

Introdução

Avaliação de usabilidade pode ser entendida como o procedimento para aquisição de informação sobre a usabilidade ou potencial usabilidade de um sistema a fim de tanto aprimorar recursos numa interface em desenvolvimento e seu material de suporte quanto avaliar uma interface já finalizada.

Tipos de avaliação

A *avaliação formativa* acontece antes da implementação e tem participação na formação do sistema, com influência sobre as características do produto em desenvolvimento, também é conhecida como de desenvolvimento.

A *avaliação somativa* acontece após a implementação com o objetivo de testar o funcionamento apropriado do sistema final. É realizada quando se tem em vista alguma melhoria em um produto.

Protótipo e interface

O objetivo de uma interface é comunicar-se com o usuário. No entanto, o projetista e o usuário possuem conhecimentos e preocupações diferentes. Com isso, e é muito difícil para o designer prever como algumas decisões de projeto influenciarão o comportamento do usuário.

Um protótipo pode oferecer a oportunidade de ajustar o design ao usuário.

É relativamente fácil observar usuários trabalhando com um protótipo e entender os problemas que acontecem.

Protótipo

Um protótipo é uma representação da interface com o usuário, onde ele poderá interagir para propor mudanças e melhorias.

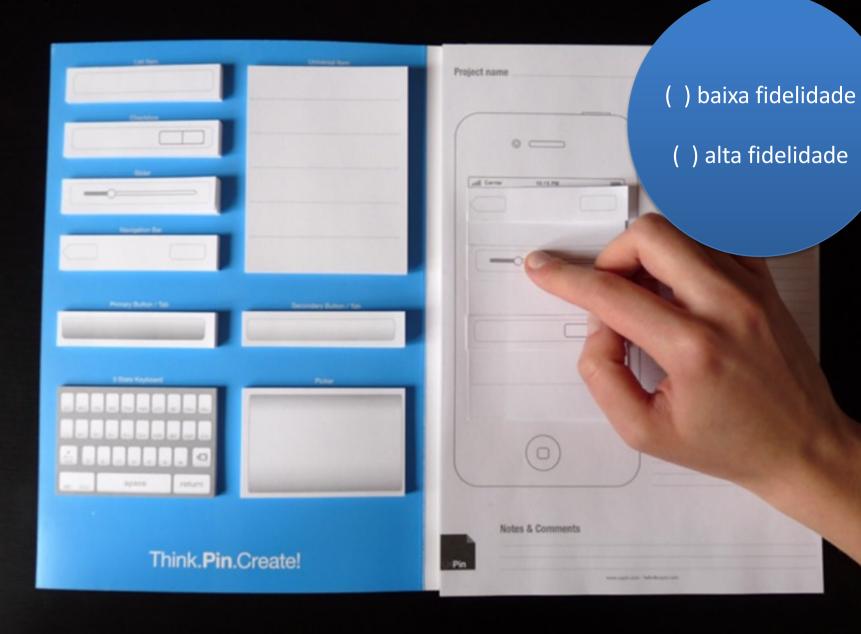
Pode ser:

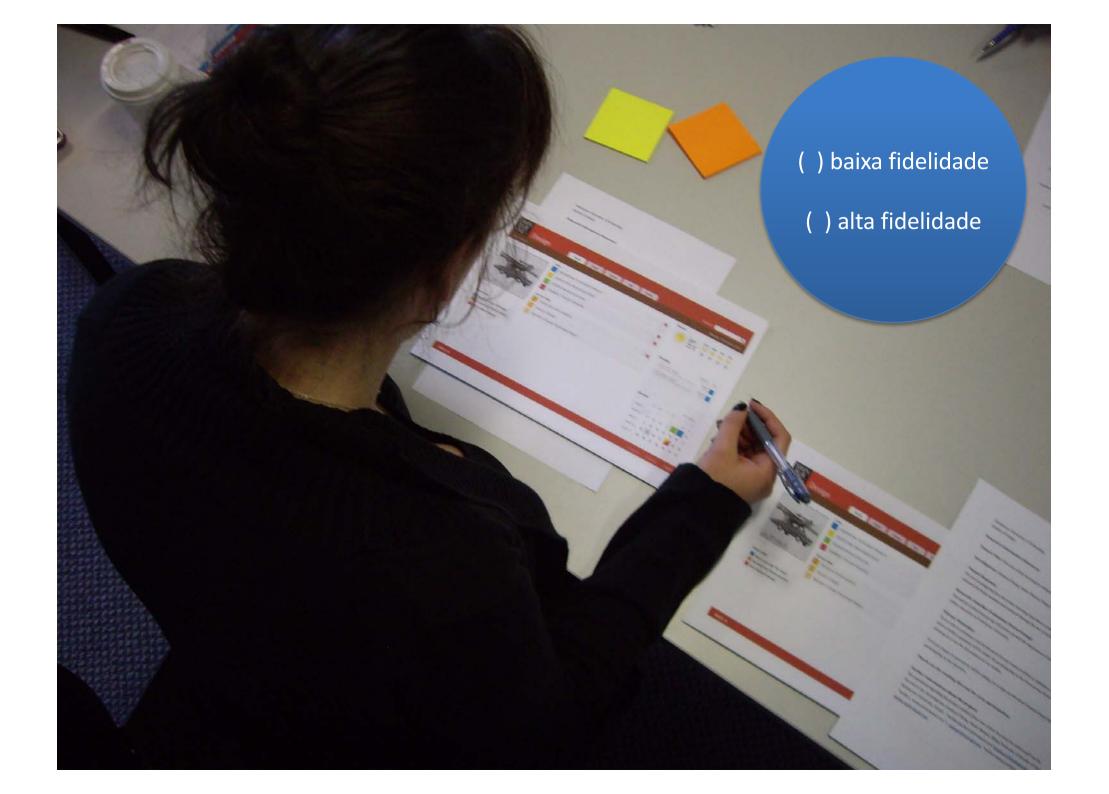
- •um pequeno guia para o usuário
- •uma simulação em papel
- •uma versão inicial do programa

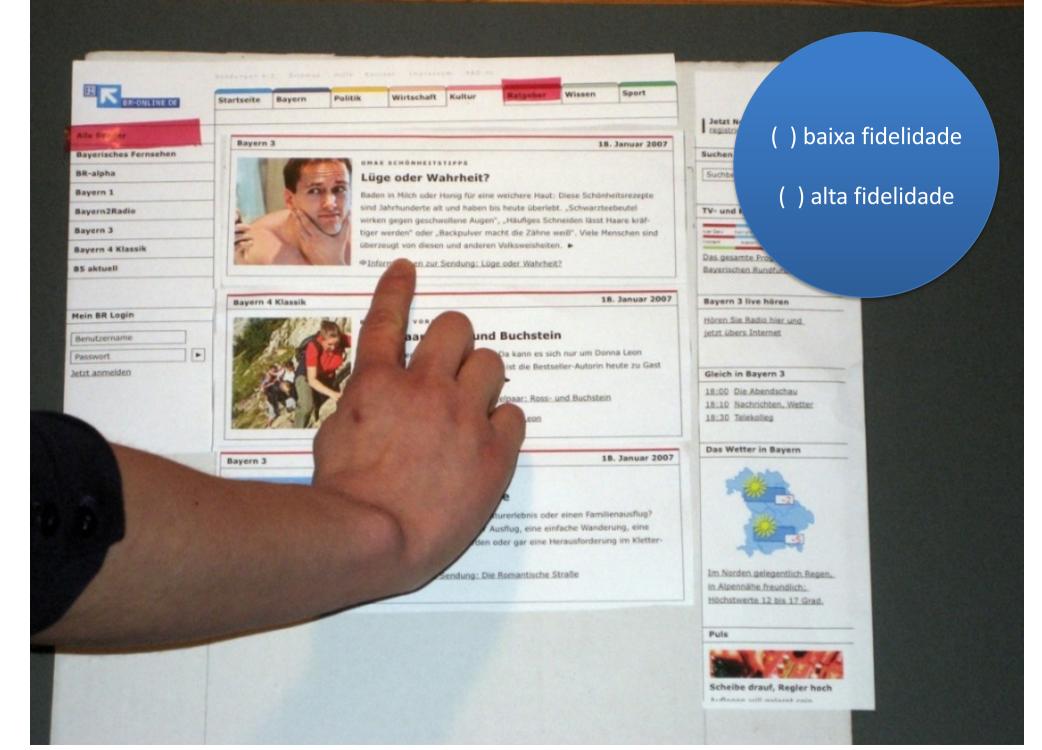
- •uma simulação usando o próprio computador a partir de uma ferramenta de prototipagem
- •o sistema a ser substituído

