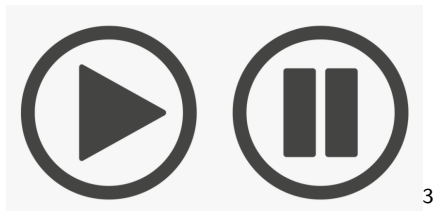
Visibilidade do estado do sistema

Barra de progresso para indicar o quanto de um *download* ou instalação de um *software* foi feito até então.



Visibilidade do estado do sistema

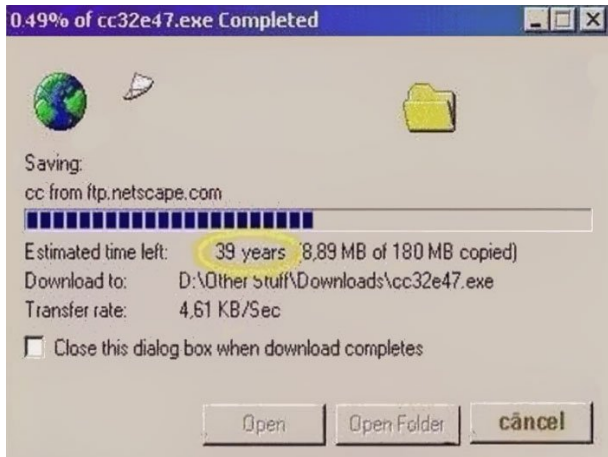
Os ícones de tocar e pausar em uma tocador de música.



³Retirado de: www.kindpng.com

Visibilidade do estado do sistema

O tempo estimado para o fim de um *download*.

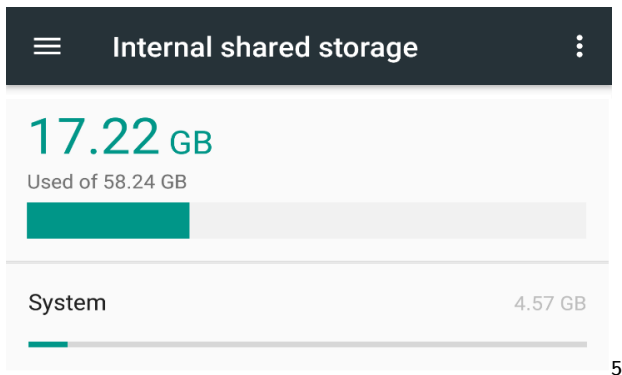


4

⁴Retirado de: twitter.com

Visibilidade do estado do sistema

Uma barra indicando o quanto de espaço de memória está utilizado.



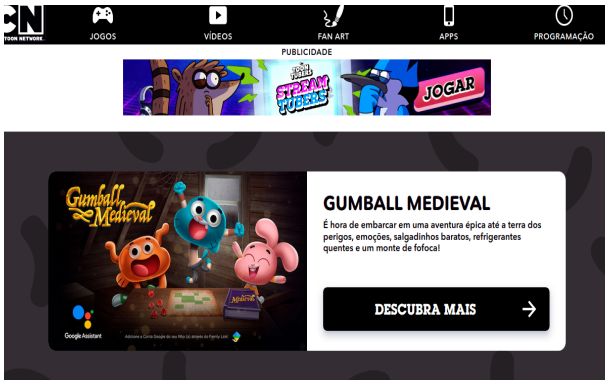
⁵Retirado de: androidpolice.com

Compatibilidade entre o sistema e o mundo real

- O sistema deve fornecer uma linguagem familiar ao contexto do usuário.
- Outro exemplo é

Compatibilidade entre o sistema e o mundo real

Um *site* voltado a crianças deve ter linguagem compatível com esse público por exemplo.



TOON NETWORK JOGOS VIDEOS FAN ART APPS PROGRAMAÇÃO

PUBLICIDADE

STREAM TUBERS JOGAR

GUMBALL MEDIEVAL

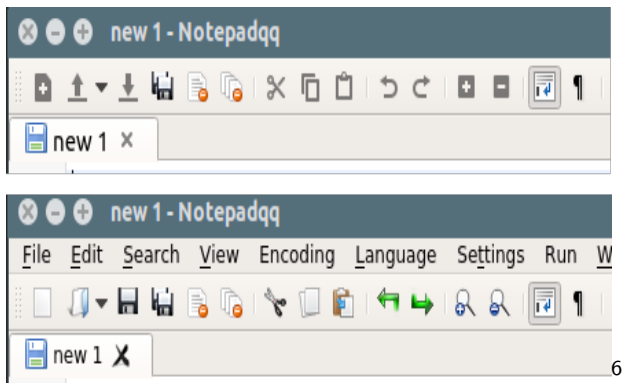
É hora de embarcar em uma aventura épica até a terra dos perigos, emoções, salgadinhos baratos, refrigerantes quentes e um monte de fofoca!

DESCUBRA MAIS →

Google Assistant

Compatibilidade entre o sistema e o mundo real

Deve-se utilizar ícones que possibilitem ao usuário ter uma noção intuitiva do que um botão faz.