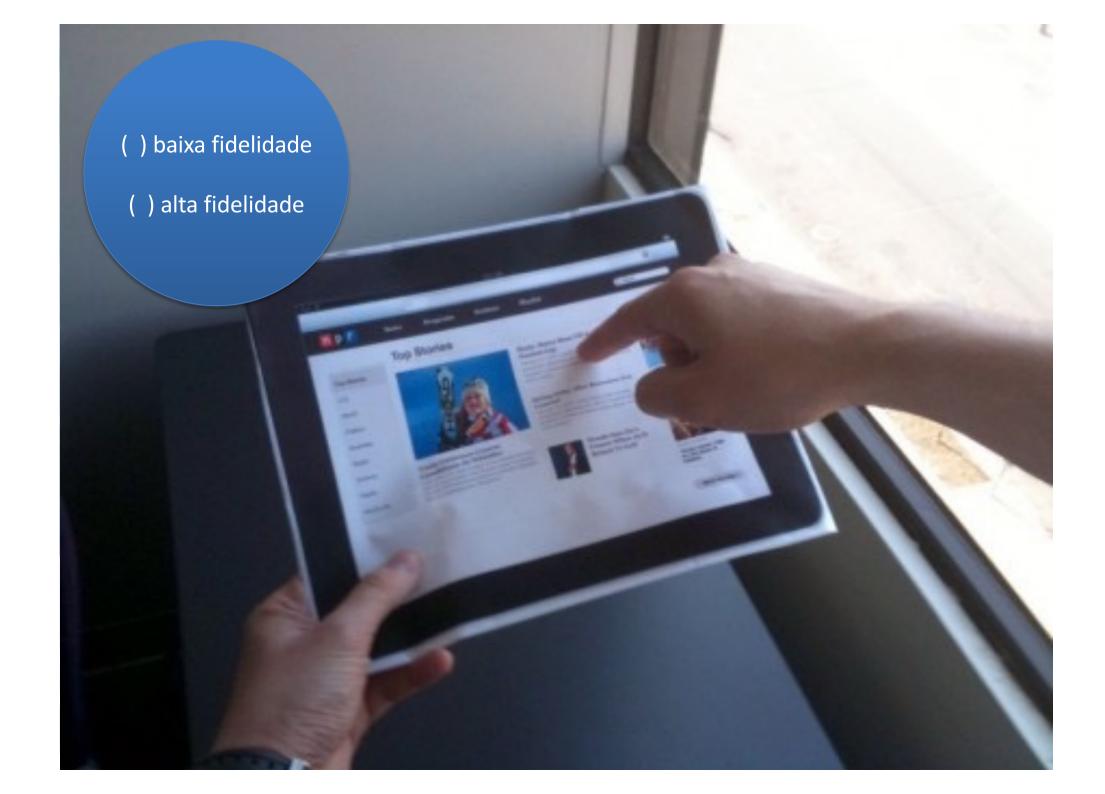
Protótipo de baixa x de alta fidelidade



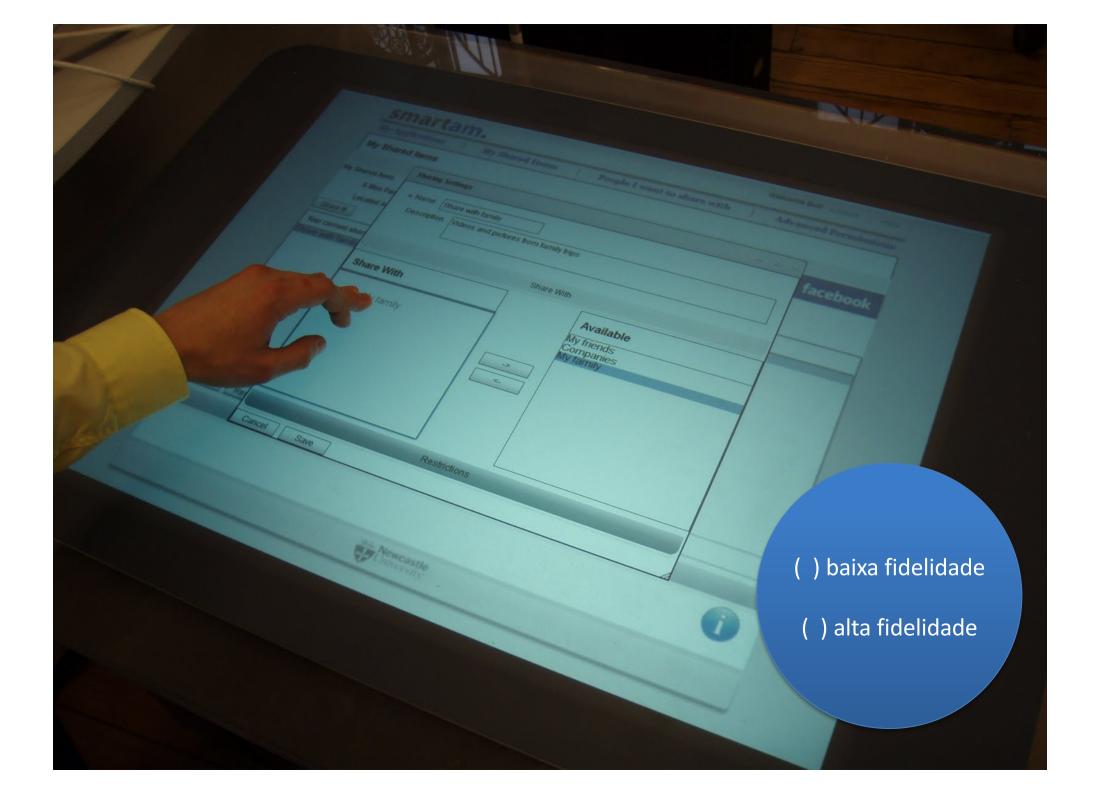












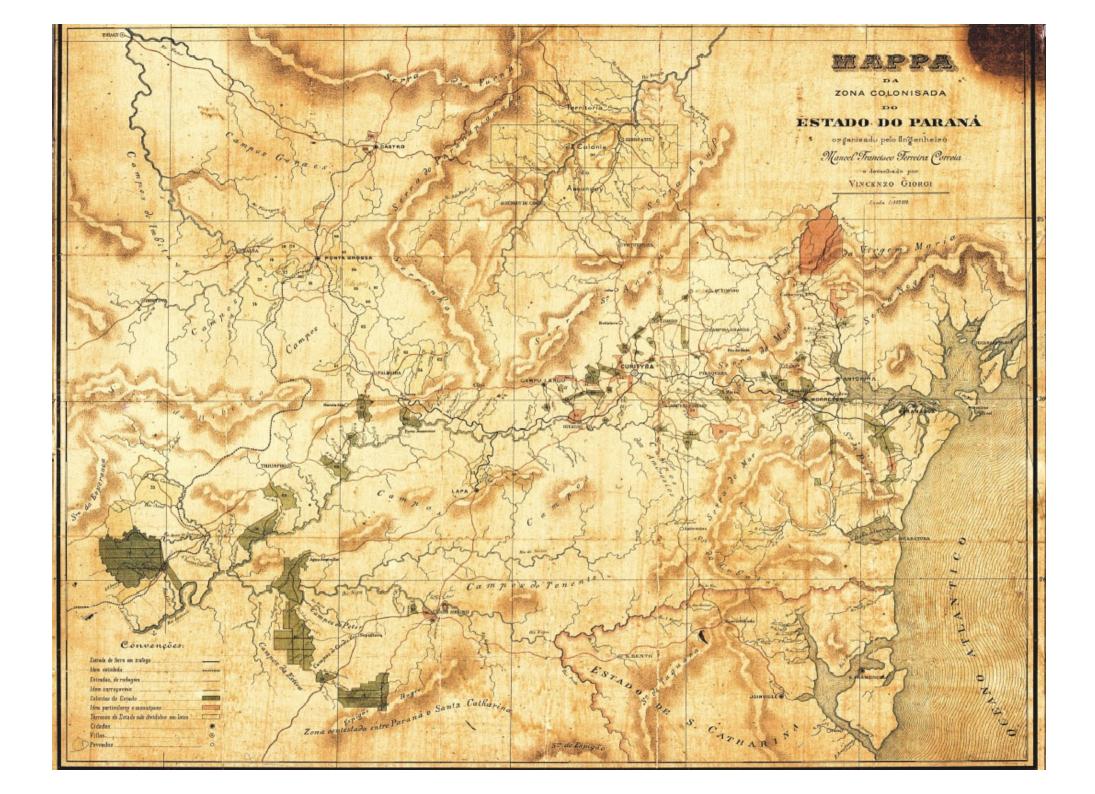
Nível de fidelidade do protótipo:

- 1. Representação da imagem
- 2. Interatividade do dispositivo

Avaliação Heurística

O termo "heurística" é originário dos estudos de História, referente à pesquisa e crítica de documentos para a descoberta de fatos.

Em educação, o termo é utilizado para definir a linha pedagógica que possibilita que os alunos adquiram conhecimento através da descoberta pessoal.





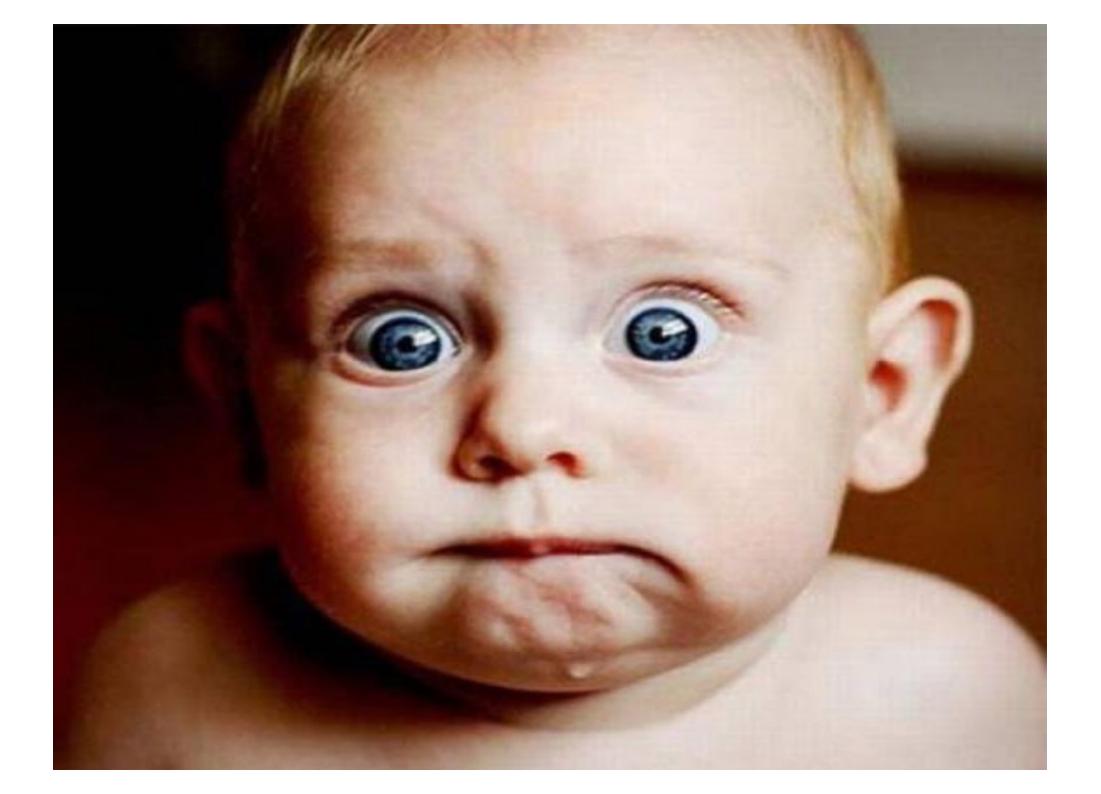
Avaliação Heurística

O termo "avaliação heurística" em IHC foi introduzido por Jakob Nielsen e Rolf Molich no início da década de 1990, quando propuseram um método através da qual o projetista aplica um número de princípios ou heurísticas ao projeto.

O termo cunhado por Nielsen e Molich descreve um método no qual um pequeno grupo de avaliadores examina uma dada interface e procura por problemas que violem alguns princípios gerais do bom projeto de interface.

Por ser uma ferramenta geral, a avaliação heurística pode ser aplicada em qualquer estágio do desenvolvimento do projeto, desde a prototipagem, como método formativo, até mesmo após a implementação, como método somativo.





Etapas

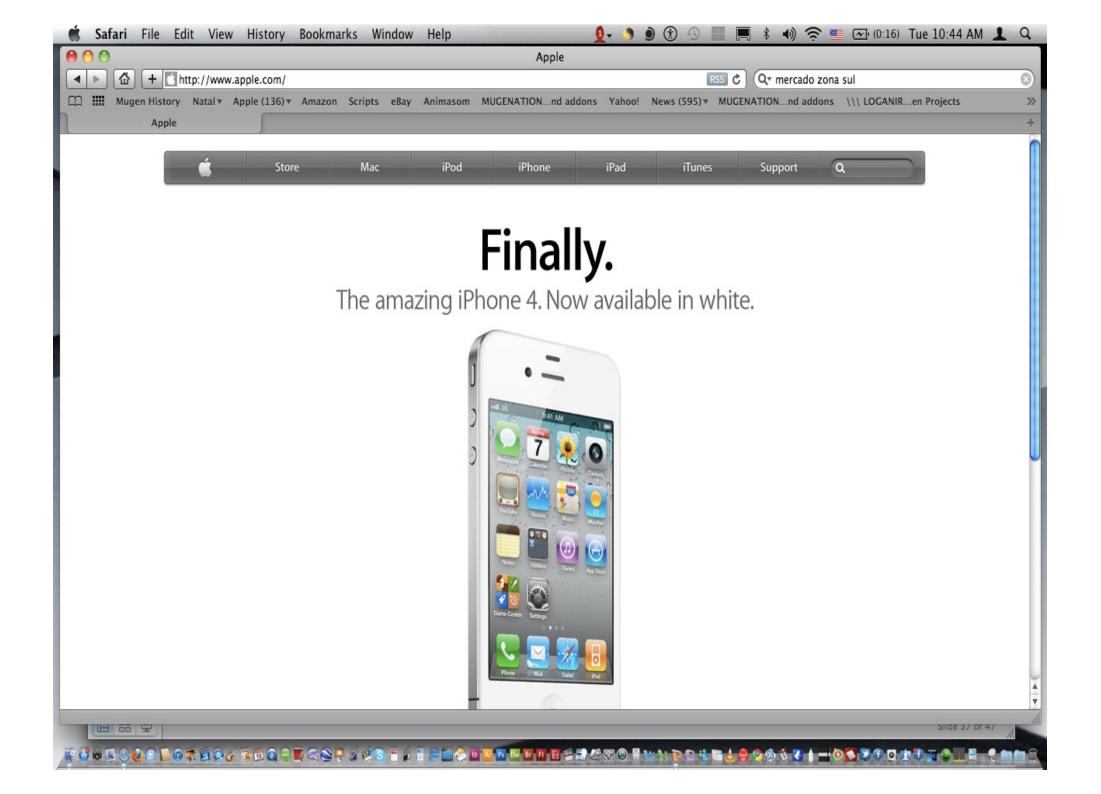
- 1) Reunir um grupo de especialistas para executar a avaliação
- 2) Pedir que os especialistas avaliem a interface isoladamente, posteriormente comparando-se os achados
- 3) Obter o *feedback* dos avaliadores
- 4) Atribuir níveis de gravidade aos problemas descobertos
- 5) Tabular dados