⁶Retirado de: superuser.com

Controle e liberdade para o usuário


- Trata da capacidade do usuário poder desfazer uma ação ou não ser forçado a seguir um único caminho.
- Três exemplos clássicos desta heurística:
 - a lixeira presente na maioria dos SOs modernos;
 - a lixeira presente nos sistemas de e-mails;
 - a função desfazer (Ctrl+z).

Controle e liberdade para o usuário

- Outro exemplo é poder explorar um *site* de *e-commerce* sem obrigatoriamente ter de fazer *login*.
- O *site* da Amazon é um bom exemplo de liberdade para o usuário, pois permite que o usuário remova itens do carrinho durante o *checkout*.


Controle e liberdade para o usuário

Sites de passagens aéreas que mostram os valores das passagens nos dias anteriores e posteriores aos selecionados também são bom exemplo de liberdade para o usuário.

Selecione seu **voo de ida** 

Seg 05 OUT R\$ 194,03	Ter 06 OUT R\$ 194,03	Qua 07 OUT R\$ 194,03	Qui 08 OUT R\$ 194,03	Sex 09 OUT R\$ 194,03	Sáb 10 OUT R\$ 194,03	Dom 11 OUT R\$ 207,03	Mais datas e valores
--------------------------	--------------------------	--------------------------	----------------------------------	--------------------------	--------------------------	--------------------------	----------------------

Dinheiro Pontos

Organizar por: Preço 

13:55 SSA > 16:15 GRU	2h 20min	Direto	Ida a partir de R\$ 194,03
11:10 SSA > 13:35 GRU	2h 25min	Direto	Ida a partir de R\$ 194,03

Controle e liberdade para o usuário

- Em alguns casos, será interessante restringir a liberdade do usuário.
- Por exemplo, adicionar um *wizard* para instalação e configuração de impressora.

Consistência e padronização

- O sistema deve possuir uma padronização léxica e gráfica.
- **Apenas um ícone ou um termo** devem indicar uma funcionalidade.
- Um ícone ou um termo devem indicar **apenas uma funcionalidade**.
- A disposição e o alinhamento de ícones, botões e textos também deve ser consistente.

Prevenção de erros

- A *interface* deve ser projetada para que o usuário cometa o menor número de erros possíveis.

Prevenção de erros

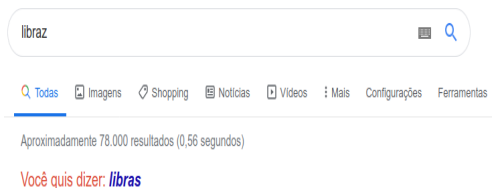
Inserir máscaras, dicas marginais e caixas de confirmação buscam garantir esta característica.

The image shows a screenshot of a web application interface. At the top, there is a dark navigation bar with the text 'Videolgniter' on the left, a search bar with the placeholder 'Pesquisar...', and links for 'Configurações' and 'Perfil' on the right. Below the navigation bar is a sidebar with a blue header 'Tela inicial' and a list item 'Usuários'. The main content area is titled 'Novo Usuário' and contains a form with the following fields: 'Nome:' with a text input field containing the placeholder 'Informe o nome...'; 'CPF:' with a text input field containing 'Informe o cpf...'; 'Endereço:' with a text input field containing 'Informe o endereço...'; 'Nível:' with a dropdown menu showing '---'; 'Email:' with a text input field containing 'Informe o email...'; 'Senha:' with a text input field containing 'Informe a senha...'; and 'Status:' with a dropdown menu showing '---'. At the bottom right of the form are two buttons: a green 'Enviar' button and a white 'Cancelar' button. A small number '7' is visible in the bottom right corner of the screenshot area.

⁷Retirado de: <https://www.youtube.com/watch?v=9cAmYXv6Aig>

Prevenção de erros

Um exemplo interessante é o de sistemas que sugerem uma busca com ortografia parecida ou de sinônimos da que foi buscada.

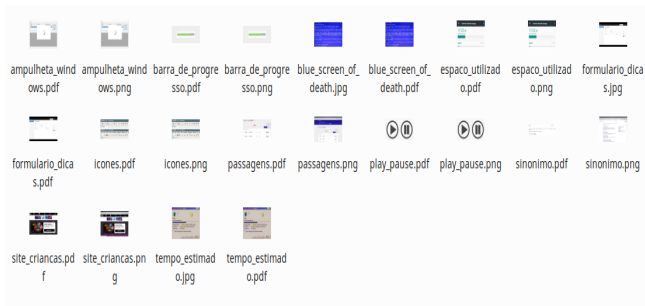


Reconhecimento ao invés de memorização

- Objetos, ações e opções devem ser visíveis ao usuário.
- O usuário não deve ter de lembrar da informação de uma parte do sistema em outra. Este deve estar sempre visível quando necessária.
- Sistemas que dependem de códigos são um mau exemplo.

Prevenção de erros

Um bom exemplo é o uso de miniaturas (*thumbnails*) ao invés de apenas os nomes dos arquivos.



Flexibilidade e eficiência de uso

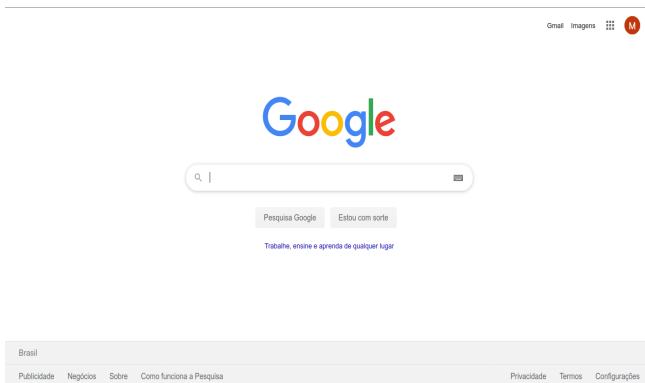
- A *interface* além de ser bastante visual para os usuários mais inexperientes, também deve possuir “aceleradores” para os usuários mais avançados.
- São exemplos desses “aceleradores”:
 - teclas de atalho;
 - boas opções padrão em formulários;
 - sistemas de recomendação.

Estética e *design* minimalistas

- Tanto a distribuição das cores quanto das informações não deve atrapalhar o usuário.
- Não se deve inserir funcionalidades ou elementos de *interface* que não serão utilizados.

Estética e *design* minimalistas

Um bom exemplo: google.com.



Estética e *design* minimalistas

Um mau exemplo: lingscars.com.



Recuperação de erros

As mensagens de erro devem ser claras e de fácil entendimento para o usuário.

```
to your computer.
SYSTEM_SERVICE_EXCEPTION

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:

*** STOP: 0x0000003B (0x00000000C0000005, 0xFFFFF97FFE8D1231, 0xFFFFFADF1
x0000000000000000)

*** ativvaxx.dll - Address FFFFF97FFE8D1231 base at FFFFF97FFE8B3000,
4e31d434

Beginning dump of physical memory
Physical memory dump complete.
```

8

Ajuda e documentação

- É importante que haja uma seção dedicada à ajuda e à documentação.
- Muitas vezes o usuário necessitará buscar mais detalhes sobre uma determinada funcionalidade, portanto, é importante que as seções de ajuda e documentação sejam bem arquitetadas e com linguagem simples e direta.

Avaliação heurística

- Caso seu sistema seja do tipo “vá-e-use”, como uma máquina de venda, é interessante dar o mínimo de informações sobre o sistema, de modo que o avaliador tenha uma experiência próxima ao usuário final.
- Já para um *software* especialista, deverá haver um treinamento similar ao que seria dado ao usuário final.

Avaliação heurística

- Há uma boa chance que avaliadores diferentes encontrem problemas diferentes.
- Os avaliadores só deverão conversar entre si após terem feito as avaliações individuais.
- Isso evita que um avaliador interfira nos demais.

Avaliação heurística

- Um ponto importante é a quantidade de avaliadores.
- A partir de uma certa quantidade de avaliadores, o número de problemas encontrados não aumentará.
- O número de avaliadores dependerá do tipo de *interface* que está sendo estudada. Geralmente, entre 3 e 5 avaliadores bastam.

Avaliação heurística

- Apesar da técnica de avaliação heurística ser bastante eficiente, os avaliadores podem gerar falsos positivos, ou seja, apontar erros que não aconteceriam em um ambiente real.
- Por isso, estimula-se mesclar os métodos de avaliação.

Avaliação heurística

- É interessante que cada avaliador passe por um cenário duas vezes.
- Na primeira, ele tomará conhecimento do cenário.
- Na segunda, poderá focar-se em elementos específicos.

Avaliação heurística

- Os avaliadores devem criar uma lista de problemas e relacioná-los às heurísticas utilizadas.
- Simplesmente dizer “eu não gosto disso” tem pouca serventia.

Avaliação heurística

- Após a listagem dos erros, os avaliadores atribuirão graus de severidade a cada erro.
- Isso é importante, pois permite que a equipe de desenvolvimento aloque recursos de maneira mais eficiente.

Avaliação heurística

O grau de severidade é resultado da combinação de:

- 1 frequência;
- 2 impacto;
- 3 persistência.

Classificação de severidade de erros

Jacob Nielsen também propôs um sistema de classificação de severidade:

- 0 - não concorda que há problema;
- 1 - problema cosmético;
- 2 - problema menor de usabilidade;
- 3 - problema maior de usabilidade (importante corrigir);
- 4 - catástrofe de usabilidade (imperativo corrigir).

Classificação de severidade de erros

Exemplo de grau de severidade:

- Problema: impossível editar endereço.
- Severidade: 2 (problema menor de usabilidade).
- Heurística violada: controle e liberdade do usuário.
- Descrição: após inserir o endereço pela primeira vez, não é possível editá-lo. Deste modo, caso o usuário errou a digitação do endereço ou tenha se mudado, não poderá alterá-lo no sistema.