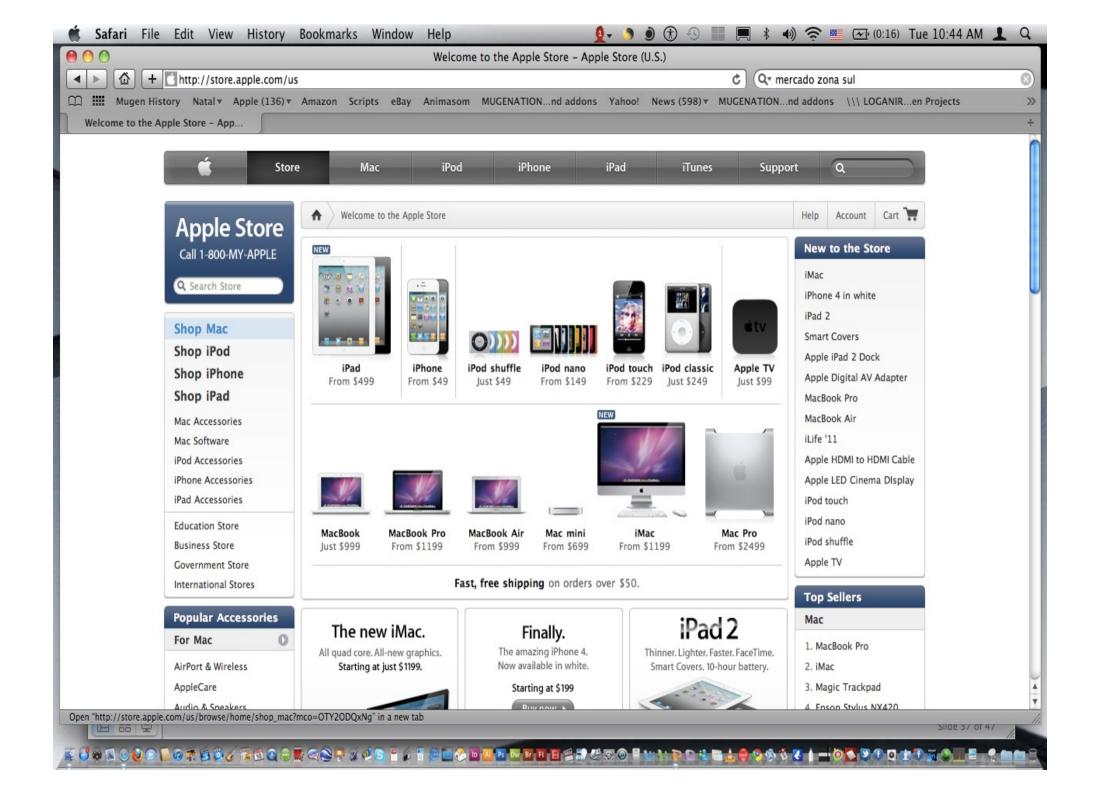
- 1. Visibilidade do status do sistema;
- 2. Equivalência entre o sistema e o mundo real;
- 3. Controle do usuário e liberdade;
- 4. Consistência e padrões;
- 5. Prevenção de erro;

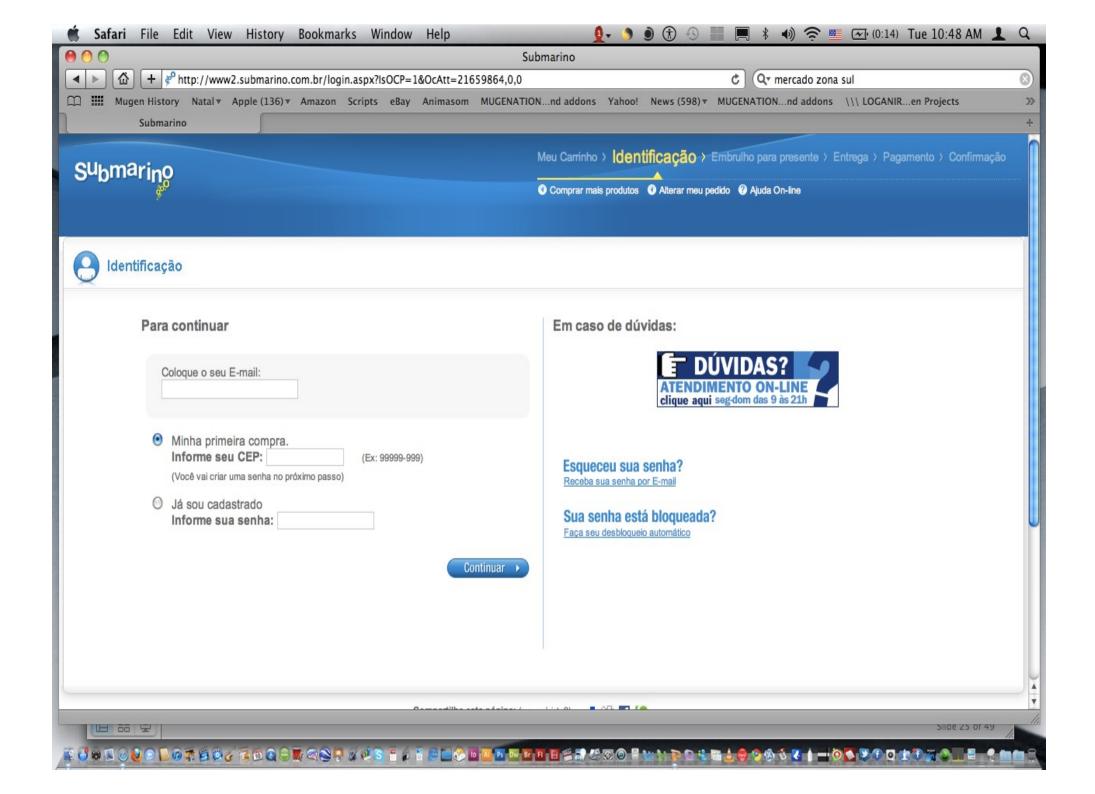
- 6. Reconhecer ao invés de relembrar;
- 7. Flexibilidade e eficiência de uso;
- 8. Estética e design mínimo;
- 9. Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas;
- 10. Ajuda e documentação.

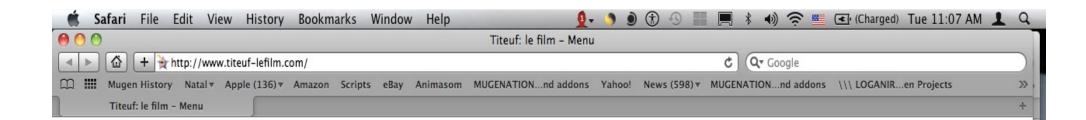
1. Visibilidade do status do sistema

O sistema deve sempre **manter o usuário informado** sobre o que está acontecendo, através de *feedback* apropriado em período de tempo razoável.









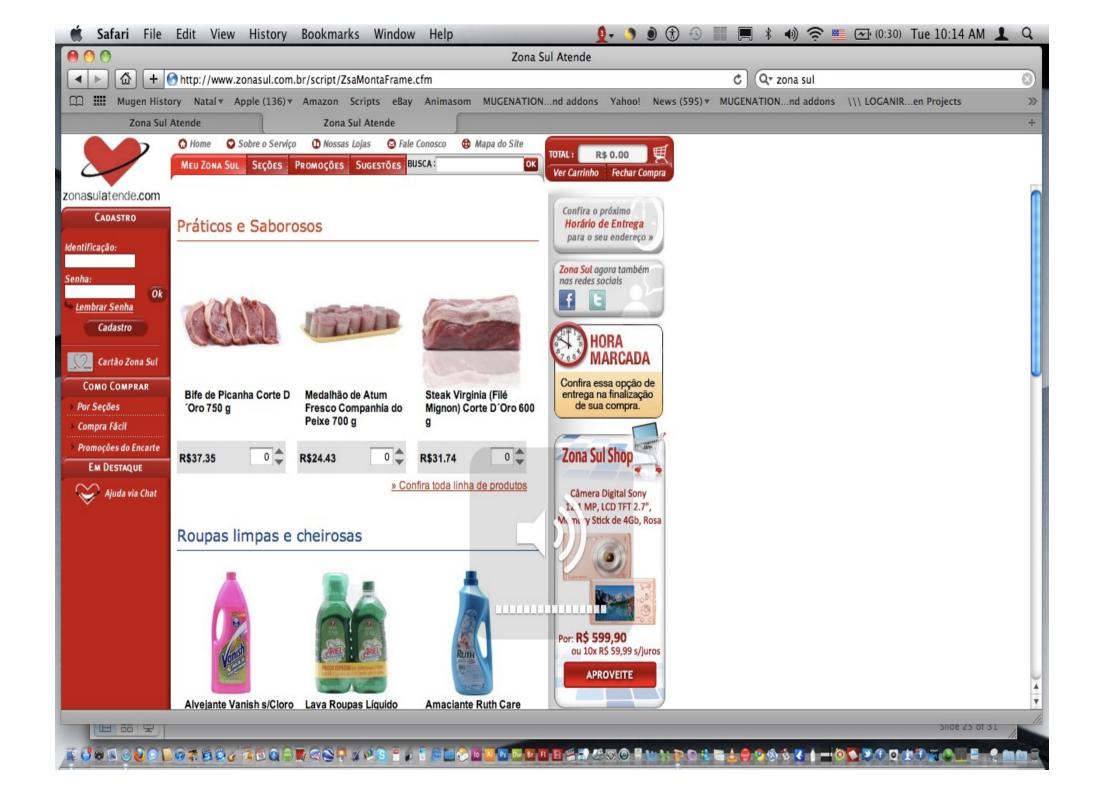


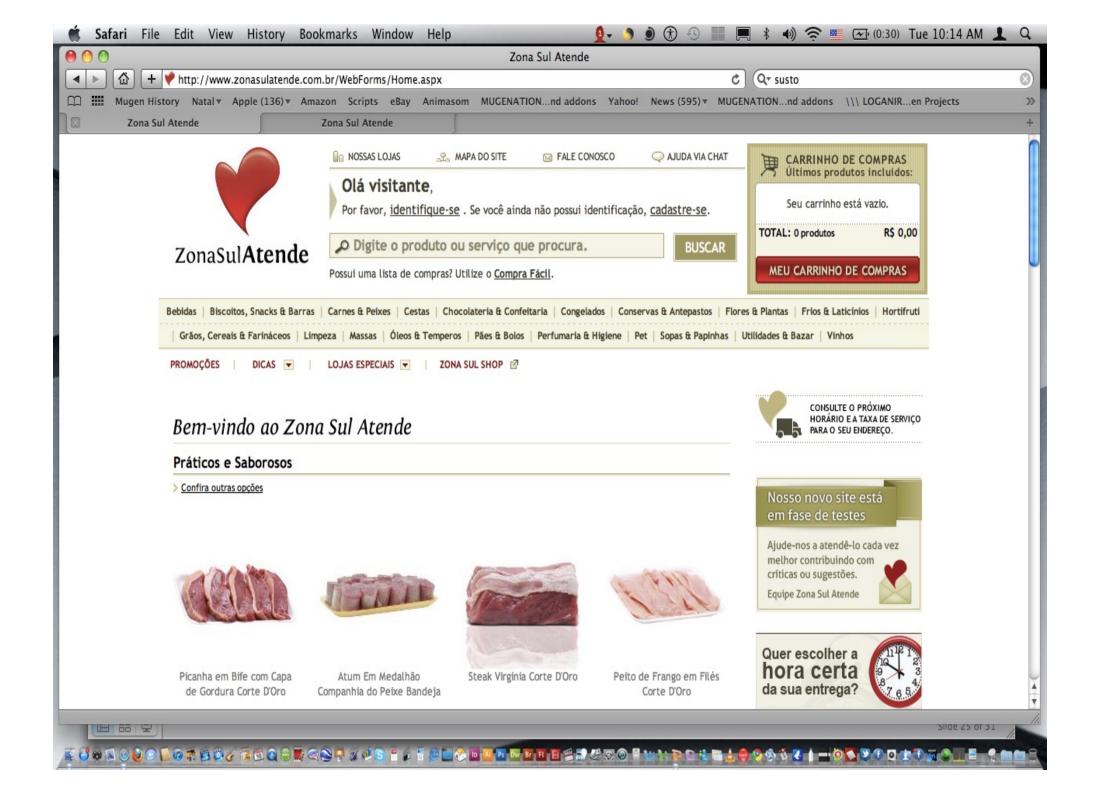
2. Equivalência entre o sistema e o mundo real

O sistema **deve falar a linguagem do usuário**, com palavras, frases e conceitos que lhes sejam familiares, ao invés de termos orientados ao sistema. Deve-se seguir convenções do mundo real, fazendo a informação aparecer em uma ordem natural e lógica.



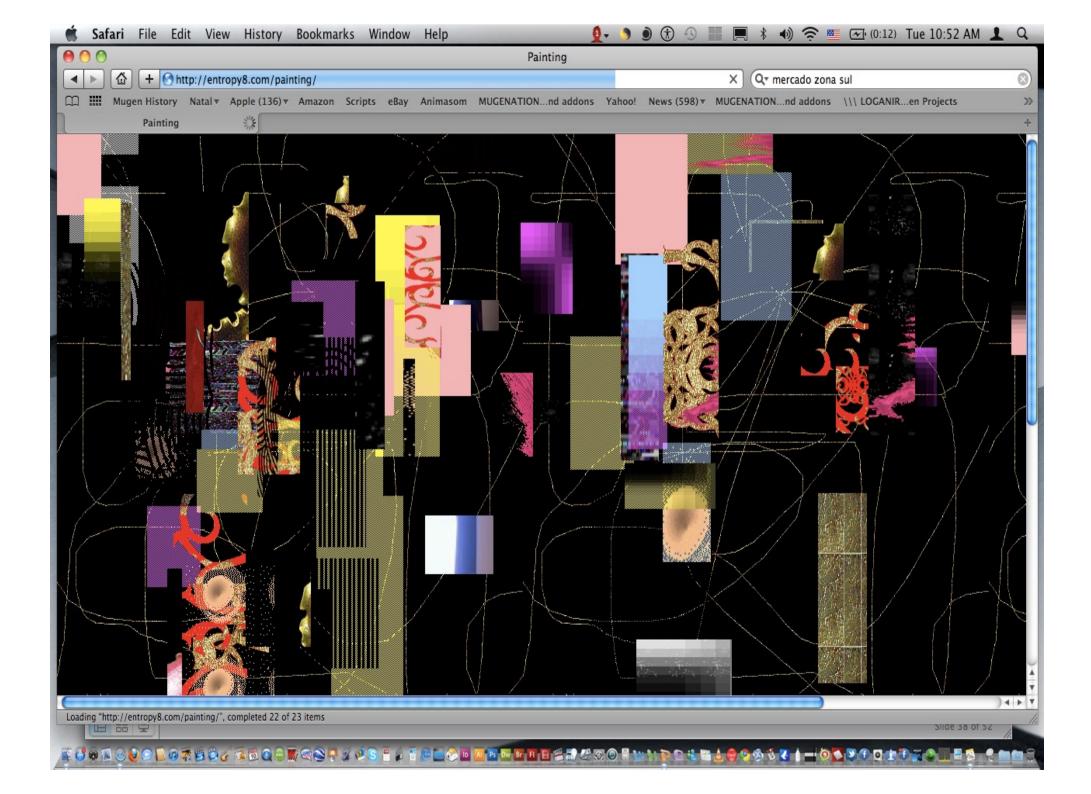






3. Controle do usuário e liberdade

Os usuários podem escolher funções do sistema por engano e precisarão de uma "saída de emergência" bem marcada para deixar o estado não desejado sem ter que passar por um extenso diálogo. Deve-se possibilitar que o usuário possa desfazer e refazer ações.



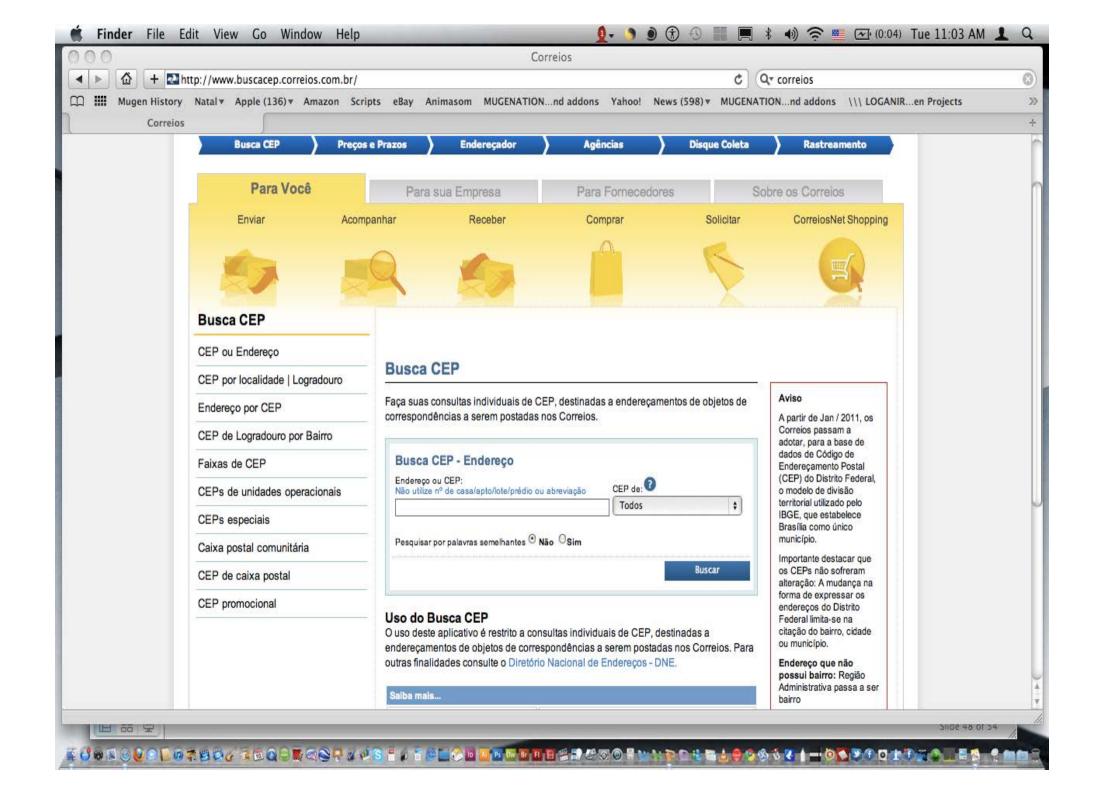
4. Consistência e padrões

Usuários não devem ter que imaginar se palavras, situações, ou ações diferentes significam a mesma coisa.

Deve-se seguir as convenções da plataforma.

5. Prevenção de erro

Muito melhor que boas mensagens de erro é um projeto cuidadoso que, em primeiro lugar, previna a ocorrência de problemas.



6. Reconhecer ao invés de relembrar

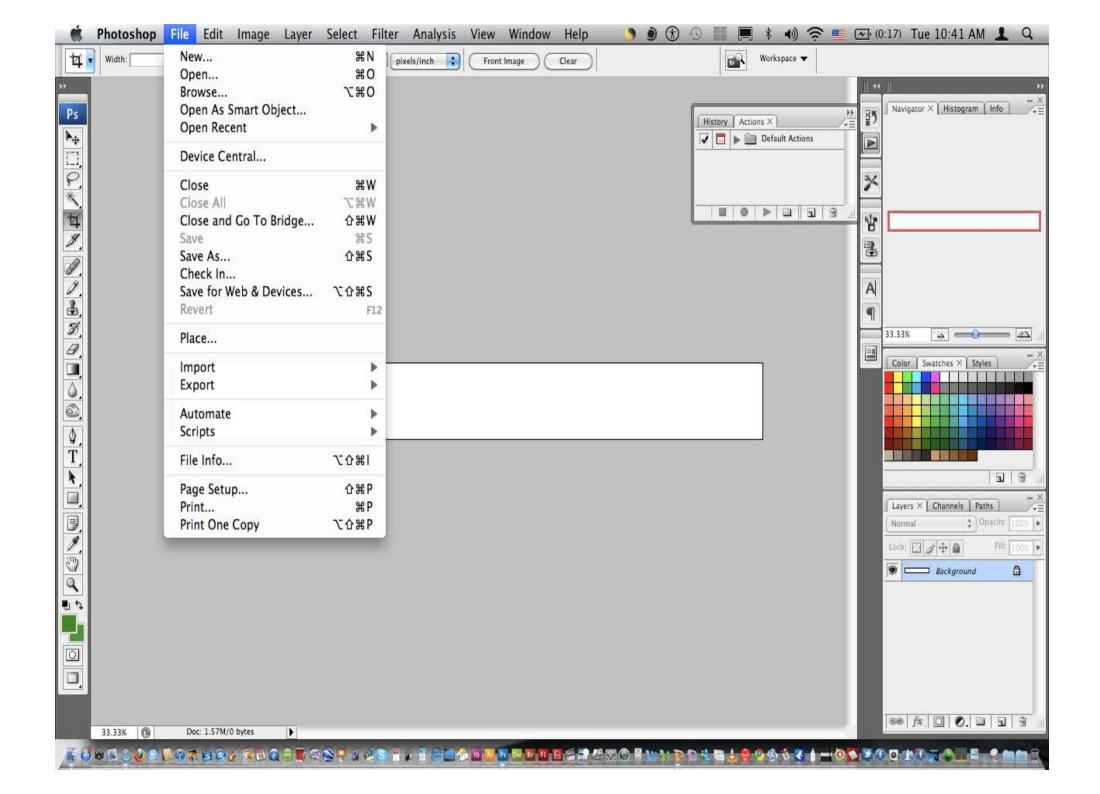
Deve-se tornar objetos, ações e opções visíveis.

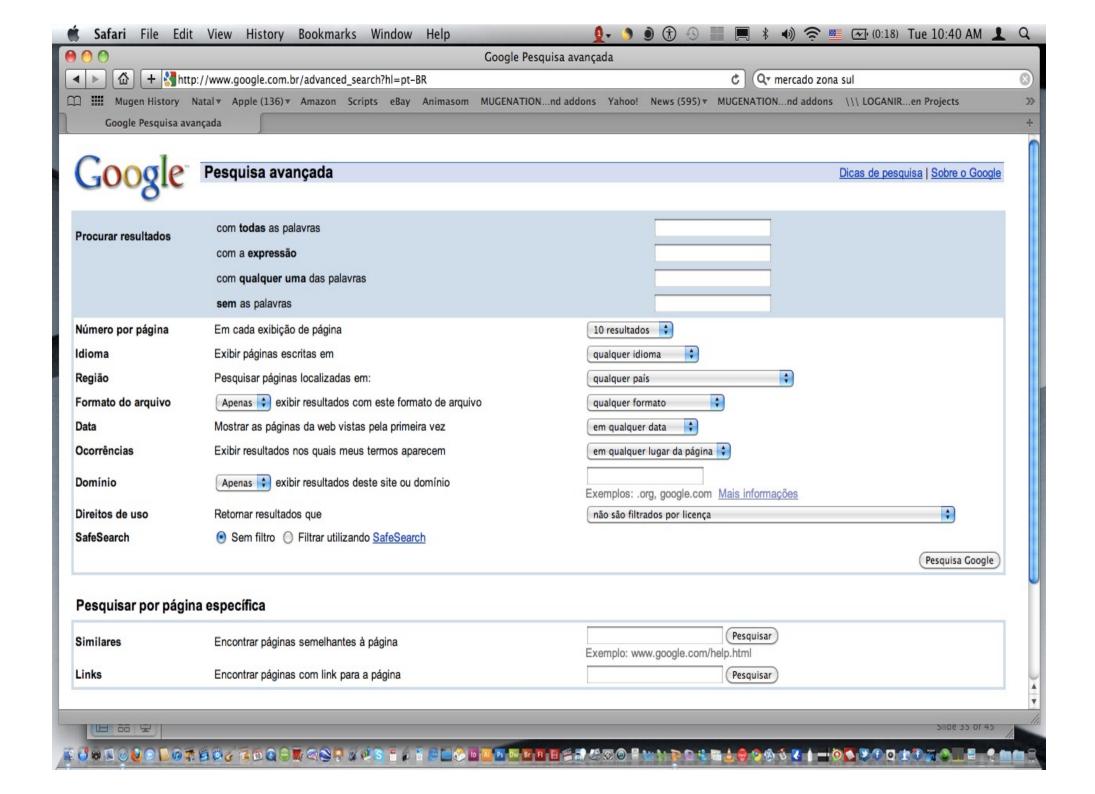
O usuário não deve ter que relembrar informação de uma parte do diálogo em outra parte. Instruções para uso do sistema devem estar visíveis ou facilmente recuperáveis sempre que necessário.



7. Flexibilidade e eficiência de uso

Aceleradores da tarefa – não vistos pelo novato – podem ainda aumentar a velocidade de interação para o usuário experiente de forma que o sistema possa atender tanto aos usuários experientes quanto aos inexperientes.

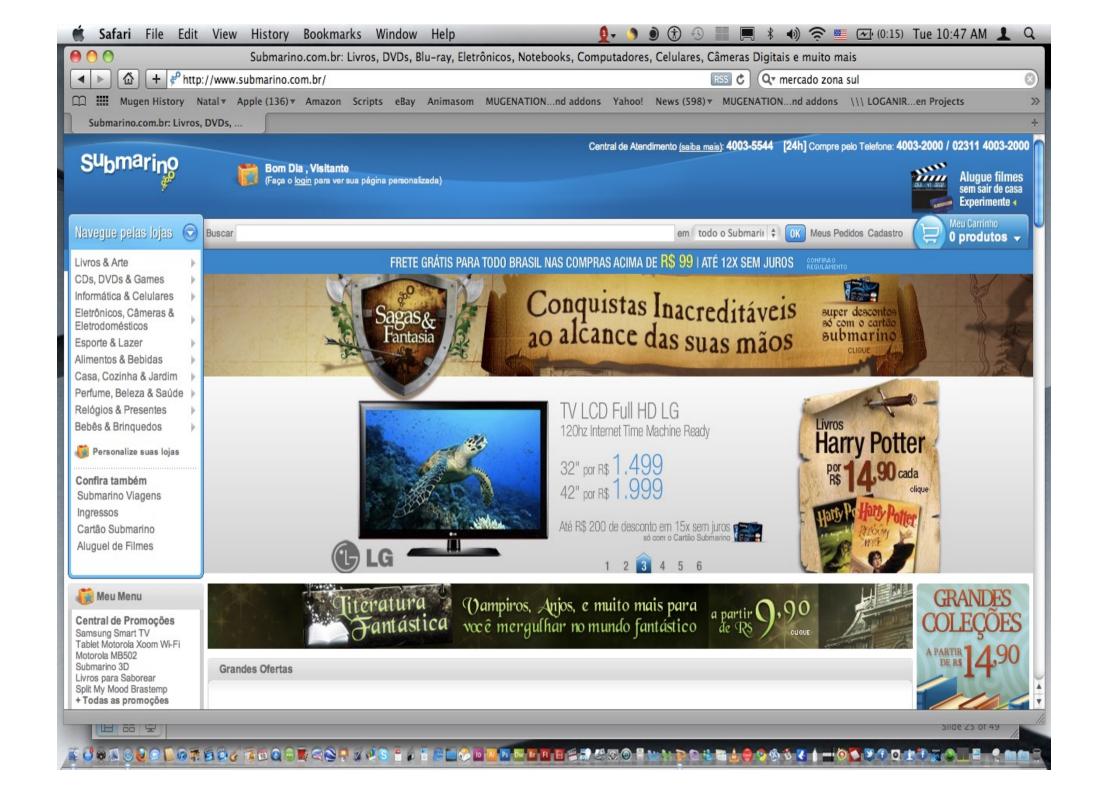


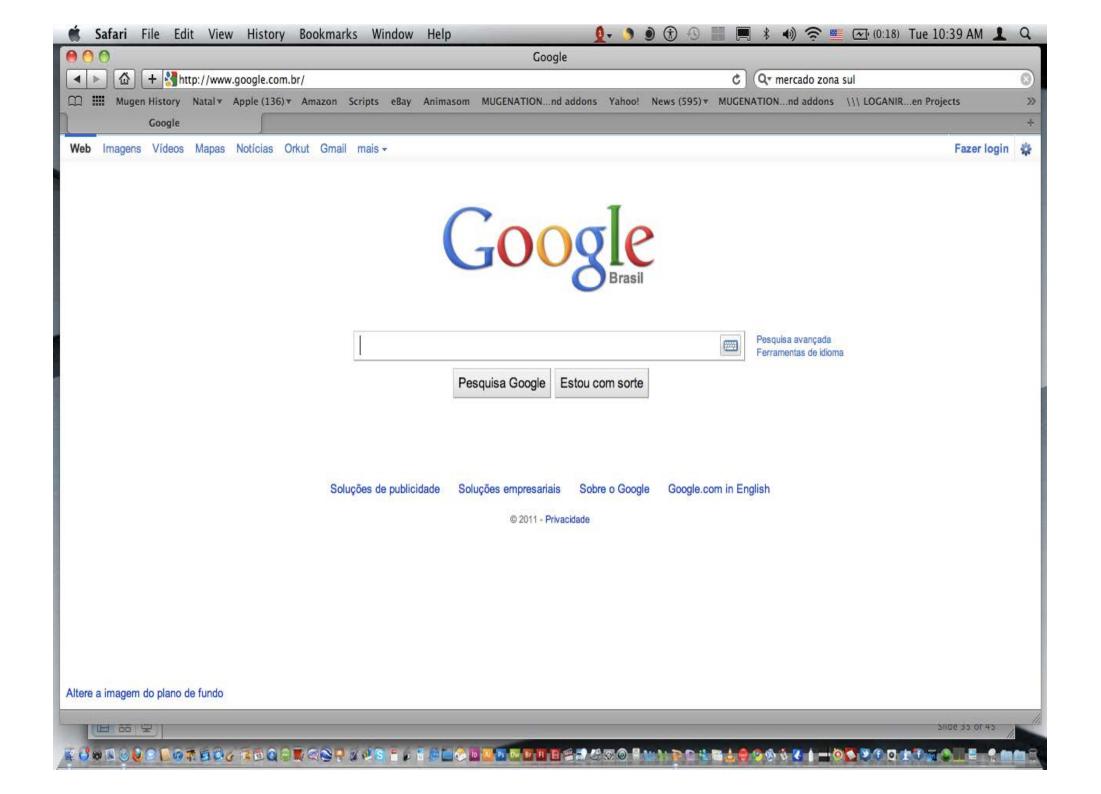


8. Estética e design minimalista

Diálogos não devem conter informação que seja irrelevante ou raramente necessária.

Toda unidade de informação extra em um diálogo compete com unidades de informação relevantes e diminui sua visibilidade relativa.





9. Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas

Mensagens de erro devem ser expressas em linguagem clara (sem códigos), indicar precisamente o problema, e sugerir construtivamente uma solução.





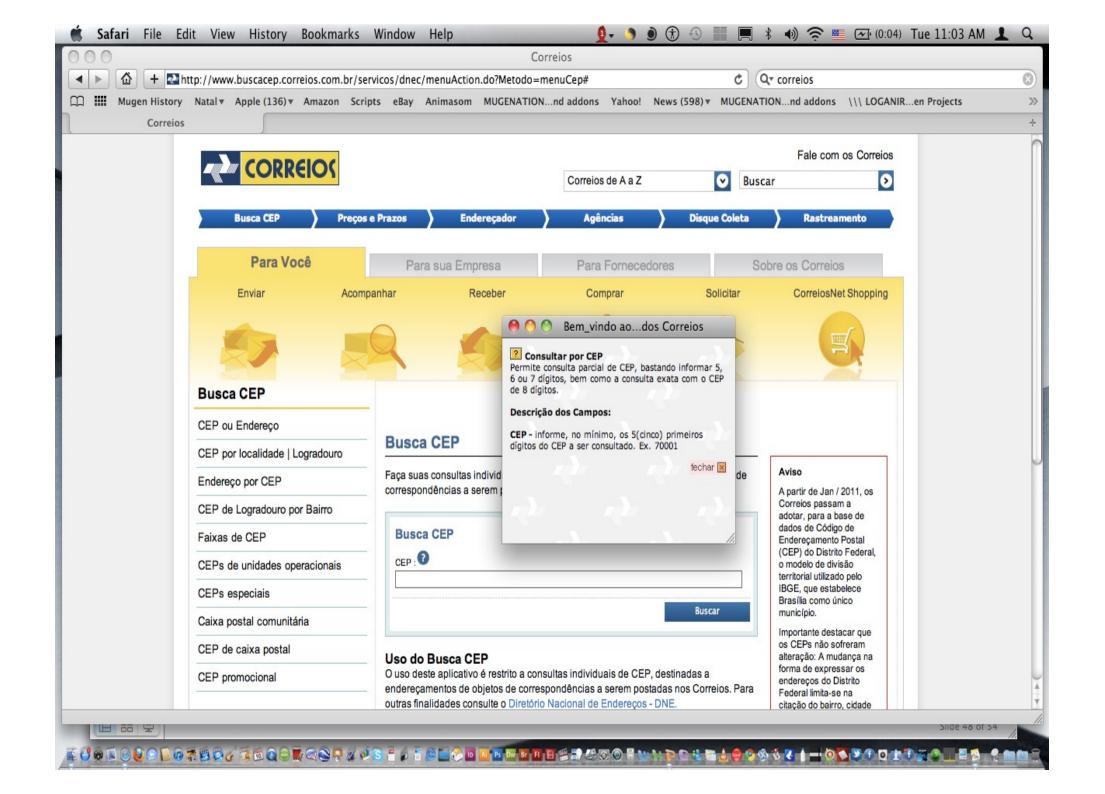
Sorry, a system error occurred.

Restart

10. Ajuda e documentação

Ainda que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário prover ajuda.

Qualquer informação deste tipo deve ser fácil de buscar, ser focada na tarefa do usuário, relacionar passos concretos a serem desenvolvido, e não ser muito longa.



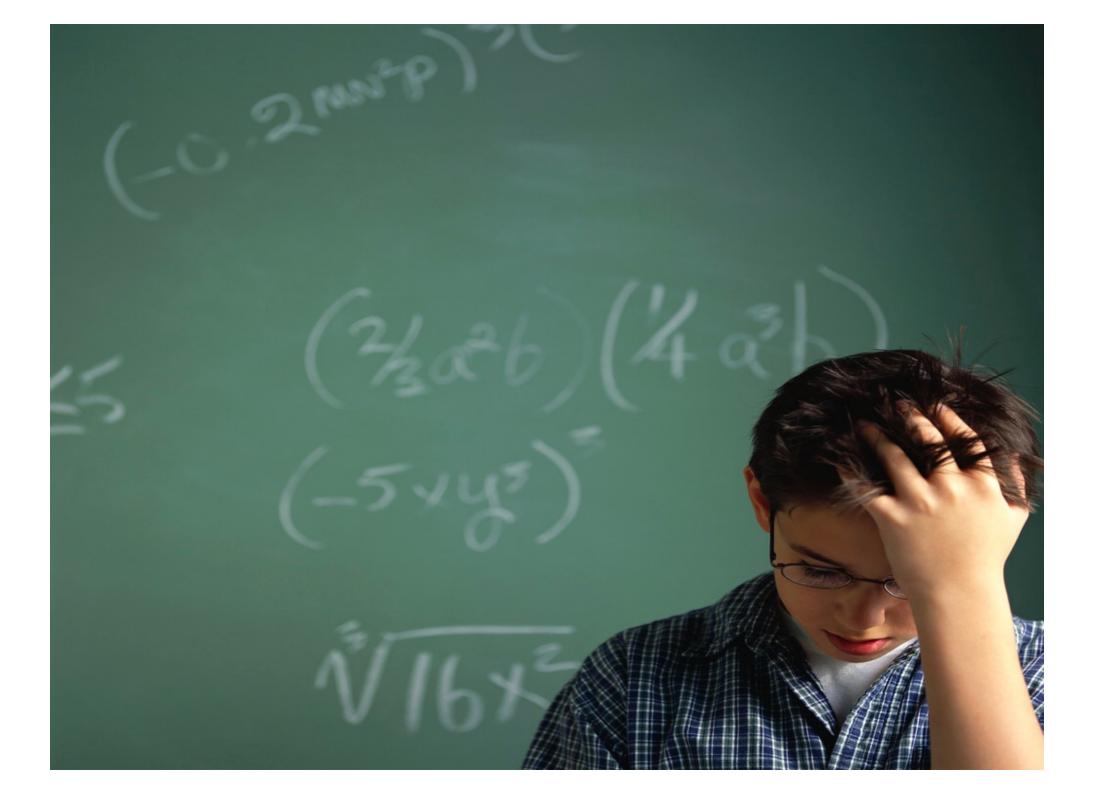
 $\int_{a}^{b} f(x) dx = F(b) - F(a)$



Gravidade dos problemas

Nielsen (1999b) destaca que a gravidade de um problema de interface é uma combinação de três fatores:

- •Freqüência implica saber se o problema é comum ou raro.
- •Impacto deve-se procurar saber se os usuários conseguem ultrapassar o problema de maneira fácil ou com dificuldade.
- •Persistência implica saber se os usuários poderão ultrapassar o problema, uma vez que tenham conhecimento de sua existência, ou serão repetidamente por ele incomodados.



Grau de severidade

Para fins de avaliação, Nielsen (1999b) estabeleceu uma escala que varia de 0 a 4 para ser usada na determinação do grau de severidade de problemas de usabilidade.

- 0 Não é encarado necessariamente como um problema de usabilidade.
- 1 Problema estético. Não necessita ser corrigido, a menos que haja tempo disponível.
- 2 Problema menor de usabilidade. Baixa prioridade para sua correção
- 3 Problema maior de usabilidade. Alta prioridade para sua correção.
- 4 Catástrofe de usabilidade: imperativo corrigi-lo.

